Time Series Forecasting

Javier Lopez

D213: Advanced Data Analytics

Table of Contents

Part I: Research Question **3**
A1. Research Question 3
A2. Objectives 3
Part II: Method Justification **4**
B. Assumptions 4
Part III: Data Preparation **4**
C1. Line Graph Visualization 4
C2. Time Step Formatting 5
C3. Stationarity Evaluation 5
C4. Data Preparation Steps 5
Part IV: Model Identification and Analysis **6**
D1. Annotated Findings with Visualizations 6
D1.1. Seasonal Component 6
D1.2. Trends 6
D1.3. Autocorrelation Function (ACF) 7
D1.4. Spectral Density 9
D1.5. Time Series Decomposition 10
D1.6. Residual of the Time Series Decomposition 11
D2. ARIMA Model Identification 11
D3. Forecasting with ARIMA 12
D4. Output and Calculations 13
D5. Code 14
Part V: Data Summary and Implications **15**
E. Findings and Assumptions Report 15
Part VI. Reporting **15**
G. Third Party Code 15
H. Sources 15

Part I: Research Question

A1. Research Question

How does Teleco's revenue vary over time, and can we forecast future revenue based on historical data?

A2. Objectives

We have established five objectives to guide our analysis and answer our research question. Firstly, we aim to identify patterns, trends, and seasonality within the revenue time series data. By understanding these patterns, we can gain insights into the underlying factors influencing revenue fluctuations. Secondly, we will develop a time series model that captures the observed patterns and seasonality in the data. This model will provide a mathematical representation of the revenue time series, allowing us to make accurate forecasts.

Next, our objective is to utilize the developed time series model to forecast future revenue. By applying the model to future time periods, we can estimate the expected revenue values, providing valuable information for future planning and decision-making. To evaluate the accuracy of our forecast, we will assess the model's performance and provide a prediction interval. This interval will give us a range within which we can expect the actual revenue values to fall, accounting for the uncertainty inherent in the forecasting process.

Finally, based on our analysis and findings, we will provide actionable insights and recommendations to Teleco. These recommendations will be derived from our understanding of the revenue patterns, the forecasted future revenue, and the evaluation of the model's accuracy. By achieving these objectives, we aim to provide Teleco with valuable insights and actionable recommendations based on a thorough analysis of the revenue time series data.

Part II: Method Justification

B. Assumptions

There are two main assumptions of a time series model. The first is stationarity, the assumption that the statistical properties of the time series, such as the mean, variance, and covariance, remain constant over time. The second is autocorrelation, the assumption that the values of the time series are correlated with the previous values.

Part III: Data Preparation

C1. Line Graph Visualization

C2. Time Step Formatting

The dataset consists of a daily time step format over a span of two years (731 days) with no apparent gaps. Each row represents a specific day, and the revenue value is recorded for that particular day.

C3. Stationarity Evaluation

We performed the Augmented Dickey-Fuller (ADF) Test to evaluate the stationarity of the telecom data. The test statistic obtained from the ADF test is -1.924612. Comparing this value with the critical values at common significance levels (-3.439352, -2.865513, and -2.568886), we found that the test statistic was less negative. Additionally, the p-value associated with the test is 0.320573, which is greater than the typical significance level of 0.05. These results indicate that there is insufficient evidence to reject the null hypothesis of non-stationarity. Therefore, based on the Dickey-Fuller Test results, the telecom company's revenue time series appears to be non-stationary.

