```
import pandas as pd

ds = pd.read_csv('iris.csv')

ds.head()
```

```
   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
```

```
# Group by the categorical variable 'Species'

grouped_stats = ds.groupby('Species').agg({
    'SepalLengthCm' : ['mean', 'median','min' ,'max' , 'std'],
    'SepalWidthCm' : ['mean', 'median','min' ,'max' , 'std'],
    'PetalLengthCm' : ['mean', 'median','min' ,'max' , 'std'],
    'PetalWidthCm' : ['mean', 'median','min' ,'max' , 'std'],
})

grouped_stats

ds.describe()
```

```
                Id  SepalLengthCm  SepalWidthCm  PetalLengthCm
PetalWidthCm
count  150.000000     150.000000    150.000000     150.000000
150.000000
mean    75.500000       5.843333      3.054000       3.758667
1.198667
std     43.445368       0.828066      0.433594       1.764420
0.763161
min      1.000000       4.300000      2.000000       1.000000
0.100000
25%     38.250000       5.100000      2.800000       1.600000
0.300000
50%     75.500000       5.800000      3.000000       4.350000
1.300000
75%    112.750000       6.400000      3.300000       5.100000
1.800000
max    150.000000       7.900000      4.400000       6.900000
2.500000
```

```
# Create a list with numeric values for each Categorical Response

species_numeric_list = {}

for species, group in ds.groupby('Species'):
    species_numeric_list[species] = group['SepalLengthCm'].tolist()

species_numeric_list
```

```
{'Iris-setosa': [5.1,
  4.9,
  4.7,
  4.6,
  5.0,
  5.4,
  4.6,
  5.0,
  4.4,
  4.9,
  5.4,
  4.8,
  4.8,
  4.3,
  5.8,
  5.7,
  5.4,
  5.1,
  5.7,
  5.1,
  5.4,
  5.1,
  4.6,
  5.1,
  4.8,
  5.0,
  5.0,
  5.2,
  5.2,
  4.7,
  4.8,
  5.4,
  5.2,
  5.5,
  4.9,
  5.0,
  5.5,
  4.9,
  4.4,
  5.1,
  5.0,
```

```
  4.5,
  4.4,
  5.0,
  5.1,
  4.8,
  5.1,
  4.6,
  5.3,
  5.0],
 'Iris-versicolor': [7.0,
  6.4,
  6.9,
  5.5,
  6.5,
  5.7,
  6.3,
  4.9,
  6.6,
  5.2,
  5.0,
  5.9,
  6.0,
  6.1,
  5.6,
  6.7,
  5.6,
  5.8,
  6.2,
  5.6,
  5.9,
  6.1,
  6.3,
  6.1,
  6.4,
  6.6,
  6.8,
  6.7,
  6.0,
  5.7,
  5.5,
  5.5,
  5.8,
  6.0,
  5.4,
  6.0,
  6.7,
  6.3,
  5.6,
  5.5,
```

```
  5.5,
  6.1,
  5.8,
  5.0,
  5.6,
  5.7,
  5.7,
  6.2,
  5.1,
  5.7],
 'Iris-virginica': [6.3,
  5.8,
  7.1,
  6.3,
  6.5,
  7.6,
  4.9,
  7.3,
  6.7,
  7.2,
  6.5,
  6.4,
  6.8,
  5.7,
  5.8,
  6.4,
  6.5,
  7.7,
  7.7,
  6.0,
  6.9,
  5.6,
  7.7,
  6.3,
  6.7,
  7.2,
  6.2,
  6.1,
  6.4,
  7.2,
  7.4,
  7.9,
  6.4,
  6.3,
  6.1,
  7.7,
  6.3,
  6.4,
  6.0,
```

```
  6.9,
  6.7,
  6.9,
  5.8,
  6.8,
  6.7,
  6.7,
  6.3,
  6.5,
  6.2,
  5.9]}

#basic statistical details like percentile, mean, sd,
def basic_statistics(group):
    return{
        'Mean' : group.mean(),
        'Standard Deviation' : group.std(),
        '25th Percentile' : group.quantile(.25),
        '50th Percentile' : group.median(),
        '75th Percentile' : group.quantile(.75),
        'Minimun' : group.min(),
        'Maximum' : group.max()
    }

#now apply this for all species
setosa_stats = basic_statistics(df[df['Species'] == 'Iris-setosa']
['SepalLengthCm'])
versicolor_stats = basic_statistics(df[df['Species'] == 'Iris-
versicolor']['SepalLengthCm'])
virginica_stats = basic_statistics(df[df['Species'] == 'Iris-
virginica']['SepalLengthCm'])

# display stats

print("Statistics for Iris-setosa:\n", setosa_stats)
print("\nStatistics for Iris-versicolor:\n", versicolor_stats)
print("\nStatistics for Iris-virginica:\n", virginica_stats)

Statistics for Iris-setosa:
 {'Mean': np.float64(5.006), 'Standard Deviation':
np.float64(0.35248968721345136), '25th Percentile': np.float64(4.8),
'50th Percentile': np.float64(5.0), '75th Percentile':
np.float64(5.2), 'Minimun': np.float64(4.3), 'Maximum':
np.float64(5.8)}

Statistics for Iris-versicolor:
 {'Mean': np.float64(5.936), 'Standard Deviation':
np.float64(0.5161711470638634), '25th Percentile': np.float64(5.6),
'50th Percentile': np.float64(5.9), '75th Percentile':
np.float64(6.3), 'Minimun': np.float64(4.9), 'Maximum':
```

```
np.float64(7.0)}

Statistics for Iris-virginica:
 {'Mean': np.float64(6.587999999999998), 'Standard Deviation':
np.float64(0.6358795932744322), '25th Percentile': np.float64(6.225),
'50th Percentile': np.float64(6.5), '75th Percentile':
np.float64(6.9), 'Minimun': np.float64(4.9), 'Maximum':
np.float64(7.9)}

# for age-income dataset
data = {
    'Age' : [22,25,47,52,46,56,55,60,23,34,44,53],
     'Income': [25000, 48000, 52000, 58000, 60000, 62000, 61000,
70000, 26000, 40000, 52000, 58000],
    'AgeGroup': ['Young', 'Young', 'Middle-aged', 'Middle-aged',
'Middle-aged', 'Senior', 'Senior', 'Senior', 'Young', 'Young',
'Middle-aged', 'Senior']
}

df = pd.DataFrame(data)

df

    Age  Income     AgeGroup
0    22   25000        Young
1    25   48000        Young
2    47   52000  Middle-aged
3    52   58000  Middle-aged
4    46   60000  Middle-aged
5    56   62000       Senior
6    55   61000       Senior
7    60   70000       Senior
8    23   26000        Young
9    34   40000        Young
10   44   52000  Middle-aged
11   53   58000       Senior

# Step 1a : grouped summary statistics

grouped_summary = df.groupby('AgeGroup')
['Income'].agg(['mean','median','min','max','std'])
print('\n---- Grouped Summary Statstics (Income grouped by AgeGroup)
----\n')
print(grouped_summary)


---- Grouped Summary Statstics (Income grouped by AgeGroup) ----

                mean   median    min    max           std
AgeGroup
Middle-aged  55500.0  55000.0  52000  60000   4123.105626
```

```
Senior       62750.0  61500.0  58000  70000   5123.475383
Young        34750.0  33000.0  25000  48000  11176.612486
```

```
# Step 1b: list of Income values for each agegroup

grouped_list = df.groupby('AgeGroup')['Income'].apply(list).to_dict()

print("\n--- List of Incomes per AgeGroup ---\n")
for group, values in grouped_list.items():
    print(f"{group} : {values}\n")
```

```
--- List of Incomes per AgeGroup ---

Middle-aged : [52000, 58000, 60000, 52000]

Senior : [62000, 61000, 70000, 58000]

Young : [25000, 48000, 26000, 40000]
```

```
df.describe()
```

```
             Age        Income
count  12.000000     12.000000
mean   43.083333  51000.000000
std    13.667868  14122.837727
min    22.000000  25000.000000
25%    31.750000  46000.000000
50%    46.500000  55000.000000
75%    53.500000  60250.000000
max    60.000000  70000.000000
```

```
setosa = ds[ds['Species']=='Iris-setosa']
versicolor = ds[ds['Species'] == 'Iris-versicolor']
virginica = ds[ds['Species']=='Iris-virginica']


print("\n --- Statistical Details for Iris-Setosa ---\n")
print(setosa.describe())
```

```
 --- Statistical Details for Iris-Setosa ---

             Id  SepalLengthCm  SepalWidthCm  PetalLengthCm
PetalWidthCm
count  50.00000       50.00000     50.000000      50.000000
50.00000
mean   25.50000        5.00600      3.418000       1.464000
0.24400
std    14.57738        0.35249      0.381024       0.173511
0.10721
```

```
min     1.00000        4.30000      2.300000       1.000000
0.10000
25%    13.25000        4.80000      3.125000       1.400000
0.20000
50%    25.50000        5.00000      3.400000       1.500000
0.20000
75%    37.75000        5.20000      3.675000       1.575000
0.30000
max    50.00000        5.80000      4.400000       1.900000
0.60000
```

```
print("\n--- Statistical Details for Iris-Versicolor ---\n")
print(versicolor.describe())

print("\n--- Statistical Details for Iris-Virginica ---\n")
print(virginica.describe())
```

```
--- Statistical Details for Iris-Versicolor ---

               Id  SepalLengthCm  SepalWidthCm  PetalLengthCm
PetalWidthCm
count   50.00000      50.000000     50.000000      50.000000
50.000000
mean    75.50000       5.936000      2.770000       4.260000
1.326000
std     14.57738       0.516171      0.313798       0.469911
0.197753
min     51.00000       4.900000      2.000000       3.000000
1.000000
25%     63.25000       5.600000      2.525000       4.000000
1.200000
50%     75.50000       5.900000      2.800000       4.350000
1.300000
75%     87.75000       6.300000      3.000000       4.600000
1.500000
max    100.00000       7.000000      3.400000       5.100000
1.800000

--- Statistical Details for Iris-Virginica ---

               Id  SepalLengthCm  SepalWidthCm  PetalLengthCm
PetalWidthCm
count   50.00000       50.00000     50.000000      50.000000
50.00000
mean   125.50000        6.58800      2.974000       5.552000
2.02600
std     14.57738        0.63588      0.322497       0.551895
0.27465
min    101.00000        4.90000      2.200000       4.500000
```

```
1.40000
25%    113.25000         6.22500      2.800000       5.100000
1.80000
50%    125.50000         6.50000      3.000000       5.550000
2.00000
75%    137.75000         6.90000      3.175000       5.875000
2.30000
max    150.00000         7.90000      3.800000       6.900000
2.50000
```