Time Series Modeling

Matthew E. Heino

Advanced Data Analytics

Introduction

This paper will explore the concepts of time series modeling. The paper will look at the procedure to make sure data is in the right format to be used for time series analysis. The paper will try to make predictions based on a model that has been based on ARIMA. There will be an evaluation of the success of the model when comes to predicting or forecasting events that will occur in the future.

Background

The data that has been provided is for revenue for a hospital. The data covers two years of operation by the hospital. The total number of observations is 731. This is for daily observations of revenue. The data is composed of two columns. The first column is the day column. This column is just a numeric value that increases as the time goes on. This column will not contain a standard date format, e.g. 2024-9-9. It will need to be changed into a format so time series data can be performed.

The next column contains revenue for the hospital system. The column is in millions of dollars. The data is not given in the full form. For example, if the value was 15,000,000 then it would stored as 15.0. This is important to note when discussing the revenue in future sections of this paper. One additional thing to be aware of is the first entry does not contain any value for revenue. This is essentially a zero value, but it can be used as a starting point for graphing the data. In regards to the trends and other possible patterns that may be apparent in this value will be the first point to be plotted.

Note: The submission will be composed of the following files:

- CSV files (cleaned data, test, and training files)

- This Word document
- A PDF of the executed Jupyter Notebook. This PDF will have the executed instead of the usual Panopto video that has been required in previous assessments.

Some of the information and observations may be found in both the Jupyter Notebook and this document. Please refer to the accompanying PDF of the Jupyter Notebook for some cursory information about the file and what it contains.

Part I: The Research Question

This section will discuss the proposed research question that can be answered with the university-provided research data. There will be a discussion of the goals and objectives that are hoped to be accomplished by the analysis of the time series data.

A1. The Research Question: The question that needs to be answered is based on the data provided. The question is, "Is it possible to predict or forecast the daily revenues for the university hospital accurately and how do these predictions compare to what is observed in the data?" Although this seems like two questions. The two are intertwined. The question needs to be something that can modeled using the appropriate time series methodology. The last half of the question goes to the point that if the model does not work what can we do to make the data work and make successful predictions for the revenue?

A2. The Goals and the Objects of the Analysis: The goals and objectives of this analysis are to be able to use the supplied data to effectively create a model that can predict the future daily revenue for the hospital system. If the model proves to be inadequate or does not provide satisfactory results, is there something that can be done to improve the value of the model either by changing the model methodology or possibly of inclusion of data that may aid in

creating a training set that better models the events that were not adequately covered in the given dataset?

The model that will be used to create the model will be the ARIMA model. This model will help to predict the revenue for a time series data. This model can be used to make predictions based on the data that has been provided.

Part II: Justification of Methodology

This section will give some of the assumptions that need to be present to perform time series analysis successfully. These assumptions will be discussed in a little detail where appropriate.

B. The Assumptions of Time Series Analysis: There are a few assumptions that need to be present for times series analysis to be successful and valid model produced. The assumptions are listed below.

- Time series data must be stationary.
    - The "stationarity" of the data makes it easier to perform statistics that are consistent over the time frame of the given dataset. As stated by Rasheed, "… the statistical properties of a system do not change over time (2011)." This will imply that the behavior of the data should have some degree of constancy to it. Data that is stationary will not have means, variance, and autocorrelation structures that do not change over time of the data (Rasheed, 2011). In essence, stationarity means that the way the data changes is constant. A stationary time series will not exhibit any trends or seasonal patterns. For instance, if we were to look at retail sales data we may notice that at certain times of the year, there are periods where the sales will tend to

be higher. This is usually evident around holidays when customers increase their spending. This would be an example of seasonality. Keeping with the idea of retail, we exhibit a trend when there is a steady uptick in sales of a particular product. This does not mean the trend is going to continue and should be taken into account when creating a model for the sales of products. An example of a product that exhibited exceptional growth in sales would be an item like the Cabbage Patch Kid dolls of the 1980s. While this product was popular in the 1980s, it has since fallen out of favor as the number one toy of the Christmas season. If we were to ignore the factors that this product was a trend item; making a model would only reflect only an upward trend. If this model was implemented now it would be of no use in predicting the sales for the item.

    - When checking the time series data we need to look at how correlated the data is. The concept here is known as autocorrelation. Autocorrelation is a measure of how the items in the data series are correlated. In terms of the meaning the extent the data is correlated with itself at various points in time.

- The data does not contain outliers or other anomalous data.
    - If outliers are included in the data and used to train or test the model it can lead to inaccuracies with the model and its associated predictions. Additionally, with the inclusion of outliers, the model may be seen as unreliable. The inclusion of outliers will often lead to erroneous predictions since the predicted values will be based on values that may force the predicted value to be higher or lower depending on the value of the outlier.
- The data is univariate, reflecting a single variable to be modeled.

- When predicting or forecasting values there is a need to focus on one variable or feature at a time. The variable in this assessment is the revenue for the hospital system. Using more than one variable leads to increased complexity for modeling the data. Focusing on a univariate time series allows for the exploration of a single variable and how this variable evolves.

- Past data points are indicative of the behavior of future data points.
  - Using the model that is proposed, there is a need to be assured that past data points are reliable or indicative of the future behavior that is exhibited by the data. The caveat here is that data points may not be a good measure of future behavior if the chosen group of data is not large enough to account for outliers or anomalous data points that may be included in the data.

Part III: The Data Preparation

In this section, there will be a discussion of how the data was prepared. There will be visualization presented as well as some observations that were observed in the data visualizations There will be a discussion of the time stepping of the data. The data will be evaluated for stationary and the appropriate steps taken to make ensure that the data is stationary by using the appropriate method.

C1. Line Graph of the Time Series Data: This section includes a visualization of the data that has not been "cleaned." The only change that has been made was changing the columns to more appropriate values. The "Day" column was changed to show the dates in a better manner for the graph. The "Day" columns were initially just a series of numbers that corresponded to a sequential observation. The graph is shown below.

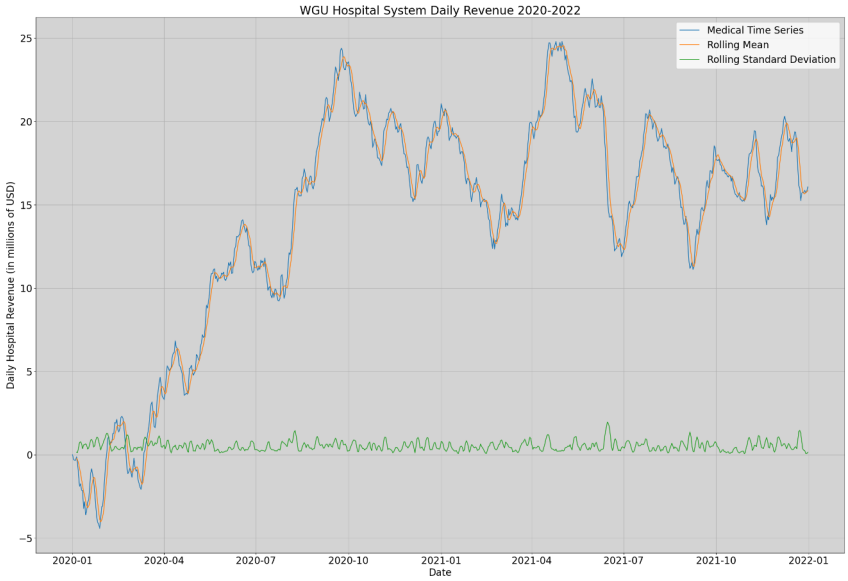


Image 1: Visualization of the data.

