

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Studijní program: Kvantitativní metody v ekonomice

Studijní obor: Statistika a ekonometrie

Autor bakalářské práce: Marie Ballarinová

Vedoucí bakalářské práce: doc. Ing. Dagmar Blatná, CSc.

NÁZEV BAKALÁŘSKÉ PRÁCE

**MOŽNOSTI VYUŽITÍ PROGRAMU R V REGRESNÍ A KONTINGENČNÍ
ANALÝZE**

školní rok 2010/2011

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze kterých jsem čerpala.

V Praze dne

.....
podpis

Poděkování

Velmi ráda bych poděkovala vedoucí bakalářské práce paní doc. Ing. Dagmaře Blatné, CSc. za cenné připomínky a vedení této bakalářské práce.

Děkuji také paní RNDr. Ivaně Malé, CSc. za získání dat z oblasti lékařství pro tuto práci.

Poděkování patří rovněž studentovy informatiky na VŠE panu Martinu Kovářovi, za cenné informace a připomínky k technické části týkající se programu R.

Děkuji také své rodině, přátelům a spolužákům při jejich podpoře.

Abstrakt

Cílem této bakalářské práce je seznámit se základy statistického programu R a jeho aplikaci na regresní a kontingenční analýzu. Práce se zaměřuje na základní uživatelské schopnosti při zpracování dat v programu R. Program je hodnocen z mnoha hledisek (například výhody programu, zpracování dat v R, práce s kontingenčními tabulkami v R, grafy v R, import a export dat atd.). Obě analýzy jsou v práci použity pro názornou ukázkou funkcí a syntaxí pro tyto oblasti statistiky.

Přínos práce je rozšíření povědomí o velké využitelnosti programu a vyzdvižení různých funkcí, jak u obou analýz, tak v základech při zpracování statistických dat. Hlavní součástí této práce jsou příkazy, které se mohou přímo aplikovat do příkazového řádku v R v praxi.

Struktura je rozdělena do pěti hlavních kapitol. První a druhá kapitola obsahuje základní popis regresní a kontingenční analýzy. Třetí kapitola seznamuje se syntaxí, s příkazy a s technickou stránkou programu R. Ve čtvrté a páté kapitole je aplikace obou analýz v programu R. U regresní analýzy je použita jednoduchá a vícenásobná regresní analýza. Hlavní náplní kontingenční analýzy je pracování s daty pomocí tabulek v programu R.

Abstract

The aim of this bachelor thesis is to introduce the basics of the statistical program R and its application to regression and contingency analysis. The work focuses on basic user skills with data processing in the program R. The program is evaluated from many perspectives (eg, program benefits, data processing in R, working with PivotTables in R, graphs in R, import and export data, etc.). Both analyses in this work are used for demonstration of commands and syntax for these statistics.

The contribution of this work is to raise awareness about the great usefulness of the program and highlight the different functions both the analysis and the basis for statistical data processing. The main part of this work is focused on commands that can be applied directly to the command line in R in practice.

The structure is divided into five main chapters. The first and second chapters provide a basic description of regression analysis and contingency. The third chapter introduces the syntax, commands and technical aspects of the program R. In the fourth and fifth chapter is application of both analyses in the program R. In the regression analysis is used simple and multiple regression analysis. The main scope of contingency analysis is working with data using tables in the program R.

Obsah

Úvod	6
1. Základní pojmy Regresní analýzy	7
1.1. Cíle regresní analýzy	7
1.2. Typy Regresních Funkcí	8
1.2.1. Model přímkové regrese a roviny	9
1.2.2. Model parabolické regrese	9
1.2.3. Model polynomické regrese p – tého stupně	10
1.2.4. Model hyperbolické regrese	10
1.2.5. Model exponenciální regrese	10
1.2.6. Model Logaritmické regrese	10
1.3. Předpoklady Regresní analýzy	10
1.4. Způsob výpočtu regresních koeficientů	11
1.4.1. Maticové vyjádření MNČ	12
2. Kontingenční analýza	13
2.1. Typy kategoriálních proměnných	13
2.2. Analýza četností kategoriálních proměnných	13
2.2.1. Nominální proměnná	14
2.2.2. Ordinální proměnná	14
2.2.3. Kvantitativní proměnná	15
2.3. Analýza kontingenčních tabulek	16
3. Program „R“	17
3.1. Úvod „R“	17
3.2. Výhody „R“	17
3.2.1. Popis R	18
3.3. Základy R	19
3.3.1. Zadávání dat pomocí „c“	19
3.3.2. Použití funkcí	19
3.4. Grafy v R	21
3.4.1. Sloupcový graf	21
3.4.2. Kruhový graf (koláč)	24
3.4.3. Boxplot	25
3.4.4. Histogram	26
3.5. Správa pomocí balíčků v R	27

3.6.	Export a Import dat	28
3.6.1.	Import dat	28
3.6.2.	Export dat	30
4.	Regresní analýza v programu R	32
4.1.	Výpočet Jednoduché lineární regrese v R	32
4.2.	Výpočet vícenásobné lineární regrese v R	36
4.2.1.	Výpočet v EXCELU pomocí MNČ.....	37
4.2.2.	Výpočet regrese v R pomocí příkazu „lm()“	38
4.2.3.	Grafické zobrazení příkladu pomocí roviny v R	40
4.2.4.	Vypočet regresních koeficientů pomocí vzorců v R	41
5.	Zpracování dat a Kontingenční analýza v programu R	45
5.1.	Kategoriální data	45
5.2.	Numerická data	46
5.3.	Práce s kontingenčními tabulkami v R.....	47
5.3.1.	Dvojměrná kontingenční tabulka	47
5.3.2.	Vícerozměrná kontingenční tabulka	50
	Závěr.....	54
	Literatura.....	55
	Seznam obrázků	56

Úvod

Regresní a kontingenční analýza jsou nejčastěji používané statistické metody. Každá z uvedených metod zkoumá statistická data, z vlastního úhlu pohledu. Regresní analýza zkoumá jednostrannou závislost vysvětlované proměnné na jedné nebo více vysvětlujících číselných proměnných. Kontingenční analýza zkoumá vztah dvou a více kategoriálních znaků. Základním nástrojem kontingenční analýzy je tzv. kontingenční tabulka, která přehledně zobrazí statistická data. Obě analýzy, můžeme zpracovat v různých statistických softwarech. Nejznámější statistické softwary, které se používají na VŠE, jsou SAS, SPSS a Statgraphic. Softwary, které řadíme mezi ekonometrické, mohou také sloužit ke zpracování statistických dat. Mezi ekonometrické softwary, které se na VŠE používají je MPL a Lingo. Hlavní součástí této práce, je nastínit základy aplikace statistického programu R na příkladě regresní a kontingenční analýzy.

Toto téma jsem si vybrala proto, abych se více seznámila se statistickým programem R, který není do běžné výuky na VŠE zařazen a mohla tak posoudit jeho ovladatelnost.

Práce má tři hlavní cíle.

- 1 cíl: Popis základů regresní a kontingenční analýzy
- 2 cíl: Popis základů programu R
- 3 cíl: Aplikace regresní a kontingenční analýzy ve statistickém programu R

Prvním cílem bude nastínit základní prvky regresní a kontingenční analýzy. Úkolem bude popsat obě analýzy pouze okrajově, neboť k hlubšímu studiu obou analýz, slouží odborné učebnice a skripta. Druhý cíl, který je jádrem práce, bude seznámení se statistickým programem R. Nastínit prostředí programu, které není běžné pro výše uvedené statistické programy na VŠE. Součástí druhého cíle bude seznámení se základní manipulací v programu R, při použití ve statistice a jeho technické správě. Třetí a poslední cíl bude názorně aplikovat regresní a kontingenční analýzu v programu R.

Práce je rozdělena do pěti kapitol a několika sub-kapitol. První kapitola obsahuje základní prvky regresní analýzy. V sub-kapitolách jsou uvedeny typy regresních analýz a jejich výpočtu.

Druhá kapitola se bude zabývat základy kontingenční analýzy - typy proměnných a jejich vlastnostmi.

Třetí kapitola se zaměřuje na základy použití programu R ve statistice a jeho technickou správu. Jednoduchým způsobem názorně vysvětluje prostředí, na kterém program pracuje, a aplikaci syntaxí a příkazů v praxi.

Aplikace regresní analýzy v programu R je obsahem čtvrté kapitoly. Je zde uveden výpočet jak jednoduché regresní analýzy, tak vícenásobné regrese a jejich grafické zobrazení.

Poslední pátá kapitola obsahuje aplikaci kontingenční analýzy v programu R.

1. Základní pojmy Regresní analýzy

V regresní analýze sledujeme kauzální (příčinou) závislost mezi dvěma i více jevy. Kauzální znamená případ, kdy existence (nastoupení, výskyt) jednoho jevu zapříčiní (vyvolá) existenci jiného jevu. Nastávají také situace, kdy „výskyt určitého jevu souvisí s výskytem jiných jevů, kdy existence skupiny jevů má za následek nastoupení jiných jevů nebo (nejčastěji) výskyt některých jevů určitým způsobem souvisí s výskytem jiných jevů“.¹

Regresní analýza slouží pro hodnocení jedné vysvětlované náhodné veličiny a jedné nebo několika vysvětlujících veličin. „Základním vodítkem regresní analýzy je potřeba nepřímo působit na vysvětlovanou veličinou volbou, ovlivňováním nebo aspoň snadnějším odhadem hodnot vysvětlujících proměnných. K naplnění této potřeby však musí mezi vysvětlovanou proměnnou a vysvětlujícími proměnnými existovat kvantifikovatelný vztah, přesněji musí existovat matematicky popsatelná závislost vysvětlované proměnné na vysvětlujících proměnných. Zobrazení této závislosti je regresní model, jehož rozhodující součástí je regresní funkce“² Regresní analýza se zabývá jednostrannými závislostmi. Zkoumání vzájemných závislostí mezi proměnnými, se používá korelační analýza.

1.1. Cíle regresní analýzy

Hlavním úkolem regresní analýzy je najít vztah mezi statistickými znaky prostřednictvím statistických údajů. Statistický soubor (údaje) můžeme získat různým pozorováním.

Možnosti při získání údajů:

- a) Údaje jsme získali pozorováním n statistických jednotek, přičemž statistický soubor byl prostorově, časově i věcně vymezen (domácnosti žijící na území ČR, společně hospodařících a žijících)
- b) Údaje jsme získali pozorováním určité statistické jednotky v n časových okamžicích či intervalech (kurs akcie dané společnosti)
- c) Pozorování vznikla n -násobným opakováním určitého pokusu, prováděného za stejných nebo přibližně stejných podmínek (technologické a jakostní vlastnosti)³

Regresní analýza hledá nejlepší řešení vypočítané (empirické) funkce k hypotetické regresní funkci. S hledáním řešení regresní analýzy souvisí další dílčí úkony, které jsou nezbytné pro splnění jejího cíle. Do tak zvaných dílčích úkonů můžeme zahrnout matematickou formulaci regresní funkce, což je většinou těžký úkol, shromáždění předpokladů o souhrnných působení neuvažovaných statistických znaků a zhodnocení kvality regresní empirické funkce. Zhodnocení regresní funkce bereme s ohledem na splnění cílů statistického zjišťování.

Mezi vysvětlující a vysvětlovanou proměnnou regresní funkce, existuje síla (intenzita, těsnost) závislosti. Intenzitou závislosti se zabývá korelační analýza. Níže uvedené příklady ukazují různé možnosti síly závislosti.

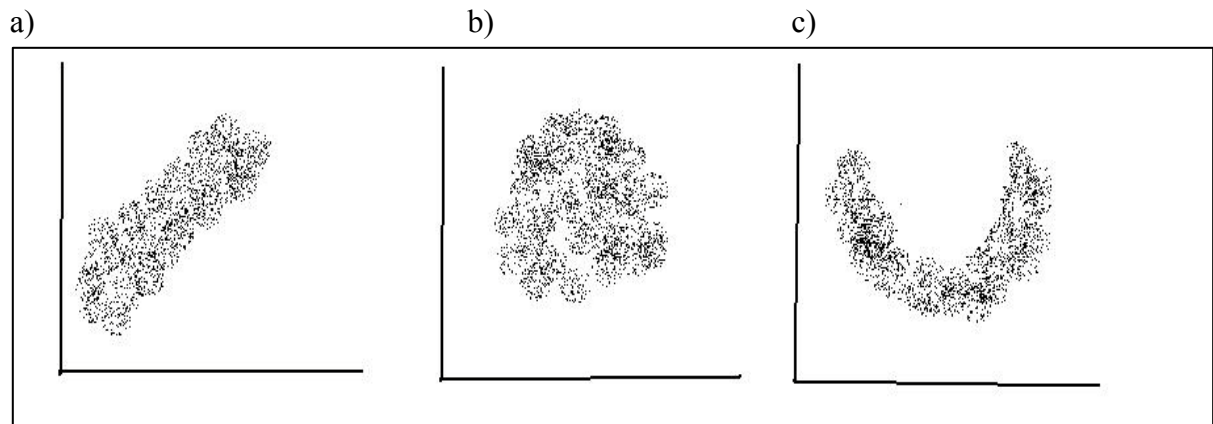
¹ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7. Str.167

² HEBÁK, Petr. *Regrese : I.část*. Praha : Ediční oddělení VŠE Praha, 1998. 138 s. ISBN 80-7079-909-9

³ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7

Příklad a) a b) mají lineární závislost neboli lineární průběh, příklad c) vykazuje nelineární průběh.

Různé typy statistických závislostí:



Obrázek 1: Typy statistických závislostí⁴

Abychom se dostali k cíli výše dané otázky, hodnocení statistických závislostí se týká průběhu závislostí u regrese a intenzity u korelace. Matematickou funkci pro regresní analýzu můžeme hledat v přírodních vědách například v matematice. Známe několik matematických funkcí, kde můžeme aplikovat regrese. Funkce lineární v parametrech a nelineární v parametrech.

1.2. Typy Regresních Funkcí

Hledání závislostí mezi empirickými hodnotami x_i a y_i a průběhu závislosti, a při známých hodnotách těchto proměnných, je základní bod dobře zvolit regresní funkci, aby dobře vystihovala danou závislost. Při odhadu typu regresní funkce musí být dobře specifikována „věcně ekonomická kritéria tj. regresní funkce by měla být zvolena na základě věcného rozboru analýzy vztahů mezi veličinami, přičemž by základem rozhodnutí měla být existující ekonomická teorie.“⁵ V případě že daná situace již v minulosti nastala, je možné použít určitý typ dříve použité regresní funkce, a následně ověřit, zda nedošlo ke změně podmínek nebo zkoumaného jevu.

Pro nejednoznačné určení typu regresní funkce podle věcně ekonomických kritérií, je vhodné použít empirický (induktivní) způsob volby. Můžeme si představit grafickou metodu pomocí bodového diagramu, kde x a y znázorňují jeden bod daného grafu. Podle průběhu funkce a následně znázorněného typu grafu, se snažíme rozhodnout který typ regresní funkce je vhodný (přímka, parabola apod.). Zde věcně ekonomická kritéria pomáhají vybrat alternativní typ regresní funkce, kterou si pak vybíráme na základě empirických údajů.

