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.
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 'Irisvirginica' of iris.csv dataset. Provide the codes with outputs and explain everything that you do in this step

```
import pandas as pd
```

```
url = r"C:\Users\Rutuja Habib\Downloads\age_income.csv"
df = pd.read_csv(url)
```

```
df
```

| | age | income |
|---|---|---|
| **0** | 25 | 49000 |
| **1** | 56 | 156000 |
| **2** | 65 | 99000 |
| **3** | 32 | 192000 |
| **4** | 41 | 39000 |
| **5** | 59 | 57000 |

```
summary_stats = df.groupby('age')['income'].describe()
```

Calculating mean, median, minimum, maximum and standard deviation of the income

```
mean_income = df.groupby('age')['income'].mean()
median_income = df.groupby('age')['income'].median()
min_income = df.groupby('age')['income'].min()
max_income = df.groupby('age')['income'].max()
std_income = df.groupby('age')['income'].std()
```

Printing all the calculated data

```
print("Mean Income by Age Group:")
print(mean_income)
print("\nMedian Income by Age Group:")
print(median_income)
print("\nMinimum Income by Age Group:")
print(min_income)
print("\nMaximum Income by Age Group:")
print(max_income)
print("\nStandard Deviation of Income by Age Group:")
print(std_income)
```

```
Mean Income by Age Group:
age
25     49000.0
32    192000.0
41     39000.0
56    156000.0
59     57000.0
65     99000.0
Name: income, dtype: float64

Median Income by Age Group:
age
25     49000.0
32    192000.0
41     39000.0
56    156000.0
59     57000.0
65     99000.0
Name: income, dtype: float64

Minimum Income by Age Group:
```

```
age
25     49000
32    192000
41     39000
56    156000
59     57000
65     99000
Name: income, dtype: int64

Maximum Income by Age Group:
age
25     49000
32    192000
41     39000
56    156000
59     57000
65     99000
Name: income, dtype: int64

Standard Deviation of Income by Age Group:
age
25   NaN
32   NaN
41   NaN
56   NaN
59   NaN
65   NaN
Name: income, dtype: float64
```

Listing down the data

```
income_by_age = df.groupby('age')['income'].apply(list)
print(income_by_age)
```

```
age
25     [49000]
32    [192000]
41     [39000]
56    [156000]
59     [57000]
65     [99000]
Name: income, dtype: object
```

Doing the same for iris.csv

```
URL= r"C:\Users\Rutuja Habib\Downloads\Iris.csv"
df=pd.read_csv(URL) #reading the file
df
```

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| **0** | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| **...** | ... | ... | ... | ... | ... | ... |
| **145** | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| **146** | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| **147** | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| **148** | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| **149** | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

150 rows × 6 columns

Adding Index to the Dataframe

```
df=df.reset_index()
df
```

| | index | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|---|
| **0** | 0 | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 1 | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 2 | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 3 | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 4 | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **145** | 145 | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| **146** | 146 | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| **147** | 147 | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| **148** | 148 | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| **149** | 149 | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

150 rows × 7 columns

Perfoming the grouping for 'sepal_length'. For any other column follow the same following steps

```
summary_stats = df.groupby('Species')['SepalLengthCm'].describe()
print(summary_stats)
```

```
                 count   mean       std  min    25%  50%  75%  max
Species
Iris-setosa       50.0  5.006  0.352490  4.3  4.800  5.0  5.2  5.8
Iris-versicolor   50.0  5.936  0.516171  4.9  5.600  5.9  6.3  7.0
Iris-virginica    50.0  6.588  0.635880  4.9  6.225  6.5  6.9  7.9
```

```
mean_sepal_length = df.groupby('Species')['SepalLengthCm'].mean()
median_sepal_length = df.groupby('Species')['SepalLengthCm'].median()
min_sepal_length = df.groupby('Species')['SepalLengthCm'].min()
max_sepal_length = df.groupby('Species')['SepalLengthCm'].max()
std_sepal_length = df.groupby('Species')['SepalLengthCm'].std()


print("Mean Sepal Length by Species Group:")
print(mean_sepal_length)
print("\nMedian Sepal Length by Species Group:")
print(median_sepal_length)
print("\nMinimum Sepal Length by Species Group:")
print(min_sepal_length)
print("\nMaximum Sepal Length by Species Group:")
print(max_sepal_length)
print("\nStandard Deviation of Sepal Length by Species Group:")
print(std_sepal_length)
```

```
Mean Sepal Length by Species Group:
Species
Iris-setosa        5.006
Iris-versicolor    5.936
Iris-virginica     6.588
Name: SepalLengthCm, dtype: float64

Median Sepal Length by Species Group:
Species
Iris-setosa        5.0
Iris-versicolor    5.9
Iris-virginica     6.5
Name: SepalLengthCm, dtype: float64

Minimum Sepal Length by Species Group:
Species
Iris-setosa        4.3
Iris-versicolor    4.9
Iris-virginica     4.9
Name: SepalLengthCm, dtype: float64

Maximum Sepal Length by Species Group:
Species
Iris-setosa        5.8
Iris-versicolor    7.0
Iris-virginica     7.9
Name: SepalLengthCm, dtype: float64

Standard Deviation of Sepal Length by Species Group:
Species
Iris-setosa        0.352490
```

```
Iris-versicolor    0.516171
Iris-virginica     0.635880
Name: SepalLengthCm, dtype: float64
```

```
seplen_by_species = df.groupby('Species')['SepalLengthCm'].apply(list)
print(seplen_by_species)
```

```
Species
Iris-setosa        [5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4, ...
Iris-versicolor    [7.0, 6.4, 6.9, 5.5, 6.5, 5.7, 6.3, 4.9, 6.6, ...
Iris-virginica     [6.3, 5.8, 7.1, 6.3, 6.5, 7.6, 4.9, 7.3, 6.7, ...
Name: SepalLengthCm, dtype: object
```