Introduction to R

Lecture 1: Getting (slowly) Started

Swapnil Singh
Lietuvos Bankas | Course Link

Lecture's Objectives

1. Why this course?
2. Getting started
3. What R can do?
4. Some visualizations for motivation

Why this course?

What is R?

- R is a language and environment for statistical computing and graphics.
- **Open-source** and extensible.
- Built upon *S*, a statistical programming language.
- Rich set of packages for data manipulation (e.g., `tidyverse` environment build upon `tidyr`, `dplyr`, etc.).
- Comprehensive plotting libraries (e.g., `ggplot2`, `lattice`).
- Wide array of statistical tests.
- Facilitates machine learning algorithms.
- Excellent packages (e.g., `sf`, `terra`, etc.) for spatial datasets and GIS applications

Comparison with other languages

R vs Python

- R: More focused on statistical modeling and data visualization.
- R: Weaker on web-scraping
- Python: More general-purpose but has strong data science libraries (e.g., `pandas`, `scikit-learn`).

R vs MATLAB

- R: Open-source and has a larger community for data science.
- MATLAB: Stronger in numerical simulation but less versatile for data manipulation.

R vs STATA

Flexibility and Extensibility

- **R**: Highly extensible through packages; good for custom statistical methods.
- **STATA**: More rigid but user-friendly for standard statistical tests.

Data Handling

- **R**: More versatile data manipulation capabilities (`dplyr`, `tidyr`).
- **STATA**: Efficient for large datasets but less flexibility.

Graphics and Visualization

- **R**: Advanced graphical capabilities (`ggplot2`).
- **STATA**: Basic graphs are easier to produce but less customizable.

Pricing and Community

- **R**: Open-source, large and active community.
- **STATA**: Commercial software, smaller community focused on social sciences.

So, Why This Course?

- Provide a basic introduction to R
- In future, extend the course to advance level (*depending upon demand*)
  - Causal inference methods
  - Big data (`vroom`)
  - Webscraping
  - Geospatial analysis
  - ...
- Many of these topics are not directly relevant for your specific work, but,

> **Knowledge is power!**

Getting Started

What you need?

For this course:

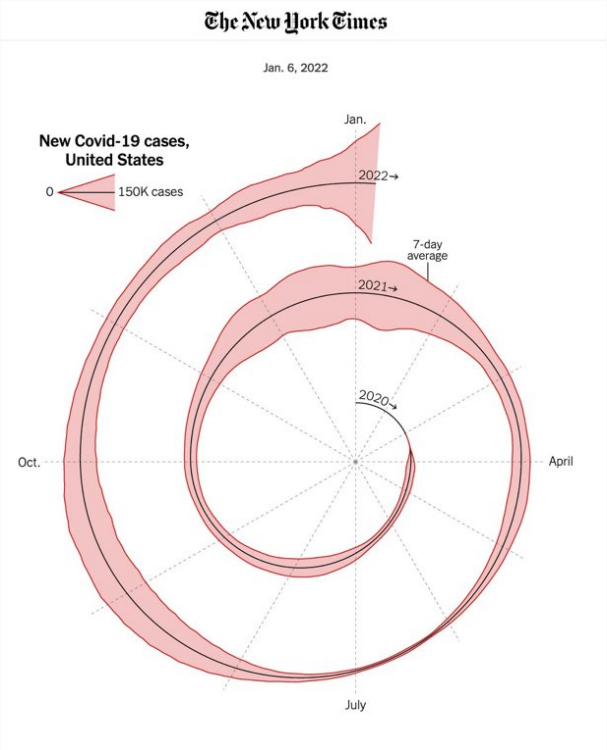
- Install `R`
- Install `RStudio`

Additional components

- Install `Git`
- Create a `Github` account. Alternatively, you can also create account on `Gitlab` or `Gitbucket`, but `Github` is the most used one
- If working only with `RStudio`, follow this website to get things working: http://happygitwithr.com
- For general purpose use of `Git`, I will suggest to use `GitKraken`

What R Can Do?

Let's create some figures!



