Principal Component Analysis

Introduction

If you work with a lot of variables, this can present problems. In technical terms, you want to "reduce the dimension of your feature space." By reducing the dimension of your feature space, you have fewer relationships between variables to consider.

Somewhat unsurprisingly, ***reducing*** the ***dimension*** of the feature space is called "***dimensionality reduction***." There are many ways to achieve dimensionality reduction, but most of these techniques fall into one of two classes:

- Feature Elimination
- Feature Extraction

Feature elimination is what it sounds like: we reduce the feature space by eliminating features. Advantages of feature elimination methods include simplicity and maintaining interpretability of your variables. As a disadvantage, though, you gain no information from those variables you've dropped.

Feature extraction

Principal component analysis is a technique for *feature extraction*—so it combines our input variables in a specific way, then we can drop the "least important" variables while still retaining the most valuable parts of all of the variables.

PCA takes the dataset with a lot of dimensions and flattens it to less (2 or 3) dimensions. It tries to find a meaningful way to flatten the data by focusing on the things that are different between cells.

Why & where to use PCA?

1. Use PCA when you want to **reduce the number** of variables to help reduce the overfitting. When you work with PCA the data will be transformed, which is great for dimension reduction and could result in better regression models. PCA can capture some influence in the target variable due to the interaction of two explanatory variables.
2. Use it when you want to ensure your variables are **independent** of one another?
3. When you are comfortable making your independent variables **less interpretable**?

How does PCA work?

- We are going to calculate a matrix that summarizes how our variables all relate to one another.
    - Calculating the **mean of each feature**.
    - **Covariance Matric** calculation
- We'll then break this matrix down into two separate components: direction and magnitude.
    - Calculating the **Eighen values** and **corresponding Eighen vectors.**
    - The principle component with **greater Eighen value is more important**

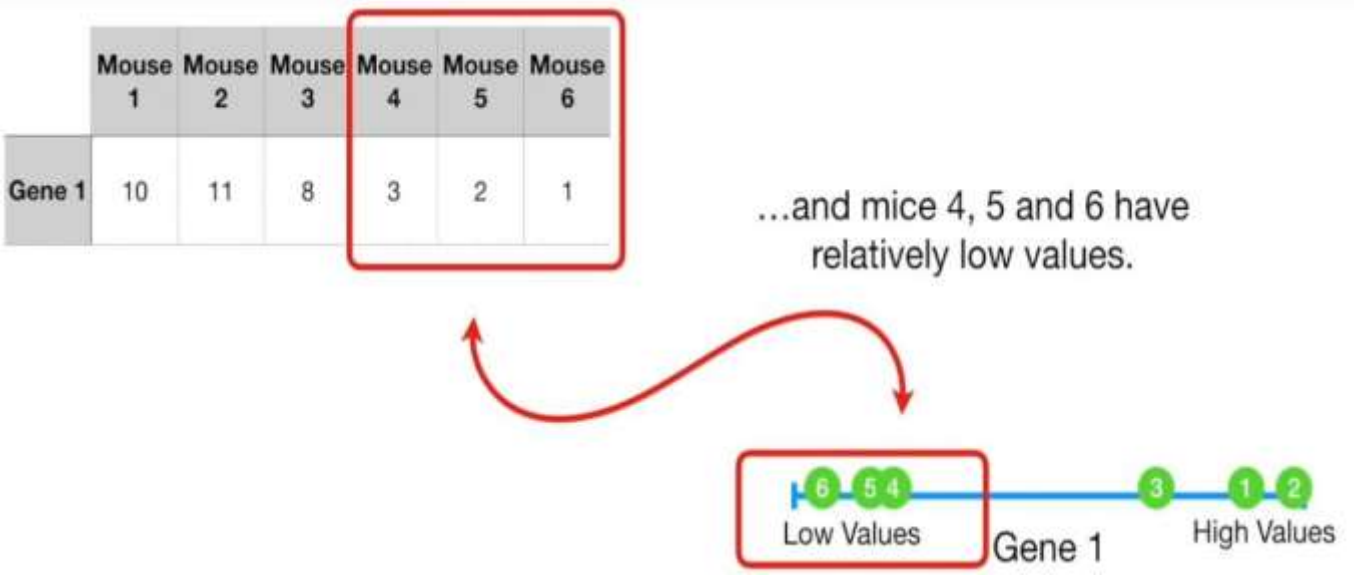
Understand PCA through less technical example

	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

Mouse 1…n = Records Gene1

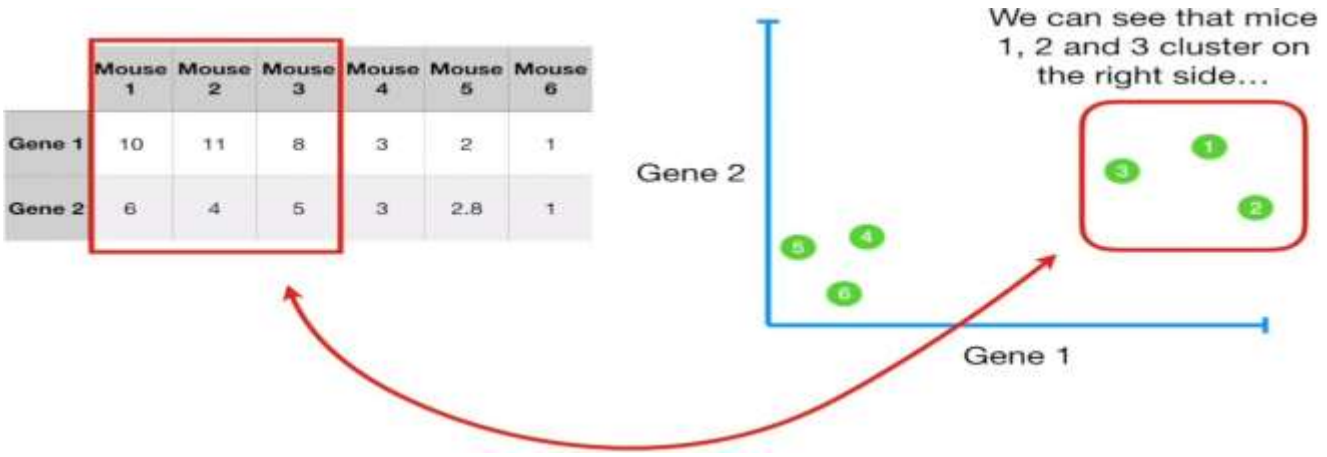
and Gene2 =Features

If we measure the records against single Gene, then we can plot the data on single number line. Mice 1,2,3 have relatively high values and mice 4,5,6 have relatively low values. In other words, mice 1,2,3 are similar to each other than mice 4,5,6.

