Peyton Bailey

November 19th, 2025

D603 – Task 3

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

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

How can we model and forecast daily revenue over the first two years of operation to predict future revenue fluctuations influenced by customer churn patterns? This forecasting question is crucial for telecommunications providers to anticipate revenue changes driven by customer retention and churn behavior, enabling them to make proactive business decisions.

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

The goals of my data analysis are to:

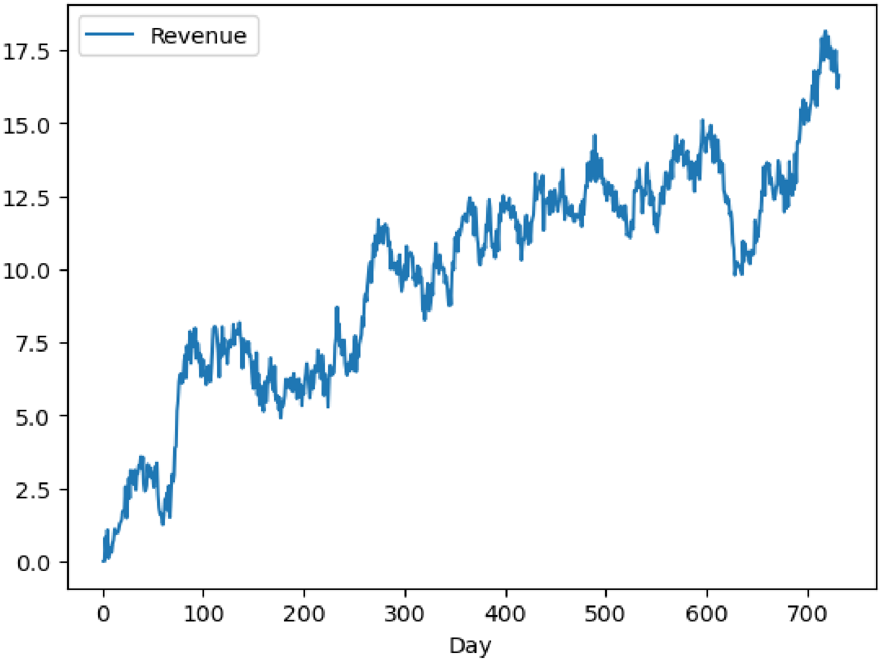
- Analyze the daily revenue time series data for patterns, trends, and seasonality.
- Build and validate a time series forecasting model that captures these dynamics using historical data to inform predictions.
- Generate accurate forecasts of future daily revenue, which can help the organization plan marketing, retention strategies, and financial expectations.
- Quantitatively assess model performance by training on initial data segments and testing on holdout sets, ensuring forecast reliability within the scope of the available two years of data.

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

Time series models allow us to understand patterns in data over time. Two properties of time series models are stationarity and autocorrelation. Stationarity refers to the property of a time series where its statistical properties remain constant over time. Autocorrelation measures the relationship between a time series and itself at different time lags. (Fiveable, 2025). To properly model a time series, the criteria are that it is stationary and the autocorrelation remains constant. Additionally, the time series must have a zero trend, meaning it neither grows nor shrinks. Additionally, variance is continuous, meaning the average distance of the data points from the zero line remains unchanged.

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

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



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

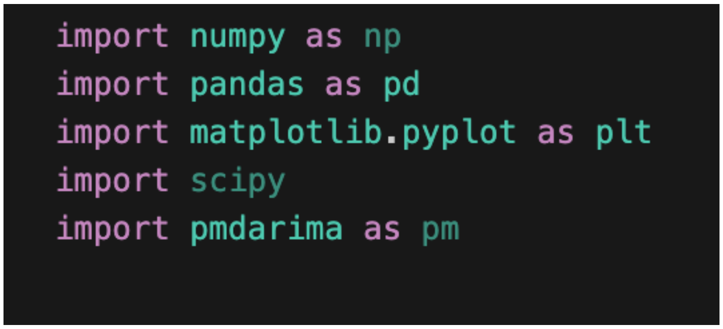
The churn time series dataset consists of 731 data points. Each data point represents one day, along with the total revenue generated on that day. The time column was initially formatted as ascending integers representing each day (Day 1, Day 2, etc). I reformatted this column to represent the date, e.g, (01/01/2019). The full dataset spans a two-year period with no gaps in measurement. When splitting this data into training and test sets, I used the first 717 points for the training data and the last 14 points (corresponding to 2 weeks) as the test data.

