

Packages

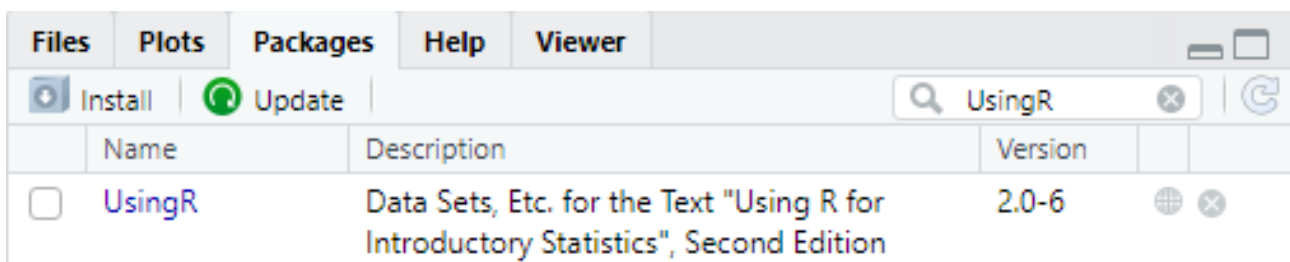
2020

Installing packages

R has a rich collection of packages. They allow researchers to share R functions and data. You can install packages available in the **CRAN** repository using `install.packages`. We already learned how we can get help

```
> ?install.packages  
> help("install.packages")
```

You can also install packages from the R Studio interface



For the next tasks you will need to install `UsingR` and `MASS` packages available in the CRAN repository.

```
> install.packages("UsingR")  
> install.packages("MASS")
```

We can also install packages from **GitHub** using `install_github`. But first you need to install `devtools` from CRAN.

Loading packages

Just because a package is installed on your computer does not mean that you have access to its functions. We need to load the package into the current R session. Getting help

```
> ?library  
Load the package in the memory
```

```
> library(UsingR)
Warning: package 'UsingR' was built under R version 4.0.3
> library(MASS)
```

Session info

You can see information about your current session - R version, OS, attached or loaded packages and their versions using `sessionInfo`.

```
> sessionInfo()
R version 4.0.2 (2020-06-22)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19041)
```

```
Matrix products: default
```

```
locale:
[1] LC_COLLATE=Bulgarian_Bulgaria.1251
LC_CTYPE=Bulgarian_Bulgaria.1251
[3] LC_MONETARY=Bulgarian_Bulgaria.1251 LC_NUMERIC=C
[5] LC_TIME=Bulgarian_Bulgaria.1251
```

```
attached base packages:
[1] stats      graphics  grDevices  utils      datasets
methods     base
```

```
other attached packages:
[1] UsingR_2.0-6      Hmisc_4.4-1      ggplot2_3.3.2
Formula_1.2-3
[5] survival_3.1-12  lattice_0.20-41  HistData_0.8-6
MASS_7.3-51.6
```

```
loaded via a namespace (and not attached):
[1] tidyselect_1.1.0      xfun_0.16          purrr_0.3.4
[4] splines_4.0.2         colorspace_1.4-1   vctrs_0.3.4
[7] generics_0.0.2        htmltools_0.5.0    yaml_2.2.1
[10] base64enc_0.1-3       rlang_0.4.7        pillar_1.4.6
[13] foreign_0.8-80        glue_1.4.2         withr_2.3.0
[16] RColorBrewer_1.1-2    jpeg_0.1-8.1
lifecycle_0.2.0
[19] stringr_1.4.0         munsell_0.5.0      gtable_0.3.0
```


Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0
1 4 4								
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1
1 4 1								
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1
0 3 1								
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0
0 3 2								
Valiant	18.1	6	225	105	2.76	3.460	20.22	1
0 3 1								

Writing `?` before the name of the data frame gives information about the data frame, if it is available.

```
> ?mtcars
```

You can see what is this data frame about and some information about the data contained in the columns.

As we already saw we can use `dim`, `nrow` and `ncol` to show the dimensions of the data frame

```
> dim(mtcars)
[1] 32 11
> nrow(mtcars)
[1] 32
> ncol(mtcars)
[1] 11
```

Next different ways to take only part of the data frame are reviewed. Shows the element in the 1 row, 2 column

```
> mtcars[1, 2]
[1] 6
```

Shows the element in the “Mazda RX4” row, “cyl” column

```
> mtcars["Mazda RX4", "cyl"]
[1] 6
```

Shows the elements in the 9 column

```
> mtcars[[9]]
[1] 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1
1 1 1 1 1 1
```

Shows the element in the “am” column

```
> mtcars[["am"]]
[1] 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1
1 1 1 1 1 1
```

Shows the element in the “am” column

```
> mtcars$am
[1] 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1
1 1 1 1 1 1
```

Attach data frame

Taking the names of the columns in the data frame

```
> names(mtcars)
[1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec"
"vs" "am" "gear"
[11] "carb"
```

Let's say we want to take the mpg column, we can use \$ after the data frame's name

```
> mtcars$mpg
[1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2
17.8 16.4 17.3 15.2 10.4
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2
27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4
```

or to attach the data frame and only to write the name of the column.

```
> attach(mtcars)
> mpg
[1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2
17.8 16.4 17.3 15.2 10.4
```

```
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2
27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4
```

`attach` loads the data frame, so it is not necessary to write the data frame's name in front of the column name.

Getting help

```
> ?attach
```

Sometime this is very helpful, but it is important to notice that if you change the attached object the data frame is not going to change. For example let's change the first value

```
> mpg[1] = 200
> mpg
 [1] 200.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4
22.8 19.2 17.8 16.4
[13] 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9
21.5 15.5 15.2 13.3
[25] 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
> mtcars$mpg
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2
17.8 16.4 17.3 15.2 10.4
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2
27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4
```