ARIMA Time Series Analysis of Telecom Revenue

Tucker Atwood, WGU MSDA

2024-06-05

A1. Research Question

Using the provided churn time series data of telecommunications revenue by day, I will answer the question, **"Can an ARIMA time series model be used to forecast future telecom revenue?"**

A2. Analysis Goal

The goal of this analysis is to **develop an Autoregressive Integrated Moving Average (ARIMA) time series model** capable of providing insight into telecom **revenue patterns**, so that **predictions** of future revenue may be made. While customer churn data is not provided for this analysis, the larger goal of the telecom stakeholders is to **decrease churn rate** by analyzing associated factors. By developing an understanding of revenue patterns and using the insights to make predictions of future revenue, stakeholders will be better equipped to analyze the impact of customer churn on revenue.

B. ARIMA Assumption

The following assumptions are relevant to conducting an ARIMA time series model:

- **The data must exhibit stationarity**, which means the mean, variance, and autocorrelation remain relatively constant and do not change as a function of time. This implies that there must be no overall trend or seasonality in the data.
- **The data must not be significantly autocorrelated**, which means individual time values are not correlated with their prior values. This implies that all observations must be independent of each other.
- If there is evidence that either of these assumptions are not met, a method of **differencing** the data may be used to remove trends and make the data **stationary**.

C1. Time Series Visualization

To prepare the data for ARIMA analysis, a **visualization** of the time series data will first be inspected. To prepare this visualization, the following steps will be taken. Note: all input and output code included in this report will be provided in the language **R**:

- The data set must be **imported**.

```
# imports data set
df <- read.csv("C:/Users/atwoo/OneDrive/Desktop/MSDADirectory/teleco_time_series.csv")
```

- The necessary **packages** for analysis must be imported. The relevant packages are:
  - **tseries**, which will allow the usage of the adf.test function. This will perform an augmented Dickey-Fuller hypothesis test to determine stationarity.
  - **astsa**, which will allow the usage of the mvspec function. This will perform the spectral density test to determine seasonality.
  - **forecast**, which will allow the usage of the auto.arima and Arima functions. These will be essential to the selection and execution of the optimal ARIMA model.

```
# imports packages to be used in analysis
library(tseries) # using tseries package to utilize adf.test function
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(astsa) # using astsa package to utilize mvspec function
library(forecast) # using forecast package to utilize auto.arima and Arima functions
```

```
##
## Attaching package: 'forecast'
```

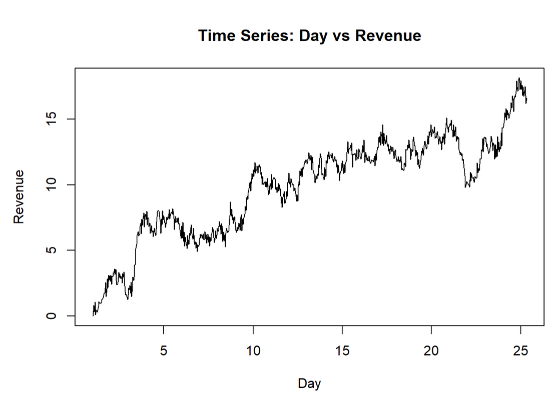
```
## The following object is masked from 'package:astsa':
##
##     gas
```

- The data must be converted into a **time series** object, with a starting value of 1 and a frequency of 30 to represent monthly periods within the data.

```
# converts data into a time series
ts <- ts(df$Revenue, start = 1, frequency = 30)
```

- The **realization of the time series** may now be visualized graphically.

```
# creates time series plot, no noticeable outliers, trend: upward linear (not stationary)
plot(ts, xlab = "Day", ylab = "Revenue", main = "Time Series: Day vs Revenue")
```



C2. Time Step Formatting

To analyze the **time step formatting** of the time series, the following steps will be taken:

- **Missing values** must be checked for.

```
# checks for missing values
sum(is.na(df))
```

```
## [1] 0
```

- An analysis of the **structure** of the time series must be conducted, including the **start** and **end** of the series, as well as a check for any **duplicated day values**.

```
# checks structure of time series and checks for duplicate day values
str(df)
```

```
## 'data.frame':    731 obs. of  2 variables:
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Revenue: num  0 0.000793 0.825542 0.320332 1.082554 ...
```

```
start(df$Day)
```

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

```
end(df$Day)
```

```
## [1] 731   1
```

```
sum(duplicated(df$Day))
```

```
## [1] 0
```

From the given outputs, the following can be determined:

- The **length** of the sequence is **731 days**, which represents two years of data when one year is a leap year.
- There are **no gaps** in measurement, since there are 731 observations, the start of the sequence is 1, the end is 731, and there are no duplicates.
- Each observation represents **one day of revenue** for the telecommunications company, and the interval between observations, one day, is consistent throughout the data.

