Machine Learning

Task 3

Nathan Hefner

B. Describe the purpose of this data mining report by doing the following:

1. Propose one question relevant to a real-world organizational situation

One question relevant to our situation when analyzing revenue over time is how and why the revenue changes over time. Are there certain time periods or anomalies that we can track and predict in future forecasts?

2. Define one goal of the data analysis. Ensure your goal is reasonable within the scope of the selected scenario and is represented in the available data.

One goal of this analysis will be to find insights on why revenue changes over time, so then we can capitalize on events to increase revenue.

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

There are multiple assumptions for a time series model, such as:

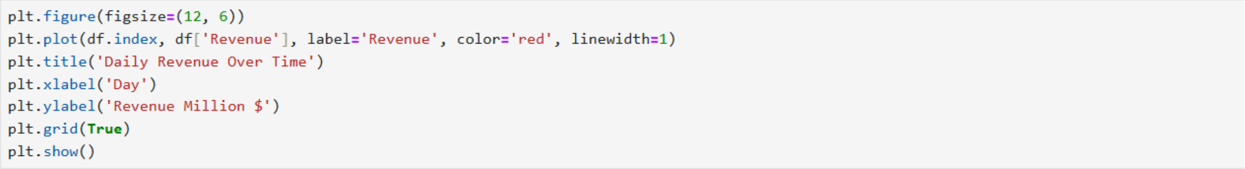
1. Linearity - Time Series Models assume a linear relationship between current and past values
2. Homoscedasticity - The residuals should have constant variance over time
3. Normality - Assumes normally distributed errors

The two most important are stationarity and autocorrelated data. There are two types of Stationarity: strict, when anomalies change the data over time, and weak, when the mean, variance, and autocovariance are constant over time. The second is Autocorrelation, meaning when observations in the time series are correlated with their past values.

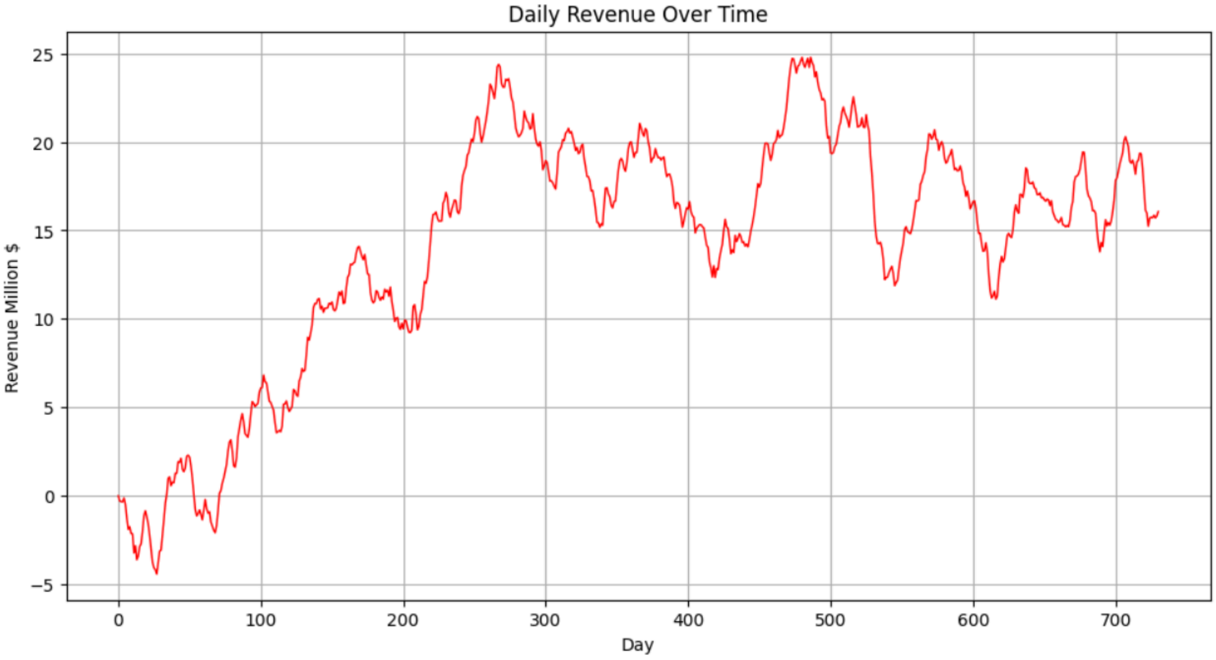
D. Summarize the data cleaning process by doing the following:

