evropský sociální fond v ČR

EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

UNIVERSITAS MASARYKIANA BRUNENSIS

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

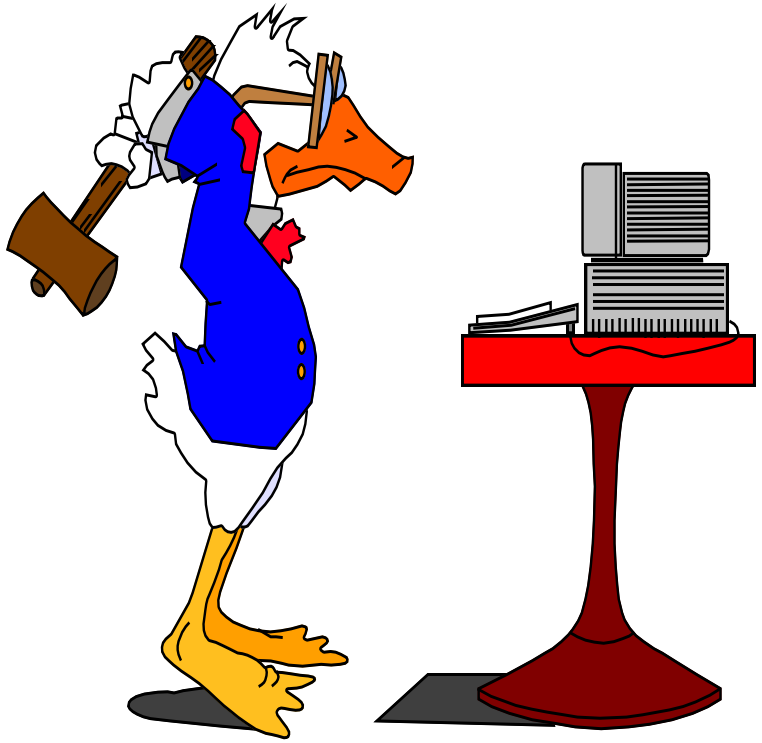
Matematické modely ve financích

Martin Řezáč

2011

Obsah:

1. Úvod do data miningu



Co je to Data Mining?

- Data mining (DM), nebo také dolování z dat či vytěžování dat, je analytická metodologie získávání netriviálních skrytých a **potenciálně užitečných informací**.

Aplikace

- Bankovnictví: schvalování úvěrů/kreditních karet
  - Predikce dobrých zákazníků.
- Pojišťovnictví: schvalování pojistných smluv
  - Odhad pravděpodobnosti pojistné události/výše škody.
- CRM (marketing):
  - Identifikace zákazníků, kteří mají v úmyslu přejít ke konkurenci.
  - Cross-selling.
  - Up-selling.
- Cílený marketing:
  - Identifikace pravděpodobných respondentů na nabídku.
- Detekce fraudu: telekomunikace, finanční transakce, pojistné podvody
  - Online/offline identifikace podvodného chování.

Aplikace

- Medicína: efektivita léčebné péče
  - Analýza pacientovy historie (předchozí nemoci a jejich průběh): nalezení vztahu mezi nemocemi.
- Farmacie: identifikace nových léků
- Vědecká analýza dat:
  - Identifikace nových galaxií.
- Design webových stránek:
  - Nalezení vztahu návštěvníka stránek a příslušná změna podoby stránek.

Aplikace

- Rozpoznávání psaného textu, řeči, obrázků.
- Supermarkety
  - Identifikace současně nakupovaného zboží
- Průmysl:
  - automatické přenastavení ovládacích prvků při změně parametrů procesu.
- Sport:
  - NBA-optimalizace herní strategie
- další…

Aplikace - Rozmístění zboží v supermarketech

- Cíl: identifikovat zboží, které je nakupováno souběžně dostatečným množstvím zákazníků.

- Výsledek: Jestliže zákazník nakupuje dětské pleny a mléko, pak si velmi pravděpodobně koupí i pivo.

- Jedna z možných interpretací:

- Správné interpretace výsledků analýz je schopen jen zkušený analytik.

Data mining a princip indukce

- Dedukce zachovává platné vztahy:
  1. Koně jsou savci.
  2. Všichni savci mají plíce.
  3. Proto platí, že všichni koně mají plíce.

- Indukce přidává informace:
  1. Všichni doposud pozorovaní koně mají plíce.
  2. Proto platí, že všichni koně mají plíce.

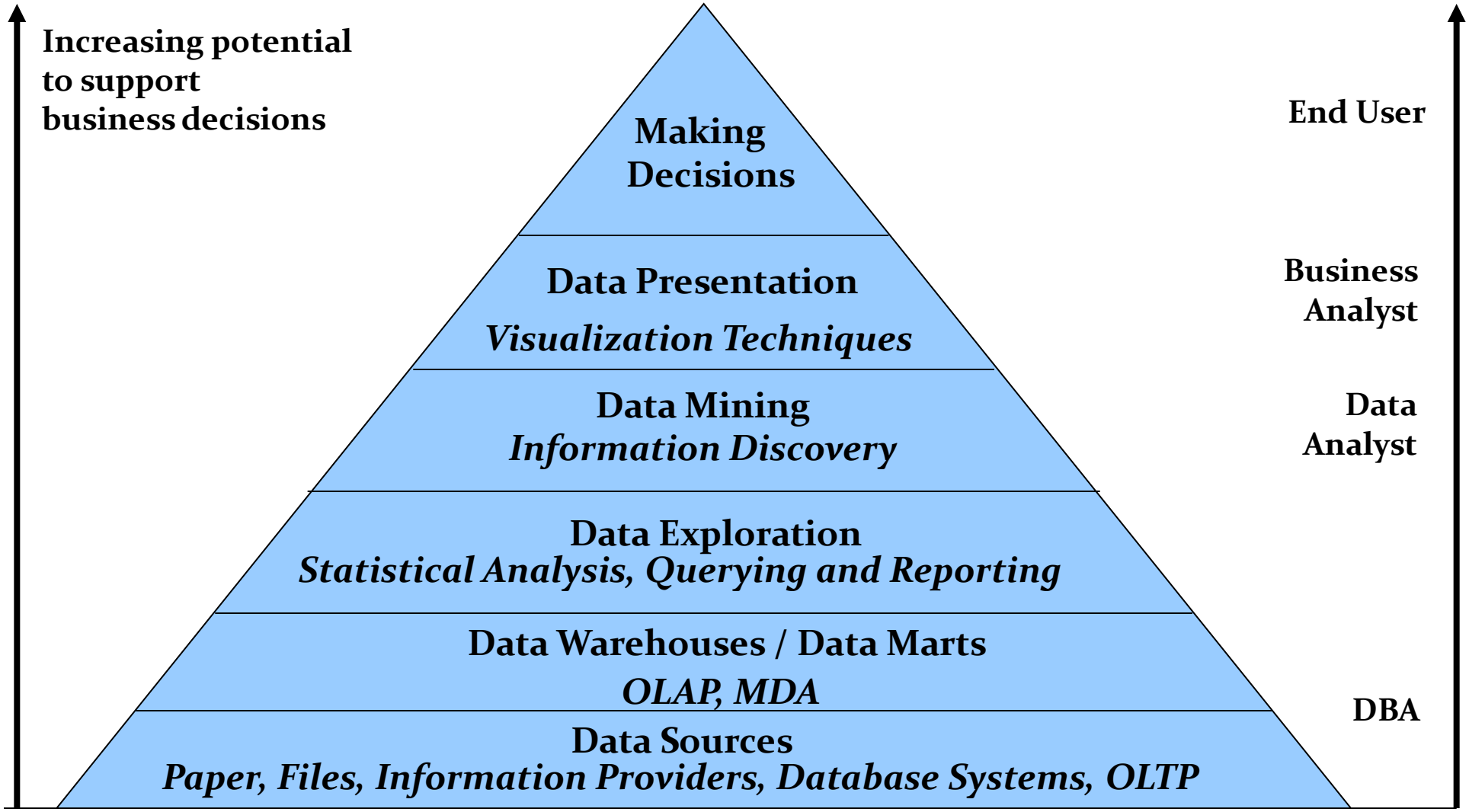
Problém s indukcí

- Z platných faktů můžeme vyvodit nepravdivé tvrzení (model).

- Příklad:
  - Evropské labutě jsou bílé
  - Indukce: „Labutě jsou bílé" jakožto obecné pravidlo.
  - Objevením Austrálie se objevili i černé labutě…
  - Problém: množina pozorování nebyla náhodná a tudíž reprezentativní.

http://cs.wikipedia.org/wiki/Labu%C5%A5_%C4%8Dern%C3%A1

Data mining –podpora business rozhodování



Historie názvu

1960 Data Fishing, Data Dredging (bagrování):

- užíváno statistiky

1989 Knowledge Discovery (KD, KDD):

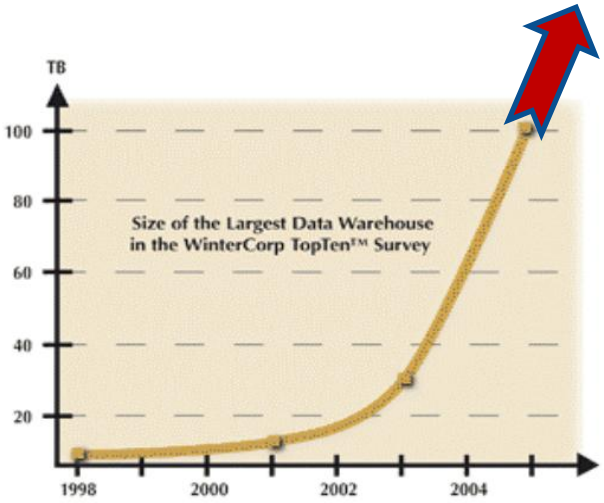
- užíváno komunitou zabývající se umělou inteligencí a strojovým učením

1990 Data Mining (DM):

- užíváno v komerční sféře a databázové komunitě

Další názvy: Data Archaeology, Information Harvesting, Information Discovery, Knowledge Extraction, …

Data mining – nutnost?



Největší světové databáze v r. 2005:

- Max Planck Inst. for Meteorology ~ 222 TB
- Yahoo ~ 100 TB
- AT&T ~ 94 TB

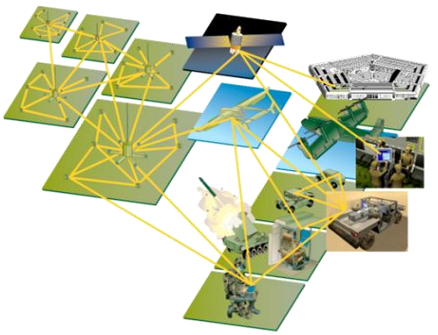
V roce 2008:

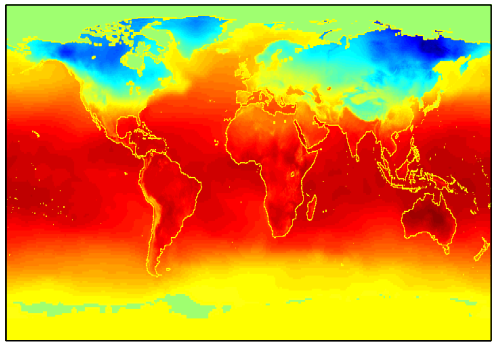
- Max Planck Inst. for Meteorology ~ 6000 TB
- Yahoo ~ 2000 TB

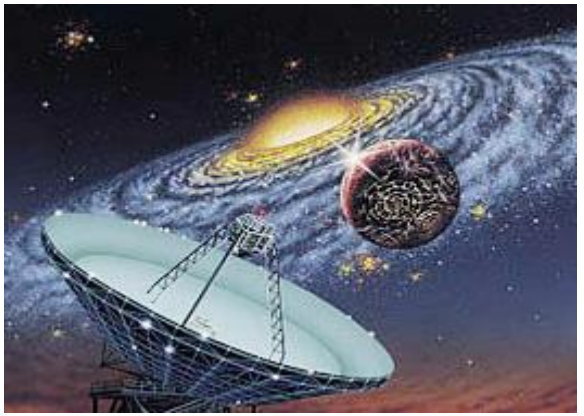
Data mining – nutnost?

- Terabytes -- 10^12 bytes: data obchodních řetězců, bank,…
- Petabytes -- 10^15 bytes: geografická data
- Exabytes -- 10^18 bytes: národní databáze zdravotních záznamů
- Zettabytes -- 10^21 bytes: databáze meteo-snímků
- Zottabytes -- 10^24 bytes: video-databáze

Data mining – nutnost?







Proč data mining? Proč dnes?

- Data jsou produkována.
- Data jsou skladována.
- Výpočetní síla je dostupná.
- Výpočetní síla je cenově dostupná.
- Konkurenční tlak je velice silný.
- Komerční produkty (DM software) jsou k dispozici.

Data mining vs. Statistická analýza

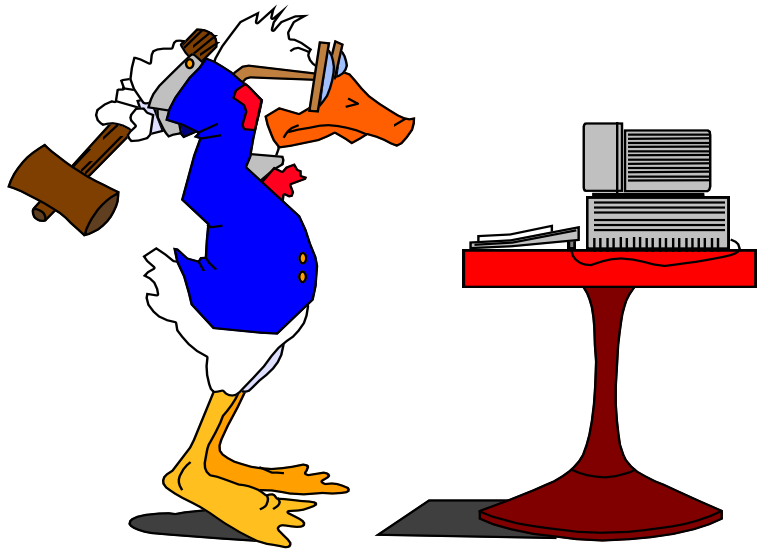
- **Data Mining**
  - Původně vyvinuto pro expertní systémy automaticky řešící zadané problémy.
  - Neklade takový důraz na přesné porozumění použité metody.
  - Pokud něco dává smysl, pak to použijme!
  - Žádné předpoklady o datech.
  - Funguje i pro velmi rozsáhlá data.
  - Vyžaduje porozumění problému z datovému a business pohledu.
- **Statistická analýza**
  - Testuje se statistická korektnost modelu.
    - Jsou statistické předpoklady modelu splněny?
  - Testování hypotéz.
  - Intervalové odhady.
  - Pracuje se s výběrem hodnot.
  - Standardní metody nejsou optimalizovány pro rozsáhlá data.
  - Vyžaduje pokročilé statistické znalosti.

Data mining

- Proces (polo-) automatické analýzy (rozsáhlých) databází k identifikaci vztahů, které jsou:
  - validní: platí na nových datech s určitou jistotou obecné platnosti
  - nové: doposud neznámé
  - užitečné: dají se v praxi nějak použít
  - srozumitelné: (vždy) se nalezený vztah dá nějak vysvětlit

Data mining není:

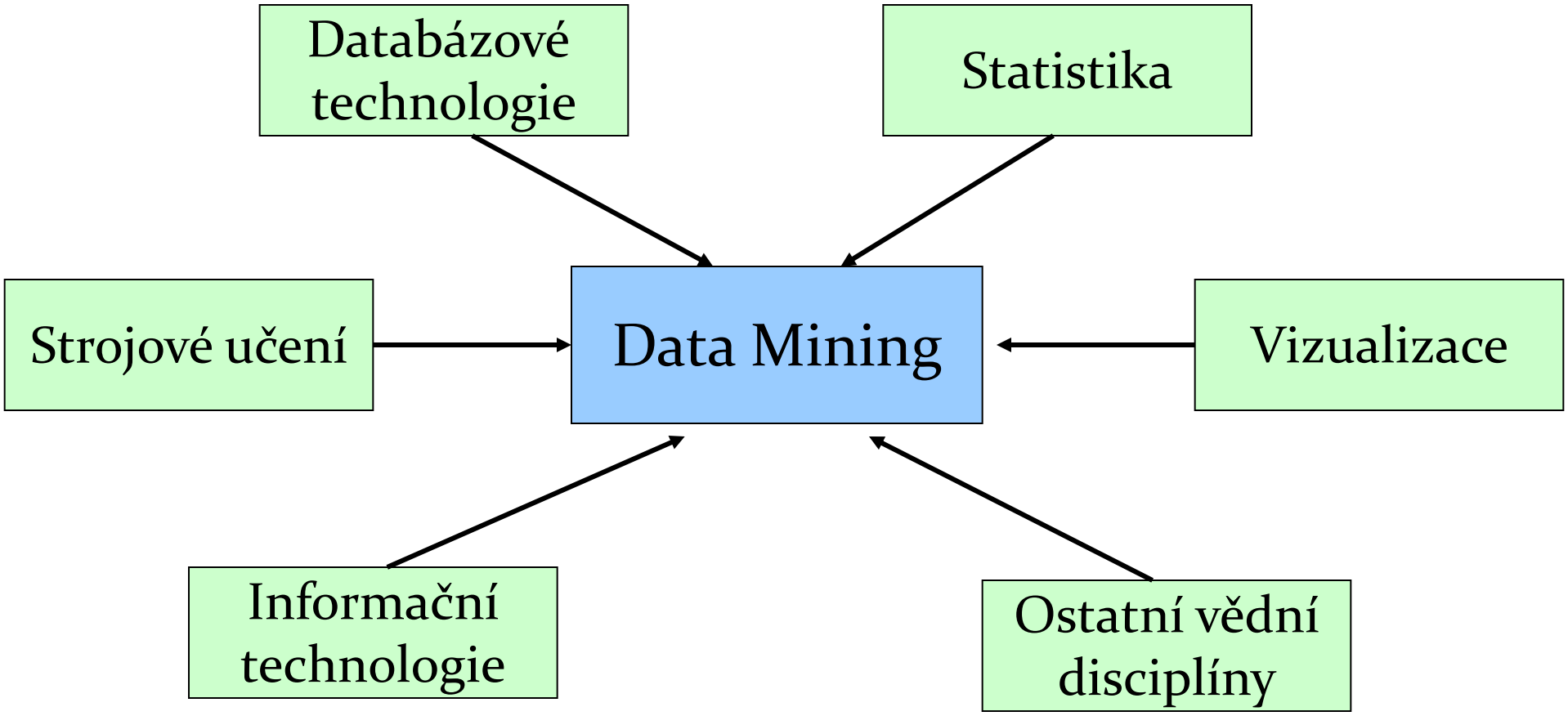
- **Brutální hromadné zpracování dat.**
- **Slepé použití algoritmů.**
- **Hledání vztahů tam, kde žádné neexistují.**



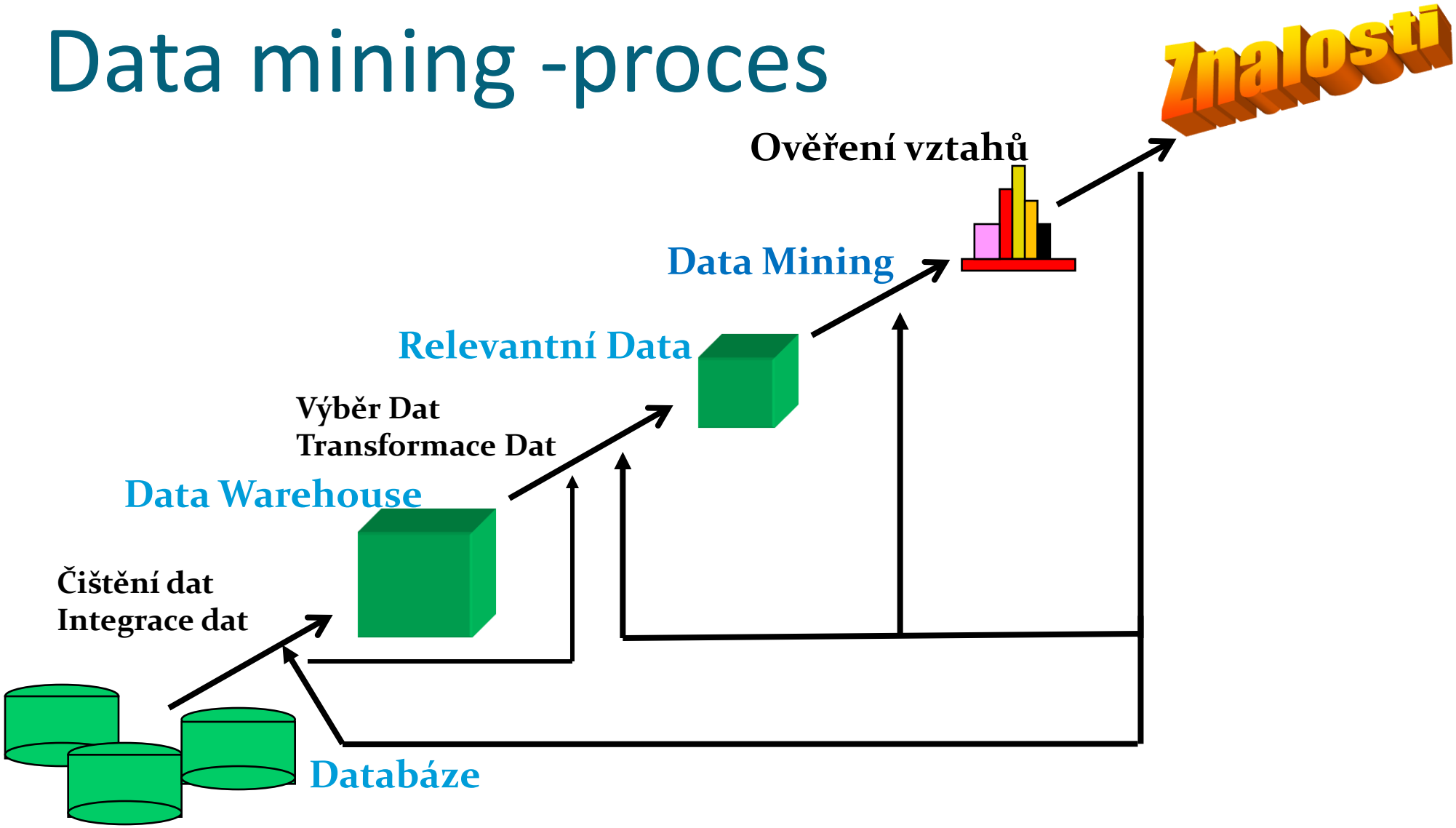
Známé ≠ Zajímavé

- Zajímavé jsou ty vztahy, které se liší od obecných očekávání.
- Data mining se vyplácí právě díky objevování dosud neznámých a překvapivých vztahů.

Vztah s ostatními disciplínami

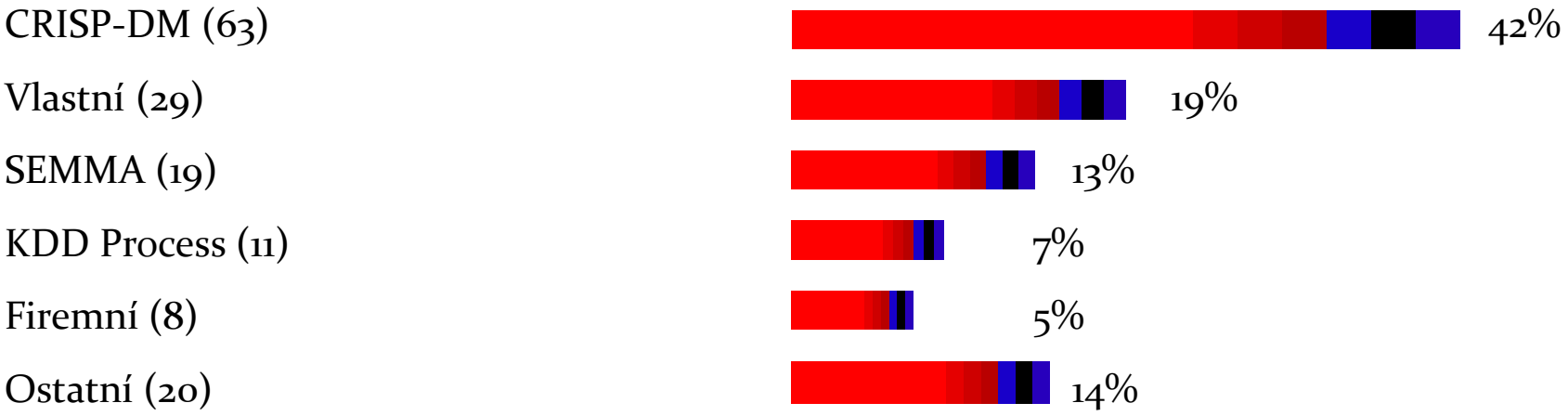


Data mining -proces



Data Mining Methodology (2007)

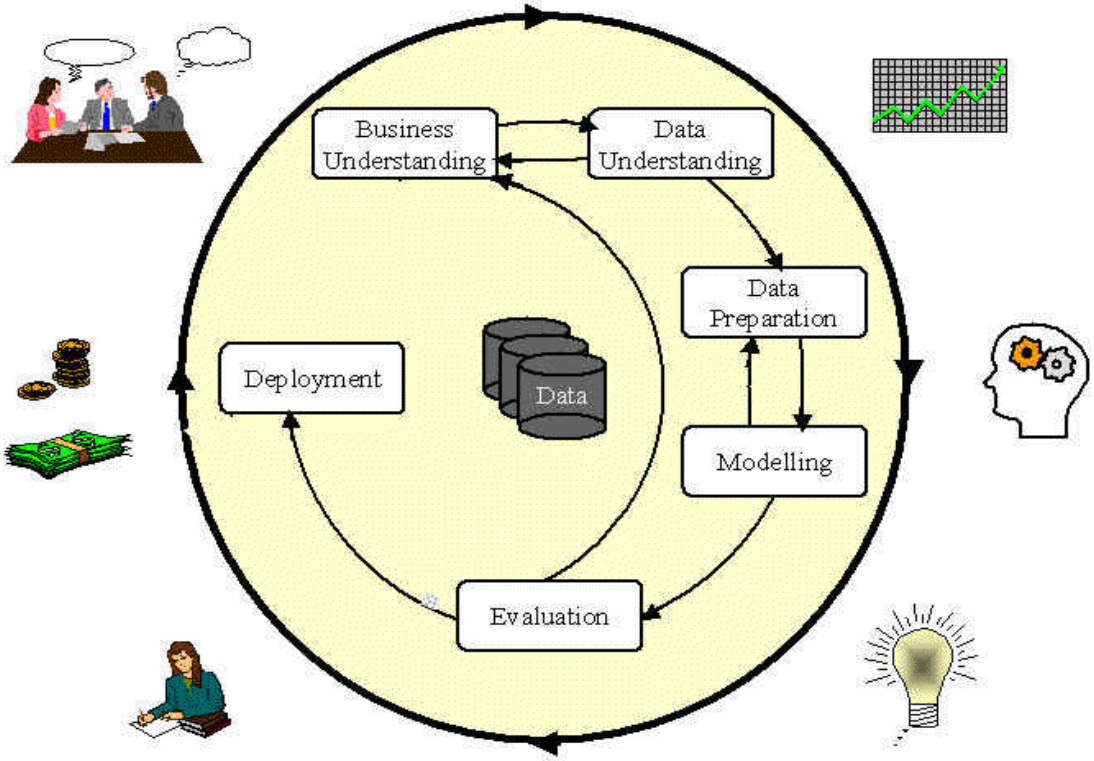
Kterou metodologii používáte pro data mining?



CRISP-DM

(CRoss Industry Standard Process for Data Mining)

1. pochopení obchodních souvislostí
2. pochopení dat
3. příprava dat
4. modelování
5. vyhodnocení modelu
6. nasazení modelu do obchodního procesu



http://community.udayton.edu/provost/it/training/documents/SPSS_CRISPWPlr.pdf

SEMMA

(Sample, Explore, Modify, Model, Assess)

• **Sample** - identifikovat vhodná učící data, určit odpovídající rozsah dat, a to jak z pohledu časového okna tak i z pohledu počtu případů. Dále se doporučuje rozdělit data na 3 skupiny:

Trénovací – využívá se pro vývoj modelu.

Validační – využívá se pro vyhodnocení modelu a pro prevenci proti přeučení (over fitting) modelu.

Testovací – využívá se pro finální vyhodnocení modelu. Zajímá nás především jak dobře se model chová na datech disjunktních s daty, na kterých byl model vyvinut.

• **Explore** - připravit popisné statistiky, které poskytnou základní představu o obsahu a kvalitě podkladových dat. Pomocí vizualizačních technik odhalit skryté trendy a závislosti v datech.

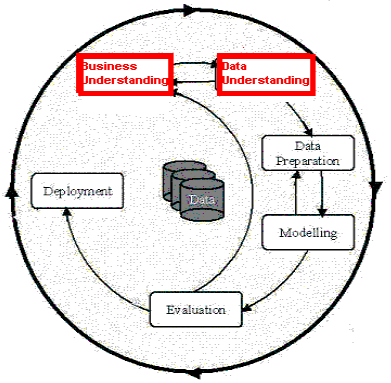
• **Modify** - na základě předchozího kroku konsolidovat data a odvodit nové proměnné. Následně transformovat data do tvaru vhodného pro modelování.

• **Model** - vytvořit příslušný model. Mezi často používané techniky patří např. neuronové sítě, rozhodovací stromy, logistické modely.

• **Assess** - vyhodnotit úspěšnost modelu a případně implementovat model do praxe.

Fáze DM procesu (1 & 2)

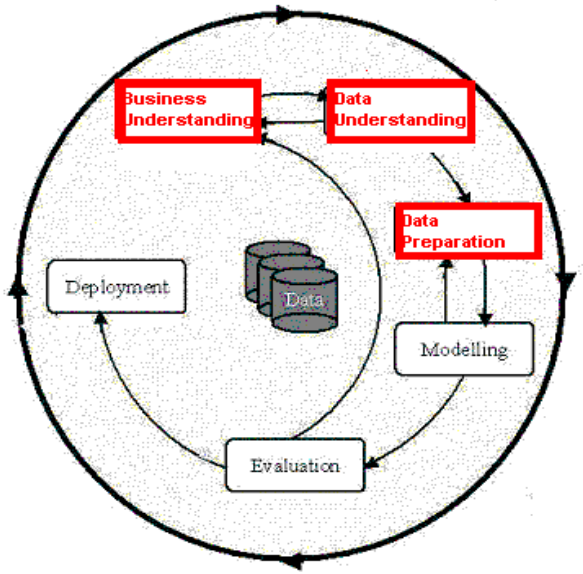
- Porozumění obchodu (Business Understanding):
  - Stanovení business cílů.
  - Stanovení data miningových cílů.
  - Statnovení kriterií úspěchu.



- Porozumění datům (Data Understanding):
  - Průzkum dat a ověření jejich kvality.
  - Nalezení odlehlých hodnot.

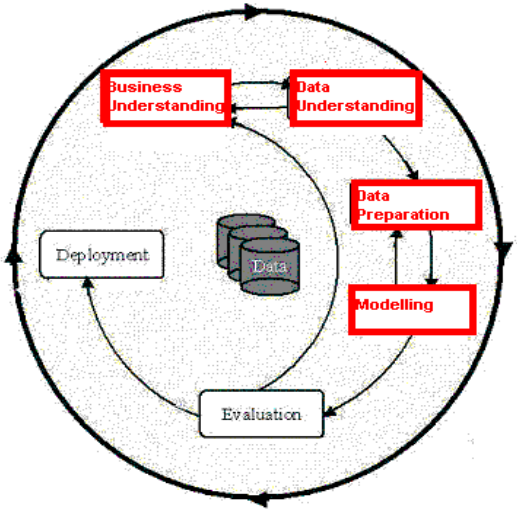
Fáze DM procesu (3)

- Příprava dat (Data preparation):
  - Obvykle zabírá přes 90% celkové času.
    - Sběr dat
    - Konsolidace a čištění
      - Vazební tabulky, agregace, chybějící hodnoty,…
    - Selekce
      - Ignorování neužitečných dat?
      - Odlehlá pozorování?
      - Výběr dat?
      - Vizualizační nástroje.
    - Transformace – vytváření nových odvozených proměnných



Fáze DM Procesu (4)

- Modelování (Model building)
  - Výběr vhodných modelovacích technik závisí na stanovených data miningových cílech.
  - Modelování je většinou iterační proces propojený s přípravou dat
  - Rozdílný přístup pro „*supervised*“ a „*unsupervised learning*“

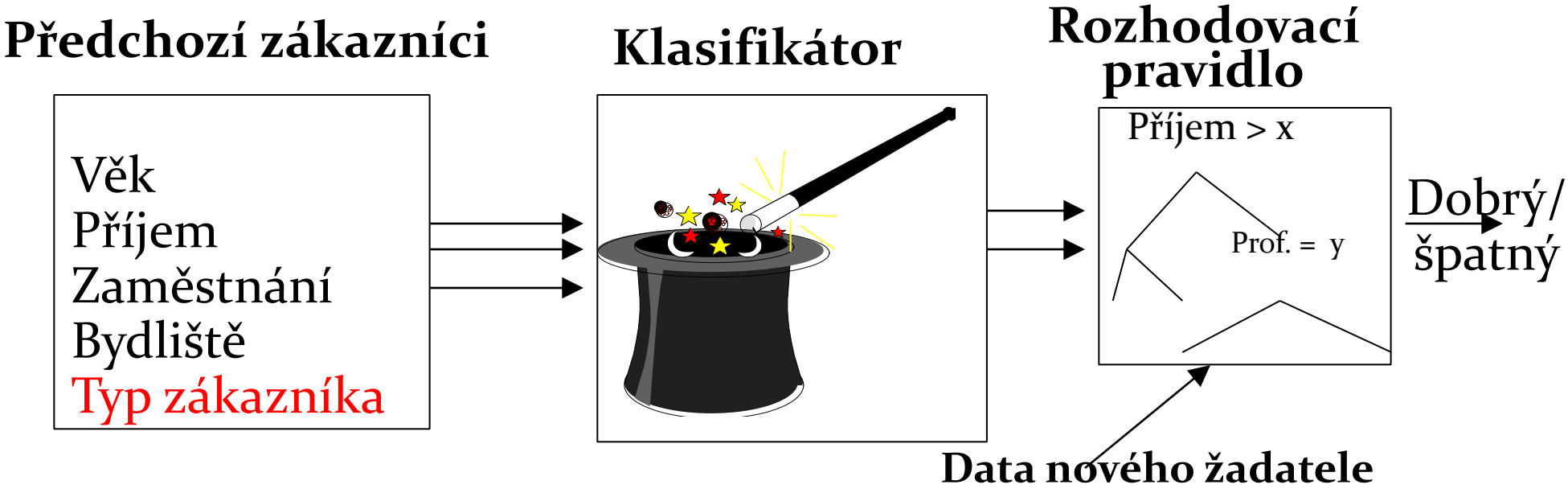


Základní přístupy k modelování

- Prediktivní: jde o matematický model předpovídající (s určitou přesností) budoucí hodnotu/chování nějaké veličiny (entity).
  - Regrese/ Klasifikace
  - Analýza časových řad
- Deskriptivní: jde o matematický model popisující historické události a předpokládané nebo reálné vazby mezi nimi.
  - Klastrová (shluková) analýza
  - Asociační pravidla
  - Detekce deviací/zlomů
  - Faktorová analýza / analýza hlavních komponent

Klasifikace

- Na základě známých údajů o „starých“ zákaznících a jejich platební morálce máme predikovat platební způsobilost nového žadatele o úvěr.



Klasifikační metody

- Cíl: Predikovat třídu Ci = f(x1, x2, .. Xn)
- Regrese: (lineární nebo polynomiální)
  - a*x1 + b*x2 + c = Ci
- Metody nejbližšího souseda (KNN)
- Rozhodovací stromy
- Pravděpodobnostní modely (GLM) – např. logistická regrese.
- Diskriminační analýza (LDA,...)
- Neuronové sítě
- Support vector machines (SVM)
- Bayesovské modely

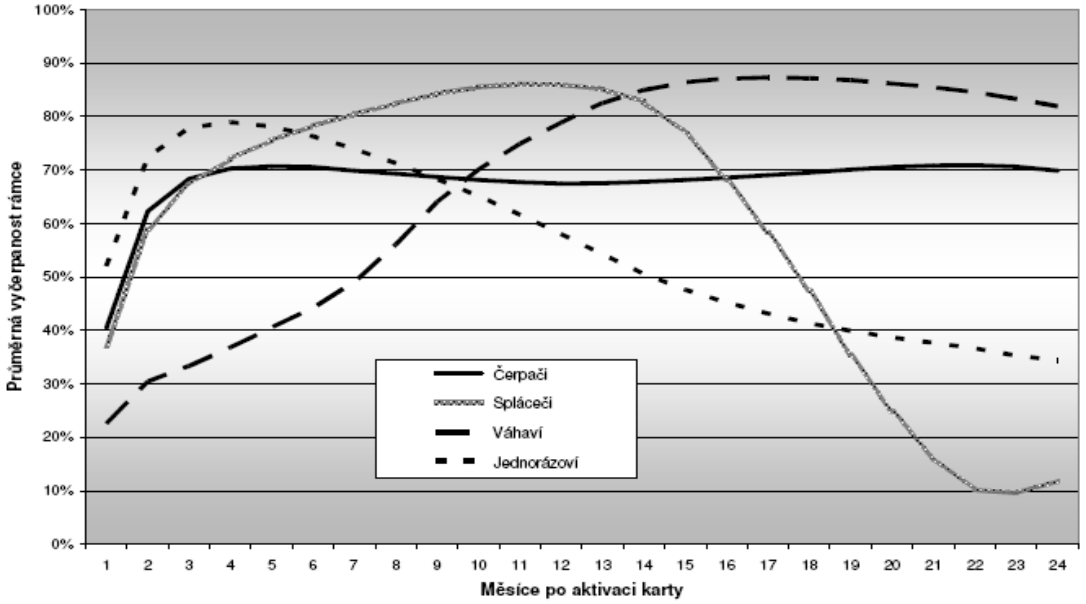
Deskriptivní modelování

- Základním cílem je získání ucelených a snadno srozumitelných informací z dostupných dat.

- Někdy součástí průzkumové (explorační) analýzy předcházející prediktivnímu modelování, někdy je vytvoření deskriptivního modelu hlavním cílem DM projektu.

Klastrová analýza

- Máme nalézt skupiny/ klastry stávajících zákazníků na základě platební historie tak, aby podobní klienti byli ve stejné skupině/ klastru.
- Základní požadavek: Kvalitní míra podobnosti (http://cs.wikipedia.org/wiki/Shluková_analýza).



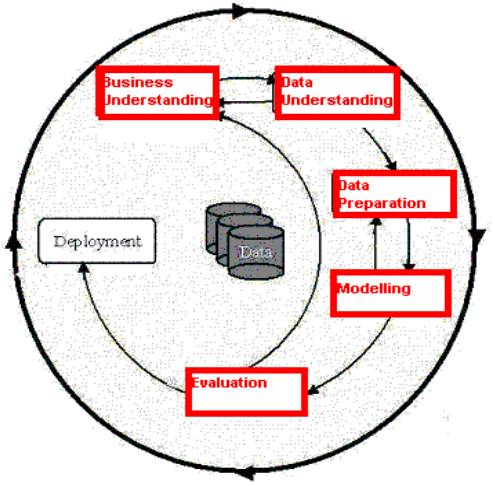
Zdroj: NEPIL, M. *Data mining v praxi.* Brno : MU v Brně, 2007. s 25-38.

Supervised vs. unsupervised learning

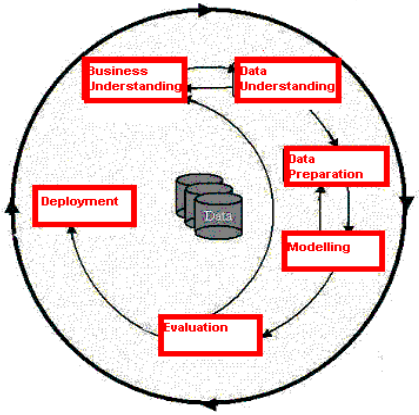
- Supervised learning:
  - Supervize: Data (pozorování, měření, atp.) jsou označena předem definovanými/známými třídami.
  - Nová/testovací data jsou následně rozřazena do těchto tříd.
  - Z pohledu kauzality daný model definuje vztah mezi vstupními daty a daty výstupními.
- Unsupervised learning:
  - Předem nejsou definované žádné třídy.
  - Pro daná data je cílem prokázat existenci nějakých tříd.
  - Z pohledu kauzality jsou všechna data chápána jako výstupní. Modelujeme závislost daných dat na jakýchsi neznámých skrytých proměnných.

Fáze DM Procesu (5)

- Vyhodnocení modelu (Model Evaluation):
  - Evaluace modelu: jak se chová na testovacích datech.
  - Metody a kritéria závisí na typu modelu:
    - Např. koincidenční matice pro klasifikační modely, průměrná chyba pro regresní modely,...
  - Interpretace modelu: důležitost a obtížnost interpretace značně závisí na zvolené modelovacím algoritmu.



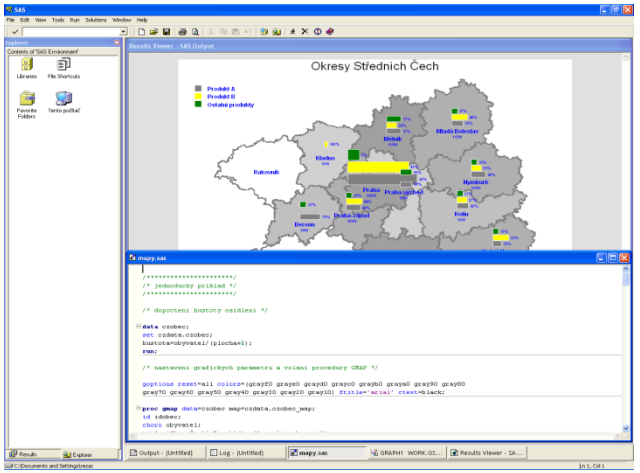
Fáze DM Procesu (6)



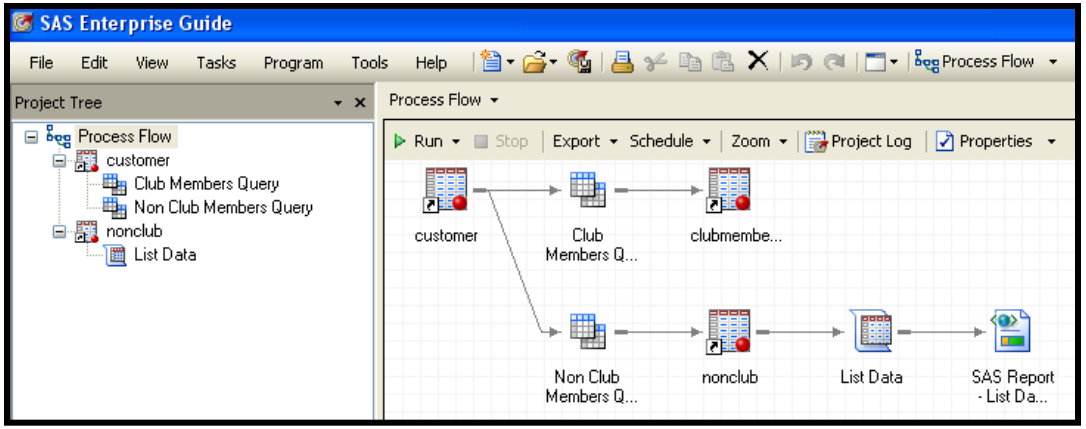
- Nasazení do praxe (Deployment)
  - Je třeba určit, jak mají být výsledky využity.
  - Kdo je bude využívat?
  - Jak často budou využívány?
- Nasazení data miningových výsledků pomocí:
  - Skórování databáze.
  - Využití výsledků pomocí obchodních pravidel.
  - Interaktivní on-line scoring.
  - …

SAS - stručné seznámení

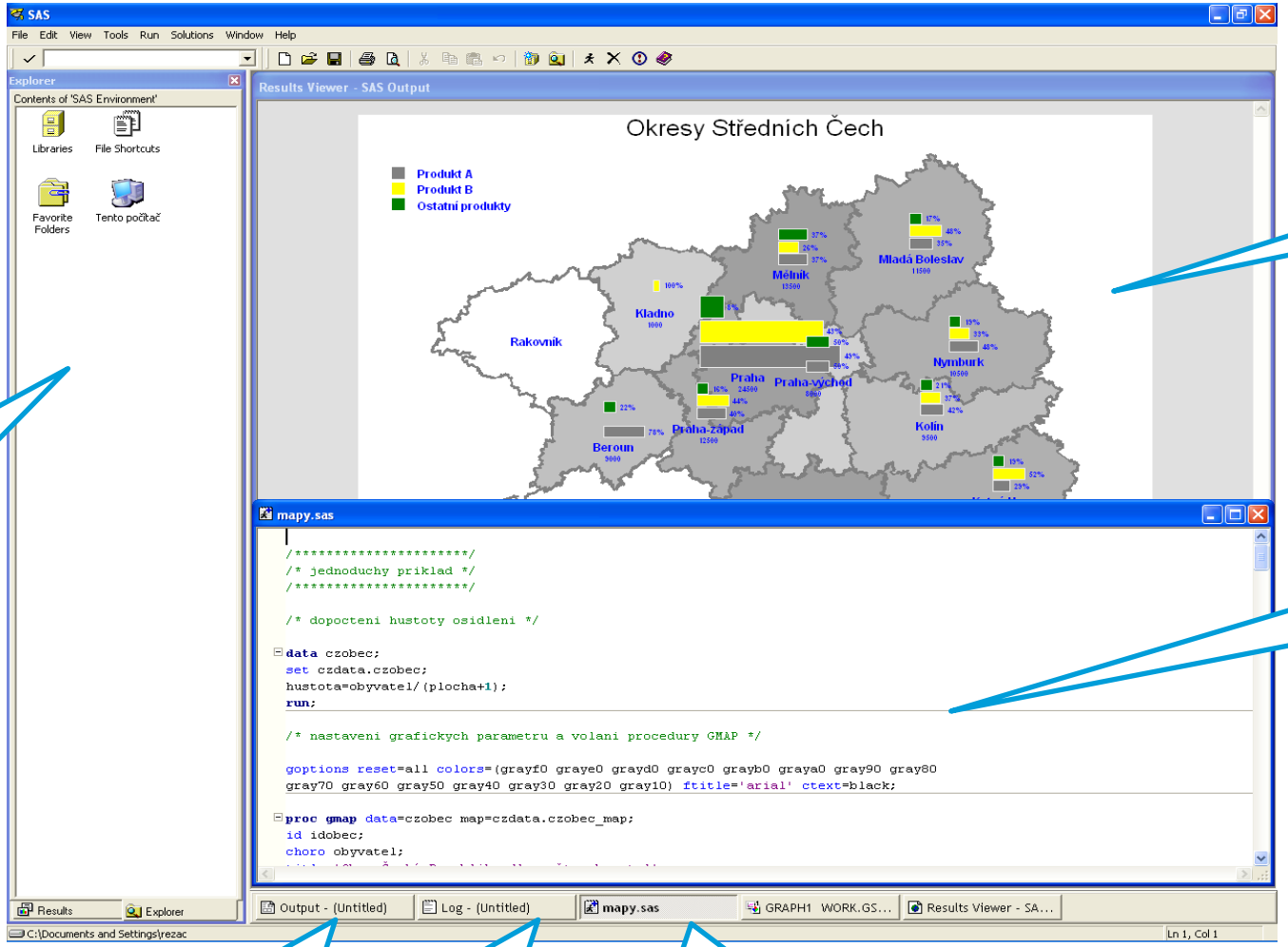
- 2 základní SAS rozhraní:
  - SAS windowing environment



  - SAS Enterprise Guide (GUI)



SAS - stručné seznámení



SAS Output

SAS Explorer window

Program editor window

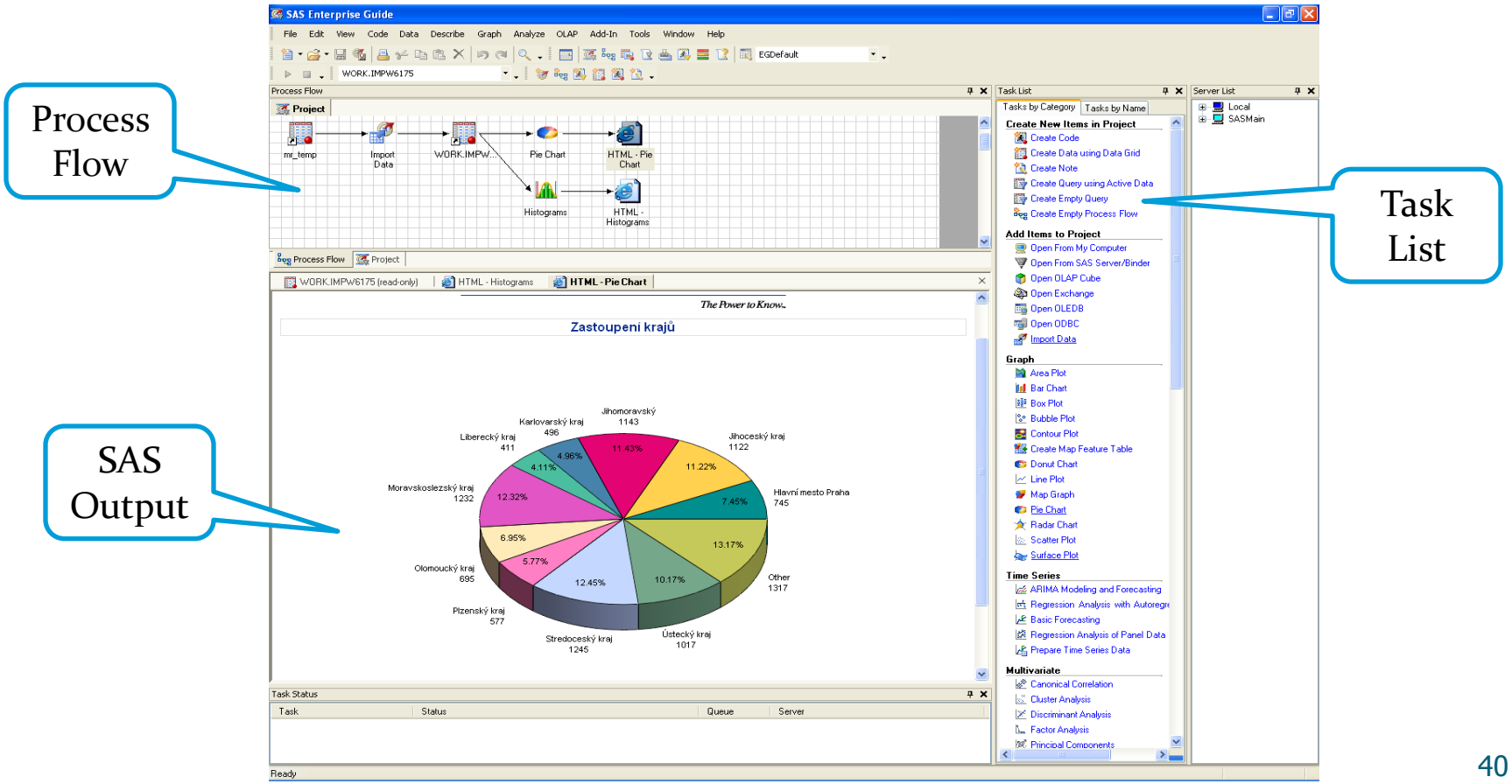
Output tab

Log tab

Editor tab

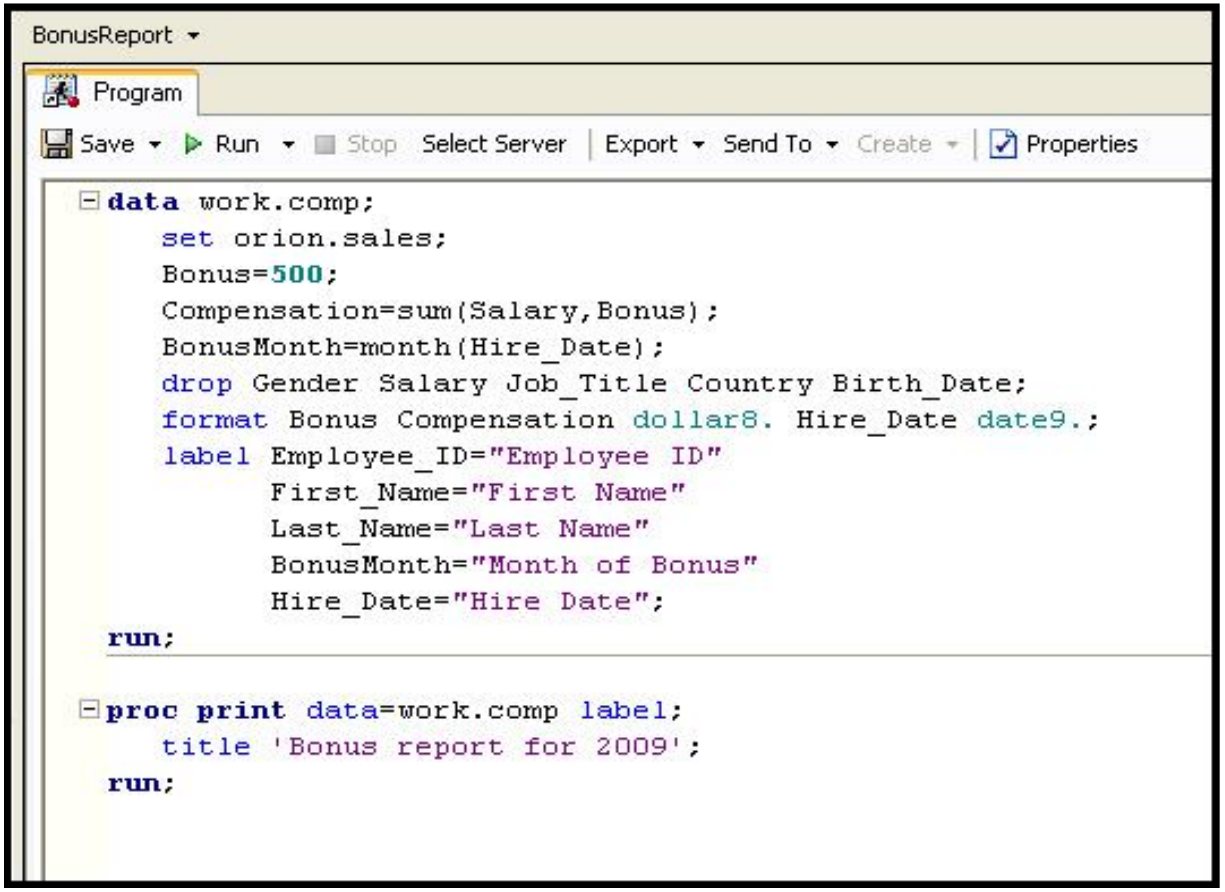
SAS - stručné seznámení

- Pomocí klikání a přetahování myší je budován procesní tok.



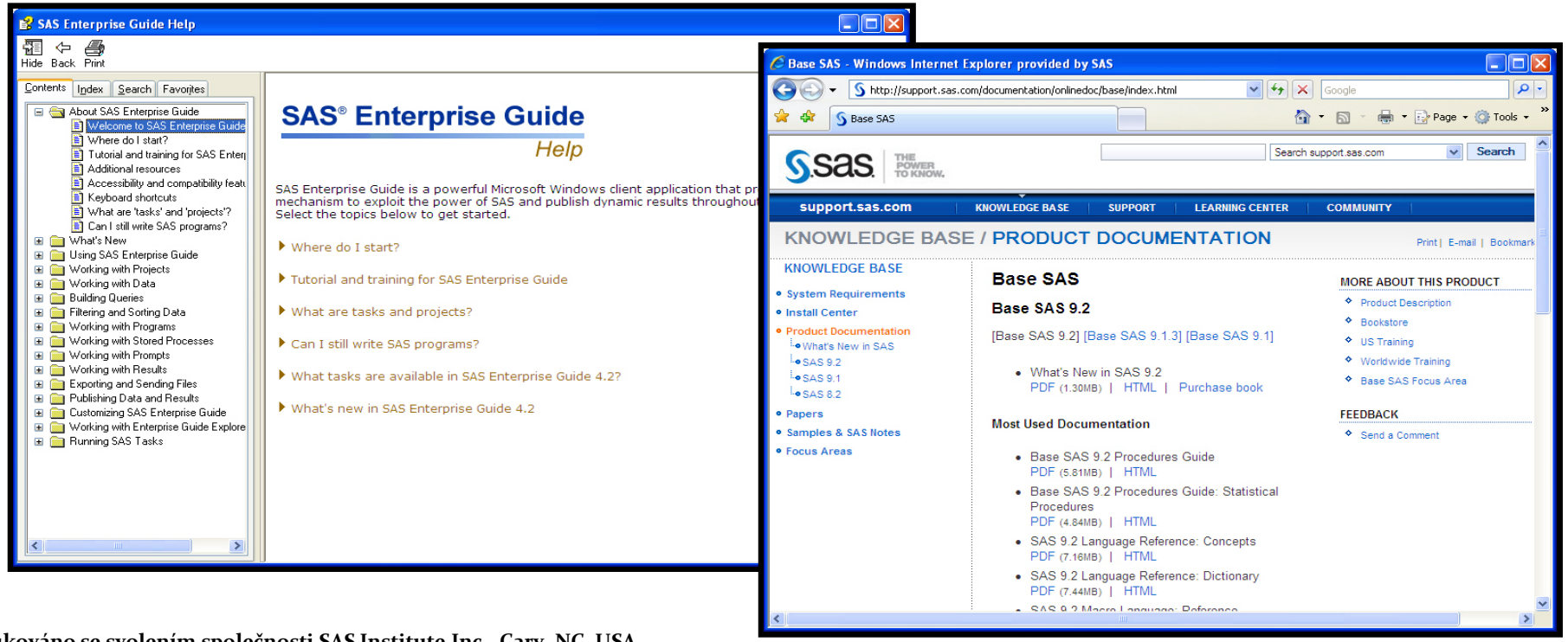
SAS Enterprise Guide (EG) Interface

- EG automaticky generuje kód, který možné dále editovat



SAS Help

- Use the SAS Enterprise Guide Help facility or SAS OnlineDoc for additional direction on SAS Enterprise Guide or the SAS programming language. Go to support.sas.com and select
- **Product Documentation** ⇨ **Base SAS**.



SAS na webu

Michal Kulich: *Malý manuál uživatele SASu*
http://www.karlin.mff.cuni.cz/~kulich/sas/SASMain.html

Phil Spector: *An Introduction to the SAS System*
http://www.stat.berkeley.edu/classes/s100/sas.pdf

Patric McLeod : *Introduction to SAS 9*
http://www.unt.edu/rss/class/sas1/

http://en.wikipedia.org/wiki/SAS_%28software%29

2. Organizace dat, úvod do SQL

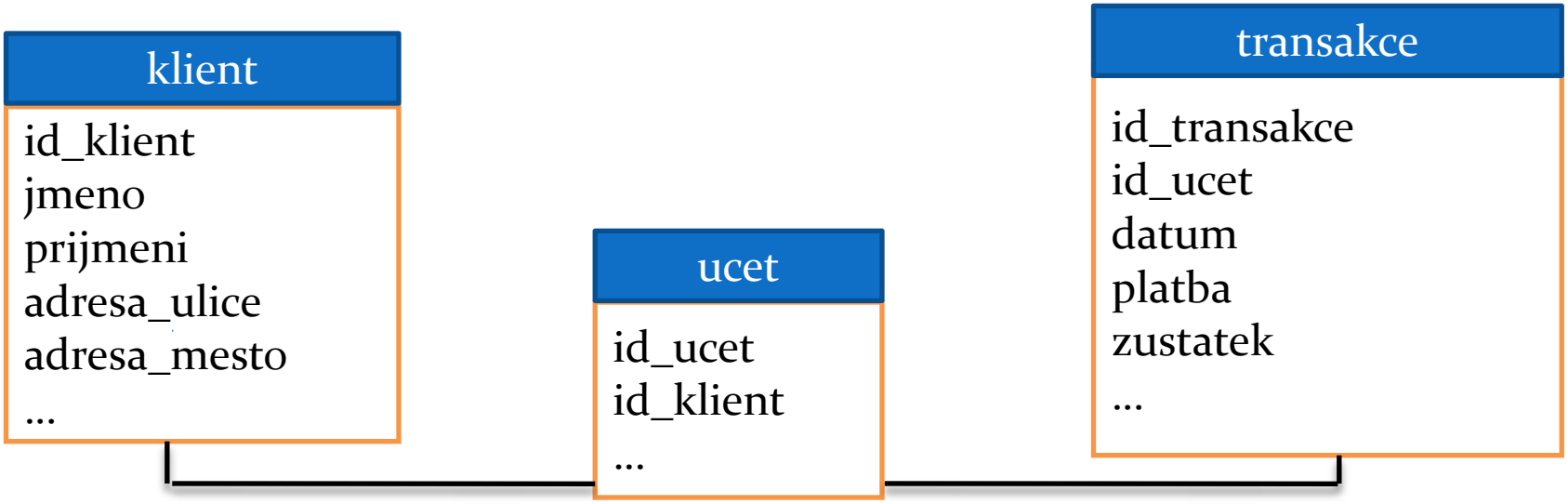
Historie skladování dat

V minulosti byla data ukládána v jednom velkém souboru, ke kterému se přistupovalo indexovanými sekvenčními metodami. Soubor byl indexován na základě předpokládaných způsobů dotazování. Velkou nevýhodou bylo to, že se informace v záznamech opakovaly a typy dotazů byly předurčeny.

Historie skladování dat

datum	jmeno	prijmeni	adresa_ulice	adresa_mesto	cislo_uctu	platba	zustatek
980103	Jan	Novak	Dlouha 5	Praha 1	9945371	100,00	100,00
980105	Jan	Novak	Dlouha 5	Praha 1	9945371	1500,00	1600,00
980106	Jan	Novak	Dlouha 5	Praha 1	9945371	-1500,00	50,00
980106	Karel	Nemec	Lucni 4	Praha 2	24867134	3000,00	6000,00
980107	Karel	Nemec	Lucni 4	Praha 2	24867134	-4000,00	2000,00
980108	Jan	Novak	Dlouha 5	Praha 1	9945371	-150,00	-100,00
980111	Karel	Nemec	Lucni 4	Praha 2	24867134	5000,00	7000,00

Relační databáze




```
SELECT klient.jmeno, klient.prijmeni, klient.adresa_ulice,
klient.adresa_mesto, ucet.cislo_uctu, transakce.zustatek
FROM klient, ucet, transakce
WHERE klient.id_klient = ucet.id_klient;
AND transakce.id_ucet = ucet.id_ucet;
AND transakce.zustatek < 100;
GROUP BY klient.adresa_mesto;
```

Relační databáze

- **Relační databáze** je databáze založená na relačním modelu. Často se tímto pojmem označuje nejen databáze samotná, ale i její konkrétní softwarové řešení.
- Relační databáze je založena na tabulkách, jejichž řádky obvykle chápeme jako záznamy a eventuelně některé sloupce v nich (tzv. cizí klíče) chápeme tak, že uchovávají informace o relacích mezi jednotlivými záznamy v matematickém slova smyslu.
- Termín *relační databáze* definoval Edgar F. Codd v roce 1970.

- způsoby kladení dotazů:
  - QBE (query by example)
  - SQL (structured query language)

Relační databáze

- Dle relační teorie lze pomocí základních operací (sjednocení, kartézský součin, rozdíl, selekce, projekce a spojení) uskutečnit veškeré operace s daty a ostatní operace jsou již jen kombinacemi těchto pěti.

Relační databáze

- Základem relačních databází jsou databázové tabulky. Jejich sloupce se nazývají atributy nebo pole, řádky tabulky jsou pak záznamy. Atributy mají určen svůj konkrétní datový typ - doménu. Řádek je řezem přes sloupce tabulky a slouží k vlastnímu uložení dat. Konkrétní tabulka pak realizuje podmnožinu kartézského součinu možných dat všech sloupců - relaci.
- **Primární klíč**
  - Primární klíč je jednoznačný identifikátor záznamu, řádku tabulky. Primárním klíčem může být jediný sloupec či kombinace více sloupců tak, aby byla zaručena jeho jednoznačnost. Pole klíče musí obsahovat hodnotu, tzn. nesmí se zde vyskytovat nedefinovaná prázdná hodnota NULL. V praxi se dnes často používají umělé klíče, což jsou číselné či písmenné identifikátory - každý nový záznam dostává identifikátor odlišný od identifikátorů všech předchozích záznamů (požadavek na unikátnost klíče), obvykle se jedná o celočíselné řady a každý nový záznam dostává číslo vždy o jednotku vyšší (zpravidla zcela automatizovaně) než je číslo u posledního vloženého záznamu (číselné označení záznamů s časem stoupá).
- **Cizí klíč**
  - Dalším důležitým pojmem jsou nevlastní/cizí klíče. Slouží pro vyjádření vztahů, relací, mezi databázovými tabulkami. Jedná se o pole či skupinu polí, která nám umožní identifikovat, které záznamy z různých tabulek spolu navzájem souvisí.

Relační databáze – vztahy mezi tabulkami

- Vztahy, neboli relace, slouží ke svázání dat, která spolu souvisejí a jsou umístěny v různých databázových tabulkách. V zásadě rozlišujeme čtyři typy vztahů.

- mezi daty v tabulkách není žádná spojitost, proto nedefinujeme žádný vztah.

- 1:1 používáme, pokud záznamu odpovídá právě jeden záznam v jiné databázové tabulce a naopak. Takovýto vztah je používán pouze ojediněle, protože většinou není pádný důvod, proč takovéto záznamy neumístit do jedné databázové tabulky. Jedno z mála využití je zpřehlednění rozsáhlých tabulek. Jako ilustraci je možné použít vztah řidič - automobil. V jednu chvíli (diskrétní časový okamžik) řídí jedno auto právě jeden řidič a zároveň jedno auto je řízeno právě jedním řidičem.

Relační databáze – vztahy mezi tabulkami

- 1:N přiřazuje jednomu záznamu více záznamů z jiné tabulky. Jedná se o nejpoužívanější typ relace, jelikož odpovídá mnoha situacím v reálném životě. Jako reálný příklad může posloužit vztah autobus - cestující. V jednu chvíli cestující jede právě jedním autobusem a v jednom autobuse může zároveň cestovat více cestujících.
- M:N je méně častým. Umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé. V databázové praxi bývá tento vztah z praktických důvodů nejčastěji realizován kombinací dvou vztahů 1:N a 1:M, které ukazují do pomocné tabulky složené z kombinace obou použitých klíčů (třetí resp. tzv. vazební tabulka). Příkladem z reálného života by mohl být vztah výrobek - vlastnost. Výrobek může mít více vlastností a jednu vlastnost může mít více výrobků. V reálném životě nicméně existuje velké množství vztahů M : N, mimo jiné také proto, že často existuje praktická potřeba zachovávat i údaje o historii těchto vztahů z časového hlediska (jeden řidič v delším časovém období řídí více rozličných aut a jedno auto v delším časovém období může mít více různých řidičů).

Slovník pojmů

ODS	Operational Data Store
DWH	DataWareHouse
DataMart	
Meta Data	
BI	Business Intelligence
OLAP	On Line Analytical Processing
OLTP	On Line Transaction Processing
ETL	Extract, Transform, Load
ELT	Extract, Load, Transform
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
DBMS	Database Management System
SQL	Structured Query Language

Slovník pojmů

ODS: Short for ***o****perational* ***d****ata* ***s****tore*, a type of database that serves as an interim area for a data warehouse in order to store time-sensitive operational data that can be accessed quickly and efficiently. In contrast to a data warehouse, which contains large amounts of static data, an ODS contains small amounts of information that is updated through the course of business transactions. An ODS will perform numerous quick and simple queries on small amounts of data, such as acquiring an account balance or finding the status of a customer order, whereas a data warehouse will perform complex queries on large amounts of data. An ODS contains only current operational data while a data warehouse contains both current and historical data.

DataMart: A database, or collection of databases, designed to help managers make strategic decisions about their business. Whereas a data warehouse combines databases across an entire enterprise, data marts are usually smaller and focus on a particular subject or department. Some data marts, called *dependent data marts,* are subsets of larger data warehouses.

Meta Data: Data about data. Metadata describes how and when and by whom a particular set of data was collected, and how the data is formatted. Metadata is essential for understanding information stored in data warehouses and has become increasingly important in XML-based Web applications.

SQL (někdy vyslovováno anglicky *es-kjů-el*, někdy též *síkvl*) je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov **Structured Query Language** (strukturovaný dotazovací jazyk).

DWH: Abbreviated *DW*, a collection of data designed to support management decision making. Data warehouses contain a wide variety of data that present a coherent picture of business conditions at a single point in time.
Development of a data warehouse includes development of systems to extract data from operating systems plus installation of a warehouse database system that provides managers flexible access to the data.
The term data warehousing generally refers to the combination of many different databases across an entire enterprise. Contrast with *data mart*.

BI: Most companies collect a large amount of data from their business operations. To keep track of that information, a business and would need to use a wide range of software programs , such as Excel, Access and different database applications for various departments throughout their organization. Using multiple software programs makes it difficult to retrieve information in a timely manner and to perform analysis of the data.
The term Business Intelligence (BI) represents the tools and systems that play a key role in the strategic planning process of the corporation. These systems allow a company to gather, store, access and analyze corporate data to aid in decision-making. Generally these systems will illustrate business intelligence in the areas of customer profiling, customer support, market research, market segmentation, product profitability, statistical analysis, and inventory and distribution analysis to name a few.

A **Database Management System** (**DBMS**) is a set of computer programs that controls the creation, maintenance, and the use of a database. Details on http://en.wikipedia.org/wiki/Database_management_system

Slovník pojmů

OLAP: Short for ***O**nline **A**nalytical **P**rocessing,* a category of software tools that provides analysis of data stored in a database. OLAP tools enable users to analyze different dimensions of multidimensional data. For example, it provides time series and trend analysis views. OLAP often is used in data mining.
The chief component of OLAP is the OLAP server, which sits between a client and a database management systems (DBMS). The OLAP server understands how data is organized in the database and has special functions for analyzing the data. There are OLAP servers available for nearly all the major database systems.

ETL: Short for ***e**xtract, **t**ransform, **l**oad*, three database functions that are combined into one tool to pull data out of one database and place it into another database.

Extract -- the process of reading data from a database.

Transform -- the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database. Transformation occurs by using rules or lookup tables or by combining the data with other data.

Load -- the process of writing the data into the target database.
ETL is used to migrate data from one database to another, to form data marts and data warehouses and also to convert databases from one format or type to another.

OLTP: Short for ***O**n-**L**ine **T**ransaction **P**rocessing*. Same as *transaction processing*.

Transaction processing: A type of computer processing in which the computer responds immediately to user requests. Each request is considered to be a *transaction.* Automatic teller machines for banks are an example of transaction processing.
The opposite of transaction processing is batch processing, in which a batch of requests is stored and then executed all at one time. Transaction processing requires interaction with a user, whereas batch processing can take place without a user being present.

EAI: Acronym for ***e**nterprise **a**pplication **i**ntegration*. EAI is the unrestricted sharing of data and business processes throughout the networked applications or data sources in an organization. Early software programs in areas such as inventory control, human resources, sales automation and database management were designed to run independently, with no interaction between the systems. They were custom built in the technology of the day for a specific need being addressed and were often proprietary systems. As enterprises grow and recognize the need for their information and applications to have the ability to be transferred across and shared between systems, companies are investing in EAI in order to streamline processes and keep all the elements of the enterprise interconnected.

ERP: Short for ***e**nterprise **r**esource **p**lanning,* a business management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. As the ERP methodology has become more popular, software applications have emerged to help business managers implement ERP in business activities such as inventory control, order tracking, customer service, finance and human resources.

Datový sklad (Data Warehouse)

- Definice (W.H. Inmon 1996):

Datový sklad je

- subjektově orientovaný
- integrovaný
- časově proměnný
- stálý

soubor dat, který slouží pro podporu rozhodování.

Datový sklad

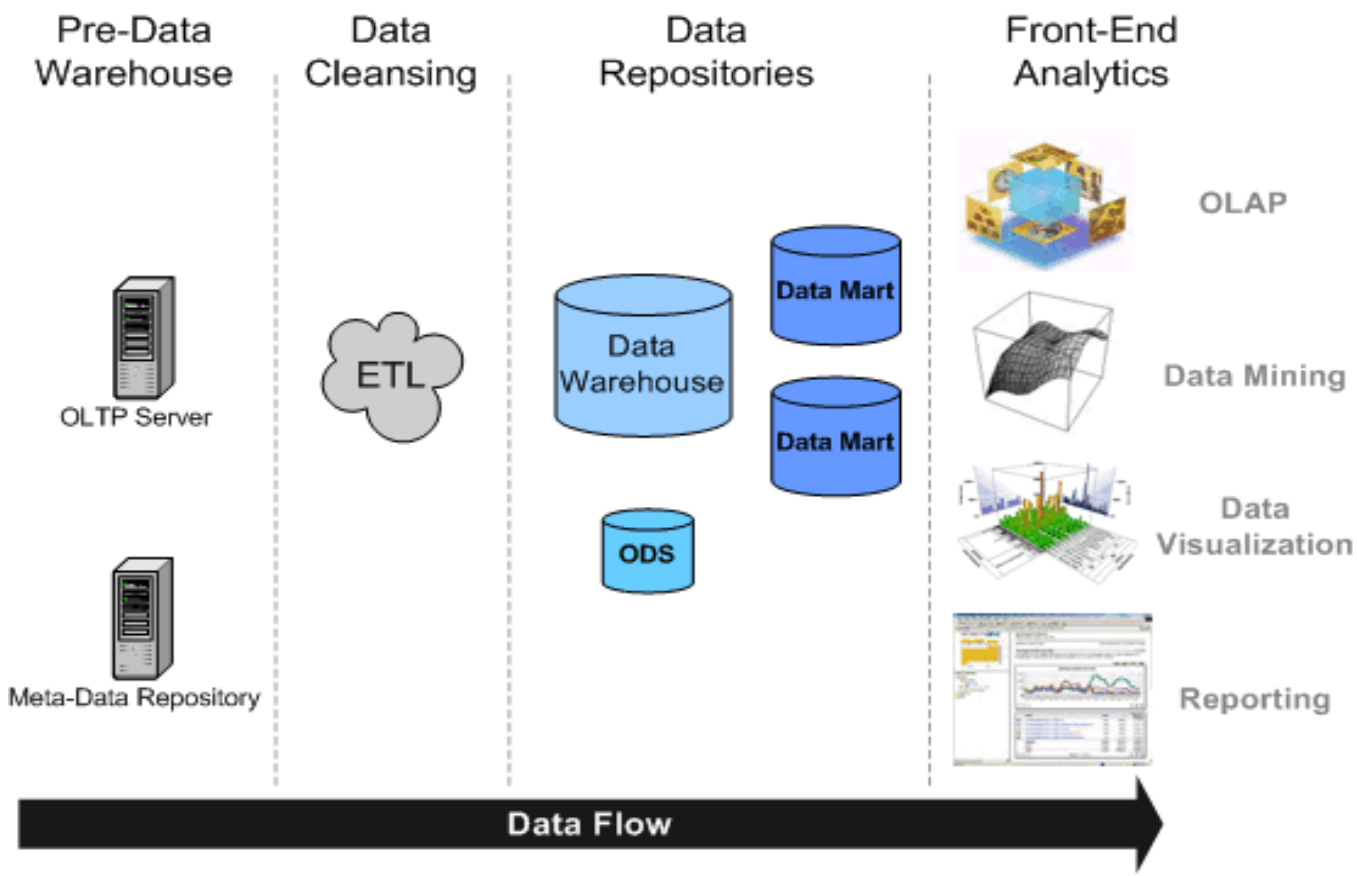
- prvotní koncepce datována počátkem 80.let
- vznik z potřeby jednoduchého přístupu ke strukturovanému úložišti kvalitních dat
- pomáhá získat odpovědi pro lepší rozhodování
- umožňuje použití dat pro dotazování, reportování a analýzu

Struktura datového skladu

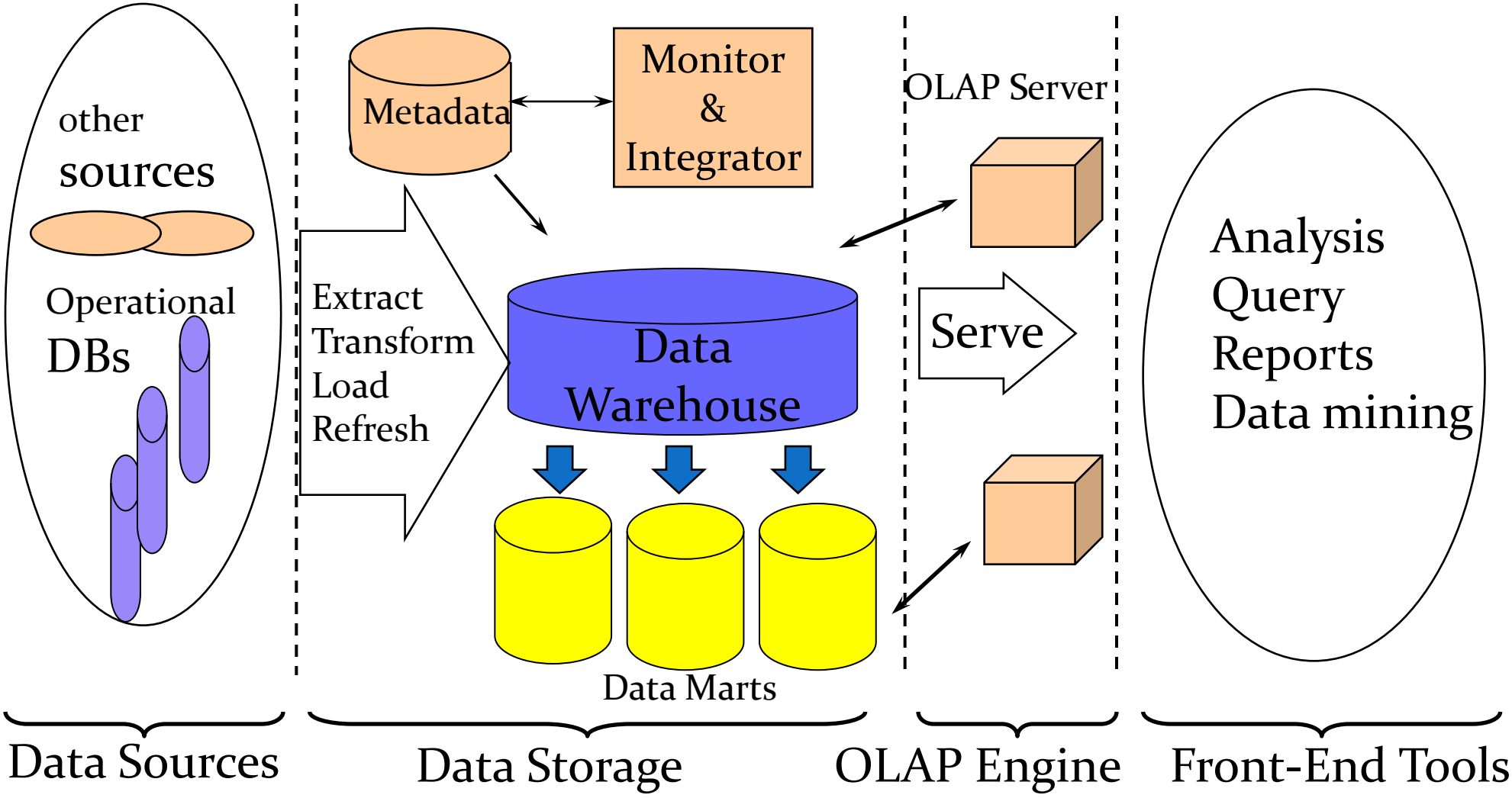
- třívrstvá architektura:
  - datový sklad
  - aplikační vrstva
  - prezentační vrstva

- fyzicky centralizovaný nebo distribuovaný

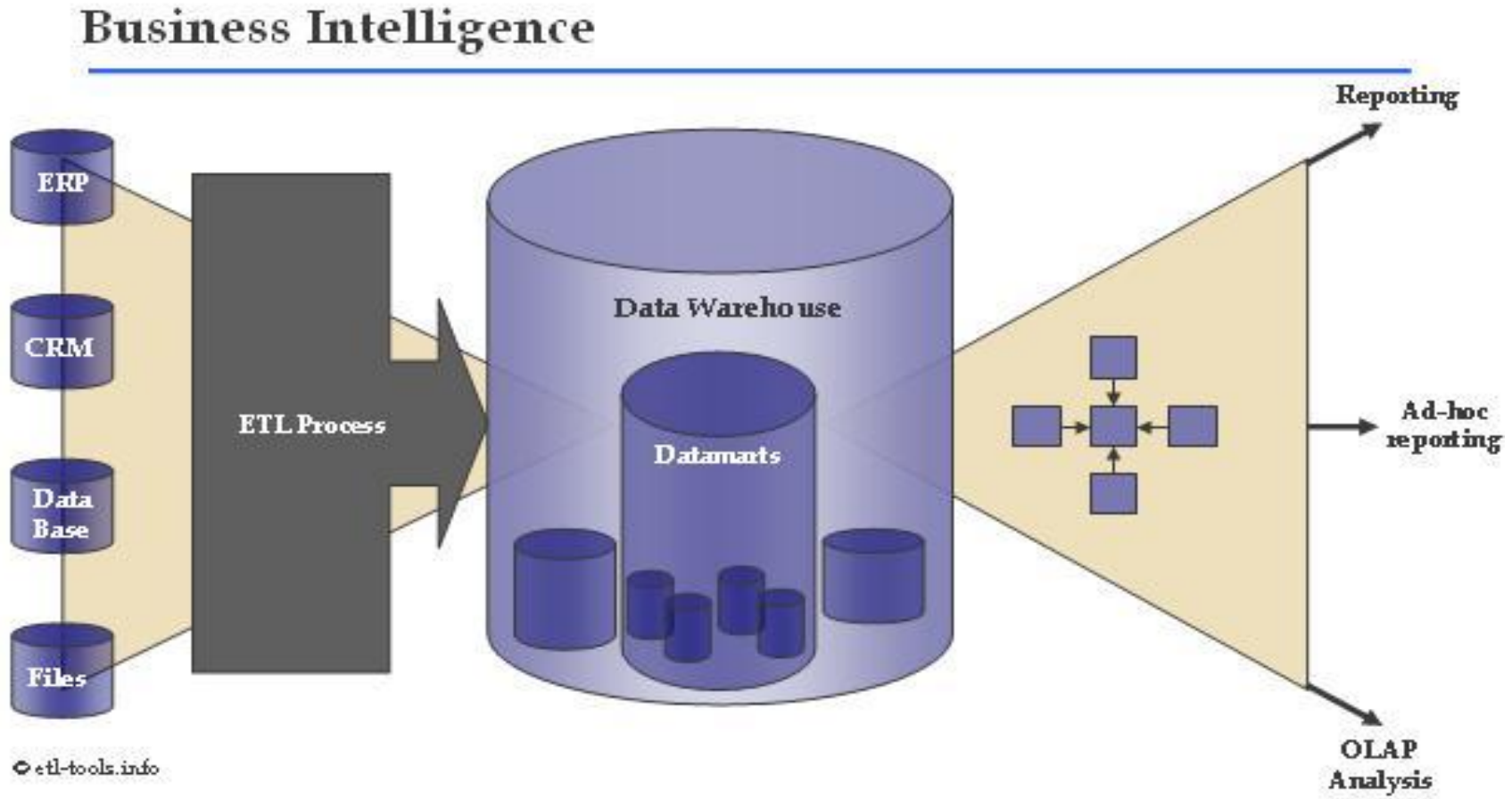
Datový sklad



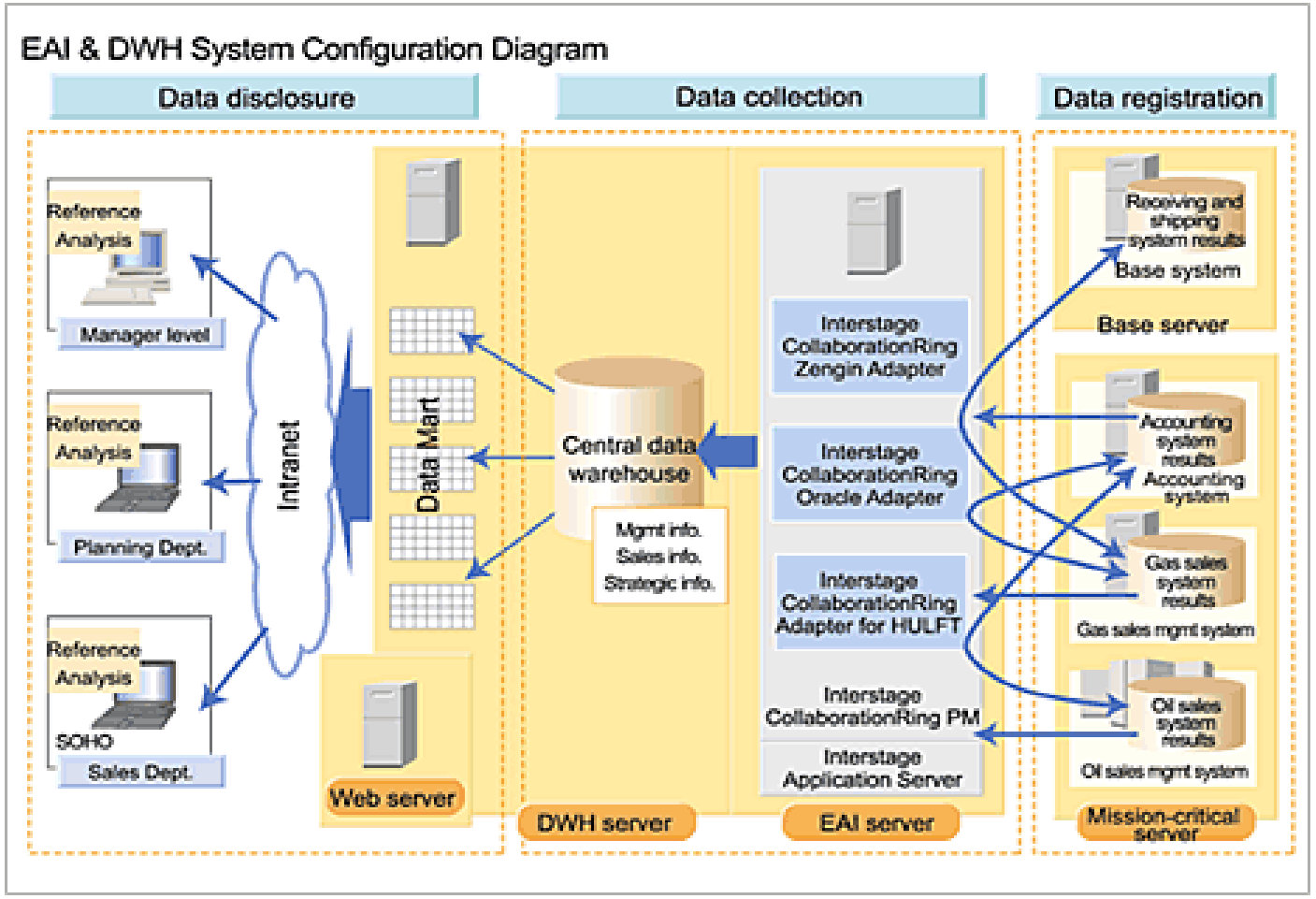
Datový sklad



Datový sklad



Datový sklad

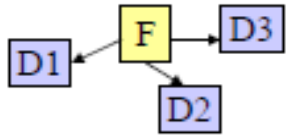


SOHO: Zkratka pro ***s**mall **o**ffice/**h**ome **o**ffice* – malé nebo domácí kancelářské prostředí a business kultura, která je s ním spojena.

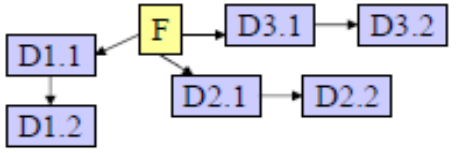
Datové Modely

- Star (hvězda)
- Snowflake (vločka)
- Starflake
- Constellation (souhvězdí)

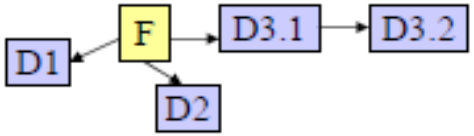
• Star Schema



• Snowflake Schema

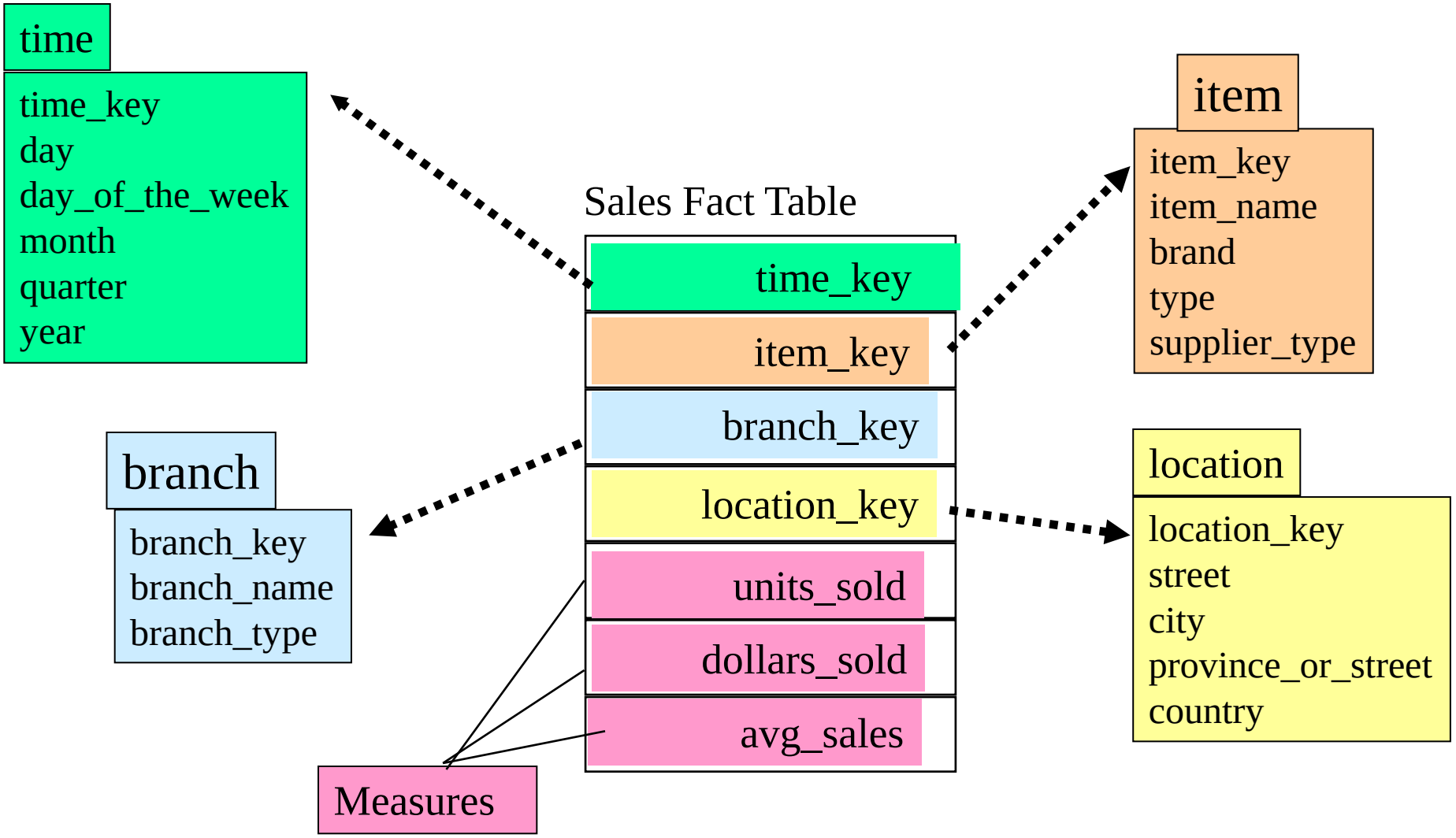


• Starflake Schema

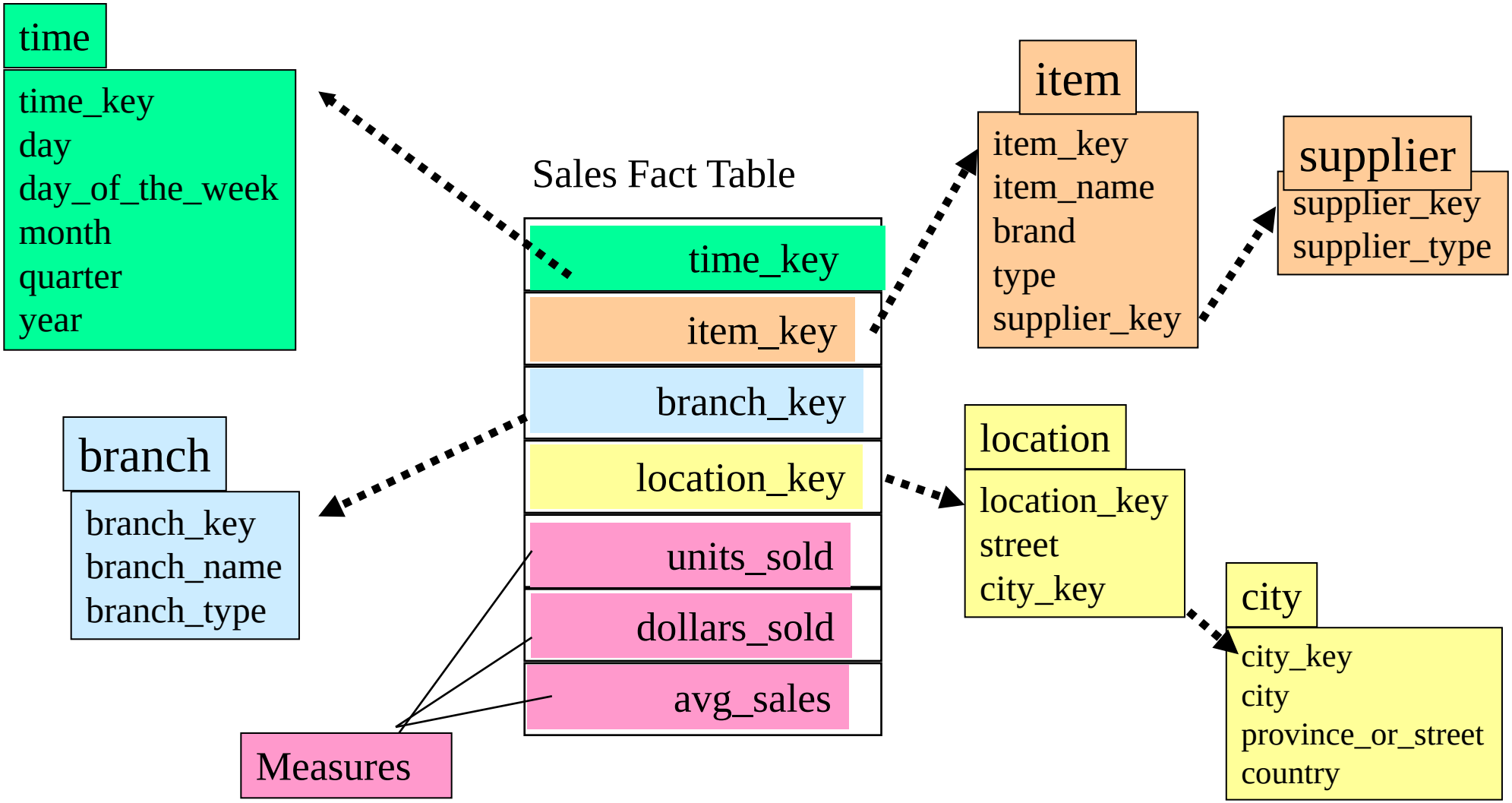


• Constellation Schema

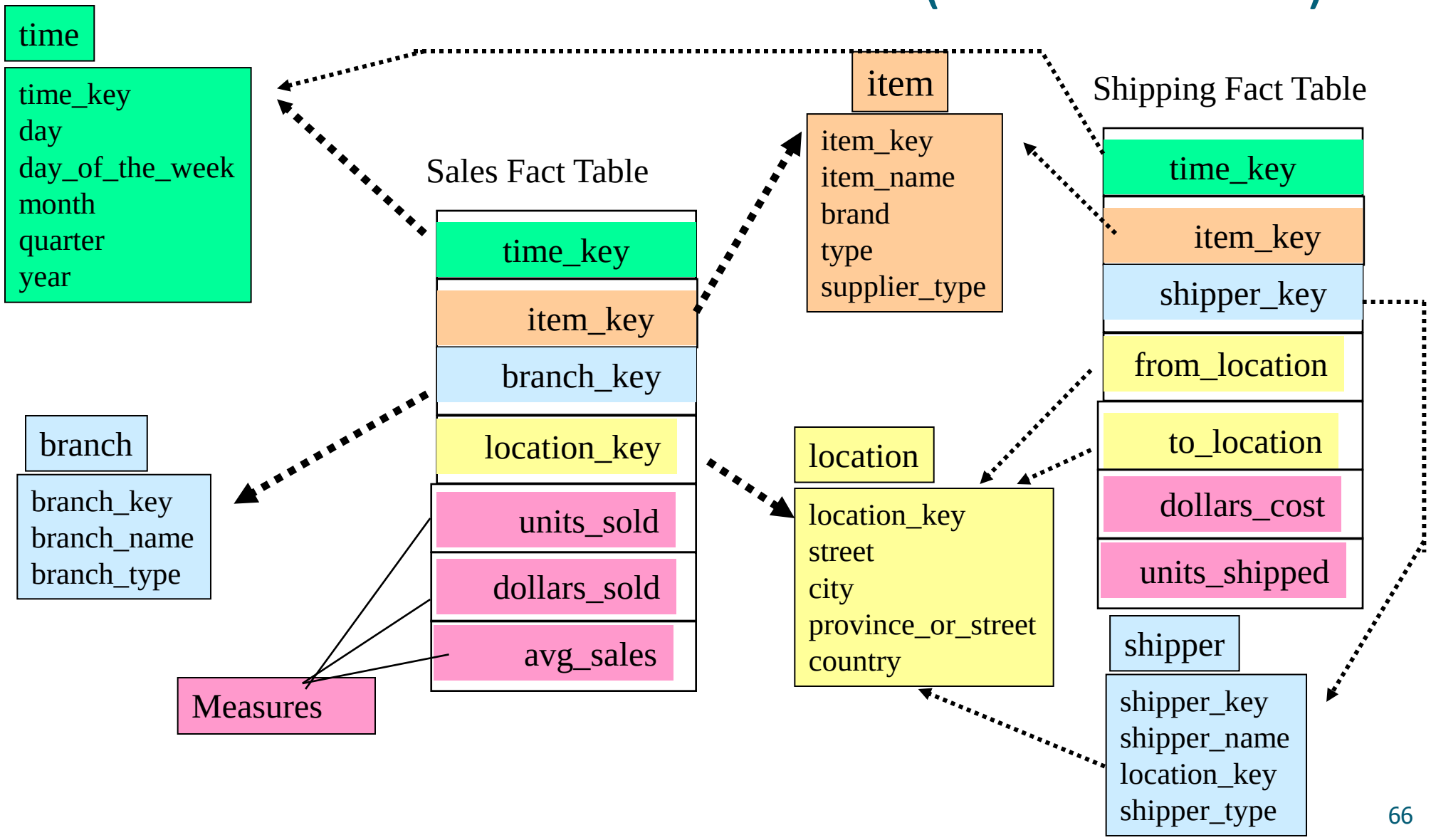
Příklad schématu hvězda (star)



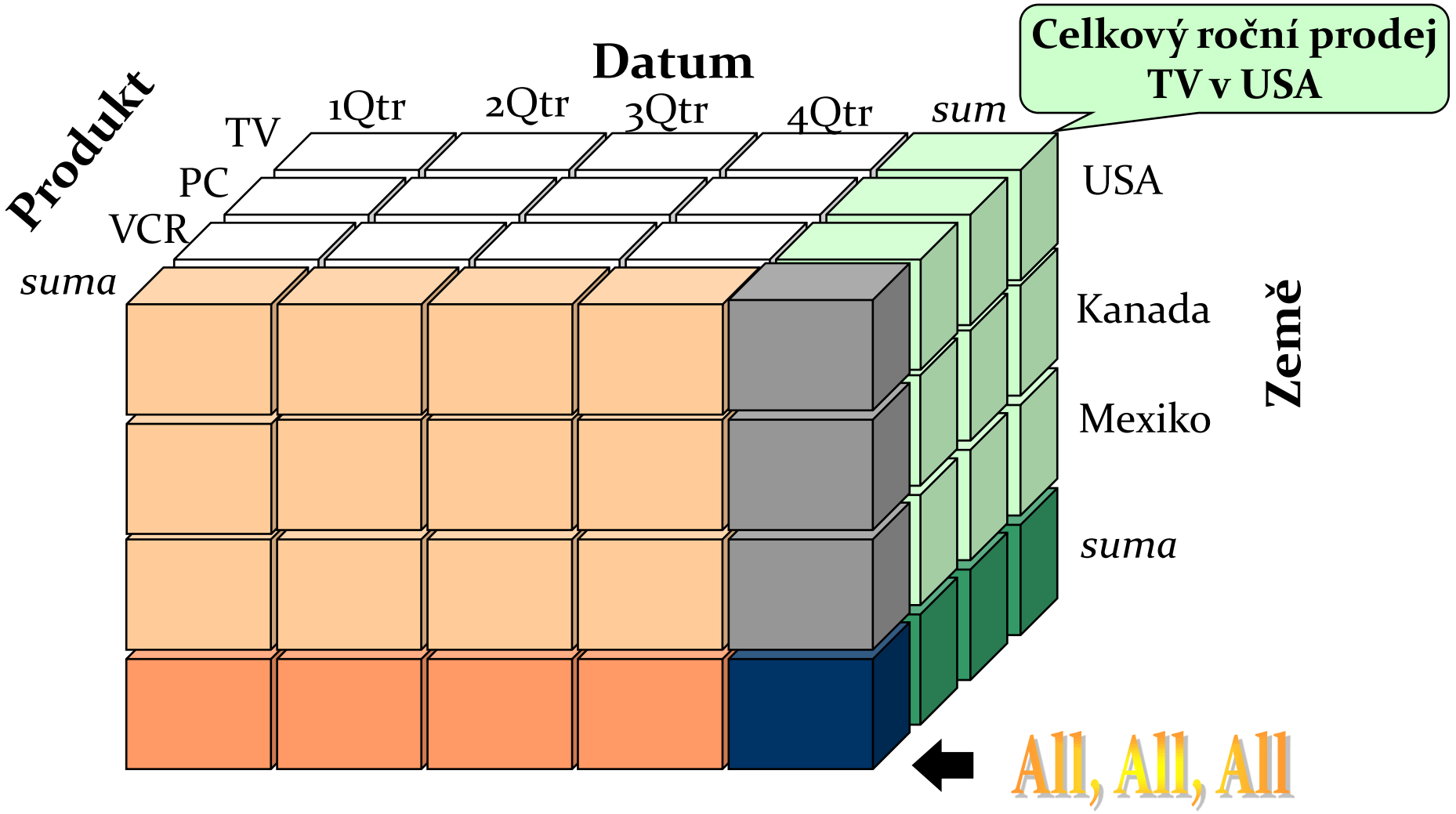
Příklad schématu vločka (Snowflake)



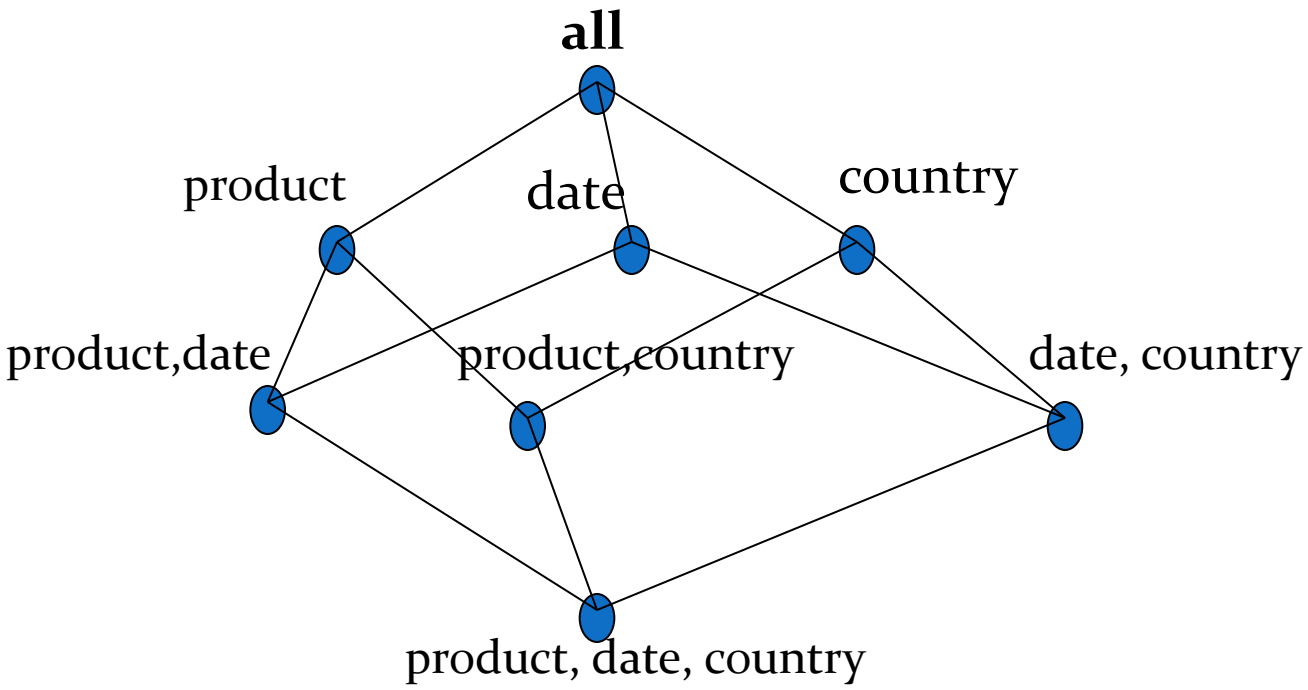
Příklad schématu souhvězdí (Constellation)



Příklad datové kostky



Datové „kvádry“ odpovídající datové kostce



0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

Typické OLAP Operace

- Roll up (drill-up): sumarizace dat
  - *Postoupení v hierarchii o úroveň výše nebo redukce dimenze (např. z kostky na čtverec).*
- Drill down (roll down): opak roll-up –zajímá nás větší detail
  - *Z vyšší úrovně sumarizace na nižší úroveň nebo zavedení nových datových dimenzí.*
- Slice and dice (krájet a kostkovat):
  - *Výběr datového podprostoru.*
- Ostatní operace:
  - *drill across: zahrnutí více datových tabulek (kostek)*
  - *drill through: přes základní úroveň datové kostky zpět k podkladovým relačním tabulkám (pomocí SQL)*

Architektura OLAP Serverů

- Relační OLAP (Relational OLAP -ROLAP)
  - Využívá relační nebo rozšířenou relační DBMS pro ukládání a správu dat datového skladu a OLAPovou střední vrstvu pro podporu chybějících částí.
  - Zahrnuje optimalizační možnosti DBMS, implementaci agregační navigační logiky a doplňkové nástroje a služby.
- Vícedimenzionální OLAP (Multidimensional OLAP - MOLAP)
  - Technologie založená na vícedimenzionálních datových polích (vč. technik pro řídké matice).
  - Rychlé indexování předem spočtených sumarizovaných dat.
- Hybridní OLAP (Hybrid OLAP - HOLAP)
  - Uživatelsky flexibilní, tj. low level: relační, high-level: pole.
- Specializované SQL servery
  - specializovaná podpora pro SQL dotazy nad star/snowflake schématy.