3. Evaluate the stationarity of the time series.

For a time series to be stationary, it must fulfill three criteria. These are: the time series has zero trend, the variance is constant, and the autocorrelation is constant (DataCamp, 2025). I used the seasonal decomposition function to visually evaluate these properties. This function allowed me to visualize the trend, seasonal, and residual components. The trend component showed a clear upward trend. The residuals were also not constant. I further confirmed this visually by plotting the rolling mean and rolling standard deviation against the time series data plot. On my plot, neither the rolling mean nor the standard deviation was flat, indicating that they were not constant, which suggests that this time series is non-stationary. I conducted further analysis using the Augmented Dickey-Fuller Test (ADF) to statistically evaluate the stationarity of the time series. The ADF test has a null hypothesis stating that the time series is non-stationary due to a trend. Any result that gives a p-value less than 0.05 would allow us to reject the null hypothesis. The results from this test on my data produced a p-value of 0.47, a value much greater than 0.05, indicating that I rejected the null hypothesis and concluded that the time series was non-stationary.

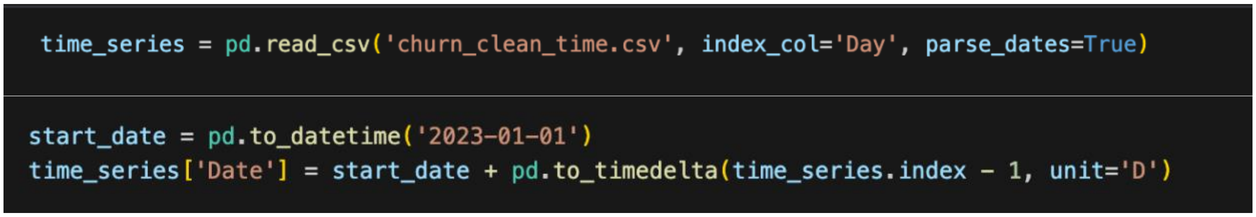
4. Explain the steps used to prepare the data for analysis, including the training and test set split.

1. The first step was importing the necessary packages.



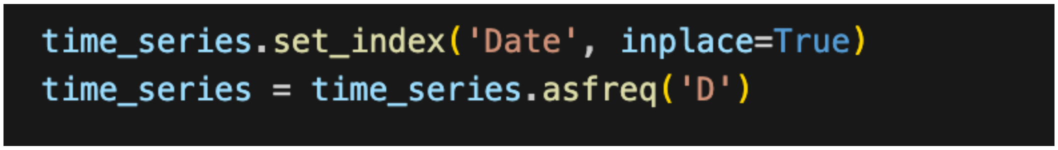
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy
import pmdarima as pm
```

2. Then, I imported and formatted the data.



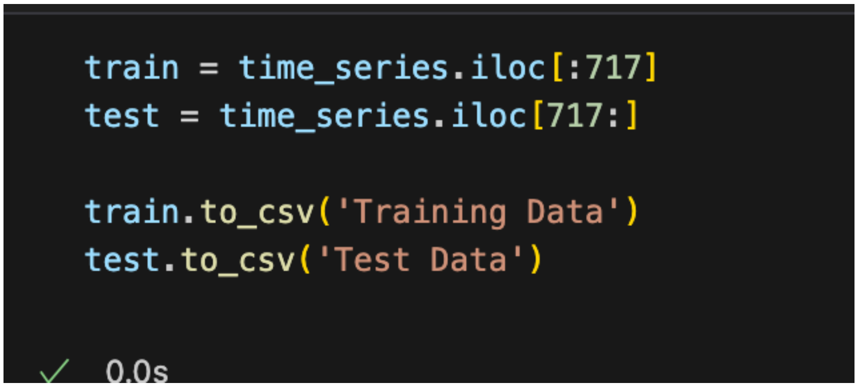
```
time_series = pd.read_csv('churn_clean_time.csv', index_col='Day', parse_dates=True)
```

