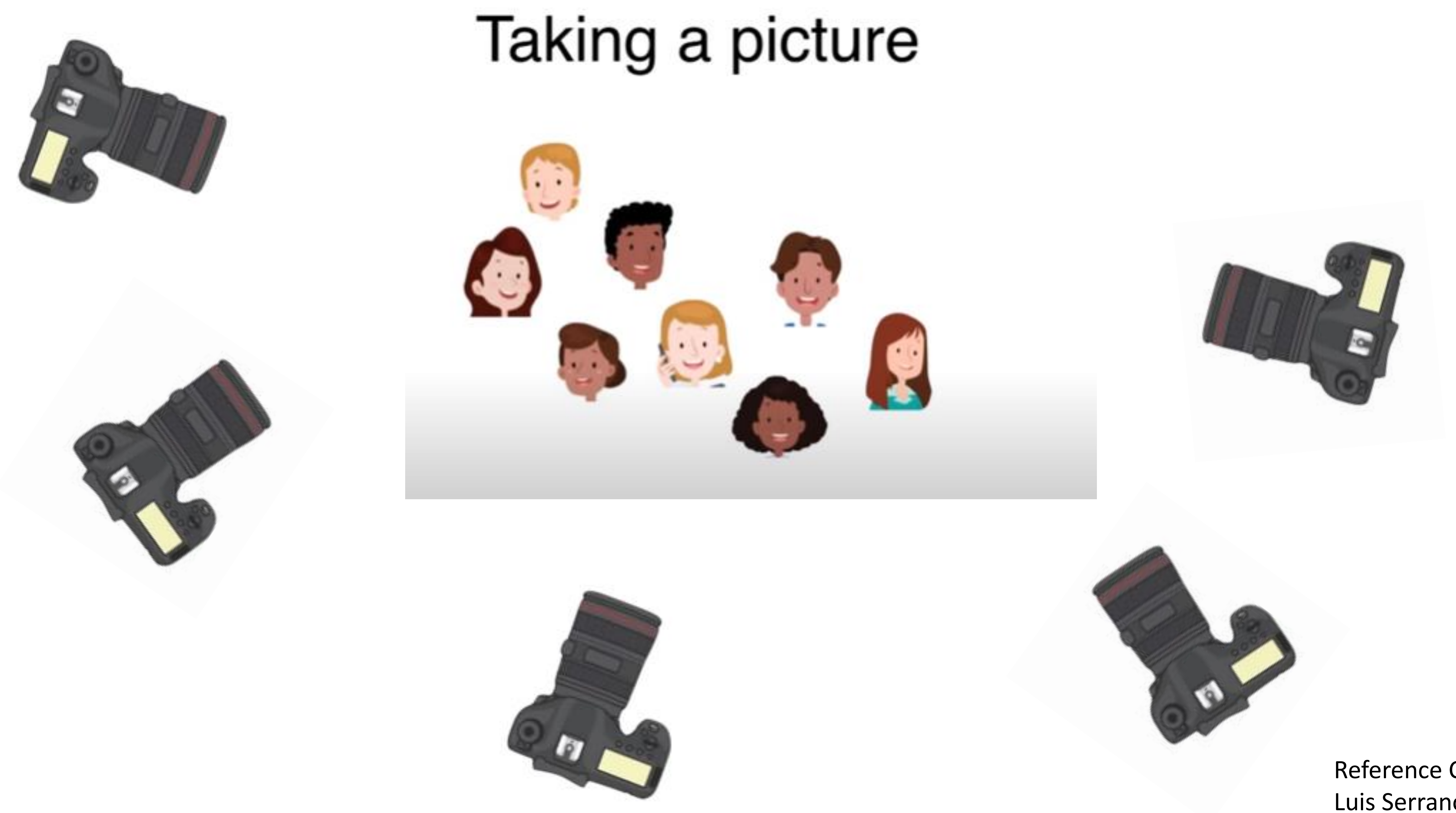
INNOMATICS RESEARCH LABS

Dimensionality Reduction

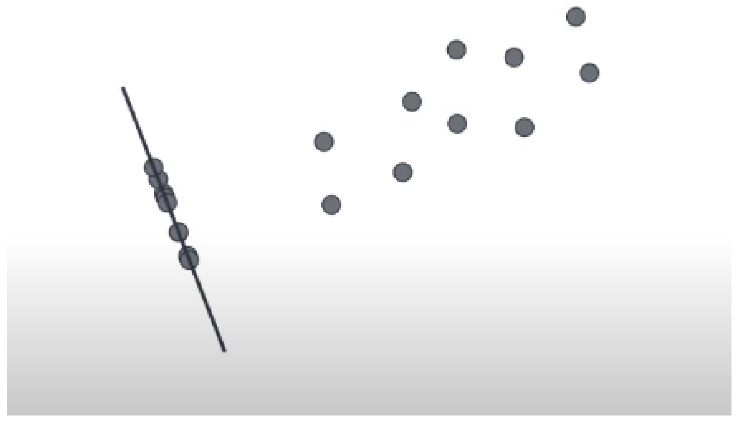
Principal Component Analysis (PCA)

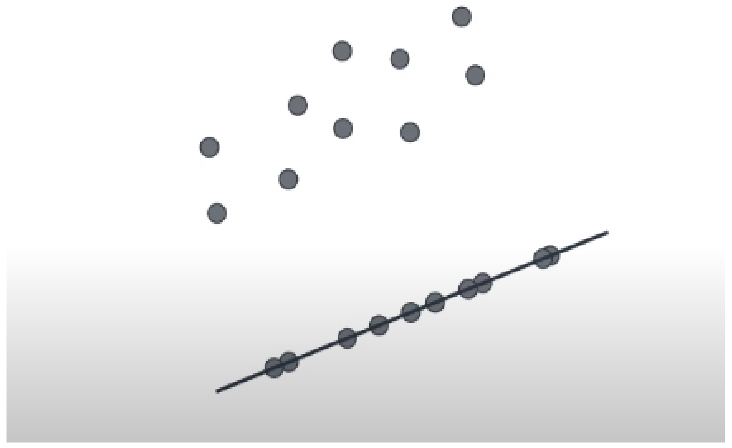
Intuition for Dimensional Reduction & PCA

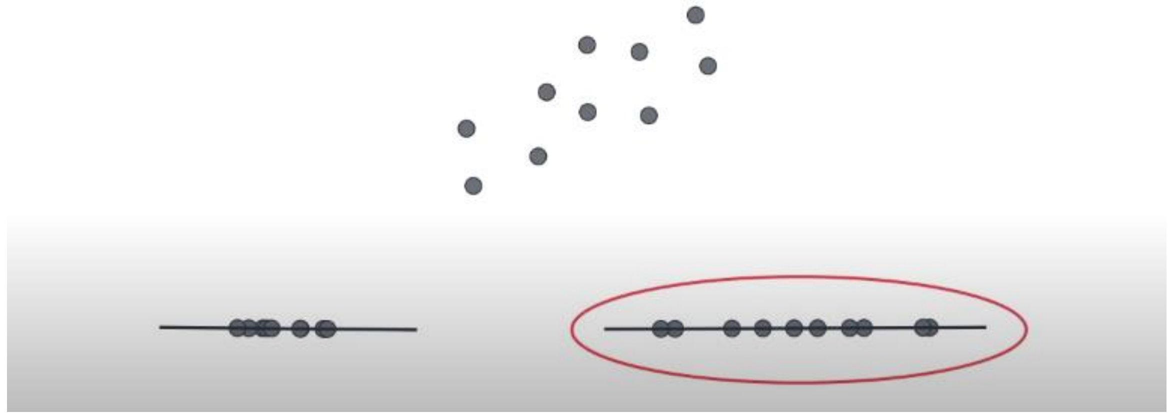


Reference Courtesy:
Luis Serrano

Dimensionality Reduction







Reference Courtesy:
Luis Serrano

Let's take an **example**

Housing data set with bunch of columns like:

1) Size
2) #Rooms
3) #Bathrooms
4) Schools Around
5) Crime Rate

So if given this dataset and let's say we want to reduce the number of columns.....how can we sort of reduce ?

1) Size feature (Size, #Rooms, #Bathrooms)
2) Location Feature (Schools around, Crime rate)

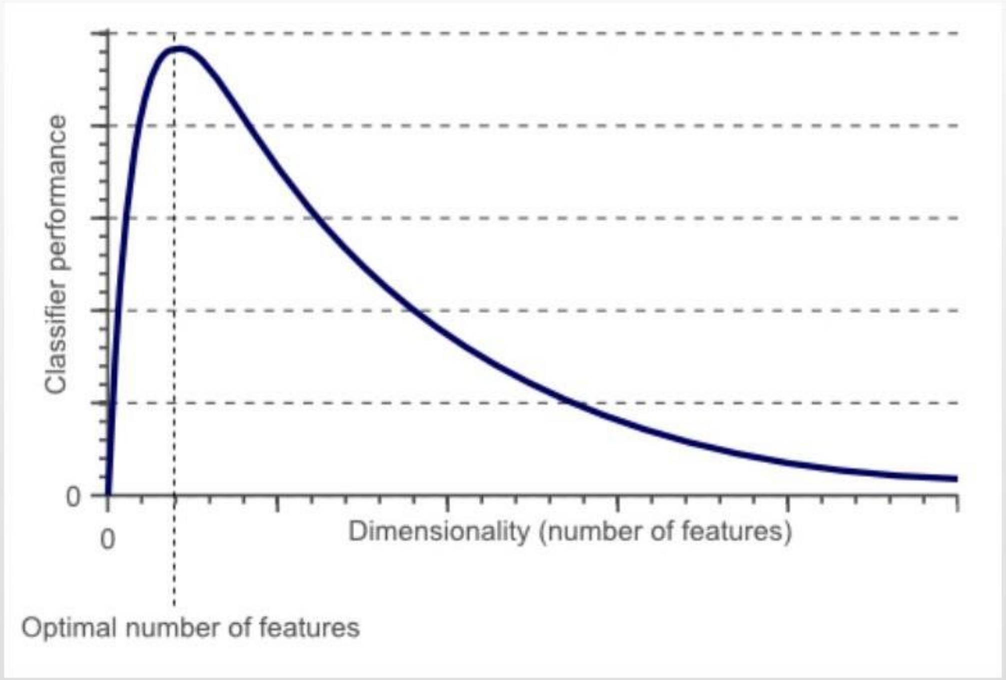
We observed how we converted the 5-D problem to a 2-D problem
Dimensional reduction

Reference Courtesy:
Luis Serrano

Dimensionality Reduction

Hughes Phenomenon

- As the number of features increases, the classifier's performance increases as well until we reach the optimal number of features.
- Adding more features based on the same size as the training set will then degrade the classifier's performance.

