MATP-4400 COVID-19 Final Notebook

Thomas Hopkins

May 2020

Contents

Final Project: Submission Links

- github repository: https://github.com/TheRensselaerIDEA/COVID-Notebooks
- My github ID: *thomashopkins32*
- github issues addressed by this work: `#18` and `#24`
- Github branch name of my submitted notebook: *feature-24*
- link to merged notebook (post these to LMS!:
  - https://github.com/TheRensselaerIDEA/COVID-Notebooks/blob/master/MATP-4400-FINAL/ThomasHopkins_FINAL_2020.Rmd
  - https://github.com/TheRensselaerIDEA/COVID-Notebooks/blob/master/MATP-4400-FINAL/ThomasHopkins_FINAL_2020.html

Overview & Problems Tackled

Two key issues in understanding the overall impact of an epidemic are determining how rapidly the disease is spreading and predicting future outcomes of the disease based on similar experiences.

I first decided to estimate the effective reproductive number R_e for New York State over time. This gives a baseline for the expected number of individuals to contract the disease if one individual were to have it in the population.

Second, I decided to look at properties of diseases that would help in predicting when a second wave of the virus may occur. This is done under the assumption that a second wave will definitely occur (something that should be analyzed separately).

The results from the first issue show that the R_e is decreasing over time while the results of the second issue suggest that a second wave will occur sometime in the Fall of this year.

Data Description

The first issue uses time series data found at data/csv/time_series/time_series_covid19_confirmed_US.csv on the GitHub Repository. It includes the totals of confirmed cases for each state in the US over time. The date range is from 1/22/20 to 5/5/20.

The second issue uses data from many sources that I compiled into a suitable format for the task. This can be found at data/csv/similar_diseases.csv on the GitHub Repository. The sources I used for the numbers found in the data file can be found in the References section below. Here is a view of the data I compiled:

```
diseases <- read.csv('../data/csv/similar_diseases.csv')
rownames(diseases) <- diseases[,'NAME']
diseases$NAME <- NULL
diseases
```

```
##                   R0 mortality_rate month_first_peak seasonal airborne
## SARS           2.750          0.065         February    False     True
## MERS           4.275          0.344            April    False     True
## H1N1           1.480          0.264             July     True     True
## Measles       12.000          0.150            April     True     True
## Cholera        2.000          0.050          October     True    False
## Yellow fever   6.150          0.075          January     True     True
## Ebola         18.000          0.500        September    False    False
## Dengue fever  27.200          0.010          January    False    False
## COVID-19       2.400          0.070         February    False     True
##               month_second_peak
## SARS                        May
## MERS                       June
## H1N1                    October
## Measles                   April
## Cholera                   March
## Yellow fever            January
## Ebola                  November
## Dengue fever            October
## COVID-19                   <NA>
```

I took features of many similar (in terms of impact on society) diseases and used them for a classification task.

`R0` denotes the average basic reproduction number

`mortality_rate` denotes the rate at which individuals who are infected die from disease

`month_first_peak` denotes the first month in which the disease became an epidemic

`airborne` denotes whether or not the disease can be spread through the air

`month_season_peak` denotes the season in which a reemergence of the disease appeared after the initial outbreak

The diseases I included in this analysis include SARS, MERS, H1N1, Measles, Cholera, Yellow fever, Ebola, and Dengue fever.

Results

Problem 1: Estimating R_e in New York

The first problem tackled was to look into the effective reproductive number (R_0) for New York State. Since R_0 is difficult to calculate with our current data, I chose to utilize the EpiEstim package to estimate R_0

called R_e. This used the incidence of COVID-19 in New York State as well as a serial interval distribution with a mean of 3.96 and a standard deviation of 4.75.

Methods

To address this problem, I decided to use the incidence of COVID-19 in New York State. This data was not readily available, so I took the cumulative cases over time and subtracted each day from the previous to get the total *new* cases for each day since the first contact.

Furthermore, I assumed that the days in which the total cumulative cases were under 70 came from outside New York State. I then separated the new cases for each day into imported cases (those with less than 70 cumulative) and local cases.

With both imported incidence and local incidence at hand, all that was left was to determine a serial interval to use. From 1, we can see that the serial interval for COVID-19 appears to have a mean of 3.96 days and a standard deviation of 4.75.

