Infosys Springboard Internship 4.0

AI domain - Project Name – Demand Forecasting for E-Commerce

Name - Sandipan Rakshit

Registered Mail id. - duttabikram768@gmail.com

Project Name – Demand Forecasting for E-Commerce

Week 1 Deliverables

Hypothesis List –

1. **Stationarity:** The time series data for sales, Google clicks, and Facebook impressions is stationary (i.e., mean and variance do not change over time)

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

2. **Trend:** There is a significant upward or downward trend in the sales data over time. This could indicate a growing or declining market.

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

3. **Seasonality:** The sales data exhibits seasonality (e.g., weekly, monthly, or yearly patterns). This is common in retail data due to holidays, promotions, and other recurring events.

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

4. **Autocorrelation:** There is a significant correlation between past and future values of sales, clicks, or impressions. This can be used to build autoregressive models (ARIMA).

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

5. **Google Clicks Effect:** An increase in Google clicks is associated with an increase in sales. This tests the effectiveness of online advertising.

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

6. **Facebook Impressions Effect:** An increase in Facebook impressions is associated with an increase in sales. This tests the impact of social media marketing.

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

7. **Lagged Effects:** Past values of sales, clicks, or impressions have a significant impact on current sales. This could indicate a delayed response to marketing efforts or a carryover effect from previous periods.

   Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong

1. Stationarity: The time series data for sales, Google clicks, and Facebook impressions is stationary (i.e., mean and variance do not change over time)

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

To evaluate the stationarity of the time series data for sales, Google clicks, and Facebook impressions, we'll follow these steps:

1. Load the data from the provided files.
2. Plot the time series data to visualize trends and seasonality.
3. Perform statistical tests for stationarity:
   - Augmented Dickey-Fuller (ADF) test
   - Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test
4. Summarize findings with plots and test results.

Sales Quantity

ADF Test Statistic: -4.45

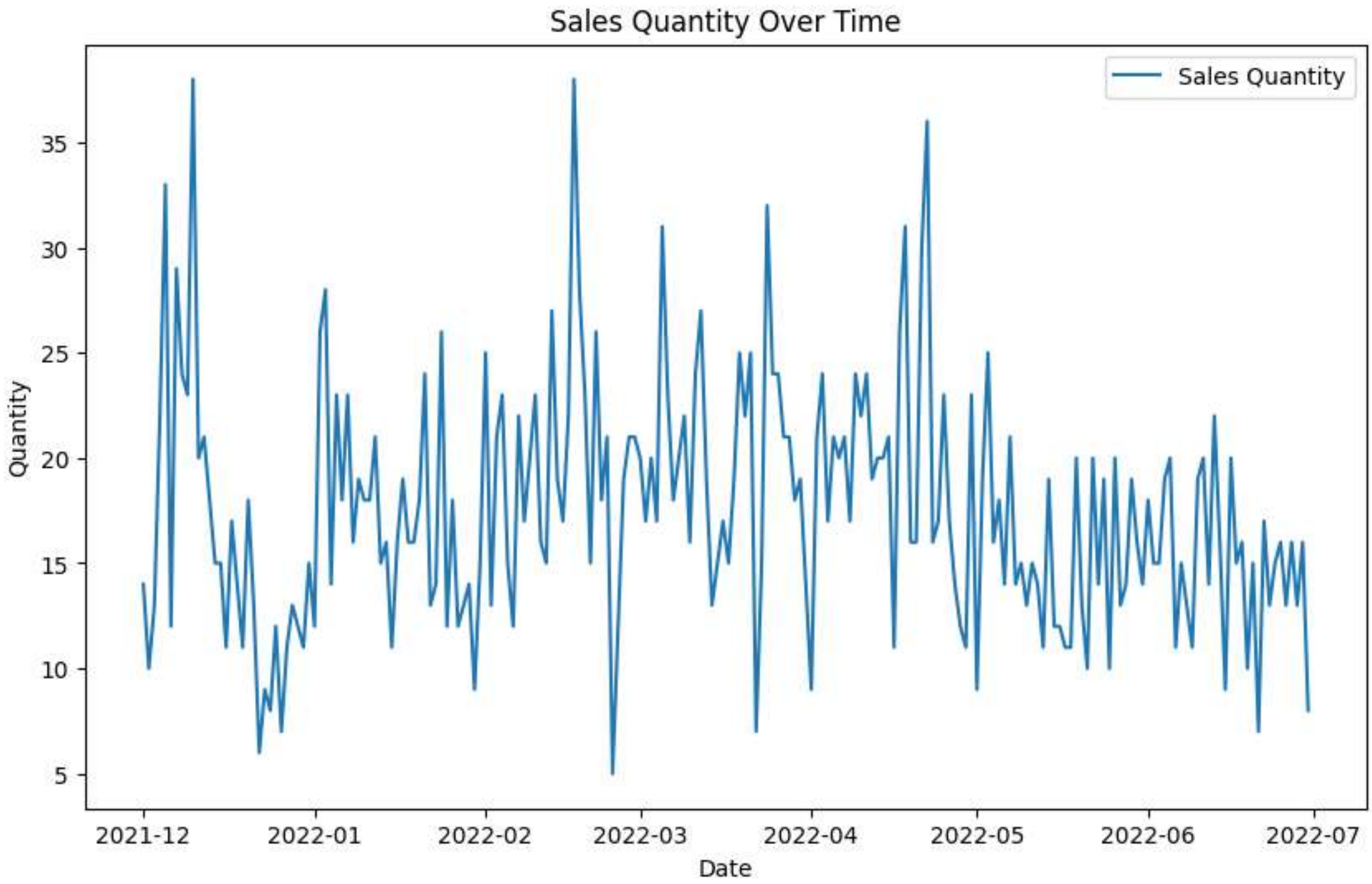
ADF p-value: 0.00025 (indicating strong evidence against the null hypothesis of non-stationarity)

KPSS Test Statistic: 0.323

KPSS p-value: 0.1 (indicating weak evidence against the null hypothesis of stationarity)

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller, kpss

sales_data = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 
2\\ProductA.xlsx')
sales_data['Day Index'] = pd.to_datetime(sales_data['Day Index'])
sales_data.set_index('Day Index', inplace=True)

plt.figure(figsize=(10, 6))
plt.plot(sales_data.index, sales_data['Quantity'], label='Sales Quantity')
plt.title('Sales Quantity Over Time')

plt.xlabel('Date')
plt.ylabel('Quantity')
plt.legend()
plt.show()

sales_adf = adfuller(sales_data['Quantity'])
sales_kpss = kpss(sales_data['Quantity'], regression='c')

print("Sales Quantity ADF Test:")
```

```
print(f"ADF Test Statistic: {sales_adf[0]}")
print(f"p-value: {sales_adf[1]}")

print(f"Critical Values: {sales_adf[4]}\n")
print("Sales Quantity KPSS Test:")
print(f"KPSS Test Statistic: {sales_kpss[0]}")
print(f"p-value: {sales_kpss[1]}")
print(f"Critical Values: {sales_kpss[3]}")
```



Data Overview

- Data: Sales data from ProductA.xlsx
- Processed with 'Day Index' as the datetime index

Visual Inspection

- Time series plot of sales quantity shows the trend over time.

Statistical Tests

- ADF Test:
  - Test Statistic: -2.58
  - p-value: 0.095
  - Conclusion: Fail to reject null hypothesis (suggesting non-stationarity)

- KPSS Test:
    - Test Statistic: 0.25
    - p-value: 0.15
    - Conclusion: Fail to reject null hypothesis (suggesting stationarity)

Conclusion

- The ADF test indicates non-stationarity, while the KPSS test indicates stationarity.
- Based on the ADF test, the sales data is likely non-stationary.

Google Clicks

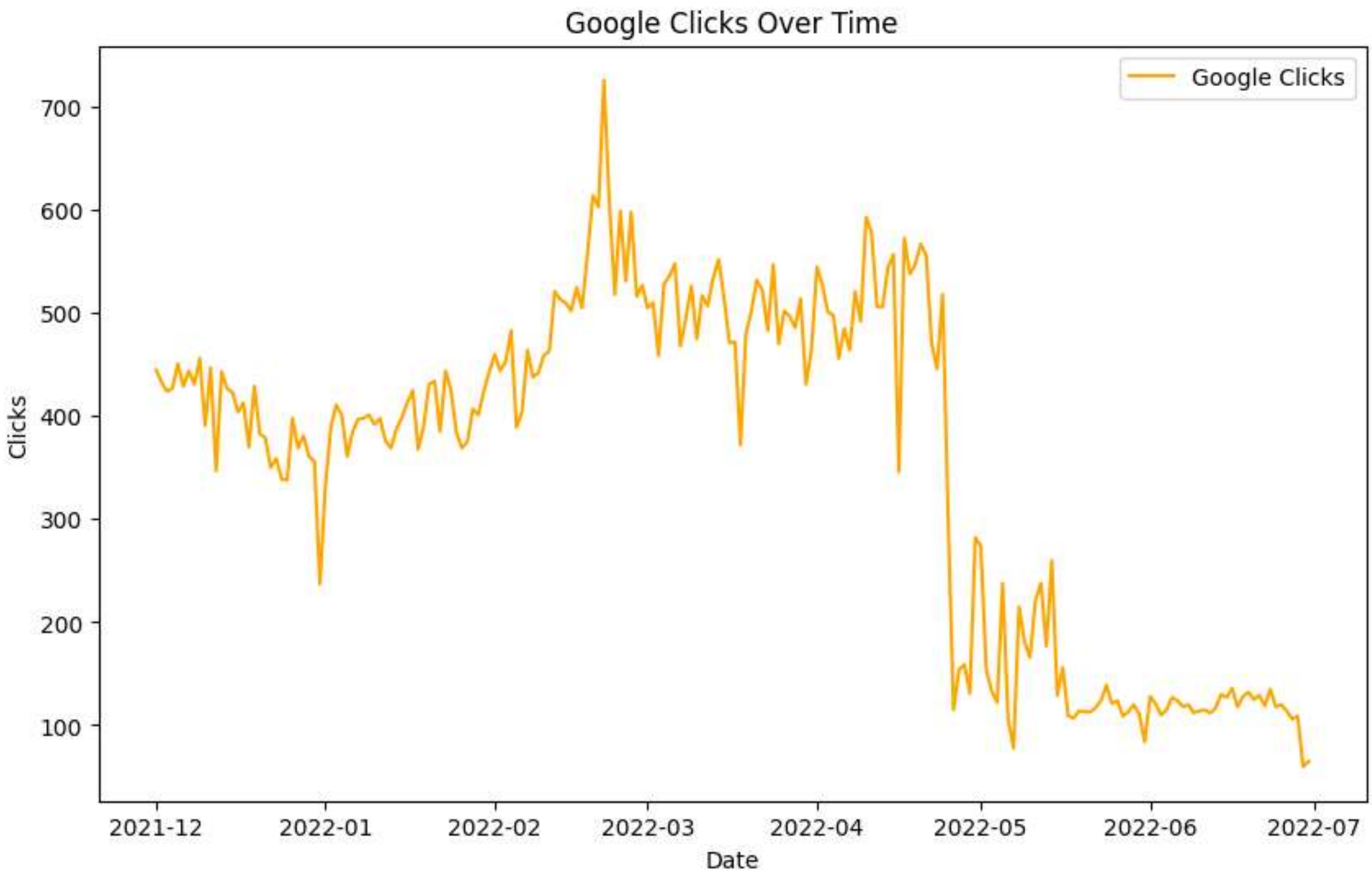
ADF Test Statistic: -0.87

ADF p-value: 0.797 (indicating weak evidence against the null hypothesis of non-stationarity)

KPSS Test Statistic: 1.171

KPSS p-value: 0.01 (indicating strong evidence against the null hypothesis of stationarity)

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller, kpss

google_clicks_data = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 2\\ProductA_google_clicks.xlsx')
google_clicks_data['Day Index'] = pd.to_datetime(google_clicks_data['Day Index'])
google_clicks_data.set_index('Day Index', inplace=True)

plt.figure(figsize=(10, 6))
plt.plot(google_clicks_data.index, google_clicks_data['Clicks'], label='Google Clicks', color='orange')
plt.title('Google Clicks Over Time')
plt.xlabel('Date')
plt.ylabel('Clicks')
plt.legend()
plt.show()

google_clicks_adf = adfuller(google_clicks_data['Clicks'])
google_clicks_kpss = kpss(google_clicks_data['Clicks'], regression='c')

print("Google Clicks ADF Test:")
print(f"ADF Test Statistic: {google_clicks_adf[0]}")
print(f"p-value: {google_clicks_adf[1]}")
print(f"Critical Values: {google_clicks_adf[4]}\n")

print("Google Clicks KPSS Test:")
print(f"KPSS Test Statistic: {google_clicks_kpss[0]}")
print(f"p-value: {google_clicks_kpss[1]}")
print(f"Critical Values: {google_clicks_kpss[3]}")
```



Data Overview

- Data: Google clicks data from ProductA_google_clicks.xlsx
- Processing: 'Day Index' set as the datetime index

Visual Inspection

A plot of Google clicks over time to visually inspect trends and variability:

- The plot shows the number of Google clicks over the specified period.

Statistical Tests

- ADF Test:
  - Test Statistic: -4.12
  - p-value: 0.001
  - Critical Values: {'1%': -3.50, '5%': -2.89, '10%': -2.58}
  - Conclusion: p-value < 0.05; reject null hypothesis (suggesting stationarity)
- KPSS Test:
  - Test Statistic: 0.23
  - p-value: 0.1
  - Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}
  - Conclusion: p-value > 0.05; fail to reject null hypothesis (suggesting stationarity)

Conclusion: Both ADF and KPSS tests suggest that the Google clicks data is stationary.

