Introduction to R

Sep. 9, 2021

Today: Learning Outcomes

After completing this lab, you should be able to:

- Explain what R is and describe its main uses.
- Give (at least) 3 benefits of using R for data analysis.
- Create R objects.
- Identify R objects of the following R classes: numeric, character, logical, data.frame.
- Apply basic operations to vectors in R.
- Explain the basic structure of function calls.
- Write scripts to manually enter data as `data.frames` in R.

What is R?

What is R?[1]

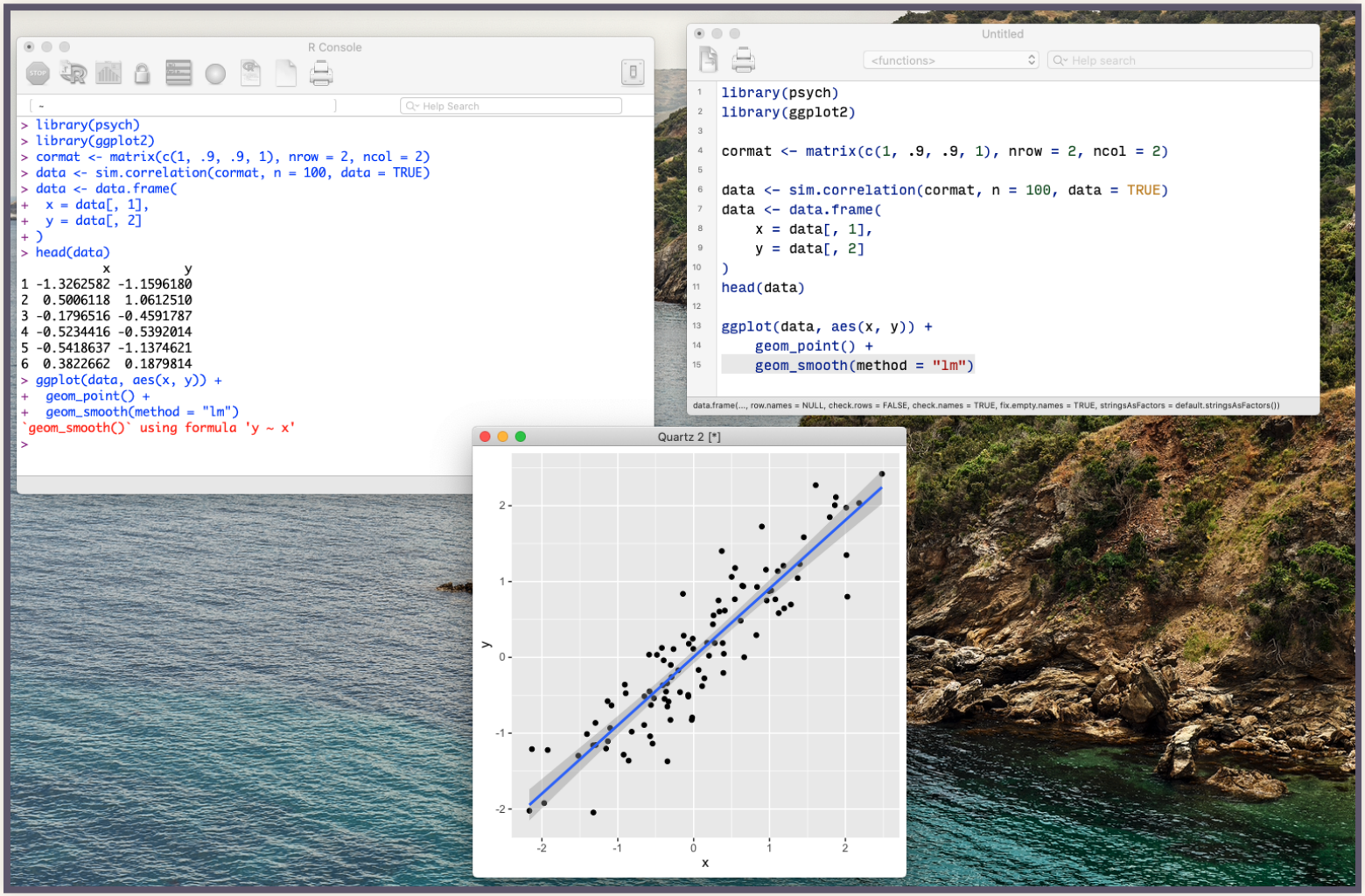
> *R is a language and environment for statistical computing and graphics.*

R is a Language

...but not that kind of language.