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

Below is the code and graph for revenue versus time



```
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Revenue'], label='Revenue', color='red', linewidth=1)
plt.title('Daily Revenue Over Time')
plt.xlabel('Day')
plt.ylabel('Revenue Million $')
plt.grid(True)
plt.show()
```



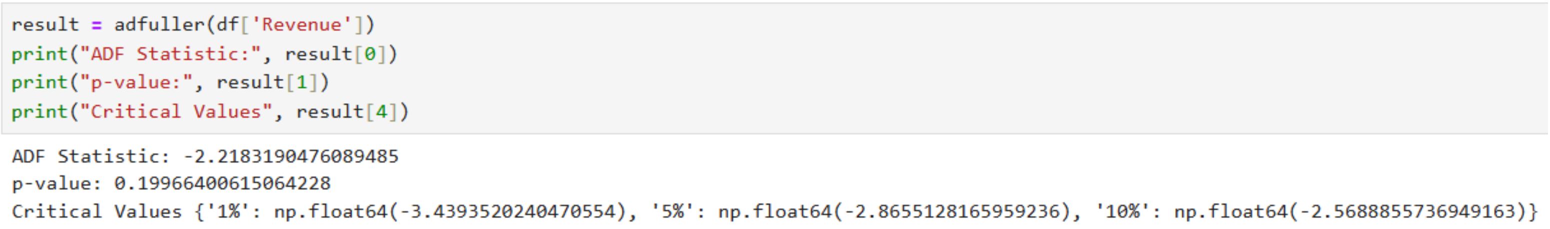
2. Describe the time step formatting of the realization, including any gaps in measurement and the length of the sequence.

The graph is Revenue Vs. Time in days, there are 731 entries, and none of which are null or duplicates, meaning there are no gaps or breaks in the data, allowing for a smooth and continuous line graph.

3. Evaluate the stationarity of the time series.

There are two ways to evaluate the stationarity of the time series: visual evaluation and the Augmented Dickey-Fuller (ADF) Test. During a visual evaluation, I take note of the trends and patterns in the graph and notice similar repeating trends along with cycles of crashes and spikes.

After this, I ran the code to run the ADF test for definitive answers as visual evaluation is not always reliable.



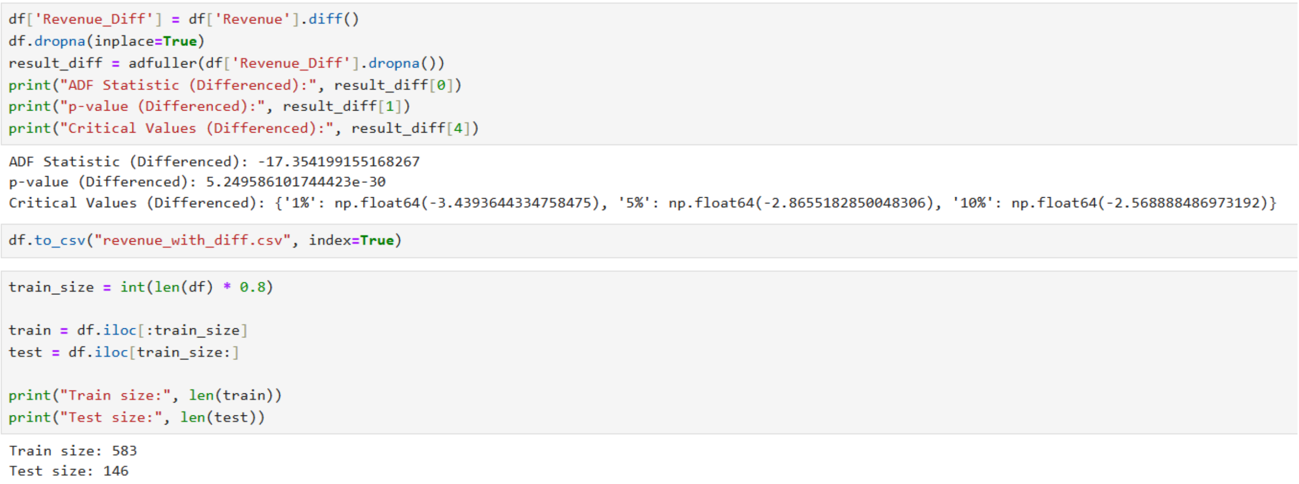
```
result = adfuller(df['Revenue'])
print("ADF Statistic:", result[0])
print("p-value:", result[1])
print("Critical Values", result[4])
```

```
ADF Statistic: -2.2183190476089485
p-value: 0.19966400615064228
Critical Values {'1%': np.float64(-3.4393520240470554), '5%': np.float64(-2.8655128165959236), '10%': np.float64(-2.5688855736949163)}
```

Given the results, we can see that the p-value is 0.199, which means we reject the null hypothesis, confirming that the data is not stationary.