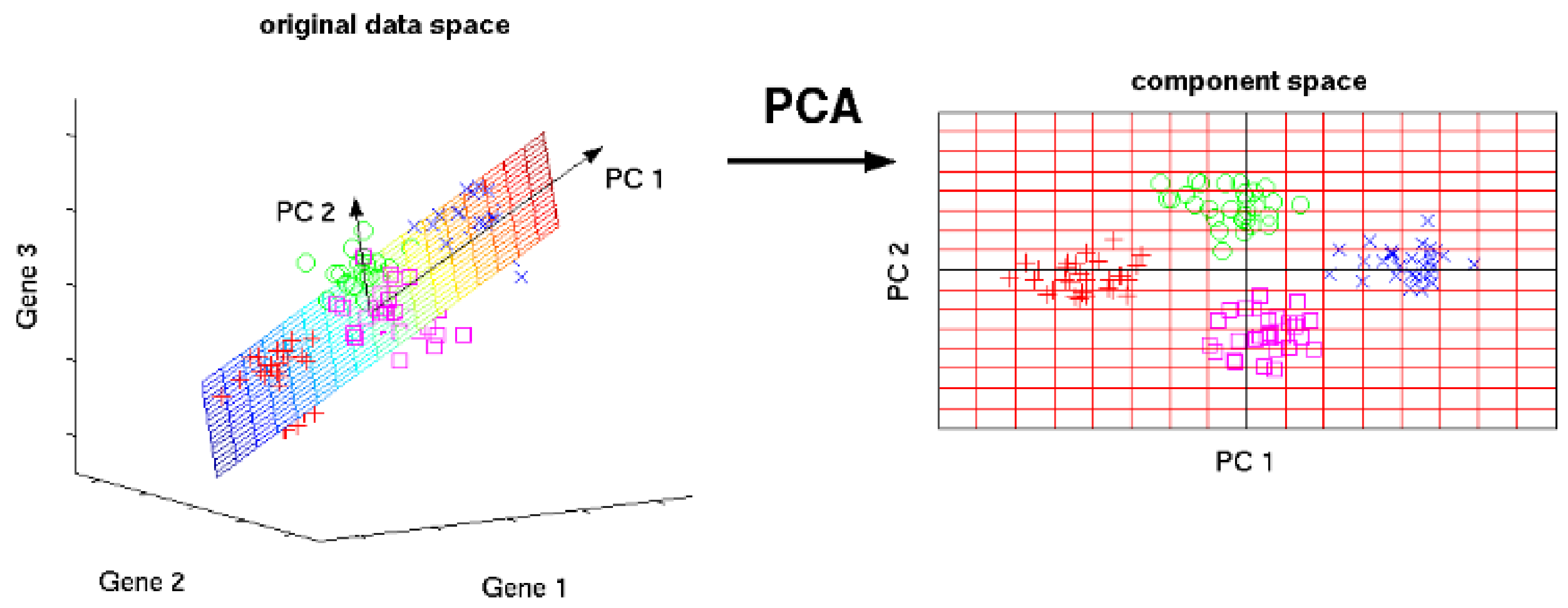


Dimensionality Reduction

- ***Dimensionality*** - in statistics refers to how many attributes a dataset has.
- Need for reduction → '***Curse of dimensionality***'.
- Curse of dimensionality refers to an exponential increase in the size of data caused by a large number of dimensions.
- As the number of dimensions of a data increases, it becomes more and more difficult to process it.
- ***Dimensionality Reduction*** is a solution - reduce the size of data by extracting relevant information and disposing rest of data as noise.

Principal Component Analysis

- principal component analysis is a method of extracting important variables (in form of components) from a large set of variables available in a data set.
- It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible. With fewer variables.
- PCA can be used for reducing dimensionality by eliminating the later principal components



When should I use PCA?

1.Do you want to reduce the number of variables, but aren’t able to identify variables to completely remove from consideration?

1.Do you want to ensure your variables are independent of one another?

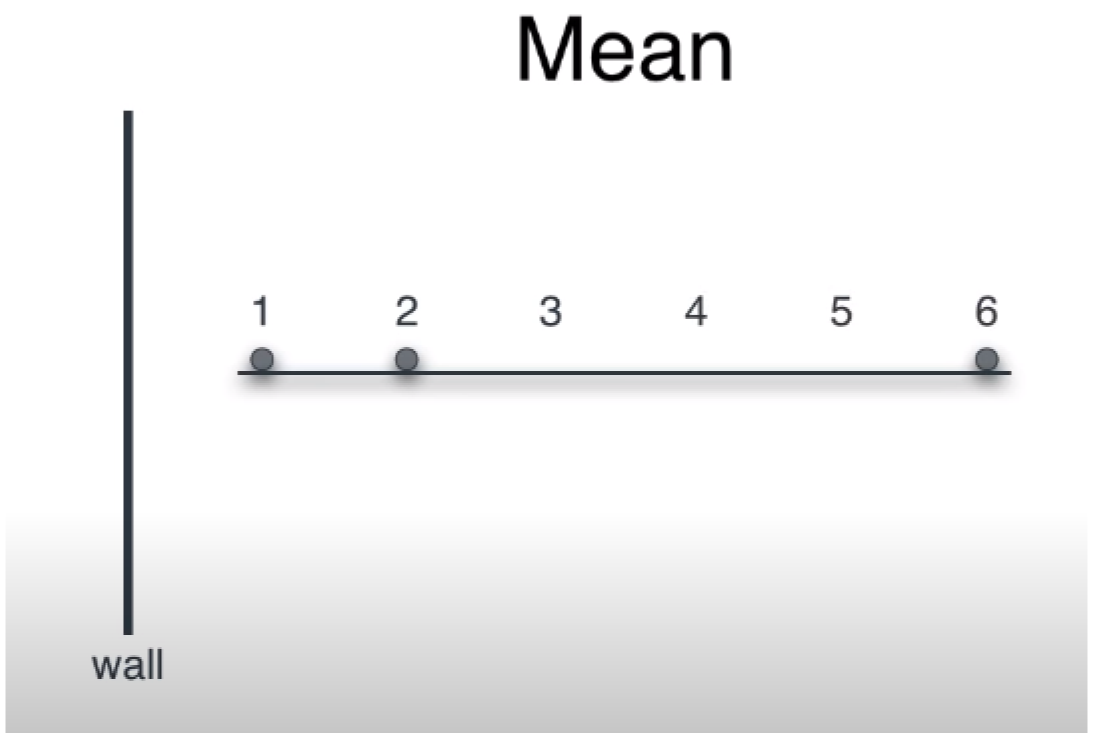
1.Are you comfortable making your independent variables less interpretable?

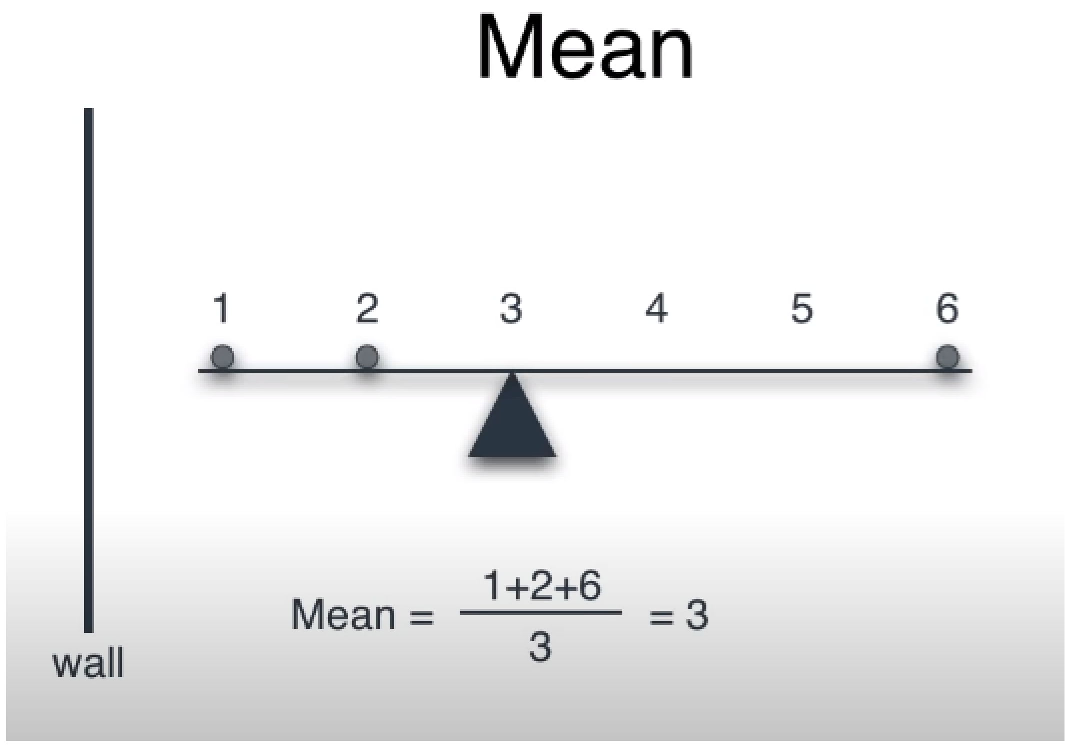
If you answered “yes” to all three questions, then PCA is a good method to use. If you answered “no” to question 3, you **should not** use PCA.

Reference Courtesy: medium.com

Mean

3 weights exactly the same. Want to balance them? How to find perfect point of balance for these 3 weights ?
Assume that bar has no weight

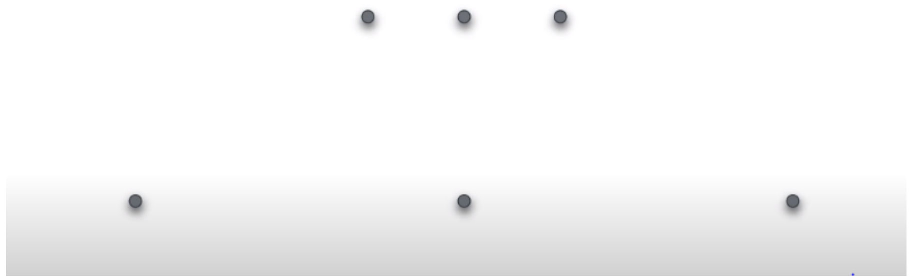


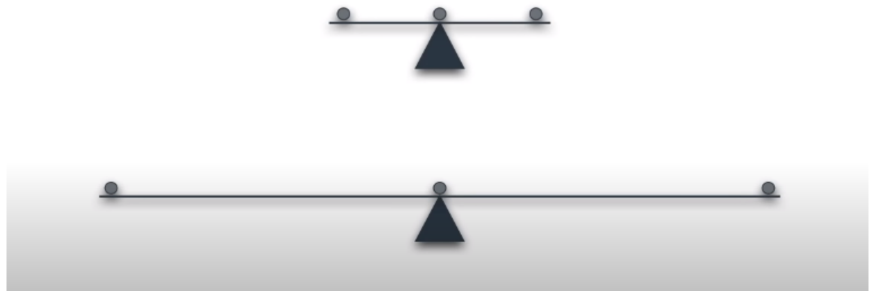


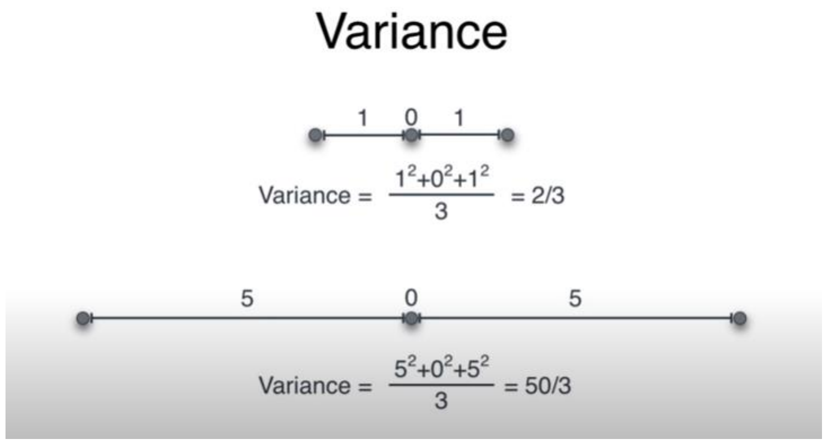
$$\text{Mean} = \frac{1+2+6}{3} = 3$$

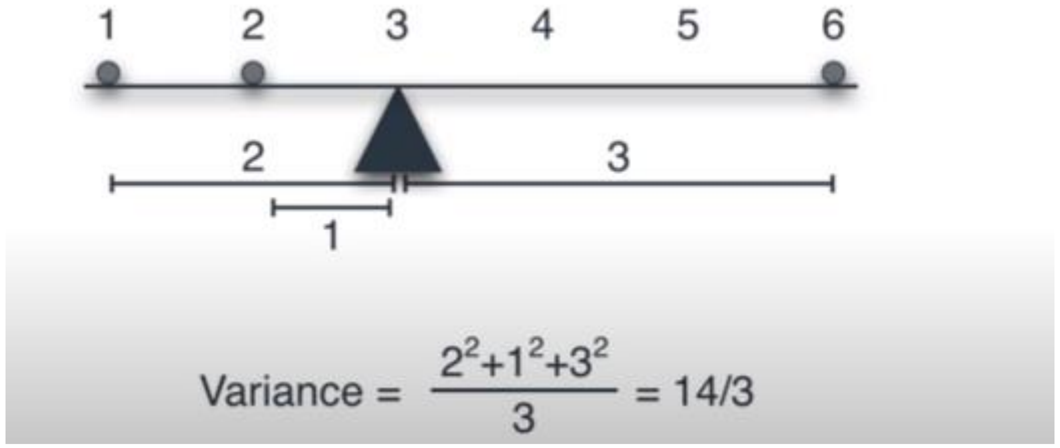
Reference Courtesy:
Luis Serrano

Variance (1-D)



