

Poslední úprava dokumentu: 19. února 2025.

Seznámení s R

1 Úvod

- 1) Postup prací pro cvičení se bude postupně objevovat na webu:

<https://web.natur.cuni.cz/uamvt/turcm6am>

- 2) V počítači si vytvořte složku k tomuto cvičení, v tomto textu ji nazveme např. **matstat**.
- 3) Do složky **matstat** zkopírujte složku **data** a soubor **MScv01.R**, které se nacházejí v mé složce **V:/turcicova/MatStat_cv** na disku V.
- 4) Data k dnešnímu cvičení a soubor **MScv01.R** najdete též na mé webové stránce (viz adresa výše) u materiálů k 1. cvičení.
- 5) Spusťte RStudio (stručný návod na jeho instalaci najdete na stránkách výše v záložce "Odkazy").
- 6) V RStudio nastavte coby pracovní adresář složku **matstat**, a to jedním z následujících způsobů:

- (a) Napište a pomocí Enter odešlete v okně Console příkaz

`setwd("popis_cesty/matstat")`

kde **popis_cesty** nahraděte za popis cesty ke složce **matstat**, který bude vypadat přibližně jako **J:/prijmeni/matstat**. (Lomítka v popisu cesty musejí být dopředná, tj. **"/"**, nikoli zpětná.)

- (b) V horní nabídce postupně zvolte

Session ➔ **Set Working directory** ➔ **Choose directory...**

a pomocí myši najděte složku **biostat**.

- 7) Zjistěte, zda se vše povedlo. Po zadání příkazu

`getwd()`

by se mělo vypsat **popis_cesty/biostat**.

- 8) Pomocí horní nabídky zaveděte do skriptového okna (alias **Script Window**, vlevo nahoře) skriptový soubor **MScv01.R**:

File ➔ **Open File...**

Několik poznámek k práci se skriptovým souborem

- ❖ Při psaní příkazů v **Script Window** jsou vám automaticky nabízeny příkazy začínající na daná písmena. Pokud se vám některý z těchto příkazů zamlouvá, vyberte ho pomocí šipky na klávesnici a poté potvrďte klávesou **Enter**.
- ❖ Znak # odděluje poznámky, tj. něco, co chcete v souboru vidět vy, ale nechcete, aby to vidělo R.
- ❖ Různé příkazy musí být na různých řádcích (doporučují), nebo na stejném řádku oddělené středníkem.
- ❖ Když chceme, aby se provedl příkaz na konkrétním řádku, nastavíme kurzor na příslušný řádek a zmáčkneme klávesovou zkratku **Ctrl-Enter**, nebo klikneme na tlačítko **Run** (v pravém horním rohu **Script Window**).
- ❖ Když chceme, aby se provedlo více příkazů najednou, označíme je myší jako blok a zmáčkneme klávesovou zkratku **Ctrl-Enter**, nebo klikneme na tlačítko **Run** (v pravém horním rohu **Script Window**).
- ❖ Čas od času si (doplňovaný) skriptový soubor uložte (stačí sem tam stisknout klávesovou zkratku **Ctrl-S**) nebo kliknout na ikonu diskety vlevo nahore.
- ❖ Nápovědu k libovolnému příkazu vyvoláme pomocí **help(prikaz)** nebo **?prikaz**. (Nápověda se pak zobrazí v pravém dolním okně.)

Další poznámky pro práci s R / RStudiem

- ❖ Všechny požadované příkazy je nutné napsat do **Script Window** nebo do **Console**.
- ❖ Po odeslání požadovaného příkazu se příkaz v tichosti provede a v **Console** se objeví výsledky výpočtu (jsou-li nějaké). V případě chybného příkazu se objeví červená chybová hláška.
- ❖ Předchozí příkazy je možné v **Consoli** vyvolávat pomocí ”šipky nahoru” na klávesnici. Vyvolaný příkaz lze dále upravit a odeslat klávesou **Enter**.
- ❖ Chceme-li provést ve výpočtu drobnou změnu (např. zpracovat proměnnou **vaha** místo proměnné **vyska**), můžeme si příslušnou část ve skriptovém okně zkopirovat níže, upravit a odeslat.
- ❖ Pro pozdější použití vytvořeného skriptu je dobré si do něj sem tam napsat nějaký komentář (za znak #).

