SEDSI 2022

Tobin Turner

2022-02-17

Contents

Chapter 1

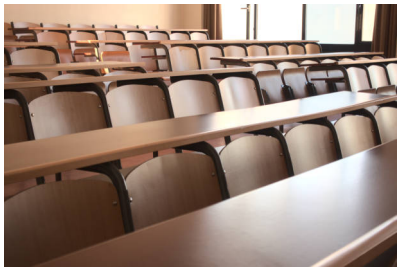
SEDSI 2022

SEDSI Fun with Reproducible Analysis and RMarkdown

Chapter 2

Motivation

A COVID Classroom



A Learning Management System Nightmare

Concise, Precisely Organized, Frequently Revised Assignments and Schedules

Date	Topic
Wednesday, February 16, 2022	SEDSI in Jacksonville
Thursday, February 17, 2022	Present at 2:45 PM
Friday, February 18, 2022	Celebrate a successful DASI Session

Chapter 3

Real life example

It's nice to know exactly what you did when your original data requires wrangling.

Conflicts and students honors...

Table 3.1: Some Actual Data We Considered

NAME	TOTAL.HOURS	PC.HOURS	ADMIT.TERM
Greer, Patrick Sterling	3.0	3.0	201101
Greer, Patrick Sterling	144.0	123.0	201101
Thompson, Charleston Hannah	0.0	0.0	201201
Thompson, Charleston Hannah	142.0	122.0	201201
Melvin, Victor Richard-Scorsese	132.0	100.0	201202
Roberson, States Taylor	126.0	99.0	201301
Allen, Kaylee Michelle	125.0	68.0	201601
Phelps, Payton Elliott	117.0	114.0	201701
Rowley, Ella Marie Dorothy	121.0	121.0	201701
Smith, Michael Leston	112.0	112.0	201701
Taylor, Darrell Tyrese	78.0	78.0	201701
Wright, Alexandra Ruby	116.0	116.0	201701
Adu, Tyler	80.0	80.0	201801
Armell, James Richard	90.0	87.0	201801
Bell, Carrie Abigail	120.5	99.5	201801
Boyd, Jeremiah Quintin	87.0	87.0	201801
Brinkley, Khalid Osmon	74.0	74.0	201801
Campbell, Blakeney Herlong	92.0	85.0	201801
Dearman, Clark Avant	101.5	82.5	201801
Drake, John Chapman	94.0	94.0	201801

Chapter 4

Some Options

This is just a cool place to put stuff[1].

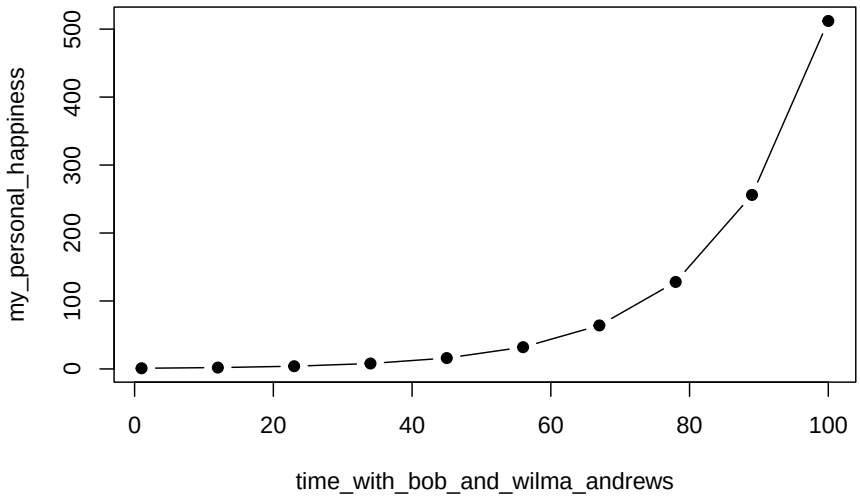
Like a schedule, for example:

4.1 Spring 2022

Date	Topic
Monday, January 10, 2022	R basics and install
Wednesday, January 12, 2022	R basics and workflows
Friday, January 14, 2022	QUIZ 1
Monday, January 17, 2022	MLK Holiday
Wednesday, January 19, 2022	Objects, Vectors, and Arithmetic
Friday, January 21, 2022	QUIZ 2
Monday, January 24, 2022	Summaries and Subscripting

[1] Footnotes are always neat. And useful. Like this one!

