COVID Project

RenDerMCh

6/23/2020

Contents

INTRODUCTION

As per the instructions of this part of the Harvard Data Science Professional Certificate program we were asked to create our own project.

I have chosen to take data from Kaggle at a certain point of time in March 2020 (closer description in each data set).

As the COVID-19 pandemic was just starting Kaggle contained a data set for an objective of predicting the further development of the disease in the world. I have taken to predict only on prior data and economic statistical measures of involved countries.

This model is by far not exhaustive nor accurate as there are many more influences on the development of the disease.

Kaggle data loading

Kaggle team provided us with data set

test

```
# LOADING Kaggle Data Set
# original location: https://www.kaggle.com/c/covid19-global-forecasting-week-1/download/
#jKMTxvK1mqhnyEnsTq9H%2Fversions%2FcBgWUNa7eY5T7KL23gkv%2Ffiles%2Ftrain.csv
# moved to below location because I am not sure kaggle will be accessible in the future
# as it is a competition set downloaded from Kaggle on 25th March at 10:35 (GMT +8)
dl <- tempfile()
download.file("https://raw.githubusercontent.com/RenDerMch/COVID/master/train.csv", dl)
kaggle_train <- read_csv(dl)
```

```
## Parsed with column specification:
## cols(
##   Id = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character(),
##   Lat = col_double(),
##   Long = col_double(),
##   Date = col_date(format = ""),
##   ConfirmedCases = col_double(),
##   Fatalities = col_double()
## )
```

```
colnames(kaggle_train)[3]<-"country"
```

The initial data set comes from John Hopkings. The Kaggle_train dataset included 17892 observations. We are including the initial the first observations to demonstrate the data structure in Table 1.

Table 1: Kaggle data set - first 6 observations

Id	Province/State	country	Lat	Long	Date	ConfirmedCases	Fatalities
1	NA	Afghanistan	33	65	2020-01-22	0	0
2	NA	Afghanistan	33	65	2020-01-23	0	0
3	NA	Afghanistan	33	65	2020-01-24	0	0
4	NA	Afghanistan	33	65	2020-01-25	0	0
5	NA	Afghanistan	33	65	2020-01-26	0	0
6	NA	Afghanistan	33	65	2020-01-27	0	0

Loading GDP Data set

We laoded GDP data per country from below mentioned Wikipedia page - we used two tables. Primary was data provided by World Bank and secondary was CIA source (only for countries not provided by World Bank).

GDP is the basic econmic indicator and we are using GDP per capita to as to bring to each person. Using GDP as a virus devleopment indicator is two fold - 1.) we believe that the richer the country the more people

travel and the higher probability of "dragging" the virus into the country and 2.) the richer the country the better healthcare and better means to treat, trace and limit the virus.

Code used:

```
# LOADING Country list of GDP PPP per Capita
temp <- read_html("https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(PPP)_per_capita")
temp <- temp %>% html_nodes("table")
#created data frame with Country name and GDP PPP per capita (source World Bank)
gdp_worldbank <- temp[[4]] %>% html_table %>%
  mutate(temp=str_replace_all(`Int$`,"\\s\\(2017\\)",""),
         gdp_ppp=as.numeric(str_replace_all(temp,",","")),
         country=`Country/Territory`) %>%
  select(country,gdp_ppp)
#created data frame with Country name and GDP PPP per capita (source CIA) with countries
#that are not in World Bank data
gdp_cia <- temp[[5]] %>% html_table %>%
  mutate(temp=str_replace_all(`Int$`,"\\s\\(2017\\)",""),
         gdp_ppp=as.numeric(str_replace_all(temp,",","")),
         country=`Country/Territory`) %>%
  select(country,gdp_ppp) %>%
  anti_join(gdp_worldbank,by="country")
# created a data fram with merge join of both
gdp<-bind_rows(gdp_worldbank,gdp_cia)
```

The GDP dataset included 243 observations (country values). We are including the first observations to demonstrate the data structure in Table 2.

Table 2: GDP per country - first 6 observations

country	gdp_ppp
Macau	135121
Luxembourg	116786
Singapore	98827
Qatar	96804
Ireland	84459
Cayman Islands (2017)	69420

Loading Popluation Density Data set

We laoded Popluation Density data per country from below mentioned Wikipedia page.

