

SEDSI 2022

Tobin Turner

2022-02-15

Contents

1	SEDSI 2022	5
2	Motivation	7
3	Real life example	9
4	Some Options	11
4.1	Spring 2022	11
4.2	Or a figure	12
4.3	Or an Image	13
4.4	Or an Equation	14
4.5	Or a table of something	14
5	Workflow Summary	15
5.1	R (engine) and Rstudio (IDE)	15
5.2	bookdown package	16
5.3	github	16
5.4	netlify	17
6	Lab 3: coronavirus visualization, data wrangling, and dates	19
6.1	Let's look like Applied Analytics Superstars and make some neat visuals.	19

Chapter 1

SEDSI 2022

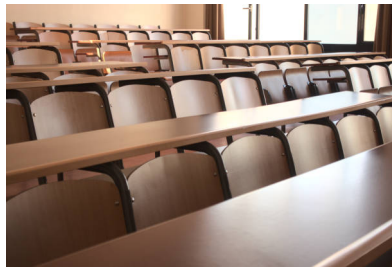


Reproducible Data and RMarkdown Tobin Turner

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

Show the student data as an example of reproducible research->
then segue to teacher from reproducible research focus.

Chapter 4

Some Options

This is just a cool place to put stuff¹.

Like a schedule, for example:

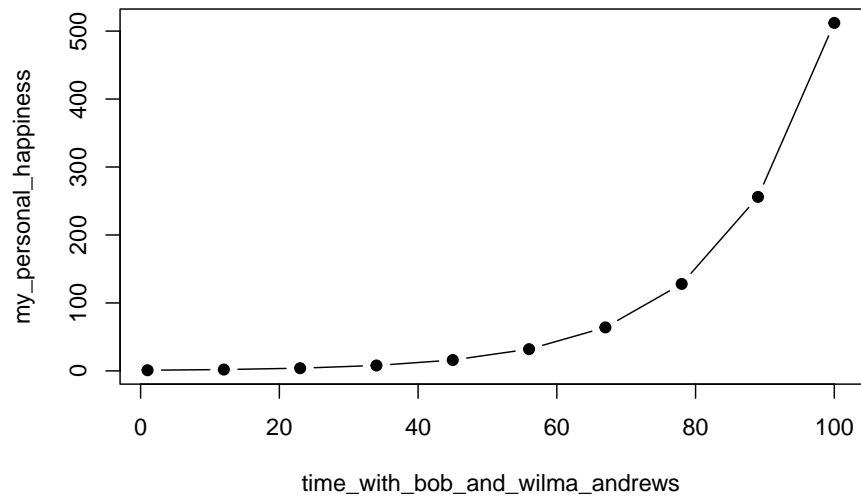
Footnotes are put inside the square brackets after a caret ². Like this one.¹

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

¹Footnotes are always neat. And useful. Like this one!
²

4.2 Or a figure



4.3 Or an Image

4.3.1 Hero 1



4.3.2 Hero 2



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

	mpg	cyl	displacement	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 an Equation

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

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

4.5 Or a table of something

4.5.1 Fun example table

Chapter 5

Workflow Summary

5.1 R (engine) and Rstudio (IDE)



5.2 bookdown package



5.3 github



5.4 netlify



netlify

Chapter 6

Lab 3: coronavirus visualization, data wrangling, and dates

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.

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

```
coronavirus::update_dataset()
#> Rows: 627405 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.
#> Updates are available on the coronavirus Dev version, do you want to update? n/Y
```

