EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Základy práce v programu RStudio

Lukáš Hájek

2020

Strategický projekt UTB ve Zlíně, reg. č. CZ.02.2.69/0.0/0.0/16_015/0002204

Informace o autorovi:

Mgr. Lukáš Hájek, M.A., Ph.D.
Univerzita Karlova, Fakulta sociálních věd, Institut politologických studií
lukas.hajek@fsv.cuni.cz, + 420 721 509 106

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OBSAH

OBSAH **3**

ÚVOD **5**

1 ZÁKLADY **6**

1.1 R JAKO KALKULAČKA 6
1.2 OBJEKTY 7
1.3 DATOVÉ STRUKTURY 8
1.4 VÝBĚR ELEMENTŮ 9
1.5 PRÁCE S DATOVÝMI SETY 11
1.6 EXPORT A IMPORT DAT 12
1.7 VÝPOČET PRŮMĚRU A DALŠÍCH CHARAKTERISTIK 13
1.8 DALŠÍ UŽITEČNÉ FUNKCE 13
1.9 PRÁCE S R 15

2 VIZUALIZACE **17**

2.1 MANIPULACE S DATOVÝMI SETY 17
2.2 HISTOGRAMY 18
2.3 BODOVÉ GRAFY 19
2.4 GGPLOT2 22
2.5 DALŠÍ MOŽNÁ ZOBRAZENÍ 22

3 NÁSTROJE **24**

3.1 KONSTRUKCE INTERVALŮ SPOLEHLIVOSTI 24
3.2 KORELACE 26
3.3 T-TEST 27
3.4 KONTINGENČNÍ TABULKY 28

4 REGRESE **30**

4.1 IMPORT A ÚPRAVA DAT 30
4.2 JEDNODUCHÁ REGRESNÍ ANALÝZA 31
4.3 VÍCENÁSOBNÁ REGRESNÍ ANALÝZA 32
4.4 EXPORT VÝSLEDKŮ REGRESNÍ ANALÝZY 34
4.5 REAKCE NA PORUŠENÍ PŘEDPOKLADŮ (TRANSFORMACE DAT) 35

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

SHRNUTÍ **38**

SEZNAM POUŽITÉ LITERATURY **39**

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

ÚVOD

Programovací jazyk R je určen pro statistickou analýzu dat a napříč světem si v posledních letech získává stále větší míru popularity (R Foundation, 2020). K tomu přispívá nejenom vysoká variabilita a možnosti uplatnění tohoto programovacího jazyka, ale i celosvětová komunita a licence poskytovaná zdarma. Program RStudio je pak prostředím, které umožňuje pohodlnou práci s programovacím jazykem R (RStudio, 2020). Také RStudio je volně stažitelné z internetu, což z programu činí ideálního pomocníka pro využití všech výhod programovacího jazyka R.

Následující řádky seznamují čtenáře se základními způsoby používání jazyka R v programu RStudio. Rozděleny jsou přitom do čtyř kapitol, které nesou názvy „Základy“, „Vizualizace“, „Nástroje“ a „Regrese“. S jejich pomocí by měl každý čtenář získat ryze praktickou znalost základních a zároveň nejvíce využívaných vlastností a možností jazyka R.

Zbývající dokument je záměrně sepsán přímo v programovacím jazyce R, který je doplněn o komentáře. Cílem je umožnit čtenáři využít následující řádky přímo jako programovací sekvenci, se kterou je možné pracovat v programu RStudio. Text využívá několik externích datových souborů, které ukrývají reálná data z oblasti nedávných voleb v České republice, díky nimž je možné demonstrovat praktické výhody jazyka R. Všechny tyto zdrojové soubory je možné získat z externího datového uložiště (Hájek, 2020). V případě nejasností či problémů s kterýmkoliv ze souborů neváhejte kontaktovat autora (lukas.hajek@fsv.cuni.cz).

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

1 ZÁKLADY

Toto je jednoduchý textový soubor.

Soubor má koncovku ".R", takže víme, že jde o R skript.

V případě, že je soubor nečitelný, je třeba zvolit jiné kódování:

File -> Reopen with Encoding... -> UTF-8

V R konzoli níže je možné vykonat příkazy přímo,

nebo je napsat zde a poslat je do konzole pomocí Command + Enter (Mac) či Ctrl + Enter (Windows).

Preferujeme druhou metodu! Umožňuje totiž uložení skriptu.

Do konzole můžeme posílat jednotlivé řádky, několik řádků nebo třeba i celý soubor.

"#" jsou označení pro začátky komentářů, které nejsou interpretovány jako R kód.

