1

STAT GR 5291
Advanced Data Analysis
Final Project Report
COVID-19 Analysis

Group 2

Linjun Huang (lh2985)
Yijin Wang (yw3479)
Chui Kong (ck2964)
Shengxi Jin (sj3002)
Xuhao Li (xl2900)
Jiongdao Zhou (jz3179)
Daijie He (dh2981)
Xinyi Wei (xw2684)
Jun Ma (jm4967)
Yexing Liu (yl4378)

Columbia University
Department of Statistics

2

Contents

3

1. Introduction

In December 2019, when humans were saying goodbye to the past and preparing for the new year, a novel coronavirus was identified in Wuhan, a city in the Hubei Province of China. These cases in Wuhan were thought to be the cause of a cluster of pneumonia cases and rapidly spread throughout China, followed by a global pandemic. In February 2020, the World Health Organization designated the disease COVID-19 formally. This epidemic is unprecedented in human history, and its impact far exceeds that of the European Black Death or the Asian SARS. The global economy has been hit hard and thus the lives of people around the world are restricted.

The data is cold. To be honest, it is distressing to count confirmed cases and death cases. But we hope to reflect this difficult time through data analysis. Through our data analysis, we hope to alert people to cherish their time and lives.

2. Objectives

Having a good model to predict the trend of COVID-19 is a sufficient way to prevent the spread of this pandemic. The objective of this project is to provide some insights about the COVID-19 transmission from a data-centric perspective. The observations and conclusions from the data exploration only depends on the current official COVID-19 data. So, there are some unexplainable outliers in the data. Our main reference is [2]. [2] provides the idea and methods about using linear regression model, time series model, and SIR model to predict the transmission of COVID-19 mainly in the early stage. [1] and [3] show many different methods of applying data visualization and the interpretations of COVID-19 data.

In this project, we mainly focus on using different predictive models to see how accurate they are in predicting the growth of the COVID-19 in the early stage. We will first do exploratory data analysis on the COVID-19 global tendency to see how the pandemic evolution. We will forecast the COVID-19 for the early stages of the transmission by applying Linear Regression and Time Series Regression to see how accurate the prediction could be. We will also use the most famous epidemiologic model: SIP to do the prediction. The SIR is a simple model that considers a population that belongs to one of the states: Susceptible (S), Infected (I), and Recovered (R). Finally, we will determine which factors impact the transmission behavior significantly by analyzing the relationship between the reported cases and the factors like GDP, medical conditions, and healthy life expectancy. This analysis could help us understand what are the key factors that impact the COVID-19 transmission

4

3. Materials and Methods

In this project, we will use the following datasets:

- COVID-19 Global Forecasting(Week 4) Dataset[1]
- Population by Country 2020[2]
- WHO health information[3]
- WHO COVID-19 dataset[4]
- World happiness report [5]
- COVID-19 Data Visualization and Prediction[6]

Methods:

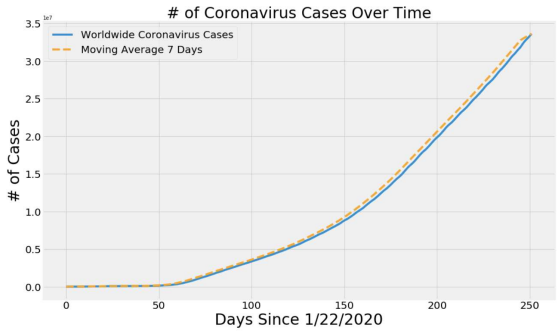
- Linear Regression Model
- Time Series Model
- SIR Model
- Epidemiologic Model: SIP

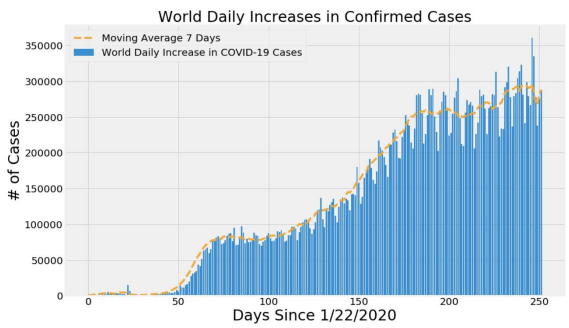
4. EDA and data visualization

In the first section, we do exploratory data analysis on the global COVID-19 to see some situations of different typical countries and regions, by following the pandemic evolution. [6]

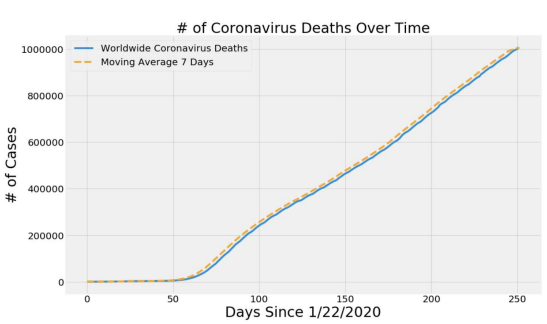
4.1. Exploring Global Coronavirus Cases

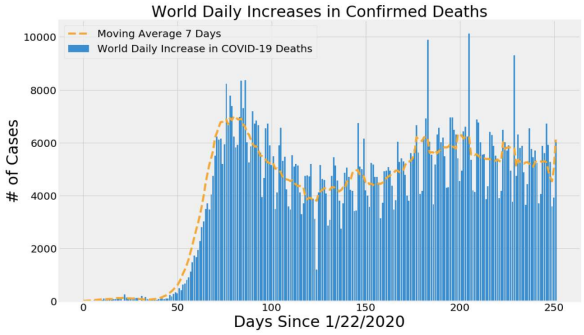
We have three data sets, the global number of people who have confirmed, dead, recovered cases of coronavirus since the beginning of 2020. We also calculate world daily increases in three cases to compare and we can see that the three trends are increased.

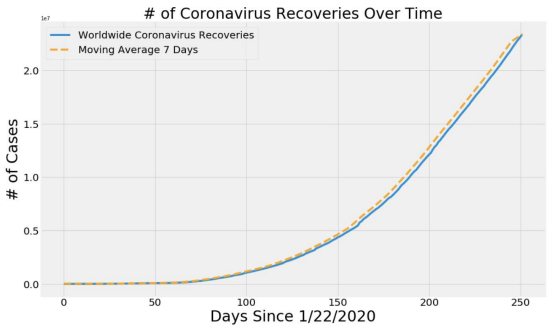


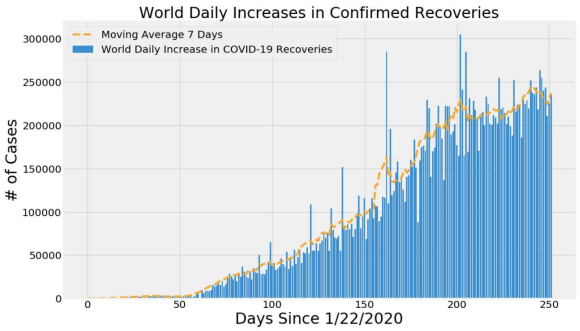


5

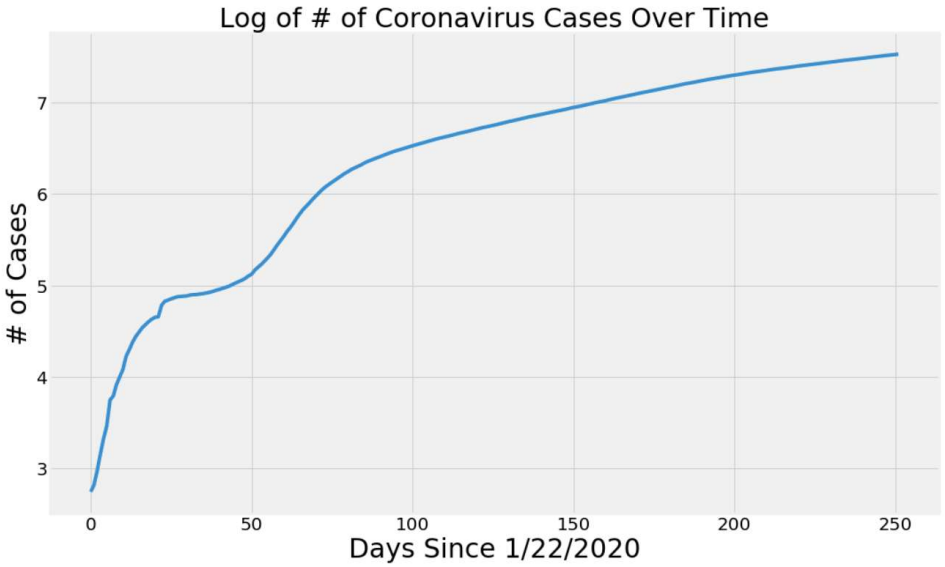




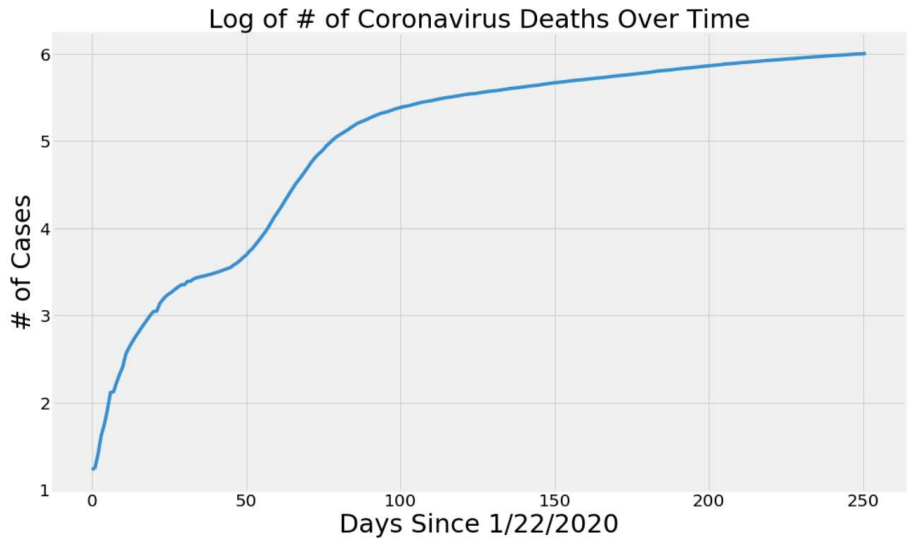


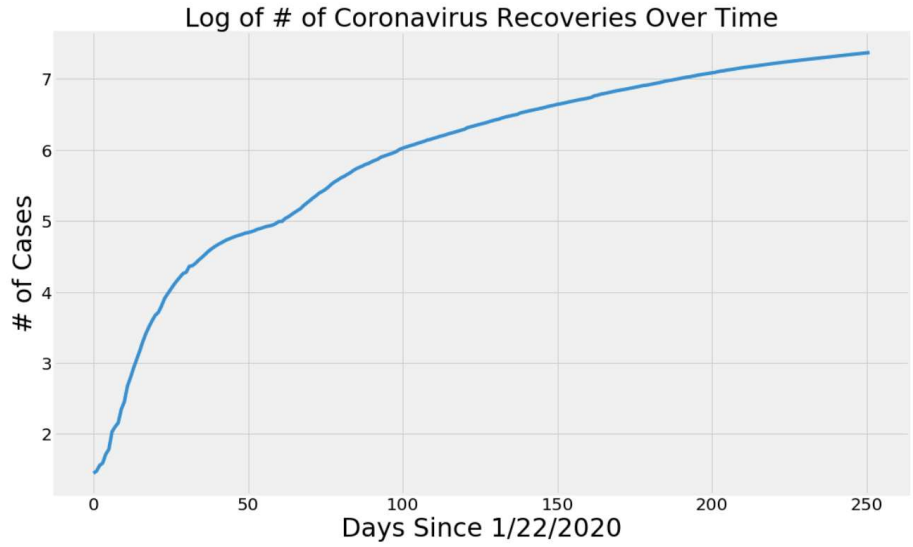


More clearly ,we transformed data by log transformation. From the graphs below, we can conclude that the situations are very serious around the world in Confirmed, Death and Recoveries Cases.



6

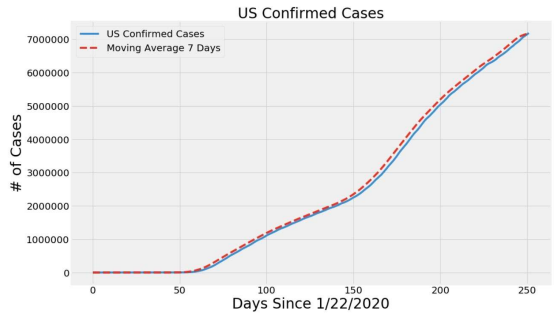


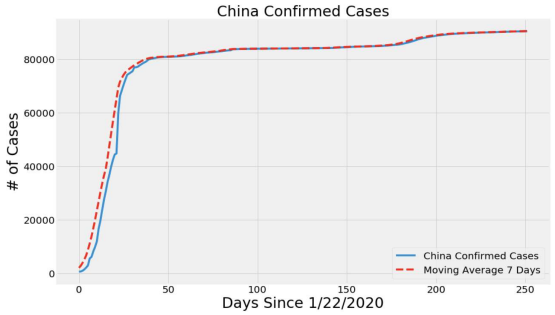


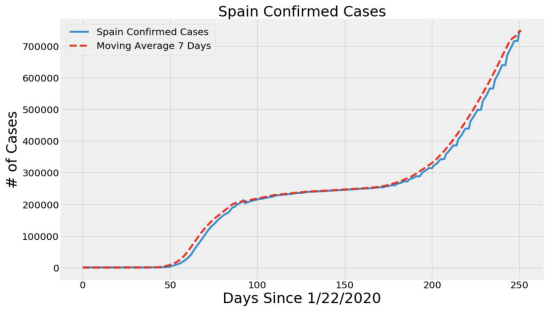
4.2. Exploring Coronavirus Cases From different Countries

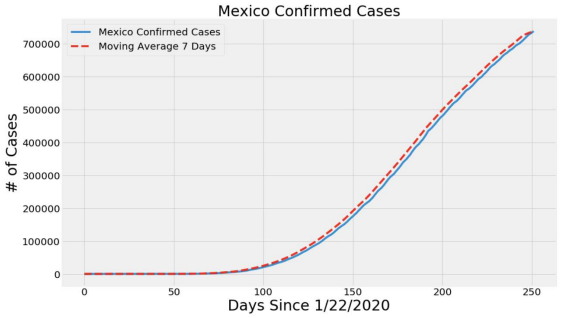
We select some typical countries in the world to analyze the pandemic in detail. Here are confirmed cases of the USA, China, Spain and Mexico. Those four countries in different continents have different trends, and thus we will explore these countries in the following.

7

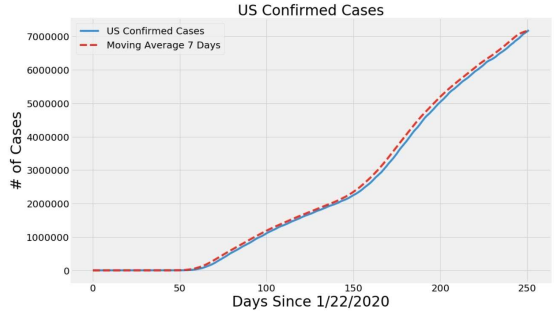


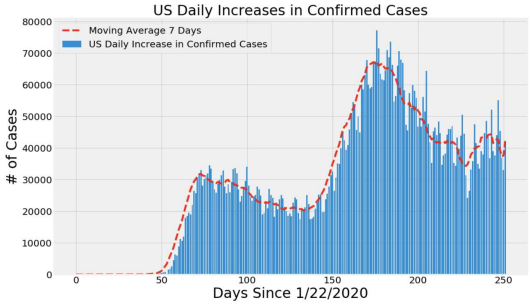


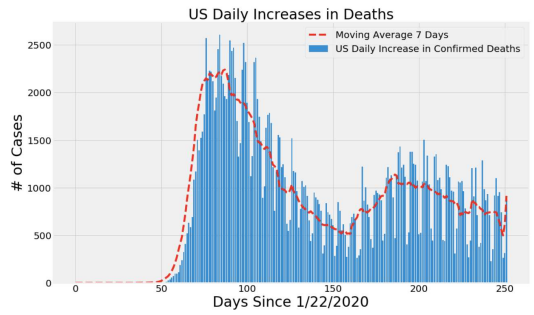


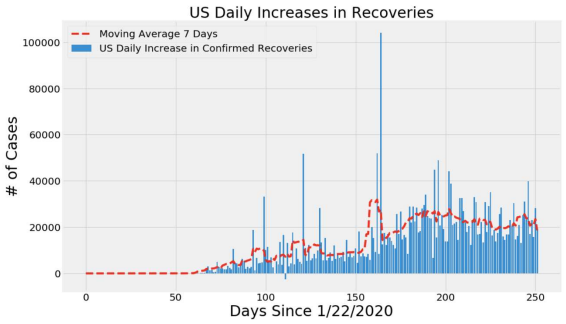


For the United States, The trend is increased about the daily increase in confirmed cases, death and recoveries. The increase in the number of daily deaths reached its peak in around 100 days. Then the government began to close public areas and encourage schools to let students take classes online, and the epidemic was alleviated but still not controlled.