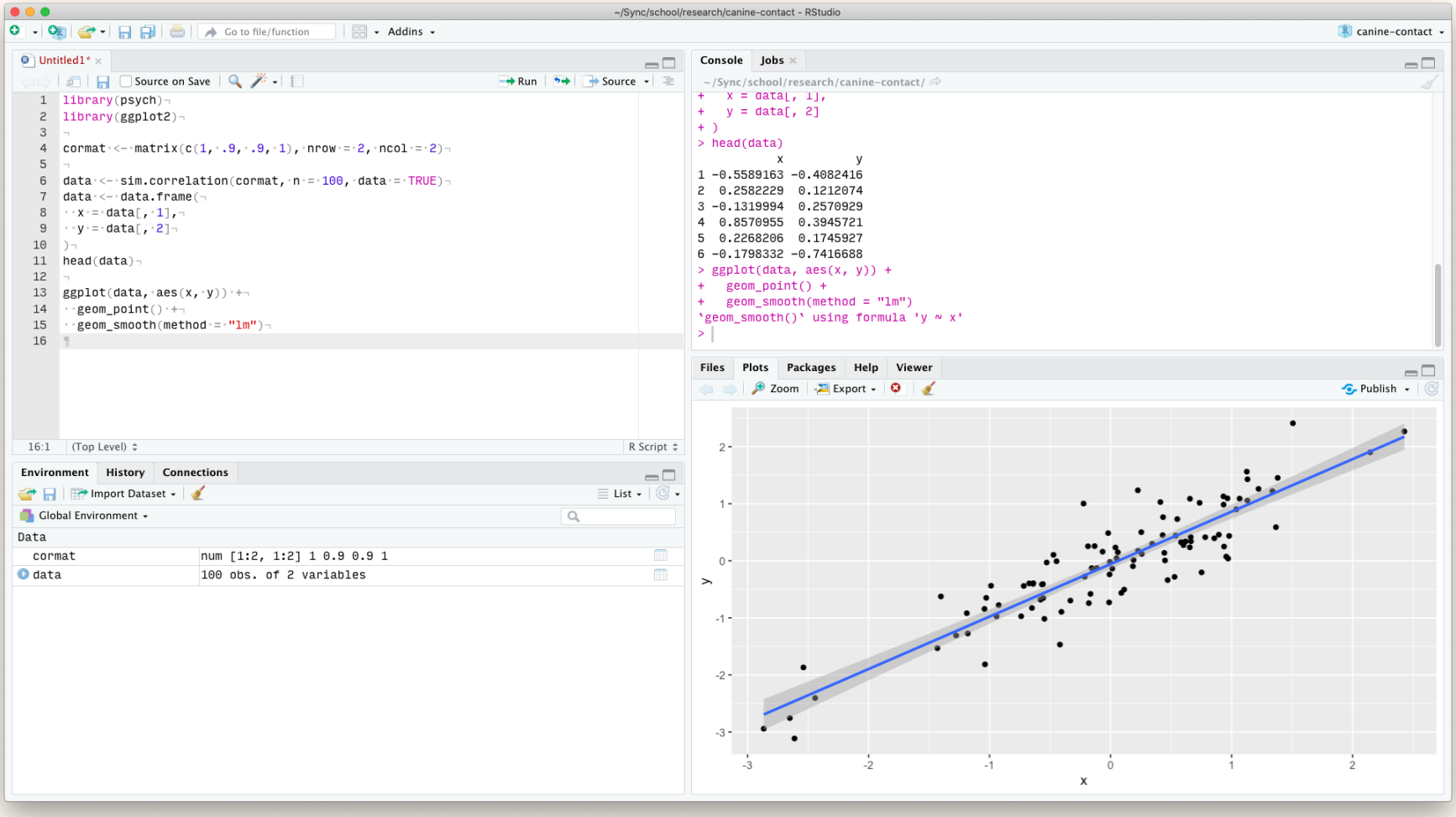
R is an Environment

R Integrated Developer Environment (IDE)



R is an Environment

RStudio IDE



What Can R Do?

Base R

- Data handling
- Mathematical operations
- Data visualization
- Programming

Data Analysis

R is first and foremost, a tool for data analysis:

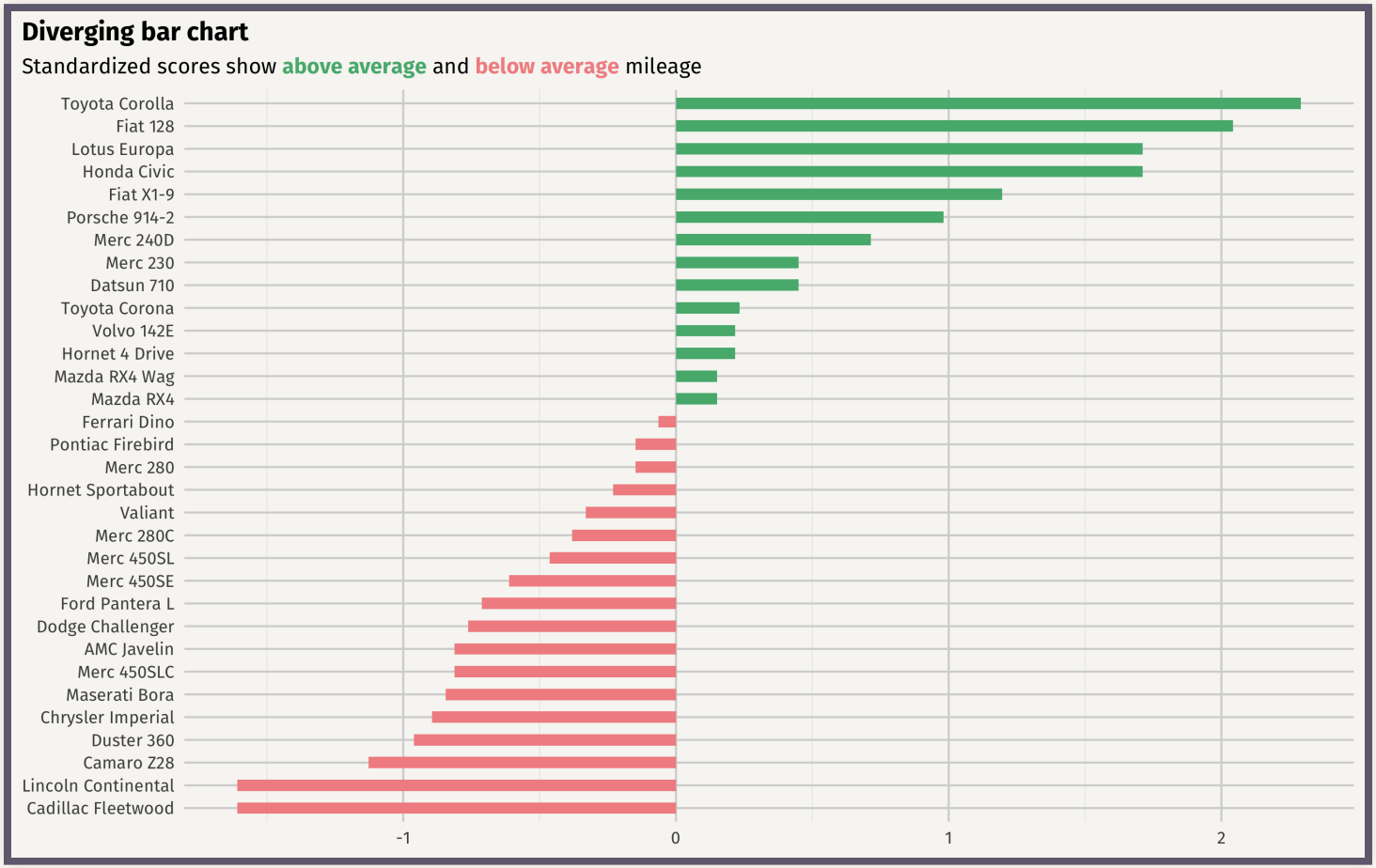
- **General linear models** (e.g., *t* tests, ANOVA, multi-level modelling).
- Genearlized linear models (survival analysis, binomial regression).
- Factor analysis.
- Structural equation modelling.
- Machine learning.
- And much more!

Data Visualization

In addition to running statistics, R can be used to visualize your data.

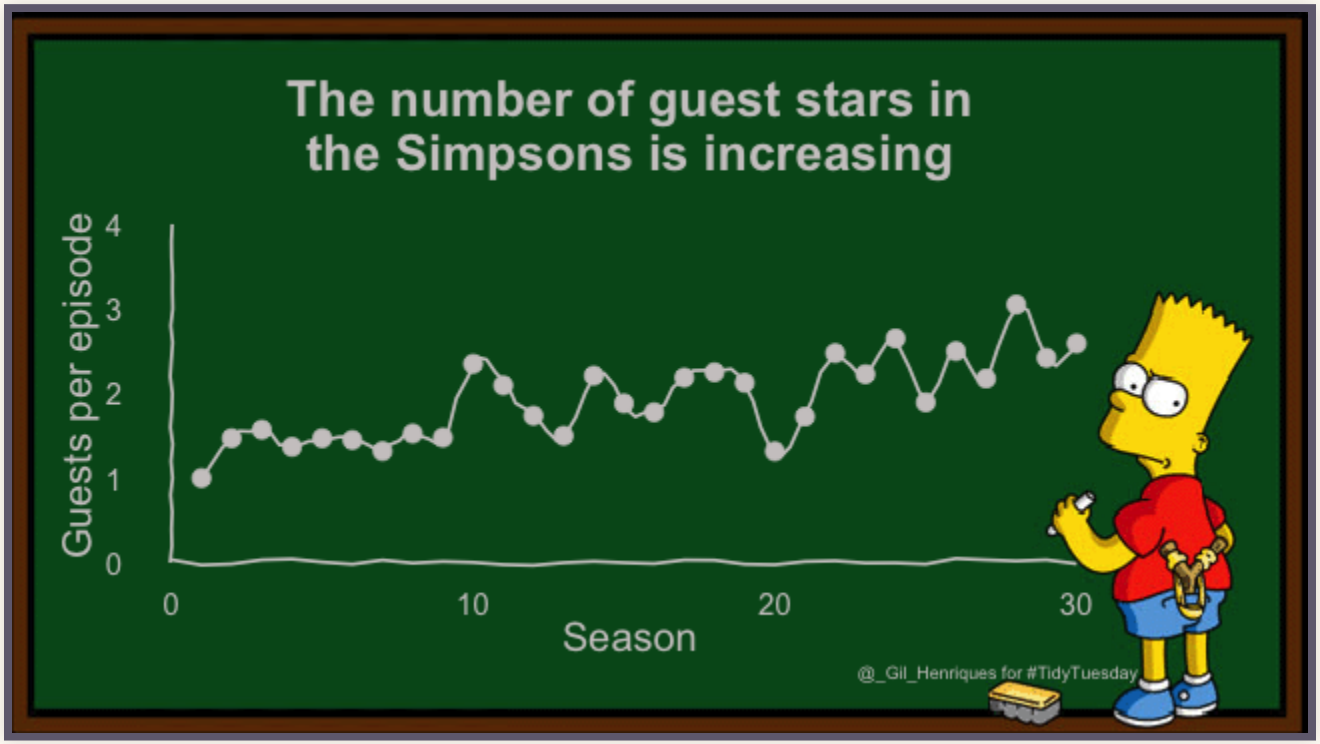
In fact, R offers state of the art data visualization.

