Computer Support

R basics

Dr.S., doc. Oleksii Yehorchenkov

Part I. R Basics

What is R?

R is a dialect of the S language.

R Philosophy

In “Stages in the Evolution of S”, John Chambers writes:

“[W]e wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming, when the language and system aspects would become more important.”

What is R?

- R is a free language and environment for statistical computing and graphics
- R is an interpreted language, not a compiled one, meaning that all commands typed on the keyboard are directly executed without requiring to build a complete program (this is like Python and unlike C, Fortran, Pascal, etc.)
- R has existed for over 25 years
- R is modular — most functionality is from add-on packages. So the language can be thought of as a *platform* for creating and running a large number of useful packages.

Free Software

With *free software*, you are granted

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

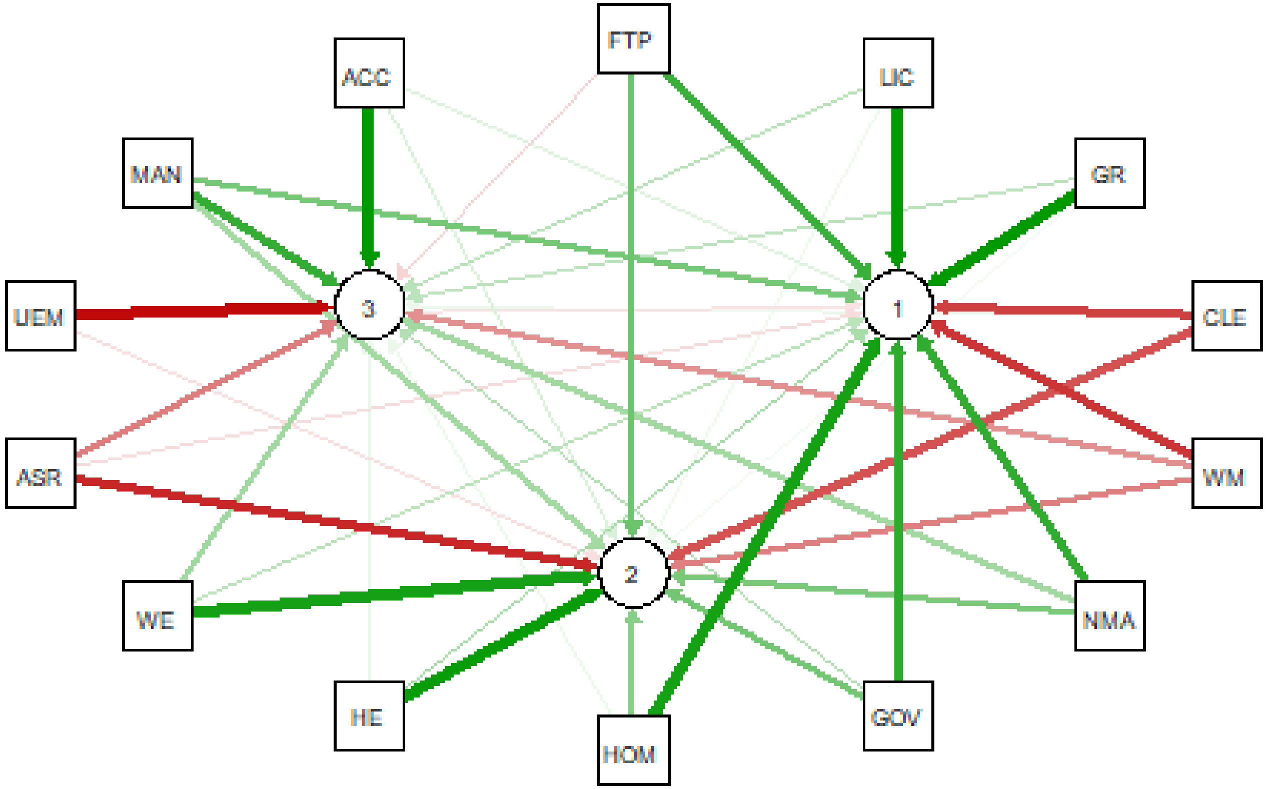
http://www.fsf.org

Why use R?

