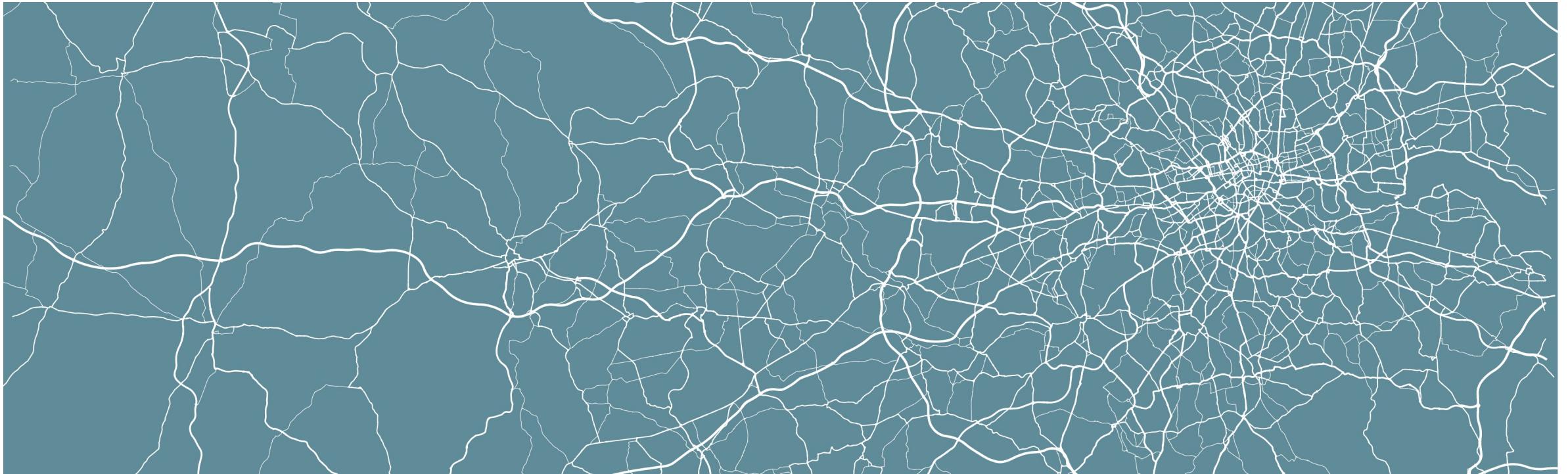


# Geocomputation

## Programming for Data Analysis



# Where are we at?

## *Part I: Foundational Concepts*

W1 Geocomputation: An Introduction

W2 GIScience and GIS Software

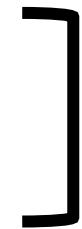
W3 Cartography and Visualisation



QGIS

W4 **Programming for Data Analysis**

W5 Programming for Spatial Analysis



R

# This week

- Short recap
- What is a programming language?
- Principles of R

# Before we start

- Go to [www.menti.com](http://www.menti.com)
- Use code: 7521 51 1

# Recap

- Vector versus raster
- Joining attributes to spatial (vector layers)
- Data classification, idea of MAUP
- Geographic Coordinate Systems and Projected Coordinate Systems



8	9	9	10	0	10	10	10	0	0	0	7	5	3	0	0	0	0	1
8	9	9	10	10	0	10	9	9	0	0	5	3	0	0	0	0	0	0
8	8	9	9	10	0	0	9	8	7	5	0	0	0	1	0	0	0	0
5	8	8	9	10	10	0	9	7	5	0	0	5	5	5	0	0	0	1
3	5	8	9	9	10	0	0	3	0	0	0	5	0	0	1	0	0	2
2	5	8	8	9	9	10	0	0	0	1	5	0	0	0	0	0	0	1
2	4	6	8	8	9	0	0	0	1	5	0	0	5	5	5	0	0	1
0	3	6	8	8	0	0	0	0	5	0	5	5	5	5	5	0	0	0
2	2	5	8	0	0	0	0	0	0	5	5	5	5	5	5	3	0	0
0	2	5	0	0	1	2	3	4	4	4	4	4	4	4	5	0	0	0
0	0	0	0	1	1	1	1	4	4	4	4	4	4	4	5	0	0	0
0	0	1	1	2	2	2	2	3	3	3	3	3	3	3	4	0	3	0
1	1	1	1	2	2	3	3	3	3	1	1	1	1	1	2	3	4	3

# Attribute join

Table 1



1		
2		

Table 2



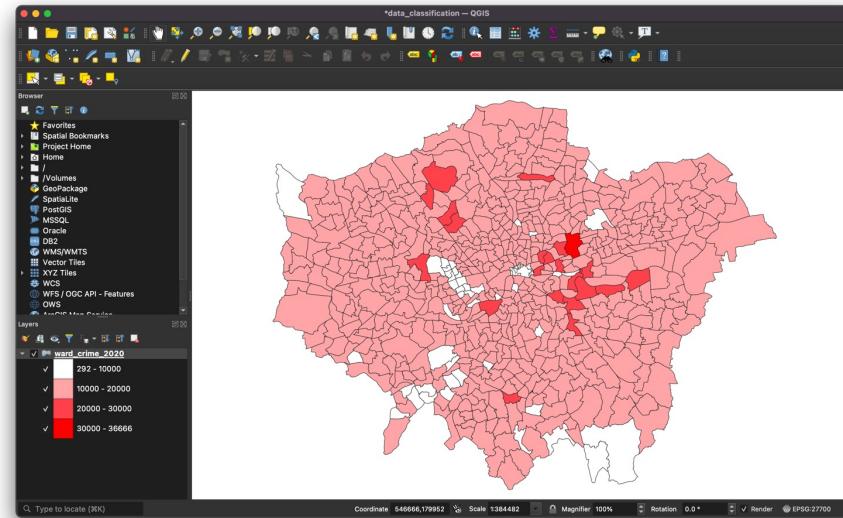
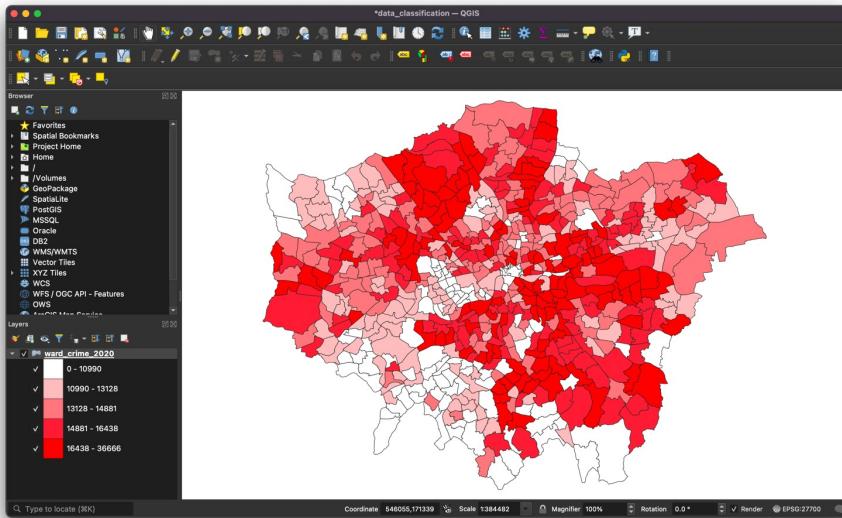
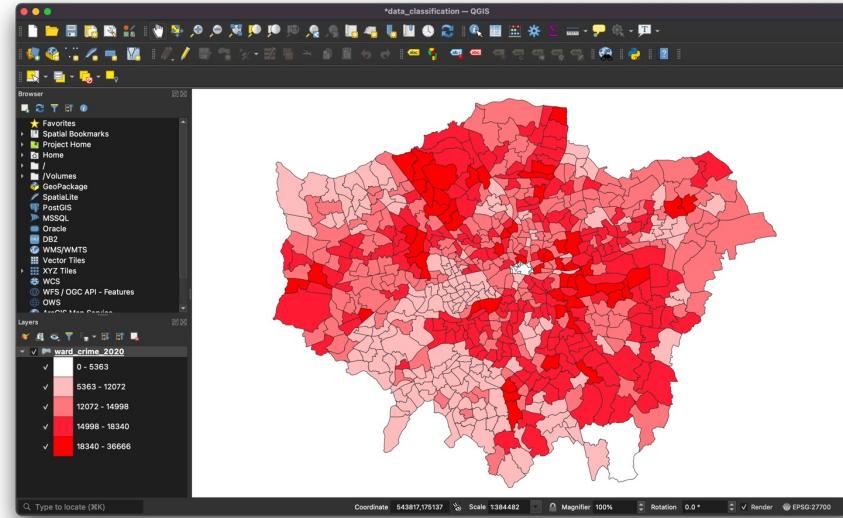
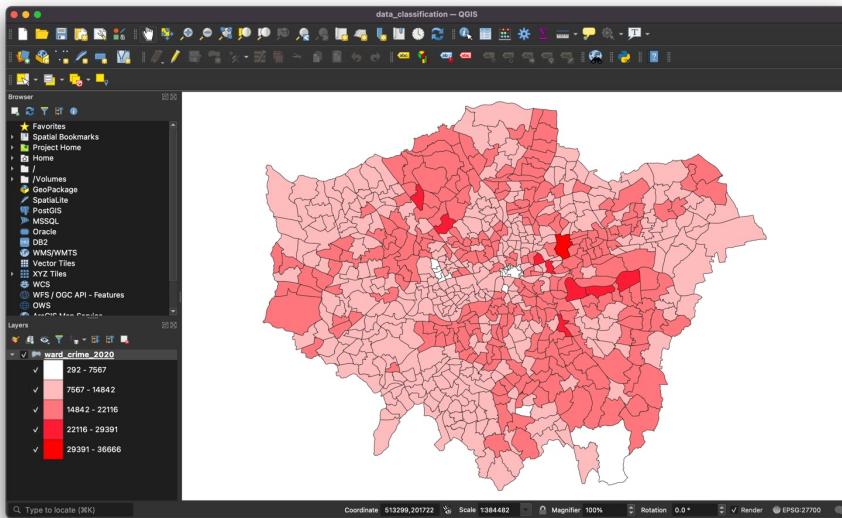
1		
3		
4		

