Project 1: Wrangling, Exploration, Visualization

SDS322E

Data Wrangling, Exploration, Visualization

Kush Patel, ksp946

Introduction

We will be looking at two datasets.

1. new_cases: This contains data of the new daily COVID-19 cases of every countries around the world. The columns contains “date” which represents date corresponds to COVID cases, “world” which is sum of COVID cases in every country at given particular date. And after that all column represents particular countries. The raw contains individual date from 12/31/2019 - 11/29/2020, so around an year worth of data.

2. new_deaths: This contains data of the new daily COVID-19 deaths of every countries around the world. The columns contains “date” which represents date corresponds to COVID deaths, “world” which is sum of COVID deaths in every country at given particular date. And after that all column represents particular countries. The raw contains individual date from 12/31/2019 - 11/29/2020, so around an year worth of data.

I choose COVID-19 data because this pandemic have directly impacted every single individual in the world. It has permanently changed many things around us. I have see many COVID-19 data graph adn other figures within past 2 years but I have never analyse covid data myself. It will be interesting to see how many cool things I can do with this data set. As this dataset contains data from every country, I can compare and contrast any two or more countries aroudn the world. Also I will be using #’s within the code to describe and answer what I am doing as there are many moving parts of this project.

```r
# read your datasets in here, e.g., with read_csv()
library(readr)
library(dplyr)
library(kableExtra)
library(stringr)
library(tidyverse)
new_cases <- read_csv("new_cases.csv")
new_deaths <- read_csv("new_deaths.csv")
```

Tidying: Reshaping

If your datasets are tidy already, demonstrate that you can reshape data with pivot wider/longer here (e.g., untidy and then retidy). Alternatively, it may be easier to wait until the wrangling section so you can reshape your summary statistics. Note here if you are going to do this.

```
# I will do tidying:reshaping of my dataset once I joined
# them together
```

