Assignment No : 3

Descriptive Statistics - Measures of Central Tendency and Variablity

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.

# Load required libraries

In [104]:

**import** pandas **as** pd **import** numpy **as** np **import** requests

# Load dataset

In [105]:

df**=**pd.read\_csv("Student Data1.csv") df

Out[105]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Roll** | **Name** | **Gender** | **Physics** | **Chemistry** | **Maths** |
| **0** 1 | Yash Kumbhar | Male | 45 | 78 | 72 |
| **1** 2 | Raj Kokare | Male | 35 | 38 | 67 |
| **2** 3 | Pavan Jadhav | Male | 89 | 100 | 99 |
| **3** 4 | Yuvraj Lokhande | Male | 45 | 43 | 41 |
| **4** 5 | Preksha Shaha | Female | 90 | 67 | 78 |
| **5** 6 | Saurabha Kokare | Male | 45 | 78 | 67 |
| **6** 7 | Addi Bankar | Female | 67 | 56 | 87 |
| **7** 8 | Ganesh Waikar | Male | 56 | 65 | 78 |
| **8** 9 | Rushi Wable | Male | 37 | 67 | 56 |
| **9** 10 | Pranjal Jagtap | Female | 78 | 67 | 45 |
| **10** 11 | Mayur Wable | Male | 56 | 78 | 45 |
| **11** 12 | Jeevan More | Male | 67 | 45 | 78 |
| **12** 13 | Suyog Kokare | Male | 78 | 87 | 100 |
| **13** 14 | Shashi Kokare | Male | 76 | 78 | 90 |
| **14** 15 | Shubham Kumbhar | Male | 56 | 90 | 89 |
| **15** 16 | Anil Kamble | Male | 54 | 89 | 67 |
| **16** 17 | Diksha Thorat | Female | 55 | 67 | 90 |
| **17** 18 | Yash Khodake | Male | 52 | 56 | 90 |
| **18** 19 | Karan Mahamuni | Male | 89 | 90 | 67 |
| **19** 20 | Gauri Jagtap | Female | 34 | 76 | 89 |

In [106]:

df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 20 entries, 0 to 19

Data columns (total 6 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | Roll | 20 | non-null |  | int64 |
| 1 |  | Name | 20 | non-null |  | object |
| 2 |  | Gender | 20 | non-null |  | object |
| 3 |  | Physics | 20 | non-null |  | int64 |
| 4 |  | Chemistry | 20 | non-null |  | int64 |
| 5 |  | Maths | 20 | non-null |  | int64 |

dtypes: int64(4), object(2) memory usage: 1.1+ KB

info() : provide information about dataset.

In [107]:

df.shape

Out[107]:

(20, 6)

shape : it gives dimension of dataset

In [108]:

df.head()

Out[108]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Roll** | **Name** | **Gender** | **Physics** | **Chemistry** | **Maths** |
| **0** 1 | Yash Kumbhar | Male | 45 | 78 | 72 |
| **1** 2 | Raj Kokare | Male | 35 | 38 | 67 |
| **2** 3 | Pavan Jadhav | Male | 89 | 100 | 99 |
| **3** 4 | Yuvraj Lokhande | Male | 45 | 43 | 41 |
| **4** 5 | Preksha Shaha | Female | 90 | 67 | 78 |

In [109]:

df.tail()

Out[109]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Roll** | **Name** | **Gender** | **Physics** | **Chemistry** | **Maths** |
| **15** 16 | Anil Kamble | Male | 54 | 89 | 67 |
| **16** 17 | Diksha Thorat | Female | 55 | 67 | 90 |
| **17** 18 | Yash Khodake | Male | 52 | 56 | 90 |
| **18** 19 | Karan Mahamuni | Male | 89 | 90 | 67 |
| **19** 20 | Gauri Jagtap | Female | 34 | 76 | 89 |

head() : it show first 5 rows tail() : it show last 5 rows

In [110]:

df.describe()

Out[110]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Roll** | **Physics** | **Chemistry** | **Maths** |
| **count** | 20.00000 | 20.00000 | 20.000000 | 20.00000 |
| **mean** | 10.50000 | 60.20000 | 70.750000 | 74.75000 |
| **std** | 5.91608 | 18.15315 | 16.954661 | 17.79969 |
| **min** | 1.00000 | 34.00000 | 38.000000 | 41.00000 |
| **25%** | 5.75000 | 45.00000 | 62.750000 | 67.00000 |
| **50%** | 10.50000 | 56.00000 | 71.500000 | 78.00000 |
| **75%** | 15.25000 | 76.50000 | 80.250000 | 89.25000 |
| **max** | 20.00000 | 90.00000 | 100.000000 | 100.00000 |

describe() : it is used to describe the dataset (provide information like mean ,std,max and min value)

# find mean ,median, minimum, maximum, standard deviation for every column

* 1. mean :- avarage value of all datapoints.

In [111]:

df[['Physics','Chemistry','Maths']].mean()

|  |  |
| --- | --- |
| Out[111]: |  |
| Physics | 60.20 |
| Chemistry | 70.75 |
| Maths | 74.75 |

dtype: float64

* 1. median :- it is the middle number in a set of data when the data is arranged in ascending or descending order

In [112]:

df[['Physics','Chemistry','Maths']].median()

|  |  |
| --- | --- |
| Out[112]: |  |
| Physics | 56.0 |
| Chemistry | 71.5 |
| Maths | 78.0 |

dtype: float64

* 1. Minimum value :-

In [113]:

df[['Physics','Chemistry','Maths']].min()

Out[113]:

Physics 34

Chemistry 38

Maths 41

dtype: int64

df.min() :- minimum value from columns in given dataset which contain numeric data

* 1. maximum value :-

In [114]:

df[['Physics','Chemistry','Maths']].max()

Out[114]:

Physics 90

Chemistry 100

Maths 100

dtype: int64

df.max() :- maximum value from columns in given dataset which contain numeric data

* 1. standard deviation :-

In [115]:

df[['Physics','Chemistry','Maths']].std()

|  |  |
| --- | --- |
| Out[115]: |  |
| Physics | 18.153150 |
| Chemistry | 16.954661 |
| Maths | 17.799690 |

dtype: float64

df.std() :- standard deviation of columns from given datset which contains numeric data.

# group by gender

In [116]:

df2**=**df.groupby('Gender') df2

Out[116]:

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001545A6A5130>

In [117]:

**for** gender,gender\_f **in** df2: print(gender)

print(gender\_f)

Female

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Roll | Name | Gender | Physics | Chemistry | Maths |
| 4 | 5 | Preksha Shaha | Female | 90 | 67 | 78 |
| 6 | 7 | Addi Bankar | Female | 67 | 56 | 87 |
| 9 | 10 | Pranjal Jagtap | Female | 78 | 67 | 45 |
| 16 | 17 | Diksha Thorat | Female | 55 | 67 | 90 |
| 19 | 20 | Gauri Jagtap | Female | 34 | 76 | 89 |
| Male |  |  |  |  |  |  |
|  | Roll | Name | Gender | Physics | Chemistry | Maths |
| 0 | 1 | Yash Kumbhar | Male | 45 | 78 | 72 |
| 1 | 2 | Raj Kokare | Male | 35 | 38 | 67 |
| 2 | 3 | Pavan Jadhav | Male | 89 | 100 | 99 |
| 3 | 4 | Yuvraj Lokhande | Male | 45 | 43 | 41 |
| 5 | 6 | Saurabha Kokare | Male | 45 | 78 | 67 |
| 7 | 8 | Ganesh Waikar | Male | 56 | 65 | 78 |
| 8 | 9 | Rushi Wable | Male | 37 | 67 | 56 |
| 10 | 11 | Mayur Wable | Male | 56 | 78 | 45 |
| 11 | 12 | Jeevan More | Male | 67 | 45 | 78 |
| 12 | 13 | Suyog Kokare | Male | 78 | 87 | 100 |
| 13 | 14 | Shashi Kokare | Male | 76 | 78 | 90 |
| 14 | 15 | Shubham Kumbhar | Male | 56 | 90 | 89 |
| 15 | 16 | Anil Kamble | Male | 54 | 89 | 67 |
| 17 | 18 | Yash Khodake | Male | 52 | 56 | 90 |
| 18 | 19 | Karan Mahamuni | Male | 89 | 90 | 67 |

groupby() : used to arrange data according to categorial attribute.

In [154]:

df2.get\_group('Male') *#gives group of male students*

Out[154]:

**Roll Name Gender Physics Chemistry Maths**

1. 1 Yash Kumbhar Male 45 78 72
2. 2 Raj Kokare Male 35 38 67
3. 3 Pavan Jadhav Male 89 100 99
4. 4 Yuvraj Lokhande Male 45 43 41

**5** 6 Saurabha Kokare Male 45 78 67

1. 8 Ganesh Waikar Male 56 65 78
2. 9 Rushi Wable Male 37 67 56
3. 11 Mayur Wable Male 56 78 45
4. 12 Jeevan More Male 67 45 78
5. 13 Suyog Kokare Male 78 87 100
6. 14 Shashi Kokare Male 76 78 90
7. 15 Shubham Kumbhar Male 56 90 89
8. 16 Anil Kamble Male 54 89 67
9. 18 Yash Khodake Male 52 56 90
10. 19 Karan Mahamuni Male 89 90 67

In [155]:

df2.get\_group('Female') *#gives group of female students*

Out[155]:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Roll** | **Name** | **Gender** | **Physics** | **Chemistry** | **Maths** |
| **4** | 5 | Preksha Shaha | Female | 90 | 67 | 78 |
| **6** | 7 | Addi Bankar | Female | 67 | 56 | 87 |
| **9** | 10 | Pranjal Jagtap | Female | 78 | 67 | 45 |
| **16** | 17 | Diksha Thorat | Female | 55 | 67 | 90 |
| **19** | 20 | Gauri Jagtap | Female | 34 | 76 | 89 |

In [156]:

df2[['Physics','Chemistry','Maths']].max()

Out[156]:

|  |  |  |  |
| --- | --- | --- | --- |
| **Gender** | **Physics** | **Chemistry** | **Maths** |
| **Female** | 90 | 76 | 90 |
| **Male** | 89 | 100 | 100 |

In [157]:

df2[['Physics','Chemistry','Maths']].mean()

Out[157]:

**Physics Chemistry Maths**

**Gender**

|  |  |  |
| --- | --- | --- |
| **Female** 64.800000 | 66.600000 | 77.800000 |
| **Male** 58.666667 | 72.133333 | 73.733333 |

In [158]:

df2[['Physics','Chemistry','Maths']].min()

Out[158]:

|  |  |  |  |
| --- | --- | --- | --- |
| **Gender** | **Physics** | **Chemistry** | **Maths** |
| **Female** | 34 | 56 | 45 |
| **Male** | 35 | 38 | 41 |

In [159]:

df[['Physics','Chemistry','Maths']].std()

|  |  |
| --- | --- |
| Out[159]: |  |
| Physics | 18.153150 |
| Chemistry | 16.954661 |
| Maths | 17.799690 |

dtype: float64

1. 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.

In [160]:

url**=**'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data' r**=**requests.get(url)

In [161]:

df4**=**pd.read\_csv(url) df4

Out[161]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **5.1** | **3.5** | **1.4** | **0.2** | **Iris-setosa** |
| **0** | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **1** | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **2** | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **3** | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| **4** | 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa |
| **...** | ... | ... | ... | ... | ... |
| **144** | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| **145** | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| **146** | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| **147** | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| **148** | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

149 rows × 5 columns

In [162]:

df4.columns**=**{"A","B","C","D","E"}

In [163]:

df4

Out[163]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **E** | **C** | **D** | **B** |
| **0** | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **1** | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **2** | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **3** | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| **4** | 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa |
| **...** | ... | ... | ... | ... | ... |
| **144** | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| **145** | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| **146** | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| **147** | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| **148** | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

149 rows × 5 columns

In [164]:

df5**=**df4.groupby("B") df5

Out[164]:

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001545A0AE490>

In [165]:

df5.get\_group('Iris-setosa')

Out[165]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **E** | **C** | **D** | **B** |
| **0** 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **1** 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **2** 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **3** 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| **4** 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa |
| **5** 4.6 | 3.4 | 1.4 | 0.3 | Iris-setosa |
| **6** 5.0 | 3.4 | 1.5 | 0.2 | Iris-setosa |
| **7** 4.4 | 2.9 | 1.4 | 0.2 | Iris-setosa |
| **8** 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa |
| **9** 5.4 | 3.7 | 1.5 | 0.2 | Iris-setosa |
| **10** 4.8 | 3.4 | 1.6 | 0.2 | Iris-setosa |
| **11** 4.8 | 3.0 | 1.4 | 0.1 | Iris-setosa |
| **12** 4.3 | 3.0 | 1.1 | 0.1 | Iris-setosa |
| **13** 5.8 | 4.0 | 1.2 | 0.2 | Iris-setosa |
| **14** 5.7 | 4.4 | 1.5 | 0.4 | Iris-setosa |
| **15** 5.4 | 3.9 | 1.3 | 0.4 | Iris-setosa |
| **16** 5.1 | 3.5 | 1.4 | 0.3 | Iris-setosa |
| **17** 5.7 | 3.8 | 1.7 | 0.3 | Iris-setosa |
| **18** 5.1 | 3.8 | 1.5 | 0.3 | Iris-setosa |
| **19** 5.4 | 3.4 | 1.7 | 0.2 | Iris-setosa |
| **20** 5.1 | 3.7 | 1.5 | 0.4 | Iris-setosa |
| **21** 4.6 | 3.6 | 1.0 | 0.2 | Iris-setosa |
| **22** 5.1 | 3.3 | 1.7 | 0.5 | Iris-setosa |
| **23** 4.8 | 3.4 | 1.9 | 0.2 | Iris-setosa |
| **24** 5.0 | 3.0 | 1.6 | 0.2 | Iris-setosa |
| **25** 5.0 | 3.4 | 1.6 | 0.4 | Iris-setosa |
| **26** 5.2 | 3.5 | 1.5 | 0.2 | Iris-setosa |
| **27** 5.2 | 3.4 | 1.4 | 0.2 | Iris-setosa |
| **28** 4.7 | 3.2 | 1.6 | 0.2 | Iris-setosa |
| **29** 4.8 | 3.1 | 1.6 | 0.2 | Iris-setosa |
| **30** 5.4 | 3.4 | 1.5 | 0.4 | Iris-setosa |
| **31** 5.2 | 4.1 | 1.5 | 0.1 | Iris-setosa |
| **32** 5.5 | 4.2 | 1.4 | 0.2 | Iris-setosa |
| **33** 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **E** | **C** | **D** | **B** |
| **34** 5.0 | 3.2 | 1.2 | 0.2 | Iris-setosa |
| **35** 5.5 | 3.5 | 1.3 | 0.2 | Iris-setosa |
| **36** 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa |
| **37** 4.4 | 3.0 | 1.3 | 0.2 | Iris-setosa |
| **38** 5.1 | 3.4 | 1.5 | 0.2 | Iris-setosa |
| **39** 5.0 | 3.5 | 1.3 | 0.3 | Iris-setosa |
| **40** 4.5 | 2.3 | 1.3 | 0.3 | Iris-setosa |
| **41** 4.4 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **42** 5.0 | 3.5 | 1.6 | 0.6 | Iris-setosa |
| **43** 5.1 | 3.8 | 1.9 | 0.4 | Iris-setosa |
| **44** 4.8 | 3.0 | 1.4 | 0.3 | Iris-setosa |
| **45** 5.1 | 3.8 | 1.6 | 0.2 | Iris-setosa |
| **46** 4.6 | 3.2 | 1.4 | 0.2 | Iris-setosa |
| **47** 5.3 | 3.7 | 1.5 | 0.2 | Iris-setosa |
| **48** 5.0 | 3.3 | 1.4 | 0.2 | Iris-setosa |

In [166]:

df5.get\_group("Iris-virginica")

Out[166]:

**A E C D B**

1. 6.3 3.3 6.0 2.5 Iris-virginica
2. 5.8 2.7 5.1 1.9 Iris-virginica
3. 7.1 3.0 5.9 2.1 Iris-virginica
4. 6.3 2.9 5.6 1.8 Iris-virginica
5. 6.5 3.0 5.8 2.2 Iris-virginica
6. 7.6 3.0 6.6 2.1 Iris-virginica
7. 4.9 2.5 4.5 1.7 Iris-virginica
8. 7.3 2.9 6.3 1.8 Iris-virginica
9. 6.7 2.5 5.8 1.8 Iris-virginica
10. 7.2 3.6 6.1 2.5 Iris-virginica
11. 6.5 3.2 5.1 2.0 Iris-virginica
12. 6.4 2.7 5.3 1.9 Iris-virginica
13. 6.8 3.0 5.5 2.1 Iris-virginica
14. 5.7 2.5 5.0 2.0 Iris-virginica
15. 5.8 2.8 5.1 2.4 Iris-virginica
16. 6.4 3.2 5.3 2.3 Iris-virginica
17. 6.5 3.0 5.5 1.8 Iris-virginica
18. 7.7 3.8 6.7 2.2 Iris-virginica
19. 7.7 2.6 6.9 2.3 Iris-virginica
20. 6.0 2.2 5.0 1.5 Iris-virginica
21. 6.9 3.2 5.7 2.3 Iris-virginica
22. 5.6 2.8 4.9 2.0 Iris-virginica
23. 7.7 2.8 6.7 2.0 Iris-virginica
24. 6.3 2.7 4.9 1.8 Iris-virginica
25. 6.7 3.3 5.7 2.1 Iris-virginica
26. 7.2 3.2 6.0 1.8 Iris-virginica
27. 6.2 2.8 4.8 1.8 Iris-virginica
28. 6.1 3.0 4.9 1.8 Iris-virginica
29. 6.4 2.8 5.6 2.1 Iris-virginica
30. 7.2 3.0 5.8 1.6 Iris-virginica
31. 7.4 2.8 6.1 1.9 Iris-virginica
32. 7.9 3.8 6.4 2.0 Iris-virginica
33. 6.4 2.8 5.6 2.2 Iris-virginica
34. 6.3 2.8 5.1 1.5 Iris-virginica

**A E C D B**

1. 6.1 2.6 5.6 1.4 Iris-virginica
2. 7.7 3.0 6.1 2.3 Iris-virginica
3. 6.3 3.4 5.6 2.4 Iris-virginica
4. 6.4 3.1 5.5 1.8 Iris-virginica
5. 6.0 3.0 4.8 1.8 Iris-virginica
6. 6.9 3.1 5.4 2.1 Iris-virginica
7. 6.7 3.1 5.6 2.4 Iris-virginica
8. 6.9 3.1 5.1 2.3 Iris-virginica
9. 5.8 2.7 5.1 1.9 Iris-virginica
10. 6.8 3.2 5.9 2.3 Iris-virginica
11. 6.7 3.3 5.7 2.5 Iris-virginica
12. 6.7 3.0 5.2 2.3 Iris-virginica
13. 6.3 2.5 5.0 1.9 Iris-virginica
14. 6.5 3.0 5.2 2.0 Iris-virginica
15. 6.2 3.4 5.4 2.3 Iris-virginica
16. 5.9 3.0 5.1 1.8 Iris-virginica

In [167]:

df5.get\_group("Iris-versicolor")

Out[167]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **E** | **C** | **D** | **B** |
| **49** 7.0 | 3.2 | 4.7 | 1.4 | Iris-versicolor |
| **50** 6.4 | 3.2 | 4.5 | 1.5 | Iris-versicolor |
| **51** 6.9 | 3.1 | 4.9 | 1.5 | Iris-versicolor |
| **52** 5.5 | 2.3 | 4.0 | 1.3 | Iris-versicolor |
| **53** 6.5 | 2.8 | 4.6 | 1.5 | Iris-versicolor |
| **54** 5.7 | 2.8 | 4.5 | 1.3 | Iris-versicolor |
| **55** 6.3 | 3.3 | 4.7 | 1.6 | Iris-versicolor |
| **56** 4.9 | 2.4 | 3.3 | 1.0 | Iris-versicolor |
| **57** 6.6 | 2.9 | 4.6 | 1.3 | Iris-versicolor |
| **58** 5.2 | 2.7 | 3.9 | 1.4 | Iris-versicolor |
| **59** 5.0 | 2.0 | 3.5 | 1.0 | Iris-versicolor |
| **60** 5.9 | 3.0 | 4.2 | 1.5 | Iris-versicolor |
| **61** 6.0 | 2.2 | 4.0 | 1.0 | Iris-versicolor |
| **62** 6.1 | 2.9 | 4.7 | 1.4 | Iris-versicolor |
| **63** 5.6 | 2.9 | 3.6 | 1.3 | Iris-versicolor |
| **64** 6.7 | 3.1 | 4.4 | 1.4 | Iris-versicolor |
| **65** 5.6 | 3.0 | 4.5 | 1.5 | Iris-versicolor |
| **66** 5.8 | 2.7 | 4.1 | 1.0 | Iris-versicolor |
| **67** 6.2 | 2.2 | 4.5 | 1.5 | Iris-versicolor |
| **68** 5.6 | 2.5 | 3.9 | 1.1 | Iris-versicolor |
| **69** 5.9 | 3.2 | 4.8 | 1.8 | Iris-versicolor |
| **70** 6.1 | 2.8 | 4.0 | 1.3 | Iris-versicolor |
| **71** 6.3 | 2.5 | 4.9 | 1.5 | Iris-versicolor |
| **72** 6.1 | 2.8 | 4.7 | 1.2 | Iris-versicolor |
| **73** 6.4 | 2.9 | 4.3 | 1.3 | Iris-versicolor |
| **74** 6.6 | 3.0 | 4.4 | 1.4 | Iris-versicolor |
| **75** 6.8 | 2.8 | 4.8 | 1.4 | Iris-versicolor |
| **76** 6.7 | 3.0 | 5.0 | 1.7 | Iris-versicolor |
| **77** 6.0 | 2.9 | 4.5 | 1.5 | Iris-versicolor |
| **78** 5.7 | 2.6 | 3.5 | 1.0 | Iris-versicolor |
| **79** 5.5 | 2.4 | 3.8 | 1.1 | Iris-versicolor |
| **80** 5.5 | 2.4 | 3.7 | 1.0 | Iris-versicolor |
| **81** 5.8 | 2.7 | 3.9 | 1.2 | Iris-versicolor |
| **82** 6.0 | 2.7 | 5.1 | 1.6 | Iris-versicolor |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **E** | **C** | **D** | **B** |
| **83** 5.4 | 3.0 | 4.5 | 1.5 | Iris-versicolor |
| **84** 6.0 | 3.4 | 4.5 | 1.6 | Iris-versicolor |
| **85** 6.7 | 3.1 | 4.7 | 1.5 | Iris-versicolor |
| **86** 6.3 | 2.3 | 4.4 | 1.3 | Iris-versicolor |
| **87** 5.6 | 3.0 | 4.1 | 1.3 | Iris-versicolor |
| **88** 5.5 | 2.5 | 4.0 | 1.3 | Iris-versicolor |
| **89** 5.5 | 2.6 | 4.4 | 1.2 | Iris-versicolor |
| **90** 6.1 | 3.0 | 4.6 | 1.4 | Iris-versicolor |
| **91** 5.8 | 2.6 | 4.0 | 1.2 | Iris-versicolor |
| **92** 5.0 | 2.3 | 3.3 | 1.0 | Iris-versicolor |
| **93** 5.6 | 2.7 | 4.2 | 1.3 | Iris-versicolor |
| **94** 5.7 | 3.0 | 4.2 | 1.2 | Iris-versicolor |
| **95** 5.7 | 2.9 | 4.2 | 1.3 | Iris-versicolor |
| **96** 6.2 | 2.9 | 4.3 | 1.3 | Iris-versicolor |
| **97** 5.1 | 2.5 | 3.0 | 1.1 | Iris-versicolor |
| **98** 5.7 | 2.8 | 4.1 | 1.3 | Iris-versicolor |

we have catogorised data according to Iris-setosa,Iris-versicolor and Iris-virginica,now display mean

,percentile and standard deviation of these species.

In [168]:

df5.mean() *#mean of species*

Out[168]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **B** | **A** | **E** | **C** | **D** |
| **Iris-setosa** | 5.004082 | 3.416327 | 1.465306 | 0.244898 |
| **Iris-versicolor** | 5.936000 | 2.770000 | 4.260000 | 1.326000 |
| **Iris-virginica** | 6.588000 | 2.974000 | 5.552000 | 2.026000 |

In [169]:

df5.std() *#standard deviation of species*

Out[169]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **B** | **A** | **E** | **C** | **D** |
| **Iris-setosa** | 0.355879 | 0.384787 | 0.175061 | 0.108130 |
| **Iris-versicolor** | 0.516171 | 0.313798 | 0.469911 | 0.197753 |
| **Iris-virginica** | 0.635880 | 0.322497 | 0.551895 | 0.274650 |

percentile :-Percentiles are descriptive statistics that tell us about the distribution of the values. The nth percentile value denotes that n% of the values in the given sequence are smaller than this value.

In [181]:

a**=**np.percentile(df4['A'],50)

In [182]:

a

Out[182]:

5.8