The choice to include Population Density Data is because we believe that the higher the population density the higher the ability for the virus to spread as people are generally closer together.

Code used:

```
temp <- read_html("https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population_densi
temp <- temp %>% html_nodes("table")
temp <- temp[[1]] %>% html_table(fill=TRUE)
c<-temp[1,]
c[6]<-"density_km2"
c[2]<-"Country"
temp <- temp[-(1:4),]
colnames(temp)<-c
```

```
temp <- temp %>% select(Country,density_km2) %>%
  mutate(country=str_replace_all(Country,"\\s\\(.+?\\)",""),
         country=str_replace_all(country,"\\[.+?\\]",""),
         density_km2=as.numeric(str_replace_all(density_km2,",",""))) %>%
  select(country,density_km2)
c<-is.na(temp$density_km2)
c<-which(c=="TRUE")
#created data frame with Country name and Popuplation Density per km^2
dens<-temp[-c,]
```

The population density dataset included 250 observations (country values). We are including the first observations to demonstrate the data structure in Table 3.

Table 3: Population Density - first 6 observations

country	density_km2
Hong Kong	6781.83
Gibraltar	4956.03
Bahrain	1982.91
Malta	1566.85
Maldives	1257.63
Bermuda	1226.52

Loading Median Age Data set

We laoded Median Age data per country from below mentioned Wikipedia page.

As it is already widely known the older the person is the more prone they are to the virus.

Code used:

```
temp <- read_html("https://en.wikipedia.org/wiki/List_of_countries_by_median_age")
temp <- temp %>% html_nodes("table")
#created data frame with Country name and their Median Age in years
age <- temp[[1]] %>% html_table %>%
  mutate(country=`Country/Territory`,
         median_age=`Median(Years)`) %>%
  select(country,median_age)
```

The median age dataset included 230 observations (country values). We are including the first observations to demonstrate the data structure in Table 4.

Table 4: Median Age - first 6 observations

country	median_age
Afghanistan	18.9
Albania	32.9
Algeria	28.1
American Samoa	25.5
Andorra	44.3
Angola	15.9

Loading Household Size Data set

We laoded Household Size data per country from below mentioned UN page.

We believe that the more generations living in a household the higher chance for the virus to spread as people from varios walks of life meet together in one household.

Code used:

```
# original location: https://population.un.org/household/exceldata/population_division_
#UN_Houseshold_Size_and_Composition_2019.xlsx
# moved to below location just to be sure the data remains static
# downloaded from UN on 25th March at 12:34 (GMT +8)
dl <- tempfile()
download.file("https://github.com/RenDerMch/COVID/raw/master/population_division_UN_Houseshold_Size_and_
householdsize <- read_xlsx(dl,sheet="UN HH Size and Composition 2019")
```

```
## New names:
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * ... and 37 more problems
```

```
householdsize<-householdsize[-(1:4),c(1,3,4,41)]
colnames(householdsize)<-c("country","source","date","perc_multigen")
#created a tibble with Country name and Percentage of multi-generation households
householdsize <- householdsize %>%
  mutate(date=as.Date(as.numeric(date),origin="1899-12-30"),
         perc_multigen=as.numeric(perc_multigen))
householdsize <- householdsize[complete.cases(householdsize),] %>%
  arrange(desc(date)) %>% group_by(country) %>%
  summarize(perc_multigen=first(perc_multigen))
```

The Multigeneration Household dataset included 111 observations (country values). We are including the first observations to demonstrate the data structure in Table 5.

Table 5: MultiGeneration Households - first 6 observations

country	perc_multigen
Afghanistan	50.39215
Albania	44.64797
Angola	17.53955
Argentina	29.58085
Armenia	54.30326
Azerbaijan	48.61187

Loading Physicians Number Data set

We laoded Physicians per 1000 people data per country from below mentioned World Bank page.

We believe that the higher the amount of Physicians per 1000 people, the more effective the health care system in managing the patients and decreasing the spread of the virus.

Code used:

```
dl <- tempfile()
download.file("http://api.worldbank.org/v2/en/indicator/SH.MED.PHYS.ZS?downloadformat=excel",
              dl)
physicians <- read_xls(dl,sheet = "Data")
```

```
## New names:
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * `` -> ...7
## * ... and 57 more problems
```

```
c<-physicians[3,]
physicians <-physicians[-(1:3),]
colnames(physicians)<-c
#created a tibble with Country name and Number of Physicians per 1000 people
physicians <- physicians %>%
  gather(year,physicians_per_1000,`1960`:`2019`,na.rm = TRUE) %>%
  mutate(country=`Country Name`,
         physicians_per_1000=as.numeric(physicians_per_1000)) %>%
  select(country,year,physicians_per_1000) %>%
  arrange(desc(year)) %>% group_by(country) %>%
  summarize(physicians_per_1000=first(physicians_per_1000))
```

The Physicians dataset included 253 observations (country values). We are including the first observations to demonstrate the data structure in Table 6.

Table 6: Physicians per 1000 people - first 6 observations

country	physicians_per_1000
Afghanistan	0.2782
Albania	1.2164
Algeria	1.7193
American Samoa	0.7810
Andorra	3.3333
Angola	0.2146

Loading Hospital Bed Number Data set

We laoded Hospital Beds per 1000 people data per country from below mentioned World Bank page.

We believe that the higher the amount of Beds per 1000 people, the better the option to isolate infected and therefore decreasing the spread of the virus.

Code used:

```
dl <- tempfile()
download.file("http://api.worldbank.org/v2/en/indicator/SH.MED.BEDS.ZS?downloadformat=excel",
              dl)
beds <- read_xls(dl,sheet = "Data")
```

```
## New names:
## * `` -> ...3
```

```
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * `` -> ...7
## * ... and 57 more problems
```

```
c<-beds[3,]
beds <-beds[-(1:3),]
colnames(beds)<-c
#created a tibble with Country name and Number of beds per 1000 people
beds <- beds %>%
  gather(year,beds_per_1000,`1960`:`2019`,na.rm = TRUE) %>%
  mutate(country=`Country Name`,
         beds_per_1000=as.numeric(beds_per_1000)) %>%
  select(country,year,beds_per_1000) %>%
  arrange(desc(year))  %>% group_by(country) %>%
  summarize(beds_per_1000=first(beds_per_1000))
```

The Hospital Beds dataset included 246 observations (country values). We are including the first observations to demonstrate the data structure in Table 7.

Table 7: Hospital beds per 1000 people - first 6 observations

country	beds_per_1000
Afghanistan	0.5
Albania	2.9
Algeria	1.9
Andorra	2.5
Angola	0.8
Antigua and Barbuda	3.8

Joining and Cleaning data sets

After loading the data sets we had to merge them for the purposes of model aplication.

First we had to modify the data in the separate data sets especially in regards to the coutry name as different sources use different country names and formats.

Example of modifying country names:

```
# cleaning up GDP table for better fit for joining by replacing Country names to match
# countries is Kaggle source data
gdp[,1]<- gdp$country %>%
  str_replace_all("Czech Republic","Czechia") %>%
  str_replace_all("Congo, Republic of the","Republic of the Congo") %>%
  str_replace_all("Congo, Democratic Republic of the","Congo (Kinshasa)") %>%
  str_replace_all("Côte d'Ivoire","Cote d'Ivoire") %>%
  str_replace_all("Taiwan","Taiwan*") %>%
  str_replace_all("United States","US") %>%
  str_replace_all("The Bahamas","Bahamas, The") %>%
  str_replace_all("The Gambia","Gambia, The")
```

we will not be including this part of code for all the data sets.

Joining the datasets together.

Code used:

```
# joining GDP to Kaggle_train set and creating data_set
data_set <- kaggle_train %>% left_join(gdp,by="country")
# joining Population Density to the data_set
data_set <- data_set %>% left_join(dens,by="country")
# joining Population AGE to the data_set
data_set <- data_set %>% left_join(age,by="country")
# joining Population household multigen to the data_set
data_set <- data_set %>% left_join(householdsize,by="country")
# joining Population Physicians to the data_set
data_set <- data_set %>% left_join(physicians,by="country")
# joining Population Beds to the data_set
data_set <- data_set %>% left_join(beds,by="country")
```

As there were multiple NAs introduced during coersions we need to deal with them. We decided to replace the NAs with an average of the observed values.

Code used:

```
#dealing with NAs
# findging the average for GDP
mean_gdp<-mean(data_set$gdp_ppp, na.rm = TRUE)
# findging the average for density
mean_dens<-mean(data_set$density_km2, na.rm = TRUE)
# findging the average for mean age
mean_age<-mean(data_set$median_age, na.rm = TRUE)
# findging the average for multigeneration households
mean_multi<-mean(data_set$perc_multigen, na.rm = TRUE)
 # findging the average for number of physicians
mean_phy<-mean(data_set$physicians_per_1000, na.rm = TRUE)
# findging the average for GDP for number of beds
mean_beds<-mean(data_set$beds_per_1000, na.rm = TRUE)

data_set<- data_set %>%  #replacing NAs with average for each of the statistical measures
  mutate(gdp_ppp=ifelse(is.na(gdp_ppp),mean_gdp,gdp_ppp),
         density_km2=ifelse(is.na(density_km2),mean_dens,density_km2),
         median_age=ifelse(is.na(median_age),mean_age,median_age),
         perc_multigen=ifelse(is.na(perc_multigen),mean_multi,perc_multigen),
         physicians_per_1000=ifelse(is.na(physicians_per_1000),
                                    mean_phy,physicians_per_1000),
         beds_per_1000=ifelse(is.na(beds_per_1000),mean_beds,beds_per_1000))

# adding region as a combination of province and country as there are for example
# 2 Georgias (once a country and once a state)
data_set3 <- data_set %>% mutate(region=ifelse(is.na(`Province/State`),country,
                                 paste(country,"/",`Province/State`)), prev_day=0)

# removing China as we are not focusing on China
data_set2 <- data_set3 %>% filter(!country=="China")
```

We also added a new column named region that combines the information for Province/State and Country as there are multiple instances of repetitions - for example Georgia can be both an asian Country as well as a State in the United States.

Also we are removing China as we will be focusing only on the rest of the world where the infection is only starting since China is already leveling off.

Splitting data set into TRAINING and TEST sets

As we will be applying different models to test which will be the best suiting for our purpose we will split the Kaggle training set into TRAINIG and TEST sets on which we will test our models. Do please note that this does not present overtraing as a final Kaggle Test data set will be loaded for the final prognosis.

Code used:

```
# adding previous day into the data set
for (i in 2:nrow(data_set2)) {
     data_set2$prev_day[i] <- data_set2$ConfirmedCases[i-1]
}

#rearrenging columns
data_set2 <- data_set2[,c(1,15,2,3,6,7,16,8:14,4,5)]

# setting prev_day=0 in for each change of state
data_set2 <- data_set2 %>%
  mutate(growth=
            ifelse(ConfirmedCases/prev_day<1|prev_day==0,
                   1,ConfirmedCases/prev_day))

# rearrenging columns
data_set2 <- data_set2[,c(1:7,17,8:16)]

# splitting the whole TRAINING SET into Train and Test based on date
train_index<- which(data_set2$Date<="2020-03-11")
train_set<-data_set2[train_index,]
test_set<-data_set2[-train_index,]
```

During the same part of code we also decided to introduces a parameter "growth" that shows the growth from one day to another as that is what we are going to be predicting.

ANALYSIS and MODELS

Loss function setup

We will be using two different loss functions Root Mean Square Error (RMSE) and Root Mean Squared Log Error (RMSLE). Where we will pay more attention to RMSLE as per Kaggle dataset instruction.

```
RMSLE <- function(true, predicted){
  sqrt(mean((log(predicted+1) - log(true+1))^2))
}

RMSE <- function(true, predicted){
  sqrt(mean((true - predicted)^2))
}
```

Generalized linear model for Date only

We are starting our models with a GLM model predicting Growth only on Date itself.

```
train_glm <- train(growth ~ Date,  #training for GLM given parameters
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
```

```
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
# predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date only is as follows:

```
results <- data_frame(method = "GLM - date",
                      RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                      RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                      REAL= temp1$conf, PRED= temp1$pred,
                      DISCREPANCY = temp1$dis)  #creating table with results

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 8: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.3

This should be the begging and we will try to do better.

Generalized linear model for Date and GDP

Next model is a GLM model predicting Growth on Date and GDP.

```
train_glm <- train(growth ~ Date + gdp_ppp,
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
```

```
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date and GDP is as follows:

```
results <- bind_rows(results,                # adding line to the result table
             data_frame(method = "GLM - date + GDP",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 9: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56

We will continue adding parameters to try to get a better result.

Generalized linear model for Date, GDP and Population Density

Next model is a GLM model predicting Growth on Date, GDP and Population Density.

```
train_glm <- train(growth ~ Date + gdp_ppp  + density_km2,
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
```

```
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date, GDP and Population Density is as follows:

```
results <- bind_rows(results,                # adding line to the result table
             data_frame(method = "GLM - date + GDP + density",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 10: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14

We see that population density actually introduces worse results so we will not continue using it.

Generalized linear model for Date, GDP and Median Age

Next model is a GLM model predicting Growth on Date, GDP and Median Age.

```
train_glm <- train(growth ~ Date + gdp_ppp + median_age,
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date, GDP and Median Age is as follows:

```
results <- bind_rows(results,                    # adding line to the result table
             data_frame(method = "GLM - date + GDP + median age",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 11: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14
GLM - date + GDP + median age	1.410285	1926.851	312183	268918.8	43264.15

There is progress from previous model so lets continue.

Generalized linear model for Date, GDP, Median Age and Multigeneration Household Percentage

Next model is a GLM model predicting Growth on Date, GDP, Median Age and Multigeneration Household Percentage.

```
train_glm <- train(growth ~ Date + gdp_ppp + median_age  + perc_multigen,
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date, GDP, Median Age and Multigeneration Household Percentage is as follows:

```
results <- bind_rows(results,                 # adding line to the result table
             data_frame(method = "GLM - date + GDP + median age + multigen",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 12: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14
GLM - date + GDP + median age	1.410285	1926.851	312183	268918.8	43264.15
GLM - date + GDP + median age + multigen	1.410890	1856.798	312183	284384.2	27798.85

Again we see no progress in RMSLE loss function and although the discrepancy is decreased this could be caused only by 1 larger outlier while RMSLE is more objective. So we will continue and not use Multigeneration Household Percentage.

Generalized linear model for Date, GDP, Median Age and Number of Physicians for 1000 citizens.

Next model is a GLM model predicting Growth on Date, GDP, Median Age and Number of Physicians for 1000 citizens.

```
train_glm <- train(growth ~ Date + gdp_ppp + median_age + physicians_per_1000,
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date, GDP, Median Age and Number of Physicians for 1000

citizens is as follows:

```
results <- bind_rows(results,                    # adding line to the result table
             data_frame(method = "GLM - date + GDP + median age + physicians",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 13: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14
GLM - date + GDP + median age	1.410285	1926.851	312183	268918.8	43264.15
GLM - date + GDP + median age + multigen	1.410890	1856.798	312183	284384.2	27798.85
GLM - date + GDP + median age + physicians	1.410388	1844.965	312183	265405.1	46777.94

Again we see no progress in RMSLE loss function and although the discrepancy is decreased this could be caused only by 1 larger outlier while RMSLE is more objective. We will try with the last parameter.

Generalized linear model for Date, GDP, Median Age and Number of Beds for 1000 citizens.

Next model is a GLM model predicting Growth on Date, GDP, Median Age and Number of Beds for 1000 citizens.

```
train_glm <- train(growth ~ Date + gdp_ppp + median_age + beds_per_1000,
                   method="glm", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Generalized linear model for Date, GDP, Median Age and Number of Beds for 1000 citizens is as follows:

```
results <- bind_rows(results,                 # adding line to the result table
             data_frame(method = "GLM - date + GDP + median age + beds",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 14: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14
GLM - date + GDP + median age	1.410285	1926.851	312183	268918.8	43264.15
GLM - date + GDP + median age + multigen	1.410890	1856.798	312183	284384.2	27798.85
GLM - date + GDP + median age + physicians	1.410388	1844.965	312183	265405.1	46777.94
GLM - date + GDP + median age + beds	1.411378	2021.929	312183	272151.2	40031.82

Again we see no progress in RMSLE loss function and although the discrepancy is decreased this could be caused only by 1 larger outlier while RMSLE is more objective.

We believe this is due to the fact that the parameters as Physicians per 1000 citizens, Beds per 1000 citizens and Multigeneration Household percentage are closely linked to GDP therefore we are actually reintroducing them to the model causing the increase in the loss function. The correlations coeficients are as follows:

Physicians per 1000 citizens and GDP per capita correlation coef. = 0.4572927

Beds per 1000 citizens and GDP per capita correlation coef. = 0.2298388

Multigeneration Household percentage and GDP per capita correlation coef. = -0.573147

Therefore we will revert to the model using only GDP, Median Age and Date and we will try a different model.

k-Nearest Neighbors for Date, GDP and Median Age

Next model is a KNN model predicting Growth on Date, GDP and Median Age.

```
train_glm <- train(growth ~ Date + gdp_ppp + median_age,
                   method="knn", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
```

```
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the k-Nearest Neighbor model for Date, GDP and Median Age is as follows:

```
results <- bind_rows(results,                 # adding line to the result table
             data_frame(method = "KNN - date + GDP + median age",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 15: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14
GLM - date + GDP + median age	1.410285	1926.851	312183	268918.8	43264.15
GLM - date + GDP + median age + multigen	1.410890	1856.798	312183	284384.2	27798.85
GLM - date + GDP + median age + physicians	1.410388	1844.965	312183	265405.1	46777.94
GLM - date + GDP + median age + beds	1.411378	2021.929	312183	272151.2	40031.82
KNN - date + GDP + median age	1.519416	497984.449	312183	27692692.1	-27380509.13

The result is so far away from the previously achieved results that we will not even try optimalization.

Projection Pursuit Regression for Date, GDP and Median Age

Next model is a PPR model predicting Growth on Date, GDP and Median Age.

```
train_glm <- train(growth ~ Date + gdp_ppp + median_age,
                   method="ppr", data=train_set)

#predicitng based on test_set (not overtraining as it is not the final set)
prediction_glm <- predict(train_glm,test_set)

#setting prediction equal to 0 for first day of the data set as we are
#predicting growth and that requires prior data
vec1<-which(test_set$Date=="2020-03-12")
vec2<-which(!test_set$Date=="2020-03-12")
predictions_gr <- prediction_glm
test_set<-test_set %>% mutate(prediction=0)
```

```
for (i in vec1) {
  test_set$prediction[i] <- (test_set$prev_day[i]*predictions_gr[i])
}
#calculating prediction as previous day * predicted growth
for (i in vec2) {
  test_set$prediction[i] <- (test_set$prediction[i-1]*predictions_gr[i])
}

temp1 <- test_set %>% filter(Date=="2020-03-24") %>%
  summarize(conf=sum(ConfirmedCases),pred=sum(prediction),dis=(conf-pred))
```

The result of the Projection Pursuit Regression model for Date, GDP and Median Age is as follows:

```
results <- bind_rows(results,                # adding line to the result table
             data_frame(method = "PPR - date + GDP + median age",
                RMSLE = RMSLE(test_set$ConfirmedCases,test_set$prediction),
                RMSE = RMSE(test_set$ConfirmedCases,test_set$prediction),
                REAL= temp1$conf, PRED= temp1$pred, DISCREPANCY = temp1$dis))

kable(results,caption = "Model Results",booktabs=TRUE) %>%
  kable_styling(full_width = F,
                latex_options = c("striped","hold_position"))
```

Table 16: Model Results

method	RMSLE	RMSE	REAL	PRED	DISCREPANCY
GLM - date	1.413739	1925.163	312183	246144.7	66038.30
GLM - date + GDP	1.410783	1895.531	312183	247396.4	64786.56
GLM - date + GDP + density	1.428067	2143.285	312183	270094.9	42088.14
GLM - date + GDP + median age	1.410285	1926.851	312183	268918.8	43264.15
GLM - date + GDP + median age + multigen	1.410890	1856.798	312183	284384.2	27798.85
GLM - date + GDP + median age + physicians	1.410388	1844.965	312183	265405.1	46777.94
GLM - date + GDP + median age + beds	1.411378	2021.929	312183	272151.2	40031.82
KNN - date + GDP + median age	1.519416	497984.449	312183	27692692.1	-27380509.13
PPR - date + GDP + median age	1.261791	5496.347	312183	633481.3	-321298.25

The result is worse than what we achieved with GLM (we are taking RMSE here into account as well) and therefore we will consider GLM as the final model.