Left Join

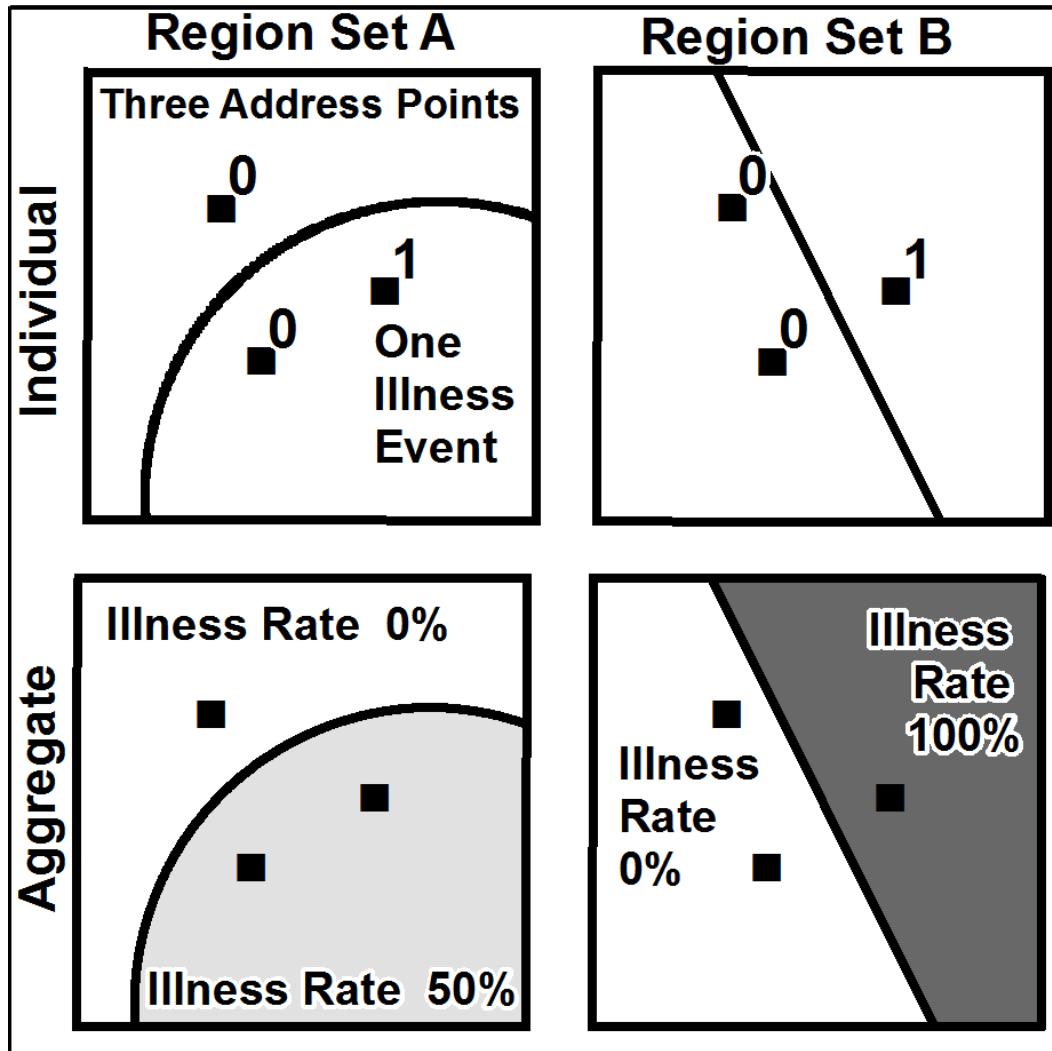


1			
2			

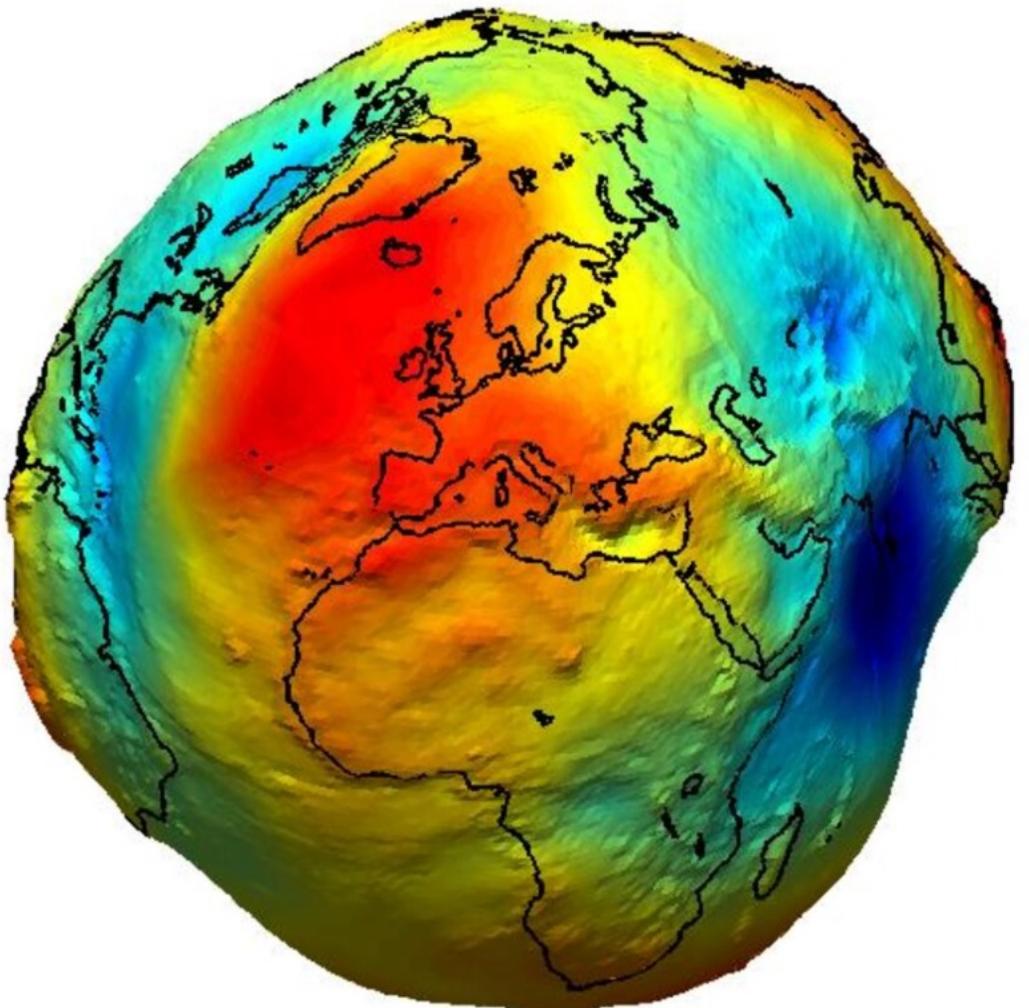
# Classifying data



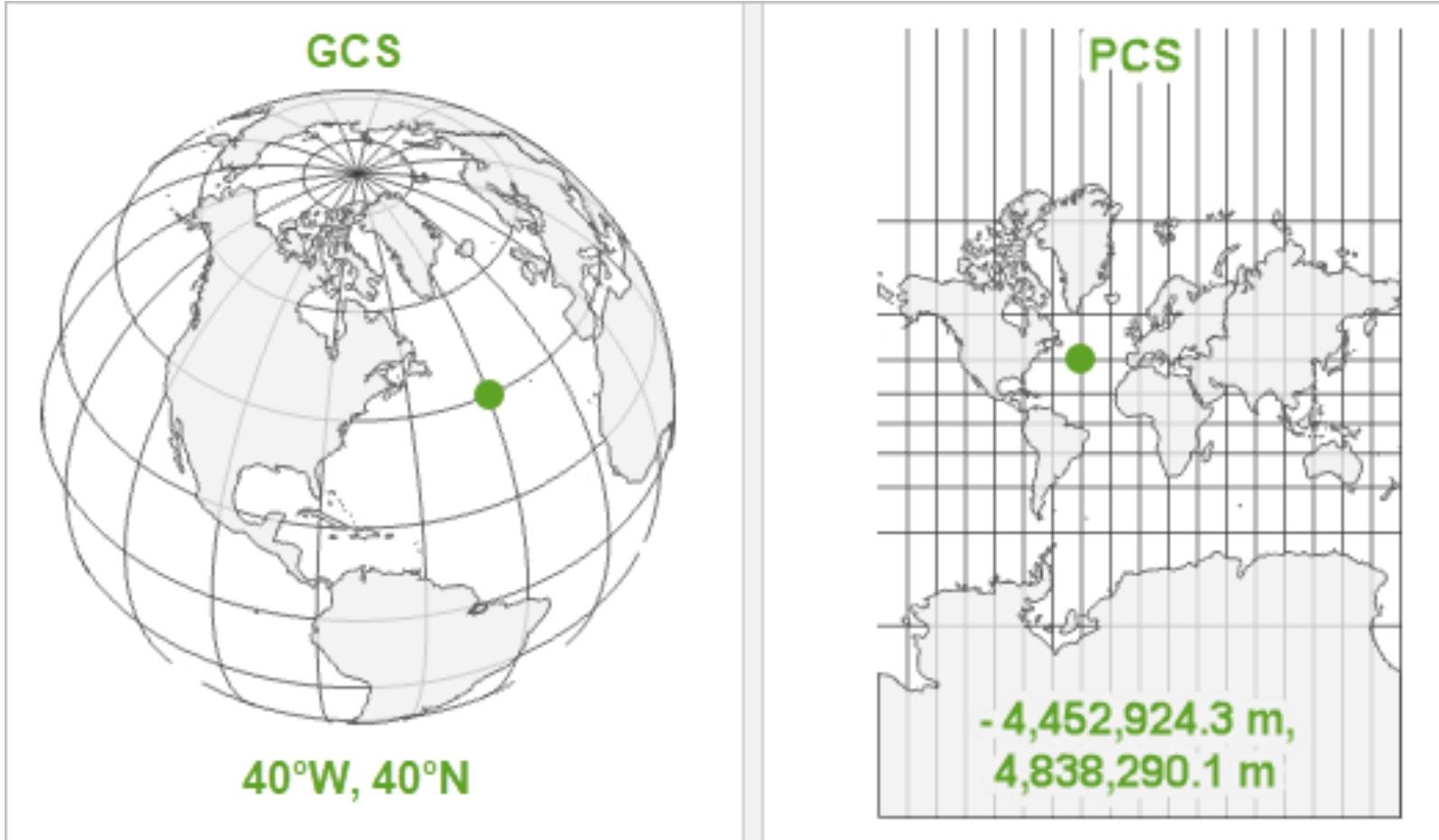
# Modifiable Areal Unit Problem



# Representing the globe



# Moving from an ellipsoid to a plane



# QGIS

- Powerful open-source GIS, widely used both within and outside of academia.
- Extensive documentation: [\[Link\]](#)
- Indispensable when learning about core spatial analysis / GIScience concepts:  
layering spatial data, attribute joins
- However (1): QGIS can be a bit **fiddly** at times. *Aligning multiple maps on a canvas?*
- However (2): Graphical User Interface involves lots of manual actions, time-involved to repeat.

# Programming languages

"Everyone does need to learn to code. It is no longer sufficient for a GI Scientists to just work with a standard GIS interface: menus, buttons and black boxes."

Brunsdon and Comber 2020

# Programming languages

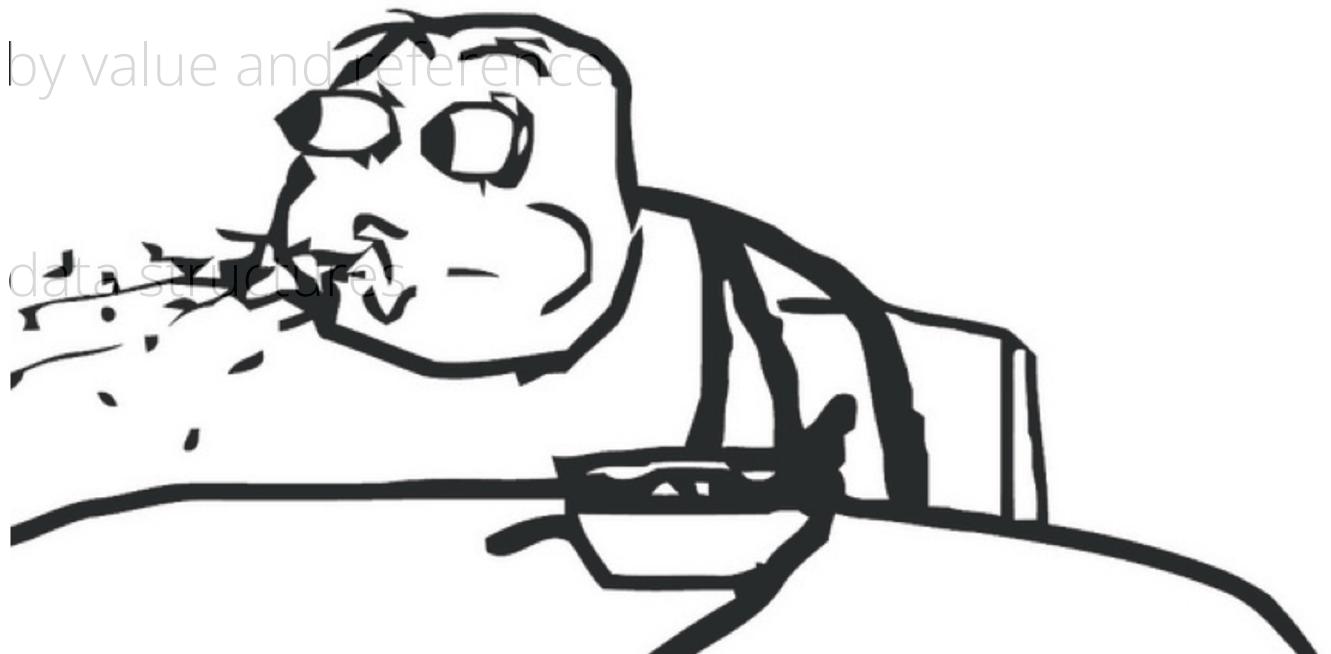
- From "point-and-click" to "writing commands".
- We will be using a programming language called R.
- Reproducible research

