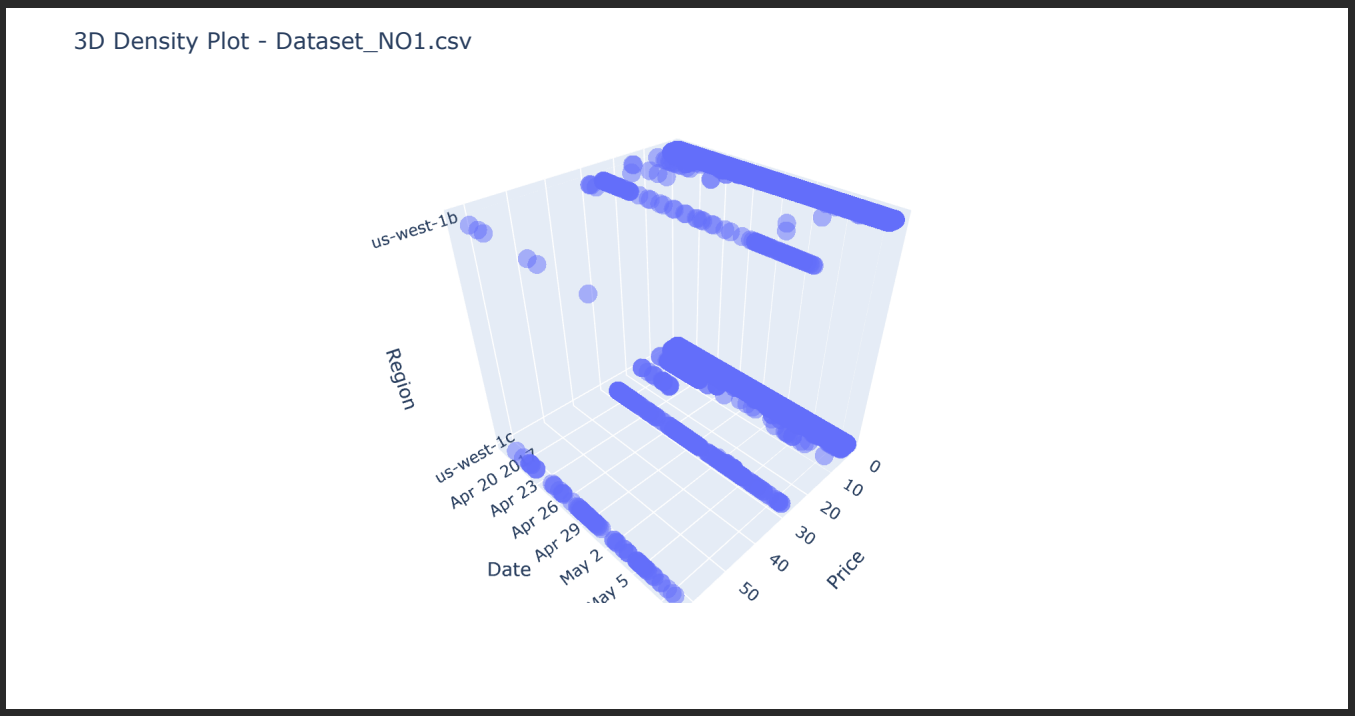
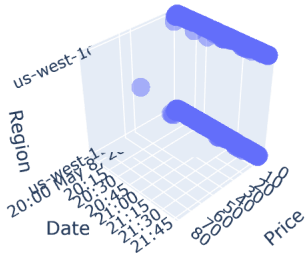
```
import os
import pandas as pd
import plotly.express as px

# List of file paths for the three datasets
dataset_paths = [
    r"Dataset_NO1.csv"
]

# Loop through each dataset and create a 3D density plot
for dataset_path in dataset_paths:
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Create a 3D density plot for "Price," "Date," and "Region"
    fig = px.scatter_3d(df, x='Price', y='Date', z='Region', opacity=0.5)

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    fig.update_layout(title=f'3D Density Plot - {filename}')

    # Show the plot
    fig.show()
```




```
import os
import pandas as pd
import plotly.express as px

# List of file paths for the three datasets
dataset_paths = [
    r"Dataset_NO2.csv"
]

# Loop through each dataset and create a 3D density plot
for dataset_path in dataset_paths:
    # Read a subset of the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'], nrows=1000)  # Adjust the number of rows as needed

    # Create a 3D density plot for "Price," "Date," and "Region"
    fig = px.scatter_3d(df, x='Price', y='Date', z='Region', opacity=0.5)

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    fig.update_layout(title=f'3D Density Plot - {filename}')

    # Show the plot
    fig.show()
```

