eda-laboratory

November 13, 2024

1 Exp 1: Install the Data Analysis and Visualizatio tool: R/ Python/ Tableau Public/ Power BI

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
# Sample data
data = {
'Name': ['John', 'Jane', 'Mike', 'Emily', 'Alex'],
'Age': [25, 30, 22, 28, 35],
'Score': [90, 85, 78, 95, 88]
}
```

[3]:
```
# Create a DataFrame from the data
df = pd.DataFrame(data)
```

[4]:
```
# Display the DataFrame
print("DataFrame:")
print(df)
```

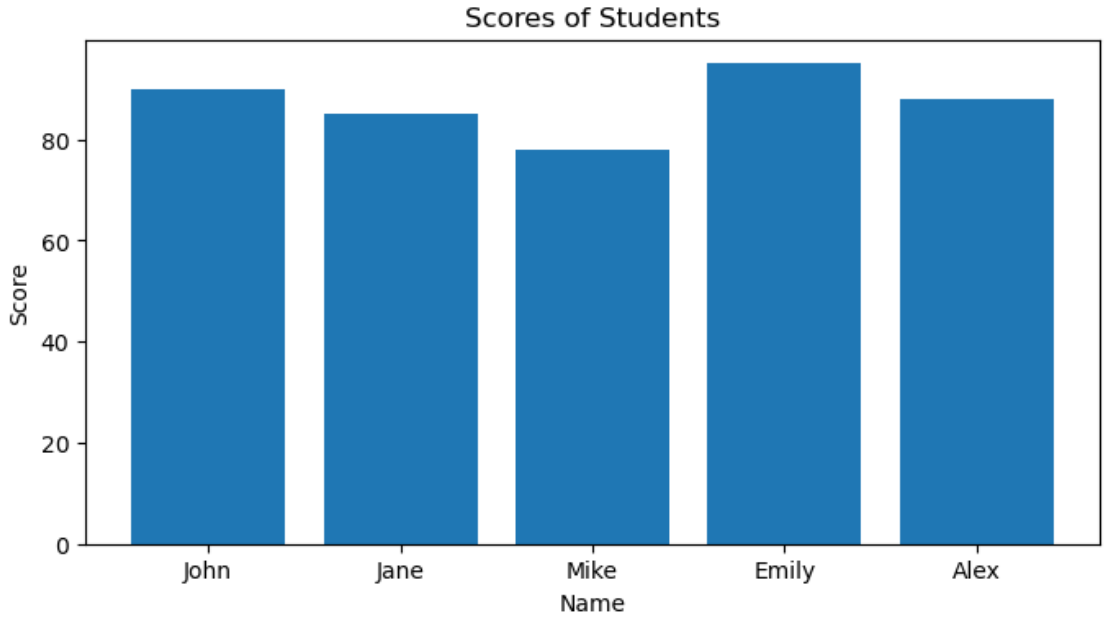
```
DataFrame:
    Name  Age  Score
0   John   25     90
1   Jane   30     85
2   Mike   22     78
3  Emily   28     95
4   Alex   35     88
```

[5]:
```
# Data Analysis
print("\nData Analysis:")
print("Mean Age:", df['Age'].mean())
print("Maximum Score:", df['Score'].max())
print("Minimum Score:", df['Score'].min())
```

```
Data Analysis:
Mean Age: 28.0
```

```
Maximum Score: 95
Minimum Score: 78
```

[6]:
```
# Data Visualization
plt.figure(figsize=(8, 4))
plt.bar(df['Name'], df['Score'])
plt.xlabel('Name')
plt.ylabel('Score')
plt.title('Scores of Students')
plt.show()
```



2 Perform EDA with Datasets like Email Dataset

[7]:
```
import pandas as pd
```

[8]:
```
data = {
'sender': ['alice@example.com', 'bob@example.com', 'alice@example.com'],
'receiver': ['bob@example.com', 'alice@example.com', 'carol@example.com'],
'subject': ['Hello', 'Meeting Reminder', 'Project Update'],
'timestamp': ['2023-08-01 10:00:00', '2023-08-02 14:30:00', '2023-08-03 09:15:
 ↪00'],
'content': ['Hi Bob,\n\nHow are you?', 'Hi Alice,\n\nDon\'t forget the meeting␣
 ↪at 3 PM.', 'HiCarol,\n\nHere\'s the latest project update.']
}
```

[9]:
```
df = pd.DataFrame(data)
df['timestamp'] = pd.to_datetime(df['timestamp'])
df.to_csv('emails.csv', index=False)
print("CSV file created successfully.")
```

```
CSV file created successfully.
```

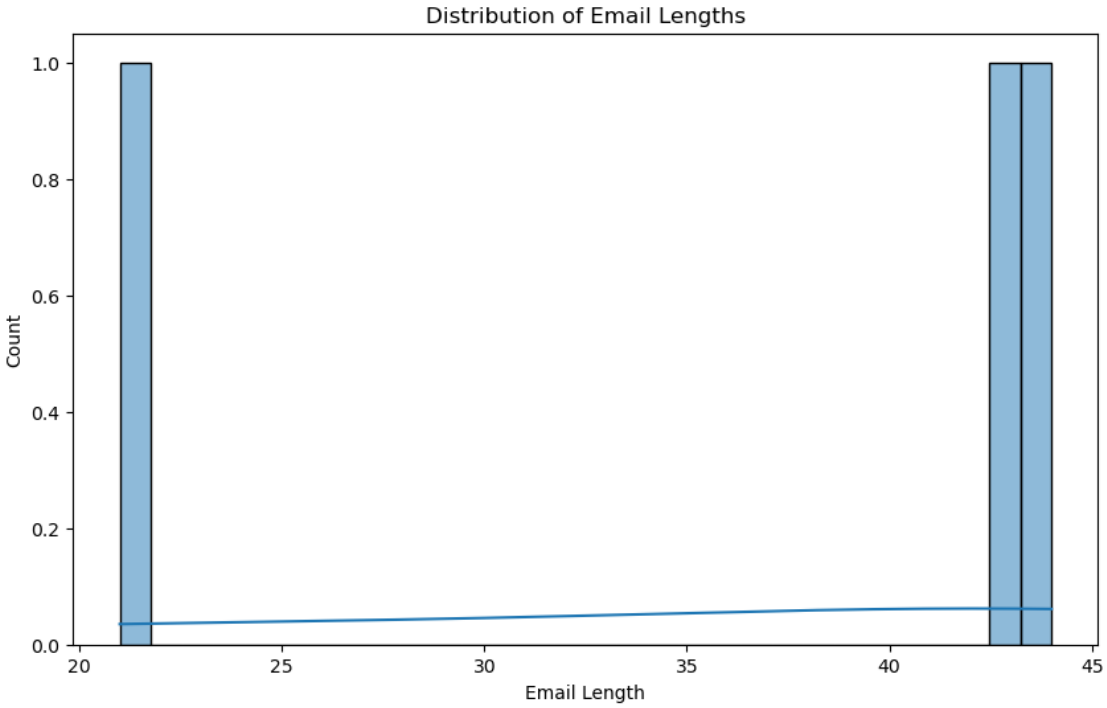
[10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\JAWAHAR A S\anaconda3\lib\site-packages\scipy\__init__.py:155:
UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for this version
of SciPy (detected version 1.26.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

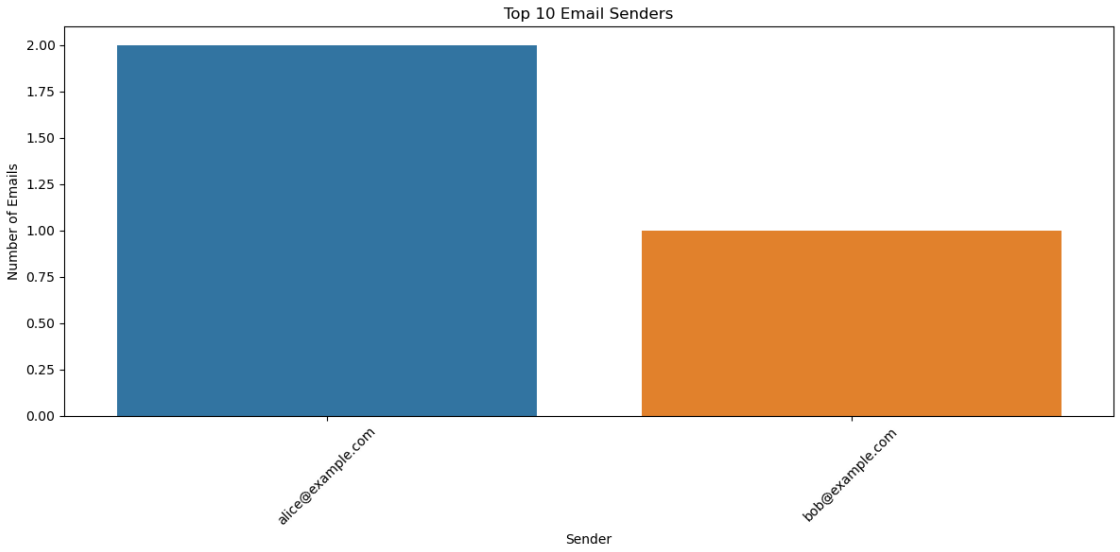
[11]:
```
df = pd.read_csv("emails.csv")
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   sender     3 non-null      object
 1   receiver   3 non-null      object
 2   subject    3 non-null      object
 3   timestamp  3 non-null      object
 4   content    3 non-null      object
dtypes: object(5)
memory usage: 248.0+ bytes
None
```

[12]:
```
df['timestamp'] = pd.to_datetime(df['timestamp'])
df.dropna(inplace=True)
df['email_length'] = df['content'].apply(len)
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='email_length', bins=30, kde=True)
plt.xlabel('Email Length')
plt.ylabel('Count')
plt.title('Distribution of Email Lengths')
plt.show()
```

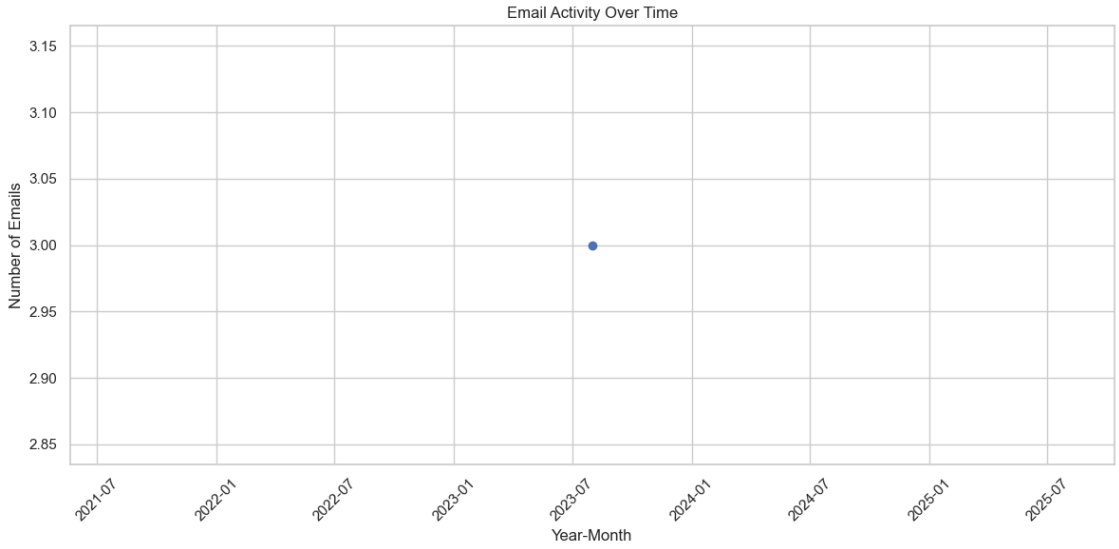


[13]:
```
top_senders = df['sender'].value_counts()[:10]
top_receivers = df['receiver'].value_counts()[:10]
plt.figure(figsize=(12, 6))
sns.barplot(x=top_senders.index, y=top_senders.values)
plt.xticks(rotation=45)
plt.xlabel('Sender')
plt.ylabel('Number of Emails')
plt.title('Top 10 Email Senders')
plt.tight_layout()
plt.show()
```



[14]:
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

df['timestamp'] = pd.to_datetime(df['timestamp'], errors='coerce')
df['year_month'] = df['timestamp'].dt.to_period('M')
email_activity = df.groupby('year_month').size()
print(email_activity)
email_activity = email_activity.reset_index()
email_activity['year_month'] = email_activity['year_month'].dt.to_timestamp()
sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))
plt.plot(email_activity['year_month'], email_activity[0], marker='o', color='b')
plt.xlabel('Year-Month')
plt.ylabel('Number of Emails')
plt.title('Email Activity Over Time')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
year_month
2023-08    3
Freq: M, dtype: int64
```



3 Exp 3: Working with Numpy Arrays, Pandas Data Frame, Basic Plots using Matplotlib

3.0.1 1) Numpy Arrays

[15]:
```
import numpy as np
arr=np.array([[1,2,3],[4,2,5]])
print("Array is of type:",type(arr))
print("No of dimensions:",arr.ndim)
print("Shape of array:",arr.shape)
print("Size of array:",arr.size)
print("Array stores elements of type:",arr.dtype)
```

```
Array is of type: <class 'numpy.ndarray'>
No of dimensions: 2
Shape of array: (2, 3)
Size of array: 6
Array stores elements of type: int32
```

[16]:
```
import numpy as np
a=np.array([[1,2,3],[3,4,5],[4,5,6]])
print(a)
print("After Slicing")
print(a[1:])
```

```
[[1 2 3]
 [3 4 5]
 [4 5 6]]
```

```
After Slicing
[[3 4 5]
 [4 5 6]]
```

[17]:
```
import numpy as np
a=np.array([[1,2,3],[3,4,5],[4,5,6]])
print('Our arrayis:')
print(a)
print('The items in the second column are:')
print(a[...,1])
print('\n')
print('The items in the second row are:')
print(a[1,...])
print('\n')
print('The items column 1 onwards are:')
print(a[...,1])
```

```
Our arrayis:
[[1 2 3]
 [3 4 5]
 [4 5 6]]
The items in the second column are:
[2 4 5]


The items in the second row are:
[3 4 5]


The items column 1 onwards are:
[2 4 5]
```

3.0.2 2) Pandas Dataframes

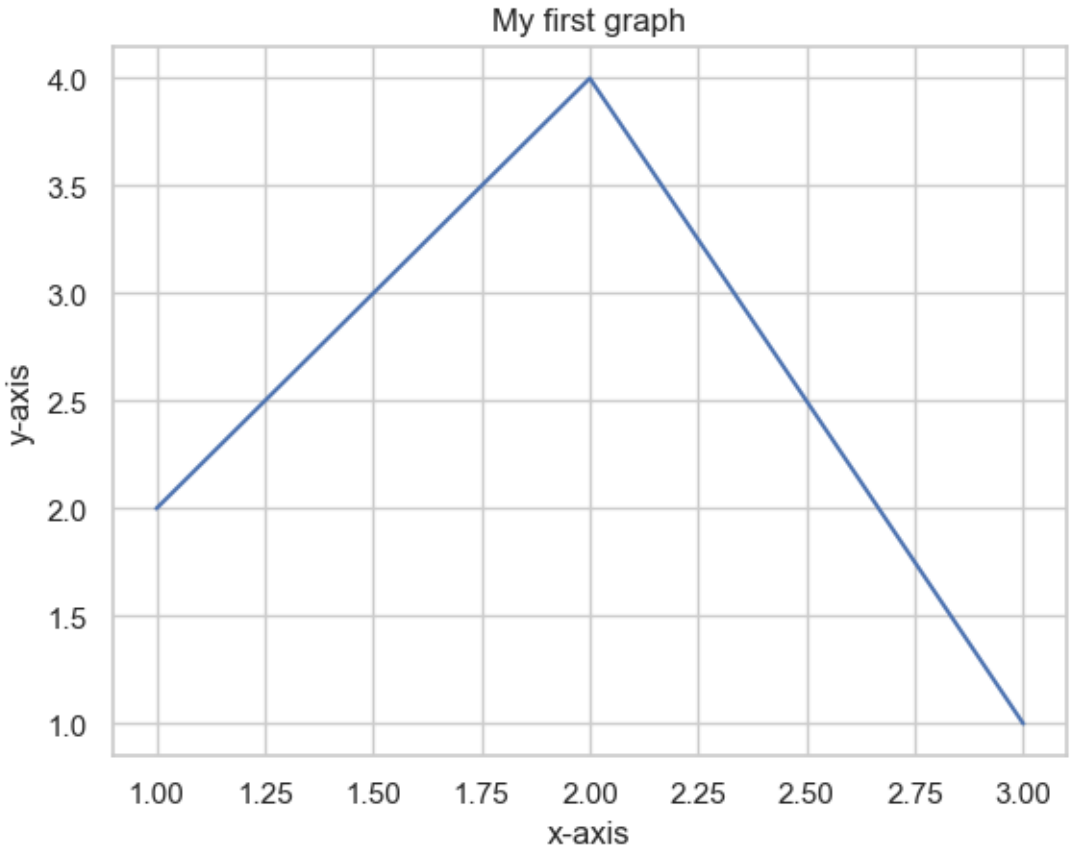
[18]:
```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'Los Angeles', 'Chicago']}

df = pd.DataFrame(data)

print(df)

print("\nAge Column:")
print(df['Age'])
```

```python
print("\nRow at index 1:")
print(df.iloc[1])

df['Country'] = ['USA', 'USA', 'USA']
print("\nDataFrame with new column 'Country':")
print(df)

print("\nBasic Statistics:")
print(df.describe())
```

```
      Name  Age         City
0    Alice   25     New York
1      Bob   30  Los Angeles
2  Charlie   35      Chicago

Age Column:
0    25
1    30
2    35
Name: Age, dtype: int64

Row at index 1:
Name            Bob
Age              30
City    Los Angeles
Name: 1, dtype: object

DataFrame with new column 'Country':
      Name  Age         City Country
0    Alice   25     New York     USA
1      Bob   30  Los Angeles     USA
2  Charlie   35      Chicago     USA

Basic Statistics:
        Age
count   3.0
mean   30.0
std     5.0
min    25.0
25%    27.5
50%    30.0
75%    32.5
max    35.0
```

3.0.3 3) Basic Plot using Matplotlib

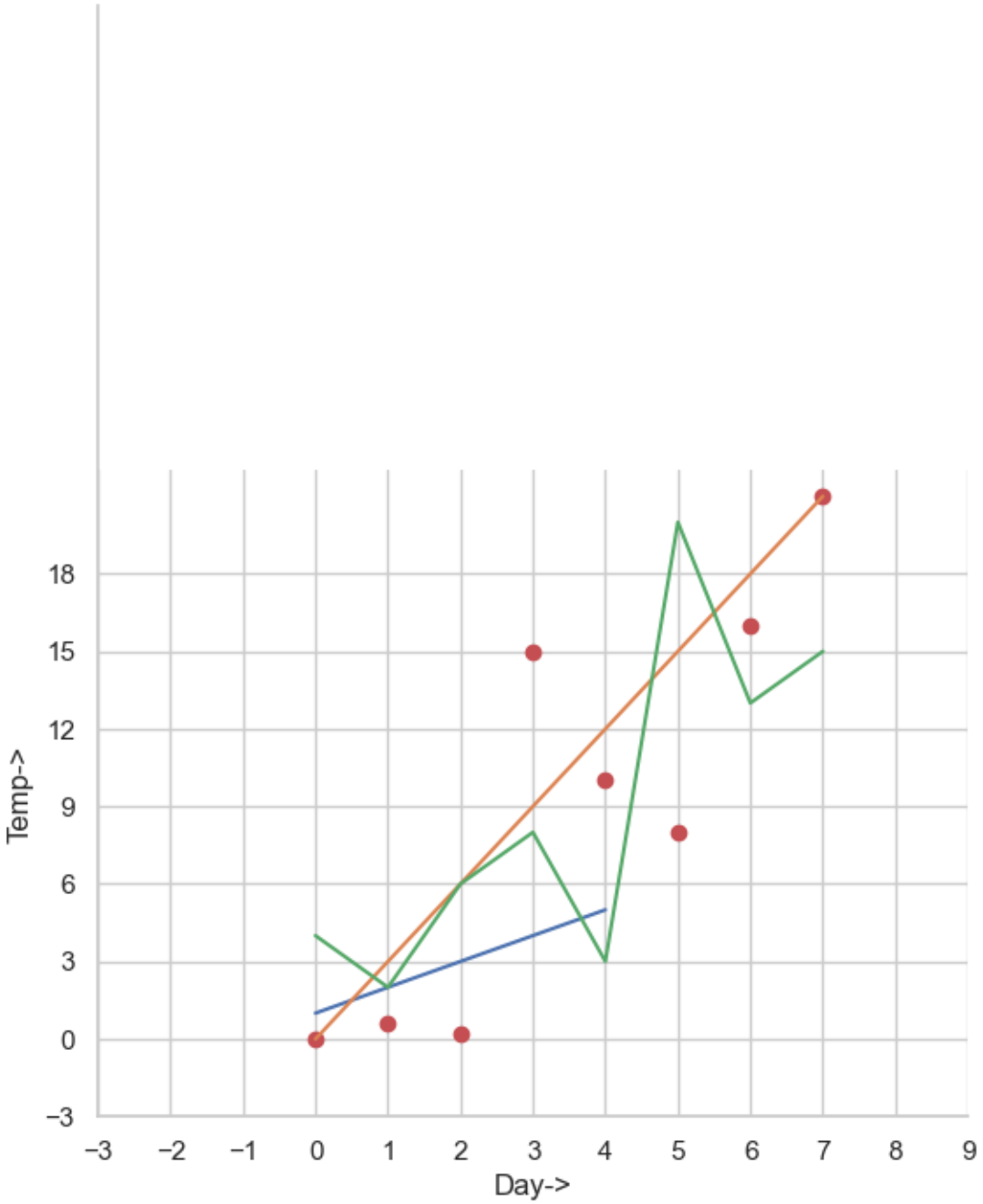
[19]:
```
import matplotlib.pyplot as plt
x=[1,2,3]
y=[2,4,1]
plt.plot(x,y)
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('My first graph')
plt.show()
```



[20]:
```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[0,0.6,0.2,15,10,8,16,21]
plt.plot(a)
plt.plot(b,"or")
plt.plot(list(range(0,22,3)))
plt.xlabel('Day->')
plt.ylabel('Temp->')
c=[4,2,6,8,3,20,13,15]
```

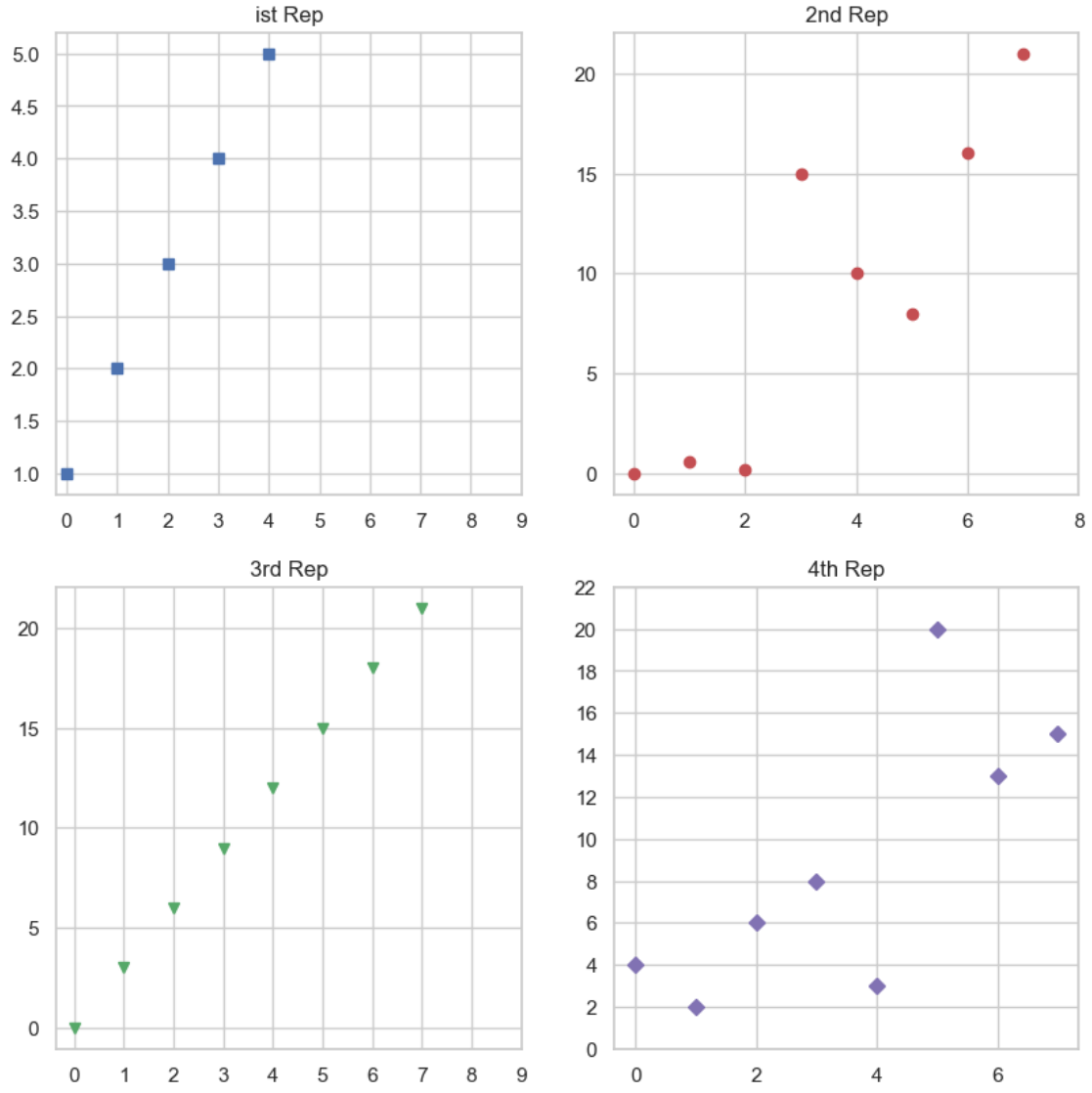
```
plt.plot(c,label='4th Rep')
ax=plt.gca()
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_bounds(-3,40)
plt.xticks(list(range(-3,10)))
plt.yticks(list(range(-3,20,3)))
```

[20]:
```
([<matplotlib.axis.YTick at 0x1c22a9f5be0>,
  <matplotlib.axis.YTick at 0x1c22a9f5460>,
  <matplotlib.axis.YTick at 0x1c22a9ef3d0>,
  <matplotlib.axis.YTick at 0x1c22a68ce20>,
  <matplotlib.axis.YTick at 0x1c22a6975b0>,
  <matplotlib.axis.YTick at 0x1c22a68c970>,
  <matplotlib.axis.YTick at 0x1c22a67d250>,
  <matplotlib.axis.YTick at 0x1c22a697340>],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



[21]:
```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[0,0.6,0.2,15,10,8,16,21]
c=[4,2,6,8,3,20,13,15]
fig=plt.figure(figsize=(10,10))
sub1=plt.subplot(2,2,1)
sub2=plt.subplot(2,2,2)
```

```
sub3=plt.subplot(2,2,3)
sub4=plt.subplot(2,2,4)
sub1.plot(a,'sb')
sub1.set_xticks(list(range(0,10,1)))
sub1.set_title('ist Rep')
sub2.plot(b,'or')
sub2.set_xticks(list(range(0,10,2)))
sub2.set_title('2nd Rep')
sub3.plot(list(range(0,22,3)),'vg')
sub3.set_xticks(list(range(0,10,1)))
sub3.set_title('3rd Rep')
sub4.plot(c,'Dm')
sub4.set_yticks(list(range(0,24,2)))
sub4.set_title('4th Rep')
plt.show()
```



4 Exp 4: Explore various variable and row filters in R for Cleaning Data

[22]:
```
import pandas as pd
import numpy as np
```

[23]:
```
data = {
'ID': range(1, 11),
'Age': np.random.randint(18, 65, size=10),
'Income': np.random.randint(30000, 90000, size=10),
'Gender': ['Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male',␣
 ↪'Female', 'Male', 'Male'],
'Education': ['High School', 'Bachelor', 'Master', 'PhD', 'Bachelor', 'Master',␣
 ↪'Bachelor','PhD','High School', 'Master']
}
```

[24]:
```
df = pd.DataFrame(data)
print(df.head())
print(df.describe())
```

```
   ID  Age  Income  Gender    Education
0   1   30   53149    Male  High School
1   2   30   84492  Female     Bachelor
2   3   60   46930    Male       Master
3   4   43   68991  Female          PhD
4   5   34   71345    Male     Bachelor
              ID        Age        Income
count  10.00000  10.000000     10.000000
mean    5.50000  37.900000  56429.700000
std     3.02765  13.253511  15976.774748
min     1.00000  20.000000  35237.000000
25%     3.25000  30.000000  46673.500000
50%     5.50000  33.000000  53941.000000
75%     7.75000  46.750000  68260.500000
max    10.00000  60.000000  84492.000000
```

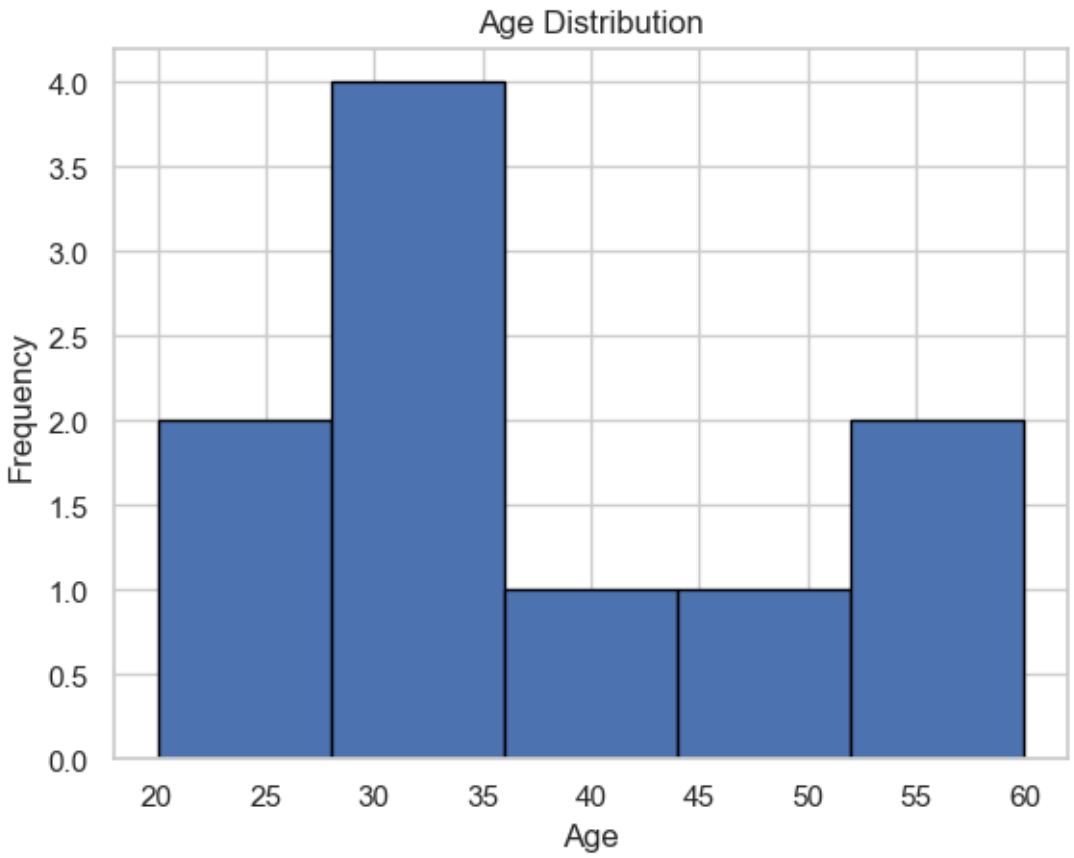
[25]:
```
print(df.isnull().sum())
print(df['Gender'].unique())
print(df['Education'].unique())
selected_columns = df[['Age', 'Income']]
print(selected_columns.head())
filtered_data = df[df['Age'] > 30]
print(filtered_data.head())
```

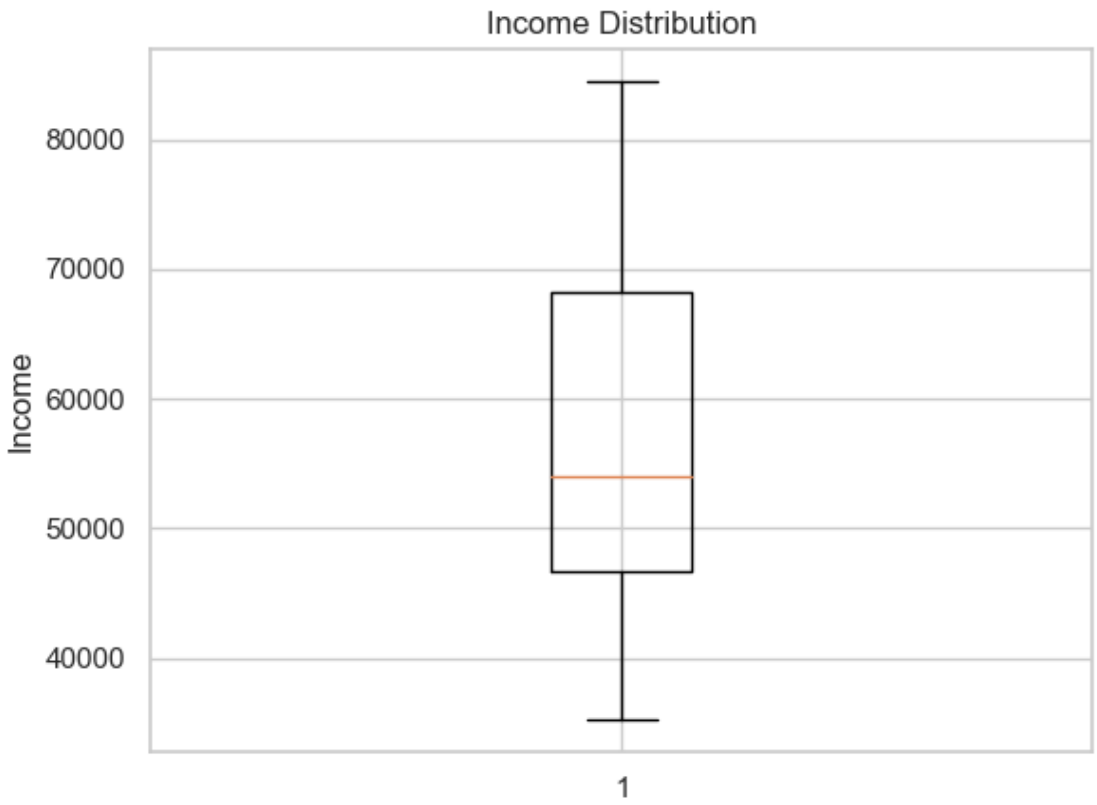
```
filtered_rows = df[(df['Gender'] == 'Male') & (df['Education'] == 'Master')]
print(filtered_rows.head())
```

```
ID           0
Age          0
Income       0
Gender       0
Education    0
dtype: int64
['Male' 'Female']
['High School' 'Bachelor' 'Master' 'PhD']
   Age  Income
0   30   53149
1   30   84492
2   60   46930
3   43   68991
4   34   71345
   ID  Age  Income  Gender Education
2   3   60   46930    Male    Master
3   4   43   68991  Female       PhD
4   5   34   71345    Male  Bachelor
6   7   48   35237    Male  Bachelor
7   8   56   36763  Female       PhD
   ID  Age  Income Gender Education
2   3   60   46930   Male    Master
9  10   20   54733   Male    Master
```

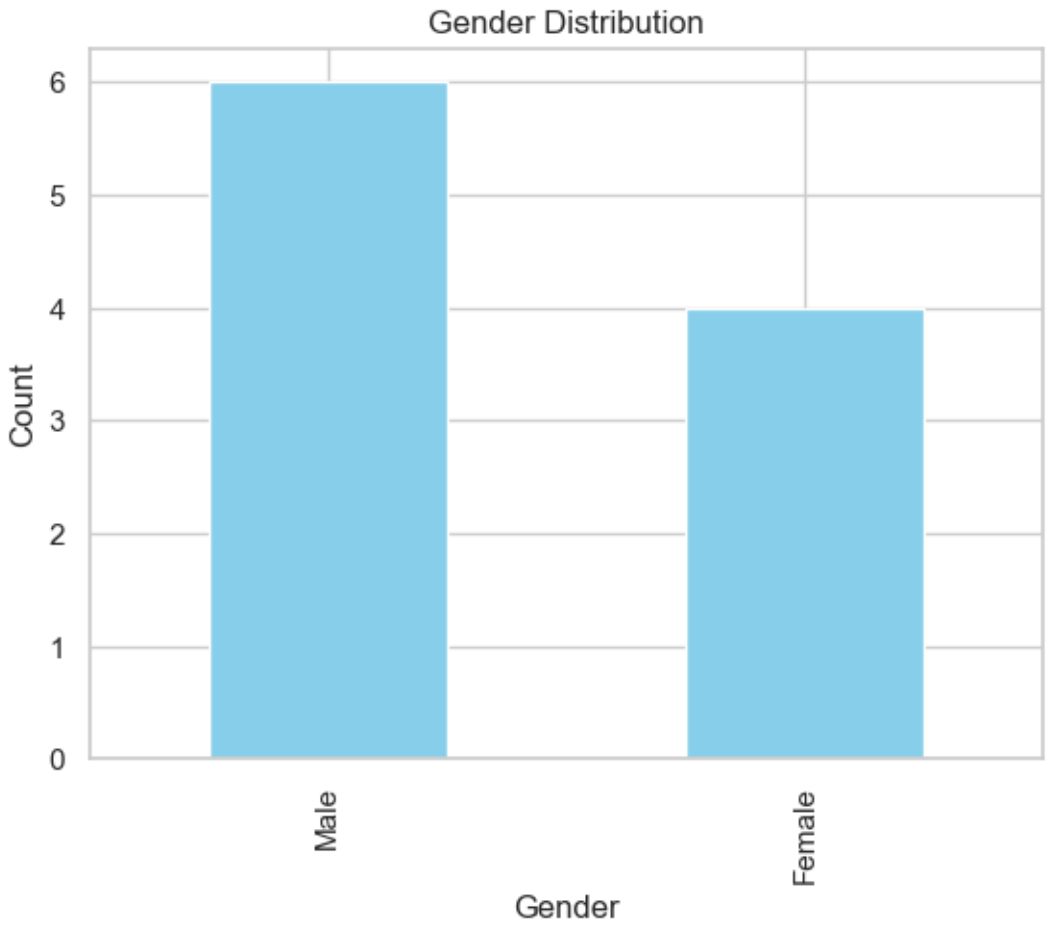
[26]:
```
import matplotlib.pyplot as plt
```

[27]:
```
plt.hist(df['Age'], bins=5, edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
plt.boxplot(df['Income'])
plt.title('Income Distribution')
plt.ylabel('Income')
plt.show()
```

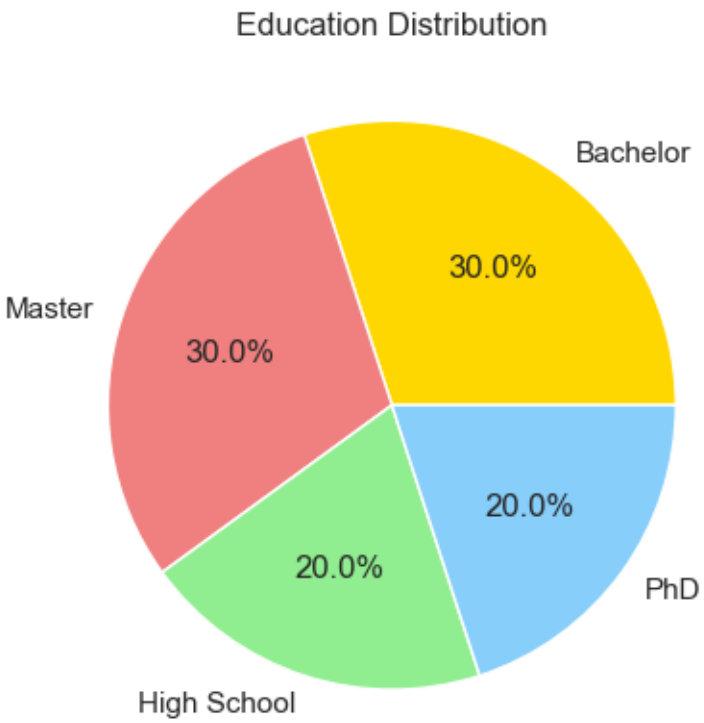

Age Distribution
Frequency
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
20
25
30
35
40
45
50
55
60
Age



[28]:
```
gender_counts = df['Gender'].value_counts()
gender_counts.plot(kind='bar', color='skyblue')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



[29]:
```
education_counts = df['Education'].value_counts()
education_counts.plot(kind='pie', autopct='%1.1f%%', colors=['gold',␣
↪'lightcoral', 'lightgreen','lightskyblue'])
plt.title('Education Distribution')
plt.ylabel('')
plt.show()
```



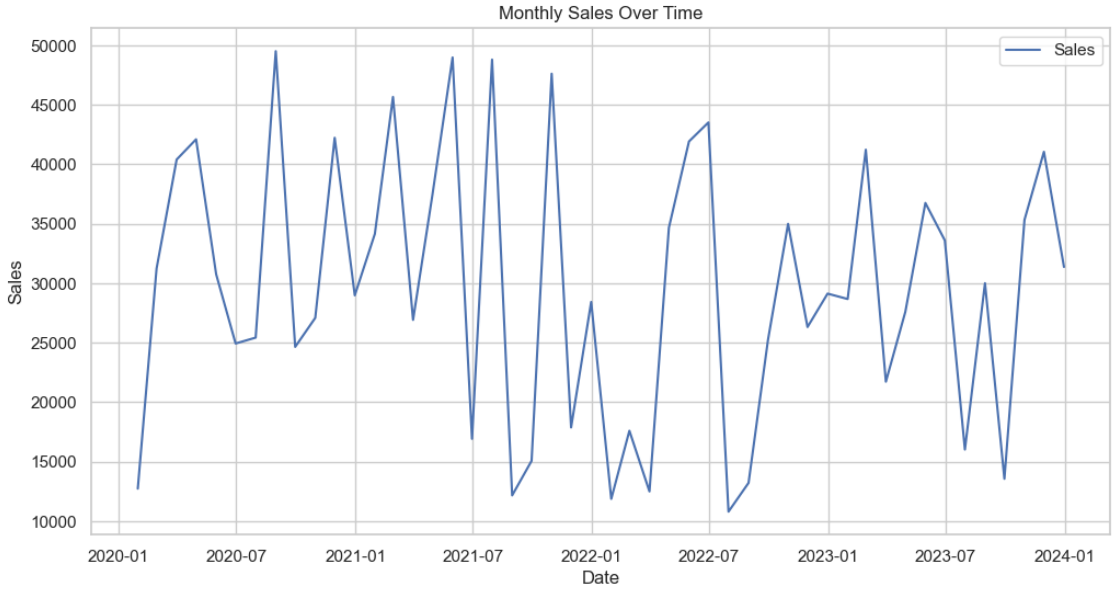
5 Exp 5: Perform Time Series Analysis and apply the various Visualization techniques

[30]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

[31]:
```
np.random.seed(0)
start_date = '2020-01-01'
end_date = '2023-12-31'
date_range = pd.date_range(start=start_date, end=end_date, freq='M')
sales_data = np.random.randint(10000, 50000, size=len(date_range))
df = pd.DataFrame({'Date': date_range, 'Sales': sales_data})
df.set_index('Date', inplace=True)
print(df.describe())

plt.figure(figsize=(12, 6))
plt.title('Monthly Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
```

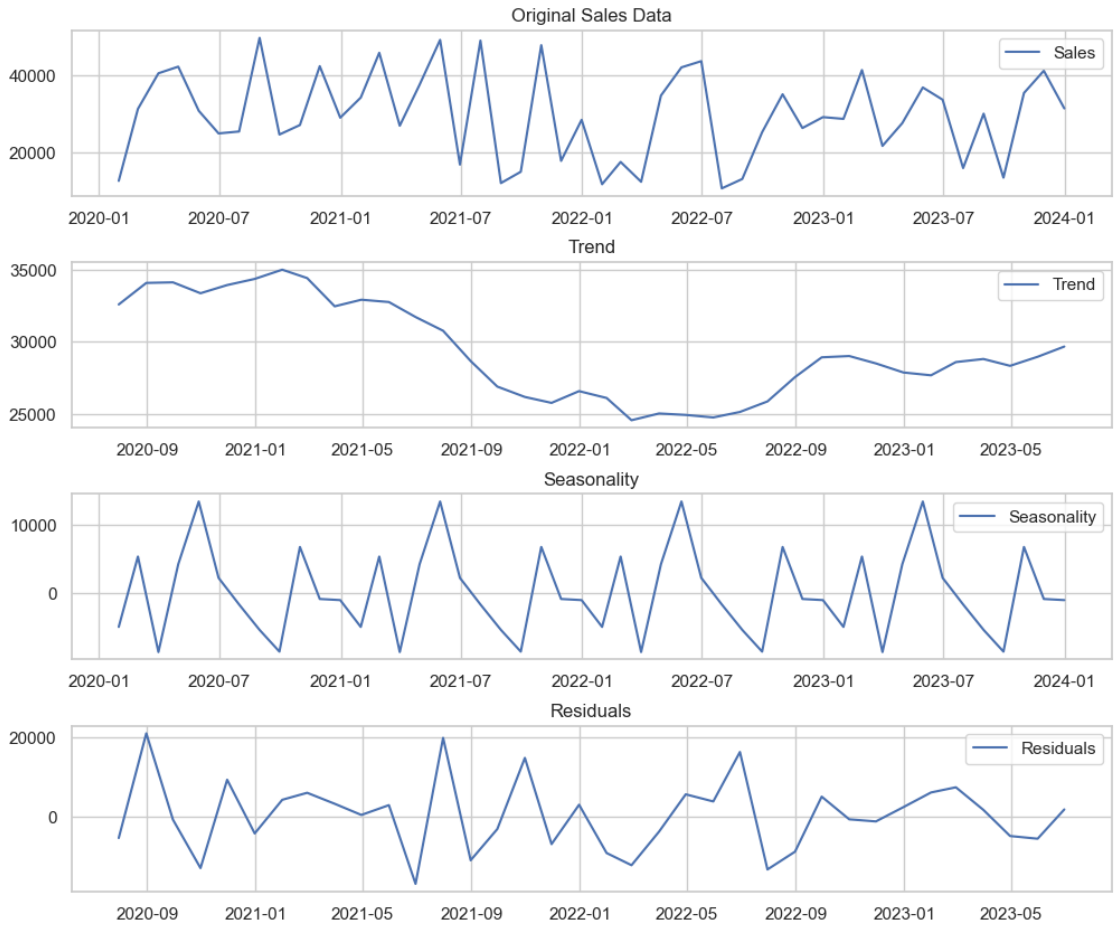
```
plt.plot(df.index, df['Sales'], label='Sales')
plt.legend()
plt.grid(True)
plt.show()
```

```
              Sales
count     48.000000
mean   29559.562500
std    11393.671539
min    10797.000000
25%    20761.500000
50%    29056.000000
75%    38202.500000
max    49512.000000
```



[32]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(df['Sales'], model='additive',period=12)
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
plt.figure(figsize=(12, 10))
plt.subplots_adjust(hspace=0.4)
plt.subplot(4, 1, 1)
plt.title('Original Sales Data')
plt.plot(df.index, df['Sales'], label='Sales')
plt.legend()
plt.subplot(4, 1, 2)
```

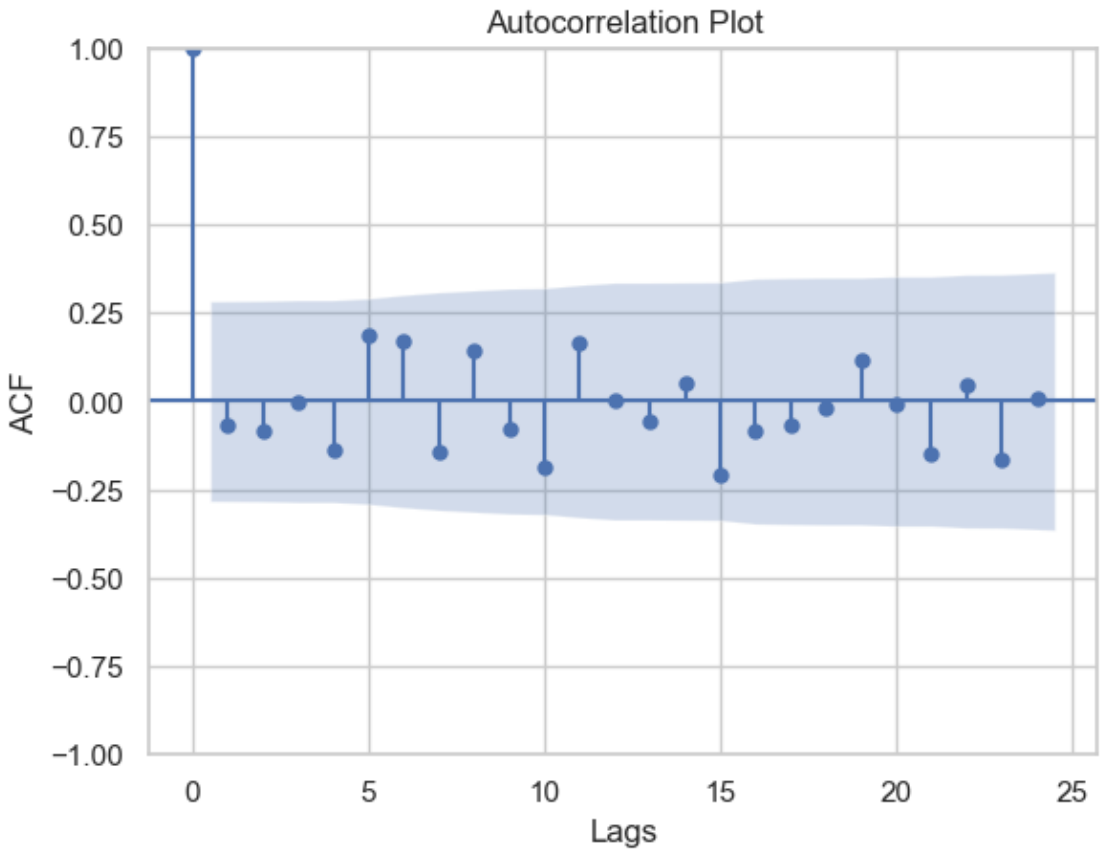
```
plt.title('Trend')
plt.plot(trend, label='Trend')
plt.legend()
plt.subplot(4, 1, 3)
plt.title('Seasonality')
plt.plot(seasonal, label='Seasonality')
plt.legend()
plt.subplot(4, 1, 4)
plt.title('Residuals')
plt.plot(residual, label='Residuals')
plt.legend()
plt.show()
```



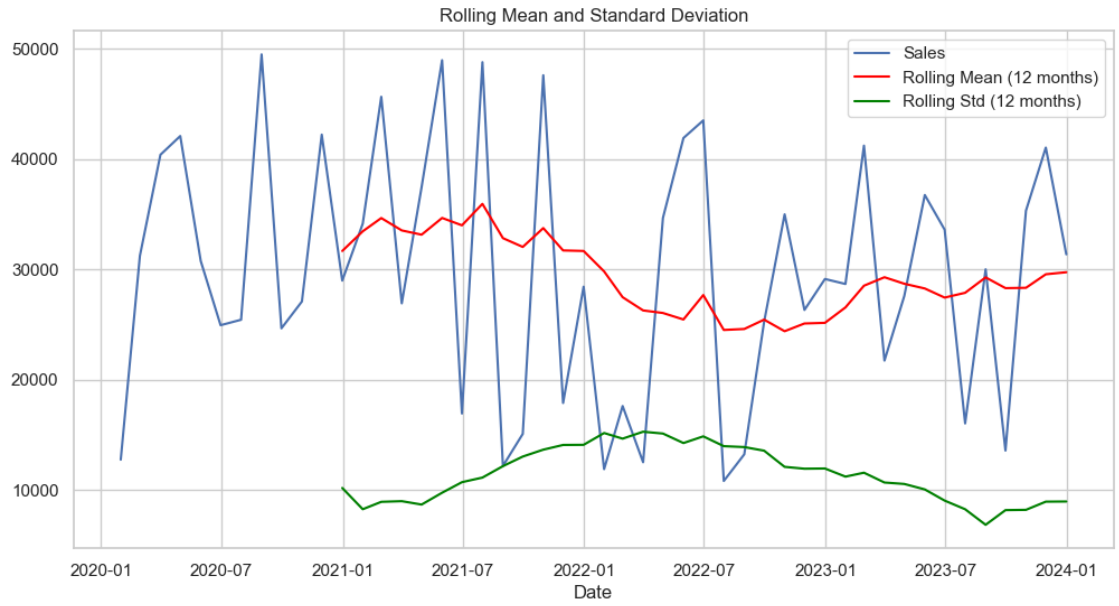
[33]:
```
from statsmodels.graphics.tsaplots import plot_acf
plt.figure(figsize=(12, 4))
plot_acf(df['Sales'], lags=24)
plt.title('Autocorrelation Plot')
plt.xlabel('Lags')
```

```
plt.ylabel('ACF')
plt.grid(True)
plt.show()
```

```
<Figure size 1200x400 with 0 Axes>
```



[34]:
```
rolling_mean = df['Sales'].rolling(window=12).mean()
rolling_std = df['Sales'].rolling(window=12).std()
plt.figure(figsize=(12, 6))
plt.title('Rolling Mean and Standard Deviation')
plt.xlabel('Date')
plt.plot(df.index, df['Sales'], label='Sales')
plt.plot(rolling_mean, label='Rolling Mean (12 months)', color='red')
plt.plot(rolling_std, label='Rolling Std (12 months)', color='green')
plt.legend()
plt.grid(True)
plt.show()
```



6 Exp 6: Perform Data Analysis and Representation on a map using various map datasets with mouse rollover effect, user interaction

[35]:
```
import pandas as pd
import numpy as np
```

[36]:
```
# Generate random data
np.random.seed(42)
num_points = 100
latitude = np.random.uniform(37.5, 38.5, num_points)
longitude = np.random.uniform(-123, -121, num_points)
value = np.random.randint(1, 100, num_points)
```

[37]:
```
# Create a DataFrame
df = pd.DataFrame({'Latitude': latitude, 'Longitude': longitude, 'Value':␣
 ↪value})
 # Save the DataFrame to a CSV file
df.to_csv('map_data.csv', index=False)
```

[38]:
```
pip install folium
```

```
Requirement already satisfied: folium in c:\users\jawahar a
s\anaconda3\lib\site-packages (0.18.0)
Requirement already satisfied: jinja2>=2.9 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from folium) (3.1.4)
```

```
Requirement already satisfied: requests in c:\users\jawahar a
s\anaconda3\lib\site-packages (from folium) (2.32.3)
Requirement already satisfied: branca>=0.6.0 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from folium) (0.8.0)
Requirement already satisfied: numpy in c:\users\jawahar a s\anaconda3\lib\site-
packages (from folium) (1.26.4)
Requirement already satisfied: xyzservices in c:\users\jawahar a
s\anaconda3\lib\site-packages (from folium) (2024.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from jinja2>=2.9->folium) (2.0.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from requests->folium) (2022.9.14)
Requirement already satisfied: idna<4,>=2.5 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from requests->folium) (2.10)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from requests->folium) (1.26.11)
Note: you may need to restart the kernel to use updated packages.
```

[39]:
```
import folium
```

[40]:
```
# Load the generated CSV file
df = pd.read_csv('map_data.csv')
```

[41]:
```
#Create a base map
base_map = folium.Map(location=[df['Latitude'].mean(), df['Longitude'].mean()],␣
 ↪zoom_start=10)
```

[42]:
```
# Add data points to the map
for index, row in df.iterrows():
    popup_text = f"Value: {row['Value']}"
    folium.Marker([row['Latitude'], row['Longitude']], popup=popup_text).
 ↪add_to(base_map)
# Save the map as an HTML file
base_map.save('interactive_map.html')
```

[43]:
```
# Import necessary libraries
import pandas as pd
import folium
# Load the data
df = pd.read_csv('map_data.csv')
#Display basic statistics
print("Basic Statistics:")
print(df.describe())
# Create a base map
```

```
base_map = folium.Map(location=[df['Latitude'].mean(), df['Longitude'].
 ↪mean()],zoom_start=10)
# Add data points to the map
for index, row in df.iterrows():
    popup_text = f"Value: {row['Value']}"
    folium.Marker([row['Latitude'], row['Longitude']], popup=popup_text).
 ↪add_to(base_map)
# Display the map
base_map
```

```
Basic Statistics:
         Latitude   Longitude       Value
count  100.000000  100.000000  100.000000
mean    37.970181 -122.004337   50.670000
std      0.297489    0.586223   28.640247
min     37.505522 -122.986096    2.000000
25%     37.693201 -122.515991   27.000000
50%     37.964142 -121.988750   47.500000
75%     38.230203 -121.467633   72.750000
max     38.486887 -121.028699   99.000000
```

[43]:
```
<folium.folium.Map at 0x1c22b7588e0>
```

7 Exp 7: Build Cartographic Visualization for Multiple Datasets involving various countries of the world, states and Districts in India

[44]:
```
pip install geopandas
```

```
Requirement already satisfied: geopandas in c:\users\jawahar a
s\anaconda3\lib\site-packages (1.0.1)
Requirement already satisfied: packaging in c:\users\jawahar a
s\anaconda3\lib\site-packages (from geopandas) (21.3)
Requirement already satisfied: pandas>=1.4.0 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from geopandas) (1.4.4)
Requirement already satisfied: numpy>=1.22 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from geopandas) (1.26.4)
Requirement already satisfied: pyogrio>=0.7.2 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from geopandas) (0.10.0)
Requirement already satisfied: shapely>=2.0.0 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from geopandas) (2.0.6)
Requirement already satisfied: pyproj>=3.3.0 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from geopandas) (3.6.1)
Requirement already satisfied: pytz>=2020.1 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\jawahar a
```

```
s\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas) (2.8.2)
Requirement already satisfied: certifi in c:\users\jawahar a
s\anaconda3\lib\site-packages (from pyogrio>=0.7.2->geopandas) (2022.9.14)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from packaging->geopandas) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\jawahar a
s\anaconda3\lib\site-packages (from python-
dateutil>=2.8.1->pandas>=1.4.0->geopandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[45]:
```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
```

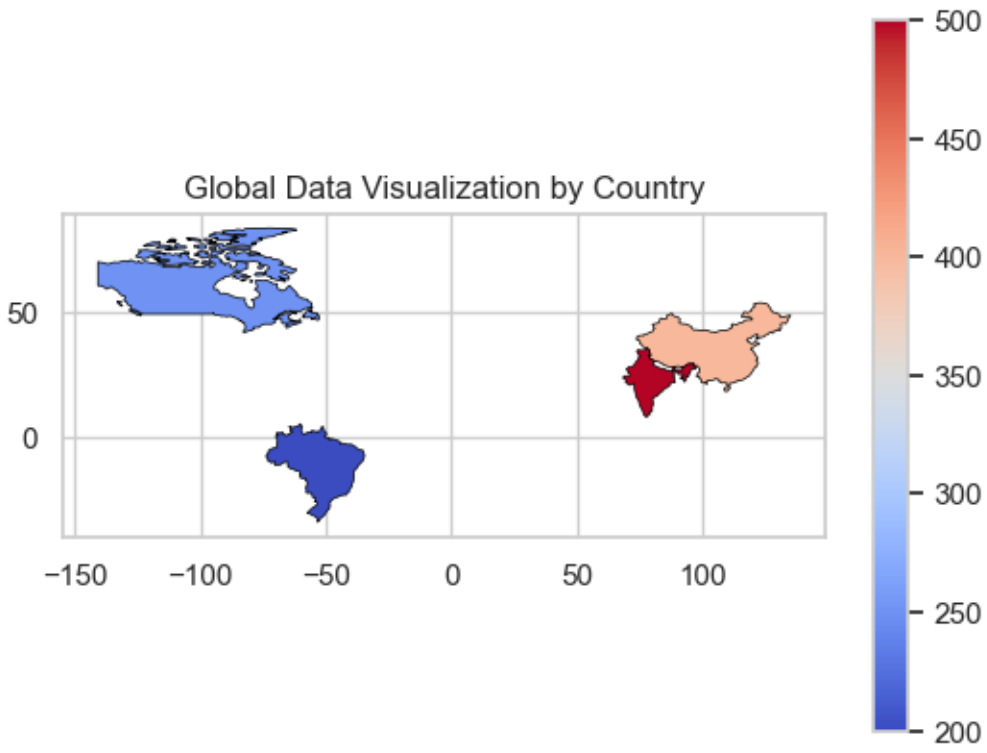
[46]:
```
world_data = pd.DataFrame({'Country': ['USA', 'Canada', 'India',␣
 ↪'Brazil','China'],
                           'Value': [100, 150, 200, 80, 120]})
india_states_data = pd.DataFrame({'State': ['Maharashtra', 'Karnataka', 'Tamil␣
 ↪Nadu', 'Uttar Pradesh','Gujarat'],
                                  'Value': [50, 75, 60, 40, 30]})
india_districts_data = pd.DataFrame({'District': ['Mumbai', 'Bengaluru',␣
 ↪'Chennai', 'Lucknow', 'Ahmedabad'],
                                     'Value': [20, 30, 25, 15, 10]})
```

[48]:
```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt

# Sample data for visualization
world_data = pd.DataFrame({
    'Country': ['United States', 'Canada', 'India', 'Brazil', 'China'],
    'Value': [300, 250, 500, 200, 400]
})

india_states_data = pd.DataFrame({
    'State': ['Maharashtra', 'Karnataka', 'Tamil Nadu', 'Uttar Pradesh',␣
 ↪'Gujarat'],
    'Value': [100, 80, 90, 50, 70]
})

india_districts_data = pd.DataFrame({
    'District': ['Mumbai', 'Bengaluru', 'Chennai', 'Lucknow', 'Ahmedabad'],
    'Value': [50, 40, 45, 30, 35]
})

# Paths to the local JSON files
```

```
world_map_path = 'countries.geo.json'           # Replace with the actual file␣
 ↪path for world map
india_states_map_path = 'india_state.geojson'  # Replace with the actual file␣
 ↪path for India states
india_districts_map_path = 'india_district.geojson'  # Replace with the actual␣
 ↪file path for India districts

# Load GeoJSON maps
world_map = gpd.read_file(world_map_path)
india_states_map = gpd.read_file(india_states_map_path)
india_districts_map = gpd.read_file(india_districts_map_path)

# Merge data with maps
world_data_geo = world_map.merge(world_data, how='left', left_on='name',␣
 ↪right_on='Country')
india_states_data_geo = india_states_map.merge(india_states_data, how='left',␣
 ↪left_on='NAME_1', right_on='State')
india_districts_data_geo = india_districts_map.merge(india_districts_data,␣
 ↪how='left', left_on='NAME_2', right_on='District')
```

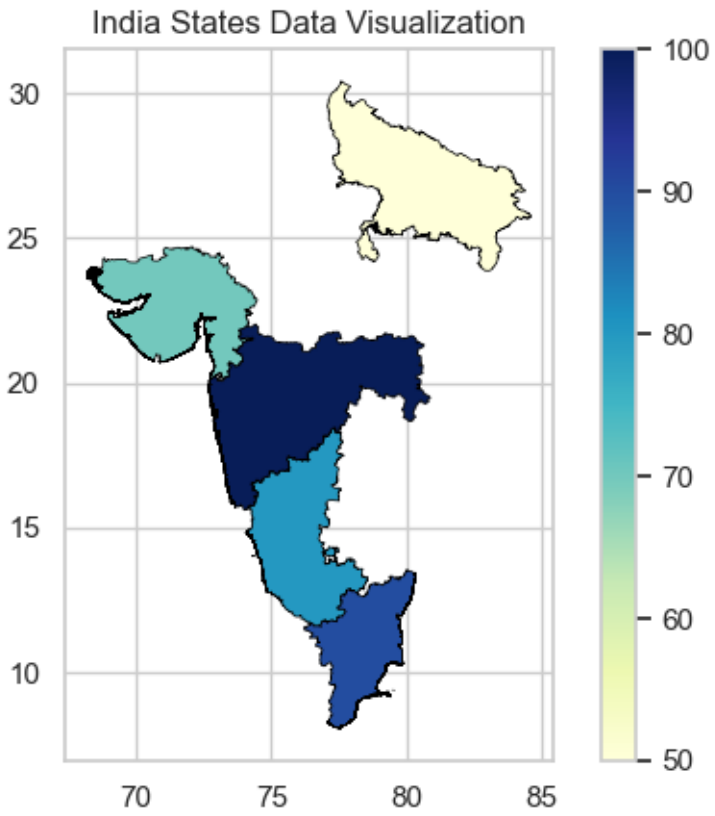
[49]:
```
# Plot World Map Visualization
plt.figure(figsize=(12, 8))
world_data_geo.plot(column='Value', cmap='coolwarm', legend=True,␣
 ↪edgecolor='black', linewidth=0.5)
plt.title('Global Data Visualization by Country')
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



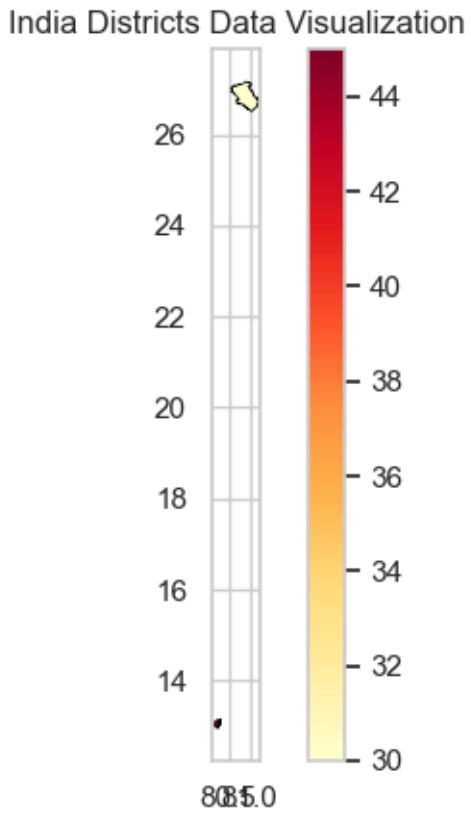
[50]:
```
# Plot India States Map Visualization
plt.figure(figsize=(10, 8))
india_states_data_geo.plot(column='Value', cmap='YlGnBu', legend=True,␣
 ↪edgecolor='black', linewidth=0.5)
plt.title('India States Data Visualization')
plt.show()
```

```
<Figure size 1000x800 with 0 Axes>
```



[51]:
```
# Plot India Districts Map Visualization
plt.figure(figsize=(10, 8))
india_districts_data_geo.plot(column='Value', cmap='YlOrRd', legend=True,␣
↪edgecolor='black', linewidth=0.5)
plt.title('India Districts Data Visualization')
plt.show()
```

```
<Figure size 1000x800 with 0 Axes>
```



8 Exp 8: Perform EDA on Wine quality dataset

[53]:
```
import pandas as pd
import numpy as np
```

[54]:
```
np.random.seed(42)
data = {
'fixed acidity': np.random.uniform(5, 15, 100),
'volatile acidity': np.random.uniform(0.1, 1.5, 100),
'citric acid': np.random.uniform(0, 1, 100),
'residual sugar': np.random.uniform(0, 10, 100),
'alcohol': np.random.uniform(8, 15, 100),
'quality': np.random.randint(1, 11, 100)
}
```

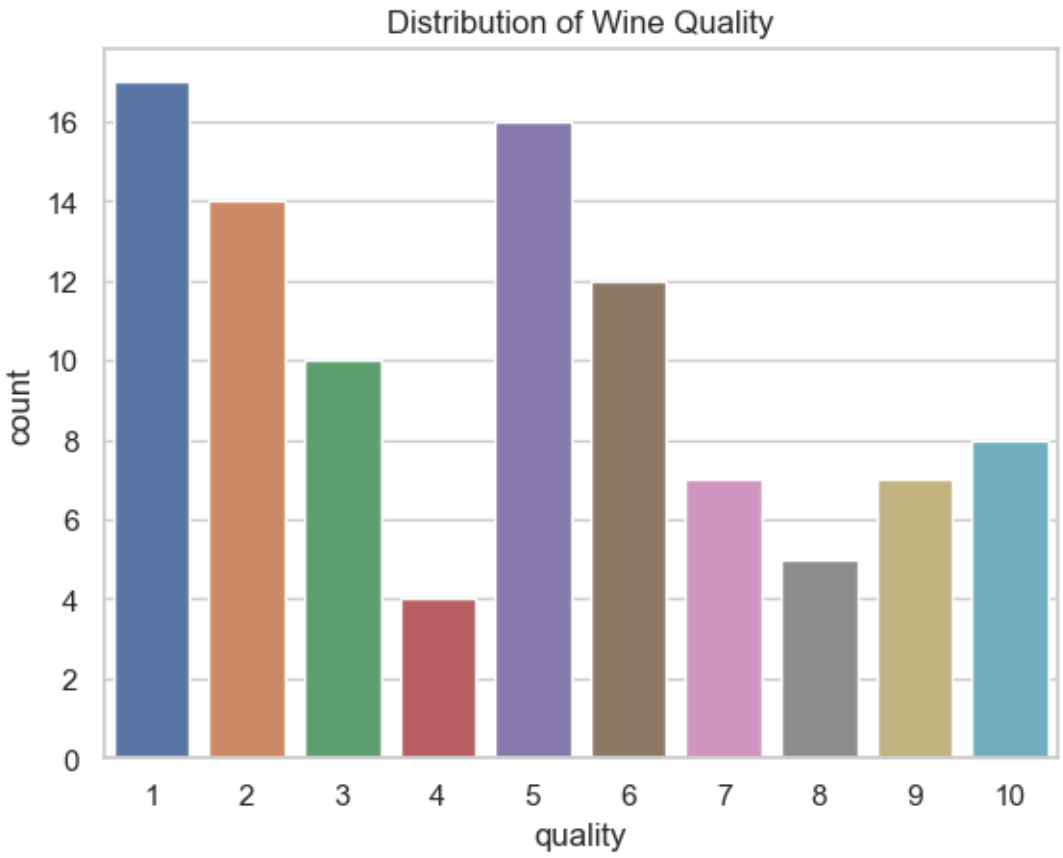
[55]:
```
wine_df = pd.DataFrame(data)
wine_df.to_csv('wine_quality_dataset.csv', index=False)
```

[56]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

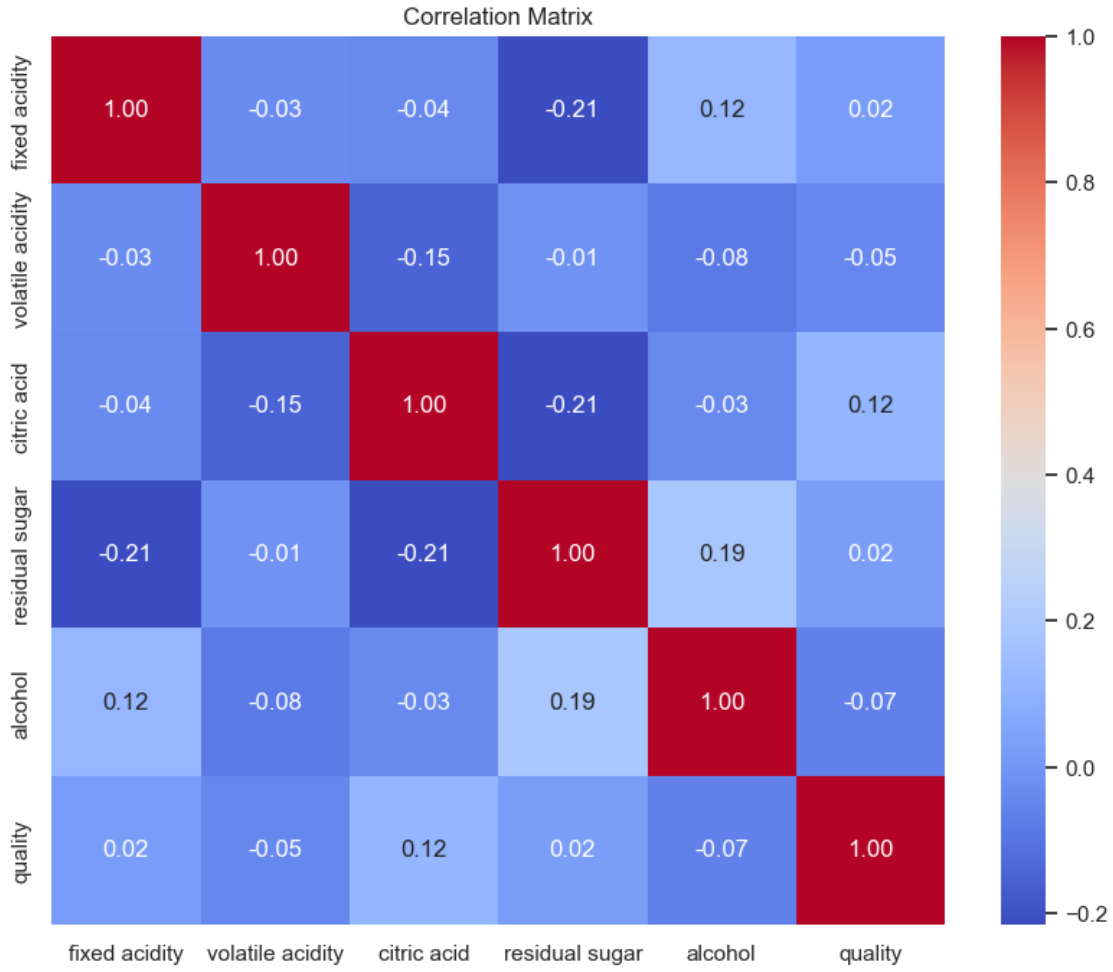
[57]:
```
# Load the dataset
wine_df = pd.read_csv('wine_quality_dataset.csv')
print(wine_df.info())
print(wine_df.describe())
print(wine_df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   fixed acidity     100 non-null    float64
 1   volatile acidity  100 non-null    float64
 2   citric acid       100 non-null    float64
 3   residual sugar    100 non-null    float64
 4   alcohol           100 non-null    float64
 5   quality           100 non-null    int64
dtypes: float64(5), int64(1)
memory usage: 4.8 KB
None
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count     100.000000        100.000000   100.000000      100.000000
mean        9.701807          0.796964     0.517601        4.911489
std         2.974894          0.410356     0.293426        2.934522
min         5.055221          0.109733     0.005062        0.143935
25%         6.932008          0.438806     0.276880        2.496149
50%         9.641425          0.807875     0.562555        5.097183
75%        12.302031          1.172657     0.752367        7.357775
max        14.868869          1.479911     0.990054        9.905051

          alcohol     quality
count  100.000000  100.000000
mean    11.612321    4.750000
std      2.230206    2.900279
min      8.075864    1.000000
25%      9.847544    2.000000
50%     11.677796    5.000000
75%     13.578189    7.000000
max     14.950754   10.000000
fixed acidity       0
volatile acidity    0
citric acid         0
residual sugar      0
alcohol             0
quality             0
dtype: int64
```

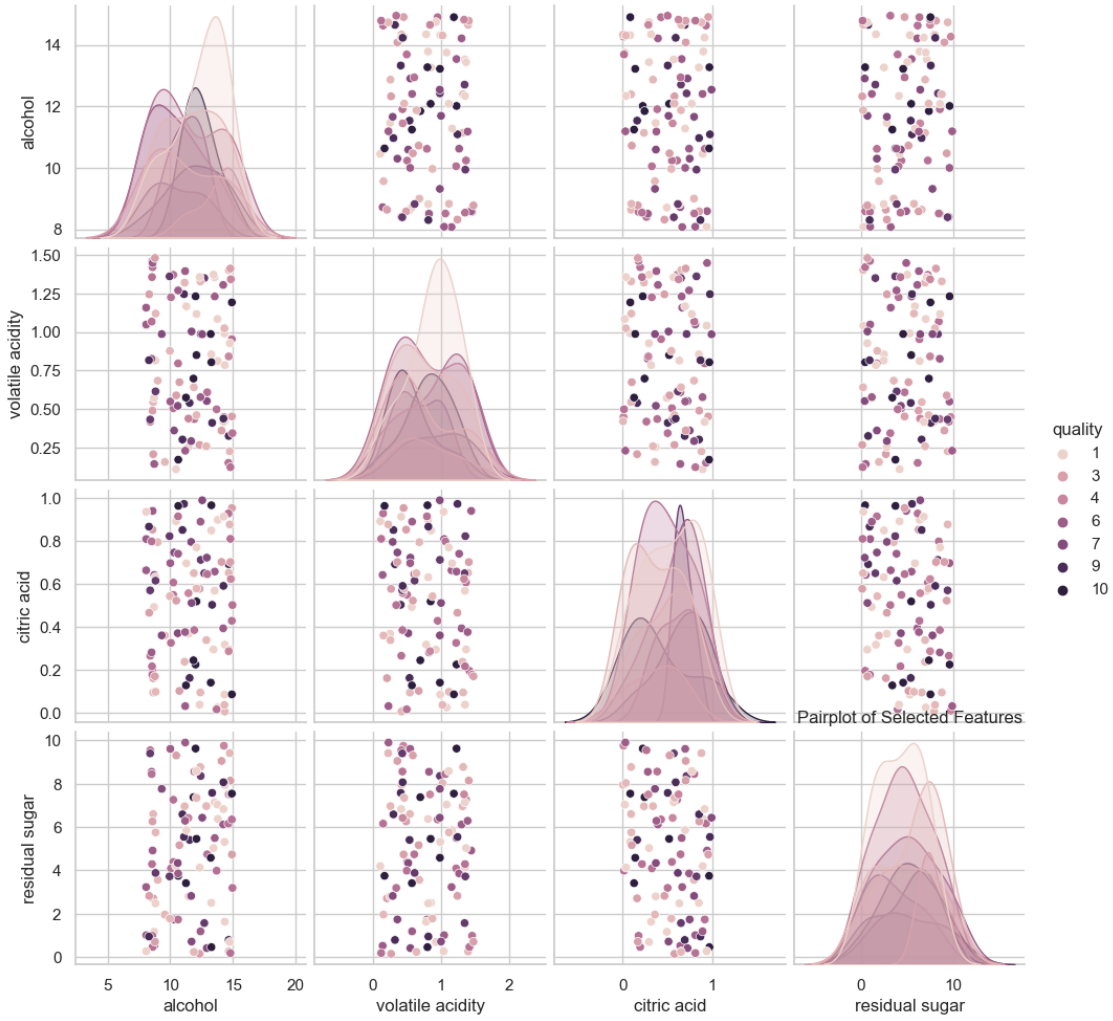
[58]:
```
sns.countplot(x='quality', data=wine_df)
plt.title('Distribution of Wine Quality')
plt.show()
```



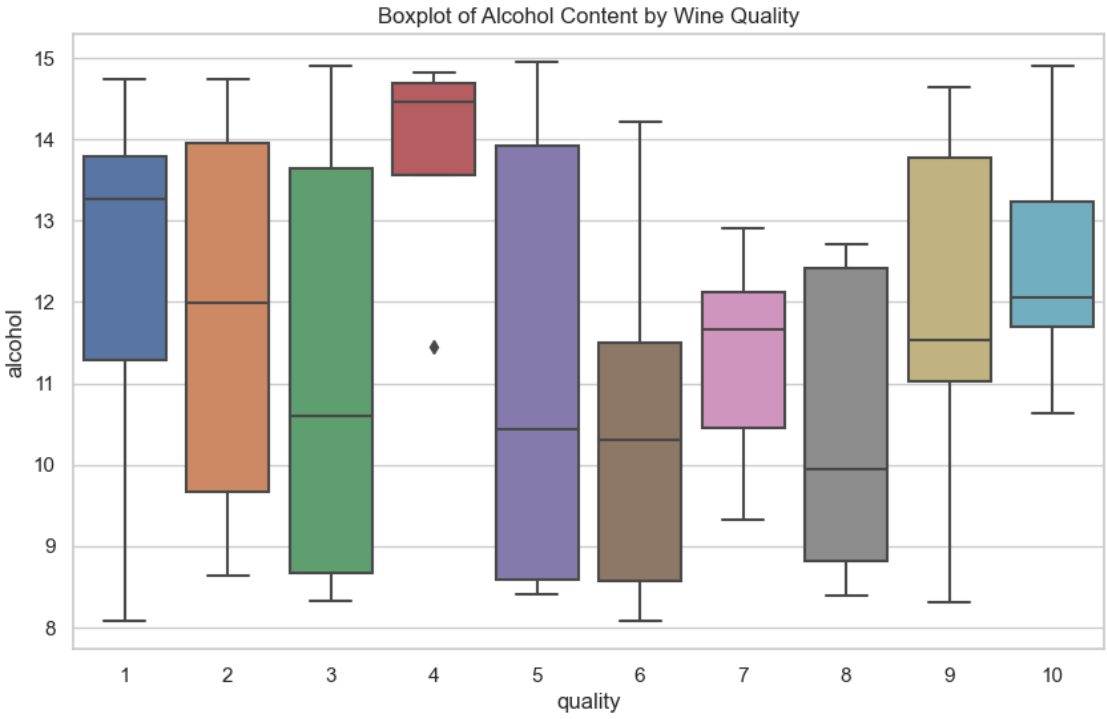
[59]:
```
correlation_matrix = wine_df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



[60]:
```
selected_features = ['alcohol', 'volatile acidity', 'citric acid', 'residual␣
↪sugar', 'quality']
sns.pairplot(wine_df[selected_features], hue='quality')
plt.title('Pairplot of Selected Features')
plt.show()
```



[61]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='quality', y='alcohol', data=wine_df)
plt.title('Boxplot of Alcohol Content by Wine Quality')
plt.show()
```



9 Exp 9: Use a Case Study on a Dataset and apply the various EDA and Visualization techniqes and present an analysis report

[62]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[63]:
```
np.random.seed(42)
employee_id = range(1, 101)
age = np.random.randint(22, 60, size=100)
gender = np.random.choice(['Male', 'Female'], size=100)
department = np.random.choice(['HR', 'Sales', 'IT', 'Marketing'], size=100)
years_of_experience = np.random.randint(1, 15, size=100)
performance_rating = np.random.randint(1, 6, size=100)
salary = np.random.randint(40000, 120000, size=100)
employee_data = pd.DataFrame({'EmployeeID': employee_id,'Age':age,'Gender':␣
 ↪gender, 'Department':department,'YearsOfExperience':␣
 ↪years_of_experience,'PerformanceRating': performance_rating, 'Salary':
 ↪salary})
employee_data.to_csv('employee_data.csv', index=False)
employee_data = pd.read_csv('employee_data.csv')
```

[64]:
```
print(employee_data.info())
print('\n')
print(employee_data.describe())
print('\n')
print(employee_data.isnull().sum())
print('\n')
department_distribution = employee_data['Department'].value_counts()
print(department_distribution)
```

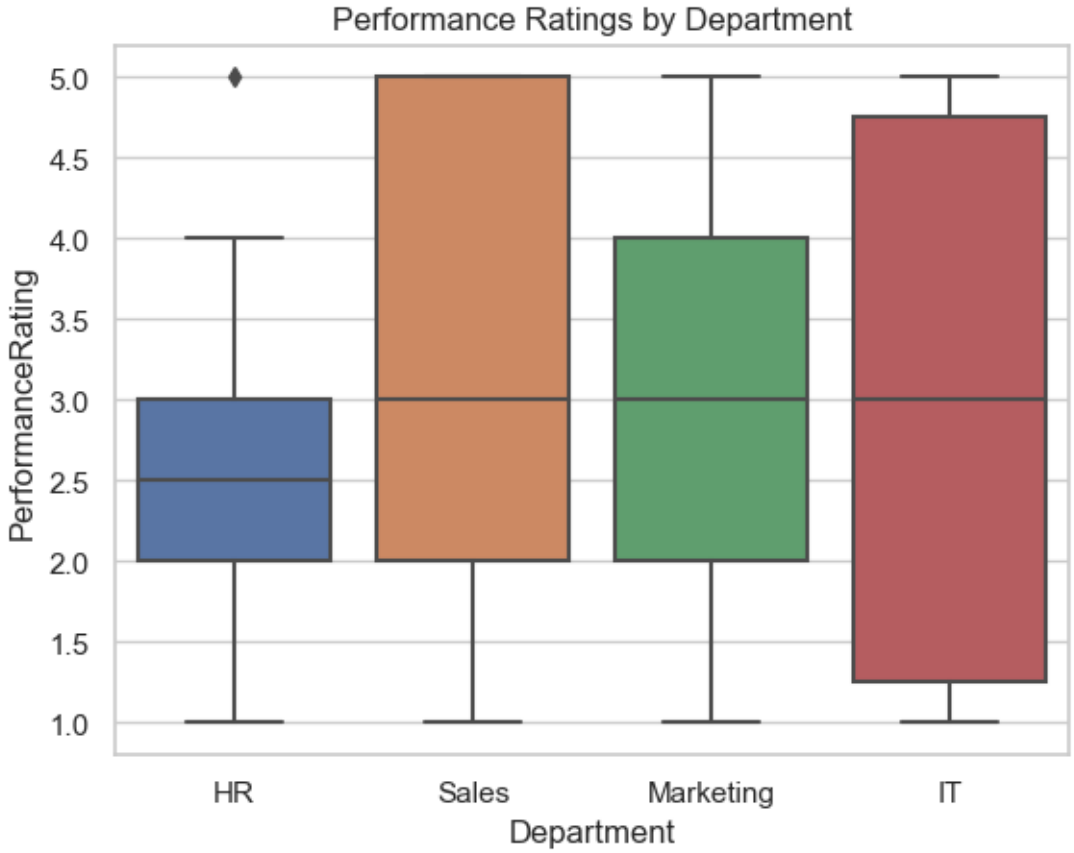
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   EmployeeID         100 non-null    int64
 1   Age                100 non-null    int64
 2   Gender             100 non-null    object
 3   Department         100 non-null    object
 4   YearsOfExperience  100 non-null    int64
 5   PerformanceRating  100 non-null    int64
 6   Salary             100 non-null    int64
dtypes: int64(5), object(2)
memory usage: 5.6+ KB
None


       EmployeeID         Age  YearsOfExperience  PerformanceRating  \
count  100.000000  100.000000         100.000000          100.00000
mean    50.500000   40.060000           6.970000            2.96000
std     29.011492   10.688255           4.381907            1.44194
min      1.000000   22.000000           1.000000            1.00000
25%     25.750000   30.000000           3.000000            2.00000
50%     50.500000   41.500000           6.000000            3.00000
75%     75.250000   48.000000          11.000000            4.00000
max    100.000000   59.000000          14.000000            5.00000

              Salary
count     100.000000
mean    81430.860000
std     22497.541898
min     40412.000000
25%     62624.750000
50%     80856.500000
75%    100741.500000
max    119605.000000
```

```
EmployeeID           0
Age                  0
Gender               0
Department           0
YearsOfExperience    0
PerformanceRating    0
Salary               0
dtype: int64
```

```
Marketing    31
IT           26
HR           22
Sales        21
Name: Department, dtype: int64
```

[65]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.histplot(employee_data['Age'], bins=20, kde=True)
plt.title('Distribution of Employee Ages')
plt.show()
```

[66]:
```
sns.boxplot(x='Department', y='PerformanceRating', data=employee_data)
plt.title('Performance Ratings by Department')
plt.show()
```



[ ]: