

## Assignment-3

April 1, 2024

### 1 Assignment No:03

#Import all the required Python Libraries.

```
[ ]: import pandas as pd
```

```
[ ]: import numpy as np
```

### 2 Creation of Dataset using Microsoft Excel.

```
[ ]: !pip install -q kaggle
```

```
[ ]: from google.colab import files
```

```
[ ]: files.upload()
```

<IPython.core.display.HTML object>

Saving income\_dataset.csv to income\_dataset.csv

```
[ ]: {'income_dataset.csv':  
b'Age,Income,Occupation,YearsExperience\r\n24,38000,Research  
Assistant,2\r\n24,40000,Marketing Assistant,2\r\n22,36000,Administrative  
Assistant,1\r\n24,42000,Software Engineer,3\r\n24,39000,Customer Service  
Representative,2\r\n24,41000,Data Analyst,3\r\n28,48000,Project  
Coordinator,4\r\n28,50000,Software Developer,5\r\n22,46000,Financial  
Analyst,3\r\n28,52000,Marketing Manager,6\r\n28,49000,HR Specialist,4\r\n28,5100  
0,Teacher,5\r\n30,50000,Engineer,5\r\n30,60000,Teacher,8\r\n28,55000,Software  
Developer,6\r\n30,52000,Marketing Manager,4\r\n30,58000,Data  
Analyst,7\r\n30,53000,Accountant,5\r\n32,56000,Product  
Manager,7\r\n32,58000,Consultant,8\r\n32,54000,Operations  
Manager,6\r\n32,59000,Software Engineer,8\r\n32,57000,Financial  
Analyst,7\r\n32,60000,Marketing Director,9\r\n35,65000,Manager,10\r\n35,70000,Ph  
ysician,12\r\n35,68000,Lawyer,11\r\n35,63000,Engineer,9\r\n34,66000,Psychologist  
,10\r\n35,67000,Architect,10\r\n'}
```

### 3 Section-A

#Load the Dataset into pandas dataframe

```
[ ]: data=pd.read_csv("income_dataset.csv")
```

```
[ ]: data
```

```
[ ]:      Age  Income      Occupation  YearsExperience  
0    24   38000      Research Assistant             2  
1    24   40000      Marketing Assistant             2  
2    22   36000      Administrative Assistant         1  
3    24   42000      Software Engineer               3  
4    24   39000  Customer Service Representative       2  
5    24   41000      Data Analyst                   3  
6    28   48000      Project Coordinator             4  
7    28   50000      Software Developer             5  
8    22   46000      Financial Analyst               3  
9    28   52000      Marketing Manager               6  
10   28   49000      HR Specialist                  4  
11   28   51000      Teacher                        5  
12   30   50000      Engineer                       5  
13   30   60000      Teacher                        8  
14   28   55000      Software Developer             6  
15   30   52000      Marketing Manager               4  
16   30   58000      Data Analyst                   7  
17   30   53000      Accountant                     5  
18   32   56000      Product Manager                 7  
19   32   58000      Consultant                     8  
20   32   54000      Operations Manager              6  
21   32   59000      Software Engineer              8  
22   32   57000      Financial Analyst               7  
23   32   60000      Marketing Director              9  
24   35   65000      Manager                        10  
25   35   70000      Physician                      12  
26   35   68000      Lawyer                        11  
27   35   63000      Engineer                       9  
28   34   66000      Psychologist                    10  
29   35   67000      Architect                      10
```

### 4 Data Preprocessing:

```
[ ]: data.head()
```

```
[ ]:      Age  Income      Occupation  YearsExperience  
0    24   38000      Research Assistant             2  
1    24   40000      Marketing Assistant             2
```

2	22	36000	Administrative Assistant	1
3	24	42000	Software Engineer	3
4	24	39000	Customer Service Representative	2

```
[ ]: data.tail()
```

```
[ ]:
Age Income Occupation YearsExperience
25 35 70000 Physician 12
26 35 68000 Lawyer 11
27 35 63000 Engineer 9
28 34 66000 Psychologist 10
29 35 67000 Architect 10
```

```
[ ]: data.describe(include="all")
```

```
[ ]:
count Age Income Occupation YearsExperience
count 30.000000 30.000000 30 30.000000
unique NaN NaN 23 NaN
top NaN NaN Teacher NaN
freq NaN NaN 2 NaN
mean 29.433333 53433.333333 NaN 6.066667
std 4.099481 9449.259534 NaN 2.981938
min 22.000000 36000.000000 NaN 1.000000
25% 28.000000 48250.000000 NaN 4.000000
50% 30.000000 53500.000000 NaN 6.000000
75% 32.000000 59750.000000 NaN 8.000000
max 35.000000 70000.000000 NaN 12.000000
```

```
[ ]: data.shape
```

```
[ ]: (30, 4)
```

```
[ ]: data[0:3]
```

```
[ ]:
Age Income Occupation YearsExperience
0 24 38000 Research Assistant 2
1 24 40000 Marketing Assistant 2
2 22 36000 Administrative Assistant 1
```

```
[ ]: data.loc[0:2]
```

```
[ ]:
Age Income Occupation YearsExperience
0 24 38000 Research Assistant 2
1 24 40000 Marketing Assistant 2
2 22 36000 Administrative Assistant 1
```

```
[ ]: data.iloc[1:3]
```

```
[ ]:
Age Income Occupation YearsExperience
1 24 40000 Marketing Assistant 2
2 22 36000 Administrative Assistant 1
```

```
[ ]: data.iloc[2:6,2:6]
```

```
[ ]:
Occupation YearsExperience
2 Administrative Assistant 1
3 Software Engineer 3
4 Customer Service Representative 2
5 Data Analyst 3
```

```
[ ]: data.isnull()
```

```
[ ]:
Age Income Occupation YearsExperience
0 False False False False
1 False False False False
2 False False False False
3 False False False False
4 False False False False
5 False False False False
6 False False False False
7 False False False False
8 False False False False
9 False False False False
10 False False False False
11 False False False False
12 False False False False
13 False False False False
14 False False False False
15 False False False False
16 False False False False
17 False False False False
18 False False False False
19 False False False False
20 False False False False
21 False False False False
22 False False False False
23 False False False False
24 False False False False
25 False False False False
26 False False False False
27 False False False False
28 False False False False
29 False False False False
```

```
[ ]: data.isna()
```

```
[ ]:      Age  Income  Occupation  YearsExperience
0  False  False      False      False
1  False  False      False      False
2  False  False      False      False
3  False  False      False      False
4  False  False      False      False
5  False  False      False      False
6  False  False      False      False
7  False  False      False      False
8  False  False      False      False
9  False  False      False      False
10 False  False      False      False
11 False  False      False      False
12 False  False      False      False
13 False  False      False      False
14 False  False      False      False
15 False  False      False      False
16 False  False      False      False
17 False  False      False      False
18 False  False      False      False
19 False  False      False      False
20 False  False      False      False
21 False  False      False      False
22 False  False      False      False
23 False  False      False      False
24 False  False      False      False
25 False  False      False      False
26 False  False      False      False
27 False  False      False      False
28 False  False      False      False
29 False  False      False      False
```

```
[ ]: data.isnull().any()
```

```
[ ]: Age          False
Income          False
Occupation      False
YearsExperience False
dtype: bool
```

```
[ ]: data.isnull().sum()
```

```
[ ]: Age          0
Income          0
Occupation      0
YearsExperience 0
dtype: int64
```

```
[ ]: data.dtypes
```

```
[ ]: Age          int64
Income          int64
Occupation      object
YearsExperience int64
dtype: object
```

```
[ ]: data.columns
```

```
[ ]: Index(['Age', 'Income', 'Occupation', 'YearsExperience'], dtype='object')
```

## 5 Descriptive statistics

```
[ ]:
```

```
[ ]: print(data.Age.describe())
print(data.Income.describe())
```

```
count    30.000000
mean     29.433333
std       4.099481
min      22.000000
25%      28.000000
50%      30.000000
75%      32.000000
max      35.000000
Name: Age, dtype: float64
count    30.000000
mean    53433.333333
std     9449.259534
min    36000.000000
25%    48250.000000
50%    53500.000000
75%    59750.000000
max    70000.000000
Name: Income, dtype: float64
Mean
```

```
[ ]: print(data.Age.mean())
print(data.Income.mean())
```

```
29.433333333333334
53433.333333333336
Median
```

```
[ ]: print(data.Age.median())
print(data.Income.median())

30.0
53500.0

STD

[ ]: data.std()

<ipython-input-26-a47ac8255c06>:1: FutureWarning: The default value of
numeric_only in DataFrame.std is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
    data.std()

[ ]: Age                4.099481
Income                9449.259534
YearsExperience        2.981938
dtype: float64

[ ]: print(data.Age.std())
print(data.Income.std())

4.099481324074642
9449.259534330002

Min

[ ]: data.min()

[ ]: Age                22
Income                36000
Occupation            Accountant
YearsExperience        1
dtype: object

[ ]: print(data.Age.min())
print(data.Income.min())

22
36000

Max

[ ]: data.max()

[ ]: Age                35
Income                70000
```

```
Occupation            Teacher
YearsExperience        12
dtype: object

[ ]: print(data.Age.max())
print(data.Income.max())

35
70000

Range

[ ]: print(data.Age.max()-data.Age.min())
print(data.Income.max()-data.Income.min())

13
34000

Variance

[ ]: data.var()

<ipython-input-33-6bf595b3cfe5>:1: FutureWarning: The default value of
numeric_only in DataFrame.var is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
    data.var()

[ ]: Age                1.680575e+01
Income                8.928851e+07
YearsExperience        8.891954e+00
dtype: float64

[ ]: data.var(axis=1)

<ipython-input-34-53c0ec746524>:1: FutureWarning: Dropping of nuisance columns
in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future
version this will raise TypeError. Select only valid columns before calling the
reduction.
    data.var(axis=1)

[ ]: 0    4.810042e+08
1    5.329868e+08
2    4.317242e+08
3    5.876222e+08
4    5.066622e+08
5    5.599645e+08
6    7.674882e+08
```

```

7      8.327836e+08
8      7.049501e+08
9      9.007442e+08
10     7.998109e+08
11     8.664392e+08
12     8.327503e+08
13     1.199240e+09
14     1.007710e+09
15     9.007443e+08
16     1.120618e+09
17     9.357153e+08
18     1.044606e+09
19     1.120560e+09
20     9.713163e+08
21     1.159547e+09
22     1.082259e+09
23     1.199180e+09
24     1.407359e+09
25     1.632237e+09
26     1.540291e+09
27     1.322076e+09
28     1.451032e+09
29     1.495329e+09
dtype: float64

```

```
[ ]: print(data.Age.var())
      print(data.Income.var())
```

```

16.805747126436778
89288505.74712645

```

Mode

```
[ ]: print(data.Age.mode())
      print(data.Income.mode())
```

```

0      28
1      32
Name: Age, dtype: int64
0      50000
1      52000
2      58000
3      60000
Name: Income, dtype: int64

```

```
[ ]: print(data.Age.mode())
```

```

0      28

```

```

1      32
Name: Age, dtype: int64

```

```
[ ]: data.describe(include="all")
```

```
[ ]:
count      Age      Income Occupation  YearsExperience
count  30.000000    30.000000         30         30.000000
unique      NaN      NaN          23          NaN
top         NaN      NaN      Teacher          NaN
freq         NaN      NaN          2          NaN
mean    29.433333  53433.333333         NaN         6.066667
std       4.099481   9449.259534         NaN         2.981938
min     22.000000  36000.000000         NaN         1.000000
25%     28.000000  48250.000000         NaN         4.000000
50%     30.000000  53500.000000         NaN         6.000000
75%     32.000000  59750.000000         NaN         8.000000
max     35.000000  70000.000000         NaN        12.000000

```

```
[ ]: summary_stats = data.groupby('Age')['Income'].describe()
```

```
[ ]: summary_stats
```

```
[ ]:
count      mean      std      min      25%      50%      75%  \
Age
22      2.0  41000.000000  7071.067812  36000.0  38500.0  41000.0  43500.0
24      5.0  40000.000000  1581.138830  38000.0  39000.0  40000.0  41000.0
28      6.0  50833.333333  2483.277404  48000.0  49250.0  50500.0  51750.0
30      5.0  54600.000000  4219.004622  50000.0  52000.0  53000.0  58000.0
32      6.0  57333.333333  2160.246899  54000.0  56250.0  57500.0  58750.0
34      1.0  66000.000000         NaN  66000.0  66000.0  66000.0  66000.0
35      5.0  66600.000000  2701.851217  63000.0  65000.0  67000.0  68000.0

```

```

max
Age
22  46000.0
24  42000.0
28  55000.0
30  60000.0
32  60000.0
34  66000.0
35  70000.0

```

## 6 Section-B

```
[ ]: files.upload()
```

<IPython.core.display.HTML object>

Saving Iris.csv to Iris.csv

```
[ ]: {'Iris.csv': b'Id,SepalLengthCm,SepalWidthCm,PetalLengthCm,PetalWidthCm,Species\\n1,5.1,3.5,1.4,0.2,Iris-setosa\\n2,4.9,3.0,1.4,0.2,Iris-setosa\\n3,4.7,3.2,1.3,0.2,Iris-setosa\\n4,4.6,3.1,1.5,0.2,Iris-setosa\\n5,5.0,3.6,1.4,0.2,Iris-setosa\\n6,5.4,3.9,1.7,0.4,Iris-setosa\\n7,4.6,3.4,1.4,0.3,Iris-setosa\\n8,5.0,3.4,1.5,0.2,Iris-setosa\\n9,4.4,2.9,1.4,0.2,Iris-setosa\\n10,4.9,3.1,1.5,0.1,Iris-setosa\\n11,5.4,3.7,1.5,0.2,Iris-setosa\\n12,4.8,3.4,1.6,0.2,Iris-setosa\\n13,4.8,3.0,1.4,0.1,Iris-setosa\\n14,4.3,3.0,1.1,0.1,Iris-setosa\\n15,5.8,4.0,1.2,0.2,Iris-setosa\\n16,5.7,4.4,1.5,0.4,Iris-setosa\\n17,5.4,3.9,1.3,0.4,Iris-setosa\\n18,5.1,3.5,1.4,0.3,Iris-setosa\\n19,5.7,3.8,1.7,0.3,Iris-setosa\\n20,5.1,3.8,1.5,0.3,Iris-setosa\\n21,5.4,3.4,1.7,0.2,Iris-setosa\\n22,5.1,3.7,1.5,0.4,Iris-setosa\\n23,4.6,3.6,1.0,0.2,Iris-setosa\\n24,5.1,3.3,1.7,0.5,Iris-setosa\\n25,4.8,3.4,1.9,0.2,Iris-setosa\\n26,5.0,3.0,1.6,0.2,Iris-setosa\\n27,5.0,3.4,1.6,0.4,Iris-setosa\\n28,5.2,3.5,1.5,0.2,Iris-setosa\\n29,5.2,3.4,1.4,0.2,Iris-setosa\\n30,4.7,3.2,1.6,0.2,Iris-setosa\\n31,4.8,3.1,1.6,0.2,Iris-setosa\\n32,5.4,3.4,1.5,0.4,Iris-setosa\\n33,5.2,4.1,1.5,0.1,Iris-setosa\\n34,5.5,4.2,1.4,0.2,Iris-setosa\\n35,4.9,3.1,1.5,0.1,Iris-setosa\\n36,5.0,3.2,1.2,0.2,Iris-setosa\\n37,5.5,3.5,1.3,0.2,Iris-setosa\\n38,4.9,3.1,1.5,0.1,Iris-setosa\\n39,4.4,3.0,1.3,0.2,Iris-setosa\\n40,5.1,3.4,1.5,0.2,Iris-setosa\\n41,5.0,3.5,1.3,0.3,Iris-setosa\\n42,4.5,2.3,1.3,0.3,Iris-setosa\\n43,4.4,3.2,1.3,0.2,Iris-setosa\\n44,5.0,3.5,1.6,0.6,Iris-setosa\\n45,5.1,3.8,1.9,0.4,Iris-setosa\\n46,4.8,3.0,1.4,0.3,Iris-setosa\\n47,5.1,3.8,1.6,0.2,Iris-setosa\\n48,4.6,3.2,1.4,0.2,Iris-setosa\\n49,5.3,3.7,1.5,0.2,Iris-setosa\\n50,5.0,3.3,1.4,0.2,Iris-setosa\\n51,7.0,3.2,4.7,1.4,Iris-versicolor\\n52,6.4,3.2,4.5,1.5,Iris-versicolor\\n53,6.9,3.1,4.9,1.5,Iris-versicolor\\n54,5.5,2.3,4.0,1.3,Iris-versicolor\\n55,6.5,2.8,4.6,1.5,Iris-versicolor\\n56,5.7,2.8,4.5,1.3,Iris-versicolor\\n57,6.3,3.3,4.7,1.6,Iris-versicolor\\n58,4.9,2.4,3.3,1.0,Iris-versicolor\\n59,6.6,2.9,4.6,1.3,Iris-versicolor\\n60,5.2,2.7,3.9,1.4,Iris-versicolor\\n61,5.0,2.0,3.5,1.0,Iris-versicolor\\n62,5.9,3.0,4.2,1.5,Iris-versicolor\\n63,6.0,2.2,4.0,1.0,Iris-versicolor\\n64,6.1,2.9,4.7,1.4,Iris-versicolor\\n65,5.6,2.9,3.6,1.3,Iris-versicolor\\n66,6.7,3.1,4.4,1.4,Iris-versicolor\\n67,5.6,3.0,4.5,1.5,Iris-versicolor\\n68,5.8,2.7,4.1,1.0,Iris-versicolor\\n69,6.2,2.2,4.5,1.5,Iris-versicolor\\n70,5.6,2.5,3.9,1.1,Iris-versicolor\\n71,5.9,3.2,4.8,1.8,Iris-versicolor\\n72,6.1,2.8,4.0,1.3,Iris-versicolor\\n73,6.3,2.5,4.9,1.5,Iris-versicolor\\n74,6.1,2.8,4.7,1.2,Iris-versicolor\\n75,6.4,2.9,4.3,1.3,Iris-versicolor\\n76,6.6,3.0,4.4,1.4,Iris-versicolor\\n77,6.8,2.8,4.8,1.4,Iris-versicolor\\n78,6.7,3.0,5.0,1.7,Iris-versicolor\\n79,6.0,2.9,4.5,1.5,Iris-versicolor\\n80,5.7,2.6,3.5,1.0,Iris-versicolor\\n81,5.5,2.4,3.8,1.1,Iris-versicolor\\n82,5.5,2.4,3.7,1.0,Iris-versicolor\\n83,5.8,2.7,3.9,1.2,Iris-versicolor\\n84,6.0,2.7,5.1,1.6,Iris-versicolor\\n85,5.4,3.0,4.5,1.5,Iris-versicolor\\n86,6.0,3.4,4.5,1.6,Iris-versicolor\\n87,6.7,3.1,4.7,1.5,Iris-versicolor\\n88,6.3,2.3,4.4,1.3,Iris-
```

```
versicolor\\n89,5.6,3.0,4.1,1.3,Iris-versicolor\\n90,5.5,2.5,4.0,1.3,Iris-versicolor\\n91,5.5,2.6,4.4,1.2,Iris-versicolor\\n92,6.1,3.0,4.6,1.4,Iris-versicolor\\n93,5.8,2.6,4.0,1.2,Iris-versicolor\\n94,5.0,2.3,3.3,1.0,Iris-versicolor\\n95,5.6,2.7,4.2,1.3,Iris-versicolor\\n96,5.7,3.0,4.2,1.2,Iris-versicolor\\n97,5.7,2.9,4.2,1.3,Iris-versicolor\\n98,6.2,2.9,4.3,1.3,Iris-versicolor\\n99,5.1,2.5,3.0,1.1,Iris-versicolor\\n100,5.7,2.8,4.1,1.3,Iris-versicolor\\n101,6.3,3.3,6.0,2.5,Iris-virginica\\n102,5.8,2.7,5.1,1.9,Iris-virginica\\n103,7.1,3.0,5.9,2.1,Iris-virginica\\n104,6.3,2.9,5.6,1.8,Iris-virginica\\n105,6.5,3.0,5.8,2.2,Iris-virginica\\n106,7.6,3.0,6.6,2.1,Iris-virginica\\n107,4.9,2.5,4.5,1.7,Iris-virginica\\n108,7.3,2.9,6.3,1.8,Iris-virginica\\n109,6.7,2.5,5.8,1.8,Iris-virginica\\n110,7.2,3.6,6.1,2.5,Iris-virginica\\n111,6.5,3.2,5.1,2.0,Iris-virginica\\n112,6.4,2.7,5.3,1.9,Iris-virginica\\n113,6.8,3.0,5.5,2.1,Iris-virginica\\n114,5.7,2.5,5.0,2.0,Iris-virginica\\n115,5.8,2.8,5.1,2.4,Iris-virginica\\n116,6.4,3.2,5.3,2.3,Iris-virginica\\n117,6.5,3.0,5.5,1.8,Iris-virginica\\n118,7.7,3.8,6.7,2.2,Iris-virginica\\n119,7.7,2.6,6.9,2.3,Iris-virginica\\n120,6.0,2.2,5.0,1.5,Iris-virginica\\n121,6.9,3.2,5.7,2.3,Iris-virginica\\n122,5.6,2.8,4.9,2.0,Iris-virginica\\n123,7.7,2.8,6.7,2.0,Iris-virginica\\n124,6.3,2.7,4.9,1.8,Iris-virginica\\n125,6.7,3.3,5.7,2.1,Iris-virginica\\n126,7.3,2.6,6.0,1.8,Iris-virginica\\n127,6.2,2.8,4.8,1.8,Iris-virginica\\n128,6.1,3.0,4.9,1.8,Iris-virginica\\n129,6.4,2.8,5.6,2.1,Iris-virginica\\n130,7.2,3.0,5.8,1.6,Iris-virginica\\n131,7.4,2.8,6.1,1.9,Iris-virginica\\n132,7.9,3.8,6.4,2.0,Iris-virginica\\n133,6.4,2.8,5.6,2.2,Iris-virginica\\n134,6.3,2.8,5.1,1.5,Iris-virginica\\n135,6.1,2.6,5.6,1.4,Iris-virginica\\n136,7.7,3.0,6.1,2.3,Iris-virginica\\n137,6.3,3.4,5.6,2.4,Iris-virginica\\n138,6.4,3.1,5.5,1.8,Iris-virginica\\n139,6.0,3.0,4.8,1.8,Iris-virginica\\n140,6.9,3.1,5.4,2.1,Iris-virginica\\n141,6.7,3.1,5.6,2.4,Iris-virginica\\n142,6.9,3.1,5.1,2.3,Iris-virginica\\n143,5.8,2.7,5.1,1.9,Iris-virginica\\n144,6.8,3.2,5.9,2.3,Iris-virginica\\n145,6.7,3.3,5.7,2.5,Iris-virginica\\n146,6.7,3.0,5.2,2.3,Iris-virginica\\n147,6.3,2.5,5.0,1.9,Iris-virginica\\n148,6.5,3.0,5.2,2.0,Iris-virginica\\n149,6.2,3.4,5.4,2.3,Iris-virginica\\n150,5.9,3.0,5.1,1.8,Iris-virginica\\n'}
```

```
[ ]: iris=pd.read_csv("Iris.csv")
```

```
[ ]: iris
```

```
[ ]:      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1           5.1           3.5           1.4           0.2
1      2           4.9           3.0           1.4           0.2
2      3           4.7           3.2           1.3           0.2
3      4           4.6           3.1           1.5           0.2
4      5           5.0           3.6           1.4           0.2
..      ...             ...             ...             ...
145    146           6.7           3.0           5.2           2.3
146    147           6.3           2.5           5.0           1.9
147    148           6.5           3.0           5.2           2.0
```

```

148 149      6.2      3.4      5.4      2.3
149 150      5.9      3.0      5.1      1.8

```

```

      Species
0      Iris-setosa
1      Iris-setosa
2      Iris-setosa
3      Iris-setosa
4      Iris-setosa
..      ...
145 Iris-virginica
146 Iris-virginica
147 Iris-virginica
148 Iris-virginica
149 Iris-virginica

```

[150 rows x 6 columns]

```
[ ]: iris.describe(include="all")
```

```

[ ]:
count      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
unique      NaN              NaN              NaN              NaN              NaN
top         NaN              NaN              NaN              NaN              NaN
freq        NaN              NaN              NaN              NaN              NaN
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      3.435000      1.300000
75%      112.750000      6.400000      3.300000      5.100000      1.800000
max      150.000000      7.900000      4.400000      6.900000      2.500000

```

```

      Species
count      150
unique       3
top      Iris-setosa
freq        50
mean        NaN
std         NaN
min         NaN
25%         NaN
50%         NaN
75%         NaN
max         NaN

```

```
[ ]: iris.shape
```

```
[ ]: (150, 6)
```

```
[ ]: iris.isnull()
```

```

[ ]:
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
0  False              False              False              False              False
1  False              False              False              False              False
2  False              False              False              False              False
3  False              False              False              False              False
4  False              False              False              False              False
..      ...
145 False              False              False              False              False
146 False              False              False              False              False
147 False              False              False              False              False
148 False              False              False              False              False
149 False              False              False              False              False

```

[150 rows x 6 columns]

```
[ ]: iris.isna()
```

```

[ ]:
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
0  False              False              False              False              False
1  False              False              False              False              False
2  False              False              False              False              False
3  False              False              False              False              False
4  False              False              False              False              False
..      ...
145 False              False              False              False              False
146 False              False              False              False              False
147 False              False              False              False              False
148 False              False              False              False              False
149 False              False              False              False              False

```

[150 rows x 6 columns]

```
[ ]: iris.isnull().any()
```

```

[ ]: Id              False
      SepalLengthCm  False
      SepalWidthCm   False
      PetalLengthCm  False
      PetalWidthCm   False
      Species        False
      dtype: bool

```

```
[ ]: iris.isnull().sum()
```

```
[ ]: Id          0
     SepalLengthCm 0
     SepalWidthCm  0
     PetalLengthCm 0
     PetalWidthCm  0
     Species       0
     dtype: int64
```

```
[ ]: iris.SepalLengthCm.isnull().sum()
```

```
[ ]: 0
```

```
[ ]: iris.dtypes
```

```
[ ]: Id          int64
     SepalLengthCm float64
     SepalWidthCm  float64
     PetalLengthCm float64
     PetalWidthCm  float64
     Species       object
     dtype: object
```

```
[ ]: iris.columns
```

```
[ ]: Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
           'Species'],
          dtype='object')
```

## 7 Descriptive statistics

```
[ ]: col_names = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width', 'Species']
```

```
[ ]: #Load all rows with Iris-setosa species in variable irisSet
     irisSet = (iris['Species']== 'Iris-setosa')
```

```
[ ]: # To display basic statistical details like percentile, mean,standard deviation,
     ↳etc. forIris-setosa use describe
```

```
print('Iris-setosa')
print(iris[irisSet].describe())
```

```
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
```

```
[ ]: # Load all rows with Iris-versicolor species in variable irisVer
     irisVer = (iris['Species']== 'Iris-versicolor')
```

```
[ ]: # To display basic statistical details like percentile, mean,standard deviation,
     ↳etc. forIris-versicolor use describe
```

```
print('Iris-versicolor')
print(iris[irisVer].describe())
```

```
Iris-versicolor
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  50.00000      50.00000      50.000000      50.000000      50.00000
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
```

```
[ ]: # Load all rows with Iris-virginica species in variable irisVir
     irisVir = (iris['Species']== 'Iris-virginica')
```

```
[ ]: # To display basic statistical details like percentile, mean,standard deviation,
     ↳etc. forIris-virginica use describe
```

```
print('Iris-virginica')
print(iris[irisVir].describe())
```

```
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
```



```
[ ]: print(iris.SepalLengthCm.describe())
print(iris.SepalWidthCm.describe())
print(iris.PetalLengthCm.describe())
print(iris.PetalWidthCm.describe())
# print(iris.Species.describe(include='all'))
```

```
count    150.000000
mean      5.843333
std       0.828066
min       4.300000
25%      5.100000
50%      5.800000
75%      6.400000
max       7.900000
Name: SepalLengthCm, dtype: float64
count    150.000000
mean      3.054000
std       0.433594
min       2.000000
25%      2.800000
50%      3.000000
75%      3.300000
max       4.400000
Name: SepalWidthCm, dtype: float64
count    150.000000
mean      3.758667
std       1.764420
min       1.000000
25%      1.600000
50%      4.350000
75%      5.100000
max       6.900000
Name: PetalLengthCm, dtype: float64
count    150.000000
mean      1.198667
std       0.763161
min       0.100000
25%      0.300000
50%      1.300000
75%      1.800000
max       2.500000
Name: PetalWidthCm, dtype: float64
```

```
[ ]: 
[ ]: 5.8433333333333334
```

Mean

```
[ ]: print(iris.SepalLengthCm.mean())
print(iris.SepalWidthCm.mean())
print(iris.PetalLengthCm.mean())
print(iris.PetalWidthCm.mean())
```

```
5.8433333333333334
3.0540000000000003
3.7586666666666666
1.1986666666666668
```

Median

```
[ ]: print(iris.SepalLengthCm.median())
print(iris.SepalWidthCm.median())
print(iris.PetalLengthCm.median())
print(iris.PetalWidthCm.median())
```

```
5.8
3.0
4.35
1.3
```

STD

```
[ ]: iris.std()
```

```
<ipython-input-56-c5ab3f85284a>:1: FutureWarning: The default value of
numeric_only in DataFrame.std is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
```

```
iris.std()
```

```
[ ]: Id          43.445368
SepalLengthCm   0.828066
SepalWidthCm    0.433594
PetalLengthCm   1.764420
PetalWidthCm    0.763161
dtype: float64
```

```
[ ]: print(iris.SepalLengthCm.std())
print(iris.SepalWidthCm.std())
print(iris.PetalLengthCm.std())
print(iris.PetalWidthCm.std())
```

```
0.828066127977863
0.4335943113621737
1.7644204199522626
0.7631607417008411
```

```

Min
[ ]: iris.min()

[ ]: Id          1
     SepalLengthCm  4.3
     SepalWidthCm   2.0
     PetalLengthCm  1.0
     PetalWidthCm   0.1
     Species       Iris-setosa
     dtype: object

[ ]: print(iris.SepalLengthCm.min())
     print(iris.SepalWidthCm.min())
     print(iris.PetalLengthCm.min())
     print(iris.PetalWidthCm.min())

4.3
2.0
1.0
0.1

Max
[ ]: iris.max()

[ ]: Id          150
     SepalLengthCm  7.9
     SepalWidthCm   4.4
     PetalLengthCm  6.9
     PetalWidthCm   2.5
     Species       Iris-virginica
     dtype: object

[ ]: print(iris.SepalLengthCm.max())
     print(iris.SepalWidthCm.max())
     print(iris.PetalLengthCm.max())
     print(iris.PetalWidthCm.max())

7.9
4.4
6.9
2.5

Range
[ ]: print(iris.SepalLengthCm.max()-iris.SepalLengthCm.min())
     print(iris.SepalWidthCm.max()-iris.SepalWidthCm.min())
     print(iris.PetalLengthCm.max()-iris.PetalLengthCm.min())

```

```

print(iris.PetalWidthCm.max()-iris.PetalWidthCm.min())

3.6000000000000005
2.4000000000000004
5.9
2.4

Variance
[ ]: iris.var()

<ipython-input-63-88c693b2b474>:1: FutureWarning: The default value of
numeric_only in DataFrame.var is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
     iris.var()

[ ]: Id          1887.500000
     SepalLengthCm  0.685694
     SepalWidthCm   0.188004
     PetalLengthCm  3.113179
     PetalWidthCm   0.582414
     dtype: float64

[ ]: iris.var(axis=1)

<ipython-input-64-4c47fcb408fe>:1: FutureWarning: Dropping of nuisance columns
in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future
version this will raise TypeError. Select only valid columns before calling the
reduction.
     iris.var(axis=1)

[ ]: 0          4.043
     1          3.140
     2          3.077
     3          3.287
     4          4.688
     ...
     145        4018.843
     146        4097.323
     147        4140.268
     148        4188.578
     149        4268.783
     Length: 150, dtype: float64

[ ]: print(iris.SepalLengthCm.var())
     print(iris.SepalWidthCm.var())

```

```
print(iris.PetalLengthCm.var())
print(iris.PetalWidthCm.var())
```

```
0.6856935123042507
0.1880040268456376
3.113179418344519
0.582414317673378
```

Quantile/Percentile

```
[ ]: print(iris.SepalLengthCm.quantile(0.25))
print(iris.SepalWidthCm.quantile(0.25))
print(iris.PetalLengthCm.quantile(0.25))
print(iris.PetalWidthCm.quantile(0.25))
```

```
5.1
2.8
1.6
0.3
```

```
[ ]: print(iris.SepalLengthCm.quantile(0.50))
print(iris.SepalWidthCm.quantile(0.50))
print(iris.PetalLengthCm.quantile(0.50))
print(iris.PetalWidthCm.quantile(0.50))
```

```
5.8
3.0
4.35
1.3
```

```
[ ]: print(iris.SepalLengthCm.quantile(0.75))
print(iris.SepalWidthCm.quantile(0.75))
print(iris.PetalLengthCm.quantile(0.75))
print(iris.PetalWidthCm.quantile(0.75))
```

```
6.4
3.3
5.1
1.8
```

Skewness

```
[ ]: print(iris.SepalLengthCm.skew())
print(iris.SepalWidthCm.skew())
print(iris.PetalLengthCm.skew())
print(iris.PetalWidthCm.skew())
```

```
0.3149109566369728
0.3340526621720866
```

```
-0.27446425247378287
-0.10499656214412734
```

```
[ ]: print((3*(iris.SepalLengthCm.mean()-iris.SepalLengthCm.median()))/(iris.
      .SepalLengthCm.std()))
```

```
0.15699229277432417
```

Mode

```
[ ]: print(iris.Species.mode())
```

```
0      Iris-setosa
1      Iris-versicolor
2      Iris-virginica
Name: Species, dtype: object
```

```
[ ]: print(iris.SepalLengthCm.mode())
```

```
0      5.0
Name: SepalLengthCm, dtype: float64
```