C3. Stationarity

To evaluate the **stationarity** of the time series, an **augmented Dickey-Fuller (ADF) hypothesis test** will be run on the data. The **null hypothesis** is that the data is **not stationary**, the **alternative hypothesis** is that the data is **stationary**, and the **alpha level** will be set to **0.01**.

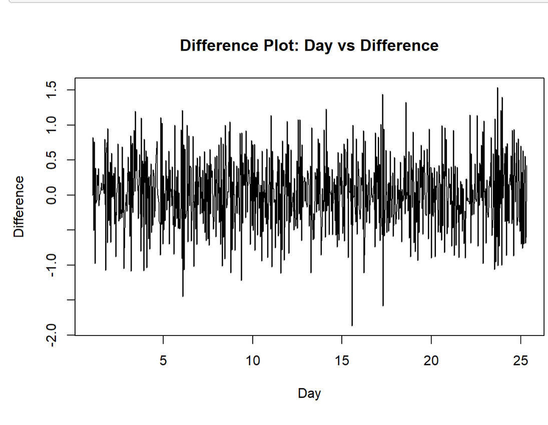
```
# runs augmented Dickey-Fuller hypothesis test
adf.test(ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts
## Dickey-Fuller = -3.6938, Lag order = 9, p-value = 0.02431
## alternative hypothesis: stationary
```

The **p-value** of the ADF test is **0.02431**. Since this is above the determined alpha level, there is **not enough evidence to reject the null hypothesis**; the data may not be stationary. This is also apparent in the above visualization of the time series, where a general **upward trend** is apparent.

To coerce the data to stationarity, the data will be **differenced**. This involves subtracting the previous value from each individual observation, which creates a time series of the differences between each observation. Later, differencing will also be reflected in the ARIMA model selection.

```
# creates differenced time series
diff_ts <- diff(ts)
plot(diff_ts, xlab = "Day", ylab = "Difference", main = "Difference Plot: Day vs Difference")
```



A **second ADF test** will be conducted to determine if the differenced data exhibits stationarity. The null hypothesis, alternative hypothesis, and alpha level will be the same as the previous ADF test.

```
# re-runs ADF test on differenced data
adf.test(diff_ts)
```

[illegible]

C4. Data Preparation

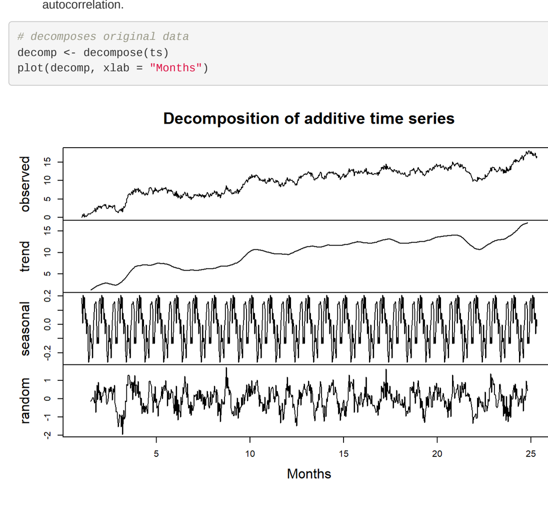
[illegible]

C5. Cleaned Data Sets

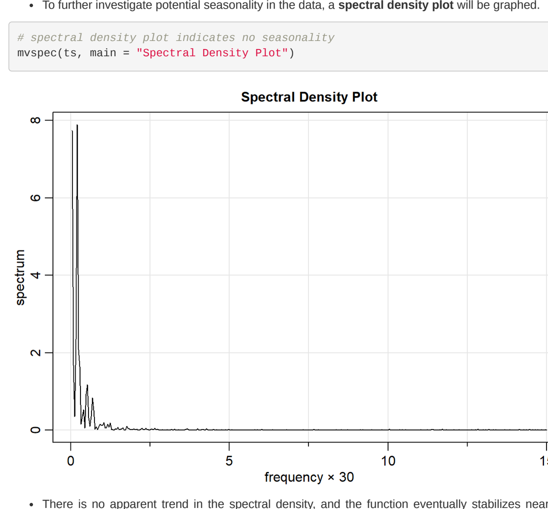
[illegible]