4.2 Or a figure



4.3 Or an Equation

Here is a **fun** equation for my SEDSI DASI friends:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{4.1}$$

Table 4.2: A table of the first 10 rows of the mtcars data.

	mpg	cyl	disp	hp	drat	wt	qsec	vs
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1

4.4 Or a table of something

4.4.1 Fun example table

4.5 Or an Image

4.5.1 Hero 1

4.5.2 Hero 2

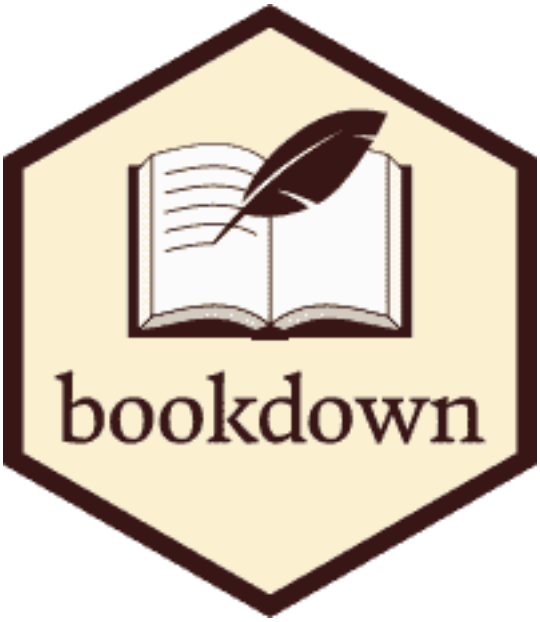
Chapter 5

Workflow Summary

5.1 R (engine) and Rstudio (IDE)

5.2 RMarkdown

5.3 bookdown package



5.4 github

5.5 netlify

Chapter 6

Lab 3: `coronavirus` visualization, data wrangling, and dates

6.1 Overview

The package is available on GitHub here and is updated daily.

I use the `coronavirus` package and use the `coronavirus::update_data()` function to keep the data current. This also has the dates preformatted which can be nice.

6.2 Let's look like Applied Analytics Superstars and make some neat visuals.

```
coronavirus::update_dataset()
#> Rows: 633609 Columns: 15
#> -- Column specification --------------------------------------
#> Delimiter: ","
#> chr  (8): province, country, type, iso2, iso3, combined_...
#> dbl  (6): lat, long, cases, uid, code3, population
#> date (1): date
#>
#> i Use `spec()` to retrieve the full column specification for this data.
```

```
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
#> No updates are available
```

```
library(coronavirus)
library(dplyr)
library(ggplot2)
```