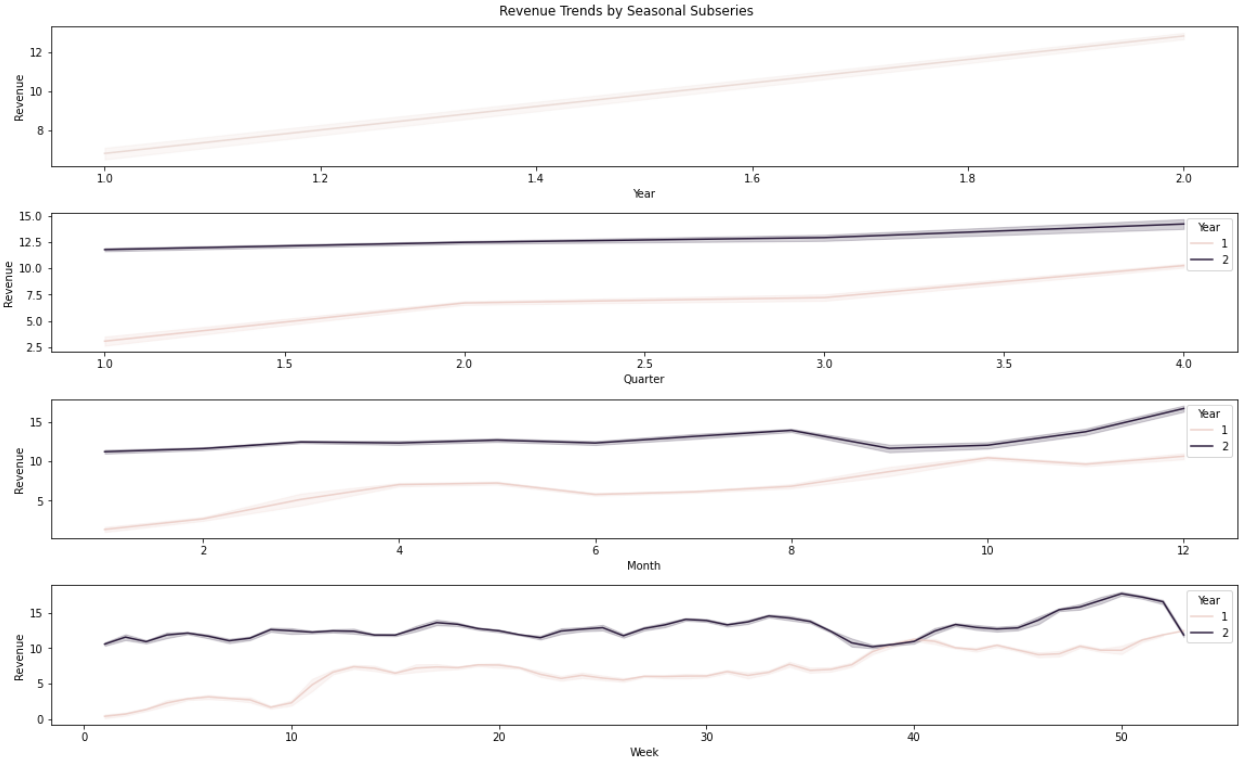
C4. Data Preparation Steps

We began preparing our data by differencing the dataset with a lag of 1 and evaluating the stationarity, an important assumption of the ARIMA models. Differencing helps remove trends or other non-stationary components. Then we split the data into training and test sets, considering the temporal order of the observations. We used all but the last 30 days of our data for training, and the remaining 30 days for testing. The training set was used for model training and the test set for evaluating the forecast.