2 Základní operace a funkce v R

- 1) Nejprve použijte R jako kalkulačku a spusťte postupně ze skriptového okna následující příkazy:

```
2  
2 + 3          # scitani  
2 - 3          # odcitani  
2 * 3          # nasobeni  
2 / 3          # deleni  
2^3            # mocnina  
3^(1/2)         # (od)mocnina  
sqrt(3)         # jeste jednou odmocnina  
exp(3)          # exponenciela (e na neco)  
log(3)          # prirozeny logaritmus (pozor! jinde se casto znaci ln)  
exp(log(3))  
sin(3)          # sinus  
cos(3)          # cosinus  
pi              # konstanta pi  
sin(pi/2)  
cos(pi/2)  
factorial(3)    # 3! (faktorial)  
choose(3, 2)     # "3 nad 2 (kombinacni cislo)"
```

- 2) Nyní budeme pracovat s proměnnými

```
a <- 2          # ulozeni hodnoty do promenne  
a = 2           # jiny zpusob ulozeni (nelze kombinovat s jinymi prikazy, viz nize)  
a               # podivame se, co tam name  
b <- 3           # ulozeni do jine promenne  
b  
a + b           # secteni ulozenyh hodnot  
a - b           # odecteni ulozenyh hodnot  
a * b           # vynasobeni ulozenyh hodnot  
a / b           # vydeleni ulozenyh hodnot
```

- 3) Ukládání výsledku do dalších proměnných

```
c <- log(b)      # ulozi, ale nevytiskne vysledek  
c                 # vytiskne vysledek  
print(c)          # tez vytiskne vysledek
```

- 4) Vytvoření vektoru hodnot a základní aritmetické operace s vektorem

```
v <- c(6,4,8,2,1,2)    # vytvoří vektor hodnot a uloží ho do promenné "v"  
v                  # vytiskne vysledek  
v+1                # ke všem položkám vektoru "v" přičte 1  
v*3                # všechny položky vektoru "v" vynásobi třema  
v^2                # všechny položky vektoru "v" umocní na druhou  
sum(v)              # sečte všechny položky vektoru "v"
```

Označme si náš vektor výše jako $v = (v_1, v_2, v_3, \dots, v_6)$. Za pomoci funkce `sum` a základních matematických operací spočítejte

$$3 \cdot \sum_{i=1}^6 v_i^2 \quad \text{a dále} \quad 3 \cdot \left(\sum_{j=1}^6 v_j \right)^2.$$



5) „Úklid“ (před začátkem jiné práce)

```
ls()          # zjistime, jaké promenne mame na definovane  
rm(list=ls()) # vycistime Rko (vsechny promenne se smazou)  
ls()          # zjistime, jaké promenne mame nactene (snad nic)
```

6) Zkontrolujte si, zda vám zůstal nastavený správný pracovní adresář:

```
getwd()
```

Pokud se vám při „úklidu“ stane, že vám zmizí nastavení pracovního adresáře, můžete ho znova nastavit příkazem

```
setwd("popis_cesty/matstat")
```

nebo pomocí

Session ➔ **Set Working directory** ➔ **To Source File Location**

což nastaví za pracovní adresář ten, v němž je uložený právě otevřený skript.

3 Načtení dat

- ❖ Data (nějaká vlastní měření apod.) budete mít většinou uložená v tabulkovém editoru (např. LibreOffice Calc, MS Excel, apod., takový soubor má příponu **.csv**) či v poznámkovém bloku (přípona **.txt**). V těchto formátech si lze již data snadno načíst do R.
- ❖ Podívejme se nyní na data **Sleep**, která jste si stáhli do své pracovní složky na disku J. (Je opravdu nezbytné, abyste měli data zkopiovaná do své složky na J!)
- ❖ Data **Sleep** obsahují následující proměnné:

<code>age</code>	Age of the person (18-60 years)
<code>gender</code>	Gender (<i>Male, Female</i>)
<code>sleep_length</code>	Total duration of sleep (in hours)
<code>sleep_quality</code>	Self-reported sleep quality (scale: 1-10)
<code>exercise</code>	Minutes spent exercising per day
<code>caffeine_intake</code>	Amount of caffeine consumed in mg
<code>screen_time_before_bed</code>	Time spent using screens before sleeping
<code>work_hours</code>	Total working hours in a day
<code>productivity_score</code>	Self-reported productivity score (scale: 1-10)
<code>mood_score</code>	Self-reported mood score (scale: 1-10)
<code>stress_level</code>	Self-reported stress level (scale: 1-10)

1) Načtěte data do R:

Bud' v pravém dolním okně v záložce **Files** klikněte na data **Sleep.csv** a zvolte **Import Dataset** a nastavte:

Name	Sleep
Delimiter	Semicolon (středník)
First Row as Names	✓
Locale: stiskněte Configure...	Decimal Mark → zvolte: , → Configure

(volbu Delimiter a Decimal Mark musíte uzpůsobit svým datům, která vidíte v náhledu). Pak stiskněte **Import** v pravém dolním rohu.

nebo využijte některý z následujících příkazů (v závislosti na zvoleném formátu dat):

```
Sleep <- read.table("data/Sleep.txt", header = TRUE, sep=",", dec=".") # txt soubory
Sleep <- read.csv2("data/Sleep.csv", header=TRUE, dec=".") # csv soubory
```

Argument `sep` udává znak, kterým jsou oddělené jednotlivé sloupce v textovém souboru, argument `dec` udává znak oddělující desetinná místa. Argument `header` říká, že první řádek souboru obsahuje názvy sloupců.

