Image Segmentation with k-means

Image Segmentation:

- Another application of clustering is image segmentation
- An image is partitioned into multiple segments (sets of pixels) based on the color of the pixels
- Goal of segmentation is to simplify the representation of an image into something that is more meaningful and easier to analyze.
- It could be used to differentiate objects in the image from background

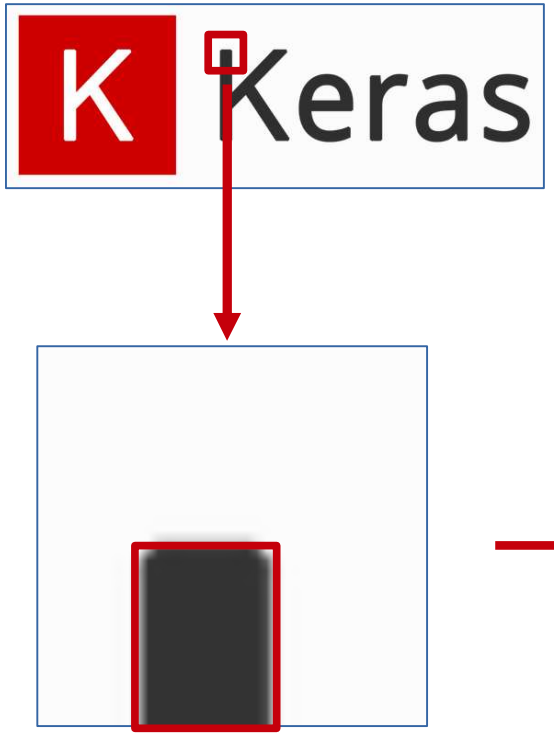
https://mathworks.com

Image Segmentation with k-means

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Image Segmentation:

What is an image?



Matrix with Pixelvalues [0,255]

Image Segmentation with k-means

Image Segmentation:

- For working with images in R we need the package **jpeg**

>install.packages("jpeg")
>library(jpeg)

- To read a file we use the function **readJPEG()**

>myimage = readJPEG("bird.jpg")

Image Segmentation with k-means

Image Segmentation:

Process the image

- Extract dimensions of the image

```
>imgDim = dim(myimage)
```

- Assign color information of the image to a data frame

```
>imgRGB = data.frame(
+       x = rep(1:imgDim[2], each = imgDim[1]),
+       y = rep(imgDim[1]:1, imgDim[2]),
+       R = as.vector(myimage[,,1]),
+       G = as.vector(myimage[,,2]),
+       B = as.vector(myimage[,,3])
+)
#2-dimensional position information has been transformed into a 1-dimensional vector
>dim(imgRGB)
```

Image Segmentation with k-means

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Image Segmentation:

Process the image

- Plot the image using ggplot2

```
>ggplot(data = imgRGB, aes(x = x, y= y)) +
+       geom_point(colour = rgb(imgRGB[c("R","G","B")])) +
+       labs(x = "X-Coord", y = "Y-Coord", title = "Original Image: Colorful Bird")
#It may take a while until the image is shown
```

- Apply k-means clustering on the image using k = 2 and the color information

```
>clusterResult = kmeans(imgRGB[, c("R","G","B")], centers = 2)
```

- Transform the clusters to color information

```
#clusteResult$centers contains the color information for each of the two clusters while
#clusterResult$cluster contains the assignment of the pixel positions to the clusters
>clusterColors = rgb(clusterResult$centers[clusterResult$cluster,])
```

- Plot the image using the clustered colours

```
>ggplot(data = imgRGB, aes(x = x, y = y)) +
+       geom_point(colour = clusterColors) +
+       labs(x = "X-Coord", y = "Y-Coord", title = "k-means Clustering of 2 Colors")
```

Image Segmentation with k-means

Image Segmentation:

Process the image

– Let us do the same for more clusters

```
#k = 3
>clusterResult = kmeans(imgRGB[, c("R","G","B")], centers = 3)
>clusterColors = rgb(clusterResult$centers[clusterResult$cluster,])
>ggplot(data = imgRGB, aes(x = x, y = y)) +
+      geom_point(colour = clusterColors) +
+      labs(x = "X-Coord", y = "Y-Coord", title = "k-means Clustering of 3 Colors")
#k = 4
>clusterResult = kmeans(imgRGB[, c("R","G","B")], centers = 4)
>clusterColors = rgb(clusterResult$centers[clusterResult$cluster,])
>ggplot(data = imgRGB, aes(x = x, y = y)) +
+      geom_point(colour = clusterColors) +
+      labs(x = "X-Coord", y = "Y-Coord", title = "k-means Clustering of 4 Colors")
```

Clustering Exercises (Image Segmentation)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exercise:

Import the image called **“cows_on_meadow.jpg”** into the variable **myimage**

Process the image

- Extract dimensions of the image
- Assign color information of the image to a data frame
- Plot the original image using ggplot2
- Do the following steps for k = 1, 2, 3, 4, 5:
  - Apply k-means clustering on the image using k clusters and the color information
  - Transform the clusters to color information
  - Create the ggplot with the clustered colors and save the image

Clustering Exercises (Image Segmentation)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exercise:

Import the image called **“cows_on_meadow.jpg”** into the variable **myimage**

```
>myimage = readJPEG("cows_on_meadow.jpg")
```

Process the image

- Extract dimensions of the image

```
>imgDim = dim(myimage)
```

- Assign color information of the image to a data frame

```
>imgRGB = data.frame(
+       x = rep(1:imgDim[2], each = imgDim[1]),
+       y = rep(imgDim[1]:1, imgDim[2]),
+       R = as.vector(myimage[,,1]),
+       G = as.vector(myimage[,,2]),
+       B = as.vector(myimage[,,3]),
)
```

Clustering Exercises (Image Segmentation)

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Exercise:

Process the image

- Plot the original image using ggplot2

```
>ggplot(data = imgRGB, aes(x = x, y = y)) +
+       geom_point(colour = rgb(imgRGB[c("R","G","B")])) +
+       labs(x = "X-Coord", y = "Y-Coord", title = "Original Image: Cows on Meadow")
```

Clustering Exercises (Image Segmentation)

Exercise:

Process the image

- Do the following steps for k = 1, 2, 3, 4, 5:
  - Apply k-means clustering on the image using k clusters and the color information
  - Transform the clusters to color information
  - Plot the image using the clustered colors

```
>plots = c()
>for(k in 1:5){
+           clusterResult = kmeans(imgRGB[,c("R","G","B")], centers = 2)
+           clusterColors = rgb(clusterResult$centers[clusterResult$cluster,])
+           plots[[k]] = ggplot(data = imgRGB, aes(x = x, y = y)) +
+                       geom_point(colour = clusterColors) +
+                       labs(x = "X-Coord", y = "Y-Coord", title = "k-means Clustering of 2 Colors")
+           #[[]] have to be used to save ggplots in a vector
+}
>plots[1]
>plots[2]
>plots[3] # and so on
```

Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

- **PCA:**
  - One of the most useful data analysis methods to identify patterns in highly complex datasets
  - Which variables in your data are the most important
  - How accurate your new understanding of the data actually is

- **PCA-Analysis: step by step**
  - We will go one step at a time through PCA, and the method used to solve it, Singular Value Decomposition
  - Aim is to learn:
    - What PCA does
    - How PCA does it
    - How to use PCA to get deeper insight into your data

Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

… in 6 different animals

Transcription of ***gene-1***
and ***gene-2*** is measured

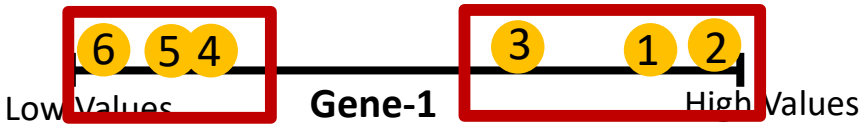
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

If we only measure one gene,
we can plot the data on a number line



Animals 4,5, and 6 have relatively low values

Animals 1, 2, and 3 have relatively high values

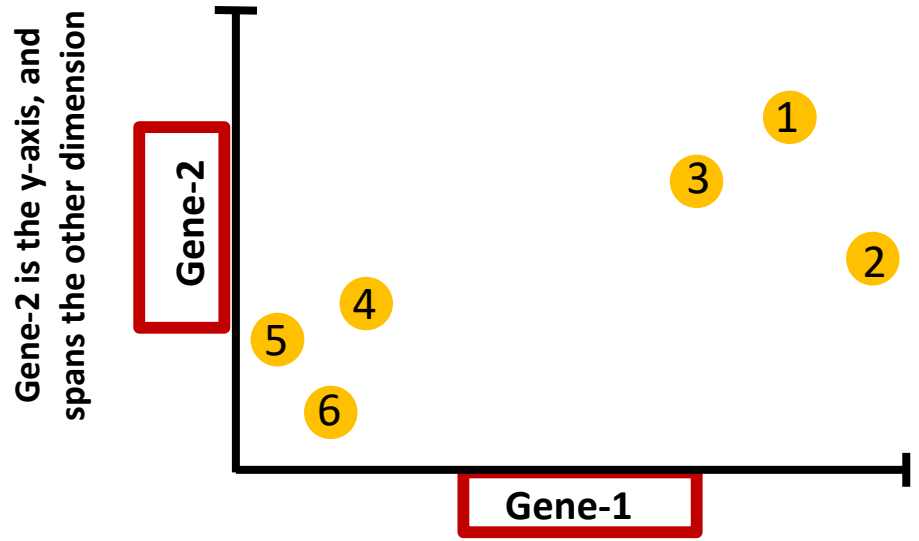
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

If we measured two genes, we can plot the data on a 2-D x/y graph



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

https://www.datascienceplus.com

Principal Component Analysis (PCA)

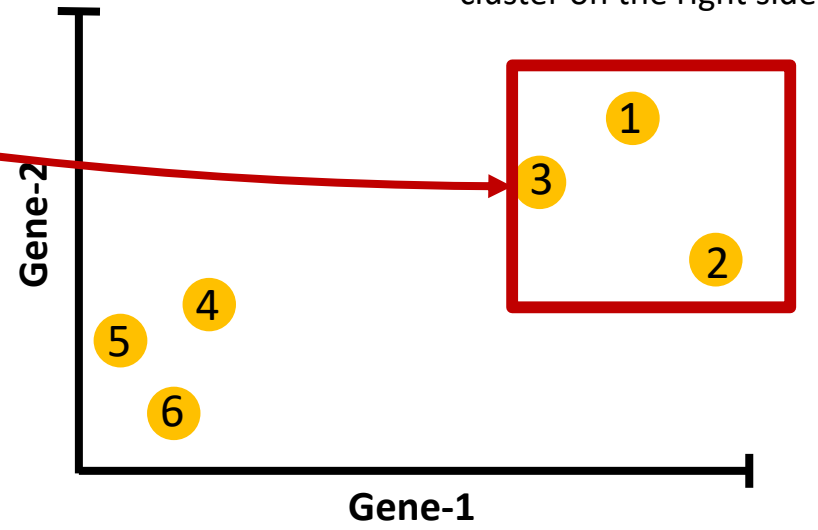
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

Animals 1, 2, and 3 have relatively high values

Animals 1, 2, and 3 cluster on the right side



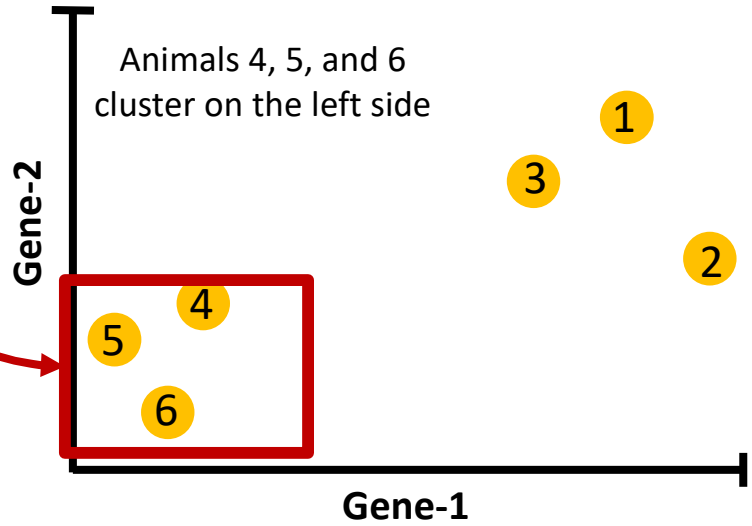
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

Animals 4,5, and 6 have relatively low values

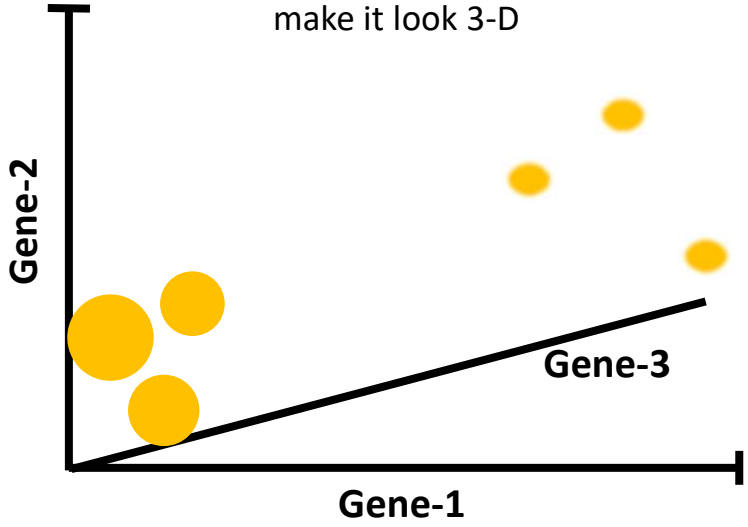


Principal Component Analysis (PCA)

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4

If we consider 3-genes, we would add another axis to the graph and make it look 3-D

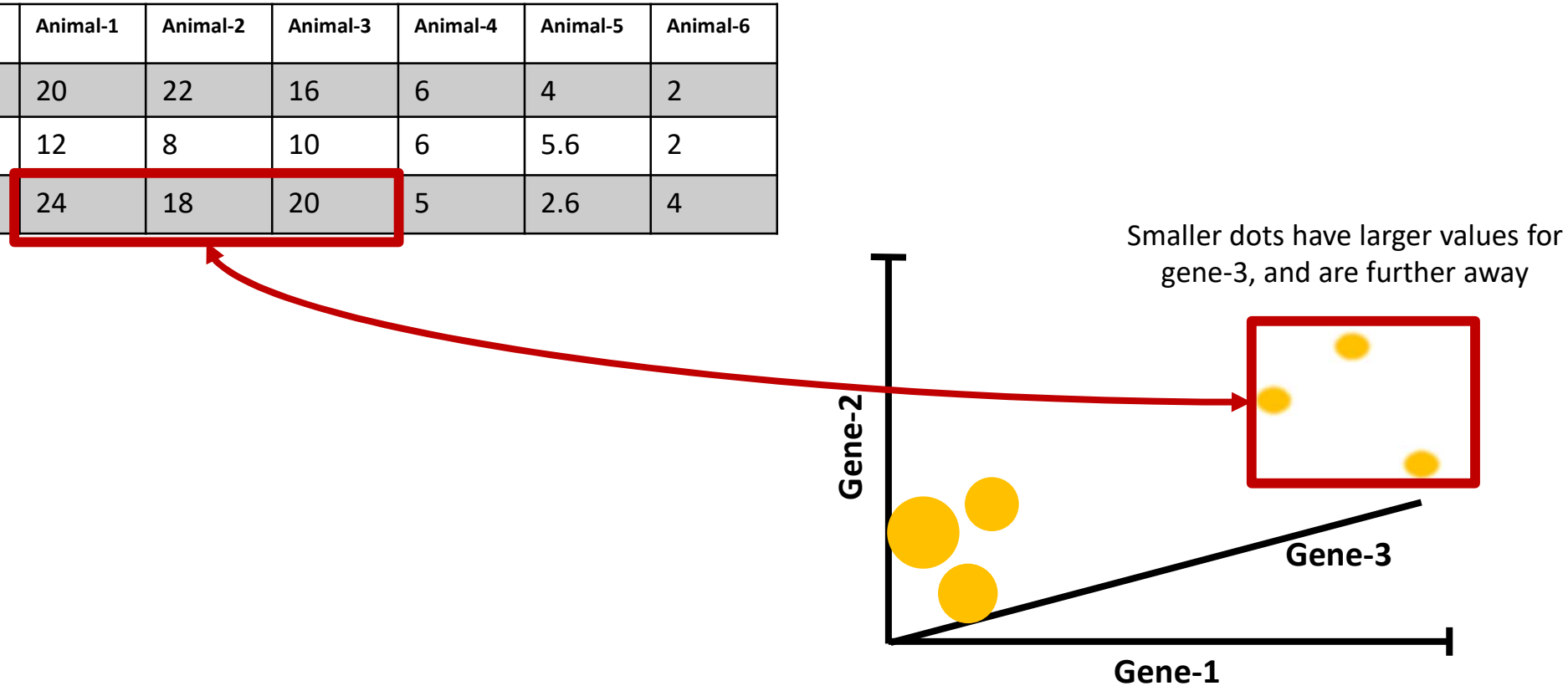


Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4

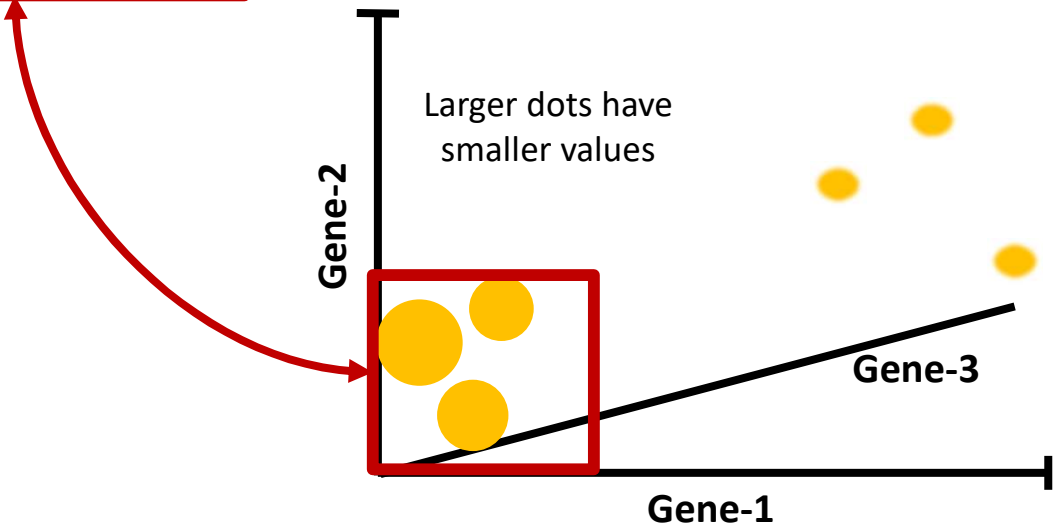


Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4



Principal Component Analysis (PCA)

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

If we want to consider 4 genes,
we **cannot** plot the data.
4 genes require 4-dimensions

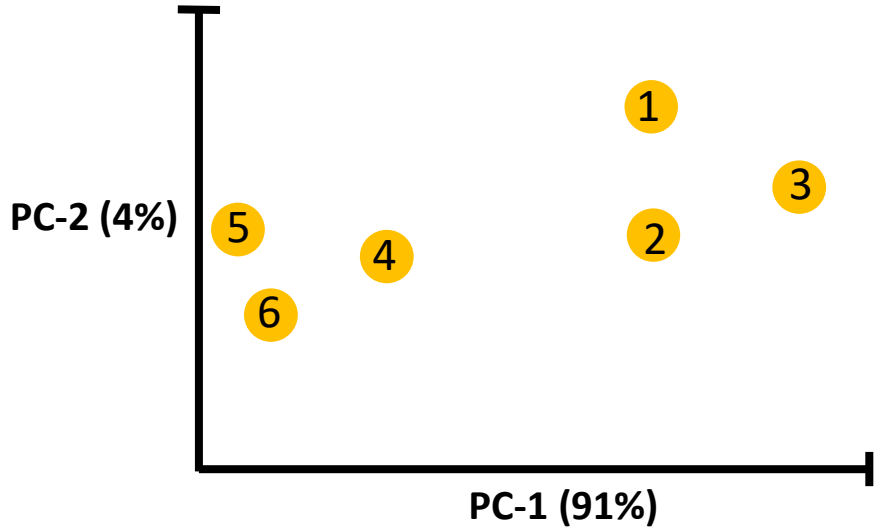
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

PCA can take four or more gene measurements and make a 2-D PCA plot…



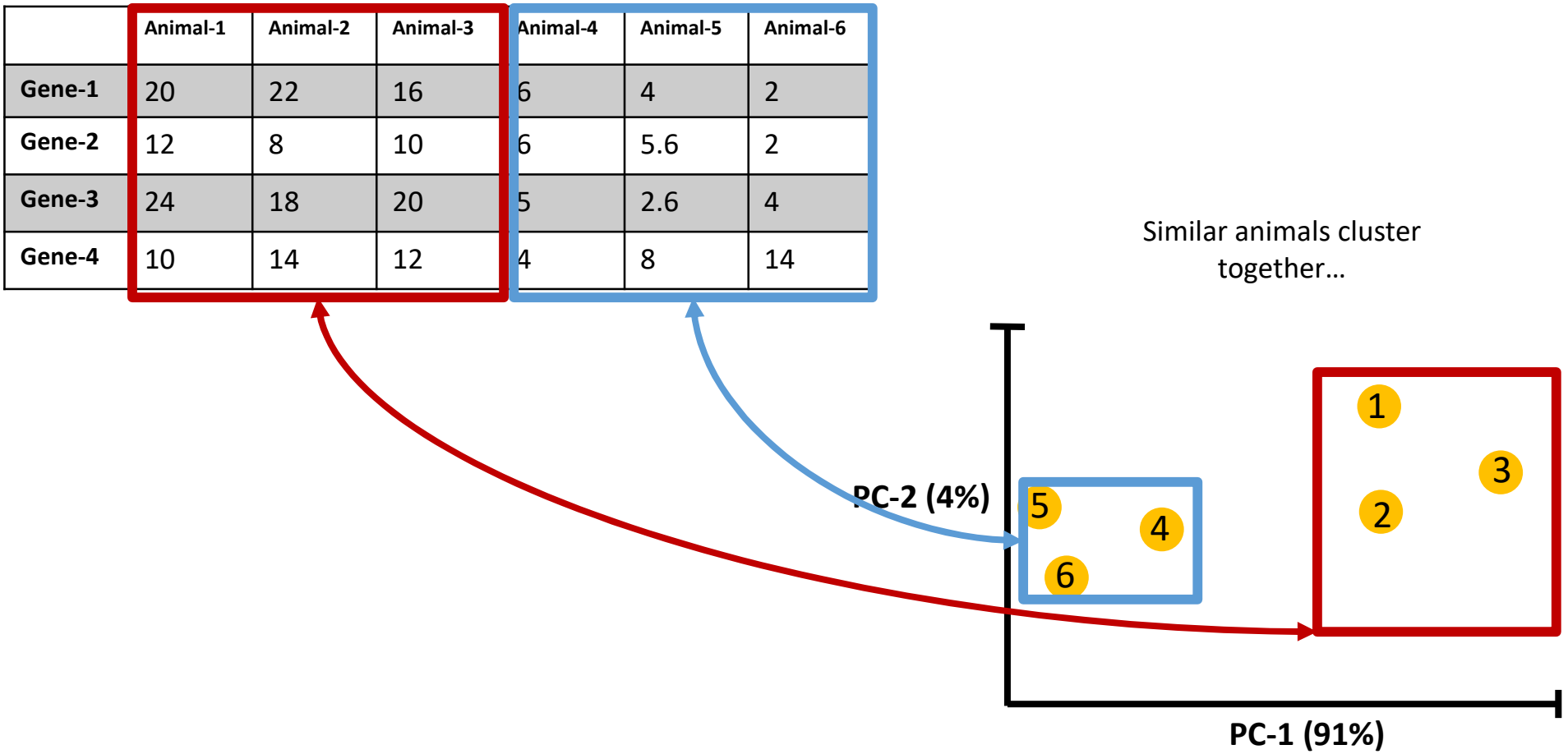
https://www.datascienceplus.com

Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14



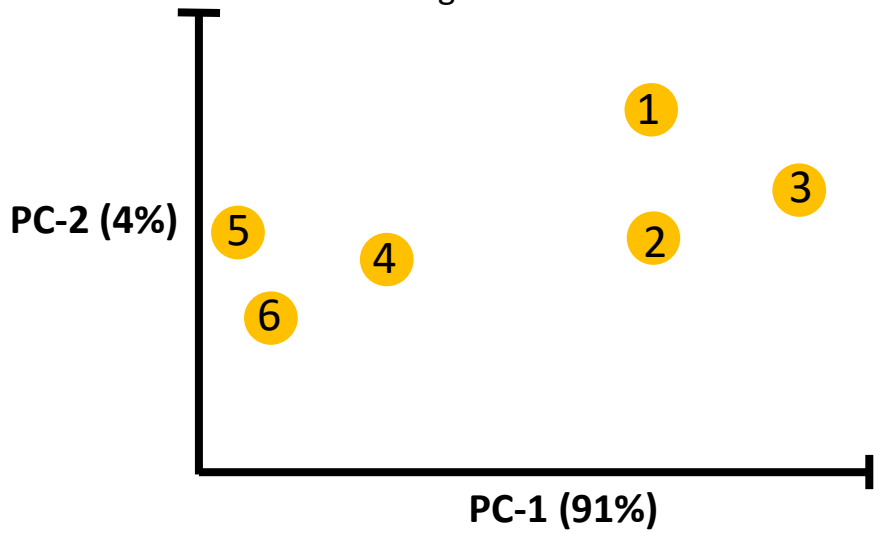
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

PCA can tell us which variable(or gene) is the most valuable for clustering the data