1.1 R jako kalkulačka

1+1 # sčítání

3-1 # odečítání

2*5 # násobení

8/2 # dělení

2^3 # mocniny

16^(1/2) # odmocniny

Používáme funkce:

sqrt(16) # funkce potřebují zadání v ()

round(10.238)

Můžeme se podívat, jak má být zadání strukturováno:

```
help(round)
round(10.238, digits=1) # "digits" je argument
round(10.238, 1)        # stejné zadání jako v předchozím případě
round(digits=1,x=10.238)
```

1.2 Objekty

```
a <- 1    # 'a' je objekt
# V pracovním prostředí můžeme mít najednou několik objektů.
b <- 3
c <- "Hello"
d <- TRUE
# Všechny současné objekty v pracovním prostředí  můžeme zobrazit:
ls()
# Objekty lze i odstranit:
rm(a)  # odstraň "a"
rm(list=ls()) # odstraň vše z pracovního prostředí (dvojitá funkce!)
help(rm)
# R používá 4 základní typy objektů:
a <- FALSE          # logické (logical)
b <- 2.45           # numerické (numeric)
c <- as.integer(4)  # celá čísla (integer)
d <- "sun"          # textové (character)
class(a)
class(b)
```

```
class(c)

class(d)

rm(list=ls())
```

1.3 Datové struktury

```
# Různé typy objektů obsahují různé typy dat:

# skalár ("scalar"): číslo, znak, nebo logická hodnota (takové objekty jsme vytvořili výše),

# vektor ("vector"): sada skalárů,

# matice ("matrix"): dvojdimenzionální sada skalárů stejného (!) typu,

# datový set ("data frame"): kolekce vektorů (potenciálně) různých (!) typů, ale stejné délky,

# sada skalárů ("array"): vícedimenzionální sada skalárů,

# list ("list"): kombinace skalárů, vektorů, matic...

# Zkusme vektory:

# Několik hodnot je zkombinováno pomocí "c()".

a <- c(1,2,3)

a

# Stejný princip funguje pro textové objekty.

country <- c("DE", "FR", "NL")

country

# Můžete použít funkce pro každý element ve vektoru, matici apod.

sqrt(a)

d <- a+2
```

log(country) # R nás upozorní na chyby - ovšem pouze když odporuje jeho zákonům!

Zkusme matice:

help(matrix)

mat <- matrix(data = NA, nrow = 3, ncol = 9) # "NA" (not available) značí prázdné pole

mat

nrow(mat) # počet řádků, neboli pozorování

ncol(mat) # počet sloupců, neboli proměnných

rm(list=ls())

1.4 Výběr elementů

Elementy objektů vybíráme pomocí [].

a <- c(10:12)

a

a[3]

a[1:3]

a[-3]

a[c(1,3)]

a[c(-1,-3)]

Výběr elementů z matice:

m <- matrix(data = c(a,a+5), nrow = 3, ncol = 2)

m

m[1,1]

m[2,]

```
m[,2] # první se uvádí číslo řádku, potom číslo sloupce - pomůcka "Roman Catholics" (row, column)

# Můžeme také použít [], abychom vyměnili konkrétní elementy.

m[1,1] <- 0

m[2,] <- 0

m

# Výběr elementů na základě podmínek.

# Dostupné podmínky:

  # == "rovná se"

  # != "nerovná se"

  # < "je menší"

  # > "je větší"

  # <= "je menší nebo rovno"

  # => "je větší nebo rovno"

# Může být zkombinováno s logickými argumenty:

  # & "a"

  # | "nebo"

x <- c(1,56,23,89,-3,5)

y <- c(24,78,32,27,8,1)

x[x > 20]            # výběr x, které je větší než 20

x[x >= 23]           # výběr x, které je větší nebo rovno 23

x[x > 20 & x != 89]   # výběr x, které je větší než 20 a nerovná se 89

x[x > 0 | x != -3]    # výběr x, které je větší než 0 nebo x, které se nerovná -3

y[x == 1]            # výběr y, když se x rovná 1
```

1.5 Práce s datovými sety

```
rm(list = ls())

var1 <- c(17.87,25.34,25.3,16.73,25.75)

var2 <- c(95,80.40,90.6,60.54,0.000)

var3 <- c(0,4.43,55.556,86.00,76.190)

dat <- data.frame(var1,var2,var3)

names(dat)

dat

# Výběr pomocí numerické pozice.

dat[1,2]

dat[1:3,2]

# Výběr pomocí názvu sloupce.

dat[1,"var2"]

dat[ ,"var2"]

# Výběr pomocí názvu proměnné.

dat$var2

# Se sloupci můžeme nakládat jako s vektory.

dat[1,2]

dat$var2[1] # stejný výběr

# Můžeme vytvářet nové proměnné.

dat$var4 <- dat$var2 + dat$var3

# Přejmenování sloupců:

colnames(dat)

colnames(dat) <- c("ODS","ANO","KSCM","STAN")
```

```
colnames(dat)[3] <- "KDU"
```

1.6 Export a import dat

```
# Odhalení pracovní složky ("working directory"):
getwd()
# Při práci s R se při otevření skriptu automaticky kód spáruje s příslušnou složkou.
# Pokud byste ale následně otevřeli jiný skript z jiné složky, R pořád bude pracovat s tou první.
# Je proto nutné po každém ukončení práce na skriptu zavřít celý program,
# nebo je třeba nastavovat pracovní složku pomocí funkce setwd().
# Pro použití některých funkcí musíme nejprve nahrát příslušnou knihovnu funkcí:
library(foreign)  # knihovna pro export/import
# Stata datové soubory
write.dta(dat, file="data.dta") # tímto exportujeme datový set do pracovní složky
list.files()
rm(dat)
dat <- read.dta("data.dta") # tímto způsobem datový set importujeme zpět
# CSV
write.csv(dat, file="data.csv", row.names = F) # export je možné provádět i v dalších formátech
list.files()
rm(dat)
dat <- read.csv("data.csv") # import dat
# SPSS
install.packages("haven") # někdy je třeba nainstalovat nový balíček funkcí
```

```
library(haven)
write_sav(dat, path = "dat.sav")
list.files()
rm(dat)
dat <- read_spss("dat.sav") # import dat
```

