Visualization, transformation and reporting with the tidyverse

Kirill Müller, Tobias Schieferdecker, Patrick Schratz

25 November 2019, 18:55 CET

Contents

Preface **5**

Links 5
Package versions used 5
License 8
Speakers 8
Introduction & overview 10

I R & RStudio **11**

1 R **13**

1.1 R as a toolkit 13
1.2 R code examples 15
1.3 R vs. RStudio 16
1.4 R vs. R packages 17
1.5 R packages 18
1.6 .Rprofile 18
1.7 .Renviron 19

2 RStudio **21**

2.1 IDE structure 21
2.2 RStudio Addins 28
2.3 RStudio projects 28
2.4 Alternatives to RStudio 29

II Visualization **31**

3 {ggplot2} basics **33**

4 {ggplot2} advanced **35**

III Tidying and transforming 37

5 Import 39

6 Transformation 41

7 Tidying 43

IV Reporting 45

V Appendix 47

8 Best practices 49
8.1 DESCRIPTION 49
8.2 R 50
8.3 roxygen2 51
8.4 testthat 51

9 53

Preface

See the controls at the top of the website for searching, font size, editing, and a link to the PDF version of the material.

Links

- This website: https://krlmlr.github.io/vistransrep/book
- Scripts and installation instructions: https://github.com/krlmlr/vistransrep-proj/tree/master
    - Prepared scripts: https://github.com/krlmlr/vistransrep-proj/tree/master/script
- The source project for this material: https://github.com/krlmlr/vistransrep

Package versions used

Click to expand

```
withr::with_options(list(width = 80), print(sessioninfo::session_info()))
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 3.6.1 (2017-01-27)
##  os       Ubuntu 16.04.6 LTS
##  system   x86_64, linux-gnu
##  ui       X11
##  language en_US.UTF-8
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       UTC
##  date     2019-11-25
```

```
##
## - Packages -------------------------------------------------------------------
##  package      * version     date       lib source
##  askpass        1.1         2019-01-13 [1] CRAN (R 3.6.1)
##  assertthat     0.2.1       2019-03-21 [1] CRAN (R 3.6.1)
##  backports      1.1.5       2019-10-02 [1] CRAN (R 3.6.1)
##  bookdown       0.16        2019-11-22 [1] CRAN (R 3.6.1)
##  broom          0.5.2       2019-04-07 [1] CRAN (R 3.6.1)
##  cellranger     1.1.0       2016-07-27 [1] CRAN (R 3.6.1)
##  cli            1.1.0       2019-03-19 [1] CRAN (R 3.6.1)
##  codetools      0.2-16      2018-12-24 [3] CRAN (R 3.6.1)
##  colorspace     1.4-1       2019-03-18 [1] CRAN (R 3.6.1)
##  crayon         1.3.4       2017-09-16 [1] CRAN (R 3.6.1)
##  crosstalk      1.0.0       2016-12-21 [1] CRAN (R 3.6.1)
##  data.table     1.12.6      2019-10-18 [1] CRAN (R 3.6.1)
##  DBI            1.0.0       2018-05-02 [1] CRAN (R 3.6.1)
##  dbplyr         1.4.2       2019-06-17 [1] CRAN (R 3.6.1)
##  digest         0.6.23      2019-11-23 [1] CRAN (R 3.6.1)
##  dplyr        * 0.8.3       2019-07-04 [1] CRAN (R 3.6.1)
##  DT             0.10        2019-11-12 [1] CRAN (R 3.6.1)
##  ellipsis       0.3.0       2019-09-20 [1] CRAN (R 3.6.1)
##  evaluate       0.14        2019-05-28 [1] CRAN (R 3.6.1)
##  fansi          0.4.0       2018-10-05 [1] CRAN (R 3.6.1)
##  farver         2.0.1       2019-11-13 [1] CRAN (R 3.6.1)
##  fastmap        1.0.1       2019-10-08 [1] CRAN (R 3.6.1)
##  forcats      * 0.4.0       2019-02-17 [1] CRAN (R 3.6.1)
##  fs             1.3.1       2019-05-06 [1] CRAN (R 3.6.1)
##  generics       0.0.2       2018-11-29 [1] CRAN (R 3.6.1)
##  ggplot2      * 3.2.1       2019-08-10 [1] CRAN (R 3.6.1)
##  ggpubr         0.2.4       2019-11-14 [1] CRAN (R 3.6.1)
##  ggsignif       0.6.0       2019-08-08 [1] CRAN (R 3.6.1)
##  git2r          0.26.1      2019-06-29 [1] CRAN (R 3.6.1)
##  glue           1.3.1       2019-03-12 [1] CRAN (R 3.6.1)
##  gtable         0.3.0       2019-03-25 [1] CRAN (R 3.6.1)
##  haven          2.2.0       2019-11-08 [1] CRAN (R 3.6.1)
##  here         * 0.1         2017-05-28 [1] CRAN (R 3.6.1)
##  hms            0.5.2       2019-10-30 [1] CRAN (R 3.6.1)
##  htmltools      0.4.0       2019-10-04 [1] CRAN (R 3.6.1)
##  htmlwidgets    1.5.1       2019-10-08 [1] CRAN (R 3.6.1)
##  httpuv         1.5.2       2019-09-11 [1] CRAN (R 3.6.1)
##  httr           1.4.1       2019-08-05 [1] CRAN (R 3.6.1)
##  jsonlite       1.6         2018-12-07 [1] CRAN (R 3.6.1)
##  knitr          1.26        2019-11-12 [1] CRAN (R 3.6.1)
##  labeling       0.3         2014-08-23 [1] CRAN (R 3.6.1)
##  later          1.0.0       2019-10-04 [1] CRAN (R 3.6.1)
##  lattice        0.20-38     2018-11-04 [3] CRAN (R 3.6.1)
##  lazyeval       0.2.2       2019-03-15 [1] CRAN (R 3.6.1)
##  leaflet      * 2.0.3       2019-11-16 [1] CRAN (R 3.6.1)
##  lifecycle      0.1.0       2019-08-01 [1] CRAN (R 3.6.1)
##  lubridate      1.7.4       2018-04-11 [1] CRAN (R 3.6.1)
##  magrittr       1.5         2014-11-22 [1] CRAN (R 3.6.1)
##  MASS           7.3-51.4    2019-03-31 [3] CRAN (R 3.6.1)
##  memoise        1.1.0       2017-04-21 [1] CRAN (R 3.6.1)
##  mime           0.7         2019-06-11 [1] CRAN (R 3.6.1)
##  modelr         0.1.5       2019-08-08 [1] CRAN (R 3.6.1)
##  munsell        0.5.0       2018-06-12 [1] CRAN (R 3.6.1)
##  nlme           3.1-140     2019-05-12 [3] CRAN (R 3.6.1)
##  nycflights13 * 1.0.1       2019-09-16 [1] CRAN (R 3.6.1)
##  openssl        1.4.1       2019-07-18 [1] CRAN (R 3.6.1)
##  pillar         1.4.2       2019-06-29 [1] CRAN (R 3.6.1)
##  pkgconfig      2.0.3       2019-09-22 [1] CRAN (R 3.6.1)
##  plotly         4.9.1       2019-11-07 [1] CRAN (R 3.6.1)
##  plyr           1.8.4       2016-06-08 [1] CRAN (R 3.6.1)
##  promises       1.1.0       2019-10-04 [1] CRAN (R 3.6.1)
##  purrr        * 0.3.3       2019-10-18 [1] CRAN (R 3.6.1)
##  R6             2.4.1       2019-11-12 [1] CRAN (R 3.6.1)
##  RColorBrewer   1.1-2       2014-12-07 [1] CRAN (R 3.6.1)
##  Rcpp           1.0.3       2019-11-08 [1] CRAN (R 3.6.1)
##  readr        * 1.3.1       2018-12-21 [1] CRAN (R 3.6.1)
##  readxl         1.3.1       2019-03-13 [1] CRAN (R 3.6.1)
##  reprex         0.3.0       2019-05-16 [1] CRAN (R 3.6.1)
##  reshape2       1.4.3       2017-12-11 [1] CRAN (R 3.6.1)
##  rlang          0.4.2.9000  2019-11-25 [1] Github (r-lib/rlang@26bf207)
##  rmarkdown      1.17        2019-11-13 [1] CRAN (R 3.6.1)
##  rprojroot      1.3-2       2018-01-03 [1] CRAN (R 3.6.1)
##  rstudioapi     0.10        2019-03-19 [1] CRAN (R 3.6.1)
##  rvest          0.3.5       2019-11-08 [1] CRAN (R 3.6.1)
##  scales         1.1.0       2019-11-18 [1] CRAN (R 3.6.1)
##  sessioninfo    1.1.1       2018-11-05 [1] CRAN (R 3.6.1)
##  shiny          1.4.0       2019-10-10 [1] CRAN (R 3.6.1)
##  stringi        1.4.3       2019-03-12 [1] CRAN (R 3.6.1)
##  stringr      * 1.4.0       2019-02-10 [1] CRAN (R 3.6.1)
##  tibble       * 2.1.3       2019-06-06 [1] CRAN (R 3.6.1)
##  tic            0.2.13.9021 2019-11-18 [1] Github (ropenscilabs/tic@9a5f965)
##  tidyr        * 1.0.0       2019-09-11 [1] CRAN (R 3.6.1)
##  tidyselect     0.2.5       2018-10-11 [1] CRAN (R 3.6.1)
##  tidyverse    * 1.3.0       2019-11-21 [1] CRAN (R 3.6.1)
##  utf8           1.1.4       2018-05-24 [1] CRAN (R 3.6.1)
##  vctrs          0.2.0       2019-07-05 [1] CRAN (R 3.6.1)
##  viridisLite    0.3.0       2018-02-01 [1] CRAN (R 3.6.1)
##  withr          2.1.2       2018-03-15 [1] CRAN (R 3.6.1)
##  xaringan       0.13        2019-10-30 [1] CRAN (R 3.6.1)
##  xfun           0.11       2019-11-12 [1] CRAN (R 3.6.1)
##  xml2           1.2.2      2019-08-09 [1] CRAN (R 3.6.1)
##  xtable         1.8-4      2019-04-21 [1] CRAN (R 3.6.1)
##  yaml           2.2.0      2018-07-25 [1] CRAN (R 3.6.1)
##  zeallot        0.1.0      2018-01-28 [1] CRAN (R 3.6.1)
##
## [1] /home/travis/R/Library
## [2] /usr/local/lib/R/site-library
## [3] /home/travis/R-bin/lib/R/library
```

