Take Home Final Exam - Steve Desilets

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 (https://www.ecdc.europa.eu/en/publications-data/data-daily-new-cases-covid-19-eueea-country), *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 (https://www.ecdc.europa.eu/sites/default/files/documents/Description-and-disclaimer_daily_reporting.pdf).

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).

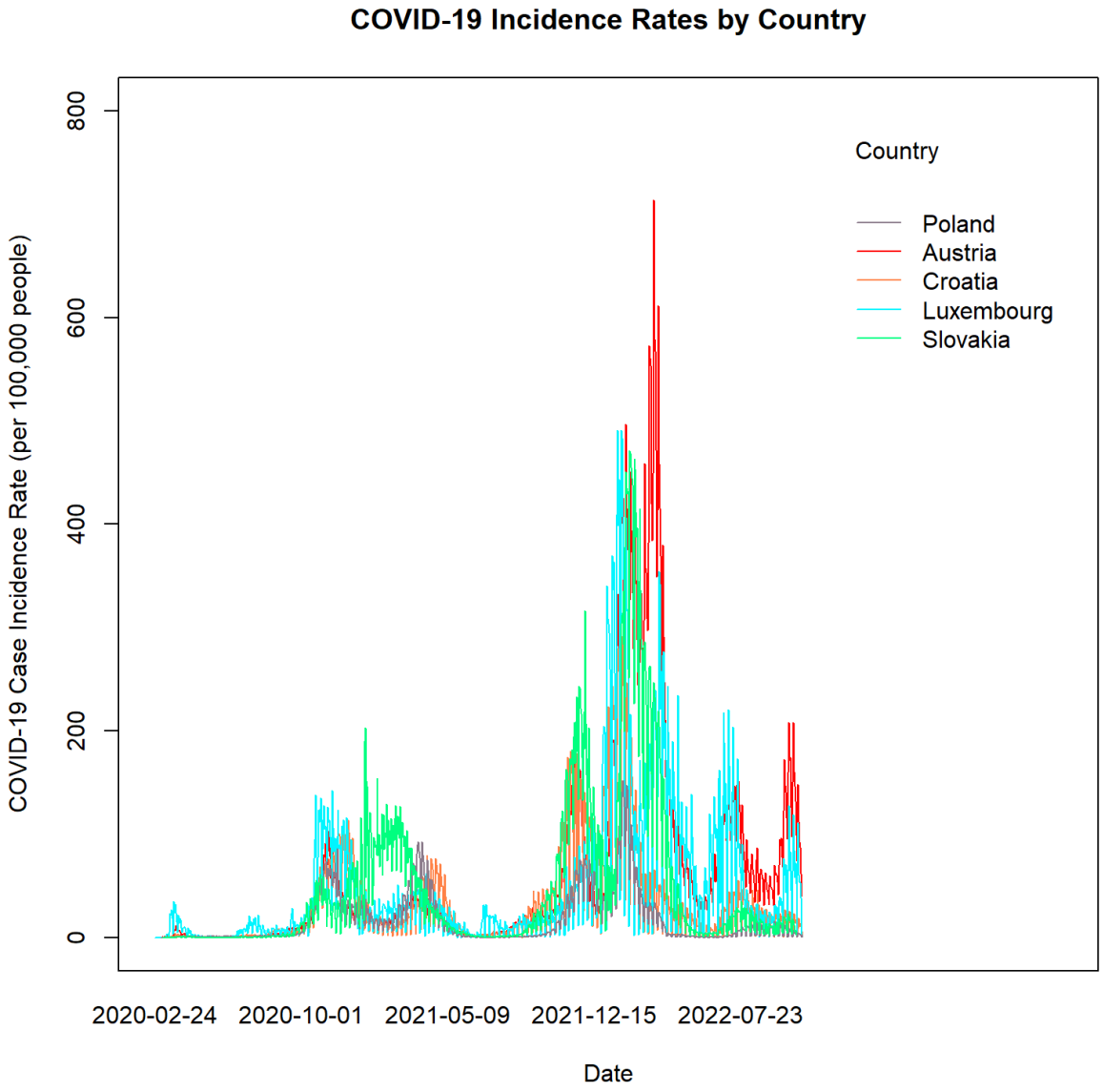
---

1. Descriptive Statistics

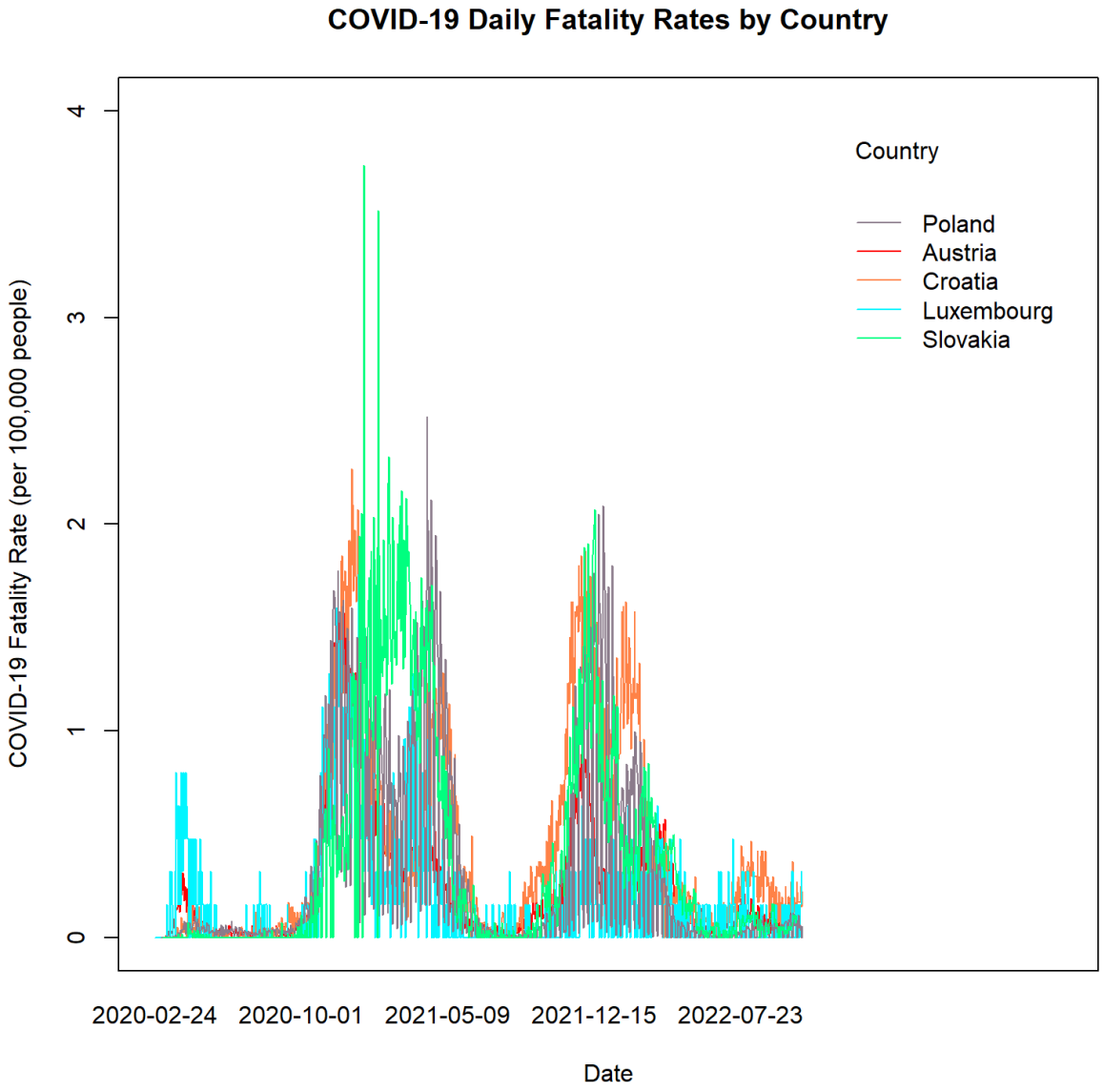
Perform an Exploratory Data Analysis (EDA). Your EDA is exactly that: yours. Your knit .html should include the visualizations and summary tables that you find valuable while exploring this dataset. **However**, at minimum, your EDA must include the following:

- 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.

```
#Create a vector for incidence_rate to be added to the data frame

incidence_rate <- (data$cases / data$popData2020) * 100000

data <- cbind(data, incidence_rate)

#Create a vector for fatality rate to be added to the data frame

fatality_rate <- (data$deaths / data$popData2020) * 100000

data <- cbind(data, fatality_rate)


#Create a visualization exploring new cases or incidence rates per country over time (with at least 5 countries)

five_countries_data <- data[data$countriesAndTerritories == "Austria" | data$countriesAndTerritories == "Croatia" | data$co
untriesAndTerritories == "Luxembourg" | data$countriesAndTerritories == "Poland" | data$countriesAndTerritories == "Slovaki
a", ]


five_countries_data$countriesAndTerritories <- droplevels(five_countries_data$countriesAndTerritories)


interaction.plot(x.factor = five_countries_data$dateRep, trace.factor = five_countries_data$countriesAndTerritories, respon
se = five_countries_data$incidence_rate, fun = mean, xlab = "Date", ylab = "COVID-19 Case Incidence Rate (per 100,000 peopl
e)", col = c("red", "sienna1", "turquoise1", "thistle4", "springgreen"), main = "COVID-19 Incidence Rates by Country", lty
= 7, lwd = 1, trace.label = "Country", ylim = c(0,800))
```




```
#Create a visualization exploring new deaths or fatality rates per country over time (with at least 5 countries)

interaction.plot(x.factor = five_countries_data$dateRep, trace.factor = five_countries_data$countriesAndTerritories, respon
se = five_countries_data$fatality_rate, fun = mean, xlab = "Date", ylab = "COVID-19 Fatality Rate (per 100,000 people)", co
l = c("red", "sienna1", "turquoise1", "thistle4", "springgreen"), main = "COVID-19 Daily Fatality Rates by Country", lty =
7, lwd = 1, trace.label = "Country", ylim = c(0,4))
```


