A1:RESEARCH QUESTION

The research question I will address is: "Can we predict future daily revenue for the organization based on historical data?" This question is highly relevant to the organization's goal of improving financial forecasting accuracy. By leveraging time series modeling techniques, we can develop a predictive model that not only informs revenue projections but also aids strategic decision-making for the executive leadership team. Insights from this model could guide decisions on optimal headcount adjustments, promotional opportunities, and operational resource planning. A well-constructed time series model will enhance the telecom company's competitiveness and foster a data-driven culture.

A2:OBJECTIVES OR GOALS

The objectives of this data analysis are as follows:

1. Utilize historical revenue data to create an accurate time series model capable of forecasting future daily revenue trends for the organization.
2. Analyze the historical revenue data to identify any underlying trends, seasonality, or patterns that could provide insights into how revenue changes over time.
3. Use the forecasts generated by the model to support the organization in making informed strategic decisions, such as workforce planning, budgeting, and promotional initiatives.

These objectives are feasible within the scope of the dataset, as they focus on predicting daily revenue, analyzing historical patterns, and applying the insights for business planning purposes.

B:SUMMARY OF ASSUMPTIONS

A time series model relies on several key assumptions to effectively analyze and predict data over time. These include:

1. **Stationarity**: A fundamental assumption is that the time series is stationary, meaning its statistical properties (e.g., mean, variance, and autocovariance) remain constant over time. Stationarity is crucial for model stability and ensures that relationships identified in the past can be generalized for forecasting. If a time series is non-stationary, transformations like differencing or logarithmic scaling are often used to achieve stationarity.
2. **Autocorrelation**: Time series models assume that observations are not independent, meaning there is a correlation between past and future values. This autocorrelation allows the model to use past values to predict future ones. Identifying significant autocorrelations helps in selecting appropriate lags and building effective models, such as ARIMA (Auto-Regressive Integrated Moving Average).
3. **Linearity and Lag Dependence**: Many time series models assume a linear relationship between past values and future predictions. These relationships are often captured in

terms of lags, where the value at a given time is dependent on previous values (lags) in a linear manner.

C2:TIME STEP FORMATTING

The time step formatting of the realization in this dataset is based on daily observations, with each time step representing a single day. The dataset includes a continuous time series without any gaps in measurement, meaning that every day from the start to the end of the sequence is represented.

- Length of the Sequence: The dataset spans 731 days, covering a period from January 1, 2021, to January 1, 2023.
- Gaps in Measurement: There are no missing dates or observations, as confirmed by inspecting the datetime index of the dataset.

This ensures the consistency and reliability of the time series for further modeling, with no need for interpolation or gap-filling techniques.

C3:STATIONARITY

To evaluate the stationarity of the time series, I performed the Augmented Dickey-Fuller (ADF) test and analyzed the visual patterns of the series. The ADF test is a statistical test used to determine whether a time series is stationary. The null hypothesis of the ADF test is that the time series is non-stationary.

- ADF Statistic: -1.924612
- p-value: 0.320573
- Critical Values:
    - 1%: -3.439
    - 5%: -2.866
    - 10%: -2.569

Since the p-value (0.320573) is greater than the significance levels (1%, 5%, and 10%), we fail to reject the null hypothesis, indicating that the time series is non-stationary in its original form.

The revenue time series plot shows increasing trends and patterns, suggesting the presence of non-stationarity. The mean and variance appear to change over time, reinforcing that the series is non-stationary. To address the non-stationarity, I applied first-order differencing. After differencing, the ADF test was conducted again:

- ADF Statistic (Differenced Series): -44.874527
- p-value: 0.000000

After differencing, the p-value is less than 0.05, which means the differenced series is stationary. This shows that the series becomes stationary after differencing, allowing us to proceed with time series modeling.