License

Speakers

Kirill Müller (@krlmlr)

Patrick Schratz (@pat-s)

- M.Sc. Geoinformatics
- Researcher/Research Engineer at University of **Jena** and **LMU Munich**
- PhD Candidate

- Unix & R enthusiast
- Author/Contributor/Maintainer of several R packages:
  - (mlr3, mlr)
  - sperrorest
  - oddsratio
  - xaringan
  - circle
  - RQGIS
  - travis
  - tic
  - ...

Introduction & overview

The `tidyverse` has quickly developed over the last years. Its first implementation as a collection of partly older packages was in the second half of 2016. All its packages "share an underlying design philosophy, grammar, and data structures."[1] It is for sure difficult to tell, if "learning the `tidyverse`" is a hard task, since the result of this assessment might differ from person to person. We do believe though, that there are concepts in its approach, which – when grasped – have the potential to increase one's productivity, since code creation will seem more natural. While this might be true for all languages (once you speak it well enough, things go smoothly), in our opinion the `tidyverse` worth exploring in depth, since it is

1. consistent: an especially well designed framework that aims at making data analysis and programming intuitive,
2. evolving: constantly deepened understanding for challenges arising in modern data analysis leads to improving ergonomic user interfaces.

This course covers several topics, which everyone working more intently with the `tidyverse` almost inevitably needs to deal with at some point or another. The topics are organized in chapters that contain mostly R code with output and text. In each section, exercises are provided.