D1. Time Series Analysis

[illegible]

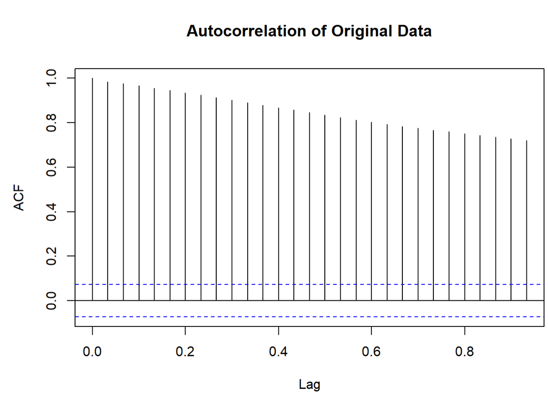


[illegible]

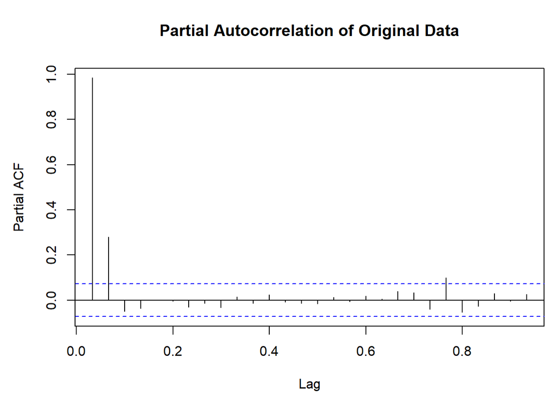


- There is no apparent trend in the spectral density, and the function eventually stabilizes near zero. This indicates there is **no apparent seasonality** in the data.
- The **autocorrelation** and **partial autocorrelation** functions will be examined for both the **original** data and the **differenced** data.

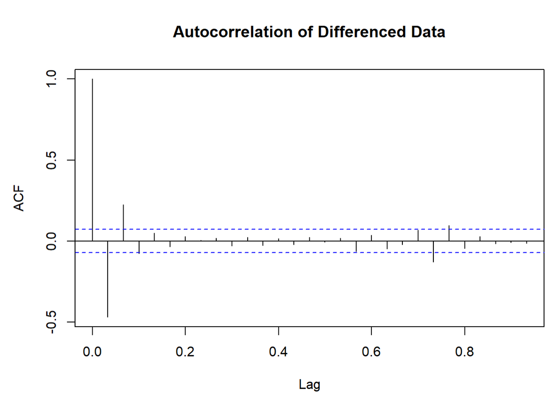
```
# autocorrelation function and partial autocorrelation function of differenced data
par(mar=c(5,4,4,2) + 0.1)
acf(ts, main = "Autocorrelation of Original Data")
```



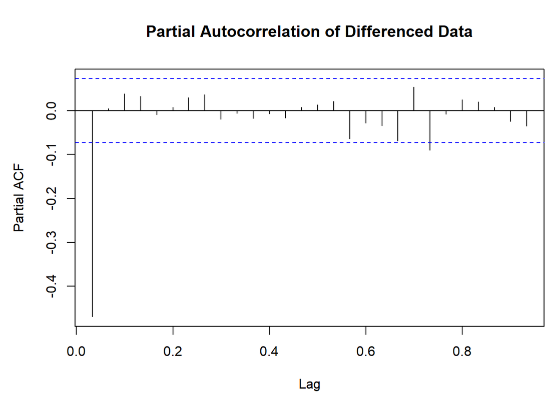

```
pacf(ts, main = "Partial Autocorrelation of Original Data")
```




```
acf(diff_ts, main = "Autocorrelation of Differenced Data")
```




```
pacf(diff_ts, main = "Partial Autocorrelation of Differenced Data")
```



- These visualizations identify the following aspects of the data:
  - The ACF of the original data, which gradually decreases but remains significant for several lags, appears to confirm that the **original data was not stationary**, and differencing the data was a correct measure to take.
  - The PACF of the original data, which shows significant values for the first two lags, indicates that the eventual ARIMA model will have an **autoregressive (AR) order**.
  - For the differenced data, the ACF decays slowly while the PACF cuts off abruptly, which also implies an **AR model** will fit best.