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

Since our data was not stationary, we had to difference the dataset to make it so. The code I used is as seen below.



```
df['Revenue_Diff'] = df['Revenue'].diff()
df.dropna(inplace=True)
result_diff = adfuller(df['Revenue_Diff'].dropna())
print("ADF Statistic (Differenced):", result_diff[0])
print("p-value (Differenced):", result_diff[1])
print("Critical Values (Differenced):", result_diff[4])
```

```
ADF Statistic (Differenced): -17.354199155168267
p-value (Differenced): 5.249586101744423e-30
Critical Values (Differenced): {'1%': np.float64(-3.4393644334758475), '5%': np.float64(-2.8655182850048306), '10%': np.float64(-2.568888486973192)}
```

```
df.to_csv("revenue_with_diff.csv", index=True)
```

```
train_size = int(len(df) * 0.8)

train = df.iloc[:train_size]
test = df.iloc[train_size:]

print("Train size:", len(train))
print("Test size:", len(test))
```

```
Train size: 583
Test size: 146
```

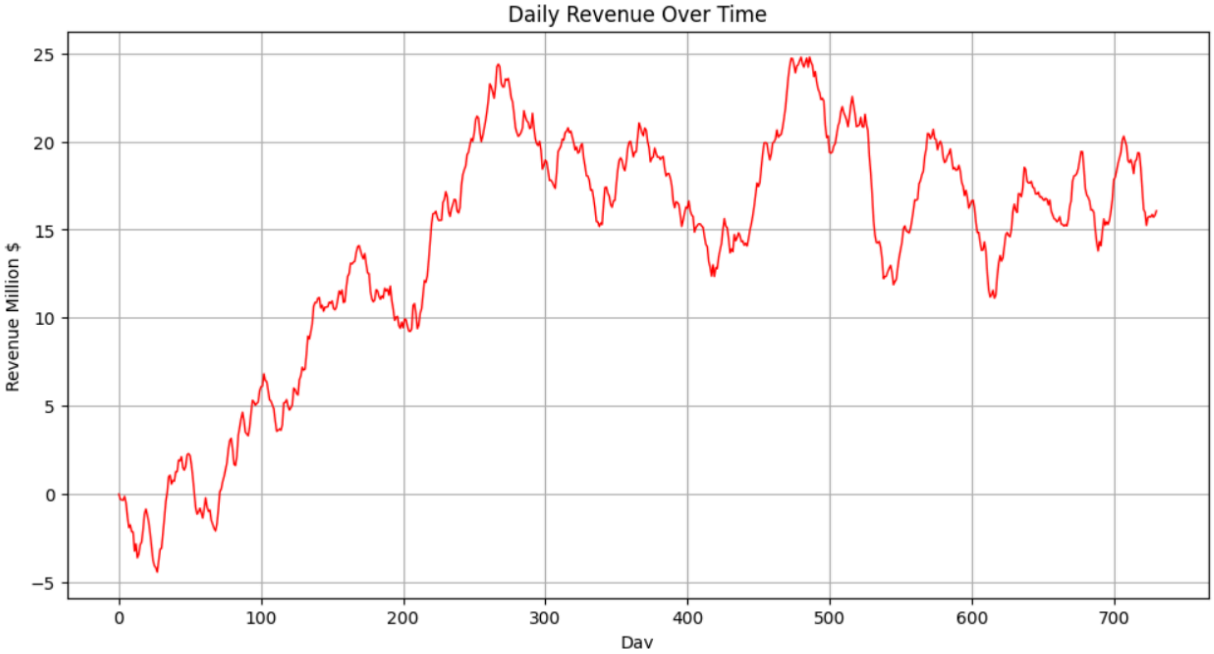
The p-value is far below our threshold, now making the data stationary and ready for deeper analysis. As you can see as well we split the data by 80/20. This will also help prepare our data for future analysis.