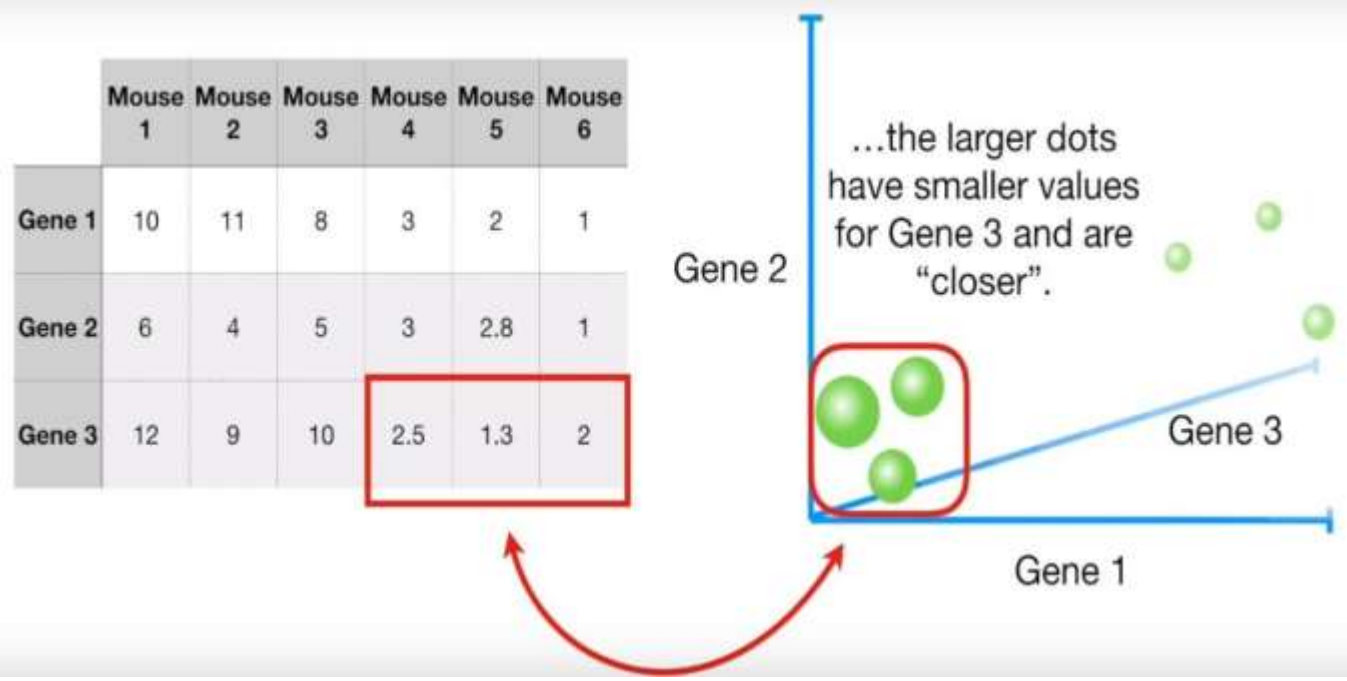


If we measured two genes then we can plot the data in on a two- dimensional x/y graph, i.e. Gene-1 is on the x-axis and Gene-2 on the y-axis.

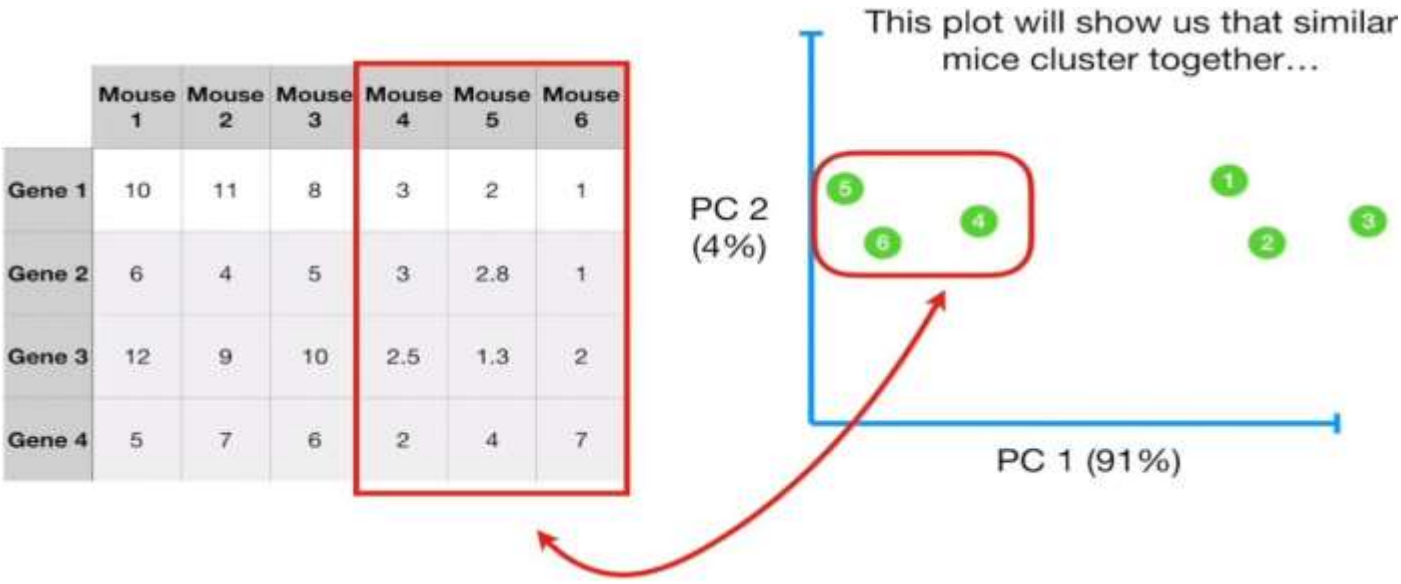
We can see mice 1,2,3 are clustered in right hand side and mice 4,5,6 cluster on the lower left hand side.



If we measured three Genes, we would add another axis to the graph and it look like 3-Dimensional.



