VKLÁDÁNÍ A ČIŠTĚNÍ DAT, ZJIŠŤOVÁNÍ ZÁKLADNÍCH INFORMACÍ O DATOVÉM SOUBORU

Vít Gabrhel

vit.gabrhel@mail.muni.cz

FSS MU,

3. 10. 2016

Harmonogram

0. Rekapitulace předchozí hodiny

1. Importování dat do R

2. Čištění dat

3. Popisné statistiky

Rekapitulace

Skript

matrix versus data.frame

```
# Jednotlivé vektory spojte dohromady v matici s názvem Matice
Matice <- data.frame(Jmena, Pohlavi, Vzdelani, Vzdelani_factor, RWA_skor)
/
Matice<- matrix(c(Jména,Pohlaví,Vzdělání,Vzdělání_Factor,RWA_Skór), nrow = 5, byrow = FALSE)
```

subset a srovnání

```
# Skrze funkci subset() vytvořte z proměnné Vzdělání_Factor objekty ZŠ_Subset a SŠ_Subjekt, které skrze logické operátory porovnejte ve skóre získaném u RWA.
```

ZŠ_subset <-subset (Matice, subset = (Vzdělání_Factor =="ZŠ"))

```
# Skórovali na škále RWA více lidé se ZŠ nebo lidé se SŠ?
sum(Subset_ZŠ$RWA_Skór) < sum(Subset_SŠ$RWA_Skór)

== versus =
== for equal to each other
```

nový sloupec v existující matici

```
# Až zpětně vám došlo, že srovnávat celkový skór nestejně velkých skupin není v pořádku. Do obou subsetů (tedy ZŠ_Subset a SŠ_Subset) přidejte sloupec ID, který bude reflektovat počet participantů.

cbind(Subset_ZŠ, ID = length(Subset_ZŠ$RWA_Skór))


ID_ZŠ = matrix(c(1, 2, 3))
ZŠ_Subset = cbind(ZŠ_Subset, ID_ZŠ)


SŠ_SubsetID =cbind(ID = 1:nrow(SŠ_Subset), SŠ_Subset)
```

Rekapitulace

Balíčky (dle Quick-R, n.d.)

Packages are collections of **R** functions, data, and compiled code in a well-defined format.

- The directory where packages are stored is called the library.
- **R** comes with a standard set of packages.
- Others are available for download and installation.
  - Once installed, they have to be loaded into the session to be used.

```
# get library location
.libPaths()

# see all packages installed
library()

# see packages currently loaded
search()
```

```
# nainstaluje konkrétní balíček
install.packages("psych")

# načte konkrétní balíček
library("psych")
```

Import dat

Obecně

Zjištění pracovní složky (get working directory)

getwd()

Nastavení pracovní složky (set working directory)

setwd("C:/Users/VG/Disk Google/Práce/MU/Výuka/FSS MU/PS2016/Praktický úvod pro používání statistického programu R/Lekce/3. VKLÁDÁNÍ A ČIŠTĚNÍ DAT, ZJIŠŤOVÁNÍ ZÁKLADNÍCH INFORMACÍ O DATOVÉM SOUBORU/Data")

nebo

setwd("C:\\Users\\VG\\Disk Google\\Práce\\MU\\Výuka\\FSS MU\\PS2016\\Praktický úvod pro používání statistického programu R\\Lekce\\3. VKLÁDÁNÍ A ČIŠTĚNÍ DAT, ZJIŠŤOVÁNÍ ZÁKLADNÍCH INFORMACÍ O DATOVÉM SOUBORU\\Data")

Import dat

Flat Files (= Prostý dabázový soubor)

= *Jednoduchá databáze* (většinou **tabulka**) uložená v **textovém souboru** ve formě **prostého textu** (Prostý databázový soubor, n.d.)

- .csv (comma-separated values)
- .txt

Existence celé řady balíčků odlišených podle preferovaného formátu (.csv, .txt) a míry automatizace (resp. počtu argumentů, které je třeba specifikovat).

