Final Assignment (Take Home Assignment)

Name: Greene Vu
Student ID: 3209044

Subject: Introduction to Data Science 11372

Task 4 Documentation and Reporting:

Task 2

1. Add four new variables to the master dataframe ("CumCases", "CumDeaths", "CumRecovered", "CumTests").

```
> #Add four new variables to the master dataframe ("CumCases", "CumDeaths", "CumRecovered", "CumTests")
> #detached plyr as it affects on "group_by" fuction
> detach("package:plyr", unload = TRUE)
Error in detach("package:plyr", unload = TRUE) : invalid 'name' argument
> #"CumCases"
> covid19_df <- covid19_df%>%
+                   group_by(Country) %>%
+                   mutate(CumCases = cumsum(NewCases),CumDeaths = cumsum(NewDeaths),
+                          CumRecovered = cumsum(Recovered),CumTests = cumsum(NewTests))
> View(covid19_df)
>
```

```
> head(covid19_df)
# A tibble: 6 x 17
# Groups:   Country [6]
  Code  Country     Date       Continent NewCases NewDeaths Recovered NewTests Population    GDP GDPCapita Month  Week
  <chr> <chr>       <date>     <chr>         <int>     <int>     <dbl>    <dbl>      <int>  <int>     <int> <dbl> <dbl>
1 AFG   Afghanistan 2020-01-01 Asia             0         0         0        0   37172386 2.20e4       619     1     1
2 DZA   Algeria     2020-01-01 Africa           0         0         0        0   42228429 1.68e5      4055     1     1
3 ARM   Armenia     2020-01-01 Europe           0         0         0        0    2951776 1.15e4      3937     1     1
4 AUS   Australia   2020-01-01 Oceania          0         0         0        0   24992369 1.41e6     57613     1     1
5 AUT   Austria     2020-01-01 Europe           0         0         0        0    8847037 4.17e5     47718     1     1
6 AZE   Azerbaijan  2020-01-01 Europe           0         0         0        0    9942334 4.07e4      4146     1     1
# … with 4 more variables: CumCases <int>, CumDeaths <int>, CumRecovered <dbl>, CumTests <dbl>
>
```

2. Add two new variables to the master dataframe ("Active", "FatalityRate").

```
#Add two new variables to the master dataframe ("Active", "FatalityRate")
#Active
covid19_df <- transform(covid19_df, Active = CumCases - (CumDeaths + CumRecovered))
#FatalityRate
covid19_df <- transform(covid19_df, FatalityRate = CumDeaths/CumCases)
```

```
> colnames(covid19_df)
 [1] "Code"         "Country"      "Date"         "Continent"    "NewCases"     "NewDeaths"
 [7] "Recovered"    "NewTests"     "Population"   "GDP"          "GDPCapita"    "Month"
[13] "Week"         "CumCases"     "CumDeaths"    "CumRecovered" "CumTests"     "Active"
[19] "FatalityRate"
>
```

3. Add four new variables to the master dataframe ("Cases_1M_Pop", "Deaths_1M_Pop", "Recovered_1M_Pop", "Tests_1M_Pop")

```
#Add four new variables ("Cases_1M_Pop", "Deaths_1M_Pop", "Recovered_1M_Pop", "Tests_1M_Pop")
#Cases_1M_Pop
covid19_df$CumTests
covid19_df <- transform(covid19_df, Cases_1M_Pop = CumCases*(10^6) / Population)
#Deaths_1M_Pop
covid19_df <- transform(covid19_df, Deaths_1M_Pop = CumDeaths*(10^6) / Population)
#Recovered_1M_Pop
covid19_df <- transform(covid19_df, Recovered_1M_Pop = CumRecovered*(10^6) / Population)
#Tests_1M_Pop
covid19_df <- transform(covid19_df, Tests_1M_Pop = CumTests*(10^6) / Population)
```

```
> colnames(covid19_df)
 [1] "Code"           "Country"        "Date"           "Continent"      "NewCases"       "NewDeaths"
 [7] "Recovered"      "NewTests"       "Population"     "GDP"            "GDPCapita"      "Month"
[13] "Week"           "CumCases"       "CumDeaths"      "CumRecovered"   "CumTests"       "Active"
[19] "FatalityRate"   "Cases_1M_Pop"   "Deaths_1M_Pop"  "Recovered_1M_Pop" "Tests_1M_Pop"
>
```

4. Find the day with the highest reported death toll across the world. Print the date and the death toll of that day.

```
#Find the day with the highest reported death toll across the world.
daily_death <- covid19_df %>% group_by(Date) %>%
  summarise(sumDeath = sum(NewDeaths))
print(summarise(daily_death, max_death_day = Date[which.max(sumDeath)],
                             max_death = max(sumDeath)))
```

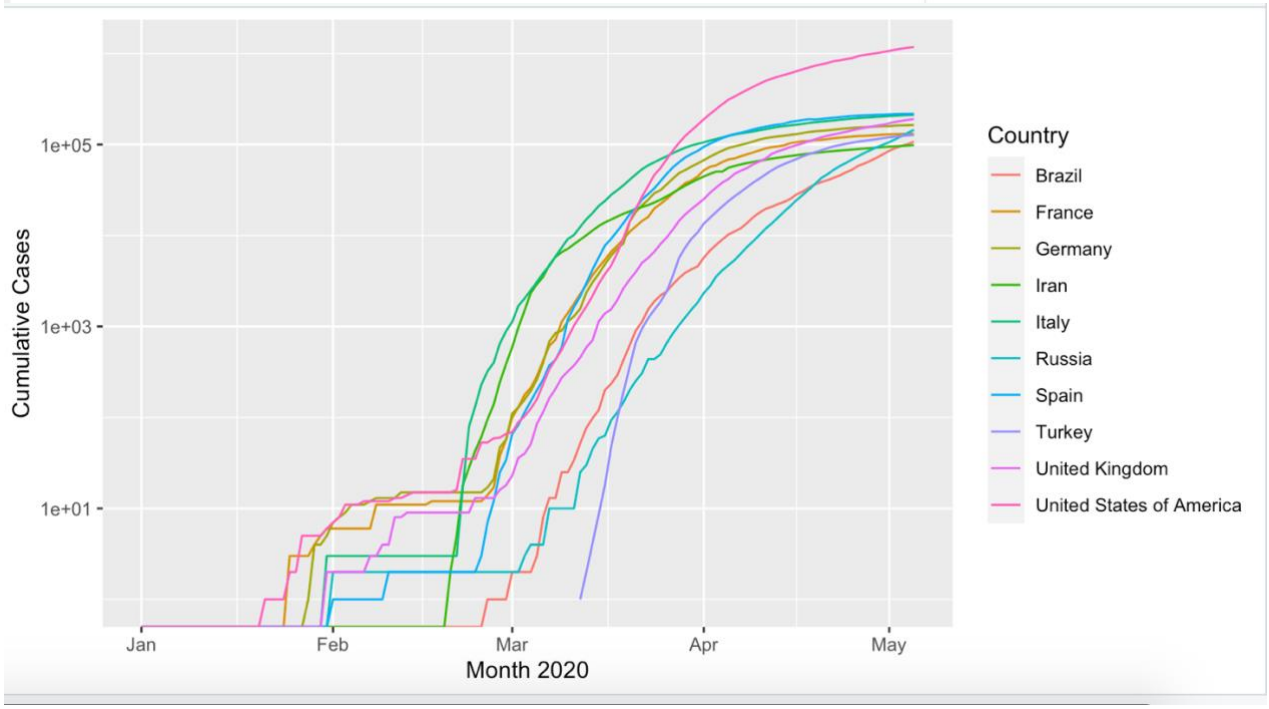
```
> #Find the day with the highest reported death toll across the world.
> daily_death <- covid19_df %>% group_by(Date) %>%
+   summarise(sumDeath = sum(NewDeaths))
> print(summarise(daily_death, max_death_day = Date[which.max(sumDeath)],
+                              max_death = max(sumDeath)))
# A tibble: 1 x 2
  max_death_day max_death
  <date>            <int>
1 2020-04-16        10520
>
```

5. Build a graph to show how the cumulative data of (Infected Cases, Deaths, Recovered, Tests) change over the time for the whole world collectively.

```
#Build a graph to show how the cumulative data of change over the time for the whole world collectively.
daily_info <- covid19_df %>% group_by(Date) %>%
  summarise(Cases = sum(CumCases), Deaths = sum(CumDeaths),
            Recovereds = sum(CumRecovered), Tests = sum(CumTests))

myplot_df<- gather(daily_info, key = key, value = value, -Date)

ggplot(myplot_df, aes(x = Date, y = value, color = key) )+
  geom_line()+
  xlab("Year 2020") +
  ylab("Number of People")+
  scale_y_log10()
```



6. Extract the last day (05/05/2020) data and save it in a separate dataframe called "lastDay_data".

```
#Extract the last day (05/05/2020) data
lastDay_data <- filter(covid19_df, Date == "2020-05-05")
```

Result:

```
> #Extract the last day (05/05/2020) data and save it in a separate dataframe called “lastDay_data”
> lastDay_data <- filter(covid19_df, Date == "2020-05-05")
> head(lastDay_data)
  Code     Country       Date     Continent NewCases NewDeaths Recovered NewTests Population    GDP
1  AFG Afghanistan 2020-05-05          Asia      190         5        24        0   37172386  21992
2  ALB     Albania 2020-05-05        Europe        8         0         0        0    2866376  13039
3  DZA     Algeria 2020-05-05        Africa      174         2        69        0   42228429 167555
4  AND     Andorra 2020-05-05        Europe        2         0        15        0      77006   3278
5  AGO      Angola 2020-05-05        Africa        0         0         0        0   30809762 126505
6  AIA    Anguilla 2020-05-05 North America        0         0         0        0      14731    311
  GDPCapita Month Week CumCases CumDeaths CumRecovered CumTests Active FatalityRate Cases_1M_Pop
1       619     5   18     2894        90          421        0   2383   0.03109883    77.853490
2      4450     5   18      803        31            0        0    772   0.03860523   280.144684
3      4055     5   18     4648       465         2064        0   2119   0.10004303   110.068030
4     39153     5   18      750        45          513        0    192   0.06000000  9739.500818
5      4247     5   18       35         2           11        0     22   0.05714286     1.136004
6     29493     5   18        3         0            0        0      3   0.00000000   203.652162
  Deaths_1M_Pop Recovered_1M_Pop Tests_1M_Pop
1    2.42115209       11.3256114            0
2   10.81505008        0.0000000            0
3   11.01153917       48.8770255            0
4  584.37004909     6661.8185596            0
5    0.06491449        0.3570297            0
6    0.00000000        0.0000000            0
```

7. Extract the whole records of the top 10 countries worldwide that have current active cases, total confirmed cases, and fatality rate in separate dataframes (i.e. top10activeW, top10casesW, top10fatalityW, top10testsMW).

```
#extract the whole records of the top 10 countries worldwide
#top10activeW
top10activeW <- head(lastDay_data[order(lastDay_data$Active, decreasing = TRUE),] , n=10)
#top10casesW
top10casesW <- head(lastDay_data[order(lastDay_data$CumCases, decreasing = TRUE),] , n=10)
#top10fatalityW
top10fatalityW <- head(lastDay_data[order(lastDay_data$FatalityRate, decreasing = TRUE),] , n=10)
#top10testsMW
top10testsMW <- head(lastDay_data[order(lastDay_data$CumTests, decreasing = TRUE),] , n=10)
```

8. Based on the last day data, print the up to date confirmed, death, recovered cases as well as the tests for every continent.

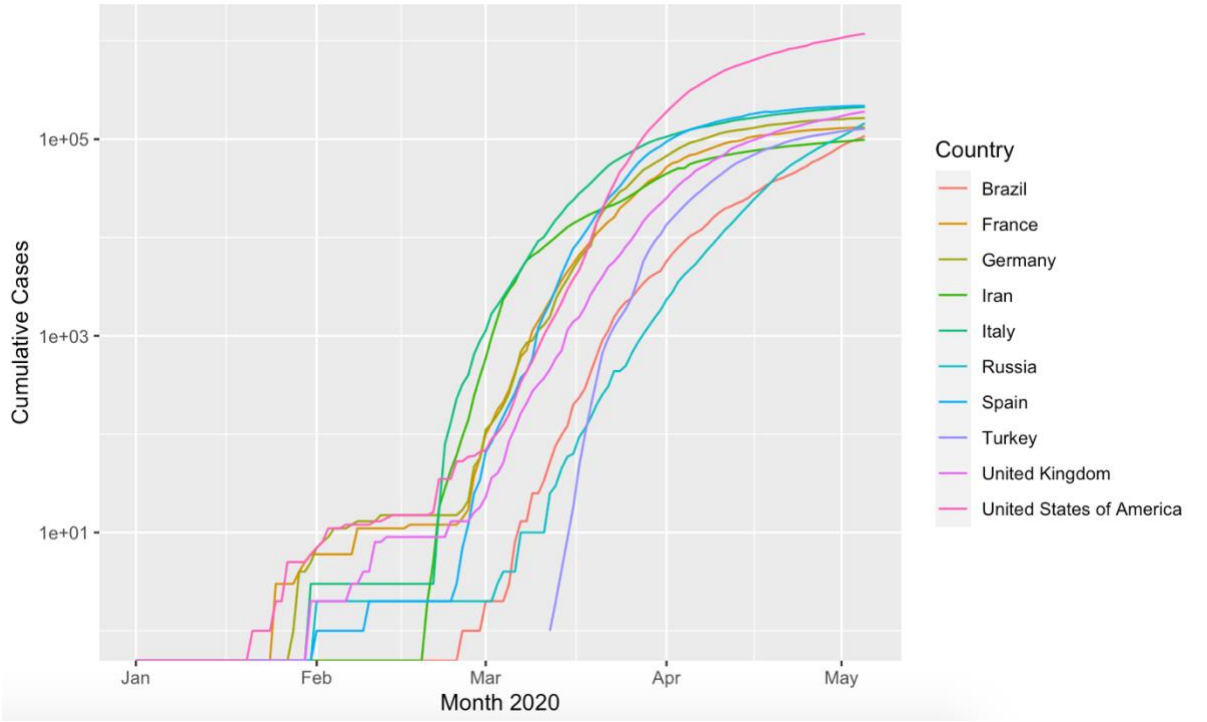
```
#print the up to date confirmed, death, recovered cases as well as the tests for every continent.
print(lastDay_data %>% group_by(Continent) %>%
        summarise(Confirmed_Death = max(CumDeaths),
                  Confirmed_Cases = max(CumCases),
                  Confirmed_Recovered = max(CumRecovered)))
```

```
# A tibble: 6 x 4
  Continent     Confirmed_Death Confirmed_Cases Confirmed_Recovered
  <chr>                   <int>           <int>               <dbl>
1 Africa                    465            7220                2746
2 Asia                     6277          127659               80475
3 Europe                  29079          218011              135100
4 North America           68934         1180634              189791
5 Oceania                    95            6825                5975
6 South America            7321          107780               48221
>
```

9. Build a graph to show the total number of cases over the time for the top 10 countries that have been obtained in question 7 (Use log for Y axis for better presentation).

```
#9.Build a graph to show the total number of cases over the time for
#the top 10 countries that have been obtained in question 7
top10_cases_byCountry <- filter(covid19_df, Country %in% c( "United States of America",
                                                "Spain","Italy","United Kingdom","Germany","Russia",
                                                "France","Turkey","Brazil","Iran"))

#gather informaton to plot
top10_cases_byCountry <- top10_cases_byCountry %>% arrange(Date, Country) %>% group_by(Country) %>%
                         select(Date,Country,CumCases)
#Plot
ggplot(top10_cases_byCountry, aes(x = Date, y = CumCases, color = Country) )+
  geom_line()+
  xlab("Month 2020") +
  ylab("Cumulative Cases")+
  scale_y_log10()
```



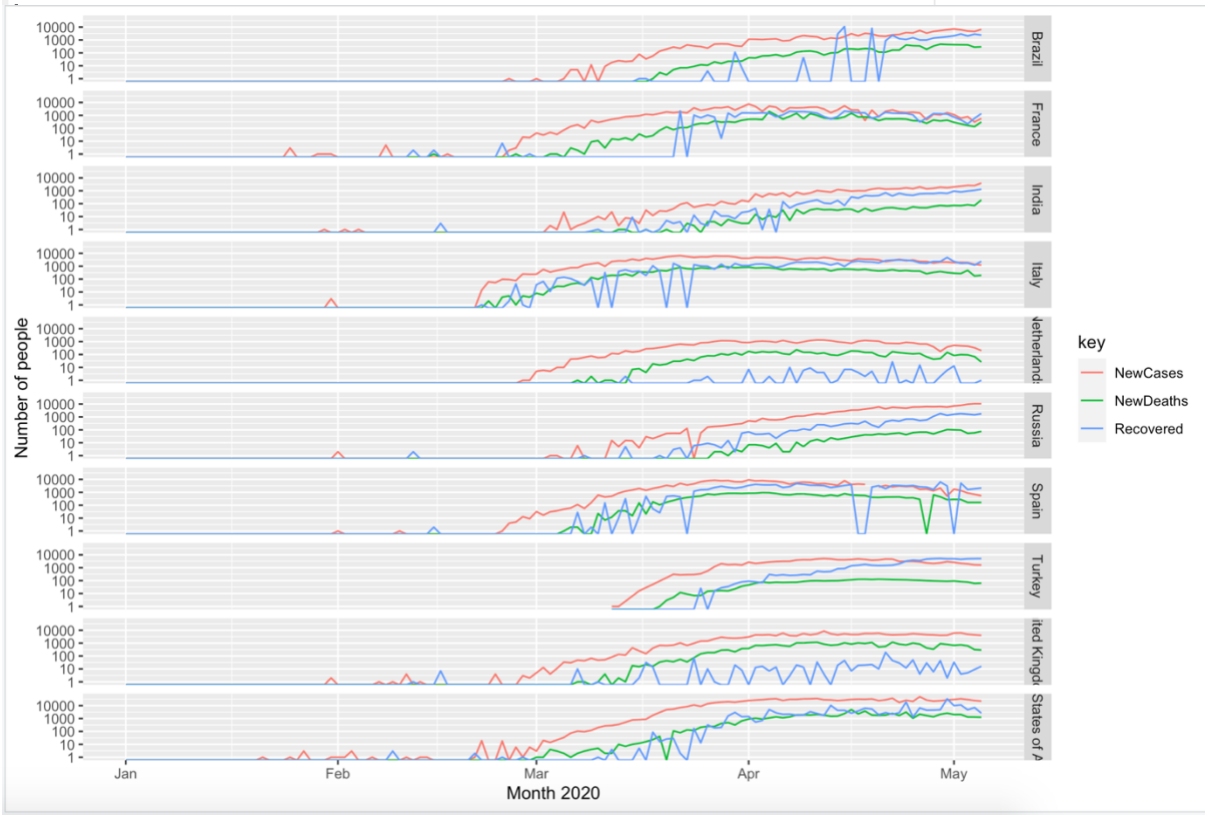
10. Build a graph for the top 10 countries with current highest active cases which was obtained previously in question 7

```
#10. Build a graph for the top 10 countries with current highest active cases
top10_active_byCountry <- filter(covid19_df, Country %in% top10activeW$Country)

top10_active_byCountry <- top10_active_byCountry %>% arrange(Date, Country) %>% group_by(Country) %>%
  select(Date,Country,NewCases, NewDeaths, Recovered)

#Choose Data for the graph
top10_active_byCountry <- gather(top10_active_byCountry, key = key, value = value, -Date, -Country)

ggplot(top10_active_byCountry, aes(x = Date, y = value, color = key) )+
  geom_line()+
  facet_grid(Country~.)+
  xlab("Month 2020") +
  ylab("Number of people")+
  scale_y_log10()
```



Task 3

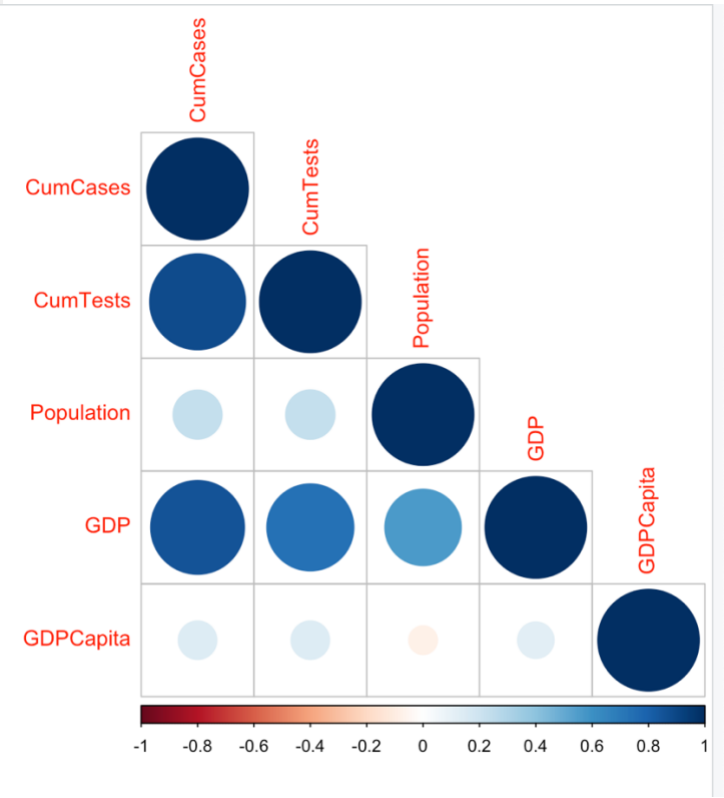
1. Based on the data of the last day, that you have extracted in the previous task, create a separate dataframe named "cor_data" with the data of these variables (CumCases, CumTests, Population, GDP, GDPCapita).

```
#Based on lastDay_data, create a separate dataframe named "cor_data"
cor_data <- lastDay_data %>%
  select(CumCases, CumTests, Population, GDP, GDPCapita)
```

```
> colnames(cor_data)
[1] "CumCases"   "CumTests"   "Population" "GDP"        "GDPCapita"
>
```

2. Compute the correlation matrix between the variables of the "cor_data" and visualise this correlation matrix.

```
#Compute the correlation matrix between the variables of the "cor_data" and visualise this correlation matrix.
#correlation
M <-cor(cor_data)
#loading package
library(corrplot)
#Visualise
corrplot(M, type="lower")
```



3. Divide the cor_data into training and testing, where training data represent 65% of the number of rows.

```
#Divide the cor_data into training and testing, where training data represent 65%
#loading package
library(caret)
#use caret function to split, SplitRatio for 65%:35% splitting
cor_data1<-createDataPartition(cor_data$CumCases,p=.65,list=FALSE)
#subsetting into Train data
train<-cor_data[cor_data1,]
#subsetting into Test data
test<-cor_data[-cor_data1,]
```

```
> dim(train)
[1] 136   5
> dim(test)
[1] 71  5
>
```

4. Train a linear regression model to predict cumulative cases from the GDP of the countries. Then, evaluate this model on the test data and print the root mean square error value.

```
#Train a linear regression model to predict cumulative cases from the GDP of the countries.
#building model
lr_model<-lm(CumCases~GDP,data=train)
summary(lr_model)
#evaluate this model on the test data and print the root mean square error value.
test$PreditedCumCases<-predict(lr_model,test)
head(test)
```

```
> #Train a linear regression model to predict cumulative cases from the GDP of the countries.
> #building model
> lr_model<-lm(CumCases~GDP,data=train)
> summary(lr_model)

Call:
lm(formula = CumCases ~ GDP, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-103377    -981    2833    3276  143187

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.311e+03  2.014e+03  -1.644    0.103
GDP          5.945e-02  1.130e-03  52.623   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22910 on 134 degrees of freedom
Multiple R-squared:  0.9538,    Adjusted R-squared:  0.9535
F-statistic:  2769 on 1 and 134 DF,  p-value: < 2.2e-16

> #evaluate this model on the test data and print the root mean square error value.
> test$PreditedCumCases<-predict(lr_model,test)
> head(test)
   CumCases CumTests Population    GDP GDPCapita PreditedCumCases
5        35        0   30809762 126505      4247       4209.81535
9      2507        0    2951776  11536      3937      -2624.95614
10      100        0     105845   2664     25655      -3152.38608
12    15621   285883    8847037 416835     47718      21469.59154
18    17489   211369    9485386  54441      5750        -74.30448
19    50267   372654   11422068 494763     43289      26102.31926
>
```

5. Train another linear regression model to predict cumulative cases from all the other variables. Then, evaluate this model on the test data and print the root mean square error value.

```
#Train another linear regression model to predict cumulative cases from all the other variables.
lr_model2<-lm(CumCases~.,data=train)
summary(lr_model2)

#evaluate this model on the test data and print the root mean square error value.
test$PreditedCumCases2<-predict(lr_model2,test)
head(test)
```

```
> #Train another linear regression model to predict cumulative cases from all the other variables.
> lr_model2<-lm(CumCases~.,data=train)
> summary(lr_model2)

Call:
lm(formula = CumCases ~ ., data = train)

Residuals:
   Min     1Q Median     3Q    Max
-79574   -485   1258   2672 124226

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.808e+02  2.075e+03  -0.376    0.707
CumTests     2.345e-02  4.165e-03   5.630 1.05e-07 ***
Population  -8.652e-05  1.407e-05  -6.151 8.71e-09 ***
GDP          5.245e-02  1.924e-03  27.261  < 2e-16 ***
GDPCapita   -7.091e-02  6.132e-02  -1.156    0.250
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18890 on 131 degrees of freedom
Multiple R-squared:  0.9693,    Adjusted R-squared:  0.9684
F-statistic:  1035 on 4 and 131 DF,  p-value: < 2.2e-16

> #evaluate this model on the test data and print the root mean square error value.
> test$PreditedCumCases2<-predict(lr_model2,test)
> head(test)
   CumCases CumTests Population    GDP GDPCapita PreditedCumCases PreditedCumCases2
5        35        0   30809762 126505      4247       4209.81535         2887.7568
9      2507        0    2951776  11536      3937      -2624.95614         -710.3054
10      100        0     105845   2664     25655      -3152.38608        -2469.3498
12    15621   285883    8847037 416835     47718      21469.59154        23637.1332
18    17489   211369    9485386  54441      5750        -74.30448         5802.6170
19    50267   372654   11422068 494763     43289      26102.31926        29850.4339
>
```