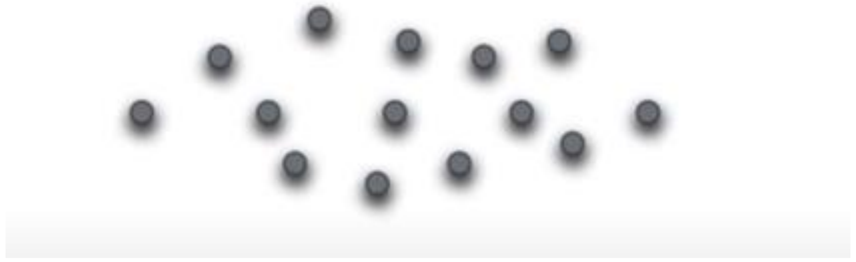


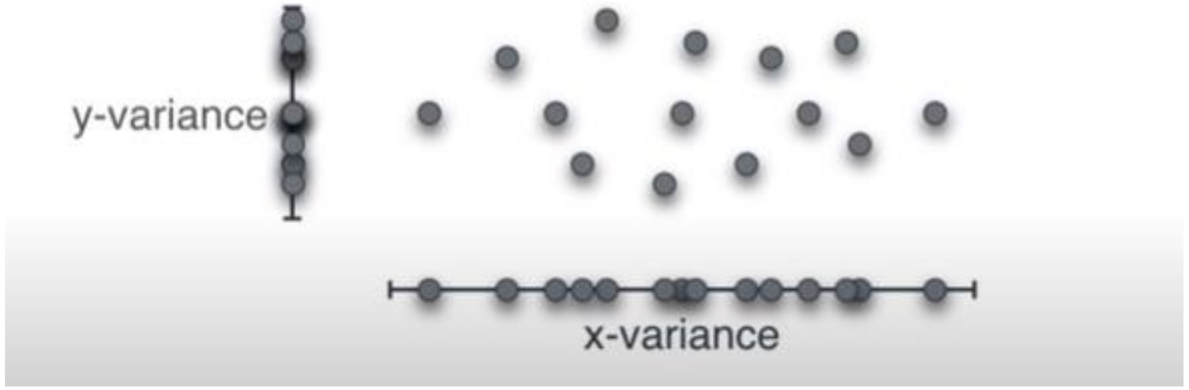

Variance
1
0
1
Variance = $\frac{1^2+0^2+1^2}{3}$ = 2/3
5
0
5
Variance = $\frac{5^2+0^2+5^2}{3}$ = 50/3



Reference Courtesy:
Luis Serrano

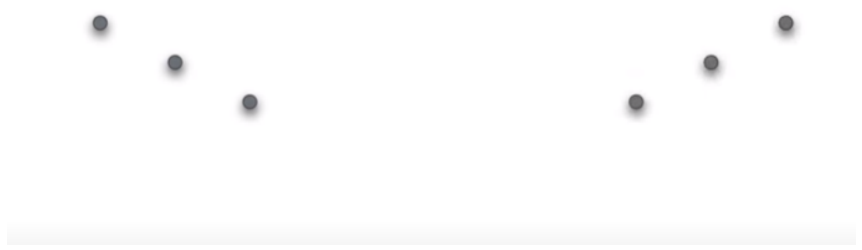
Variance (2-D)

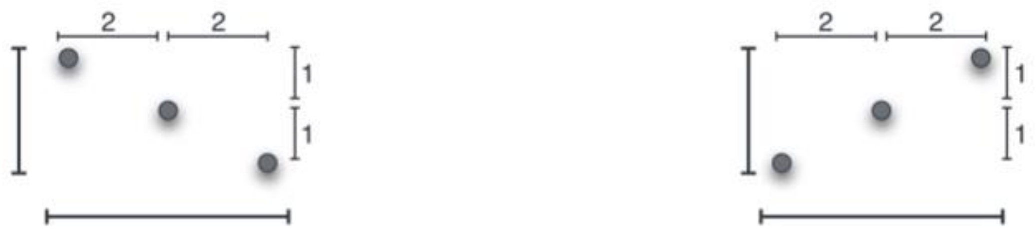




Reference Courtesy:
Luis Serrano

Intuition behind a metric ?

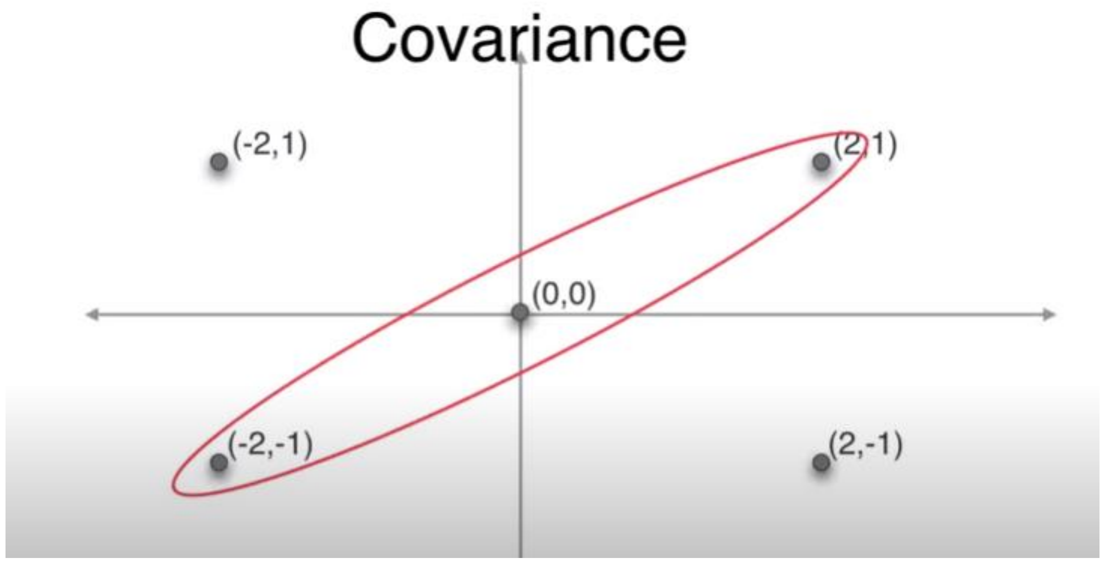


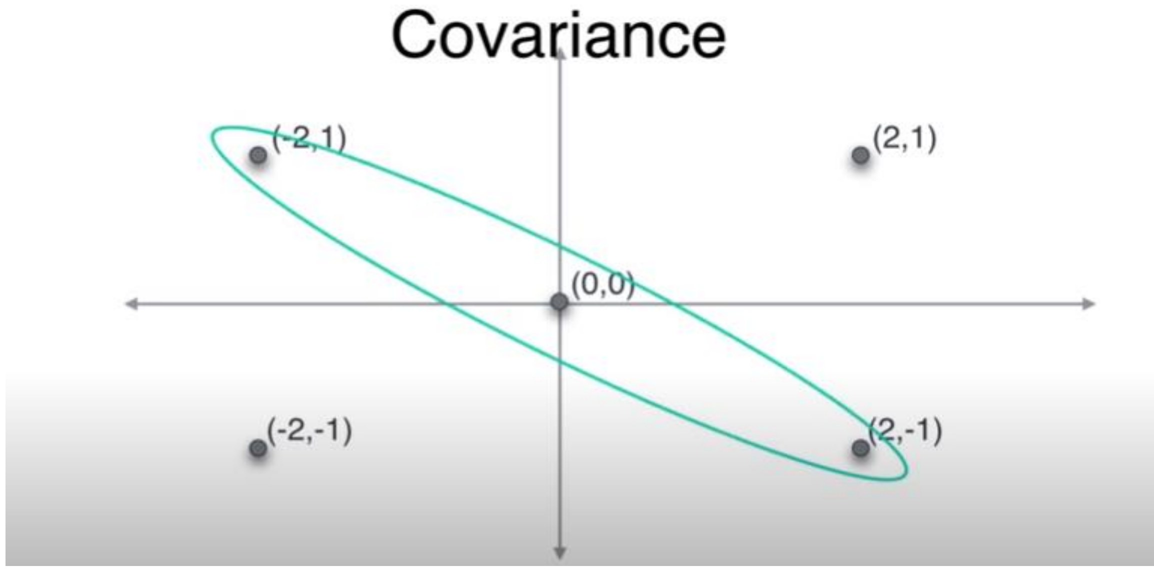


$$\text{x-variance} = \frac{2^2+0^2+2^2}{3} = 8/3$$

$$\text{y-variance} = \frac{1^2+0^2+1^2}{3} = 2/3$$

Reference Courtesy:
Luis Serrano





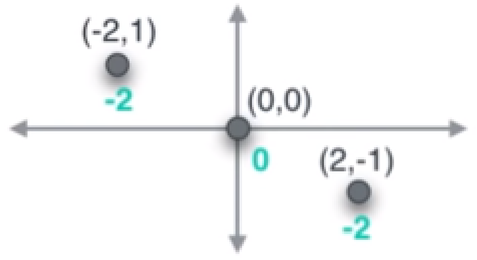
How does co-variance consider these 2 different ?

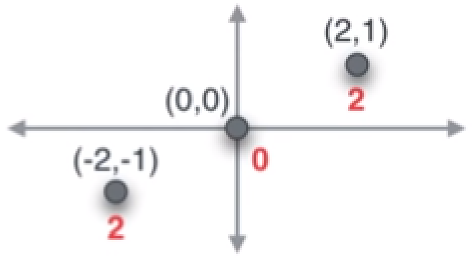
Product of coordinates

Co-Variance is the sum of product of co-ordinates

Reference Courtesy:
Luis Serrano

Covariance

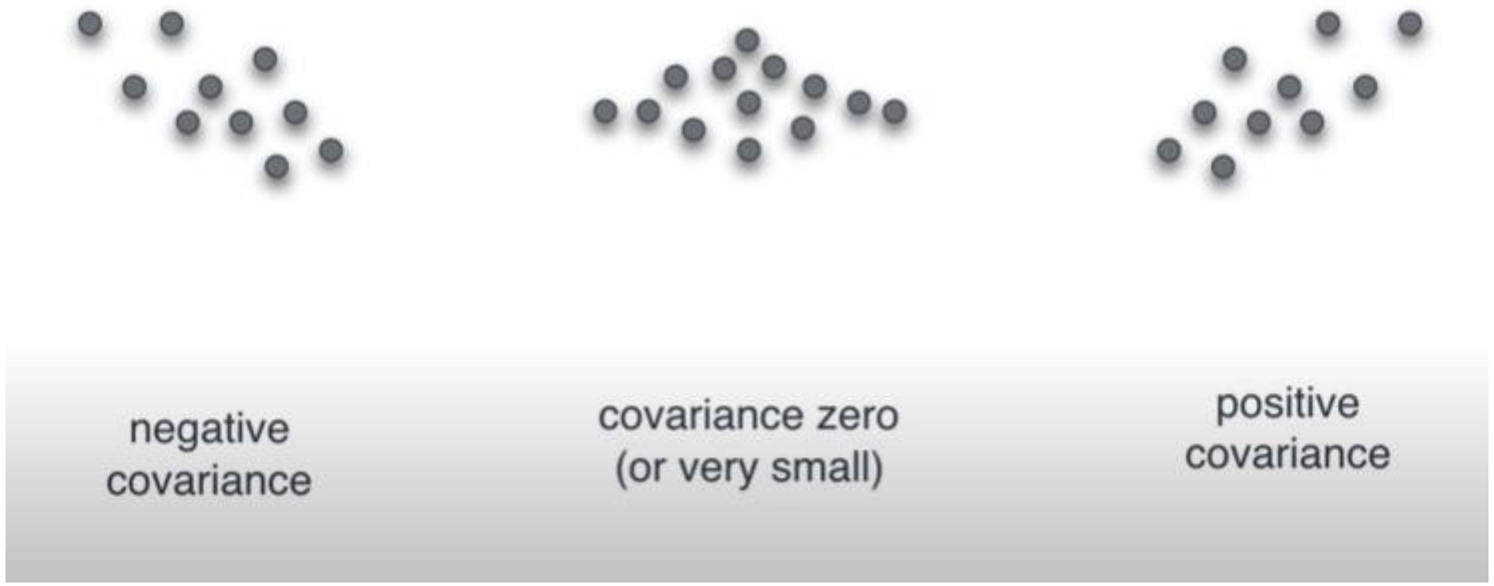




$$\text{covariance} = \frac{(-2) + 0 + (-2)}{3} = -4/3$$

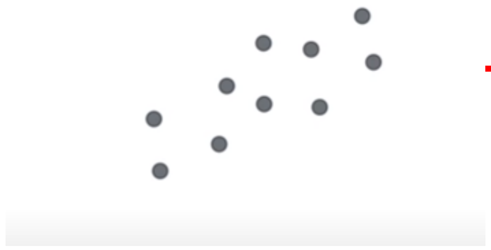
$$\text{covariance} = \frac{2 + 0 + 2}{3} = 4/3$$

Reference Courtesy:
Luis Serrano

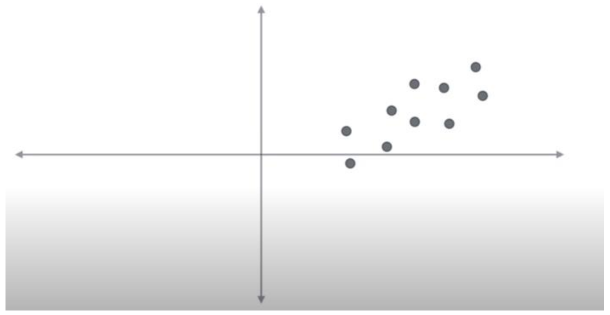


Reference Courtesy:
Luis Serrano

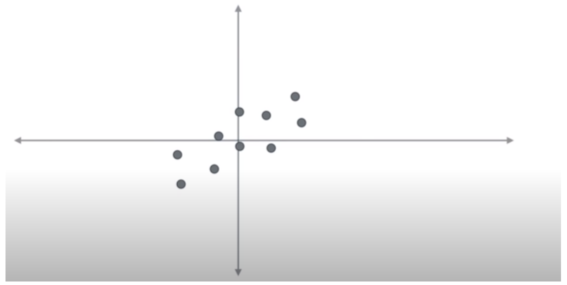
How to find the perfect projection?



