DATUM V R

- import dat do R
  - `as.Date()` – nepracuje s časem
  - `chron()` – pracuje i s časem; library `chron`
- formáty:
  - `character`
  - `numeric`
  - `POSIXlt`
  - `POSIXct`

- POZN.: `as.Date()` a `chron()` nepracuje s kontrolou časových zón, typy proměnných (class) `POSIXlt` a `POSIXct` ano.

  - `Sys.time()` – funkce vrátí systémový čas

- `POSIXct`
  - počet vteřin od počátku 1. ledna 1970. Záporná hodnota reprezentuje počet sekund před tímto datem a kladná hodnota reprezentuje počet sekund za tímto datem.

- `POSIXlt`
  - list (!)
  1. sekundy
  2. minuty
  3. hodiny
  4. den měsíce (1 – 31)
  5. měsíc (0 – 11, 11 = listopad)
  6. rok od 1900 (tzn. pro 2018 je to 118tý rok)
  7. den týdne (0 - 6, 0 = neděle)
  8. den roku (0 - 365)
  9. daylight savings indicator (positive if it is daylight savings)

OBECNÉ FORMÁTY DATA

- s výjimkou formátu (class/třídy) `POSIXlt` jsou data uložena jako počet dnů, nebo sekund od určitého referenčního data
- tzn. data budou mít obecně **numerický** režim (class/třídu)
  - formát `POSIXlt` ukládá hodnoty jako seznam komponent

NOTE:

- pro antropologická data je ve většině případů dostačující formát `as.Date()`, resp. bez konkrétního času, pouze datum (den-měsíc-rok, rok-měsíc-den,...)

- eventuálně může být dostačující datum pouze jako numerická proměnná, nebo faktor
  - datum výzkumu atp.

`as.Date()` funkce

- parametr `format = _____`
- výchozí nastavení je v podobě rok (4 cifry), měsíc a poté den
  - oddělené pomocí pomlček nebo lomítka

Kód	Hodnota/význam	Příklad
`%d`	Den měsíce; 0 - 31	01 – 31
`%m`	Měsíc (číslo); 00 - 12	00 - 12
`%a`	Den (zkratka názvu)	Mon
`%A`	Den (celý název	Monday
`%b`	Měsíc (zkratka názvu)	Jan
`%B`	Měsíc (celý název)	January
`%y`	Rok (2 cifry)	18
`%Y`	Rok (4 cifry)	2018

Pozn.: %y je závislé na systému, používat opatrně!

```
> as.Date("13/3/1992", format = "%d/%m/%Y")
[1] "1992-03-13"
> as.Date("Březen 03, 1992", format = "%B %d, %Y")
[1] "1992-03-03 "
```

IMPORT DAT NUMERICKÉHO FORMÁTU

- **Excel**
  - pro data po roce 1900 je počáteční datum: 30. prosince 1899
    - *Note: není to 31. prosince 1899, protože návrháři Excelu se domnívali, že se jedná o přestupný rok, nejednalo.*

```
> data <- c(17431, 15349)
> data2 <- as.Date(data, origin = "1899-12-30")
> data2
[1] "1947-09-21" "1942-01-08"
```

- **Excel on Mac**
- pro data po roce 1900 je počáteční datum: 1. ledna 1904

```
> data <- c(17426, 38540)
> data3 <- as.Date(data, origin = "1904-01-01")
> data3
[1] "1951-09-17" "2009-07-08"
```

Formát datum	Formát číslo
21. 9. 1947	17431
8. 1. 1942	15349

- Note: obecně se doporučuje mít data v takové podobě, aby s nimi bylo možné ihned pracovat. Tzn. pozor na formát!

Změna formátu načteného data

- funkce `format()`
- načtený vektor dat změní dle nově zadaného typu formátu

```
> rok_narozeni
[1] "1856-07-10" "1892-01-03" "1948-06-21"
> format(rok_narozeni, "%a %b %d")
[1] "čt VII 10" "ne I 03" "po VI 21"
> format(rok_narozeni, "%A %B %d %Y")
[1] "čtvrtek červenec 10 1856" "neděle leden 03
  1892" "pondělí červen 21 1948"
```

- *Note: pozor zda máte mirror czech, pokud máte prostředí anglické musíte používat anglické názvy dní, měsíců apod.*

Důvod správného formátování

- V případě, že máme fatum korektně vloženo do R, lze s ním dále pracovat.

```
> min(rok_narozeni)
[1] "1856-07-10"
> max(rok_narozeni)
[1] "1948-06-21„

> rok_narozeni
[1] "1856-07-10" "1892-01-03" "1948-06-21"
> plus <- rok_narozeni + 5
> plus
[1] "1856-07-15" "1892-01-08" "1948-06-26"
```

Datové formáty - obecně

- V antropologii ve specifických případech.
- Vždy s nimi pracovat velmi OPATRNĚ!
- Časté chyby
  - v převodu mezi jednotlivými časovými zónami,
  - chybné nastavené origin date,
  - pohybujeme se v jiné zóně než data, apod.

DOPLŇUJÍCÍ INFORMACE

- Cole, B. (2012): *Handling date-times in R*.

[http://biostat.mc.vanderbilt.edu/wiki/pub/Main/ColeBeck/datestimes.pdf]

- University of California, Berkeley, Department of Statistics

[https://www.stat.berkeley.edu/~s133/dates.html]

DOMÁCÍ ÚKOL 12

- A) Vytvořte objekt `roky`, do kterého uložíte data narození Vámi zvolených pěti významných osobností.
- B) Zjistěte průměrný rok objektu `roky`.
- C) Objekt roky následně upravte za pomoci funkce `format()` tak, abyste vytvořili vektor dní narození zmíněných pěti osobností. Výsledný vektor dny, bude obsahovat **POUZE** dny v týdnu.