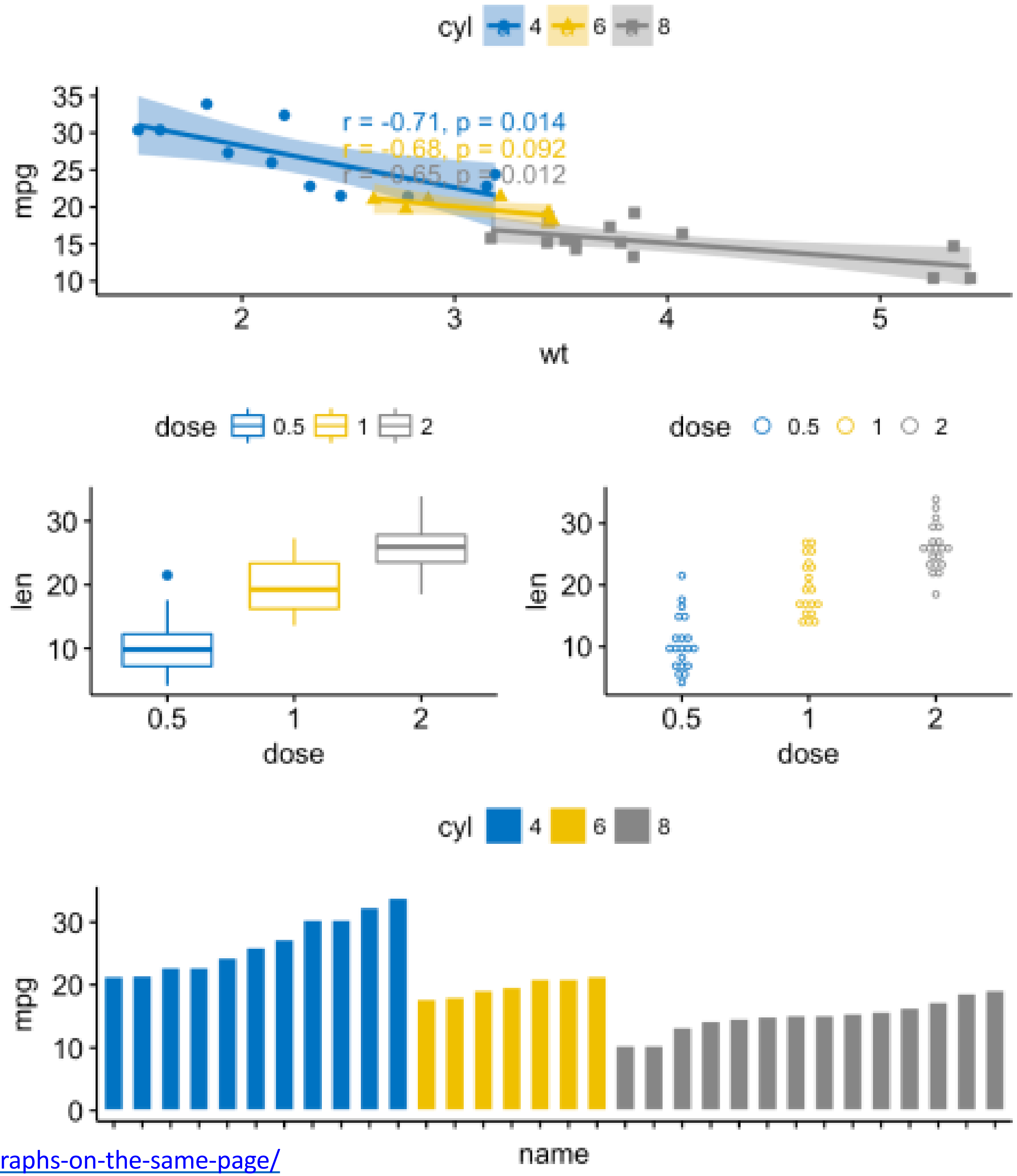
- The most popular software for data analysis
- Extremely flexible: can be used to manipulate, analyze, and visualize any kind of data
- Cutting edge statistical tools
- Publication quality graphics
- 15,000+ add on packages covering all aspects of statistics and machine learning
- Active community of users

For what can we use R?

For complex data analysis

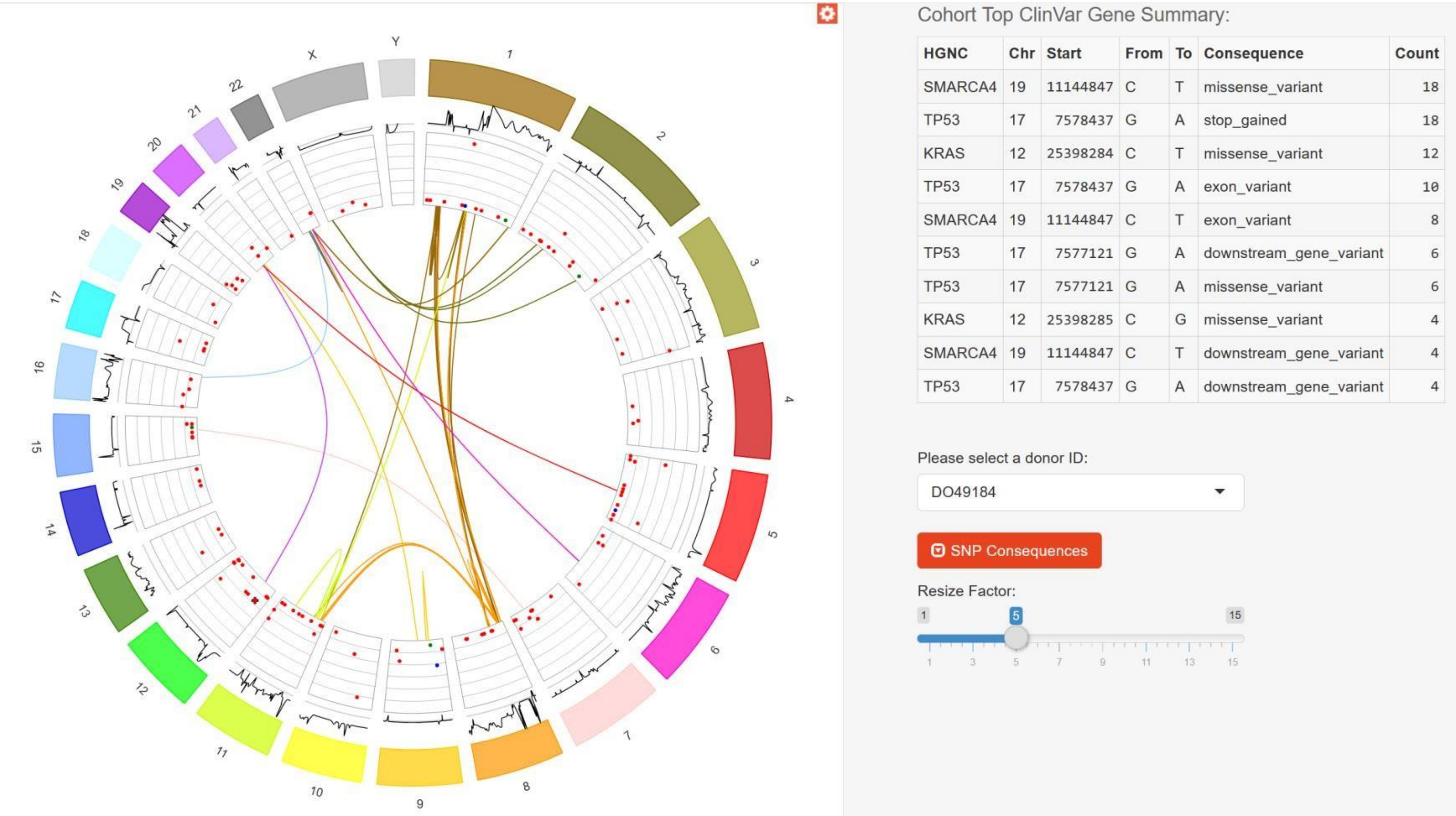


For high quality visualization



https://ggplot2.tidyverse.org/
https://www.r-bloggers.com/2017/07/ggplot2-easy-way-to-mix-multiple-graphs-on-the-same-page/