# Programming languages

- Identifiers and primitive data types
- Assignment, arithmetic, logical and relational operators
- Expression and statements, debugging
- Flow of control: selection and repetition
- Functions, parameters passing, call by value and reference
- Object-oriented programming
- 1/2 dimensional arrays, strings and data structures

# Programming languages

- Identifiers and primitive data types
- Assignment, arithmetic, logical and relational operators
- Expression and statements, debugging
- Flow of control: selection and repetition
- Functions, parameters passing, call by value and reference
- Object-oriented programming
- 1/2 dimensional arrays, strings and data structures



# Programming languages

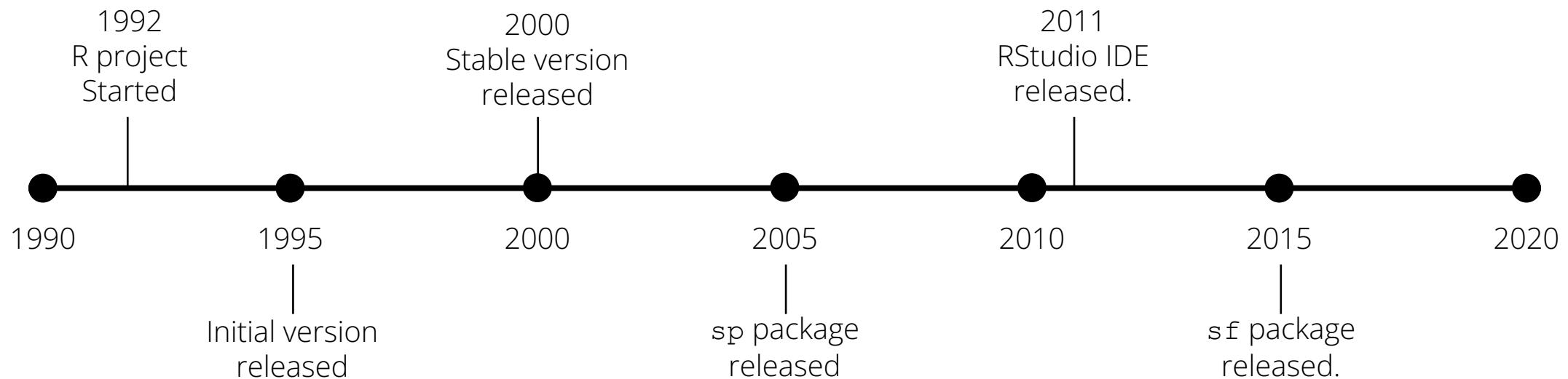
- Ιδεντιφιερς ανδ πριμιτιφε δατα τψπες
- Ασσιγνμεντ, αριτημετιξ, λογιξαλ ανδ ρελατιοναλ οπερατορς
- Εχπρεσσιον ανδ στατεμεντς, δεβυγγινγ
- Φλωρ οφ ξοντρολ· σελεξτιον ανδ ρεπετιτιον
- Φυνξτιονς, παραμετερς πασσινγ, ξαλλ βψ αφλυε ανδ ρεφερενξε
- Οβσεξτ-οριεντεδ προγραμμινγ
- 1/2 διμενσιοναλ αρραψ, στρινγς ανδ δατα στρυξτυρες

# What is R?

- R is used primarily through interactive command-line.
- R can create and use different types of data but works predominantly very well with a linear collection of things (vector) and tables.
- R is extremely extendable through packages.
- R is brilliant when it comes to static graphics and dealing with spatial data.

# A little history

- R is programming language which specialises in statistics.
- 1975: Bell Labs develops a language for Statistical Analysis
- 1992: Ross Ihaka and Robert Gentleman develop opensource version of S



# Principles of R

- R is used primarily through interactive command-line.
- R can create and use different types of data but works predominantly very well with a linear collection of things (vector) and tables.
- R is extremely extendable through packages.
- R is brilliant when it comes to static graphics and dealing with spatial data.

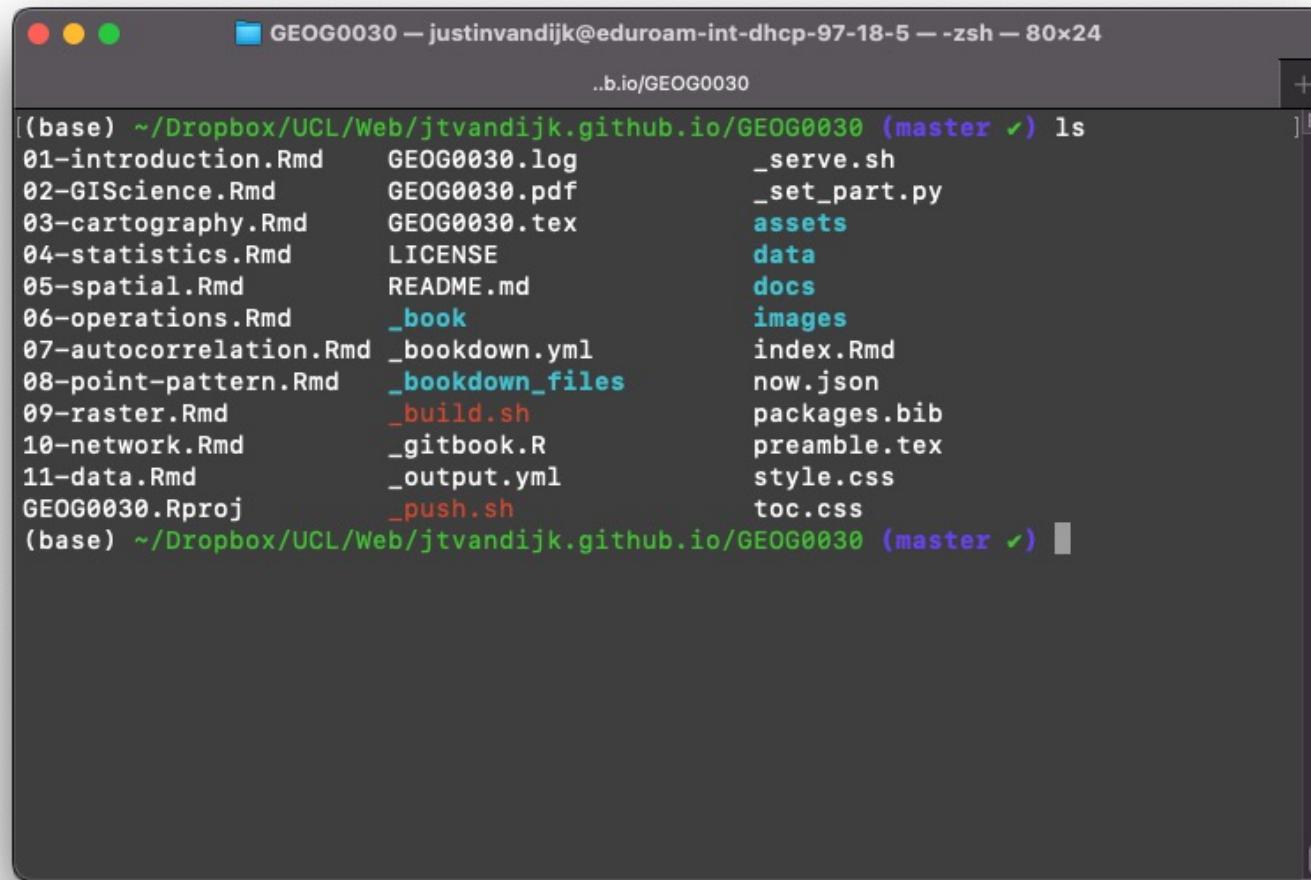
# Command-Line Interface

- Command-Line Interface versus (CLI) Graphical User Interface (GUI)
- The GUI allows the user to interact with the system using graphical elements such as windows, icons, menus while the CLI allows the user to interact with the system using commands.

# Graphical User Interface



# Command-Line Interface



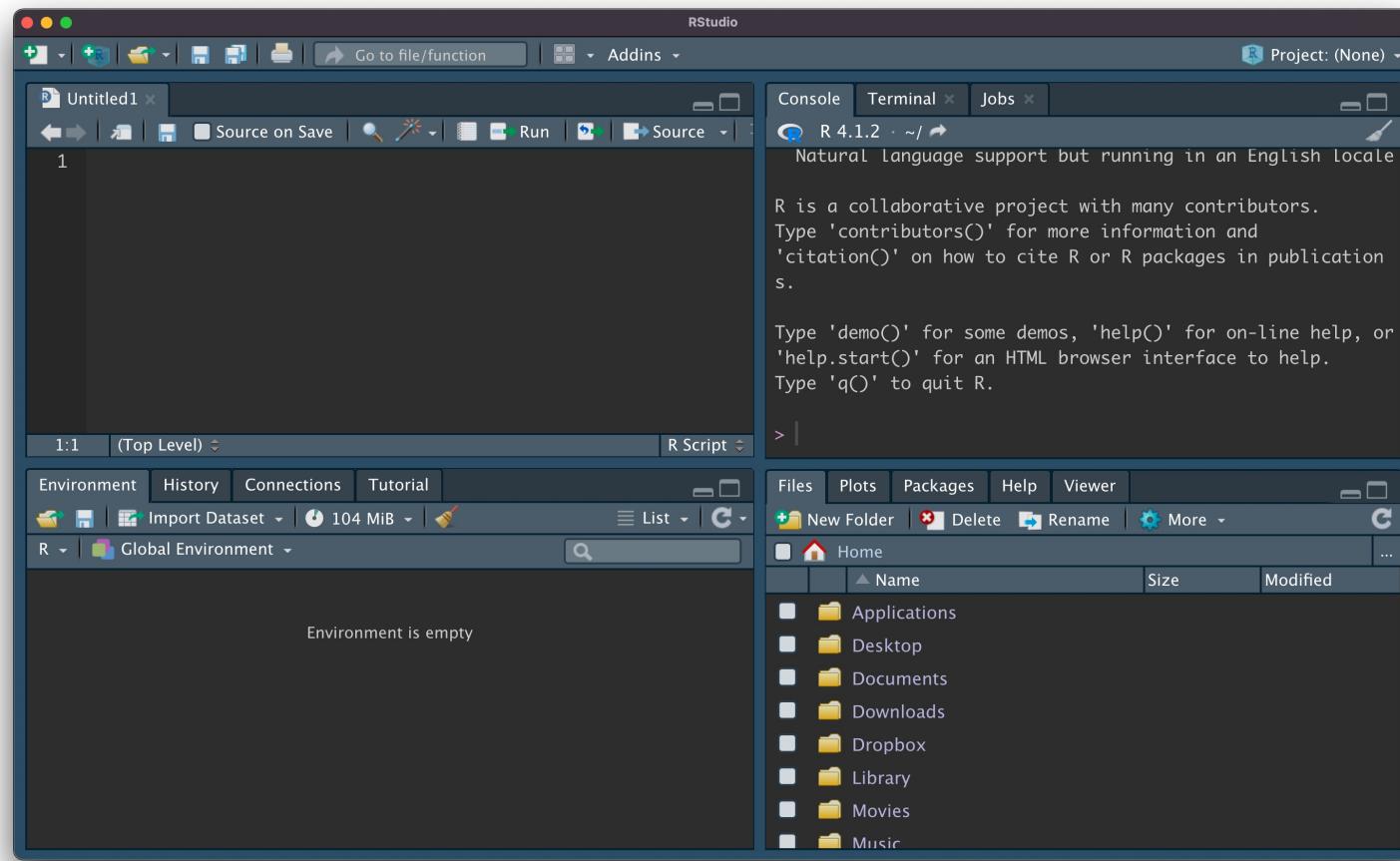
The screenshot shows a macOS terminal window titled "GEOG0030 — justinvandijk@eduroam-int-dhcp-97-18-5 — zsh — 80x24". The window displays the results of the "ls" command in a directory structure. The output is as follows:

```
(base) ~/Dropbox/UCL/Web/jtvandijk.github.io/GEOG0030 (master ✓) ls
01-introduction.Rmd    GEOG0030.log          _serve.sh
02-GIScience.Rmd       GEOG0030.pdf         _set_part.py
03-cartography.Rmd     GEOG0030.tex        assets
04-statistics.Rmd      LICENSE             data
05-spatial.Rmd         README.md          docs
06-operations.Rmd      _book               images
07-autocorrelation.Rmd _bookdown.yml      index.Rmd
08-point-pattern.Rmd   _bookdown_files    now.json
09-raster.Rmd          _build.sh          packages.bib
10-network.Rmd         _gitbook.R          preamble.tex
11-data.Rmd            _output.yml        style.css
GEOG0030.Rproj         _push.sh           toc.css
(base) ~/Dropbox/UCL/Web/jtvandijk.github.io/GEOG0030 (master ✓)
```