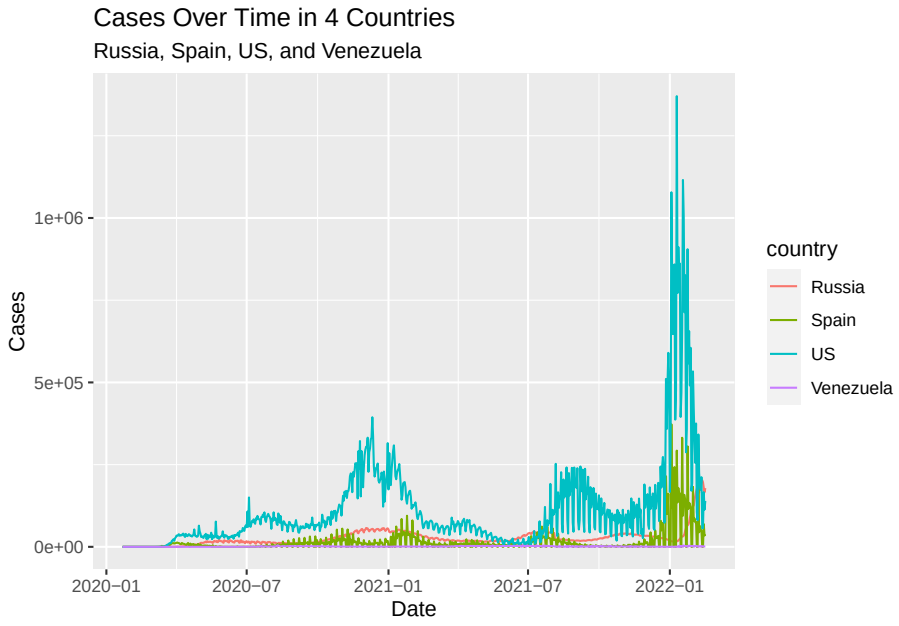
I'd recommend you always start by trying to understand a bit about the data.

```
head(coronavirus)
#>         date province country     lat      long      type
#> 1 2020-01-22  Alberta  Canada 53.9333 -116.5765 confirmed
#> 2 2020-01-23  Alberta  Canada 53.9333 -116.5765 confirmed
#> 3 2020-01-24  Alberta  Canada 53.9333 -116.5765 confirmed
#> 4 2020-01-25  Alberta  Canada 53.9333 -116.5765 confirmed
#> 5 2020-01-26  Alberta  Canada 53.9333 -116.5765 confirmed
#> 6 2020-01-27  Alberta  Canada 53.9333 -116.5765 confirmed
#>   cases   uid iso2 iso3 code3    combined_key population
#> 1     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 2     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 3     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 4     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 5     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 6     0 12401   CA  CAN   124 Alberta, Canada    4413146
#>   continent_name continent_code
#> 1  North America             NA
#> 2  North America             NA
#> 3  North America             NA
#> 4  North America             NA
#> 5  North America             NA
#> 6  North America             NA
```

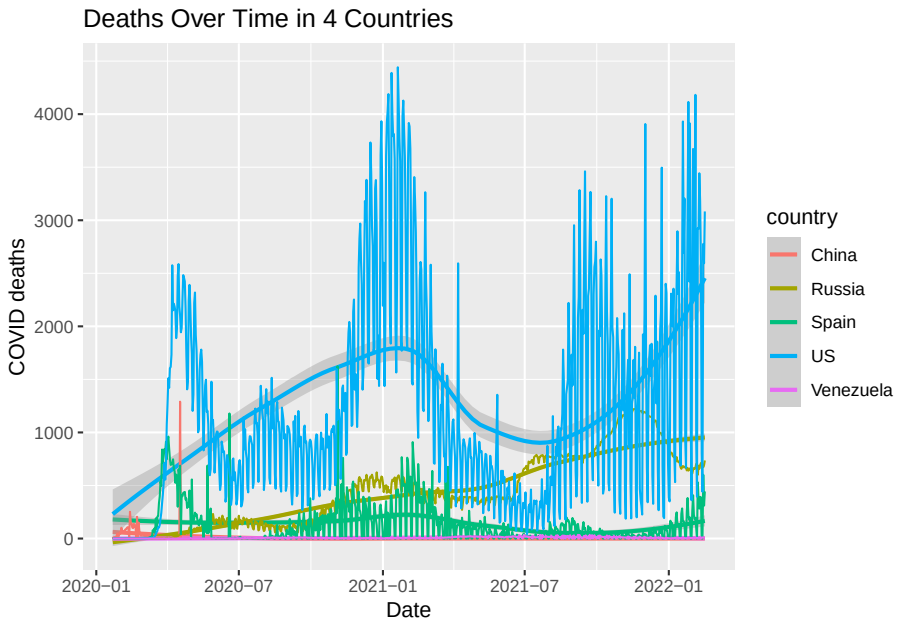
For example, what does this summary let us know?

```
summary(coronavirus$cases)
#>      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#> -30974748         0         0       669        29   1368563
```

