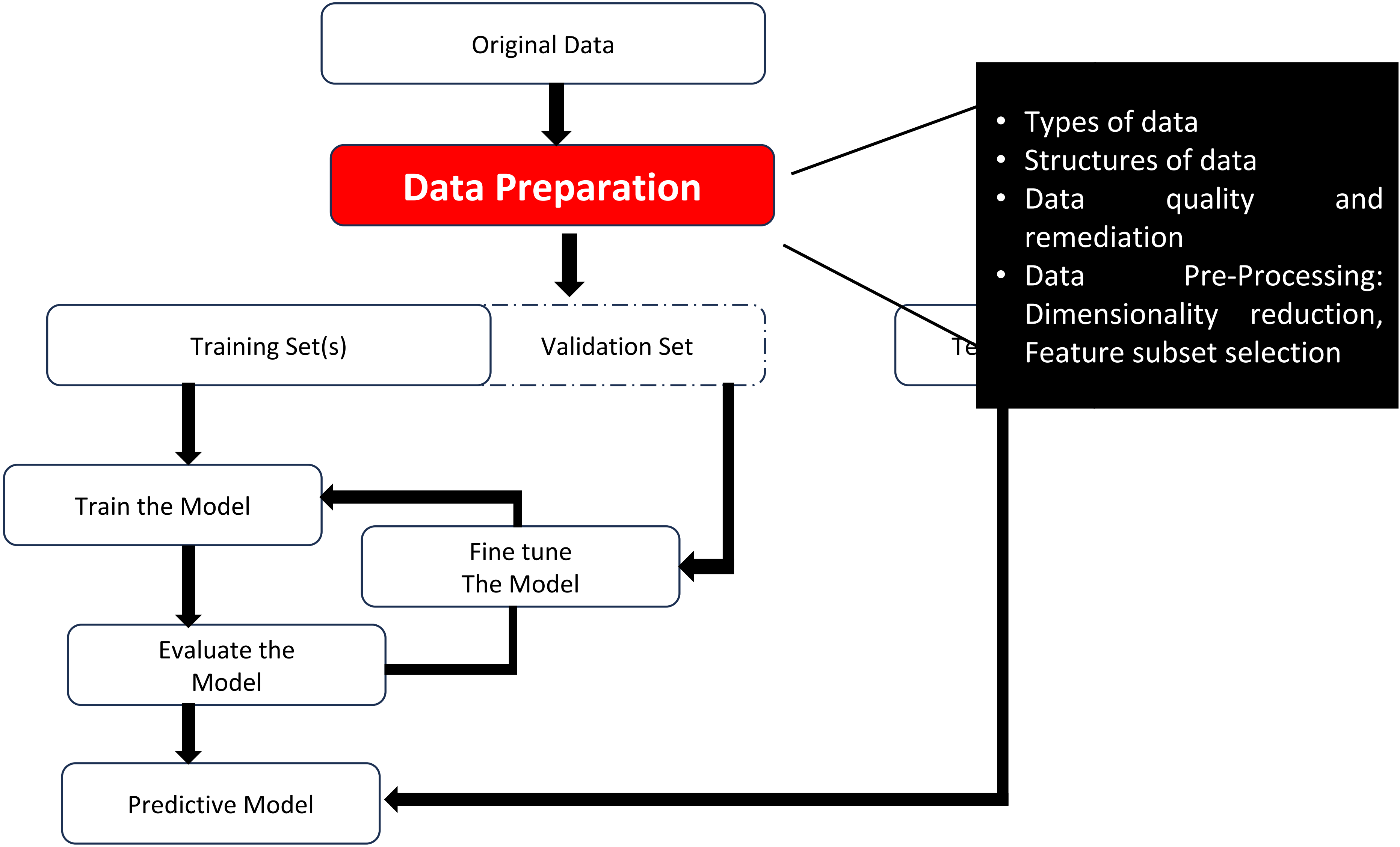
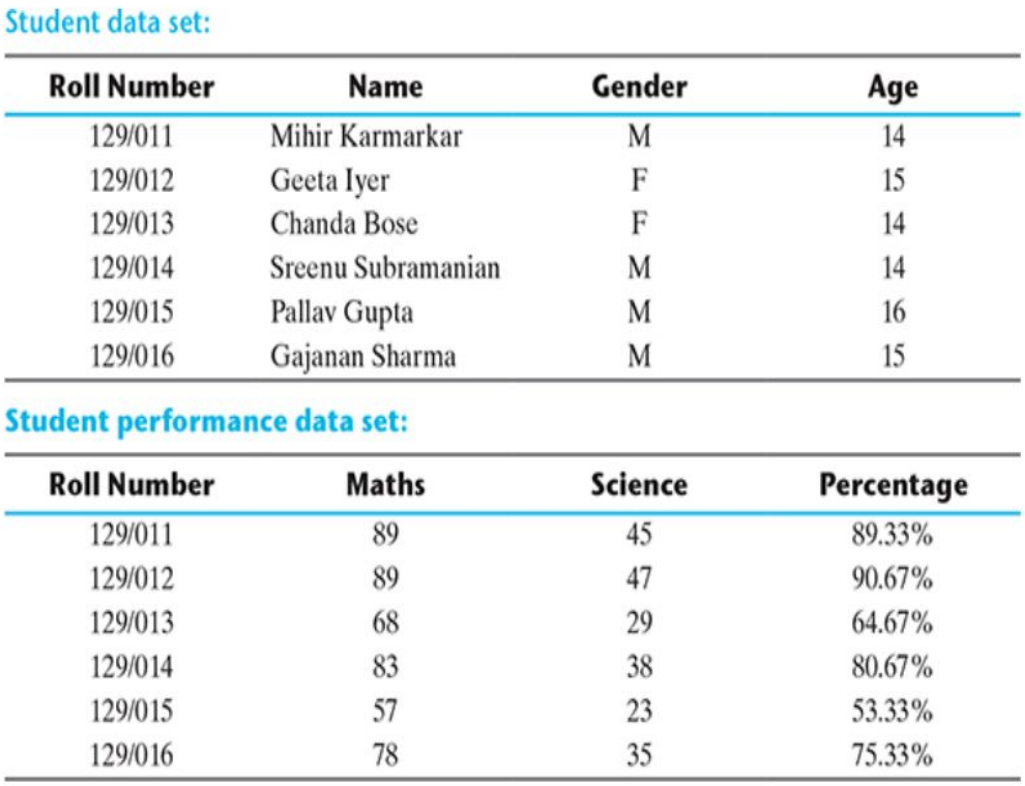
Preparing Data: Machine Learning

Prof. K.R. Makvana



- Types of data
- Structures of data
- Data quality and remediation
- Data Pre-Processing: Dimensionality reduction, Feature subset selection

BASIC TYPES OF DATA IN MACHINE LEARNING



Student data set:

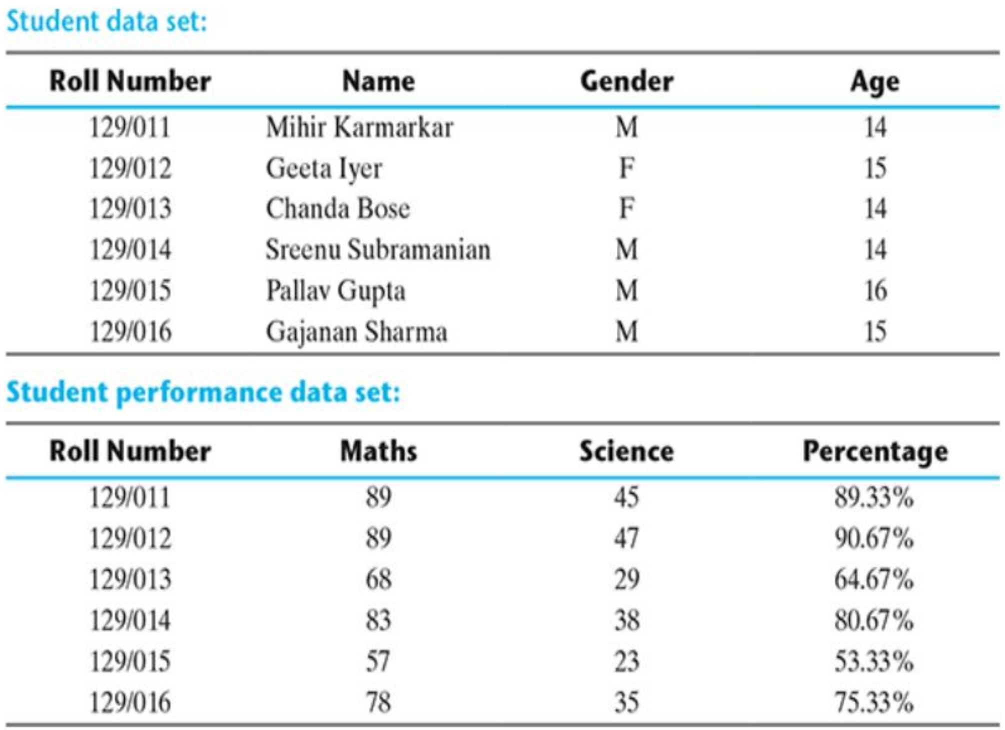
Roll Number	Name	Gender	Age
129/011	Mihir Karmarkar	M	14
129/012	Geeta Iyer	F	15
129/013	Chanda Bose	F	14
129/014	Sreenu Subramanian	M	14
129/015	Pallav Gupta	M	16
129/016	Gajanan Sharma	M	15

Student performance data set:

Roll Number	Maths	Science	Percentage
129/011	89	45	89.33%
129/012	89	47	90.67%
129/013	68	29	64.67%
129/014	83	38	80.67%
129/015	57	23	53.33%
129/016	78	35	75.33%

- Data can broadly be divided into following two types:
  - Qualitative data
  - Quantitative data
- Information that cannot be measured using some scale of measurement are called qualitative data. Eg. Gender, Name, Roll Number.

BASIC TYPES OF DATA IN MACHINE LEARNING



Student data set:

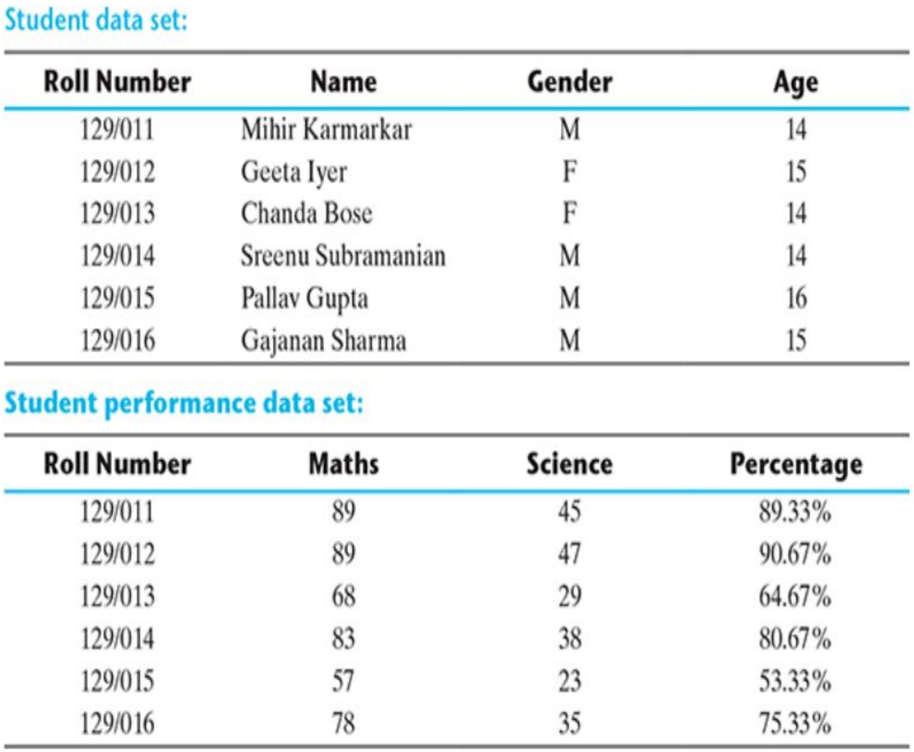
Roll Number	Name	Gender	Age
129/011	Mihir Karmarkar	M	14
129/012	Geeta Iyer	F	15
129/013	Chanda Bose	F	14
129/014	Sreenu Subramanian	M	14
129/015	Pallav Gupta	M	16
129/016	Gajanan Sharma	M	15

Student performance data set:

Roll Number	Maths	Science	Percentage
129/011	89	45	89.33%
129/012	89	47	90.67%
129/013	68	29	64.67%
129/014	83	38	80.67%
129/015	57	23	53.33%
129/016	78	35	75.33%

- Qualitative data is also called categorical data. Which is further subdivided into;
  - Nominal data
  - Ordinal data
- Nominal data is one which has no numeric value, but a named value
- Nominal values cannot be quantified. Examples of nominal data are;
  - Blood group: A, B, O, AB, etc
  - Nationality: Indian, American, British, etc.
  - Gender: Male, Female, Other

BASIC TYPES OF DATA IN MACHINE LEARNING



Student data set:

Roll Number	Name	Gender	Age
129/011	Mihir Karmarkar	M	14
129/012	Geeta Iyer	F	15
129/013	Chanda Bose	F	14
129/014	Sreenu Subramanian	M	14
129/015	Pallav Gupta	M	16
129/016	Gajanan Sharma	M	15

Student performance data set:

Roll Number	Maths	Science	Percentage
129/011	89	45	89.33%
129/012	89	47	90.67%
129/013	68	29	64.67%
129/014	83	38	80.67%
129/015	57	23	53.33%
129/016	78	35	75.33%

- Qualitative data is also called categorical data. Which is further subdivided into;
  - Nominal data
  - Ordinal data
- Operations allowed in Nominal data is **Mode** only.
- **Ordinal data,** in addition to possessing the properties of nominal data, can also be naturally ordered.
- Examples;
  - Customer satisfaction: ‘Very Happy’, ‘Happy’, ‘Unhappy’, etc.
  - Grades: A, B, C, etc.
  - Hardness of Metal: ‘Very Hard’, ‘Hard’, ‘Soft’, etc.

Qualitative data ⍰ Ordinal Data

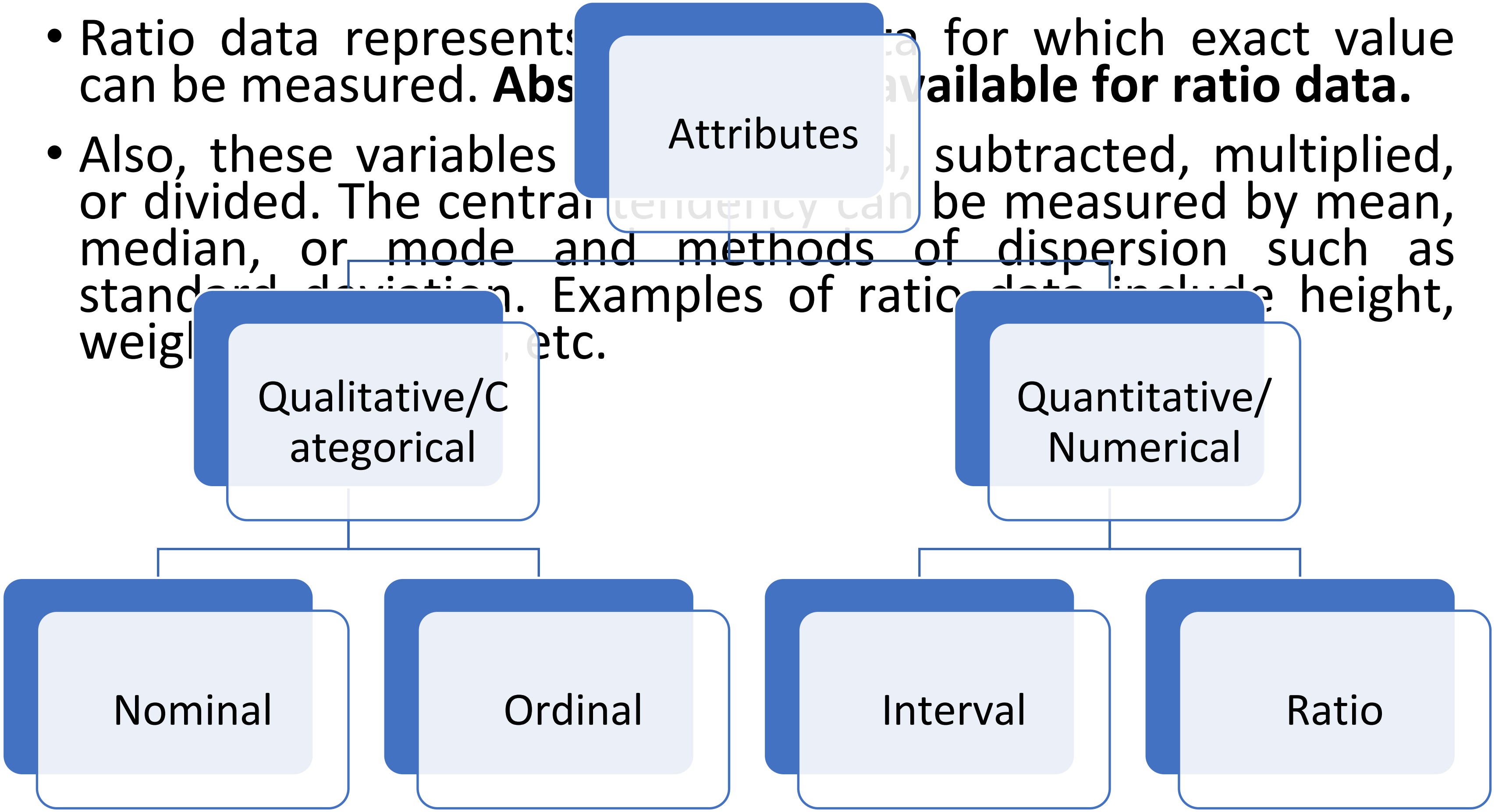
- Like nominal data, basic counting is possible for ordinal data. Hence, the mode can be identified. Since ordering is possible in case of ordinal data, median, and quartiles can be identified in addition. Mean can still not be calculated.

Quantitative data

- **Quantitative data** (numeric data) relates to information about the quantity of an object – hence it can be measured.
  - Temperature , age, etc.
- There are two types of quantitative data:
  - Interval data
  - Ratio data
- Interval data is numeric data for which **not only the order is known, but the exact difference between values is also known**. An ideal example of interval data is Celsius temperature.
- For interval data, mathematical operations such as addition and subtraction are possible. For that reason, for interval data, the central tendency can be measured by mean, median, or mode. Standard deviation can also be calculated.
- However, interval data do not have something called a ‘true zero’ value. For example, there is nothing called ‘0 temperature’ or ‘no temperature’.

Quantitative data Ratio Data

- Ratio data represents ta for which exact value can be measured. **Abs vailable for ratio data.**
- Also, these variables d, subtracted, multiplied, or divided. The central tendency can be measured by mean, median, or mode and methods of dispersion such as standard deviation. Examples of ratio data include height, weig , etc.



Discrete Vs Continuous data

- Discrete attributes: Countable finite or Countably infinite values
  - roll number, street number, pin code
  - count, rank of students
  - binary attribute include: male/ female, positive/negative, yes/no, etc
- Continuous attributes: Any possible real numbers
  - length, height, weight, price, etc.
- http://archive.ics.uci.edu/datasets

Exploring numerical data

- There are two most effective mathematical plots to explore numerical data;
  - box plot and histogram
- But before exploration of data it is essential to understand statistical computation of numerical data i.e. mean, median, standard deviation, variance, etc.
-

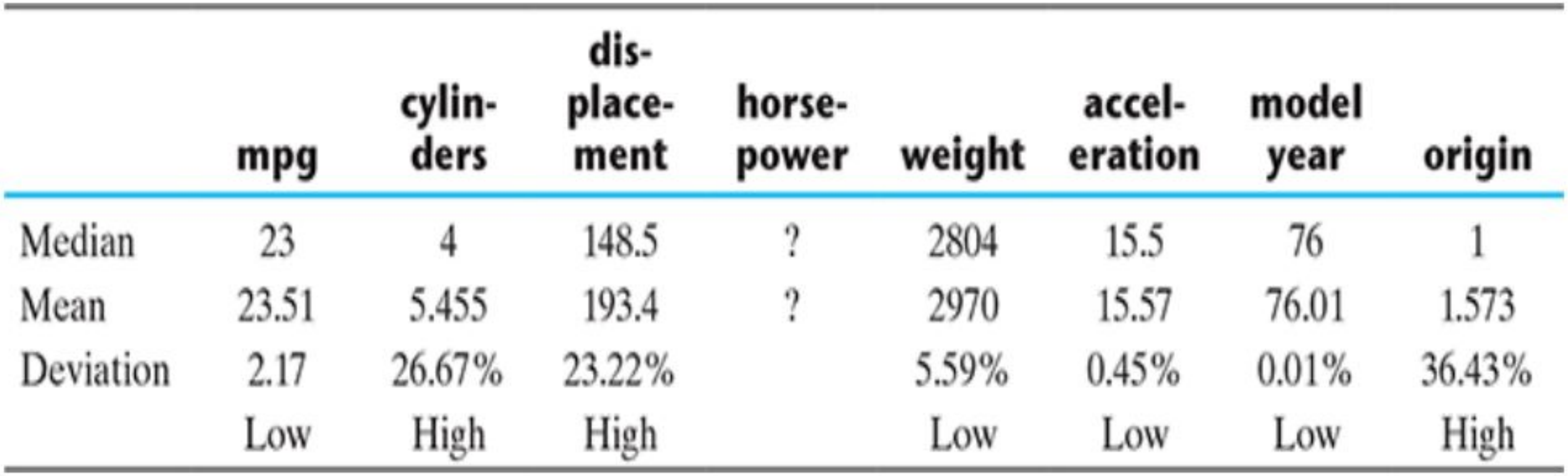
Mean and Median

- The **mean** gives the arithmetic mean of the input values. It is the sum of elements divided by the total number of elements.
  - $\mu = \frac{\sum_{k=1}^{n} x_k}{n}$
- Median: The median gives the middle values in the given data. In the case of the median, we have two different formulas. If we have an odd number of terms in the data set we use the following formula

What is the purpose of mean and median?

Mean and median are impacted differently by data values appearing at **the beginning or at the end of the range**. Mean being calculated from the cumulative sum of data values, is **impacted if too many data elements are having values closer to the far end of the range**, i.e. close to the maximum or minimum values. It is especially **sensitive to outliers**, i.e. the values which are unusually high or low, compared to the other values. Mean is likely to get shifted drastically even due to the presence of a small number of outliers. If we observe that for certain attributes the deviation between values of mean and median are quite high, we should investigate those attributes further and try to find out the root cause along with the need for remediation

Mean vs. Median for Auto MPG dataset



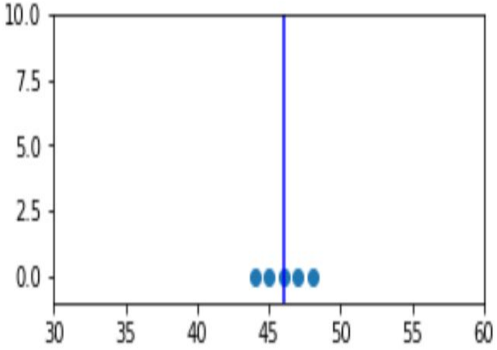
	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
Median	23	4	148.5	?	2804	15.5	76	1
Mean	23.51	5.455	193.4	?	2970	15.57	76.01	1.573
Deviation	2.17	26.67%	23.22%		5.59%	0.45%	0.01%	36.43%
	Low	High	High		Low	Low	Low	High

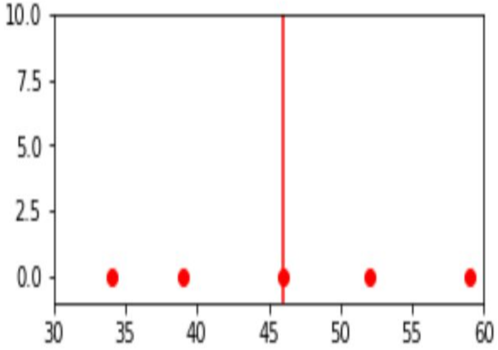
Granular view of the data spread

- Mean and median represent central tendency of data and from the deviation between mean and median, we can identify how data is dispersed.
- However, this manual review is not efficient for huge datasets available, so we will take granular view of data sets in the form of;
    - **Data dispersion methods**
        - **Variance and standard deviation**
    - **Position of different data value**
        - **Quartile**
- **Let us try to understand data dispersion using simple example**

Measuring Data dispersion

- Attribute 1 values : 44, 46, 48, 45, and 47
- Attribute 2 values : 34, 46, 59, 39, and 52
- Mean and median of both attributes are 46





To measure the extent of dispersion of a data, or to find out how much the different values of a data are spread out, the variance/std of the data is measured

Measuring Data dispersion

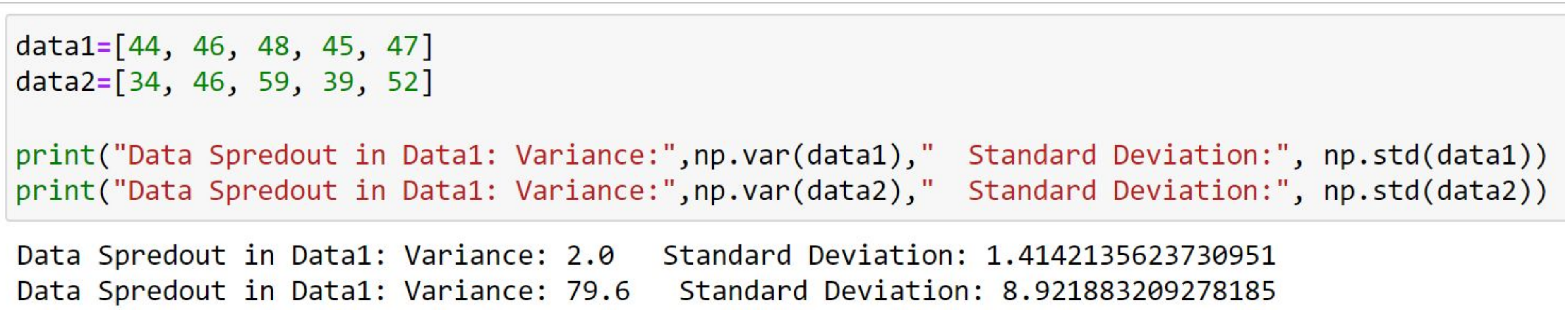
- **Variance:** The variance is defined as the total of the square distances from the mean (µ) of each term in the distribution, divided by the number of distribution terms (N).

$$\text{Variance}(\sigma^2) = \frac{\sum(x_i - \mu)^2}{N}$$

- **Standard Deviation:** By evaluating the deviation of each data point relative to the mean, the standard deviation is calculated as the **square root of variance**.

$$\text{Standard deviation}(\sigma) = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

Larger value of variance or standard deviation indicates more dispersion in the data and vice versa.

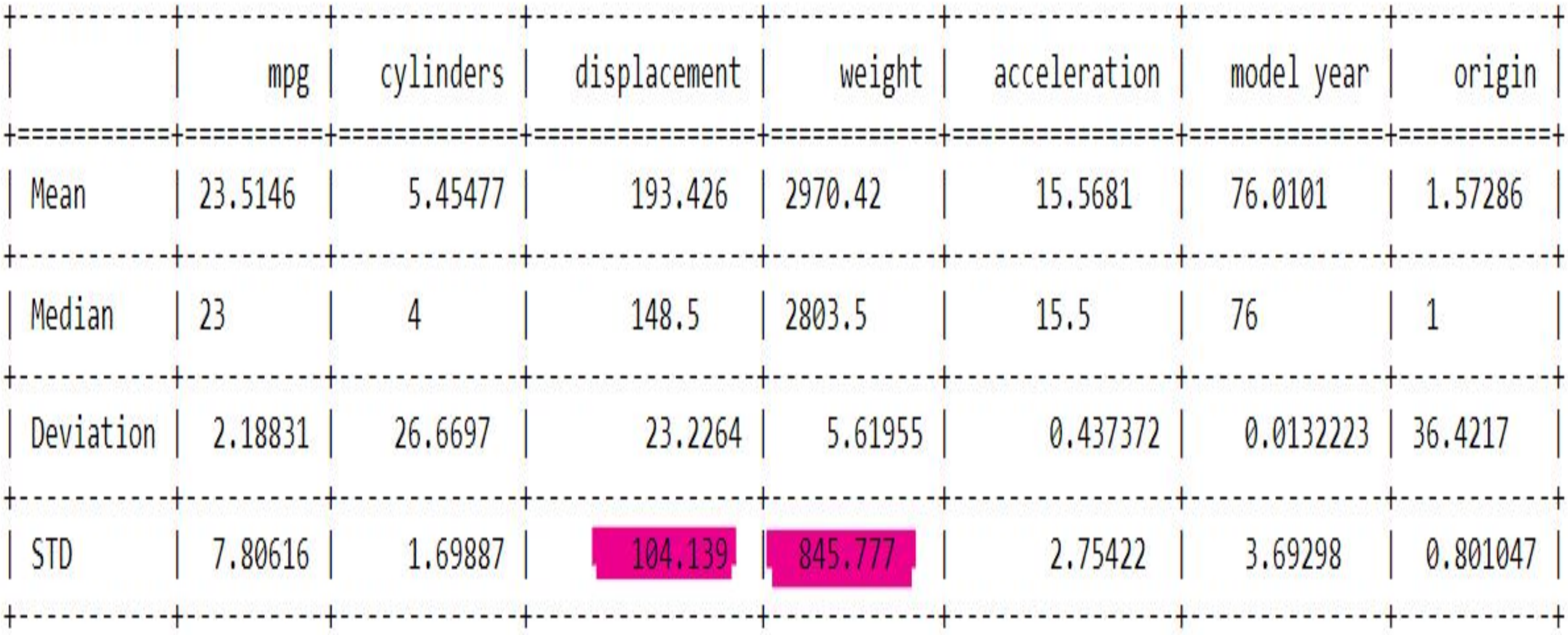


```
data1=[44, 46, 48, 45, 47]
data2=[34, 46, 59, 39, 52]

print("Data Spredout in Data1: Variance:",np.var(data1),"  Standard Deviation:", np.std(data1))
print("Data Spredout in Data1: Variance:",np.var(data2),"  Standard Deviation:", np.std(data2))
```

```
Data Spredout in Data1: Variance: 2.0   Standard Deviation: 1.4142135623730951
Data Spredout in Data1: Variance: 79.6   Standard Deviation: 8.921883209278185
```

So it is quite clear from the measure that attribute 1 values are quite concentrated around the mean while attribute 2 values are extremely spread out. Since this data was small, a visual inspection and understanding were possible and that matches with the measured value.

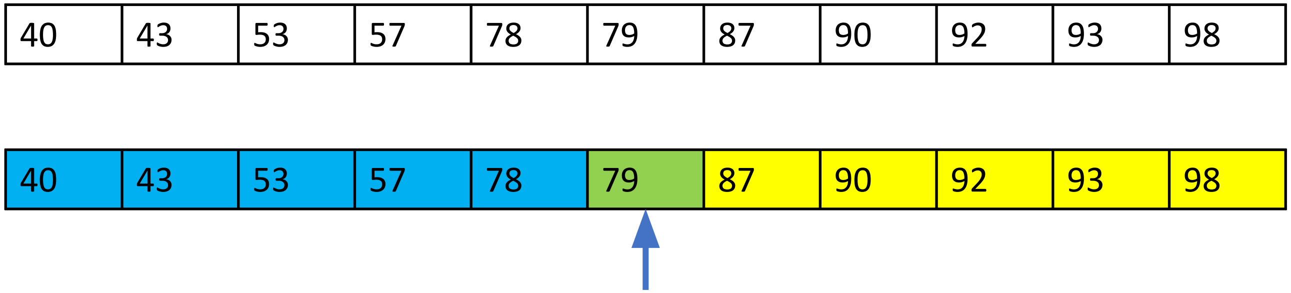
Looking at auto MPG dataset



```
+-----------+---------+-------------+----------------+-----------+----------------+--------------+-----------+
|           |     mpg |   cylinders |   displacement |    weight |   acceleration |   model year |    origin |
+===========+=========+=============+================+===========+================+==============+===========+
| Mean      | 23.5146 |     5.45477 |        193.426 | 2970.42   |        15.5681 | 76.0101      | 1.57286   |
+-----------+---------+-------------+----------------+-----------+----------------+--------------+-----------+
| Median    | 23      |     4       |        148.5   | 2803.5    |        15.5    | 76           | 1         |
+-----------+---------+-------------+----------------+-----------+----------------+--------------+-----------+
| Deviation | 2.18831 |   26.6697   |        23.2264 |   5.61955 |       0.437372 |    0.0132223 | 36.4217   |
+-----------+---------+-------------+----------------+-----------+----------------+--------------+-----------+
| STD       | 7.80616 |     1.69887 |        104.139 |   845.777 |        2.75422 |      3.69298 |  0.801047 |
+-----------+---------+-------------+----------------+-----------+----------------+--------------+-----------+
```

Measuring data value position: IQR

- a = [40,43,53,57,78,79,87,90,92,93,98]
- b = [5,22,39,75,79,85,90,91,93,93,94,95] (Do by self)

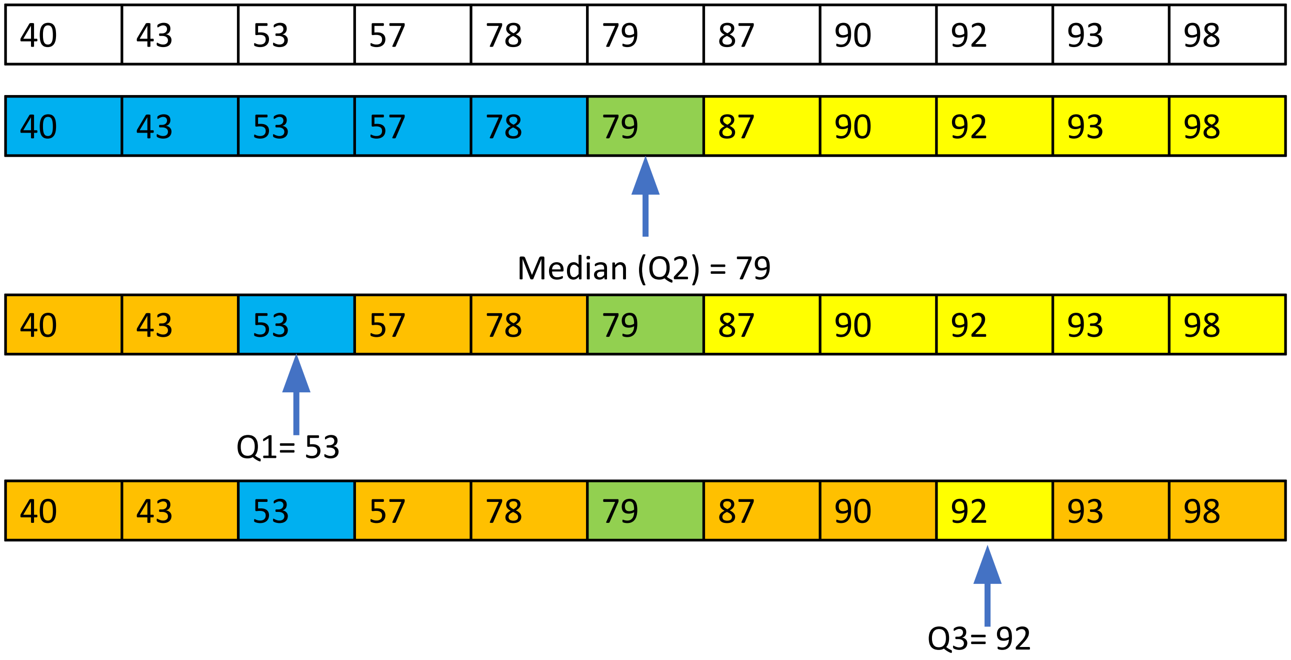


40	43	53	57	78	79	87	90	92	93	98

40	43	53	57	78	79	87	90	92	93	98

Median (Q2) = 79

Measuring data value position: IQR



40	43	53	57	78	79	87	90	92	93	98

40	43	53	57	78	79	87	90	92	93	98

Median (Q2) = 79

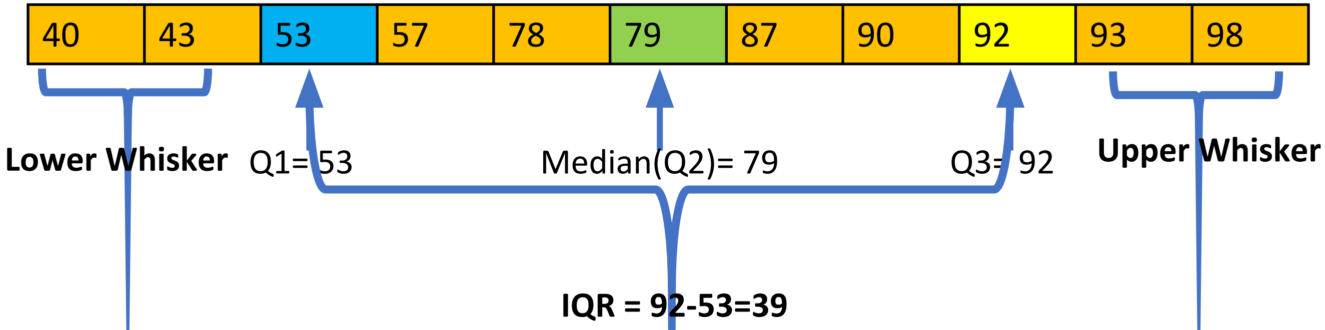
40	43	53	57	78	79	87	90	92	93	98

Q1= 53

40	43	53	57	78	79	87	90	92	93	98

Q3= 92

Measuring data value position: IQR



40	43	53	57	78	79	87	90	92	93	98

Lower Whisker

Q1= 53

Median(Q2)= 79

Q3= 92

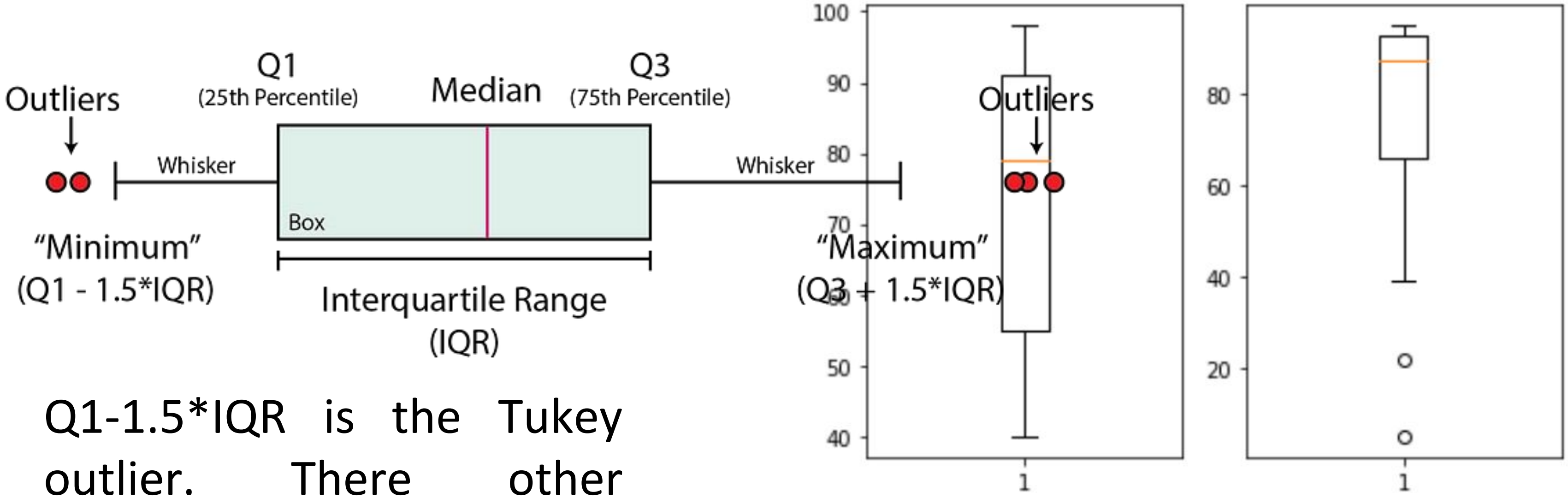
Upper Whisker

IQR = 92-53=39

A **box plot** is an extremely effective mechanism to get a one-shot view and understand the nature of the data i.e. spread as well as **outliers**.

Boxplot

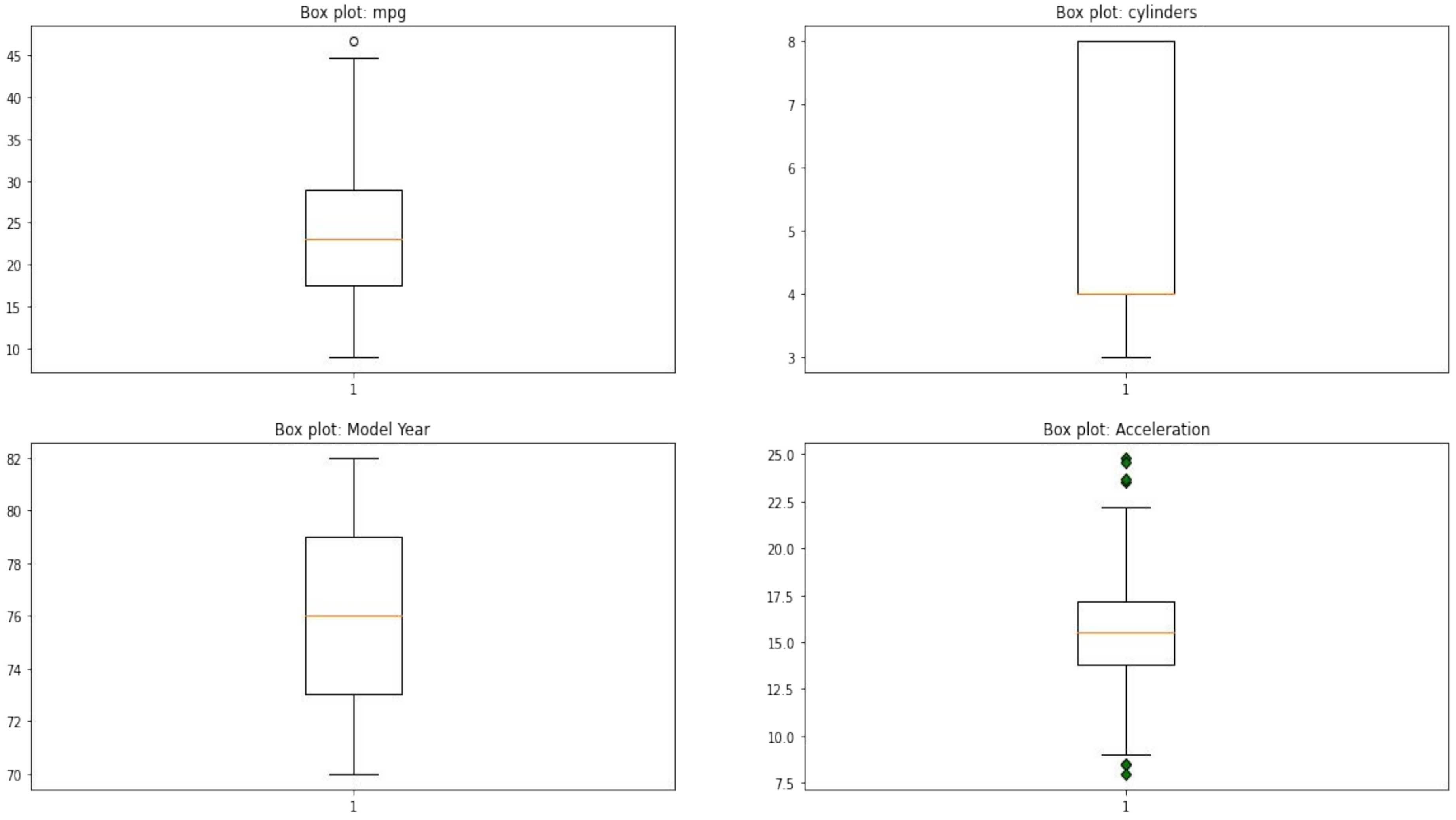
a = [40,43,53,57,78,79,87,90,92,93,98]
b = [5,22,39,75,79,85,90,91,93,93,94,95]



Q1-1.5*IQR is the Tukey outlier. There other methods for identification of outliers.

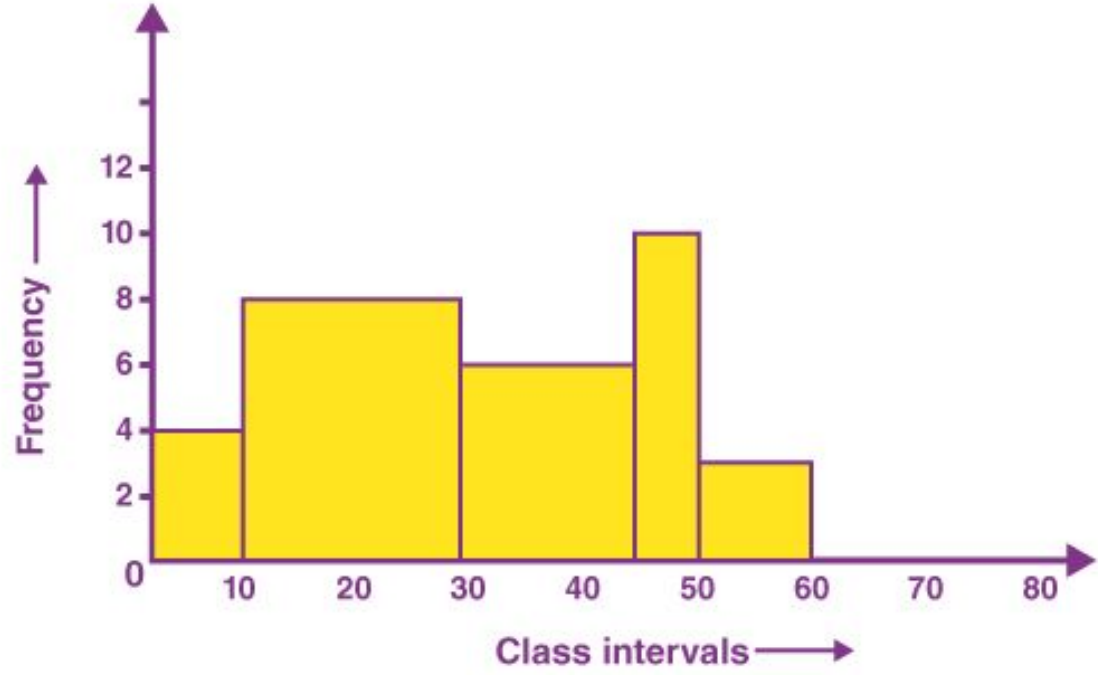
Analyze the two boxplot and write conclusions

Box Plot of Auto MPG data set



Histogram

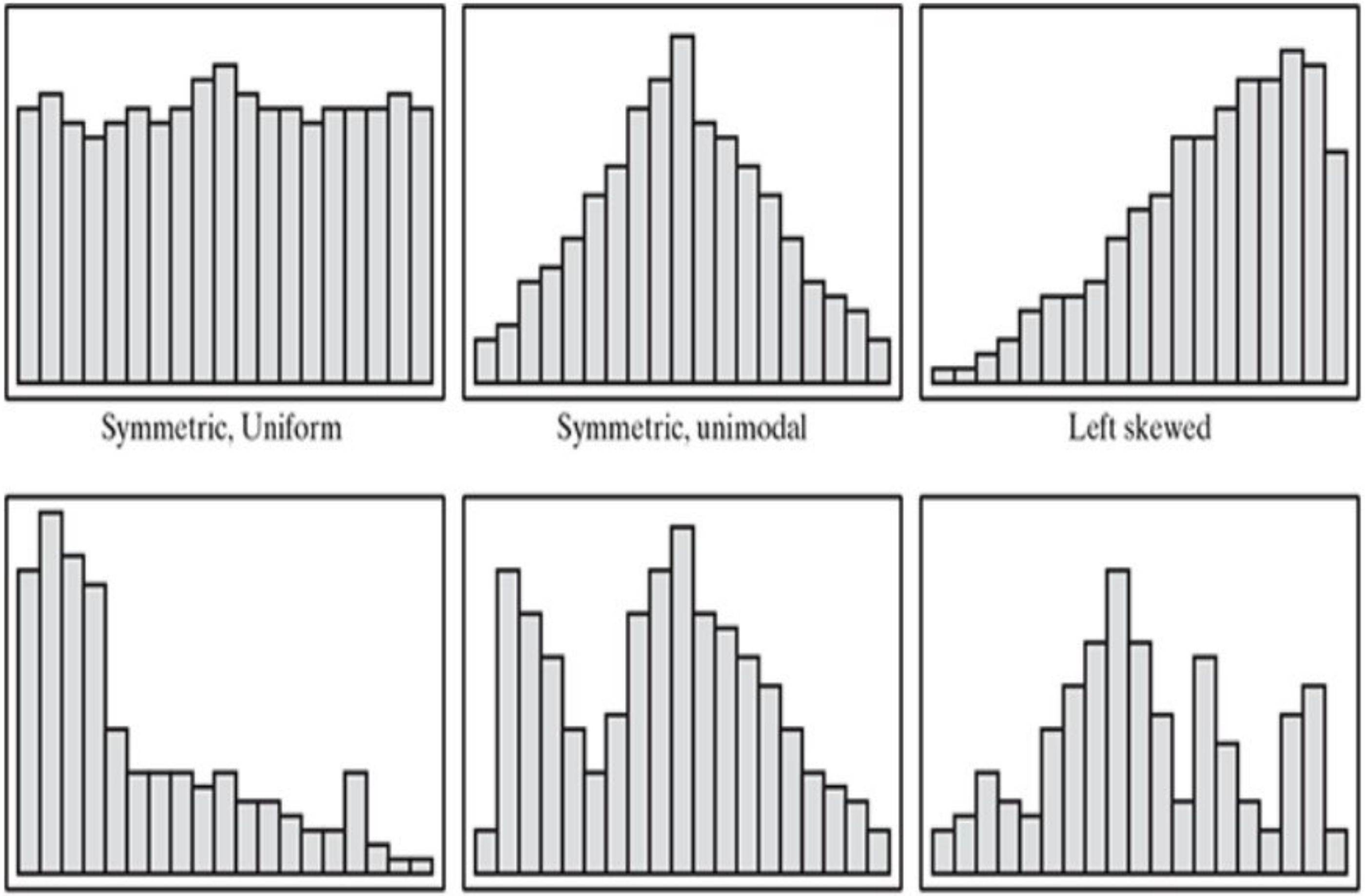
- A histogram divides the variable into bins, counts the data points in each bin, and shows the bins on the x-axis and the counts on the y-axis



Types of Histogram

- The histogram can be classified into different types based on the frequency distribution of the data.
  - Uniform histogram
  - Symmetric histogram
  - Bimodal histogram
  - Probability histogram

Types of Histogram



Symmetric, Uniform

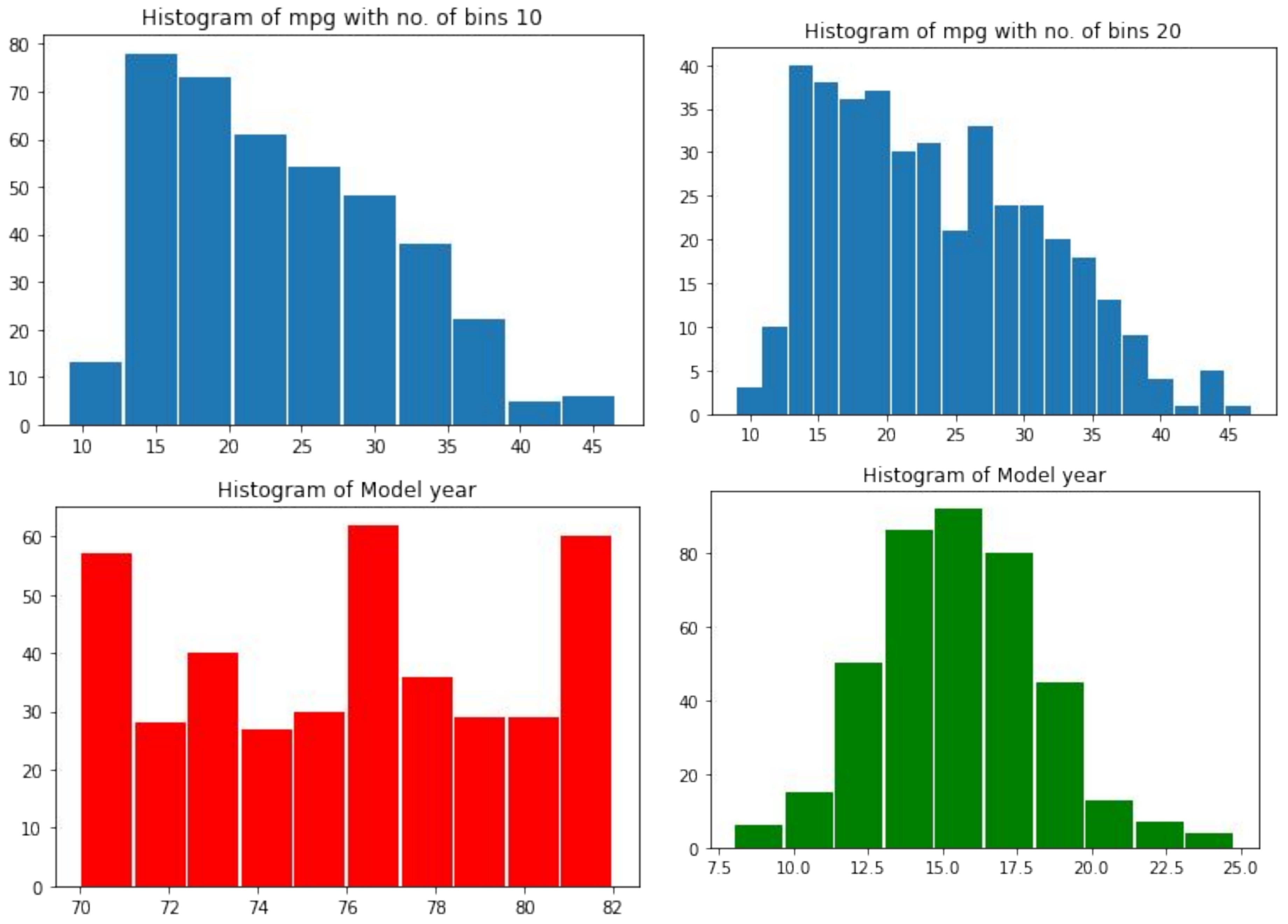
Symmetric, unimodal

Left skewed

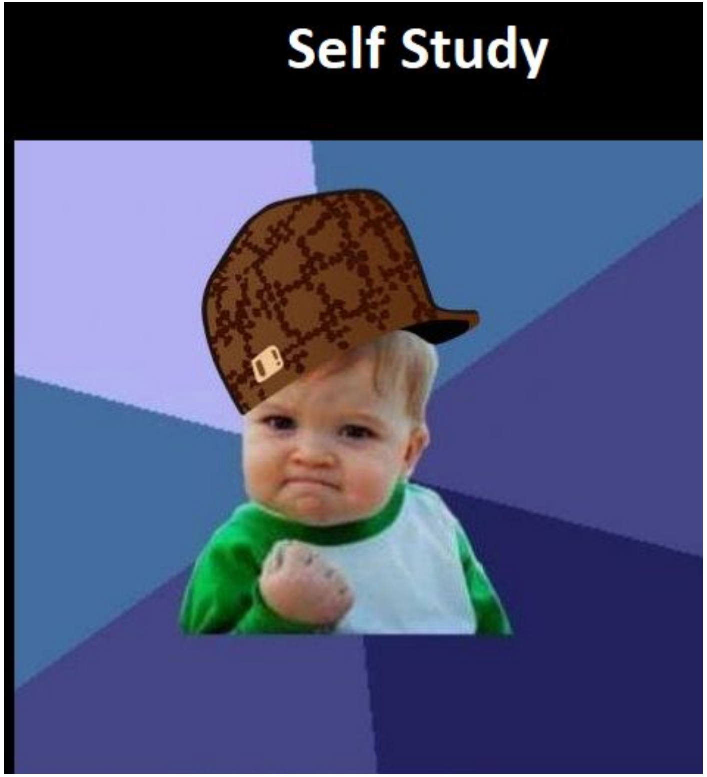
Right skewed

Bimodal

Multimodal


Histogram of mpg with no. of bins 10
0
10
20
30
40
50
60
70
80
10
15
20
25
30
35
40
45
Histogram of mpg with no. of bins 20
0
5
10
15
20
25
30
35
40
10
15
20
25
30
35
40
45
Histogram of Model year
0
10
20
30
40
50
60
70
72
74
76
78
80
82
Histogram of Model year
0
20
40
60
80
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0

Self Study When Histograms Fail



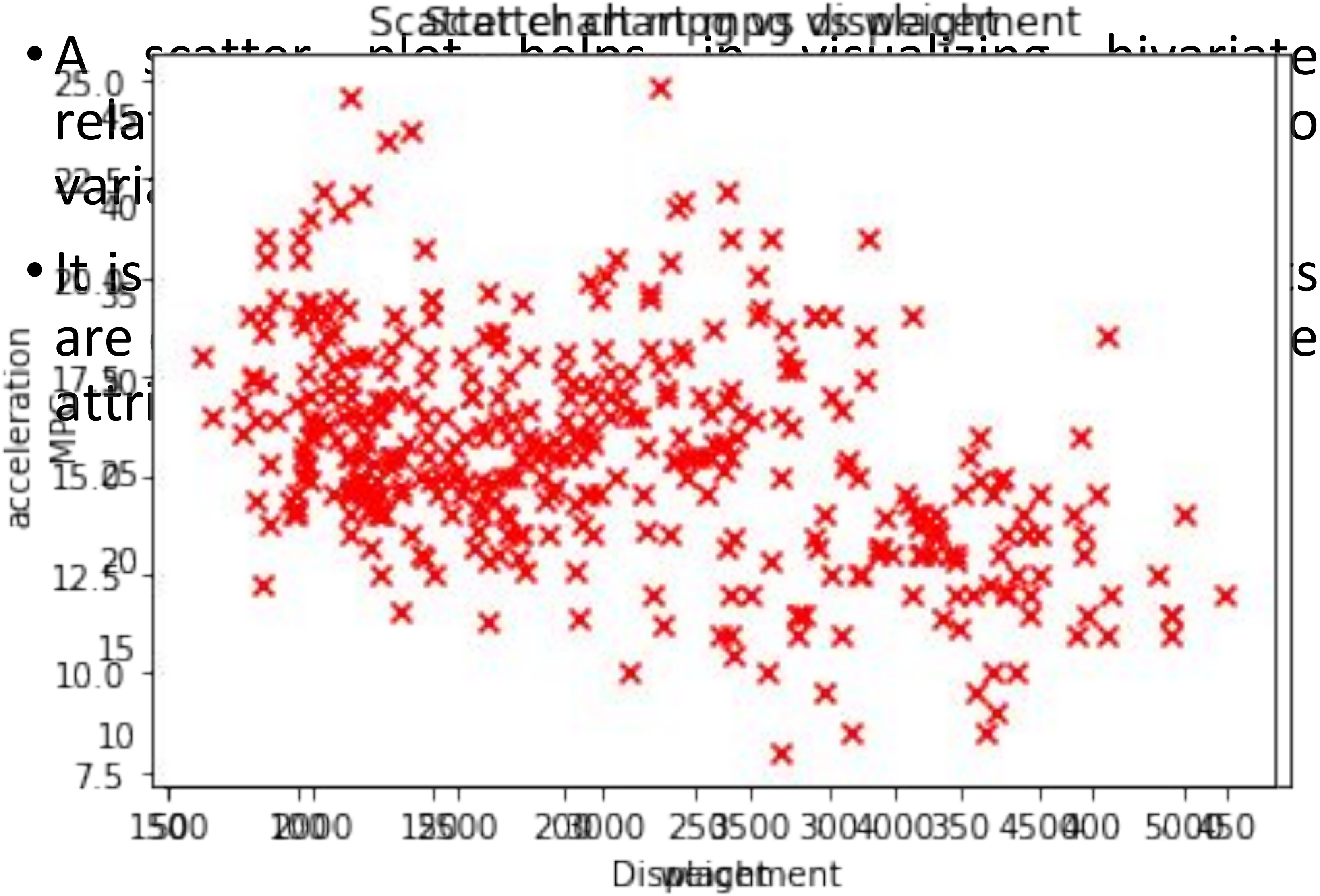
https://towardsdatascience.com/histograms-and-density-plots-in-python-f6bda88f5ac0

Data Exploration: Multiple attributes

- Till now we have been exploring single attributes in isolation. One more important angle of data exploration is to explore relationship between attributes.
  - Scatter plot
  - Two-way cross-tabulations

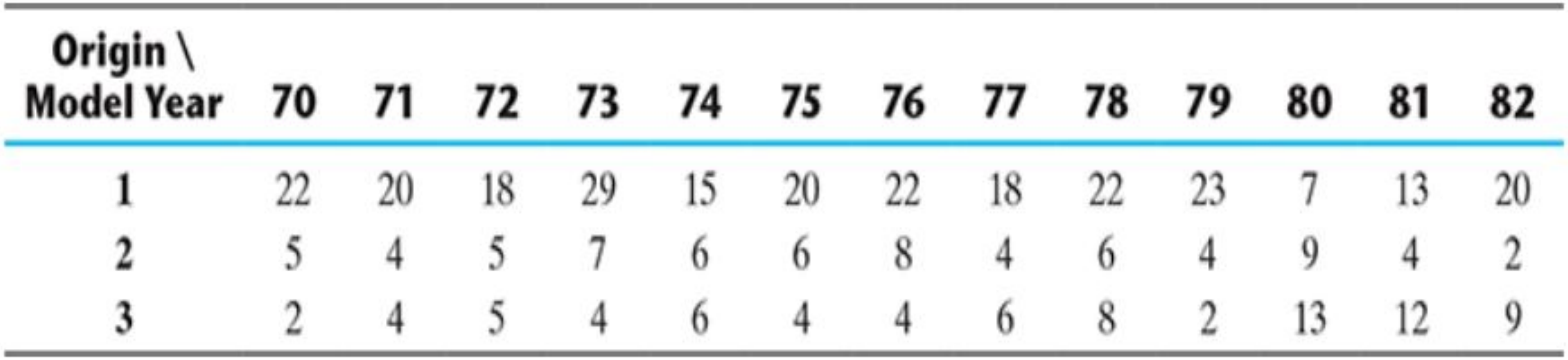
Scatter plot

- A scatter plot helps in visualizing bivariate rela… o varia…
- It is … s are … e attri…

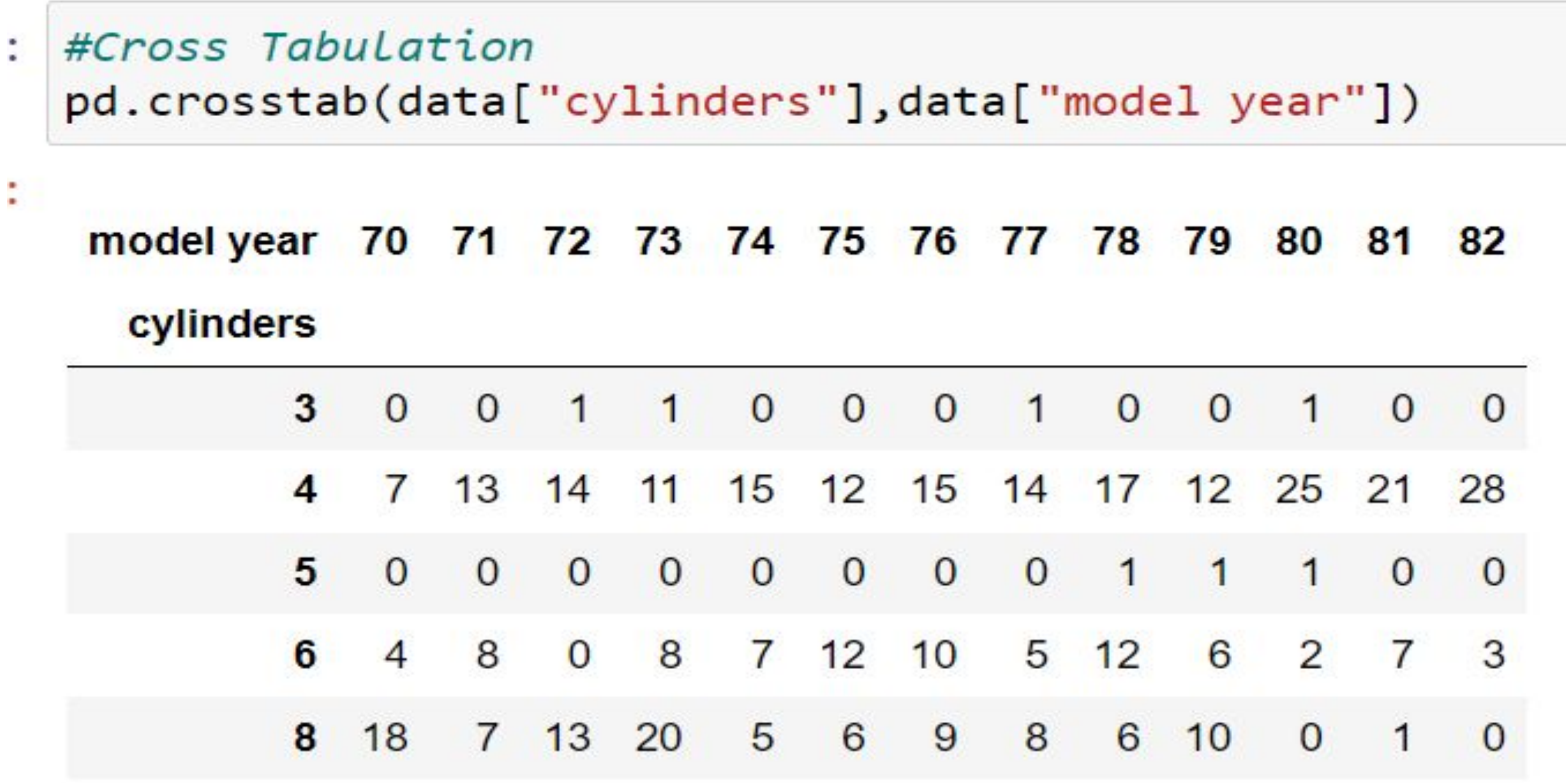


Two-way cross-tabulations

- A cross-tabulation (Frequency table/Contingency table) is simple but effective way to inspect relationship between two or more ***categorical or discrete*** variables

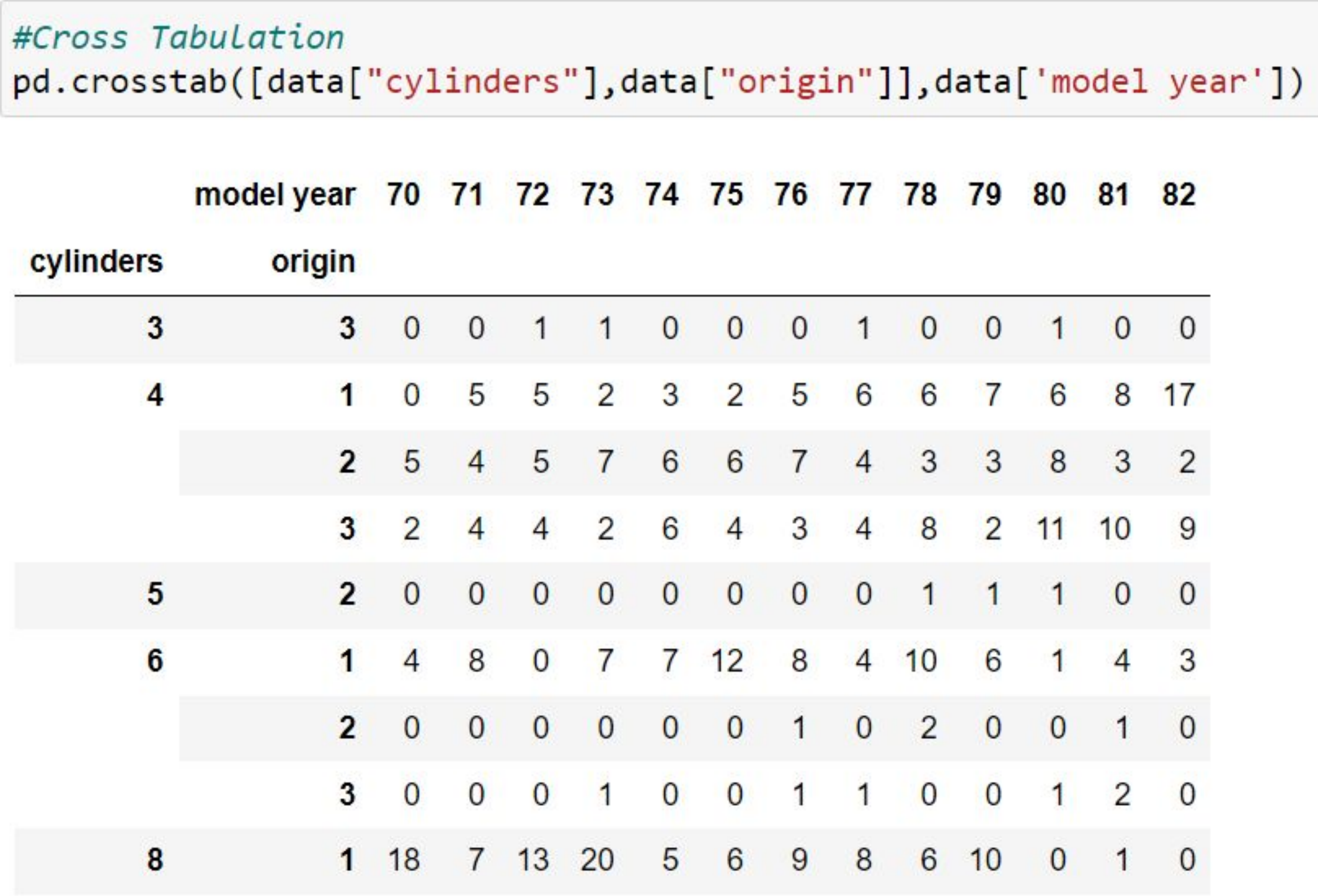


Origin \ Model Year	70	71	72	73	74	75	76	77	78	79	80	81	82
1	22	20	18	29	15	20	22	18	22	23	7	13	20
2	5	4	5	7	6	6	8	4	6	4	9	4	2
3	2	4	5	4	6	4	4	6	8	2	13	12	9



```python
#Cross Tabulation
pd.crosstab(data["cylinders"],data["model year"])
```

```
model year  70  71  72  73  74  75  76  77  78  79  80  81  82
 cylinders
         3   0   0   1   1   0   0   0   1   0   0   1   0   0
         4   7  13  14  11  15  12  15  14  17  12  25  21  28
         5   0   0   0   0   0   0   0   0   1   1   1   0   0
         6   4   8   0   8   7  12  10   5  12   6   2   7   3
         8  18   7  13  20   5   6   9   8   6  10   0   1   0
```



```python
#Cross Tabulation
pd.crosstab([data["cylinders"],data["origin"]],data['model year'])
```

	model year	70	71	72	73	74	75	76	77	78	79	80	81	82
cylinders	**origin**													
3	**3**	0	0	1	1	0	0	0	1	0	0	1	0	0
4	**1**	0	5	5	2	3	2	5	6	6	7	6	8	17
	2	5	4	5	7	6	6	7	4	3	3	8	3	2
	3	2	4	4	2	6	4	3	4	8	2	11	10	9
5	**2**	0	0	0	0	0	0	0	0	1	1	1	0	0
6	**1**	4	8	0	7	7	12	8	4	10	6	1	4	3
	2	0	0	0	0	0	0	1	0	2	0	0	1	0
	3	0	0	0	1	0	0	1	1	0	0	1	2	0
8	**1**	18	7	13	20	5	6	9	8	6	10	0	1	0

DATA QUALITY AND REMEDIATION

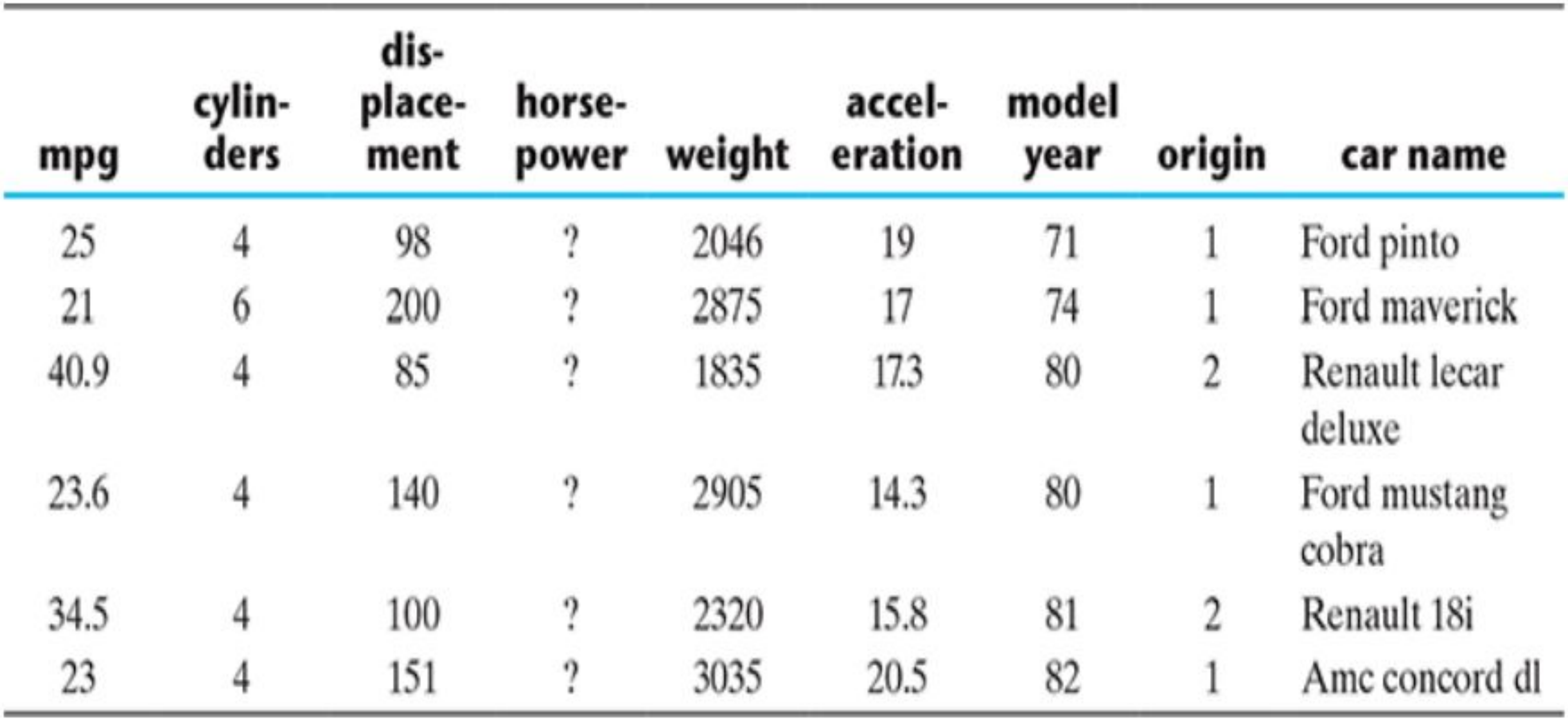
Data quality

- Quality of dataset can be affected due to;
    - Wrong sample set selection
    - Missing values / outliers
- Data Remediation
    - Handling outliers
    - Handling missing values

Handling outliers

- **Remove outliers:** If the number of records which are outliers is not many, a simple approach may be to remove them.
- **Imputation:** One other way is to impute the value with mean or median or mode. The value of the most similar data element may also be used for imputation.
- **Capping:** For values that lie outside the 1.5|×| IQR limits, we can cap them by replacing those observations below the lower limit with the value of 5th percentile and those that lie above the upper limit, with the value of 95th percentile.

Data Cleaning



mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
25	4	98	?	2046	19	71	1	Ford pinto
21	6	200	?	2875	17	74	1	Ford maverick
40.9	4	85	?	1835	17.3	80	2	Renault lecar deluxe
23.6	4	140	?	2905	14.3	80	1	Ford mustang cobra
34.5	4	100	?	2320	15.8	81	2	Renault 18i
23	4	151	?	3035	20.5	82	1	Amc concord dl

[illegible]

Handling Missing Values

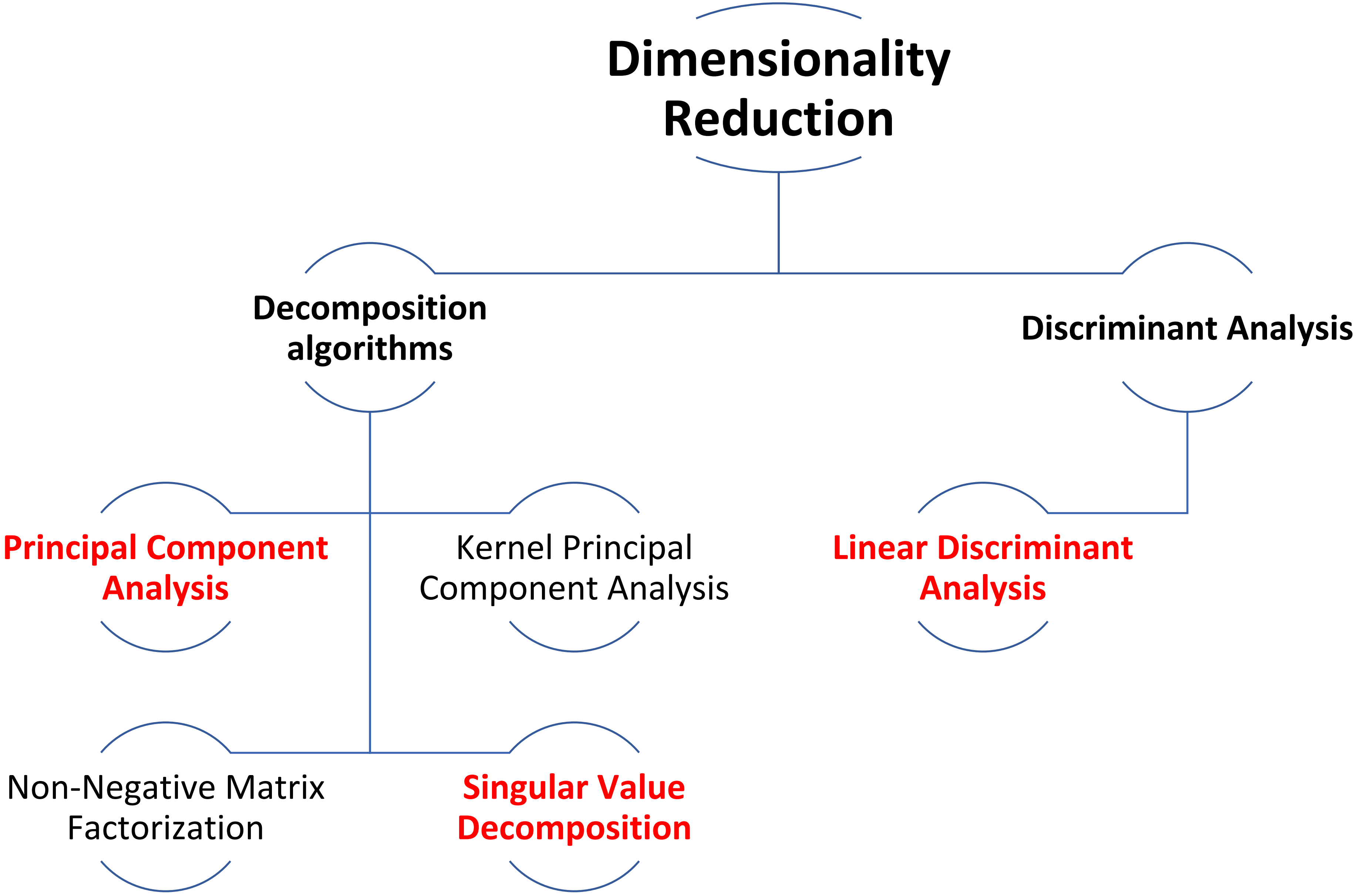
- Eliminate records having a missing value of data elements
- Imputing missing values (mean-median / similarity based mean or median)
- Assigning similar values for relevant attributes

Dimensionality Reduction

- In both Statistics and Machine Learning, the number of attributes, features or input variables of a dataset is referred to as its **dimensionality**.
- ***Dimensionality reduction*** simply refers to the process of reducing the number of attributes in a dataset while keeping as much of the variation in the original dataset as possible.
- **Dimensionality reduction** refers to the techniques of reducing the dimensionality of a data set by creating new attributes by combining the original attributes.
- The most common approach for dimensionality reduction is known as **Principal Component Analysis** (PCA)/**Feature Subset Selection.**

Dimensionality Reduction

- Advantages of Dimensionality Reduction
  - **A lower number of dimensions in data means less training time and less computational resources and increases the overall performance of machine learning algorithms**
  - **Dimensionality reduction avoids the problem of *overfitting***
  - **Dimensionality reduction is extremely useful for *data visualization***
  - **Model accuracy improves due to less misleading data**
  - **Algorithms train faster thanks to fewer data**
  - **It removes noise and redundant features**



Dimensionality Reduction

Decomposition algorithms

Discriminant Analysis

Principal Component Analysis

Kernel Principal Component Analysis

Linear Discriminant Analysis

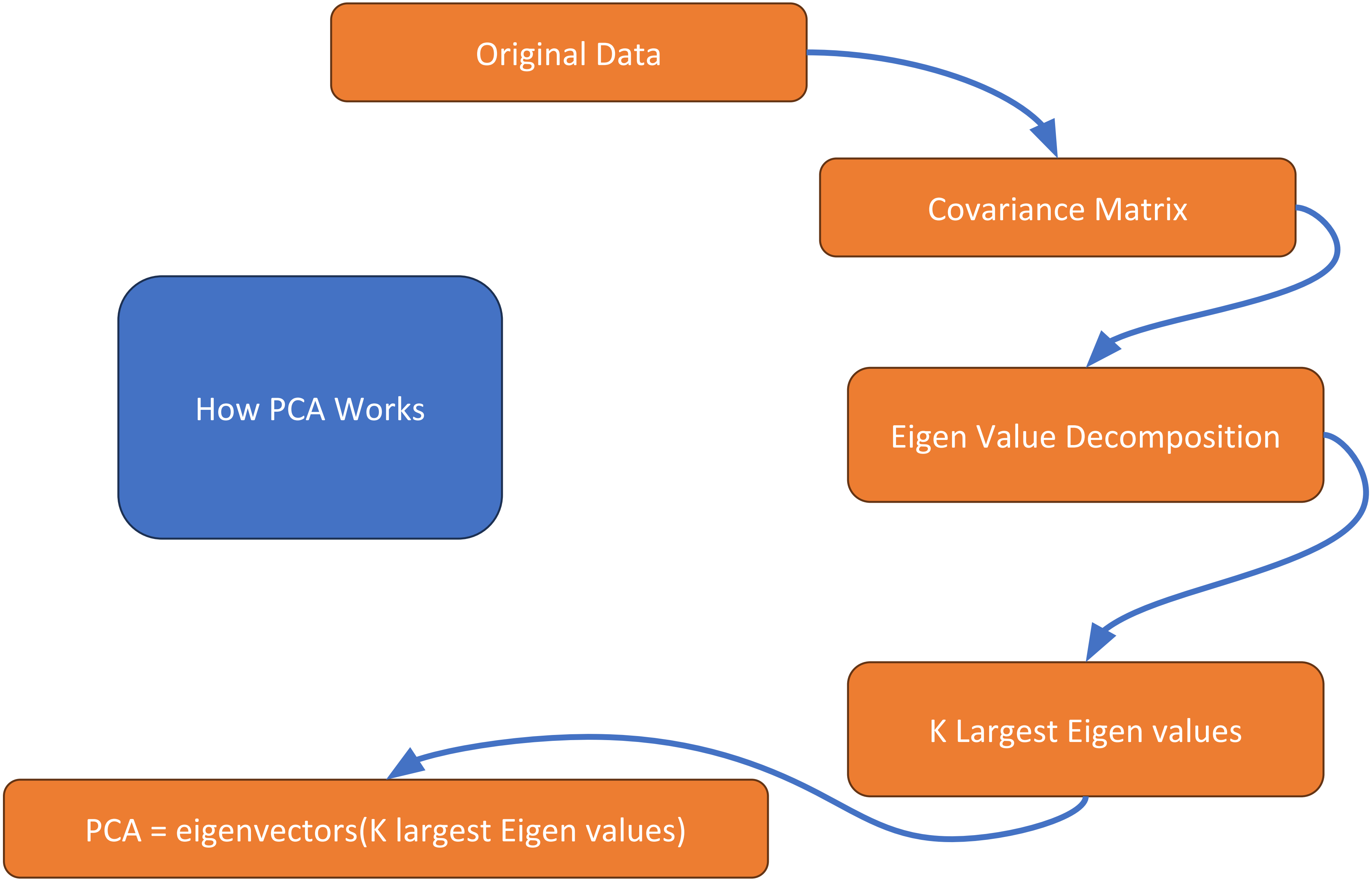
Non-Negative Matrix Factorization

Singular Value Decomposition

Principal Component Analysis (PCA)

- PCA is a statistical technique to convert a set of correlated variables into a set of transformed, uncorrelated variables called principal components.
- It is a projection based method that transforms the data by projecting it onto a set of **orthogonal(perpendicular) axes**.
- **It is unsupervised learning algorithm**
- **We are expecting high variance in for PCA to cover majority of original dataset information.**

How Does PCA Work?



How PCA Works

Original Data

Covariance Matrix

Eigen Value Decomposition

K Largest Eigen values

PCA = eigenvectors(K largest Eigen values)

PCA: Covariance Matrix

a **covariance matrix** is a square matrix giving the covariance between each pair of elements of a given random vector.

Any covariance matrix is symmetric and positive semi-definite and its main diagonal contains variances (i.e., the covariance of each element with itself). – source Wikipedia

PCA: Covariance Matrix

Happiness in getting fruits	
Apple	Banana
1	1
3	0
-1	-1

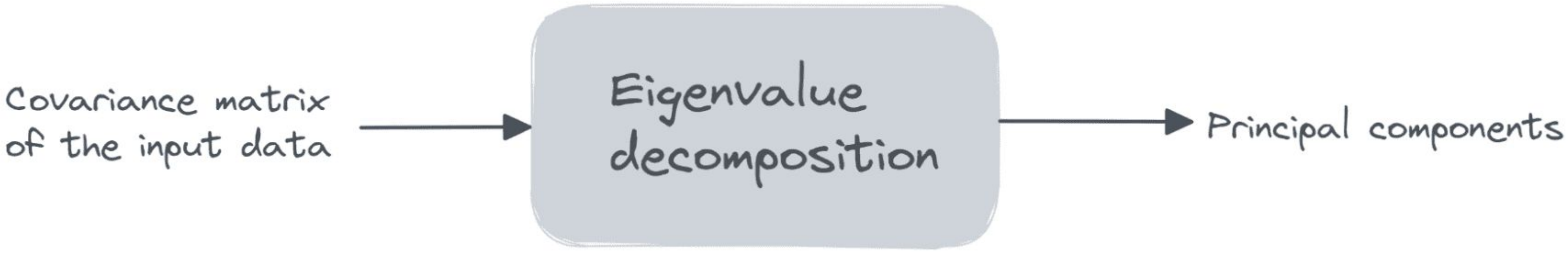
$$\left\{\begin{array}{c|c} \text{Cov(A,A)} & \text{Cov(A,B)} \\ \hline \text{Cov(B,A)} & \text{Cov(B,B)} \end{array}\right\}$$

$$cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

It’s simple to see that the covariance matrix is a square matrix of order **num_features**.

$$\left\{\begin{array}{cc} 4 & 1 \\ 1 & 1 \end{array}\right\}$$

PCA: Eigen Value Decomposition



Covariance matrix of the input data

Eigenvalue decomposition

Principal components

Eigen values and vectors

A	x	Ax
$\begin{pmatrix}1 & 2\\2 & 4\end{pmatrix}$	$\begin{pmatrix}1\\2\end{pmatrix}$	$\begin{pmatrix}5\\10\end{pmatrix}$
	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}3\\6\end{pmatrix}$

A vector which undergoes pure scaling without any rotation is known as **eigen vector.** Scaling factor (stretch ratio is known as **eigen values.**

$$Ax = \begin{pmatrix}5\\10\end{pmatrix}$$

$$Ax = 5\begin{pmatrix}1\\2\end{pmatrix}$$

$$Ax = \lambda x$$

In PCA

A=cov. Matrix

x= eigen vector

λ = eigen value

PCA: Eigen Value Decomposition (Optional)

$$Ax = \lambda \mathrm{x}$$

$$A = \left\{ \begin{matrix} 4 & 1 \\ 1 & 1 \end{matrix} \right\}$$

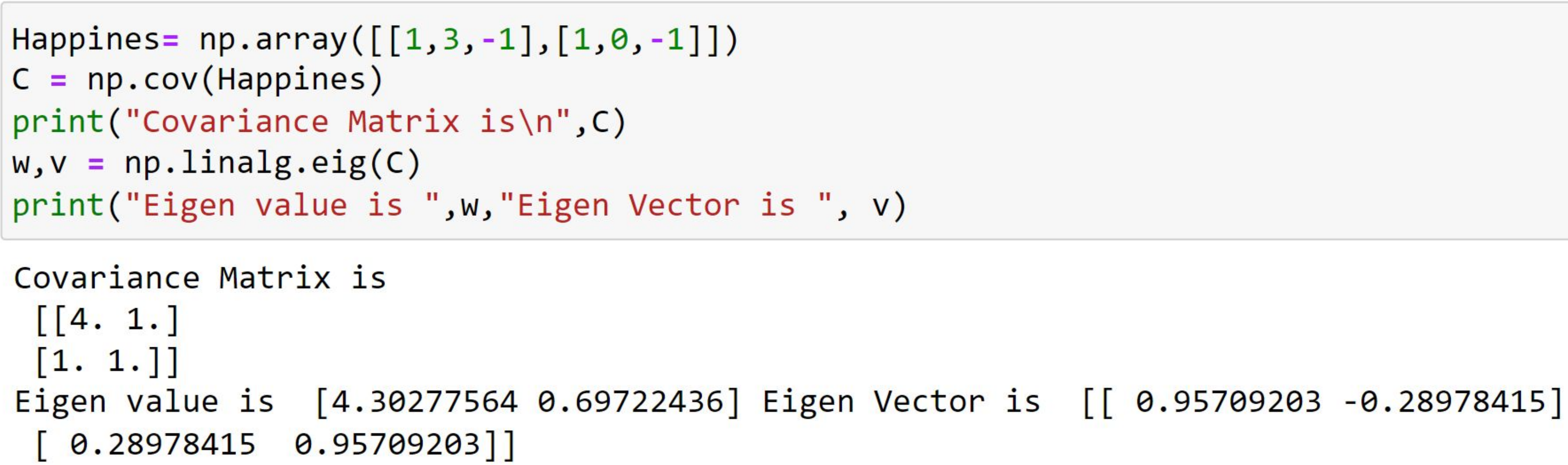
$$Ax = \lambda(x.I)$$

$$Ax - \lambda(x.I) = 0$$

$$x(A - \lambda I) = 0$$

This represents a **homogeneous system of linear equations** and it has a non-trivial solution only when the determinant of the coefficient matrix is 0.

$$|A - \lambda I| = 0$$



```
Happines= np.array([[1,3,-1],[1,0,-1]])
C = np.cov(Happines)
print("Covariance Matrix is\n",C)
w,v = np.linalg.eig(C)
print("Eigen value is ",w,"Eigen Vector is ", v)
```

```
Covariance Matrix is
 [[4. 1.]
 [1. 1.]]
Eigen value is  [4.30277564 0.69722436] Eigen Vector is  [[ 0.95709203 -0.28978415]
 [ 0.28978415  0.95709203]]
```

Because the covariance matrix is a symmetric and positive semi-definite, the eigen decomposition takes the following form:

$$X^T X = D \Lambda D^T$$

The first k principal components are the *eigenvectors* corresponding to the *k largest eigenvalues*.

Principal Components Using SVD

- Another matrix factorization technique that can be used to compute principal components is **singular value decomposition or SVD.**
- **[U,S,V] = SVD (C)**
    - **C is covariance matrix (nXn)**

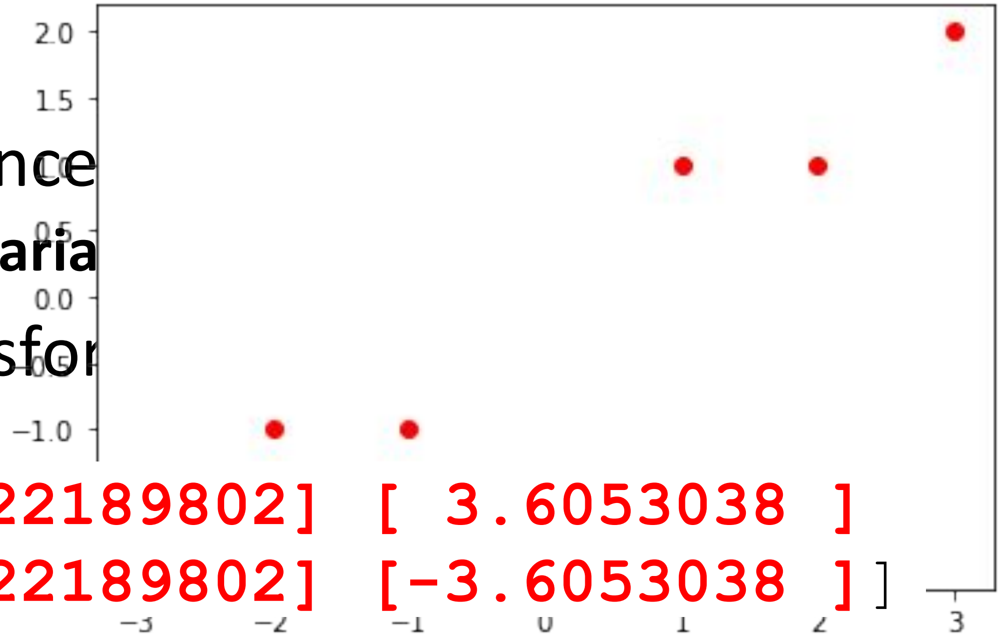
$$U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

PCA in scikit-learn

- Steps to perform PCA in scikit learn library
  - Feature Scaling (Mean normalization)
  - Run PCA algorithm to fit data to obtain 2/3 new axis (principal component) from original data set
    - fit function in scikit learn automatically carries out mean normalization
  - Examine variance by each principal component with original data set.
    - explained_variance_ration function in scikit learn
  - Transform (project) data into the new axes.
    - transform

PCA in scikit-learn (Example)

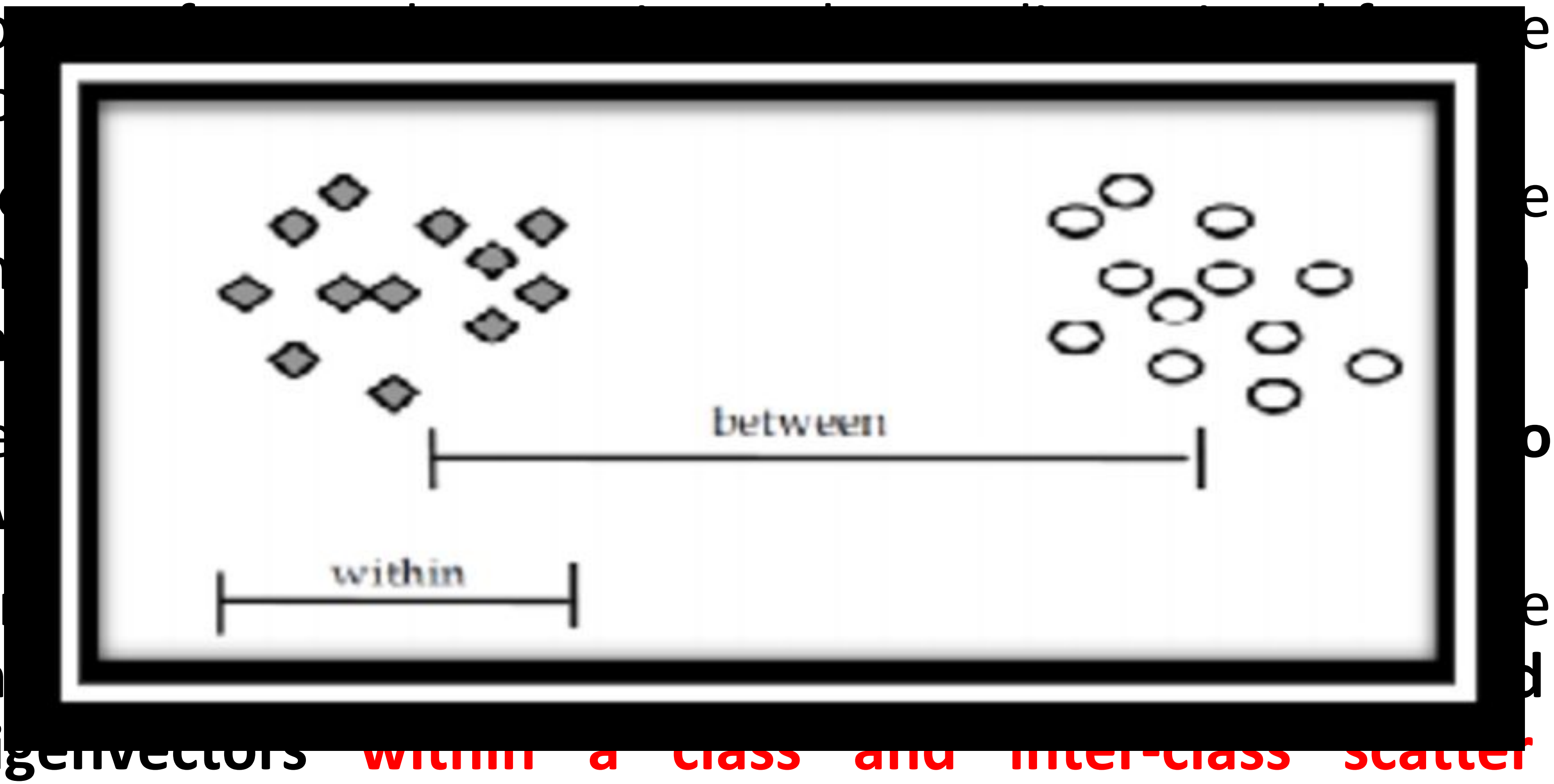
- X = np.array([[1,1],[2,1],[3,2],[-1,-1],[-2,-1],[-3,-2]])
- pca = PCA(n_components=1)
- pca.fit(X)
- pca.explained_variance
  - **0.99244289 (High Varia**
- new_axis = pca.transfor



```
[[ 1.38340578] [ 2.22189802] [ 3.6053038 ]
 [-1.38340578] [-2.22189802] [-3.6053038 ]]
```

Linear discriminant analysis (LDA)

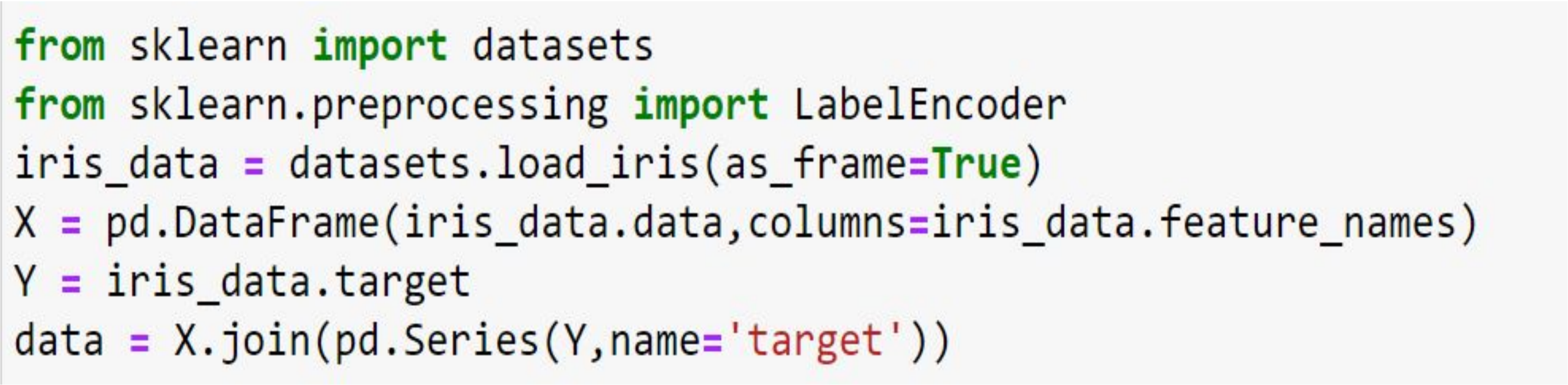
- The objective of LDA is similar to the sense that it intends to e sp



- H e th n **m**
- se **o** **av**
- U e m d **eigenvectors** **within a class and inter-class scatter matrices**.

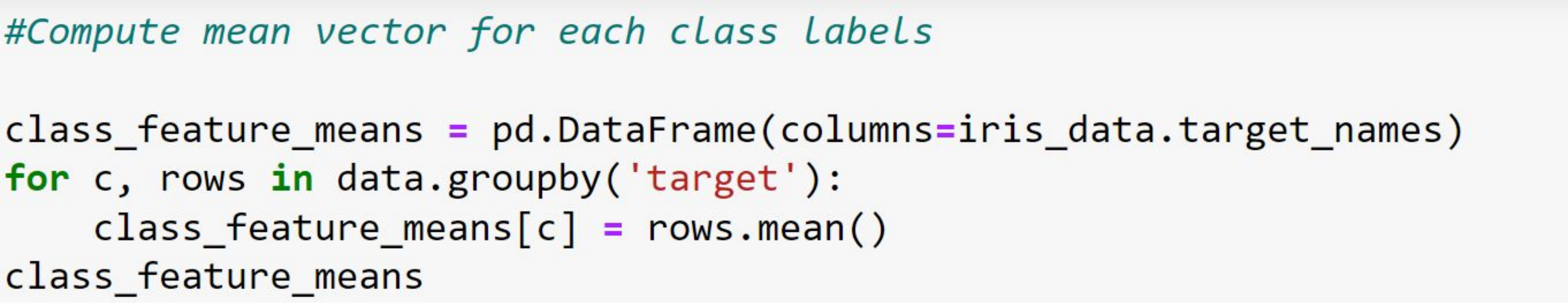
HOW LDA WORKS

- Calculate the mean vectors for the individual classes.
- Calculate intra-class and inter-class scatter matrices.
- Calculate eigenvalues and eigenvectors for S_W and S_B , where S_W is the intra-class scatter matrix and S_B is the inter-class scatter matrix.
- Identify the top ‘k’ eigenvectors having top ‘k’ eigenvalues

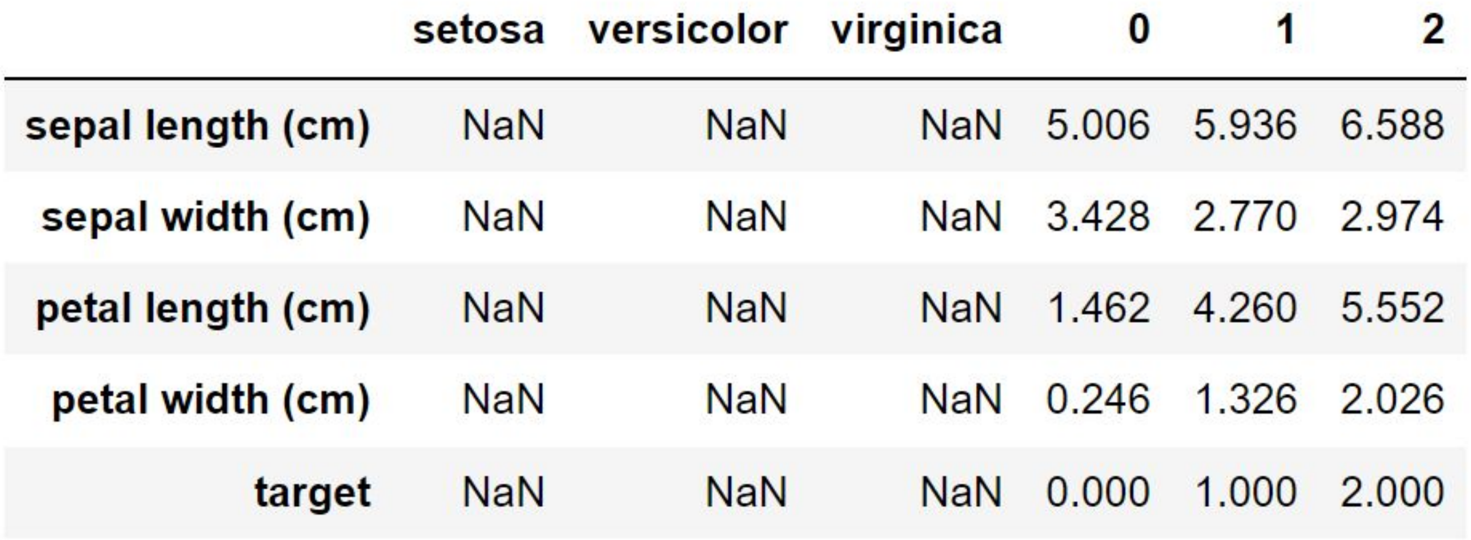


```
from sklearn import datasets
from sklearn.preprocessing import LabelEncoder
iris_data = datasets.load_iris(as_frame=True)
X = pd.DataFrame(iris_data.data,columns=iris_data.feature_names)
Y = iris_data.target
data = X.join(pd.Series(Y,name='target'))
```

Compute mean vector for each class labels



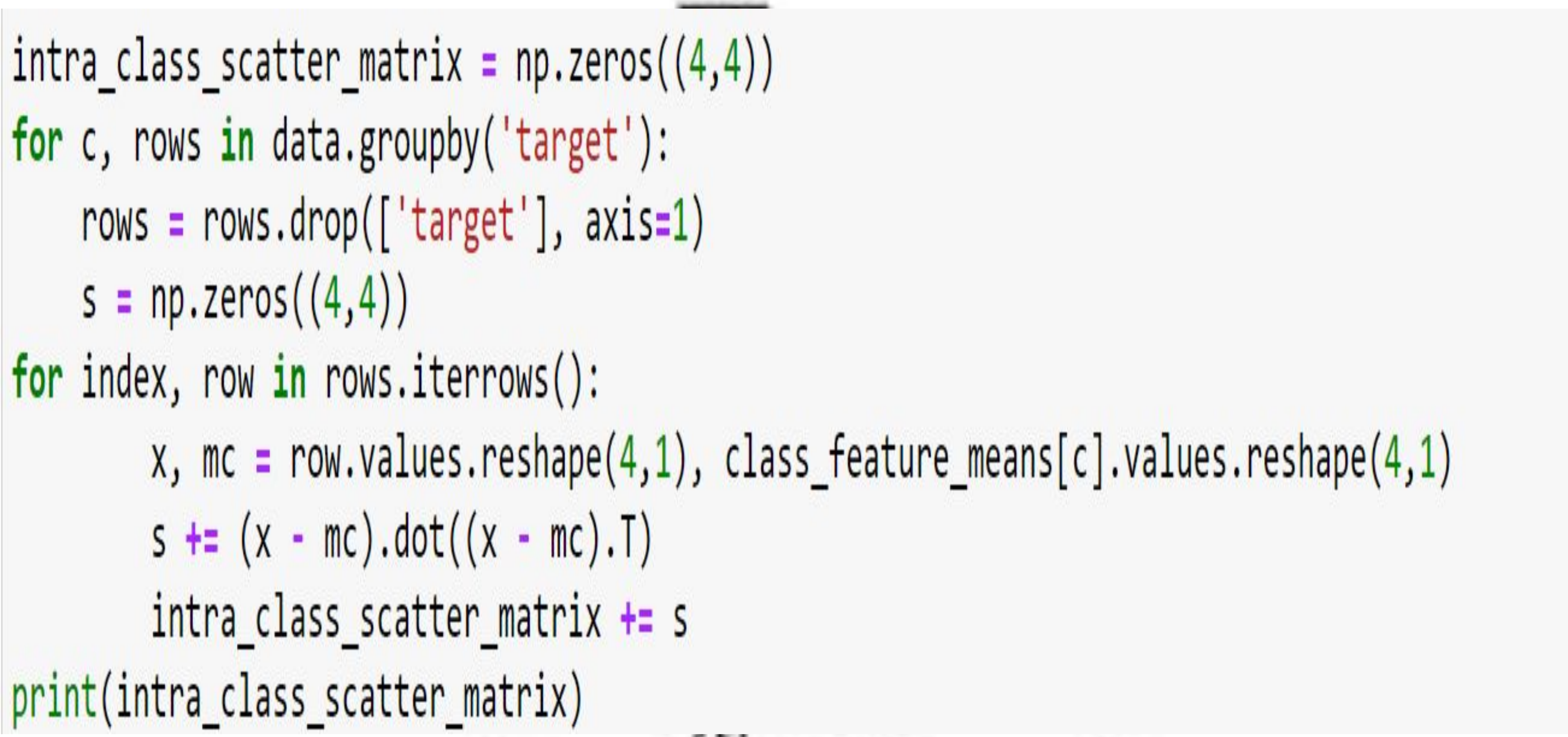
```
#Compute mean vector for each class labels

class_feature_means = pd.DataFrame(columns=iris_data.target_names)
for c, rows in data.groupby('target'):
    class_feature_means[c] = rows.mean()
class_feature_means
```



	setosa	versicolor	virginica	0	1	2
sepal length (cm)	NaN	NaN	NaN	5.006	5.936	6.588
sepal width (cm)	NaN	NaN	NaN	3.428	2.770	2.974
petal length (cm)	NaN	NaN	NaN	1.462	4.260	5.552
petal width (cm)	NaN	NaN	NaN	0.246	1.326	2.026
target	NaN	NaN	NaN	0.000	1.000	2.000

Calculate intra-class scatter matrix

c



```
intra_class_scatter_matrix = np.zeros((4,4))
for c, rows in data.groupby('target'):
    rows = rows.drop(['target'], axis=1)
    s = np.zeros((4,4))
for index, row in rows.iterrows():
        x, mc = row.values.reshape(4,1), class_feature_means[c].values.reshape(4,1)
        s += (x - mc).dot((x - mc).T)
        intra_class_scatter_matrix += s
print(intra_class_scatter_matrix)
```

$x \in D_i$

Calculate Inter-class scatter matrix

$$S_B = \sum_{i=1}^{c} N_i(\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

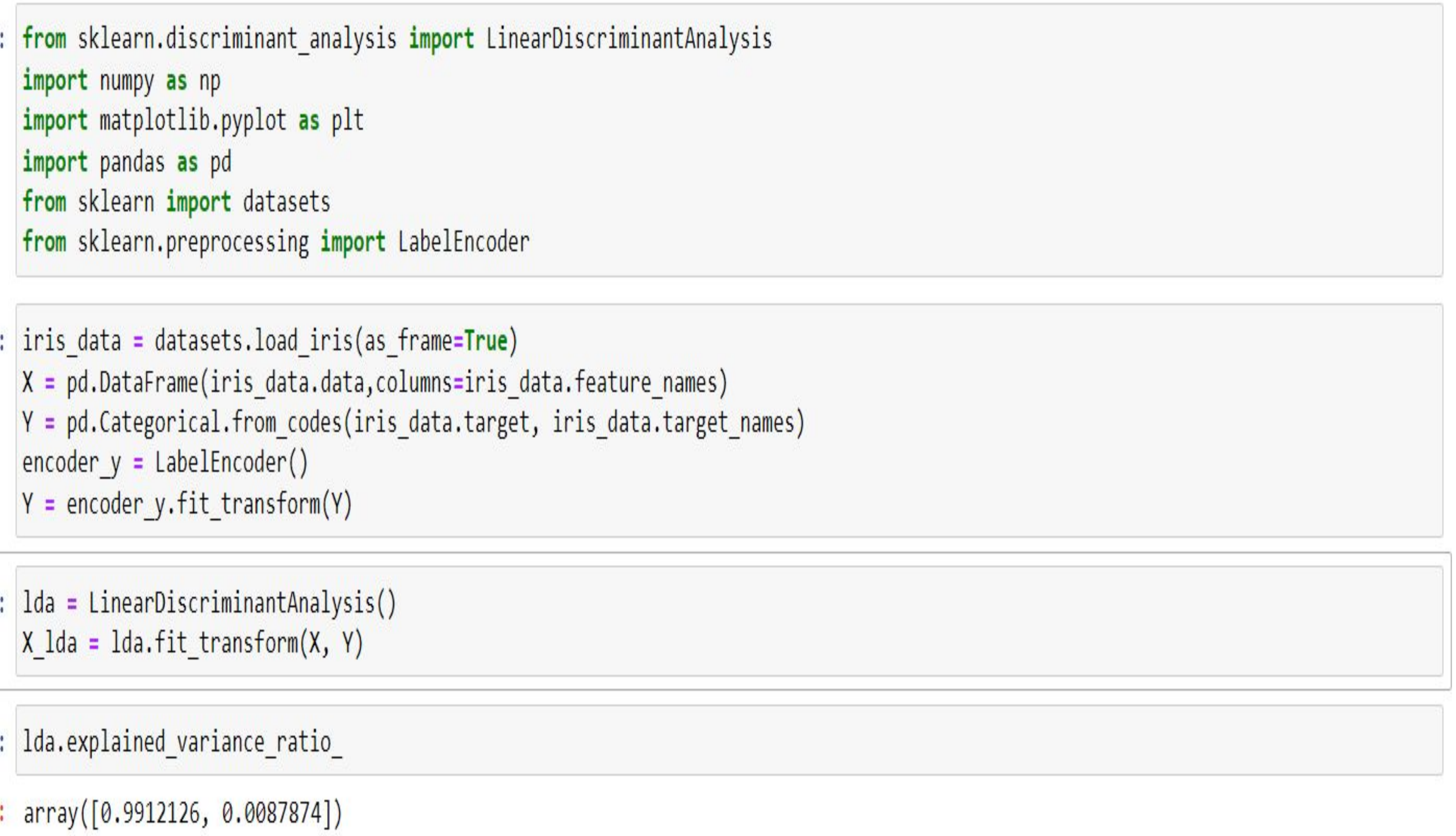
$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$$

$$m = \frac{1}{n} \sum_{i}^{n} x_i$$

Calculate eigenvalues and eigenvectors

$$S_W^{-1} S_B$$

LBA using scikit learn



```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from sklearn.preprocessing import LabelEncoder
```

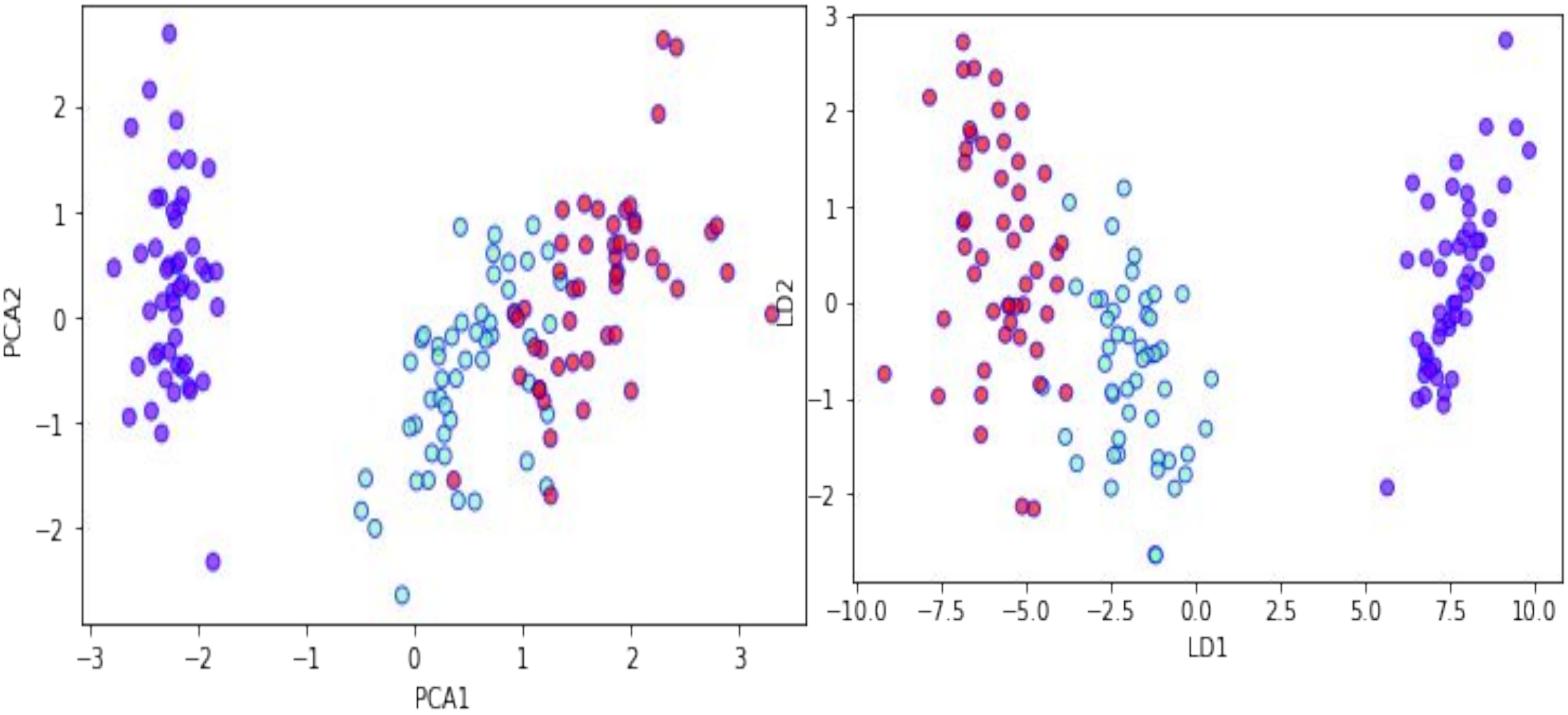
```
iris_data = datasets.load_iris(as_frame=True)
X = pd.DataFrame(iris_data.data,columns=iris_data.feature_names)
Y = pd.Categorical.from_codes(iris_data.target, iris_data.target_names)
encoder_y = LabelEncoder()
Y = encoder_y.fit_transform(Y)
```

```
lda = LinearDiscriminantAnalysis()
X_lda = lda.fit_transform(X, Y)
```

```
lda.explained_variance_ratio_
```

```
array([0.9912126, 0.0087874])
```

Comparison between PCA and LCA



Here are some key differences between PCA and LDA:

1. **Objective**: PCA is an unsupervised technique that aims to maximize the variance of the data along the principal components. The goal is to identify the directions that capture the most variation in the data. LDA, on the other hand, is a supervised technique that aims to maximize the separation between different classes in the data. The goal is to identify the directions that capture the most separation between the classes.

2. **Supervision**: PCA does not require any knowledge of the class labels of the data, while LDA requires labeled data in order to learn the separation between the classes.

3. **Dimensionality Reduction**: PCA reduces the dimensionality of the data by projecting it onto a lower-dimensional space, while LDA reduces the dimensionality of the data by creating a linear combination of the features that maximizes the separation between the classes.

4. **Output**: PCA outputs principal components, which are linear combinations of the original features. These principal components are orthogonal to each other and capture the most variation in the data. LDA outputs discriminant functions, which are linear combinations of the original features that maximize the separation between the classes.

5. **Interpretation**: PCA is often used for exploratory data analysis, as the principal components can be used to visualize the data and identify patterns. LDA is often used for classification tasks, as the discriminant functions can be used to separate the classes.

6. **Performance**: PCA is generally faster and more computationally efficient than LDA, as it does not require labeled data. However, LDA may be more effective at capturing the most important information in the data when class labels are available.