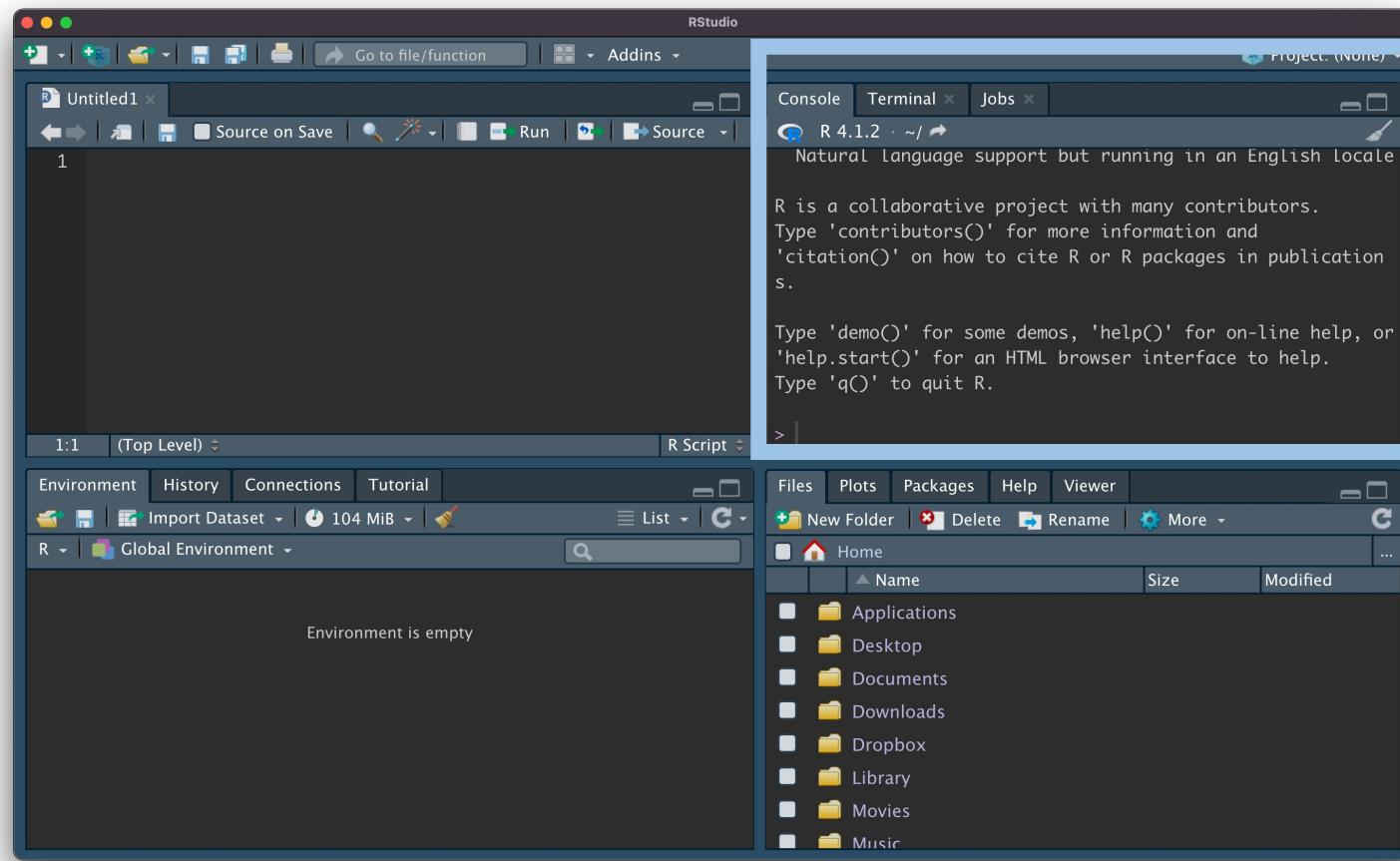
# Interaction through language

- Capturing complex instruction with language is much easier than with skeuomorphism.
- Repeating stuff is easy.
- Much steeper learning curve but greater rewards as well.

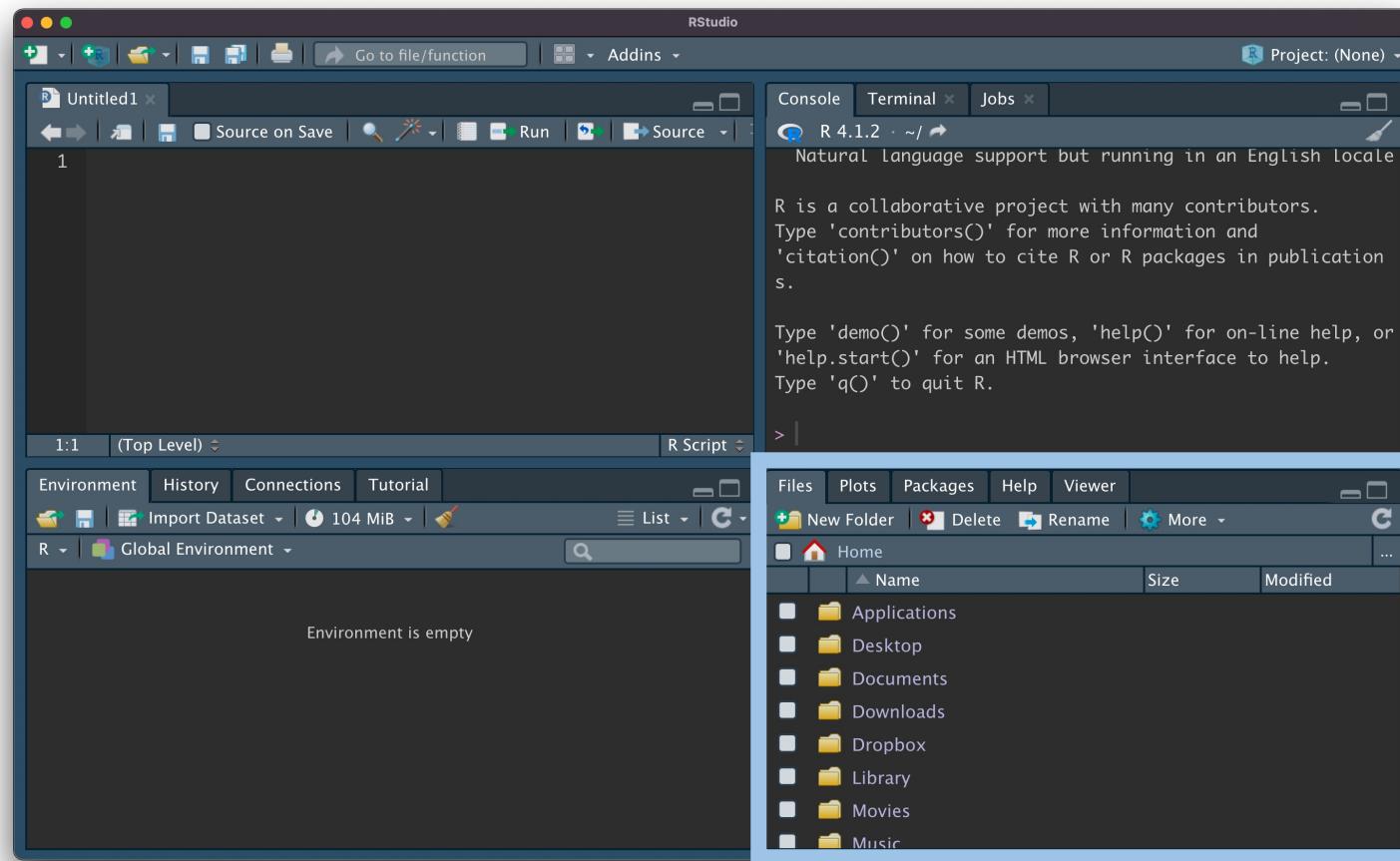
# How does this work in R



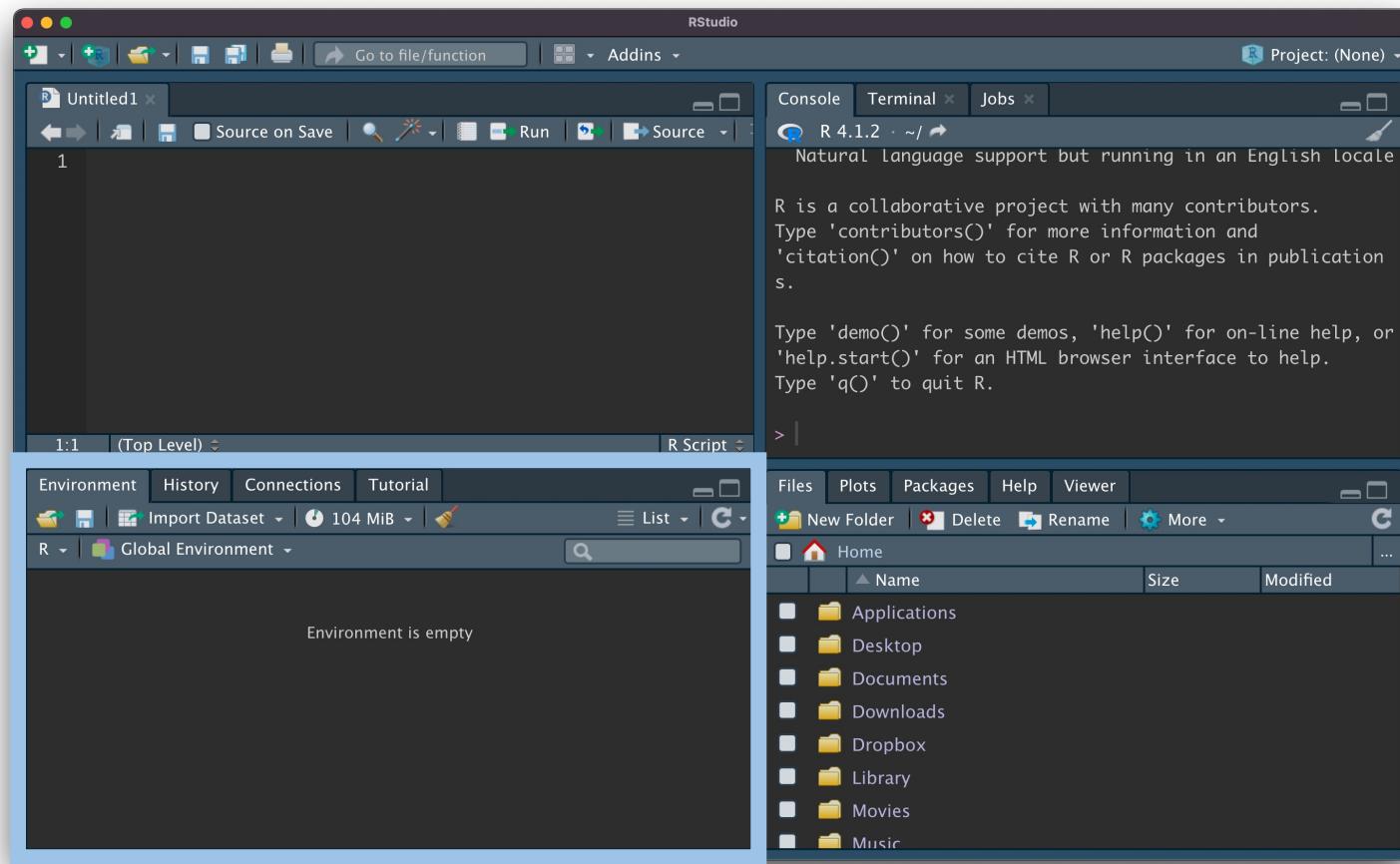
# How does this work in R



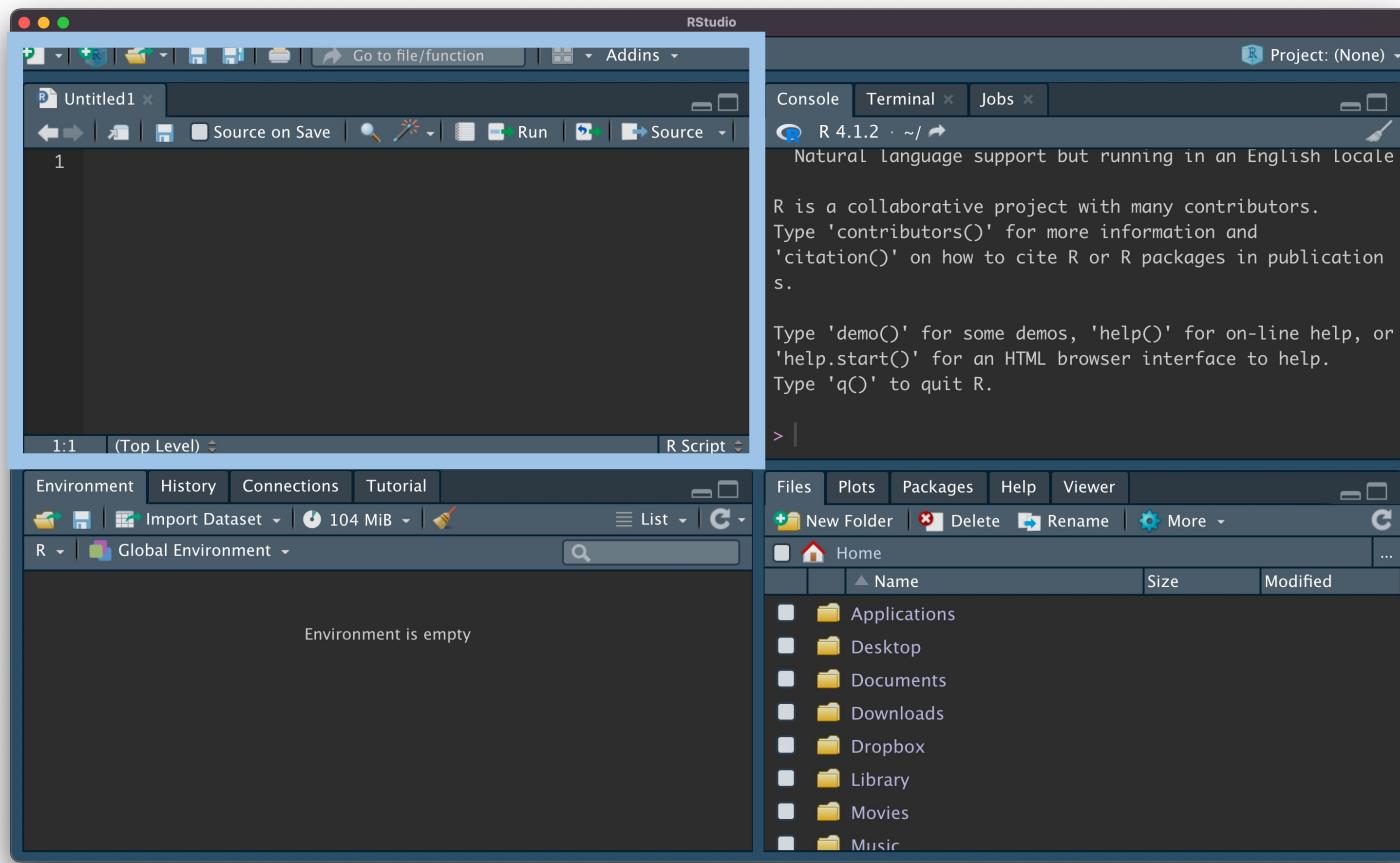
# How does this work in R



# How does this work in R



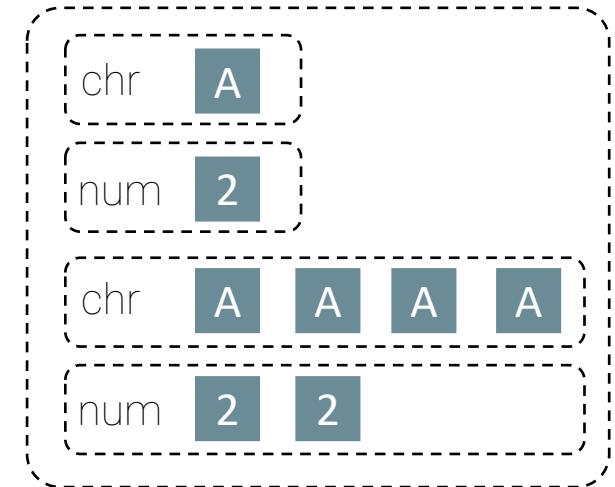
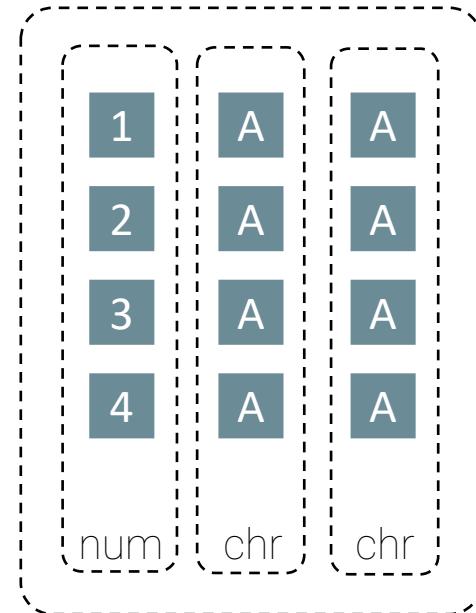
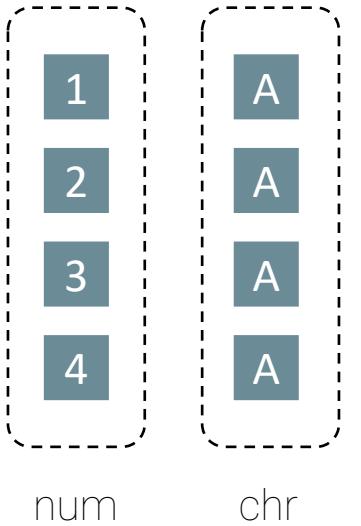
# How does this work in R



# Principles of R

- R is used primarily through interactive command-line.
- R can create and use different types of data but works predominantly very well with a linear collection of things (vector) and tables.
- R is extremely extendable through packages.
- R is brilliant when it comes to static graphics and dealing with spatial data.

# Data types



Scalar

Vector

Dataframe

List

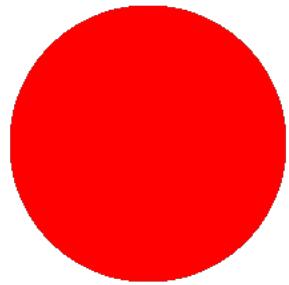
# Variables

- All scalars, vectors, tables, and lists can be assigned to a variable.
- Variables are used to store information to be referenced and manipulated in a computer programme.

# Functions

- Variables can be used as an input for functions.
- Functions are pieces of code that accomplish a specific task.
- Functions usually "take in" data, process it, and "return" a result.
- Once a function is written, it can be used over and over and over again.