Joining/Merging

```
nrow(new_cases)  #total no. of rows in this new_cases is 335
```

```
## [1] 335
```

```
ncol(new_cases)  #total no. of column in new_cases dataset is 216
```

```
## [1] 216
```

```
new_cases %>% n_distinct("date")  # there are 335 total unique IDs in new_cases dat
aset, as total unique IDs is equal to total number of rows, every rows in this data
set is unique
```

```
## [1] 335
```

```
nrow(new_deaths)  #total no. of rows in new_deaths dataset is 335
```

```
## [1] 335
```

```
ncol(new_deaths)  #total no. of column in new_deaths dataset is 216
```

```
## [1] 216
```

```
new_deaths %>% n_distinct("date")  # there are 335 total unique IDs (date) in new_d
eaths dataset
```

```
## [1] 335
```

```
full_join(new_cases, new_deaths, by = "date", suffix = c("_cases",
    "_deaths"))
```

```
## # A tibble: 335 x 431
##    date       World_cases Afghanistan_cas… Albania_cases Algeria_cases
##    <date>           <dbl>            <dbl>         <dbl>         <dbl>
##  1 2019-12-31          27                0            NA             0
##  2 2020-01-01           0                0            NA             0
##  3 2020-01-02           0                0            NA             0
##  4 2020-01-03          17                0            NA             0
##  5 2020-01-04           0                0            NA             0
```

```
## 6 2020-01-05         15                0            NA            0
## 7 2020-01-06          0                0            NA            0
## 8 2020-01-07          0                0            NA            0
## 9 2020-01-08          0                0            NA            0
## 10 2020-01-09         0                0            NA            0
## # … with 325 more rows, and 426 more variables: Andorra_cases <dbl>,
## #   Angola_cases <dbl>, Anguilla_cases <dbl>, `Antigua and
## #   Barbuda_cases` <dbl>, Argentina_cases <dbl>, Armenia_cases <dbl>,
## #   Aruba_cases <dbl>, Australia_cases <dbl>, Austria_cases <dbl>,
## #   Azerbaijan_cases <dbl>, Bahamas_cases <dbl>, Bahrain_cases <dbl>,
## #   Bangladesh_cases <dbl>, Barbados_cases <dbl>, Belarus_cases <dbl>,
## #   Belgium_cases <dbl>, Belize_cases <dbl>, Benin_cases <dbl>,
## #   Bermuda_cases <dbl>, Bhutan_cases <dbl>, Bolivia_cases <dbl>, `Bonaire Sint
## #   Eustatius and Saba_cases` <dbl>, `Bosnia and Herzegovina_cases` <dbl>,
## #   Botswana_cases <dbl>, Brazil_cases <dbl>, `British Virgin
## #   Islands_cases` <dbl>, Brunei_cases <dbl>, Bulgaria_cases <dbl>, `Burkina
## #   Faso_cases` <dbl>, Burundi_cases <dbl>, Cambodia_cases <dbl>,
## #   Cameroon_cases <dbl>, Canada_cases <dbl>, `Cape Verde_cases` <dbl>, `Cayman
## #   Islands_cases` <dbl>, `Central African Republic_cases` <dbl>,
## #   Chad_cases <dbl>, Chile_cases <dbl>, China_cases <dbl>,
## #   Colombia_cases <dbl>, Comoros_cases <dbl>, Congo_cases <dbl>, `Costa
## #   Rica_cases` <dbl>, `Cote d'Ivoire_cases` <dbl>, Croatia_cases <dbl>,
## #   Cuba_cases <dbl>, Curacao_cases <dbl>, Cyprus_cases <dbl>, `Czech
## #   Republic_cases` <dbl>, `Democratic Republic of Congo_cases` <dbl>,
## #   Denmark_cases <dbl>, Djibouti_cases <dbl>, Dominica_cases <dbl>, `Dominican
## #   Republic_cases` <dbl>, Ecuador_cases <dbl>, Egypt_cases <dbl>, `El
## #   Salvador_cases` <dbl>, `Equatorial Guinea_cases` <dbl>,
## #   Eritrea_cases <dbl>, Estonia_cases <dbl>, Ethiopia_cases <dbl>, `Faeroe
## #   Islands_cases` <dbl>, `Falkland Islands_cases` <dbl>, Fiji_cases <dbl>,
## #   Finland_cases <dbl>, France_cases <dbl>, `French Polynesia_cases` <dbl>,
## #   Gabon_cases <dbl>, Gambia_cases <dbl>, Georgia_cases <dbl>,
## #   Germany_cases <dbl>, Ghana_cases <dbl>, Gibraltar_cases <dbl>,
## #   Greece_cases <dbl>, Greenland_cases <dbl>, Grenada_cases <dbl>,
## #   Guam_cases <dbl>, Guatemala_cases <dbl>, Guernsey_cases <dbl>,
## #   Guinea_cases <dbl>, `Guinea-Bissau_cases` <dbl>, Guyana_cases <dbl>,
## #   Haiti_cases <dbl>, Honduras_cases <dbl>, Hungary_cases <dbl>,
## #   Iceland_cases <dbl>, India_cases <dbl>, Indonesia_cases <dbl>,
## #   International_cases <dbl>, Iran_cases <dbl>, Iraq_cases <dbl>,
## #   Ireland_cases <dbl>, `Isle of Man_cases` <dbl>, Israel_cases <dbl>,
## #   Italy_cases <dbl>, Jamaica_cases <dbl>, Japan_cases <dbl>,
## #   Jersey_cases <dbl>, Jordan_cases <dbl>, Kazakhstan_cases <dbl>, …
```

```
full_data <- full_join(new_cases, new_deaths, by = "date", suffix = c("_cases",
    "_deaths"))
nrow(full_data)  #total no. of rows in joined dataset are 335
```

```
## [1] 335
```

```
ncol(full_data)  #total no. of column in joined dataset are 431
```

```
## [1] 431
```

```
anti_join(new_cases, new_deaths, by = "date")
```

```
## # A tibble: 0 x 216
## # … with 216 variables: date <date>, World <dbl>, Afghanistan <dbl>,
## #   Albania <dbl>, Algeria <dbl>, Andorra <dbl>, Angola <dbl>, Anguilla <dbl>,
## #   `Antigua and Barbuda` <dbl>, Argentina <dbl>, Armenia <dbl>, Aruba <dbl>,
## #   Australia <dbl>, Austria <dbl>, Azerbaijan <dbl>, Bahamas <dbl>,
## #   Bahrain <dbl>, Bangladesh <dbl>, Barbados <dbl>, Belarus <dbl>,
## #   Belgium <dbl>, Belize <dbl>, Benin <dbl>, Bermuda <dbl>, Bhutan <dbl>,
## #   Bolivia <dbl>, `Bonaire Sint Eustatius and Saba` <dbl>, `Bosnia and
## #   Herzegovina` <dbl>, Botswana <dbl>, Brazil <dbl>, `British Virgin
## #   Islands` <dbl>, Brunei <dbl>, Bulgaria <dbl>, `Burkina Faso` <dbl>,
## #   Burundi <dbl>, Cambodia <dbl>, Cameroon <dbl>, Canada <dbl>, `Cape
## #   Verde` <dbl>, `Cayman Islands` <dbl>, `Central African Republic` <dbl>,
## #   Chad <dbl>, Chile <dbl>, China <dbl>, Colombia <dbl>, Comoros <dbl>,
## #   Congo <dbl>, `Costa Rica` <dbl>, `Cote d'Ivoire` <dbl>, Croatia <dbl>,
## #   Cuba <dbl>, Curacao <dbl>, Cyprus <dbl>, `Czech Republic` <dbl>,
## #   `Democratic Republic of Congo` <dbl>, Denmark <dbl>, Djibouti <dbl>,
## #   Dominica <dbl>, `Dominican Republic` <dbl>, Ecuador <dbl>, Egypt <dbl>, `El
## #   Salvador` <dbl>, `Equatorial Guinea` <dbl>, Eritrea <dbl>, Estonia <dbl>,
## #   Ethiopia <dbl>, `Faeroe Islands` <dbl>, `Falkland Islands` <dbl>,
## #   Fiji <dbl>, Finland <dbl>, France <dbl>, `French Polynesia` <dbl>,
## #   Gabon <dbl>, Gambia <dbl>, Georgia <dbl>, Germany <dbl>, Ghana <dbl>,
## #   Gibraltar <dbl>, Greece <dbl>, Greenland <dbl>, Grenada <dbl>, Guam <dbl>,
## #   Guatemala <dbl>, Guernsey <dbl>, Guinea <dbl>, `Guinea-Bissau` <dbl>,
## #   Guyana <dbl>, Haiti <dbl>, Honduras <dbl>, Hungary <dbl>, Iceland <dbl>,
## #   India <dbl>, Indonesia <dbl>, International <dbl>, Iran <dbl>, Iraq <dbl>,
## #   Ireland <dbl>, `Isle of Man` <dbl>, Israel <dbl>, Italy <dbl>, …
```

```
anti_join(new_deaths, new_cases, by = "date")  #anti_join here tells us that there are no IDs that appear in one data but not in the others
```

```
## # A tibble: 0 x 216
## # … with 216 variables: date <date>, World <dbl>, Afghanistan <dbl>,
## #   Albania <dbl>, Algeria <dbl>, Andorra <dbl>, Angola <dbl>, Anguilla <dbl>,
## #   `Antigua and Barbuda` <dbl>, Argentina <dbl>, Armenia <dbl>, Aruba <dbl>,
## #   Australia <dbl>, Austria <dbl>, Azerbaijan <dbl>, Bahamas <dbl>,
## #   Bahrain <dbl>, Bangladesh <dbl>, Barbados <dbl>, Belarus <dbl>,
## #   Belgium <dbl>, Belize <dbl>, Benin <dbl>, Bermuda <dbl>, Bhutan <dbl>,
## #   Bolivia <dbl>, `Bonaire Sint Eustatius and Saba` <dbl>, `Bosnia and
## #   Herzegovina` <dbl>, Botswana <dbl>, Brazil <dbl>, `British Virgin
## #   Islands` <dbl>, Brunei <dbl>, Bulgaria <dbl>, `Burkina Faso` <dbl>,
## #   Burundi <dbl>, Cambodia <dbl>, Cameroon <dbl>, Canada <dbl>, `Cape
## #   Verde` <dbl>, `Cayman Islands` <dbl>, `Central African Republic` <dbl>,
## #   Chad <dbl>, Chile <dbl>, China <dbl>, Colombia <dbl>, Comoros <dbl>,
## #   Congo <dbl>, `Costa Rica` <dbl>, `Cote d'Ivoire` <dbl>, Croatia <dbl>,
## #   Cuba <dbl>, Curacao <dbl>, Cyprus <dbl>, `Czech Republic` <dbl>,
## #   `Democratic Republic of Congo` <dbl>, Denmark <dbl>, Djibouti <dbl>,
## #   Dominica <dbl>, `Dominican Republic` <dbl>, Ecuador <dbl>, Egypt <dbl>, `El
## #   Salvador` <dbl>, `Equatorial Guinea` <dbl>, Eritrea <dbl>, Estonia <dbl>,
## #   Ethiopia <dbl>, `Faeroe Islands` <dbl>, `Falkland Islands` <dbl>,
## #   Fiji <dbl>, Finland <dbl>, France <dbl>, `French Polynesia` <dbl>,
## #   Gabon <dbl>, Gambia <dbl>, Georgia <dbl>, Germany <dbl>, Ghana <dbl>,
## #   Gibraltar <dbl>, Greece <dbl>, Greenland <dbl>, Grenada <dbl>, Guam <dbl>,
## #   Guatemala <dbl>, Guernsey <dbl>, Guinea <dbl>, `Guinea-Bissau` <dbl>,
```

```
## #   Guyana <dbl>, Haiti <dbl>, Honduras <dbl>, Hungary <dbl>, Iceland <dbl>,
## #   India <dbl>, Indonesia <dbl>, International <dbl>, Iran <dbl>, Iraq <dbl>,
## #   Ireland <dbl>, `Isle of Man` <dbl>, Israel <dbl>, Italy <dbl>, …
```

