```
pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
import pandas as pd
import numpy as np
```

1.Create a Pandas DataFrame from the following dictionary:

data = {

'Name': ['Alice', 'Bob', 'Carol'],

'Age': [24, 27, 22],

'Salary': [50000, 55000, 48000]

}

• Add a new column called Bonus which is 10% of the salary.

```
# Given data of dictionary
data = {
    'Name': ['Alice', 'Bob', 'Carol'],
    'Age': [24, 27, 22],
    'Salary': [50000, 55000, 48000]
}
df = pd.DataFrame(data)
df['Bonus'] = df['Salary'] * 0.10
# Displaying
print(df)
```

```
    Name  Age  Salary   Bonus
0  Alice   24   50000  5000.0
1    Bob   27   55000  5500.0
2  Carol   22   48000  4800.0
```

2.Given the DataFrame created in the above question:

• Display the first two rows.

• Compute the mean salary.

```
ftr = df.head(2)
print("First two rows:")
print(ftr)
print("   ")
mean_sal = df['Salary'].mean()
print("\nMean Salary:")
print(mean_sal)
```

```
First two rows:
    Name  Age  Salary   Bonus
0  Alice   24   50000  5000.0
1    Bob   27   55000  5500.0


Mean Salary:
51000.0
```

3. Using the same DataFrame:

• Extract all rows where the Age is greater than 25.

• Filter out rows where the Salary is less than 50000.

```
data = {
    'Name': ['Alice', 'Bob', 'Carol'],
    'Age': [24, 27, 22],
    'Salary': [50000, 55000, 48000]
}
df = pd.DataFrame(data)
```

```
# Extract rows where Age is greater than 25
age = df[df['Age'] > 25]
#rows where Salary is less than 50000
sal= df[df['Salary'] < 50000]
print("Rows where Age > 25:")
print(age)
print("\nRows where Salary < 50000:")
print(sal)
```

```
Rows where Age > 25:
  Name  Age  Salary
1  Bob   27   55000

Rows where Salary < 50000:
    Name  Age  Salary
2  Carol   22   48000
```

4. Create the following DataFrame:

data = {

'A': [1, 2, None, 4],

'B': [None, 2, 3, None],

'C': [1, 2, 3, 4]

}

df = pd.DataFrame(data)

• Fill missing values in column A with the mean of the column.
• Drop rows where all values are None.

```
import pandas as pd

# Create the DataFrame
data = {
    'A': [1, 2, None, 4],
    'B': [None, 2, 3, None],
    'C': [1, 2, 3, 4]
}
df = pd.DataFrame(data)

# Fill missing values in column A with the mean of the column
df = df.assign(A=df['A'].fillna(df['A'].mean()))

# Drop rows where all values are None
df.dropna(how='all', inplace=True)

# Display the DataFrame
print(df)
```

```
          A    B  C
0  1.000000  NaN  1
1  2.000000  2.0  2
2  2.333333  3.0  3
3  4.000000  NaN  4
```

5. Create a DataFrame for employees:

data = {

'Department': ['HR', 'HR', 'IT', 'IT', 'Finance'],

'Employee': ['Alice', 'Bob', 'Carol', 'David', 'Eve'],

'Salary': [50000, 45000, 60000, 65000, 70000]

}

• Group the data by Department and compute the total and mean salary for each department.

```
data = {
    'Department': ['HR', 'HR', 'IT', 'IT', 'Finance'],
    'Employee': ['Alice', 'Bob', 'Carol', 'David', 'Eve'],
    'Salary': [50000, 45000, 60000, 65000, 70000]
}
df = pd.DataFrame(data)
df2 = df.groupby('Department')['Salary'].agg(['sum', 'mean']).reset_index
print(df2)
```

```
<bound method DataFrame.reset_index of               sum     mean
Department
Finance      70000  70000.0
HR           95000  47500.0
IT          125000  62500.0>
```

6.Given two DataFrames:

df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Carol']})

df2 = pd.DataFrame({'ID': [2, 3, 4], 'Age': [25, 30, 22]})

• Perform an inner join on the ID column.

```
df2 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Carol']})
df3 = pd.DataFrame({'ID': [2, 3, 4], 'Age': [25, 30, 22]})
# inner join on the ID column.
result = pd.merge(df2, df3, on='ID', how='inner')
# Displaying
print(result)
```

```
   ID   Name  Age
0   2    Bob   25
1   3  Carol   30
```

7. Given a small dataset, clean and display basic statistics using Pandas

```
data = {
    'Name': ['Anuram', 'Varma', 'Sai', 'Kiran', 'Avinash'],
    'Age': [20, 19, None, 21, 19],
    'Salary': [70000, 55000, 48000, None, 58000],
    'Department': ['SE', 'IT', 'Lead', 'IT', 'Dev']
}
# Create DataFrame
df = pd.DataFrame(data)

# Fill missing values in 'Age' and 'Salary' with the mean of the respective columns
df = df.assign(Age=df['Age'].fillna(df['Age'].mean()), Salary=df['Salary'].fillna(df['Salary'].mean()))

# Display basic statistics
basic_stats = df.describe()
# Display the cleaned DataFrame and basic statistics
print("Cleaned DataFrame:")
print(df)
print("\nBasic Statistics:")
print(basic_stats)
```

```
Cleaned DataFrame:
      Name    Age   Salary Department
0   Anuram  20.00  70000.0         SE
1    Varma  19.00  55000.0         IT
2      Sai  19.75  48000.0       Lead
3    Kiran  21.00  57750.0         IT
4  Avinash  19.00  58000.0        Dev

Basic Statistics:
             Age        Salary
count   5.000000      5.000000
mean   19.750000  57750.000000
std     0.829156   7949.056548
min    19.000000  48000.000000
25%    19.000000  55000.000000
50%    19.750000  57750.000000
75%    20.000000  58000.000000
max    21.000000  70000.000000
```

8. small EDA task

• Load the Titanic dataset (titanic.csv) into a Pandas DataFrame.

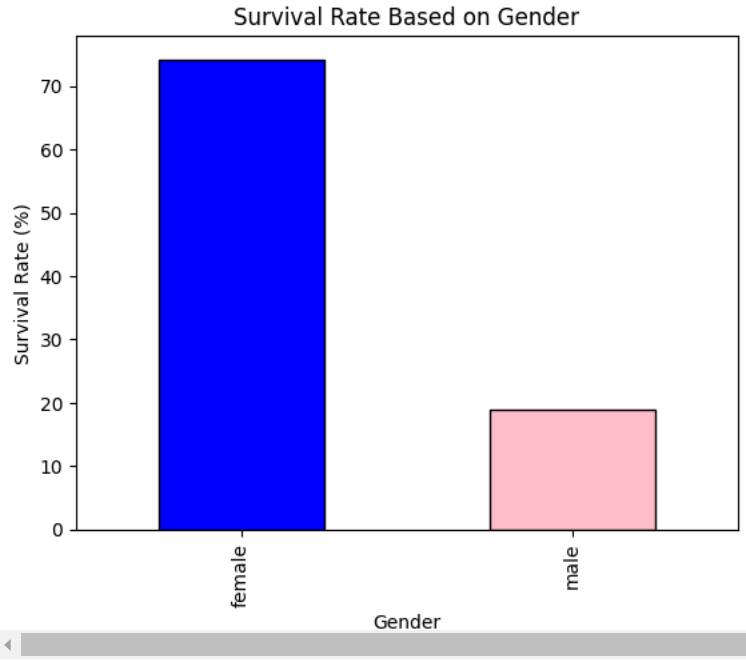
• Perform the following tasks:

- Display the number of missing values in each column.
- Find the average age of passengers.
- Create a bar plot to show the survival rate based on gender.

```
import matplotlib.pyplot as plt
```

```
# Loading the Titanic dataset
df = pd.read_csv('/content/Titanic-Dataset.csv')

# Displaying the number of missing values in each column
missing_values = df.isnull().sum()
print("Missing values in each column:")
print(missing_values)

# Finding the average age of passengers
average_age = df['Age'].mean()
print("\nAverage age of passengers:", average_age)

# Creating a bar plot to show the survival rate based on gender
survival_rate = df.groupby('Sex')['Survived'].mean() * 100
survival_rate.plot(kind='bar', color=['blue', 'pink'], edgecolor='black')

# Customize the plot
plt.xlabel('Gender')
plt.ylabel('Survival Rate (%)')
plt.title('Survival Rate Based on Gender')
plt.show()
```

```
Missing values in each column:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64

Average age of passengers: 29.69911764705882
```



9. Data Transformation

• Load a dataset containing daily temperatures and dates. Perform the following:

- Convert the date column into a DateTime object.
- Create a new column that categorizes temperatures into High, Medium, or Low.

```
import pandas as pd
data = {
    'Date': ['2024-12-01', '2024-12-02', '2024-12-03', '2024-12-04', '2024-12-05'],
    'Temperature': [22, 25, 19, 30, 18]
}
# Created DataFrame
df = pd.DataFrame(data)

# Converting the date column into a DateTime object
df['Date'] = pd.to_datetime(df['Date'])
```

```
df['Date'] = pd.to_datetime(df['Date'])
# Created a new column that categorizes temperatures into High, Medium, or Low
def categorize_temperature(temp):
    if temp >= 25:
        return 'High'
    elif temp >= 20:
        return 'Medium'
    else:
        return 'Low'

df['Temperature_Category'] = df['Temperature'].apply(categorize_temperature)
# Display the transformed DataFrame
print(df)
```

```
        Date  Temperature Temperature_Category
0 2024-12-01           22               Medium
1 2024-12-02           25                 High
2 2024-12-03           19                  Low
3 2024-12-04           30                 High
4 2024-12-05           18                  Low
```