Performance Assessment 1 – Revision 1

D213 – Advanced Data Analytics

Jessica Hosey

MSDA, College of Information Technology

Western Governors University

January 4th, 2025

Part I: Research Question

A.1. Summarize one research question relevant to a real-world organizational situation captured in the selected data set and that you will answer using time series modeling techniques.

Using time series techniques, can we forecast the revenue for the WGU hospital in the next quarter (three months)?

A.2. Define the objectives or goals of the data analysis. Ensure your objectives or goals are reasonable within the scenario's scope and represented in the available data.

This project aims to provide a model of the forecasted revenue of the next quarter (90 days) for the WGU hospital using the time series modeling technique ARIMA. The ARIMA model will be used on a training dataset before forecasting on a 20% test dataset of the WGU hospital data provided.

Part II: Method Justification

B. Summarize the assumptions of a time series model, including stationarity and autocorrelated data.

Assumptions of a times series model are as follows:

- No trends or seasonality are found in the dataset
- There are no outliers in the dataset
- There is only one variable being modeled and tested (univariate)
- Future data is forecasted based on prior data behavior.

Seasonality is when a repeating pattern is seen within the data. Repeated number of sales seen year after year during the same month would mean the data has seasonality or a trend. The data must show no trends or seasonality to complete ARIMA time series modeling. The data must be differenced or shifted to eliminate the trends before moving forward with ARIMA modeling with stationary data (Sewell 2024).

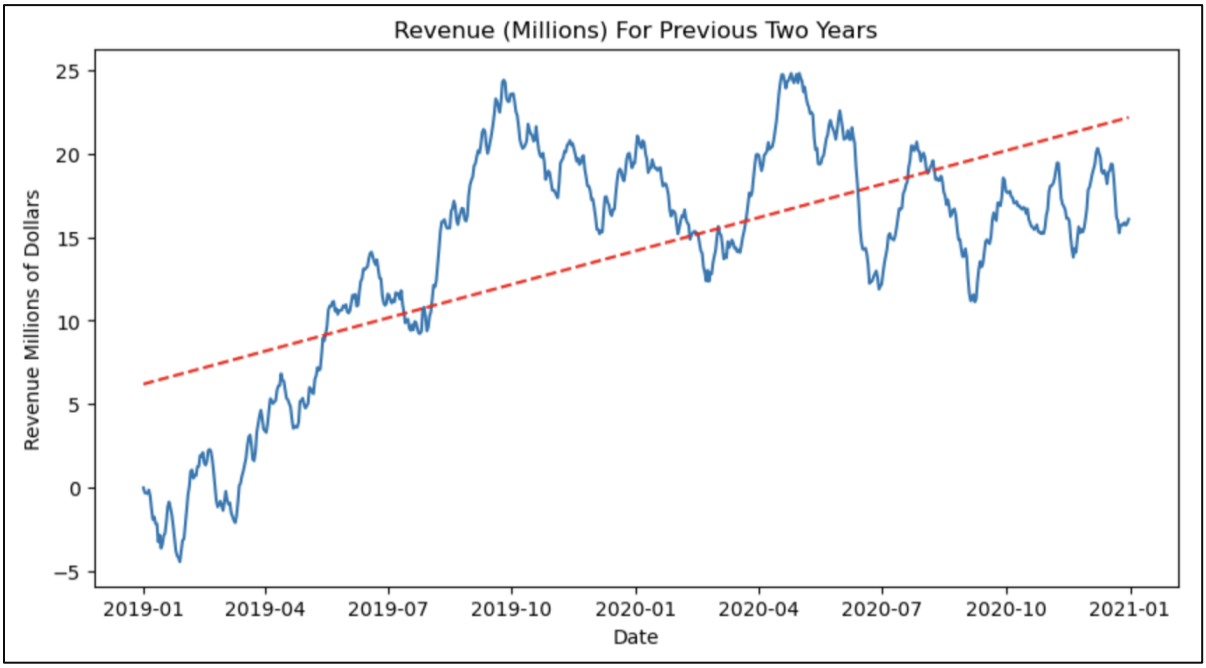
No outliers must be in the dataset, messing with the mean and standard deviation of the data. No points should be found very high or low from the general area where the other data points are located. If outliers are present, it will lower the moving average and decrease our ability to forecast the data correctly.

Univariate testing is being completed, which is another assumption. This goes for any good research project, as testing more than one variable could lead to incorrect forecasts on the dataset. One variable can only be forecasted at a time. For this project, we forecast the revenue for the next quarter (90 days).