```
start_date = pd.to_datetime('2023-01-01')
time_series['Date'] = start_date + pd.to_timedelta(time_series.index - 1, unit='D')
```



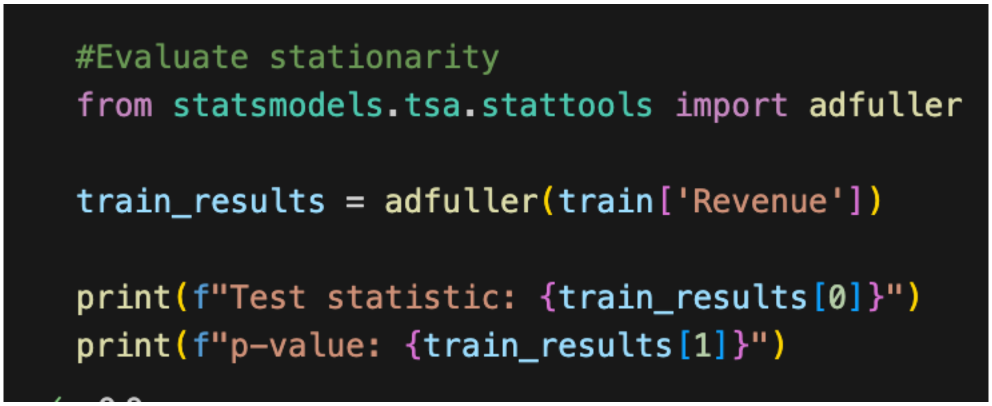
```
time_series.set_index('Date', inplace=True)
time_series = time_series.asfreq('D')
```

3. Next, I split the data into train and test sets.



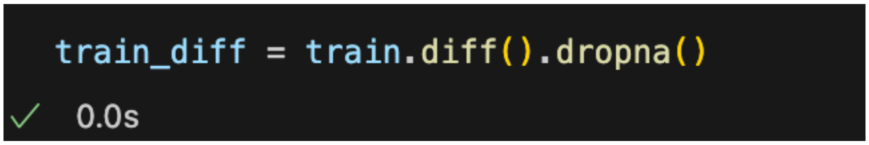
```
train = time_series.iloc[:717]
test = time_series.iloc[717:]

train.to_csv('Training Data')
test.to_csv('Test Data')

✓ 0.0s
```

4. Then, I evaluated stationarity using the Ad-Fuller test.



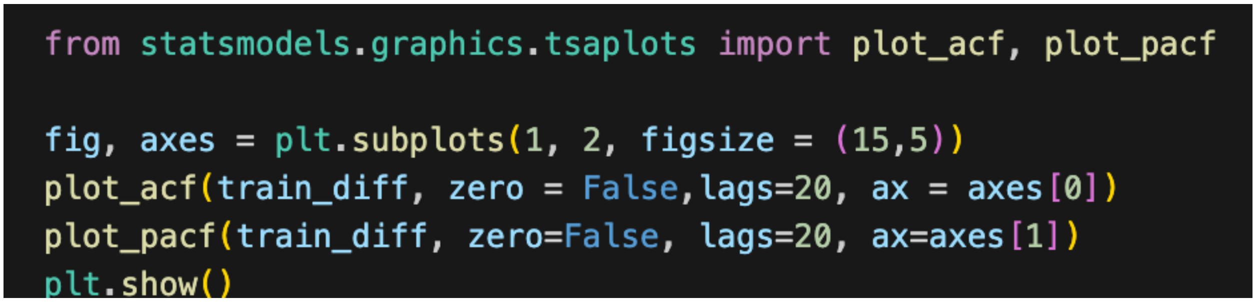
```
#Evaluate stationarity
from statsmodels.tsa.stattools import adfuller

train_results = adfuller(train['Revenue'])

print(f"Test statistic: {train_results[0]}")
print(f"p-value: {train_results[1]}")
```

5. Then I performed first-order differencing on the data to make it stationary.



```
train_diff = train.diff().dropna()
✓ 0.0s
```

6. Finally, I created ACF and PACF plots to determine the order of the ARIMA model