NY Times, 2022-01-06, Covid Cases Spiral, US

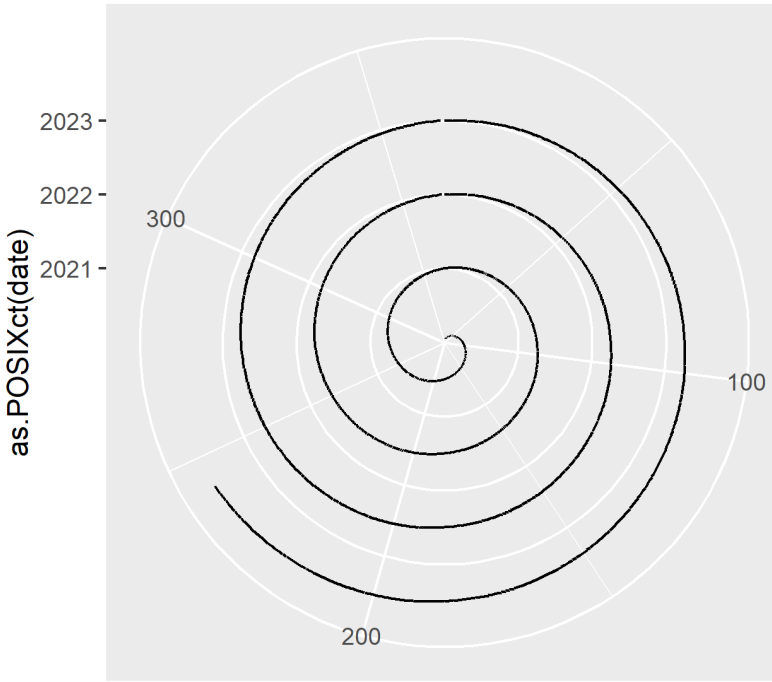
Let's create some figures!

Hats off to ByData blog!

```
pacman::p_load("tidyverse", "ggtext", "here", "lubridate")
owid_url <- "https://github.com/owid/covid-19-data/blob/master/public/data/owid-covid-
covid <- suppressMessages(read_csv(owid_url))

# get the data only for the US
covid_cases <- covid %>%
  dplyr::filter(location == 'United States') %>%
  dplyr::select(date, new_cases, new_cases_smoothed) %>%
  dplyr::arrange(date) %>%


  # add some additional data to complete the year
  dplyr::add_row(date = as_date("2020-01-01"), new_cases = 0, new_cases_smoothed = 0,
          .before = 1) %>%
  tidyr::complete(date = seq(min(.$date), max(.$date), by = 1),
           fill = list(new_cases = 0, new_cases_smoothed = 0)) %>%
  dplyr::mutate(day_of_year = yday(date),
         year = year(date)
         )
```

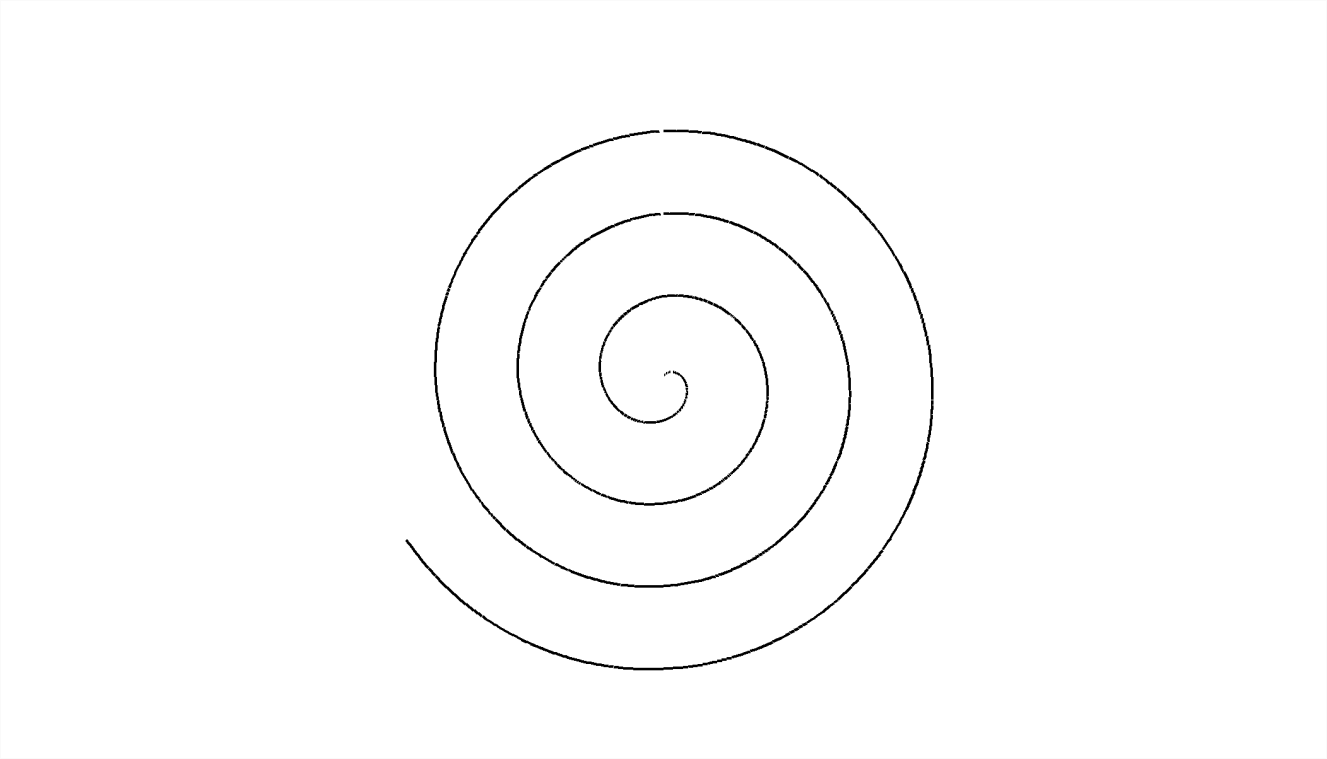
How a basic plot looks like?

```
basic.plot ← covid_cases ▷
  ggplot() +
  geom_segment(aes(x = day_of_year, xend = day_of_year + 1,
                   y = as.POSIXct(date), yend = as.POSIXct(date))) +
  coord_polar()
basic.plot
```