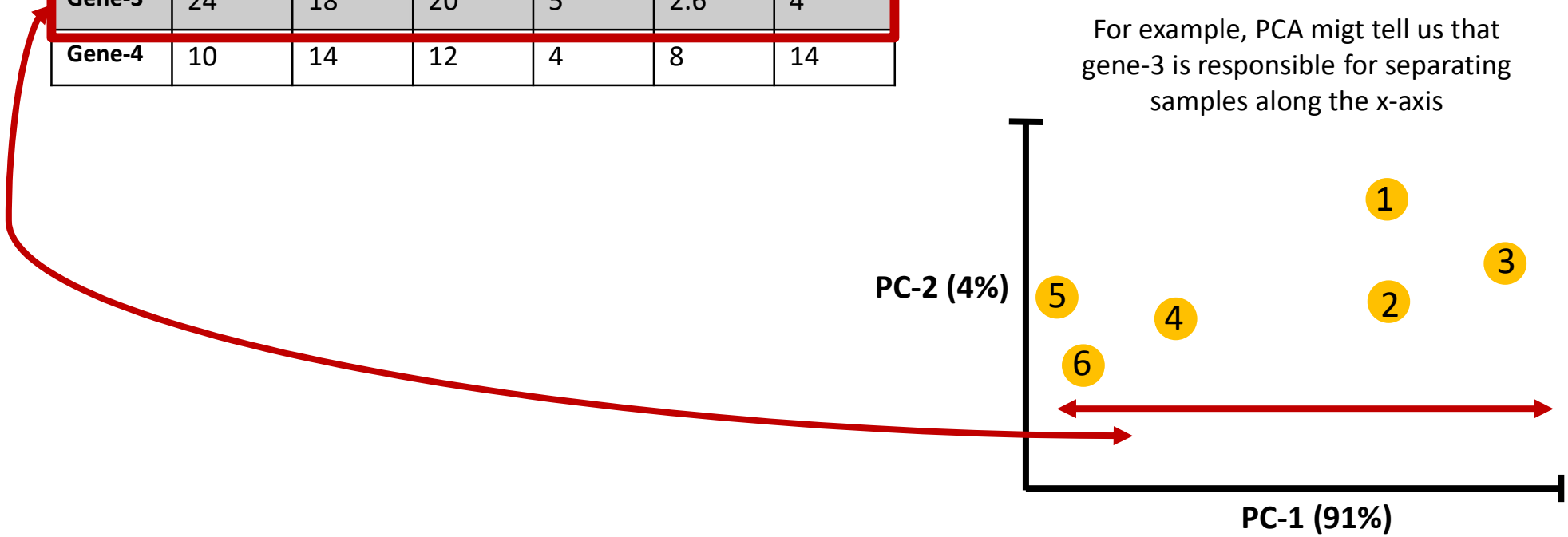
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

For example, PCA migt tell us that gene-3 is responsible for separating samples along the x-axis



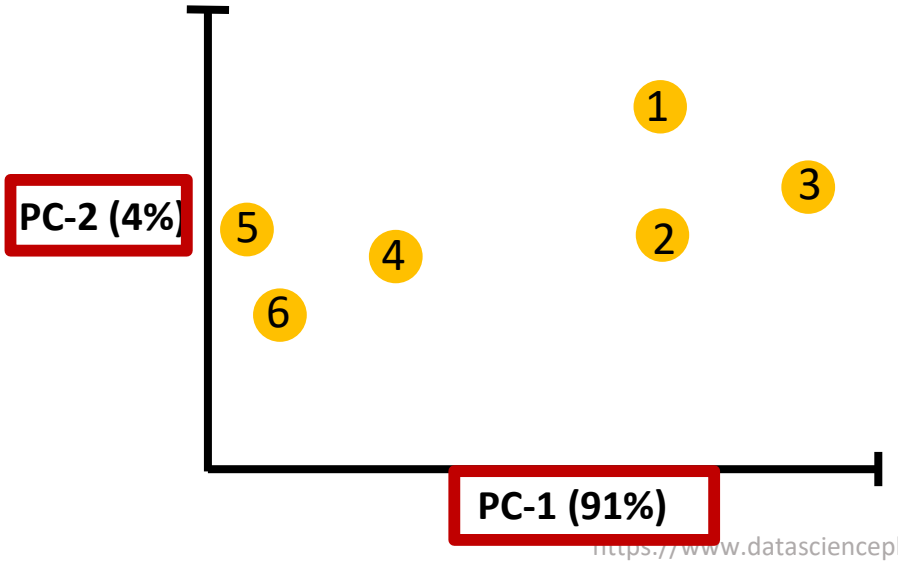
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

PCA can also tell us how accurate the 2-D graph is



Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

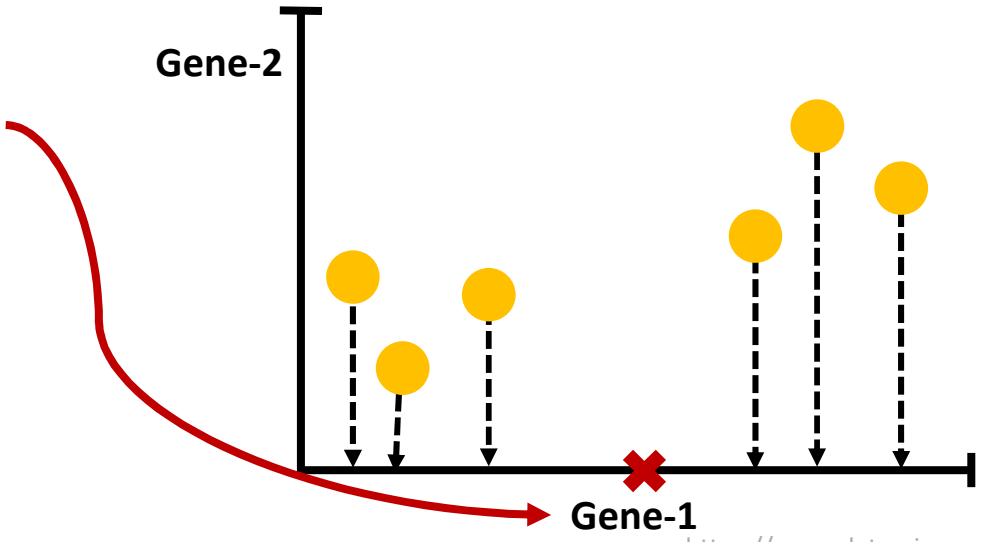
To understand what PCA does and how it works, let‘s go back to the dataset that only has 2 genes

Principal Component Analysis (PCA)

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

- We will start by plotting the data
- Then we will calculate the average measurement for Gene 1



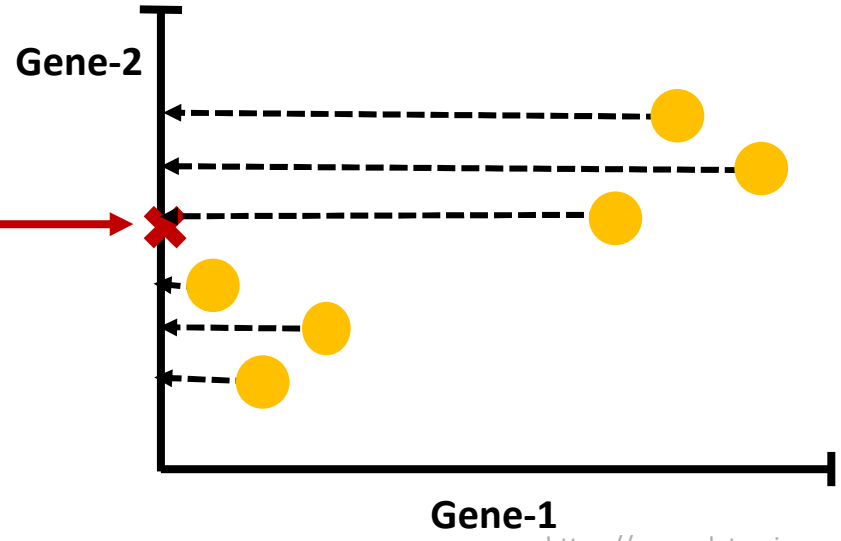
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

- We will start by plotting the data
- Then we will calculate the average measurement for Gene 1
- We will also calculate the average measurement for Gene 2



https://www.datascienceplus.com

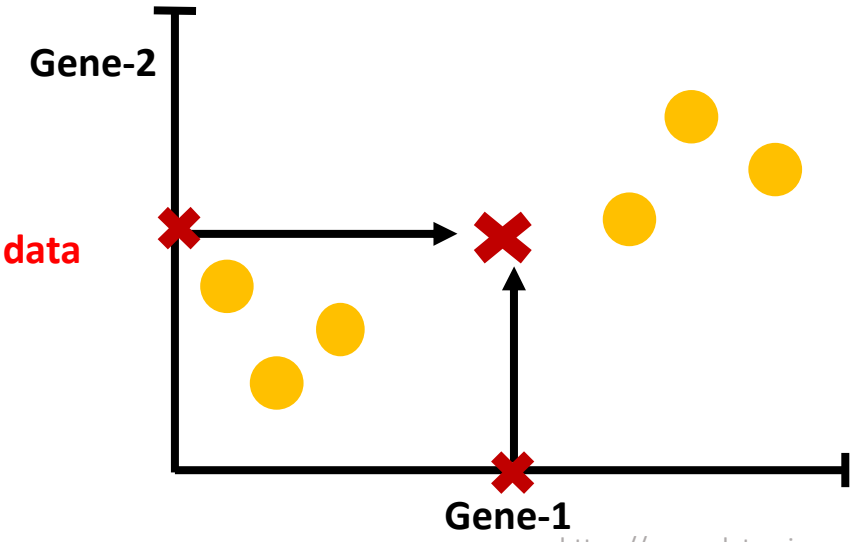
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2

- We will start by plotting the data
- Then we will calculate the average measurement for Gene 1
- We will also calculate the average measurement for Gene 2
- **With the average values, we can calculate the center of the data**



https://www.datascienceplus.com

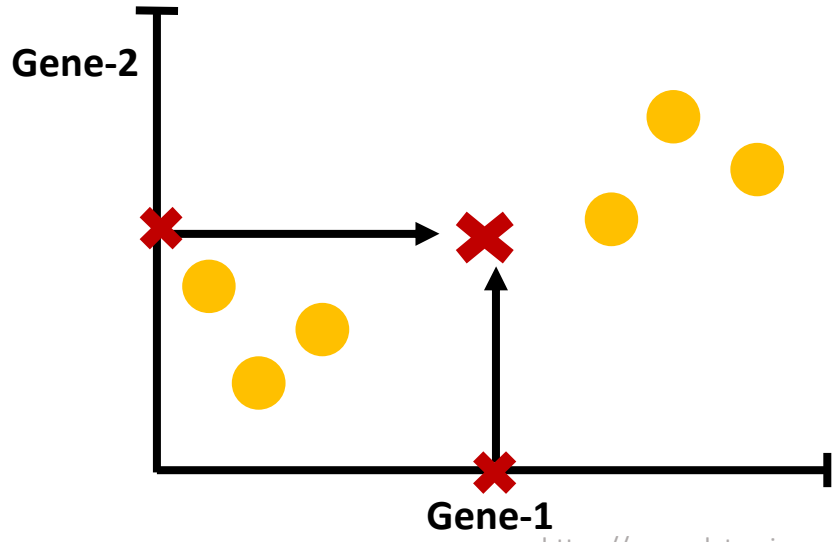
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	[illegible]	6	4	2
Gene-2	12	8	10	6	5.6	2

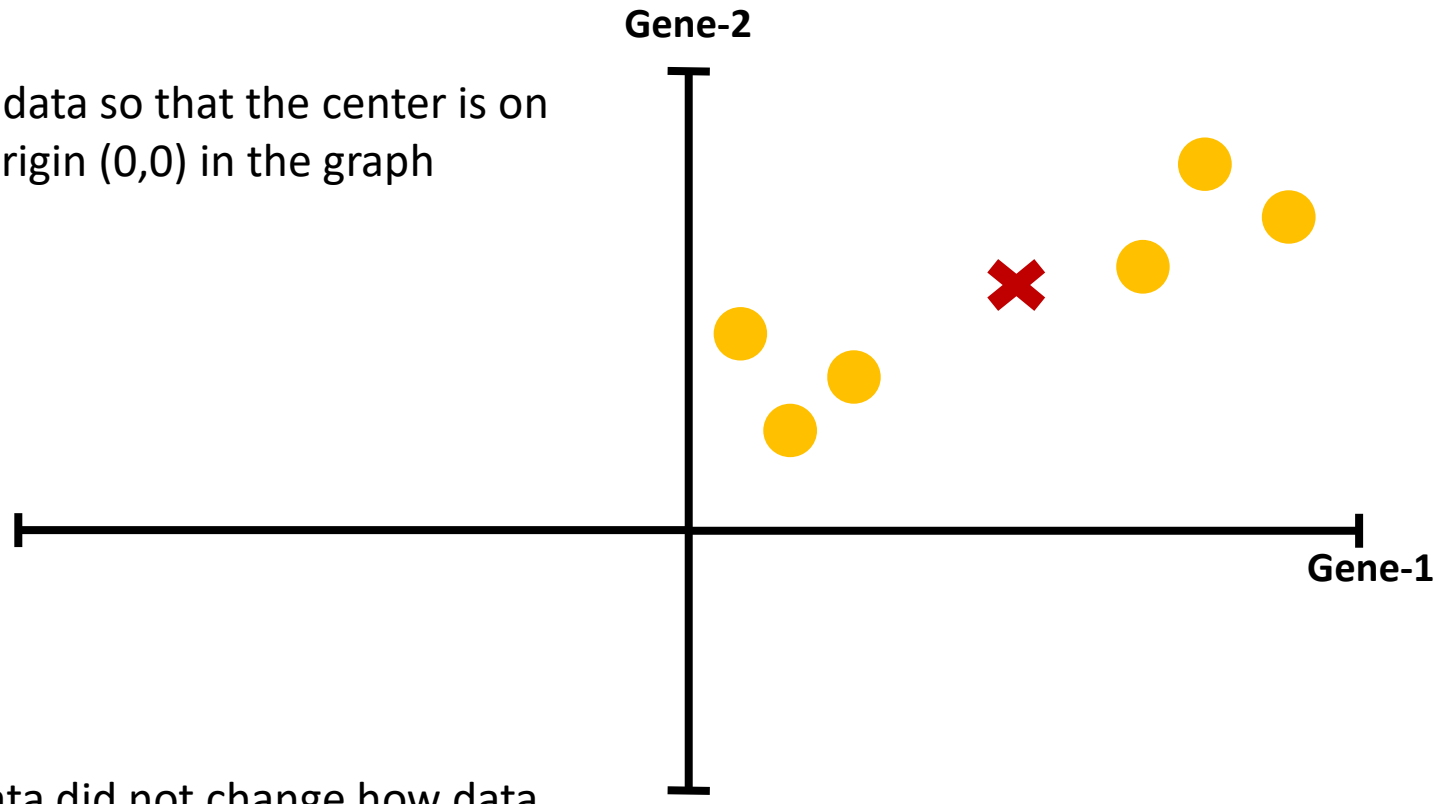
From this point on we will focus on what happens in the graph; we no longer need the original data



https://www.datascienceplus.com

Principal Component Analysis (PCA)

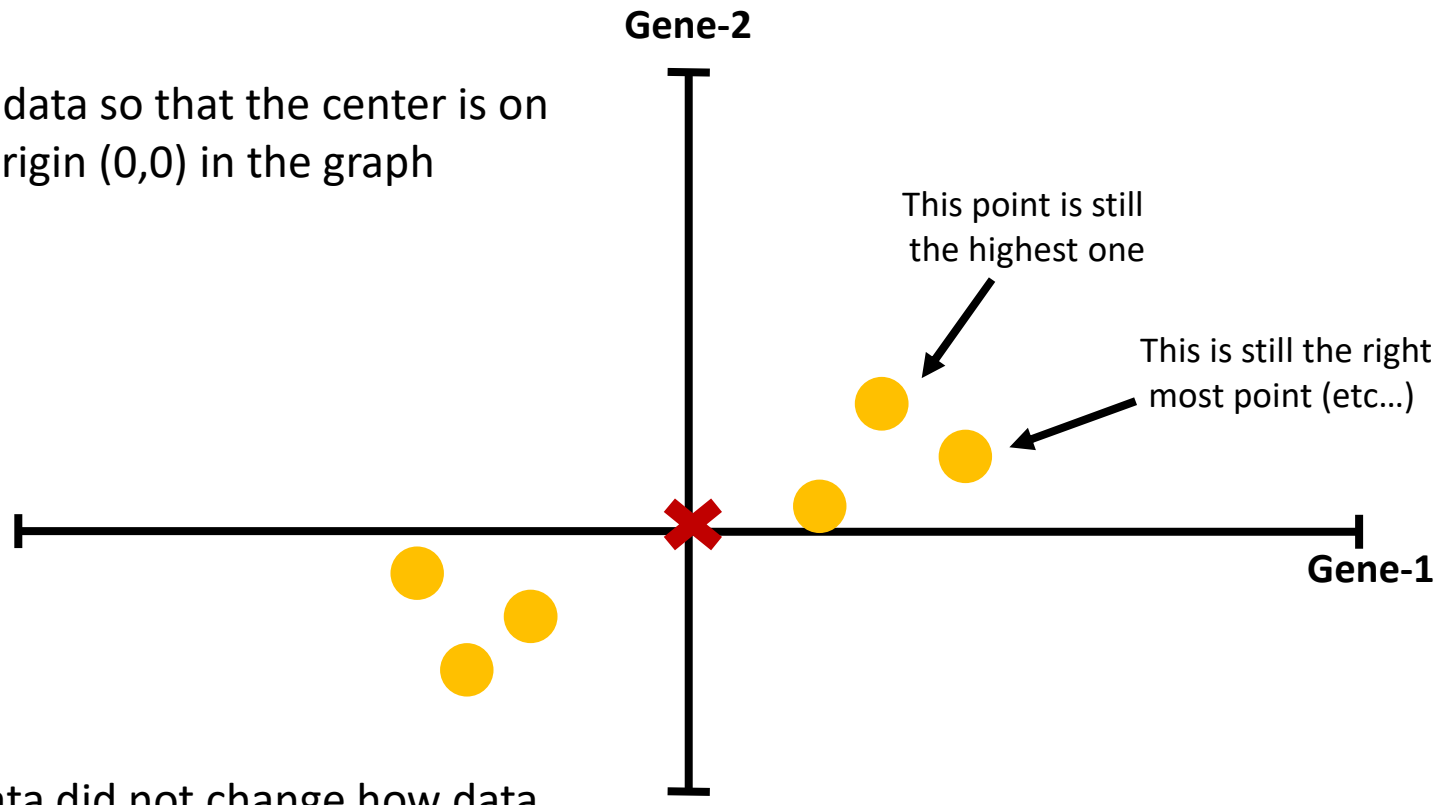
Now we will shift the data so that the center is on the top of the origin (0,0) in the graph



Note: shifting the data did not change how data points are positioned relative to each other

Principal Component Analysis (PCA)

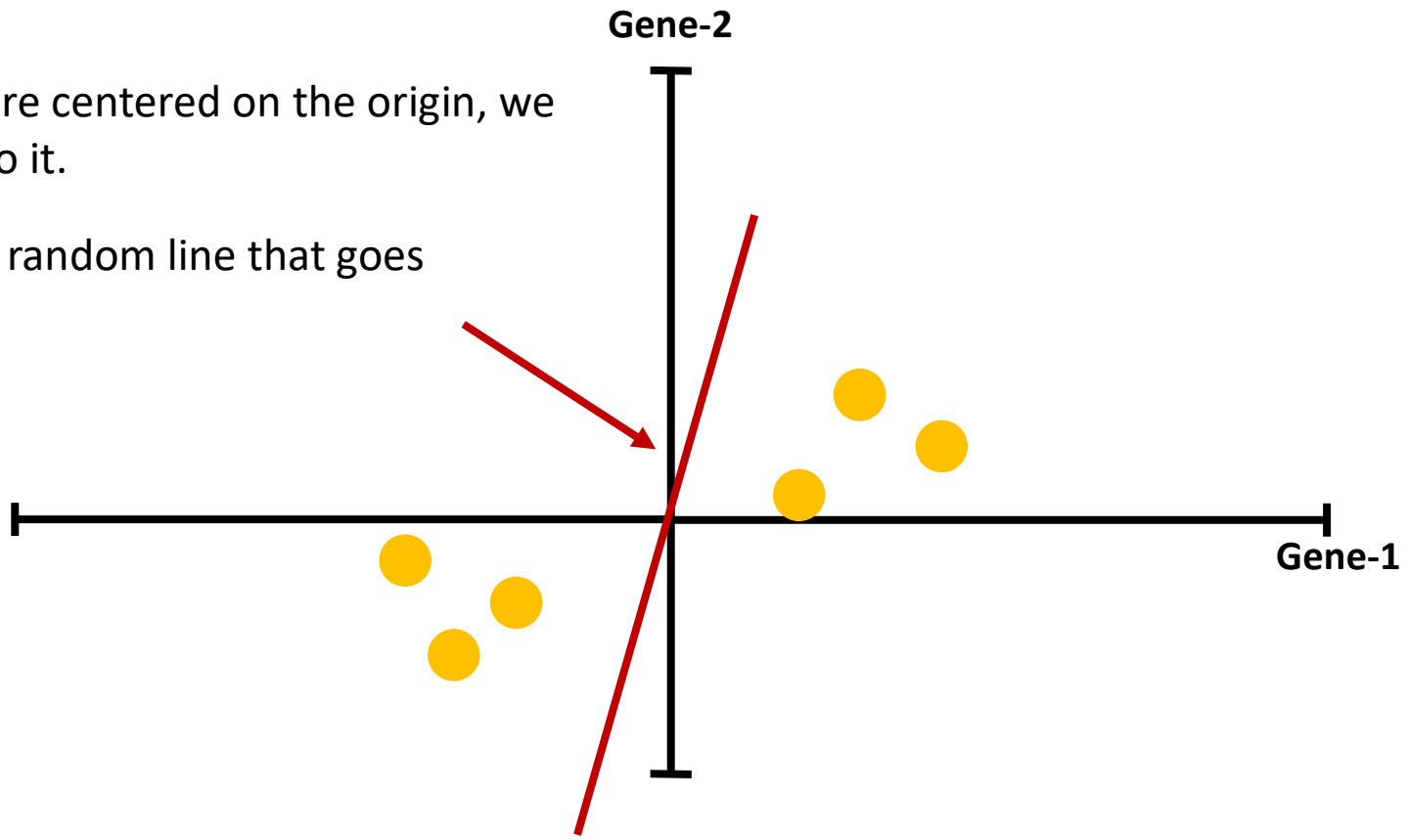
Now we will shift the data so that the center is on the top of the origin (0,0) in the graph



Note: shifting the data did not change how data points are positioned relative to each other

Principal Component Analysis (PCA)

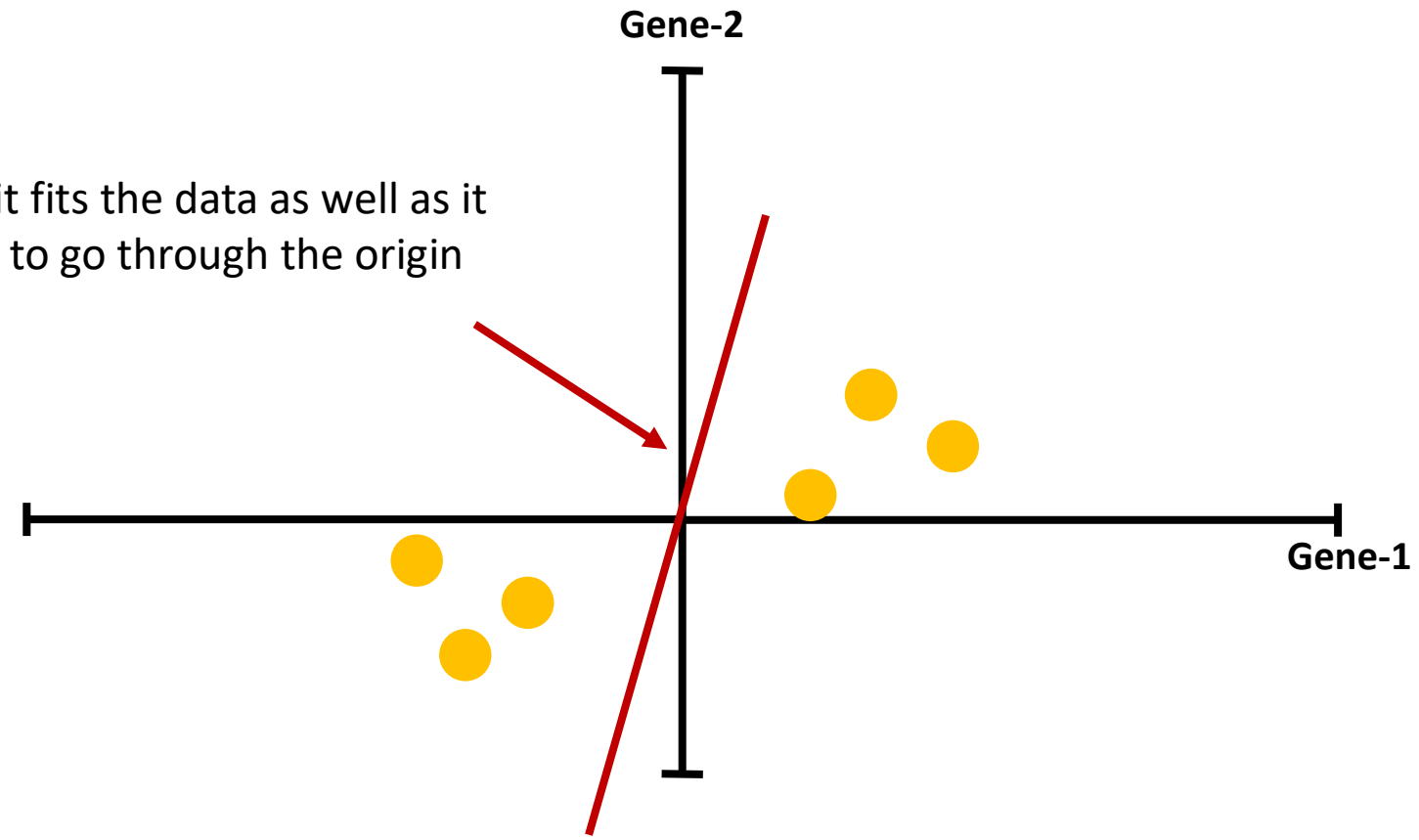
- Now that the data are centered on the origin, we can try to fit a line to it.
- For this aim, draw a random line that goes through the origin



https://www.datascienceplus.com

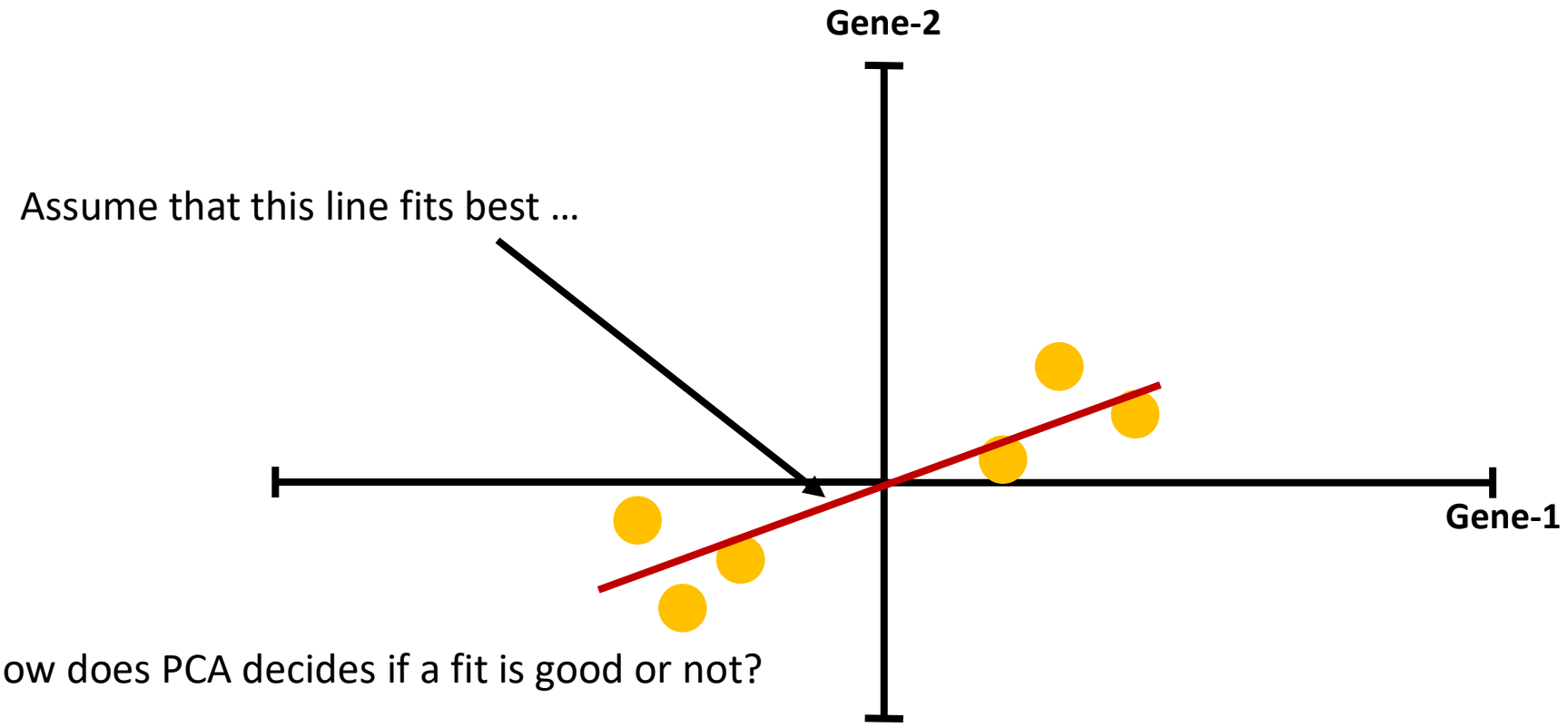
Principal Component Analysis (PCA)

