Lecture - 06

→ CHI - SQUARE Test.

→ Covariance.

→ Pearson Correlation Coefficient.

→ Spearman Rank Correlation.

→ Practical Implementation [z-test, t-test, Chi-square]

→ f-Test [ANNOVA]

CHI - SQUARE TEST:

→ Chi-square test Claims about population proportions.

→ It is a non parametric test that is performed on Categorical (nominal or ordinal) data.

Problem: [Based on Chi-square test]

1. In the 2000 Indian Census, the age of the individual in a small town were found to be the following.

Less than 18	Age (18 - 35)	Above 35
20%	30%	50%

In 2010, age of $n = 500$ individuals were Sampled. Below are the results.

<18	18-35	>35
121	288	91

Would you Conclude the population distribution of ages has Changed in the last 10 years? [using $\alpha = 0.05$]

Given,

In 2000,

<18	18-35	>35
20%	30%	50%

	<18	18-35	>35	
2010	121	288	91	→ Observed
2000 with respect to Value with Sample mean	0.2×500 $= 100$	0.3×500 $= 150$	0.5×500 $= 250$	→ Expected

	<18	18-35	>35	
2010	121	288	91	Observation
2000	100	150	250	Expected

(i) Null Hypothesis (H_0),

The Observation data meets the distribution of 2000 Census.

(ii) Alternate Hypothesis (H_1),

The observation data does not meets the distribution of 2000 Census.

(iii) Significance value $\alpha = 0.05$ (95% of CI)
[Given].

(iv) Degree of freedom = n-1

not take n=500, because it is applicable for Categorical values. So, take the category of <18, 18-35, >35.

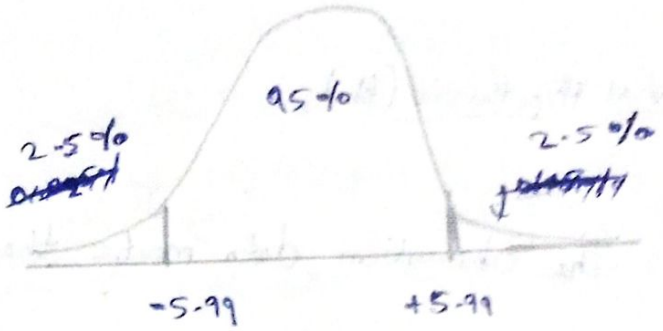
3 diffoment Samples.

$= n-1 = 3-1 = 2$ $\boxed{df=2}$

(V) Decision Boundary,

The Population distribution is high or low. So,

Use two-tail test. $[df = 2, \alpha = 0.05]$



$df = 2, \alpha = 0.05$

⇓ Chi-square table value

5.99

(vi) Calculate Test Statistics:

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

$f_o \rightarrow$ observation value

$f_e \rightarrow$ expected value

$\chi^2 \rightarrow$ Representation of Chi-square.

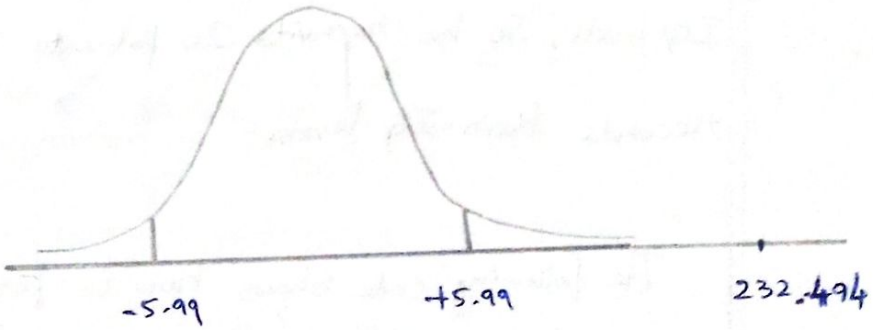
$$= \frac{(121-100)^2}{100} + \frac{(288-150)^2}{150} + \frac{(91-250)^2}{250}$$

$$\boxed{\chi^2 = 232.494}$$

(vii) State Decision:

$\chi^2 = 232.494 > 5.99$ { Reject the null hypothesis }

So, we accept alternate hypothesis.



Obviously, The population distribution of ages was Changed and increased in the last 10 years.

Z-test	T-Test	CHI-SQUARE TEST
Formula,	Formula,	Formula, Test stats ↙ $\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$
Point estimate ± margin of Error	Point estimate ± margin of error	
$\bar{x} \pm z_{\alpha/2}\left(\frac{\sigma}{\sqrt{n}}\right) \rightarrow CI$	$\bar{x} \pm t_{\alpha/2}\left(\frac{s}{\sqrt{n}}\right) \rightarrow CI$	upperbound = $[\chi^2(1-\alpha), \infty]$
Test Statistics,	Test statistics,	lowerbound = $(0, \chi^2(\alpha)]$
$z = \frac{\bar{x} - \mu}{\left[\frac{\sigma}{\sqrt{n}}\right]}$	$t = \frac{\bar{x} - \mu}{\left[\frac{s}{\sqrt{n}}\right]}$	CI = $[\chi^2(1-\alpha/2), \chi^2(\alpha/2)]$

PRACTICAL: [Python] Z-test

Suppose the IQ in a certain population is normally distributed with a mean $\mu = 100$ and standard deviation of $\sigma = 15$.

A researcher wants to know if a new drug affects IQ levels, so he recruits 20 patients to try it and records their IQ levels.

The following code shows how to perform a One Sample Z-test in Python to determine if the new drug causes a significant difference in IQ levels.

Program:

```
from Statsmodels.stats.weightstates import ztest as ztest

#Enter IQ level for 20 patients
data = [88, 92, 94, 94, 96, 97, 97, 97, 99, 99,
        105, 109, 109, 110, 112, 112, 113, 114, 115]

ztest(data, value=100) # value is Population Mean
```

Output:

(1.5976240527147705, 0.11012667014384426)

↓ Z test value

↓ P-value

Conditions:

P-value < Significance value

⇓

Reject the null hypothesis

P-value > Significance value

⇓

Accept the null hypothesis

Based on P-value, The Computer rejects or accepts the hypothesis.

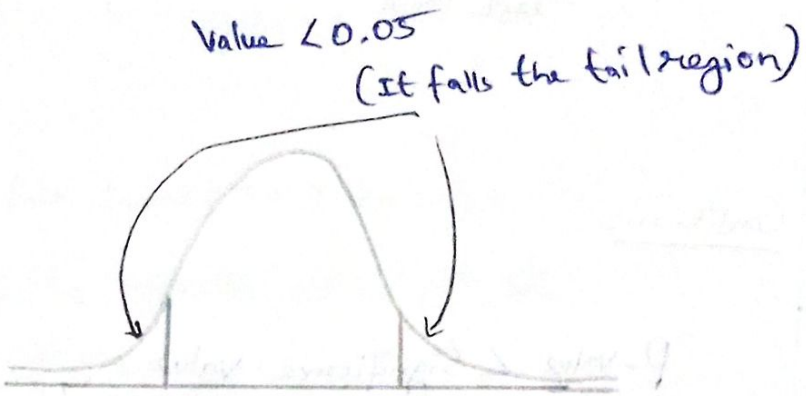
The above Program P-value > Significance Value.

0.11 > 0.05 (defalut significance value)

So, Accept the null hypothesis.

~~So, the~~

So, The new drug cannot a significant difference in IQ levels.



Co-Variance:

Let us Consider, Two features are

x	y
Weight	Height
50	160
60	170
70	180
75	181

Note,

x↑ y↑

x↓ y↓

Let us Consider another One,

X	Y
No of hours Study	play
2	6
3	4
4	3
5	2

Note,

X ↑ Y ↓

X ↓ Y ↑

Co-variance is the measure of two relationship between two features in the form of Quantity.

formula,

$$\text{Covariance}, \; Cov(x,y) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N}$$

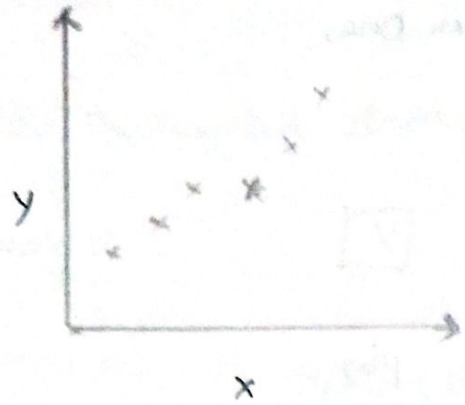
Working with Sample $(n-1)$ instead of N in the formula

Apply this formula, we get

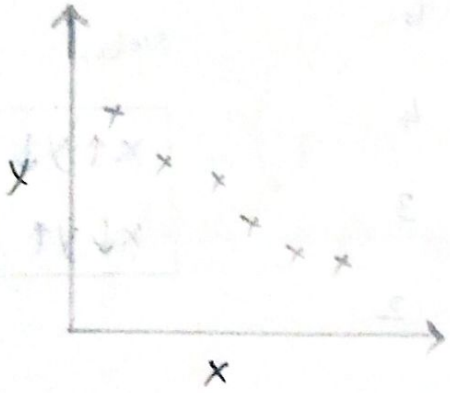
= +ve (or) -ve

↓ Positive Correlation

↓ Negative Correlation



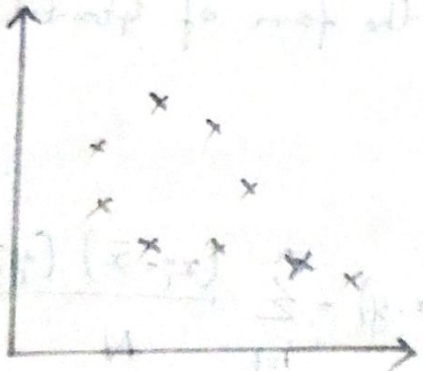
Positive Correlation $x\uparrow y\uparrow$ $x\downarrow y\downarrow$



Negative Correlation $x\uparrow y\downarrow$ $x\downarrow y\uparrow$

$\Downarrow$ (or) say

Negative Covariance



Covariance = 0

[There no relationship between the features]

Disadvantage of Covariance:

1. Return the positive or negative value but the values are quite large like +100, +1000, -3000 like this. So we cannot measure within the particular

range.

2. The Magnitude there is no such limit. (ie. between this range to this range). So, it is hard to measure.

Advantage:

We can find the direction is positive or negative. But the problem is not measurable.

To Solve this, we use Pearson Correlation Coefficient.

Pearson Correlation Coefficient:

This is used for measure the relationship within particular range from (-1 to 1).

→ The more towards +1 then say More Positively Correlation.

→ The more towards -1 then say More negatively Correlation.

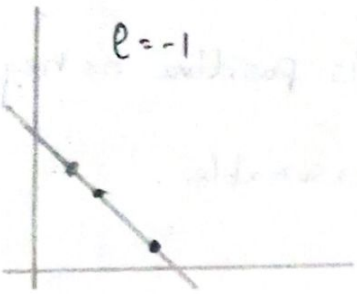
Formula,

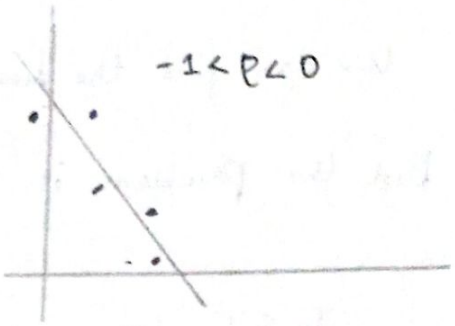
$\sigma_x \rightarrow$ sd of feature x

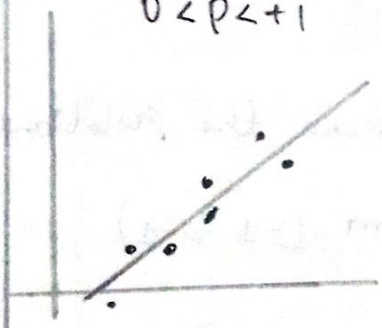
$\sigma_y \rightarrow$ sd of feature y

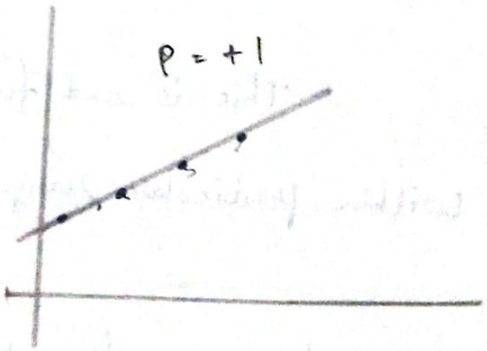
$$\rho(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

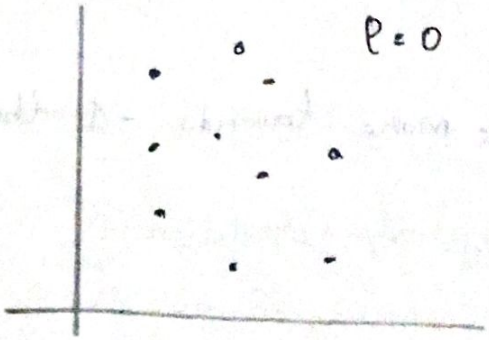
Value = {-1 to 1} (Ans)









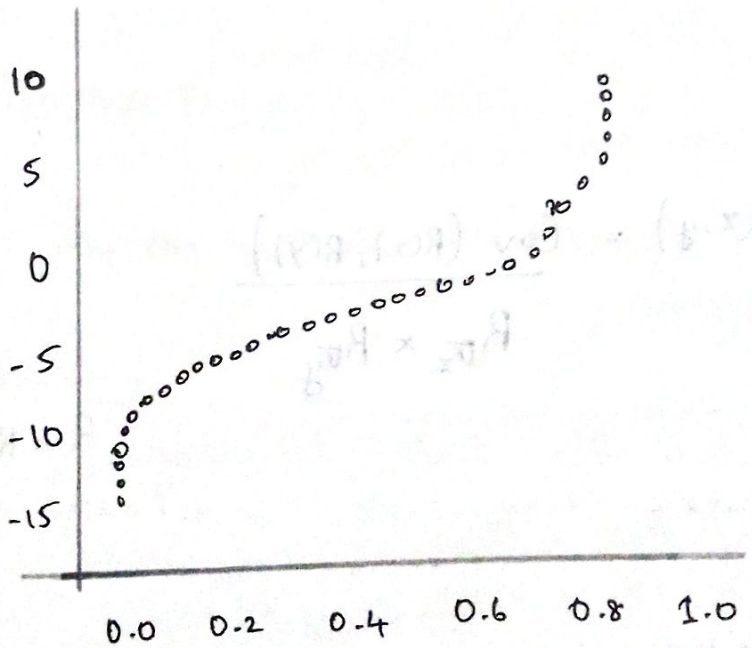


Does not follow straight line we say range from -1 < ρ < 0 (or) 0 < ρ < +1 [Range]

Disadvantage:

→ The Pearson Correlation Coefficient is only works well for Linear properties. Does not fit for non-Linear properties. So, we jump into Spearman rank Correlation.

Spearman's rank Correlation Coefficient:



In this Scenario,

Spearman Correlation = 1

Pearson Correlation = 0.88

Spearman Considered best because, it is non-linear Property. So works well and also show accurate Correlation.

Advantage:

→ It is somemore similar to pearson Correlation, The difference is it works well for non-linear Property as well.

→ It also Calculate the relation accurately. (i.e. One point overlab with some distance also Calculate. But, pearson Cannot).

formula,

$$Spear(x,y) = \frac{Cov(R(x), R(y))}{R\sigma_x \times R\sigma_y}$$

R → Rank

How to calculate?

Let Consider,

x	y	R(x)	R(y)
170	75	2	2
160	62	3	3
150	60	4	4
145	55	5	5
180	85	1	1

x → Height

y → weight

Based on the higher value, Rank will be assigned.

The formula only works with R(x), R(y), Ro_x and Ro_y. It neglect the Original feature X and Y while performing.

The main thing of the Spearman Rank Correlation is that capture non-linear properties as well.

PRACTICAL EXAMPLE: [T-Test]

The QIA is previous Z-test QIA.

PROGRAM:

```
ages = [10, 20, 35, 50, 28, 40, 55, 18, 16, 55, 30, 25, 43, 18,
        30, 28, 14, 24, 16, 17, 32, 35, 26, 27, 65, 18, 43, 23,
        21, 20, 19, 70]

import numpy as np

ages_mean = np.mean(ages)

SampleSize = 10

age_sample = np.random.choice(ages, sample_size)
```

```
from scipy.stats import ttest_1samp

ttest_1samp(age_sample, 30)   # 30 population mean
```

Output:

```
Ttest1SampResult(statistic = 0.3141021657 74, pvalue = 0.760602 4)
```

Another Example: [T-Test]

```
# Ages of the college students (population)
# 1 class students mean of all the ages.

import numpy as np
import pandas as pd
import scipy.stats as stats
import math

np.random.seed(6)

School_ages = stats.poisson.rvs(loc=18, mu=35, size=1500)
ClassA_ages = stats.poisson.rvs(loc=18, mu=30, size=60)
```

```
# To find School age mean

School_age_mean = np.mean(school_ages)

ttest_1samp(classA_ages, popmean = School_age_mean)

#return (statistic = -9.604795, Pvalue = 1.1390270710162e-13)

# check the hypothesis

if P_Value < 0.05:

        Print ("Accept Ho")

else:

        Print ("Reject HO")
```

Output: Accept Ho [Pvalue less than 0.05 because it is exponent value]

PRACTICAL FOR CORRELATION:

```
import Seaborn as Sns

df = sns.load_dataset ('iris')

df.head ()
```

```
df.corr()  # return correlation for every two features.

# To visualize correlation use pairplot.

sns.pairplot(df)
```