John's Hopkins COVID 19 Data Analysis

2022-12-06

Note: To run this RMD, you need to have the Forecast library installed. Install it with `install.packages('forecast')`.

Project Description

In this project, we are going to analyze the John's Hopkins COVID 19 Dataset. We will conduct two visualizations and one model.

The data is made public by John's Hopkins University, and keeps track of the number of cases and deaths per day for every unique combination of locality and country/region.

Import and Set-up Data

To begin, I will import the data and make it available in our environment. I will follow the same procedure Dr. Wall used in class, and also tidy the data in the same way that she did. I will include analaysis beyond the scope of our course, though.

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov

file_names <- c('time_series_covid19_confirmed_global.csv', 'time_series_covid19_deaths_global.csv', 't

urls <- str_c(url_in, file_names)

global_cases <- read_csv(urls[1], show_col_types = FALSE)
global_deaths <- read_csv(urls[2], show_col_types = FALSE)
us_cases <- read_csv(urls[3], show_col_types = FALSE)
us_deaths <- read_csv(urls[4], show_col_types = FALSE)
```

Tidying the Dataset

Now that we have the data imported into our R environment, I am going to use pivot tables to tidy the data a bit, similar to how we used them in class.

We currently have a column for each date - this is not an easy way to read the COVID data. I will attempt to use the pivot table to extract the data that I care about for this analysis.

```
global_cases <- global_cases %>%
  pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat, Long))

global_deaths <- global_deaths %>%
  pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long))
```

```
us_cases <- us_cases %>%
  pivot_longer(cols = -c(`Province_State`, `Country_Region`, Lat, Long_, UID, iso2, iso3, code3, FIPS,
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat, Long_, UID, iso2, iso3, code3, FIPS, Combined_Key))

us_deaths <- us_deaths %>%
  pivot_longer(cols = -c(`Province_State`, `Country_Region`, Lat, Long_, UID, iso2, iso3, code3, FIPS,
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long_, UID, iso2, iso3, code3, FIPS, Combined_Key))
```

Great! Now, we have dataframes for each of our variables that are a lot easier to work with. We can see the region in question, the state or province, the date entry, and the number of cases on the given day.

Now, we can combine the cases and deaths into one object for global and US cases, respectively. In the same step, I will use Lubridate to turn our date value into a date object. I'll also get rid of the old dataframes, since we have a new representation of the same data, and the old data are consuming significant memory.

```
global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`, Province_State = `Province/State`) %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Province/State", "Country/Region", "date")
```

```
us <- us_cases %>%
  full_join(us_deaths) %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Admin2", "Province_State", "Country_Region", "date")

## Warning: 3342 failed to parse.
```

```
remove(global_cases, global_deaths, us_cases, us_deaths)
```

Next, we should check to see if there are any outliers we should remove. You'll notice from above that some of our US cases failed to parse due to missing dates. Let's address that issue now, and remove points without a date. We'll also get rid of observations that have no cases. Also, we will make sure a state exists for each of these.

```
us <- us %>% filter(!is.null(date))
us <- us %>% filter(cases > 0)
us <- us %>% filter(!is.null(Province_State))
global <- global %>% filter(cases > 0)
```

Awesome! Now, we are ready to do some visualizations.

Visdualizing the Data

Fastest Onset of COVID 19

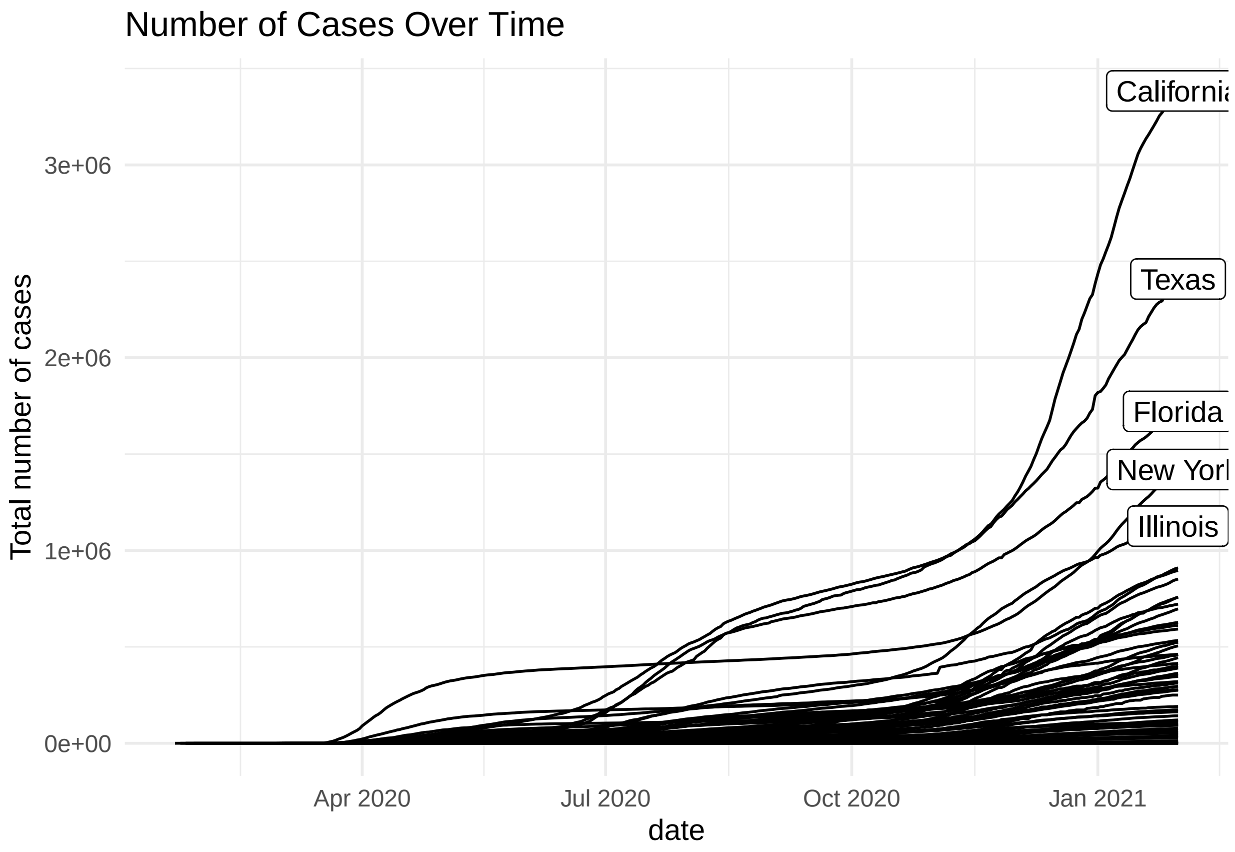
Which US states had the fastest onset of COVID 19 cases? We can find out using visualization.

I will start by grouping the cases in the US by state in the first 5 months of the disease onset.

```
group_state <- us %>%
  filter(date < '2021-02-01') %>%
  group_by(Province_State, date) %>%
  summarise(total_cases = sum(cases),
                         .groups = 'drop')
```

Now that we have this data stored, we can use `ggplot` to analyze which state grew in cases the fastest.

```
ggplot(data=group_state, aes(x = date, y = total_cases, group = Province_State)) +
  geom_line() +
  geom_label(aes(label = Province_State), data = group_state %>% filter(date == max(date)) %>% filter(t
  ggtitle("Number of Cases Over Time") +
  theme_minimal() +
  ylab("Total number of cases")
```



Here, we can see the states with the most rapid onset of COVID 19 over the first three months of the outbreak.

Identifying Countries with Highest Case Death Rate

I want to use the data we have to identify the highest rate of deaths to cases in order to determine which countries were likely to handle COVID in the worst manner.

I will group by country, much like I did in the prior visualization.

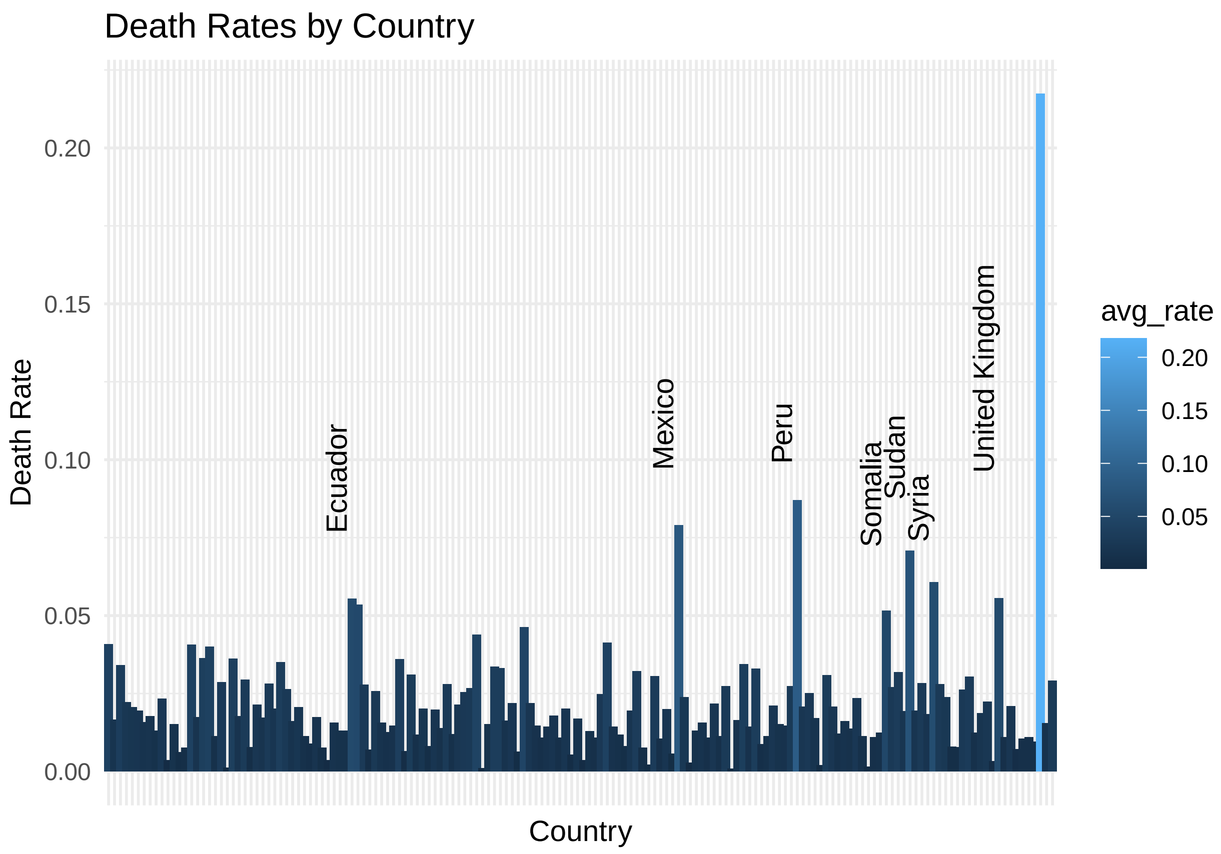
```
group_glob <-global %>%
  filter(deaths > 0) %>%
  group_by(Country_Region, date) %>%
  summarise(total_cases = sum(cases),
            total_deaths = sum(deaths),
            rate = total_deaths / total_cases,
                         .groups = 'drop') %>%
  filter(rate < 1) %>%
  filter(total_cases > 400) %>%
  filter(total_deaths > 200)
```

Now, I will calculate a bar-graph visualization of country with the worst death rate. In this way, we should be able to see which country handled COVID 19 in the worst manner during the outbreak.

```
max_rates <- group_glob %>% group_by(Country_Region) %>% summarise(avg_rate = mean(rate))

ggplot(data=max_rates, aes(x = Country_Region, y = avg_rate)) +
  geom_bar(aes(fill = avg_rate), stat = "identity", width = 1.5) +
  geom_text(aes(label = Country_Region), data = max_rates %>% filter(avg_rate > 0.05), vjust = -0.25, h
  ggtitle("Death Rates by Country") +
  theme_minimal() +
  theme(axis.text.x=element_blank()) +
  ylab("Death Rate") +
  xlab("Country")
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
```



Great! Now, we can see the countries with the highest death rates.

Modeling the Data

For my model today, I want to try to forecast the global death rate for the next 12 months.

To do this, I will use something called Time Series Forecasting using a model called ARIMA.

```
df <- global %>% group_by(date) %>% summarise(total_cases = sum(cases), total_deaths = sum(deaths), rat

t_series <- ts(df)

armia_model <- auto.arima(t_series)
summary(armia_model)
```

```
## Series: t_series
## ARIMA(4,1,3)
##
## Coefficients:
##           ar1     ar2     ar3      ar4     ma1      ma2      ma3
##       -0.3186  0.8004  0.5639  -0.1097  0.3980  -0.4587  -0.3647
## s.e.   0.2723  0.0959  0.2260   0.0641  0.2724   0.0930   0.1657
##
## sigma^2 = 8.709e-08:  log likelihood = 7034.11
## AIC=-14052.22   AICc=-14052.08   BIC=-14012.58
##
## Training set error measures:
##                        ME         RMSE          MAE        MPE      MAPE
## Training set 1.00315e-06 0.0002939883 8.129262e-05 0.01306927 0.2527365
##                   MASE     ACF1
## Training set 0.5504545 0.021647
```

Excellent! Now, we can use our time series forecast to predict the global error rate over the next 12 months.

I wanted to try Time Series Forecasting because it's new to me, and I am pleased with how easy it was to use after calculating the death rate daily from our dataset.

Identifying Model Bias

In this case, I think we do need to be aware of some potential model bias. I think that there are external forces that could impact our time series forecast. For example, what about new vaccines that quell the death rate further than expected?

It's our job as data scientists to be able to identify when a model might have an issue that makes it invalid, so before using this model in practice, we would need to consider any kind of external pressure to the dataset like a new vaccine or even a new variant of the virus.