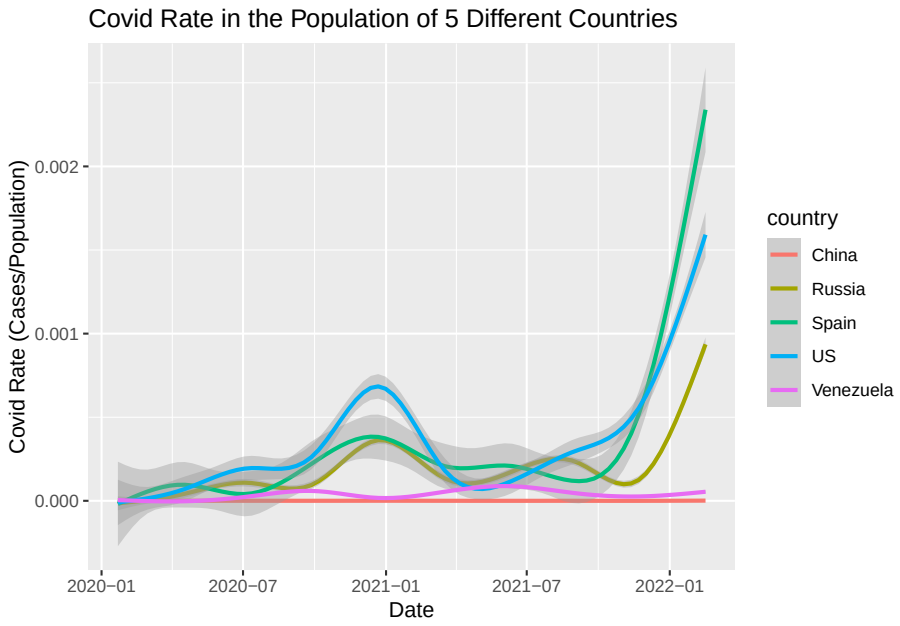
1. Can you create a visual showing the cases over time for Russia, Spain, US, and Venezuela? Also, why might `filter(cases >= 0)` be worth using?



2. Can you show deaths over time for Russia, Spain, US, and Venezuela? And can you play with your geoms and make something neat?