Součástí R je balíček "**utils**":

- read.table(sep = "")
- read.csv(sep =)
- read.csv2(sep = ";")
- read.delim(sep = "\t")

?read.table

Import dat

Flat Files - Utils - .csv

```
# Import swimming_pools.csv:
pools = read.csv("swimming_pools.csv")


# Print the structure of pools
str(pools)

# Import swimming_pools.csv correctly: pools
pools = read.csv("swimming_pools.csv", stringsAsFactors = FALSE)

# Check the structure of pools
str(pools)
```

Import dat

Flat Files - Utils - .txt

```
hotdogs_1 = read.delim("hotdogs_1.txt", header = TRUE)
hotdogs_2 = read.delim("hotdogs_2.txt", header = FALSE, col.names = c("type", "calories", "sodium"))

summary(hotdogs_1)
str(hotdogs_1)

# Select the hot dog with the least calories: Cal
Cal <- hotdogs_1[which.min(hotdogs_1$Calories), ]

# Select the observation with the most sodium: Sod
Sod = hotdogs_1[which.max(hotdogs_1$Sodium), ]

str(hotdogs_1)
```

Import dat

Excel - readxl

```
# Instalace a nahrání balíčku
install.packages("readxl")
library("readxl")

# Dva základní příkazy:
excel_sheets() # Výčet listů v daném excelovském (.xls, .xlsx)  souboru
read_excel() # Načtení souboru excelovského formátu


excel_sheets("latitude.xlsx")
```

Import dat

Excel - readxl

```
# Read the first sheet of latitude.xlsx:
latitude_1 = read_excel("latitude.xlsx", sheet = "1700")
latitude_1


# Read the second sheet of latitude.xlsx:
latitude_2 = read_excel("latitude.xlsx", sheet = 2)
latitude_2


# Put latitude_1 and latitude_2 in a list:
lat_list = list(latitude_1, latitude_2)
```

Import dat

Excel - readxl - col_names

Apart from path and sheet, there are several other arguments you can specify in read_excel(). One of these arguments is called col_names.

```
# Import the the first Excel sheet of latitude_nonames.xlsx (R gives names):
latitude_3 = read_excel("latitude.xlsx", sheet = 3, col_names = FALSE)
latitude_3

# Import the the first Excel sheet of latitude_nonames.xlsx (specify col_names):
latitude_4 = read_excel("latitude.xlsx", sheet = 3, col_names = c("country", "latitude"))
latitude_4

# Print the summary of latitude_3
summary(latitude_3)

# Print the summary of latitude_4
summary(latitude_4)
```

Import dat

Excel - readxl - skip

Another argument that can be very useful when reading in Excel files that are less tidy, is skip.

- With skip, you can tell R to ignore a specified number of rows inside the Excel sheets you're trying to pull data from.

Have a look at this example:

read_excel("latitude.xlsx", **skip = 15**)

In this case, the first 15 rows in the first sheet of "data.xlsx" are ignored.

Pozor na posunutí matice!

read_excel("latitude.xlsx", skip = 15, **col_names = FALSE**)

Import dat

Excel - readxl - slučování listů do jedné matice a chybějící hodnoty

```
latitude_all <- cbind(latitude_1, latitude_2[-1])
latitude_all

# Argument [-1] se týká prvního sloupce v rámci dané matice

# Remove all rows with NAs from latitude_all
latitude_all_clean = na.omit(latitude_all)

# Print out a summary of latitude_all
summary(latitude_all_clean)
```

Import dat

SPSS - foreign

```
# Balíček foreign (základní součást R)
library("foreign")
```

K načtení dat z SPSS (.sav, .por) slouží příkaz read.spss()

- Aby měla nahraná data povahu data frame, je nutné uvnitř příkazu read.spss() jako argument zadat "to.data.frame = TRUE"

```
# Načtení dat
demo_1 = read.spss(".../international.sav", to.data.frame = TRUE)

# Načtení několika prvních řádků
head(demo_1)
```

Import dat

SPSS - foreign

Jak nastavit "value labels" z SPSS jako "factors" v R?

Skrze argument "**se.value.labels**" v rámci příkazu "**read.spss()**". Tento argument upřesňuje, zda mají být "value labels" konvertovány do R jako "factors".

- Argument je "TRUE by default", výchozím stavem je tedy provedení výše uvedené konverze

```
# Načtení dat
demo_2 = read.spss("…/international.sav", to.data.frame = TRUE, use.value.labels = FALSE)

# Načtení několika prvních řádků
head(demo_2)
```

Import dat

SPSS - foreign

Jak nastavit "value labels" z SPSS jako "factors" u dílčích proměnných v R?

```
# Summary demo_2$contint
summary(demo_2$contint)
class(demo_2$contint)

# Konverze demo_2$contint na faktor
demo_2$contint = as.factor(demo_2$contint)

# Summary demo_2$contint znovu
summary(demo_2$contint)
class(demo_2$contint)
```