State of the Art Data Visualization



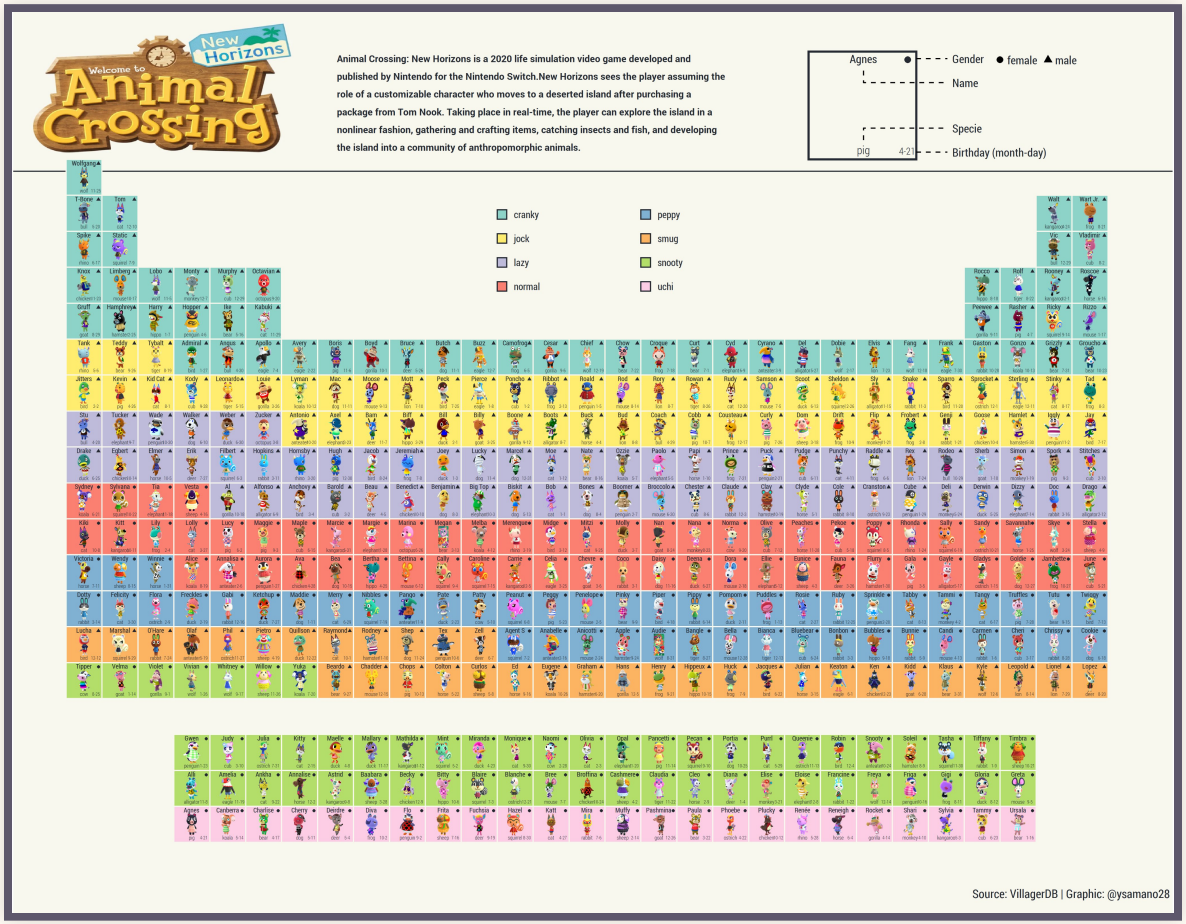
See Top 50 ggplot Visualizations (With Full R Code) for the code for this and other plots.