The only way to forecast a company's earnings is to see the general behavior of previous months/years of revenue. If the company is decreasing its revenue yearly, then the forecast would indicate a downward trend, which would be the opposite for a company with yearly gains in revenue.

Part III: Data Preparation

C.1. Provide a line graph visualizing the realization of the time series.



C.2. Describe the time step formatting of the realization, including any gaps in measurement and the sequence length.

The dataset had only two columns: day and revenue. The day column was numbered 1 – 731 to indicate the days for two years of WGU hospital revenue. The revenue column was how much money, in millions, the hospital gained or lost each day of operation. The date column was a significant issue as it does not provide specifics about the actual dates of a year, for example, 02-29-2022. Therefore, we changed the day column to give an exact date for that day starting with 2019-01-01. Since there are 731 days in this dataset, one of the two years is a leap year with one more additional day. A leap year occurs every four years, so starting on 2019-01-01, this would include a leap year (2020) in the dataset. To change this issue, we added a column for the date using .to_datetime, and input the start date 2019-01-01. This added a new column for the date, so we needed to remove the Day column and reorganize the order of columns. As seen in the images below, we used .drop and .set_index to restructure this dataset.

	Day	Revenue
0	1	0.000000
1	2	-0.292356
2	3	-0.327772
3	4	-0.339987
4	5	-0.124888

	Day	Revenue	Date
0	1	0.000000	2019-01-01
1	2	-0.292356	2019-01-02
2	3	-0.327772	2019-01-03
3	4	-0.339987	2019-01-04
4	5	-0.124888	2019-01-05

Date	Revenue
2019-01-02	-0.292356
2019-01-03	-0.035416
2019-01-04	-0.012215
2019-01-05	0.215100
2019-01-06	-0.366702

C.3. Evaluate the stationarity of the time series.

Stationarity means that the data will not have a trend or seasonality. From the screenshot in C1, the data is not stationary, as the trend line (red dashed line) has an upward trend. The dataset needs to be differenced or shifted to become stationary. In addition, the Augmented Dickey-Fuller (ADF) test was used to evaluate the stationarity of the dataset. The results of the first ADF test indicate that the p-value is 0.199, which is significant and tells us that the data is not stationary.

```
#seeing initial stats prior to differencing
result = adfuller(df['Revenue'])
print("Test Statistics: ", result[0])
print("p-value: ", result[1])
print("Critical Values: ",result[4])
```

```
✔              Last executed at 2024-12-20 14:34:11 in 40ms
Test Statistics:  -2.2183190476089467
p-value:  0.19966400615064306
Critical Values:  {'1%': -3.4393520240470554, '5%': -2.8655128165959236, '10%': -2.5688855736949163}
```

Augmented Dicky Fuller results before differencing.

C.4. Explain the steps you used to prepare the data for analysis, including the training and test set split.

Four significant steps must be taken before fitting the model and analyzing the results. Firstly, the 'day' column must be changed to the actual date. See C2 above for the code used. Secondly, the Date column then needs to become the index of the dataset. Again, see part C2 above for the code that transforms the date column. Thirdly, the Revenue data is not stationary, and we need to remove the trends/seasonality from the data before making the ARIMA model. The steps taken will be explained below. Fourthly, the data must be split into training and testing datasets before fitting them to an ARIMA model and using them to help answer the research question. The steps taken will be explained below.

Our third step in preparing the dataset was eliminating stationarity in the Revenue data. We must decide whether to make a difference or shift the data to do this. The augmented Dicky Fuller (ADF) test indicated the p-value was 0.19, which is significant, as the dataset is not stationary.

```
#seeing initial stats prior to differencing
result = adfuller(df['Revenue'])
print("Test Statistics: ", result[0])
print("p-value: ", result[1])
print("Critical Values: ",result[4])
```

```
✔              Last executed at 2024-12-24 08:53:04 in 140ms
Test Statistics:  -2.2183190476089467
p-value:  0.19966400615064306
Critical Values:  {'1%': -3.4393520240470554, '5%': -2.8655128165959236, '10%': -2.5688855736949163}
```

I chose to difference the data, using `.diff().dropna(),` and printed another ADF test.

```
Test Statistics:  -17.374772303557066
p-value:  5.113206978840171e-30
Critical Values:  {'1%': -3.4393520240470554, '5%': -2.8655128165959236, '10%': -2.5688855736949163}
Reject null hypothesis, this time series data is stationary.
```

Results returned with 0.00000000000000000000000000005 p-values, indicating that it is well below the 0.05 threshold and the data is now stationary.