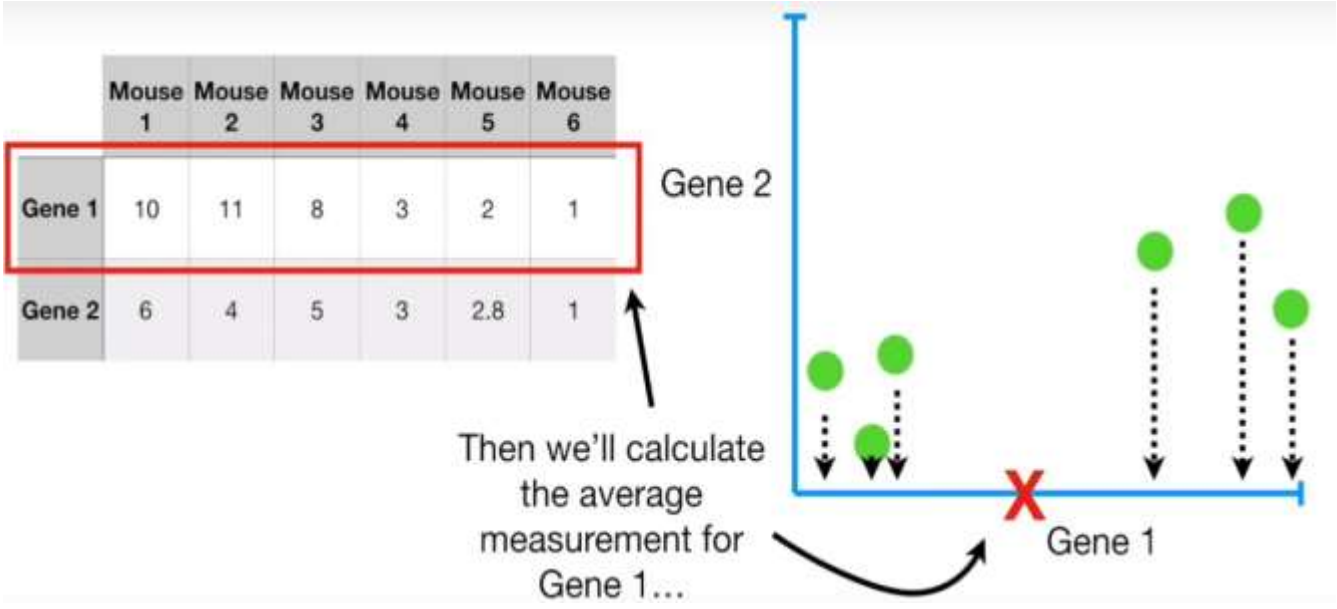
If we measured 4-Genes and it requires 4- dimension, no longer able to plot. So, we are going to solve this problem with PCA that can take 4 or mores Gene measurements (features) and make a 2-D (not fixed) PCA plot.



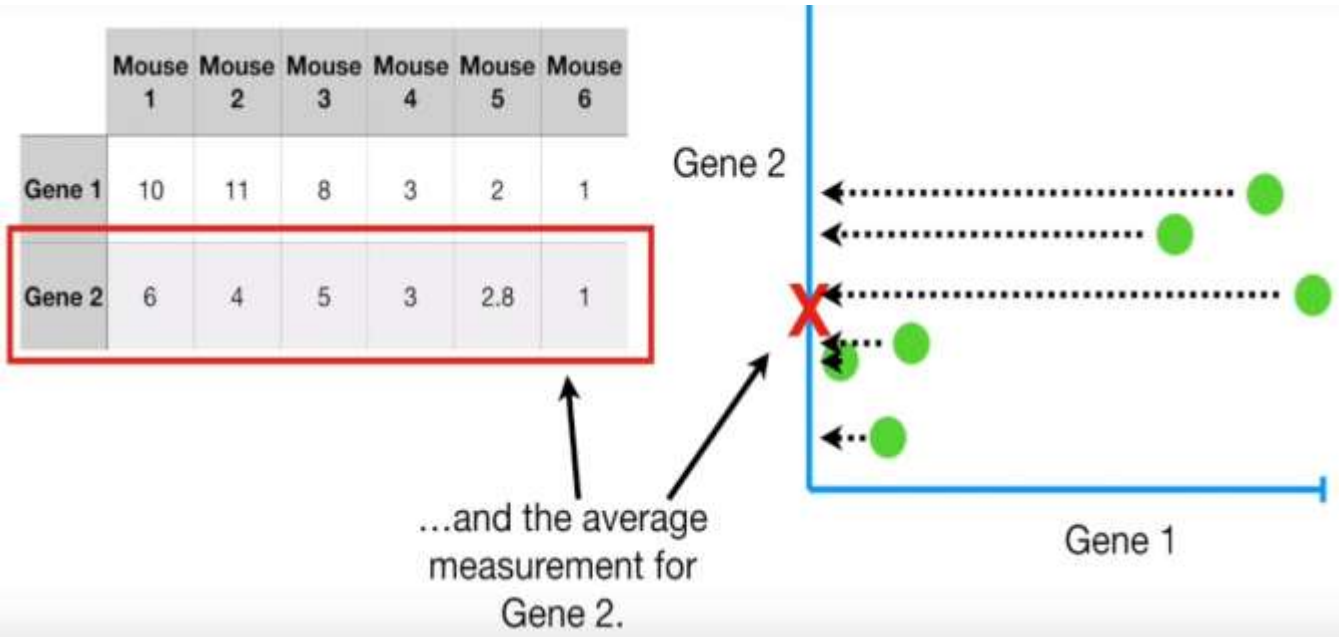
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2
Gene 4	5	7	6	2	4	7

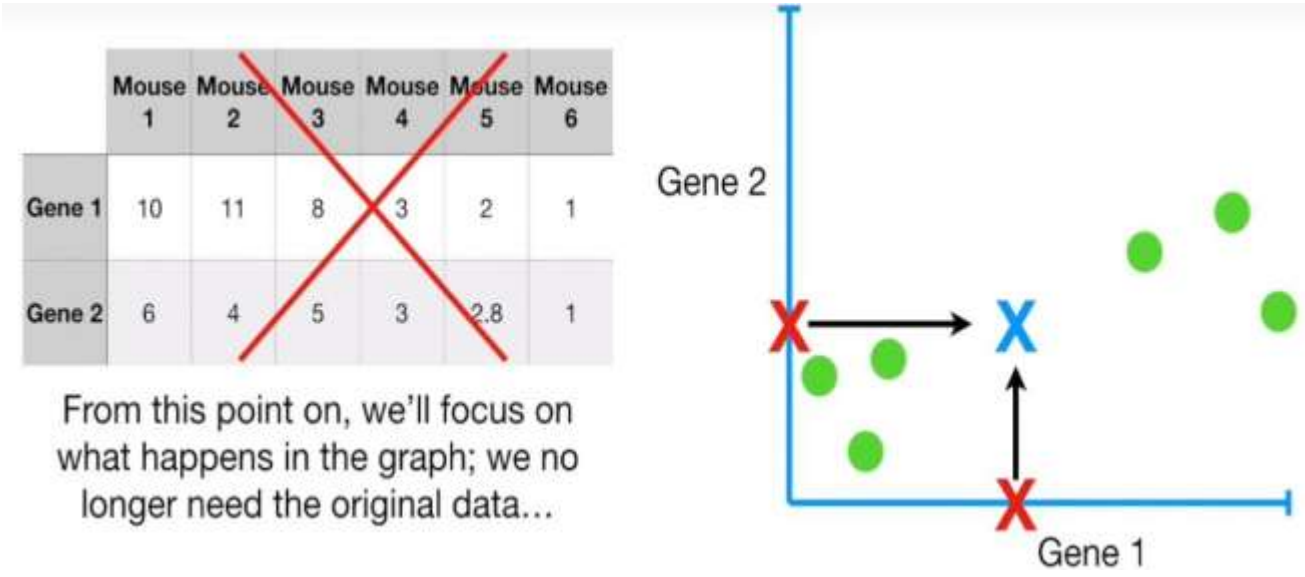
X-axis explain 91% variation and y-axis 4% of data.