In both dataset (new_cases & new_deaths), the total number of observation were same and the unique ID (date) was also the same. So when I performed full_join to both data, there were no observation which were dropped. This is very important as now we can looked at full merged data with no observation lackiing from either datasets. The total no. of rows in full_join dataset are 335 which is equal to total no. of rows in original datasets. Total no. of column in full_joint is 431 which is double (minus common ID variable) as compared to original dataset as a result of joint.

Wrangling

```
full_data %>% pivot_longer(cols = -c("date", "World_cases", "World_deaths"),
    names_to = "name", values_to = "value") %>% separate(name,
    sep = "_", into = c("Country", "type")) %>% pivot_wider(names_from = "type",
    values_from = "value")
```

```
## # A tibble: 71,690 x 6
##    date       World_cases World_deaths Country             cases deaths
##    <date>           <dbl>        <dbl> <chr>               <dbl>  <dbl>
##  1 2019-12-31          27            0 Afghanistan             0      0
##  2 2019-12-31          27            0 Albania                NA     NA
##  3 2019-12-31          27            0 Algeria                 0      0
##  4 2019-12-31          27            0 Andorra                NA     NA
##  5 2019-12-31          27            0 Angola                 NA     NA
##  6 2019-12-31          27            0 Anguilla               NA     NA
##  7 2019-12-31          27            0 Antigua and Barbuda    NA     NA
##  8 2019-12-31          27            0 Argentina              NA     NA
##  9 2019-12-31          27            0 Armenia                 0      0
## 10 2019-12-31          27            0 Aruba                  NA     NA
## # … with 71,680 more rows
```

```
clean <- full_data %>% pivot_longer(cols = -c("date", "World_cases",
    "World_deaths"), names_to = "name", values_to = "value") %>%
    separate(name, sep = "_", into = c("Country", "type")) %>%
    pivot_wider(names_from = "type", values_from = "value")
```