Z výše uvedených poznatků se rozlišuje regresní funkce na teoretickou (hypotetickou) a empirickou (výběrovou) regresní funkcí. Teoretická regresní funkce je nezměřitelná (nepozorovatelná) a empirickou vypočítáme většinou pomocí empirických údajů.

⁴ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7

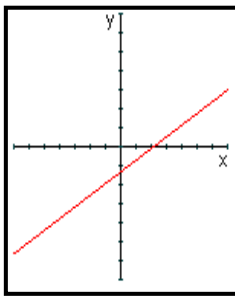
⁵ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7

„Empirickou regresní funkci můžeme považovat za odhad teoretické regresní funkce.“⁶ Pro každé pozorování platí rovnice: $y_i = Y_i + \epsilon_i$. Kde Y_i je i – tá teoretická regresní funkce, y_i je i - tá hodnota vysvětlované proměnné y , a hodnota ϵ_i znázorňuje odchylku mezi y_i a Y_i . Tato odchylka nám znázorňuje vliv jiné proměnné na proměnnou y_i , než kterou uvažujeme. To znamená, že na empirické pozorování mají vliv také náhodné chyby. Odchylka ϵ_i má nulovou střední hodnotu a systematicky nezkrsluje. Parametry regresní funkce β_0 a $\beta_1 \dots \beta_p$ neboli neznámé konstanty můžeme zapsat jako $Y=f(x; \beta_0, \beta_1 \dots \beta_p)$. Cíl ke kterému směřujeme je odhadnout tyto parametry, a empirickou regresní funkci pak zapisujeme jako $\hat{y}_i = f(x; b_0, b_1, \dots b_p)$. Kde \hat{y}_i je odhad teoretické hodnoty Y odpovídající hodnotě x_i .

V případě že ϵ_i nebude existovat (pro každé i), znázorňuje funkce Y pouze předpis. Předpis, který přiřazuje hodnotě proměnné x hodnotu proměnné y . Tato situace se v praxi nazývá pevná závislost, kdy pravděpodobnost teoretické regresní funkce je rovná jedné, a jednalo by se o model deterministický. Modely, kde hodnotu ϵ_i můžeme zaznamenat, jsou stochastické.

„Stanovení empirické regresní funkce v podstatě znamená, že každou empirickou hodnotu y_i nahradíme určitou „vyrovnanou „ hodnotou \hat{y}_i , která bude „ležet“ na zvolené regresní čáře.“⁷

1.2.1. Model přímkové regrese a roviny



Model regresní přímky a regresní roviny, jsou případy, kdy parametry vysvětlující proměnné jsou lineární. Jsou to nejjednodušší a nejčastější typy regresní funkce.

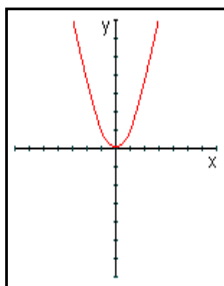
V modelu regresní roviny vystupují dvě vysvětlující proměnné, ale v modelu regresní přímky pouze jedna vysvětlující proměnná.

$$Y = \beta_0 + \beta_1 x$$

$$Y = \beta_0 + \beta_1 x + \beta_2 x$$

Obrázek 2: Přímková funkce ⁸

1.2.2. Model parabolické regrese



Model parabolické regrese je také často používaný k hodnocení závislosti v ekonomické oblasti. Tento model je lineární z pohledu všech parametrů, ale nelineární z pohledu vysvětlujících proměnných.

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2$$

Obrázek 3: Parabolická funkce ⁹

⁶ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7

⁷ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7

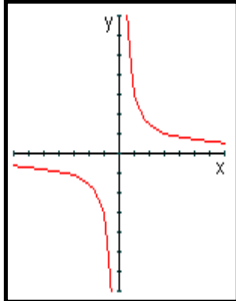
⁸ <http://www.esphere.cz/kostka/Matematika/Funkce/specifikace.htm>

⁹ <http://www.esphere.cz/kostka/Matematika/Funkce/specifikace.htm>

1.2.3. Model polynomické regrese p – tého stupně

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_p x^p$$

1.2.4. Model hyperbolické regrese

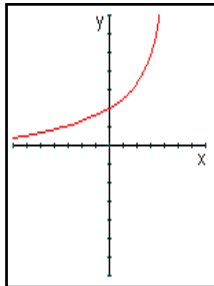


Model hyperbolické regrese je dalším typem, který je často používán k popisu závislosti v ekonomické oblasti. Je lineární v parametrech, a tudíž na ní můžeme aplikovat metodu nejmenších čtverců.

$$Y = \beta_0 + \beta_1/x$$

Obrázek 4: Hyperbolická funkce ¹⁰

1.2.5. Model exponenciální regrese

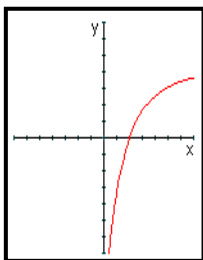


Exponenciální regrese není lineární ve svých parametrech a tudíž metodu nejmenších čtverců, pro odhad parametrů, nemůžeme použít. Použití metody nejmenších čtverců na nelineární model vede k soustavě nelineárních rovnic.

$$Y = \beta_0 \beta_1^x$$

Obrázek 5: Exponenciální funkce ¹¹

1.2.6. Model Logaritmické regrese



Logaritmická regrese zastupuje lineární model v parametrech.

$$Y = \beta_0 + \beta_1 \log x$$

Obrázek 6: Logaritmická funkce ¹²

1.3. Předpoklady Regresní analýzy

Pro existenci jednoduché lineární regrese musí být splněny základní statistické požadavky, které jsou důležité při výpočtu.

¹⁰ <http://www.esphere.cz/kostka/Matematika/Funkce/specifikace.htm>

¹¹ <http://www.esphere.cz/kostka/Matematika/Funkce/specifikace.htm>

¹² <http://www.esphere.cz/kostka/Matematika/Funkce/specifikace.htm>

- 1) Parametry modelu β_i mohou nabývat libovolných hodnot.
- 2) Normalita náhodné složky (rezidui)
- 3) Nulová střední hodnota náhodné složky
- 4) Homoskedasticita náhodné složky
- 5) LRM je lineární v parametrech
- 6) Nulová kovariance náhodných složek $Cov(e_i, e_j) = 0$ pro každé $i \neq j$, kde $i, j = 1, 2, \dots, n$.

Jednoduchý lineární regresní model

$$Y_i = \beta_0 + \beta_1 x_i + e_i$$

Kde β_0 a β_1 jsou tzv. parametry modelu a e_i reziduum - náhodné složky.

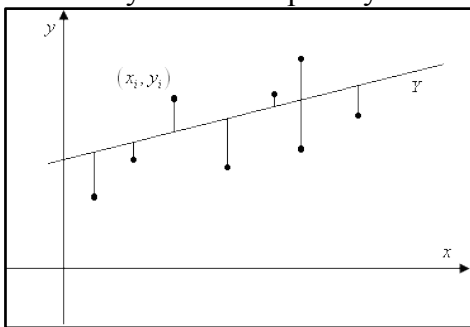
Hodnoty β_0 a β_1 jsou neznámé, a odhadujeme je z patřičných dat. Hodnota e_i je množství vypořizovaných odchylek z lineární přímky modelu. K tomu abychom mohli odhadnout β_0 a β_1 je použita metoda nejmenších čtverců – MNC.

1.4. Způsob výpočtu regresních koeficientů

Při řešení Regresního modelu se nabízí řada metod. Důležitou metodou pro řešení regresního modelu je „metoda nejmenších čtverců.“ Je to matematicko-statistická metoda. Napomáhá nám nalézt danou aproximační funkci pro empiricky zjištěné hodnoty.

Při stanovení regresní funkce, je důležité najít takovou, která vystihuje nejvíce danou závislost mezi daty. Pro stanovení empirické regresní funkce, se nahradí každá empirická hodnota y_i vyrovnanou hodnotou \hat{y}_i , která bude ležet na dané regresní přímce.

Ukázka vyrovnání empirických hodnot hodnotami teoretickými.



Obrázek 7: Ukázka Vyrovnání hodnot¹³

První podmínka pro výše uvedený problém je $\sum_{i=1}^n (y_i - \hat{y}_i) = \sum_{i=1}^n e_i = 0$

Znamená, aby součet e_i (reziduum) se rovnal nule. Reziduum je odhad náhodné složky ε .

Pro kompletní doplnění podmínek pro MNC, je nutná další důležitá podmínka. Podmínka, aby součet čtverců chyb ε_i byl minimální: $Q = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \varepsilon_i^2 \longrightarrow \min$.

Nejběžnějším a nejjednodušším typem regresní funkce, je model přímkové regrese. Na přímkové regresi, si ukážeme další kroky odhadu parametru regresní funkce.

Při dosazení přímkové regresní funkce do výše uvedené podmínky bude vypadat následovně

$$Q = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 = \sum_{i=1}^n \varepsilon_i^2 \longrightarrow \min.$$

Pro výpočet Q je nutné vypočítat jejich první parciální derivace parametru β_1 a β_0 a položit je nule. Po dosazení jejich odhadů za parametry a po derivaci, dostaneme následující dvě rovnice.¹⁴

¹³ <http://homen.vsb.cz/~oti73/cdpast1/KAP09/KAP09.HTM>

$$2\sum_{i=1}^n (y_i - b_0 - b_1 x_i)(-1) = 0$$

$$2\sum_{i=1}^n (y_i - b_0 - b_1 x_i)(-x_i) = 0$$

Úpravami dostaneme následující dva vzorečky.

$$b_0 = \frac{\sum y_i \sum x_i^2 - \sum x_i \sum y_i x_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (1.1)$$

$$b_1 = \frac{n \sum y_i x_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (1.2)$$

1.4.1. Maticové vyjádření MNČ

Pro výpočet regresních koeficientů s větším počtem proměnných, je nevhodnější využít maticové vyjádření výše uvedených vzorců.

$$y = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} \quad (nx1) \quad b = \begin{bmatrix} b_1 \\ \dots \\ b_n \end{bmatrix} \quad \text{Vektor odhadovaných parametrů}$$

$$F = \begin{bmatrix} 1 & f_{11} & f_{1p} \\ 1 & f_{21} & f_{2p} \\ \dots & \dots & \dots \\ 1 & f_{n1} & f_{np} \end{bmatrix} \quad \text{matice funkcí (nx(p+1))}$$

$F'F b = F'y$ maticové vyjádření rovnice:

$$\sum_{i=1}^n y_i f_p(x_i) = b_0 \sum_{i=1}^n f_p(x_i) + b_1 \sum_{i=1}^n f_1(x_i) f_p(x_i)$$

F' transportní matice k matici F

$F'F$ Předpoklad, že k této matici existuje matice inverzní

Dále platí $b = (F'F)^{-1}F'y$

Pro aplikaci například na přímkovou regresi, vypadá maticové vyjádření MNČ následovně.

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \quad F = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} \quad F'F = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix} * \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}$$

$$F'y = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} \sum y_i & x_i \end{bmatrix}$$

$$F'F b = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} nb_0 & b_1 \sum x_i \\ b_0 \sum x_i & b_1 \sum x_i^2 \end{bmatrix}$$

¹⁴ SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7

2. Kontingenční analýza

S kategoriálními daty se často setkáváme v sociální oblasti. Hlavní motiv zkoumání jsou znaky, které nejsou přímo měřitelné. Pro získání hodnot znaků je běžné dotazování v dotazníku. Respondent odpovídá na problém svou odpovědí, nebo si vybírá z nabídky. Odpovědím, kterým se přiřazují slovní nebo číselné kódy, jsou hodnoty znaku neboli kategorie.

- Národnost (nelze hodnoty uspořádat)
- Úroveň vzdělání (lze hodnoty uspořádat)
- Počet členů v domácnosti (hodnoty lze mezi sebou uspořádat a vypočítat mezi nimi rozdíl)

Uvedené příklady ukazují vztahy mezi kategoriemi.¹⁵

Kontingenční analýza se zabývá zkoumáním vztahů mezi dvěma a více kategoriálními proměnnými.

2.1. Typy kategoriálních proměnných

Typ kategoriální proměnné můžeme rozlišovat podle různých kritérií. Kategorie, které jsou popsány výše, tvoří škálu hodnot. Základní typy škály jsou nominální, ordinální, intervalové a poměrové. Podle daných typů škál rozlišujeme kategoriální proměnné.

- Nominální proměnná* (Hodnoty, které nabývá nominální proměnná jsou různé. Nedokážeme určit jejich pořadí, jen je považujeme za různé)
- Ordinální proměnné* (Jsou to proměnné, u kterých hodnoty můžeme seřadit podle jejich důležitosti, ale nedokážeme vypočítat jejich rozdíl.)
- Kvantitativní proměnné* (Tyto proměnné se dále člení na intervalové a poměrové. Hodnoty, které nabývají kvantitativní proměnné, můžeme jak seřadit podle velikosti, tak vypočítat, o kolik se mezi sebou liší.)

2.2. Analýza četností kategoriálních proměnných

Analýzu četností kategoriálních proměnných provádíme pomocí základních charakteristik. Do základních charakteristik zahrnujeme míru polohy, míru variability a koncentrace. Důležitou součástí zkoumání je znalost rozdělení četností. Pomocí rozdělení četností zjistíme, kolikrát se v souboru dat jednotlivé varianty objevují. Základní rozdělení četností je *absolutní* a *relativní* četnost. Absolutní četnost sleduje celkový počet N sledovaných znaků. Každý n_i sledovaný znak je součástí N souboru. Sečtení každého sledovaného znaku n_i dává soubor N . Relativní četnost je podíl každého znaku n_i na celkovém souboru N , neboli

$$p_i = \frac{n_i}{N} \quad (2.1.)$$

Pro výpočet u ordinálních proměnných a číselných musíme hodnoty logicky uspořádat.

¹⁵ ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1. Str. 18

2.2.1. Nominální proměnná

Tyto proměnné nemůžeme vyjádřit ani číslem ani určit jejich pořadí. Zahrnujeme zde například typ profese, druh spotřebitelského zboží atd..

A. Míra polohy

Při proměnné, která je nominální, je míra polohy dána modální kategorií. Modální kategorie je tzv. kategorie s největší četností. Modální kategorie se označují M_o . Její relativní četnost je p_i a absolutní četnost N_i . Modální kategorie se může vyskytovat u sledované hodnoty jednou nebo i vícekrát.¹⁶

B. Míra variability

Míru variability můžeme zjistit několika způsoby. Jednou z nich je pomocí variačního poměru v , který se vypočítá následovně:

$$v = 1 - p_{m_o} \quad (2.2)$$

Další přesnější možností je *nominální rozptyl* neboli *nomvar* (2.3) a *normalizovaný nominální rozptyl* (2.4).

$$\text{nomvar} = 1 - \sum_{i=1}^K \left(\frac{N_i}{N} \right)^2 \quad (2.3)$$

$$\text{normalizovaný nominální rozptyl} = K * \text{nomvar} / (K-1) \quad (2.4)$$

Nomvar vyjadřuje relativní počet každé dvojice, které nepatří do stejné kategorie. Hodnoty, které nabývá nomvar, udává interval $\langle 0; k - 1 \rangle$. Normalizovaný nominální rozptyl nabývá hodnot z intervalu $\langle 0; 1 \rangle$.

