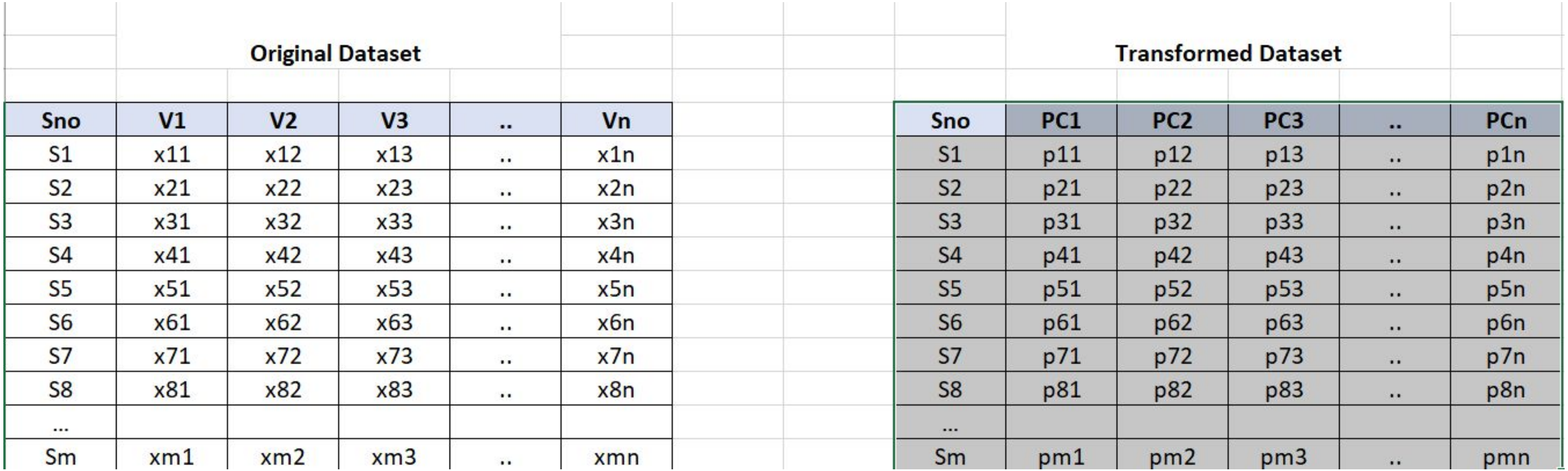
PCA is meant to transform the data into projections onto principal components



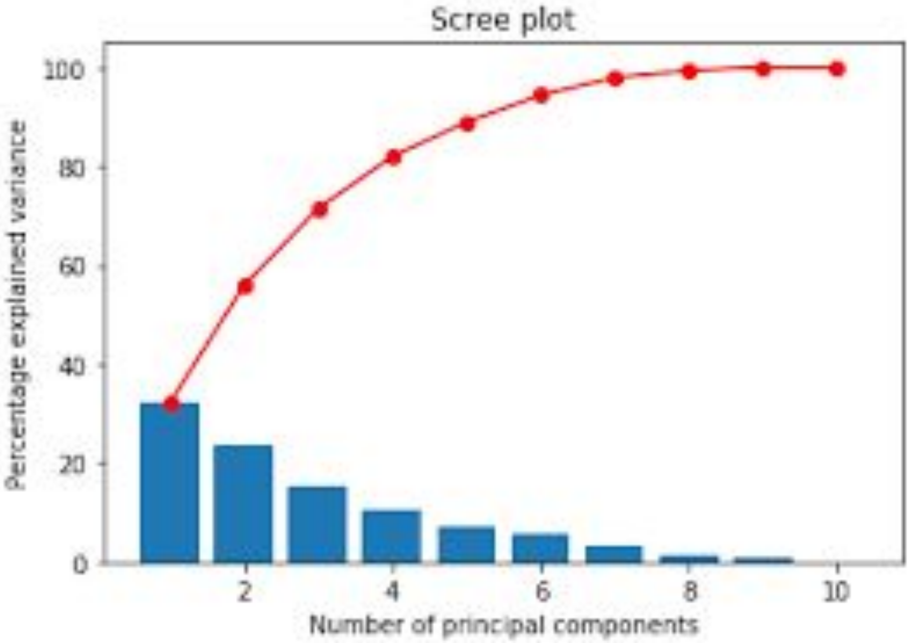
Original Dataset

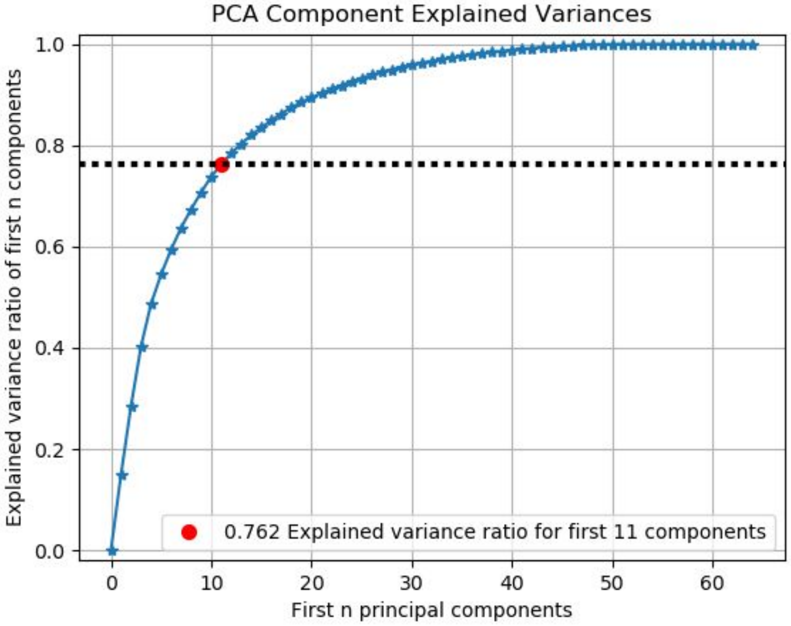
Sno	V1	V2	V3	..	Vn
S1	x11	x12	x13	..	x1n
S2	x21	x22	x23	..	x2n
S3	x31	x32	x33	..	x3n
S4	x41	x42	x43	..	x4n
S5	x51	x52	x53	..	x5n
S6	x61	x62	x63	..	x6n
S7	x71	x72	x73	..	x7n
S8	x81	x82	x83	..	x8n
...					
Sm	xm1	xm2	xm3	..	xmn