D2. ARIMA Model Identification

To identify the most likely optimal ARIMA model, an **auto.arima function** will be used on the **training** time series data. This will provide the following information about the optimal model:

- The **p** value represents the autoregressive (AR) order of the model. An AR model is built by regressing individual observations on their previous values. The order of the model represents the number of regressions run on the model. This p value is separate from the p-value involved in hypothesis testing.
- The **d** value represents the integrated (I) order of the model. This indicates that the data has been differenced. The value represents the number of times the data has been differenced. For this analysis, it has already been hypothesized that the data should be differenced; however, the original data will be fed into the auto.arima function so that differencing may be confirmed to have been the correct measure to take.
- The **q** value represents the moving average (MA) order of the model. An MA model is built by regressing individual observations on previous error values. The order of the model represents the number of regressions run on the model.
- An ARIMA model may or may not be conducted with **drift**, which represents a constant value added into the regression model.
- An ARIMA model is described by its p, d, and q values in the manner: ARIMA(p,d,q).

The results of the auto.arima function are summarized as follows:

```
# finds optimal ARIMA model
auto.arima(train)
```

```
## Series: train
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1   drift
##       -0.4597  0.0234
## s.e.   0.0367  0.0132
##
## sigma^2 = 0.2187:  log likelihood = -383.95
## AIC=773.89   AICc=773.93   BIC=787
```

The model identified as the optimal ARIMA model is **ARIMA(1,1,0) with drift**. The fit of the model can be described by the Akaike Information Criterion (AIC) score, which is 773.89. This metric can be used to compare different ARIMA models to evaluate their fit for the time series data.

To check the results of the auto.arima function, select other ARIMA models will be conducted, with their AIC values measured, to determine whether the designated ARIMA(1,1,0) model with drift will be the best fit for this data.

```
# compares auto.arima's optimal order (1,1,0) to a few other ARIMA models
AM1 <- Arima(train, order = c(1,1,0), include.drift = TRUE)
AM2 <- Arima(train, order = c(0,1,0), include.drift = TRUE)
AM3 <- Arima(train, order = c(2,1,0), include.drift = TRUE)
AM4 <- Arima(train, order = c(1,1,1), include.drift = TRUE)
AM5 <- Arima(train, order = c(1,0,0), include.drift = TRUE)
AM6 <- Arima(train, order = c(0,1,1), include.drift = TRUE)
aic_vals <- data.frame(Arima_Order = c("1,1,0","0,1,0","2,1,0","1,1,1","1,0,0","0,1,1"),
                       AIC_value = c(AM1$aic, AM2$aic, AM3$aic, AM4$aic, AM5$aic, AM6$aic))
aic_vals
```

```
##   Arima_Order AIC_value
## 1       1,1,0  773.8926
## 2       0,1,0  910.7897
## 3       2,1,0  775.8863
## 4       1,1,1  775.8875
## 5       1,0,0  897.3403
## 6       0,1,1  799.4644
```

As confirmed in the above results, the ARIMA(1,1,0) model with drift has the lowest AIC value among the models tested. Therefore, the model to be used in this analysis will be **ARIMA(1,1,0) with drift**. A summary of the selected model is provided:

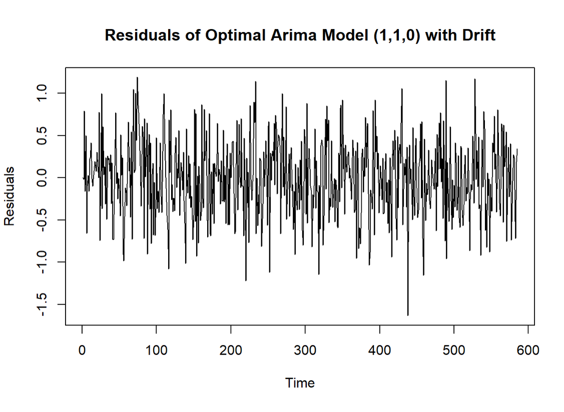
```
# summary of optimal ARIMA model, including evaluation metrics
summary(AM1)
```

```
## Series: train
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1   drift
##       -0.4597  0.0234
## s.e.   0.0367  0.0132
##
## sigma^2 = 0.2187:  log likelihood = -383.95
## AIC=773.89   AICc=773.93   BIC=787
##
## Training set error measures:
##                         ME      RMSE       MAE  MPE MAPE      MASE         ACF1
## Training set -1.378086e-05 0.4664731 0.3757458 -Inf  Inf 0.8764001 -0.002299464
```