E. Analyze the time series dataset by doing the following:

1. Report the annotated findings with visualizations of your data analysis

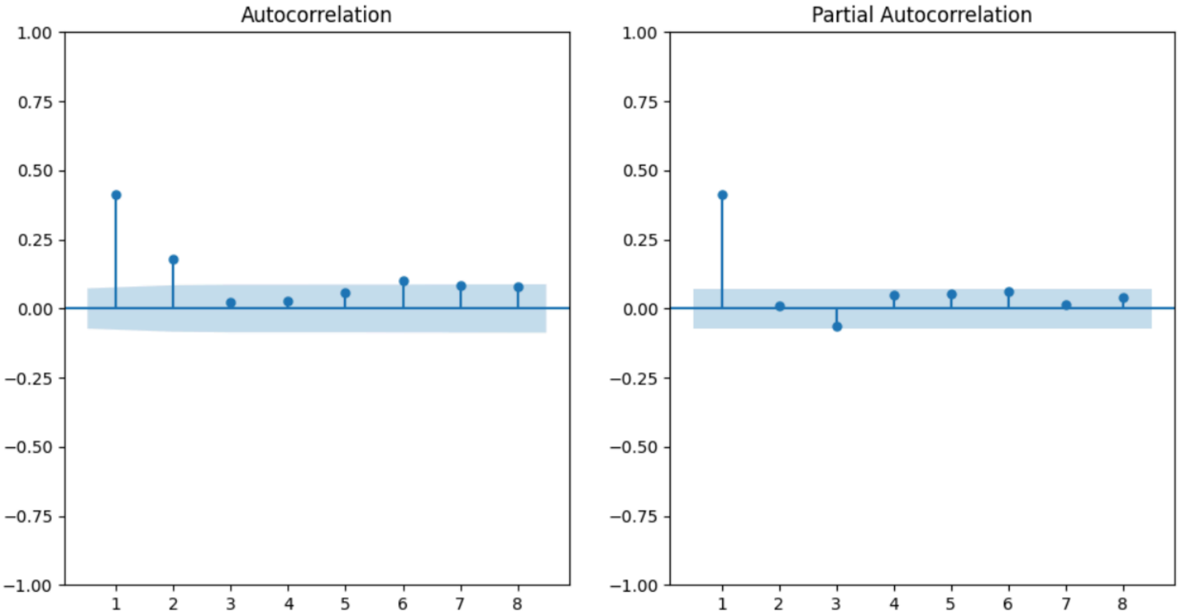
After differencing the data and using the training dataset, we can now analyze different graphs to gain more insight into this Time Series Model

Trends



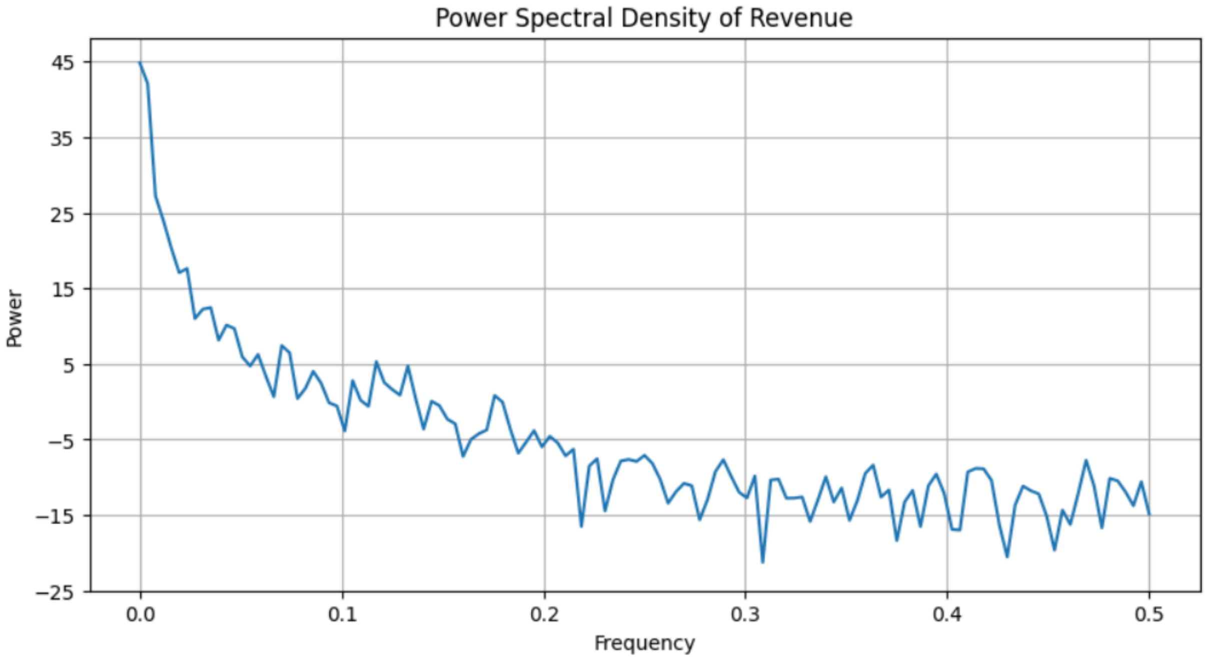
This graph shows that there’s an initial upward momentum movement in the early days that appears to plateau before day 300 as it cycles back downwards and back upwards multiple times. This could also suggest seasonal effects or external factors.