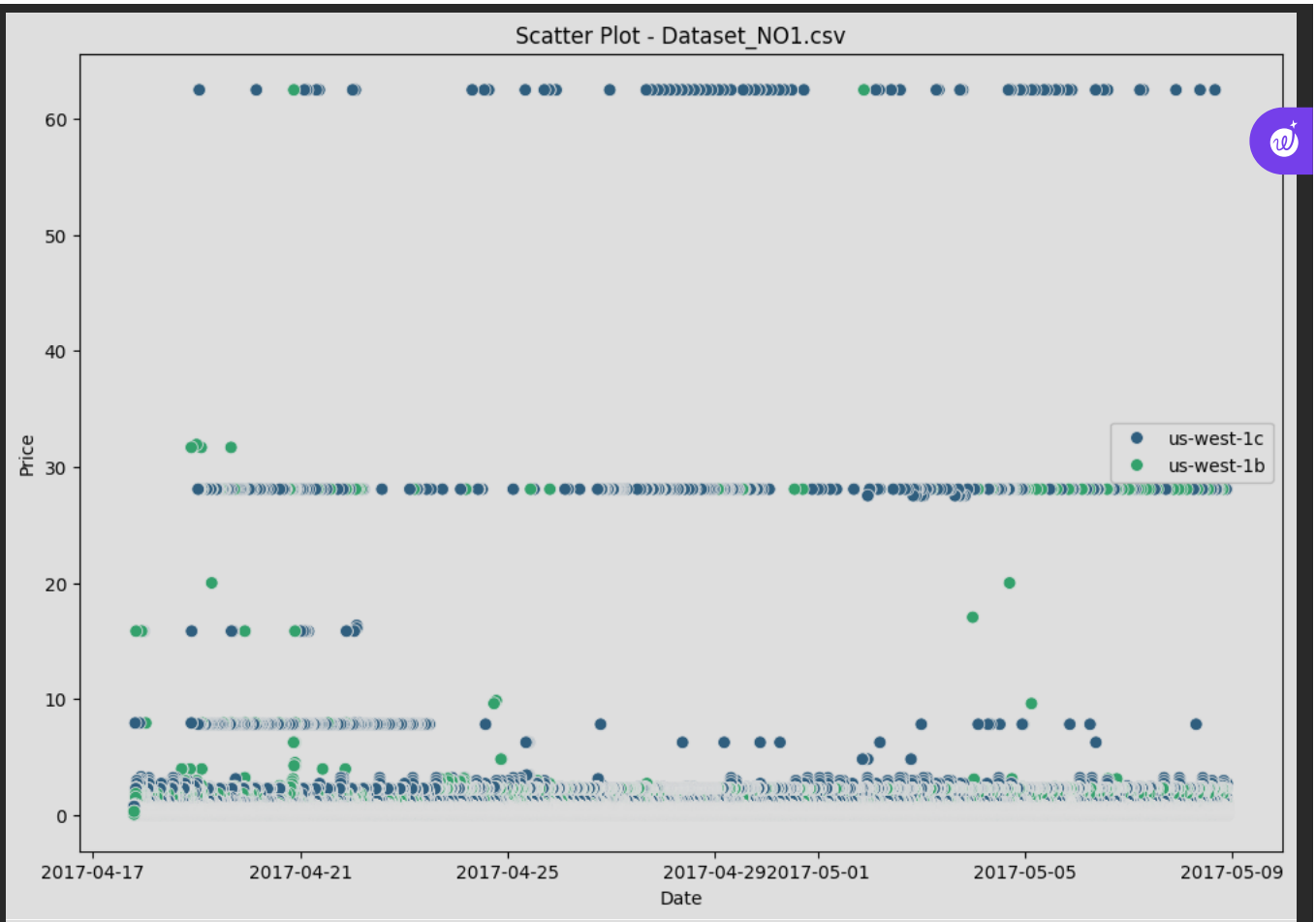
3D Density Plot - Dataset_NO2.csv

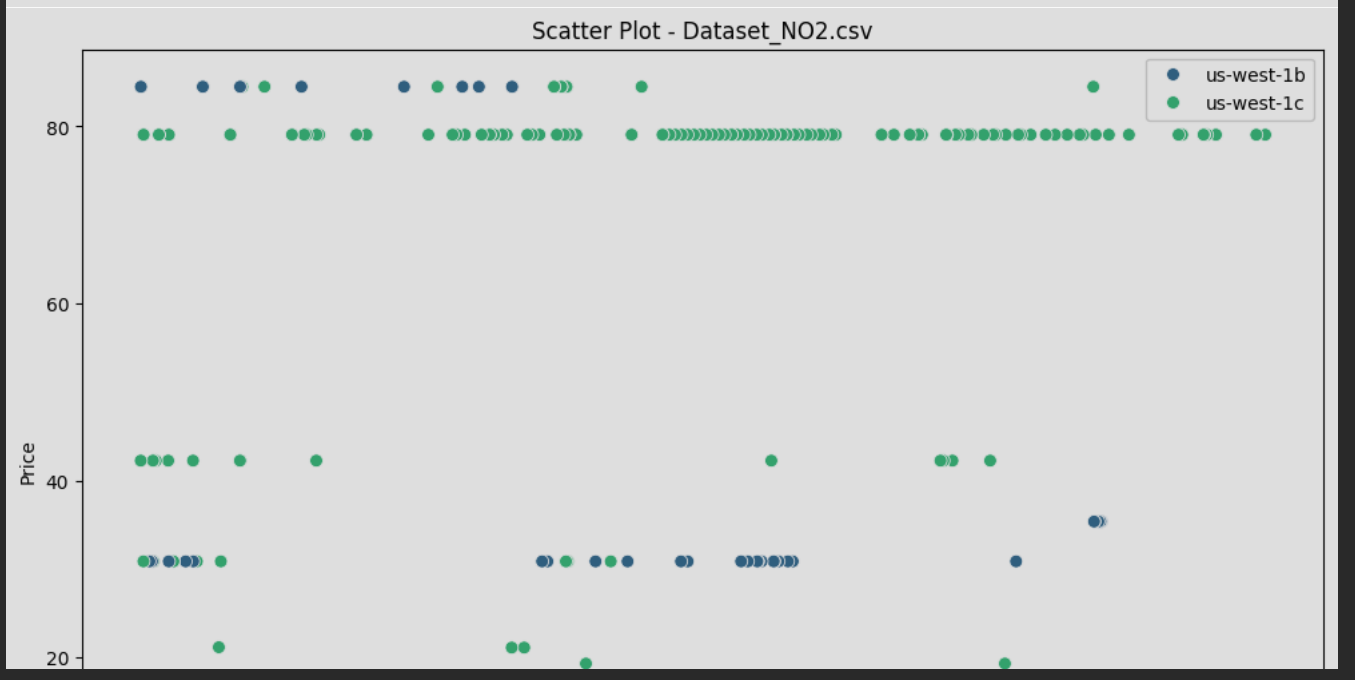



```
import os
import pandas as pd
import plotly.express as px

# List of file paths for the three datasets
dataset_paths = [
    r"Dataset_NO3.csv",
]

# Loop through each dataset and create a 3D density plot
for dataset_path in dataset_paths:
    # Read a subset of the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'], nrows=1000)  # Adjust the number of rows as needed

    # Create a 3D density plot for "Price," "Date," and "Region"
    fig = px.scatter_3d(df, x='Price', y='Date', z='Region', opacity=0.5)

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    fig.update_layout(title=f'3D Density Plot - {filename}')

    # Show the plot
    fig.show()
```