To understand PCA working, take 2-D data:

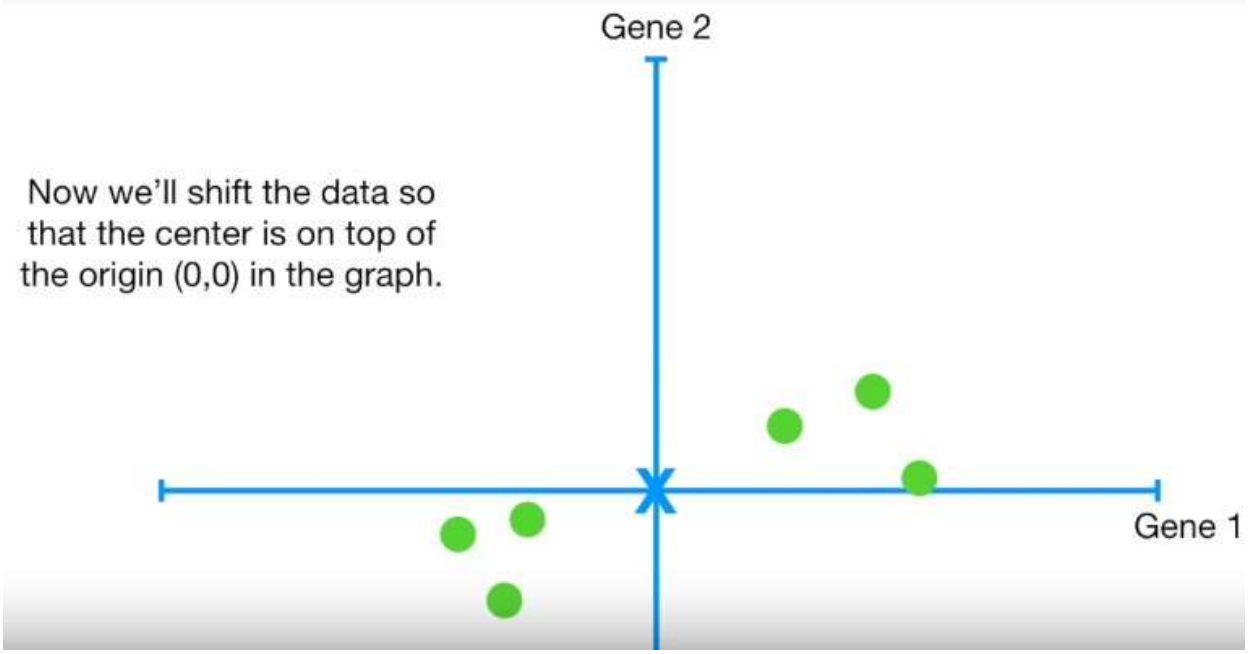


	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



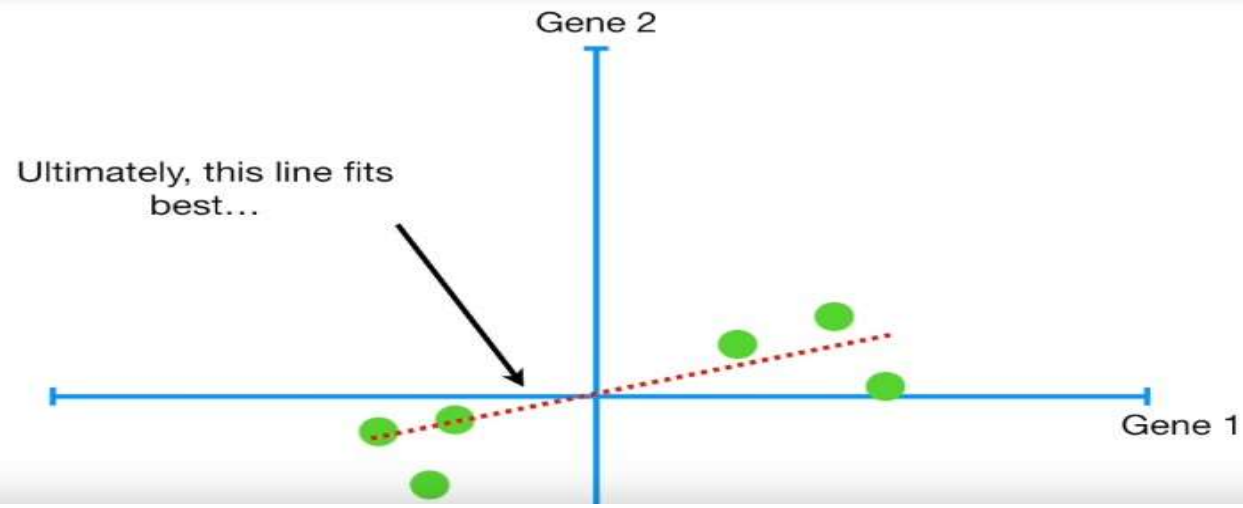


With the average value, we can calculate the center of the data.



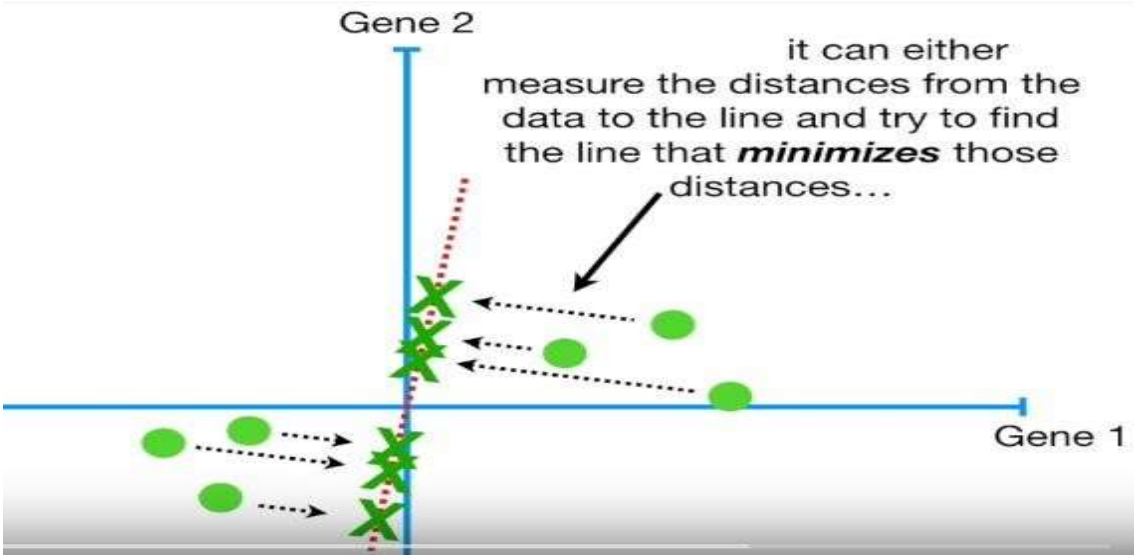
Note: Shifting the data did not change how the data points are positioned *relative* to each other.

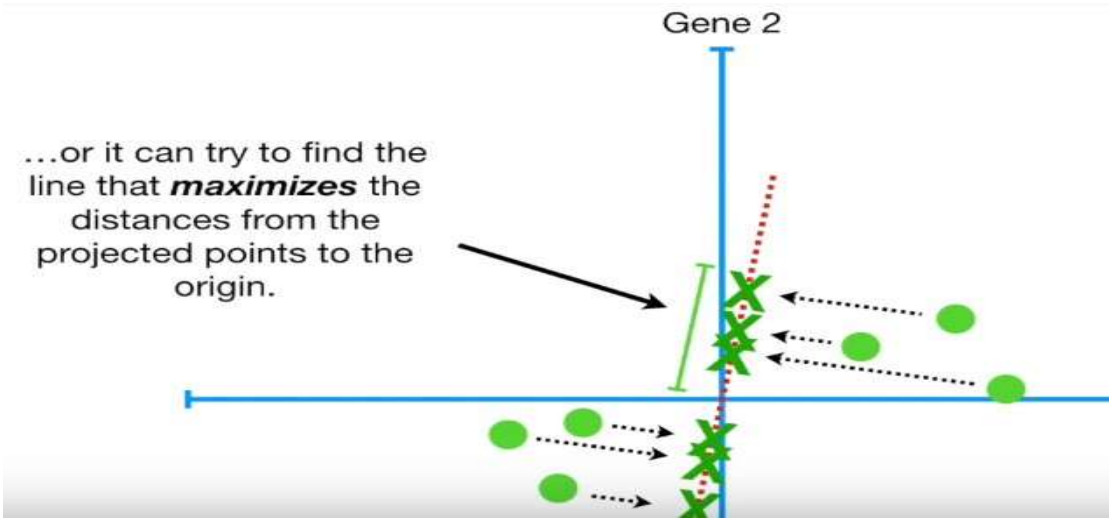
Now, the data are centered on the origin, we can try to fit a line to it. We start by drawing the random line that goes through the origin. Then we rotate the line until it best fits the data.

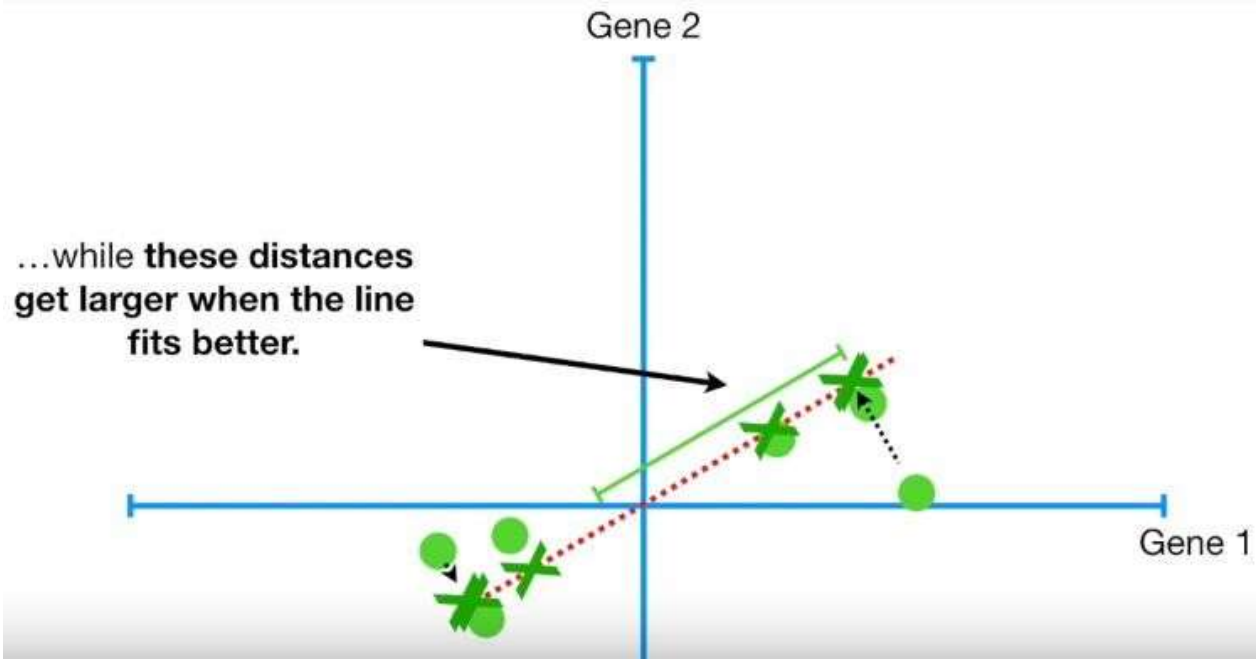


How PCA decides if a fit is good or not?

