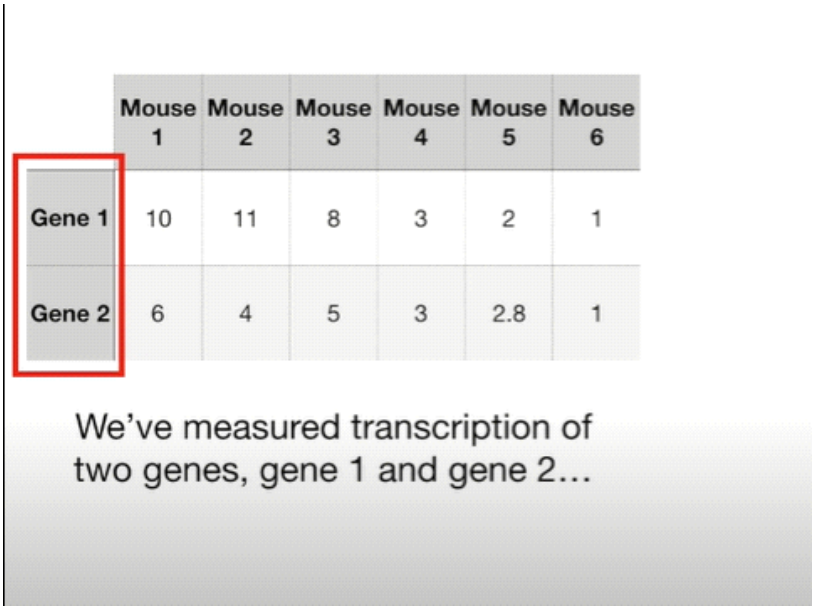
Principle component analysis

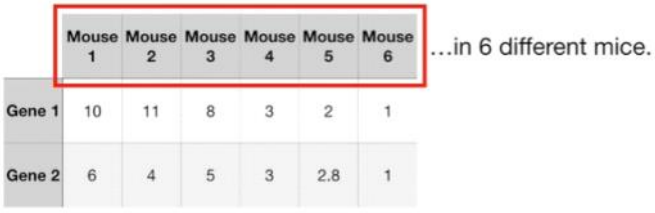
Friday, July 15, 2022 10:56 PM

STEP FOR PCA :

1_DEFINE THE DATA
2_PLOT THE DATA
3_MAKING DATA MEAN CENTERED
3_DIVIDE DATA BY STD. DEV . TO AVOID differeNT UNITS (m/cm) IN DATASET / TO MAKE VARIANCE = 1 (SCALING)
3_compute the data covariance matrix of data
4_FINDING BEST FITTED LINE (PC1) , pc1 is linear combination of first and 2nd category.and these combination tells the loading score and also which category is more imp. That is has more variation than other one,has more spread.
5_SPREAD OF DATA δ LINEAR COMBINATION
6_CALCULATE LENGTH OF PC1
7_EIGEN VECTOR AND EIGEN VALUE OF PC1, the linear combination (best fitted line,pc1) when we make it singular vector, then it is called eigen vector, the max. ssd o f pc1 is called e.value
8_FINDING ORTHOGONAL LINE (TO BEST FITTED LINE) (PC2)
9_PCA PLOT
10_VARIATION

We tested 2 genes of 6 mice

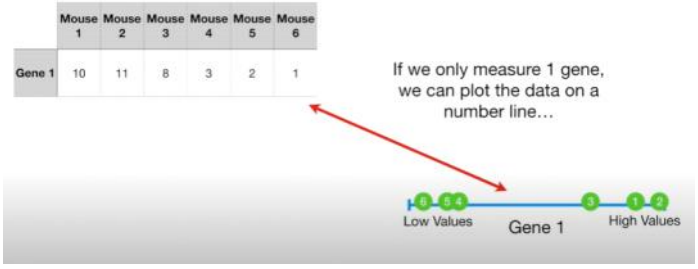


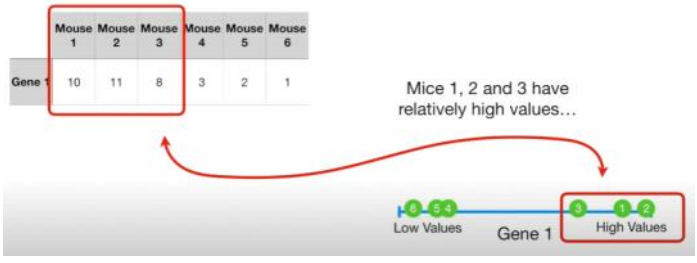


	Sample 1	Sample 2	Sample 3	Sample 4	...
Variable 1	10	11	8	3	...
Variable 2	6	4	5	3	...

...and the genes as variables that we measure for each sample.

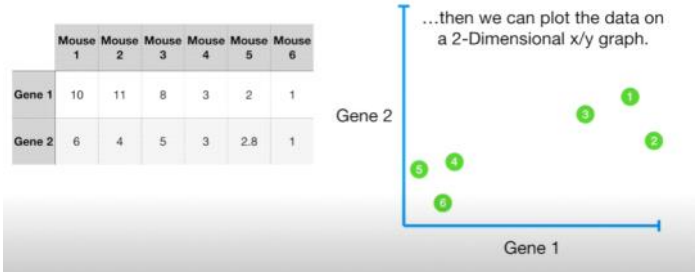
3 m genes ak jesi han or teen m ak jesi



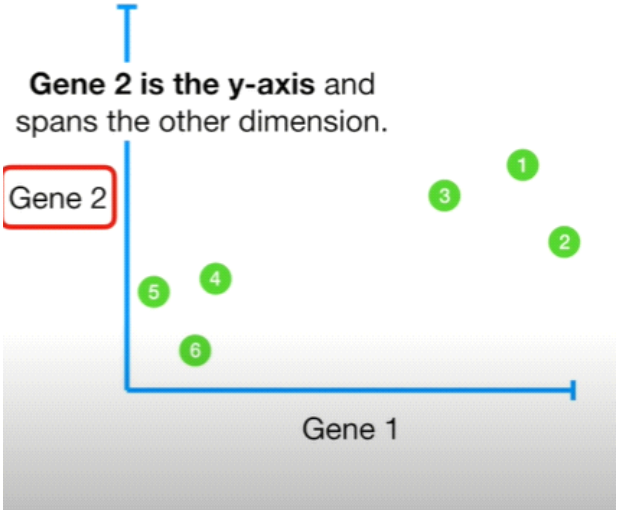


Even though it's a simple graph, it shows us that mice 1, 2 and 3 are more similar to each other than they are to mice 4, 5 6.

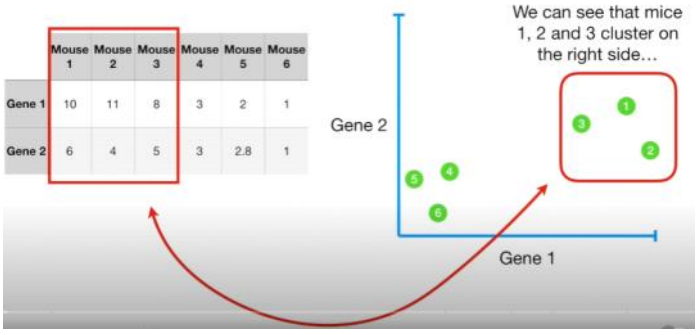
Gene 1 ko x-axis par se kia or gene 2 ko y-axis par

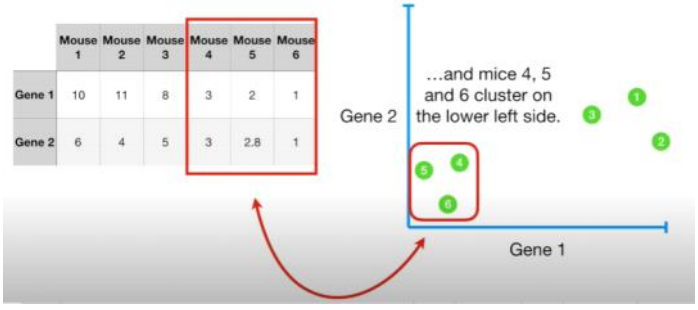


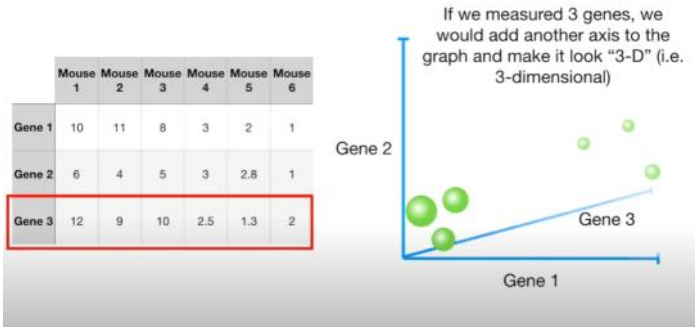
Gene 1 is the x-axis and spans one of the 2 dimensions in this graph.

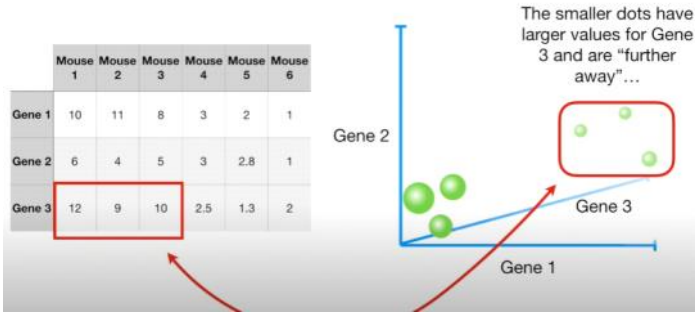


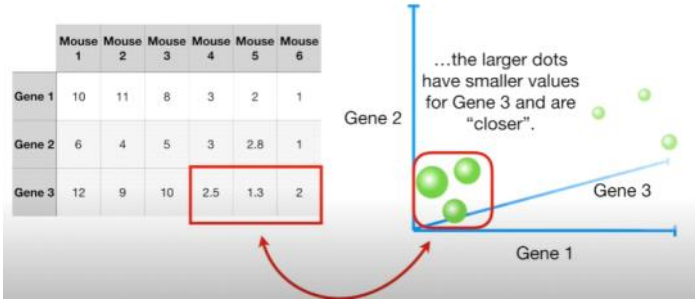
// DISCUSSING 3D and 4D in BTW. //

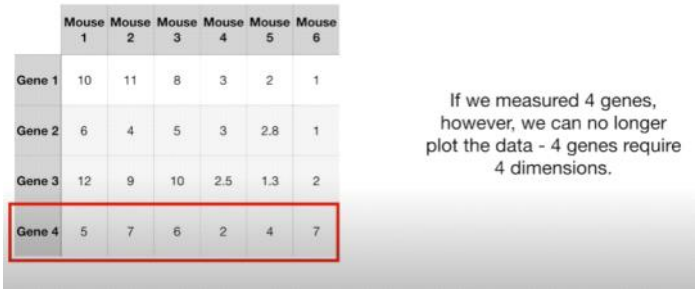




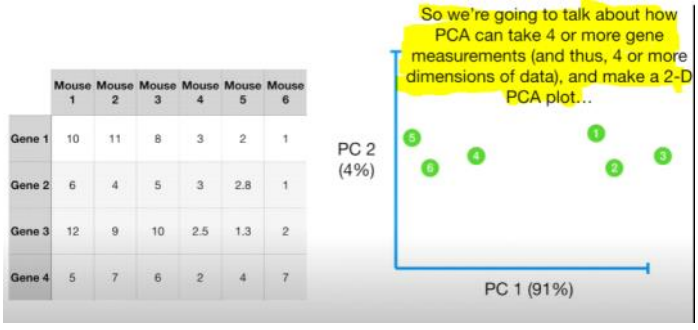


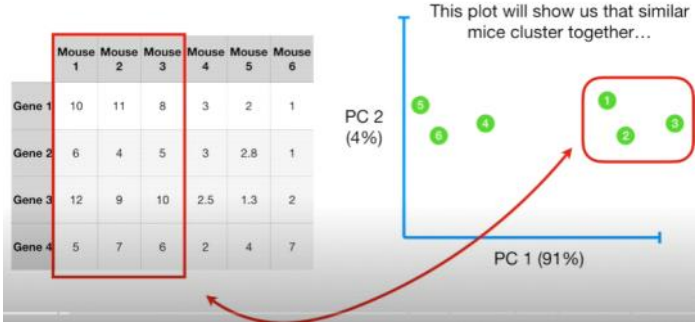


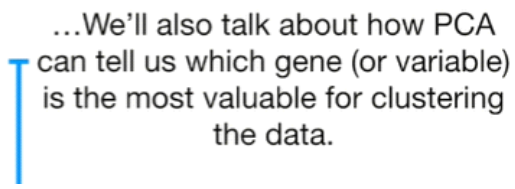


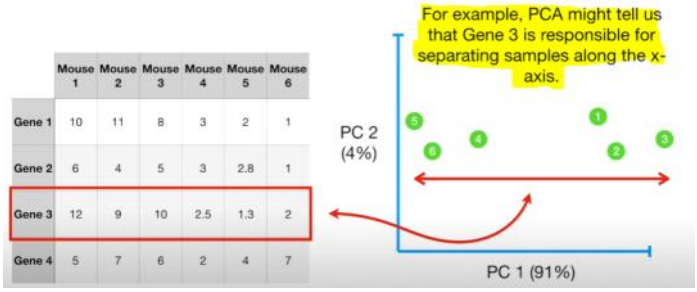


PCA can take 4 dimensions and represent them in 2D









	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2
Gene 4	5	7	6	2	4	7

// Back to 2D mice, gene data //

1_DEFINE THE DATA :

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

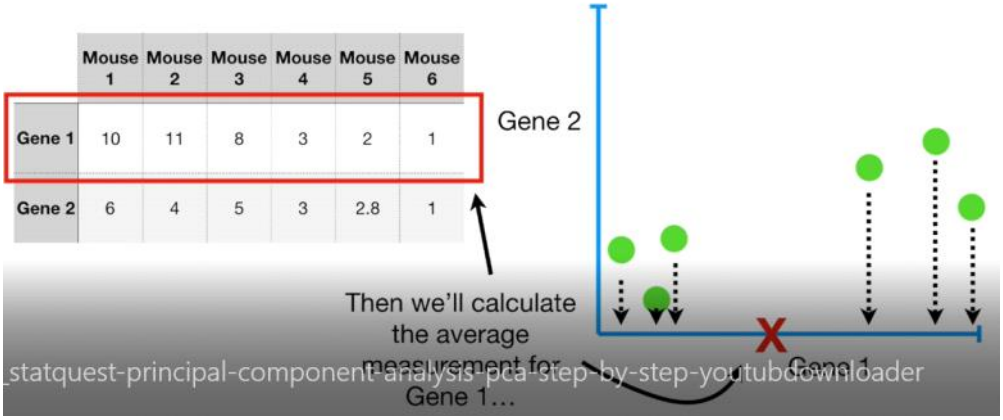
To understand what PCA does and how it works, let's go back to the dataset that only had 2 genes…

2_PLOT THE DATA :

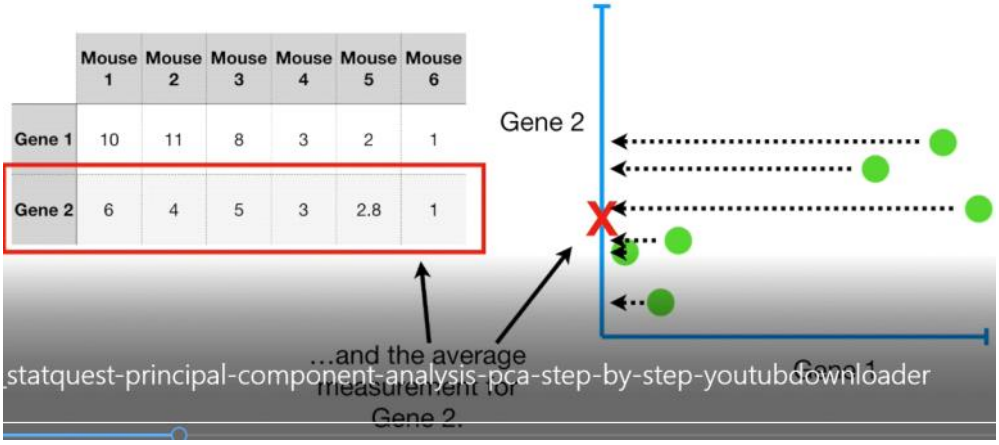
Gene 1 on x-axis, Gene 2 on y-axis , So we see that similar samples cluster together.

