Take Home Final Exam

For the take-home part of the MSDS 401 Final Exam, you are tasked with analyzing data on new daily covid-19 cases and deaths in European Union (EU) and European Economic Area (EEA) countries. A data file may be downloaded here, *or* you may use the provided **read.csv()** code in the 'setup' code chunk below to read the data directly from the web csv. Either approach is acceptable; the data should be the same.

Once you have defined a data frame with the daily case and death and country data, you are asked to: (1) perform an Exploratory Data Analysis (EDA), (2) perform some hypothesis testing, (3) perform some correlation testing, and (4) fit and describe a linear regression model. Each of these four (4) items is further explained below and "code chunks" have been created for you in which to add your R code, just as with the R and Data Analysis Assignments. You may add additional code chunks, as needed. You should make comments in the code chunks or add clarifying text between code chunks that you think further your work.

A data dictionary for the dataset is available here.

Definitions:

- "Incidence rate" is equal to new daily cases per 100K individuals. Country population estimates can be found in 'popData2020.' You will calculate a daily incidence rate in item (1), for each country, that we will explore further in items (2) and (3).
- "Fatality rate" is equal to new daily deaths per 100K individuals. Country population estimates can be found in 'popData2020.' You will calculate a daily fatality rate in item (1), for each country, that we will explore further in items (2) and (3).

---

1. Descriptive Statistics

Perform an Exploratory Data Analysis (EDA). Your EDA is exactly that: yours. Your knit .html should include the visualizations and summary tables that you find valuable while exploring this dataset. **However**, at minimum, your EDA must include the following:

- Creation of a vector, 'incidence_rate,' equal to the daily new cases per 100K individuals, per country. Country populations are provided in 'popData2020.' This vector should be added to the 'data' data frame.
- Creation of a vector, 'fatality_rate,' equal to the new deaths per 100K individuals, per country. Country populations are provided in 'popData2020.' This vector should be added to the 'data' data frame.
- A visualization exploring new cases or incidence rates, per country, over time. You may choose a subset of countries, if you wish, but your visualization should include at least five (5) countries and include the entire time frame of the dataset.
- A visualization exploring new deaths or fatality rates, per country, over time. You may choose a subset of countries, if you wish, but your visualization should include at least five (5) countries.

- A table or visualization exploring some other aspect of the data. For example, you could explore case fatality rates per country; the number of deaths divided by the total number of cases. Note that to do this, you would want to like across the entire time of the dataset, looking at the total cases and deaths, per country.

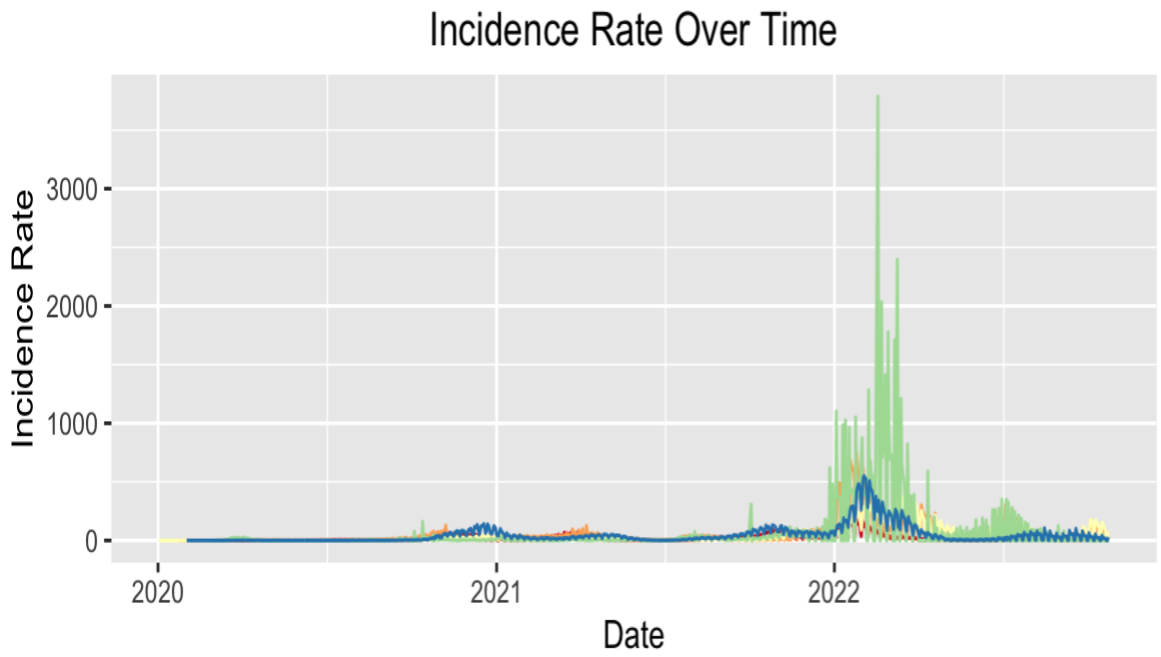
```
##Inspecting our data:


summary(data)
##     dateRep                day             month             year
##  Min.   :2020-01-01   Min.   : 1.00   Min.   : 1.000   Min.   :2020
##  1st Qu.:2020-10-17   1st Qu.: 8.00   1st Qu.: 4.000   1st Qu.:2020
##  Median :2021-06-17   Median :16.00   Median : 6.000   Median :2021
##  Mean   :2021-06-17   Mean   :15.68   Mean   : 6.431   Mean   :2021
##  3rd Qu.:2022-02-13   3rd Qu.:23.00   3rd Qu.: 9.000   3rd Qu.:2022
##  Max.   :2022-10-26   Max.   :31.00   Max.   :12.000   Max.   :2022
##
##      cases              deaths         countriesAndTerritories     geoId
##  Min.   :-348846   Min.   : -217.00   Finland  : 1024         FI     : 1024
##  1st Qu.:    111   1st Qu.:    0.00   France   : 1006         FR     : 1006
##  Median :    705   Median :    5.00   Czechia  : 1003         CZ     : 1003
##  Mean   :   6088   Mean   :   40.87   Lithuania:  997         LT     :  997
##  3rd Qu.:   3483   3rd Qu.:   31.00   Germany  :  992         DE     :  992
##  Max.   : 501635   Max.   :13743.00   Sweden   :  982         SE     :  982
##  NA's   :93        NA's   :292        (Other)  :22725         (Other):22725
##  countryterritoryCode  popData2020
##  FIN    : 1024         Min.   :   38747
##  FRA    : 1006         1st Qu.: 2095861
##  CZE    : 1003         Median : 6951482
##  LTU    :  997         Mean   :15348035
##  DEU    :  992         3rd Qu.:11522440
##  SWE    :  982         Max.   :83166711
##  (Other):22725
##looks like there are some negative numbers, specifically ones that don't make sense.
There probably wasn't -348846 cases in a country. Negative values are only appearing i
n the cases and death column


data %>% filter(data$cases < 0 | data$deaths < 0)
##       dateRep day month year   cases deaths countriesAndTerritories geoId
```

