

01. Programovací jazyk R a práce s ním

Vít Gabrhel

R101

2019-09-28

Harmonogram

- Organizace kurzu
- Představení jazyka R
- Instalace R a RStudio, orientace v prostředí RStudio, úvod do psaní kódu v R
- Základní operace – aritmetika, přiřazování hodnot proměnným, zjišťování třídy proměnných

Organizace kurzu

- Syllabus
- Web

Představení jazyka R

Co je R?

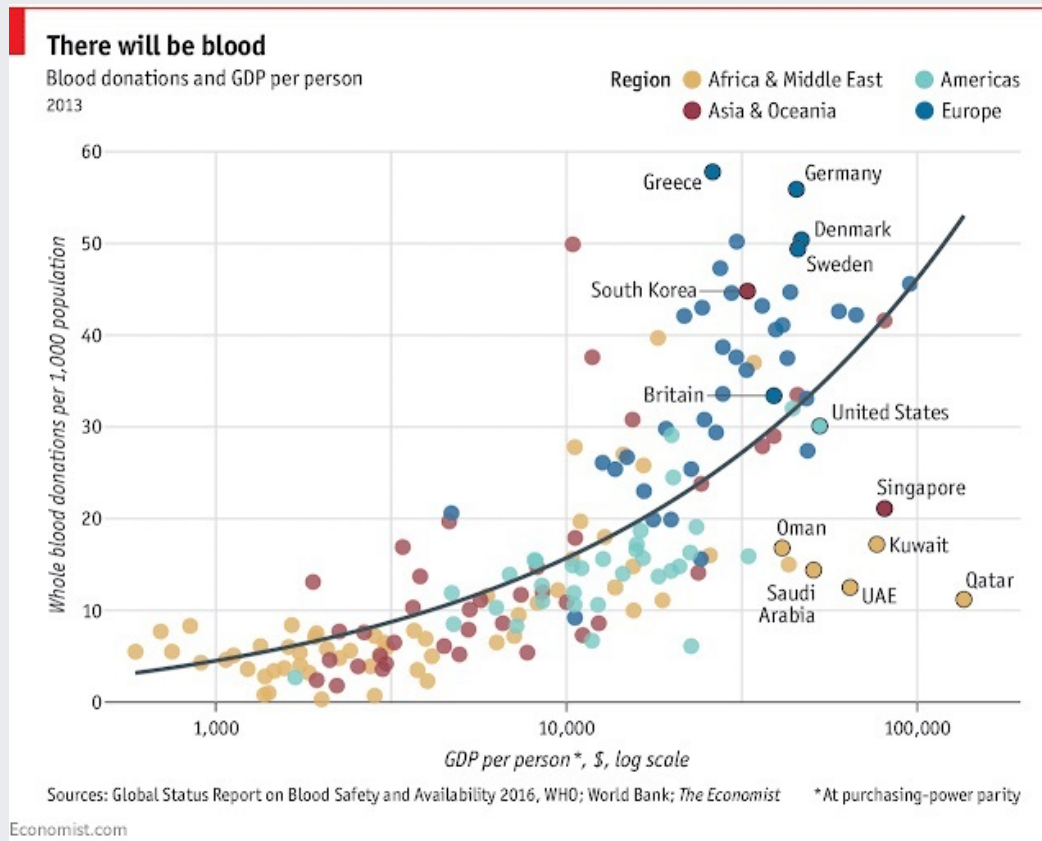
1) Programovací jazyk umožňující tvorbu "statistického softwaru" (tj. analýzy)

$$Y' = a + bx_1 + \dots + bx_n$$

```
MLR_Model_1 <- lm(y ~ x1 + x2, data=MLR_Data)
```

Co je R?

2) Prostředí na provádění analýz a vizualizací



Představení jazyka R

„Historie“

R je kombinací

- **Programovacího jazyka S**, určeného pro analýzu dat (Bell Labs, J. Chambers, 1976)
- **Lexikálního scopingu**, umožňujícího přiřazování dat k entitám označeným jménem

R vyvinuli R. Ihaka a R. Gentleman z University of Auckland – první verze byla spuštěna v roce **1994**

- Název je slovní hříčkou – jednak parafrází na „S“, ale i odkazem na iniciály obou autorů

Myšlenkové zázemí a současný kontext

Principy R

Open-Source a s ním související „svobody“

- Svoboda používat R pro jakékoliv účely, včetně svobody studovat to, jak R funguje a svoboda modifikovat ho ke svým vlastním potřebám
- Svoboda šířit R a v něm vytvořené obsahy
 - Svoboda zlepšovat R včetně sdílení těchto zlepšení komunitě
 - Open source
 - Transparentnost
 - Komunita

Bez záruk -> větší míra odpovědnosti na uživatele

Aktuální kontext:

- Principy R a Replikační krize

Getting started

- R
- RStudio
- Alternativy pro RStudio

Úvod do psaní kódu v R

Čísla a Texty

```
Bitcoin = 3 + 4  
CryptoCurrency <- c("Bitcoin", "Sichuan")  
names(CryptoCurrency) <- c("Kryptoměna", "Provincie")
```

U **textové proměnné** je nutné používat **uvozovky**.

```
Had <- "Proč neprogramuješ v Pythonu?"
```

Mřížka či hastag (#) odděluje kód od komentáře.

```
# Luděk má 3 BTC, Eleonora 4 ETH.  
# Dohromady mají 7 jednotek kryptoměn.  
CryptoCurrency = 3 + 4
```

Pozor na malá a velká písmena (= case sensitive)!

"=" a "<-" jsou ekvivalentní formy zápisu

Úvod do psaní kódu v R

Česká klávesnice, Windows a psaní symbolů (třeba "<") skrze pravý alt.

- Pravý alt + další písmeno na klávesnici, např.:
 - alt + . = >
 - alt + f = [
 - alt + v = @

`x = 3 + 4` versus `3 + 4 = y`

```
Error in 3 + 4 = y : target of assignment expands to non-language object
```

Parametr (například vektor nebo datová matice) nemůže v jazyce R začínat číslem.

Úvod do psaní kódu v R

Jak si "uklidit" pracovní prostor?

- Příkaz `remove()`

```
remove(CryptoCurrency)
```

Jak vyčistit konzoli?

- `Ctrl + l`

Jak si "říct o pomoc"?

- Symbol `"?"` před funkcí pro vyhledávání v dokumentaci R

```
?remove
```

- `"??"` před funkcí pro vyhledávání příkazu napříč knihovnamy

```
??remove
```

R jako kalkulačka

- Sčítání: "+"

```
420 + 0
```

```
## [1] 420
```

- Násobení: "*"

```
2 * 3
```

```
## [1] 6
```

- Umocňování: "^"

```
16 ^ 2
```

```
## [1] 256
```

R jako kalkulačka

- Odčítání: "-"

```
333 - 12
```

```
## [1] 321
```

- Dělení: "/"

```
1 / 2
```

```
## [1] 0.5
```

- Druhá odmocnina: `sqrt(x)`

```
sqrt(256)
```

```
## [1] 16
```

Objekty a třídy.

Reprezentují hodnoty (např. "0" nebo "Hruška") nebo jiné *objekty* (např. proměnná jako objekt v rámci datové matice, dalšího objektu).

- Skrze označení (**name**) je možné **vyvolat hodnotu** nebo **sadu hodnot** v tomto objektu.

```
Bitcoin <- 3  
Bitcoin
```

```
## [1] 3
```

```
Etherium <- 4  
Etherium
```

```
## [1] 4
```

```
Kryptoměny = Bitcoin + Etherium  
Kryptoměny
```

```
## [1] 7
```

Objekty a třídy.

Numerické versus stringové objekty

```
Bitcoin = Jedna  
Error: object 'Jedna' not found
```

```
Bitcoin = "Jedna"  
Kryptoměny = Bitcoin + Ethereum  
Error in Bitcoin + Ethereum : non-numeric argument to binary operator
```


Objekty a třídy.

Předchozí příklady se týkaly vyvolání hodnoty. Co vyvolání **sady hodnot**?

- Příkaz "c()" sloučí "argumenty" (např. 1, 0.5 nebo "Muž") do jednoho vektoru, tedy do sekvence dat

```
# Nejdříve vytvoříme objekt se zeměmi, kde se "těží" kryptoměny:  
Zeme_CC <- c("Čína", "ČR", "GB", "Čína")  
Zeme_CC
```

```
## [1] "Cína" "CR" "GB" "Cína"
```

```
# Názvy sítí, kde se sdružují těžaři:  
Network <- c("Antpool", "Slush", "Bitclub.Network", "Bixin")  
Network
```

```
## [1] "Antpool" "Slush" "Bitclub.Network" "Bixin"
```

```
# Jaký je podíl zmíněných sítí na celku?  
Share_in_Percents <- c(25, 3, 5, 8)  
Share_in_Percents
```

```
## [1] 25 3 5 8
```

Objekty a třídy.

- **Decimal** values like 4.5 are called **numerics**.
- **Natural numbers** like 4 are called **integers**.
 - Integers are also numerics.
- **Boolean** values (TRUE or FALSE) are called **logical**.
- **Text** (or **string**) values are called **characters**.
 - Uvozovky ("") indikují, že nějaký text je "character"

Objekty a třídy.

```
# Vytvořte objekt my_numeric s hodnotou 23.1  
print(my_numeric <- 23.1)
```

```
## [1] 23.1
```

```
# Vytvořte objekt my_character s hodnotou "universe"  
my_character <- "Čína"  
my_character
```

```
## [1] "Čína"
```

```
# Vytvořte objekt my_logical s hodnotou FALSE  
my_logical <- FALSE  
my_logical
```

```
## [1] FALSE
```

Objekty a třídy.

Logical

- Boolean values (TRUE or FALSE) are called **logical**.
 - `as.logical` - classifies **0** as **FALSE** and anything other than **0** as **TRUE**

```
Share_in_Percents = c(25, 3, 5, 8)
as.logical(Share_in_Percents)
```

```
## [1] TRUE TRUE TRUE TRUE
```

```
# is.logical - Create or test for objects of type "logical"
is.logical(Share_in_Percents)
```

```
## [1] FALSE
```

Třídy objektů

Jak poznat třídu dat?

- Funkce `class()`:

```
class(my_numeric)
```

```
## [1] "numeric"
```

```
class(my_character)
```

```
## [1] "character"
```

```
class(my_logical)
```

```
## [1] "logical"
```

Třídy objektů

- Factor

```
treatment <- c("A", "A", "B", "B", "Placebo", "Placebo")  
class(treatment)
```

```
## [1] "character"
```

```
is.factor(treatment)
```

```
## [1] FALSE
```

```
treatment.f <- as.factor(treatment)  
class(treatment.f)
```

```
## [1] "factor"
```

```
is.factor(treatment.f)
```

```
## [1] TRUE
```

Doporučené zdroje

- **DataCamp** - *learning*
- **GitHub** - *code management*
- **R-bloggers** - *inspirace*
- **Kaggle** - *data sources*