C4:STEPS TO PREPARE THE DATA

The following steps were taken to prepare the data for time series analysis:

1. Loading the Dataset:

- The dataset was loaded using `pandas`, with the `Day` column parsed as a datetime index to ensure correct time series handling.
- Command: `pd.read_csv('teleco_time_series.csv', index_col='Day', parse_dates=True)`

2. Adjusting the Date Format:

- To ensure that the dates were properly formatted, the datetime index was adjusted to have a continuous sequence from the starting point of January 1, 2021, using:
- Command: `df.index = pd.date_range(start='2021-01-01', periods=len(df), freq='D')`

3. Handling Missing Data:

- I checked for missing or duplicated values in the dataset using `.duplicated().sum()` and `.isnull().sum()` methods, confirming that there were no missing or duplicated values.

4. Descriptive Statistics:

- Descriptive statistics were generated using `df.describe()` to better understand the spread and behavior of the revenue data, including mean, standard deviation, and quantiles.

5. Stationarity Check and Differencing:

- An Augmented Dickey-Fuller (ADF) test was performed to evaluate stationarity. Since the original time series was found to be non-stationary (p-value > 0.05), I applied **first-order differencing** to remove trends and achieve stationarity.
- The differencing was done using: `df['Revenue'].diff().dropna()`
- After differencing, the ADF test was repeated, confirming that the series was stationary.

6. Data Visualization:

- The time series was plotted using `matplotlib` to visualize trends and seasonality.

- Additionally, ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots were generated to assess the autocorrelations and partial autocorrelations in the data.
- Seasonal decomposition was also performed to break down the series into trend, seasonal, and residual components using `seasonal_decompose()`.

7. Splitting the Data into Training and Test Sets:

- The data was split into **training** and **test** sets using an 80-20 ratio. The first 80% of the data was used for training, and the remaining 20% for testing.
- This was done using the `train_test_split()` function from `sklearn` with `shuffle=False` to preserve the time series order.
- The sizes of the training and test sets were confirmed:
    - Training Set: 584 observations
    - Test Set: 147 observations

8. Saving the Split Data:

- Both the training and test sets were saved as separate CSV files for easy access during model training and evaluation.

C5:PREPARED DATA SET

See Attached.

D1:REPORT FINDINGS AND VISUALIZATIONS

1. Seasonal Component:

- **Presence of Seasonality**: Based on the seasonal decomposition plot, there is a clear and regular seasonal pattern in the data. The seasonality component shows periodic fluctuations at consistent intervals, indicating the presence of a seasonal pattern in the daily revenue.
- **Visualization**: The seasonal component from the decomposition plot shows repetitive cycles, confirming the presence of seasonality in the time series.

2. Trends:

- **Presence of Trends**: The original revenue time series shows a positive trend, indicating that the revenue is generally increasing over time, with some fluctuations and dips.
- **Visualization**: The trend component in the decomposed time series plot shows an upward trajectory with some periods of stagnation and slight decline, followed by further growth.

3. Autocorrelation Function (ACF):

- **Findings**: The ACF plot shows strong autocorrelations at the first few lags, indicating that past values are highly correlated with future values. There is a gradual decrease in autocorrelation values as lag increases, suggesting that revenue at a certain day is strongly influenced by its past values.
- **Visualization**: The ACF plot shows significant spikes at the initial lags, which is typical for time series data that have both trend and seasonal components.

4. Spectral Density:

- **Findings**: The spectral density plot shows peaks at certain frequencies, which indicates the presence of cyclic behavior or periodicity in the time series. The strong peak at low frequencies further supports the idea of trend and seasonality.
- **Visualization**: The spectral density plot highlights the dominant periodic components in the data, with a clear indication of low-frequency components associated with long-term trends.

5. Decomposed Time Series:

- **Findings**: The decomposed time series shows the original revenue data broken down into three components:
    1. **Trend**: An upward-moving line with some fluctuations.
    2. **Seasonal**: A repeating pattern with a consistent interval.
    3. **Residuals**: The remaining noise after the trend and seasonality have been removed.
- **Visualization**: The decomposition plot presents each component (trend, seasonal, and residual) clearly, making it easy to observe how each part contributes to the overall time series.

6. Residuals Analysis:

- **Findings**: The residuals from the decomposed series show no clear pattern, indicating that after removing the trend and seasonality, the remaining noise is random. This confirms that the time series is well modeled by the decomposition, and there is no significant remaining trend or structure in the residuals.
- **Visualization**: The residuals plot from the decomposition shows data points scattered around zero, without any clear direction or correlation, which is a sign of a good model.

Visual Summary:

1. **Time Series Plot**:
    - Shows a positive trend with periodic fluctuations.
2. **Autocorrelation Function (ACF) Plot**:

    - Displays significant autocorrelations at early lags, confirming that past values are predictive of future values.
3. **Spectral Density Plot**:
    - Highlights the dominant low-frequency components, indicating cyclic behavior and trends.
4. **Decomposed Time Series Plot**:
    - Breaks down the series into trend, seasonality, and residuals, confirming the presence of clear seasonal patterns and a strong upward trend.
5. **Residuals Plot**:
    - Shows that the residuals are random and without trends, confirming that the trend and seasonality have been successfully removed from the data.

These visualizations confirm the presence of both trend and seasonality in the time series, and the residual analysis suggests that the model effectively captures the data's structure.

D2:ARIMA MODEL

To account for the trend and seasonality observed in the time series, an ARIMA model was identified and tuned using the following steps:

1. Differencing for Stationarity:

Since the original series was non-stationary, first-order differencing was applied to remove the trend and achieve stationarity. This process transforms the time series into a stationary one by subtracting each value from the previous one.

- Differencing applied: d = 1 (first-order differencing).

2. Identifying Autoregressive (AR) Terms:

The Partial Autocorrelation Function (PACF) plot shows significant correlation at lag 1, indicating the presence of an autoregressive component in the data.

- AR term (p): From the PACF plot, p = 1 (an AR(1) process).

3. Identifying Moving Average (MA) Terms:

The Autocorrelation Function (ACF) plot shows a significant spike at lag 1, suggesting the presence of a moving average component.

- MA term (q): From the ACF plot, q = 1 (an MA(1) process).

4. Seasonality Considerations:

Since the dataset shows clear seasonality (daily data with periodic cycles), we need to account for seasonal effects using a Seasonal ARIMA (SARIMA) model. This includes:

- Seasonal Differencing (D): Seasonal differencing was applied to remove seasonal effects.
- Seasonal AR, MA terms: The ACF and PACF plots were analyzed to identify the seasonal lags for autoregressive and moving average terms.
- Seasonal Periodicity: Based on the data (daily observations), the seasonality was set to 12 months (if relevant) or daily seasonal cycles if smaller.

5. Using Auto ARIMA for Model Selection:

To find the best combination of AR, MA, and differencing terms, auto_arima was applied. This automatically tests various combinations of parameters and selects the best one based on AIC (Akaike Information Criterion).

- Auto ARIMA suggested an ARIMA(1, 1, 1) for the non-seasonal component and Seasonal ARIMA(1, 1, 1, 12) for the seasonal component.

6. Final Model:

The final SARIMA model selected based on the data is:

- ARIMA(1, 1, 1)(1, 1, 1, 12)
- Non-seasonal components:
    - AR term (p): 1
    - Differencing (d): 1
    - MA term (q): 1
- Seasonal components:
    - Seasonal AR term (P): 1
    - Seasonal Differencing (D): 1
    - Seasonal MA term (Q): 1
    - Seasonal Periodicity (S): 12

7. Model Summary:

After fitting the model to the training data, the ARIMA summary output showed that the model effectively captures both the trend and seasonality, with parameters significantly different from zero, confirming their importance in the model.

8. Diagnostics and Forecasting:

Model diagnostics (residuals, ACF, PACF of residuals, etc.) showed that the residuals were random and uncorrelated, indicating a good fit.

- Forecasting: The model was used to generate predictions, and the forecast aligned well with the test data, capturing both the seasonal patterns and trend accurately.

D4:OUTPUT AND CALCULATIONS

The forecast results show the predicted mean values for future revenue, as well as the standard error (`mean_se`) and confidence intervals (`mean_ci_lower`, `mean_ci_upper`).

Example forecast for August 8, 2022:

- Predicted Revenue: $13.311 million
- Standard Error: 0.468 million
- 95% Confidence Interval: [$12.393 million, $14.228 million]

This forecast is extended over 147 days, providing a detailed prediction for the test period.

Confidence Intervals:

The confidence interval shows the range within which we expect the actual revenue to fall, with 95% certainty.

Example confidence interval for December 31, 2022:

- Lower Bound: $5.728 million
- Upper Bound: $21.016 million

As the forecast horizon extends, the confidence intervals widen, indicating increased uncertainty in long-term predictions. This is typical in time series forecasting.

RMSE (Root Mean Squared Error):

The RMSE measures the average error between the predicted revenue and the actual values. In this case:

- RMSE: `2.177 million`

This RMSE indicates that, on average, the forecasted revenue values deviate from the actual revenue by approximately $2.177 million. This is a reasonable level of error, considering the variability in the revenue data.

Conclusion:

- The ARIMA(1,1,1) model provides reasonable forecasts for future revenue, with a manageable level of error as indicated by the RMSE.
- The forecast is accompanied by 95% confidence intervals, which offer a range of potential outcomes for each day, reflecting the uncertainty in the predictions.
- The widening confidence intervals towards the end of the forecast horizon emphasize the growing uncertainty as time progresses.

This forecast can be used for strategic decision-making, while the RMSE indicates how closely the model's predictions align with actual revenue.

```
#Forecast

forecast = model_fit.get_forecast(steps=len(test), dynamic=True)

#Confidence interval

conf_int = forecast.conf_int()



#plotting forcast

plt.figure(figsize=(15, 7))

plt.plot(df, label='Actual')

plt.plot(forecast.predicted_mean, label='Forecast')

plt.fill_between(conf_int.index, conf_int.iloc[:, 0], conf_int.iloc[:, 1],
color='k', alpha=.25, label='Confidence Interval')

plt.xlabel('Date')

plt.ylabel('Revenue')

plt.title('Predicted Forecast')

plt.legend(loc='upper left', fontsize=8)

plt.show()

#Forecast description

print(forecast.summary_frame())

#Confidence interval description

print(conf_int)

#Values from DF

df_values = df['Revenue'].iloc[len(df)-len(test):].values
```

```
#RMSE

rmse = np.sqrt(mean_squared_error(df_values, forecast.predicted_mean))

print(rmse)
```

E1:RESULTS

Selection of an ARIMA Model:

- The ARIMA(1,1,1) model was selected based on the analysis of the time series. The decision to use an ARIMA(1,1,1) model was driven by the following observations:
    - Autoregressive (AR) term (p = 1): Significant lag-1 autocorrelation in the data, as observed in the PACF plot.
    - Differencing (d = 1): The time series was non-stationary, as indicated by the Augmented Dickey-Fuller test, and first-order differencing was necessary to make it stationary.
    - Moving Average (MA) term (q = 1): The ACF plot suggested a moving average component at lag-1.
- The model was evaluated using AIC (Akaike Information Criterion), and the ARIMA(1,1,1) provided a good balance between model complexity and accuracy.

Prediction Interval of the Forecast:

- The prediction intervals for the forecast were calculated at a 95% confidence level, indicating the range within which the actual revenue is expected to fall with 95% certainty.
- As seen in the forecast results, the confidence intervals start narrow and widen as the forecast horizon extends, indicating increasing uncertainty in long-term predictions. For example:
    - August 8, 2022: [$12.393 million, $14.228 million]
    - December 31, 2022: [$5.728 million, $21.016 million]
- This widening reflects the inherent uncertainty as the model forecasts further into the future.

Justification of the Forecast Length:

- The forecast was generated for a period of 147 days (the length of the test set, which represents 20% of the data).

- This length is justified because it provides a meaningful period for short-term planning while allowing for evaluation against real observed data in the test set. It balances the need for accuracy in near-term forecasts with the goal of capturing seasonal patterns.

Model Evaluation Procedure and Error Metric:

- The model was evaluated using the Root Mean Squared Error (RMSE) metric. RMSE measures the average magnitude of the error between the predicted values and the actual values in the test set, providing a clear measure of how well the model performs.
- The RMSE for the ARIMA(1,1,1) model was 2.177 million, indicating that, on average, the forecasted revenue deviated from the actual revenue by about $2.177 million.
- The model's residuals were examined using diagnostic plots, confirming that the residuals were random and normally distributed, with no significant autocorrelation, indicating a good fit.

Key Findings:

- The ARIMA(1,1,1) model was a suitable choice for forecasting the telecom company's revenue based on historical data. It effectively captured the trend and autocorrelation in the time series.
- The prediction intervals provide a reasonable range for forecasting future values, with increased uncertainty for long-term predictions.
- The forecast length of 147 days (about five months) allowed for a detailed evaluation of the model's short-term performance while considering the company's operational needs.
- The RMSE metric provided a clear measure of the model's performance, with a manageable error margin for strategic decision-making.

E3:RECOMMENDATION

Based on the results of the ARIMA(1,1,1) model and the revenue forecasts, I recommend the following course of action:

1. Leverage Forecasts for Strategic Decision-Making:

The ARIMA(1,1,1) model provides actionable forecasts for the next 147 days, with a reasonable RMSE of 2.177 million. These forecasts can be used to support strategic decisions, particularly in the following areas:

- Revenue Projections: Use the predicted revenue to inform budget planning and cash flow management. The confidence intervals around the forecast allow the company to prepare for best- and worst-case scenarios.
- Resource Allocation: The forecasted revenue can help in determining optimal headcount adjustments (increases or reductions) or resource allocation during periods of expected higher or lower revenue.

- Sales and Marketing Strategy: During periods where a decline in revenue is forecasted, the company can proactively adjust sales and marketing efforts to mitigate potential losses.

2. Monitor Forecast Accuracy Regularly:

Given that the prediction intervals widen as the forecast extends further into the future, it is important to regularly update the model with new data. This will improve the accuracy of the forecasts and ensure that the model stays relevant in the face of changing trends.

- Action: Implement a monthly review process where the ARIMA model is retrained using the most recent data. This will allow for continuous refinement of the forecasts, ensuring that they remain reliable for decision-making.

3. Plan for Uncertainty:

The forecast confidence intervals highlight increasing uncertainty for long-term forecasts, particularly after several months. The company should be prepared for a range of outcomes, especially in the later stages of the forecast period.

- Action: Develop contingency plans based on the lower and upper bounds of the revenue forecast. For instance, if revenue is forecasted to decline below a certain threshold, have a plan for cost-cutting measures, such as reducing discretionary expenses or delaying non-essential projects.

4. Explore Additional Revenue Growth Opportunities:

Based on the insights from the forecast, the company could take steps to explore opportunities for revenue growth:

- Product Promotions: Identify periods where revenue may be lower and launch targeted promotions or discounts to stimulate demand.
- Service Enhancements: Consider adding or improving services that may enhance customer retention and drive additional revenue, especially during predicted slow periods.

5. Adopt Data-Driven Culture:

The success of this forecasting model demonstrates the value of data-driven decision-making. Encourage the executive leadership team (ELT) and other departments to regularly rely on these forecasts and incorporate data analytics into other business processes.

- Action: Build internal data analytics capabilities and provide training for decision-makers on how to interpret and use forecast data effectively.

The ARIMA(1,1,1) model offers valuable insights into future revenue trends. By leveraging these forecasts, the company can enhance strategic planning, optimize resource allocation, and proactively address potential revenue fluctuations. Regular model updates and the adoption of a data-driven approach will ensure the company remains competitive and agile in its decision-making.

G:SOURCES FOR THIRD-PARTY CODE

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

dddd_y. (2020, August 1). *Unable to import auto_arima from PMDARIMA Jupyter*. Stack Overflow. https://stackoverflow.com/questions/63474036/unable-to-import-auto-arima-from-pmdarima-jupyter

Iordanova, T. (n.d.). *An introduction to non-stationary processes*. Investopedia. https://www.investopedia.com/articles/trading/07/stationary.asp

Prabhakaran, S. (2022, April 4). *Augmented dickey-fuller (ADF) test - must read guide - ml+*. Machine Learning Plus. https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

Verma, Y. (2024, August 2). *Quick way to find P, D and Q values for Arima*. AIM. https://analyticsindiamag.com/ai-mysteries/quick-way-to-find-p-d-and-q-values-for-arima/