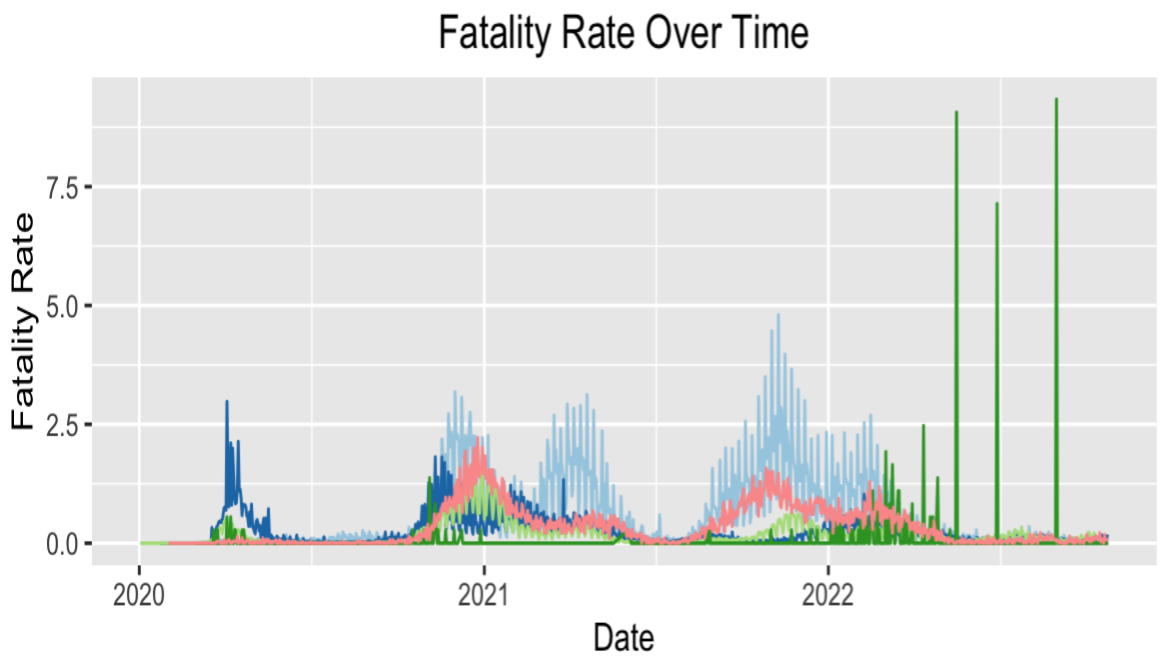
```
##  1  2022-01-21  21     1 2022   -3203      3                  Cyprus    CY
##  2  2021-03-24  24     3 2021   -2001      8                 Denmark    DK
##  3  2021-05-20  20     5 2021 -348846    133                  France    FR
##  4  2020-09-04   4     9 2020    8975    -20                  France    FR
##  5  2020-07-21  21     7 2020     584    -12                  France    FR
##  6  2020-06-02   2     6 2020    -766    107                  France    FR
##  7  2020-05-19  19     5 2020     524   -217                  France    FR
##  8  2020-01-27  27     1 2020      -3      0                  France    FR
##  9  2021-02-21  21     2 2021   -2859      0                  Greece    EL
## 10 2022-02-23  23     2 2022    3147     -1                 Iceland    IS
## 11 2021-10-04   4    10 2021    -968      0                 Iceland    IS
## 12 2022-06-09   9     6 2022    -228      0                 Ireland    IE
## 13 2022-02-15  15     2 2022   -5532     17                 Ireland    IE
## 14 2021-09-02   2     9 2021   -1707      0                 Ireland    IE
## 15 2021-05-07   7     5 2021     430     -3                 Ireland    IE
## 16 2020-12-08   8    12 2020     214     -2                 Ireland    IE
## 17 2020-11-23  23    11 2020     250     -1                 Ireland    IE
## 18 2020-10-25  25    10 2020    1020     -1                 Ireland    IE
## 19 2020-10-22  22    10 2020     -12      3                 Ireland    IE
## 20 2020-10-02   2    10 2020     466     -5                 Ireland    IE
## 21 2020-07-30  30     7 2020      85     -1                 Ireland    IE
## 22 2020-07-08   8     7 2020       4     -4                 Ireland    IE
## 23 2020-06-01   1     6 2020      72     -1                 Ireland    IE
## 24 2020-05-25  25     5 2020      59     -2                 Ireland    IE
## 25 2020-06-24  24     6 2020     577    -31                   Italy    IT
## 26 2020-06-19  19     6 2020    -148     47                   Italy    IT
## 27 2020-03-20  20     3 2020     -41      0               Lithuania    LT
## 28 2020-08-16  16     8 2020     -42      0                   Malta    MT
## 29 2022-03-11  11     3 2022    5385     -1                  Norway    NO
## 30 2021-11-08   8    11 2021    2390     -1                  Norway    NO
## 31 2021-06-07   7     6 2021     203     -1                  Norway    NO
##    countryterritoryCode popData2020
## 1                   CYP      888005
## 2                   DNK     5822763
## 3                   FRA    67320216
## 4                   FRA    67320216
## 5                   FRA    67320216
```