Among the information provided by the summary of the ARIMA model is the **Root Mean Squared Error (RMSE) of 0.4665**, which will be used in section E1 to assist in evaluating the performance of the model.

To ensure the model is a good fit, the residuals of the model will be visualized to ensure a lack of trends.

```
# residual plot of optimal ARIMA
res <- residuals(AM1)
plot(res, main = "Residuals of Optimal Arima Model (1,1,0) with Drift", ylab = "Residuals")
```

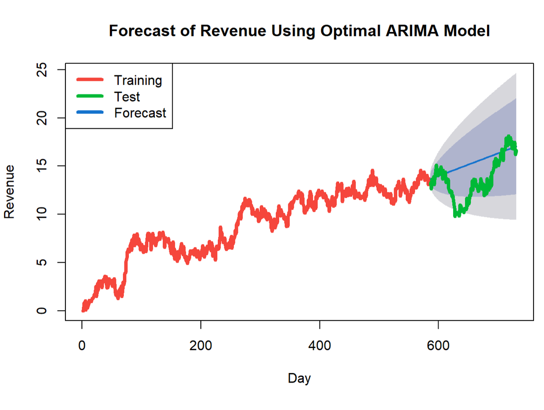


There does not appear to be any trend or pattern in the residuals of the ARIMA model, which indicates the **model fits the time series data well**.

D3. ARIMA Model Forecast

The ARIMA(1,1,0) model with drift will be used to **forecast** future revenue values aligning with the length of the test set. Since an 80-20 split was chosen for the training-test sets, there were 585 observations included in the training set and 146 observations included in the test set. Therefore, the ARIMA model will be used to forecast **the next 146 values following the training set**. Plotting the forecasted data against the test set will assist in determining whether the ARIMA model was successful in predicting future revenue values. An **80% confidence interval** and **95% confidence interval** for the forecasted data will also be provided.

```
# plots training set, forecast data, forecast confidence interval, and test set
fc <- forecast(AM1, h = 146)
plot(fc, col = 2, lwd = 4, xlab = "Day", ylab = "Revenue", main = "Forecast of Revenue Using Optimal ARIMA Mode
l")
points(test, type = "l", col = 3, lwd = 4)
legend("topleft", legend = c("Training","Test","Forecast"), col = c(2,3,4), lty = 1, lwd = 4)
```



D4. Output and Calculations

All **output and calculations** associated with this analysis, including data preparation, model selection, model execution, and forecasting, have been provided above as part of the R Markdown document.

D5. Implementation Code

All **implementation code** associated with this analysis has been provided above as part of the R Markdown document.

E1. Analysis Results

The ARIMA time series analysis and **results** described above can be summarized as follows:

- The provided data set was converted into a time series, cleaned, and split into training and test sets.
- In evaluating the stationarity of the time series, it was determined that the data must be differenced in order to remove non-stationarity.
- It was estimated that the best ARIMA model may be an autoregressive (AR) model.
- It was determined there was no seasonality in the data.
- An auto.arima function was used on the training set to confirm that the optimal ARIMA model had an AR order of 1 and an integrative (differenced) order of 1. The model can be described as ARIMA(1,1,0) with drift. This **model was selected** to be used for time series analysis.
- The original data had been split into training and test sets with an 80-20 split, which is a standard method of splitting training and test data. This resulted in 585 observations in the training set and 146 observations in the test set. The number of observations in the test set, 146, was used as the **prediction interval** of the forecast.
- This **forecast length was chosen** so that the forecasted revenue values could be compared directly to the test set data to evaluate the performance of the ARIMA model. This forecast length encompasses approximately 5 months, which may be satisfactory, but depending on context it may warrant careful discretion in making conclusions based on the resulting predictions. Though it is entirely possible that nothing out of the ordinary changes for the telecommunications company within those 5 months, this does leave a reasonable amount of time for unexpected circumstances to occur. The forecast length is justifiable for this analysis, but results and conclusions should be approached with caution.
- The **model evaluation procedure** can be summarized as follows:
  - The **ARIMA(1,1,0) model with drift** was selected not only as a result of the auto.arima function, but also after analysis of the ACF and PACF graphs had implied an AR model (implying a p value of 1) and analysis of the stationarity implied a differencing of the data (implying a d value of 1).
  - This model's **AIC** value was compared to the AIC values of select other ARIMA models. Since ARIMA(1,1,0) with drift had the lowest AIC value of all selected models, it was confirmed the **chosen model was correct**.
  - The Root Mean Squared Error **(RMSE)** of the chosen model on the **training** set was **0.4665**. This value is close to 0 and indicates the **model fits the data well**.
  - However, the **RMSE of the test set is more important**, as this measures how well the model forecasts in comparison to the test set. The RMSE of the test set is calculated below.

```
# RMSE of test set
residuals <- test - fc$mean
sqrt(mean(residuals^2))
```

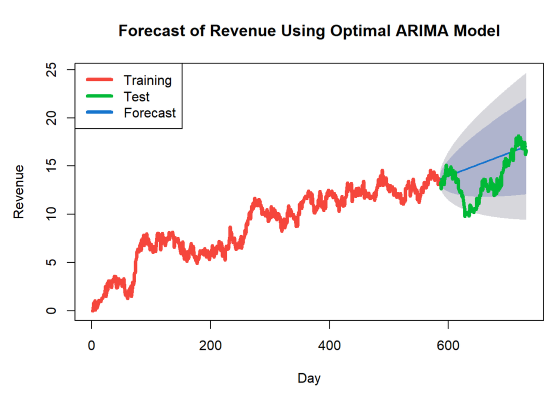
```
## [1] 2.474181
```

  - The RMSE of the test set was found to be **2.4742**, which indicates the model's predicted values were **moderately close to the true values** within the test set, but with significant room for improvement.
  - This can be confirmed in the **visualization** of the plotted forecast and its confidence intervals against the true test set values, which is included in section D3. There are some intervals where the test values coincide with the forecasted values quite well, such as near the beginning and end, but in between these stretches, the test set values dip significantly below the forecasted values, even slightly leaving the 95% confidence interval shortly.
  - For the most part, however, the test set values remain at least within the 80% confidence interval of the forecasted data, which once again implies a **moderate fit** to the test set.

E2. Forecast Visualization

A visualization of the forecasted data against the test set data, along with the training data used to conduct the ARIMA model, is included in section D3. For convenience, it will also be included here.

```
# plots training set, forecast data, forecast confidence interval, and test set
fc <- forecast(AM1, h = 146)
plot(fc, col = 2, lwd = 4, xlab = "Day", ylab = "Revenue", main = "Forecast of Revenue Using Optimal ARIMA Mode
l")
points(test, type = "l", col = 3, lwd = 4)
legend("topleft", legend = c("Training","Test","Forecast"), col = c(2,3,4), lty = 1, lwd = 4)
```



E3. Recommended Course of Action

The chosen ARIMA time series model was **moderately successful** in predicting future revenue values for the telecommunications company. The model likely can be used to make general predictions about future revenue, which may assist in understanding the relationship between customer churn rate and company revenue; however, mindful attention should be paid to the limitations of making short-term predictions with this model. Though the model's forecasts correctly followed the general trend in the test set, there was clear deviation from the forecast in some intervals. **It is recommended to use this ARIMA time series model for general revenue forecasting, but with caution, or to conduct further analysis which may increase the accuracy of this model.**

F. Reporting

This report was summarized in an R Markdown document and has been included as an .html file with the following name: **"Atwood_D213_Task1_RMD.Rmd."**

G. Third-Party Code References

WGU Courseware was used as a resource to learn the methods, concepts, and functions used to create the codes in this project, including DataCamp course tracks (datacamp.com), Dr. William Sewell's presentations, and Dr. Elleh's presentation materials. There are no codes that have been taken directly from any other resources.

H. Content References

There is no content in this analysis that has been quoted, paraphrased, summarized, or otherwise requires direct citation.