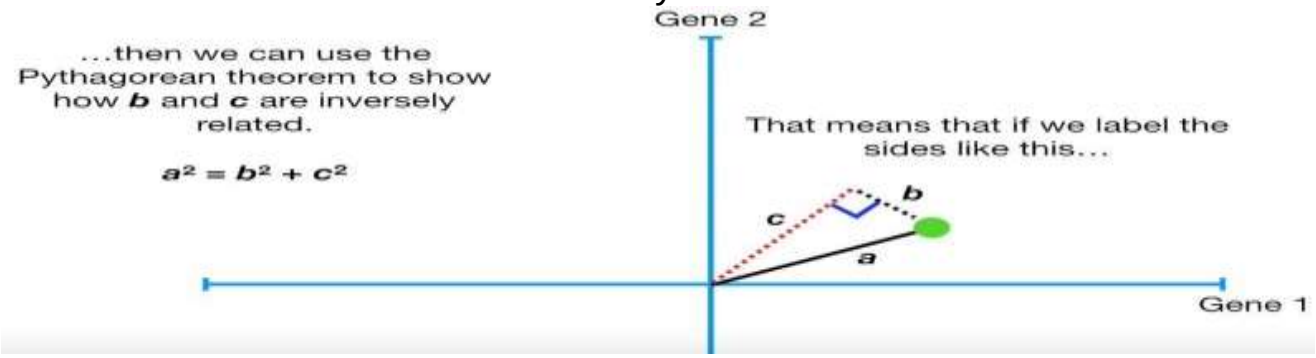
There are two ways:


Gene 2
it can either measure the distances from the data to the line and try to find the line that *minimizes* those distances...
Gene 1


Gene 2
...or it can try to find the line that *maximizes* the distances from the projected points to the origin.

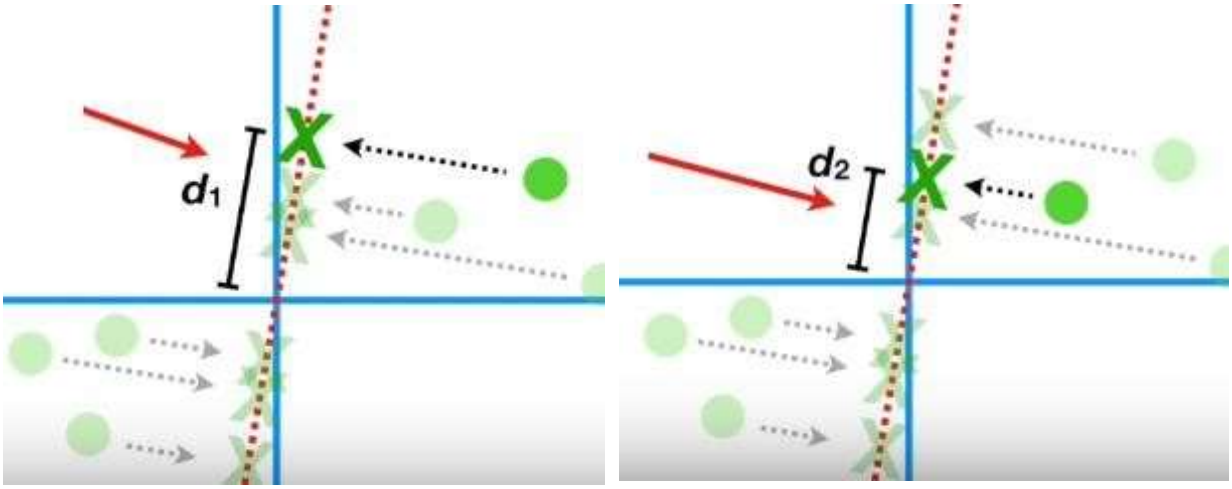

Gene 2
...while **these distances get larger when the line fits better.**
Gene 1

Let's understand in mathematical way:

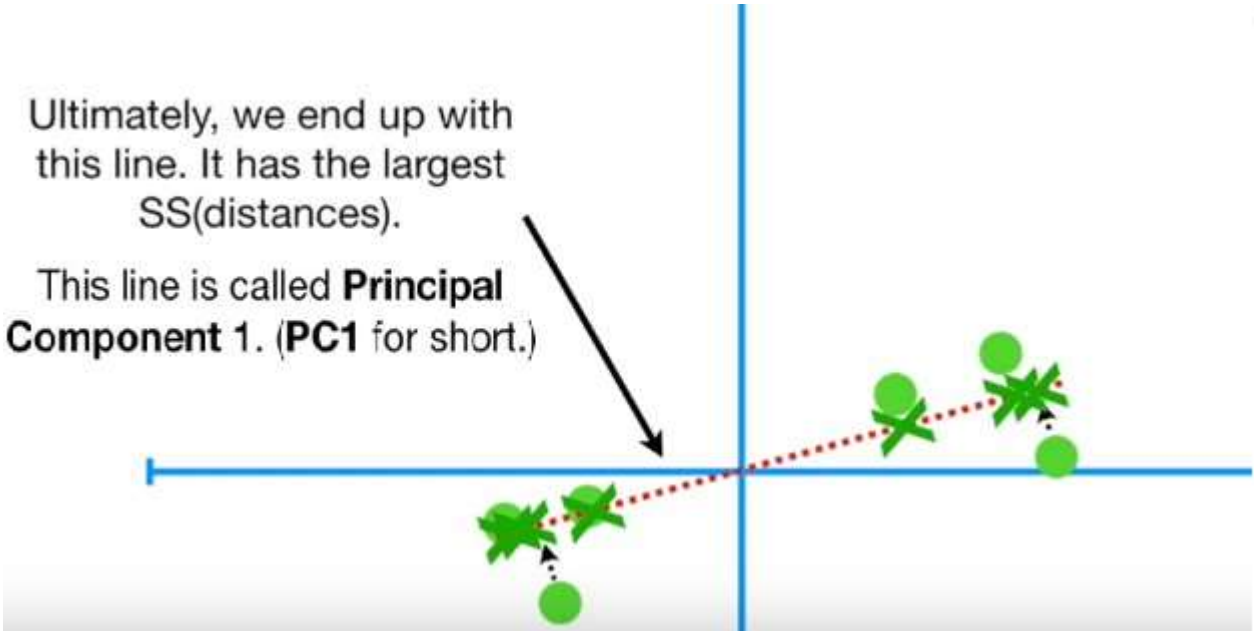


a^2 doesn't changes.

