### **Experiment 2 : Visualization of data**

### **1. Correlation Heatmap**

A heatmap is used to visualize the correlation between numerical features in the dataset. It helps in identifying relationships between different variables.

#### **Code:**

****import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(12, 8))

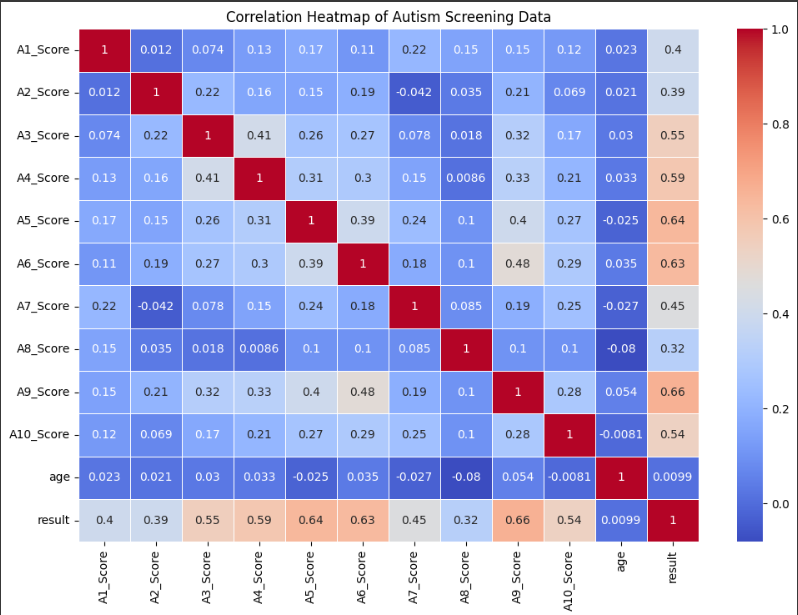
numeric\_data = data.select\_dtypes(include=['number'])  # Select only numeric columns

sns.heatmap(numeric\_data.corr(), annot=True, cmap='coolwarm', linewidths=0.5)

plt.title("Correlation Heatmap of Autism Screening Data")

plt.show()

**Explanation:**

* The function data.corr() calculates the correlation matrix.
* sns.heatmap() is used to create the heatmap.
* The annot=True argument ensures that correlation values are displayed.
* The colormap coolwarm visually distinguishes positive and negative correlations.
* 

### 

### **2. COUNT PLOT**

#### A **count plot** is used to visualize the frequency distribution of categorical variables. This helps in understanding how often each category appears in the dataset.

#### **Code:**

#### plt.figure(figsize=(8, 5))

#### sns.countplot(x='gender', hue='Class/ASD', data=data, palette='pastel')

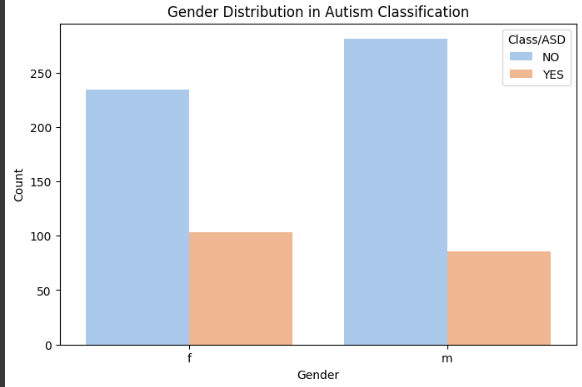
#### plt.title("Gender Distribution in Autism Classification")

#### plt.xlabel("Gender")

#### plt.ylabel("Count")

#### plt.show()

#### **Explanation:**

* plt.figure(figsize=(8, 5)) sets the figure size to ensure a clear visualization.
* sns.countplot(x='gender', hue='Class/ASD', data=data, palette='pastel')
* Creates a **count plot** where the x-axis represents **gender** categories.
* The hue='Class/ASD' argument splits each gender category into two groups: **ASD (Yes)** and **ASD (No)**.
* The palette='pastel' applies a soft color scheme for better readability.
* plt.title("Gender Distribution in Autism Classification") sets the plot title.
* plt.xlabel("Gender") and plt.ylabel("Count") label the axes appropriately.
* 

### **3.BAR Graph**

#### A **bar graph** is used to visualize and compare the frequency of categories in a dataset. It helps in understanding how different groups contribute to the dataset.

#### **Code:**

plt.figure(figsize=(12, 6))

data['contry\_of\_res'].value\_counts().head(10).plot(kind='bar', color='orange')

plt.title("Top 10 Countries in Dataset")

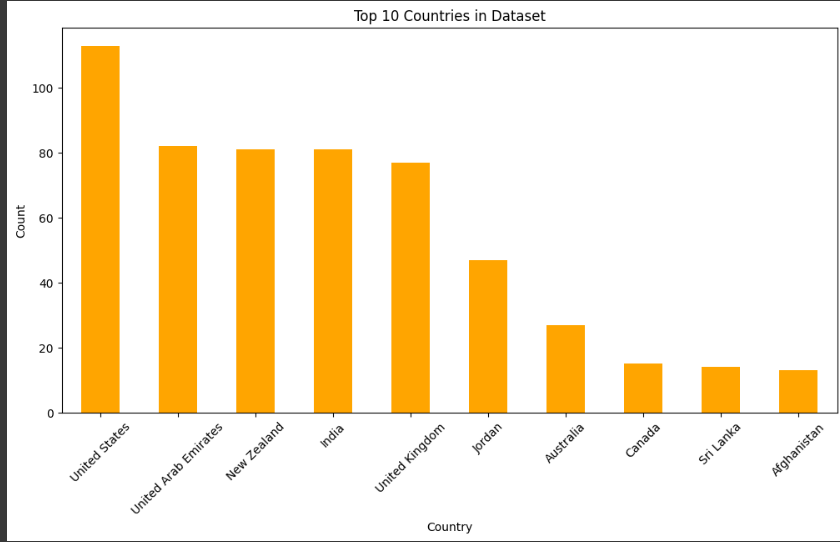
plt.xlabel("Country")

plt.ylabel("Count")

plt.xticks(rotation=45)

plt.show()

#### **Explanation:**

* plt.figure(figsize=(12, 6)) sets the figure size to make the plot large enough for better readability.
* data['contry\_of\_res'].value\_counts().head(10) calculates the **count of occurrences** for each country and selects the **top 10 countries** with the most records.
* .plot(kind='bar', color='orange') creates a **bar graph** with **orange bars** to represent the frequency of records per country.
* plt.title("Top 10 Countries in Dataset") sets the **title** of the plot.
* plt.xlabel("Country") and plt.ylabel("Count") label the **x-axis** and **y-axis** respectively.
*  plt.xticks(rotation=45) rotates the **x-axis labels** (country names) to make them easier to read, especially if the country names are long.
* 

### 

### **4 Pie Chart**

### **Distribution of Gender in Autism Screening Data** A **pie chart** is used to visualize the proportion of different categories within a dataset. It helps in understanding the relative distribution of a categorical variable.

### **Code:**

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

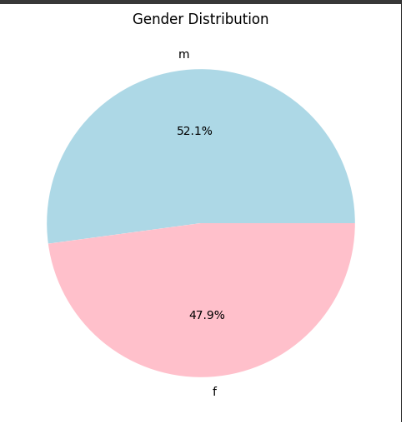
data['gender'].value\_counts().plot.pie(autopct='%1.1f%%', colors=['lightblue', 'pink'])

plt.title("Gender Distribution")

plt.ylabel("")  # Hide y-label

plt.show()

### **Explanation:**

* data['gender'].value\_counts() calculates the count of each gender category.
* .plot.pie() creates a pie chart to display these proportions.
* autopct='%1.1f%%' displays the percentage of each category inside the pie slices.
* colors=['lightblue', 'pink'] assigns colors to different gender groups for better visualization.
* plt.title("Distribution of Gender in Autism Screening Data") sets the plot title.
* plt.ylabel("") removes the y-axis label to improve aesthetics.
* 

### 

### **5) Box Plot of Microsoft Stock Closing Price**

A box plot is used to visualize the distribution of Microsoft stock's closing prices, highlighting key statistics like the median, quartiles, and outliers.

### **Code:**

plt.figure(figsize=(10, 6))

sns.boxplot(x='Class/ASD', y='result', data=data, palette='muted')

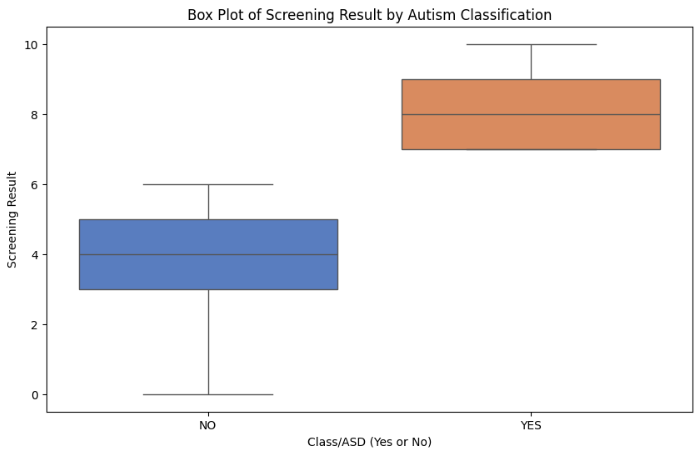
plt.title("Box Plot of Screening Result by Autism Classification")

plt.xlabel("Class/ASD (Yes or No)")

plt.ylabel("Screening Result")

plt.show()

### **Explanation:**

* plt.figure(figsize=(10, 6)) sets the figure size to ensure better visibility.
* sns.boxplot(x='Class/ASD', y='result', data=data, palette='muted')
* **Creates a box plot**, which displays the distribution of screening results for individuals classified as **ASD (Yes)** and **ASD (No)**.
* The **x-axis (Class/ASD)** represents the autism classification (Yes or No).
* The **y-axis (result)** represents the screening result score.
* The palette='muted' applies a subtle color scheme for a clear distinction between categories.
* plt.title("Box Plot of Screening Result by Autism Classification") sets the title of the plot.
* plt.xlabel("Class/ASD (Yes or No)") and plt.ylabel("Screening Result") label the axes appropriately.
  + - 

#### 

#### **Result:**

Thus the visualization techniques in Time Series Analysis and Forecasting has been studied successfully.