```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

fig, axes = plt.subplots(1, 2, figsize = (15,5))
plot_acf(train_diff, zero = False,lags=20, ax = axes[0])
plot_pacf(train_diff, zero=False, lags=20, ax=axes[1])
plt.show()
```

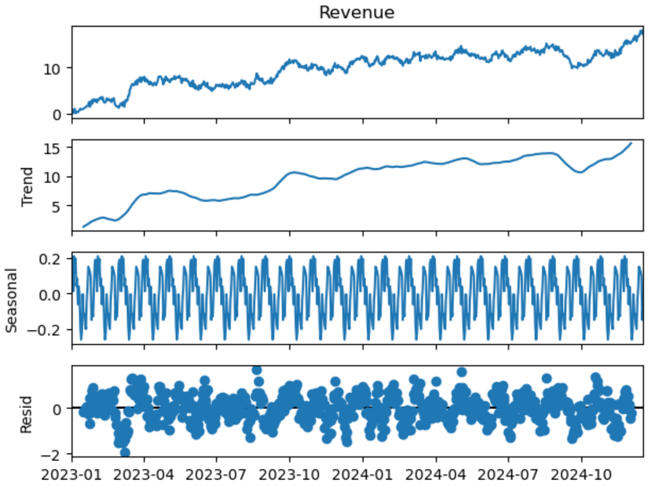
5. Provide a copy of the cleaned dataset.

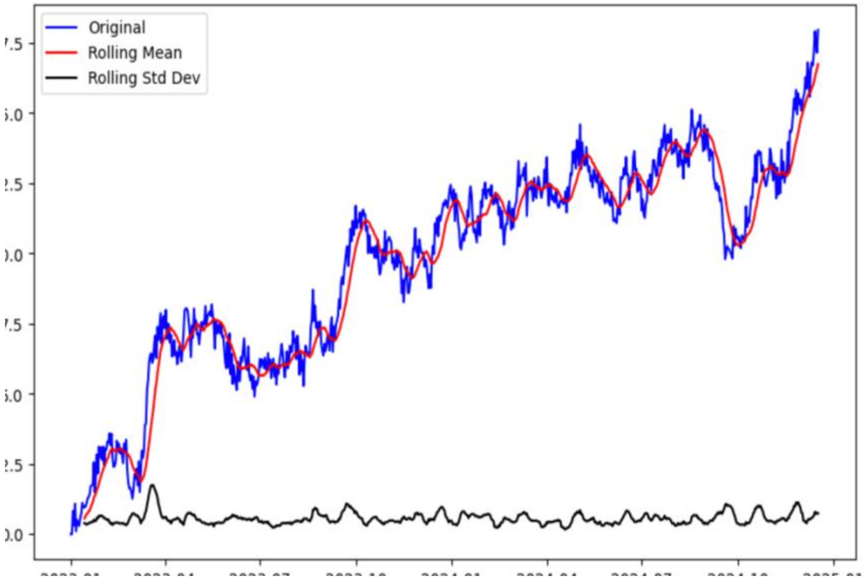
E. Analyze the time series dataset by doing the following:

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

- **trends**
- **the autocorrelation function**
- **the spectral density**
- **the decomposed time series**
- **confirmation of the lack of trends in the residuals of the decomposed series**

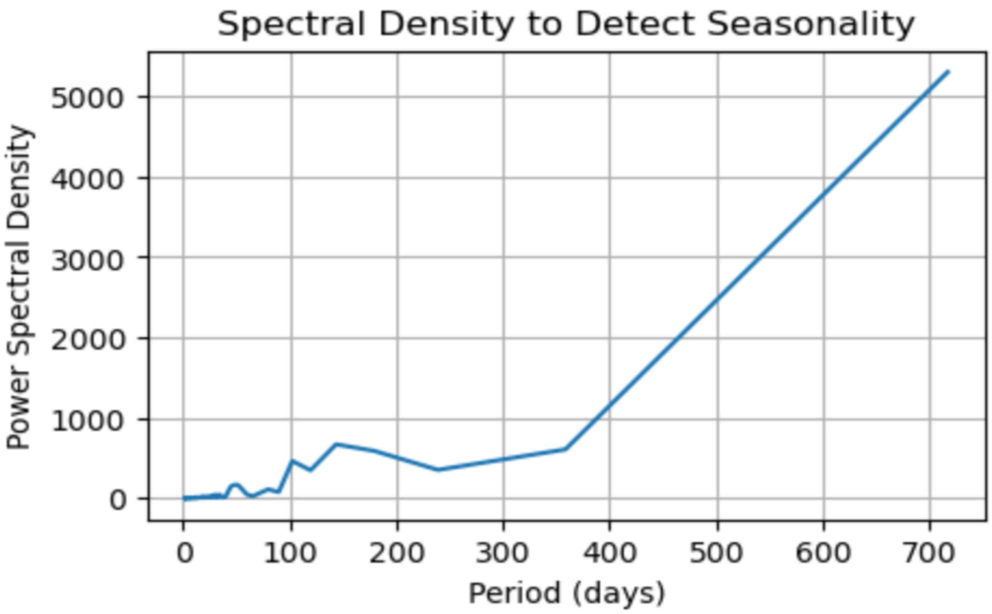
The seasonal decomposition plot displays three components of a time series: trend, seasonality, and noise (or residuals). The decomposition of