RStudio

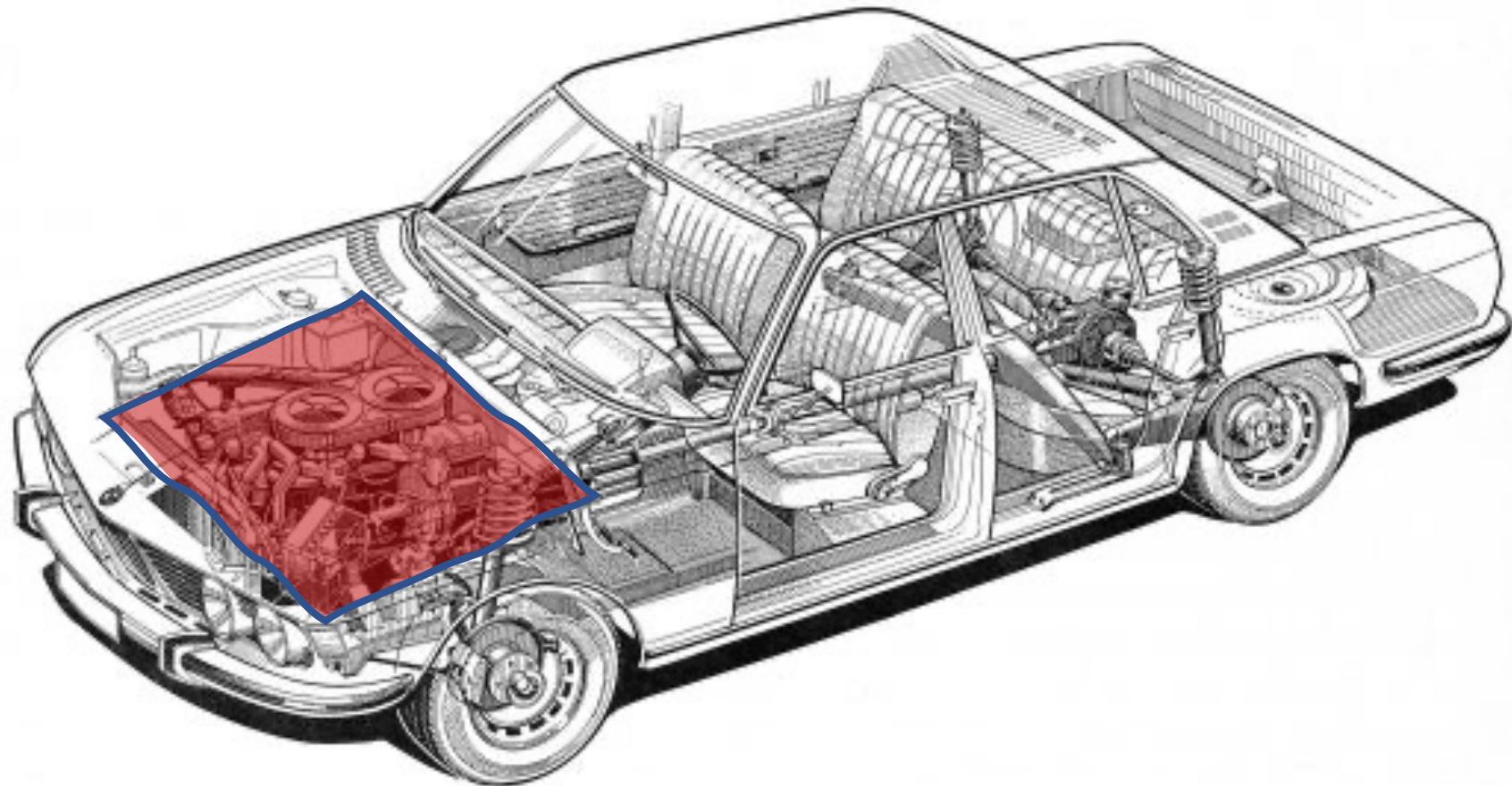


LIVE

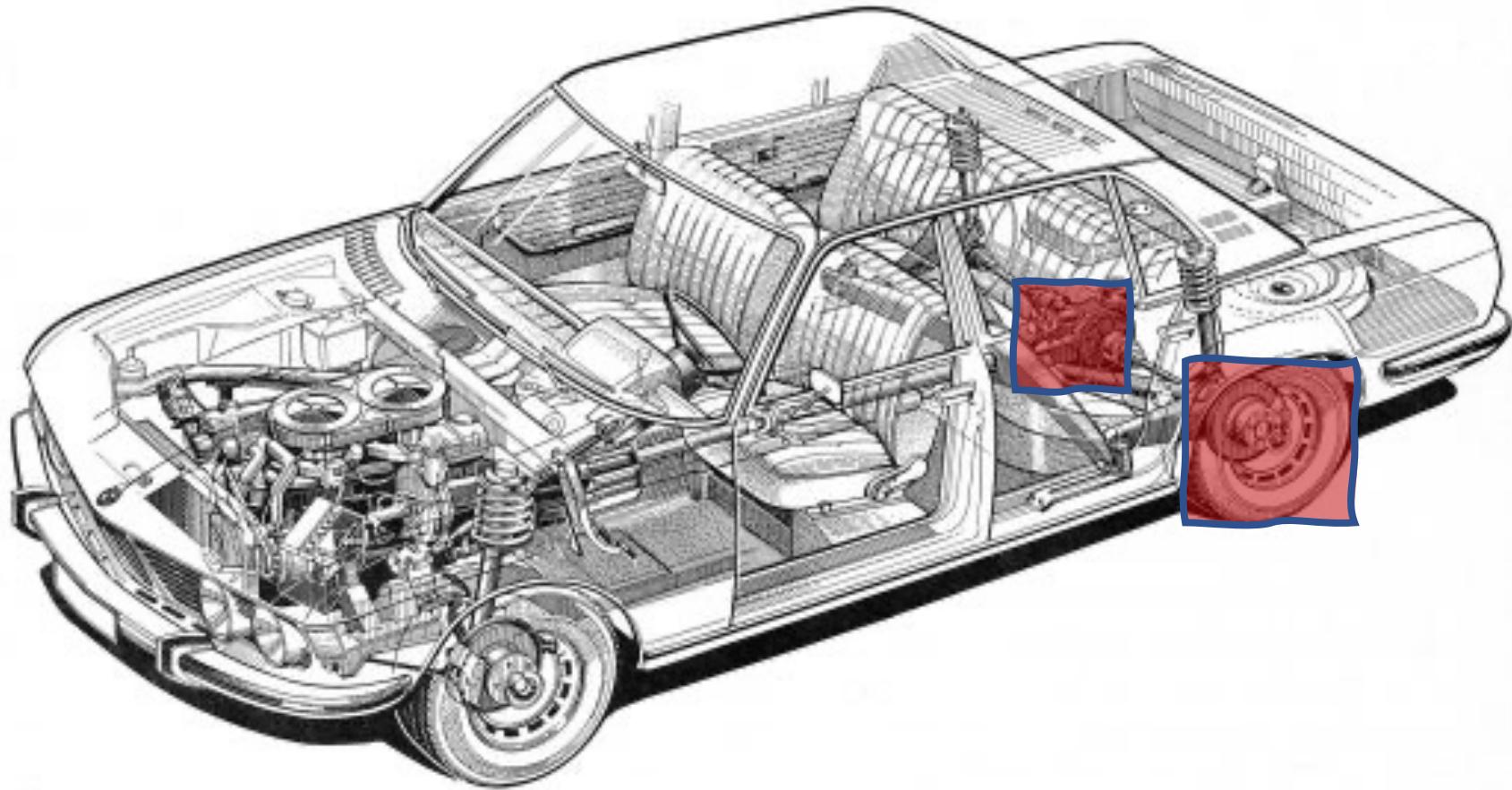
# Principles of R

- R is used primarily through interactive command-line.
- R can create and use different types of data but works predominantly very well with a linear collection of things (vector) and tables.
- R is extremely extendable through packages.
- R is brilliant when it comes to static graphics and dealing with spatial data.

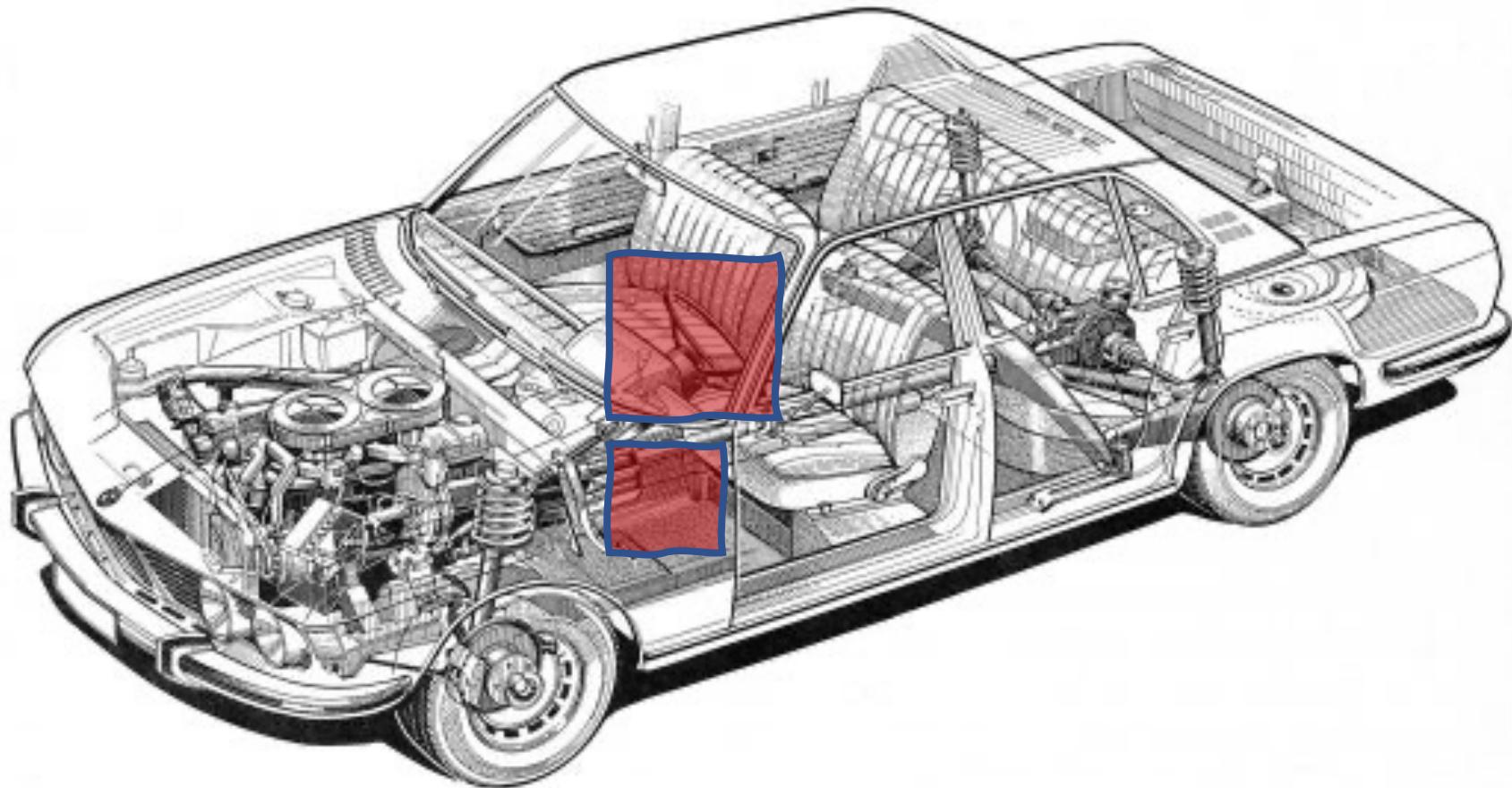
# Packages



# Packages



# Packages



# Packages

- Available on The Comprehensive R Archive Network ([CRAN](#)).
- CRAN package repository features over 19,103 available packages (26/01/2023).
- Packages provide extensions to R.

# tensorflow

The screenshot shows the README.md page for the tensorflow R package on GitHub. The page is titled "TensorFlow for R". It includes a green "R-CMD-check passing" badge and a "CRAN 2.7.0" badge. The main text describes TensorFlow as an open source software library for numerical computation using data flow graphs. It mentions that nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. Below this, it states that the TensorFlow API is composed of a set of Python modules that enable constructing and executing TensorFlow graphs. The tensorflow package provides access to the complete TensorFlow API from within R.

## Installation

To get started, install the tensorflow R package from GitHub as follows:

```
devtools::install_github("rstudio/tensorflow")
```

Then, use the `install_tensorflow()` function to install TensorFlow:

```
library(tensorflow)
install_tensorflow()
```

You can confirm that the installation succeeded with:

```
hello <- tf$constant("Hello")
```

On the right side of the GitHub interface, there are sections for "Contributors" (+ 10 contributors), "Environments" (1 environment, github-pages Active), and "Languages" (R 100.0%).

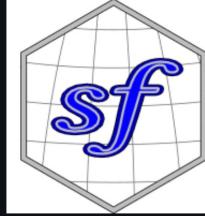
sf

sf

Simple Features for R

A package that provides simple features access for R. Package sf:

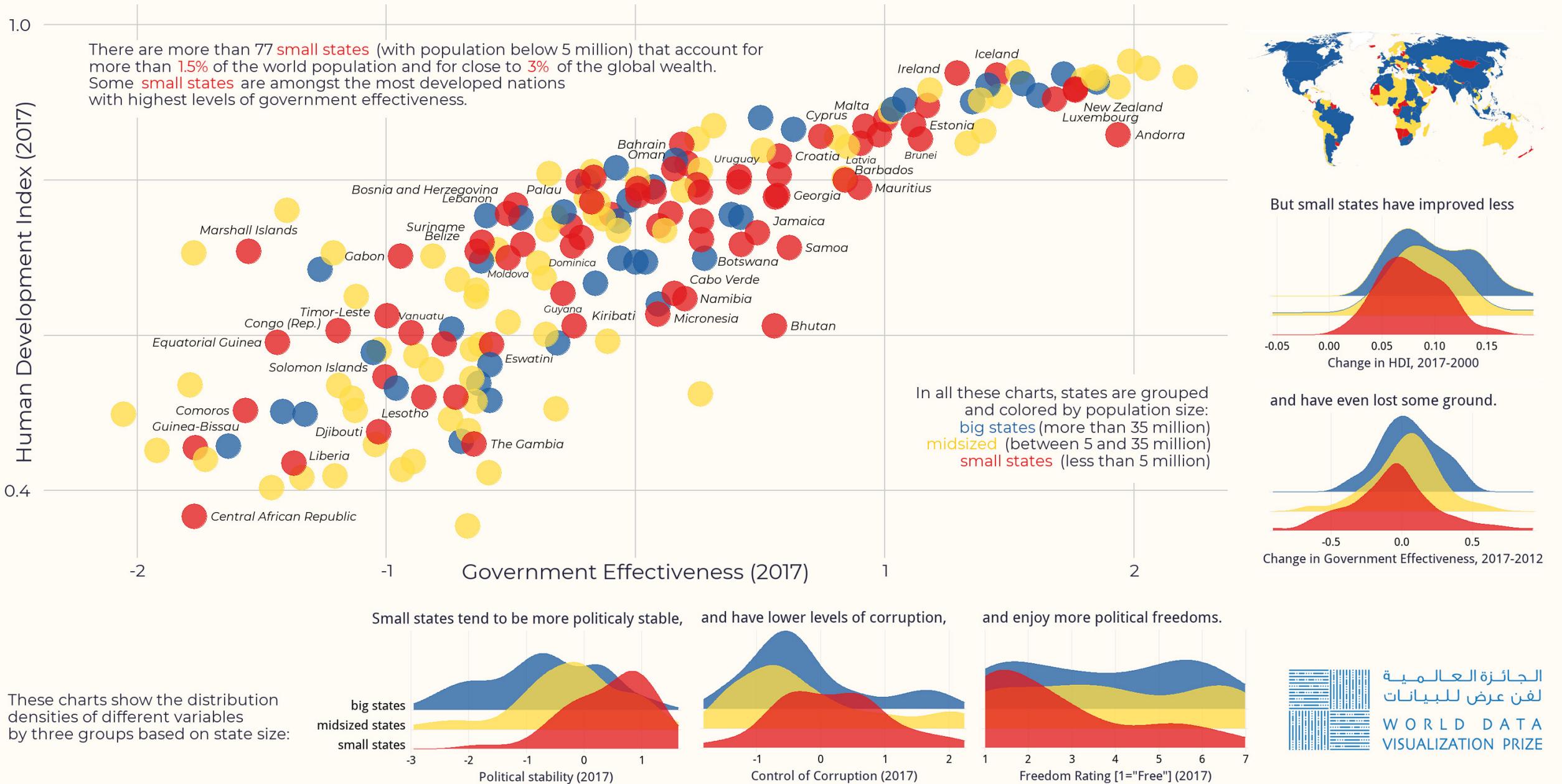
- represents simple features as records in a `data.frame` or `tibble` with a geometry list-column
- represents natively in R all 17 simple feature types for all dimensions (XY, XYZ, XYM, XYZM)
- interfaces to GEOS for geometrical operations on projected coordinates, and to s2geometry for geometrical operations on ellipsoidal coordinates
- interfaces to GDAL, supporting all driver options, `Date` and `POSIXct` and list-columns
- interfaces to PROJ for coordinate reference system conversion and transformation
- uses well-known-binary serialisations written in C++/Rcpp for fast I/O with GDAL and GEOS
- reads from and writes to spatial databases such as PostGIS using DBI
- is extended by lwgeom for selected liblwgeom/PostGIS functions
- is extended by stars for raster data, and raster or vector data cubes (spatial time series)