Here, I used pivot_longer on full_data to tidy dataset and have each country their own rows. So the dataset became more longer and wider. Then I separated cases and deaths into their own separate categories. Finally, I used pivot_wider to assign cases and deaths values their own column. Now, the clean data looks much more organized and tidy. The final clean dataset have column data, column for World_cases and World_deaths which are numerical and catergorical value in Country column. In total, this dataset have 6 variables and 72690 observations.

```
# the new column 'ratio' contains ratio of daily deaths to
# cases on a give date, we have used na.omit to omit any raws
# this do not have any data. Because of na.omit, we lost many
# raws, the clean dataset have 71690 obs, this one have 58685
# obs.
clean %>% mutate(ratio = deaths/(cases + 1)) %>% na.omit() %>%
    arrange(desc(ratio))
```

```
## # A tibble: 58,685 x 7
##    date       World_cases World_deaths Country     cases deaths ratio
##    <date>           <dbl>        <dbl> <chr>       <dbl>  <dbl> <dbl>
##  1 2020-08-25      213162         4247 Kazakhstan      0    108   108
##  2 2020-10-02      323180         8775 Ukraine         0     64    64
##  3 2020-06-27      189698         4563 Argentina       0     43    43
##  4 2020-04-20       70507         4986 Cameroon        0     21    21
##  5 2020-06-26      178867         6565 France          0     21    21
##  6 2020-07-28      215352         4654 Argentina       0     17    17
##  7 2020-04-25       73149         5343 Ecuador         0     16    16
##  8 2020-06-19      138809         6276 Argentina       0     16    16
##  9 2020-07-05      187773         4333 Argentina       0     16    16
## 10 2020-06-29      159070         3063 El Salvador     0     12    12
## # … with 58,675 more rows
```

```
# Using group_by, summarize, and arrange core function to see
# which country have most total_death
clean %>% group_by(Country) %>% na.omit() %>% summarize(total_death = sum(deaths))
%>%
    arrange(desc(total_death))
```

```
## # A tibble: 214 x 2
##    Country        total_death
##    <chr>                <dbl>
##  1 United States       266063
##  2 Brazil              172561
##  3 India               136696
##  4 Mexico              105459
##  5 United Kingdom       58030
##  6 Italy                54363
##  7 France               52127
##  8 Iran                 47486
##  9 Spain                44668
## 10 Russia               39527
## # … with 204 more rows
```

```
# Using filter and select core function to visualize only
# China's cases and dealths
clean %>% filter(Country == "China") %>% select(date, Country,
    cases, deaths)
```

```
## # A tibble: 335 x 4
##    date       Country cases deaths
##    <date>     <chr>   <dbl>  <dbl>
##  1 2019-12-31 China      27      0
##  2 2020-01-01 China       0      0
##  3 2020-01-02 China       0      0
##  4 2020-01-03 China      17      0
##  5 2020-01-04 China       0      0
##  6 2020-01-05 China      15      0
##  7 2020-01-06 China       0      0
##  8 2020-01-07 China       0      0
##  9 2020-01-08 China       0      0
```

```
## 10 2020-01-09 China       0      0
## # … with 325 more rows
```

```
# Using Stringr function to detect name of country starting
# with letter C
clean %>% distinct(Country) %>% filter(str_detect(Country, "[C]"))
```

```
## # A tibble: 22 x 1
##    Country
##    <chr>
##  1 Cambodia
##  2 Cameroon
##  3 Canada
##  4 Cape Verde
##  5 Cayman Islands
##  6 Central African Republic
##  7 Chad
##  8 Chile
##  9 China
## 10 Colombia
## # … with 12 more rows
```

```
# Using Stringr function to replace country name
clean %>% filter(Country == "United Kingdom") %>% mutate(Country2 = str_replace(Cou
ntry,
    "United Kingdom", "UK"))
```

```
## # A tibble: 335 x 7
##    date       World_cases World_deaths Country        cases deaths Country2
##    <date>           <dbl>        <dbl> <chr>          <dbl>  <dbl> <chr>
##  1 2019-12-31          27            0 United Kingdom     0      0 UK
##  2 2020-01-01           0            0 United Kingdom     0      0 UK
##  3 2020-01-02           0            0 United Kingdom     0      0 UK
##  4 2020-01-03          17            0 United Kingdom     0      0 UK
##  5 2020-01-04           0            0 United Kingdom     0      0 UK
##  6 2020-01-05          15            0 United Kingdom     0      0 UK
##  7 2020-01-06           0            0 United Kingdom     0      0 UK
##  8 2020-01-07           0            0 United Kingdom     0      0 UK
##  9 2020-01-08           0            0 United Kingdom     0      0 UK
## 10 2020-01-09           0            0 United Kingdom     0      0 UK
## # … with 325 more rows
```

```
# Using 5 unique functions inside of summarize
clean %>% group_by(Country) %>% summarize(Mean_cases = mean(cases,
    na.rm = T), SD_cases = sd(cases, na.rm = T), Max_cases = max(cases,
    na.rm = T), Median_cases = median(cases, na.rm = T), Min_cases = min(cases,
    na.rm = T)) %>% slice(1:10) %>% knitr::kable()
```

Country	Mean_cases	SD_cases	Max_cases	Median_cases	Min_cases
Afghanistan	141.0584615	207.3809333	1063	58.0	0
Albania	138.3082707	172.0773052	836	90.0	0
Algeria	246.8449848	270.8235920	2102	165.0	0

Country	Mean_cases	SD_cases	Max_cases	Median_cases	Min_cases
Andorra	25.5555556	50.7237027	299	1.0	0
Angola	59.8690476	81.8104572	355	21.0	0
Anguilla	0.0161290	0.1550171	2	0.0	0
Antigua and Barbuda	0.5529412	2.6569144	39	0.0	0
Argentina	5273.7388060	5150.5079506	18326	4100.0	0
Armenia	413.3987730	611.0726714	4527	185.5	0
Aruba	19.1785714	32.6312834	176	2.5	0

```
clean %>% group_by(Country) %>% summarize(Mean_death = mean(deaths,
    na.rm = T), SD_death = sd(deaths, na.rm = T), Max_death = max(deaths,
    na.rm = T), Median_death = median(deaths, na.rm = T), Min_deaths = min(deaths,
    na.rm = T)) %>% slice(1:10) %>% knitr::kable()
```

Country	Mean_death	SD_death	Max_death	Median_death	Min_deaths
Afghanistan	5.4246154	8.7720451	56	2	0
Albania	2.9586466	3.3299924	19	2	0
Algeria	7.2735562	6.1894333	42	7	0
Andorra	0.2911877	0.7592246	6	0	0
Angola	1.3690476	1.9152133	13	1	0
Anguilla	0.0000000	0.0000000	0	0	0
Antigua and Barbuda	0.0156863	0.1528889	2	0	0
Argentina	142.9440299	244.7596556	3351	58	0
Armenia	6.5705521	8.7975307	41	3	0
Aruba	0.1785714	0.4593616	3	0	0

```
# as we can see here that the mean cases are significantly
# higher then median world cases, this tells that our data is
# skewed towards higher value
clean %>% summarize(Mean_world_cases = mean(World_cases, na.rm = T),
    SD_w_cases = sd(World_cases, na.rm = T), Max_w_cases = max(World_cases,
        na.rm = T), Median_w_cases = median(World_cases, na.rm = T),
    Min_w_cases = min(World_cases, na.rm = T)) %>% slice(1:10)
```

```
## # A tibble: 1 x 5
##   Mean_world_cases SD_w_cases Max_w_cases Median_w_cases Min_w_cases
##              <dbl>      <dbl>       <dbl>          <dbl>       <dbl>
## 1          185884.    172558.      679758         135663           0
```

```
clean %>% summarize(Mean_world_deaths = mean(World_deaths, na.rm = T),
    SD_w_death = sd(World_deaths, na.rm = T), Max_w_death = max(World_deaths,
        na.rm = T), Median_w_death = median(World_deaths, na.rm = T),
    Min_w_deaths = min(World_deaths, na.rm = T)) %>% slice(1:10)
```

```
## # A tibble: 1 x 5
##   Mean_world_deaths SD_w_death Max_w_death Median_w_death Min_w_deaths
##               <dbl>      <dbl>       <dbl>          <dbl>        <dbl>
## 1             4339.      2925.       12583           4839            0
```

```
# Total observation in categorical variable
clean %>% group_by(Country) %>% summarize(total_obs = n())
```

```
## # A tibble: 214 x 2
##    Country               total_obs
##    <chr>                     <int>
##  1 Afghanistan                 335
##  2 Albania                     335
##  3 Algeria                     335
##  4 Andorra                     335
##  5 Angola                      335
##  6 Anguilla                    335
##  7 Antigua and Barbuda         335
##  8 Argentina                   335
##  9 Armenia                     335
## 10 Aruba                       335
## # … with 204 more rows
```

```
# overall mean and sd cases of COVID based on country
clean %>% group_by(Country) %>% summarize(mean_cases = round(mean(cases,
    na.rm = T)), sd_cases = round(sd(cases, na.rm = T)))
```

```
## # A tibble: 214 x 3
##    Country               mean_cases sd_cases
##    <chr>                      <dbl>    <dbl>
##  1 Afghanistan                  141      207
##  2 Albania                      138      172
##  3 Algeria                      247      271
##  4 Andorra                       26       51
##  5 Angola                        60       82
##  6 Anguilla                       0        0
##  7 Antigua and Barbuda            1        3
##  8 Argentina                   5274     5151
##  9 Armenia                      413      611
## 10 Aruba                         19       33
## # … with 204 more rows
```

```
# how many distinct countries and how many observations are
# there
clean %>% summarize(mean_cases = round(mean(cases, na.rm = T)),
    n(), n_distinct(Country))
```

```
## # A tibble: 1 x 3
##   mean_cases `n()` `n_distinct(Country)`
##        <dbl> <int>                 <int>
## 1       1061 71690                   214
```

```
# monthly average of COVID cases based on country
clean %>% mutate(month = format(date, "%m"), year = format(date,
    "%Y")) %>% group_by(year, month, Country) %>% summarize(monthy_average = sum(ca
ses,
    na.rm = T))
```