For interactive web applications



Cohort Top ClinVar Gene Summary:

HGNC	Chr	Start	From	To	Consequence	Count
SMARCA4	19	11144847	C	T	missense_variant	18
TP53	17	7578437	G	A	stop_gained	18
KRAS	12	25398284	C	T	missense_variant	12
TP53	17	7578437	G	A	exon_variant	10
SMARCA4	19	11144847	C	T	exon_variant	8
TP53	17	7577121	G	A	downstream_gene_variant	6
TP53	17	7577121	G	A	missense_variant	6
KRAS	12	25398285	C	G	missense_variant	4
SMARCA4	19	11144847	C	T	downstream_gene_variant	4
TP53	17	7578437	G	A	downstream_gene_variant	4

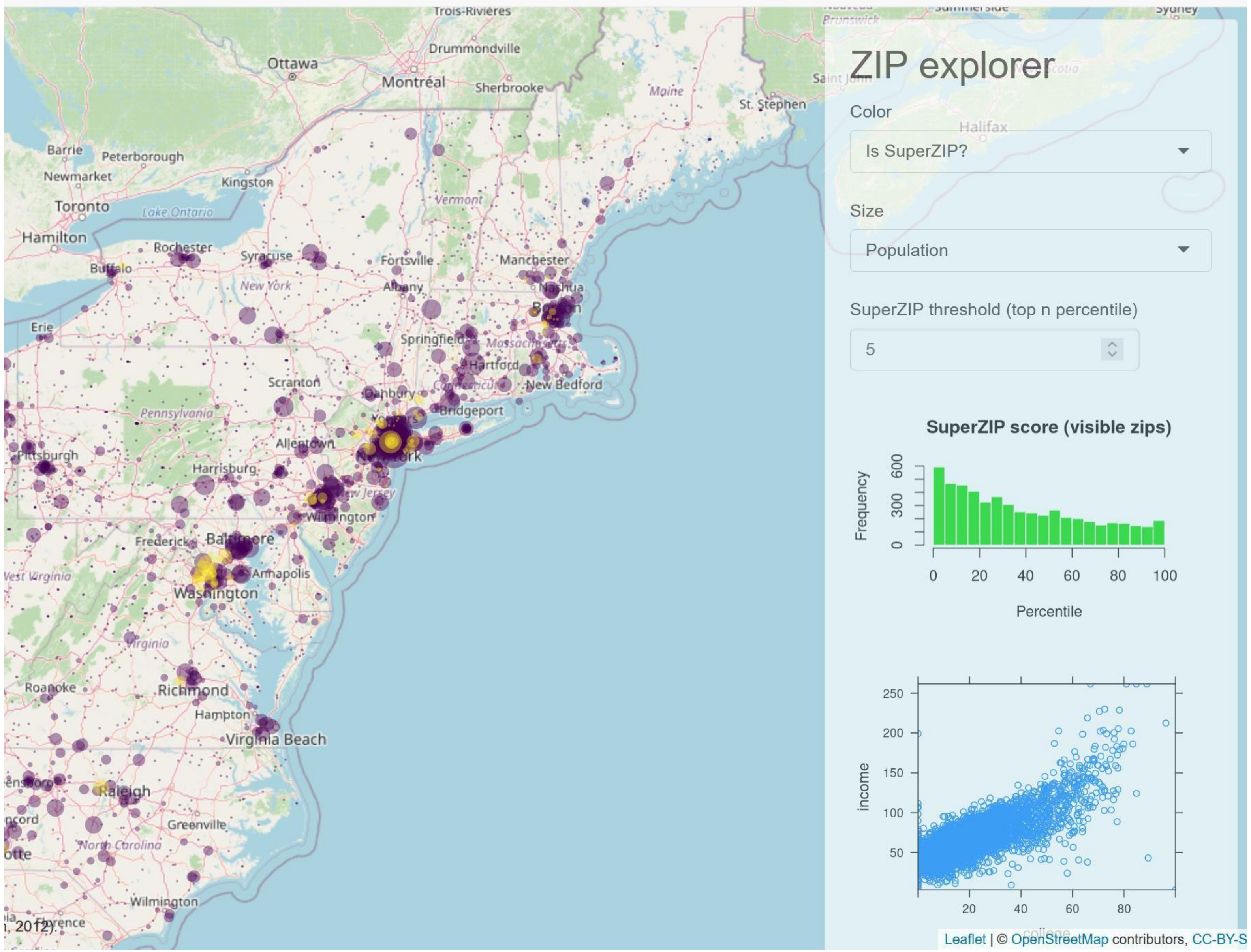
Please select a donor ID:

DO49184

Resize Factor: 5

https://shiny.rstudio.com/gallery/genome-browser.html/

For working with spatial data



ZIP explorer

Color

Is SuperZIP?