[1] citation from tidyverse homepage

Part I

R & RStudio

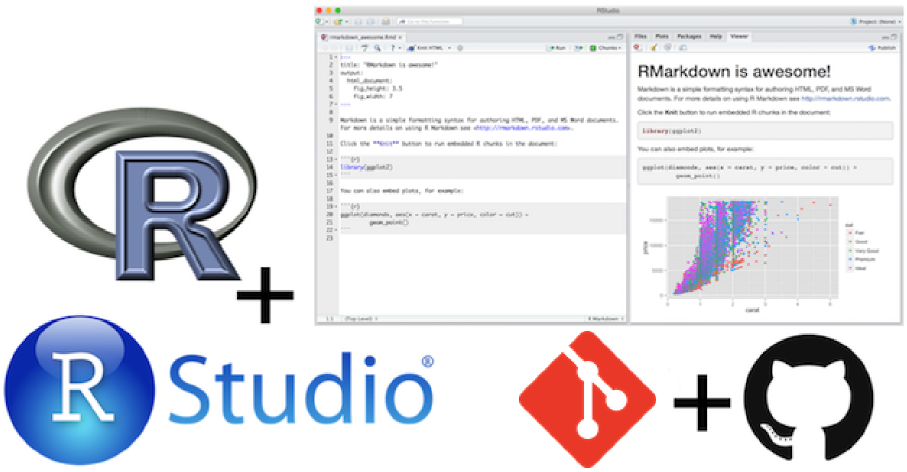
Chapter 1

R

1.1 R as a toolkit

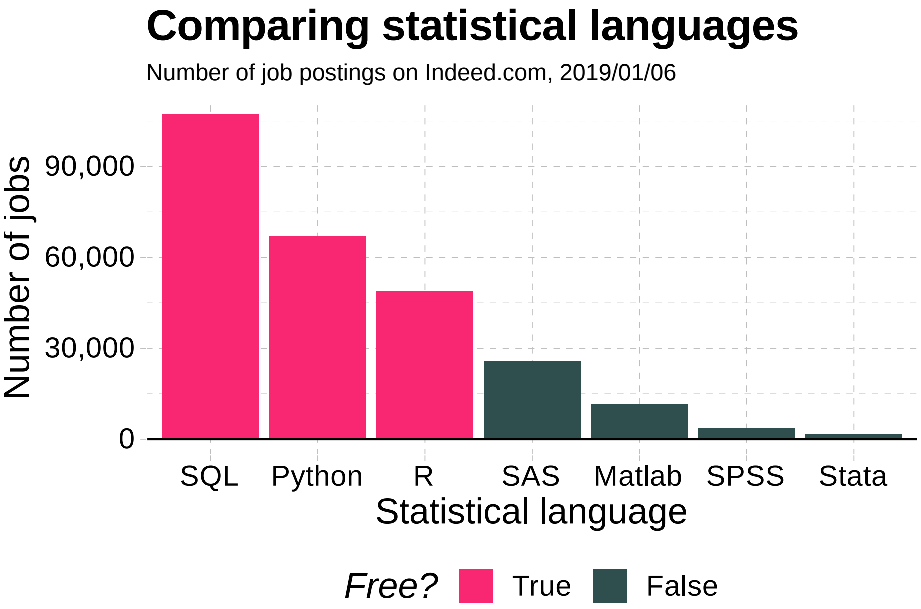
- Scriptability → R
- Literate programming (code, narrative, output in one place) → R Markdown
- Version control → Git / GitHub

1.1.1 Why R and RStudio?



~~~
---
title: "RMarkdown is awesome!"
output:
  html_document:
    fig_height: 3.5
    fig_width: 7
---

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

Click the **Knit** button to run embedded R chunks in the document:

```{r}
library(ggplot2)
~~~

You can also embed plots, for example:

```{r}
ggplot(diamonds, aes(x = carat, y = price, color = cut)) +
        geom_point()
```
```

RMarkdown is awesome!

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Click the **Knit** button to run embedded R chunks in the document:

```
library(ggplot2)
```

You can also embed plots, for example:

```
ggplot(diamonds, aes(x = carat, y = price, color = cut)) +
        geom_point()
```

Figure 1.1: R as a toolkit



1.1.2 Some R basics

- You will load packages at the **start of every new R session**.
  - "Base" R comes with tons of useful built-in functions. It also provides all the tools necessary for you to write your own functions.
  - However, many of R's best data science functions and tools come
```

    from external packages written by other users.

- R easily and infinitely parallelizes. For free.
    - Compare the cost of a Stata/MP license, nevermind the fact that you effectively pay per core…

## 1.2 R code examples

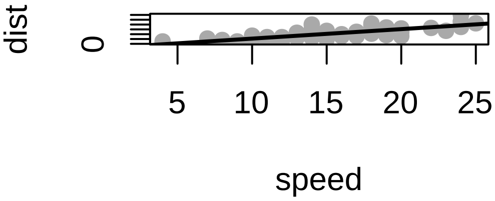
### 1.2.1 Linear regression

```
fit <- lm(dist ~ 1 + speed, data = cars)
summary(fit)
```

```
##
## Call:
## lm(formula = dist ~ 1 + speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

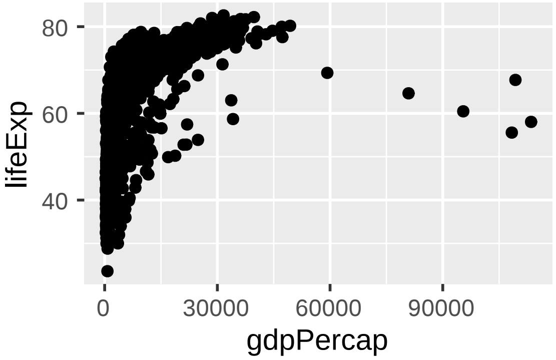
1.2.2 Base R plot

```
plot(cars, pch = 19, col = "darkgray")
abline(fit, lwd = 2)
```



1.2.3 ggplot2

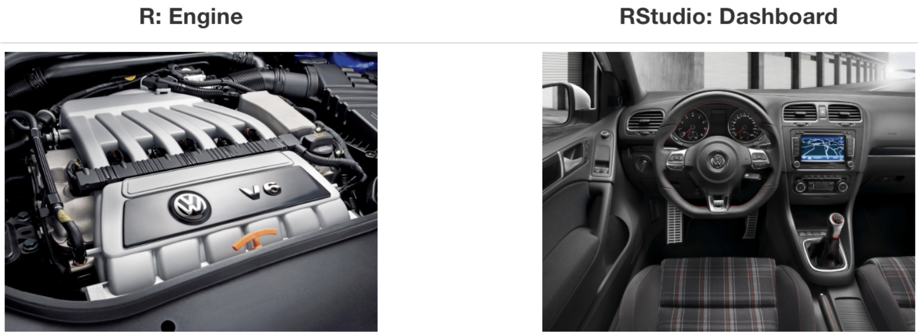
```
library(ggplot2)
library(gapminder) ## For the gapminder data

ggplot(
  data = gapminder,
  mapping = aes(x = gdpPercap, y = lifeExp)
) +
  geom_point()
```



1.2.4 gganimate

1.3 R vs. RStudio

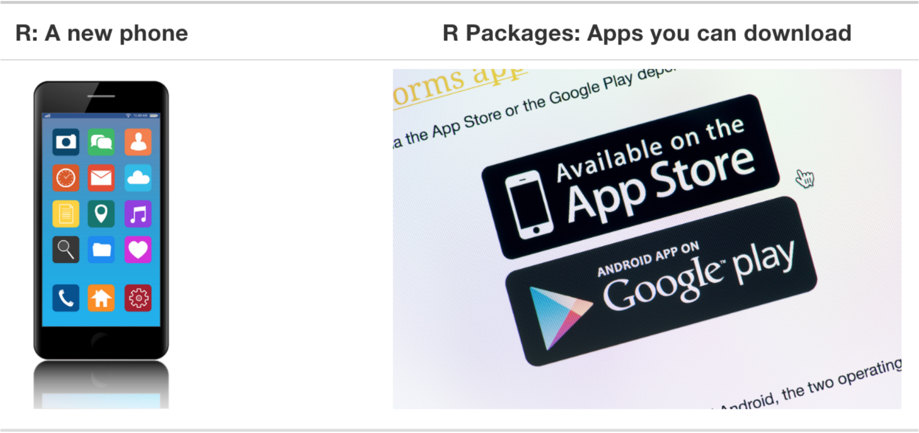
- R is a statistical **programming language**



R: Engine

RStudio: Dashboard

Figure 1.2: Engine vs. dashboard



R: A new phone

R Packages: Apps you can download

Figure 1.3: R versus R packages

- RStudio is a convenient interface for R (an **integrated development environment**, IDE)
- At its simplest:
  - R is like a car's engine
  - RStudio is like a car's dashboard

1.4 R vs. R packages

- R packages **extend** the functionality of R by providing additional functions, data, and documentation.