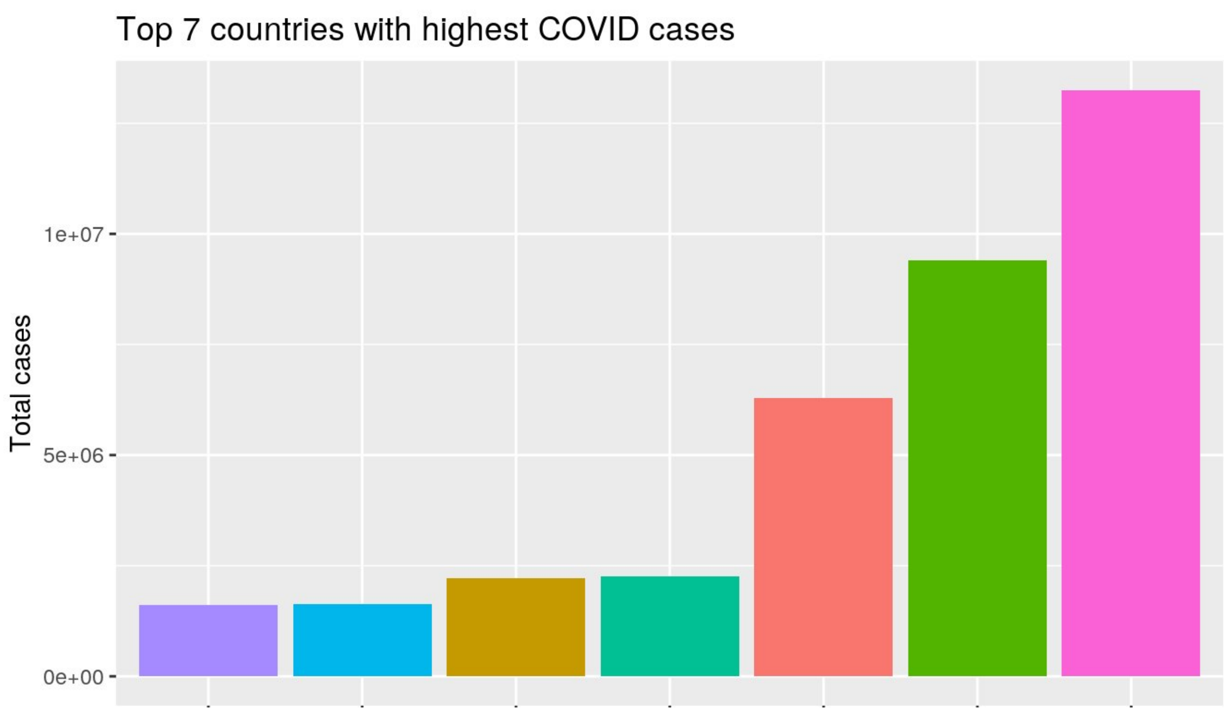
```
## # A tibble: 2,568 x 4
```

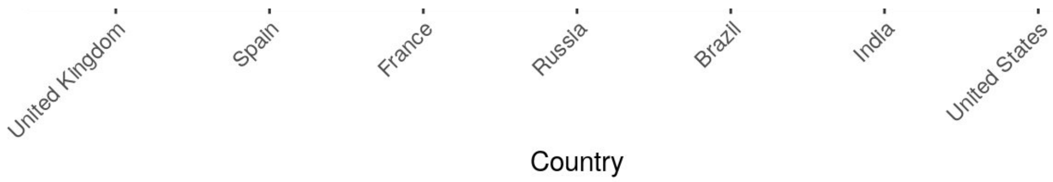
```
## # Groups:   year, month [12]
##    year  month Country             monthy_average
##    <chr> <chr> <chr>                        <dbl>
##  1 2019  12    Afghanistan                      0
##  2 2019  12    Albania                          0
##  3 2019  12    Algeria                          0
##  4 2019  12    Andorra                          0
##  5 2019  12    Angola                           0
##  6 2019  12    Anguilla                         0
##  7 2019  12    Antigua and Barbuda              0
##  8 2019  12    Argentina                        0
##  9 2019  12    Armenia                          0
## 10 2019  12    Aruba                            0
## # … with 2,558 more rows
```

Data wrangling can be used to extract any particular information from the table. For intense, I can arrange countries by total number of covid death. From the code I see that US is no. 1 and UK is no.5 when comes to covid deaths. I can also filter out specific country I am looking for. I filtered out China to looked at it cases and covid deaths. Next, I used Stringr function to replace the name of Country. Summarized function can be used to find statistical summarary of data including mean, median, max, min, and sd. Using n_distinct function under summarized, I found out how many distint countries are there in my dataset, here I had 214 distinct countries. Lastly, I computed monthly covid cases average of all countries, it help to better visualized which month was worst and which was better.