Size

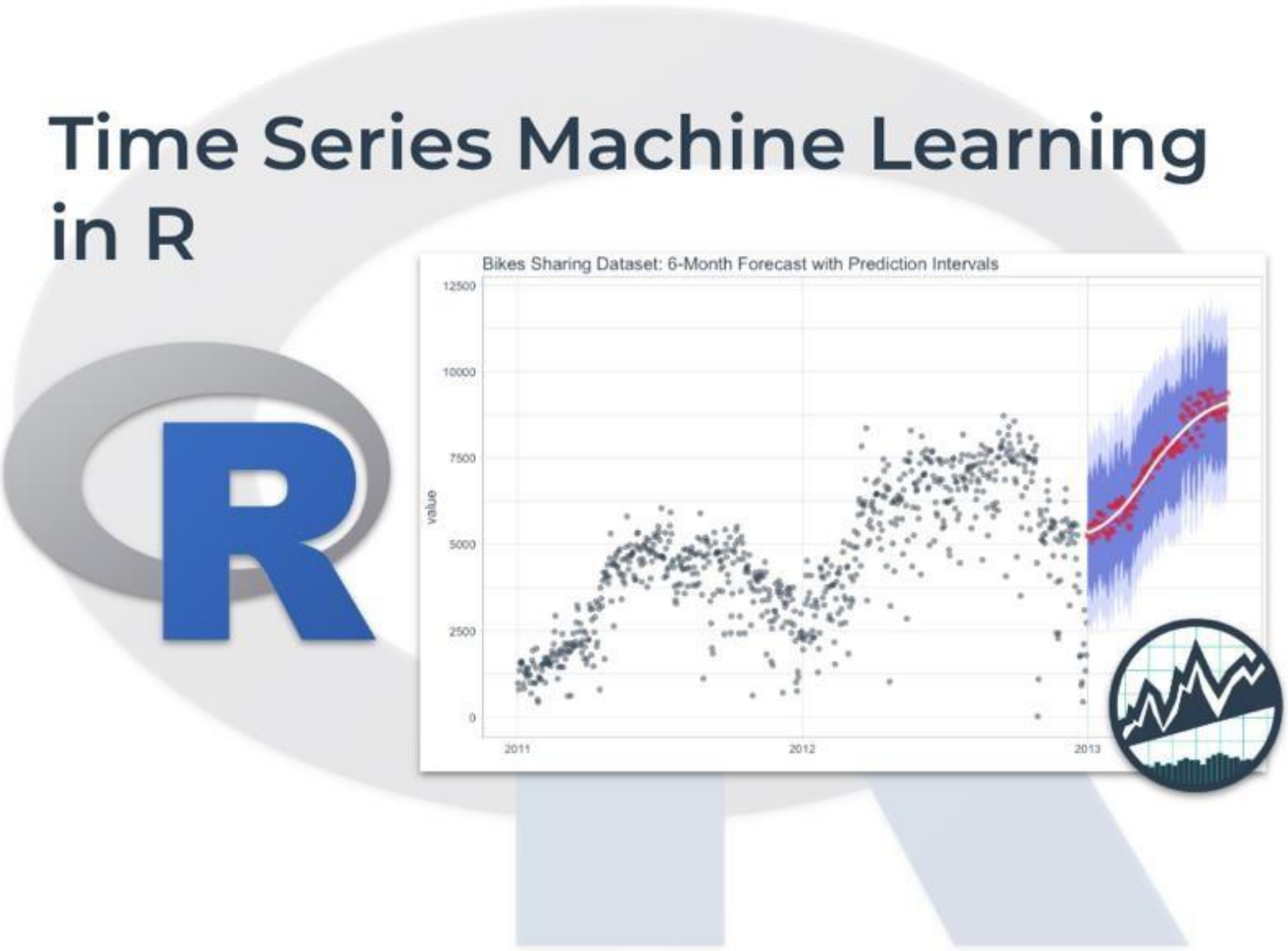
Population

SuperZIP threshold (top n percentile)

5

https://shiny.rstudio.com/gallery/superzip-example.html

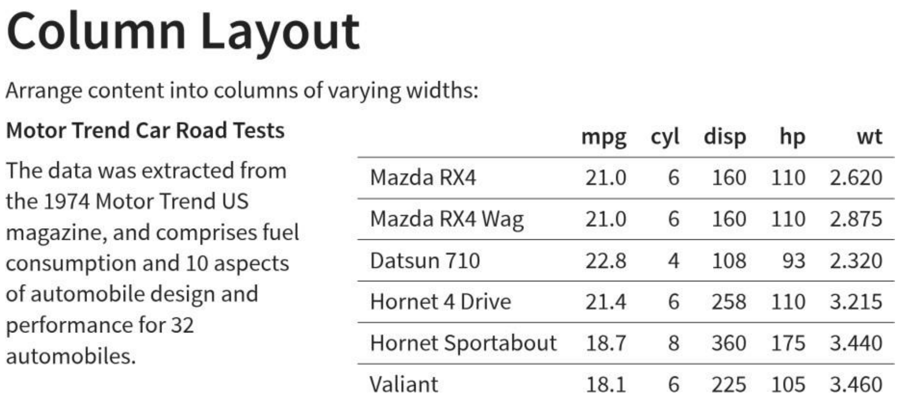
For creating different models



Time Series Machine Learning in R

https://www.business-science.io/time-series/2020/03/18/time-series-machine-learning.htmlery/superzip-example.html