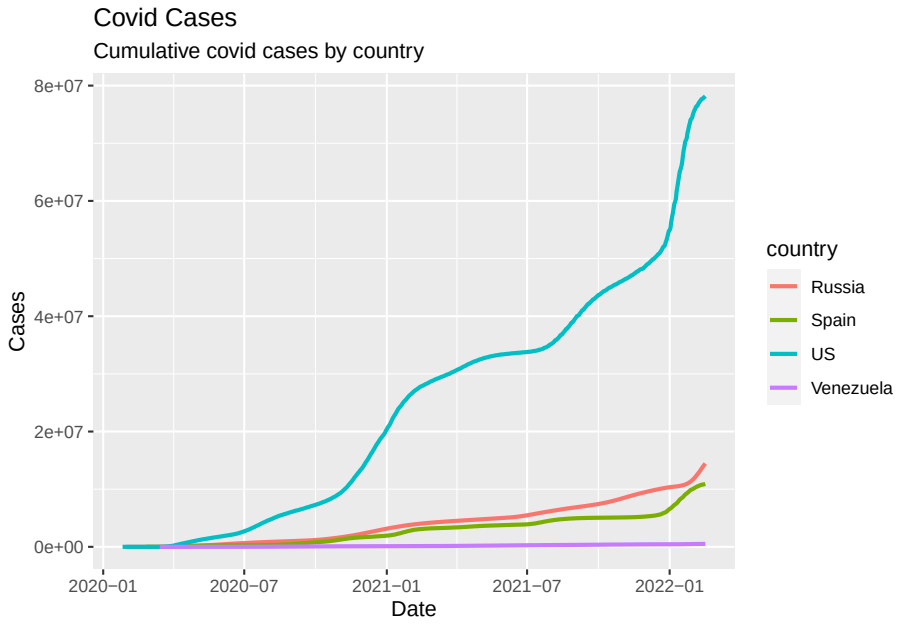


3. Now let's do a plot of COVID rate (# confirmed cases / population). Something like this.



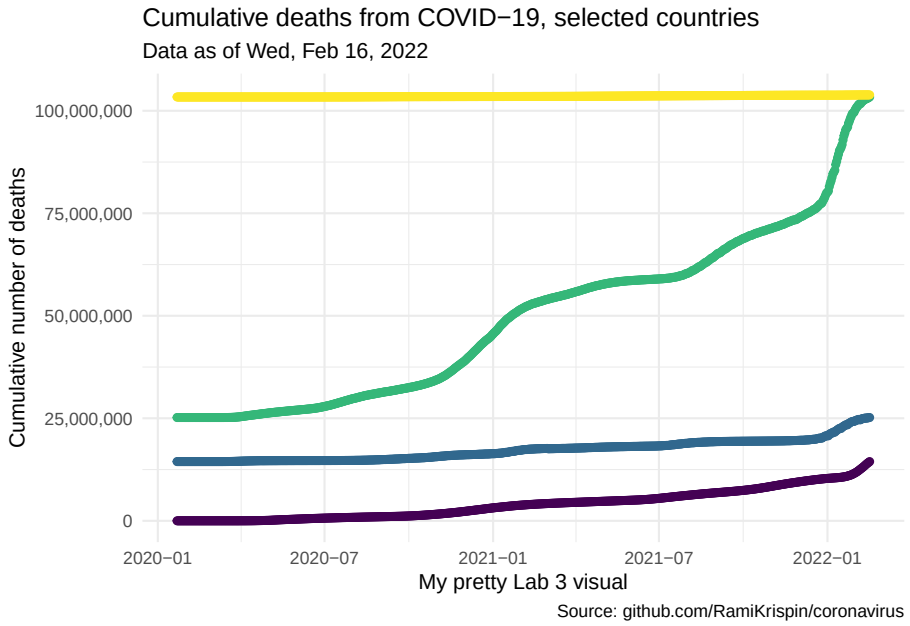
4. What is and **is not** useful about the previous illustration?

5. Make a chart with cumulative cases. Something like this:



6. With a little more time and a few extra packages, we **could** make a graph prettier. Try.

```
library(scales)
library(ggrepel)
library(glue)
library(lubridate)
```



7. Now let's **really** have some fun. Let's illustrate death rates relative to confirmed cases. Why is this more challenging than anything we've done so far in this lab? We're going to have to make this data **tidy**.

One way to play this game.

Let's make a little table of just date, country, and deaths (with a meaningful variable name), and then count observations by coutry just to make sure eveything looks nice.

```
#>         date country deaths
#> 1 2020-01-22  Russia      0
#> 2 2020-01-23  Russia      0
#> 3 2020-01-24  Russia      0
#> 4 2020-01-25  Russia      0
#> 5 2020-01-26  Russia      0
#> 6 2020-01-27  Russia      0
#>     country   n
#> 1    Russia 757
#> 2     Spain 754
#> 3        US 757
#> 4 Venezuela 756
```

Let's make a little table of just confirmed cases.

```
#>         date country confirmed
#> 1 2020-01-22  Russia         0
#> 2 2020-01-23  Russia         0
#> 3 2020-01-24  Russia         0
#> 4 2020-01-25  Russia         0
#> 5 2020-01-26  Russia         0
#> 6 2020-01-27  Russia         0
#>     country   n
#> 1    Russia 757
#> 2     Spain 757
#> 3        US 757
#> 4 Venezuela 757
```

Let's join these together. I use `left_join`.