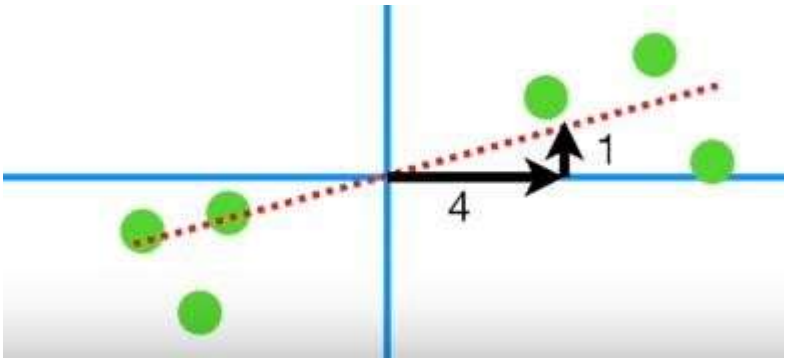
Note: Its actually easier to calculate 'C', the distance from the projected point to the origin, so PCA finds the best fitting line by maximizing the sum of the squared distance from the projected points to the origin.



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



PC1 has the slop of 0.25. In other words for every 4 units go out along the x-axis (Gene-1), we go up 1-unit along the y-axis (Gene- 2). That means data are mostly spread out along the x-axis (Gene-1) and only little bit spread out along the y-axis (Gene-2).

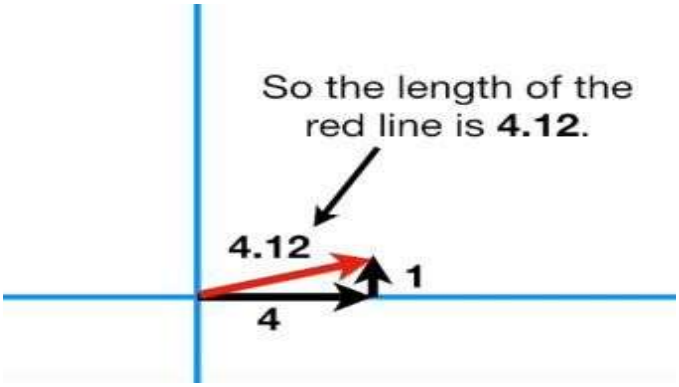


To make PC-1

Mix 4-part of gene-1

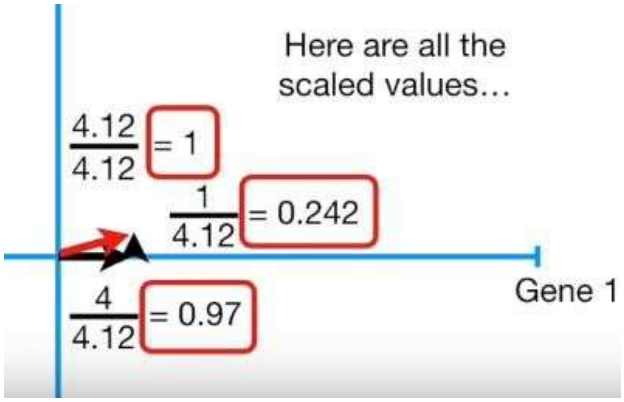
With 1-part of

Mathematically, we call it linear combination of Gene-1 and Gene-2 (PCA is a liner combination of variables).



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

We do PCA with SVD (single value decomposition), we scaled PCA so that length =1 (unit vector).



To make PC-1

Mix 0.97-part of gene-1 With 0.242-part of Gene-2

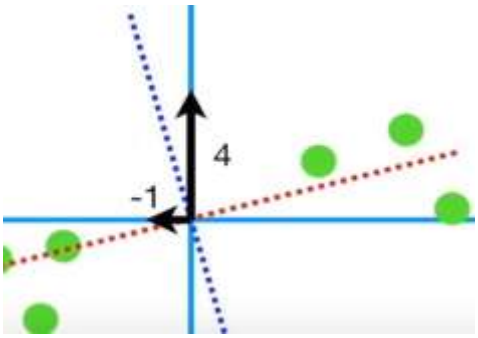
This one-unit long vector, consisting of 0.97 parts of gene-1 and 0.242 parts of Gene-2, is called the **SINGULAR VECTOR** or **EIGENVECTOR** for PC1. The proportion of each gene are called **LOADING SCORES**.

SS(distances for PC1) = Eigenvalue for PC1

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

Lets work on PC2

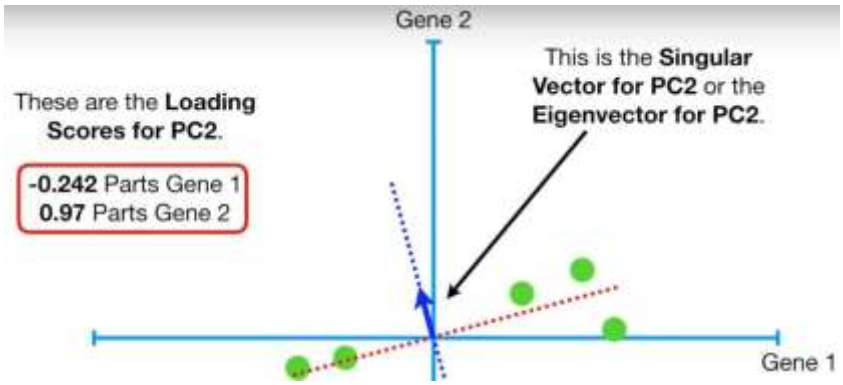
PC2 is simply the line through the origin that is perpendicular to PC1.