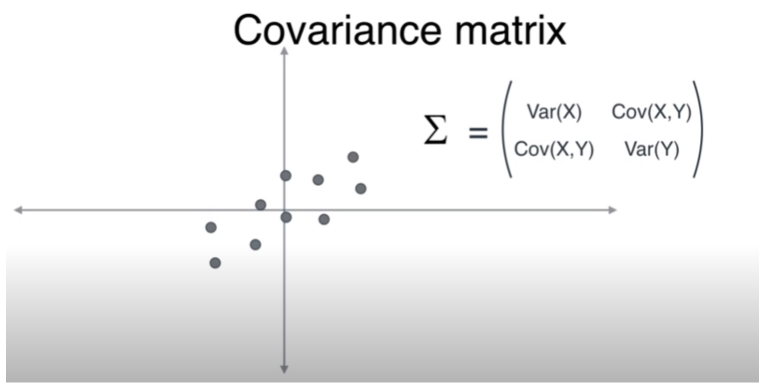
Let’s say we have
a dataset like this



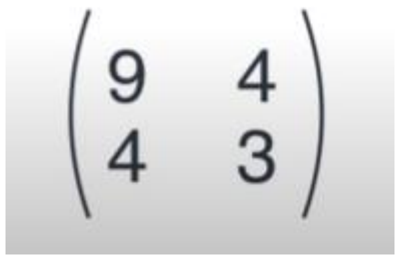
Putting into a
co-ordinate axis



Center it (take the avg.
of co-ordinates, point
of balance)



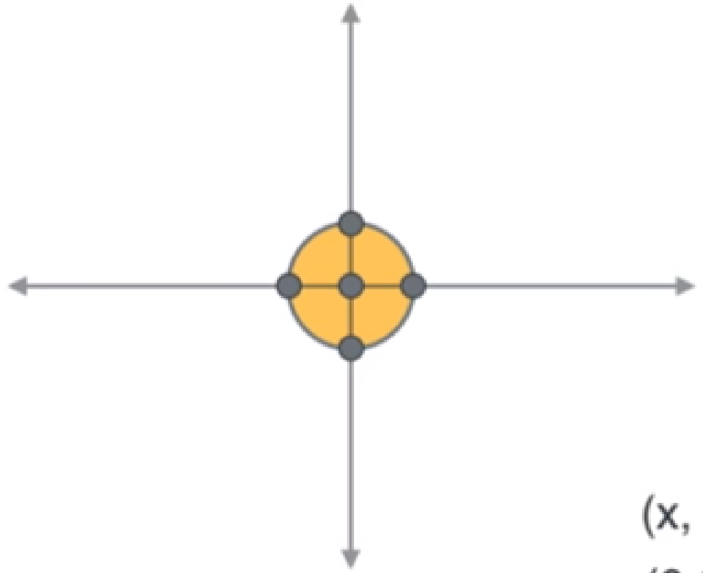
$$\Sigma = \begin{pmatrix} \mathrm{Var}(X) & \mathrm{Cov}(X,Y) \\ \mathrm{Cov}(X,Y) & \mathrm{Var}(Y) \end{pmatrix}$$



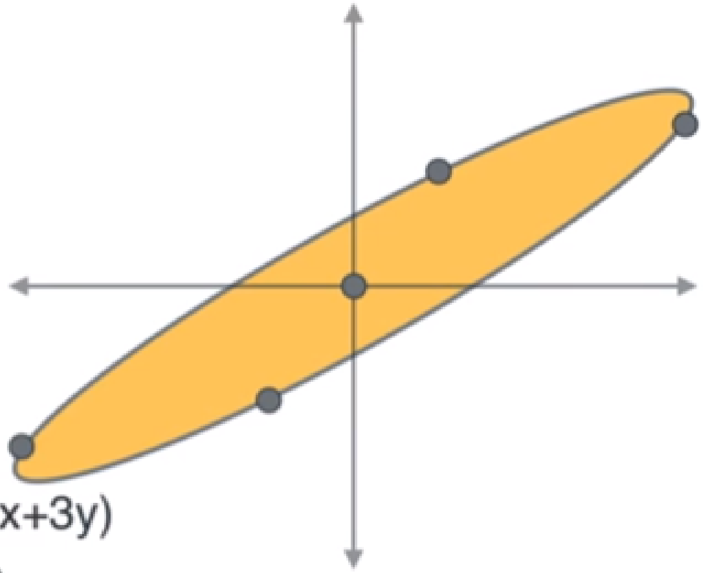
$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

Reference Courtesy:
Luis Serrano

Linear Transformations



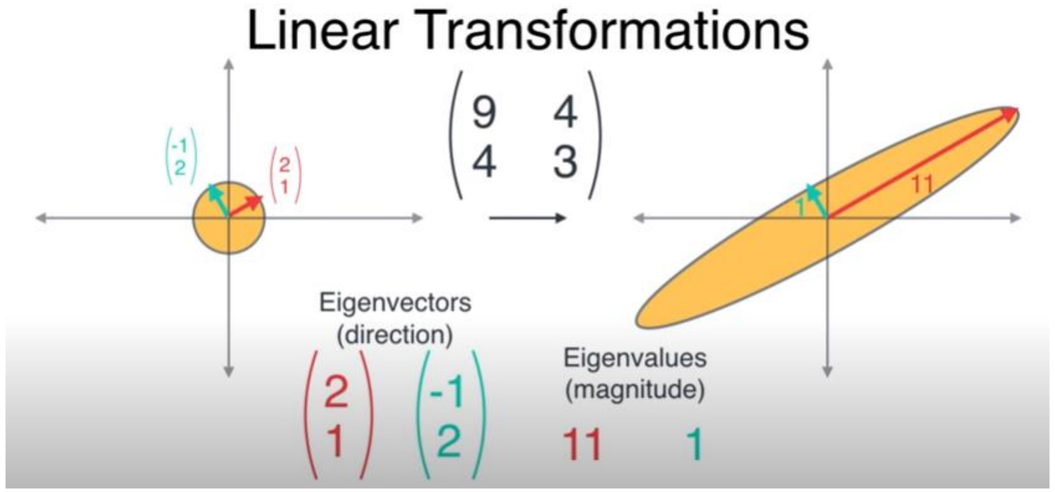
$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} \longrightarrow$$

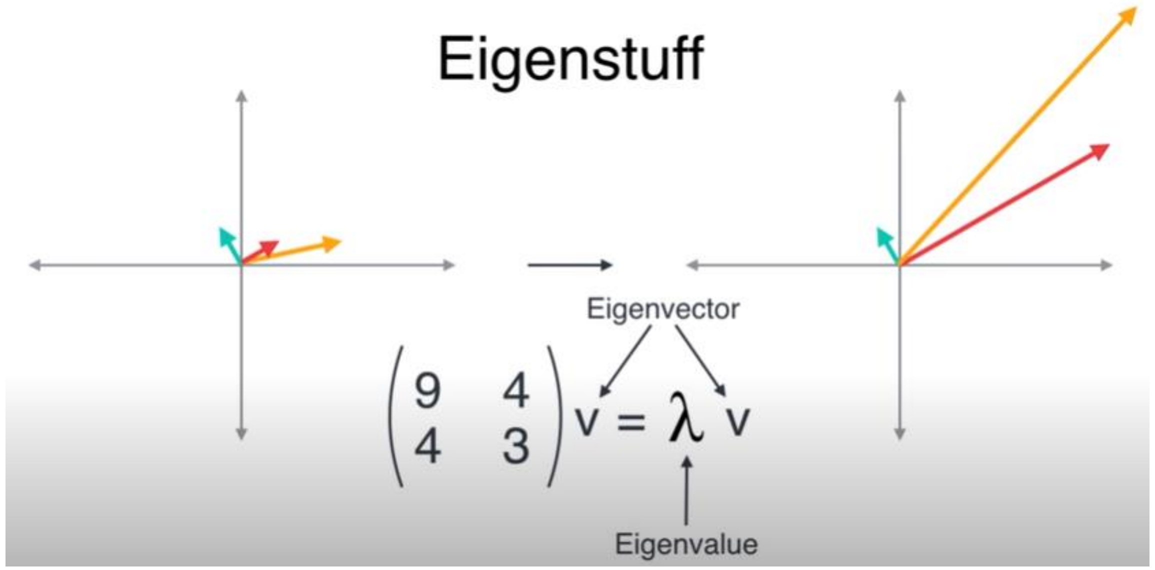


$(x, y) \longrightarrow (9x+4y, 4x+3y)$

(0,0)	(0,0)
(1,0)	(9,4)
(0,1)	(4,3)
(-1,0)	(-9,-4)
(0,-1)	(-4,-3)

Reference Courtesy:
Luis Serrano


Linear Transformations
$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$
$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$
$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$
11
1
Eigenvectors (direction)
$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$
Eigenvalues (magnitude)
11 1


Eigenstuff
Eigenvector
$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} v = \lambda v$
Eigenvalue

Think about this as like this :
Linear Transformation is basically stretching the plane in 2 directions. Direction given by the **Eigen Vectors** & the Amount of stretching (magnitude) is given by **Eigen Values**

Reference Courtesy: Luis Serrano

Eigenvalues

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

Characteristic Polynomial

$$\begin{vmatrix} x-9 & -4 \\ -4 & x-3 \end{vmatrix} = (x-9)(x-3) - (-4)(-4) = x^2 - 12x + 11$$

$$= (x-11)(x-1)$$

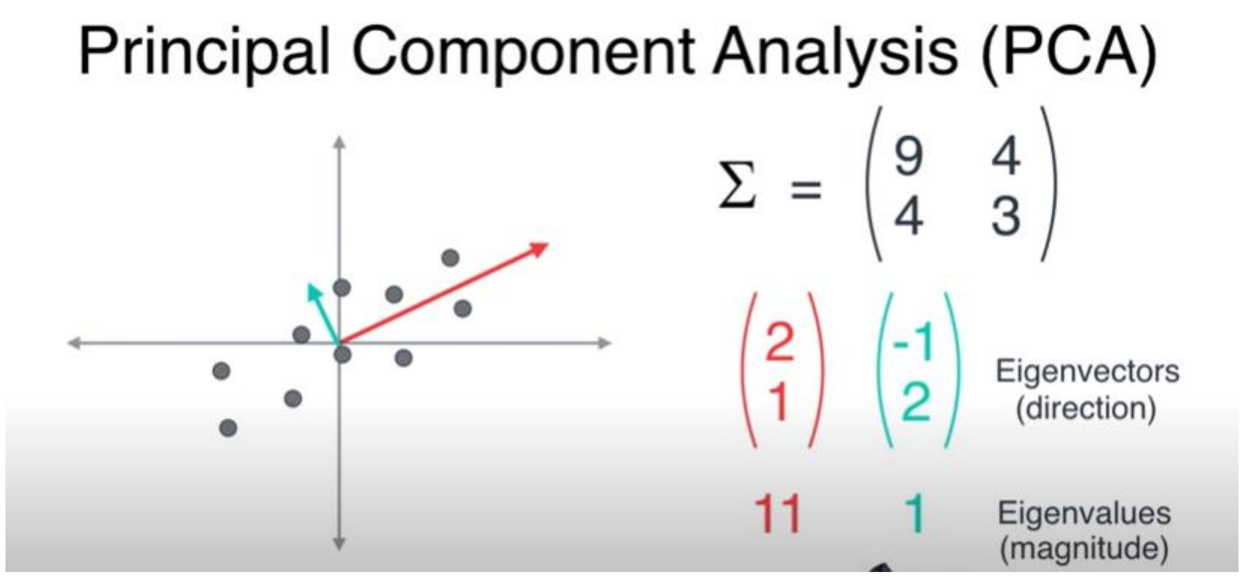
Eigenvalues 11 and 1

Eigenvectors

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = 11\begin{pmatrix} u \\ v \end{pmatrix} \qquad \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = 1\begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