1.7 Výpočet průměru a dalších charakteristik

```
rm(list=ls())
dat <- read.dta("data.dta")
colnames(dat) <- c("var1","var2","var3","var4")
# Výpočet průměru:
mean(dat$var1)
# Pokud to jde, používáme co nejvíce obecný zápis.
# V případě jakékoliv změny pak totiž kód stále funguje.
# Ostatní funkce:
median(dat$var1) # medián
var(dat$var1) # rozptyl
sqrt(var(dat$var1)) # odmocnina
sd(dat$var1) # standardní odchylka
range(dat$var1) # rozpětí
# A tak dále - většina statistických funkcí existuje - je třeba jen znát jejich název.
```

1.8 Další užitečné funkce

```
# Řazení:
```

```
sort(c(0,7,8,3,4,9))

# Opakované zapisování:

rep(1,7)

# Délka objektů:

length(c(8,9,4,5,6))

# Přehled objektu:

table(c(1,1,2,3,4,5,5,5,6,6,7,9,9,9,9,9))

# Unikátní hodnoty objektu:

unique(c(1,1,2,3,4,5,5,5,6,6,7,9,9,9,9,9))

# Kombinace textů:

paste0(c("ČSSD","ODS","KSČM","ANO"),"2014")

paste(c("ČSSD","ODS","KSČM","ANO"),"2014")

# Výběr textu:

substr("Číslo schůze: 36",15,16)

# Kontrola obsahu textu:

grepl("doporučuje","Toto je text schůze, který se doporučuje")

# Délka textu:

nchar("Petr Pan")

# Transformace objektů:

as.integer("2")

as.integer(2.86)

as.numeric(2)

as.character(367)

# A mnoho dalších...
```

1.9 Práce s R

Náš skript máme uložený - je to textový soubor, ale koncovka .R odkazuje právě na R.

Uložit ale můžeme i celé pracovní prostředí

save(list = ls(), file= "seminar1.RData")

list.files()

rm(list = ls())

load("seminar1.RData")

Často bude nutné nainstalovat příslušný balíček funkcí.

POZOR!!! Pro instalaci balíčků je třeba R alespoň jednou otevřít jako správce.

V některých případech může být také nutné nainstalovat nejnovější verzi Java:

https://www.java.com/en/download/manual.jsp - na moderních počítačích instalace 64-bitové verze Java.

Existují i velmi zábavné balíčky:

install.packages("cowsay") # https://cran.r-project.org/web/packages/cowsay/index.html

library(cowsay)

say("ahoj", by = "shark")

Instalovat balíky stačí jenom jednou, nahrát je ale musíte pokaždé při zapnutí R.

Proto je dobré každý skript začít následujícími jednotnými příkazy:

rm(list = ls())

load("seminar1.RData")

save(list = ls(), file= "seminar1.RData")

getwd()

```
rm(list = ls())

rm()

install.packages("cowsay")

library(foreign)

library(cowsay)

# Dále už pak následují analýzy...

# Pomůcka: např. Quick-R [http://www.statmethods.net], Stack Overflow [https://stackoverflow.com] a další.

# Podívejte se i jinde na webu na další návody. Nebojte se googlit... Musíte googlit...

# Na Facebooku sledujte například "R bloggers".

# Používejte komentáře (#) k rozdělení a popisu kódu.

# Cílem je, aby byl přehledný a pochopitelný pro vás i ostatní.

# Používejte RStudio pro lepší práci s R. Co nejméně používejte klikací funkce - pište příkazy.

# Veškerý kód musí být zapsán ve skriptu tak, aby bylo možné postup replikovat.

# Pravděpodobně budete chtít použít kód v budoucnosti, takže ho uložte.
```

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

2 VIZUALIZACE

2.1 Manipulace s datovými sety

Je vhodné mít vytvořenou zvláštní složku na import dat - zde jsou vloženy původní datasety.

Import datového setu s kandidáty v senátních volbách.

zdroj: https://volby.cz/opendata/se2018/se2018_opendata.htm

install.packages("readxl")

library(readxl)

serk <- as.data.frame(read_excel("data/serk.xlsx", sheet = 1))

dim(serk) # dataset má 237 řádků (pozorování - N) a 26 sloupců (proměnných)

colnames(serk)

head(serk)

Důležité je podívat se na kódovací knihu a pracovat s ní.