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

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

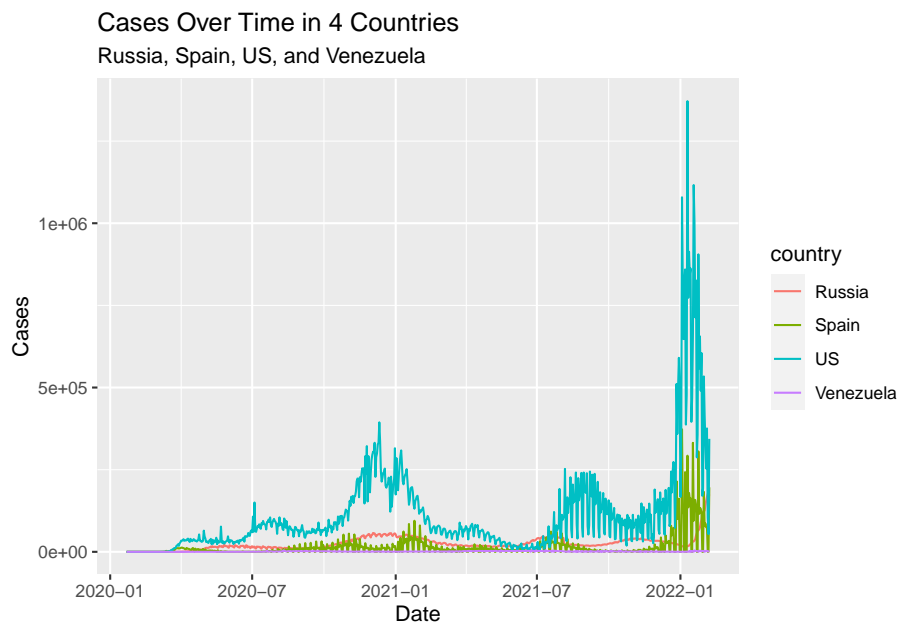
```
head(coronavirus)
#>      date province country    lat    long    type
#> 1 2020-01-22  Alberta  Canada 53.9333 -116.5765 confirmed
#> 2 2020-01-23  Alberta  Canada 53.9333 -116.5765 confirmed
#> 3 2020-01-24  Alberta  Canada 53.9333 -116.5765 confirmed
#> 4 2020-01-25  Alberta  Canada 53.9333 -116.5765 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?

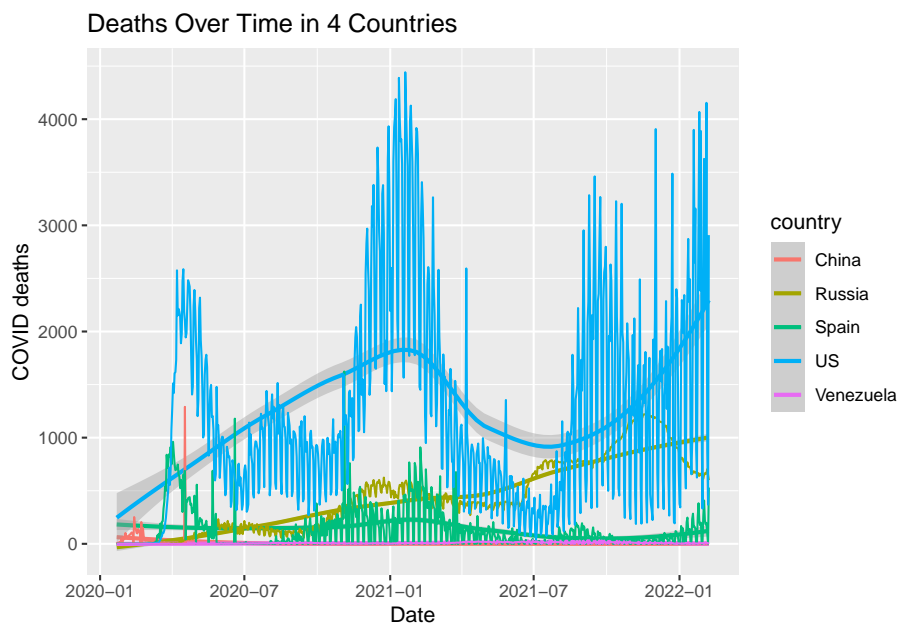
```
summary(coronavirus$cases)
#>      Min.    1st Qu.    Median      Mean    3rd Qu.      Max.
#> -30974748         0         0        651         30   1369637
```

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?