Part IV: Model Identification and Analysis

D1. Annotated Findings with Visualizations

D1.1. Seasonal Component

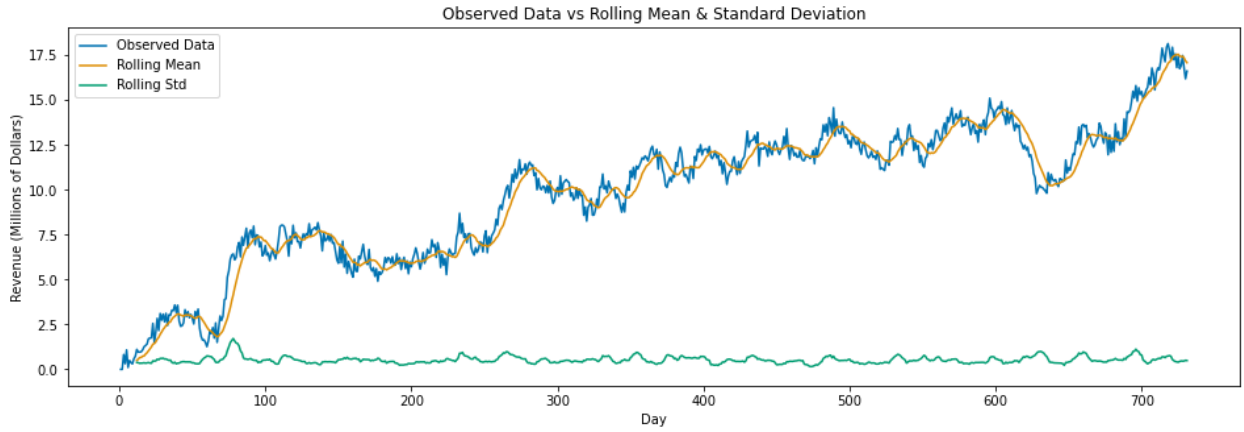


We aggregated our data into years, quarters, months, and weeks to aid us in visually assessing the presence or lack of a seasonal component. The figure above shows the visualization of each seasonal subseries for Year 1 and 2. Based on the lack of a repetitive pattern in the figure, we assessed that there was no seasonality in our observed data.

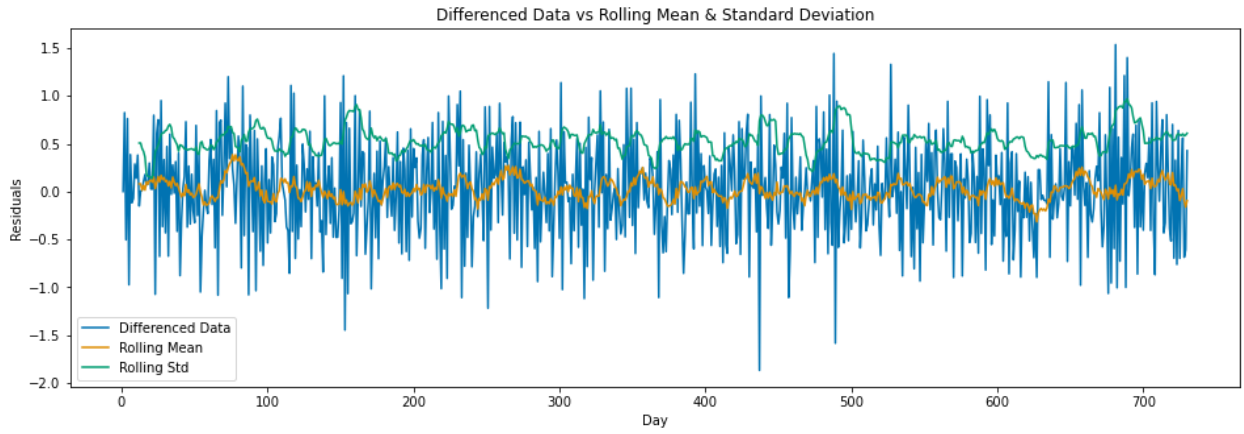
D1.2. Trends

We plotted the observed data with the rolling mean and standard deviation over a monthly window to assess the existence of any trend in our data. From the line plots we

observed a strong positive trend in the data, which confirmed the results from the ADF test. In the figure below, the rolling mean is shown in orange over the observed data shown in blue.



We calculated the first-order difference to detrend our data and plotted the rolling mean and standard deviation from the differenced data. In the line plots, shown in the figure below, we observed that the data was flattened and showed no significant trend.

