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

from sklearn.metrics import mean_squared_error

from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.seasonal import seasonal_decompose
```

```
df_bax_m = pd.read_csv(r'/content/drive/MyDrive/PRN23039142546/df_bax_cleaned_to_view_outliers.csv', index_col=0,parse_dates=True)
df_bax_m.head()
```

Date	Price	Open	High	Low	Vol.	Change %
2010-05-24	1482.42	1491.98	1491.98	1482.42	926980.0	-0.64
2010-05-25	1454.85	1482.42	1482.42	1454.85	1660000.0	-1.86
2010-05-26	1472.29	1456.50	1472.29	1454.85	1500000.0	1.20
2010-05-27	1453.82	1472.29	1478.07	1453.82	2480000.0	-1.25
2010-05-30	1455.16	1453.82	1462.04	1453.72	5910000.0	0.09

Next steps: Generate code with df_bax_m | View recommended plots | New interactive sheet

```
df_bax_m.info() #check data types
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3659 entries, 2010-05-24 to 2025-03-27
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Price     3659 non-null   float64
 1   Open      3659 non-null   float64
 2   High      3659 non-null   float64
 3   Low       3659 non-null   float64
 4   Vol.      3659 non-null   float64
 5   Change %  3659 non-null   float64
dtypes: float64(6)
memory usage: 200.1 KB
```

```
df_naive = df_bax_m.copy()
```

```
price = df_naive['Price']
```

```
split_point = int(len(price)*0.8)
#price[split_point:]
train,test = price[:split_point],price[split_point:]
```

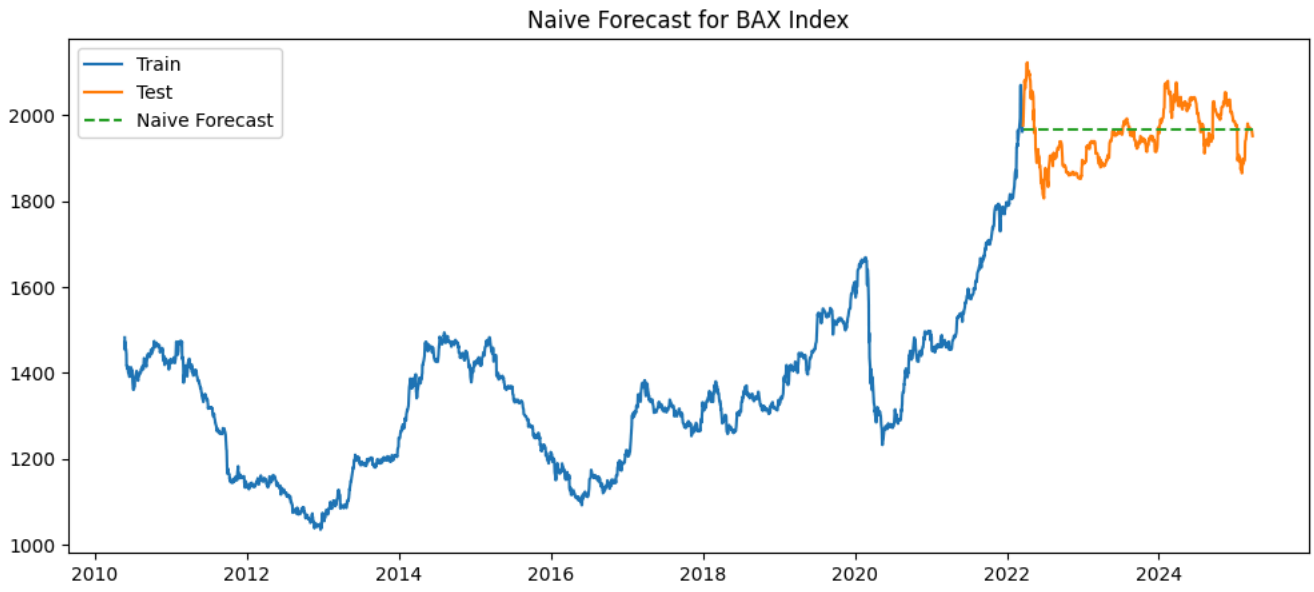
```
naive_forecast = pd.Series(train.iloc[-1], index=test.index)
```

```
rmse = np.sqrt(mean_squared_error(test, naive_forecast))
```

```
print(f"Naive Forecast - RMSE: {rmse:.2f}")
```

```
Naive Forecast - RMSE: 64.48
```

```
plt.figure(figsize=(12,5))
plt.plot(train, label='Train')
plt.plot(test, label='Test')
plt.plot(naive_forecast,label='Naive Forecast', linestyle='--')
plt.legend()
plt.title("Naive Forecast for BAX Index")
plt.show()
```



Check seasonality in BAX dataset

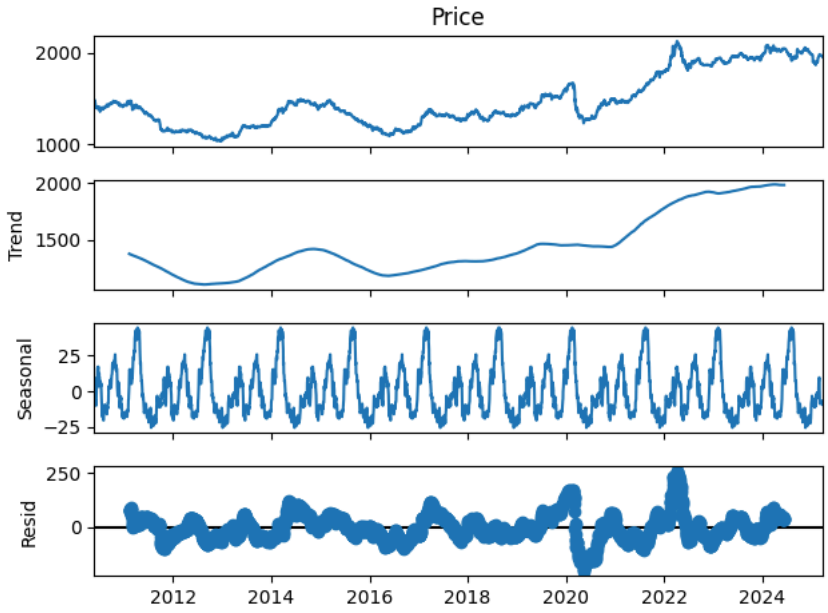
```
df = df_bax_m.copy()
```

```
series = df['Price']
```

```
# line plot

plt.figure(figsize=(12,4))
plt.plot(series)
plt.title("BAX Index - Price over time")
plt.xlabel("Data")
plt.ylabel("Price")
plt.grid(True)
plt.show()
```


```
# seasonal decomposition (using additive model for now)
decomposition = seasonal_decompose(series, model='additive', period=365) # Assuming daily data with yearly seasonality
decomposition.plot()
plt.tight_layout()
plt.show()
```




```
# autocorrelation plot
plot_acf(series, lags=400)
plt.xlim(340, 380) # Focus on 365 lag
#plt.title("Autocorrelation Plot (ACF)")
plt.title("Autocorrelation Plot (ACF) around 365 lag")
plt.show()
```