The fourth step in preparing the dataset was to separate the dataset into training and testing files to be used when forecasting the next quarter of revenue for the WGU hospital. An 80-20 split was used for this

project (see the code used below in the screenshot).

```
#Split data into a training set and a test set
train, test = train_test_split(df_stationary, test_size=0.2, shuffle=False, random_state=369)
train
```

✔ Last executed at 2024-12-24 08:53:18 in 20ms

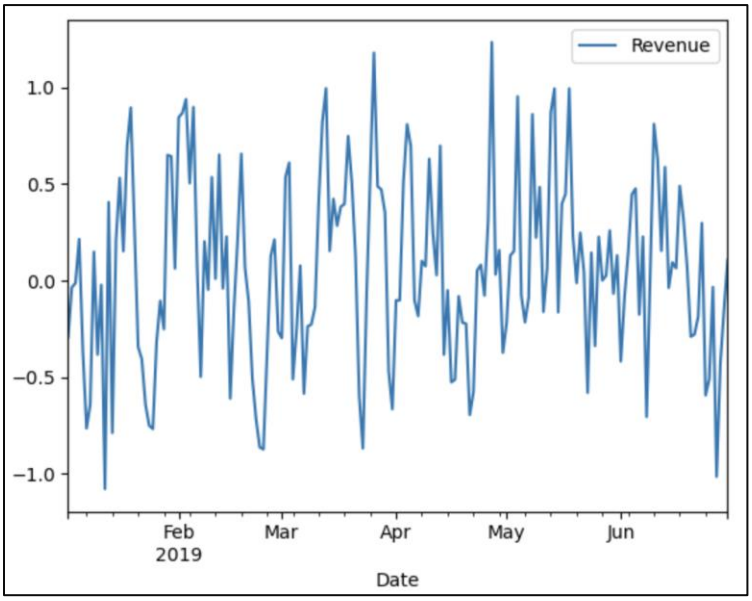
C.5. Provide a copy of the cleaned data set.

See the files attached in the task submission.

Part IV: Model Identification and Analysis

D.1. Report the annotated findings with visualizations of your data analysis, including the following elements:

- **the presence or lack of a seasonal component**

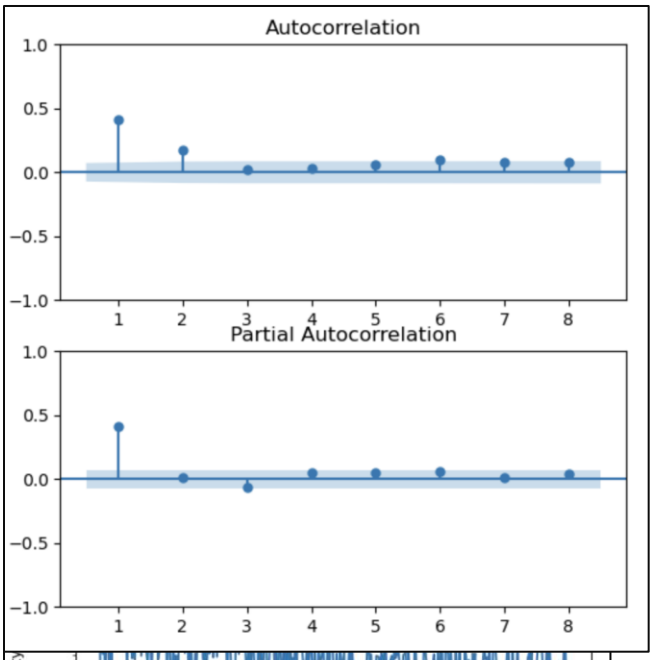


The revenue displays fluctuations over the six months. Peaks and troughs occur at regular intervals, suggesting a recurring seasonal pattern. However, there are no specific trends that are visually seen.

- **trends**

There are extreme highs and lows in revenue during specific months of revenue. These changes indicate that there are significant external factors that dictate these fluctuations. High variation is seen at the beginning and middle of the year, suggesting some revenue instability.

- **the autocorrelation function**

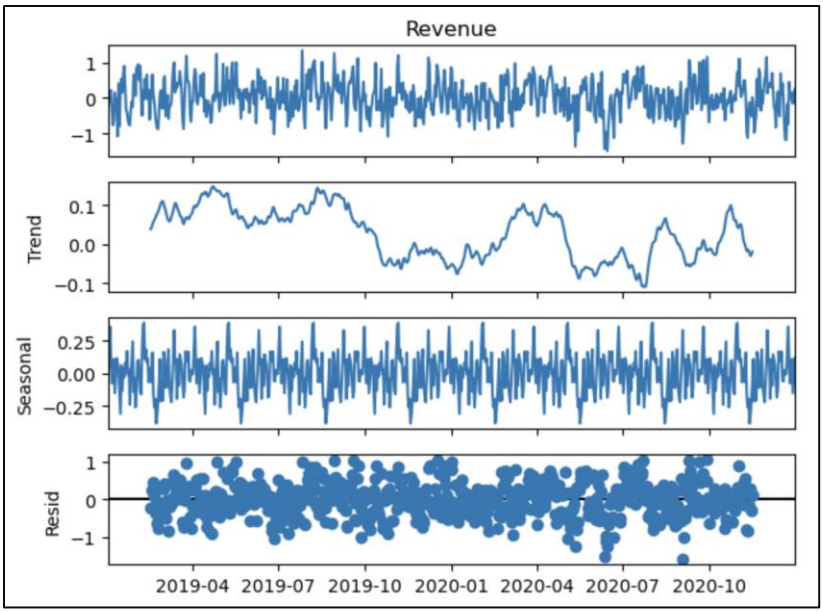


The lag 1 spike indicates a strong positive autocorrelation in the data at this lag. Subsequent lags demonstrate reduced autocorrelation values within the confidence interval, suggesting no significant correlation at these lags. These results indicate that the time series likely follows an AR(1) process, where only the immediate past value significantly influences the current values.

- **the spectral density**

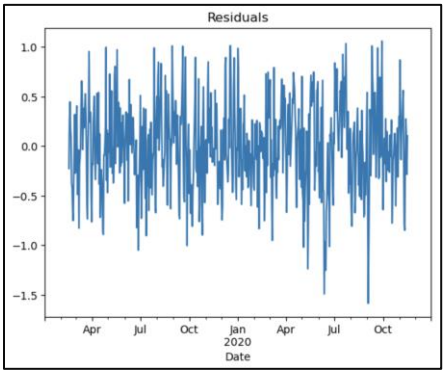
The spectral density graph shows the power changes across different components in the revenue time series. At lower frequencies, the spectral density has higher values, indicating a stronger long-term trend or cycles in the data. As the frequency increases, it becomes more evenly distributed, indicating the presence of noise or less high-frequency components. Peaks at specific frequencies may correspond to patterns in the revenue. The lack of sharp, distinct peaks suggests that the revenue signal may have multiple overlapping components rather than one specific frequency. The drop-off at the lowest frequencies indicates strong periodic behavior that could be linked to seasonal trends or external events affecting revenue. Flattening at higher frequencies indicates random noise or minor short-term changes in revenue. The concentration at lower frequencies solidifies the importance of using long-term patterns for forecasting and strategy. Identifying specific frequencies corresponding to high spectral density could help find key cyclical drivers influencing revenue fluctuations.

- **the decomposed time series**



The raw time series data in the original series shows fluctuations in the data; however, there is no clear upward or downward trend. The trend component indicates long-term volatility in the data. There are some periods of increases and decreases in revenue, with significant peaks near mid 2019 and again in 2020. The seasonal component demonstrates repeating patterns with consistent intervals, indicating seasonality in the data. The magnitude of the seasonality is constant over time. The residual component demonstrates noise and random variation in the data once the trend and seasonality are removed. The residuals are scattered with no apparent trend or pattern. This indicates that most structures in the data are based on trends and seasonality.

- **confirmation of the lack of trends in the residuals of the decomposed series**



Again, the residuals indicate no specific upward or downward trends in the data. It also confirms that decomposition successfully removed the trend from the dataset. In addition, the seasonality was also removed from the dataset with decomposition, and the residuals appeared to be random noise. This validates the decomposition process as successfully isolating components from the original dataset.

D.2. Identify an autoregressive integrated moving average (ARIMA) model that accounts for the observed trend and seasonality of the time series data.

We used auto ARIMA (see screenshot below for code and output) to find the best model for the dataset. This tries fitting multiple ARIMA models, gives us a printout of all possible versions, and searches for the model with the lowest AIC value. For this dataset, auto ARIMA found the model with a p = 1, d = 0, and q = 0 to have the lowest AIC of 879.982. This model had a low p-value of 0.00; therefore, the first lag has a strong relationship with time. The Ljung-Box test indicates the autocorrelation between the residuals and noise. A 0.02 Ljung-Box test result shows that the residuals of the ARIMA model have no significant autocorrelation and are due to white noise. Autocorrelation also means that the ARIMA model is not a perfect fit for our time series. Because of that, the model may miss some relationships in the dataset. However, the chosen model (1, 0, 0) is suitable for forecasting.

```
#ARIMA model using auto_arima
from pmdarima import auto_arima
stepwise_fit=auto_arima(df_stationary['Revenue'], trace=True, suppress_warnings=True)
stepwise_fit.summary()
```

✔ Last executed at 2024-12-24 08:53:57 in 2.90s

```
Performing stepwise search to minimize aic
 ARIMA(2,0,2)(0,0,0)[0] intercept   : AIC=883.277, Time=0.45 sec
 ARIMA(0,0,0)(0,0,0)[0] intercept   : AIC=1015.972, Time=0.10 sec
 ARIMA(1,0,0)(0,0,0)[0] intercept   : AIC=881.359, Time=0.05 sec
 ARIMA(0,0,1)(0,0,0)[0] intercept   : AIC=906.199, Time=0.08 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=1015.481, Time=0.03 sec
 ARIMA(2,0,0)(0,0,0)[0] intercept   : AIC=883.300, Time=0.06 sec
 ARIMA(1,0,1)(0,0,0)[0] intercept   : AIC=883.314, Time=0.09 sec
 ARIMA(2,0,1)(0,0,0)[0] intercept   : AIC=883.348, Time=0.20 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=879.982, Time=0.03 sec
 ARIMA(2,0,0)(0,0,0)[0]             : AIC=881.911, Time=0.04 sec
 ARIMA(1,0,1)(0,0,0)[0]             : AIC=881.927, Time=0.05 sec
 ARIMA(0,0,1)(0,0,0)[0]             : AIC=905.166, Time=0.03 sec
 ARIMA(2,0,1)(0,0,0)[0]             : AIC=881.947, Time=0.12 sec

Best model:  ARIMA(1,0,0)(0,0,0)[0]
Total fit time: 1.329 seconds
```

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	730
Model:	SARIMAX(1, 0, 0)	**Log Likelihood**	-437.991
Date:	Tue, 24 Dec 2024	**AIC**	879.982
Time:	15:53:57	**BIC**	889.168
Sample:	01-02-2019	**HQIC**	883.526
	- 12-31-2020		
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

D.3. Perform a forecast using the derived ARIMA model identified in part D2.

```
#Time Series Model -- Using p, d, q values from the auto_arima results -- (ARIMA(1,0,0))
model = ARIMA(df_stationary, order=(1, 0, 0), freq='D')
results = model.fit()
print(results.summary())
```

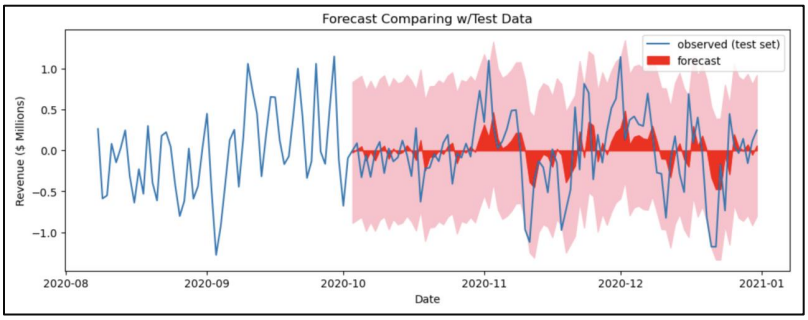
✔ Last executed at 2024-12-24 08:53:57 in 75ms