- Rotate the line until it fits the data as well as it can, given that it has to go through the origin



https://www.datascienceplus.com

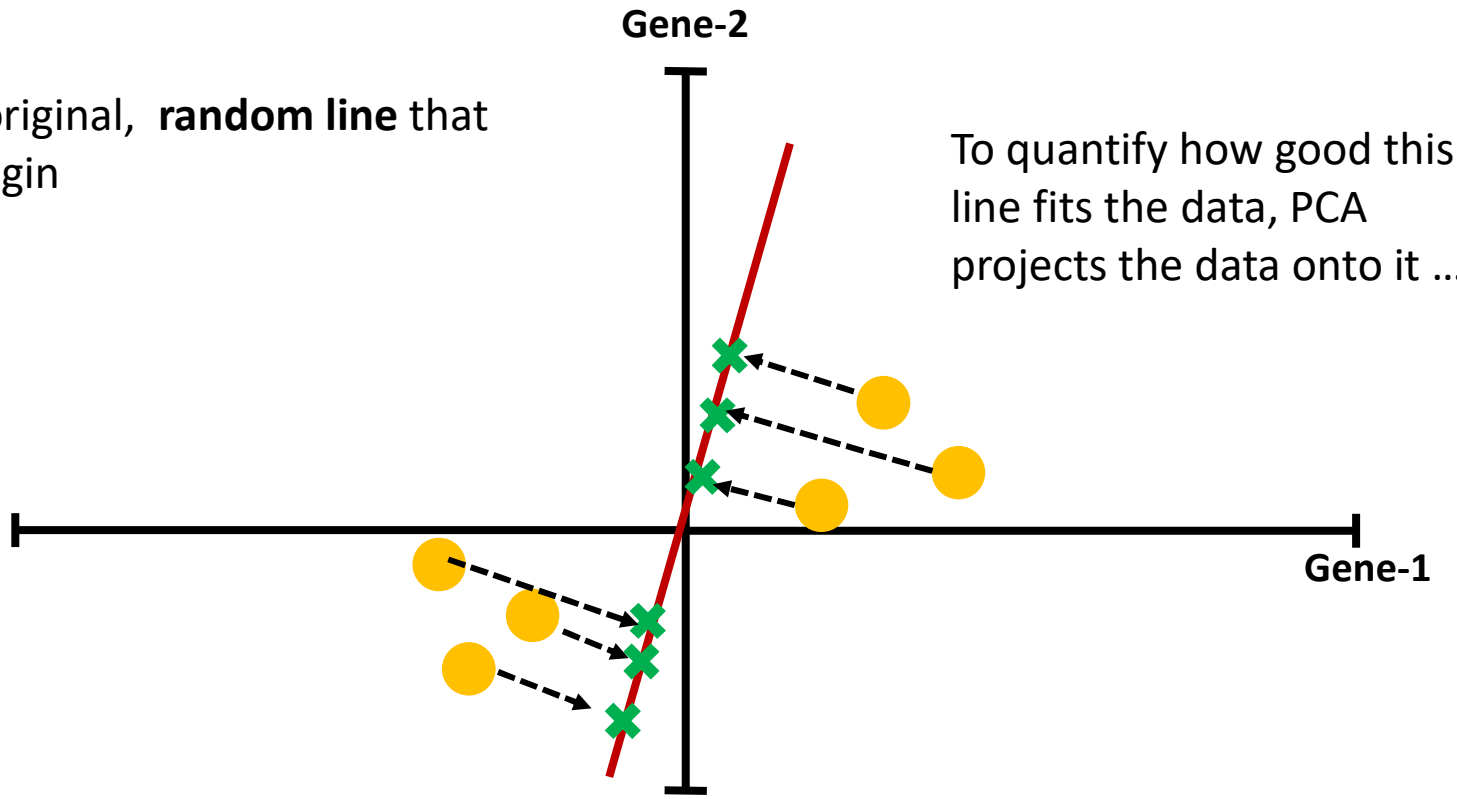
Principal Component Analysis (PCA)



- But how does PCA decides if a fit is good or not?

Principal Component Analysis (PCA)

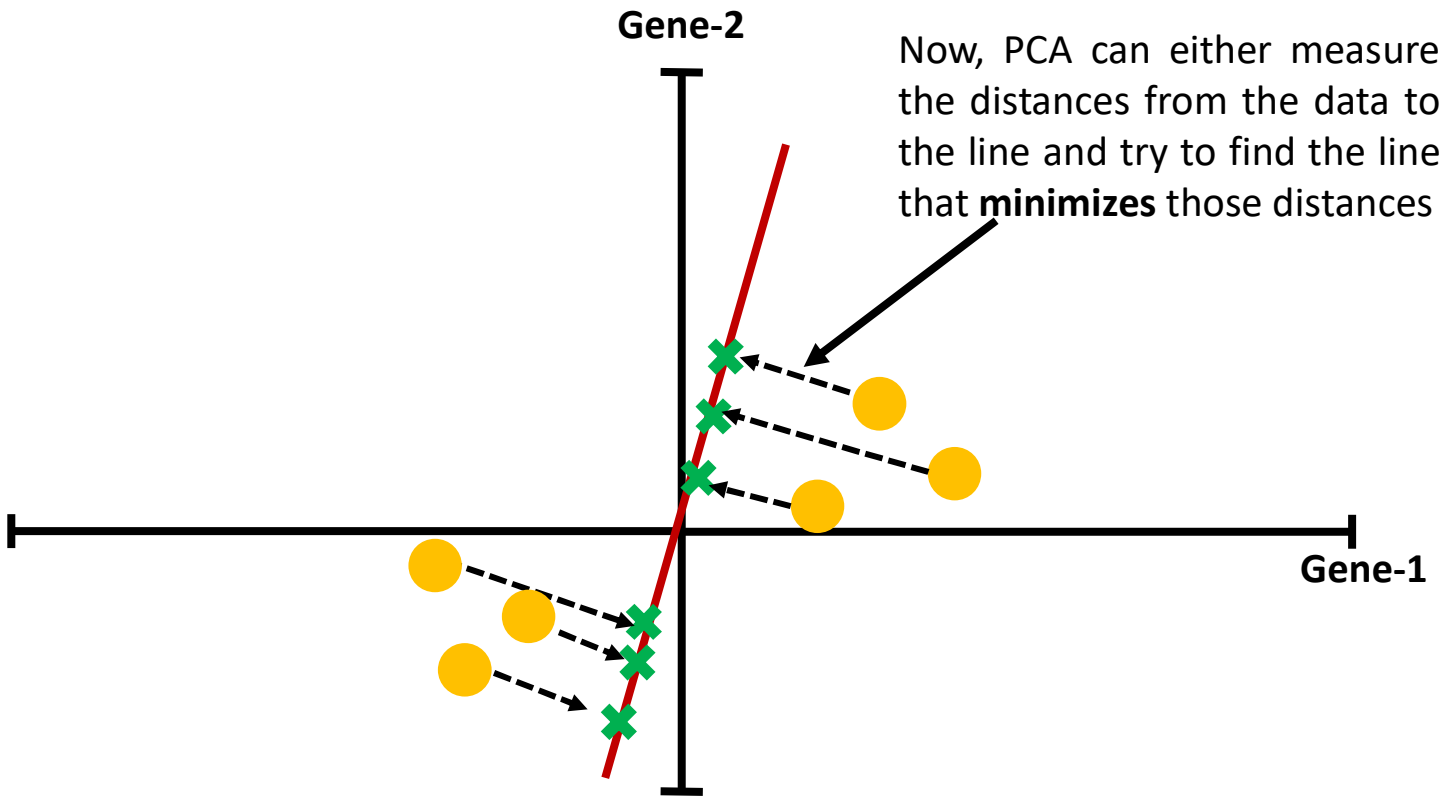
- Consider again the original, **random line** that goes through the origin



To quantify how good this line fits the data, PCA projects the data onto it …

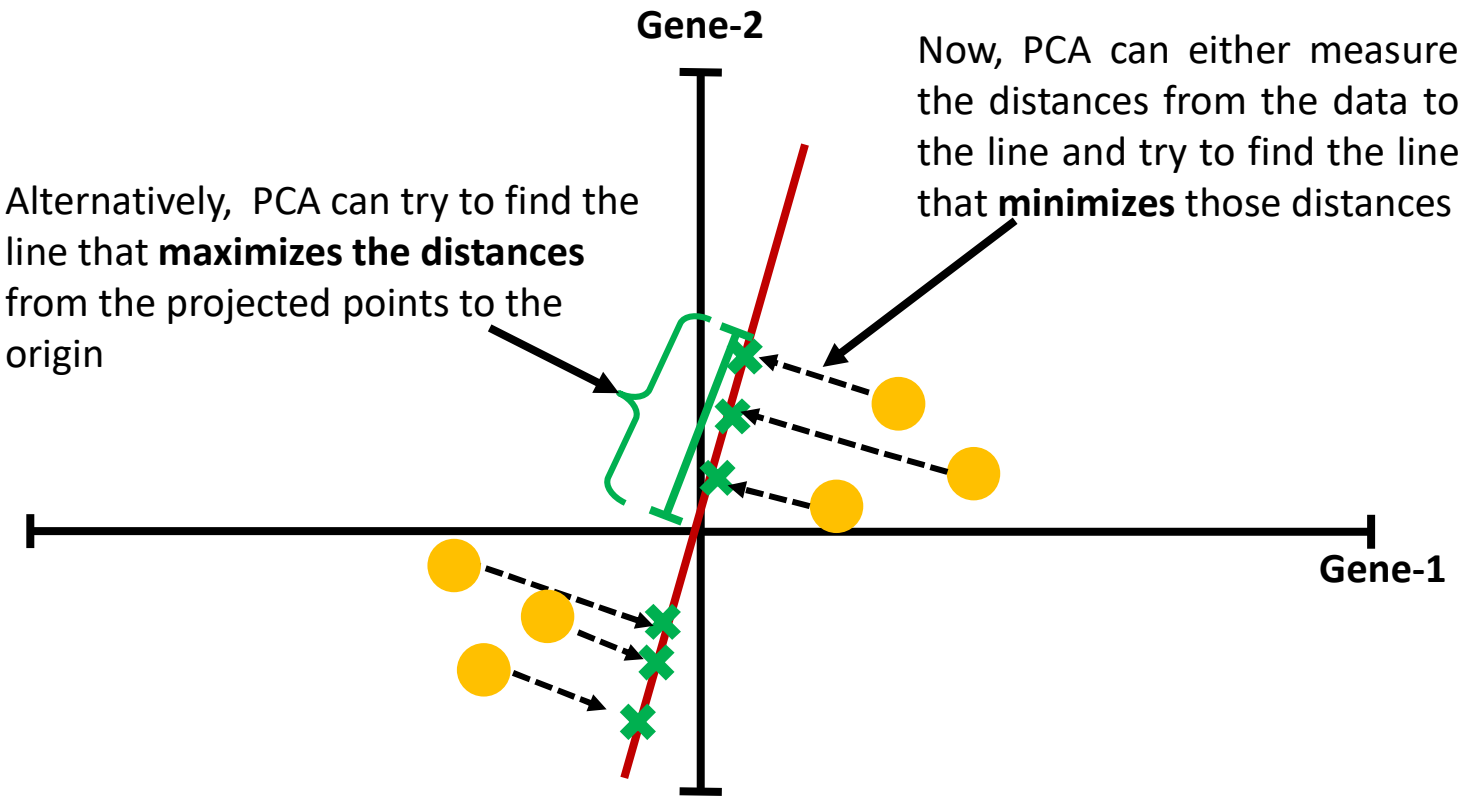
https://www.datascienceplus.com

Principal Component Analysis (PCA)



Now, PCA can either measure the distances from the data to the line and try to find the line that **minimizes** those distances

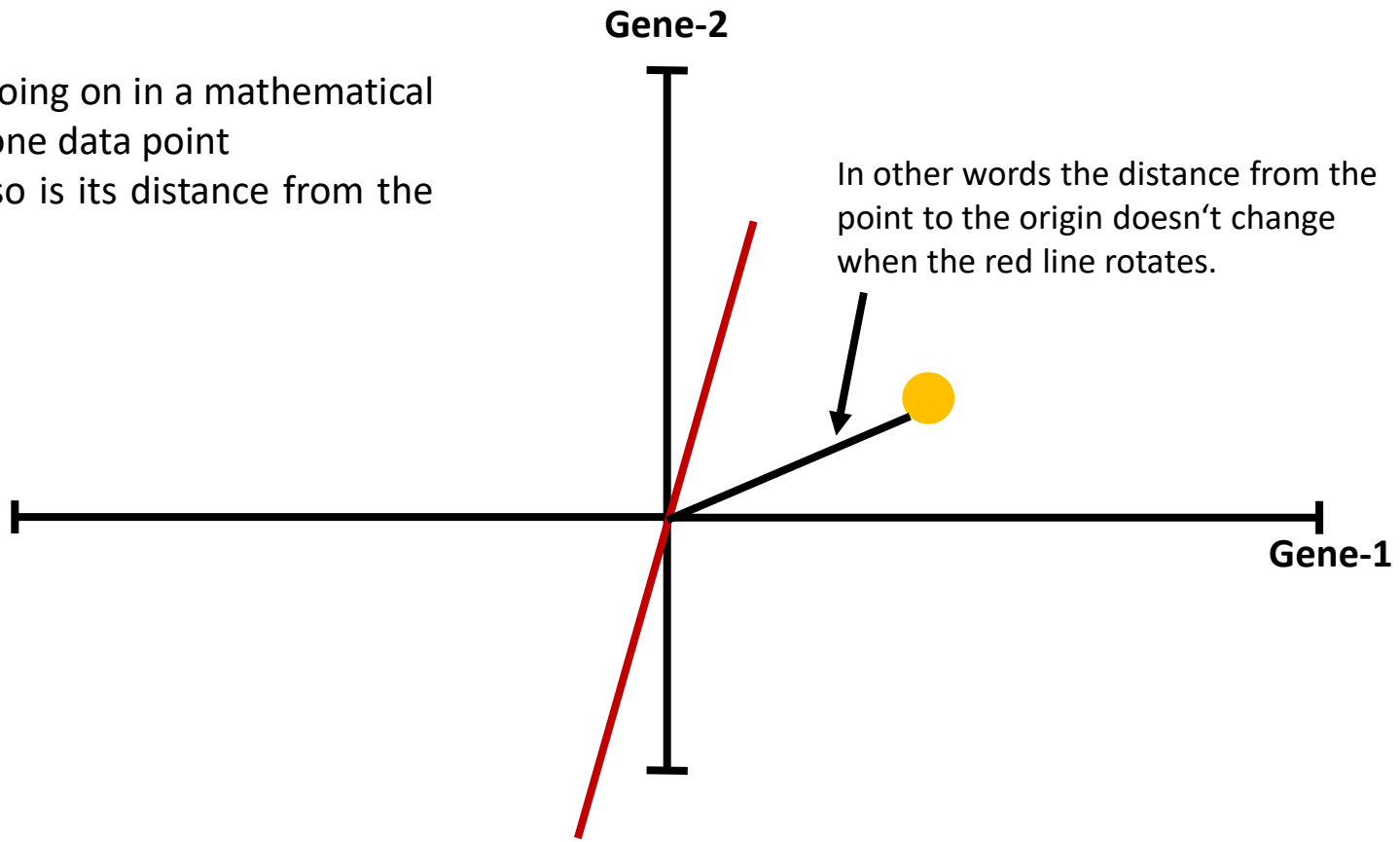
Principal Component Analysis (PCA)



https://www.datascienceplus.com

Principal Component Analysis (PCA)

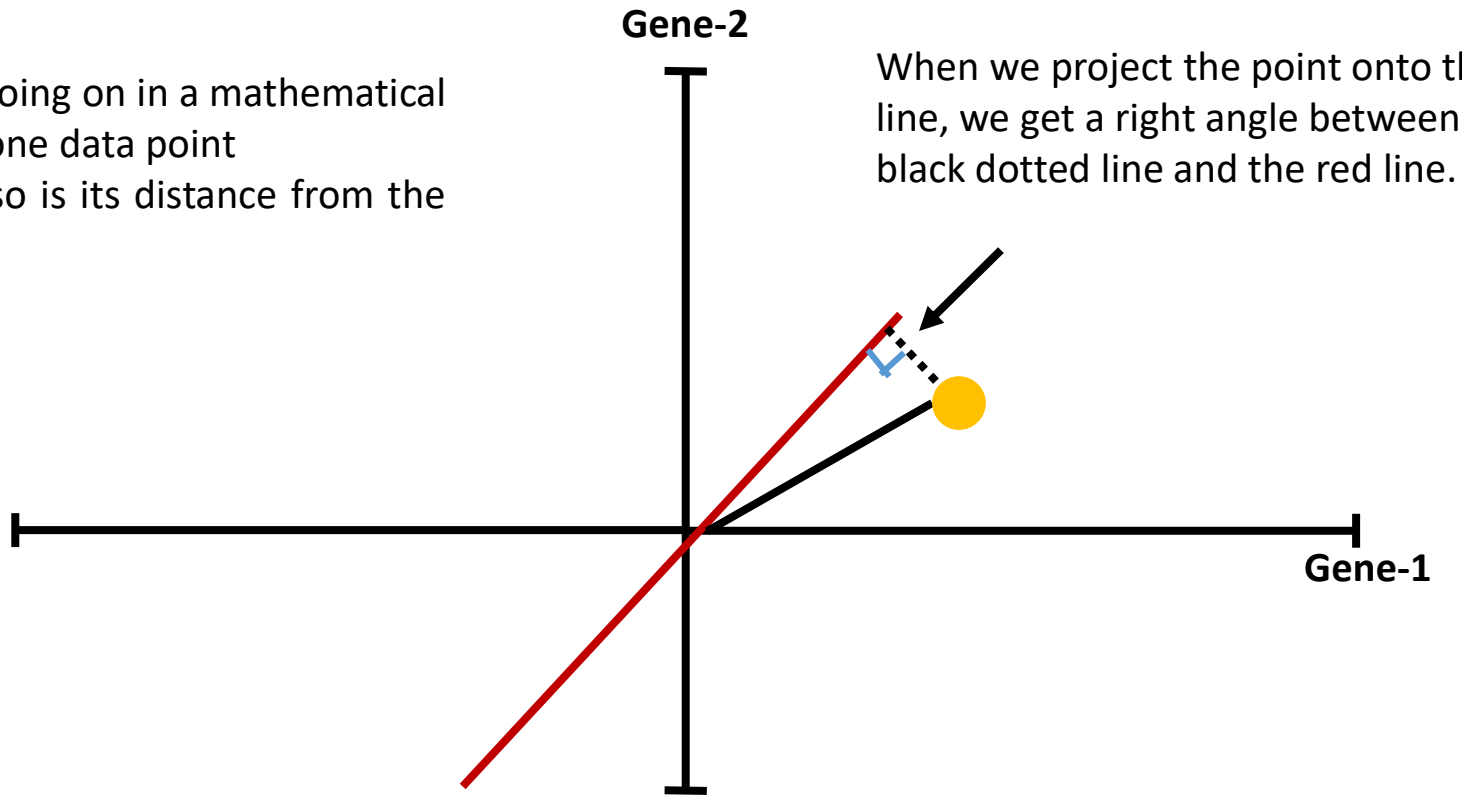
- To understand what is going on in a mathematical way, let‘s just consider one data point
- This point is fixed and so is its distance from the origin



https://www.datascienceplus.com

Principal Component Analysis (PCA)

- To understand what is going on in a mathematical way, let‘s just consider one data point
- This point is fixed and so is its distance from the origin

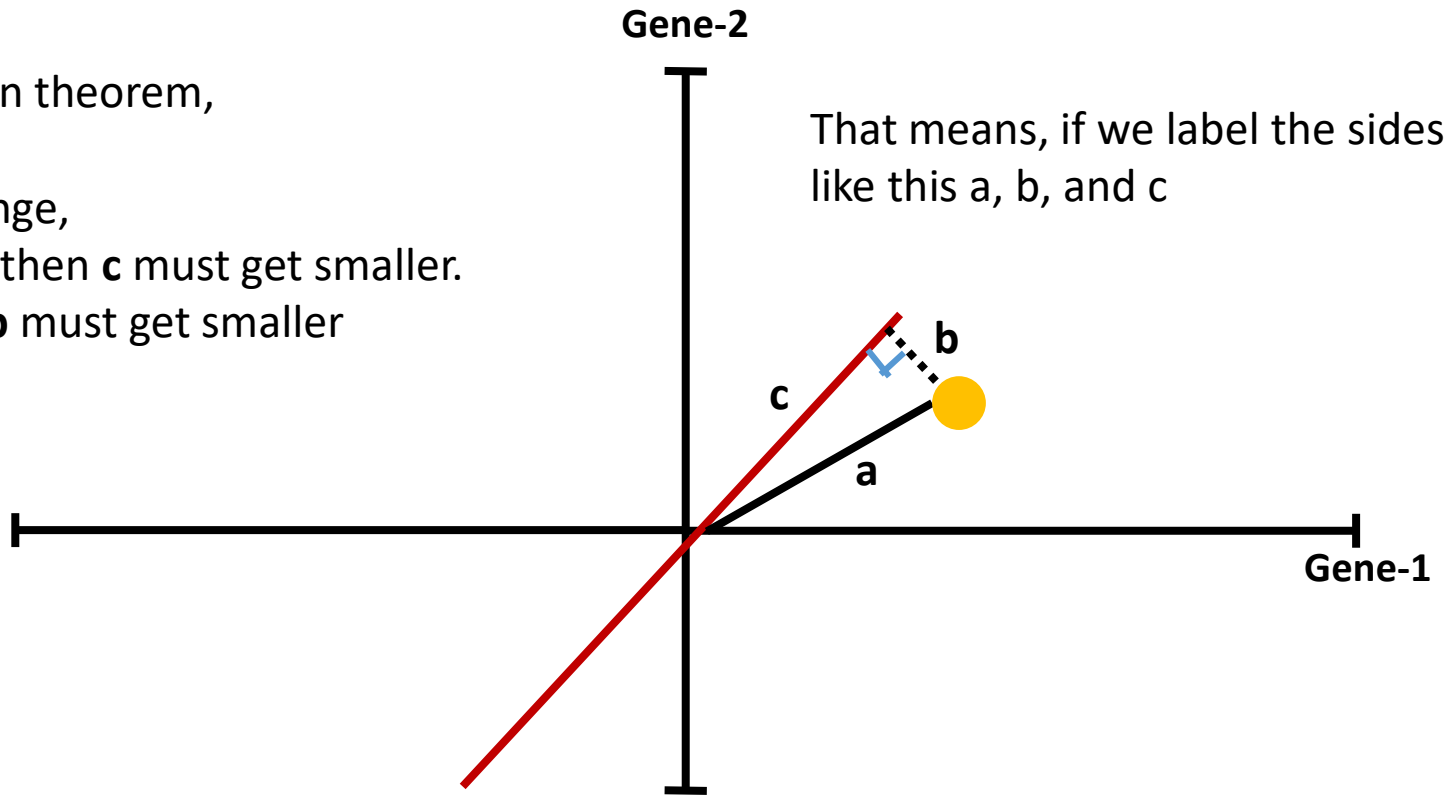


When we project the point onto the line, we get a right angle between the black dotted line and the red line.

Principal Component Analysis (PCA)

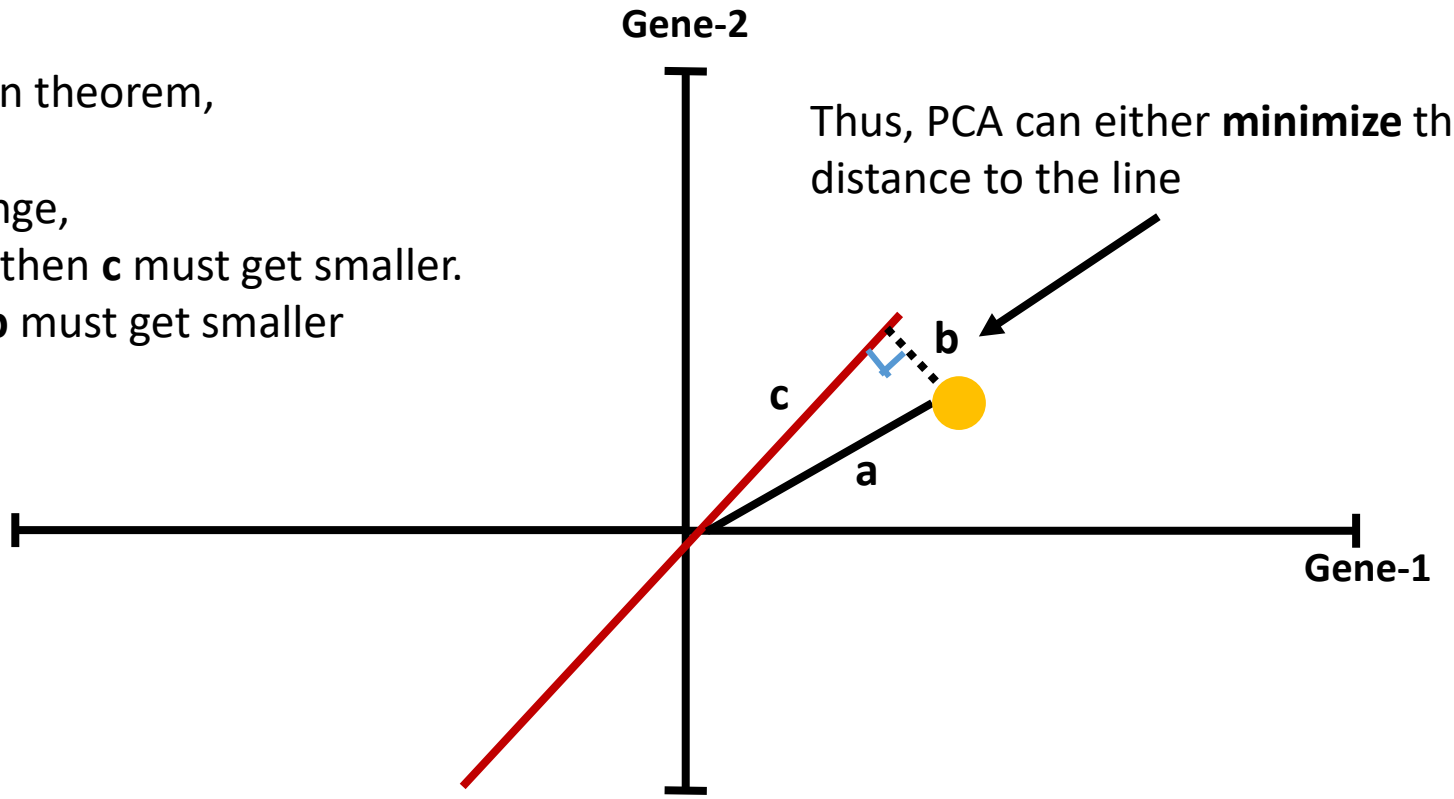
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Based on Pythagorean theorem,
  - $a^2=b^2 + c^2$.
- Since a^2 doesn't change,
  - if **b** gets bigger, then **c** must get smaller.
  - if **c** gets bigger, **b** must get smaller



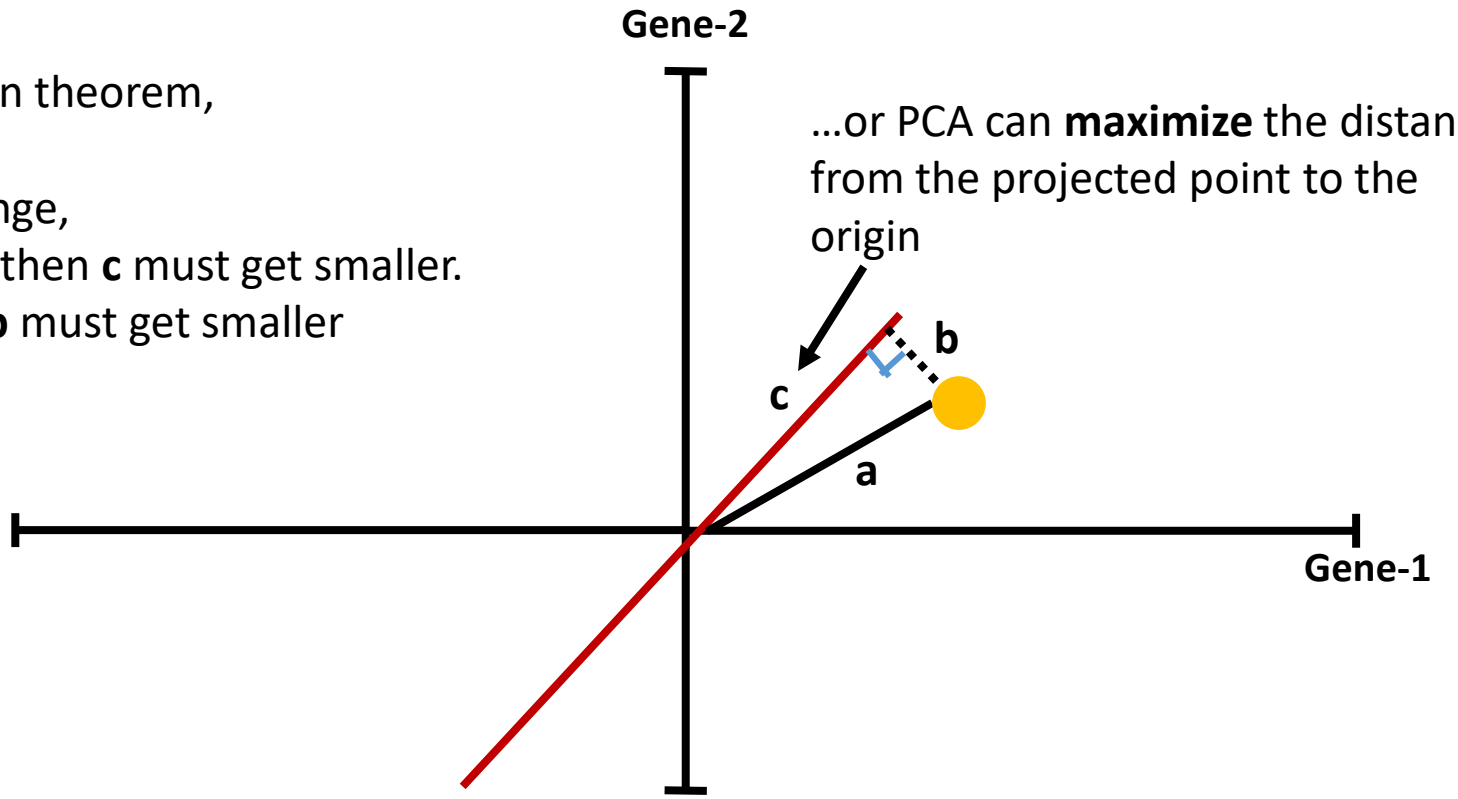
Principal Component Analysis (PCA)

- Based on Pythagorean theorem,
  - $a^2=b^2 + c^2$.
- Since a^2 doesn‘t change,
  - if **b** gets bigger, then **c** must get smaller.
  - if **c** gets bigger, **b** must get smaller



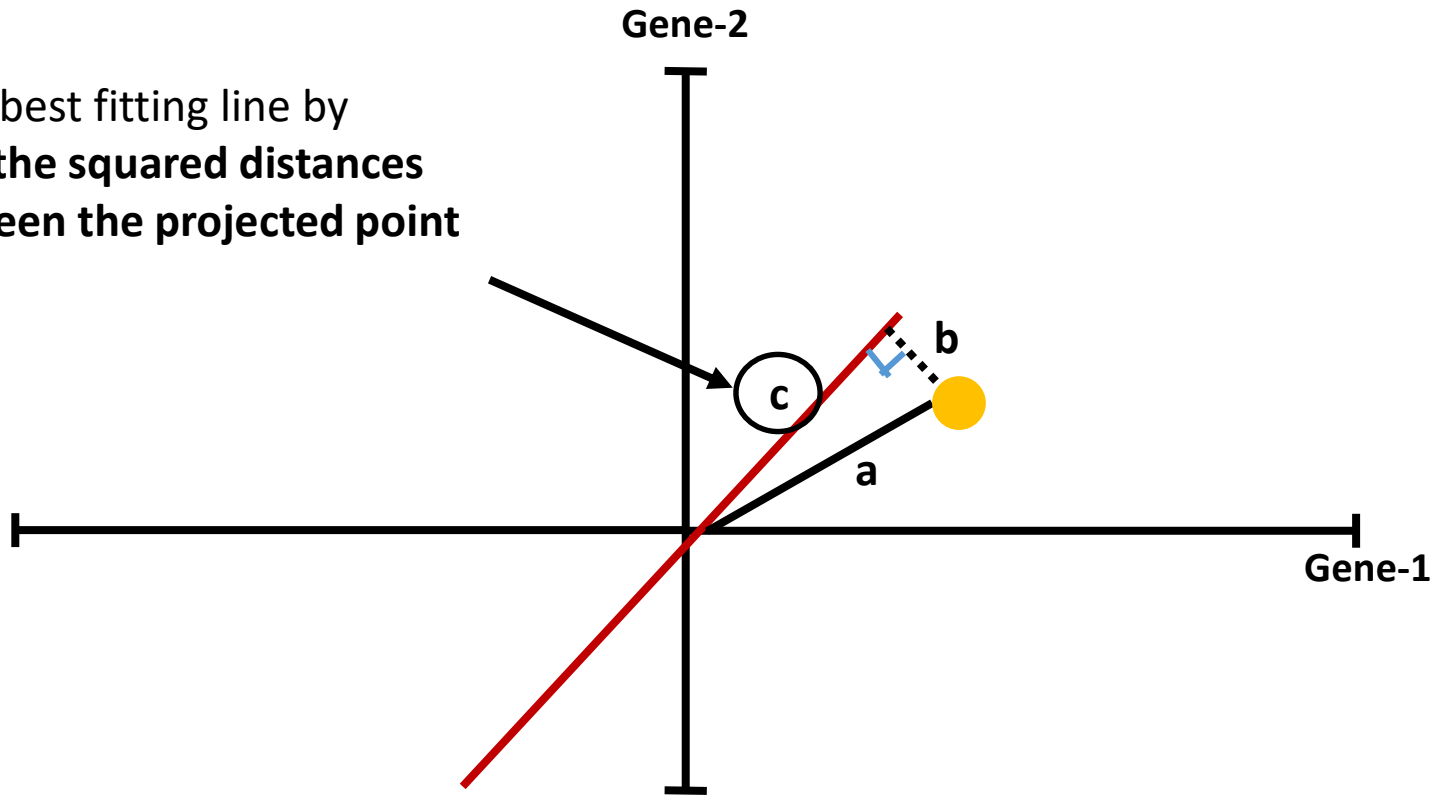
Principal Component Analysis (PCA)

- Based on Pythagorean theorem,
  - $a^2=b^2 + c^2$.
- Since a^2 doesn‘t change,
  - if **b** gets bigger, then **c** must get smaller.
  - if **c** gets bigger, **b** must get smaller



Principal Component Analysis (PCA)

Normally, PCA finds the best fitting line by **maximizing the sum of the squared distances from the distance between the projected point and the origin**

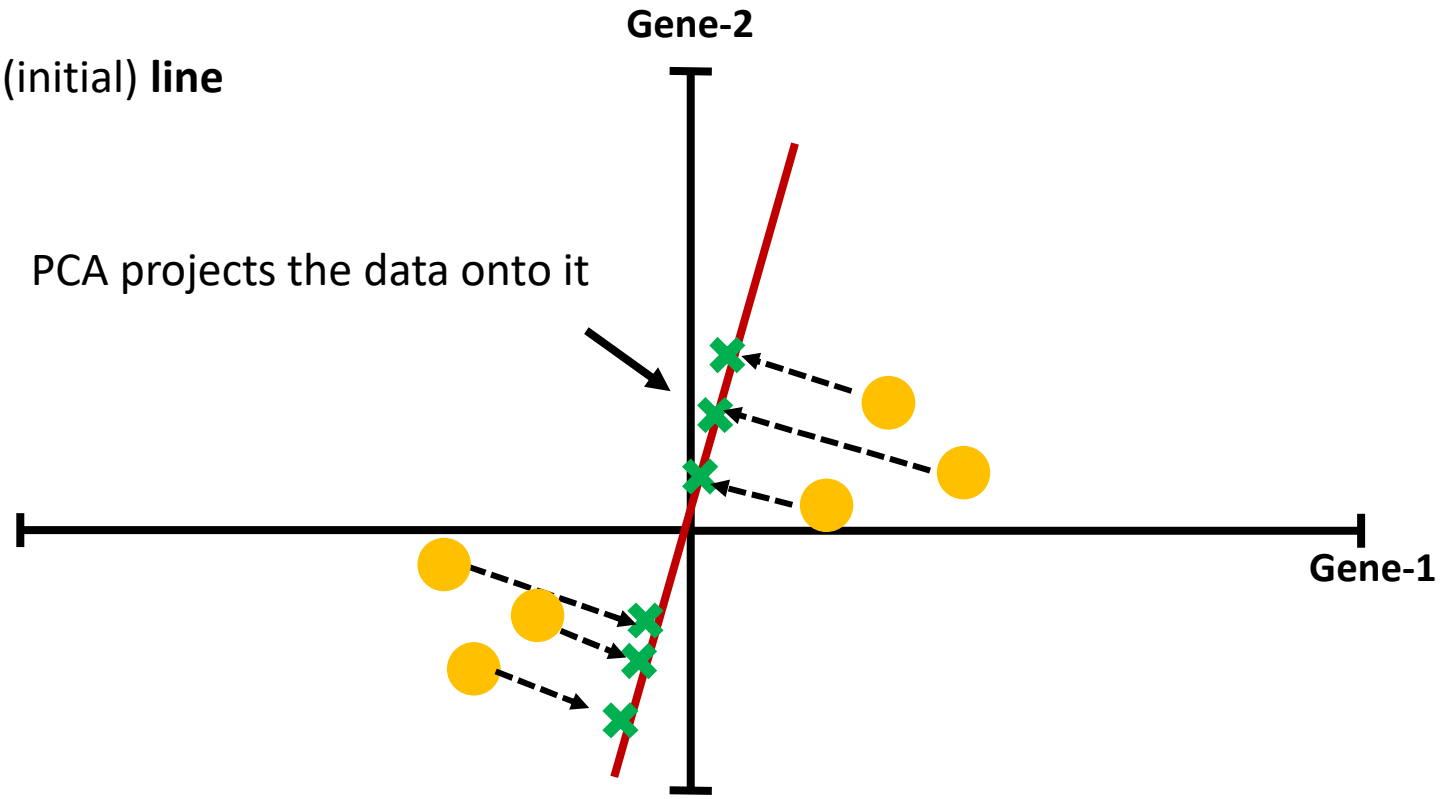


https://www.datascienceplus.com

Principal Component Analysis (PCA)

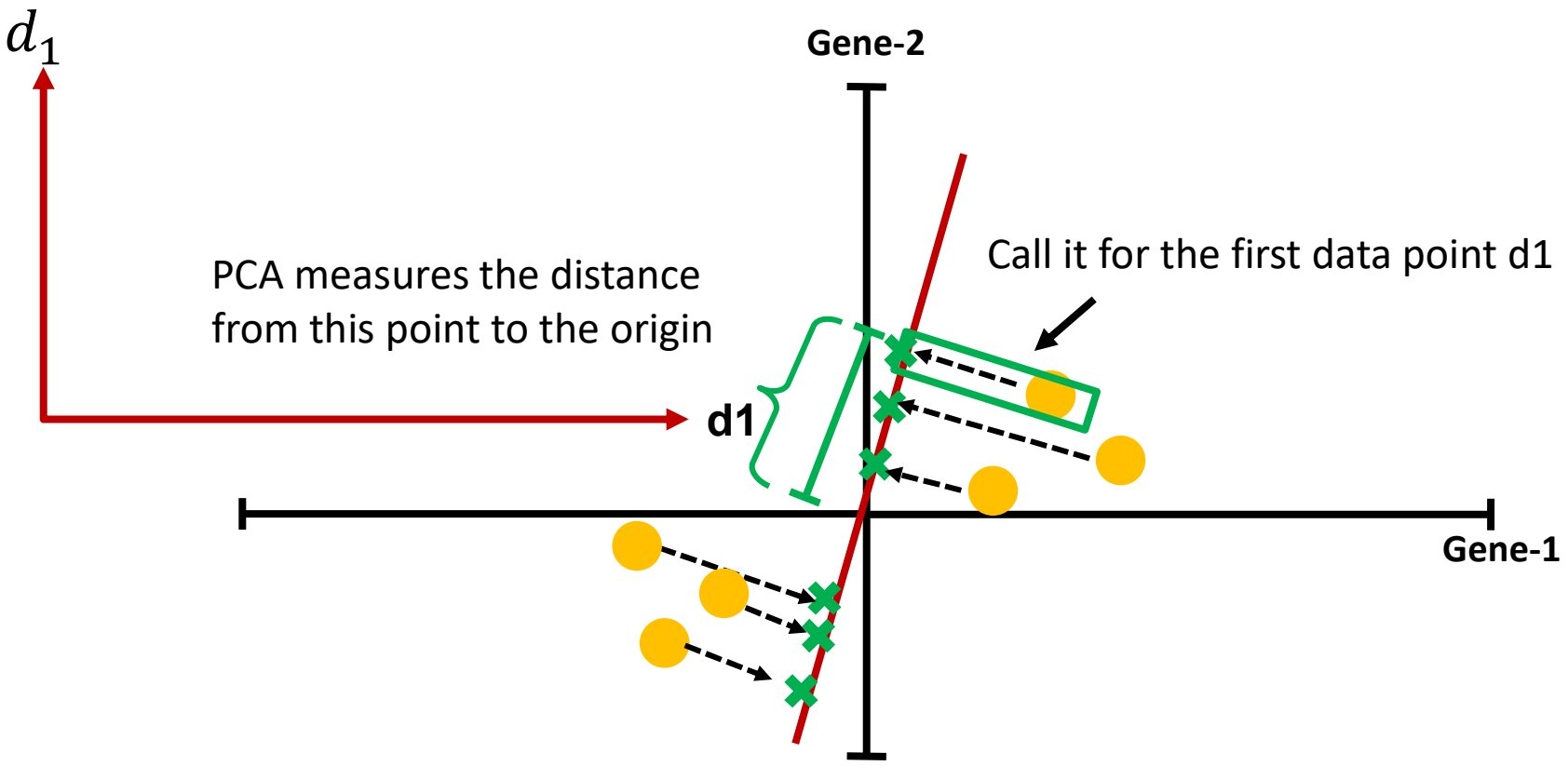
For the original (initial) **line**

PCA projects the data onto it

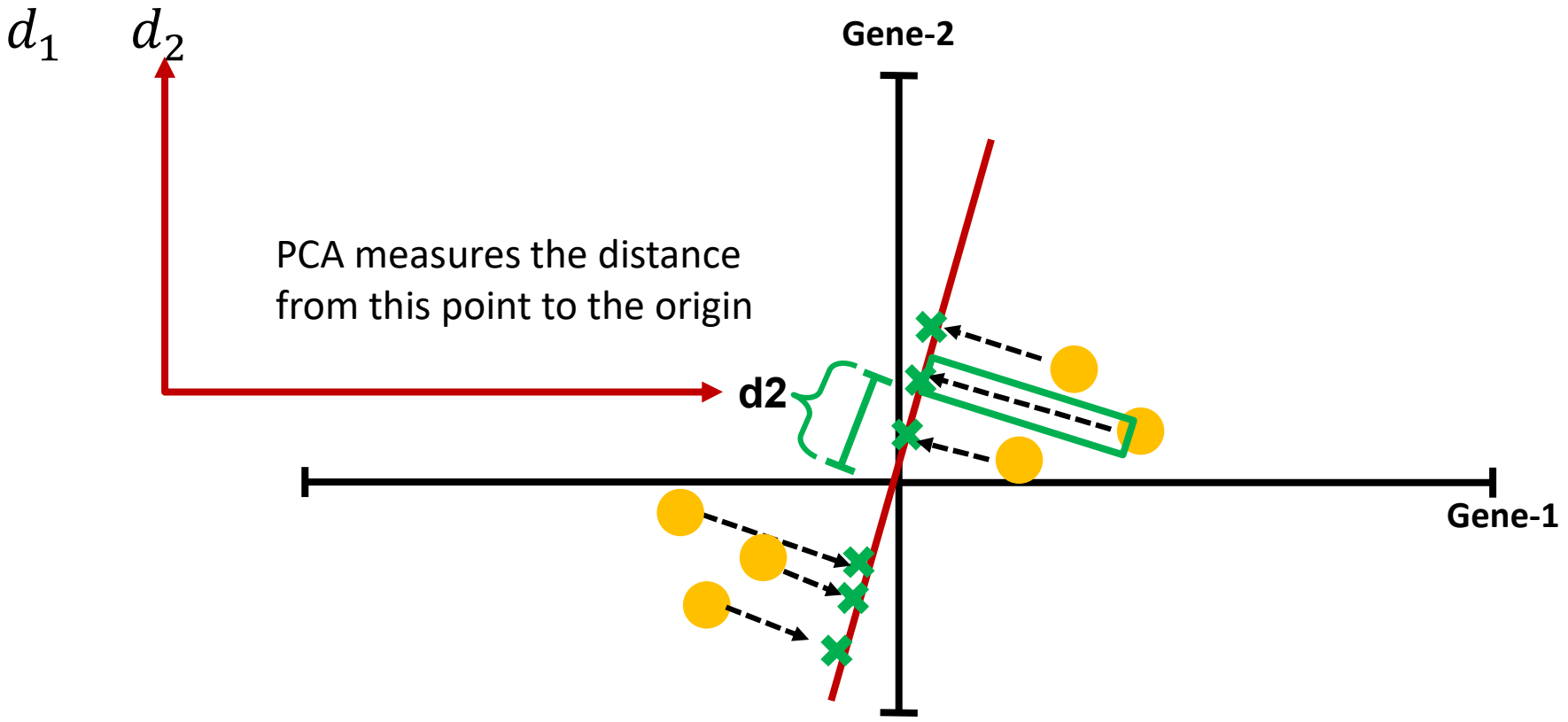


https://www.datascienceplus.com

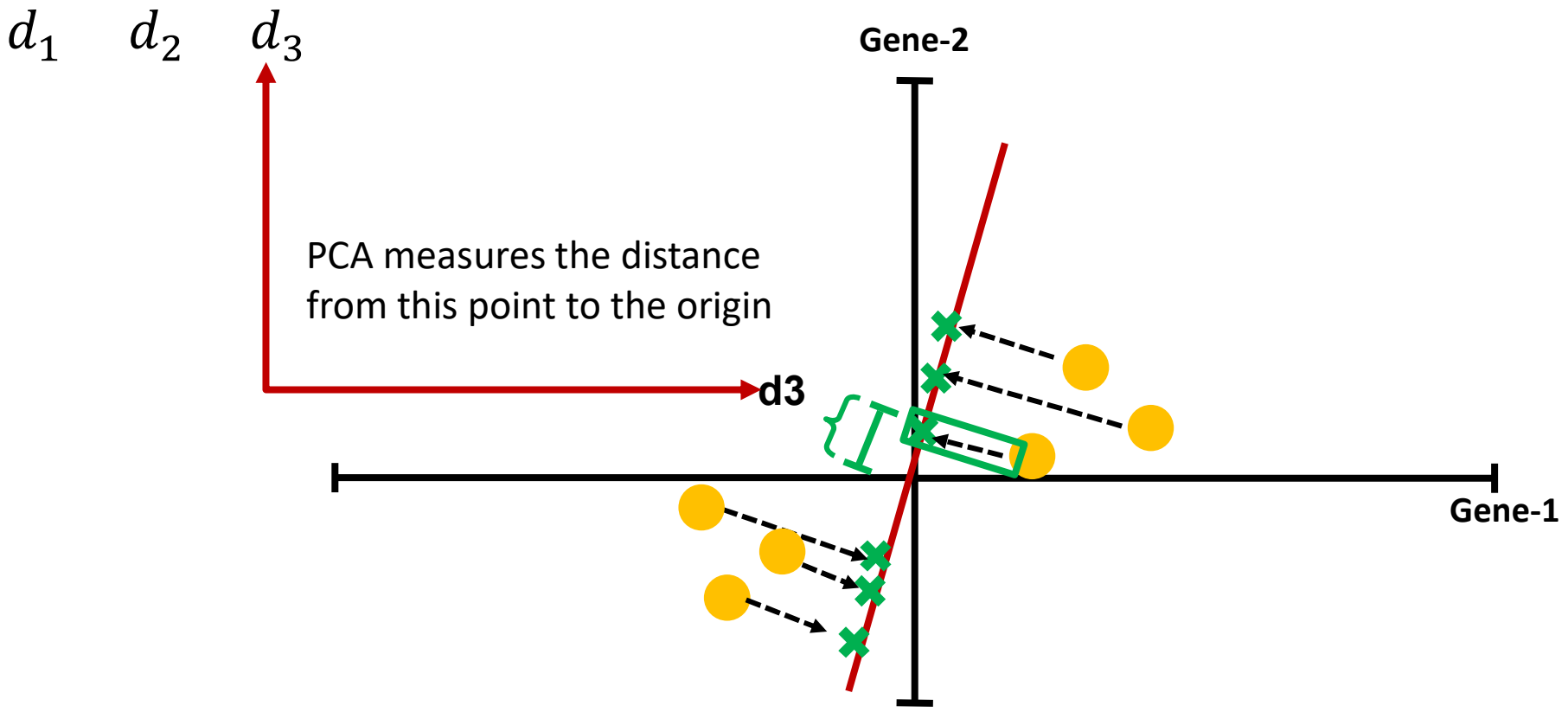
Principal Component Analysis (PCA)