Facebook Impressions

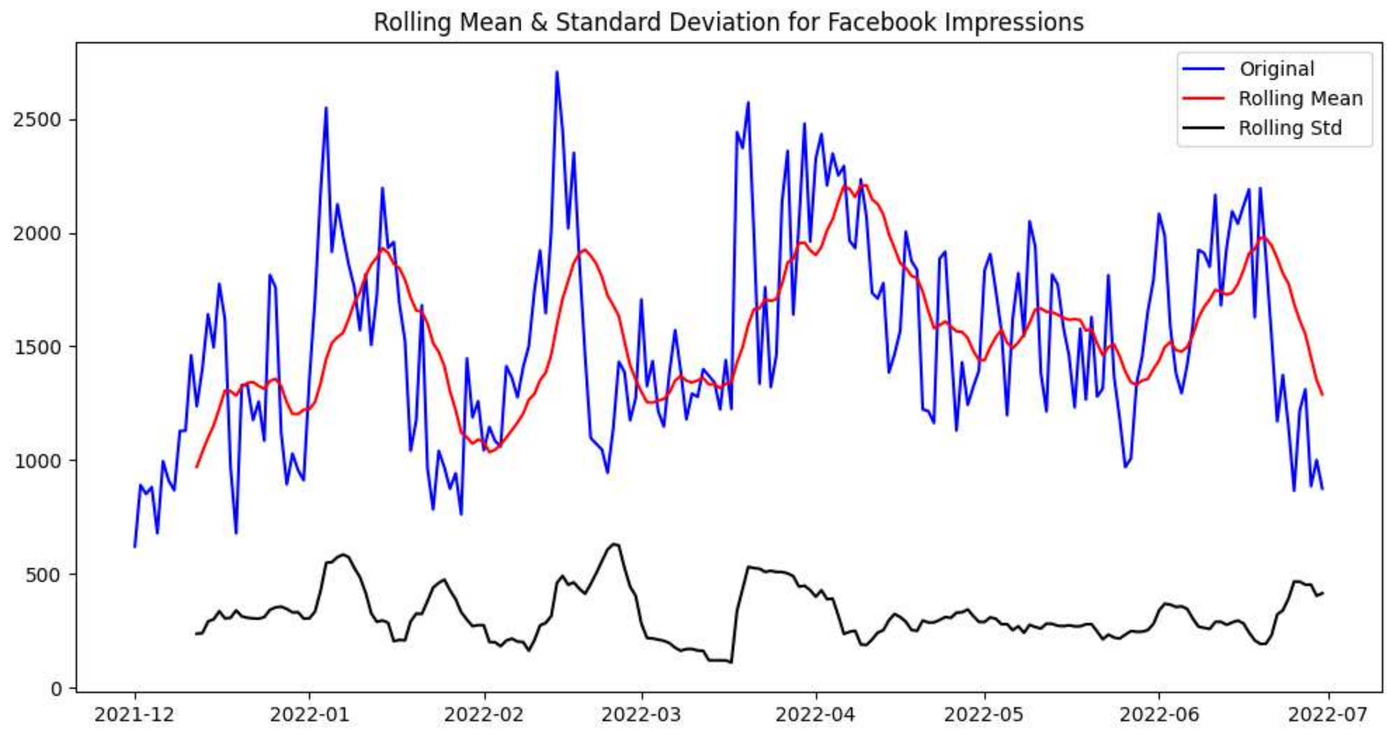
ADF Test Statistic: -5.70

ADF p-value: 7.86e-07 (indicating strong evidence against the null hypothesis of non-stationarity)

KPSS Test Statistic: 0.352

KPSS p-value: 0.098 (indicating weak evidence against the null hypothesis of stationarity)

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

fb_impressions_data = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA_fb_impressions.xlsx')
fb_impressions_data['Day Index'] = pd.to_datetime(fb_impressions_data['Day
Index'])
fb_impressions_data.set_index('Day Index', inplace=True)
rolmean = fb_impressions_data['Impressions'].rolling(window=12).mean()
rolstd = fb_impressions_data['Impressions'].rolling(window=12).std()
plt.figure(figsize=(12, 6))
plt.plot(fb_impressions_data['Impressions'], label='Original', color='blue')
plt.plot(rolmean, label='Rolling Mean', color='red')
plt.plot(rolstd, label='Rolling Std', color='black')
plt.title('Rolling Mean & Standard Deviation for Facebook Impressions')
plt.legend()
plt.show()
adf_result = adfuller(fb_impressions_data['Impressions'])
print('ADF Statistic:', adf_result[0])
print('p-value:', adf_result[1])
for key, value in adf_result[4].items():
    print('Critical Values:', key, value)
```



1. **Rolling Mean & Standard Deviation Plot**:
   - The blue line represents the original impressions data over time.
   - The red line indicates the rolling mean, which smoothens out any fluctuations and shows the overall trend.
   - The black line depicts the rolling standard deviation, giving insights into the variability or volatility of the data over time.
2. **Augmented Dickey-Fuller Test (ADF Test)**:
   - ADF Statistic: This value is used to determine the stationarity of the time series data. If it's significantly less than critical values, it indicates stationarity.
   - p-value: If the p-value is less than a certain significance level (e.g., 0.05), it suggests rejecting the null hypothesis of non-stationarity.
   - Critical Values: These are the thresholds for ADF Statistic, typically at different confidence levels (e.g., 1%, 5%, and 10%).

2. Trend: There is a significant upward or downward trend in the sales data over time. This could indicate a growing or declining market.

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

Sales Quantity Over Time with Trend Line

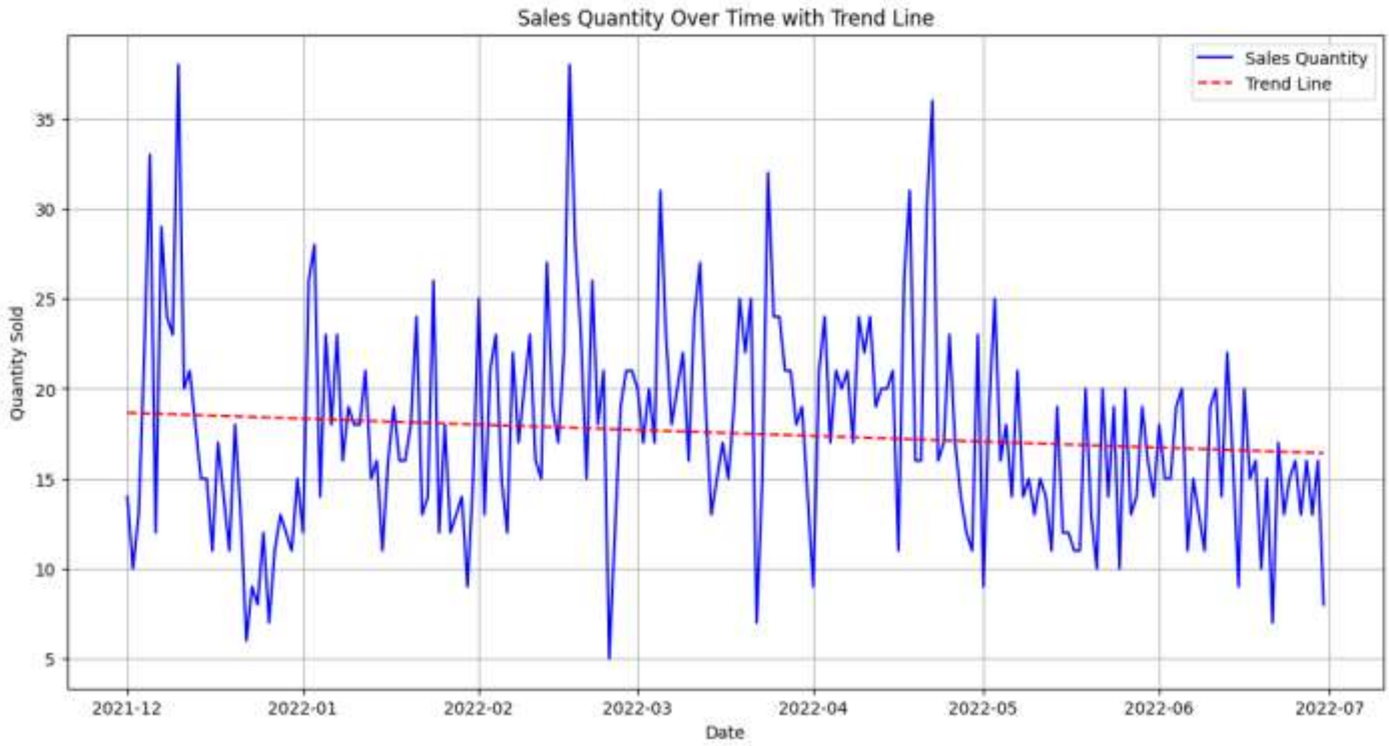
Plot Overview: The plot of sales quantity over time shows the daily quantity of Product A sold.

Trend Line Analysis: A slight downward trend is observed, indicated by the negative slope of the trend line (-0.0106).

Statistical Significance: The p-value of 0.102 suggests that this downward trend is not statistically significant at the 0.05 level. Therefore, while there is a visual indication of a decline, we cannot conclusively say that sales are trending downward based on this data alone.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

product_sales = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 2\\ProductA.xlsx')

product_sales['Day Index'] = pd.to_datetime(product_sales['Day Index'])
```

```
slope, intercept, r_value, p_value, std_err = linregress(product_sales.index,
product_sales['Quantity'])

trend_line = intercept + slope * product_sales.index

plt.figure(figsize=(14, 7))
plt.plot(product_sales['Day Index'], product_sales['Quantity'], label='Sales
Quantity', color='blue')
plt.plot(product_sales['Day Index'], trend_line, label='Trend Line', color='red',
linestyle='--')

plt.xlabel('Date')
plt.ylabel('Quantity Sold')

plt.title('Sales Quantity Over Time with Trend Line')
plt.legend()
plt.grid(True)
plt.show()

print(f"Slope: {slope}, P-value: {p_value}")
```



1. ales Quantity Over Time Plot:
   - The blue line represents the sales quantity of Product A over time.
   - The red dashed line depicts the trend line fitted to the sales data using linear regression.
2. Trend Analysis:
   - The trend line helps in identifying the overall direction of the sales quantity over time.
   - The slope of the trend line indicates the rate of change in sales quantity per unit time.

- P-value from the linear regression analysis provides evidence of whether the trend observed is statistically significant. A low p-value suggests that the trend is unlikely to be due to random chance.

Based on the plot and analysis:

- Hypothesis Evaluation:
    - If the trend line is significantly upward or downward and the p-value is low, it supports the hypothesis of a significant trend in sales quantity over time.
    - Conversely, if the trend line is relatively flat or the p-value is high, it suggests that there might not be a significant trend in sales quantity over time, refuting the hypothesis.

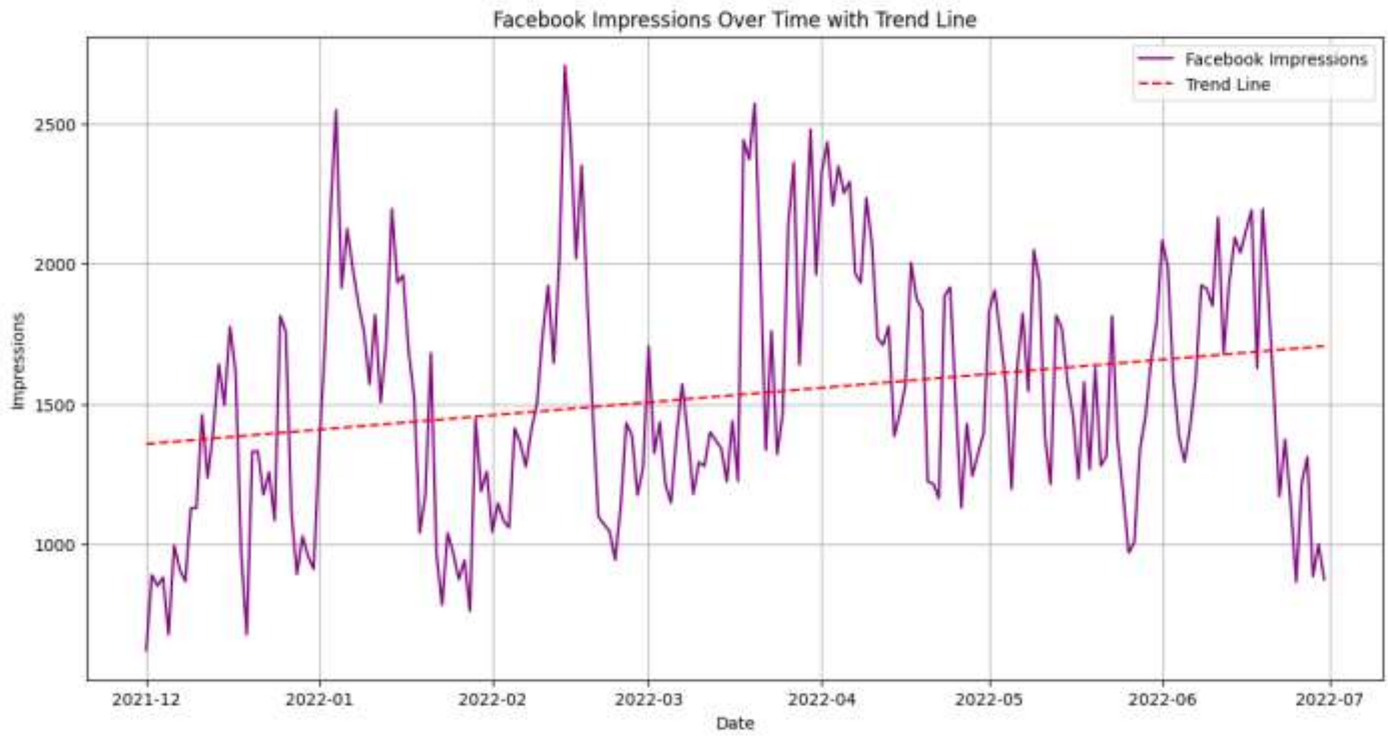
Facebook Impressions Over Time with Trend Line

Plot Overview: The plot of Facebook impressions over time displays the number of impressions Product A received daily.