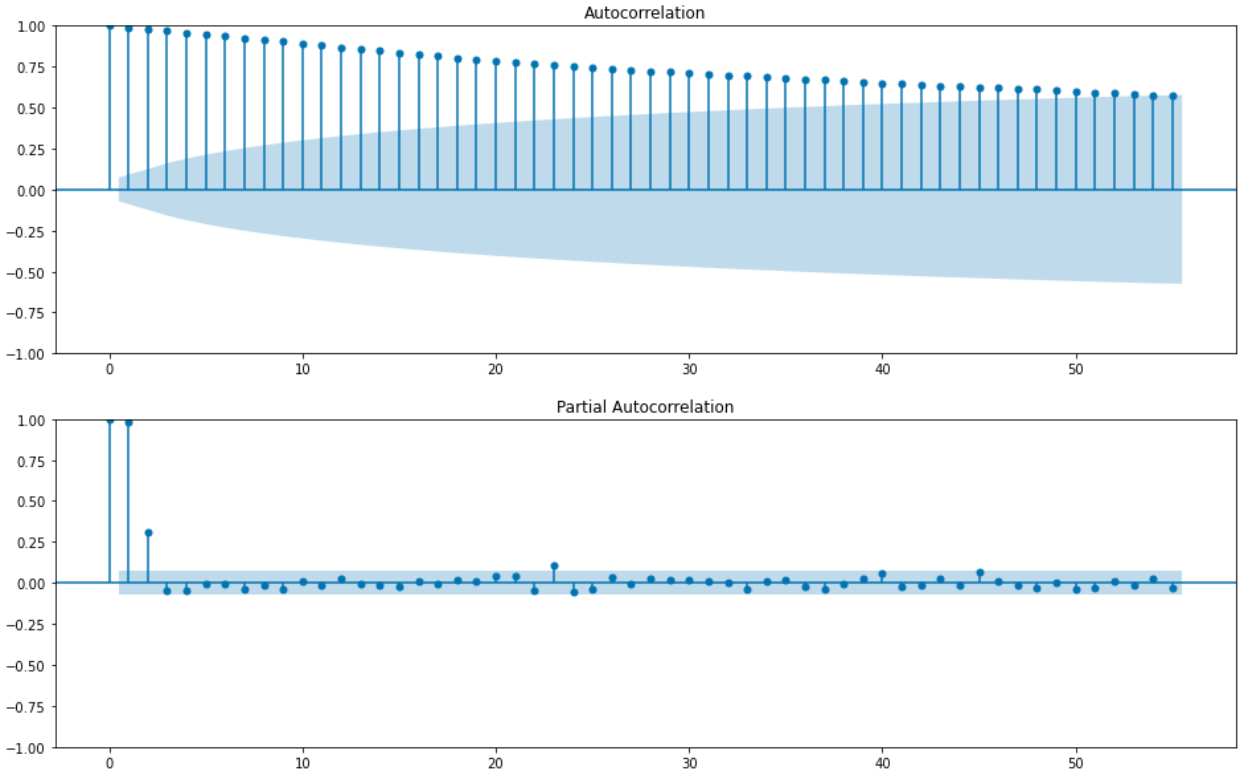


D1.3. Autocorrelation Function (ACF)

The Autocorrelation Function (ACF) is a statistical tool used to measure the correlation between a time series and its lagged (prior) values. It helps us understand the relationship between an observation and its past observation at different lags. The ACF and PACF plots are especially useful when determining the p and q values for an ARIMA model. The correlation

coefficients are on the y-axis and the lag on the x-axis. The lag represents the number of time steps between the current observation and the prior observation being correlated.

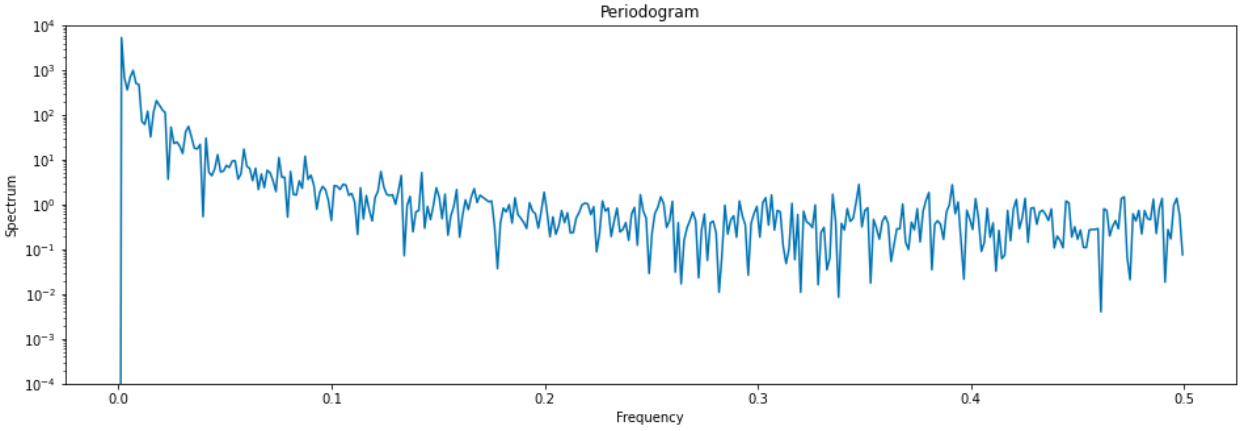
If the autocorrelation is positive at certain lags, it suggests a positive relationship between the current observation and prior observations at those lags. Conversely, if the autocorrelation is negative, it indicates a negative relationship. ACF values close to zero or within the confidence interval suggest no significant autocorrelation. From the ACF plot below we can see that the correlation coefficient values fall within the confidence intervals at around 55 lags with a correlation coefficient of about 0.50. We can also note that as the number of lags increases, the correlation coefficient decreases, suggesting that as observations are further from each other, their influence on each other also decreases.