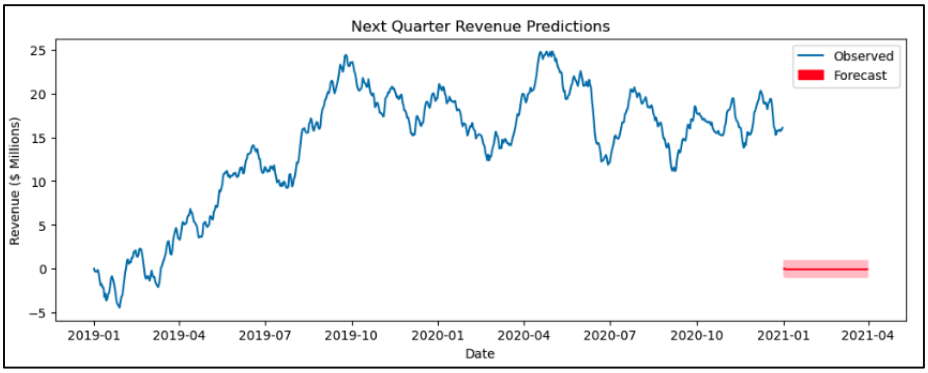
SARIMAX Results

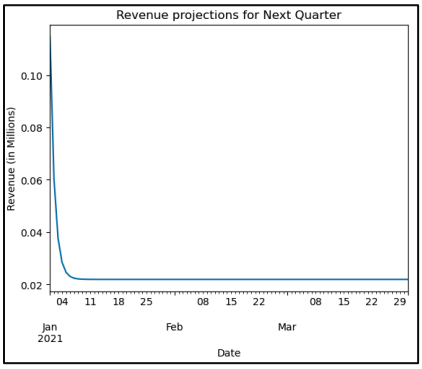
Dep. Variable:	Revenue	No. Observations:	730
Model:	ARIMA(1, 0, 0)	Log Likelihood	−437.680
Date:	Tue, 24 Dec 2024	AIC	881.359
Time:	15:53:57	BIC	895.139
Sample:	01-02-2019	HQIC	886.675
	- 12-31-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0219	0.028	0.789	0.430	−0.033	0.076
ar.L1	0.4130	0.034	12.218	0.000	0.347	0.479
sigma2	0.1942	0.011	17.851	0.000	0.173	0.215

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	1.92
Prob(Q):	0.93	Prob(JB):	0.38
Heteroskedasticity (H):	1.00	Skew:	−0.02
Prob(H) (two-sided):	0.98	Kurtosis:	2.75







D.4. Provide the output and calculations of the analysis you performed.

```
#Print mean absolute error
mae = np.mean(np.abs(results.resid))
print("Mean Absolute Error: ", mae)
```

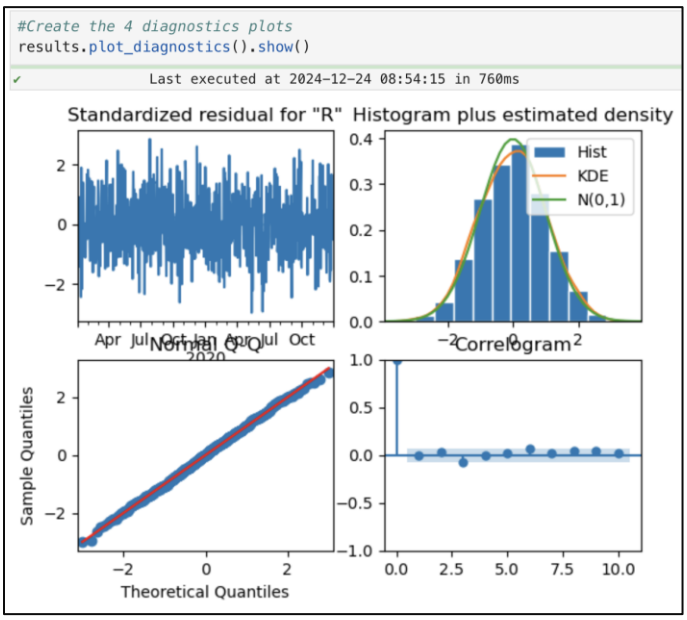
✔ Last executed at 2024-12-24 0

```
Mean Absolute Error:  0.3559307416455106
```

Mean Absolute Error (MAE) indicates the performance of the model and its ability to forecast. The lower the MAE, the better the model's ability to predict. Our model had a relatively small MAE; therefore, it will provide reasonably accurate revenue predictions.

```
#Create the 4 diagnostics plots
results.plot_diagnostics().show()
```

✔ Last executed at 2024-12-24 08:54:15 in 760ms



Standardized Residuals plot indicates that the residuals fluctuate randomly and have no apparent pattern. This means that the model fits the time series data well. The histogram plus estimated density plot visually demonstrates a normal distribution, and the KDE line is close to the normal distribution line, which supports normality for the residuals. The normal Q-Q plot confirms that the residuals (blue data points) align with a normal distribution (red line). For this model, the data points surround the red line, confirming the model's residuals do follow a normal distribution. The Correlogram (ACF) plot shows the correlation between residuals at specific lags. For this model, the autocorrelations are all within the blue shaded region: confidence intervals. This suggests no significant autocorrelation in the residuals and indicates that the model fits well with the data's time dependence.

D.5. Provide the code used to support the implementation of the time series model.

See the files attached in the task submission.

Part V: Data Summary and Implications

E.1. Discuss the results of your data analysis, including the following points:

- **the selection of an ARIMA model**

The ARIMA model (1, 0, 0) was chosen based on stationarity, model order, and diagnostic plots. The data's stationarity was determined by using the augmented Dicky Fuller test and the p-value. Both tests concluded that the dataset was stationary and had a p-value of less than 0.05. The model order (AR=1) was chosen over a moving average term as the auto ARIMA code outputs an autoregressive suggestion. The AR model had the lowest AIC value and confirmed that the model would fit the data's structure well. The plots and statistical tests completed after running the model further confirmed how well the model fits the dataset and how well the model would predict a forecast. Those statistical tests confirmed that the model's residuals were normally distributed, uncorrelated, and had a constant variance.