Nesmíme zapomenout ani na poslední míru variability a to je entropii H . Která je dána vzorcem (2.5). U entropie také nalézáme normalizovanou entropii (2.6).

$$H = - \sum_{i=1}^K p_i \ln p_i \quad \in \langle 0; \ln K \rangle \quad (2.5)$$

$$\text{Normalizovaná entropie} = H / \ln K \quad \in \langle 0; 1 \rangle \quad (2.6)$$

2.2.2. Ordinální proměnná

U ordinální proměnné můžeme logicky uspořádat, ale nelze přesně říct, o kolik se dvě varianty liší. Zahrnujeme zde například stupeň vzdělání (základní, střední, vysoká) nebo stupeň dosažené šarže u vojenských důstojníků atd.

¹⁶ ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1. Str 19

A. Míra polohy

K ordinální proměnné a jejím mírám polohy patří jak *modální kategorie (Mo)* tak *mediánová kategorie (Me)*. U mediánové kategorie musí být kumulativní četnost menší nebo rovna než 0,5.¹⁷

B. Míra variability

K určení variability ordinální proměnné slouží *ordinální rozptyl* neboli *dorvar* (2.7) a jeho normalizovaná podoba (2.8).

$$dorvar = 2 \sum_{i=1}^{K-1} P_i(1 - P_i) \quad \epsilon \langle 0; (K - 1)/2 \rangle \quad (2.7)$$

$$\text{norm.ordinální rozptyl} = 2 * dorvar / (K-1) \quad \epsilon \langle 0; 1 \rangle \quad (2.8)$$

2.2.3. Kvantitativní proměnná

Kvantitativní proměnné můžeme jak uspořádat, tak vypočítat mezi nimi rozdíl. Jde o proměnné v podobě čísla.

A. Míra polohy

Pro kvantitativní proměnné platí charakteristiky pro ordinální proměnnou a navíc i aritmetický průměr (2.9).

$$\bar{X} = \sum_{i=1}^K x_i p_i \quad (2.9)$$

B. Míra variability

Výpočet míry variability pro kvantitativní proměnnou je jednoduchý. Zahrnujeme zde rozptyl (2.10), směrodatnou odchylku (2.11) a variační koeficient (2.12).

$$\sigma^2 = \sum_{i=1}^K (x_i - \bar{X})^2 p_i \quad (2.10)$$

$$\sigma = \sqrt{\sigma^2} \quad (2.11)$$

$$V = \frac{\sigma}{\bar{X}} \quad (2.12)$$

¹⁷ ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1. Str. 20

2.3. Analýza kontingenčních tabulek

Kontingenční tabulky slouží pro analýzu dvou a více kategoriálních dat. Jsou základem pro testování závislosti a výpočty měr intenzity závislosti. Jsou důležitou součástí pro analýzu kategoriálních dat v statistických softwarech. Sloupce kontingenční tabulky odpovídají statistickým znakům a řádky statistickým jednotkám. Můžeme mít kontingenční tabulku s absolutními četnostmi nebo relativními četnostmi.¹⁸

A. Kontingenční tabulka s absolutními četnostmi

	Znak Y			
	1.kategorie	..	j-tá kategorie	celkem
Znak X 1.kategorie	n_{11}		n_{1j}	n_{1+}
i-tá kategorie	n_{i1}		n_{ij}	n_{i+}
r-tá kategorie	n_{r1}		n_{rj}	n_{r+}
celkem	n_{+1}		n_{+j}	n

19

B. Kontingenční tabulka s relativními četnostmi

	Znak Y			
	1.kategorie	..	j-tá kategorie	celkem
Znak X 1.kategorie	p_{11}		p_{1j}	p_{1+}
i-tá kategorie	p_{i1}		p_{ij}	p_{i+}
r-tá kategorie	p_{r1}		p_{rj}	p_{r+}
celkem	n_{+1}		p_{+j}	1

20

¹⁸ ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1. Str.33

¹⁹ ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1. Str. 33

²⁰ ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1. Str. 34

3. Program „R“

3.1. Úvod „R“

Program „R“ je programovací jazyk a softwarové prostředí pro statistické výpočty a grafiku. „R“ je realizace programovacího jazyka „S“. Program „R“ je považován za jiné provedení jazyka „S“. Mezi programovacími jazyky „S“ a „R“ můžeme najít některé důležité rozdíly. Některé kódy psané pod „S“ fungují také v nezměněném stavu pod „R“. Byl vytvořen na univerzitě „of Auckland“ na Novém Zélandu místními profesory „Ross Ihaka a Robert Gentleman“, v roce 1995. Pojmenování programu vyplynulo z prvních písmen jmen autorů. Tento program se stal také součástí projektu „GNU“²¹. Zdrojový kód „R“ je volně k dispozici pod „GNU“ a pre-binární verze jsou stanoveny pro různé operační systémy. „R“ používá rozhraní příkazového řádku. (<http://www.r-project.org/>)

3.2. Výhody „R“

Tento program nabízí velkou škálu statistické (lineární a nelineární modelování, statistické testy, klasifikace, shlukování atd.) a grafické techniky. „R“ je velice rozšiřitelný program. Základní silnou stránkou programu „R“ je snadnost a rychlost reagovat na daný příkaz. Velmi snadno generuje různé grafy, včetně matematických symbolů a vzorců. Uživatel si v „R“ vždy zachovává plnou kontrolu nad danou problematikou. Obsahuje následující výhody, které umožní uživateli dobře zpracovat data.

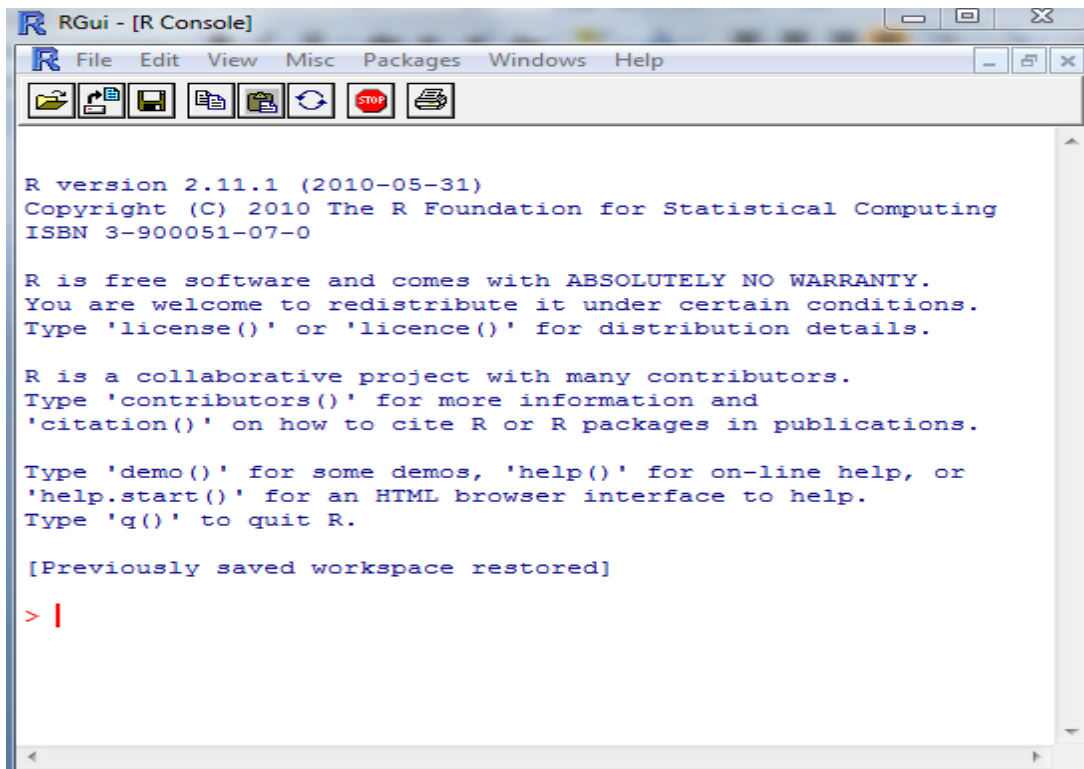
Výhody:

- Efektivní zpracování dat a skladovacích zařízení
- Sada operátorů pro výpočty na pole (zejména matice)
- Velký, koherentní, ucelený soubor dílčích nástrojů pro analýzu dat
- Grafické zařízení pro analýzu dat a zobrazení buď on-screen nebo tištěně
- Dobře rozvinutý, jednoduchý a efektivní programovací jazyk, který obsahuje uživatelem definované funkce a vstupní a výstupní zařízení

„R“ podobně jako „S“ je navržen, aby umožnil uživatelům přidávat další funkce tím, že definuje nové funkce. Pokročilí uživatelé mohou psát kód také v jazyce „C“ přímo pro manipulaci s „R“. Zajímavou výhodou „R“ je snadné rozšíření přes balíčky. Existuje zhruba osm balíčků dodávaných přímou distribucí programu R a další, které pokrývají velkou škálu moderní statistiky, jsou k dispozici na „CRAN“. (<http://www.r-project.org/>)

²¹ General Public License

3.2.1. Popis R



Obrázek 8: Ukázka programu R

Při otevření R se automaticky načte úvodní text, který je modře rozlišen. Text obsahuje základní informace o programu. Informuje například, že R je software, který je volně dostupný. Tyto úvodní informace se zobrazují vždy. Na horní liště jsou zobrazeny příkazy pomocí ikon, které se nacházejí také v menu nad okýnky. Rozhraní programu R disponuje pouze základním menu, složitější operace je nutné zadávat přes psané příkazy. Uživatel má možnost v R pracovat s více okny najednou.

Menu obsahuje:

File = Source R code, New script, Open script, Display file(s), Load Workspace, Save Workspace, Load History, Save History, Change dir..., Print, Save to file, Exit.

Edit = Copy, Paste, Paste commands only, Copy and Paste, Select all, Clear console, Data editor, GUI preferences,

View = toolbar, statusbar

Misc = Stop current computation, Stop all computations, Buffered output, Word completion, Filename completion, List objects, Remove all objects, List search path,

Packages = Load Packages, Set CRAN mirror, Select repositories, Install package(s), Update packages, Install packages from local zip files

Windows = Cascade, Tile Horizontally, Tile Vertically, Arrange Icons, 1 R Console

Help = Console, FAQ on R, FAQ on R for Windows, Manuals (PDF), R functions (text), Html help, Search help, search.project.org, Apropos, R project home page, CRAN home page, about

3.3. Základy R

Statistika je studium a analýza dat. První věc, která se musí udělat před použitím programu „R“, je jak zadávat data a manipulovat s daty. Program má výhodu, že uživatel může komunikovat interaktivním způsobem. Ptáte se na otázku a „R“ je schopné dát odpověď. Komunikuje se zde pomocí příkazového řádku.

3.3.1. Zadávání dat pomocí „c“

Nejvíce oblíbený příkaz v programu „R“ pro rychlé zadávání malých dat je funkce „*c(hodnoty)*“. Umožňuje kombinovat a zřetězit pojmy dohromady.

Ukázka užití:

V prvním ročníku studia na dané vysoké škole, studuje určitý počet studentů s následujícím stářím studentů: 20 19 28 22 24 21 35

```
> student=c(20,19,28,22,24,21,35)
> student
[1] 20 19 28 22 24 21 35
>
```

Pro získání již jednou zapsaného příkazu, může uživatel použít šipky na klávesnici nebo posuvné kolečko na myši.

Pokud si uživatel neví rady, co znamená daný příkaz, napíše do příkazového řádku níže uvedenou syntaxi s otazníkem. R následovně otevře v internetovém prohlížeči stránku, kde je daná funkce vysvětlená (funguje to u každého příkazu).

```
>?c
```

3.3.2. Použití funkcí

Program R ukládá data jako vektor, protože tak zachovává pořádek v datech (nedojede k jejich záměně). Při změně daného čísla ve vektoru, nemusíme provádět změnu celého vektoru, ale pouze dané položky. Pro R je vektor také matematický objekt. To znamená, že mezi vektory můžeme provádět početní operace (sčítání, odečítání, násobení, dělení atd.). R dokáže zobrazit výběry dat.

Ukázka užití:

Uvažujme předešlý příklad studentů dané vysoké školy. Dvaceti dvouletý student byl z ročníku vyloučen a nahradil ho osmnáctiletý student.
20 19 28 18 24 21 35

```
> student=c(20,19,28,22,24,21,35)
```

```

> student
[1] 20 19 28 22 24 21 35
> student=c(20,19,28,22,24,21,35)
> student2=student
> student2[4]=18
> student2
[1] 20 19 28 18 24 21 35
>

```

R nám umožňuje zobrazit (nezobrazit) hodnoty, které v danou situaci potřebujeme (nepotřebujeme). Dokáže také zobrazit pořadí určité hodnoty. Pomocí R můžeme také zjistit, nejstaršího a nejmladšího studenta.

```

> student2
[1] 20 19 28 18 24 21 35
> student2[c(1,2,3)]
[1] 20 19 28
> student2[4]
[1] 18
> student2[6]
[1] 21
> student2[-5]
[1] 20 19 28 18 21 35
> max(student2)
[1] 35
> min(student2)
[1] 18
>

```

Jestliže chceme zjistit, zda jsou v ročníku studenti stejného věku, použijeme příkaz „`=`“. Pokud se v datech objeví stejné hodnoty, R je označí „*TRUE*“ pokud ne a pokud ano tak „*FALSE*“.

Do ročníku přistoupil další osmnáctiletý student. 20 19 28 18 24 21 35 18

```

> student3=student2
> student3[8]=18
> student3
[1] 20 19 28 18 24 21 35 18
> student3==18
[1] FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
>

```

Pokud nás zajímá, kolik studentů máme ve třídě, můžeme použít další příkazy. Můžeme také dvojím způsobem zjistit, jaké místo zabírají dva osmnáctiletí studenti. Jednou pomocí příkazu „*which(...)*“ a druhým pomocí „*pages(...)*“. V R jako u každého programování, se musí dbát na typ závorčky.

```

> n=length(student3)

```

```

> pages=1:n
> pages
[1] 1 2 3 4 5 6 7 8
> pages[student3==18]
[1] 4 8
> which[student3==18]
Error in which[student3 == 18] :
  object of type 'closure' is not subsettable
> which(student3==18)
[1] 4 8
>

```

Prostřednictvím příkazu „**sum**(*proměnná*)“ sečteme všechny hodnoty.

```

> sum(student3)
[1] 183
> sum(student3>23)
[1] 3

```

Pokud bychom potřebovali zjistit základní statistické hodnoty (medián, modus, maximum, minimum atd.), R má několik možností zobrazení. Zobrazí je buď jednotlivě, nebo veškeré informace pod jedním příkazem. Pomocí příkazu „**median**(*proměnná*)“ a „**mean**(*proměnná*)“ se zobrazí hodnoty samostatně. Stejný princip pro zobrazení jednotlivých statistických hodnot existuje i další funkce například („*var*“). Při hromadném zobrazení statistických hodnot R umožňuje funkci „**fivenum**(*proměnná*)“ nebo „**summary**(*proměnná*)“.

```

> mean(student3)
[1] 22.875
> median(student3)
[1] 20.5
> var(student3)
[1] 35.55357
> fivenum(student3)
[1] 18.0 18.5 20.5 26.0 35.0
> summary(student3)
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.
18.00 18.75 20.50 22.88 25.00 35.00
>

```

3.4. Grafy v R

3.4.1. Sloupcový graf

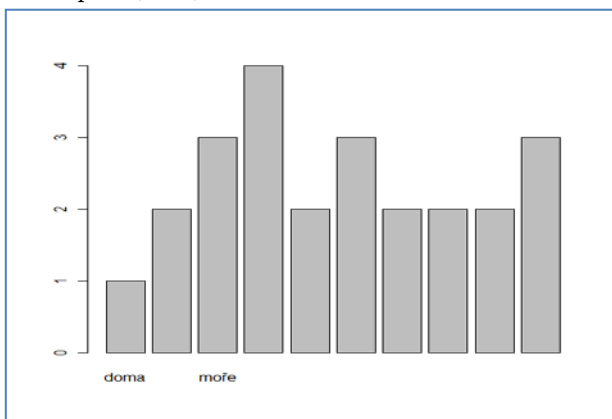
Sloupcový graf je jeden z nejběžnějších typů grafů v R. Pomocí příkazu „**barplot**(*proměnná*)“ a správnými úpravami, se daný graf zobrazí.