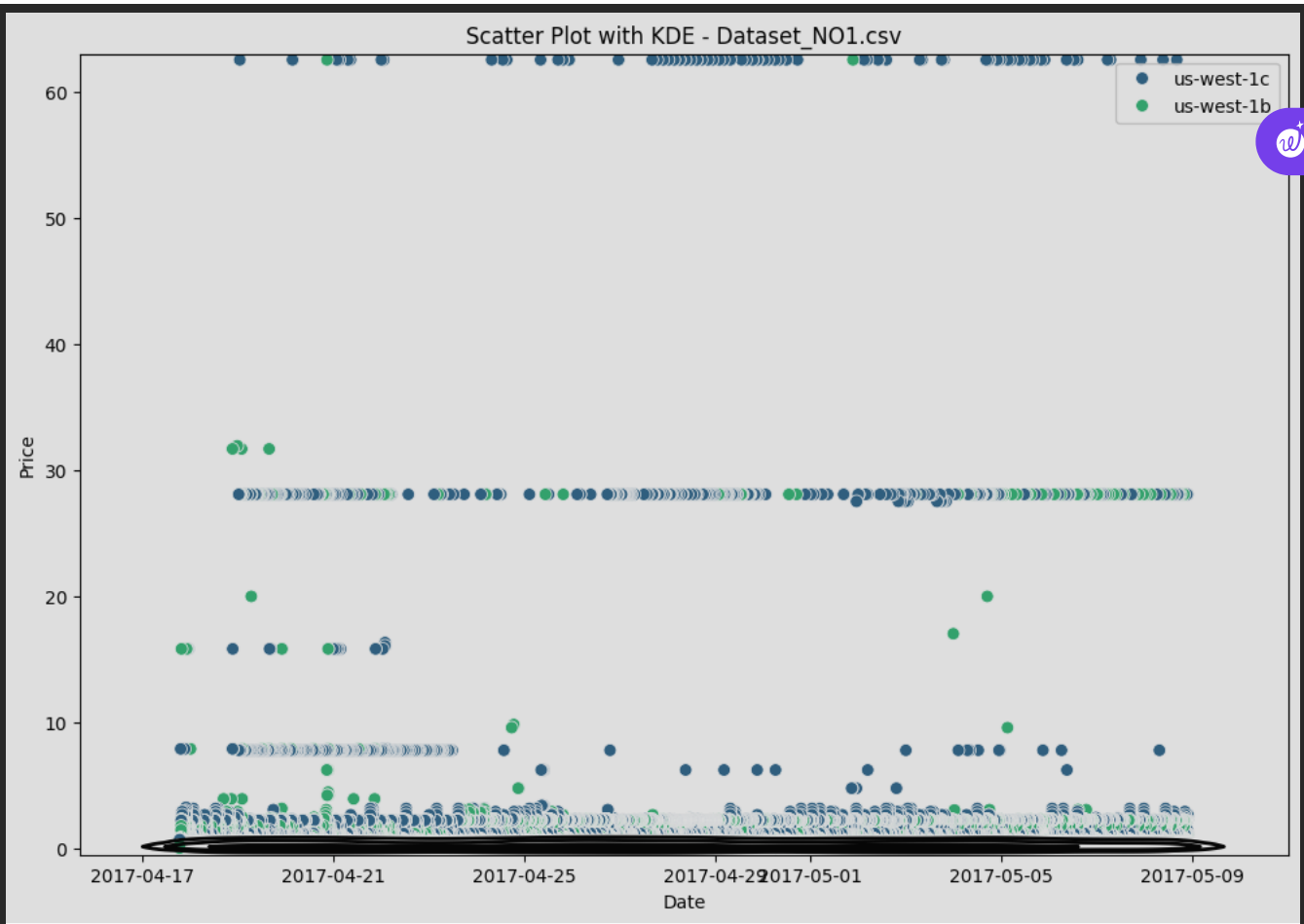
3D Density Plot - Dataset_NO3.csv

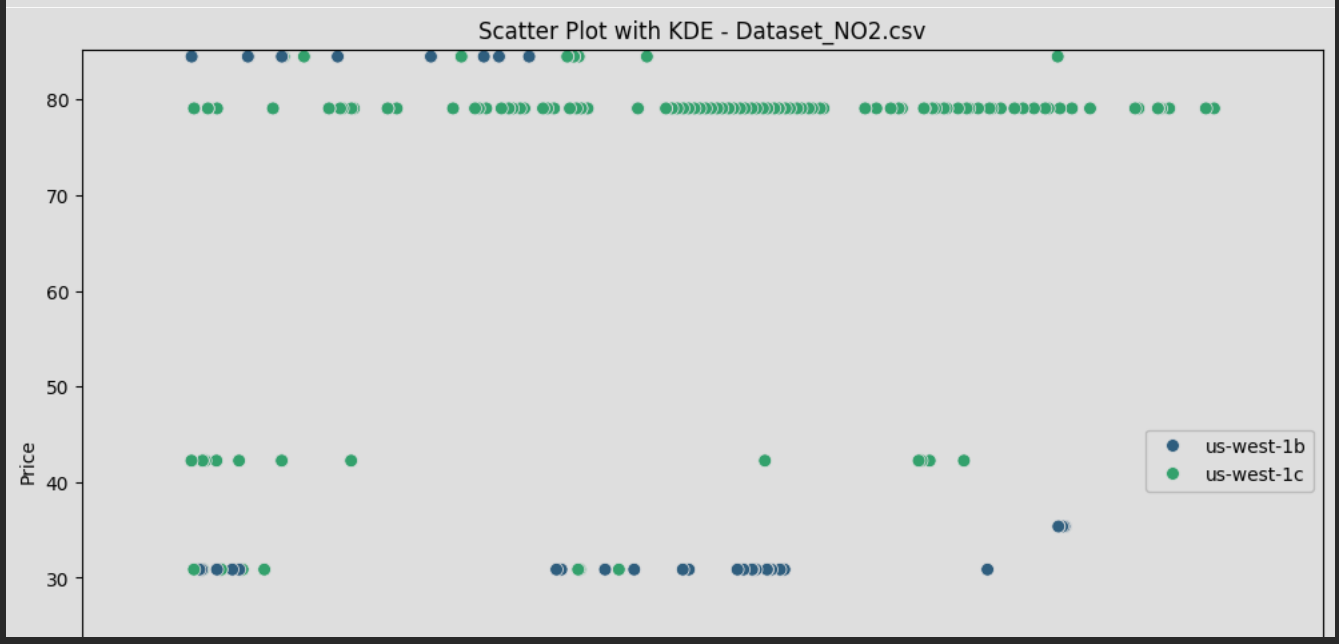
```
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create a scatter plot
for i, dataset_path in enumerate(dataset_paths, 1):
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Create a scatter plot
    plt.figure(figsize=(12, 8))
    sns.scatterplot(data=df, x='Date', y='Price', hue='Region', palette='viridis', s=50)

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Scatter Plot - {filename}')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.show()
```


Scatter Plot - Dataset_NO1.csv
Price
Date
us-west-1c
us-west-1b
0
10
20
30
40
50
60
2017-04-17