To make PC-2

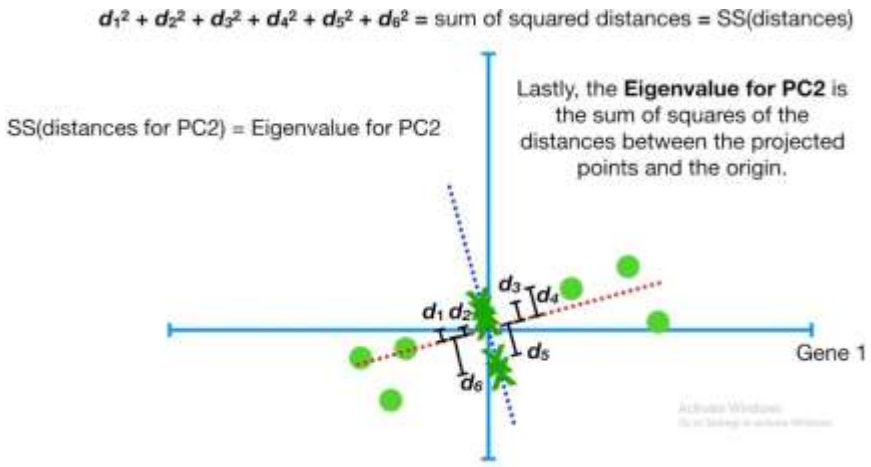
-1 part of gene-

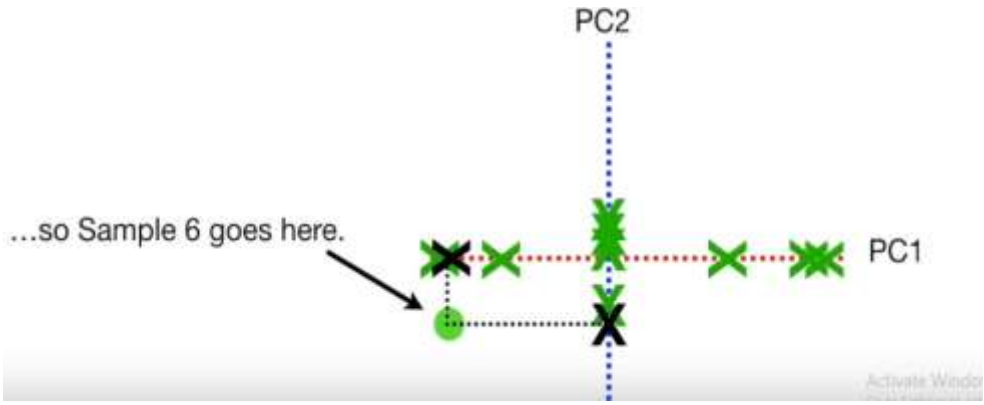
1 4 part of



Gene-2 is 4 times as important as Gene-1.

Now, we rotate everything such that PC1 become horizontal and points are project accordingly.





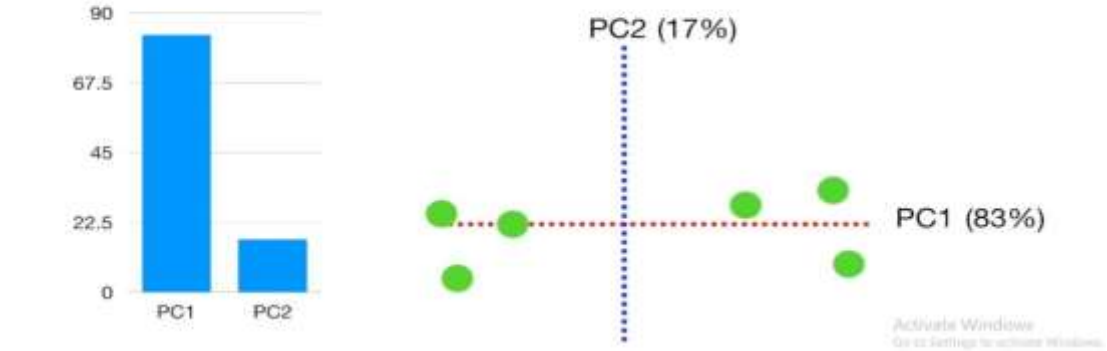
Calculate total variance calculated by PCA

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

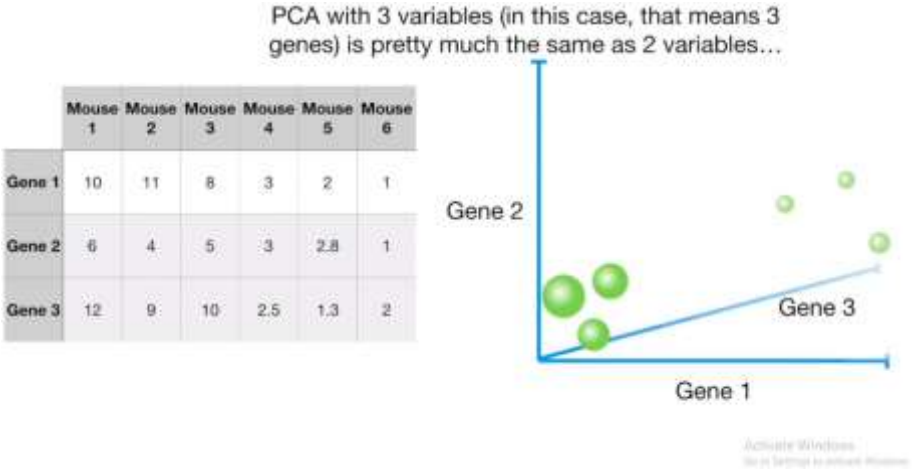
$$\frac{SS(\text{distances for PC2})}{n-1} = \text{Variation for PC2}$$

Suppose, The variation of PC-1 is = 15 The variation of PC-2 is = 3
Total variations around both PCs=18, PC-1 account 15/18 (83%) of total variations around the PCs Similarly, PC-2, 17%

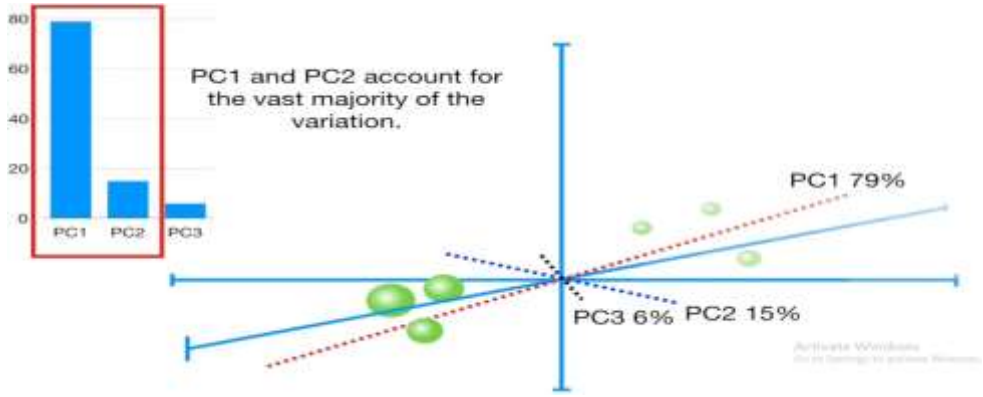
TERMINOLOGY ALERT!!!! A **Scree Plot** is a graphical representation of the percentages of variation that each PC accounts for.



Consider little more complex example (3-D)



	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



Applications of PCA

- **Quantitative Finance**: PCA is a methodology to reduce the dimensionality of a complex problem. Say, a fund manager has 200 stocks in his portfolio. To analyze these stocks quantitatively a stock manager will require a co-relational matrix of the size 200 * 200, which makes the problem very complex. With PCA the task would become much easier.
- PCA has been used on **Medical Data** to show correlation of Cholesterol with low density lipo-protein.
- PCA has been used in the **Detection and Visualization of Computer Network Attacks**.