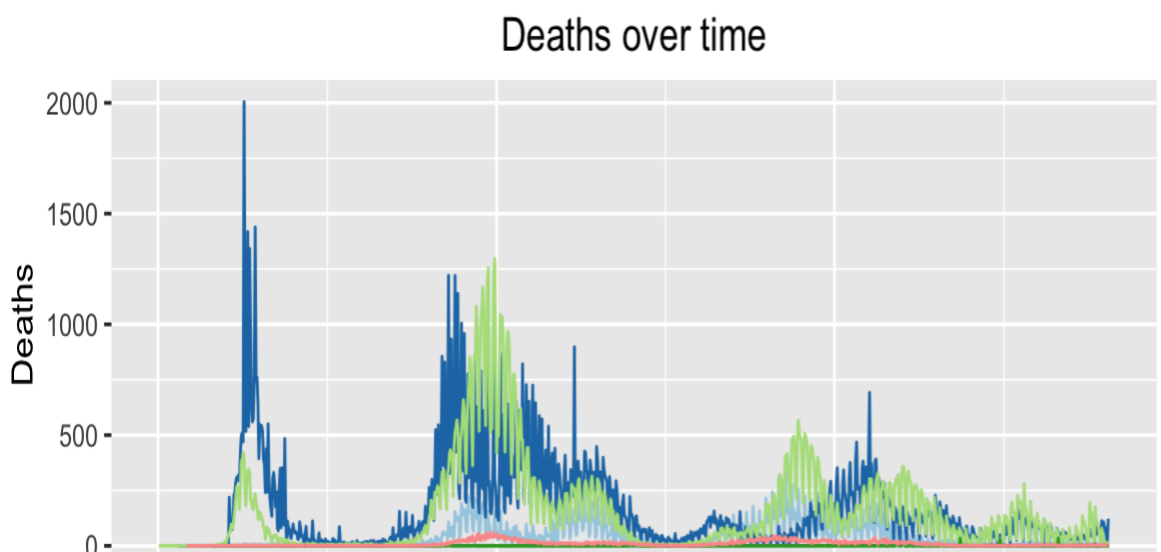
```
## 6                    FRA    67320216
## 7                    FRA    67320216
## 8                    FRA    67320216
## 9                    GRC    10718565
## 10                   ISL      364134
## 11                   ISL      364134
## 12                   IRL     4964440
## 13                   IRL     4964440
## 14                   IRL     4964440
## 15                   IRL     4964440
## 16                   IRL     4964440
## 17                   IRL     4964440
## 18                   IRL     4964440
## 19                   IRL     4964440
## 20                   IRL     4964440
## 21                   IRL     4964440
## 22                   IRL     4964440
## 23                   IRL     4964440
## 24                   IRL     4964440
## 25                   ITA    59641488
## 26                   ITA    59641488
## 27                   LTU     2794090
## 28                   MLT      514564
## 29                   NOR     5367580
## 30                   NOR     5367580
## 31                   NOR     5367580
##Getting a closer look, it does appear these negative values are errors. It is imposs
ible for there to be -2 deaths in a country, for example. Looking further at the data
by expanding the data frame and searching, there also appears to be some data listed a
s N/A so we need to take a closer look at that.


sum(is.na(data)) ## 385

## [1] 385

names(which(sapply(data, anyNA))) ## also only appearing in the data and death columns

## [1] "cases"  "deaths"

#to clean all of this up, we will do the following:
```

```
data <- data %>% replace_na(list(cases = 0, deaths = 0)) %>% mutate(cases = ifelse(cas
es < 0, 0, cases), deaths = ifelse(deaths < 0, 0, deaths))


##incidence rate:


data$incidence_rate = data$cases/data$popData2020*100000


##fatality rate


data$fatality_rate = data$deaths/data$popData2020*100000


##A visualization exploring new cases or incidence rates, per country


EC <- c("Bulgaria", "France", "Germany", "Iceland", "Lithuania" )

newCaseSample <-subset(data, countriesAndTerritories %in% EC)


newCaseSample$dateRep <- as.Date(newCaseSample$dateRep)

newCaseGG <- ggplot(newCaseSample, aes(x = dateRep, y = incidence_rate, group = countr
iesAndTerritories, color = countriesAndTerritories)) +

   geom_line() +

   labs(title = "Incidence Rate Over Time", x = "Date", y = "Incidence Rate") +

  scale_color_brewer(palette="Spectral") +

   theme(legend.position = "right") +

   theme(plot.title = element_text(hjust = 0.5))



   ## A visualization exploring new deaths or fatality rates, per country


   fatalityGG <- ggplot(newCaseSample, aes(x = dateRep, y = fatality_rate, group = cou
ntriesAndTerritories, color = countriesAndTerritories)) +

   geom_line() +

   labs(title = "Fatality Rate Over Time", x = "Date", y = "Fatality Rate") +

  scale_color_brewer(palette="Paired") +

   theme(legend.position = "right") +

   theme(plot.title = element_text(hjust = 0.5))
```

```
   ##  A table or visualization exploring some other aspect of the data: I am going to
look at total deaths over time


   totalcasesGG <- ggplot(newCaseSample, aes(x = dateRep, y = deaths, group = countrie
sAndTerritories, color = countriesAndTerritories)) +
   geom_line() +
   labs(title = "Deaths over time", x = "Date", y = "Deaths") +
  scale_color_brewer(palette="Paired") +
   theme(legend.position = "right") +
   theme(plot.title = element_text(hjust = 0.5))


   grid.arrange(newCaseGG, fatalityGG,totalcasesGG, nrow=3)
```


Incidence Rate Over Time
Incidence Rate
3000
2000
1000
0
2020
2021
2022
Date
countriesAndTerrit
Bulgaria
France
Germany
Iceland
Lithuania


Fatality Rate Over Time
Fatality Rate
7.5
5.0
2.5
0.0
2020
2021
2022
Date
countriesAndTerrit
Bulgaria
France
Germany
Iceland
Lithuania


Deaths over time
Deaths
2000
1500
1000
500
0
countriesAndTerrit
Bulgaria
France
Germany
Iceland
Lithuania

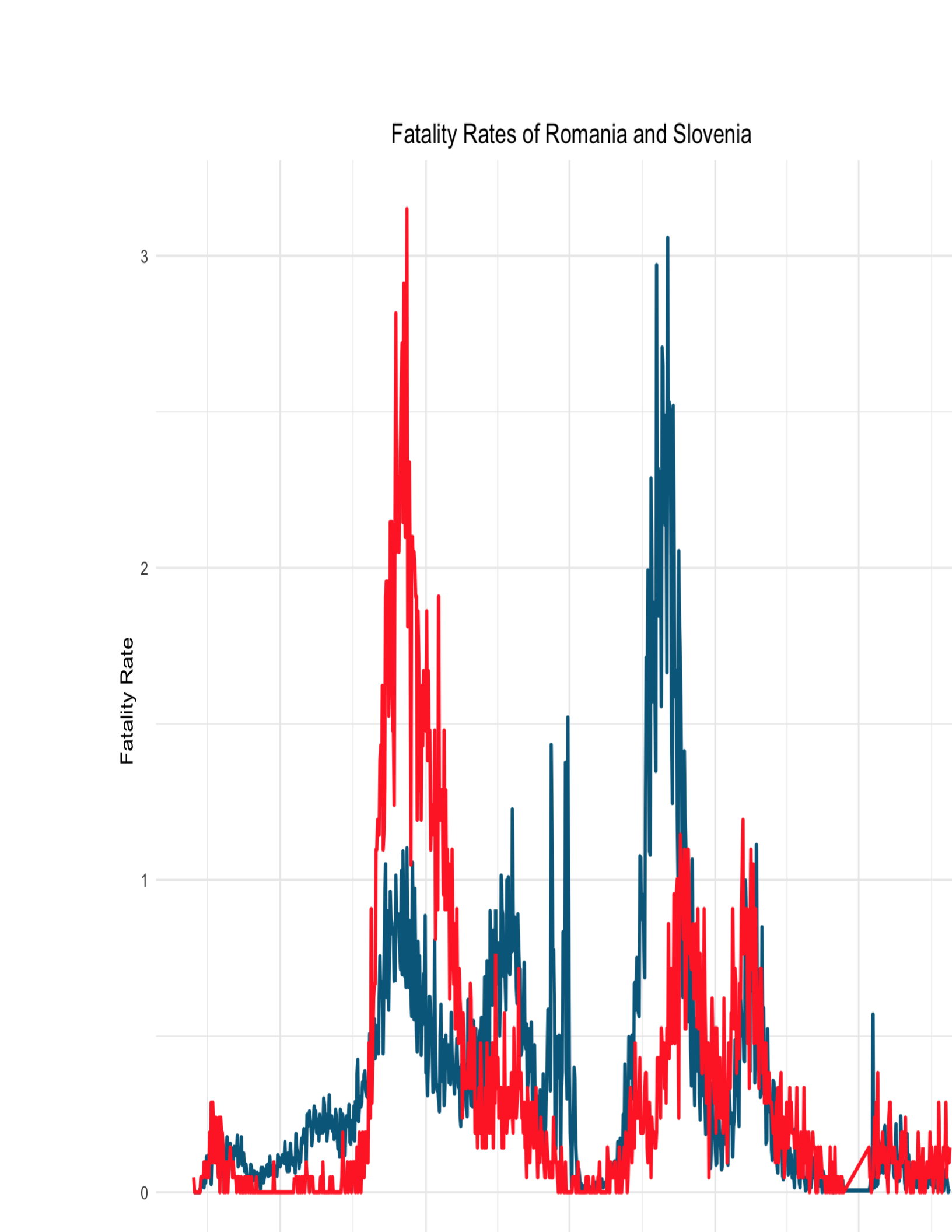
2. Inferential Statistics

Select two (2) countries of your choosing and compare their incidence or fatality rates using hypothesis testing. At minimum, your work should include the following:

- Visualization(s) comparing the daily incidence or fatality rates of the selected countries,
- A statement of the null hypothesis.
- A short justification of the statistical test selected.
    - Why is the test you selected an appropriate one for the comparison we're making?
- A brief discussion of any distributional assumptions of that test.
    - Does the statistical test we selected require assumptions about our data?
    - If so, does our data satisfy those assumptions?
- Your selected alpha.
- The test function output; i.e. the R output.
- The relevant confidence interval, if not returned by the R test output.
- A concluding statement on the outcome of the statistical test.
    - i.e. Based on our selected alpha, do we reject or fail to reject our null hypothesis?

```
Rom <- which(data$countriesAndTerritories=="Romania")

Solv <- which(data$countriesAndTerritories=="Slovenia")

dataRom <- data[Rom,]

dataSolv <- data[Solv,]

merged_data <- merge(dataRom, dataSolv, by.x = "dateRep", by.y = "dateRep", all = FALS
E)

columns_to_keep <- c("dateRep", "fatality_rate.x","fatality_rate.y")


mergedRS <- merged_data[, columns_to_keep, drop = FALSE]

colnames(mergedRS) <- c("Date","fatality_rate_Romania","fatality_rate_Slovenia")



RomSolv <- ggplot(mergedRS, aes(x=Date)) +

  geom_line(aes(y=fatality_rate_Romania, color = "Romania"), size = 1) +

  geom_line(aes(y=fatality_rate_Slovenia, color = "Slovenia"), size = 1) +

  theme_minimal() +

  labs(x = "Date", y = "Fatality Rate", title = "Fatality Rates of Romania and Sloveni
a", color = "Country") +

  scale_color_manual(values = c("Romania" = "deepskyblue4", "Slovenia" = "firebrick1")
) +

  theme(legend.position = "right") +

   theme(plot.title = element_text(hjust = 0.5))
```

```
print(RomSolv)
```