- **the prediction interval of the forecast**

Based on the research question, forecasting the next quarter's revenue (90 days), the prediction interval used was 90 days. This is 90 days after the last 'step' or date in the training/testing datasets.

- **a justification of the forecast length**

Again, 90 days (steps) was chosen as the prediction interval because the research question was to predict or forecast the hospital's next quarter revenue.

- **the model evaluation procedure**

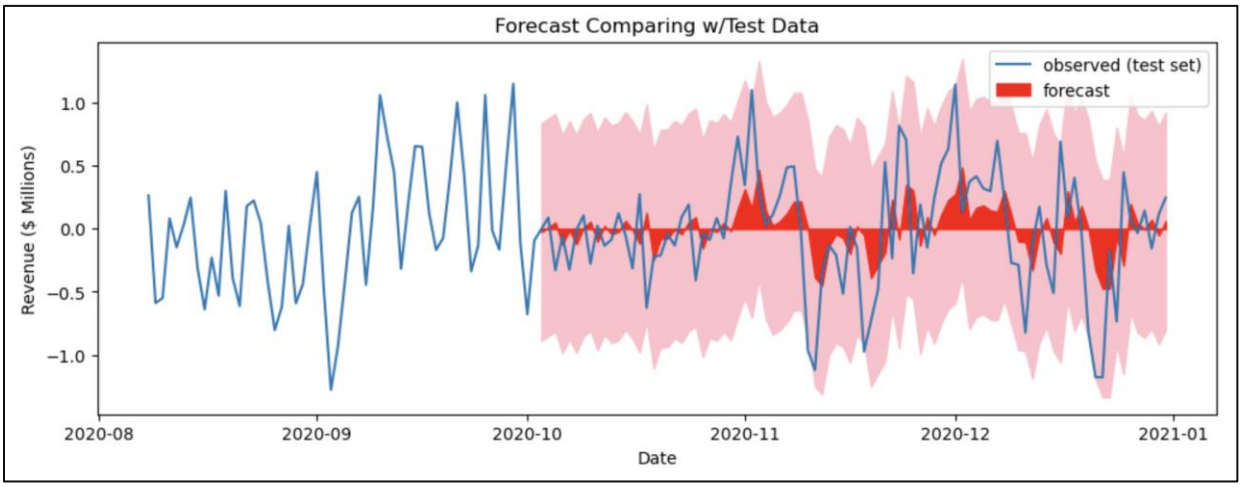
Residual tests were conducted to evaluate the model's fit to the data. The results indicated no autocorrelation. The residuals were normally distributed and showed a constant variance. These suggested that the model was a good fit for the data. The Ljung-Box test results were 0.93, which confirms that the residuals are uncorrelated.