Transformed Dataset

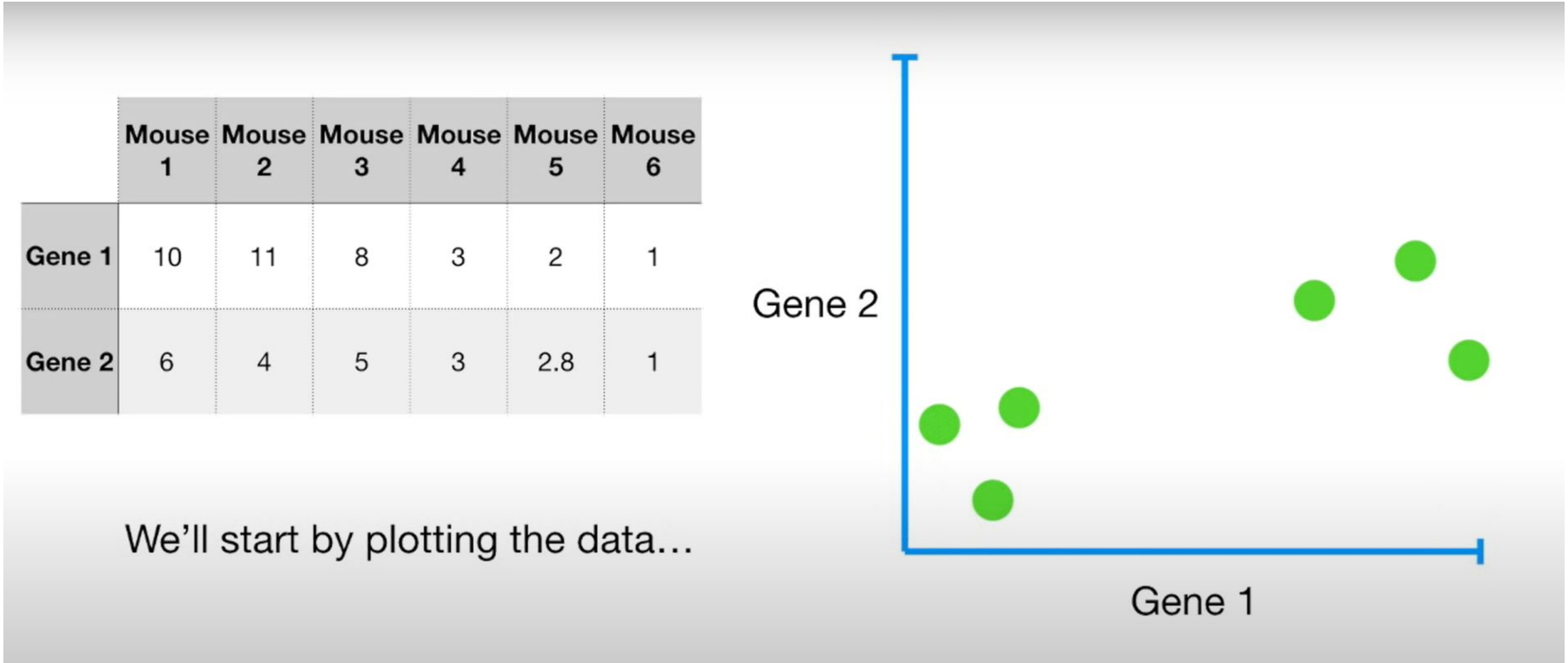
Sno	PC1	PC2	PC3	..	PCn
S1	p11	p12	p13	..	p1n
S2	p21	p22	p23	..	p2n
S3	p31	p32	p33	..	p3n
S4	p41	p42	p43	..	p4n
S5	p51	p52	p53	..	p5n
S6	p61	p62	p63	..	p6n
S7	p71	p72	p73	..	p7n
S8	p81	p82	p83	..	p8n
...					
Sm	pm1	pm2	pm3	..	pmn

Number of components that explain the cumulative variance is obtained from the **Scree plot**





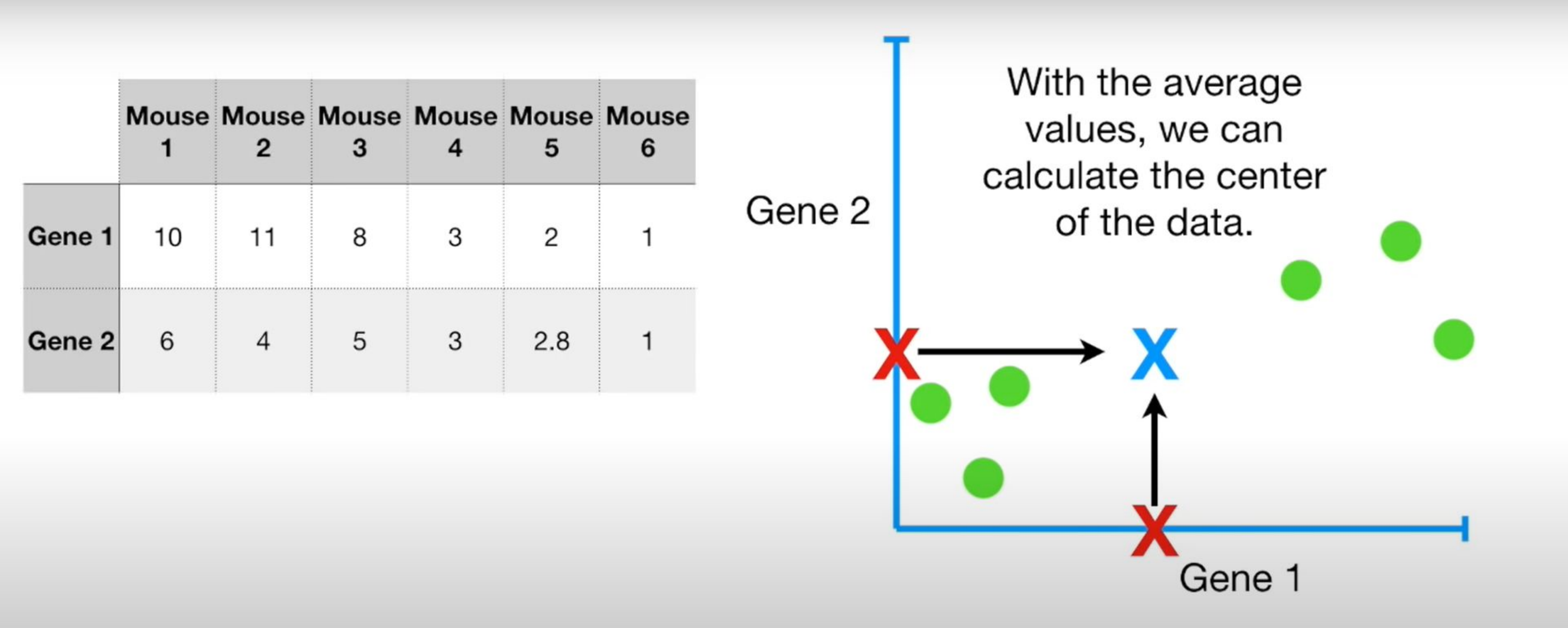
Sample Dataset that has data for 6 mouse and plotting Gene1 and Gene 2