Reference Courtesy:
Luis Serrano

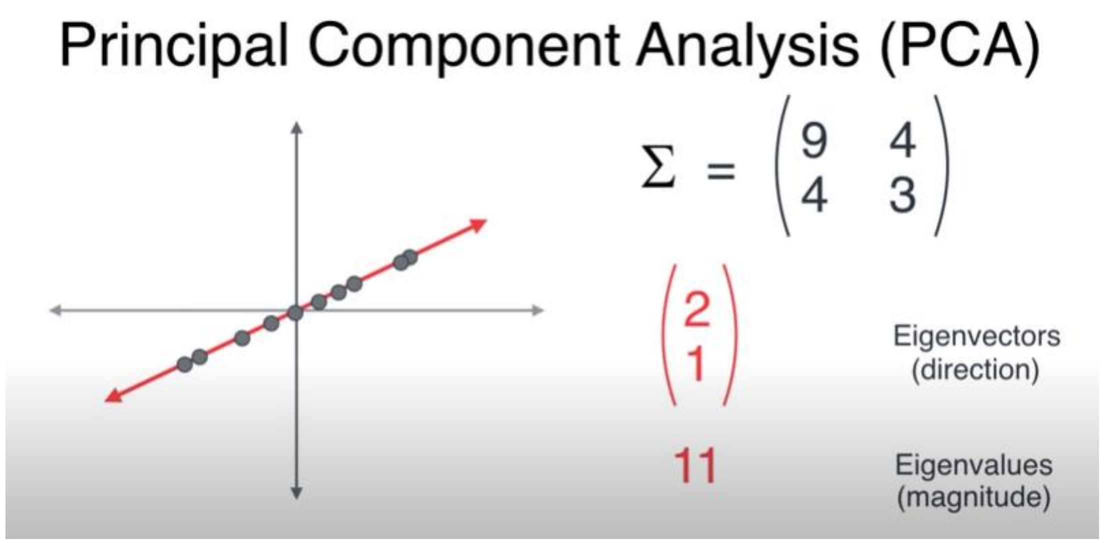


Principal Component Analysis (PCA)

$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$ Eigenvectors (direction)

11 1 Eigenvalues (magnitude)



Principal Component Analysis (PCA)

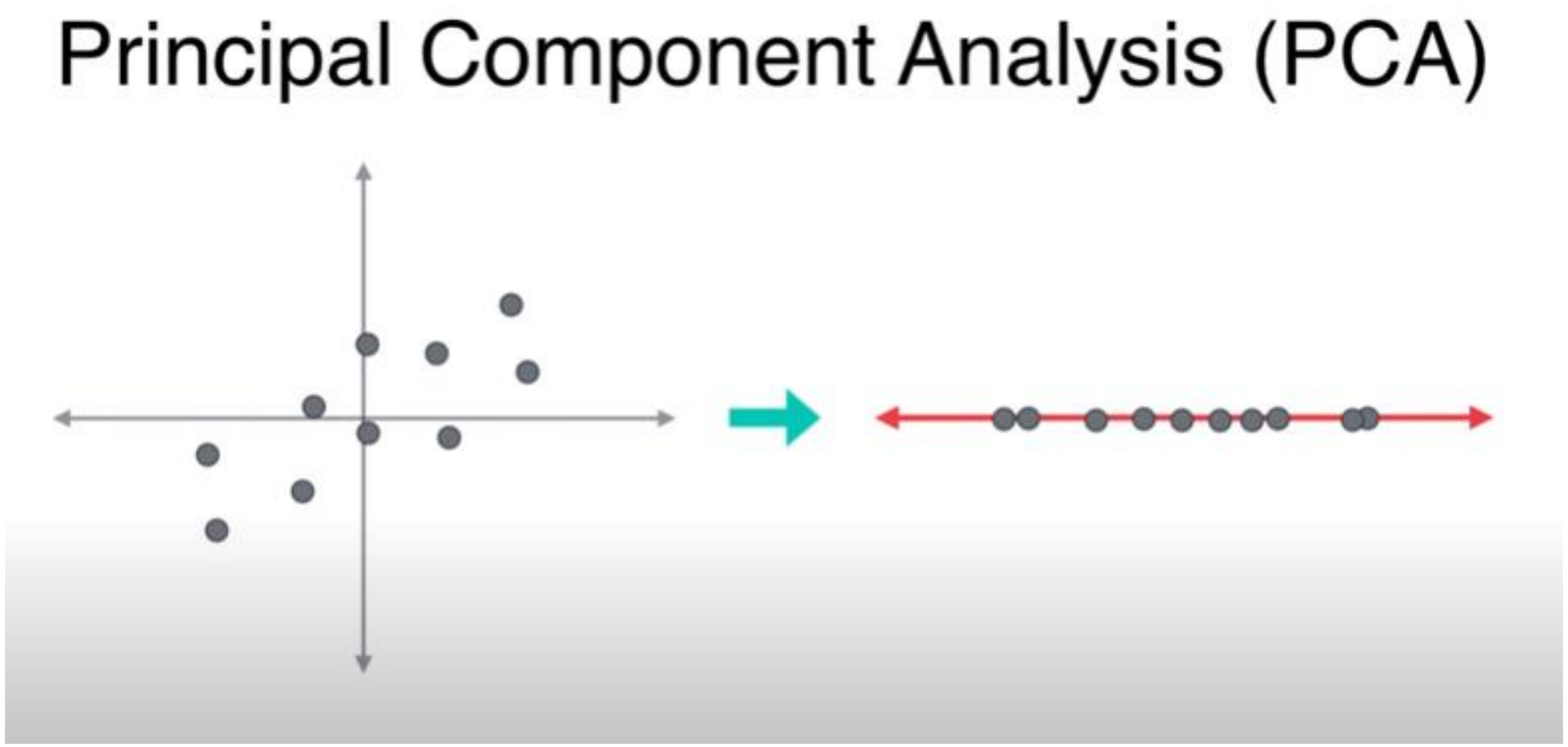
$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ Eigenvectors (direction)