the original data shows a strong and persistent trend component, visible seasonal effects, and non-random residuals. The decomposition of the differenced series (and its residuals) shows no evident trend and stationary residuals with no systematic drift.

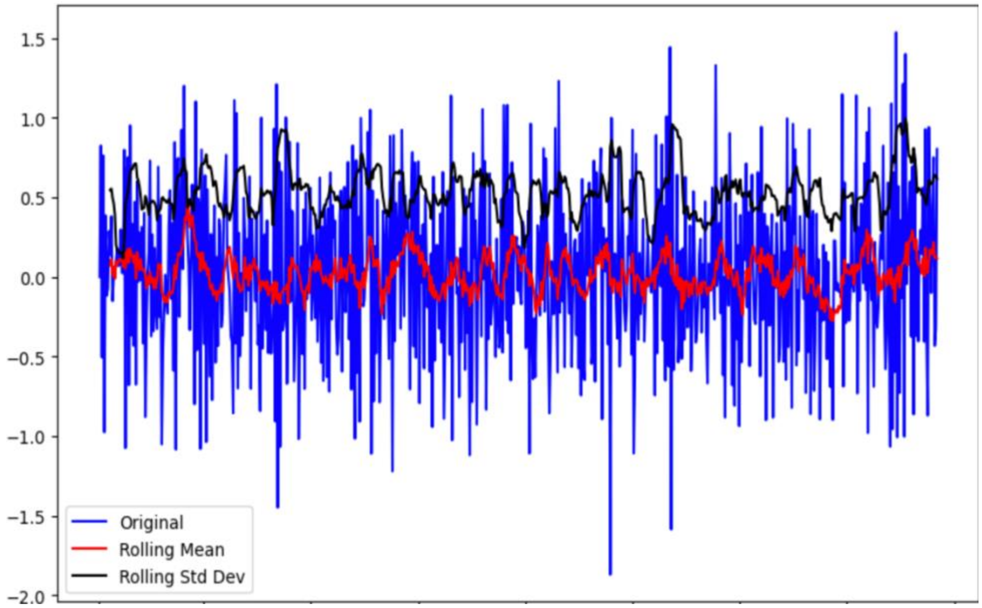




The original series, plotted with its rolling mean and rolling standard deviation, shows both metrics having drift and varying substantially over time. This confirms that the original data is non-stationary and dominated by trend and possible structural change.

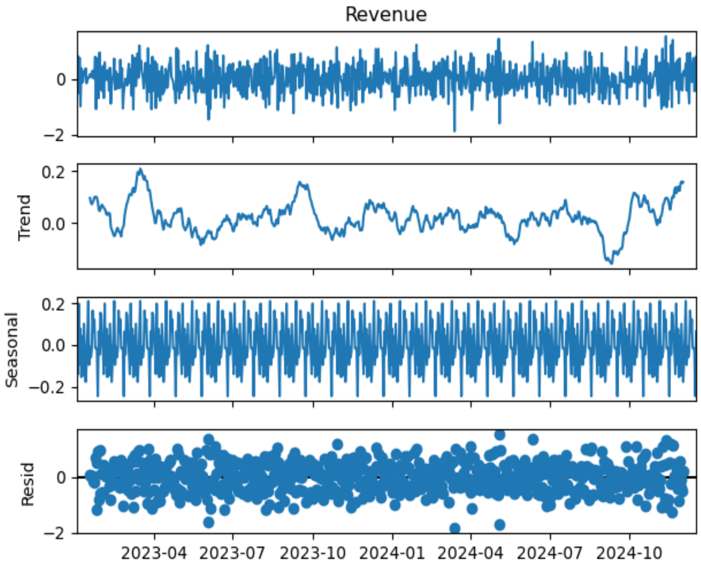
The original series spectral density plot shows power concentrated at very low frequencies and lacks sharp seasonal spikes, suggesting minimal or no regular seasonality structure in the original data.

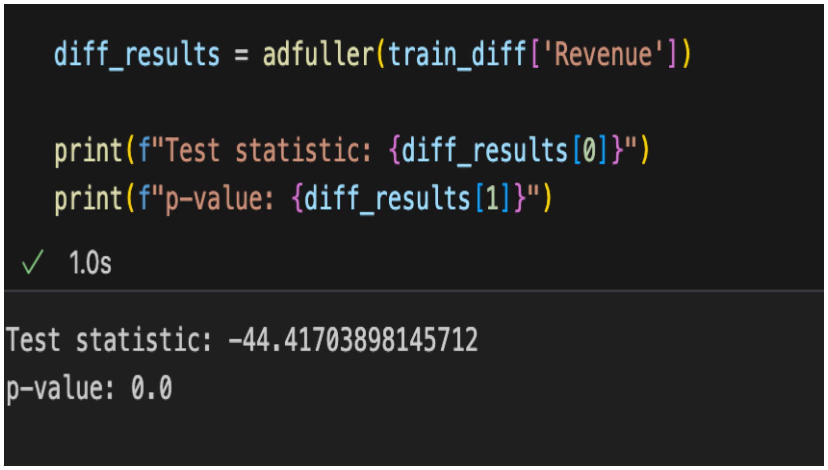