Ukázka užití:

Předpokládáme skupinu 10 osob, které byly dotazovány, kde rádi tráví dovolenou. Odpovědět mohli následovně: 1- doma, 2- čechy, 3- moře, 4- hory

1 2 3 4 2 3 2 2 2 3

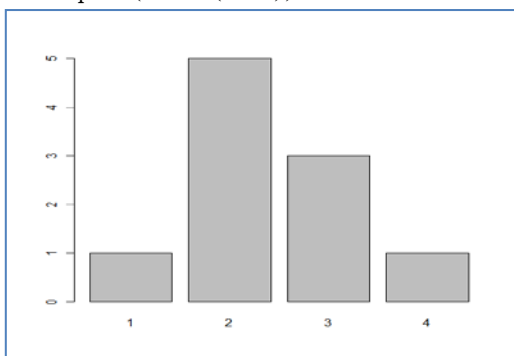
```
> dov=c(1,2,3,4,2,3,2,2,2,3)
> dov
[1] 1 2 3 4 2 3 2 2 2 3
> barplot(dov)
```



Obrázek 9: Sloupcový graf

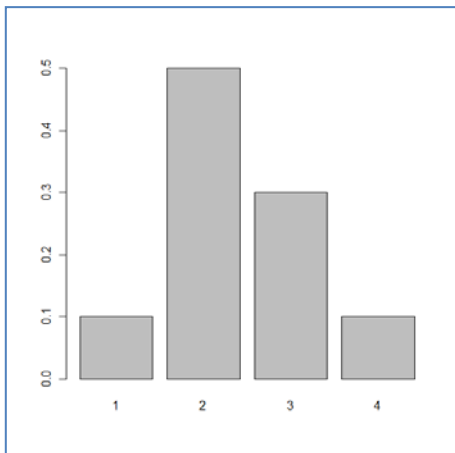
Výše uvedený graf (obr.9) je bohužel nepřehledný a nepoužitelný. Pro větší přehlednost v sloupcovém grafu, se data rozdělí prostřednictvím složeného příkazu „**barplot(table(proměnná))**“. K získání sloupcového grafu relativní četností se příkaz vydělí délkou vektoru tzn. „**barplot(table(proměnná) / length (proměnná))**“. Zabarvení dílčích sloupců je příkaz „**col(hodnoty barvy)**“. U hodnot barvy sloupců, se píše název dané barvy. Tyto názvy barev se vyskytují například ve speciálních programech, které podporují barevné rozlišení.

```
> barplot(table(dov))
```



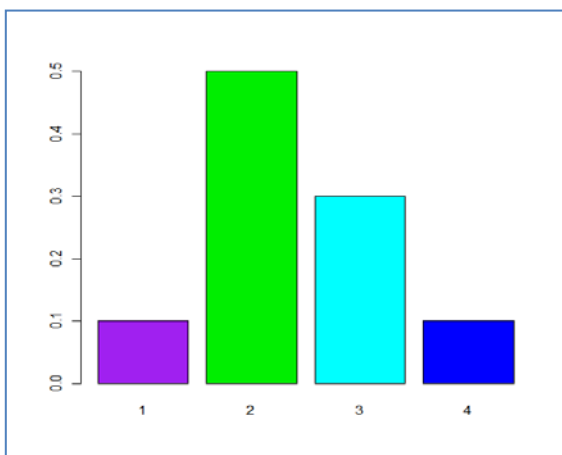
Obrázek 10 Sloupcový graf

```
> barplot(table(dov)/length(dov))
```



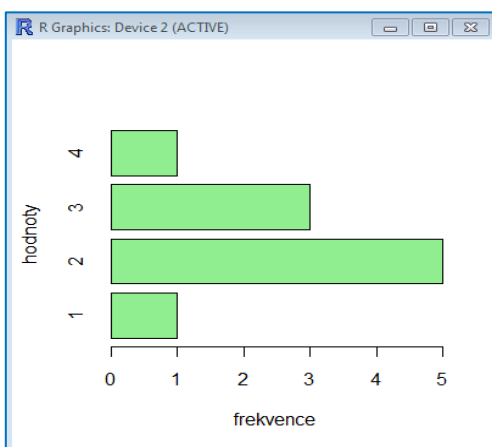
Obrázek 11: Sloupcový graf

```
> barplot(table(dov)/length(dov),col=c("purple","green2","cyan","blue"))
```



Obrázek 12: Sloupcový graf

```
> barplot(table(dov), horiz=T, col="lightgreen", xlab="frekvence", ylab="hodnoty")
>
```



Obrázek 13: Horizontální sloupcový graf

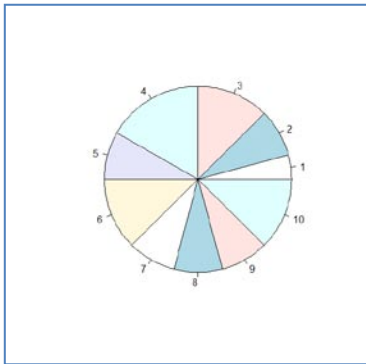
3.4.2. Kruhový graf (koláč)

Kruhový graf se objeví pod příkazem „**pie(proměnná)**“. Je potřeba, aby daná data byla správně převedena. Prostřednictvím příkazu „**proměnná2=table(proměnná1)**“ převede R data do správné formy. Pojmenování dat a rozlišení podle barev, program R také umožňuje. Pod příkazem „**names(proměnná)=c(„hodnoty“)**“ se dané oblasti přejmenují. Podobně „**.....col(„hodnoty“)**“ umožní danou oblast grafu barevně rozlišit.

```
> table(dov)
dov
1 2 3 4
1 5 3 1
```

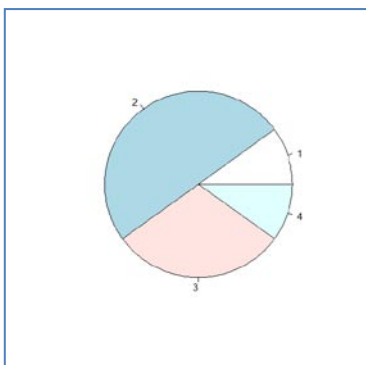
Samotný příkaz „**pie(proměnná)**“ v daném příkladě nemá smysl. Tento příkaz zobrazí každou sledovanou hodnotu a její velikost na celku samostatně. Proto se používají dále níže uvedené příkazy.

```
> pie(dov)
```



Obrázek 14: Kruhový graf

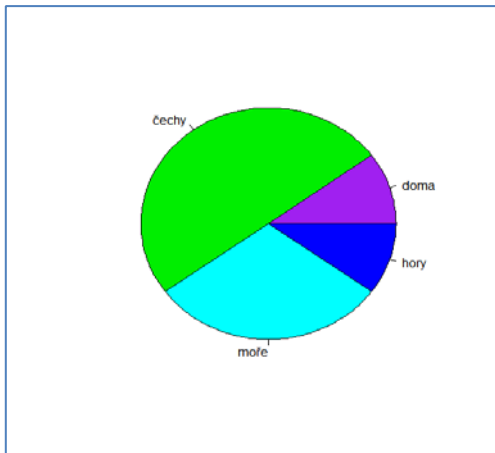
```
> dov=c(1,2,3,4,2,3,2,2,2,3)
> koláč=table(dov)
> pie(koláč)
```



Obrázek 15: Kruhový graf

```
> names(koláč)=c("doma", "čechy", "moře", "hory")
```

```
> pie(koláč)
> pie(koláč,col=c("purple","green2","cyan","blue"))
```



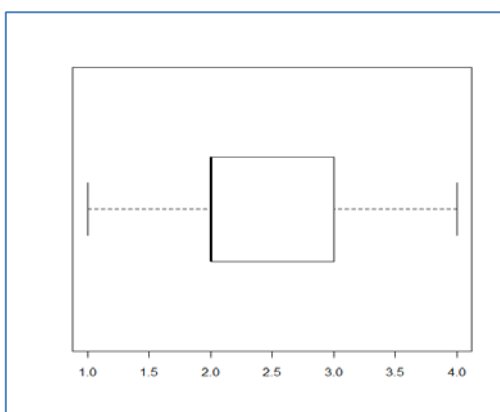
Obrázek 16: Kruhový graf

Pojmenování oblasti dat se píše vždy v uvozovkách, aby nedocházelo k záměně funkce či proměnné. Prostřednictvím uvozovek, program R chápe tyto názvy jako řetězec slov. Barevné rozlišení se také píše do uvozovek. Různé typy barev získáme ze základní nabídky například ve “Adobe Photoshopu”.

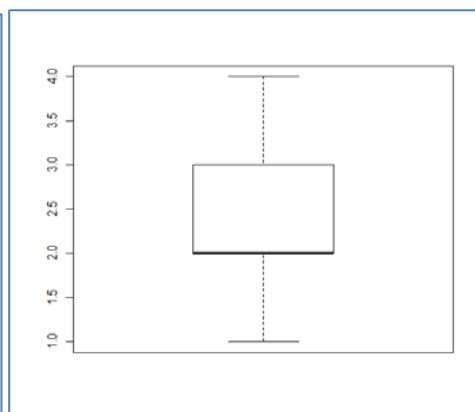
3.4.3. Boxplot

Boxplot neboli krabicový graf je jeden z nejběžnějších grafů ve statistice. Zobrazuje pět číselných statistických hodnot. Nejmenší a největší pozorované hodnoty, poté Q1 (dolní kvartil), Q2 (medián), Q3 (horní kvartil). Boxplot také uvádí odlehlé hodnoty od pozorování.

```
> boxplot(dov)
> boxplot(dov,horizontal=TRUE)
```



Obrázek 17: Boxplot 1

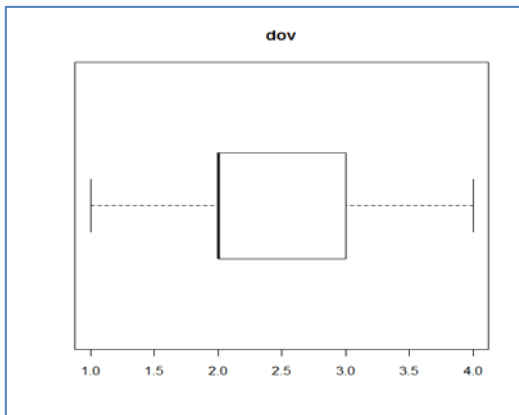


Obrázek 18: Boxplot 2

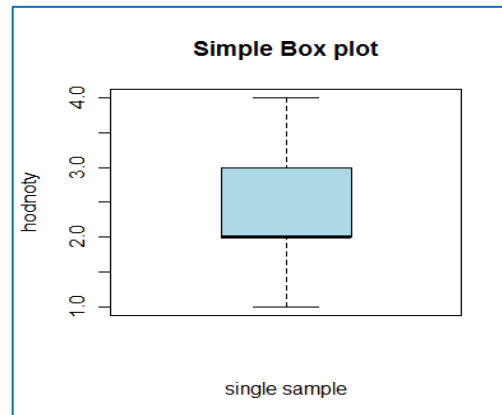
Část příkazu „**horizontal=TRUE**)“ znamená, že daný boxplot bude na horizontální ose. V případě vertikální osy platí část příkazu „**vertikal=TRUE**)“.

```
> boxplot(dov,main="dov", horizontal=TRUE)
```

```
> boxplot(dov, xlab="single sample", ylab="hodnoty", main="Simple Box plot",
col="lightblue")
```



Obrázek 19: Boxplot 3

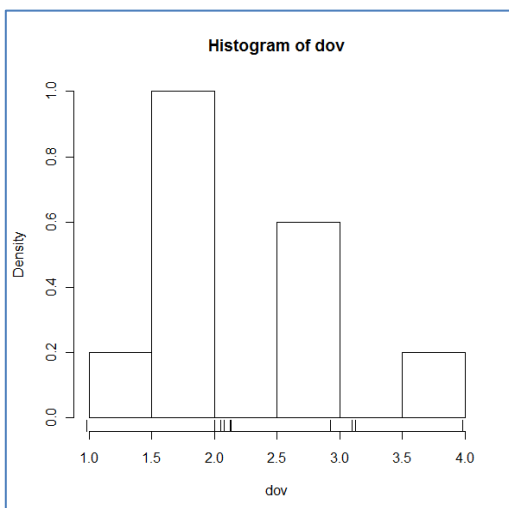


Obrázek 20: Boxplot 4

3.4.4. Histogram

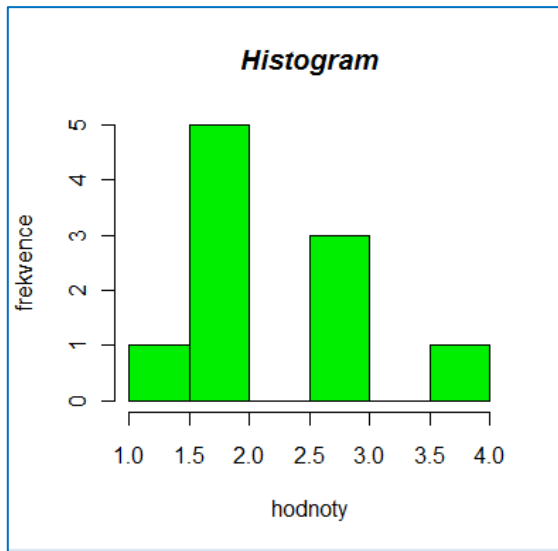
```
> hist(dov)
> hist(dov,probability=TRUE)
> rug(jitter(dov))
```

Při použití jednoduchého příkazu „`hist(proměnná)`“ se zobrazí pouze strohý graf. Aby se zobrazila pravděpodobnost, musí se tento příkaz rozšířit na „`hist(proměnná,probability=TRUE)`“. Prostřednictvím příkazu „`rug(jitter(proměnná))`“ se na histogramu objeví tzv. pravděpodobnostní šum.



Obrázek 21: Histogram

```
> hist(dov, col="cornsilk", xlab="hodnoty", ylab="frekvence", main="Histogram",
font.main=4)
```



Obrázek 22: Barevný histogram

3.5. Správa pomocí balíčků v R

Všechny funkce a datové soubory jsou v R uloženy v balíčcích. Základní balíčky (standard) jsou součástí zdrojového kódu. Obsahují základní funkce, které umožňují práci s datovými soubory, základní statistické a grafické funkce. Vše je v každém R automaticky instalováno. Tento program umožňuje koncovým uživatelům vytvoření vlastních balíčků. Výhoda vytvoření dalších balíčků zdokonaluje program R. Momentálně, existuje několik set balíčků, které vytvořili koncoví uživatelé. Balíčky obsahují, buď speciální statistické funkce, nebo data. Existují také balíčky, které rozšiřují současné funkce. Většina z těchto balíčků jsou k dispozici ke stažení na adrese <http://CRAN.R-project.org/> a další jsou k dispozici na <http://www.bioconductor.org/>.