Jak nastavit "value labels" z SPSS u "factors" v R u dílčích proměnných?

```
continents = c("Africa", "Americas", "Asia", "Europe")
demo_2$contint = factor(demo_2$contint, levels = c(1, 2, 3, 4), labels = continents)
summary(demo_2$contint)
```

Čištění dat

Explorace hrubých dat - base

```
# Matice
bmi_1 = read_excel("bmi.xlsx", sheet = 2)
```

```
# Check the class of bmi
class(bmi_1)
```

```
# Check the dimensions of bmi
dim(bmi_1)
```

```
# View the column names of bmi
colnames(bmi_1)
```

```
# Struktura dat
str(bmi_1)
```

```
# Sumarizace
Summary(bmi_1)
```

```
# Prvních 10 a posledních 10 řádků
head(bmi_1, n = 10)
tail(bmi_1, n = 10)
```

Čištění dat

Explorace hrubých dat - psych

```
# Load psych
install.packages("psych")
library("psych")

# Check the structure of bmi, the psych way
describe(bmi_1)
```

Čištění dat

Explorace hrubých dat - grafy

```
# Matice
bmi_2 = read_excel("bmi.xlsx", sheet = 3)
bmi_all = cbind(bmi_1, bmi_2[-1])

# Histogram
hist(bmi_1$BMI)

# Scatterplot
plot(bmi_all$BMI_1980, bmi_all$BMI_2000)
```

Čištění dat

Příprava dat pro analýzu

```
students = read.csv2("student.csv")

# Preview students with str()
str(students)

# Coerce failures to character
students$failures <- as.character(students$failures)

# Coerce Medu to factor
students$Medu <- as.factor(students$Medu)

# Coerce Fedu to factor
students$Fedu <- as.factor(students$Fedu)

# Look at students once more with str()
str(students)
```

Čištění dat

Příprava dat pro analýzu - dílčí manipulace se strings

```
# Load the stringr package
install.packages("stringr")
library("stringr")

# Trim all leading and trailing whitespace
name = c("    Filip ", "Nick  ", " Jonathan")
str_trim(name)

# Pad these strings with leading zeros
pad = c("23485W", "8823453Q", "994Z")
str_pad(pad, width = 9, side = "left", pad = "0")

# Print state abbreviations
latitude_1$country

# Make states all uppercase and save result to states_upper
states_upper = toupper(latitude_1$country)
states_upper

# Make states_upper all lowercase again
states_lower = tolower(latitude_1$country)
states_lower
```

Čištění dat

Příprava dat pro analýzu - dílčí manipulace se strings

```
# Look at the head of students
head(students)

# Detect all "health" in Mjob
str_detect(students$Mjob, "health")
```

```
# In the sex column, replace "F" with "Female"...
students$sex <- str_replace(students$sex, "F", "Female")

# ...And "M" with "Male"
students$sex <- str_replace(students$sex, "M", "Male")
```

Čištění dat

Příprava dat pro analýzu - missing values

```
name = c("Sára", "Tom", "David", "Alice")
n_friends = c(244, NA, 145, 43)
status = c("Going out!", "", "Movie night...", "")
social_df = data.frame(cbind(name, n_friends, status))
```

```
# Call is.na() on the full social_df to spot all NAs
is.na(social_df)

# Use the any() function to ask whether there are any NAs in the data
any(is.na(social_df))

# View a summary() of the dataset
summary(social_df)

# Call table() on the status column
table(social_df$status)
```

```
# Replace all empty strings in status with NA
social_df$status[social_df$status == ""] <- NA

# Print social_df to the console
social_df

# Use complete.cases() to see which rows have no missing values
complete.cases(social_df)

# Use na.omit() to remove all rows with any missing values
na.omit(social_df)
```

Čištění dat

Příprava dat pro analýzu - odlehlé a chybné hodnoty

```
# Look at a summary() of students
summary(students)

# View a histogram of the age variable
hist(students$age)

# View a histogram of absences, but force zeros to be bucketed to the right of zero
hist(students$age, right = FALSE)
```

Zdroje

Packages (n.d.) Packages. In Quick-R. Staženo dne 2. 10. 2016 z http://www.statmethods.net/interface/packages.html

Prostý databázový soubor. (n.d.). In Wikipedia. Staženo dne 2. 10. 2016 z https://cs.wikipedia.org/wiki/Prost%C3%BD_datab%C3%A1zov%C3%BD_soubor