

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
```

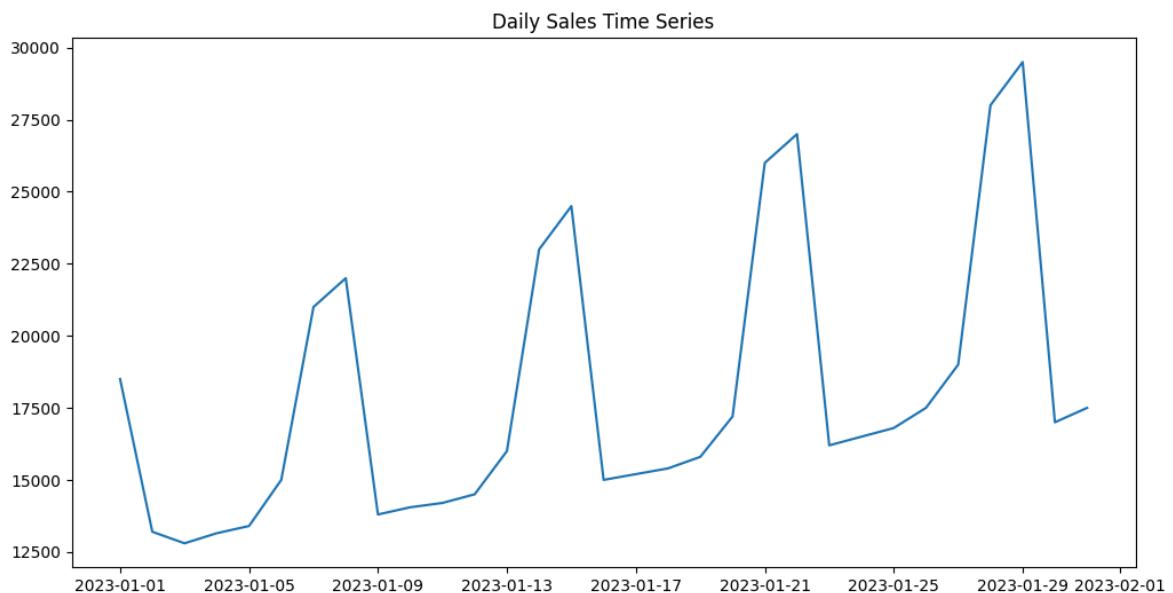
```
In [3]: data = pd.read_csv(r"C:\Users\santh\Downloads\retail_sales_dataset\retail_sales.
```

```
In [4]: data['Date'] = pd.to_datetime(data['Date'])

data.set_index('Date', inplace=True)

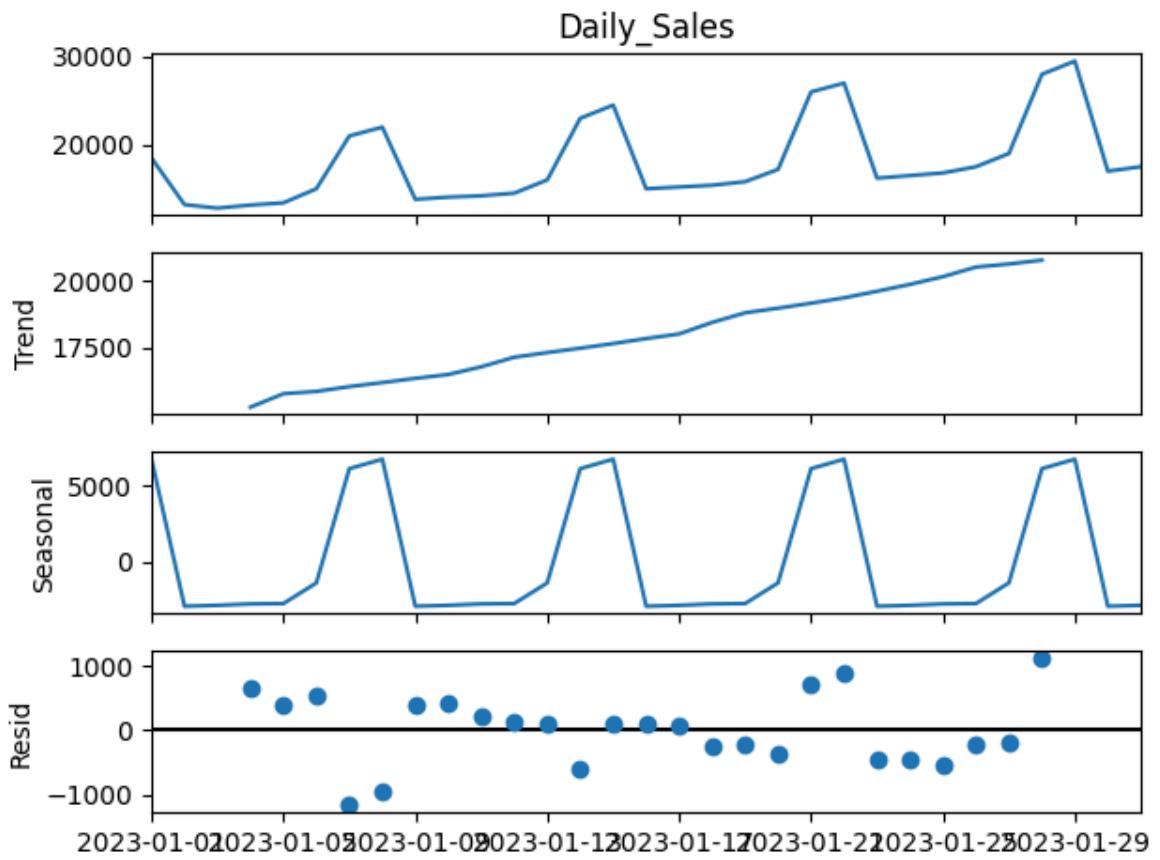
ts = data['Daily_Sales']