# Principles of R

- R is used primarily through interactive command-line.
- R can create and use different types of data but works predominantly very well with a linear collection of things (vector) and tables.
- R is extremely extendable through packages.
- R is brilliant when it comes to static graphics and dealing with spatial data.

# Small States Can Be Big Players in Development and Good Governance



# London Cycle Hire Journeys

Thicker, yellower lines mean more journeys





5 more reasons on why you should use R

R is free as in

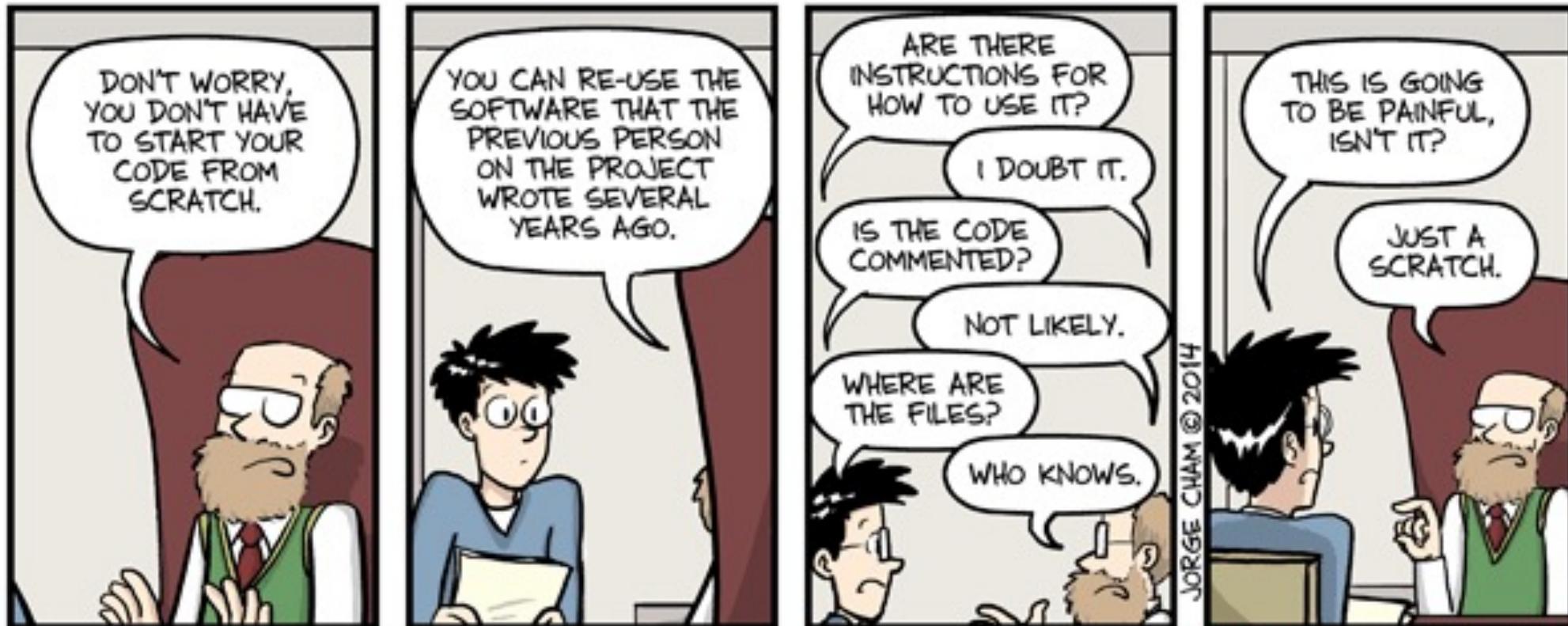


Freedom

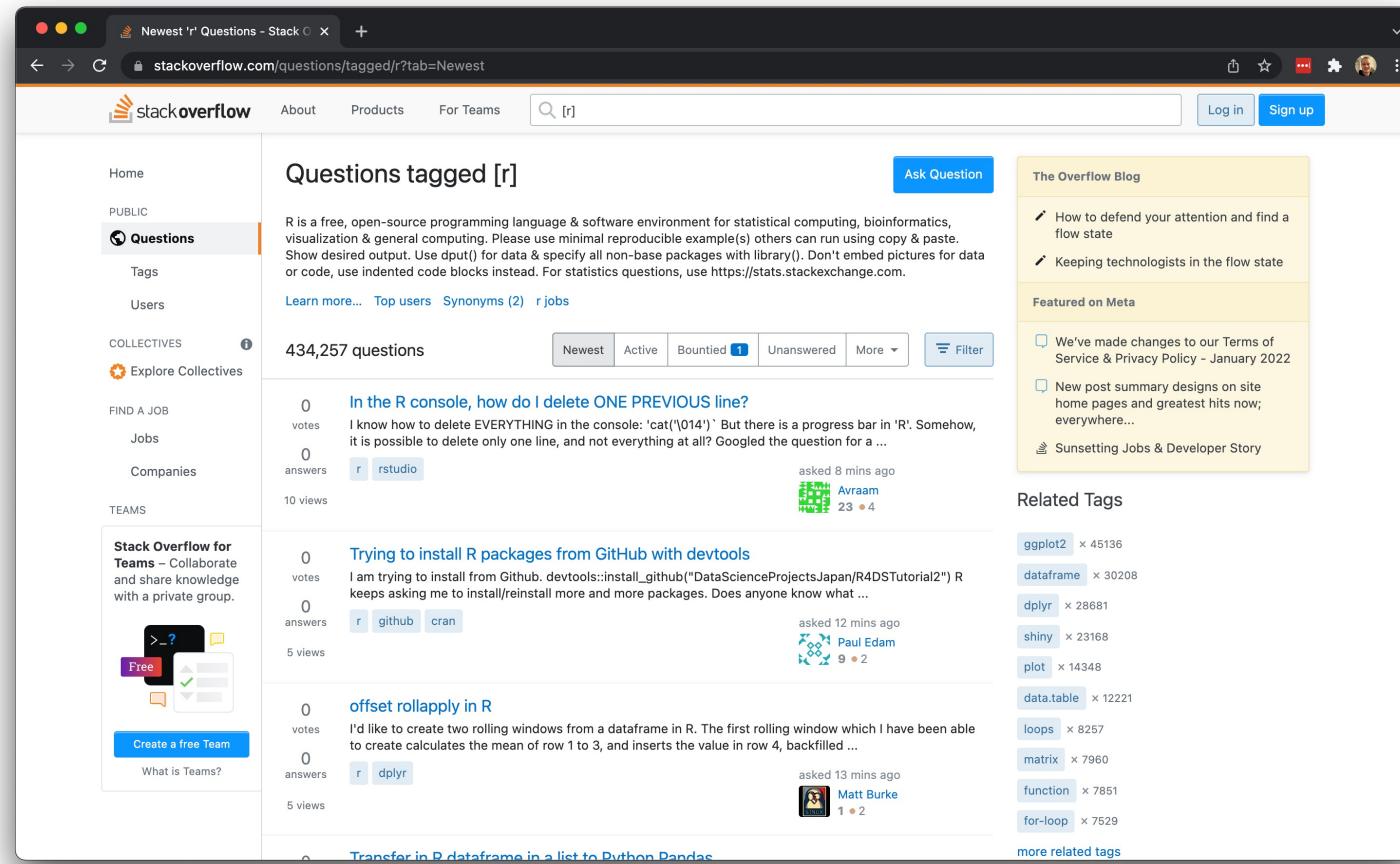


Free beer

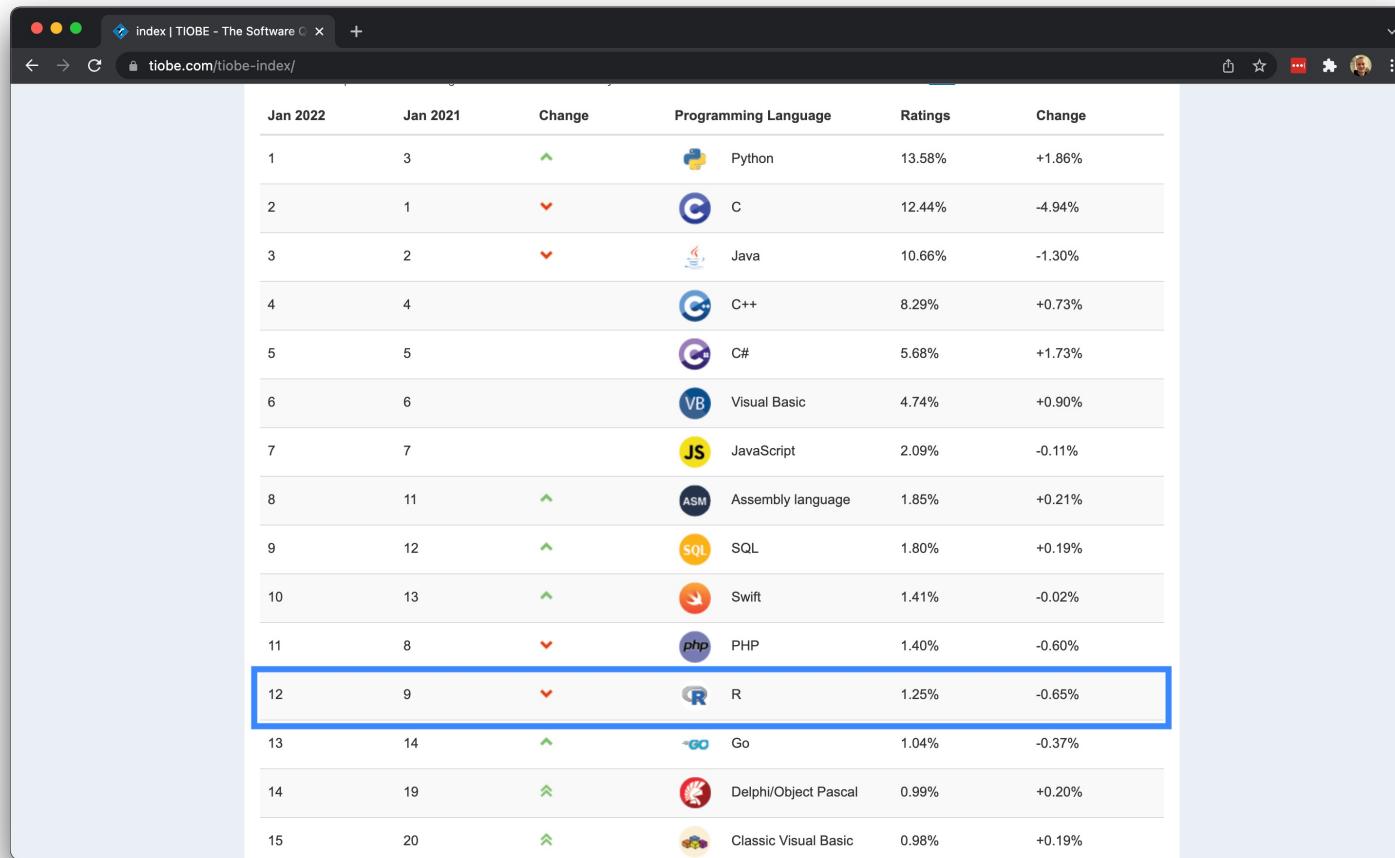
# R allows you to produce your outputs programmatically