The graph above shows a few items that are apparent in the data. The blue line shows the overall progression of the data in the time series. This data is revenue for each of the days that are found in the dataset. The orange line is the rolling mean of the data in the time series data. It is the mean that was taken over the period but with a lag of five days. That is why the orange line starts a little later in the graph. The graphs below show the rolling mean and the rolling standard deviation. Looking at the standard deviation the standard deviation varies over the course of the data's time frame.

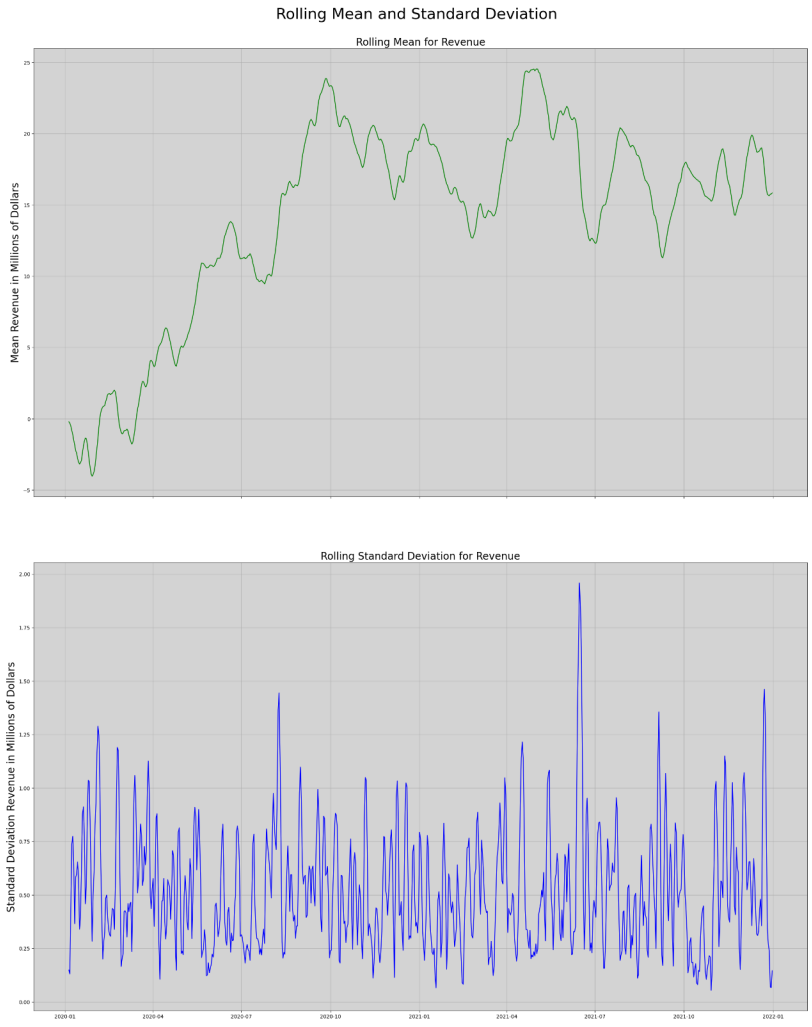


Image 2: Visualization of the rolling mean and rolling standard deviation.

Reviewing the graph there is a trend in the positive direction for the hospital as evidenced by the green dashed trend line. There are a few other observations to be discussed here. There are a few periods where the hospital did not make any money. This is evident in the period before March of 2020. Although there was a period where the revenue was positive, for the period before April of 2020 most of the day's revenue is considered to be negative.

Another period to consider is the time frame starting on or about July 2021. The data does not seem to follow the same trend of increasing revenue. The points after this time frame

seem to fall within 15 million in revenue with a relatively flat trend, The trend upwards is not as pronounced as it has been observed in previous periods.

The code that was used to create this visualization can be found in this section of the Jupyter Notebook. The code will not included here.

C2. Time Step Formatting: The data that was provided time frame is based on days and each observation corresponds to one day's revenue. The data does not seem to have any gaps there are 731 observations as indicated earlier. The "Day" column will initially be used as the index for the data frame. This index will later be converted to a **DateTime** object to better encompass the information that is contained within the column. The current value of the column is just a simple numeric in the range of 731. The first observation in the columns does not have any revenue.

As stated previously the first row has a zero for the column and will eventually be dropped during the "difference" section of the assessment. There is currently no logical reason to keep this data point in the data frame once the main data frame is created.

The data has 731 observations. This corresponds to two years of data. Each of these has the revenue that was observed for the day. The values are meant to reflect the revenue for the hospital. The format of these values is not given in millions but is given in a format that has been discussed in a previous section of the paper.

There are no apparent gaps in the data. A gap would be indicated by a flat line or some other visual peculiarity in the graphs that were shown in Images 1 and 2.

C3. Evaluation of the Stationarity: To proceed with analysis, the data must be evaluated for stationarity. This means that does not exhibit any type of trend, the variance, and

there is constant autocorrelation within the data. This stationarity means that the statistics of the data do not change over the course of the data's period.

To determine if the given dataset is stationary we can look at a simple test to see if the data is truly stationary. This test is referred to as the Dickey-Fuller test. This test will prove a few key values. The one that will be examined here is the p-value. We want a p-value that is below the critical value. The usual critical value is 0.05, and this is the value that will be used to determine if the data is stationary. Upon running the code in section C3, the values that were returned are shown below.

```
ADF Statistic:   -2.218319
p-value:         0.199664
Critical Values:

        1%: -3.439
        5%: -2.866
        10%: -2.569
```

Reviewing the values that were returned it is obvious that the **p-value** for the given data is above the critical value that was stated earlier. This means that we are looking at data that exhibits trends and other phenomena that indicate that the statistics like mean and variance will possibly change over the period that the data was collected. Hence, this data is not stationary.

When deciding on whether data is stationary we can look at a few hypotheses. The first is the null hypothesis in the case of this assessment it was proposed that if the p-value is below the critical value we must accept this hypothesis. In the case of the first run of the Dickey-Fuller test, this hypothesis could not be rejected.

To solve the problem of non-stationary data it is possible to "difference" the data. This is where there will be a computation of the differences in the data between the consecutive observations in the dataset. This will make it possible to stabilize the mean. This concept was

brought up in a previous section and will be revisited here. This stabilization of the mean from one observation will make it less likely to exhibit trends or seasonality (Hyndman & Athanasopoulus, 2023). It is still likely that the data may still exhibit one or both of the phenomena of trends and/or seasonality. This will be shown in an example later in this document.

After "differencing" the data, we see that the p-value is now 0 as illustrated in the output shown below. We are now able to reject the null hypothesis and accept that the data is now stationary.

```
ADF Statistic:    -17.315500
p-value:          0.000000
Critical Values:

        1%: -3.439
        5%: -2.866
        10%: -2.569
```

To show this visually we can look are a graph of the data after the "differencing" was executed on the data.