Některé balíčky mají tzv. „jmenné prostory“ (namespace). Jmenné prostory mají za úkol dělat tři věci: za prvé umožňují spisovateli balíčku skrýt funkce a data, které jsou pro interní použití, brání koncovému uživateli zasahovat do funkce a poskytuje odkazovat se na daný objekt v rámci balíčku.

```
> library()
```

```
Packages in library 'C:/PROGRA~2/R/R-211~1.1/library':
```

base	The R Base Package
boot	Bootstrap R (S-Plus) Functions (Canty)
class	Functions for Classification
cluster	Cluster Analysis Extended Rousseeuw et al.
codetools	Code Analysis Tools for R
datasets	The R Datasets Package
foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
graphics	The R Graphics Package
grDevices	The R Graphics Devices and Support for Colours and Fonts
grid	The Grid Graphics Package

KernSmooth	Functions for kernel smoothing for Wand & Jones (1995)
lattice	Lattice Graphics
MASS	Main Package of Venables and Ripley's MASS
Matrix	Sparse and Dense Matrix Classes and Methods
methods	Formal Methods and Classes
mgcv	GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL
nlme	Linear and Nonlinear Mixed Effects Models
nnet	Feed-forward Neural Networks and Multinomial Log-Linear Models
rpart	Recursive Partitioning
spatial	Functions for Kriging and Point Pattern Analysis
splines	Regression Spline Functions and Classes
stats	The R Stats Package
stats4	Statistical Functions using S4 Classes
survival	Survival analysis, including penalised likelihood.
tcltk	Tcl/Tk Interface
tools	Tools for Package Development
utils	The R Utils Package

Prostřednictvím příkazu „**search()**“ se dá zjistit, které balíčky jsou natrvalo instalovány v programu R. Pokud, se nezobrazí v daném seznamu balíčků, který obsahuje potřebnou funkci, musí se ručně nainstalovat z vhodných webových stránek.

```
> search()
[1] ".GlobalEnv"      "package:stats"   "package:graphics"
[4] "package:grDevices" "package:utils"   "package:datasets"
[7] "package:methods" "Autoloads"      "package:base"
```

Instalace balíčku v R je velmi jednoduchá. Pokud uživatel je připojený k internetu, může použít příkaz „> **install.packages()**“. Zobrazí se tabulka, kde si uživatel vybere správný jazyk instalace a poté výběr balíčku. Jestliže program R potřebuje nainstalovat balíček přímo od vývojářů R, stačí příkaz „> **library(boot)**“. Pro zjištění správně načtených balíčků, se použije příkaz „**search()**“.

3.6. Export a Import dat

3.6.1. Import dat

Import dat z jiných statistických programů umožňuje i prostředí programu R. Je možné data importovat z Excelu, SPSS, Stata, Systat, S-PLUS a SAS. Postup při importu dat z excelovského souboru je následující:

1) Daný excelovský soubor uložíme pod koncovkou „csv“. Zkratka „csv“ je jednoduchý souborový formát zápisu různých dat, který umožňuje předání dat mezi různými systémy. CSV (**Comma-separated values**) neboli hodnoty oddělené čárkami, oddělují jednotlivé položky v řádku čárkami.

2) Při uzavření excelovského souboru, uloženého jako „csv“, zjistíme přímou lokalizaci souboru na počítači. Kliknutím pravým tlačítkem myši na soubor, zvolíme v nabídce „Vlastnosti“. Odrážka „obecné“ zahrnuje informaci o „umístění“. Umístění zobrazuje přímou lokalizaci složky v počítači. Tuto lokalizaci si uživatel opíše nebo zkopíruje do jiné složky.

3) R nabízí pro otevření dat z jiného programu následující funkci: „read.csv2“

```
| > regrese=read.csv2(file="file://C:/Users/Mary/Desktop/regrese1.csv") |
```

Tento příkaz je přímo pro otevření daného excelovského souboru jménem regrese1, uloženého jako „csv“. Pro excelovskou složku „regrese1“ je lokalizace v počítači následující „C:/Users/Mary/Desktop/“. Pro jiné dokumenty, a jiný uživatelský počítač je lokalizace samozřejmě jiná. Příkaz je doplněný na začátku o slovo „file//“, na konci se píše jméno souboru.

4) Nyní má R ve své paměti hodnoty excelovského souboru. Při zadání příkazu „regrese“(jméno zadané proměnné), se vypíšou všechny hodnoty. R pracuje s danými hodnotami jako s maticí.

```
| > regrese |
```

	STUDPRUM	BODY	MATURITA	POHLAVI	PRIPRAVA	BYDLENI
1	2.70	160	1.80	1	6	1
2	2.60	135	2.30	0	7	0
3	2.50	154	2.00	1	9	1
4	2.50	129	2.13	0	10	0
5	2.40	150	2.00	0	6	0
6	2.40	147	2.00	1	8	0
7	2.35	132	2.33	0	9	0
8	2.33	130	2.20	1	10	1
9	2.33	142	2.20	1	12	0
10	2.33	160	1.80	1	8	0
11	2.33	145	2.00	1	9	1
12	2.30	138	2.20	0	7	1
13	2.30	140	2.20	0	7	1
...
22						

Daný příklad obsahuje sto studentů a jejich školní výsledky.

Pro výpočet statistických hodnot, můžeme použít příkaz „*summary(proměnná)*“. R vypíše každou proměnnou zvlášť, a pro ně jednotlivé statistické hodnoty.

²² Převzato z přednášky 4ST321

```
> summary(regrese)
  STUDPRUM      BODY      MATURITA      POHLAVI      PRIPRAVA      BYDLENÍ
Min. :1.120    Min. :120.0  Min. :1.200    Min. :0.00    Min. : 5.00    Min. :0.0
1st Qu.:1.800    1st Qu.:137.8  1st Qu.:1.788  1st Qu.:0.00    1st Qu.: 7.00    1st Qu.:0.0
Median :2.000    Median :149.0  Median :2.000  Median :1.00    Median : 8.00    Median :0.0
Mean :1.966     Mean :150.0    Mean :1.954    Mean :0.56     Mean : 8.87     Mean :0.3
3rd Qu.:2.140    3rd Qu.:160.0  3rd Qu.:2.200  3rd Qu.:1.00    3rd Qu.:10.00   3rd Qu.:1.0
Max. :2.700     Max. :192.0    Max. :2.800    Max. :1.00     Max. :15.00     Max. :1.0
```

Pokud uživatel potřebuje z daného souboru pouze jednu proměnnou, a sní patřičně počítat, vybere si prostřednictvím příkazu „**data=regrese[2]**“ pouze proměnnou „BODY“. Jak již bylo předem psáno, program R má všechny proměnné jako v matici. Pro vyznačení studentů do uvozovek, aby se rozlišili proměnné, pomůže příkaz „**write.table**“

```
> data=regrese[2]          > write.table(data)          > write.csv2(data)
> data                    "BODY"                          "";"BODY"
  BODY                    "1" 160                          "1";160
1  160                    "2" 135                          "2";135
2  135                    "3" 154                          "3";154
3  154                    "4" 129                          "4";129
4  129                    "5" 150                          "5";150
5  150                    "6" 147                          "6";147
6  147                    "7" 132                          "7";132
7  132                    "8" 130                          "8";130
8  130                    "9" 142                          "9";142
9  142                    "10" 160                         "10";160
10 160
```

3.6.2. Export dat

Export dat do různých programů, jak statistických (SPSS, SAS a Stata) a nestatistických (EXELL a textový editor) je v R přes příkazový řádek.

Důležité na exportu je umístění daného souboru v počítači. Export může být u každého uživatele jiný, neboť záleží, kterou verzí programu R uživatel používá, a jaké balíčky má stažené.

a) Export dat do Textového souboru

```
> write.table(y, "c:/mydata.txt", sep="\t")
```

b) Export dat do EXELLU

Export dat do exelovského souboru může být pro uživatele docela obtížný. Nejprve si musí stáhnout balíček obsahující syntaxi „**xlsReadWrite**“. Pokud uživatel má stažený balíček, který obsahuje „**scran ()**“ musí tento balíček odstranit, neboť pro export pomocí „**xlsReadWrite**“ by se oba balíčky navzájem přerušovali a nefungovalo by to.

Pomocí uvedeného příkazu odstraní balíček obsahující syntaxi „**scran ()**“

```
> xls.getshlib()
```

Poté bude fungovat uživateli následující příkaz pro starou verzi EXELU tzn. verze 1997-2003.

```
| > write.xls(y, "C:/mytest.xls") |
```

Příkaz „**write.xls()**“ je syntaxe. Závorka obsahuje proměnnou y, kterou jsme si zadaly do programu R. Poté, musíme oddělit čárkou proměnnou od cesty, kde má být soubor uložen. Klasická syntaxe pro uložení je všude stejná.

Pro uživatele, který má novou verzi office-programů (např. Exxel 07), je výše popsaná cesta obtížná. Uživatel musí mít na svém počítači instalovanou „JAVU“. Pro program R musí si nainstalovat potřebné balíčky, jak pro čtení nových dokumentů, tak pro podporující Javu. Například: „xlsx“, „ijava“, „rjava“ atd.

c) Export dat do SPSS

```
| >write.foreign(y, "c:/data.txt","c:/data.sps",package="SPSS") |
```

d) Export dat do SAS

```
| >write.foreign(y, "c:/data.txt", "c:/data.sas", package="SAS") |
```

e) Export dat do STATA

```
| >write.dta(y, "c:/data.dta") |
```


4. Regresní analýza v programu R

Lineární regrese napomáhá vysvětlit průběh závislosti mezi proměnnými. Při správném rozhodnutí u vybraní regresního typu, jsou důležitá věcně ekonomická kritéria a empirické způsoby. Pro zjištění empirických jevů u daného souboru dat pomáhá využití statistických programů. Mezi statistické programy zařazujeme i program R. Pomocí R zjistíme grafickou metodou typ regresní funkce. Průběh závislosti mezi proměnnými x a y u grafické metody tvoří body. Dvojice pozorování x a y na grafu znázorňuje tzv. bodový diagram. Můžeme využít již zmíněné grafy v R.

Při určování typu regresní funkce u vícenásobné lineární regrese nám nepomůže možnost zachycení pomocí grafického průběhu. U tohoto typu lineární regrese se opíráme o matematicko-statistické kritéria (směrodatné chyby, míry těsnosti, různé testy), která nám posoudí vhodný typ regresní funkce.

4.1. Výpočet Jednoduché lineární regrese v R

Jednoduchá lineární regrese je nejčastější forma závislosti mezi proměnnými. R využívá příkaz „**lm(Y ~ model)**“ pro jakýkoliv typ regrese. K použití příkazu „**lm(.....)**“ je potřeba si stáhnout daný balíček, který jej obsahuje, neboť v základním složení balíčků v R tento typ příkazu neobsahuje. Na webové adrese <http://cran.ma.imperial.ac.uk/> si může uživatel najít funkci, kterou momentálně potřebuje. Daná webová adresa je pro britský kontinent. Na webové adrese <http://www.r-project.org/> si může uživatel vybrat zemi, která je pro něj jazykem bližší. Po výběru se dostane k potřebným balíčků. Je zde také jednodušší možnost, jak si nainstalovat potřebnou funkci z balíčků. Na horní liště v programu R je v menu „Packages“. Pokud je uživatel připojený k internetové síti, může z nabídky „packages“ vybrat „Instal package(s)“. Nejprve se mu zobrazí různé země, a je na uživateli, který jazyk dané země je mu bližší. Po vybrání země, se objeví seznam balíčků, a jestli uživatel má ve vědomí, který si musí nainstalovat, klikne na něj.

Pro příkaz „**lm(.....)**“ je stažený balíček „Linear mixed models“.

$y \sim x_1$ y model pro x_1 (pouze u jednoduché lineární regrese)

```
>lm(y~x1y)
```

Jestliže data nejsou lineární, v R můžeme data logicky aproximovat na lineární tvar. Tato metoda je stejná jako u ostatních statistických programů.

Následující tabulka ukazuje různé způsoby syntaxe v R.

syntax	model
$Y \sim A$	$Y = \beta_0 + \beta_1 A$
$Y \sim -1 + A$	$Y = \beta_1 A$
$Y \sim A + I(A^2)$	$Y = \beta_0 + \beta_1 A + \beta_2 A^2$
$Y \sim A + B$	$Y = \beta_0 + \beta_1 A + \beta_2 B$
$Y \sim A:B$	$Y = \beta_0 + \beta_1 AB$
$Y \sim A*B$	$Y = \beta_0 + \beta_1 A + \beta_2 B + \beta_3 AB$
$Y \sim (A + B + C)^2$	$Y = \beta_0 + \beta_1 A + \beta_2 B + \beta_3 C + \beta_4 AB + \beta_5 AC + \beta_6 AC$

23

²³ „Using R for Linear Regression“

Ukázka užití:

Byl proveden experiment, vliv určité látky (mg) působící na snížení tepu lidského srdce. Byly zjištěny níže uvedeny hodnoty u 13 osob. Nezávislá proměnná je dávkování léku (v mg), a závislou proměnnou je rozdíl mezi nejnižší hodnotou po podání a před podáním léku.