Visualizing

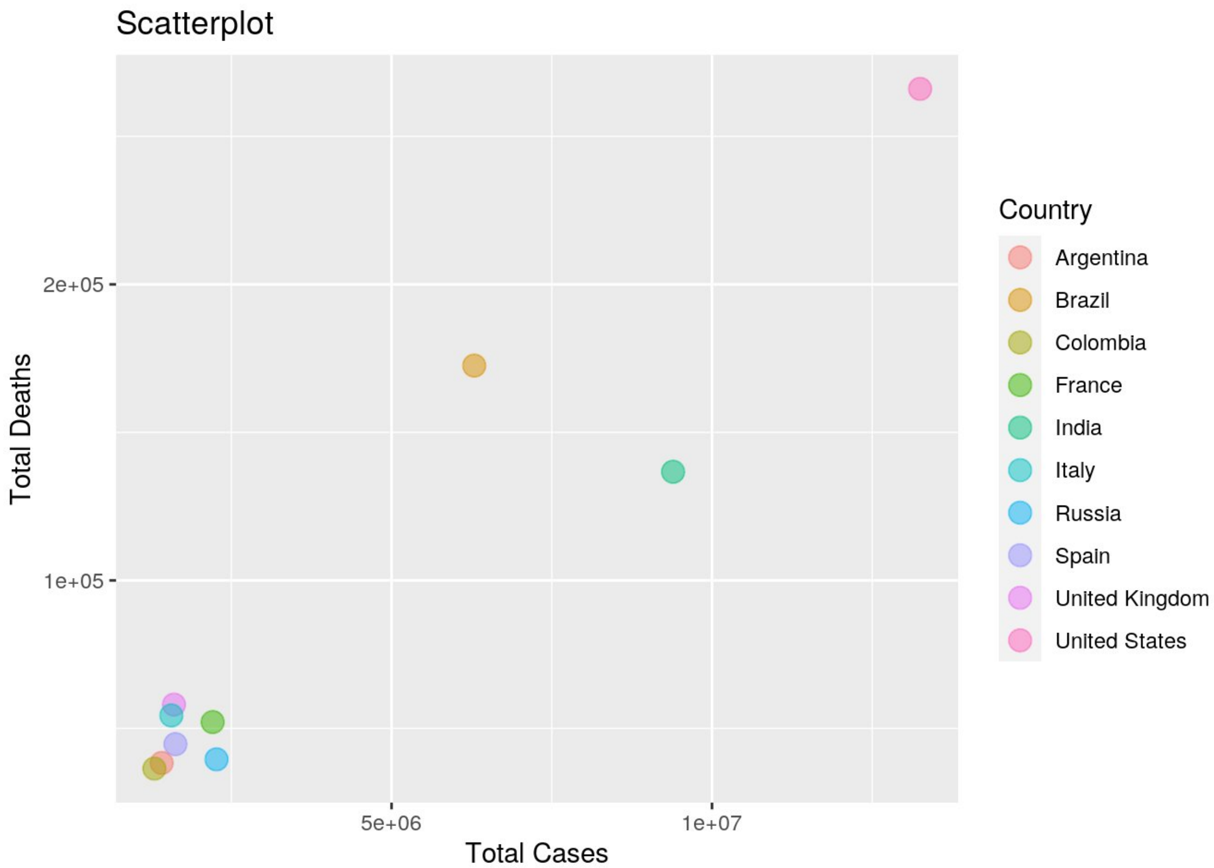
```
clean %>% group_by(Country) %>% summarize(total_cases = sum(cases,
    na.rm = T), sd = sd(cases, na.rm = T)) %>% arrange(desc(total_cases)) %>%
    slice(1:7) %>% ggplot(aes(x = reorder(Country, total_cases),
    y = total_cases, fill = Country)) + geom_bar(stat = "identity") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") + ggtitle("Top 7 countries with highest COVID cas
es") +
    xlab("Country") + ylab("Total cases")
```