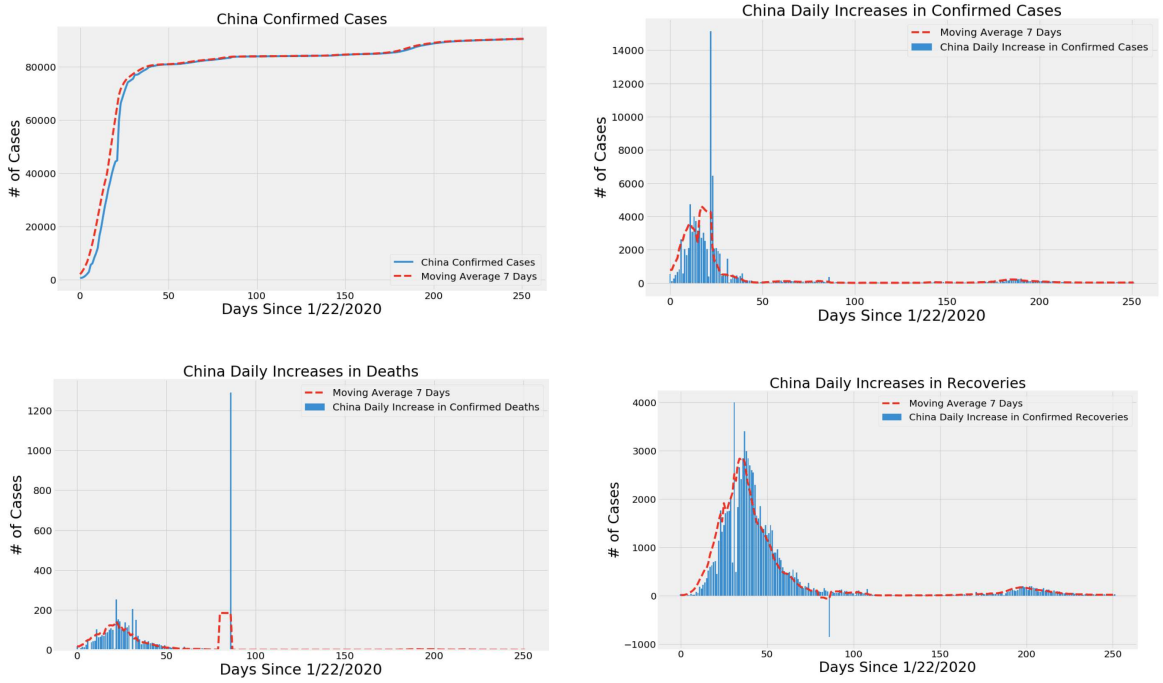






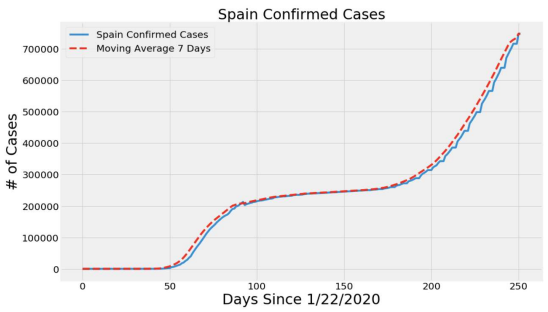
8

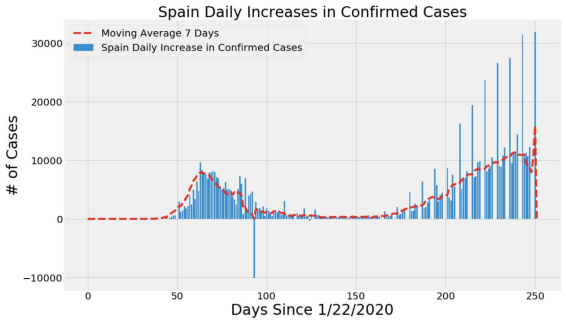
For China, in the first forty days, when it rose almost linearly, it slowed down rapidly and the daily growth tended to zero. Since March, Chinese government has taken actions to prevent the future outbreak like shut down factories, cities and achieved the results. On the graph ‘Daily Increases in Deaths’, there is a significant surge, we think about two explanations. Firstly, it is possible that it is the outlier of our data. Secondly, it is possible that on that day, the Chinese government started counting the previous increased death cases together. We believe the second one is more reasonable. However, there exists the outlier on the graph ‘Daily increases in Recoveries’ because the increased case cannot be negative. There is a small triangle area around 200 days on the Recovery graph, we think that’s possible because effective vaccines had not yet been invented at the time, a small number of people still have the possibility of relapse.

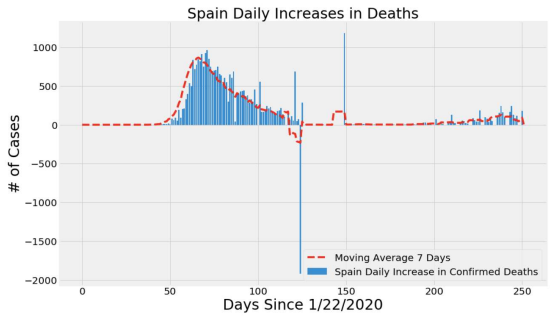


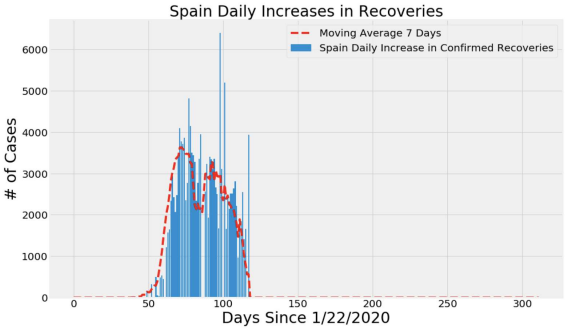
The trend of the epidemic in Spain is similar to that in other European countries. After a moderate level(between 100 and 180 days since 01/22/2020), the confirmed cases have increased sharply. People move closely among European countries, so the epidemic situation is similar. We suppose one reason is because governments in Europe have loosened supervision after controlling the epidemic, and the other is because the virus has changed and thus the situation has become more serious. But according to the news, the reason is governments stopped counting the number of coronavirus cases but the situation is more and more serious and governments have to value the problem. There also exists some outliers on the graph (the negative data).

9

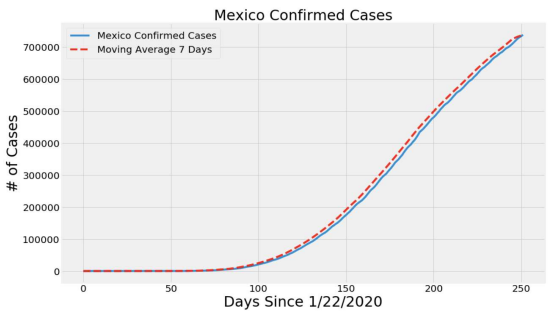


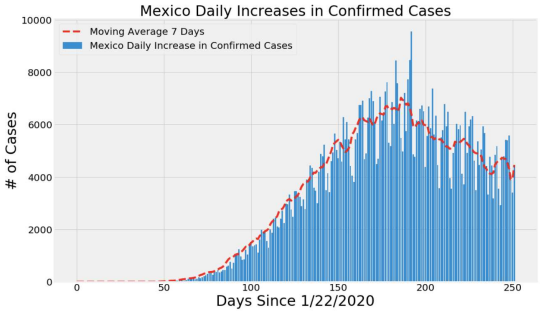


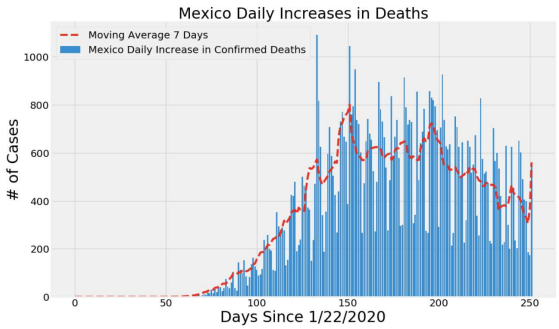


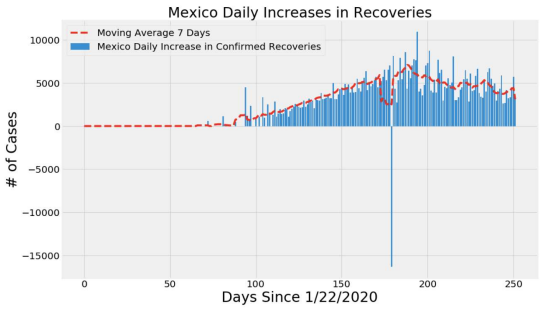


The final country we select is Mexico, because it represents many countries in South America and South Asia. After the outbreak of the epidemic, due to weak government supervision, there is basically a linear increase.





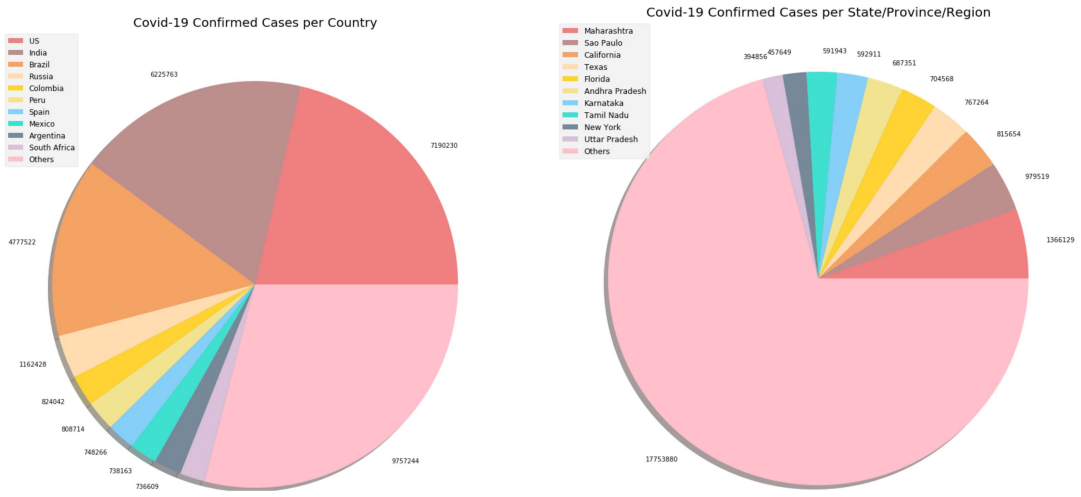




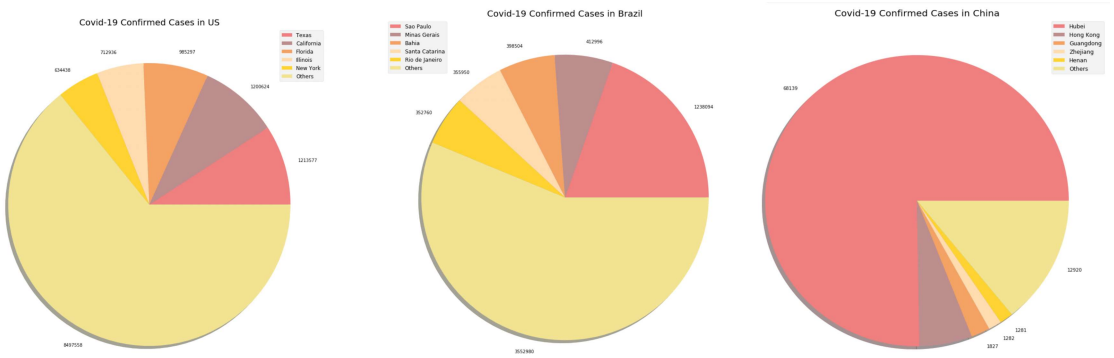
10

4.3. Pie Charts

The pie chart usually will give us the information of each category in whole. The first plot shows the confirmed cases per country, and it shows that the U.S, India and Brazil occupy more than 50% of total cases. The second graph is confirmed cases per state/province region in the world. It shows that Maharashtra (India), São Paulo (Brazil), California(US) are the top three states which have the most cases globally.



And then we take a closer look at the situation in the U.S, Brazil and China which are affected a lot by COVID-19. The pie charts below show that the confirmed cases of the U.S and Brazil distributed more equally among states. However, in China, most cases occur in HuBei Province. The situation in China may be caused by Chinese government giving the lockdown order immediately after the outbreak of the epidemic in Hubei.

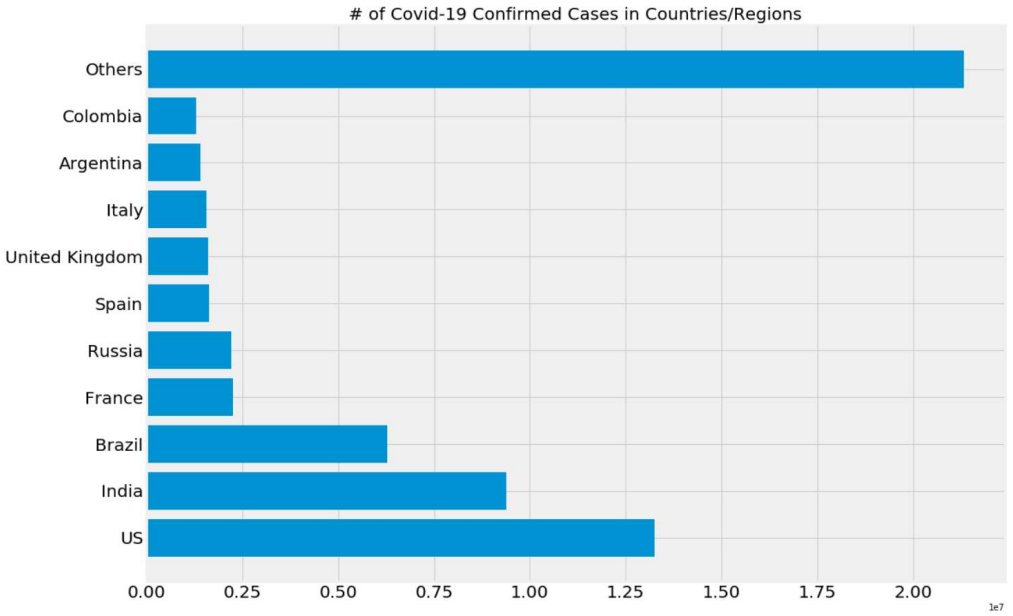


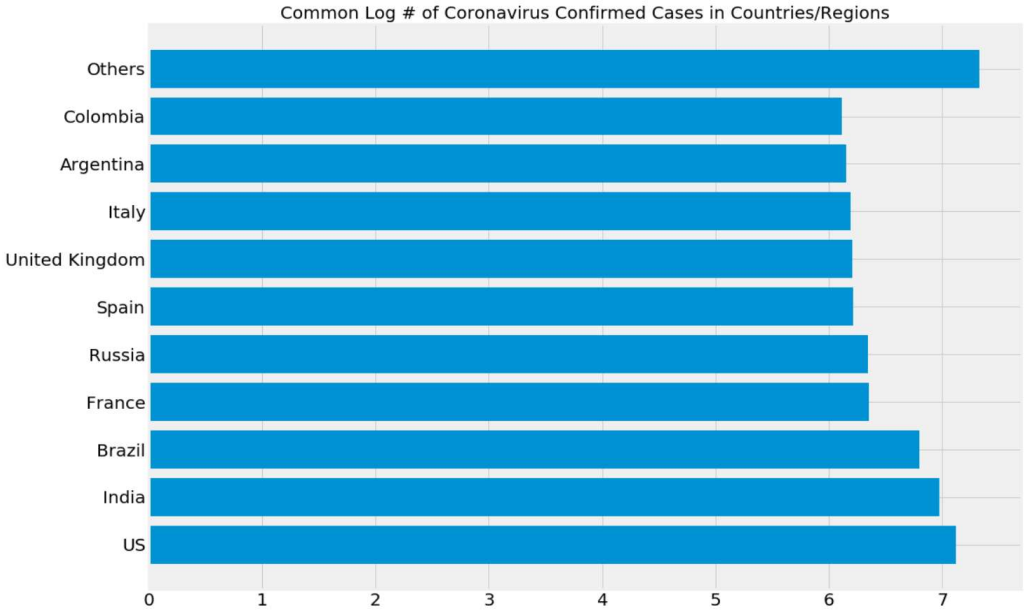
4.4. Bar Charts

Bar graphs usually show absolute values or proportions for each of the categories. let's

11

see some general facts in the world. The below two bar charts show global confirmed cases. The U.S, India and Brazil have the most cases. Compared to the chart plot, the bar plot could give us more clear information about the number of confirmed cases around the world. The log plot beside could make us see more clearly which country and state suffer most.

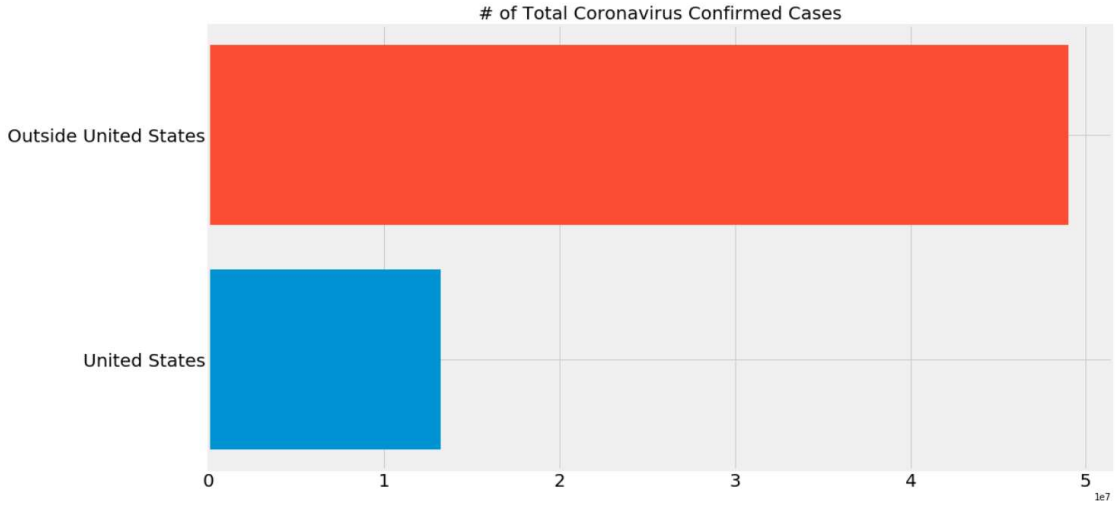


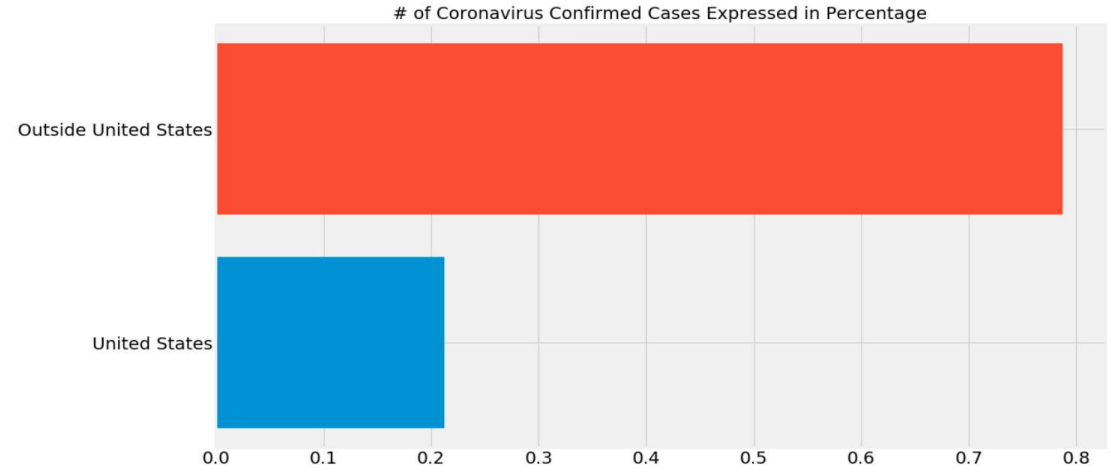


Now the U.S has the most confirmed cases in the world, so we also take a lot at the proportion of confirmed cases in the U.S and outside the U.S. From the plot below, we can see

12

that the U.S cases have about 22% of total cases, and the outside U.S cases have about 78% of total cases. COVID-19 affects a lot of people in the U.S.

