Kmit

SONET

Data Science
(Level-I)

- P.V.N.Balarama Murthy

Descriptive Statistics

Statistics

- Statistics plays very important role in Data Science
- All the required algorithms for analysis are available in statistics
- Data representation in graphical and numeric is handled in statistics

Statistics

- Why Statistics?
- Definition
- Types of Statistics
- Key-Terms

Why Statistics?

- Everyone is using statistics unknowingly.
- Ex: to buy a mobile… we may compare…

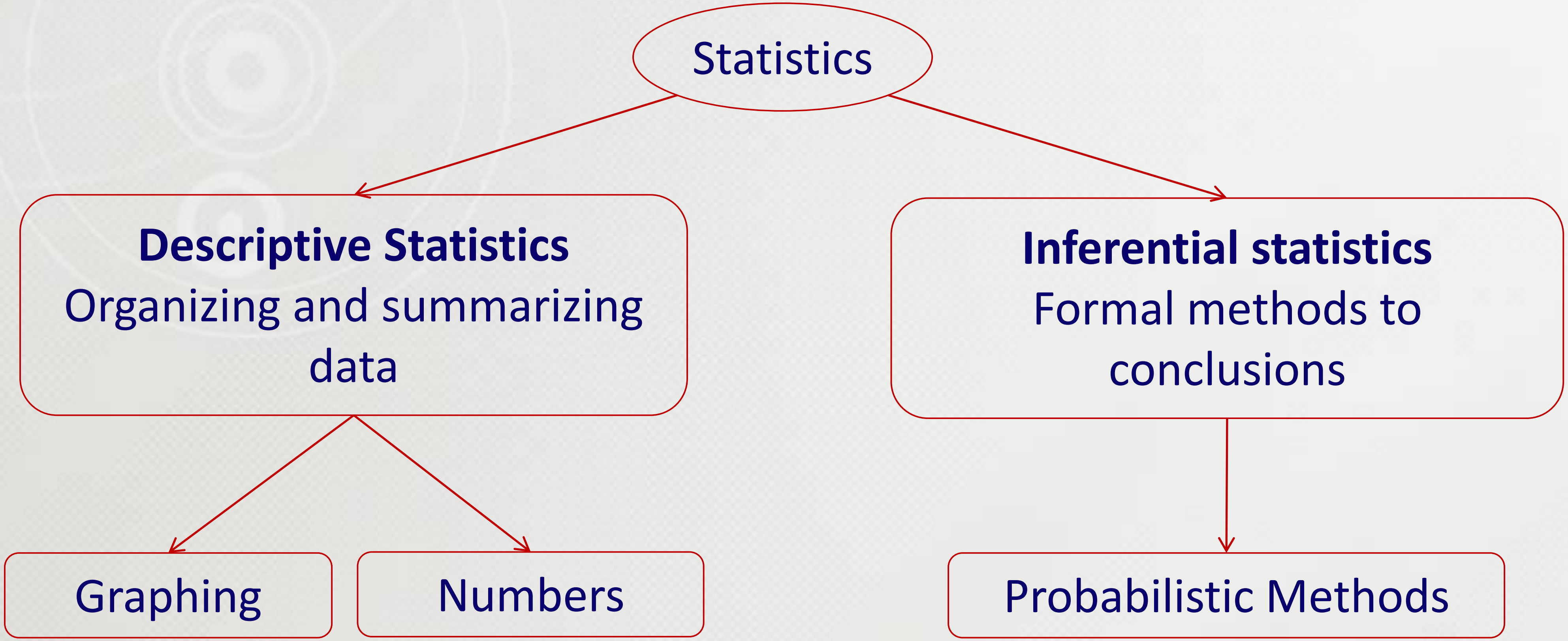
  cost Performance Warranty Etc…
- We can find statistics in (almost everywhere):

  News paper Television Sports Stock Market

- Sources provide sample information
  - We can understand and take decision
- It is the body of methods for making **wise decisions** in the face of uncertainty on the basis of numerical data and calculated risks.

Statistics

- The science of Numbers that deals with the collection, analysis, interpretation and presentation of data.



Descriptive Statistics

- Deals with Organizing and summarizing data.
- used when we know all of the values in order to describe a phenomenon

- There are 3 steps
  - Describing Data
  - Summarize / Analyze
  - Visualize

Descriptive statistics

- Variables : properties of some event, object, or person that can take on different values or amounts.

Variable

Independent (Can be changed by Experimenter) A=5, B=6	**Qualitative** (Can not be numeric) Hair_color="black", gender="M"	**Discrete** (values with discrete steps (integer)) No_of_child=2
OR	OR	OR
Dependent (Measured based on other variable) C=A+B	**Quantitative** (Can be measured as numbers) Height=5.6, Weight = 61	**Continuous** (continuous but not discrete steps(float)) Response_time=2.3

- Used for conducting analysis on one variable at a time or univariate Analysis

Descriptive statistics

The variable can be considered as data set.

X={54,56,34,44,34,65}

Common Approaches to analyze data set:

Central Tendency

“in the middle” or “popular” value in data.

Mean
Median
Mode

“most frequent observation”
most common height
most common income
most common size of shoe

Variability

How “far” away from the mean or median
how “spread out” are the data

Range
Variance
Quartile
Standard Deviation

Central Tendency

Mean: add all values and then divide by the total number of values.

Marks scored by a student in 5 subjects : 56, 31, 56, 8 and 32

Mean = $\frac{56+31+56+8+32}{5} = \frac{183}{5} = 36.60$

Mean $\bar{X} = \frac{X_1 + X_2 + \ldots + X_{N-1} + X_N}{N}$ = > $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$

Calculating the mean in Python :

```
(56+31+56+8+32)/5
```

```
36.6
```

```
x = [56, 31, 56, 8, 32]
print (sum(x)/len(x))
```

```
36.6
```

Numpy provides a builtin function mean()

```
import numpy as np
x = np.array([56, 31, 56, 8, 32])
print (np.mean(x))
```

```
36.6
```

SONET

Machine Learning

Central Tendency

- **Median:** The median of a set of observations is just the middle value(50 percentile).
- Make all the elements in ascending or descending order

- Marks scored by a student in 5 subjects : 56, 31, 56, 8 and 32

Ascending Order: $8, 31, \mathbf{32}, 56, 56$

- Median = 32

- Scott took 8 math tests in one marking period. What is the median test score? Scores =[89, 73, 84, 91, 87, 77, 94, 67]
- Median = (scores[3]+scores[4])/2 = 134.5
- Note: Median does not depend on every value in your data set.

Numpy provides a function median():

```
import numpy as np
x = np.array([56, 31, 56, 8, 32])
print (np.median(x))
```

```
32.0
```

Central Tendency

Mean Vs Median:

- If data are nominal scale (Label/String), you probably shouldn't be using either the mean or the median.
- If data are ordinal scale(values in order), you're more likely to want to use the median than the mean.
- For interval(integer)and ratio(float) scale data, either one is generally acceptable.

Central Tendency

Mean Vs Median:

Choosing median or mean plays an important role in analysis

Example: Income of 3 employees is given : Ramu-50,000 krishna-65,000 Sastry-60,000

Subbu added with 10,00,00,000

```
import numpy as np
income = [50000,65000,60000]
print (np.mean(income))
print (np.median(income))
```

```
58333.3333333
60000.0
```

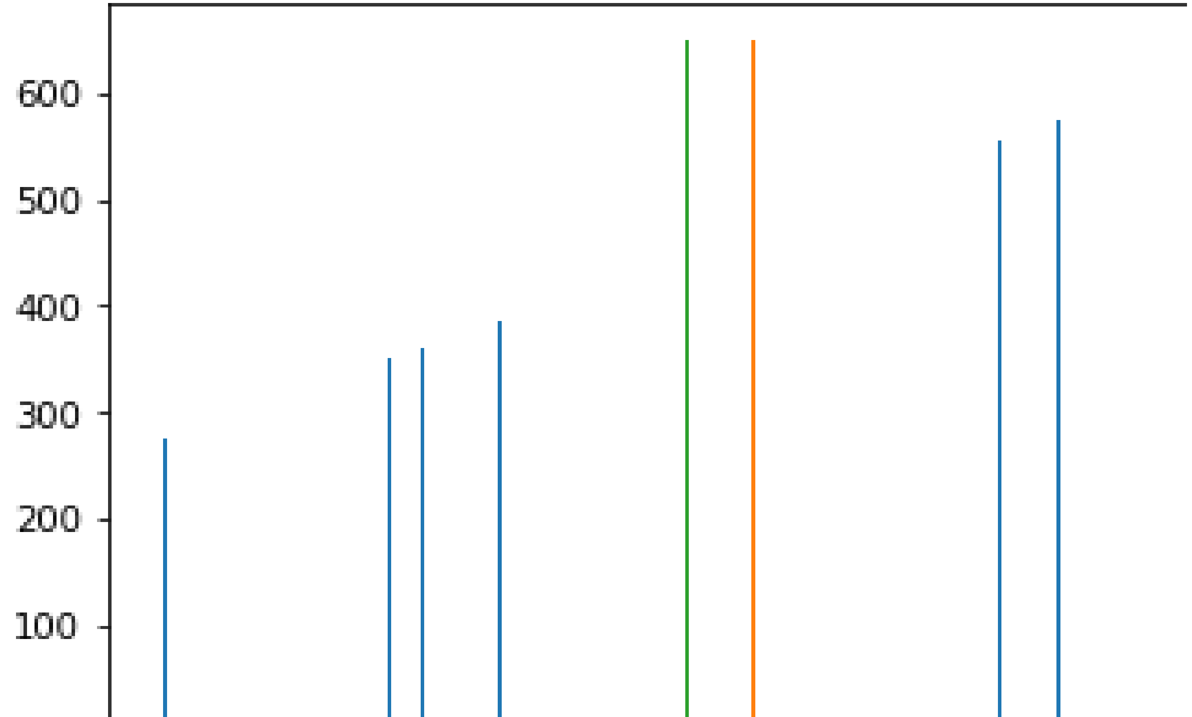
```
import numpy as np
incme_new = [50000,65000,60000,100000000]
print (np.mean(incme_new))
print (np.median(incme_new))
```

```
25043750.0
62500.0
```

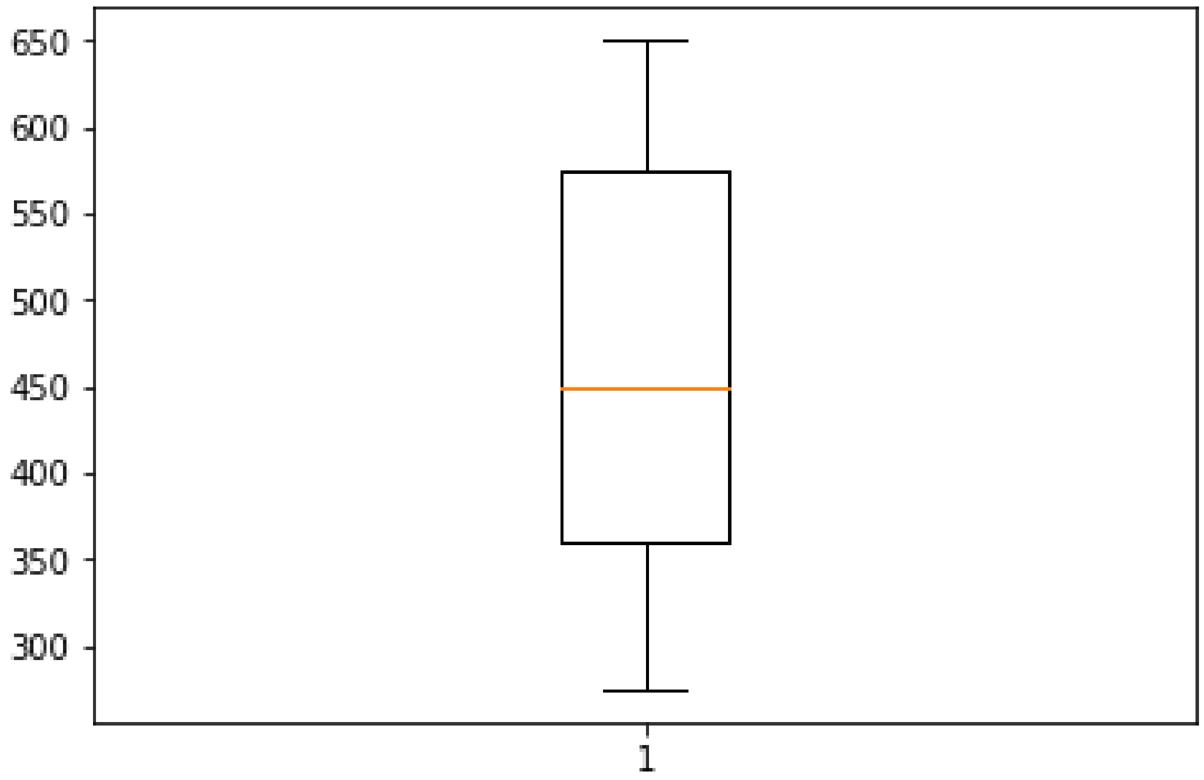
Note: For extreme-valued observations called outliers, mean is not suitable. Because the result would be misleading.

```python
import matplotlib.pyplot as plt
data = np.array([650,450,275,350,387,575,555,649,361])
print("mean = ",np.mean(data))
print("median = ",np.median(data))
plt.bar(data,data)
plt.bar(np.mean(data),data)
plt.bar(np.median(data),data)
plt.show()
```

```
mean =  472.444444444
median =  450.0
```




```python
plt.boxplot(data)
plt.show()
```

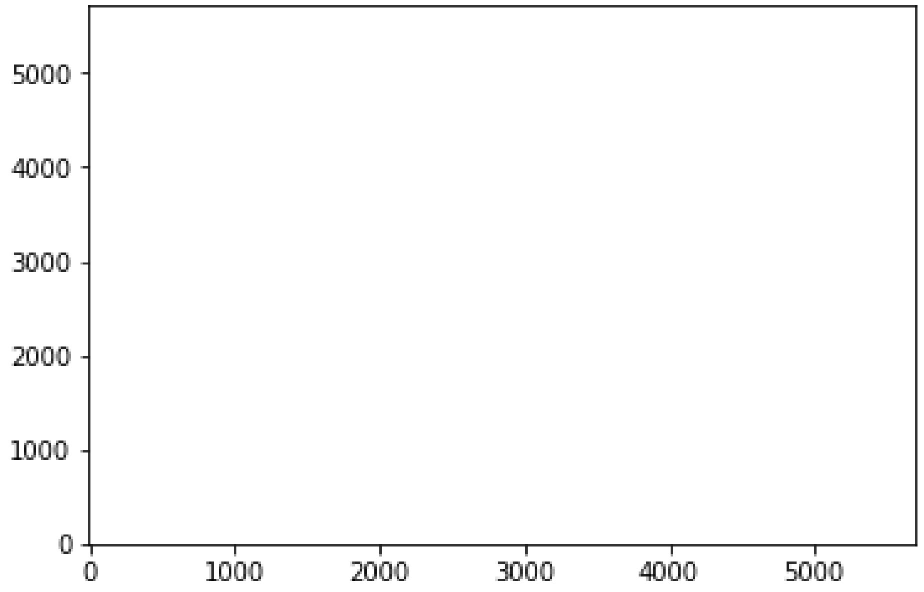


Central Tendency

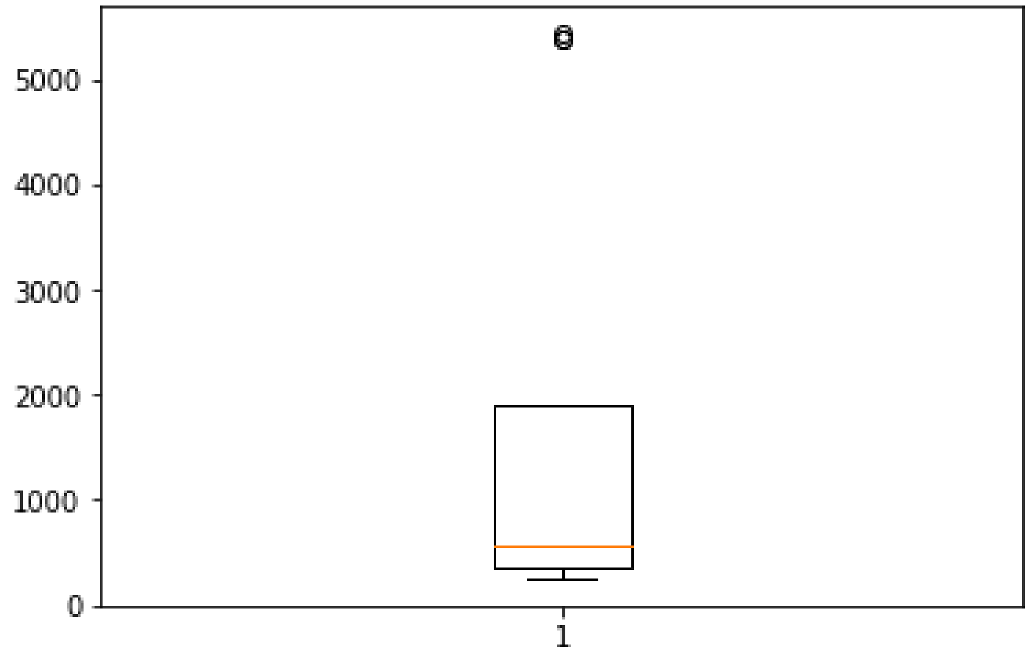
Mean Vs Median:

```
import matplotlib.pyplot as plt
data = np.array([650,430,275,252,367,573,555,749,361,5400,5402,5438])
print(np.mean(data))
print(np.median(data))
plt.bar(data,data)
plt.bar(np.mean(data),data)
plt.bar(np.median(data),data)
plt.show()
```

```
1704.33333333
564.0
```




```
plt.boxplot(data)
plt.show()
```



Measures of Dispersion or Variability

- Dispersion – measures or summarize the amount of spread or variation in the distribution of values in a variable.
- how values are spread out within a distribution.
- There are a number of different measures:
  - Range
  - Variance
  - Standard Deviation

Range

Range = **highest Value - lowest value** in the data set.

- Ex: Consider Weekly scores of 10 students.

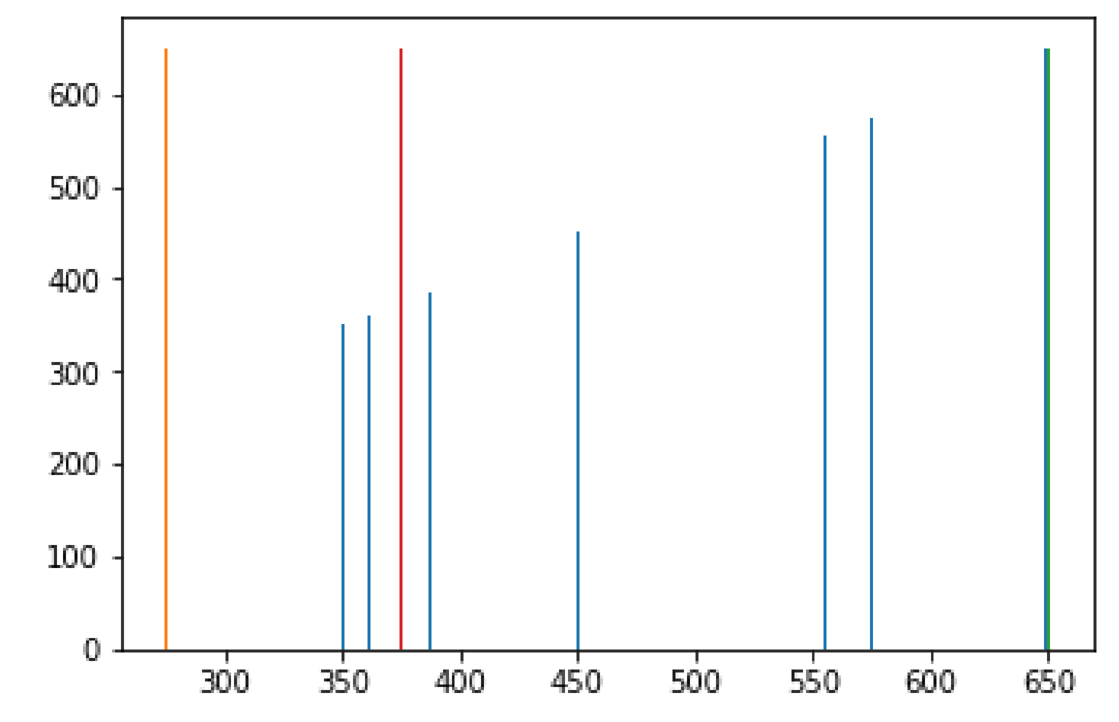
  Weekly_scores=[180, 220, 280, 320, 280, 180, 350, 280, 330, 220]

  range = 350-180 = 170.

- **if r = 0, all elements are same. If r is large, the data is more spread out. It does not depend on whole dataset.**

```
data = np.array([650,450,275,350,387,575,555,649,361])
range=np.max(data)-np.min(data)
print("range = ",range)
print("max = ", np.max(data))
print("min = ", np.min(data))
plt.bar(data,data)
plt.bar(np.min(data),data)
plt.bar(np.max(data),data)
plt.bar(range,data)
plt.show()
```

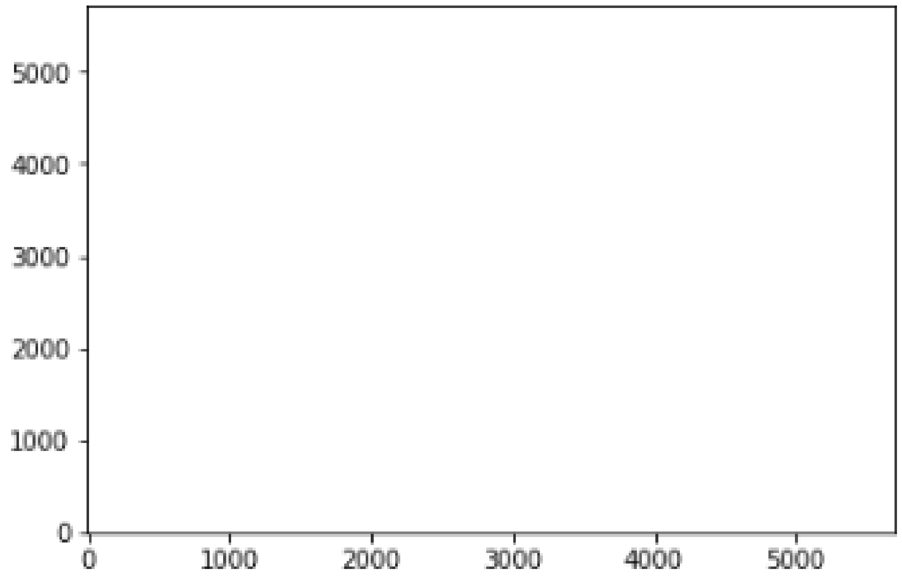
```
range =  375
max =  650
min =  275
```



Range

```
data = np.array([650,430,275,252,367,573,555,749,361,5400,5402,5438])
range=np.max(data)-np.min(data)
print("range = ",range)
print("max = ", np.max(data))
print("min = ", np.min(data))
plt.bar(data,data)
plt.bar(np.min(data),data)
plt.bar(np.max(data),data)
plt.bar(range,data)
plt.show()
```

```
range =  5186
max =  5438
min =  252
```



Percentile

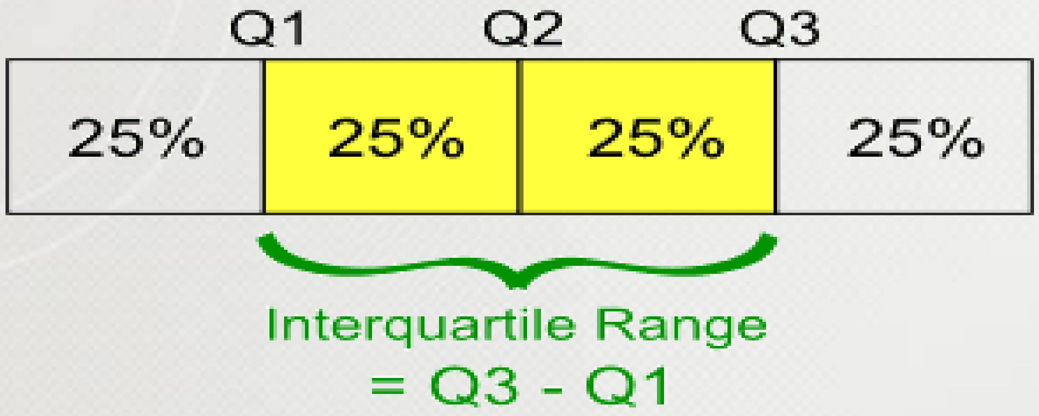
The value at location associated with the given percentage of data set.

```
data = np.array([650,430,275,252,367,573,555,749,361,5400])
print(data)
print(np.sort(data))
print(np.percentile(data,0))
print(np.percentile(data,10))
print(np.percentile(data,20))
print(np.percentile(data,30))
print(np.percentile(data,40))
print(np.percentile(data,50))
print(np.percentile(data,75))
print(np.percentile(data,90))
```

```
[ 650  430  275  252  367  573  555  749  361 5400]
[ 252  275  361  367  430  555  573  650  749 5400]
252.0
272.7
343.8
365.2
404.8
492.5
630.75
1214.1
```

Interquartile Range

- The **interquartile range** (**IQR**) is a measure of variability, based on dividing a data set into quartiles. Quartiles divide a rank-ordered data set into four equal parts. The values that divide each part are called the first, second, third, & fourth quartiles; and they are denoted by Q1, Q2, and Q3, respectively.



- A **measure of where the "middle fifty" is in a data set**. It is where the bulk of the values lie.
- It Calculates difference between 25th quantile and 75th quantile.

Interquartile Range

```
def interquartile_range(x):
    return np.percentile(x,75)-np.percentile(x,25)
weights = [100, 49, 41, 40, 25,78, 57, 89, 45,34,12,34, 45, 56,67, 87, 89, 90,81, 73]
print(np.sort(weights))
print ("Q1 = ",np.percentile(weights,25))
print ("Q3 = ",np.percentile(weights,75))
print ("Q2 = ", np.percentile(weights,50))

print ("InterQuantileRange = ", interquartile_range(weights))
```

```
[ 12  25  34  34  40  41  45  45  49  56  57  67  73  78  81  87  89  89
  90 100]
Q1 =  40.75
Q3 =  82.5
Q2 =  56.5
InterQuantileRange =  41.75
```

Variability

- **Mean Absolute Deviation:** It is the average distance (deviation) of all the elements in the dataset from the mean of the same dataset.

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$$

```
data = np.array([2,3,4,5,6,7,8,9])
print (np.mean(data)-data)
print (np.abs(np.mean(data)-data))
print (np.mean(abs(np.mean(data)-data)))
```

```
[ 3.5  2.5  1.5  0.5 -0.5 -1.5 -2.5 -3.5]
[ 3.5  2.5  1.5  0.5  0.5  1.5  2.5  3.5]
2.0
```

Variance

- **Variance :** determines how close the scores in the distribution are to the middle of the distribution.
- The average squared difference of the scores from the mean.

For complete data set $\sigma^2 = \frac{\sum(X-\mu)^2}{N}$ **For Sample:** $s^2 = \frac{\sum(X-M)^2}{N-1}$

- Observations: 2,3,4,5,6,7,8,9
- Mean =5.5
- $(|\text{mean-observations}|)^2$ = 12.25 6.25 2.25 0.25 0.25 2.25 6.25 12.25
- variance = (12.25 +6.25+2.25+0.25+0.25+2.25+6.25+12.25)/8 =5.25

- Range Vs Variance:
  - range looks at the extremes (**biggest value-smallest value)**
  - variance **looks at all the data points. It determines distribution of data around mean**.
  - Variance - **average** (**squared deviations** from the **Mean)**

Variance

- Procedure:
  - Find out the Mean
  - for each element: subtract the Mean, then square the result (the *squared difference*).
  - Find the average (squared differences).

```
# assumes x has at least two elements
def variance(x):
  n = len(x)
  deviations = de_mean(x)
  return sum_of_squares(deviations)/(n-1)

>>>dataset=[2,3,4,5,6,7,8,9]
>>> print variance(scores)
>>> 6.0
```

- **Variance = 0 , when all the elements are identical**

Standard Deviation

- Standard Deviation – the square root of Variance.
- used to find the outliers.
- useful measure of variability when the distribution is normal

The "Population Standard Deviation“:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

The "Sample Standard Deviation":

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

Correlation

- **Correlation** – show whether and how strongly pairs of variables are related.
  - For Example**:** height, weight are related; taller people tend to be heavier than shorter people.
  - The **relationship isn't perfect**. People of the same height vary in weight,
  - consider two people you know where the shorter one is heavier than the taller one. The average weight of people 5'5'' is less than the average weight of people 5'6'', etc.
  - **Correlation can tell you just how much of the variation in peoples' weights is related to their heights.**

Correlation

- **Correlation works for quantifiable data** (Numerical Data).
- It cannot be used for purely categorical data, such as gender, brands purchased, or favourite colour.
- Correlation is used to understand the relationship between variables such as:
  - Whether the relationship is +ve or –ve
  - the strength of the relationship.
- **Positive correlation**: is a relationship between two variables where if one variable increases, the other one also increases and vice-versa
  - Eg: family size , family expenditure will increase or decrease together.

Correlation

- **Negative Correlation**: there is an inverse relationship between two variables - when one variable decreases, the other increases and vice-versa.
  - **Eg: negative correlation**, between **price** and **demand** for goods and services. As the price of goods and services increases, the quantity demanded falls.
- **Coefficient of Correlation (r):** Statistical correlation is measured is known as "**coefficient of correlation (r)**". Its numerical value ranges from +1.0 to -1.0. It gives us **the strength of relationship.**

Correlation

- In general, r > 0 – positive relationship
- r<0 – negative relationship
- r = 0 – no relationship (meaning the variables are independent and not related)
- when r = +1.0 – describes a perfect +ve correlation
- when r = -1.0 – describes a perfect -ve correlation
- closer the coefficients are to +1.0 and -1.0, greater is the strength of the relationship between the variables.

Covariance

- **Covariance** - indicates how two variables are related ie, is **how two variables are deviated from their means**.
- A positive covariance - variables are positively related,
- A negative covariance - variables are inversely related.

$$COV(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n-1}$$

x = independent variable, *y* = dependent variable
n = number of data points in the sample
x', y' = the mean of *x, y respectively*

- A **large +ve covariance** means – **x tends to be large, when y is large, and small when y is small**.
- A **large -ve covariance** means – **x tends to be large, when y is small, and small when y is large**

SONET

Machine Learning

Covariance

```
def de_mean(x):
   """ to calculate x – x’ """
   x_bar = mean(x)
   return [i - x_bar for i in x]

def covariance(x,y):
    n=len(x)
    return dot(de_mean(x),de_mean(y))/(n-1)
```

```
ht=[1.73,1.68,1.71,1.89,1.79]
wt=[65.4,59.2,63.6,88.4,68.7]
print("Covariance = ", np.cov(ht,wt))
print("Covariance = ", np.cov(ht,wt)[0][1])
```

```
Covariance =  [[  6.90000000e-03   9.18750000e-01]
 [  9.18750000e-01   1.28648000e+02]]
Covariance =  0.91875
```

- **This example is showing large +ve covariance** meaning – ht **tends to be large, when** wt **is large, and small when** wt **is small**.

SONET

Machine Learning

Correlation

- **#correlation – divides out the standard deviations of both variables x,y.**

```
def correlation(x,y):
    sd_x = std_deviation(x)
    sd_y = std_deviation(y)
    if sd_x > 0 and sd_y > 0:
        return covariance(x,y) / sd_x / sd_y
    else:
        return 0
```

```
>>> print correlation(ht,wt)
0.975149690509
```

```
ht=[1.73,1.68,1.71,1.89,1.79]
wt=[65.4,59.2,63.6,88.4,68.7]
print("Covariance = ", np.cov(ht,wt)[0][1])
print("Correlation Coefficient = ", np.corrcoef(ht,wt)[0][1])
```

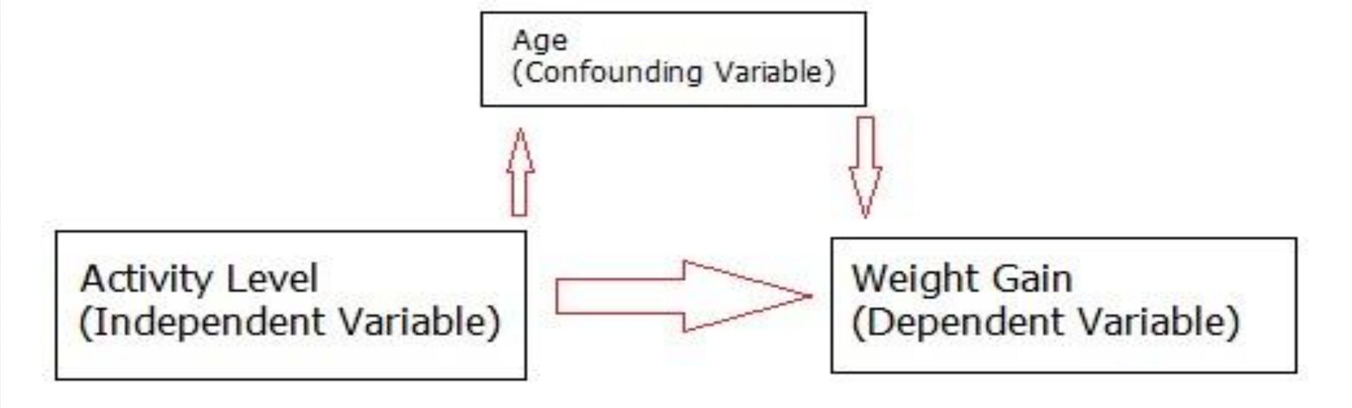
```
Covariance =  0.91875
Correlation Coefficient =  0.975149690509
```

Correlation

- While 'r' (correlation coefficient) is a powerful tool, it has to be handled with care.
- mostly correlation coefficients only **measure linear relationship**. It is also possible that there is strong **non linear relationship** between the variables, r is close to 0 or even 0. In such a case, a scatter diagram can roughly indicate the existence or otherwise of a non linear relationship.
- Must be careful in interpreting the value of 'r'. For Ex, one could compute 'r' between the size of shoe and intelligence of individuals, heights and income. Which is non sense.
- 'r' should not be used to say anything about cause and effect relationship.

Simpson's Paradox

- It is a commonly occurring situation when analyzing the data. That is correlations can be **misleading** when **confounding variables** are ignored.
- A **confounding variable** is an "**extra**" variable that you didn't account for in your experimental design.
- Issues:
  - Increase variance
  - introduce bias
  - useless results
  - correlation
- Avoidance:
  - Control variables
  - random assignment



A confounding variable can have a hidden effect on your experiment's outcome.

Conclusion

Discussed about ...

- Files – Reading – Writing

Next Session …..

Data Pre-Processing

THANK YOU