Principal Component Analysis (PCA)



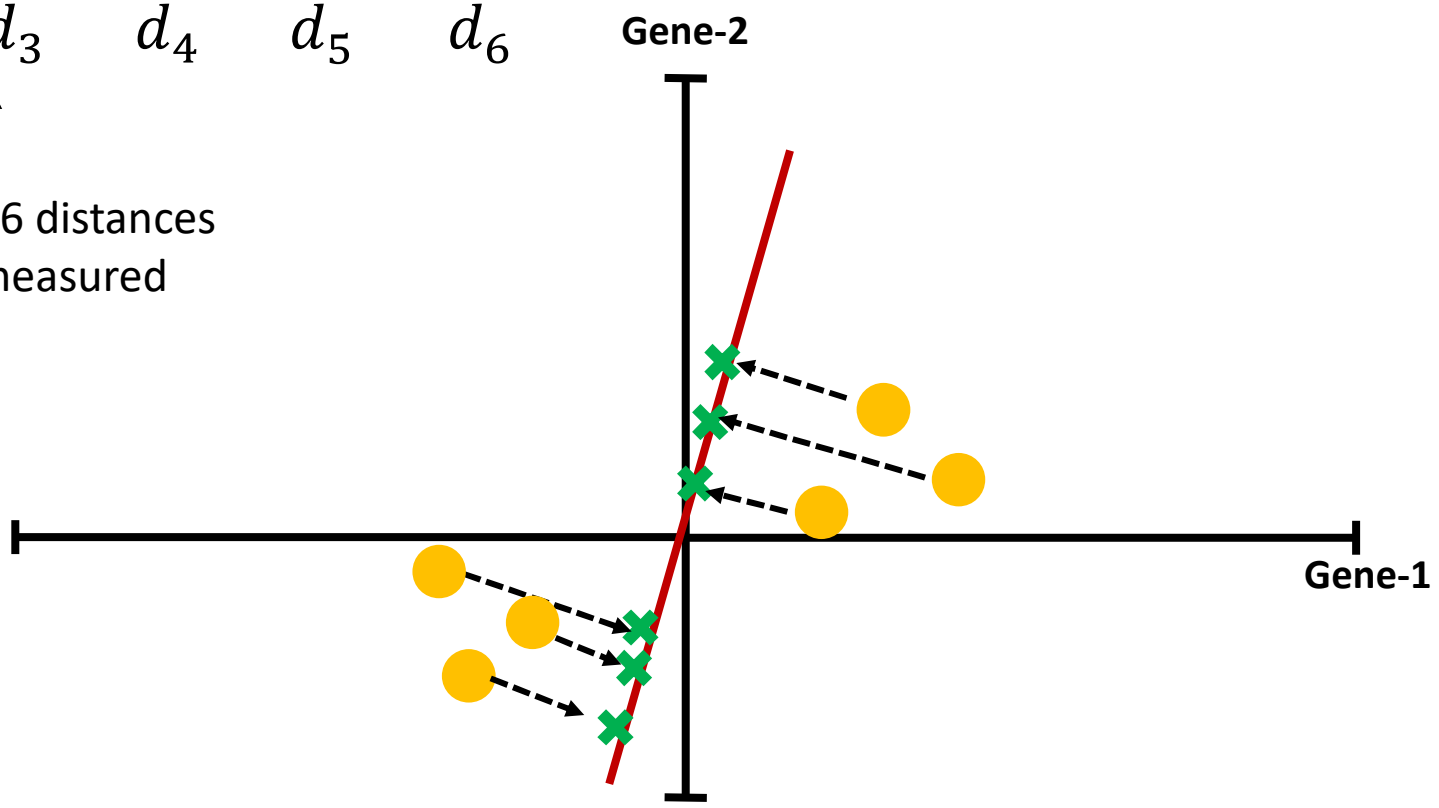
Principal Component Analysis (PCA)



Principal Component Analysis (PCA)

d_1 d_2 d_3 d_4 d_5 d_6

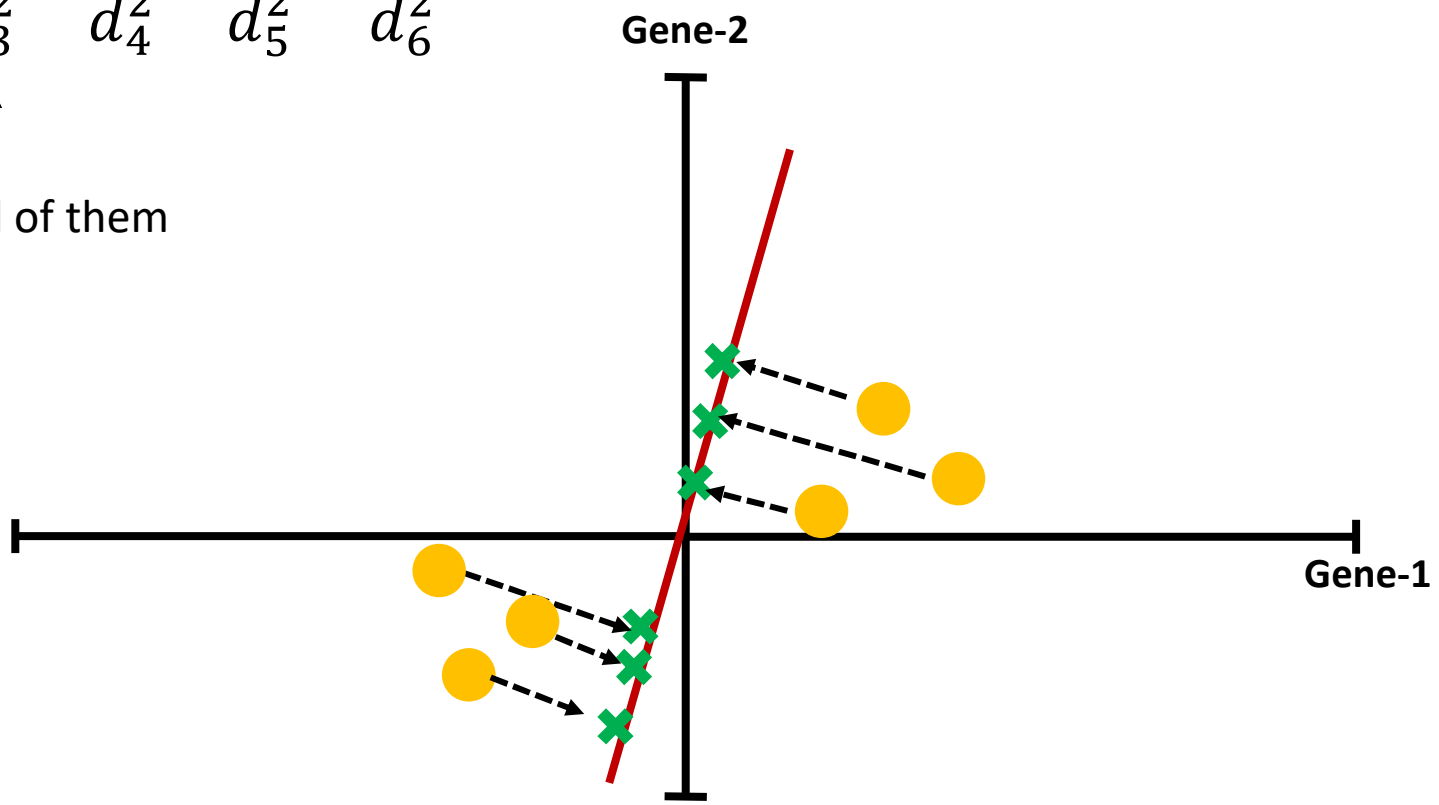
Here are all 6 distances that we measured



Principal Component Analysis (PCA)

d_1^2 d_2^2 d_3^2 d_4^2 d_5^2 d_6^2

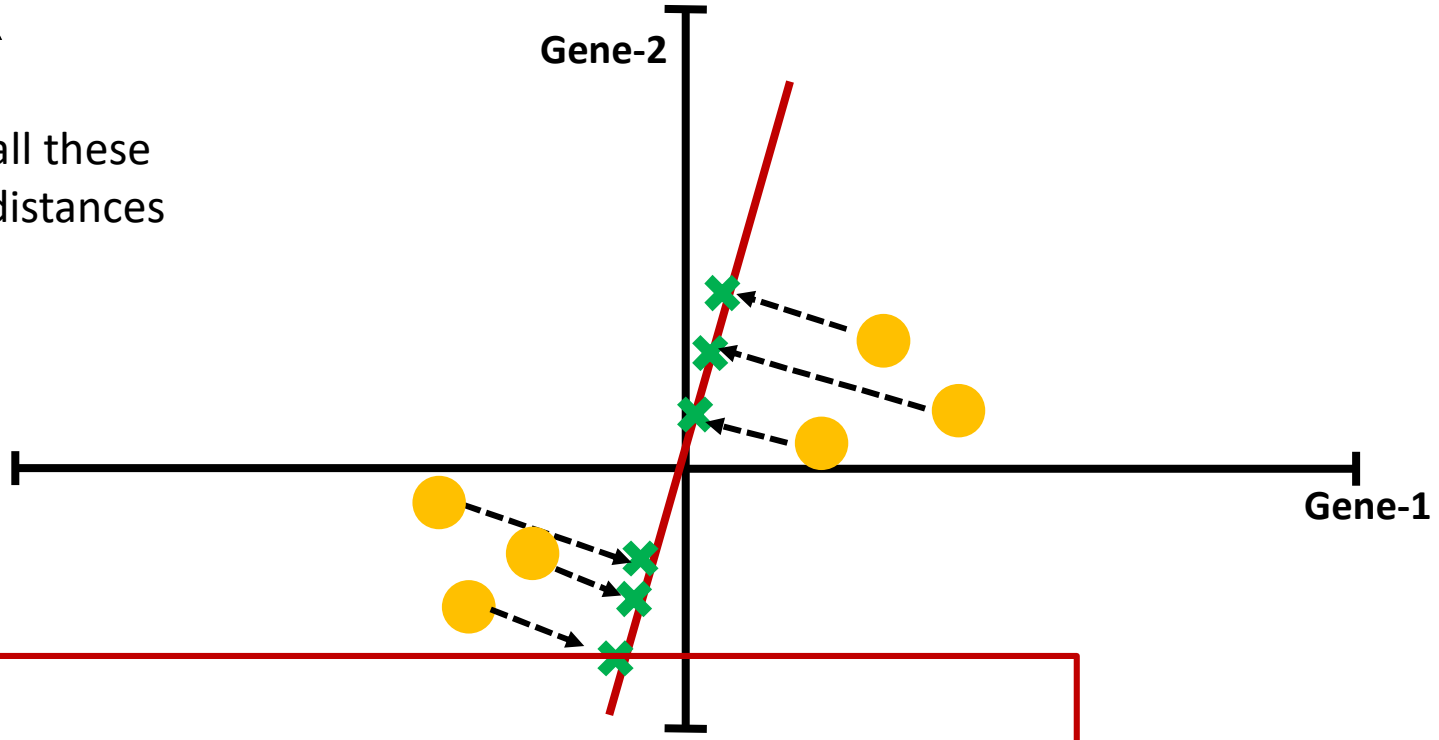
Square all of them



Principal Component Analysis (PCA)

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

Sum up all these squared distances



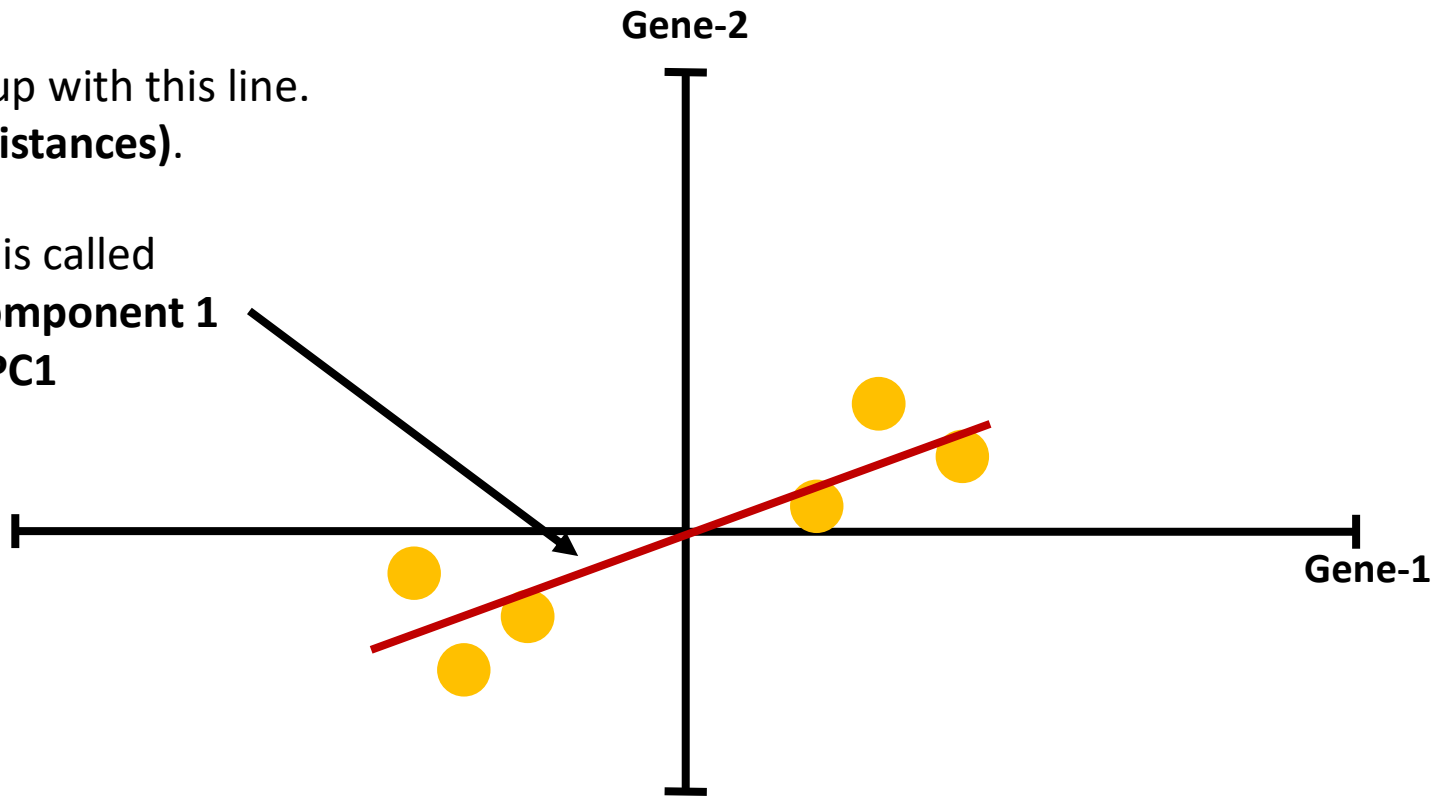
Next steps:

- Rotate the line
- Repeat all analysis until we identify the line which provides the largest **sum of squared distances**

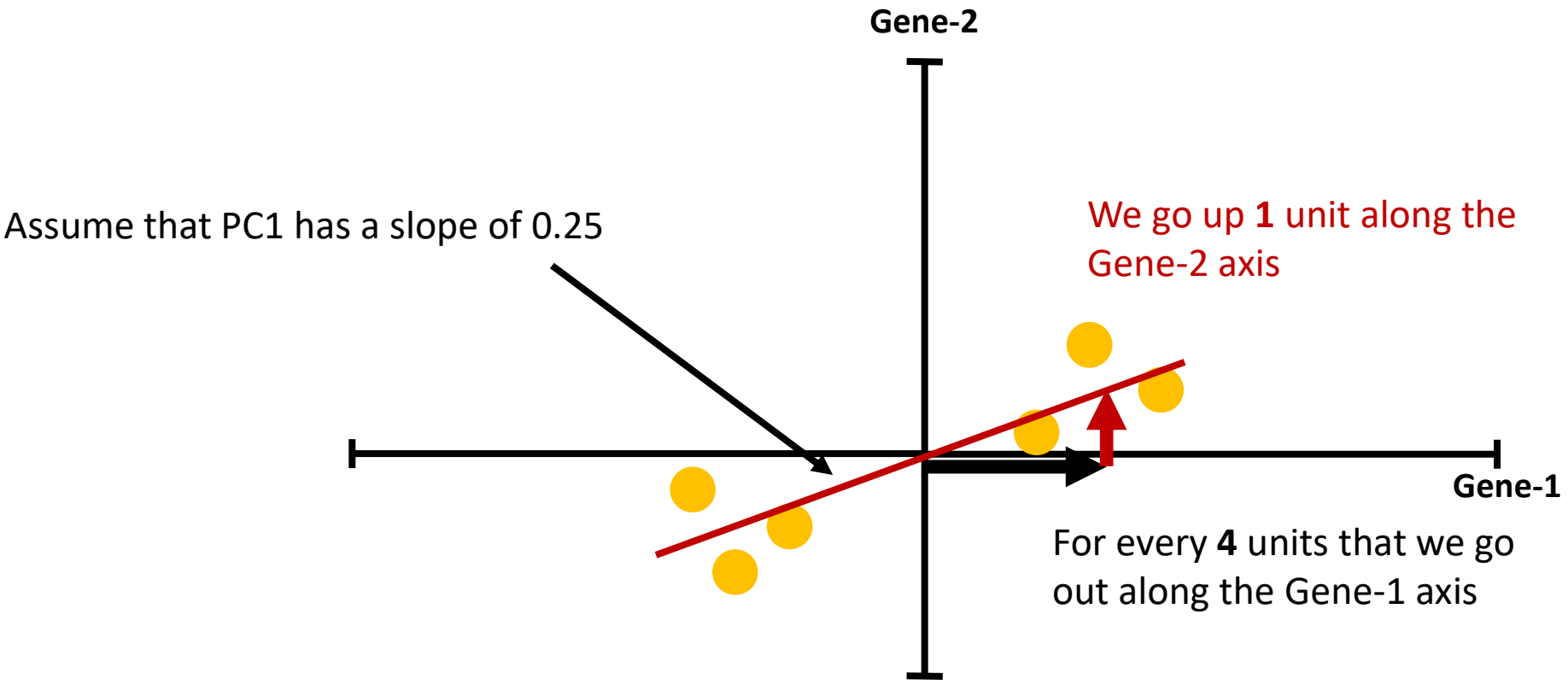
Principal Component Analysis (PCA)

Assume that we end up with this line.
It has the largest **SS(distances)**.

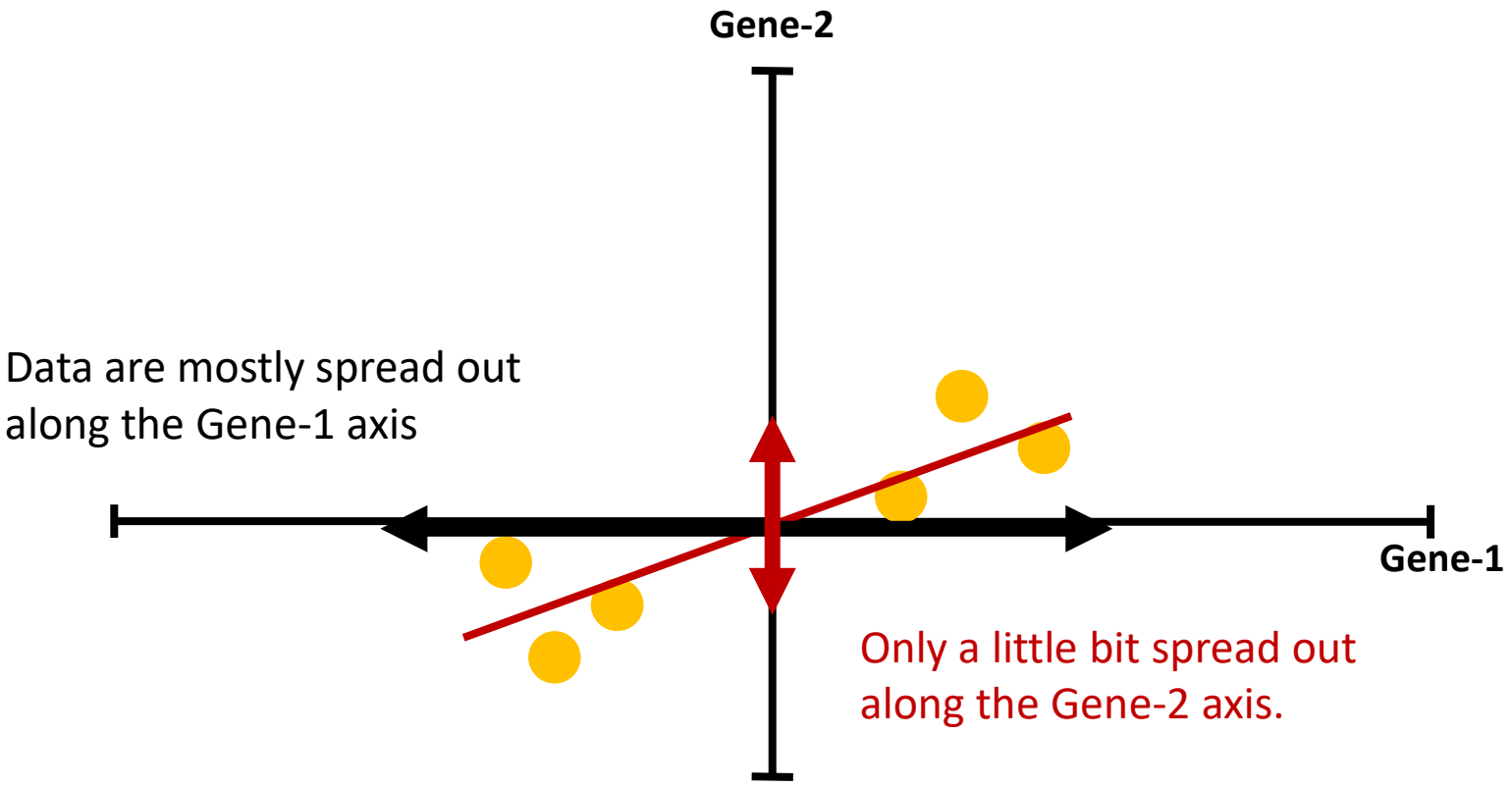
This line is called
Principal Component 1
or **PC1**



Principal Component Analysis (PCA)



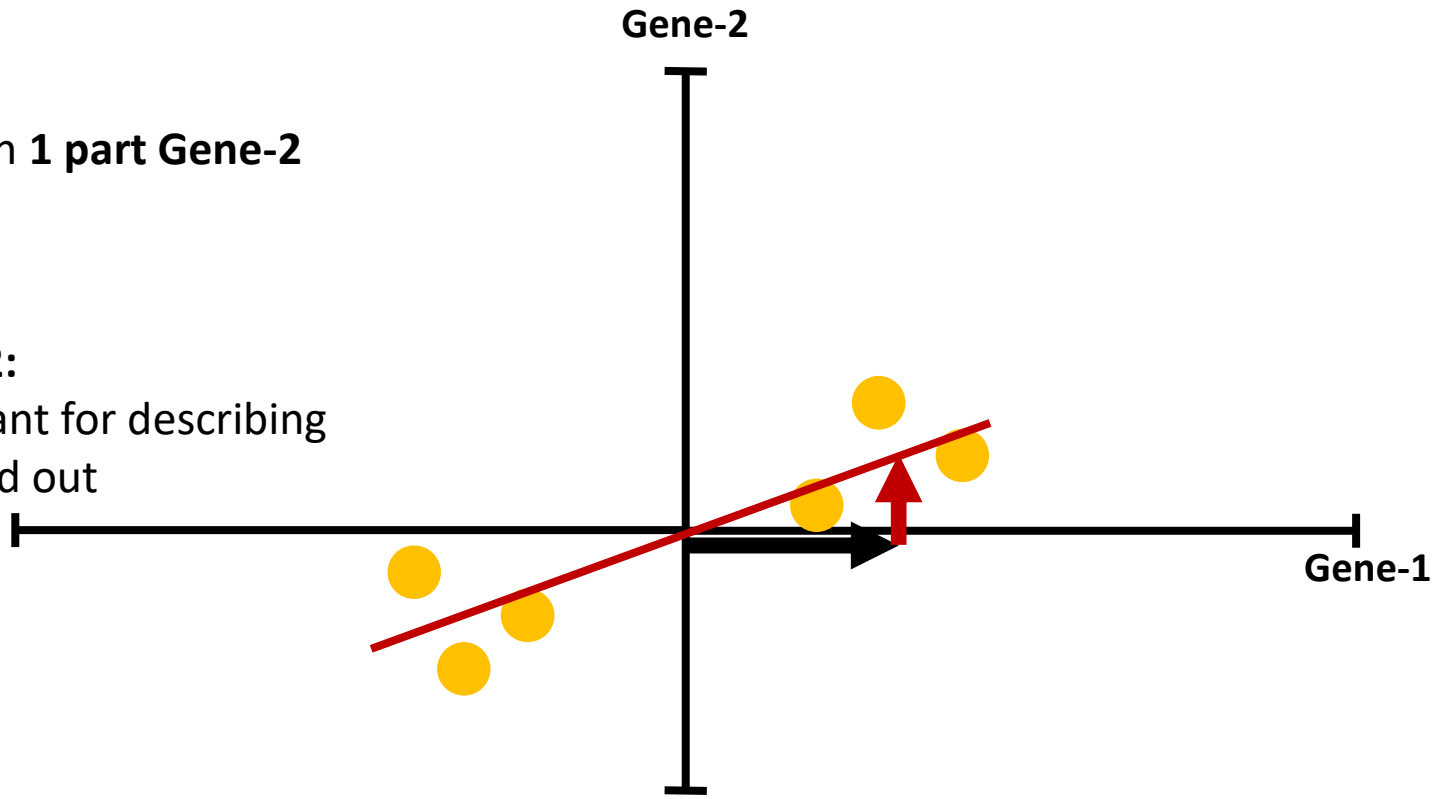
Principal Component Analysis (PCA)