11 Eigenvalues (magnitude)

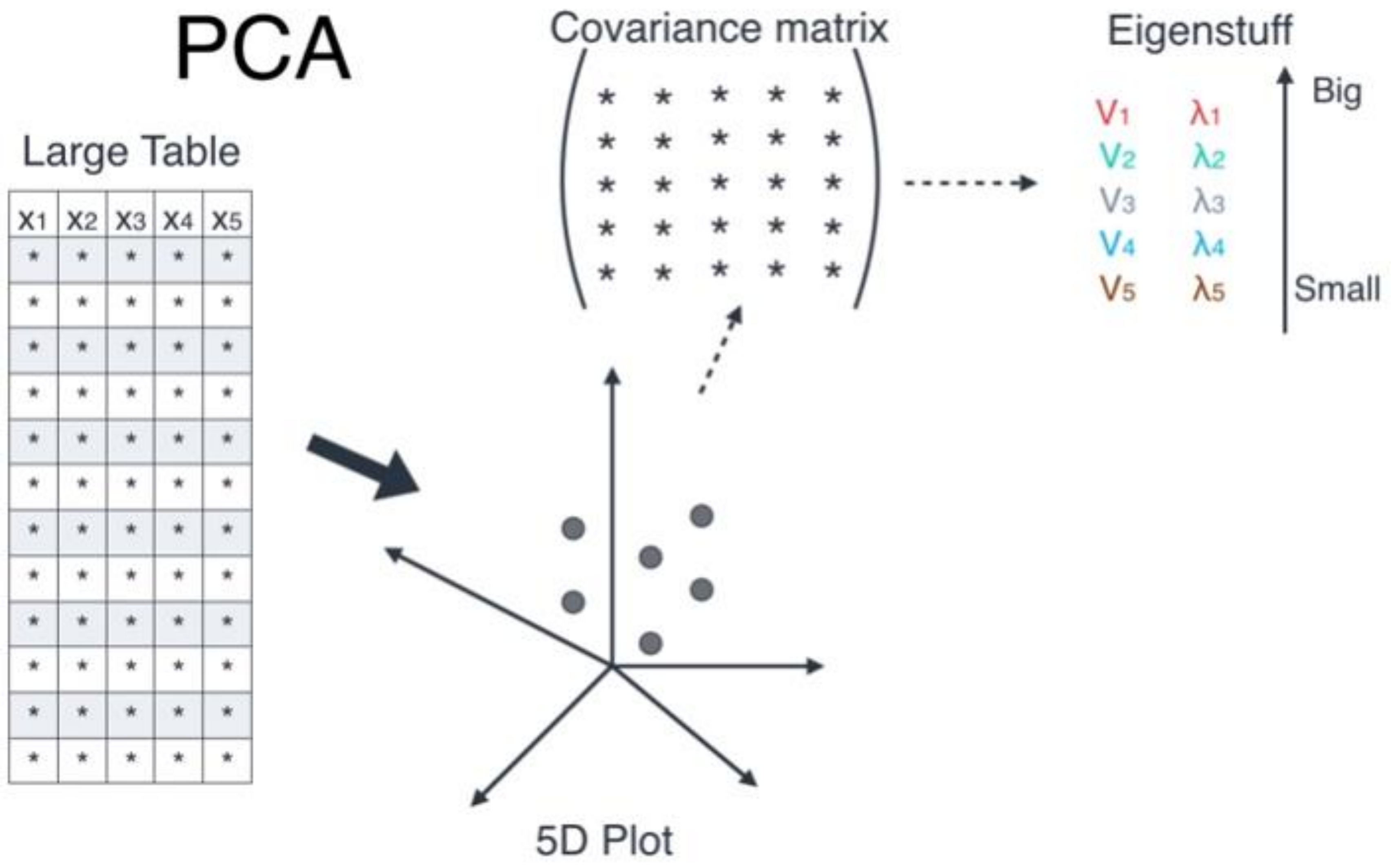
Reference Courtesy:
Luis Serrano

2-D into 1-D



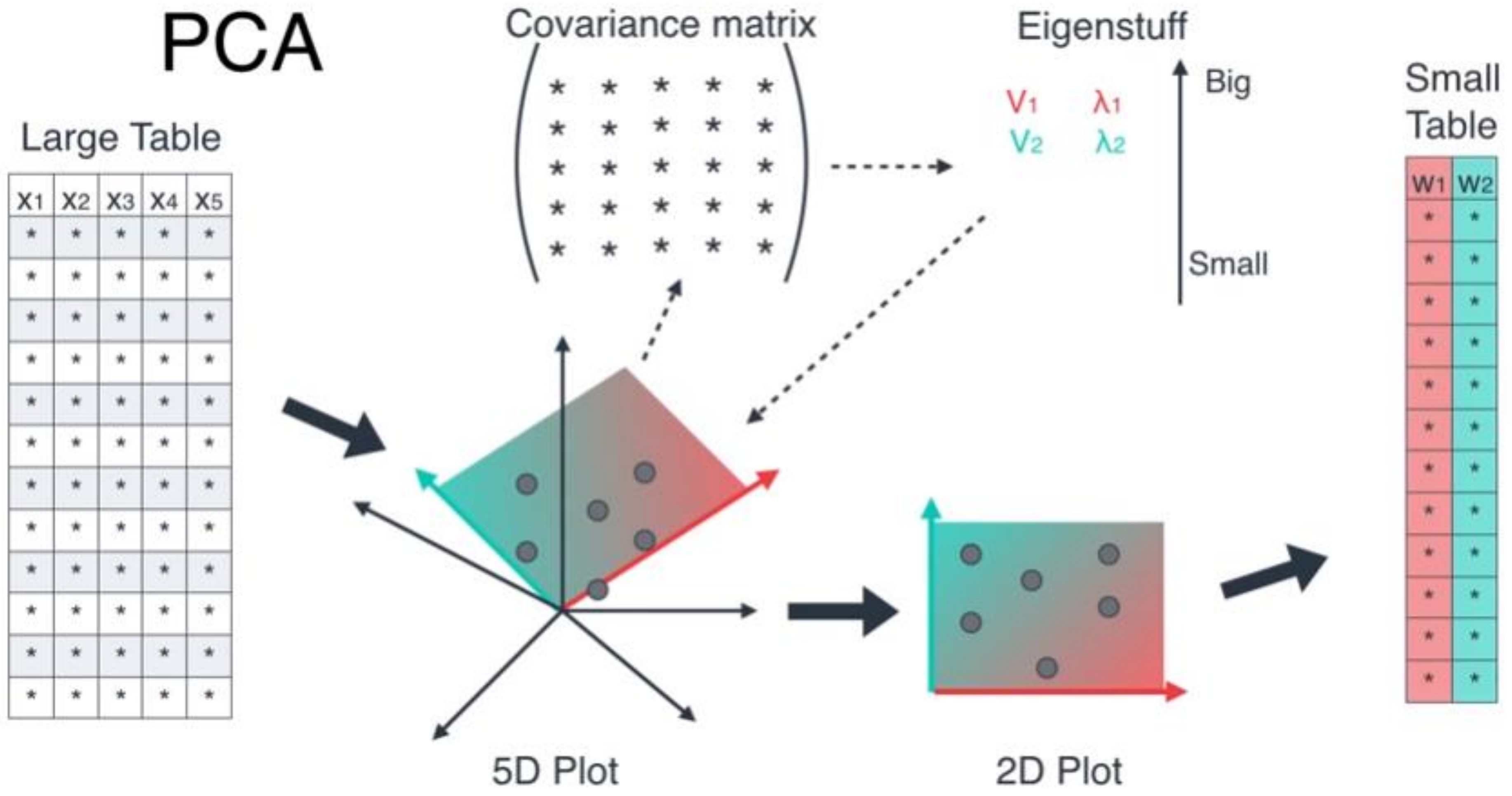
Reference Courtesy:
Luis Serrano

PCA



Reference Courtesy:
Luis Serrano

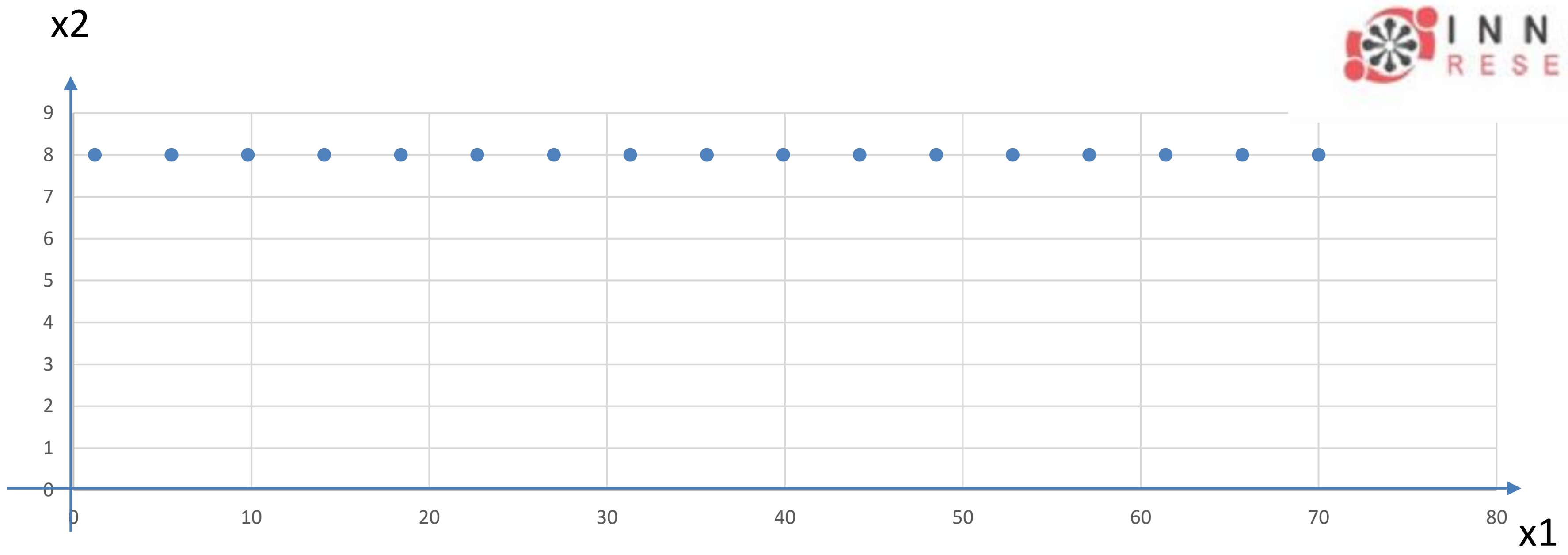
PCA



Reference Courtesy:
Luis Serrano

Which Variable is relevant variable ?

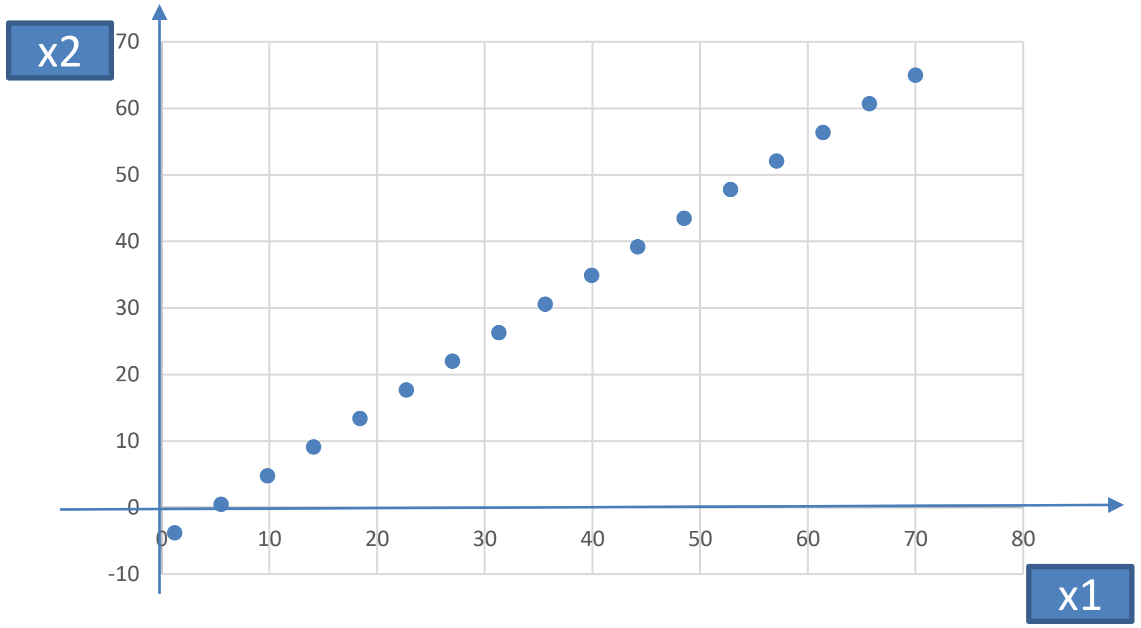
y	x1	x2
10.2	1.2	8
22.7	5.5	8
35.2	9.8	8
47.7	14.1	8
60.2	18.4	8
72.7	22.7	8
85.2	27	8
97.7	31.3	8
110.2	35.6	8
122.7	39.9	8
135.2	44.2	8
147.7	48.5	8
160.2	52.8	8
172.7	57.1	8
185.2	61.4	8
197.7	65.7	8
210.2	70	8



All the variation in the explanatory variable is in x1, direction only
This direction is called dominant **Principal Component** of explanatory variables
The x_2 direction is **redundant**, since not variance along that axis

How many relevant features are here ?

y	x1	x2
10.2	1.2	-3.8
22.7	5.5	0.5
35.2	9.8	4.8
47.7	14.1	9.1
60.2	18.4	13.4
72.7	22.7	17.7
85.2	27	22
97.7	31.3	26.3
110.2	35.6	30.6
122.7	39.9	34.9
135.2	44.2	39.2
147.7	48.5	43.5
160.2	52.8	47.8
172.7	57.1	52.1
185.2	61.4	56.4
197.7	65.7	60.7
210.2	70	65



This seems variations are both in x_1 and x_2 and both are correlated

How many relevant features are here ?

y	x1	x2
10.2	1.2	-3.8
22.7	5.5	0.5
35.2	9.8	4.8
47.7	14.1	9.1
60.2	18.4	13.4
72.7	22.7	17.7
85.2	27	22
97.7	31.3	26.3
110.2	35.6	30.6
122.7	39.9	34.9
135.2	44.2	39.2
147.7	48.5	43.5
160.2	52.8	47.8
172.7	57.1	52.1
185.2	61.4	56.4
197.7	65.7	60.7
210.2	70	65

If we create new variable

$$X_1 = x_1 + x_2$$
$$X_2 = x_1 - x_2$$

In terms of new variable $X_1$and X_2it clear that there is only one true relevant feature

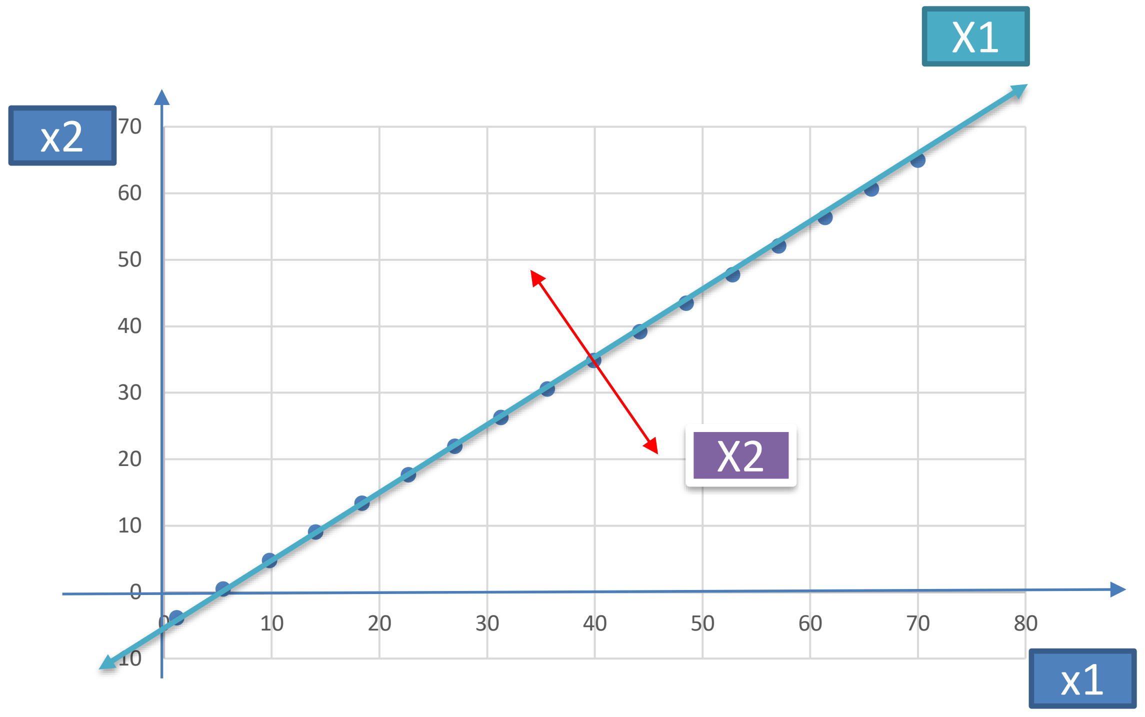
y	X1	X2
10.2	-2.6	5
22.7	6	5
35.2	14.6	5
47.7	23.2	5
60.2	31.8	5
72.7	40.4	5
85.2	49	5
97.7	57.6	5
110.2	66.2	5
122.7	74.8	5
135.2	83.4	5
147.7	92	5
160.2	100.6	5
172.7	109.2	5
185.2	117.8	5
197.7	126.4	5
210.2	135	5

Transformed variables: Interpretation

The transformation we did is equivalent to viewing from the data points from a rotated co-ordinate system.

