

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

```
In [2]: import pandas as pd
df = pd.read_csv('/home/mca01/Downloads/wages.csv')
```

```
In [3]: df
```

```
Out[3]:
```

	earn	height	sex	race	ed	age
0	79571.299011	73.89	male	white	16	49
1	96396.988643	66.23	female	white	16	62
2	48710.666947	63.77	female	white	16	33
3	80478.096153	63.22	female	other	16	95
4	82089.345498	63.08	female	white	17	43
...
1374	30173.380363	71.68	male	white	12	33
1375	24853.519514	61.31	female	white	18	86
1376	13710.671312	63.64	female	white	12	37
1377	95426.014410	71.65	male	white	12	54
1378	9575.461857	68.22	male	white	12	31

1379 rows × 6 columns

```
In [4]: df.describe()
```

Out[4]:

	earn	height	ed	age
count	1379.000000	1379.000000	1379.000000	1379.000000
mean	32446.292622	66.592640	13.354605	45.328499
std	31257.070006	3.818108	2.438741	15.789715
min	-98.580489	57.340000	3.000000	22.000000
25%	10538.790721	63.720000	12.000000	33.000000
50%	26877.870178	66.050000	13.000000	42.000000
75%	44506.215336	69.315000	15.000000	55.000000
max	317949.127955	77.210000	18.000000	95.000000

In [5]: `df.shape`

Out[5]: (1379, 6)

In [6]: `df.size`

Out[6]: 8274

Summary Statistics

In [7]: `df.min()`

Out[7]:

```
earn      -98.580489
height      57.34
sex        female
race        black
ed           3
age         22
dtype: object
```

In [8]: `df.max()`

Out[8]:

```
earn      317949.127955
height      77.21
sex         male
race        white
ed          18
age         95
dtype: object
```

Mode,Median,Mean

In [9]: `df['age'].mode()`

Out[9]:

```
0      36
dtype: int64
```

```
In [10]: df['age'].median()
```

```
Out[10]: 42.0
```

```
In [11]: df['age'].mean()
```

```
Out[11]: 45.328498912255256
```

Standard deviation

```
In [12]: round(df['age'].std(),4)
```

```
Out[12]: 15.7897
```

```
In [13]: round(df['age'].describe(),3)
```

```
Out[13]: count    1379.000
         mean      45.328
         std       15.790
         min       22.000
         25%       33.000
         50%       42.000
         75%       55.000
         max       95.000
         Name: age, dtype: float64
```

summary by groups

```
In [14]: df['age'].groupby(df['ed']).describe()
```

Out[14]:

	count	mean	std	min	25%	50%	75%	max
ed								
3	1.0	68.000000	NaN	68.0	68.00	68.0	68.00	68.0
4	2.0	67.000000	1.414214	66.0	66.50	67.0	67.50	68.0
5	5.0	69.800000	13.367872	55.0	62.00	66.0	77.00	89.0
6	7.0	67.571429	11.443443	44.0	66.00	71.0	73.50	79.0
7	3.0	53.000000	8.000000	45.0	49.00	53.0	57.00	61.0
8	28.0	57.071429	17.090020	24.0	46.00	60.0	68.50	87.0
9	23.0	53.782609	17.929043	25.0	46.00	55.0	65.50	88.0
10	37.0	52.459459	21.595517	22.0	35.00	52.0	72.00	91.0
11	39.0	46.333333	15.863452	22.0	34.00	42.0	56.50	76.0
12	520.0	44.644231	15.938900	22.0	32.00	41.0	55.00	95.0
13	119.0	44.403361	16.024515	22.0	32.00	41.0	52.00	87.0
14	192.0	44.197917	15.370550	22.0	32.00	41.0	54.25	85.0
15	66.0	43.515152	16.020412	24.0	30.00	42.0	51.75	87.0
16	187.0	42.229947	13.778804	23.0	33.00	38.0	47.00	95.0
17	70.0	43.571429	11.917070	27.0	34.25	43.0	47.00	83.0
18	80.0	49.225000	12.323668	29.0	40.75	48.0	56.25	86.0