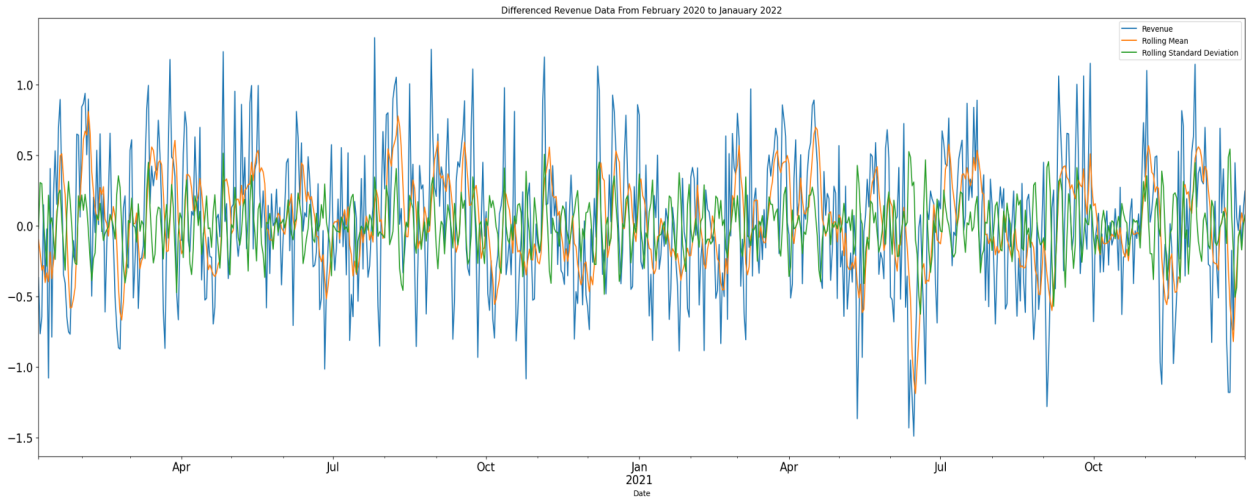


Image 3: Visualization of the "differenced" data.

Looking at the data now you can see that it does not exhibit any type of discernible trends or possible seasonality. This was not what was exhibited in the first visualization, refer to Image 1. There is a pronounced trend, but looking at **Image 3** it would be hard to ascertain if there is any type of seasonality in the data. This concept of seasonality will be further explored in a subsequent section of this document.

C4. Steps to Clean the Data: For the data to be ready for processing and be made into a state that facilitates time series data analysis it must be in a form that is suitable for this type of analysis. In the state of the original data, there were a few things that needed to be accomplished to get the data into a state that is acceptable for analysis.

The first thing is to read the data and look at what the data contains. This was done both in code and through reviewing the data in Excel. This step was to get a feel for what is in the columns of the data file. The data cleaning steps are briefly summarized in the numbered bullets below.

1. Read the data from the CSV file into the pandas data frame.
2. Review the data frame and determine the appropriate index for the newly created frame. In this case, it will be the column that contains the date. This column will be converted into a more suitable form as described in the Jupyter Notebook.
3. Review the data frame for any missing data. There was none as indicated by looking at the data using the **info** method.
4. Create rolling mean and standard deviation fields to be graphed and reviewed later on in the analysis of the frame.

5. Determine if the data is stationary by running the appropriate tests. In the case of this assessment, the Augmented Dickey-Fuller test was used to ascertain the stationarity of the time series data.
    - This testing was executed twice once before the data was differenced and once after.
    - Differencing the data made the time series stationary, but it lost the revenue values that were associated with it.
6. Create the train and test data sets that will be used to create the model and test the model. All code for this can be found in the appropriately commented sections of the notebook. The train and test data were created from the original time series data. The rationale is that it will be used in conjunction with the **auto_arima** function that will discussed in a subsequent section of this document. The train data include all observations except for the last 30-days. This remaining will be used to test model on how accurate it is.

Please refer to the labeled sections for the code that accomplished each of the steps that were listed above.

C5. Copy of the Cleaned Dataset: The cleaned dataset that was created is composed of the differenced data. The cleaned data can be found in the following file:

- **Heino D213 Task Stationary.csv**

This file will be attached to this submission along with other relevant files that are required for the assessment.

Part IV: Model Identification and Analysis

In this section, there will be an analysis of the time series data. This will allow for the presence of a seasonal component to the time series data. A look for trends in the data. A

calculation of the autocorrelation function along with the graphs that visualize this function. A visualization of the spectral density of the time series data. A visualization of the data in a decomposed seasonal state. A discussion if there is or is not a lack of trends in the residuals of the time series data.

D1. Annotated Findings: In this section, there will be visualizations that will include a visualization of the data and the possible presence of seasonality in the data. Visualizations that will show if any discernible trends are inherent in the data. A plot of the spectral density of the data. A visualization of the decomposed data. A visual that will look to confirm that there are no trends in the residuals of the data.

The Seasonal Component. In this section you will find an image that shows the data and if it depicts any semblance of seasonality. This image is shown below.

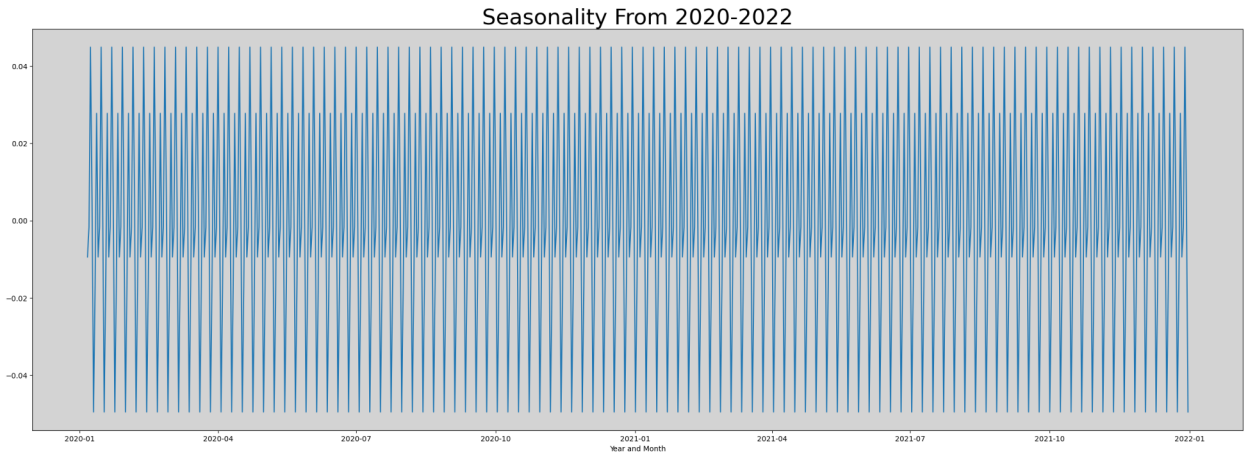


Image 4: Visualization of seasonal data.

Looking at the image above we can see that there might be a notion of seasonality in the data. There is a need to slice a time series section from the data to see if there is a seasonality component to the data. Taking a time slice from January 31, 2020, and March 30, 2020, there does seem to be some seasonality to the data. Please reference the image below for an illustration of the seasonality.

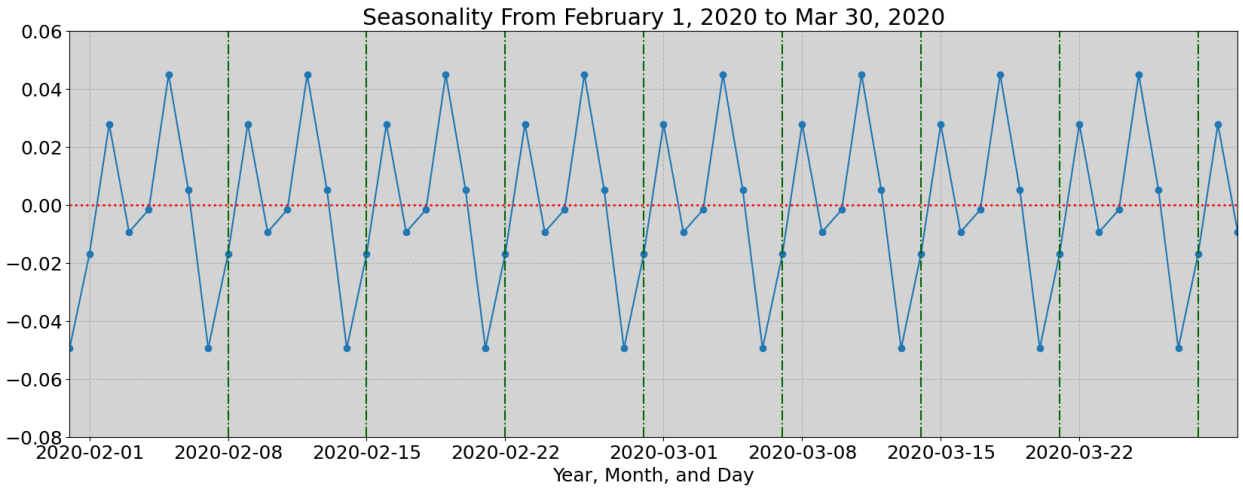


Image 5: Visualization of a subset of seasonal data.

Looking at the graph there is a component of seasonality to the data. If you look at the intervals between the green dashed lines you can see that the data does have a "cycle: that seems to repeat itself. The seasonality was not reduced when the data was differenced in a previous section of the assessment and analysis.

Check for Trends. The graph that is shown in this section is an attempt to picture if the data exhibits any trends. The graph is shown below.

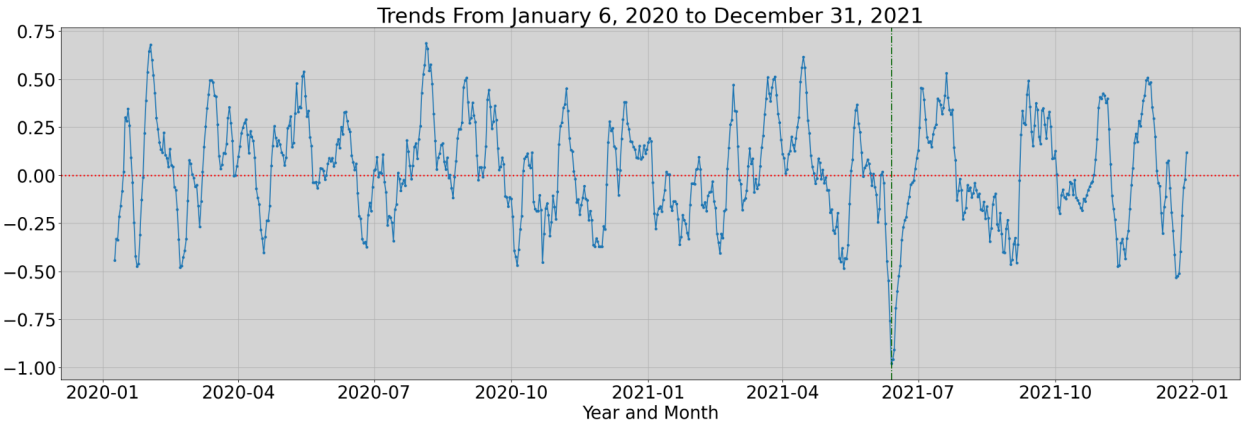


Image 6: Visualization of a subset of seasonal data.

In the graph above there does not seem to be any noticeable trends in the data. There is one interesting point in the series. This is denoted by the green dashed line. There seems to be an

anomaly in the data. The point here seems to be far outside the other observations of the data that are included in the time series.

The Correlational Function. This section will deal with visualizing the autocorrelation there will be a few attempts to adequately model the function visually. The first attempt will be to use the **plot_acf** and the **plot_pacf** functions. The graphs of these functions are shown below.

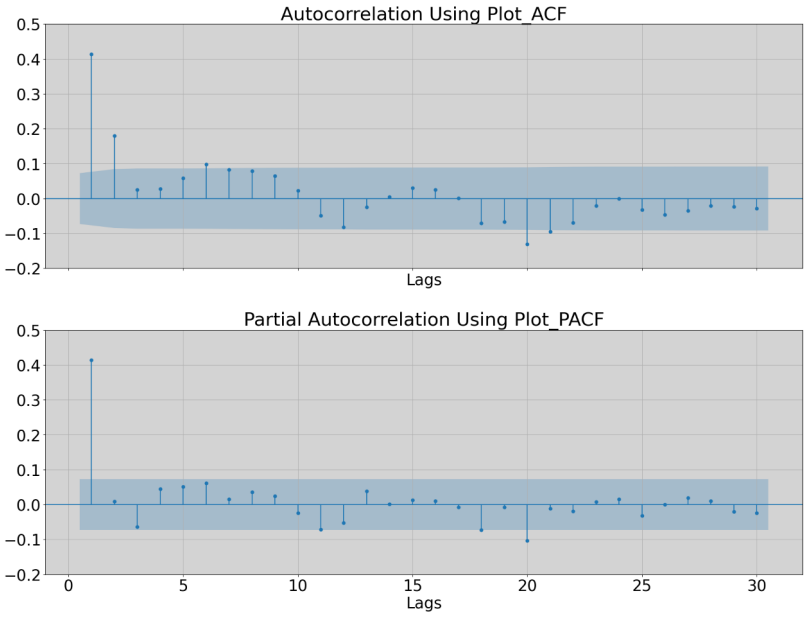


Image 7: Visualization of ACF and PACF of the data.

In the graph above, there is only one point of interest that stands out from the rest which is lag which is number 1 (as numbered on the graph above). This one falls outside the confidence interval that the other points seem to fall in or very close to the interval.

The plot below shows these values in a side-by-side comparison the ACF shows the correlation of the time series with itself, while the PACF shows the time series correlation of the time series after the effects of the previous lags have been removed (Ahmed, 2023).