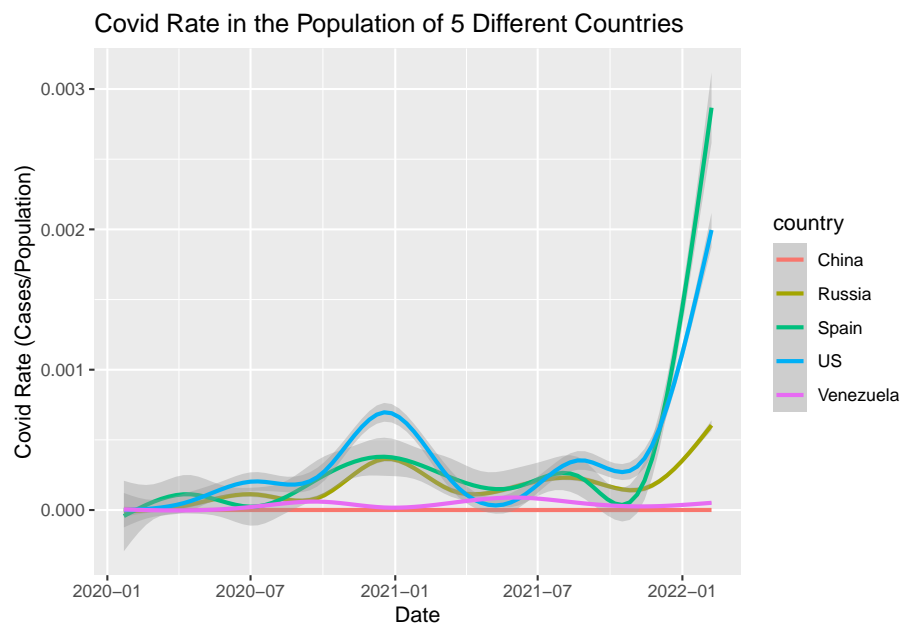
6.1. LET'S LOOK LIKE APPLIED ANALYTICS SUPERSTARS AND MAKE SOME NEAT VISUALS.21



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



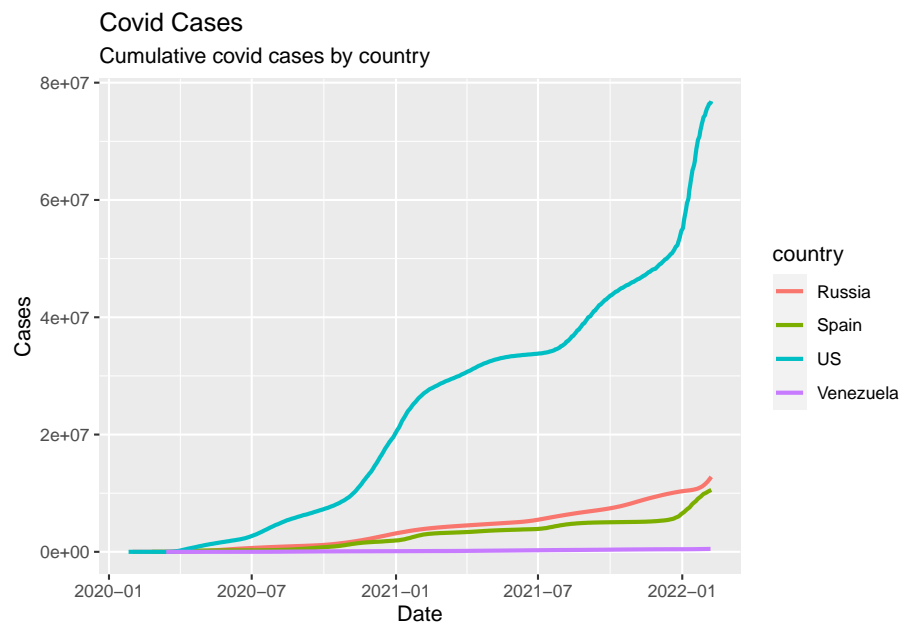
3. Now let's do a plot of COVID rate ($\#$ confirmed cases / population).
Something like this.



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

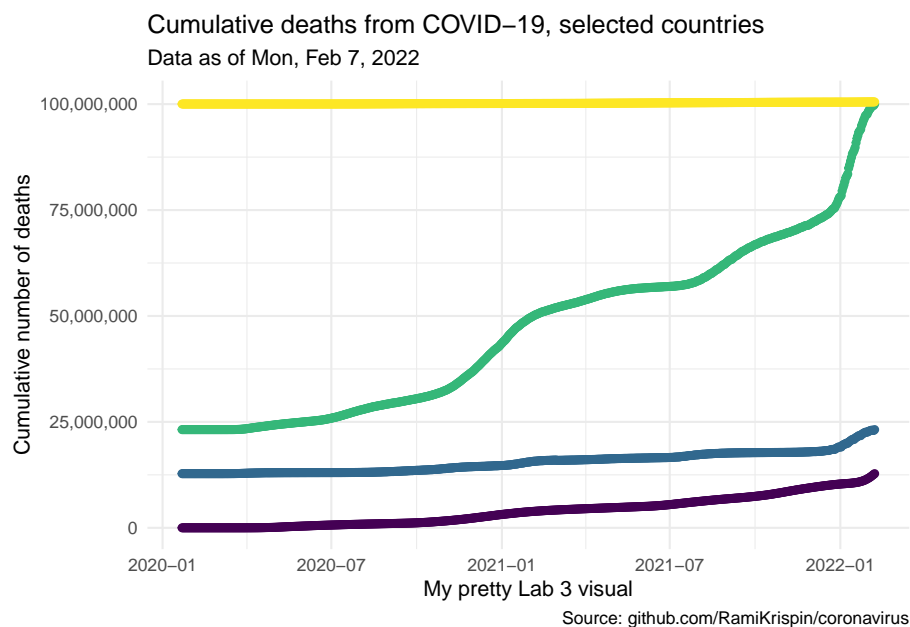
5. Make a chart with cumulative cases. Something like this:

6.1. LET'S LOOK LIKE APPLIED ANALYTICS SUPERSTARS AND MAKE SOME NEAT VISUALS.23



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 country just to make sure everything 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 748
#> 2   Spain 745
#> 3    US 748
#> 4 Venezuela 747
```