After differencing, the series appears as stochastic, random fluctuations about zero without an obvious slope, supporting the visual assertion of trend elimination and stationarity. The rolling mean and standard deviation after differencing show that both metrics are stable and oscillate closely around zero, confirming stationarity and the removal of the underlying trend and changing variance.

The seasonal decomposition plot of the differenced series confirms trend elimination—the trend component hovers close to zero, the seasonal component is much reduced, and the residuals remain stationary and randomly scattered. The lack of noticeable drift in the trend panel and the uniform scatter in the residual panel further reinforce the success of stationarity and the appropriateness of ARIMA modeling.



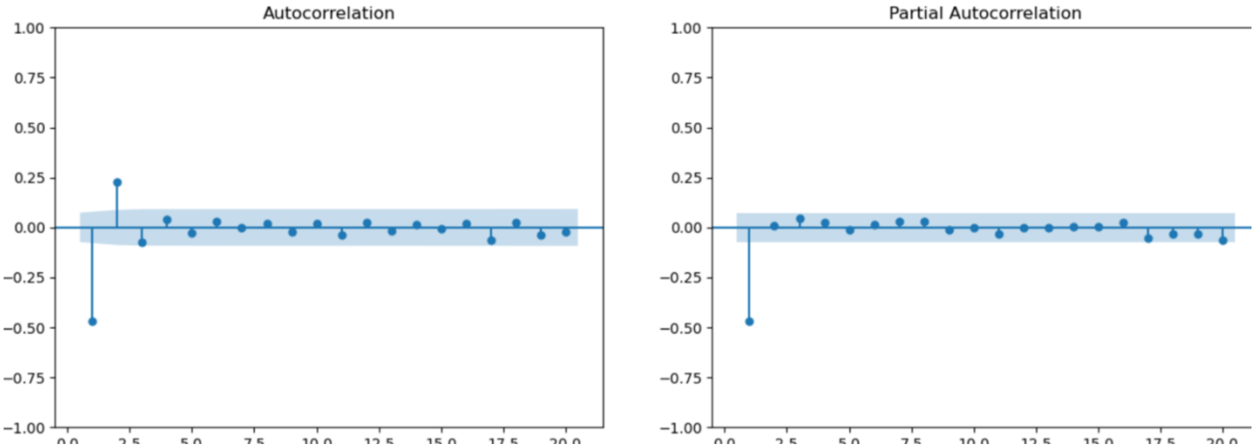


```
  diff_results = adfuller(train_diff['Revenue'])

  print(f"Test statistic: {diff_results[0]}")
  print(f"p-value: {diff_results[1]}")
```

```
Test statistic: -44.41703898145712
p-value: 0.0
```

The Augmented Dickey-Fuller test result on the differenced data yields a dramatically low test statistic and a p-value of 0.0, which conclusively rejects the null hypothesis of non-stationarity, confirming that the differenced data is stationary.

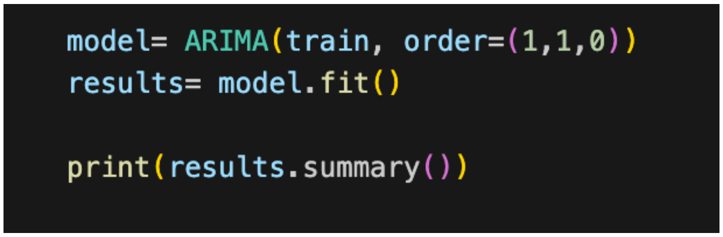
The correlogram and both ACF/PACF plots for the residuals and differenced series demonstrate no significant autocorrelation (most lags fall within the 95% confidence bands), confirming both randomness and independence of the errors.



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

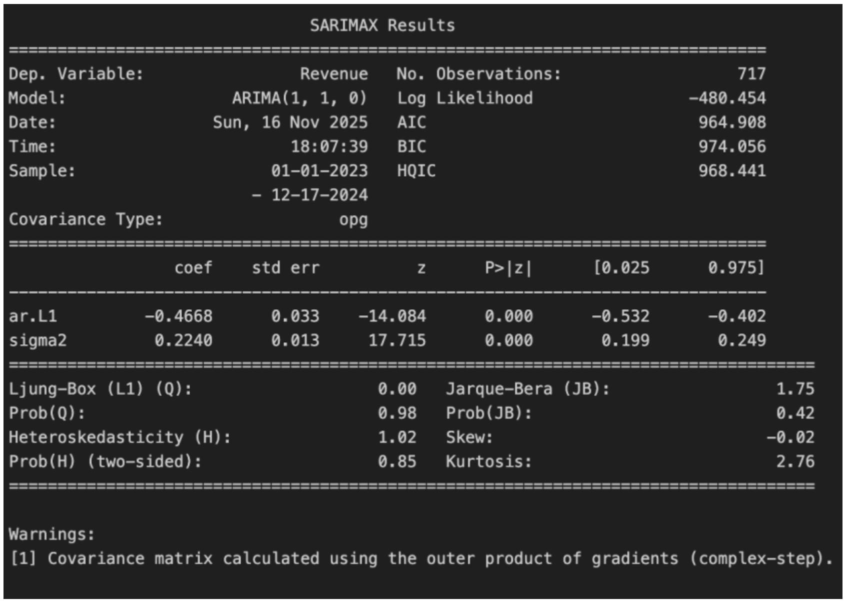
I have selected the ARIMA(1,1,0) model based on my analysis above. The order of (1,1,0) represents the p, q, and r values, respectively. I will provide further explanation on why I selected this model order in the next section.

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



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

print(results.summary())
```

