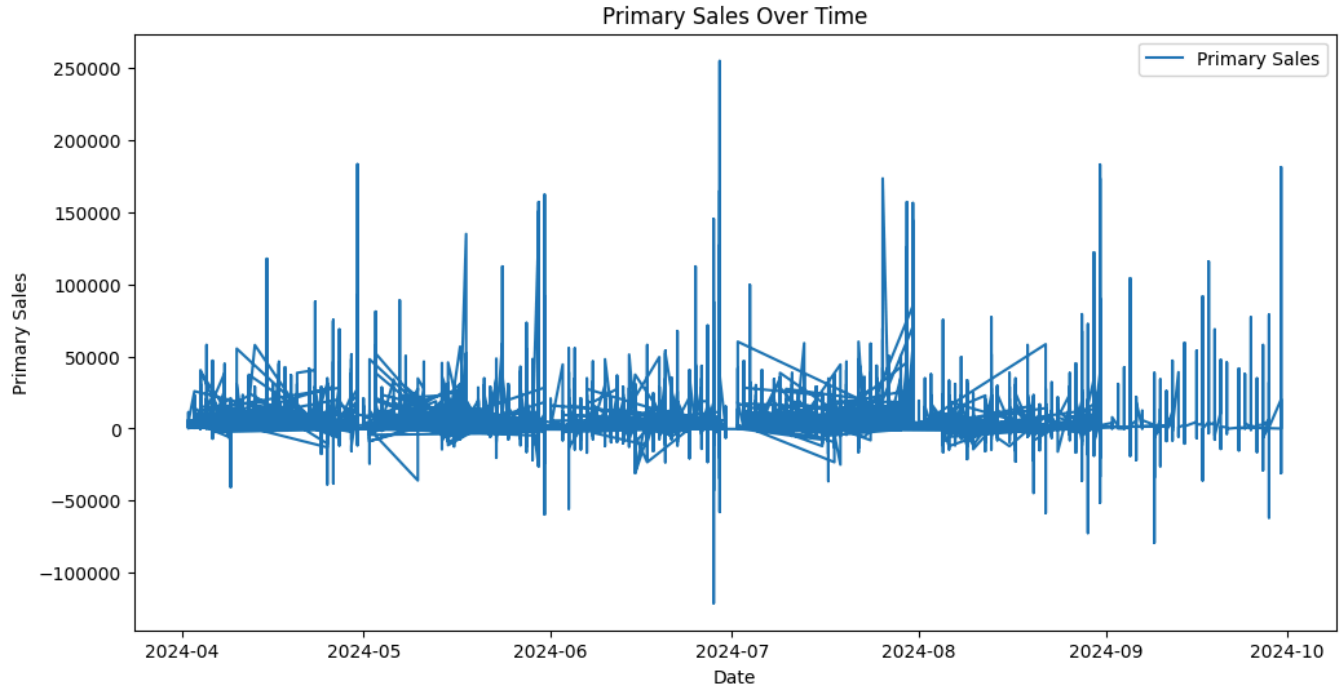
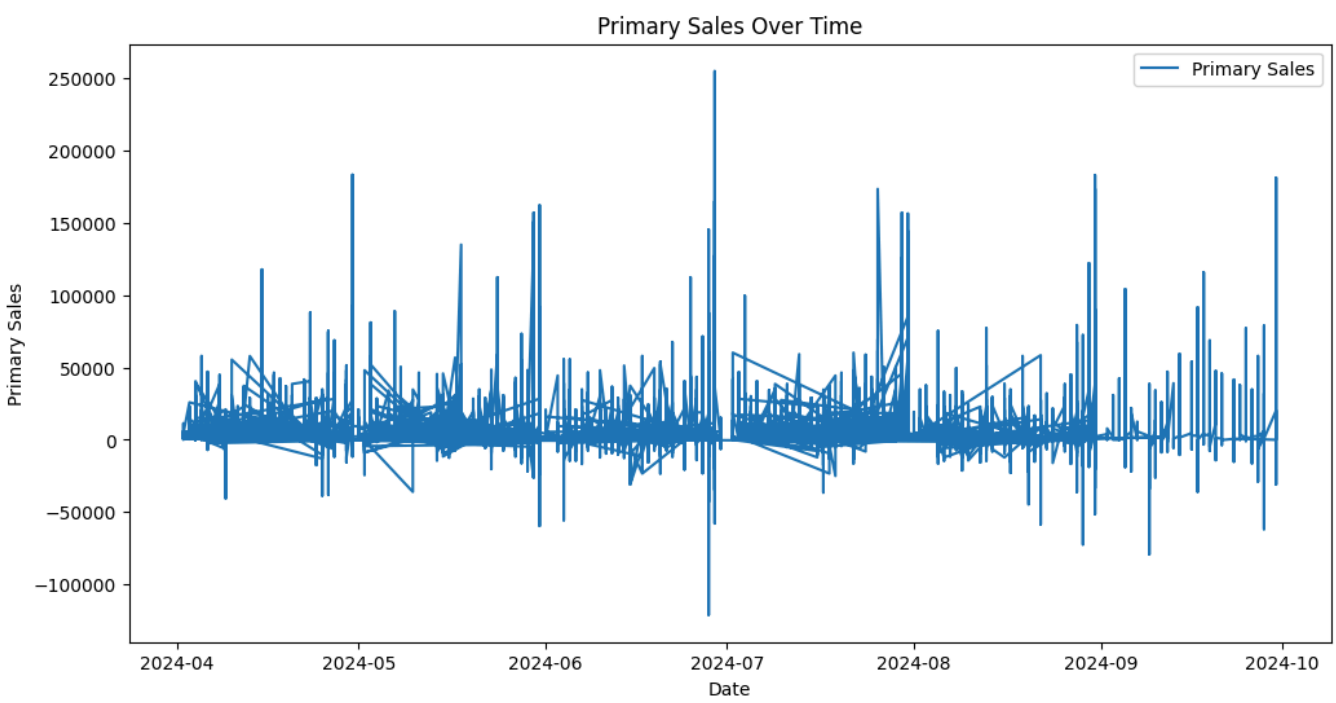
```
import pandas as pd
import matplotlib.pyplot as plt


df = pd.read_csv("/content/ELB-Sales-Data.csv", parse_dates=['Date'])

# Step 1: Preprocessing and basic exploration
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)

# Step 2: Handling missing values
# Check for missing values in the dataset
print(df.isnull().sum())

# Step 3: Descriptive statistics (mean, median, etc.)
print(df.describe())
```

```
Sr                          0
ID                          0
Customer                    0
Primary Sales               0
Cancelled                   0
Is added to primary log     0
Against Expiry              0
Breakage                    0
Claim Offer                 0
Quantity                    0
Net Rate                    0
Free Item                   0
HQ                        180
Sales Team                180
Return for Reason       89300
Sales Return                0
Sales Invoice Name          2
Rate Difference             0
Item Name                   0
Free Item Value             0
dtype: int64
                  Sr  Primary Sales      Cancelled  Is added to primary log  \
count  107444.000000  107444.000000  107444.000000            107444.000000
mean    81398.343044    2514.400220       0.018726                 0.998325
std     31018.861523    6077.772576       0.135556                 0.040896
min      9354.000000 -120939.060000       0.000000                 0.000000
25%     54538.750000       0.000000       0.000000                 1.000000
50%     81399.500000    1157.200000       0.000000                 1.000000
75%    108260.250000    3085.800000       0.000000                 1.000000
max    135121.000000  254574.000000       1.000000                 1.000000

       Against Expiry       Breakage    Claim Offer       Quantity  \
count   107444.000000  107444.000000  107444.000000  107444.000000
mean       -66.504029      -9.788137     -34.735094      26.366619
std        535.978101     299.959327     315.690782      54.720325
min     -44550.000000  -61088.000000  -24940.285710    -490.000000
25%          0.000000       0.000000       0.000000       3.000000
50%          0.000000       0.000000       0.000000      10.000000
75%          0.000000       0.000000       0.000000      30.000000
max      21901.490000    8100.000000    7675.714286    2000.000000

            Net Rate      Free Item   Sales Return  Rate Difference  \
count  107444.000000  107444.000000  107444.000000    107444.000000
mean       88.986826       0.054252     -60.959267       -13.028095
std        76.230676       0.226514     983.102573        200.668543
min      -125.360000       0.000000 -118743.300000    -17607.700000
25%        51.430000       0.000000       0.000000         0.000000
50%        86.790000       0.000000       0.000000         0.000000
75%       122.030000       0.000000       0.000000         0.000000
max      1684.730000       1.000000    5785.800000      1225.000000

       Free Item Value
count    107444.000000
mean        -47.930296
std         428.920348
min      -38507.142860
25%           0.000000
50%           0.000000
75%           0.000000
max        6171.428571
```



```
# Step 4: Basic analysis of primary sales over time
plt.figure(figsize=(12,6))
plt.plot(df['Primary Sales'], label='Primary Sales')
plt.title('Primary Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Primary Sales')
plt.legend()
plt.show()
```

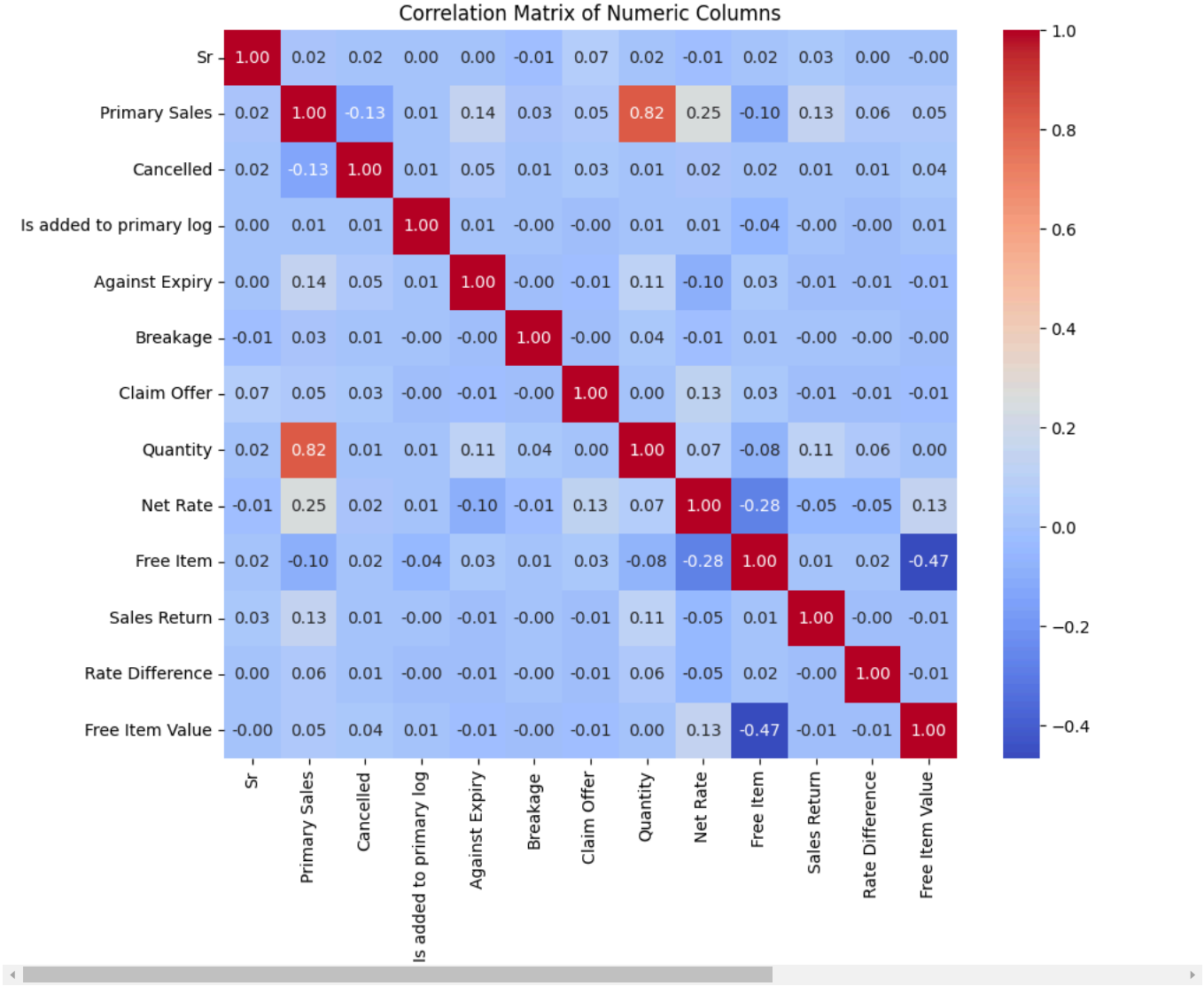



```
# Step 5: Check for correlations between columns, if needed
# Step 5: Check for correlations between numeric columns
# Select only numeric columns
numeric_df = df.select_dtypes(include=['number'])

# Compute the correlation matrix
correlation_matrix = numeric_df.corr()

# Print the correlation matrix
print(correlation_matrix)
```

```
                              Sr  Primary Sales  Cancelled
Sr                      1.000000       0.022398   0.016671
Primary Sales           0.022398       1.000000  -0.127082
Cancelled               0.016671      -0.127082   1.000000
Is added to primary log 0.003917       0.007564   0.005659
Against Expiry          0.003879       0.140877   0.054284
Breakage               -0.014025       0.025809   0.012673
Claim Offer             0.071049       0.045520   0.030296
Quantity                0.017140       0.823273   0.012064
Net Rate               -0.009896       0.252527   0.023944
Free Item               0.018841      -0.099085   0.019049
Sales Return            0.026632       0.133903   0.012378
Rate Difference         0.001803       0.056880   0.011003
Free Item Value        -0.000917       0.046230   0.040957

                        Is added to primary log  Against Expiry  Breakage  \
Sr                                     0.003917        0.003879 -0.014025
Primary Sales                          0.007564        0.140877  0.025809
Cancelled                              0.005659        0.054284  0.012673
Is added to primary log                1.000000        0.005041 -0.000897
Against Expiry                         0.005041        1.000000 -0.004049
Breakage                              -0.000897       -0.004049  1.000000
Claim Offer                           -0.004507       -0.013652 -0.003590
Quantity                               0.009441        0.113441  0.041105
Net Rate                               0.005188       -0.099676 -0.008352
```

```
Sales Return                          -0.001908       -0.007694 -0.002023
Rate Difference                       -0.002660       -0.008056 -0.002119
Free Item Value                        0.009617       -0.013866 -0.003646

                           Claim Offer  Quantity  Net Rate  Free Item  \
Sr                            0.071049  0.017140 -0.009896   0.018841
Primary Sales                 0.045520  0.823273  0.252527  -0.099085
Cancelled                     0.030296  0.012064  0.023944   0.019049
Is added to primary log      -0.004507  0.009441  0.005188  -0.044443
Against Expiry               -0.013652  0.113441 -0.099676   0.029718
Breakage                     -0.003590  0.041105 -0.008352   0.007816
Claim Offer                   1.000000  0.002637  0.127351   0.026353
Quantity                      0.002637  1.000000  0.069684  -0.079472
Net Rate                      0.127351  0.069684  1.000000  -0.277883
Free Item                     0.026353 -0.079472 -0.277883   1.000000
Sales Return                 -0.006823  0.110531 -0.048036   0.014851
Rate Difference              -0.007144  0.058586 -0.049906   0.015550
Free Item Value              -0.012295  0.001484  0.129866  -0.466571

                           Sales Return  Rate Difference  Free Item Value
Sr                             0.026632         0.001803        -0.000917
Primary Sales                  0.133903         0.056880         0.046230
Cancelled                      0.012378         0.011003         0.040957
Is added to primary log       -0.001908        -0.002660         0.009617
Against Expiry                -0.007694        -0.008056        -0.013866
Breakage                      -0.002023        -0.002119        -0.003646
Claim Offer                   -0.006823        -0.007144        -0.012295
Quantity                       0.110531         0.058586         0.001484
Net Rate                      -0.048036        -0.049906         0.129866
Free Item                      0.014851         0.015550        -0.466571
Sales Return                   1.000000        -0.004026        -0.006929
Rate Difference               -0.004026         1.000000        -0.007255
Free Item Value               -0.006929        -0.007255         1.000000
```

```
# Optionally, visualize the correlation matrix as a heatmap
import seaborn as sns

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix of Numeric Columns')
plt.show()
```

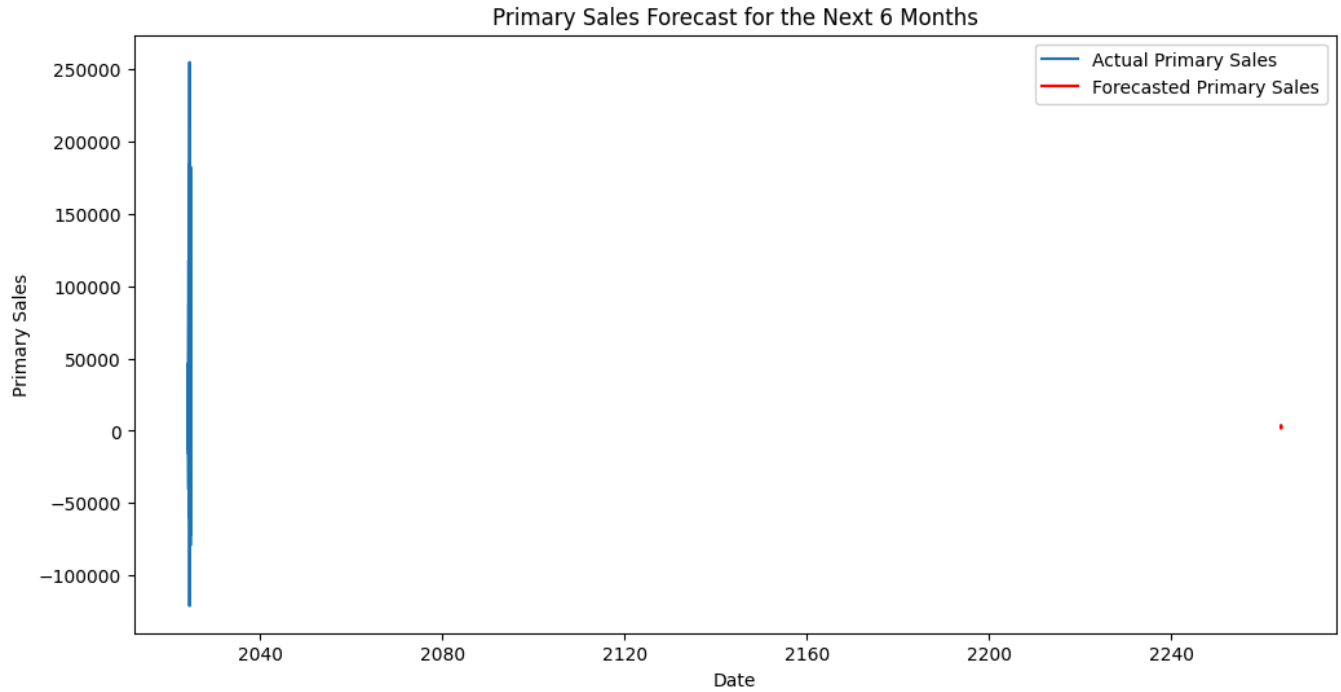



```
# Step 6: Investigating seasonality or trends (optional)
# For example, resample by month to analyze monthly trends
monthly_sales = df['Primary Sales'].resample('M').sum()
monthly_sales.plot(figsize=(12,6), title='Monthly Primary Sales')
plt.ylabel('Total Primary Sales')
plt.show()
```

```
<ipython-input-30-70c8bf812830>:3: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead
  monthly_sales = df['Primary Sales'].resample('M').sum()
```

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Step 7: Forecasting with Exponential Smoothing (Holt-Winters method)
# Ensure the dataset has no missing values in 'Primary Sales'
df.dropna(subset=['Primary Sales'], inplace=True)

# Fit the Holt-Winters model for trend and seasonality
model = ExponentialSmoothing(df['Primary Sales'], trend='add', seasonal='add', seasonal_periods=12)
model_fit = model.fit()

# Step 8: Forecast the next 6 months (adjust as needed)
forecast = model_fit.forecast(steps=6)

# Step 9: Plot the forecast
plt.figure(figsize=(12,6))
plt.plot(df['Primary Sales'], label='Actual Primary Sales')
plt.plot(forecast.index, forecast, label='Forecasted Primary Sales', color='red')
plt.title('Primary Sales Forecast for the Next 6 Months')
plt.xlabel('Date')
plt.ylabel('Primary Sales')
plt.legend()
plt.show()

# Output the forecasted sales for the next 6 months
print("Forecasted Primary Sales for the Next 6 Months:")
print(forecast)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/holtwinters/model.py:84: RuntimeWarning: overflow encountered in matmul
  return err.T @ err
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/holtwinters/model.py:918: ConvergenceWarning: Optimization failed to converg
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Predic
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In th
  return get_prediction_index(
```




```
Forecasted Primary Sales for the Next 6 Months:
107444    1889.948875
107445    2425.420405
107446    2236.112058
107447    3904.247024
107448    2518.473797
107449    2112.713454
dtype: float64
```