Většinu původních datasetů je třeba nejprve vyčistit.

serk <- serk[serk$PLATNOST=="A",] # u seznamu kandidátů je typicky nutné odstranit ty neplatné

Kombinace více datových setů:

cns <- read_excel("data/SE2018ciselnik20181004/cns.xlsx", sheet = 1)

colnames(cns)

serk <- merge(x = serk, y = cns[,c("NSTRANA","ZKRATKAN8")], by = "NSTRANA") # propojení datových setů

cpp <- read_excel("data/SE2018ciselnik20181004/cpp.xlsx", sheet = 1)

colnames(cpp)

```
serk <- merge(x = serk, y = cpp[,c("PSTRANA","ZKRATKAP8")], by = "PSTRANA") # propojení datových setů
head(serk)
# Dataset je možné dále třídit a vytvářet menší datové sety:
serk_o2 <- serk[serk$OBVOD==2,]
serk_vek <- serk[,c("JMENO","PRIJMENI","VEK")]
```

2.2 Histogramy

```
rm(serk_vek,serk_o2,data)
# Funkce pro zakreslení histogramu:
help(hist)
hist(serk$VEK)
# Určitě by to ale šlo udělat hezčím způsobem:
hist(x = serk$VEK,
    main = "Vek kandidatu na senatora", xlab = "Vek", ylab = "Frekvence",
    col = "forestgreen")
# http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf paleta barev v R
# Další atributy histogramu:
hist(x = serk$VEK,
    main = "Vek kandidatu na senatora", xlab = "Vek", ylab = "Frekvence",
    col = "forestgreen",
    breaks = 20, ylim = c(0,40), xlim = c(40,100), labels = T)
# Export jakéhokoliv grafu:
pdf("export/histogram_vek.pdf", width = 10, height = 5)
```

```
hist(serk$VEK,
    main = "Vek kandidatu na senatora", xlab = "Vek", ylab = "Frekvence",
    col = "forestgreen",
    breaks = 20, ylim = c(0,25), xlim = c(40,80), labels = T)
dev.off() # zakončuje zapisování grafu a ruší veškeré grafické nastavení
# Exportovat je možné v různých formátech - jpeg, bmp, tiff apod.
```

2.3 Bodové grafy

```
# Využijeme generickou funkci "plot".
# Podívejme se na zakreslení věku kandidátů a jejich procentuálních zisků v 1. kole:
plot(x = serk$VEK, y = serk$PROC_K1)
# Graf opět trochu zkrášlíme:
plot(x = serk$VEK, y = serk$PROC_K1,
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu")
# Důležitá vlastnost je možná změna podoby bodů:
plot(x = serk$VEK, y = serk$PROC_K1,
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu",
    pch = 16, cex = 1, col = "red")
#   http://www.sthda.com/english/wiki/r-plot-pch-symbols-the-different-point-shapes-available-in-r symboly
# Přidat můžeme hranici znázorňující zisk mandátu už v 1. kole:
help(abline)
```

```
abline(h = 50)
plot(serk$VEK,serk$PROC_K1,
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu",
    pch = 16, cex = 1, col = "red")
abline(h = 50, lt = "dashed", lw = 3) # i linie lze zakreslit podle konkrétní potřeby
# Přidáme ještě popisek této přímky:
help(text)
text(x = 45, y = 50,
    labels = "Zisk mandatu v 1. kole", pos = 3)
# Přidat je možné třeba i specifický bod:
help(points)
points(x = 75, y = 40, pch = 4, cex = 2)
# Můžeme samozřejmě zobrazit jenom část bodů:
plot(serk$VEK[serk$ZKRATKAN8=="ODS"],serk$PROC_K1[serk$ZKRATKAN8=="ODS"]
,
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu ODS",
    pch = 16, cex = 1, col = "darkblue")
# Do grafu je možné přidat popisky:
text(serk$VEK[serk$ZKRATKAN8=="ODS"],
serk$PROC_K1[serk$ZKRATKAN8=="ODS"],
    labels = serk$PRIJMENI[serk$ZKRATKAN8=="ODS"], pos = 3)
# Při vhodných příležitostech do grafů náleží legenda.
```

```
plot(serk$VEK[serk$ZKRATKAN8=="ODS"],serk$PROC_K1[serk$ZKRATKAN8=="ODS"]
,
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu ODS",
    pch = 16, cex = 1, col = "darkblue")
abline(h = mean(serk$PROC_K1[serk$ZKRATKAN8=="ODS"]), col = "blue", lwd = 3,
lty = "dashed")
help(legend)
legend("topleft", legend=c("Kandidati ODS", "Prumerny zisk"),
     col=c("darkblue", "blue"), lty=c(NA,"dashed"), lwd = 3, pch = c(16,NA),
cex=0.8)
# Exportujme nyní srovnání ODS a ČSSD:
pdf("export/body_ods_cssd.pdf", width = 10, height = 5)
par(mfrow=c(1,2)) # tento příkaz rozděluje zakreslovací plochu do několika polí
(počet řádků a sloupců)
plot(serk$VEK[serk$ZKRATKAN8=="ODS"],serk$PROC_K1[serk$ZKRATKAN8=="ODS"]
,
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu ODS",
    pch = 16, cex = 1, col = "darkblue")
plot(serk$VEK[serk$ZKRATKAN8=="ČSSD"],serk$PROC_K1[serk$ZKRATKAN8=="ČSS
D"],
    xlab = "Vek", ylab = "Zisk v 1. kole (%)",
    main = "Vek a zisky senatorskych kandidatu CSSD",
    pch = 16, cex = 1, col = "darkorange")
```

```
dev.off()
# Pozor na srovnání obou grafů - může být zavádějící...
```