COVID-19 Daily Fatality Rates by Country
COVID-19 Fatality Rate (per 100,000 people)
0
1
2
3
4
2020-02-24
2020-10-01
2021-05-09
2021-12-15
2022-07-23
Date
Country
Poland
Austria
Croatia
Luxembourg
Slovakia

```
# Create a table exploring some other aspect of the data (using data from across the entire timespan of the dataset)

# First I'm gatherting data regarding the minimum, mean, and maximum values for the incidence rates and fatality rates for
each country

Min_incidence_rate <- aggregate(incidence_rate ~ countriesAndTerritories, data = data, FUN = min)

Mean_incidence_rate <- aggregate(incidence_rate ~ countriesAndTerritories, data = data, FUN = mean)

Max_incidence_rate <- aggregate(incidence_rate ~ countriesAndTerritories, data = data, FUN = max)

Min_fatality_rate <- aggregate(fatality_rate ~ countriesAndTerritories, data = data, FUN = min)

Mean_fatality_rate <- aggregate(fatality_rate ~ countriesAndTerritories, data = data, FUN = mean)

Max_fatality_rate <- aggregate(fatality_rate ~ countriesAndTerritories, data = data, FUN = max)



#Next I'm creating vectors of the skewness and kurtosis of the incidence rates and fatality rates for each country

Austria_data <- data[data$countriesAndTerritories == "Austria", ]
Belgium_data <- data[data$countriesAndTerritories == "Belgium", ]
Bulgaria_data <- data[data$countriesAndTerritories == "Bulgaria", ]
Croatia_data <- data[data$countriesAndTerritories == "Croatia", ]
Cyprus_data <- data[data$countriesAndTerritories == "Cyprus", ]
Czechia_data <- data[data$countriesAndTerritories == "Czechia", ]
Denmark_data <- data[data$countriesAndTerritories == "Denmark", ]
Estonia_data <- data[data$countriesAndTerritories == "Estonia", ]
Finland_data <- data[data$countriesAndTerritories == "Finland", ]
France_data <- data[data$countriesAndTerritories == "France", ]
Germany_data <- data[data$countriesAndTerritories == "Germany", ]
Greece_data <- data[data$countriesAndTerritories == "Greece", ]
Hungary_data <- data[data$countriesAndTerritories == "Hungary", ]
Iceland_data <- data[data$countriesAndTerritories == "Iceland", ]
Ireland_data <- data[data$countriesAndTerritories == "Ireland", ]
Italy_data <- data[data$countriesAndTerritories == "Italy", ]
Latvia_data <- data[data$countriesAndTerritories == "Latvia", ]
Liechtenstein_data <- data[data$countriesAndTerritories == "Liechtenstein", ]
Lithuania_data <- data[data$countriesAndTerritories == "Lithuania", ]
Luxembourg_data <- data[data$countriesAndTerritories == "Luxembourg", ]
Malta_data <- data[data$countriesAndTerritories == "Malta", ]
Netherlands_data <- data[data$countriesAndTerritories == "Netherlands", ]
Norway_data <- data[data$countriesAndTerritories == "Norway", ]
Poland_data <- data[data$countriesAndTerritories == "Poland", ]
Portugal_data <- data[data$countriesAndTerritories == "Portugal", ]
Romania_data <- data[data$countriesAndTerritories == "Romania", ]
Slovakia_data <- data[data$countriesAndTerritories == "Slovakia", ]
Slovenia_data <- data[data$countriesAndTerritories == "Slovenia", ]
Spain_data <- data[data$countriesAndTerritories == "Spain", ]
Sweden_data <- data[data$countriesAndTerritories == "Sweden", ]

incidence_rate_skewness_vector <- c(skewness(Austria_data$incidence_rate, na.rm = TRUE),
  skewness(Belgium_data$incidence_rate, na.rm = TRUE),
  skewness(Bulgaria_data$incidence_rate, na.rm = TRUE),
  skewness(Croatia_data$incidence_rate, na.rm = TRUE),
  skewness(Cyprus_data$incidence_rate, na.rm = TRUE),
  skewness(Czechia_data$incidence_rate, na.rm = TRUE),
  skewness(Denmark_data$incidence_rate, na.rm = TRUE),
  skewness(Estonia_data$incidence_rate, na.rm = TRUE),
  skewness(Finland_data$incidence_rate, na.rm = TRUE),
  skewness(France_data$incidence_rate, na.rm = TRUE),
  skewness(Germany_data$incidence_rate, na.rm = TRUE),
  skewness(Greece_data$incidence_rate, na.rm = TRUE),
  skewness(Hungary_data$incidence_rate, na.rm = TRUE),
  skewness(Iceland_data$incidence_rate, na.rm = TRUE),
```

```
  skewness(Ireland_data$incidence_rate, na.rm = TRUE),
  skewness(Italy_data$incidence_rate, na.rm = TRUE),
  skewness(Latvia_data$incidence_rate, na.rm = TRUE),
  skewness(Liechtenstein_data$incidence_rate, na.rm = TRUE),
  skewness(Lithuania_data$incidence_rate, na.rm = TRUE),
  skewness(Luxembourg_data$incidence_rate, na.rm = TRUE),
  skewness(Malta_data$incidence_rate, na.rm = TRUE),
  skewness(Netherlands_data$incidence_rate, na.rm = TRUE),
  skewness(Norway_data$incidence_rate, na.rm = TRUE),
  skewness(Poland_data$incidence_rate, na.rm = TRUE),
  skewness(Portugal_data$incidence_rate, na.rm = TRUE),
  skewness(Romania_data$incidence_rate, na.rm = TRUE),
  skewness(Slovakia_data$incidence_rate, na.rm = TRUE),
  skewness(Slovenia_data$incidence_rate, na.rm = TRUE),
  skewness(Spain_data$incidence_rate, na.rm = TRUE),
  skewness(Sweden_data$incidence_rate, na.rm = TRUE))


fatality_rate_skewness_vector <- c(skewness(Austria_data$fatality_rate, na.rm = TRUE),
  skewness(Belgium_data$fatality_rate, na.rm = TRUE),
  skewness(Bulgaria_data$fatality_rate, na.rm = TRUE),
  skewness(Croatia_data$fatality_rate, na.rm = TRUE),
  skewness(Cyprus_data$fatality_rate, na.rm = TRUE),
  skewness(Czechia_data$fatality_rate, na.rm = TRUE),
  skewness(Denmark_data$fatality_rate, na.rm = TRUE),
  skewness(Estonia_data$fatality_rate, na.rm = TRUE),
  skewness(Finland_data$fatality_rate, na.rm = TRUE),
  skewness(France_data$fatality_rate, na.rm = TRUE),
  skewness(Germany_data$fatality_rate, na.rm = TRUE),
  skewness(Greece_data$fatality_rate, na.rm = TRUE),
  skewness(Hungary_data$fatality_rate, na.rm = TRUE),
  skewness(Iceland_data$fatality_rate, na.rm = TRUE),
  skewness(Ireland_data$fatality_rate, na.rm = TRUE),
  skewness(Italy_data$fatality_rate, na.rm = TRUE),
  skewness(Latvia_data$fatality_rate, na.rm = TRUE),
  skewness(Liechtenstein_data$fatality_rate, na.rm = TRUE),
  skewness(Lithuania_data$fatality_rate, na.rm = TRUE),
  skewness(Luxembourg_data$fatality_rate, na.rm = TRUE),
  skewness(Malta_data$fatality_rate, na.rm = TRUE),
  skewness(Netherlands_data$fatality_rate, na.rm = TRUE),
  skewness(Norway_data$fatality_rate, na.rm = TRUE),
  skewness(Poland_data$fatality_rate, na.rm = TRUE),
  skewness(Portugal_data$fatality_rate, na.rm = TRUE),
  skewness(Romania_data$fatality_rate, na.rm = TRUE),
  skewness(Slovakia_data$fatality_rate, na.rm = TRUE),
  skewness(Slovenia_data$fatality_rate, na.rm = TRUE),
  skewness(Spain_data$fatality_rate, na.rm = TRUE),
  skewness(Sweden_data$fatality_rate, na.rm = TRUE))



incidence_rate_kurtosis_vector <- c(kurtosis(Austria_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Belgium_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Bulgaria_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Croatia_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Cyprus_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Czechia_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Denmark_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Estonia_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Finland_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(France_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Germany_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Greece_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Hungary_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Iceland_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Ireland_data$incidence_rate , excess = TRUE, na.rm = TRUE),
```

```
  kurtosis(Italy_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Latvia_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Liechtenstein_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Lithuania_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Luxembourg_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Malta_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Netherlands_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Norway_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Poland_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Portugal_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Romania_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Slovakia_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Slovenia_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Spain_data$incidence_rate , excess = TRUE, na.rm = TRUE),
  kurtosis(Sweden_data$incidence_rate , excess = TRUE, na.rm = TRUE))


fatality_rate_kurtosis_vector <- c(kurtosis(Austria_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Belgium_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Bulgaria_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Croatia_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Cyprus_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Czechia_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Denmark_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Estonia_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Finland_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(France_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Germany_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Greece_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Hungary_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Iceland_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Ireland_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Italy_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Latvia_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Liechtenstein_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Lithuania_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Luxembourg_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Malta_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Netherlands_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Norway_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Poland_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Portugal_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Romania_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Slovakia_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Slovenia_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Spain_data$fatality_rate , excess = TRUE, na.rm = TRUE),
kurtosis(Sweden_data$fatality_rate , excess = TRUE, na.rm = TRUE))

# Last, I'm consolidating these data into one output table


Table_Q1 <- cbind(Min_incidence_rate,
                  Mean_incidence_rate,
                  Max_incidence_rate,
                  incidence_rate_skewness_vector,
                  incidence_rate_kurtosis_vector,
                  Min_fatality_rate,
                  Mean_fatality_rate,
                  Max_fatality_rate,
                  fatality_rate_skewness_vector,
                  fatality_rate_kurtosis_vector)

Table_Q1 <- Table_Q1[ -c(3, 5, 9, 11, 13) ]

knitr::kable(Table_Q1, caption = "COVID-19 Case Incidence and Fatality Rate Summary Data by Country", col.names = c("Countr
y / Territory", "Minimum Incidence Rate", "Mean Incidence Rate", "Maximum Incidence Rate", "Incidence Rate Skewness", "Inci
```

```
dence Rate Kurtosis", "Minimum Fatality Rate", "Mean Fatality Rate", "Maximum Fatality Rate", "Fatality Rate Skewness", "Fa
tality Rate Kurtosis"))
```

COVID-19 Case Incidence and Fatality Rate Summary Data by Country

Country / Territory	Minimum Incidence Rate	Mean Incidence Rate	Maximum Incidence Rate	Incidence Rate Skewness	Incidence Rate Kurtosis	Minimum Fatality Rate	Mean Fatality Rate	Maximum Fatality Rate	Fatality Rate Skewness	Fatality Rate Kurtosis
Austria	0.0112346	62.50380	713.4990	2.804071	8.618152	0.0000000	0.2418045	1.6290187	2.142925	4.8251229
Belgium	0.1128233	41.30725	659.8776	4.333468	24.783834	0.0086787	0.2997713	2.8119044	2.859592	8.7768535
Bulgaria	0.0000000	19.07311	178.3648	2.542765	8.045589	0.0000000	0.5698376	4.8047308	1.790024	3.4010700
Croatia	0.0000000	31.55484	291.0675	2.598088	7.593191	0.0000000	0.4331307	2.2670345	1.268016	0.6115412
Cyprus	-360.6961673	84.57202	1732.6479	4.301553	28.430240	0.0000000	0.1689180	3.1531354	4.484163	34.6891362
Czechia	0.0000000	38.71890	535.3219	3.105840	14.220167	0.0000000	0.3875197	2.4406348	1.515588	1.1686274
Denmark	-34.3651287	60.56171	946.6296	3.792474	14.476286	0.0000000	0.1341821	0.8586989	1.835955	3.0939097
Estonia	0.0000000	46.93332	637.1071	3.743835	15.385145	0.0000000	0.2167177	1.5801640	1.562298	2.2893999
Finland	0.0000000	23.60095	200.6410	2.449324	5.407304	0.0000000	0.1132399	1.0316197	2.096562	5.3241487
France	-518.1890682	54.06142	745.1476	3.312371	15.637163	-0.3223400	0.2310978	2.9768175	3.041058	14.2396022
Germany	0.0012024	42.77224	370.2371	2.369289	5.570674	0.0000000	0.1851521	1.5583158	2.443626	7.2890440
Greece	-26.6733467	49.34027	2013.8517	8.121405	94.274806	0.0000000	0.3225873	128.2167902	30.692049	948.5014928
Hungary	0.0000000	22.64498	461.0971	3.876996	21.816718	0.0000000	0.5069102	11.8531851	3.951647	36.6623060
Iceland	-265.8362032	72.48320	3785.4197	7.253294	74.358200	-0.2746242	0.0772721	9.3372220	13.584913	202.2123705
Ireland	-111.4325080	34.97592	526.1822	4.219865	21.987739	-0.1007163	0.1677209	4.4113737	4.011560	28.5483277
Italy	-0.2481494	40.21235	382.4905	2.817053	9.439408	-0.0519772	0.3075065	1.6649484	1.434883	1.4413394
Latvia	0.0000000	51.29859	628.6186	3.770176	14.848069	0.0000000	0.3266242	4.1411666	2.590446	10.0360492
Liechtenstein	0.0000000	55.94009	934.2659	3.463006	15.099481	0.0000000	0.2838269	41.2935195	16.229866	343.3415239
Lithuania	-1.4673829	45.44572	551.5928	3.492324	14.850199	0.0000000	0.3369341	2.2189693	1.435019	1.7200552
Luxembourg	0.0000000	49.41390	490.8099	2.859602	9.960028	0.0000000	0.1877863	1.7568854	2.109810	4.7328602
Malta	-8.1622500	23.23262	272.6580	3.257385	13.800278	0.0000000	0.1628902	1.3603750	1.642578	2.3725272
Netherlands	0.0057446	50.46419	2249.3183	9.492834	160.667200	0.0000000	0.1352749	1.3327524	2.272674	5.9049780
Norway	0.0000000	27.85975	549.2047	4.205214	18.819283	-0.0186304	0.0815956	6.4461079	9.633586	126.1027695
Poland	0.0000000	16.88021	151.9016	2.173977	5.555605	0.0000000	0.3219467	2.5185640	1.655272	1.7906897
Portugal	0.0000000	57.34468	676.1715	3.210740	11.853241	0.0000000	0.2613354	2.8846409	3.723402	15.9817398
Romania	0.0879515	18.17810	207.0378	3.192095	13.432833	0.0000000	0.3773707	3.0576075	2.499214	7.7280319
Slovakia	0.0000000	48.69687	470.7328	2.728149	8.379565	0.0000000	0.3939264	3.7377198	1.731778	3.0114167
Slovenia	0.0000000	60.72749	1157.4241	4.720083	30.019603	0.0000000	0.3420516	3.1490638	2.360565	5.8471637
Spain	0.0000000	32.56649	349.4335	3.299927	11.949155	0.0000000	0.2589486	1.9289026	2.459454	7.0989574
Sweden	0.0000000	25.68473	521.7191	4.899373	27.880328	0.0000000	0.2012189	1.1716190	1.681870	2.1028864

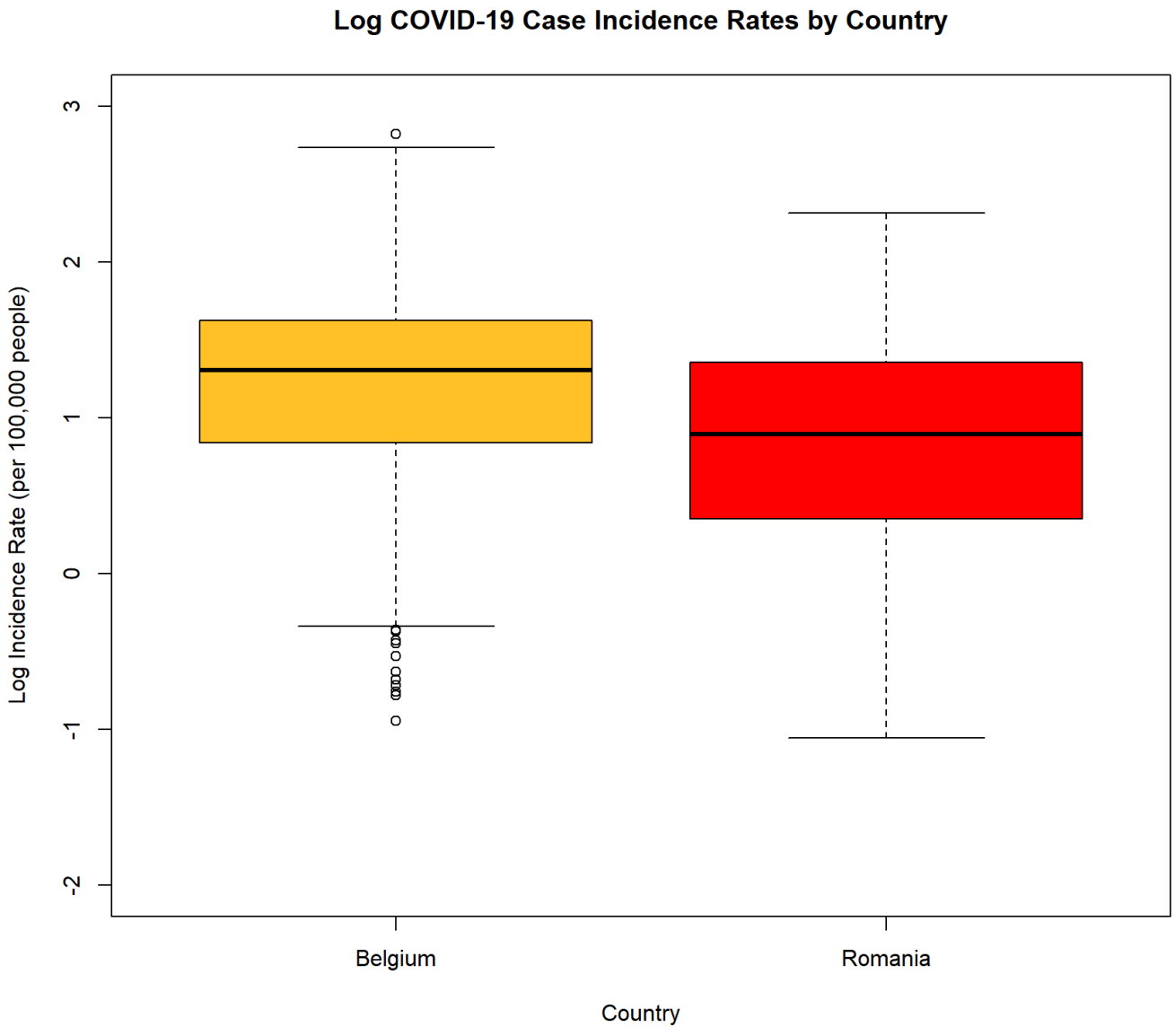
2. Inferential Statistics

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?

```
# Create a visualization comparing the daily incidence or fatality rates of the selected countries

data$log_incidence_rate <- log10(data$incidence_rate)

two_countries_data <- data[data$countriesAndTerritories == "Belgium" | data$countriesAndTerritories == "Romania" , ]

two_countries_data$countriesAndTerritories <- droplevels(two_countries_data$countriesAndTerritories)

boxplot(two_countries_data$log_incidence_rate ~ two_countries_data$countriesAndTerritories, ylab = "Log Incidence Rate (per 100,000 people)", main = "Log COVID-19 Case Incidence Rates by Country", xlab = "Country", col = c("goldenrod1", "red"), ylim = c(-2, 3))
```



```
# State the null hypothesis
print("The null hypothesis is that the log of the average daily fatality rate from COVID-19 in Belgium is equal to the log of the average daily fatality rate from COVID-19 in Romania during the sample time frame.")
```

```
## [1] "The null hypothesis is that the log of the average daily fatality rate from COVID-19 in Belgium is equal to the log of the average daily fatality rate from COVID-19 in Romania during the sample time frame."
```

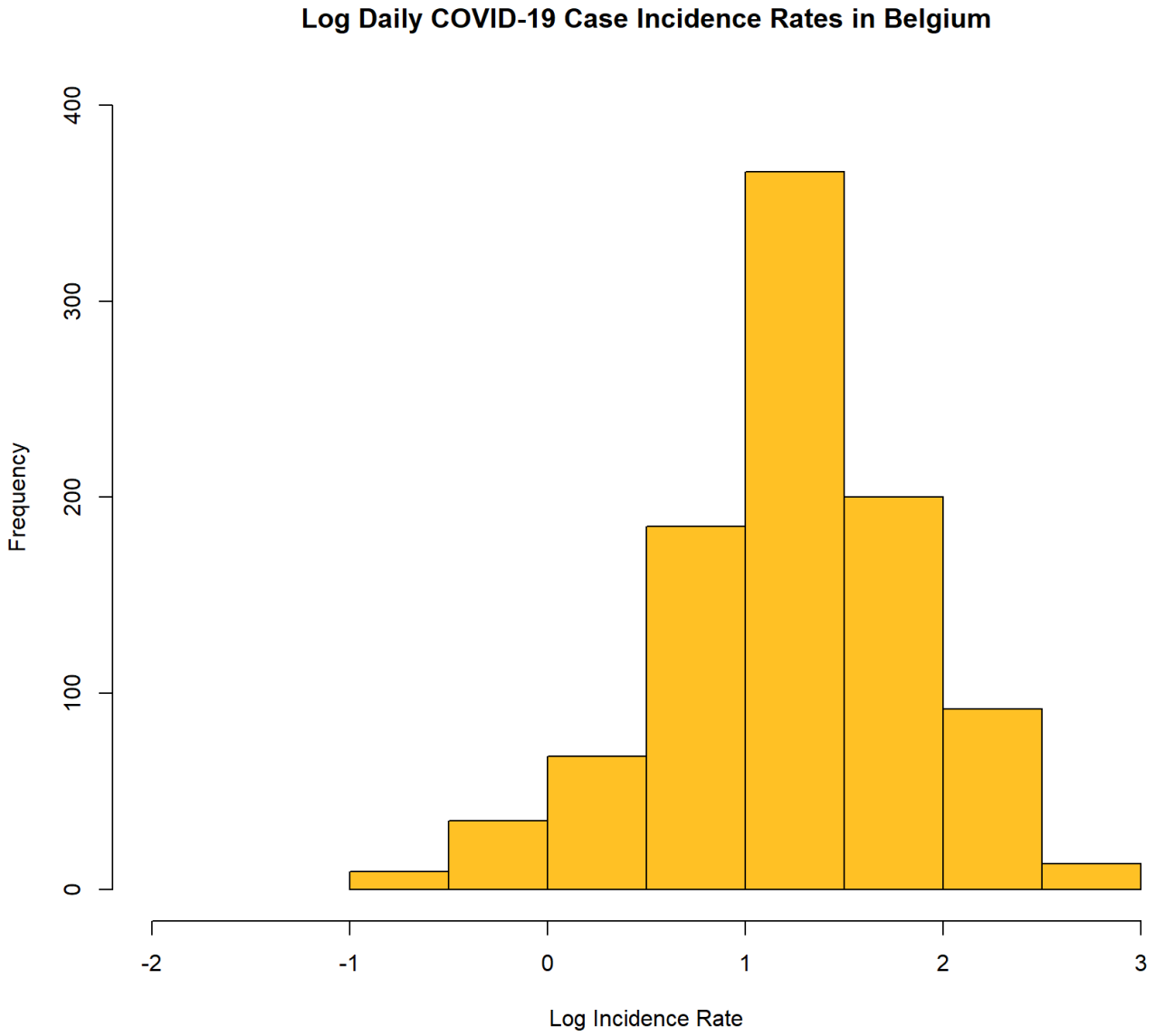
```
# Provide a justification for the statistical test selected
print("I am selecting the t-test for the difference in two means for populations with unknown population variances.  I'll use this test to compare the means of the log daily COIVD-19 case incidence rates in Belgium and Romania.  This statistical test has nice properties for our given scenario since we're interested in comparing two continuous variables and since we don't know the true population variances of log infection rates in either country.")
```

```
## [1] "I am selecting the t-test for the difference in two means for populations with unknown population variances.  I'll use this test to compare the means of the log daily COIVD-19 case incidence rates in Belgium and Romania.  This statistical test has nice properties for our given scenario since we're interested in comparing two continuous variables and since we don't know the true population variances of log infection rates in either country."
```

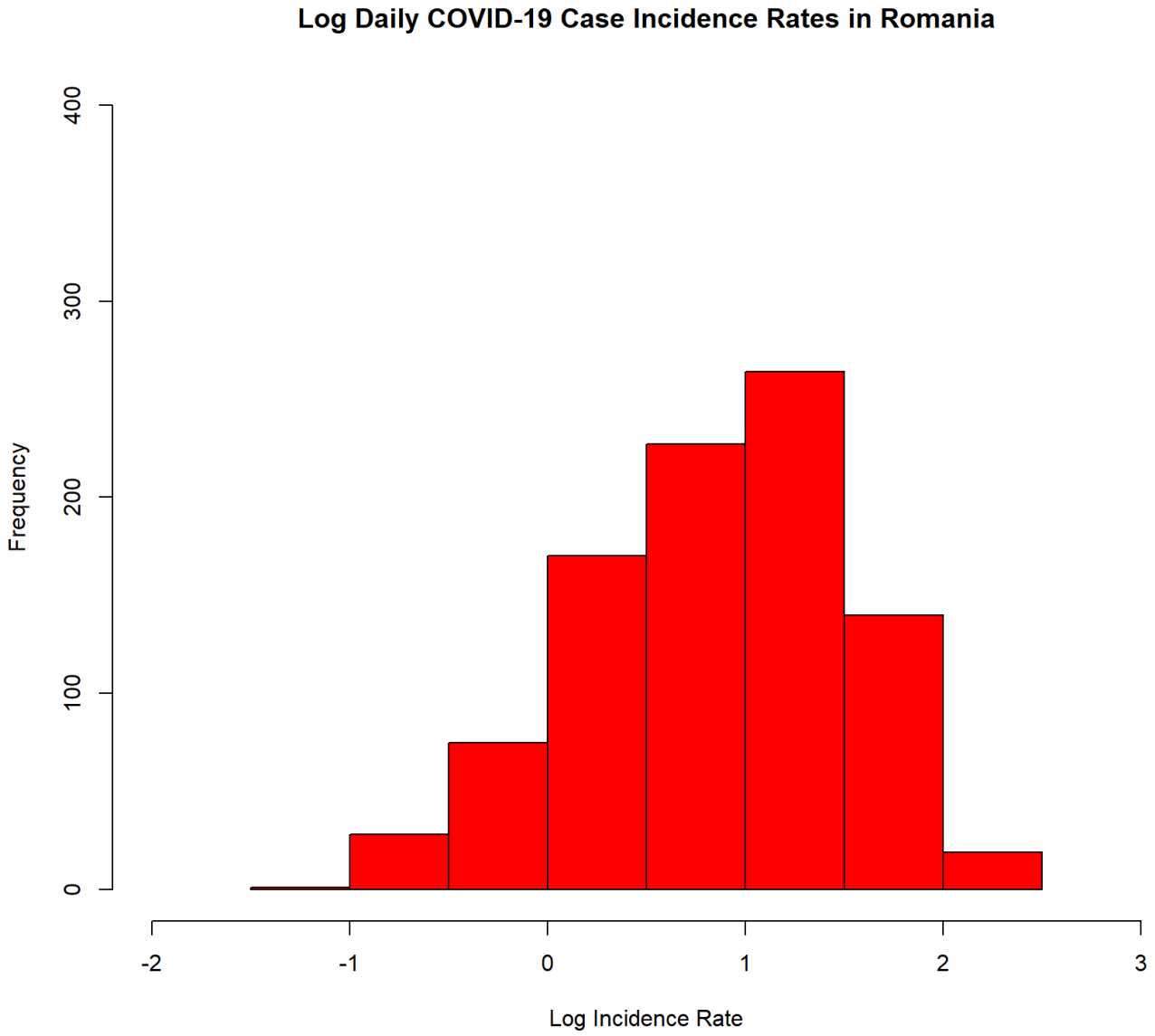
```
# Discuss the distributional assumptions of the test and convey whether our data satisfies those assumptions

print("One assumption necessary for a t-test is that underlying data are normally distributed.  As the table in part 1 conveys, all the incidence rate and fatality rate distributions in every country exhibit strong skew or kurtosis.  Consequently, we've performed a log transformation on the data to help satisfy this normality assumption.  Notably, though, the table in part 1 also displays that almost every country exhibited minimum fatality or incidence rates of less than or equal to zero during this time period, so using the log-transformed data for most countries would result in undefined values.  Fortunately, there are four countries (Belgium, Romania, Germany, and Austria), with minimum daily incidence rates above zero.  For that reason, we've chosen two of those four countries for this statistical test.  As we see in the histograms and Q-Q plots below, the normality assumption may not be 100% perfectly met, but this is as best as we can do given the strong non-normality of the original underlying data.  There also were no null values for incidence rates in these countries, which gives us better confidence in the cleanliness of the incidence rate data from these countries.")
```

```
## [1] "One assumption necessary for a t-test is that underlying data are normally distributed.  As the table in part 1 conveys, all the incidence rate and fatality rate distributions in every country exhibit strong skew or kurtosis.  Consequently, we've performed a log transformation on the data to help satisfy this normality assumption.  Notably, though, the table in part 1 also displays that almost every country exhibited minimum fatality or incidence rates of less than or equal to zero during this time period, so using the log-transformed data for most countries would result in undefined values.  Fortunately, there are four countries (Belgium, Romania, Germany, and Austria), with minimum daily incidence rates above zero.  For that reason, we've chosen two of those four countries for this statistical test.  As we see in the histograms and Q-Q plots below, the normality assumption may not be 100% perfectly met, but this is as best as we can do given the strong non-normality of the original underlying data.  There also were no null values for incidence rates in these countries, which gives us better confidence in the cleanliness of the incidence rate data from these countries."
```

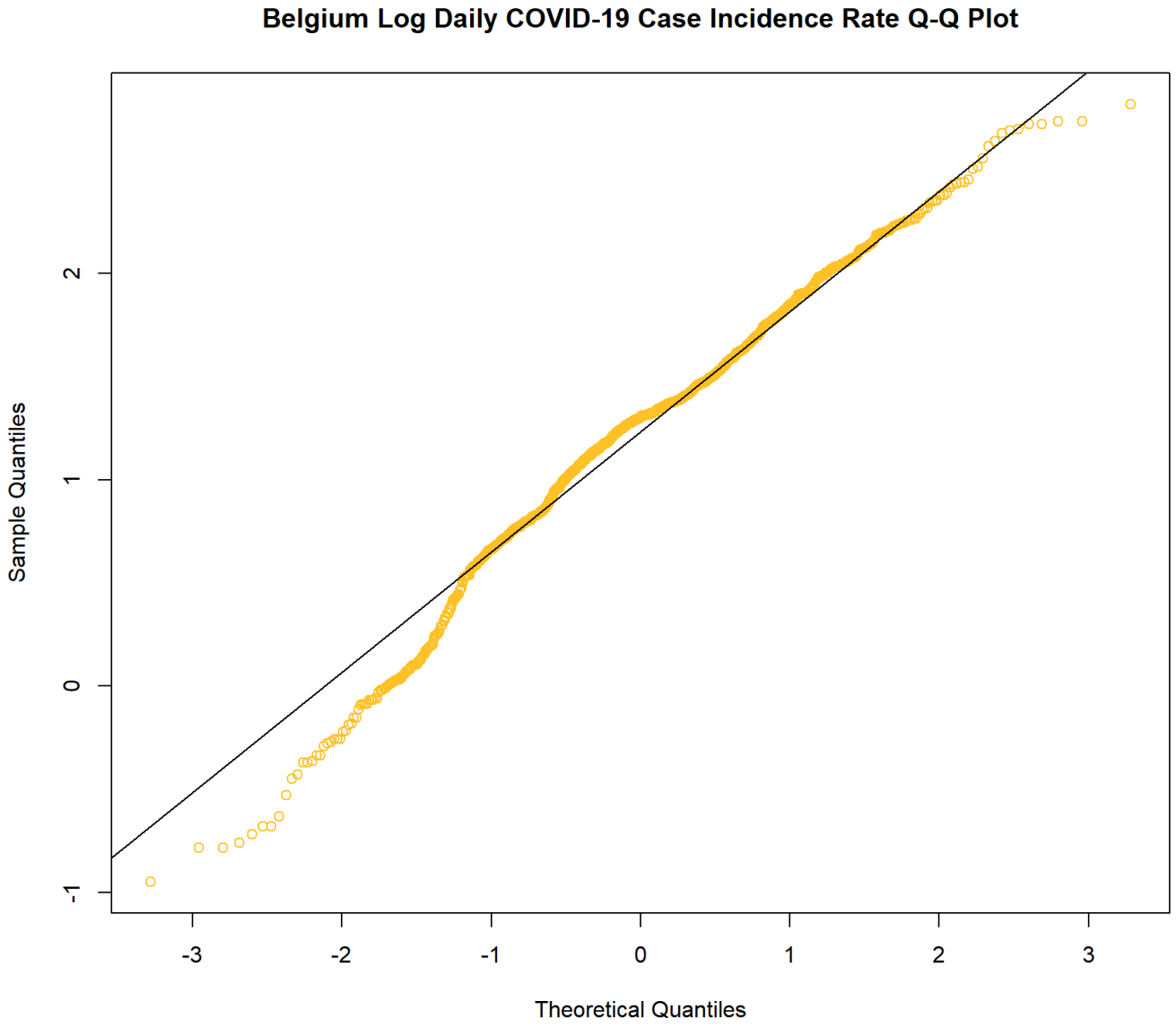
```
Belgium_data <- data[data$countriesAndTerritories == "Belgium", ]
Romania_data <- data[data$countriesAndTerritories == "Romania", ]

hist(Belgium_data$log_incidence_rate, xlab = "Log Incidence Rate", ylab = "Frequency", main = "Log Daily COVID-19 Case Incidence Rates in Belgium", col = "goldenrod1", xlim = c(-2, 3), ylim = c(0, 400))
```



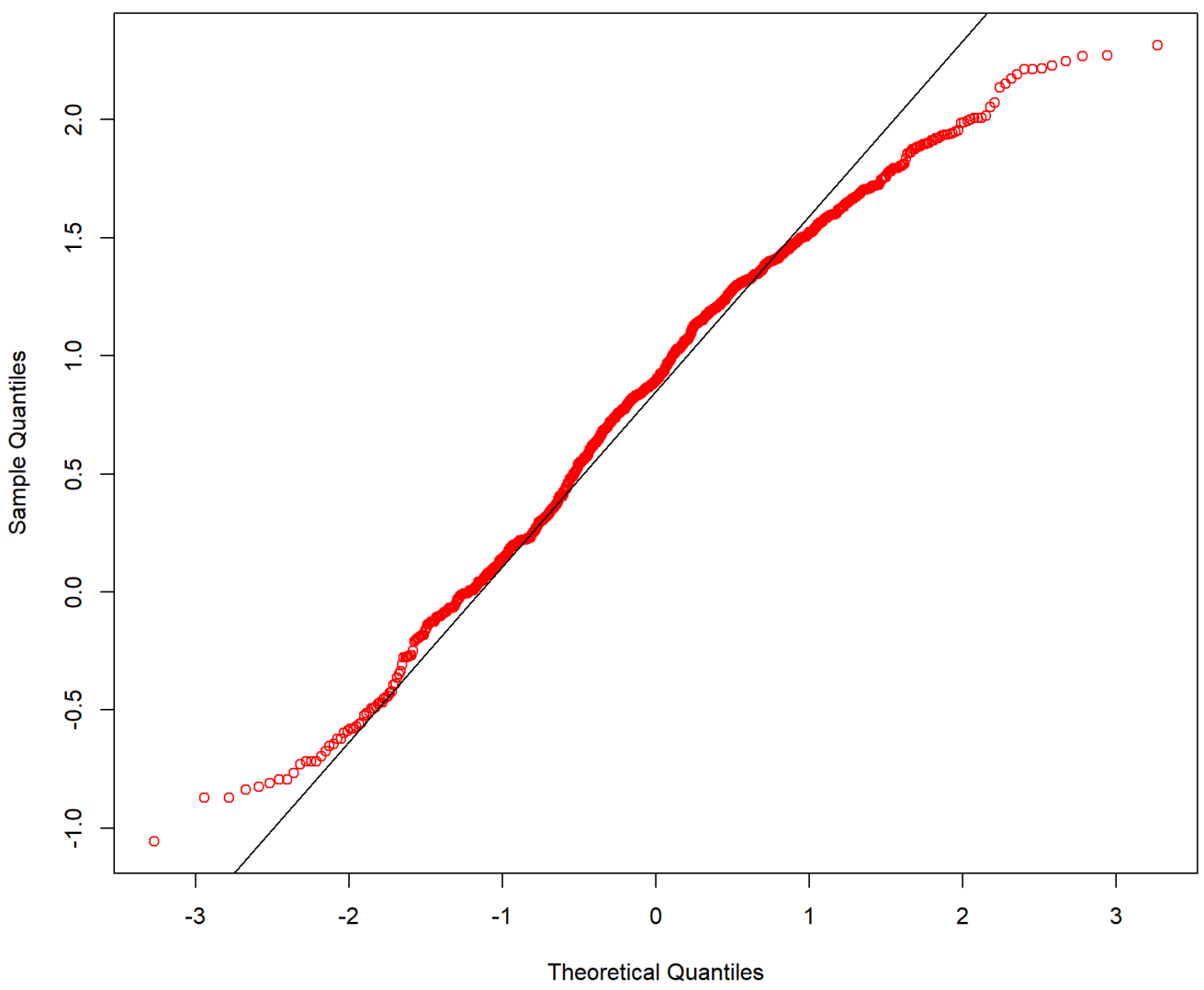

```
hist(Romania_data$log_incidence_rate, xlab = "Log Incidence Rate", ylab = "Frequency", main = "Log Daily COVID-19 Case Inci
dence Rates in Romania", col = "red", xlim = c(-2, 3), ylim = c(0, 400))
```




```
qqnorm(y = Belgium_data$log_incidence_rate, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", col= "goldenrod1", m
ain = "Belgium Log Daily COVID-19 Case Incidence Rate Q-Q Plot")
qqline(y = Belgium_data$log_incidence_rate, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7, col = "black")
```




```
qqnorm(y = Romania_data$log_incidence_rate, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", col= "red", main =
"Romania Log Daily COVID-19 Case Incidence Rate Q-Q Plot")
qqline(y = Romania_data$log_incidence_rate, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7, col = "black")
```




```
# State my selected alpha
print ("I have selected to conduct this statistical test at the 95% confidence level so the alpha for this test is 0.05.")
```

```
## [1] "I have selected to conduct this statistical test at the 95% confidence level so the alpha for this test is 0.05."
```

```
# Provide the outputs of my statistical test
t.test(x = Belgium_data$log_incidence_rate, y = Romania_data$log_incidence_rate, alternative = c("two.sided"), paired = FAL
SE, var.equal = FALSE, conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  Belgium_data$log_incidence_rate and Romania_data$log_incidence_rate
## t = 12.552, df = 1874, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3155481 0.4324151
## sample estimates:
## mean of x mean of y
## 1.2299245 0.8559429
```

```
# State the confidence interval corresponding to my statistical test
print("The 95% confidence interval for the difference of the average log daily COVID-19 incidence rates in Belgium and Roma
nia is (0.3155481, 0.4324151).")
```

```
## [1] "The 95% confidence interval for the difference of the average log daily COVID-19 incidence rates in Belgium and Rom
ania is (0.3155481, 0.4324151)."
```

```
# Write a concluding statement from the outcome of the statistical test
print("The t-test conveys that we should reject the null hypothesis and that the average log daily COVID-19 case incidence
rates in Belgium and Romania were not equal throughout the measurement period.")
```

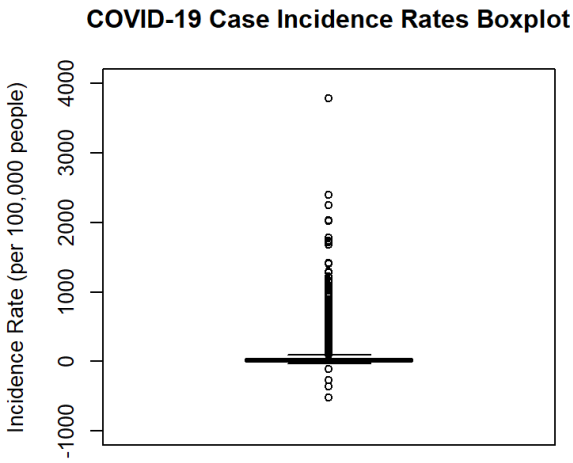
```
## [1] "The t-test conveys that we should reject the null hypothesis and that the average log daily COVID-19 case incidence
rates in Belgium and Romania were not equal throughout the measurement period."
```

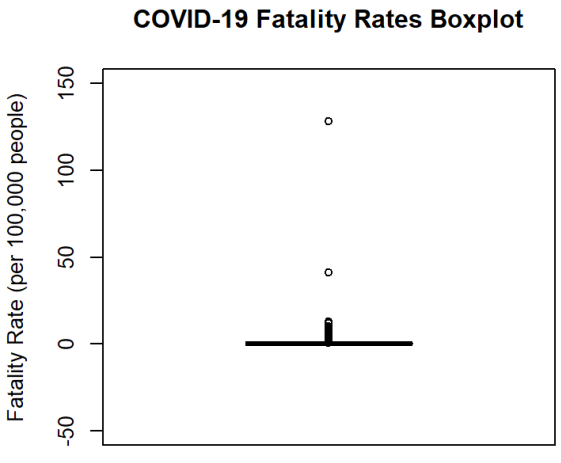
3. Correlation

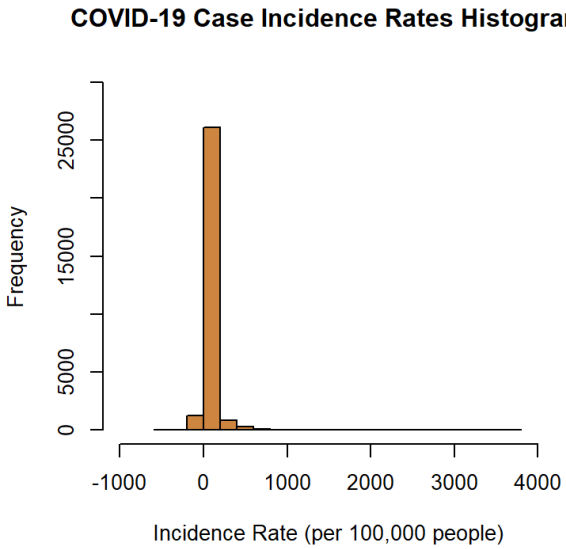
Considering all countries, explore the relationship between incidence rates and fatality rates. At minimum, your work should include the following:

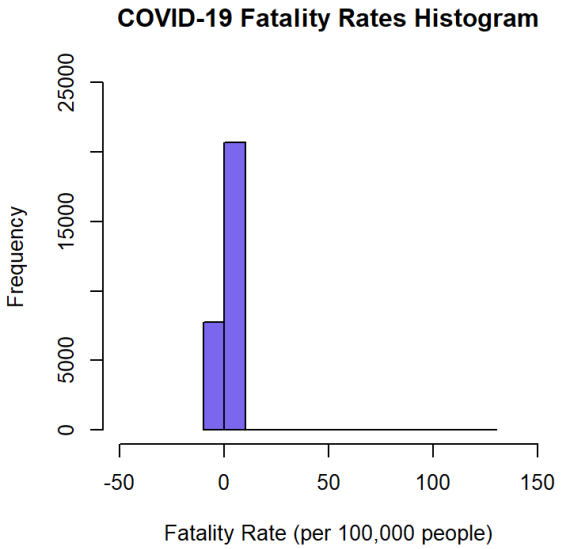
- Visualization(s) showing the distributions of daily incidence and fatality rates, regardless of country. Please note that both country and date should be disregarded here.
- A short statement identifying the most appropriate correlation coefficient.
    - For the correlation we're interested in, which 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()*.

```
# Create visualization(s) of showing the distribution of daily incidence and fatality rates


par(mfrow = c(2,2))

boxplot(data$incidence_rate, ylab = "Incidence Rate (per 100,000 people)", main = "COVID-19 Case Incidence Rates Boxplot",
col = "peru", ylim = c(-1000, 4000))

boxplot(data$fatality_rate, ylab = "Fatality Rate (per 100,000 people)", main = "COVID-19 Fatality Rates Boxplot", col = "s
lateblue2", ylim = c(-50, 150))

hist(x = data$incidence_rate, xlab = "Incidence Rate (per 100,000 people)", ylab = "Frequency", main = "COVID-19 Case Incid
ence Rates Histogram", col = "peru", xlim = c(-1000, 4000), ylim = c(0, 30000))

hist(x = data$fatality_rate , xlab = "Fatality Rate (per 100,000 people)", ylab = "Frequency", main = "COVID-19 Fatality Ra
tes Histogram", col = "slateblue2", xlim = c(-50, 150), ylim = c(0, 25000))
```

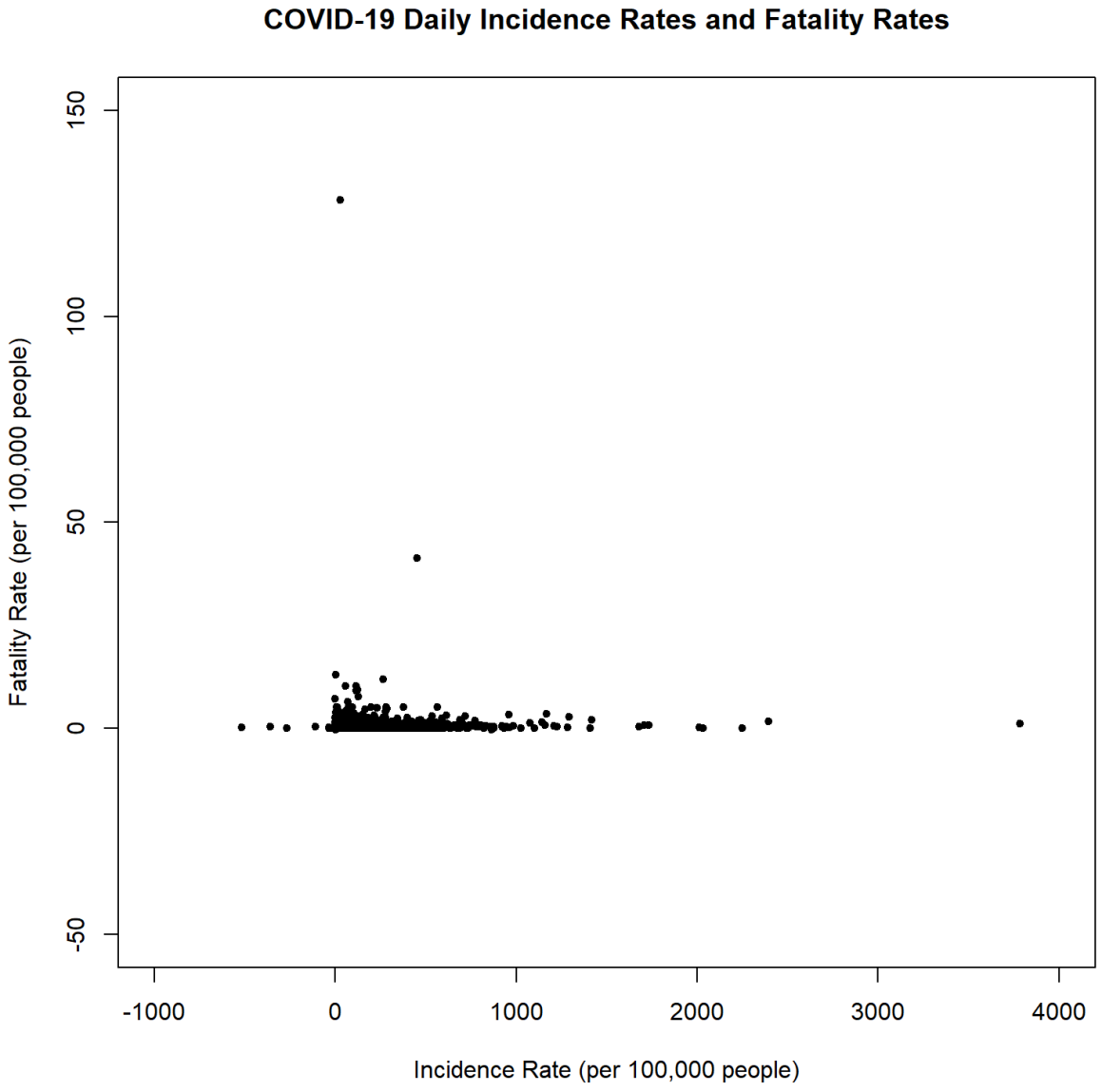









```
par(mfrow = c(1,1))

plot(x = data$incidence_rate, y = data$fatality_rate, ylab = "Fatality Rate (per 100,000 people)", xlab = "Incidence Rate
(per 100,000 people)", main = "COVID-19 Daily Incidence Rates and Fatality Rates", pch = 20, ylim = c(-50, 150), xlim = c(-
1000, 4000))
```




```
print("Unfortunately, the plots displaying the distributions of incidence rates and fatality rates in all countries are qui
te hard to read - largely due to a small number of extreme outliers in these distributions.  The fact that these plots are
difficult to read reflects the broader trend that this data is quite messy.  For example, these outliers suggest that nearl
y 3.8% of one country's population newly reported a COVID-19 case on the same day and that over 0.3% of one country's popul
ation died of COVID-19 on the same day.  Both of these figures seem extremely high and would normally warrant further inves
tigation.  Furthermore, there are 26,816 rows of null values for deaths in this dataset and 26,921 rows of null values for
cases in this dataset as well.  These null values would also normally warrant further investigation to determine whether th
e missing data can be retrieved.  Additionally, five countries reported negative cases on some days and nine countries repo
rted negative deaths on some days.  Since negative cases and deaths are not possible, these values would also normally warr
ant investigation or potentially even removal from the dataset.  However, since I have no further power to investigate the
sources of these data, I cannot fix these data cleanliness problems, so I cannot make these plots any more legible.  Furthe
rmore, I would normally consider the log transformation of some of these data to make the visualizations more readable, but
I cannot do that either since there are many zeros in these datasets and the log of zero is undefined.")
```

```
## [1] "Unfortunately, the plots displaying the distributions of incidence rates and fatality rates in all countries are qu
ite hard to read - largely due to a small number of extreme outliers in these distributions.  The fact that these plots are
difficult to read reflects the broader trend that this data is quite messy.  For example, these outliers suggest that nearl
y 3.8% of one country's population newly reported a COVID-19 case on the same day and that over 0.3% of one country's popul
ation died of COVID-19 on the same day.  Both of these figures seem extremely high and would normally warrant further inves
tigation.  Furthermore, there are 26,816 rows of null values for deaths in this dataset and 26,921 rows of null values for
cases in this dataset as well.  These null values would also normally warrant further investigation to determine whether th
e missing data can be retrieved.  Additionally, five countries reported negative cases on some days and nine countries repo
rted negative deaths on some days.  Since negative cases and deaths are not possible, these values would also normally warr
ant investigation or potentially even removal from the dataset.  However, since I have no further power to investigate the
sources of these data, I cannot fix these data cleanliness problems, so I cannot make these plots any more legible.  Furthe
rmore, I would normally consider the log transformation of some of these data to make the visualizations more readable, but
I cannot do that either since there are many zeros in these datasets and the log of zero is undefined."
```

```
# Provide a short statement identifying the most appropriate correlation coefficient

print("The most appropriate correlation coefficient in this scenario is the Spearman correlation coefficient, which works w
ell with variables containing data that are not normally distributed.  Since the fatality rates and incidence rates are bot
h ratio-level data and since both variables have extremely strong outliers and skewness that suggest that the data are not
normally distributed, we can proceed with the Spearman correlation.")
```

```
## [1] "The most appropriate correlation coefficient in this scenario is the Spearman correlation coefficient, which works
well with variables containing data that are not normally distributed.  Since the fatality rates and incidence rates are bo
th ratio-level data and since both variables have extremely strong outliers and skewness that suggest that the data are not
normally distributed, we can proceed with the Spearman correlation."
```

```
# Provide the calculated correlation coefficient / coefficient test output

non_null_data  <- data[ -c(1:10) ]

non_null_data <- na.omit(non_null_data)

cor(x = non_null_data$incidence_rate, y = non_null_data$fatality_rate, method = "spearman" )
```

```
## [1] 0.5791862
```

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?

```
# Provide a description of the model_df dataframe

str(model_df)
```

```
## 'data.frame':    20 obs. of  6 variables:
##  $ country    : Factor w/ 30 levels "Austria","Belgium",..: 1 2 3 5 7 9 10 11 13 15 ...
##  $ pop        : int  8901064 11522440 6951482 888005 5822763 5525292 67320216 83166711 9769526 4964440 ...
##  $ sq_km      : num  83858 30510 110994 9251 44493 ...
##  $ gdp_pps    : num  128 118 51 91 129 111 104 123 71 190 ...
##  $ pop_dens   : num  106.1 377.7 62.6 96 130.9 ...
##  $ total_cases: int  5402162 4607296 1275481 596297 3219571 1335318 36612627 35287690 2141513 1670377 ...
```

```
print("The model_df dataframe consist of 20 observations of 6 variables. Data are summarized at the country-level for 20 co
untries.  Consequently, one of our variables is the country, which is a nominal variable.  The other five variables are rat
io-level variables representing: 1) the population of the country, 2) the area of the country in square kilometers, 3) the
Gross Domestic Product per capita (in purchasing power standard), 4) the population density (in people per square kilomete
r), and 5) the total cases in the country during the measurement period ")
```

```
## [1] "The model_df dataframe consist of 20 observations of 6 variables. Data are summarized at the country-level for 20 c
ountries.  Consequently, one of our variables is the country, which is a nominal variable.  The other five variables are ra
tio-level variables representing: 1) the population of the country, 2) the area of the country in square kilometers, 3) the
Gross Domestic Product per capita (in purchasing power standard), 4) the population density (in people per square kilomete
r), and 5) the total cases in the country during the measurement period "
```

```
#Provide the output of the fitted model

model_4C <- lm(total_cases ~ pop + pop_dens + gdp_pps , data = model_df)
summary(model_4C)
```

```
##
## Call:
## lm(formula = total_cases ~ pop + pop_dens + gdp_pps, data = model_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8241030 -1086127    -9952  1651494  8715365
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.859e+06  2.749e+06  -1.404    0.179
## pop          4.268e-01  3.625e-02  11.774 2.71e-09 ***
## pop_dens     6.499e+02  2.399e+03   0.271    0.790
## gdp_pps      2.831e+04  2.524e+04   1.121    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3655000 on 16 degrees of freedom
## Multiple R-squared:  0.8982, Adjusted R-squared:  0.8791
## F-statistic: 47.03 on 3 and 16 DF,  p-value: 3.686e-08
```

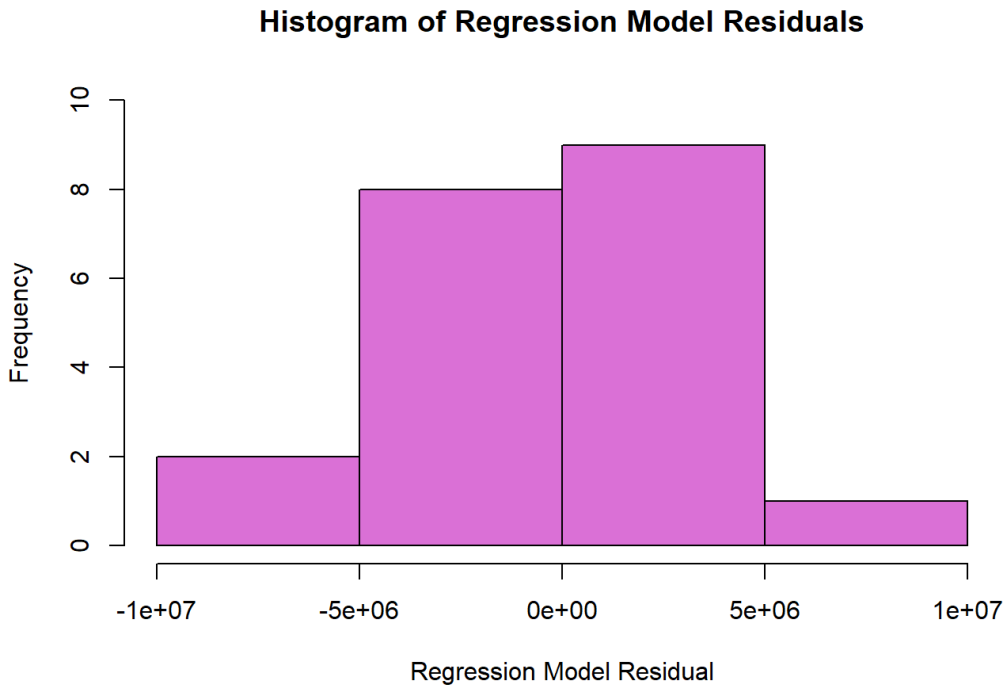
```
# Write a short statement about the fit of the model.  Include commentary about which coefficients are statistically signif
icant, what the R^2 of the model is, and whether we should consider a model with fewer parameters

print("The R squared of this model is 0.8982 and the adjusted R squared of this model is 0.8791.  Though the model appears
to explain a high percentage of the variation in total cases, only one of the predictors in the model is actually statistic
ally significant at the 0.05 level. Specifically, population is highly statistically signficant.  Meanwhile, population den
sity and GDP per capita are not statistically signficant at the 0.05 level.  As a result, if we want to use this model for
prediction, then we may want to consider fitting a model without the statistically insignificant predictors so that we are
not overfitting to noise from the testing dataset when predicting.")
```

```
## [1] "The R squared of this model is 0.8982 and the adjusted R squared of this model is 0.8791.  Though the model appears
to explain a high percentage of the variation in total cases, only one of the predictors in the model is actually statistic
ally significant at the 0.05 level. Specifically, population is highly statistically signficant.  Meanwhile, population den
sity and GDP per capita are not statistically signficant at the 0.05 level.  As a result, if we want to use this model for
prediction, then we may want to consider fitting a model without the statistically insignificant predictors so that we are
not overfitting to noise from the testing dataset when predicting."
```

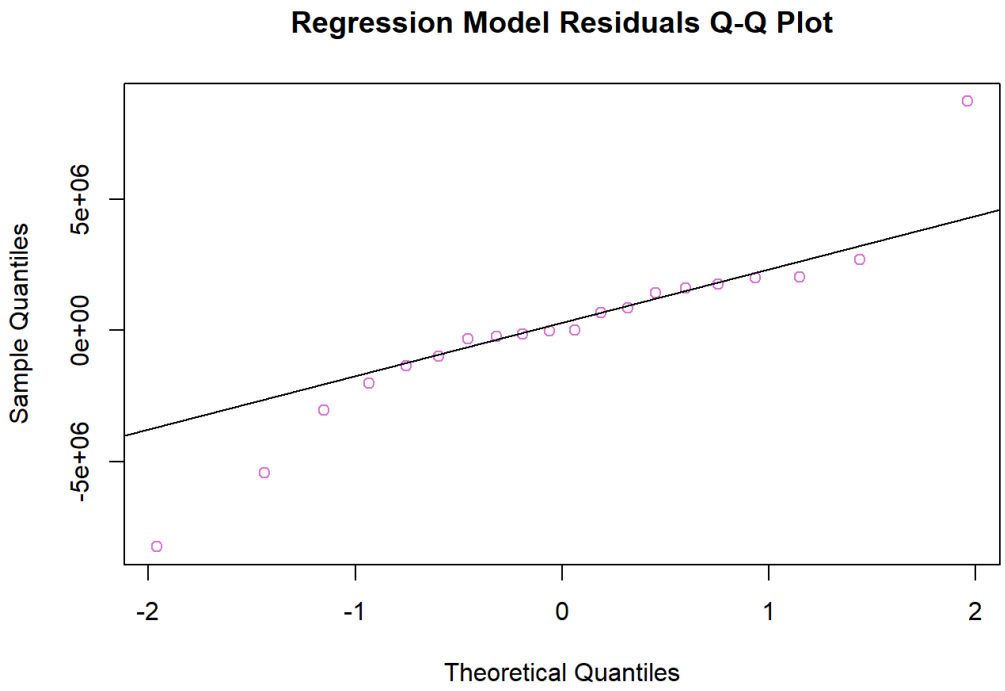
```
# Bonus Analysis: Examine the residuals to see how well the linear regression assumptions are being met

print("Regression analysis makes four assumptions about regression models: 1) the model is linear, 2) the error terms have
constant variances, 3) the error terms are independent, and 4) the error terms are normally distributed. Below, we  plot a
histogram, Q-Q Plot and scatterplot of the residuals to examine how well these four assumptions are met.  The residual plot
s below suggest that while the normality and indpendence assumptions may be somewhat met, the linearity and homoscedasticit
y assumptions are likely not met, so this may not be the best model imaginable.")
```

```
## [1] "Regression analysis makes four assumptions about regression models: 1) the model is linear, 2) the error terms have
constant variances, 3) the error terms are independent, and 4) the error terms are normally distributed. Below, we  plot a
histogram, Q-Q Plot and scatterplot of the residuals to examine how well these four assumptions are met.  The residual plot
s below suggest that while the normality and indpendence assumptions may be somewhat met, the linearity and homoscedasticit
y assumptions are likely not met, so this may not be the best model imaginable."
```

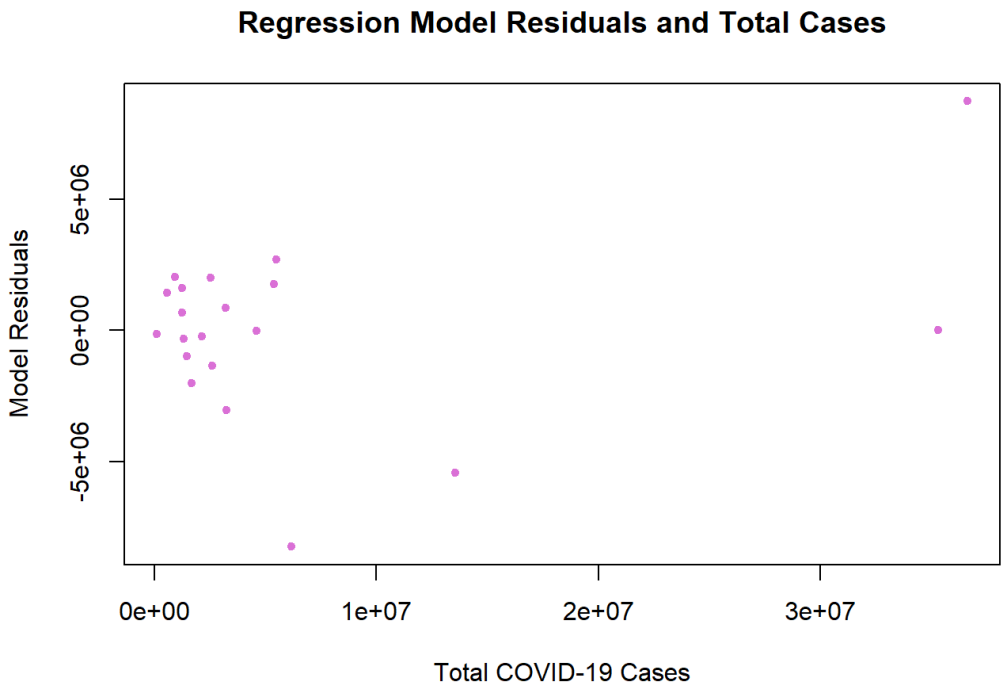
```
hist(residuals(model_4C), xlab = "Regression Model Residual", ylab = "Frequency", main = "Histogram of Regression Model Res
iduals", col = "orchid", ylim = c(0, 10))
```




```
qqnorm(y = model_4C$residuals, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", col= "orchid", main = "Regression
Model Residuals Q-Q Plot")
qqline(y = model_4C$residuals, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7, col = "black")
```




```
plot(x = model_df$total_cases, y = model_4C$residuals, ylab = "Model Residuals", xlab = "Total COVID-19 Cases", main = "Reg
ression Model Residuals and Total Cases", pch = 20, col = "orchid")
```



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.

Luxembourg_data <- data[data$countriesAndTerritories == "Luxembourg", ]
Netherlands_data <- data[data$countriesAndTerritories == "Netherlands", ]

sum(Luxembourg_data$cases)
```

```
## [1] 301031
```

```
sum(Netherlands_data$cases)
```

```
## [1] 8494705
```

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

predict(model_4C, newdata = newdata)
```

```
##       1       2
## 3953237 7522800
```

```
# Provide commentary about how well the model performed at these two predictions, including how well these predictions comp
are to those made for countries in the fitted dataset

model_predicted_values <- predict(model_4C, newdata = model_df)

model_df <- cbind(model_df, model_predicted_values, model_4C$residuals)

model_df$absolute_percent_error <- abs(((model_df$model_predicted_values - model_df$total_cases) / model_df$total_cases) *
100)

mean(model_df$absolute_percent_error)
```

```
## [1] 75.25618
```

```
print("For Luxembourg, the model predicted 3,953,237 cases, which was 3,652,206 more than the actual case count of 301,031.
For the Netherlands, the model predicted 7,522,800 cases, which was 971,905 fewer than  the actual case count of 8,494,705.
The model performed reasonably well at predicting the total cases for the Netherlands (since it was only off by roughly 11.
4%).  However, with a percent error of 1,213.2%, the model did not perform well at predicting the total case count in Luxem
bourg.  Similarly, the model did not perform well at predicting the total cases for the countries in the training dataset,
since the mean absolute percent error of those 20 predictions was over 75%.")
```

```
## [1] "For Luxembourg, the model predicted 3,953,237 cases, which was 3,652,206 more than the actual case count of 301,03
1.  For the Netherlands, the model predicted 7,522,800 cases, which was 971,905 fewer than  the actual case count of 8,494,
705.  The model performed reasonably well at predicting the total cases for the Netherlands (since it was only off by rough
ly 11.4%).  However, with a percent error of 1,213.2%, the model did not perform well at predicting the total case count in
Luxembourg.  Similarly, the model did not perform well at predicting the total cases for the countries in the training data
set, since the mean absolute percent error of those 20 predictions was over 75%."
```