- They are written by a world-wide community of R users and can be downloaded for no cost

1.5 R packages

- **CRAN**: A group of people who check that packages fulfill certain standards
- **Mirror**: A location on the web where to download R packages from. Because many thousand people download them daily, the load is distributed on different machines. Pick one which is geographically close to you
- **R base/recommended packages**: The base installation of R ships with a bunch of default packages. In addition, there are some more packages listed as "recommended".

"base" packages are managed by the R core team and will only be updated for every R release.

Packages listed as "recommended" inherit the attributes of being widely used and having a long history in the R community.

```
##     Package Priority
## 1      base     base
## 2  compiler     base
## 3  datasets     base
## 4  graphics     base
## 5 grDevices     base
## 6      grid     base
## 7   methods     base
## 8  parallel     base
```

```
##       Package    Priority
## 1        boot recommended
## 2       class recommended
## 3     cluster recommended
## 4   codetools recommended
## 5     foreign recommended
## 6  KernSmooth recommended
## 7     lattice recommended
## 8        MASS recommended
## 9      Matrix recommended
## 10       mgcv recommended
##  [ reached 'max' / getOption("max.print") -- omitted 2 rows ]
```

1.6 .Rprofile

- File in your home directory `~/.Rprofile`
- Will be executed before every R session starts
- Useful to set global options and for loading of often used packages

1.7 .Renviron

- File in your home directory `~/.Renviron`
- Used to set environment variables
- Used to store “Access tokens” (Github, CI provider, C++ flags)

Chapter 2

RStudio

2.1 IDE structure

→ Exists to **boost** your productivity

→ Change the defaults to your liking so you *actually* can be **productive**

→ Keybindings = productivity

Since RStudio v1.3 a portable JSON settings file exists.

If you want to have sane settings without much hassle, you can execute the following R code: `source("https://bit.ly/rstudio-pat")`

This code will change/overwrite your existing RStudio settings and

- set custom keybindings
- move the console panel to the top-right (by default bottom-left)
- Enable/Disable some core settings to have a better overall experience

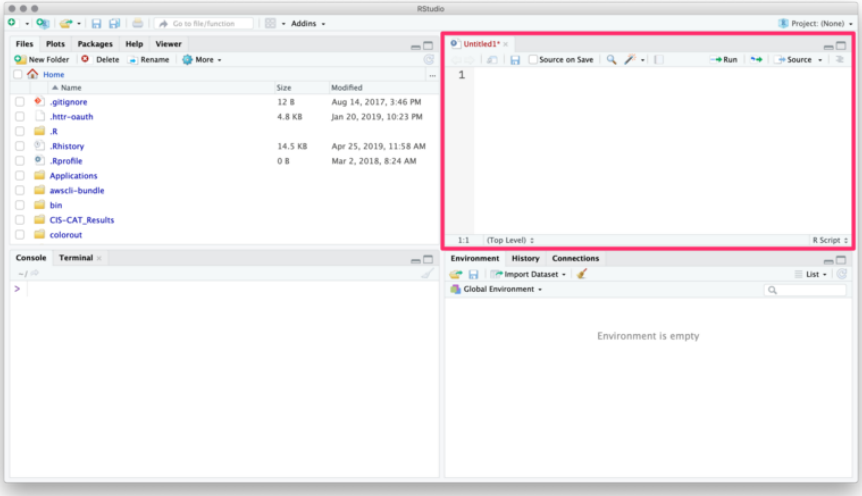
---

R scripts (source code) are written in the *Source* pane (Editor).

(Source of all following RStudio screenshots: https://github.com/edrubin/EC525S19)

You can use the menubar or ++N / +CTRL+N to create new R scripts.

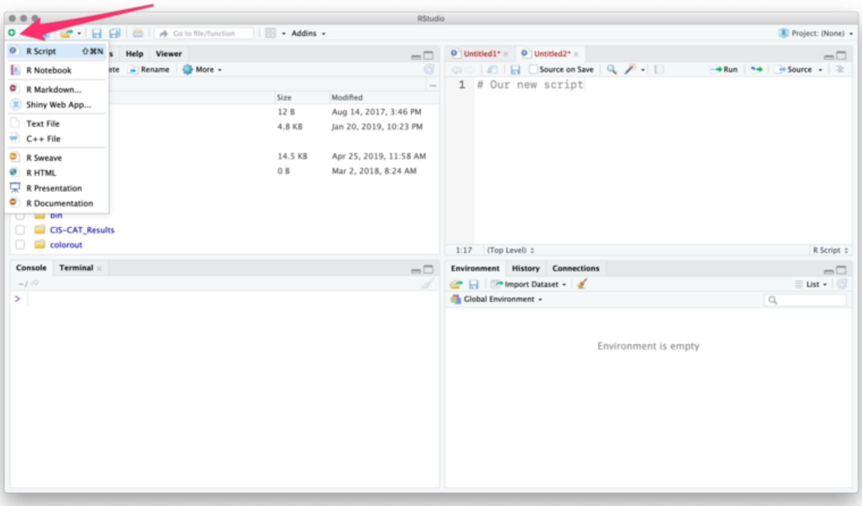
To execute commands from your R script, use +Enter / CTRL+Enter.