3_MAKING DATA MEAN CENTERED :

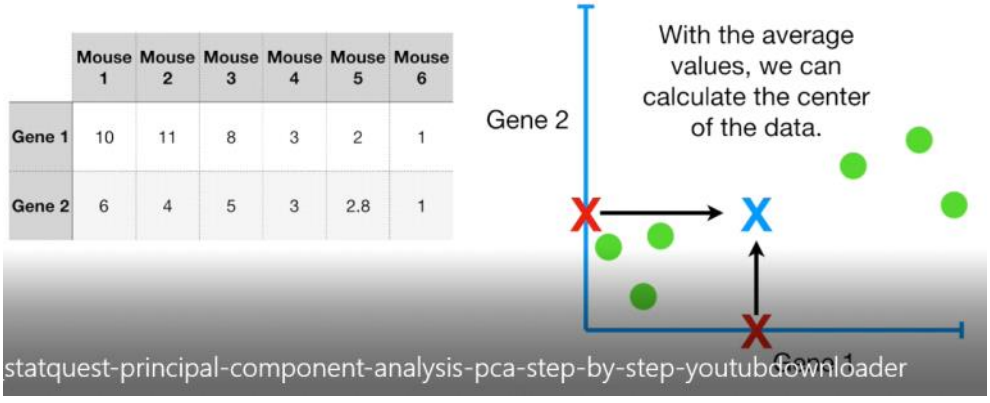
Taking average of gene 1

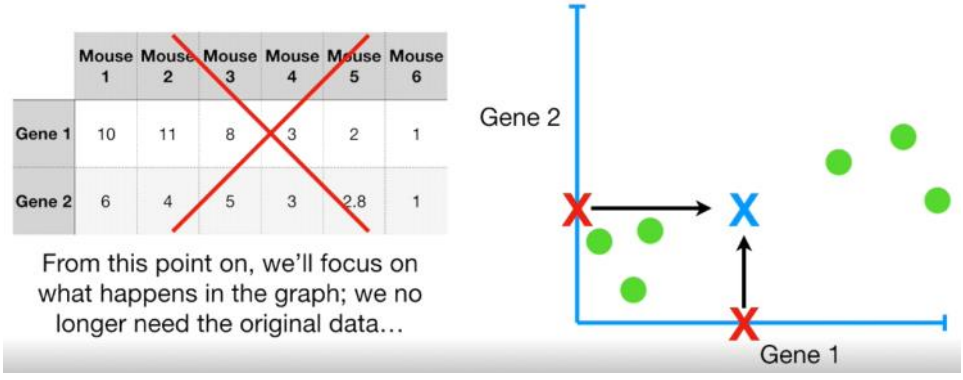


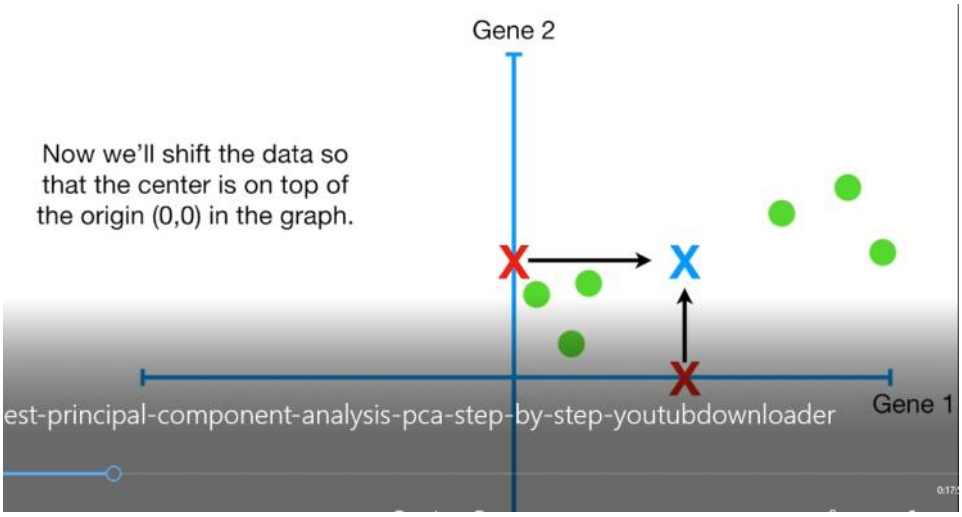
Taking average of gene 2

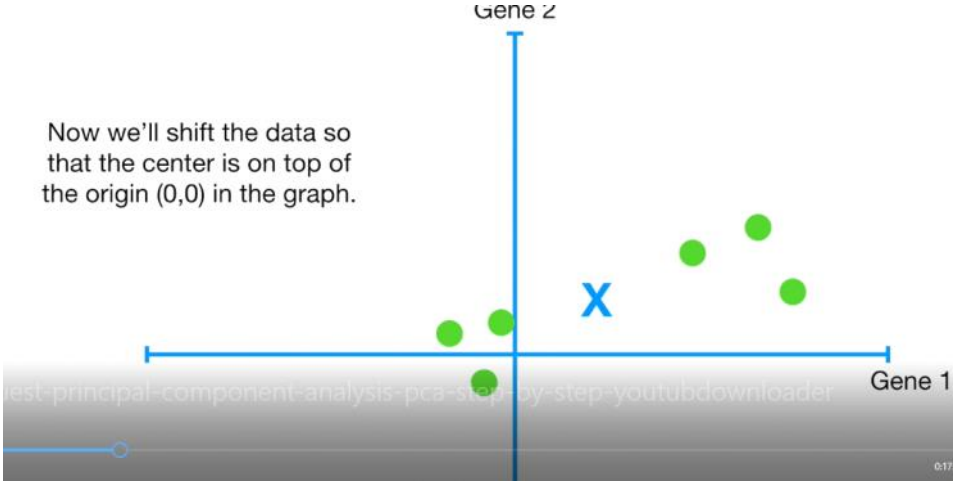


Taking average of both averages.

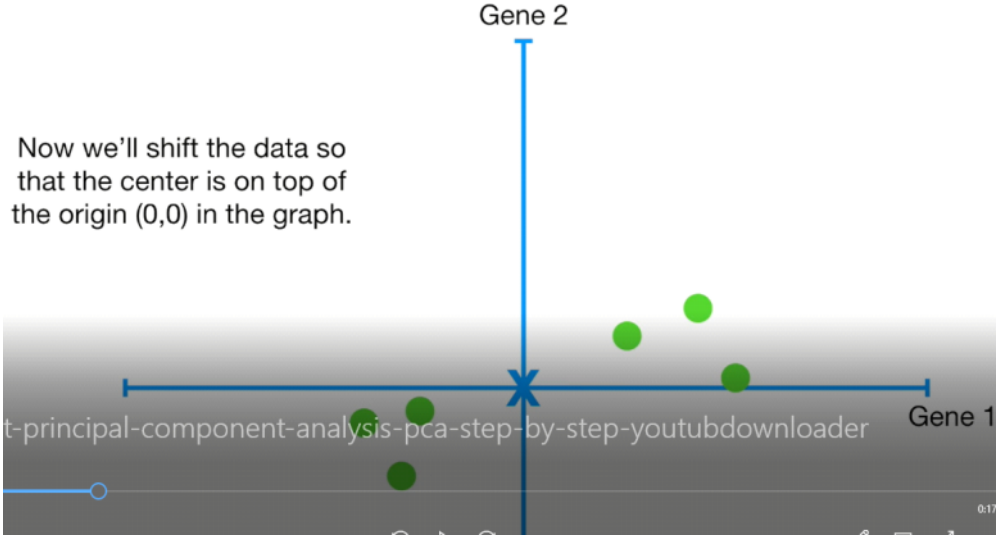




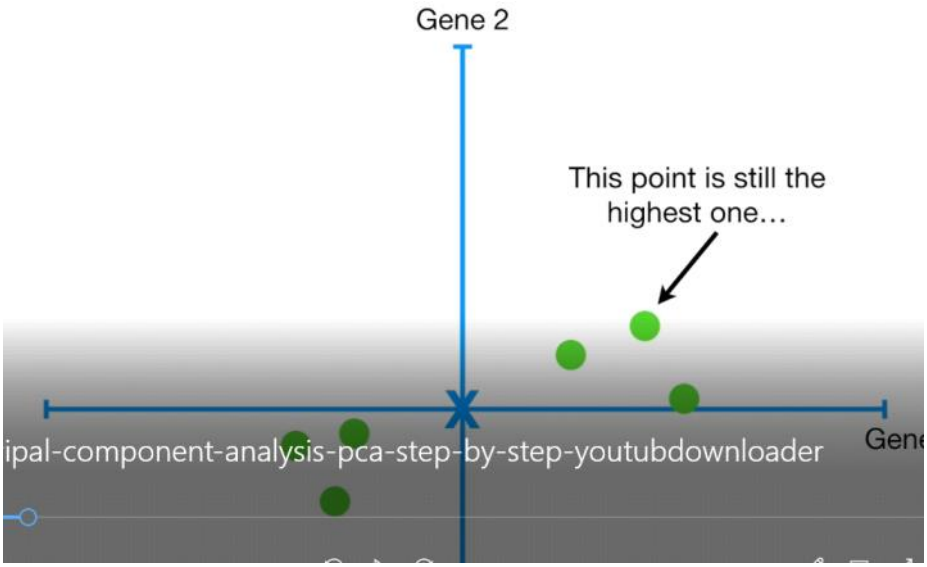




Mean centered to origin (mean centered data) :



NOTE: Shifting the data did not change how the data points are positioned *relative to each other*.

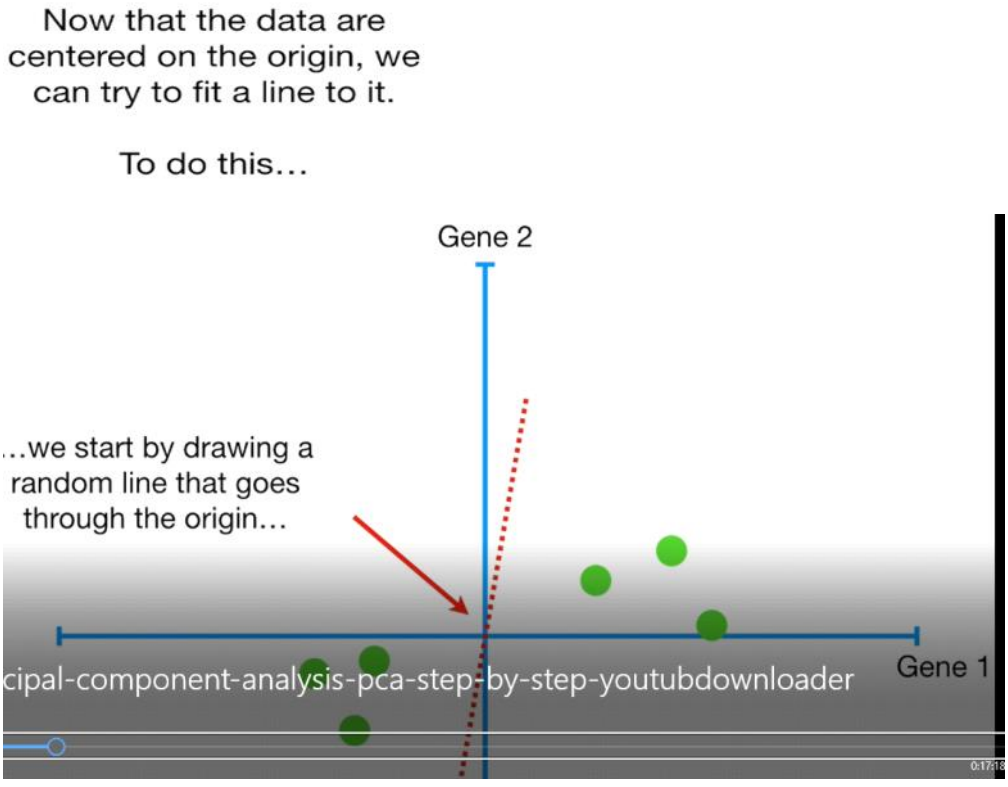


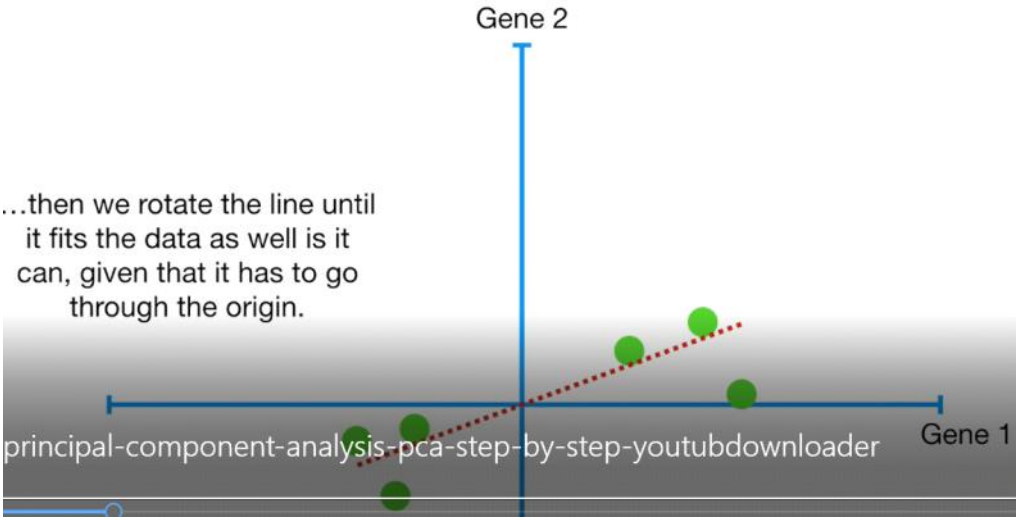
3_Scaling data:

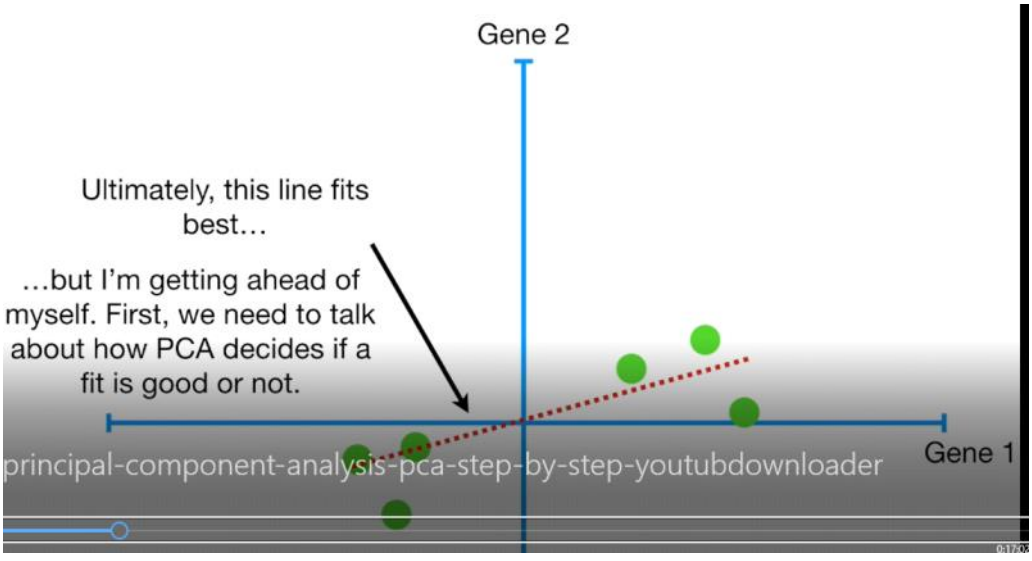
We saling data to avoid different units , by dividing each sample by its std dev

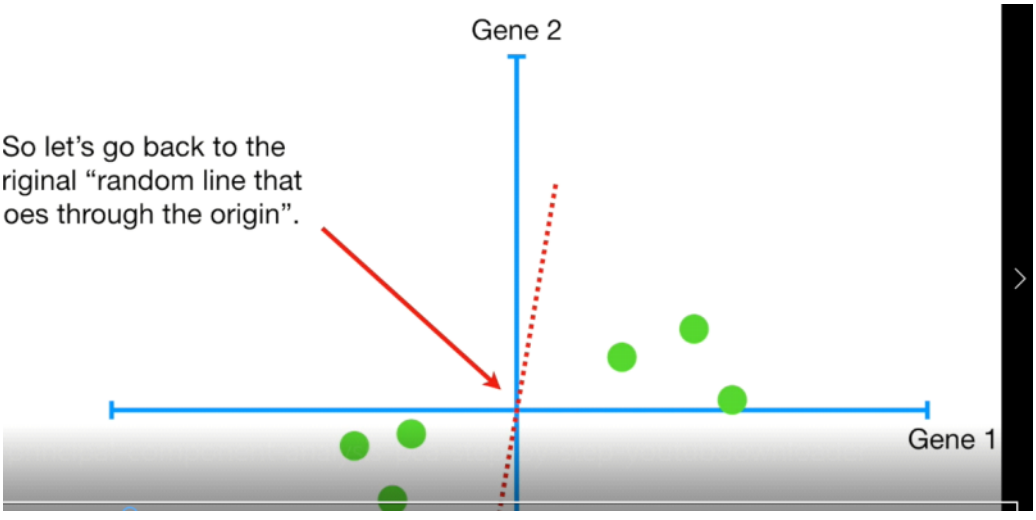
4_FINDING BEST FITTED LINE (PC1):

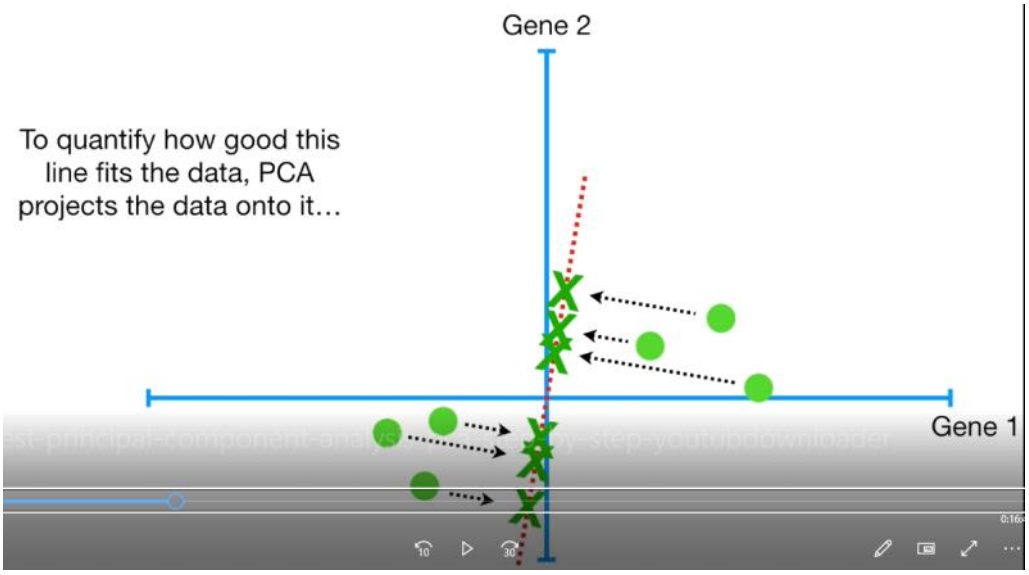
Using PCA()



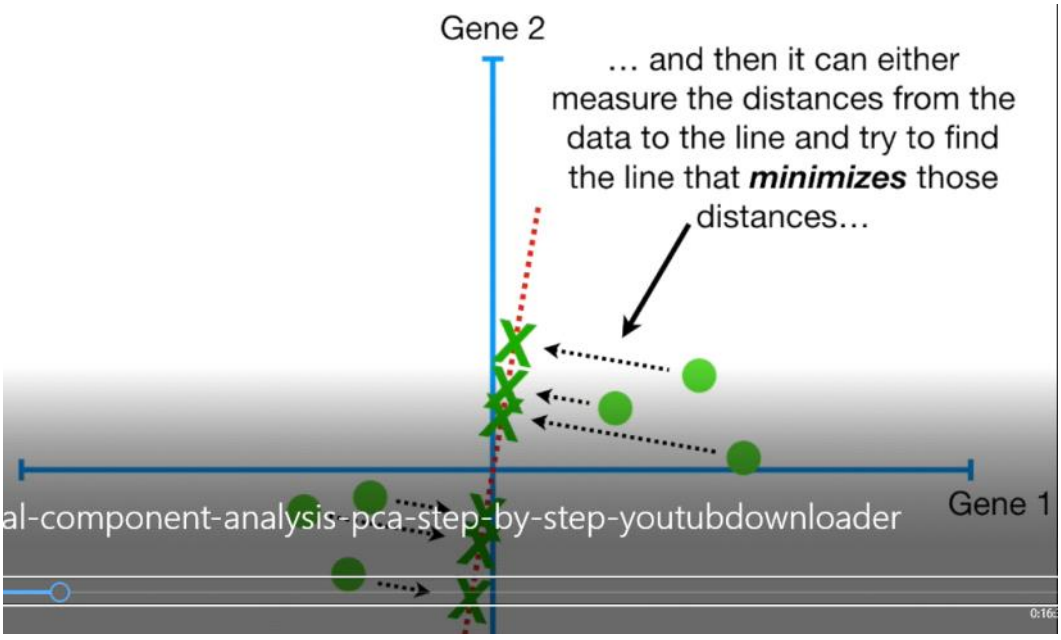

Gene 2
…then we rotate the line until it fits the data as well is it can, given that it has to go through the origin.
principal-component-analysis-pca-step-by-step-youtubdownloader
Gene 1


Gene 2
Ultimately, this line fits best…
…but I'm getting ahead of myself. First, we need to talk about how PCA decides if a fit is good or not.
principal-component-analysis-pca-step-by-step-youtubdownloader
Gene 1

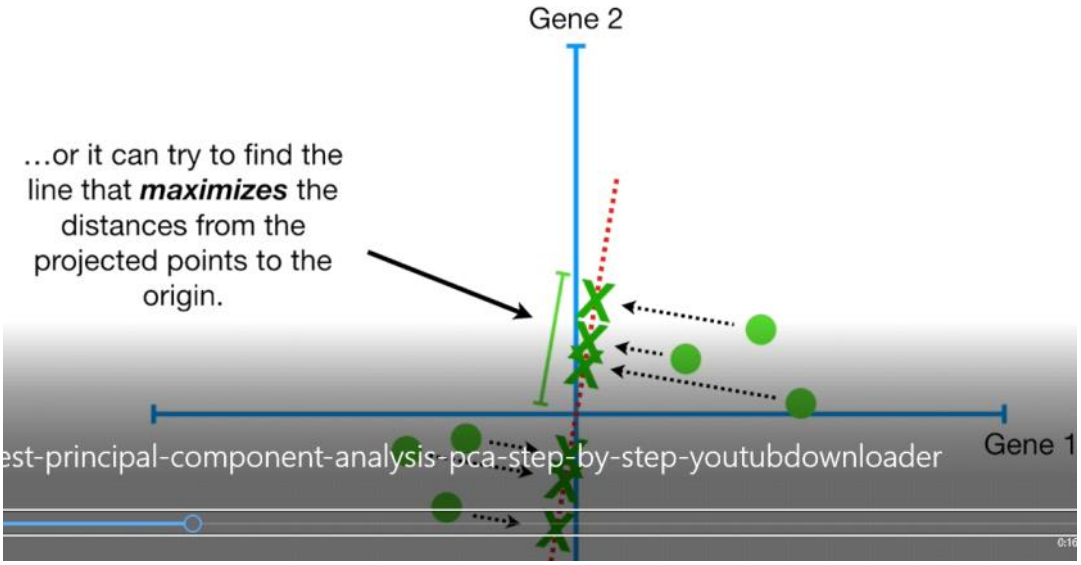

Gene 2
So let's go back to the riginal "random line that oes through the origin".
Gene 1



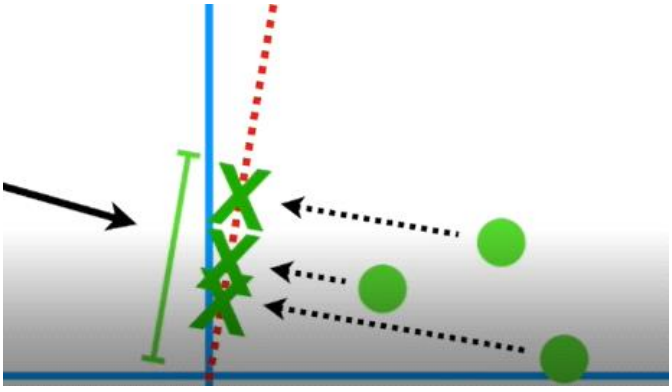
OPTION-1 :



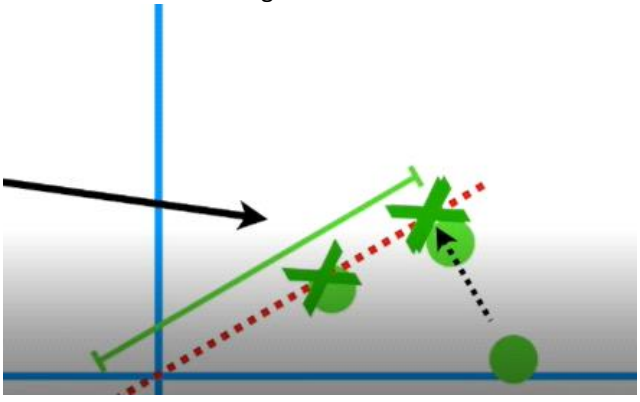
OPTION-2 :



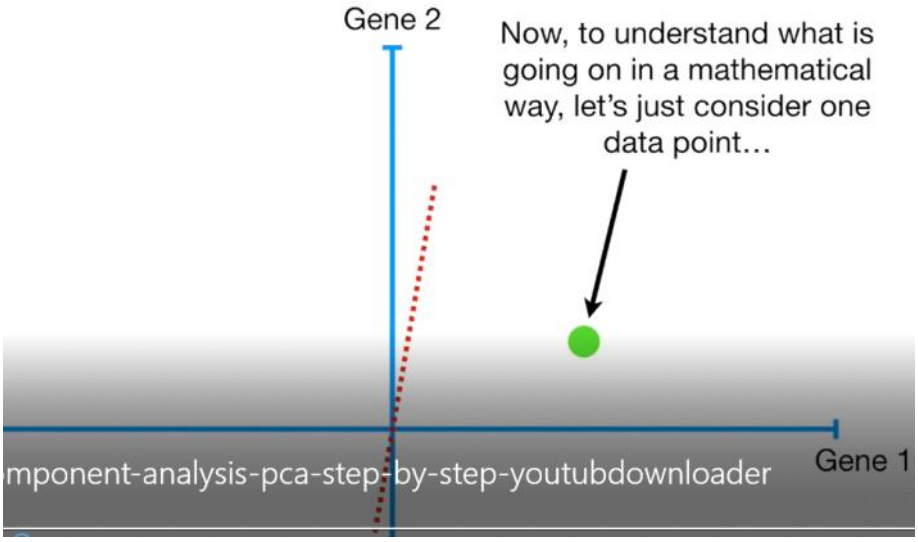
We can measure the distances from data points to the fitting line **OR** we can measure the distances from projected points on the fitting line to the origin.
The distances from the point of projection to the origin increase when the line best fits the data.