How a basic plot looks like?

```
basic.plot + theme_void()
```



Moving onto advanced level now!

Moving onto advanced level now!

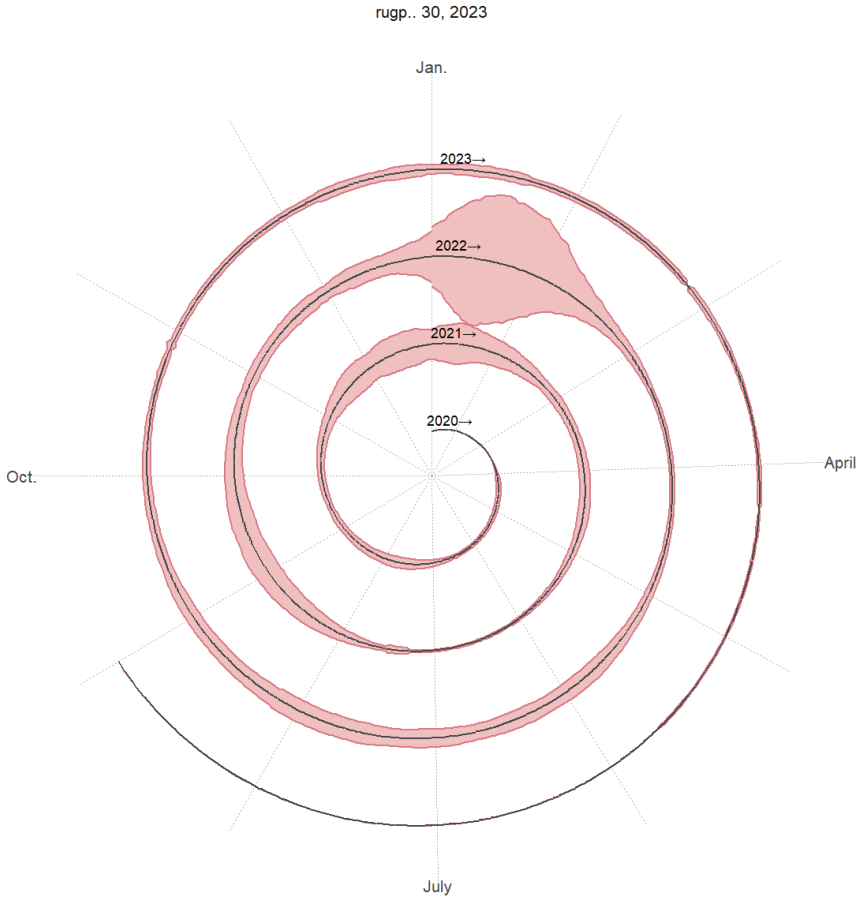
```
month_length ← c(31, 28, 31, 30, 31, 30,
                 31, 31, 30, 31, 30, 31)

month_breaks ← cumsum(month_length) - 30

basic.plot ←  basic.plot + scale_x_continuous(minor_breaks = month_breaks,
                     breaks = month_breaks[c(1, 4, 7, 10)],
                     labels = c("Jan.", "April", "July", "Oct.")) +
  theme(
    plot.background = element_rect(color = NA, fill = "white"),
    panel.grid.major.x = element_line(color = "grey70", size = 0.2, linetype = "dottec
    panel.grid.minor.x = element_line(color = "grey70", size = 0.2, linetype = "dottec
    axis.text.x = element_text(color = base_grey, size = 5, hjust = 0.5),
  )
```

Moving onto advanced level now!