Green = X1
Red = X2

The dominant principal component is X1 axis



Principal Component Analysis

PCA is the method which allows you to identify the “directions” in which most of the variations in the data is present.

Equivalently, it can be thought as method to identify the “directions ” along which there is least variations (or least useful information). Identifying this would allow us to drop this irrelevant direction in our regression/model building.

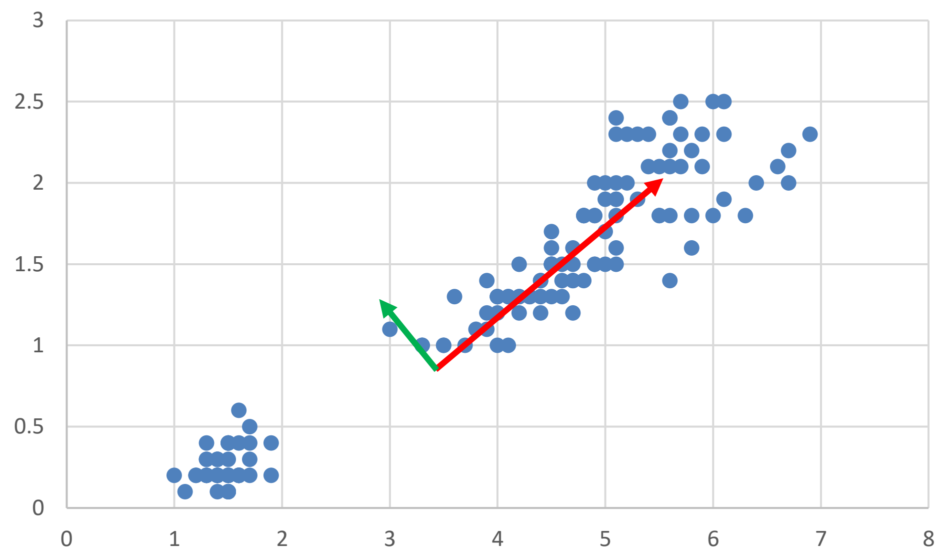
Thus, **PCA is a method that brings together:**

1.A measure of how each variable is associated with one another. (**Covariance matrix**.)
2.The directions in which our data are dispersed. (**Eigenvectors**.)
3.The relative importance of these different directions. (**Eigenvalues**.)

PCA combines our predictors and allows us to drop the eigenvectors that are relatively unimportant.

Principal Component directions

y	x1	x2
5.1	1.4	0.2
4.9	1.4	0.2
4.7	1.3	0.2
4.6	1.5	0.2
5	1.4	0.2
5.4	1.7	0.4
4.6	1.4	0.3
5	1.5	0.2
4.4	1.4	0.2
4.9	1.5	0.1
5.4	1.5	0.2
4.8	1.6	0.2
4.8	1.4	0.1
4.3	1.1	0.1
5.8	1.2	0.2
5.7	1.5	0.4
5.4	1.3	0.4
5.1	1.4	0.3
5.7	1.7	0.3
5.1	1.5	0.3
5.4	1.7	0.2



```
data = pd.read_csv('./data/sample_data.csv')
data.head()
```

```
X = data[['x1','x2']]
```

```
sns.relplot('x1','x2',data=X,aspect=2.5)
plt.show()
```

PCA Methodology

x1	x2
1.4	0.2
1.4	0.2
1.3	0.2
1.5	0.2
1.4	0.2
1.7	0.4
1.4	0.3
1.5	0.2
1.4	0.2
1.5	0.1
1.5	0.2
1.6	0.2
1.4	0.1
1.1	0.1
1.2	0.2
1.5	0.4
1.3	0.4
1.4	0.3
1.7	0.3
1.5	0.3
1.7	0.2

Starts with analyzing covariance matrix of features

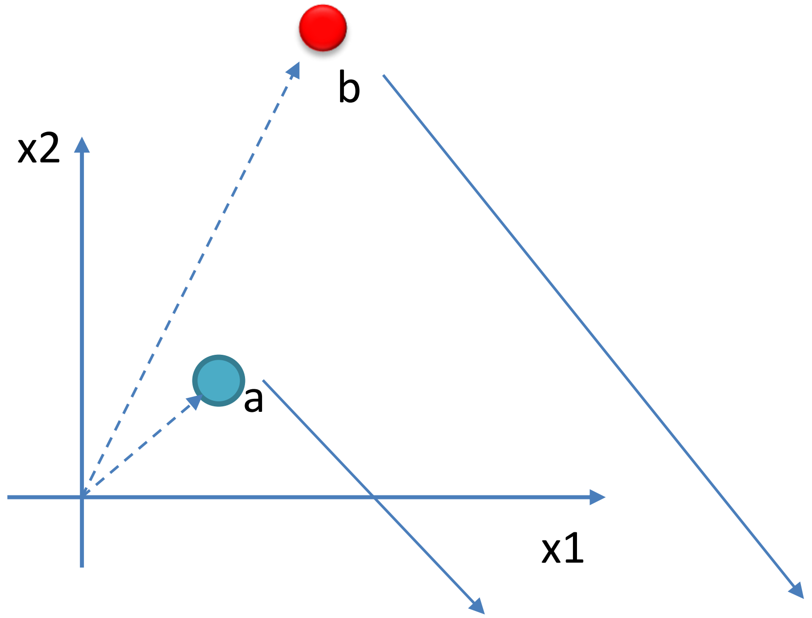
$$cov = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) \\ cov(x_1, x_2) & cov(x_2, x_2) \end{bmatrix}$$

```
X.cov()
```

```
          x1        x2
x1  3.113179  1.296387
x2  1.296387  0.582414
```

The covariance matrix contains information about both correlation and the "special direction" of maximal variance

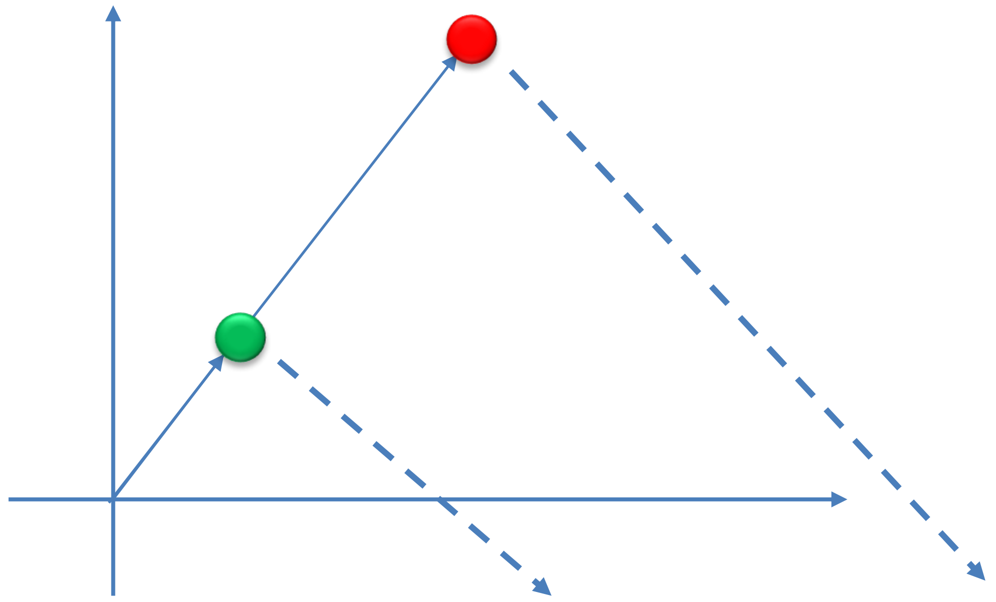
Matrix as a transformation on a vector



$$\begin{bmatrix} 3.11 & 1.29 \\ 1.29 & 0.58 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4.4 \\ 1.87 \end{bmatrix}$$

Matrix as a transformation on a vector

Special Vectors



$$\begin{bmatrix} 3.11 & 1.29 \\ 1.29 & 0.58 \end{bmatrix} \begin{bmatrix} 0.9215 \\ 0.3882 \end{bmatrix} = \begin{bmatrix} 3.3722 \\ 1.4208 \end{bmatrix} = 3.66 \begin{bmatrix} 0.9215 \\ 0.3882 \end{bmatrix}$$

For given matrix there are special directions, along which its effect only to stretch (without rotation). Such direction is called **Eigen direction or Eigen vectors**

Eigen vectors mathematics

The eigenvectors and eigenvalues of matrix **A** are defined to be the nonzero **x** and λ values that solve

$$A\,X = \lambda\,X$$ (A is just stretching)

For a n-dim square matrix, there are atmost n eigen-vectors and eigen-values.

Eigen vectors & PCA

Eigenvectors are the principal component directions

Eigenvalues are the magnitude of stretch

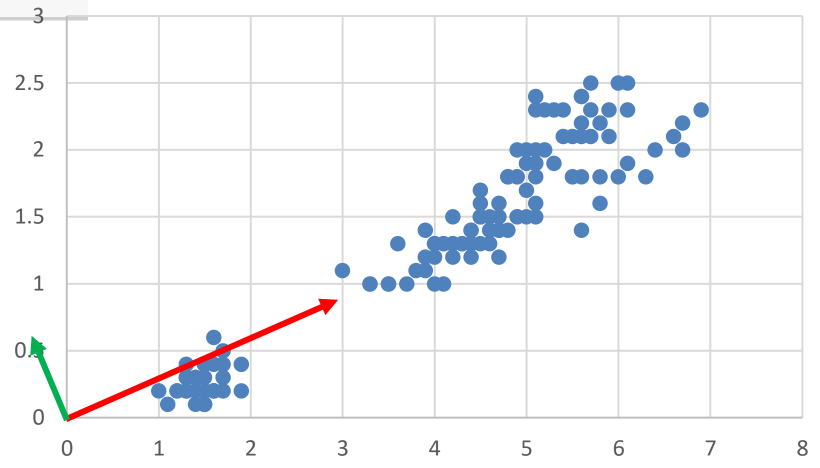
Eigenvalues represent the magnitude of variance of those directions

Eigen values and Eigen vectors

```python
eigvalue, eigvector = np.linalg.eig(X.cov())
print('INFO: Eigenvectos = \n',eigvector)
print('\nINFO: Eigenvalues =',eigvalue)
```

```
INFO: Eigenvectos =
 [[ 0.92154695 -0.38826694]
 [ 0.38826694  0.92154695]]

INFO: Eigenvalues = [3.65937449 0.03621925]
```



$$E_1 = \begin{bmatrix} 0.9215 \\ 0.3882 \end{bmatrix} \Rightarrow \lambda_1 = 3.6593$$

$$E_2 = \begin{bmatrix} -0.3882 \\ 0.9215 \end{bmatrix} \Rightarrow \lambda_2 = 0.0362$$

Dominant
Principal Component

Eigen vector and Eigen values

Remember, in the other example when we transformed the data points from original variable x_1, x_2 into new transformed variables, X1 and X2, we could reduce the dimensions ?