State of the Art Data Visualization



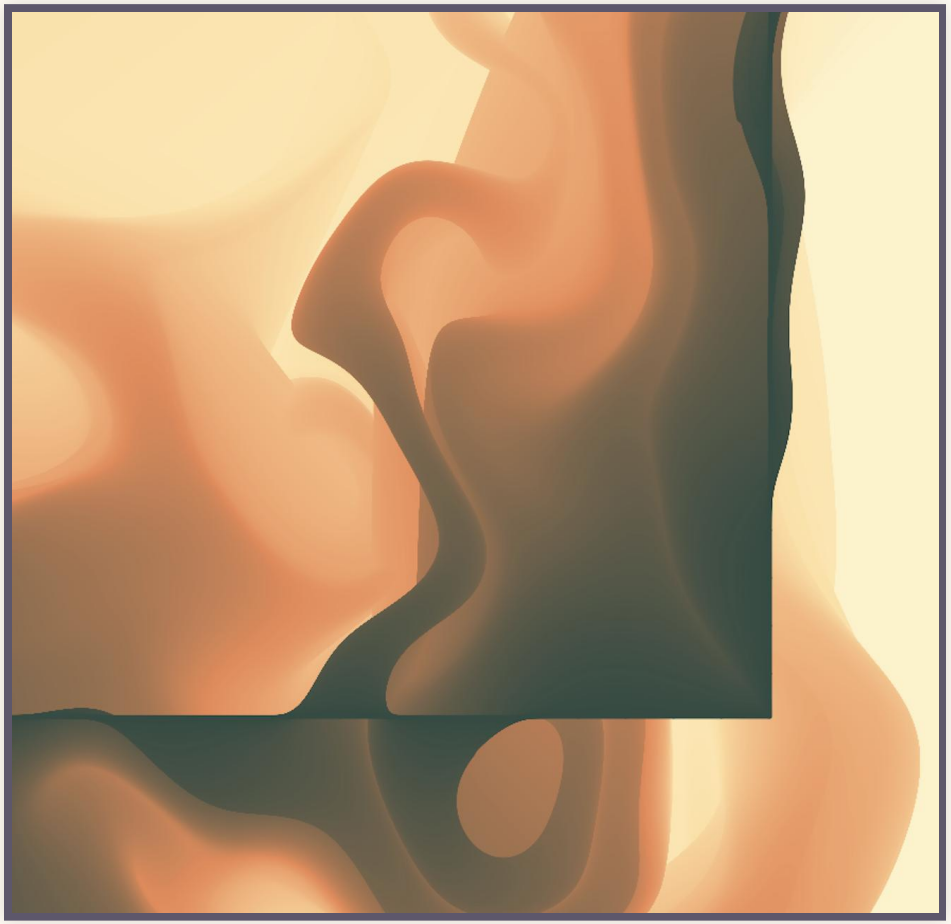
Want to make this figure? There's a tutorial for that!

State of the Art Data Visualization[3]



From @ysamano28

State of the Art Data Visualization



From @thomasp85.

R Programming

R is a programming language. It can do basically anything that can be done with a computer.

R Packages

- 18142 packages on the CRAN as of September 10, 2021.
- More packages available on GitHub and other repositories.
- If there is no package that does what you want, you can build your own!

RMarkdown

RStudio also has RMarkdown, which can be used to create documents that combine text, R code, and R output.

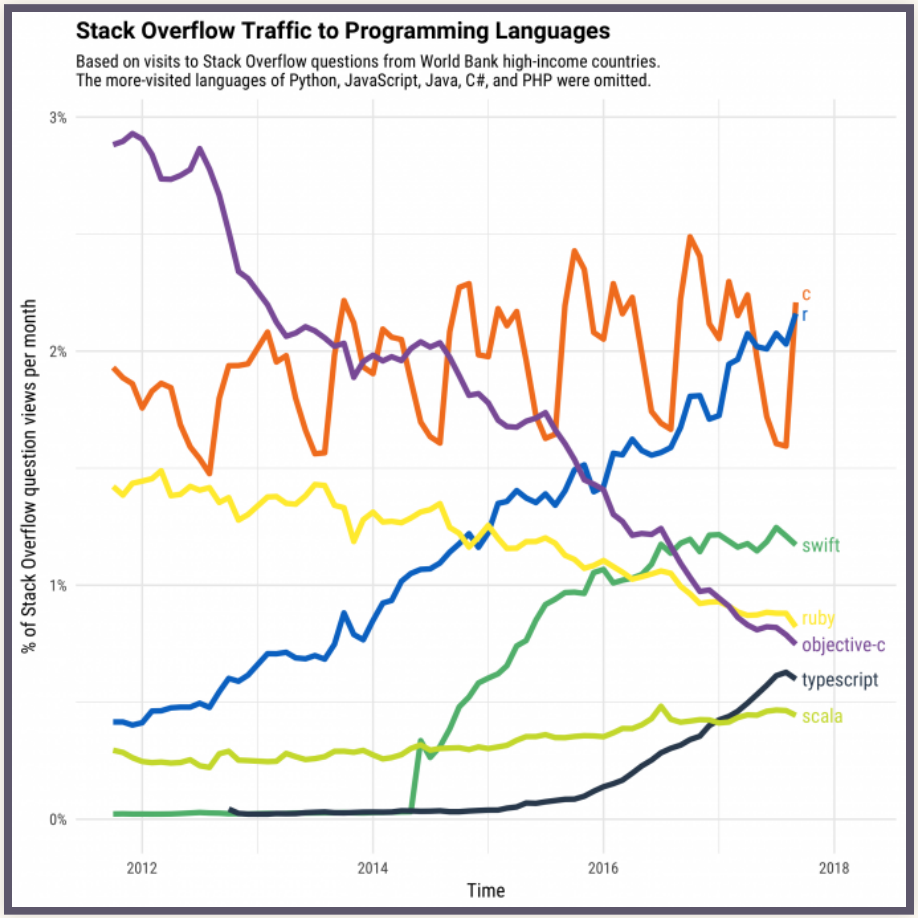
For example, these slides were created in RStudio.

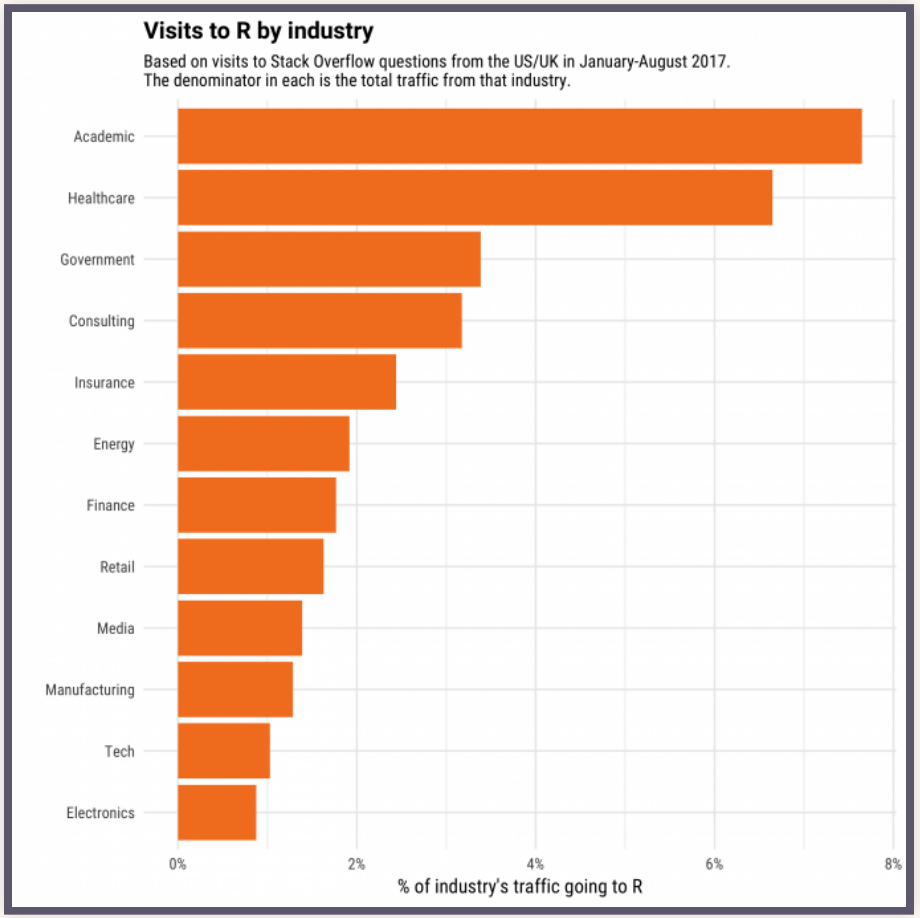
Why Should I Use R?

R is Popular in Industry

The above companies are all customers of RStudio.

R is Popular in Academia[4]





Should I Use R?

Pros

- Open source.
- Flexible.
- Reproducible.
- Saves time in the long run.
- State of the art data visualization.
- Supportive community.

Cons

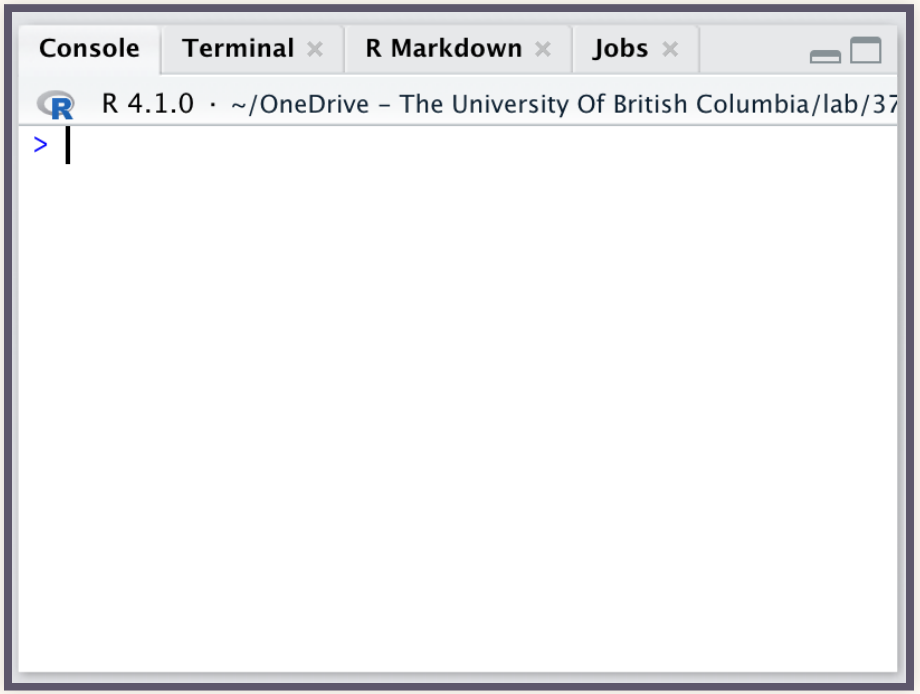
- Steep learning curve.
- Inconsistent documentation.
- Limited quality control.
- Limited support from supervisors

Should I Use R?

Well, yes!

Lab Activity

Console Pane



- R commands are executed in the source pane.
- Type a command next the > and press “return” to execute it.
- For example, execute `1 + 1` in the console.

Mathematical Operations

You can use R like a calculator.

Some R Operators

- Multiplication: *
- Division: /
- Addition: +
- Subtraction: -
- Parentheses: ( )

tRy it! Mathematical Operations

Use the R console to solve the following equations:

1. $1/20$
2. $\frac{1}{20 \times 20}$
3. $1 - .95^{10}$
4. $\sqrt{19}$

Mathematical Operations: Solutions

Problem 1

1/20

```
1 / 20
```

```
## [1] 0.05
```

Mathematical Operations: Solutions (2)

Problem 2

$$\frac{1}{20 \times 20}$$

```
1 / (20 * 20)
```

```
## [1] 0.0025
```

Mathematical Operations: Solutions (3)

Problem 3

$$1 - .95^{10}$$

```
1 - .95^10
```

```
## [1] 0.4012631
```

```
1 - .95**10
```

```
## [1] 0.4012631
```

Mathematical Operations: Solutions (4)

Problem 4

$$\sqrt{19}$$

```
19^(1/2)
```

```
## [1] 4.358899
```

```
sqrt(19)
```

```
## [1] 4.358899
```

Object Assignment

Object Assignment

- Create objects `<-` (the assignment operator)
- Shortcut: Use Alt + - (the minus sign) in RStudio

```
x <- 3 * 4

object_name <- value
```

Think “object name gets value”

What's Inside the Box?

```
x <- 5

# print() is the default function of an R object
x
```

```
## [1] 5
```

Working with Objects

You can treat that object as whatever you assigned it to be.

```
x * 5
```

```
## [1] 25
```

What is the value of x?

```
x
```

```
## [1] 5
```

Give Objects New Values

```
x <- 5

x <- x * 5
```

Now, what is the value of x?

```
x
```

```
## [1] 25
```

Object Values

```
x <- 4 + 4

x
```

```
## [1] 8
```

Vectors

Object Classes

- Every object in R has a class.
- The class of an object tells R what operations can be performed on that object.
- E.g., R can sum two objects that both have the class “numeric”.

Some Basic R Classes

There are many R classes. We'll start with three:

- Numeric.
- Character.
- Logical.

Numeric

Most data are numbers.

```
124.3
```

```
## [1] 124.3
```

```
6
```

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

Character

Surrounded by quotes.

```
"Hello World"
```

```
## [1] "Hello World"
```

```
"6"
```

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

When Will I Encounter Characters?

Characters in Data Sets

- Responses to open-ended questions.
- Responses to categorical variables.
- Dates.

There are special data types for categorical variables and dates which we will introduce later.

Logical

TRUE/FALSE or T/F

```
TRUE
T

FALSE
F
```

Logical Expressions

Logicals are most often encountered as output of a logical expression.

```
1 > 0
```

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

tRy it! Data Types

Use the assignment operator, `<-`, to define R objects with the names and values shown in the table below.

name	class	value
x	numeric	10
y	numeric	4.2
haiku	character	a haiku about R
excited	logical	True or false, are you excited to learn R?

What is a Vector?

- A list of one or more elements.
- All elements are the same type.
- The objects you have created thus far are all vectors of length 1.

To make longer vectors, we need to combine multiple elements together. We do that using a function.

Functions

What's a Function?

- A class of R object.
- Functions contain unexecuted code.
- Calling a function executes its code.

The Form of a Function

Functions take the form:

```
function_name(arg1 = val1, arg2 = val2, ...)
```

- Functions are recognizeable because they have ( ) next to their name.
- `arg` = the name of the argument.
- `val` = the value supplied to the argument.

An Example Function

```
add_two_numbers(x, y) {
  x + y
}
```

This function has the name, `add_two_numbers`, and two arguments, named `x` and `y`. When the function is called, it executes the code `x + y`.

Calling the Example Function

```
add_two_numbers(x = 2, y = 4)
```

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

```
add_two_numbers(x = 9, y = 3)
```

```
## [1] 12
```

Your First Function: `class()`

`class()` is a simple function. It has one argument, named `x`, and returns the class of that object.

```
class(10)
```

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

tRy it! `class()`

Use `class()` to identify the classes of the R objects you created (1.3.3 in the lab manual).

Combine Values into a Vector: c()

The `c( )` function is a special. It accepts any number of arguments and combines them into a vector.

Numeric/Double Vector

```
c(5, 7, 7, 3, 12)
```

```
## [1]  5  7  7  3 12
```

Character Vector

```
c("Hello", "World", "Hello World", 17)
```

```
## [1] "Hello"       "World"       "Hello World" "17"
```

Named Vectors

Optionally, the values passed to `c( )` can be named, which will produce a named vector:

```
c(A = 1, B = 2)
```

```
## A B
## 1 2
```

tRy it! Vectors

Use the assignment operator, `<-`, and the concatenate function, `c( )` to define R objects with the names and values shown in the table below (section 1.3.4 of the lab manual).

name	value
dub	a numeric/double vector of length 3
dub2	a numeric/double vector of length 4

Object Coercion

If some of the objects in `c( )` are different types, `c( )` will coerce them to be the same type.

> *The output type is determined from the highest type of the components in the hierarchy logical < numeric < character.*

tRy it! Object Coerceion

What is the type of each of the following?

```
c(TRUE, 1, 3)

c(TRUE, FALSE, 1 < 0)

c(TRUE, "FALSE")
```

Data Frames

Tabular/Rectangular Data

- Data are often stored in tables.
- Each column is one variable.
- Each row is one case (often but not always a person).

Example of Tabular Data

name	age	group	shs_1	shs_2	shs_3
Declan	17	A	3	4	4
Ava	19	A	4	3	3
Liam	20	B	5	5	4
Charlotte	19	B	4	5	5

Tabular Data in R: `data.frame()`

Data frames are a collection of vectors of the same length.

Each column in a `data.frame` is a vector.

```
##         name age group shs_1 shs_2 shs_3
## 1     Declan  17     A     3     4     4
## 2        Ava  19     A     4     3     3
## 3      Liam  20     B     5     5     4
## 4 Charlotte  19     B     4     5     5
```

Combine Vectors into a Data Frame

- Function works similarly to `c()`.
- Accepts any number of arguments.
- Each argument should be a vector of the same length.
- The vectors become columns in the data frame.
- The name of the argument becomes the name of the column.

tRy it! Make a `data.frame`

Follow along with section 1.3.5 of your lab manual to create this data frame in R:

name	age	group	shs_1	shs_2	shs_3
Declan	17	A	3	4	4
Ava	19	A	4	3	3
Liam	20	B	5	5	4
Charlotte	19	B	4	5	5

Extract: $

To extract columns from a data.frame object, use $.

```
df$shs_1
```

```
## [1] 3 4 5 4
```

Add a New Column: $

If you use $ and <- to assign a value to a column that is not in your `data.frame`, R will create a a new column for that `data.frame`.

Use this to create the `shs_tot` column described, as described in your lab manual.

Documentation: help()

Probably the most popular function.

```
help(c)

# OR

?c
```

How to Read an R Help Page[2]

The name of the function, and the library it is in.

mean {base} R Documentation

Arithmetic Mean

Description

What it does.

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

The function's name, and in the parentheses the named arguments it expects, in the order it expects them. If an argument has a default value, it is shown. Arguments without default values (e.g. x) must be provided by you.

Arguments

More details on each named argument. This will tell you what class of thing each argument has to be—an object, a number, a data frame, a logical value, etc.

`x`	An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.
`trim`	the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
`na.rm`	a logical value indicating whether `NA` values should be stripped before the computation proceeds.
`...`	further arguments passed to or from other methods.

The ellipsis allows other arguments to be passed to and from the function.

Value

What the function returns—i.e., the result of whatever operation or calculation it performs. This can be a single number, as here, or a multi-part object such as a list, a data frame, a plot, or a model.

If `trim` is zero (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If x is not logical (coerced to numeric), numeric (including integer) or complex, `NA_real_` is returned, with a warning.
If `trim` is non-zero, a symmetrically trimmed mean is computed with a fraction of `trim` observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

`weighted.mean`, `mean.POSIXct`, `colMeans` for row and column means.

Other related functions

Examples

Self-contained examples that you can

```
x <- c(0.10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

Examples of how to use the functions. These can be run at the console. These may use built-in datasets or other R functions.

[Package *base* version 3.4.3 Index]

Visit the package's Index page to look for Demos and Vignettes detailing how it works.

src: ***A.1.1 How to read an R help page*** **in "Data Visualization: A practical introduction" Kieran Healy (2018).** <http://socviz.co/>

References

References

1. https://www.r-project.org/about.html
2. https://socviz.co/appendix.html#a-little-more-about-r