- One truth about generating good figures: last 10 percent improvement takes 90 percent of time

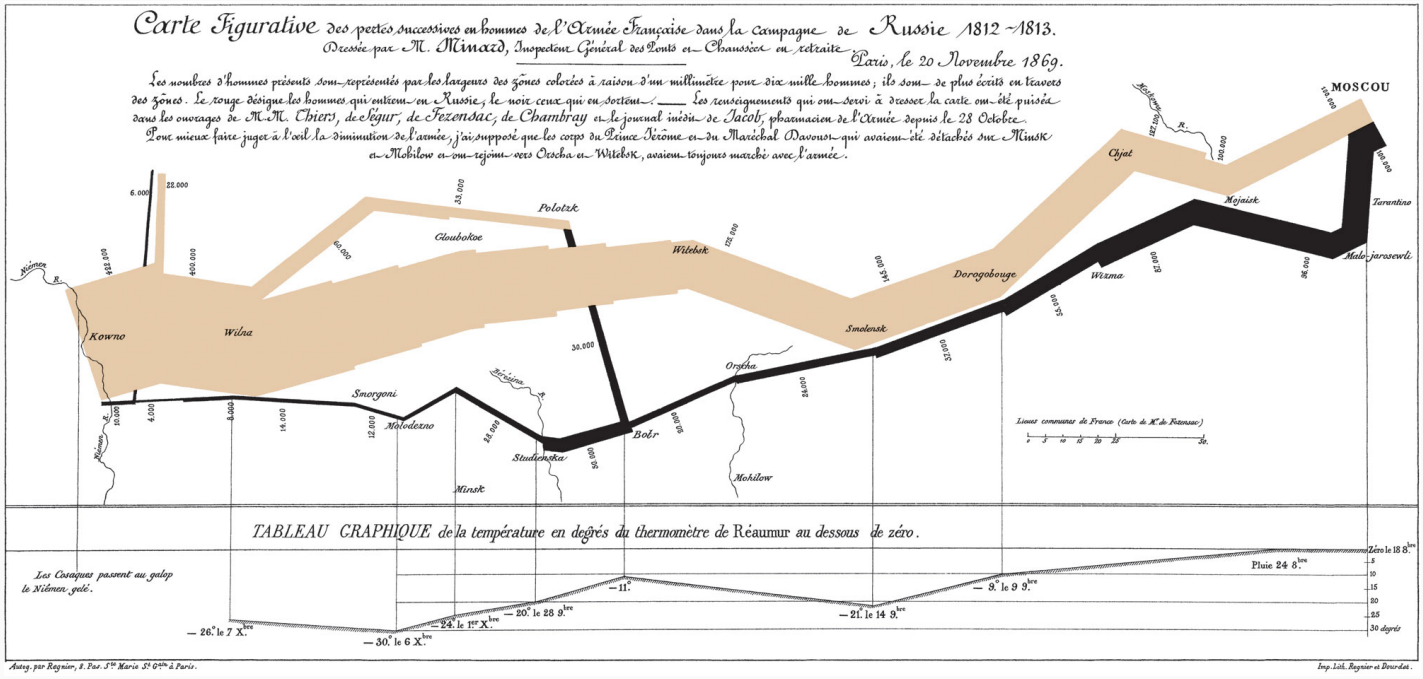
```
month_length ← c(31, 28, 31, 30, 31, 30,
                 31, 31, 30, 31, 30, 31)

month_breaks ← cumsum(month_length) - 30

basic.plot ←   basic.plot + scale_x_continuous(minor_breaks = month_breaks,
                    breaks = month_breaks[c(1, 4, 7, 10)],
                    labels = c("Jan.", "April", "July", "Oct.")) +
  theme(
    plot.background = element_rect(color = NA, fill = "white"),
    panel.grid.major.x = element_line(color = "grey70", size = 0.2, linetype = "dottec
    panel.grid.minor.x = element_line(color = "grey70", size = 0.2, linetype = "dottec
    axis.text.x = element_text(color = base_grey, size = 5, hjust = 0.5),
  )
```

Advanced level now



Minard's 1812 Plot

- Hats off to Andrew Heiss



Forward and Retreat path of Napoleon's Army