Množství (mg) látky na redukci srdečního tepu = z (x)
0.5,0.75,1,1.25,1.5,1.75,2,2.25,2.5,2.75,3,3.25,3.5
O kolik byl snížen tep po podání látky = v (y)
10,8,12,12,14,12,16,18,17,20,18,20,21

24

```
> z=c(0.5,0.75,1,1.25,1.5,1.75,2,2.25,2.5,2.75,3,3.25,3.5)
> v=c(10,8,12,12,14,12,16,18,17,20,18,20,21)
> summary(lm(v~z))
```

Call:

```
lm(formula = v ~ z)
```

Residuals:

```
  Min      1Q  Median      3Q     Max
-2.2088 -0.3626 -0.1648  0.8571  1.7473
```

Coefficients:

```
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.0549     0.8876   7.949 6.94e-06 ***
z             4.0879     0.4020  10.169 6.25e-07 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.356 on 11 degrees of freedom

Multiple R-squared: 0.9039, Adjusted R-squared: 0.8951

F-statistic: 103.4 on 1 and 11 DF, p-value: 6.25e-07

```
>
```

$$b_0=7,0549$$

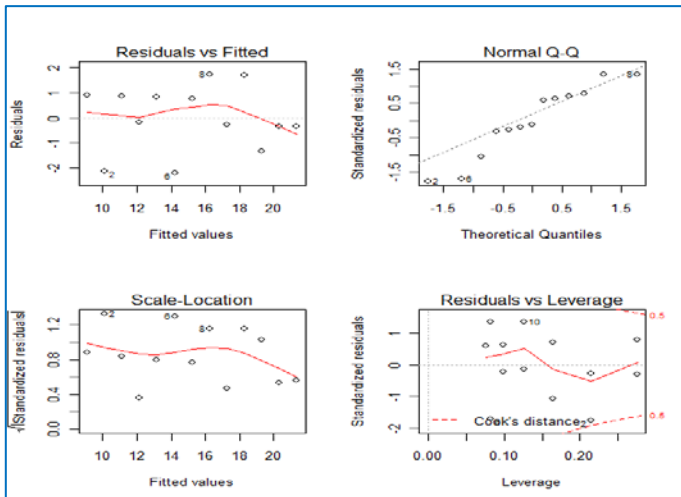
$$b_1= 4,0879$$

$$Y=7,0549 + 4,0879x$$

Ke zvýšení rozdílu měření tepu srdce před a po podání léku o jednotku je třeba podat o 4,09 mg léku více. Příklad je danou lineární regresí popsán na 90,4%. Zbytek je část, která není popsána modelem.

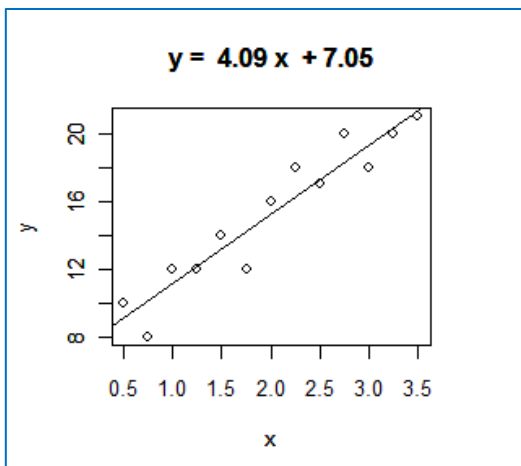
²⁴ Biostatistics:A foundation for analysis in the health science, Wayne W.Daniel, ISBN: 0-471-52514-6 (str.376)

```
> plot(lm(v~z))
Waiting to confirm page change...
Waiting to confirm page change...
Waiting to confirm page change...
Waiting to confirm page change...
> par(mfrow=c(2,2))
> plot(lm(v~z))
> plot(z,v)
```



Obrázek 23: Regrese

```
> abline(lm(v~z))
> lm.result=simple.lm(z,v)
```



Obrázek 24: Lineární regrese

Při testu hypotézy o významnosti parametrů regresní funkce použijeme následující vzorce. Směrodatné chyby regresních koeficientů, se můžou vyčíst z výše uvedené tabulky pod příkazem „**summary(...)**“. Druhou možností je použít následující vzorce a aplikovat je v R.

$$SE(b_1) = \frac{s}{\sqrt{\sum(x_i - \bar{x})^2}} \quad t(b_1) = \frac{b_1 - \beta_1}{SE(b_1)}$$

$$SE(b_0) = s \sqrt{\frac{\sum x_i^2}{n \sum(x_i - \bar{x})^2}} \quad t(b_0) = \frac{b_0 - \beta_0}{SE(b_0)}$$

$$H_0: \beta_1 = 0$$

$$H_1: \beta_1 \neq 0$$

Na 5 % hladině významnosti testujeme hypotézu o nulové hodnotě reg. parametru

Jelikož $t = 10,16917$ tabulky →

$$s = 1,355$$

$$SE = 0,402$$

zadaný příklad má 11 stupňů volnosti a hodnota $t_{0,975} = 2,201$, zamítáme hypotézu o nulové hodnotě regresního parametru. To znamená, že daný regresní parametr je v modelu významný.

```
> data=(lm(v~z))
> es=resid(data)
> b1=(coef(data))['z']
> s=sqrt(sum(es^2)/(13-2))
> SE=s/sqrt(sum((z-mean(z))^2))
> t=(b1-(0))/SE
> pt(t,11,lower.tail=FALSE)
```

z

3.125215e-07

> SE

[1] 0.4019909

> s

[1] 1.355788

> b1

z

4.087912

> t

z

10.16917

>

$$H_0: \beta_0 = 0$$

$$H_1: \beta_0 \neq 0$$

Na 5 % hladině významnosti testujeme hypotézu o nulové hodnotě reg. parametru

Jelikož $t = 7,94854$ tabulky →

$$s = 1,355$$

$$SE = 0,8875$$

zadaný příklad mám 11 stupňů volnosti a hodnota $t_{0,975} = 2,201$, zamítáme hypotézu o nulové hodnotě regresního parametru. Daný reg. parametre je v modelu významný.

```
> SE=s*sqr(sqrt(sum(z^2)/(13*sum((z-mean(z))^2)))
```

```
> SE
```

```
[1] 0.8875718
```

```
> b0=7.0549
```

```
> t=(b0-(0))/SE
```

```
> pt(t,11,lower.tail=FALSE)
```

```
[1] 3.472605e-06
> t
[1] 7.94854
> s
[1] 1.355788
>
```

4.2. Výpočet vícenásobné lineární regrese v R

Pro výpočet vícenásobné lineární regrese v R můžeme použít dva způsoby. Při vybrání způsobu výpočtu záleží pouze na uživateli. Nejjednodušší příkaz pro výpočet jak jednoduché lineární regrese, tak vícenásobné lineární regrese je pomocí příkazu „lm()“, který musí být stažen a nainstalován. Pro výpočet vícenásobné regrese a při použití metody nejmenších čtverců může uživatel aplikovat pomocí příkazového řádku vzorec.

Ukázka užití

Mějme 20 pacientů s věkovým rozmezím 45-56. U každého z pacientů byl naměřen daný krevní tlak a váha. Pomocí vícenásobné lineární regrese vypočítáme, jestli krevní tlak závisí na věku a váze.

t-patient	Y=krevní tlak (mm Hg)	X1=věk	X2=váha
1,00	105,00	47,00	85,40
2,00	115,00	49,00	94,20
3,00	116,00	49,00	95,30
4,00	117,00	50,00	94,70
5,00	112,00	51,00	89,40
6,00	121,00	48,00	99,50
7,00	121,00	49,00	99,80
8,00	110,00	47,00	90,90
9,00	110,00	49,00	89,20
10,00	114,00	48,00	92,70
11,00	114,00	47,00	94,40
12,00	115,00	49,00	94,10
13,00	114,00	50,00	91,60
14,00	106,00	45,00	87,10
15,00	125,00	52,00	101,30
16,00	114,00	46,00	94,50
17,00	106,00	46,00	87,00
18,00	113,00	46,00	94,50
19,00	110,00	48,00	90,50
20,00	122,00	56,00	95,70

25

Pomocí MNC vypočítám dané údaje. Pro kontrolu R uvádím výpočet také v Excelu.

²⁵ Biostatistics: A foundation for analysis in the health science, Wayne W. Daniel, ISBN: 0-471-52514-6 (str. 450)

4.2.1. Výpočet v EXCELU pomocí MNČ

V Excelu nejprve musí uživatel vypočítat matici transponovanou s danou maticí. Po výpočtu použije příkaz v Excelu pro výpočet inverzní matice. Inverzní matici vynásobí maticí, která vznikne vynásobením transponované matice X a matice Y.

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Xt

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
47,00	49,00	49,00	50,00	51,00	48,00	49,00	47,00	49,00	48,00	47,00	49,00	50,00	45,00	52,00	46,00	46,00	46,00	48,00	56,00
85,40	94,20	95,30	94,70	89,40	99,50	99,80	90,90	89,20	92,70	94,40	94,10	91,60	87,10	101,30	94,50	87,00	94,50	90,50	95,70

X

1	47,00	85,40
1	49,00	94,20
1	49,00	95,30
1	50,00	94,70
1	51,00	89,40
1	48,00	99,50
1	49,00	99,80
1	47,00	90,90
1	49,00	89,20
1	48,00	92,70
1	47,00	94,40
1	49,00	94,10
1	50,00	91,60
1	45,00	87,10
1	52,00	101,30
1	46,00	94,50
1	46,00	87,00
1	46,00	94,50
1	48,00	90,50
1	56,00	95,70

Xt*X		
20	972	1861,8
972	47358	90566,6
1861,8	90566,6	173665,4

inverze matice		
31,875	-0,268	-0,202
-0,268	0,010	-0,002
-0,202	-0,002	0,003

inverzní matice * Xt 3x20

-16,5794
0,708251
1,032961

$$Y = -16,58 + 0,708x_1 + 1,033 x_2$$

4.2.2. Výpočet regrese v R pomocí příkazu „lm()“

První myšlenka, která musí uživatele napadnout, je dané data pomocí příkazu „c()“ zavést do systému R. Po zadání dat použijeme příkaz „lm()“, a při vybrání vhodné syntaxe vypočítáme hodnoty regresních parametrů. Prostřednictvím příkazu „summary()“ a nového názvu celého vypočítaného souboru (např. regrese), zjistíme další důležité statistické hodnoty.

```
>y=c(105,115,116,117,112,121,121,110,110,114,114,115,114,106,125,114,106,113,1
10,122)
> x1=c(47,49,49,50,51,48,49,47,49,48,47,49,50,45,52,46,46,46,48,56)
>x2=c(85.4,94.2,95.3,94.7,89.4,99.5,99.8,90.9,89.2,92.7,94.4,94.1,91.6,87.1,101.3,94.
5,87,94.5,90.5,95.7)
> lm(y~x1+ x2)

Call:
lm(formula = y ~ x1 + x2)

Coefficients:
(Intercept)      x1      x2
-16.5794      0.7083      1.0330

> regrese=lm(y~x1+ x2)
> summary(regrese)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min     1Q  Median     3Q    Max
-0.89968 -0.35242  0.06979  0.35528  0.82781

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -16.57937    3.00746  -5.513 3.80e-05 ***
x1           0.70825    0.05351  13.235 2.22e-10 ***
x2           1.03296    0.03116  33.154 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5327 on 17 degrees of freedom
Multiple R-squared: 0.9914, Adjusted R-squared: 0.9904
F-statistic: 978.2 on 2 and 17 DF, p-value: < 2.2e-16
```

Z výstupu se dají vyčíst hodnoty regresních parametrů a zapsat je do rovnice

$$Y = -16,58 + 0,708x_1 + 1,033 x_2$$

V horní části výstupu je vidět daný tvar modelu, vysvětlovaná proměnná, vysvětlující proměnné. V odstavci „Residuals“ jsou základní statistické hodnoty modelu. Odstavec „Coefficients“ zaznamenává ve sloupcích nejen hodnoty hledaných regresních koeficientů. Ve sloupci „Estimate“ se nacházejí odhady regresních koeficientů, sloupec „Std.Error“ zaznamenává odhady směrodatných chyb odhadů, sloupec „t value“ jsou hodnoty testové statistiky a poslední sloupec „Pr....“ zaznamenává minimální hladiny významnosti.

Další hodnoty výstupu obsahují poměr determinace (0,9914), upravený poměr determinace (0,9904), residuální chybu (0,5327), F.statistiku atd.

Výpočet residuálních chyb pro každé y, vypočítáme pomocí příkazu „resid()“.

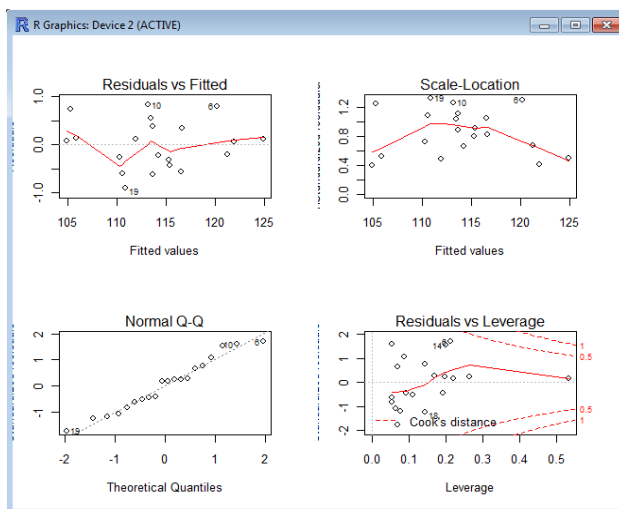
```
> resid(regrese)
  1    2    3    4    5    6    7    8    9   10   11
0.07667 -0.42989 -0.56614  0.34538  0.11182  0.80367 -0.21447 -0.60461 -
0.26508  0.82781 -0.21998
 12   13   14   15   16   17   18   19   20
-0.32659  0.54756  0.73714  0.11133  0.38498  0.13219 -0.61502 -0.89968
0.06291
```

Výpočet předpokládaných hodnot pro y, vypočítáme pomocí příkazu „fitted()“.

```
> fitted(regrese)
  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
104.9 115.4 116.6 116.7 111.9 120.2 121.2 110.6 110.3 113.2 114.2 115.3 113.5
105.3 124.9 113.6 105.9
 18   19   20
113.6 110.9 121.9
```

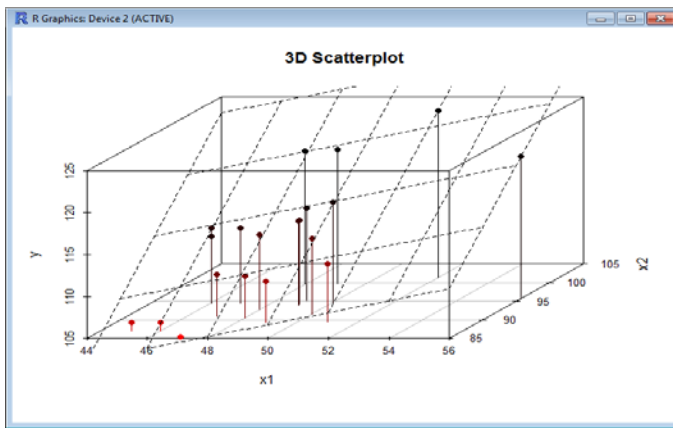
Pro zobrazení každého grafu s jiným významem, souvisí následující příkaz.

```
> layout(matrix(1:4,2,2))
> plot(regrese)
```



Obrázek 25: Regrese

4.2.3. Grafické zobrazení příkladu pomocí roviny v R

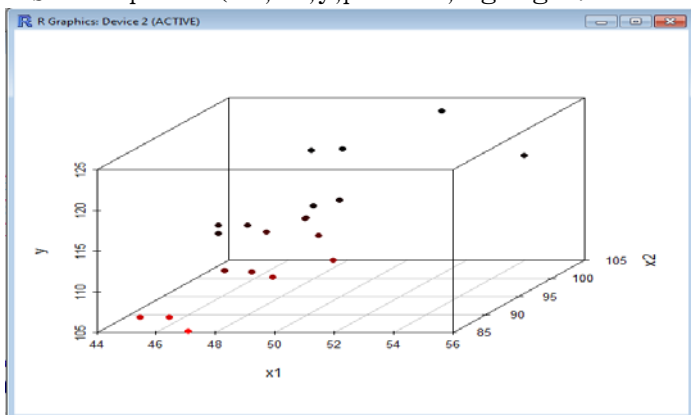


Obrázek 26: Grafické zobrazení vícenásobné regrese

Pro zobrazení daného 3D grafu „Scatterplot“ je nutné si stáhnout balíček „scatterplot3d_0.3-31“. Tento balíček je v souboru „zip“, a proto si uživatel musí daný balíček stáhnout do PC a ručně nainstalovat pomocí horní lišty v R (packages -> Install package(s) from local zip).

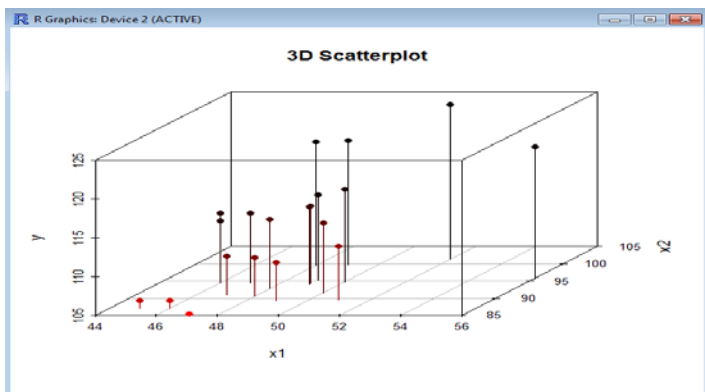
Postup

```
> scatterplot3d(x1,x2,y,pch=16,highlight.3d=TRUE)
```



Obrázek 27: Postup-Grafického zobrazení 1

```
> scatterplot3d(x1,x2,y,pch=16,highlight.3d=TRUE,type="h",main="3D Scatterplot")
```



Obrázek 28: Postup-Grafického zobrazení 2


```
[1] 20
> K<-3
> Ye <- beta[1]*x[,1]+ beta[2]*x[,2]+ beta[3]*x[,3]
> erro <- y-Ye
> sum(erro^2)
[1] 4.824

> sigma2 <- sum(erro^2)/(N-K)
> sigma2
[1] 0.2838
> sigma <- sqrt(sigma2)
> sigma
[1] 0.5327
```

Výpočet Kovarianční matice se provede podle následujících úvah:

$$C(b)=\sigma^2(F'F)^{-1}$$

$$C(b)=S^{(b)}=s^2(F'F)^{-1}$$

Hlavní diagonála matice C(b) obsahuje odhady rozptylů odhadovaných parametrů. Odhad rozptylu odhadu regresních parametru b_j daným vzorcem $s_{bj}^2 = s^2 x^{(jj)}$

Míra přesnosti odhadu je dána směrodatnou chybou odhadu $s_{bj} = s \sqrt{x^{(jj)}}$

```
> var.beta <- sigma2*xx
> var.beta
      [,1] [,2] [,3]
[1,] 9.04481 -0.0759540 -0.0573558
[2,] -0.07595 0.0028637 -0.0006792
[3,] -0.05736 -0.0006792 0.0009707
```

Prostřednictvím příkazu „**diag(proměnná)**“, se vypíšou hodnoty obsahující na hlavní diagonále dané kovarianční matice. Příkaz „**sqrt(proměnná)**“ odmocní danou proměnnou.

```
> diag(var.beta)
[1] 9.0448079 0.0028637 0.0009707
```

Následující hodnoty jsou odhady směrodatných chyb odhadů.

```
> sqrt(diag(var.beta))
[1] 3.00746 0.05351 0.03116
> (sqrt(diag(var.beta)))/beta[,1]
[1] -0.18140 0.07556 0.03016
```

Poměr determinace vypočítáme podle následujícího vzorce $R^2 = \frac{S_T}{S_y}$

Kde $S_T = \sum(Y_i - \bar{y})^2$ (teoretický součet čtverců) a $S_y = \sum(y_i - \bar{y})^2$ (reziduální součet čtverců)

Koeficient determinace v daném případě vyšel 99,14 % to znamená, že variabilita krevního tlaku v daném souboru byla vysvětlena zvoleným regresním modelem na 99,14 %.

Nevysvětlená část může být způsobena působením dalších vlivů. Z výsledku teda vyplývá, že problém byl vyřešen správným modelem.

```
> R2 <- 1-sum(erro^2)/sum((y-mean(y))^2)
> R2
[1] 0.9914
>
```

Výpočet korelačního koeficientu udává vlastnosti závislosti. Zaznamenáváme ho v intervalu $<-1,1>$ a znaménko určuje směr závislosti. Naměřené hodnoty směřující k nule odpovídají na slabou lineární závislost mezi proměnnými x_1 a x_2 a hodnoty směřující k 1 ukazují na vysokou kladnou korelaci. Hodnoty, které naopak směřují k -1, se nacházejí ve vysoké záporné korelaci.

$r = \sqrt{R^2}$ jestliže je b_{21} (b_{12}) kladné
 $r = -\sqrt{R^2}$ jestliže je b_{21} (b_{12}) záporné

```
> sqrt(R2)
[1] 0.9957
```

Hodnoty regresních parametrů i na potřetí vyšli jak je níže popsáno (x_1 -věk, x_2 - váha). Níže popsané výsledky udávají že $b_1 = 0,708$ tzn. věk pacienta, a při zvýšení o jednotku (1 rok) věku pacienta při neměnné váze pacienta, se krevní tlak zvýší o 0,708 mm Hg. Při zvýšení váhy o jednotku (1 kg) a při neměnném věku, se zvýší krevní tlak o 1,033 mm Hg. Z pohledu relativních směrodatných chyb odhadu regresních parametrů je velmi nízká u $b_1 = 7,5\%$ a u $b_2 = 3,01\%$.

$$b_0 = -16,58$$

$$b_1 = 0,708$$

$$b_2 = 1,033$$

$$Y = -16,58 + 0,708x_1 + 1,033 x_2$$

Celkový F-test je pro daný příklad vhodný, neboť je jsou zde více jak jedna vysvětlující proměnná. Tento test udává významnost modelu jako celku. F-test používá Fischerovo rozdělení. Vypadá následovně:

$$F = \frac{R^2}{1-R^2} \frac{n-(k+1)}{k}$$

Pro samotný výpočet F testu je vhodné si nadefinovat hodnoty, a vypočítat F test v R vzorcem. Druhá možnost je to vyčíst pod příkazem „**summary(proměnná)**“.

```
> n=20
> k=2
> Ftest=(R2/(1-R2))*%*((n-(k+ 1))/k)
> Ftest
      [,1]
[1,] 979.872
```

Hypotézy pro F- test:

$H_0 = R^2$ statisticky nevýznamný

$H_1 = R^2$ statisticky významný

F test se porovná s tabulkovou hodnotou pro k a n-k-1 stupňů volnosti

Tabulková hodnota pro daný příklad je 19,44

Z výše uvedených hodnot vyplývá, že se zamítá nulová hypotéza o nevýznamnosti modelu.

5. Zpracování dat a Kontingenční analýza v programu R

Jednorozměrná data, jsou v programu R popsána vektorem. Můžeme pracovat s kategoriálními, numerickými a diskretními daty. Pro každý druh dat, je manipulace v programu R jiná.

5.1. Kategoriální data

Kategoriální data jsou data, která zachycují nečíselná data. Jedná se o kvalitativní znaky. Příklad na kvalitativní znak může nastat, když tazatel odpoví na otázku kouření „ano“ či „ne“, a nebo daný produkt mu vyhovuje „dobře“, „velmi dobře“, „výborně“. Data řazená do kategorií se píšou do programu R v uvozovkách.

Ukázka užití:

V průzkumu o třídění odpadu, odpovídalo 10 lidí na otázku, zda třídí ve svém okolí odpad. Měli následující odpovědi: ano, ne, občas.

ANO ANO NE OBČAS NE NE ANO ANO OBČAS OBČAS NE

```
>
odpad=c("ANO","ANO","NE","OBČAS","NE","NE","ANO","ANO","OBČAS","OBČAS","N
E")
> table(odpad)
odpad
  ANO  NE OBČAS
    4   4    3
> factor(odpad)
[1] ANO ANO NE OBČAS NE NE ANO ANO OBČAS OBČAS NE
Levels: ANO NE OBČAS
> odpad
[1] "ANO" "ANO" "NE" "OBČAS" "NE" "NE" "ANO" "ANO" "OBČAS"
[10] "OBČAS" "NE"
>
```

Pro procentní zhodnocení dat jsou vhodné následující dvě možnosti:

```
> 100*table(odpad)/(sum(table(odpad)))
odpad
  ANO  NE OBČAS
36.36364 36.36364 27.27273
>
```

nebo

```
> table(odpad)/length(odpad)
odpad
  ANO  NE OBČAS
0.3636364 0.3636364 0.2727273
>
```

5.2. Numerická data

Numerická data jsou data, která číselně zobrazují určité hodnoty. Pro zjištění statistických hodnot v R platí příkazy podobné jako v předchozím příkladě („fivenum“, „summary“ atd.) Rozdíl při použití „fivenum“ a „summary“ tkví v tom, že při sudých hodnotách se doporučuje příkaz „summary“ a při lichých hodnotách „fivenum“.

Ukázka užití:

Na jedné vysoké škole, psalo 10 studentů test z matematiky. Studenti mohli dostat nejméně 0 až 10 bodů. Výsledek byl následující: 4;3,1;0,7;8;5;8,2;2;6,6;8,1;0,9

```
> test=c(4,3.1,0.7,8,5,8.2,2,6.6,8.1,0.9)
> summary(test)
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
 0.700  2.275  4.500  4.660  7.650  8.200
> quantile(test,c(.25,.75))
 25%  75%
2.275 7.650
> quantile(test,.25)
 25%
2.275
>
```

Hodnoty kvantilů se dají zjistit buď dohromady pomocí příkazu **“quantile(proměnná,c(.25,.75))”** nebo jednotlivě příkazem **“quantile(proměnná,.25)”**.

Hodnota rozpětí mezi kvartily se vypočítá pomocí příkazu **„IQR(proměnná)“**. Hodnotu mediánu průměrné odchylky zjistíme pomocí příkazu **„mad(proměnná)“** nebo ručním výpočtem v R.

```
> IQR(test)
[1] 5.375
> mad(test)
[1] 4.4478
```

nebo

```
> median(abs(test-median(test)))*1.4826
[1] 4.4478
>
```

5.3. Práce s kontingenčními tabulkami v R

Kontingenční tabulky slouží ke srovnání dvou napozorovaných statistických znaků. Řádky v kontingenční tabulce souvisí s první napozorovanou hodnotou a sloupce souvisí s druhou napozorovanou hodnotou. Typ kontingenční tabulky se hodnotí podle počtu řádků (r) a sloupců (s).

5.3.1. Dvojměrná kontingenční tabulka

Dvojměrná kontingenční tabulka je například čtyřpolní tabulka ve tvaru 2×2 . Tabulku s číselnými daty zadáváme do R pomocí příkazu „**matrix(c(hodnoty))**“. Pomocí daného příkazu jsou data maticí.

Ukázka užití:

Byla dána anketa, jestli lidé věří v posmrtný život nebo nevěří. Bylo dotazováno 1000 respondentů. Z toho věří 300 žen a 200 mužů, a nevěří 100 žen a 400 mužů.

Data v daném případě jsou pojmenována proměnnou život. Při zavedení proměnné a její hodnot do systému se použije příkaz „**matrix(c(hodnoty))**“ a syntaxi „<-“, která naznačuje R, že hodnoty k proměnné patří. Uvedené syntaxe za příkazem „**matrix(...)**“ naznačují, že tabulka bude typu 2×2 ($nrow=2$) a že zadané hodnoty se budou řadit řádkově ($byrow=TRUE$).

```
> život<-matrix(c(300,100,200,400),nrow=2,byrow=TRUE)
> život
  [,1] [,2]
[1,] 300 100
[2,] 200 400
> dimnames(život)<-list(c("ženy","muži"),c("ano","ne"))
> život
  ano ne
ženy 300 100
muži 200 400
> names(dimnames(život))<-c("pohlaví","odpověď")
> život
  odpověď
pohlaví ano ne
  ženy 300 100
  muži 200 400
>
```

Příkaz „**dimnames(proměnná)**“ pojmenovává různé části tabulky v případě složeného příkazu a u jednoduchého příkazu, jak je výše uvedené, pojmenovává název celé tabulky. Za příkazem „**dimnames(proměnná)**“, musí uživatel napsat „<-“, Touto syntaxí dává najevo R, že k danému pojmenování patří další část. Po této tzv. šipce následuje příkaz „**list(c(proměnné),c(proměnné))**“ která vede R ke konstrukci tabulky. Po samotném napsání proměnné život do příkazového řádku, se zobrazí potřebná tabulka.

Pro nadefinování proměnných v tabulce na dílčí proměnné, použije uživatel příkaz „**names(dimnames())**“. Daný příkaz je tzv. složený příkaz, který dokáže pojmenovat hodnoty nad proměnnými. V uvedeném příkladě vidíme, že udává proměnnou (muži, ženy) *pohlaví* a proměnnou *odpověď* (ano,ne).

Výpočet celkové velikosti vzorku (p_{ij}) se provádí příkazem „**sum(proměnná)**“. V níže nadefinovaném příkladě je celkový počet vzorku pojmenovaný proměnnou *celkem*.

```
> celkem<-sum(život)
> celkem
[1] 1000
> život/celkem
  odpověď
pohlaví ano  ne
  ženy 0.3 0.1
  muži 0.2 0.4
>
```

Z následujících syntaxí bylo zjištěno, že je celkem 1000 dotazovaných. Zjistilo se také procentuální zastoupení odpovědí obou pohlaví a odpovědí na celkovém počtu dotazovaných.

```
> počet1<-apply(život,1,sum)
> počet2<-apply(život,2,sum)
> počet1
ženy muži
 400 600
> počet2
ano ne
500 500
```

Pro další kumulativní výpočet proměnných se použije následující příkaz se syntaxí (která říká, jak již bylo výše uvedeno, že daná proměnná patří k následujícímu příkazu) „**<-**“ a příkaz „**apply(proměnná,1 nebo 2,sum)**“. Tento příkaz říká programu R, že má použít hodnoty dané proměnné (v daném příkladu je proměnnou *život*) za závorkou, a číslo 1 označuje řádky a číslo 2 sloupce, které pomocí konečného příkazu „**sum**“ mají být sečteny.

Po samotném napsání proměnných (*počet1,počet2*) do příkazového řádku, se objeví hodnoty kumulativně sečtené.

```
> kumproc<-sweep(život,1,počet1,"/")
> kumproc
  odpověď
pohlaví  ano    ne
  ženy 0.7500000 0.2500000
  muži 0.3333333 0.6666667
```

Dobrý
výpočet

```
> kumproc<-sweep(život,2,počet1,"/")
> kumproc
      odpověď
pohlaví ano      ne
  ženy 0.75 0.1666667
  muži 0.50 0.6666667
```

Špatný
výpočet

Prostřednictvím minulých výpočtů a příkazu „**sweep**(*proměnná*,*1*,*proměnná*,“/“) R vypočítá výše uvedené hodnoty. Příkaz „**sweep**“ je zajímavý svým složením. První část za závorkou je *proměnná*, v našem případě celá tabulka *život*. Číslo 1 a další *proměnná* (v našem případě tj.*počet1*) zaznamenává, že vypočtené hodnoty budou podílem po řádcích. Tento systém dává i logické vysvětlení, neboť kdyby se napsalo číslo 2, program R by začal počítat po sloupcích. Jelikož v daném příkladu výpočet po sloupcích je nelogický, musíme zde uvést číslo 1. V jiných příkladech by číslo 2 mohlo určitě figurovat. A nakonec poslední část, je syntaxe “/“, která k příkazu patří, neboť je to binární hodnota.

Z vypočtených hodnot vyčteme, že 75% žen z celkové populace dotazovaných žen věří na posmrtný život a 25% nevěří, a 33,3% mužů z celkového počtu dotazovaných věří a 66,7% nevěří.

```
> round(kumproc,3)
      odpověď
pohlaví ano      ne
  ženy 0.750 0.250
  muži 0.333 0.667
```

Příkaz „**round**(*proměnná*,*3*)“ zaokrouhlí hodnoty v tabulce v našem případě na tři desetinná čísla. Číslo za *proměnnou*, naznačuje programu R, že má dané data zaokrouhlit na tři desetinná čísla.

```
> mumi<-sweep(život,2,počet2,"/")
> mumi
      odpověď
pohlaví ano      ne
  ženy 0.6 0.2
  muži 0.4 0.8
```

Zde je podobný případ jak výše uvedený. Jsou zde procentuálně vypočítané hodnoty přes sloupce, závislé na *proměnné počet2*. Uživatel může rovnou napsat příkaz „**sweep**()“, bez pojmenování další *proměnné*(u nás *mumi*).

```
>sweep(život,2,počet2,"/")
      odpověď
pohlaví ano      ne
  ženy 0.6 0.2
  muži 0.4 0.8
```

5.3.2. Vícerozměrná kontingenční tabulka

Ve vícerozměrné kontingenční tabulce se sleduje více napozorovaných znaků než dva. Vícerozměrné kontingenční tabulky se tvoří podobným způsobem jako u dvourozměrných, ale znázornění je docela obtížné.

Ukázka užití:

Mějme 8 měst. V každém městě bylo provedeno šetření mezi populací, zda kouří nebo ne a jestli byla u nich nalezena vyšetřujícím lékařem rakovina. Šetření se provádělo ve velkých městech celého světa. Neboť zde byl velký předpoklad nalezení více kouřících lidí s rakovinou. Respondent na každou otázku zvlášť (kouří, má rakovinu) mohl odpovědět pouze „ano“ nebo „ne“.²⁶

Pro zadání daných hodnot do programu je příkaz „`c(„proměnné“)`“ nejvhodnější způsob.

```
> města<-c("beijing","Shang","SHEy","Nanji","Harbin","Zhengu","Taig","Nanchi")
> města
[1] "beijing" "Shang" "SHEy" "Nanji" "Harbin" "Zhengu" "Taig"
[8] "Nanchi"
```

Níže popsaný příkaz udává pokyn programu, že proměnná *města* a její hodnoty se mají 4x po sobě opakovat. Příkaz „`factor(proměnná)`“ se používá pro zakódování vektoru. Další složené příkazy uvnitř závorčky objasňují opakování každé hodnoty proměnné města 4x.

Celý složený příkaz „`factor(rep(proměnná,rep(4,length(proměnná))),levels=města`“ obsahuje příkaz „`rep ()`“ tzn. že říká programu R, že má hodnoty v závorce opakovat.

Uvnitř příkazu „`rep ()`“, je uvedena nejprve proměnná, jejíž hodnoty má opakovat (*města*). Pro větší konkrétnost se musí napsat opět příkaz „`rep ()`“, který naznačuje další podmínku opakování. Jádro příkazu obsahuje, kolikrát se daná proměnná má opakovat (4) a délku souboru hodnot proměnné (*města*). Nakonec se píše „Level“ proměnné. Při samotném napsání proměnné (*město*) do příkazového řádku, se objeví výsledek výše popsaného složeného příkazu.

```
> město<-factor(rep(města,rep(4,length(města))),levels=města)
> město
[1] beijing beijing beijing beijing Shang Shang Shang Shang SHEy
[10] SHEy SHEy SHEy Nanji Nanji Nanji Nanji Harbin Harbin
[19] Harbin Harbin Zhengu Zhengu Zhengu Zhengu Taig Taig Taig
[28] Taig Nanchi Nanchi Nanchi Nanchi
Levels: beijing Shang SHEy Nanji Harbin Zhengu Taig Nanchi
>
```

Zadaný příklad obsahuje také dotazy na respondenta ohledně kouření a rakoviny. Aby tabulka byla kompletní, slouží k tomu následující dva složené příkazy, jak ke kouření, tak k otázce rakovina. Pro zadání hodnot „ano“ a „ne“ se použije podobná složená podmínka, která byla

²⁶ An Introduction to Categorical Data Analysis Using R, Brett Presnell, March 28 2000

výše popsaná. Malou změnou v následující podmínce, je pouze „(.....**c(2,2)**),8)“. Tento drobný rozdíl, říká programu R, že má opakovat 2x ano a 2x ne a to vždy po sobě v 8 hodnotách. Konec složeného příkazu logicky ukončuje tzv. „Level“, který je v tomto případě jiný.

Pro odpověď u otázky související s rakovinou, je v její složené podmínce pouze číslo 16. Znamená to, že odpovědi „ano“ a „ne“ bude opakovat po sobě 16x, a zase v 17 hodnotě začne znovu 16x opakovat, až se dostane na konec

```
> kouří<-factor(rep(rep(c("ano","ne"),c(2,2)),8),levels=c("ano","ne"))
> kouří
 [1] ano ano ne ne ano ano ne ne ano ano ne ne ano ano ne ne ano ano ne
 [20] ne ano ano ne ne ano ano ne ne ano ano ne ne
Levels: ano ne
> rakovina<-factor(rep(c("ano","ne"),16),levels=c("ano","ne"))
> rakovina
 [1] ano ne ano ne ano ne ano ne ano ne ano ne ano ne ano ne ano
 [20] ne ano ne ano ne ano ne ano ne ano ne ano ne
Levels: ano ne
>
```

Hodnoty, které byly v dotazníkovém šetření zjištěny, zavedeme do systému. Pomocí známého příkazu „**c()**“ a názvem proměnné „*data*“. Již jsou všechny hodnoty zavedeny v systému, stačí jen udělat tabulku. Pro tabulku použijeme příkaz „**data.frame(proměnná,proměnná,proměnná,proměnná)**“. Tabulka byla nazvána „*tab1*“ a po samotném napsání do příkazového řádku této proměnné se udělá následující tabulka (obr. 30).

```
> data<-
c(126,100,35,61,908,688,497,807,913,747,336,598,235,172,58,121,402,308,121,215,
182,156,72,98,60,99,11,43,104,89,21,36)
```

```

> tab1<-data.frame(město,kouří,rakovina,data)
> tab1
  město kouří rakovina data
1  beijing  ano      ano  126
2  beijing  ano      ne   100
3  beijing  ne       ano   35
4  beijing  ne       ne    61
5    Shang  ano      ano  908
6    Shang  ano      ne  688
7    Shang  ne       ano  497
8    Shang  ne       ne  807
9    SHEy   ano      ano  913
10   SHEy   ano      ne  747
11   SHEy   ne       ano  336
12   SHEy   ne       ne  598
13  Nanji   ano      ano  235
14  Nanji   ano      ne  172
15  Nanji   ne       ano   58
16  Nanji   ne       ne  121
17  Harbin  ano      ano  402
18  Harbin  ano      ne  308
19  Harbin  ne       ano  121
20  Harbin  ne       ne  215
21  Zhengu  ano      ano  182
22  Zhengu  ano      ne  156
23  Zhengu  ne       ano   72
24  Zhengu  ne       ne   98
25   Taig   ano      ano   60
26   Taig   ano      ne   99
27   Taig   ne       ano   11
28   Taig   ne       ne   43
29  Nanchi  ano      ano  104
30  Nanchi  ano      ne   89
31  Nanchi  ne       ano   21
32  Nanchi  ne       ne   36

```

Obrázek 30: Tab 1

Pro větší přehlednost je možnost zavedení i jiné tabulky. Prostřednictvím tří níže uvedených složených příkazů se zobrazí následující tabulka (obr.31).

```

> x<-tapply(data,list(kouří,rakovina,město),c)
> names(dimnames(x))<-c("kouří","rakovina","město")

```

```
> ftable(x, row.vars=c("město", "kouří"), col.vars="rakovina")
      rakovina ano  ne
město  kouří
beijing ano      126 100
      ne        35  61
Shang   ano      908 688
      ne        497 807
SHEy    ano      913 747
      ne        336 598
Nanji   ano      235 172
      ne         58 121
Harbin  ano      402 308
      ne        121 215
Zhengu  ano      182 156
      ne         72  98
Taig    ano       60  99
      ne         11  43
Nanchi  ano      104  89
      ne         21  36
> |
```

Obrázek 31: f-table

Níže uvedené příkazy souvisí pro členění vrstev hodnot proměnných. Dané separované tabulky jsou vhodné pro různé testy v kontingenční tabulce např. test CMH.²⁷

```
> ni.k<-apply(x,c(1,3),sum)
> ni.k
      město
kouří beijing Shang SHEy Nanji Harbin Zhengu Taig Nanchi
ano   226 1596 1660 407  710  338 159  193
ne    96 1304  934 179  336  170  54  57

> n.jk<-apply(x,c(2,3),sum)
> n.jk
      město
rakovina beijing Shang SHEy Nanji Harbin Zhengu Taig Nanchi
ano      161 1405 1249 293  523  254  71  125
ne       161 1495 1345 293  523  254 142  125
>
```

²⁷ Cochranovy-Mantelovy-Haenszelovy statistiky

Závěr

Cílem této bakalářské práce bylo seznámení s programem R a jeho využitím v konkrétních situacích ve statistice. Přesné vymezení cílů, které mi pomohli zrealizovat tuto těžkou problematiku, byly již v úvodu nastíněny. Výsledek předložených cílů se pokusím zodpovědět.

První cíl byl popsat základy regresní a kontingenční analýzy. Uznala jsem za vhodné, uvést různé typy regresního modelu a jeho grafické zobrazení. V dalších sub-kapitolách jsem pouze nastínila předpoklady použití regresní analýzy funkce a výpočet odhadů regresních funkcí. U kontingenční analýzy, jsem uvedla typy proměnných a jejich vlastnosti.

Druhý cíl byl jádrem práce. Jak jsem již uvedla, prostředí programu R je velmi obtížné pochopit. Program R pracuje v prostředí různých příkazů a psaní syntaxe do příkazového řádku. V třetí kapitole, jsou uvedeny základy práce s programem R ve statistice a jeho technické zajištění. Základy jsem pojala v rámci zadávání dat do programu a práce s nimi (například: tabulky, grafy). Výhody programu R jsou studijní a ekonomické. V rámci studia, uživatel důkladně porozumí statistické látce, neboť ji zde vidí z jiného pohledu, než u klasických statistických programů například u SASu. Ekonomická výhoda je prostá. Program R je volně přístupný na webovém portálu, a proto se zde nemusíme zabývat koupí licencí, jak u ostatních programů. Nevýhodou může být uživatelův čas a nevědomosti. Neboť R není program se snadnou manipulací. Skoro veškeré informace o programu R bývají většinou v cizím než v českém jazyce. Dále může být velká nevýhoda pro uživatele, který není orientovaný na programování, protože si program R nemůže naprogramovat pro své potřeby a je odkázán na již naprogramované balíčky.

Třetí cíl, spočíval v aplikaci regresní a kontingenční analýzy v programu R. U regresní analýzy, jsem uvedla dvě možnosti způsobů výpočtu regresních koeficientů, které R nabízí. Protože je zde velká škála možností, určitě by se daly nalézt i další způsoby výpočtu. Při aplikaci kontingenční analýzy v programu R, jsou v páté kapitole uvedené různé možnosti práce s kontingenčními tabulkami. Kontingenční analýza je tak rozsáhlá, že do této práce jsem uvedla pouze práci s kontingenčními tabulkami.

Literatura

- [1] SEGER, Jan; HINDLS, Richard. *Statistické metody v tržním hospodářství*. Praha 1 : VICTORIA PUBLISHING , a.s., 1995. 435 s. ISBN 80-7187-058-7
- [2] ŘEZÁNKOVÁ, Doc.Ing.Hana. *Analýza kategoriálních dat*. 2005. Praha : Oeconomica, 2005. 99 s. ISBN 80-245-0926-1
- [3] HEBÁK, Petr. *Regrese : I.část*. Praha : Ediční oddělení VŠE Praha, 1998. 138 s. ISBN 80-7079-909-9
- [4] *Biostatistics:A foundation for analysis in the health science*, Wayne W.Daniel, ISBN: 0-471-52514-6
- [5] Hindls, R. – Hronová, S. – Seger, J. – Fischer, J. (2006): *Statistika pro ekonomy*. 7. vyd. Praha: Professional Publishing. 418 str. ISBN 80-86946-16-9
- [6] Marek, L. a kol. (2005): *Statistika pro ekonomy – aplikace*. 1. vyd. Praha: Professional Publishing. 423 str. ISBN 80-86419-68-1
- [7] Pecáková, I. (2008): *Statistika v terénních průzkumech*. 1. vyd. Praha: Professional Publishing. 231 str. ISBN 978-80-86946-74-0

Internetové zdroje

- [8] <http://www1.lf1.cuni.cz/~ldohna/linear/index.htm>
- [9] <http://www.esphere.cz/kostka/Matematika/Funkce/specifikace.htm>
- [10] <http://homen.vsb.cz/~oti73/cdpast1/KAP09/KAP09.HTM>
- [11] <http://www.karlin.mff.cuni.cz/~kulich/vyuka/Rdoc/index.html>
- [12] <http://www.gardenersown.co.uk/Education/Lectures/R/regression.htm>
- [13] http://cs.wikipedia.org/wiki/Regresn%C3%AD_anal%C3%BDza
- [14] www.montefiore.ulg.ac.be/~kvansteen/FGBIO0009-1/Fac20092010/Class8/Using%2520R%2520for%2520linear%2520regression.pdf
- [15] www.cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf
- [16] www.cran.r-project.org/doc/manuals/R-data.pdf
- [17] www.stat.ufl.edu/~presnell/Courses/sta4504-2000sp/R/R-CDA.pdf

Seznam obrázků

Obrázek 1: Typy statistických závislostí	8
Obrázek 2: Přímková funkce	9
Obrázek 3: Parabolická funkce	9
Obrázek 4: Hyperbolická funkce	10
Obrázek 5: Exponenciální funkce	10
Obrázek 6: Logaritmická funkce	10
Obrázek 7: Ukázka Vyrovnání hodnot.....	11
Obrázek 8: Ukázka programu R.....	18
Obrázek 9: Sloupcový graf.....	22
Obrázek 10 Sloupcový graf.....	22
Obrázek 11: Sloupcový graf.....	23
Obrázek 12: Sloupcový graf.....	23
Obrázek 13: Horizontální sloupcový graf	23
Obrázek 14: Kruhový graf.....	24
Obrázek 15:Kruhový graf.....	24
Obrázek 16: Kruhový graf.....	25
Obrázek 17: Boxplot 1 Obrázek 18:Boxplot 2	25
Obrázek 20: Boxplot 4	26
Obrázek 21: Histogram.....	26
Obrázek 19: Boxplot 3	26
Obrázek 22: Barevný histogram	27
Obrázek 23: Regrese	34
Obrázek 24: Lineární regrese	34
Obrázek 25: Regrese	39
Obrázek 26: Grafické zobrazení vícenásobné regrese	40
Obrázek 27: Postup-Grafického zobrazení 1	40
Obrázek 28: Postup-Grafického zobrazení 2	40
Obrázek 29: Postup-Grafického zobrazení 3	41
Obrázek 30: Tab 1.....	52
Obrázek 31: f-table.....	53