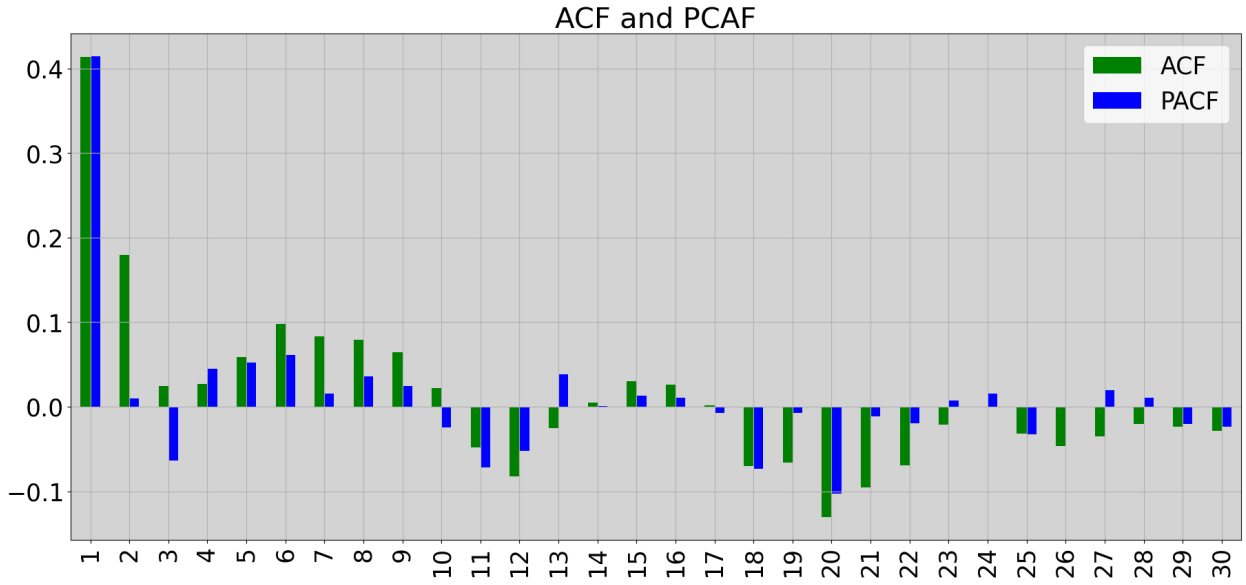
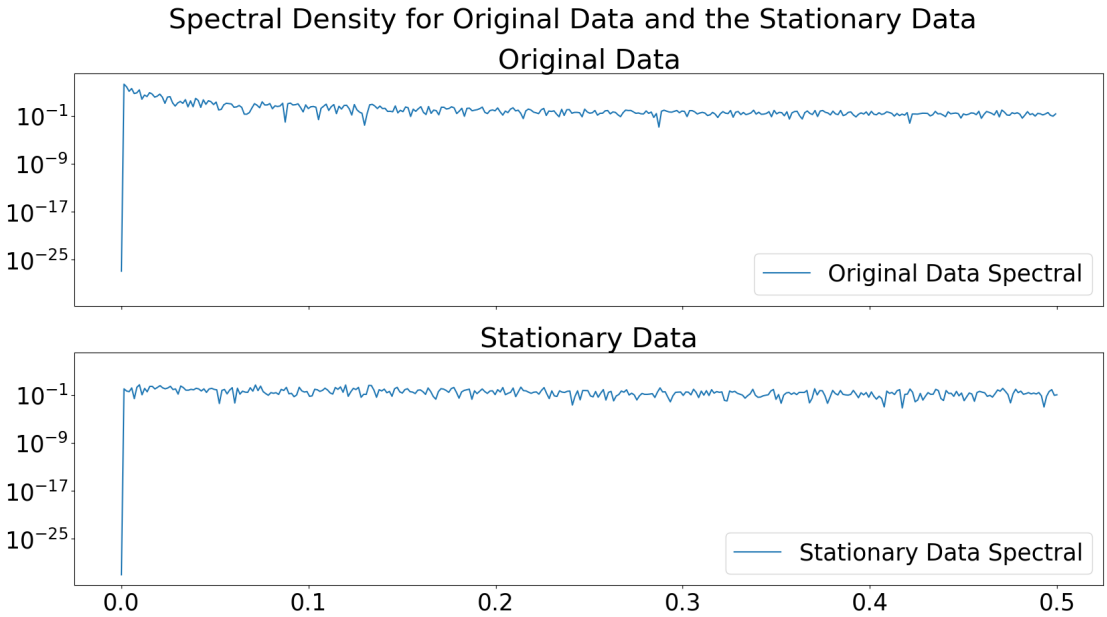


Image 8: Visualization of ACF and PACF together.

This graph was created using the acf and pacf functions in the statsmodels library. This was to show how these values compare to each other.

The code for these visuals can be found in the Jupyter Notebook in this section.

Spectral Density Plots. This section contains the spectral density of the time series data. The graphs are shown below.



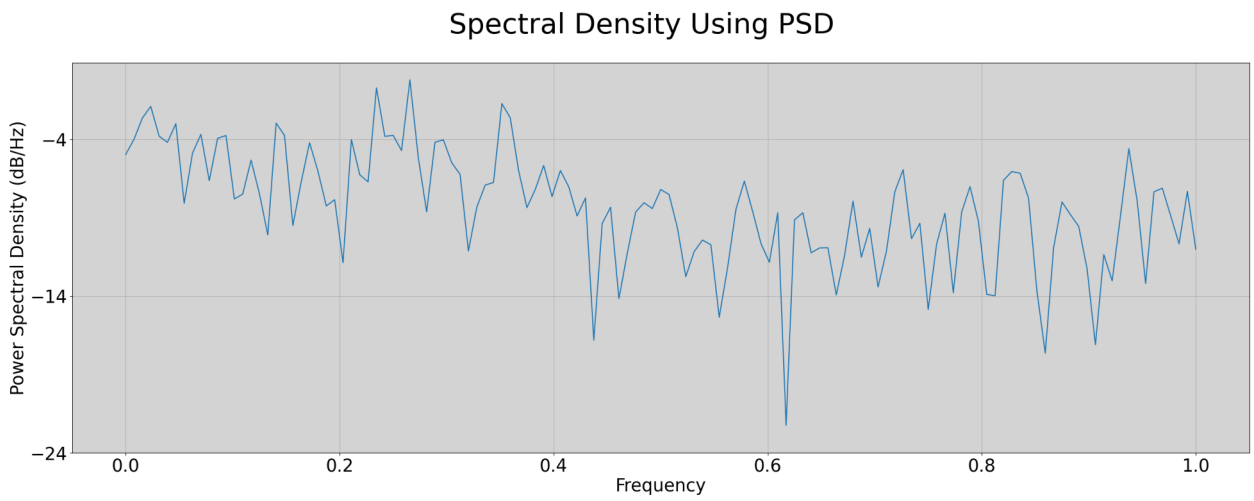


Image 9: Visualization of Spectral Density of the time series data.

These graphs were created using a few different functions that are available in Python's libraries. To see the methods used please refer to the Jupyter Notebook. Reviewing these graphs there is no indication of a seasonal component or a trend based on these graphs. This is contrary to what was stated in the previous section (The Seasonal Component).

Decomposed Series. This section there is a visualization of the decomposed data. The visualization is composed of three elements:

- Graph of the differenced data.
- A trend graph of the data.
- A graph of the seasonality of the data.
- A graph of the residuals.