- 2) Data si můžete prohlédnout pomocí

```
View(Sleep)
```

nebo vypsat pomocí

```
print(Sleep) # nedoporučuji pro dlouhé datové soubory (což je i ten náš)
```

abyste zkontovali, že načtení dat do R proběhlo v pořádku.

4 Přístup k jednotlivým položkám dat

- 3) Jednotlivé řádky či sloupce datové tabulky si lze vypsat pomocí hranatých závorek

```
Sleep[3, ] # vypíše 3. řádek
Sleep[ ,5] # vypíše 5. sloupec
Sleep[3,5] # vypíše hodnotu nacházející se ve 3. řádku a 5. sloupci
```

- 4) K jednotlivým proměnným můžete přistupovat pomocí `Sleep$`, to jest

```
Sleep$age
Sleep$exercise
```

a podobně pro všechny ostatní proměnné. Chcete-li si pro kontrolu vypsat pouze prvních či posledních 6 hodnot, můžete použít

```
head(Sleep$age) # prvnich 6 hodnot
tail(Sleep$age) # posledních 6 hodnot
```

- 5) K jednotlivým složkám vektoru `age` se lze dostat opět operátorem `[]`.

```
Sleep$age[10] # věk 10. člověka
Sleep$age[1:5] # věk 1.-5. člověka
Sleep$age[c(2, 5, 7)] # věk 2., 5. a 7. člověka
```

- 6) Abyste nemuseli při přístupu k jednotlivým proměnným stále opisovat `Sleep$`, je možné zadat

```
attach(Sleep)
```

a pak lze již názvy jednotlivých proměnných volat přímo

```
age
exercise
```

Notace

```
Sleep$age
Sleep$exercise
```

přitom funguje stále!

7) Obdobně jako v bodě 5) se lze dostat k naměřeným údajům konkrétního člověka pomocí []:

```
Sleep[10, 5]      # hodnota 5. proměnné u 10. člověka (to už jsme viděli výše)
Sleep[10, "exercise"]    # totéž
Sleep[10, c(5, 6)]    # hodnota 5. a 6. proměnné u 10. člověka
Sleep[10, c("exercise", "caffeine_intake")]    # totéž
```

8) Chceme-li vypsat údaje více lidí současně

```
Sleep[7:11, ]      # všechny údaje 7.-11. člověka
Sleep[c(2, 5, 12), ]    # všechny údaje 2., 5. a 12. člověka
Sleep[c(2, 5, 12), c(5, 6)]    # hodnota 5. a 6. proměnné u 2., 5. a 12. člověka
Sleep[c(2, 5, 12), c("exercise", "caffeine_intake")]    # totéž
```

5 Výpočet nových proměnných

Pokud jsme si dříve data připojili pomocí `attach`, je potřeba před tvorbou nových proměnných (či jakýmkoli jiným zásehem do dat) data odpojit pomocí

```
detach(Sleep)
```

9) V datech je příjem kofeinu uveden v miligramech. Co kdybychom chtěli zpracovávat množství kofeinu v gramech? Vytvořme novou proměnnou, kterou nazveme `caffeine_intake.g`, jež bude udávat příjem kofeinu v gramech.

```
Sleep <- transform(Sleep, caffeine_intake.g = caffeine_intake / 1000)      # nebo
Sleep$caffeine_intake.g <- Sleep$caffeine_intake / 1000
```

 10) Prohlédněte si znova data a zjistěte, zda se vše povedlo.

 11) Samostatně vytvořte proměnnou `screen_time_before_bed.h` udávající dobu strávenou u obrazovky před spaním v hodinách.

 12) Samostatně vytvořte proměnnou `adulthood`, která bude udávat, kolik let již je osoba plnoletá, tj. `adulthood = age - 18`.

6 Uložení dat

Do datové tabulky `Sleep` jsme přidali několik nových proměnných (`caffeine_intake.g`, `adulthood`, `screen_time_before_bed.h`). V budoucnu s nimi možná budeme znova pracovat. Abychom si tyto proměnné nemuseli vytvářet znova, uložíme si současnou formu datové tabulky `Sleep`:

1) bud' ve formátu csv

```
write.table(Sleep, file = "data/SleepUpr.csv", row.names=FALSE,
            col.names=TRUE, sep=";", dec=".")
```

2) nebo ve formě R datového formátu (přípona `RData`)

```
save(Sleep, file = "data/Sleep.RData")      # doporučuji!
```

7 Konec práce

Než zavřete všechna okna, nezapomeňte si uložit skriptový soubor:

File ➔ Save as

nebo klávesovou zkratkou **Ctrl+s**. (Při standardním zavírání otevřených oken budete tak jako tak dotázáni, zda chcete tak učinit).