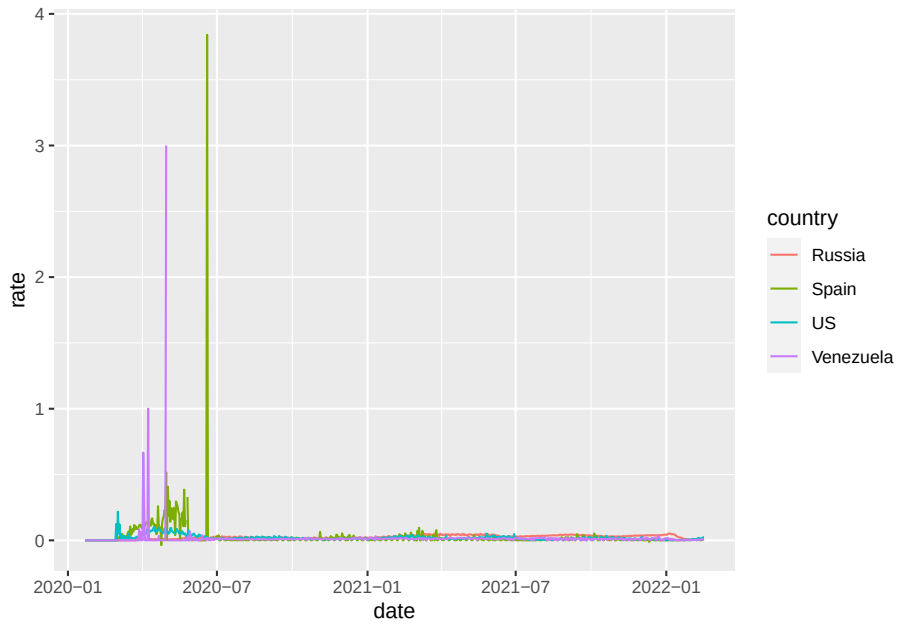
```
#>         date country deaths confirmed
#> 1 2020-01-22  Russia      0         0
#> 2 2020-01-23  Russia      0         0
#> 3 2020-01-24  Russia      0         0
#> 4 2020-01-25  Russia      0         0
#> 5 2020-01-26  Russia      0         0
#> 6 2020-01-27  Russia      0         0
#>     country   n
#> 1    Russia 757
#> 2     Spain 757
#> 3        US 757
#> 4 Venezuela 757
```

Let's add some cumulative statistics as well.

```
#>         date country deaths confirmed cumulative_cases
#> 1 2020-01-22  Russia      0         0                0
#> 2 2020-01-23  Russia      0         0                0
#> 3 2020-01-24  Russia      0         0                0
#> 4 2020-01-25  Russia      0         0                0
#> 5 2020-01-26  Russia      0         0                0
#> 6 2020-01-27  Russia      0         0                0
#>   cumulative_deaths rate
#> 1                 0    0
#> 2                 0    0
#> 3                 0    0
#> 4                 0    0
#> 5                 0    0
#> 6                 0    0
```

Now we can plot some more fun stuff.