For creating books and presentations



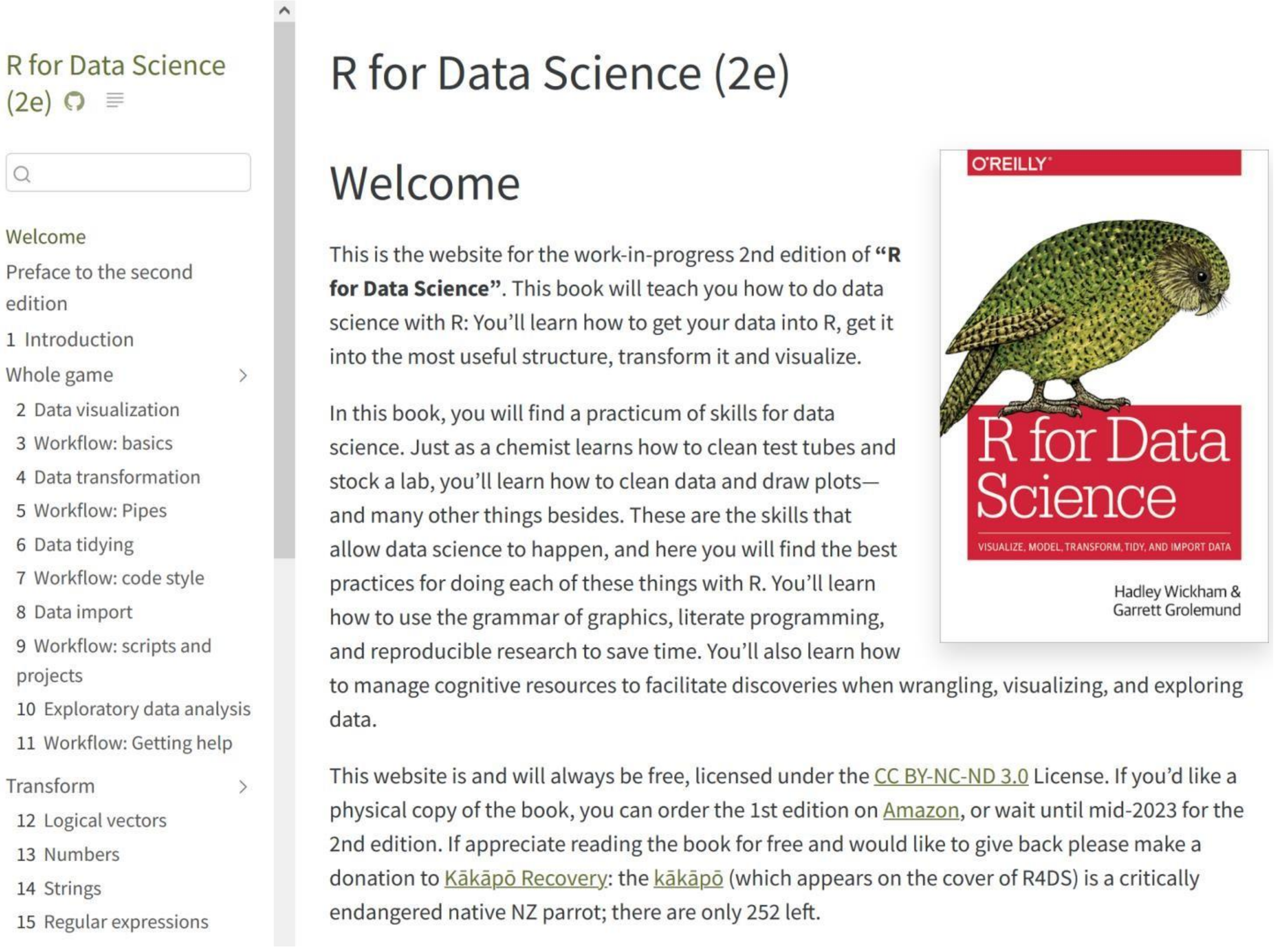
Column Layout

Arrange content into columns of varying widths:

Motor Trend Car Road Tests

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles.

	mpg	cyl	disp	hp	wt
Mazda RX4	21.0	6	160	110	2.620
Mazda RX4 Wag	21.0	6	160	110	2.875
Datsun 710	22.8	4	108	93	2.320
Hornet 4 Drive	21.4	6	258	110	3.215
Hornet Sportabout	18.7	8	360	175	3.440
Valiant	18.1	6	225	105	3.460



R for Data Science (2e)

R for Data Science (2e)

Welcome

O'REILLY
R for Data Science
VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA
Hadley Wickham & Garrett Grolemund

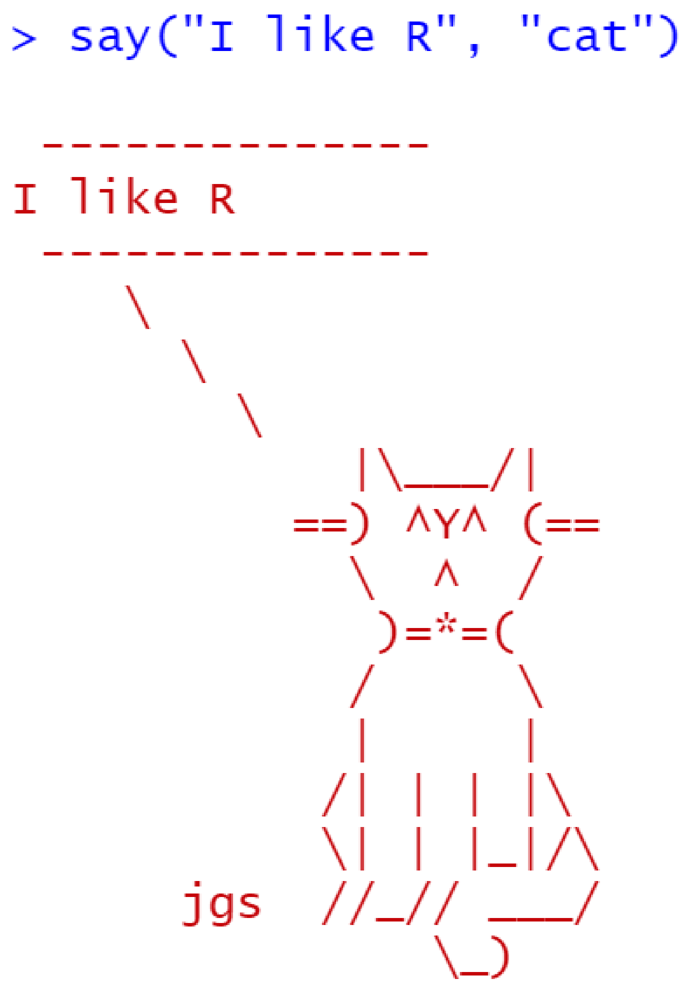
This is the website for the work-in-progress 2nd edition of **"R for Data Science"**. This book will teach you how to do data science with R: You'll learn how to get your data into R, get it into the most useful structure, transform it and visualize.