Principal Component Analysis (PCA)

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

<u>The ratio</u>
4 x Gene-1 : 1 x Gene-2:
Gene-1 is more important for describing how the data are spread out

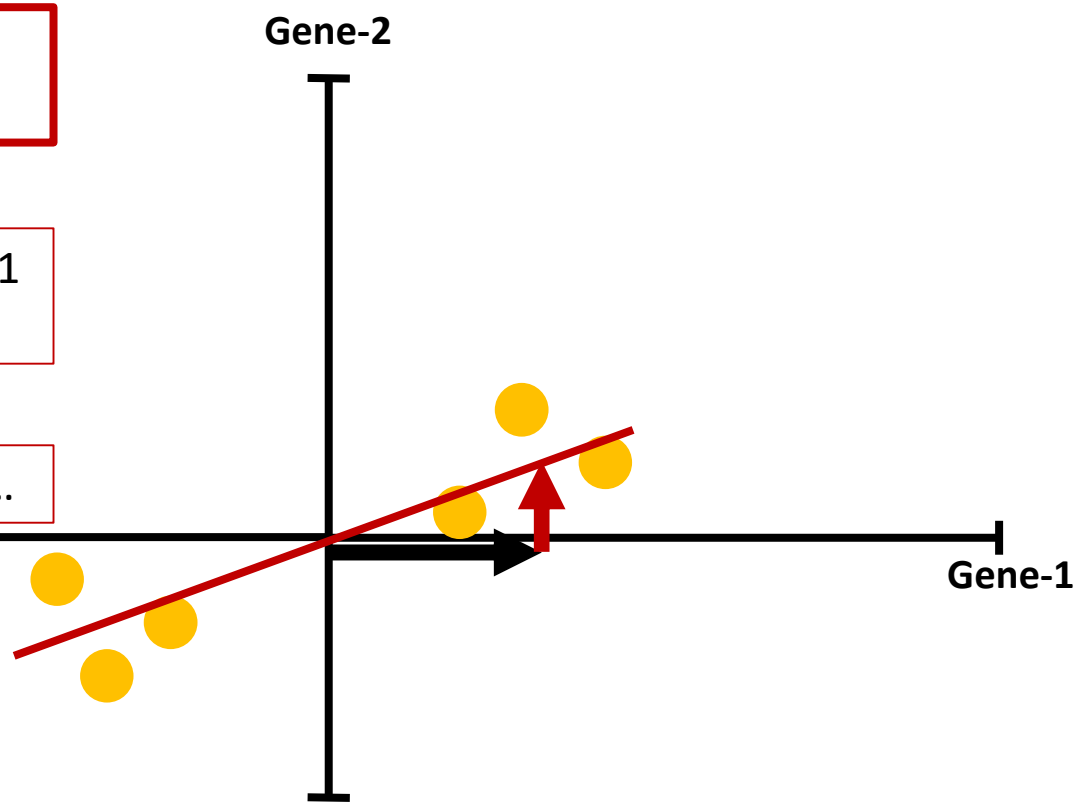


Principal Component Analysis (PCA)

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

It is called a **linear combination** of Gene-1 and 2

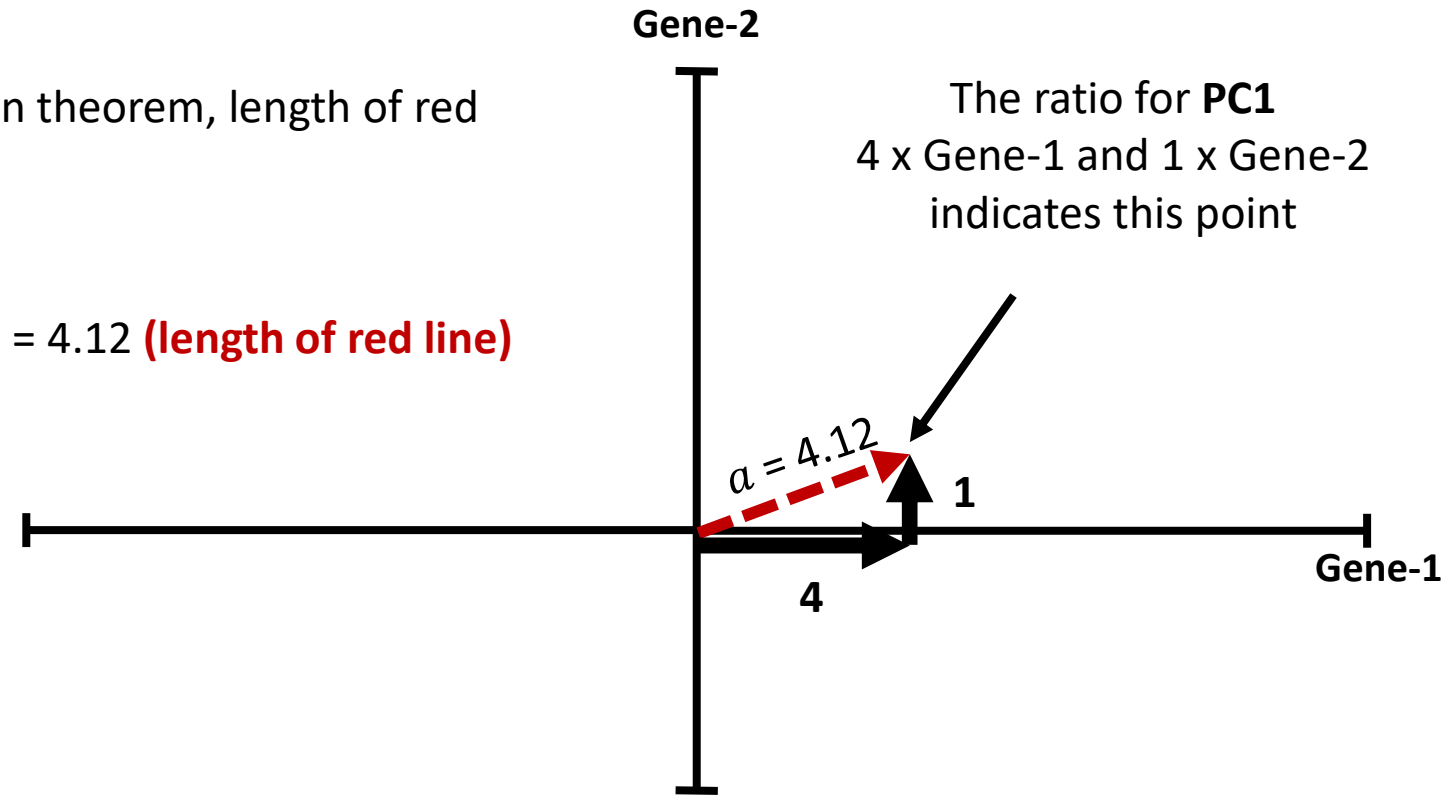
PC1 is a linear combination of variables…



Principal Component Analysis (PCA)

Based on Pythagorean theorem, length of red line:

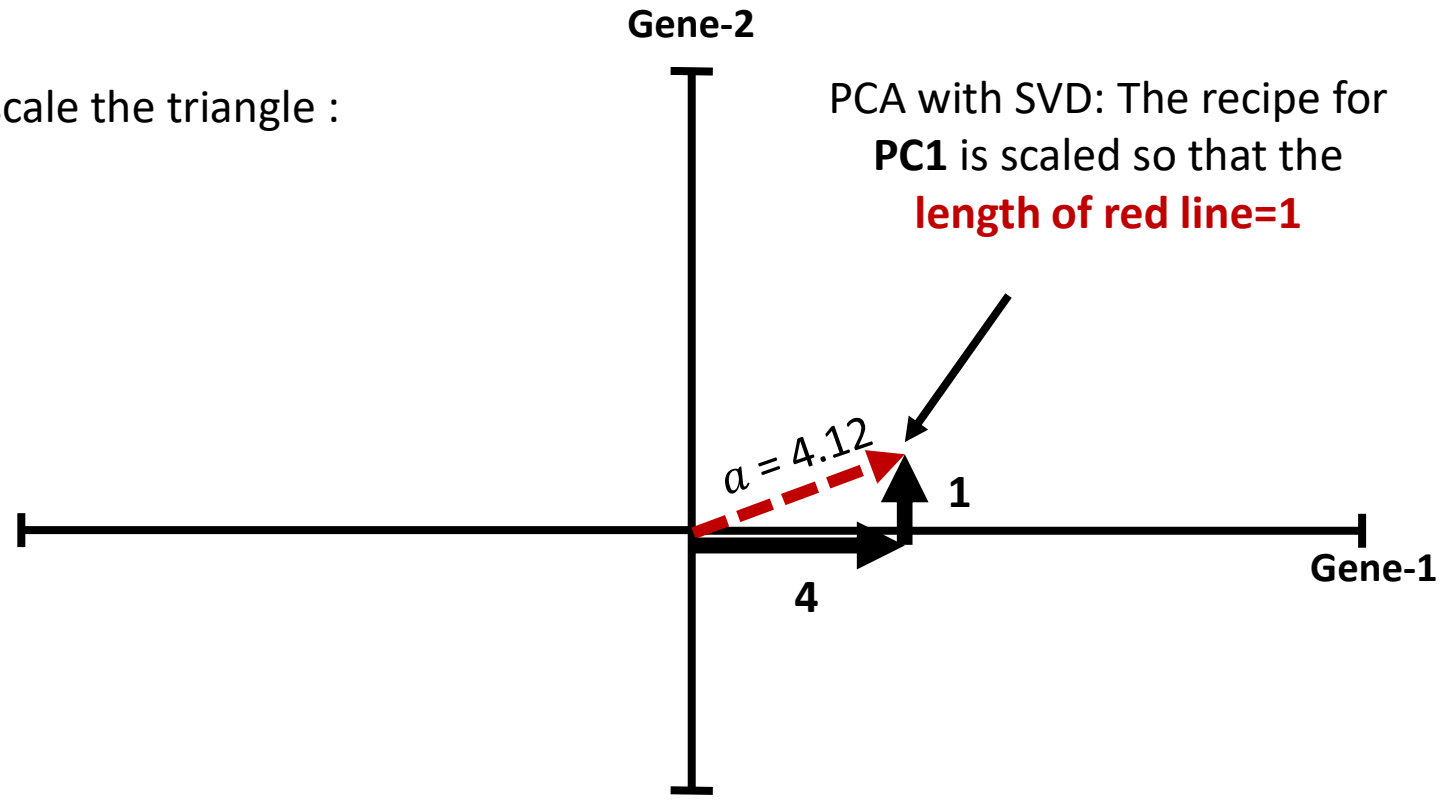
- $a^2=b^2 + c^2$
- $a^2=4^2 + 1^2$
- $a^2= 17$ ➔ $a = 4.12$ **(length of red line)**



Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

For SVD, we have to scale the triangle :

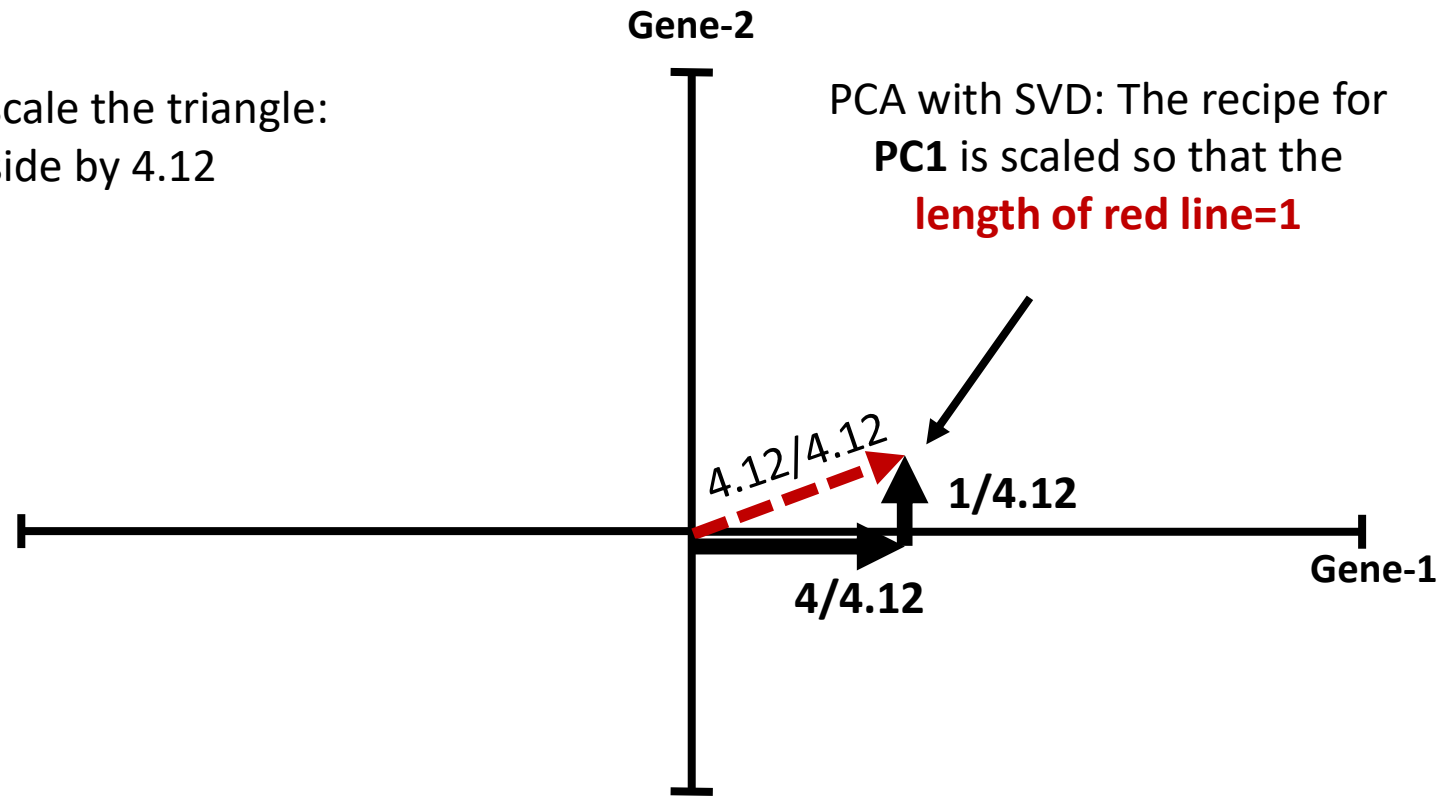


SVD: Singular Value Decomposition

Principal Component Analysis (PCA)

For SVD, we have to scale the triangle:

- divide each side by 4.12



SVD: Singular Value Decomposition

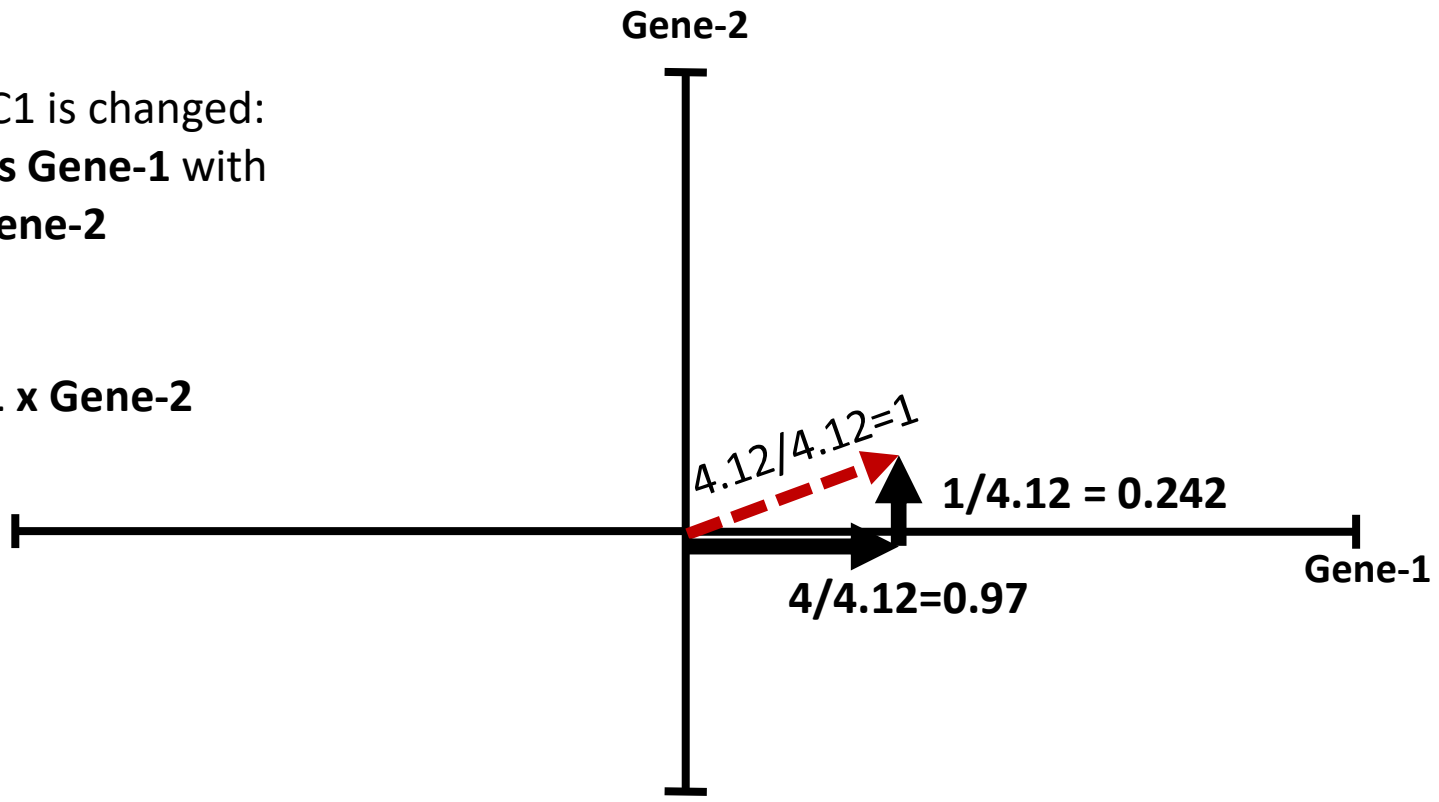
Principal Component Analysis (PCA)

The recipe to make PC1 is changed:

- Mix **0.97 parts Gene-1** with **0.242 parts Gene-2**

The ratio is still same:

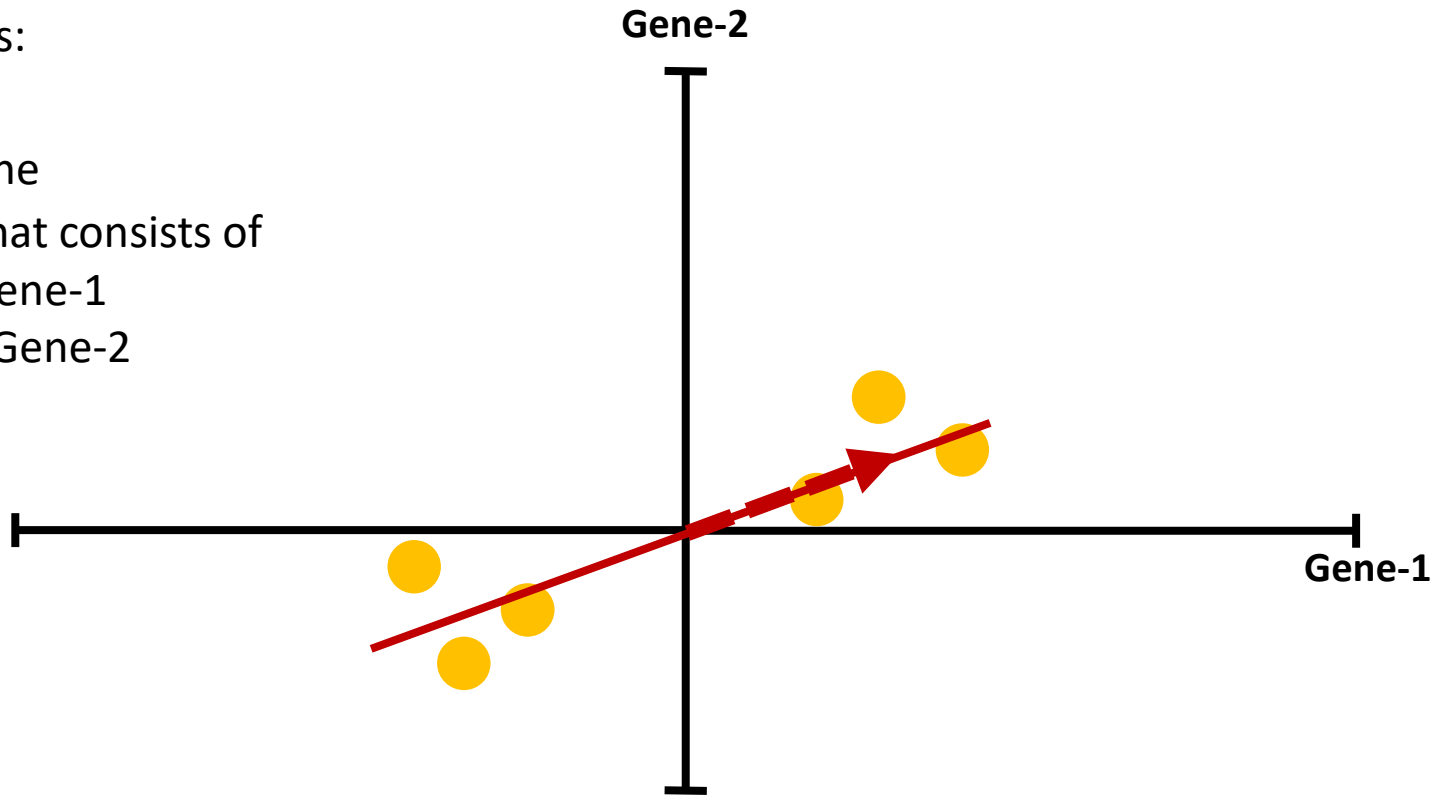
- **4 x Gene-1 : 1 x Gene-2**



Principal Component Analysis (PCA)

Consider now the steps:

- The data
- The best fitting line
- The unit vector that consists of
  - 0.97 parts Gene-1
  - 0.242 parts Gene-2

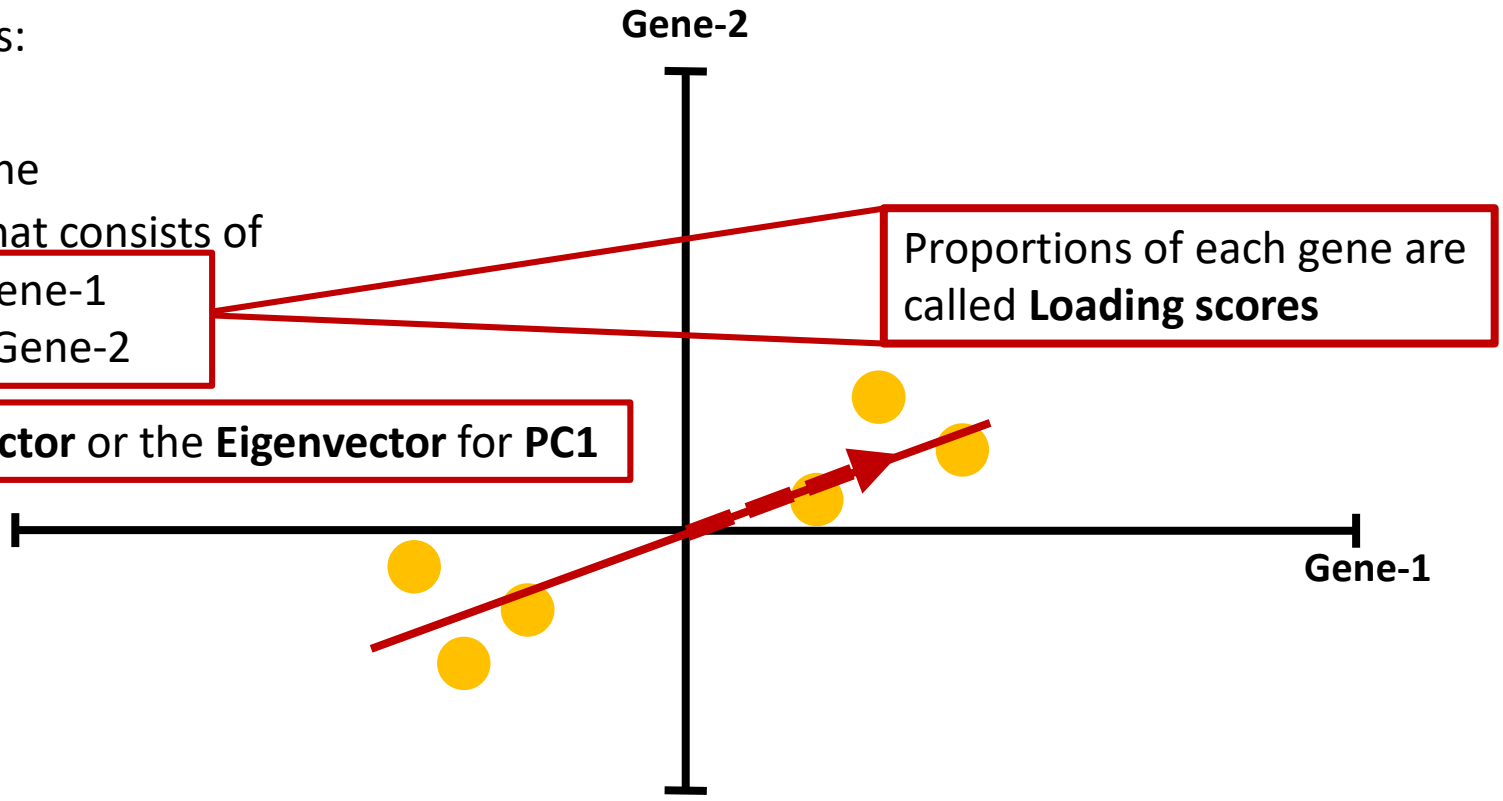


Principal Component Analysis (PCA)