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

6.1. LET'S LOOK LIKE APPLIED ANALYTICS SUPERSTARS AND MAKE SOME NEAT VISUALS.25

```
#>      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 748
#> 2    Spain 748
#> 3      US 748
#> 4 Venezuela 748
```

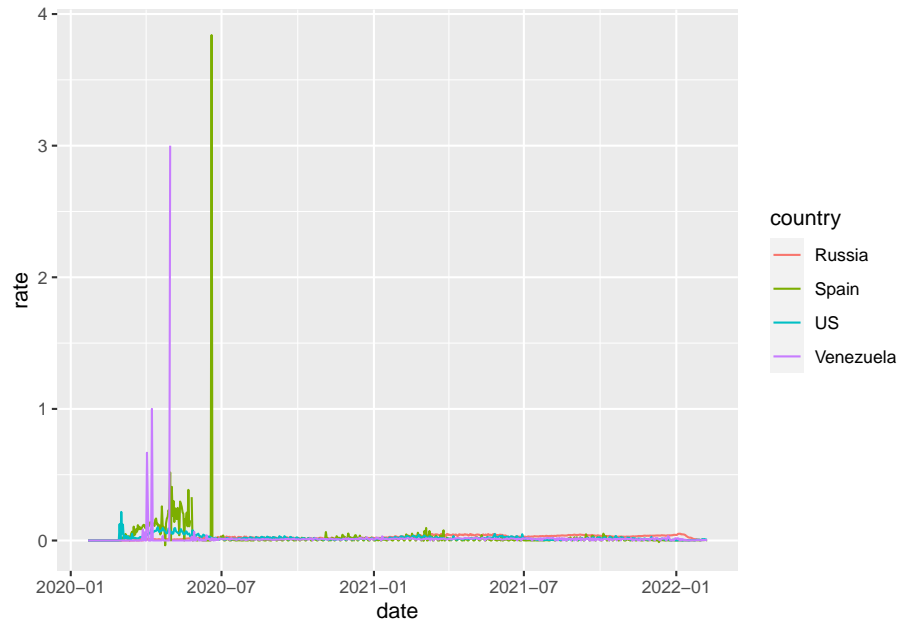
Let's join these together. I use `left_join`.

```
#>      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 748
#> 2    Spain 748
#> 3      US 748
#> 4 Venezuela 748
```

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.



```
#>      date      country      deaths
#> Min.   :2020-01-22 Length:2992 Min.    : 0.0
#> 1st Qu.:2020-07-26 Class :character 1st Qu. :  5.0
#> Median :2021-01-29 Mode  :character Median  :123.0
#> Mean   :2021-01-29      Mean   :447.5
#> 3rd Qu.:2021-08-04      3rd Qu.:633.0
#> Max.   :2022-02-07      Max.   :4442.0
#>                                     NA's   :4
#> confirmed cumulative_cases cumulative_deaths
#> Min.   : -74937 Min.   : 0 Min.   : 0
#> 1st Qu.:  450 1st Qu.:12782791 1st Qu.:16728
#> Median : 7723 Median :23178262 Median :97790
#> Mean   :33599 Mean   :41557773 Mean  :132138
#> 3rd Qu.:27682 3rd Qu.:100031030 3rd Qu.:240231
#> Max.   :1369637 Max.   :100527313 Max.   :358016
#>                                     NA's   :2120
#>      rate
#> Min.   :-0.036576
#> 1st Qu.: 0.004592
#> Median : 0.012829
#> Mean   : 0.021843
#> 3rd Qu.: 0.023418
#> Max.   : 3.840391
```

6.1. LET'S LOOK LIKE APPLIED ANALYTICS SUPERSTARS AND MAKE SOME NEAT VISUALS.27

```
#> NA's :4
```