All that remains is to calculate the R_e and plot it over time. I used the `EpiEstim` package's `estimate_R()` function to do this.

Results

First I read in the time-series data for New York State. I need to calculate the incidence of COVID-19 and differentiate between imported and local cases. We assume the entries with less than 70 cases are considered imported. By imported we mean that the infected individual came from outside the region. Local cases are the entries with more than 70 cases.

```
# read in the time-series US data
covid_TS_states <- read.csv('../data/csv/time_series/time_series_covid19_confirmed_US.csv')
# filter data by New York and sum up the total number of cases
covid_TS_NY <- covid_TS_states[covid_TS_states$Province_State=='New York',] %>%
  group_by(Province_State) %>%
  summarize_if(is.numeric, sum, na.rm=TRUE)
# change the date columns into actual date objects
col_names_dates <- str_replace_all(str_remove(colnames(covid_TS_NY)[7:ncol(covid_TS_NY)], 'X'), coll('.
dates <- as.Date(col_names_dates, format = "%m/%d/%y")
# extract the time-series for cases
covid_cases_NY <- as.numeric(covid_TS_NY[,7:ncol(covid_TS_NY)])
# take cases larger than 70 total infected as local infections
local_cases_NY <- covid_cases_NY[covid_cases_NY > 70]
# take cases less than 70 as imported (from outside NYS) infections
imported_cases_NY <- covid_cases_NY[covid_cases_NY <= 70]
# pad both arrays with 0s to make them the same length
local_cases_NY <- as.numeric(padarray(local_cases_NY, padsize=c(0,length(dates)-length(local_cases_NY))
imported_cases_NY <- as.numeric(padarray(imported_cases_NY, padsize=c(0,length(dates)-length(imported_c
# create a dataframe with dates, local, and imported columns
covid_cases_NY.df <- data.frame(dates=dates, local=local_cases_NY, imported=imported_cases_NY)
# take only cases after the first 41 rows since there were 0 infected at that time
covid_incidence_NY.df <- covid_cases_NY.df[41:nrow(covid_cases_NY.df),]
# compute the incidence (number of new cases) for both imported and local cases
covid_incidence_NY.df$local <- ave(covid_incidence_NY.df$local, FUN=function(x) c(0, diff(x)))
covid_incidence_NY.df$imported <- ave(covid_incidence_NY.df$imported, FUN=function(x) c(0, diff(x)))
# quick fix for this specific value since it is the cutoff point between local and imported
covid_incidence_NY.df$local[6] <- covid_incidence_NY.df$local[6] - covid_cases_NY.df$imported[45]
covid_incidence_NY.df[covid_incidence_NY.df < 0] <- 0
head(covid_incidence_NY.df)
```

```
##         dates local imported
## 41 2020-03-02     0        0
## 42 2020-03-03     0        1
## 43 2020-03-04     0        9
## 44 2020-03-05     0       12
## 45 2020-03-06     0        8
## 46 2020-03-07    45        0
```

Now I calculate the estimated effective reproductive number (R_e) for COVID-19 in New York State. This is done by using the EpiEstim `estimate_R()` function. We then plot the time-series for incidence, the explored serial intervals, and R_e. We use a parametric serial interval curve with a mean of 3.96 and a standard deviation of 4.75.

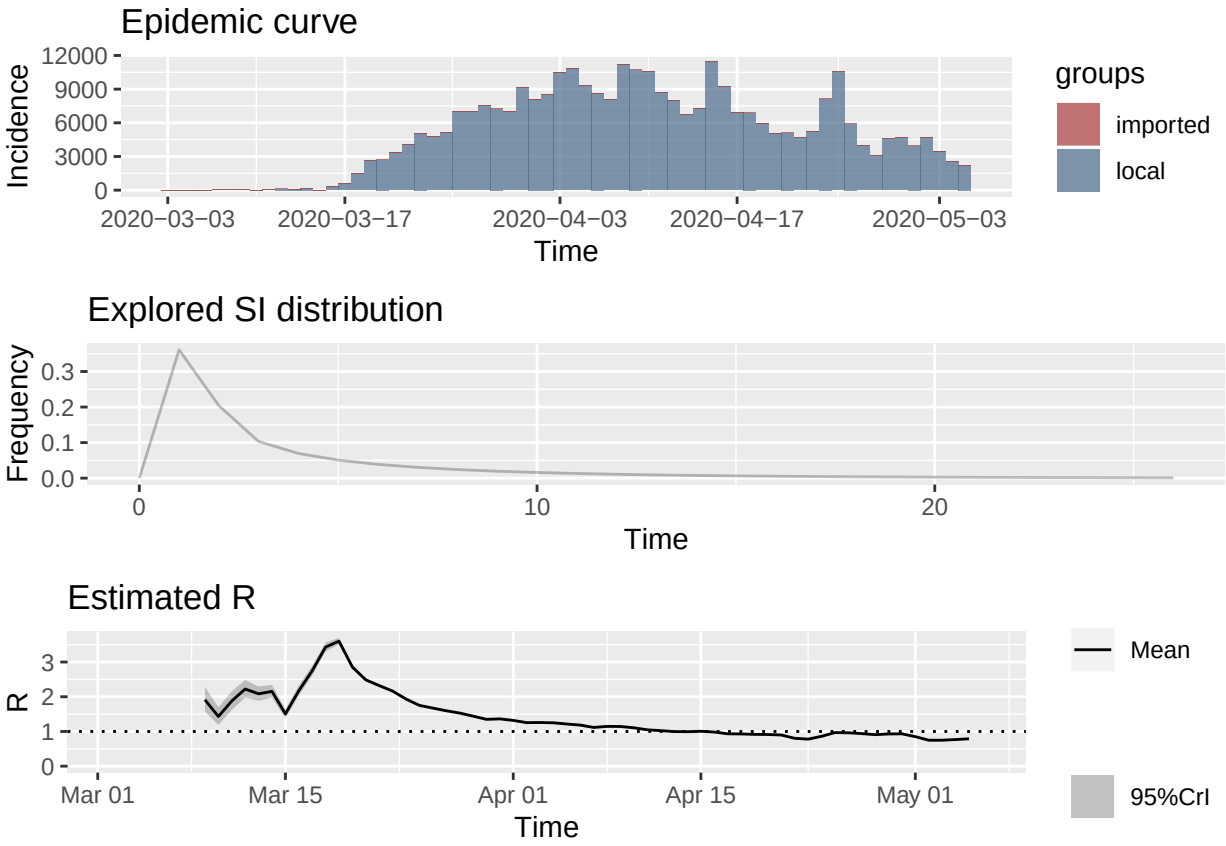
Source for serial interval: https://www.medrxiv.org/content/10.1101/2020.02.19.20025452v4

```
# set up plots so they can be viewed all at once
plot_Ri <- function(estimate_R_obj) {
    p_I <- plot(estimate_R_obj, "incid", add_imported_cases = TRUE)  # plots the incidence
    p_SI <- plot(estimate_R_obj, "SI")  # plots the serial interval distribution
    p_Ri <- plot(estimate_R_obj, "R")
    return(gridExtra::grid.arrange(p_I, p_SI, p_Ri, ncol = 1))
}
# calculate the curves using estimate_R function from EpiEstim package
NY_res_parametric_si <- estimate_R(covid_incidence_NY.df,
    method = "parametric_si", config = make_config(list(mean_si = 3.96, std_si = 4.75)))
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

```
# plot results
plot_Ri(NY_res_parametric_si)
```

```
## The number of colors (8) did not match the number of groups (2).
## Using `col_pal` instead.
```



Discussion

From the plots, it seems that the R_e for New York State currently hovers around the 1.0 line. This means that on average, one individual will transmit COVID-19 to one other individual. The goal for New York should be to get this number as close to zero as possible, although any R_e less than 1.0 is favorable. These plots also highlight what R_e was earlier in the timeline and show how social distancing and other epidemic policies put in place have reduced R_e over time. Overall, these results indicate that the response put in place by New York State is working and the epidemic curve will begin to flatten out.

Problem 2: Predicting Second Wave of COVID-19

The second problem I looked at was predicting when the second outbreak of COVID-19 may occur. This is under the assumption that it is inevitable that a second outbreak will occur. Second outbreaks in epidemics are sometimes worse than the initial outbreak (see https://www.history.com/news/spanish-flu-second-wave-resurgence). To prevent something like this happening again, it is important to try and determine a time-frame in which individuals and governments can prepare for a second wave of COVID-19. The results of my analysis indicate that the month of October is the most likely scenario in which a second wave will occur worldwide.

Please note that this result was determined based on the aspects and timelines of other diseases with similar outbreaks worldwide. There are many issues with this method of analysis that are described in the Discussion found below.

Methods

The problem of predicting a block of time in which COVID-19 will re-emerge can be done using classification. In this approach I first gathered data of other relevant diseases in terms of the features described above in the Data Description section. I started this analysis by designating each month of the year as a possible time block for a second wave. This is the class for prediction. Next, I used principal componenet analysis to get

a better understanding of which diseases were similar to begin with. Then, I created a model with linear discriminant analysis in order to predict which month of the year COVID-19 would fall into in terms of a second wave.

Results

I start by reading in the similar disease data (which includes COVID-19). I convert the categorical data found in `month_second_peak` to a factor. Lastly, the training data and training labels are created.

Here is another overveiw of what the data looks like:

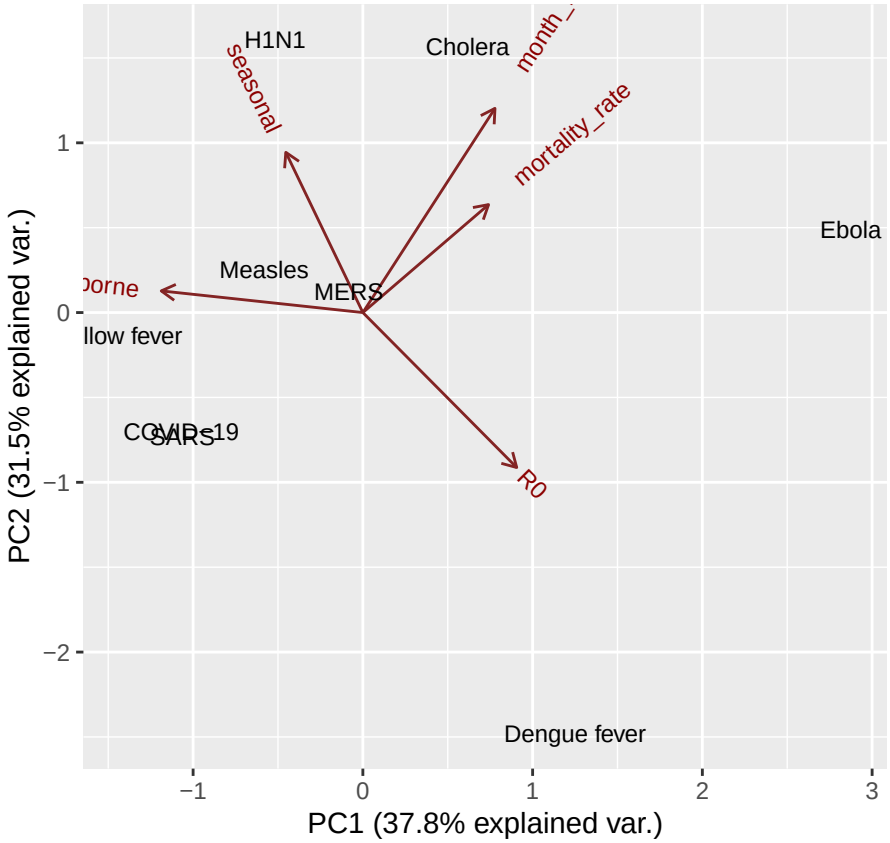
```
# read in the csv file containing the data
diseases <- read.csv('../data/csv/similar_diseases.csv')
# change rownames to match names of the disease
rownames(diseases) <- diseases[,'NAME']
# get rid of NAME column
diseases$NAME <- NULL
# change categorical data types to factors
months <- c('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'Oct
diseases$month_first_peak <- as.numeric(factor(diseases$month_first_peak, levels=months))
diseases$seasonal <- as.numeric(as.factor(diseases$seasonal))
diseases$airborne <- as.numeric(as.factor(diseases$airborne))
diseases$month_second_peak <- factor(diseases$month_second_peak, levels=months)
# take out COVID-19 feature set for prediction later
covid_features <- diseases['COVID-19',]
covid_features$month_second_peak <- NULL
# create training data and training labels
train_data <- diseases[-c(nrow(diseases)),names(diseases) != 'month_second_peak']
train_labels <- diseases[-c(nrow(diseases)),'month_second_peak']
diseases
```

```
##                   R0 mortality_rate month_first_peak seasonal airborne
## SARS           2.750          0.065                2        1        2
## MERS           4.275          0.344                4        1        2
## H1N1           1.480          0.264                7        2        2
## Measles       12.000          0.150                4        2        2
## Cholera        2.000          0.050               10        2        1
## Yellow fever   6.150          0.075                1        2        2
## Ebola         18.000          0.500                9        1        1
## Dengue fever  27.200          0.010                1        1        1
## COVID-19       2.400          0.070                2        1        2
##               month_second_peak
## SARS                        May
## MERS                       June
## H1N1                    October
## Measles                   April
## Cholera                   March
## Yellow fever            January
## Ebola                  November
## Dengue fever            October
## COVID-19                   <NA>
```

I then scale the data and perform principal component analysis. The results are shown in the biplot.

```
# recreate dataframe with COVID-19 features
train_data_pca <- rbind(train_data, covid_features)
# scale numeric data
```

```
scaled_train_data <- scale(train_data_pca)
# perform PCA on scaled training data
my.pca <- prcomp(scaled_train_data, retx=TRUE)
# create biplot
ggbiplot(my.pca, choices=1:2, labels=rownames(train_data_pca), scale=0)
```



Next, I perform LDA and see how the training data prediction compare to the actual in a table.

```
# perform LDA on data
lda.fit <- lda(train_data, grouping=train_labels)
```

```
## Warning in lda.default(x, grouping, ...): groups February July August September
## December are empty
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

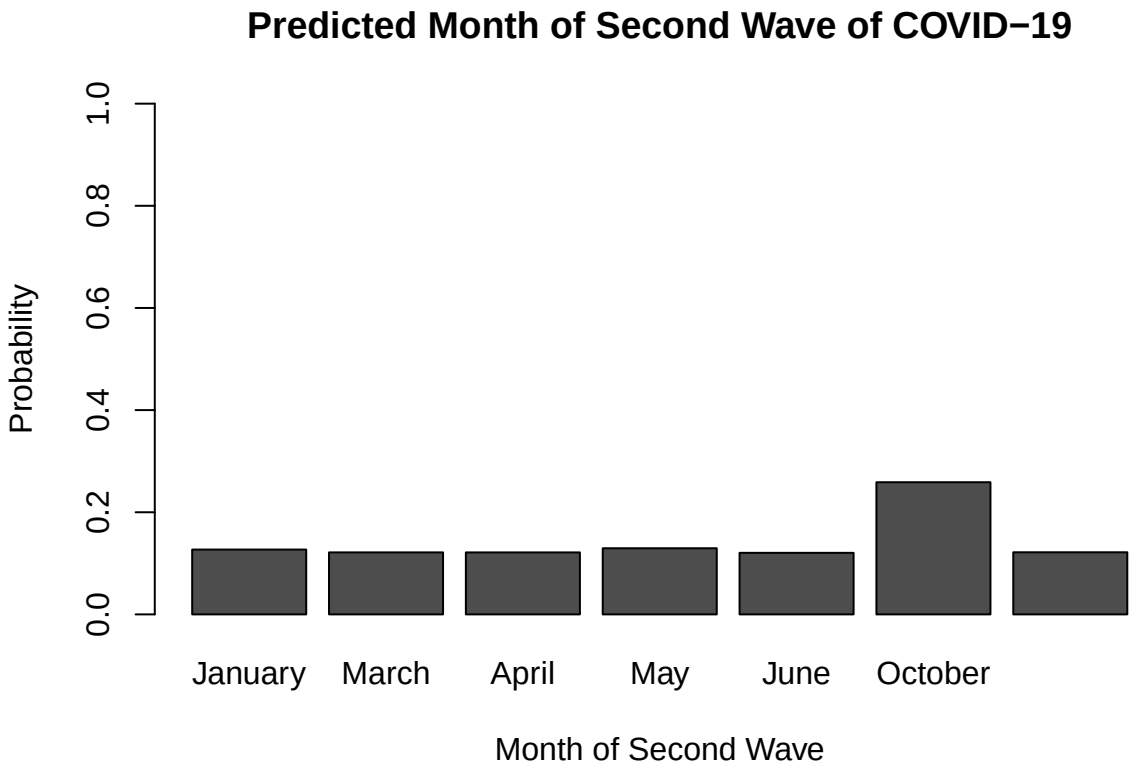
```
# get accuracy on training data
train_preds <- predict(lda.fit, train_data)$class
confusion.matrix <- table(train_labels, train_preds)
kable(confusion.matrix, type="html",digits = 2)
```

	January	February	March	April	May	June	July	August	September	October	November	De
January	0	0	0	0	0	0	0	0	0	1	0	
February	0	0	0	0	0	0	0	0	0	0	0	
March	0	0	0	0	0	0	0	0	0	1	0	
April	0	0	0	0	0	0	0	0	0	1	0	
May	0	0	0	0	0	0	0	0	0	1	0	
June	0	0	0	0	0	0	0	0	0	1	0	

	January	February	March	April	May	June	July	August	September	October	November	De
July	0	0	0	0	0	0	0	0	0	0	0	
August	0	0	0	0	0	0	0	0	0	0	0	
September	0	0	0	0	0	0	0	0	0	0	0	
October	0	0	0	0	0	0	0	0	0	2	0	
November	0	0	0	0	0	0	0	0	0	1	0	
December	0	0	0	0	0	0	0	0	0	0	0	

I use the LDA model to predict the month in which COVID-19 will have a second wave. The probabilities for each month (only months > 0) are shown in the barplot.

```
# predict outcome for COVID-19
covid_pred <- predict(lda.fit, covid_features)
barplot(covid_pred$posterior,
        ylab="Probability",
        xlab="Month of Second Wave",
        ylim=c(0, 1.0),
        main="Predicted Month of Second Wave of COVID-19")
```



Discussion

From the principal component analysis biplot, we can see that COVID-19 is most similar to the SARS outbreak of 2003. This is expected since they are both coronaviruses. However, it is interesting to note how changing the seasonality of COVID-19 as more research is completed may shift COVID-19's placement towards H1N1 and Yellow fever.

The table of predictions above highlights some issues with the dataset. First of all, there are too few diseases used in this analysis to provide accurate results, especially for a classification task of this magnitude. All of the predictions place the second wave of each disease in October. This is most likely due to the small

number of data points as well as the limitations of the selected features. These features were chosen as to what I thought would be good factors for determining the time-frame for a second wave of COVID-19. Better features would look into the biology and chemistry of each disease. A better result would most likely result from using the protein sequences of each disease and comparing them that way for instance.

Lastly, the probabilities outlined above in the barplot show that October is most likely the month in which COVID-19 will have its second wave. This outcome does make sense from the perspective that the disease's infection rate may decrease as temperature increases, but we did not consider that possibility in this analysis directly. So, what this outcome really means is that based on other epidemics' second wave, it is most likely that COVID-19 will have a second wave in October.

Summary and COVIDMINDER Recommendations

The first key insight into COVID-19 that result from this analysis are that the R_e is hovering around the 1.0 line, which is a good sign for New York State. This shows that whatever we are doing to keep the disease from spreading is working for the most part.

The second is that October is the month in which we should be planning for when COVID-19 may come back in a second wave. Individuals and governments should plan ahead for the Fall and be ready when the time comes. More research into this prediction is necessary to achieve a more accurate analysis in this regard, however.

I believe that the first problem I addressed in this analysis could be included in COVIDMINDER. It could be a useful insight into how government and individual actions are limiting the spread of the virus in New York State. The second problem I addressed should be researched more before including in COVIDMINDER due to the problems with the small dataset and feature selection.

References

- https://www.medrxiv.org/content/10.1101/2020.02.19.20025452v4
- EpiEstim package
- https://www.cdc.gov/about/history/sars/timeline.htm
- https://www.hindawi.com/journals/av/2011/734690/
- https://wwwnc.cdc.gov/eid/article/26/2/19-0697_article
- https://www.ncbi.nlm.nih.gov/pubmed/31813836
- https://ajph.aphapublications.org/doi/pdf/10.2105/AJPH.2013.301704r
- https://www.cdc.gov/h1n1flu/surveillanceqa.htm
- https://www.ncbi.nlm.nih.gov/pubmed/28757186
- https://journals.plos.org/plosntds/article/file?rev=2&id=10.1371/journal.pntd.0006158&type=printable
- https://www.ncbi.nlm.nih.gov/pubmed/27846442
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3381442/
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4397933/
- https://www.who.int/emergencies/mers-cov/MERS-epicurve-July-2019.png