	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

We'll start by plotting the data…

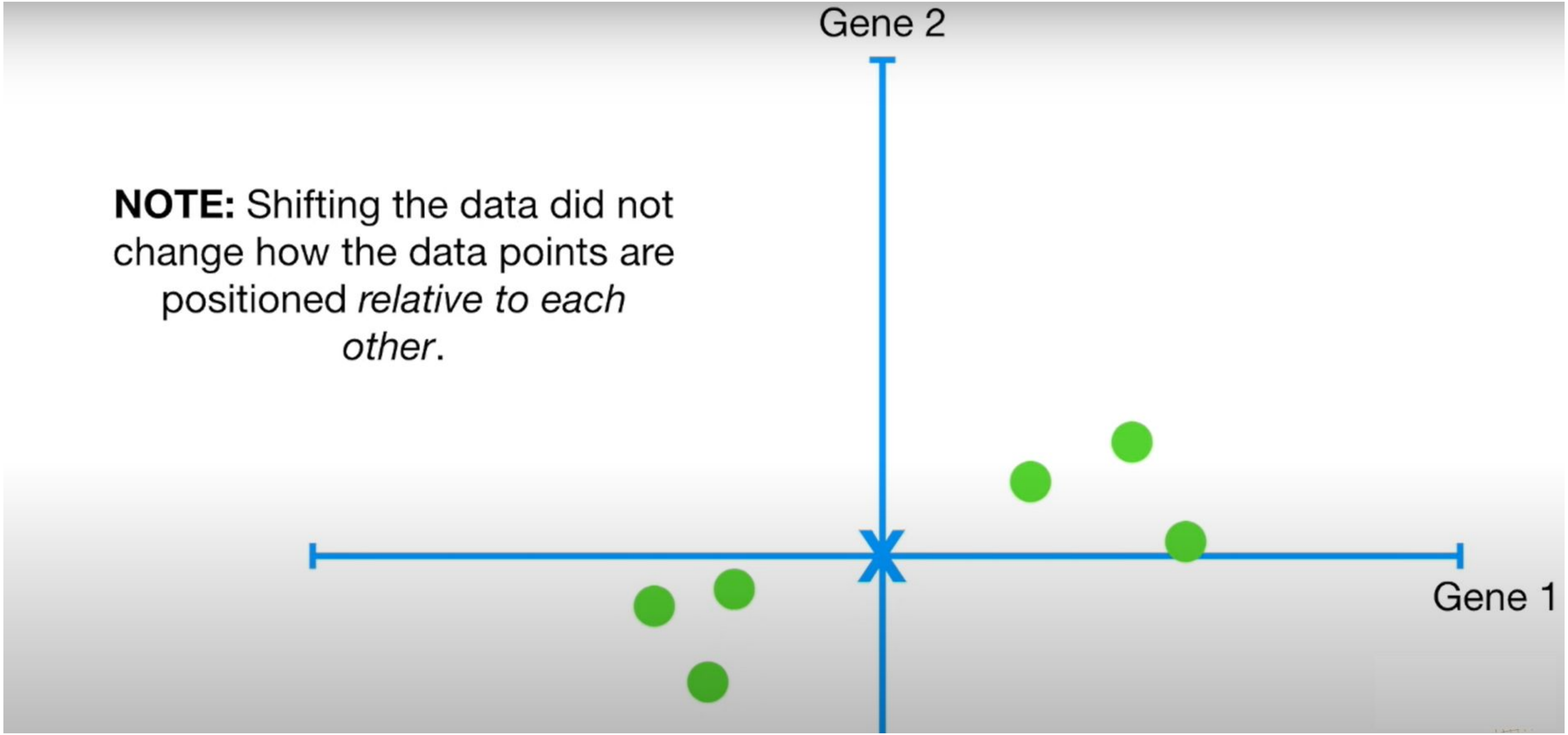
Compute the average of Gene1 and Gene2



	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

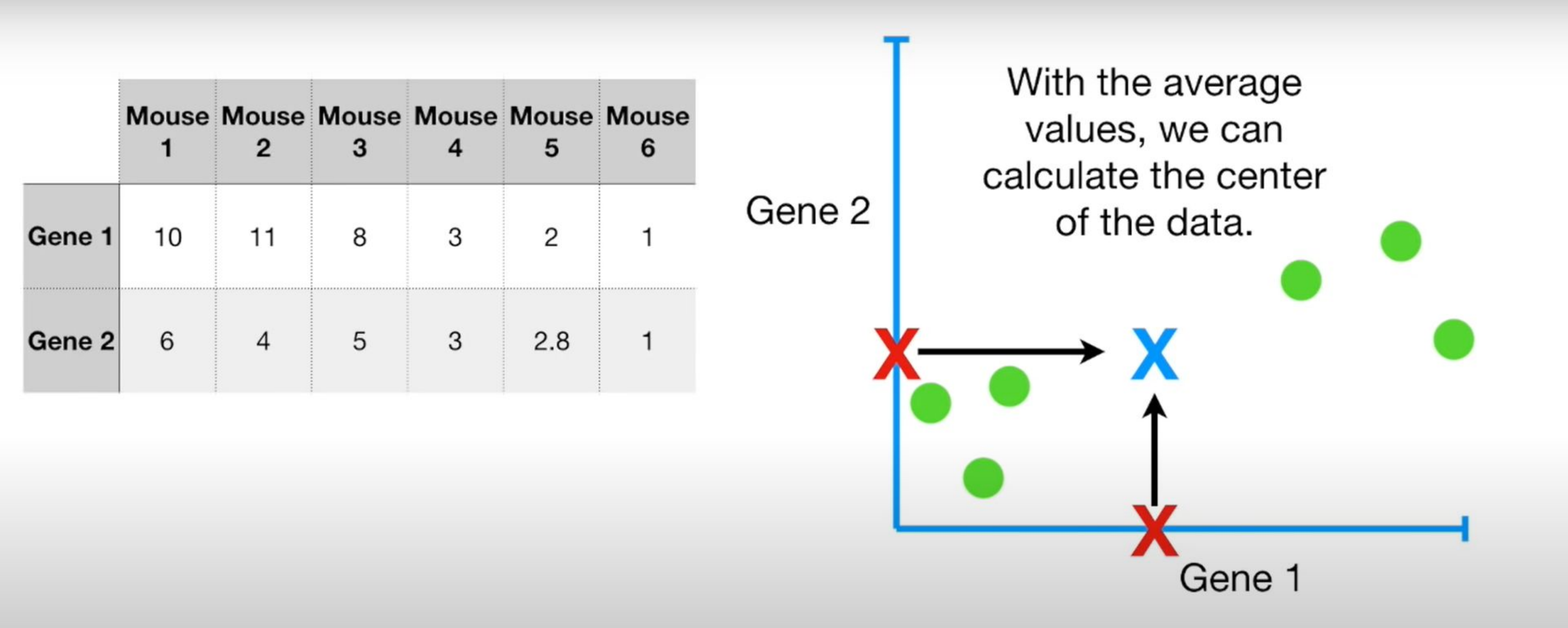
With the average values, we can calculate the center of the data.

Shifting the origins to the average of the two data points



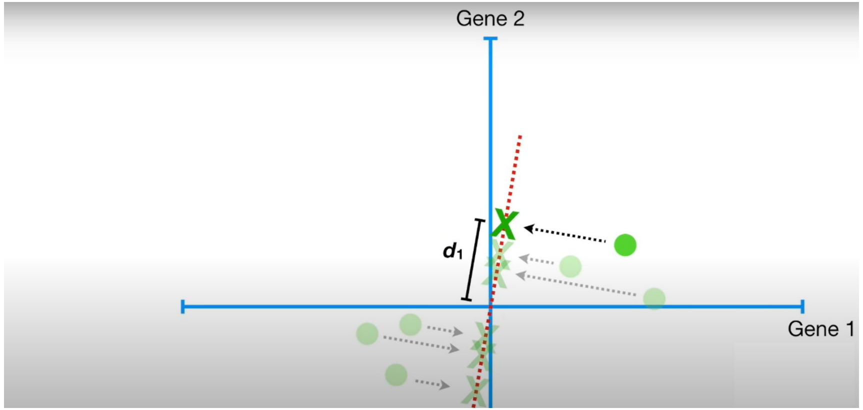
NOTE: Shifting the data did not change how the data points are positioned *relative to each other*.

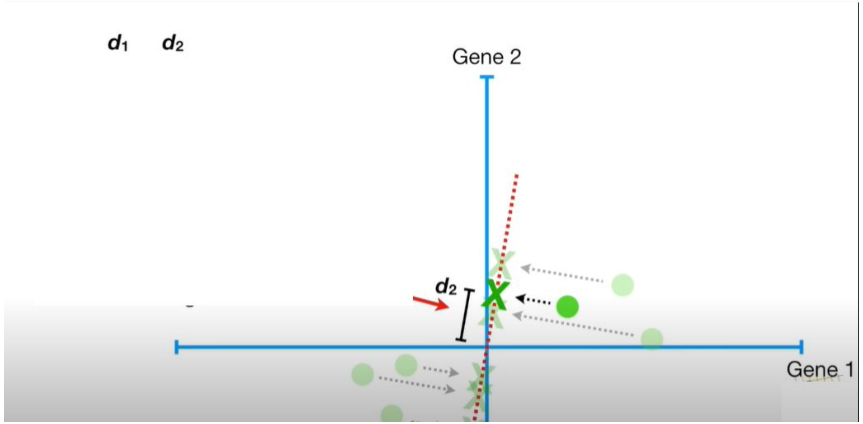
Compute the average of Gene1 and Gene2

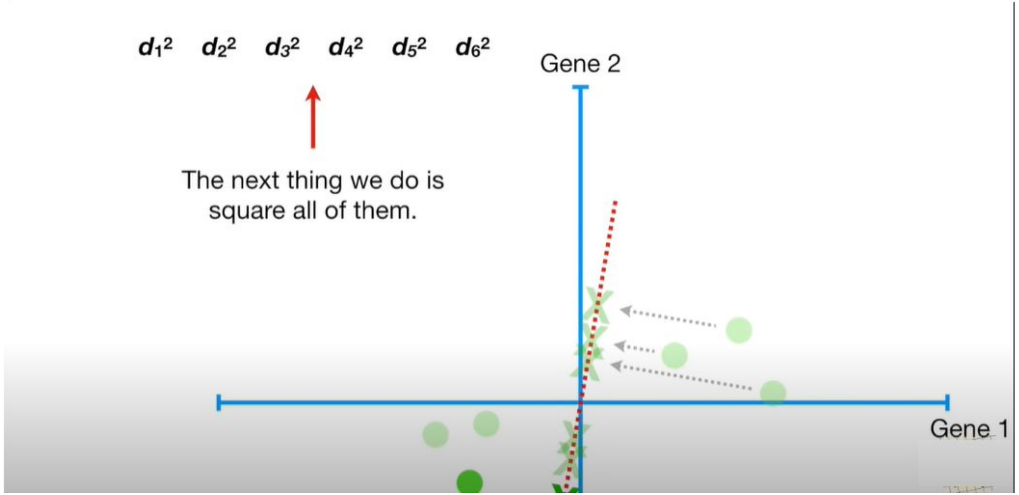


	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

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

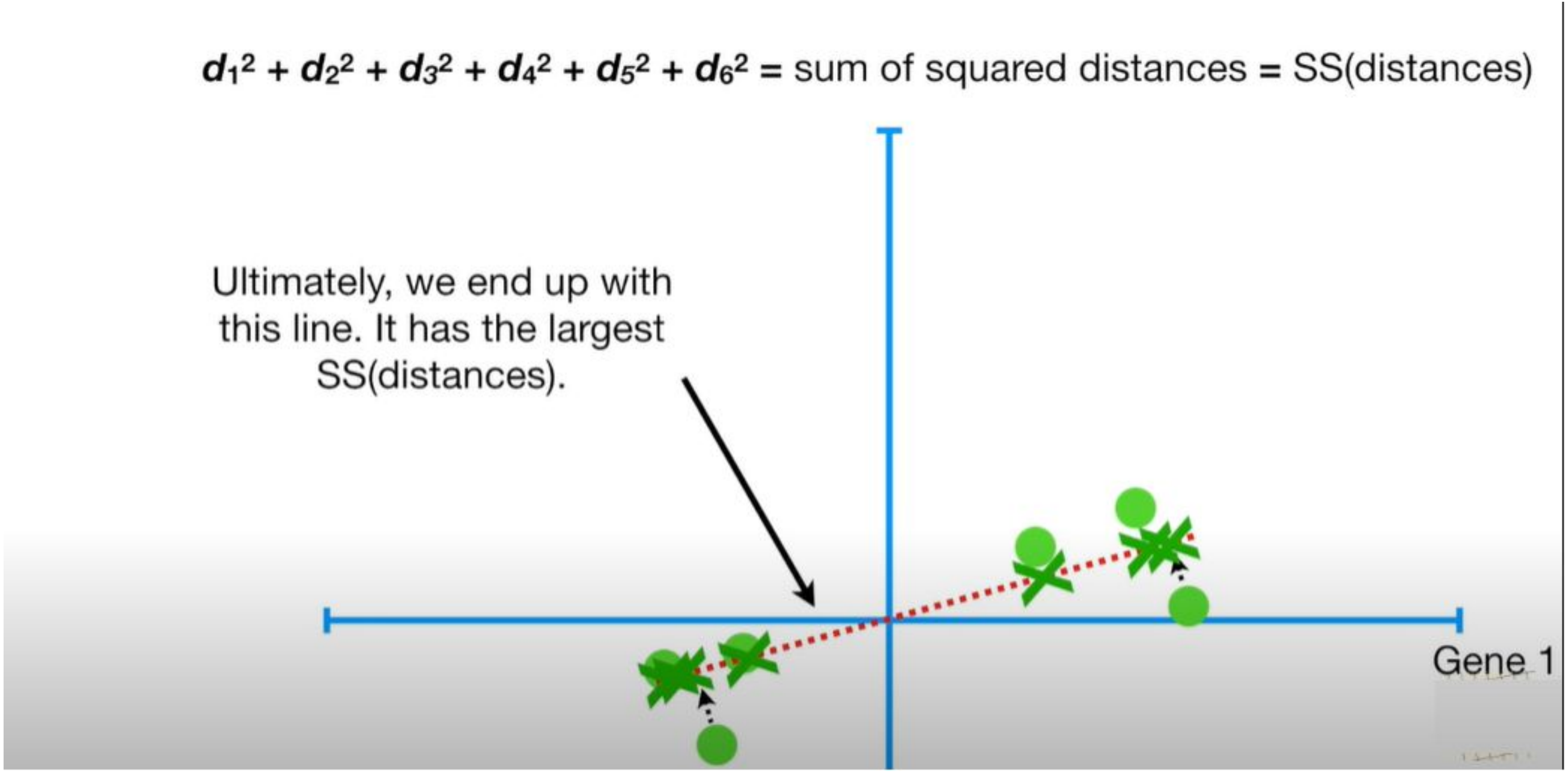






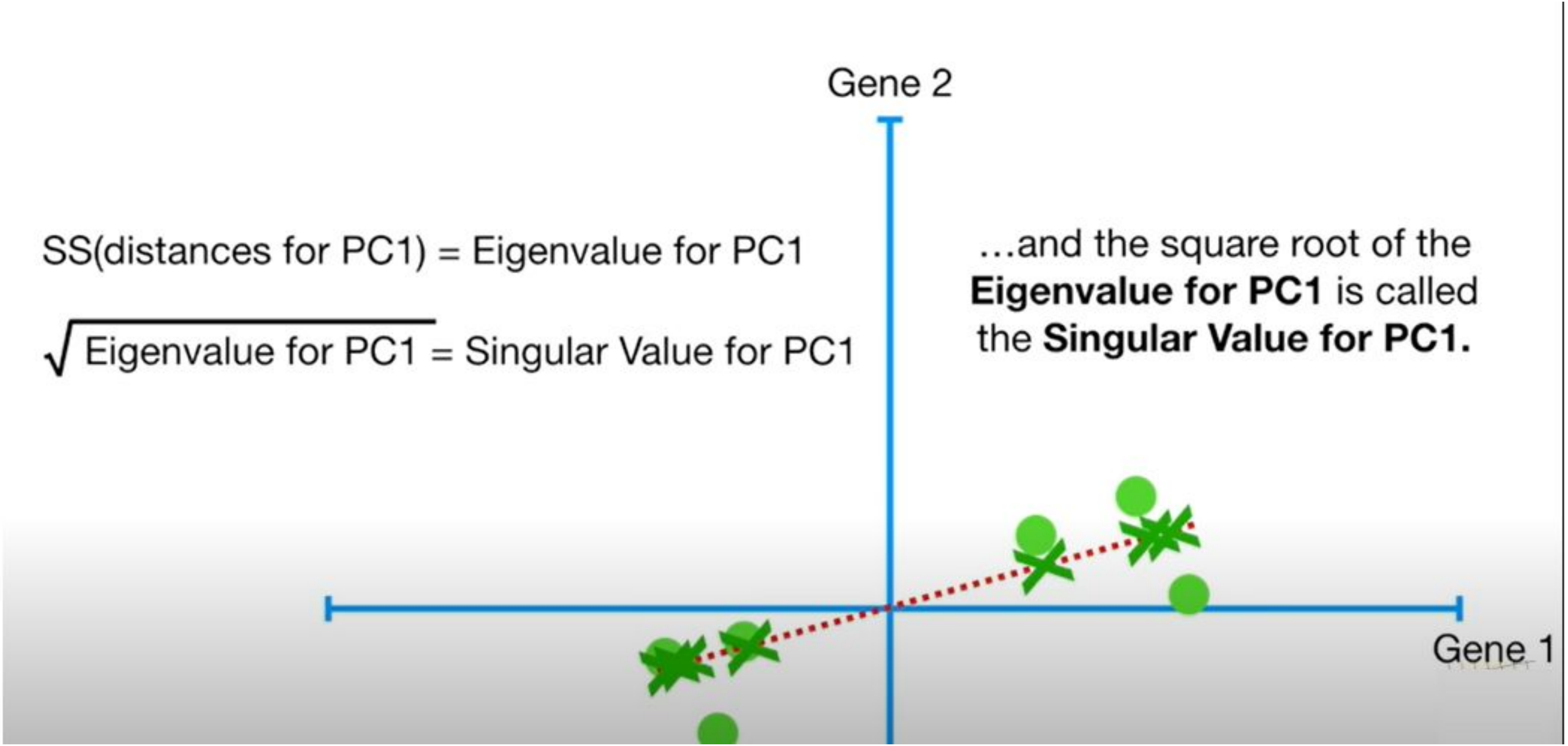
The next thing we do is square all of them.

This line is called PC1



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

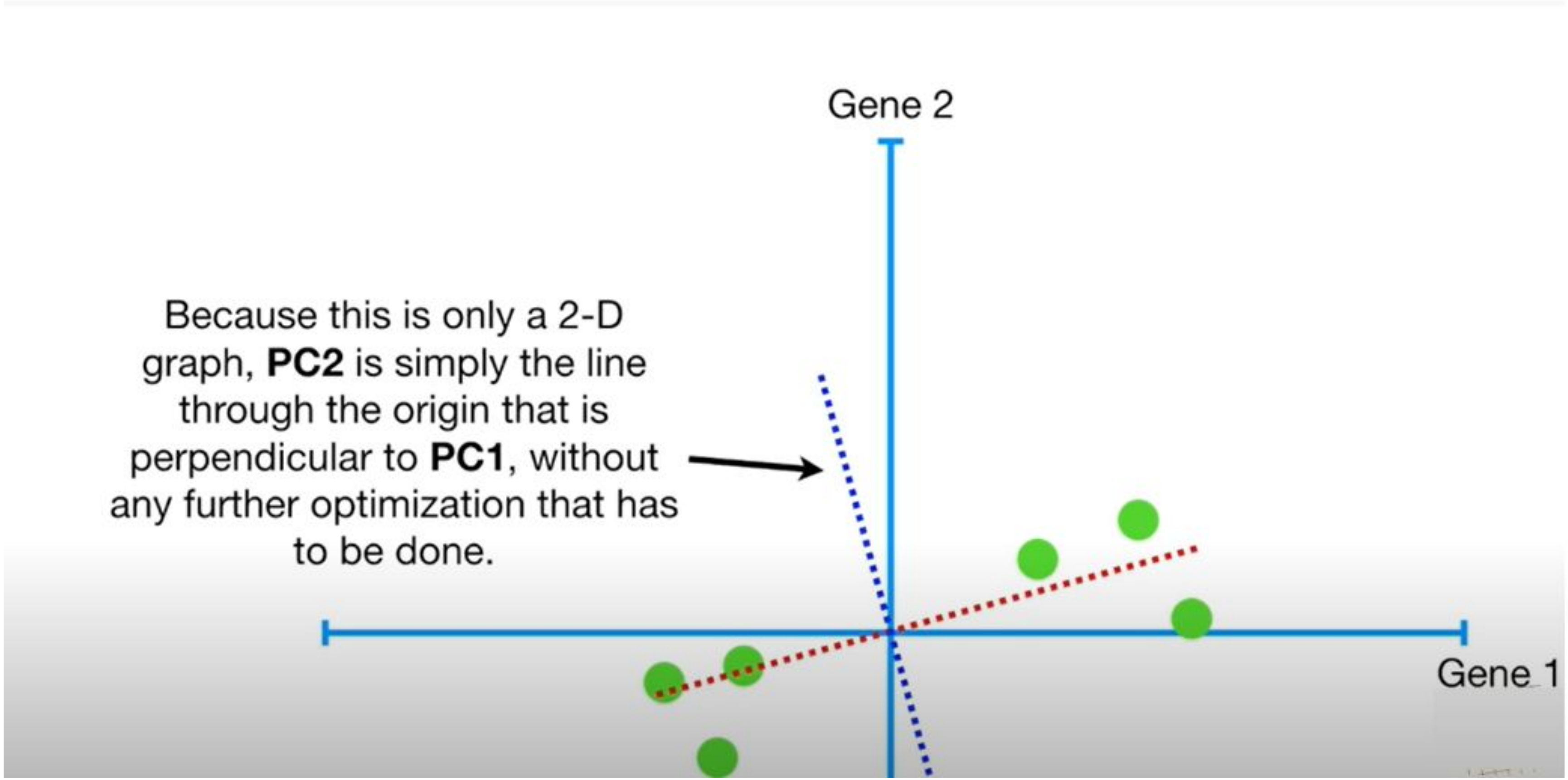
Ultimately, we end up with this line. It has the largest SS(distances).



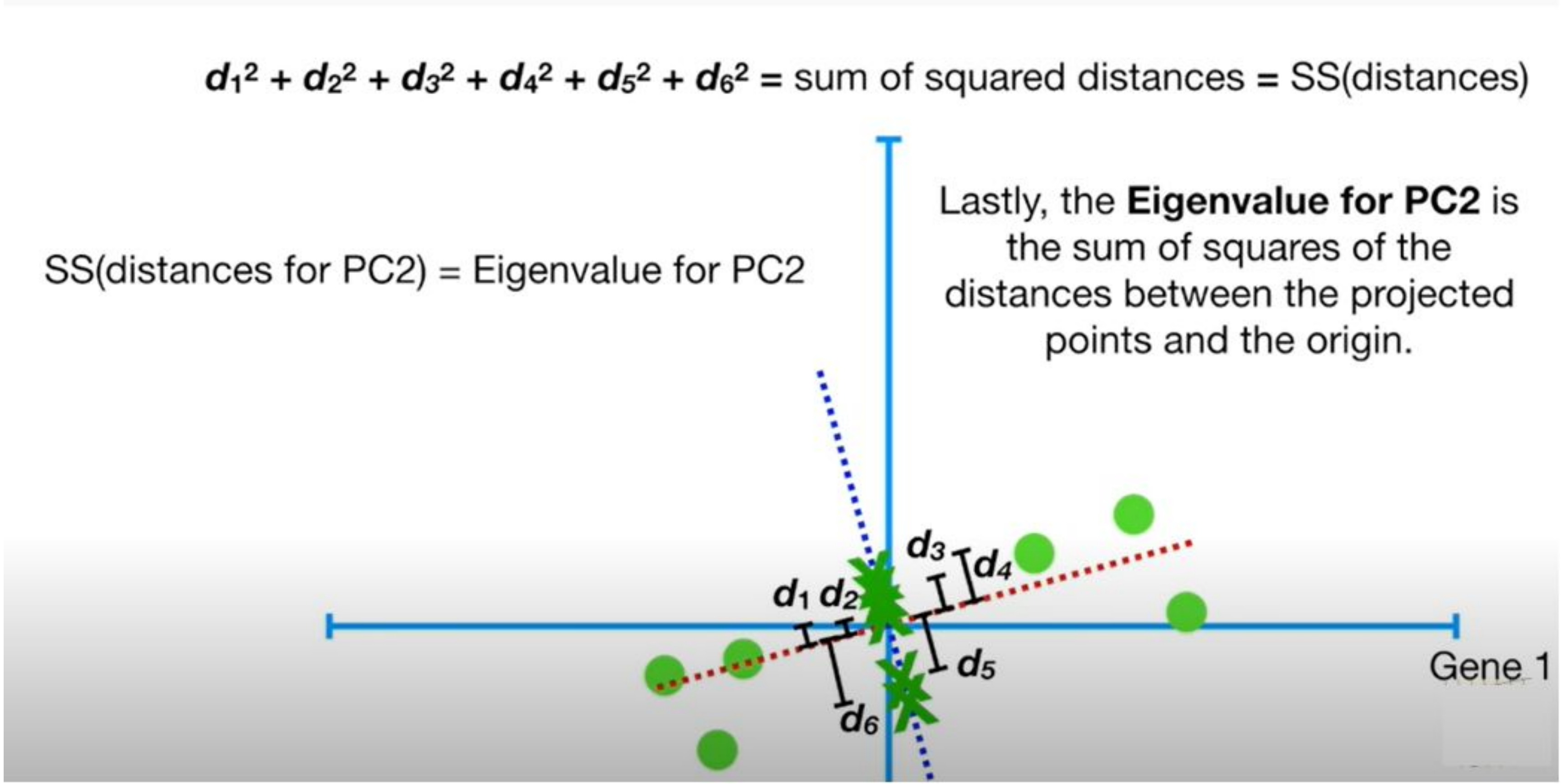
SS(distances for PC1) = Eigenvalue for PC1

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

…and the square root of the **Eigenvalue for PC1** is called the **Singular Value for PC1.**



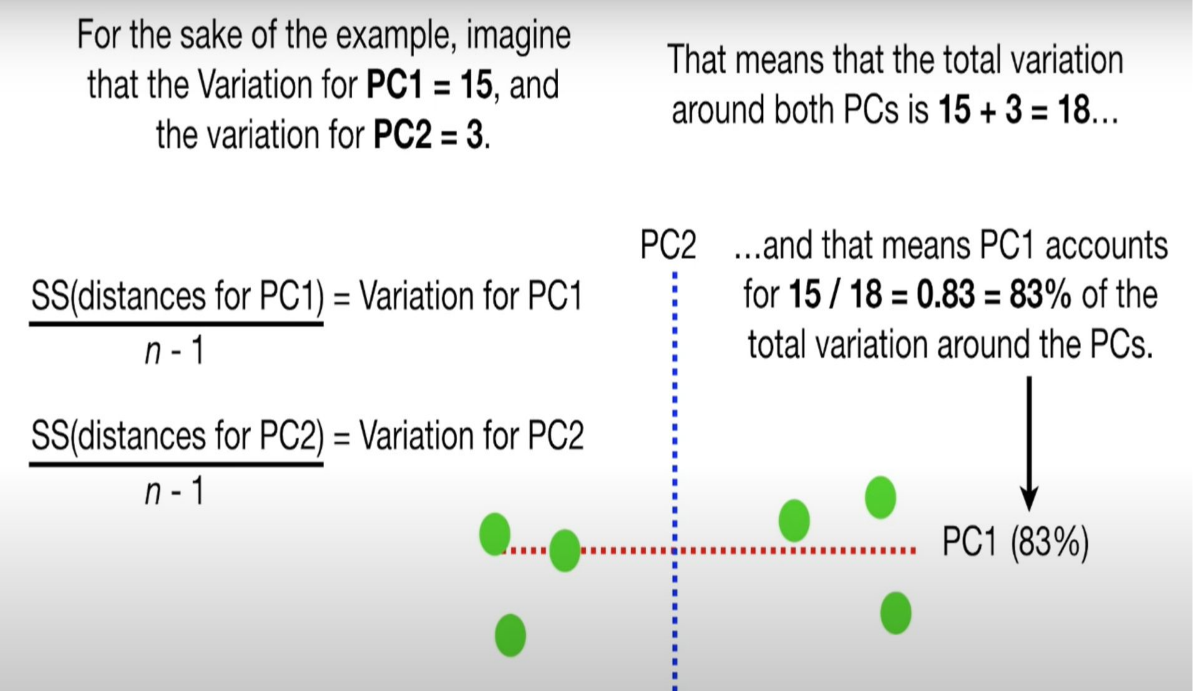
Because this is only a 2-D graph, **PC2** is simply the line through the origin that is perpendicular to **PC1**, without any further optimization that has to be done.



$\boldsymbol{d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2 =}$ sum of squared distances **=** SS(distances)

Lastly, the **Eigenvalue for PC2** is the sum of squares of the distances between the projected points and the origin.

SS(distances for PC2) = Eigenvalue for PC2



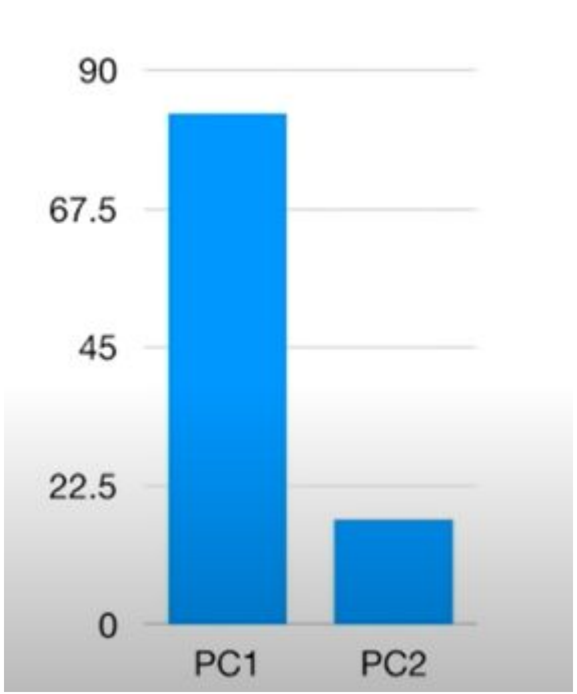
For the sake of the example, imagine that the Variation for **PC1 = 15**, and the variation for **PC2 = 3**.

