

# Survival kit

## Special characters

Some characters are somewhat difficult to write on the Czech keyboard:

| Character | Shortcut  | Alterantive     |
|-----------|-----------|-----------------|
| #         | Alt + 35  | Alt + x         |
| \$        | Alt + 36  | Alt + û         |
| &         | Alt + 38  | Alt + c         |
| ~         | Alt + 126 | Alt + +         |
|           |           | Alt + w         |
| >         |           | Alt + .         |
| }         |           | Alt + b/n       |
| []        |           | Alt + f/g       |
| ‘         |           | Alt + ý + Space |
| ^         | Alt + 94  |                 |

## RStudio shortcuts

| Sign/Action      | Shortcut             |
|------------------|----------------------|
| ->               | Ctrl + Alt + -       |
| %>%              | Ctrl + Shift + m     |
| run selected row | Ctrl + Enter         |
| run whole script | Ctrl + Shift + Enter |
| comment line     | Ctrl + Shift + c     |
| insert section   | Ctrl + Shift + r     |

## Function reference

code description

## Basics

`install.packages("packagename")` installs a package  
`library(packagename)` attaches a package from a library

Getting help on anything:

`?function` equals `help(function)`

`??function`

Lookign at data:

`head(x)` and `tail(x)`

`View(x)`

## Summary functions

`mean(x)` returns mean `meadian(x)` returns median

## Data manipulation with dplyr

load package `library(dplyr)`

`%>%` *pipe* operator, `x %>% f()` equals `f(x)`, allows chaining functions in a pipeline  
(in newer versions of R, there is a native pipe operator `|>`)

`count(data)` count how many observations there are in a data frame

`summarise(data,...)` applies summary functions to a data frame

`group_by(data, x)` group *data* by values in variable *x*

Chain this for example like this:

```
data %>%  
  group_by(y) %>%  
  summarise(prumer = mean(x),  
            median = median(x))
```

## Plotting with ggplot2

Load package `library(ggplot2)`.

Alert: in `ggplot2`, individual pieces of code are chained using `+`!

`ggplot()` basic function call for each plot

`aes(x, y, color, fill...)` aesthetics

## Geometries

`geom_histogram()` produces histogram

`geom_point()` produces scatterplot

Create a plot like this:

```
data %>%  
  ggplot() +  
  aes(x) +  
  geom_histogram()
```