ROLAP

- Data uložená v relační databázi – nejsou duplikována, ovšem není k nim možný přístup bez připojení k zdrojové databázi.

- dotazy OLAP se převádějí do klasických dotazů SQL – může být nevýhodou (limitované možnosti SQL, pomalejší odezva).

- Vhodný jen pro omezené množství dat.

MOLAP

- „tradiční“ OLAP.
- Data uložena v multidimenzionálních kostkách mimo relační databázi. Jsou tudíž duplikována a je možný přístup i bez spojení s původním zdrojem dat.
- Hlavní výhodou je rychlá odezva na dotazy. Vše je předpočítáno a uloženo při tvorbě kostek.

HOLAP

- ponechává původní data v relačních tabulkách, agregace ukládá v multidimenzionálním formátu
- poskytuje propojení mezi rozsáhlými objemy dat v relačních tabulkách
- výhoda rychlejšího výkonu multidimenzionálně uložených agregací

Budování datového skladu

- metoda „velkého třesku“:
  - analýza požadavků podniku
  - vytvoření podnikového datového skladu
  - vytvoření datových tržišť

- přírůstková (evoluční) metoda

Plnění datového skladu

- počáteční plnění + pravidelná aktualizace

- plnění pomocí datových pump
- postupy ETL:
  - extrakce
  - transformace
  - loading

Co je SQL?

The SQL procedure uses Structured Query Language to perform the following tasks:

- retrieve and manipulate SAS data sets
- create and delete SAS data sets
- generate reports
- add or modify values in a SAS data set
- add, modify, or drop columns in a SAS data set

Úvod do SQL

- General form of an SQL procedure query to generate output:

```
PROC SQL;
    SELECT variables
        FROM SAS-data-set;
```

Úvod do SQL

- Create a listing report of product activity.
  - Step 1: Invoke the SQL procedure.

```
proc sql;
```

  - Step 2: Identify the variables to display on the report.

```
proc sql;
    select CustomerID, CustomerFirstName,
           CustomerLastName
```

Úvod do SQL

- Step 3: Identify the input data set.

```
proc sql;
   select CustomerID, CustomerFirstName,
          CustomerLastName
      from univ.mastercustomers;
```

- Step 4: End the procedure with a QUIT statement.

```
proc sql;
   select CustomerID, CustomerFirstName,
          CustomerLastName
      from univ.mastercustomers;
quit;
```

Úvod do SQL

- SQL joins have the following characteristics:
  - They do not require sorted data.
  - They can be performed on up to 32 data sets at one time.
  - They allow complex matching criteria using the WHERE clause.

Úvod do SQL

- General form of an SQL procedure join to generate output:

```
PROC SQL;
      SELECT variables
            FROM SAS-data-set1 AS alias1,
                        SAS-data-set2 AS alias2
             WHERE alias1.variable=alias2.variable;
```

Úvod do SQL

- Create a listing report by joining data sets **univ.mastercustomers** and **univ.customerorders** by **CustomerID**.

  - Step 1: Invoke the SQL procedure and list the variables to display.

```
proc sql;
   select CustomerID, CustomerFirstName,
          CustomerLastName, OrderID,
          UnitPrice, Quantity
```

Úvod do SQL

- Step 2: Identify the data sets to join and provide a table alias for each.

- Because **CustomerID** exists in both data sets, identify which **CustomerID** to use.

```
proc sql;
   select m.CustomerID, CustomerFirstName,
          CustomerLastName, OrderID,
          UnitPrice, Quantity
      from univ.mastercustomers as m,
         univ.customerorders as c
```

Úvod do SQL

- Step 3: State the condition on which observations are matched and terminate the query.

```
proc sql;
   select m.CustomerID, CustomerFirstName,
          CustomerLastName, OrderID,
          UnitPrice, Quantity
      from univ.mastercustomers as m,
           univ.customerorders as c
      where m.CustomerID=c.CustomerID;
quit;
```

Úvod do SQL

Create a new variable named **TotSale** by multiplying **Quantity** by **UnitPrice**. Name the new variable **TotSale**.

```
proc sql;
   select m.CustomerID, CustomerFirstName,
          CustomerLastName, OrderID,
          UnitPrice, Quantity,
          Quantity * UnitPrice as TotSale
      from univ.mastercustomers as m,
           univ.customerorders as c
      where m.CustomerID=c.CustomerID;
quit;
```

Úvod do SQL

- General form of a PROC SQL query to create a SAS data set:

```
PROC SQL;
    CREATE TABLE SAS-data-set AS
        SELECT ...
        other SQL clauses;
```

Úvod do SQL

- Join the tables **univ.mastercustomers** and **univ.customerorders** to create a new data set.

```
proc sql;
   create table work.ordertotals as
      select m.CustomerID,
             CustomerFirstName,
             CustomerLastName, OrderID,
             UnitPrice, Quantity,
             Quantity*UnitPrice as TotSale
         from univ.mastercustomers as m,
              univ.customerorders as c
         where m.CustomerID=c.CustomerID;
quit;
```

Úvod do SQL

- General form of an SQL procedure query using labels and formats:

```
PROC SQL;
      SELECT variable LABEL='column-header'
                        FORMAT=format.
        FROM SAS-data-set ;
```

Úvod do SQL

- Enhance the previous report.

```
proc sql;
   select m.CustomerID,
          CustomerFirstName format=$10.,
          CustomerLastName format=$15.,
          OrderID,
            UnitPrice format=dollar7.2,
          Quantity,
          Quantity * UnitPrice as TotSale
               format=dollar8.2
               label='Total Sale Amount'
      from univ.mastercustomers as m,
             univ.customerorders as c
      where m.CustomerID=c.CustomerID;
quit;
```

Úvod do SQL

- Partial Output

Customer ID	Customer First Name	Customer Last Name	OrderID	Unit Price	Quantity	Sale Amount
062096	Craig	Knapmeyer	1240062267	$36.00	3	$108.00
062096	Craig	Knapmeyer	1240832690	$27.00	4	$108.00
062284	Robert	Britt	1238409388	$15.00	1	$15.00
062284	Robert	Britt	1238409388	$33.00	1	$33.00
064810	Randall	Goodman	1238248877	$175.00	4	$700.00
064810	Randall	Goodman	1238248877	$283.00	1	$283.00
064810	Randall	Goodman	1238273875	$220.00	1	$220.00
064810	Randall	Goodman	1238768955	$52.00	1	$52.00
064810	Randall	Goodman	1238842450	$24.00	1	$24.00
064810	Randall	Goodman	1239353817	$59.00	2	$118.00
064810	Randall	Goodman	1239489696	$11.00	2	$22.00
064810	Randall	Goodman	1239608721	$22.00	3	$66.00
064810	Randall	Goodman	1239608721	$46.00	3	$138.00
064810	Randall	Goodman	1240590287	$21.00	2	$42.00

Úvod do SQL

- General form of an SQL procedure query to generate summary output:

```
PROC SQL;
    SELECT group-variable,
           SUM(analysis-variable)
      FROM SAS-data-set
      GROUP BY group-variable;
```

- If a summary function is used in the SELECT clause with only one argument, then an overall statistic is calculated down the column.

Úvod do SQL

- Step 1: Identify the variables to display, the input data sets, and the matching criteria.

```
proc sql;
   select m.CustomerID,
          CustomerFirstName format=$10.,
          CustomerLastName format=$15.,
          sum(Quantity) label= 'Total Quantity',
          sum(Quantity*UnitPrice) as TotSale
                 format=dollar12.2
                 label='Total Sale Amount'
      from univ.mastercustomers as m,
           univ.customerorders as c
      where m.CustomerID=c.CustomerID;
```

Úvod do SQL

- Step 2: Identify the grouping variable(s).

```
proc sql;
   select m.CustomerID,
          CustomerFirstName format=$10.,
          CustomerLastName format=$15.,
          sum(Quantity) label='Total Quantity',
          sum(Quantity*UnitPrice) as TotSale
               format=dollar12.2
               label='Total Amount Purchased'
     from univ.mastercustomers as m,
          univ.customerorders as c
     where m.CustomerID=c.CustomerID
     group by m.CustomerID, CustomerFirstName,
              CustomerLastName;
quit;
```

Úvod do SQL

- General form of an SQL procedure query to generate ordered output:

```
PROC SQL;
      SELECT group-variable,
                        SUM(analysis-variable)
           FROM SAS-data-set
            GROUP BY group-variable
            ORDER BY variable1 <, variable2> ;
```

- The default is ascending order.

Úvod do SQL

- Order the report by total sale.

```
proc sql;
   select m.CustomerID,
          CustomerFirstName format=$10.,
          CustomerLastName format=$15.,
          sum(Quantity) label='Total Quantity',
          sum(Quantity*UnitPrice) as TotSale
               format=dollar12.2
               label='Total Amount Purchased'
     from univ.mastercustomers as m,
          univ.customerorders as c
     where m.CustomerID=c.CustomerID
     group by m.CustomerID, CustomerFirstName,
              CustomerLastName
     order by TotSale;
quit;
```

Úvod do SQL

- Order the report by total sale – **v sestupném pořadí**

```
proc sql;
   select m.CustomerID,
          CustomerFirstName format=$10.,
          CustomerLastName format=$15.,
          sum(Quantity) label='Total Quantity',
          sum(Quantity*UnitPrice) as TotSale
               format=dollar12.2
               label='Total Amount Purchased'
     from univ.mastercustomers as m,
          univ.customerorders as c
     where m.CustomerID=c.CustomerID
     group by m.CustomerID, CustomerFirstName,
              CustomerLastName
     order by TotSale desc;
quit;
```

Inner JOIN

- The INNER JOIN keywords can be used to join tables. The ON clause replaces the WHERE clause for specifying columns to join. PROC SQL provides these keywords primarily for compatibility with the other joins (OUTER, RIGHT, and LEFT JOIN). Using INNER JOIN with an ON clause provides the same functionality as listing tables in the FROM clause and specifying join columns with a WHERE clause.

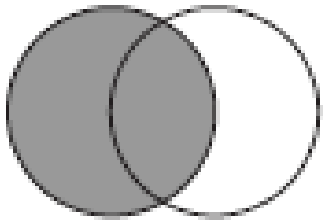
```
proc sql outobs=6;
title 'Oil Production/Reserves
of Countries';
select p.country, barrelsperday
'Production', barrels
'Reserves'
from sql.oilprod p,
sql.oilrsrvs r
where p.country = r.country
order by barrelsperday desc;
```

=

```
proc sql ;
select p.country,
barrelsperday
'Production', barrels
'Reserves'
from sql.oilprod p inner
join sql.oilrsrvs r
on p.country = r.country
order by barrelsperday
desc;
```

Left JOIN

- *Outer joins are inner joins that are augmented with rows from one table that do not* match any row from the other table in the join. The resulting output includes rows that match and rows that do not match from the join's source tables. Nonmatching rows have null values in the columns from the unmatched table. Use the ON clause instead of the WHERE clause to specify the column or columns on which you are joining the tables. However, you can continue to use the WHERE clause to subset the query result.

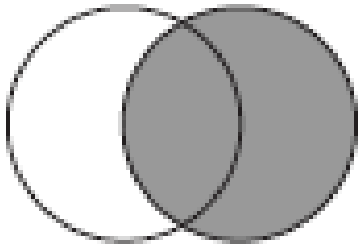


- A left outer join lists matching rows and rows from the left-hand table (the first table listed in the FROM clause) that do not match any row in the right-hand table. A left join is specified with the keywords LEFT JOIN and ON.

```
proc sql;
select Capital format=$20., Name 'Country'
format=$20.,
Latitude, Longitude
from sql.countries a left join sql.worldcitycoords b
on a.Capital = b.City and
a.Name = b.Country;
```

Right JOIN

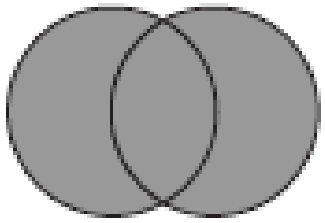
- A right join, specified with the keywords RIGHT JOIN and ON, is the opposite of a left join: nonmatching rows from the right-hand table (the second table listed in the FROM clause) are included with all matching rows in the output.



```
proc sql outobs=10;
select City format=$20., Country
'Country' format=$20., Population
from sql.countries right join
sql.worldcitycoords
on Capital = City and
Name = Country
order by City;
```

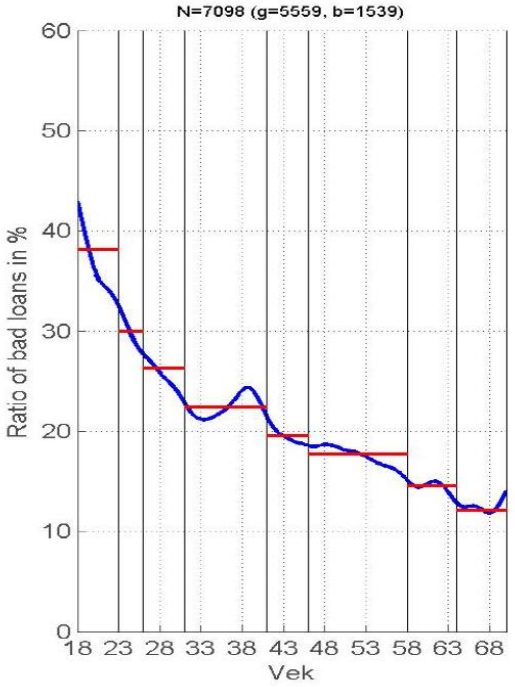
Inner/Full Outer/Left/Right JOIN

• A full outer join, specified with the keywords FULL JOIN and ON, selects all matching and nonmatching rows.



```
proc sql outobs=10;
select City '#City#(WORLDCITYCOORDS)'
format=$20.,
Capital '#Capital#(COUNTRIES)'
format=$20.,
Population, Latitude, Longitude
from sql.countries full join
sql.worldcitycoords
on Capital = City and
Name = Country;
```

3. Příprava dat –čištění, kategorizace, agregace, transformace dat, úvod do SAS Data Step



Čištění dat: Praktické zkušenosti

- Pokud vaše nová data obsahují více než 30 čísel, tak je v nich skoro jistě nějaká chyba.
- Čištění a příprava dat zabírá obvykle 80 – 90 % analytikova času.
- Pokud budete VELMI pečliví v této fázi, ušetříte si daleko víc času a nervů později – jinak stavíte dům na písku.

GIGO

- Garbage in, Garbage out (smetí dovnitř, smetí ven)
  - sebelepší model (proces) nevyrobí ze smetí nic jiného než opět smetí.

Co způsobí nekvalitní data

- Správa nekvalitních/nadbytečných dat
- Nedoručené zásilky (marketing, fakturace)
- Nesprávné výsledky zpracování (reporting, analýzy, data mining)
- Špatné fungování systému (nekompatibilita)
- Ztráta image, nespokojení klienti

Co způsobí nekvalitní data

- Při mailingové kampani jedné britské **maloobchodní společnosti** se ukázalo, že jedna pětina oslovených už zemřela. Přesto (nebo pro to?) byli obesláni s pozdravným oslovením „**Drahý pane Zesnulý**“. [1)]
- Jistá **pojišťovna** zjistila, že většina jejich zákazníků má **zaměstnání „Astronaut**“ – další pátrání ukázalo, že „Astronaut“ je první volba v seznamu v jejich CRM systému. [1)]
- 44 000-98 000 Američanů ročně umírá na základě **odvratitelné medicínské chyby** jako přepsání při psaní receptu, špatně popsaný výsledek krevní zkoušky, nečitelná informace v pacientských záznamech atd. Je to **osmá nejčastější příčina úmrtí** v USA [2)]
- 7.5.1999 bombardovaly **ozbrojené síly USA** čínské velvyslanectví v Jugoslávii. Vyšetřování zjistilo: CIA používá zastaralý mapový materiál; ještě k tomu pracovník předložil v důsledku chyby v datech **špatnou adresu** – „Doslovně nakreslil X na nesprávné místo“ [3)]

1) Peel, M: Letters to the dead and other data dereliction. © 2007 Financial Times Deutschland. http://www.ftd.de, vydání z 2.10.2007
2) Oash, J. (1999): IT Can Reduce Medical Errors. Obsaženo v: Wang, Pierce, Madnick: Information Quality, 2005
3) BBC: Americas chinese embassy warning ignored. © 1999 BBC. http://news/bbc.co.uk/1/hi/world/americas/37775.stm, vydání z 2.10.2007

Datová kvalita

- Profiling, DQ Assessment – zjištění v jakém stavu jsou data
- Deduplikace, clustering, unifikace, konsolidace
- Prevence:
  - Data Governance – soustavná péče o data
  - Master Data Management – řešení pro správu klíčových dat

Čištění dat: Ověření souboru

- Ověření souboru s daty / zdrojů dat
  - Jsou to správná data (čas vzniku, výzkum…)?
  - Jsou kompletní, bez duplicit, umím je číst…
- Zkoumání případů
  - Mají identifikátory?
    - Jsou tyto ID správné?
  - Neopakují se (duplicity)?
    - Existují i „skoro“ duplicity – dva podobné, ale ne přesně totožné záznamy o tomtéž subjektu.
  - Nejsou vynechány?

Čištění dat: Ověření proměnných

- Zkoumání metadat o proměnných
  - Jsou tam všechny proměnné a správně značené?
  - Je jasné, co znamenají (kódovníky, definice…)? Dokumentace OK?
    - Pozor na mezinárodní studie, produkty konsorcií agentur a opakované vlny výzkumů. Jemné nuance metody mohou způsobit hrubý nesoulad !
  - Neopakuje se některá proměnná vícekrát?

Čištění dat: Průzkum proměnných

- Nabývá přípustných hodnot (x out of range)?
- „Divné“ kódy („xxx“, „9999“…)
- Duplicitní kódy pro stejnou věc („Ž“, „ž“, „žena“, „zena“…)
- Kódování češtiny/ruštiny/…

Čištění dat: Průzkum proměnných

- Překlepy apod.
  - Editovací distance (Levenshteinova (Владимир Иосифович Левенштейн), …) pomohou odhalit překlep
  - Editovací distance = počet elementárních editovacích kroků potřebných pro změnu jednoho řetězce na druhý. Viz http://www.merriampark.com/ld.htm k Levenshteinově distanci
    - Je zde aplet, který ji umí počítat
  - Shlukování řetězců podle ED

Čištění dat: Průzkum proměnných

- Slučování podobných kategorií (prodavač – prodejce – prodavačka);
- Málo četné kategorie (národnost brazilská…) – je třeba sloučit/přiřadit k nějaké(kým) více četné(ným) kategorii(ím) na základě nějakého vhodného kriteria.
- Je distribuce přiměřená našemu očekávání (interval hodnot, rozptyl, šikmost, špičatost, modální hodnoty…)? Není např. příliš „ořezaná“ či naopak „roztažená“?
  - Někdy se obtížně poznává: Např. věk v části dat může být kódován jako poslední dvojčíslí roku narození, a v jiné části dat jako *2007 – rok narození*.

Čištění dat: Průzkum proměnných

- Shluky (clumping), typicky kolem zaokrouhlených hodnot
  - Příjem –lidé rádi zaokrouhlují směrem nahoru.
  - Nebo třeba kolem hranic věkových kvót, vzniklé tím, jak tazatelé „upravují“ věky respondentů, aby se vešli do kvót.
- Chybějící hodnoty (příčiny vzniku, zastoupení,…)!!!
- Pozor na kódy časů (amer. x evrop. konvence), regionů apod.!

Čištění dat: Vazby mezi daty

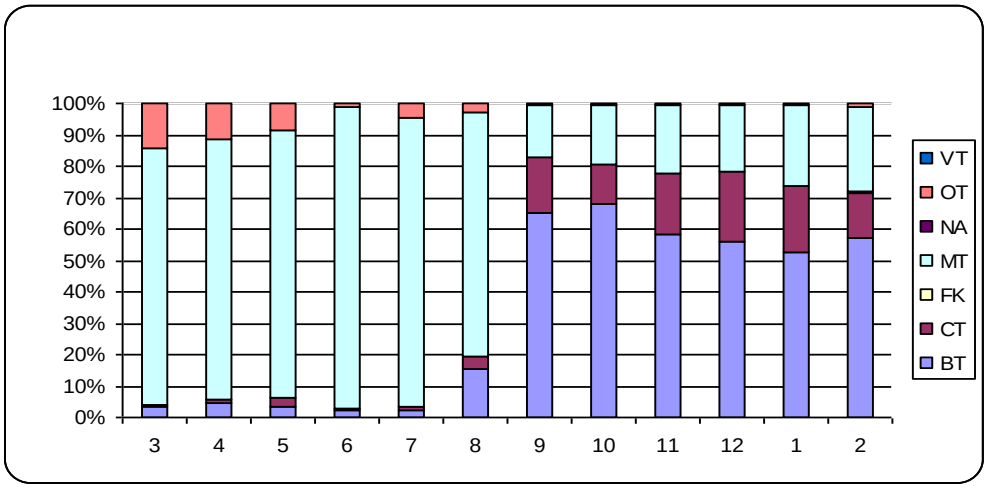
- Více proměnných
  - Kontingenční tabulky, box ploty s kategoriemi, bodové grafy a jejich matice, korelační koeficienty
  - Logické vazby (např. 10tiletý nemůže být ženatý, 30tiletý nemůže pracovat 20let,…)
    - Hledání pomocí programu/kódu – podmínky vyjádříme pomocí prostředků matematické logiky a necháme počítač, aby vyhledal případy, kde nejsou splněny.

Čištění dat: Vazby mezi daty

- Více proměnných
  - Extrémní hodnoty vícerozměrného rozdělení
    - Bodový graf
    - Mahalanobisova vzdálenost od těžiště: $[(\boldsymbol{x}-\boldsymbol{t})^T \boldsymbol{S}^{-1} (\boldsymbol{x}-\boldsymbol{t})]^{-1/2}$, kde $\boldsymbol{t}$ je vektor těžiště, $\boldsymbol{x}$ zkoumaný bod a $\boldsymbol{S}$ kovarianční matice
      - např. P. Filzmoser (2004) A multivariate outlier detection method, http://www.statistik.tuwien.ac.at/public/filz/papers/minsko4.pdf
  - Další vlastnosti; např. existují očekávané korelace?

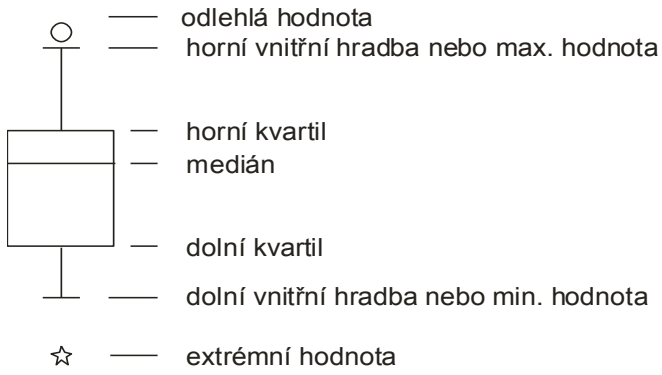
Čištění dat: Vazby mezi daty

- korektní vkládání dat do DB
  - text. pole s názvem zboží vs. rolovací seznam s typem zboží



  - pořadí hodnot v rolovacím seznamu – problém první (defaultní) hodnoty

Čištění dat: Odlehlé hodnoty



- kvartilová odchylka: $q = x_{0.75} - x_{0.25}$
- vnitřní hradby: $x_{0.25} - 1.5q$, $x_{0.75} + 1.5q$
- vnější hradby: $x_{0,25} - 3q$, $x_{0,75} + 3q$

- **Odlehlá hodnota** leží mezi vnějšími a vnitřními hradbami, tj. v intervalu $(x_{0,75} + 1{,}5q, x_{0,75} + 3q)$ či v intervalu $(x_{0,25} - 3q, x_{0,25} - 1{,}5q)$.
- **Extrémní hodnota** leží za vnějšími hradbami, tj. v intervalu $(x_{0,75} + 3q, \infty)$
- či v intervalu $(-\infty, x_{0,25} - 3q)$.

Čištění dat: Opravy chyb

- Zpět k pramenům!
- Vyřazení podezřelých případů:
  - Záměrné podvody, např. nespolehliví tazatelé (shluková analýza!).
  - Neověřitelná data.
- Vyřazení podezřelých hodnot.
- Rekódování na správné hodnoty (imputace hodnot):
  - imputace – průměrem, mediánem, max./min. hodnotou, pomocí modelu.

Transformace dat

- Binarizace (dummy proměnné)
  - Dummy proměnné představují techniku využívající dichotomické proměnné (kódované 0 nebo 1) pro vyjádření jednotlivých hodnot nominálních proměnných.
  - Název „dummy“ poukazuje na fakt, že přítomnost znaku označeného kódem 1 reprezentuje faktor, nebo soubor faktorů, který není měřitelný žádným lepším způsobem v rámci dané analýzy.

Dummy proměnné

- Dummy proměnná přiřazuje hodnotu 1 danému pozorování vybrané proměnné a hodnotu 0 ve zbývajících případech.
- Pro pohlaví (2 kategorie), např. přiřadí 1 pro ženu a 0 pro muže. V tomto případě je postačující vytvoření právě jedné dummy proměnné.
- Pro rasu (4 kategorie), je třeba vytvořit více dummy proměnných.

  P1=1, pokud rasa=„běloch“ a 0 jinak.

  P2=1, pokud rasa=„černoch“ a 0 jinak.

  P3=1, pokud rasa=„asiat“ a 0 jinak.

  P4=1, pokud rasa=„ostatní“ a 0 jinak.
- Důležité: Všechny 4 proměnné nejsou zahrnuty do regrese (způsobilo by to perfektní multikolinearitu, P4=1-P3-P2-P1).
- Počet dummy proměnných=počet kategorií -1.
- Vynechaná proměnná je „referenční“ proměnnou.
- Konstanta obsahuje informaci o této referenční proměnné.
- Koeficienty zahrnutých proměnných jsou brány ve vztahu ke konstantě.

Transformace dat

- Kategorizace spojitých proměnných
  - decily
- Agregace
- Segmentace

Categorization of predictors

- Every variable should be categorized (divided to reasonable number of categories

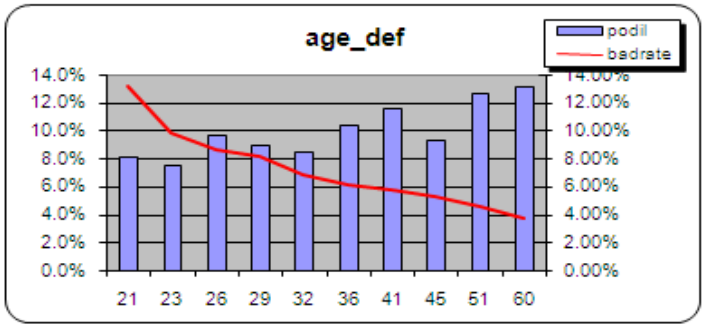
  - Best separation (default rates within categories are different as much as possible)

  - Time stability (ordering in categories by default rate is the same in different periods of development sample)

age_def		Gini:	0.2212

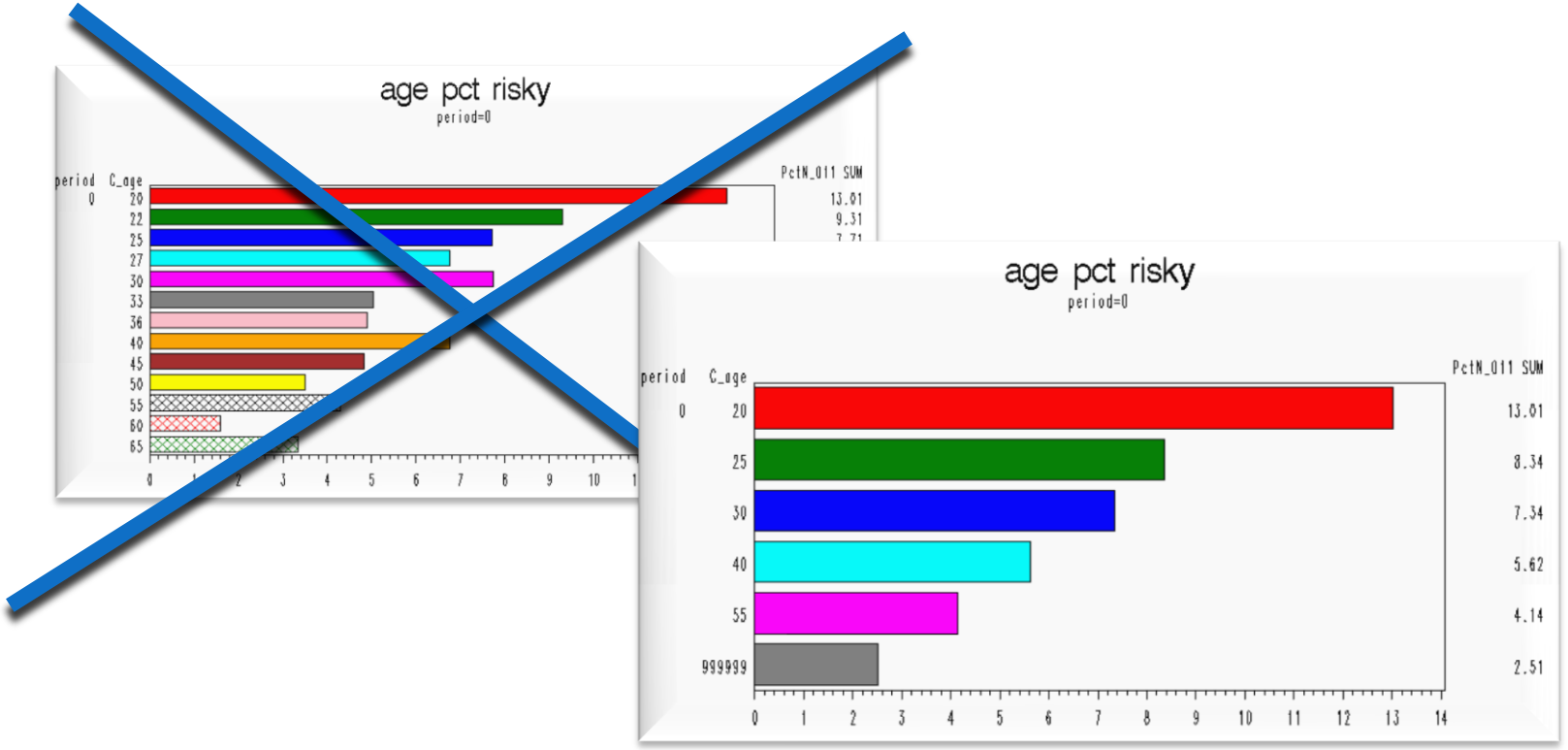
Info.Value:	0.1558

	pocet	podil	badrate
21	35 059	8.2%	13.11%
23	32 401	7.5%	9.81%
26	41 807	9.7%	8.61%
29	38 510	9.0%	8.07%
32	36 271	8.4%	6.79%
36	44 648	10.4%	6.11%
41	50 015	11.6%	5.74%
45	40 099	9.3%	5.21%
51	54 526	12.7%	4.52%
60	56 551	13.2%	3.71%
Total	429 887	100.0%	6.79%



Categorization of predictors

- We want to find out real statistical dependencies, not random differences in default.



Transformace dat - WOE

- **Good** celkový počet dobrých klientů ve vzorku
- **Bad** celkový počet špatných klientů ve vzorku

- $\mathbf{good_i^s}$, $\mathbf{bad_i^s}$ počet dobrých, resp. špatných klientů v *i*-té kategorii příslušné *s-té* proměnné.
- celková šance

$$odds\_all = \frac{good}{bad}$$

- šance i-té kategorie s-té proměnné

$$odds_i^{\ s} = \frac{good_i^{\ s}}{bad_i^{\ s}}$$

- poměr šancí (OR)

$$odds\_ratio_i^{\ s} = \frac{odds_i^{\ s}}{odds\_all}$$

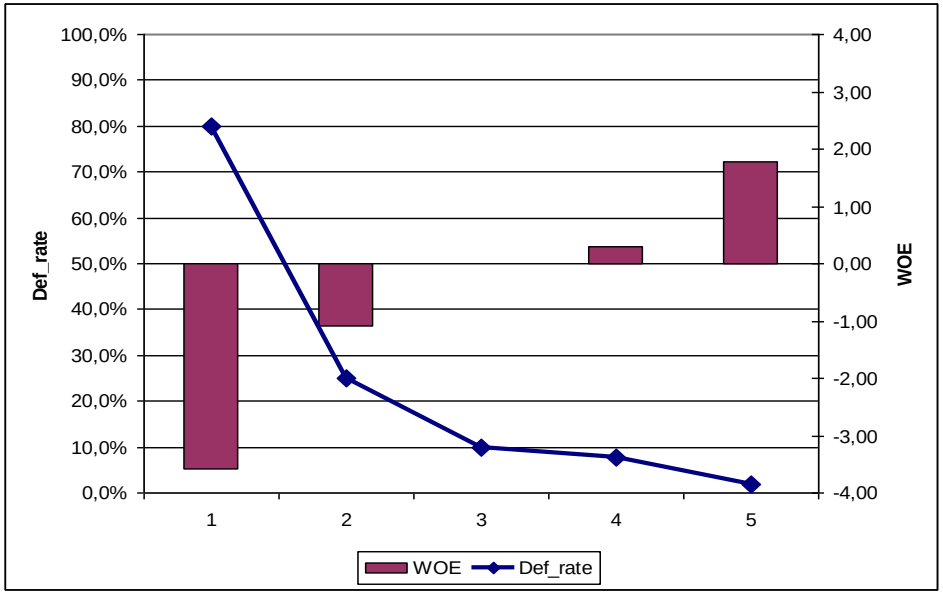
- WOE (weights of evidence)

$$WOE_i^{\ s} = \ln\left(odds\_ratio_i^{\ s}\right) = \ln\left(\frac{\frac{good_i^{\ s}}{bad_i^{\ s}}}{\frac{good}{bad}}\right) = \ln\left(\frac{\frac{good_i^{\ s}}{good}}{\frac{bad_i^{\ s}}{bad}}\right)$$

Transformace dat -WOE

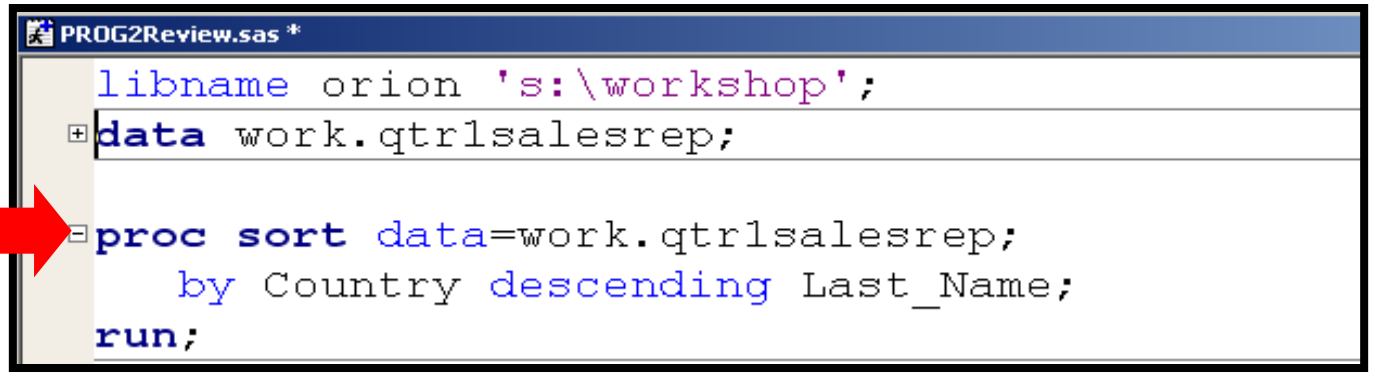
cat.	# bad clients	#good clients	Def_rate	odds	OR	% bad [1]	% good [2]	[3] = [2] / [1]	WOE = ln[3]
1	4	1	80,0%	0,25	0,03	40,0%	1,1%	0,03	-3,58
2	2	6	25,0%	3,00	0,33	20,0%	6,7%	0,33	-1,10
3	2	18	10,0%	9,00	1,00	20,0%	20,0%	1,00	0,00
4	1	12	7,7%	12,00	1,33	10,0%	13,3%	1,33	0,29
5	1	53	1,9%	53,00	5,89	10,0%	58,9%	5,89	1,77
All	10	90	10,0%	9,00					
ALL	100								

80% = 4/ (4+1)
0,25 = 1/4
0,03 = 0,25 / 9
40% = 4 / 10
1,1% = 1 / 90



The SORT Procedure

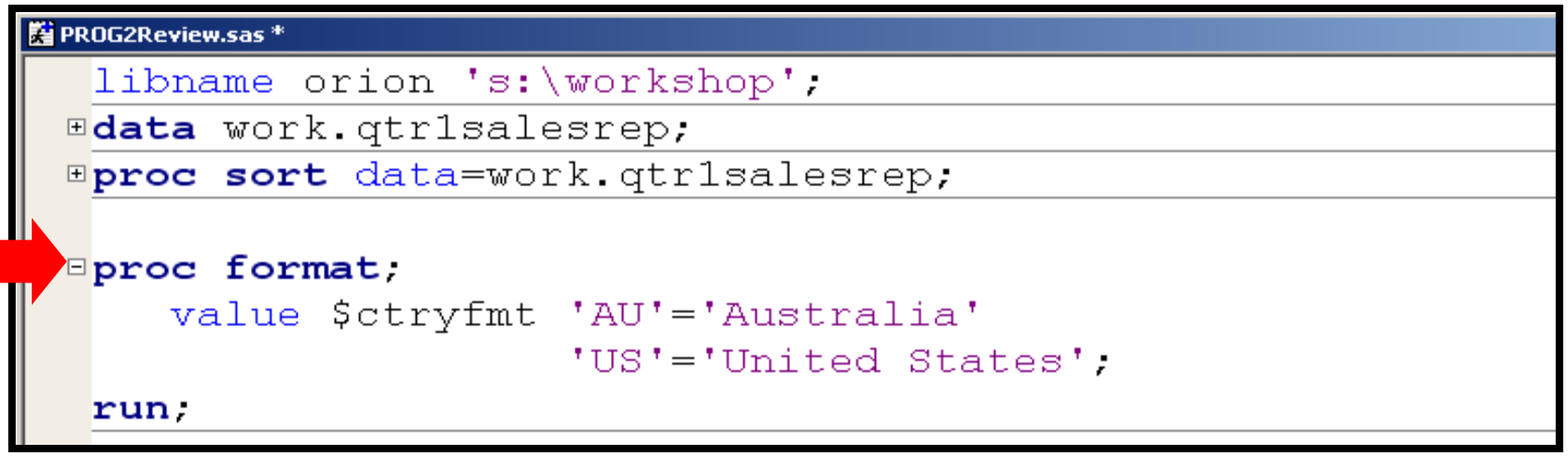
- The SORT procedure rearranges the observations in **work.qtr1salesrep** and places them in order by descending **Last_Name** within **Country**.



- The OUT= option in the SORT procedure can be used to create an output data set, instead of overwriting the input data set.

The FORMAT Procedure

- The FORMAT procedure creates user-defined formats and informats, and stores them in the SAS catalog **work.formats** by default.



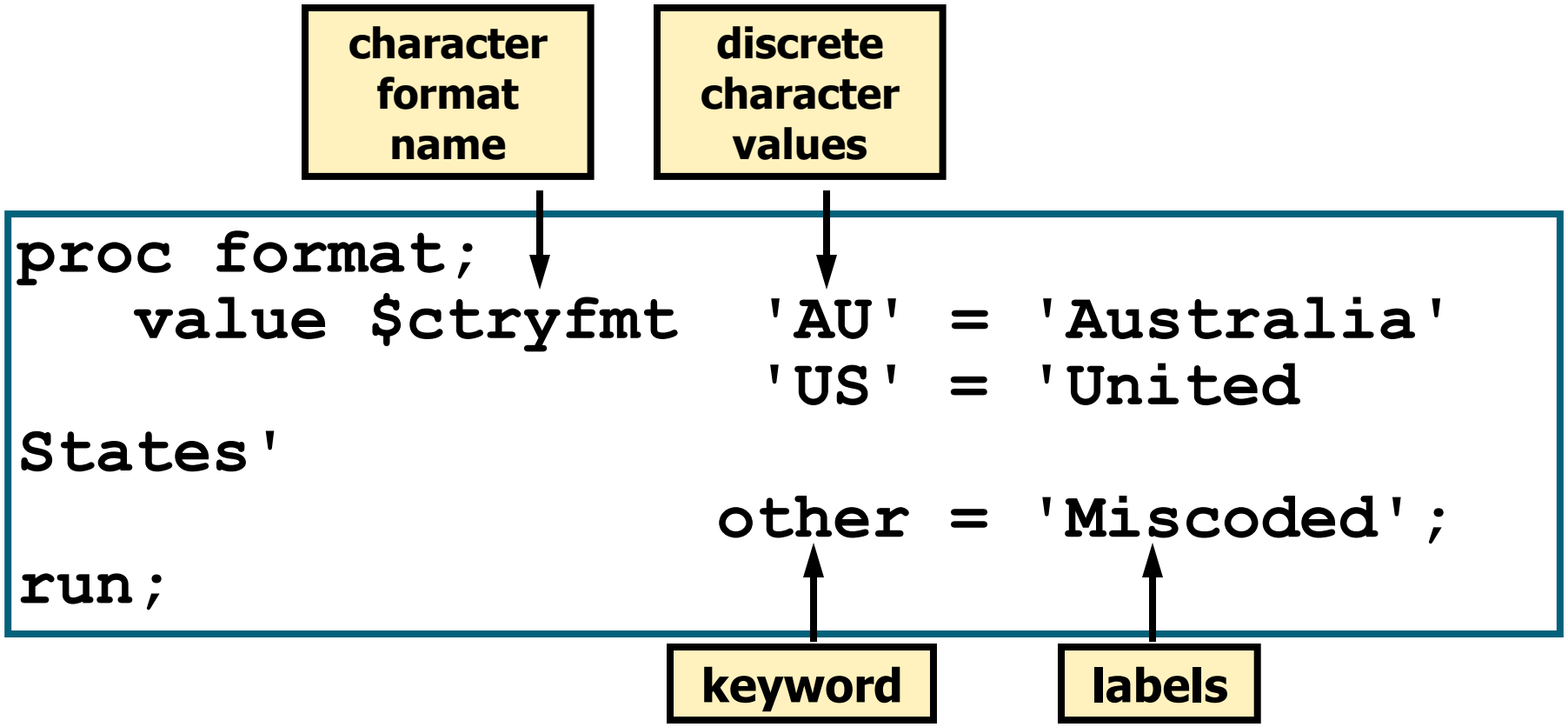
- Více na: http://www2.sas.com/proceedings/sugi27/p056-27.pdf

The FORMAT Procedure

- *Range(s)* can be
  - single values
  - ranges of values
  - lists of values.

- *Labels*
  - can be up to 32,767 characters in length
  - are typically enclosed in quotation marks, although it is not required.

Character User-Defined Format



- The OTHER keyword matches all values that do not match any other value or range.

Character User-Defined Format

Part 1

```
proc format;
   value $ctryfmt   'AU' = 'Australia'
                    'US' = 'United States'
                  other = 'Miscoded';
run;

proc print data=orion.sales label;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   label Employee_ID='Sales ID'
         Job_Title='Job Title'
         Salary='Annual Salary'
         Birth_Date='Date of Birth'
         Hire_Date='Date of Hire';
```

Part 2

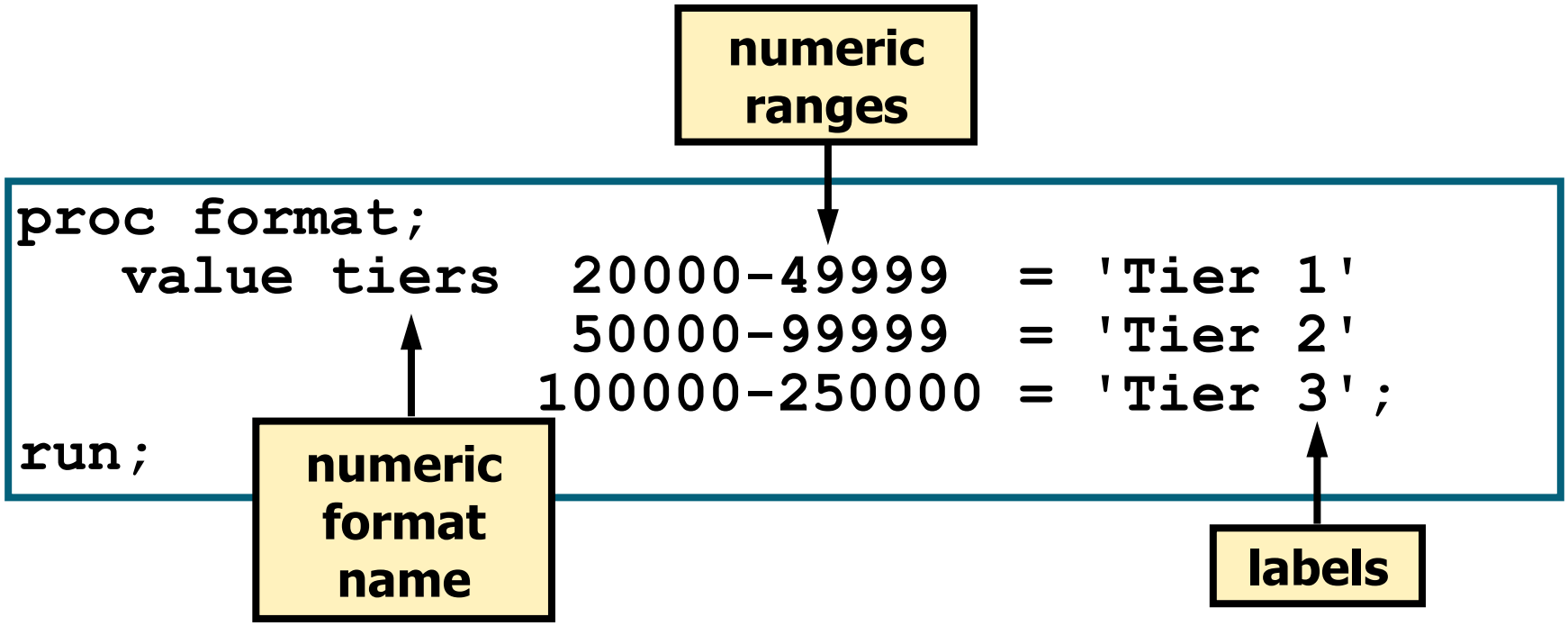
```
   format Salary dollar10.0
          Birth_Date Hire_Date monyy7.
          Country $ctryfmt.;
run;
```

Character User-Defined Format

- Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
60	120178	Sales Rep. II	$26,165	Australia	NOV1954	APR1974
61	120179	Sales Rep. III	$28,510	Australia	MAR1974	JAN2004
62	120180	Sales Rep. II	$26,970	Australia	JUN1954	DEC1978
63	120198	Sales Rep. III	$28,025	Australia	JAN1988	DEC2006
64	120261	Chief Sales Officer	$243,190	United States	FEB1969	AUG1987
65	121018	Sales Rep. II	$27,560	United States	JAN1944	JAN1974
66	121019	Sales Rep. IV	$31,320	United States	JUN1986	JUN2004
67	121020	Sales Rep. IV	$31,750	United States	FEB1984	MAY2002
68	121021	Sales Rep. IV	$32,985	United States	DEC1974	MAR1994
69	121022	Sales Rep. IV	$32,210	United States	OCT1979	FEB2002
70	121023	Sales Rep. I	$26,010	United States	MAR1964	MAY1989
71	121024	Sales Rep. II	$26,600	United States	SEP1984	MAY2004
72	121025	Sales Rep. II	$28,295	United States	OCT1949	SEP1975

Numeric User-Defined Format

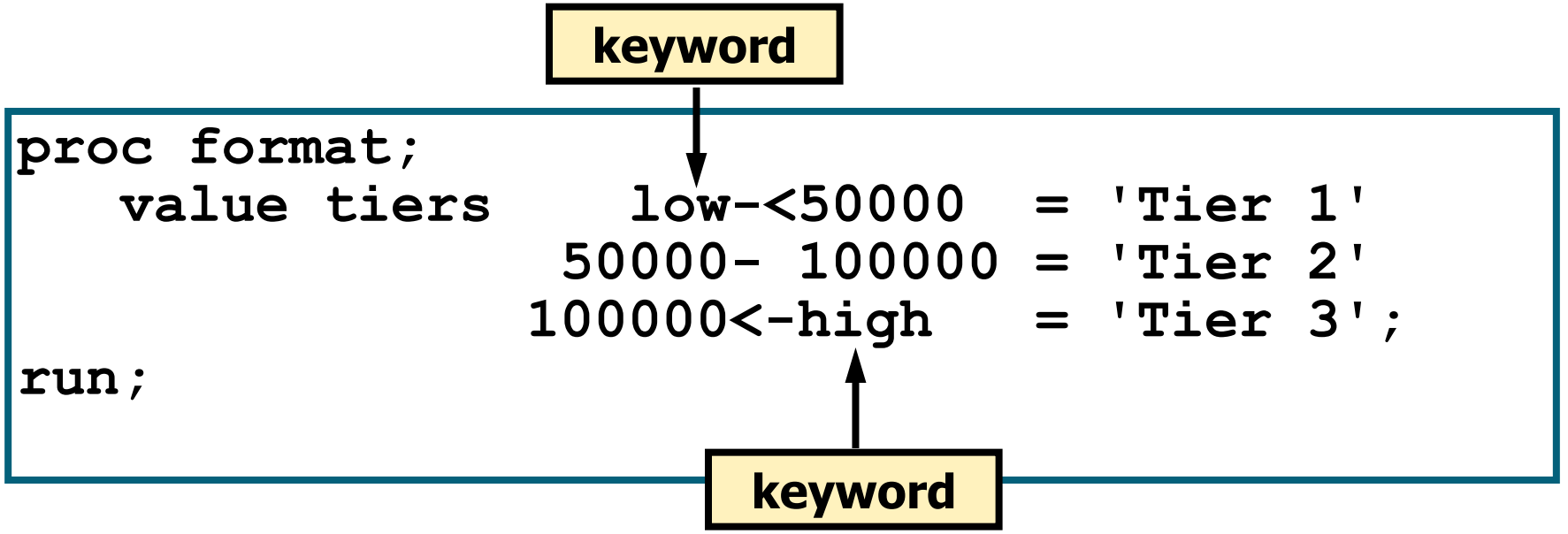


Numeric User-Defined Formats

- The less than (<) symbol excludes values from ranges.
  - Put < after the value if you want to exclude the first value in a range.
  - Put < before the value if you want to exclude the last value in a range.

50000 - 100000	Includes 50000	Includes 100000
50000 - < 100000	Includes 50000	Excludes 100000
50000 < - 100000	Excludes 50000	Includes 100000
50000 < - < 100000	Excludes 50000	Excludes 100000

Numeric User-Defined Format



LOW encompasses the lowest possible value.

HIGH encompasses the highest possible value.

Other User-Defined Format Examples

```
proc format;
   value $grade      'A' = 'Good'
                 'B'-'D' = 'Fair'
                     'F' = 'Poor'
                 'I','U' = 'See Instructor'
                   other = 'Miscoded';
run;
```

```
proc format;
   value mnthfmt     1,2,3 = 'Qtr 1'
                     4,5,6 = 'Qtr 2'
                     7,8,9 = 'Qtr 3'
                  10,11,12 = 'Qtr 4'
                         . = 'missing'
                     other = 'unknown';
run;
```

Multiple User-Defined Formats

- Multiple VALUE statements can be in a single PROC FORMAT step.

```
proc format;
   value $ctryfmt   'AU' = 'Australia'
                    'US' = 'United States'
                  other = 'Miscoded';
   value tiers      low-<50000  = 'Tier 1'
                 50000- 100000 = 'Tier 2'
                100000<-high   = 'Tier 3';
run;
```

The FORMAT Procedure

```
proc format;
value $goods_t
'BT'='A'
'BZ'='D,
''='missing'
'  '='missing'
.='missing'
;
run;
```

```
proc tabulate data=lib1.tab1
missing;
title "D vs. goods_type";
class goods_type D;
table (goods_type all),(D
all)*(n colpctn='c%'
rowpctn='r%');
format goods_type $goods_t.;
run;
```

The FORMAT Procedure

```
proc format;
value good_typ
1=1
2=3
3=10
;
run;
```

```
Data lib1.tab1;
Set lib1.tab1;
   goods_type3=goods_type2;
   format goods_typen3n
good_typ.;
run;
```

The FORMAT Procedure

```
proc format;
invalue good_t2e
'BT'=4
'BZ'=5
'CK'=5
other=-1
;
run;
```

```
data lib1.tab1;
set lib1.tab1;
goods_type1=upcase(goods_type);
goods_type3n=input(goods_type1,goo
d_t2e.);
evid_id=put(evid_id,z10.);
;
run;
```

Replacing Missing Values

The COALESCE function enables you to replace missing values in a column with a new value that you specify. For every row that the query processes, the COALESCE function checks each of its arguments until it finds a nonmissing value, then returns that value. If all of the arguments are missing values, then the COALESCE function returns a missing value. For example, the following query replaces missing values in the LowPoint column in the SQL.CONTINENTS table with the words **Not Available:**

```
proc sql;
title 'Continental Low
Points';
select Name,
coalesce(LowPoint,
'Not Available') as
LowPoint
from sql.continents;
```

Output 2.14 Using the COALESCE Function to Replace Missing Values

Continental Low Points

Name	LowPoint
Africa	Lake Assal
Antarctica	Not Available
Asia	Dead Sea
Australia	Lake Eyre
Central America and Caribbean	Not Available
Europe	Caspian Sea
North America	Death Valley
Oceania	Not Available
South America	Valdes Peninsula

The DATA Step

- The SAS DATA step
  - is the original SAS programming language for data manipulation
  - can be used as a complete **programming language**
  - is generated by SAS Enterprise Guide when data is imported or in support of other tasks.

Advantages of the DATA Step over SQL

DATA Step	SQL
Can read data from many different sources	Can only read from SAS database tables
Can create multiple tables in a single pass of the data	Can only output one table at a time
Has comprehensive conditional processing	Only has the CASE clause
Can deal with repetitive programming using loops and arrays	Does not support loops or arrays

Advantages of SQL over the DATA Step

SQL	DATA Step
Is very flexible when joining multiple tables with non-common key variables	Can require several steps to join multiple tables with different key variables
Can, in some cases, replace multiple SAS steps	Can require several steps
Is the native language of databases	Might need to generate SQL to get to data that is not SAS data

Choose the right tool for the task to be completed.

The DATA Statement

- The *DATA statement* begins a DATA step and provides the name of the SAS data set being created.
- General form of the DATA statement:

```
DATA output-SAS-data-set;
   SET input-SAS-data-set;
<additional SAS statements>
RUN;
```

- The DATA statement can create temporary or permanent data sets.

The SET Statement

- The *SET statement* reads observations from a SAS data set for further processing in the DATA step.
- General form of the SET statement:

```
DATA output-SAS-data-set;
   SET input-SAS-data-set;
<additional SAS statements>
RUN;
```

- By default, the SET statement does the following:
  - names the SAS data set(s) to be read
  - reads all observations and all variables from the input data set
  - can read temporary or permanent data sets

Business Scenario: Reading a SAS Data Set

This program does the following:

- reads all the rows and all the columns from the **sales** data set in the **orion** library
- writes all the rows and all the columns to a data set named **comp** in the Work library

```
data work.comp;
   set orion.sales;
run;
```

Partial Listing of **comp**

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
120102	Tom	Zhou	M	108255	Sales Manager	AU	3510	10744
120103	Wilson	Dawes	M	87975	Sales Manager	AU	-3996	5114
120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	-5630	5114
120122	Christina	Ngan	F	27475	Sales Rep. II	AU	-1984	6756
120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	1732	9405
120124	Lucian	Daymond	M	26480	Sales Rep. I	AU	-233	6999

Selecting Variables

- You can control the variables **written out** to SAS data sets using the following:
  - the DROP statement to specify the variables that you want **excluded**
  - the KEEP statement to specify the variables that you want **included**
- General form of DROP and KEEP statements:

DROP *variable1 variable2* …**;**

KEEP *variable1 variable2* …**;**

Business Scenario: Selecting Variables

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
run;
```

This program can do these tasks:

- read all the rows and columns from **orion.sales**
- write all the rows and the three columns not excluded via the DROP statement to a data set called **comp** in the Work library

Partial Listing of **comp**

Employee_ID	First_Name	Last_Name
120102	Tom	Zhou
120103	Wilson	Dawes
120121	Irenie	Elvish
120122	Christina	Ngan
120123	Kimiko	Hotstone
120124	Lucian	Daymond

Selecting Rows

Partial Listing of **austemp**

	Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country
1	120102	Tom	Zhou	M	108255	Sales Manager	AU
2	120103	Wilson	Dawes	M	87975	Sales Manager	AU
3	120125	Fong	Hofmeister	M	32040	Sales Rep. IV	AU
4	120128	Monica	Kletschkus	F	30890	Sales Rep. IV	AU
5	120129	Alvin	Roebuck	M	30070	Sales Rep. III	AU
6	120135	Alexei	Platts	M	32490	Sales Rep. IV	AU
7	120144	Viney	Barbis	M	30265	Sales Rep. III	AU
8	120154	Caterina	Hayawardhana	F	30490	Sales Rep. III	AU
9	120158	Daniel	Pilgrim	M	36605	Sales Rep. III	AU
10	120159	Lynelle	Phoumirath	F	30765	Sales Rep. IV	AU
11	120161	Rosette	Martines	F	30785	Sales Rep. III	AU
12	120166	Fadi	Nowd	M	30660	Sales Rep. IV	AU

Orion wants to subset the data to only include Australian employees with a salary greater than $30,000.

Selecting Rows with the WHERE Statement

You can control which rows are read from a SAS data set by using the WHERE statement.

General form of the WHERE statement:

WHERE *expression*;

- Only one WHERE statement can be included in a DATA step.
- The expressions that can be used are the same as expressions built in the Filter Data tab using either the Edit Filter window or the Advanced Expression Editor.

Comparison Operators -examples

```
where Gender = 'M';
```

```
where Gender eq ' ';
```

```
where Salary ne .;
```

```
where Salary >= 50000;
```

```
where Country in ('AU','US');
```

```
where Country in ('AU' 'US');
```

Values must be separated by commas or blanks.

Arithmetic Operators - examples

```
where Salary / 12 < 6000;
```

```
where (Salary / 12 ) * 1.10 >= 7500;
```

```
where Salary + Bonus <= 10000;
```

Logical Operators - examples

```
where Gender ne 'M' and Salary >=50000;
```

```
where Gender ne 'M' or Salary >= 50000;
```

```
where Country = 'AU' or Country = 'US';
```

```
where Country not in ('AU' 'US');
```

Multiple Choice Poll – Correct Answer

- Which WHERE statement correctly subsets for numeric months May, June, or July and character names with a missing value?

a. `where Months in (5 - 7) and Names = . ;`

b. `where Months in (5 , 6 , 7) and Names = ' ';`

c. `where Months in ('5' , '6' , '7') and Names = '.';`

Creating New Variables

- Assignment statements are used in the DATA step to update existing variables or create new variables.
- An assignment statement does the following:
  - evaluates an expression
  - assigns the resulting value to a variable

General form of an assignment statement:

```
variable=expression;
```

```
DATA output-SAS-data-set;
     SET input-SAS-data-set;
     variable = expression;
RUN;
```

SAS Expressions

- An *expression* contains *operands* and *operators* that form a set of instructions that produce a value.

Operands are

- variable names
- constants.

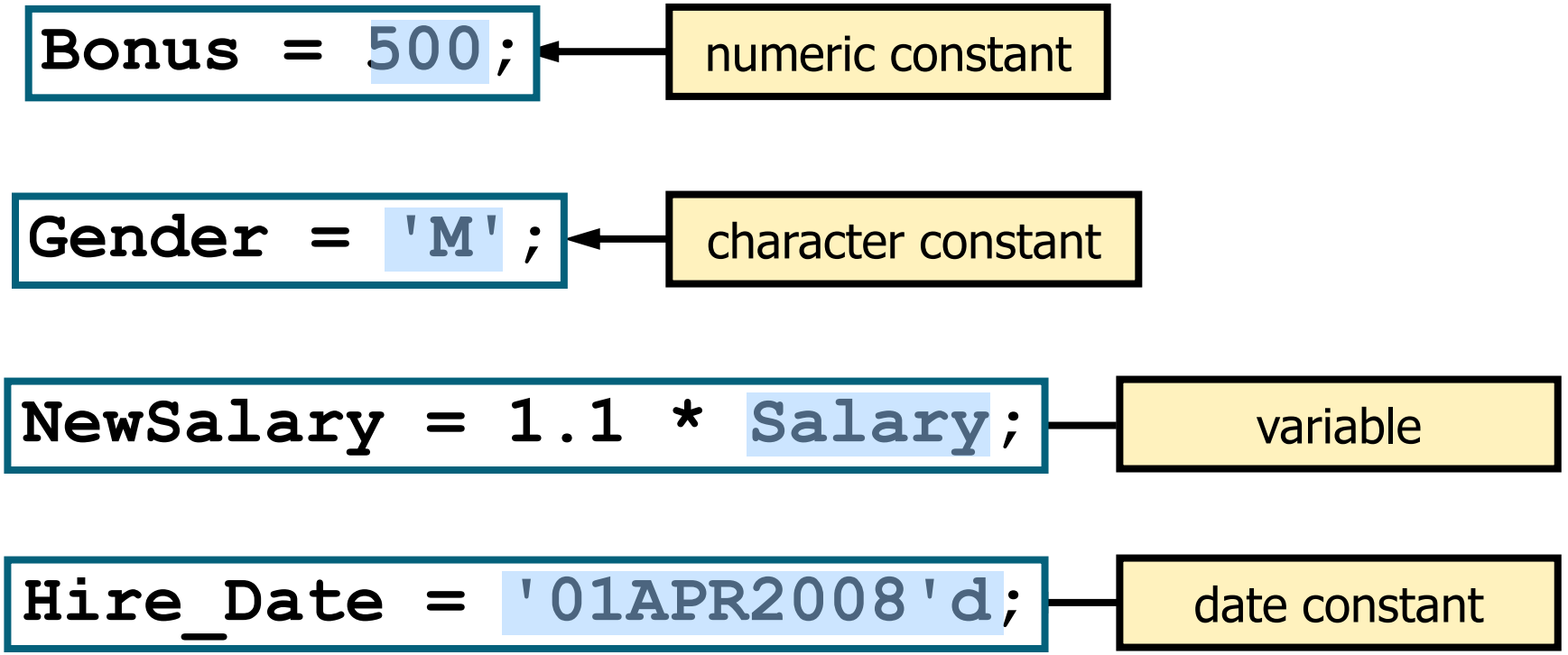
Operators are

- symbols that request arithmetic calculations
- SAS functions.

- *An expression* entered in an assignment statement is identical to an expression built using the SAS Enterprise Guide Advanced Expression Editor.

Operands

- Operands are constants (character, numeric, or date) and variables (character or numeric).
- Examples:



SAS Date Constants

The constant '***ddMMMyyyy***'**d** (example: '14dec2000'd) creates a SAS date value from the date enclosed in quotation marks.

dd	is a one- or two-digit value for the day.
MMM	is a three-letter abbreviation for the month (JAN, FEB, MAR, and so on).
yyyy	is a four-digit value for the year.
d	is required to convert the quoted string to a SAS date.

Operators

- Operators are symbols that represent an arithmetic calculation and SAS functions.

- Examples:

```
Revenue = Quantity * Price;
```

```
NewCountry = upcase(Country);
```

Arithmetic Operators

- *Arithmetic operators* indicate that an arithmetic calculation is performed.

Symbol	Definition	Priority
**	exponentiation	I
-	negative prefix	I
*	multiplication	II
/	division	II
+	addition	III
-	subtraction	III

- If a missing value is an operand for an arithmetic operator, the result is a missing value.

Multiple Choice Poll – Correct Answer

- What is the result of the assignment statement given the values of **`var1`** and **`var2`**?

a. . (missing)
b. 0
c. 5
d. 10

var1	var2
.	10

```
num = var1 + var2 / 2;
```

If an operand is missing for an arithmetic operator, the result is missing.

Using SAS Functions

- SAS functions can do the following:
  - perform arithmetic operations
  - compute sample statistics (for example: sum, mean, and standard deviation)
  - manipulate SAS dates
  - process character values
  - perform many other tasks

Sample statistics functions ignore missing values.

- SAS functions can be used in the DATA step or in the Advanced Expression Editor of the Query Builder to create new columns or filter data.

Multiple Choice Poll – Correct Answer

- What is the result of the assignment statement given the values of **`var1`**, **`var2`**, and **`var3`**?

a. . (missing)

b. 0

c. 4

d. 6

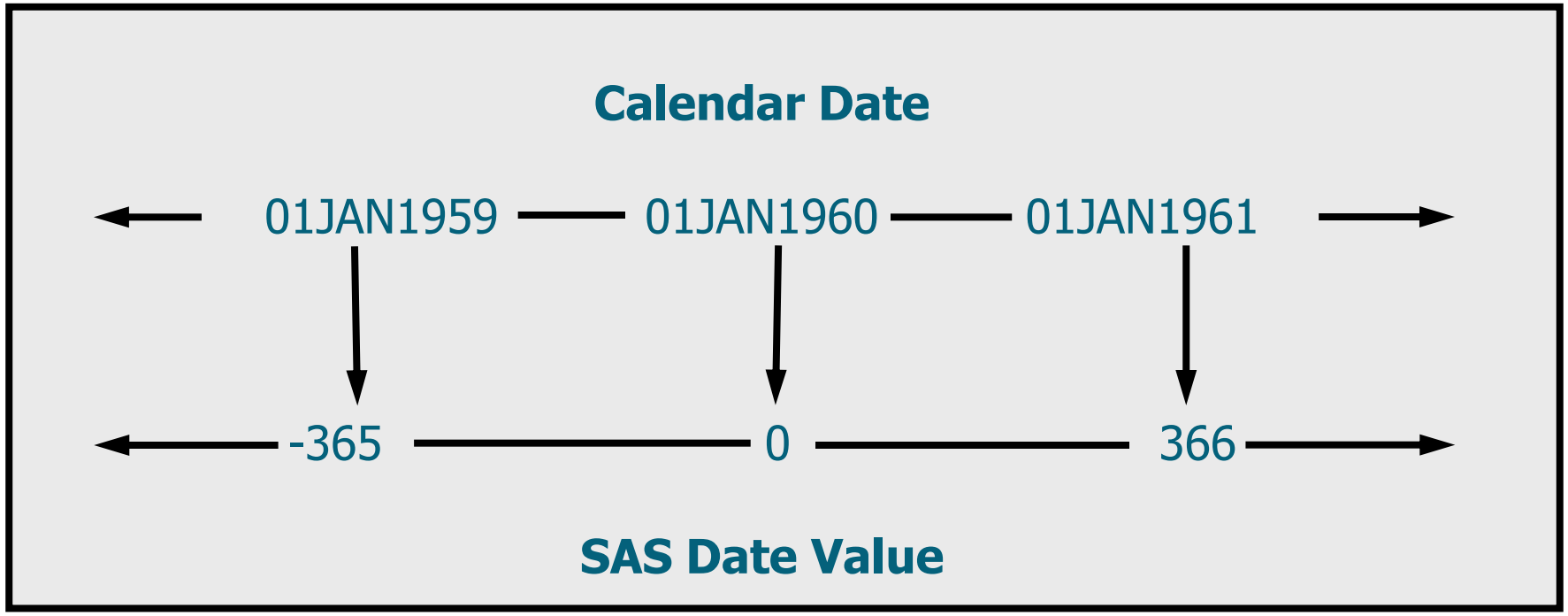
Var1	Var2	Var3
9	.	3

```
Average = mean(Var1,Var2,Var3);
```

Using Date Functions

You can use SAS date functions to do the following:

- create SAS date values
- extract information from SAS date values



Date Functions: Creating SAS Dates

TODAY()	obtains the date value from the system clock.
MDY(*month,day,year*)	uses numeric *month*, *day*, and *year* values to return the corresponding SAS date value.

- Example:

```
Days_Since_Order = today() - Order_Date;
```

Date Functions: Extracting Information

YEAR(*SAS-date*)	extracts the year from a SAS date and returns a four-digit value for year.
QTR(*SAS-date*)	extracts the quarter from a SAS date and returns a number from 1 to 4.
MONTH(*SAS-date*)	extracts the month from a SAS date and returns a number from 1 to 12.
DAY(*SAS-date*)	extracts the day of the month from a SAS date and returns a number from 1 to 31.
WEEKDAY(*SAS-date*)	extracts the day of the week from a SAS date and returns a number from 1 to 7, where 1 represents Sunday, and so on.

- Example:

```
BonusMonth = month(Hire_Date);
```

The LABEL Statement

- Permanent labels can also be assigned in the DATA step.
- General form of the LABEL statement:

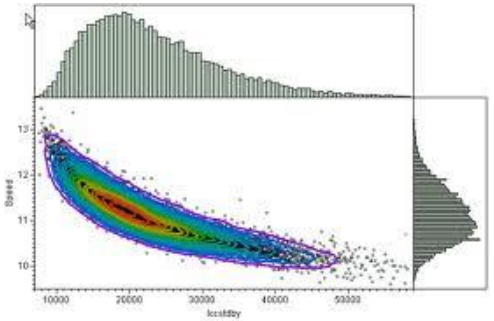
```
LABEL variable = 'label'
      variable = 'label'
      variable = 'label';
```

  - A label can be up to 256 characters.
  - Any number of variables can be associated with labels in a single LABEL statement.
  - Using a LABEL statement in a DATA step permanently associates labels with variables by storing the label in the descriptor portion of the SAS data set.

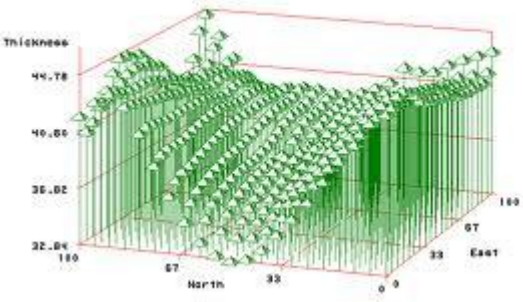
Business Scenario: Formats and Labels

```
data work.comp;
   set orion.sales;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
   drop Gender Salary Job_Title Country
        Birth_Date;
   format Bonus Compensation dollar8.
          Hire_Date date9.;
   label Employee_ID="Employee ID"
         First_Name="First Name"
         Last_Name="Last Name"
         BonusMonth="Month of Bonus"
         Hire_Date="Hire Date";
run;
```

4. Explorační analýza, vizualizace dat, kontingenční tabulky



Surface Plot of Kriged Coal Seam Thickness



Explorační analýza – PROČ?

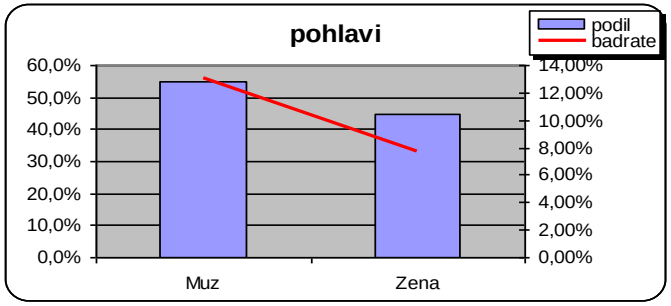
- Je třeba pochopit data:
  - najít chyby v datech
  - najít vzory v datech
  - najít porušení statistických předpokladů, testování hypotéz
  - …a především proto, že pokud to neuděláme, budeme mít velké problémy později.

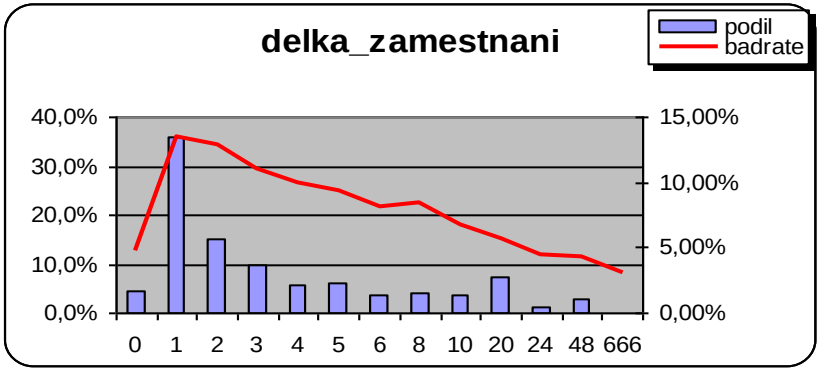
Explorace dat - jednorozměrná

- Frekvenční tabulky, histogramy:

	pocet	podil	badrate
Muz	248 768	55,0%	13,08%
Zena	203 194	45,0%	7,69%
Total	451 962	100,0%	10,66%

delka_zamestnani	pocet	podil	badrate
0	20 825	4,6%	4,69%
1	163 144	36,1%	13,43%
2	67 462	14,9%	12,80%
3	43 778	9,7%	10,97%
4	26 256	5,8%	10,01%
5	27 526	6,1%	9,32%
6	15 893	3,5%	8,16%
8	18 036	4,0%	8,39%
10	17 195	3,8%	6,72%
20	33 641	7,4%	5,60%
24	5 176	1,1%	4,48%
48	12 934	2,9%	4,28%
666	96	0,0%	3,13%
Total	451 962	100,0%	10,66%

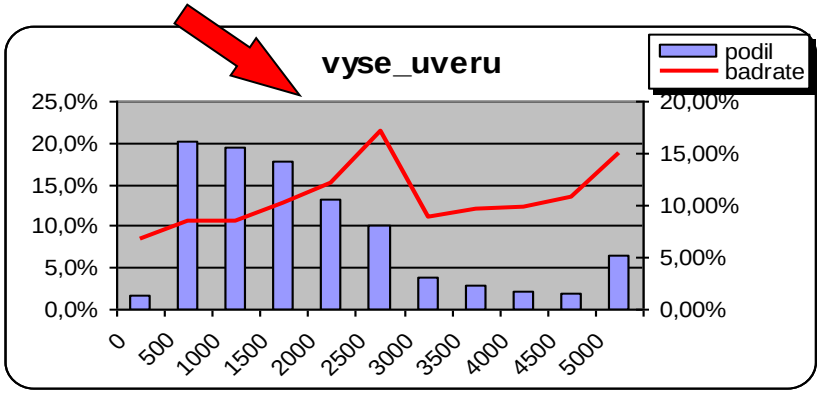




Explorace dat - jednorozměrná

- výše úvěru vs. bad rate

OK? Nebo je to způsobeno jiným faktorem???



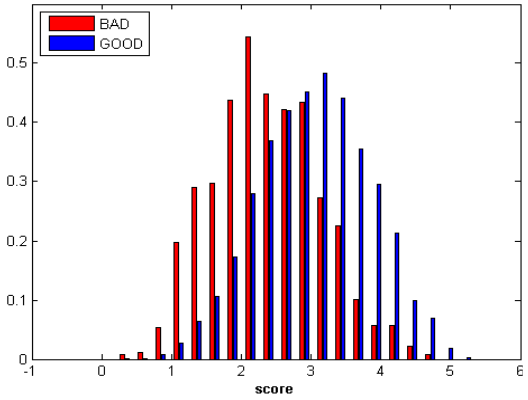
Explorace dat - jednorozměrná

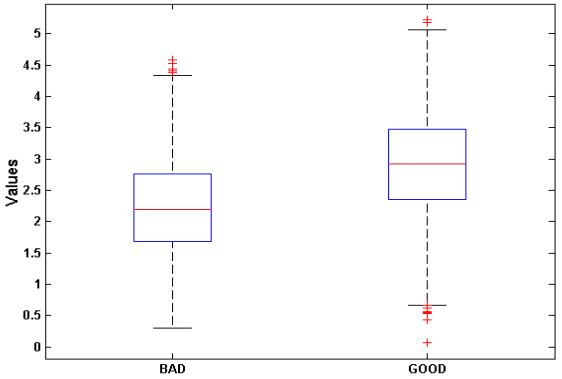
- spojité proměnné:
  - Průměr
  - Modus
  - Kvantily
  - Rozptyl
  - Min./maximální hodnota

- je vhodná kategorizace

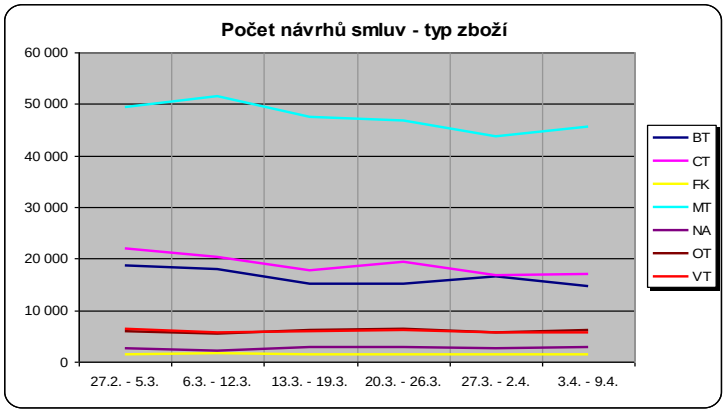
Explorace dat - jednorozměrná

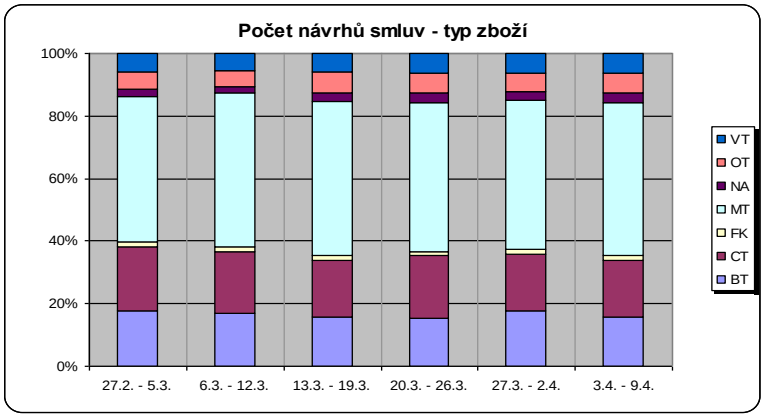
- Histogramy, box ploty





- Stabilita v čase





Explorace dat - vícerozměrná

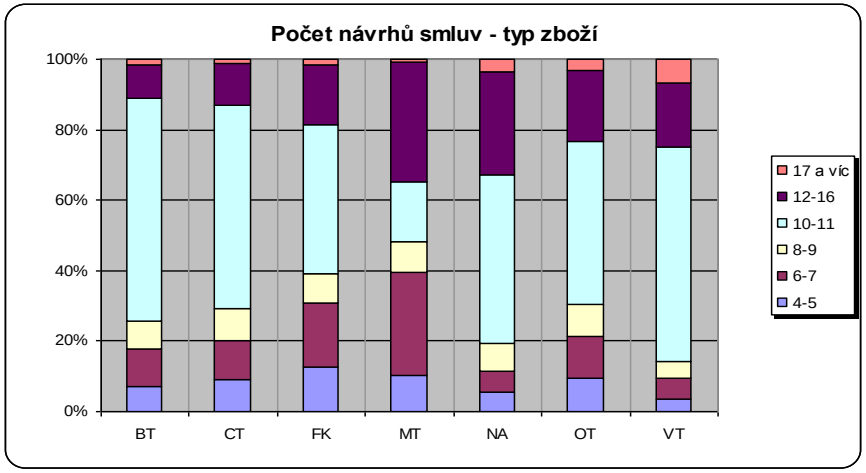
- Kontingenční tabulky

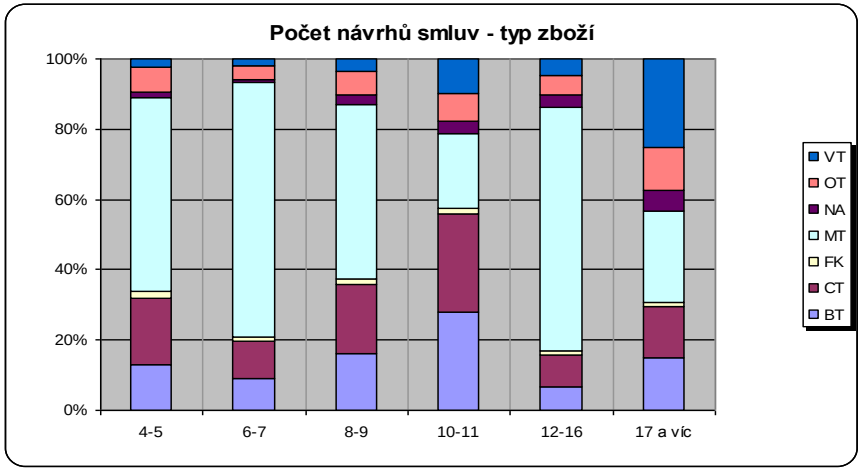
	do 5 000	5 000 - 10 000	10 000 - 15 000	víc než 15 000
BT	4 291	8 581	9 176	9 044
CT	7 587	12 493	6 500	7 236
FK	258	1 017	851	557
MT	27 191	39 551	16 524	5 992
NA	426	1 088	1 114	2 737
OT	2 478	3 689	2 103	3 475
VT	384	1 001	963	9 086

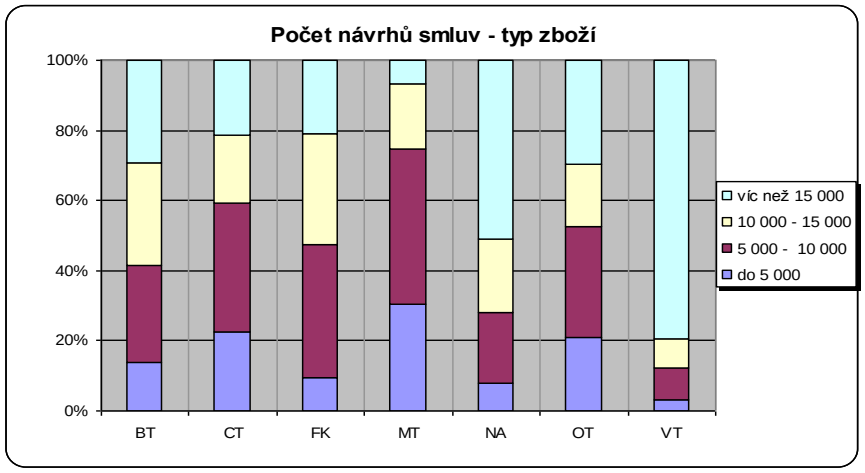
row%	do 5 000	5 000 - 10 000	10 000 - 15 000	víc než 15 000
BT	13,8%	27,6%	29,5%	29,1%
CT	22,4%	36,9%	19,2%	21,4%
FK	9,6%	37,9%	31,7%	20,8%
MT	30,5%	44,3%	18,5%	6,7%
NA	7,9%	20,3%	20,8%	51,0%
OT	21,1%	31,4%	17,9%	29,6%
VT	3,4%	8,8%	8,4%	79,5%

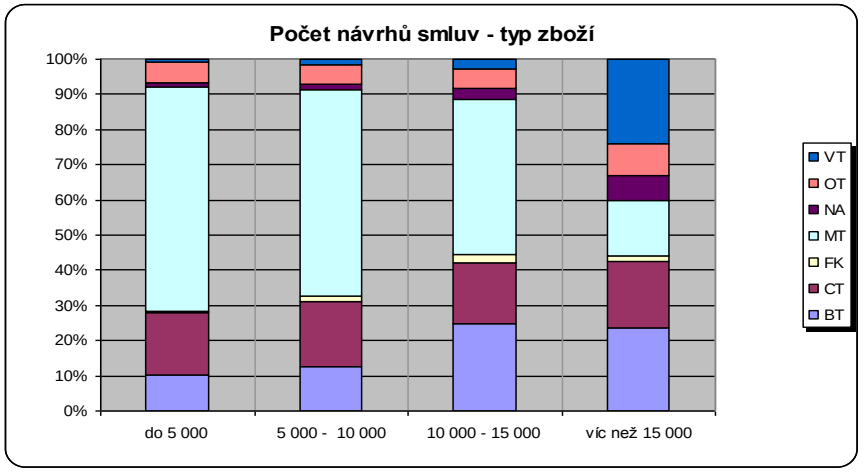
col%	do 5 000	5 000 - 10 000	10 000 - 15 000	víc než 15 000
BT	10,1%	12,7%	24,6%	23,7%
CT	17,8%	18,5%	17,5%	19,0%
FK	0,6%	1,5%	2,3%	1,5%
MT	63,8%	58,7%	44,4%	15,7%
NA	1,0%	1,6%	3,0%	7,2%
OT	5,8%	5,5%	5,6%	9,1%
VT	0,9%	1,5%	2,6%	23,8%

Explorace dat - vícerozměrná



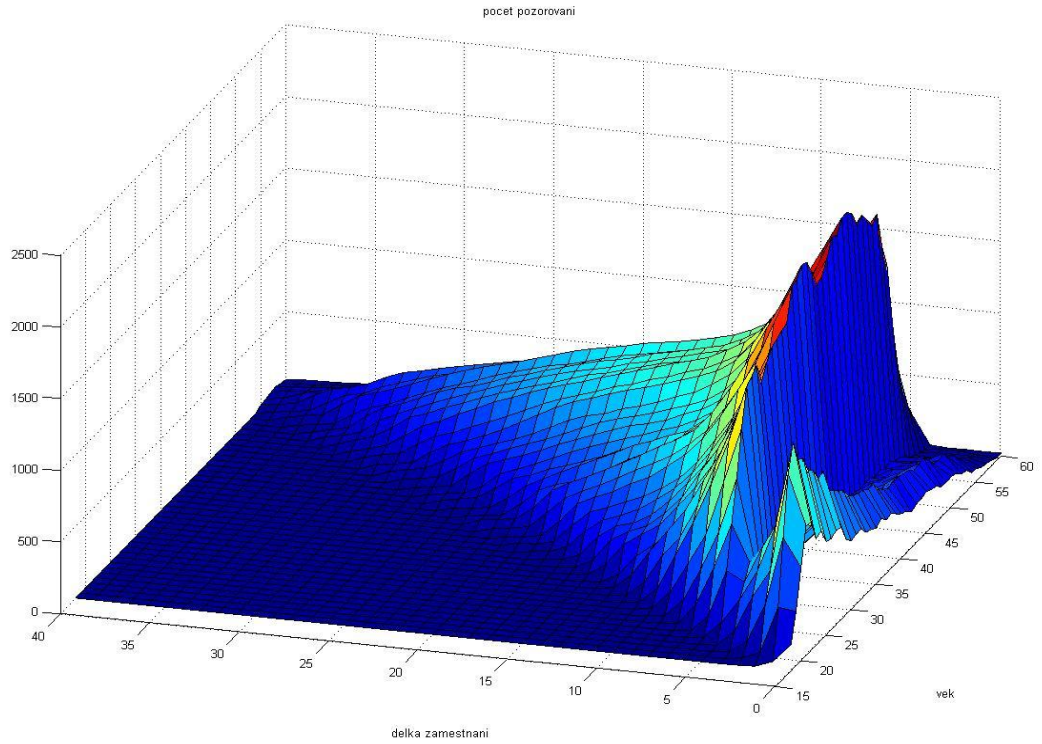






Explorace dat - vícerozměrná

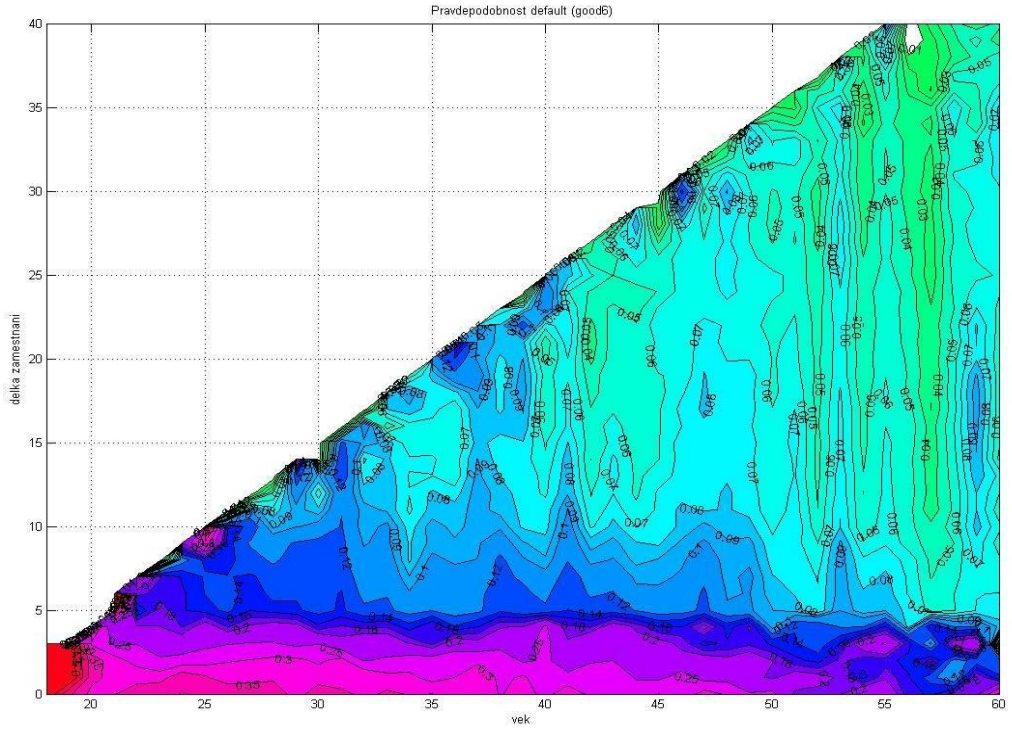
- Věk vs. délka zaměstnání



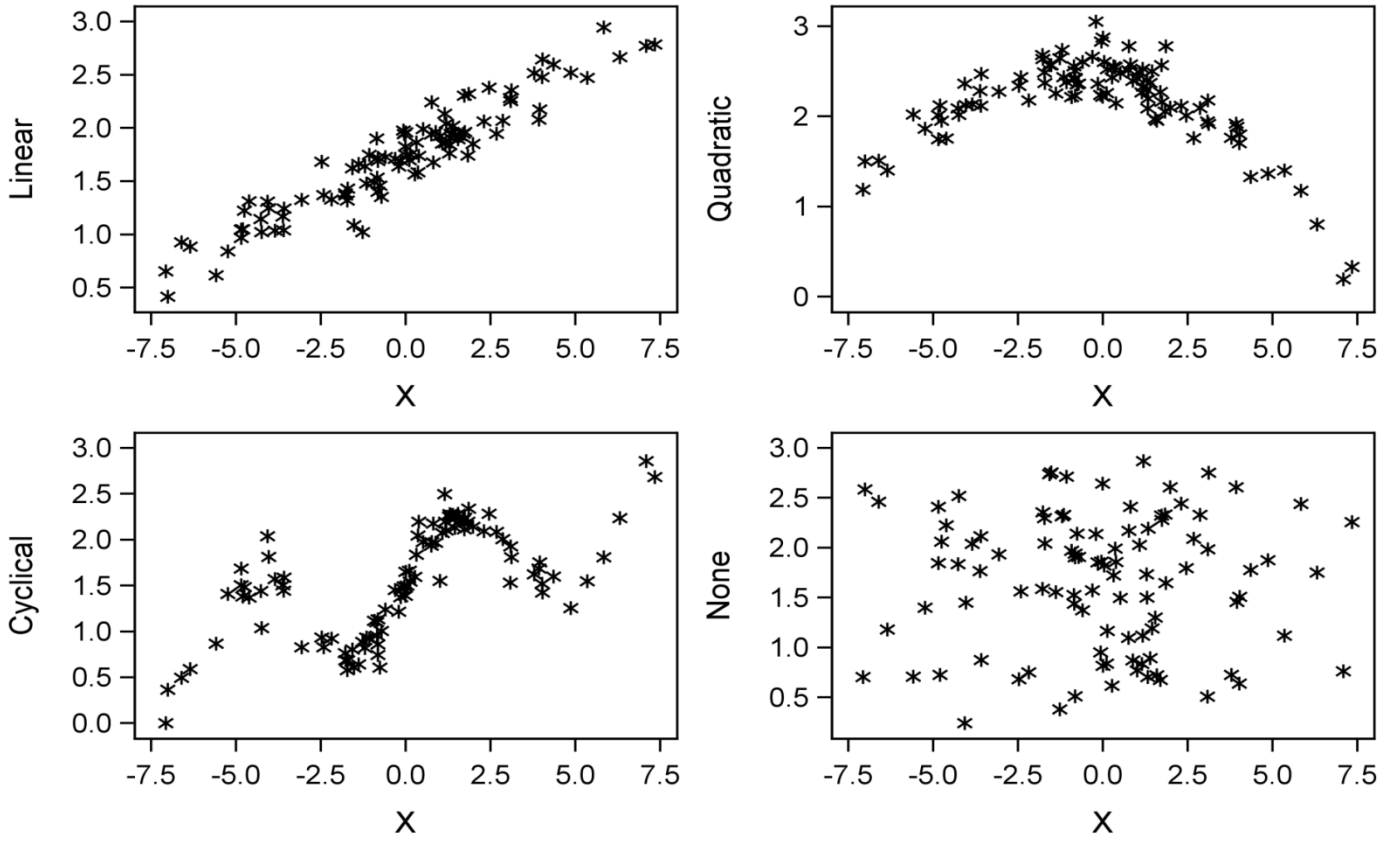
5 let
...defaultní hodnota???

Explorace dat - vícerozměrná

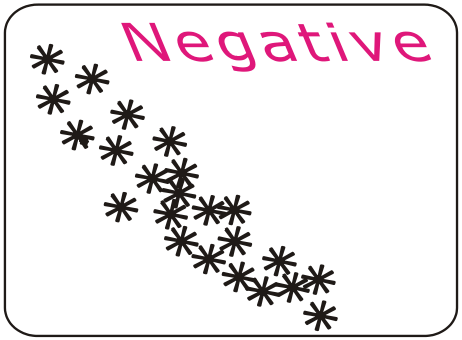
- Věk vs. délka zaměstnání vs. default

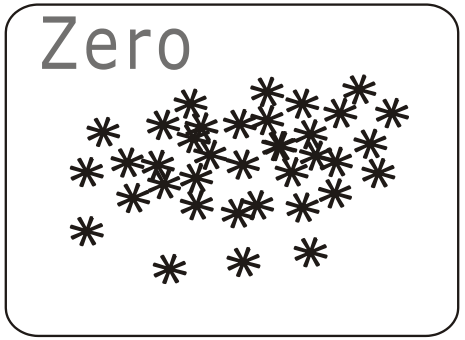


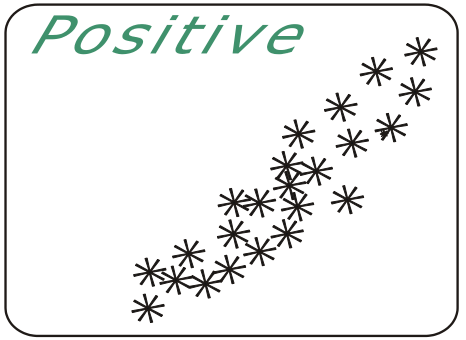
Relationships between Continuous Variables – scatter plots

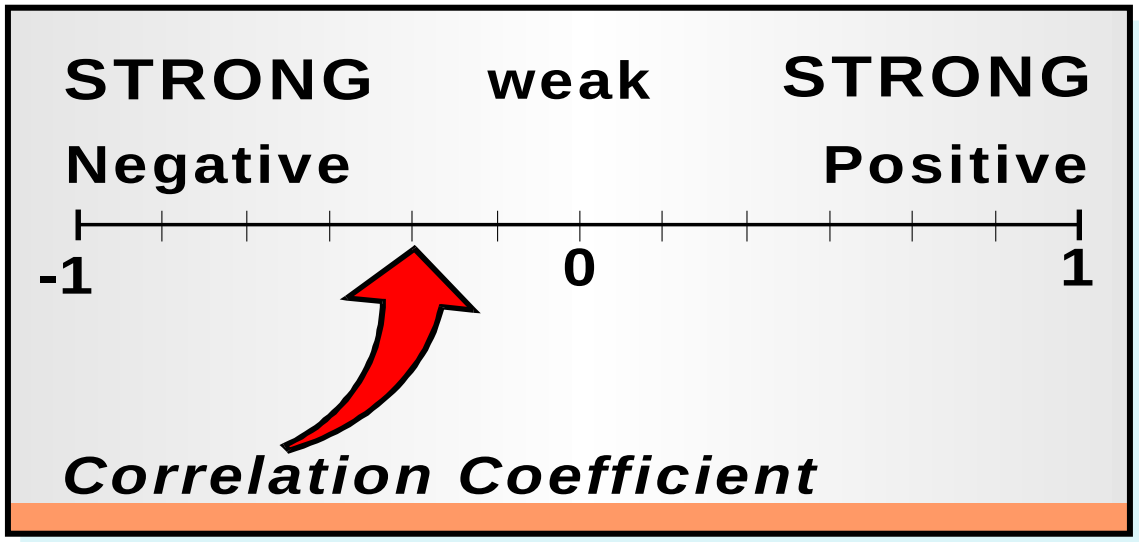


Correlation



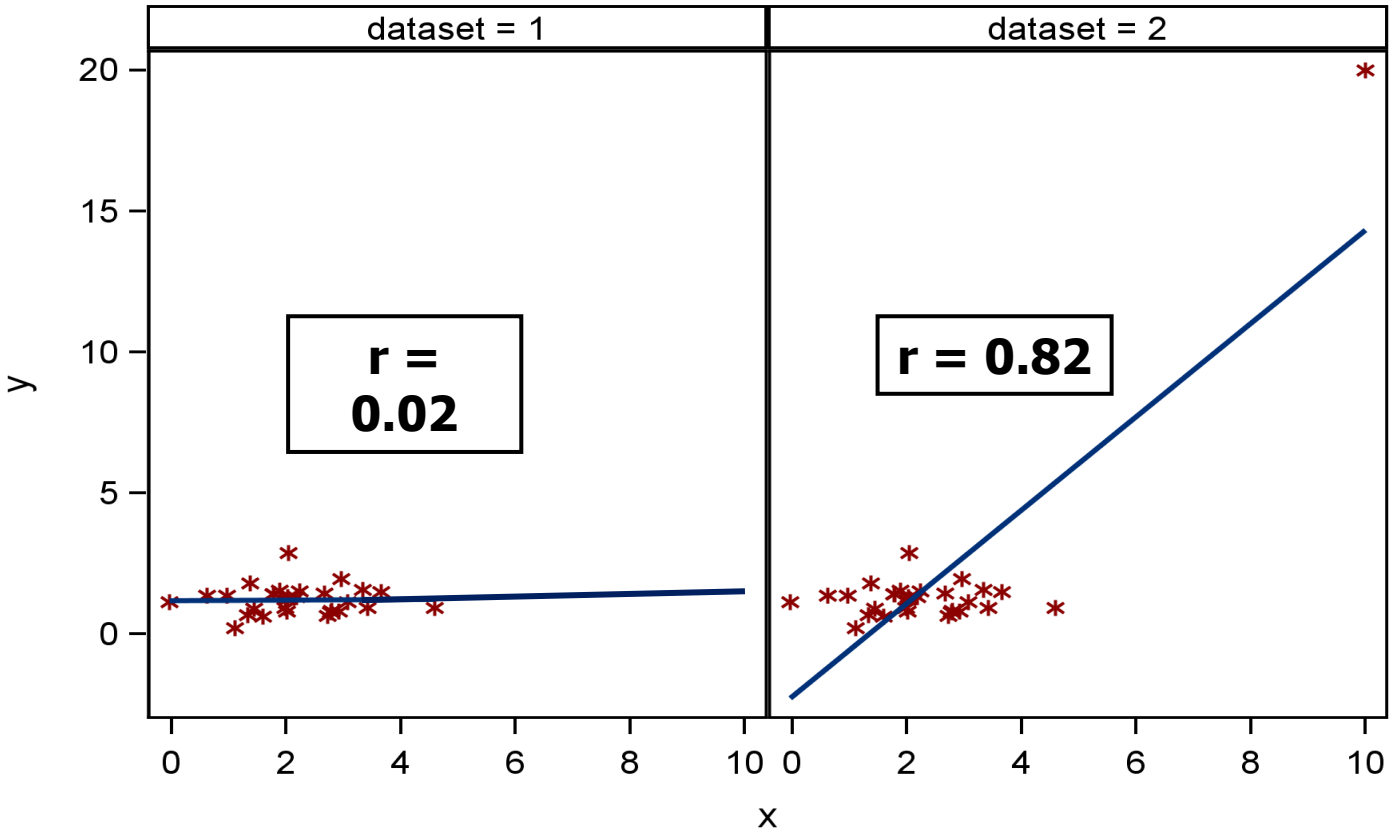






Extreme Data Values

Odlehlé (extrémní) hodnoty mohou zcela zkreslit výsledky analýzy.



Diskriminační síla proměnných pro prediktivní modely

Weight of evidence, information value

r ... number of levels (categories) of the categorical variable
g_i ... number of ″goods″ the in i-th category
b_i ... number of ″bads″ the in i-th category
$G := \Sigma g_i$ ... total number of ″goods″
$B := \Sigma b_i$ ... total number of ″bads″

Weight of evidence for the i-th category: $\boldsymbol{woe_i = ln (g_i / G) - ln (b_i / B)}$

Information value for the i-th category: $\boldsymbol{Inf\_val_i = [(g_i / G) - (b_i / B)] \cdot woe_i}$

Total information value for the corresponding variable: $\boldsymbol{Inf\_val = \Sigma\, inf\_val_i}$

Diskriminační síla proměnných

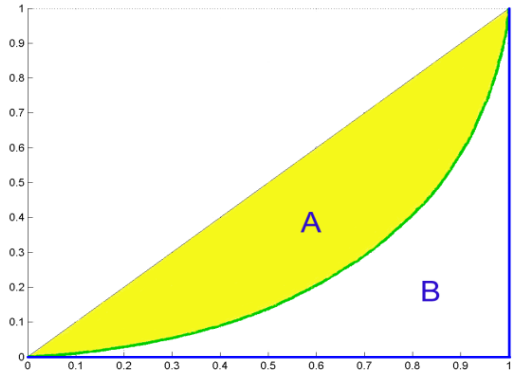
Incorporation Date											
Raw	RegVar	Percant	B	G	TOT	G/B Odds	%Good	%Bad	Bad Rate	WoE	IV
0 & NOI	inc_1	12%	139	952	1091	7	11%	19%	12,7%	-0,557	0,046116
1	inc_2	13%	133	1073	1206	8	12%	19%	11,0%	-0,394	0,023731
2-7	miss	42%	299	3601	3900	12	42%	42%	7,7%	0,007	2,04E-05
8-15	inc_3	22%	108	1942	2050	18	23%	15%	5,3%	0,408	0,030887
16+	inc_4	11%	39	1019	1058	26	12%	5%	3,7%	0,781	0,050288
Total			718	8587	9305	12			**7,7%**		**0,151**

- **<0.02 unpredictive**
- **0.02 – 0.1 weak**
- **0.1 – 0.3 medium**
- **0.3 – 0.5 strong**
- **> 0.5 too high …je třeba prověřit, pravděpodbně je něco špatně**

Diskriminační síla proměnných

- Lorenzova křivka, Giniho index

$$x = F_{m.BAD}(a)$$
$$y = F_{n.GOOD}(a), a \in [L, H].$$

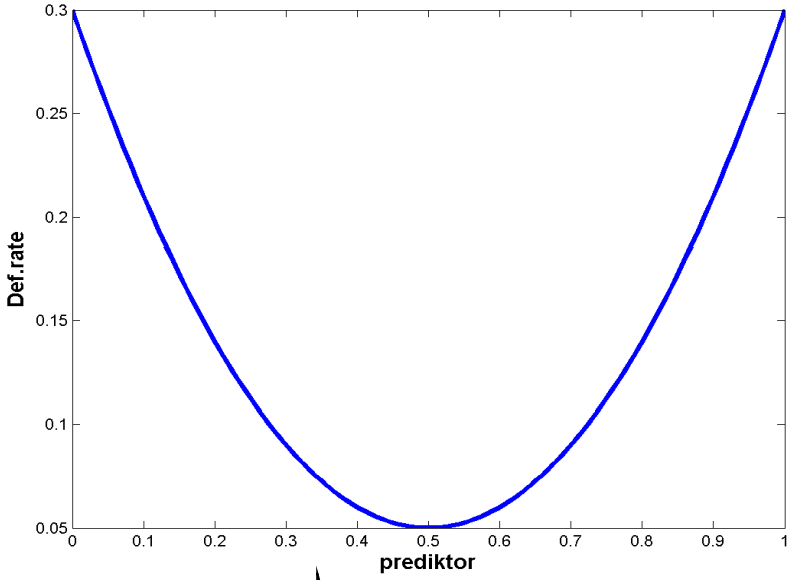


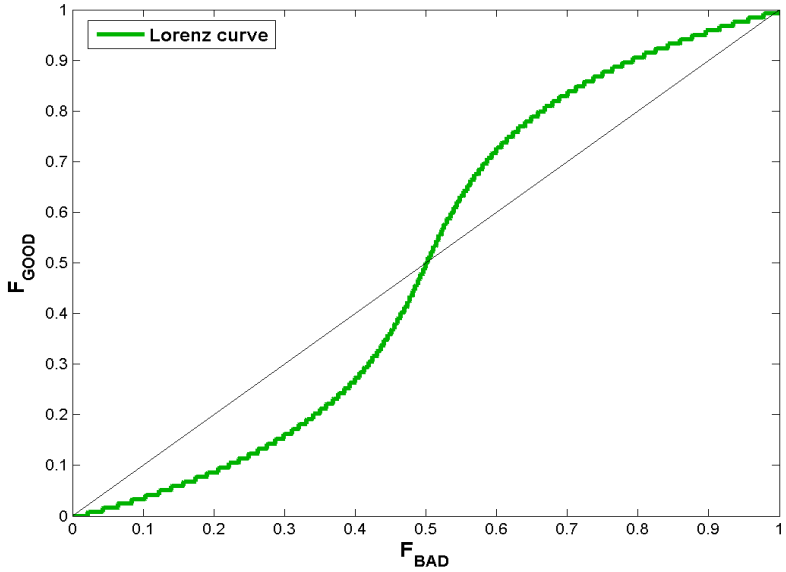
$$Gini = \frac{A}{A+B} = 2A$$

$$Gini = 1 - \sum_{k=2}^{n+m} (F_{m.BAD\,k} - F_{m.BAD\,k-1}) \cdot (F_{n.GOOD_{\,k}} + F_{n.GOOD\,k-1})$$

Diskriminační síla proměnných

- Lorenzova křivka …kontrola monotónnosti vysvětlované proměnné (def. rate) na dané vysvětlující proměnné





Kategorizace (WOE)

Diskriminační síla proměnných

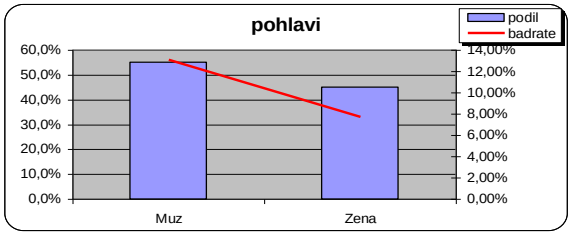
- Giniho index

<0.05	unpredictive
0.05 – 0.1	weak
0.1 – 0.2	medium
0.2 – 0.5	strong
> 0.5	too high …je třeba prověřit, pravděpodbně je něco špatně

Diskriminační síla proměnných

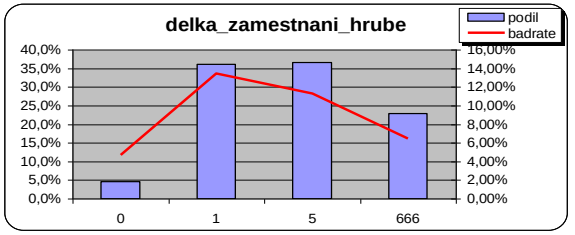
pohlavi	Gini:	0,1401	Info.Value:	0,0828

	pocet	podil	badrate
Muz	248 768	55,0%	13,08%
Zena	203 194	45,0%	7,69%
Total	451 962	100,0%	10,66%



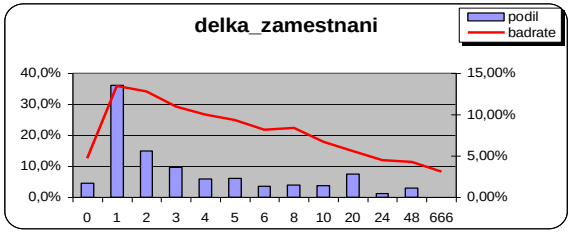
delka_zamestnani_hrube	Gini:	0,1611	Info.Value:	0,1100

	pocet	podil	badrate
0	20 825	4,6%	4,69%
1	163 144	36,1%	13,43%
5	165 022	36,5%	11,29%
666	102 971	22,8%	6,45%
Total	451 962	100,0%	10,66%

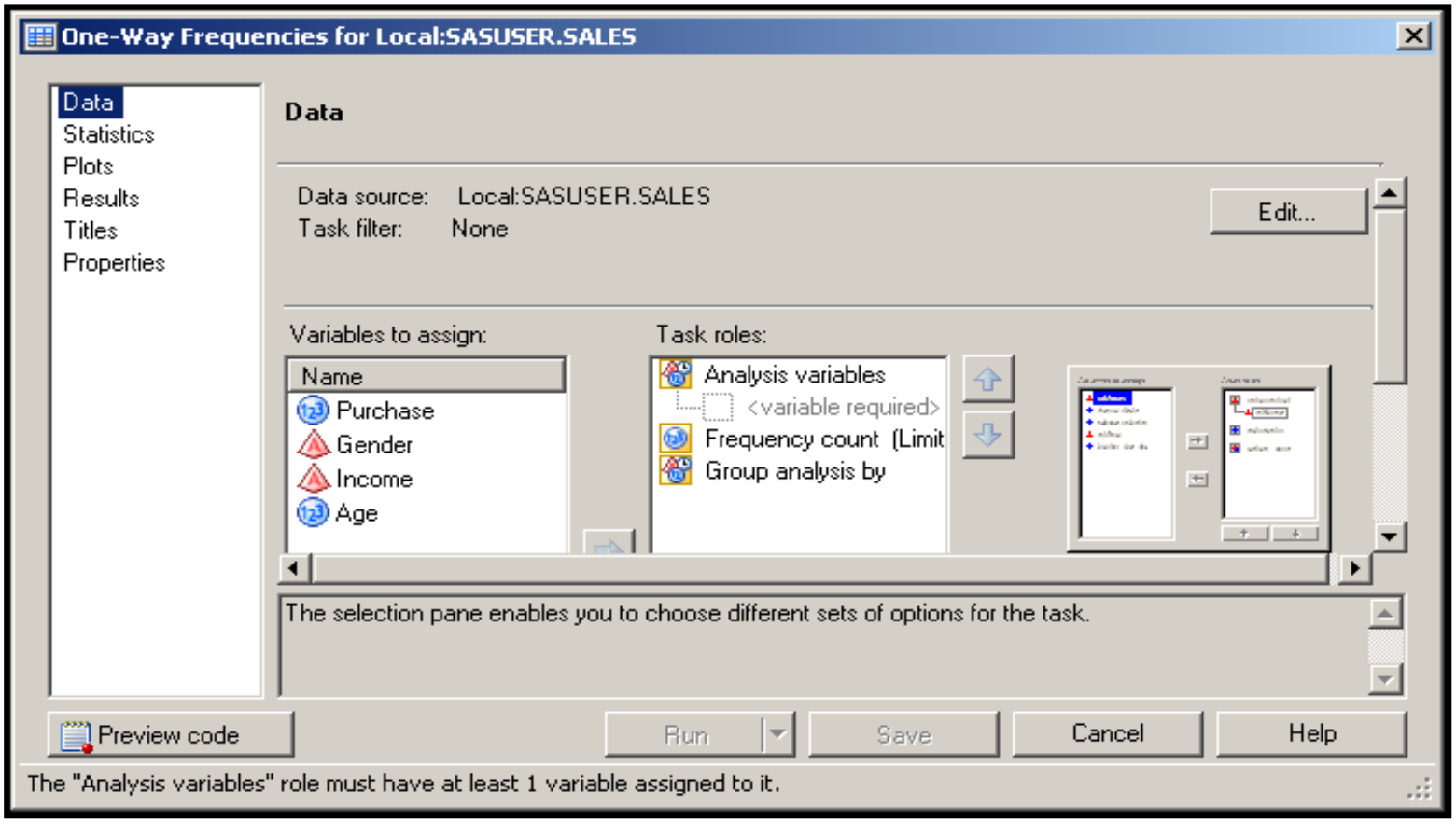


delka_zamestnani_jemne	Gini:	0,1762	Info.Value:	0,1285

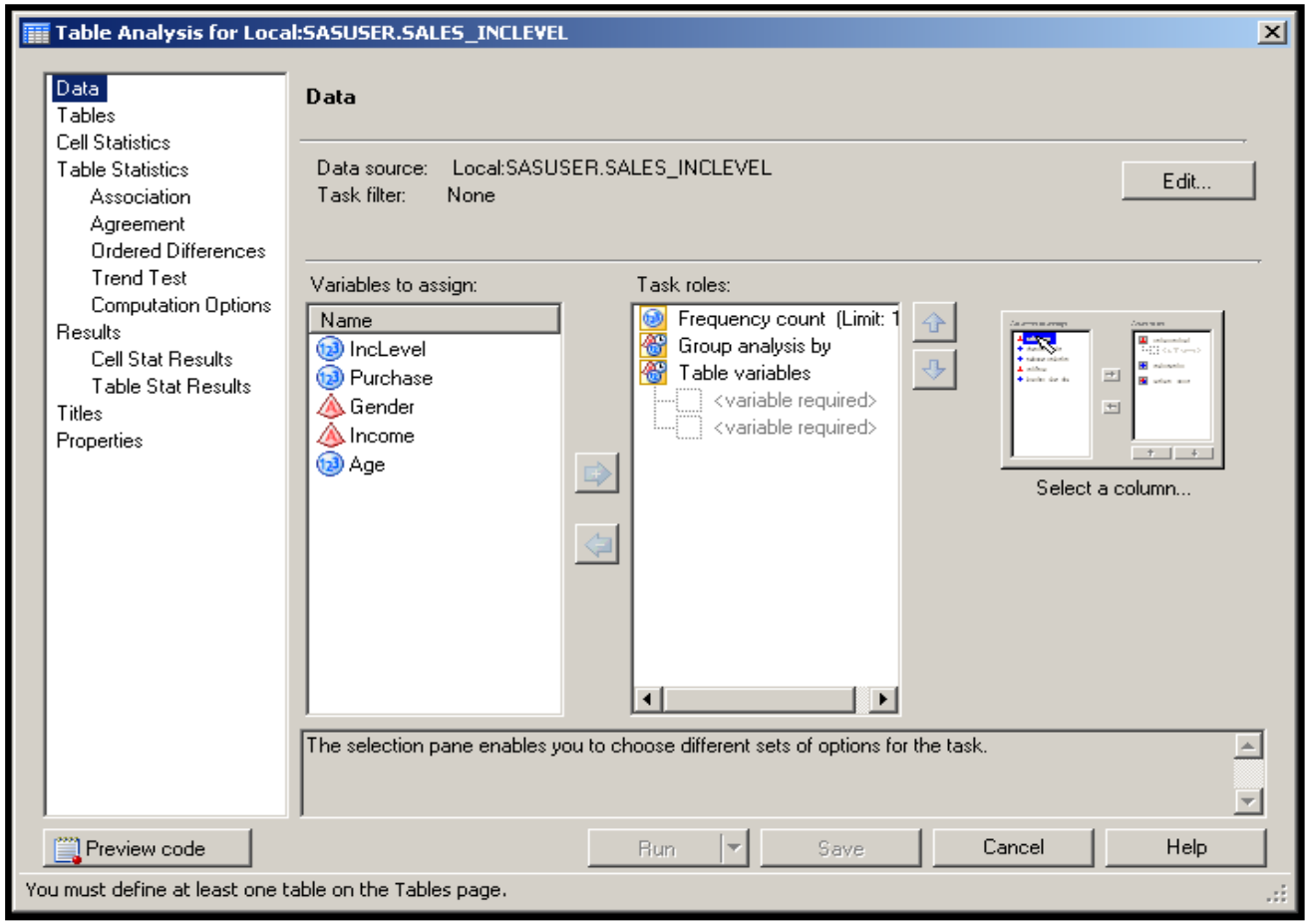
delka_zamestnani	pocet	podil	badrate
0	20 825	4,6%	4,69%
1	163 144	36,1%	13,43%
2	67 462	14,9%	12,80%
3	43 778	9,7%	10,97%
4	26 256	5,8%	10,01%
5	27 526	6,1%	9,32%
6	15 893	3,5%	8,16%
8	18 036	4,0%	8,39%
10	17 195	3,8%	6,72%
20	33 641	7,4%	5,60%
24	5 176	1,1%	4,48%
48	12 934	2,9%	4,28%
666	96	0,0%	3,13%
Total	451 962	100,0%	10,66%



The One-Way Frequencies Task



The Table Analysis Task



The FREQ Procedure

- The FREQ procedure can do the following:
  - produce one-way to *n*-way frequency and crosstabulation (contingency) tables
  - compute chi-square tests for one-way to *n*-way tables and measures of association and agreement for contingency tables
  - automatically display the output in a report and save the output in a SAS data set
- General form of the FREQ procedure:

```
PROC FREQ DATA=SAS-data-set
<option(s)>;
      TABLES variable(s) <I option(s)>;
RUN;
```

The FREQ Procedure

A FREQ procedure with no TABLES statement generates one-way frequency tables for all data set variables.

```
proc freq data=orion.sales;
run;
```

This PROC FREQ step creates a frequency table for the following nine variables:

- Employee_ID
- First_Name
- Last_Name
- Gender
- Salary
- Job_Title
- Country
- Birth_Date
- Hire_Date

The TABLES Statement

The TABLES statement specifies the frequency and crosstabulation tables to produce.

```
proc freq data=orion.sales;
   tables Gender Country;
run;
```

one-way frequency tables

An asterisk between variables requests a *n*-way crosstabulation table.

```
proc freq data=orion.sales;
   tables Gender*Country;
run;
```

two-way frequency table

The TABLES Statement

A one-way frequency table produces frequencies, cumulative frequencies, percentages, and cumulative percentages.

```
proc freq data=orion.sales;
   tables Gender Country;
run;
```

The FREQ Procedure

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	68	41.21	68	41.21
M	97	58.79	165	100.00

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

The TABLES Statement

An *n*-way frequency table produces cell frequencies, cell percentages, cell percentages of row frequencies, and cell percentages of column frequencies, plus total frequency and percent.

```
proc freq data=orion.sales;
   tables Gender*Country;
run;
```

rows

columns

The TABLES Statement

The FREQ Procedure

Table of Gender by Country

Gender	Country		
Frequency Percent Row Pct Col Pct	AU	US	Total
F	27 16.36 39.71 42.86	41 24.85 60.29 40.20	68 41.21
M	36 21.82 37.11 57.14	61 36.97 62.89 59.80	97 58.79
Total	63 38.18	102 61.82	165 100.00

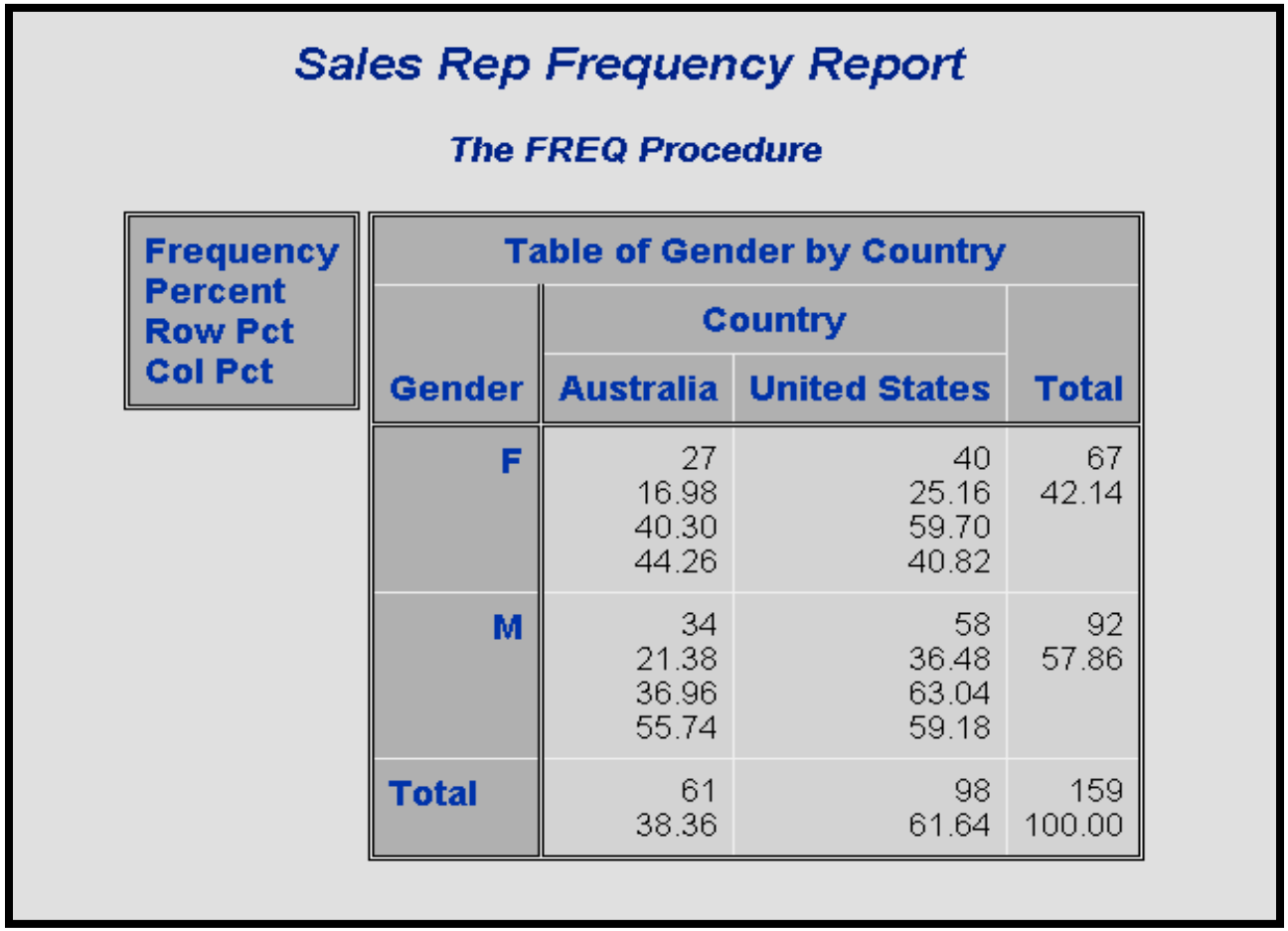
Additional SAS Statements

- Additional statements can be added to enhance the report.

```
proc format;
   value $ctryfmt 'AU'='Australia'
                  'US'='United States';
run;

options nodate pageno=1;

ods html file='p112d01.html';
proc freq data=orion.sales;
   tables Gender*Country;
   where Job_Title contains 'Rep';
   format Country $ctryfmt.;
   title 'Sales Rep Frequency Report';
run;
ods html close;
```

Additional SAS Statements

- HTML Output



Sales Rep Frequency Report

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Gender by Country		
	Country		
Gender	Australia	United States	Total
F	27 16.98 40.30 44.26	40 25.16 59.70 40.82	67 42.14
M	34 21.38 36.96 55.74	58 36.48 63.04 59.18	92 57.86
Total	61 38.36	98 61.64	159 100.00

Options to Suppress Display of Statistics

- Options can be placed in the TABLES statement after a forward slash to suppress the display of the default statistics.

Option	Description
NOCUM	suppresses the display of cumulative frequency and cumulative percentage.
NOPERCENT	suppresses the display of percentage, cumulative percentage, and total percentage.
NOFREQ	suppresses the display of the cell frequency and total frequency.
NOROW	suppresses the display of the row percentage.
NOCOL	suppresses the display of the column percentage.

Additional TABLES Statement Options

- Additional options can be placed in the TABLES statement after a forward slash to control the displayed output.

Option	Description
LIST	displays *n*-way tables in list format.
CROSSLIST	displays *n*-way tables in column format.
FORMAT=	formats the frequencies in *n*-way tables.

LIST and CROSSLIST Options

Gender	Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	Australia	27	16.36	27	16.36
F	United States	41	24.85	68	41.21
M	Australia	36	21.82	104	63.03
M	United States	61	36.97	165	100.00

```
tables Gender*Country / list;
```

Table of Gender by Country

Gender	Country	Frequency	Percent	Row Percent	Column Percent
F	Australia	27	16.36	39.71	42.86
	United States	41	24.85	60.29	40.20
	Total	68	41.21	100.00	
M	Australia	36	21.82	37.11	57.14
	United States	61	36.97	62.89	59.80
	Total	97	58.79	100.00	
Total	Australia	63	38.18		100.00
	United States	102	61.82		100.00
	Total	165	100.00		

```
tables Gender*Country / crosslist;
```

PROC FREQ Statement Options

- Options can also be placed in the PROC FREQ statement.

Option	Description
NLEVELS	displays a table that provides the number of levels for each variable named in the TABLES statement.
PAGE	displays only one table per page.
COMPRESS	begins the display of the next one-way frequency table on the same page as the preceding one-way table if there is enough space to begin the table.

NLEVELS Option

```
proc freq data=orion.sales nlevels;
   tables Gender Country Employee_ID;
run;
```

Partial PROC FREQ Output

The FREQ Procedure

Number of Variable Levels

Variable	Levels
Gender	2
Country	2
Employee_ID	165

Output Data Sets

- PROC FREQ produces output data sets using two different methods.
  - The TABLES statement with an OUT= option is used to create a data set with frequencies and percentages.

TABLES *variables* **/ OUT=***SAS-data-set* <*options*>**;**

  - The OUTPUT statement with an OUT= option is used to create a data set with specified statistics such as the chi-square statistic.

OUTPUT OUT=*SAS-data-set* <*options*>**;**

The MEANS Procedure

- The *MEANS procedure* provides data summarization tools to compute descriptive statistics for variables across all observations and within groups of observations.

General form of the MEANS procedure:

```
PROC MEANS DATA=SAS-data-set <statistic(s)> <option(s)>;
    VAR analysis-variable(s);
    CLASS classification-variable(s);
RUN;
```

The MEANS Procedure

- By default, the MEANS procedure reports the number of nonmissing observations, the mean, the standard deviation, the minimum value, and the maximum value of all numeric variables.

```
proc means data=orion.sales;
run;
```

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
Employee_ID	165	120713.90	450.0866939	120102.00	121145.00
Salary	165	31160.12	20082.67	22710.00	243190.00
Birth_Date	165	3622.58	5456.29	-5842.00	10490.00
Hire_Date	165	12054.28	4619.94	5114.00	17167.00

p112d05

The VAR Statement

The *VAR statement* identifies the analysis variables and their order in the results.

```
proc means data=orion.sales;
   var Salary;
run;
```

The MEANS Procedure

Analysis Variable : Salary

N	Mean	Std Dev	Minimum	Maximum
165	31160.12	20082.67	22710.00	243190.00

The CLASS Statement

- The *CLASS statement* identifies variables whose values define subgroups for the analysis.

```
proc means data=orion.sales;
   var Salary;
   class Gender Country;
run;
```

The MEANS Procedure

Analysis Variable : Salary

Gender	Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
F	AU	27	27	27702.41	1728.23	25185.00	30890.00
	US	41	41	29460.98	8847.03	25390.00	83505.00
M	AU	36	36	32001.39	16592.45	25745.00	108255.00
	US	61	61	33336.15	29592.69	22710.00	243190.00

The CLASS Statement

```
proc means data=orion.sales;
   var Salary;
   class Gender Country;
run;
```

classification variables

The MEANS Procedure

Analysis Variable : Salary

analysis variable

Gender	Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
F	AU	27	27	27702.41	1728.23	25185.00	30890.00
	US	41	41	29460.98	3847.03	25200.00	83505.00
M	AU	36	36	32001.3			00
	US	61	61	33336.15	29592.69	22710.00	243190.00

statistics for analysis variable

The CLASS statement adds the N Obs column, which is the number of observations for each unique combination of the class variables.

PROC MEANS Statistics

- The statistics to compute and the order to display them can be specified in the PROC MEANS statement.

```
proc means data=orion.sales sum mean range;
   var Salary;
   class Country;
run;
```

The MEANS Procedure

Analysis Variable : Salary

Country	N Obs	Sum	Mean	Range
AU	63	1900015.00	30158.97	83070.00
US	102	3241405.00	31778.48	220480.00

PROC MEANS Statistics

Descriptive Statistic Keywords				
CLM	CSS	CV	LCLM	MAX
MEAN	MIN	MODE	N	NMISS
KURTOSIS	RANGE	SKEWNESS	STDDEV	STDERR
SUM	SUMWGT	UCLM	USS	VAR
Quantile Statistic Keywords				
MEDIAN | P50	P1	P5	P10	Q1 | P25
Q3 | P75	P90	P95	P99	QRANGE
Hypothesis Testing Keywords				
PROBT	T			

PROC MEANS Statement Options

- Options can also be placed in the PROC MEANS statement.

Option	Description
MAXDEC=	specifies the number of decimal places to use in printing the statistics.
FW=	specifies the field width to use in displaying the statistics.
NONOBS	suppresses reporting the total number of observations for each unique combination of the class variables.

MAXDEC= Option

```
proc means data=orion.sales maxdec=0;
```

Analysis Variable : Salary

Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
AU	63	63	30159	12699	25185	108255
US	102	102	31778	23556	22710	243190

```
proc means data=orion.sales maxdec=1;
```

Analysis Variable : Salary

Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
AU	63	63	30159.0	12699.1	25185.0	108255.0
US	102	102	31778.5	23555.8	22710.0	243190.0

FW= Option

```
proc means data=orion.sales;
```

Analysis Variable : Salary

Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
AU	63	63	30158.97	12699.14	25185.00	108255.00
US	102	102	31778.48	23555.84	22710.00	243190.00

```
proc means data=orion.sales fw=15;
```

Analysis Variable : Salary

Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
AU	63	63	30158.96825397	12699.13932690	25185.00000000	108255
US	102	102	31778.48039216	23555.84171928	22710.00000000	243190

NONOBS Option

```
proc means data=orion.sales;
```

Analysis Variable : Salary

Country	N Obs	N	Mean	Std Dev	Minimum	Maximum
AU	63	63	30158.97	12699.14	25185.00	108255.00
US	102	102	31778.48	23555.84	22710.00	243190.00

```
proc means data=orion.sales nonobs;
```

Analysis Variable : Salary

Country	N	Mean	Std Dev	Minimum	Maximum
AU	63	30158.97	12699.14	25185.00	108255.00
US	102	31778.48	23555.84	22710.00	243190.00

Output Data Sets

- PROC MEANS produces output data sets using the following method:

OUTPUT OUT=*SAS-data-set* <*options*>**;**

- The output data set contains the following variables:
  - BY variables
  - class variables
  - the automatic variables **_TYPE_** and **_FREQ_**
  - the variables requested in the OUTPUT statement

OUTPUT Statement OUT= Option

The statistics in the PROC statement impact only the MEANS report, not the data set.

```
proc means data=orion.sales sum mean range;
   var Salary;
   class Gender Country;
   output out=work.means1;
run;

proc print data=work.means1;
run;
```

p112d06

OUTPUT Statement OUT= Option

Obs	Gender	Country	_TYPE_	_FREQ_	_STAT_	Salary
1			0	165	N	165.00
2			0	165	MIN	22710.00
3			0	165	MAX	243190.00
4			0	165	MEAN	31160.12
5			0	165	STD	20082.67
6		AU	1	63	N	63.00
7					MIN	25185.00
8					MAX	108255.00
9		AU	1	63	MEAN	30158.97
10		AU	1	63	STD	12699.14
11		US	1	102	N	102.00
12		US	1	102	MIN	22710.00
13		US	1	102	MAX	243190.00
14		US	1	102	MEAN	31778.48
15		US	1	102	STD	23555.84
16	F		2	68	N	68.00
17	F		2	68	MIN	25185.00
18	F		2	68	MAX	83505.00
19	F		2	68	MEAN	28762.72
20	F		2	68	STD	6974.15

default statistics

OUTPUT Statement OUT= Option

- The OUTPUT statement can also do the following:
  - specify the statistics for the output data set
  - select and name variables

```
proc means data=orion.sales noprint;
   var Salary;
   class Gender Country;
   output out=work.means2
          min=minSalary max=maxSalary
          sum=sumSalary mean=aveSalary;
run;

proc print data=work.means2;
run;
```

- The NOPRINT option suppresses the display of all output.

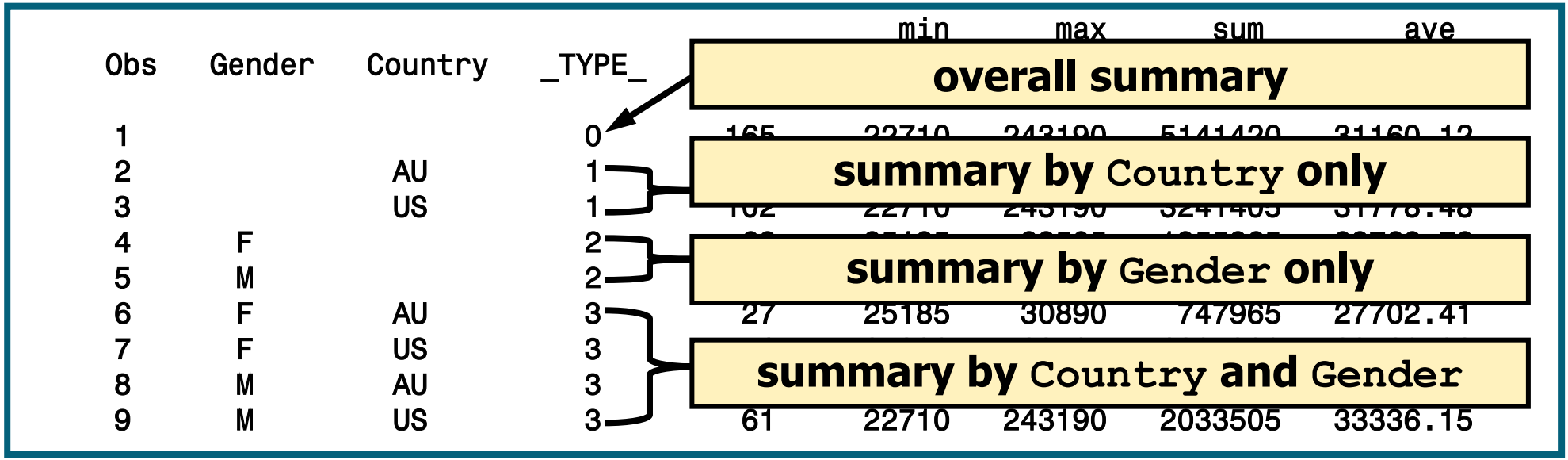
OUTPUT Statement OUT= Option

- PROC PRINT Output

Obs	Gender	Country	_TYPE_	_FREQ_	min Salary	max Salary	sum Salary	ave Salary
1			0	165	22710	243190	5141420	31160.12
2		AU	1	63	25185	108255	1900015	30158.97
3		US	1	102	22710	243190	3241405	31778.48
4	F		2	68	25185	83505	1955865	28762.72
5	M		2	97	22710	243190	3185555	32840.77
6	F	AU	3	27	25185	30890	747965	27702.41
7	F	US	3	41	25390	83505	1207900	29460.98
8	M	AU	3	36	25745	108255	1152050	32001.39
9	M	US	3	61	22710	243190	2033505	33336.15

OUTPUT Statement OUT= Option

- **_TYPE_** is a numeric variable that shows which combination of class variables produced the summary statistics in that observation.



OUTPUT Statement OUT= Option

Obs	Gender	Country	_TYPE_	_FREQ_	min Salary	max Salary	sum Salary	ave Salary
1			0	165	22710	243190	5141420	31160.12
2		AU	1	63	25185	108255	1900015	30158.97
3		US	1	102	22710	243190	3241405	31778.48
4	F		2	68	25185	83505	1955865	28762.72
5	M		2	97	22710	243190	3185555	32840.77
6	F	AU	3	27	25185	30890	747965	27702.41
7	F	US	3	41	25390	83505	1207900	29460.98
8	M	AU	3	36	25745	108255	1152050	32001.39
9	M	US	3	61	22710	243190	2033505	33336.15

TYPE	Type of Summary	_FREQ_
0	overall summary	165
1	summary by **Country** only	63 AU + 102 AU = 165
2	summary by **Gender** only	68 F + 97 M = 165
3	summary by **Country** and **Gender**	27 F AU + 41 F US + 36 M AU + 61 M US = 165

OUTPUT Statement OUT= Option

- Options can be added to the PROC MEANS statement to control the output data set.

Option	Description
NWAY	specifies that the output data set contain only statistics for the observations with the highest **_TYPE_** value.
DESCENDTYPES	orders the output data set by descending **_TYPE_** value.
CHARTYPE	specifies that the **_TYPE_** variable in the output data set is a character representation of the binary value of **_TYPE_**.

OUTPUT Statement OUT= Option

without options

Obs	Gender	Country	_TYPE_	_FREQ_	min Salary	max Salary	sum Salary	ave Salary
1			0	165	22710	243190	5141420	31160.12
2		AU	1	63	25185	108255	1900015	30158.97
3		US	1	102	22710	243190	3241405	31778.48
4	F		2	68	25185	83505	1955865	28762.72
5	M		2	97	22710	243190	3185555	32840.77
6	F	AU	3	27	25185	30890	747965	27702.41
7	F	US	3	41	25390	83505	1207900	29460.98
8	M	AU	3	36	25745	108255	1152050	32001.39
9	M	US	3	61	22710	243190	2033505	33336.15

with NWAY

Obs	Gender	Country	_TYPE_	_FREQ_	min Salary	max Salary	sum Salary	ave Salary
1	F	AU	3	27	25185	30890	747965	27702.41
2	F	US	3	41	25390	83505	1207900	29460.98
3	M	AU	3	36	25745	108255	1152050	32001.39
4	M	US	3	61	22710	243190	2033505	33336.15

p112d06

OUTPUT Statement OUT= Option

with DESCENDTYPES

Obs	Gender	Country	_TYPE_	_FREQ_	min Salary	max Salary	sum Salary	ave Salary
1	F	AU	3	27	25185	30890	747965	27702.41
2	F	US	3	41	25390	83505	1207900	29460.98
3	M	AU	3	36	25745	108255	1152050	32001.39
4	M	US	3	61	22710	243190	2033505	33336.15
5	F		2	68	25185	83505	1955865	28762.72
6	M		2	97	22710	243190	3185555	32840.77
7		AU	1	63	25185	108255	1900015	30158.97
8		US	1	102	22710	243190	3241405	31778.48
9			0	165	22710	243190	5141420	31160.12

OUTPUT Statement OUT= Option

with CHARTYPE

Obs	Gender	Country	_TYPE_	_FREQ_	min Salary	max Salary	sum Salary	ave Salary
1			00	165	22710	243190	5141420	31160.12
2		AU	01	63	25185	108255	1900015	30158.97
3		US	01	102	22710	243190	3241405	31778.48
4	F		10	68	25185	83505	1955865	28762.72
5	M		10	97	22710	243190	3185555	32840.77
6	F	AU	11	27	25185	30890	747965	27702.41
7	F	US	11	41	25390	83505	1207900	29460.98
8	M	AU	11	36	25745	108255	1152050	32001.39
9	M	US	11	61	22710	243190	2033505	33336.15

The SUMMARY Procedure

- The SUMMARY procedure provides data summarization tools to compute descriptive statistics for variables across all observations and within groups of observations.

General form of the SUMMARY procedure:

```
PROC SUMMARY DATA=SAS-data-set <statistic(s)>
                                <option(s)>;
     VAR analysis-variable(s);
     CLASS classification-variable(s);
RUN;
```

The SUMMARY Procedure

- The SUMMARY procedure uses the same syntax as the MEANS procedure.
- The only differences to the two procedures are the following:

PROC MEANS	PROC SUMMARY
The PRINT option is set by default, which displays output.	The NOPRINT option is set by default, which displays no output.
Omitting the VAR statement analyzes all the numeric variables.	Omitting the VAR statement produces a simple count of observations.

The TABULATE Procedure

- The TABULATE procedure displays descriptive statistics in tabular format.

General form of the TABULATE procedure:

```
PROC TABULATE DATA=SAS-data-set <options>;
      CLASS classification-variable(s);
      VAR analysis-variable(s);
      TABLE  page-expression,
                  row-expression,
                  column-expression </ option(s)>;
RUN;
```

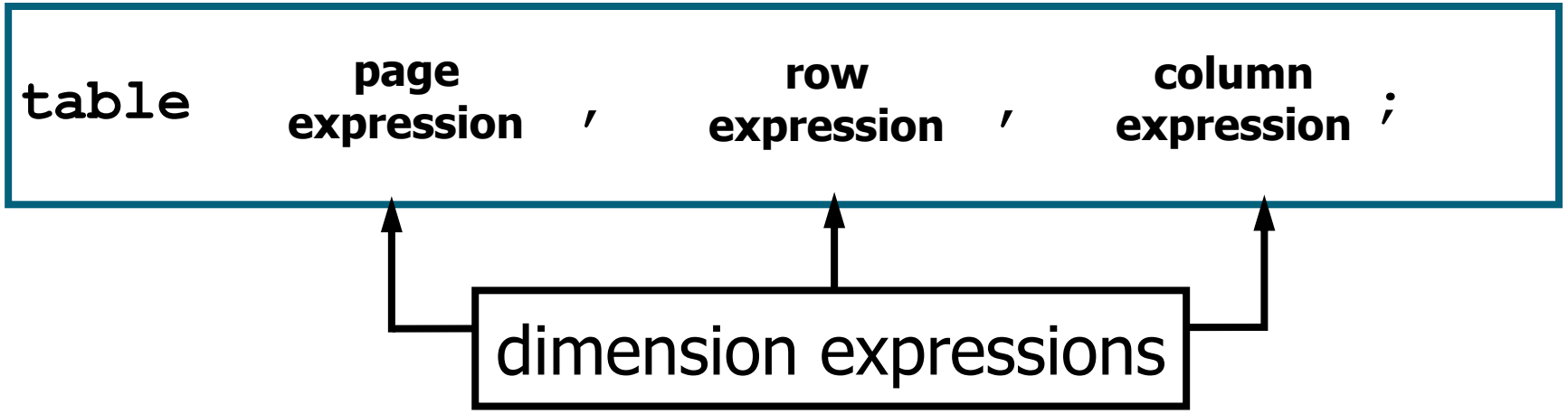
Dimensional Tables

- The TABULATE procedure produces one-, two-, or three-dimensional tables.

	page dimension	row dimension	column dimension
one-dimensional			✓
two-dimensional		✓	✓
three-dimensional	✓	✓	✓

The TABLE Statement

- The TABLE statement describes the structure of the table.



- Commas separate the dimension expressions.
- Every variable that is part of a dimension expression must be specified as a classification variable (CLASS statement) or an analysis variable (VAR statement).

The TABLE Statement

```
table  page expression , row expression , column expression ;
```

- Examples:

```
table Country;
```

```
table Gender , Country;
```

```
table Job_Title , Gender , Country;
```

The CLASS Statement

- The CLASS statement identifies variables to be used as classification, or grouping, variables.
- General form of the CLASS statement:

CLASS *classification-variable(s)*;

- N, the number of nonmissing values, is the default statistic for classification variables.
- Examples of classification variables:

  `Job_Title`, **`Gender`**, and **`Country`**

The VAR Statement

- The VAR statement identifies the numeric variables for which statistics are calculated.
- General form of the VAR statement:

VAR *analysis-variable(s)*;

- SUM is the default statistic for analysis variables.
- Examples of analysis variables:

  `Salary` and **`Bonus`**

One-Dimensional Table

```
proc tabulate data=orion.sales;
   class Country;
   table Country;
run;
```

Country	
AU	US
N	N
63.00	102.00

Two-Dimensional Table

```
proc tabulate data=orion.sales;
   class Gender Country;
   table Gender, Country;
run;
```

	Country	
	AU	US
	N	N
Gender		
F	27.00	41.00
M	36.00	61.00

Three-Dimensional Table

```
proc tabulate data=orion.sales;
   class Job_Title Gender Country;
   table Job_Title, Gender, Country;
run;
```

Job_Title Sales Rep. I

	Country	
	AU	US
Gender		
F		
M		

Job_Title Sales Rep. II

	Country	
	AU	US
	N	N
Gender		
F	10.00	14.00
M	8.00	14.00

p112d08

Dimension Expression

- Elements that can be used in a dimension expression:
  - classification variables
  - analysis variables
  - the universal class variable ALL
  - keywords for statistics

- Operators that can be used in a dimension expression:
  - blank, which concatenates table information
  - asterisk *, which crosses table information
  - parentheses (), which group elements

Dimension Expression

```
proc tabulate data=orion.sales;
   class Gender Country;
   var Salary;
   table Gender all, Country*Salary;
run;
```

	Country	
	AU	US
	Salary	Salary
	Sum	Sum
Gender		
F	747965.00	1207900.00
M	1152050.00	2033505.00
All	1900015.00	3241405.00

PROC TABULATE Statistics

Descriptive Statistic Keywords				
	CSS	CV	LCLM	MAX
MEAN	MIN	MODE	N	NMISS
KURTOSIS	RANGE	SKEWNESS	STDDEV	STDERR
SUM	SUMWGT	UCLM	USS	VAR
PCTN	REPPCTN	PAGEPCTN	ROWPCTN	COLPCTN
PCTSUM	REPPCTSUM	PAGEPCTSUM	ROWPCTSUM	COLPCTSUM

Quantile Statistic Keywords				
MEDIAN | P50	P1	P5	P10	Q1 | P25
Q3 | P75	P90	P95	P99	QRANGE

Hypothesis Testing Keywords				
PROBT	T			

PROC TABULATE Statistics

```
proc tabulate data=orion.sales;
   class Gender Country;
   var Salary;
   table Gender all, Country*Salary*(min max);
run;
```

	Country			
	AU		US	
	Salary		Salary	
	Min	Max	Min	Max
Gender				
F	25185.00	30890.00	25390.00	83505.00
M	25745.00	108255.00	22710.00	243190.00
All	25185.00	108255.00	22710.00	243190.00

Additional SAS Statements

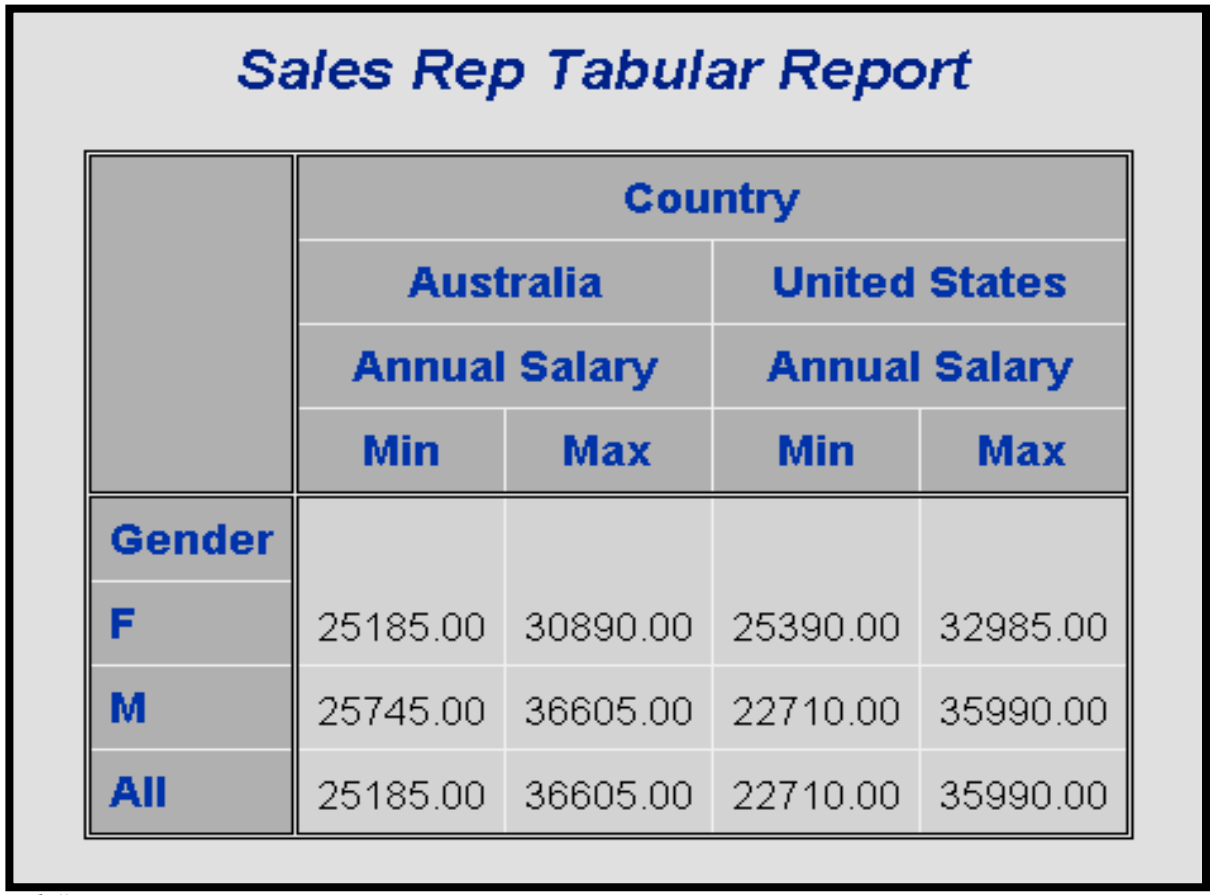
- Additional statements can be added to enhance the report.

```
proc format;
   value $ctryfmt 'AU'='Australia'
                  'US'='United States';
run;

options nodate pageno=1;

ods html file='p112d08.html';
proc tabulate data=orion.sales;
   class Gender Country;
   var Salary;
   table Gender all, Country*Salary*(min max);
   where Job_Title contains 'Rep';
   label Salary='Annual Salary';
   format Country $ctryfmt.;
   title 'Sales Rep Tabular Report';
run;
ods html close;
```

p112d08

Additional SAS Statements

- HTML Output



Sales Rep Tabular Report

	Country			
	Australia		United States	
	Annual Salary		Annual Salary	
	Min	Max	Min	Max
Gender				
F	25185.00	30890.00	25390.00	32985.00
M	25745.00	36605.00	22710.00	35990.00
All	25185.00	36605.00	22710.00	35990.00

Output Data Sets

- PROC TABULATE produces output data sets using the following method:

```
PROC TABULATE DATA=SAS-data-set
                  OUT=SAS-data-set <options>;
```

- The output data set contains the following variables:
  - BY variables
  - class variables
  - the automatic variables **_TYPE_**, **_PAGE_**, and **_TABLE_**
  - calculated statistics

PROC Statement OUT= Option

```
proc tabulate data=orion.sales
              out=work.tabulate;
   where Job_Title contains 'Rep';
   class Job_Title Gender Country;
   table Country;
   table Gender, Country;
   table Job_Title, Gender, Country;
run;

proc print data=work.tabulate;
run;
```

p112d09

PROC Statement OUT= Option

- Partial PROC PRINT Output

Obs	Job_Title	Gender	Country	_TYPE_	_PAGE_	_TABLE_	N
1			AU	001	1	1	61
2			US	001	1	1	98
3		F	AU	011	1	2	27
4		F	US	011	1	2	40
5		M	AU	011	1	2	34
6		M	US	011	1	2	58
7	Sales Rep. I	F	AU	111	1	3	8
8	Sales Rep. I	F	US	111	1	3	13
9	Sales Rep. I	M	AU	111	1	3	13
10	Sales Rep. I	M	US	111	1	3	29
11	Sales Rep. II	F	AU	111	2	3	10
12	Sales Rep. II	F	US	111	2	3	14
13	Sales Rep. II	M	AU	111	2	3	8
14	Sales Rep. II	M	US	111	2	3	14
15	Sales Rep. III	F	AU	111	3	3	7
16	Sales Rep. III	F	US	111	3	3	8
17	Sales Rep. III	M	AU	111	3	3	10
18	Sales Rep. III	M	US	111	3	3	9

PROC Statement OUT= Option

- **_TYPE_** is a character variable that shows which combination of class variables produced the summary statistics in that observation.

- Partial PROC PRINT Output

Obs	Job_Title	Gender	Country	_TYPE_	_PAGE_	_TABLE_	N
1			AU	001	1	1	61
2			US	001	1	1	98
3		F	AU	011	1	2	27
4		F	US	011			
5		M	AU	011			
6		M	US	011			

0 for **Job_Title**, 1 for **Gender**, and 1 for **Country**

PROC Statement OUT= Option

- **_PAGE_** is a numeric variable that shows the logical page number that contains that observation.
- Partial PROC PRINT Output

Obs	Job_Title	Gender	Country	_TYPE_	_PAGE_	_TABLE_	N
7	Sales Rep. I	F	AU	111	1		
8	Sales Rep. I	F	US	111	1		
9	Sales Rep. I	M	AU	111	1		
10	Sales Rep. I	M	US	111	1	3	29
11	Sales Rep. II	F	AU	111	2		
12	Sales Rep. II	F	US	111	2		
13	Sales Rep. II	M	AU	111	2		
14	Sales Rep. II	M	US	111	2	3	14
15	Sales Rep. III	F	AU	111	3		
16	Sales Rep. III	F	US	111	3		
17	Sales Rep. III	M	AU	111	3		
18	Sales Rep. III	M	US	111	3		

Page 1 for Sales Rep. I

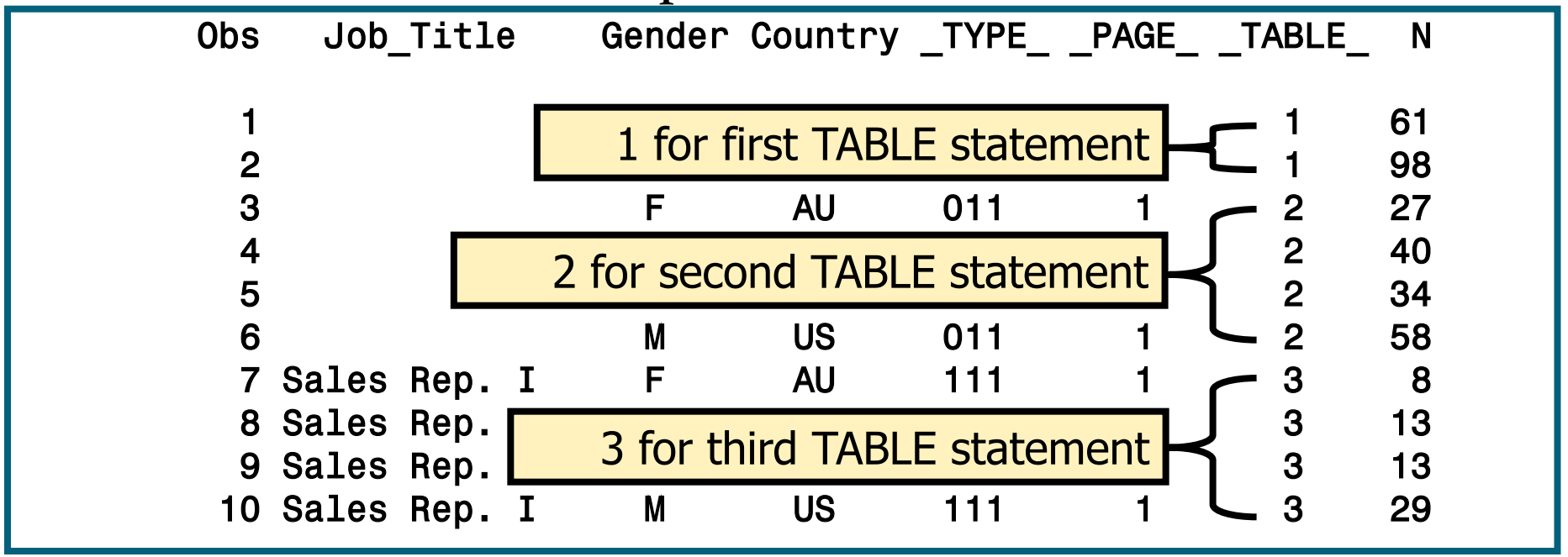
Page 2 for Sales Rep. II

Page 3 for Sales Rep. III

PROC Statement OUT= Option

- **_TABLE_** is a numeric variable that shows the number of the TABLE statement that contains that observation.

- Partial PROC PRINT Output



Více o PROC TABULATE:

- In the SUGI 28 proceedings:
  - *"The Simplicity and Power of the TABULATE Procedure"*, by Dan Bruns

  http://www2.sas.com/proceedings/sugi28/197-28.pdf
- Online (from the SUGI 27 proceedings):
  - *"Anyone Can Learn PROC TABULATE"*, by Lauren Haworth, http://www2.sas.com/proceedings/sugi27/p060-27.pdf

The UNIVARIATE Procedure

- The UNIVARIATE procedure produces summary reports that display descriptive statistics.

- General form of the UNIVARIATE procedure:

```
PROC UNIVARIATE DATA=SAS-data-set;
     VAR variable(s);
RUN;
```

- The VAR statement specifies the analysis variables and their order in the results.

The UNIVARIATE Procedure

The following PROC UNIVARIATE step shows default descriptive statistics for **Salary**.

```
proc univariate data=orion.nonsales;
   var Salary;
run;
```

- Without the VAR statement, SAS will analyze all numeric variables.

The UNIVARIATE Procedure

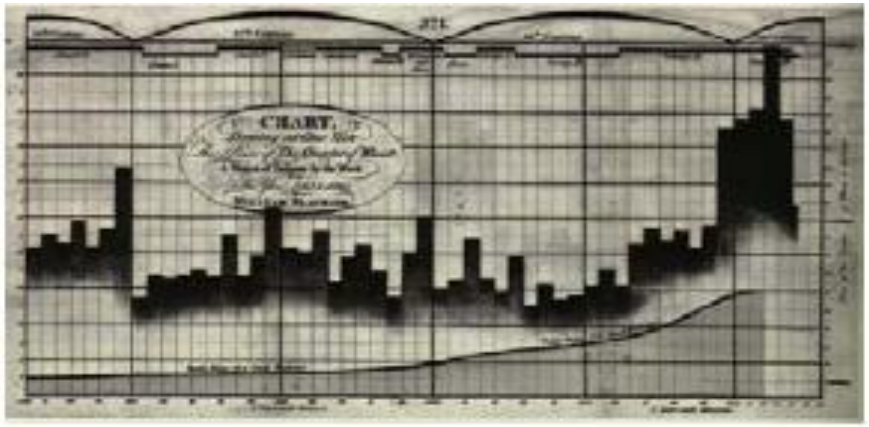
- The UNIVARIATE procedure can produce the following sections of output:
  - Moments
  - Basic Statistical Measures
  - Tests for Locations
  - Quantiles
  - Extreme Observations
  - Missing Values

Vizualizace – zdroje

- Na prvním místě se obvykle citují knihy prof. Tufteho, např. Tufte E.R. (1983) The Visual Display of Quantitative Information, Graphic Press, Chesire, Conn.
- Weby o vizualizaci, např.
  - http://www.math.yorku.ca/SCS/Gallery/noframes.html - galerie s poučným výkladem a příklady i nezdařených či lživých grafů
  - http://www.agocg.ac.uk/ - John Lansdown (1992) Aspects of Design in Computer Graphics: Some Notes – http://www.agocg.ac.uk/train/hitch/hitch.htm
- Jiné weby, např. stránky různých vizualizačních programů a organizací
  - http://www.cybergeography.org/atlas/atlas.html nebo http://miner3d.com/products/gallery.html

Vizualizace – historie

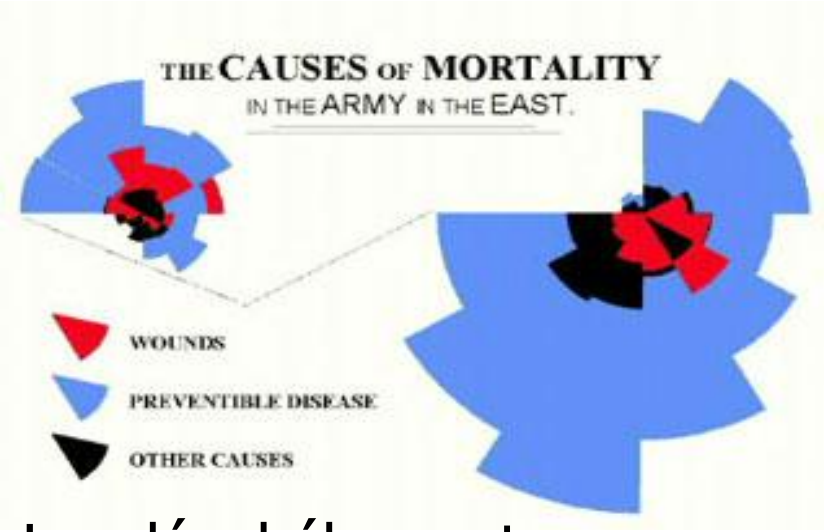
- William Playfair, 1786: první publikovaná prezentační grafika



- Dr. John Snow, 1845: epidemie cholery v Londýně

Vizualizace – historie

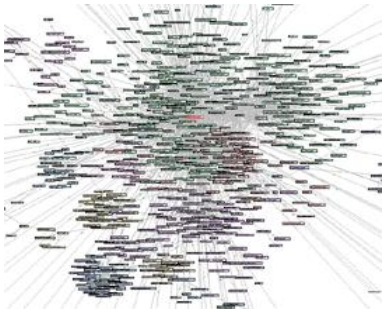
- Florence Nightingale, 1858: důvody úmrtí v průběhu Krymské války (1853-1856)



- Harry Beck, 1931: schéma Londýnského metra

Vizualizace – investigativní analýza

- http://www.i2inc.com/



Law Enforcement

» Counterterrorism
» Narcotics investigations
» Organized crime
» Intelligence analysis
» Fraud
» Missing persons
» Major investigations
» Counterfeiting
» Immigration control
» Major event security
» Money laundering
» Gang investigations

Government

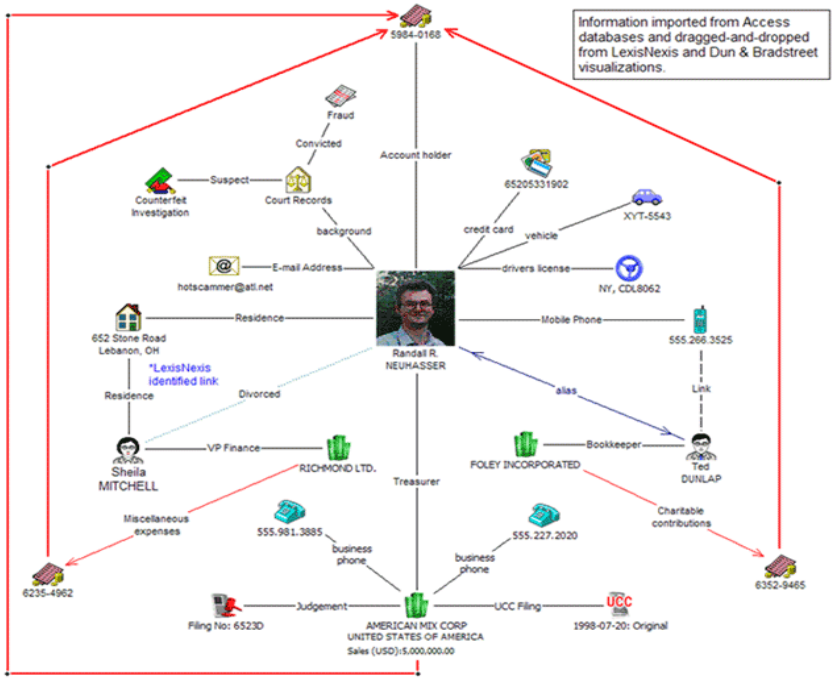
» Criminal prosecutions
» National security
» Military intelligence
» Embassy security
» Postal inspection and fraud
» Prison investigations
» Park and wildlife services
» Antitrust investigations
» Tax fraud investigations
» Customs investigations

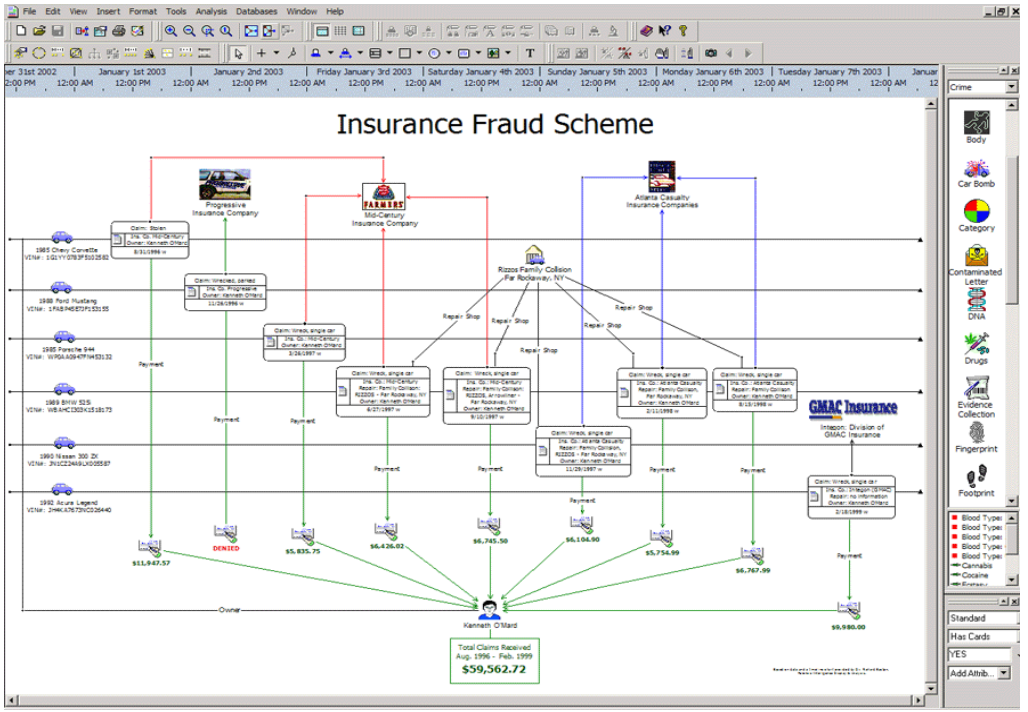
Commercial

» Forensic accounting
» Money laundering
» Insider trading violations
» Corporate security
» Anti-pirating investigations
» Entertainment copyright violations
» Competitive intelligence
» Civil lawsuits
» Fraud:
  » Credit card
  » Insurance
  » Retail
  » Health care
  » Commercial
  » Telephone

Vizualizace – investigativní analýza

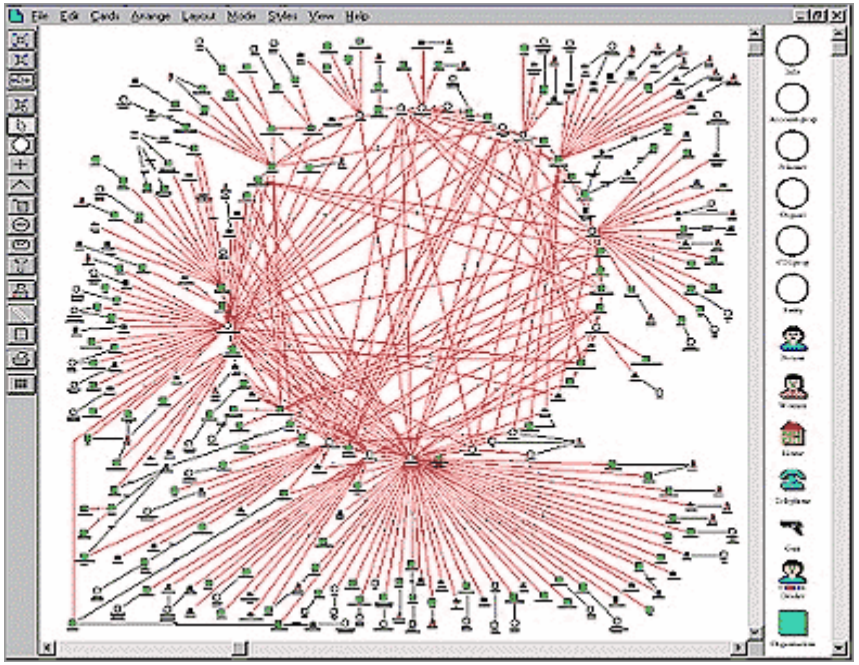
- osobní kontakty, pojistné podvody

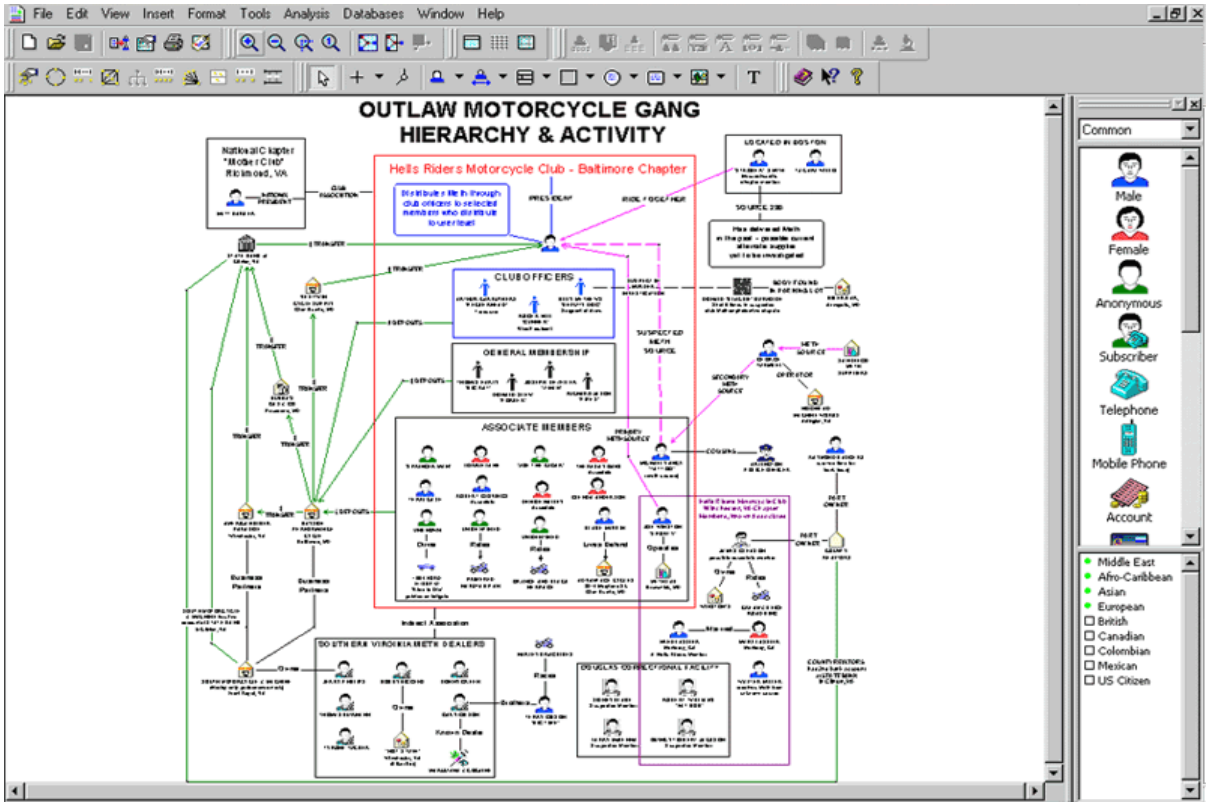




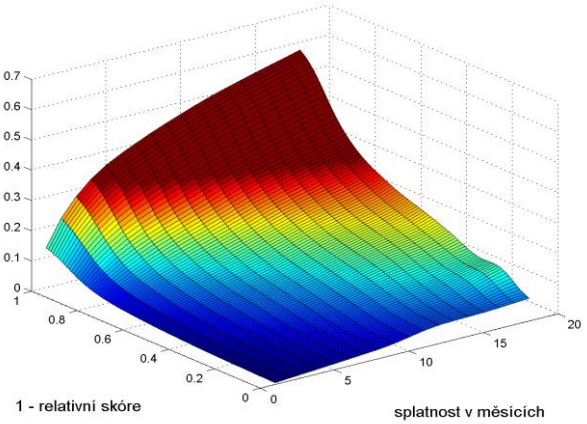
Vizualizace – investigativní analýza

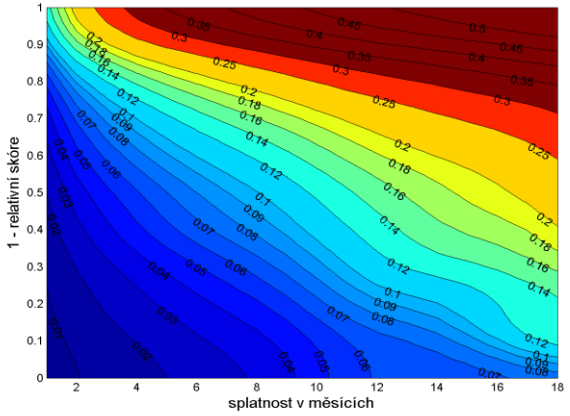
- Praní špinavých peněz, kriminální gangy

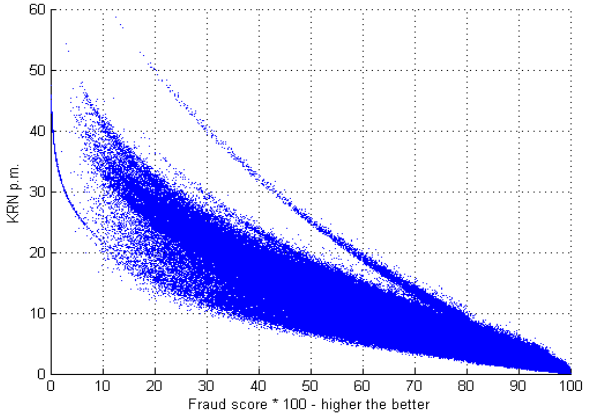


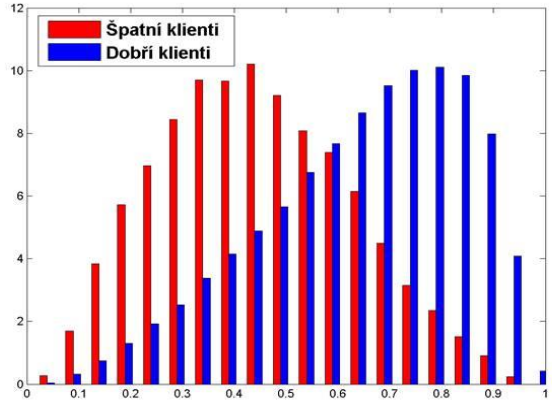


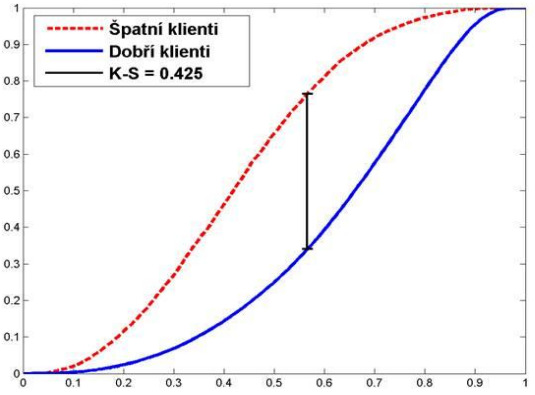
Vizualizace – risk management

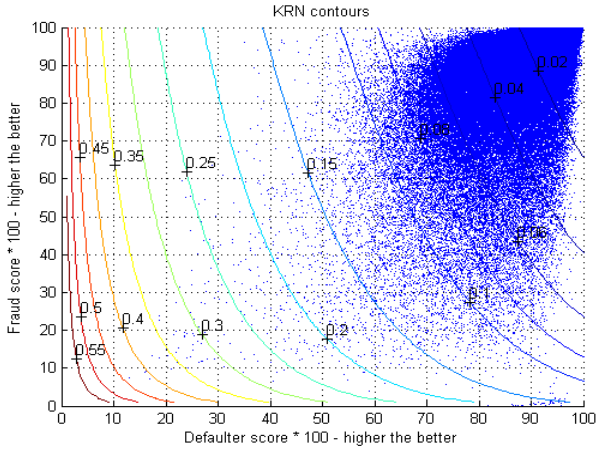




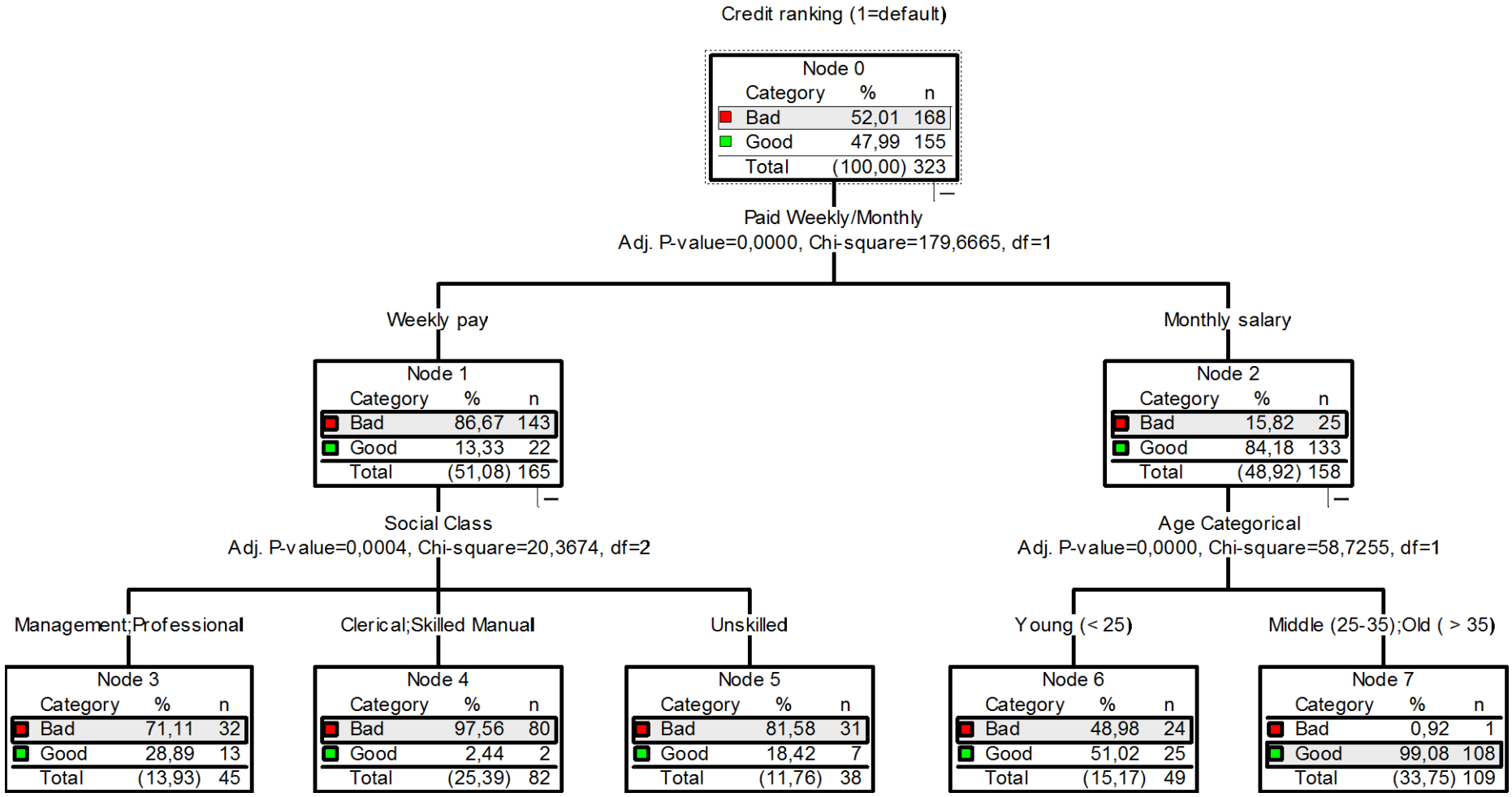




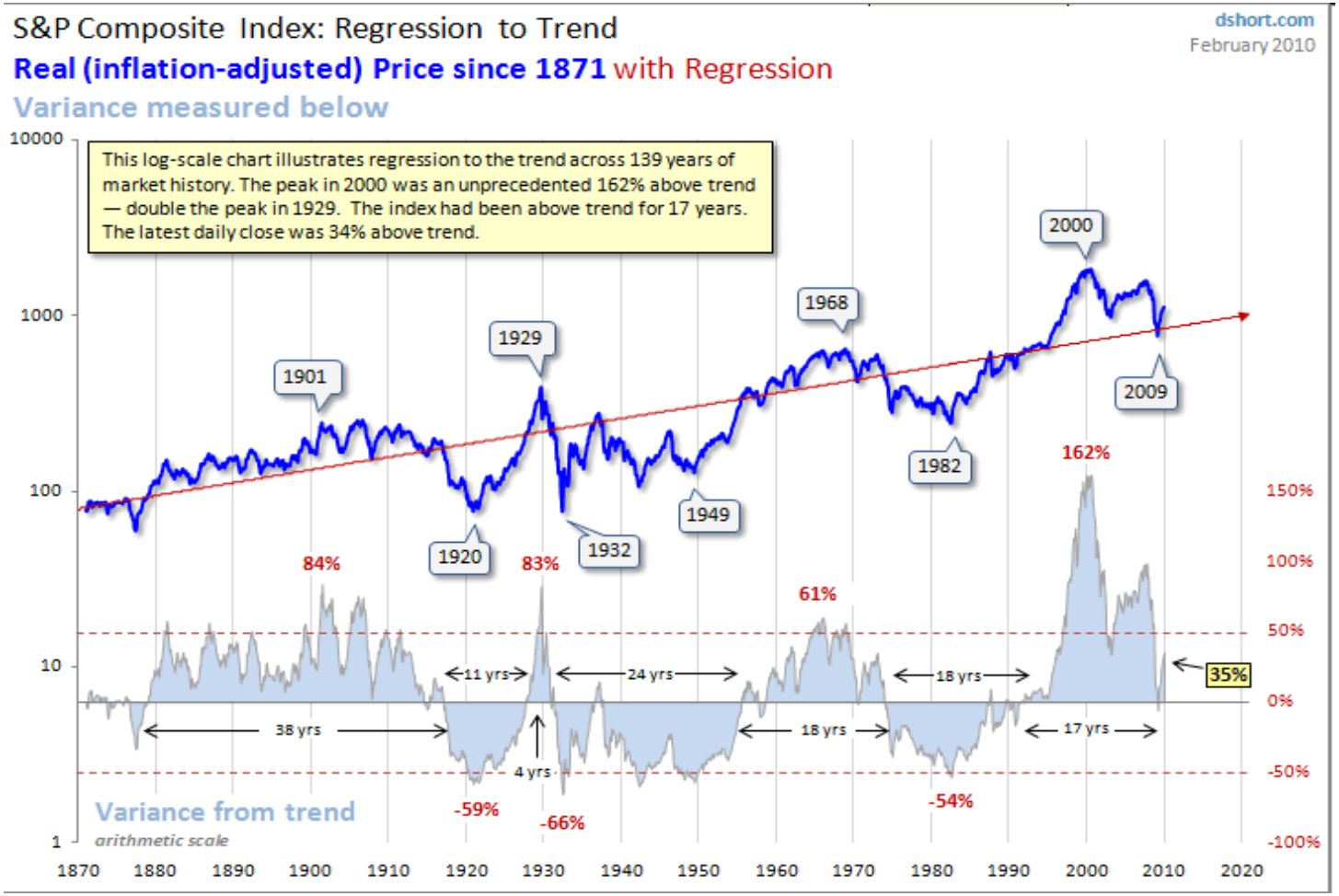




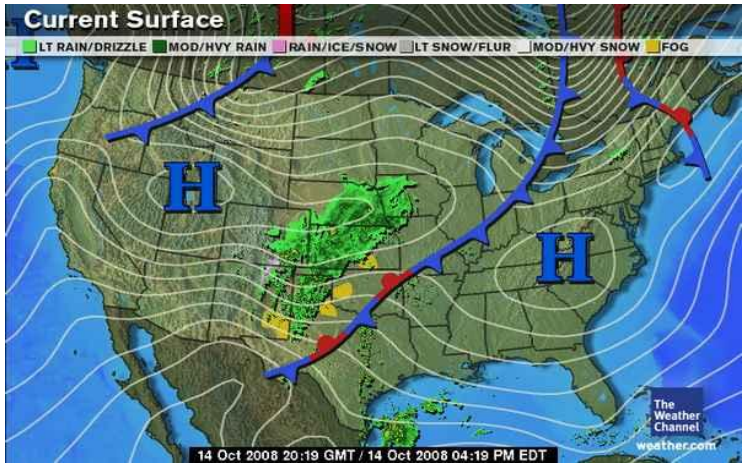
Vizualizace - dendrogram

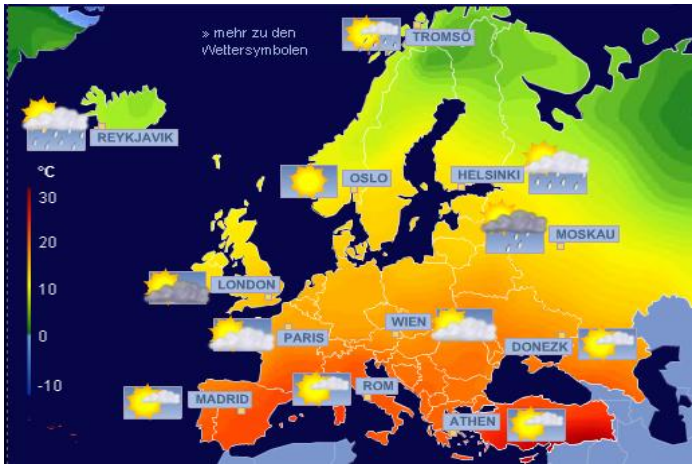


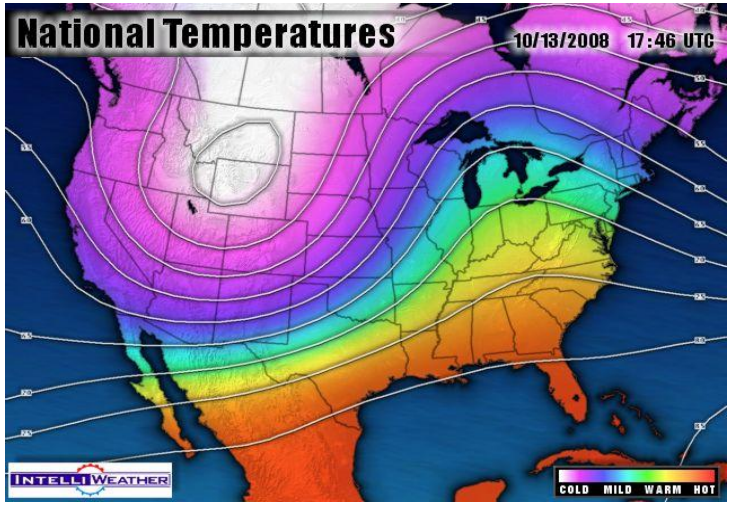
Vizualizace – ekonomie

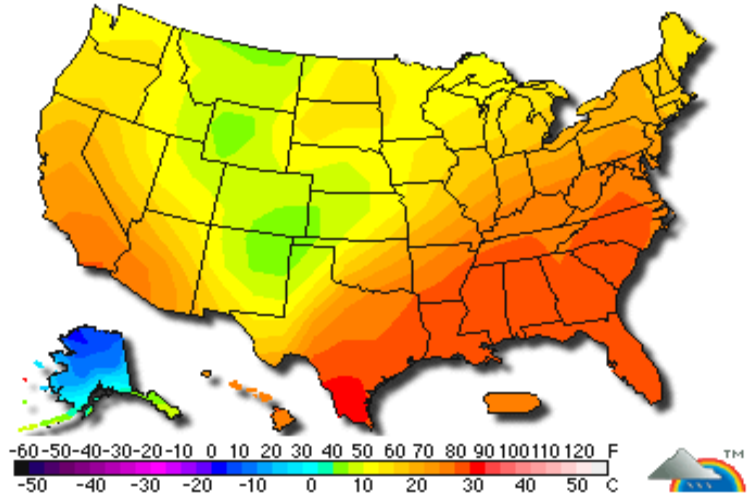


Meteo-vizualizace



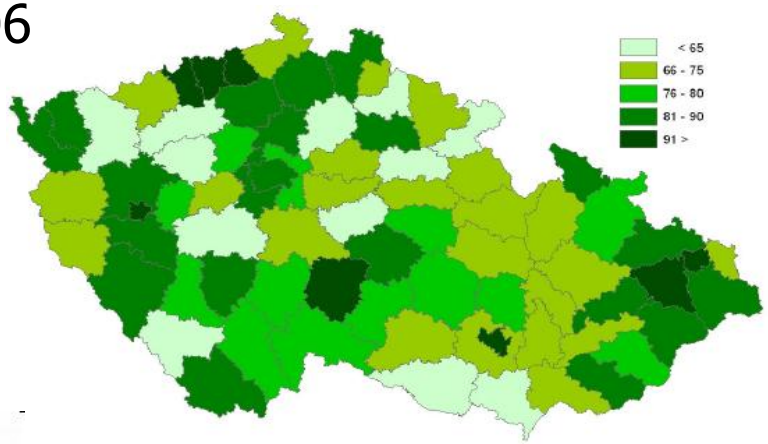


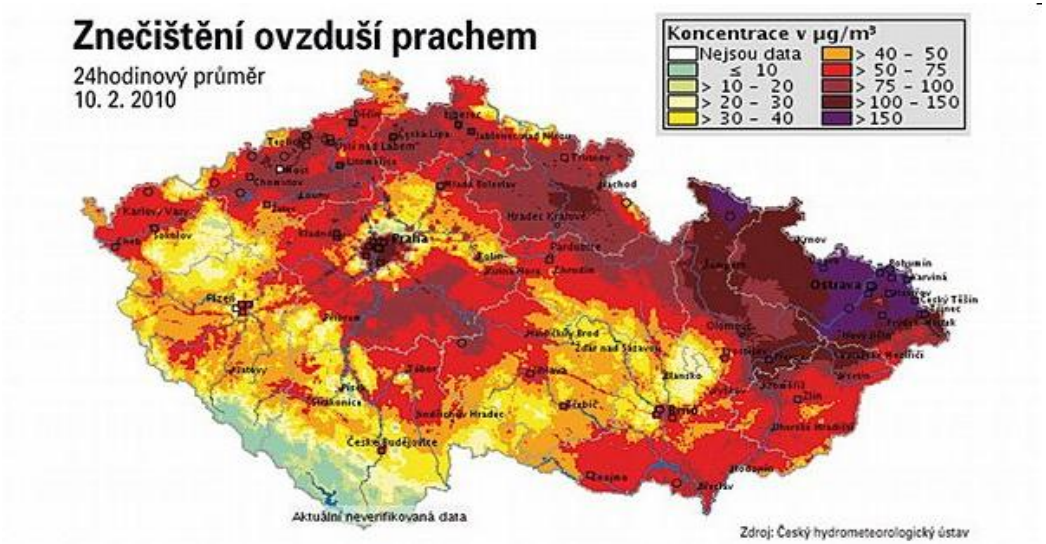




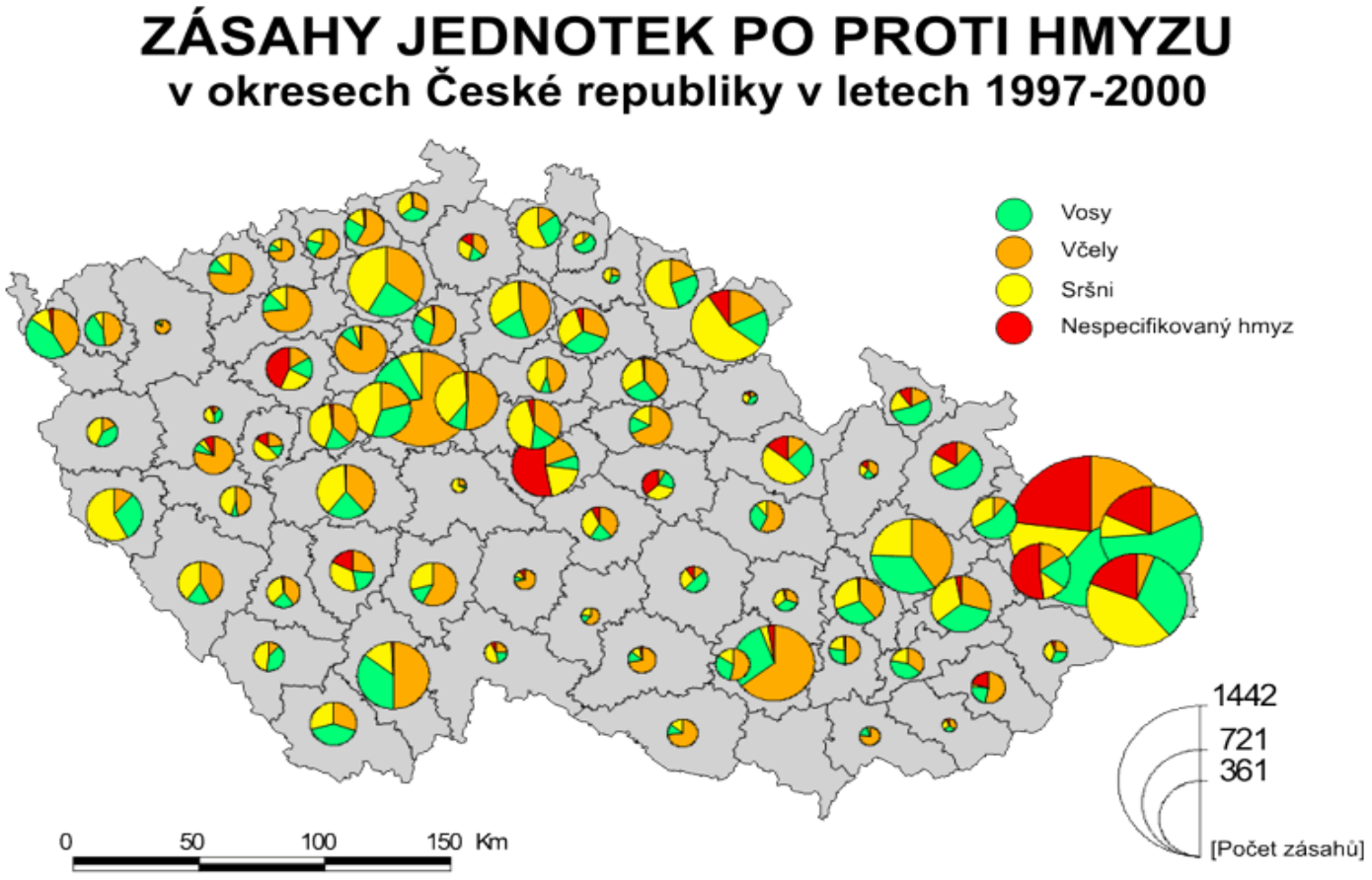
Kartogram

- Obce s počtem 500 a více obyvatel s vysokorychlostním připojením k internetu, podle okresů (%), k 31.12.2006

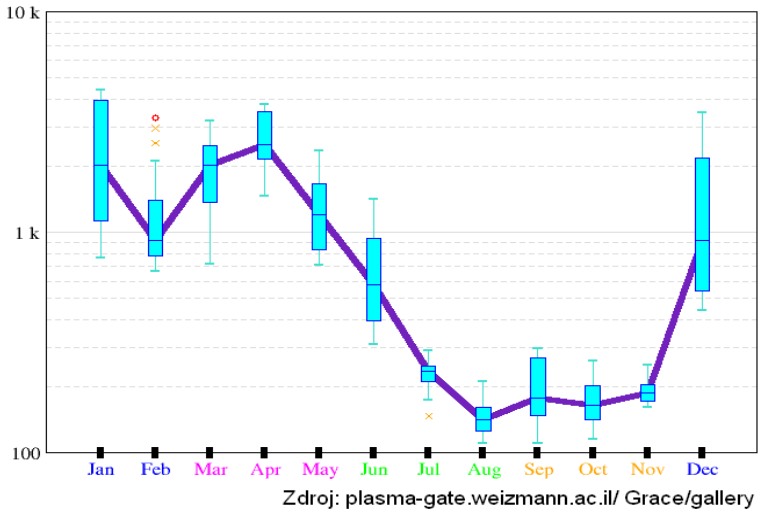




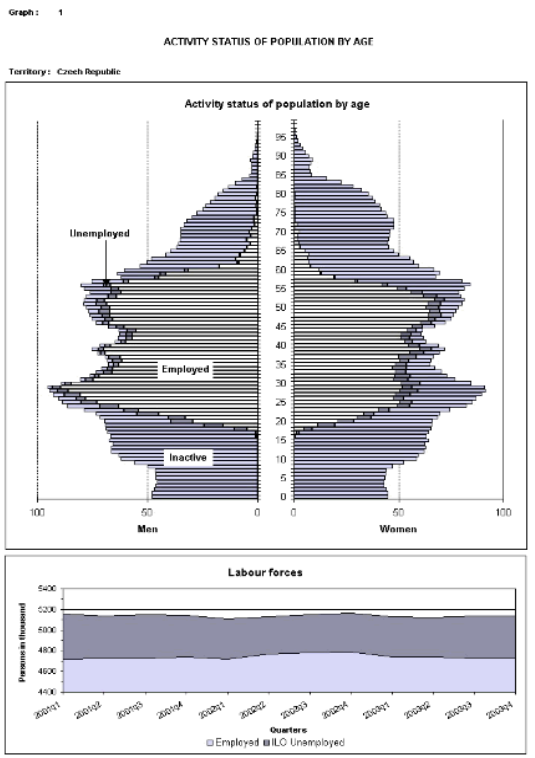
Kartodiagram

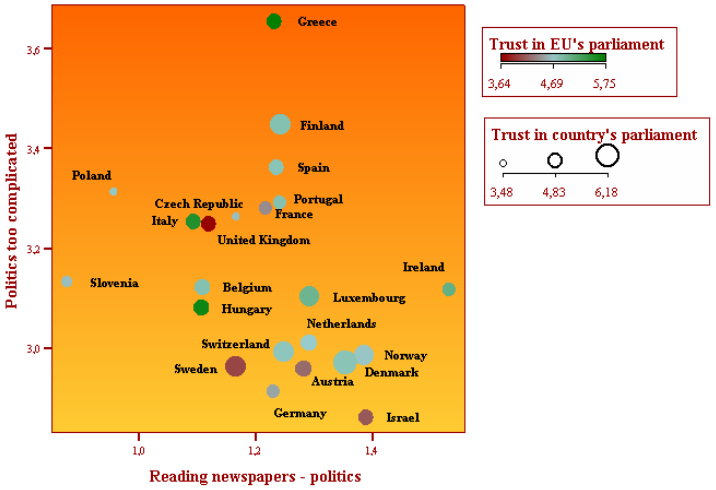


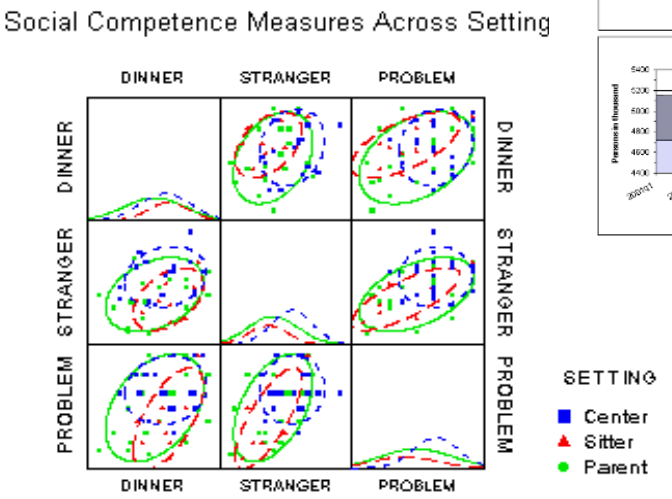
Grafy –další typy



Zdroj: plasma-gate.weizmann.ac.il/ Grace/gallery

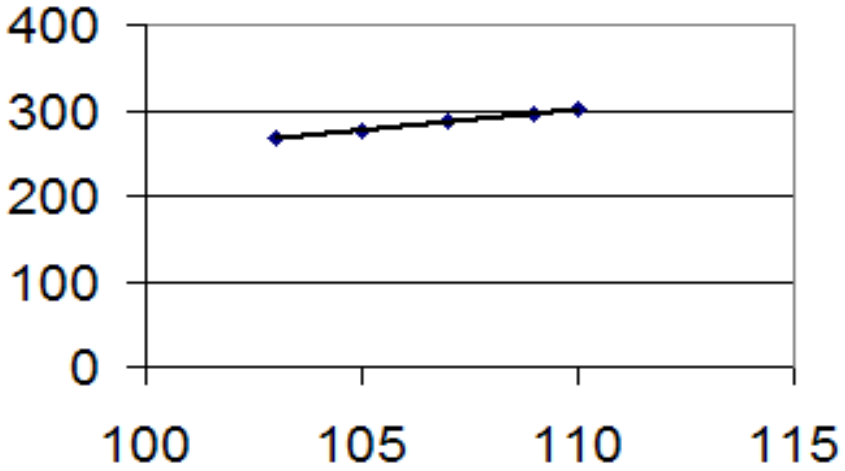


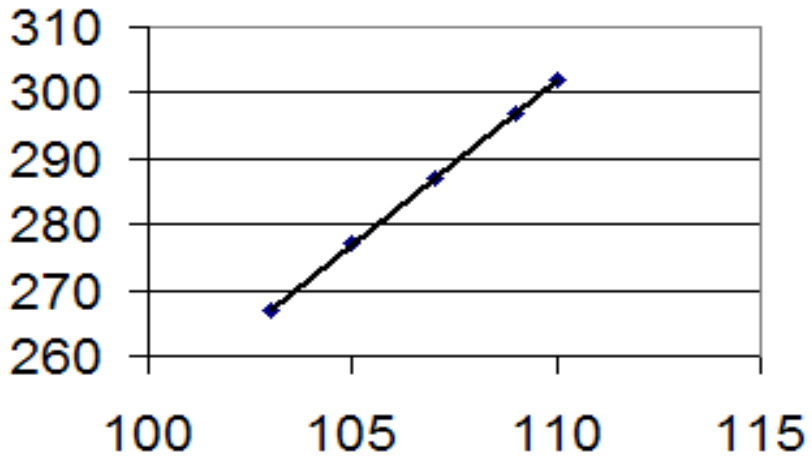




Měřítko grafu

- Která přímka roste strměji?





x	y
103	567
105	577
107	587
109	597
110	602

Měřítko grafu

- Pohled tvůrce grafu:
  - Zvýraznění trendu – pozitivní výsledky.
  - Potlačení trendu – negativní výsledky.

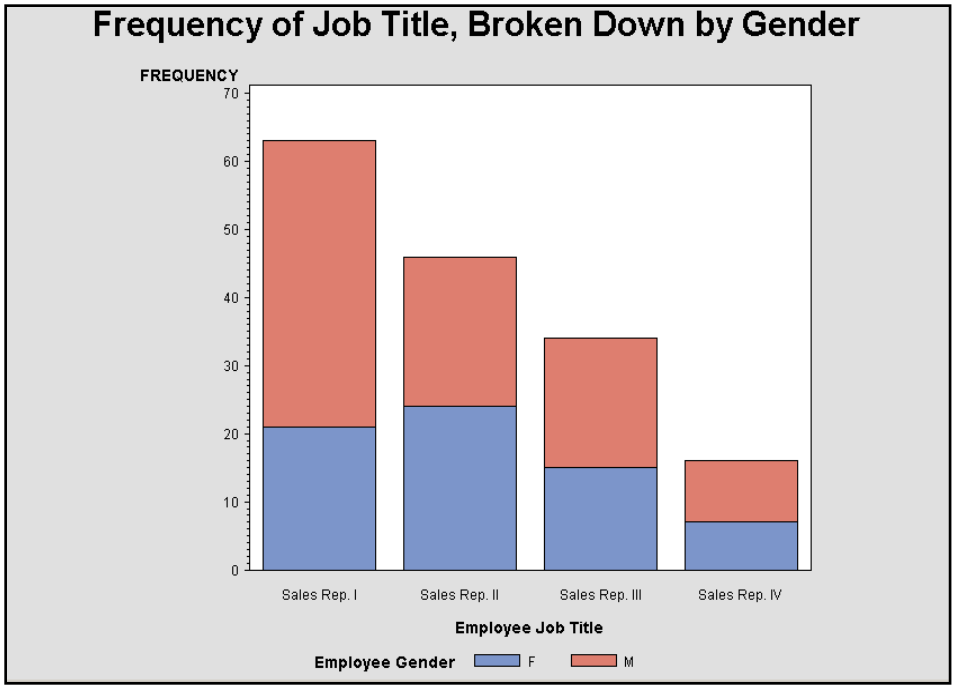
- Pohled uživatele grafu:
  - Grafy bez uvedeného měřítka jsou silně podezřelé.
  - Nepodléhat podsouvané informaci o růstu/poklesu.

What Is SAS/GRAPH Software?

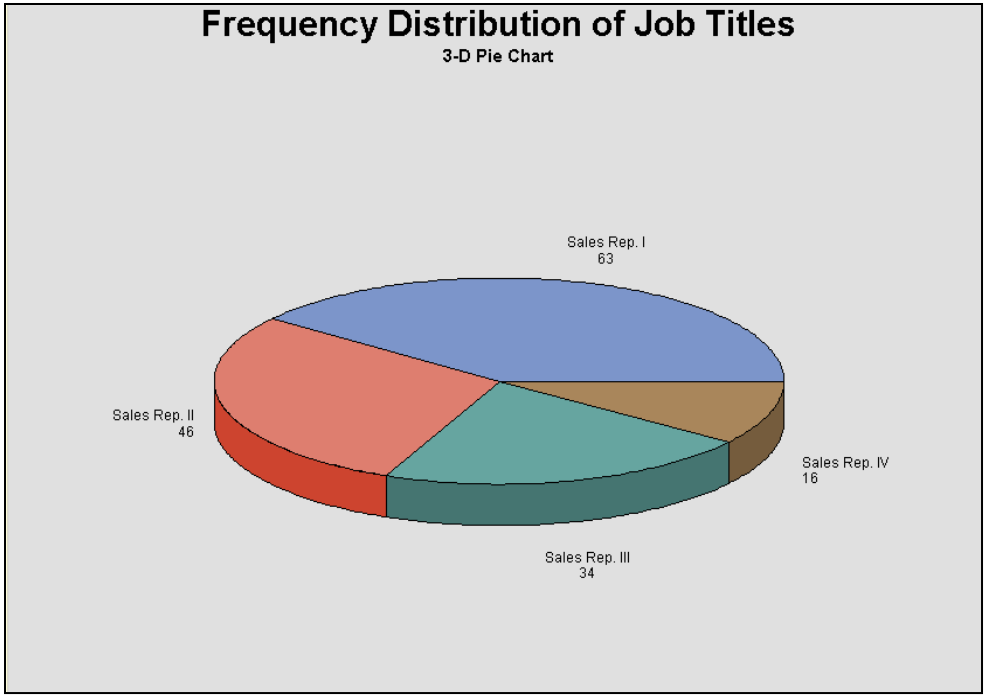
- *SAS/GRAPH software* is a component of SAS software that enables you to create the following types of graphs:
  - bar, block, and pie charts
  - two-dimensional scatter plots and line plots
  - three-dimensional scatter and surface plots
  - contour plots
  - maps
  - text slides
  - custom graphs

Základní typy grafů

- Bar Charts (GCHART Procedure)

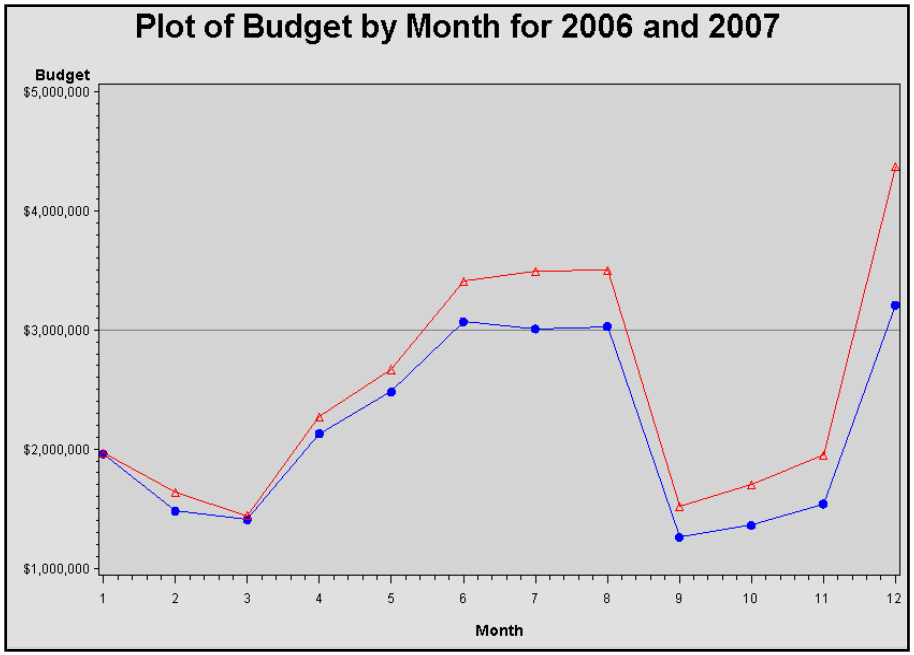


- Pie Charts (GCHART Procedure)

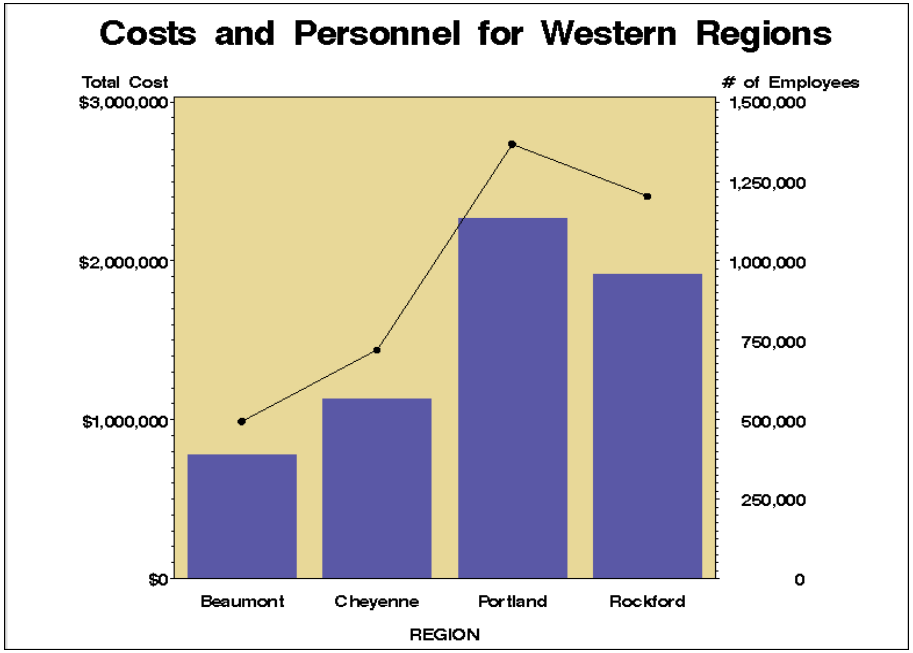


Základní typy grafů

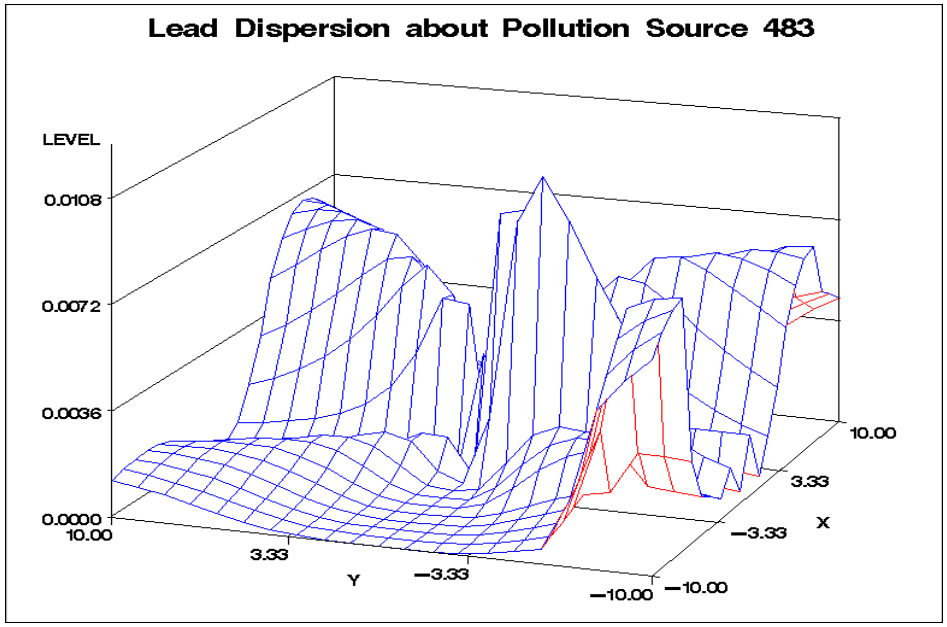
- Scatter and Line Plots (GPLOT Procedure)

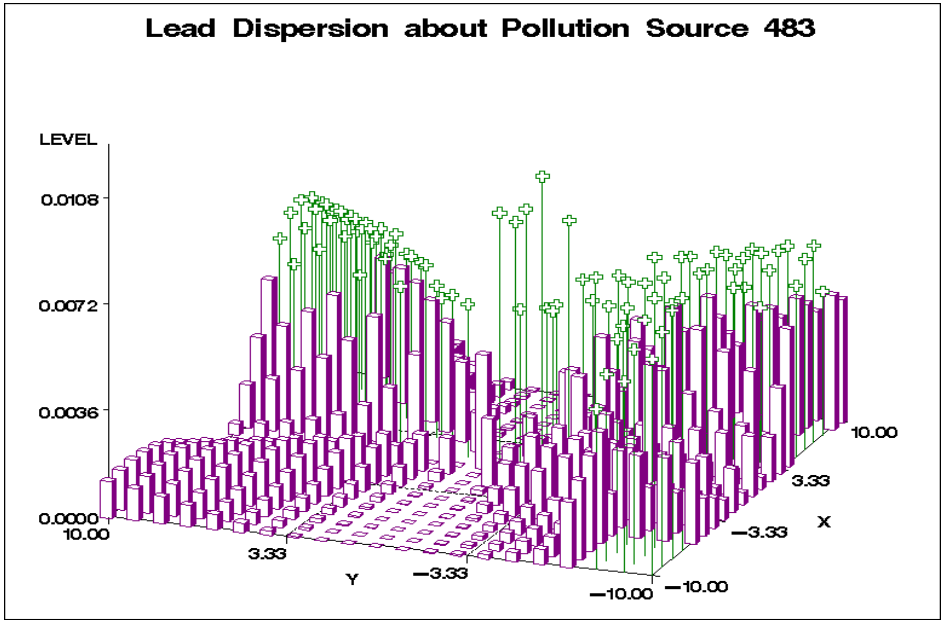


- Bar Charts with Line Plot Overlay (GBARLINE Procedure)



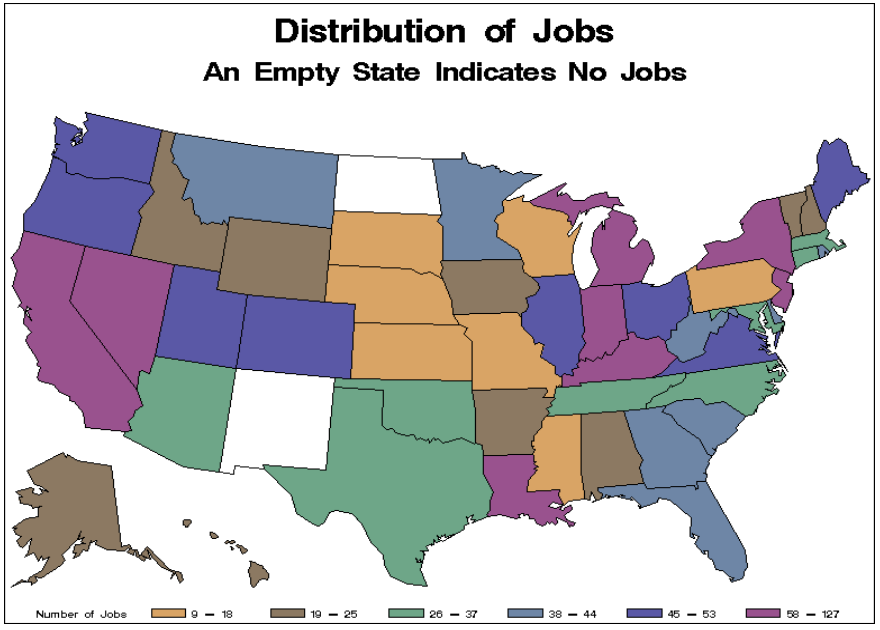
Three-Dimensional Surface and Scatter Plots (G3D Procedure)



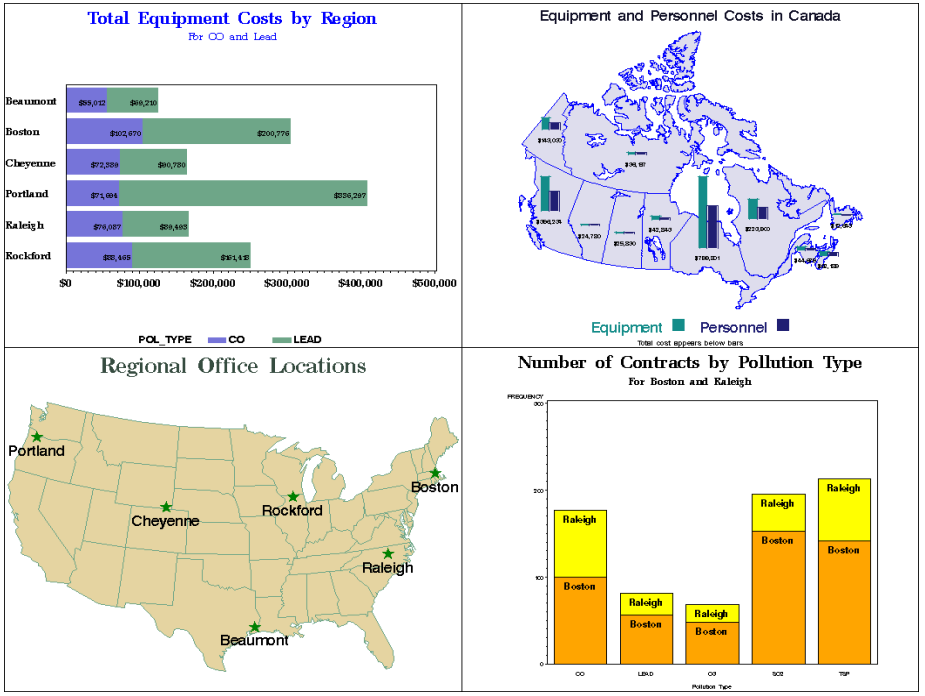


Maps (GMAP Procedure)

- Maps (GMAP Procedure)



- Multiple graphs on a page (GREPLAY Procedure)



Producing Bar and Pie Charts with the GCHART Procedure

- General form of the PROC GCHART statement:

PROC GCHART DATA=*SAS-data-set*;

- Use one of these statements to specify the chart type:

HBAR *chart-variable* . . . </ *options*>;
HBAR3D *chart-variable* . . . </ *options*>;

VBAR *chart-variable* . . . </ *options*>;
VBAR3D *chart-variable* . . . </ *options*>;

PIE *chart-variable* . . . </ *options*>;
PIE3D *chart-variable* . . . </ *options*>;

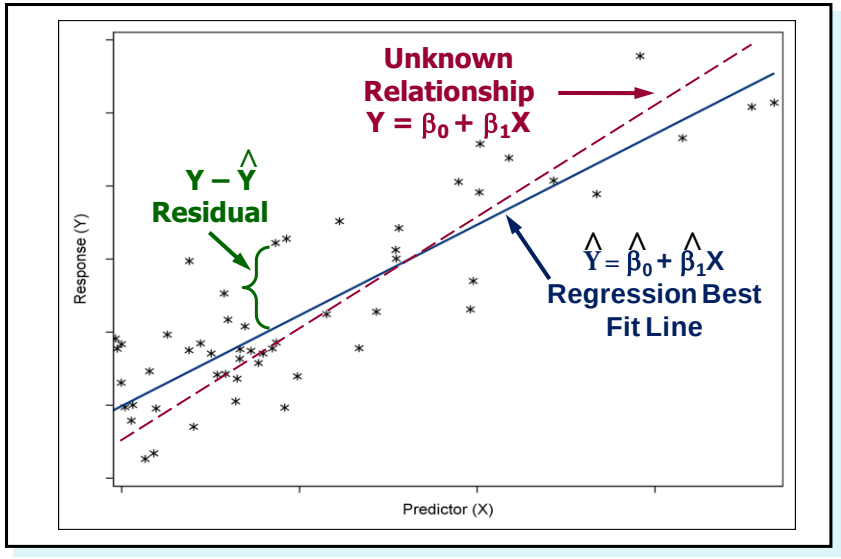
Producing Plots with the GPLOT Procedure

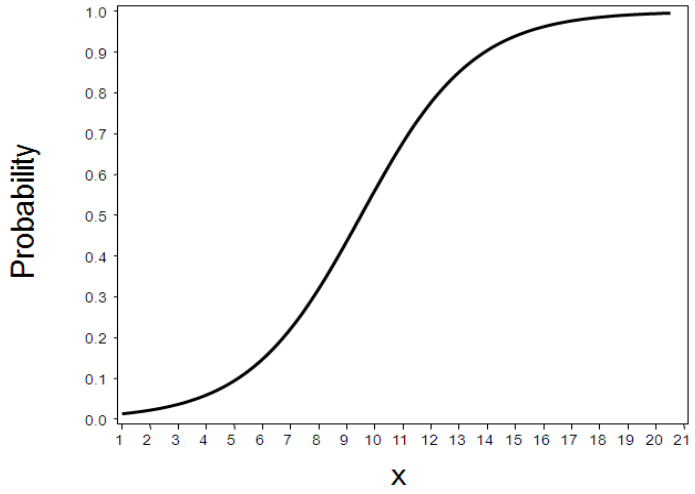
- You can use the GPLOT procedure to plot one variable against another within a set of coordinate axes.

- General form of a PROC GPLOT step:

```
PROC GPLOT DATA=SAS-data-set;
     PLOT vertical-variable*horizontal-variable </ options>;
RUN;
QUIT;
```

5. Regrese. Logistická regrese





Overview

	Type of Predictors		
Type of Response	**Categorical**	**Continuous**	**Categorical and Continuous**
Continuous	Analysis of Variance	Linear Regression	Analysis of Covariance (Regression with dummy variables)
Categorical	Logistic Regression or Contingency Tables	Logistic Regression	Logistic Regression

Přehled procedur SASu pro regresi

- SAS/STAT:

  CATMOD, GAM, GENMOD, GLIMMIX, GLM, LIFEREG, LOESS, LOGISTIC, MIXED, NLIN, NLMIXED, ORTHOREG, PHREG, PLS, PROBIT, REG, ROBUSTREG, RSREG, SURVEYLOGISTIC, SURVEYPHREG, SURVEYREG, TRANSREG.

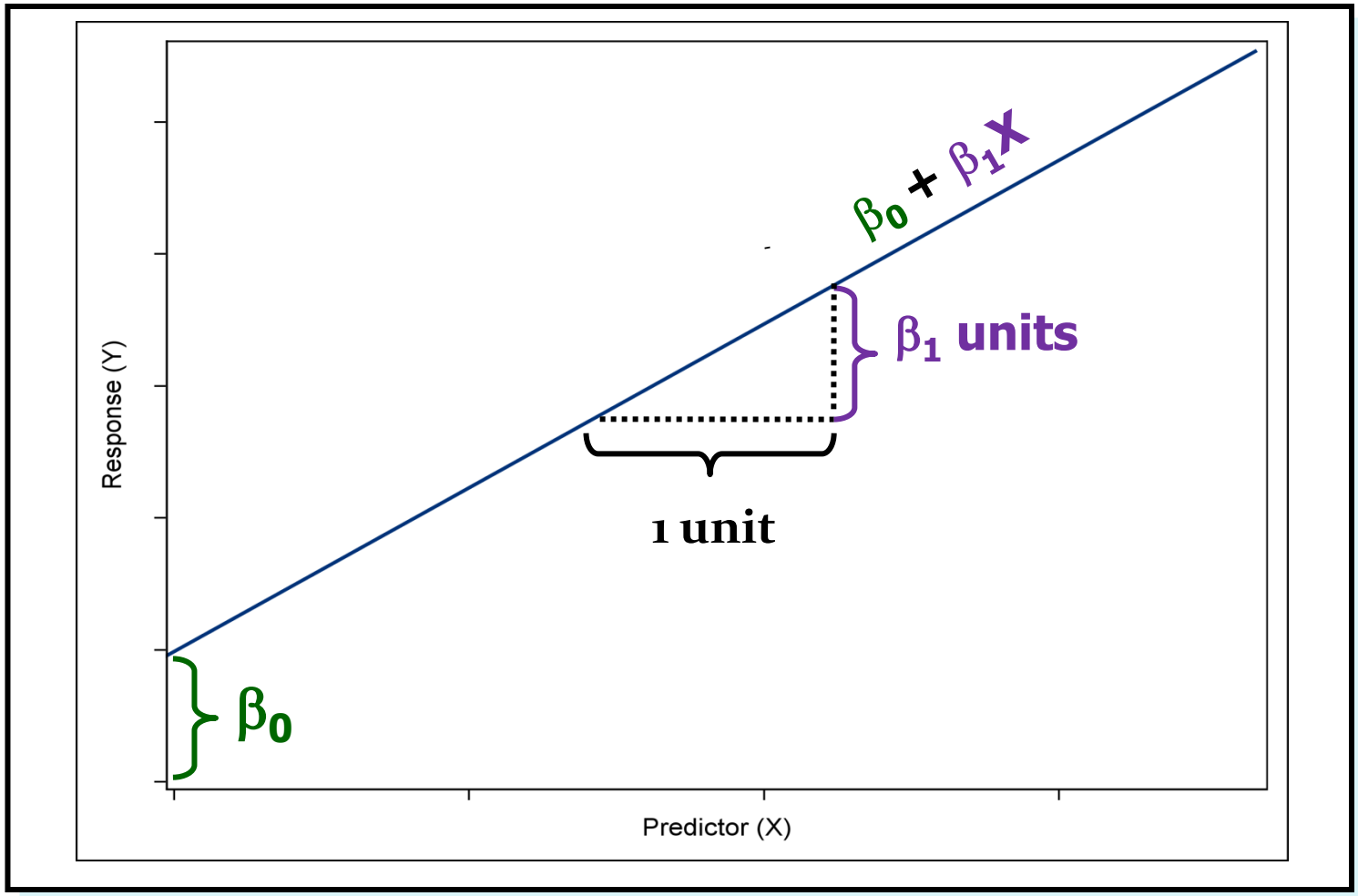
logistická regrese

„klasická“ lineární regrese

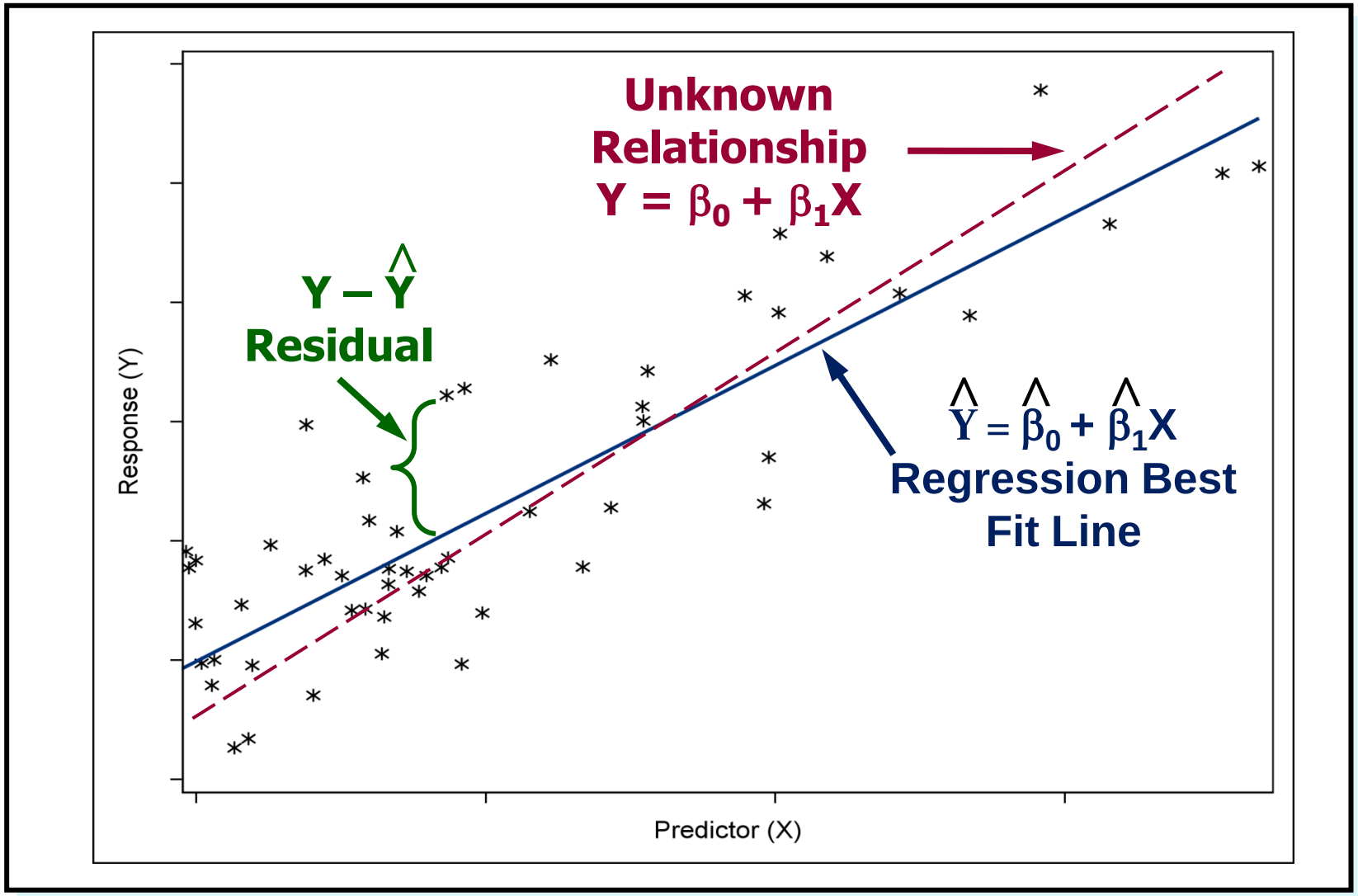
- SAS/ETS:

  AUTOREG, COUNTREG, MODEL, PANEL, PDLREG, SYSLIN.

Simple Linear Regression Model



Simple Linear Regression Model



The REG Procedure

- General form of the REG procedure:

```
PROC REG DATA=SAS-data-set <options>;
      MODEL dependent(s)=regressor(s) </ options>;
RUN;
```

Popis + jednoduchý příklad:

http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_reg_sect003.htm

Lineární regrese – PROC REG

PROC REG <options> ;
<label:>MODEL dependents=<regressors> </ options> ;
BY variables ;
FREQ variable ;
ID variables ;
VAR variables ;
WEIGHT variable ;
ADD variables ;
DELETE variables ;
<label:>MTEST <equation, ...,equation> </ options> ;
OUTPUT <OUT=SAS-data-set>< keyword=names> <...keyword=names> ;
PAINT <condition | ALLOBS> </ options > | < STATUS | UNDO> ;
RESTRICT equation, ...,equation ;
REWEIGHT <condition | ALLOBS> </ options > | < STATUS | UNDO> ;
PLOT <yvariable*xvariable> <=symbol> <...yvariable*xvariable> <=symbol> </ options> ;
PRINT <options> <ANOVA> <MODELDATA> ;
REFIT ;
RESTRICT equation, ...,equation ;
REWEIGHT <condition | ALLOBS> </ options > | < STATUS | UNDO> ;
<label:>TEST equation,<,...,equation> </ option> ;

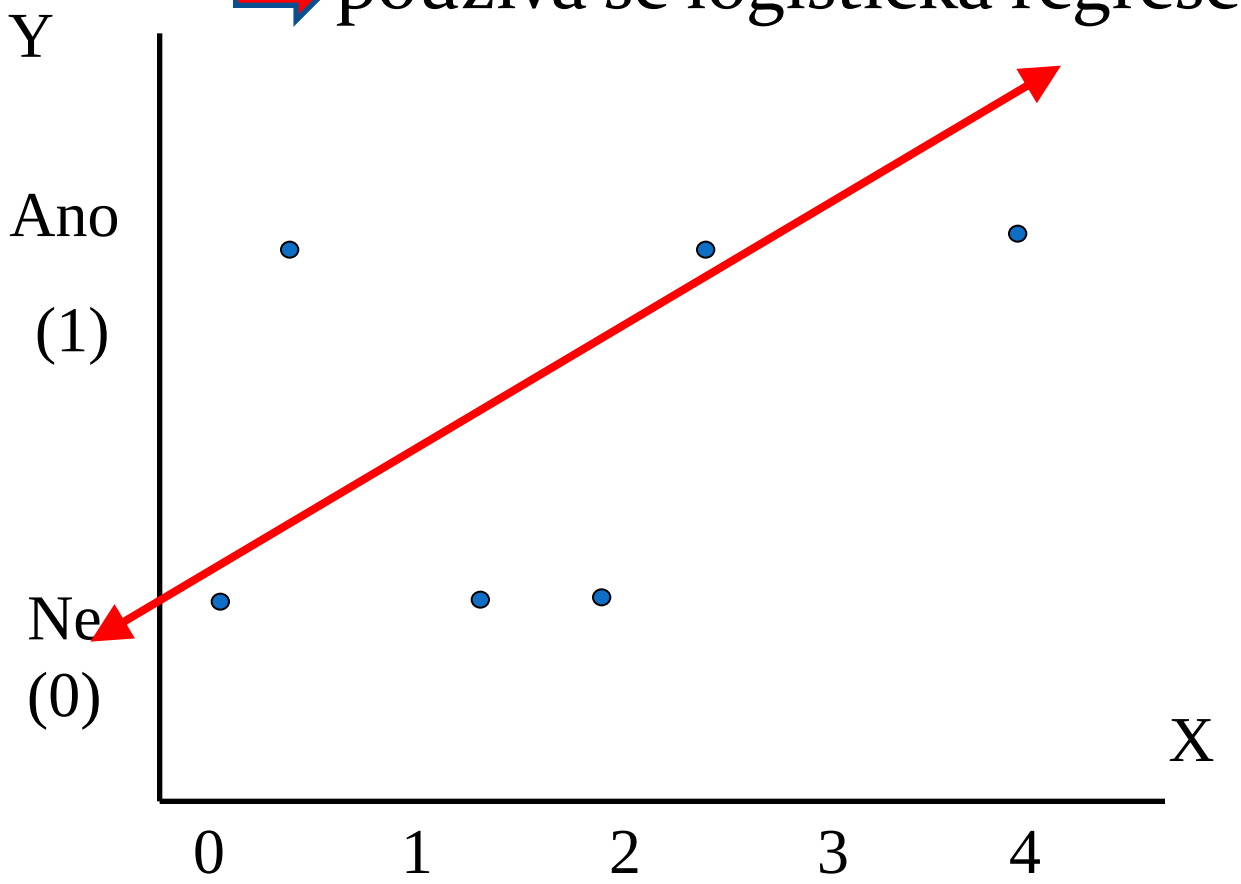
Více na: http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_reg_sect001.htm

Modelování kategoriální responze

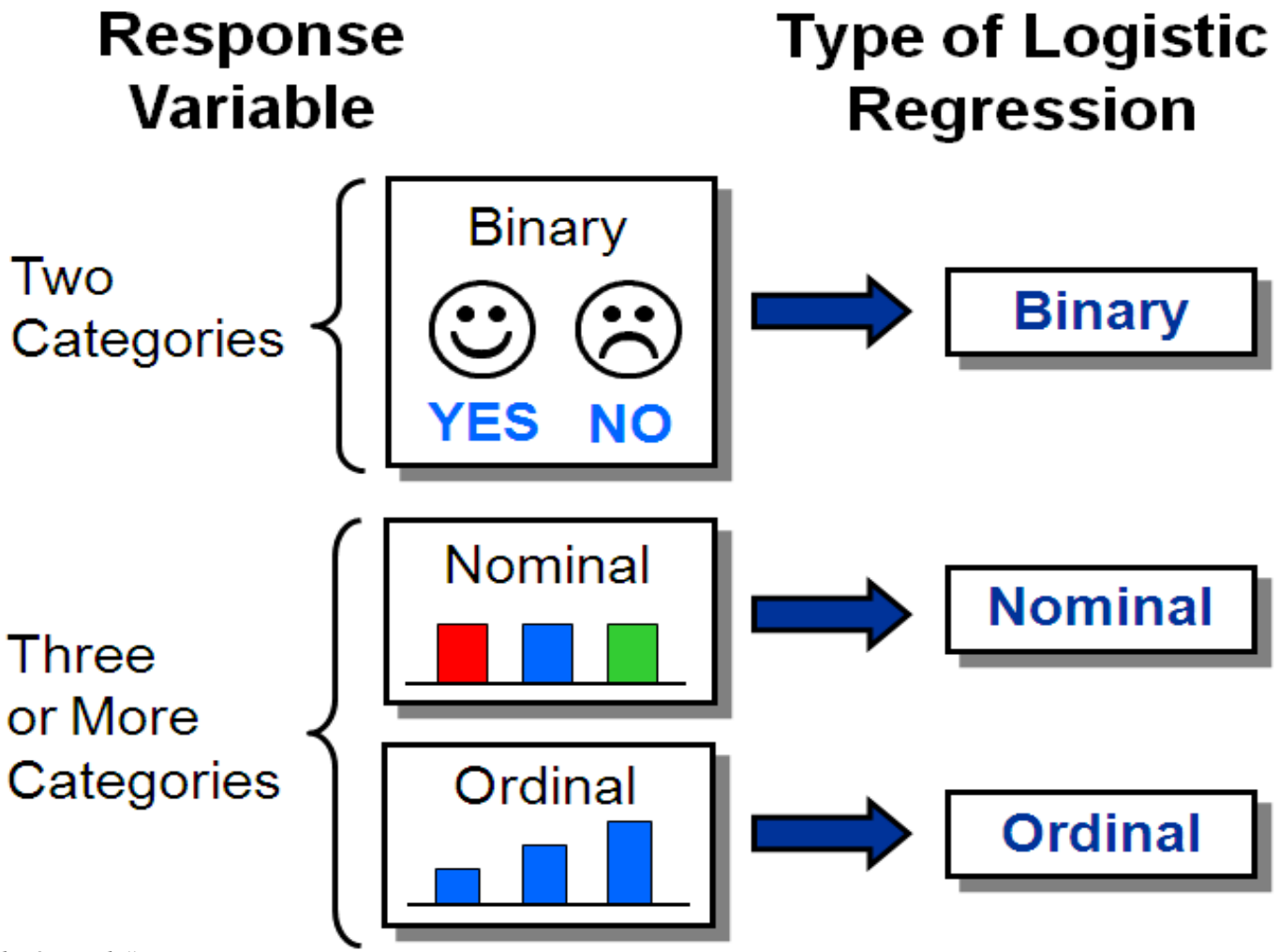
- Nastane default?

St.	X	Y
1	2.6	1
2	1.4	0
3	.65	1
4	4.1	1
5	.25	0
6	1.9	0

„klasická“ regrese není vhodná
➡ používá se logistická regrese.



Types of Logistic Regression



Why Not Ordinary Least Squares Regression?

$$Y_i = \beta_0 + \beta_1 X_{1i} + \varepsilon_i$$

- If the response variable is categorical, then how do you code the response numerically?
- If the response is coded (1=Yes and 0=No) and your regression equation predicts 0.5 or 1.1 or -0.4, what does that mean practically?
- If there are only two (or a few) possible response levels, is it reasonable to assume constant variance and normality?

What About a Linear Probability Model?

$$p_i = \beta_o + \beta_1 X_{1i} + \varepsilon_i$$

- Probabilities are bounded, but linear functions can take on any value. (Once again, how do you interpret a predicted value of -0.4 or 1.1?)
- Given the bounded nature of probabilities, can you assume a linear relationship between X and p throughout the possible range of X?
- Can you assume a random error with constant variance?
- What is the observed probability for an observation?

Měření pravděpodobnosti úspěchu

- Pravděpodobnost je měřena pomocí šance úspěchu (události).

- Jestliže P je pravděpodobnost události, pak (1-P) je pravděpodobnost, že nenastane.

- Šance události = P / 1-P

Logistická regrese

Simultánní efekt nezávislých (explanačních) proměnných na šanci

$$\text{Odds} = P/1\text{-}P = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k}$$

Jestliže logaritmujeme obě strany

$$\text{Log}\{P/1\text{-}P\} = \log e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k}$$

$$\text{Logit } P = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + .. + \beta_k X_k$$

Logit Transformation

- Logistic regression models transform probabilities called logits*.

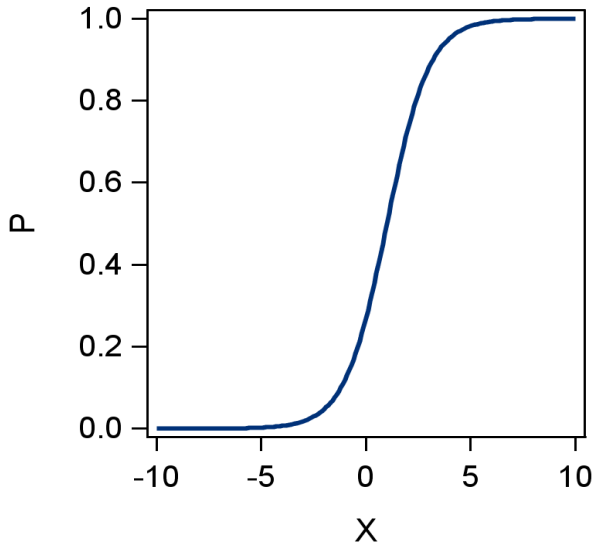
$$\text{logit}(p_i) = \ln\left(\frac{p_i}{(1-p_i)}\right)$$

where

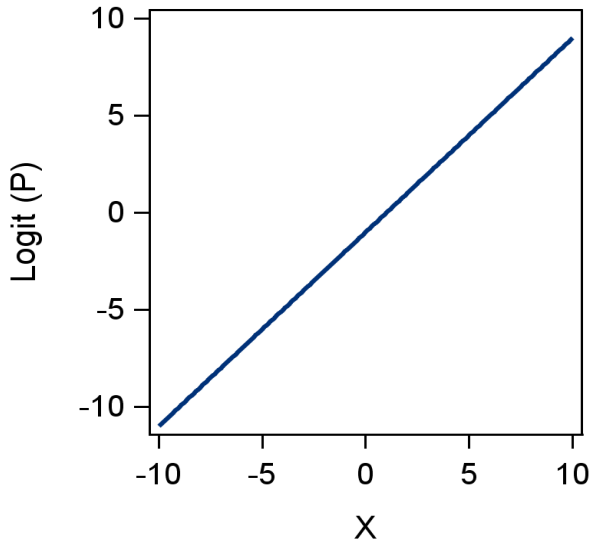
i	indexes all cases (observations)
p_i	is the probability the event (a default, for example) occurs in the i^{th} case
ln	is the natural log (to the base e).

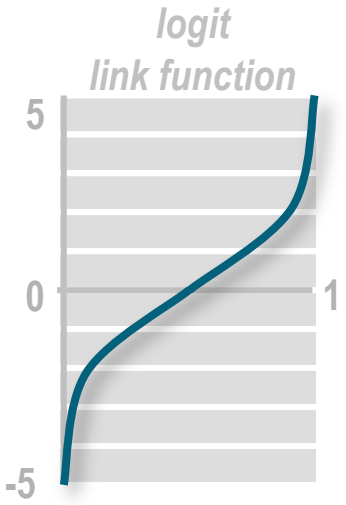
* The logit is the natural log of the odds.

Logit link function



Logit Transform





The logit link function transforms probabilities (between 0 and 1) to logit scores (between $-\infty$ and $+\infty$).

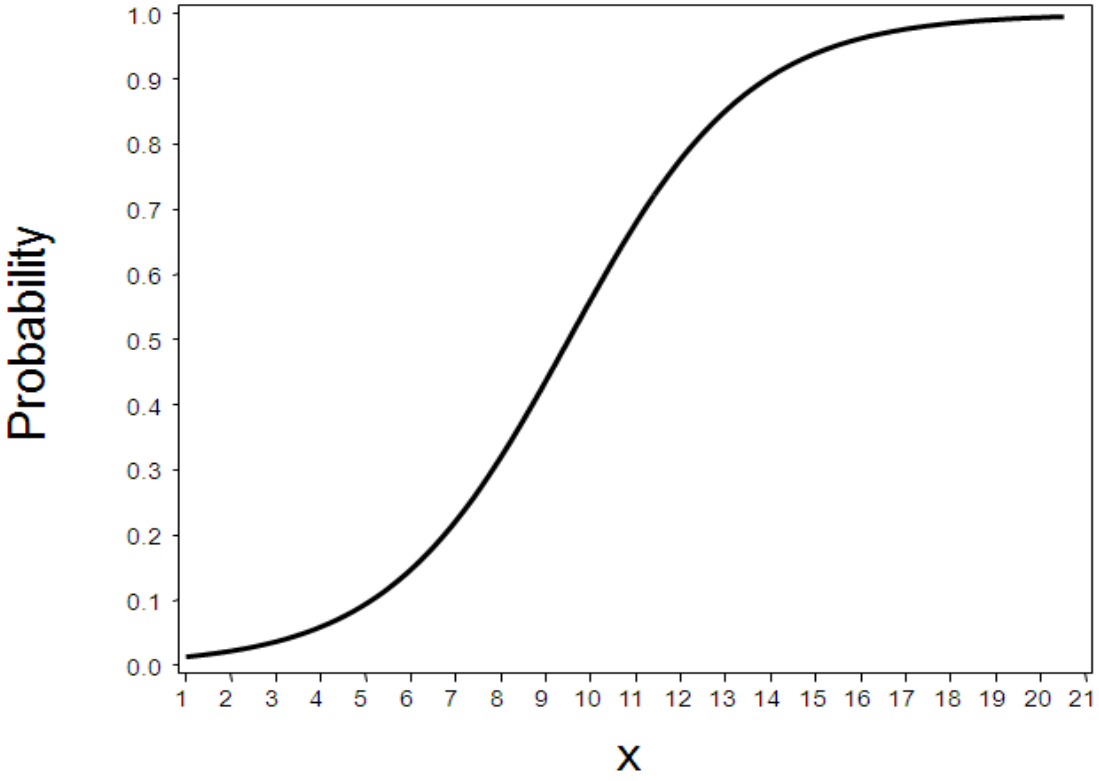
Logistic Regression Model

$$\text{logit}(p_i) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

where

- $\text{logit}(p_i)$= logit of the probability of the event
- β_0=intercept of the regression equation
- β_k=parameter estimate of the k^{th} predictor variable

Logistic Regression Curve



$$p_i = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\ldots+\beta_k X_k)}}$$

Logistic Regressions -example

$$\text{logit}(\hat{p}) = \hat{w}_0 + \hat{w}_1 x_1 + \hat{w}_2 x_2$$

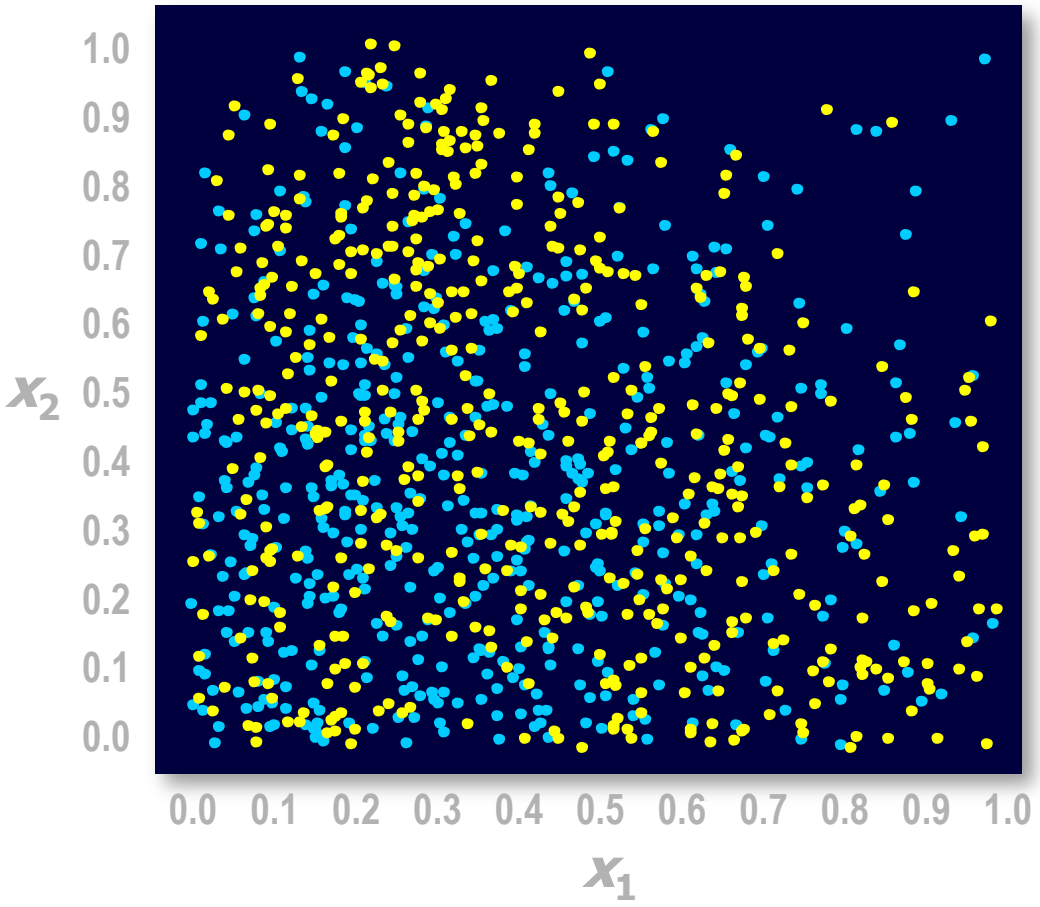
$$\hat{p} = \frac{1}{1 + e^{-\text{logit}(\hat{p})}}$$

Find parameter estimates by *maximizing*

$$\sum \log(\hat{p}_i) + \sum \log(1 - \hat{p}_i)$$

primary outcome training cases — *secondary outcome training cases*

log-likelihood function

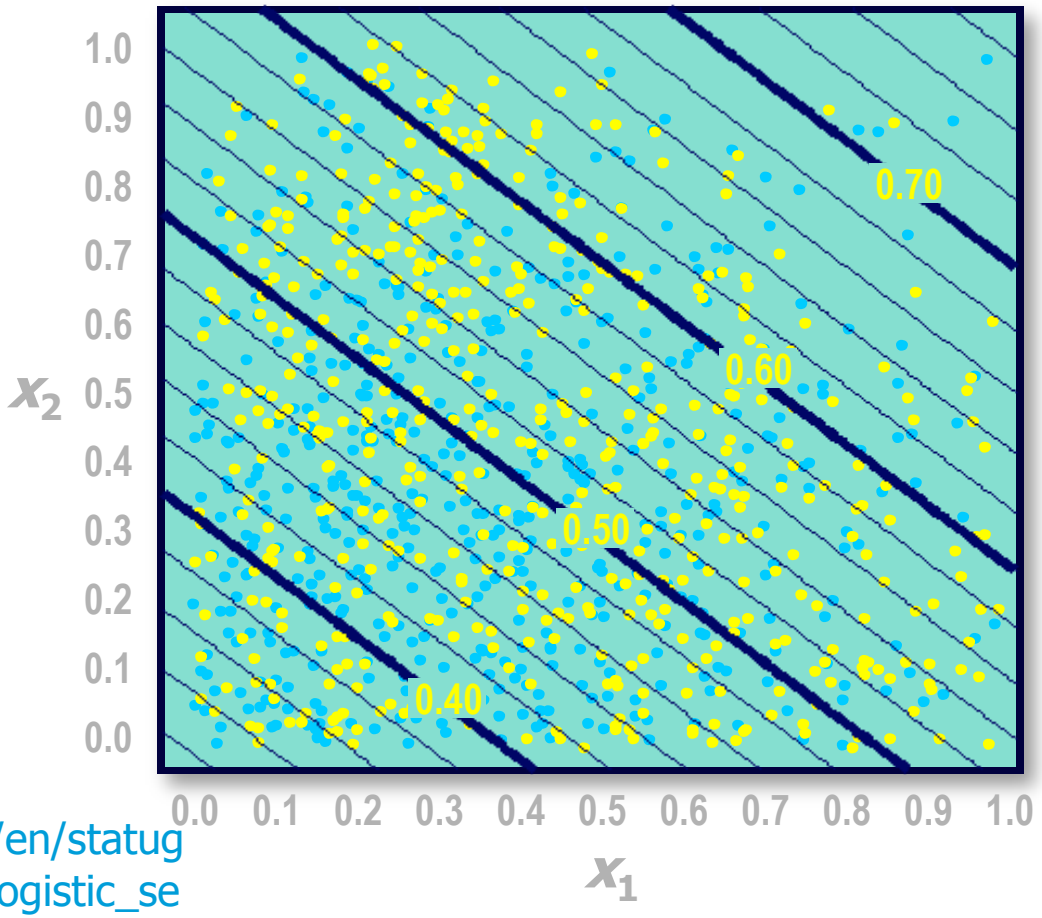


Logistic Regressions -example

$$\text{logit}(\hat{p}) = -0.81 + 0.92\,x_1 + 1.11\,x_2$$

$$\hat{p} = \frac{1}{1 + e^{-\text{logit}(\hat{p})}}$$

Using the maximum likelihood estimates, the prediction formula assigns a logit score to each x_1 and x_2.



Další příklad na:
http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_logistic_sect002.htm

Odhad parametrů

- Metoda maximální věrohodnosti vede na soustavu nelineárních rovnic.
- Tuto soustavu řešíme Netwon-Raphsonovou iterační metodou.
- Více na:
  - http://www.stat.cmu.edu/~cshalizi/402/lectures/14-logistic-regression/lecture-14.pdf
  - http://czep.net/stat/mlelr.pdf
  - http://www.stat.psu.edu/~jiali/course/stat597e/notes2/logit.pdf

Maximálně věrohodný odhad (MLE)

MLE is a general purpose method for parametric model estimation.
We will make use of it to estimate the logistic regression.

If we have a model with parametric structure $\boldsymbol{\theta}$, we can compute the **likelihood** that the model will generate a sequence of n observations $\mathbf{D} = (d_1, \ldots, d_n)$.

$$L(\boldsymbol{\theta}|\mathbf{D}) = P(\mathbf{D}|\boldsymbol{\theta})$$

The model which best fits the data is selected as the one which maximizes this likelihood.

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathbf{D})$$

If we *assume independence between the observations*, this then gives

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} P(d_i|\boldsymbol{\theta})$$

Zdroj: http://www2.imperial.ac.uk/~abellott/Credit%20Scoring%202.pdf

Maximálně věrohodný odhad

This MLE can be expressed more conveniently in terms of log-likelihoods (since log is monotonic on its argument):

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log P(d_i|\boldsymbol{\theta})$$

Remember:

- We do not know the true value of the parameter $\boldsymbol{\theta}$, but we want to estimate it.
- To distinguish the estimate from the true value, in our notation, we put a "hat" on the estimate: $\widehat{\boldsymbol{\theta}}$.

MLE has several nice asymptotic properties:

- Consistency
- Asymptotic normality
- Efficiency.

Maximálně věrohodný odhad

Consider the training data set D_{train} with n observations (borrowers).

Remember

- $\mathbf{x}_i$ denotes values for predictor variables for observation i.
- y_i denotes the outcome for observation i, either 0 or 1.

Then the likelihood of the outcome for each observation i is given by

$$\begin{array}{ll} P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta}) & \text{if } y_i{=}0, \\ 1 - P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta}) & \text{if } y_i{=}1 \end{array}$$

which is

$$P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta})^{1-y_i}\big(1 - P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta})\big)^{y_i}$$

giving log-likelihood for each observation:

$$(1 - y_i)\log P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta}) + y_i \log\big(1 - P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta})\big)$$

Maximálně věrohodný odhad

Assuming independence between observations, this gives the log-likelihood function for $\boldsymbol{\beta}$:

$$\log L(\boldsymbol{\beta}|D_{\text{train}}) = \sum_{i=1}^{n} (1 - y_i) \log\left(\frac{1}{1 + e^{-(\beta_0 + \boldsymbol{\beta}\cdot\mathbf{x}_i)}}\right) + y_i \log\left(\frac{1}{1 + e^{\beta_0 + \boldsymbol{\beta}\cdot\mathbf{x}_i}}\right)$$

Differentiating by each coefficient in $\boldsymbol{\beta}$ and setting the derivative equal to zero to find the maxima gives

$$\sum_{i=1}^{n} \left(1 - y_i - \left(\frac{1}{1 + e^{-(\beta_0 + \boldsymbol{\beta}\cdot\mathbf{x}_i)}}\right)\right) = 0$$

and

$$\sum_{i=1}^{n} x_{ij} \left(1 - y_i - \left(\frac{1}{1 + e^{-(\beta_0 + \boldsymbol{\beta}\cdot\mathbf{x}_i)}}\right)\right) = 0$$

for each attribute j=1 to m.

These are non-linear equations that can be solved by computer intensive processes such as Newton-Raphson methods.

Standard errors on the MLE

Since $\widehat{\boldsymbol{\theta}}$ is only an estimate of the best model to explain the data, it is possible to derive standard errors $\hat{\boldsymbol{s}}$ on the estimates.

Asymptotic normality for MLE is such that

$$\frac{(\hat{\theta}_j - \theta_j)}{\hat{s}_j} \rightarrow N(0,1) \text{ as } n \rightarrow \infty$$

where $\hat{\theta}_j$, θ_j and $\hat{s}_j$ are the jth components of $\widehat{\boldsymbol{\theta}}$, $\boldsymbol{\theta}$ and $\hat{\boldsymbol{s}}$ respectively and $N(0,1)$ is the standard normal distribution.

This property then allows us to generate:-

- Generate a hypothesis tests using the Wald chi-square statistic;
- Generate confidence intervals around the estimate.

MLE- testování hypotéz

We test the hypothesis that an estimated coefficient is not zero against the null hypothesis that it is zero. That is, we testing if a parameter has a genuine effect in the model.

- Null hypothesis: $H_0: \theta_j = 0$
- Alternative hypothesis: $H_1: \theta_j \neq 0$

The Wald test says reject H_0 if $\frac{|\hat{\theta}_j|}{\hat{s}_j} > z_{\alpha/2}$ for some significance level α, where $z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2)$ and Φ is the CDF for the standard normal distribution.

MLE – konfidenční intervaly

The asymptotic normality property also allows us to compute confidence intervals (CIs):

$$P\left(\hat{\theta}_j - z_{\alpha/2}\hat{s}_j < \theta_j < \hat{\theta}_j + z_{\alpha/2}\hat{s}_j\right) \to 1 - \alpha$$

as $n \to \infty$.

This is a range of possible values of the parameter within a given confidence level $1 - \alpha$.

Note: the larger the confidence level, the broader the confidence interval.

Likelihood Ratio Test

The maximized likelihood gives a measure of how well the model fits the data (1=perfect fit, 0=no fit). The ratio of likelihoods between two models, A "nested" in B, can be used to test whether the fit of A improves on B.

Definitions

Suppose we have two models A and B with the same structure except A has more parameters than B:

$$\boldsymbol{\theta}_A = (\theta_1, \ldots, \theta_{m+r}) \text{ and } \boldsymbol{\theta}_B = (\theta_1, \ldots, \theta_m)$$

Then A *is nested in* B.

The *likelihood ratio statistic* is $\lambda = 2\log\left(\frac{L(\hat{\boldsymbol{\theta}}_A)}{L(\hat{\boldsymbol{\theta}}_B)}\right)$.

Newton-Raphsonova metoda

- Základní princip metody:

$$p(x,\beta)=\frac{1}{1+e^{-\beta^T x}} \qquad L(\beta)=\sum_{i=1}^{n} y_i \beta^T x_i - \log(1+e^{\beta^T x_i}) \qquad \beta^{new}=\beta^{old}-\frac{\partial^2 L(\beta)}{\partial\beta\partial\beta^T}^{-1}\frac{\partial L(\beta)}{\partial\beta}$$

- Maticový zápis:

$$\beta^{new}=(X^T W X)^{-1} X^T W (X\beta^{old}+W^{-1}(y-p))$$

y ... vektor pozorování vysvětlované proměnné y_i

X ... matice plánu, typu $n\times(p+1)$

p ... vektor pravděpodobností $p(x_i,\beta^{\text{old}})$

W ... $n\times n$ diagonální matice vah, s diag. prvky $p(x_i,\beta^{\text{old}})\cdot(1-(x_i,\beta^{\text{old}}))$

- Jde o numerickou iterační metodu -> je třeba zkontrolovat, zda byla splněna podmínka konvergence (metoda „dokonvergovala“ k optimálnímu řešení)

Výhody logistické regrese

- Málo parametrů
- Snadné použití i interpretace
- Lze snadno začlenit i diskrétní prediktory
- Funguje dobře i na datech, která se poměrně značně liší od gaussovských směsí
- A především většinou dobře funguje, pokud věnujeme odpovídající pozornost přípravě dat
  - praktická zkušenost: ve čtyřech případech z pěti je logistická regrese na datech, která analyzuji, buď nejlepší nebo zhruba stejně dobrá jako jiné metody.

Interpretace, rozdíly proti OLS

- Regresní koeficienty b: kladné znamenají, že proměnná svým růstem zvyšuje šanci zařazení do skupiny kódované číslem 1, a naopak záporné indikují pokles této šance
- Často se používá $\exp(b_i)$: je to faktor, kterým se násobí šance $p/(1-p)$ při jednotkovém nárůstu x_i a neměnných ostatních x_k
  - Pozor na různá měřítka, v nichž x_i mohou být měřena;

- Místo F-testu celkové validity nyní máme chí-kvadrátový test pro totéž
- Místo t-testu signifikance proměnných v modelu jsou Waldovy statistiky; je to v podstatě totéž a čteme to stejně
- Místo R^2 jsou jen pseudo-R^2

Příklad

The following logistic regression output was produced on a data set of 40,000 credit cards.

Likelihood Ratio = 1819 (p-value < 0.001)

Variable	Coefficient	Estimate	Standard error	Wald chi-square	P > chi-square
Intercept	β_0	-0.181	0.084	4.6	0.032
Age	β_1	+0.0353	0.0013	757.6	<0.001
Income (log)	β_2	-0.0164	0.0100	2.67	0.10
Residential phone	β_3	+0.622	0.030	430.8	<0.001
Home owner *		0			
Renter	β_4	-0.155	0.039	15.6	<0.001
Lives with parents	β_5	+0.256	0.045	32.1	<0.001
Months in residence	β_6	-0.00025	0.00011	5.4	0.020
Months in current job	β_7	+0.00210	0.00025	72.9	<0.001

* Notice that the Home owner category is set as base residency category and so has no coefficient estimate. We will discuss this in a later lecture.

Zdroj: http://www2.imperial.ac.uk/~abellott/Credit%20Scoring%202.pdf

Příklad

We have used logistic regression to model the negative outcome (ie $y = 0$).

- This may seem odd given that the outcome of interest is the positive one (eg default).
- However, this model ensures the log-odds scores are the right way round: ie increasing scores imply increasing creditworthiness.
- There is no material difference. If we had modelled $y = 1$, the signs on the coefficient estimates would be reversed but everything else would be the same.

Interpretations:

- The estimates (highlighted) form the scorecard.
- Estimates greater than 0 indicate relative decrease in risk.
- Estimates less than 0 indicate relative increase in risk.
- Small p-values indicate coefficients that are statistically significantly different to zero (how small?).
- Large p-values indicate coefficients that have a good chance of actually being zero.

Příklad

Remember in the exercise in Chapter 1 we gave details of six borrowers. You were asked to select three to accept and three to reject.

Here the scores assigned by the model above are shown. The observations with the three lowest scores are rejected by the model. The actual outcome in each case is also shown. *How does your performance compare with the model?*

Age	Monthly Income (£)	Residential phone?	Residence type?	Months in residence	Months in current job	Score	Model accept or reject?	Actual outcome
22	1,145	Yes	Home owner	48	12	1.11	Reject	Good
46	15,500	Yes	Renter	48	192	2.14	Accept	Good
71	900	Yes	Renter	96	12	2.68	Accept	Good
32	5,000	Yes	Renter	48	168	1.61	Accept	Bad
25	1,385	Yes	Renter	12	0	1.05	Reject	Bad
43	3,145	No	Home owner	96	36	1.25	Reject	Bad

Příklad

Variable	Value	Coefficient	Estimate	Value × Estimate
Intercept	n/a	β_0	-0.181	-0.181
Age	22	β_1	+0.0353	+0.777
Income (log)	log(1145) =7.04	β_2	-0.0164	-0.116
Residential phone	1	β_3	+0.622	+0.622
Home owner *	1		0	0
Renter	0	β_4	-0.155	0
Lives with parents	0	β_5	+0.256	0
Months in residence	48	β_6	-0.00025	-0.012
Months in current job	12	β_7	+0.00210	0.025
Score (sum)				**+1.115**

Compute the PD of the borrower.

Score = 1.115

$$P(y = 1|s) = \frac{1}{1+e^s} \approx 0.25.$$

Multinomiální logistická regrese

- Taktéž polytomická regrese
- Závisle proměnná má *M* kategorií, více než dvě. Např.: kterou stranu respondent volí?
- Základní idea:
  - Prohlásit jednu kategorii za referenční
  - Spočítat *M*–1 obyčejných logistických modelů pro každou ze zbylých kategorií oproti referenční
  - A predikovat tu kategorii, kde vyšla největší pravděpodobnost přes všechny modely

Budování modelu

- Forward
  - začíná se s prázdným modelem
  - postupné přidávání proměnných
- Backward
  - začíná se s plným modelem (všechny proměnné)
  - postupné odebírání proměnných
- Stepwise
  - začíná se s prázdným modelem
  - postupně se přidávají a odebírají proměnné
- Enter
  - je předepsán seznam proměnných v modelu

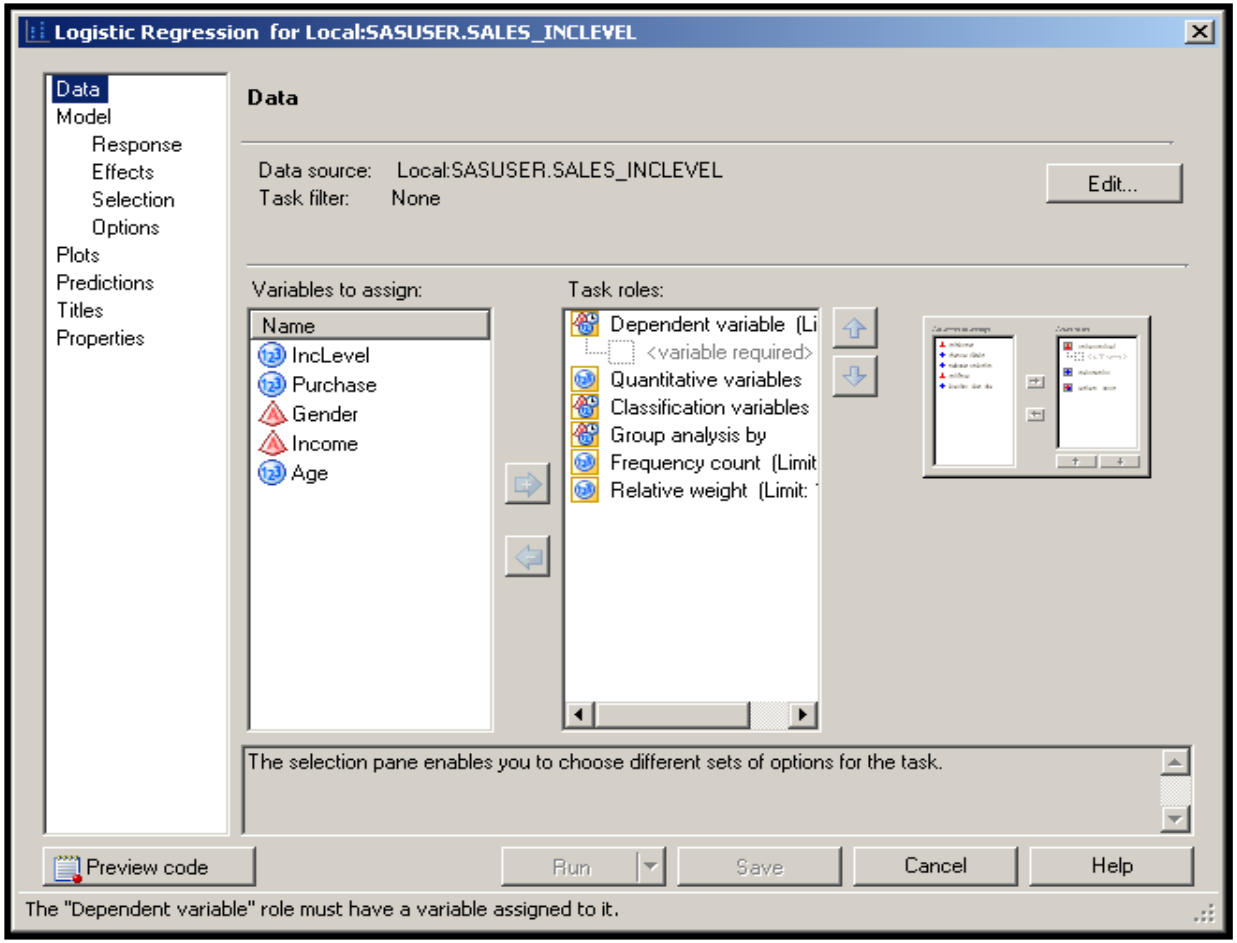
Logistic Regression with Sequential Steps

- Forward regression
  - starts with a baseline model (intercept-only)
  - searches all variables and finds the strongest one
  - keeps adding variables in order of strength until no significant improvement is achieved in the model.
- Backwards regression
  - starts with a full model using all variables
  - removes the weakest input variable provided that taking it out does not cause a significant reduction in the fit of the model
  - continues removing the weakest input variables in order unless there is a significant reduction in the fit of the model; at which point the algorithm stops.

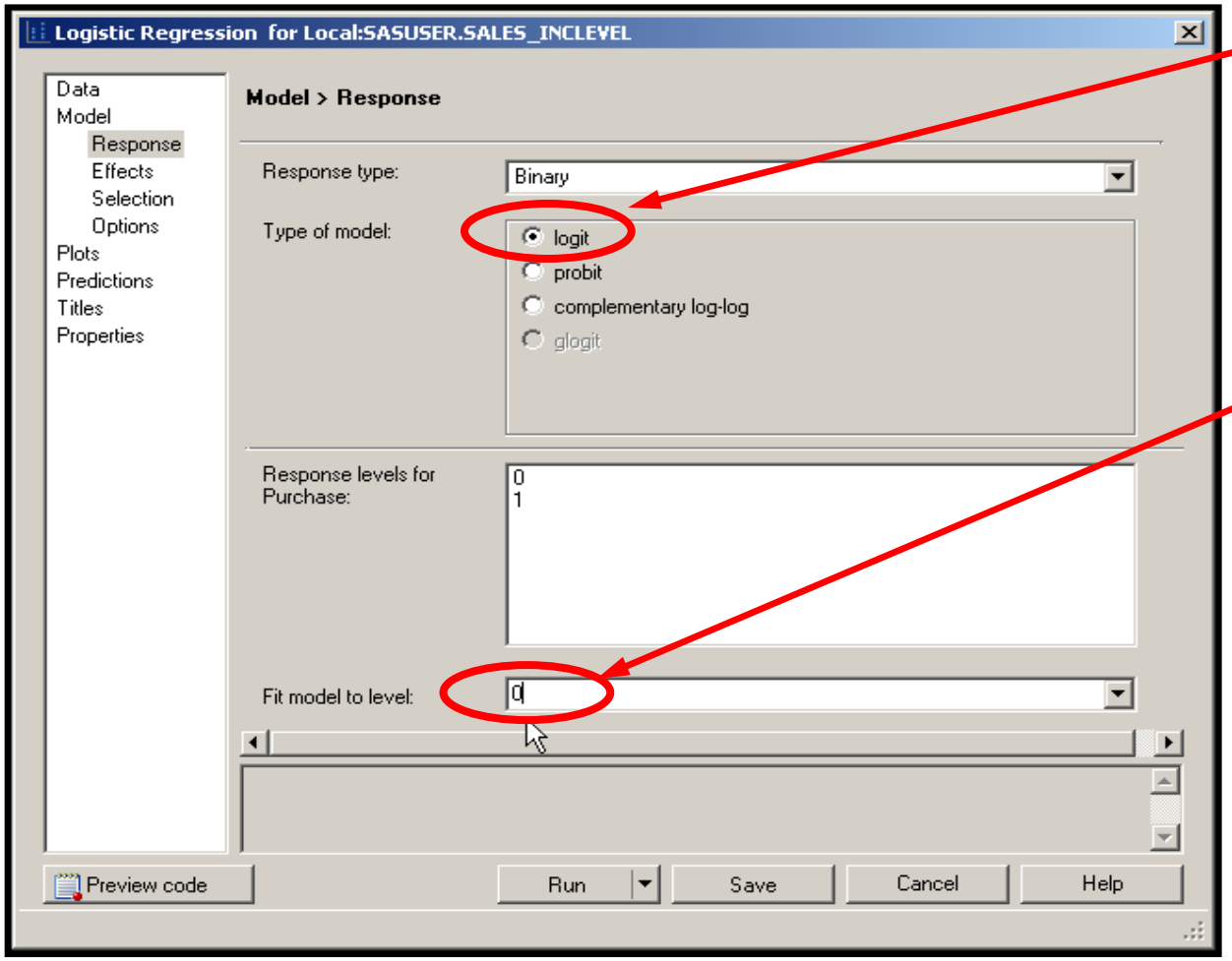
Logistic Regression with Sequential Steps

- Stepwise regression
  - is a combination of forward and backward regression
  - begins the same way as forward
  - re-evaluates the statistical significance of all included variables after each new variable is added.
- ✎ If a previously included variable becomes statistically insignificant when a new variable is added, that variable is then removed.
- ✎ The algorithm stops when no more variables can be found that add significantly to the fit of the model **and** all variables remaining in the model are statistically significant.

The Logistic Regression Task



Which link function, which response Level to Model?



Volba linkovací funkce.

Specify the level of the response variable that you want to model.
For example, do you want to model the probability of a 0 or a 1?

LOGISTIC Procedure

- General form of the LOGISTIC procedure:

```
PROC LOGISTIC DATA=SAS-data-set
        <options>;
        CLASS variables </ options>;
        MODEL response=predictors </ options>;
        UNITS independent1=list ... </ options>;
        ODDSRATIO <'label'> variable </ options>;
        OUTPUT OUT=SAS-data-set
        keyword=name                    </ options>;
RUN;
```

Více např. na: http://www.okstate.edu/sas/v8/sashtml/onldoc.htm
http://www.okstate.edu/sas/v8/saspdf/stat/chap39.pdf

LOGISTIC Procedure - příklad

```
ods html file="logistic_vyvoj.html" style=sasweb;
proc logistic data=dm1.data_vyvoj descending;
model good4=goods_type_w phone_w a_uver_w
        fam_state_w income_w credit_w vek_w
            ;
run;
ods html close;
```

LOGISTIC Procedure - příklad

```
proc logistic data=dm1.score_base outest=work.model_def;

CLASS AGE_d EDUCATION_d CAR_AGE_d / param=glm;
MODEL def_bad = AGE_d EDUCATION_d CAR_AGE_d
total_income_d(init_pay_by_INCOME_d)

/ SELECTION=FORWARD HIERARCHY=MULTIPLECLASS;
score  out=work.tab_scored_def;
run;
```

LOGISTIC Procedure - příklad

```
proc logistic
data=dm1.score_base outest=work.model_def namelen=200;
where client_type="1-Novy";
CLASS sex_k child_num_k fam_state_k age_k;
MODEL def_bad = AGE_w EDUCATION_w
        AGE_w*EDUCATION_w
        sex_k|child_num_k|fam_state_k|age_k@4

/selection=stepwise slentry=0.6 slstay=0.1 details corrb
;
run;
```

LOGISTIC Procedure - příklad

```
proc logistic
data=dm1.score_base  inest=hc.modelSU namelen=200;
CLASS sex_k child_num_k fam_state_k age_k;
MODEL def_bad = AGE_w EDUCATION_w
        AGE_w*EDUCATION_w
        sex_k|child_num_k|fam_state_k|age_k@4

/selection=none maxiter=0;
output out=dm1.data_all_scr (keep=id_credit score def_bad
        compress=yes)
prob=score;
run;
```

What Happens to Classification Variables?

- The Logistic Regression task assumes a linear relationship between predictors and the logit for the response.
  - For categorical variables, that assumption cannot be met.
- Specification as a Classification variable creates "design variables" representing the information in the categorical variables.
  - The design variables are the ones actually used in model calculations.
  - There are many possible "parameterizations" of the design variables.

Effects (Default) Coding: Three Levels

			Design Variables	
CLASS	Value	Label	1	2
IncLevel	1	Low Income	1	0
	2	Medium Income	0	1
	3	High Income	-1	-1

Effects Coding: An Example

$$\text{logit}(p) = \beta_0 + \beta_1 * D_{\text{Low income}} + \beta_2 * D_{\text{Medium income}}$$

β_0 = the average value of the logit across all categories

β_1 = the difference between the logit for Low income and the average logit

β_2 = the difference between the logit for Medium income and the average logit

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-0.5363	0.1015	27.9143	<.0001
IncLevel	1	1	-0.2259	0.1481	2.3247	0.1273
IncLevel	2	1	-0.2200	0.1447	2.3111	0.1285

Reference Cell Coding: Three Levels

			Design Variables	
CLASS	Value	Label	1	2
IncLevel	1	Low Income	1	0
	2	Medium Income	0	1
	3	High Income	0	0

Reference Cell Coding: An Example

$$\text{logit}(p) = \beta_0 + \beta_1 * D_{\text{Low income}} + \beta_2 * D_{\text{Medium income}}$$

β_0 = the value of the logit when income is High

β_1 = the difference between the logits for Low and High income

β_2 = the difference between the logits for Medium and High income

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-0.0904	0.1608	0.3159	0.5741
IncLevel	1	1	-0.6717	0.2465	7.4242	0.0064
IncLevel	2	1	-0.6659	0.2404	7.6722	0.0056

Odds Ratio Calculation from the Current Logistic Regression Model

- Logistic regression model:

$$\text{logit}(\text{p}) = \log(odds) = \beta_0 + \beta_1 * (\text{gender})$$

- Odds ratio (females to males):

$$\text{odds}_{\text{females}} = e^{\beta_0 + \beta_1}$$

$$\text{odds}_{\text{males}} = e^{\beta_0}$$

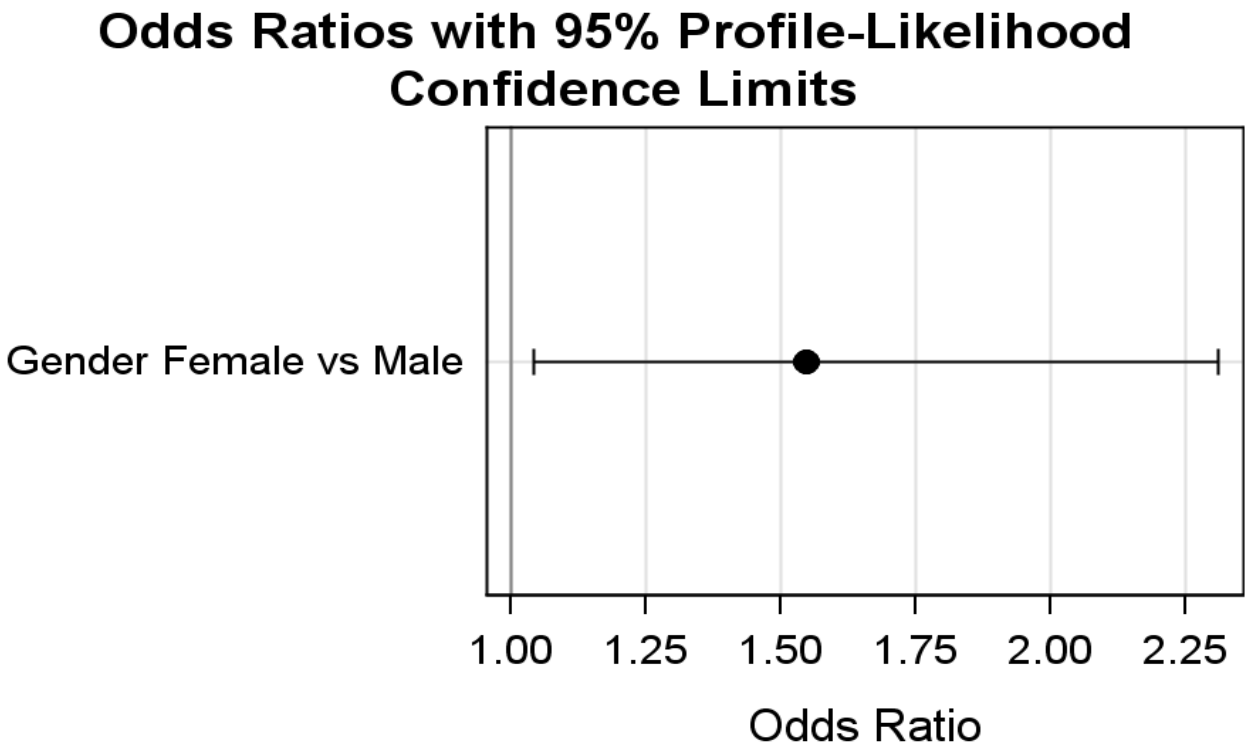
$$\text{odds ratio} = \frac{e^{\beta_0 + \beta_1}}{e^{\beta_0}} = e^{\beta_1}$$

Odds Ratios for Categorical Predictors

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Gender Female vs Male	1.549	1.040	2.305

Profile Likelihood Confidence Interval for Odds Ratios				
Effect	Unit	Estimate	95% Confidence Limits	
Gender Female vs Male	1.0000	1.549	1.043	2.312

Odds Ratio Plot

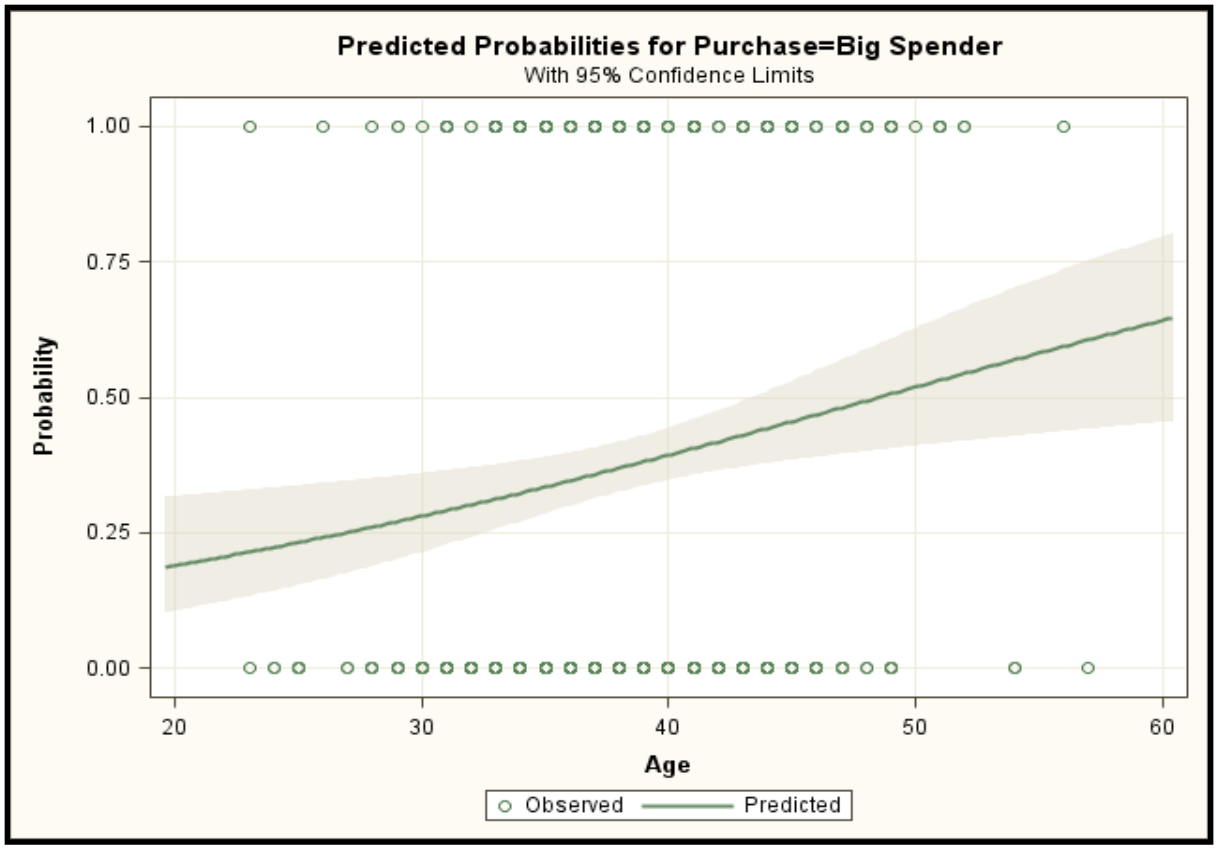


Odds Ratios for Continuous Predictors

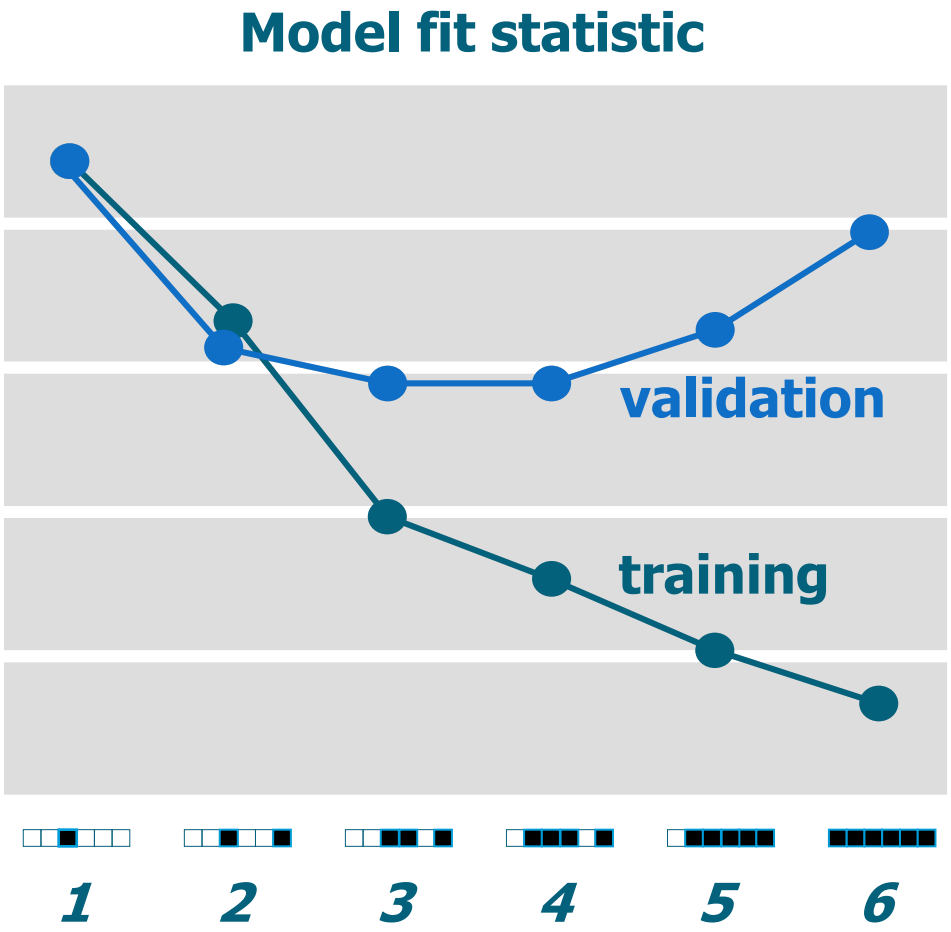
Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Age	1.052	1.016	1.090

Profile Likelihood Confidence Interval for Odds Ratios				
Effect	Unit	Estimate	95% Confidence Limits	
Age	10.0000	1.663	1.176	2.373

Predicted Probability Plots – Continuous

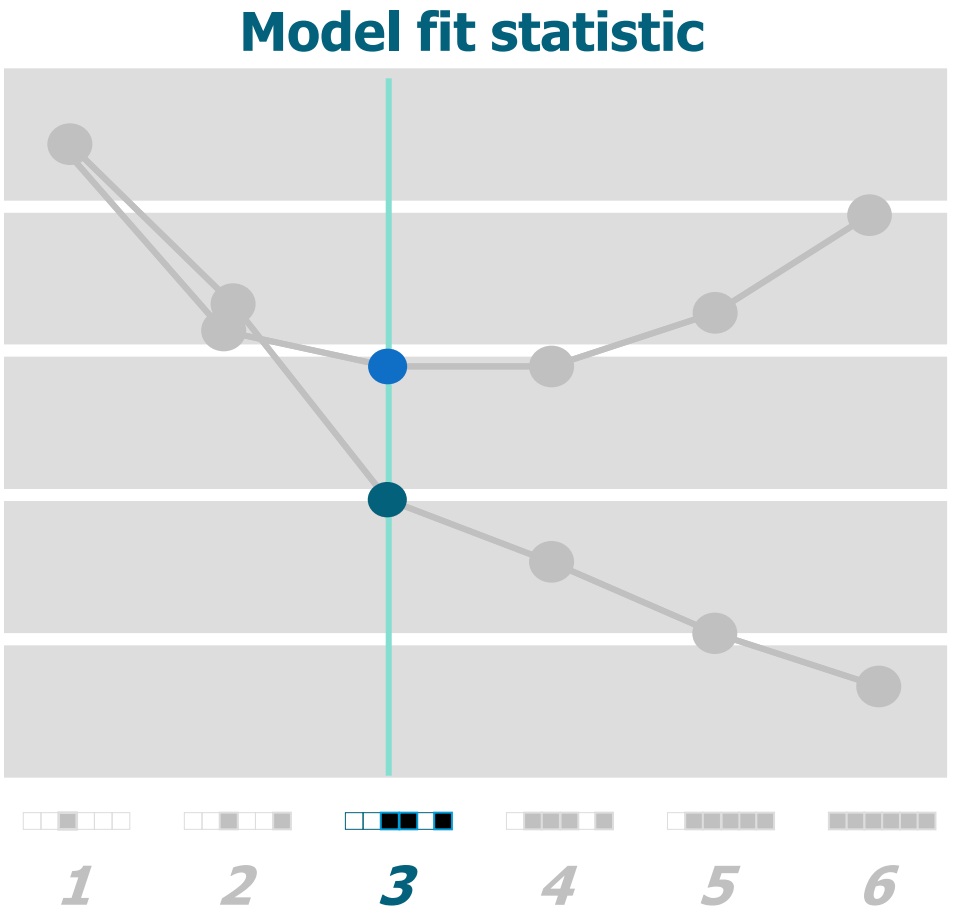


Model Fit versus Complexity



Evaluate each sequence step.

Select Model with Optimal Validation Fit



Evaluate each sequence step.

Choose simplest optimal model.

Model Assessment: Comparing Pairs

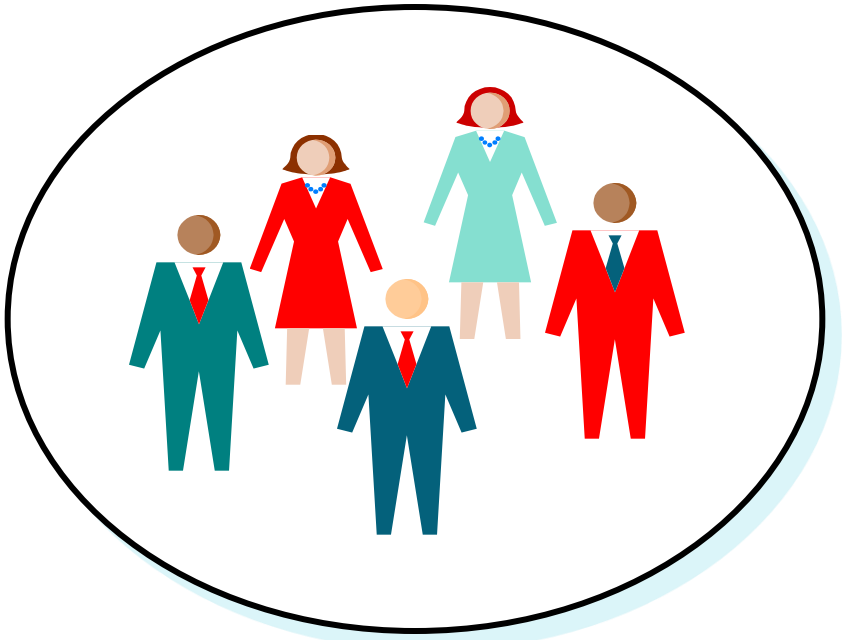
- Counting concordant, discordant, and tied pairs is a way to assess how well the model predicts its own data and therefore how well the model fits.
- In general, you want a high percentage of concordant pairs and low percentages of discordant and tied pairs.

- Následuje příklad určení těchto párů na modelu predikujícím zda daná osoba nakoupí zboží za více než 100$.

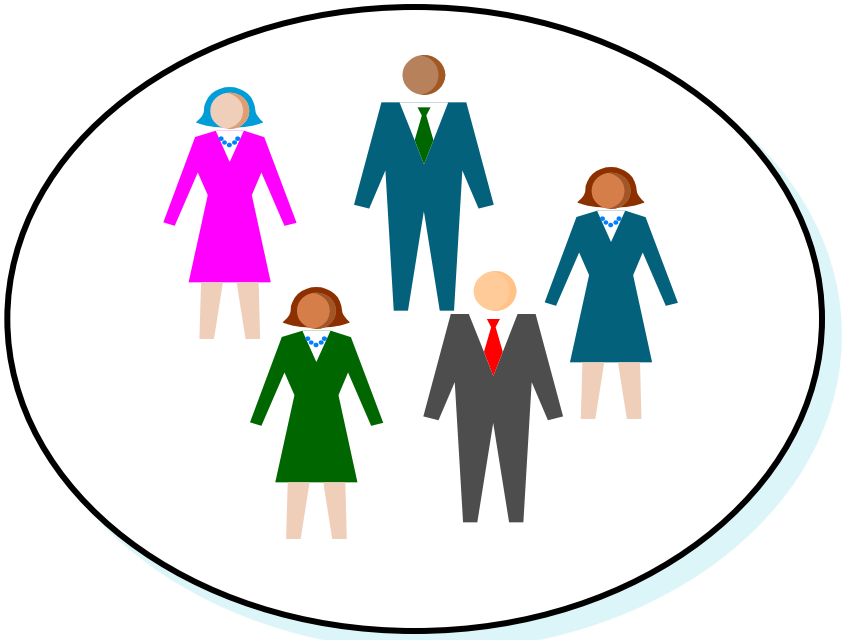
Comparing Pairs

To find concordant, discordant, and tied pairs, compare everyone who had the outcome of interest against everyone who did not.

< $100



$100 +



Concordant Pair

Compare a woman who bought more than $100 worth of goods from the catalog and a man who did not.

< $100

P(100+) = .32

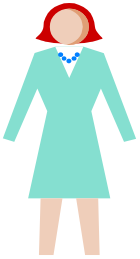
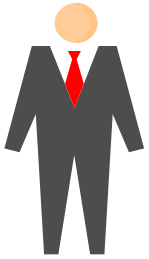
$100 +

P(100+) = .42

The actual sorting agrees with the model.
This is a **concordant** pair.

Discordant Pair

Compare a man who bought more than $100 worth of goods from the catalog and a woman who did not.

< $100	$100 +
	
P(100+) = .42	P(100+) = .32

The actual sorting disagrees with the model.
This is a **discordant** pair.

Tied Pair

Compare two women. One bought more than $100 worth of goods from the catalog, and the other did not.

< $100

P(100+) = .42

$100 +

P(100+) = .42

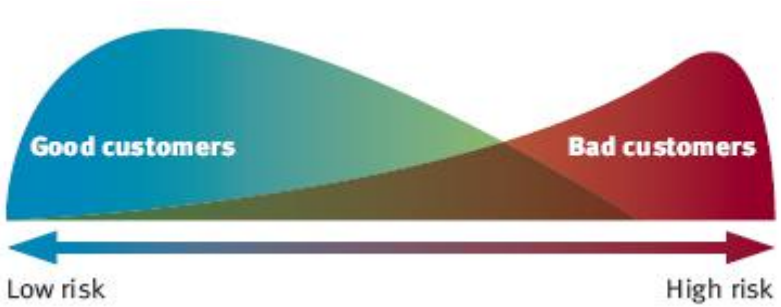
The model cannot distinguish between the two.
This is a **tied** pair.

Model: Concordant, Discordant, and Tied Pairs

- PROC Logistic standardně nabízí četnosti (relativní) jednotlivých typů párů a z nich odvozené statistiky kvality modelu:

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	30.1	Somers' D	0.107
Percent Discordant	19.5	Gamma	0.215
Percent Tied	50.4	Tau-a	0.050
Pairs	43578	c	0.553

6. Credit scoring- historie, základní pojmy



Úvod

- Credit Scoring je soubor prediktivních modelů a jejich základních technik, které slouží jako podpora finančním institucím při poskytování úvěrů.

- Tyto techniky rozhodují, kdo dostane úvěr, jaká má být výše úvěru a jaké další strategie zvýší ziskovost dlužníků vůči věřitelům.

- Credit Scoringové techniky kvantifikují a posuzují rizika při poskytování úvěrů konkrétnímu spotřebiteli.

Úvod

- Nerozeznají a nestanovují "dobré" nebo "špatné" (očekává se negativní chování, tj. např. default) žádosti o úvěr na individuální bázi, nýbrž poskytují statistické šance, nebo pravděpodobnosti, že žadatel s daným skóre se stane "dobrým" nebo "špatným".

- Tyto pravděpodobnosti nebo skóre, spolu s dalšími obchodními úvahami jako jsou předpokládaná míra schvalování, zisk nebo ztráty, jsou pak použity jako základ pro rozhodování o poskytnutí/neposkytnutí úvěru.

Why do we need score?

- “HISTORICAL EVOLUTION”:

Money lender

- lend only to people which he knows

Operators

- they make decision based on client's information and their experience

Automatic scoring

- make decision on statistical base

PAST EXPERIENCE -> ESTIMATION FOR FUTURE

Why score?

ADVANTAGES:

- Automatization of approval proces
- Cost – effective
- Less fraud possibilities

DISADVANTAGES

- Statistical based, not take in account client like individual

Úvod

- Zatímco historie úvěru sahá 4000 let nazpět (první zaznamenaná zmínka o úvěru pochází ze starověkého Babylonu - 2000 let před n.l.), historie credit scoringu je pouze 50-70 let stará.

- První přístup k řešení problému identifikace skupin v populaci představil ve statistice Fisher (1936). V roce 1941, Durand jako první rozpoznal, že tyto techniky mohou být použity k rozlišování mezi dobrými a špatnými úvěry.

Úvod

- Významným milníkem při posuzování úvěrů byla druhá světová válka.
- Do té doby bylo standardem individuální posuzování žadatele o úvěr. Dále bylo standardem, že ve finanční sféře byli zaměstnáni (téměř) výhradně muži.
- Odchod značné části mužské populace do služeb armády měl za následek potřebu předat zkušenosti dosavadních posuzovatelů žádostí o úvěr novým pracovníkům.
- Díky tomu vznikla jakási rozhodovací pravidla a došlo k „automatizaci“ posuzování žádostí o úvěr.

Úvod

- Příchod kreditních karet ke konci šedesátých let minulého století a růst výpočetního výkonu způsobil obrovský rozvoj a využití credit scoringových technik. Událost, která zajistila plnou akceptaci credit scoringu, bylo přijětí zákonů „Equal Credit Opportunity Acts" (o rovné příležitosti přístupu k úvěrům) a jeho pozdějších znění přijatých v USA v roce 1975 a 1976. Tyto stanovily za nezákonné diskriminace v poskytování úvěru, vyjma situace, pokud tato diskriminace „byla empiricky odvozená a statisticky validní".

Úvod

- V osmdesátých letech minulého století začala být využívána logistická regrese, dodnes v mnoha oblastech považovaná za průmyslový standard, a lineární programování. O něco později se objevily na scéně metody umělé inteligence, např. neuronové sítě. Mezi další používané techniky lze zařadit metody nejbližšího souseda, splajny, waveletové vyhlazování, jádrové vyhlazování, Bayesovské metody, regresní a klasifikační stromy, support vector machines, asociační pravidla, klastrová analýza a genetické algoritmy.

Historie -detail

Date	Event
2000 BC	First use of credit in Assyria, Babylon, and Egypt.
1100s	First pawnshops in Europe established by charitable institutions, and by 1350 they were being run as commercial concerns.
1536	Charging of interest deemed acceptable by the Protestant church.
1730	First advertisement for credit placed by Christopher Thornton of Southwark, London who offered furniture that could be paid off weekly.
1780s	First use of cheques in England.
1803	First consumer reports by Mutual Communications Society in London.
1832	First publication of the *American Railroad Journal*.
1841	Mercantile Agency is first American credit reporting agency.
1849	Harrod's established as one of the world's first department stores.
1851	First use of credit ratings for trade creditors by John M. Bradstreet.
1856	Singer Sewing Machines offers consumer credit.
1862	Poor's Publishing publishes *Manual of the Railroads of the United States*.
1869	First American consumer bureau is Retailers Commercial Agency (RCA) in Brooklyn.
1886	Sears established, and launches its catalogue in 1893.

Zdroj: Anderson

pawnshop = zastavárna
deemed acceptable = považován za přijatelný
Advertisement for credit = reklama na úvěr
Mercantile agency = obchodní agentura

Historie -detail

Date	Event
1906	National Association of Retail Credit Agencies formed in the USA.
1909	John M. Moody publishes first credit rating grades for publicly traded bonds.
1913	Henry Ford uses production lines to produce affordable automobiles.
1927	Establishment of Schufa Holdings AG, first credit bureau in Germany.
1934	First public credit registry (PCR) established in Germany.
1936	R.A. Fisher's use of statistical techniques to discriminate between iris species.
1941	David Durand writes report, suggesting statistics can assist credit decisions.
194?	Henry Wells uses credit scoring at Spiegel Inc.
1950	Diners Club and American Express launch first charge cards.
1950s	Sears uses propensity scorecards for catalogue mailings.
1956	FI consultancy established in California, USA.
1958	First use of application scoring by American Investments.
1960s	Widespread adoption of credit scoring by credit card companies.
1966	Credit Data Corp. becomes first automated credit bureau.
1970	Fair Credit Reporting Act governs credit bureaux.
1974	Equal Credit Opportunity Act causes widespread adoption of credit scoring.
1975	FI implements first behavioural scoring system for Wells Fargo.
1978	Stannic implements first vehicle finance scorecards in South Africa.
1982	CCN offers Credit Account Information Sharing (CAIS), its consumer credit bureau service.
1984	FI develops first bureau scores used for pre-screening.
1987	MDS develops first bureau scores used for bankruptcy prediction.
1995	Mortgage securitisers Freddy Mac and Fannie Mae adopt credit scoring.
2000	Moody's KMV introduces RiskCalc for financial ratio scoring (FRS).
2000s	Basel II implemented by many banks.

Zdroj: Anderson

affordable = dostupný
iris species = druhy kosatců
Charge card = kreditní karta
Propensity scorecard = scoringová karta pro modelování náchylnosti (k nákupu)
FI = sploločnost Fair, Isaac…dnes FICO
Mortgage = hypotéka

Historie -detail

Table 2.4. Genealogies and milestones—credit cards

Date	Event
1914	Western Union introduces embossed metal plate first charge card in the United States.
1920s	Introduction of 'shopper's plates', early version of modern store cards.
1950	Diners Club and American Express launch first charge cards.
1951	Diners Club launches first credit card in New York city.
1960	Bank Americard established, later to become Visa.
1966	Master Charge established, later to become MasterCard.
1966	Barclaycard established in the United Kingdom.

Table 2.5. Genealogies and milestones—credit scoring consultancies

Name	Year	Notes
Fair Isaac (FI)		
FI	1956	Founded San Francisco CA, by Bill Fair and Earl Isaac
	1958	First scorecard development, for American Investments
	1984	Develops first bureau score for pre-screening
	1995	First use of scoring by mortgage securitisers
Experian-Scorex		
Management Decision Systems (MDS)	1974	Founded by John Coffman and Gary Chandler
	1982	MDS purchased by CCN
Scorex	1984	Founded in Monaco by Jean-Michel Trousse
MDS	1987	MDS develops first monthly bureau score, for bankruptcy
Experian-Scorex	2003	Created as subsidiary of Experian, after purchase of Scorex

Zdroj: Anderson

Historie -detail

Table 2.7. Genealogies and milestones—credit bureaux

Name	Year	Notes
Dun & Bradstreet		
Mercantile Agency	1841	Founded, New York NY, by Lewis Tappan.
	1849	Benjamin Douglass takes over, and expands.
John M. Bradstreet Co.	1849	Founded, Cincinnati OH.
	1851	First use of credit rating grades.
R.G. Dun & Co.	1859	Robert G. Dun incorporates Mercantile Agency.
Dun & Bradstreet	1933	Merger orchestrated by Arthur Whiteside.
Experian		
Manchester Guardian Society	1827	Founded, Manchester, UK.
Chilton Corp.	1897	Founded, Dallas TX. Publishes 'Red Book'.
Michigan Merchants	1932	Founded, later to become Credit Data Corp.
TRW	1968	Purchases Credit Data Corp., and changes name to TRW-Credit Data.
TRW	1976	Information Systems and Services (IS&S) division produces first business credit report.
CCN	1980	Founded, when Great Universal Stores (GUS) spins off information services division
	1884	Purchases Manchester Guardian Society
TRW	1989	Purchases Chilton Corp.
Experian	1996	Founded, through TRW divestiture of TRW-CD & IS&S. Purchased by GUS, who merges it with CCN.
Equifax		
London Assn. for the Protection of Trade	1842	Founded, London, UK
RCA	1869	Founded, Brooklyn, NY
RCC	1899	Founded, Atlanta, GA
	1934	Purchases RCA
United Assn. for the Protection of Trade	1965	LAPT renamed
Equifax	1975	RCC renamed to Equifax
	1994	Purchases UAPT-Infolink and Canadian Bonded Credits
TransUnion		
TransUnion	1968	Founded, as holding company for Union Tank Car Company (UTCC)
	1969	Purchases the Credit Bureau of Cook County

Zdroj: Anderson

Historie -detail

Table 2.8. Genealogies and milestones—credit rating agencies

Name	Year	Notes
Standard & Poor's (S&P)		
Poor's Publishing Co.	1862	Founded, by Henry Varnum Poor
S&P	1941	Poor's Publishing and Standard Statistics merge
Moody's Investor Services (MIS)		
John Moody & Co.	1900	Founded, by John Moody, but fails in 1907
John Moody	1909	First use of rating grades for bonds
Moody's Investor Services	1914	Incorporation of MIS
	1962	MIS purchased by D&B
Moody's KMV	2002	Created as MIS subsidiary after merger of Risk Management Services and KMV
Fitch IBCA		
Fitch Publishing Co.	1913	Founded, by John Knowles Fitch
IBCA	1978	Founded
Fitch IBCA	1997	Merger of Fitch Publishing and IBCA

Zdroj: Anderson

Historie –další zajímavé čtení

http://www.fundinguniverse.com/company-histories/Fair-Isaac-and-Company-Company-History.html

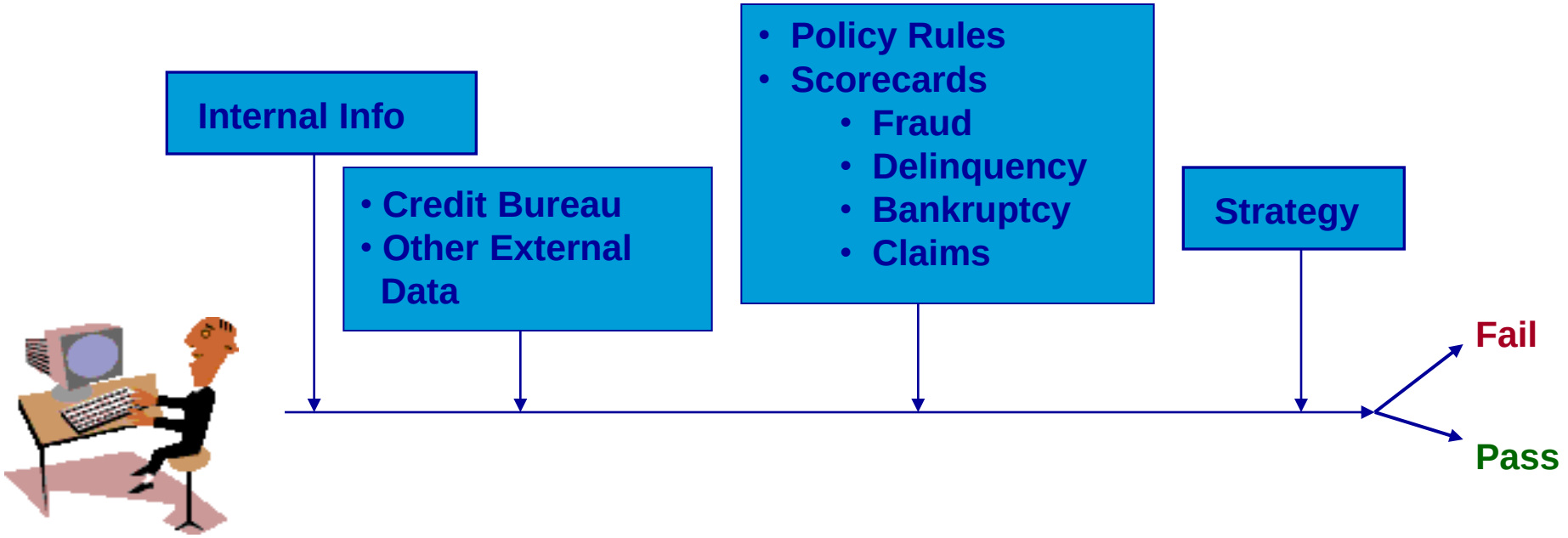
http://www.fico.com/en/Company/News/Pages/03-10-2009.aspx

http://www.directlendingsolutions.com/history_credit_scoring.htm

http://www.pbs.org/wgbh/pages/frontline/shows/credit/more/scores.html

http://en.wikipedia.org/wiki/Credit_score

Risk Management – Acquisition



Data Acquisition

Risk Management – Customer

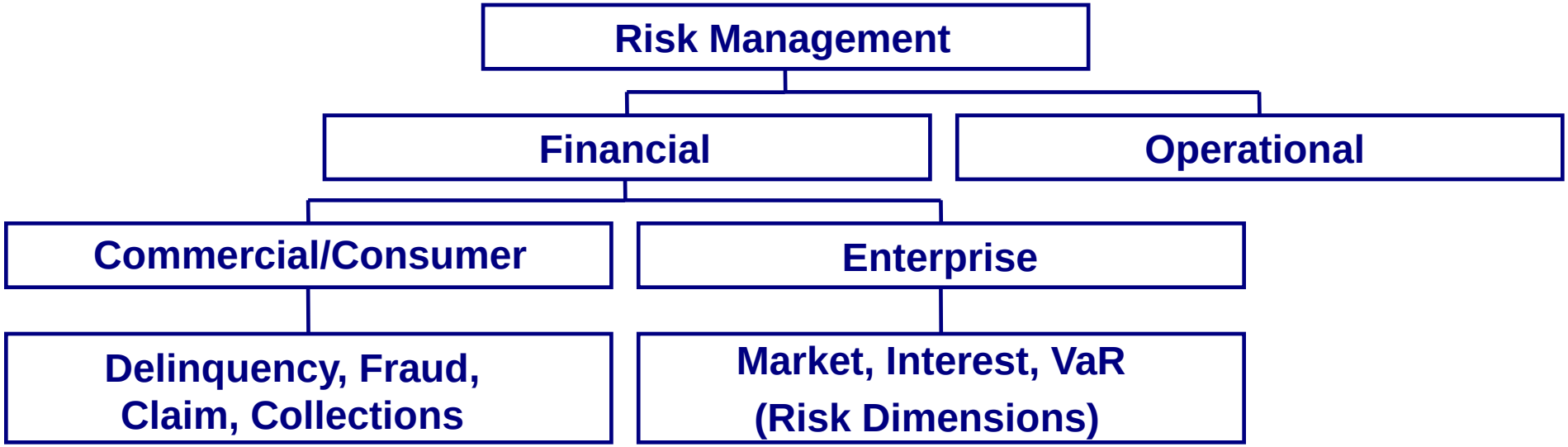
- Credit Line Management
- Usage Monitoring
- Transaction Fraud
- Transaction Approval
- Renewal/Reissue

- Collections
- Claims

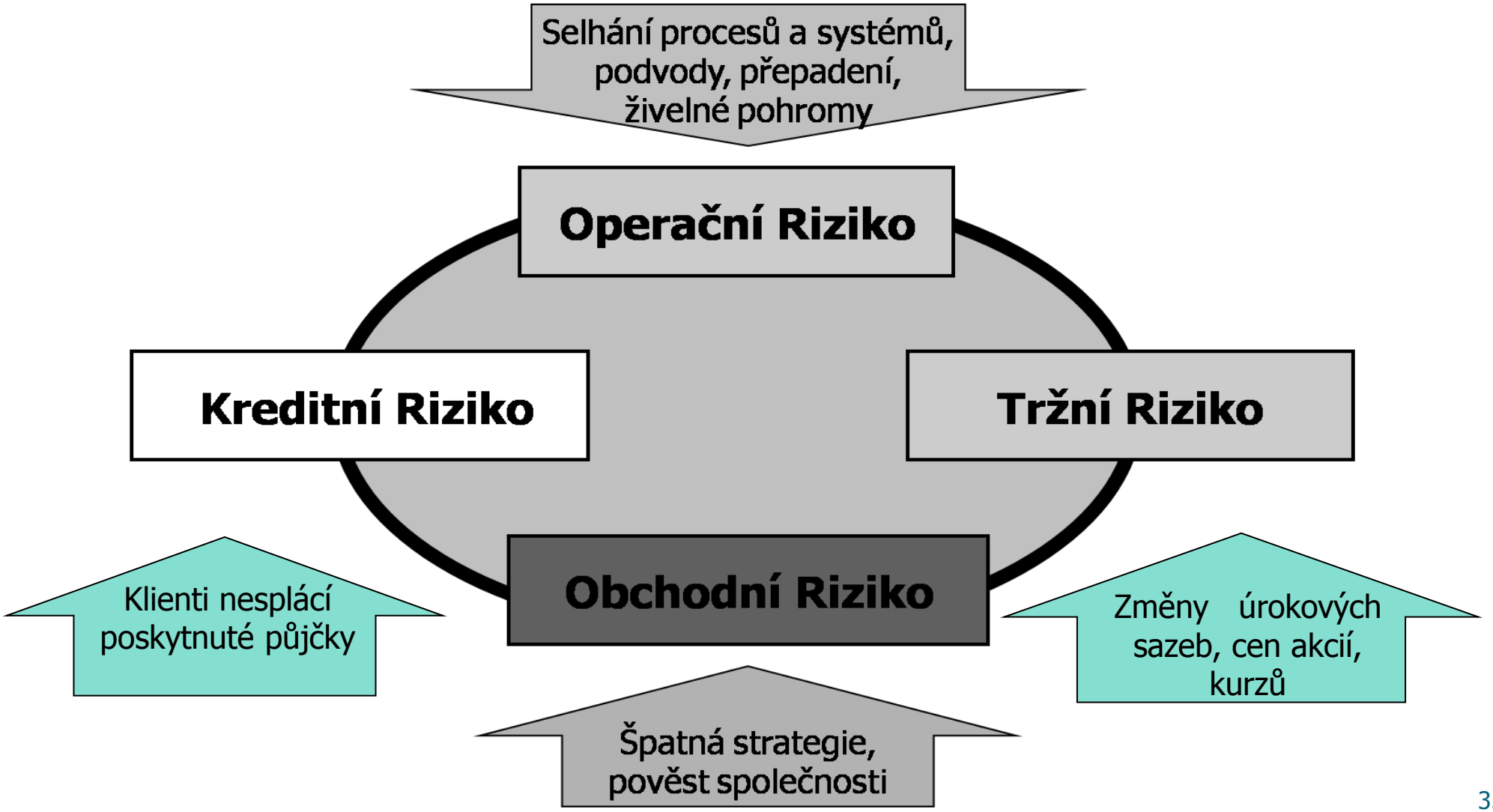
√ Scorecards
√ Policy Rules
√ Strategies

.. Lots of analysis

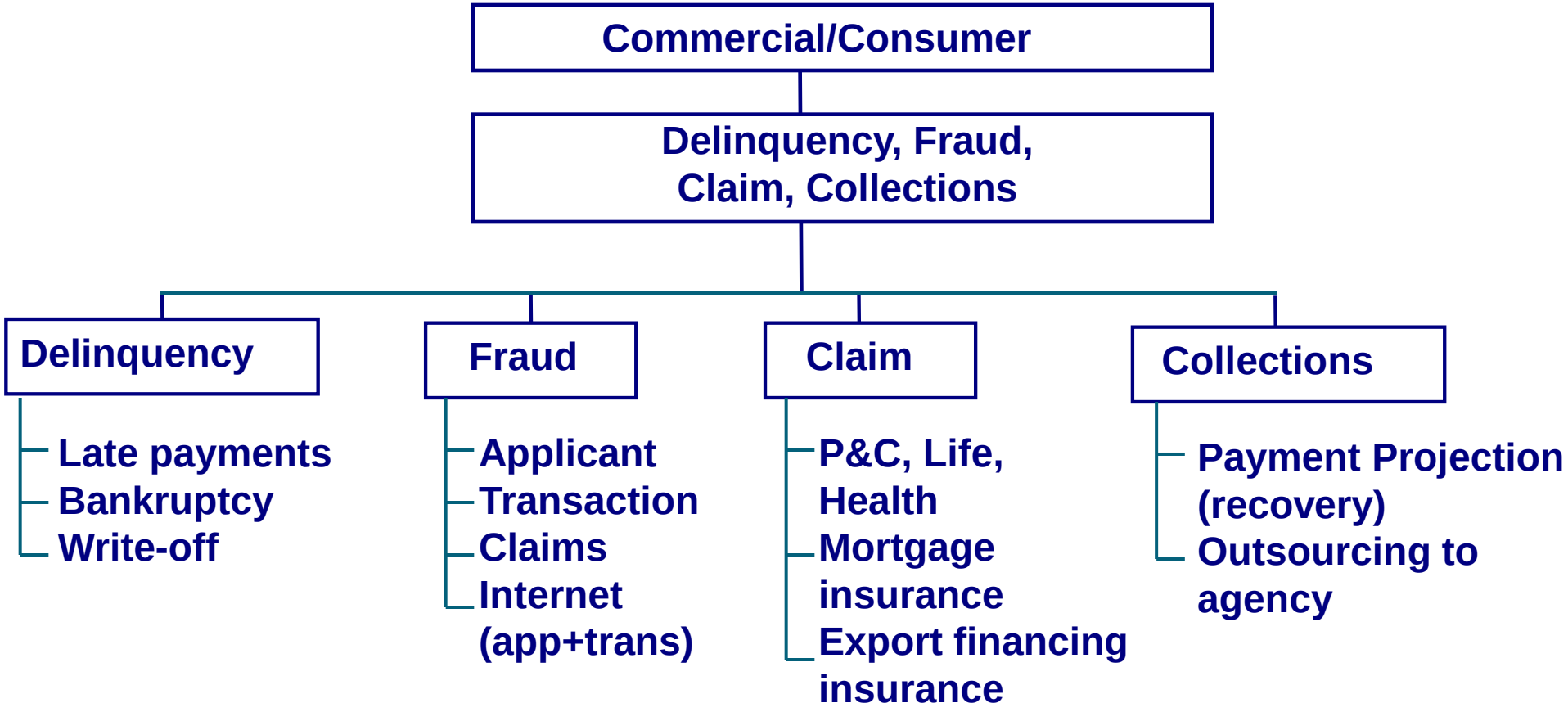
Risk Management



Risk Management a druhy rizik



Risk Management



P&C: Property & Casualty Insurance (majetkové a úrazové pojištění)

Why Manage Risk?

- Reduce exposure to high-risk accounts.
- Decrease bad debt and claims payouts.
- Ensure better pricing to reflect risk.
- Detect fraud early-on.
- Increase approval rates (the "right kind" – potentially increasing revenue).
- Handle most approvals/declines quickly (customer service).
- Analysts/investigators only focus on difficult accounts.
- Ensure consistent, equal and objective treatment of each applicant across the organization.
- Offer more efficient marketing initiatives.

Users of Risk Management

- Banks
  - Citibank, Royal Bank, CIBC, BankOne
- Finance Companies
  - GE Capital, HFC, GMAC
- Insurance
  - Life, Property and Casualty, Health
- Government
  - Ministries/Departments of Health (Medicare), Ministries of Finance (IRS), Workers Compensation.

Users of Risk Management

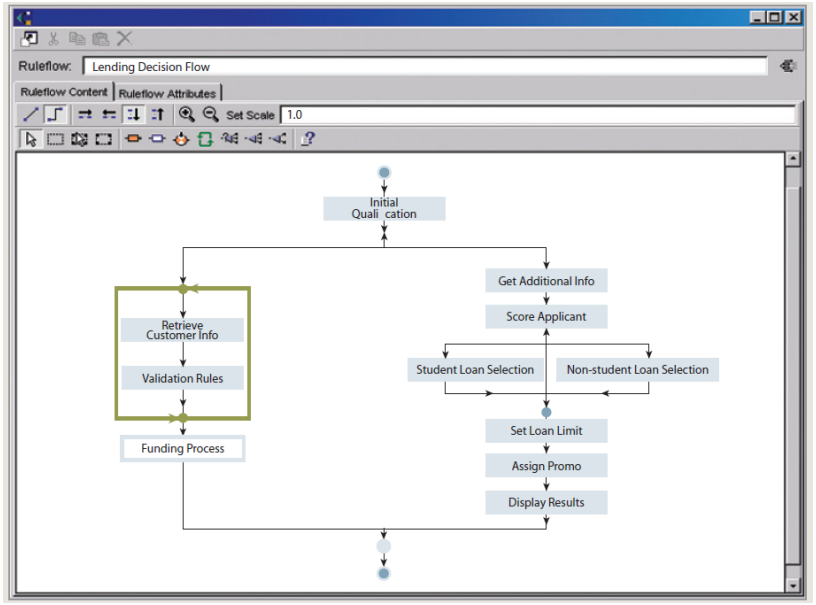
- Utilities
  - Hydro/Power/Energy, Water
- Communications
  - Bell, Sprint, AT&T (land lines and cellular)
- Retail
  - JC Penneys, Sears, Hudsons Bay Company, Target
- Manufacturers/Industrials
  - Those who give credit to small businesses.

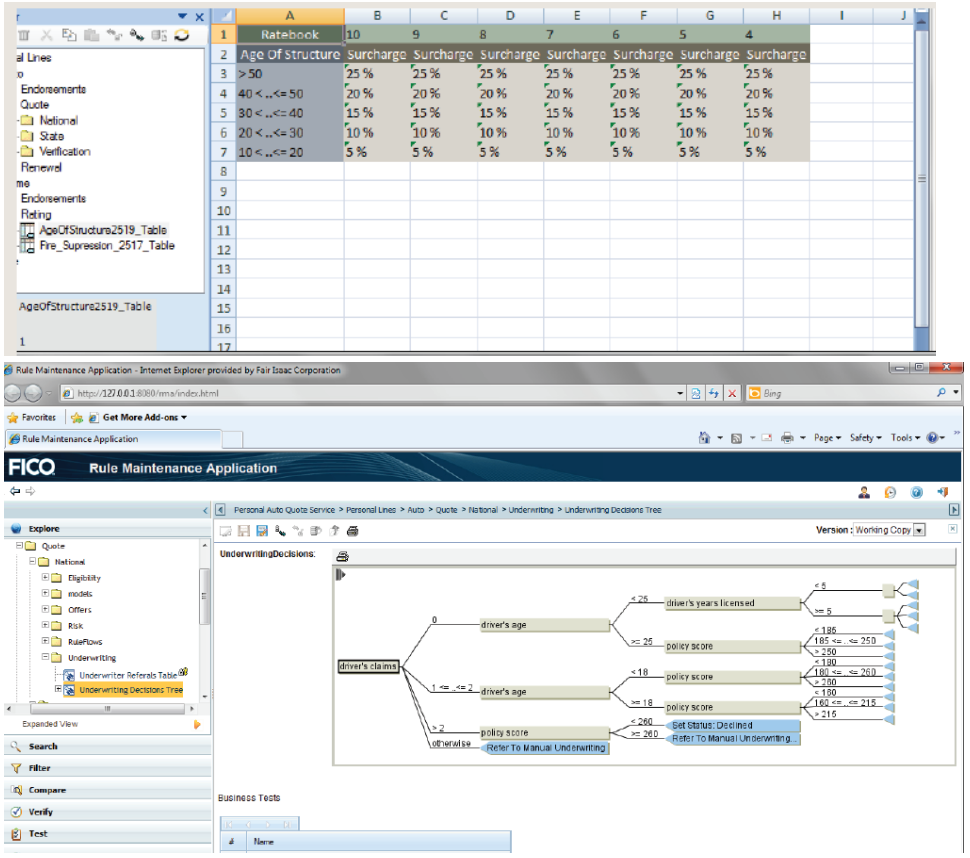
Risk Management "Toolbox"

- Risk Data Mart/Data Warehouse
- Risk prediction models (scorecards)
- Reporting
- Analysis tools

- Operational/strategy implementation software (for example, FICO™ Blaze Advisor®, FICO® TRIAD® Customer Manager, Experian Probe SM, Experian NBSM, Cardpac, VisionPlus, Pro-Logic Ovation).

FICO™ Blaze Advisor®



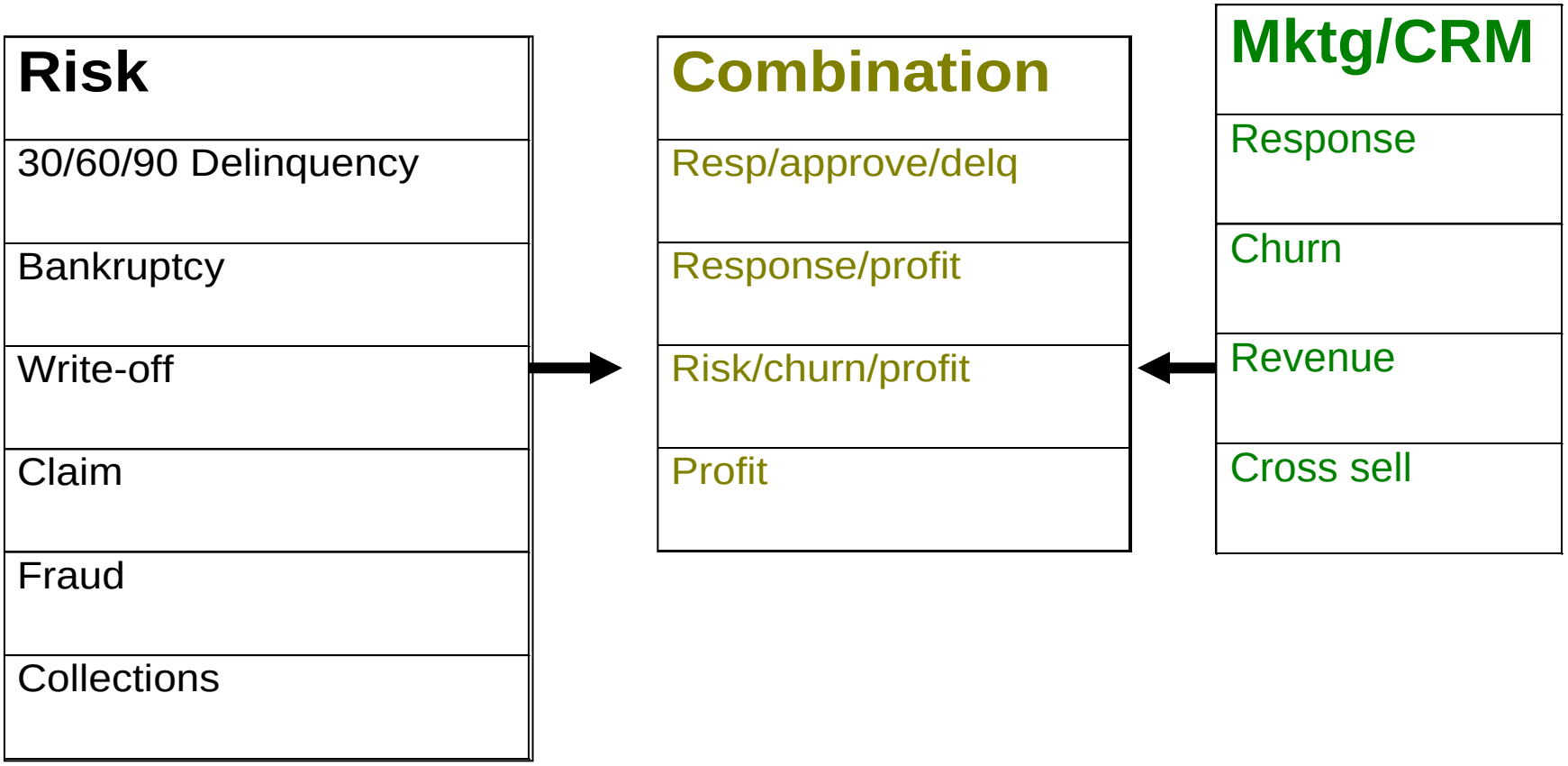


Zdroj: http://www.fico.com/account/resourcelookup.aspx?theID=430

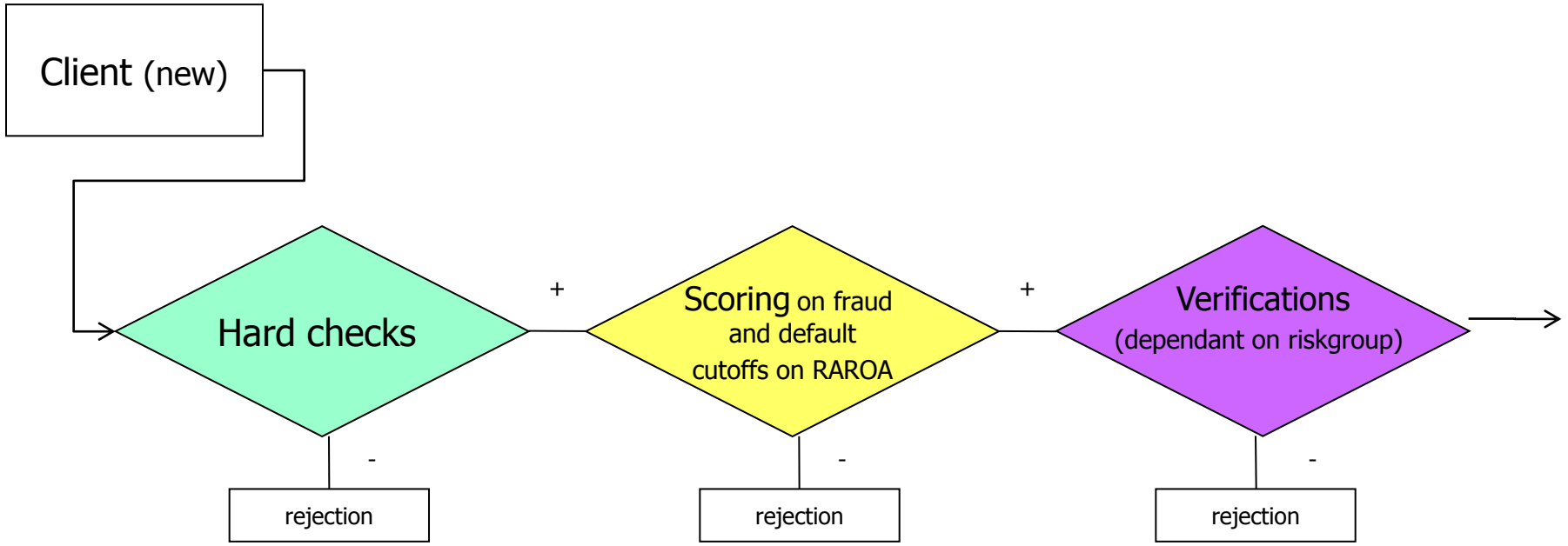
Scorecards

- Predict the probability of a negative event.
  - Custom – based on clients own data
  - Generic – based on pooled industry or bureau data (Beacon, Empirica)
  - Application – new applicants
  - Behavioral – current customers

Scorecard Types



Scoring in approval process



Policy declines – low age, unsufficient length of employment, "terorrist" etc.

What is the probability that client will pay?

Will the contract be profitable?

Is the number of client's phone valid? Etc.

Fraud Risk

- Fraud risk is one of the fastest growing areas in risk management.
- Examples include bank/retail card fraud, insurance fraud, health care fraud, welfare fraud, franchise fraud, internet fraud, mortgage fraud, investment fraud, tax fraud, merchant fraud.
- E-commerce presents opportunities.
- The F.B.I. estimates that between 10–15% of loan applications contain material misrepresentations.

Reporting and Analysis

- Scorecard and portfolio performance
- Approval rates, applicant profile, loss rates, high risk segments
- Behavior tracking to develop better strategies
- Capturing fraud, approval/decline, pricing, credit line management, collections, cross sells qualification, claims.

Risk Applications

- Retail/banking (consumer and commercial)
  - Application and behavior scorecards for all credit products.
  - Strategy design for credit limit setting, authorizations and collections/reissue/suspension.
  - Fraud application and transaction detection
  - Pricing/down payment
  - ATM limits, check holds
  - Pre-qualifying direct marketing lists.
- Automotive/finance
  - Loans and leasing
    - Application, behavioral, fraud, collection scorecards
  - Pricing/down payment.

Risk Applications

- Government
  - Fraud detection (for example, Welfare, health insurance)
  - Entitlement/claims assessment (for example, Workers compensation)
- Communications
  - Security deposit
  - International call access
  - Contract/"pay as you go"
  - Telephone fraud
  - "Shadow limit" setting
  - Suspension of service
  - Collections.

Risk Applications

- Insurance
  - Rate setting
  - Fraud detection
  - Claims management
  - Risk control for CRM initiatives.
- Utilities
  - Security deposit
  - Collections.

Risk Applications

- Manufacturers/pharmaceuticals/industrials
  - Assessing credit risk of business clients
  - Credit risk assessment of franchisees (for example, gas stations)
  - Payment terms
  - Collections
  - Merchant fraud.

Risk Applications

- Optimizing work flow in adjudication departments
- Evaluating/pricing portfolios
- Securitization
- Setting economic/regulatory capital allocation
- Reducing turnaround time (automated scoring)
- Comparing quality of business from different channels/regions/suppliers.

Resources

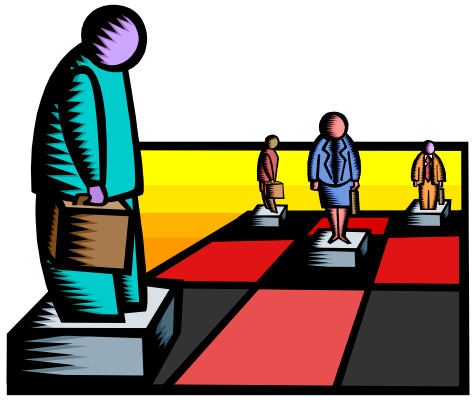
- www.ftc.gov/bcp/conline/pubs/credit/scoring.htm
- www.creditscoring.com
- www.my-credit-score.com
- www.fairisaac.com, www.myfico.com
- www.experian.com
- www.creditinfocenter.com
- www.consumersunion.org/finance/scorewc200.htm
- www.phil.frb.org/files/br/brs097lm.pdf
- www.nacm.org
- www.rmahq.org
- www.riskmail.org
- www.occ.treas.gov

Resources

- **Credit Scoring & Its Applications**
  by Lyn Thomas, Jonathan Crook, David Edelman
- **Credit Risk Modeling: Design and Application**
  by Elizabeth Mays (Editor)
- **Internal Credit Risk Models: Capital Allocation and Performance Measurement**
  by Michael K Ong
- **Handbook of Credit Scoring**
  by Elizabeth Mays
- **Applications of Performance Scoring to Accounts Receivables Management in Consumer Credit**
  by John Y. Coffman
- **Introduction to Credit Scoring,**
  by E.M. Lewis

Scorecard Development roles-objectives

- Understand the critical resources needed to successfully complete a scorecard development and implementation project.
- Understand some of the operational considerations that go into scorecard design.



Major Roles

- Scorecard Developer
  - Data miner, data issues
- Credit Scoring Manager/Risk Manager
  - Strategic view, corporate policies, implementation
- Product Manager
  - Client base, target market, marketing direction.

Major Roles

- Operational Managers
  - Customer Service, Adjudication, Collections
  - Strategy execution, impact on customers
- IT/IS Managers
  - external/internal data, implementation platforms.

Minor Roles

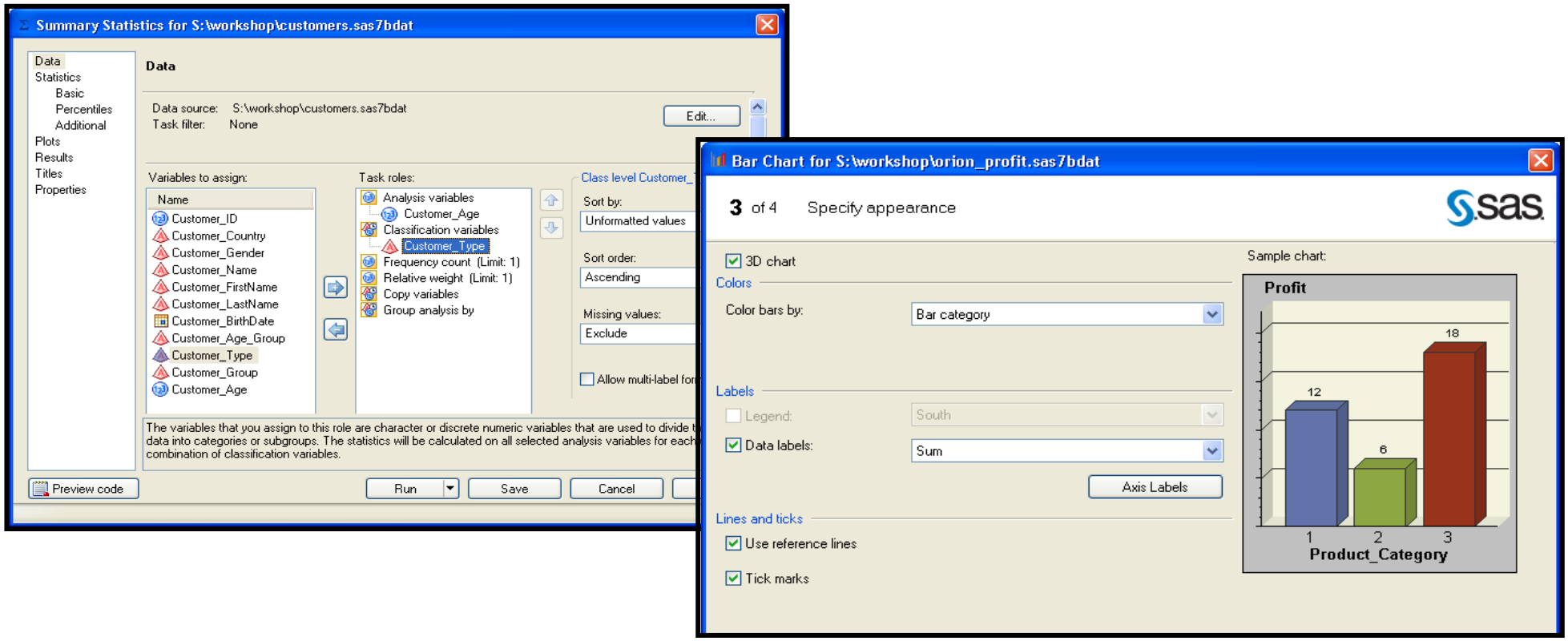
- Project Manager
  - Coordination, time lines
- Corporate Risk staff
  - Corporate policies, capital allocation
- Legal.

Why All of These Roles?

- Can I use this variable?
  - Legal, technical (derived variables, implementation platform), future application form design
- Segmentation
  - Marketing, application form design, systems
- What is the impact on this segment?
  - Operational, marketing, risk manager, corporate risk.

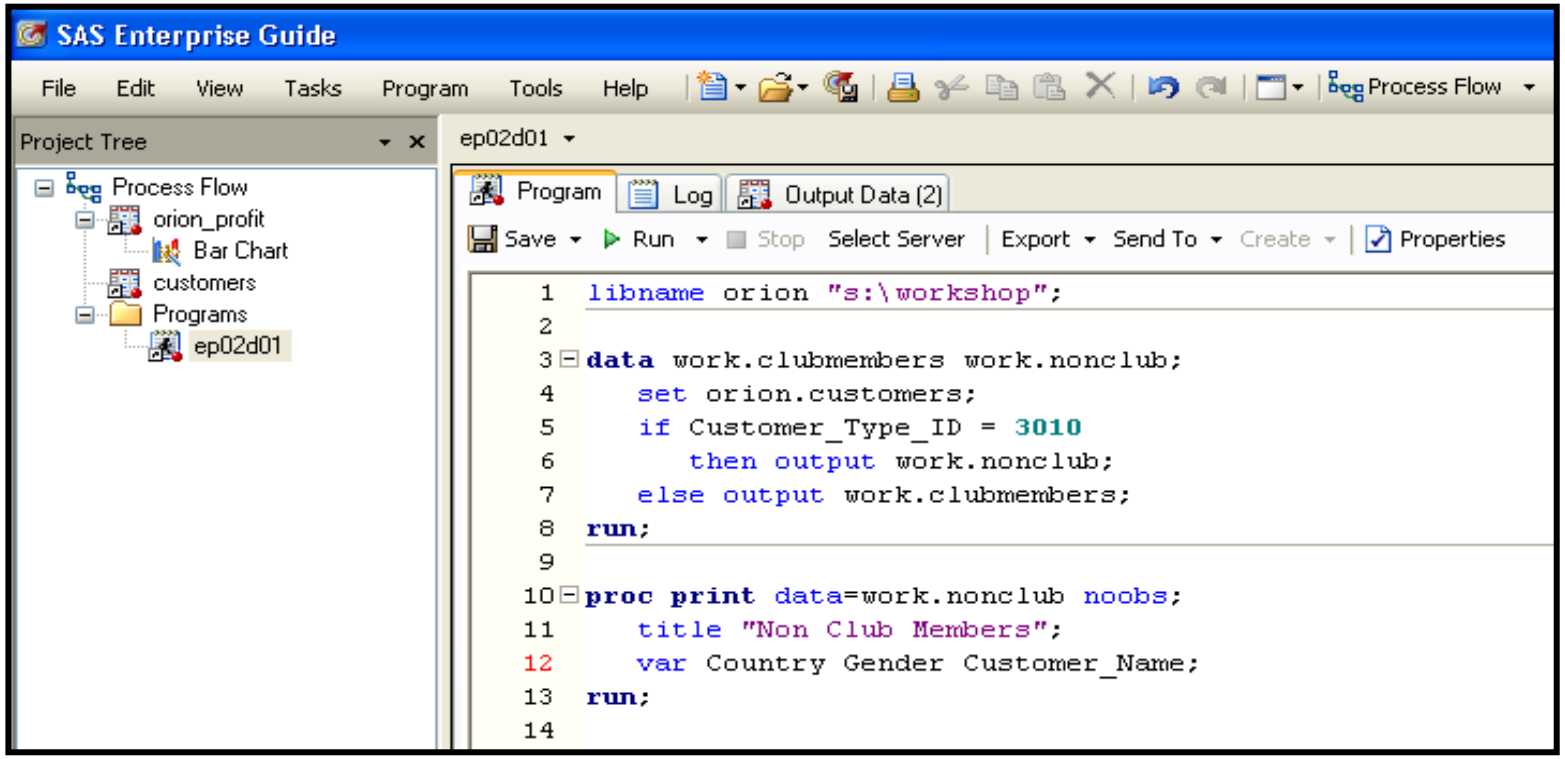
Introduction to SAS Enterprise Guide

- SAS Enterprise Guide provides a point-and-click interface for managing data and generating reports.



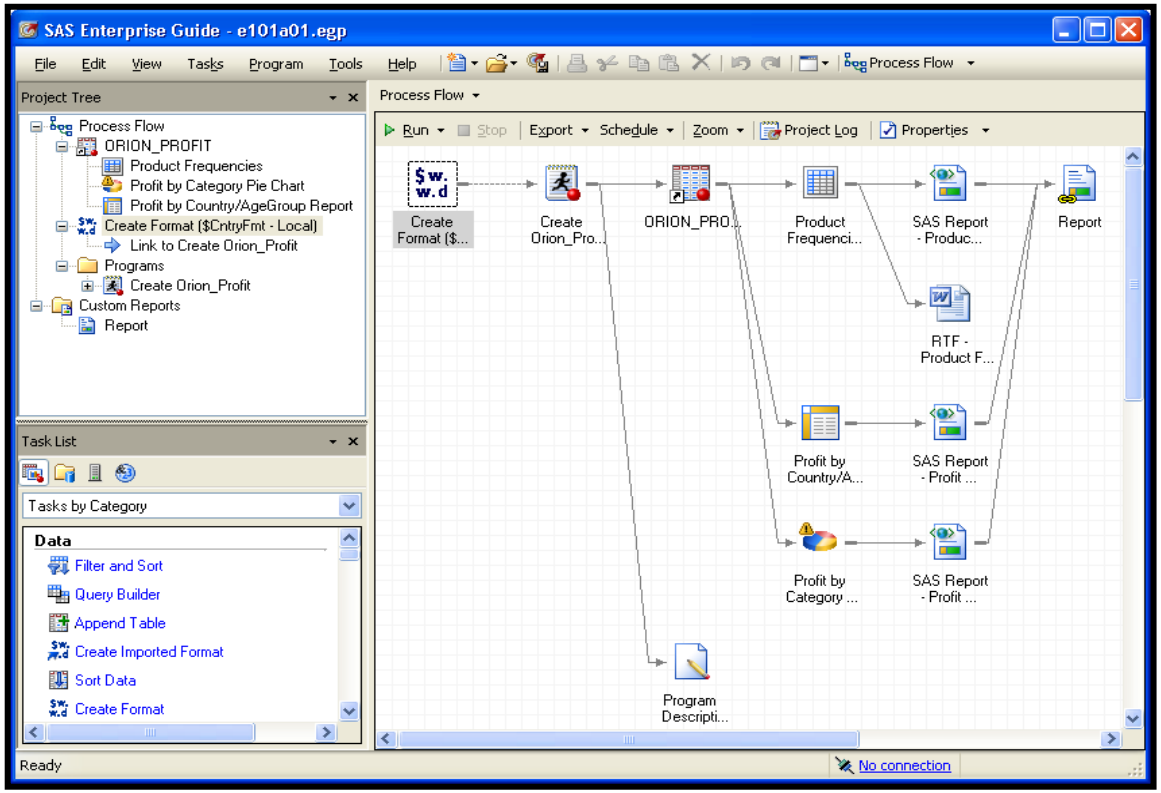
SAS Enterprise Guide Interface

- SAS Enterprise Guide also includes a full programming interface that can be used to write, edit, and submit SAS code.



SAS Enterprise Guide Interface: The Project

- A project serves as a collection of
  - data sources
  - SAS programs and logs
  - tasks and queries
  - results
  - informational notes for documentation.



You can control the contents, sequencing, and updating of a project.

SAS Programs

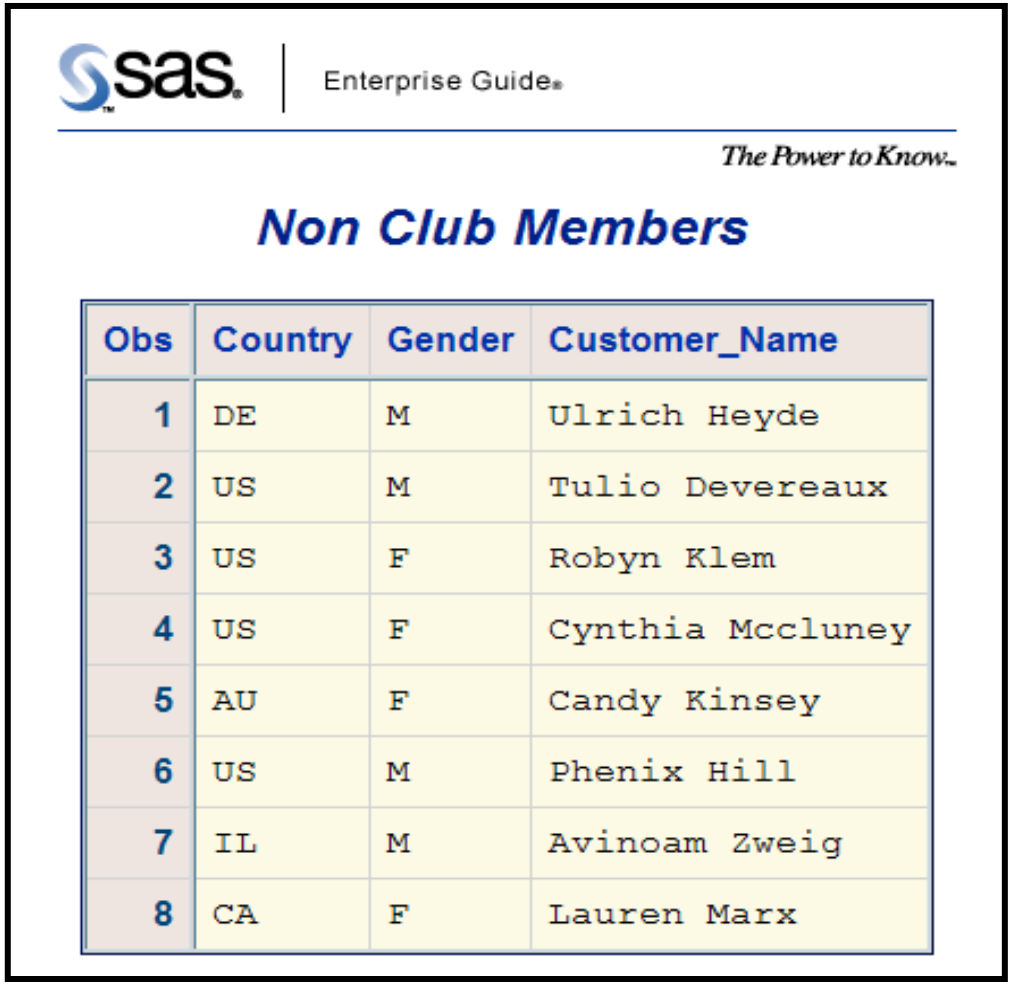
```
data work.clubmembers work.nonclub;
   set orion.customer;
   if Customer_Type_ID = 3010
      then output work.nonclub;
   else output work.clubmembers;
run;

proc print data=work.nonclub;
   title "Non Club Members";
   var Country Gender Customer_Name;
run;
```

DATA Step

PROC Step

ep02d01.sas

PROC PRINT Output

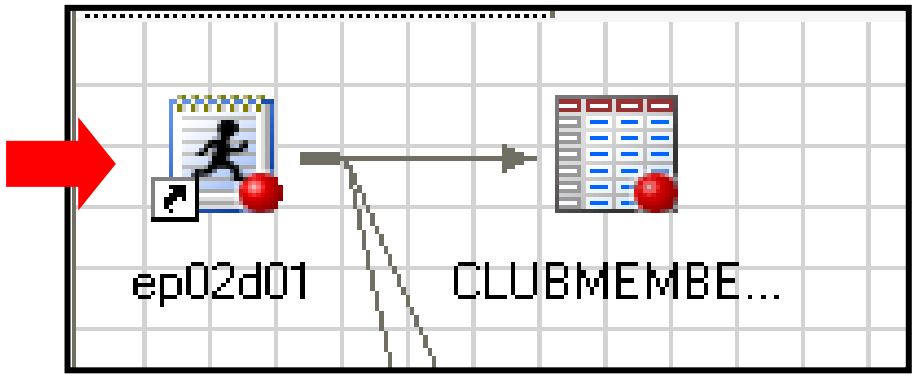


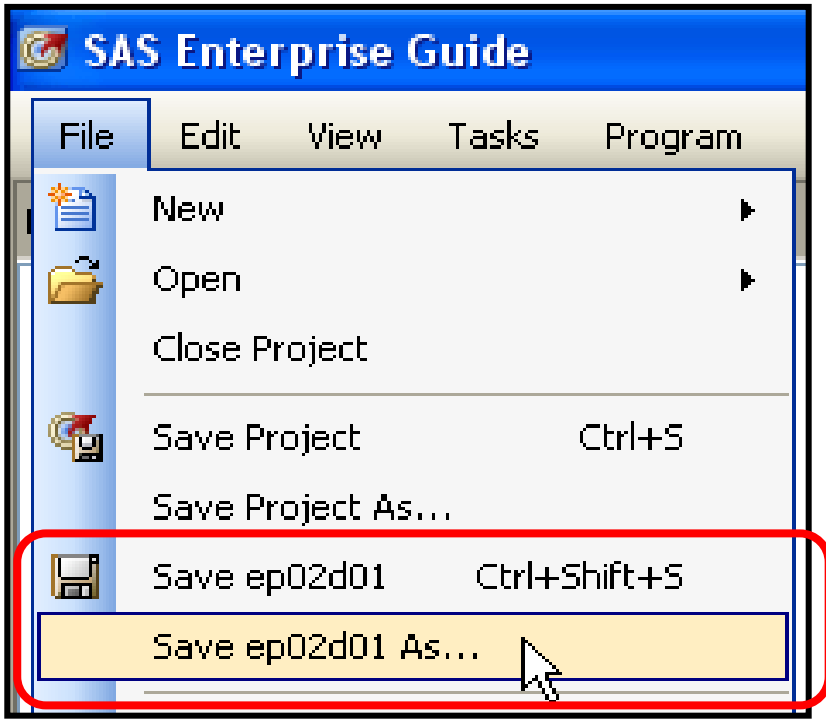
Non Club Members

Obs	Country	Gender	Customer_Name
1	DE	M	Ulrich Heyde
2	US	M	Tulio Devereaux
3	US	F	Robyn Klem
4	US	F	Cynthia Mccluney
5	AU	F	Candy Kinsey
6	US	M	Phenix Hill
7	IL	M	Avinoam Zweig
8	CA	F	Lauren Marx

Saving SAS Programs

- The SAS program in the project is a shortcut to the physical storage location of the .sas file. Select the program icon and then select **File** ⇨ **Save *program name*** to save the program as the same name, or **Save *program name* As...** to choose a different name or storage location.

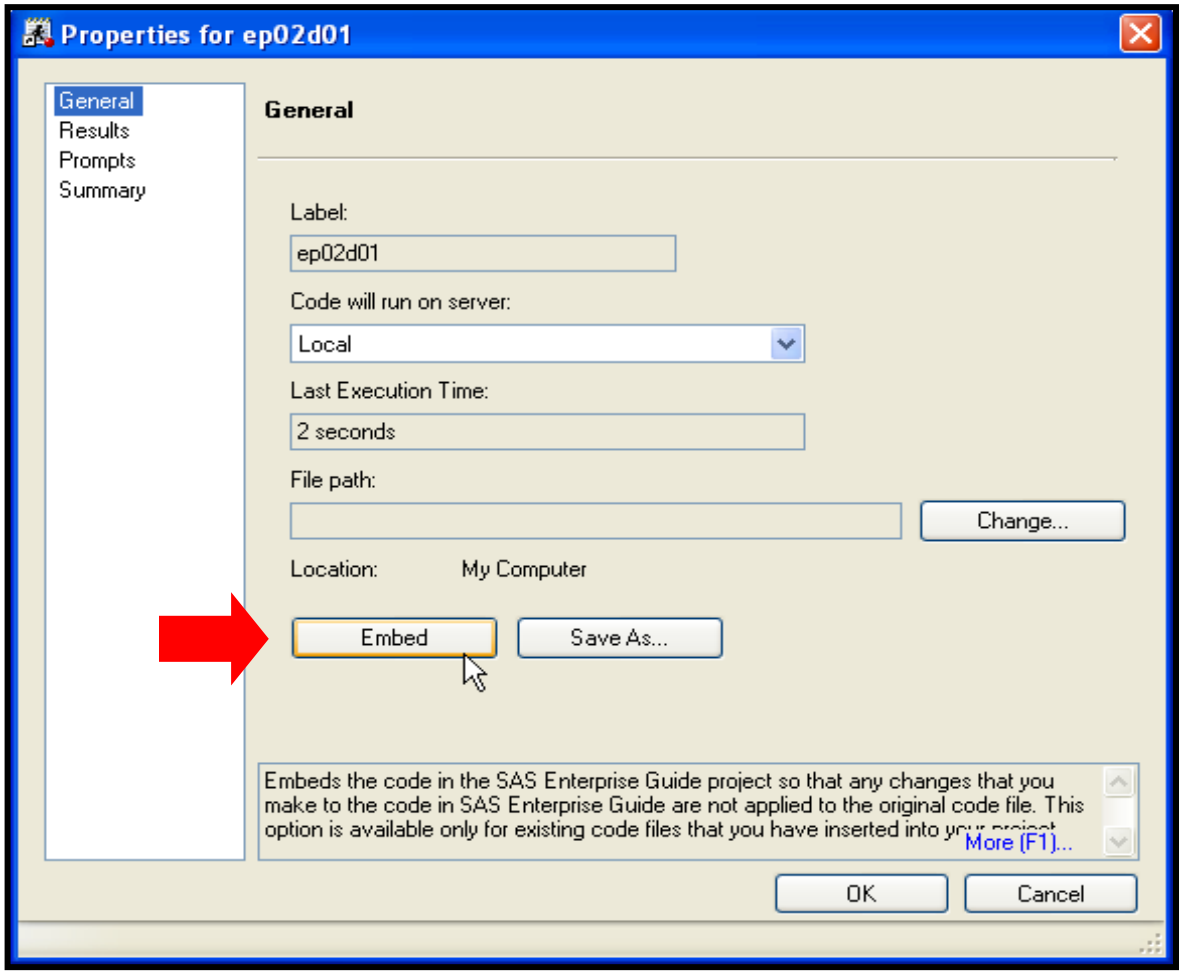




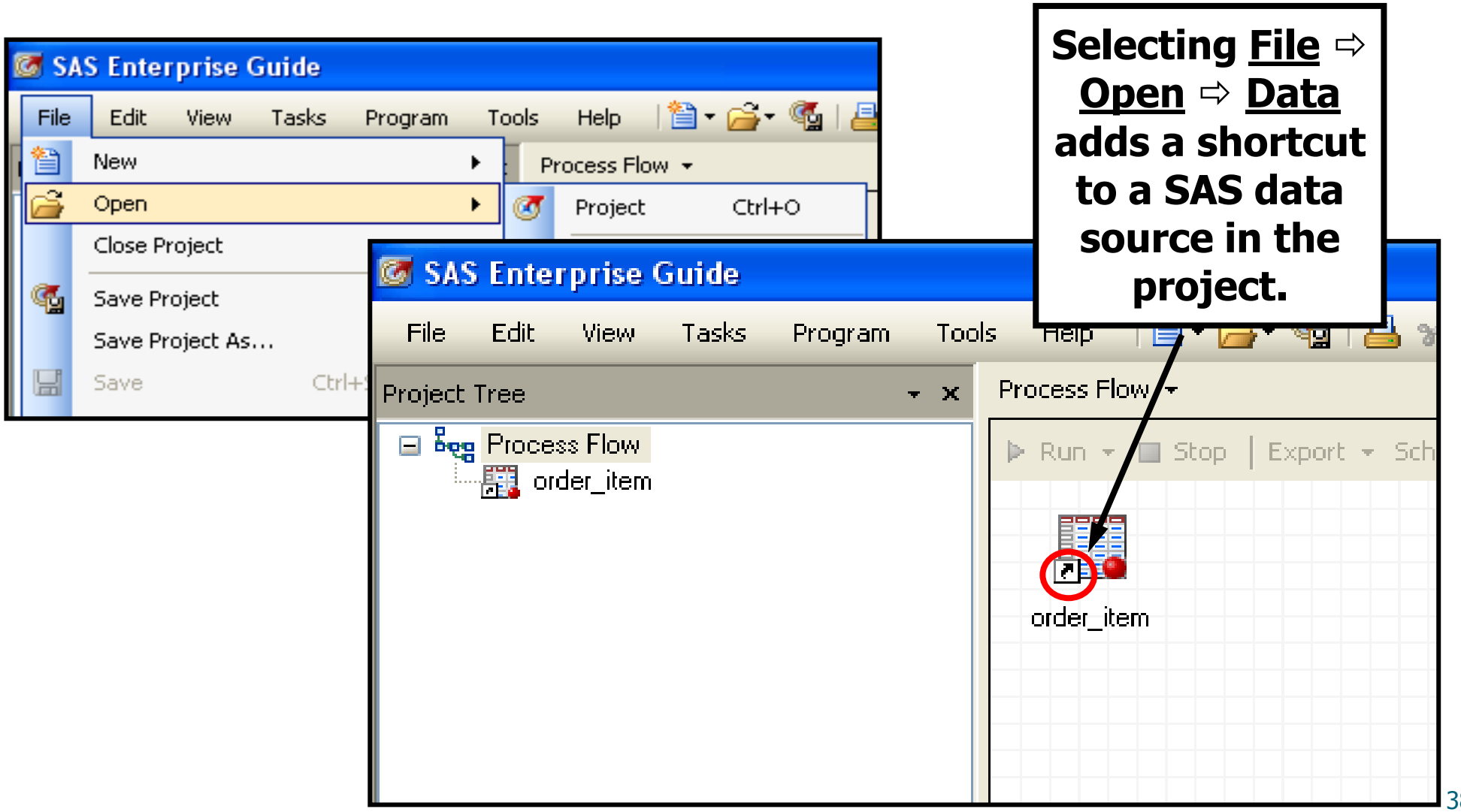
Embedding Programs in a Project

- A SAS program can also be embedded in a project so that the code is stored as part of the project .epg file.

- Right-click on the **Code** icon in a project and select **Properties** ⇨ **Embed**.

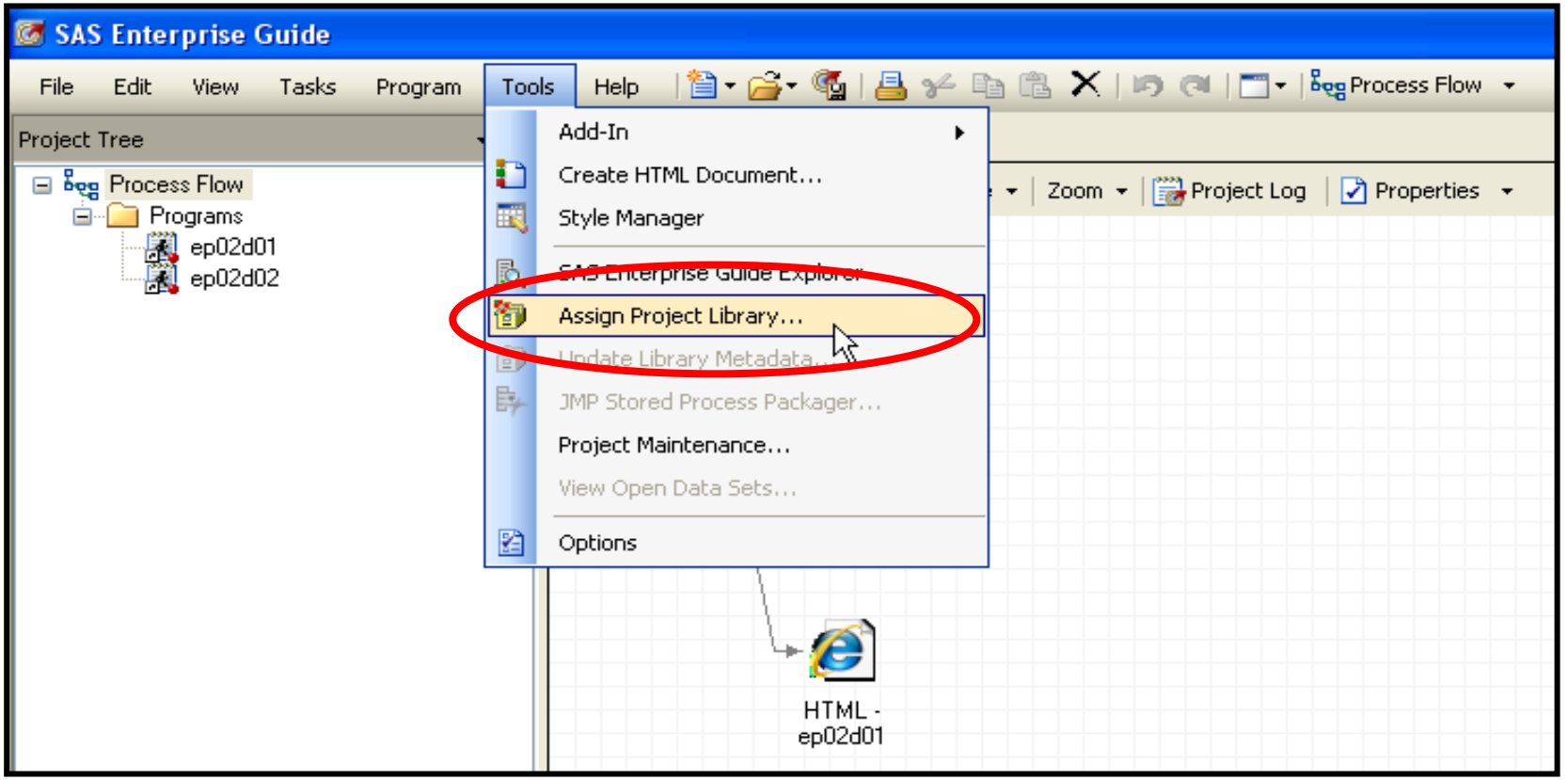


How Do You Include Data in a Project?



Assigning a Libref

- You can use the Assign Project Library task to define a SAS library for an individual project.

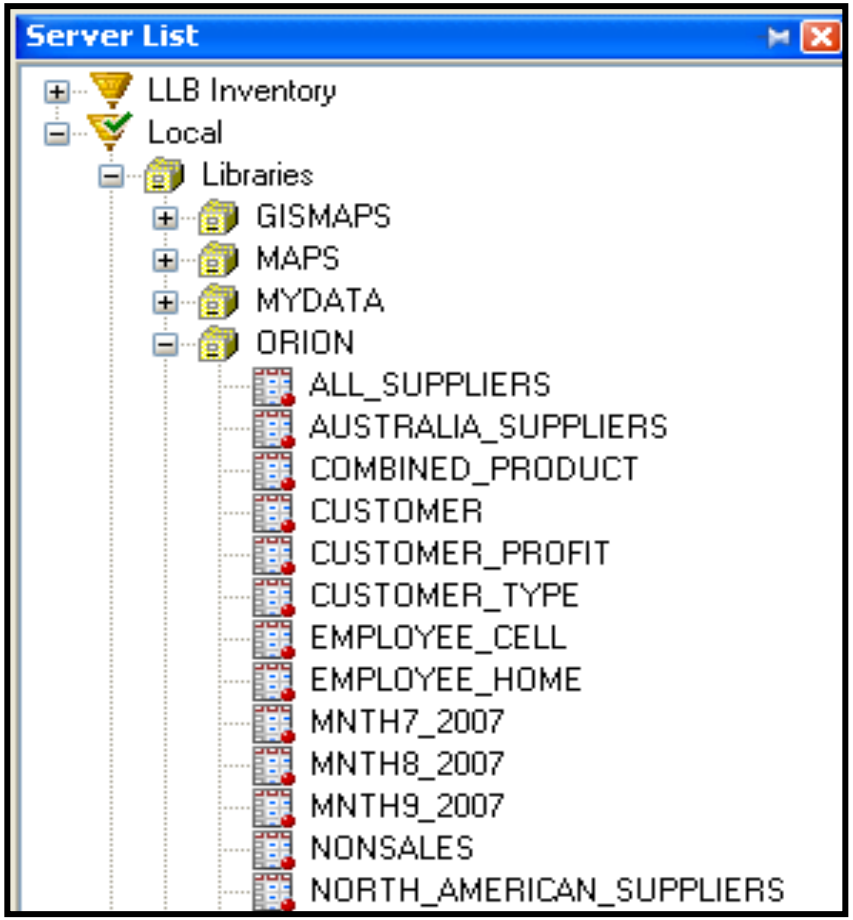


Browsing a SAS Library

- During an interactive SAS Enterprise Guide session, the Server List window enables you to manage your files in the windowing environment.

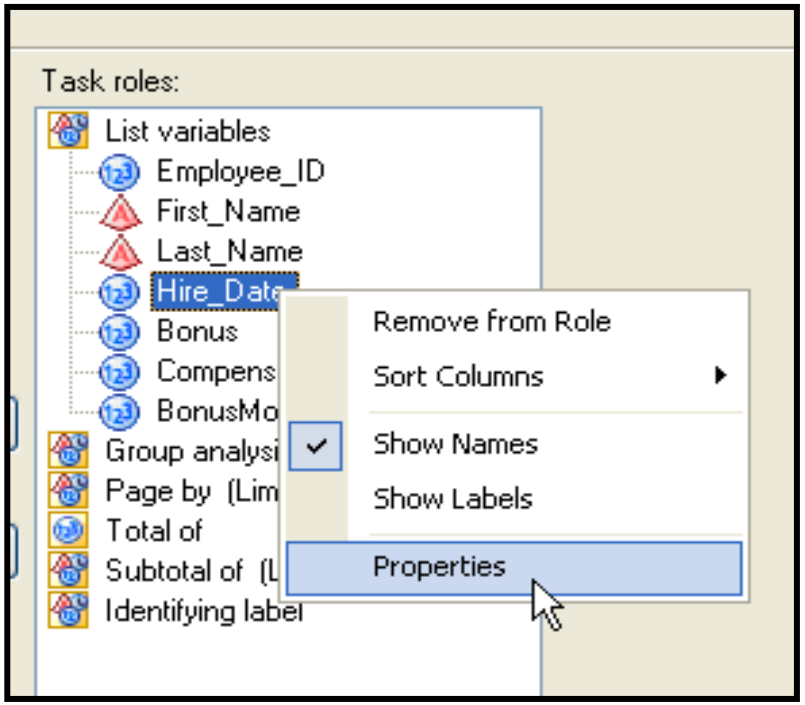
In the Server List window, you can do the following:

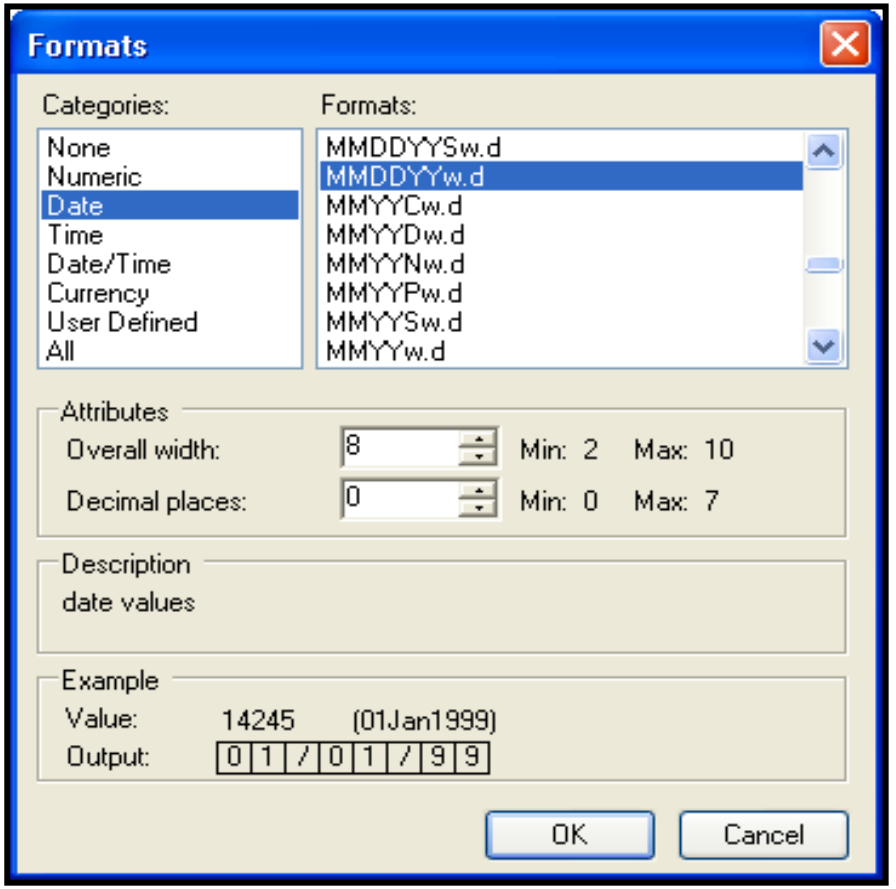
- view a list of all the servers and libraries available during your current SAS Enterprise Guide session
- drill down to see all tables in a specific library
- display the properties of a table
- delete tables
- move tables between libraries



Applying Formats

- Display formats can be applied in a SAS Enterprise Guide task or query by modifying the properties of a variable.

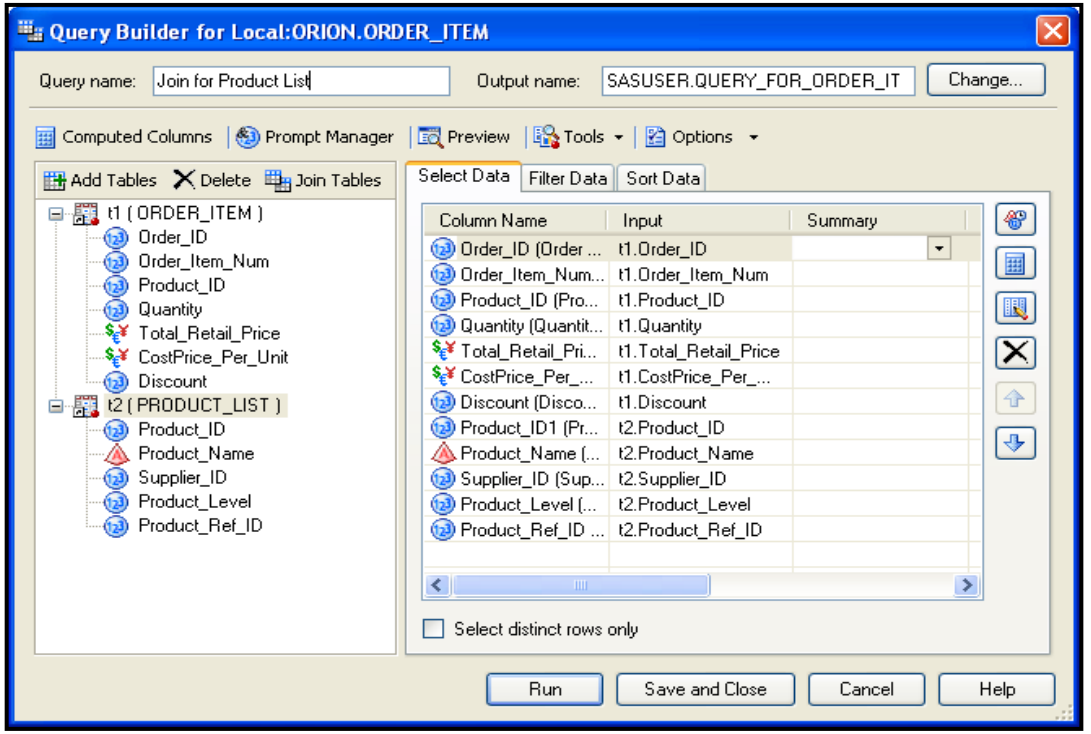




Query Builder Join

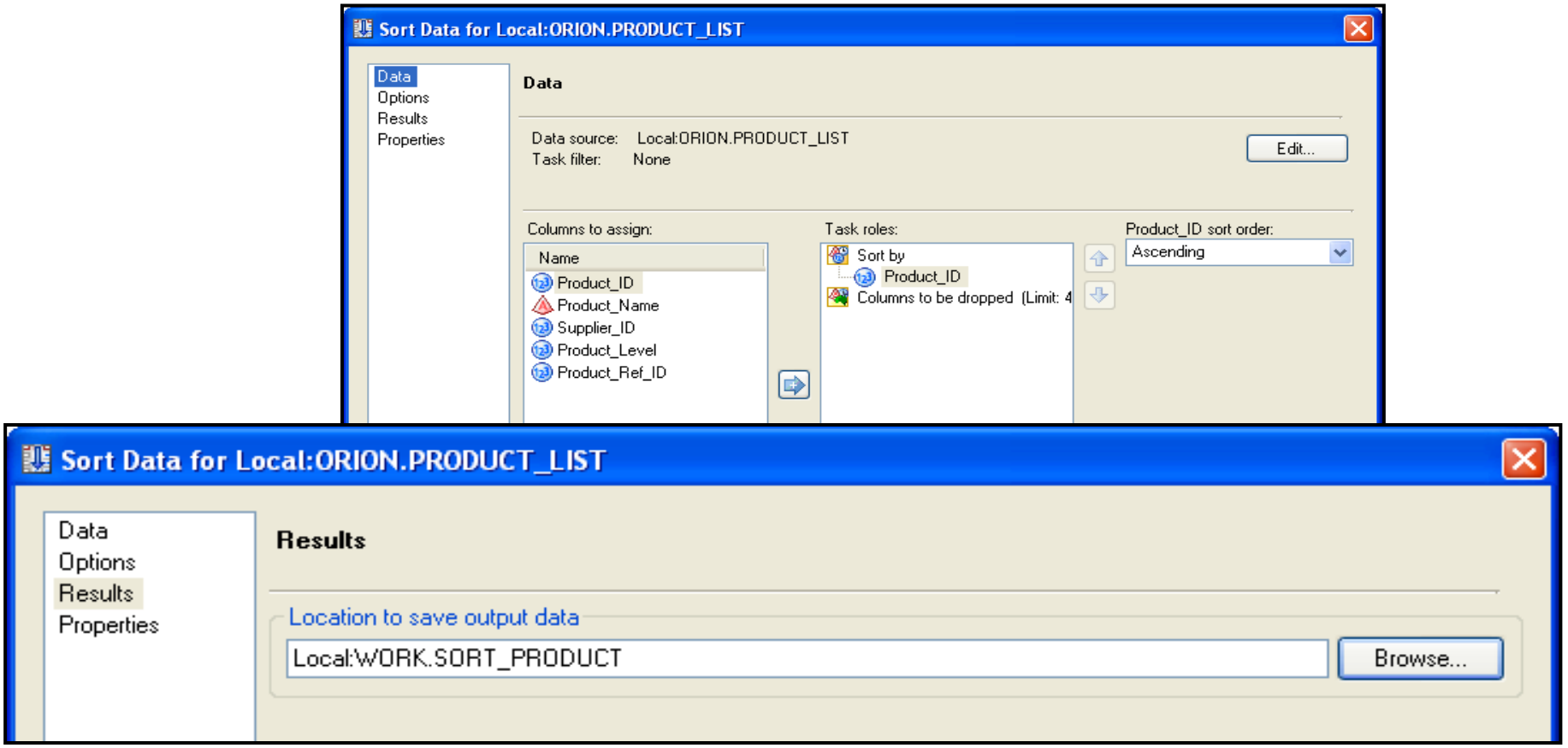
- When you use the Query Builder to join tables in SAS Enterprise Guide, SQL code is generated.

- SQL does not require sorted data.
- SQL can easily join multiple tables on different key variables.
- SQL provides straightforward code
  to join tables
  based on a non-equal comparison of common columns (greater than, less than, between).



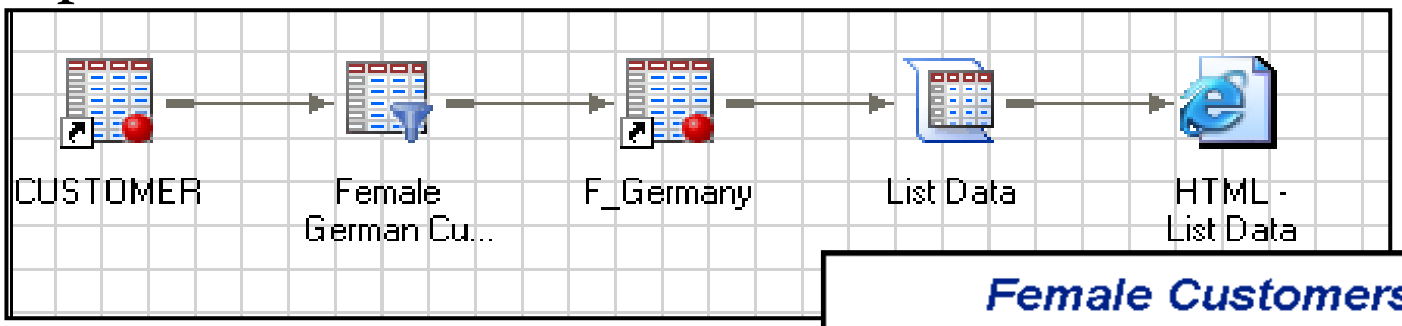
Sort Data Task

- The Sort Data task enables you to create a new data set sorted by one or more variables from the original data.



Business Scenario

- Orion Star wants to send information about a specific promotion to female customers in Germany. The report can be created by querying the **`orion.customer`** data set to include only the desired customers, and then by producing a report with the List Data task.

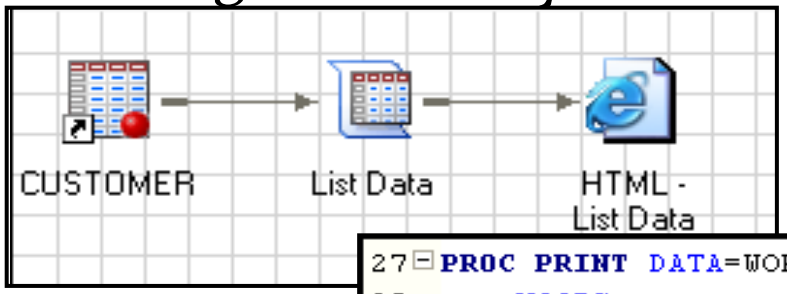


Female Customers in Germany

Customer Country	Customer First Name	Customer Last Name	Customer Birth Date
DE	Cornelia	Krahl	27FEB1974
DE	Elke	Wallstab	16AUG1974
DE	Ines	Deisser	20JUL1969

Business Scenario

- The same report can be generated more efficiently by subsetting the data directly within the List Data task. This requires modification of the code generated by SAS Enterprise Guide.



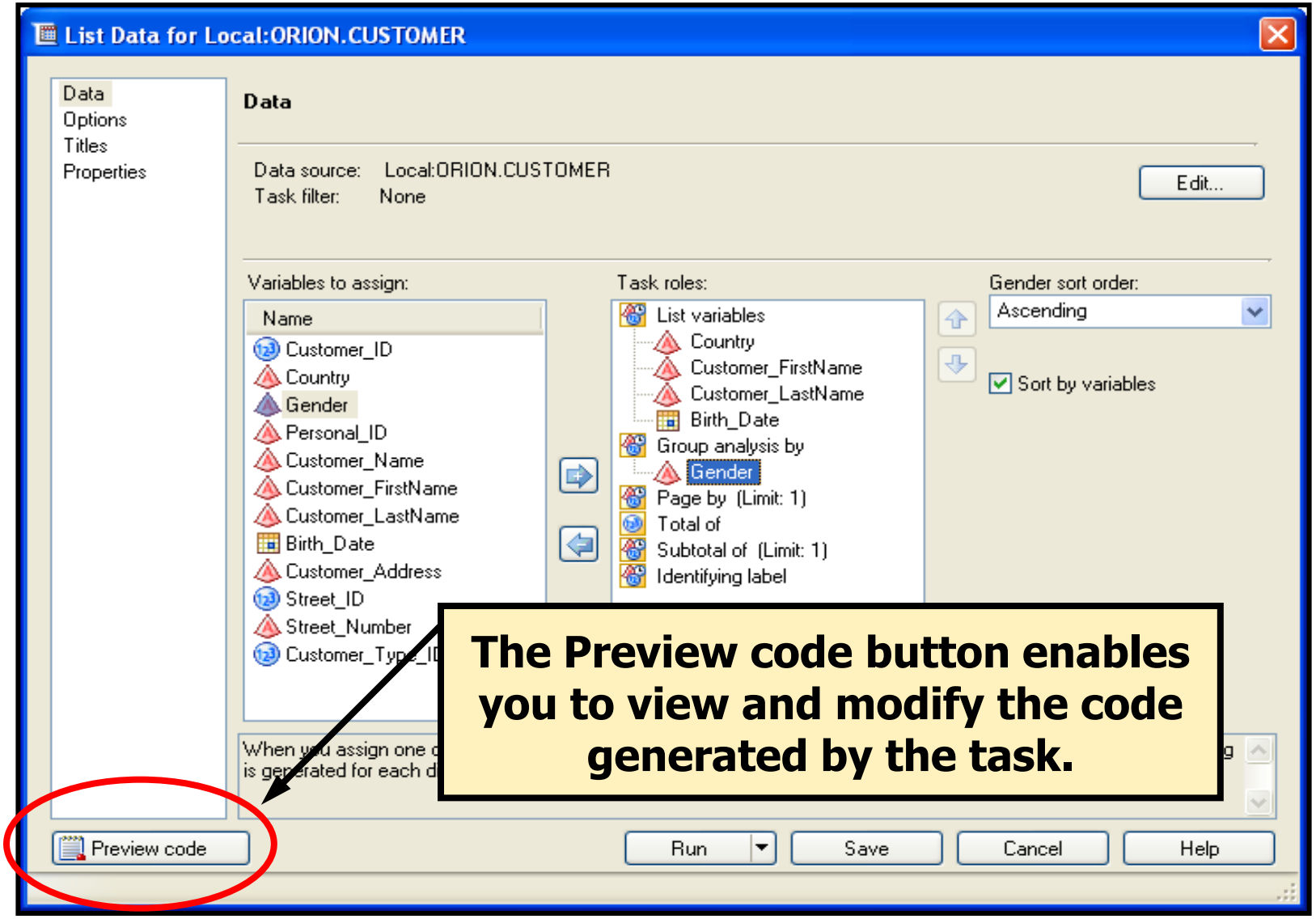

```
27 PROC PRINT DATA=WORK.SORTTempTableSorted
28     NOOBS
29     LABEL
30     ;
31
32 /* Start of custom user code. */
33 where Country = 'DE' and Gender = 'F';
34
35 /* End of custom user code. */
36     VAR Country Customer_FirstName Customer_LastName Birth_Date;
37 RUN;
38 /* ---------------------------------------------------------------
39    End of task code.
40    --------------------------------------------------------------- */
41 RUN; QUIT;
```

Understanding Generated Task Code

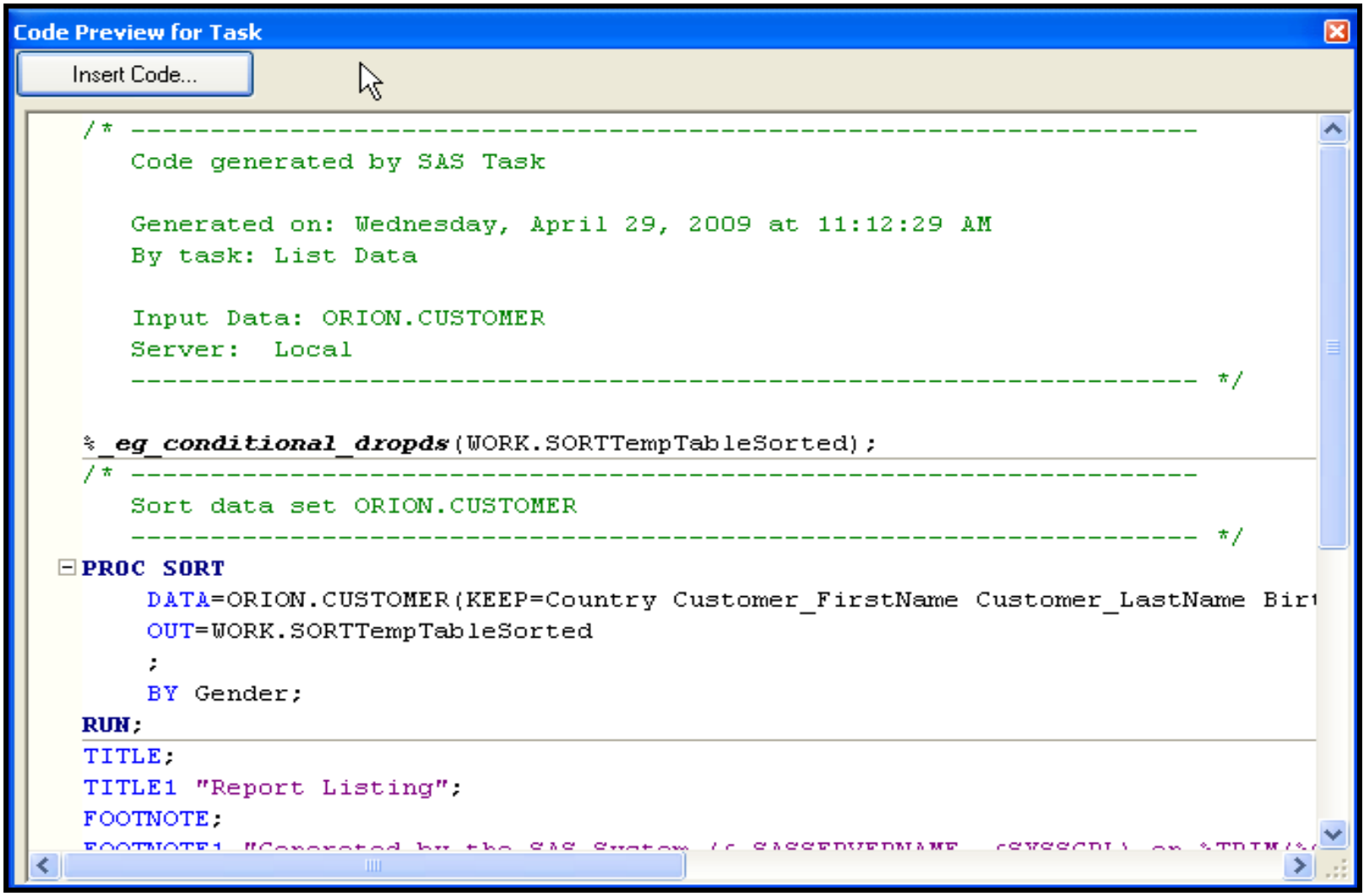
- There are many situations where task results created by SAS Enterprise Guide can be further enhanced or customized by modifying the code.

- However, before you can effectively modify the code, you must first understand the code that SAS Enterprise Guide generates.

List Data Task



List Data Task – Code Preview



Using the List Data Task to Generate Code

- This demonstration illustrates building a List Data task and examining the code generated by SAS Enterprise Guide.

Customer Listing

Customer Gender=F

Customer Country	Customer First Name	Customer Last Name	Customer Birth Date
US	Sandrina	Stephano	09JUL1979
DE	Cornelia	Krahl	27FEB1974
US	Karen	Ballinger	18OCT1984
DE	Elke	Wallstab	16AUG1974

Customer Gender=M

Customer Country	Customer First Name	Customer Last Name	Customer Birth Date
US	James	Kvarniq	27JUN1974
US	David	Black	12APR1969
DE	Markus	Sepke	21JUL1988
DE	Ulrich	Heyde	16JAN1939

List Data Task – Generated Code

- The initial comment block shows information about the task.

```
/* ---------------------------------------------------------------
   Code generated by SAS Task

   Generated on: Wednesday, April 29, 2009 at 1:13:33 PM
   By task: List Data

   Input Data: ORION.CUSTOMER
   Server:  Local
   --------------------------------------------------------------- */
```

List Data Task – Generated Code

- The first line uses a macro to delete temporary tables or views if they already exist. If the Group by role is used in the task, the data must be ordered by the grouping variable. PROC SORT is used by default. Only variables assigned to roles are kept in the new data set.

```
%_eg_conditional_dropds(WORK.SORTTempTableSorted);
/* ---------------------------------------------------------------
   Sort data set ORION.CUSTOMER
   --------------------------------------------------------------- */
PROC SORT
     DATA=ORION.CUSTOMER(KEEP=Country Customer_FirstName
                              Customer_LastName Birth_Date
                              Gender)
     OUT=WORK.SORTTempTableSorted
     ;
     BY Gender;
RUN;
```

List Data Task – Generated Code

- If the Group by role is **not** used, SQL creates a temporary view of the required data. Again, only variables assigned to roles in the task are included in the view. This comment i**ncorrectly** states that sorting occurs.

```
%_eg_conditional_dropds(WORK.SORTTempTableSorted);
/* ----------------------------------------------------------
   Sort data set ORION.CUSTOMER
   ---------------------------------------------------------- */

PROC SQL;
     CREATE VIEW WORK.SORTTempTableSorted AS
          SELECT T.Country, T.Customer_FirstName,
                 T.Customer_LastName, T.Birth_Date
     FROM ORION.CUSTOMER as T
;
QUIT;
```

List Data Task – Generated Code

- The main part of the code includes the titles, footnotes, and procedure code to generate the report. PROC PRINT is the procedure used with the List Data task.

```
TITLE;
TITLE1 "Customer Listing";
FOOTNOTE;
FOOTNOTE1 "Generated by the SAS System (&_SASSERVERNAME, &SYSSCPL)
           on %TRIM(%QSYSFUNC(DATE(), NLDATE20.))
           at %TRIM(%SYSFUNC(TIME(), NLTIMAP20.))";

PROC PRINT DATA=WORK.SORTTempTableSorted
    NOOBS="Row number"
    LABEL
    ;
    VAR Country Customer_FirstName Customer_LastName Birth_Date;
    BY Gender;
RUN;
```

TITLE and FOOTNOTE are examples of *global* statements and can be included anywhere in a SAS program.

List Data Task – Generated Code

- At the end, the final lines of code delete any temporary tables created to build the task, and delete any assigned titles and footnotes.

```
/* ---------------------------------------------------------
   End of task code.
   --------------------------------------------------------- */
RUN; QUIT;
%_eg_conditional_dropds(WORK.SORTTempTableSorted);
TITLE; FOOTNOTE;
```

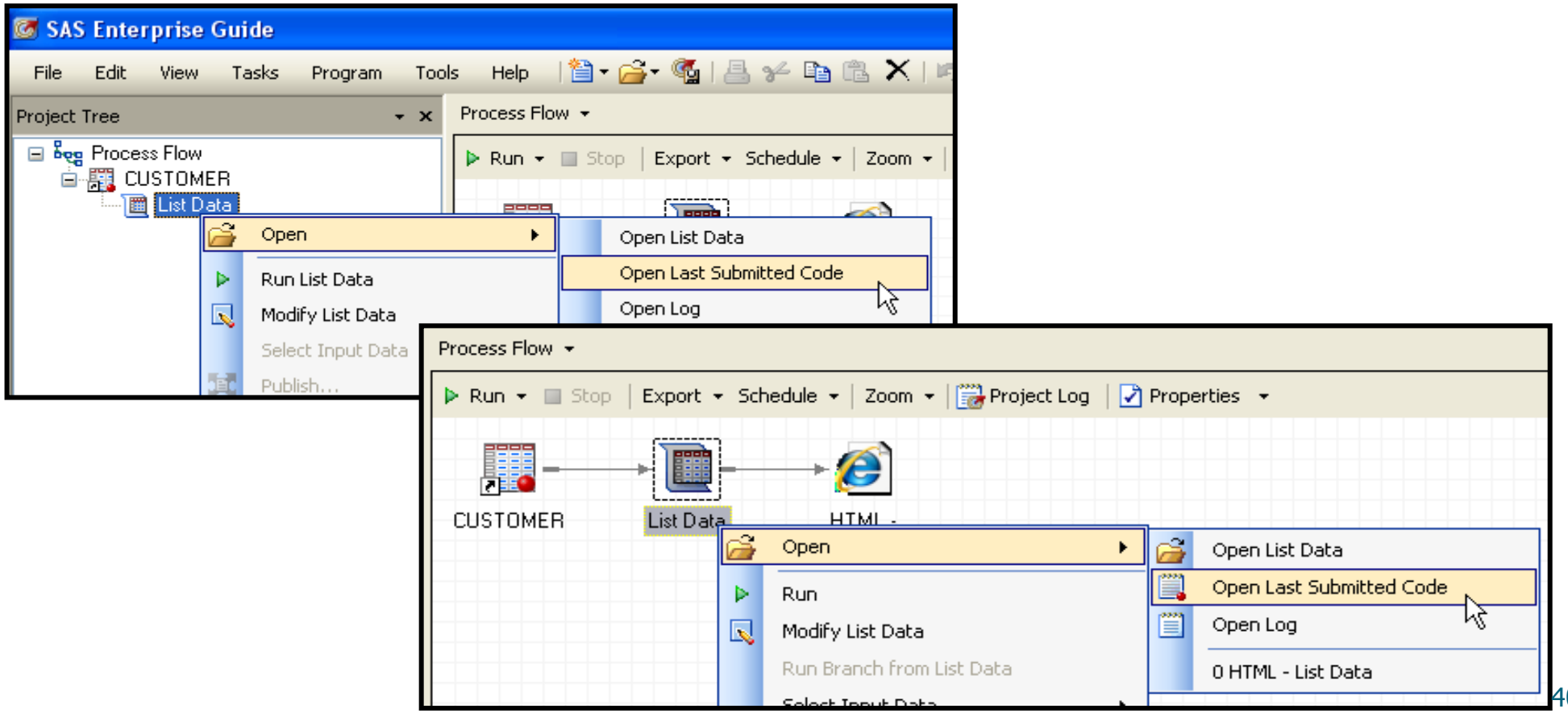
Techniques to Modify Code

- Three methods can be used to modify code generated by SAS Enterprise Guide:

1. Edit the last submitted task code in a separate Code window.
2. Automatically submit custom code before or after every task and query.
3. Insert custom code in a task.

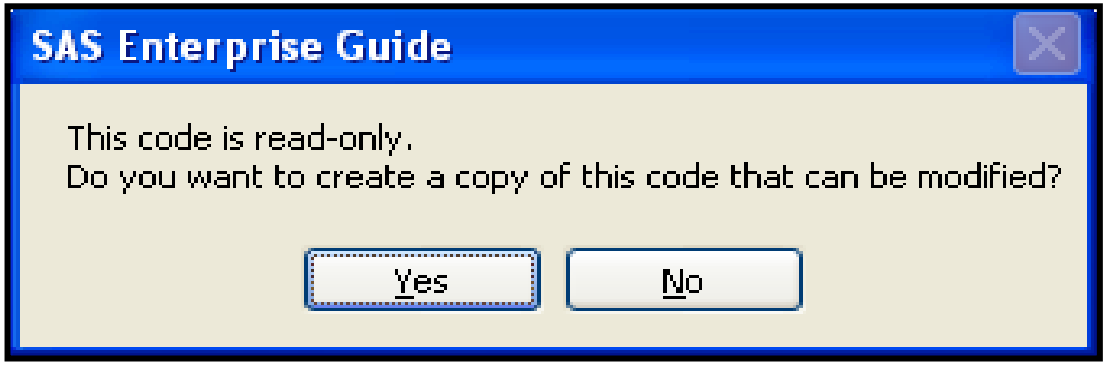
Edit Last Submitted Code

- After a task runs, the code can be viewed from either the Project Tree or Process Flow.



Edit Last Submitted Code

The task code is read-only and cannot be edited directly. To create a copy of the code from the Last Submitted Code window, select any key while in the SAS program window. SAS Enterprise Guide offers to make a copy.



After the code is copied, there is no link between the task and the new code. Any changes in the task are not reflected in the copied code, and modifications to the code do not affect the task.

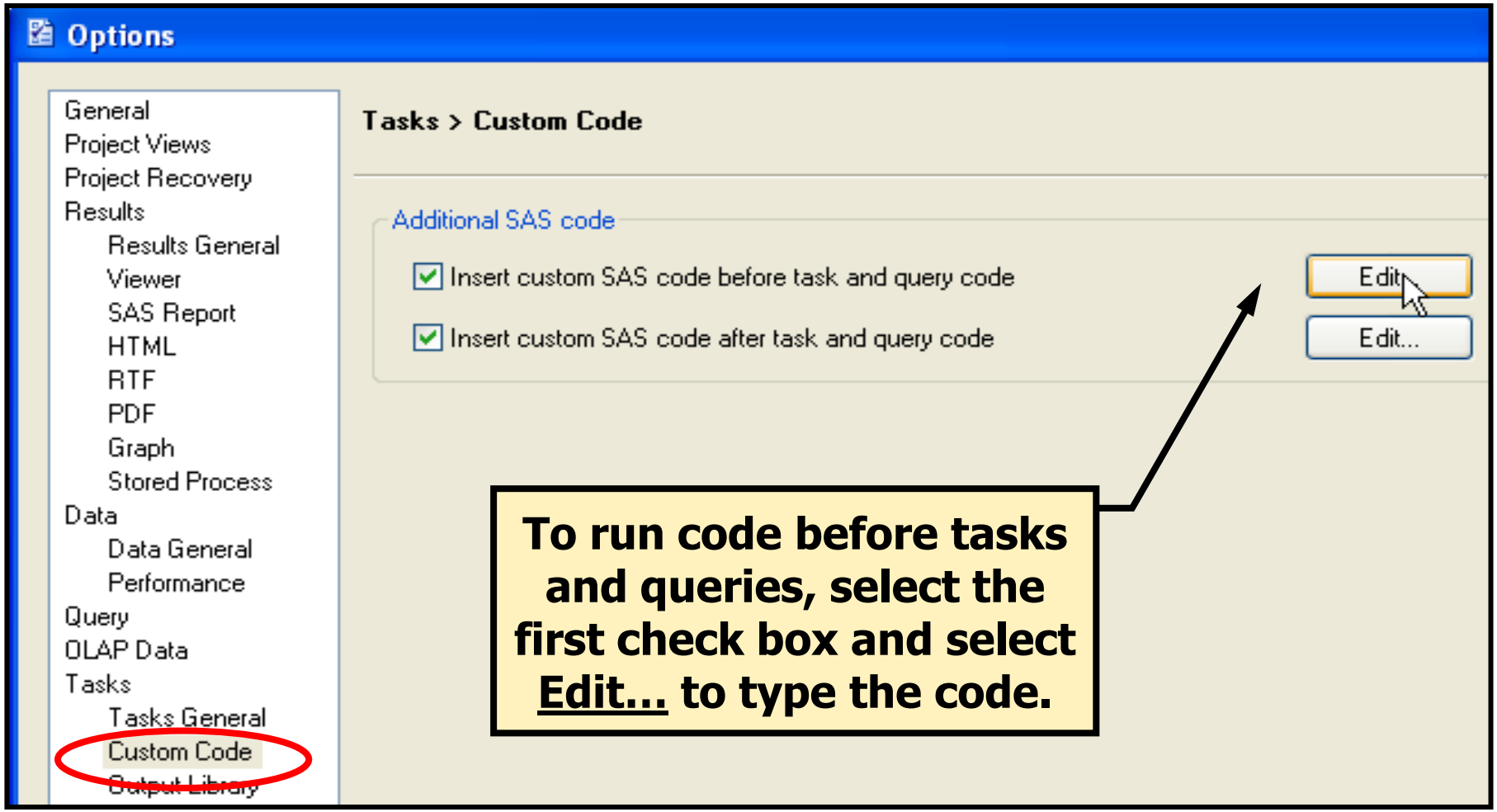
Summary of Editing Last Submitted Code

Custom code linked to task?	No
Can be used to modify query code?	Yes
Extent of modification allowed?	Anything in the program can be changed.
Custom code included when exported?	Yes. You **must** export the edited program and select the option in the Export wizard.

Automatically Submit Custom Code Before or After Every Task and Query

- There are times when you might need to run a SAS statement or program before or after any task or query is executed. The Custom Code option enables you to insert custom code before or after all tasks and queries.

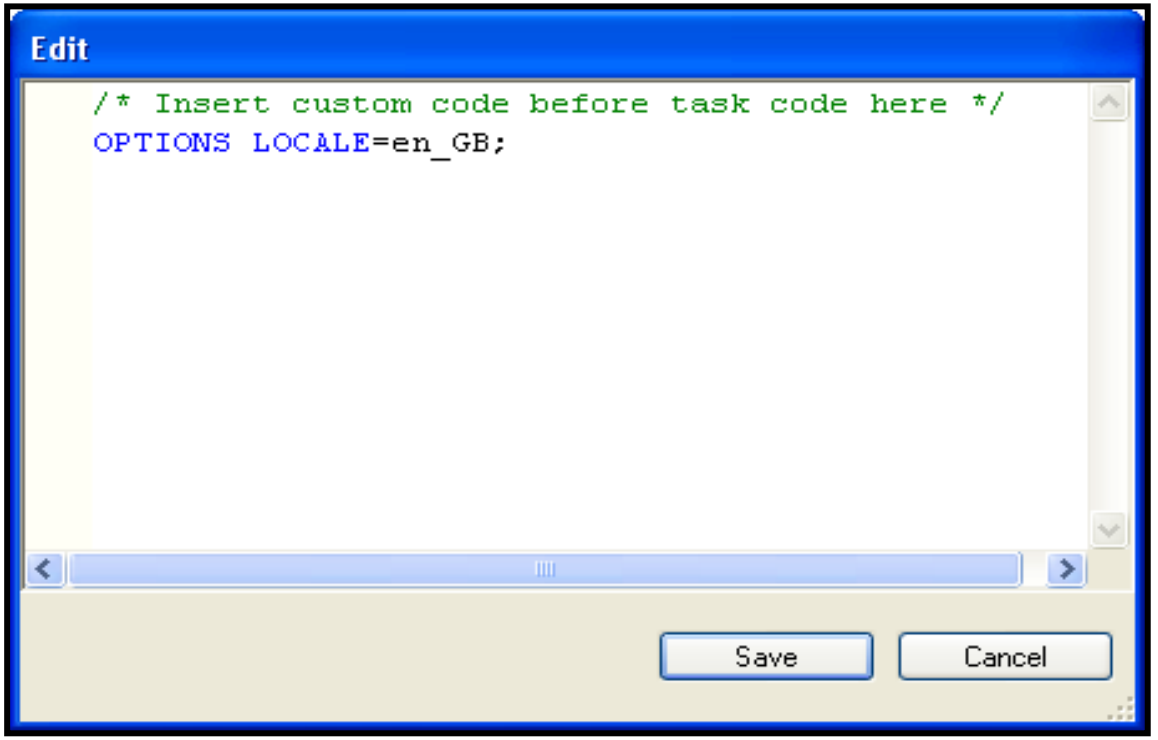
Automatically Submit Custom Code Before or After Every Task and Query



Automatically Submit Custom Code Before or After Every Task and Query

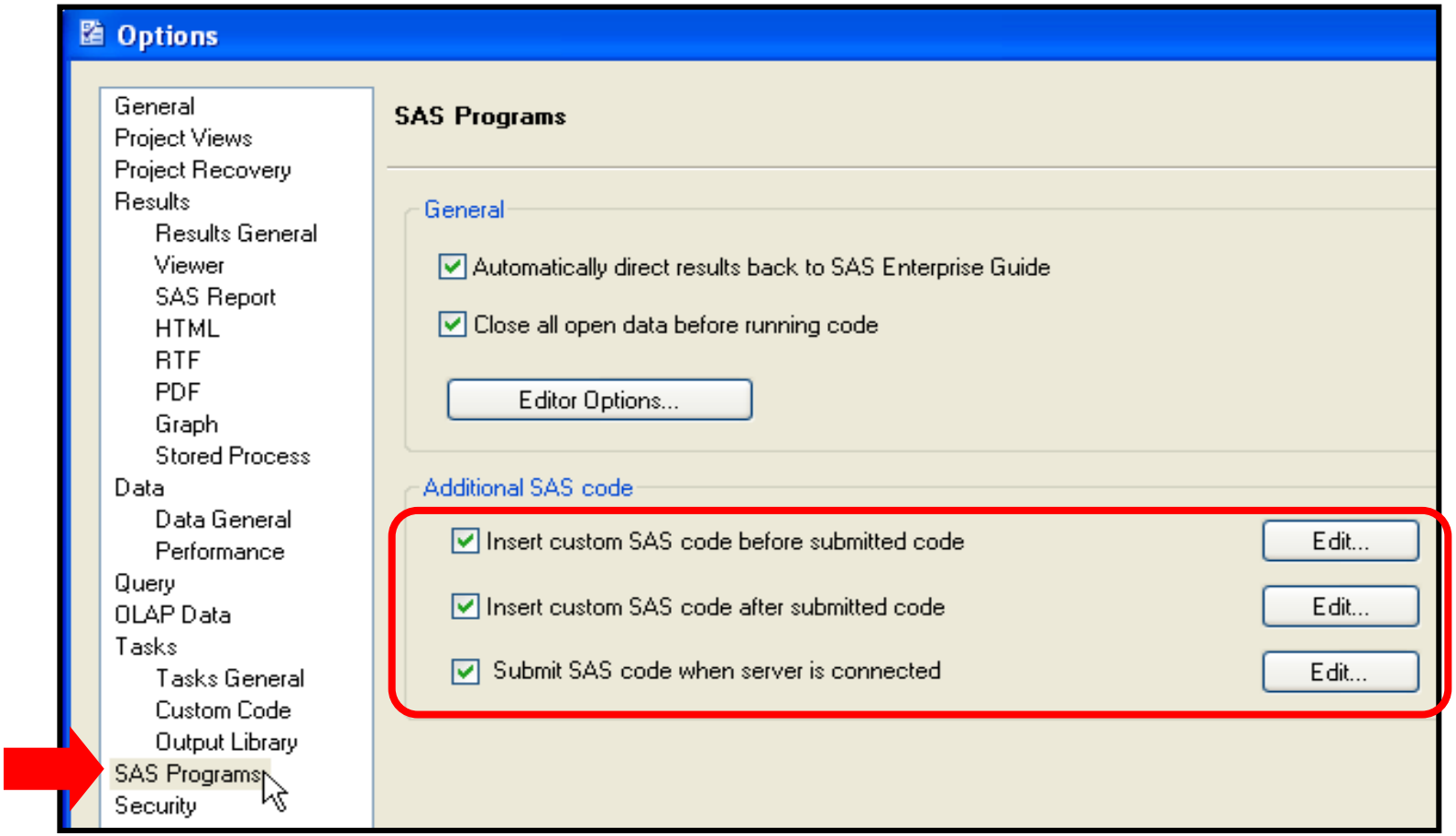
Global statements or complete program steps can be entered.

Example: Set the LOCALE= option to Great Britain.



Insert Code Before or After SAS Programs

- Similar options exist to automatically submit code before or after SAS programs written and submitted in Code windows in SAS Enterprise Guide.

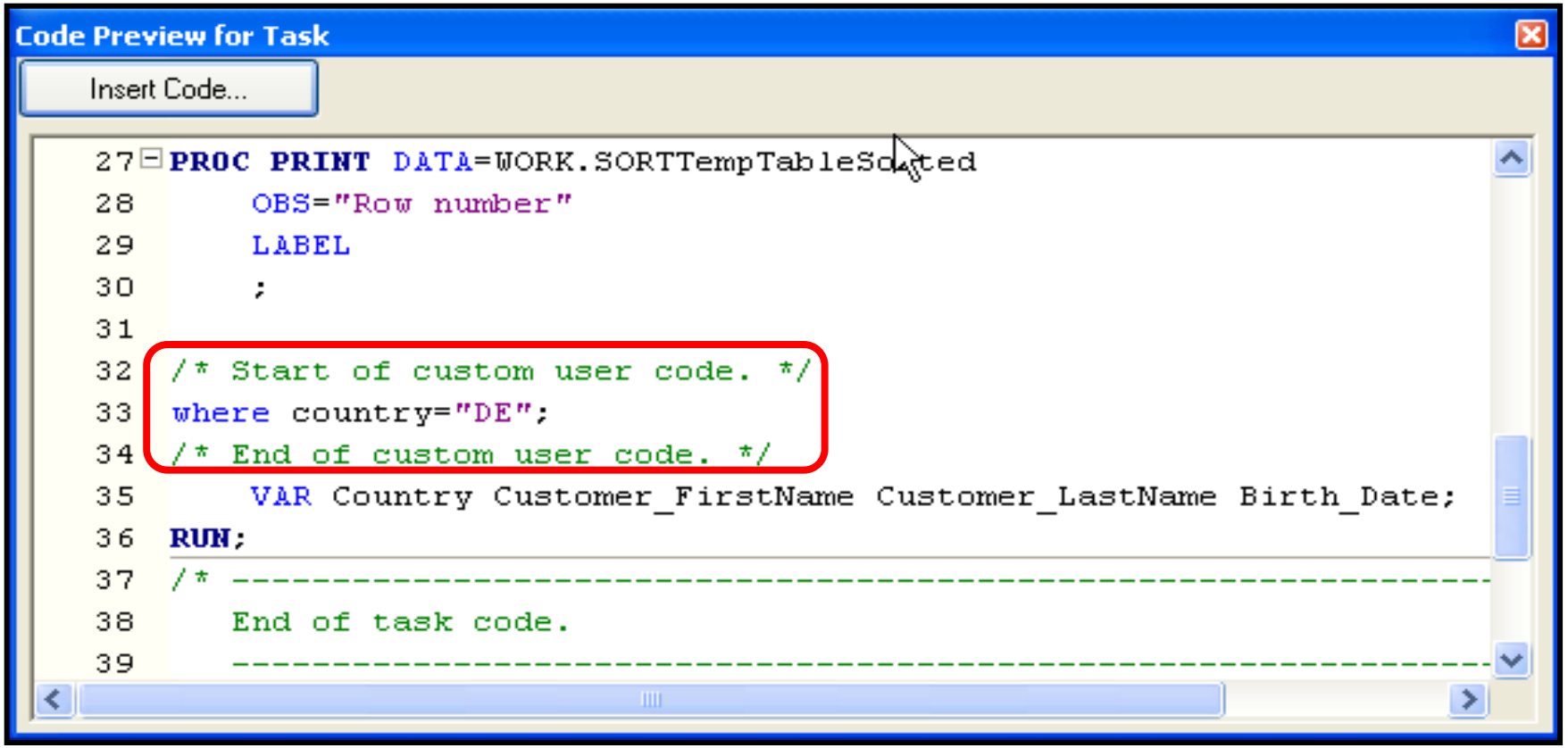


Summary of Submitting Custom Code Before or After Every Task and Query

Custom code linked to task?	Yes
Can be used to modify query code?	Yes
Extent of modification allowed?	Statements can only be submitted before or after the task code.
Custom code included when exported?	Yes, select the option in the Export wizard.

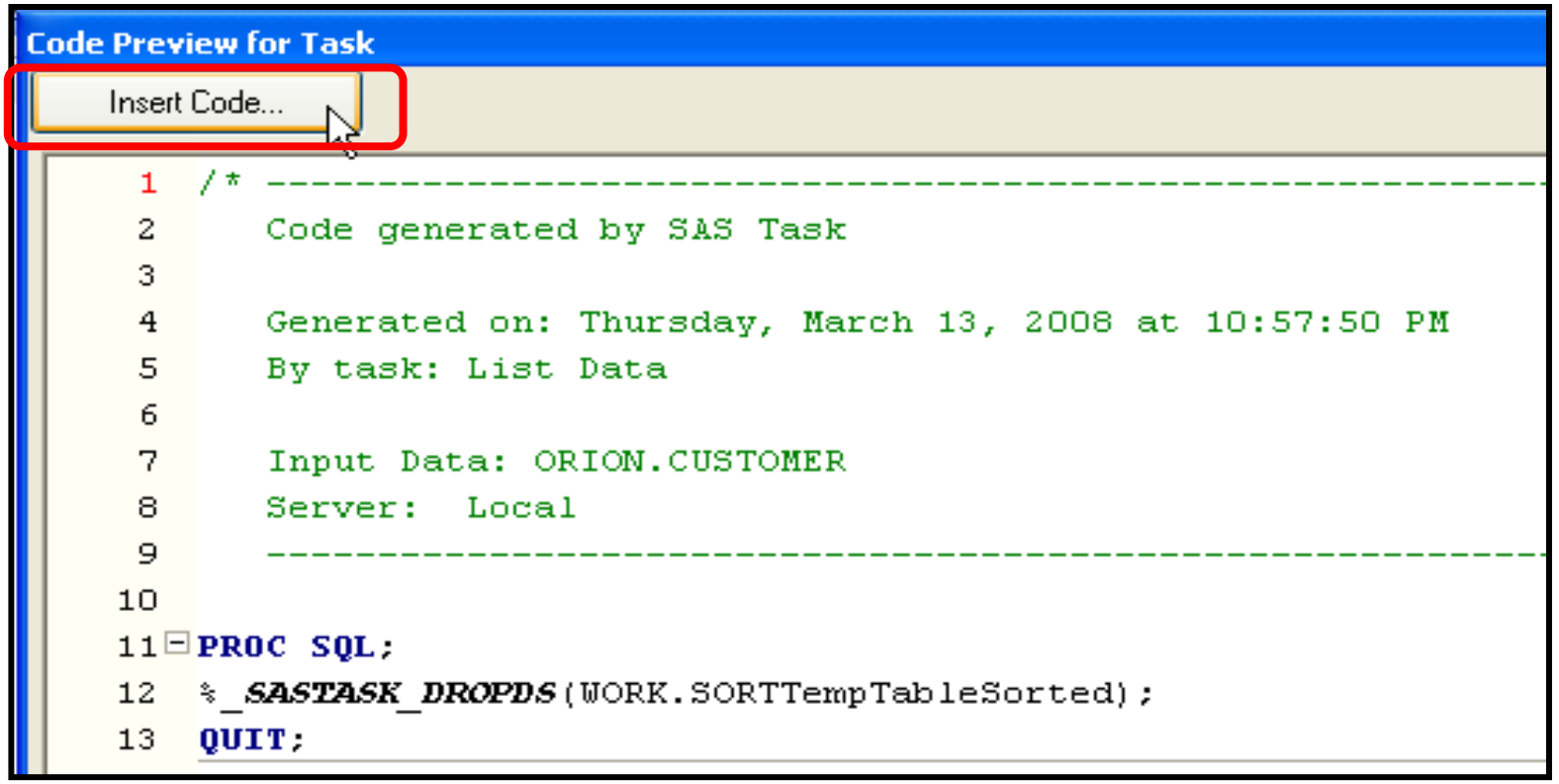
Insert Custom Code in a Task

- In most task dialog boxes, you have the ability to insert custom code within the generated SAS program. This technique has the significant benefit that the task interface can still be used to modify the report.

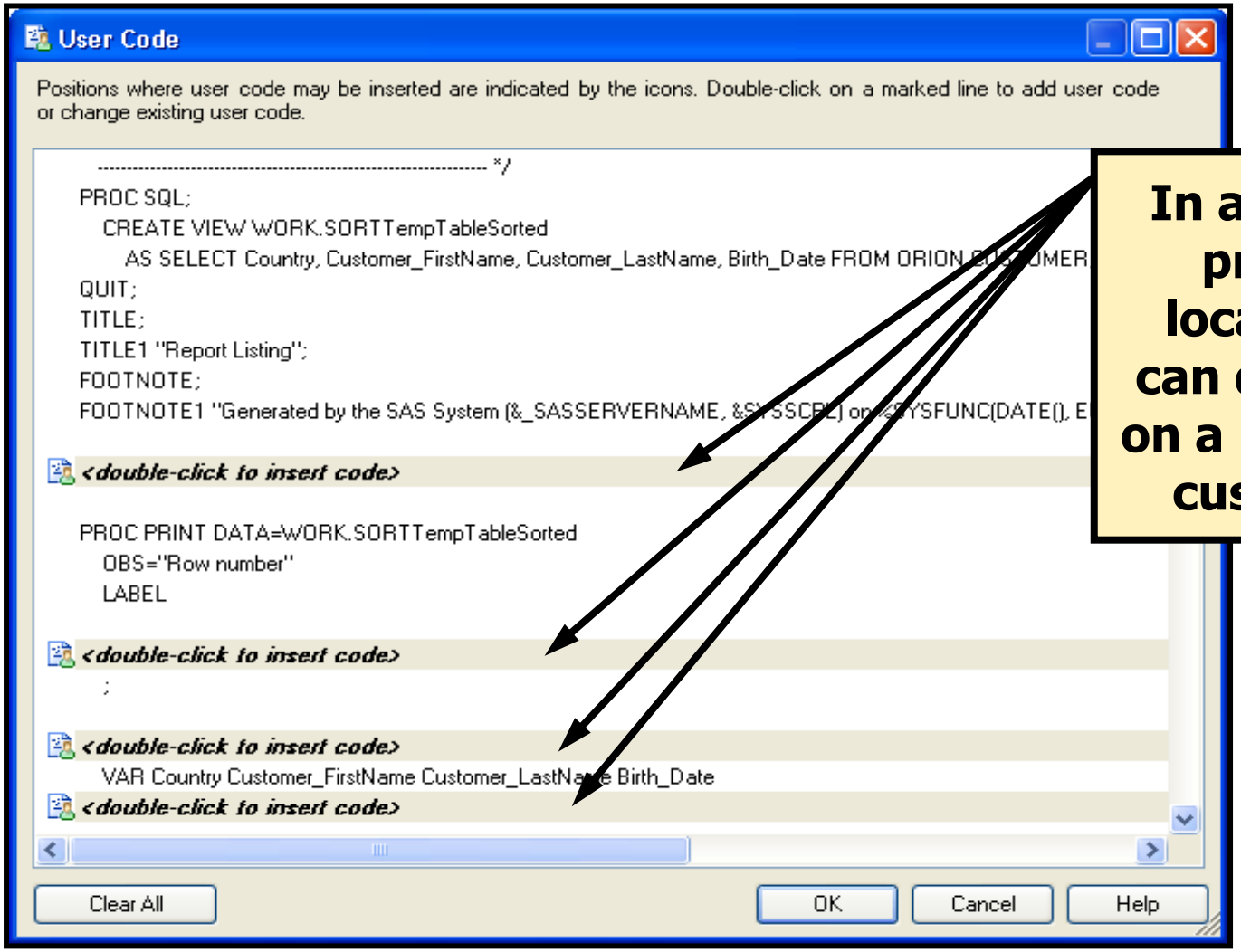


Insert Custom Code in a Task

- In the Code Preview window, select **Insert Code...** to add custom code in predefined locations in the SAS program.



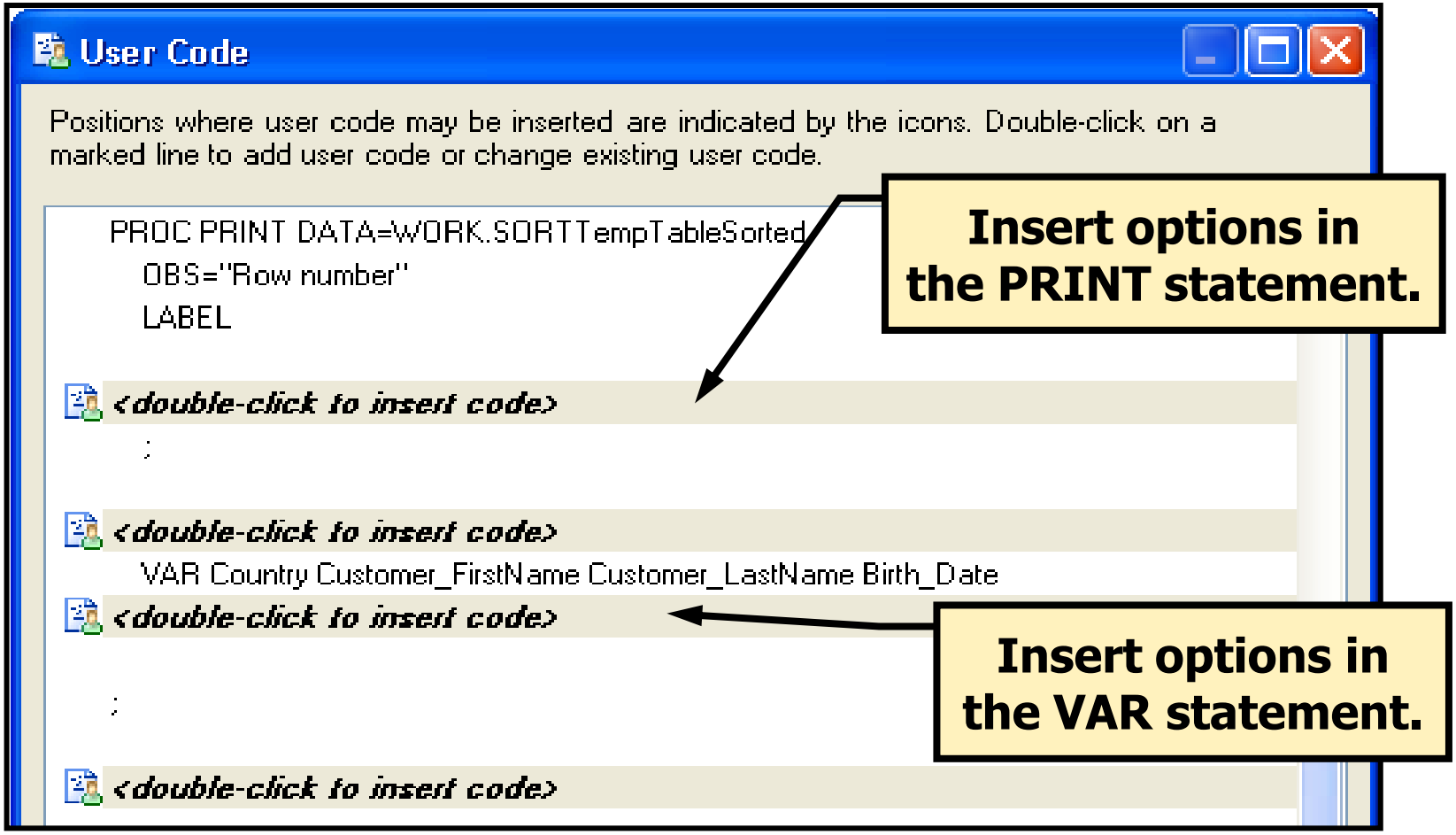
Insert Custom Code in a Task



In any of these predefined locations, you can double-click on a line to insert custom code.

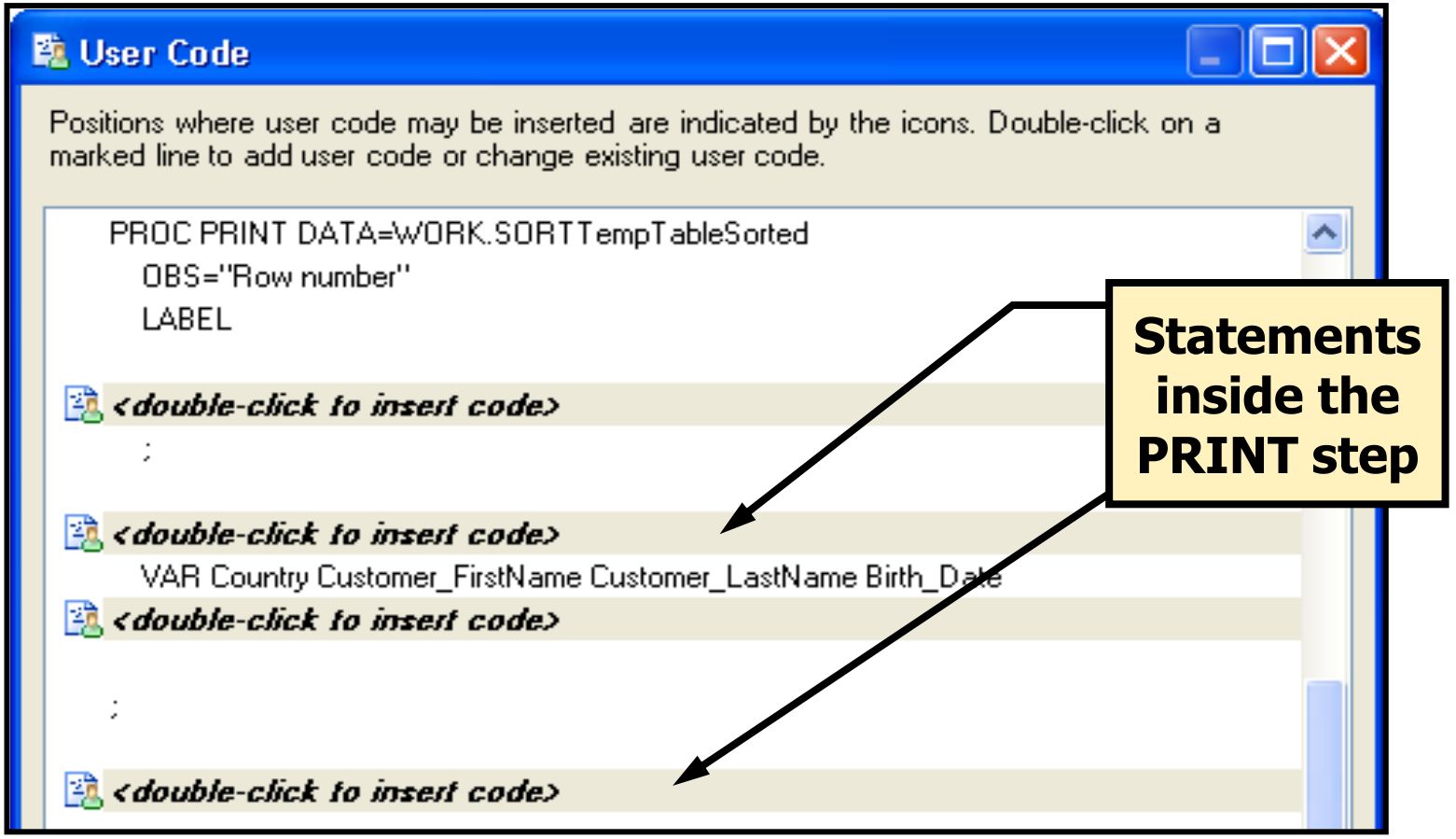
Insert Custom Code in a Task

- Some insert points enable custom options to be added to existing statements.



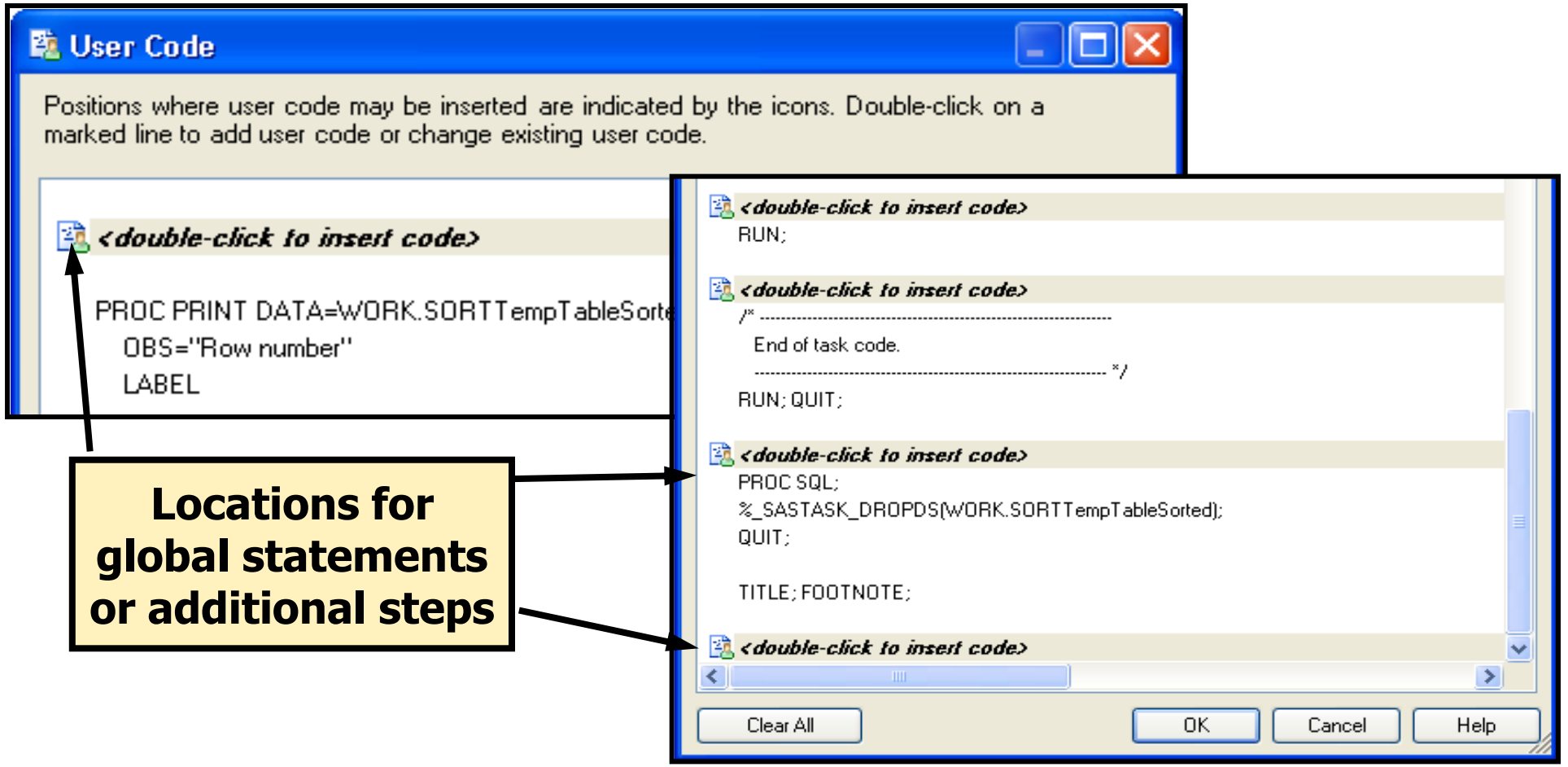
Insert Custom Code in a Task

- Other insert points enable entire statements to be added inside a step in the program.

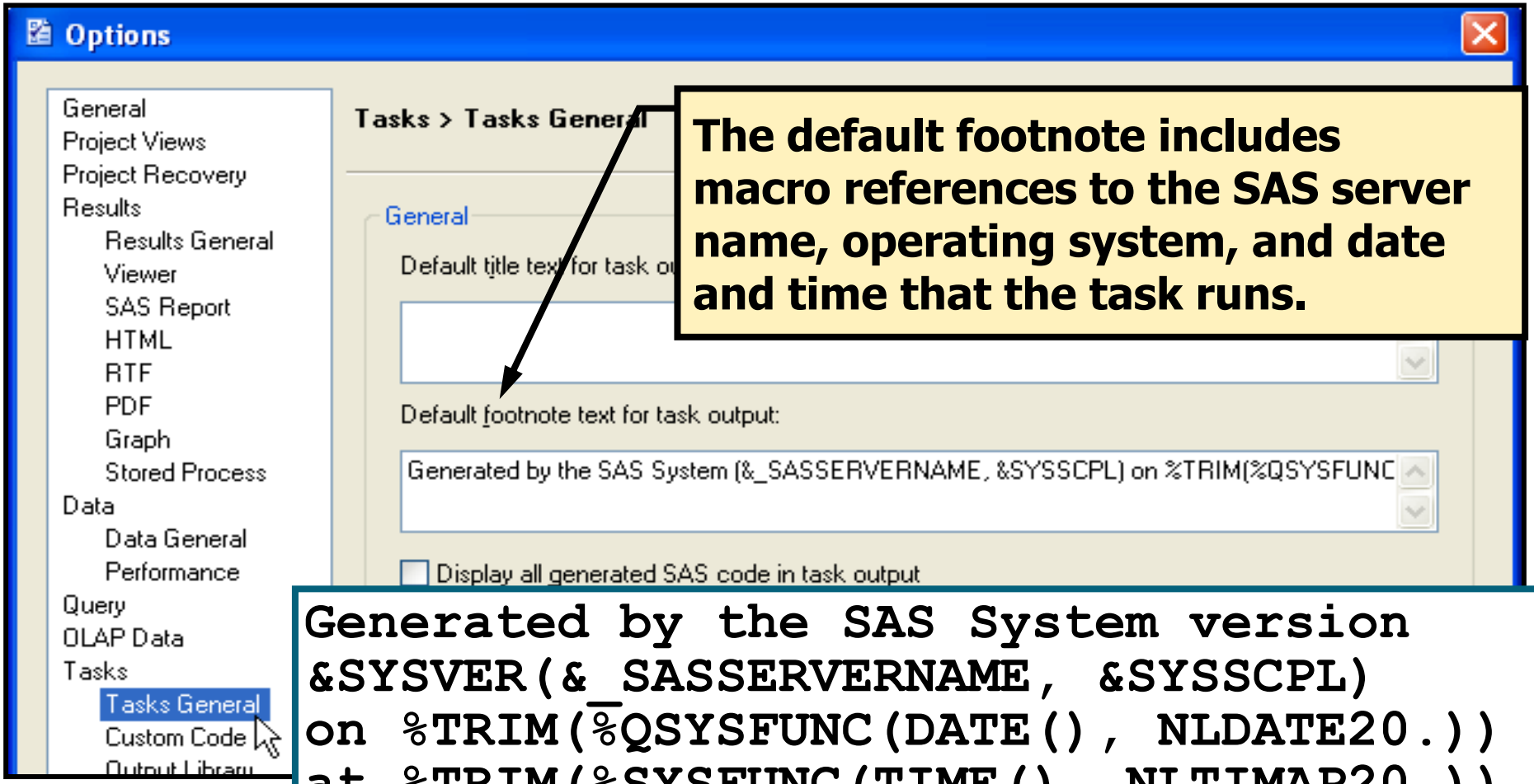


Insert Custom Code in a Task

- Additional locations enable global statements or additional steps to be inserted before or after the main code.

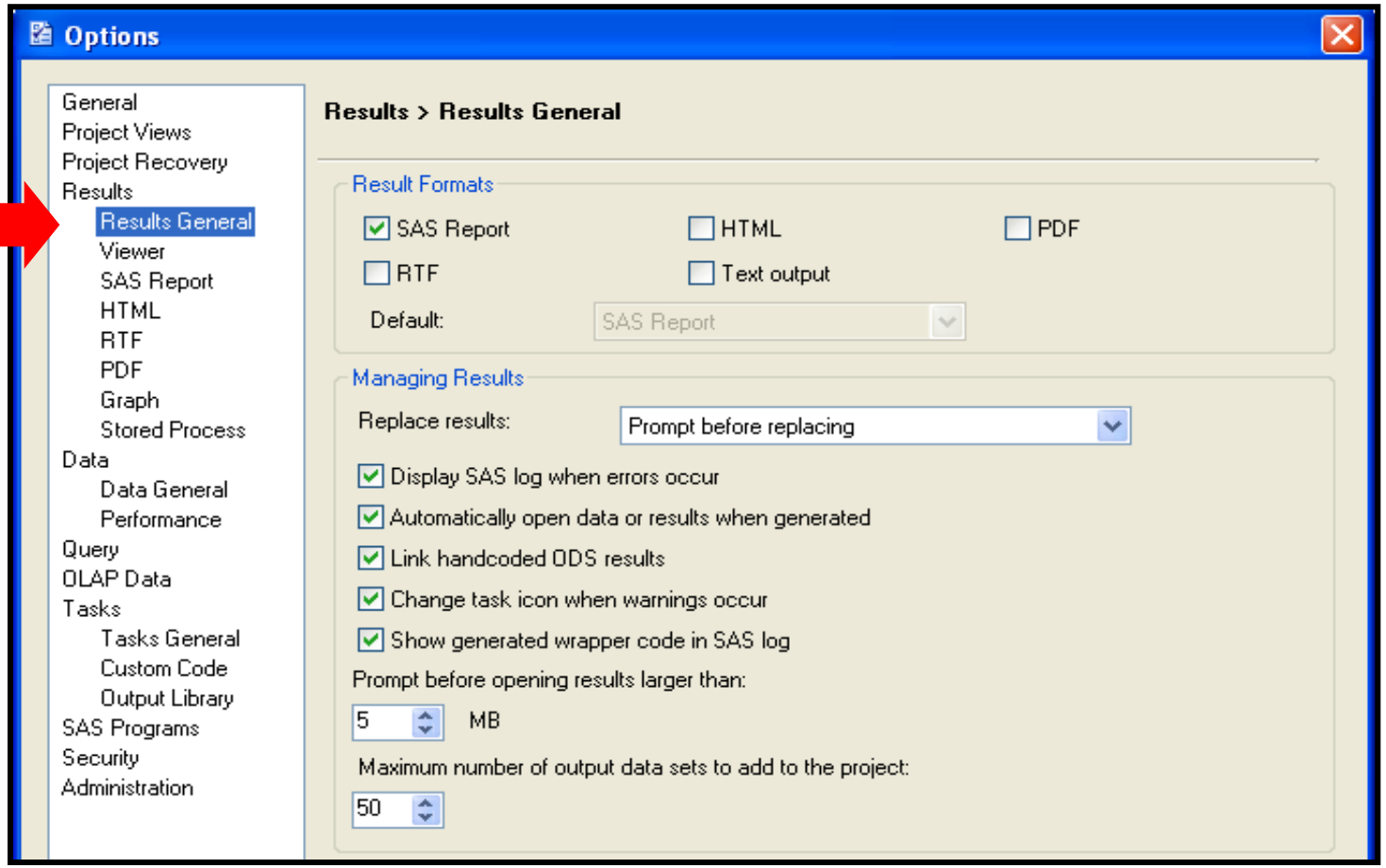


Default SAS Enterprise Guide Footnote



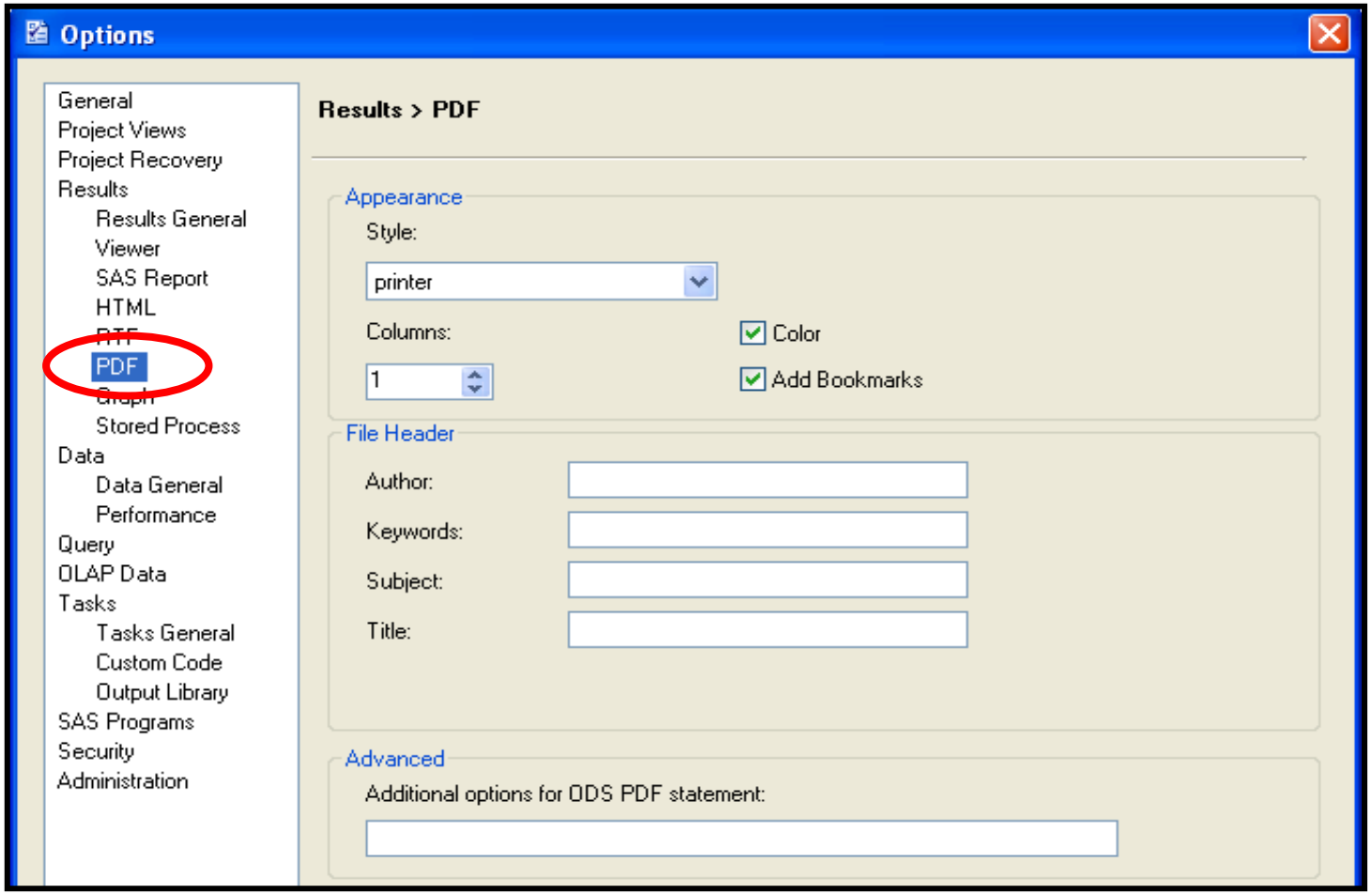
ODS and SAS Enterprise Guide

- Default result formats can be set under **Tools** ⇨ **Options**.



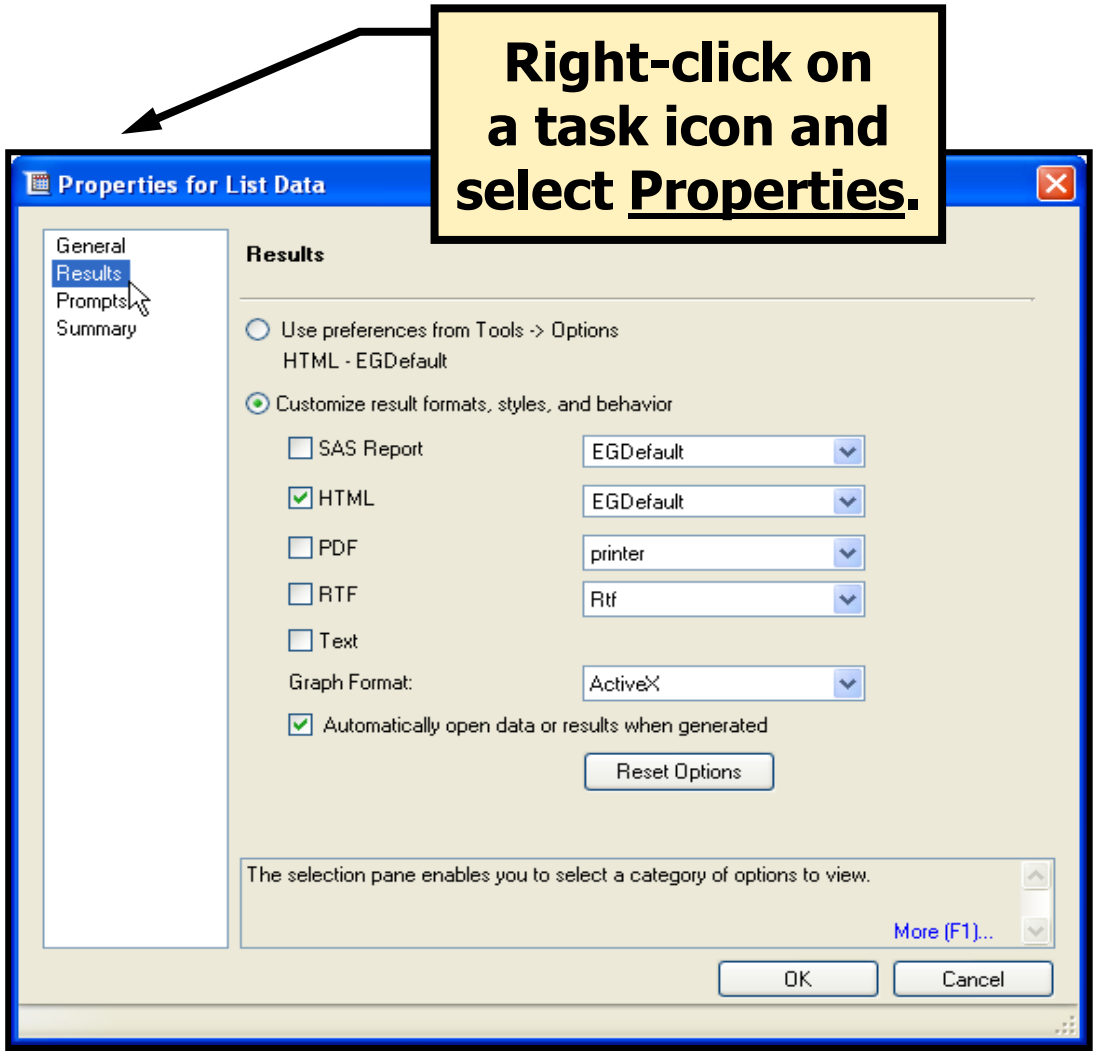
ODS and SAS Enterprise Guide

- Additional settings can be made for each result format.



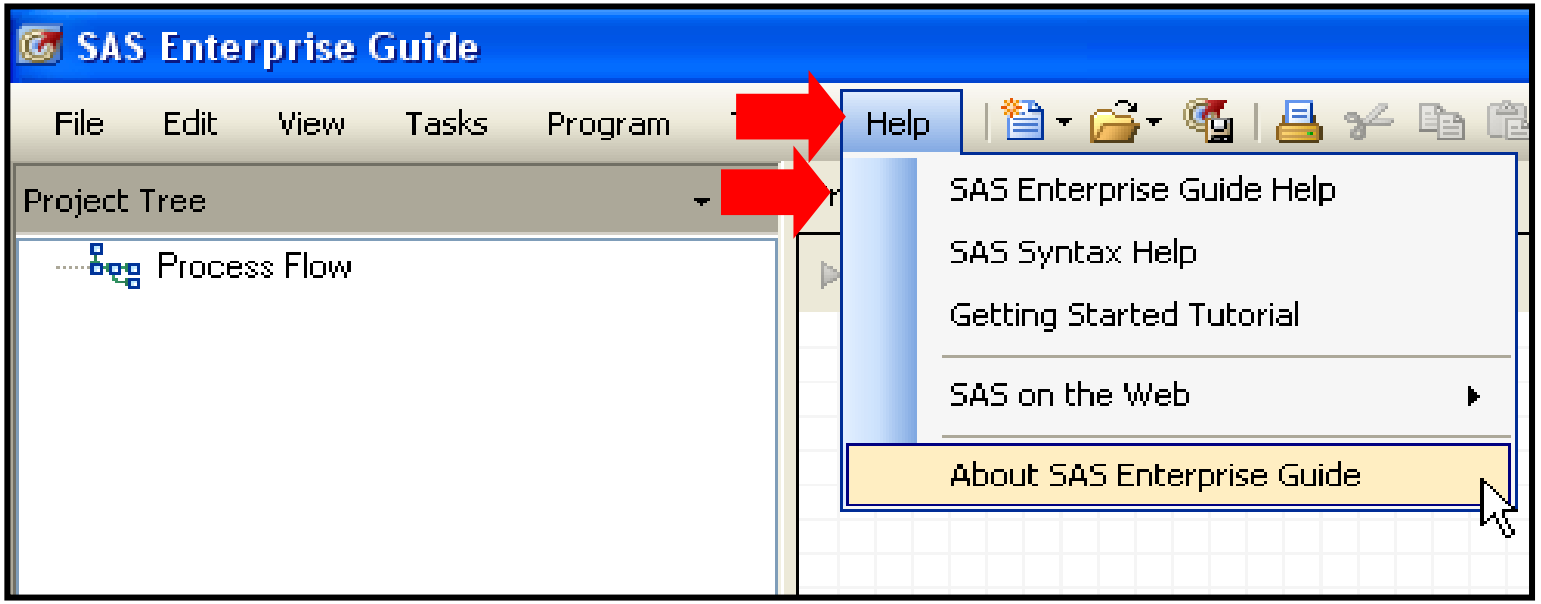
ODS and SAS Enterprise Guide

- Task properties can be used to override the default for an individual task.
- Generated output can be switched off completely and handled by inserting code.

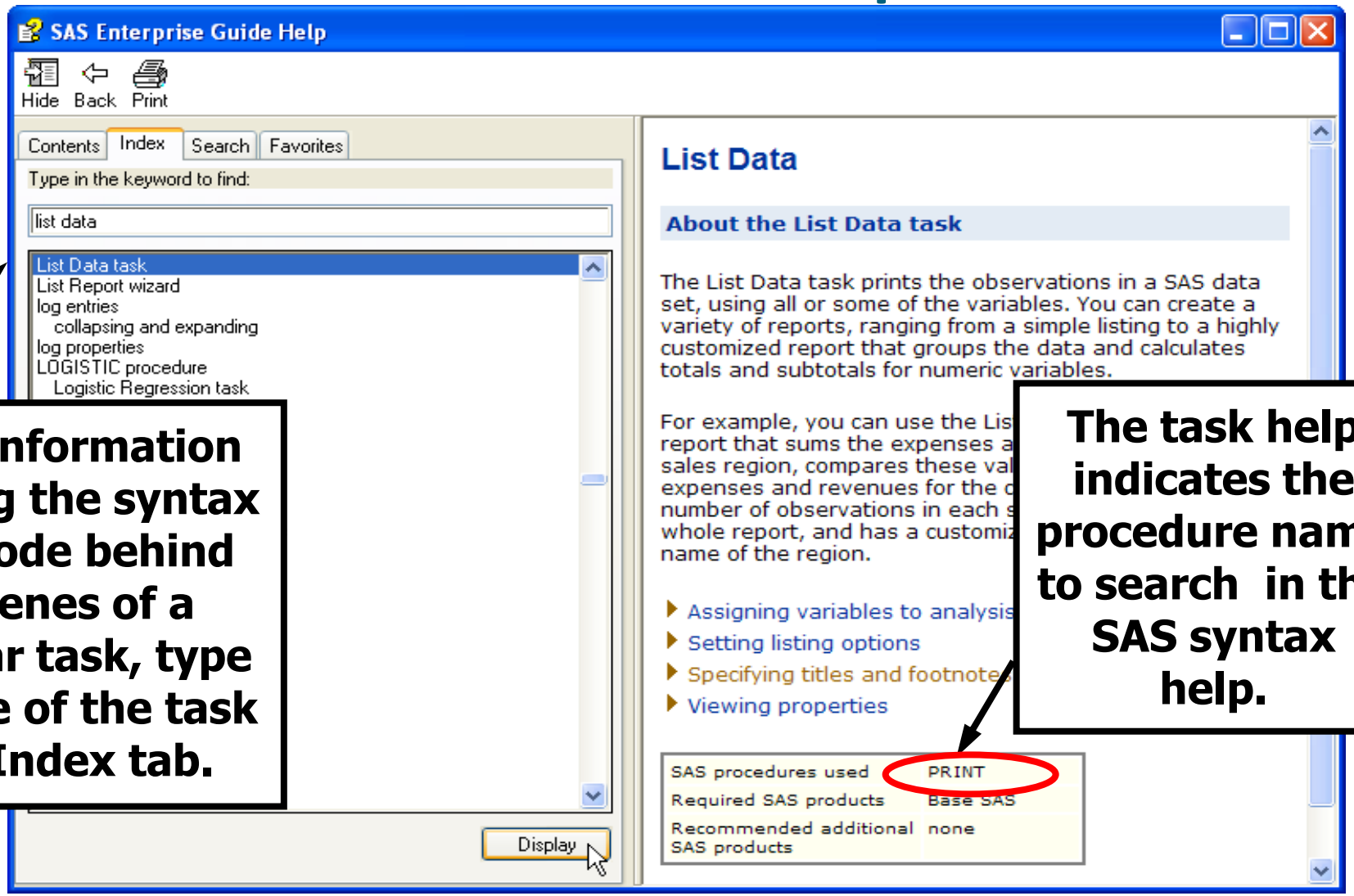


SAS Enterprise Guide Help (Review)

- If Help files were installed along with SAS Enterprise Guide, you can select **Help** to access the Help facility regarding both the point-and-click functionality of SAS Enterprise Guide as well as SAS syntax.



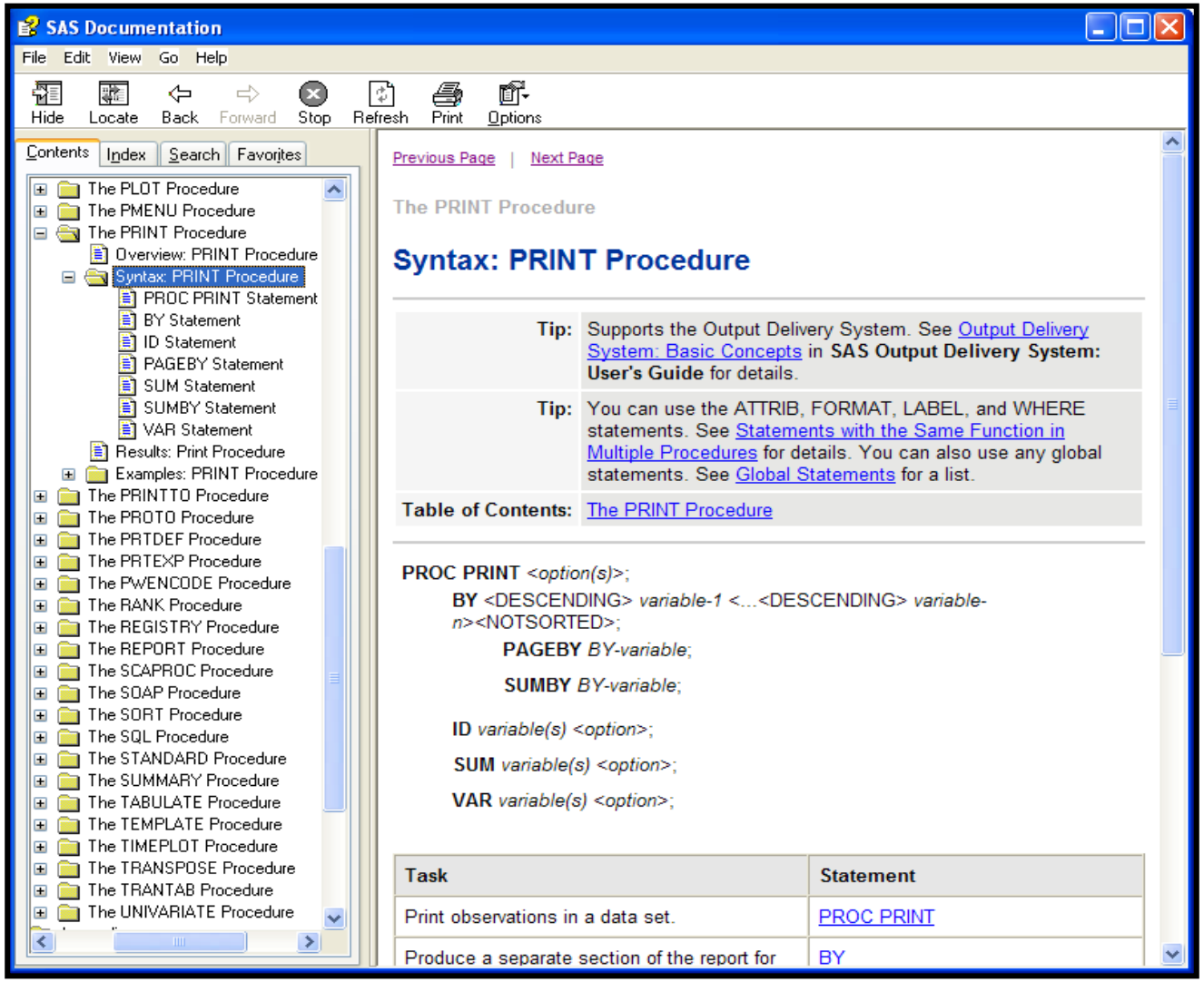
Task and Procedure Help



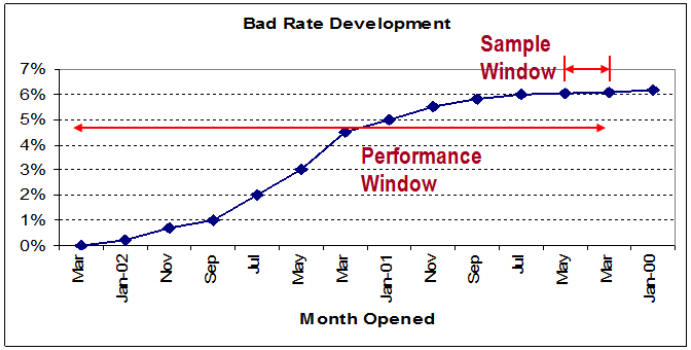
To find information regarding the syntax of the code behind the scenes of a particular task, type the name of the task in the Index tab.

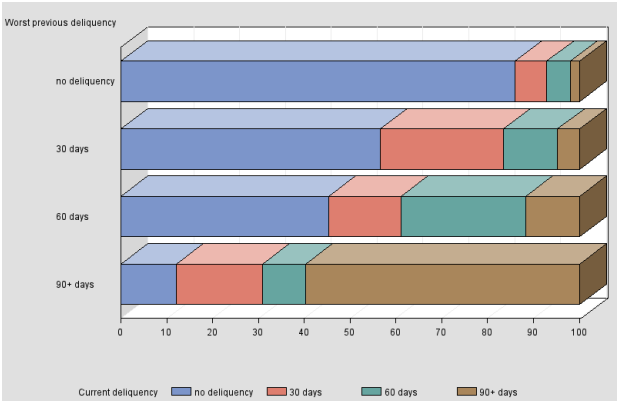
The task help indicates the procedure name to search in the SAS syntax help.

Procedure Syntax Help



7. Metodologie vývoje scoringových funkcí





Objectives

- Understand how scorecards to predict credit risk are developed.
- Understand the analyses and issues for implementation of scorecards.

Main Stages – Development

- Stage 1: Preliminaries and Planning
  - Create Business Plan
    - Identify organizational objectives
    - Internal versus External development, and scorecard type
  - Create Project Plan
    - Identify project risks
    - Identify project team.

Main Stages – Development

- Stage 2: Data Review and Project Parameters
  - Data availability and quality
  - Data gathering for definition of project parameters
  - Definition of project parameters
    - Performance window and sample window
    - Performance categories definition (target)
    - Exclusions
  - Segmentation
  - Methodology
  - Review of implementation plan.

Main Stages – Development

- Stage 3: Development Database Creation
  - Development sample specification
  - Sampling
  - Development data collection and construction
  - Adjusting for prior probabilities.

Main Stages – Development

- Stage 4: Scorecard Development
  - Missing values and outliers
  - Initial characteristic analysis
  - Preliminary scorecard
  - Reject inference
  - Final scorecard production
    - Scaling
    - Points allocation
    - Misclassification
  - Scorecard strength
  - Validation.

Main Stages – Development

- Stage 5: Scorecard Management Reports
  - Gains tables and charts
  - Characteristic reports.

Main Stages – Implementation

- Stage 1: Pre-Implementation Validation

- Stage 2: Strategy Development
  - Scoring strategy
  - Setting cutoffs
  - Strategy considerations
  - Policy rules
  - Overrides.

Main Stages – Post Implementation

- Post-Implementation
  - Scorecard and Portfolio Monitoring Reports
  - Review.

Development

Stage 1: Preliminaries and Planning

Objectives

- Create a business plan to ensure a viable and smooth project.

- ***"All Models are wrong. Some are useful"***

George Box

Create Business Plan

- Identify organizational objectives.
  - Reasons for model development
    - Profit, revenue, loss, automation, operational efficiency
  - Role of scorecards in decision making
    - sole arbiter or decision support tool?

Create Business Plan

- Internal/External Development and Scorecard Type
  - Capability and resources
    - Staff, tools, expertise, data
    - Market segment
  - Custom, generic, judgmental
    - segment, data, time.

Create Project Plan

- Scope and timelines
- Deliverables (scorecard format and documentation,…)
- Implementation strategy
  - Testing, coding
  - Strategy development
  - FYI list.

- Seamless process from planning to development and implementation.

Create Project Plan

- Identify Project Risks
  - Data risks
    - Availability, quality, quantity
    - Weak data
  - Operational risks
    - Organizational priority
    - Implementation delays
    - System interpretation of data.

Create Project Plan

- Identify Project Team
  - Roles clearly defined
    - Signoff, executor, advisor, FYI
    - Critical path.

Development

Stage 2: Data Review and Project Parameters

Objectives

- Identify data requirements.
- Perform pre-modeling analysis.
  - Understand the business
  - Exclusions
  - What is a “bad”? – target definition
  - Sample Window/ Performance Window.

Data Availability and Quality

- Number of “goods”, “bads” and “rejects”
  - Initial idea at this stage, estimated from performance reports
- Internal data
  - Reliable, accessible
- External data
  - Accessible, format
  - Retro pull.

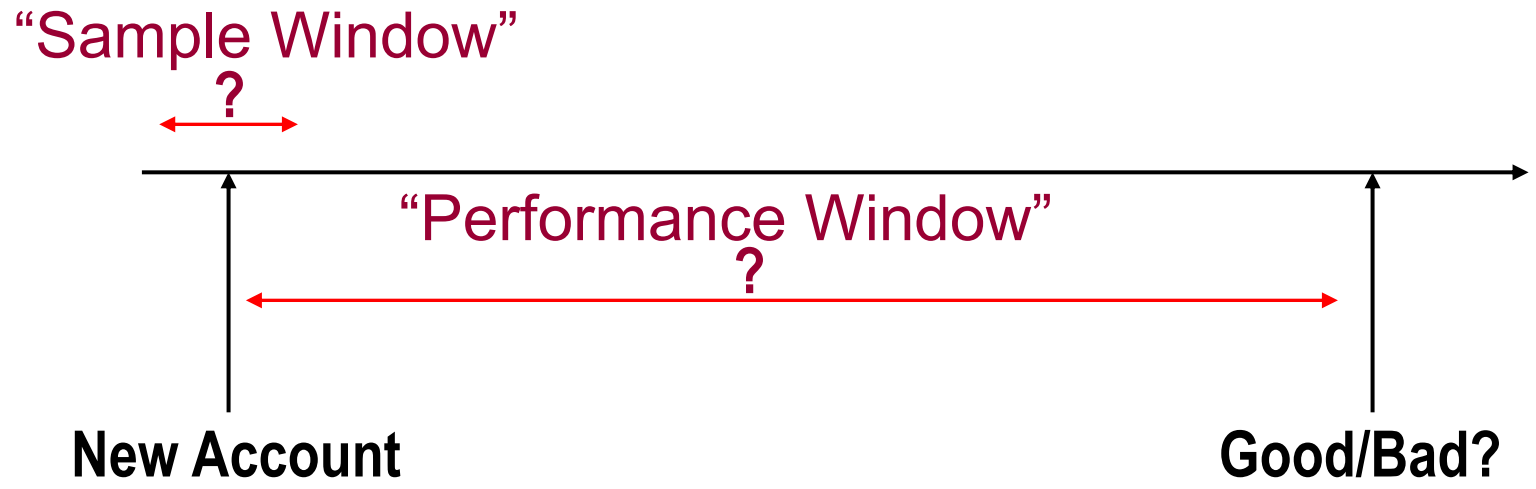
Data Gathering

- To determine “bad” definition and exclusions:
  - All applications over the last 2–5 years (or a large sample)
  - account/ID number
  - Date opened/applied
  - Accept/reject indicator
  - Arrears/payment history
  - Product/channel and other identifiers
  - Account status
  - Other items to understand the business.

Exclusions

- “Include those whom you would score during normal day to day operations”
  - VIP
  - Staff
  - Fraud
  - Pre-approved
  - Underage
  - Cancelled (sometimes).

Performance



Parameters

- Performance Window
  - How far back do I go to get my sample?
- Sample Window
  - Time frame from which sample will be taken.
- Definition of “bad”
- Bad and approval rates (when oversampling).

Parameters

- Seasonality
  - Plot approval rate/applications across time
  - Establish any ‘abnormal’ zones (for example, talk to marketing).
- Sample used in development must be from a normal business period, to get as accurate a picture as possible of the target population.

Parameters – "Bad"

- Plot "bad" rate by "month opened" (cohort)
- For different definitions of bad
  - 30/60/90 days past due
  - Charge off/write-off
  - Bankrupt
  - Claim
  - Profit based
  - Less than *x*% owed collected
- "Ever" versus "Current" bad
  - Ever bad should be used where possible
  - Considered "bad" if you reach status anytime during performance window.

Cohort Analysis – Example

Bad = 90 days					
Open Date	**1 Qtr**	**2 Qtr**	**3 Qtr**	**4 Qtr**	**5 Qtr**
Jan-99	0.00%	0.44%	0.87%	1.40%	2.40%
Feb-99	0.00%	0.37%	0.88%	1.70%	2.30%
Mar-99	0.00%	0.42%	0.92%	**1.86%**	**2.80%**
Apr-99	0.00%	**0.65%**	**1.20%**	**1.90%**	
May-99	0.00%	0.10%	0.80%	1.20%	
Jun-99	0.00%	0.14%	0.79%	1.50%	
Jul-99	0.00%	0.23%	0.88%		
Aug-99	0.00%	0.16%	0.73%		
Sep-99	0.00%	0.13%	0.64%		
Oct-99	0.20%	**0.54%**			
Nov-99	0.00%	**0.46%**			
Dec-99	0.00%	0.38%			
Jan-00	0.30%				
Feb-00	0.00%				
Mar-00	0.00%				

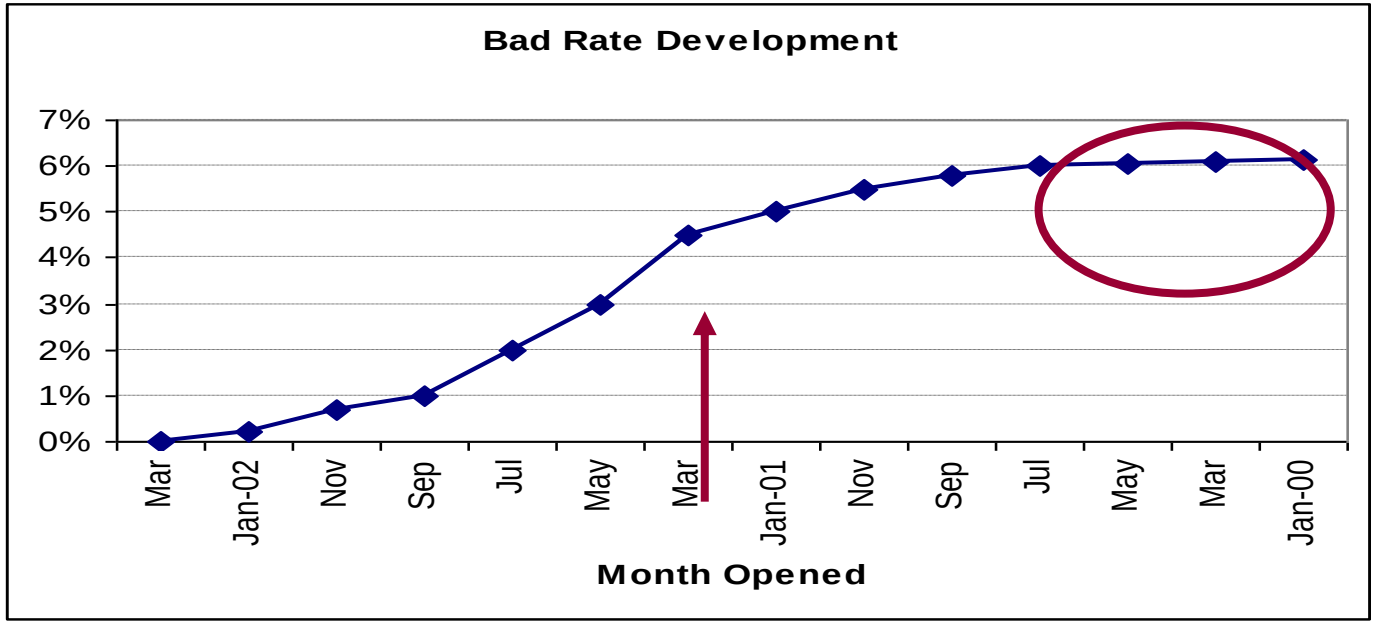
Current versus Ever – Example

- Current bad definition: No Delinquency
- Ever bad definition: 3 months delinquent.

Month	1	2	3	4	5	6	7	8	9	10	11	12
Delq	0	0	1	1	0	0	0	1	2	3	0	0

Month	13	14	15	16	17	18	19	20	21	22	23	24
Delq	0	0	1	2	0	0	0	1	0	1	0	0

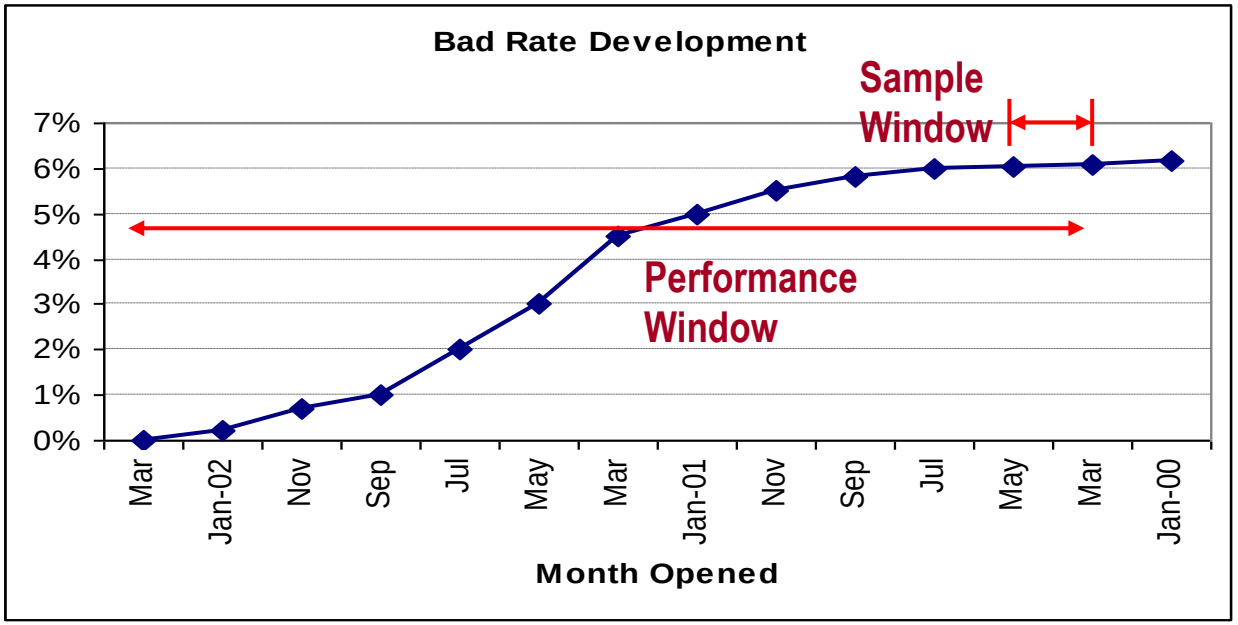
Determining Parameters



- mth opened from earliest to latest, and "bad rate" as of this month. For simplicity, this is straight delinquency .. No profit.
- notice at one point the bad rate levels off - this means everyone who was going to go bad has gone bad I.e. they have been given enough time. This is telling us that for this bad defn, accts from jan-march are mature enough.
-lesson 1: need sample that is mature enough, so that you wont be defining a "bad" as a good just because you haven't given them enough time.
-if you take accts from the middle (enter), some of the accts haven't matured yet so your bad rate is understated.

-Example: response scoring .. How long do you wait for the responses to come in. the period of measurement is 'perf window'.

Determining Parameters



So for each definition of "bad" you'll get a sample window of mature accounts, and a performance window indicating the time taken for the bad rate to mature. Also the approval rate for this sample window.

Couple of notes on this "maturing" process.

- 30 day definition will mature quicker than 90 day. Cause it takes ppl less time to go 30 day than 90 day. Chargeoff even more.
- for the same bad defn, credit card quicker than mortgage (18-24 mths vs. 3-5 yrs) .

- Why are we doing all this for the different definition?
- because each one will produce different counts and based on reasons on the next slide, we'll determine the best set of parameters.

Determining Parameters – Bad

- Organizational objectives/purpose
- Tighter definition – more precise, low counts
- Looser definition – differentiation sub-optimal
- Interpretable and trackable
- Consistency
- Reality – the best definition under the circumstances (lack of data, history).

Lets look at the considerations.
- objectives: this may seem obvious, but it is not to a lot of ppl. If you're building a scorecard to predict profit, then use profit. Some orgs want a delinquency based defn, but also include profit. E.g. if acct is chronically 2 mths late, but still profitable.. You can't set 2 mths as a "bad" - whereas in a pure delq scorecard this may be possible.
- tighter/looser: tighter means 90 day, 120 day, writeoff .. Better differentiation, but low count. Remember 2000 bads.
- looser means more count, but sub-opt diff.
- interpretable e.g. bad is 2 times 60 days, 3 times 30 days or 1 times 90 days. Sounds good, but hell to track and interpret. Keep it simple.
- consistency across other cards, products. Also if accounting writes off acct at 7 mths, then keep it consistent with that.
- typically most delq cards are 90 days.

- Reality: you take what you got. Lack of history allows only a 30 day definition .. Take it. Can't measure real bad rate .. Use proxy. (example LOC like an account)

Sample Definitions – Bad

- Ever 90 days delinquent
- Bankrupt
- Claim over $1000
- 3 x 30 days, or 2 x 60 days, or 1 x 90 days
- Negative NPV
- Not profitable
- 50% recovered within 3 months
- Fraud over $500
- *Closed within 6 months*.

Confirming “Bad” Definition

- Analytical
  - “Roll rate” analysis
  - Current versus worst delinquency comparison
  - Profitability analysis

- Consensus.

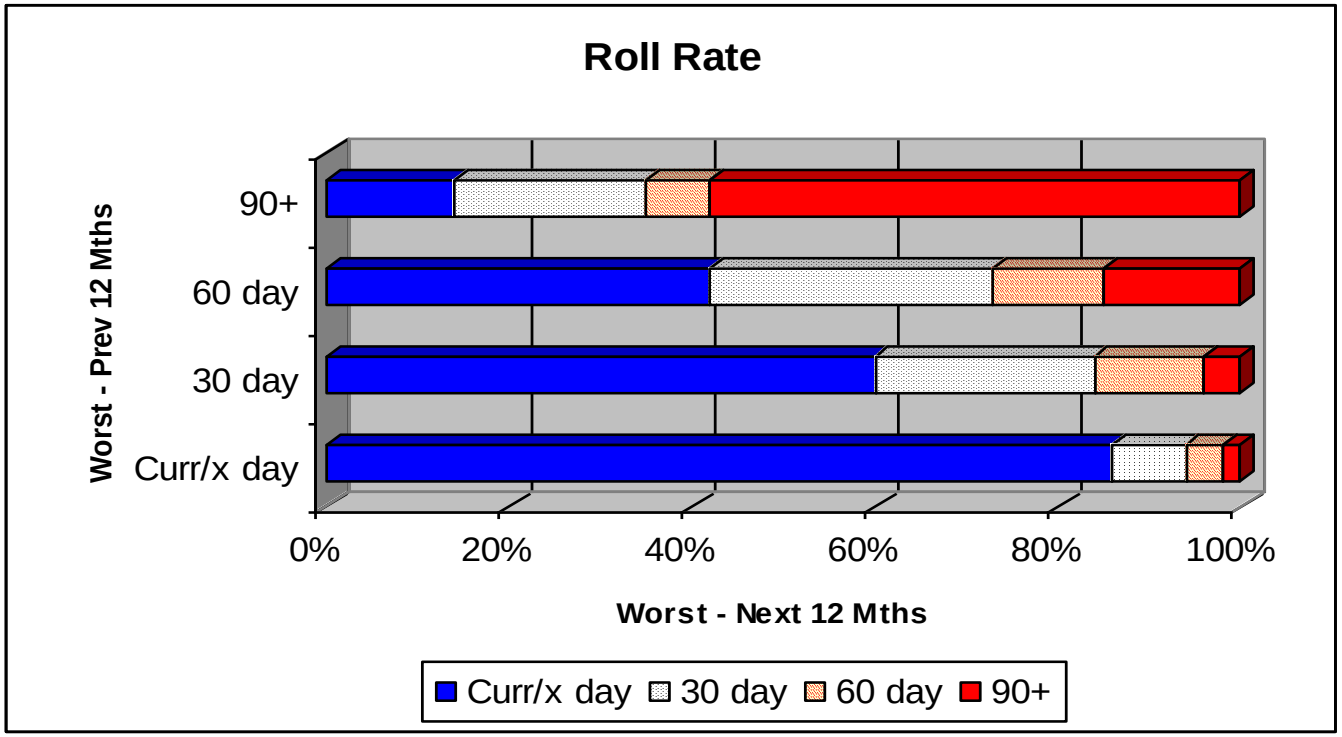
Roll Rate Analysis

- Compare Worst delinquency
  - for example, Previous 12 months versus Next 12 Months

Month	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
Arrears	0	0	1	2	0	0	0	1	2	3	0	0

Month	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*
Arrears	1	2	3	3	3	4	3	0	0	0	0	0

Roll Rate Analysis



You find out which 'bad defn' is truly bad' - also known as POINT OF NO RETURN.

Lets look at 30 day: out of everyone who had worst 30 day, majority became current, only a few became worse - this is not a good bad defn.
- out of those 60 days, some went over .. Most went back I.e. became better

-but those who were 90 day .. **Majority did not become better. This confirms our definition.**

-**In general .. Once you hit 90 days, you're not coming back. That's a true bad. Rem: this is based on 'bad' objective. If other, perhaps there is a different point in time..**

Roll Rate Analysis

- Look for ‘point of no return’.
- Consider objectives.
- Consider sample counts.
- Typically for delinquency, after 90 days most accounts do not cure.

Current versus Worst Comparison

				Worst Delinquency			
		Current	30 days	60 days	90 days	120 days	writeoff
Current	Current	100%	68%	34%	15%	4%	
Delinquency	30 days		16%	22%	8%	5%	
	60 days		8%	19%	17%	8%	
	90 days		4%	14%	32%	11%	
	120 days		2%	8%	18%	54%	
	writeoff		2%	3%	10%	18%	100%

32% 56% 40% 18% 44% 60% 72%

Parameters – Goods/Indeterminates

- Good
  - Never delinquent
  - Ever *x*- days delinquent
  - No claims
  - Profitable, positive NPV
  - No fraud
  - No bankruptcy
  - Recovery > 75%, $ value
  - Must be good throughout performance window
- Indeterminate
  - Mild delinquency, roll rate not conclusive either way
  - Inactive
  - Offer declined
  - Voluntary cancellations*
  - High balance < $50

Default – definice cílové prom. (good/bad)

- Obvykle je tato definice založena na klientově počtu dnů po splatnosti (Days Past Due, DPD) a částce po splatnosti. S částkou po splatnosti je spojena potřeba stanovení jisté míry tolerance, tedy stanovení co je považováno za významný dluh a co nikoli. Např. nemusí dávat smysl považovat za dluh částky menší než 100 Kč.
- Dále je třeba stanovit časový horizont (performance window), na kterém jsou dva zmíněné parametry sledovány.
- Za dobrého klienta lze např. označit klienta, který:
  - je po splatnosti méně než 60 dnů(s tolerancí 100 Kč) v prvních 6-ti měsících od první splátky,
  - je po splatnosti méně než 90 dnů (s tolerancí 30 Kč) v průběhu celé své platební historie (ever).

Default – definice cílové prom.

- Volba těchto parametrů závisí do značné míry na typu finančního produktu (jistě se bude lišit volba parametrů pro spotřebitelské úvěry pro malé částky se splatností kolem jednoho roku a pro hypotéky, které jsou obvykle spojeny s velmi vysokou finanční částkou a se splatností až několik desítek let) a na další využití této definice (řízení rizik, marketing, ...).

Default – definice cílové prom.

- Další praktickým problémem definice dobrého klienta je souběh několika smluv jednoho klienta. Například je možné, že zákazník je po lhůtě splatnosti na více smlouvách, ale s rozdílnými dny po splatnosti a s různými částkami. V tomto případě jsou většinou částky klienta dlužné v jednom konkrétním časovém okamžiku sečteny, a ze dnů po splatnosti na jednotlivých smlouvách je brána maximální hodnota. Tento přístup lze uplatnit pouze v některých případech, a to zejména v situaci, kdy jsou k dispozici kompletní účetní data. Situace je podstatně složitější v případě agregovaných údajů, např. na měsíční bázi.

Default – definice cílové prom.

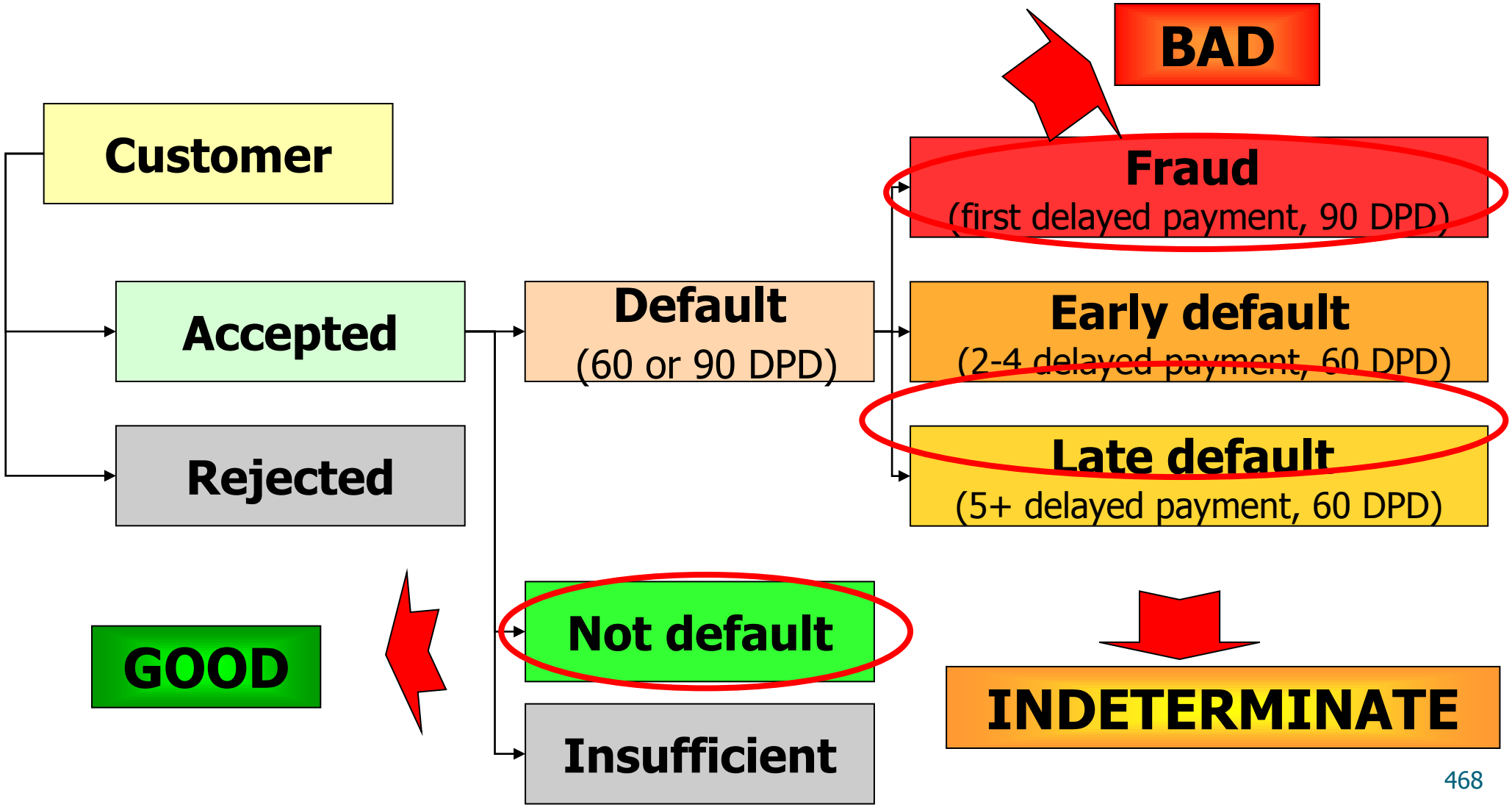
- Obecně uvažujeme následující typy klientů:

  - dobrý (good),
  - špatný (bad),
  - nedefinovaný (indeterminate),
  - s nedostatečnou úvěrovou historií (insufficient),
  - vyřazený (excluded),
  - zamítnutý (rejected).

Default – definice cílové prom.

- První dva typy byly diskutovány. Třetí typ, tj. indeterminate, je na hranici mezi dobrým a špatným klientem a při jeho použití přímo ovlivňuje definici dobrých/špatných klientů. Uvažujeme-li pouze DPD, klienti s vysokými DPD (např. 90 +) jsou typicky označeni za špatné, nedelikventní klienti (jejich DPD je rovno nule) jsou označeni za dobré. Za indeterminate jsou pak označeni delikventní klienti, kteří nepřekročí danou hranici DPD.
- Čtvrtý typ klientů jsou typicky klienti s velmi krátkou platební historií, u kterých je nemožná korektní definice cílové proměnné.
- Vyřazení klienti jsou klienti, jejichž data jsou natolik špatná, že by vedla ke zkreslení modelu(např. fraudy). Další skupinu tvoří klienti, kteří nejsou standardně hodnoceni daným modelem (VIP klienti)
- Poslední typ klientů jsou ti klienti, jejichž žádost o úvěr byla zamítnuta.

Definice dobrého/špatného klienta



Performance Definitions

- “Goods” and “bads” (and rejects) are used for model development.
- Indeterminates included for Gains chart and forecasting.

Segmentation

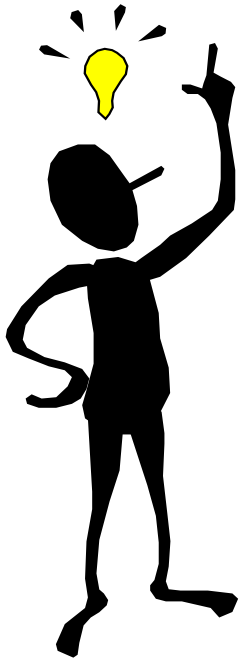
- Can one scorecard work efficiently for all the different populations within your portfolio?
- Or would more than one scorecard be better?

- Segmentation maximizes predictiveness for unique segments within your population.

Segmentation

- Experience (Heuristic)
  - Knowledge/experience, operational/industry based, common sense.
- Statistical
  - Let the data speak.
- "**Distinct** applicant/account sub-populations"
- "Better predictive power than single model".

Experience Based Segmentation



- Product
  - Card type, loan type (auto, home, unsecured), lease, used versus new, brand
- Demographics
  - Geographical (region, urban/rural, state/province, internal definition, neighborhood), age, time at bureau
- Source of business
  - Channel (net, branch, store-front, 'take one', brokers)
- Applicant type
  - new/existing, first time home buyer, groups (retired, students, engineers), thin/thick file, clean/dirty file
- Product Owned
  - Credit Card for existing mortgage/loan holders.

Experience Based Segmentation

- Consider future plans, not just historic operations

- How do we detect new segments?
  - Marketing/risk analysis:
    - Bad rates
    - Approval rate
    - Profit, and so on.
  - Look for significant performance difference.

Experience Based Segmentation

- Need to confirm experience using analytics.
- Definition of segments
  - What is a thin file?
  - What is ‘young’ versus ‘old’?
  - What is the best demographic split?
  - What break is best for ‘tenure at bank’?

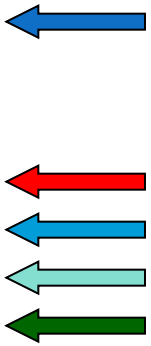
Confirming Experience

- Rule of thumb:
- "When the same information predicts differently across unique segments"

Bad Rate			
	Age > 30	Age < 30	Unseg
Res Status			
Rent	2.1%	4.8%	2.9%
Own	1.3%	1.8%	1.4%
Parents	3.8%	2.0%	3.2%
Trades			
0	5.0%	2.0%	4.0%
1-3	2.0%	3.4%	2.5%
4+	1.4%	5.8%	2.3%

Confirming Experience

Attributes	Bad Rates
Age	
Over 40 yrs	1.80%
30-40 yrs	2.50%
Under 30	6.90%
Source of business	
Internet	20%
Branch	3%
Broker	8%
Phone	14%
Applicant Type	
First Time buyer	5%
Renewal Mortgage	1%



That Is the Easy Way

- You can also build full segmented models, and compare "lift", sensitivity, and so on, with a base model.
- It is best to perform this analysis for both experience and statistically based segmentation.

Comparing Improvement

- Use different methods to measure improvement (lift, KS, c-stat, precision, and so on.)

Segment	Total c-stat	Seg c-stat	Improvement
Age < 30	0.65	0.69	6.15%
Age > 30	0.68	0.71	4.41%
Tenure < 2	0.67	0.72	7.46%
Tenure > 2	0.66	0.75	13.64%
Gold card	0.68	0.69	1.47%
Platinum card	0.67	0.68	1.49%

Comparing Improvement

- Portfolio stats will put improvements into measurable portfolio terms.

		After Segmentation		Before Segmentation	
Segment	**Size**	**Approve**	**Bad**	**Approve**	**Bad**
Total	100%	%	%	%	%
Age < 30	65%	%	%	%	%
Age > 30	35%	%	%	%	%
Tenure < 2	12%	%	%	%	%
Tenure > 2	88%	%	%	%	%
Gold card	23%	%	%	%	%
Platinum card	77%	%	%	%	%

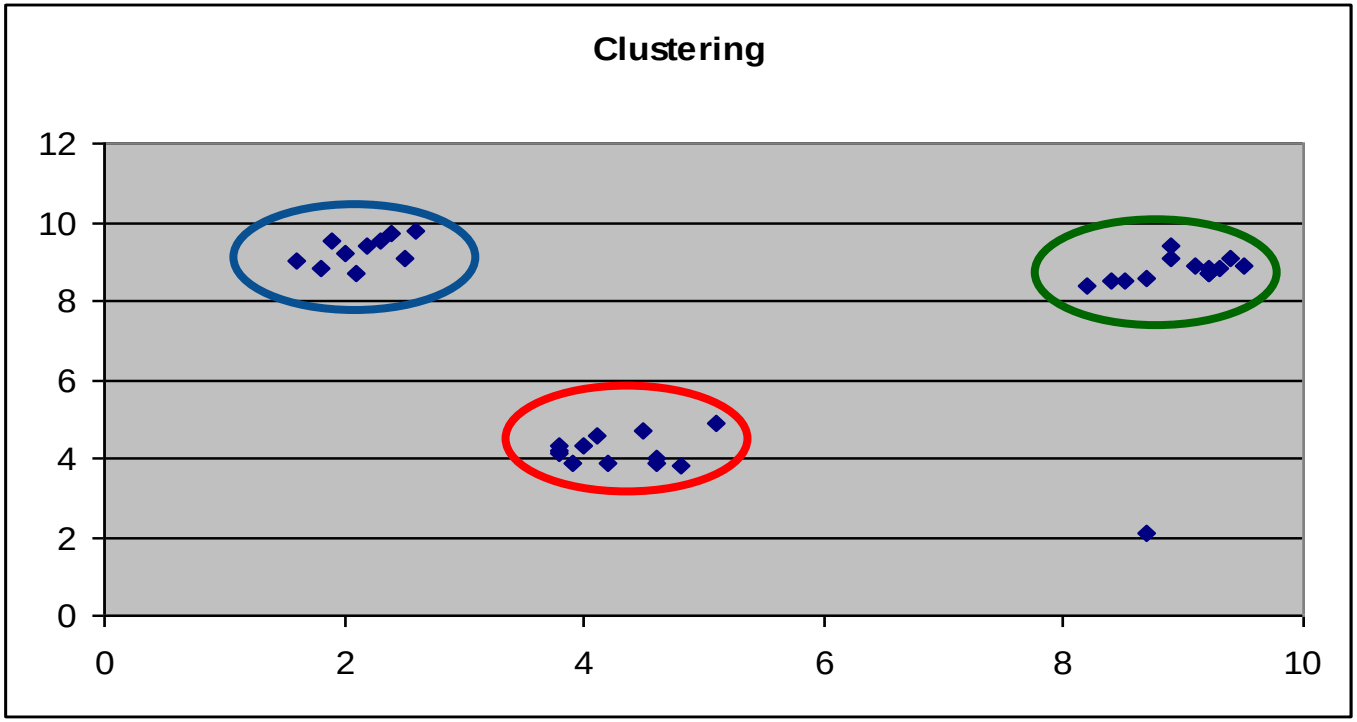
Choosing Segmentation

- Cost of scorecards (internal/external)
- Implementation
- Processing
- Data storage
- Monitoring/strategy development
- Segment size
- Do I have to?

Statistically Based Segmentation

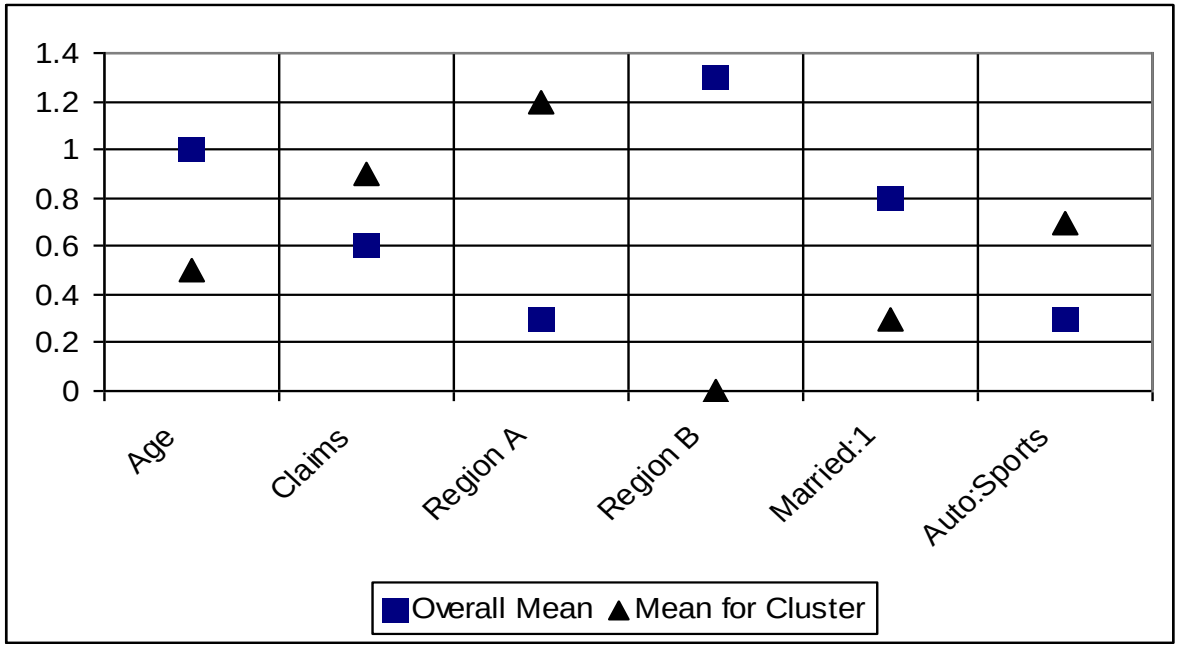
- Less preconceived notions
  - Clustering
  - Decision Trees.

Clustering



Showing 3 distinct groups and one outlier.

Clustering



Here is an insurance example of one cluster.

- What do we see here?

- lower than avg age
- more claims
- live in region A only
- likely to be single

and drive a sports car.

- this is obviously a high risk segment.
(confirm this group with claims analysis)

- Similar groups according to characteristics, not performance – so confirm performance for the clusters and combine those with similar risk behavior. We're not building a marketing profile, but a RISK PROFILE.

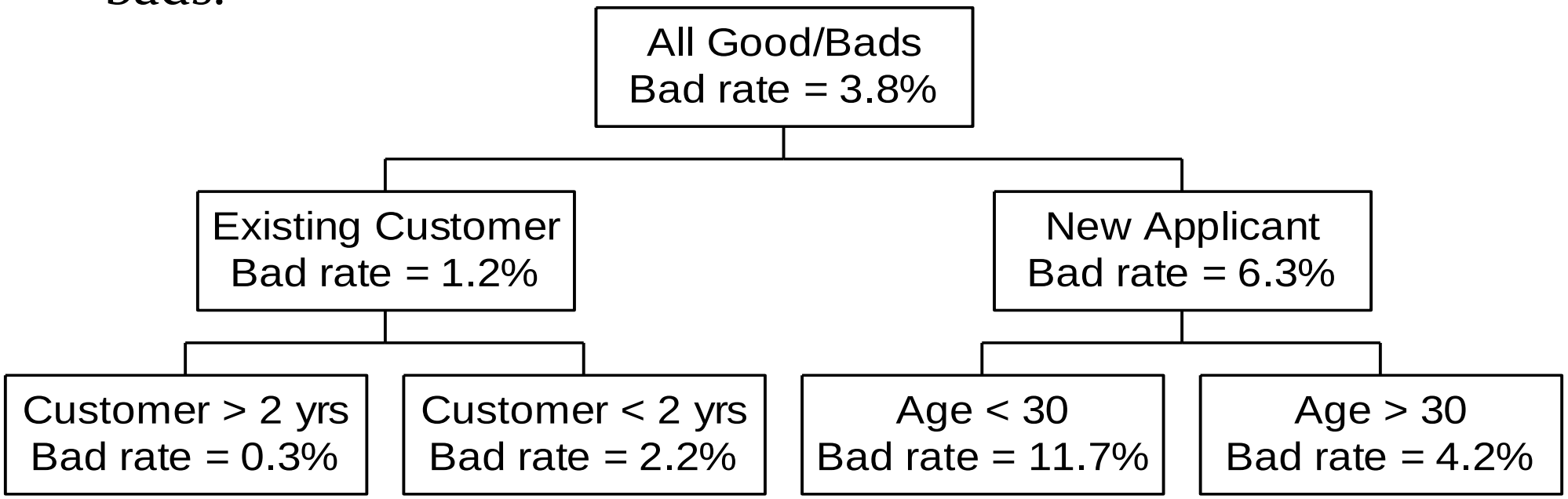
Clustering

- Defining characteristics for each group
- From previous example,
  - Young males region A
  - Young females region A, and so on.

- **Performance analysis to confirm segmentation**.

Decision Trees

- Isolates segments based on performance (target)
- Easily interpretable and differentiates between goods and bads.



So Now We Know …

- the business
- sample and performance windows
- “bad”, “good”, “indeterminate”
- exclusions
- bad rate, approval rate
- number of scorecards needed, and their segments.

Methodology/Format

- Implementation platform and format
- Interpretability, implementation
- Legal compliance
- Data quality, sample size, target type
- Tracking and diagnosis
- Specify parameters for scorecard (range of scores, “points to double the odds”).

Why ‘Scorecard’ Format?

- Easiest to interpret, justify, implement
- Reasons for decline/low scores can be explained to auditors, Mgmt, regulators, adjudicators
- No black box
- Diagnosis, tracking, monitoring
- Development process fairly simple to understand.

Review Implementation Plan

- Number of scorecards
- Data requirements
- Manage expectations
- Continuity.

Everyone is aware of what's going on.
This is a business process, not a mystery novel. You'd be surprised how many people in companies like to spring surprises on other departments.

Cvičení

Jsou k dispozici následující data:

Accepts.sas7bdat (64589 řádků)
Rejects.sas7bdat (35411 ř.)
Applicants.sas7bdat (100.000 ř.)

...24 sloupců
ID of applicant, Date of application/opening, Accept / Reject, 30-days deliquency, 30-days deliquency date, 60-days deliquency, 60-days deliquency date, 90-days deliquency, 90-days deliquency date, Worst previous deliquency, Current deliquency, Age, Age groups, Sex, Existing client?, Phone member?, Region, Income, Income groups, Debt, Income/Debt ratio, Income/Debt ratio groups, Probability of 60-days deliquency (old), Score (old).

Základní popis dat:

```
title 'Accepts';
proc means data=indata.accepts n nmiss min median mean
max;
  var age income debt idratio;
run;

title 'Accepts';
proc freq data=indata.accepts;
table sex client phone region;
table (sex client phone region)*bad60;
table bad30*(bad60 bad90) bad60*bad90;
run;

title 'All applicants';
goptions ftext='arial';
proc catalog c=gseg kill;
quit;
proc gchart data=indata.applicants;
vbar age / midpoints=18 to 75 name='_1data_a';
vbar income / name='_1data_b';
vbar debt / name='_1data_c';
vbar idratio / name='_1data_d';
vbar type / name='_1data_e';
vbar scoreold / levels=10 name='_1data_f';
vbar pbad60old / levels=30 name='_1data_f';
run;
quit;

proc univariate data=indata.applicants normal;
var age income debt idratio;
histogram age income debt idratio;
run;
```

Cvičení

Vybrané výstupy uvedeného kódu:

The FREQ Procedure

Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
M	45138	69.88	45138	69.88
Z	19451	30.12	64589	100.00

Sex

Client	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	60188	93.19	60188	93.19
1	4401	6.81	64589	100.00

Existing client?

Phone	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	8081	12.51	8081	12.51
1	56508	87.49	64589	100.00

Phone member?

Region	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	12537	19.41	12537	19.41
2	16335	25.29	28872	44.70
3	10679	16.53	39551	61.23
4	10797	16.72	50348	77.95
5	7199	11.15	57547	89.10
6	7042	10.90	64589	100.00

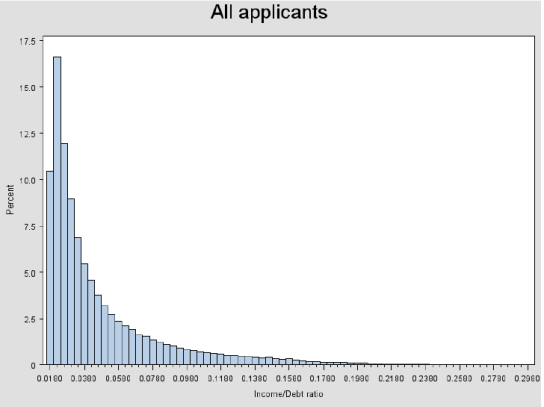
Region

Frequency
Percent
Row Pct
Col Pct

Table of Phone by Bad60

Phone(Phone member?)	Bad60(60-days deliquency) 0	1	Total
0	7805 12.08 96.58 12.47	276 0.43 3.42 13.97	8081 12.51
1	54809 84.86 96.99 87.53	1699 2.63 3.01 86.03	56508 87.49
Total	62614 96.94	1975 3.06	64589 100.00

The UNIVARIATE Procedure



Accepts

The MEANS Procedure

Variable	Label	N	N Miss	Minimum	Median	Mean	Maximum
Age	Age	64589	0	18.0000000	43.0000000	43.3129945	74.0000000
Income	Income	64589	0	15000.07	19631.47	19854.56	35790.94
Debt	Debt	64589	0	100444.85	560744.83	576945.05	1611457.12
IDRatio	Income/Debt ratio	64589	0	0.0175377	0.0345483	0.0500680	0.2994807

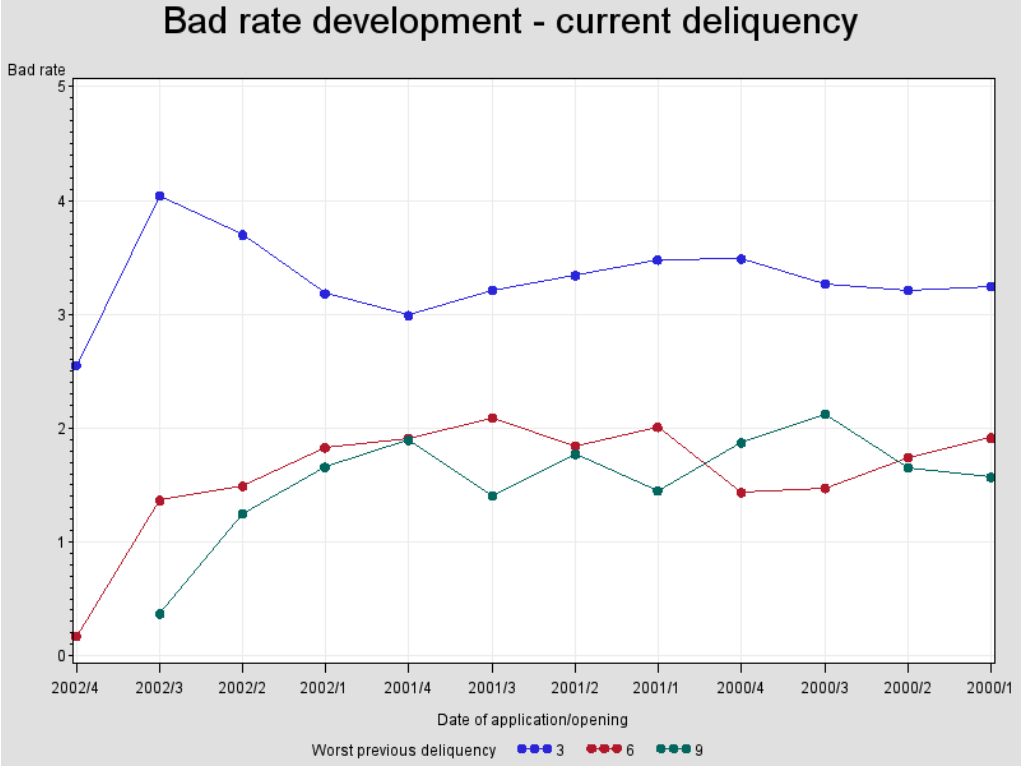
The UNIVARIATE Procedure
Variable: IDRatio (Income/Debt ratio)

Moments			
N	100000	Sum Weights	100000
Mean	0.04766914	Sum Observations	4766.91379
Std Deviation	0.03680037	Variance	0.00135427
Skewness	2.10159641	Kurtosis	4.8128053
Uncorrected SS	362.660032	Corrected SS	135.425362
Coeff Variation	77.1995691	Std Error Mean	0.00011637

Basic Statistical Measures			
Location		Variability	
Mean	0.047669	Std Deviation	0.03680
Median	0.033093	Variance	0.00135
Mode	.	Range	0.28194
		Interquartile Range	0.03334

Cvičení

```
/* 2a. Bad rate development, roll rate analysis */


%let performancewindow='31dec2002'd>=datappl;
%let deliq=worstdeliq;

proc freq data=indata.accepts /*noprint*/;
table datappl*&deliq / out=&deliq (keep=datappl &deliq pct_row
where=(&deliq ne '0')) outpct missing;
format datappl yyqs7.;
where &performancewindow;
run;

ods html path="&appl_root" file="2.&deliq..html";
goptions reset=all ftext='arial';
symbol1 i=j v=dot;
axis1 label=('Bad rate');
proc catalog c=gseg kill;
quit;
title 'Bad rate development - current deliquency';
proc gplot data=&deliq;
plot pct_row*datappl=&deliq / name='_2curdel' grid hreverse
vaxis=axis1 hminor=0;
run;
quit;
ods html close;
```



Cvičení

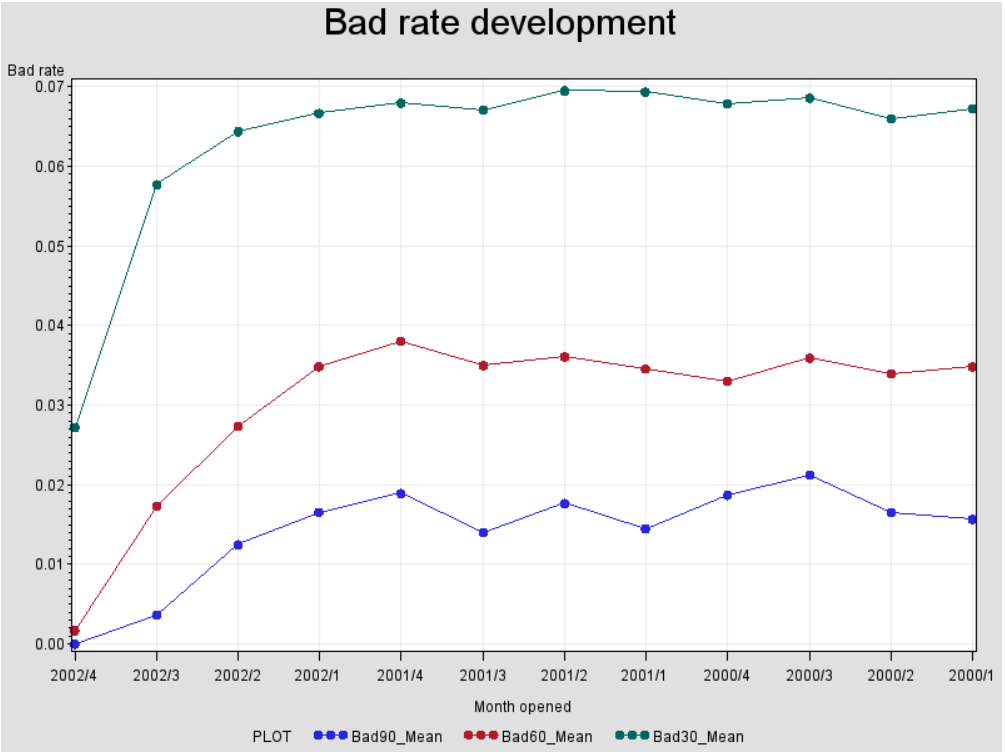
```
/* analyza kohort */

%let target=bad30;
%let date=dat30;

data cohorts;
set indata.accepts (keep=datappl bad: dat:);
if &target then qtr=int(yrdif(datappl,&date,'act/act')*4)+1;
datappl=intnx('month',datappl,0);
format datappl mmyys7.;
run;
proc freq data=cohorts noprint;
table datappl / out=cohorts1 (drop=percent
rename=(count=counttotal));
table datappl*qtr / out=cohorts (drop=percent);
run;
data cohorts;
merge cohorts cohorts1;
by datappl;
if first.datappl then cumpct=.;
if qtr ne . then do;
    cumpct+(count/counttotal);
    output;
end;
run;

ods html path="&appl_root" file='2.cohorts.html';
title "Cohort analysis for &target";
proc tabulate data=cohorts missing format=percent8.4;
class datappl qtr;
var cumpct;
table datappl,qtr*cumpct="'*sum=";
run;
ods html close;
```

Cohort analysis for bad30

Date of application/opening	qtr 1	2	3
01/2000	5.987%	6.652%	.
02/2000	6.327%	6.887%	7.055%
03/2000	5.491%	6.441%	6.494%
04/2000	5.458%	6.254%	6.367%
05/2000	6.600%	7.648%	.
06/2000	5.219%	5.724%	.
07/2000	5.437%	6.130%	.
08/2000	6.321%	7.401%	7.455%
09/2000	6.345%	7.019%	.
10/2000	6.023%	6.782%	.
11/2000	5.613%	6.104%	6.213%
12/2000	6.400%	7.385%	.
01/2001	6.064%	7.109%	.
02/2001	5.836%	6.809%	.
03/2001	6.271%	6.766%	6.876%
04/2001	6.935%	7.870%	7.925%
05/2001	6.201%	6.884%	.
06/2001	5.391%	5.996%	6.051%
07/2001	4.859%	5.729%	.
08/2001	6.339%	7.377%	.
09/2001	6.378%	7.118%	.
10/2001	6.448%	7.200%	.
11/2001	5.956%	6.415%	.
12/2001	5.521%	6.704%	6.761%
01/2002	5.668%	6.812%	.
02/2002	5.913%	6.684%	6.812%
03/2002	5.924%	6.408%	.
04/2002	5.776%	6.436%	.
05/2002	5.304%	5.974%	.
06/2002	6.079%	6.900%	.
07/2002	5.374%	5.817%	.
08/2002	5.405%	5.622%	.
09/2002	5.493%	5.889%	.
10/2002	4.608%	.	.
11/2002	2.563%	.	.

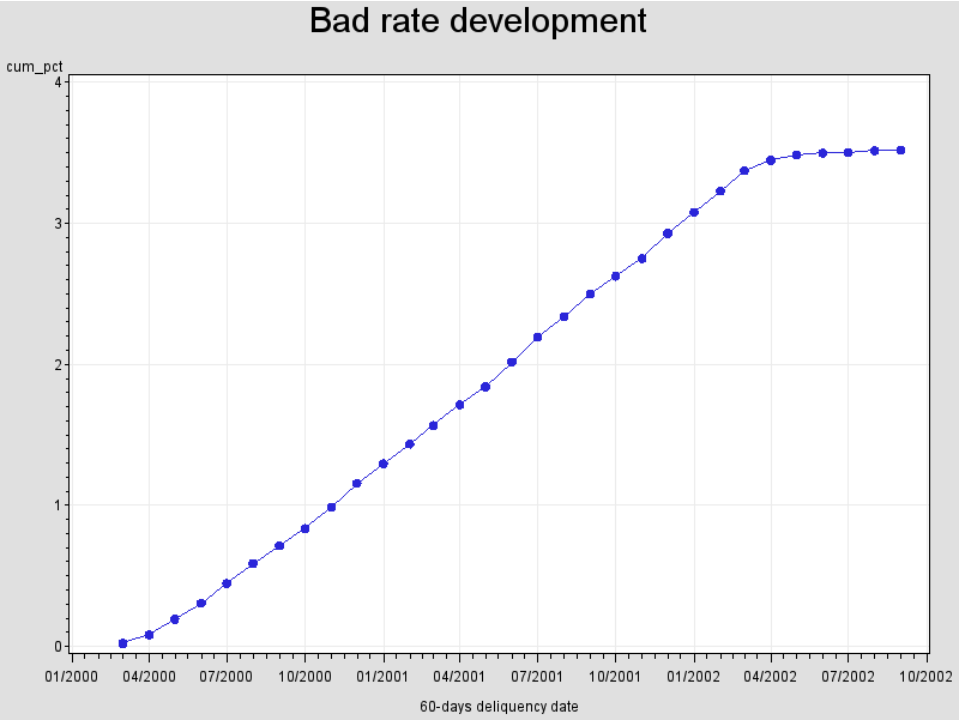
Cvičení

```
/* performance window */

%let performancewindow='31dec2002'd>=datappl;

proc tabulate data=indata.accepts out=brdev
(drop=_type_ _table_ _page_);
class datappl;
var bad90 bad60 bad30;
table datappl,(bad90 bad60 bad30)*mean*format=percent8.2;
format datappl yyqs7.;
where &performancewindow;
label datappl='Month opened';
run;

ods html path="&appl_root" file='2.perf.html';
goptions reset=all ftext='arial';
symbol1 i=j v=dot;
axis1 label=('Bad rate');
proc catalog c=gseg kill;
quit;
title 'Bad rate development';
proc gplot data=brdev;
plot (bad:)*datappl / name='_2perf' grid overlay legend hreverse
vaxis=axis1 hminor=0;
run;
quit;
ods html close;
```



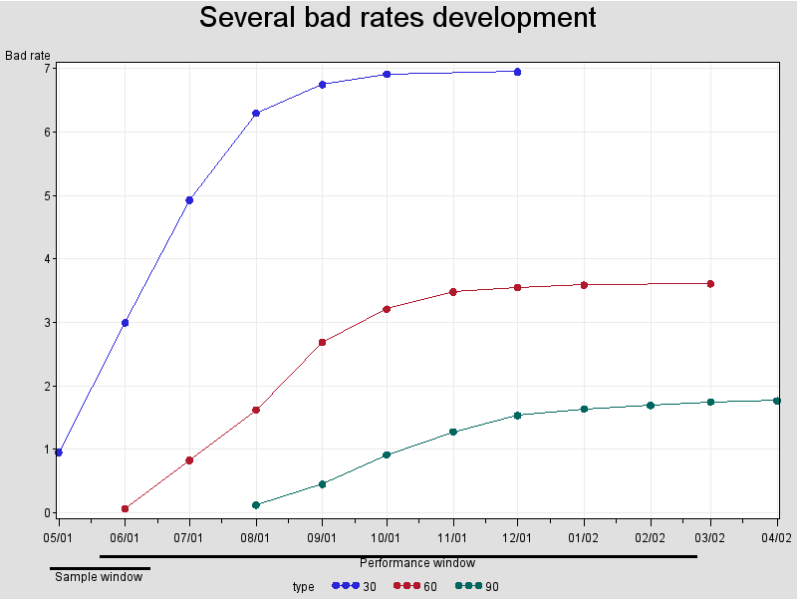
Cvičení

```
/* bad rate development */

%let samplewindow='30jun2001'd>=datappl>='01apr2001'd;
%let samplewindow='31dec2001'd>=datappl;

proc freq data=indata.accepts noprint;
table dat60 / out=development missing;
format dat60 mmyys7.;
where &samplewindow;
run;
data development;
set development;
if _n_>1 then do;
    dat60=intnx('month',dat60,0);
    cum_pct+percent;
    output;
end;
label datappl='Month of opening';
run;

ods html path="&appl_root" file='2.badratedev.html';
goptions reset=all ftext='arial';
symbol1 i=j v=dot;
axis1 label=('Bad rate');
proc catalog c=gseg kill;
quit;
title 'Bad rate development';
proc gplot data=development;
plot cum_pct*dat60 / name='_2brd' grid;
run;
quit;
ods html close;
```



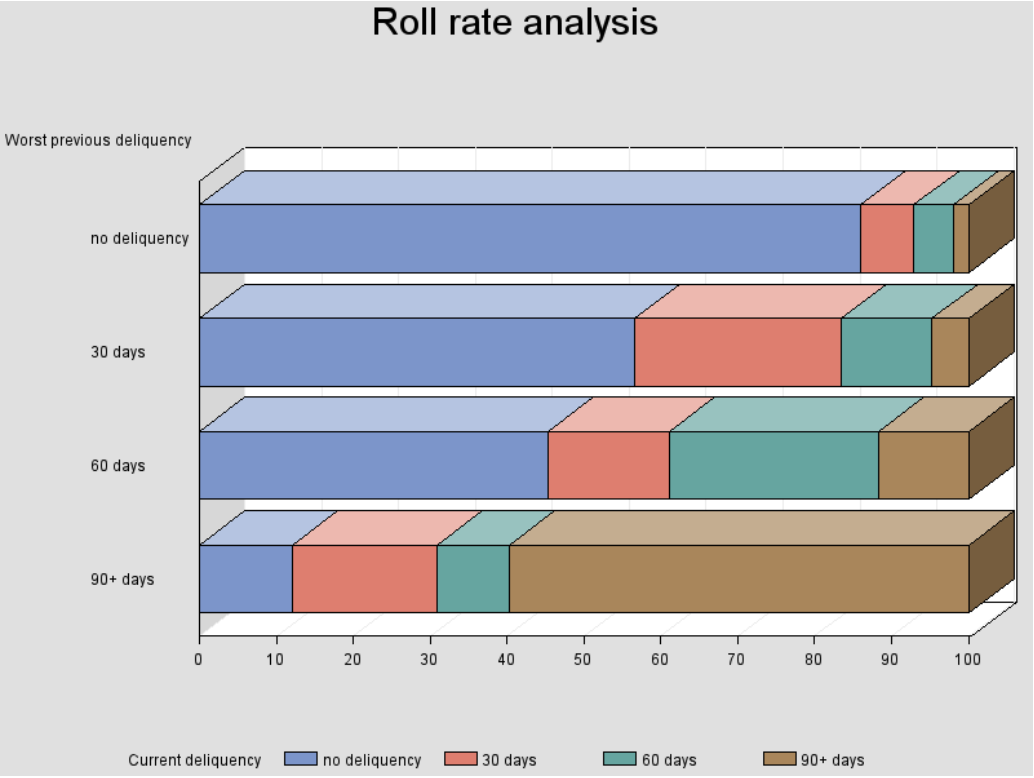
Cvičení

```
/* BRDEV macro */

%macro brdev(data,out,datevar,targetvar,samplewindow);
proc freq data=&data noprint;
table &datevar / out=&out missing;
format &datevar mmyys7.;
where &samplewindow;
run;
data &out (keep=date cum_pct);
set &out;
if _n_>1 then do;
    date=intnx('month',&datevar,0);
    cum_pct+percent;
    output;
end;
format date mmyys7.;
run;
%mend brdev;

%let samplewindow='30jun2001'd>=datappl>='01apr2001'd;
%brdev(indata.accepts,development,dat60,bad60,&samplewindow)

/* several bad rate development */

%let samplewindow='30jun2001'd>=datappl>='01apr2001'd;
%brdev(indata.accepts,development30,dat30,bad30,&samplewindow)
%brdev(indata.accepts,development60,dat60,bad60,&samplewindow)
%brdev(indata.accepts,development90,dat90,bad90,&samplewindow)

data developmentsev;
set development30 (in=__30) development60 (in=__60) development90;
if __30 then type='30';
else if __60 then type='60';
else type='90';
Run;

data anno;
function='label';x=20;y=2;text='Sample window';output;
size=2;function='move';x=10;y=2.5;output;
function='draw';x=30;y=2.5;output;
function='move';x=20;y=3.5;output;
function='draw';x=140;y=3.5;output;
run;

ods html path="&appl_root" file='2.badratedev_several.html';
goptions reset=all ftext='arial';
symbol1 i=j v=dot;
axis1 label=('Bad rate');
proc catalog c=gseg kill;
quit;
title 'Several bad rates development';
proc gplot data=developmentsev annotate=anno;
plot cum_pct*date=type / grid vminor=0 name='_2brds' vaxis=axis1;
format date mmyys5.;
label date='Performance window';
run;
quit;
ods html close;
```

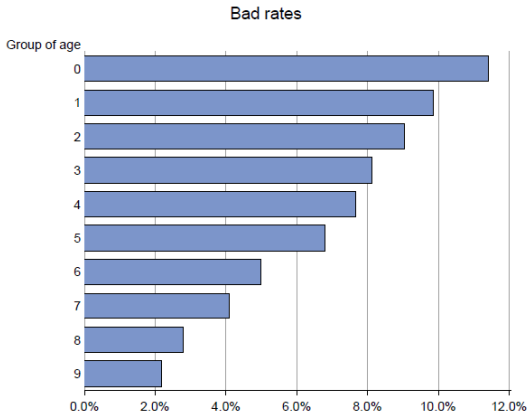


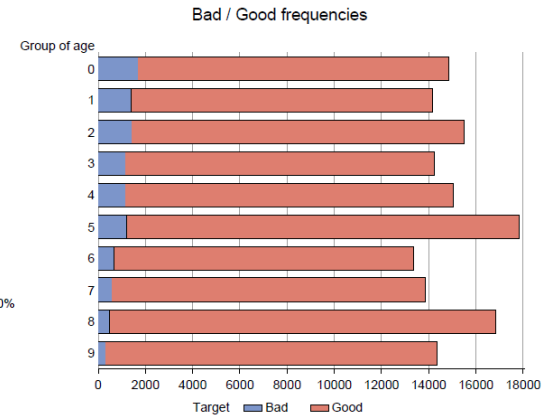
Cvičení

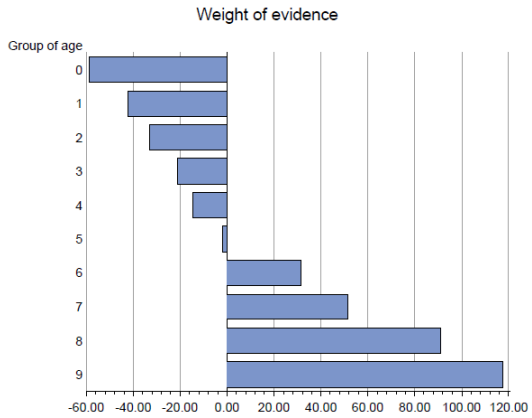
```
/* Roll rate analysis */

ods html path="&appl_root" file='2.roll_rate.html';

proc format;
value $deliq (notsorted)
    '0'=' no deliquency'
    '3'='30 days'
    '6'='60 days'
    '9'='90+ days';
run;

proc tabulate data=indata.accepts out=rollrate missing;
class curdeliq worstdeliq;
tables worstdeliq,curdeliq*rowpctn;
format curdeliq $deliq. worstdeliq $deliq.;
title 'Roll rate analysis';
run;
proc gchart data=rollrate;
hbar3d worstdeliq / sumvar=pctn_01 subgroup=curdeliq nostats
clipref autoref raxis=axis1;
axis1 label=none minor=none;
run;
quit;

ods html close;
```



8. Příprava dat II







Development

Stage 3: Development Database Creation

Development Sample Specification

- Development sample spec. means specifying what we need in the database we will use for development. We are not going to take a dump of everything from the CDW or datamart.
- Make the development process manageable and efficient:
  - list of characteristics (or "variables" to be considered for devp. You don't want to have the entire DW.)
  - sample sizes (for each segment and category. No point regressing on 100k when 3k will suffice.)
  - parameters from previous section.
- Do all this bearing in mind the number of scorecards you want developed and for which segments.

Characteristic Selection

How do you select characteristics? Reinforce: there is a need for some thought to be put into process in selecting characteristics ..
You get together with risk, mktg, product. And get operations areas such as collections aboard (WHO knows your bad guys better than anyone else?)

- Expected predictive power
- Reliability: (is this manipulated? or prone to be manipulated?, e.g. salary. Check historical data - cannot be confirmed or too expensive to confirm. Can it be interpreted e.g. occupation/industry type is the worst cases. Do poeple usually leave this blank.)
  - manipulation (non-confirmable)
  - interpretation (present and future)
  - missing
- Legal issues (Cant ask/get some info?.. Might get into trouble with some?)

Characteristic Selection

- Ease in collection
  - Do you want to spend time chasing missing info for a credit card?... may be OK for a mortgage. How easy it is to get this piece of info?
- Policy rules
  - Don't include anything that is unchangeable PR, e.g. bankruptcy. If you are going to decline all bankrupcy, no need to use it in scorecard.
- Derived variables – ratios
  - Can do a lot of ratios .. But put some business thought into it.
- Future direction.
  - Will this info be collected in the future (e.g. app form redesign)?
  - Industry direction - not relevant today but will change. can include in card or collect for future e.g. higher credit lines. Talk to credit bureaus industry trend and how they affect the scorecard.

What are you doing: you're looking at objectives, company operations, business knowledge, ground realities etc.

This is not just a stats exercise!!!

Sampling

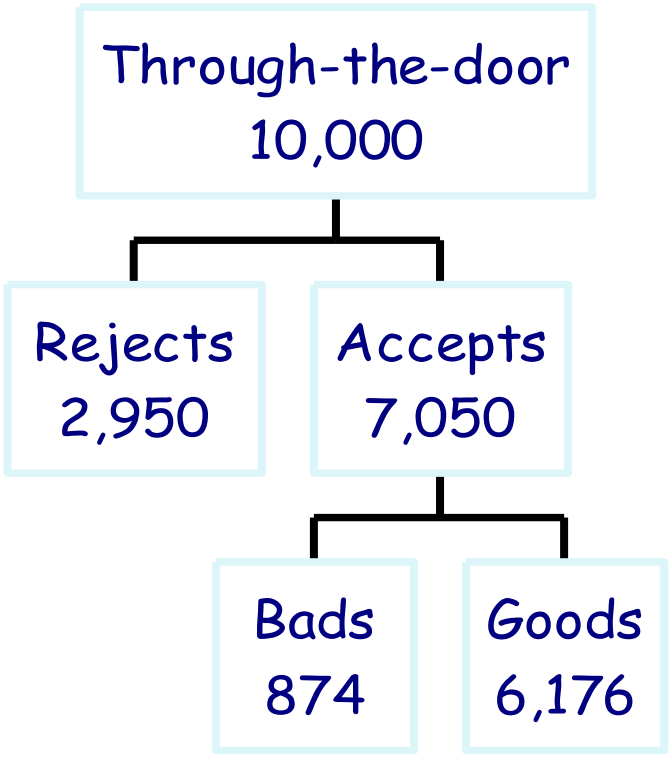
- Development, validation
  - 70:30, 80:20
  - If sample is small, do 100%, but validate with several 50–80%.
- Good, bad, reject
  - 2000 of each (or higher)
  - Oversampling (oversampling is common when modeling rare events … it leads to better predictions)
  - Proportional sample – not recommended for low bad rates.
  - Take what you got for bads and sample the goods.
- Ensure that each group has sufficient numbers for meaningful analysis.

Data Collection and Database Construction

- Random and representative
  - for each segment applicants (and accounts)
- One for unsegmented (to measure lift from segmentation)
- Data quirks, changes (preferably documented)
  - e.g. code for renters changed from R to E .. Stopped collecting some data item, new data fields, started collecting data recently etc. etc.
- Objective: Data collected, as specified.

Adjusting for Prior Probabilities

- When oversampling
- Adjust to actual:
  - Approval rate
  - Bad rate
- Analysis and reports reflect reality
- Do not need if you only want to know relationships or rank ordering.



Adjusting for Oversampling

- Separate sampling is standard practice (helps when you just did ‘bad’ definition)
- Prior probabilities must be known
- Can adjust before fitting the model or after.

- Two ways:
  - Offset
  - Sampling weights (frequency variable).

Offset Method

- Logit $(p_i)=\beta_o+ \beta_1 x_1+ ....+ \beta_k x_k$
- When oversampling, logits shifted by the *offset*:
- Logit $(p^*_i)= ln\ (\rho_1 \pi_o / \rho_o \pi_1)\ + \beta_o+ \beta_1 x_1+ ....+ \beta_k x_k$
- Where
  - ρ_1 and ρ_o= proportion of target classes in the sample
  - π_1 and π_o= proportion of target classes in the population.

Offset Method

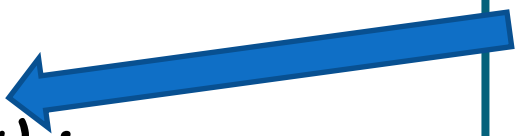
- Adjustment post-model (after model development):

- $p\hat{}_i = (p\hat{}^*_i \rho_0 \pi_1) \; / \; [(1 - p\hat{}^*_i)\, \rho_1 \pi_0 + p\hat{}^*_i \rho_0 \pi_1)]$

- Where $p\hat{}^*_i$ is the unadjusted estimate of posterior probability.

SAS Programs – Pre-model Adjustment

```
data develop;
set develop;
   off=(offset calc);
run;

proc logistic data=develop ...;
model ins=……./ offset=off;
run;

proc score ….;
   p=1 / (1+exp(-ins));
proc print;
var p ….;
run;
```



$ln\,(\rho_1\pi_0 / \rho_0\pi_1)$

SAS Program – Post-model Adjustment

```
proc logistic data=develop...;
run;

proc score ... out=scored...;
run;

data scored;
set scored;
   off = (offset calc);
   p=1 / (1+exp(-(ins-off)));
run;

proc print data=scored ..;
var p ...;
run;
```

Sampling Weights

- Adjusts data to reflect true population
  - Weights: π_1/ρ_1 and π_0/ρ_0
  - Or set weight of bad=1 and weight of good = p(good)/p(bad) for population.
    - For example, p(bad)=4%, 2000 goods, 2000 bads. Sample will show 2000 bads and 48,000 goods.
  - Normalization causes less distortion in p values and standard errors.
  - Use FREQ variable in EM or calculate sample weight and use weight=sampwt in the LOGISTIC procedure.

SAS Program

- When using the WEIGHT statement, some output is not correct.

```
data develop;
set develop;
sampwt=( π0/ ρ0)* (ins=0) +
( π1/ ρ1)* (ins=1);
run;
proc logistic data=develop …;
weight=sampwt;
model ins=…….;
run;
```

What Is the Difference?

- The parameter estimates will be different.
- When linear-logistic model is correctly specified, offset is better.
- When logistic model is an approximation of some non-linear model, weights are better.
- For scorecards, weighting is better since it corrects the parameter estimates used to derive scores (prior probabilities only affect the predicted probabilities).

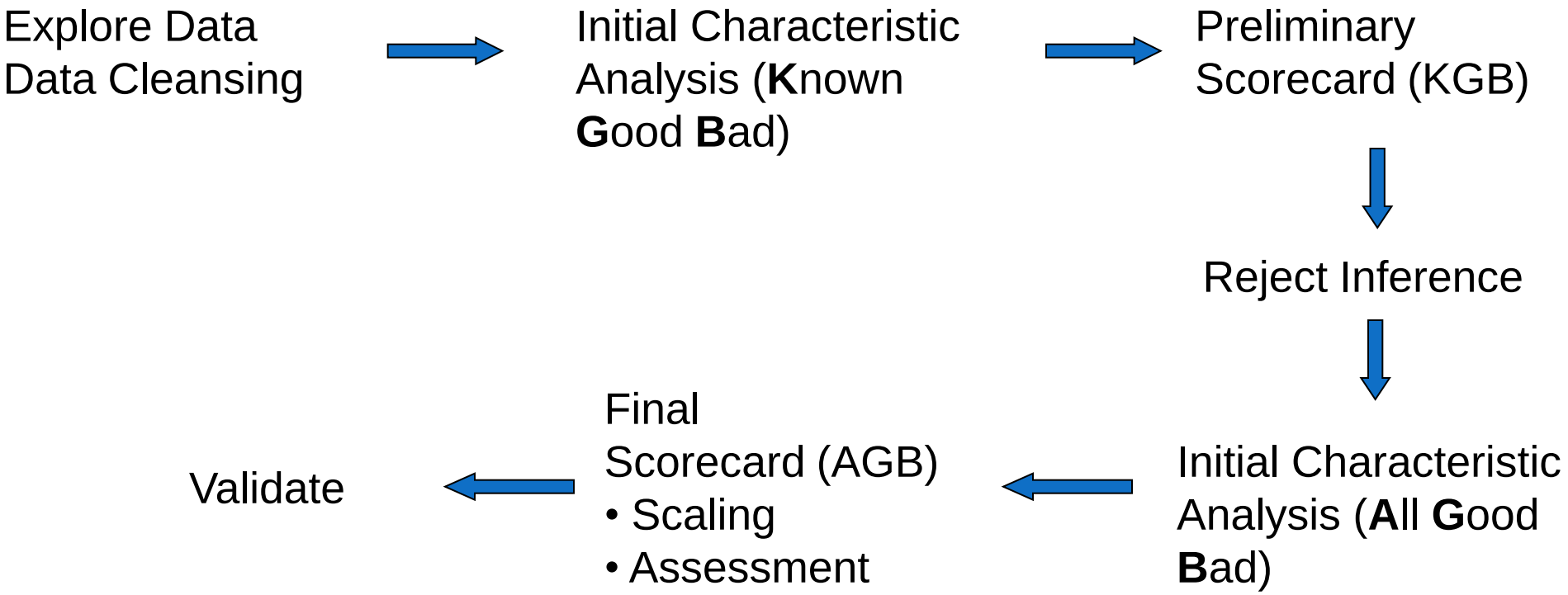
Development

Stage 4: Scorecard Development

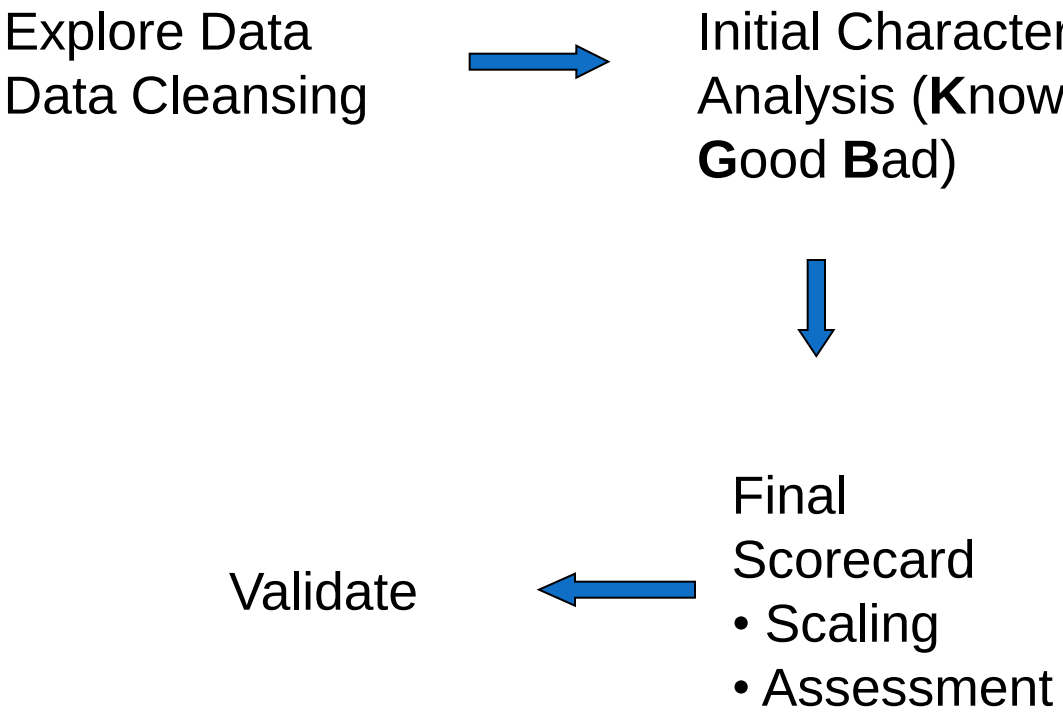
Objective

- Understand a methodology for developing and assessing risk scorecards.
  - Grouped attributes
  - Logistic regression
  - Reject inference
  - Scaled points.

Process Flow – Application Scorecard



Process Flow – Behavior Scorecard



Before you start …

- Explore the data, visualize (Insight in SAS EM)
- Distributions
  - mean, max/min, range, missing
- Compare with overall portfolio distributions
- Data integrity (any garbage, outliers)
- Ensure data meets the data specifications done earlier.
- Check that ‘0’s mean zero, not missing values.
- Population stability check:
  - Month by month table of distribution for each predictor (e.g. 200701 men 55%, women 45%, 200702 men 57%, women 43%)

Missing Values and Outliers

- Missing (ALL financial data has missing and garbage values)
  - Complete Case Analysis - Exclude everything with missing data .. In CS, you'll end up with nothing ☹.
  - Exclude characteristics or records with significant missing values
  - Group 'missing' as a distinct attribute -the weight of missing will tell you what missing contains. If it is close to neutral, good since it is random. Recommended – recognize that missing data has information value and may not be randomly missing. Find the value and use it. Plus, including missing 'points' in scorecard will take care of ppl who leave it blank.
  - Impute missing values – don't use mean/most likely, model based on decision tree may be better.
- Outliers (and mis-keys)
  - Exclude/replace records.

Missing Values

- Missing data is not usually random
- Missing data can be related to the target
  - New at job may leave yrs at empl blank
  - Low income or commercial customers leave income blank
- Do bad customers leave certain fields blank?
- Including and grouping missing data can answer this question.

Initial Characteristic Analysis

- Analyze individual characteristics
  - Identify strong characteristics
    - Best differentiators between ‘good’ and ‘bad’
    - Screening
- Select characteristics for regression (variable selection).

Initial Characteristic Analysis

- Start by performing initial grouping for each characteristic and rank order Information Value (PROC DMSPLIT or SPLIT, or EM node)
- Alternate: rank order characteristics by Chi Square or other method
- Fine tune grouping for stronger characteristics

- May want to perform other analysis prior to this (for example, use PC to identify collinear characteristics)
- Some people use principal components (PROC VARCLUS) to identify which characteristics they need from each cluster. And then concentrate on the best out of each.

Criteria for Variable Selection

- Predictive power of attribute:
  Weight of Evidence
- Range and trend of WOE across attributes
- Predictive power of characteristic:
  Information Value, Gini index(coefficient)
- Operational/business considerations.

Weight of Evidence

Age	Count	Distr Count	Goods	Distr Good	Bads	Distr Bad	Bad rate	Weight
Missing	50	3.00%	43	2.40%	8	4.10%	16%	-55.497
18-22	200	10.00%	152	8.40%	48	24.90%	24%	-108.405
23-26	300	15.00%	246	13.60%	54	28.00%	18%	-72.039
27-29	450	23.00%	405	22.40%	45	23.30%	10%	-3.951
30-35	500	25.00%	475	26.30%	25	13.00%	5%	70.771
35-44	350	18.00%	349	19.30%	11	5.70%	3%	122.044
44 +	150	8.00%	147	8.10%	3	1.60%	2%	165.509
Total	**2,000**		**1,807**		**193**		**9.65%**	
Information Value = 0.066								

$$\text{Ln}\left[\frac{\text{Distr Good}}{\text{Distr Bad}}\right] \times 100$$

Weight of Evidence

- Measures strength of each (grouped) attribute in separating goods and bads
- (Distr Good / Distr Bad) = odds of being good
- Negative weight: more bads than goods
- Logical trend
- For age 23-26:

WOE = ln (0.136 / 0.28) = -0.722 *(x 100 = -72.2)*

Information Value (Strength)

Age	Count	Distr Count	Goods	Distr Good	Bads	Distr Bad	Bad rate	Weight
Missing	50	3.00%	43	2.40%	8	4.10%	16%	-55.497
18-22	200	10.00%	152	8.40%	48	24.90%	24%	-108.405
23-26	300	15.00%	246	13.60%	54	28.00%	18%	-72.039
27-29	450	23.00%	405	22.40%	45	23.30%	10%	-3.951
30-35	500	25.00%	475	26.30%	25	13.00%	5%	70.771
35-44	350	18.00%	349	19.30%	11	5.70%	3%	122.044
44 +	150	8.00%	147	8.10%	3	1.60%	2%	165.509
Total	**2,000**		**1,807**		**193**		**9.65%**	
Information Value = 0.066								

$$\sum \left\{ \left[\text{Distr Good} - \text{Distr Bad} \right] \times \text{Weight} \right\}$$

Kullback, S., Information Theory and Statistics (1959)

Information Value

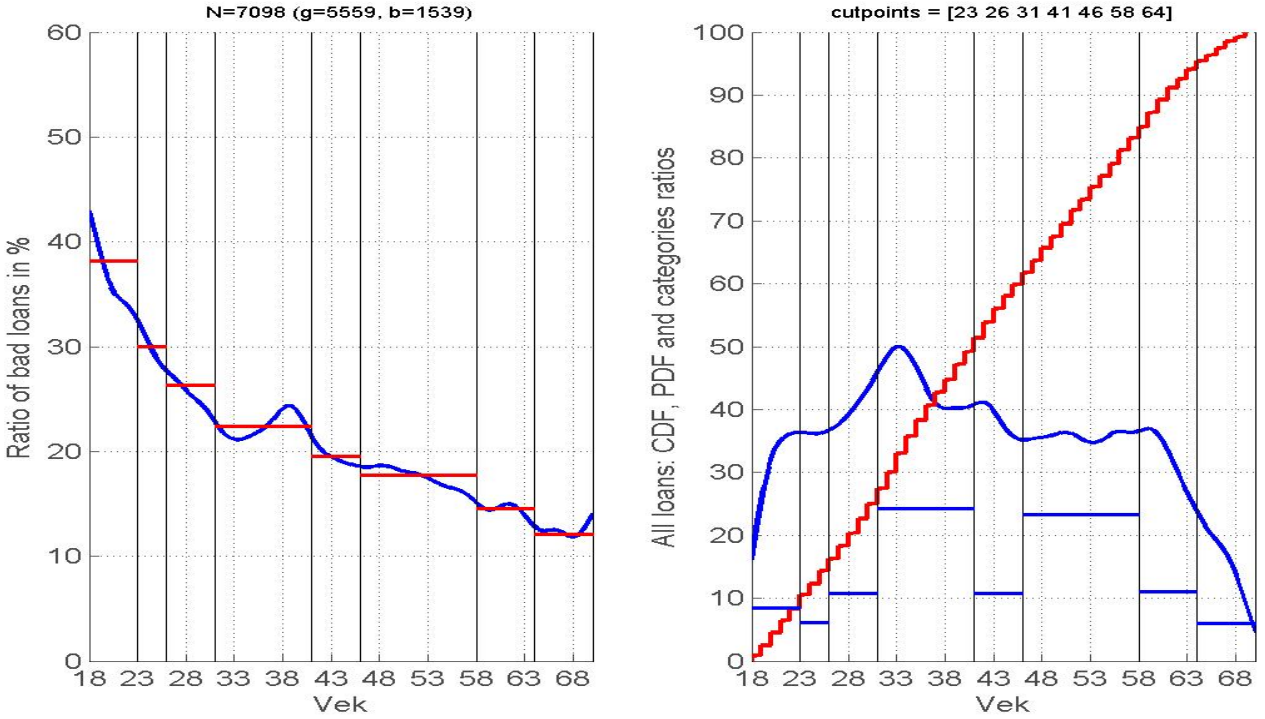
- Σ [(Distr Good - Distr Bad) x {ln (Distr Good / Distr Bad)}]
- When figures used in decimals format (for example, 0.136).
- Rule of thumb:
  - < 0.02: unpredictive
  - 0.02 – 0.1: weak
  - 0.1 – 0.3: medium
  - 0.3 +: strong
- Too strong? (IV>0.5) – use it in a controlled way (add them in the end of regression to see if they add any incremental value)

Grouping

- Groups with similar WOE are put together
- For continuous variables, groups are created so as to maximize difference from one group to next – and maintain logical trend for WOE

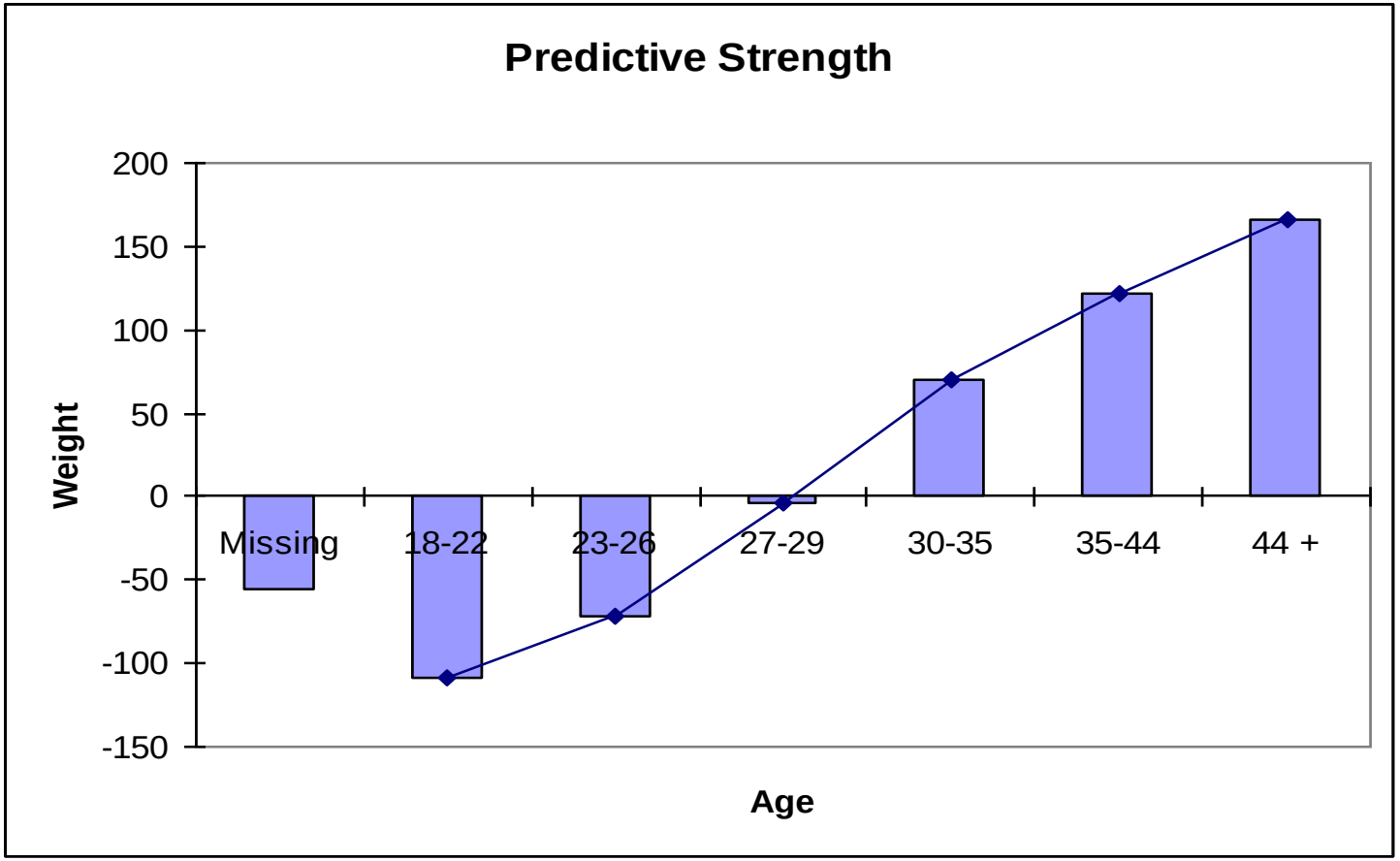
- Why Group?
  - Easier way to deal with outliers with interval variables, and for rare classes
  - Format of the scorecard
  - Easy to understand relationships
  - Model non-linear dependencies with linear models
  - Control the process

Grouping



Grouping of the demographic scorecard variable “age”. On the left pictures, the dependence of bad rate (smoothed using normal probability density function) on the variables is presented. On the right, the cumulative distribution function is presented. Vertical lines represent the borders between categories, horizontal red lines in the left picture represent the mean bad rate in categories, horizontal blue lines in the right picture represent the relative distribution of observations in the categories.

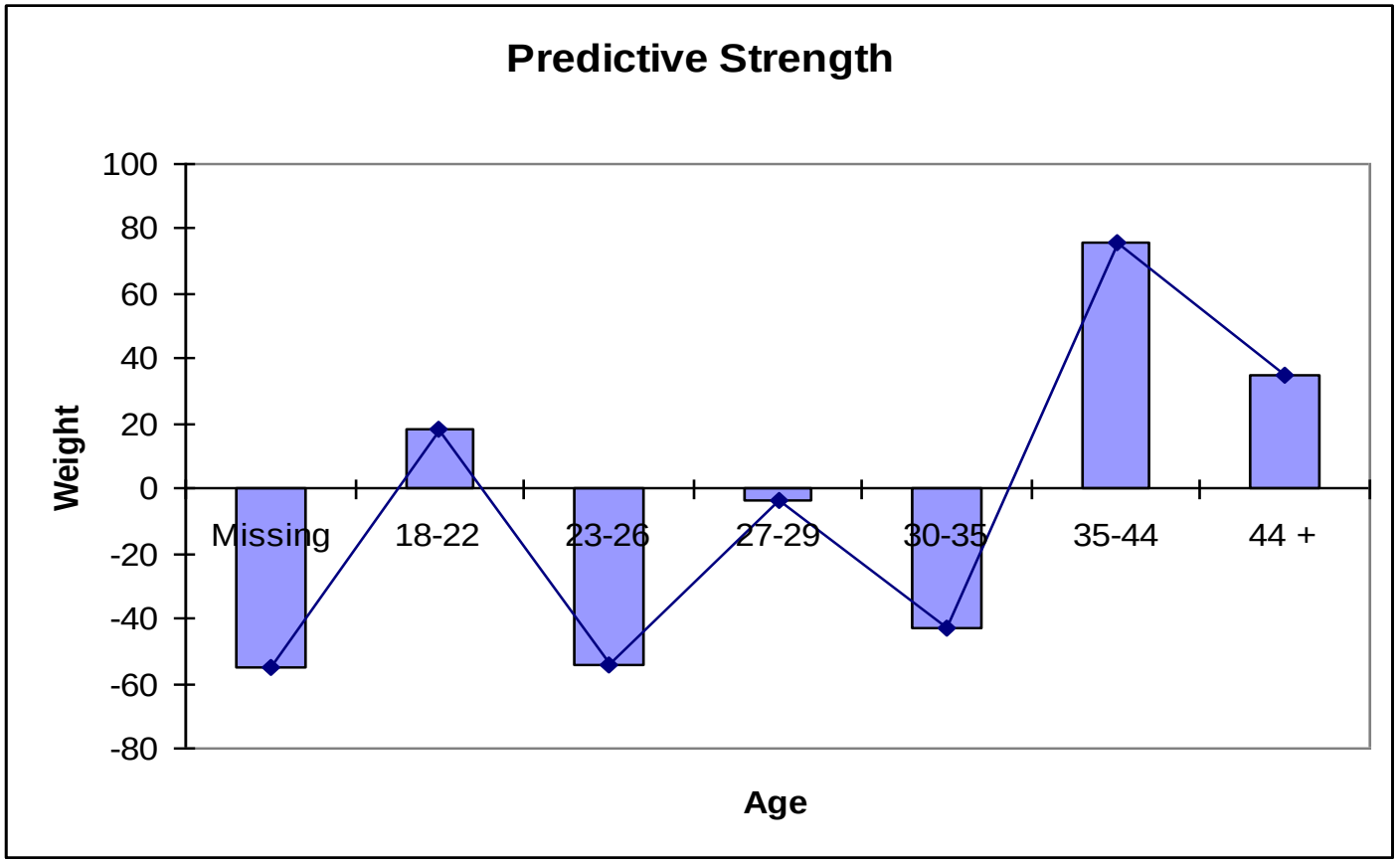
Logical Trend



Logical Trend

- Final weightings make sense.
- Enables buy-in from risk managers.
- Confirms business experience
  - young people are higher risk
  - higher debt service means higher risk
- Reduces overfitting if done right – model overall trend, not quirks. Remember how long the scorecard has to last. This is not going to be used for the next campaign and then discarded.
- Linear relationship not always true, but need trend to confirm, and back up with business experience. E.g. revolving open burden shows a 'banana curve' everywhere and is now accepted as that. People don't try to make it straight.

Logical Trend

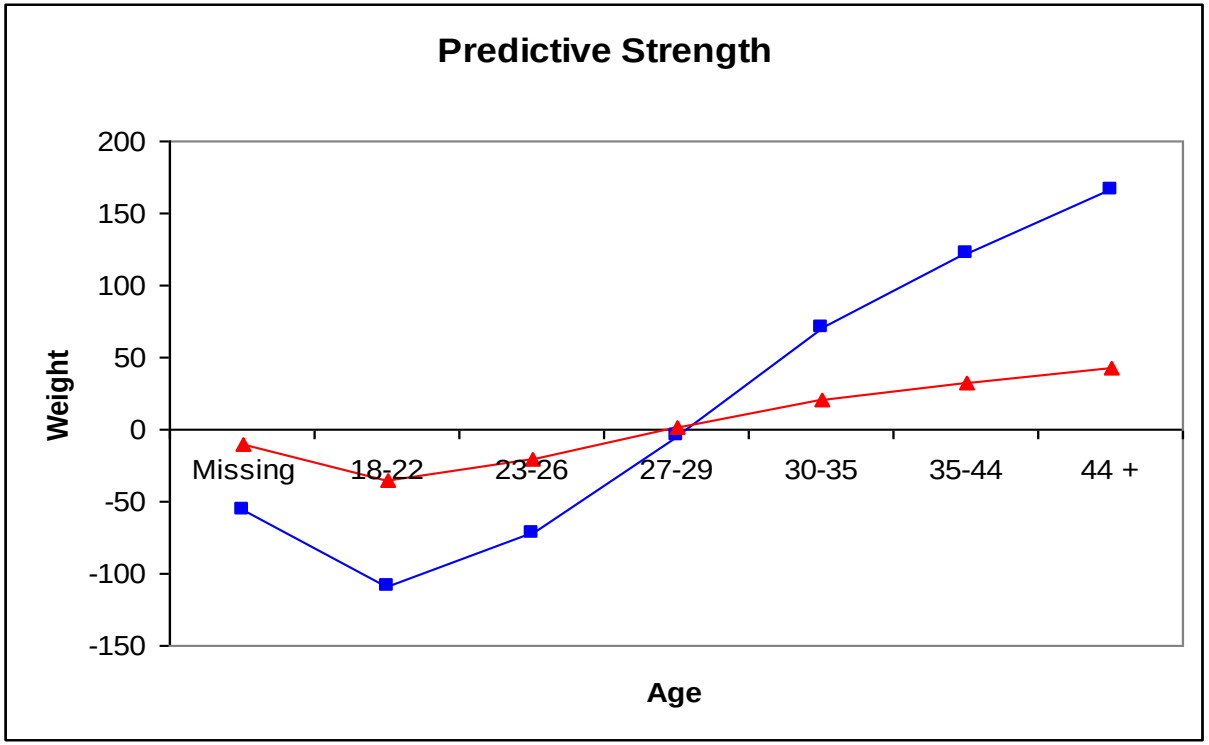


Obviously not a logical trend!!!

Logical Trend

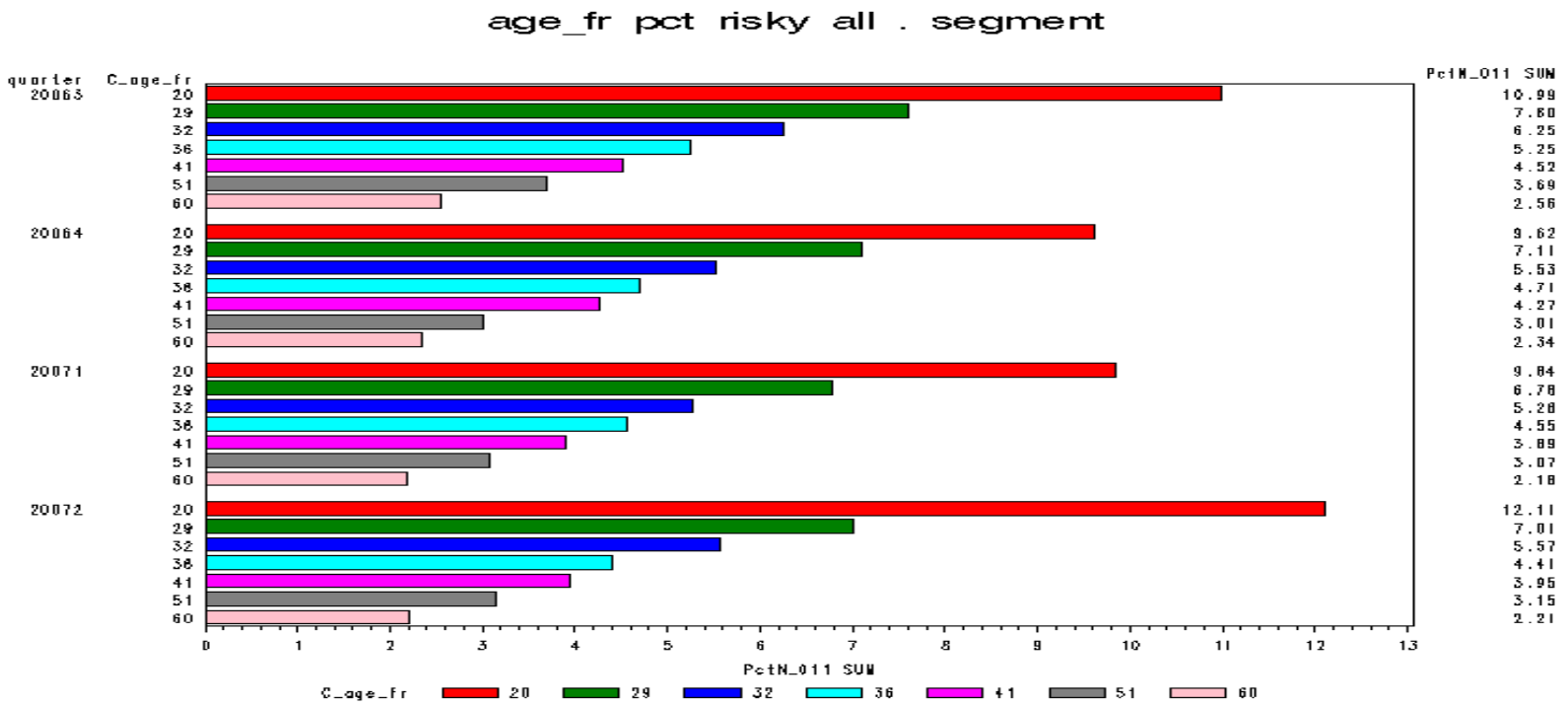
Which line shows logical trend?
Both are logical. What's the difference?

Blue line shows good differentiation.
Red line is flat, and this characteristic is likely very week and will be reflected in the *IV.*



Stability check

Check the stability of grouping throughout the whole developmnet time window:



Business Factors

- Nominal values
  - group based on similar weight (for example, postal code, occupation)
  - investigate splits on urban/rural, regional
- Breaks concurrent with policy rules
- Sanity check.

Variable Selection

List of information values of variables (predictors)

No	Character	IV Rank	Information Value
1	Max delinq L9M	1	0.176
2	Months since delinquent	2	0.176
3	Active contract (Y/N)	3	0.045
4	Average Delinquency L9M	4	0.087
5	**Months since >10 dpd**	5	0.144
6	Max delinq L3M	6	0.117
7	Average Delinquency L3M	7	0.108
8	Age of oldest contract	8	0.013
9	Number of months on collections as % total time on book	9	0.132
10	Months since >20 dpd	10	0.091
11	**Months since >30 dpd**	11	0.054
12	Num rejected applications L9M	12	0.033
13	Times 30+ dpd L9M	13	0.042
14	Total Payment L3M	14	0.018
15	Months since >40 dpd	15	0.030
16	Current balance as % of highest ever balance	16	0.048
17	Times 30+ dpd L3M	17	0.024
18	Payment Method	18	0.001

Cvičení –profile

```
/* 2b. Profiles */

%let input=income;
%let groups=yes;
%let n_groups=4;

/* grouping 1 - kvantily */

proc rank data=indata.accepts (keep=&input) groups=&n_groups
out=bins;
var &input;
ranks bin;
run;
proc summary data=bins nway missing;
class bin;
output out=bins (drop=_type_) min(&input)=start max(&input)=end;
run;
data bins;
set bins;
label=compress(put(start,best.))||' - '||compress(put(end,best.));
fmtname='__bin';
type='N';
run;
proc format cntlin=bins;
run;
```

```
%macro profile(input,groups);

/* Profile of &input according to BAD60 */

proc summary data=indata.accepts;
class &input;
output out=__bins (drop=_type_ rename=(_freq_=__n))
       sum(bad60)=__n1;
%if %upcase(&groups)=YES %then %do;
   format &input __bin.;
%end;
run;

data __bins;
set __bins end=__finish;
if _n_=1 then do;
   __all_n=__n;
   __all_n1=__n1;
   __all_n0=__n-__n1;
   retain __all_n:;
end;
else do;
   __p=__n/__all_n;
   __n0=__n-__n1;
   __p1=__n1/__all_n1;
   __p0=__n0/__all_n0;
   __r1=__n1/__n;
   __r0=__n0/__n;
   __woe=log((__p0)/(__p1))*100;
   __all_iv+(__p0-__p1)*__woe/100;
   output;
end;
if __finish then do;
   call symput('groups',compress(put(_n_-1,best.)));
   call symput('iv',compress(put(__all_iv,8.4)));
   call symput('br',compress(put(__all_n1/__all_n,best.)));
end;
attrib
   __n   label='N'
   __p   label='%'                    format=percent8.1
   __n1  label="N of Bad"
   __n0  label="N of Good"
   __p1  label="% of Bad"  format=percent8.1
   __p0  label="% of Good"  format=percent8.1
   __r1  label="Bad rate"  format=percent8.1
   __r0  label="Good rate"  format=percent8.1
   __woe label='WOE'             format=8.2
   &input   label="Group of &input"
;
drop __all:;
Run;

.
.
.
```

```
data __chart (keep=&input __sub __n __p __r);
set __bins (keep=&input __n0 __p0 __r0 __n1 __p1 __r1);
length __sub $4;
__sub="Good";
__n=__n0;
__p=__p0;
__r=__r0;
output;
__sub="Bad";
__n=__n1;
__p=__p1;
__r=__r1;
output;
attrib
   __n   label='N'      format=8.0
   __p   label='%'      format=percent8.1
   __r   label='Rate'   format=percent8.1
   __sub label='Target'
;
run;

proc datasets nolist;
delete gseg / memtype=catalog;
quit;

ods listing close;

goptions reset=all ftext='arial' htext=1.5 ftitle='arial' htitle=2;

proc gchart data=__chart;
axis1 style=0;
axis2 minor=none order=(0 to 1 by .25) label=none;
axis3 minor=none label=none;
axis4 minor=(n=4) label=none;
where __sub="Bad";
hbar &input / discrete sumvar=__r noframe nostats
            maxis=axis1 raxis=axis3 autoref cref=graya0 clipref
            name="__1";
title "Bad rates";
run;
where;
hbar &input / discrete subgroup=__sub sumvar=__n noframe nostats
            maxis=axis1 raxis=axis3 autoref cref=graya0 clipref
            name="__2";
title "Bad / Good frequencies";
run;
Quit;
```

```
proc gchart data=__bins;
hbar &input / discrete sumvar=__woe noframe nostats
            maxis=axis1 raxis=axis4 autoref cref=graya0 clipref
            name="__3";
title "Weight of evidence";
run;
hbar &input / discrete sumvar=__p1 noframe nostats
            maxis=axis1 raxis=axis4 autoref cref=graya0 clipref
            name="__4";
title "Bad distribution";
run;
quit;


ods html path="&appl_root" file="5.profile.html" style=statdoc;

proc report data=__bins nofs style(summary)=[htmlclass="Header"];
columns ("Attributes of &input" &input) ('Total' __n __p)
        ("Good" __n0 __p0) ("Bad" __n1 __p1) ('Measures' __r1 __woe);
define &input / group;
compute after;
__r1.sum=&br;
__woe.sum=.;
endcomp;
rbreak after / summarize;
title "Bad / Good by &input";
footnote "IV=&iv (<0.02 unpredictive, <0.1 week, <0.3 medium, <0.5 strong, >0.5 over)";
run;

goptions device=gif;
proc greplay nofs;
footnote;
igout gseg;
tc sashelp.templt;
template l2r2;
treplay 1:__1 2:__2 3:__3 4:__4 name="5_profil";
run;
quit;
title;
footnote;

ods html close;

ods listing;

%mend profile;

%profile(&input,&groups)
```

Bad / Good by income

Attributes of income	Total		Good		Bad		Measures	
Group of income	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
15000.067206 - 17541.45177	16147	25.0%	15631	25.0%	516	26.1%	3.2%	-4.55
17541.61083 - 19631.429437	16147	25.0%	15688	25.1%	459	23.2%	2.8%	7.52
19631.471069 - 21723.106242	16148	25.0%	15683	25.0%	465	23.5%	2.9%	6.19
21723.273059 - 35790.940583	16147	25.0%	15612	24.9%	535	27.1%	3.3%	-8.29
	64589	100.0%	62614	100.0%	1975	100.0%	3.1%	.

IV=0.0046 (<0.02 unpredictive, <0.1 week, <0.3 medium, <0.5 strong, >0.5 over)

Cvičení

```
/*profile multiple characteristics at once*/
%model_profilevar
(
    data=data.accepts,
    interval=age income idratio ,
    binary=sex phone client,
    ordinal=age_grp income_grp region,
    groups=5,
    target=bad30,
    rep_out=&appl_root
)
```

Bad / Good by Sex

Attributes of sex		Total		Good		Bad		Measures	
Group of sex	Sex	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
M	M	45138	69.9%	42061	69.6%	3077	74.7%	6.8%	-7.16
Z	Z	19451	30.1%	18410	30.4%	1041	25.3%	5.4%	18.59
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Unpredictive (IV = 0.0133, 2 groups)

Bad / Good by Phone member?

Attributes of phone		Total		Good		Bad		Measures	
Group of phone	Phone member?	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
0	0	8081	12.5%	7431	12.3%	650	15.8%	8.0%	-25.04
1	1	56508	87.5%	53040	87.7%	3468	84.2%	6.1%	4.07
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Unpredictive (IV = 0.0102, 2 groups)

Bad / Good by Existing client?

Attributes of client		Total		Good		Bad		Measures	
Group of client	Existing client?	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
0	0	60188	93.2%	56251	93.0%	3937	95.6%	6.5%	-2.74
1	1	4401	6.8%	4220	7.0%	181	4.4%	4.1%	46.23
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Unpredictive (IV = 0.0126, 2 groups)

Cvičení

Bad / Good by Age groups

Attributes of age_grp		Total		Good		Bad		Measures	
Group of age_grp	Age groups	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
do 30	do 30	2957	4.6%	2662	4.4%	295	7.2%	10.0%	-48.69
30 - 60	30 - 60	58713	90.9%	55057	91.0%	3656	88.8%	6.2%	2.52
nad 60	nad 60	2919	4.5%	2752	4.6%	167	4.1%	5.7%	11.53
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Unpredictive (IV = 0.0146, 3 groups)

Bad / Good by Income groups

Attributes of income_grp		Total		Good		Bad		Measures	
Group of income_grp	Income groups	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
do 17	do 17	12070	18.7%	11213	18.5%	857	20.8%	7.1%	-11.54
17 - 22	17 - 22	37859	58.6%	35567	58.8%	2292	55.7%	6.1%	5.52
22 - 27	22 - 27	13680	21.2%	12820	21.2%	860	20.9%	6.3%	1.50
nad 27	nad 27	980	1.5%	871	1.4%	109	2.6%	11.1%	-60.85
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Unpredictive (IV = 0.0118, 4 groups)

Cvičení

Bad / Good by Region

Attributes of region		Total		Good		Bad		Measures	
Group of region	Region	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
1	1	12537	19.4%	11404	18.9%	1133	27.5%	9.0%	-37.77
2	2	16335	25.3%	15498	25.6%	837	20.3%	5.1%	23.18
3	3	10679	16.5%	10034	16.6%	645	15.7%	6.0%	5.77
4	4	10797	16.7%	10170	16.8%	627	15.2%	5.8%	9.95
5	5	7199	11.1%	6783	11.2%	416	10.1%	5.8%	10.47
6	6	7042	10.9%	6582	10.9%	460	11.2%	6.5%	-2.59
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Weak predictivity (IV = 0.0483, 6 groups)

Cvičení

Bad / Good by Age

Attributes of age		Total		Good		Bad		Measures	
Group of age	Age	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
0	18 - 35	13688	21.2%	12420	20.5%	1268	30.8%	9.3%	-40.49
1	36 - 40	11385	17.6%	10485	17.3%	900	21.9%	7.9%	-23.15
2	41 - 45	14645	22.7%	13918	23.0%	727	17.7%	5.0%	26.52
3	46 - 51	12383	19.2%	11806	19.5%	577	14.0%	4.7%	33.17
4	52 - 74	12488	19.3%	11842	19.6%	646	15.7%	5.2%	22.18
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	.

Weak predictivity (IV = 0.0931, 5 groups)

Cvičení

Bad / Good by Income

Attributes of income		Total		Good		Bad		Measures	
Group of income	Income	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
0	15.000 - 17.105	12917	20.0%	12011	19.9%	906	22.0%	7.0%	-10.23
1	17.105 - 18.822	12918	20.0%	12204	20.2%	714	17.3%	5.5%	15.18
2	18.822 - 20.398	12918	20.0%	12122	20.0%	796	19.3%	6.2%	3.64
3	20.398 - 22.339	12918	20.0%	12102	20.0%	816	19.8%	6.3%	0.99
4	22.340 - 35.791	12918	20.0%	12032	19.9%	886	21.5%	6.9%	-7.82
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

Unpredictive (IV = 0.0080, 5 groups)

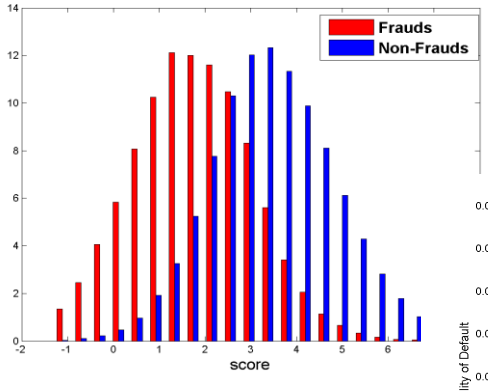
Cvičení

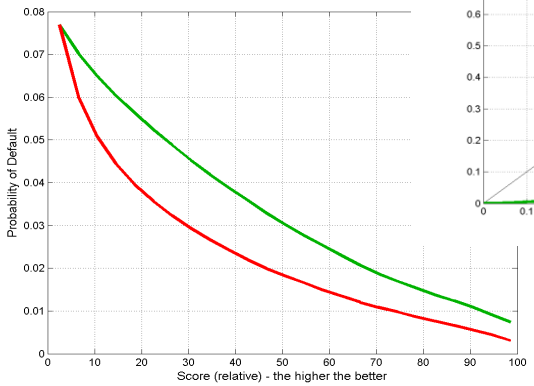
Bad / Good by Income/Debt ratio

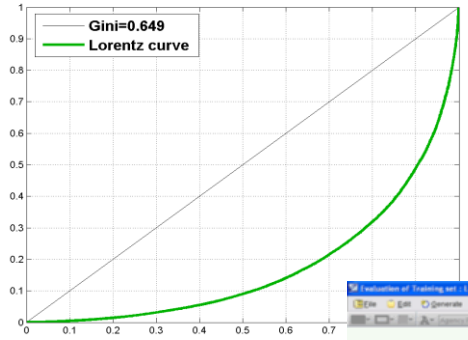
Attributes of idratio		Total		Good		Bad		Measures	
Group of idratio	Income Debt ratio	N	%	N of Good	% of Good	N of Bad	% of Bad	Bad rate	WOE
0	0.0175 - 0.0225	12917	20.0%	11994	19.8%	923	22.4%	7.1%	-12.23
1	0.0225 - 0.0293	12918	20.0%	11998	19.8%	920	22.3%	7.1%	-11.87
2	0.0293 - 0.0421	12918	20.0%	12061	19.9%	857	20.8%	6.6%	-4.25
3	0.0421 - 0.0714	12918	20.0%	12216	20.2%	702	17.0%	5.4%	16.98
4	0.0714 - 0.2995	12918	20.0%	12202	20.2%	716	17.4%	5.5%	14.89
		64589	**100.0%**	**60471**	**100.0%**	**4118**	**100.0%**	**6.4%**	**.**

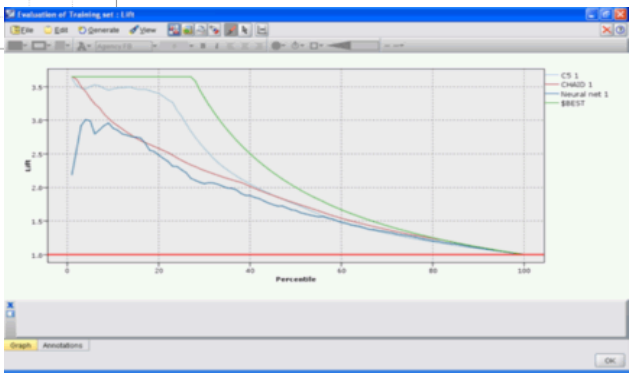
Unpredictive (IV = 0.0160, 5 groups)

9. Evaluace modelu – LC(ROC), Gini, KS, Lift









Úvod

- Je nemožné využívat predikční modely efektivně bez znalosti jejich kvality/diskriminační síly.

- Většinou je k dispozici celá řada modelů a je třeba vybrat jen jeden – ten nejlepší.

Měření kvality modelu

- Uvažujeme dva základní skupiny indexů kvality. První je založena na distribuční funkci. Mezi nejpoužívanější indexy patří

  - Kolmogorovova-Smirnovova statistika (KS)
  - Giniho index
  - C-statistika
  - Lift.

- Druhá skupina indexů je založena na pravděpodobnostní hustotě. Mezi nejznámější indexy patří

  - Střední diference (Mahalanobisova vzdálenost)
  - Informační statistika/hodnota (I_{Val}).

Indexy založené na distribuční funkci - KS

$$D_K = \begin{cases} 1, & klient \;\; je\, dobrý \\ 0, & jinak. \end{cases}$$

Počet dobrých klientů: n

Počet špatných klientů: m

Proporce dobrých/špatných klientů: $p_G = \frac{n}{n+m}$, $p_B = \frac{m}{n+m}$

- Empirické distribuční funkce:

$$F_{n.GOOD}(a) = \frac{1}{n}\sum_{i=1}^{n} I(s_i \le a \wedge D_K = 1)$$

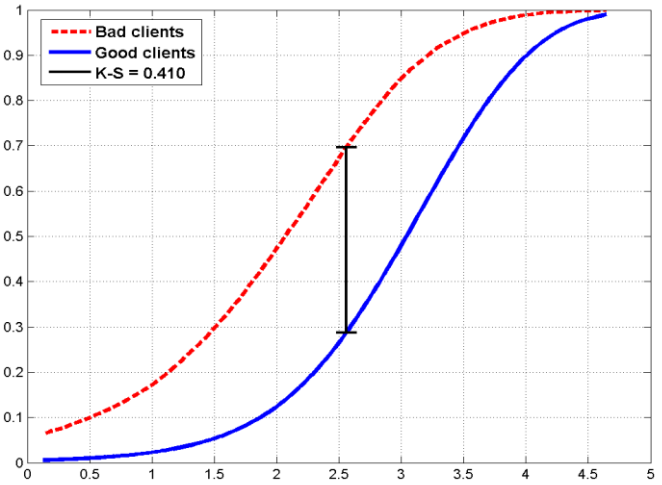
$$F_{m.BAD}(a) = \frac{1}{m}\sum_{i=1}^{m} I(s_i \le a \wedge D_K = 0)$$

$$F_{N.ALL}(a) = \frac{1}{N}\sum_{i=1}^{N} I(s_i \le a) \qquad a \in [L, H]$$

$$I(A) = \begin{cases} 1 & A \;\; platí \\ 0 & jinak \end{cases}$$

- Kolmogorovova-Smirnovova statistika (KS)

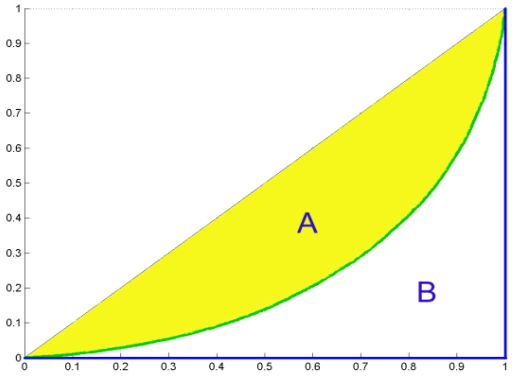
$$KS = \max_{a \in [L,H]} \left| F_{m,BAD}(a) - F_{n,GOOD}(a) \right|$$



Lorenzova křivka

- Lorenzova křivka (LC)

$$x = F_{m.BAD}(a)$$
$$y = F_{n.GOOD}(a),\ a \in [L, H].$$



- Giniho index

$$Gini = \frac{A}{A+B} = 2A$$

$$Gini = 1 - \sum_{k=2}^{n+m} \left(F_{m.BAD\,k} - F_{m.BAD\,k-1}\right) \cdot \left(F_{n.GOOD\,k} + F_{n.GOOD\,k-1}\right)$$

kde $F_{m.BAD\,k}$ ($F_{n.GOOD_k}$) je k-tá hodnota vektoru empirické distribuční funkce špatných (dobrých) klientů

Somersovo D, Kendalovo τ_α

- Giniho index je speciální případ Somersova *D* (Somers (1962)), které je pořadovou asociační mírou definovanou jako

$$D_{YX} = \frac{\tau_{XY}}{\tau_{XX}}$$

kde τ_{XY} je Kendallovo τ_a definované jako $\tau_{XY} = E[sign(X_1 - X_2)sign(Y_1 - Y_2)]$

kde (X_1, Y_1) (X_2, Y_2) jsou bivariantní, stochasticky nezávislé, náhodné vektory nad touž datovou populací, a $E[\cdot]$ značí střední hodnotu. V našem případě je *Y=1* jestliže je klient dobrý a *Y=0* jestliže je klient špatný. Proměnná *X* reprezentuje skóre.

Thomas (2009) uvádí, že Somersovo *D* hodnotící výkonnost daného credit scoringového modelu lze vypočítat pomocí

$$D_S = \frac{\sum_i g_i \sum_{j<i} b_j - \sum_i g_i \sum_{j>i} b_j}{n \cdot m}$$

kde g_i (b_j) je počet dobrých (špatných) klientů v i-tém intervalu skóre.

Somersovo D, Mann-Whitney U

- Dále platí, že D_S může být vyjádřeno pomocí Mann-Whitneyho U-statistiky.
  - Seřaď datový vzorek ve vzestupném pořadí podle skóre a sečti pořadí dobrých klientů ve vzniklé posloupnosti. Označme tento součet jako R_G. Potom

$$U = R_G - \frac{1}{2} n(n+1)$$

$$D_S = 2\frac{U}{n \cdot m} - 1$$

Konkordantní, diskordantní páry

- Konkordantní pár (X_1,Y_1), (X_2,Y_2):

$$\operatorname{sgn}(X_2 - X_1) = \operatorname{sgn}(Y_2 - Y_1)$$

- Diskordantní pár:

$$\operatorname{sgn}(X_2 - X_1) = -\operatorname{sgn}(Y_2 - Y_1)$$

- V našem případě X představuje skóre a Y ukazatel dobrého klienta (D_K). Protože dobrý klient má hodnotu Y=1 a špatný Y=0, je zřejmé, že u konkordantního páru má dobrý klient vyšší hodnotu skóre než klient špatný.

Somersovo D, Goodman-Kruskal gamma

- Uvažujme tedy dva náhodně vybrané klienty, přičemž jeden je dobrý (Y_1=1) a druhý špatný (Y_2=0), skóre prvního označme s_1, druhého s_2. Pak
  - Konkordantní pár (Concordant): $s_1>s_2$
  - Diskordantní pár (Discordant): $s_1<s_2$
  - Vázaný pár (Tied): $s_1=s_2$

- Somersovo D:

$$D_S = \frac{\#Concordant - \#Discodrant}{\#Concordant + \#Discodrant + \#Tied}$$

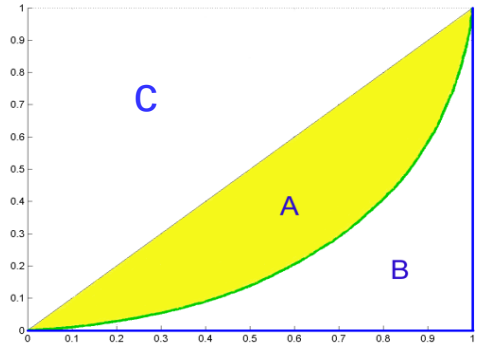
- Goodmanovo-Kruskalovo Gamma:

$$\gamma = \frac{\#Concordant - \#Discodrant}{\#Concordant + \#Discodrant}$$

C-statistika

- C-statistika:

$$c-stat = A + C = \frac{1 + Gini}{2}$$



Tato statistika je rovna pravděpodobnosti, že náhodně vybraný dobrý klient má vyšší skóre než náhodně vybraný špatný klient, tj.

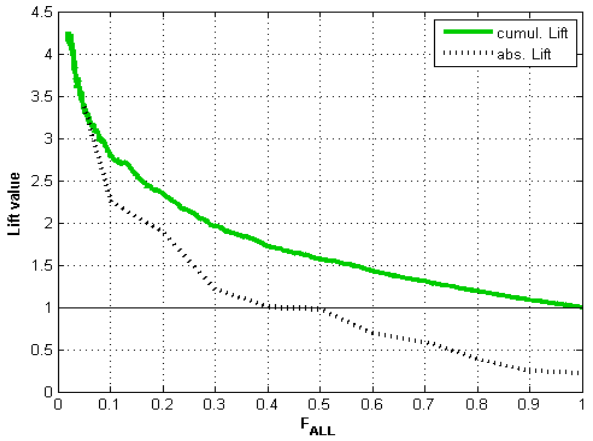
$$c-stat = P\left(s_1 \geq s_2 \mid D_{K_1} = 1 \wedge D_{K_2} = 0\right)$$

Lift

- Další možnou mírou kvality scoringového modelu je Lift, který říká kolikrát je daný model, při dané úrovni zamítání, lepší než náhodný model. Přesněji řečeno jde o poměr proporce špatných klientů se skóre menším nebo rovno dané hodnotě skóre a, $a \in [L, H]$, ku proporci špatných klientů v celé populaci. Formálně jej lze zapsat takto:

$$Lift(a) = \frac{CumBadRate(a)}{BadRate} = \frac{\frac{\sum_{i=1}^{n+m} I(s_i \le a \wedge Y = 0)}{\sum_{i=1}^{n+m} I(s_i \le a)}}{\frac{\sum_{i=1}^{n+m} I(Y = 0)}{\sum_{i=1}^{n+m} I(Y = 0 \vee Y = 1)}} = \frac{\frac{\sum_{i=1}^{n+m} I(s_i \le a \wedge Y = 0)}{\sum_{i=1}^{n+m} I(s_i \le a)}}{\frac{n}{N}}$$

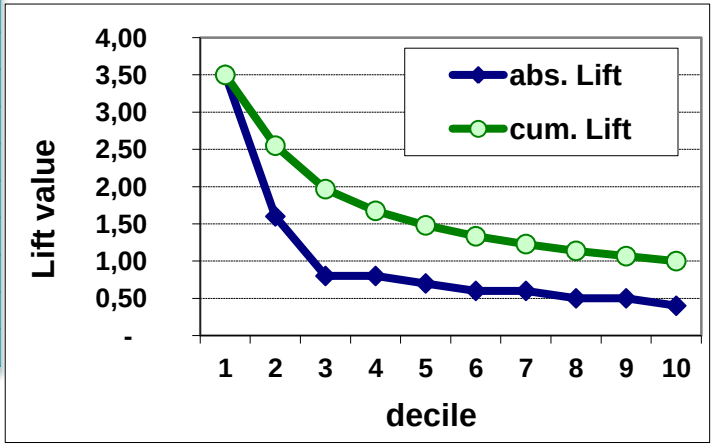
$$absLift(a) = \frac{BadRate(a)}{BadRate}$$



Lift

- Usually it is computed using table with numbers of all and bad clients in some score bands (deciles).

decile	# cleints	absolutely			cumulatively		
		# bad clients	Bad rate	abs. Lift	# bad clients	Bad rate	cum. Lift
1	100	35	35.0%	3.50	35	35.0%	3.50
2	100	16	16.0%	1.60	51	25.5%	2.55
3	100	8	8.0%	0.80	59	19.7%	1.97
4	100	8	8.0%	0.80	67	16.8%	1.68
5	100	7	7.0%	0.70	74	14.8%	1.48
6	100	6	6.0%	0.60	80	13.3%	1.33
7	100	6	6.0%	0.60	86	12.3%	1.23
8	100	5	5.0%	0.50	91	11.4%	1.14
9	100	5	5.0%	0.50	96	10.7%	1.07
10	100	4	4.0%	0.40	100	10.0%	1.00
All	1000	100	10.0%				

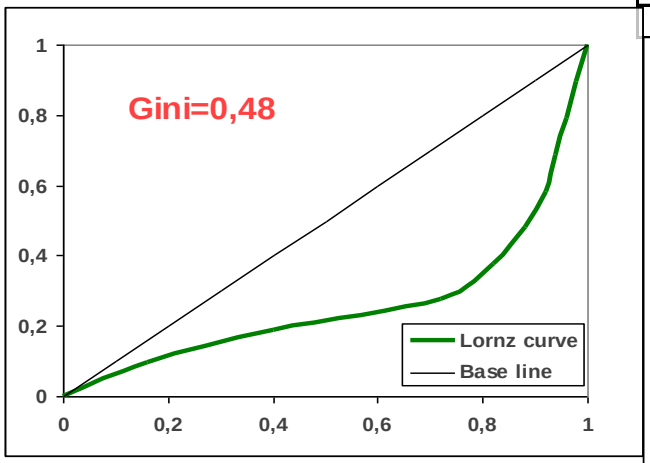


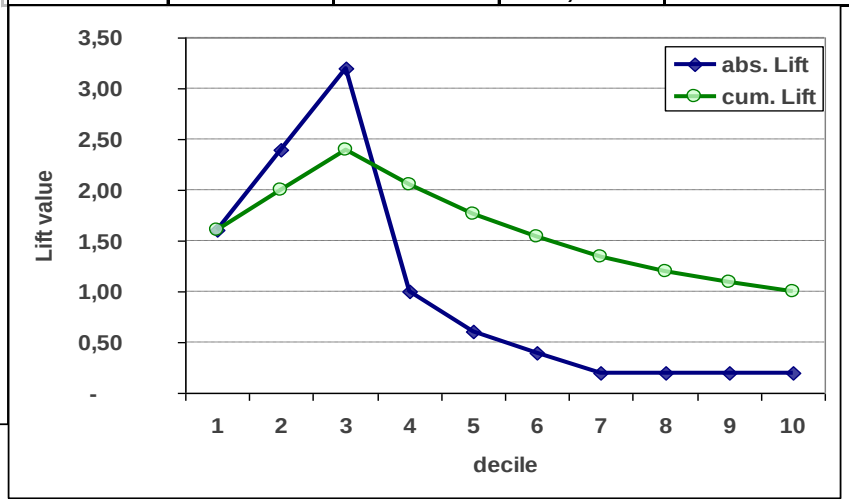
- It takes positive values. Cumulative form ends in value 1.
- Upper limit of Lift depends on p_B .

Lift

- Pokud bad rate není monotonní:
  - LC vypadá OK
  - Gini se mírně sníží
  - Lift ovšem vypadá podivně

decile	# cleints	absolutely			cumulatively		
		# bad clients	Bad rate	abs. Lift	# bad clients	Bad rate	cum. Lift
1	100	8	8,0%	1,60	8	8,0%	1,60
2	100	12	12,0%	2,40	20	10,0%	2,00
3	100	16	16,0%	3,20	36	12,0%	2,40
4	100	5	5,0%	1,00	41	10,3%	2,05
5	100	3	3,0%	0,60	44	8,8%	1,76
6	100	2	2,0%	0,40	46	7,7%	1,53
7	100	1	1,0%	0,20	47	6,7%	1,34
8	100	1	1,0%	0,20	48	6,0%	1,20
9	100	1	1,0%	0,20	49	5,4%	1,09
10	100	1	1,0%	0,20	50	5,0%	1,00
All	1000	50	5,0%				



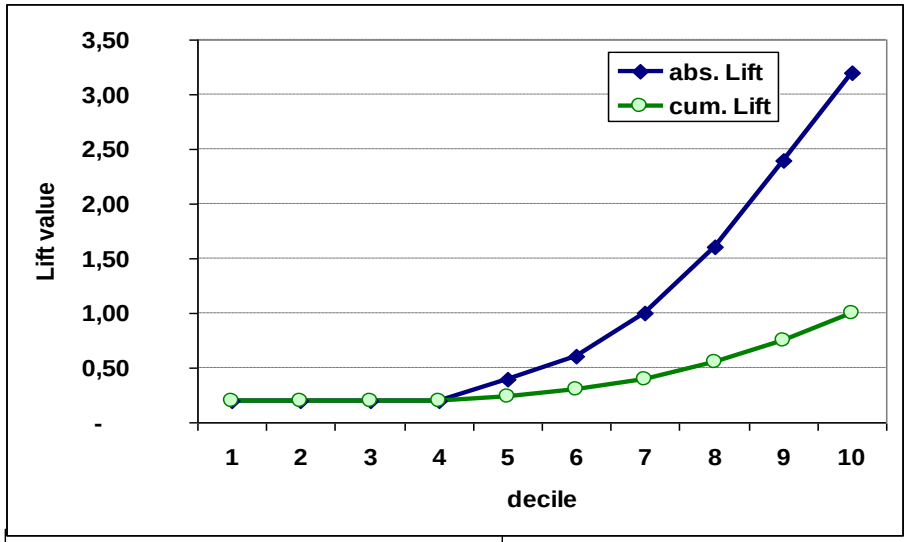


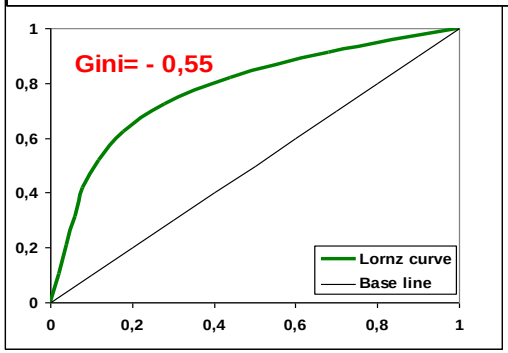
Lift

- Pokud má skóre zcela opačný smysl, obdržíme „opačné“ obrázky.

decile	# cleints	absolutely			cumulatively		
		# bad clients	Bad rate	abs. Lift	# bad clients	Bad rate	cum. Lift
1	100	16	16,0%	3,20	16	16,0%	3,20
2	100	12	12,0%	2,40	28	14,0%	2,80
3	100	8	8,0%	1,60	36	12,0%	2,40
4	100	5	5,0%	1,00	41	10,3%	2,05
5	100	3	3,0%	0,60	44	8,8%	1,76
6	100	2	2,0%	0,40	46	7,7%	1,53
7	100	1	1,0%	0,20	47	6,7%	1,34
8	100	1	1,0%	0,20	48	6,0%	1,20
9	100	1	1,0%	0,20	49	5,4%	1,09
10	100	1	1,0%	0,20	50	5,0%	1,00
All	1000	50	5,0%				

decile	# cleints	absolutely			cumulatively		
		# bad clients	Bad rate	abs. Lift	# bad clients	Bad rate	cum. Lift
1	100	1	1,0%	0,20	1	1,0%	0,20
2	100	1	1,0%	0,20	2	1,0%	0,20
3	100	1	1,0%	0,20	3	1,0%	0,20
4	100	1	1,0%	0,20	4	1,0%	0,20
5	100	2	2,0%	0,40	6	1,2%	0,24
6	100	3	3,0%	0,60	9	1,5%	0,30
7	100	5	5,0%	1,00	14	2,0%	0,40
8	100	8	8,0%	1,60	22	2,8%	0,55
9	100	12	12,0%	2,40	34	3,8%	0,76
10	100	16	16,0%	3,20	50	5,0%	1,00
All	1000	50	5,0%				



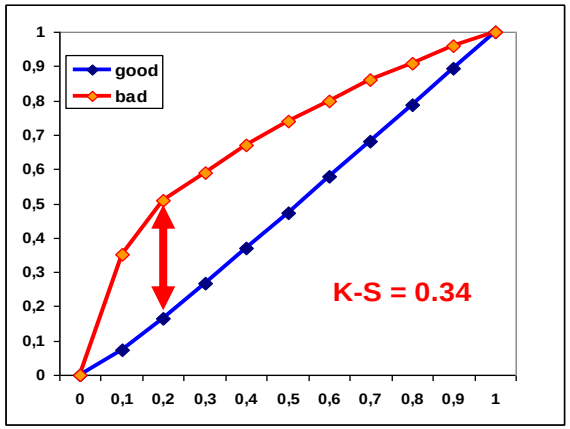


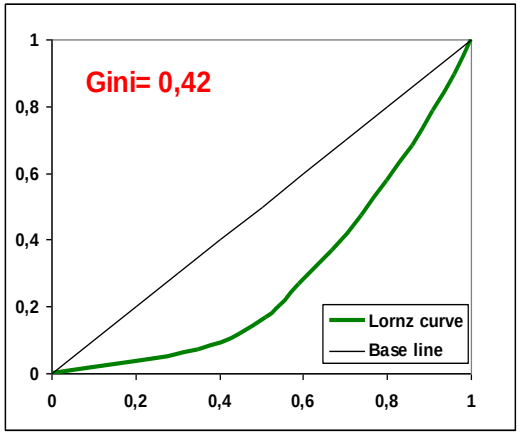
Lift vs. Gini a KS

- **Je evidentní, že pouze Gini nestačí!!!**

SC 1:

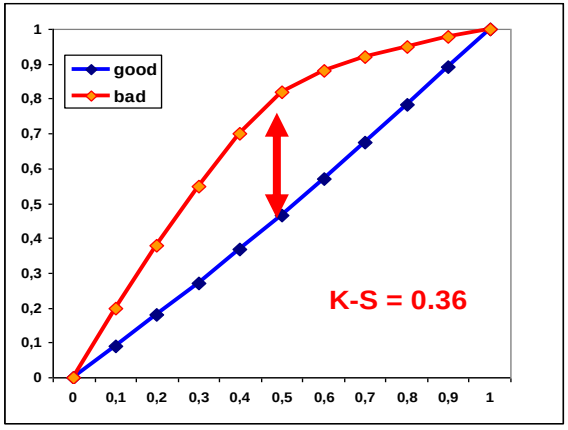
decile	# cleints	# bad clients	Bad rate
1	100	35	35,0%
2	100	16	16,0%
3	100	8	8,0%
4	100	8	8,0%
5	100	7	7,0%
6	100	6	6,0%
7	100	6	6,0%
8	100	5	5,0%
9	100	5	5,0%
10	100	4	4,0%
All	1000	100	10,0%

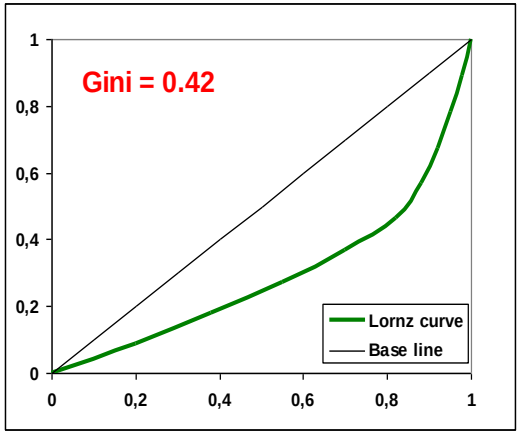




SC 2:

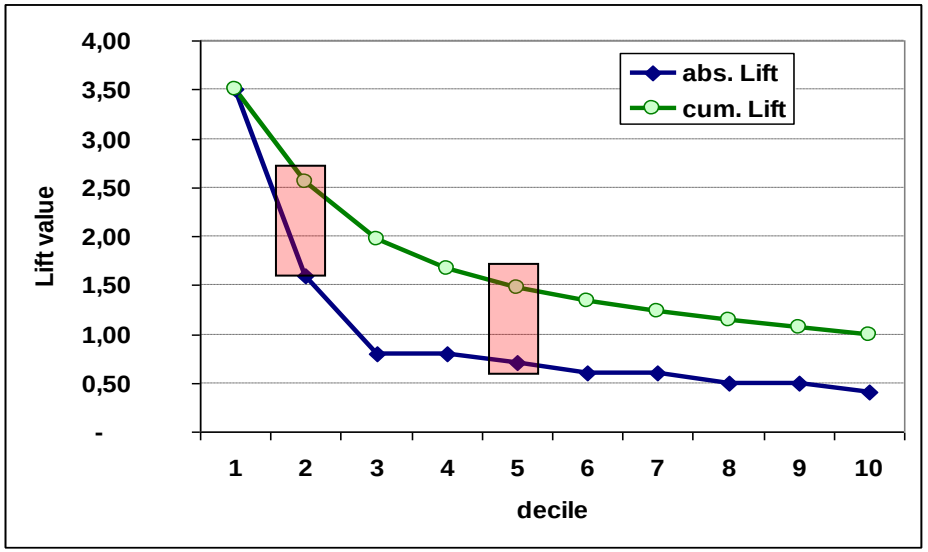
decile	# cleints	# bad clients	Bad rate
1	100	20	20,0%
2	100	18	18,0%
3	100	17	17,0%
4	100	15	15,0%
5	100	12	12,0%
6	100	6	6,0%
7	100	4	4,0%
8	100	3	3,0%
9	100	3	3,0%
10	100	2	2,0%
All	1000	100	10,0%



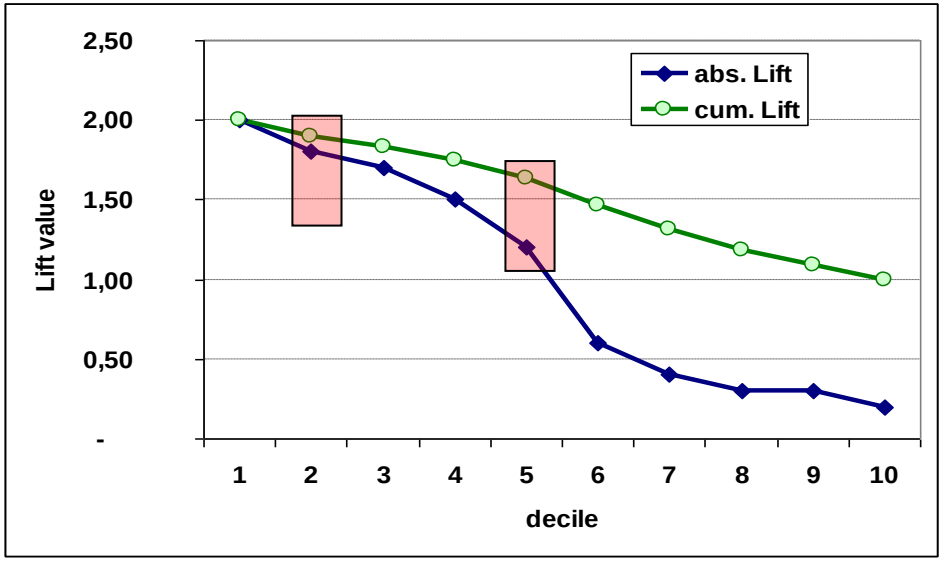


Lift vs. Gini a KS

- **SC 1:**



- **SC 2:**



$Lift_{20\%} = 2.55 \quad > \quad Lift_{20\%} = 1.90$

$Lift_{50\%} = 1.48 \quad < \quad Lift_{50\%} = 1.64$

SC 2 je lepší, pokud je předpokládaná míra zamítaní (reject rate) přibližně 50%.
SC 1 je významně lepší, pokud je předpokládaný reject rate přibližně 20%.

Lift, QLift

- Lift can be expressed and computed by formula:

$$Lift(a) = \frac{F_{m.BAD}(a)}{F_{N.ALL}(a)}, \quad a \in [L, H]$$

- In practice, Lift is computed corresponding to 10%, 20%, . . . , 100% of clients with the worst score. Hence we define:

$$QLift(q) = \frac{F_{m.BAD}(F^{-1}_{N.ALL}(q))}{F_{N.ALL}(F^{-1}_{N.ALL}(q))} = \frac{1}{q} F_{m.BAD}(F^{-1}_{N.ALL}(q)), \quad q \in (0,1]$$

$$F^{-1}_{N.ALL}(q) = \min\{a \in [L, H], F_{N.ALL}(a) \geq q\}$$

- Typical value of q is 0.1. Then we have

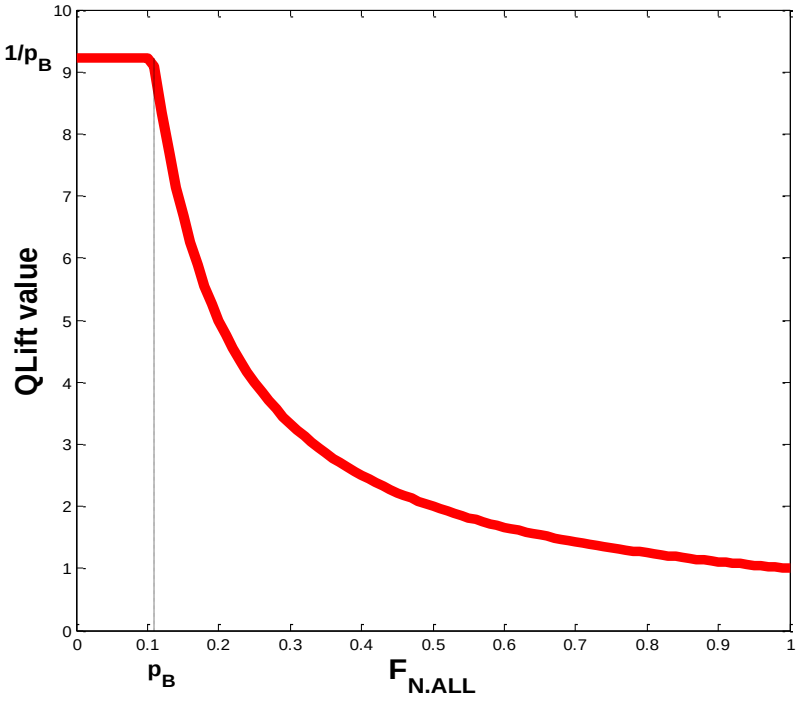
$$QLift_{10\%} = QLift(0.1) = 10 \cdot F_{m.BAD}(F^{-1}_{N.ALL}(0.1))$$

Lift and QLift for ideal model

- It is natural to ask how look Lift and QLift in case of ideal model. Hence we derived following formulas.

- Lift for ideal model:

$$Lift_{ideal}(a) = \begin{cases} \frac{1}{p_B}, & a \leq c \\ \frac{1}{F_{N.ALL}(a)}, & a > c \end{cases}$$



- QLift for ideal model:

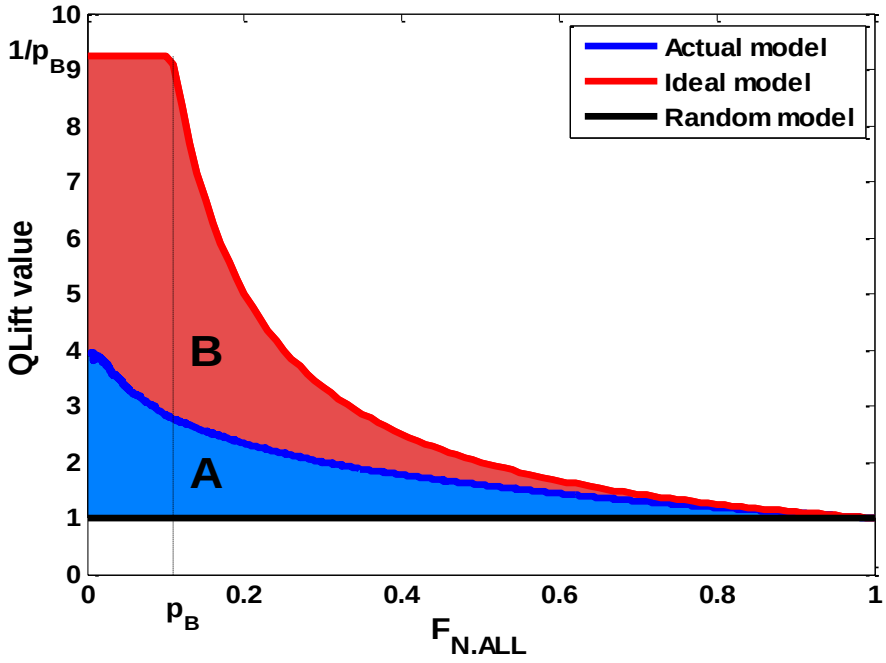
$$QLift_{ideal}(q) = \begin{cases} \frac{1}{p_B}, & q \in (0, p_B] \\ \frac{1}{q}, & q \in (p_B, 1] \end{cases}$$

We can see that the upper limit of Lift and QLift is equal to $1/p_B$.

Lift Ratio (LR)

- Once we know form of QLift for ideal model, we can define Lift Ratio as analogy to Gini index.

$$LR = \frac{A}{A+B} = \frac{\int_0^1 QLift(q)\,dq - 1}{\int_0^1 QLift_{ideal}(q)\,dq - 1}$$

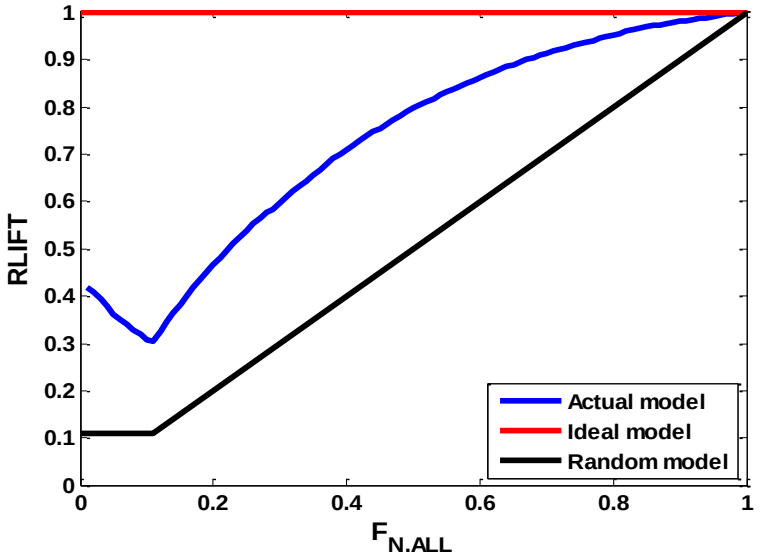


- It is obvious that it is global measure of model's quality and that it takes values from 0 to 1. Value 0 corresponds to random model, value 1 match to ideal model. Meaning of this index is quite simple. The higher, the better. Important feature is that Lift Ratio allows us to fairly compare two models developed on different data samples, which is not possible with Lift.

Rlift, IRL

- Since Lift Ratio compares areas under Lift function for actual and ideal models, next concept is focused on comparison of Lift functions themselves. We define Relative Lift function by

$$RLift(q) = \frac{QLift(q)}{QLift_{ideal}(q)}, \quad q \in (0, 1]$$



- In connection to RLift we define Integrated Relative Lift (IRL):

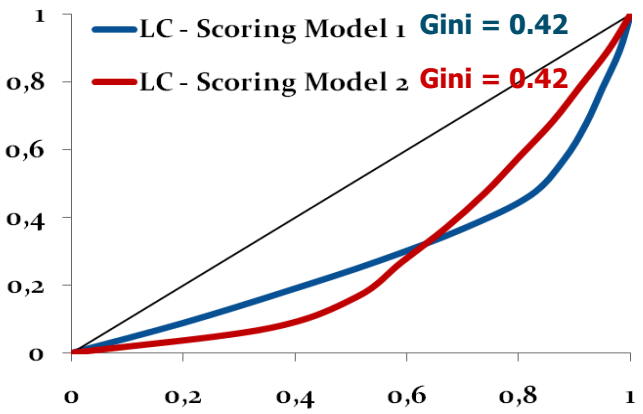
$$IRL = \int_0^1 RLift(q)\, dq$$

- It takes values from $0.5 + \frac{p_B^2}{2}$, for random model, to 1, for ideal model. Following simulation study shows interesting connection to c-statistics.

Příklad

- We consider two scoring models with score distribution given in the table below.
- We consider standard meaning of scores, i.e. higher score band means better clients (the highest probability of default have clients with the lowest scores, i.e. clients in score band 1).
- Gini indexes are equal for both models.
- From the Lorenz curves is evident, that the first model is stronger for higher score bands and the second one is better for lower score bands.
- The same we can read from values of QLift.

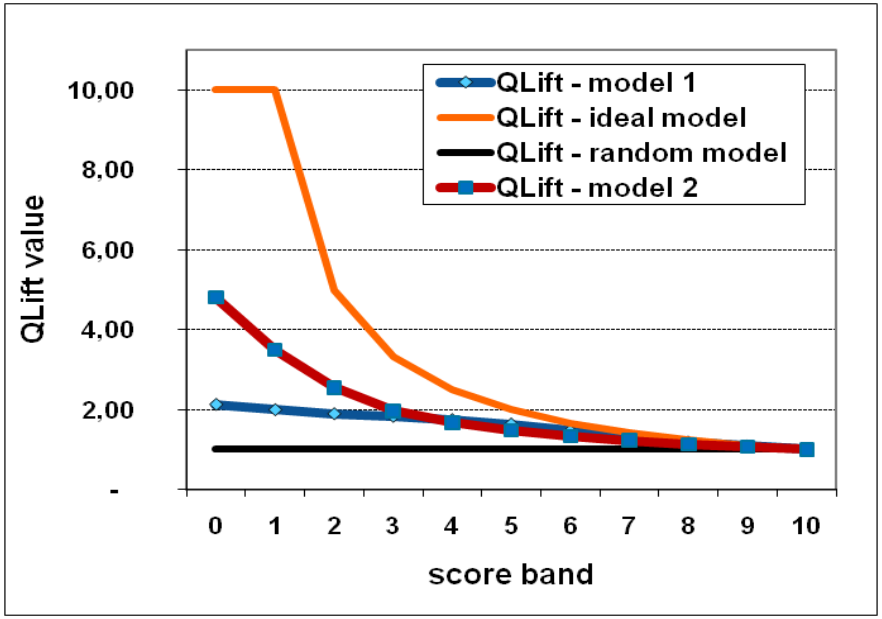
			Scoring Model 1				Scoring Model 2			
score band	# clients	q	# bad clients	# cumul. bad clients	# cumul. bad rate	QLift	# bad clients	# cumul. bad clients	# cumul. bad rate	QLift
1	100	0.1	20	20	20.0%	**2.00**	35	35	35.0%	**3.50**
2	100	0.2	18	38	19.0%	**1.90**	16	51	25.5%	**2.55**
3	100	0.3	17	55	18.3%	**1.83**	8	59	19.7%	**1.97**
4	100	0.4	15	70	17.5%	**1.75**	8	67	16.8%	**1.68**
5	100	0.5	12	82	16.4%	**1.64**	7	74	14.8%	**1.48**
6	100	0.6	6	88	14.7%	**1.47**	6	80	13.3%	**1.33**
7	100	0.7	4	92	13.1%	**1.31**	6	86	12.3%	**1.23**
8	100	0.8	3	95	11.9%	**1.19**	5	91	11.4%	**1.14**
9	100	0.9	3	98	10.9%	**1.09**	5	96	10.7%	**1.07**
10	100	1.0	2	100	10.0%	**1.00**	4	100	10.0%	**1.00**
All	1000		100				100			

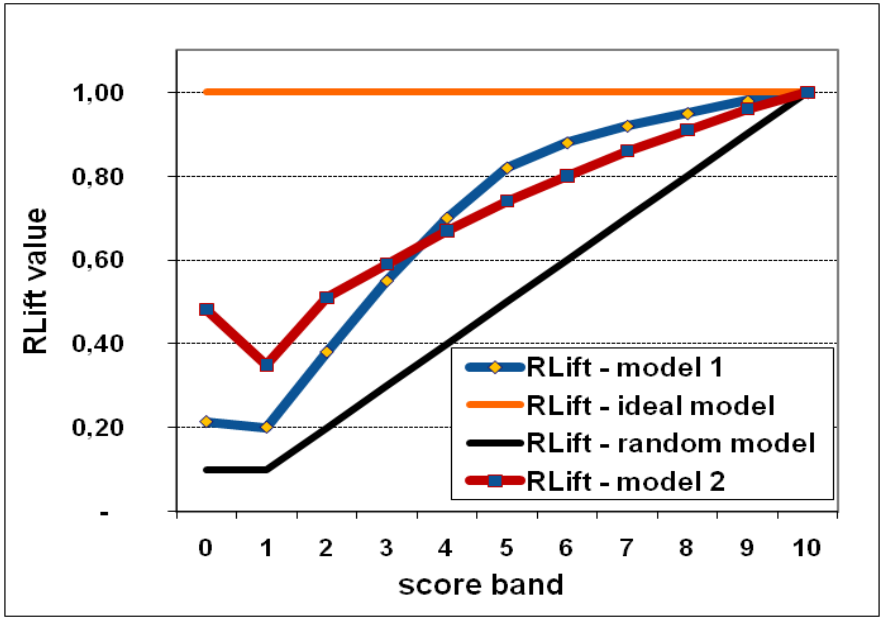


Příklad

- Since Qlift is not defined for *q=0*, we extrapolated the value by

$$QLift(0) = 3 \cdot QLift(0.1) - 3 \cdot QLift(0.2) + QLift(0.3)$$



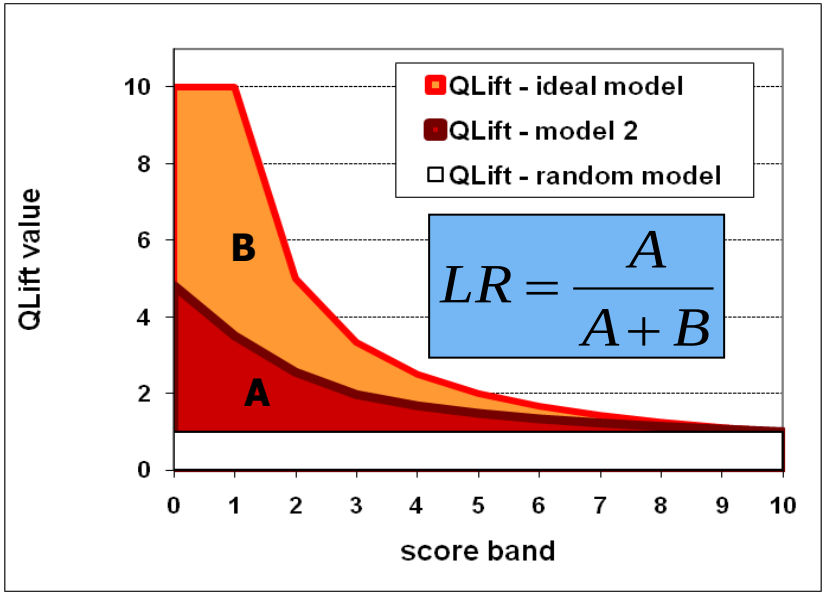


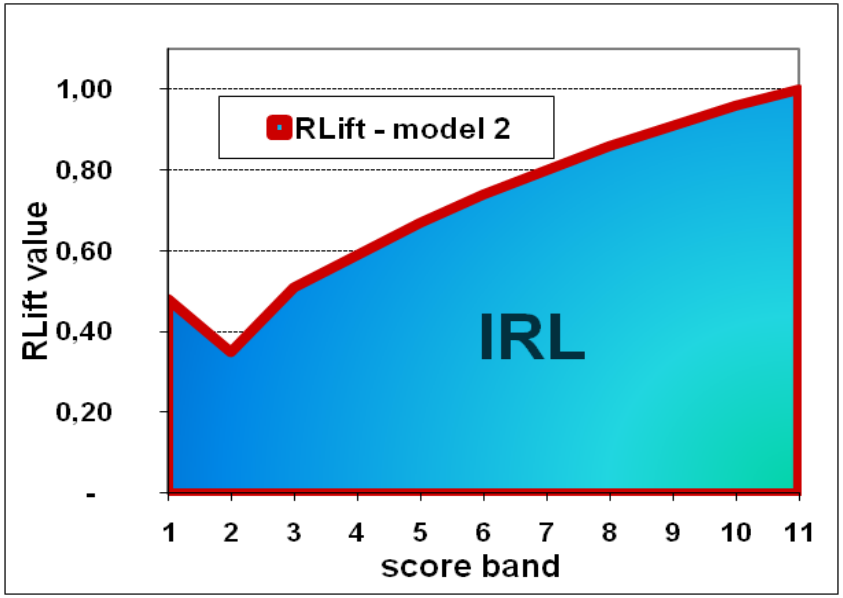
According to both Qlift and Rlift curves we can state that:

- If expected reject rate is up to 40%, then model 2 is better.
- If expected reject rate is more than 40%, then model 1 is better.

Příklad

- Now, we consider indexes LR and IRL:





	scoring model 1	scoring model 2
GINI	**0.420**	**0.420**
QLift(0.1)	**2.000**	**3.500**
LR	**0.242**	**0.372**
IRL	**0.699**	**0.713**

Using LR and IRL we can state that model 2 is better than model 1 although their Gini coefficients are equal.

Střední diference

- Střední diference (Mahalanobis distance):

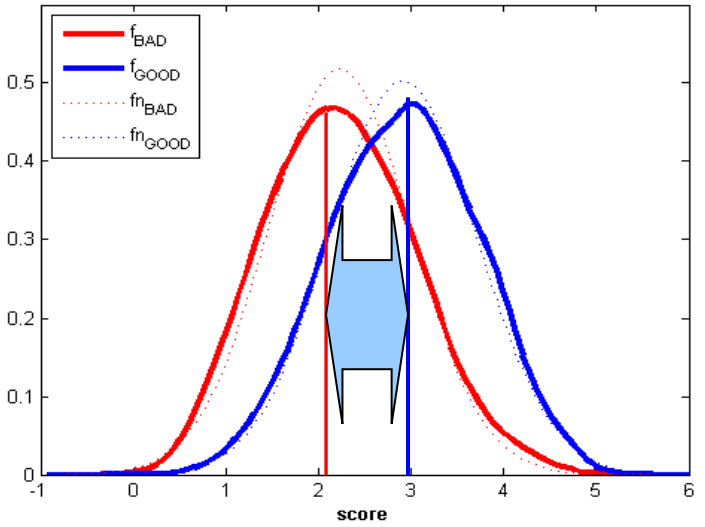
$$D = \frac{M_g - M_b}{S}$$

kde S je společná směrodatná odchylka:

$$S = \left(\frac{nS_g^{\ 2} + mS_b^{\ 2}}{n+m} \right)^{\frac{1}{2}}$$

M_g , M_b jsou střední hodnoty dobrých (špatných) klientů

S_g , S_b jsou příslušné směrodatné odchylky.



Normálně rozložené skóre

- Předpokládejme, že skóre dobrých a špatných klientů je normálně rozloženo, tj. jejich pravděpodobnostní hustoty mají tvar

$$f_{GOOD}(x) = \frac{1}{\sigma_g \sqrt{2\pi}} e^{-\frac{(x-\mu_g)^2}{2\sigma_g^2}} \qquad f_{BAD}(x) = \frac{1}{\sigma_b \sqrt{2\pi}} e^{-\frac{(x-\mu_b)^2}{2\sigma_b^2}}$$

- Odhady parametrů μ_b, μ_b, σ_g a σ_b :

  M_g , M_b jsou aritmetické průměty skóre dobrých (špatných) klientů

  S_g , S_b jsou směrodatné odchylky skóre dobrých (špatných) klientů

- Společná směrodatná odchylka:

$$S = \left(\frac{nS_g^{\,2} + mS_b^{\,2}}{n+m} \right)^{\frac{1}{2}}$$

- Odhady střední hodnoty a směrodatné odchylky skóre všech klientů μ_{ALL} , σ_{ALL}:

$$M = M_{ALL} = \frac{nM_g + mM_b}{n+m} \qquad S_{ALL} = \left(\frac{nS_g^{\,2} + mS_b^{\,2} + n(M_g - M)^2 + m(M_b - M)^2}{(n+m)} \right)^{\frac{1}{2}}$$

Normálně rozložené skóre

- Předpokládejme, že směrodatné odchylky obou skóre jsou rovny hodnotě σ, pak:

$$D = \frac{\mu_g - \mu_b}{\sigma}$$

$$D = \frac{M_g - M_b}{S}$$

$$KS = \Phi\left(\frac{D}{2}\right) - \Phi\left(\frac{-D}{2}\right) = 2 \cdot \Phi\left(\frac{D}{2}\right) - 1$$

$$Gini = 2 \cdot \Phi\left(\frac{D}{\sqrt{2}}\right) - 1$$

$$Lift_q = \frac{1}{q}\Phi\left(\frac{\sigma_{ALL}}{\sigma} \cdot \Phi^{-1}(q) + p_G \cdot D\right)$$

$$Lift_q = \frac{1}{q}\Phi\left(\frac{S_{ALL}}{S}\Phi^{-1}(q) + p_G \cdot D\right)$$

Kde $\Phi(\cdot)$ je distribuční funkce standardizovaného normálního rozložení, $\Phi_{\mu,\sigma^2}(\cdot)$ je distribuční funkce s parametry μ , σ^2 a $\Phi^{-1}(\cdot)$ je standardizovaná kvantilová funkce.

Normálně rozložené skóre

- Obecně, tj. bez předpokladu rovnosti směrodatných odchylek skóre:

$$D^* = \frac{\mu_g - \mu_b}{\sqrt{\sigma_g^2 + \sigma_b^2}}$$

$$D^* = \frac{M_g - M_b}{\sqrt{S_g^2 + S_b^2}}$$

$$KS = \Phi\left(\frac{a}{b}\sigma_b \cdot D^* - \frac{1}{b}\sigma_g\sqrt{a^2 D^{*2} + 2b \cdot c}\right) - \Phi\left(\frac{a}{b}\sigma_g \cdot D^* - \frac{1}{b}\sigma_b\sqrt{a^2 D^{*2} + 2b \cdot c}\right)$$

kde $a = \sqrt{\sigma_b^2 + \sigma_g^2}, \quad b = \sigma_b^2 - \sigma_g^2, \quad c = \ln\left(\frac{\sigma_g}{\sigma_b}\right)$

$$KS = \Phi\left(\frac{\sqrt{S_b^2 + S_g^2}}{S_b^2 - S_g^2} S_b \cdot D^* - \frac{1}{S_b^2 - S_g^2} S_g \sqrt{\left(S_b^2 + S_g^2\right) D^{*2} + 2 \cdot \left(S_b^2 - S_g^2\right) \ln\left(\frac{S_g}{S_b}\right)}\right)$$

$$-\Phi\left(\frac{\sqrt{S_b^2 + S_g^2}}{S_b^2 - S_g^2} S_g \cdot D^* - \frac{1}{S_b^2 - S_g^2} S_b \sqrt{\left(S_b^2 + S_g^2\right) D^{*2} + 2 \cdot \left(S_b^2 - S_g^2\right) \ln\left(\frac{S_g}{S_b}\right)}\right)$$

Normálně rozložené skóre

- Obecně, tj. bez předpokladu rovnosti směrodatných odchylek skóre:

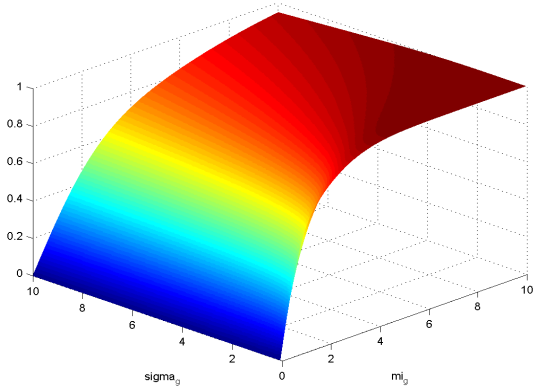
$$Gini = 2 \cdot \Phi\left(D^*\right) - 1$$

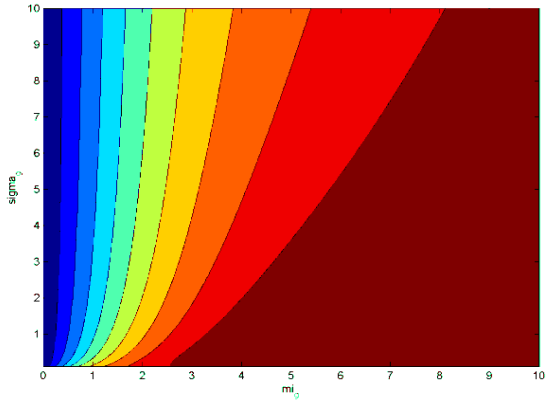
$$Lift_q = \frac{1}{q} \Phi_{\mu_b, \sigma_b^2}\left(\mu_{ALL} + \sigma_{ALL} \cdot \Phi^{-1}(q)\right) = \frac{1}{q} \Phi\left(\frac{\sigma_{ALL} \cdot \Phi^{-1}(q) + \mu_{ALL} - \mu_b}{\sigma_b}\right)$$

$$Lift_q = \frac{1}{q} \Phi\left(\frac{S_{ALL} \cdot \Phi^{-1}(q) + M - M_b}{S_b}\right)$$

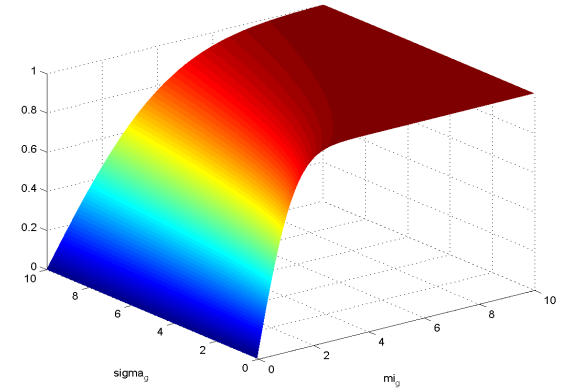
Normálně rozložené skóre

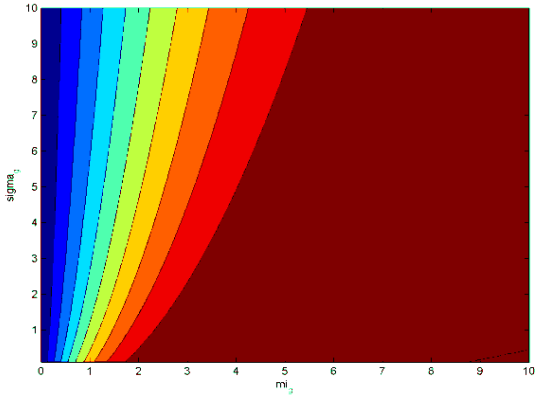
- **KS:** $\mu_b = 0, \sigma_b^2 = 1$





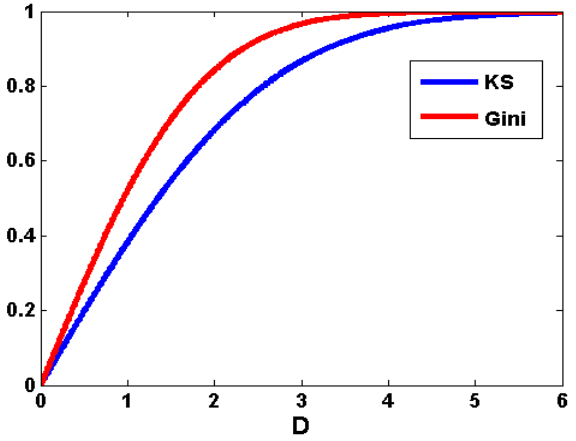
- **Gini** $\mu_b = 0, \sigma_b^2 = 1$





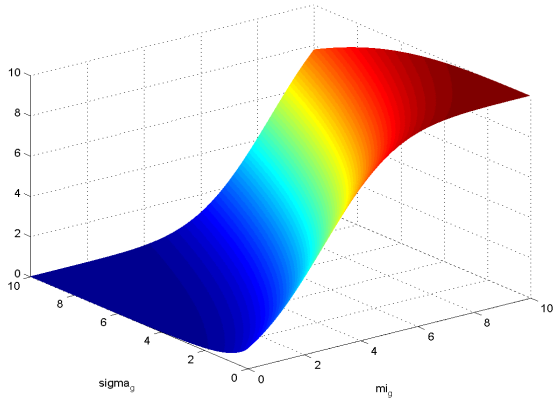
- KS i Gini reagují velmi silně na změnu μ_g, ale zůstávají téměř beze změny ve směru σ_g^2.

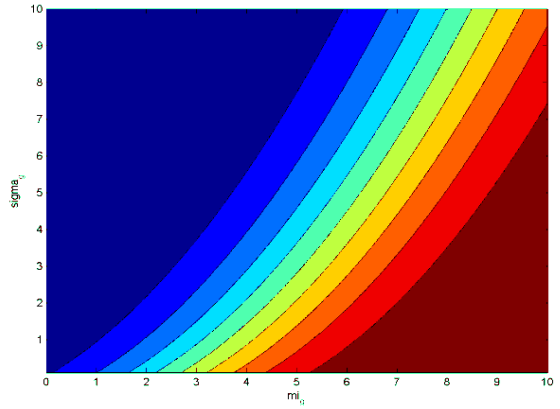
- **Gini > KS**



Normálně rozložené skóre

- **Lift$_{10\%}$:** $\mu_b = 0$, $\sigma_b^2 = 1$





- V případě indexu Lift$_{10\%}$ je evidentní silná závislost na μ_g a významně vyšší závislost na σ_g^2 než v případě KS a Gini.

ROC (Receiver operating characteristic)

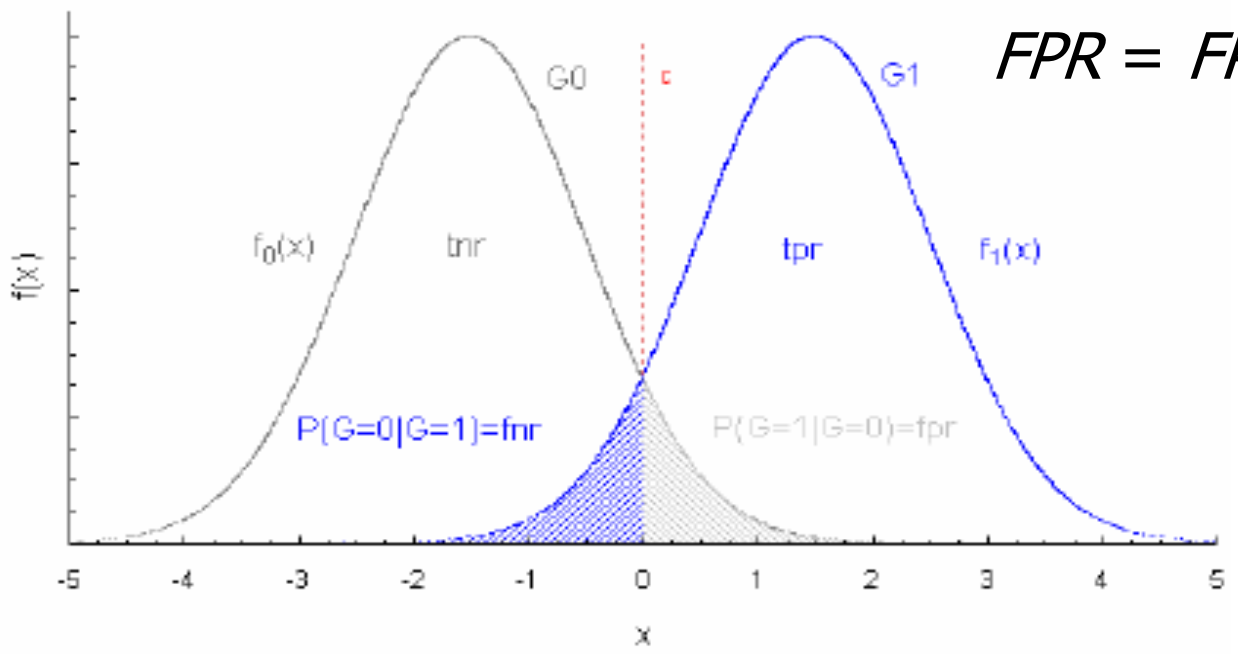
- *TN* (*true negative*) - počet správně klasifikovaných negativních případů
- *TP* (*true positive*) – počet správně klasifikovaných positivních případů
- *FP (false positive)* – počet nesprávně klasifikovaných negativních případů
- *FN (false negative)* – počet nesprávně klasifikovaných positivních případů

	Predikce		
Skuteč.	**G0**	**G1**	**Celkem**
G0	*TN*	*FP*	*N*
G1	*FN*	*TP*	*P*
Celkem	*PNeg*	*PPos*	*n*

ROC –TPR, FPR

$$TPR = TP / P = TP / (TP + FN)$$

$$FPR = FP / N = FP / (FP + TN)$$



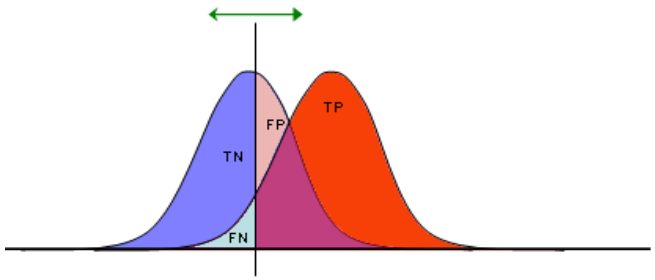
$$tpr(c) = P(X > c \mid G_1) = 1 - F_1(c)$$

$$tnr(c) = P(X < c \mid G_0) = F_0(c)$$

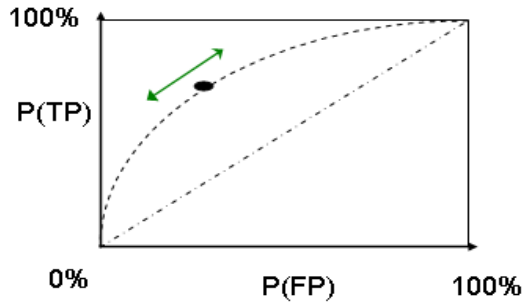
$$fpr(c) = P(X > c \mid G_0) = 1 - F_0(c)$$

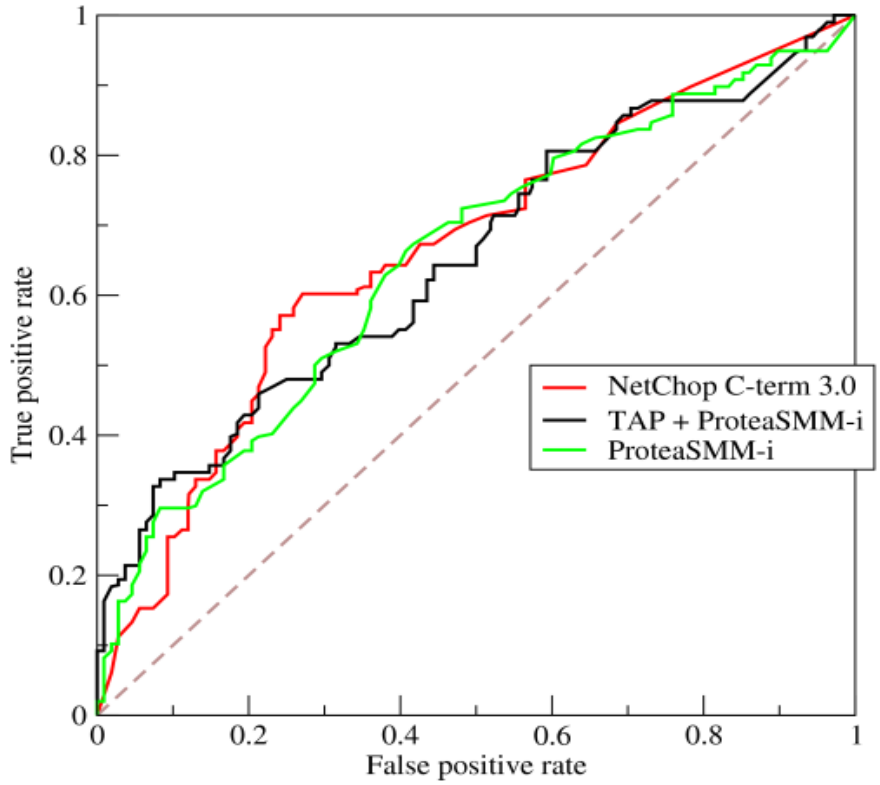
$$fnr(c) = P(X < c \mid G_1) = F_1(c)$$

ROC



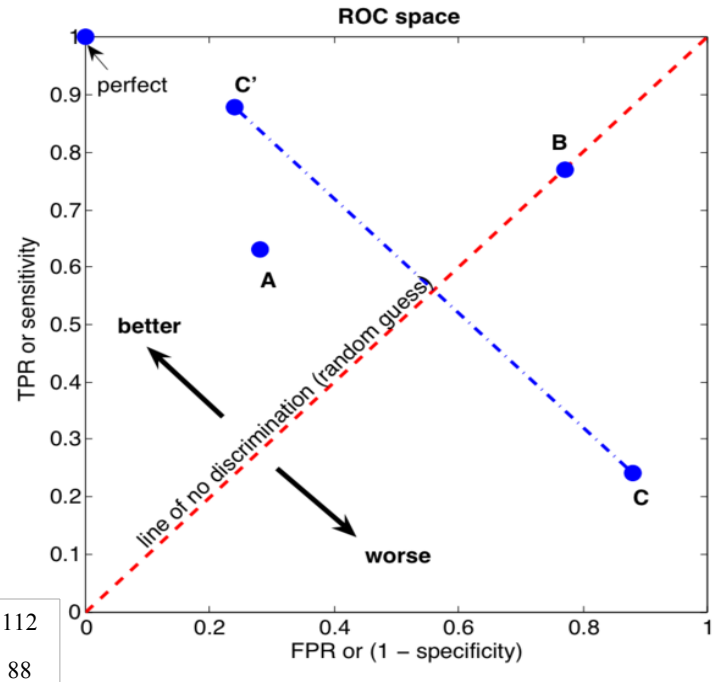
TP	FP
FN	TN
1	1





ROC - ACC

Accuracy:
ACC = (*TP* + *TN*) / (*P* + *N*)



A			B			C			C'		
TP=63	FP=28	91	TP=77	FP=77	154	TP=24	FP=88	112	TP=88	FP=24	112
FN=37	TN=72	109	FN=23	TN=23	46	FN=76	TN=12	88	FN=12	TN=76	88
100	100	200	100	100	200	100	100	200	100	100	200
TPR = 0.63			TPR = 0.77			TPR = 0.24			TPR = 0.88		
FPR = 0.28			FPR = 0.77			FPR = 0.88			FPR = 0.24		
ACC = 0.68			ACC = 0.50			ACC = 0.18			ACC = 0.82		

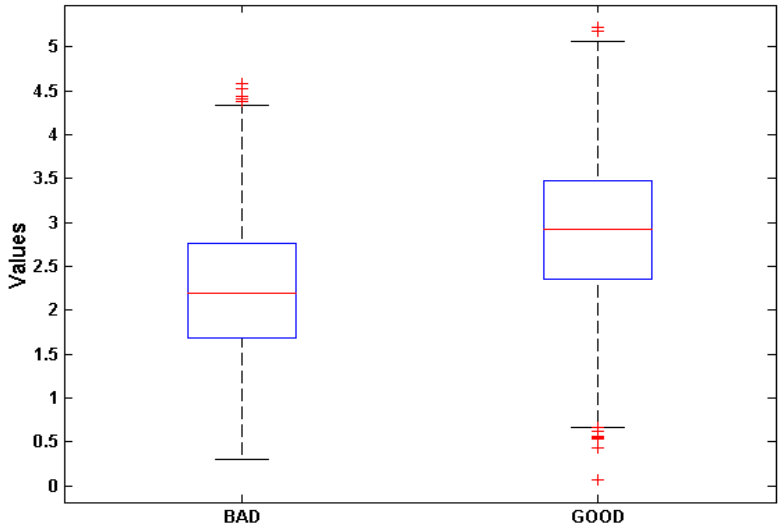
ROC – AUC, Gini

AUC (area under curve, neboli plocha pod ROC křivkou) je rovna pravděpodobnosti, že daný model ohodnotí náhodně vybraného dobrého klienta vyšším skóre než náhodně vybraného špatného klienta. Dá se ukázat, že plocha pod ROC křivkou se dá vyjádřit pomocí Mann-Whitneymu U, které testuje rozdíl mediánů mezi dvěma skupinami spojitých skóre. AUC se dá vyjádřit i pomocí Giniho koeficientu pomocí vzorce

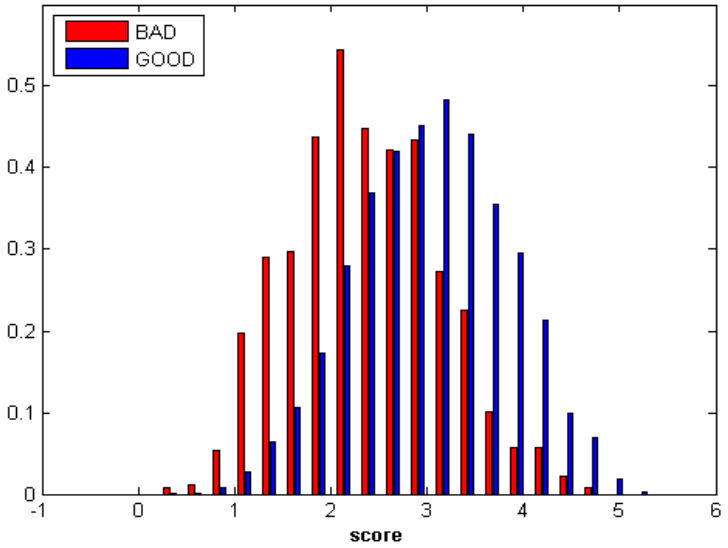
$Gini + 1 = 2xAUC$

Další evaluační grafy

Boxplot

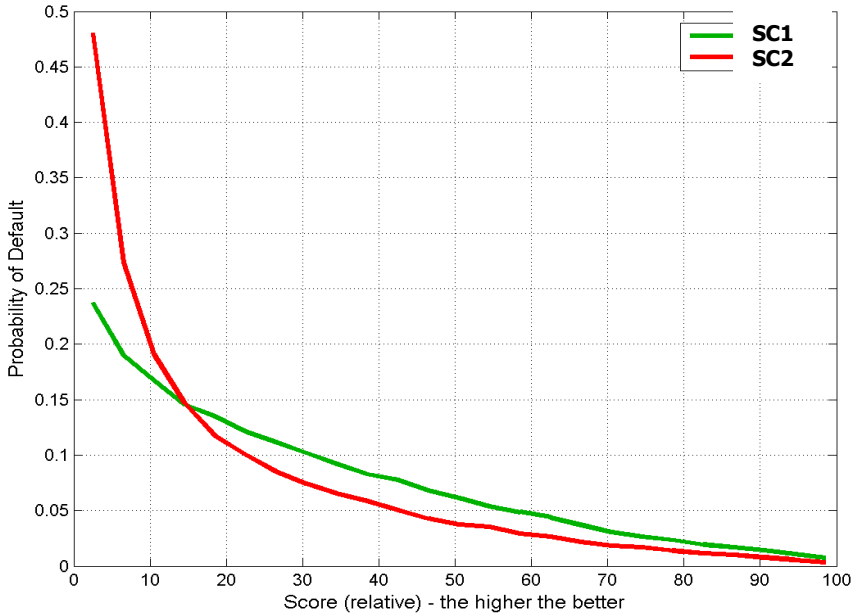


Histogram

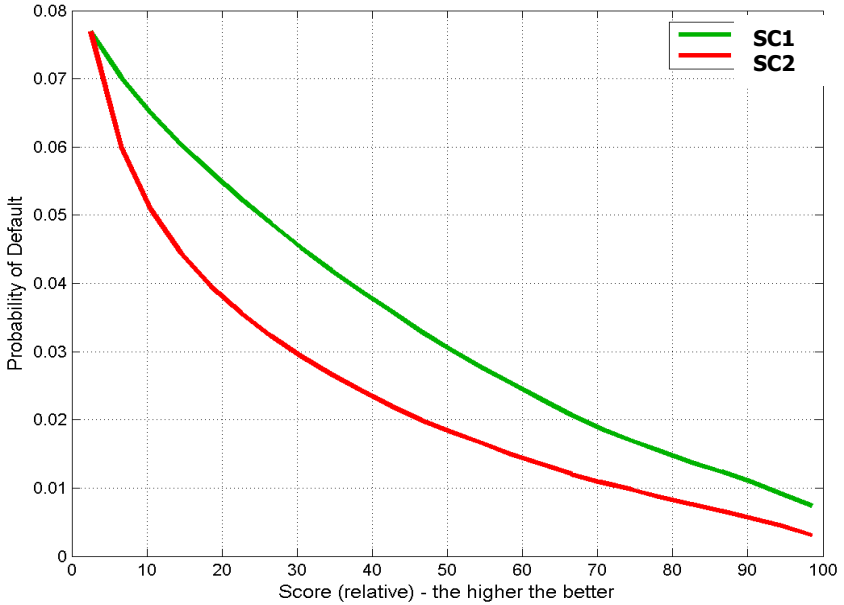


Další evaluační grafy

PD - absolutně

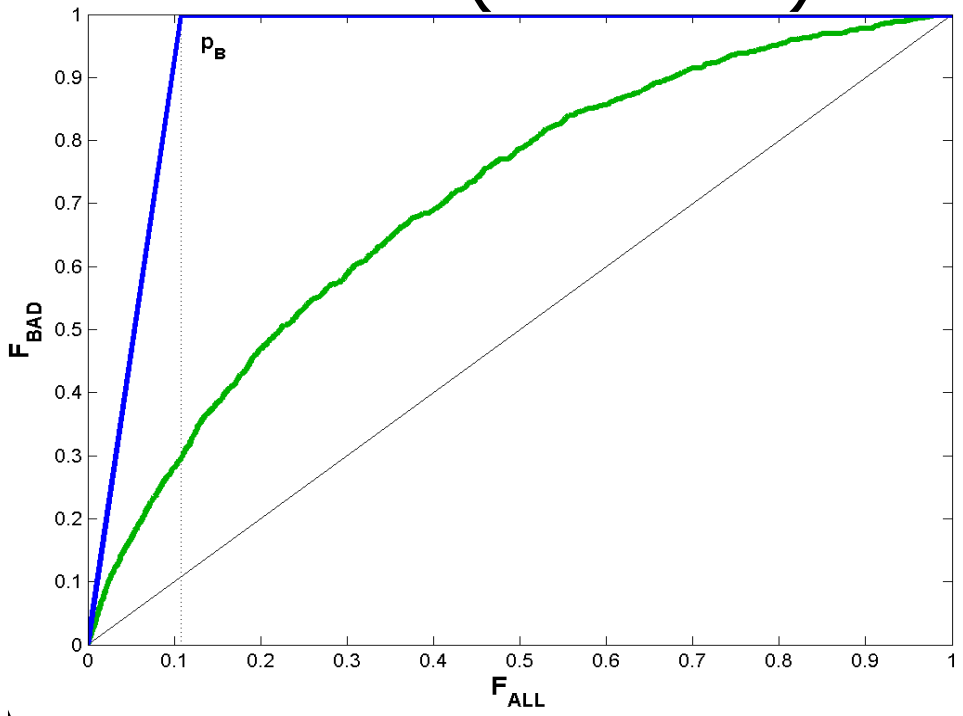


PD - kumulativně



Další evaluační grafy

CAP (Lift chart):



AR (Accuracy Ratio)

$$\mathrm{AR} = \frac{\text{Plocha mezi CAP a diagonálou}}{\text{Plocha mezi CAP ideálního modelu a diagonálou}}$$

$$= \frac{\text{Plocha mezi CAP a diagonálou}}{0.5(1-p_B)} = Gini$$

V tomto případě máme na x-ové ose proporci všech klientů (F_{ALL}) a na y-vé ose proporci špatných klientů (F_{BAD}). Ideální model je tentokrát reprezentován lomenou čarou z bodu [0, 0] přes [pB, 1] do bodu [1, 1]. Výhoda tohoto obrázku je ta, že je možné odečíst proporci zamítnutých špatných klientů vs. celková proporce zamítnutých klientů. Např. vidíme, že pokud chceme zamítnout 70% špatných klientů, musíme zamítat přibližně 40% všech žadatelů.

Postupy evaluace

- **evaluace na učících datech**

Evaluace na učících datech použitých k učícímu procesu není ke zjištění kvality modelu vhodná a má nízkou vypovídací schopnost, protože často může dojít k přeučení modelu. Odhad predikční kvality modelu na učících datech se nazývá resubstituční nebo interní odhad. Odhady ukazatelů kvality modelů provedených na učících datech jsou nadhodnocené, proto se místo nich používají testovací data, která se v rámci přípravy dat pro tyto účely vyčlení.

Postupy evaluace

- **evaluace na testovacích datech**

Evaluace na testovacích datech již má patřičnou vypovídací schopnost, jelikož tato data nebyla použita k sestavení modelu. Na testovací data jsou kladeny určité požadavky. Soubor testovacích dat by měl obsahovat dostatečné množství dat a měl by reprezentovat či vystihovat charakteristiky učících dat. Empiricky doporučený poměr učících a testovacích dat je 75%, resp. 25% případů. Zajištění patřičné reprezentativnosti je realizováno pomocí náhodného stratifikovaného výběru.

Postupy evaluace

➢ **křížové ověřování** *(cross-validation)*

V případě nedostatečného počtu pozorování, kdy rozdělení datového souboru na učící a testovací data za účelem vyhodnocení modelu není možné, je vhodné použít metodu křížového ověřování. Výhodou této metody na rozdíl od dělení datového souboru je, že každý případ z dat je použit k sestavení modelu a každý případ je alespoň jednou použit k testování. Postup je následující:

- Soubor dat je náhodně rozdělen do *n* disjunktních podmnožin tak, že každá podmnožina obsahuje přibližně stejný počet záznamů. Výběry jsou stratifikovány podle tříd (příslušnosti k určité třídě), aby bylo zajištěno, že podíly jednotlivých tříd podmnožin jsou zhruba stejné jako v celém souboru.
- Z těchto *n* disjunktních podmnožin se vyčlení *n-1* podmnožin pro sestavení modelu (konstrukční podmnožina) a zbývající podmnožina (validační podmnožina) je použita k jeho vyhodnocení. Model je tedy evaluován na podmnožině dat, ze kterých nebyl sestaven a na této množině dat je odhadována jeho predikční kvalita.
- Celý postup se zopakuje *n*-krát a dílčí odhady ukazatelů kvality se zprůměrňují. Velikost validační podmnožiny lze přibližně stanovit jako poměr počtu případů ku počtu validačních podmnožin.

Postupy evaluace

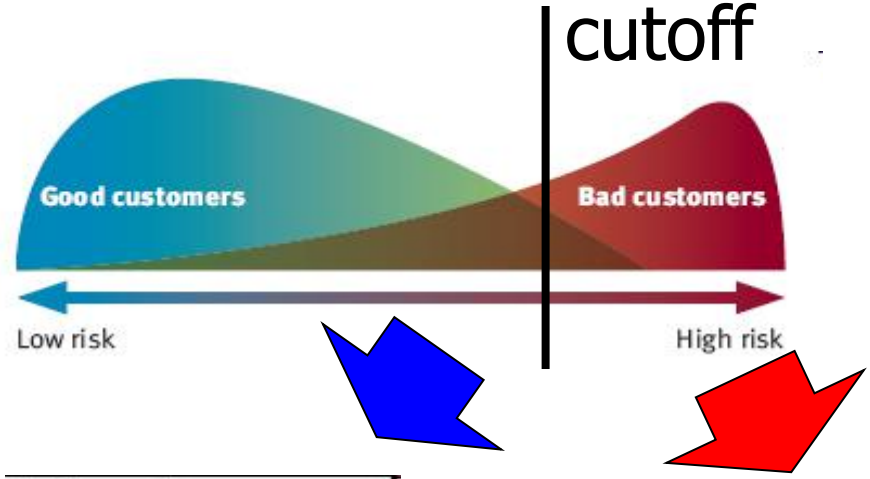
- ***bootstrap* metoda**

Metoda *bootstrap* zkoumá charakteristiky jednotlivých resamplovaných vzorků, které byly pořízeny z empirického výběru. Pokud původní výběr osahuje *m* prvků, tak každý má naději objevit se v resamplovaném výběru. Při úplném resamplování o velikosti vzorku *n* jsou uvažovány všechny možné výběry a existuje tedy m^n možných výběrů. Úplné resamplování je teoreticky proveditelné, ale vyžádalo by si mnoho času. Alternativou je simulace *Monte Carlo*, pomocí níž se aproximuje úplné resamplování tak, že se provede *B* náhodných výběrů (obvykle se volí 500 – 10000 výběrů) s tím, že každý prvek je vždy nahrazen (vrácen zpět do osudí). Jsou-li dána data $X=\{X1, \ldots, Xn\}$ a je-li požadován odhad parametru θ, provede se z původních dat *B* výběrů a pro každý výběr je spočítán odhad parametru θ . *Bootstrap* odhad parametru je určen jako průměr dílčích odhadů. V případě evaluace modelů bude parametrem θ zvolený ukazatel predikční kvality.

- ***jackknife***

Tato metoda je založena na sekvenční strategii odebírání a vracení prvků do výběru o velikosti *n*. Pro datový soubor, který obsahuje *n* prvků, procedura generuje *n* vzorků s počtem prvků *n-1*. Pro každý zmenšený výběr o velikosti *n*-1 je odhadnuta hodnota parametru. Dílčí odhady se následně zprůměrují podobně jako u metody *bootstrap*.

10. Cutoff, RAROA, Monitoring



Možné zamítací škály – cutoff

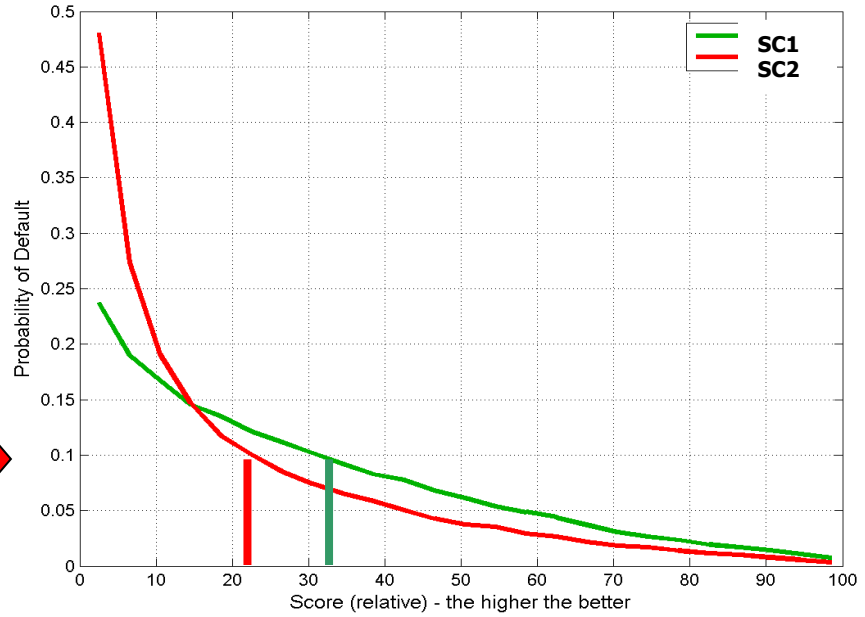
- cutoff hodnota určuje mez, při které je žádost o úvěr schválena/zamítnuta

- Je možné použít tyto zamítací škály:
  - **PD – Pravděpodobnost Defaultu (Probability of Default)**
  - **KRN - Kreditní Rizikové Náklady (CRE – Credit Risk Expenses)**
  - **Marže (Margin)**
  - **RAROA**
  - **…**

Cutoff na škále PD

cutoff = 0.1 (tj. zamítám všechny s pravděpodobností defaultu větší než 10 %)

Cut off



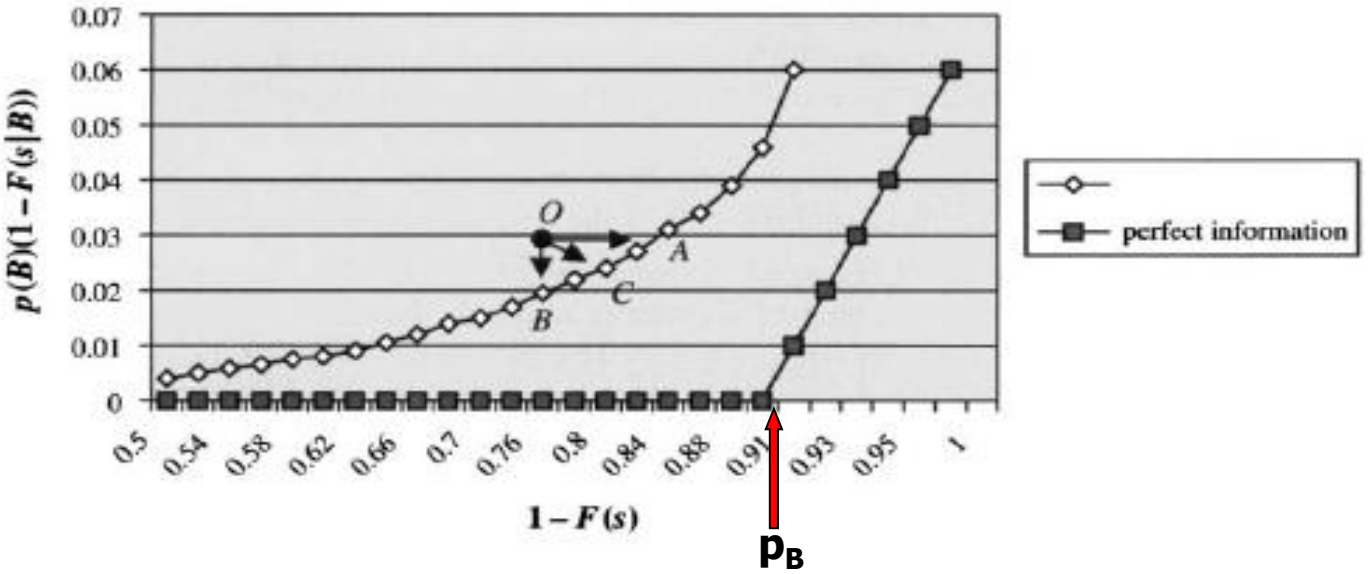
- Pro SC1 je reject rate 22 %.
- Pro SC2 je reject rate 33 %.

Strategická křivka (Strategy curve)

Bad acceptance rate $= p_B\left(1 - F\left(s|B\right)\right)$

Acceptance rate $= 1 - F(s)$

Actual bad rate $= \dfrac{p_B\left(1 - F\left(s|B\right)\right)}{1 - F(s)}$



Při zavádění nové scoringové funkce typicky dochází k tomu, že stávající nastavení schvalovacího procesu (nastavení cutoff) je reprezentováno bodem *O* , který leží nad novou strategickou křivkou. Otázkou pak je směr, kterým se chceme vydat při stanovení nového cutoff. Pokud se posuneme do bodu *A*, potom zachováme poměr schválených špatných klientů, ale současně zvýšíme celkový poměr schválených klientů. Při posunu do bodu *B* schválíme stejný poměr klientů, ale snížíme poměr schválených špatných klientů a tedy i poměr špatných klientů (bad rate). Posunem do bodu *C* zachováme bad rate při současném zvýšení poměru schválených klientů.

Nastavení cutoff maximalizující zisk (profit)

Profit - náhodná veličina definovaná jako:

$$R = \begin{cases} 0, & \text{je – li úvěr zamítnut} \\ L, & \text{je – li úvěr schválen a stane se dobrým} \\ -D, & \text{je – li úvěr schválen a stane se špatným} \end{cases}$$

Označme p_G a p_B proporci dobrých a špatných klientů v populaci. $q(G|s)$ ($q(B|s)$) označuje podmíněnou pravděpodobnost, že klient mající skóre s bude dobrý (špatný), přičemž $q(G|s) + q(B|s) = 1$. Nechť $p(s)$ je proporce populace se skóre s.

Střední hodnota profitu při schválení klientů se skóre s:

$$E\{R|s\} = Lq(G|s) - D(1 - q(G|s)) = (L + D)q(G|s) - D$$

Tedy k maximalizaci profitu je třeba schválit ty klienty, jejichž skóre splňuje podmínku:

$$q(G|s) \geq \frac{D}{D+L}$$

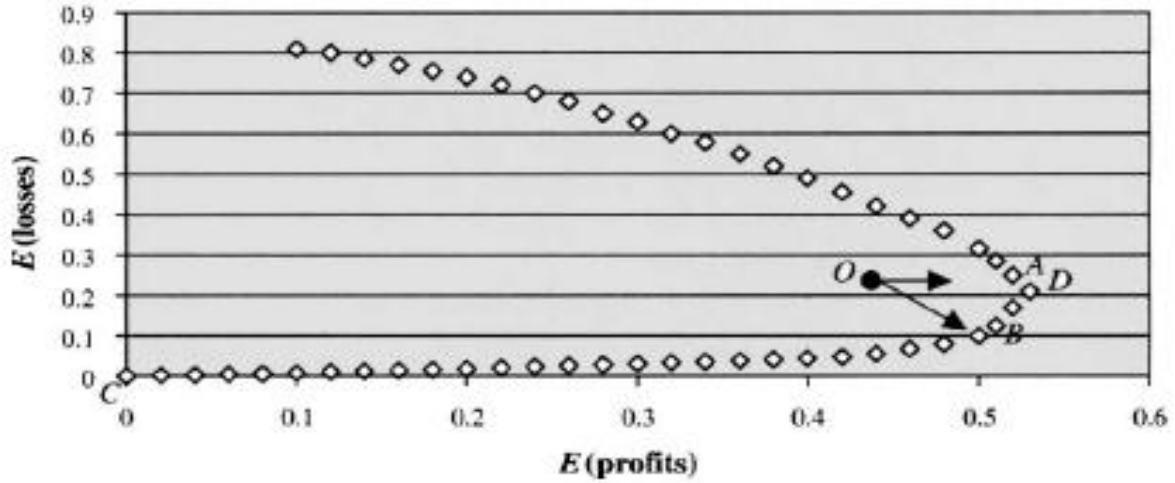
Nastavení cutoff maximalizující profit

Nechť A označuje množinu skóre, kde je splněna předchozí podmínka. Pak je střední hodnota zisku (profitu) na jednoho klienta dána vztahem:

$$E^*\{R\} = \sum_{s \in A} ((L + D)q(G|s) - D)p(s).$$

Pokud L a D navíc závisí na skóre s, je situace ještě o něco složitější. Více viz Thomas et al. (2002).

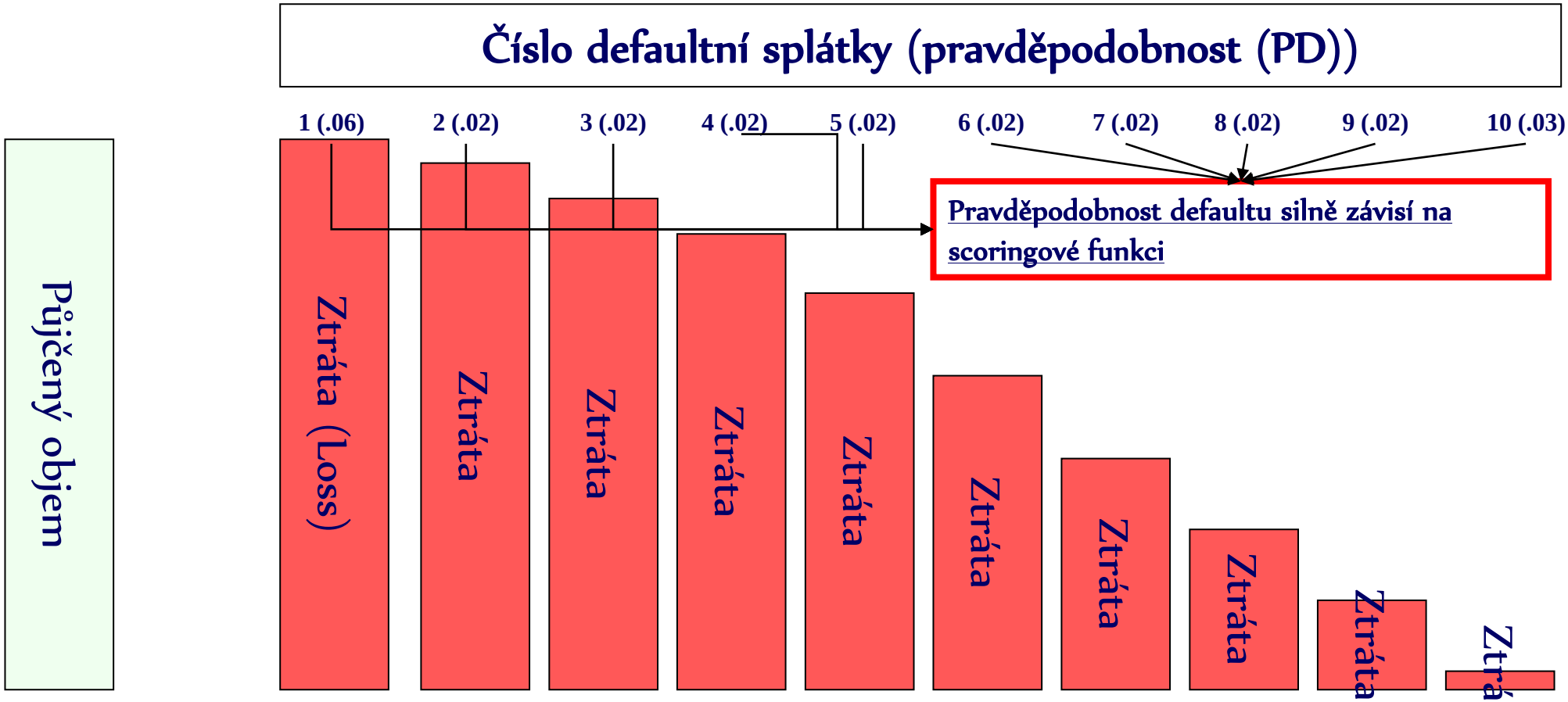
Nastavení cutoff maximalizující profit



Body na spodní části křivky odpovídají vyšším cutoff hodnotám, a tedy i menšímu počtu přijatých špatných klientů, zatímco body na horní části křivky odpovídají menším hodnotám cutoff, tj. vyššímu počtu přijatých špatných klientů. Efektivní hranicí je tedy spodní část křivky od bodu C do bodu D.

Jestliže aktuální nastavení schvalovacího procesu odpovídá bodu *O*, opět máme možnost posunu na křivku odpovídající nové scoringové funkci. První možností je zachování poměru schválených špatných klientů, tj. posun do bodu *A*. Druhou možností je zachování celkového poměru schválených klientů, tj. posun do bodu *B*. Je zřejmé, že posun do bodu *A* není vhodná volba, protože tento bod neleží na efektivní hranici a lze snadno dosáhnout stejného očekávaného zisku při nižší očekávané ztrátě.

Definice KRN (CRE)



CRE = ((1-Recovery) * SUM(PD * Loss))/(Expected Average Volume)

Profit = (Interest rate – CRE)*Expected Average Volume

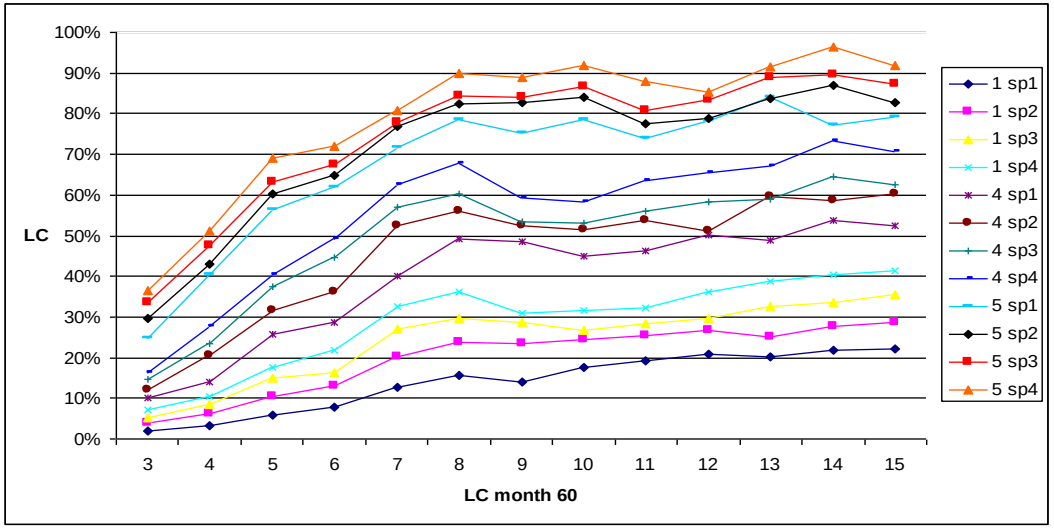
Úroková míra

Očekávaný průměrný objem úvěru

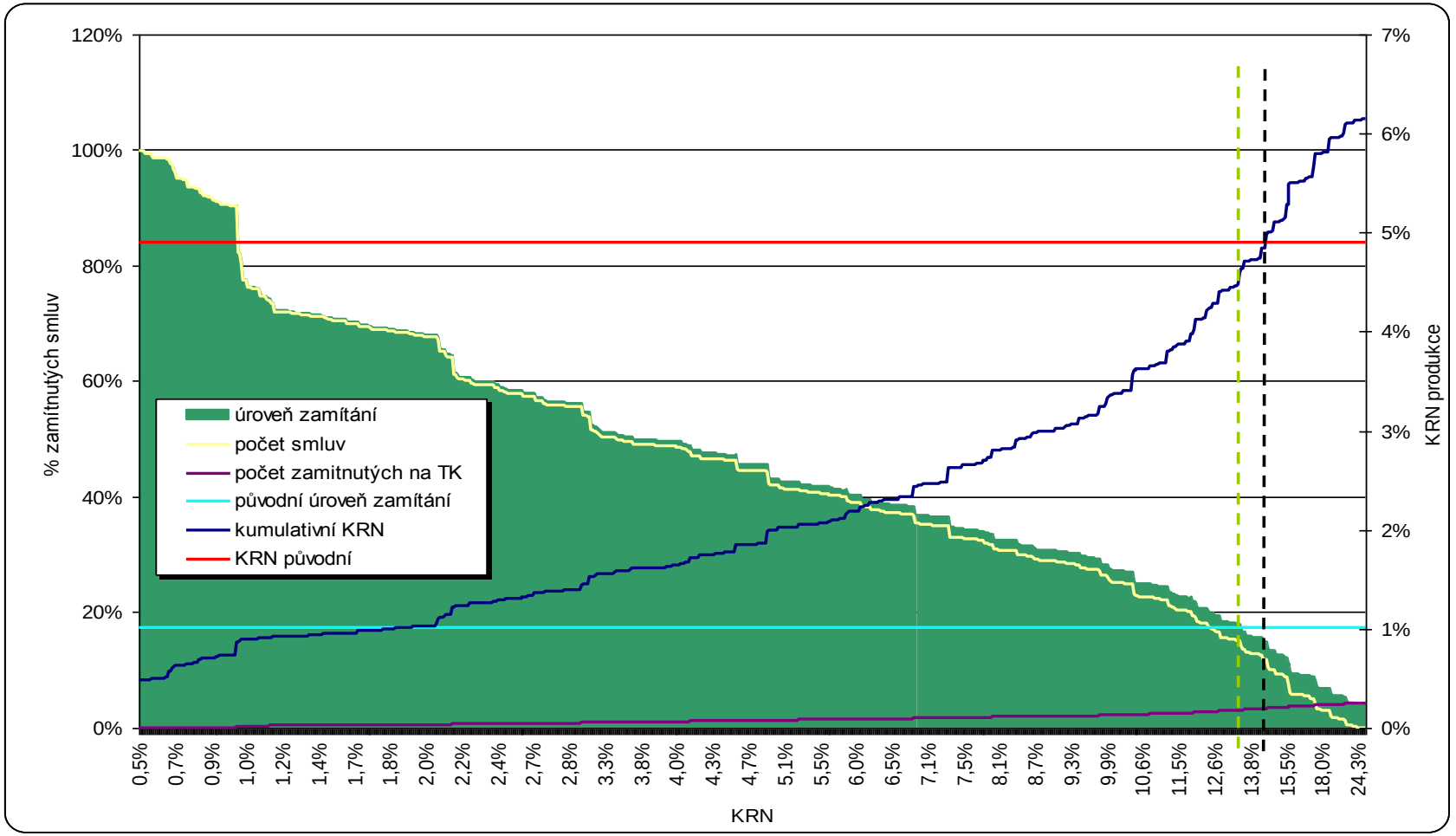
Recovery (=Late collection(LC))

Číslo defaultní splátky	score			
	band1	band2	band3	band4
1.	**20%**	**25%**	**30%**	**35%**
2.-4.	**50%**	**55%**	**60%**	**65%**
5. +	**75%**	**80%**	**85%**	**90%**

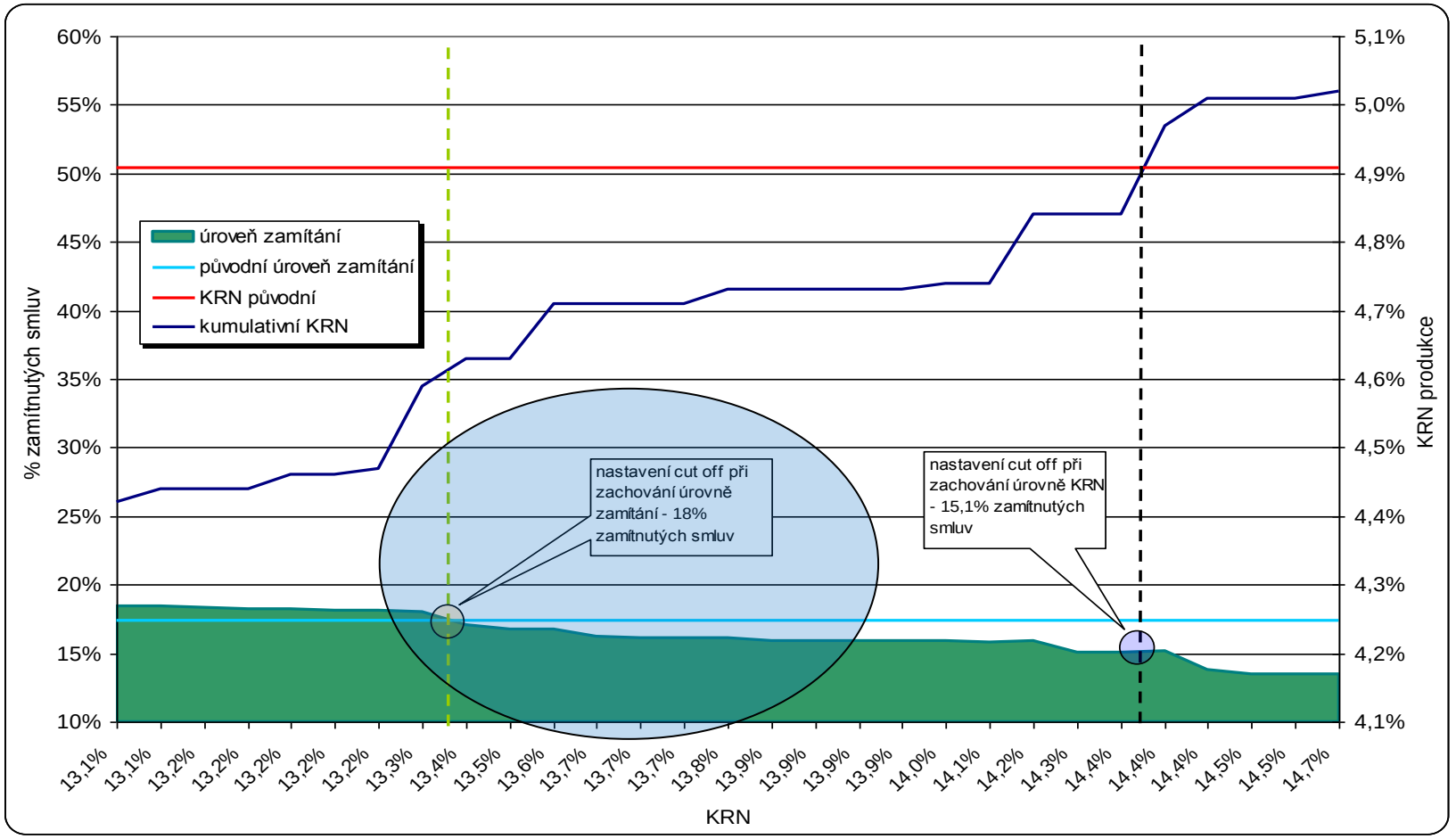
odhad



Cutoff na škále KRN



Cutoff na škále KRN



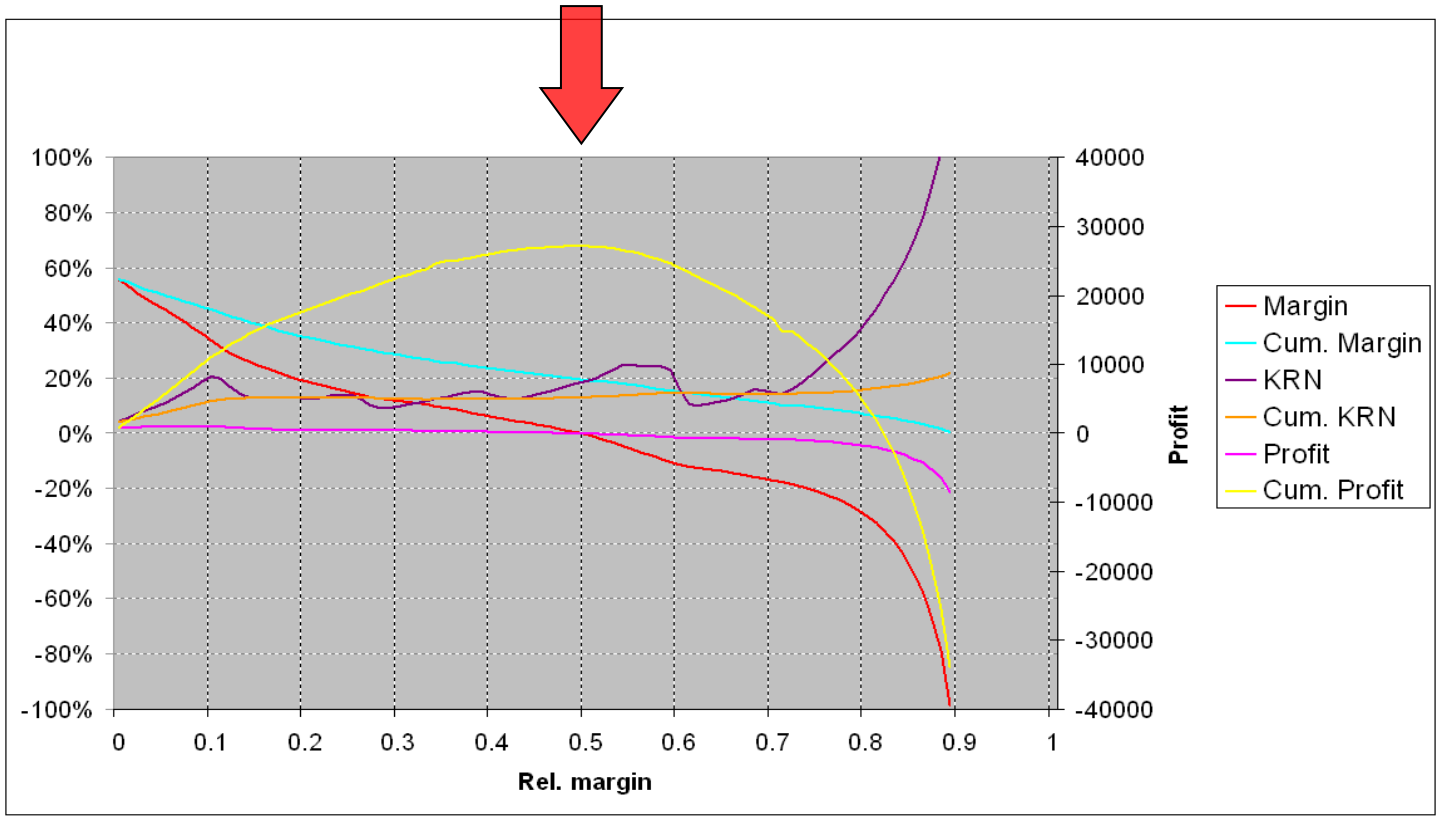
(Očekávaná) Marže

(Očekávaná) Marže = Úroková míra (vč. poplatků) – KRN – OPEX

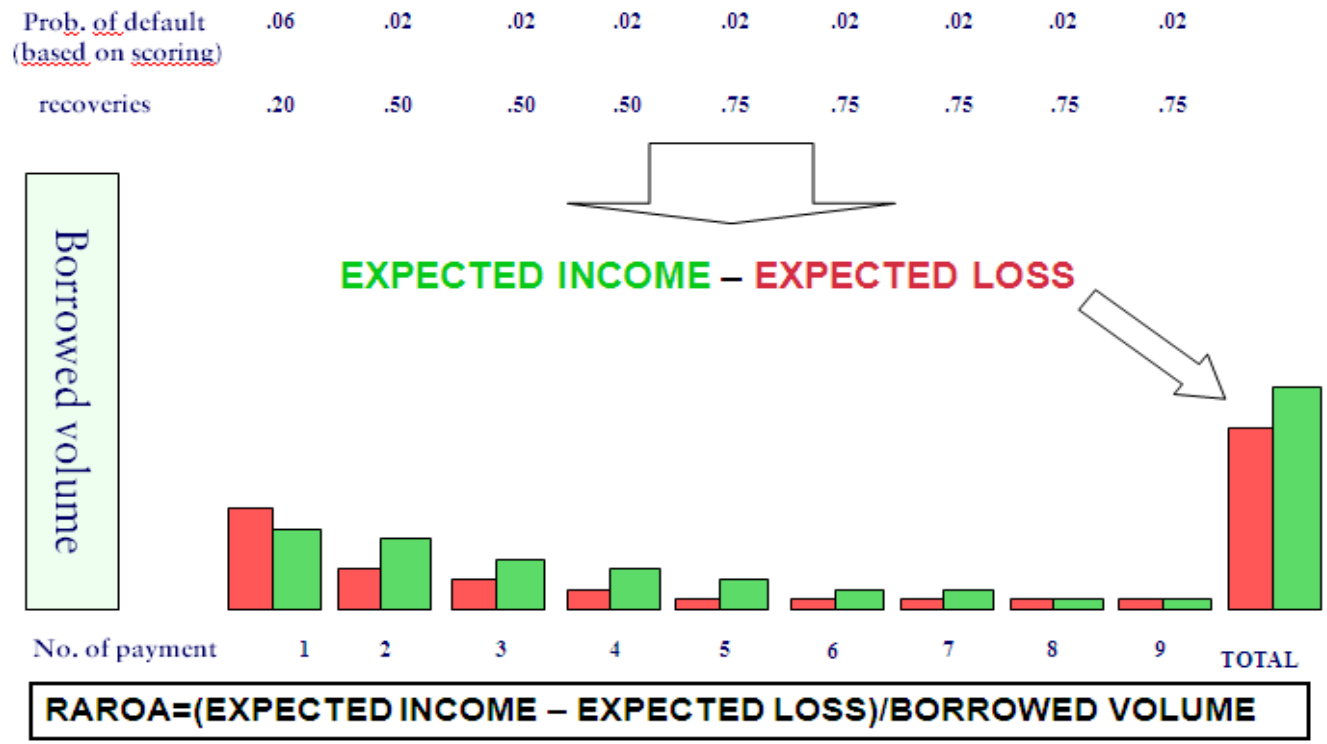
- ***Úroková míra***
  - *Efektivní míra ideálního finančního toku (-výše úvěru-poplatky; anuita; anuita; … ; anuita).*
- ***KRN***
  - *Viz výše.*
- ***OPEX***
  - *Cena peněz.*
  - *Režijní náklady, variabilní náklady, podpora prodejní sítě.*
  - *Náklady na administrátory – vlastní zaměstnance zajišťující zpracování úvěru.*

Marže (Margin)

- Optimální cutoff: *marže=0*



RAROA
(Risk Adjusted Return On Assets)



RAROA

- $t \ldots$ pořadí splátky úvěru, 0 je okamžik poskytnutí úvěru
- $T \ldots$ počet splátek
- $x(t) \ldots$ nesplacená část úvěru podle splátkového plánu v čase $t, \ldots t = 0, \ldots, T$. $x(0)$ je výše úvěru, $x(T) = 0$.
- $u(t) \ldots$ úroková část anuity t, $t = 1, \ldots, T$.
- $j(t) \ldots$ část anuity odpovídající splátce jistiny t, $t = 1, \ldots, T$.
- $k(t) \ldots$ komise od klienta v čase t, $t = 0, \ldots, T$.
- $A \ldots$ výše anuity (absolutně). $A = u(t) + j(t), t = 1, \ldots, T$.
- $p(t) \ldots$ pravděpodobnost 90 denního defaultu úvěru na splátce t, $t = 1, \ldots, T$
- $\mathrm{E}\, Z \ldots$ očekávaná ztráta z úvěru
- $\mathrm{E}\, P \ldots$ očekávaný úrokový příjem z úvěru
- $RC \ldots$ absolutní výše z dlužné částky klienta 90 dní po splatnosti, která je klientem splacena v budoucnu, přepočtena přes NPV k okamžiku devadesátidenního defaultu klienta
- $r(t, f) \ldots$ procento výtěžnosti z dlužné částky klienta, který je poprvé 90 dní po splatnosti na splátce t a klient má hodnotu podvodnického skóre (nesplacení první splátky) f. Procento zohledňuje NPV všech budoucích splátek klienta po okamžiku defaultu.

RAROA

- GM ... hrubý očekávaný zisk z klienta
- s je sazba úvěru p.a.
- i... cena zdrojů vyjádřená v procentu p.a.
- c... komise z obchodu poskytnutá obchodnímu partnerovi vyjádřená jako procento z jistiny
- NM_I ... čistý očekávaný zisk typu I z klienta po odečtení ceny zdrojů
- NM_{II} ... čistý očekávaný zisk z klienta typu II po odečtení ceny zdrojů a komisí z obchodu.¨¨
- ROA... ukazatel Return on Asset počítaného z hrubého zisku
- ROA_I ... ukazatel Return on Asset typu I počítaný z čistého zisku typu I
- ROA_{II} ... ukazatel Return on Asset typu II počítaný z čistého zisku typu II
- KRN je úroková míra p.a. vyjadřující rizikovost úvěru.

RAROA

$$\mathrm{E}\,Z = \sum_{t=1}^{T} p(t) \cdot x(t-1).$$

$$\mathrm{E}\,P = k(0) + \sum_{t=1}^{T} \left(1 - \sum_{s=1}^{t} p(s)\right) (u(t) + k(t)).$$

$$GM = \mathrm{E}\,P - \mathrm{E}\,Z + RC.$$

$$RC = \sum_{t=1}^{T} p(t) \cdot r(t,f) \cdot x(t-1)$$

$$NM_I = GM - \sum_{t=1}^{T} \left(1 - \sum_{s=1}^{t} p(s)\right) \frac{i}{12} \cdot x(t-1).$$

$$NM_{II} = NM_I - c \cdot x(0).$$

$$ROA = \frac{GM}{x(0)}.$$

$$ROA_I = \frac{NM_I}{x(0)}.$$

$$ROA_{II} = \frac{NM_{II}}{x(0)}.$$

$$\frac{\mathrm{KRN}}{12} \cdot \sum_{t=1}^{T} \left(1 - \sum_{s=1}^{t} p(s)\right) x(t-1) = EZ - RC.$$

$$\sum_{t=1}^{T} \left(1 - \sum_{s=1}^{t} p(s)\right) x(t-1) = \frac{\sum_{t=1}^{T} \left(1 - \sum_{s=1}^{t} p(s)\right) u(t)}{s/12},$$

$$\mathrm{KRN} = \frac{EZ - RC}{EP} \cdot s.$$

Výhody RAROA

	Case A		Case B	
	Ideal flow	Expected flow	Ideal flow	Expected flow
	-1000	-1000	-1000	-1000
1	400	200	150	110
2	400	180	150	100
3	400	170	150	90
4	400	160	150	80
5			150	70
6			150	60
7			150	50
8			150	40
9			150	30
10			150	16
11			150	10
12			150	0

- A – krátkodobý úvěr s vysokým rizikem fraudu
- B – dlouhodobý úvěr s vysokým rizikem defaultu

Úroková míra (A) = 22%
Úroková míra (B) = 10%

KRN(A) = 44%
KRN(B) = 20%
cutoff na škále KRN preferuje B

Marže (A) = -22%
Marže (B) = -10%
cutoff na škále marže preferuje B

RAROA (A) = -0.29
RAROA (B) = -0.36
cutoff na škále RAROA preferuje A

Úvěr A je lepší, protože z něj plyne vyšší zisk (710>656), navíc je ho dosaženo mnohem dříve.

Cutoff segmentace

- ***Možná segmentace podle:***
  - ***Prodejní síť (skupina obchodních míst)***
  - ***Profitabilita produktu***
  - ***Kvalita prodejního místa***
  - ***Typ zboží (pro spotřebitelské úvěry)***
  - ***Výše úvěru***
  - ***…***

Cutoff scénáře

all

scenario	All credits		Approved credits		
	Reject rate		Avg. margin		Avg. KRN
	Credits	Volume	Credits	Volume	
0reject	0.0%	0.00%	-7.83%	-21.34%	30.70%
tk	22.4%	24.74%	-3.97%	-15.72%	26.33%
current	36.8%	46.79%	2.32%	-4.58%	19.11%
all30	29.5%	37.07%	3.18%	-3.36%	19.78%
total30	30.3%	38.55%	4.11%	-1.29%	19.26%
total32	31.5%	39.99%	4.63%	-0.77%	18.96%
total35	35.8%	46.01%	7.32%	3.22%	16.07%
all40	38.9%	48.97%	8.17%	3.41%	15.19%
tk_ekonom	59.3%	70.03%	19.39%	17.14%	13.47%
ekonom	50.9%	63.64%	19.24%	17.03%	14.23%

new

scenario	All credits		Approved credits		
	Reject rate		Avg. margin		Avg. KRN
	Credits	Volume	Credits	Volume	
0reject	0.0%	0.00%	-4.19%	-16.40%	26.19%
tk	9.0%	10.25%	-2.39%	-13.71%	24.50%
current	24.5%	34.89%	3.26%	-3.42%	17.99%
all30	16.5%	23.87%	4.07%	-2.34%	18.60%
total30	17.3%	25.42%	4.85%	-0.59%	18.19%
total32	18.6%	27.16%	5.36%	-0.03%	17.87%
total35	23.2%	33.80%	7.74%	3.52%	15.34%
all40	26.7%	37.35%	8.61%	3.88%	14.45%
tk_ekonom	50.7%	62.90%	19.53%	17.26%	13.05%
ekonom	47.5%	60.46%	19.48%	17.23%	13.26%

Cutoff impact evaluation

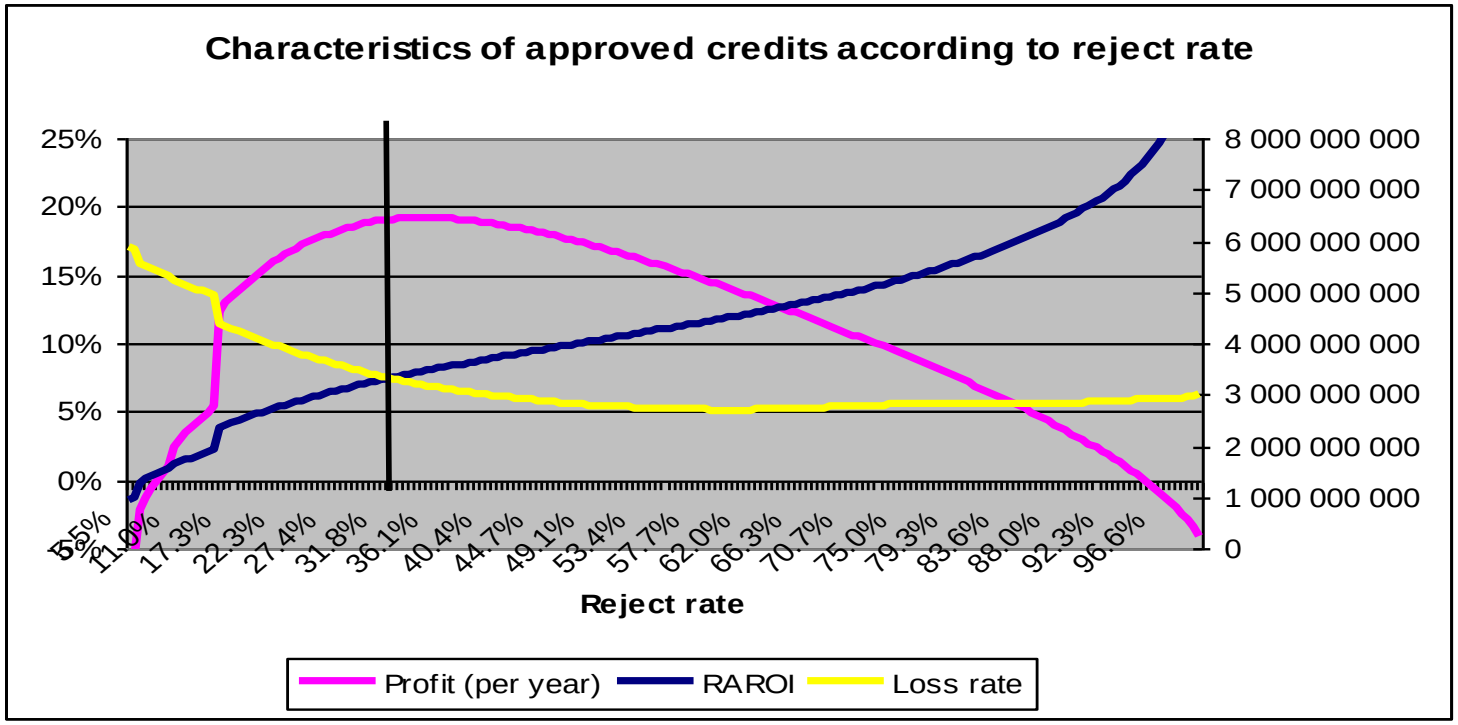
Evaluation of Reject rate, Profitability, Default and Loss rates before and after cutoff change according to Distribution channel or Segment of scorecard.

Cutoff impact evaluation table

	Before Christmas (approved credits)				After Christmas (approved credits)			
	Reject rate	RAROA	Loss rate	Profit (per year)	Reject rate	RAROA	Loss rate	Profit (per year)
Segment 1	24.7%	3.65%	11.33%	414 363 110	24.3%	3.75%	11.19%	428 757 430
Segment 2	12.1%	4.01%	8.22%	160 364 072	12.9%	3.95%	8.29%	159 917 943
Segment 3	45.1%	9.64%	9.69%	747 636 468	45.1%	9.8%	9.5%	758 966 512
Segment 4	22.2%	5.80%	4.89%	52 213 720	20.1%	5.62%	5.05%	51 715 263
Segment 5	20.9%	6.77%	5.41%	54 312 614	19.7%	6.61%	5.48%	53 975 903
Segment 6	33.4%	7.04%	7.22%	212 090 365	32.6%	7.04%	7.16%	211 684 371
Segment 7	49.3%	9.30%	8.93%	36 840 287	49.2%	9.4%	8.8%	37 140 165
Segment 8	19.3%	4.68%	2.96%	15 668 962	14.9%	4.54%	3.16%	15 636 910
Segment 9	32.0%	8.41%	5.06%	3 679 430	27.2%	7.97%	5.26%	3 535 809
Segment 10	33.4%	7.14%	6.69%	1 823 050 341	33.4%	7.2%	6.6%	1 832 986 599
Segment 11	28.5%	6.34%	7.36%	2 633 609 071	28.6%	6.47%	7.24%	2 651 352 740
ALL	**32.6%**	**6.64%**	**8.37%**	**6 153 828 440**	**32.6%**	**6.96%**	**8.17%**	**6 205 669 645**

Cutoff sensitivity analysis

Profitability, Default and Loss rates according to reject rate into one graph

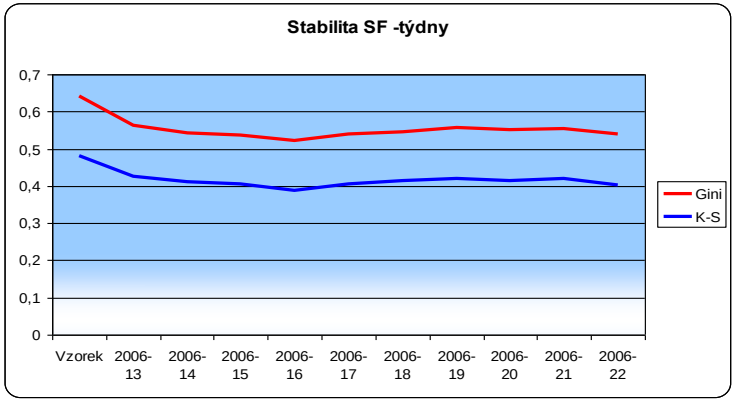


<u>Decision</u>

Reasoning, why the final cutoffs were chosen

Monitoring

	výv. vzorek [1]	týden1 [2]	[3]=[2] -[1]	[4]=[2]/[1]	[5]=ln[4]	[6]=[3]*[5]
skóre_1	10,00%	5,63%	-0,044	0,563	-0,574	0,025
skóre_2	10,00%	11,21%	0,012	1,121	0,114	0,001
skóre_3	10,00%	11,00%	0,010	1,100	0,095	0,001
skóre_4	10,00%	10,97%	0,010	1,097	0,092	0,001
skóre_5	10,00%	10,31%	0,003	1,031	0,031	0,000
skóre_6	10,00%	10,12%	0,001	1,012	0,012	0,000
skóre_7	10,01%	9,62%	-0,004	0,961	-0,039	0,000
skóre_8	10,00%	9,89%	-0,001	0,989	-0,011	0,000
skóre_9	10,00%	10,31%	0,003	1,031	0,030	0,000
skóre_10	10,00%	10,94%	0,009	1,095	0,091	0,001
					PSI	0,030



Monitoring scoringových modelů

- Není překvapivé, že prediktivní modely se ve statistickém slova smyslu chovají nejlépe na vývojovém vzorku dat. Výstupy těchto modelů, např. skóre nebo rating klienta, jsou počítány pomocí jistých vzorců, jejichž koeficienty příslušející nezávislým proměnným (prediktorům) jsou odvozeny na datech vývojového vzorku. Posun distribuce výstupu daného modelu je pak zapříčiněn právě změnou vstupních hodnot modelu, tj. prediktorů, v průběhu času. V podstatě ihned (alespoň většinou) po nasazení prediktivního modelu do praxe dochází k jistému poklesu jeho prediktivní síly, který je způsoben určitou změnou vstupních hodnot modelu. Zásadní je v praxi nastavení takových procesů, které odhalí, že se tak děje, proč se tak děje a jak vážný problém to ve svých důsledcích znamená.

Monitoring scoringových modelů

- Faktorů způsobujících posun v distribuci prediktorů, a následně posun v distribuci výstupu prediktivního modelu, je několik:

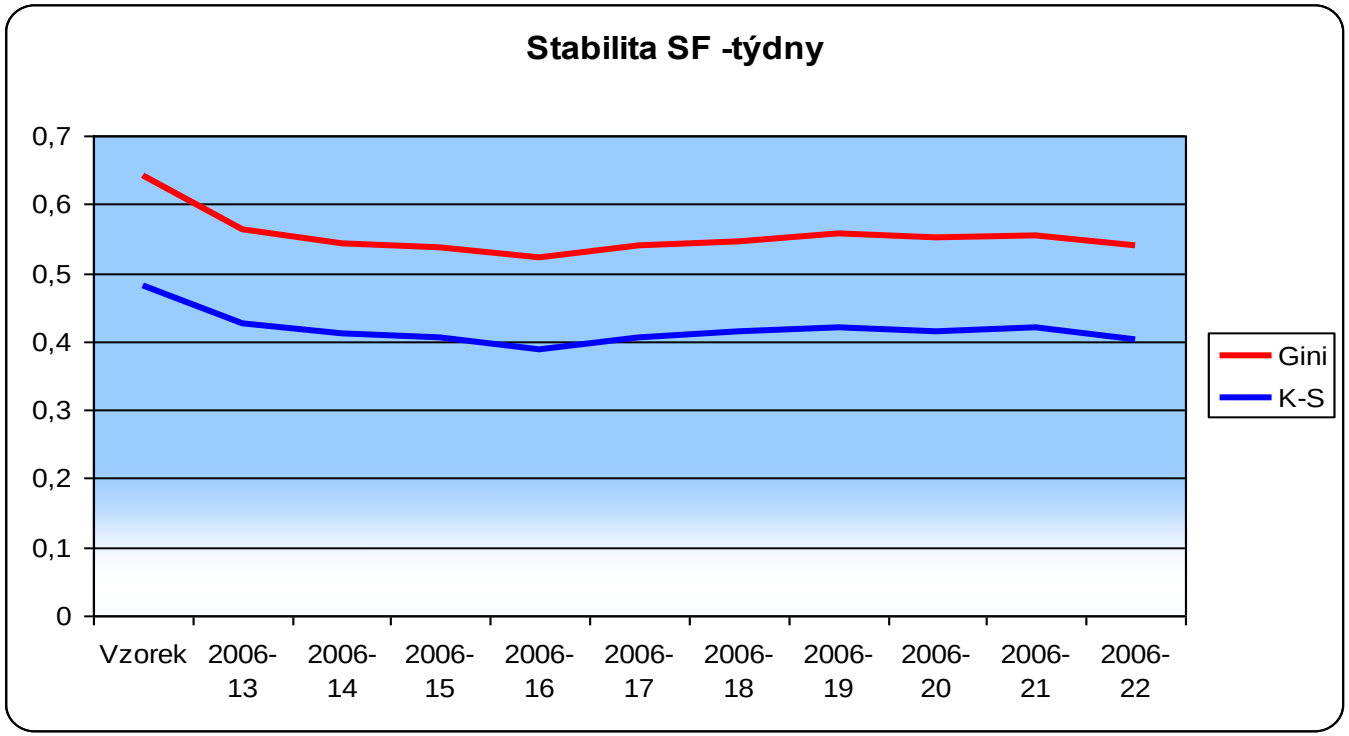
  - Přirozený posun v datech/změna demografické struktury dat
  - Databázové chyby
  - Změna datového zdroje
  - Změna definice/formátu vstupních dat
  - Změna datového univerza
  - Ostatní

Monitoring scoringových modelů

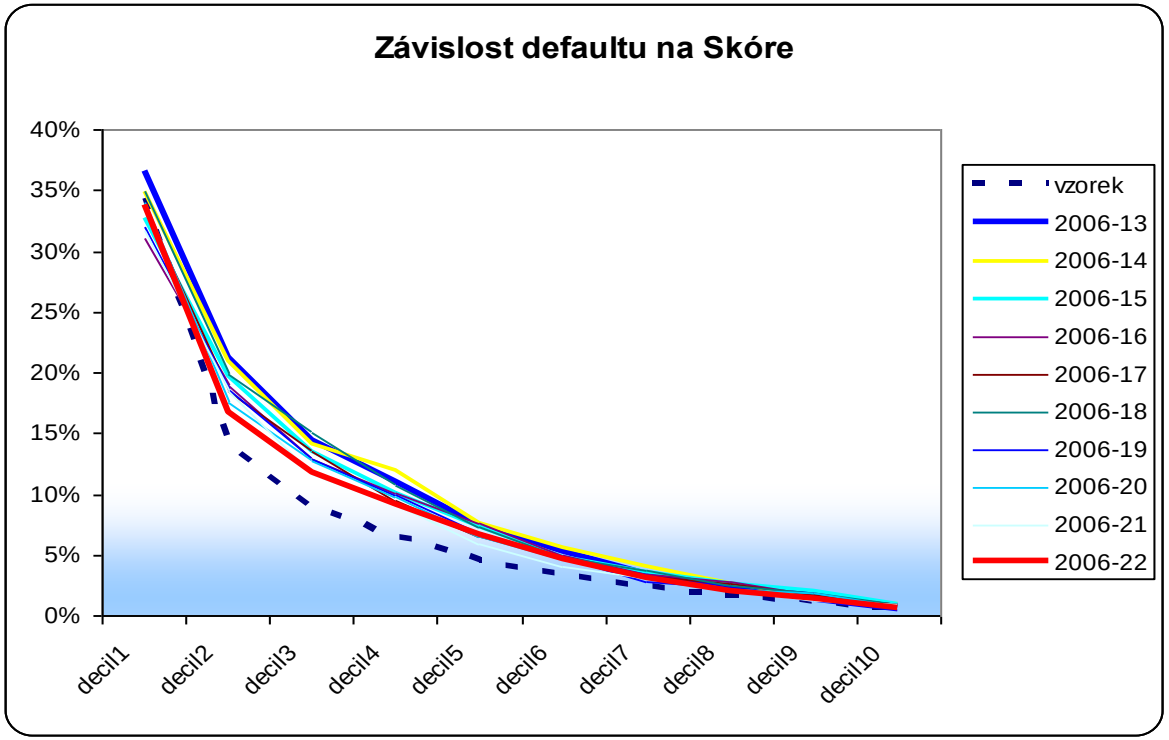
- Typickým příkladem prvního uvedeného důvodu je příjem klienta (všeobecným trendem je růst příjmu populace). Změnou definice/formátu vstupních dat je myšlena například situace, kdy je rozšířen číselník hodnot, kterých může vstupní proměnná nabývat. Změnou datového univerza je myšlen případ kdy je vyvinutý prediktivní model použit např. pro odlišný/nový segment portfolia nebo odlišný/nový produkt.

Monitoring scoringových modelů

- K-S, Gini:



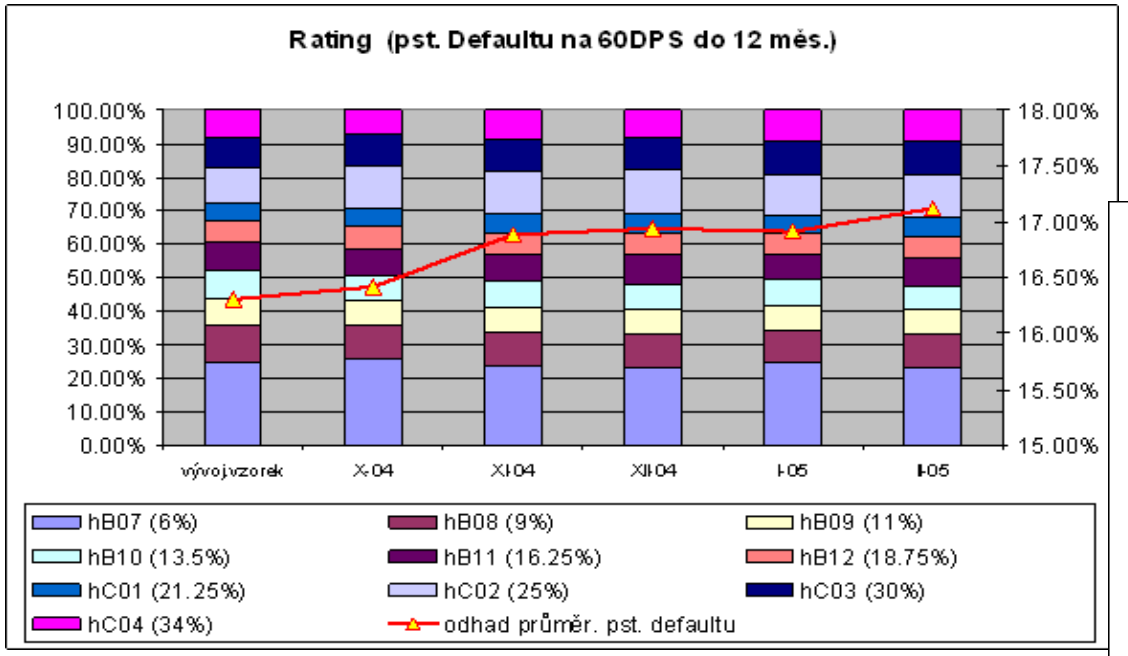
Monitoring scoringových modelů

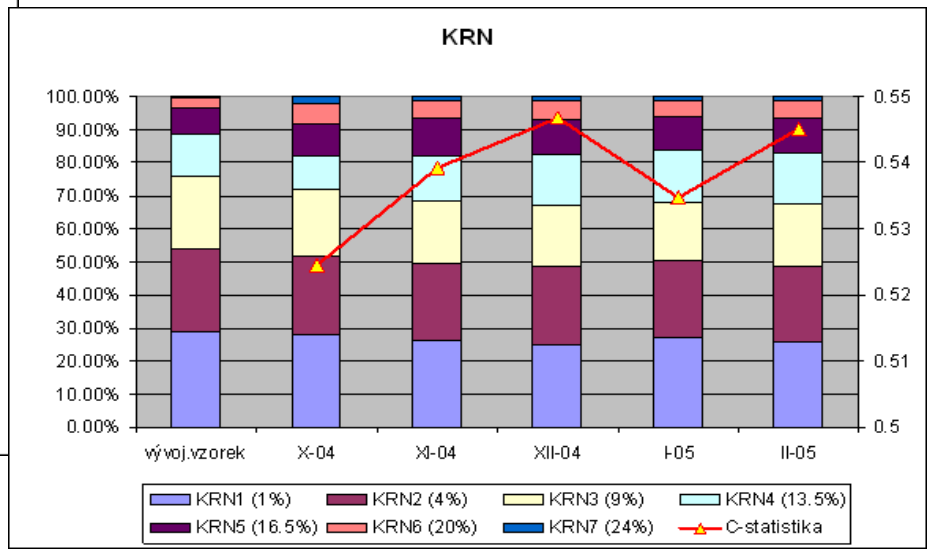


- Čím strmější křivka tím lépe.
- V průběhu času se zplošťuje – jde o to, jak moc.

Monitoring scoringových modelů

- c-statistika:





Monitoring scoringových modelů

- Chceme posoudit zda se distribuce skóre na vývojovém vzorku liší od distribuce skóre v daném časovém intervalu:

$$\chi^2 = \sum_{i=1}^{r} \frac{(O_i - E_i)^2}{E_i}$$

$$PSI = \sum_{i=1}^{r} (O_i - E_i) \ln\left(\frac{O_i}{E_i}\right)$$

Monitoring scoringových modelů

	výv. vzorek [1]	týden1 [2]	[3]=[2] -[1]	[4]=[2]/[1]	[5]=ln[4]	[6]=[3]*[5]
skóre_1	10,00%	5,63%	-0,044	0,563	-0,574	0,025
skóre_2	10,00%	11,21%	0,012	1,121	0,114	0,001
skóre_3	10,00%	11,00%	0,010	1,100	0,095	0,001
skóre_4	10,00%	10,97%	0,010	1,097	0,092	0,001
skóre_5	10,00%	10,31%	0,003	1,031	0,031	0,000
skóre_6	10,00%	10,12%	0,001	1,012	0,012	0,000
skóre_7	10,01%	9,62%	-0,004	0,961	-0,039	0,000
skóre_8	10,00%	9,89%	-0,001	0,989	-0,011	0,000
skóre_9	10,00%	10,31%	0,003	1,031	0,030	0,000
skóre_10	10,00%	10,94%	0,009	1,095	0,091	0,001
					PSI	0,030

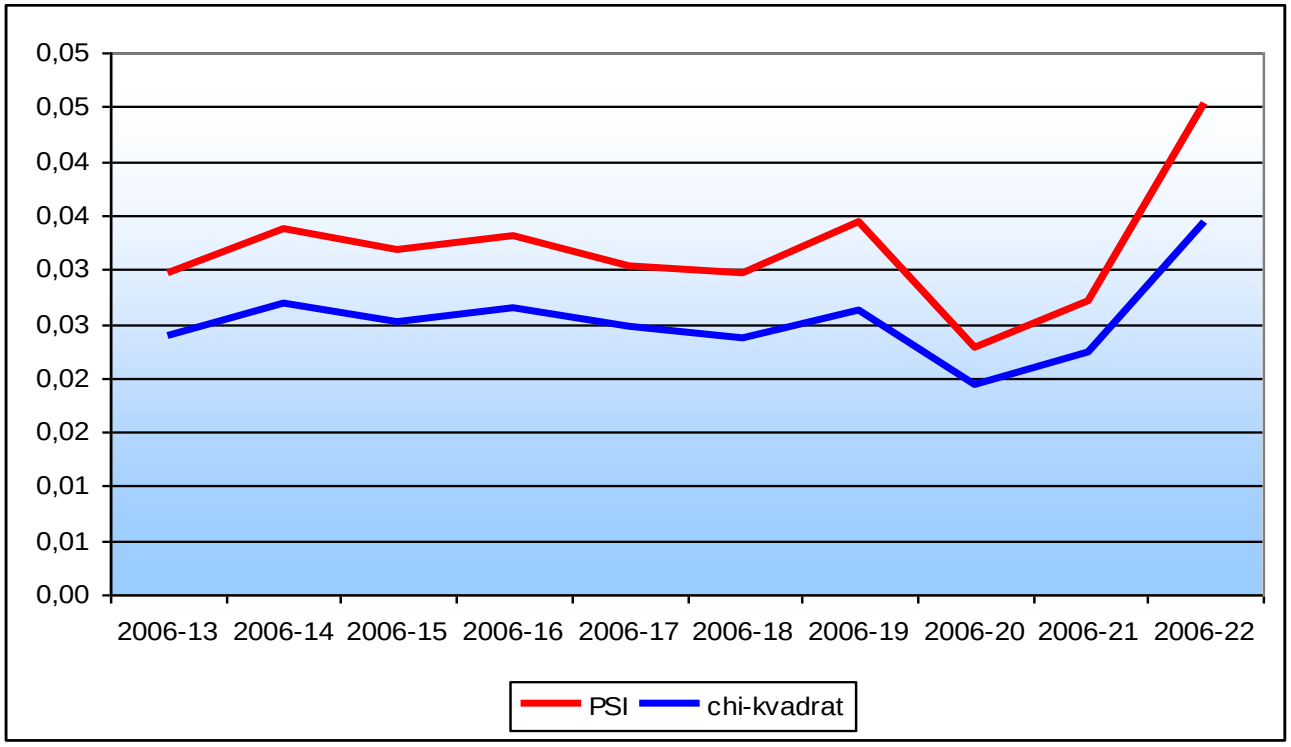
Monitoring scoringových modelů

$PSI \leq 0{,}1$ značí žádný nebo jen velmi malý rozdíl daných distribucí skóre.

$0{,}1 < PSI \leq 0{,}25$ znamená, že došlo k nějakému posunu distribuce, nicméně nikterak významnému.

$PSI > 0{,}25$ signalizuje významný posun v distribuci skóre, tj. zamítáme hypotézu o shodě daných distribucí.

Monitoring scoringových modelů

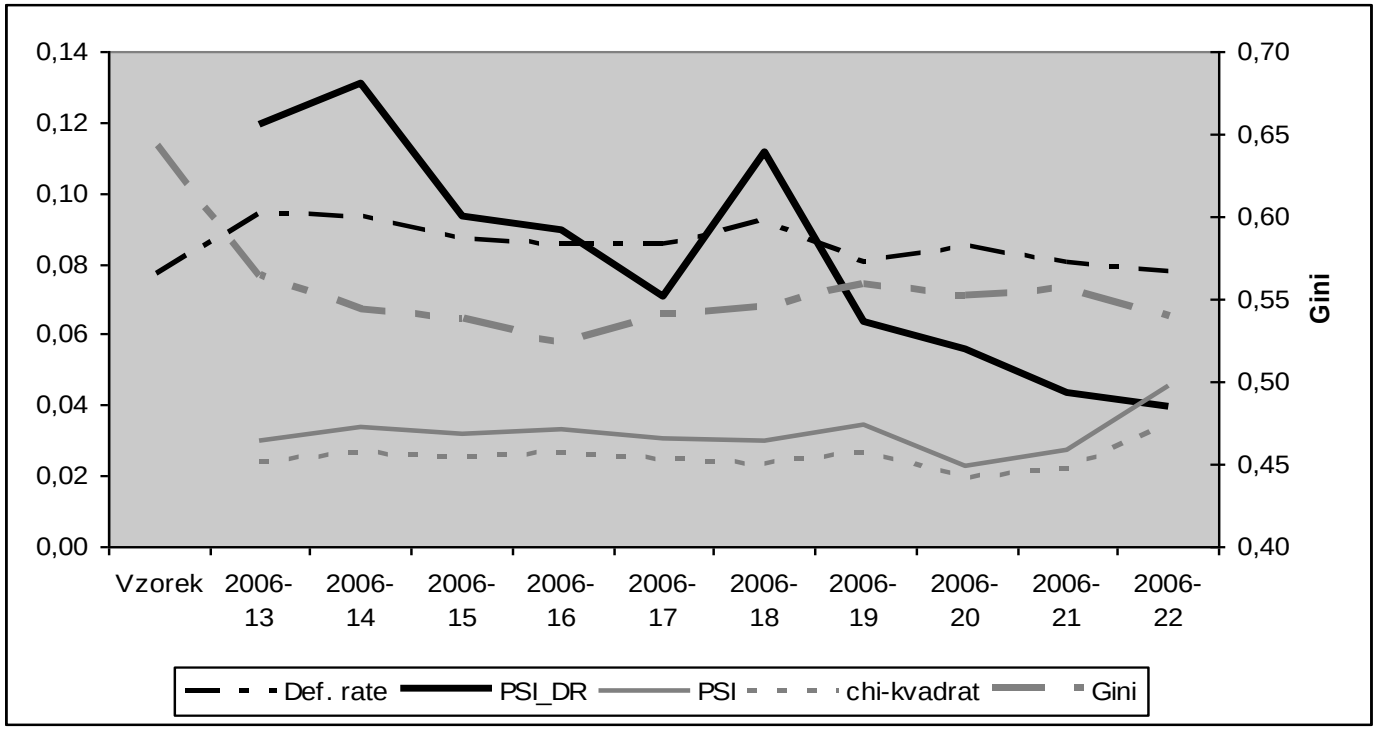


Monitoring scoringových modelů

$$PSI_{DR} = \sum_{i=1}^{r} (DR2_i - DR1_i) \ln\left(\frac{DR2_i}{DR1_i}\right)$$

	def_rate	Gini	PSI_DR	PSI	chi-kvardat
vzorek	7,69%	0,643			
200613	9,38%	0,564	0,120	0,030	0,024
200614	9,35%	0,542	0,131	0,034	0,027
200615	8,70%	0,537	0,093	0,032	0,025
200616	8,57%	0,523	0,089	0,033	0,026
200617	8,59%	0,540	0,071	0,030	0,025
200618	9,19%	0,544	0,111	0,030	0,024
200619	8,03%	0,558	0,063	0,034	0,026
200620	8,52%	0,552	0,055	0,023	0,019
200621	8,05%	0,555	0,043	0,027	0,022
200622	7,76%	0,539	0,039	0,045	0,034

Monitoring scoringových modelů



Champion-challenger (mistr – vyzyvatel)

- K rozšíření využití strategie champion-challenger došlo v devadesátých letech minulého století. Princip je velmi jednoduchý. Předpokládejme, že existuje nějaký způsob dělání něčeho (např. aktuálně používaný scoringový model pro schvalování/zamítání žádostí o úvěr). Tento způsob nazveme mistrem (champion). Nicméně existují další, jeden nebo více, alternativní způsoby jak dosáhnout téhož (nebo velmi podobného) cíle. Tyto nazveme vyzyvateli (challengers). Na náhodném vzorku otestujeme vyzyvatele a porovnáme s mistrem. To nám umožní nejen porovnat efektivnost vyzyvatelů a mistra, ale získáme možnost identifikovat existenci a rozsah vedlejších efektů. Výsledkem pak může být zjištění, že některý z vyzyvatelů je lepší než mistr a tento vyzyvatel se stane novým mistrem.

11. Reference

Literatura - knihy

- Anderson, R. (2007). *The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation*, Oxford: Oxford University Press.

- Giudici, P. (2003). *Applied Data Mining: statistical methods for business and industry*, Chichester : Wiley.

- Han, J., Kamber, M. (2006). *Data mining: Concepts and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann.

- Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer-Verlag.

- Hosmer, D. W., Lemeshow S. (2000). *Applied Logistic Regression, Textbook and Solutions Manual* , 2nd ed., New York: John Wiley and Sons.

Literatura - knihy

- Siddiqi, N. (2006). *Credit Risk Scorecards: developing and implementing intelligent credit scoring*, New Jersey: Wiley.

- Thomas, L.C. (2009). *Consumer Credit Models: Pricing, Profit, and Portfolio*, Oxford: Oxford University Press.

- Thomas, L.C., Edelman, D.B., Crook, J.N. (2002). *Credit Scoring and Its Applications*, Philadelphia: SIAM Monographs on Mathematical Modeling and Computation.

- Wilkie, A.D. (2004). Measures for comparing scoring systems, In: Thomas, L.C., Edelman, D.B., Crook, J.N. (Eds.), *Readings in Credit Scoring*. Oxford: Oxford University Press, pp. 51-62.

- Witten, I.H., Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, San Francisco: Morgen Kaufmann.

Literatura - časopisy

- Crook, J.N., Edelman, D.B., Thomas, L.C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research,* 183 (3), 1447-1465

- Hand, D.J. and Henley, W.E. (1997). Statistical Classification Methods in Consumer Credit Scoring: a review. *Journal. of the Royal Statistical Society*, Series A., 160,No.3, 523-541.

- Harrell, F.E., Lee, K.L. and Mark, D.B. (1996). Multivariate prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15, 361-387.

- Lilliefors, H.W. (1967). On the Komogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62, 399-402.

Literatura - časopisy

- Nelsen, R. B. (1998). Concordance and Gini's measure of association. *Journal of Nonparametric Statistics*, 9, Isssue 3, 227–238.

- Newson R. (2006). Confidence intervals for rank statistics: Somers' *D* and extensions. *The Stata Journal*, 6(3), 309-334.

- Somers R. H. (1962). A new asymmetric measure of association for ordinal variables. *American Sociological Review,* 27, 799-811.

- Thomas, L.C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting*, 16(2), 149-172 .

Literatura - web

- Coppock, D.S. (2002). *Why Lift?, DM Review Online*, www.dmreview.com/news/53291.html

- Xu, K. (2003). *How has the literature on Gini's index evolved in past 80 years?*, www.economics.dal.ca/RePEc/dal/wparch/howgini.pdf

- Xin Ming Tu, Wan Tang (2006). *Categorical Data Analysis*. http://www.urmc.rochester.edu/smd/biostat/people/faculty/TuSite/bst466/handouts.htm

- Jiawei Han and Micheline Kamber (2006). *Data Mining: Concepts and Techniques*. http://www.cs.illinois.edu/~hanj/bk2/

- Jens Peter Dittrich (2007). *Data warehousing*. http://www.dbis.ethz.ch/education/ss2007/07_dbs_datawh/Data_Mining.pdf

- Joe Carthy (2006). Data Warehousing. http://www.csi.ucd.ie/staff/jcarthy/home/DataMining/DM-Lecture02-01.ppt

- Jan Spousta (?). Přednášky k data miningu. [cit. 19.03.2009] http://samba.fsv.cuni.cz/~soukup

Další zajímavé zdroje informací

- http://www.cs.uiuc.edu/homes/hanj/
- http://www-users.cs.umn.edu/~kumar/
- http://www.kdnuggets.com/
- http://www.kdnuggets.com/datasets/competitions.html
- http://www.crc.man.ed.ac.uk/conference/
- http://www.crc.man.ed.ac.uk/conference/archive/
- http://www.kmining.com/info_conferences.html
- http://en.wikipedia.org/wiki/Data_mining
- http://cs.wikipedia.org/wiki/Data_mining
- http://en.wikipedia.org/wiki/Credit_scorecards

Užitečné zdroje dat

- http://archive.ics.uci.edu/ml/
- http://kdd.ics.uci.edu/

- http://sede.neurotech.com.br:443/PAKDD2009/
- http://www.dataminingbook.com/
- http://www.stat.uni-muenchen.de/service/datenarchiv/welcome_e.html
- http://www.kaggle.com/