Consider now the steps:

- The data
- The best fitting line
- The unit vector that consists of
  - 0.97 parts Gene-1
  - 0.242 parts Gene-2

➢ Called as **Singular Vector** or the **Eigenvector** for **PC1**



Proportions of each gene are called **Loading scores**

Principal Component Analysis (PCA)

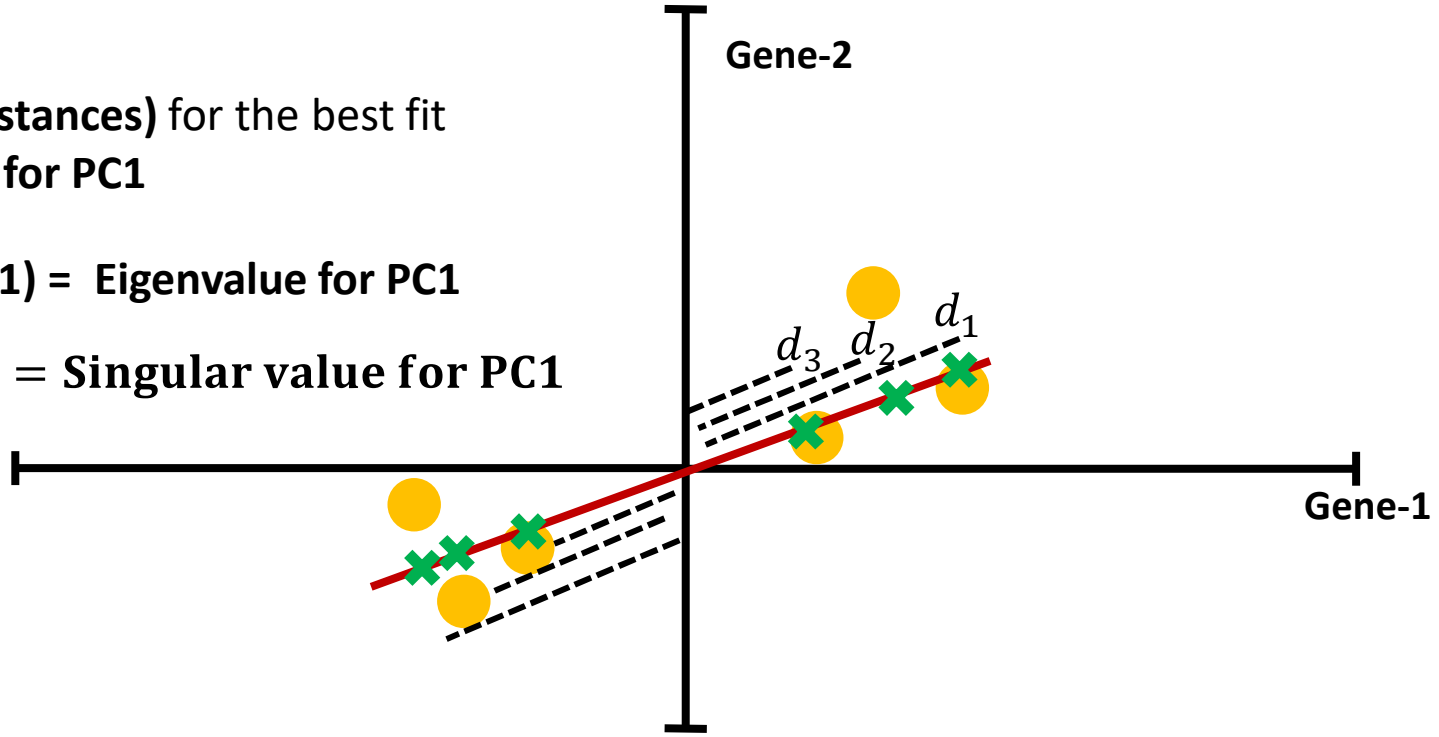
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

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

PCA also calls **SS(Distances)** for the best fit line the **Eigenvalue for PC1**

SS(Distances for PC1) = Eigenvalue for PC1

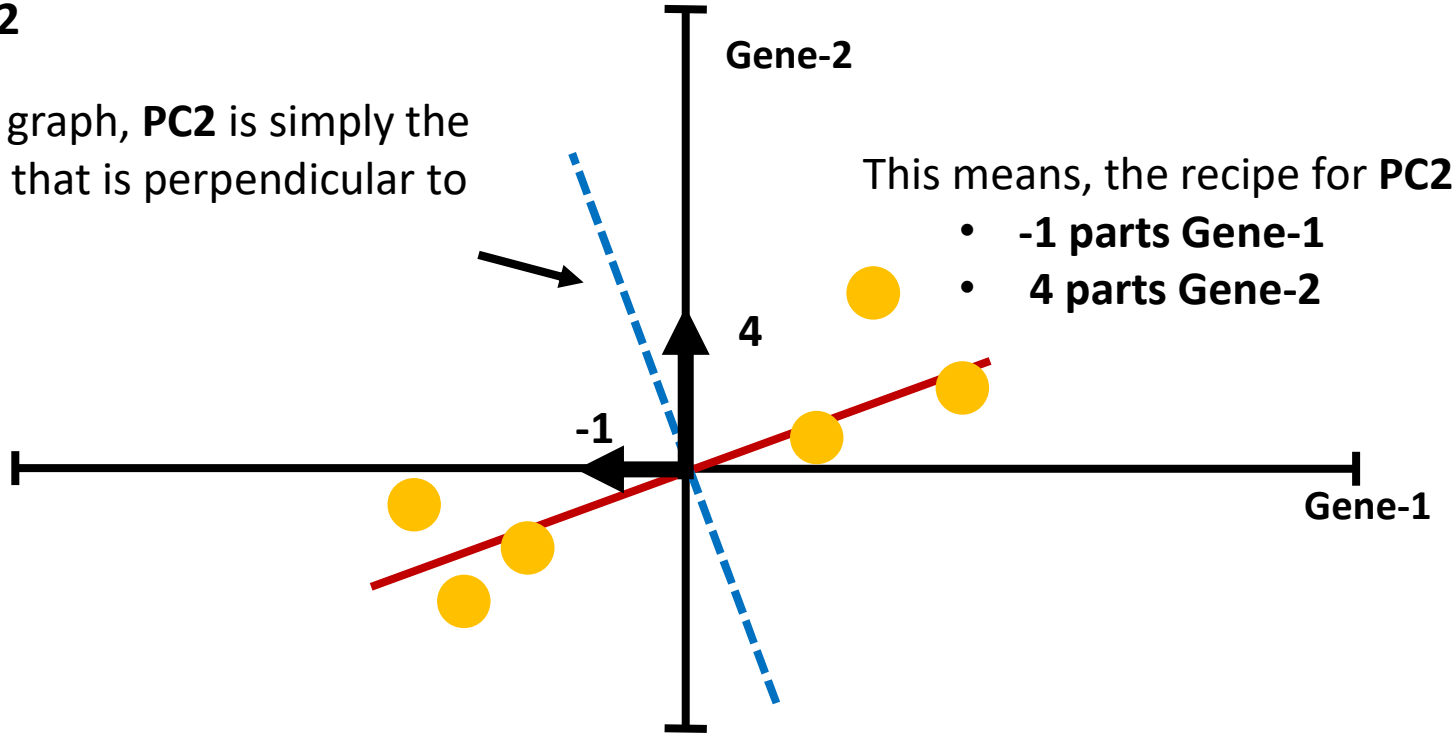
$$\sqrt{\textbf{Eigenvalue for PC1}} = \textbf{Singular value for PC1}$$



Principal Component Analysis (PCA)

Now, let's work on **PC2**

Since this is only a 2-D graph, **PC2** is simply the line through the origin that is perpendicular to **PC1**.



This means, the recipe for **PC2** is:

- **-1 parts Gene-1**
- **4 parts Gene-2**

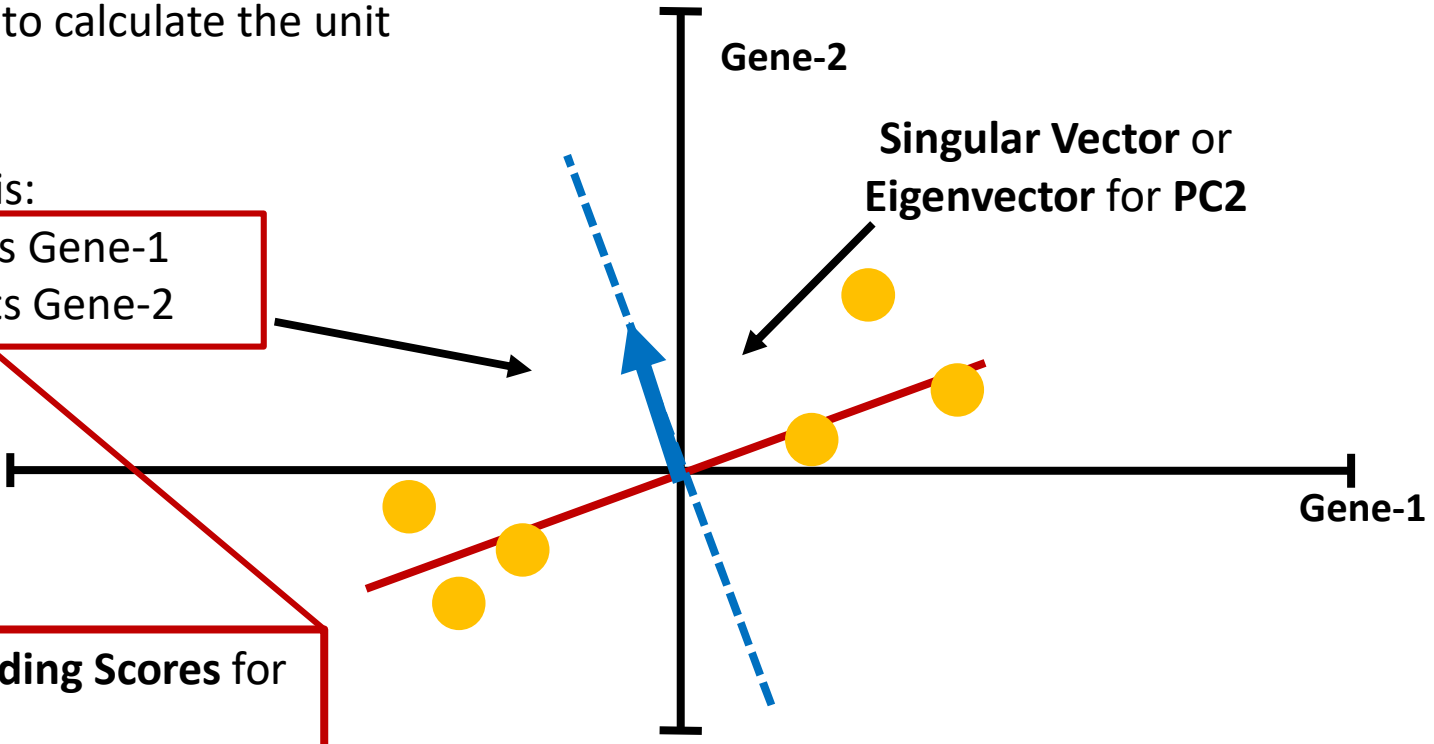
Principal Component Analysis (PCA)

If we scale everything to calculate the unit vector:

The recipe for **PC2** is:

- **-0.242** parts Gene-1
- **0.97** parts Gene-2

They are the **Loading Scores** for **PC2**

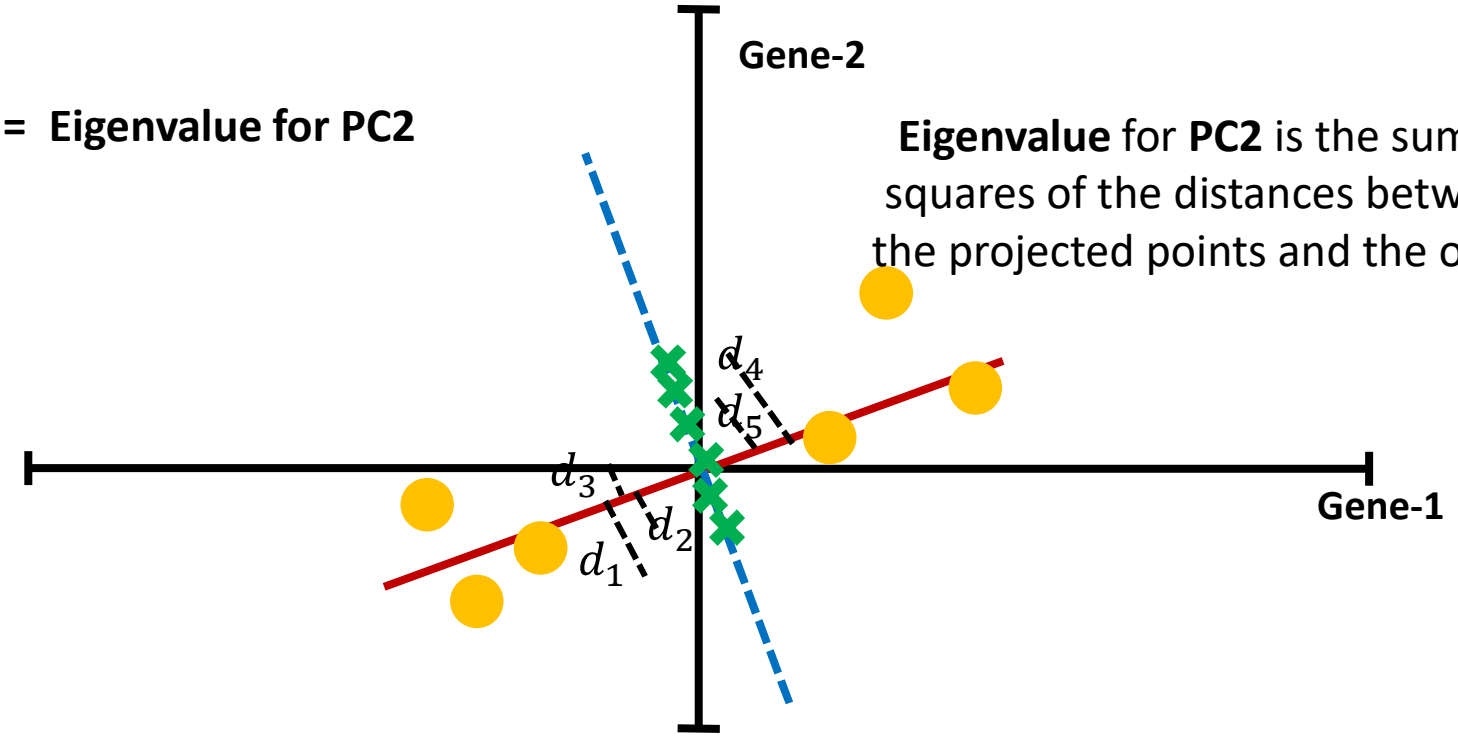


Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

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

SS(Distances for PC2) = Eigenvalue for PC2



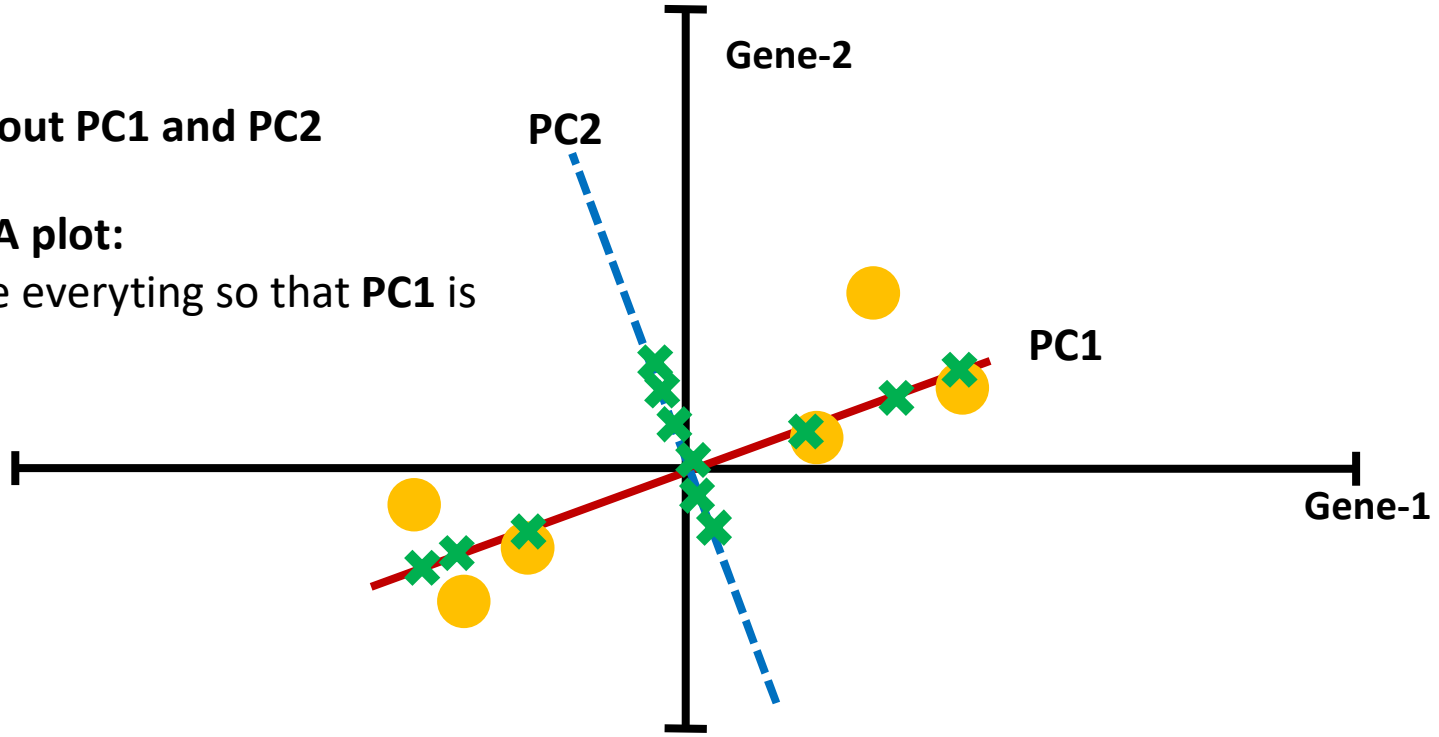
Eigenvalue for **PC2** is the sum of squares of the distances between the projected points and the origin

Principal Component Analysis (PCA)

Now, that we worked out PC1 and PC2

To draw the final PCA plot:

- Simply rotate everyting so that **PC1** is horizontal

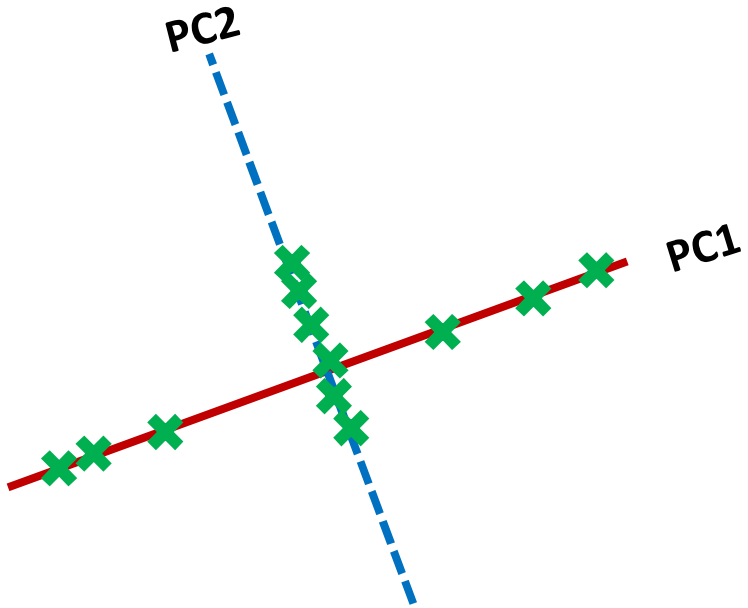


Principal Component Analysis (PCA)

Now, that we worked out PC1 and PC2

To draw the final PCA plot:

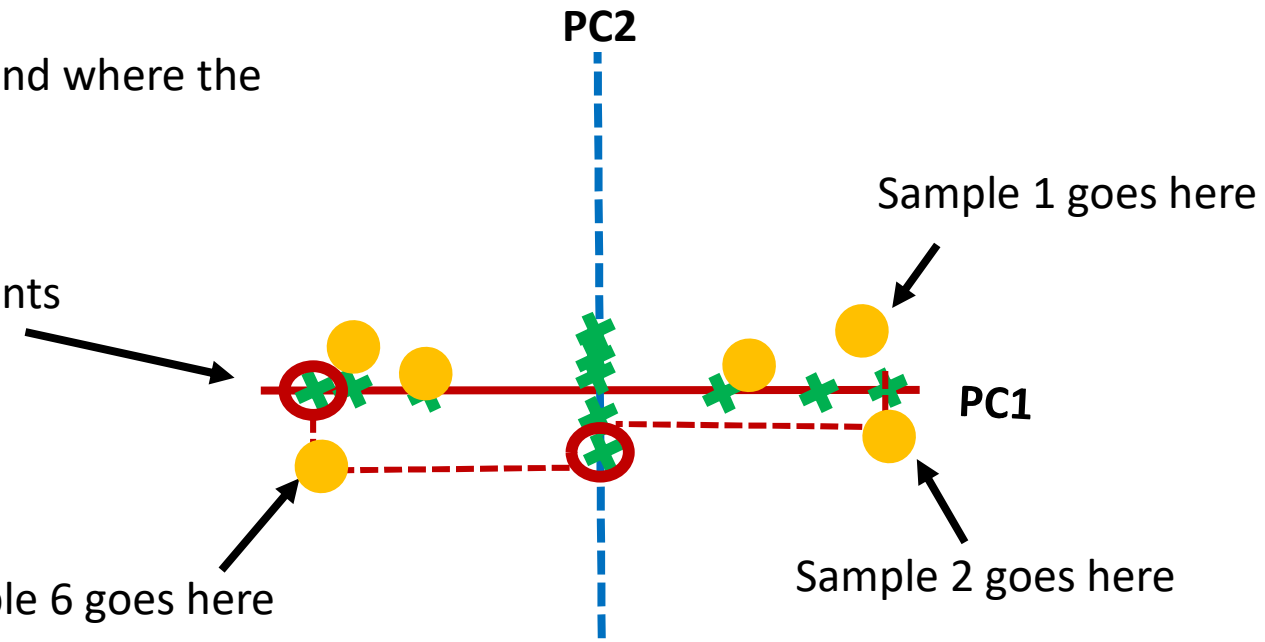
- Simply rotate everyting so that **PC1** is horizontal



Principal Component Analysis (PCA)

Now, use the projected points to find where the samples go in the PCA plot

Assume that, these projected points correspond to Sample 6

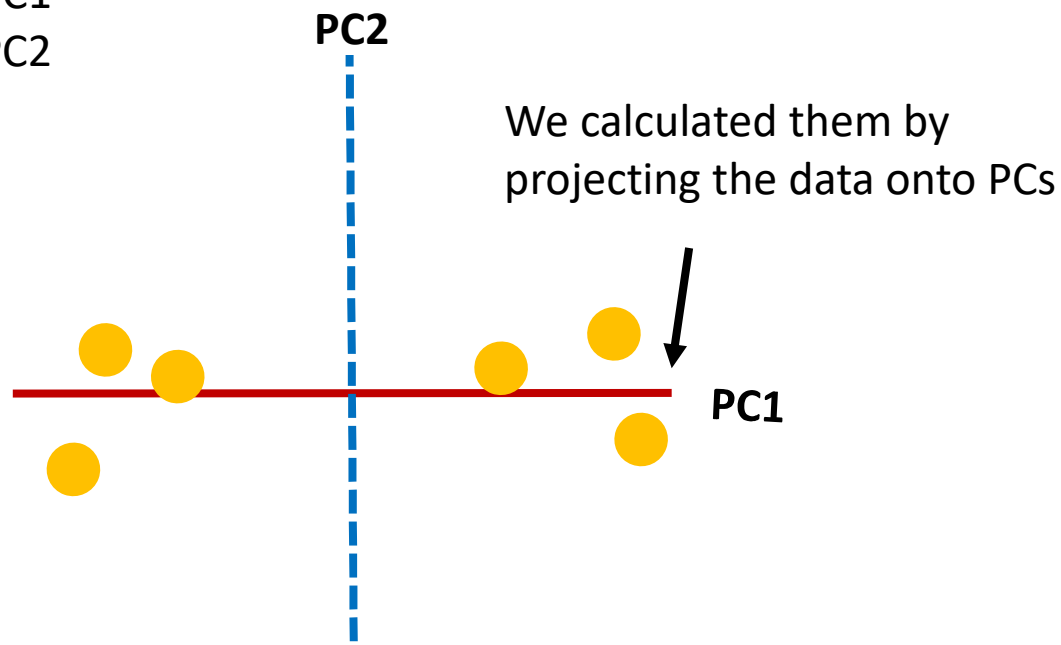


That is how PCA is done using Singular Value Decomposition (SVD)

Principal Component Analysis (PCA)

Remember the eigenvalues

- SS(Distances for PC1) = Eigenvalue for PC1
- SS(Distances for PC2) = Eigenvalue for PC2