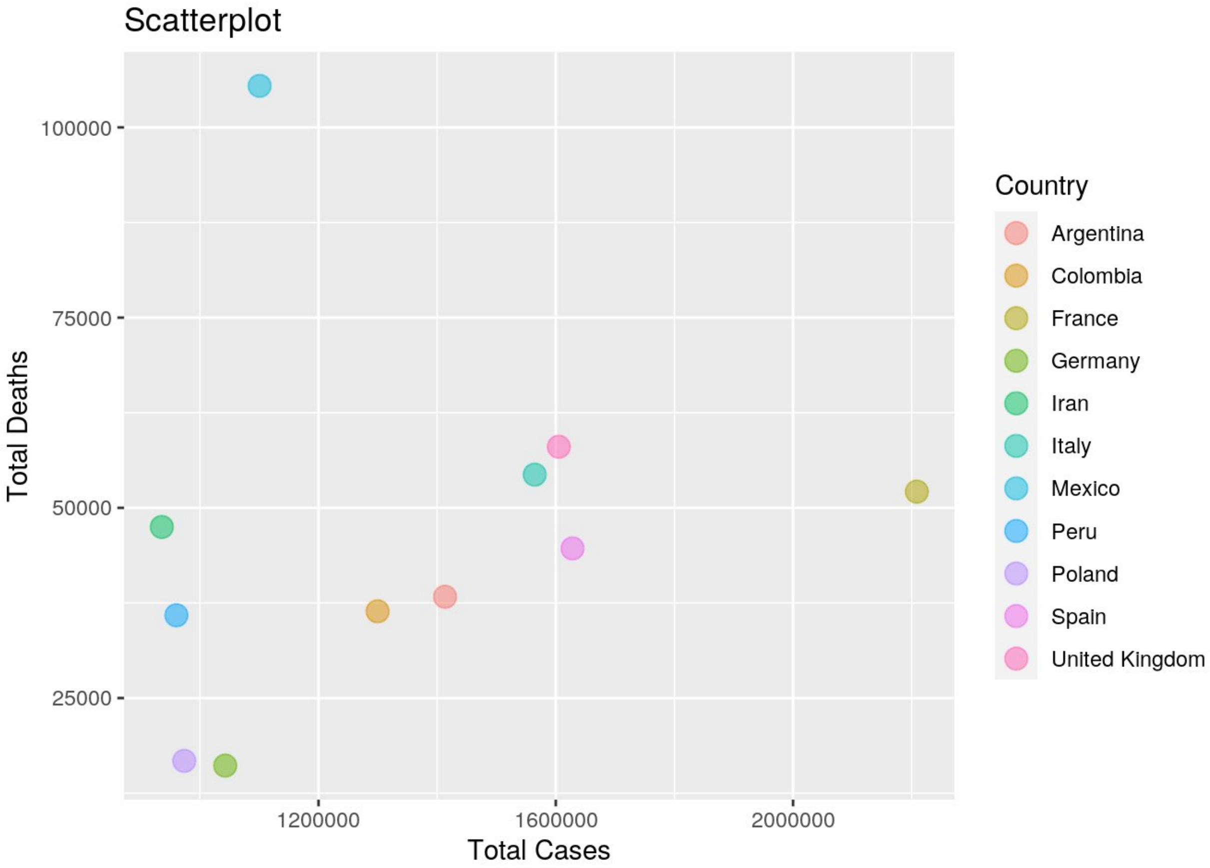


This is the bar graph of top seven countries in the world when comes to total COVID cases. We have name of country on x-axis and total cases on y-axis. As we can see from the graph US has highest COVID cases followed by India and Brazil. One thing to keep in note that this is not entire covid data, this data is only from Dec 2019 to Nov 2020, so any cases after that date have not been recorded. One of the other useful graph besides this would be cases by capita because that will give accurate representation of cases compared to countries population.

```
# plot A
clean %>% group_by(Country) %>% summarize(total_cases = sum(cases,
    na.rm = T), total_deaths = sum(deaths, na.rm = T)) %>% arrange(desc(total_case
s)) %>%
    slice(1:10) %>% ggplot(aes(total_cases, total_deaths)) +
    geom_point(aes(color = Country), size = 4, alpha = 0.5) +
    theme_grey() + ggtitle("Scatterplot ") + xlab("Total Cases") +
    ylab("Total Deaths")
```




```
# plot B
clean %>% group_by(Country) %>% summarize(total_cases = sum(cases,
    na.rm = T), total_deaths = sum(deaths, na.rm = T)) %>% arrange(desc(total_case
s)) %>%
    slice(5:15) %>% ggplot(aes(total_cases, total_deaths)) +
    geom_point(aes(color = Country), size = 4, alpha = 0.5) +
    theme_grey() + ggtitle("Scatterplot ") + xlab("Total Cases") +
    ylab("Total Deaths")
```

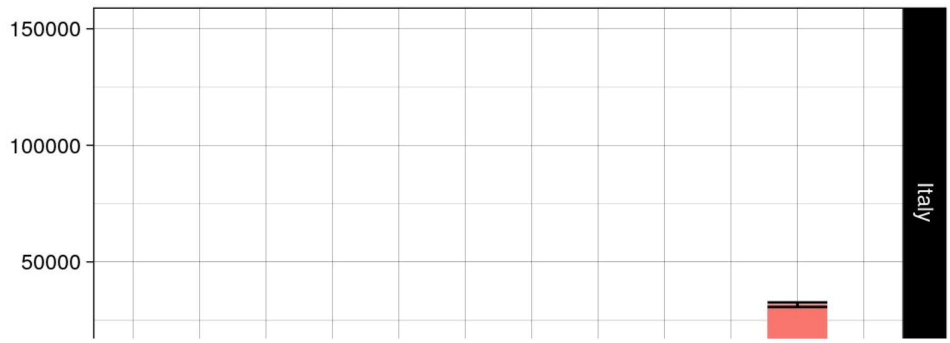


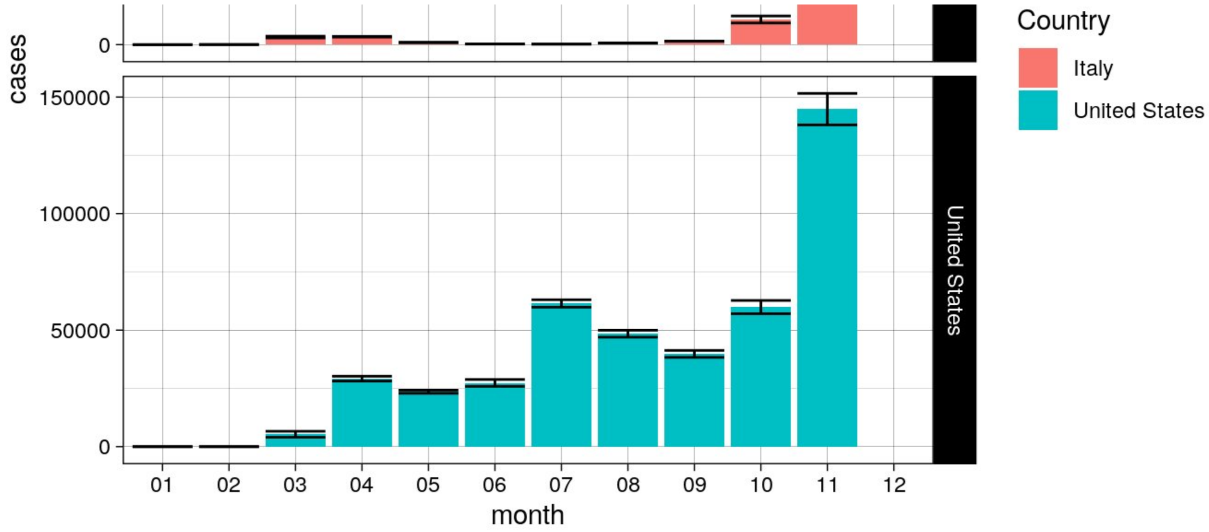
Plot A shows scatterplots between total cases and total deaths of top 10 countries with highest cases. As we can see that US, Inida, and Brazil stood apart in the graph but it is very hard to tell the difference between rest of the countries.