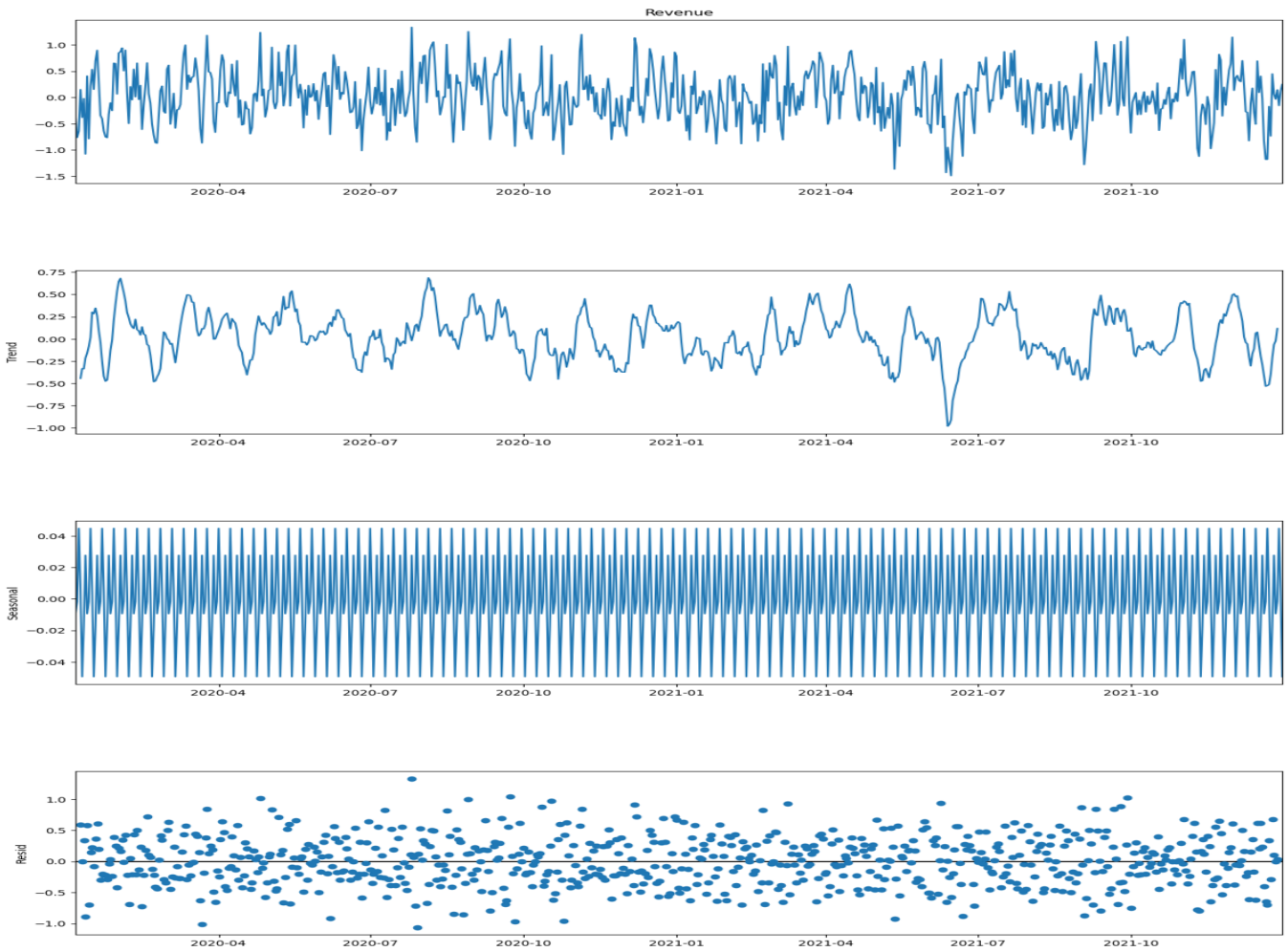


Image 10: Visualization of decomposed time series data.

Referring to the trend graph, the outlier point shows that was discussed in, Check for Trends and Image 6, shows up in this trend graph.

Lack of trends in the Residuals. The graph below illustrates the trend in the residuals. Reviewing the graph there does not seem to be a noticeable trend in the residuals. The residuals seem to fit relatively close to the 0 y-axis with only a few data points that may be considered outside the region of the other residual data points.

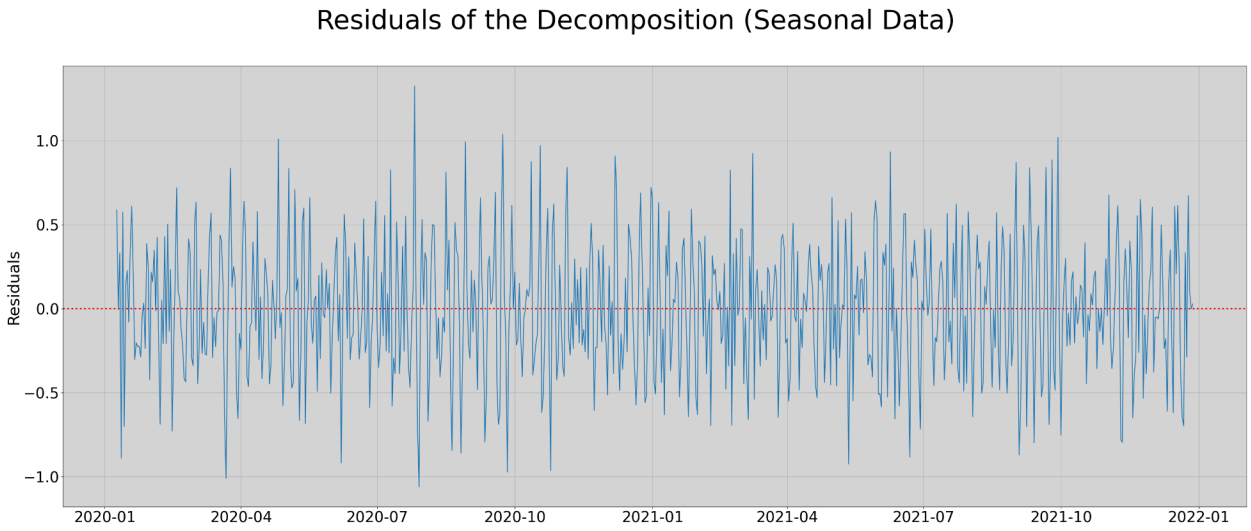


Image 10: Visualization of decomposed time series data.

All code for these graphs can be found in the accompanying Jupyter Notebook.

D2. Identification of the ARIMA model: To develop an accurate with the given data there will be an automated process employed to get the most accurate model. This process will use the **auto_arima** method from the **pmdarima** library. The **auto_arima** method will try different combinations of values for the **p**, **d**, and **q**. These values correspond to the order of the autoregressive model, the d is the order of the differencing, and the q is the order of the moving average component in the model.

The output from the **auto_arima** function is shown below. The model parameters that were chosen are the following:

- 1 – the order of the autoregressive component
- 1 – the degree of differencing
- 0 – order of the moving average component of the model

To interpret these values more practically, the first number indicates the degree of the AR component of the model. In this case, it is a 1st order AR model. The second value is the degree of the differencing of the model. There will be one execution of differencing undertaken with the

time series data. This is why the data was split using the original time series data and not any other forms of the data. The last value is the MA component of the model. There is no inclusion of this component in the model. The model will not utilize a moving average in the creation of the model.

This screenshot shows the execution of the **auto_arima** function.

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=847.047, Time=0.53 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=972.824, Time=0.10 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=845.079, Time=0.12 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=867.928, Time=0.19 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=972.766, Time=0.03 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=847.071, Time=0.18 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=847.073, Time=0.15 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=847.066, Time=0.58 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=843.938, Time=0.06 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=845.924, Time=0.06 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=845.927, Time=0.08 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=867.209, Time=0.06 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=845.897, Time=0.24 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0]
Total fit time: 2.386 seconds
```

The screenshot below shows the output of the summary function.

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	701
Model:	SARIMAX(1, 1, 0)	**Log Likelihood**	-419.969
Date:	Fri, 02 Feb 2024	**AIC**	843.938
Time:	02:11:13	**BIC**	853.040
Sample:	0	**HQIC**	847.456
	- 701		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4143	0.035	11.863	0.000	0.346	0.483
sigma2	0.1943	0.011	17.535	0.000	0.173	0.216

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1.68
Prob(Q):	0.94	**Prob(JB):**	0.43
Heteroskedasticity (H):	0.99	**Skew:**	-0.02
Prob(H) (two-sided):	0.91	**Kurtosis:**	2.76

Using the values from this summary it is possible to create an equation for the model. The equation for the model is:

- $X_t = 0.4143\ X_{t-1} + 0.1943 + \varepsilon_t$

The model is then created using the standard ARIMA function. The output shown below is the summary of the model created using this method.

SARIMAX Results

Dep. Variable:	Revenue	**No. Observations:**	731
Model:	ARIMA(1, 1, 0)	**Log Likelihood**	-437.991
Date:	Fri, 02 Feb 2024	**AIC**	879.982
Time:	02:11:13	**BIC**	889.168
Sample:	01-01-2020	**HQIC**	883.526
	- 12-31-2021		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4142	0.034	12.258	0.000	0.348	0.480
sigma2	0.1943	0.011	17.842	0.000	0.173	0.216

Ljung-Box (L1) (Q):	0.02	**Jarque-Bera (JB):**	1.92
Prob(Q):	0.90	**Prob(JB):**	0.38
Heteroskedasticity (H):	1.00	**Skew:**	-0.02
Prob(H) (two-sided):	0.97	**Kurtosis:**	2.75

The values in this output are very close to the values that were given by the **auto_arima** function used earlier. The code for this and other outputs for this section can be found in the appropriate section of the Jupyter Notebook.

D3. Forecast Using the ARIMA model: In this section, we will look at the ability of the model to predict both future daily revenue as well how well it can predict revenue values that we have the data for. This model will make predictions 30 days into the future as well as make predictions and see how well these values match with the known values.

Making a 30-day prediction. Using the **forecast** function we can predict for the next 30 days that were not included in the supplied time series data. The output is shown below.