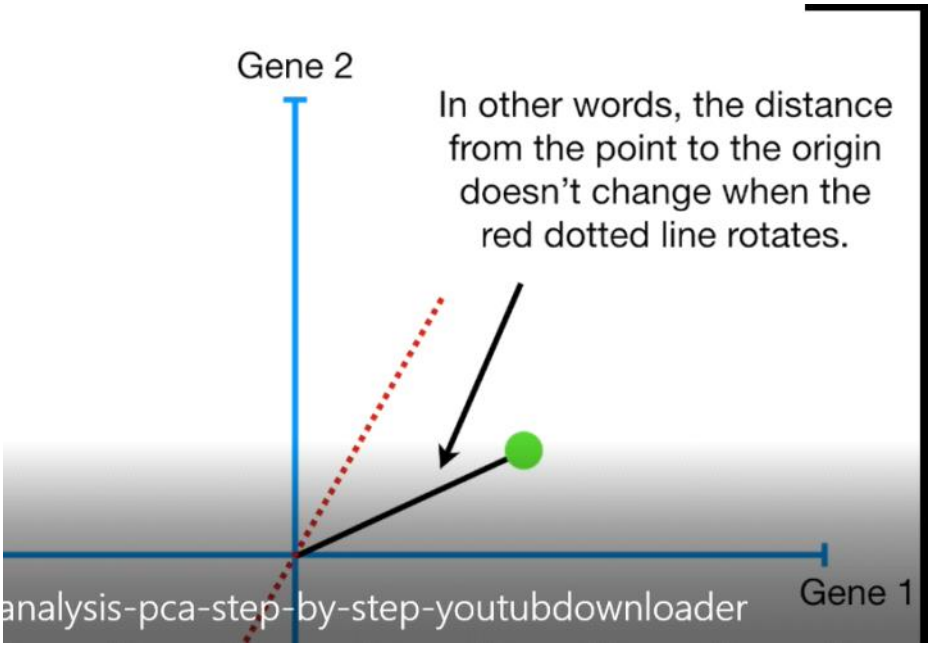


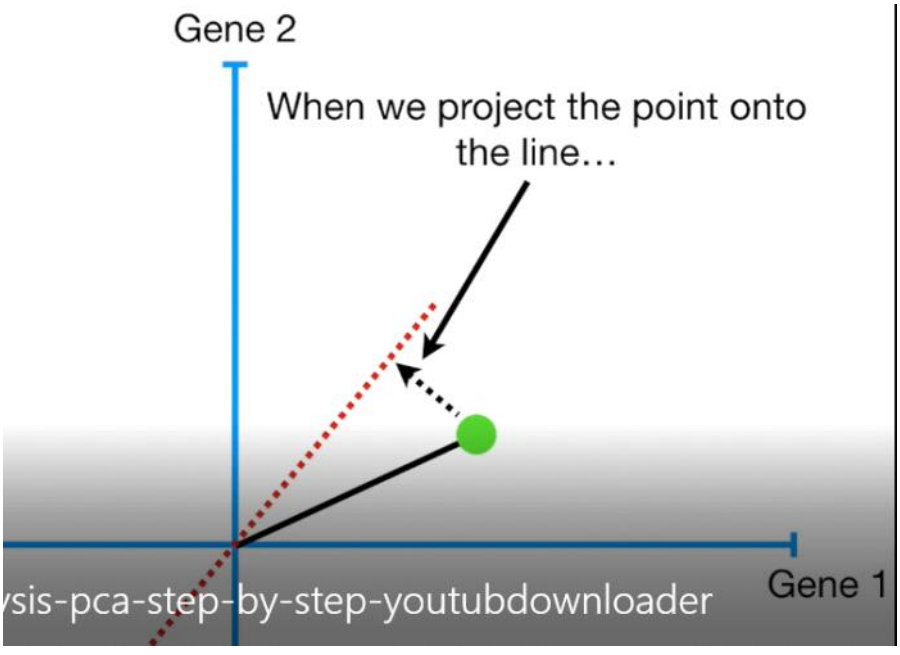
When distance from origin increases the distance from data point decrease.

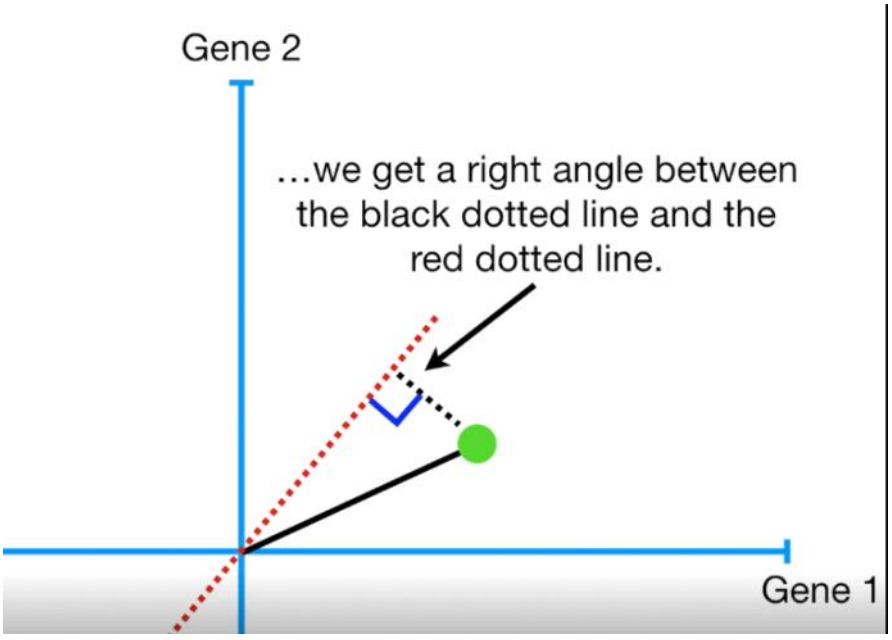


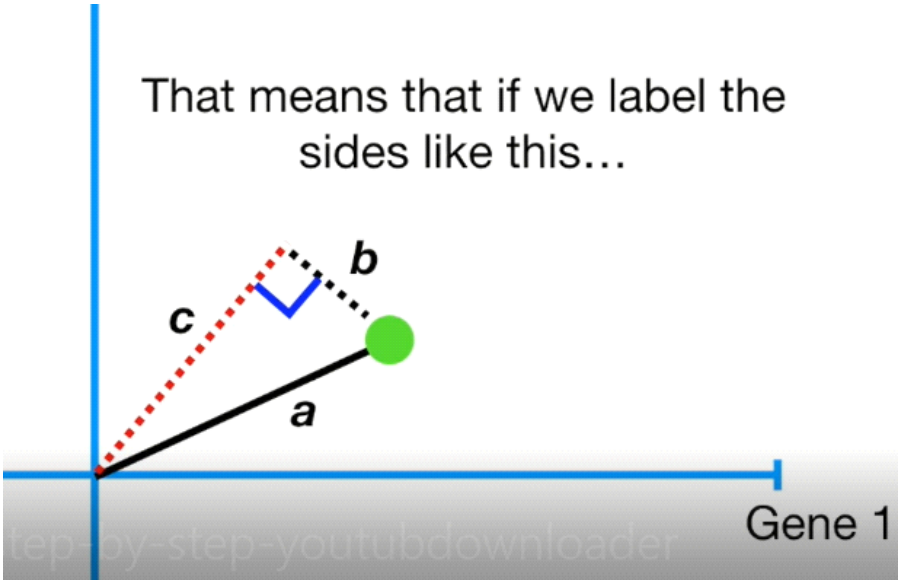
Logically it is more appropriate the minimize the distance btw the Datapoints to fitting line but Instead the PCA chooses second option and maximizes the distance btw. Origin and projected data points, because it is easier to do.


Gene 2
Now, to understand what is going on in a mathematical way, let's just consider one data point…
mponent-analysis-pca-step-by-step-youtubdownloader
Gene 1


Gene 2
In other words, the distance from the point to the origin doesn't change when the red dotted line rotates.
analysis-pca-step-by-step-youtubdownloader
Gene 1

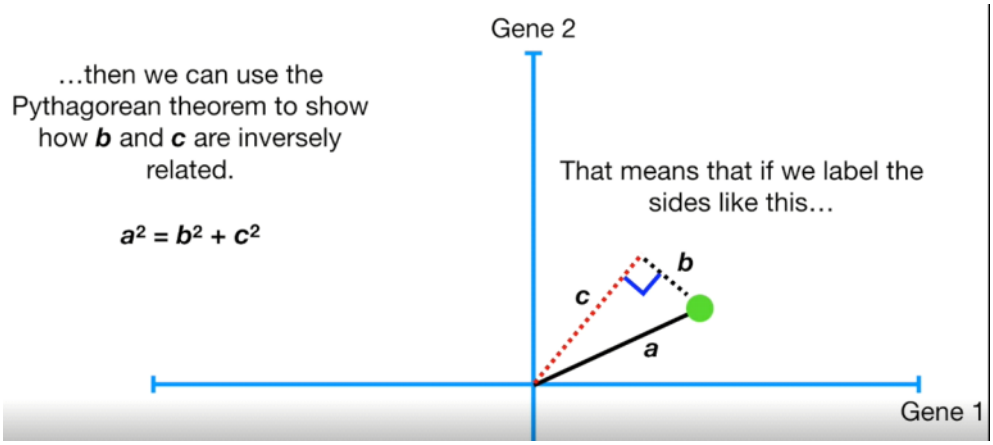

Gene 2
When we project the point onto the line…
ysis-pca-step-by-step-youtubdownloader
Gene 1

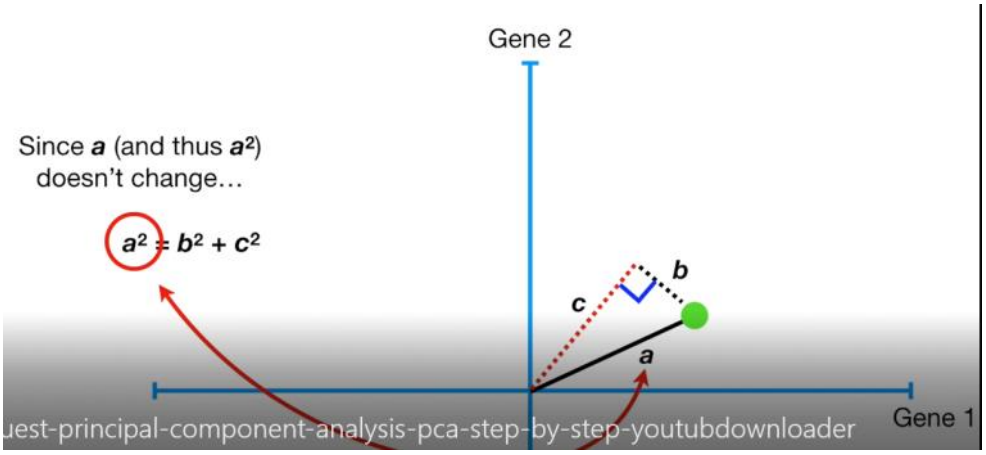


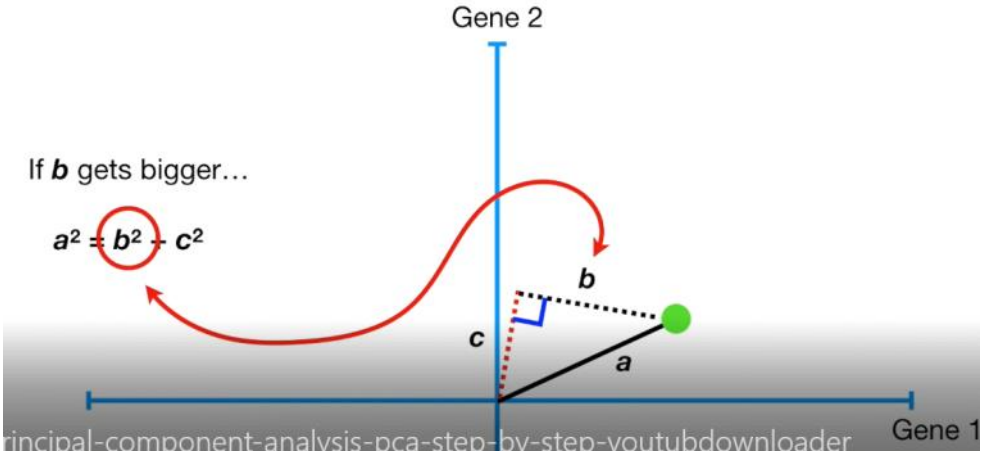


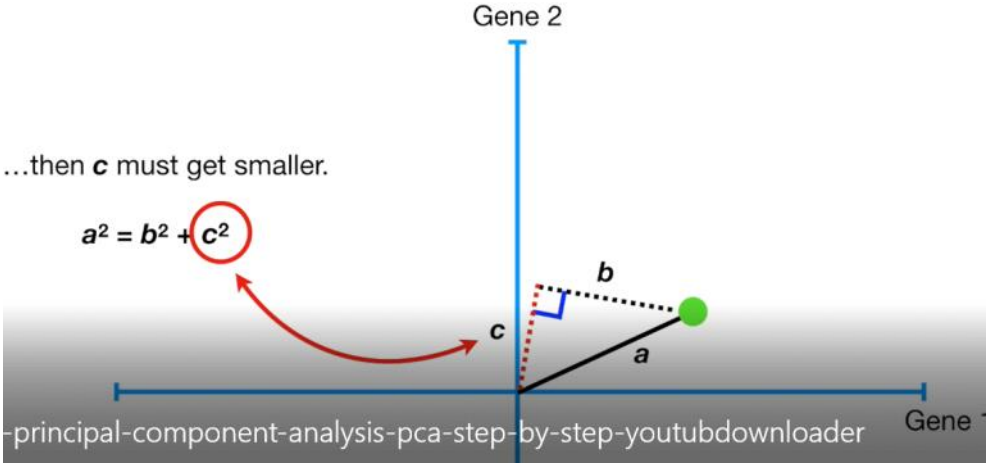
…then we can use the Pythagorean theorem to show how ***b*** and ***c*** are inversely related.