2.4 ggplot2

```
# Pro vizualizace je možné využít i balík ggplot2.
# Pomůckou může být například: https://ggplot2.tidyverse.org
# Někdo preferuje klasický způsob zobrazení, někdo naopak ggplot2 - oba způsoby jsou rovnocenné.
install.packages("ggplot2")
library(ggplot2)
ggplot(serk, aes(x=VEK, y=PROC_K1)) +
      geom_point(size=2, shape=23) +
      labs(title="Ukázka ggplot2", x="Věk", y = "Zisk v 1. kole (%)")
```

2.5 Další možná zobrazení

```
# Koláčové grafy.
help(piechart)
# Liniové grafy:
dny <- c(1:30) # počet dní měření
pruzkumy <- rnorm(n = 30, mean = 35, sd = 4) # náhodný výběr 30 měření
plot(x = dny, y = pruzkumy,
    xlab = "Den", ylab = "Volebni preference",
    type = "o", col = "darkviolet", lwd = 2, lty = "dashed")
text(x = dny, y = pruzkumy,
```

```
    labels = round(pruzkumy,1), pos = 4)
# a mnohá další...
```

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

3 NÁSTROJE

3.1 Konstrukce intervalů spolehlivosti

```
##########################
## Analyticky
##########################
# V souboru "weight.Rdata" je možné najít hodnoty váhy u 50 lidí.
# Vypočítejte 95% interval spolehlivosti průměru.
# Nahrajme data:
library(foreign)
load(file= "data/weight.Rdata")
# Podívejme se na data:
weight
summary(weight)
hist(weight)
length(weight)
# Standardní chyba průměru vzorku: sd/sqrt(n).
weight.se <- sd(weight)/sqrt(length(weight))
weight.se
# využití normální distribuce
ci.low <- mean(weight) + qnorm(0.025, mean = 0, sd = 1)*weight.se
ci.up <-  mean(weight) + qnorm(0.975, mean = 0, sd = 1)*weight.se
ci.an <- c(ci.low, ci.up)
```

```
ci.an # průměr ostatních vzorků bude uvnitř tohoto intervalu v 95 % případů
# využití t-distribuce
ci.t <- mean(weight) + qt(c(0.025,0.975), df = 49)*weight.se
ci.t
##########################
## Simulace
##########################
# Nastavíme počet simulací:
nsim <- 1000
# Provedeme 1000 výběrů z normálního rozložení
# s průměrem a standardní chybou jako v datech. Proč?
simweight <- rnorm(n = nsim, mean = mean(weight), sd = weight.se)
simweight
# Zakresleme histogram:
hist(simweight, col="grey80", breaks=20, freq=FALSE,
    main="Výběrová distribuce průměru vzorku (simulační)", xlab = "Vaha")
# Do grafu nyní přidáme:
# linii pro hustotu rozložení
lines(density(simweight))
# výpočet empirických kvantilů z distribuce
ci.sim <- quantile(simweight,c(0.025,0.975))
# zakreslení kvantilů
segments(ci.sim[1], 0, ci.sim[1], 0.2, col="red",lwd=2, lty = 2)
segments(ci.sim[2], 0, ci.sim[2], 0.2, col="red",lwd=2, lty = 2)
```

```
# zakreslení intervalů spolehlivosti z předchozí analytické kalkulace
segments(ci.an[1], 0, ci.an[1], 0.2, col="green",lwd=2, lty = 2)
segments(ci.an[2], 0, ci.an[2], 0.2, col="green",lwd=2, lty = 2)
ci.sim - ci.an # rozdíl by měl být blízko nuly!
```

3.2 Korelace

```
rm(list=ls())
# Importujme data o výsledcích prezidentských voleb v roce 2018 v obcích.
library(readxl)
prezident <- read_excel("data/prezident2018_kor.xlsx")
# zdroj: úprava na základě dat https://volby.cz/open-data/prez2018/prez2018_opendata.htm
# Jaký byl vztah mezi ziskem Jiřího Drahoše v 1. a 2. kole?
cor(prezident$prez18_1k_drahos_proc, prezident$prez18_2k_drahos_proc)
# Zobrazit můžeme hned několik korelačních vztahů najednou.
# Nejprve proto vytvoříme korelační matici.
help(cor)
correl <- cor(prezident[,2:12])
install.packages("ggcorrplot")
library(ggcorrplot)
p.mat <- cor_pmat(prezident[,2:12]) # spočítáme také p-hodnoty jednotlivých vztahů
# Chceme totiž odhalit vztahy, které nejsou statisticky signifikantní.
# Nyní můžeme zakreslit všechny výsledky.
ggcorrplot(round(correl,1), hc.order = F, type = "lower", p.mat = p.mat,
```

```
        insig = "blank", lab = F, colors = c("#DC1819", "white" ,"#000A50"),
        method = "circle", show.diag = F, outline.color = "grey",
        title = "Prelivy hlasu v prezidentskych volbach")
# Interpretujte zjištěné výsledky
```

3.3 T-test

```
rm(list=ls())
# Soubor fotp.xlsx obsahuje informace o skóre svobody tisku organizace Freedom House.
library(readxl)
data <- read_excel("data/fotp.xlsx", sheet = 2, na = "-")
# zdroj: upraveno na základě https://freedomhouse.org/report/table-country-scores-fotp-2017
# Z datasetu vyřadíme země, kde neproběhlo měření ve všech letech.
data <- data[!is.na(rowSums(data[,2:10])),]
# Proveďte test, zda je rozdíl ve skóre mezi lety 2016 a 2015 statisticky signifikantní.
t.test(data$`2016`, data$`2015`, paired = TRUE, alternative = "two.sided")
# Jelikož je p-hodnota nižší než 0.05 (95% úroveň jistoty), je rozdíl signifikantní.
# Průměrné skóre bylo v roce 2015 o téměř půl bodu nižší než v roce 2016.
# V roce 2016 tak došlo ke zhoršení svobody tisku oproti předchozímu roku.
# Zkuste si změnit parametry funkce t.test - změňte úroveň jistoty a uvidíte, že p-hodnota zůstává stejná,
# ale samozřejmě se mění intervaly spolehlivosti. Sledujte, kdy obsahují nulu a kdy nikoliv
```

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

(to je totiž přímo spojeno právě s p-hodnotou).

3.4 Kontingenční tabulky

```
rm(list = ls())
# Existuje vztah mezi úrovní vzdělání a důvěrou v prezidenta republiky?
data <- read.csv("data/V1809_F1.csv") # data CVVM ze září roku 2018
# zdroj: Český sociálněvědní datový archiv http://nesstar.soc.cas.cz/webview/
# Vytvoření nových předmětných proměnných pro jednoduchost:
data$duvera_prez <- data$PI.1A
data$vzdelani <- data$t_VZD
# Zmenšení datového setu:
data <- data[,c("duvera_prez","vzdelani")]
# Kontrola podoby proměnných:
table(data$vzdelani)
table(data$duvera_prez)
data <- data[data$duvera_prez!=9,] # vynechání osob, které odpověděli, že neví
# Rekódování proměnných:
data$duvera_prez <- as.factor(data$duvera_prez)
data$vzdelani <- as.factor(data$vzdelani)
library(plyr)
data$duvera_prez <- revalue(data$duvera_prez, c("1"="rozhodne duveruje", "2"="spise duveruje",
                                                "3"="spise neduveruje", "4"="rozhodne neduve-
ruje"))
data$vzdelani <- revalue(data$vzdelani, c("1"="zakladni", "2"="vyuceni",
```

```
                                "3"="stredni", "4"="vysokoskolske"))
# Vytvoření kontingenční tabulky:
tabulka <- table(data$duvera_prez,data$vzdelani)
# Chí-kvadrát test:
chisq <- chisq.test(tabulka)
chisq
# Porovnání pozorovaných a očekávaných hodnot:
chisq$observed
chisq$expected
# Zakreslení reziduí chí-kvadrát testu jednotlivých buněk:
install.packages("corrplot")
library(corrplot)
corrplot(chisq$residuals, is.cor = FALSE) # pozitivní rezidua jsou modrá, zatímco negativní červená
# Výpočet podílu jednotlivých vztahů na celkovém výsledku chí-kvadrát testu:
podil <- 100*chisq$residuals^2/chisq$statistic
round(podil, 3)
corrplot(podil, is.cor = FALSE)
```

4 REGRESE

4.1 Import a úprava dat

```
# Analýza volební účasti ve volbách do Evropského parlamentu 2019:

eurovolby <- read.csv("data/eurovolby_2019.csv")

# zdroj: https://volby.cz/opendata/ep2019/ep2019_opendata.htm

eurovolby <- eurovolby[eurovolby$TYP_OBCE != "Městská část",] # vyřadíme městské části

# Počet obyvatel v obcích k 1. lednu 2019:

library(readxl)

obyvatel <- read_excel("data/obyvatele_0119.xlsx", sheet = 1)

# zdroj: https://www.czso.cz/csu/czso/pocet-obyvatel-v-obcich-za0wri436p

# Nezaměstnanost v obcích v dubnu 2019:

nezamestnanost <- read_excel("data/nezamestnanost_0419.xlsx", sheet = 1)

# zdroj: MPSV

# Kombinace datových setů:

eurovolby <- merge(x = eurovolby, y = nezamestnanost[,c("KOD","NEZAMESTNANYCH")], by.x = "KODZASTUP", by.y = "KOD")

eurovolby <- merge(x = eurovolby, y = obyvatel[,c("KODZASTUP","OBYVATEL","ZENY","VEK")], by = "KODZASTUP")

# Úprava proměnné:

eurovolby$ZENY_PROC <- eurovolby$ZENY/eurovolby$OBYVATEL*100
```

4.2 Jednoduchá regresní analýza

Naší alternativní hypotézou je, že míra nezaměstnanosti v obcích měla vliv na volební účast.

Co je nulová hypotéza?

Nejprve se podívejme na rozložení dat:

```
hist(eurovolby$NEZAMESTNANYCH, breaks = 50)

hist(eurovolby$UCAST, breaks = 50)

plot(eurovolby$NEZAMESTNANYCH, eurovolby$UCAST) # kontrola vztahu dat
```

Regresní analýza:

```
reg_ucast_m1 <- lm(formula = UCAST ~ NEZAMESTNANYCH, data = eurovolby)

summary(reg_ucast_m1)
```

Ve výsledku vidíme hodnotu průniku (32.60) a koeficientu nezaměstnanosti (-0.86).

Obě hodnoty mají určitou chybu a t-hodnoty.

Zásadní je poslední sloupec Pr(>|t|) - u koeficientu vidíme, že je p-hodnota v podstatě nulová.