Name	Size	Modified
.gitignore	12 B	Aug 14, 2017, 3:46 PM
.httr-oauth	4.8 KB	Jan 20, 2019, 10:23 PM
.R		
.Rhistory	14.5 KB	Apr 25, 2019, 11:58 AM
.Rprofile	0 B	Mar 2, 2018, 8:24 AM
Applications		
awscli-bundle		
bin		
CIS-CAT_Results		
colorout		

Environment is empty

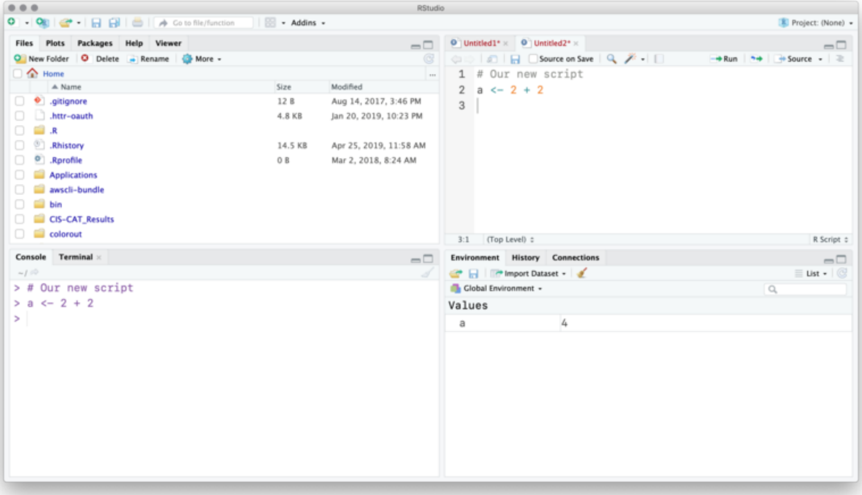
Figure 2.1: Source pane



```
# Our new script
```

Environment is empty

Figure 2.2: New script



Name	Size	Modified
.gitignore	12 B	Aug 14, 2017, 3:46 PM
.httr-oauth	4.8 KB	Jan 20, 2019, 10:23 PM
.R		
.Rhistory	14.5 KB	Apr 25, 2019, 11:58 AM
.Rprofile	0 B	Mar 2, 2018, 8:24 AM
Applications		
awscli-bundle		
bin		
CIS-CAT_Results		
colorout		

```
# Our new script
a <- 2 + 2
```

```
> # Our new script
> a <- 2 + 2
>
```

Values

a	4

Figure 2.3: Execute commands

RStudio will execute the command in the console.

You can see the new object in the *Environment* pane.

The *History* tab records your old commands.

The *Files* pane is the file explorer.

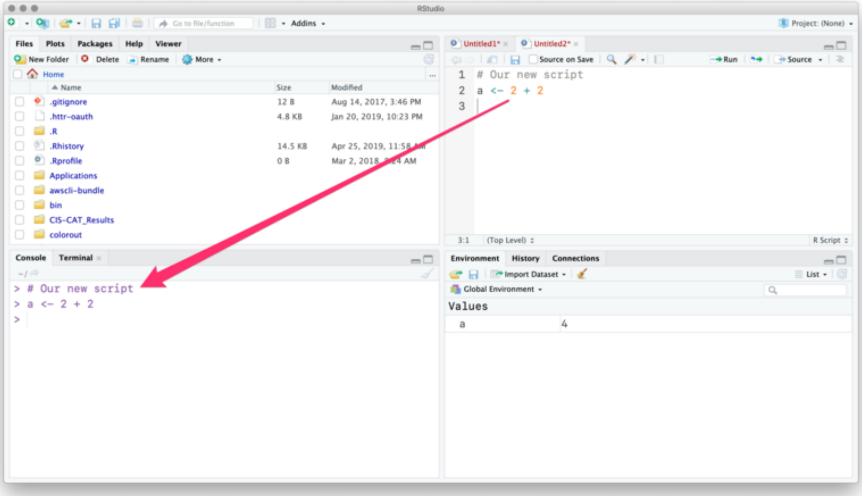
The *Plots* pane/tab shows… plots.

Packages shows installed packages

Packages shows installed packages and whether they are *loaded*.

The *Help* tab shows help documentation (also accessible via `?`).

Finally, you can customize the actual layout



Name	Size	Modified
.gitignore	12 B	Aug 14, 2017, 3:46 PM
.httr-oauth	4.8 KB	Jan 20, 2019, 10:23 PM
.R		
.Rhistory	14.5 KB	Apr 25, 2019, 11:58 AM
.Rprofile	0 B	Mar 2, 2018, 8:24 AM
Applications		
awscli-bundle		
bin		
CIS-CAT_Results		
colorout		

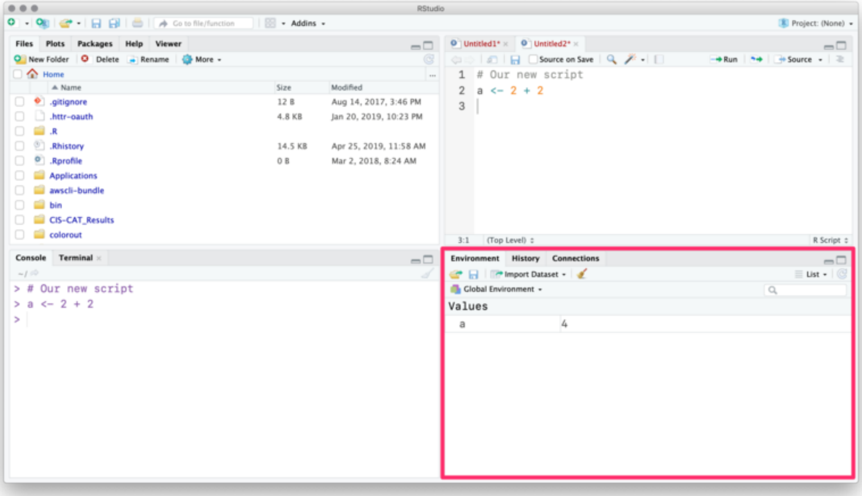
```
# Our new script
a <- 2 + 2
```

```
> # Our new script
> a <- 2 + 2
>
```

Values

a	4

Figure 2.4: Console output



Name	Size	Modified
.gitignore	12 B	Aug 14, 2017, 3:46 PM
.httr-oauth	4.8 KB	Jan 20, 2019, 10:23 PM
.R		
.Rhistory	14.5 KB	Apr 25, 2019, 11:58 AM
.Rprofile	0 B	Mar 2, 2018, 8:24 AM
Applications		
awscli-bundle		
bin		
CIS-CAT_Results		
colorout		

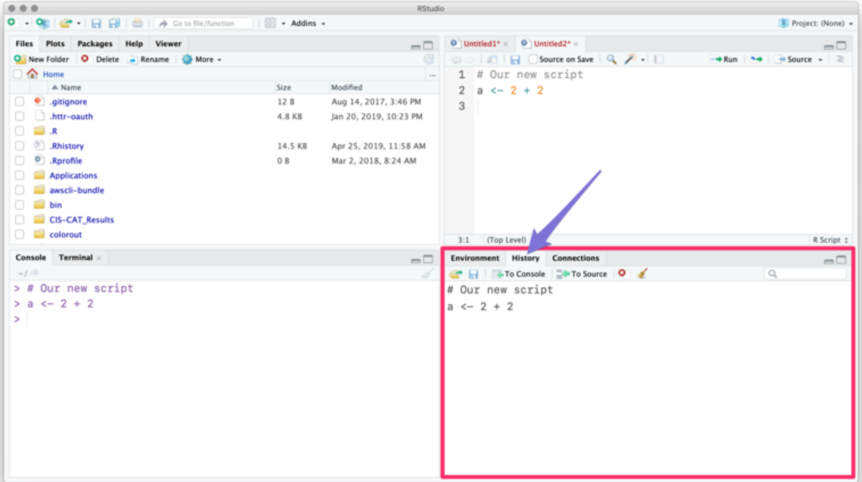
```
# Our new script
a <- 2 + 2
```

```
> # Our new script
> a <- 2 + 2
>
```

