Practical No-3

Date of Conduction : **Date of Checking:**

Descriptive Statistics - Measures of Central Tendency and variability

Perform the following operations on any open source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.

Python Code:

```
# Import necessary libraries
import pandas as pd

# Load the Titanic dataset (you can replace this with your dataset)
titanic_df =
pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/mas
ter/titanic.csv')

# Display first few rows of the dataset
print("First few rows of the Titanic dataset:")
print(titanic_df.head())

# 1. Provide summary statistics grouped by a categorical variable
# Let's use the 'Pclass' (passenger class) as the categorical variable and
'Age' as the quantitative variable
grouped_stats = titanic_df.groupby('Pclass')['Age'].describe()

# Display the summary statistics
print("\nSummary statistics of Age grouped by Pclass:")
print(grouped_stats)

# 2. Create a list that contains a numeric value for each response to the
categorical variable
# In this case, create a list of mean ages for each passenger class
mean_age_by_class = titanic_df.groupby('Pclass')['Age'].mean().tolist()

# Display the list of mean ages for each passenger class
print("\nMean Age for each Passenger Class:")
print(mean_age_by_class)
```

Explanation:

- The code loads the Titanic dataset using pd.read_csv.
- It then displays the first few rows of the dataset to provide an overview.

- The dataset is grouped by the 'Pclass' (passenger class) variable, and summary statistics for the 'Age' variable within each group are calculated using the describe() function.
- The resulting grouped summary statistics are displayed.
- Additionally, a list containing the mean age for each passenger class is created.

OUTPUT:

```
"C:\Users\Ram Kumar Solanki\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:\Users\Ram Kumar Solanki\PycharmProjects\MBA_BFS\main.py"
```

First few rows of the Titanic dataset:

```
   PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0            1         0       3  ...   7.2500   NaN         S
1            2         1       1  ...  71.2833   C85         C
2            3         1       3  ...   7.9250   NaN         S
3            4         1       1  ...  53.1000  C123         S
4            5         0       3  ...   8.0500   NaN         S

[5 rows x 12 columns]
```

Summary statistics of Age grouped by Pclass:

```
        count       mean        std   min   25%   50%   75%   max
Pclass
1       186.0  38.233441  14.802856  0.92  27.0  37.0  49.0  80.0
2       173.0  29.877630  14.001077  0.67  23.0  29.0  36.0  70.0
3       355.0  25.140620  12.495398  0.42  18.0  24.0  32.0  74.0
```

Mean Age for each Passenger Class:

```
[38.233440860215055, 29.87763005780347, 25.14061971830986]

Process finished with exit code 0
```

2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset. Provide the codes with outputs and explain everything that you do in this step.

Python Code:

```
# Import necessary libraries
import seaborn as sns
import pandas as pd

# Load the Iris dataset
iris = sns.load_dataset('iris')

# Display the first few rows of the dataset
print("First few rows of the Iris dataset:")
print(iris.head())

# 1. Display basic statistical details for 'Iris-setosa'
setosa_stats = iris[iris['species'] == 'setosa'].describe()

# Display the statistical details for 'Iris-setosa'
print("\nStatistical details for 'Iris-setosa':")
print(setosa_stats)

# 2. Display basic statistical details for 'Iris-versicolor'
versicolor_stats = iris[iris['species'] == 'versicolor'].describe()

# Display the statistical details for 'Iris-versicolor'
print("\nStatistical details for 'Iris-versicolor':")
print(versicolor_stats)

# 3. Display basic statistical details for 'Iris-virginica'
virginica_stats = iris[iris['species'] == 'virginica'].describe()

# Display the statistical details for 'Iris-virginica'
print("\nStatistical details for 'Iris-virginica':")
print(virginica_stats)
```

Explanation:

- The code loads the Iris dataset using sns.load_dataset('iris') from the Seaborn library.
- The first few rows of the dataset are displayed to provide an overview.
- The dataset is then filtered for each species ('Iris-setosa', 'Iris-versicolor', 'Iris-virginica') separately.
- The describe() function is used to calculate basic statistical details for each species, including percentiles, mean, standard deviation, etc.
- The statistical details for each species are displayed.

OUTPUT:

```
"C:\Users\Ram    Kumar    Solanki\PycharmProjects\pythonProject\venv\Scripts\python.exe"
"C:\Users\Ram Kumar Solanki\PycharmProjects\MBA_BFS\main.py"
```

First few rows of the Iris dataset:

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

Statistical details for 'Iris-setosa':

```
       sepal_length  sepal_width  petal_length  petal_width
count      50.00000    50.000000     50.000000    50.000000
mean        5.00600     3.428000      1.462000     0.246000
std         0.35249     0.379064      0.173664     0.105386
min         4.30000     2.300000      1.000000     0.100000
25%         4.80000     3.200000      1.400000     0.200000
50%         5.00000     3.400000      1.500000     0.200000
75%         5.20000     3.675000      1.575000     0.300000
max         5.80000     4.400000      1.900000     0.600000
```

Statistical details for 'Iris-versicolor':

```
       sepal_length  sepal_width  petal_length  petal_width
count     50.000000    50.000000     50.000000    50.000000
mean       5.936000     2.770000      4.260000     1.326000
std        0.516171     0.313798      0.469911     0.197753
min        4.900000     2.000000      3.000000     1.000000
25%        5.600000     2.525000      4.000000     1.200000
```

```
50%       5.900000    2.800000     4.350000    1.300000
75%       6.300000    3.000000     4.600000    1.500000
max       7.000000    3.400000     5.100000    1.800000
```

Statistical details for 'Iris-virginica':

```
       sepal_length  sepal_width  petal_length  petal_width
count      50.00000    50.000000     50.000000     50.00000
mean        6.58800     2.974000      5.552000      2.02600
std         0.63588     0.322497      0.551895      0.27465
min         4.90000     2.200000      4.500000      1.40000
25%         6.22500     2.800000      5.100000      1.80000
50%         6.50000     3.000000      5.550000      2.00000
75%         6.90000     3.175000      5.875000      2.30000
max         7.90000     3.800000      6.900000      2.50000
```

```
Process finished with exit code 0
```