R for data vis

Henry Partridge | Blocks | 18 November 2021

What is R?

R is an open source programming language for statistical analysis and data visualisation. It was developed by Ross Ihaka and Robert Gentleman of the University of Auckland and released in 1995. There are now over 18,000 packages available for R which provide functions for machine learning, genomics, time series forecasting, and interactive graphics amongst many others.

R is widely used in academia and by well known companies like Google, Netflix and Airbnb for data analytics. Many graphics published by news outlets like the Financial Times, the Economist and the BBC are generated in R.

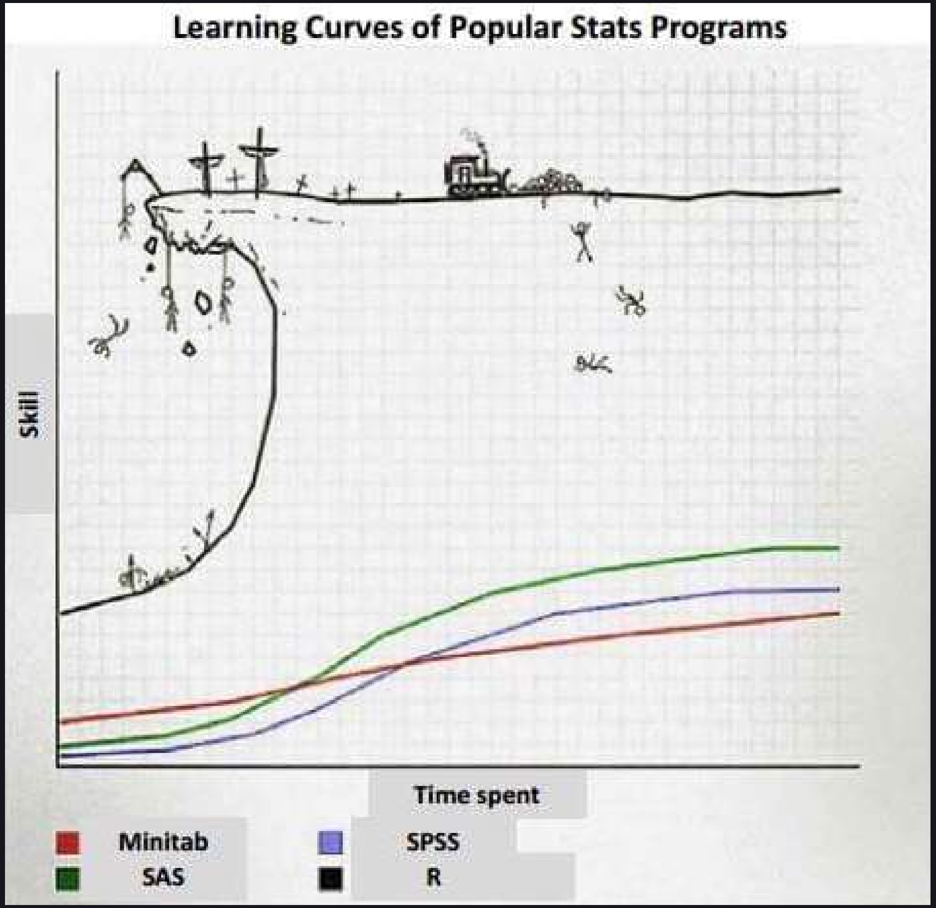
Why use R?

- free
- open source
- advanced statistical analysis
- publication ready graphics

Since R is a language it is also:

- open
- shareable
- reproducibile
- human readable
- diffable

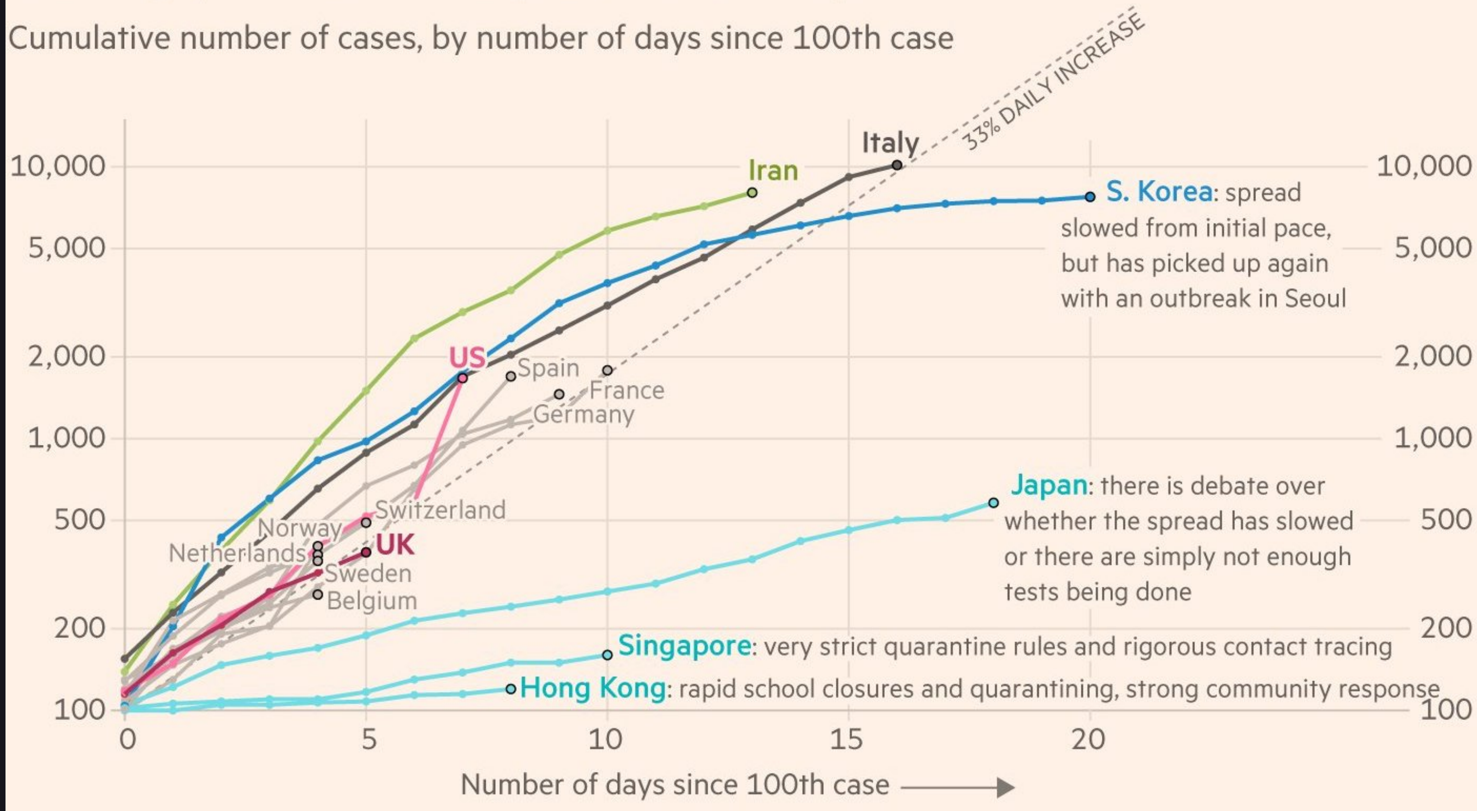
Are there any disadvantages?



Example visualisations

Most western countries are on the same coronavirus trajectory. Hong Kong and Singapore have managed to slow the spread