Values

a	4

Figure 2.5: Environment pane



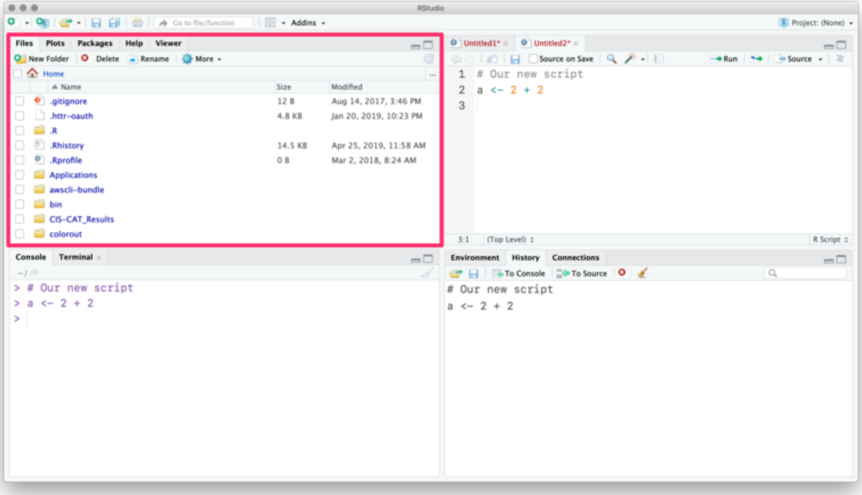
Name	Size	Modified
.gitignore	12 B	Aug 14, 2017, 3:46 PM
.httr-oauth	4.8 KB	Jan 20, 2019, 10:23 PM
.R		
.Rhistory	14.5 KB	Apr 25, 2019, 11:58 AM
.Rprofile	0 B	Mar 2, 2018, 8:24 AM
Applications		
awscli-bundle		
bin		
CIS-CAT_Results		
colorout		

```
# Our new script
a <- 2 + 2
```

```
> # Our new script
> a <- 2 + 2
>
```

```
# Our new script
a <- 2 + 2
```

Figure 2.6: History pane


RStudio
Go to file/function
Addins
Project: (None)
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Untitled1*
Untitled2*
Source on Save
Run
Source
3:1
(Top Level)
R Script
Console
Terminal
~/
Environment
History
Connections
To Console
To Source

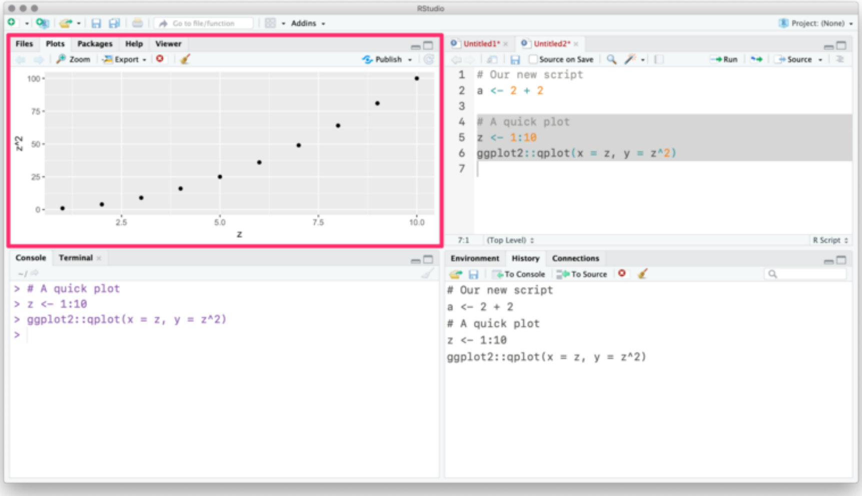
Name	Size	Modified
.gitignore	12 B	Aug 14, 2017, 3:46 PM
.httr-oauth	4.8 KB	Jan 20, 2019, 10:23 PM
.R		
.Rhistory	14.5 KB	Apr 25, 2019, 11:58 AM
.Rprofile	0 B	Mar 2, 2018, 8:24 AM
Applications		
awscli-bundle		
bin		
CIS-CAT_Results		
colorout		

```
# Our new script
a <- 2 + 2
```

```
> # Our new script
> a <- 2 + 2
>
```

```
# Our new script
a <- 2 + 2
```

Figure 2.7: Files pane

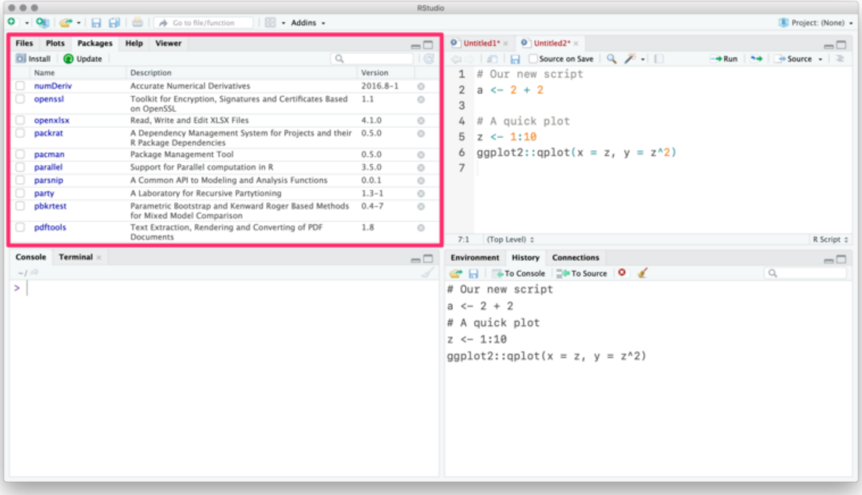


```
# Our new script
a <- 2 + 2

# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

```
> # A quick plot
> z <- 1:10
> ggplot2::qplot(x = z, y = z^2)
>
```

```
# Our new script
a <- 2 + 2
# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