```
Forecast:

 2022-01-01    16.171559
2022-01-02    16.213862
2022-01-03    16.231384
2022-01-04    16.238642
2022-01-05    16.241649
2022-01-06    16.242894
2022-01-07    16.243410
2022-01-08    16.243623
2022-01-09    16.243712
2022-01-10    16.243749
2022-01-11    16.243764
2022-01-12    16.243770
2022-01-13    16.243773
2022-01-14    16.243774
2022-01-15    16.243774
2022-01-16    16.243774
2022-01-17    16.243774
2022-01-18    16.243774
2022-01-19    16.243774
2022-01-20    16.243774
2022-01-21    16.243774
2022-01-22    16.243774
2022-01-23    16.243774
2022-01-24    16.243774
2022-01-25    16.243774
2022-01-26    16.243774
2022-01-27    16.243774
2022-01-28    16.243774
2022-01-29    16.243774
2022-01-30    16.243774
Freq: D, Name: predicted_mean, dtype: float64
```

Screenshot of the 30-day Forecast out of sample.

This shows that revenue for the next thirty days will be around $16.24 million. This will correspond to the data that we have in the time series data. The last recorded day was December 31st, 2021.

D4. Calculations: All calculations that were done were done by the library functions that were used to create the code in the previous sections. There were no hand calculations done during this assessment.

D5. Code: All the code that was used to create this section can be found in the relevant section of the Jupyter Notebook. Please refer to the notebook for the code and the associated output of the code.

Part V. Data Summary and Implications

E1. Results of the model. The model will attempt to create forecasts 30 days into the future and will try to predict the last 30 days of available time series data. This seems to be a reasonable time frame. Since the supplied data does exhibit a slight change in the growth of the revenue during this time frame. Looking any further back may not be practical. As the time frame extends either into the past or into the future the model is more than likely to become less accurate.

The next objective is to look at the model and how it did with predicting versus the test data that was created earlier in the assessment. The time frame will be the last thirty days. This gives a good account of monthly activity. The longer the time frame the more inaccurate the model could become based on the possible inclusion of a seasonal component that was not adjusted for in the ARIMA model.

This seasonality could be adjusted for setting the relevant parameters of the function. The **seasonal_order** argument for the ARIMA method. This course of action was not undertaken but may be explored in the future.

The model performed well with the parameters supplied by the **auto_arima** function. There were a few ways to look at how the model performed. The first was to use the **plot_diagnostics** function. The image below shows the results of executing this method.

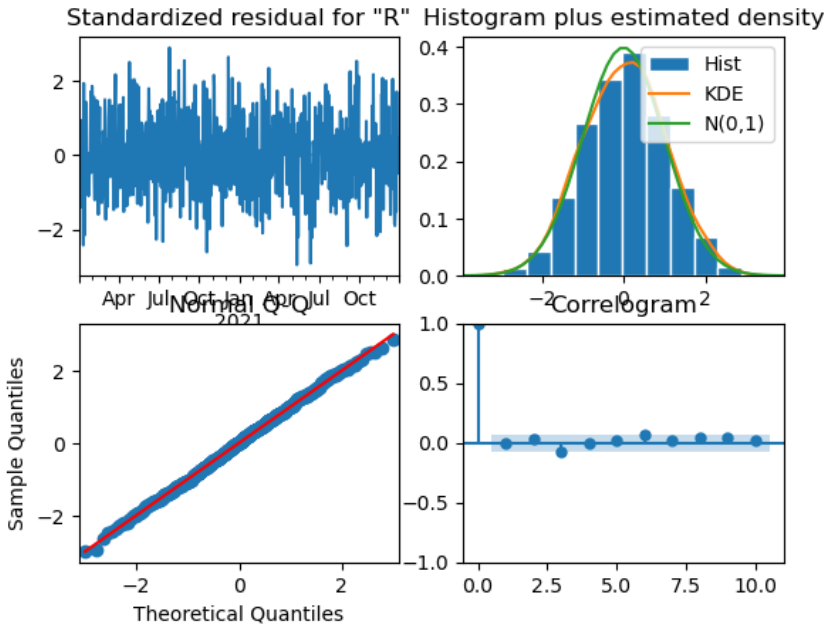


Image 11: Visualization of the plot diagnostics.

Looking at the diagnostic plot the plot in the upper left is the plot of the one-step ahead of the residuals of the model. If the model is working correctly there should be no pattern in the residuals. Based on this plot there are no obvious patterns in the residuals (Team & Team, 2023).

In the top right plot, we are looking for the measured and a "smoothed" version of the histogram, while the green line shows a normal distribution. We want the two lines to be the same with only an occasional deviation. Based on the plot there seems to be very close in their values. There are only a few deviations between the two lines on the graph (Team & Team, 2023).

The bottom left graph is the normal Q-Q plot. This plot looks at the distribution of the residuals to the normal distribution. We want to see that the residuals are along the red line. Reviewing this graph, it seems that all the residuals reside along the red line indicating that the model's distribution of residuals is normal (Team & Team, 2023).

The last plot is the correlogram. This is an ACF plot of the residuals and we want to see that these points fall within the 95% interval as indicated by the shaded area of the graph. Looking at this graph it is apparent that the residuals all fall within the shaded interval.

One more metric is to look at the root mean square (RMSE). This will tell how close the predictions and the test data are. The result of the calculation is shown below.