Fatality Rates of Romania and Slovenia
Fatality Rate
3
2
1
0

```
##Null Hypothesis: The median of the paired fatality rate differences equals 1

##Alternative hypothesis: The median of the paired fatality rate differences is not eq
ual to 1


## First we check fo variance:


var.test(mergedRS$fatality_rate_Romania,mergedRS$fatality_rate_Slovenia, data = merged
RS)

##

##  F test to compare two variances

##

## data:  mergedRS$fatality_rate_Romania and mergedRS$fatality_rate_Slovenia

## F = 0.79086, num df = 923, denom df = 923, p-value = 0.000372

## alternative hypothesis: true ratio of variances is not equal to 1

## 95 percent confidence interval:

##  0.6950694 0.8998456

## sample estimates:

## ratio of variances

##          0.7908572

## from the variance test we can see there is an unequal variance between the two grou
ps and are not normally distributed,  however it is still paired data (paired to the d
ate). Because of this, it is good to use the Wilcox text, as it is non-parametric and
looks for comparisons between two populations, but doesn't require normal distribution
s like the student t-test (my first thought to go to here) does. Using this also helps
us avoid a lot of distributional assumptions, but we still have to assume



wilcox.test(mergedRS$fatality_rate_Romania, mergedRS$fatality_rate_Slovenia, paired =
TRUE, conf.int = TRUE, conf.level = 0.95)

##

##  Wilcoxon signed rank test with continuity correction

##

## data:  mergedRS$fatality_rate_Romania and mergedRS$fatality_rate_Slovenia

## V = 238553, p-value = 0.0004606

## alternative hypothesis: true location shift is not equal to 0

## 95 percent confidence interval:

##  0.01351232 0.04545163

## sample estimates:

## (pseudo)median
```

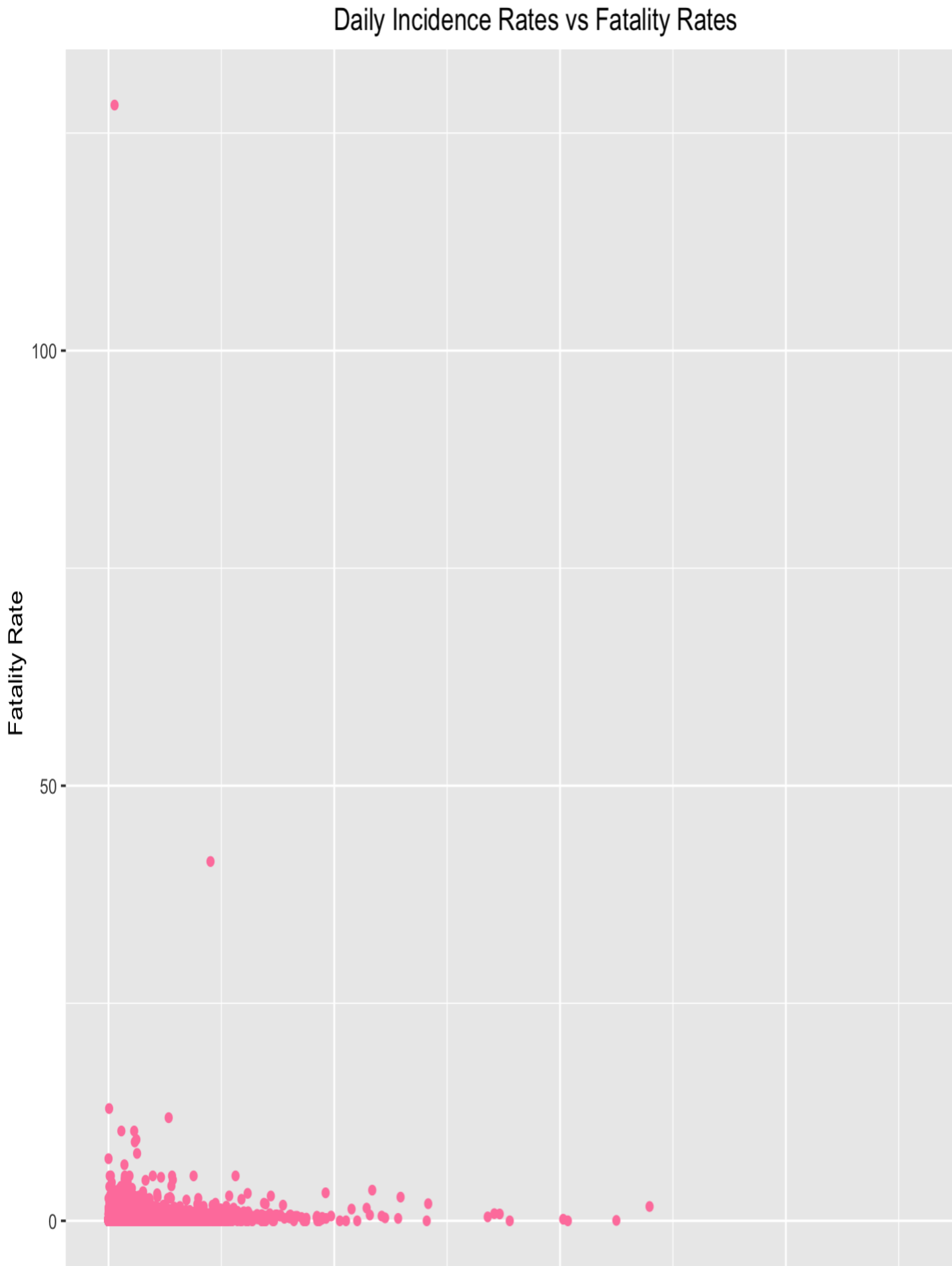
```
##     0.02988073
##the alternative hypothesis, not the null hypothesis, is correct.
```

3. Correlation

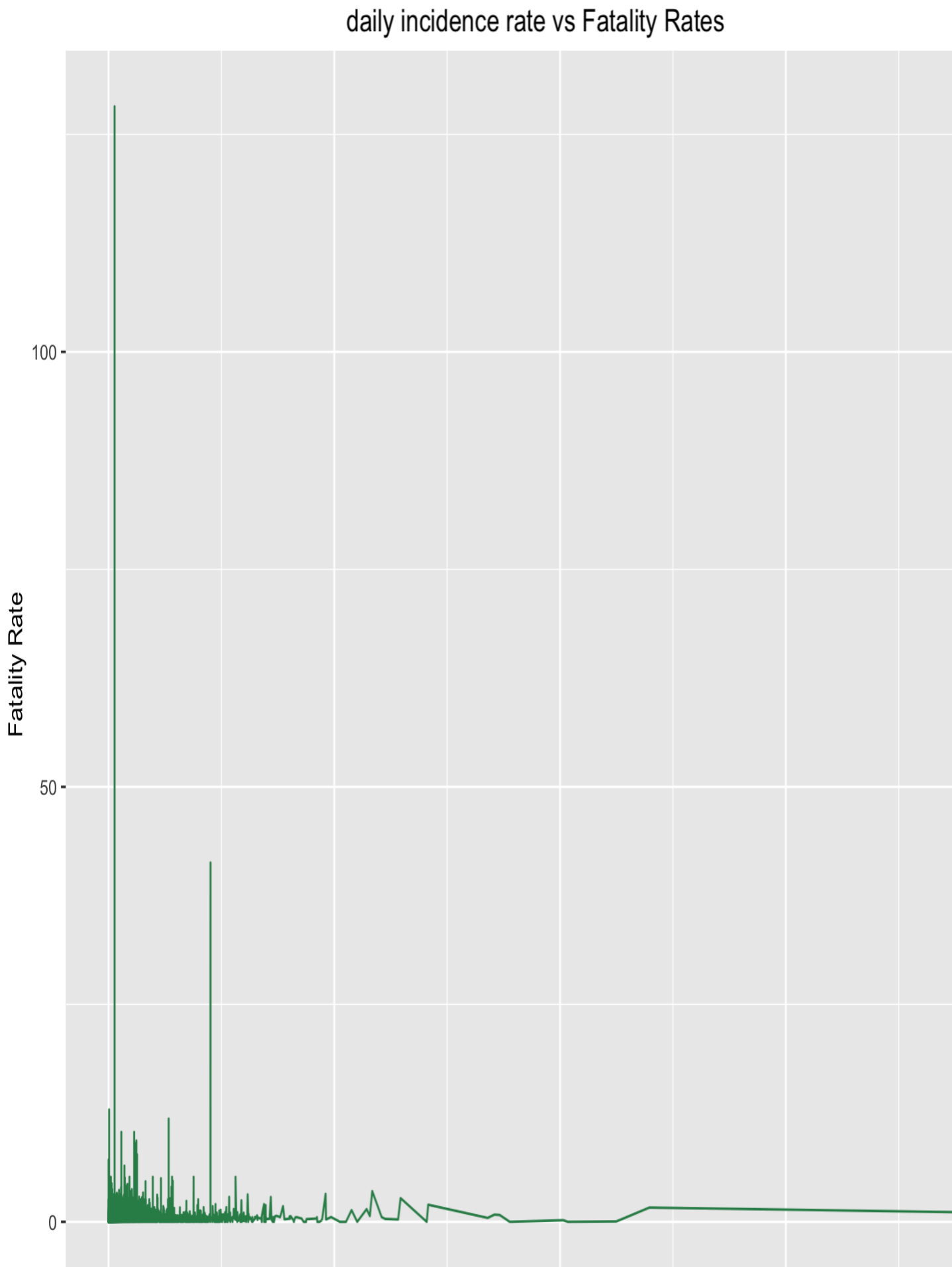
Considering all countries, explore the relationship between incidence rates and fatality rates. At minimum, your work should include the following:

- Visualization(s) showing the distributions of daily incidence and fatality rates, regardless of country. Please note that both country and date should be disregarded here.
- A short statement identifying the most appropriate correlation coefficient.
    - For the correlation we're interested in, which correlation coefficient is most appropriate?
    - Why do you find the correlation coefficient selected to be the most appropriate?
- The calculated correlation coefficient or coefficient test output; e.g. *cor()* or *cor.test()*.

```
ggplot(data, aes(x = incidence_rate, y = fatality_rate)) +
  geom_point(color = "palevioletred1") +
  labs(title = "Daily Incidence Rates vs Fatality Rates", x = "Incidence Rate", y = "F
atality Rate") +
  theme(plot.title = element_text(hjust = 0.5))
```


Daily Incidence Rates vs Fatality Rates
Fatality Rate
100
50
0

```
ggplot(data, aes(x = incidence_rate, y = fatality_rate)) +
  geom_line(color = "seagreen") +
  labs(title = "daily incidence rate vs Fatality Rates", x = "Cases", y = "Fatality Ra
te") +
  theme(plot.title = element_text(hjust = 0.5))
```


daily incidence rate vs Fatality Rates
Fatality Rate
100
50
0

```
IFpearson <- cor(data$incidence_rate, data$fatality_rate, method = "pearson")

IFspearman<- cor(data$incidence_rate, data$fatality_rate, method = "spearman")

IFkendall <- cor(data$incidence_rate, data$fatality_rate, method = "kendall")

cat("Pearson Correlation:", IFpearson, "\n") # 0.1097366

## Pearson Correlation: 0.1097366

cat("Spearman Correlation:", IFspearman, "\n") # 0.5694821

## Spearman Correlation: 0.5694821

cat("Kendall Tau Correlation:", IFkendall, "\n") # 0.4135004

## Kendall Tau Correlation: 0.4135004

## Pearson, Spearman, and Kendall Tau are our three most common correlation tests, and
with good reason. Pearson helps with two variables (in this cae, incidence and fatalit
y), and helps us with try to figure out if normal distribution or linear relationship
exist between the two variables (it does not here.) This is somewhat logical, as we al
l know spikes in covid were not linear, but confirms this,.


## For Spearman, we're still looking at two variables but it abandon the need for norm
al distribution or linear relationship, Because of the result we got with the pearson
test we can now use this one. We get a stronger result here than on the Pearson one, b
ut still relatively tame. One would assume that more cases = more death, and this does
n't disprove that, but shows that while that may be true there is also other stuff goi
ng on


##For Kendall, we are expanding on Spearman, or rather finding a correlaton coefficien
t that has both smaller  error sensitivity and a  smaller asymptotic variance. As one
may predict, it's number is more similar to Speaman than it is Pearson but is more con
servative than Spearman's number on the correlation coefficient.
```

4. Regression

Here, we will fit a model on data from twenty (20) countries considering total new cases as a function of population, population density and gross domestic product (GDP) per capita. Note that the GDP per capita is given in “purchasing power standard,” which considers the costs of goods and services in a country relative to incomes in that country; i.e. we will consider this as appropriately standardized.

Code is given below defining a new data frame, ‘model_df,’ which provides the total area and standardized GDP per capita for the twenty (20) countries for our model fit. You are responsible for creating a vector of the total new cases across the time frame of the dataset, for each of those countries, and adding that vector to our ’model_df” data frame.

```
# The code below creates a new data frame, 'model_df,' that includes the area,

# GDP per capita, population and population density for the twenty (20)

# countries of interest. All you should need to do is execute this code, as is.


# You do not need to add code in this chunk. You will need to add code in the

# 'regression_b,' 'regression_c' and 'regression_d' code chunks.
```

```
twenty_countries <- c("Austria", "Belgium", "Bulgaria", "Cyprus", "Denmark",
                      "Finland", "France", "Germany", "Hungary", "Ireland",
                      "Latvia", "Lithuania", "Malta", "Norway", "Poland",
                      "Portugal", "Romania", "Slovakia", "Spain", "Sweden")

sq_km <- c(83858, 30510, 110994, 9251, 44493, 338145, 551695, 357386, 93030,
           70273, 64589, 65300, 316, 385178, 312685, 88416, 238397, 49036,
           498511, 450295)

gdp_pps <- c(128, 118, 51, 91, 129, 111, 104, 123, 71, 190, 69, 81, 100, 142,
             71, 78, 65, 71, 91, 120)

model_df <- data %>%
  select(c(countriesAndTerritories, popData2020)) %>%
  filter(countriesAndTerritories %in% twenty_countries) %>%
  distinct(countriesAndTerritories, .keep_all = TRUE) %>%
  add_column(sq_km, gdp_pps) %>%
  mutate(pop_dens = popData2020 / sq_km) %>%
  rename(country = countriesAndTerritories, pop = popData2020)
```

Next, we need to add one (1) more column to our 'model_df' data frame. Specifically, one that has the total number of new cases for each of the twenty (20) countries. We calculate the total number of new cases by summing all the daily new cases, for each country, across all the days in the dataset.

```
### The following code will be removed for students to complete the work themselves.

total_cases <- data %>%
  select(c(countriesAndTerritories, cases)) %>%
  group_by(countriesAndTerritories) %>%
  dplyr::summarize(total_cases = sum(cases, na.rm = TRUE)) %>%
  filter(countriesAndTerritories %in% twenty_countries) %>%
  select(total_cases)

model_df <- model_df %>%
  add_column(total_cases)
```

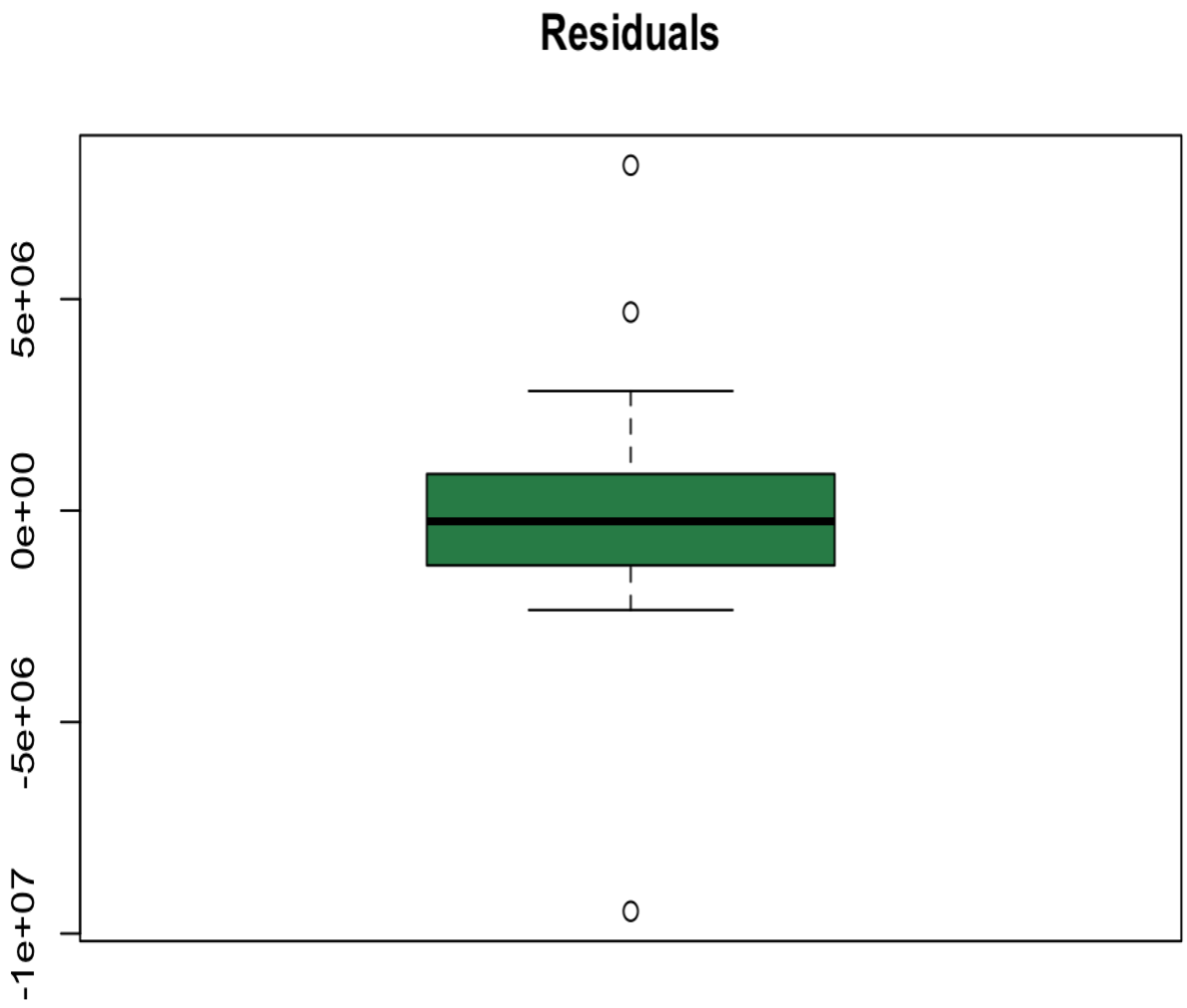
Now, we will fit our model using the data in 'model_df.' We are interested in explaining total cases (response) as a function of population (explanatory), population density (explanatory), and GDP (explanatory).

At minimum, your modeling work should including the following:

- A description - either narrative or using R output - of your 'model_df' data frame.
    - Consider: what data types are present? What do our rows and columns represent?
- The *lm() summary()* output of your fitted model. As we did in the second Data Analysis Assignment, you can pass your fitted model object - i.e. the output of **lm()** - to *summary()* and get additional details, including R^2, on your model fit.
- A short statement on the fit of the model.
    - Which, if any, of our coefficients are statistically significant?
    - What is the R^2 of our model?
    - Should we consider a reduced model; i.e. one with fewer parameters?

```
summary(model_df)
##      country        pop               sq_km           gdp_pps
##  Austria : 1   Min.   :  514564   Min.   :   316   Min.   : 51.0
##  Belgium : 1   1st Qu.: 5266795   1st Qu.: 60701   1st Qu.: 71.0
##  Bulgaria: 1   Median : 7926273   Median : 90723   Median : 95.5
##  Cyprus  : 1   Mean   :17305840   Mean   :192118   Mean   :100.2
##  Denmark : 1   3rd Qu.:13474040   3rd Qu.:342955   3rd Qu.:120.8
##  Finland : 1   Max.   :83166711   Max.   :551695   Max.   :190.0
##  (Other) :14
##     pop_dens        total_cases
##  Min.   :  13.94   Min.   :  115285
##  1st Qu.:  57.67   1st Qu.: 1320359
##  Median : 100.50   Median : 2570210
##  Mean   : 179.14   Mean   : 6498053
##  3rd Qu.: 121.55   3rd Qu.: 5430242
##  Max.   :1628.37   Max.   :36962242
##
modelone = lm(total_cases~.,model_df[,-1])
summary(modelone)
##
## Call:
## lm(formula = total_cases ~ ., data = model_df[, -1])
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max 
## -8164687  -770205   253808  1293224  9473515 
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -3.491e+06  2.868e+06  -1.217    0.242    
## pop          4.557e-01  5.129e-02   8.886 2.31e-07 ***
## sq_km       -5.340e+00  6.917e+00  -0.772    0.452    
## gdp_pps      3.121e+04  2.619e+04   1.192    0.252    
## pop_dens     9.569e+00  2.600e+03   0.004    0.997    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3755000 on 15 degrees of freedom
## Multiple R-squared:  0.9002, Adjusted R-squared:  0.8736 
## F-statistic: 33.83 on 4 and 15 DF,  p-value: 2.411e-07
pred = predict(modelone,model_df)
res = pred - model_df$total_cases
boxplot(res,main = "Residuals", col = 'seagreen')
```




```
##The only thing that get flagged as statistically significant is population. The R^2
of the model is 0.8736 so that tells us a lot of variation  can be explained by functi
on of population,  density, and GDP. If we are going to reduce parameters, these resul
ts also tell us that the sq_km should be the one to go with it's negative coefficient
results and possibly redundency.
```

The last thing we will do is use our model to predict the total new cases of two (2) countries not included in our model fit. At minimum, your work should include:

- The predicted total new cases for both countries.
- The actual total new cases for both countries.
- A short statement on the performance of the model in these two (2) cases.

- Compare the new predictions to those made on the fitted dataset. You may compare the predicted values or the residuals.

```
# The code below defines our 'newdata' data frame for applying our model to the
# population, population density and GDP per capita for two (2). Please execute
# the code as given.

newdata <- data.frame(country = c("Luxembourg", "Netherlands"),
                      pop = c(626108, 17407585),
                      gdp_pps = c(261, 130),
                      pop_dens = c(626108, 17407585) / c(2586, 41540))

# Add code here returning the actual  total cases from our dataset for the
# Netherlands and Luxembourg.

actualtotal <- data %>%
  select(c(countriesAndTerritories, cases)) %>%
  group_by(countriesAndTerritories) %>%
  dplyr::summarize(total_cases = sum(cases, na.rm = TRUE)) %>%
  filter(countriesAndTerritories %in% c("Luxembourg", "Netherlands")) %>%
  select(total_cases)



# Add code here returning the total cases for the Netherlands and Luxembourg
# predicted by our model.

## we get an error message about sq_km when trying to run prediction, so we have to fi
x our model

modeltwo = lm(total_cases~pop+gdp_pps+pop_dens,model_df[,-1])

prediction = predict(modeltwo,newdata)

cbind(actualtotal,prediction)
##   total_cases prediction
## 1      301031    3954252
```

```
## 2     8494705    7543125
##prediction and residual are about 10k off from each other in both cases.
```