Figure 2.8: Plots pane



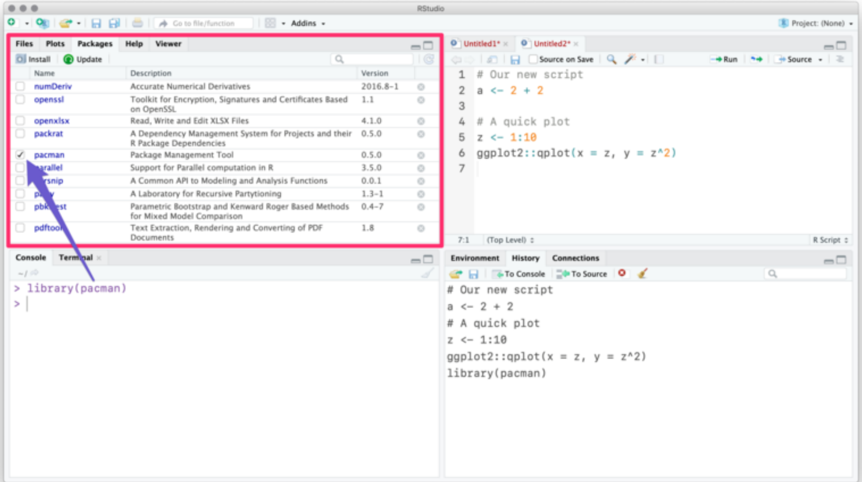
	Name	Description	Version
□	numDeriv	Accurate Numerical Derivatives	2016.8-1
□	openssl	Toolkit for Encryption, Signatures and Certificates Based on OpenSSL	1.1
□	openxlsx	Read, Write and Edit XLSX Files	4.1.0
□	packrat	A Dependency Management System for Projects and their R Package Dependencies	0.5.0
□	pacman	Package Management Tool	0.5.0
□	parallel	Support for Parallel computation in R	3.5.0
□	parsnip	A Common API to Modeling and Analysis Functions	0.0.1
□	party	A Laboratory for Recursive Partytioning	1.3-1
□	pbkrtest	Parametric Bootstrap and Kenward Roger Based Methods for Mixed Model Comparison	0.4-7
□	pdftools	Text Extraction, Rendering and Converting of PDF Documents	1.8

```
# Our new script
a <- 2 + 2

# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

```
>
```

```
# Our new script
a <- 2 + 2
# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

Figure 2.9: Packages pane



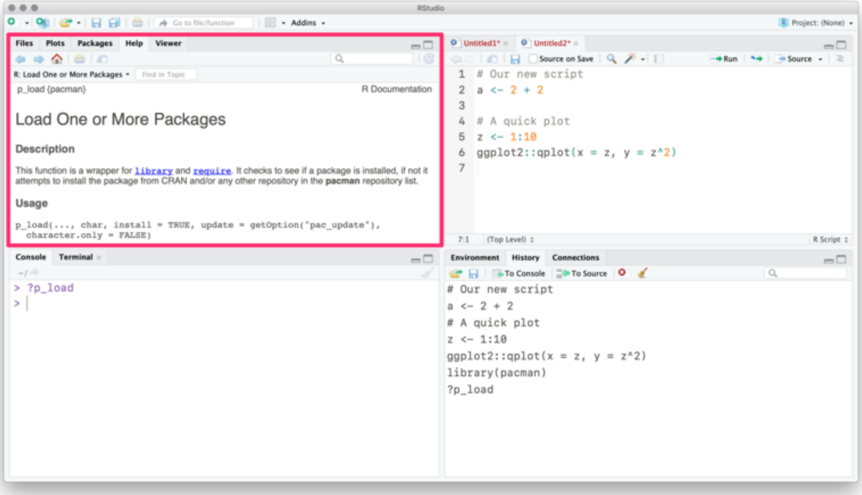
	Name	Description	Version
□	numDeriv	Accurate Numerical Derivatives	2016.8-1
□	openssl	Toolkit for Encryption, Signatures and Certificates Based on OpenSSL	1.1
□	openxlsx	Read, Write and Edit XLSX Files	4.1.0
□	packrat	A Dependency Management System for Projects and their R Package Dependencies	0.5.0
☑	pacman	Package Management Tool	0.5.0
□	parallel	Support for Parallel computation in R	3.5.0
□	parsnip	A Common API to Modeling and Analysis Functions	0.0.1
□	party	A Laboratory for Recursive Partytioning	1.3-1
□	pbkrtest	Parametric Bootstrap and Kenward Roger Based Methods for Mixed Model Comparison	0.4-7
□	pdftools	Text Extraction, Rendering and Converting of PDF Documents	1.8

```
# Our new script
a <- 2 + 2

# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

```
> library(pacman)
> 
```

```
# Our new script
a <- 2 + 2
# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
library(pacman)
```

Figure 2.10: Loaded and installed packages



p_load {pacman} R Documentation

Load One or More Packages

Description

This function is a wrapper for library and require. It checks to see if a package is installed, if not it attempts to install the package from CRAN and/or any other repository in the **pacman** repository list.

Usage

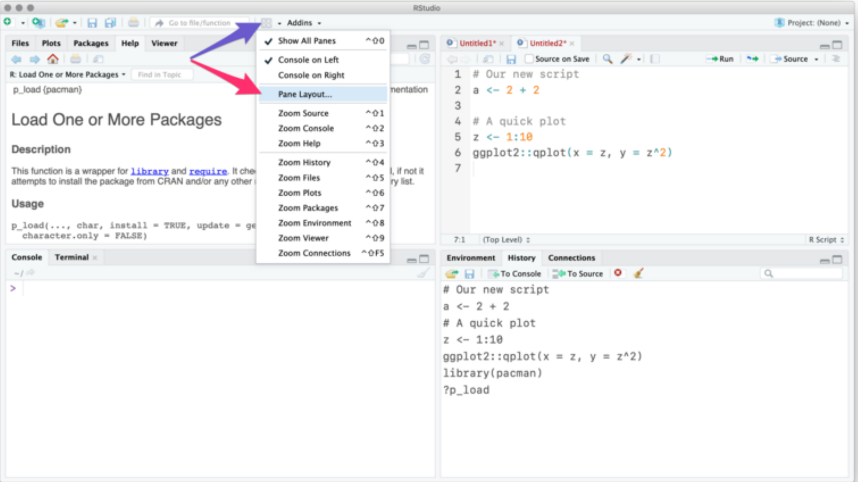
```
p_load(..., char, install = TRUE, update = getOption("pac_update"),
  character.only = FALSE)
```

```
# Our new script
a <- 2 + 2

# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

```
> ?p_load
> 
```

```
# Our new script
a <- 2 + 2
# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
library(pacman)
?p_load
```

Figure 2.11: Help pane



p_load {pacman}

Load One or More Packages

Description

This function is a wrapper for library and require. It che …mentation …, if not it attempts to install the package from CRAN and/or any other … ry list.