plt.figure(figsize=(12,6))
plt.plot(ts)
plt.title("Daily Sales Time Series")
plt.show()
```



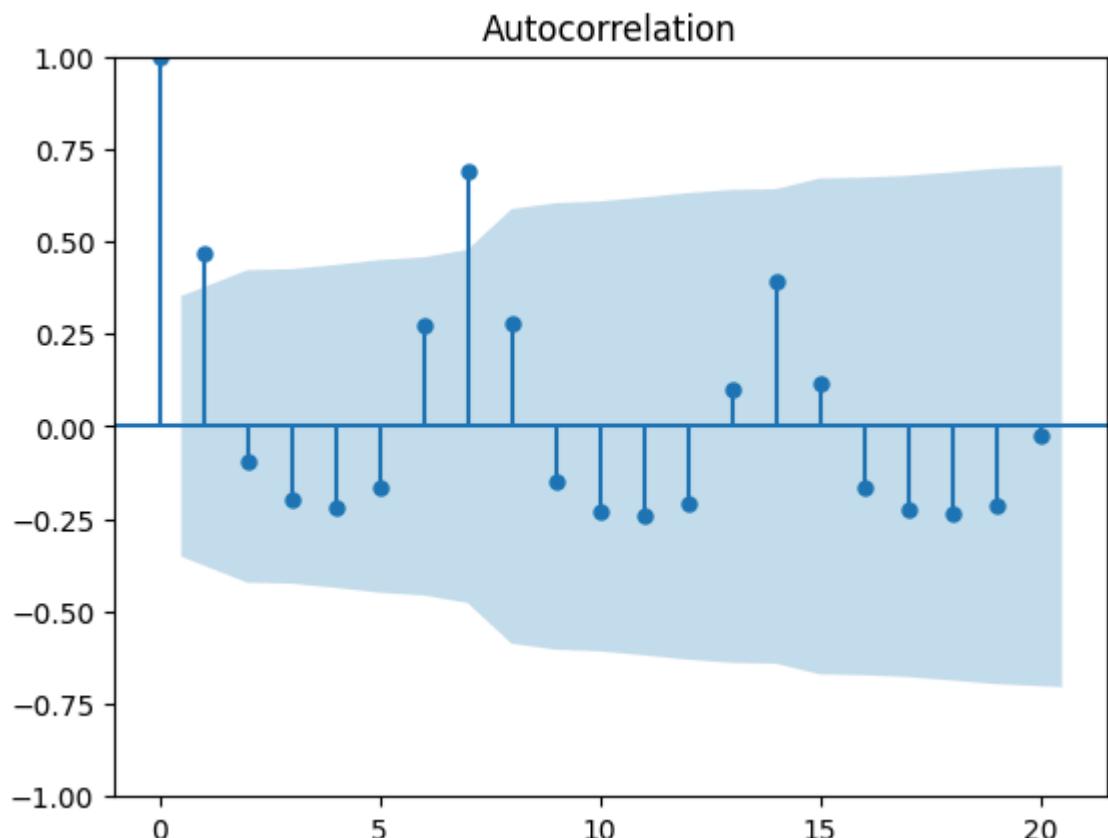
```
In [5]: decomposition = seasonal_decompose(ts, model='additive', period=7)

decomposition.plot()
plt.show()
```



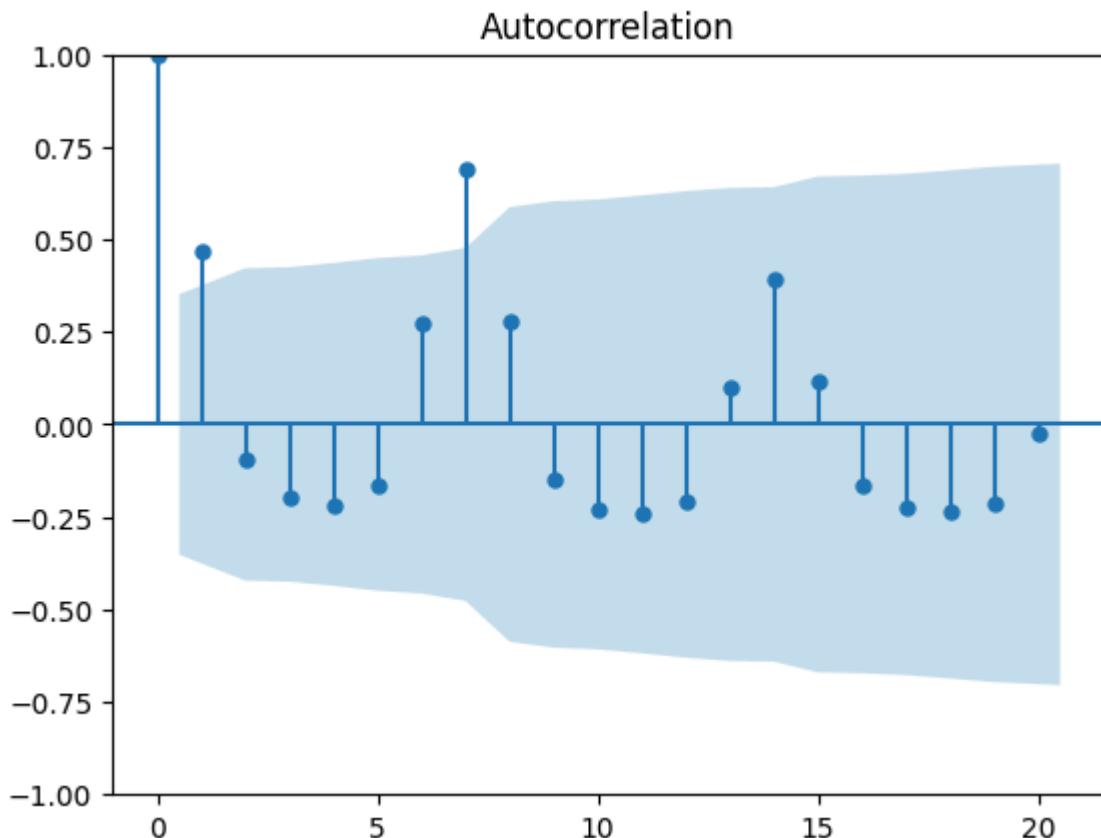
```
In [6]: plt.figure(figsize=(10,4))
plot_acf(ts, lags=20)
plt.show()
```

<Figure size 1000x400 with 0 Axes>



```
In [7]: plt.figure(figsize=(10,4))
plot_acf(ts, lags=20)
plt.show()
```

<Figure size 1000x400 with 0 Axes>



```
In [8]: result = adfuller(ts)

adf_stat = result[0]
p_value = result[1]
critical_values = result[4]

print("ADF Statistic:", adf_stat)
print("p-value:", p_value)
```

ADF Statistic: -0.01226606993787269

p-value: 0.9575058587612261

```
In [9]: for key, value in critical_values.items():
    print("Critical Value", key, ":", value)

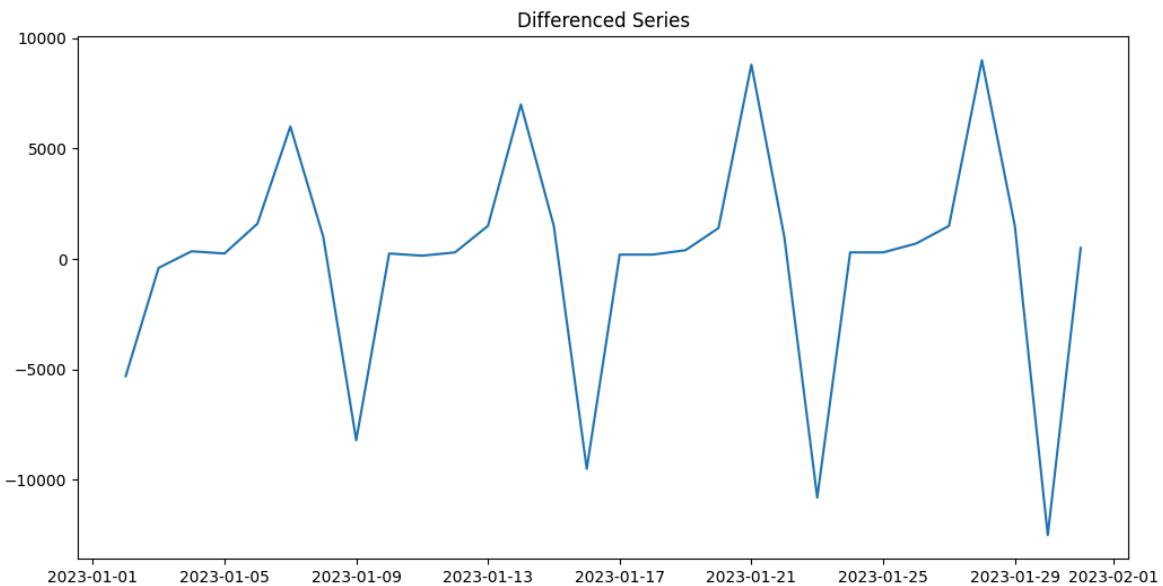
diff_ts = ts.diff().dropna()

plt.figure(figsize=(12,6))
plt.plot(diff_ts)
plt.title("Differenced Series")
plt.show()
```

Critical Value 1% : -3.7377092158564813

Critical Value 5% : -2.9922162731481485

Critical Value 10% : -2.635746736111111



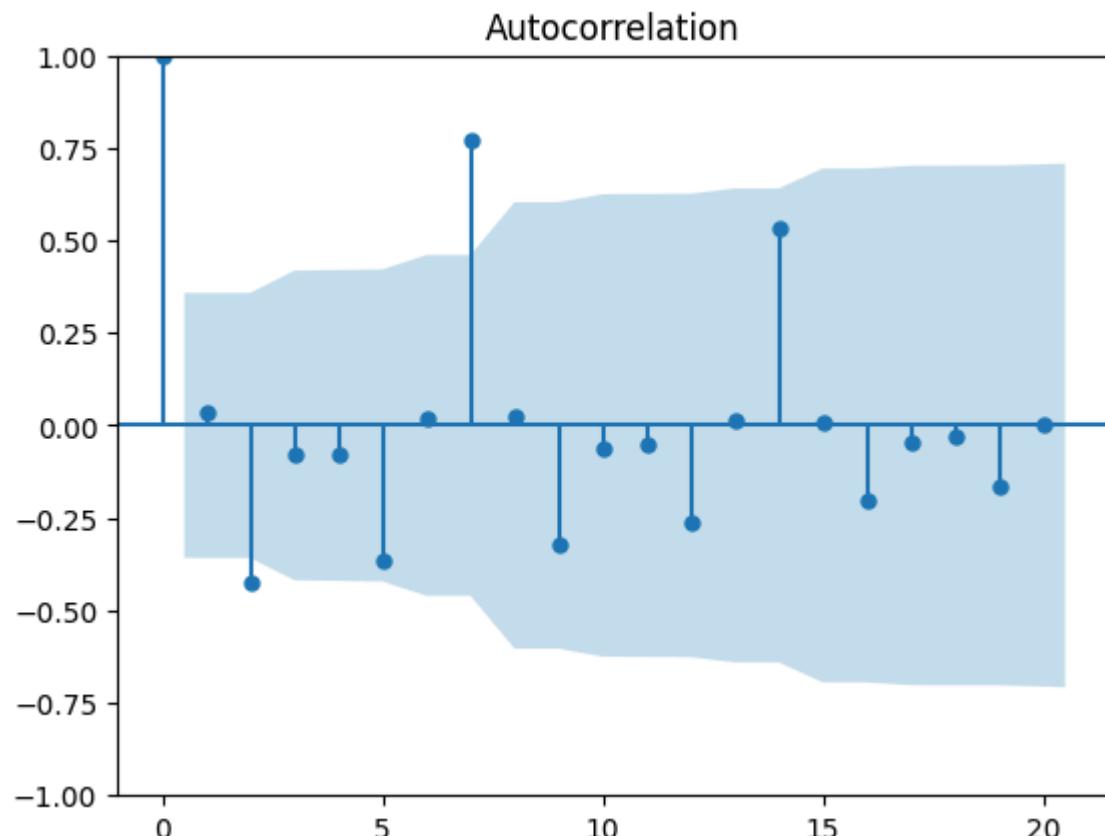
```
In [10]: result_diff = adfuller(diff_ts)

print("ADF Statistic After Differencing:", result_diff[0])
print("p-value After Differencing:", result_diff[1])
```

```
ADF Statistic After Differencing: -49.49101445793401
p-value After Differencing: 0.0
```

```
In [11]: plt.figure(figsize=(10,4))
plot_acf(diff_ts, lags=20)
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```



```
In [13]: plt.figure(figsize=(10,4))
plot_pacf(diff_ts, lags=14)
```

```
plt.show()
```

<Figure size 1000x400 with 0 Axes>