Autocorrelation Function



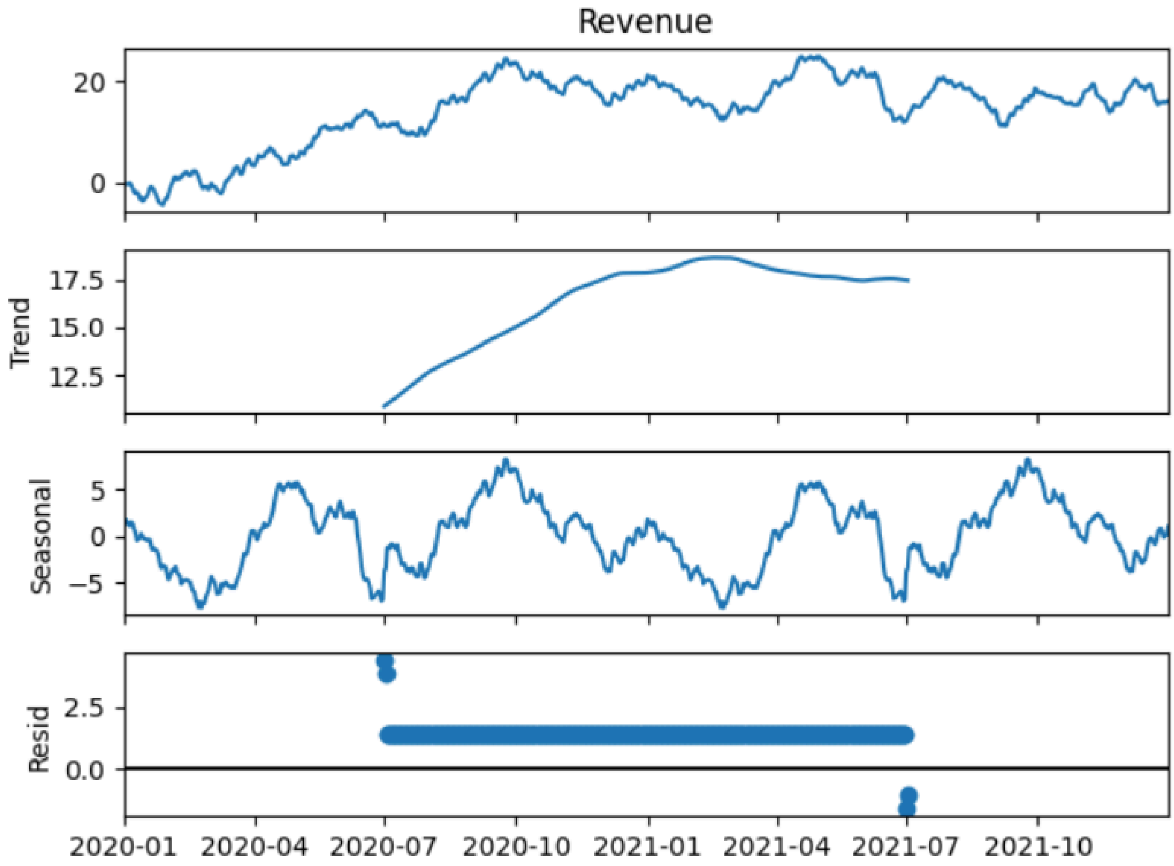
The Autocorrelation Function graph shows us that there is a high correlation, as seen with lag 1 being nearly at 1, meaning revenue earned today is strongly correlated with revenue earned the previous day. There is also a slow decay, which means the past influences the present. There are also many bands inside the blue confidence interval, which indicates that the data points are insignificant. This also allows us to determine p=1 and q=0 for the model in future analysis.

Spectral Density



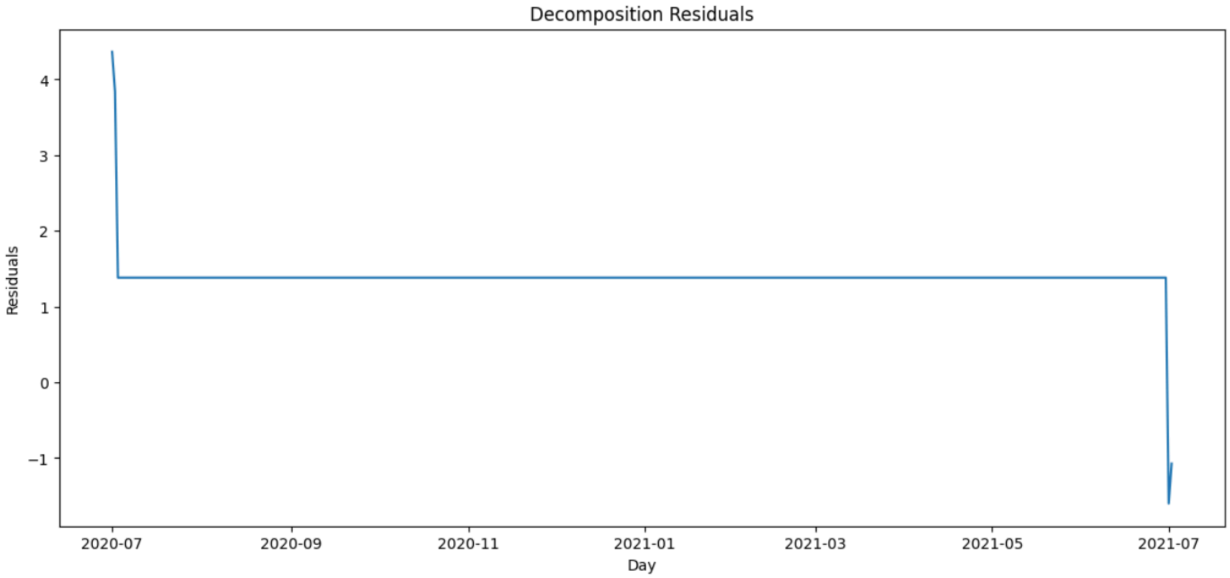
This graph shows a peak near 0. This indicates a strong trend of overall upward momentum. Power drops off rapidly, suggesting that high-frequency changes are less significant. There is also a lack of other strong peaks; this shows there’s no clear seasonal or external factors affecting the data.

Decomposed Time Series



These graphs show numerous things. The trend shows there's a period of growth and upward momentum, but then it stagnates and dips a little. Seasonal shows a clear cycle, meaning the data has seasonal effects. The residual has some outliers, which shows there may be external factors or anomalous events occurring.

Residuals



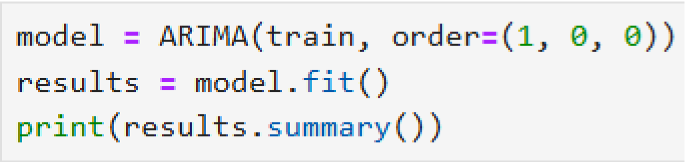
The residuals are mostly flat except for a large spike at the beginning and a large dip at the end, which could indicate them being outliers.

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

Since the data is consistent and the past is a strong indicator of the future, and the effects of seasonal changes or external factors are minimal, I will use the ARIMA Model as it is optimal for our analysis, using the Autocorrelation and Partial Autocorrelation graphs from earlier we can determine the most optimal parameters for our ARIMA Model will be [1, 0, 0].

3. Perform a forecast using the derived ARIMA model identified in part E2.

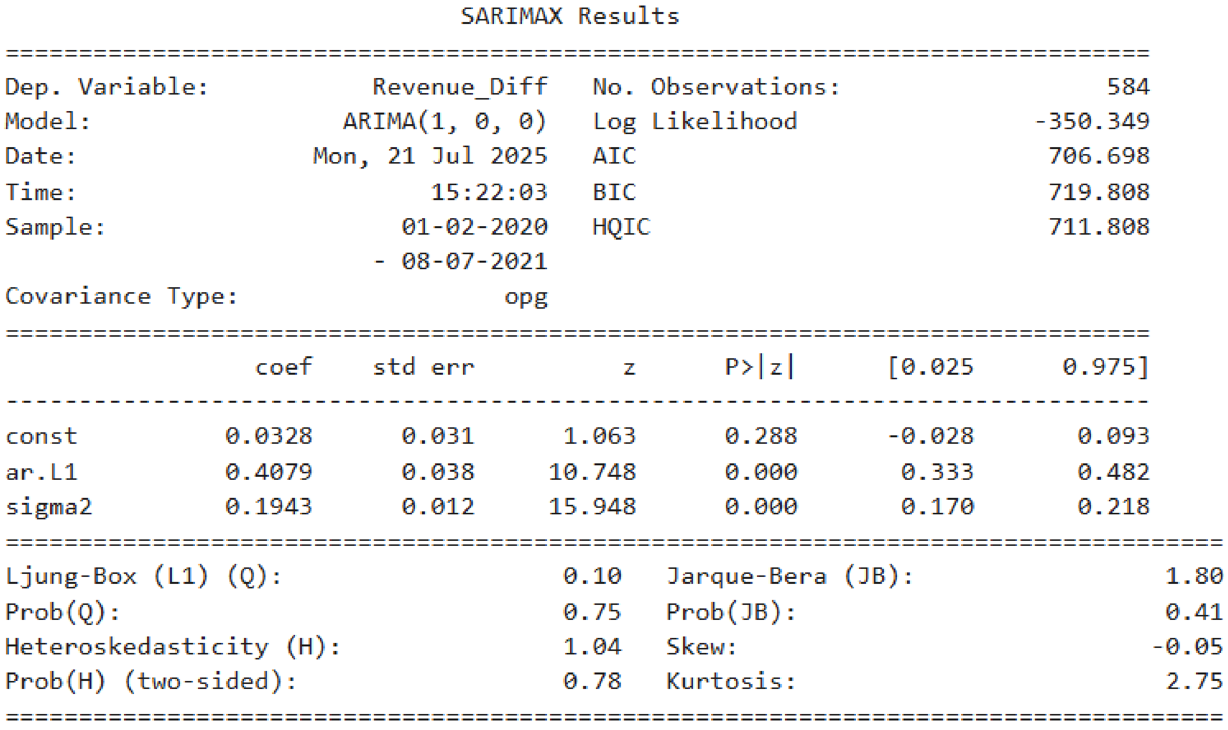
Below is the code I used to forecast with the ARIMA model we found was most optimal.



```
model = ARIMA(train, order=(1, 0, 0))
results = model.fit()
print(results.summary())
```

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

Below is the output.



```
                               SARIMAX Results
==============================================================================
Dep. Variable:            Revenue_Diff   No. Observations:                  584
Model:                  ARIMA(1, 0, 0)   Log Likelihood                -350.349
Date:                 Mon, 21 Jul 2025   AIC                            706.698
Time:                         15:22:03   BIC                            719.808
Sample:                     01-02-2020   HQIC                           711.808
                          - 08-07-2021
Covariance Type:                   opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0328      0.031      1.063      0.288      -0.028       0.093
ar.L1          0.4079      0.038     10.748      0.000       0.333       0.482
sigma2         0.1943      0.012     15.948      0.000       0.170       0.218
===================================================================================
Ljung-Box (L1) (Q):                   0.10   Jarque-Bera (JB):                 1.80
Prob(Q):                              0.75   Prob(JB):                         0.41
Heteroskedasticity (H):               1.04   Skew:                            -0.05
Prob(H) (two-sided):                  0.78   Kurtosis:                         2.75
===================================================================================
```

F. Summarize your findings and assumptions by doing the following:

1. Discuss the results of your data analysis

ARIMA Model

The model we chose was ARIMA because the trends and graphs did not support that seasonality had a strong effect on the data. After differencing the data, we made Autocorrelation and Partial Autocorrelation graphs to find our p, d, and q values. Which ended up being [1, 0, 0]. This model captures the trend component while ignoring seasonality.