```
Value of the RMSE:  0.21968551113004772
```

(The code that was used to create this calculation can be found in the appropriate section of the Jupyter. It is found in cells 44 and 45.)

This means that the data is relatively accurate in predicting the revenue for a given date. To interpret this value means that the predicted revenue for a date will only be off by approximately $219,000. How close the predicted values are to the observed will be better illustrated in the next section of this document.

E2. Visualization of the Model: The model was used to predict the last 30 days of the available data. This was to show how well the model is at predicting value relative to what was observed. The graph shows the observed data for the last 30 observations along with the 30 days worth of predictions.

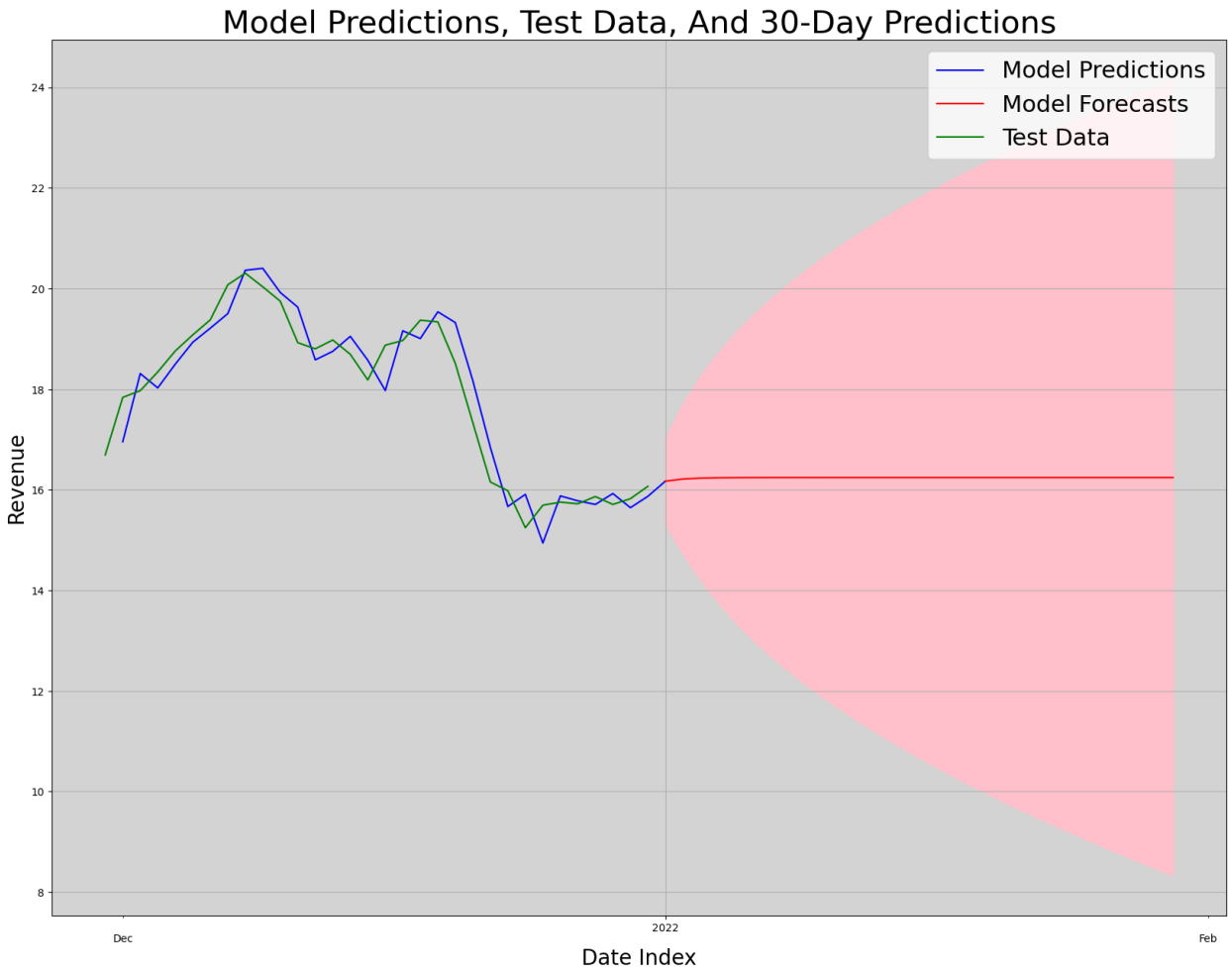


Image 12: Visualization of the Test Data, Model Predictions, and 30 day Out of Sample forecast.

The model's predictions seem to track well with what was observed. It was able to find the data points well. There is some difference in the data, but that difference does not seem to be too severe. No model will be perfect in predicting the values there is always some room for error between the prediction and the actual observed value.

E3. Course of Action: Based on the results of the model and the metrics used to evaluate the model. The model can be used to forecast the revenue in the future. Although it might be prudent to look at future predictions in a very small time frame. Looking at trends in the previous

data there are periods where the revenue has either increased dramatically or has fallen off dramatically. It might be beneficial to look at what caused such drastic changes in the revenue stream of the hospital and see if these trends are anomalies or if is it something that occurs in a more consistent time frame that may not be apparent in the given dataset.

Part VI: Reporting

F. This submission will include a PDF version of the executed code. This will accompany the other files that are required for the assessment.

References

G. Web Sources

This section will contain all the web resources that were used to create the code for this Jupyter Notebook. Any in-text citations will be found in section H.

matplotlib.dates — Matplotlib 3.8.2 documentation. (n.d.). Matplotlib. Retrieved January 22, 2024, from https://matplotlib.org/stable/api/dates_api.html

matplotlib.pyplot.psd — Matplotlib 3.8.2 documentation. (n.d.). Matplotlib. Retrieved January 24, 2024, from https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.psd.html

matplotlib.pyplot.semilogy — Matplotlib 3.8.2 documentation. (n.d.). Matplotlib.https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.semilogy.html

pandas.to_timedelta — pandas 2.2.0 documentation. (n.d.). Retrieved January 22, 2024, from https://pandas.pydata.org/docs/reference/api/pandas.to_timedelta.html

SciPy.Signal.Periodogram — SciPY V1.12.0 Manual. (n.d.). Retrieved January 23, 2024, from https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.periodogram.html

statsmodels.regression.linear_model.OLSResults.get_prediction – statsmodels 0.15.0 (+200). (n.d.). Retrieved January 24, 2024, from https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLSResults.get_prediction.html

statsmodels.tsa.stattools.acf – statsmodels 0.14.1. (n.d.). https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.acf.html

statsmodels.tsa.stattools.pacf – statsmodels 0.15.0 (+200). (n.d.). Retrieved January 23, 2024, from https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.pacf.html

H. In-text citations.

Any in-text citations will be found in this section. This section does not include references to code. These citations are found in section G.

Ahmed, I. (2023, May 31). What are ACF and PACF Plots in Time Series Analysis? *Medium*. Retrieved February 2, 2024, from https://ilyasbinsalih.medium.com/what-are-acf-and-pacf-plots-in-time-series-analysis-cb586b119c5d

Brownlee, J. (2023, November 18). *How to create an ARIMA model for time series forecasting in Python*. MachineLearningMastery.com. Retrieved January 24, 2024, from https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Hyndman, R., & Athanasopoulus, G. (2023). *Forecasting: Principles and Practice* (2nd ed.) [Book]. Otexts. https://otexts.com/fpp2/stationarity.html

Pulagam, S. (2021, December 14). Time Series forecasting using Auto ARIMA in python - Towards Data Science. *Medium*. Retrieved January 24, 2024, from https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd

Rasheed, R. (2011, July 11). *Why Does Stationarity Matter in Time Series Analysis?* Medium. Retrieved January 25, 2024, from https://towardsdatascience.com/why-does-stationarity-matter-in-time-series-analysis-e2fb7be74454

Team, T. A., & Team, T. A. (2023, January 6). Time Series Forecasting with ARIMA Models In Python [Part 2]. *Towards AI*. Retrieved February 2, 2024, from https://towardsai.net/p/l/time-series-forecasting-with-arima-models-in-python-part-2

What's the acceptable value of Root Mean Square Error (RMSE), Sum of Squares due to error (SSE) and Adjusted R-square? | ResearchGate. (n.d.). ResearchGate. Retrieved February 2, 2024, from https://www.researchgate.net/post/Whats-the-acceptable-value-of-Root-Mean-Square-Error-RMSE-Sum-of-Squares-due-to-error-SSE-and-Adjusted-R-square#:~:text=Based%20on%20a%20rule%20of,more%20is%20acceptable%20as%20well.