# R is supported by a large vibrant community



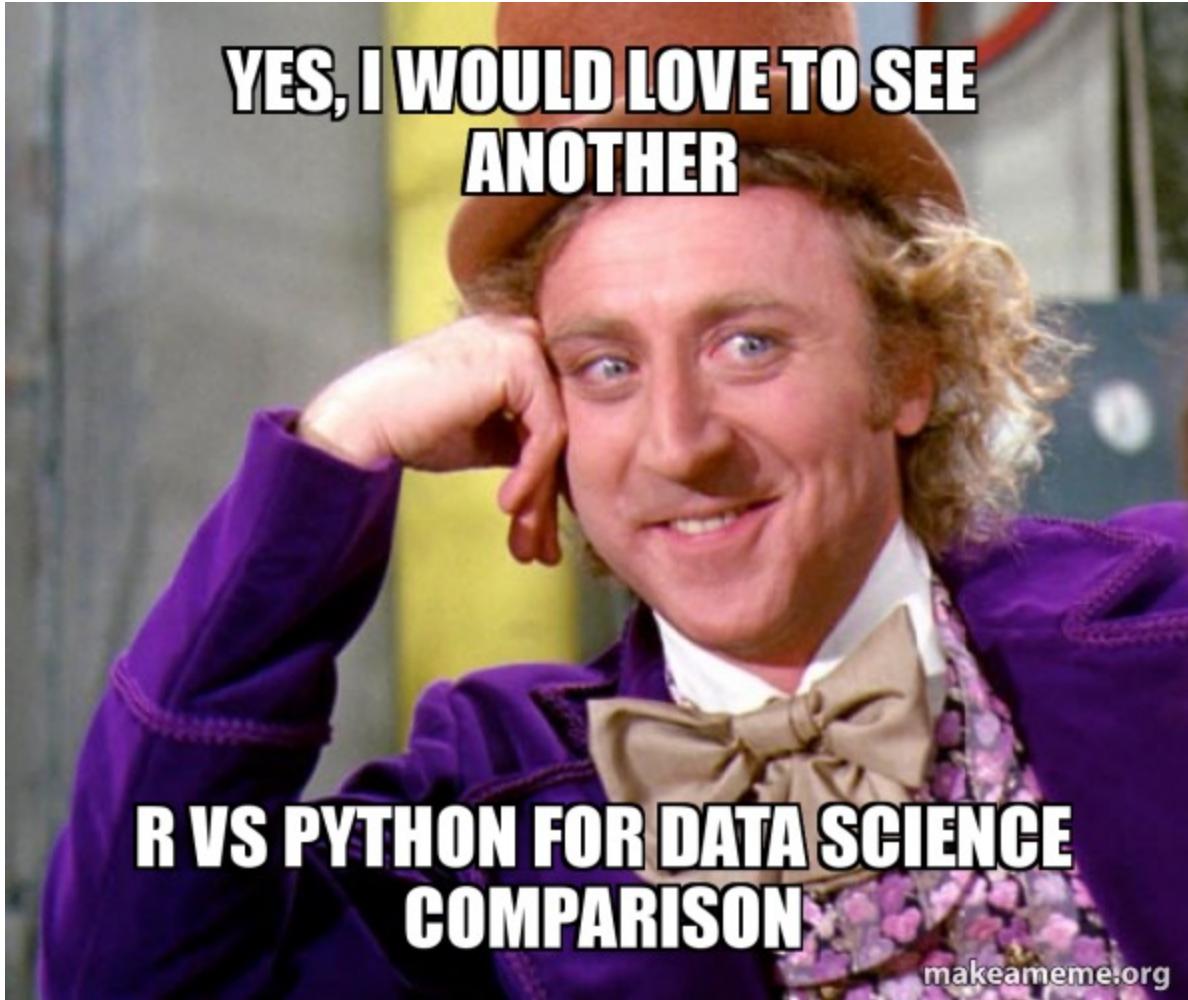
# R is popular and in-demand



The screenshot shows a web browser displaying the TIOBE Software Index for January 2022. The table lists the top 15 programming languages based on popularity. R is highlighted with a blue border around its row.

	Jan 2022	Jan 2021	Change	Programming Language	Ratings	Change
1	3	▲		Python	13.58%	+1.86%
2	1	▼		C	12.44%	-4.94%
3	2	▼		Java	10.66%	-1.30%
4	4			C++	8.29%	+0.73%
5	5			C#	5.68%	+1.73%
6	6			Visual Basic	4.74%	+0.90%
7	7			JavaScript	2.09%	-0.11%
8	11	▲		Assembly language	1.85%	+0.21%
9	12	▲		SQL	1.80%	+0.19%
10	13	▲		Swift	1.41%	-0.02%
11	8	▼		PHP	1.40%	-0.60%
12	9	▼		R	1.25%	-0.65%
13	14	▲		Go	1.04%	-0.37%
14	19	▲		Delphi/Object Pascal	0.99%	+0.20%
15	20	▲		Classic Visual Basic	0.98%	+0.19%

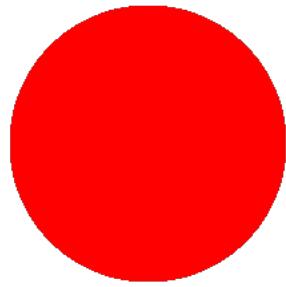
Python?



# R Markdown

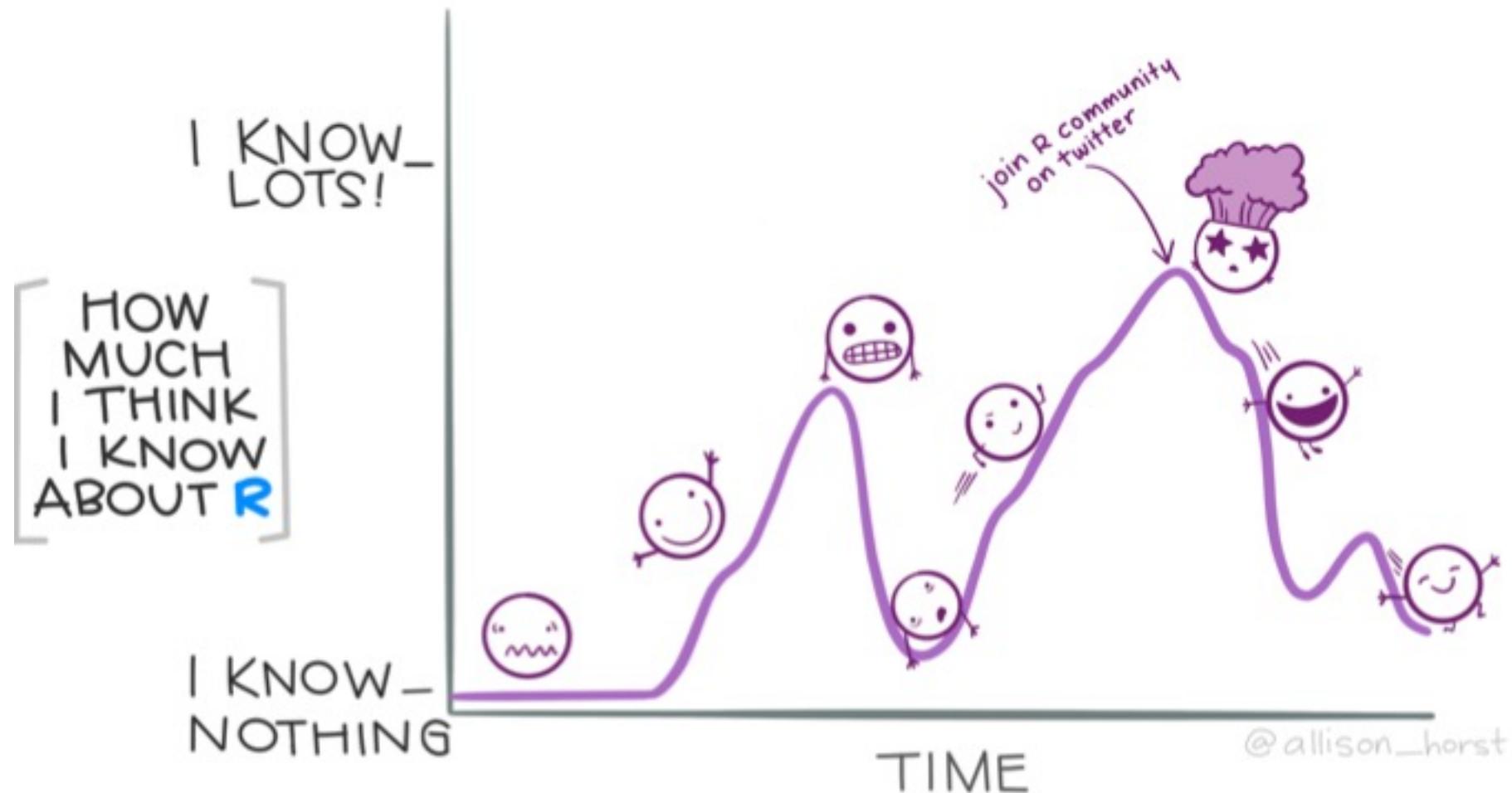
- Markdown is a lightweight **markup language** for creating formatted text using a plain-text editor.
- R Markdown is an extension of the markdown syntax that enables R code to be embedded in them in a way which can later be executed.
- Why do we want this? Typesetting.

RStudio



LIVE

# Learning curve



# Conclusion

- R comes from the rock-stars of the computer science industry.
- R is primarily command line based.
- R is extremely powerful, versatile and popular.
- R is free and Open Source.
- Plenty of tools and community around R.
- It is arguably one of the best transferable skills you can learn.
- Supports reproducible academic research.

# Further resources

## Analysis in R

- [R for Data Science](#)
- [Advanced R](#)
- [Geocomputation in R](#)

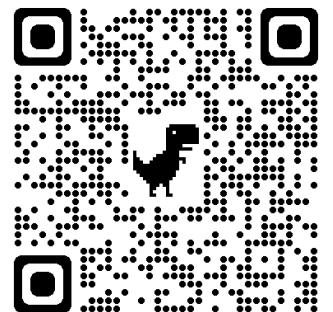
## R Markdown

- R Markdown [resources on Rstudio](#)
- [Definitive Guide to R Markdown](#)

# Questions

Justin van Dijk

j.t.vandijk@ucl.ac.uk



# Conclusion



# Conclusion