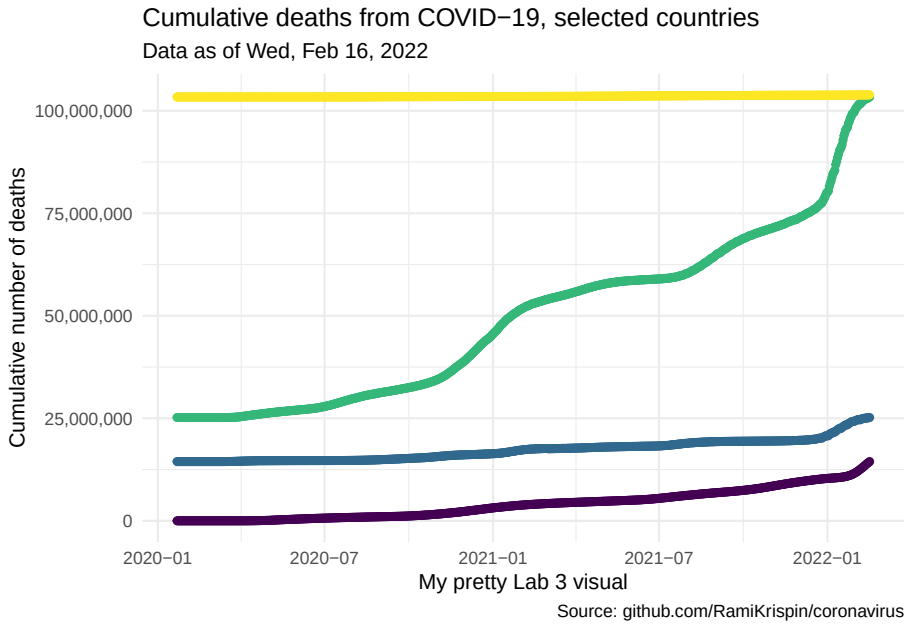
```
ggplot(data = df3,
       mapping = aes(x = date,
                     y = rate,
                     color = country)) +
  geom_line()
```




```
summary(df3)
#>       date               country              deaths
#>  Min.   :2020-01-22   Length:3028        Min.   :   0.0
#>  1st Qu.:2020-07-29   Class :character   1st Qu.:   5.0
#>  Median :2021-02-03   Mode  :character   Median : 126.0
#>  Mean   :2021-02-03                      Mean   : 452.7
#>  3rd Qu.:2021-08-11                      3rd Qu.: 639.2
#>  Max.   :2022-02-16                      Max.   :4442.0
#>                                          NA's   :4
#>    confirmed         cumulative_cases    cumulative_deaths
#>  Min.   : -74937.0   Min.   :        0   Min.   :     0
#>  1st Qu.:    461.5   1st Qu.: 14445698   1st Qu.: 16977
#>  Median :   7814.0   Median : 25190092   Median : 99431
#>  Mean   :  34303.5   Mean   : 43523860   Mean   :135068
#>  3rd Qu.:  28170.0   3rd Qu.:103362932   3rd Qu.:248203
```

```
#>  Max.   :1368563.0   Max.   :103870974   Max.   :364273
#>                                          NA's   :2147
#>       rate
#>  Min.   :-0.036576
#>  1st Qu.: 0.004568
#>  Median : 0.012750
#>  Mean   : 0.021680
#>  3rd Qu.: 0.023227
#>  Max.   : 3.840391
#>  NA's   :4
library(scales)
library(ggrepel)
library(glue)
library(lubridate)
as_of_date <- df3 %>%
  summarise(max(date)) %>%
  pull()
as_of_date_formatted <- glue("{wday(as_of_date, label = TRUE)}, {month(as_of_date, label = TRUE)}

ggplot(data = df3,
       mapping = aes(x = date,
                     y = cumulative_cases,
                     color = country)) +
  # represent cumulative cases with lines
  geom_line(size = 0.7, alpha = 0.8) +
  # add points to line endings
  geom_point() +
  # add country labels, nudged above the lines
  # geom_label_repel(nudge_y = 1, direction = "y", hjust = 1) +
  # turn off legend
  guides(color = FALSE) +
  # use pretty colors
  scale_color_viridis_d() +
  # better formatting for y-axis
  scale_y_continuous(labels = label_comma()) +
  # use minimal theme
  theme_minimal() +
  # customize labels
  labs(
    x = "My pretty Lab 3 visual",
    y = "Cumulative number of deaths",
    title = "Cumulative deaths from COVID-19, selected countries",
    subtitle = glue("Data as of", as_of_date_formatted, .sep = " "),
    caption = "Source: github.com/RamiKrispin/coronavirus"
  )
```


Cumulative deaths from COVID−19, selected countries
Data as of Wed, Feb 16, 2022
Cumulative number of deaths
100,000,000
75,000,000
50,000,000
25,000,000
0
2020−01
2020−07
2021−01
2021−07
2022−01
My pretty Lab 3 visual
Source: github.com/RamiKrispin/coronavirus

Chapter 7

Thoughts? Questions? Discussion?

7.1 Thank you for your time!