Můžeme tedy téměř se 100% pravděpodobností zamítnout nulovou hypotézu.

Dále vidíme vysvětlovací schopnost modelu, která činí 3.2 % vsysvětleného rozptylu závisle proměnné.

F-statistický test ukazuje na celkovou kvalitu modelu - čím je výsledek vyšší, tím lépe.

Zakreslení výsledku:

```
plot(eurovolby$NEZAMESTNANYCH, eurovolby$UCAST,
     xlab = "Míra nezaměstnanosti v obci (%)", ylab = "Volební účast (%)")

abline(reg_ucast_m1, col = "red", lwd = 2)
```

```
# Intervaly spolehlivosti:
confint(reg_ucast_m1, level = 0.99)
# Zakreslení:
abline(confint(reg_ucast_m1, level = 0.99)[,1], lt = "dashed", col = "red")
abline(confint(reg_ucast_m1, level = 0.99)[,2], lt = "dashed", col = "red")
abline(confint(reg_ucast_m1, level = 0.95)[,1], lt = "dashed", col = "darkgreen") # menší interval
abline(confint(reg_ucast_m1, level = 0.95)[,2], lt = "dashed", col = "darkgreen")
legend("topright", legend=c("99% interval spolehlivosti", "95% interval spolehlivosti"), # legenda
    col=c("red", "darkgreen"), lty="dashed", cex=0.8)
# Všimněte si, že regresní přímka protíná křížení průměrů obou proměnných:
abline(v = mean(eurovolby$NEZAMESTNANYCH), col = "yellow", lwd = 2)
abline(h = mean(eurovolby$UCAST), col = "yellow", lwd = 2)
# Proč tomu tak je?
```

4.3 Vícenásobná regresní analýza

```
# Naší alternativní hypotézou dále je, že nezaměstnanost není jediným faktorem,
# který ovlivnil volební účast ve volbách do Evropského parlamentu.
# Dalšími faktory, u kterých předpokládáme vliv, jsou:
# velikost obce, průměrný věk v obci a procentuální zastoupení žen.
# Znovu se nejprve podívejme na rozložení dat:
hist(eurovolby$UCAST, breaks = 50)
hist(eurovolby$NEZAMESTNANYCH, breaks = 50)
hist(eurovolby$VEK, breaks = 50)
```

```
hist(eurovolby$ZENY_PROC, breaks = 50)

hist(eurovolby$OBYVATEL, breaks = 50)

# Regresní analýza:

reg_ucast_m2 <- lm(formula = UCAST ~ NEZAMESTNANYCH

                + OBYVATEL + VEK + ZENY_PROC, data = eurovolby)

summary(reg_ucast_m2)

# Jaké koeficienty nezávisle proměnných jsou statisticky signifikantní na úrovni 99 %?

# Podíváme se na splnění předpokladů:

install.packages("car")

library(car)

plot(eurovolby$NEZAMESTNANYCH, eurovolby$UCAST) # lineární vztah proměnných

vif(reg_ucast_m2) # test multikolinearity

cooks.distance(reg_ucast_m2)[cooks.distance(reg_ucast_m2)>1] # test odlehlých hodnot

par(mfrow=c(2,2))

plot(reg_ucast_m2) # test homoskedasticity a normální distribuce reziduí

dev.off()

durbinWatsonTest(reg_ucast_m2) # test nezávislosti reziduí

# Simulace intervalů spolehlivosti:

range(eurovolby$NEZAMESTNANYCH) # nezaměstnanost se pohybuje cca mezi 0 a 20 %

# Nejprve vytvoříme simulační matici nezávisle a kontrolních proměnných.
```

```
# Kontrolní proměnné nastavujeme do pozice středních či co nejvíce frekventovaných hodnot.
# Jde tedy o simulaci co nejběžnějšího a tudíž nejpravděpodobnějšího scénáře:
ci <- data.frame(NEZAMESTNANYCH = seq(0,20,1), OBYVATEL = mean(eurovolby$OBYVATEL),
                 VEK = mean(eurovolby$VEK), ZENY_PROC = mean(eurovolby$ZENY_PROC))
# Následně odhalíme hodnoty závisle proměnné
sim <- predict(reg_ucast_m2, newdata = ci, interval = "confidence", level = 0.99)
# Výsledek zakreslíme:
plot(eurovolby$NEZAMESTNANYCH, eurovolby$UCAST,
    ylab = "Volební účast (%)", xlab = "Míra nezaměstnanosti (%)",
    main = "Vztah míry nezaměstnanosti a volební účasti (komplexní model)")
lines(c(0:20),sim[,1], lt = "solid", lwd = 2, col = "blue")
polygon(c(c(0:20),rev(c(0:20))), c(sim[,2],rev(sim[,3])),
       col = adjustcolor("blue",alpha = 0.2),  border = FALSE) # zakreslení polygonu
lines(c(0:20),sim[,2], lt = "dashed", lwd = 1, col = "blue")
lines(c(0:20),sim[,3], lt = "dashed", lwd = 1, col = "blue")
abline(reg_ucast_m1, col = "red", lwd = 2) # srovnání s jednoduchou regresí
```

4.4 Export výsledků regresní analýzy

```
# Existuje několik možností - jednou z nejlepších je funkce stargazer.
install.packages("stargazer")
library(stargazer)
help(stargazer)
```

Regresní tabulku zapisuje například jako formát html.