So for plot B, we took scatterplots between total cases and total dealth of top 5 to 15 countries, since it will be easier visually to compare them. From graph we see there is significant distinction between point of France and Mexico. France have high covid cases but low covid deaths, which tells that France covid mortality rate is low compared to average countries. This might be because of better hospitalization or any other reasons. On the other hand Mexico have low covid cases but realievely high covid death, which full Mexico higher in term of covid mortality.

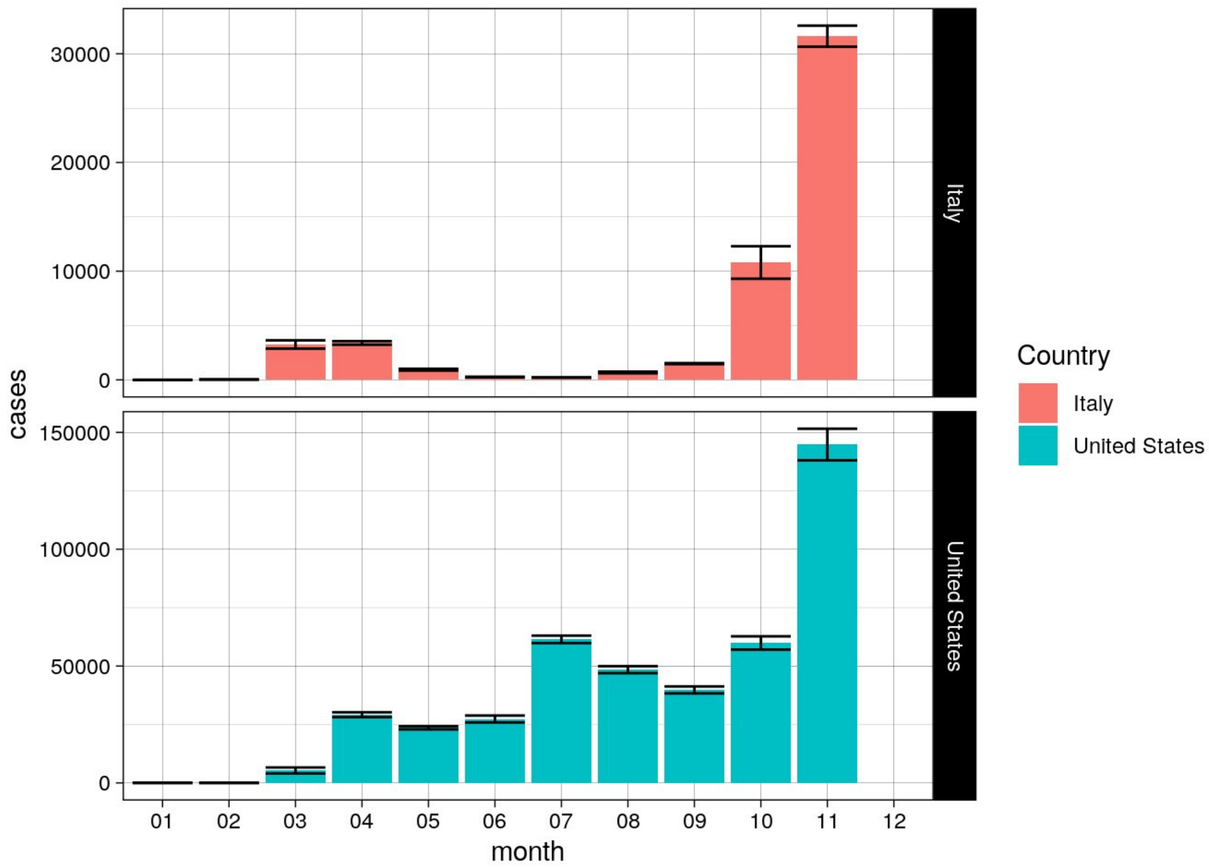
```
US_Italy <- clean %>% mutate(month = format(date, "%m"), year = format(date,
    "%Y")) %>% group_by(year, month, Country) %>% filter(Country ==
    "United States" | Country == "Italy")

# plot C
ggplot(data = US_Italy, aes(x = month, y = cases, fill = Country)) +
    geom_bar(stat = "summary") + geom_errorbar(stat = "summary") +
    facet_grid(Country ~ .) + theme_linedraw()
```






```r
# plot D
ggplot(data = US_Italy, aes(x = month, y = cases, fill = Country)) +
    geom_bar(stat = "summary") + geom_errorbar(stat = "summary") +
    facet_grid(Country ~ ., scales = "free_y") + theme_linedraw()
```



Plot C consists of bar graph of average-monthly covid cases in Italy and United States. The x-axis have months, months 1-11 is of year 2020 and month 12 is from year 2019, it is just the way data was collected. There is also error bar on top of every bar which shows standard deviation of monthly cases. It gives visual representation of monthly covid cases of US compared to Italy. In US, the first wave of covid was peaked around month of July and second wave peaked in November. In Italy, there was high period in March and April 2020 (this was around the time when whole world was watching Italy going to lockdown), later in the year, the second time cases really starting to rise was in October and November. Also note that the y-axis scale on both graph is same, so it help to accurately visualized the difference.

Plot D consist of same information, but this time we have different value for y-axis. See how this graph looks very different then previous graph, although it is same data. This can be potentially misleading as viewers thinks that in November cases in US and Italy are almost same but the reality is very different. This technique of misleading is widely used by news channel to influence views of people. This is also the reason why the axis in graph should be clearly labal.

Concluding Remarks

Using data wrangling, exploration, and visualization we can compare and contract the handling of COVID pandemic of different countries or same country over period of time. We can also visualized when was the peak of COVID in particular country and how long it took to come back to average cases. We can perform countless number of functions and built various graph using just on full_joint dataset. In conclusion overall COVID pandemic had two waves between the period Dec 2019 and Nov 2020, first wave was around summer when many people were desperate to go in public places after months of lockdown and second wave was during the ending of year 2020. One of the downfall of this type of covid data might be that not all countires uses same matrix to record the cases, and many times it happens that a country does not reports all covid cases, so it can be hard to compare real impact of covid as the number can be way more higher.