Prediction Interval of the Forecast

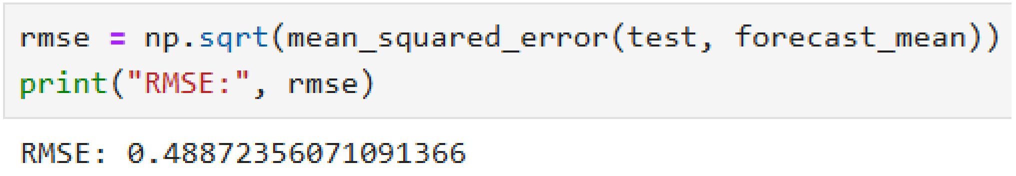
The prediction interval was 146 days because we used the standard 80/20 training/test split. Meaning out of the 730 days, we used 584 training days to predict 146 days of data. 95% confidence intervals for prediction were also used for 146 rows.

Justification of Forecast Length

The length is 146 days because of the standard 80/20 training/test split. We want to ensure there’s enough data for the model to be trained on so it can accurately predict the remaining 20% (146 days).

Model Evaluation Procedure and Error Metric

I calculated the Root Mean Squared Error with the code below



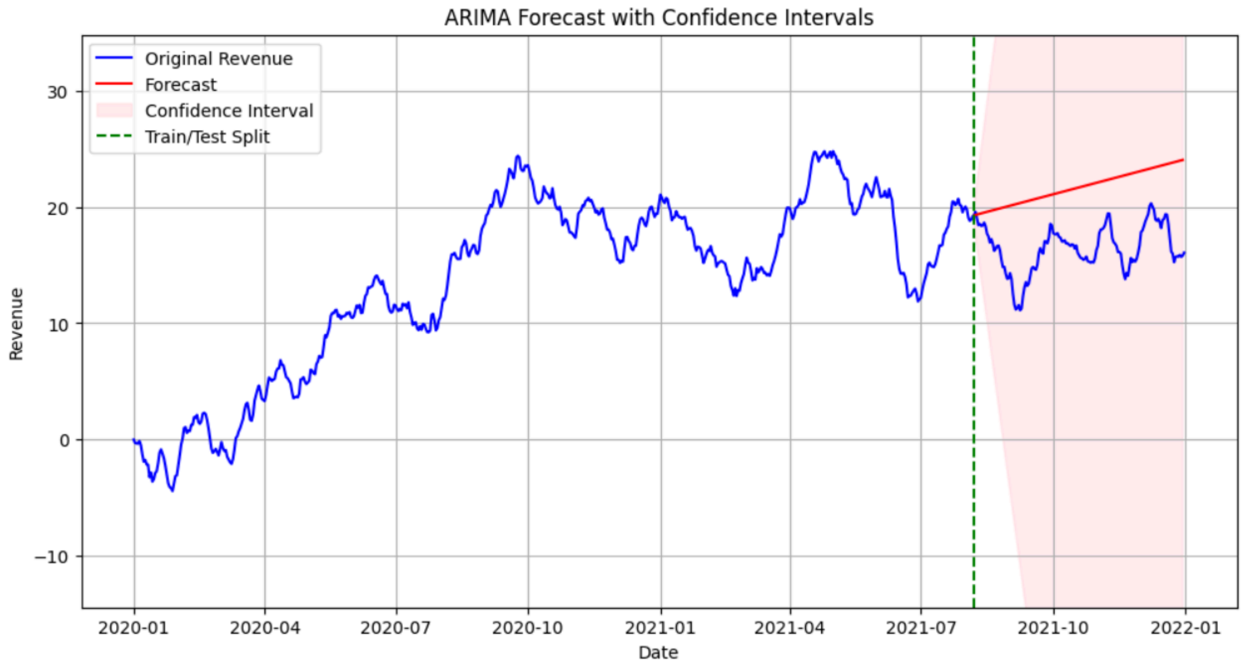
```
rmse = np.sqrt(mean_squared_error(test, forecast_mean))
print("RMSE:", rmse)
```

```
RMSE: 0.48872356071091366
```

This value is the average distance from the predicted values to the actual values. A rough value of 0.49 shows strong predictions for future revenue, meaning the training dataset has tested well against the test dataset.

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

Below is the code for my visualization for the forecast of the final model compared to the test set.



This graph shows the confidence cone, the original revenue, and the forecast that has an upward momentum, but is still relatively close to the original revenue.

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

After training the model with the dataset and testing the data, which tests relatively well, I would recommend using this model for short-term projections and forecasting only. I believe every quarter will be optimal because the forecast accuracy will fall off after a certain point in time. The new data will also help reinforce its forecasting ability for future use.

Coding Sources
Datacamp ARIMA Models in Python
https://app.datacamp.com/learn/courses/arima-models-in-python

Sources:
No sources used except WGU Material