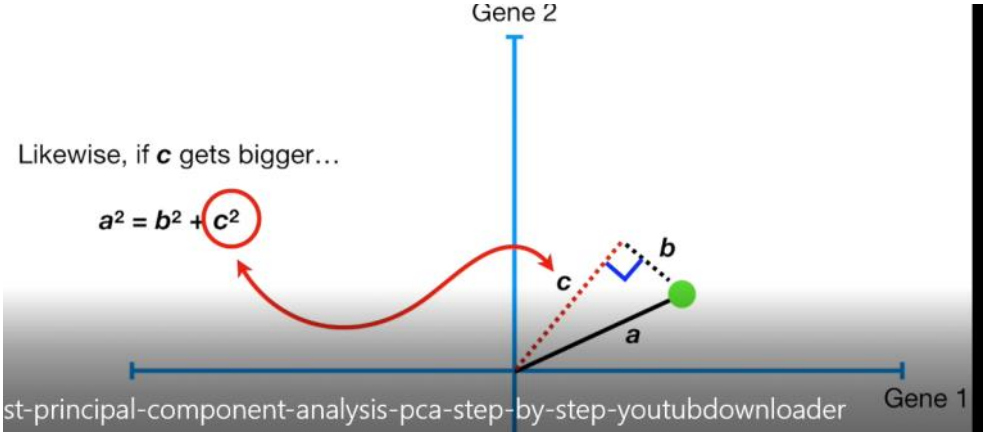
$$a^2 = b^2 + c^2$$

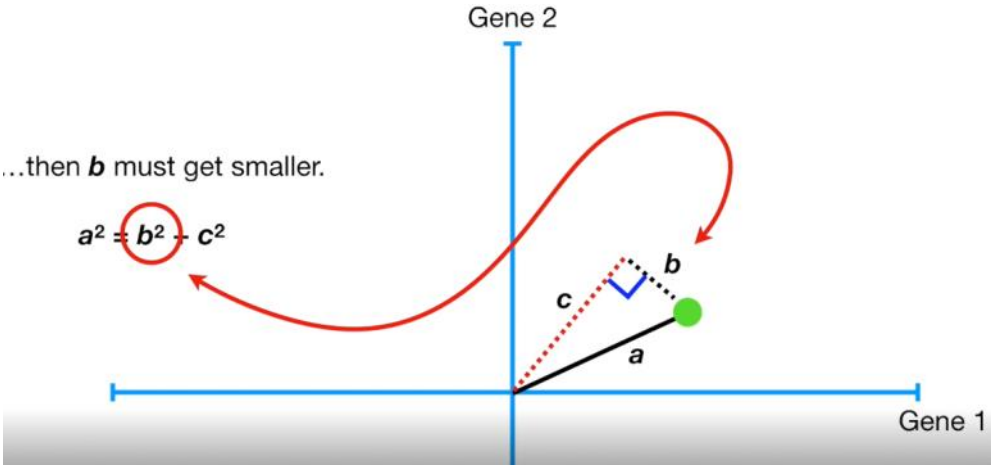

Gene 2
…then we can use the Pythagorean theorem to show how b and c are inversely related.
a² = b² + c²
That means that if we label the sides like this…
b
c
a
Gene 1

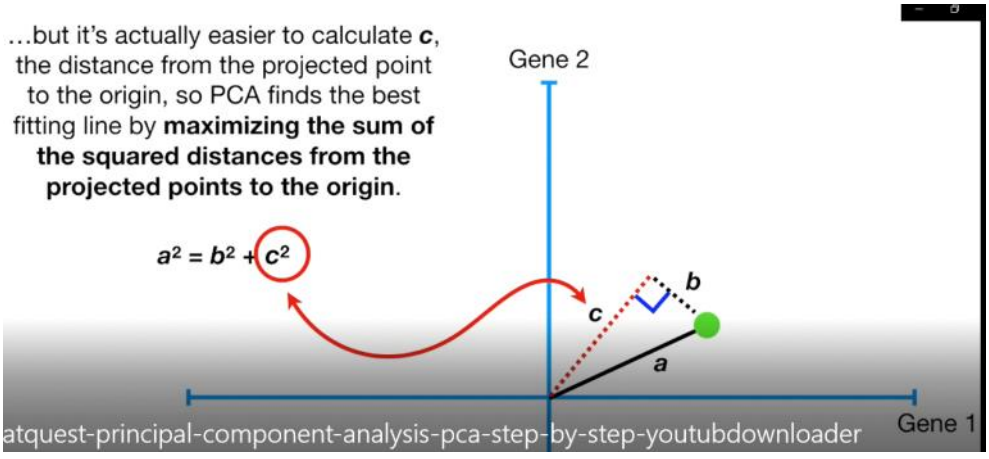

Gene 2
Since a (and thus a²) doesn't change…
a² = b² + c²
b
c
a
Gene 1
uest-principal-component-analysis-pca-step-by-step-youtubdownloader

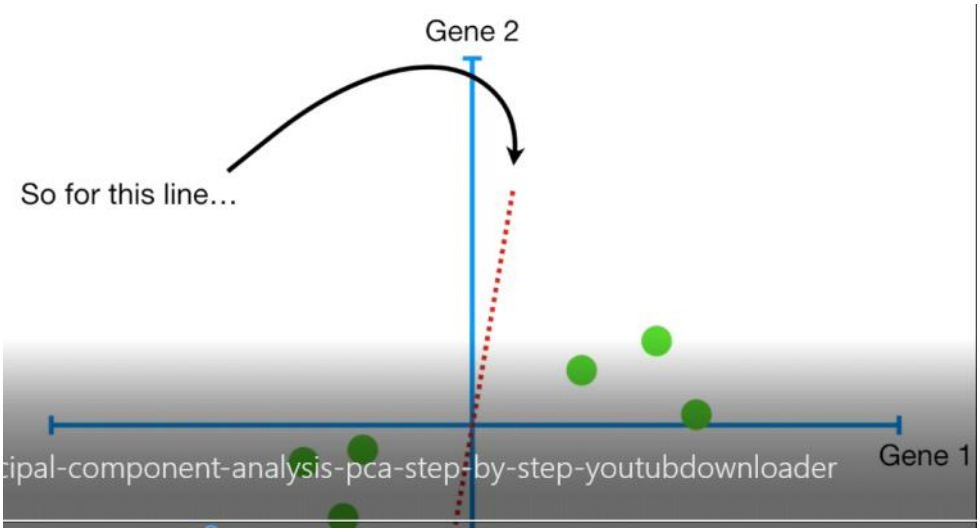

Gene 2
If b gets bigger…
a² = b² + c²
b
c
a
Gene 1
rincipal-component-analysis-pca-step-by-step-youtubdownloader

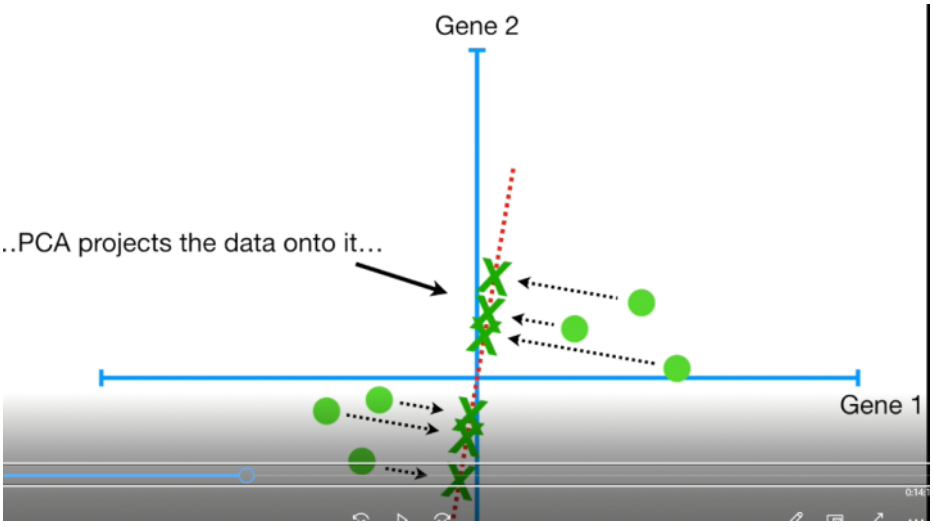

Gene 2
…then c must get smaller.
a² = b² + c²
b
c
a
-principal-component-analysis-pca-step-by-step-youtubdownloader

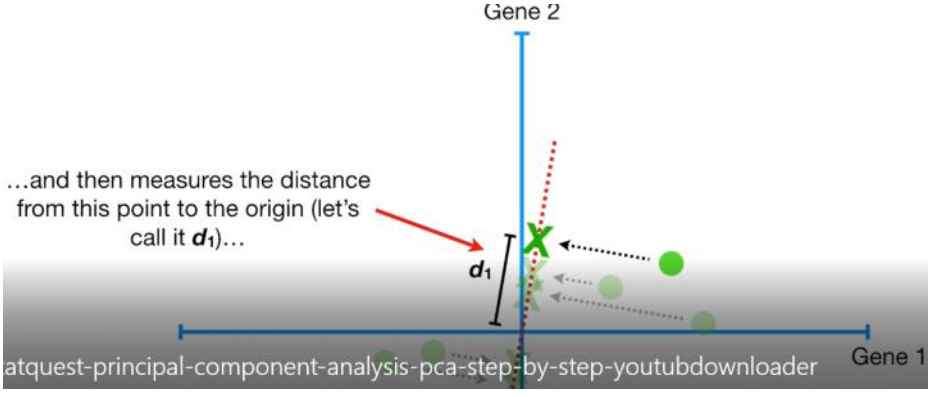

Gene 2
Likewise, if c gets bigger…
a² = b² + c²
b
c
a
st-principal-component-analysis-pca-step-by-step-youtubdownloader
Gene 1

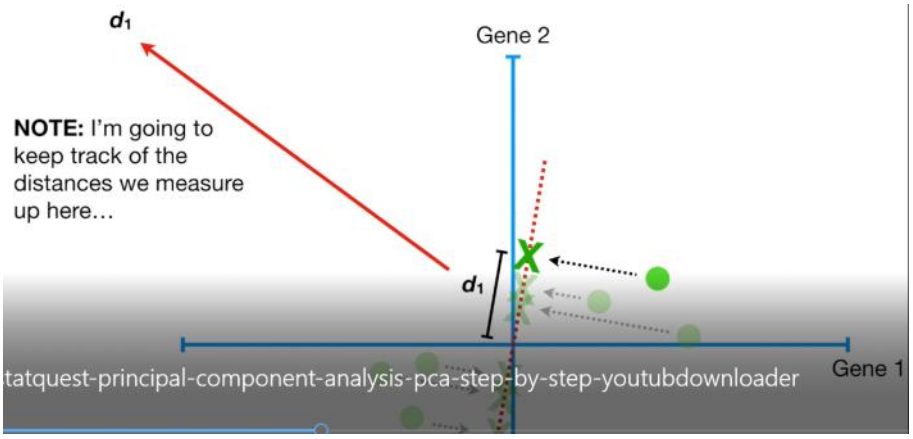

Gene 2
…then b must get smaller.
a² = b² + c²
b
c
a
Gene 1

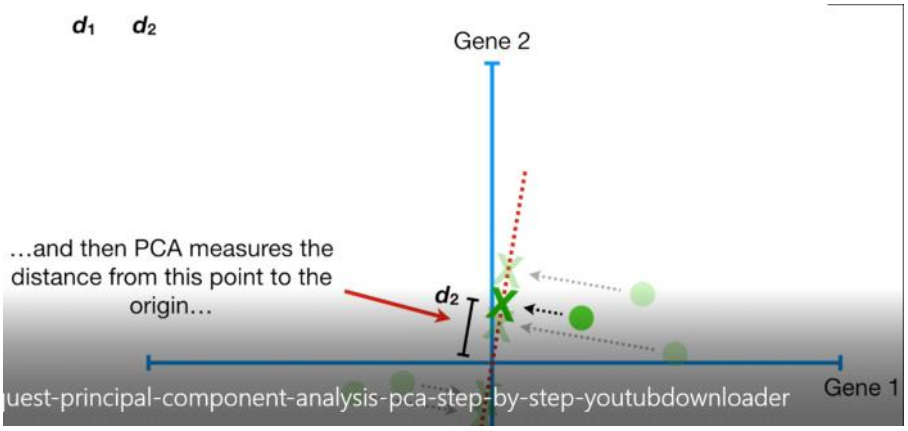

…but it's actually easier to calculate c, the distance from the projected point to the origin, so PCA finds the best fitting line by maximizing the sum of the squared distances from the projected points to the origin.
Gene 2
a² = b² + c²
b
c
a
atquest-principal-component-analysis-pca-step-by-step-youtubdownloader
Gene 1

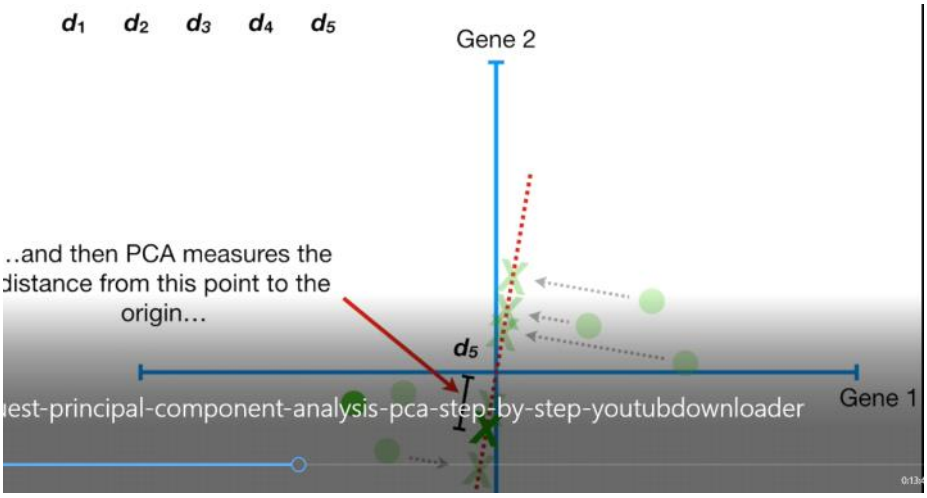

Gene 2
So for this line…
Gene 1
cipal-component-analysis-pca-step-by-step-youtubdownloader


Gene 2
..PCA projects the data onto it…
Gene 1


Gene 2
…and then measures the distance from this point to the origin (let's call it d_1)…
d_1
Gene 1
atquest-principal-component-analysis-pca-step-by-step-youtubdownloader

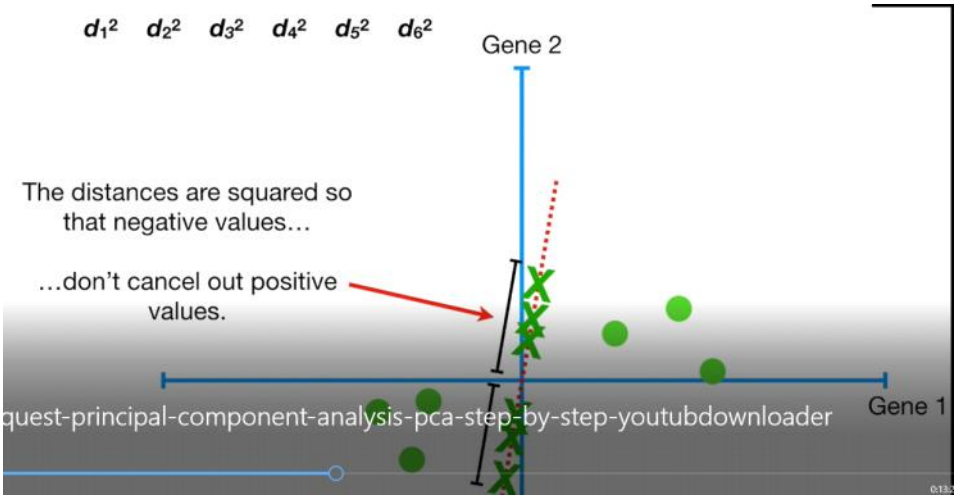






d_1 d_2 d_3 d_4 d_5 d_6

Here are all 6 distances that we measured.



$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)

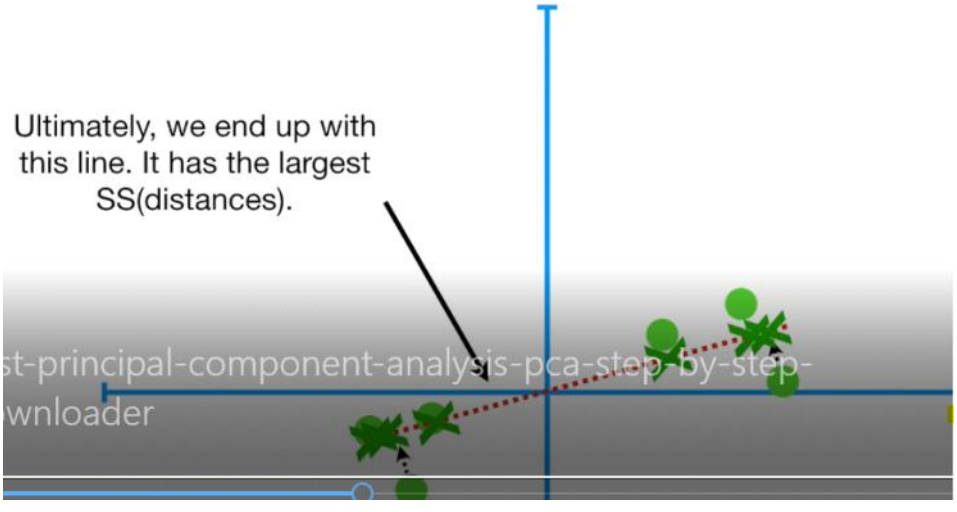
Now we keep rotating the line and doing same process untill we get the biggest SSD, As that will be the best fitting for data points :

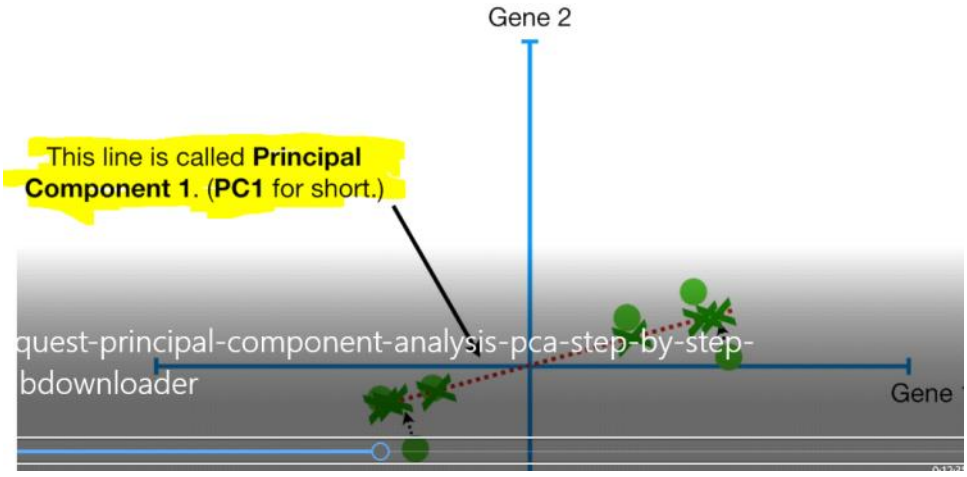
Now we rotate the line…

…project the data onto the line…

…and then sum up the squared distances from the projected points to the origin…

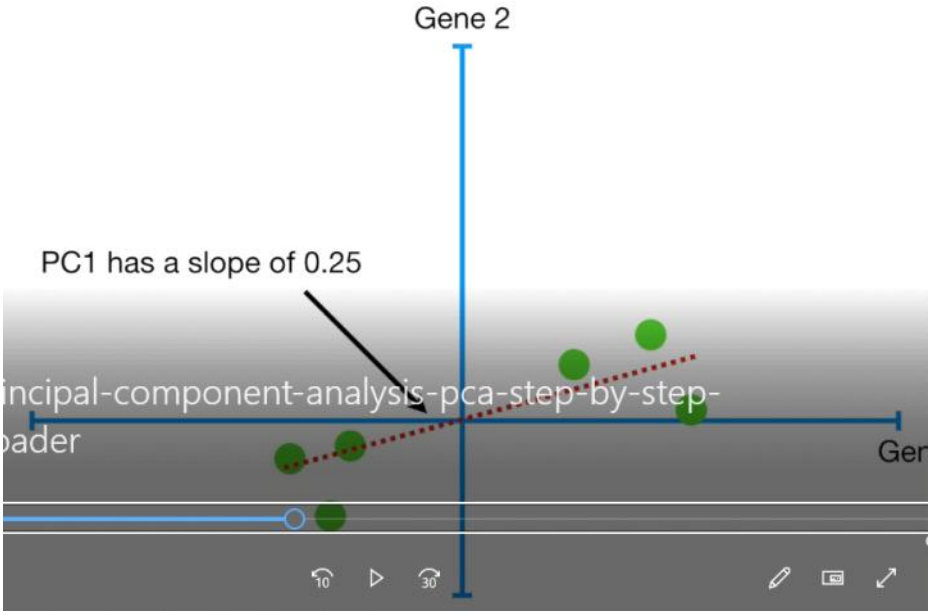
…and we repeat until we end up with the line with the largest sum of squared distances between the projected points and the origin.

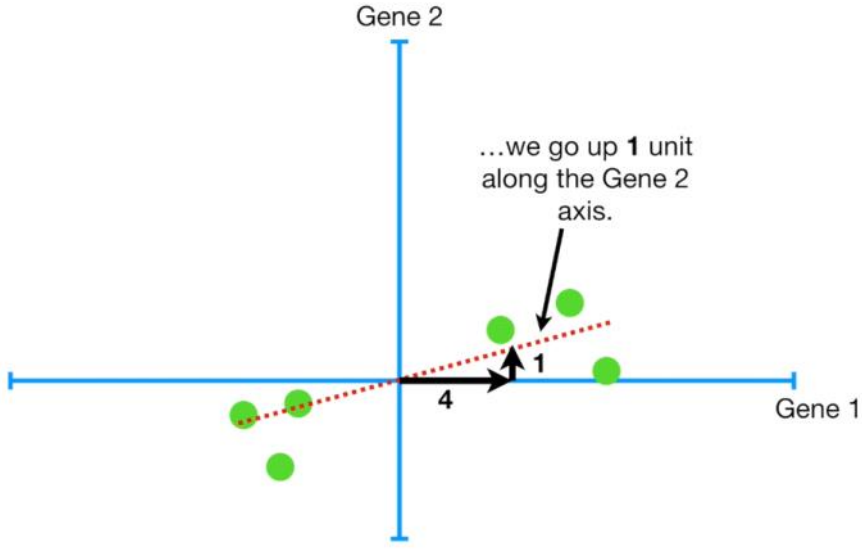


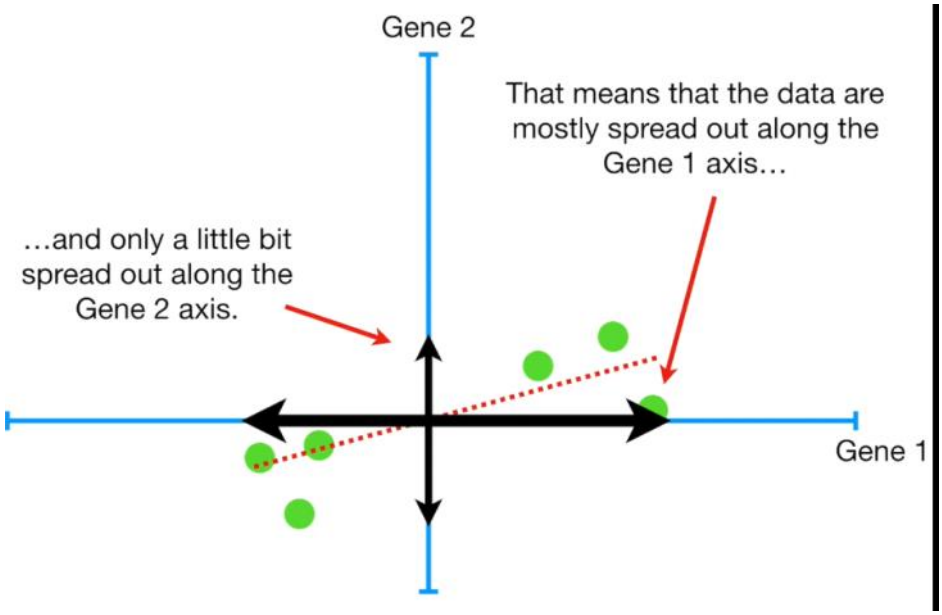


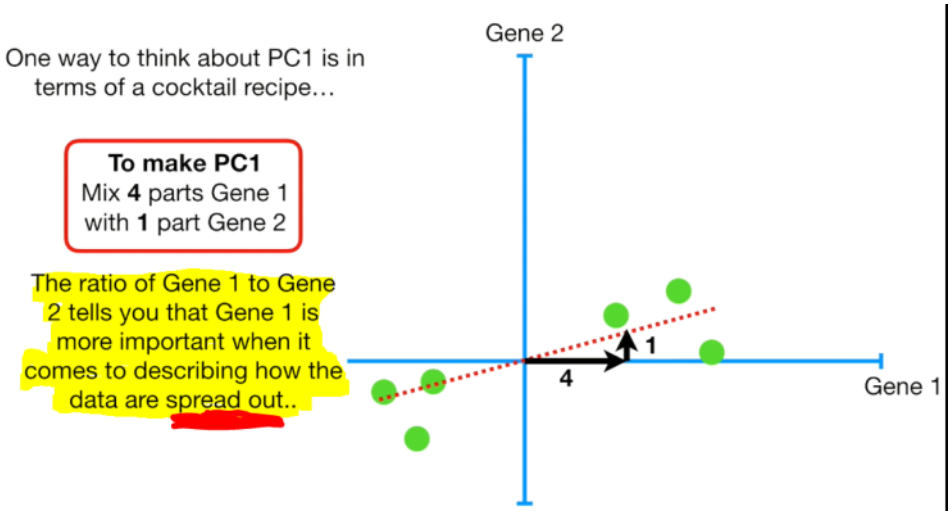
5_SPREAD OF DATA δ LINEAR COMBINATION :

⇒ 1/4 = 0.25







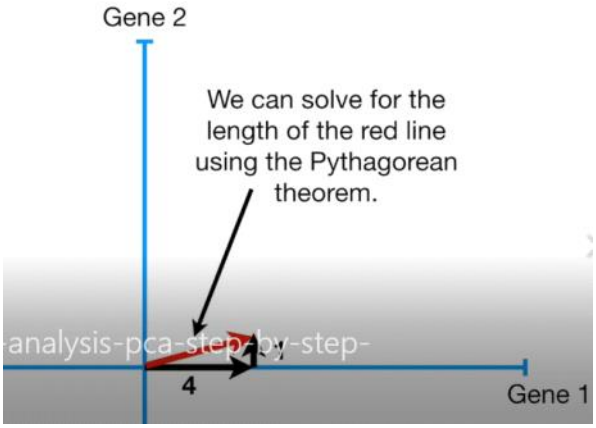


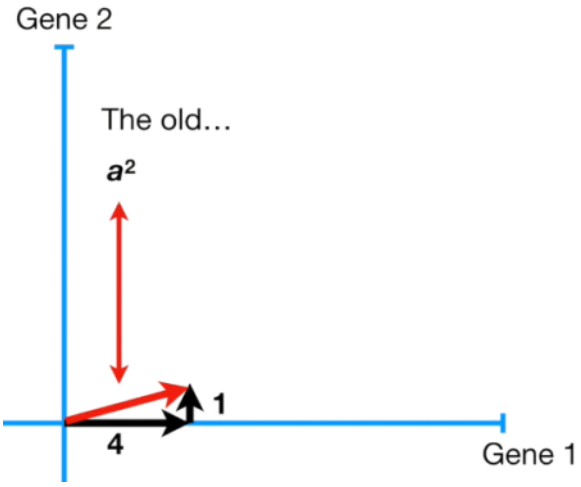
To make PC1
Mix **4** parts Gene 1 with **1** part Gene 2

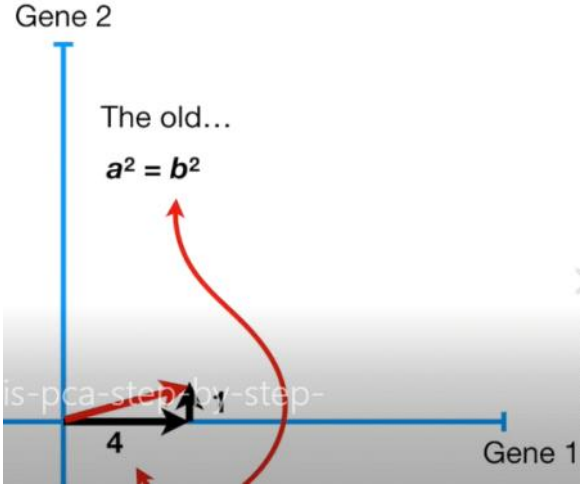
Terminology Alert!!!!
Mathematicians call this cocktail recipe a "*linear combination*" of Genes 1 and 2.

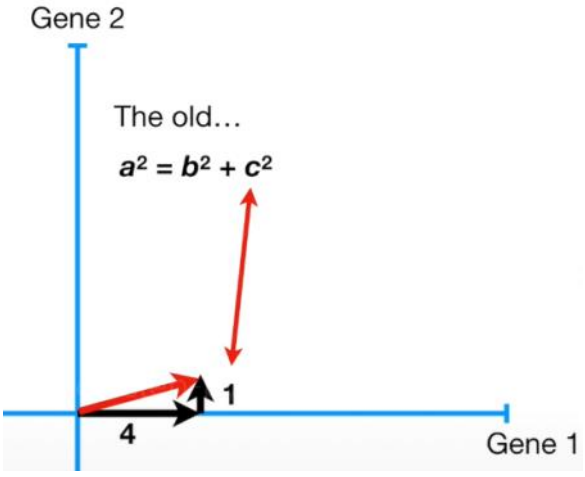
someone says, "PC1 is a linear combination of variables…"

6_CALCULATE LENGTH OF PC1:


Gene 2
We can solve for the length of the red line using the Pythagorean theorem.
4
Gene 1


Gene 2
The old…
a^2
1
4
Gene 1


Gene 2
The old…
$a^2 = b^2$
4
Gene 1

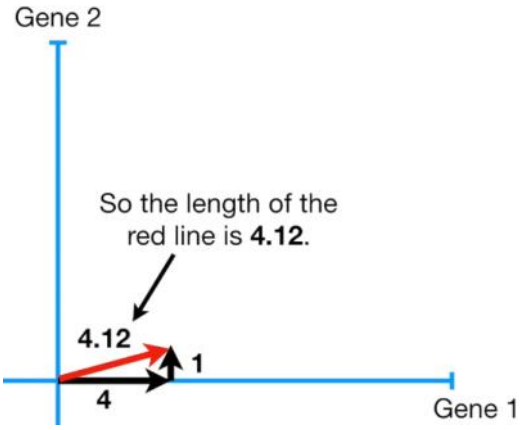


$$a^2 = b^2 + c^2$$

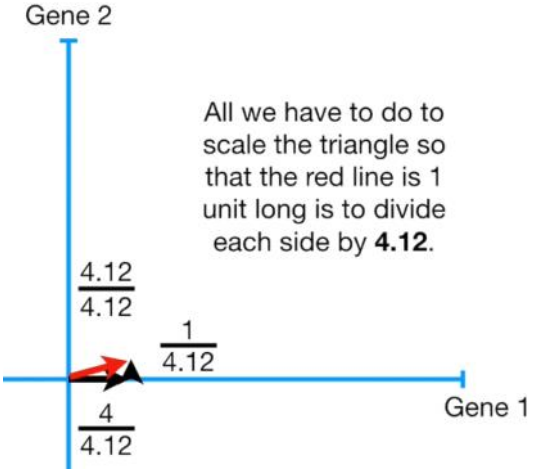
$$a^2 = 4^2 + 1^2$$

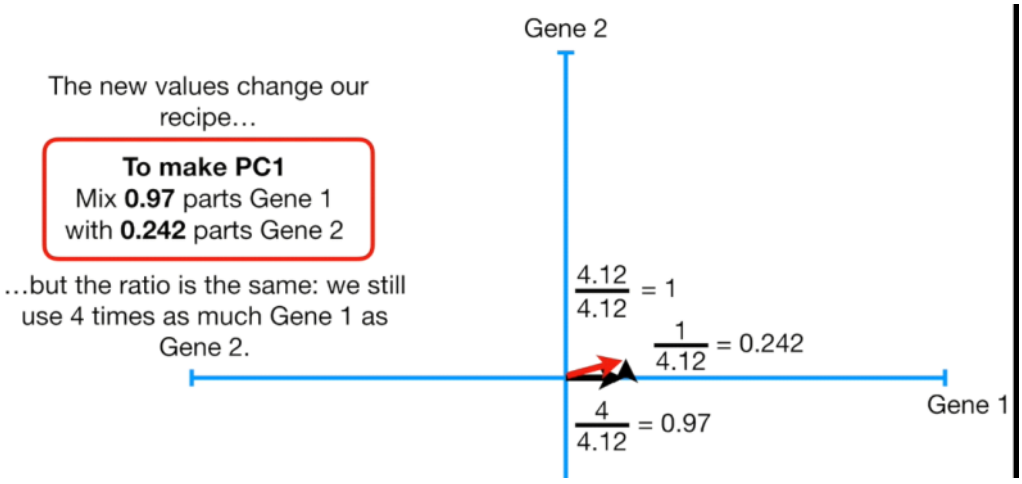
$$a = \sqrt{4^2 + 1^2} = 4.12$$

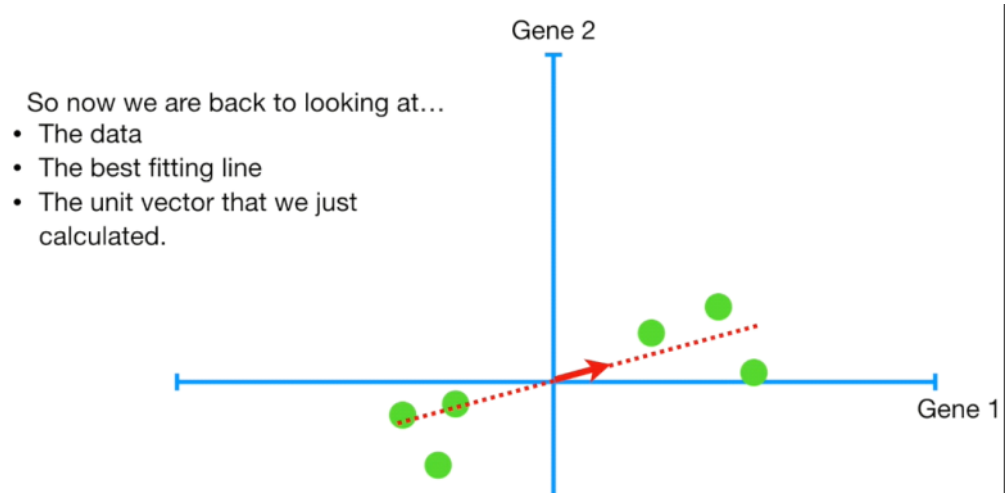
7_EIGEN VECTOR AND EIGEN VALUE:



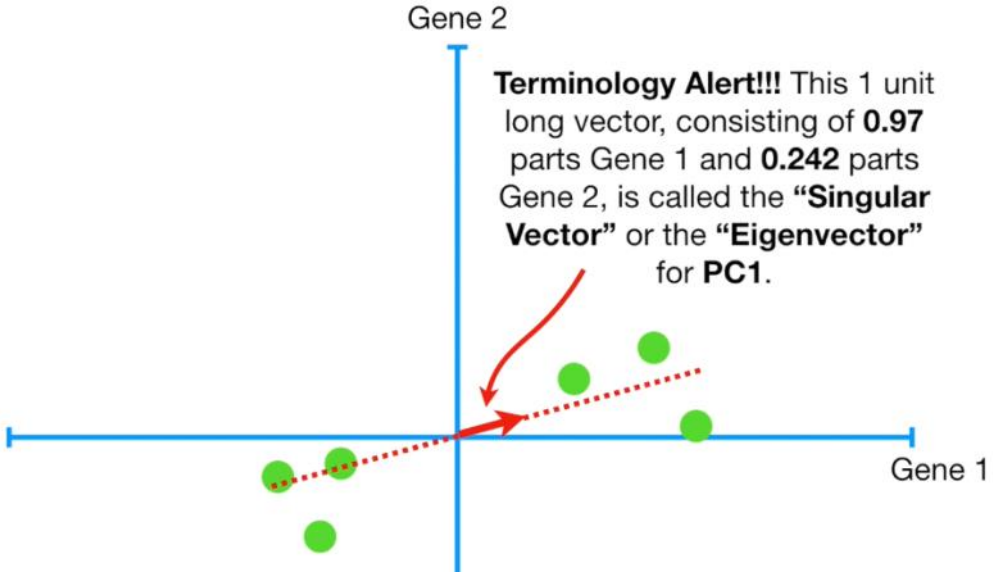
To make the length of PC1 = 1, so that it becomes a singular vector :







EIGEN VECTOR :



LOADING SCORES :

To make PC1
Mix 0.97 parts Gene 1
with 0.242 parts Gene 2

…and the proportions of each gene are called **"Loading Scores"**.

EIGEN VALUES :

Also, while I'm at it, PCA calls the SS(distances) for the best fit line the **Eigenvalue for PC1**…

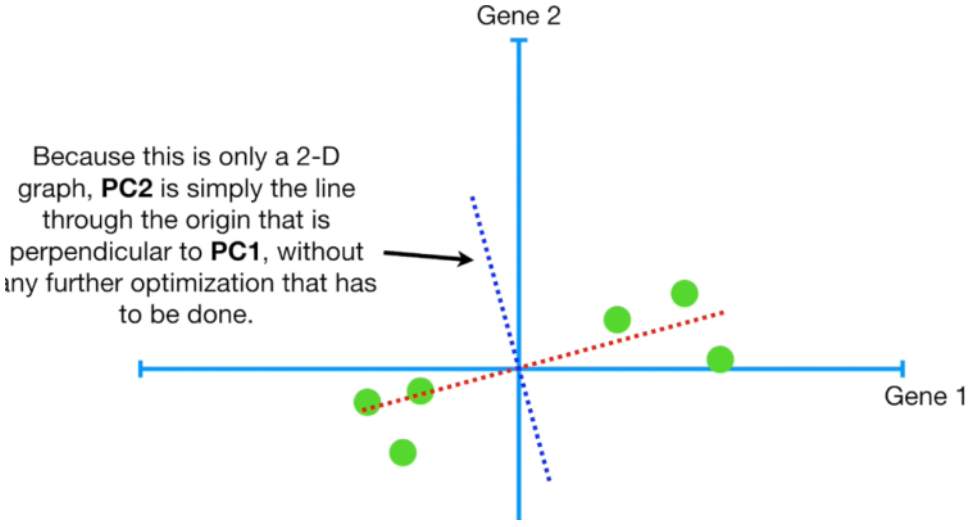
$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)

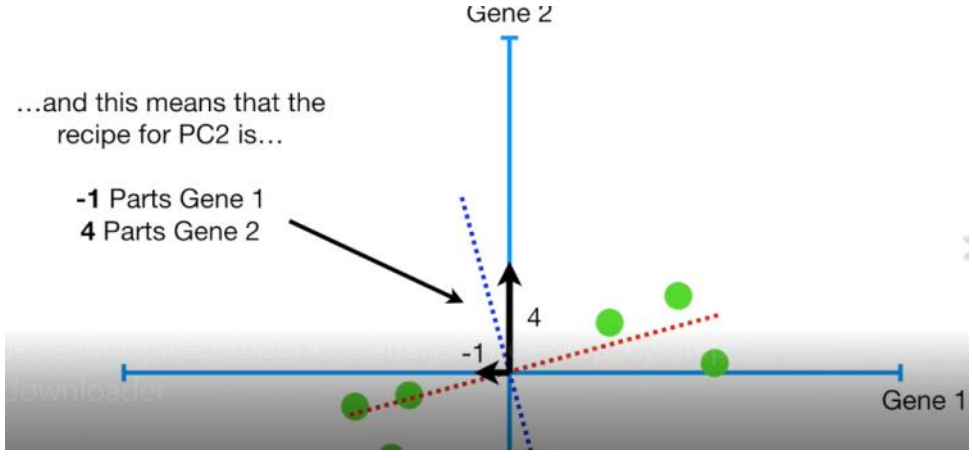
SS(distances for PC1) = Eigenvalue for PC1

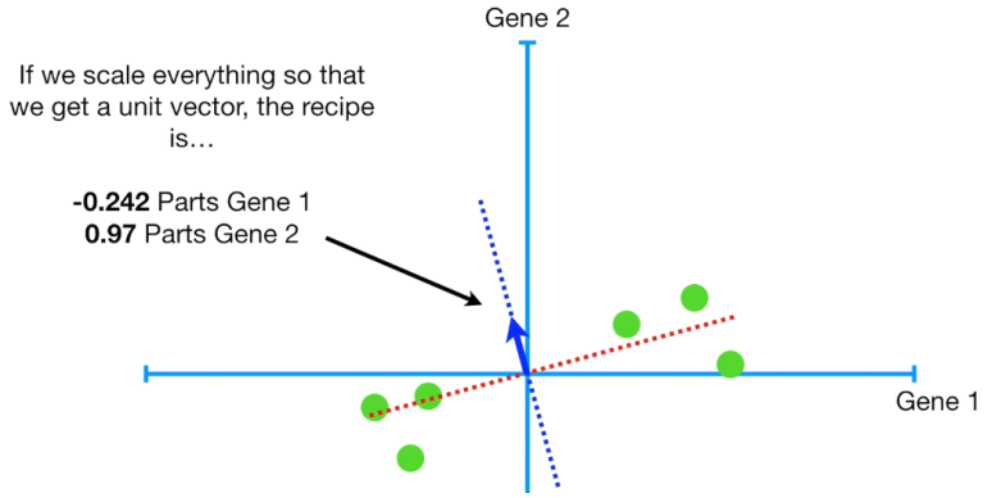
$\sqrt{\text{Eigenvalue for PC1}}$ = Singular Value for PC1

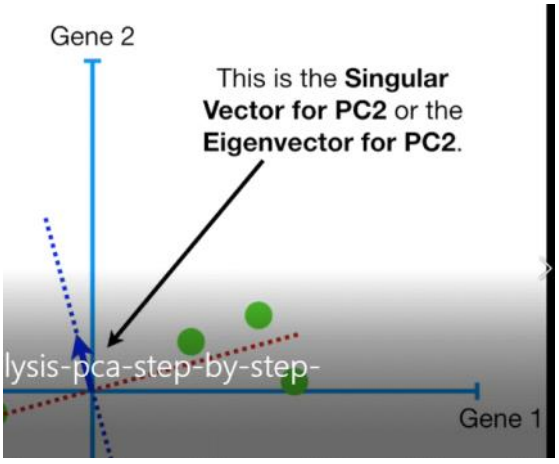
8_FINDING ORTHOGONAL LINE (TO BEST FITTED LINE) **(PC2):**

Now that we've got **PC1** all figured out let's work on **PC2**!!!





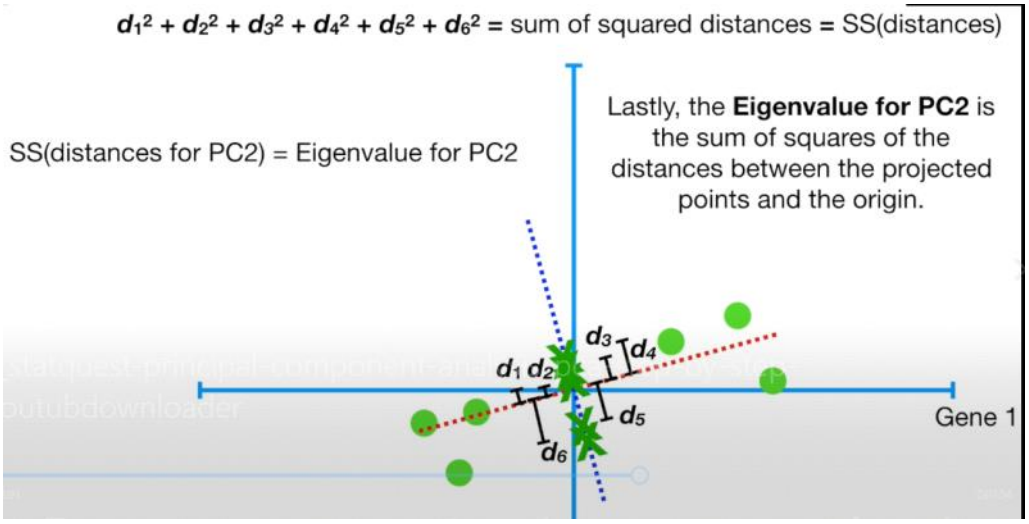


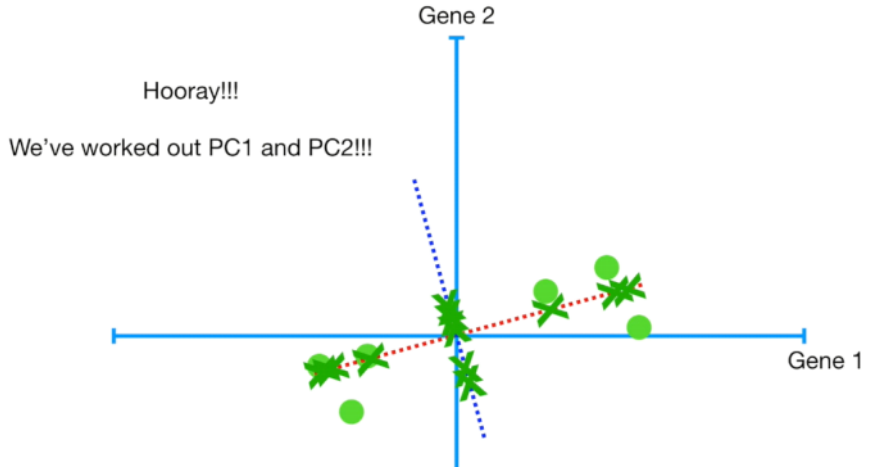


These are the **Loading Scores for PC2**.

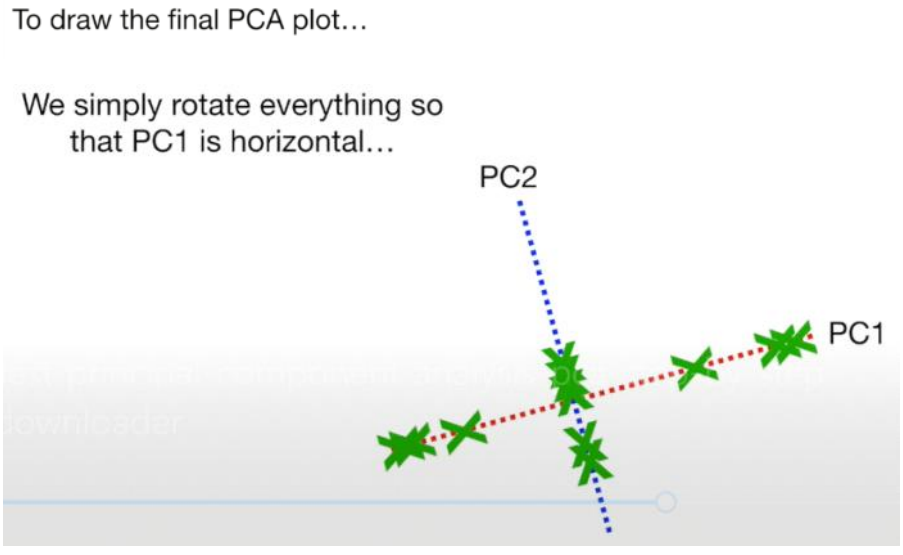
-0.242 Parts Gene 1
0.97 Parts Gene 2

They tell us that, in terms of how the values are projected onto PC2, Gene 2 is 4 times as important as Gene 1.

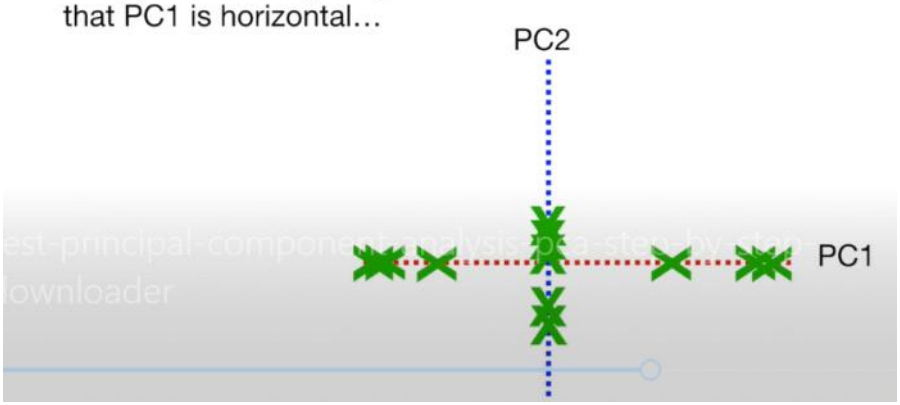

$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)
SS(distances for PC2) = Eigenvalue for PC2
Lastly, the **Eigenvalue for PC2** is the sum of squares of the distances between the projected points and the origin.
d_1 d_2
d_3
d_4
d_5
d_6
Gene 1


Gene 2
Hooray!!!
We've worked out PC1 and PC2!!!
Gene 1

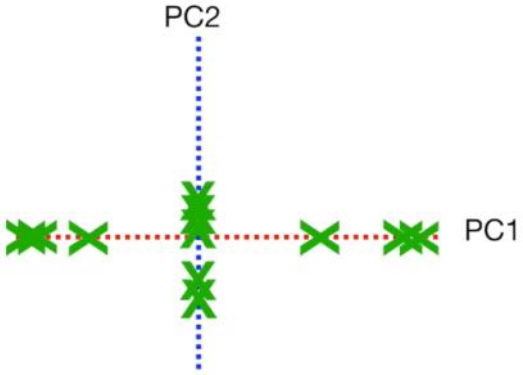
9_PCA PLOT :


To draw the final PCA plot…
We simply rotate everything so that PC1 is horizontal…
PC2
PC1

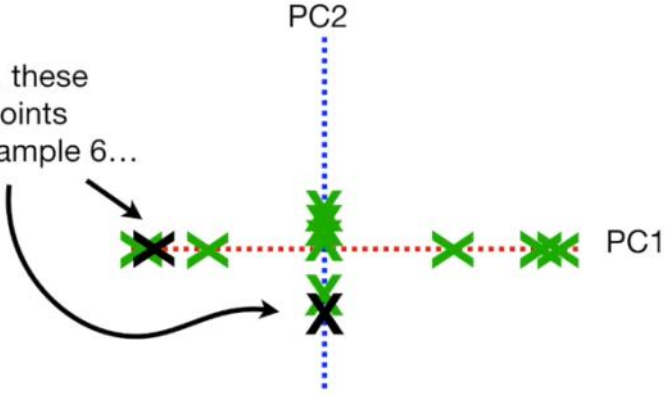
We simply rotate everything so that PC1 is horizontal…



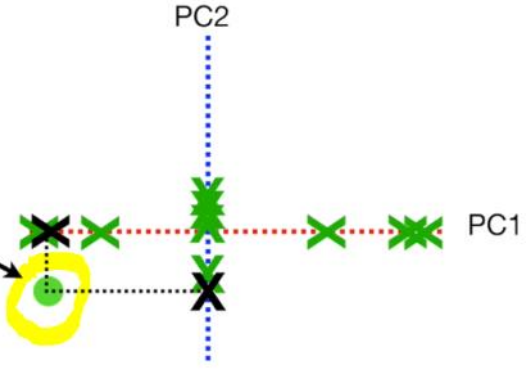
…then we use the projected points to find where the samples go in the PCA plot.

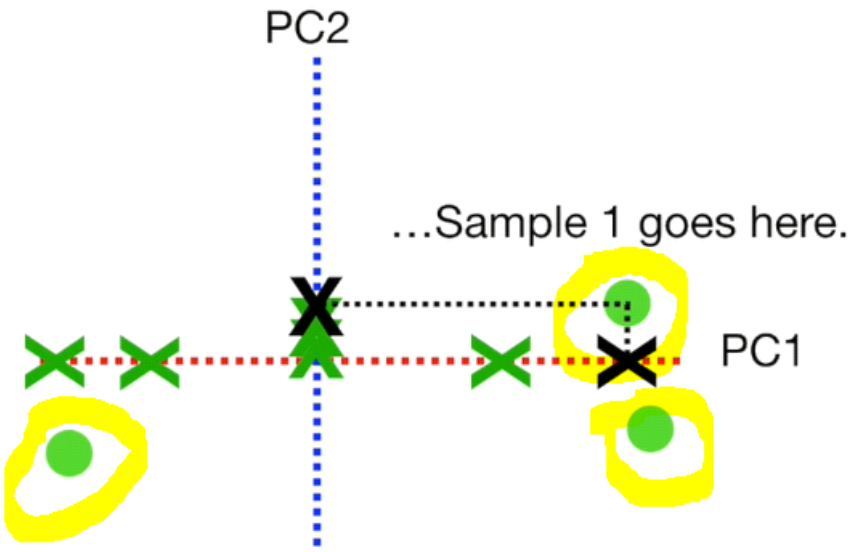


For example, these projected points correspond to Sample 6…

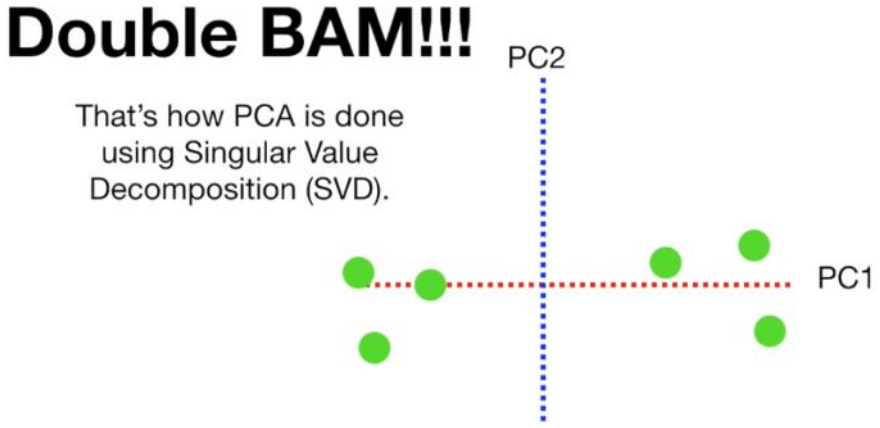


..so Sample 6 goes here.

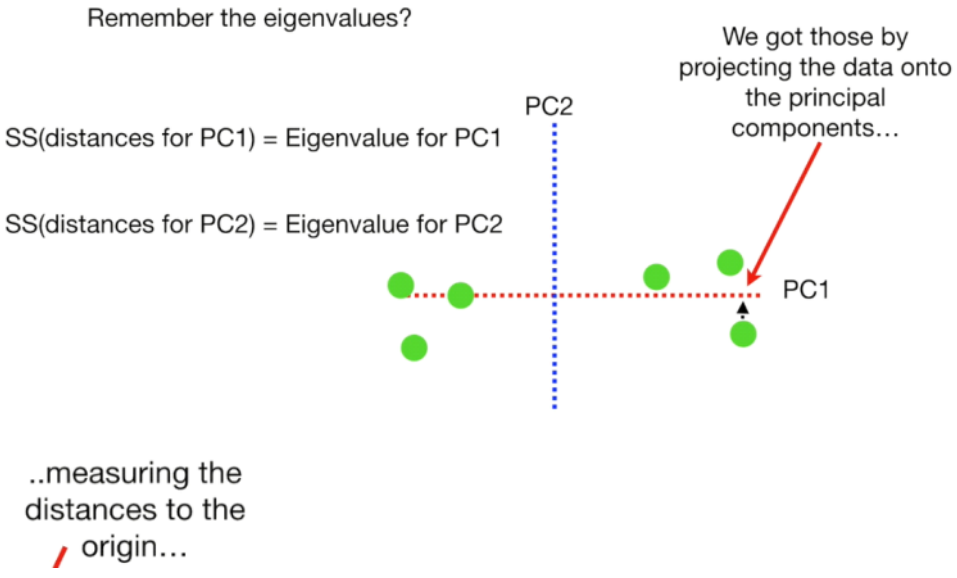




The **singular value decomposition** (SVD)
allows us to discover some of the same kind of information as the Eigen decomposition.



10_VARIATION :



…then squaring and adding them together.

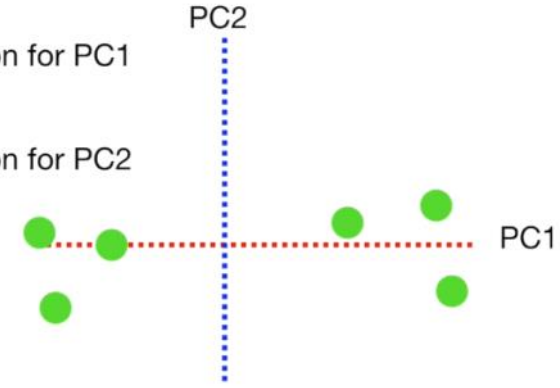
SS(distances for PC1) = Eigenvalue for PC1

SS(distances for PC2) = Eigenvalue for PC2

We can convert them into variation around the origin (0, 0) by dividing by the sample size minus 1 (i.e. n - 1).

$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Variation for PC1}$$

$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Variation for PC2}$$



2 For the sake of the example, imagine that the Variation for **PC1 = 15**, and the variation for **PC2 = 3**.

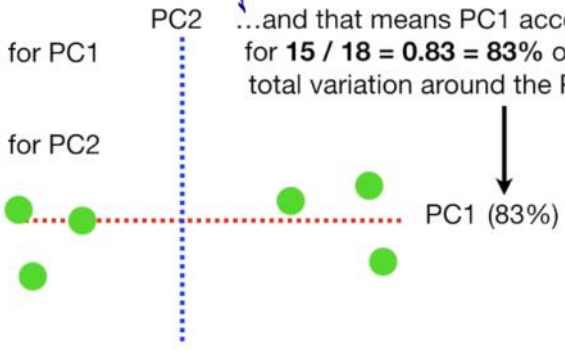
3 That means that the total variation around both PCs is **15 + 3 = 18**…

1

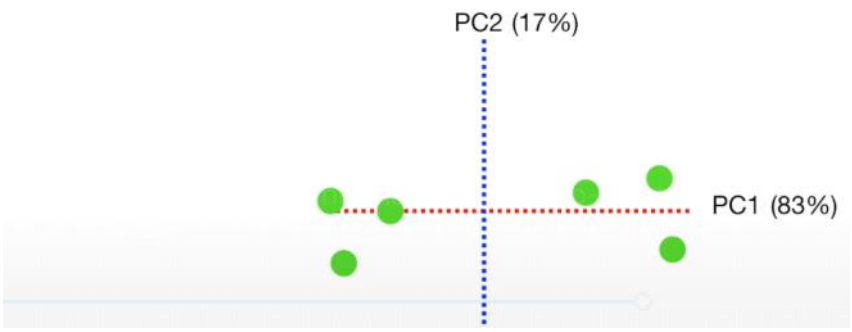
$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Variation for PC1}$$

$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Variation for PC2}$$

4 …and that means PC1 accounts for **15 / 18 = 0.83 = 83%** of the total variation around the PCs.



PC2 accounts for **3 / 18 = 0.17 = 17%** of the total variation around the PCs.



TERMINOLOGY ALERT!!!! A **Scree Plot** is a graphical representation of the percentages of variation that each PC accounts for.