D1.4. Spectral Density

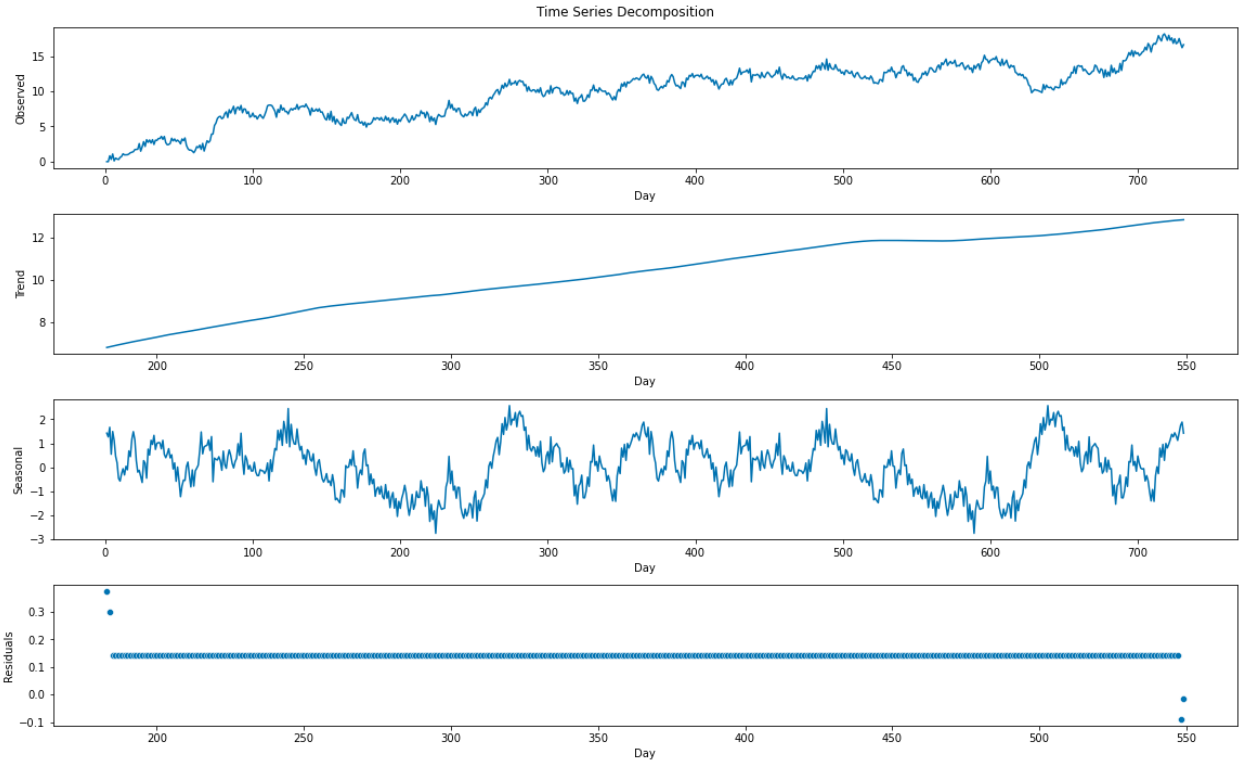
Spectral density provides a way to analyze the frequency content of a time series, representing how the variance of a time series is distributed across different frequencies. By calculating and plotting the spectral density, we can gain insights into the dominant frequencies in our data. Higher values indicate strong periodicity or cyclical patterns at certain frequencies, while low values suggest a lack of strong periodicity. The spectral density plot below, visualized in a periodogram, shows the magnitude of power for each frequency component. The x-axis represents the frequencies, measured in cycles per unit of time, and the y-axis represents the corresponding power or variance at each frequency.