Formát dat pro načtení do R

- nejčastěji ve formátu `.csv` nebo `.txt`
- formáty `.csv` nebo `.txt` generujeme nejčastěji z Excelu, pokud nepoužíváme jiný software na generaci dat

ZÁSADY TVORBY TABULEK

- **přehledně**
- **srozumitelně**
- neslučujeme buňky
  - nejlépe zdržíme-li se excesivního formátování buněk
- nepropojujeme navzájem listy Excelu
  - nejlépe jsou-li všechna data na jednom listu
- názvy proměnných volíme vhodným způsobem s ohledem na typ dat

Datové tabulky

- **než začneme datovou tabulku vytvářet:**
- 1) jaká máme data
  - sin vs. dx, males vs. females, faktorové proměnné, souřadnice...
- 2) pro jakou analýzu data chystáme?
  - výpočet základních statistik,
  - zobrazení,
  - korelační analýza, ...
- 3) pozor na typy formátů buněk (zejména Excel)
  - používáme datum?
  - používáme čísla?
  - české vs. anglické prostředí
    - Note: použití vhodných názvů proměnných

NEPOUŽÍVAT DIAKRITIKU!

- 4) Máme neexistující/nulové hodnoty?
  - nejčastěji v případě, kdy není možné změřit rozměr, tzn. NA hodnoty
  - dopředu si „nechystejte“ hodnoty buněk (zejména používáme-li funkce Excelu)
    - pokud používáme funkce Excelu nezapomene **vzorce ODSTRANIT**!
- 5) Pozor na jednotky
  - zejména v případě měření se doporučuje mít všechny proměnné stejnými jednotkami (vše např. v mm)
- 6) Pozor na desetinnou čárku vs. tečku
- 7) Pozor na mezeru v prázdných buňkách
- 8) Sporné hodnoty (odhad, nepřesné měření) – vůbec neuvádíme, nebo vytvoříme novou proměnnou, která definuje přesnost proměnných, NIKDY nedopisujeme poznámky k numerických proměnným do té SAMÉ buňky!

- 9) Nejčastěji volíme strukturu tak, že proměnné představují sloupce a případy představují řádky
- 10) Nedoporučuje se používat nestandartní symboly pro kategorické proměnné
  - Ideálně se vyvarovat používá symbolů jako *, +, -, /, atp.
  - V případě duální kategorické proměnné používáme ideálně 0 a 1

PŘIPOMENUTÍ

- R je **case sensitive !!!**
  - `A` a `a` jsou odlišné symboly
- Výrazy/proměnné nezačínáme ‘ **.** ’ , ‘ _ ’ ani cifrou
- Výrazy/proměnné nenazýváme již existující názvy funkcí!
  - Názvy objektů: F, T, c apodobně také ne.

Nápověda

- nápověda přímo v R
  - Případně k jednotlivým balíčkům jejich dokumentace
- https://stackoverflow.com/
- https://www.r-bloggers.com/
- https://www.statmethods.net/index.html
- Department of Biostatistics, Vanderbilt University: http://biostat.mc.vanderbilt.edu/wiki/Main/ProgrammingTipsForStatisticians
- University of California, Berkeley, Department of Statistics, **Concepts in Computing with Data**, Phil Spector (2011):
- https://www.stat.berkeley.edu/~s133/all2011.pdf

- chybová hlášení čteme! - případně chybu můžeme zadat do vyhledávače Googlu
  - stejně tak čteme i varovná hlášení!

NEZAPOMÍNÁME

- vše průběžně kontrolujeme (zejména načtení dat!)
- je vhodné místy promazat globální prostředí
  - můžeme si přepsat objekty,
  - je možné, že pracujeme s chybně vytvořeným/načteným objektem
- vypnout a zapnout RSTudio
  - zejména v případě problémů s balíčky
- kód vhodně doplňujeme poznámkami
  - **Pravidlo 1**: tak abych věděl/a po půl roce, co jsem dělal/a
  - **Pravidlo 2**: tak aby to pochopil i někdo jiný
- Snažíme se kód zbytečně neopakovat
  - **Pravidlo**: Don‘t Repeat Yoursel (DRY)

    ale
  - **Pravidlo**: Be Clear and Correct First; Fast and Clever Second

Balíčky v R

- Funkce `install.packages()`
  - Instalujeme – „stáhneme“ – balíček do adresáře R (ve většině případů win-library)
  - Balíčky stahujeme z CRAN (=> nutné připojení k internetu)
- Funkce `librabry()`
  - V projektu (skriptu) „řekneme“ R, že budeme pracovat s funkcemi, které jsou obsaženy v tomto balíčku.
  - Je možné v Rstudiu nastavit, které balíčky se mají spouštět automaticky společně se základní sadou <- NEDOPORUČUJE SE!
    - Balíčky se aktualizují
    - Zapomeneme na to a zapomeneme balíčky citovat! => necitujeme POUZE základní sadu.
- Balíčky je nutné v některých případech aktualizovat.

POZNÁMKY K BALÍČKŮM

- Funkce `install.packages()` má četné množství argumentů, př.:
  - `dependencies = TRUE`
    - výchozí nastavení je `dependencies = FALSE`
    - Pokud TRUE pak se automaticky nainstalují i balíčky na kterých,právě instalovaný balíček závisí
  - `type = "win.binary"`
    - Většinou není třeba zadávat (výjimkou Mac users)

- Obecně se doporučuje začít funkcí: `install.packages(„název_balíčku")` a teprve po tom, co se objeví problém, zjišťovat kde – R nám to zpravidla poví!