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



```
                               SARIMAX Results
==============================================================================
Dep. Variable:                Revenue   No. Observations:                  717
Model:                 ARIMA(1, 1, 0)   Log Likelihood                -480.454
Date:                Sun, 16 Nov 2025   AIC                            964.908
Time:                        18:07:39   BIC                            974.056
Sample:                    01-01-2023   HQIC                           968.441
                         - 12-17-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4668      0.033    -14.084      0.000      -0.532      -0.402
sigma2         0.2240      0.013     17.715      0.000       0.199       0.249
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 1.75
Prob(Q):                              0.98   Prob(JB):                         0.42
Heteroskedasticity (H):               1.02   Skew:                            -0.02
Prob(H) (two-sided):                  0.85   Kurtosis:                         2.76
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

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

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

   - **the selection of an ARIMA model**

I selected an ARIMA model because of its strong ability to analyze time series data. There was no strong seasonal component within the data, so a SARIMA model was unnecessary. I selected the order (1,1,0), which represents the p,d, and q values, respectively. The initial time series dataset was non-stationary. I performed the first order of differencing, which made the data stationary. Therefore, the value of "d", representing the differencing order, is "1" for my model. The "p" value of 1 represents the number of autoregressive lags. This value was determined based on the ACF and PACF plots of my differenced data. The ACF plot tails off, and the PACF cuts off after lag 1. Based on this knowledge, I assigned "p" a value of 1 and "q" a value of 0, resulting in my final model order of (1, 1, 0).

   - **the prediction interval of the forecast**

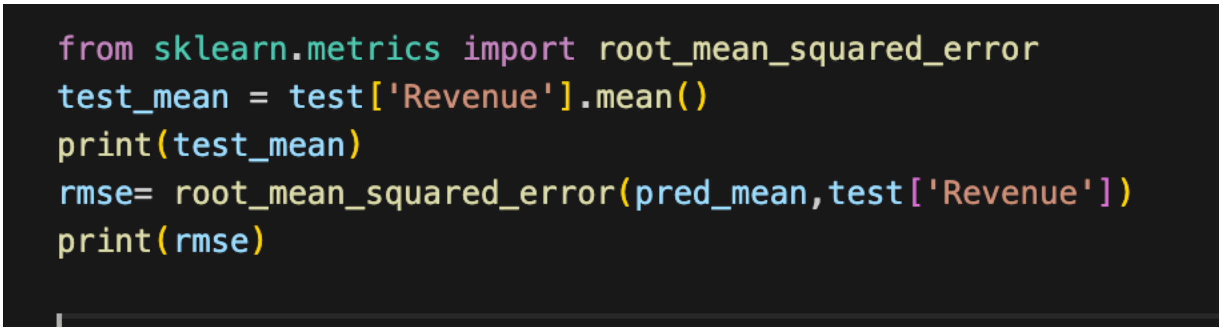
I decided to assign more data to training and leave fourteen data points for testing, which represents two weeks' worth of data.

- **a justification of the forecast length**

Initially, I performed a standard 80/20 data split. Eighty percent of the data was assigned to training, and the remaining twenty percent was held back for testing. However, my results were extremely underwhelming, with my forecast only producing a flat line.

- **the model evaluation procedure and error metric**

I evaluated the model performance with the root mean squared error (RMSE) metric. RMSE represents the average deviation of the model’s predictions from the actual values, measured in the unit of the target variable, which in this case is “Revenue”. The mean of the test set’s “Revenue” is 17.22, and the RMSE is 0.72, which is roughly 4.2% of the actual mean. This means that, on average, the predictions are off by approximately 4%, which is a relatively low error, indicating that the model fits the test data well.

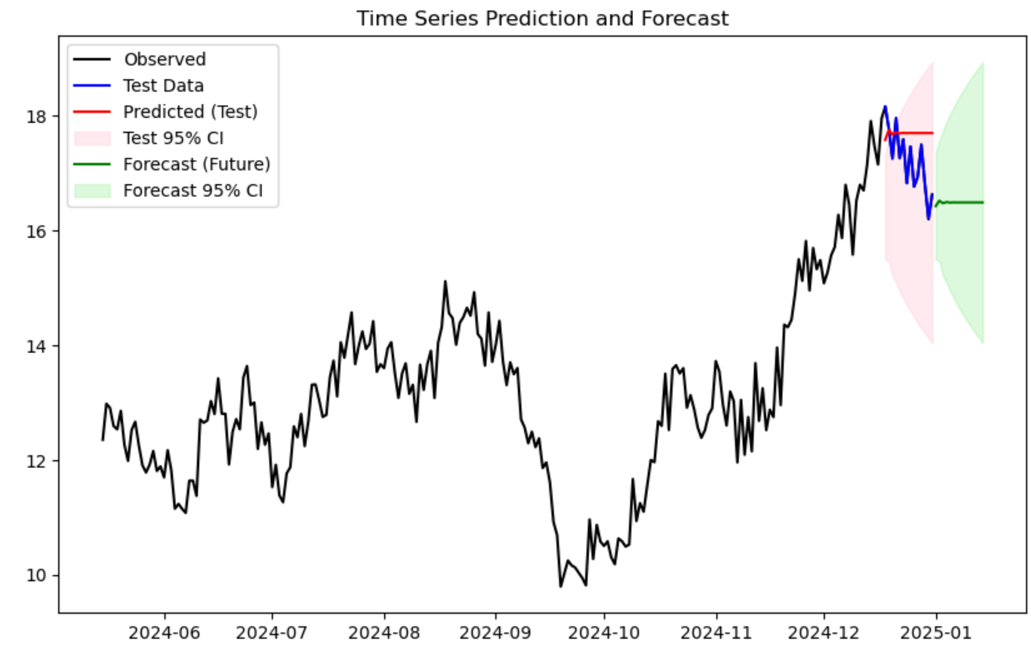


```
from sklearn.metrics import root_mean_squared_error
test_mean = test['Revenue'].mean()
print(test_mean)
rmse= root_mean_squared_error(pred_mean,test['Revenue'])
print(rmse)
```

2. Provide an annotated visualization of the forecast of the final model compared to the test set that includes the following:

- **the original output with the new prediction line and confidence cone**

- **correct labeling**



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

Based on the test and forecast results, the current ARIMA model provides fairly accurate short-term predictions, though with some lag during sharp downturns. For continued forecasting, I recommend ongoing performance monitoring, model retraining as new data becomes available, and considering advanced models or exogenous variables if forecast errors increase or patterns change.

REFERENCES

DataCamp. (2025). *ARIMA Models in Python* [Online course]. Retrieved from **https://www.datacamp.com/courses/arima-models-in-python**

Fiveable. (2025, August 22). *Stationarity and autocorrelation: Study guide* Retrieved from **https://fiveable.me/data-inference-and-decisions/unit-9/stationarity-autocorrelation/study-guide/aIbhtYchKGCgiCOi**