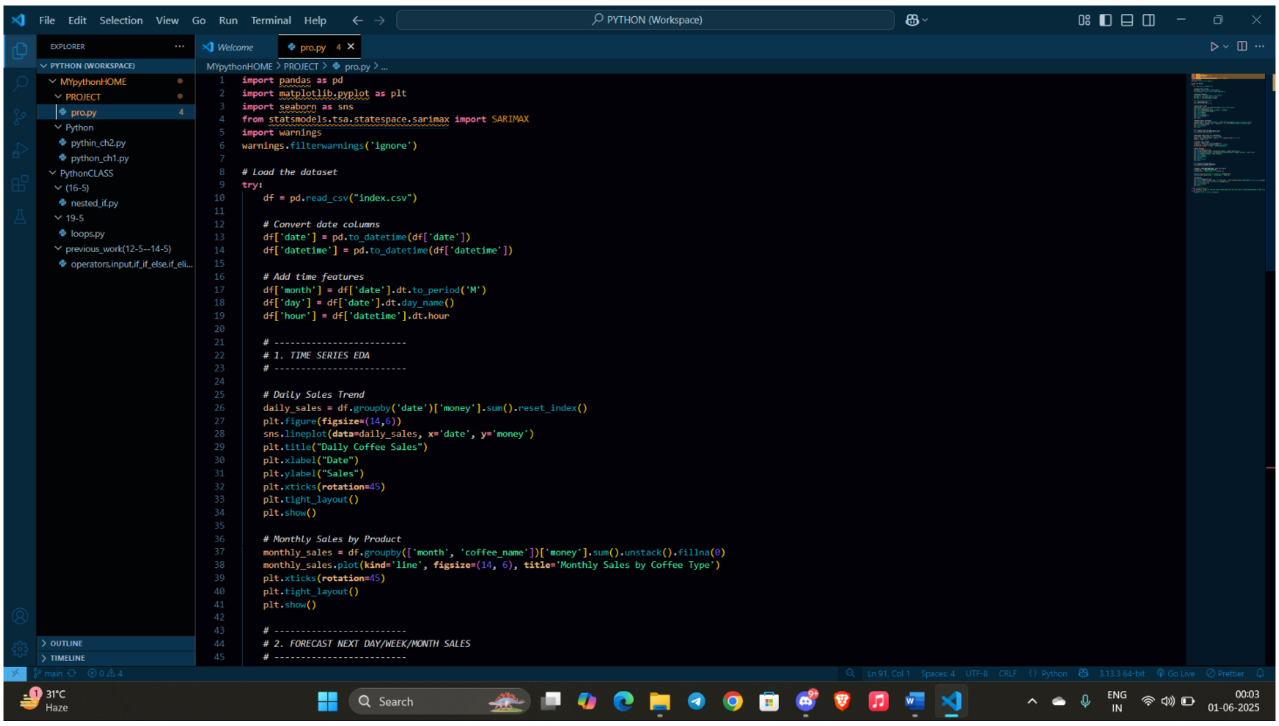
Coffee sales

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from statsmodels.tsa.statespace.sarimax import SARIMAX

import warnings

warnings.filterwarnings('ignore')


# Load the dataset

try:

    df = pd.read_csv("index.csv")


    # Convert date columns

    df['date'] = pd.to_datetime(df['date'])

    df['datetime'] = pd.to_datetime(df['datetime'])


    # Add time features

    df['month'] = df['date'].dt.to_period('M')

    df['day'] = df['date'].dt.day_name()

    df['hour'] = df['datetime'].dt.hour


    #-------------------------

    # 1. TIME SERIES EDA

    #-------------------------


    # Daily Sales Trend

    daily_sales = df.groupby('date')['money'].sum().reset_index()

    plt.figure(figsize=(14,6))
```

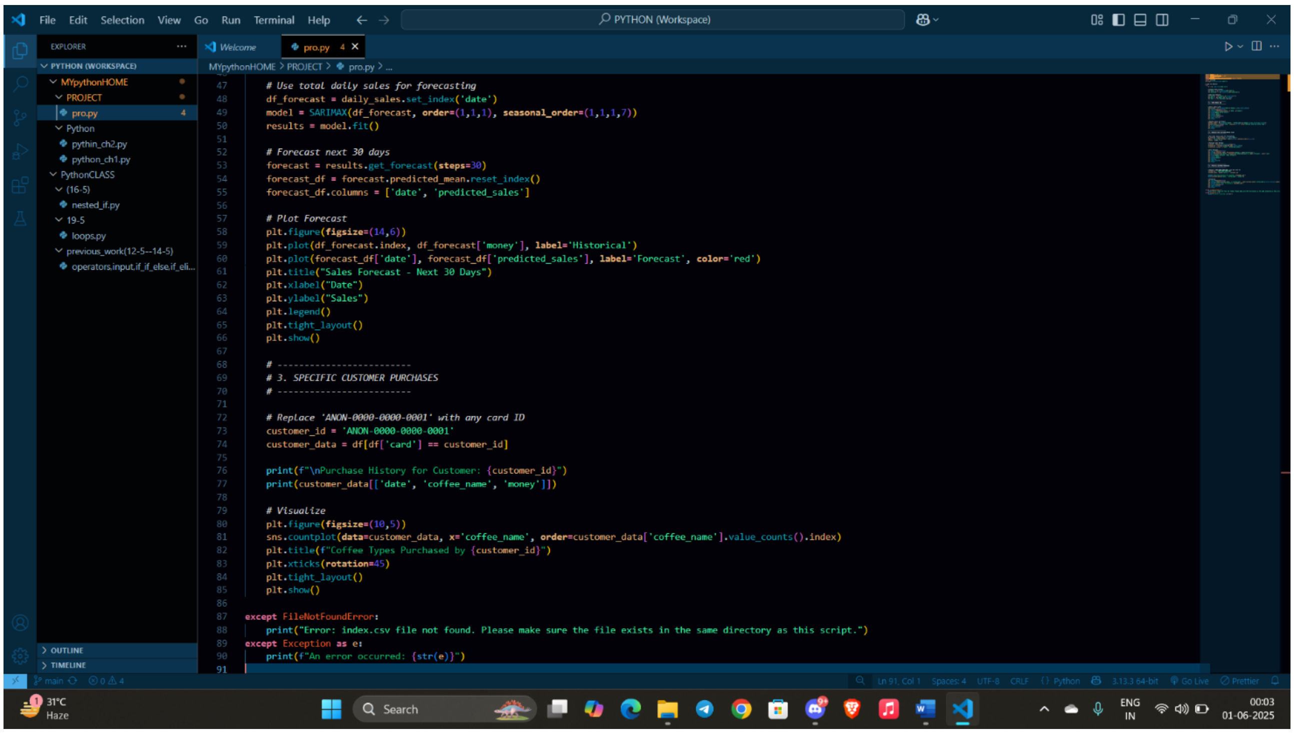
```
sns.lineplot(data=daily_sales, x='date', y='money')
plt.title("Daily Coffee Sales")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Monthly Sales by Product
monthly_sales = df.groupby(['month', 'coffee_name'])['money'].sum().unstack().fillna(0)
monthly_sales.plot(kind='line', figsize=(14, 6), title='Monthly Sales by Coffee Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

#-------------------------
# 2. FORECAST NEXT DAY/WEEK/MONTH SALES
#-------------------------

# Use total daily sales for forecasting
df_forecast = daily_sales.set_index('date')
model = SARIMAX(df_forecast, order=(1,1,1), seasonal_order=(1,1,1,7))
results = model.fit()

# Forecast next 30 days
forecast = results.get_forecast(steps=30)
forecast_df = forecast.predicted_mean.reset_index()
forecast_df.columns = ['date', 'predicted_sales']
```

```
# Plot Forecast

plt.figure(figsize=(14,6))

plt.plot(df_forecast.index, df_forecast['money'], label='Historical')

plt.plot(forecast_df['date'], forecast_df['predicted_sales'], label='Forecast', color='red')

plt.title("Sales Forecast- Next 30 Days")

plt.xlabel("Date")

plt.ylabel("Sales")

plt.legend()

plt.tight_layout()

plt.show()


#-------------------------

# 3. SPECIFIC CUSTOMER PURCHASES

#-------------------------


# Replace 'ANON-0000-0000-0001' with any card ID

customer_id = 'ANON-0000-0000-0001'

customer_data = df[df['card'] == customer_id]


print(f"\nPurchase History for Customer: {customer_id}")

print(customer_data[['date', 'coffee_name', 'money']])


# Visualize

plt.figure(figsize=(10,5))

sns.countplot(data=customer_data, x='coffee_name',
order=customer_data['coffee_name'].value_counts().index)

plt.title(f"Coffee Types Purchased by {customer_id}")
```

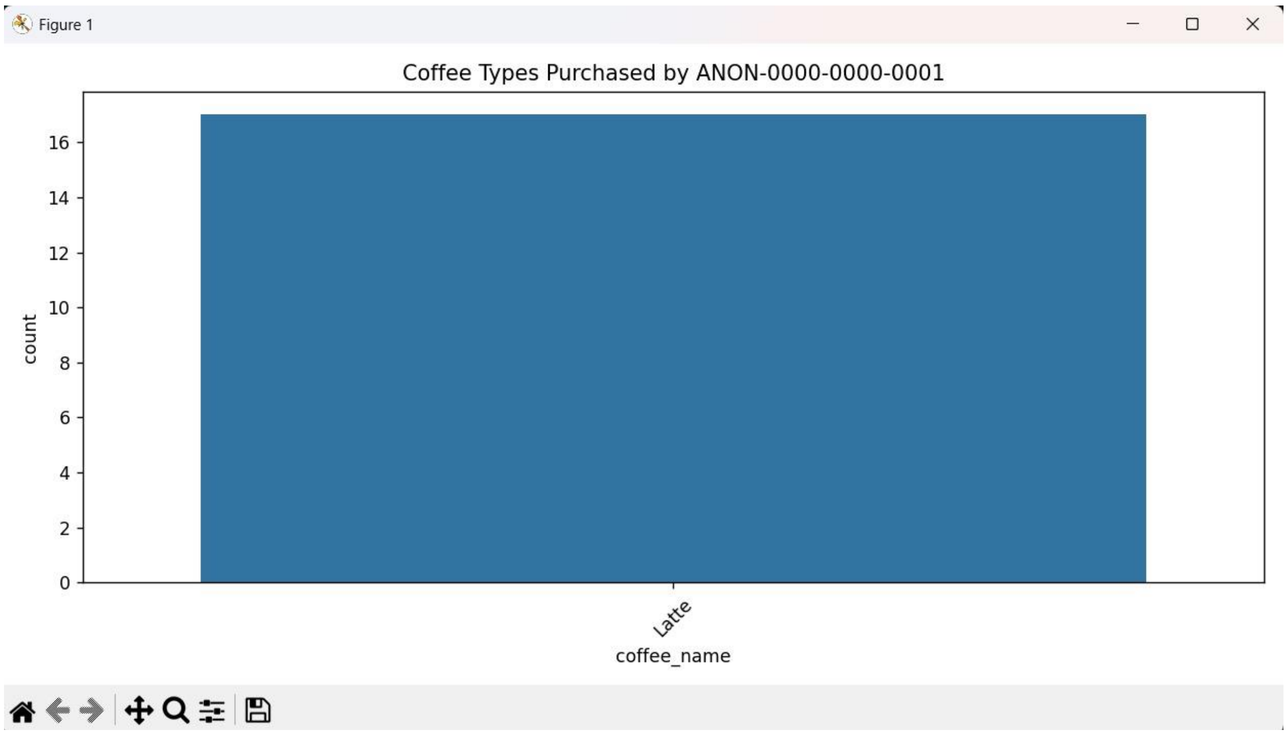
```python
    plt.xticks(rotation=45)

    plt.tight_layout()

    plt.show()


except FileNotFoundError:

    print("Error: index.csv file not found. Please make sure the file exists in the same directory as this script.")

except Exception as e:

    print(f"An error occurred: {str(e)}")
```



```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.statespace.sarimax import SARIMAX
import warnings
warnings.filterwarnings('ignore')

# Load the dataset
try:
    df = pd.read_csv("index.csv")

    # Convert date columns
    df['date'] = pd.to_datetime(df['date'])
    df['datetime'] = pd.to_datetime(df['datetime'])

    # Add time features
    df['month'] = df['date'].dt.to_period('M')
    df['day'] = df['date'].dt.day_name()
    df['hour'] = df['datetime'].dt.hour

    # ------------------------
    # 1. TIME SERIES EDA
    # ------------------------

    # Daily Sales Trend
    daily_sales = df.groupby('date')['money'].sum().reset_index()
    plt.figure(figsize=(14,6))
    sns.lineplot(data=daily_sales, x='date', y='money')
    plt.title("Daily Coffee Sales")
    plt.xlabel("Date")
    plt.ylabel("Sales")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

    # Monthly Sales by Product
    monthly_sales = df.groupby(['month', 'coffee_name'])['money'].sum().unstack().fillna(0)
    monthly_sales.plot(kind='line', figsize=(14, 6), title='Monthly Sales by Coffee Type')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

    # ------------------------
    # 2. FORECAST NEXT DAY/WEEK/MONTH SALES
    # ------------------------
```



```python
    # Use total daily sales for forecasting
    df_forecast = daily_sales.set_index('date')
    model = SARIMAX(df_forecast, order=(1,1,1), seasonal_order=(1,1,1,7))
    results = model.fit()

    # Forecast next 30 days
    forecast = results.get_forecast(steps=30)
    forecast_df = forecast.predicted_mean.reset_index()
    forecast_df.columns = ['date', 'predicted_sales']

    # Plot Forecast
    plt.figure(figsize=(14,6))
    plt.plot(df_forecast.index, df_forecast['money'], label='Historical')
    plt.plot(forecast_df['date'], forecast_df['predicted_sales'], label='Forecast', color='red')
    plt.title("Sales Forecast - Next 30 Days")
    plt.xlabel("Date")
    plt.ylabel("Sales")
    plt.legend()
    plt.tight_layout()
    plt.show()

    # ------------------------
    # 3. SPECIFIC CUSTOMER PURCHASES
    # ------------------------

    # Replace 'ANON-0000-0000-0001' with any card ID
    customer_id = 'ANON-0000-0000-0001'
    customer_data = df[df['card'] == customer_id]

    print(f"\nPurchase History for Customer: {customer_id}")
    print(customer_data[['date', 'coffee_name', 'money']])

    # Visualize
    plt.figure(figsize=(10,5))
    sns.countplot(data=customer_data, x='coffee_name', order=customer_data['coffee_name'].value_counts().index)
    plt.title(f"Coffee Types Purchased by {customer_id}")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

except FileNotFoundError:
    print("Error: index.csv file not found. Please make sure the file exists in the same directory as this script.")
except Exception as e:
    print(f"An error occurred: {str(e)}")
```



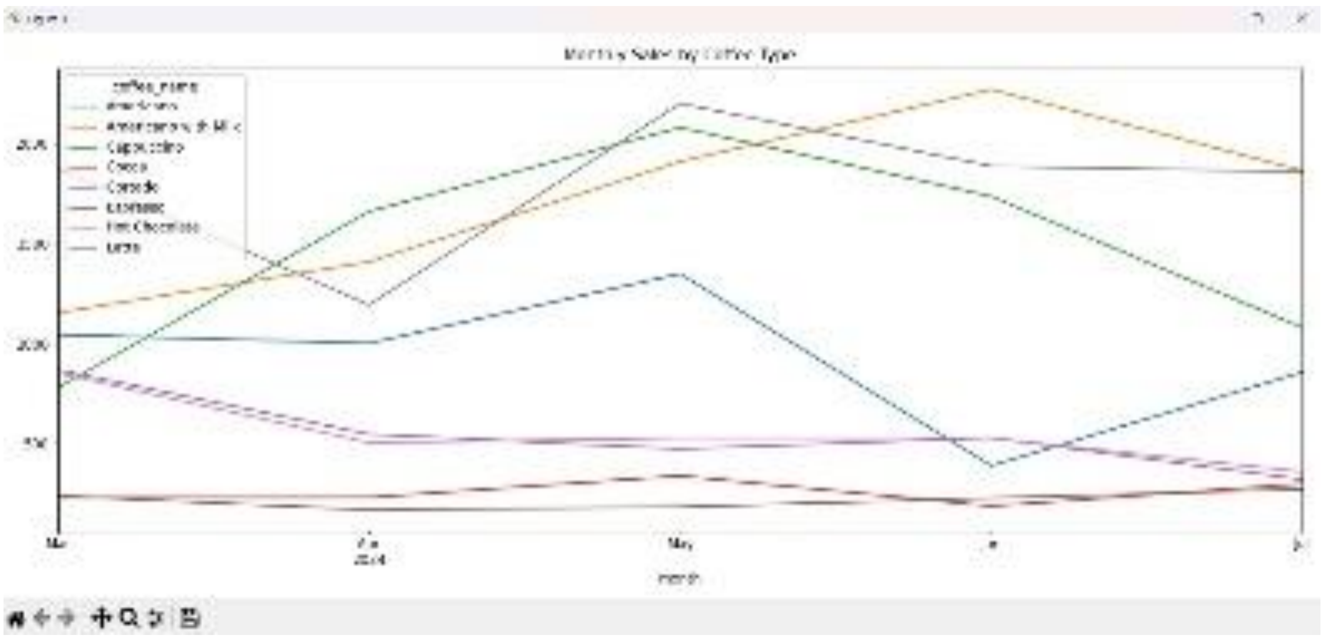


Figure 1
Monthly Sales by Coffee Type
coffee_name
Americano
Americano with Milk
Cappuccino
Cocoa
Cortado
Espresso
Hot Chocolate
Latte
Mar
Apr
2024
May
Jun
Jul
month

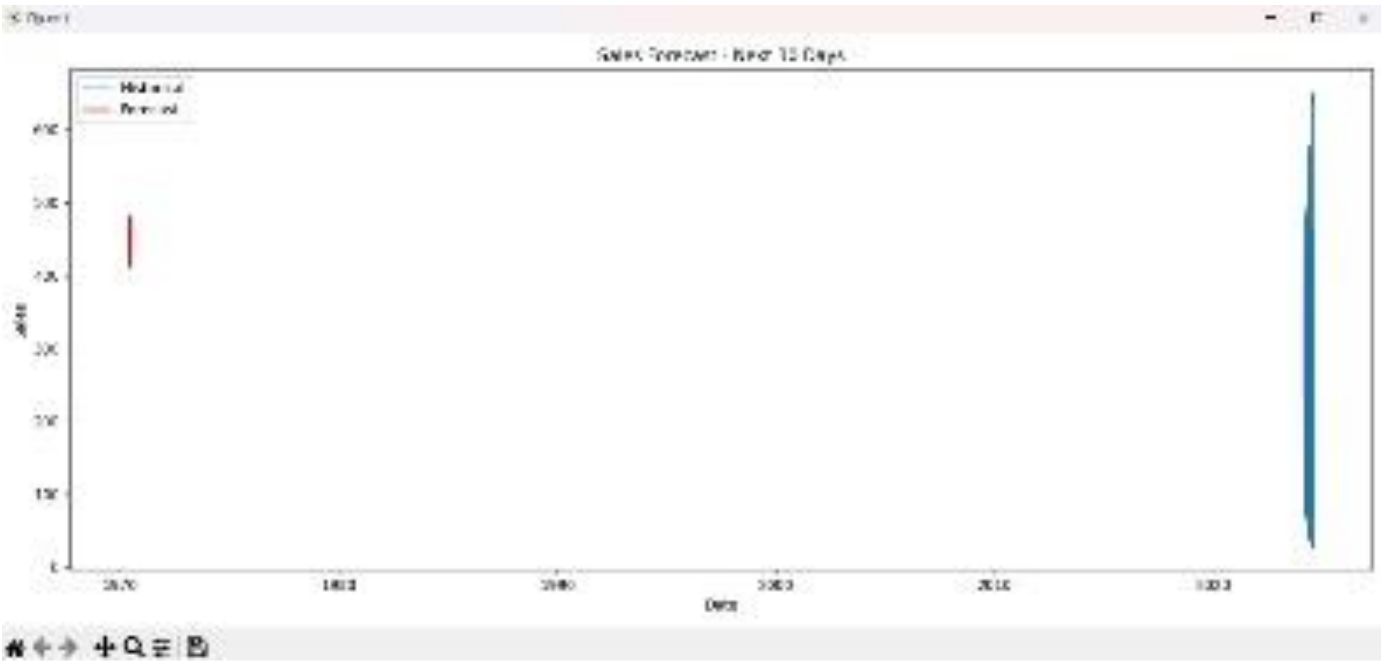


Figure 1
Sales Forecast - Next 10 Days
Historical
Forecast
Sales
1970
1980
1990
2000
2010
2020
Date

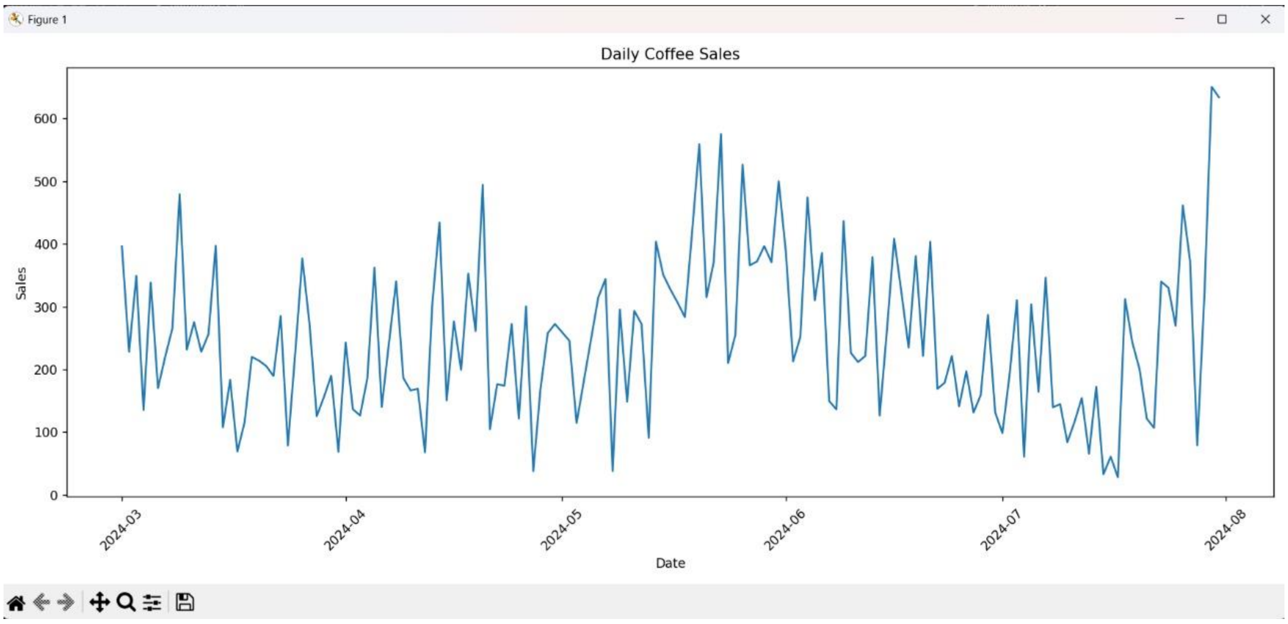


Figure 1
Daily Coffee Sales
Sales
Date
0
100
200
300
400
500
600
2024-03
2024-04
2024-05
2024-06
2024-07
2024-08