Usage

```
p_load(..., char, install = TRUE, update = ge
  character.only = FALSE)
```

```
# Our new script
a <- 2 + 2

# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
```

```
>
```

```
# Our new script
a <- 2 + 2
# A quick plot
z <- 1:10
ggplot2::qplot(x = z, y = z^2)
library(pacman)
?p_load
```

Figure 2.12: Customize layout

2.2 RStudio Addins

RStudio can be further enhanced by so called “addins”. These are clickable snippets that execute certain actions in RStudio.

They aim to make repetitive tasks easier and to save you time. There is an addin called addinslist which lists all available addins. It can be installed as a normal package from CRAN:

```
install.packages("addinslist")
```

To have an addin available in RStudio after installation, RStudio needs to be restarted.

2.3 RStudio projects

Without a project, you will need to define **long** file paths which **only exist on your machine**.

```
sample_df <- read.csv("/Users/<yourname>/somewhere/on/this/machine/sample.csv")
```

With a project, R automatically references the project’s folder as the current working directory.

From there on, you can use *relative paths* to point to files.

```
sample_df <- read.csv("sample.csv")
```

Double-plus bonus: The *here* package extends *RStudio project* philosophy even more and helps in cases when not using RStudio (e.g. on the command line).

2.4 Alternatives to RStudio

- Using R directly in the terminal via radian (optimized R console interpreter)
- R is supported in other “general purpose IDE’s” (VScode, Sublime Text, Atom, Vim, etc.)

Part II

Visualization

Chapter 3

{ggplot2} basics

Embracing the grammar of graphics.

This chapter discusses plotting with the ggplot2 package.

Chapter 4

{ggplot2} advanced

Part III

Tidying and transforming

Chapter 5

Import

Ingesting data.

This chapter discusses data import with RStudio, with the help of the readr, readxl, and rio packages.

Chapter 6

Transformation

Using a consistent grammar of data manipulation.

This chapter discusses data transformation with the dplyr package.

Chapter 7

Tidying

Rows, columns, cells.

This chapter discusses pivoting and data tidying with the help of the tidyr package.

Part IV

Reporting

Part V

Appendix

Chapter 8

Best practices

R code is often organized in packages that can be installed from centralized repositories such as CRAN or GitHub. If you are new to writing R packages, this course cannot give a complete introduction into packages. It is still useful to embrace some very few concepts of R packages to gain access to a vast toolbox and also organize your code in a standardized way familiar to other users. With the first steps in place, the road to your first R package may become less steep.

- Create a `DESCRIPTION` file to declare dependencies and allow easy reloading of the functions you define
- Store your functions in `.R` files in the `R/` directory in your project
  - Scripts that you execute live in `script/` or a similar directory
- Use roxygen2 to document your functions close to the source
- Write tests for your functions, e.g. with testthat

See R packages for a more comprehensive treatment.

8.1 DESCRIPTION

Create and open a new RStudio project. Then, create a `DESCRIPTION` file with `usethis::use_description()`:

```
# install.packages("usethis")
usethis::use_description()
```

Double-check success:

```
# install.packages("devtools")
devtools::load_all()
```

Declare that your project requires the tidyverse and the here package:

```
usethis::use_package("here")
# Currently doesn't work, add manually
# https://github.com/r-lib/usethis/issues/760
# usethis::use_package("tidyverse")
```

8.2 R

With a `DESCRIPTION` file defined, create a new `.R` file and save it in the `R/` directory. (Create this directory if it does not exist.) Create a function in this file, save the file:

```
hi <- function(text = "Hello, world!") {
  print(text)
  invisible(text)
}
```

Do not source the file.

Restart R (with Ctrl + Shift + F10 in RStudio).

Run `devtools::load_all()` again, you can use the shortcut Ctrl + Shift + L or Cmd + Shift + L in RStudio.

Check that you can run `hi()` in the console:

```
hi()
```

```
## [1] "Hello, world!"
```

```
hi("Wow!")
```

```
## [1] "Wow!"
```

Edit the function:

```
hi <- function(text = "Wow!") {
  print(text)
  invisible(text)
}
```

Save the file, but do not source it.

Run `devtools::load_all()` again, you can use the shortcut Ctrl + Shift + L or Cmd + Shift + L in RStudio.

Check that the new implementation of `hi()` is active:

```
hi()
```

```
## [1] "Wow!"
```

All functions that are required for your project are stored in this directory. Do not store executable scripts, use a `script/` directory.

8.3 roxygen2

The following intuitive annotation syntax is a standard way to create documentation for your functions:

```
#' Print a welcome message
#'
#' This function prints "Wow!", or a custom text, on the console.
#'
#' @param text The text to print, "Wow!" by default.
#'
#' @return The `text` argument, invisibly.
#'
#' @examples
#' hi()
#' hi("Hello!")
hi <- function(text = "Wow!") {
  print(text)
  invisible(text)
}
```

This annotation can be rendered to a nicely looking HTML page with the roxygen2 and pkgdown packages. All you need to do is provide (and maintain) it.

8.4 testthat

Automated tests make sure that the functions you write today continue working tomorrow. Create your first test with `usethis::use_test()`:

```
# install.packages("testthat")
usethis::use_test("hi")
```

The file `tests/testthat/test-hi.R` is created, with the following contents:

```
test_that("multiplication works", {
  expect_equal(2 * 2, 4)
})
```

Replace this predefined text with a test that makes more sense for us:

```
test_that("hi() works", {
  expect_output(hi(), "Wow")
  expect_output(hi("Hello"), "Hello")
})
```

Run the new test with `devtools::test()`, you can use the shortcut Ctrl + Shift + T or Cmd + Shift + T in RStudio.

Check that the test actually detects failures by modifying the implementation of `hi()` and rerunning the test:

```
hi <- function(text = "Oops!") {
  print(text)
  invisible(text)
}
```

Run the new test with `devtools::test()`, you can use the shortcut Ctrl + Shift + T or Cmd + Shift + T in RStudio. One test should be failing now.

Chapter 9

- R for data science: https://r4ds.had.co.nz/
- Row oriented workflows: https://github.com/jennybc/row-oriented-workflows#readme
- Advanced R: http://adv-r.had.co.nz/
- Tidy evaluation: https://tidyeval.tidyverse.org/
- R packages: http://r-pkgs.had.co.nz/
- roxygen2: Vignettes in https://cran.r-project.org/package=roxygen2, especially:
  - Introduction to roxygen2
  - Generating Rd files for an overview of available tags
  - Write R documentation in Markdown
- How R searches and finds stuff: http://blog.obeautifulcode.com/R/How-R-Searches-And-Finds-Stuff/
- What they forgot to teach you: https://whattheyforgot.org/
- Parallel processing with a purrr-like interface: https://davisvaughan.github.io/furrr/
- Tidyverse principles: https://principles.tidyverse.org/
- Recursive lists to use in teaching and examples: https://github.com/jennybc/repurrrsive