Cumulative number of cases, by number of days since 100th case



S. Korea: spread slowed from initial pace, but has picked up again with an outbreak in Seoul

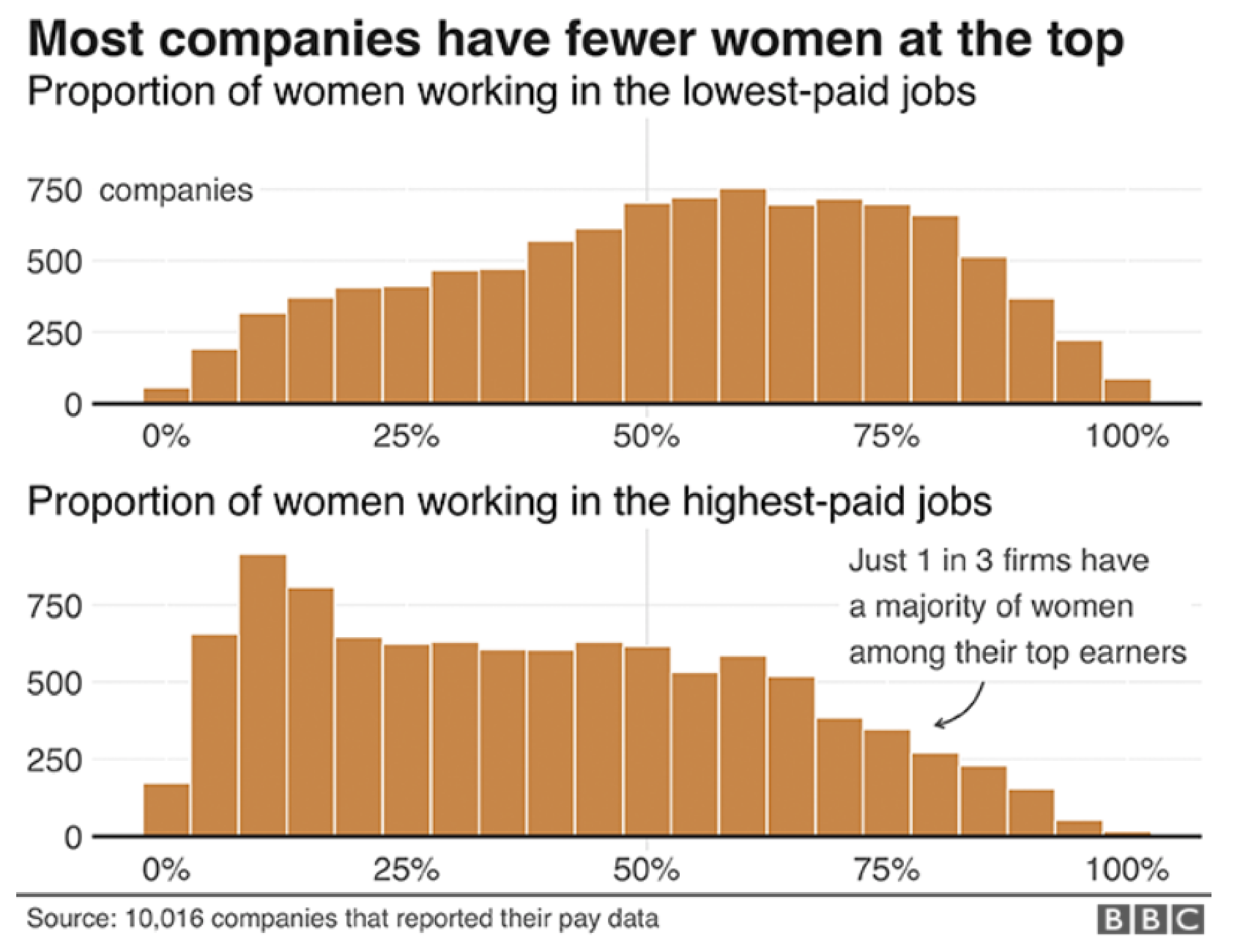
Japan: there is debate over whether the spread has slowed or there are simply not enough tests being done

Singapore: very strict quarantine rules and rigorous contact tracing

Hong Kong: rapid school closures and quarantining, strong community response

Source: FT analysis of Johns Hopkins University, CSSE
FT graphic: John Burn-Murdoch / @jburnmurdoch
© *FT*

John Burn-Murdoch, Financial Times



Most companies have fewer women at the top

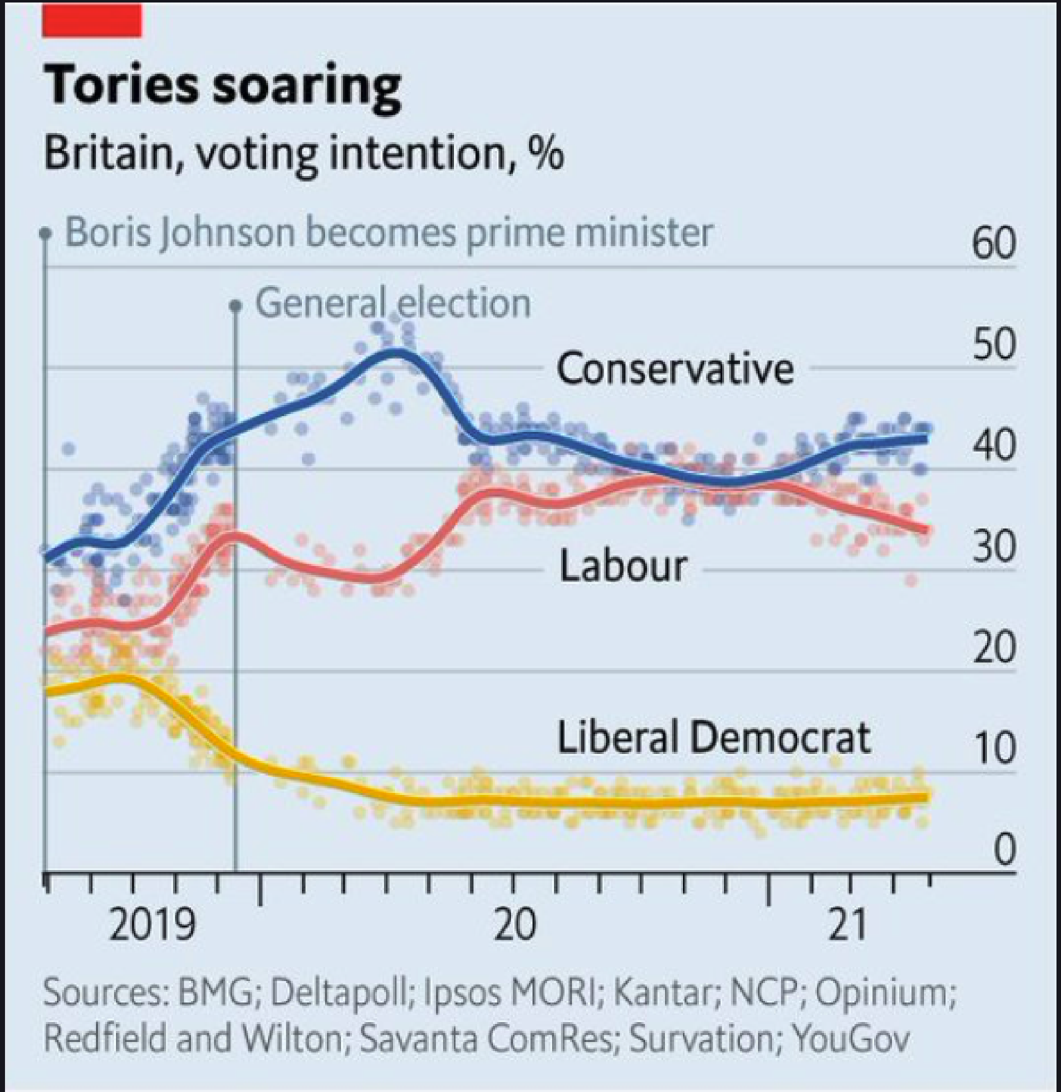
Proportion of women working in the lowest-paid jobs

Proportion of women working in the highest-paid jobs

Just 1 in 3 firms have a majority of women among their top earners

Source: 10,016 companies that reported their pay data

Clara Guibourg, BBC



Helen Atkinson, Economist

RStudio

RStudio

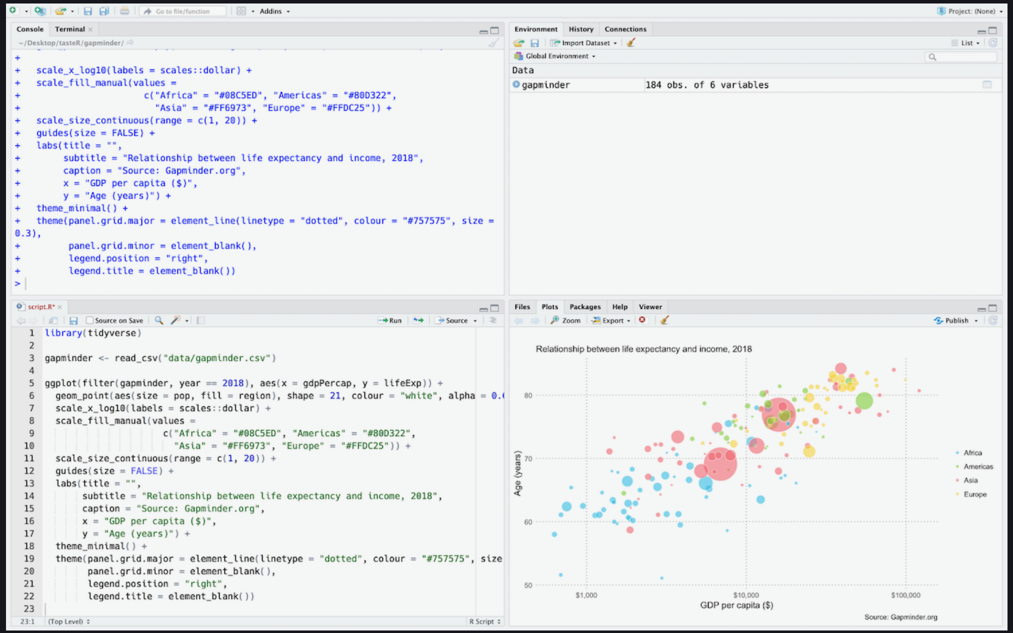
RStudio is an integrated development environment (IDE) for R. It's intuitive interface makes working with R much easier. It supports syntax highlighting, tab completion and is integrated with R Markdown.

RStudio is freely available under the GNU Affero General Public License v3. A commercial desktop license is also available.

The panes

RStudio has four different panes:

- The **Console** pane (top left) is used to execute R commands immediately.
- The **Environment** pane (top right) shows the datasets, models, and plots that are loaded in the current R session. This pane also contains tabs with a scrollable history of executed code, connections to databases and Git options.
- The **Files** pane (bottom right) shows plots and interactive web content, help documentation, previous commands, and R packages that you can install and load.
- The **Source** pane (bottom left) appears when you open a new file e.g. *File -> New File -> R Script*. Code can be saved in dedicated .R scripts and executed in the console with Ctl-Enter/Cmd-Enter. Syntax highlighting and tab completion are also available.



```
+
+   scale_x_log10(labels = scales::dollar) +
+   scale_fill_manual(values =
+                       c("Africa" = "#08C5ED", "Americas" = "#80D322",
+                         "Asia" = "#FF6973", "Europe" = "#FFDC25")) +
+   scale_size_continuous(range = c(1, 20)) +
+   guides(size = FALSE) +
+   labs(title = "",
+        subtitle = "Relationship between life expectancy and income, 2018",
+        caption = "Source: Gapminder.org",
+        x = "GDP per capita ($)",
+        y = "Age (years)") +
+   theme_minimal() +
+   theme(panel.grid.major = element_line(linetype = "dotted", colour = "#757575", size =
0.3),
+         panel.grid.minor = element_blank(),
+         legend.position = "right",
+         legend.title = element_blank())
>
```

Data

gapminder 184 obs. of 6 variables

```
library(tidyverse)

gapminder <- read_csv("data/gapminder.csv")

ggplot(filter(gapminder, year == 2018), aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes(size = pop, fill = region), shape = 21, colour = "white", alpha = 0.
  scale_x_log10(labels = scales::dollar) +
  scale_fill_manual(values =
                      c("Africa" = "#08C5ED", "Americas" = "#80D322",
                        "Asia" = "#FF6973", "Europe" = "#FFDC25")) +
  scale_size_continuous(range = c(1, 20)) +
  guides(size = FALSE) +
  labs(title = "",
       subtitle = "Relationship between life expectancy and income, 2018",
       caption = "Source: Gapminder.org",
       x = "GDP per capita ($)",
       y = "Age (years)") +
  theme_minimal() +
  theme(panel.grid.major = element_line(linetype = "dotted", colour = "#757575", size
         panel.grid.minor = element_blank(),
         legend.position = "right",
         legend.title = element_blank())

```

Setup

Organise your project

Adopting a consistent folder structure for your data analyses will help to ensure that your projects are reproducible. A project can be organised using a simple file structure like this:

```
project/
│
├── data/          # store your datasets
│
├── script.R       # your R script
│
└── output/        # all your plots, models etc
```

Set your working directory

Point your R session to your project folder using: *Session > Set Working Directory > Choose Directory*

NB It's not good practice to set your working directory at the top of your R script because absolute paths don't promote reproducibility.

Optional: Set up a project in RStudio

Open a new R script

File > New File > R Script

Install R packages

You only need to install an R package once. Subsequent package updates can be handled by selecting *Packages > Update* in the **Files** pane of RStudio.

```r
install.packages("tidyverse")
```

Load R packages

Packages need to be loaded at the start of every R session to give you access to the functions you need.

```r
library(tidyverse)
```

Import

Importing data

R can handle a range of data formats: .xlsx, .csv, .txt, .sav, .shp etc. Some data formats require specific packages.

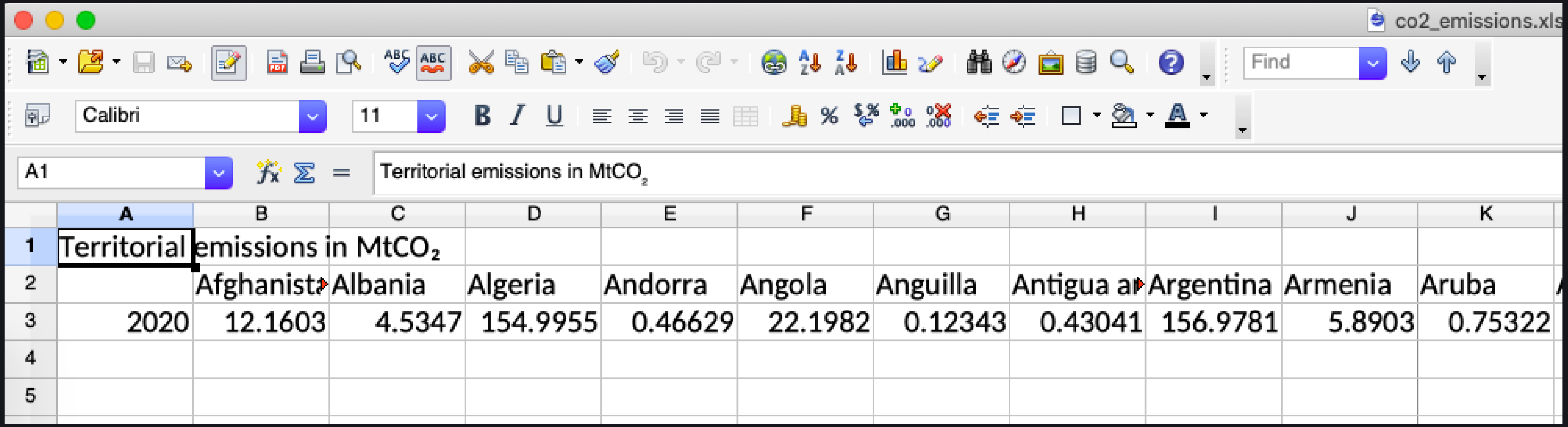
We are going to install and load another package called readxl so that we can import an Excel file.

```r
install.packages("readxl")
library(readxl)
```

Next we'll download and import some CO_2 emissions data collected by the Global Carbon Project.

```r
read_xlsx("data/co2_emissions.xlsx")
```

```
# A tibble: 2 × 222
  `Territorial emi… ...2   ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11
              <dbl> <chr>  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1                NA Afgha… Alba… Alge… Ando… Ango… Angu… Anti… Arge… Arme… Aruba
2              2020 12.16… 4.53… 154.… 0.46… 22.1… 0.12… 0.43… 156.… 5.89… 0.75…
# … with 211 more variables: ...12 <chr>, ...13 <chr>, ...14 <chr>,
#   ...15 <chr>, ...16 <chr>, ...17 <chr>, ...18 <chr>, ...19 <chr>,
#   ...20 <chr>, ...21 <chr>, ...22 <chr>, ...23 <chr>, ...24 <chr>,
#   ...25 <chr>, ...26 <chr>, ...27 <chr>, ...28 <chr>, ...29 <chr>,
#   ...30 <chr>, ...31 <chr>, ...32 <chr>, ...33 <chr>, ...34 <chr>,
#   ...35 <chr>, ...36 <chr>, ...37 <chr>, ...38 <chr>, ...39 <chr>,
#   ...40 <chr>, ...41 <chr>, ...42 <chr>, ...43 <chr>, ...44 <chr>, …
```



A1 Territorial emissions in MtCO$_2$

	A	B	C	D	E	F	G	H	I	J	K
1	Territorial emissions in MtCO$_2$										
2		Afghanist	Albania	Algeria	Andorra	Angola	Anguilla	Antigua ar	Argentina	Armenia	Aruba
3	2020	12.1603	4.5347	154.9955	0.46629	22.1982	0.12343	0.43041	156.9781	5.8903	0.75322
4											
5											

Tidy

Tidying data

```
read_xlsx("data/co2_emissions.xlsx", skip = 1)
  rename(year = 1) %>%
  pivot_longer(-year, names_to = "country", va
  pivot_wider(names_from = country, values_fro
  pivot_longer(-year, names_to = "country", va
```

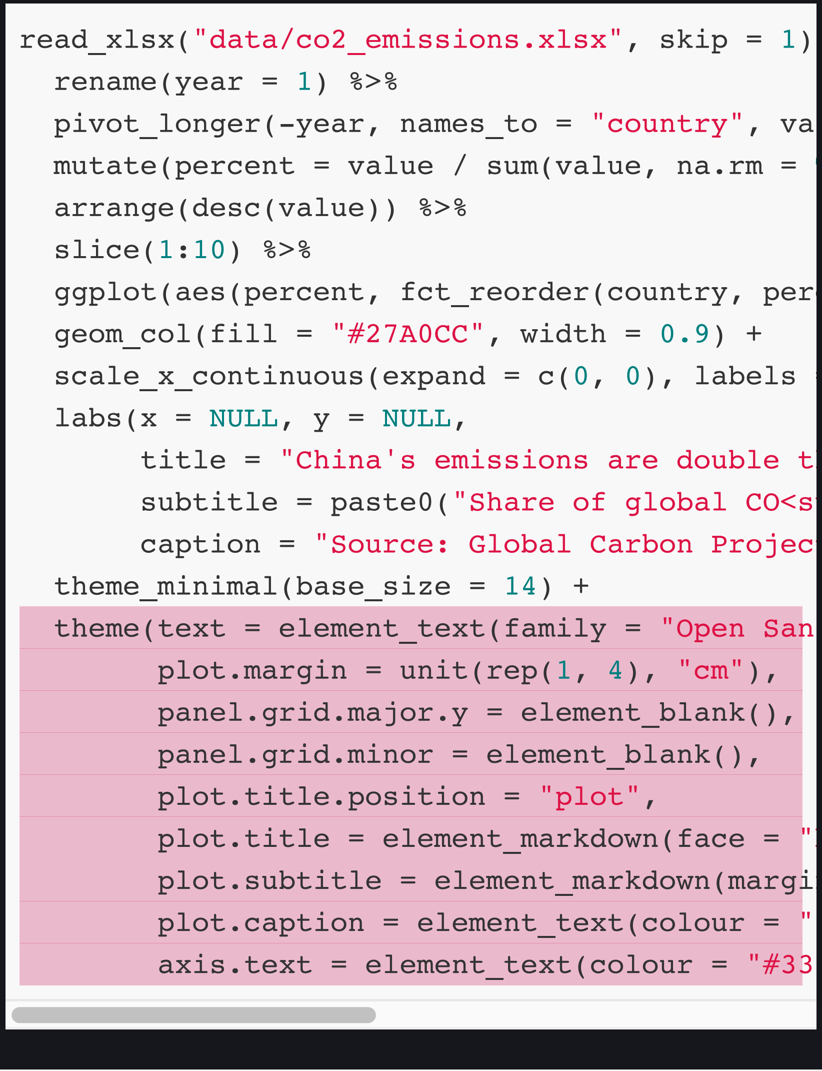
```
# A tibble: 221 × 3
    year country                value
   <dbl> <chr>                  <dbl>
 1  2020 Afghanistan           12.2
 2  2020 Albania                4.53
 3  2020 Algeria              155.
 4  2020 Andorra                0.466
 5  2020 Angola                22.2
 6  2020 Anguilla               0.123
 7  2020 Antigua and Barbuda    0.430
 8  2020 Argentina            157.
 9  2020 Armenia                5.89
10  2020 Aruba                  0.753
# … with 211 more rows
```

Transforming data

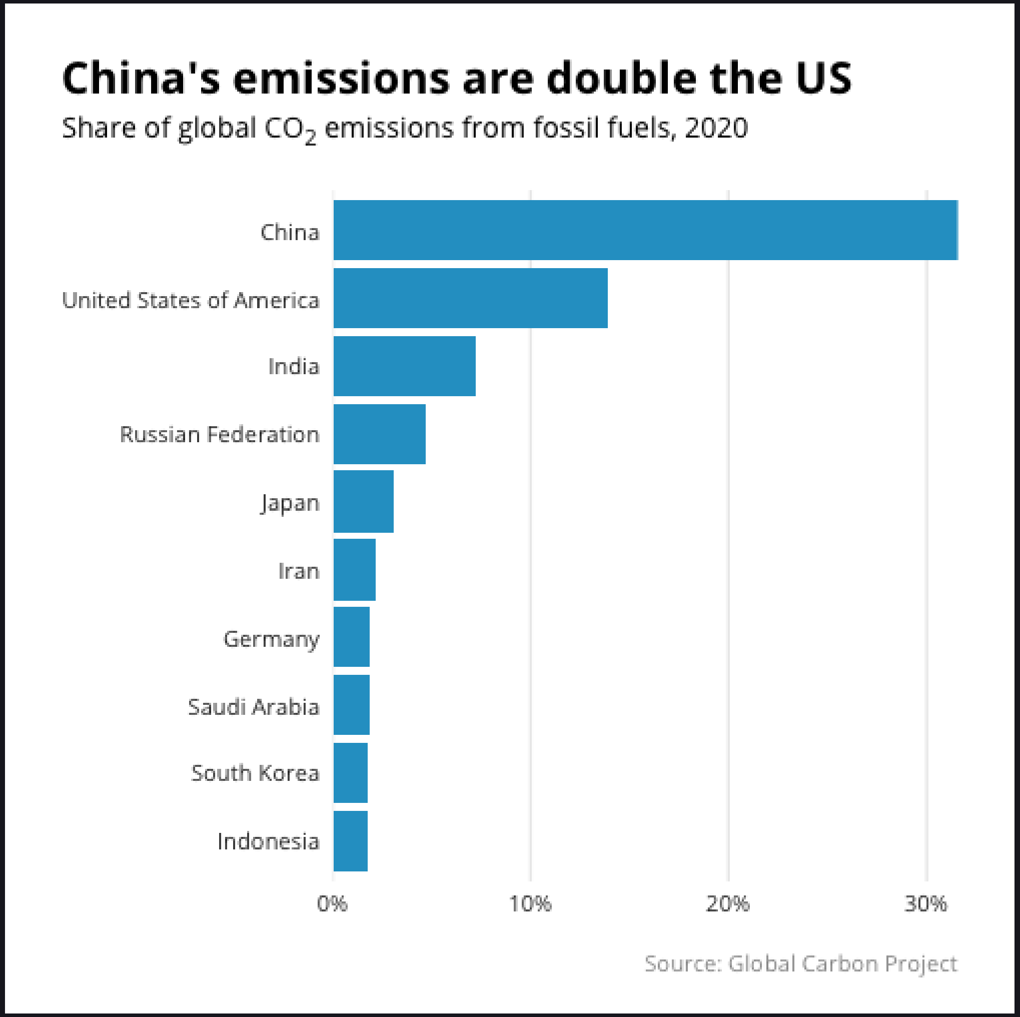
```r
read_xlsx("data/co2_emissions.xlsx", skip = 1)
  rename(year = 1) %>%
  pivot_longer(-year, names_to = "country", va
  mutate(percent = value / sum(value, na.rm =
  arrange(desc(value)) %>%
  slice(1:10)
```

```
# A tibble: 10 × 4
    year country                       value percent
   <dbl> <chr>                         <dbl>   <dbl>
 1  2020 China                        10668.  0.316
 2  2020 United States of America      4713.  0.139
 3  2020 India                         2442.  0.0722
 4  2020 Russian Federation            1577.  0.0467
 5  2020 Japan                         1031.  0.0305
 6  2020 Iran                           745.  0.0220
 7  2020 Germany                        644.  0.0191
 8  2020 Saudi Arabia                   626.  0.0185
 9  2020 South Korea                    598.  0.0177
10  2020 Indonesia                      590.  0.0174
```

Visualising data



```
read_xlsx("data/co2_emissions.xlsx", skip = 1)
  rename(year = 1) %>%
  pivot_longer(-year, names_to = "country", va
  mutate(percent = value / sum(value, na.rm = 
  arrange(desc(value)) %>%
  slice(1:10) %>%
  ggplot(aes(percent, fct_reorder(country, per
  geom_col(fill = "#27A0CC", width = 0.9) +
  scale_x_continuous(expand = c(0, 0), labels 
  labs(x = NULL, y = NULL,
       title = "China's emissions are double t
       subtitle = paste0("Share of global CO<s
       caption = "Source: Global Carbon Projec
  theme_minimal(base_size = 14) +
  theme(text = element_text(family = "Open San
        plot.margin = unit(rep(1, 4), "cm"),
        panel.grid.major.y = element_blank(),
        panel.grid.minor = element_blank(),
        plot.title.position = "plot",
        plot.title = element_markdown(face = "
        plot.subtitle = element_markdown(margi
        plot.caption = element_text(colour = "
        axis.text = element_text(colour = "#33
```



China's emissions are double the US

Share of global CO_2 emissions from fossil fuels, 2020

Source: Global Carbon Project

Further resources

Beginners

- RStudio primers
- R for Data Science by Hadley Wickham and Garrett Grolemund

Data visualisation

- Fundamentals of Data Visualization by Claus Wilke
- Data Visualization: A practical introduction by Kieran Healy
- SDS 375: Data Visualization in R
- BBC Visual and Data Journalism cookbook for R graphics

Statistics

- Discovering Statistics Using R by Andy Field
- Statistics: An Introduction Using R by Michael J. Crawley

Help

- StackOverflow
- RStudio Community
- Twitter #rstats hashtag