The periodogram shows no significant local maximums, suggesting that there is likely no seasonality in the observed data. However, the convex shape of the trend suggests that there is a gradual change in the power spectrum over time. The convex shape of the trend indicates that the power at different frequencies increases or decreases smoothly as you move across the frequency spectrum. It suggests that there might be a gradual trend or a slow-changing pattern in the time series rather than sharp, distinct periodicity.

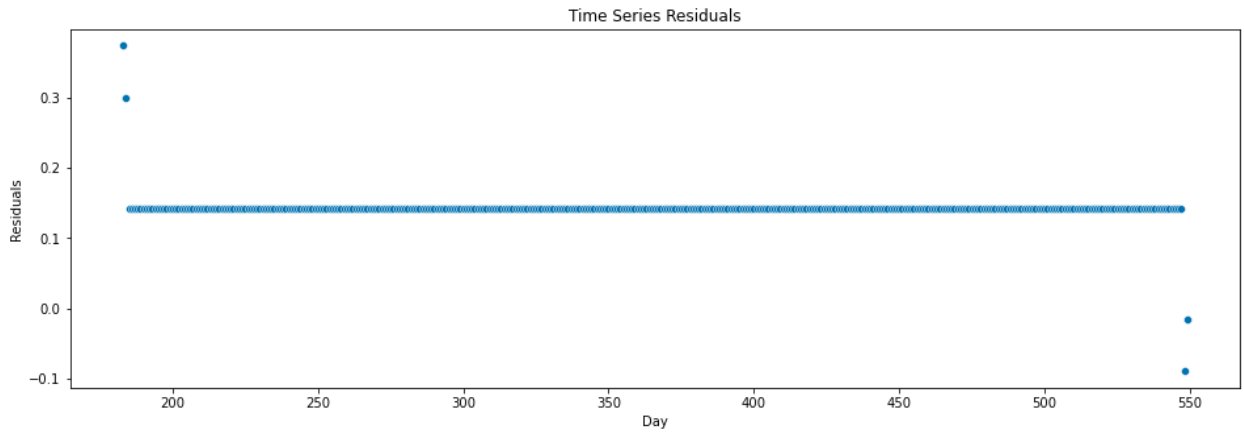


D1.5. Time Series Decomposition

Decomposing a time series allows us to separate the seasonality and trends from our data. Visualizing these components can be valuable in identifying any repetitive patterns or fluctuating periods in the observed data. The time series decomposition figure below shows the two observed periods of 365 days in our data. From the trend component we can validate our previous observations of the existence of a positive trend over time, while the seasonal component depicts the fluctuations in revenue over each period.



D1.6. Residual of the Time Series Decomposition



The residuals in a time series decomposition can provide valuable insight into underlying trends in the observed data. The time series residuals in the scatter plot above shows four isolated values and a flat line at around 0.15, showing no discernible trend in the residuals. This suggests that after accounting for the trend and seasonal components, the remaining variation in the data, represented by the residuals, does not exhibit any trend. This aligns with the expectation that the residuals should be random and free from any remaining patterns or trends.

D2. ARIMA Model Identification

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=987.305, Time=0.37 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=1162.819, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=983.122, Time=0.06 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=1019.369, Time=0.07 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=1162.139, Time=0.03 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=985.104, Time=0.08 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=985.106, Time=0.04 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=986.045, Time=0.31 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=984.710, Time=0.03 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0] intercept
Total fit time: 1.032 seconds
```

We used pmdarima's Auto-ARIMA function to search for the model with the (p, d, q) values that yielded the lowest AIC. The resulting model had order values of (1, 1, 0) with a seasonal order (P, D, Q, s) of (0, 0, 0, 0).

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	731
Model:	SARIMAX(1, 1, 0)	**Log Likelihood**	-488.561
Date:	Fri, 02 Jun 2023	**AIC**	983.122
Time:	15:09:25	**BIC**	996.901
Sample:	0	**HQIC**	988.438
	- 731		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0332	0.018	1.895	0.058	-0.001	0.068
ar.L1	-0.4692	0.033	-14.296	0.000	-0.534	-0.405
sigma2	0.2232	0.013	17.801	0.000	0.199	0.248

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	2.05
Prob(Q):	0.96	**Prob(JB):**	0.36
Heteroskedasticity (H):	1.02	**Skew:**	-0.02
Prob(H) (two-sided):	0.85	**Kurtosis:**	2.74

The value of p suggests that the current value of the time series is linearly dependent on its previous value, meaning there is autocorrelation at a lag of 1. The value of d suggests that the first-order difference of the data should be calculated to remove trends before feeding it to the ARIMA model. The q value of 0 suggests that there is no significant autocorrelation in the residuals of the time series at different lags. Additionally, the seasonal order suggests that there is no seasonal component present in the time series. The model summary above shows a p-value of 0.058 for the intercept that is above the confidence interval of 0.05 making the intercept statistically insignificant.

D3. Forecasting with ARIMA

We tested the ARIMA model by forecasting the last 30 days of our time series and comparing it to the observed values. The line plots in section E2 show the predicted means and the corresponding confidence intervals compared to the observed values in the test series. The similar predicted means and observed means in the plots suggest that the model captures the underlying patterns of the time series reasonably well.

D4. Output and Calculations

The model summary below shows that coefficients in our model have a 0.00 p-value, making them statistically significant. The ar.L1 coefficient of -0.4712 suggests that the observed value has a relatively weak negative correlation with the previous value at lag 1. This means that as the previous value increases, the current value is likely to decrease, and vice versa. The sigma2 coefficient of 0.2221 indicates how much the observed values deviate from the predicted values by the model. This value provides insight into the goodness of the fit of the model to the observed data.

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	701
Model:	ARIMA(1, 1, 0)	**Log Likelihood**	-466.837
Date:	Tue, 06 Jun 2023	**AIC**	937.673
Time:	10:03:01	**BIC**	946.776
Sample:	0	**HQIC**	941.192
	- 701		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.4712	0.033	-14.142	0.000	-0.537	-0.406
sigma2	0.2221	0.013	17.624	0.000	0.197	0.247

Ljung-Box (L1) (Q):	0.04	**Jarque-Bera (JB):**	1.44
Prob(Q):	0.85	**Prob(JB):**	0.49
Heteroskedasticity (H):	0.96	**Skew:**	-0.00
Prob(H) (two-sided):	0.76	**Kurtosis:**	2.78

We used the forecasted values and the test series to calculate the Root Mean Squared Error (RMSE) of the model. The resulting calculation, taking the square root of the mean squared error, yielded a value of 1.832. This means that on average, we can expect the predictions in our model to have a discrepancy of 1.832 units compared to the test values. The lower the RMSE, the better the model's predictive performance is considered to be.

D5. Code

```
# Use auto-arima to evaluate seasonality and find the optimal p, d, q values

stepwise_fit = auto_arima(ts, trace=True, supress_warnings=True)

# Print the model summary

stepwise_fit.summary()

# Split the train and test data

train = diff_ts[:-30]

test = diff_ts[-30:]

# Create the ARIMA model with the identified order

model = ARIMA(train.reset_index(drop=True), order=(1,1,0))

model = model.fit()

model.summary()

# Predict the last month

fcast = model.get_forecast(steps=len(test))

fcast_vals = fcast.predicted_mean

ci = fcast.conf_int()

# Calculate the mean of the test data

print(f'Test Series Mean: {test.mean()}')

# Calculate the root mean squared error

rmse = np.sqrt(mse(fcast_vals, test))

print(f'Root Mean Squared Error (RMSE): {rmse}')

# Fit the model with the original data

model2 = ARIMA(ts.reset_index(drop=True), order=(1,1,0))

model2 = model2.fit()

# Forecast the next 30 days with confidence intervals

fcast2 = model2.get_forecast(steps=30)

fcast_vals2 = fcast2.predicted_mean
```

```
ci2 = fcast2.conf_int()
```

Part V: Data Summary and Implications

E. Findings and Assumptions Report

The Findings and Assumptions Report can be found attached to this project submission (Findings and Assumption Report.html).

Part VI. Reporting

G. Third Party Code

1. Stationarity and detrending (ADF/KPSS). Stationarity and detrending (ADF/KPSS) - statsmodels 0.15.0 (+8). (n.d.). https://www.statsmodels.org/devel/examples/notebooks/generated/stationarity_detrending_adf_kpss.html

H. Sources

No additional sources were used in the creation of this project.