Následně stačí stejný soubor otevřít ve wordu a využít příslušnou tabulku.

stargazer(reg_ucast_m1, type = "html", out = "export/reg_ucast.html")

Exportovat je možné i několik modelů současně.

stargazer(list(reg_ucast_m1, reg_ucast_m2), type = "html", out = "export/reg_ucast.html")

4.5 Reakce na porušení předpokladů (transformace dat)

Podívejme se znovu na distribuci dat u počtu obyvatel v obcích.

hist(eurovolby$OBYVATEL[eurovolby$OBYVATEL < 10000], breaks = 50) # data jsou pozitivně zešikmená

Předpokladem lineární regresní analýzy ale je, že distribuce dat se co nejvíce blíží normálnímu rozložení.

Je proto třeba přistoupit k transformaci dat -> logaritmus je v tomto případě ideálním řešením.

hist(log(eurovolby$OBYVATEL), breaks = 50)

eurovolby$OBYVATEL_LOG <- log(eurovolby$OBYVATEL) # vytvoříme novou proměnnou

Model nyní spočítáme s novou ("lepší") proměnnou.

Regresní analýza:

reg_ucast_m3 <- lm(formula = UCAST ~ NEZAMESTNANYCH

+ OBYVATEL_LOG + VEK + ZENY_PROC, data = eurovolby)

summary(reg_ucast_m3)

Podívejme se na výsledky všech tří modelů:

```
stargazer(list(reg_ucast_m1, reg_ucast_m2, reg_ucast_m3), type = "html", out = "export/reg_ucast.html")

# Vysvětlovací schopnost modelu č. 3 se oproti předchozím zvýšila a

# hlavně jsme odhalili statistickou významnost počtu obyvatel.

# Provedeme simulaci:

range(eurovolby$OBYVATEL_LOG)

ci <- data.frame(NEZAMESTNANYCH = mean(eurovolby$NEZAMESTNANYCH), OBYVATEL_LOG = seq(2,15,0.1),

            VEK = mean(eurovolby$VEK), ZENY_PROC = mean(eurovolby$ZENY_PROC))

# Následně odhalíme hodnoty závisle proměnné

sim <- predict(reg_ucast_m3, newdata = ci, interval = "confidence", level = 0.99)

plot(eurovolby$OBYVATEL_LOG, eurovolby$UCAST,

    ylab = "Volební účast (%)", xlab = "Počet obyvatel (log)",

    main = "Vztah velikosti obce a volební účasti (komplexní model)")

lines(seq(2,15,0.1),sim[,1], lt = "solid", lwd = 2, col = "orange")

lines(seq(2,15,0.1),sim[,2], lt = "dashed", lwd = 1, col = "orange")

lines(seq(2,15,0.1),sim[,3], lt = "dashed", lwd = 1, col = "orange")

# Pozor ale na interpretaci! Proměnná není ve své původní podobě, ale je zlogaritmovaná!

plot(eurovolby$OBYVATEL, eurovolby$UCAST,

    ylab = "Volební účast (%)", xlab = "Počet obyvatel",

    main = "Vztah velikosti obce a volební účasti (komplexní model)",

    xlim = c(0,10000))

lines(c(exp(seq(2,15,0.1))),sim[,1], lt = "solid", lwd = 2, col = "orange")
```

```
lines(c(exp(seq(2,15,0.1))),sim[,2], lt = "dashed", lwd = 1, col = "orange")
lines(c(exp(seq(2,15,0.1))),sim[,3], lt = "dashed", lwd = 1, col = "orange")
# Vidíme, že vztah díky transformaci už není lineární, ale právě logaritmický.
# Každou regresní analýzu by měly následovat analýzy robustnosti výsledků.
# Testuje se, jak moc jsou výsledky solidní - vytváříme různé modely s odlehlými hodnotami a bez nich,
# operacionalizujeme proměnné různým způsobem a sledujeme, zda vztah pořád přetrvává atd.
```

SHRNUTÍ

Předložený dokument názorně demonstruje, že programovací jazyk R je ve spolupráci s programem RStudio schopen poskytnout všechny nástroje pro základní i pokročilou analýzu statistických dat. Hlavní výhodou jazyka R je oproti základním statistickým programům zejména možnost ukládání programovacích sekvencí do podoby skriptů, které je následně možné spouštět opakovaně i při změně primárních datových vstupů.

Zároveň však dokument přirozeně není vyčerpávajícím představením všech výhod a možností jazyka R a programu RStudio. Jako další v řadě se nabízí využití nelineárních analýz dat či analýzy textů. Bez nadsázky je možné programovací prostředí R využít ke všem myslitelným analytickým postupům, které dnešní prostředí nejenom sociálních věd nabízí. Předložený dokument tak představuje pouze pootevřené okno do mnohem většího světa ke zkoumání a využití.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

SEZNAM POUŽITÝCH ZDROJŮ

Hájek, Lukáš, 2020, "Základy práce v programu RStudio", https://doi.org/10.7910/DVN/MDGOSM, Harvard Dataverse, V1.

R Foundation. (2020, Aug 29). The R Project for Statistical Computing. https://www.r-project.org

RStudio. (2020, Aug 29). RStudio. https://rstudio.com

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY