

Debug

```
n [*]: import pdb

#interactive debugging
def seq(n):
    for i in range(n):
        pdb.set_trace() # breakpoint
        print(i)
    return

seq(5)

# c : continue
# q: quit
# h: help
# list
# p: print
# p locals()
```

Exploratory Data Analysis :

This is very imp step to analyze data before create a model .

First check shape of data

Then no of columns

Then check data balanced or in balanced

Then plot 2d,3d,Pair plots to get more ideas on data .

```

: # pairwise scatter plot: Pair-Plot
# Dis-advantages:
##Can be used when number of features are high.
##Cannot visualize higher dimensional patterns in 3-D and 4-D.
##Only possible to view 2D patterns.
plt.close();
sns.set_style("whitegrid");
sns.pairplot(iris, hue="species", size=3);
plt.show()
# NOTE: the diagonal elements are PDFs for each feature. PDFs are explained

```

Pair plot will not work great if data is more than 10 dimensions .

(3.4) Histogram, PDF, CDF

```

5]: # What about 1-D scatter plot using just one feature?
#1-D scatter plot of petal-length
import numpy as np
iris_setosa = iris.loc[iris["species"] == "setosa"];
iris_virginica = iris.loc[iris["species"] == "virginica"];
iris_versicolor = iris.loc[iris["species"] == "versicolor"];
#print(iris_setosa["petal_length"])
plt.plot(iris_setosa["petal_length"], np.zeros_like(iris_setosa["petal_length"]))
plt.plot(iris_versicolor["petal_length"], np.zeros_like(iris_versicolor["petal_length"]))
plt.plot(iris_virginica["petal_length"], np.zeros_like(iris_virginica["petal_length"]))
plt.show()
#Disadvantages of 1-D scatter plot: Very hard to make sense as points
#are overlapping a lot.
#Are there better ways of visualizing 1-D scatter plots?

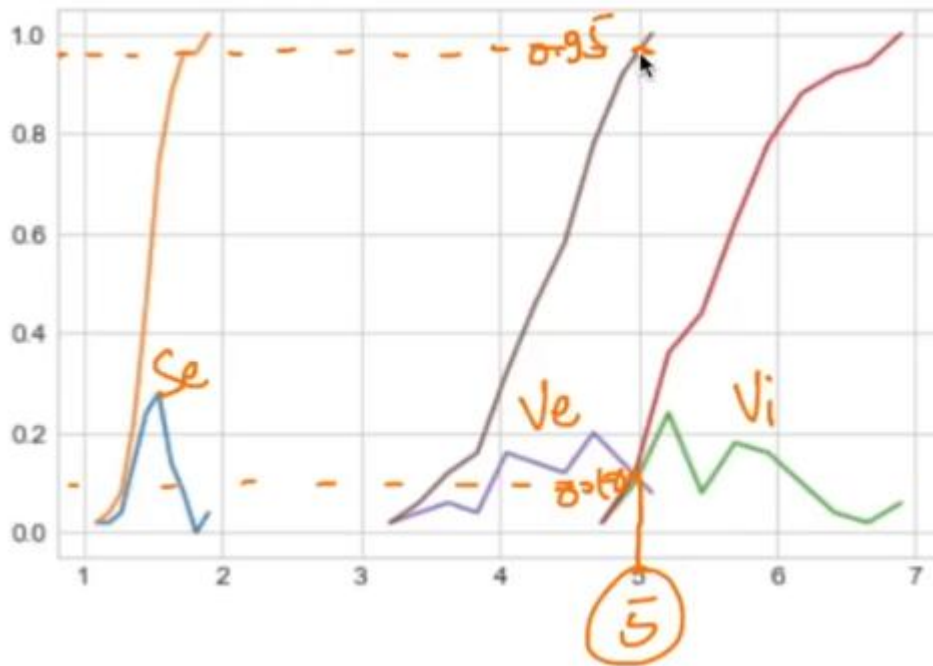
```

```

: sns.FacetGrid(iris, hue="species", size=5) \
    .map(sns.distplot, "petal_length") \
    .add_legend();
plt.show();

```

Pdf and cdf plots are very imp to tell how much probability our prediction will be correct .



(3.5) Mean, Variance and Std-dev

```
: #Mean, Variance, Std-deviation,
print("Means:")
print(np.mean(iris_setosa["petal_length"]))
#Mean with an outlier.
print(np.mean(np.append(iris_setosa["petal_length"],50)));
print(np.mean(iris_virginica["petal_length"]))
print(np.mean(iris_versicolor["petal_length"]))

print("\nStd-dev:");
print(np.std(iris_setosa["petal_length"]))
print(np.std(iris_virginica["petal_length"]))
print(np.std(iris_versicolor["petal_length"]))
```

Means are very imp to see or understand data for example :

```
Means:
✓ 1.464
2.41568627451
✓ 5.552
4.26
```

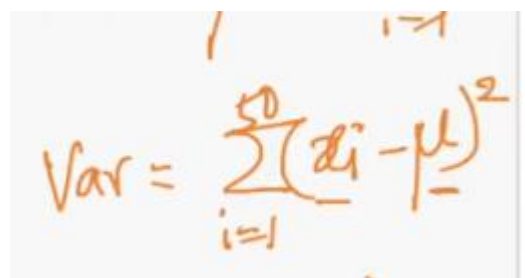
If we see above 1,3,and 4 we say petal length is almost similar mean for last 2 flowers and 1st flower has very low mean .

So we can say just by seeing mean is 1st flower is easily separable from last 2 flowers .

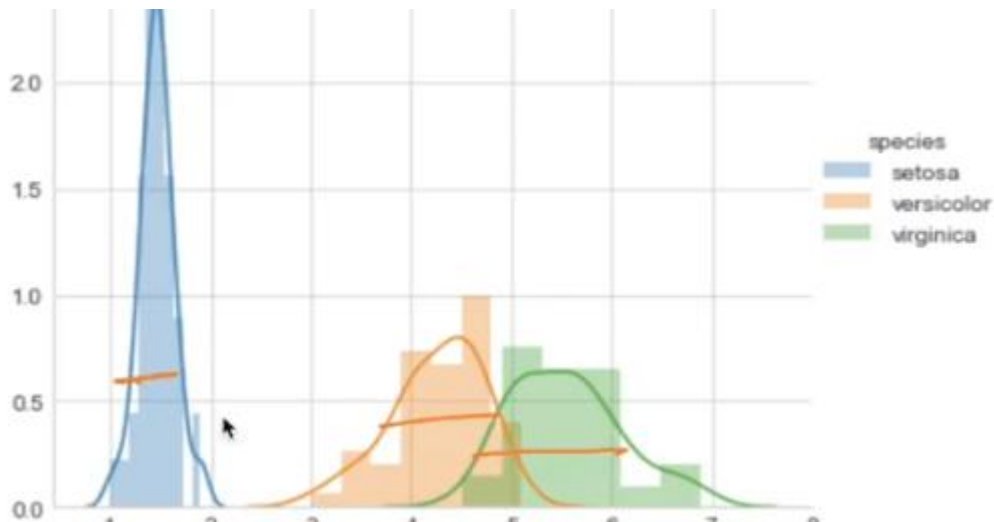
But the problem with mean is just if there is only very small mistake it will make very large impact on mean . for example if all numbers lie between 1 to 2 and only one number = 50 then mean will change very high just because of one wrong data point .

Variance is just to understand how much data spread widely in short range of data for each flower .

We can measure variance just by doing sub from mean of data point and take square after that .


$$Var = \sum_{i=1}^n (x_i - \mu)^2$$

μ is mean and x_i are the data points .



That orange line indicate variance of data .

Standard deviation is just square root of variance.

All the are gets corrupted just by one single mistake in data .

(3.6) Median, Percentile, Quantile, IQR, MAD

```

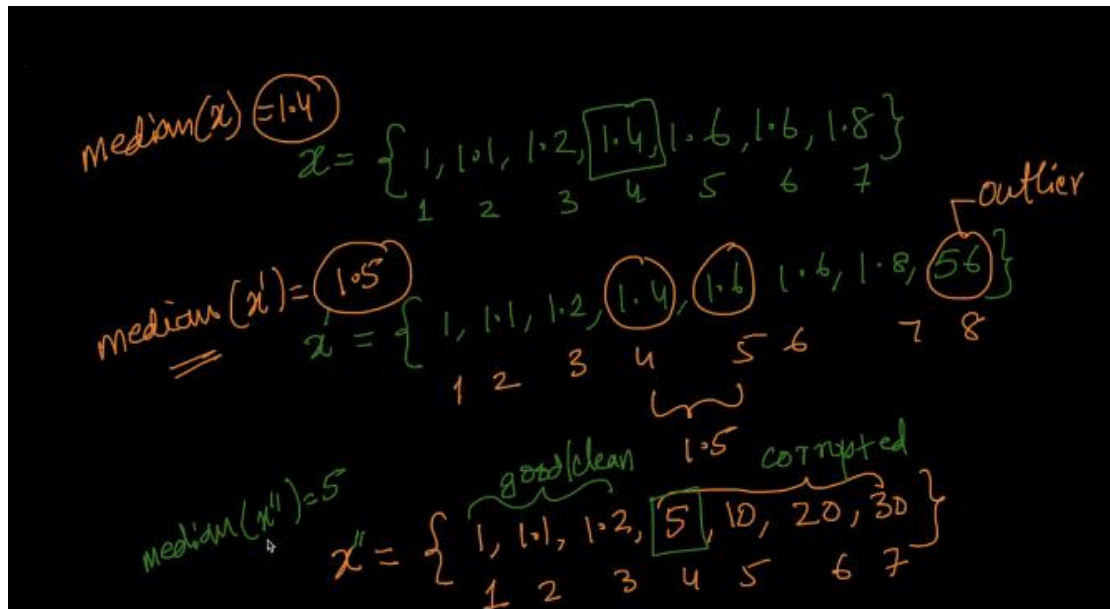
In [167]: #Median, Quantiles, Percentiles, IQR.
print("\nMedians:")
print(np.median(iris_setosa["petal_length"]))
#Median with an outlier
print(np.median(np.append(iris_setosa["petal_length"],50)));
print(np.median(iris_virginica["petal_length"]))
print(np.median(iris_versicolor["petal_length"]))

print("\nQuantiles:")
print(np.percentile(iris_setosa["petal_length"],np.arange(0, 100, 25)))
print(np.percentile(iris_virginica["petal_length"],np.arange(0, 100, 25)))
print(np.percentile(iris_versicolor["petal_length"], np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(iris_setosa["petal_length"],90))
print(np.percentile(iris_virginica["petal_length"],90))
print(np.percentile(iris_versicolor["petal_length"],90))

```

To solve problem of mean we have median it will not corrupt as only one,2 wrong data point .



```
print("\nQuantiles:")
print(np.percentile(iris_setosa["petal_length"], np.arange(0, 100, 25)))
print(np.percentile(iris_virginica["petal_length"], np.arange(0, 100, 25)))
print(np.percentile(iris_versicolor["petal_length"], np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(iris_setosa["petal_length"], 90))
print(np.percentile(iris_virginica["petal_length"], 90))
print(np.percentile(iris_versicolor["petal_length"], 90))

from statsmodels import robust
print("\nMedian Absolute Deviation")
print(robust.mad(iris_setosa["petal_length"]))
print(robust.mad(iris_virginica["petal_length"]))
print(robust.mad(iris_versicolor["petal_length"]))
```

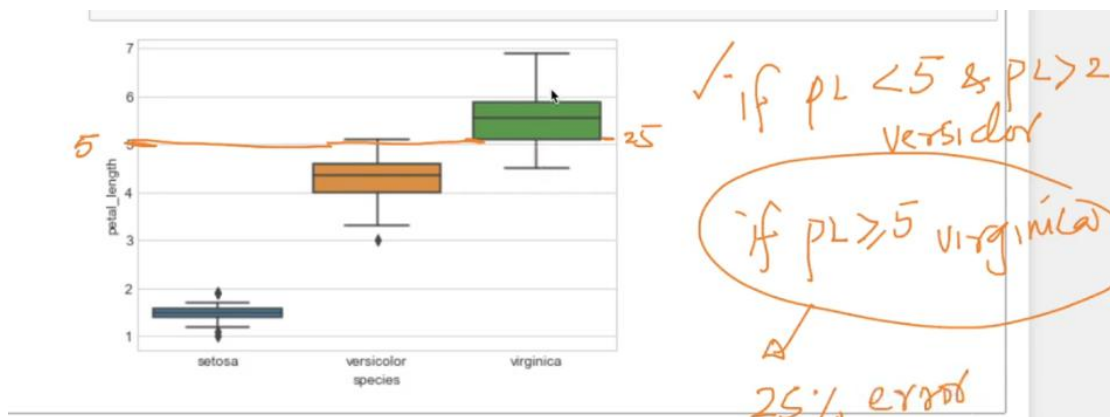


```
#Box-plot with whiskers: another method of visualizing the 1-D scatter plot
# The Concept of median, percentile, quantile.
# How to draw the box in the box-plot?
# How to draw whiskers: [no standard way] Could use min and max or use other
# IQR like idea.
```

```
#NOTE: IN the plot below, a technique called inter-quartile range is used in
#Whiskers in the plot below do not correspond to the min and max values.
```

```
#Box-plot can be visualized as a PDF on the side-ways.
```

```
sns.boxplot(x='species', y='petal_length', data=iris)
plt.show()
```



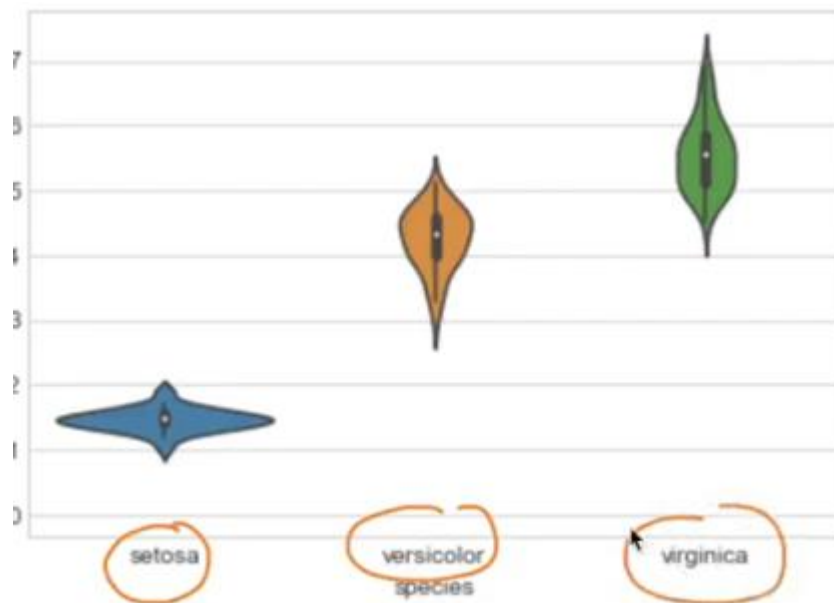
First line is 25% ,50,75%

(3.8) Violin plots

```
|: # A violin plot combines the benefits of the previous two plots
#and simplifies them

# Denser regions of the data are fatter, and sparser ones thinner
#in a violin plot

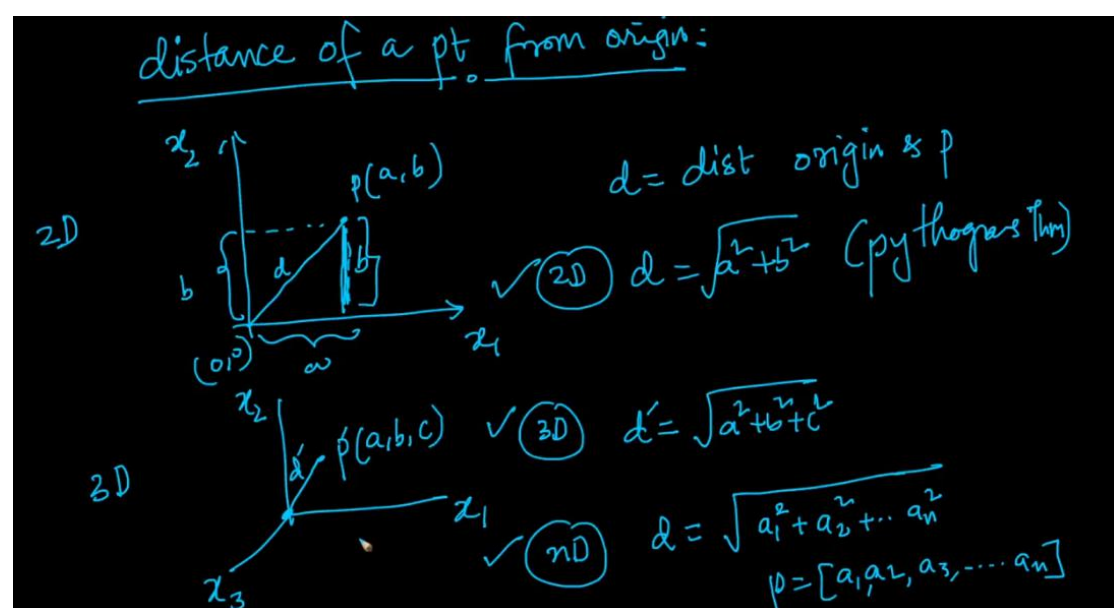
sns.violinplot(x="species", y="petal_length", data=iris, size=8)
plt.show()
```



(3.11) Multivariate probability density, contour plot.

```
8]: #2D Density plot, contours-plot
sns.jointplot(x="petal_length", y="petal_width", data=iris_setosa, kind="kd
plt.show();
```

Darker circle says there is lot of data at that point .



Above image there is distance from origin formula simply derived from pythagorus therom .

dist b/w 2 pts

(2D)

$p(a_1, a_2)$
 $q(b_1, b_2)$
 d
 $b_1 - a_1$
 $a_2 - b_2$

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

(3D)

$p(a_1, a_2, a_3)$
 $q(b_1, b_2, b_3)$

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$$

(nD)

$p(a_1, a_2, \dots, a_n)$
 $q(b_1, b_2, \dots, b_n)$

$$d_{pq} = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

Find Distance between 2 points using above formula .

✓ row vector: $A = [a_1, a_2, a_3, \dots, a_n]$ $(1 \times n)$
 \uparrow rows \uparrow columns

✓ column vector $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$ $(n \times 1)$
 \uparrow rows \uparrow columns

$(b_{n \times 1})$ $(A_{1 \times n})$

{

$A_{m \times n} \rightarrow \text{double}$

$\begin{bmatrix} 1 & 2 & 3 & \dots & n \\ 1 \\ 2 \\ \vdots \\ m \end{bmatrix}$

$m \times n$

Dot Products of vectors :

Addition:

$$a = [a_1, a_2, \dots, a_n]$$

$$b = [b_1, b_2, \dots, b_n]$$

$$c = a + b = [a_1 + b_1, a_2 + b_2, \dots, a_n + b_n]$$

Multiplication: dot product ; ~~cross product~~

$$a \cdot b = a_1 b_1 + a_2 b_2 + a_3 b_3 + \dots + a_n b_n$$

$$a \cdot b = a^T b$$

$$= [a_1, a_2, \dots, a_n] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$1 \times n$ $n \times 1$

$a \cdot b = \|a\| \|b\| \cos \theta$

$\|a\|$ = length of a = dist of a from origin

$\|b\|$ = length of b

θ = angle between a and b

$a = (a_1, a_2)$

$b = (b_1, b_2)$

$\|a\| = \sqrt{a_1^2 + a_2^2}$

$\|b\| = \sqrt{b_1^2 + b_2^2}$

$a \cdot b = a_1 b_1 + a_2 b_2 = \|a\| \|b\| \cos \theta$

$\cos \theta = \frac{a_1 b_1 + a_2 b_2}{\|a\| \|b\|}$

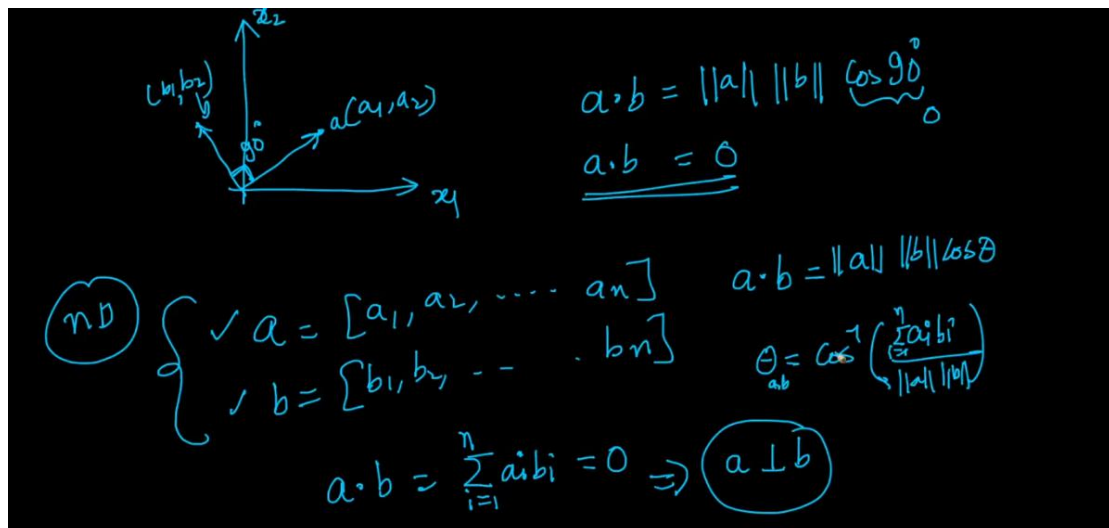
Double line represent length of vector .

We can find length by above formula just by taking square of $a_1^2 + a_2^2$

Dot product of 2 vector means length of a + length of b cos theta .

Then we can easily calculate angle between two vectors using above formula .

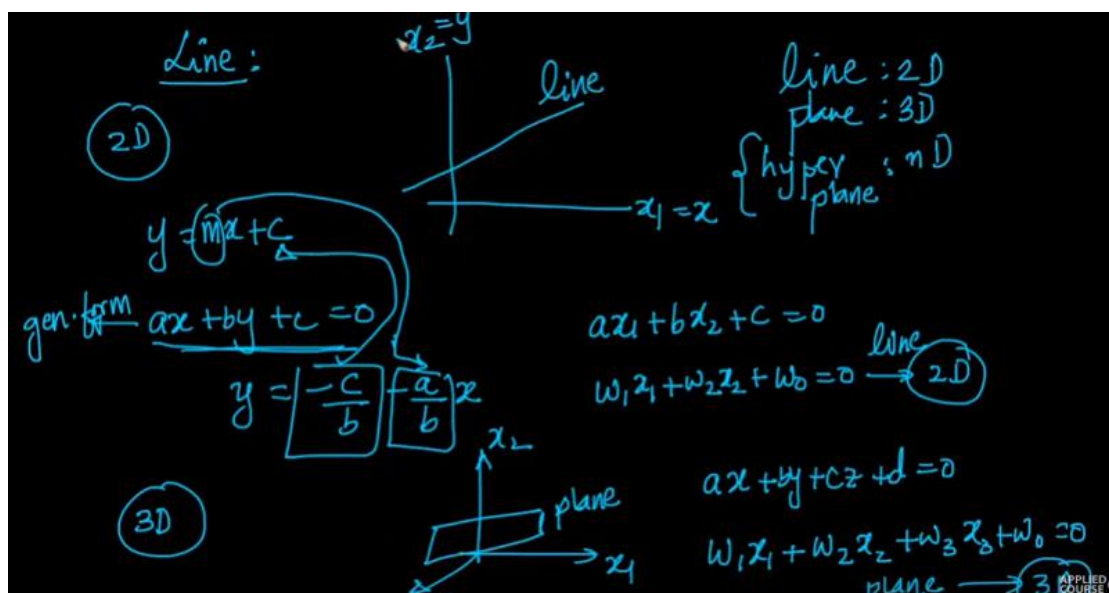
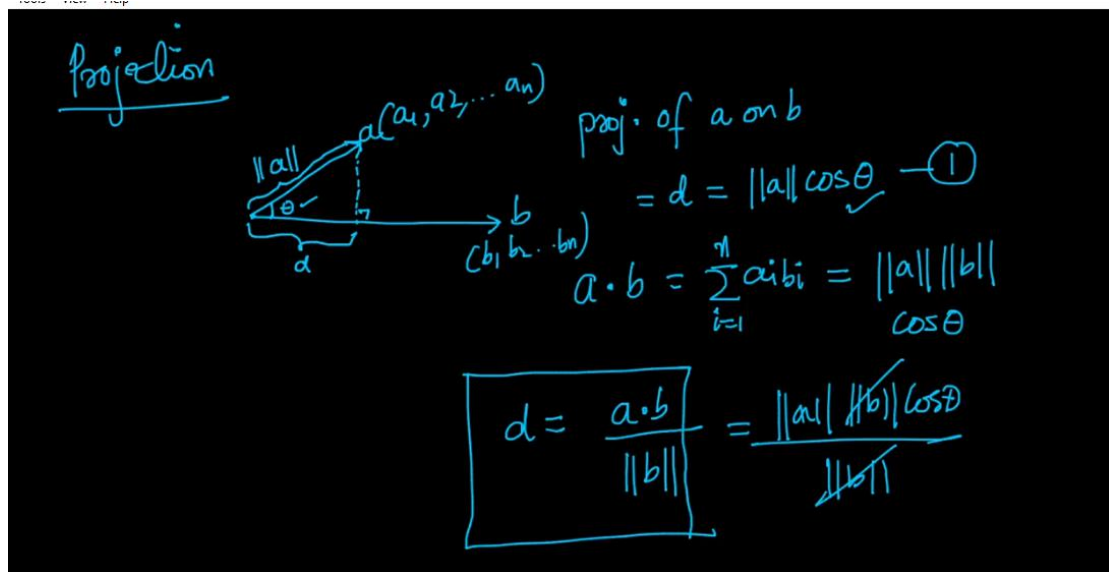
See below fig. If we get angle between two vector then we can say both vectors are perpendicular to each other .



The image shows handwritten mathematical notes on a blackboard. At the top left, a 2D coordinate system with axes x_1 and x_2 is shown. Two vectors, $a(a_1, a_2)$ and $b(b_1, b_2)$, originate from the origin. The angle between them is labeled θ . To the right of the diagram, the dot product formula is written as $a \cdot b = \|a\| \|b\| \cos \theta$, and below it, $a \cdot b = 0$ is underlined. Below these, the general formula for the dot product is given as $a \cdot b = \|a\| \|b\| \cos \theta$. Next to it, the angle θ is expressed as $\theta = \cos^{-1} \left(\frac{\sum_{i=1}^n a_i b_i}{\|a\| \|b\|} \right)$. At the bottom, the condition for perpendicularity is derived: $a \cdot b = \sum_{i=1}^n a_i b_i = 0 \Rightarrow a \perp b$, where $a \perp b$ is circled. To the left of this derivation, the vectors are defined as $a = [a_1, a_2, \dots, a_n]$ and $b = [b_1, b_2, \dots, b_n]$, with a circled 'nD' indicating n-dimensional space.

Dot product is calculated same way for n dimension vectors .

Projection of a on b calculated by drawing line from a to b and then use same distance formula.



η -dim $\textcircled{w_0} + \underbrace{[w_1, w_2, \dots, w_n]}_{1 \times n} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 0$

$\textcircled{2}$

$\underline{w}_{n \times 1} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

$\underline{x}_{n \times 1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

$\Pi: \quad \textcircled{w_0} + \underline{w}^T \underline{x} = 0$

$\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

APPLIED

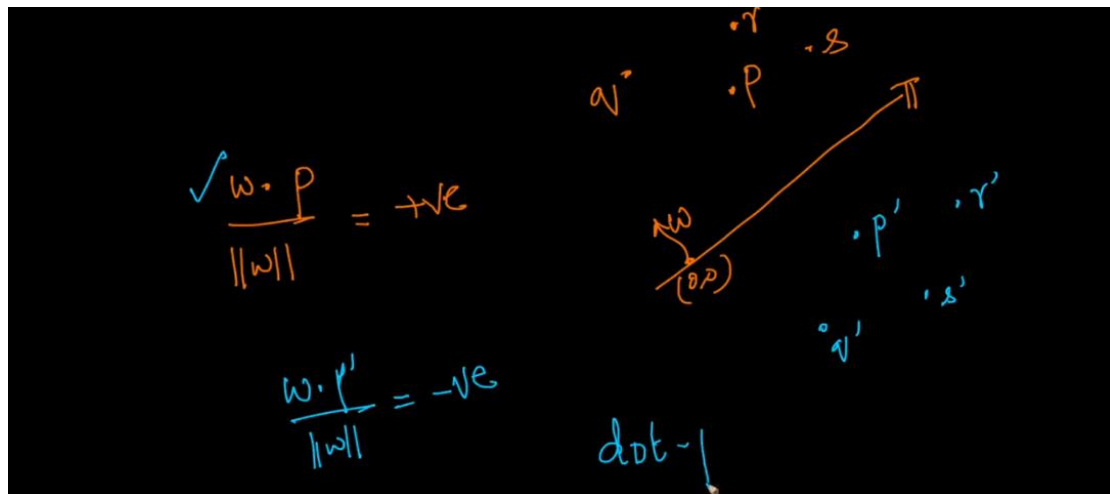
If both vector are column we take transpose of one for taking dot of two vectors.

$d = \frac{\underline{w} \cdot \underline{P}}{\|\underline{w}\|} = +ve$

$d' = \frac{\underline{w} \cdot \underline{P'}}{\|\underline{w}\|} = -ve$

We have a plane on one side there is point p and w also in same direction so angle between both of them less than 90° so it will be positive .

Similarly for p' angle between w and p' will be greater than 90° so d will be -negative .



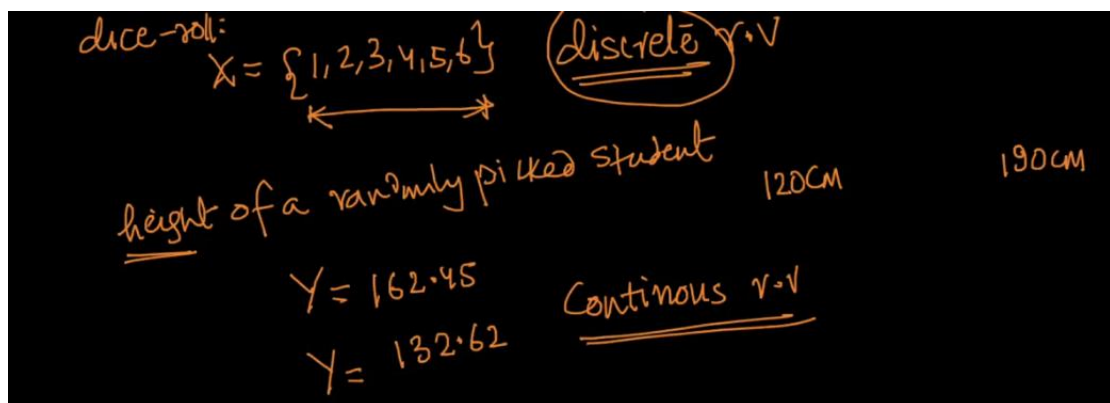
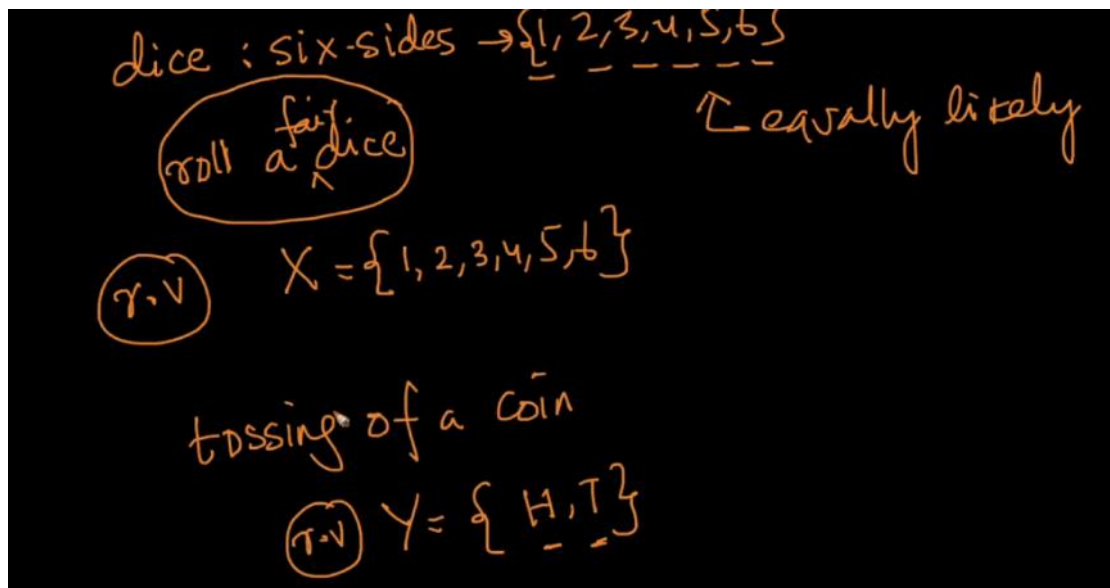
In above image we can see that p, q, r points are on one side and p', q', r' are on other side .

Means we can classify flower from one with another with the help of dot product and with the help of angle we can say where point lies .

Probability :

Random variable take all the possible values denoted by capital letter .

In below image X and Y are the random variables .

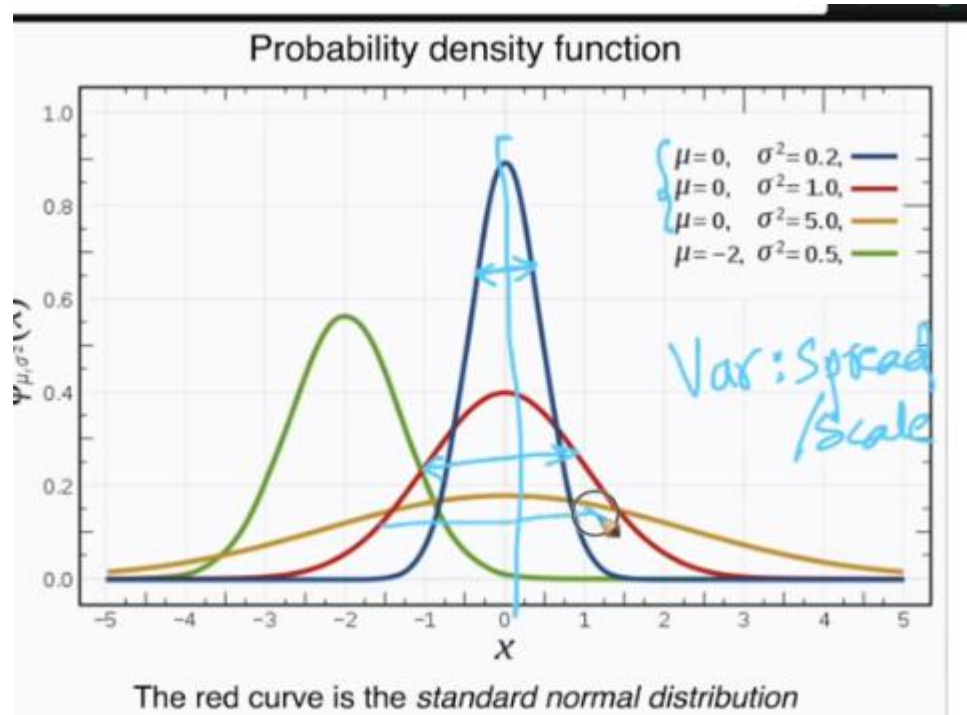


See above image for more understanding of types of random variables .

Gaussian :

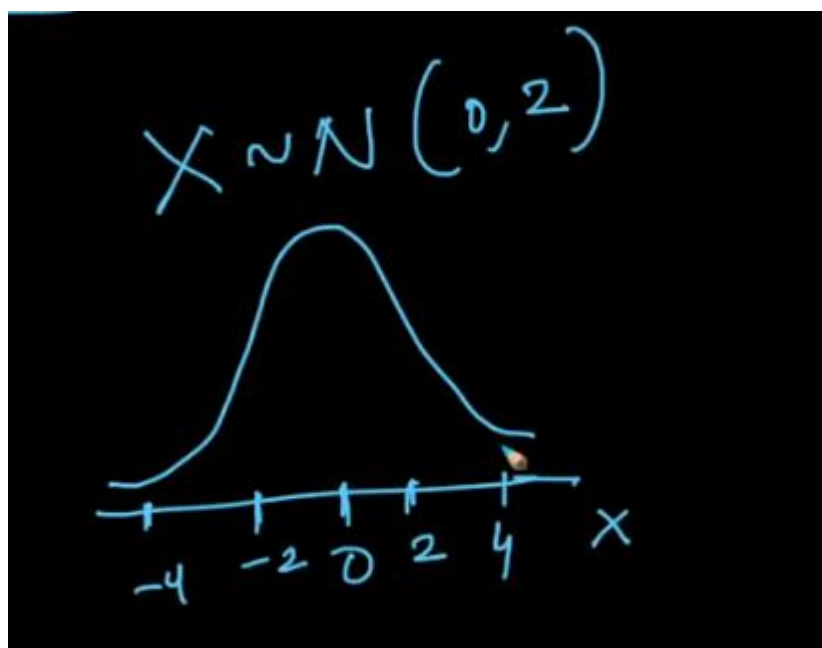
See below fig. Mu is a mean and sigma is variance .

As sigma increases PDF will also increase so our graph will wide vice versa .



In PDF peak is the value of μ .

Lets see below image



0 is mean that's why our graph will be high peak at 0 .

$$\begin{aligned}
 &X \sim N(\mu, \sigma^2) \\
 &P(X=x) = P(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \\
 &\quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \\
 &\quad \sigma=1 \quad \quad \quad \text{160} \quad \quad \quad \text{exp}\{x\} = \underline{\underline{e^x}} \\
 &\text{Let } \mu=0, \sigma^2=1, \sigma=1 \\
 &P(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\}
 \end{aligned}$$

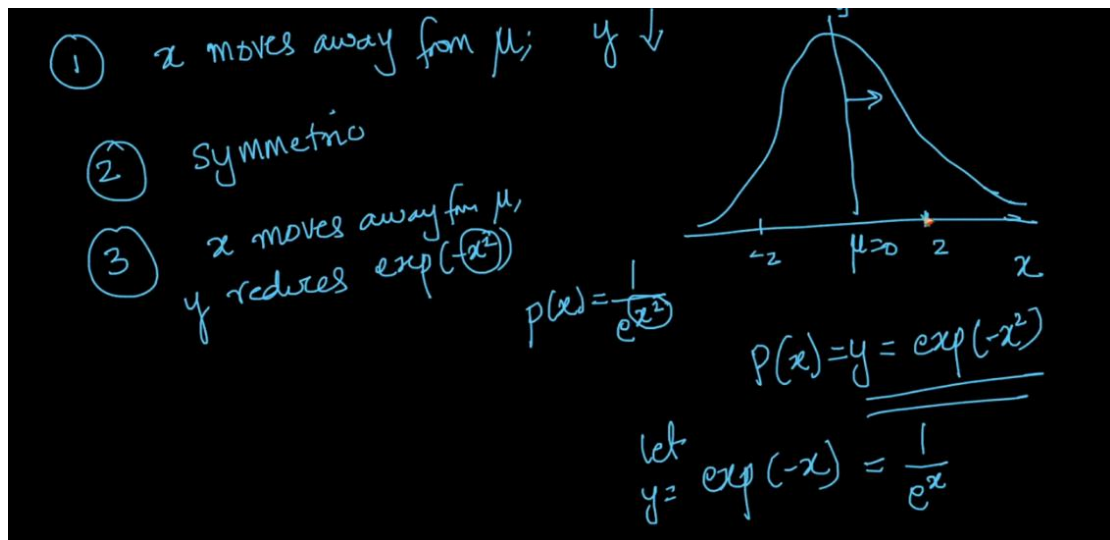
See above image small x is height . we are calculating probability of height of 160 .

Lets assume $\mu = 0$ and Variance = 1 after simplify above equation we got the equation at the bottom of image .

$$\begin{aligned}
 &P(x) = \frac{\text{const}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\} = y \\
 &P(x) = y = \exp(-x^2) \\
 &\quad \quad \quad \begin{array}{l} x \uparrow \quad -x^2 \downarrow \\ \exp(-x^2) \downarrow \end{array}
 \end{aligned}$$

After removing constant our function very simple .

See as x increases y decreases function is symmetric because of negative sign so for example we take $+2$ or -2 result will be same that's why it is symmetric .

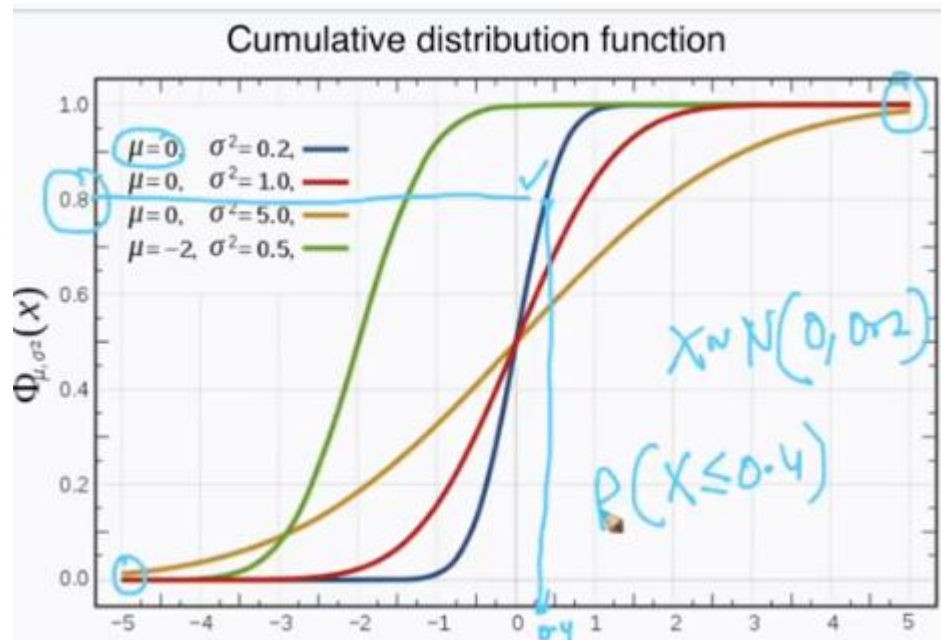


Above is conclusion of Gaussian . As x increases y reduce very very fast.

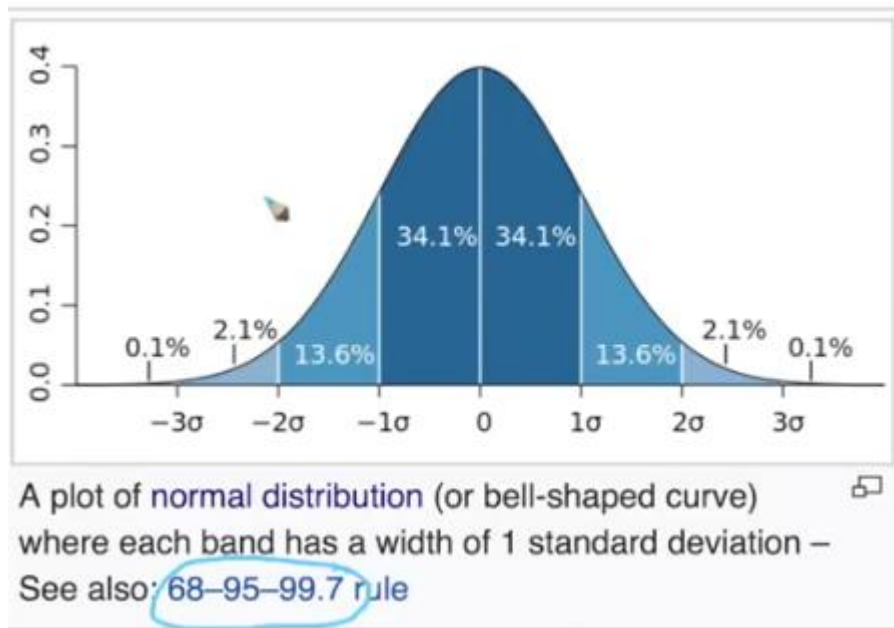
$$y = \exp(-x^2)$$

$x = 0$	$y = 1$
$x = 1$	$y = \exp(-1) = \frac{1}{e^1} = 0.3678$
$x = 2$	$y = \exp(-4) = \frac{1}{e^4} = 0.018$
$x = 3$	$y = \exp(-9) = \frac{1}{e^9} = 0.000123$

See above image for change in small x how much Y reduce drastically .



Sky blue line of CDF height indicates that probability of $x < 0.4 = 80\%$



So using above rule we can say between -1 SD and +1 SD 68 % data we can find .

Standard normal Variable (Z)

① $Z \sim N(0, 1)$
 $\mu = 0$
 $\sigma^2 = 1$

② Let $(X) \sim N(\mu, \sigma^2)$
 (p_L) $\rightarrow [x_1, x_2, \dots, x_{50}]$
 μ, σ^2

Standardization:

$x'_i = \frac{(x_i) - \mu}{\sigma}$ $i=1, 2, \dots, 50$

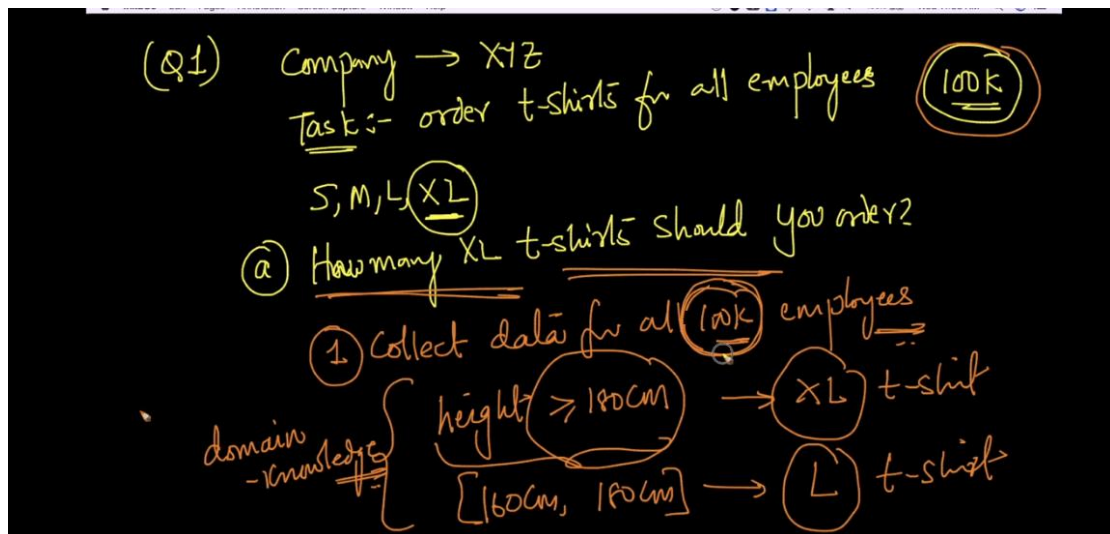
$x'_i \sim N(0, 1)$
 \rightarrow Standard normal variable

→ 95% of x'_i 's $-2 \leq$
 → 68% of x'_i 's lie b/w -1 & 1

Here we are standardizing random variable between -1 to +1 .

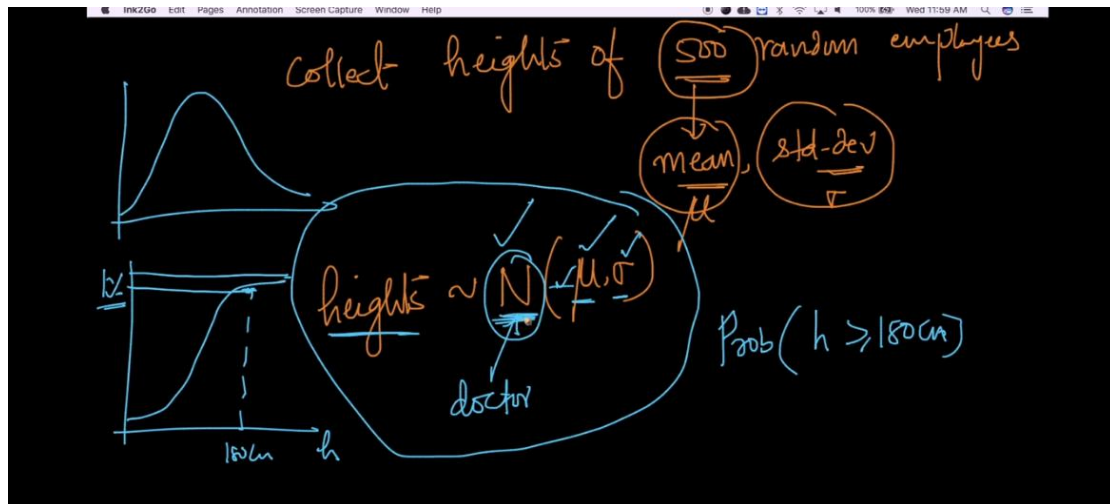
By doing this we can make random variable data standardize so we can easily say between -1 to +1 68 % of data lies .

What is the use of Distribution :

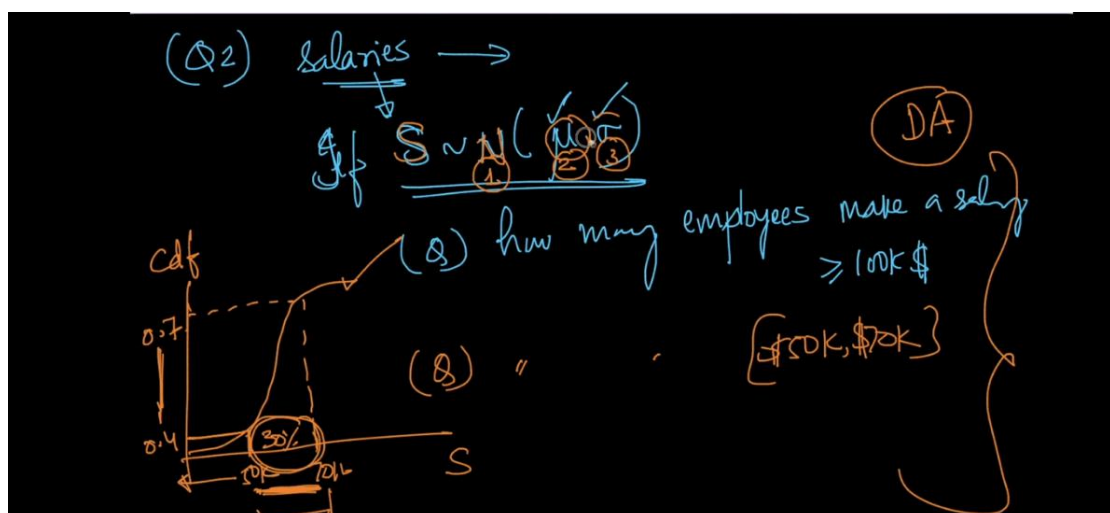


See above image example for buying T shirts for 100k employee in company .

First method is to collect data for size from 100k people but its impossible .



So we collect height random people data then we calculate mean and SD. and some one expert said height data is normal distributed then we can simply draw CDF as in above image and we can only 1% people have height > 180 cm so we can buy XL T-shirts 1% of 100k.

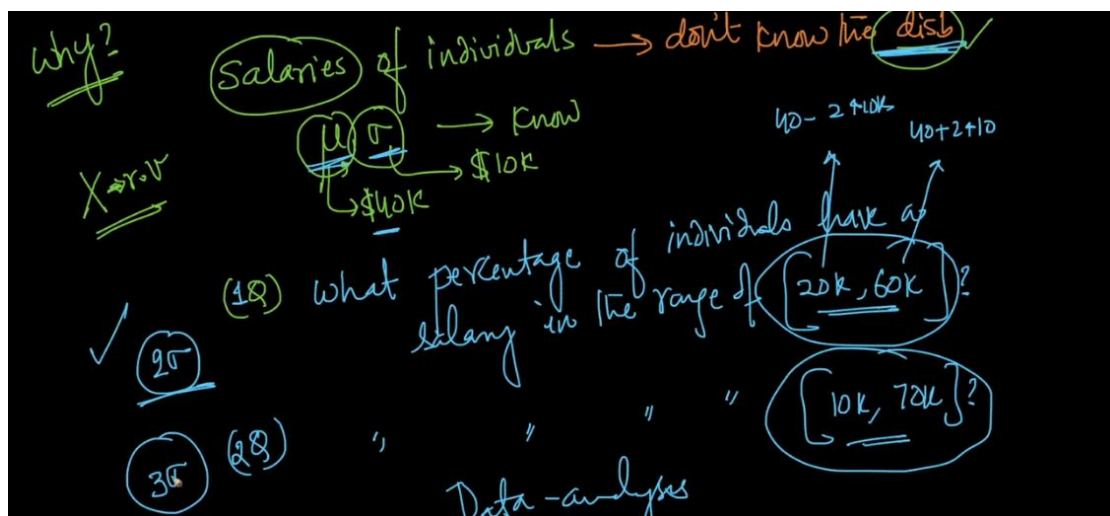


Another example is salary normal dist ad we want to know how much people have salary between \$50k - \$70k we can easily tell using CDF refer above image .

But how would I know salary or any data is normally dist or not ?

That's why QQ plots are used to determine distribution of data .

If data is not Gaussian distributed we cant use CDF .



Now I have only mean and SD I don't know distribution can I find answer of question in image .?

Yes we can solve using Chebyshev's inequality :

Salaries: $\mu = 40k$, $\sigma = 10k$
(1Q) $[20k, 60k]$
 $20k = \mu - 2\sigma$
 $40k = \mu$
 $60k = \mu + 2\sigma$

$P(\mu - 2\sigma \leq X < \mu + 2\sigma) > 1 - \frac{1}{k^2}$

$P(20k < X < 60k) > 1 - \frac{1}{2^2}$

$P(20k < X < 60k) > 0.75$

See blue mark area that is the formula so by putting salary data in above formula we can say that probability of salary between 20 to 60 > 75 % .

For Continuous Random Variable we use PDF and

For Discrete Random Variable we use PMF
(Probability Mass Function)

Uniform Distribution :

Unlike Gaussian here we have a,b, n parameters .


Below example of dice . so $= \{1,2,3,4,5,6\}$ is our data .

A = 1

$$B = 6$$

$$N = b - a + 1 = 6$$

a=1
b=b
n=b → # outcomes



Notation	$\mathcal{U}\{a, b\}$ or $\text{unif}\{a, b\}$
Parameters	$a \in \{\dots, -2, -1, 0, 1, 2, \dots\}$ $b \in \{\dots, -2, -1, 0, 1, 2, \dots\}, b \geq a$ <i>let</i> $n = b - a + 1$
Support	$k \in \{a, a + 1, \dots, b - 1, b\}$
pmf	1

Code :

```
# Sample 30 points randomly from the 150 point dataset
n=150;
m=30;
p = m/n;

sampled_data = [];

for i in range(0,n):
    if random.random() <= p:
        sampled_data.append(d[i,:])

len(sampled_data)
```

Handwritten notes:
tu(a,1) (circled)
random.random() (circled)
sampled_data.append(d[i,:]) (circled)

Bernouli and Bionomial Distributions .

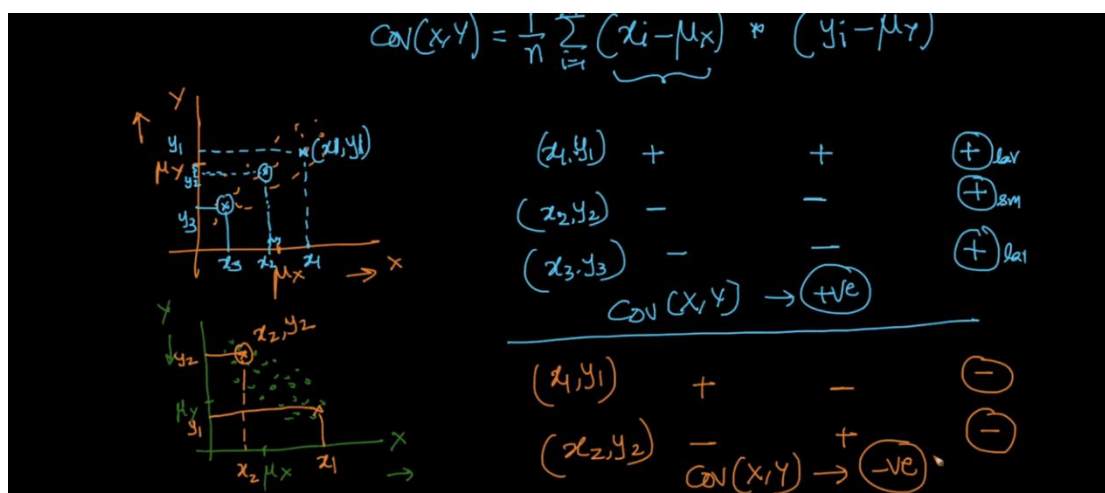
Co-variance :

$$\text{COV}(X, Y) = \frac{1}{n} \sum_{i=1}^n \{x_i - \mu_x\} * (y_i - \mu_y)$$

$$\text{Var}(X) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x) * (x_i - \mu_x)$$

$$\checkmark \text{COV}(X, X) = \text{Var}(X)$$

$$\begin{aligned} \text{COV}(X, Y) &= +ve & x \uparrow, Y \uparrow \\ \text{COV}(X, Y) &= -ve & x \uparrow, Y \downarrow \end{aligned}$$



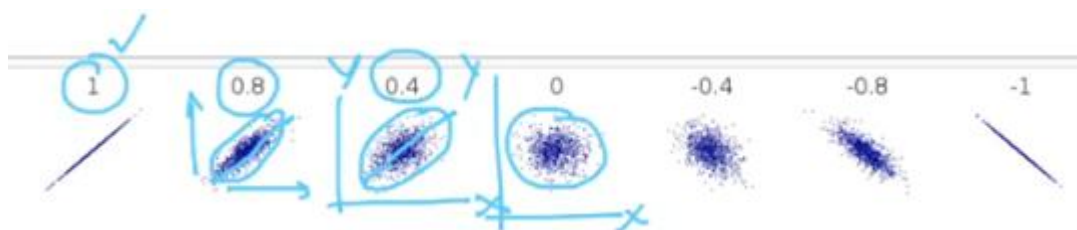
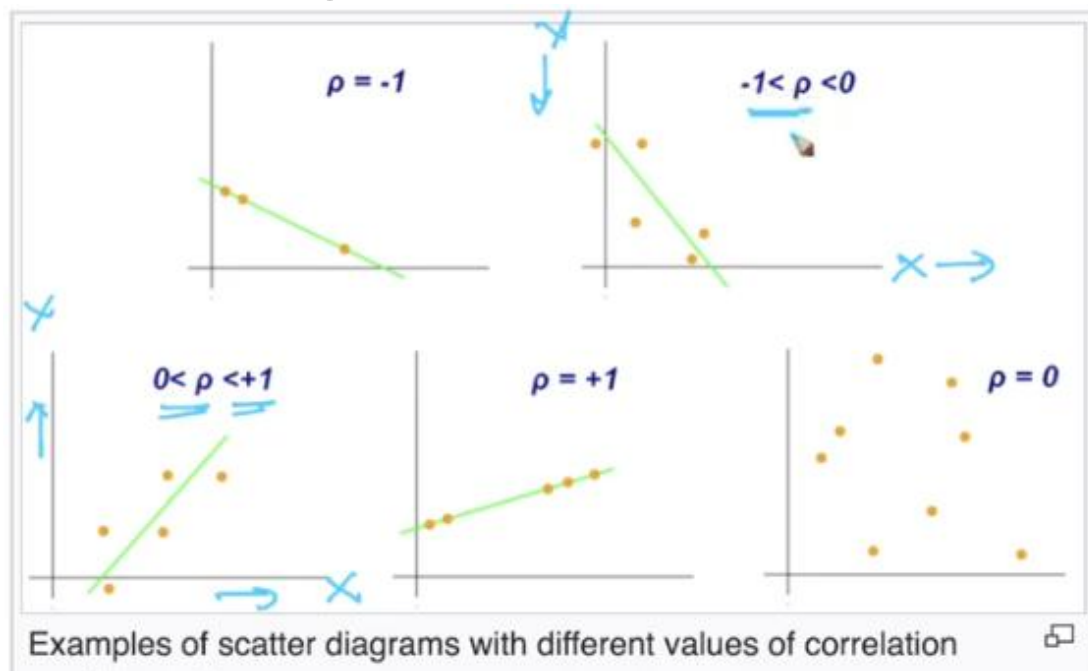
Suppose x is height in inch and Y is weight in Kg .Now if I just change units of height and weight from inch to cm and kg to lbs my co-variance may not be same this is very big disadvantage of Co-variance .

But we want to fix it so we use Pearson coefficient.

Also in Co-variance as x increase y increase then it means Positive Co-variance but how much Positive we don't know this is also a drawback .

Pearson Coefficient :

See below image how PCC lies between +1 to -1



PCC only cares about whether there is linear relation in data or not .

If yes that means data lie perfectly on line so PCC will be 1 at that time .

Now 3rd type is PCC can not handle non linear data relation and this is drawback of PCC .



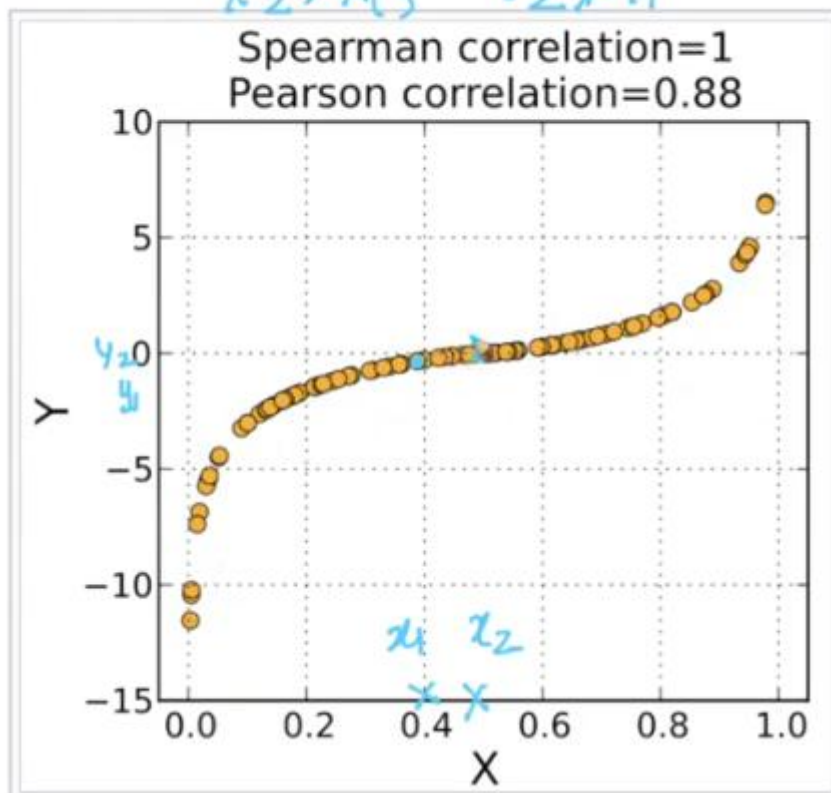
monotonically
non-dec
 $X_2 > X_1 ; Y_2 \geq Y_1$
 $X_2 > X_1 ; Y_2 > Y_1$

on
after
the

on
king of

e

tion



Spearman rank - corr. coeff (r)

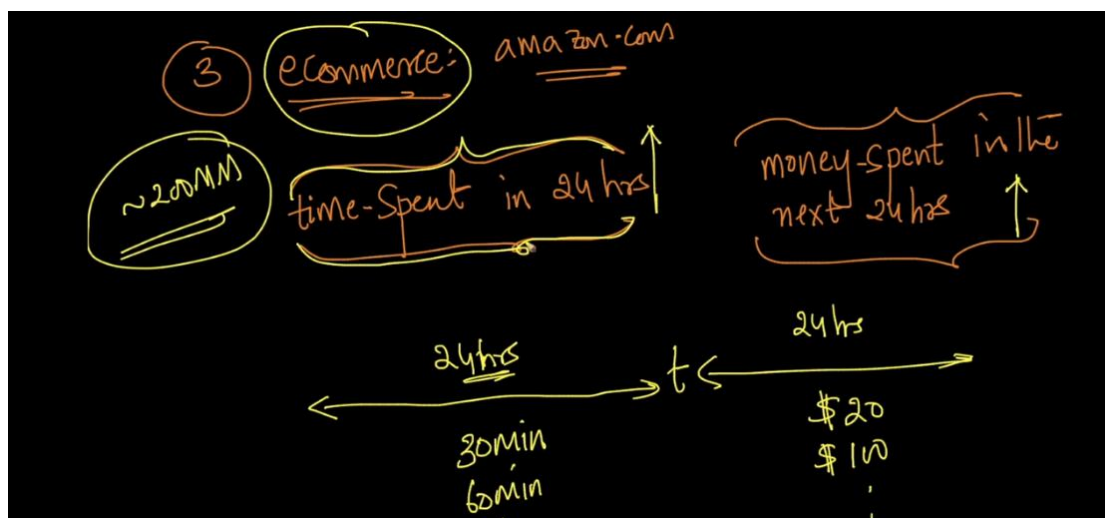
$P_{x,y} \rightarrow$ linear relationship

$r = P_{x,y}$

	x	y	r_x	r_y
s_1	160	52	4	3
s_2	150	66	2	4
s_3	170	68	6	5
s_4	140	46	1	1
s_5	158	51	3	2

Above SRC will remove non linear problem of PCC it will take PCC on rank of x , and Y it will not care about where data is linear or not ..

Use of Correlations :



See above amazon example as new user increase on a website sell also increase . so this relation help amazon to do some thing so number of new users on website will increase .

$\{x_1, x_2, \dots, x_{10}\}$
 $\{180, 162, 158, 172, 168, 150, 171, 183, 165, 176\}$ → heights of people in cm
POINT ESTIMATE of $\mu = \frac{1}{10} \sum_{i=1}^{10} x_i = 168.5 \text{ cm}$ ✓
 $\text{C.I.} \rightarrow \mu \in [162.1, 174.9]$ with 95% probability
 pop-mean Interval Confidence

Confidence Interval :

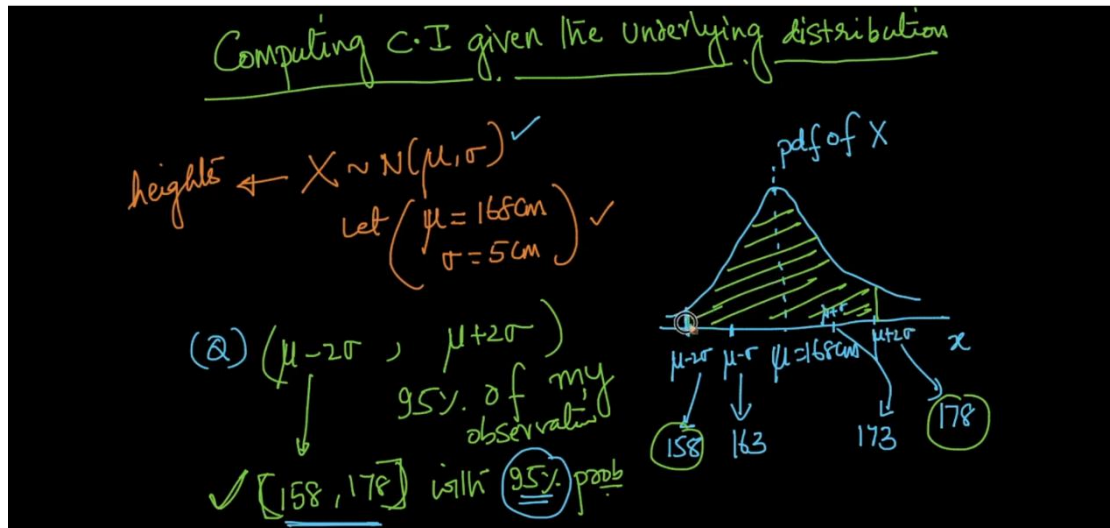
Normal means we calculate by taking average of all points lets consider example in above image of height data for 10 students ..

That's called Point Estimate .

Another way is to give interval with confidence like I am 95% confidence population mean will lie between [162,171] with 95 % confident Probability .

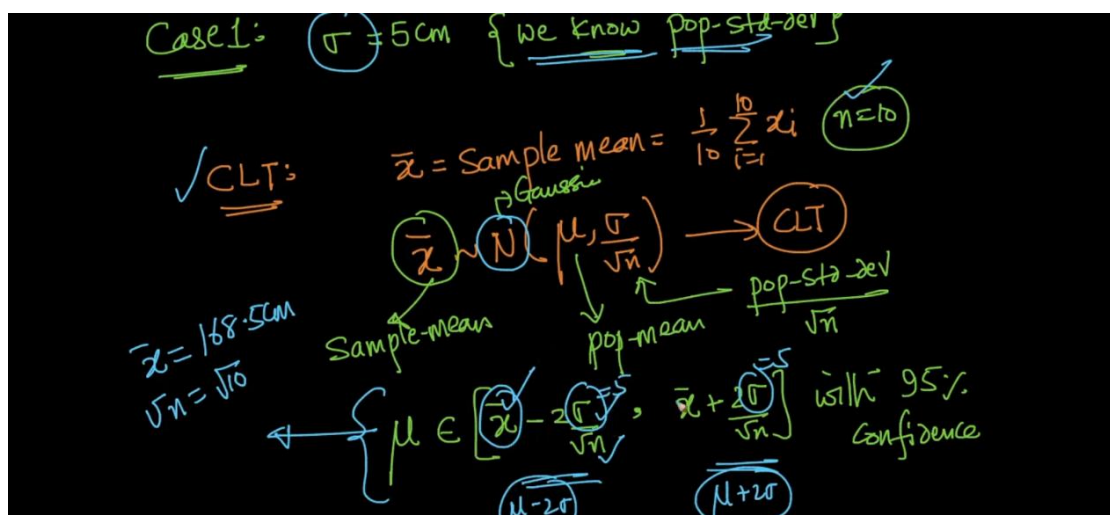
How can we calculate Probability of interval .

Lets consider below image .



Here we assume Gaussian dist so as per mean we draw PDF plot from that we can calculate interval ..

If we know mean and SD of Random variables we can easily Find confident Interval Using Central limit T CLT .

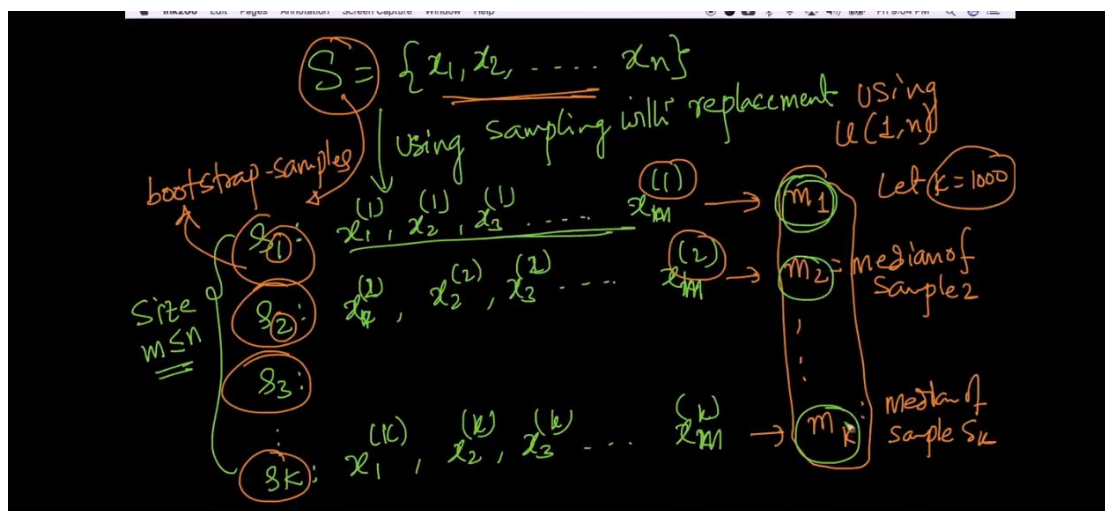


But If we don't know SD then we Use T distribution .

Confidence Interval Using Bootstrapping :

Suppose I have only samples of data I don't have anything like mean, SD .

Now I want to find CI using median of sample so for that we can use Bootstrapping method .



See above image we have S sample data .

From that we create k samples new sample code available in notebook .

P - Value and Hypothesis :

Lets see below image suppose we have 2 class data of students height .

$M1$ = mean of class 1

$M2$ = mean of class 2

Now $m_1 - m_2 = 0$ means there is no difference in height of students of class1 and class 2 means null hypothesis is true .

Now we want to know what is the probability that even null hypo is true there is difference of 10cm in height of 2 class students .

So we use P value if p value is 0.9 just assume means there is 90% probability we get 10cm difference between 2 class .

If P value is high then we can accept null hypo.

If $p = 0.05$ means only 5 % chance so we reject null hypothesis .

③ p-value: prob. of obs $(\mu_2 - \mu_1)$ if null hyp is true.

assume H_0 is true.

if p-value = 0.9

\Rightarrow prob of 10cm is 0.9 if H_0 is true

cl1 cl2

✓ 50 50 ✓

Hypothesis Testing :

Hypothesis Testing: \rightarrow confusing idea

example 1:

Task: Given a coin, determine if the coin is biased towards heads or not basic prob

$\left\{ \begin{array}{l} \checkmark \text{ biased towards heads: } P(H) > 0.5 \\ \text{not-biased " " : } P(H) = 0.5 \end{array} \right.$

if $p(\text{obs} | H_0) < 5\%$

then H_0 may be incorrect

\downarrow

assumption or H_0 is not true

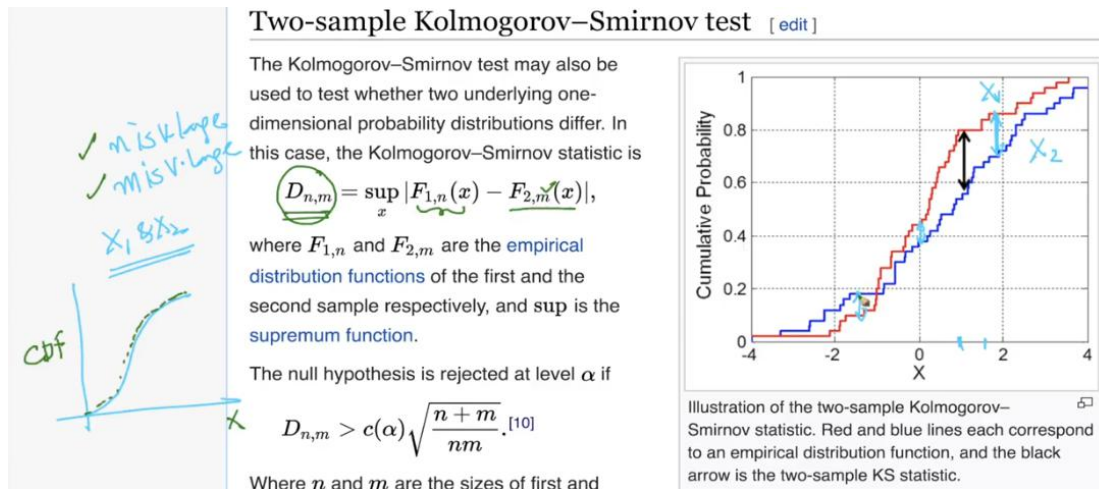
\downarrow

reject $H_0 \Rightarrow$ reject coin is not biased towards heads

\uparrow

accept coin is biased

Lets see obs is observation which we did like some experiment so it will never wrong because we saw that . so our assumption means null hypo may be correct or not we need to see .



KS test :

Suppose we have x_1 and x_2 two variables and we want to see whether both are from same dist or not .

So first we draw CDF for both lets say n = no of observation for X_1 and m = no of observation for X_2 .

If m and n are very large then CDF graph diff will very less shown in above fig . but if we have less amount of data then we find some gap between two graphs .

Now as per above test we take maximum diff between two CDF that difference is called $D_{n,m}$ shown in above image .

Now I want to identify where can I reject or accept null hypo .

Here null hypo is whether two variables are from same distribution .

So lets assume I want p value = 0.05

So as we find $D_{n,m}$ we use another formula of $c(\alpha)$. so just see below image

$$\underline{\underline{D_{n,m}}} > c(\alpha) \sqrt{\frac{n+m}{nm}} \quad [10]$$

Where n and m are the sizes of first and second sample respectively. The value of $c(\alpha)$ is given in the table below for the most common le

α	✓ 0.10	✓ 0.05	✓ 0.025	✓ 0.01	✓ 0.005	✓ 0.001
$c(\alpha)$	1.22	1.36	1.48	1.63	1.73	1.95

Now lets say we assume $\alpha = 0.05$ then check $c(\alpha)$ in look up table .

Just put value in above equation .

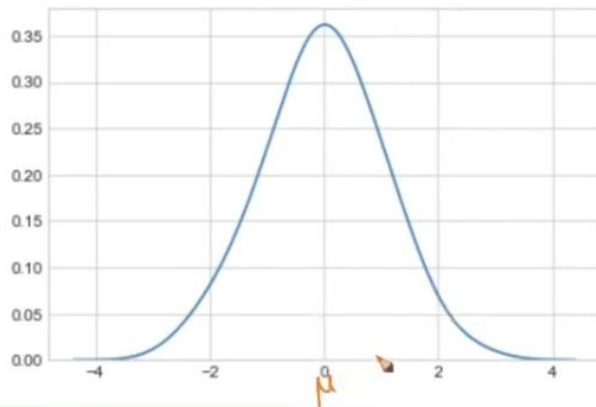
If $D_{m,n} > 0.047$ that means above graph difference is $>$ then just reject null hypo means both data are not same distributed .

$n=1000$
 $m=500$
 $\alpha=0.05$
If $D_{n,m} > 0.047$
then reject H_0
at 0.05 sig. level.

Code For ks Test :

K-S Test

```
In [29]: import numpy as np ✓  
import seaborn as sns ✓  
from scipy import stats ✓  
import matplotlib.pyplot as plt ✓  
  
#generate a gaussian r.v X  
x = stats.norm.rvs(size=1000);  
sns.set_style('whitegrid')  
sns.kdeplot(np.array(x), bw=0.5)  
plt.show()
```



```
30]: stats.kstest(x, 'norm')
```

```
30]: KstestResult(statistic=0.021308397286061931, pvalue=0.75424031453335627)
```

See p value is too high so it is normally distributed .

Hypothesis testing Example :

Task:

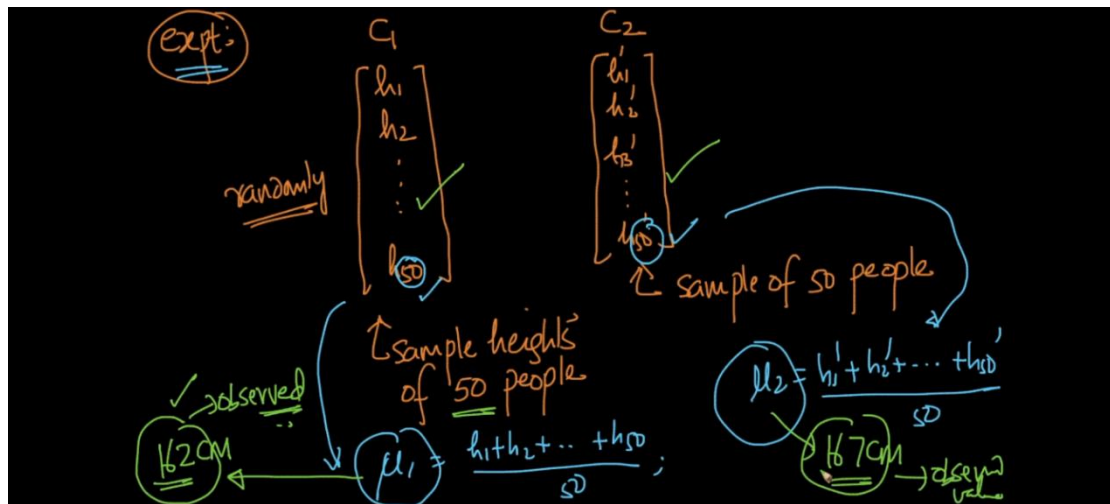
(C₁) $\mu_1 = 1MM$ (C₂) $\mu_2 = 2MM$

determine if the population means of heights of people in these two cities is same or not

μ_1, μ_2 are same/different

population mean \rightarrow sample mean

See Experiment



So we got

Mean 1 = 162

Mean 2 = 167

So difference is our observation which is 5 here

Next step :

test-statistic:

$\mu_2 - \mu_1 = (x) = 167 - 162 \text{ cm} = 5 \text{ cm}$

$\downarrow \quad \downarrow$
 $C_2 \quad C_1$

Null hyp (H_0): There is no difference in population means

compute:

$p(x = 5 \text{ cm} | H_0)$

\downarrow

diff in sample means with sample size of 50

Explain :

✓ $P(\underline{x=5} | H_0) \rightarrow$ compute it \rightarrow next video
prob. of observing a diff. of 5cm in sample mean
heights of sample size 20 between C_1 & C_2 if
there is no population diff in mean-heights

Assume case 1 :

Case 1: $P(\underline{x=5} | H_0) = 0.2 = \underline{20\%}$
 There is a 20% chance of obs a diff of 5cm
 in sample mean heights of C_1 & C_2 (with sample of
 20) if there is no pop mean diff.
 $P(\underline{x=5} | H_0)$
 $P(\text{Obs} | \text{assumption}) = 20\% \rightarrow$ Significant
 \Rightarrow assumption must be true
 \Rightarrow accept H_0 .

Case 2 :

Case 2: $P(\underline{x=5} | H_0) = 0.03 = 3\%$
 $P(\underline{\text{Obs}} | \text{assumption}) = 3\% \rightarrow$ small
 $\rightarrow < 5\%$
 \Rightarrow assumption must be incorrect
 \Rightarrow reject $H_0 \Rightarrow$ accept H_1

We are rejecting null hypo means there is a difference between two city mean .

Now how to compute that 20% and 3 % case above ?

We have following data from that we need to compute p values :

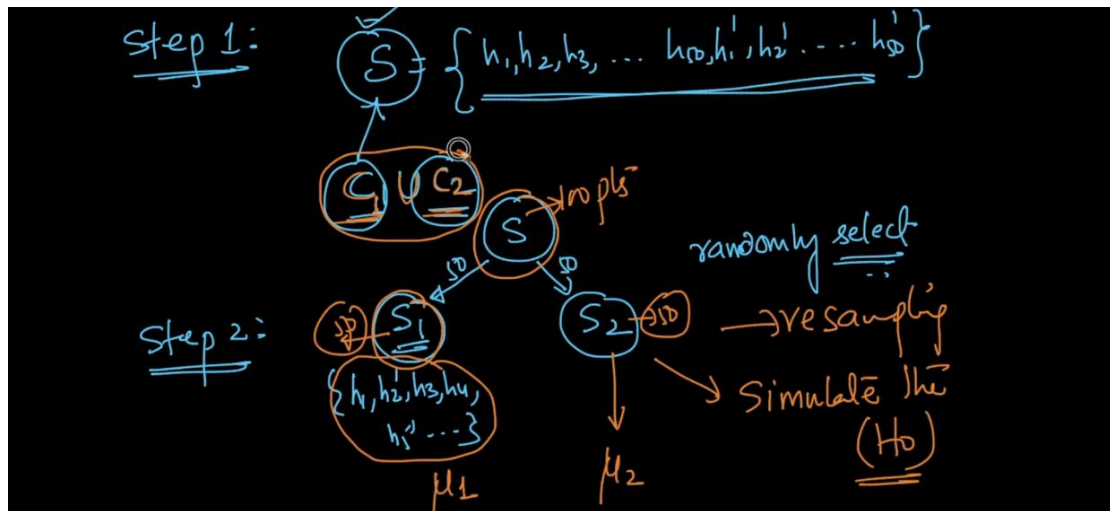
① $z = \frac{-162 + 167}{5} = 1$ ← diff in sample means with sample size of 10

② H_0 : no diff in population means.

③

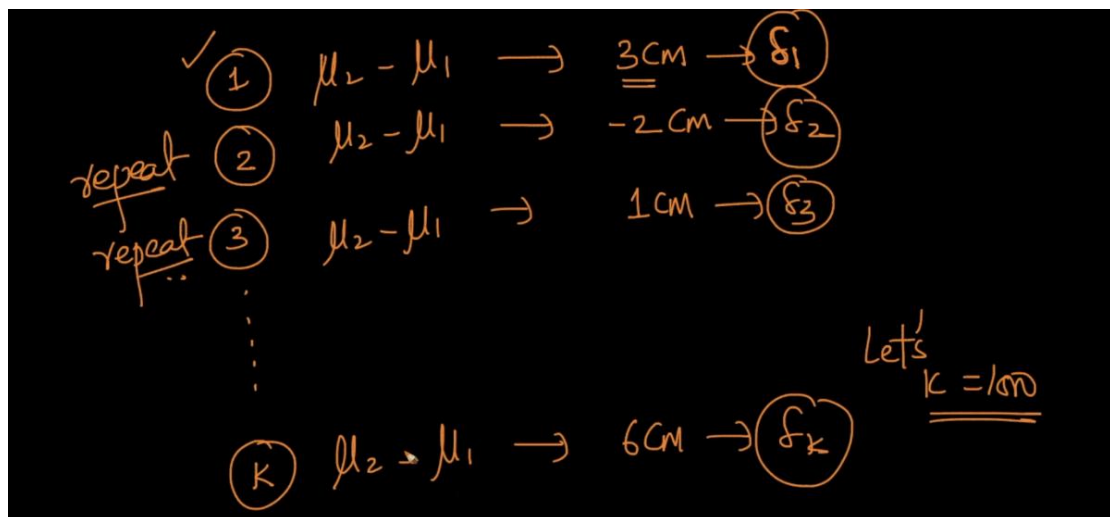
C_1	C_2
$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{50} \end{bmatrix}$	$\begin{bmatrix} h_1' \\ h_2' \\ \vdots \\ h_{50}' \end{bmatrix}$

Step is we combine all data points for both C_1 and C_2 into another big set lets say S so now we have 100 points .

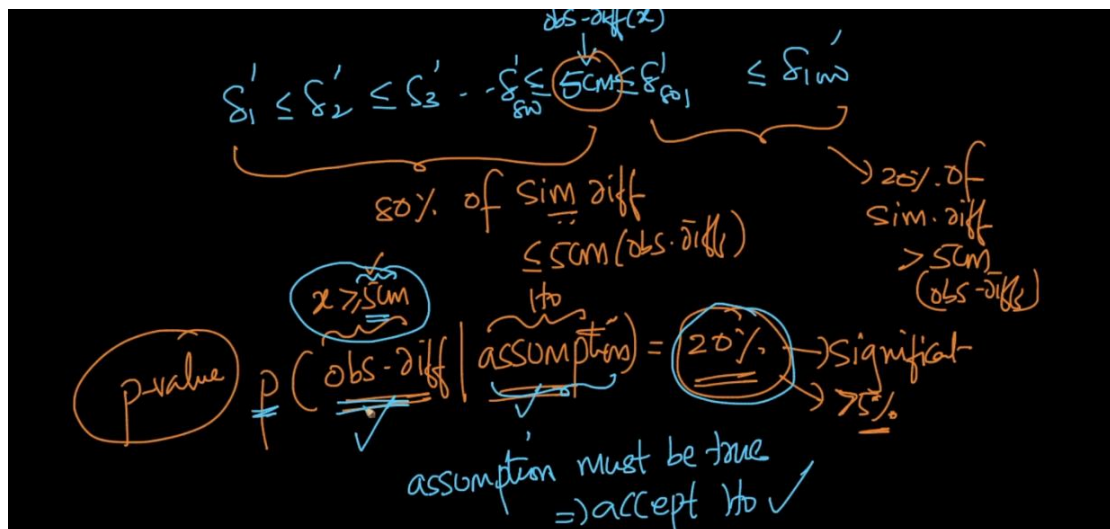


From S we take random 50,50 points lets say new set s_1, s_2 then we compute mean of both and will find difference between them so we get delta 1 .

We repeat same process 1k times so we will get 1k delta ..



Noe next step is sort all 1k deltas

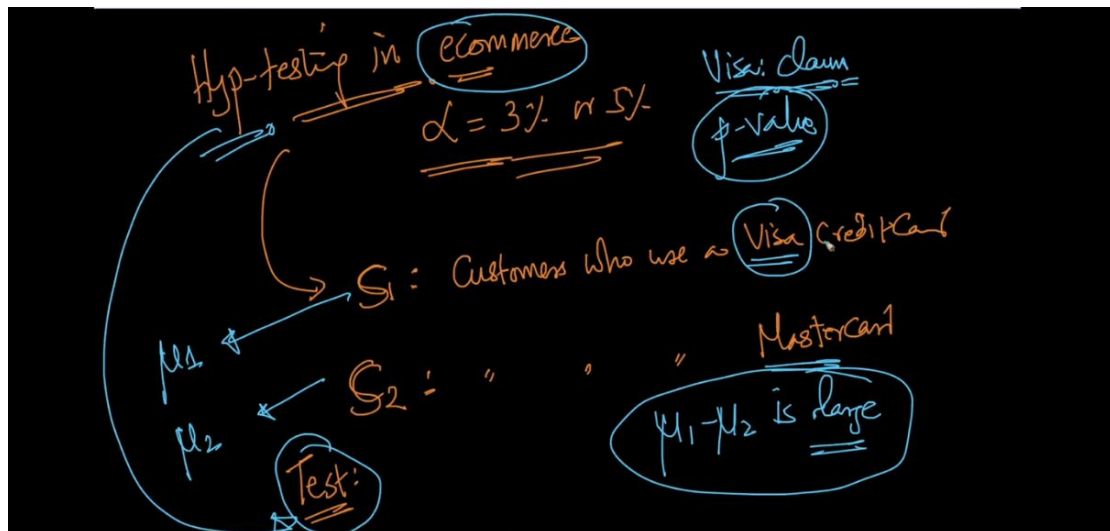


See now we have sorted all deltas so see how much % of delta is $>$ our main observation means 5cm so we get 20% of deltas are above 5cm in above example so our p value = 20% which is too high so we will accept null hypo and reject alternative hypo .

Accept means there is no difference between sample mean of 2 city .

This is called Permutation and Re-sampling test

Example of hypo test :



We can solve lots of different problem using this test .

Now visa says people spend max amount through visa than other cards .

So we can take mean of visa and other card and find p value and we can give probability whether it is true or not ..

Data Pr processing :

Column norm we use to make all our data within 0,1 . for example suppose student height can be measure in inch,feet,cm whatever unit data collect we will normalize all data between 0 , 1 range .

Now lets say we have a column petal length with values[a1,a1 an]

Now from that data we create new data points simply we take

F min = minimum value in above column

F max = maximum value in above column .

Now use below formula :

$$A1' = \frac{a1 - F \text{ min}}{F \text{ max} - F \text{ min}}$$

.

.

.

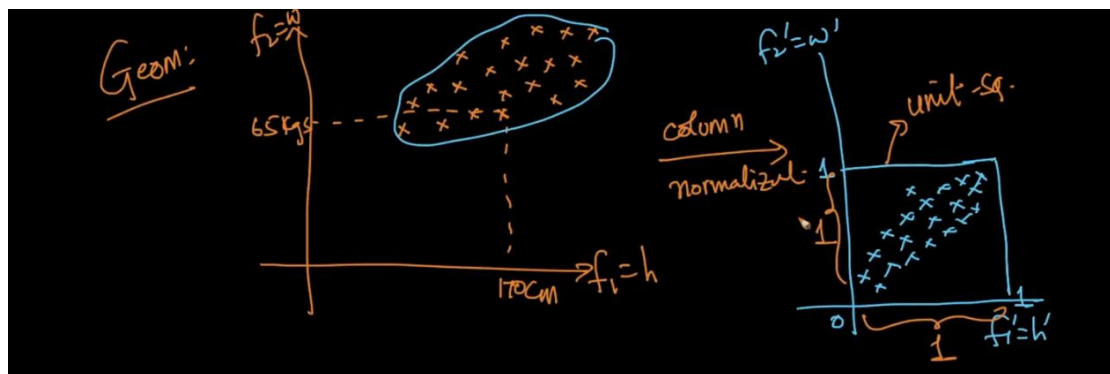
An'

Same way we find new data using above formula and that will lie between 0 and 1 .

This is called column normalization we don't care about what scale data collect we will transform that data into 0,1 range .

Column: $1.2, 1.3, 1.4, 1.9, 1.5$ \rightarrow n -values of f_j
 $a_1, a_2, \dots, a_i, \dots, a_n$
 $\max(a_i) = a_{\max} \geq a_i \quad (i:1 \rightarrow n)$
 $\min(a_i) = a_{\min} \leq a_i \quad (i:1 \rightarrow n)$
 $a'_1, a'_2, a'_3, a'_4, \dots, a'_i, \dots, a'_n$
 $a'_i \in [0, 1]$
 $a'_i = \frac{a_i - a_{\min}}{a_{\max} - a_{\min}}$
 $a'_{\min} = \frac{a_{\min} - a_{\min}}{a_{\max} - a_{\min}} = 0$
 $a'_{\max} = \frac{a_{\max} - a_{\min}}{a_{\max} - a_{\min}} = 1$

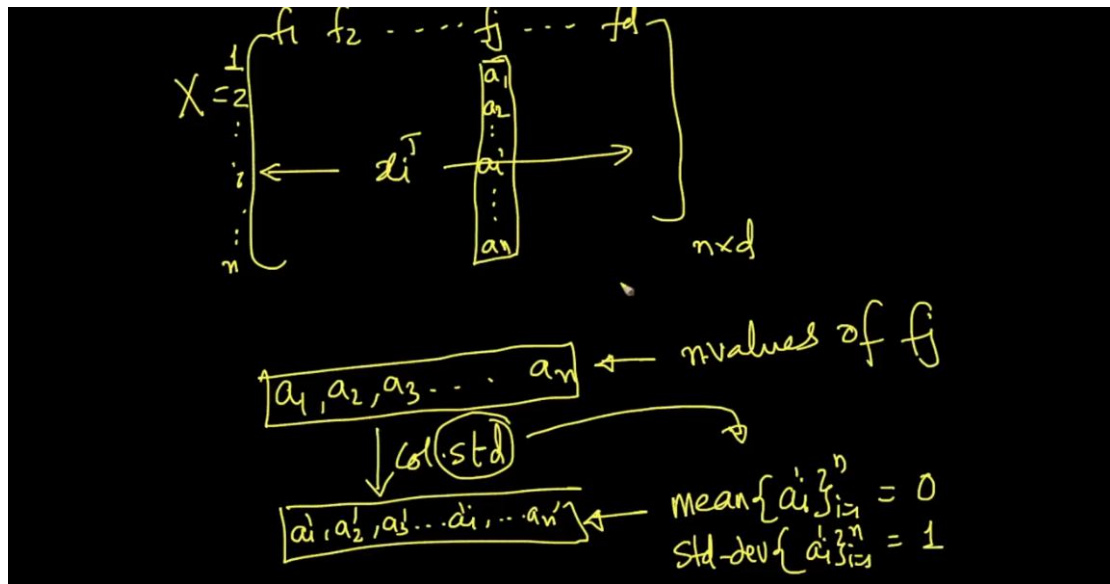
See below image of height example how we transform Geometry .



Here we transform height data into 0,1 range .

Now next Concept :

Column Standardization :

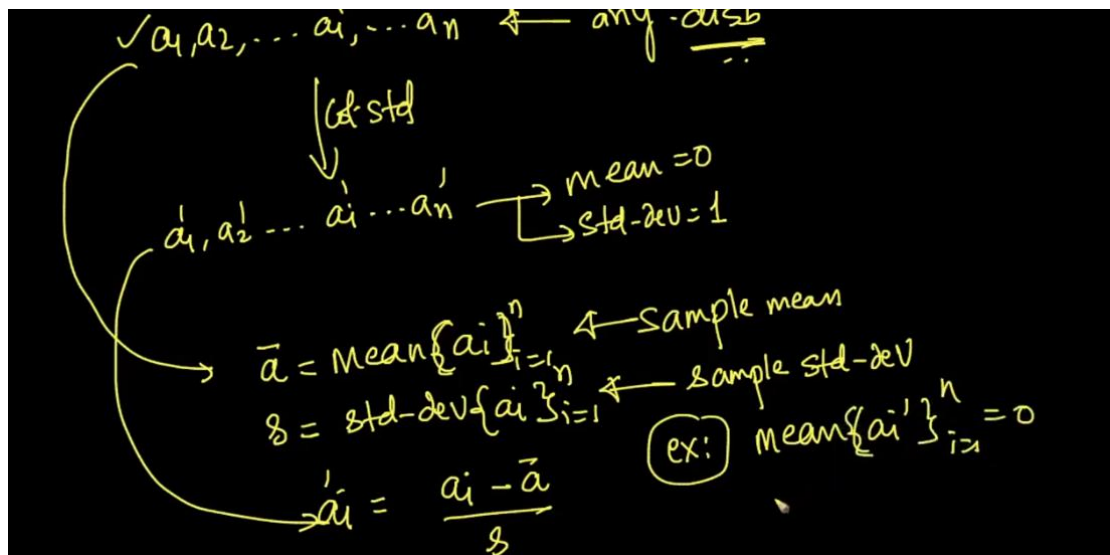


Remember in normalization we make all data points between 0,1 .

Here we will make our new data from existing data lets say I have height data I will create new data from that in such a way that mean of new data will be 0 and SD will be 1 .

See in image also same explain .

See below image there is one formula by using that formula we can do that transformation .

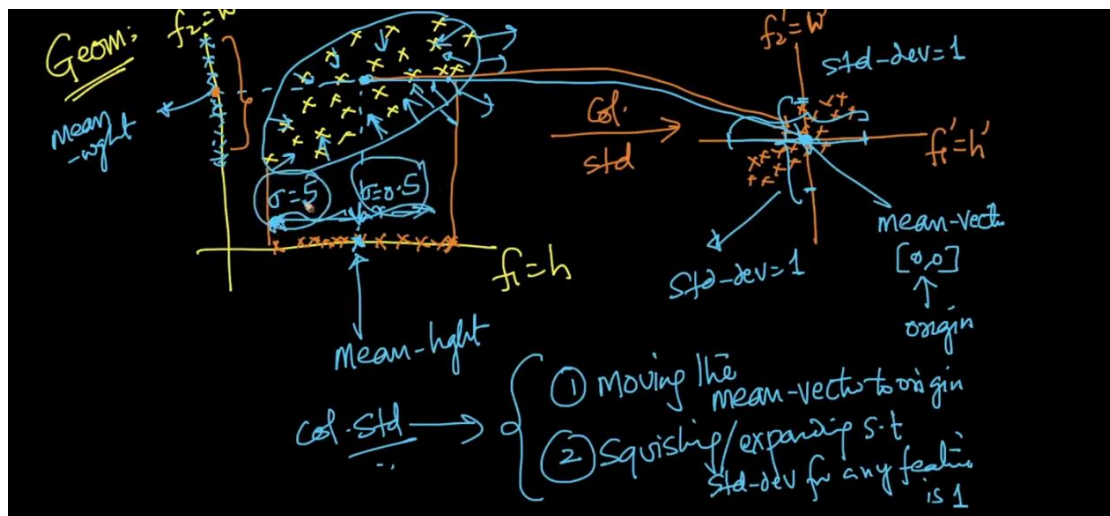


A bar here means mean of original data lets say mean of height column .

Then A' here calculate by subtracting a_1, a_2, \dots, a_n from \bar{a} .

So we get new list of data called A' and if we calculate mean of that list we will get 0 and SD will be 1 .

See below image will say how data will transform from non zero to zero mean .



Co-variance matrix :

$X = \begin{bmatrix} f_1 & f_2 & \dots & f_d \\ x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}$

$n \times d$

Let (X) Col. Standardized \Rightarrow mean $\{f_i\} = 0$
std-dev $\{f_i\} = 1$

avg μ_1 μ_2 μ_d

$$\text{Cov}(f_1, f_2) = \frac{1}{n} \sum_{i=1}^n (x_{1i} - \mu_1) (x_{2i} - \mu_2)$$

$\mu_1 = \text{mean}(f_1)$ $\mu_2 = \text{mean}(f_2)$

Now see above Co-variance formula μ_1 and μ_2 are standardize means they are 0 so we can re write above formula see below image :

$$\text{Cov}(f_1, f_2) = \frac{1}{n} \sum_{i=1}^n x_{i1} * x_{i2}$$

Diagram illustrating the calculation of covariance between two features f_1 and f_2 . The data is organized into two columns, f_1 and f_2 , with rows indexed from 1 to n . The data points are represented by symbols: dots, checkmarks, crosses, and asterisks. The covariance is calculated as the average of the product of corresponding elements in the two columns, $x_{i1} * x_{i2}$.

$$\text{Cov}(f_1, f_2) = \frac{f_1^T f_2}{n}$$

Co-variance matrix is nothing but :

$$S_{d \times d} = X_{d \times n}^T X_{n \times d}$$

if x has been col. std

If x is col standardized .

MNIST data set :

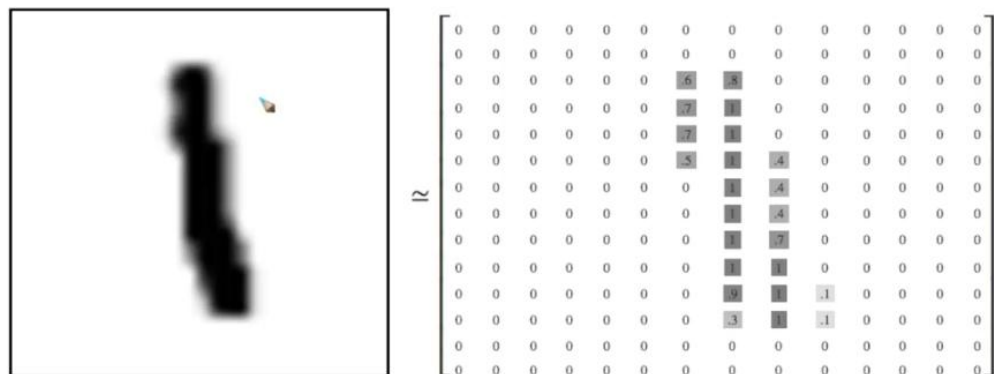
MNIST

MNIST is a simple computer vision dataset. It consists of 28x28 pixel images of handwritten digits, such as:



Now that each digit is a image of 28 * 28 pixel .

So we will convert this images to matrix .



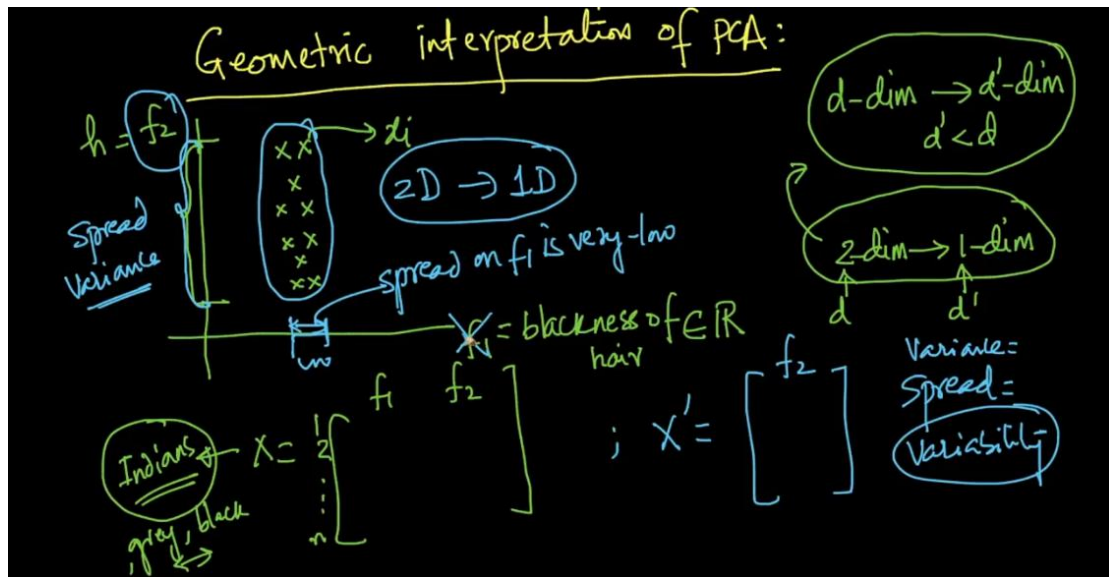
Lets say image of 1 its 28* 28 pixel means 28*28 matrix with pixel values . all the white area is 0 all dark black has value 1 like that .

So now we will convert this matrix into single column called flattening .

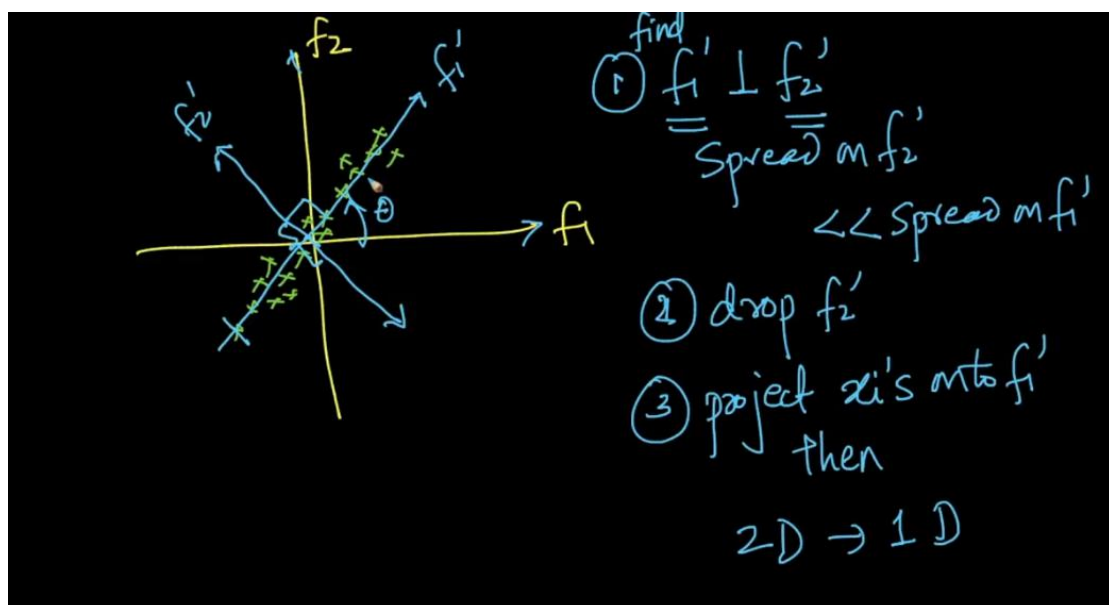
So $28 * 28 = 784 * 1$ column .

So we will make this as 1 row similarly we can add for 60k image so we get final matrix as $60k * 784$.

Now we have 784 features matrix and its impossible to visualize using normal plots so for that we use PCA .



See variance means data spread for f2 is very large than f1 so as data spread is large means that is a important feature so we can not remove that feature . so remove feature f1 and we can convert 2d to 1d data .



- ① $f_1' \perp f_2'$
 $\quad \quad \quad \text{Spread on } f_2'$
 $\quad \quad \quad \ll \text{Spread on } f_1'$
- ② drop f_2'
- ③ project x_i 's onto f_1'
then
 $2D \rightarrow 1D$

② drop f_2'

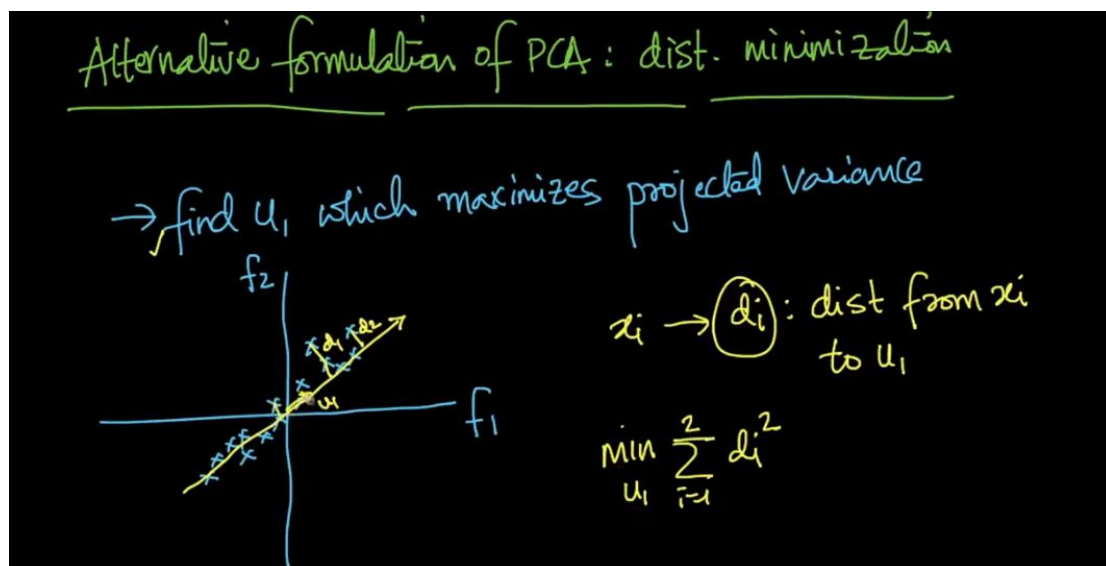
③ project x_i 's onto f_1'
then

$$2D \rightarrow 1D$$

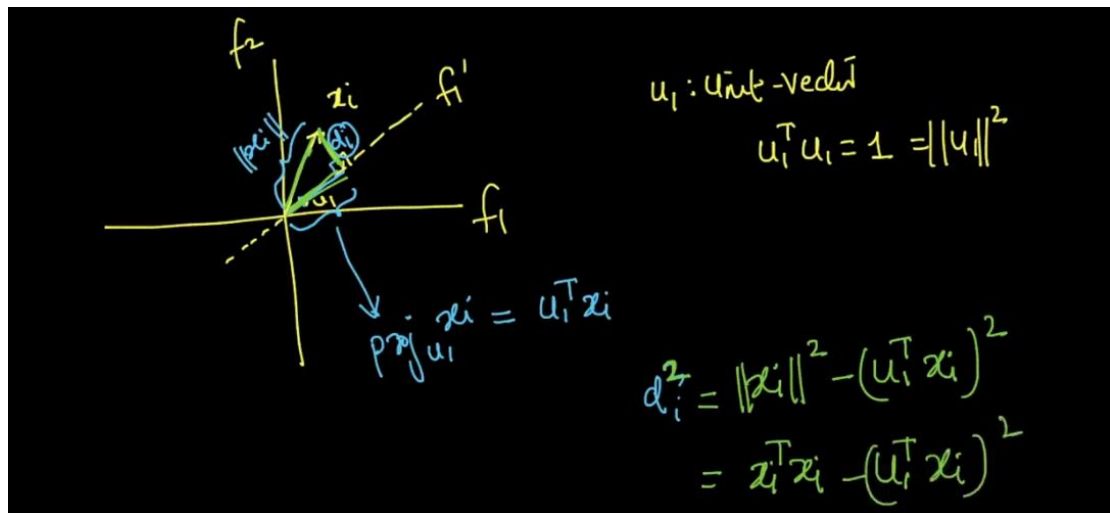
Now if we see yellow lines we cant remove any line means any feature because data spread is high on both lines .

So we just move f_1 and f_2 little by new angle so we found f_1' has too much spread as compare to f_2 so we can drop f_2' in this way we convert 2D to 1D .

So here our job is to find perfect direction of f_1 or f_2 so we get max data spread on one line so we can remove another one .

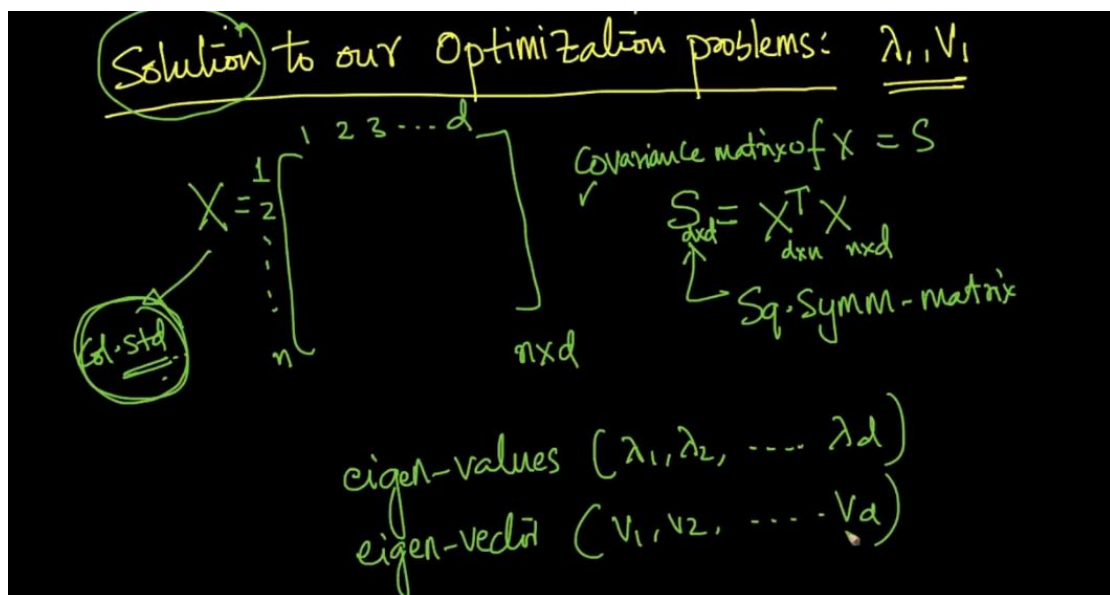


Here want to find min distance from x (data point) to u_1 (direction) .



See above image for finding d square we use simple pythagorus therom .

Solution for optimization problem min distance and Maximize variance for PCA because more variance more data will cover in that features .



Now see above image lets say we have x matrix so we will take co-variance matrix S of X with is X transpose of X very simple .

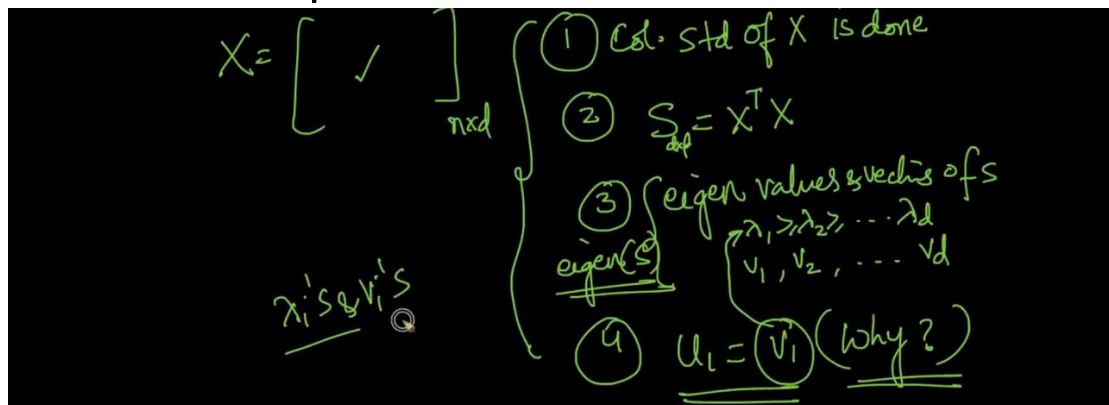
Then simply by using numpy we calculate eigen values and vectors .

E values = lamda

E vectors = v1,v2 ...

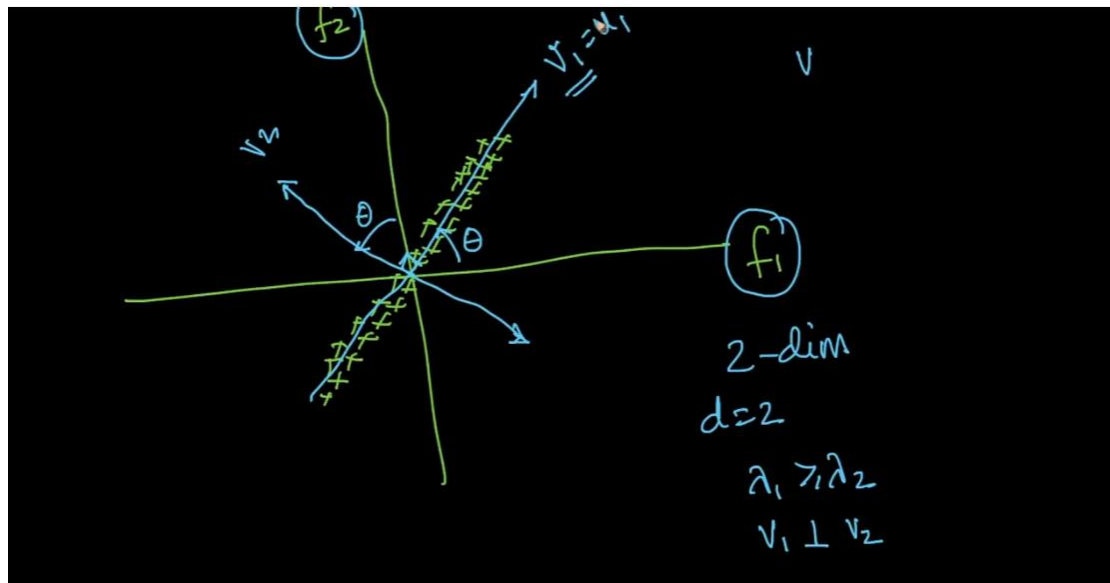
Now after calculate E values and Vectors we see vectors maximum value and that value is nothing but our u1 direction .

See below steps :



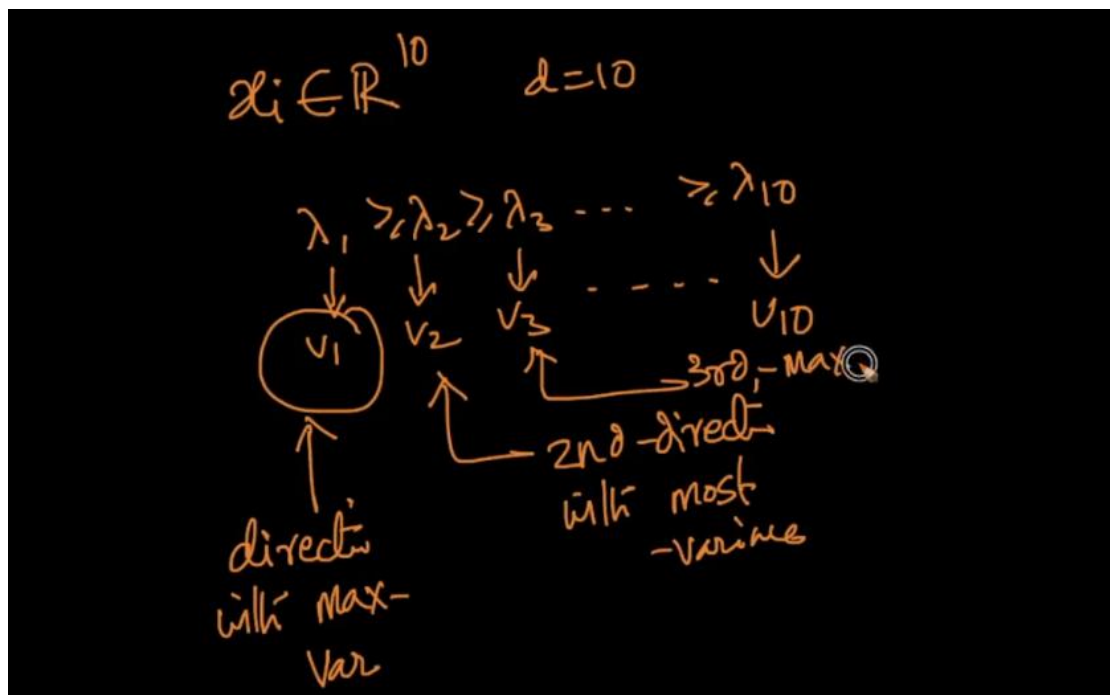
1. Calculate standardization of X matrix
2. Find S called Co-variance matrix of X
3. Find E values and Vectors of S
4. $U1 = V1$

Now we got direction for move f1 and f2 . lets see below image .



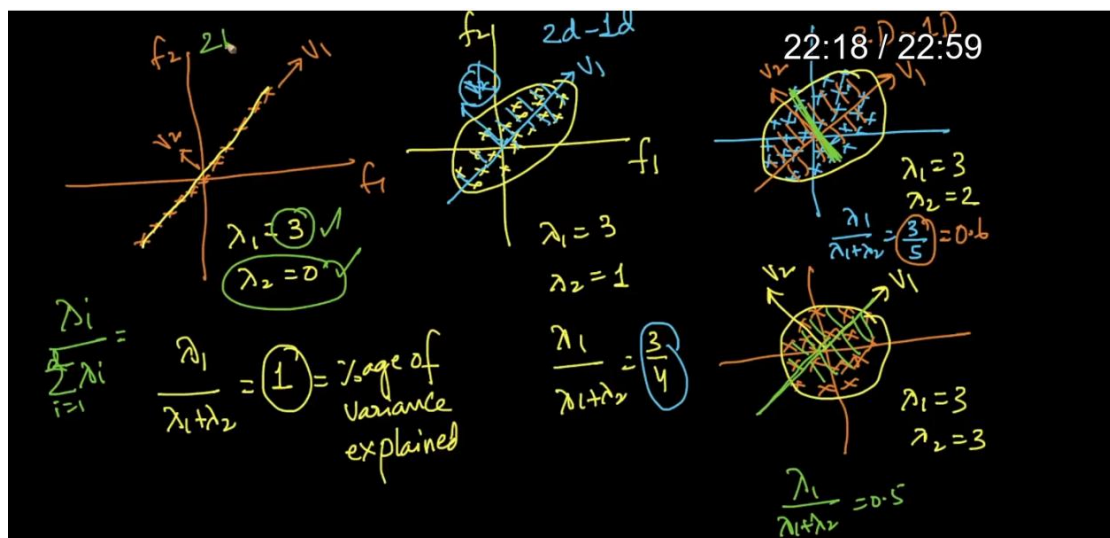
Now we move f_1, f_2 in new direction of v_1 so f_1 we will have lots of data as too much variance shown in image.

Now let's say we have 10 dimensional data then?



Now see above image v1 means direction with maximum variance ,v2 means variance with 2nd maximum variance . so on ... up to 10th direction .

Now we got importance of E vectors lets understand importance of E values .



See above 4 images in first $y_1 = 2$ and $y_2 = 0$
So as per formula we get $3/3+0 = 1$ means 100% data cover in 1st case .

Now 2nd we have $y_1=3, y_2=1$, $3/4 = 0.25$ that means we can cover from f_1 75% of data and 25% of data is spread on another axis .

So on ..

So E values gives us what % of data variance on features this is importance of E values and E vectors .

PCA is also called as maximum variance method .

Now I want to visualize 10 dimension data how can I do that ?

$$X = \begin{bmatrix} f_1 & f_2 & \dots & f_{10} \\ 1 \\ 2 \\ \vdots \\ n \end{bmatrix} \xrightarrow{\text{dim reduction (PCA)}} X' = \begin{bmatrix} v_1 & v_2 \\ 1 \\ 2 \\ \vdots \\ n \end{bmatrix}$$

$$S = X^T X$$

$$\text{eigen}(S) = \lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_{10}$$

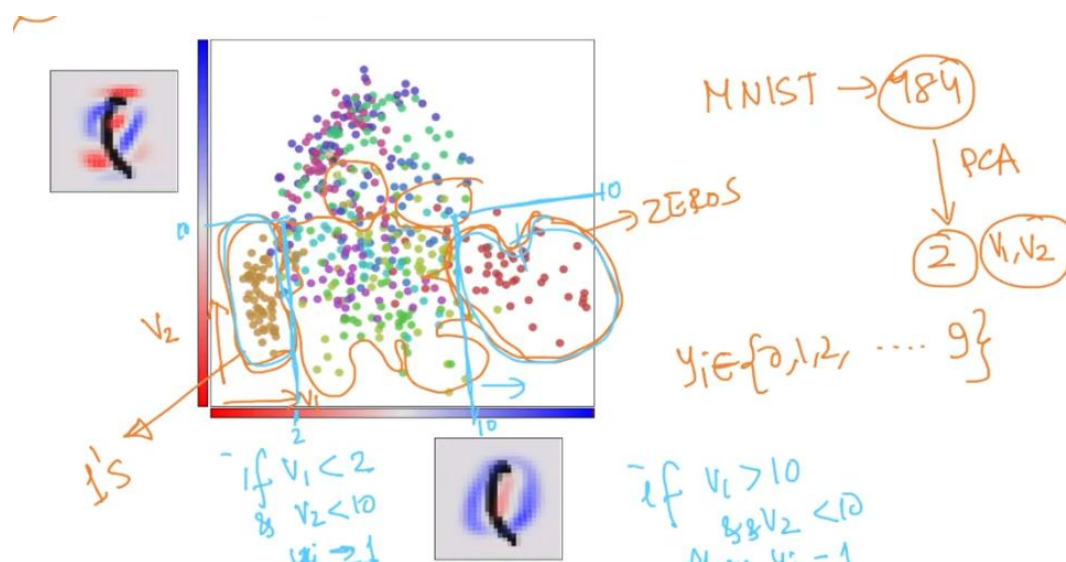
$$\downarrow \quad \downarrow \quad \downarrow \quad \dots \quad \downarrow$$

$$v_1 \quad v_2 \quad v_3 \quad \dots \quad v_{10}$$

$$x_i' = \begin{bmatrix} x_i^T v_1 & x_i^T v_2 \end{bmatrix}$$

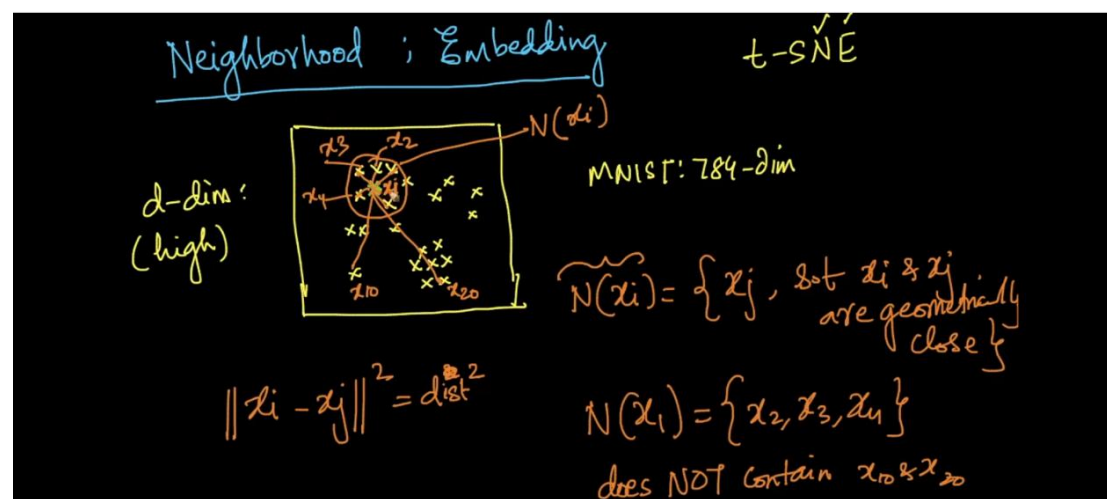
See X is 10 dimension matrix from that we find S that Co-variance matrix then we find E values and Vectors . Then from X we create new data using top two E vectors v1 and v2 , X' .

So we will do multiplication of Xi with V1 and V2 and create new data X' which has 2 dimensions only .

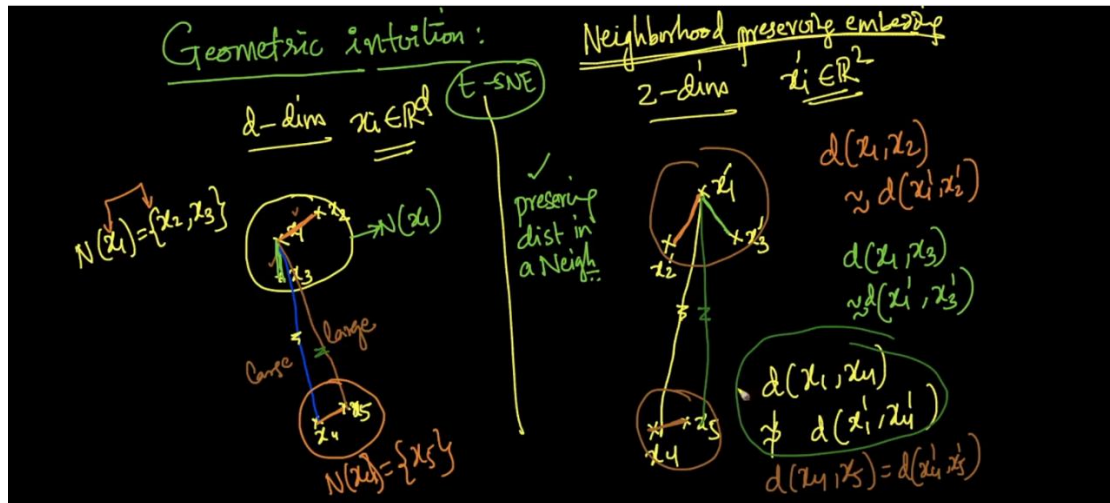


See above image this is the disadvantage of PCA .
 PCA not able to classify or not good to give good features this is very old techniques .

T SNE :



Neigh means see above circle we will take distance between to points if it is small then it belongs to that class .



See above image left side we have n dimension data now we want to make it 2d using T SNE .

This method cares of about neighbourhood relation . see above $n(x_1) = \{x_2, x_3\}$ but not x_4 and x_5 because x_4, x_5 are too far away from x_1 .

So T SNE will create another data lets say $x_1', x_2' \dots x_{n'}$.

And it will take care of same distance in x_1 neigh but it will not care about x_4, x_5 distance ..

In T SNE T is T dist which is used to solve crowding problem .

See below image how beautifully T SNE separate 784 dimension to 2d .



t-SNE:-
group points based on
their visual
similarity

A t-SNE plot of MNIST 8y → 2d