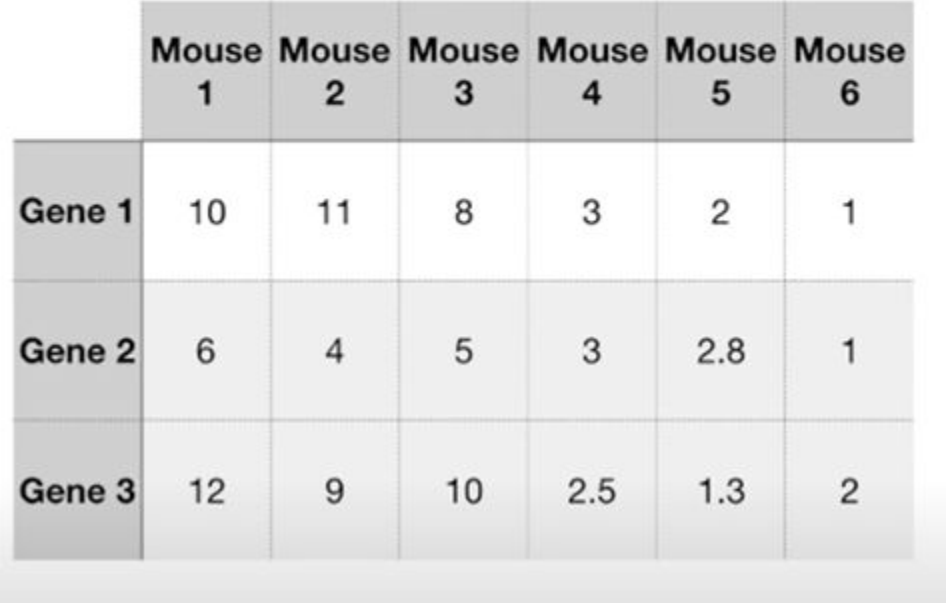
That means that the total variation around both PCs is **15 + 3 = 18**…

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

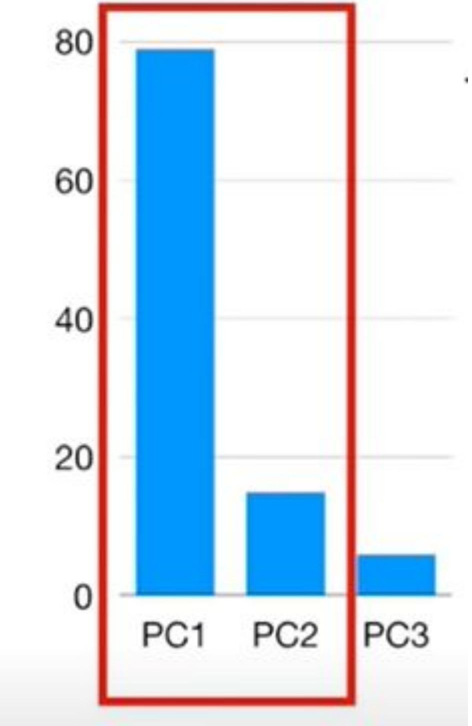
$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Variation for PC2}$$

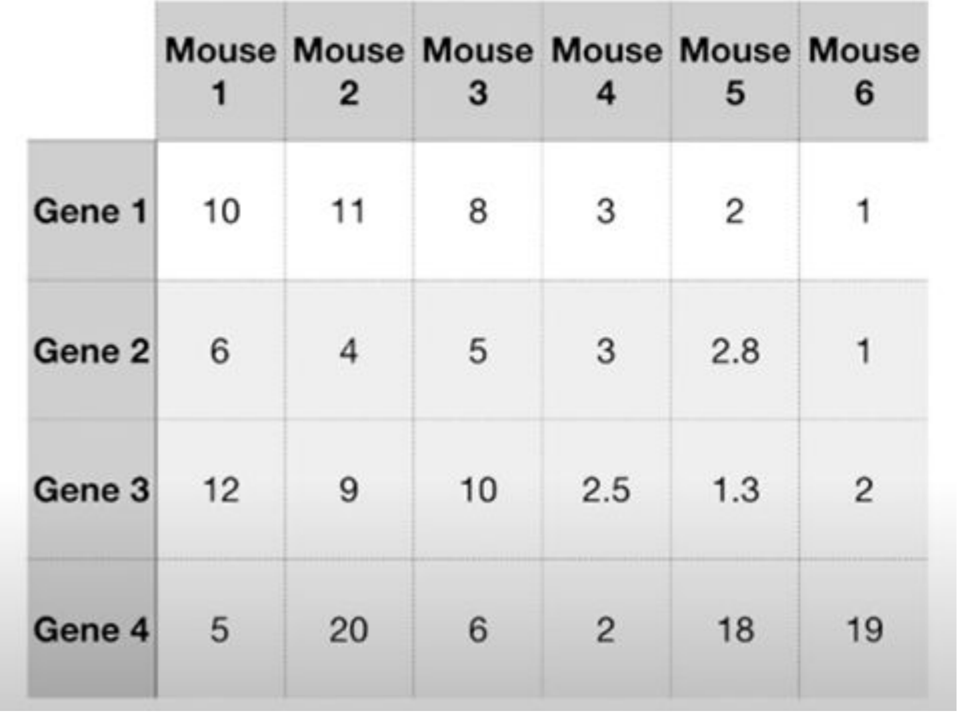
…and that means PC1 accounts for **15 / 18 = 0.83 = 83%** of the total variation around the PCs.



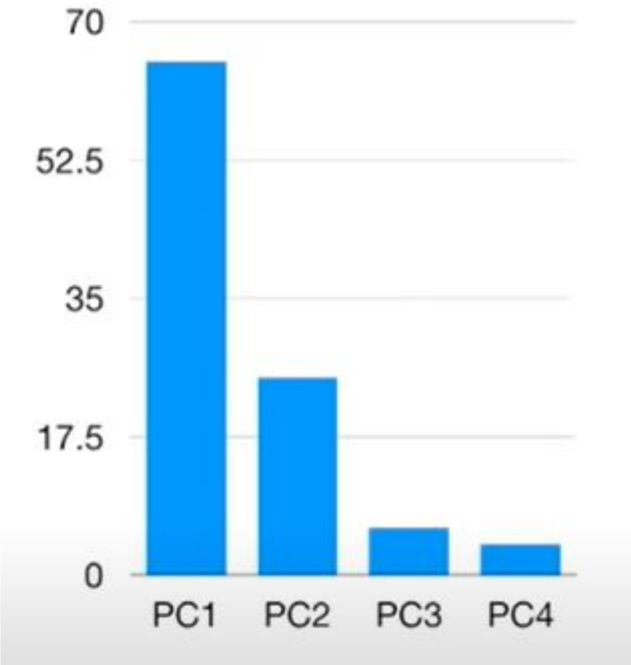


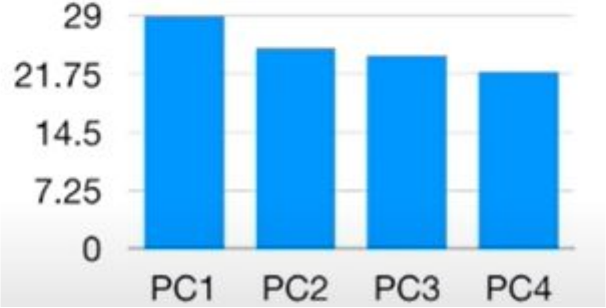
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2





	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	20	6	2	18	19




29
21.75
14.5
7.25
0
PC1
PC2
PC3
PC4