RESULTS

Preparing the KAGGLE TEST data set

As we did not want to overtrain, we are only now loading the Kaggle test set and will modify and adjust it as per the principles discussed in the previous chapters while preparing the Train set.

```
# LOADING Kaggle Data Set
# original location: https://www.kaggle.com/c/covid19-global-forecasting-week-1/download/
#jKMTxvK1mqhnyEnsTq9H%2Fversions%2FcBgWUNa7eY5T7KL23gkv%2Ffiles%2Ftrain.csv
# moved to below location because I am not sure kaggle will be accessible in the future
# as it is a competition set downloaded from Kaggle on 25th March at 10:35 (GMT +8)
dl <- tempfile()
```

```
download.file("https://raw.githubusercontent.com/RenDerMch/COVID/master/test.csv", dl)
kaggle_test <- read_csv(dl)
```

```
## Parsed with column specification:
## cols(
##   ForecastId = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character(),
##   Lat = col_double(),
##   Long = col_double(),
##   Date = col_date(format = "")
## )
```

```
colnames(kaggle_test)[3]<-"country"
remove_index<- which(kaggle_test$Date<="2020-03-24")
kaggle_test<-kaggle_test[-remove_index,]
# creating region as combination of state and country and
#removing China - same as training set
kaggle_test <- kaggle_test %>% mutate(region=ifelse(is.na(`Province/State`),
                                  country,paste(country,"/",`Province/State`)))
kaggle_test <- kaggle_test %>% filter(!country=="China")

# joining GDP to Kaggle_train set and creating data_set
kaggle_test <- kaggle_test %>%
  left_join(gdp,by="country")

# joining Population AGE to the data_set
kaggle_test <- kaggle_test %>%
  left_join(age,by="country")

# adding a the last day of TRAIN set
last_day<- data_set2 %>% filter(Date=="2020-03-24") %>%
  mutate(prev_day=ConfirmedCases) %>% select(region,prev_day)

kaggle_test <- kaggle_test %>%
  left_join(last_day,by="region")

#dealing with NAs
mean_gdp<-mean(kaggle_test$gdp_ppp, na.rm = TRUE)
mean_age<-mean(kaggle_test$median_age, na.rm = TRUE)

kaggle_test<- kaggle_test %>%
  mutate(gdp_ppp=ifelse(is.na(gdp_ppp),mean_gdp,gdp_ppp),
         median_age=ifelse(is.na(median_age),mean_age,median_age))
```

Now we proceed with the projection with a GLM model using Date, GDP and Median age to predict the growth.

```
#training as per the chosen model
train_glm <- train(growth ~ Date + gdp_ppp + median_age,
                   method="glm", data=data_set2)

#predicting growth
prediction_glm <- predict(train_glm,kaggle_test)
```

```
vec1<-which(kaggle_test$Date=="2020-03-25")
vec2<-which(!kaggle_test$Date=="2020-03-25")
predictions_gr <- prediction_glm
kaggle_test<-kaggle_test %>% mutate(prediction=0)
for (i in vec1) {
  kaggle_test$prediction[i] <- (kaggle_test$prev_day[i]*predictions_gr[i])
}
for (i in vec2) {
  kaggle_test$prediction[i] <- (kaggle_test$prediction[i-1]*predictions_gr[i])
}

kaggle_result<- kaggle_test %>% select (ForecastId,`Province/State`,
                                        country,Lat,Long,Date,prediction)
```

CONCLUSION

We have predicted that solely based on Date and Economic date as of 23rd April 2020 there should have been 2083045290.05424 cases of COVID-19 in the world (aside from China).

Luckily this was not the case.