2017-04-21
2017-04-25
2017-04-29
2017-05-01
2017-05-05
2017-05-09

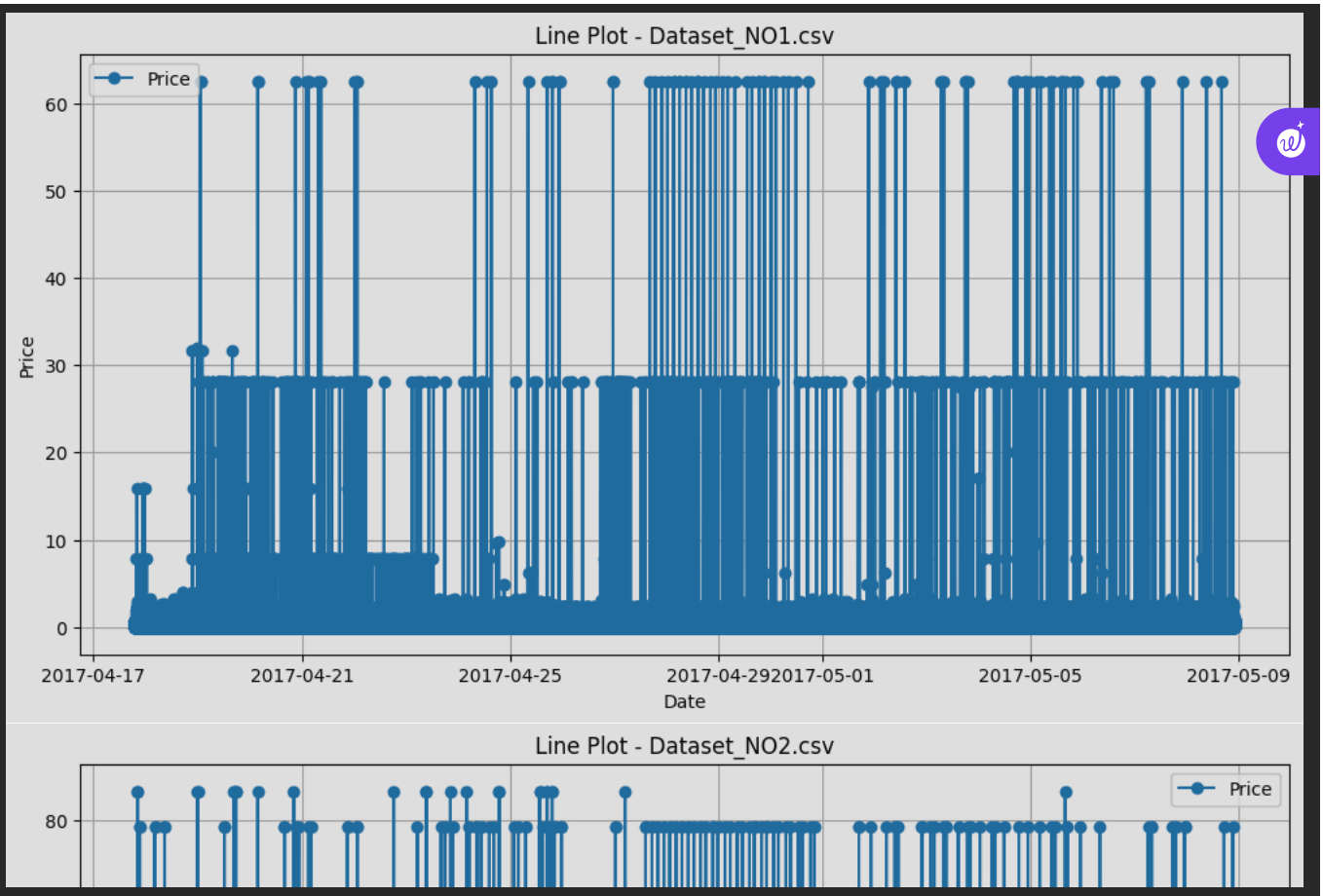

Scatter Plot - Dataset_NO2.csv
Price
us-west-1b
us-west-1c
20
40
60
80

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

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create a scatter plot with normal distribution
for i, dataset_path in enumerate(dataset_paths, 1):
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Create a scatter plot with KDE
    plt.figure(figsize=(12, 8))
    sns.scatterplot(data=df, x='Date', y='Price', hue='Region', palette='viridis', s=50)
    sns.kdeplot(data=df, x='Date', y='Price', fill=False, levels=5, color='black', linewidths=2)

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Scatter Plot with KDE - {filename}')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.show()
```

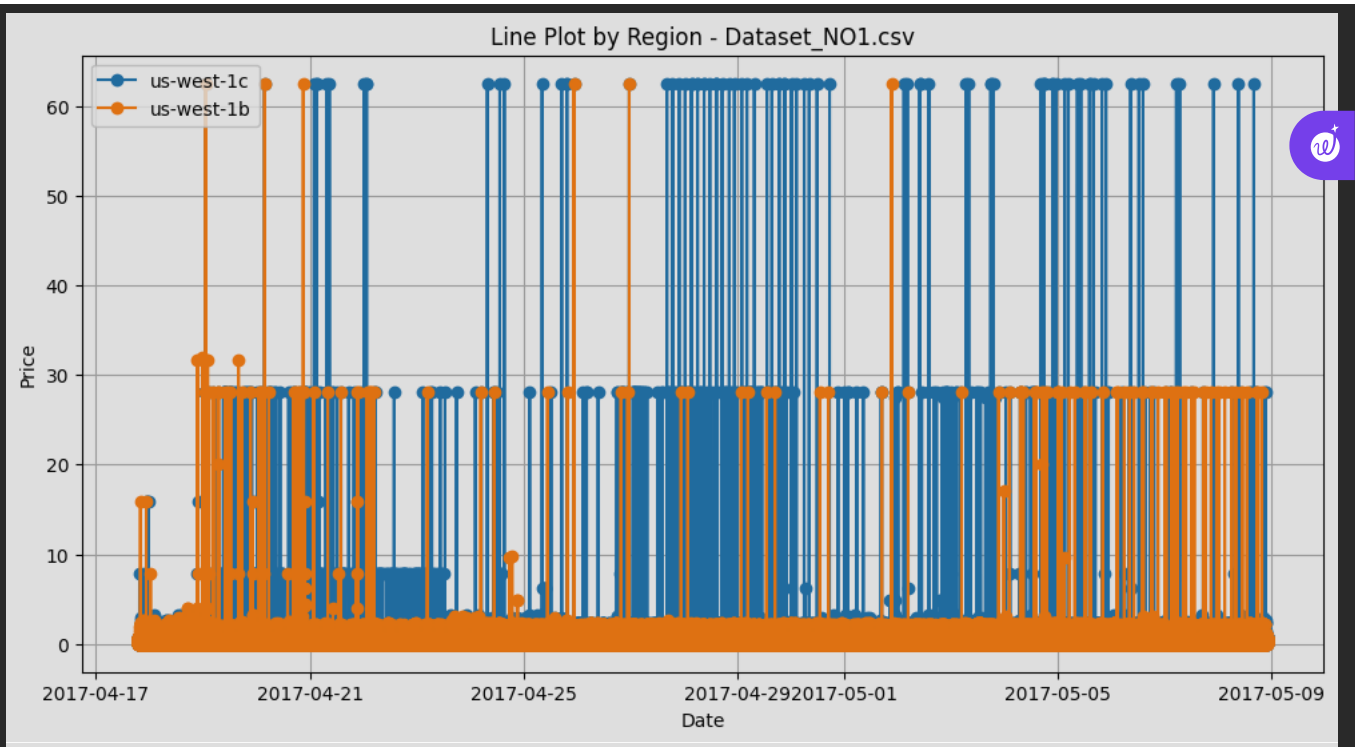

Scatter Plot with KDE - Dataset_NO1.csv
us-west-1c
us-west-1b
Price
Date
0
10
20
30
40
50
60
2017-04-17
2017-04-21
2017-04-25
2017-04-29
2017-05-01
2017-05-05
2017-05-09

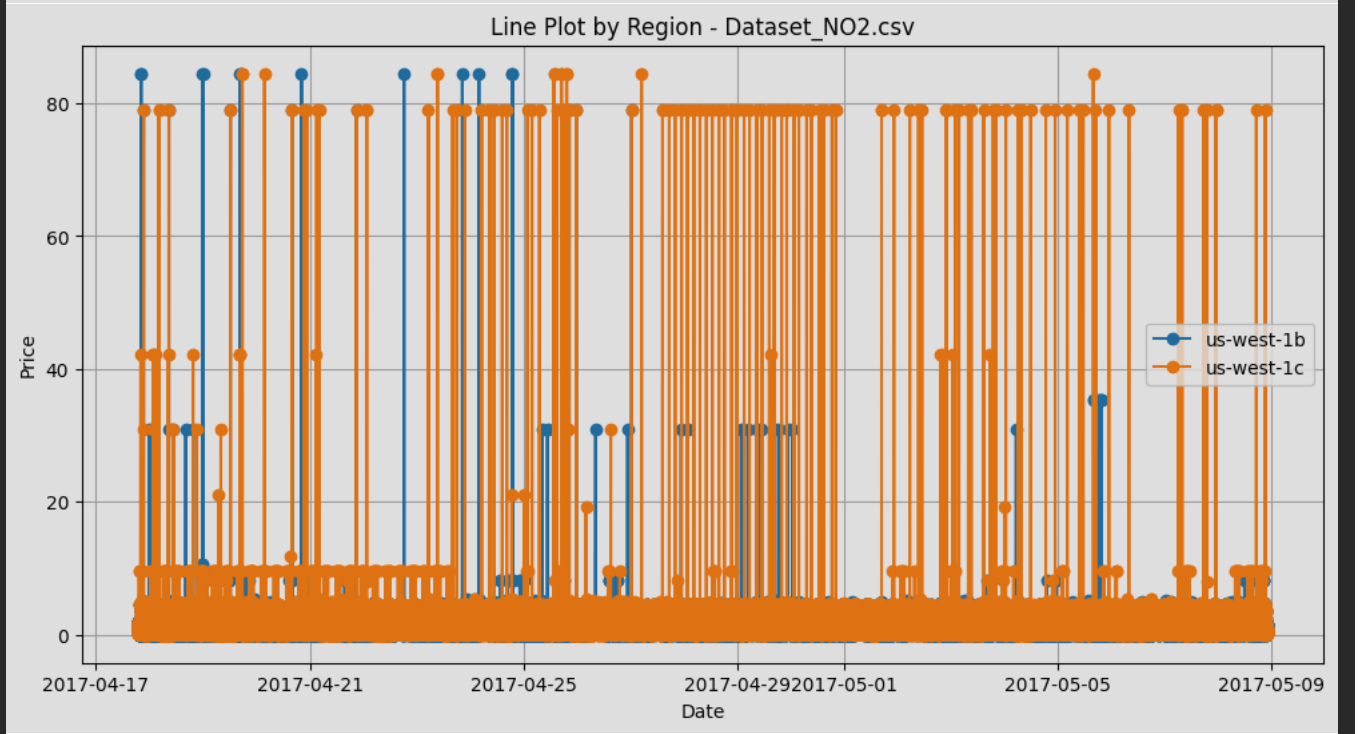

Scatter Plot with KDE - Dataset_NO2.csv
us-west-1b
us-west-1c
Price
30
40
50
60
70
80

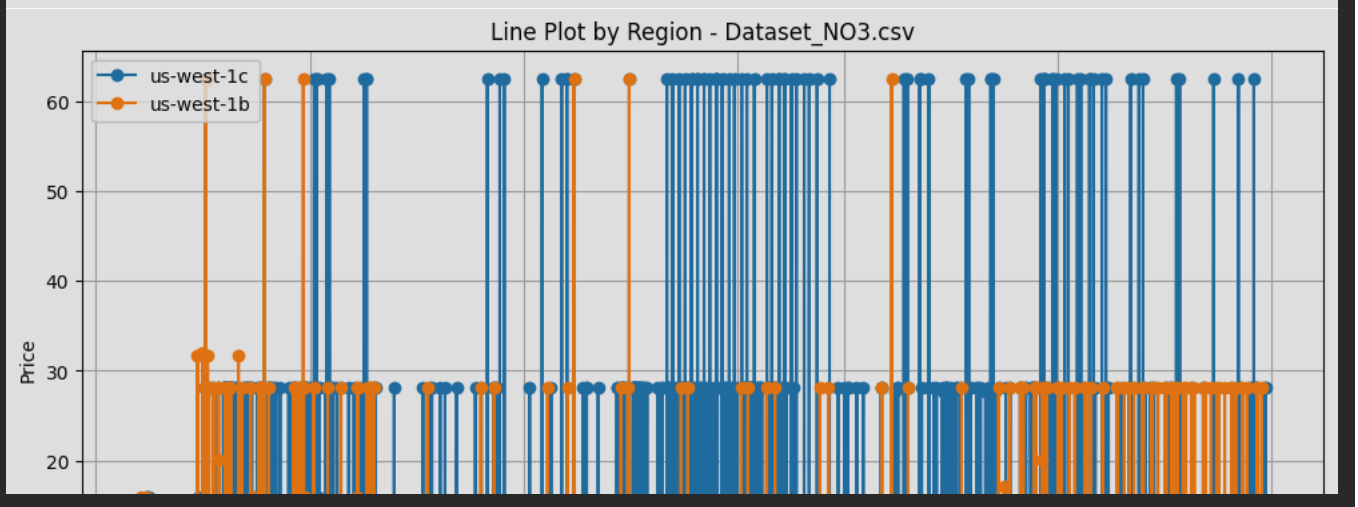
```
import os
import pandas as pd
import matplotlib.pyplot as plt

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservat
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create line plots
for dataset_path in dataset_paths:
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Plot line chart for 'Date' and 'Price'
    plt.figure(figsize=(12, 6))
    plt.plot(df['Date'], df['Price'], label='Price', marker='o', linestyle='-')

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Line Plot - {filename}')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.grid(True)
    plt.show()
```



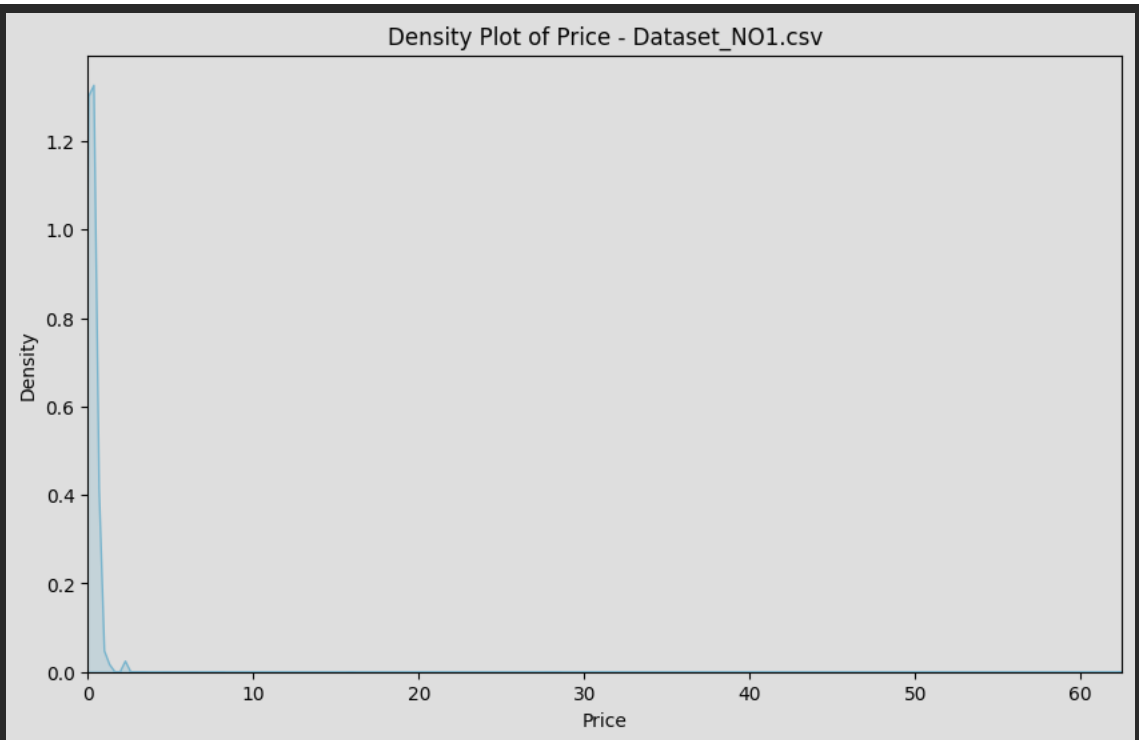

```
import os
import pandas as pd
import matplotlib.pyplot as plt

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create line plots for each region
for dataset_path in dataset_paths:
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Get unique regions in the dataset
    regions = df['Region'].unique()

    # Plot line chart for each region
    plt.figure(figsize=(12, 6))
    for region in regions:
        region_data = df[df['Region'] == region]
        plt.plot(region_data['Date'], region_data['Price'], label=region, marker='o', linestyle='-')

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Line Plot by Region - {filename}')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.grid(True)
    plt.show()
```

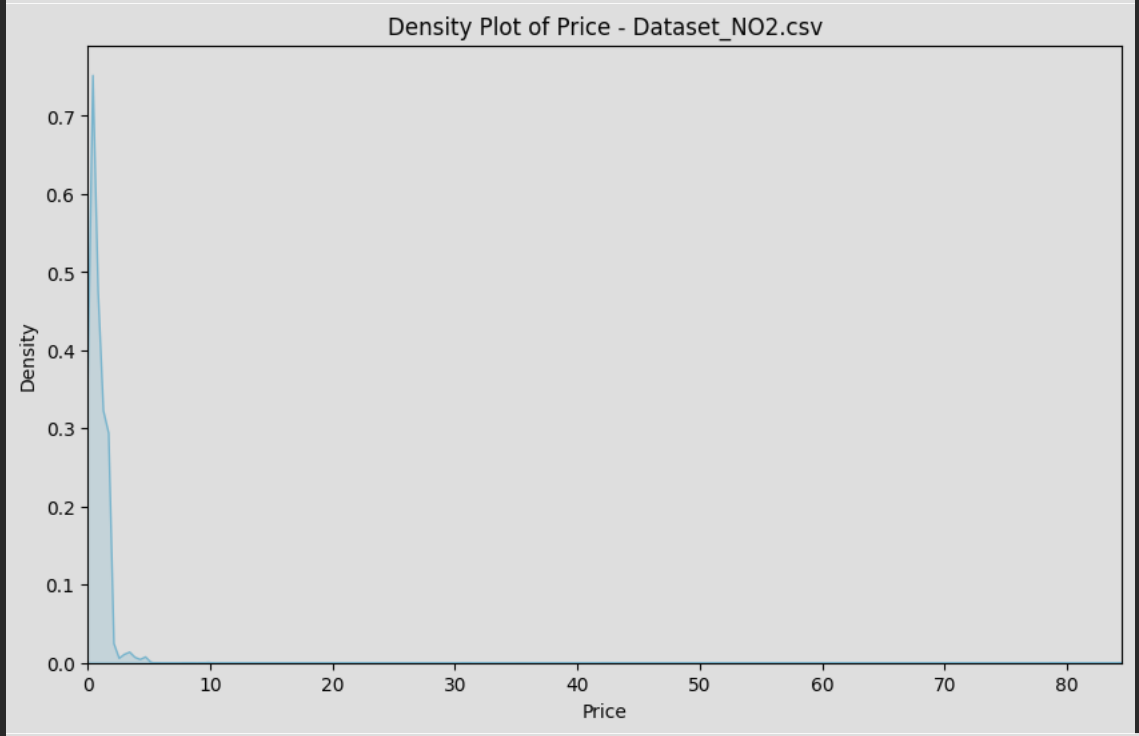

Line Plot by Region - Dataset_NO1.csv
us-west-1c
us-west-1b
Price
Date
2017-04-17
2017-04-21
2017-04-25
2017-04-292017-05-01
2017-05-05
2017-05-09

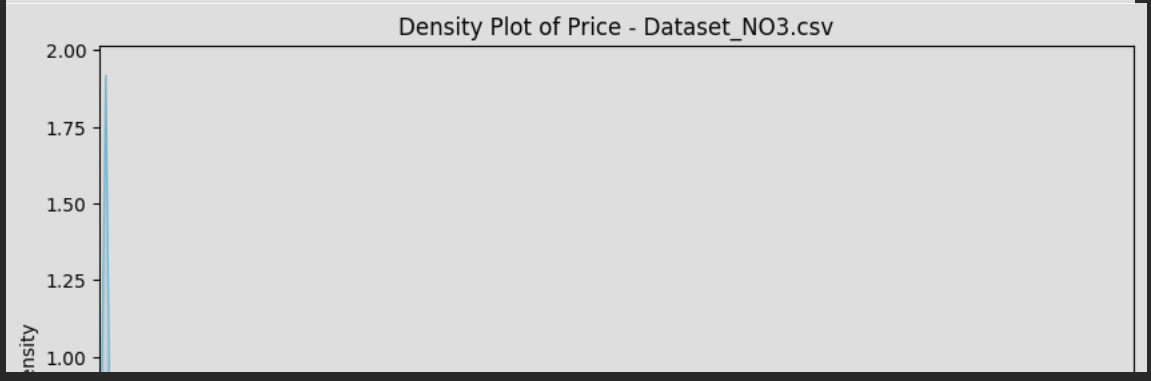

Line Plot by Region - Dataset_NO2.csv
us-west-1b
us-west-1c
Price
Date
2017-04-17
2017-04-21
2017-04-25
2017-04-292017-05-01
2017-05-05
2017-05-09


Line Plot by Region - Dataset_NO3.csv
us-west-1c
us-west-1b
Price

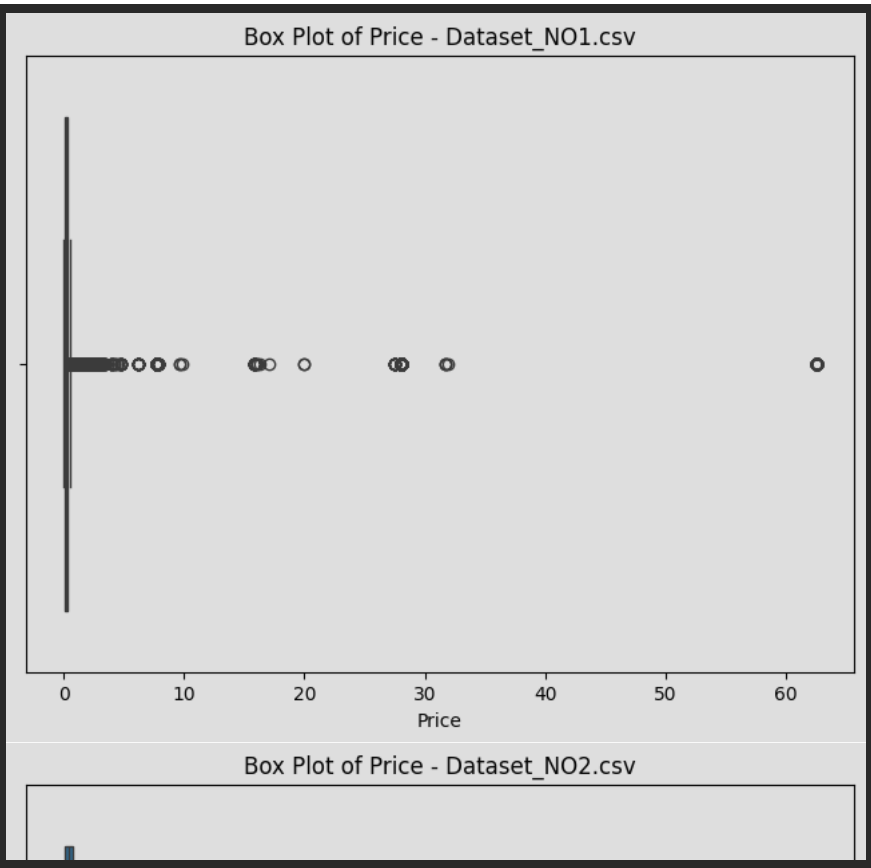
```
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create density plots
for dataset_path in dataset_paths:
    # Read the data from the CSV fil
    #e
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Create a KDE plot for the 'Price' column
    plt.figure(figsize=(10, 6))
    sns.kdeplot(data=df, x='Price', fill=True, common_norm=False, color='skyblue')

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Density Plot of Price - {filename}')
    plt.xlabel('Price')
    plt.ylabel('Density')
    plt.show()



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

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create density plots
for dataset_path in dataset_paths:
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path, parse_dates=['Date'])

    # Set the maximum value of the x-axis to the maximum value in the "Price" column
    max_price = df['Price'].max()

    # Create a KDE plot for the 'Price' column
    plt.figure(figsize=(10, 6))
    sns.kdeplot(data=df, x='Price', fill=True, common_norm=False, color='skyblue')

    # Set the x-axis limit to the maximum value in the "Price" column
    plt.xlim(0, max_price)

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Density Plot of Price - {filename}')
    plt.xlabel('Price')
    plt.ylabel('Density')
    plt.show()
```


Density Plot of Price - Dataset_NO1.csv
Density
Price
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
10
20
30
40
50
60


Density Plot of Price - Dataset_NO2.csv
Density
Price
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
60
70
80


Density Plot of Price - Dataset_NO3.csv
1.00
1.25
1.50
1.75
2.00

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

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create a box plot for the 'Price' column
for dataset_path in dataset_paths:
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path)

    # Create a box plot for the 'Price' column
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=df['Price'])

    # Use os.path.basename to get the filename without the path
    filename = os.path.basename(dataset_path)

    plt.title(f'Box Plot of Price - {filename}')
    plt.xlabel('Price')
    plt.show()
```




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

# List of file paths for the three datasets
dataset_paths = [
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
    r"C:\Users\kevin\OneDrive\Desktop\code_fixed\Optimal-timing-for-bandwith-reservation-main\optimal_timing_for_bandwidth_reservation\
]

# Loop through each dataset and create a box plot for the 'Price' column using Seaborn displot
for dataset_path in dataset_paths:
    # Read the data from the CSV file
    df = pd.read_csv(dataset_path)

    # Create a box plot for the 'Price' column using Seaborn displot
    plt.figure(figsize=(8, 6))
    sns.histplot(df['Price'], kde=True, color='skyblue', stat='density')
    plt.title(f'Distribution Plot of Price - {os.path.basename(dataset_path)}')
    plt.xlabel('Price')
    plt.ylabel('Density')
    plt.show()
```