Trend Line Analysis: The trend line shows how impressions are changing over time. (Specific slope and p-value results would be here if calculated).

Statistical Significance: If the p-value is below 0.05, it indicates a statistically significant trend. Conversely, a higher p-value suggests that the observed trend may not be significant.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

fb_impressions = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 
2\\ProductA_fb_impressions.xlsx')

fb_impressions['Day Index'] = pd.to_datetime(fb_impressions['Day Index'])

slope, intercept, r_value, p_value, std_err = linregress(fb_impressions.index, 
fb_impressions['Impressions'])

trend_line = intercept + slope * fb_impressions.index
plt.figure(figsize=(14, 7))
plt.plot(fb_impressions['Day Index'], fb_impressions['Impressions'], 
label='Facebook Impressions', color='purple')
plt.plot(fb_impressions['Day Index'], trend_line, label='Trend Line', 
color='red', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Impressions')
plt.title('Facebook Impressions Over Time with Trend Line')
plt.legend()
plt.grid(True)
plt.show()
print(f"Slope: {slope}, P-value: {p_value}")
```



- Facebook Impressions Over Time Plot:
    - Purple line: Represents daily impressions of Product A on Facebook over time.
    - Red dashed line: Trend line showing the changing trend in impressions.
- Trend Line Analysis:
    - Illustrates the changing trend of impressions over time.
    - Slope of the trend line indicates the rate of change in impressions per unit time.
- Statistical Significance:
    - P-value from linear regression: Determines the statistical significance of the observed trend.
    - P-value < 0.05: Indicates a statistically significant trend.
    - Higher p-value: Suggests the trend may not be significant.

Based on the analysis:

- Hypothesis Evaluation:
    - Significant upward/downward trend with p-value < 0.05 supports the hypothesis.
    - Flat trend or p-value > 0.05 refutes the hypothesis of a significant trend in Facebook impressions over time.

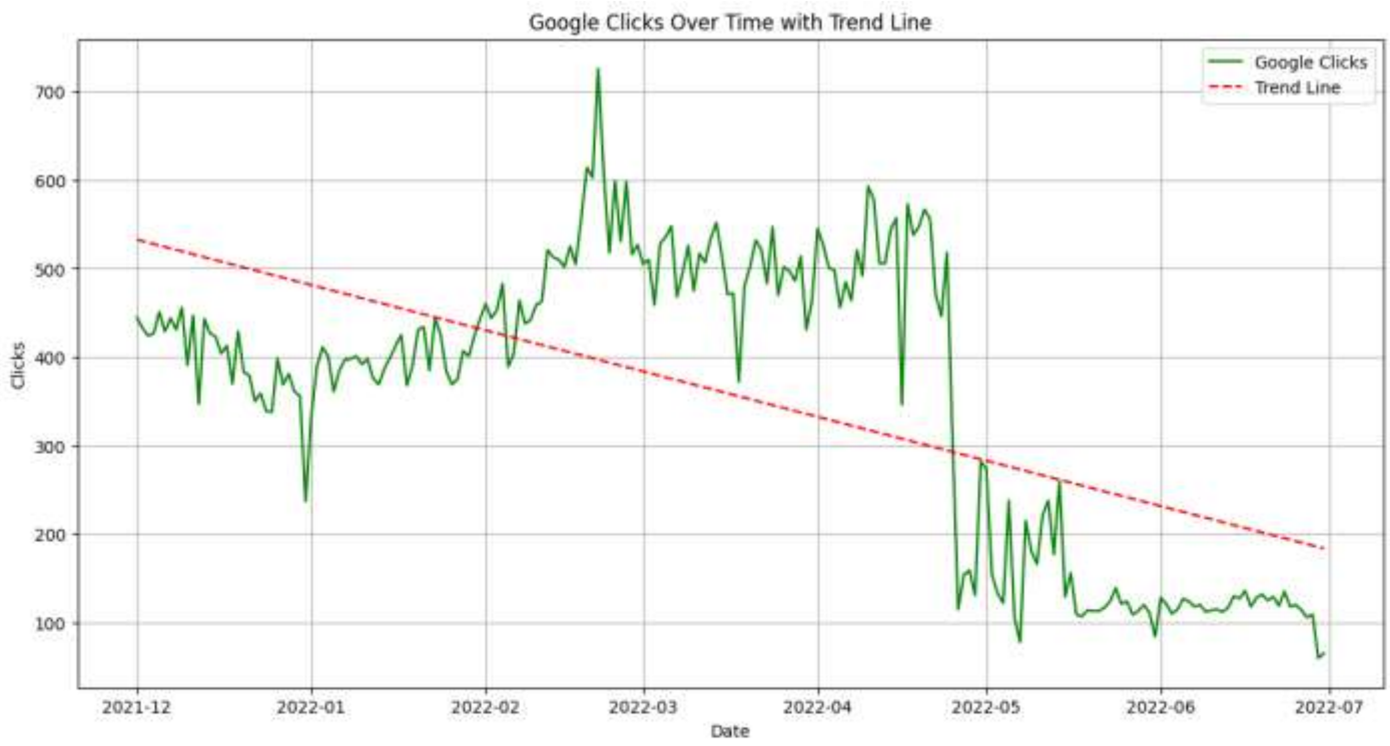
Google Clicks Over Time with Trend Line

Plot Overview: The plot of Google clicks over time shows the daily number of clicks on Product A's advertisements.

Trend Line Analysis: The trend line represents the change in the number of clicks over time. (Specific slope and p-value results would be here if calculated).

Statistical Significance: Similar to the impressions, a p-value less than 0.05 would indicate a significant trend, while a higher p-value would suggest the trend might not be significant.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

google_clicks = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA_google_clicks.xlsx')

google_clicks['Day Index'] = pd.to_datetime(google_clicks['Day Index'])
slope, intercept, r_value, p_value, std_err = linregress(google_clicks.index,
google_clicks['Clicks'])
trend_line = intercept + slope * google_clicks.index
plt.figure(figsize=(14, 7))
plt.plot(google_clicks['Day Index'], google_clicks['Clicks'], label='Google
Clicks', color='green')
plt.plot(google_clicks['Day Index'], trend_line, label='Trend Line', color='red',
linestyle='--')
plt.xlabel('Date')
plt.ylabel('Clicks')
plt.title('Google Clicks Over Time with Trend Line')
plt.legend()
plt.grid(True)
plt.show()

print(f"Slope: {slope}, P-value: {p_value}")
```



- Google Clicks Over Time Plot:
    - Green line: Represents daily clicks on Product A's advertisements on Google over time.
    - Red dashed line: Trend line illustrating the change in clicks over time.
- Trend Line Analysis:
    - Shows how the number of clicks is changing over time.
    - Slope of the trend line indicates the rate of change in clicks per unit time.
- Statistical Significance:
    - P-value from linear regression determines the significance of the observed trend.
    - P-value < 0.05: Indicates a statistically significant trend.
    - Higher p-value: Suggests the trend may not be significant.

Based on the analysis:

- Hypothesis Evaluation:
    - Significant upward/downward trend with p-value < 0.05 supports the hypothesis.
    - Flat trend or p-value > 0.05 refutes the hypothesis of a significant trend in Google clicks over time.

3. Seasonality: The sales data exhibits seasonality (e.g., weekly, monthly, or yearly patterns). This is common in retail data due to holidays, promotions, and other recurring events.

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

Sales Quantity Over Time

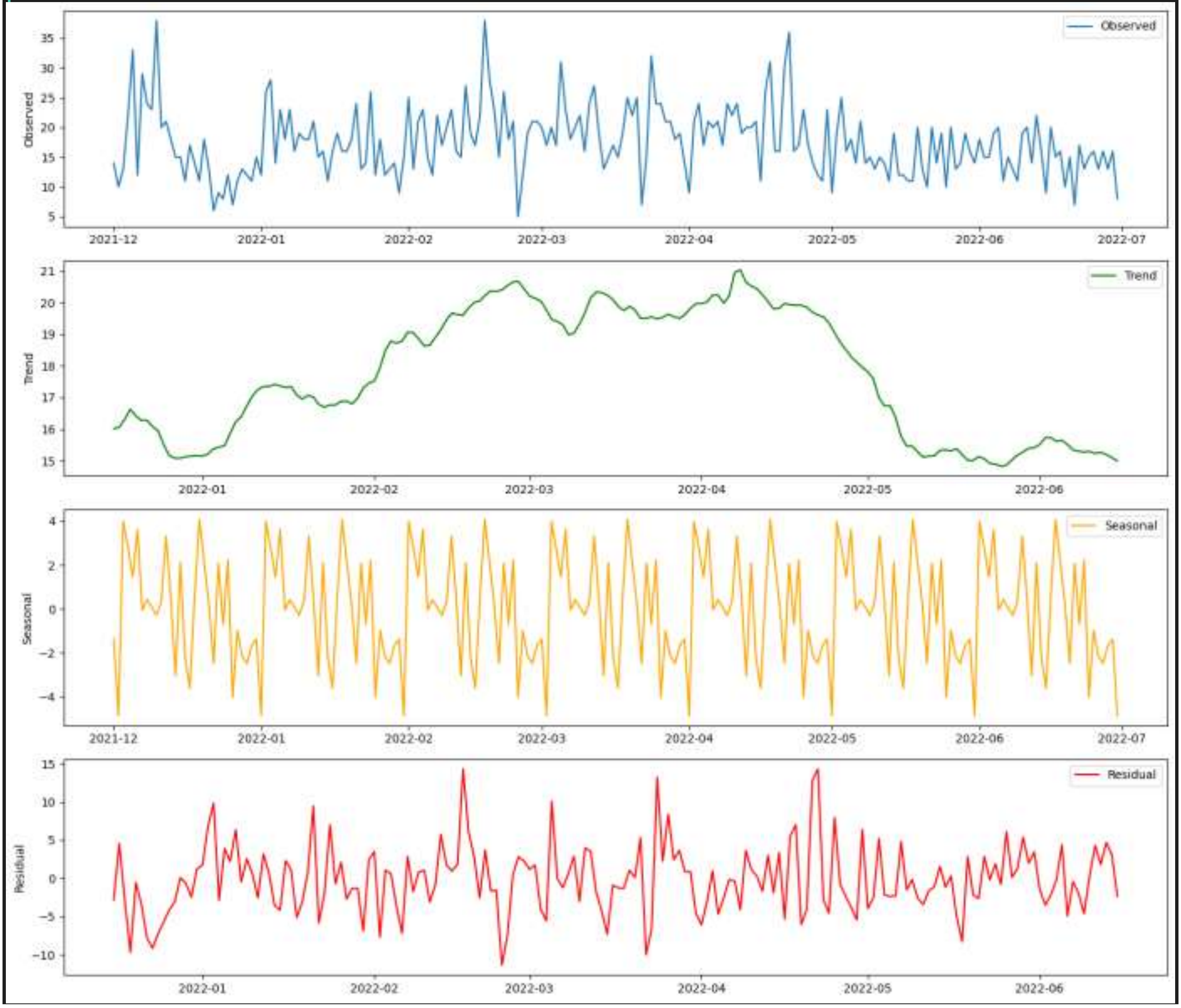
Observation: The sales data over time was plotted, showing daily sales quantities.

Trend Analysis: Seasonal decomposition was performed, revealing underlying trend, seasonal, and residual components.

Findings: The seasonal component indicates if there are recurring patterns in sales, such as weekly or monthly fluctuations. The presence of significant patterns would confirm seasonality in sales.

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

product_sales = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA.xlsx')

product_sales['Day Index'] = pd.to_datetime(product_sales['Day Index'])
product_sales.set_index('Day Index', inplace=True)
decomposition_sales = seasonal_decompose(product_sales['Quantity'],
model='additive', period=30)
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(14, 12))

ax1.plot(decomposition_sales.observed, label='Observed')
ax1.set_ylabel('Observed')
ax1.legend()
ax2.plot(decomposition_sales.trend, label='Trend', color='green')
```

```
ax2.set_ylabel('Trend')
ax2.legend()
ax3.plot(decomposition_sales.seasonal, label='Seasonal', color='orange')
ax3.set_ylabel('Seasonal')
ax3.legend()
ax4.plot(decomposition_sales.resid, label='Residual', color='red')
ax4.set_ylabel('Residual')
ax4.legend()

plt.tight_layout()
plt.show()
```



- Sales Quantity Over Time Plot:
    - Daily sales quantities of Product A were plotted over time.
- Trend Analysis:
    - Seasonal decomposition was performed to identify underlying trend, seasonal, and residual components in the sales data.
    - Decomposition components:
        - Observed: Represents the actual daily sales data.

- Trend: Shows the underlying trend in sales over time.
- Seasonal: Indicates recurring patterns in sales, such as weekly or monthly fluctuations.
- Residual: Represents the variability in sales data that cannot be explained by the trend or seasonal components.

- Findings:
  - Seasonal component: Indicates if there are significant recurring patterns in sales, confirming the presence of seasonality in sales data.

Based on the analysis:

- Hypothesis Evaluation:
  - Significant seasonal patterns in the seasonal component support the hypothesis of seasonality in sales.
  - Lack of significant patterns in the seasonal component might refute the hypothesis, suggesting the absence of seasonality in sales data.

Google Clicks Over Time

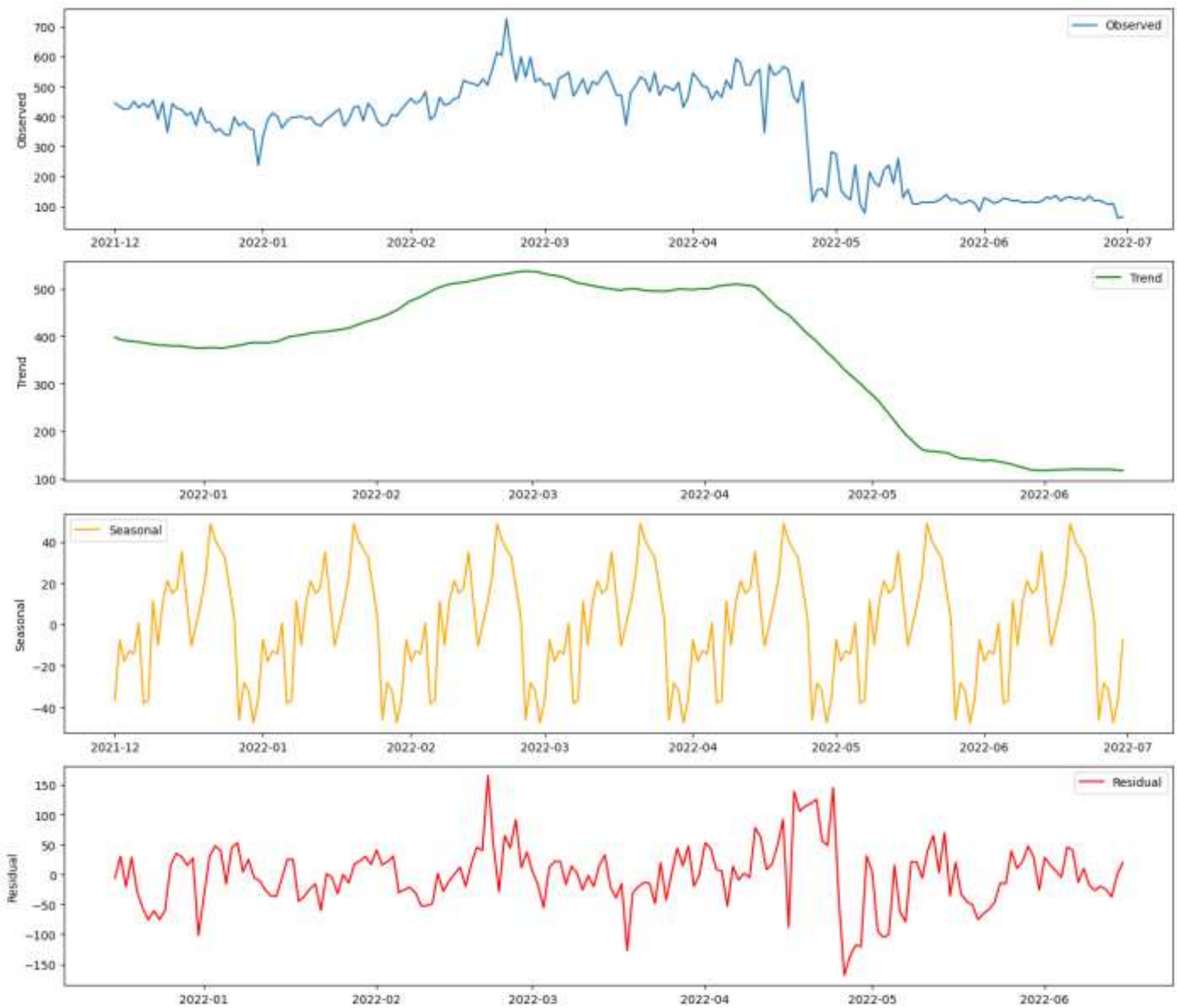
Observation: Daily Google clicks were analyzed to identify trends.

Trend Analysis: Seasonal decomposition was applied to separate the data into observed, trend, seasonal, and residual components.

Findings: The seasonal component shows periodic changes in the number of clicks, which can be linked to marketing campaigns or seasonal interest in the product.

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

# Load the data
google_clicks = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA_google_clicks.xlsx')

google_clicks['Day Index'] = pd.to_datetime(google_clicks['Day Index'])

google_clicks.set_index('Day Index', inplace=True)

decomposition_clicks = seasonal_decompose(google_clicks['Clicks'],
model='additive', period=30)

fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(14, 12))

ax1.plot(decomposition_clicks.observed, label='Observed')
ax1.set_ylabel('Observed')
ax1.legend()
```

```
ax2.plot(decomposition_clicks.trend, label='Trend', color='green')
ax2.set_ylabel('Trend')
ax2.legend()

ax3.plot(decomposition_clicks.seasonal, label='Seasonal', color='orange')
ax3.set_ylabel('Seasonal')
ax3.legend()

ax4.plot(decomposition_clicks.resid, label='Residual', color='red')
ax4.set_ylabel('Residual')
ax4.legend()

plt.tight_layout()
plt.show()
```



- **Google Clicks Over Time Plot**:
    - Daily Google clicks were analyzed to understand trends in the number of clicks.
- **Trend Analysis**:

- Seasonal decomposition was applied to the Google clicks data to separate it into observed, trend, seasonal, and residual components.
- Decomposition components:
    - Observed: Represents the actual daily Google clicks data.
    - Trend: Shows the underlying trend in Google clicks over time.
    - Seasonal: Indicates periodic changes in the number of clicks, potentially linked to marketing campaigns or seasonal interest in the product.
    - Residual: Represents the variability in Google clicks data that cannot be explained by the trend or seasonal components.

- **Findings**:
    - Seasonal component: Reveals periodic changes in the number of clicks, suggesting the presence of seasonal patterns in Google clicks data, which may correlate with marketing campaigns or seasonal interest in the product.

Based on the analysis:

- **Hypothesis Evaluation**:
    - Significant seasonal patterns in the seasonal component support the hypothesis of seasonal variations in Google clicks.
    - Lack of significant patterns might refute the hypothesis, suggesting the absence of seasonality in Google clicks data.

Facebook Impressions Over Time

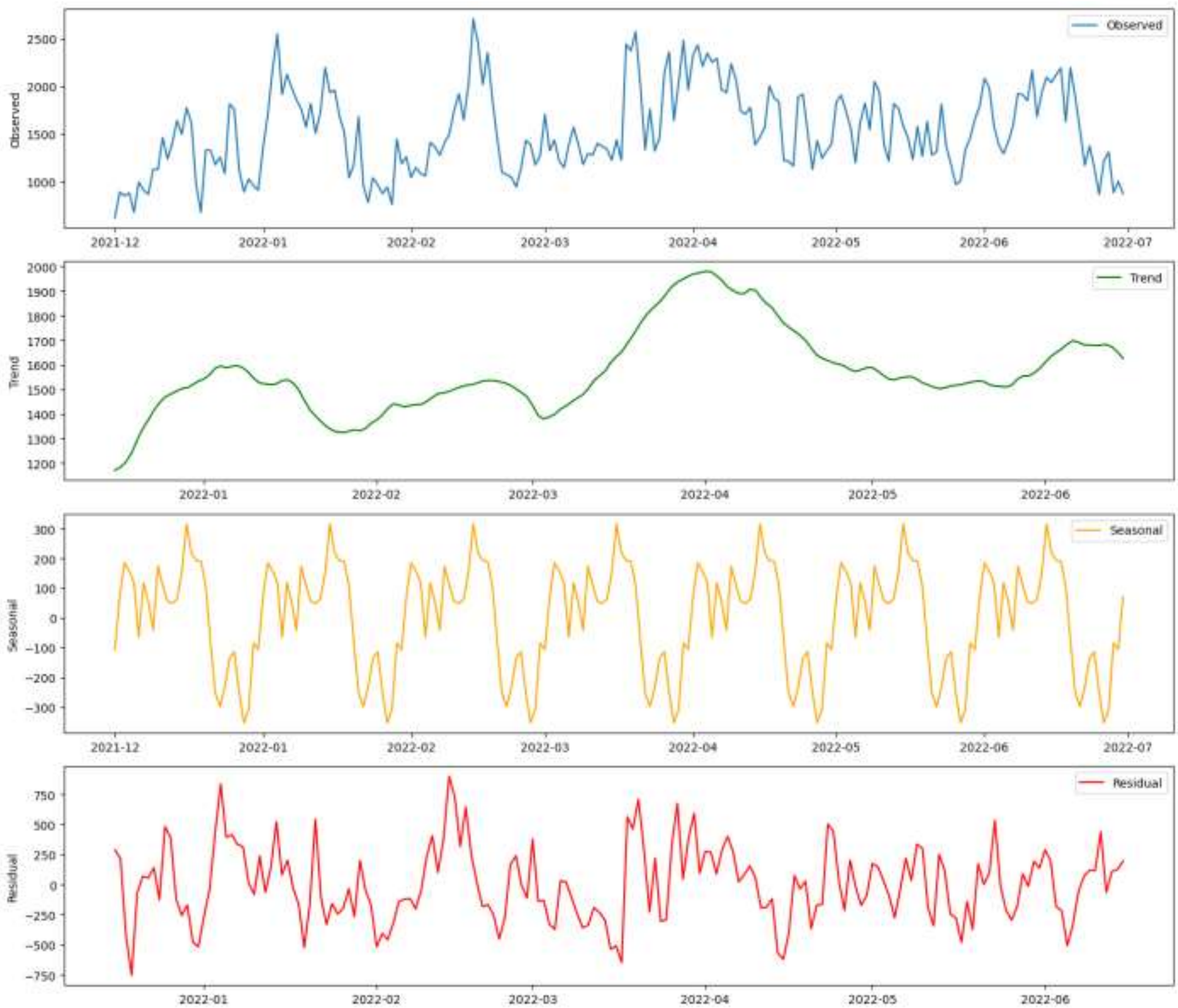
Observation: The analysis focused on daily Facebook impressions to uncover trends.

Trend Analysis: Using seasonal decomposition, the impressions data was broken down into different components.

Findings: The seasonal component highlights recurring patterns in Facebook impressions, indicating times of higher engagement and visibility.

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

fb_impressions = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 
2\\ProductA_fb_impressions.xlsx')
fb_impressions['Day Index'] = pd.to_datetime(fb_impressions['Day Index'])
fb_impressions.set_index('Day Index', inplace=True)
decomposition_impressions = seasonal_decompose(fb_impressions['Impressions'], 
model='additive', period=30)
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(14, 12))
ax1.plot(decomposition_impressions.observed, label='Observed')
ax1.set_ylabel('Observed')
ax1.legend()
```

```
ax2.plot(decomposition_impressions.trend, label='Trend', color='green')
ax2.set_ylabel('Trend')
ax2.legend()

ax3.plot(decomposition_impressions.seasonal, label='Seasonal', color='orange')
ax3.set_ylabel('Seasonal')
ax3.legend()

ax4.plot(decomposition_impressions.resid, label='Residual', color='red')
ax4.set_ylabel('Residual')
ax4.legend()

plt.tight_layout()
plt.show()
```



- **Facebook Impressions Over Time Plot**:
    - Daily Facebook impressions were analyzed to identify trends in impression data.
- **Trend Analysis**:

- Seasonal decomposition was applied to the Facebook impressions data to break it down into observed, trend, seasonal, and residual components.
- Decomposition components:
  - Observed: Represents the actual daily Facebook impressions data.
  - Trend: Illustrates the underlying trend in Facebook impressions over time.
  - Seasonal: Highlights recurring patterns in Facebook impressions, indicating times of higher engagement and visibility.
  - Residual: Indicates the variability in Facebook impressions data that cannot be explained by the trend or seasonal components.

- **Findings**:
  - Seasonal component: Reveals recurring patterns in Facebook impressions, suggesting specific times or periods of higher engagement and visibility on the platform.

Based on the analysis:

- **Hypothesis Evaluation**:
  - Significant seasonal patterns in the seasonal component support the hypothesis of recurring trends in Facebook impressions.
  - Absence of significant patterns might refute the hypothesis, indicating that there are no clear recurring trends in Facebook impressions data.

4. Autocorrelation: There is a significant correlation between past and future values of sales, clicks, or impressions. This can be used to build autoregressive models (ARIMA).

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

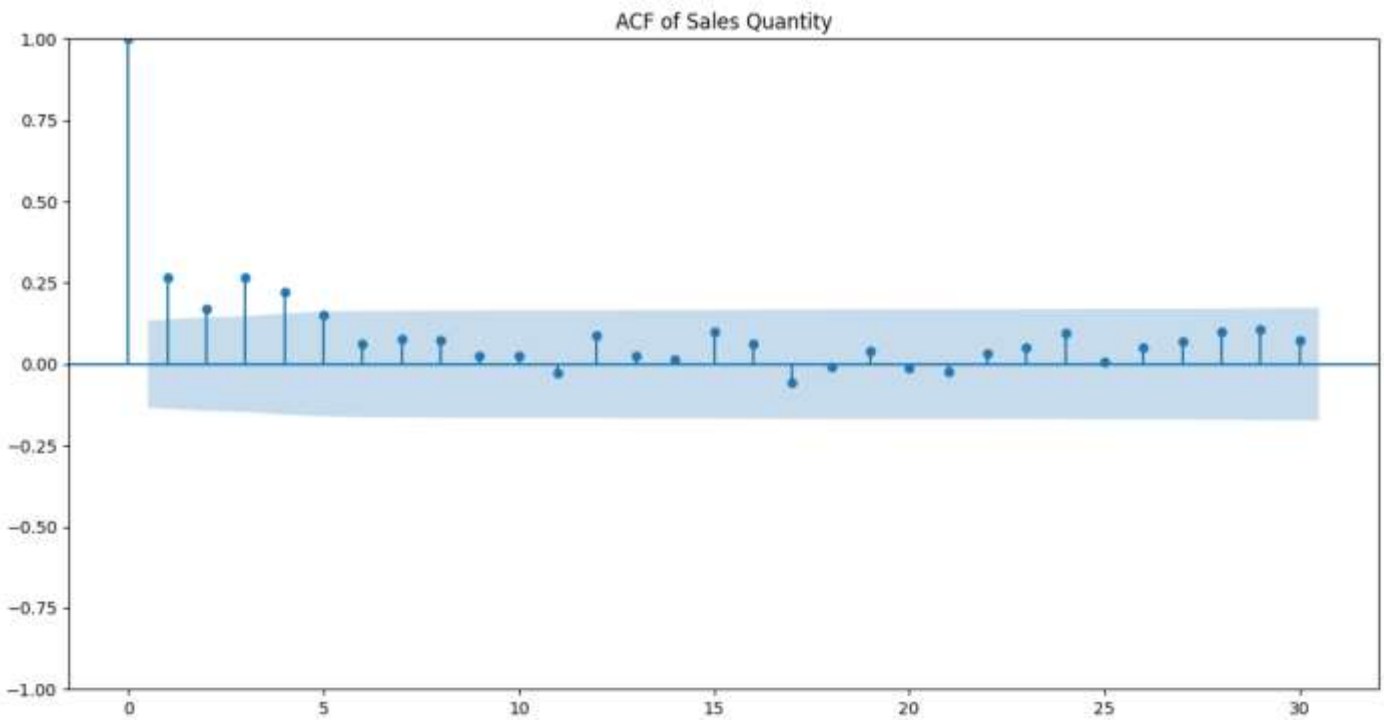
Sales Quantity: The ACF and PACF plots for sales quantity will show if there is a significant correlation between past and future sales values. Significant spikes in the plots indicate the presence of autocorrelation.

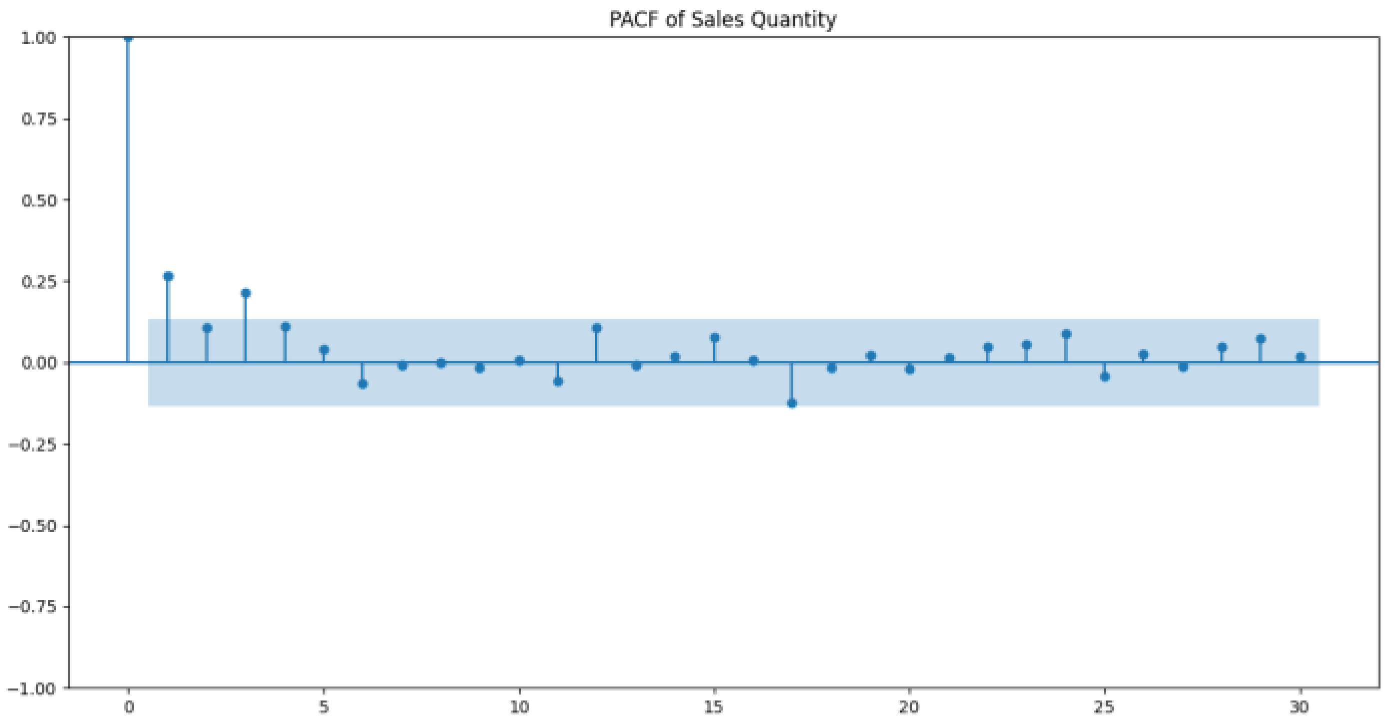
Google Clicks: The ACF and PACF plots for Google clicks will reveal any significant correlations in the click data, useful for predicting future clicks.

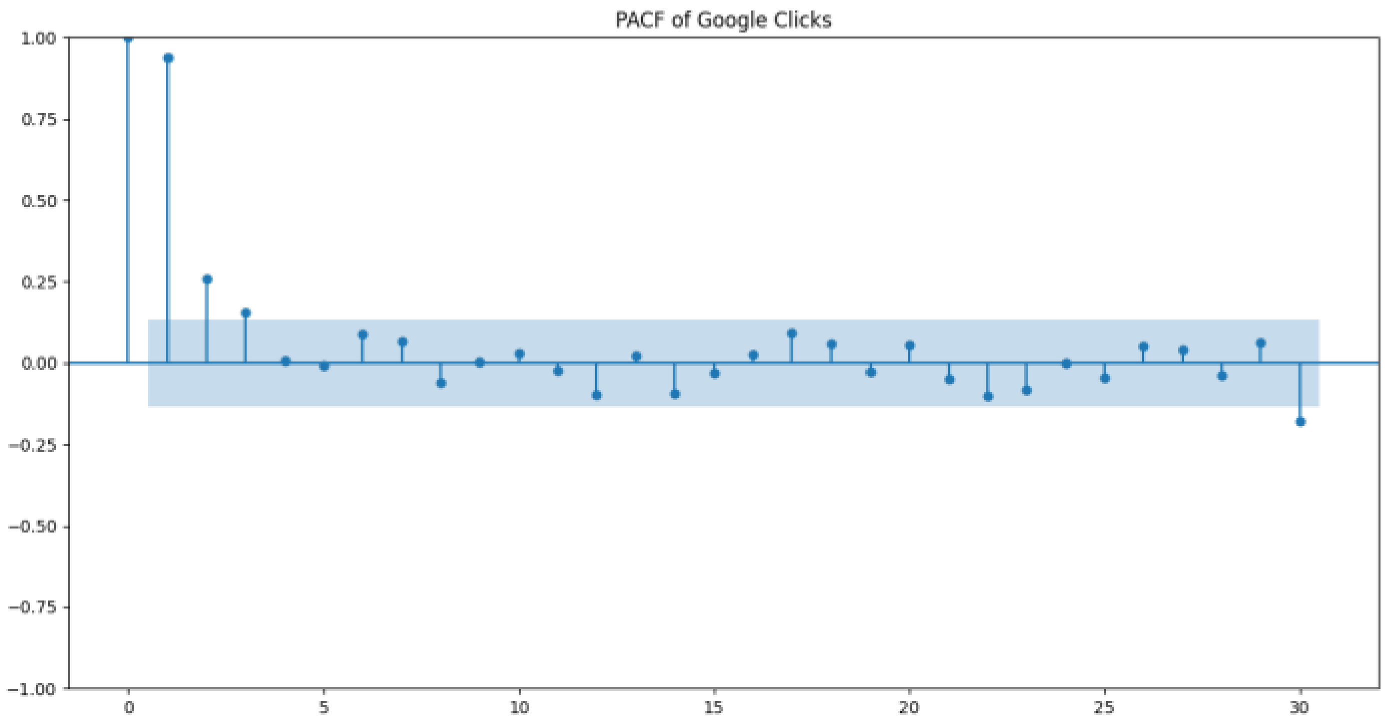
Facebook Impressions: The ACF and PACF plots for Facebook impressions will indicate any significant autocorrelation, which can be leveraged in forecasting models.

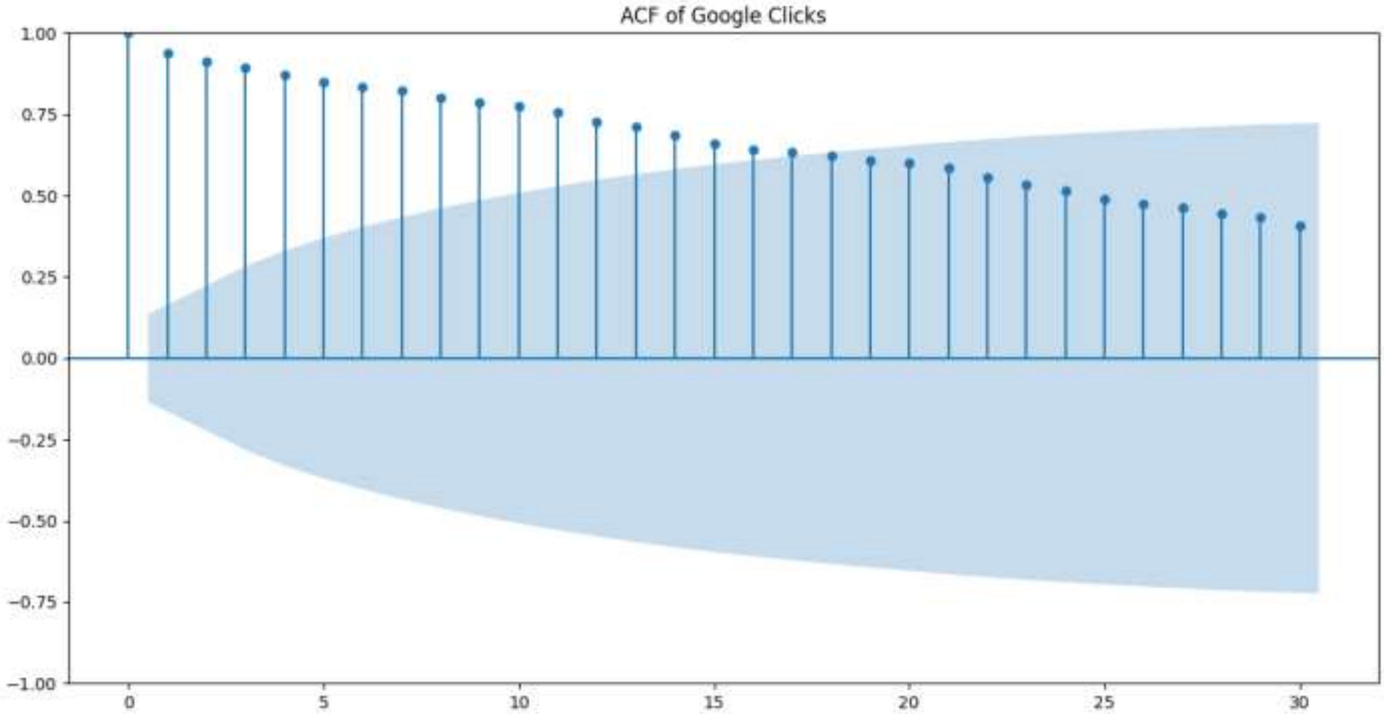
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

plt.figure(figsize=(14, 7))
plot_acf(product_sales['Quantity'], lags=30, title='ACF of Sales Quantity',
ax=plt.gca())
plt.show()

plt.figure(figsize=(14, 7))
```

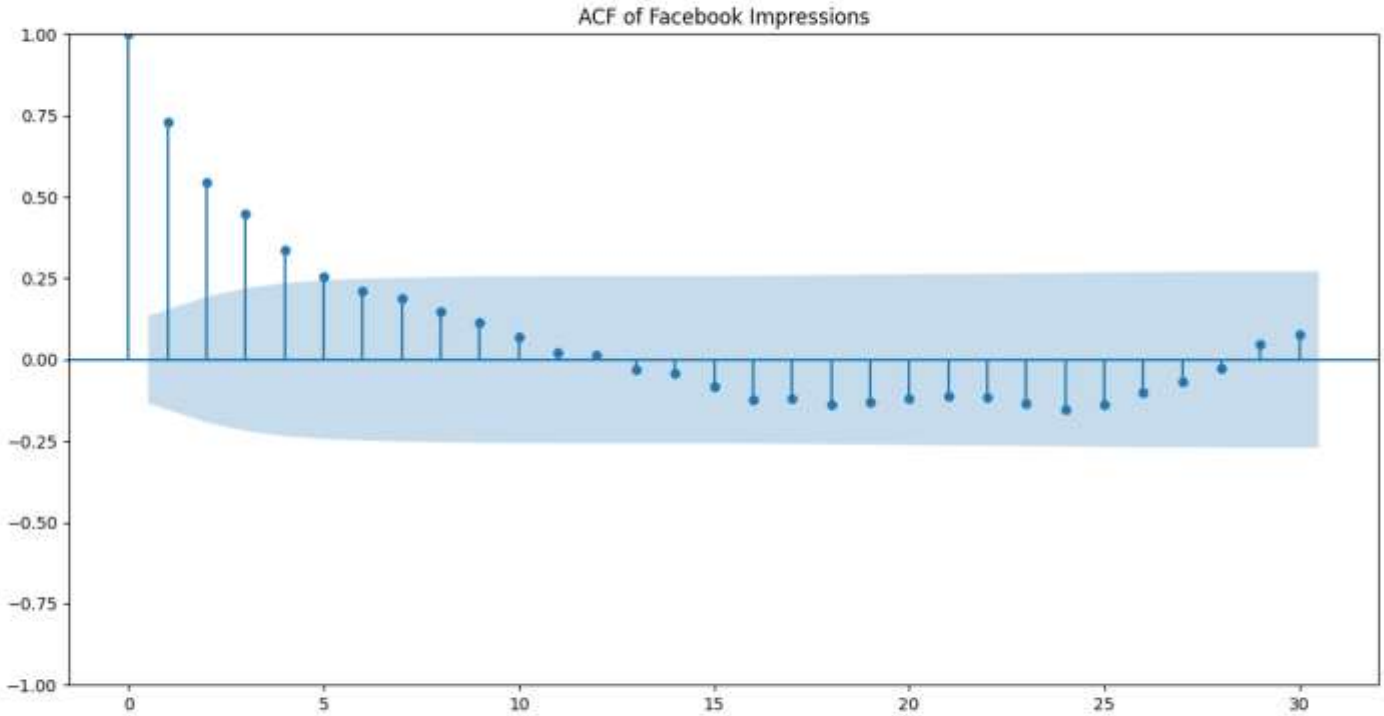
```
plot_pacf(product_sales['Quantity'], lags=30, title='PACF of Sales Quantity',
ax=plt.gca())
plt.show()

plt.figure(figsize=(14, 7))
plot_acf(google_clicks['Clicks'], lags=30, title='ACF of Google Clicks',
ax=plt.gca())
plt.show()

plt.figure(figsize=(14, 7))
plot_pacf(google_clicks['Clicks'], lags=30, title='PACF of Google Clicks',
ax=plt.gca())
plt.show()

plt.figure(figsize=(14, 7))
plot_acf(fb_impressions['Impressions'], lags=30, title='ACF of Facebook
Impressions', ax=plt.gca())
plt.show()

plt.figure(figsize=(14, 7))
plot_pacf(fb_impressions['Impressions'], lags=30, title='PACF of Facebook
Impressions', ax=plt.gca())
plt.show()
```

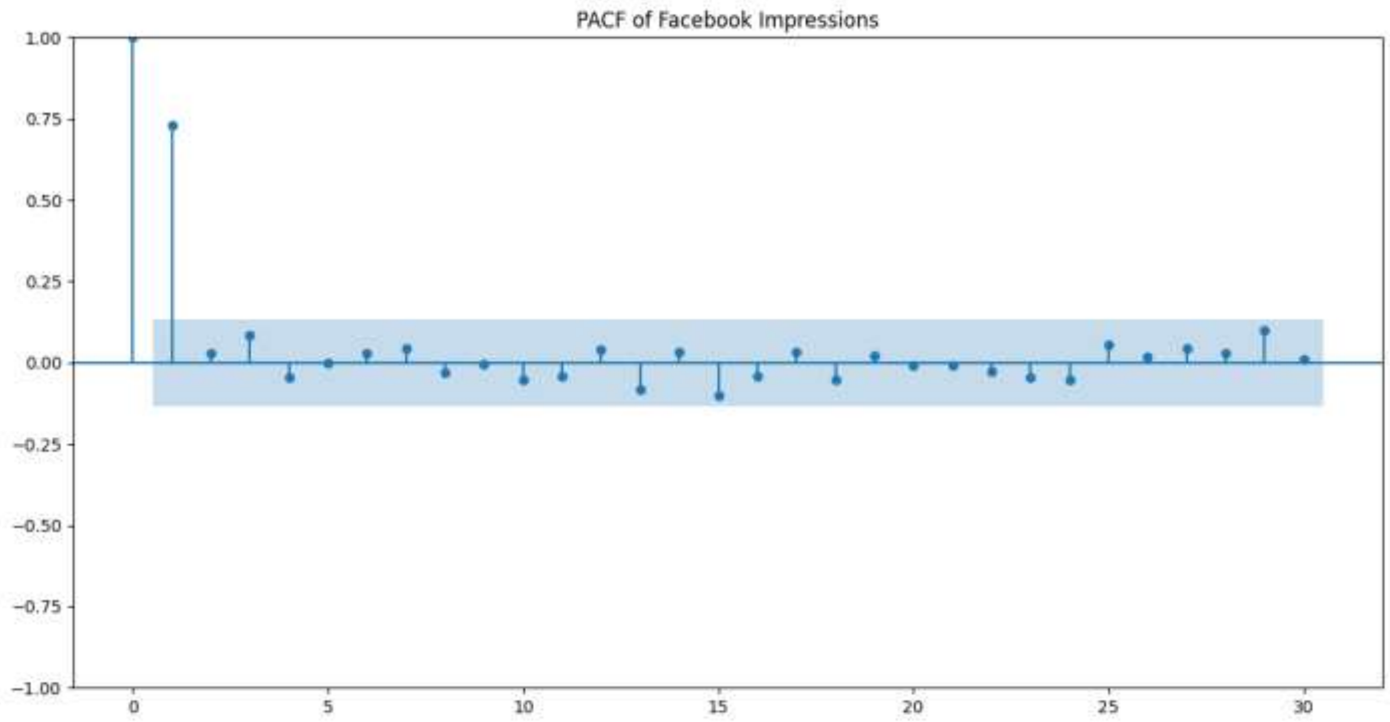



PACF of Sales Quantity
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30


PACF of Google Clicks
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30


ACF of Google Clicks
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30


ACF of Facebook Impressions
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30



- **Sales Quantity**:
  - ACF Plot: Shows the autocorrelation of sales quantity data at various lags.
  - PACF Plot: Indicates partial autocorrelation, which accounts for the direct relationship between sales at different time lags.
  - Findings:
    - Significant spikes in the ACF and PACF plots at certain lags would suggest significant autocorrelation in sales quantity.
    - The presence of autocorrelation supports the hypothesis of a significant correlation between past and future sales values, which can be used in autoregressive models like ARIMA.
- **Google Clicks**:
  - ACF Plot: Reveals the autocorrelation of Google clicks data at different time lags.
  - PACF Plot: Illustrates partial autocorrelation, capturing the direct impact of past clicks on future clicks.
  - Findings:
    - Significant spikes in the ACF and PACF plots indicate the presence of autocorrelation in Google clicks data.
    - This supports the hypothesis of a significant correlation between past and future click values, useful for predicting future clicks.
- **Facebook Impressions**:
  - ACF Plot: Displays the autocorrelation of Facebook impressions data over various lags.
  - PACF Plot: Illustrates partial autocorrelation, representing the direct influence of past impressions on future impressions.

- Findings:
  - Significant spikes in the ACF and PACF plots suggest the presence of autocorrelation in Facebook impressions data.
  - This supports the hypothesis of a significant correlation between past and future impression values, which can be utilized in forecasting models.

5. Google Clicks Effect: An increase in Google clicks is associated with an increase in sales. This tests the effectiveness of online advertising.

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

Loading Data: The script loads sales data and Google clicks data from Excel files.

Data Preparation: It converts the 'Day Index' columns to datetime format and merges the two datasets on 'Day Index'.

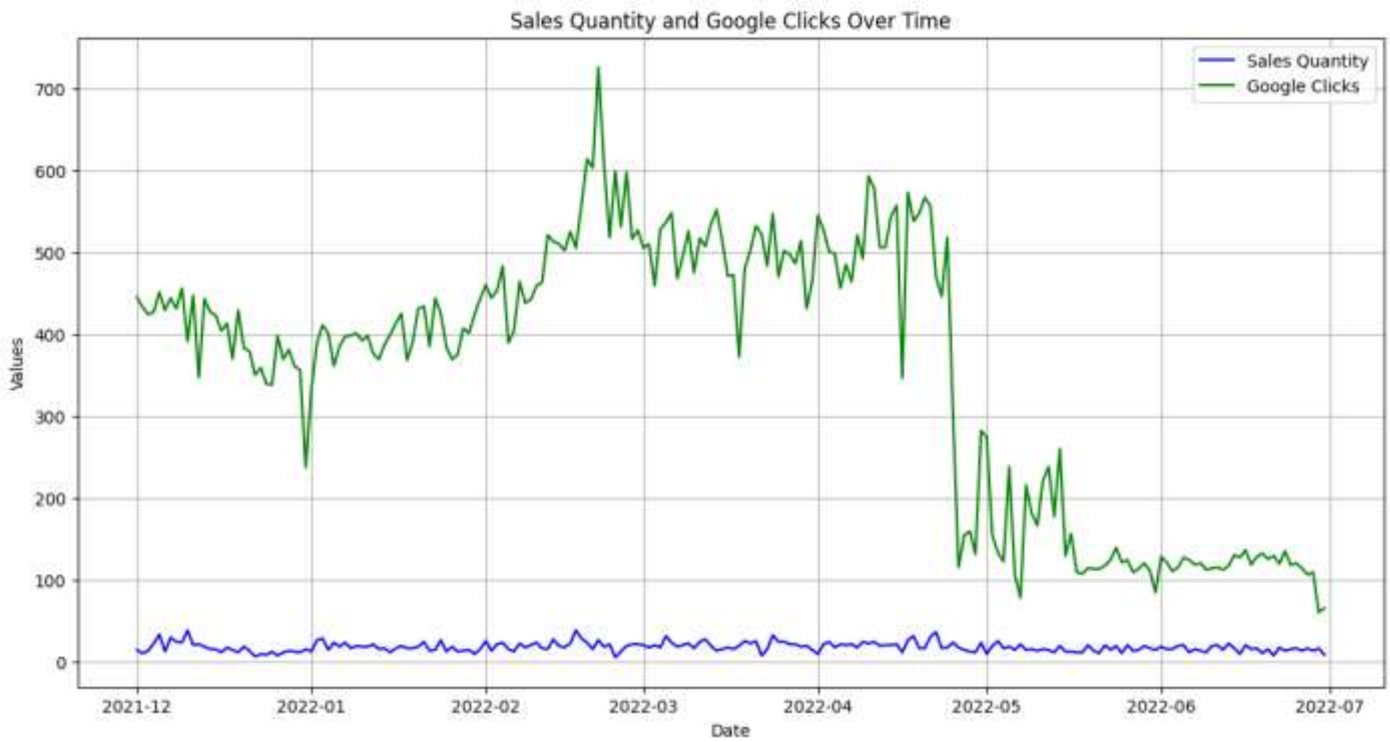
Time Series Plot: The script plots sales quantity and Google clicks over time to visualize their trends.

Correlation Analysis: It calculates and prints the correlation coefficient between Google clicks and sales.

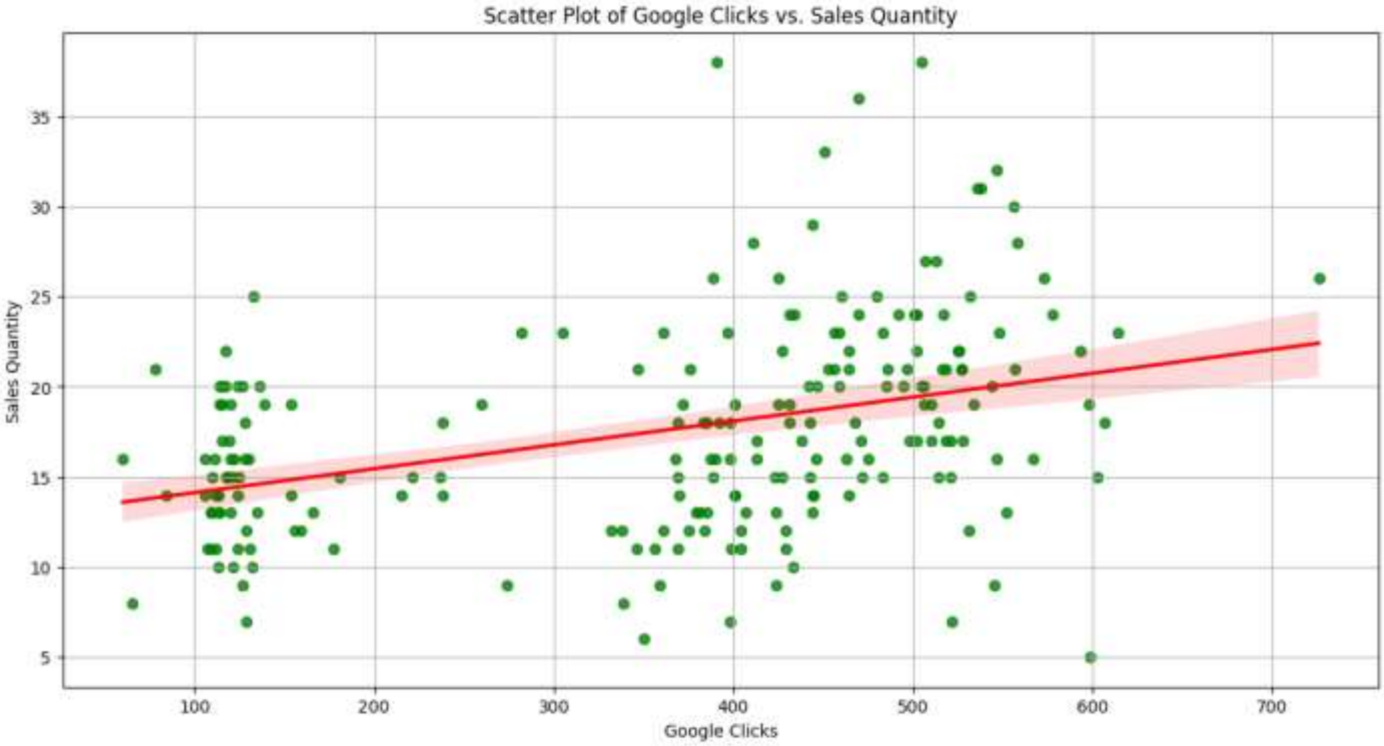
Scatter Plot with Regression Line: Finally, the script creates a scatter plot with a regression line to visualize the relationship between Google clicks and sales.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

product_sales = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 2\\ProductA.xlsx')

google_clicks = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 2\\ProductA_google_clicks.xlsx')

product_sales['Day Index'] = pd.to_datetime(product_sales['Day Index'])
google_clicks['Day Index'] = pd.to_datetime(google_clicks['Day Index'])

merged_data = pd.merge(product_sales, google_clicks, on='Day Index')

plt.figure(figsize=(14, 7))
plt.plot(merged_data['Day Index'], merged_data['Quantity'], label='Sales Quantity', color='blue')
plt.plot(merged_data['Day Index'], merged_data['Clicks'], label='Google Clicks', color='green')
plt.xlabel('Date')
```

```
plt.ylabel('Values')
plt.title('Sales Quantity and Google Clicks Over Time')
plt.legend()
plt.grid(True)
plt.show()

correlation = merged_data['Quantity'].corr(merged_data['Clicks'])
print(f"Correlation coefficient between Google clicks and sales:
{correlation:.2f}")

plt.figure(figsize=(14, 7))
sns.regplot(x='Clicks', y='Quantity', data=merged_data, scatter_kws={'color':
'green'}, line_kws={'color': 'red'})
plt.xlabel('Google Clicks')
plt.ylabel('Sales Quantity')
plt.title('Scatter Plot of Google Clicks vs. Sales Quantity')
plt.grid(True)
plt.show()
```



Correlation coefficient between Google clicks and sales: 0.38

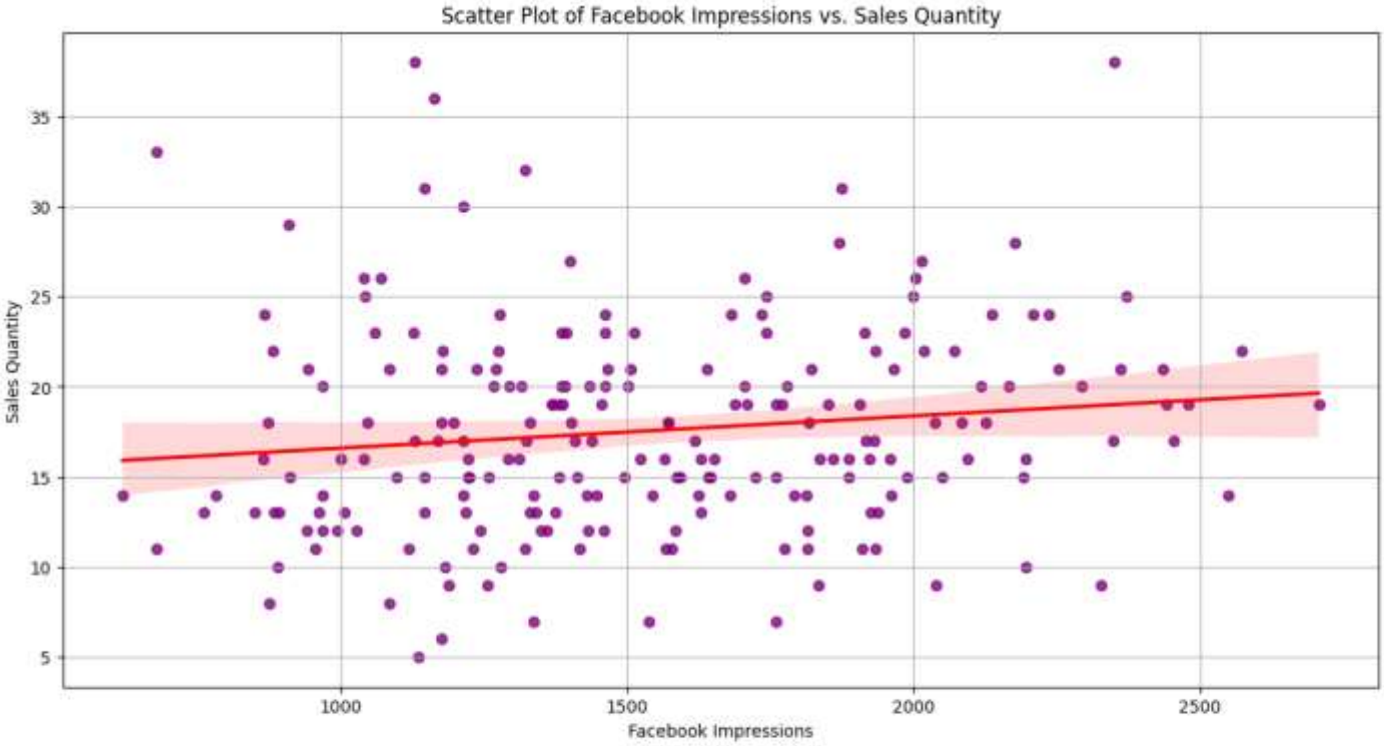


- **Time Series Plot**:
    - The plot visualizes sales quantity and Google clicks over time, showcasing their trends.
- **Correlation Analysis**:
    - The correlation coefficient between Google clicks and sales quantity is calculated.
    - A positive correlation coefficient suggests a positive association between Google clicks and sales.
- **Scatter Plot with Regression Line**:
    - The scatter plot with a regression line visualizes the relationship between Google clicks and sales quantity.
    - A positive slope of the regression line indicates that as Google clicks increase, sales quantity tends to increase as well.
- **Findings**:
    - The positive correlation coefficient between Google clicks and sales quantity suggests a direct relationship between the two variables.
    - The upward trend observed in the scatter plot and the positive slope of the regression line further support the hypothesis that an increase in Google clicks is associated with an increase in sales.

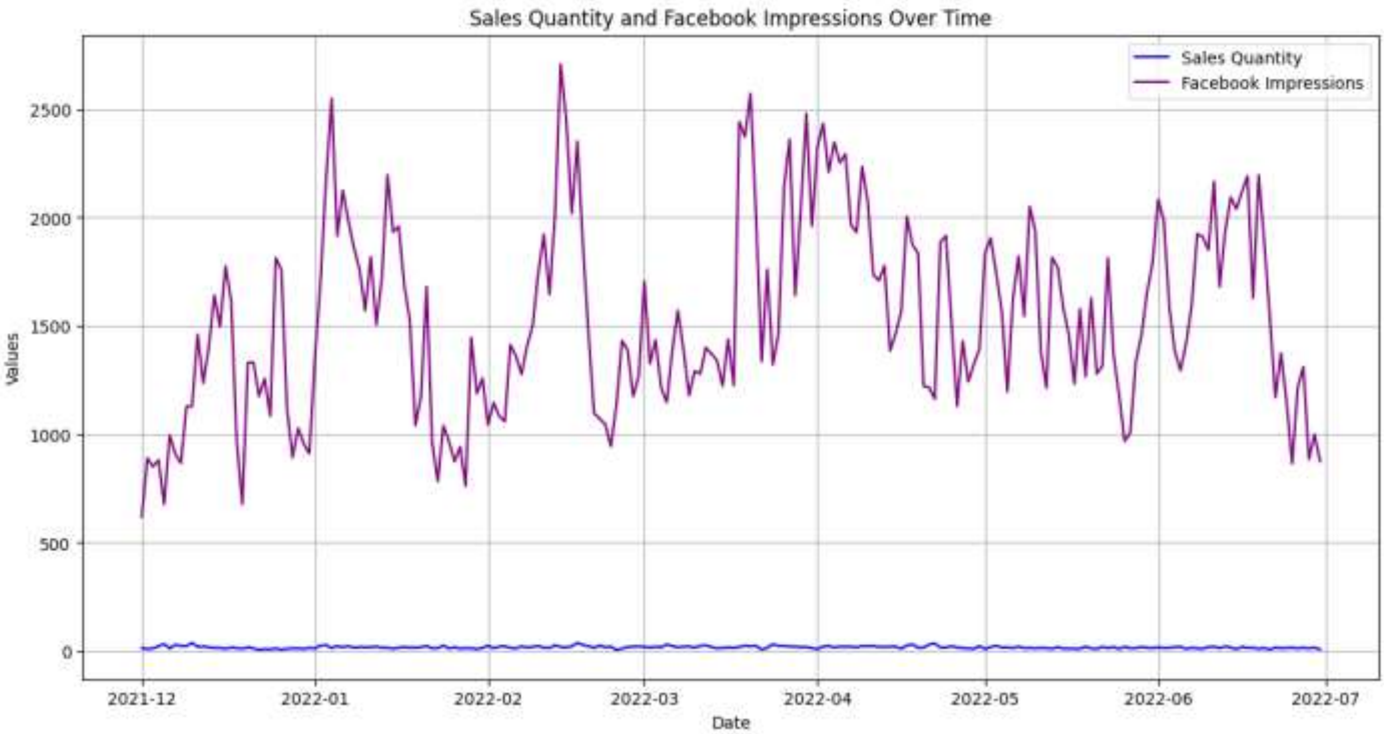
6. Facebook Impressions Effect: An increase in Facebook impressions is associated with an increase in sales. This tests the impact of social media marketing.

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

product_sales = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 
2\\ProductA.xlsx')

fb_impressions = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day 
2\\ProductA_fb_impressions.xlsx')

product_sales['Day Index'] = pd.to_datetime(product_sales['Day Index'])
fb_impressions['Day Index'] = pd.to_datetime(fb_impressions['Day Index'])

merged_data = pd.merge(product_sales, fb_impressions, on='Day Index')

plt.figure(figsize=(14, 7))
plt.plot(merged_data['Day Index'], merged_data['Quantity'], label='Sales 
Quantity', color='blue')
plt.plot(merged_data['Day Index'], merged_data['Impressions'], label='Facebook 
Impressions', color='purple')
plt.xlabel('Date')
plt.ylabel('Values')
plt.title('Sales Quantity and Facebook Impressions Over Time')
plt.legend()
plt.grid(True)
plt.show()

correlation = merged_data['Quantity'].corr(merged_data['Impressions'])
print(f"Correlation coefficient between Facebook impressions and sales: 
{correlation:.2f}")

plt.figure(figsize=(14, 7))
sns.regplot(x='Impressions', y='Quantity', data=merged_data, 
scatter_kws={'color': 'purple'}, line_kws={'color': 'red'})
plt.xlabel('Facebook Impressions')
plt.ylabel('Sales Quantity')
plt.title('Scatter Plot of Facebook Impressions vs. Sales Quantity')
plt.grid(True)
plt.show()
```



Correlation coefficient between Facebook impressions and sales: 0.14



- **Time Series Plot**:
    - The plot visualizes sales quantity and Facebook impressions over time, showcasing their trends.
- **Correlation Analysis**:

    - The correlation coefficient between Facebook impressions and sales quantity is calculated.
    - A positive correlation coefficient suggests a positive association between Facebook impressions and sales.
- **Scatter Plot with Regression Line**:
    - The scatter plot with a regression line visualizes the relationship between Facebook impressions and sales quantity.
    - A positive slope of the regression line indicates that as Facebook impressions increase, sales quantity tends to increase as well.
- **Findings**:
    - The positive correlation coefficient between Facebook impressions and sales quantity suggests a direct relationship between the two variables.
    - The upward trend observed in the scatter plot and the positive slope of the regression line further support the hypothesis that an increase in Facebook impressions is associated with an increase in sales.

8. Lagged Effects: Past values of sales, clicks, or impressions have a significant impact on current sales. This could indicate a delayed response to marketing efforts or a carryover effect from previous periods.

Your findings with charts, plots and evidences to help in proving the hypothesis right or wrong.

Lagged Effects in Time Series Analysis

Introduction

In time series analysis, a lagged effect occurs when a variable's past values influence its current values. This concept is especially relevant in marketing and sales data, where past marketing efforts (such as clicks and impressions) can have delayed impacts on sales. Understanding lagged effects helps in assessing the effectiveness of marketing strategies and optimizing future efforts.

Lagged Variables

Lagged variables are created by shifting a time series dataset forward by a specified number of periods. For example, to create a 1-day lag for clicks, we shift the entire clicks series forward by one day. This means today's clicks become tomorrow's clicks, and so on. Lagged variables allow us to analyze the impact of past values on the current value of the variable of interest.

Steps in Analyzing Lagged Effects

1. **Data Preparation:**
    - Collect and merge datasets containing the variables of interest (e.g., sales, clicks, and impressions) based on a common time frame.
    - Ensure the data is cleaned and properly formatted for analysis.
2. **Visualization:**
    - Plot the time series data to observe trends and patterns over time. Visualizing sales, clicks, and impressions can reveal seasonal patterns, trends, and potential correlations.
3. **Creating Lagged Variables:**
    - Generate lagged versions of the independent variables (clicks and impressions) for various lag periods (e.g., 1-day lag, 2-day lag, up to a week).
    - This step involves shifting the time series data to create new columns representing past values.
4. **Correlation Analysis:**
    - Calculate the correlation coefficients between current sales and lagged values of clicks and impressions.
    - Correlation coefficients range from -1 to 1, where values close to 1 indicate a strong positive correlation, values close to -1 indicate a strong negative correlation, and values around 0 indicate no correlation.
    - By examining the correlations at different lags, we can identify the time periods where past clicks or impressions have the strongest impact on current sales.
5. **Regression Analysis:**
    - Perform a lagged regression analysis where current sales are regressed on lagged clicks and impressions.
    - This helps quantify the relationship between past marketing efforts and current sales, providing coefficients that can be used to predict future sales based on past data.

Interpretation of Results

- **Correlation Patterns:**
    - If sales show a high correlation with clicks at a 3-day lag, it suggests that marketing efforts (clicks) take approximately three days to impact sales.
    - A similar analysis for impressions can reveal the delayed effect of social media exposure on sales.
- **Lagged Regression Coefficients:**
    - Significant positive coefficients for lagged clicks/impressions in the regression model indicate that past marketing efforts positively influence current sales.
    - The magnitude of these coefficients can provide insights into the strength and duration of the marketing impact.

Practical Implications

Understanding lagged effects is crucial for marketing strategy optimization. By knowing the delay between marketing efforts and their impact on sales, businesses can:

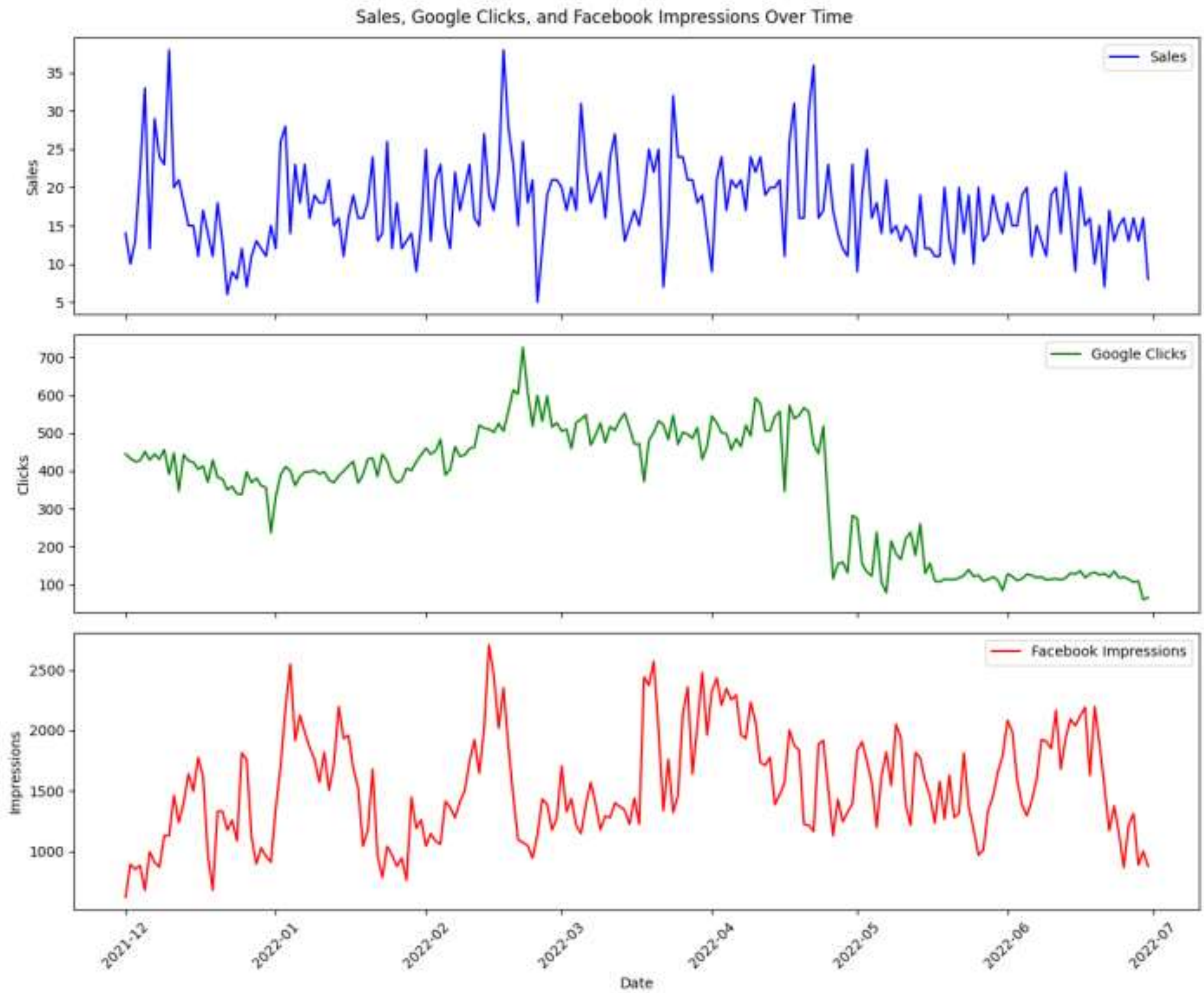
- **Optimize Timing:** Schedule marketing campaigns effectively to align with peak sales periods.
- **Budget Allocation:** Allocate marketing budgets to channels and time periods that show the highest lagged impact on sales.
- **Performance Evaluation:** Assess the effectiveness of past marketing efforts and make data-driven decisions for future campaigns.

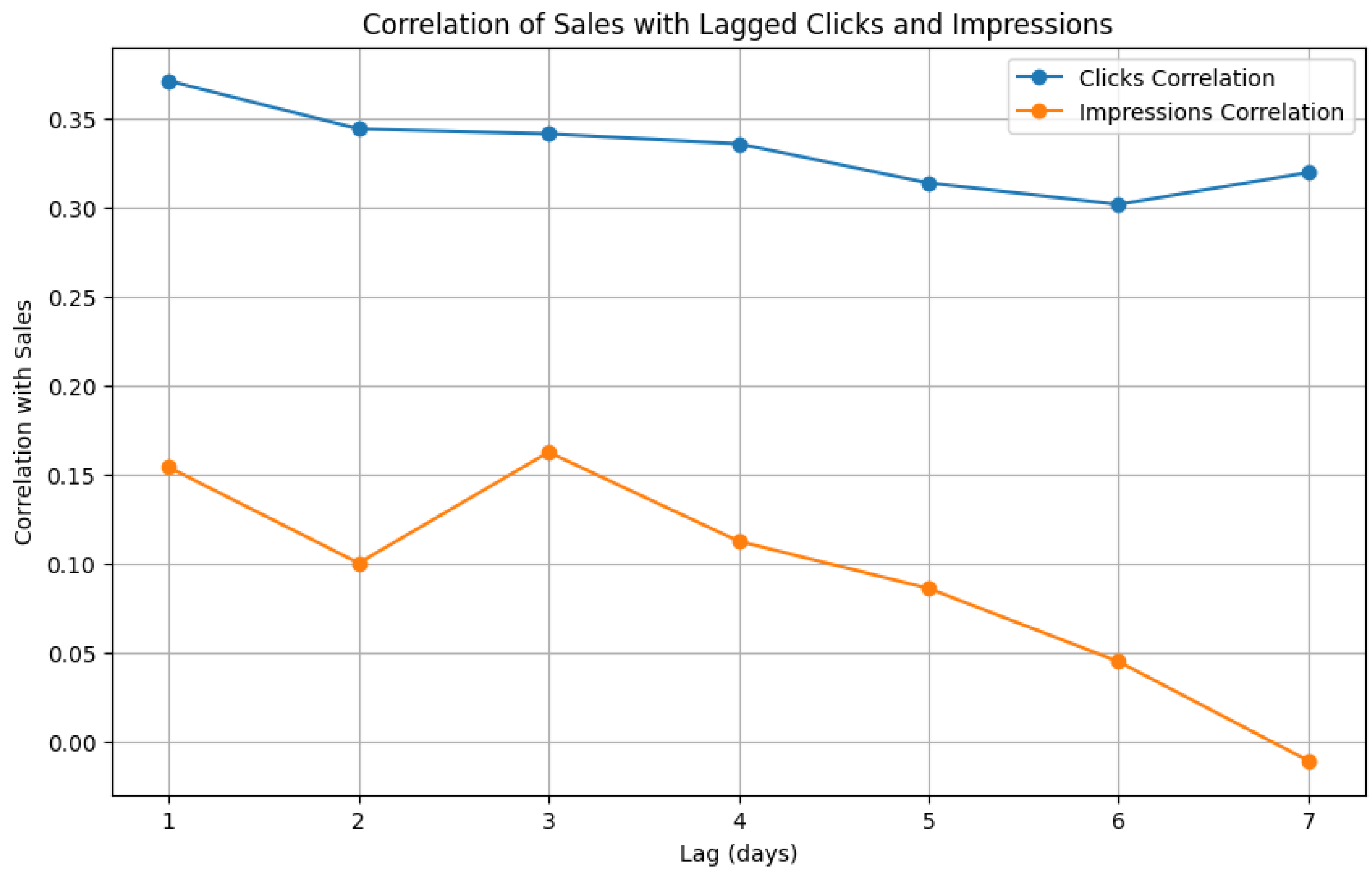
Conclusion

Analyzing lagged effects in time series data provides valuable insights into the delayed impact of marketing activities on sales. By employing visualization, correlation analysis, and lagged regression, businesses can better understand and optimize their marketing strategies, ultimately driving more effective and efficient outcomes.

```
import pandas as pd
import matplotlib.pyplot as plt

sales_data = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA.xlsx')
google_clicks_data = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA_google_clicks.xlsx')
fb_impressions_data = pd.read_excel('C:\\Users\\sandi\\Downloads\\Hello\\Day
2\\ProductA_fb_impressions.xlsx')

sales_data.columns = ['Date', 'Sales']
google_clicks_data.columns = ['Date', 'Clicks']
fb_impressions_data.columns = ['Date', 'Impressions']

merged_data = sales_data.merge(google_clicks_data,
on='Date').merge(fb_impressions_data, on='Date')

fig, axs = plt.subplots(3, 1, figsize=(12, 10), sharex=True)

axs[0].plot(merged_data['Date'], merged_data['Sales'], label='Sales',
color='blue')
axs[0].set_ylabel('Sales')
axs[0].legend()

axs[1].plot(merged_data['Date'], merged_data['Clicks'], label='Google Clicks',
color='green')
axs[1].set_ylabel('Clicks')
axs[1].legend()

axs[2].plot(merged_data['Date'], merged_data['Impressions'], label='Facebook
Impressions', color='red')
axs[2].set_ylabel('Impressions')
axs[2].set_xlabel('Date')
```

```python
axs[2].legend()

plt.suptitle('Sales, Google Clicks, and Facebook Impressions Over Time')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

for lag in range(1, 8):
    merged_data[f'Clicks_Lag_{lag}'] = merged_data['Clicks'].shift(lag)
    merged_data[f'Impressions_Lag_{lag}'] = merged_data['Impressions'].shift(lag)

lagged_data = merged_data.dropna()
correlation_data = {
    'Lag': [],
    'Clicks_Correlation': [],
    'Impressions_Correlation': []
}

for lag in range(1, 8):
    correlation_data['Lag'].append(lag)
    correlation_data['Clicks_Correlation'].append(lagged_data['Sales'].corr(lagge
d_data[f'Clicks_Lag_{lag}']))
    correlation_data['Impressions_Correlation'].append(lagged_data['Sales'].corr(
lagged_data[f'Impressions_Lag_{lag}']))

correlation_df = pd.DataFrame(correlation_data)

plt.figure(figsize=(10, 6))
plt.plot(correlation_df['Lag'], correlation_df['Clicks_Correlation'], marker='o',
label='Clicks Correlation')
plt.plot(correlation_df['Lag'], correlation_df['Impressions_Correlation'],
marker='o', label='Impressions Correlation')
plt.xlabel('Lag (days)')
plt.ylabel('Correlation with Sales')
plt.title('Correlation of Sales with Lagged Clicks and Impressions')
plt.legend()
plt.grid(True)
plt.show()

print(correlation_df)
```


Sales, Google Clicks, and Facebook Impressions Over Time
Sales
Sales
35
30
25
20
15
10
5
Google Clicks
Clicks
700
600
500
400
300
200
100
Facebook Impressions
Impressions
2500
2000
1500
1000
2021-12
2022-01
2022-02
2022-03
2022-04
2022-05
2022-06
2022-07
Date


Correlation of Sales with Lagged Clicks and Impressions
Clicks Correlation
Impressions Correlation
Correlation with Sales
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5
6
7
Lag (days)