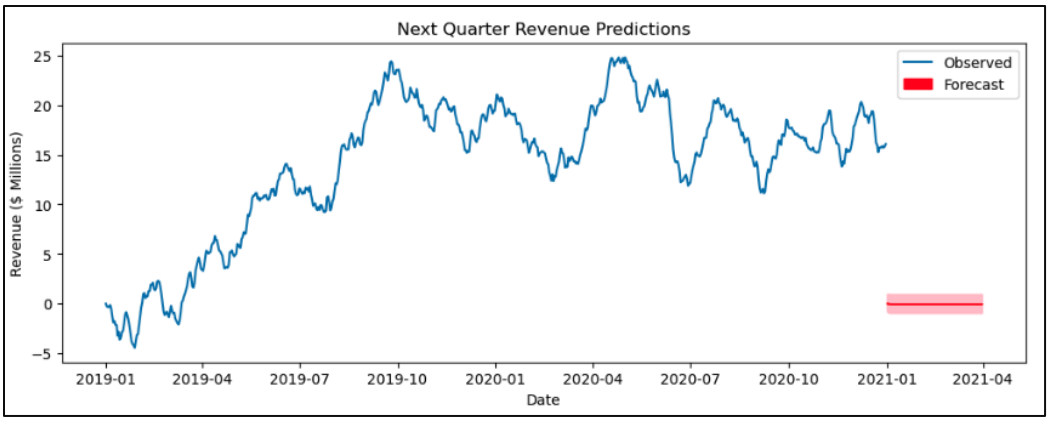
- **the model error metric**

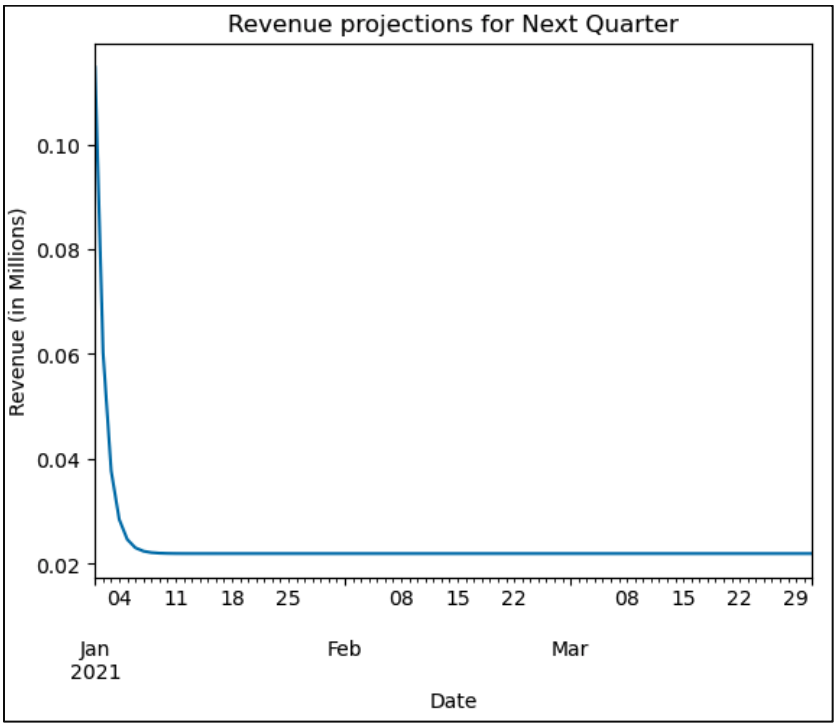
Mean Absolute Error (MAE) was completed to measure the model's prediction accuracy. The lower the number for MAE, the better the model's prediction accuracy. For our model, MAE was approximately 0.36; this is very low (near 0) and indicates the model's predictions are relatively accurate. In addition to MAE, the AIC value helped choose a model (with the lowest AIC) compared to the others computed from auto ARIMA.

E.2. Provide an annotated visualization of the forecast of the final model compared to the test set.

The forecast (screenshot below) follows the observed revenue in the test dataset. However, there is a delay in when (different dates) the revenue changes will occur with the model.







E.3. Recommend a course of action based on your results.

Based on the forecasted quarter revenue results, the company should continue to monitor and forecast revenue for short-term forecasts. This will help the business identify quarters with potential revenue decline and take a proactive measure(s) to boost revenue. While monitoring the changes in revenue, the hospital can test how external factors, for example, staffing, services provided, or hours of operation, directly change the amount of revenue gained or lost during that period. The more data gathered (more than 2 years that was provided) would further enhance the insights provided and improve the model's predictability.

Part VI: Reporting

F. Code Document

See the files attached in the task submission.

G. Sources

DataCamp. (2024). The best of the best models. Retrieved December 29, 2024, from https://campus.datacamp.com/courses/arima-models-in-python/the-best-of-the-best-models?learningMode=course&ex=1

Elleh, Festus (2024). D213 Task 1 Cohort Webinar PPT [PowerPoint slides]. WGU. https://westerngovernorsuniversity.sharepoint.com/:p:/r/sites/DataScienceTeam/_layouts/15/doc2.aspx?sourcedoc=%7BDC38E039-2BB1-4EBE-B156-F79C16334FC6%7D&file=D213%20Task%201%20Cohort%20Webinar%20PPT.pptx&action=edit&mobileredirect=true&DefaultItemOpen=1&ct=1736010748112&wdOrigin=OFFICECOM-WEB.MAIN.REC&cid=7c06c730-bb58-40f2-8ea6-99e17317091f&wdPreviousSessionSrc=HarmonyWeb&wdPreviousSession=db952004-fb17-45da-a8c9-e344d9403590

GeeksforGeeks. (July 05, 2022). Plot the power spectral density using Matplotlib – Python. Retrieved December 29, 2024, from https://www.geeksforgeeks.org/plot-the-power-spectral-density-using-matplotlib-python/

Sewell, William. (2024). Time Series Analysis [PowerPoint slides]. WGU. https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william_sewell_wgu_edu/EZXf-NityGNLhehsXVZLyZYBd3bBpEuAh-W71wZJnLcEyA?e=Lu16z9

Stack Overflow. (n.d.). Converting day count to date time. Retrieved December 29, 2024, from https://stackoverflow.com/questions/61389654/converting-day-count-to-date-time

Stack Overflow. (n.d.). Add trendline for time series graph. Retrieved December 29, 2024, from https://stackoverflow.com/questions/61011711/add-trendline-for-timeseries-graph