In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you'll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You'll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You'll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualizing, and exploring data.

This website is and will always be free, licensed under the CC BY-NC-ND 3.0 License. If you'd like a physical copy of the book, you can order the 1st edition on Amazon, or wait until mid-2023 for the 2nd edition. If appreciate reading the book for free and would like to give back please make a donation to Kākāpō Recovery: the kākāpō (which appears on the cover of R4DS) is a critically endangered native NZ parrot; there are only 252 left.

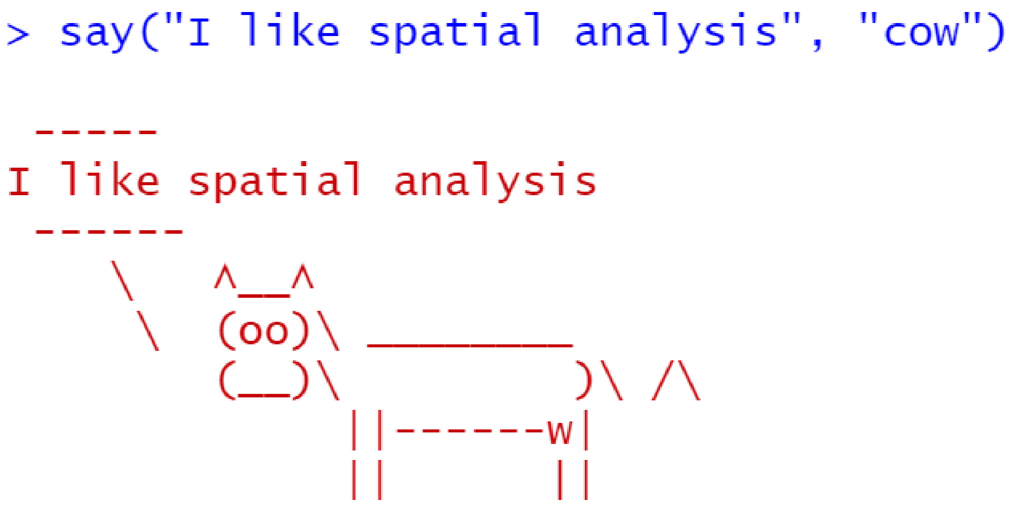
https://r4ds.hadley.nz/
https://quarto.org/

For drawing a cow and a cat



```
> say("I like R", "cat")

 --------------
I like R
 --------------
    \
      \
        \
            |\___/|
          ==) ^Y^ (==
            \  ^  /
             )=*=(
            /     \
            |     |
           /|  |  |  |\
           \|  |  |_|/\
      jgs  //_// ___/
               \_)
```

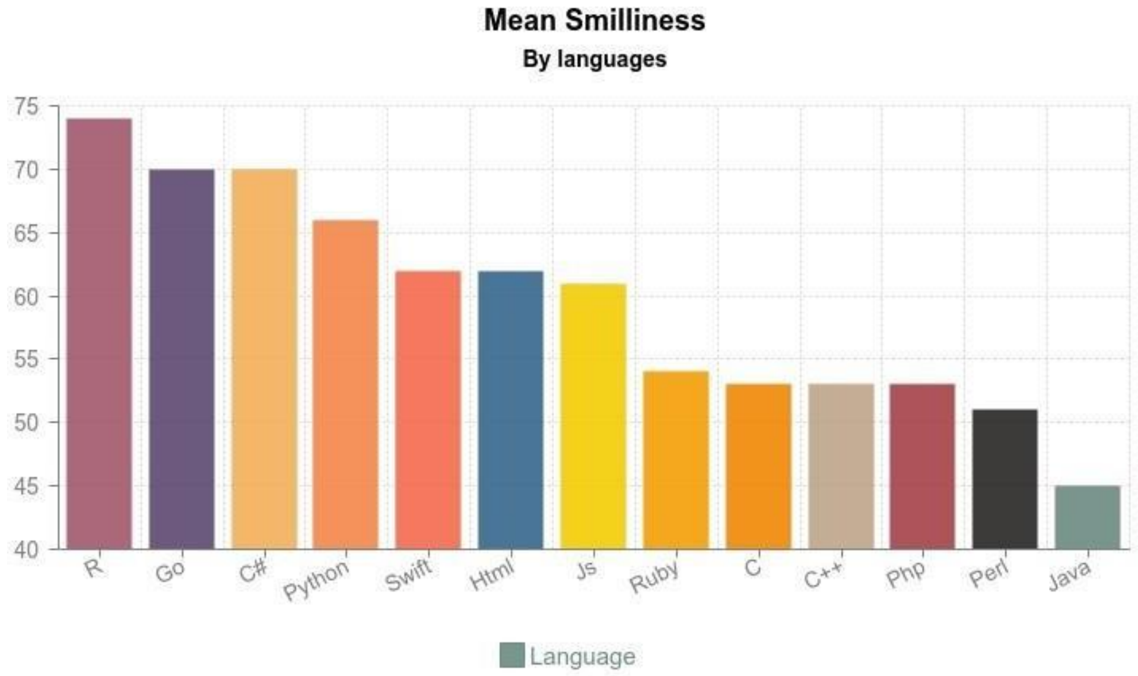


```
> say("I like spatial analysis", "cow")

 -----
I like spatial analysis
 ------
    \   ^__^
     \  (oo)\ ________
        (__)\         )\ /\
             ||------w|
             ||      ||
```

https://cran.r-project.org/web/packages/cowsay/vignettes/cowsay_tutorial.html

Every programmer has a language he doesn’t like, so much so he can’t even smile for his profile photo. Face API measures the amount that you are smiling, using a value between 0 and 1



https://medium.com/swlh/what-programming-language-has-the-happiest-developers-f0636b08e898

Terminology

Interpreter: A program that reads another program and executes it

Program: A set of instructions that specifies a computation.

Operator: A special symbol that represents a simple computation like addition, multiplication, or string concatenation.

Value: One of the basic units of data, like a number or string, that a program manipulates.

Syntax: The rules that govern the structure of a program.

Terminology

Variable: A name that refers to a value.

Assignment: A statement that assigns a value to a variable.

Statement: A section of code that represents a command or action.

Function: A named sequence of statements that performs some useful operation. Functions may or may not take arguments and may or may not produce a result.

Allen Downey. Think Python

Syntax rules

- R is case sensitive
- R ignores white space
- Variable names should start with a letter (A-Z and a-z) and can include letters, digits (0-9), dots (.), and underscores (_)
- Comments can be inserted using a hash # symbol
- Functions must be written with parentheses, even if there is nothing within them; for example: ls()

Objects

R has five basic or “atomic” classes of objects:

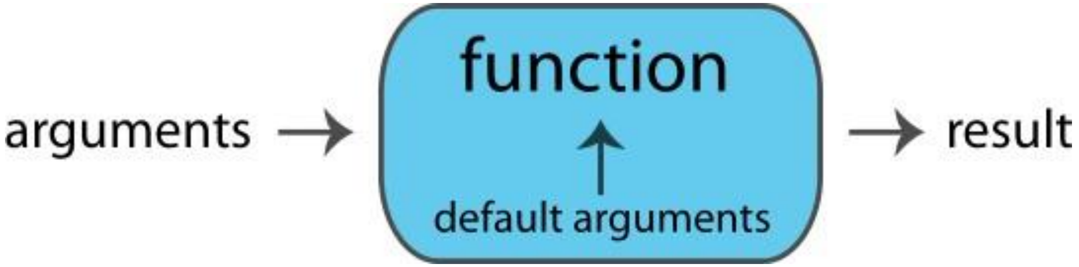
- character
- numeric (real numbers)
- integer
- complex
- logical (True/False)

The most basic object is a vector

- A **vector** can only contain objects of the same class
- BUT: The one exception is a **list**, which is represented as a vector but can contain objects of different classes (indeed, that’s usually why we use them)

Function calls

Functions perform actions — they take some input, called arguments and return some output (i.e., a result). Here's a schematic of how a function works



```
round(x = 2.34, digits = 1) # match by name
## [1] 2.3
```

```
round(2.34, 1) # match by position
## [1] 2.3
```

Assignment

Objects (data structures) can be assigned names and used in subsequent operations:

- The **gets** <- operator (less than followed by a dash) is used to save objects
- The name on the left **gets** the object on the right

```
sqrt(10) # calculate square root of 10; result is not stored anywhere

## [1] 3.162278

x <- sqrt(10) # assign result to a variable named x
```

Asking for help

1. You can ask R for help using the **help** function, or the **?** Shortcut

```
help(help)

?help

?sqrt
```

The help function can be used to look up the documentation for a function, or to look up the documentation to a package. We can learn how to use the stats package by reading its documentation like this:

```
help(package = "stats")
```

2. If you know the name of the package you want to use, then Googling “R package-name” will often get you to the documentation.

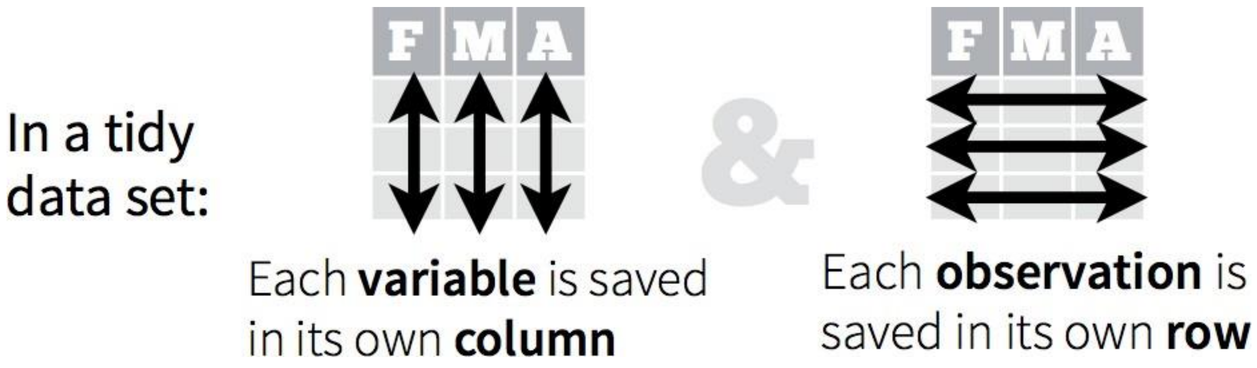
Installing & using packages

R is a modular environment that is extended by the use of packages. Packages are collections of functions or commands that are designed to perform specific tasks (e.g., fit a type of regression model). A large number of contributed packages are available (> 15,000).

Using an R package is a two step process:

- Install the package onto your computer using the **install.packages()** function. This only needs to be done the first time you use the package.
- Load the package into your R session's search path using the **library()** function. This needs to be done each time you use the package.

The tidyverse

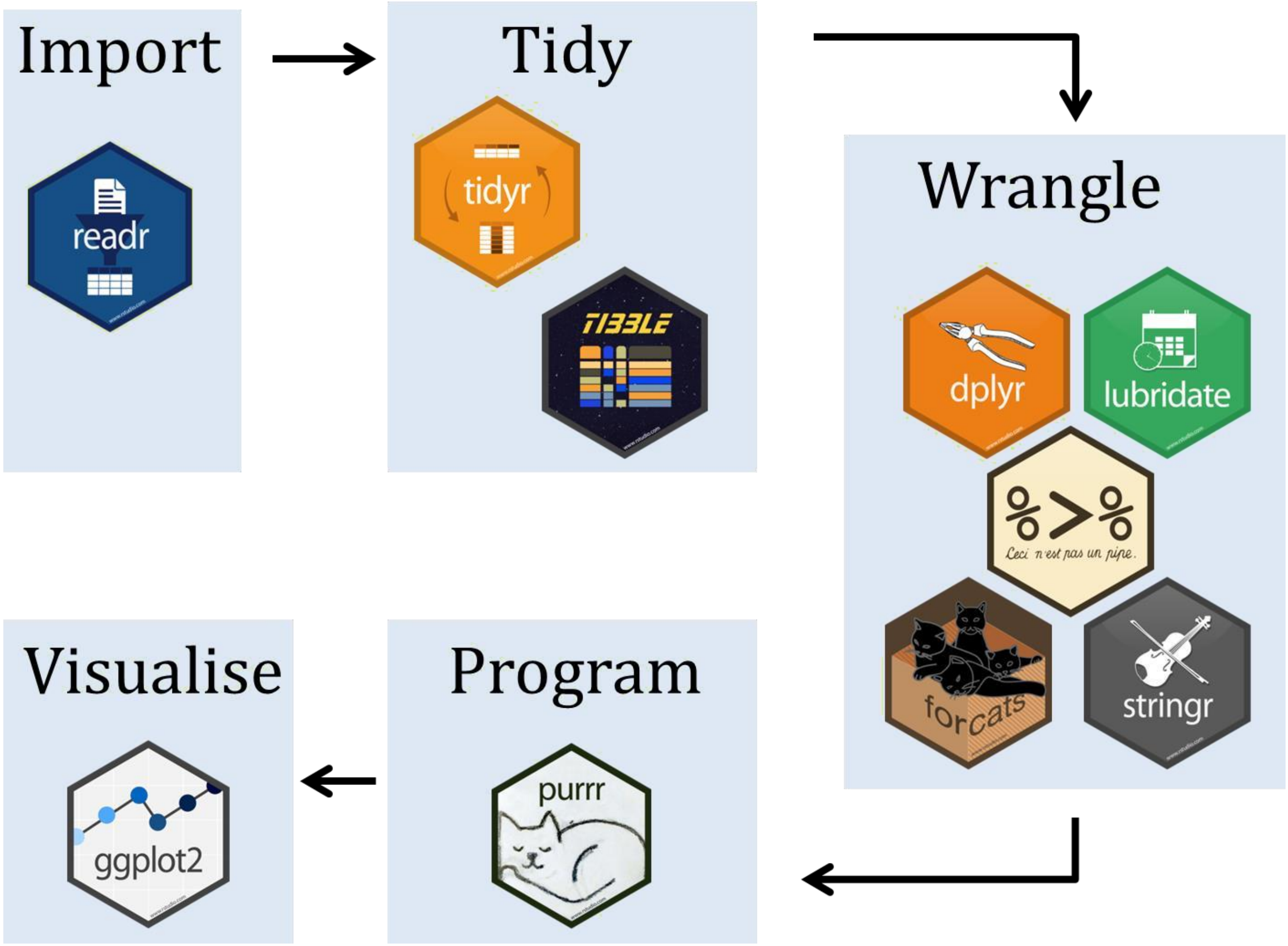


In a tidy data set:

Each **variable** is saved in its own **column**

Each **observation** is saved in its own **row**

The tidyverse



Import

Tidy

Wrangle

Program

Visualise

Data Frames

- Data frames are used to store tabular data
- They are represented as a special type of list where every element of the list has to have the same length
- Each element of the list can be thought of as a column and the length of each element of the list is the number of rows
- Unlike matrices, data frames can store different classes of objects in each column (just like lists); matrices must have every element be the same class
- Data frames also have a special attribute called row.names
- Data frames are usually created by calling read.table () or read.csv()
- Can be converted to a matrix by calling data.matrix

How to read data from a file

To read data from a file, you have to know what kind of file it is. The table below lists functions from the readr package, which is part of tidyverse, that can import data from common plain-text formats.

- **read_csv()** reads comma delimited files, **read_csv2()** reads semicolon separated files (common in countries where , is used as the decimal place), **read_tsv()** reads tab delimited files, and **read_delim()** reads in files with any delimiter.
- **read_fwf()** reads fixed width files. You can specify fields either by their widths with **fwf_widths()** or their position with **fwf_positions()**. **read_table()** reads a common variation of fixed width files where columns are separated by white space.

How to read data from a file

```
# read data to dataframe “df”

df <- read_csv("dataSets/states.csv")

# show the six first rows

head(df)
# A tibble: 6 x 21
  state region    pop   area density metro waste energy miles toxic
  <chr> <chr>   <dbl>  <dbl>   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
1 Alab~ South  4.04e6  52423  77.1    67.4 1.11     393 10500 27.9
2 Alas~ West   5.50e5 570374   0.960  41.1 0.910    991  7200 37.4
3 Ariz~ West   3.66e6 113642  32.2    79   0.790    258 9700 19.6
4 Arka~ South  2.35e6  52075  45.2    40.1 0.850    330 8900 24.6
5 Cali~ West   2.98e7 155973 191.     95.7 1.51     246 8700  3.26
6 Colo~ West   3.29e6 103730 31.8     81.5 0.730    273  8300  2.25
# ... with 11 more variables: green <dbl>, house <dbl>, senate <dbl>,
#   csat <dbl>, vsat <dbl>, msat <dbl>, percent <dbl>, expense <dbl>,
# income <dbl>, high <dbl>, college <dbl>
```