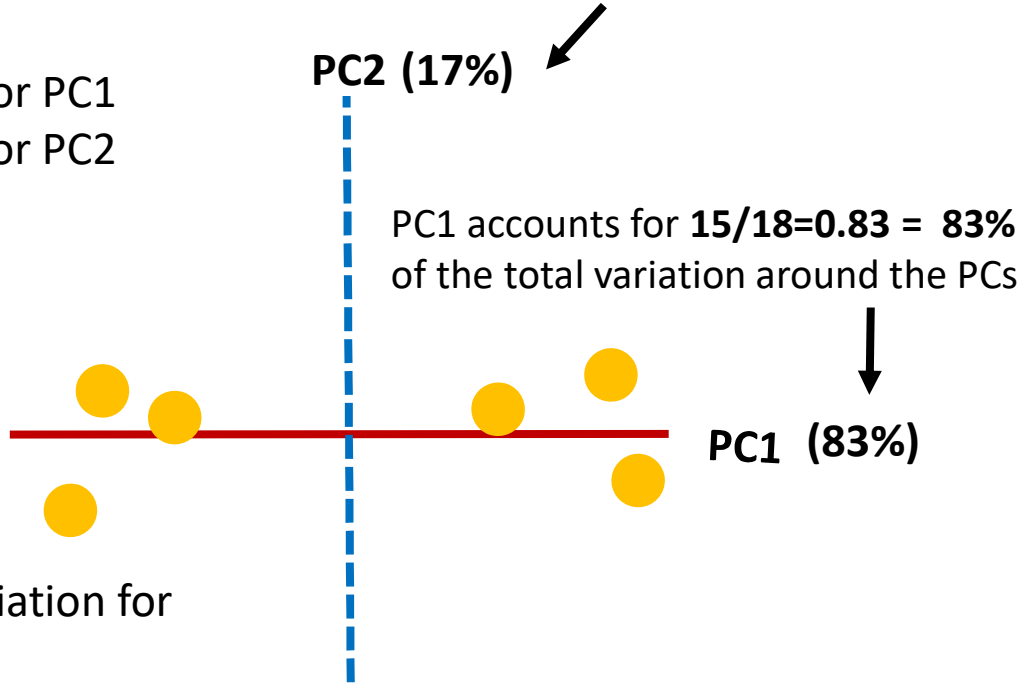


Principal Component Analysis (PCA)

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

- SS(Distances for PC1)/(n-1) = Variation for PC1
- SS(Distances for PC2)/(n-1) = Variation for PC2

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



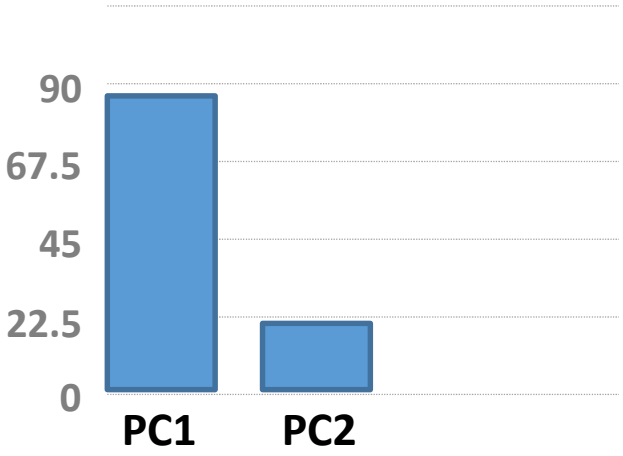
For the sake of simplicity, imagine that the variation for **PC1=15** and the variation for **PC2=3**

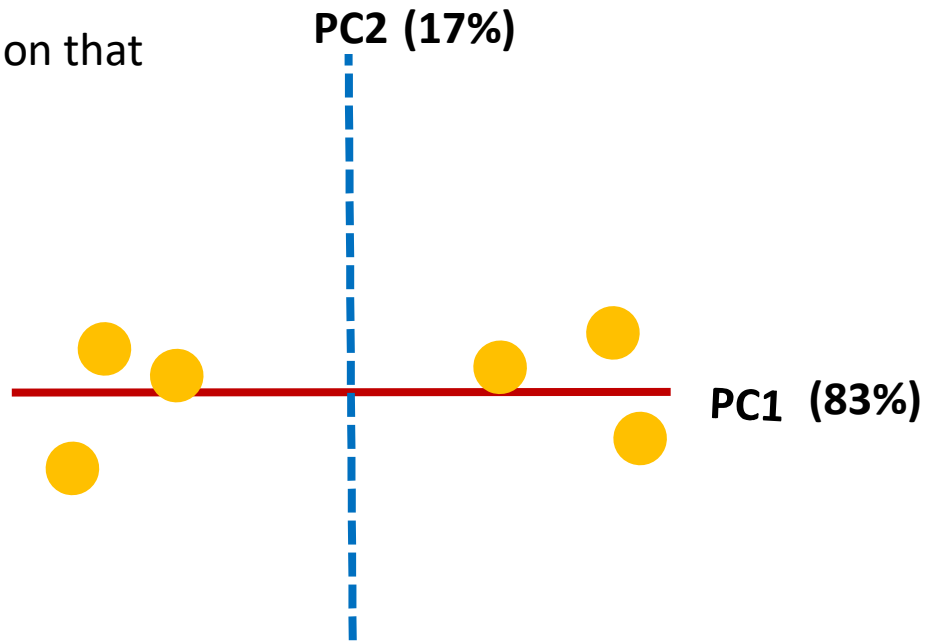
- Total variation around both PCs is **15 + 3=18**

Principal Component Analysis (PCA)

Graphical representation:

- **Scree Plot**
  - Representation of percentages of variation that each PC accounts for





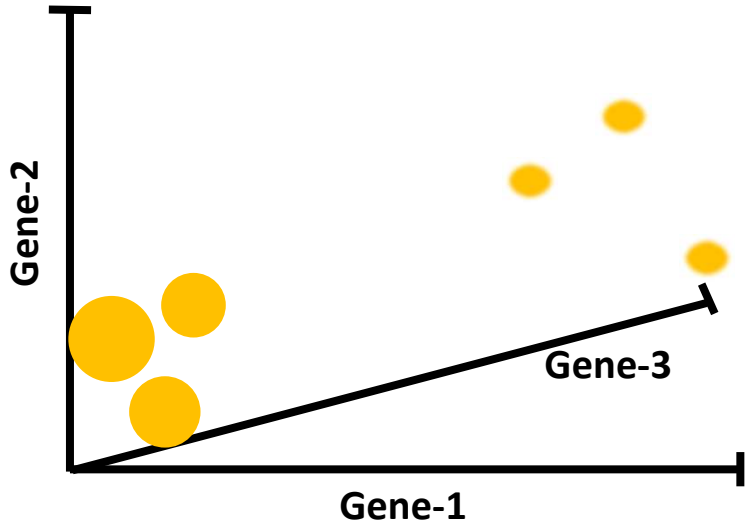
https://www.datascienceplus.com

Principal Component Analysis (PCA)

More complicated example:

- PCA with 3 variables is pretty much the same as 2 variables

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4

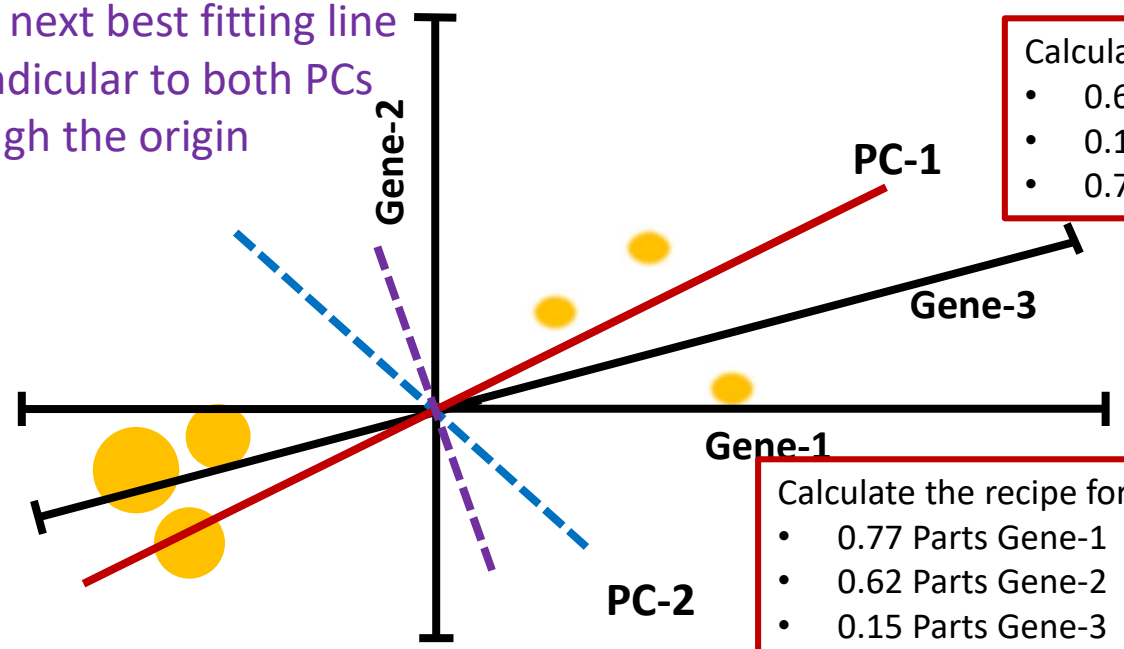


https://www.datascienceplus.com

Principal Component Analysis (PCA)

More complicated example:

- PCA with 3 variables is pretty much the same as 2 variables
  - **Step 1:** Center the data
  - **Step 2:** Find the best fitting line that goes through the origin
  - **Step 3:** Find the next best fitting line (satisfy all previous requirements)
  - **Step 4:** Find the next best fitting line
    - It is perpendicular to both PCs
    - Goes through the origin



Calculate the recipe for **PC-1:**
- 0.62 Parts Gene-1
- 0.15 Parts Gene-2
- 0.77 Parts Gene-3

Calculate the recipe for **PC-2:**
- 0.77 Parts Gene-1
- 0.62 Parts Gene-2
- 0.15 Parts Gene-3

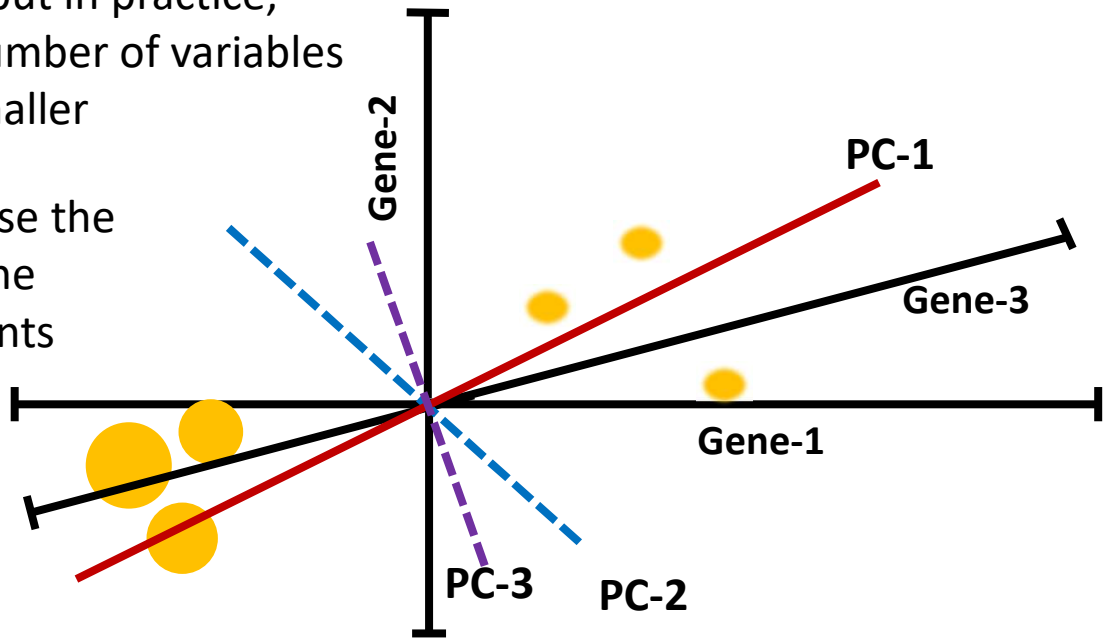
Principal Component Analysis (PCA)

More complicated example:

If we had more genes, we'd just keep on finding more and more principle components by adding perpendicular lines and rotating them

In theory there is one per gene (variable), but in practice, the number of PCs is either equal to the number of variables or the number of samples, whichever is smaller

After the identification of all PCs, we can use the eigenvalues (SS(distances)) to determine the proportion of variation that each PC accounts for…



https://www.datascienceplus.com

Principal Component Analysis (PCA)

More complicated example:

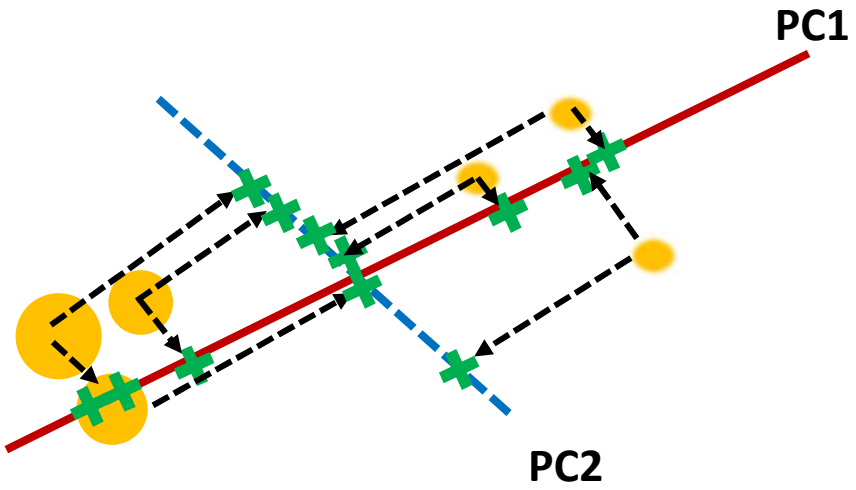
Assume that

- PC1 accounts for 79% of the variation
- PC2 accounts for 15% of the variation

To convert the 3-D graph into a 2-D PCA graph, we just strip away everything but the data and PC1 and PC2

Project the samples onto PC1

Project the samples also onto PC2



https://www.datascienceplus.com

Principal Component Analysis (PCA)

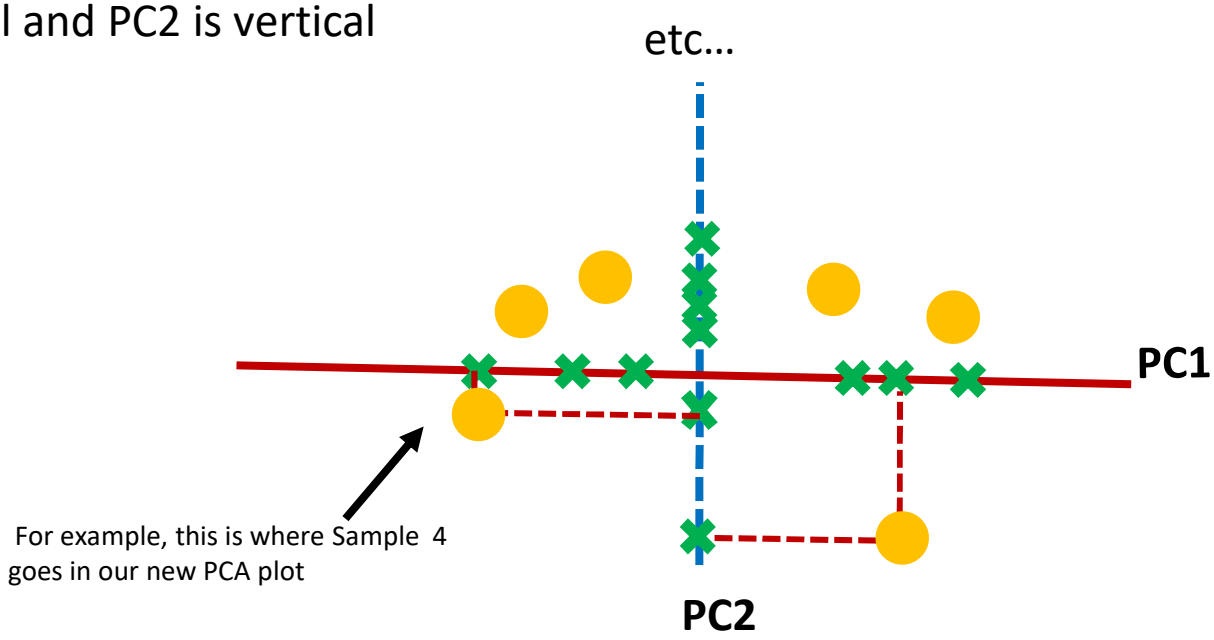
More complicated example:

Assume that

- PC1 accounts for 79% of the variation
- PC2 accounts for 15% of the variation

Then we rotate so that PC1 is horizontal and PC2 is vertical

- It is easier to look at ☺



https://www.datascienceplus.com

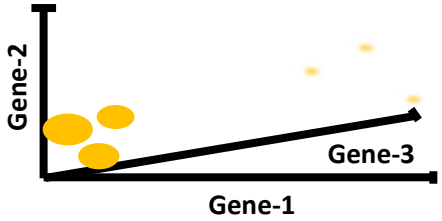
Principal Component Analysis (PCA)

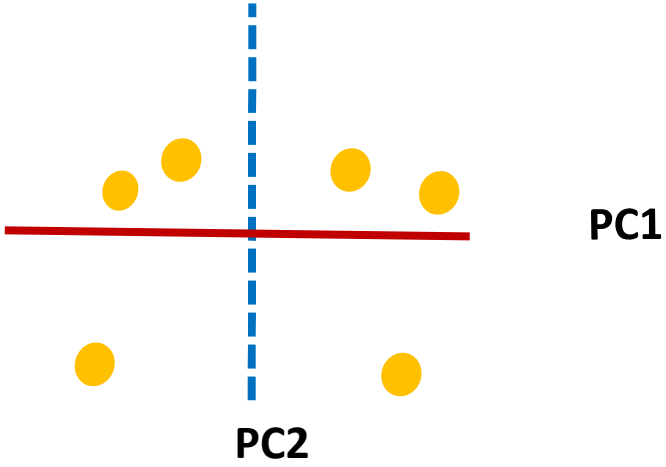
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

More complicated example:

To summarize

- We started with an 3-D graph that was kind of hard to read
- We calculated the principle components
- We determined with the eigenvalues the very informative components (**Scree Plot)**
- **Lastly, we used the most informative PCs to draw a 2-D graph**





https://www.datascienceplus.com

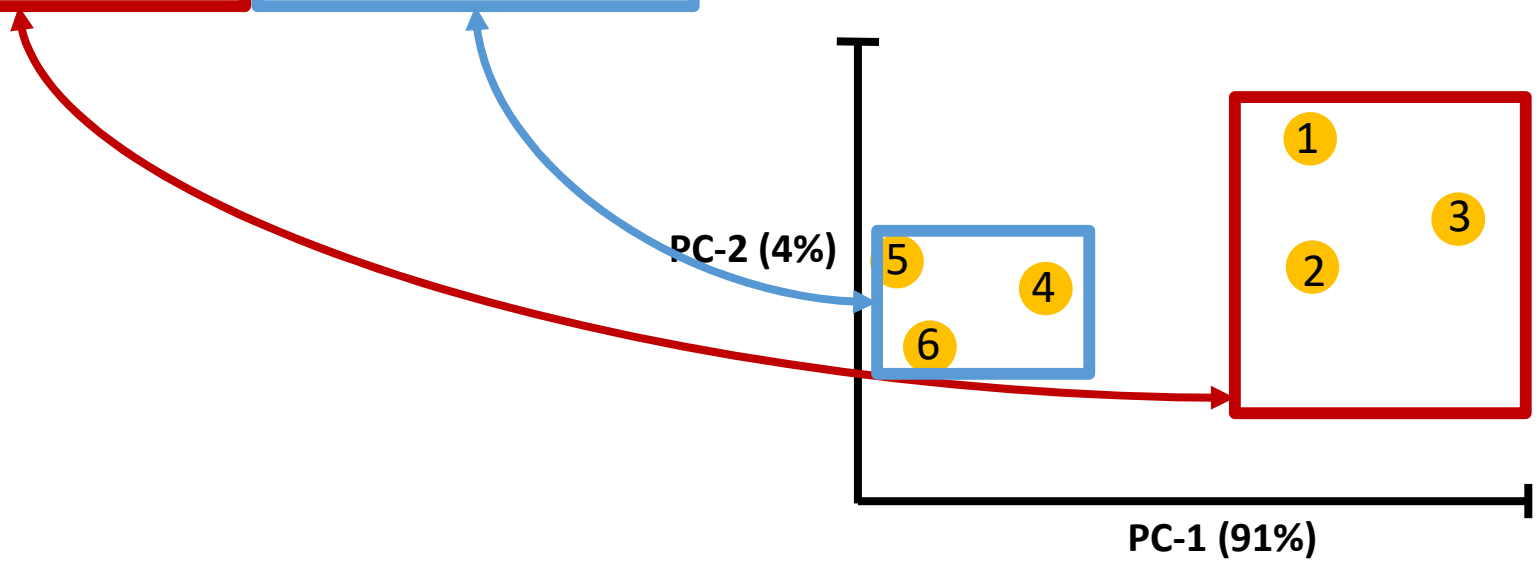
Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Simple data set

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

Similar animals cluster together…



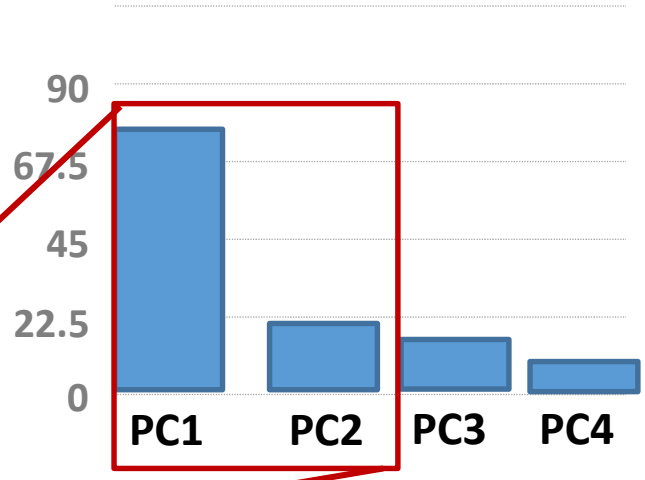
Principal Component Analysis (PCA)

More complicated example:

- Consider 4 genes per animal
- It is not possible to draw the 4-D graph

	Animal-1	Animal-2	Animal-3	Animal-4	Animal-5	Animal-6
Gene-1	20	22	16	6	4	2
Gene-2	12	8	10	6	5.6	2
Gene-3	24	18	20	5	2.6	4
Gene-4	10	14	12	4	8	14

- In this case, we use the scree plot
  - it doesn‘t care if we can draw a graph or not
- **Final step:** Select PCs with highest account for the variation und use them to draw a 2-D PCA graph



https://www.datascienceplus.com

Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Example:

Import the data file called **"plantData.csv"** into the variable **mydata**

> mydata= read.table("plantData.csv", header=TRUE, sep="," , stringsAsFactors = TRUE)

Process the dataset

- Divide **mydata** in two data frames as **df1** and **df2**
  - **df1** contains the class attribute "Species"
  - **df2** contains the remaining information
- Apply **principal component analysis** (function in R: **prcomp**)
- Analyze the results of PCA
  - Draw a 2-D plot of the elements using the first two principal components
  - Calculate how much variation each principal component accounts for and plot it (Scree Plot)
  - Determine which attributes have the largest effect on the first two principal components using the loading scores

Principal Component Analysis (PCA)

Example:

Process the dataset

- Divide **mydata** in two data frames as **df1** and **df2**

```
> df1=mydata[, "Species"]
> df2=mydata[, c(1,2,3,4)]
```

- Apply **principal component analysis** (?prcomp for help)

```
> pca = prcomp(df2, scale. = TRUE) #apply pca
#Note: By default, prcomp() expects the samples to be rows and the attributes to be columns
#If this is not the case, transpose the data frame using t()
```

- Analyze the results of PCA

Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Example:

Process the dataset

- Analyze the results of PCA
  - Draw a 2-D plot of the elements using the first two principal components

```
>plot(pca$x[,1], pca$x[,2])
#ggplot2 requires the data in a data frame
>pca.data = data.frame(Species = df1, X = pca$x[,1], Y = pca$x[,2])
>ggplot(data = pca.data, aes(x = X, y = Y, color = Species))  + geom_point() +
+                labs(x = "PC1", y = "PC2")
```

Principal Component Analysis (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Example:

Process the dataset

- Analyze the results of PCA
  - Calculate how much variation each principal component accounts for and plot it (Scree Plot)

```
>pca.var = pca$sdev^2
#Transform absolute values to percentage
>pca.var.per = round(pca.var / sum(pca.var)*100,1)
>barplot(pca.var.per, xlab = "Principal Component", ylab = "Percent Variation")
>ggplot(data = data.frame(Percentage = pca.var.per),
+              aes(x = 1:length(Percentage), y = Percentage)) + geom_col() +
+              labs(x = "Principal Component", y = "Percent Variation")
```

Principal Component Analysis (PCA)

Example:

Process the dataset

- Analyze the results of PCA
  - Determine which attributes have the largest effect on the first two principal components using the loading scores

```
> pc1_loading_scores = pca$rotation[,1]
#Transform to absolute values to sort according to effect size
>pc1_attribute_scores = abs(pc1_loading_scores)
>pc1_attribute_scores_ranked = sort(pc1_attribute_scores, decreasing = TRUE)

>pc2_loading_scores = pca$rotation[,2]
#Transform to absolute values to sort according to effect size
>pc2_attribute_scores = abs(pc2_loading_scores)
>pc2_attribute_scores_ranked = sort(pc2_attribute_scores, decreasing = TRUE)

>pc1_attribute_scores_ranked
>pc2_attribute_scores_ranked
```

Clustering Exercises (PCA)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exercise:

Import the data file called **"ArtificialGen.csv"** into the variable **mydata**

Process the dataset

- Inspect the data and check for missing values
- Divide **mydata** in two data frames as **df1** and **df2**
  - **df1** contains the attribute "AnimalClass"
  - **df2** contains the remaining information
- Apply **principal component analysis**
- Analyze the results of PCA
  - Draw a 2-D plot of the elements using the first two principal components
  - Calculate how much percentage of variation each principal component accounts for and plot it
  - Determine which variables have the largest effect on the first two principal components using the loading scores