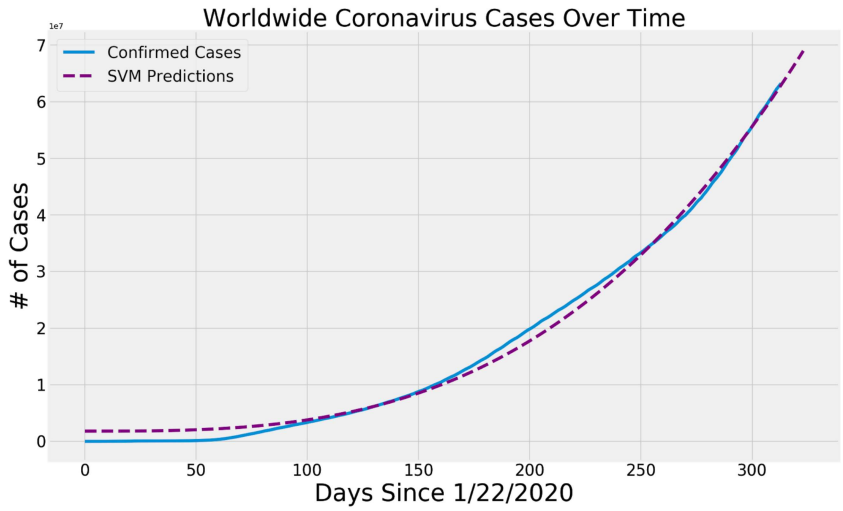


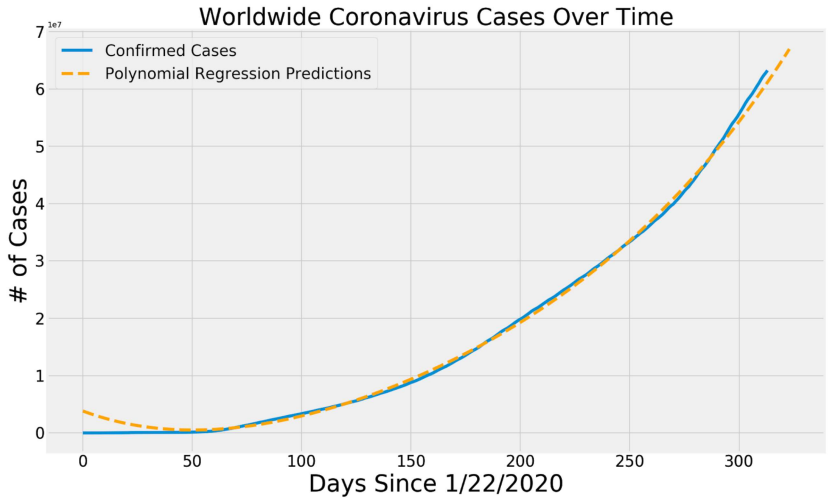


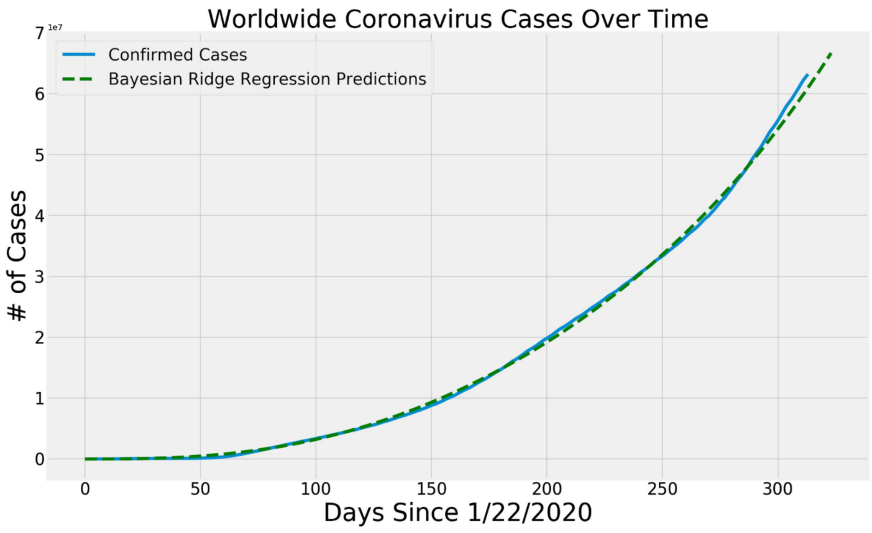
4.5. Prediction for confirmed cases

We use SVM, Polynomial Regression and Bayesian Ridge Regression to do the predictions on the worldwide confirmed cases relatively.

13


Worldwide Coronavirus Cases Over Time
Confirmed Cases
SVM Predictions
# of Cases
Days Since 1/22/2020

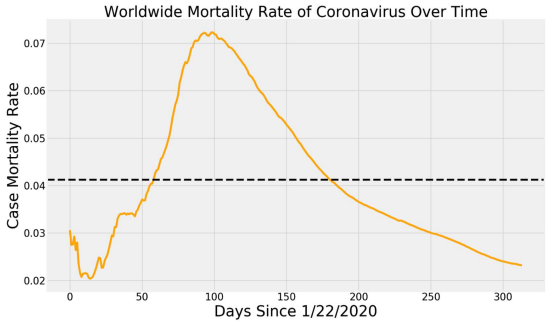

Worldwide Coronavirus Cases Over Time
Confirmed Cases
Polynomial Regression Predictions
# of Cases
Days Since 1/22/2020

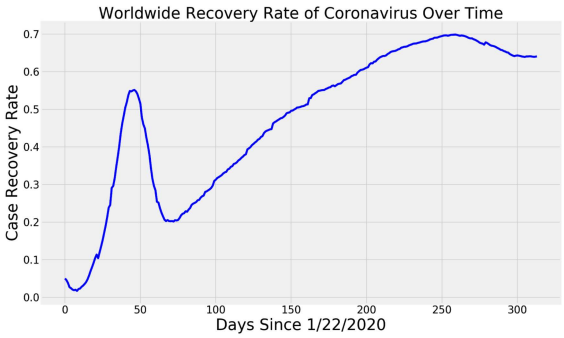

Worldwide Coronavirus Cases Over Time
Confirmed Cases
Bayesian Ridge Regression Predictions
# of Cases
Days Since 1/22/2020

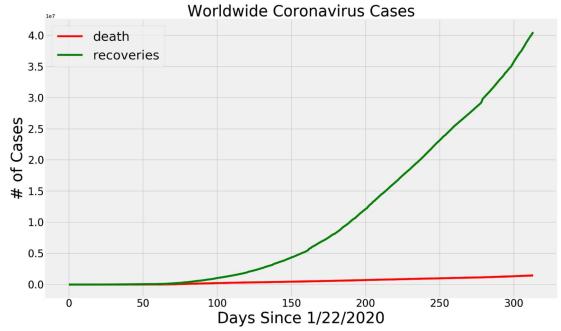
14

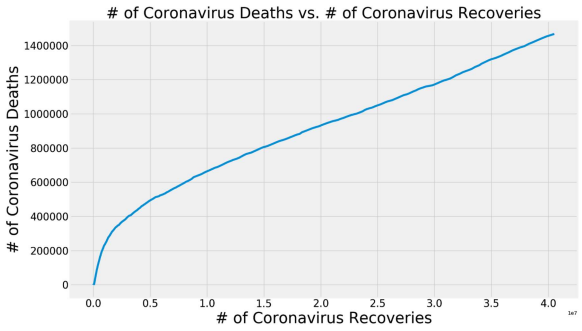
In SVM we choose ‘poly’ to be the kernel given that the result of ‘rbf’ is more likely to be overfitting. The prediction data on confirmed cases of SVM is higher, reaching 6.898e + 07 within 10 days after the establishment of the model. However, the prediction data of the other two models are still below 6.7e + 07 in 10 days. Each of the model’s predictions shows an upward trend with positive curvature.

4.6. Mortality rate and deaths against recoveries









From the plots, the mortality rate increased in the first 91 days, then decreased after a period of fluctuation. Which can be explained by the gradually mature treatment method and sufficient medical resources.

The recovery rate increased in the first 47 days, and then there was a steep decline(probably caused by the shortage of medical resources due to the shock increase of confirmed cases), then slowly recovered from the 68th day.

The slope deaths against recoveries decreased and remained stable over time, which shows that the worldwide medical situation is gradually keeping pace with the spread of the epidemic. Plots of part 4 is output by code from [6].

15

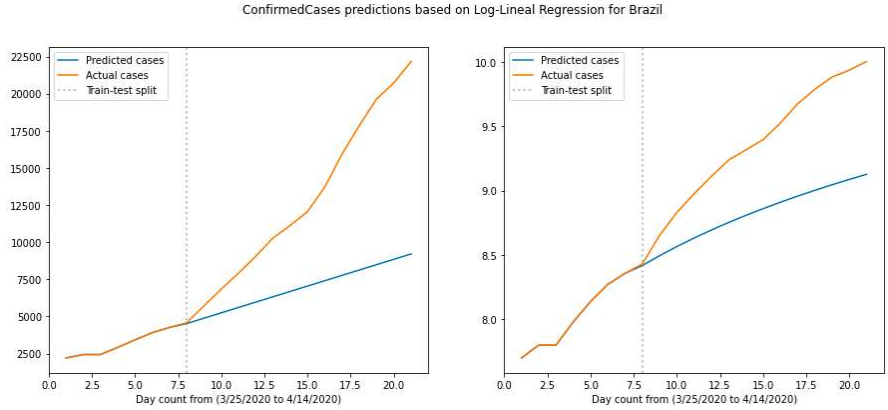
5. Predict trend using linear regression

In this section, we aimed to construct linear regression models to make predictions of the following 14 days for some of the selected countries by utilizing a period of the past dataset (14 days or 7 days).

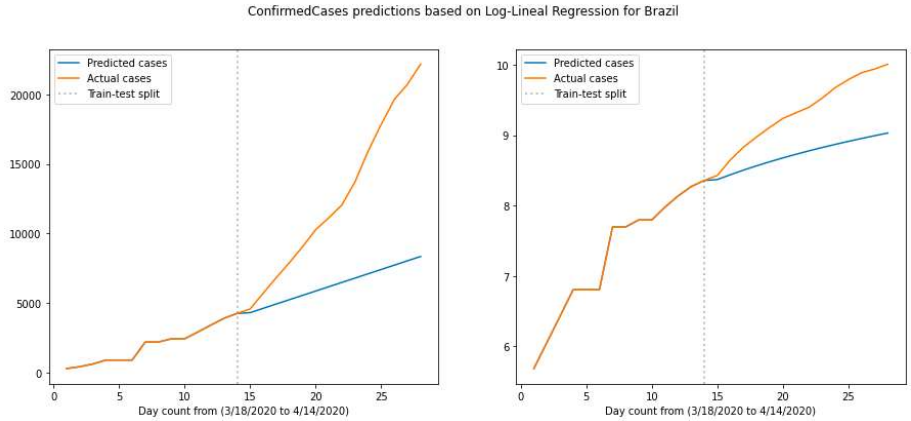
First, we choose Brazil as an example. Following picture on the left side is the linear regression model constructed with the past 7 days dataset and it is aimed to make predictions of the following 14 days. However, with careful consideration, we realized that a log transformation might be needed for model improvement. So we construct a new prediction model with log transformation and output picture is present on the right side. Indeed, the log-transformed model performs better than the linear regression model.

Also, we use the data of 14 days prior to the prediction period to do the regression. Likewise, a log-transformed model performs better than the linear regression model in this scenario. We want to see if a longer training dataset will give out a better prediction.

7 days model of Brazil:



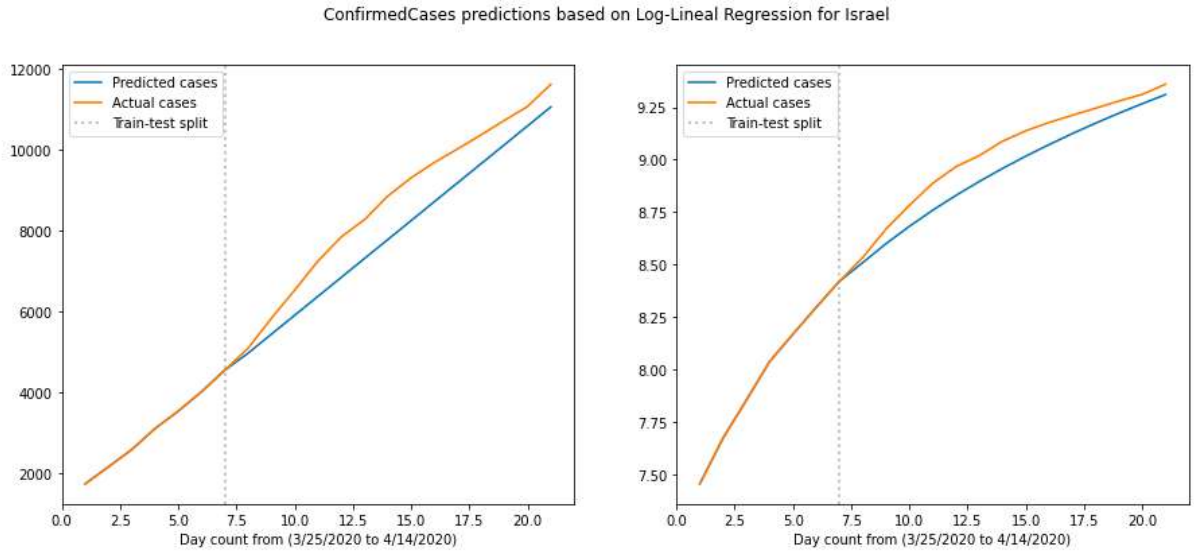
14 days model of Brazil:



16

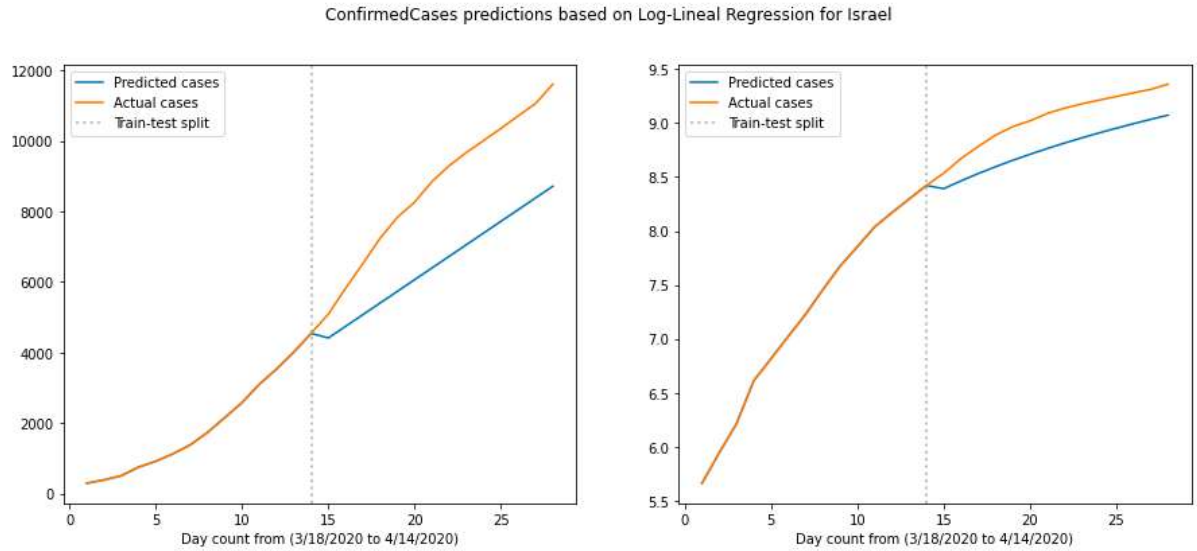
Similarly, we constructed linear regression models to do predictions of the cases in Israel, Italy, Japan, China, France and Germany, using prior 7 or 14 days training data with log transformation.

7 days model of Israel:



(Left: linear model Right: log-transformed model)

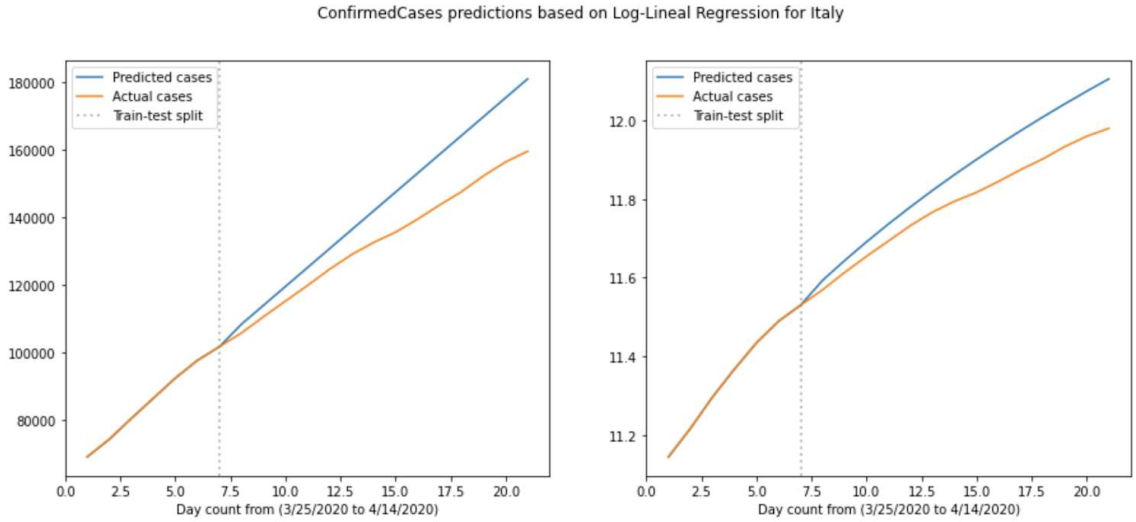
14 days model of Israel:



(Left: linear model Right: log-transformed model)

17

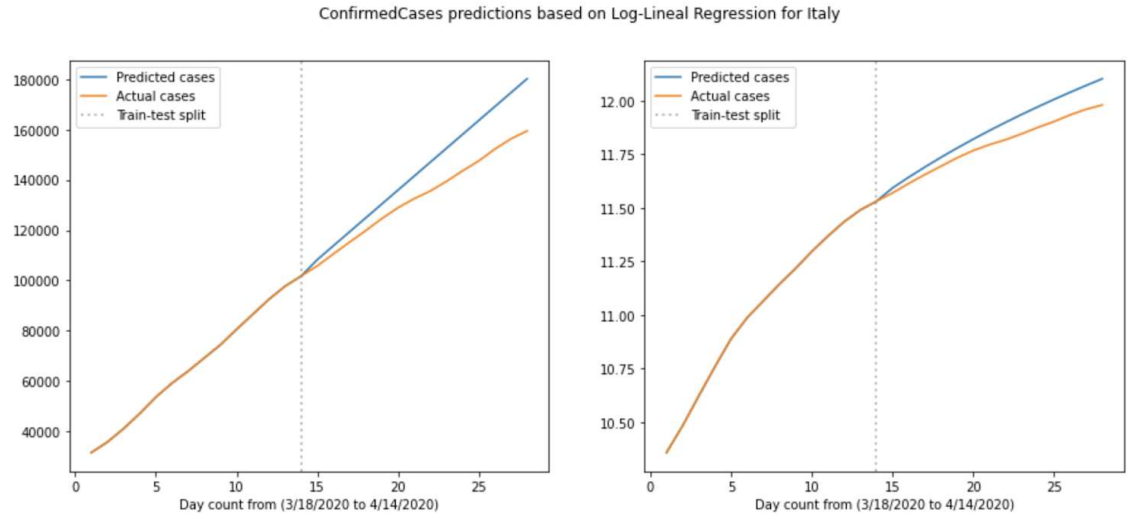
7 days model of Italy:



(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model does in this scenario.

14 days model of Italy:

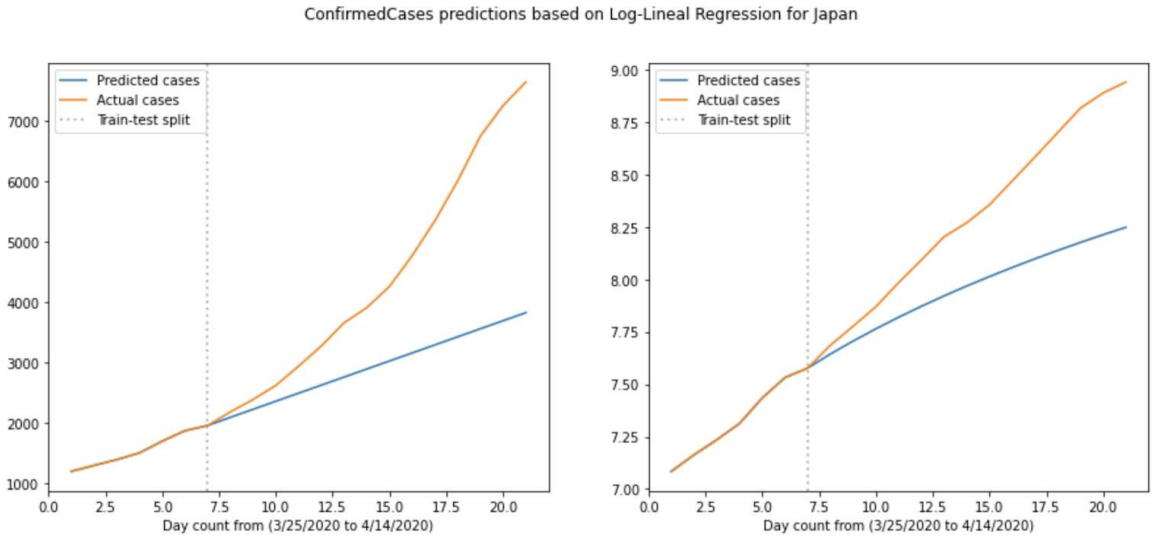


(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model does in this scenario.

18

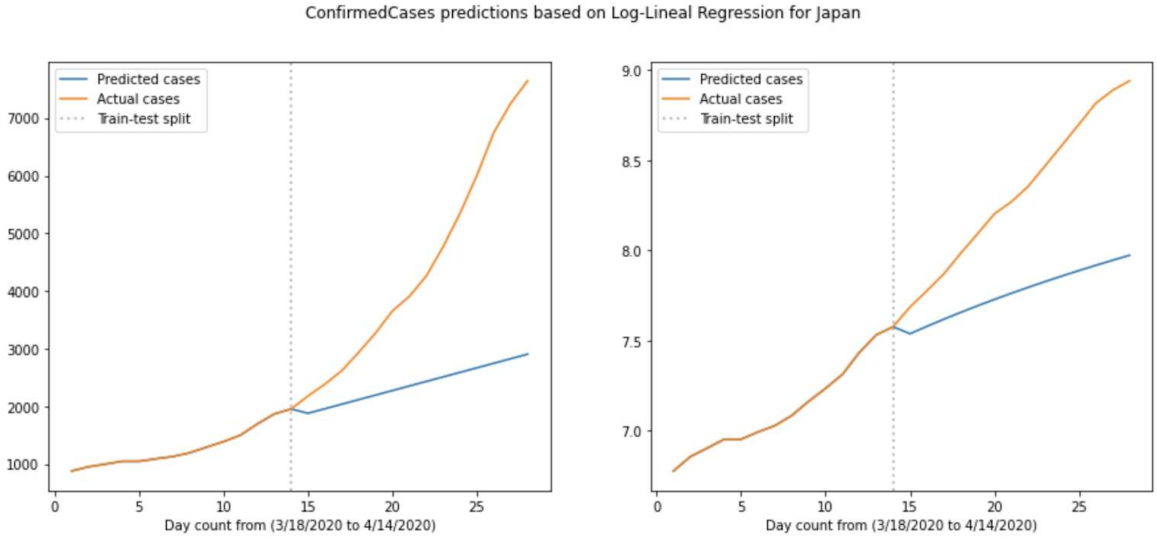
7 days model of Japan:



(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model does in this scenario.

14 days model of Japan:

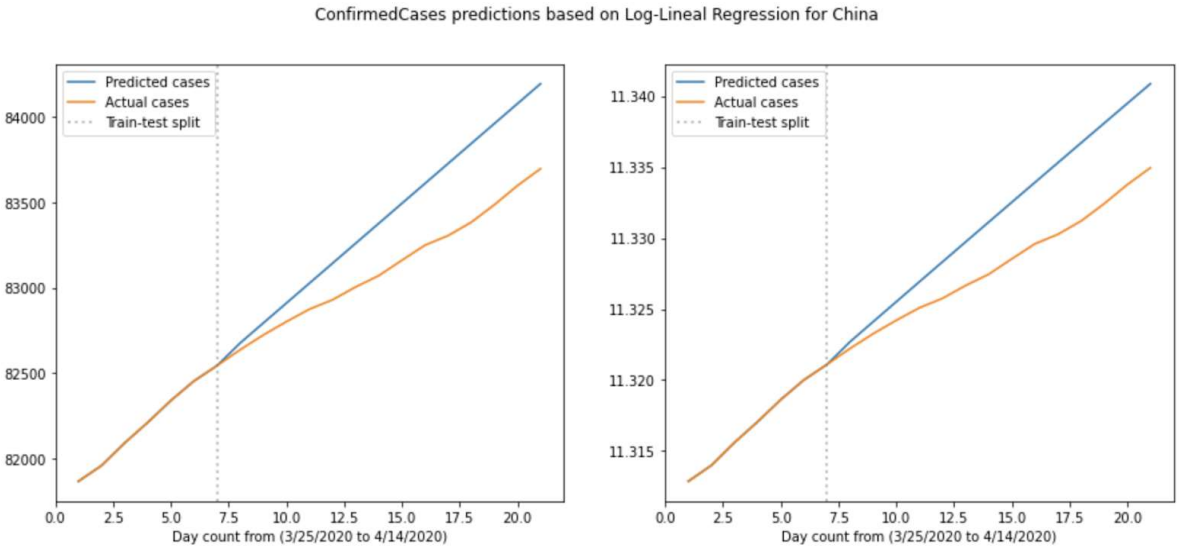


(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model does in this scenario.

19

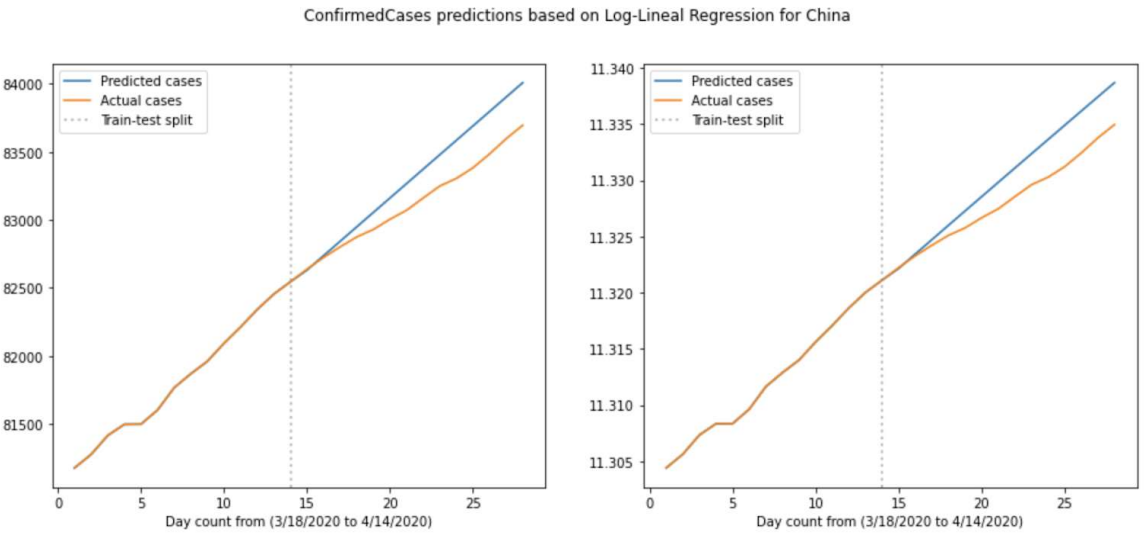
7 days model of China:



(Left: linear model Right: log-transformed model)

Unlike previous cases, a log-transformed model performs similarly as the linear regression model does in this scenario.

14 days model of China:

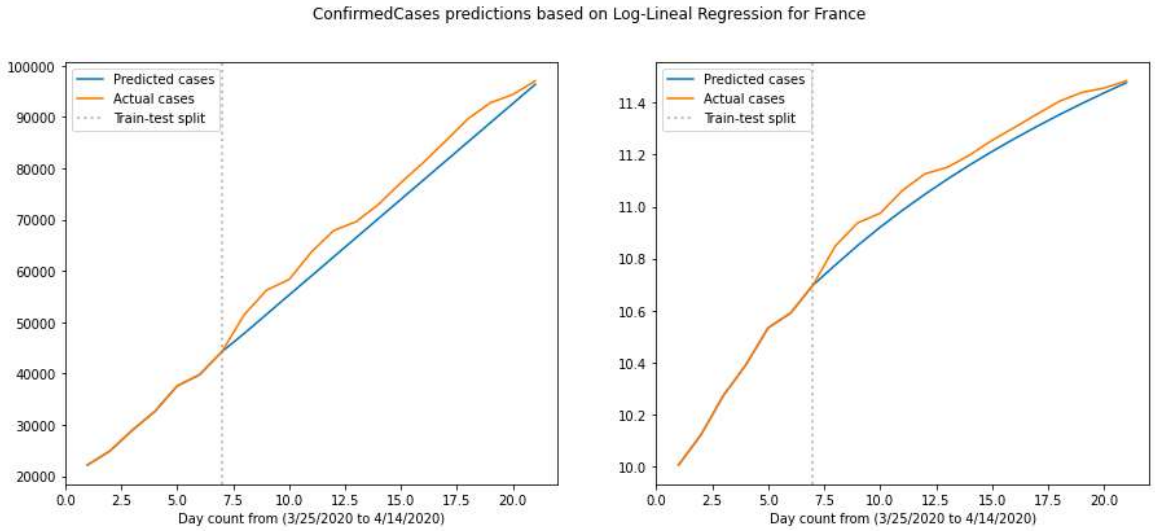


(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model does in this scenario.

20

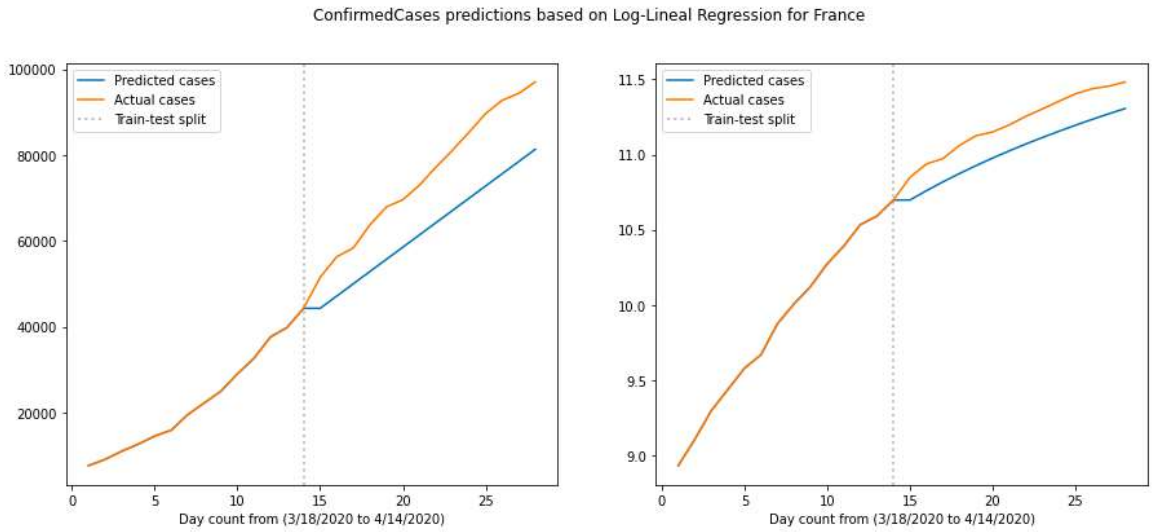
7 days model of France:



(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model in this scenario.

14 days model of France:

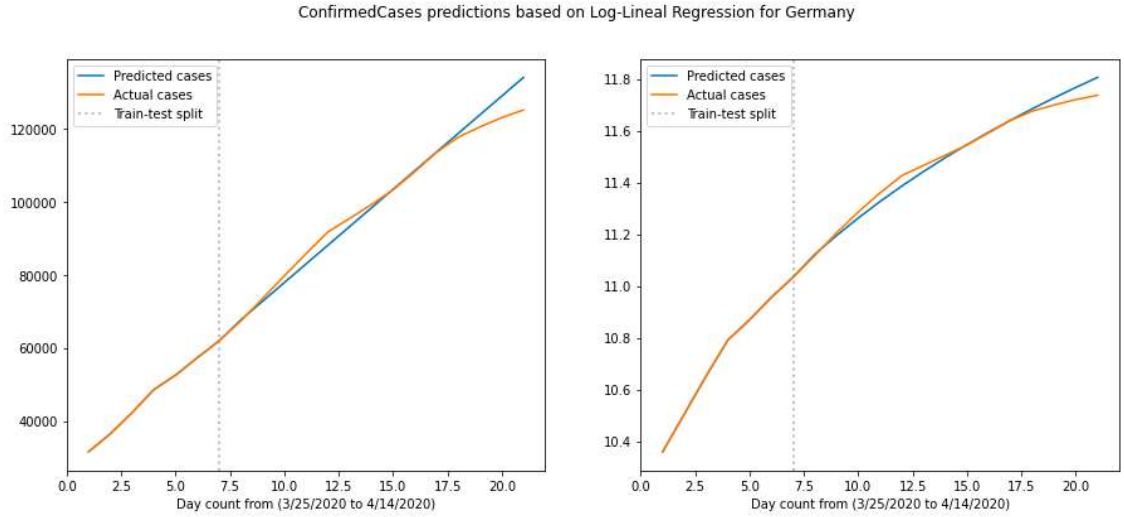


(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model in this scenario.

21

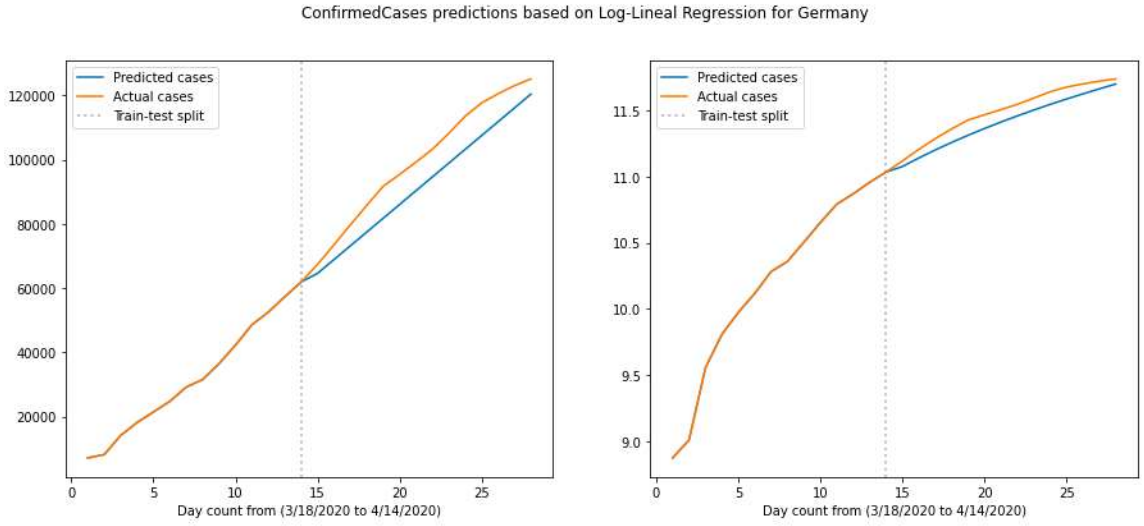
7 days model of Germany:



(Left: linear model Right: log-transformed model)

Likewise, a log-transformed model performs better than the linear regression model in this scenario.

14 days model of Germany:

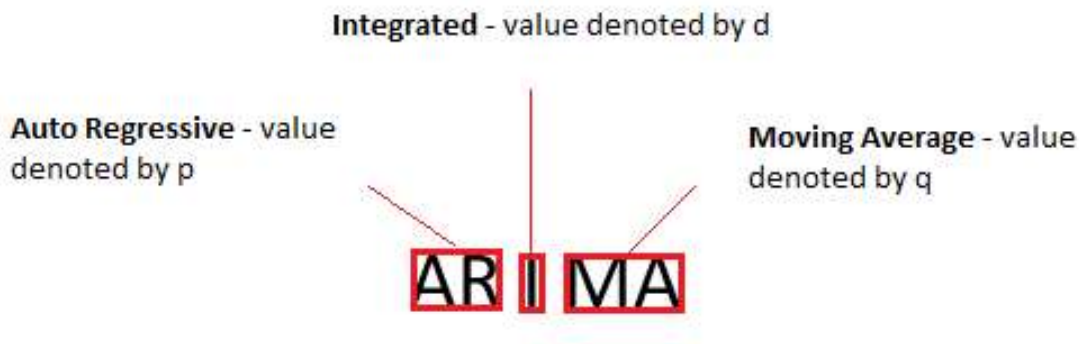


Based on our study and research, we realized that the log-transformed model performs slightly better than the linear model in most selected countries. Also we observe that 7-day models perform slightly better than 14-day models, i.e. when reducing the training set to only a few days prior to the testing region, results become better. This may come from exponential behavior,but is only true for the early stages of the spreading.

The result of linear regression model with log transformation has shown that there is still a lot of space for improvement in the model performance. Thus, further analysis certainly is needed. Next, we would like to study how time series models perform with our research dataset.

6. Predict trend using time series

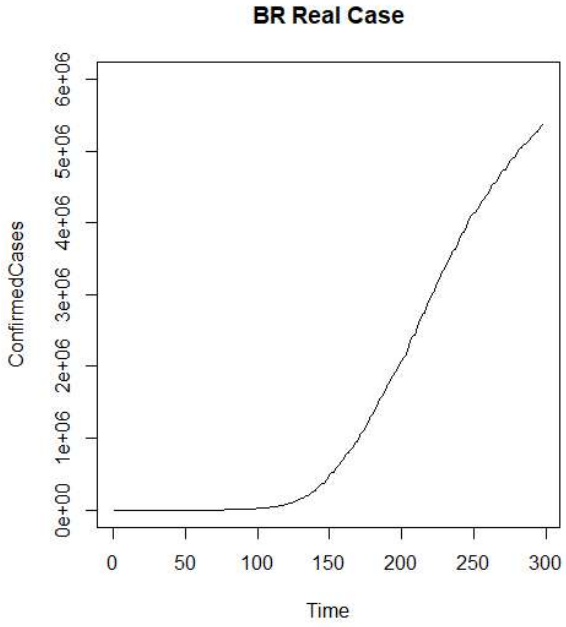
In this section, we used time series models to predict the trend of confirmed cases in several countries. Autoregressive integrated moving average (ARIMA) models are used, since they are capable of predicting the future performance of a variable only based on its historical evolution. The regression of the predicted variable is based on its previous values. The p parameter defines the number of these terms. Parameter d is related to the number of differences needed for stationarity. The error of the regression is based on a linear combination of error terms from the past. Parameter q is the number of lagged forecast errors in the prediction equation.

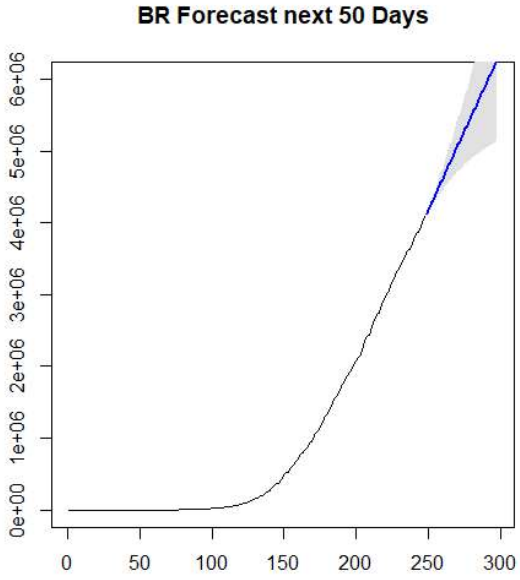


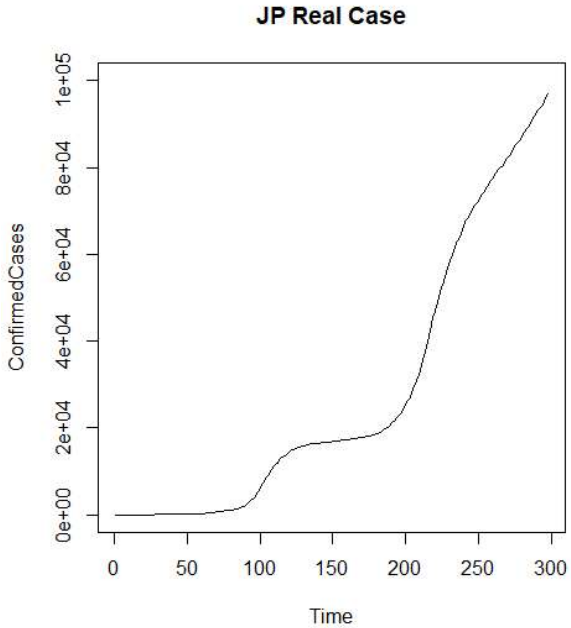
We use the auto.arima function to choose the optimal parameters based on the confirmed cases in the first 248 days. It calculates different combinations of parameters and chooses optimal ones with lower AIC, BIC. Then we use the arima function with optimal parameters to forecast the confirmed cases in the future 50 days. The forecasts are shown as blue lines, with the 80% prediction intervals as a dark shaded area. The forecasts of the confirmed cases in Brazil, Japan, India, Israel, Italy, Germany, France, South Africa, and China are shown below:

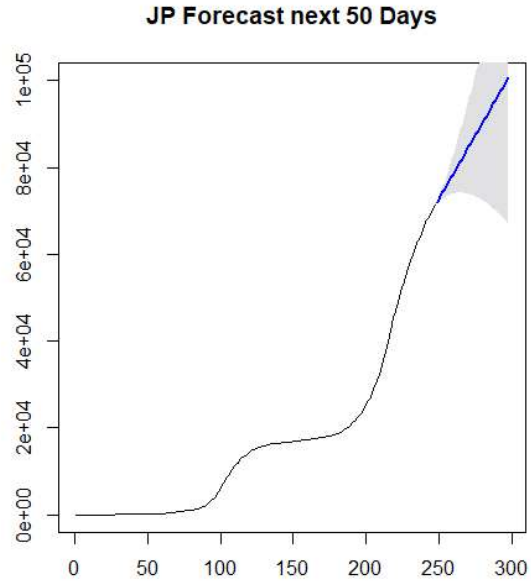
If the trend is still steeply going upwards in the forecasting plot compared to the real case, then we may conclude that the government did not effectively control the pandemic. If the upward trend slows down in the forecasting plot compared to the real case, then we conclude the government did a good job.

23

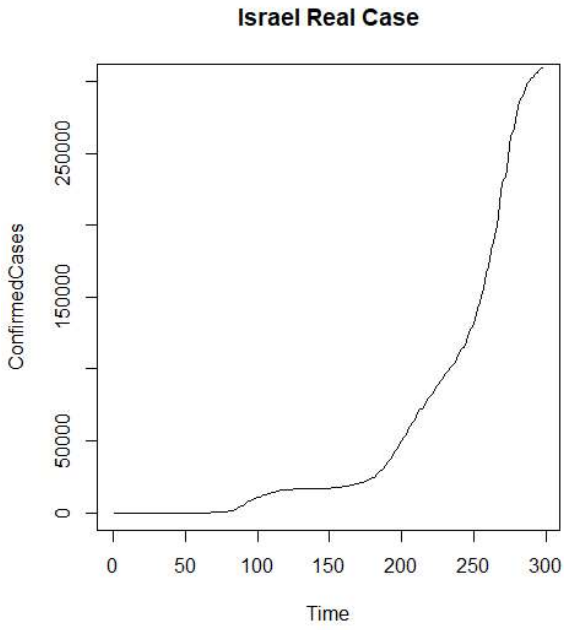

BR Real Case
ConfirmedCases
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
0
50
100
150
200
250
300
Time

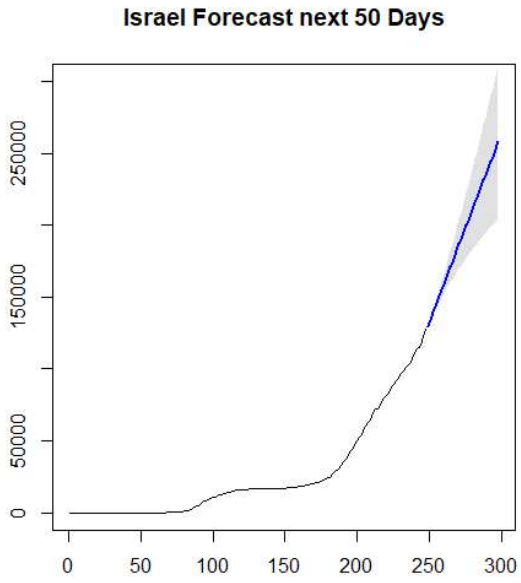

BR Forecast next 50 Days
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
0
50
100
150
200
250
300

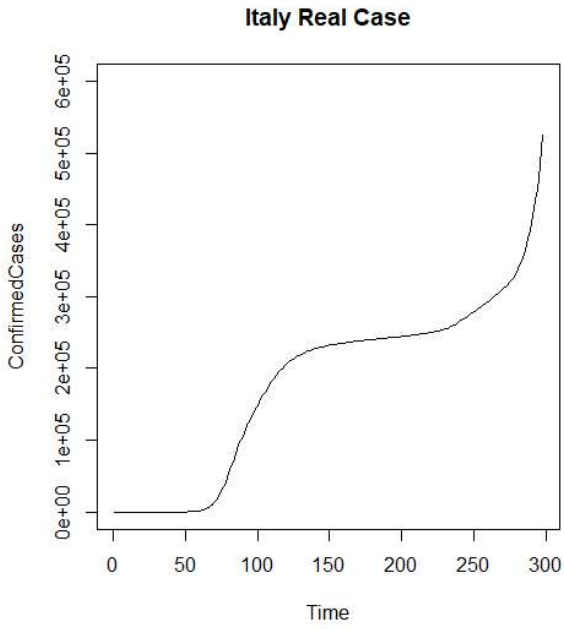

JP Real Case
ConfirmedCases
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
0
50
100
150
200
250
300
Time

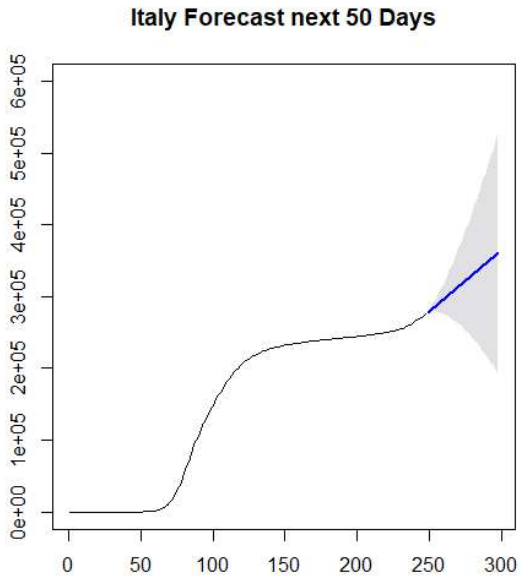

JP Forecast next 50 Days
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
0
50
100
150
200
250
300

24


Israel Real Case
ConfirmedCases
0
50000
150000
250000
0
50
100
150
200
250
300
Time

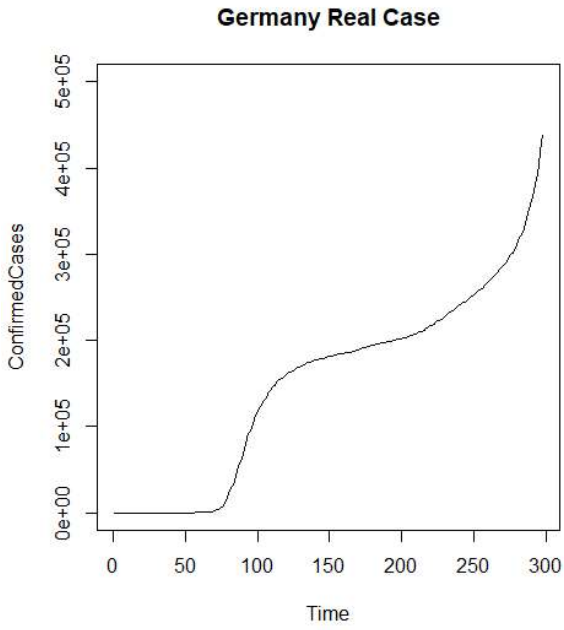

Israel Forecast next 50 Days
0
50000
150000
250000
0
50
100
150
200
250
300

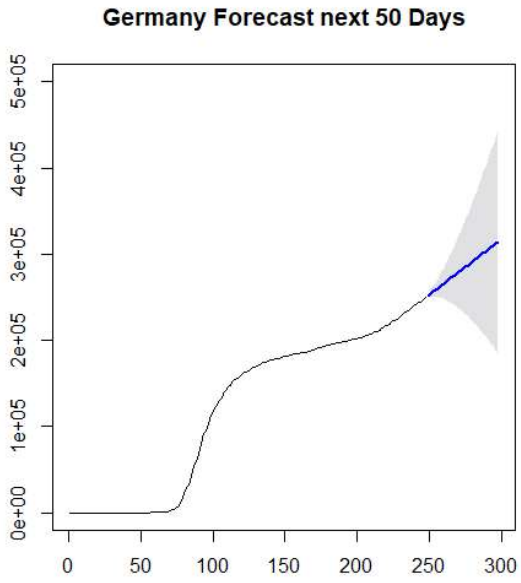

Italy Real Case
ConfirmedCases
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
0
50
100
150
200
250
300
Time

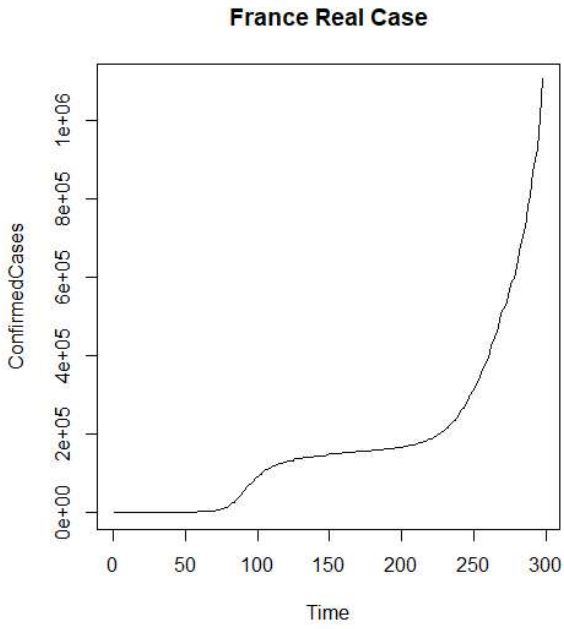

Italy Forecast next 50 Days
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
0
50
100
150
200
250
300

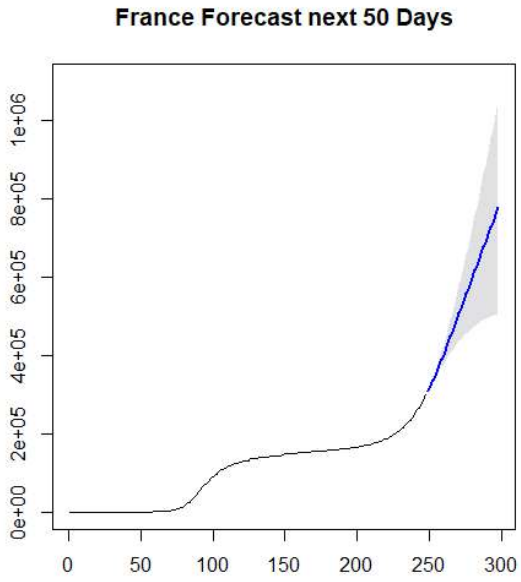
:

25

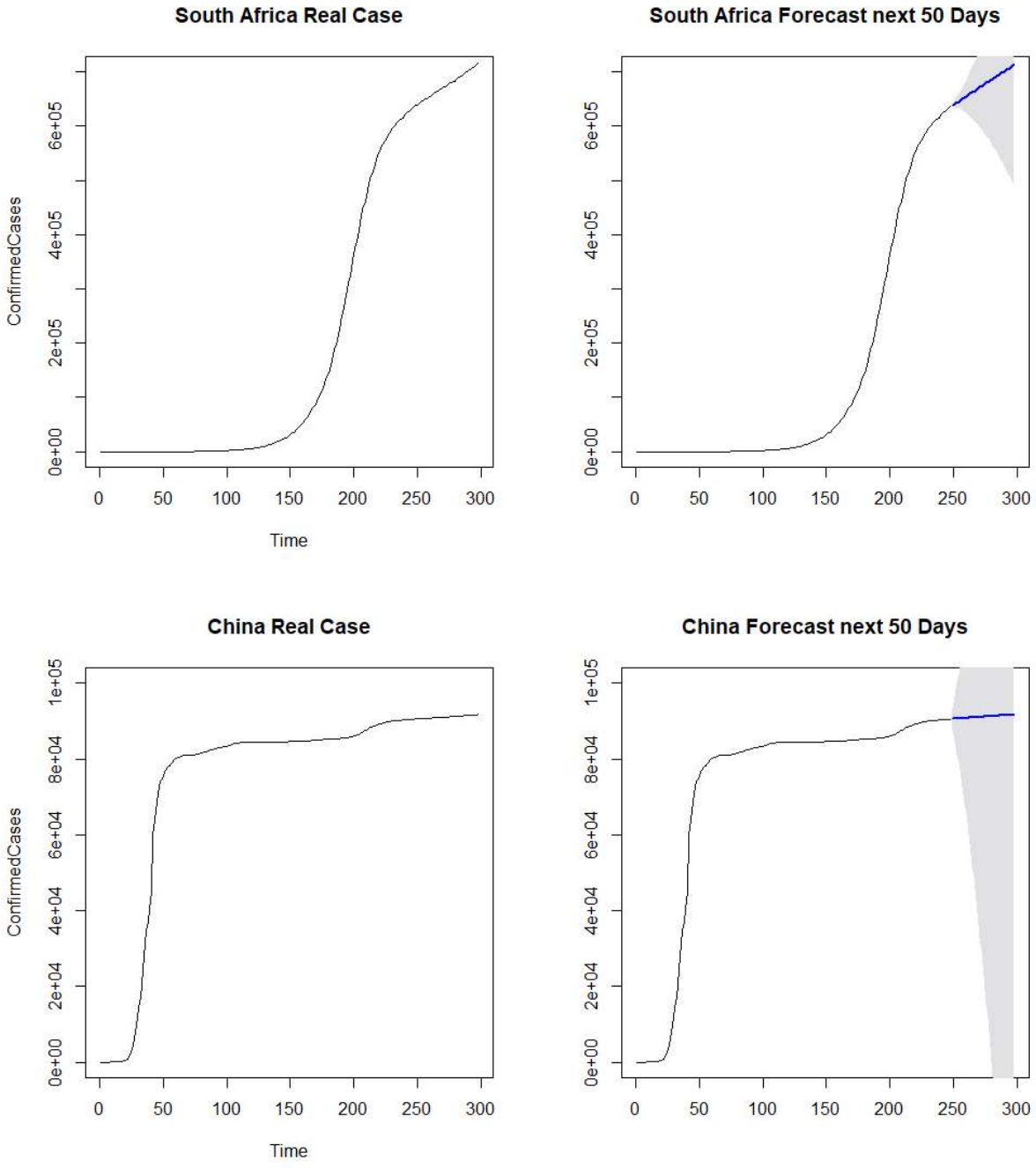

Germany Real Case
ConfirmedCases
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
0
50
100
150
200
250
300
Time


Germany Forecast next 50 Days
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
0
50
100
150
200
250
300


France Real Case
ConfirmedCases
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
0
50
100
150
200
250
300
Time


France Forecast next 50 Days
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
0
50
100
150
200
250
300

26



From the graphs, we can see that the governments of Italy, Germany, France did a bad job during the 50 days after the 248 days. The governments of Brazil, India, China did a better job during that time period, or we can say that the COVID-19 have not been spread in Brazil and India in that time period depending on the real situations.

27

7. SIR models

7.1. A brief summary of the SIR model

The SIR model is the most famous epidemiologic compartmental model. In this model, the population is assigned to three states:

1. (S) Susceptible: the people can be infected due to the transmission from the infected people, but they haven't contracted the disease.
2. (I) Infected: the people have contracted the disease.
3. (R) Recovered/Deceased: The disease may lead to one of the two outcomes: either the person recovered from the disease and developing immunity to the disease, or the person is deceased.

Image by Kai Sasaki from lewuathe.com

The person in state Susceptible has a rate βIS goes to state Infectious. Also, the person in state Infectious has a rate γI goes to state Recovered. The model can be described as the following system of differential equations with some initial conditions. We have beta is the contagion rate of the pathogen and gamma is the recovery rate.

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

28

7.2. Numerical results

We can solve this system of differential equations via Runge-Kutta method of 4th order for 3 dimensions. The basic idea of Runge-Kutta is that we can approximate the numerical result through the following equation.

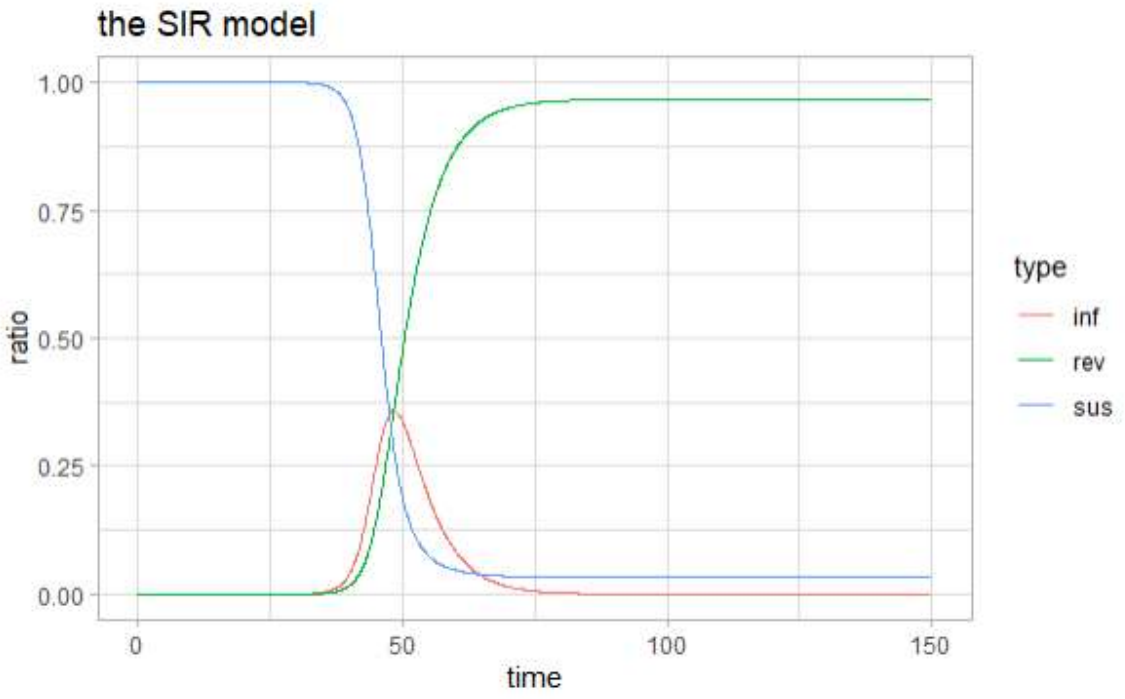
- Consider the differential Equation
  $\frac{dy}{dx} = f(x, y),\ y(x_0) = y_0$
  Calculate successively
- $k_1 = hf(x_0, y_0)$
- $k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$
- $k_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right)$
- $k_4 = hf(x_0 + h, y_0 + k_3)$

  Find $k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

$\therefore\ y_1 = y_0 + k$ and $x_1 = x_0 + h$

Image from: https://www.mathworks.com/matlabcentral/fileexchange/55430-runge-kutta-4th-order

If we set the parameter N = world population, beta = 0.7, gamma = 0.2, h = 0.1 and observate the trend from time = 0 to time = 150.

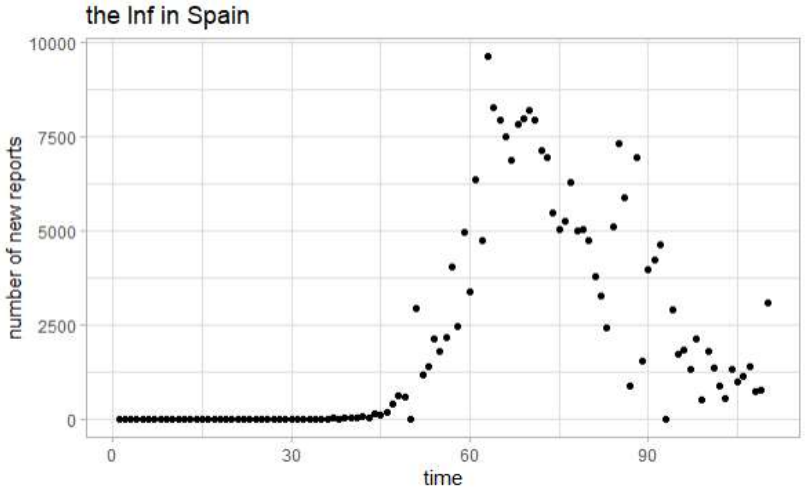


From the plot, we have the following observations:
1. the number of infected and suspected people will go to zero eventually.
2. the ratio of recovered state people will converge very close to 1.

29

7.3. Fit SIR model to real data (Spain) and predict

We want to find the best curve that fits the infected population in Spain. We will fit a SIR model based on the first 100 days data and see how it performs on the following 50 days. The following plot is the infected population in Spain scatter plot in 100 days.

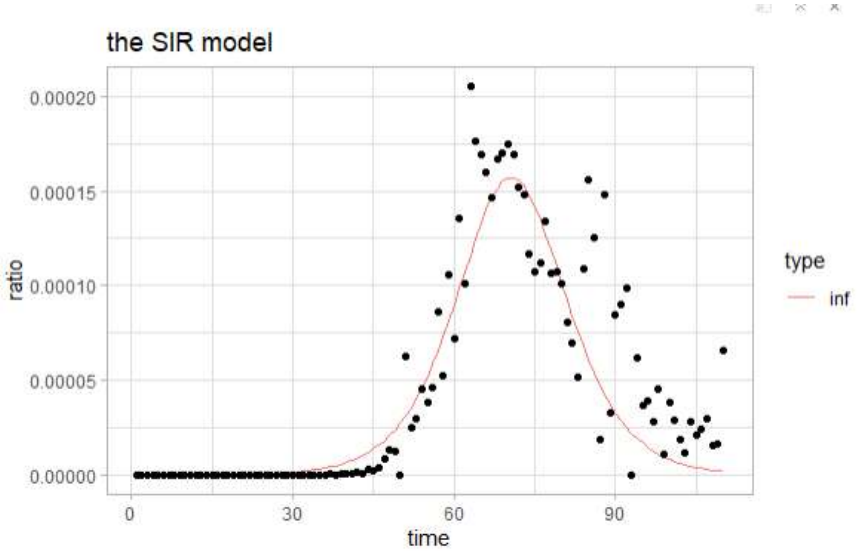


Then, we want to fit the SIR infected population curve to the first 100 days real data. I optimized the sum absolute difference between the data points and the fitted SIR curve to find the optimal beta and gamma.

```r
optim(c(0.1, 0.1), loss_function)$par
```

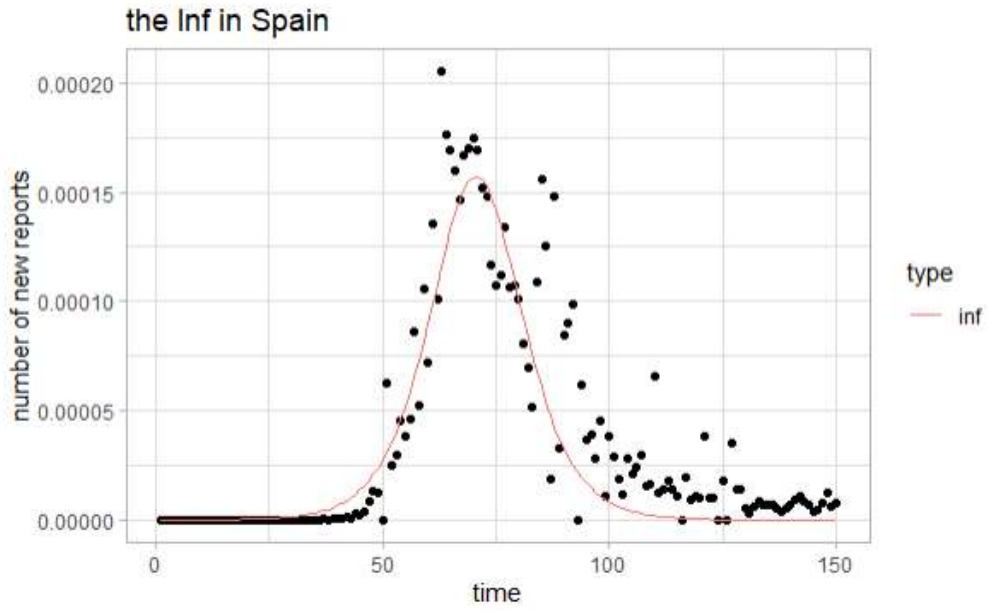
```
[1] 8.3683 8.2204
```

The optimal beta is 8.3683 and the optimal gamma is 8.2204. From the following plot, we can see that the SIR infected curve fits very well on the data points.



30

After we obtain the beta and gamma, we can do the prediction in the following 50 days. From the plot, we can see that the SIR model has a quite accurate prediction.



In summary, the SIR model depends on the numerical solution to the system of differential equations via Runge-Kutta method of 4th order for 3 dimensions. To fit the model to the real data, we optimize the sum absolute difference between the SIR model value with the data value to obtain the best beta and gamma for the model. To do the prediction, we plug in the value of beta and gamma into the SIR model to see the trend. The SIR model is strongly dependent on the data. If the infected data included the decreasing trend after the peak, the SIR model could produce accurate predictions.

8. Other analysis

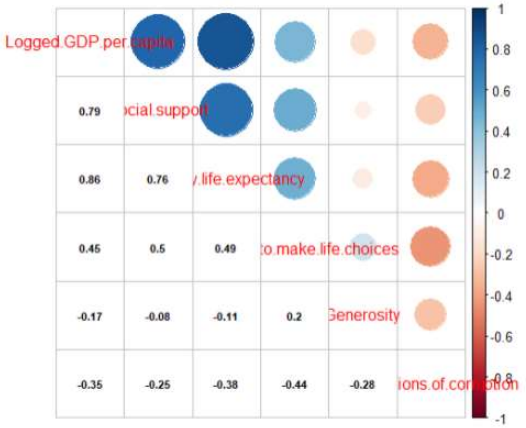
8.1. Features

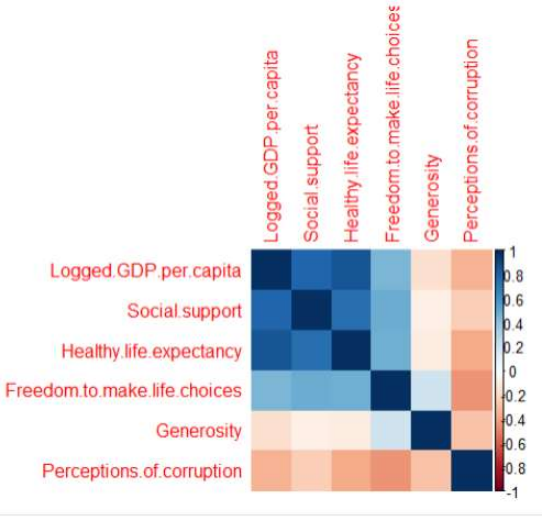
We want to find features in the global happiness measurement as the independent variable X that would relate to the dependent variable Y: the report cases in millions for a country or other measurements of how the country performs during the COVID pandemic. Since different countries have different populations, It is more appropriate to use this measure rather than using the number of report cases directly.

The features in the global happiness measurement that may be used are logged GDP per capita, social support index, healthy life expectancy, freedom to make life choices, generosity

and perception of corruption. And later, we will add some other features to see what might be significantly related to the report cases in millions in a country.

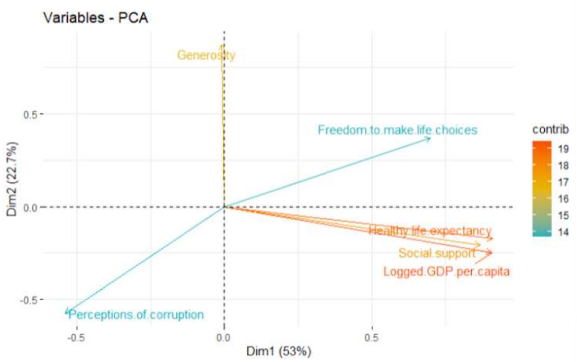
8.2. Correlation table of the selected features



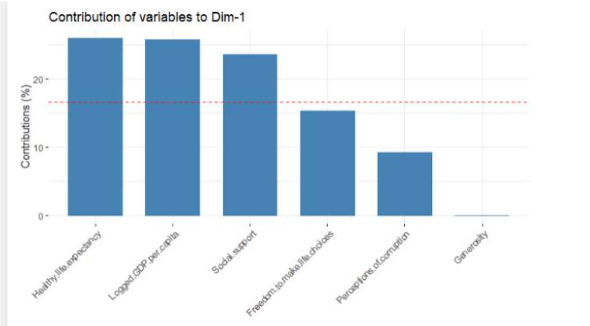


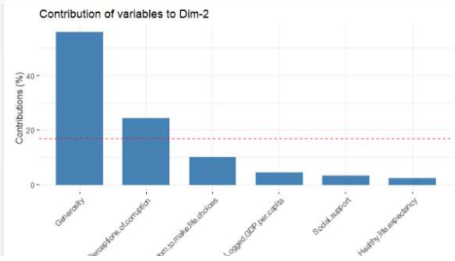
The correlation table of the features. The logged GDP per capita, social support, healthy life expectancy and freedom to make life choices positively relate to each other and negatively relate to generosity and perceptions of corruption.

8.3. Principal Component Analysis




```
Importance of components:
                          PC1   PC2    PC3    PC4    PC5    PC6
Standard deviation     1.78 1.166 0.7732 0.7080 0.4748 0.3656
Proportion of Variance 0.53 0.227 0.0996 0.0835 0.0376 0.0223
Cumulative Proportion  0.53 0.757 0.8566 0.9402 0.9777 1.0000
```





32

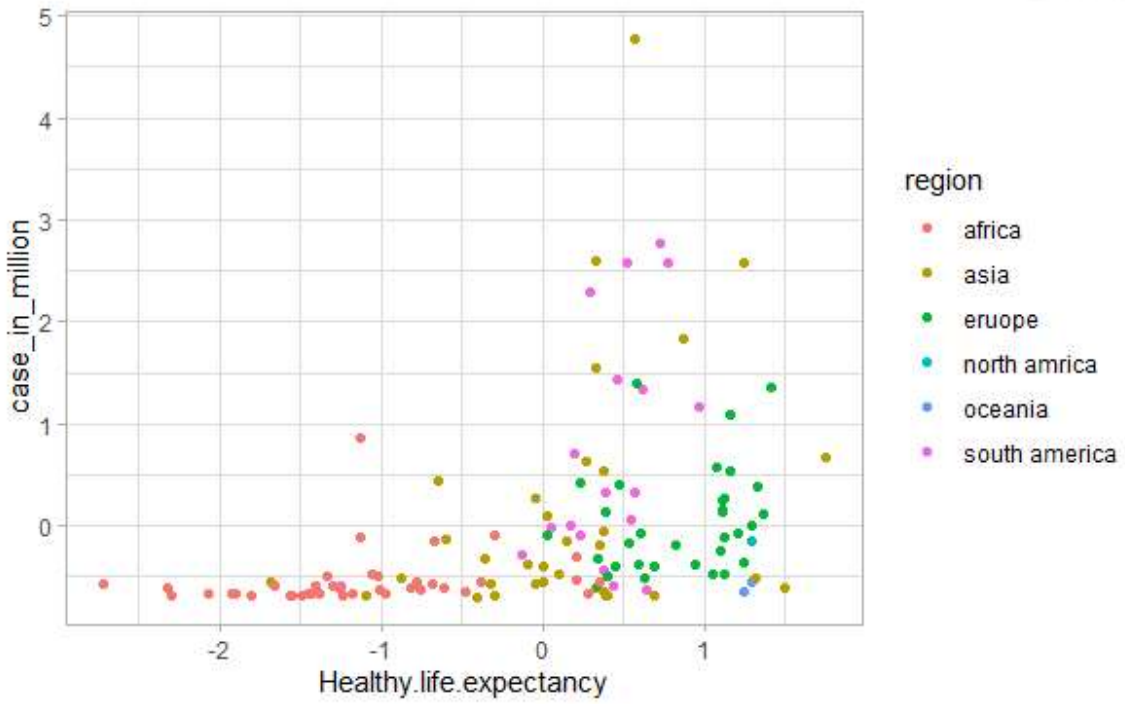
We first standardize data and then apply PCA. By applying the PCA, we could see that the two dominant PCs are the first two PCs. The cumulative proportion is 75.7%. We could see that the logged GDP per capita, Social support and Healthy life expectancy are pointing toward the same direction. Generosity is another direction. Freedom to make life choices and perceptions of corruption are pointing in the opposite direction.

8.4. Scatter plots

The variable reported cases in million data was updated on Nov 1, 2020. The plot of the healthy life expectancy vs deaths in 100 cases provides an example of how the relation may look like. We could see some trends here, but it is not obvious. So, we need to try other features.

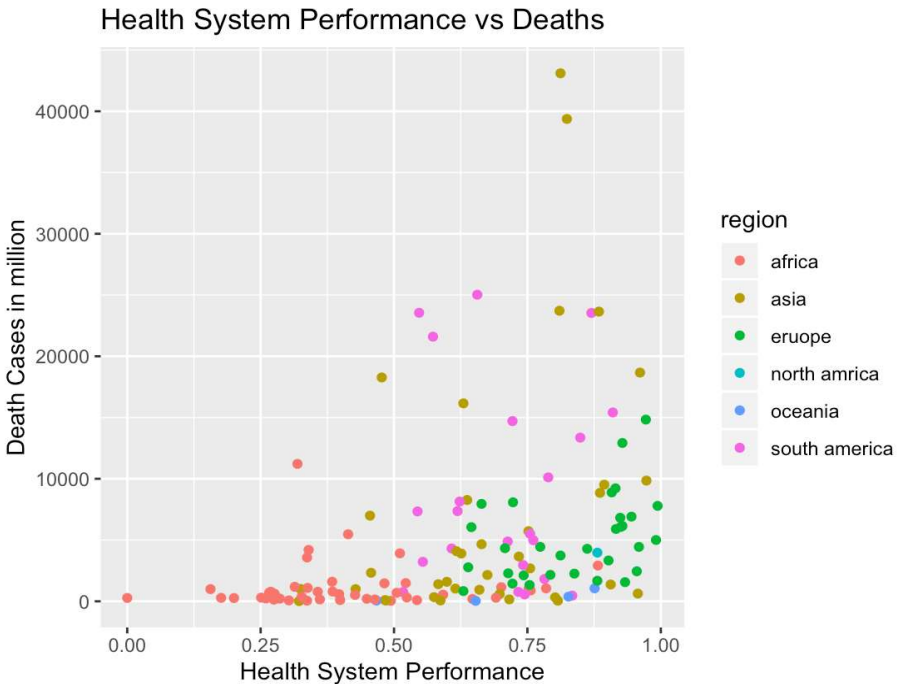
The variable death cases in million data was updated on Nov 1, 2020.Also, we used health system performance to analyze its relation to the COVID death cases. The health system performance index comes from a WHO report, which describes countries' health system conduct ranking from 0 to 1. Here is a plot of its relationship. However, the trend is not obvious and we may not get any solid conclusion based on this plot. Thus more analysis required.

Healthy life expectancy vs Deaths



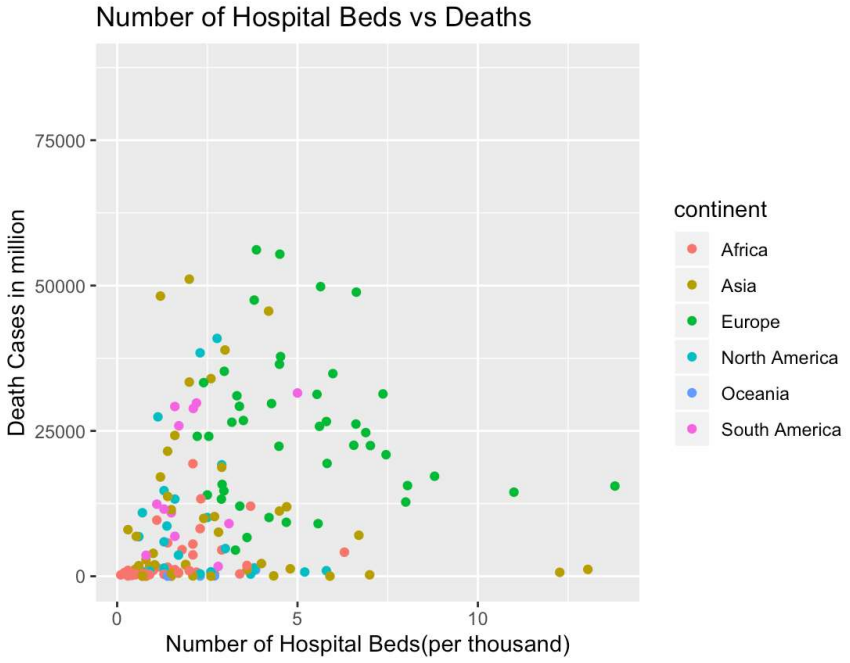
33

Healthy system performance vs Deaths



8.5. Try other features

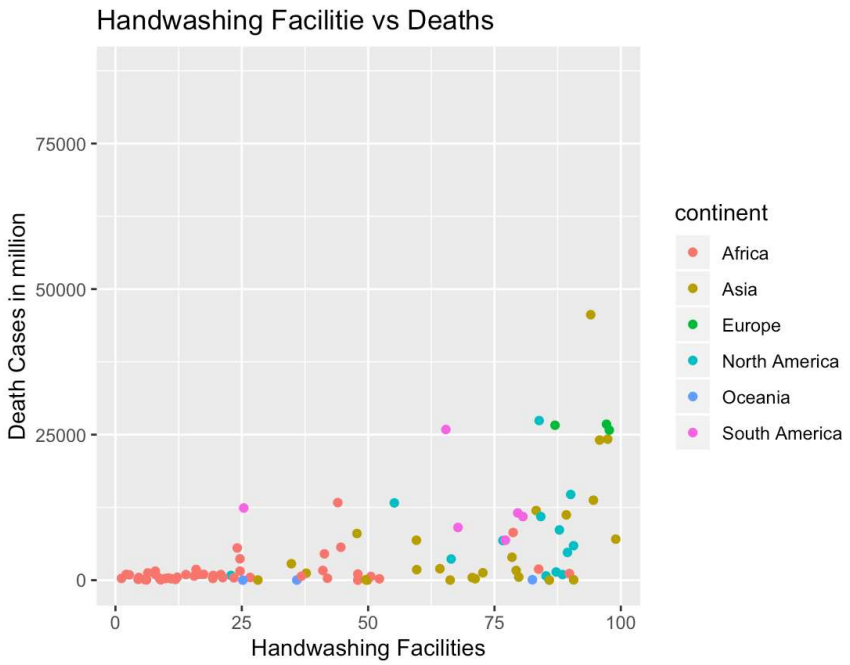
Number of Hospital Beds vs Deaths



34

Then we aimed to check if medical resources have impacts on COVID death cases. First, we chose the number of hospital beds as a variable. The plot shows that on Nov 30, 2020, there is no sign of strong relationship between two variables. We cannot conclude that the more the hospital beds are, the less death cases are.

Handwashing Facility vs Deaths



Second, we chose handwashing facilities as a variable. The plot is quite surprising since there might be a weak relationship between the handwashing facilities and death cases in Asia. The more handwashing facilities indicates more deaths. However, more analysis is required to validate our thoughts.

```
Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                -1911.05    1767.26  -1.081   0.2830
hospital_beds_per_thousand   553.55     584.40   0.947   0.3465
handwashing_facilities       120.60      30.85   3.909   0.0002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7594 on 76 degrees of freedom
  (110 observations deleted due to missingness)
Multiple R-squared:  0.2439,    Adjusted R-squared:  0.224
F-statistic: 12.26 on 2 and 76 DF,  p-value: 2.428e-05
```

35

Considering the impact of medical services on the COVID deaths cases, we performed the regression model. Since the p-value for handwashing facilities is extremely small, we might conclude that this feature is significant. However, the R-square is small, then the independent variable is not explaining much in the variation of our dependent variables, despite the significance of the identified independent variable.

9. Results

Firstly, by generally looking into the pandemic of 2020 using data visualization, we found that the U.S, India and Brazil suffered most from COVID-19. Besides, the confirmed cases of the U.S and Brazil distributed more equally among states while in China, most cases occurred in HuBei Province since that province was the place where the very first case was found. Also, we did three simple predictions for the worldwide data and got higher predicted confirmed cases from SVM than Polynomial Regression and Bayesian Ridge Regression. Then we explore the trend of mortality and recovery rate. The trend of mortality rate reflects the more and more mature treatment methods and sufficient medical resources worldwide. The trend of recovery rate reflects the shortage of medical resources due to the shock increase of confirmed cases. The relation between deaths and recoveries shows the worldwide medical situation is gradually keeping pace with the spread of the epidemic.

Then we tried to go deeper into some representative countries and do predictions and see if the models can work well. We have conducted research for Germany, China, France, Japan, Italy, Israel, and Brazil. Based on our study and research, we realized that the log-transformed model performs slightly better than the linear model in most selected countries. Meanwhile, we observed that 7-day models performed slightly better than 14-day models, i.e. when reducing the training set to only a few days prior to the testing region, results became better. This may come from exponential behavior, but is only true for the early stages of the spreading.

For later stages, we chose time series models and compared the prediction result and the real case trend, which reflects the efficiency of government behavior. From the forecasts, we can see that the governments of Italy, Germany, France did a bad job during the 50 days after the 248 days while the governments of Brazil, India, China did a better job during that time period. Combining the real situations, we can conclude that the COVID-19 have not been spread in Brazil and India in that time period.

Furthermore, knowing that the SIR model is usually working well in predicting epidemics, we turned to the SIR model. It is based on the numerical solution to the system of differential equations via Runge-Kutta method of 4th order for 3 dimensions. To fit the model to the real data, we optimize the sum absolute difference between the SIR model value with the

data value to obtain the beta and gamma. To do the prediction, we plug in the value of beta and gamma into the SIR model to see the trend. From the SIR model fitting in the real data in Spain, we can conclude that the model is strongly dependent on the data. And if the infected data included the decreasing trend after the peak, the SIR model could produce accurate predictions. It requires more historical data than the linear regression model did and its prediction is more accurate and reliable.

After all these predictions, we are curious about if mood status will influence health, so we tried to see if there is a relationship between citizen happiness and the case trend. We set the features in the global happiness measurements as the independent variables X that would relate to the dependent variable Y which is the report cases in millions for a country or other measurements of how the country performs during the COVID pandemic. Our first step is variable selection through PCA and test the relationship between the report cases in million and healthy life expectancy. We did not get any solid conclusions from the analysis. In addition, we analyzed more on how medical services (X variables) in each continent affect death cases (Y variable). The conclusion is that each variable related to medical services is not significant enough to have impacts on deaths.

37

Reference

Paper reference

[1] Bian, Xingyu, Coronavirus(COVID-19) Visualization & Prediction, https://github.com/therealcyberlord/coronavirus_visualization_and_prediction

[2] Sanchez, Patrick, COVID Global Forecast: SIR model + ML regressions, Kaggle, https://www.kaggle.com/saga21/COVID-global-forecast-sir-model-ml-regressions

[3] Kp, Devakumar, COVID-19 - Analysis, Visualization & Comparisons, Kaggle, https://www.kaggle.com/imdevskp/COVID-19-analysis-visualization-comparisons

Data Source:
[1] COVID19 Global Forecasting(Week 4) Dataset
https://www.kaggle.com/c/COVID19-global-forecasting-week-4

[2] Population by Country 2020
https://www.kaggle.com/tanuprabhu/population-by-country-2020

[3] WHO health information
https://www.who.int/healthinfo/paper30.pdf

[4] WHO COVID19 dataset
https://COVID19.who.int/

[5] World happiness report
https://www.kaggle.com/mathurinache/world-happiness-report

[6] data source COVID-19 Data Visualization and Prediction
https://github.com/CSSEGISandData/COVID-19

[7] Daily updated – Our World in Data COVID-19 database
https://github.com/owid/COVID-19-data/tree/master/public/data

38

Appendix

Visualization and Prediction

```
#Worldwide Overview
adjusted_dates = adjusted_dates.reshape(1, -1)[0]
plt.figure(figsize=(16, 10))
plt.plot(adjusted_dates, world_cases)
plt.plot(adjusted_dates, world_confirmed_avg, linestyle='dashed', color='orange')
plt.title('# of Coronavirus Cases Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('# of Cases', size=30)
plt.legend(['Worldwide Coronavirus Cases', 'Moving Average {} Days'.format(window)], prop={'size': 20})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()


#World Daily Increased in Confirmed Cases
plt.figure(figsize=(16, 10))
plt.bar(adjusted_dates, world_daily_increase)
plt.plot(adjusted_dates, world_daily_increase_avg, color='orange', linestyle='dashed')
plt.title('World Daily Increases in Confirmed Cases', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('# of Cases', size=30)
plt.legend(['Moving Average {} Days'.format(window), 'World Daily Increase in COVID-19 Cases'], prop={'size': 20})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

```
#Countries
def country_plot(x, y1, y2, y3, y4, country):
    confirmed_avg = moving_average(y1, window)
    confirmed_increase_avg = moving_average(y2, window)
    death_increase_avg = moving_average(y3, window)
    recovery_increase_avg = moving_average(y4, window)

    plt.figure(figsize=(16, 10))
    plt.plot(x, y1)
    plt.plot(x, confirmed_avg, color='red', linestyle='dashed')
    plt.legend(['{} Confirmed Cases'.format(country), 'Moving Average {} Days'.format(window)], prop={'size': 20})
    plt.title('{} Confirmed Cases'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('# of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

    plt.figure(figsize=(16, 10))
    plt.bar(x, y2)
    plt.plot(x, confirmed_increase_avg, color='red', linestyle='dashed')
    plt.legend(['Moving Average {} Days'.format(window), '{} Daily Increase in Confirmed Cases'.format(country)], pr
    plt.title('{} Daily Increases in Confirmed Cases'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('# of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

    plt.figure(figsize=(16, 10))
    plt.bar(x, y3)
    plt.plot(x, death_increase_avg, color='red', linestyle='dashed')
    plt.legend(['Moving Average {} Days'.format(window), '{} Daily Increase in Confirmed Deaths'.format(country)], p
    plt.title('{} Daily Increases in Deaths'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('# of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

    plt.figure(figsize=(16, 10))
    plt.bar(x, y4)
    plt.plot(x, recovery_increase_avg, color='red', linestyle='dashed')
    plt.legend(['Moving Average {} Days'.format(window), '{} Daily Increase in Confirmed Recoveries'.format(country)
    plt.title('{} Daily Increases in Recoveries'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('# of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()
```

39

```
def get_country_info(country_name):
    country_cases = []
    country_deaths = []
    country_recoveries = []

    for i in dates:
        country_cases.append(confirmed_df[confirmed_df['Country/Region']==country_name][i].sum())
        country_deaths.append(deaths_df[deaths_df['Country/Region']==country_name][i].sum())
        country_recoveries.append(recoveries_df[recoveries_df['Country/Region']==country_name][i].sum())
    return (country_cases, country_deaths, country_recoveries)


def country_visualizations(country_name):
    country_info = get_country_info(country_name)
    country_cases = country_info[0]
    country_deaths = country_info[1]
    country_recoveries = country_info[2]

    country_daily_increase = daily_increase(country_cases)
    country_daily_death = daily_increase(country_deaths)
    country_daily_recovery = daily_increase(country_recoveries)

    country_plot(adjusted_dates, country_cases, country_daily_increase, country_daily_death, country_daily_recovery,

countries = ['US',  'China',  'Spain',  'Mexico']

for country in countries:
    country_visualizations(country)
```

Bar chart visualization

```
world_cases = np.sum(country_confirmed_cases)
us = latest_data[latest_data['Country_Region']=='US']['Confirmed'].sum()
outside_us = world_cases - us
plt.figure(figsize=(16, 9))
plt.barh('United States', us)
plt.barh('Outside United States', outside_us)
plt.title('# of Total Coronavirus Confirmed Cases', size=20)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
plt.figure(figsize=(16, 9))
plt.barh('United States', us /world_cases)
plt.barh('Outside United States', outside_us /world_cases)
plt.title('# of Coronavirus Confirmed Cases Expressed in Percentage', size=20)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

Pie chart visualization

```
    c = ['lightcoral', 'rosybrown', 'sandybrown', 'navajowhite', 'gold',
        'khaki', 'lightskyblue', 'turquoise', 'lightslategrey', 'thistle', 'pink']
    plt.figure(figsize=(20,15))
    plt.title('Covid-19 Confirmed Cases per Country', size=20)
    plt.pie(visual_confirmed_cases, colors=c,shadow=True, labels=visual_confirmed_cases)
    plt.legend(visual_unique_countries, loc='best', fontsize=12)
    plt.show()
```

40

```
pie_chart_countries = ['US', 'Brazil', 'Russia', 'India', 'Peru', 'Mexico', 'Canada',
                       'Australia', 'China', 'Italy', 'Germany', 'France', 'United Kingdom',
for i in pie_chart_countries:
    regions = list(latest_data[latest_data['Country_Region']==i]['Province_State'].unique())
    confirmed_cases = []
    no_cases = []
    for i in regions:
        cases = latest_data[latest_data['Province_State']==i]['Confirmed'].sum()
        if cases > 0:
            confirmed_cases.append(cases)
        else:
            no_cases.append(i)

    for i in no_cases:
        regions.remove(i)

    regions = [k for k, v in sorted(zip(regions, confirmed_cases), key=operator.itemgetter(1

    for i in range(len(regions)):
        confirmed_cases[i] = latest_data[latest_data['Province_State']==regions[i]]['Confirm

    if(len(regions)>5):
        regions_5 = regions[:5]
        regions_5.append('Others')
        confirmed_cases_5 = confirmed_cases[:5]
        confirmed_cases_5.append(np.sum(confirmed_cases[5:]))
        plot_pie_charts(regions_5,confirmed_cases_5, 'Covid-19 Confirmed Cases in {}'.format
    else:
        plot_pie_charts(regions,confirmed_cases, 'Covid-19 Confirmed Cases in {}'.format(i))
```

```
# use this to find the optimal parameters for SVR
c = [0.01, 0.1, 1]
gamma = [0.01, 0.1, 1]
epsilon = [0.01, 0.1, 1]
shrinking = [True, False]
svm_grid = {'C': c, 'gamma' : gamma, 'epsilon': epsilon, 'shrinking' :
shrinking}

svm = SVR(kernel='poly', degree=3)
svm_search = RandomizedSearchCV(svm, svm_grid,
scoring='neg_mean_squared_error', cv=3, return_train_score=True,
n_jobs=-1, n_iter=30, verbose=1)
svm_search.fit(X_train_confirmed, y_train_confirmed)
svm_search.best_params_
svm_confirmed = svm_search.best_estimator_
svm_confirmed = SVR(shrinking=True, kernel='poly',gamma=0.01,
epsilon=1,degree=3, C=0.1)
svm_confirmed.fit(X_train_confirmed, y_train_confirmed)
svm_pred = svm_confirmed.predict(future_forcast)

# transform our data for polynomial regression
poly = PolynomialFeatures(degree=4)
poly_X_train_confirmed = poly.fit_transform(X_train_confirmed)
poly_X_test_confirmed = poly.fit_transform(X_test_confirmed)
```

41

```
poly_future_forcast = poly.fit_transform(future_forcast)

bayesian_poly = PolynomialFeatures(degree=5)
bayesian_poly_X_train_confirmed =
bayesian_poly.fit_transform(X_train_confirmed)
bayesian_poly_X_test_confirmed =
bayesian_poly.fit_transform(X_test_confirmed)
bayesian_poly_future_forcast = bayesian_poly.fit_transform(future_forcast)

# polynomial regression
linear_model = LinearRegression(normalize=True, fit_intercept=False)
linear_model.fit(poly_X_train_confirmed, y_train_confirmed)
test_linear_pred = linear_model.predict(poly_X_test_confirmed)
linear_pred = linear_model.predict(poly_future_forcast)
print('MAE:', mean_absolute_error(test_linear_pred, y_test_confirmed))
print('MSE:',mean_squared_error(test_linear_pred, y_test_confirmed))


# bayesian ridge polynomial regression
tol = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]
alpha_1 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]
alpha_2 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]
lambda_1 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]
lambda_2 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]
normalize = [True, False]

bayesian_grid = {'tol': tol, 'alpha_1': alpha_1, 'alpha_2' : alpha_2,
'lambda_1': lambda_1, 'lambda_2' : lambda_2,
                 'normalize' : normalize}

bayesian = BayesianRidge(fit_intercept=False)
bayesian_search = RandomizedSearchCV(bayesian, bayesian_grid,
scoring='neg_mean_squared_error', cv=3, return_train_score=True,
n_jobs=-1, n_iter=40, verbose=1)
bayesian_search.fit(bayesian_poly_X_train_confirmed, y_train_confirmed)
bayesian_search.best_params_
bayesian_confirmed = bayesian_search.best_estimator_
test_bayesian_pred =
bayesian_confirmed.predict(bayesian_poly_X_test_confirmed)
bayesian_pred = bayesian_confirmed.predict(bayesian_poly_future_forcast)
print('MAE:', mean_absolute_error(test_bayesian_pred, y_test_confirmed))
print('MSE:',mean_squared_error(test_bayesian_pred, y_test_confirmed))
```

42

Linear Model

```
def data_split(data,train_start, train_end, test_start, test_end):
    #obtain the subset of the time
    train_period = pd.date_range(start=train_start, end=train_end)
    train_data = data.loc[data['Date'].isin(train_period)]
    test_period = pd.date_range(start=test_start, end=test_end)
    test_data = data.loc[data['Date'].isin(test_period)]
    total_period = pd.date_range(start=train_start, end=test_end)
    whole_data = data.loc[data['Date'].isin(total_period)]
    return train_data, test_data, whole_data
```

```
def data_trend(data, country, train_start, train_end, test_start, test_end):
  #split data
  data_list = data_split(data,train_start, train_end, test_start, test_end)
  train_data = data_list[0]
  test_data = data_list[1]
  whole_data = data_list[2]
  #add day counter
  le = preprocessing.LabelEncoder()
  train_data["Day_counter"] = le.fit_transform(train_data.Date)
  test_data["Day_counter"] = le.fit_transform(test_data.Date)
  whole_data["Day_counter"] = le.fit_transform(whole_data.Date)
  #train data & test data
  y_train = train_data[train_data['CountryRegion']==country]['ConfirmedCases']
  x_train = np.expand_dims(np.arange(1,15), axis=1)
  y_test = test_data[test_data['CountryRegion']==country]['ConfirmedCases']
  x_test = np.expand_dims(np.arange(15,29), axis=1)
  #fit linear model
  regr = linear_model.LinearRegression()
  regr.fit(x_train, y_train)
  y_pred = regr.predict(x_test)
  y_actual = whole_data[whole_data['CountryRegion']==country]['ConfirmedCases']
  y_predict = np.concatenate((y_train, y_pred))
  #plot1
  fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,6))
  y1 = y_actual
  x1 = np.arange(1,29)
  ax1.plot(x1, y1, 'bo--')
  ax1.set_title(country + " ConfirmedCases between " + train_start + " and " + test_end)
  ax1.set_xlabel("Days")
  ax1.set_ylabel("ConfirmedCases")

  y2 = np.log(y_actual)
  x2 = np.arange(1,29)
  ax2.plot(x2, y2, 'bo--')
  ax2.set_title(country + " Log ConfirmedCases between " + train_start + " and " + test_end)
  ax2.set_xlabel("Days")
  ax2.set_ylabel("Log ConfirmedCases")
```

43

Linear Model Prediction

```
def plot_linreg_14days(data, country, train_start, train_end, test_start, test_end):
  #split data
  data_list = data_split(data,train_start, train_end, test_start, test_end)
  train_data = data_list[0]
  test_data = data_list[1]
  whole_data = data_list[2]
  #add day counter
  le = preprocessing.LabelEncoder()
  train_data["Day_counter"] = le.fit_transform(train_data.Date)
  test_data["Day_counter"] = le.fit_transform(test_data.Date)
  whole_data["Day_counter"] = le.fit_transform(whole_data.Date)
  #train data & test data
  y_train = train_data[train_data['CountryRegion']==country]['ConfirmedCases']
  x_train = np.expand_dims(np.arange(1,15), axis=1)
  y_test = test_data[test_data['CountryRegion']==country]['ConfirmedCases']
  x_test = np.expand_dims(np.arange(15,29), axis=1)
  #fit linear model
  regr = linear_model.LinearRegression()
  regr.fit(x_train, y_train)
  y_pred = regr.predict(x_test)
  y_actual = whole_data[whole_data['CountryRegion']==country]['ConfirmedCases']
  y_predict = np.concatenate((y_train, y_pred))
  #plot1
  fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,6))
  ax1.plot(np.arange(1,29), y_predict)
  ax1.plot(np.arange(1,29), y_actual)
  ax1.axvline(14, linewidth=2, ls = ':', color='grey', alpha=0.5)
  ax1.legend(['Predicted cases', 'Actual cases', 'Train-test split'], loc='upper left')
  ax1.set_xlabel("Day count from (" + train_start + " to " + test_end + ")")

  ax2.plot(np.arange(1,29), np.log(y_predict))
  ax2.plot(np.arange(1,29), np.log(y_actual))
  ax2.axvline(14, linewidth=2, ls = ':', color='grey', alpha=0.5)
  ax2.legend(['Predicted cases', 'Actual cases', 'Train-test split'], loc='upper left')
  ax2.set_xlabel("Day count from (" + train_start + " to " + test_end + ")")
  plt.suptitle(("ConfirmedCases predictions based on Log-Lineal Regression for "+country))
```

Time Series Model

```
data<-read.csv('Desktop/global data.csv',head=T)
library(dplyr)
Brazil<-data%>%filter(CountryRegion=='Brazil')%>%select(ConfirmedCases)
Brts<-as.ts(Brazil,start = as.Date('2020-01-03'),frequency = 1)
plot.ts(Brts)
```

```
Brtrain<-Brts[1:248]
par(mfrow=c(1,1))
plot.ts(Brtrain)
library(tseries)
library(forecast)
Brpdq<-auto.arima(Brtrain)
Brpdq
Brmodel<-arima(Brtrain,order=c(3,2,2),method='ML')
Brforecast<-forecast(Brmodel,h=50,level=c(99.5))
par(mfrow=c(1,2))
plot.ts(Brts,ylim=c(0,6000000),main='BR Real Case')
plot(Brforecast,ylim=c(0,6000000),main='BR Forecast next 50 Days')
```

44

SIR model

The main reference of the codes in the SIR model is [2].

functions

```{r}
fun_S <- function(s, i, beta){
  f_S <- -beta*s*i
}

fun_I <- function(s, i, beta, gamma){
  f_I <- beta*s*i - gamma*i
}

fun_R <- function(i, gamma){
  f_R <- gamma*i
}
```

solve the system of differential equations

```{r}
# Runge-Kutta method of 4th order for 3 dimensions

RK4 <- function(s, i, r, f_S, f_I, f_R, beta, gamma, h){
  s1 <- f_S(s,i,beta)*h
  i1 <- f_I(s,i,beta, gamma)*h
  r1 <- f_R(i, gamma)*h

  sk1 = s + s1*h/2
  ik1 = i + i1*h/2

  s2 <- f_S(sk1,ik1,beta)*h
  i2 <- f_I(sk1,ik1,beta, gamma)*h
  r2 <- f_R(ik1, gamma)*h

  sk2 = s + s2*h/2
  ik2 = i + i2*h/2

  s3 <- f_S(sk2,ik2,beta)*h
  i3 <- f_I(sk2,ik2,beta, gamma)*h
  r3 <- f_R(ik2, gamma)*h

  sk4 = s + s3
  ik4 = i + i3

  s4 <- f_S(sk4,ik4,beta)*h
  i4 <- f_I(sk4,ik4,beta, gamma)*h
  r4 <- f_R(ik4, gamma)*h

  s = s + (s1 + 2*(s2 + s3)+s4)/6
  i = i + (i1 + 2*(i2 + i3)+i4)/6
  r = r + (r1 + 2*(r2 + r3)+r4)/6
  return(c(s,i,r))
}
```

45

define the SIR function
N is the total population
beta is the rate S to I
gamma is the rate I to R
h is the step size of numerical integration

inital condition is I0: the ratio of inital inflected people in the total population

```{r}
sir <- function(N, B, I0, beta, gamma, h){
  # initial condition
  s = (N-1)/N - b0
  i = 1/N + b0
  r = 0

  s_ls <- rep(0, B)
  i_ls <- rep(0, B)
  r_ls <- rep(0, B)

  for(j in 1:B){
    s_ls[j] <- s
    i_ls[j] <- i
    r_ls[j] <- r
    res <- RK4(s,i,r,fun_S, fun_I, fun_R, beta, gamma, h)
    s <- res[1]
    i <- res[2]
    r <- res[3]
  }
  return(data.frame(time = 1:B*h, sus = s_ls,inf = i_ls, rev =  r_ls))
}
```

Fit the SIR model with real data

```{r}
Spain_pop <- 46.94e6

inf0 = 0
sus0 = Spain_pop - inf0
rec0 = 0

sir_model <- function(par, x, beta, gamma, N = Spain_pop){
  sus = -beta * par[1] *par[2] / N
  rec = gamma * par[2]
  inf = -(sus + rec)
}
```

46

```r
N = 46.94e6
B = length(Spain_I)
b0 = 0

loss_function <- function(par){
  beta <- par[1]
  gamma <- par[2]
  sse = sum(abs(sir(N, B, b0, beta, gamma, h)$inf -
df_Spain_I$Spain_Inf/N))
}
```

```r
optim(c(0.1, 0.1), loss_function)$par
```

```
[1] 8.3683 8.2204
```

Plot the fitting

```r
N = 46.94e6
B = length(Spain_I)
b0 = 0
beta = 8.3683
gamma = 8.2204
h = 1

result <- sir(N, B, b0, beta, gamma, h)

result <- result %>%
  gather('sus', 'inf', 'rev', key = 'type', value = 'ratio')

g1 <- ggplot(data = result %>% filter(type == 'inf')) +
  geom_line(mapping = aes(x = time, y = ratio, color = type)) +
  geom_point(data = df_Spain_I, mapping = aes(x = day, y = Spain_Inf/N))+
  theme_light() +
  labs(x = "time",
       y = "ratio",
       title = "the SIR model",
       colors = "state")
g1
```

47

Plot the prediction

```r
Spain_I <- diff(Spain_data3)
Spain_I <- as.vector(Spain_I)[0:150]
day_ls <- 1:length(Spain_I)

df_Spain_I <- data.frame(day = day_ls, Spain_Inf = Spain_I) %>%
  mutate(Spain_Inf = ifelse(Spain_Inf >= 0, Spain_Inf, 0))

N = 46.94e6
B = length(Spain_I)
b0 = 0
beta = 8.3683
gamma = 8.2204
h = 1

result <- sir(N, B, b0, beta, gamma, h)
result <- result %>%
  gather('sus', 'inf', 'rev', key = 'type', value = 'ratio')

g2 <- ggplot() +
  geom_point(data = df_Spain_I,mapping = aes(x = day, y = Spain_Inf/N)) +
  geom_line(data = result %>% filter(type == 'inf'),
            mapping = aes(x = time, y = ratio, color = type)) +
  theme_light() +
  labs(x = "time",
       y = "number of new reports",
       title = "the Inf in Spain")
g2
```

Other analysis

```r
colnames(features_select)
features_select_std <- features_select %>%
  mutate(Logged.GDP.per.capita = scale(Logged.GDP.per.capita),
         Social.support = scale(Social.support),
         Healthy.life.expectancy = scale(Healthy.life.expectancy),
         Freedom.to.make.life.choices =
scale(Freedom.to.make.life.choices),
         Generosity = scale(Generosity),
         Perceptions.of.corruption = scale(Perceptions.of.corruption))

head(features_select_std)
```

R Console

tbl_df
6 x 7

Country.Region <chr>	Logged.GDP.per.capita <dbl>	Social.support <dbl>
Afghanistan	-1.4766160	-2.66668365
Albania	0.1198818	-1.07615635
Algeria	0.2179010	-0.02759412
Argentina	0.4408231	0.74255643
Armenia	-0.1393494	-0.39138590
Australia	1.1836304	1.09351769

6 rows | 1-3 of 7 columns

48

Correlation plot

```
```{r}
corrplot.mixed(cor(features_select_std %>%
select(-Country.Region)),lower.col = "black",number.cex = 0.7)
corrplot(cor(features_select_std %>% select(-Country.Region)), method =
"color")
```
```

PCA

```
```{r}
library(factoextra)
options(digits = 5)
pca=prcomp(features_select_std %>% select(-Country.Region))
par(mfrow=c(1,3))
summary(pca)
fviz_pca_var(pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
fviz_contrib(pca, choice = "var", axes = 1, top = 10)
# Contributions of variables to PC2
fviz_contrib(pca, choice = "var", axes = 2, top = 10)
# Contributions of variables to PC3
fviz_contrib(pca, choice = "var", axes = 3, top = 10)
```
```

Healthy life expectancy vs case in million plot

```
```{r}
g3 <- ggplot(data = data_all, mapping = aes(x = Healthy.life.expectancy, y
= case_in_million, col = region)) +
  geom_point() +
  theme_light()
g3
```
```

Other features analysis (medical services)

```
coviddata<-read.csv("/Users/wowam/Desktop/owid-covid-data.csv")
head(coviddata)
coviddata<-coviddata[coviddata$date=="2020-11-30",]
coviddata<-coviddata[!(coviddata$location %in% c("World","International")),]

ggplot(coviddata, aes(x=hospital_beds_per_thousand, y=total_cases_per_million,
                      color = continent)) + geom_point() +
  labs(x="Number of Hospital Beds(per thousand)", y="Death Cases in million") +
  labs(title = "Number of Hospital Beds vs Deaths")

ggplot(coviddata, aes(x=handwashing_facilities, y=total_cases_per_million,
                      color = continent)) + geom_point() +
  labs(x="Handwashing Facilities", y="Death Cases in million") +
  labs(title = "Handwashing Facilitie vs Deaths")

model <- lm(total_cases_per_million~hospital_beds_per_thousand+handwashing_facilities, coviddata)
summary(model)
plot(model)
```