The matrix of eigen-vectors as a whole also allows you to transform each one of our data-points into new variables X_1 & X_2

Transform into Principal Component

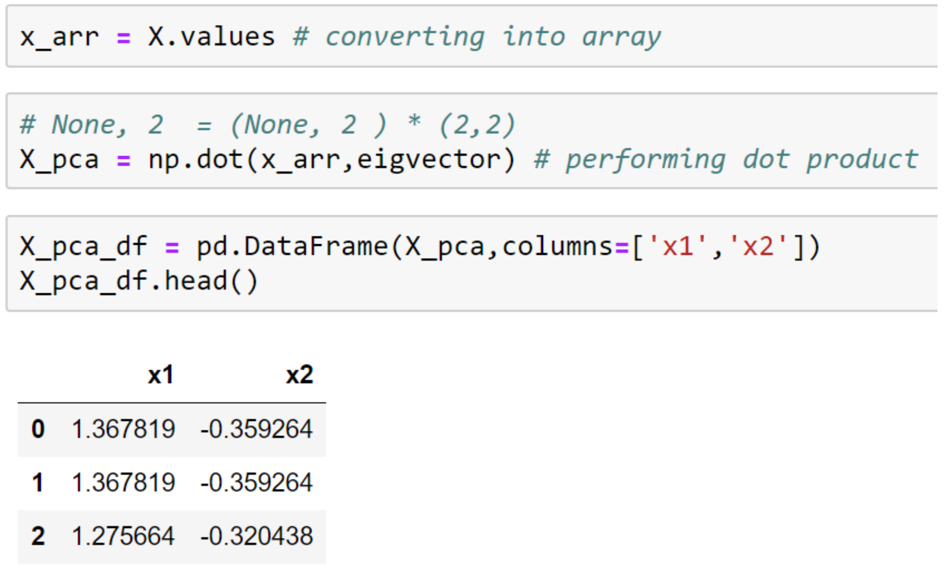
Any record in our data set (x_1, x_2)
When multiplied by the matrix of eigenvector, we get the new coordinates in the rotated principal component axis.

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 0.9215 & -0.3882 \\ 0.3882 & 0.9215 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 0.9215 * x_1 - 0.3882 * x_2 \\ 0.3882 * x_1 + 0.9215 * x_2 \end{bmatrix}$$

x1	x2
1.4	0.2
1.4	0.2
1.3	0.2
1.5	0.2
1.4	0.2
1.7	0.4
1.4	0.3
1.5	0.2
1.4	0.2
1.5	0.1
1.5	0.2
1.6	0.2
1.4	0.1
1.1	0.1
1.2	0.2
1.5	0.4
1.3	0.4
1.4	0.3
1.7	0.3
1.5	0.3
1.7	0.2

X1	X2
1.367819	-0.35926
1.367819	-0.35926
1.275664	-0.32044
1.459974	-0.39809
1.367819	-0.35926
1.721937	-0.29144
1.406646	-0.26711
1.459974	-0.39809
1.367819	-0.35926
1.421147	-0.49025
1.459974	-0.39809
1.552129	-0.43692
1.328992	-0.45142
1.052528	-0.33494
1.18351	-0.28161
1.537627	-0.21378
1.353318	-0.13613
1.406646	-0.26711
1.68311	-0.38359
1.498801	-0.30594
1.644283	-0.47574



```
x_arr = X.values # converting into array
```

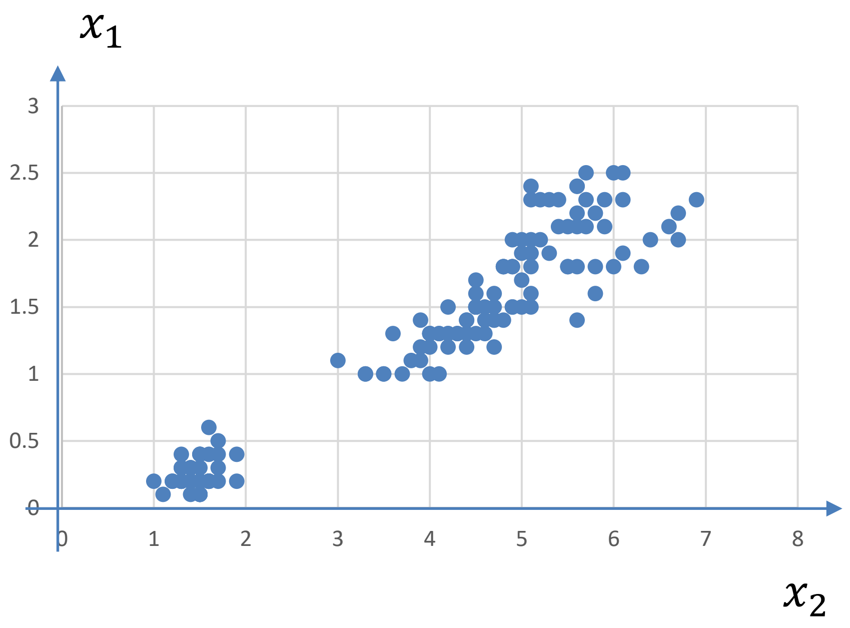
```
# None, 2  = (None, 2 ) * (2,2)
X_pca = np.dot(x_arr,eigvector) # performing dot product
```

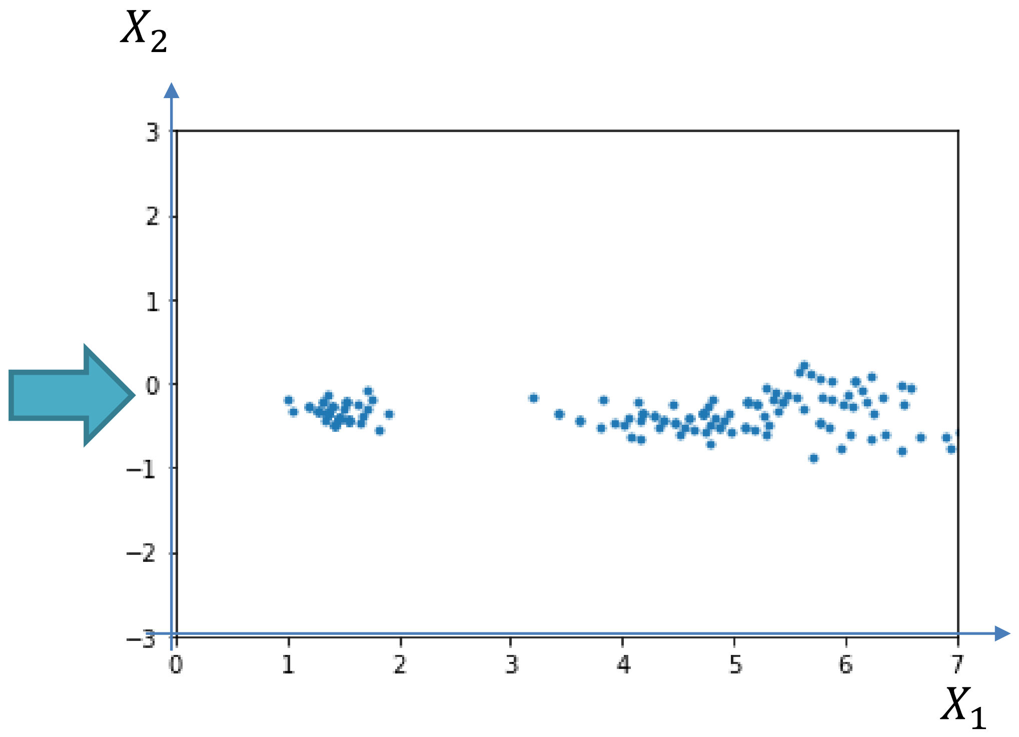
```
X_pca_df = pd.DataFrame(X_pca,columns=['x1','x2'])
X_pca_df.head()
```

	x1	x2
0	1.367819	-0.359264
1	1.367819	-0.359264
2	1.275664	-0.320438

Var(X1) = 3.65   Var(X2) = 0.0362

Data transformed into basis of Principal Component

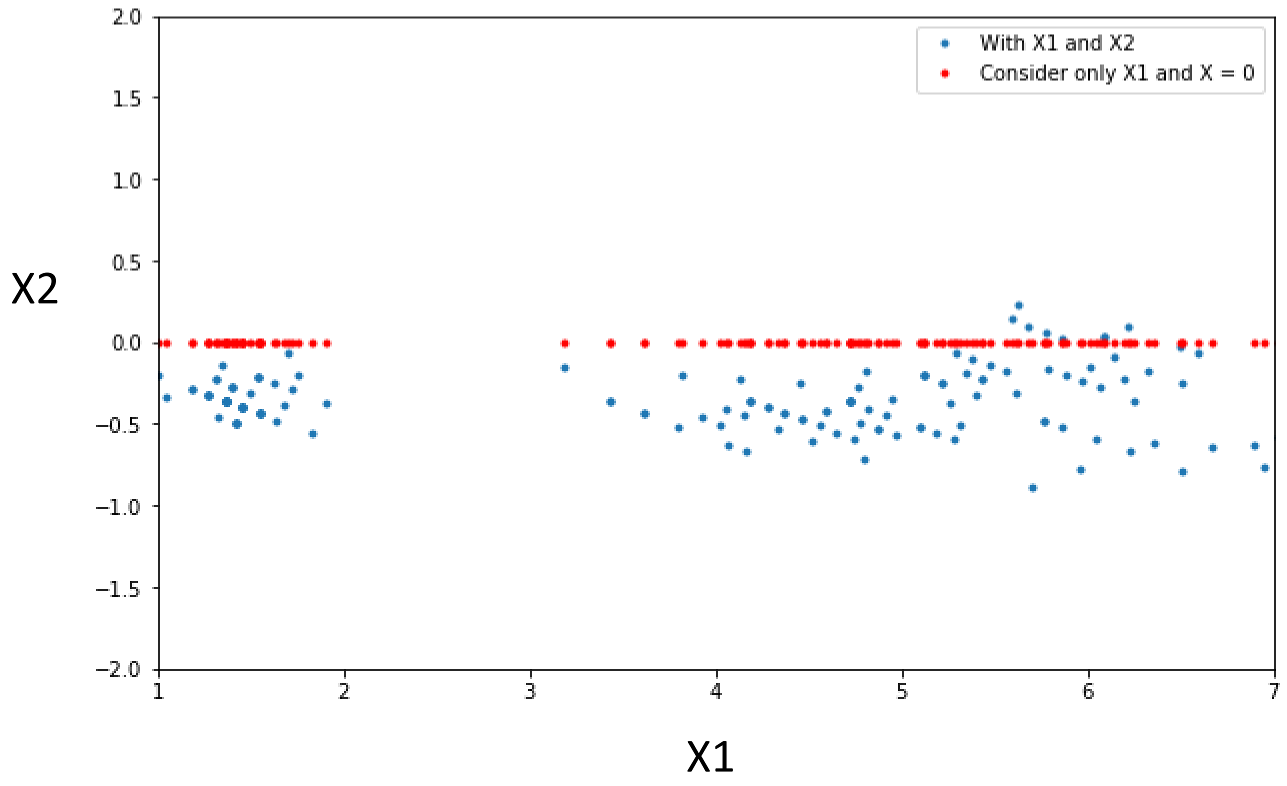




X2 variable has small variance, we can now drop it by setting it to zero

Dimensionality Reduction

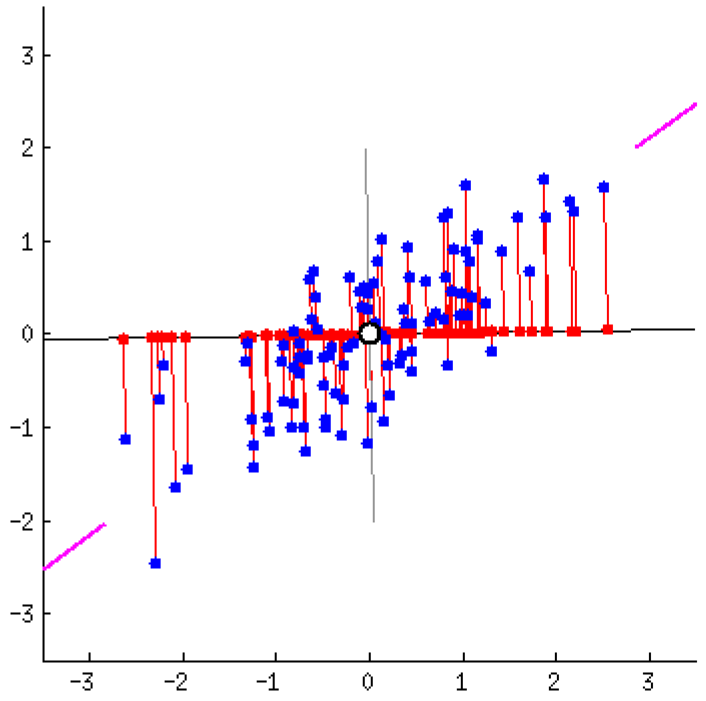
X_2 has been set to zero



So, now instead of doing regression for x_1, x_2. PCA allows to do regression only with X_1.
This is point of doing PCA. It allows us to ignore variable with low variance

INNOMATICS
RESEARCH LABS

Principle Axis Animation



Principal Component Analysis

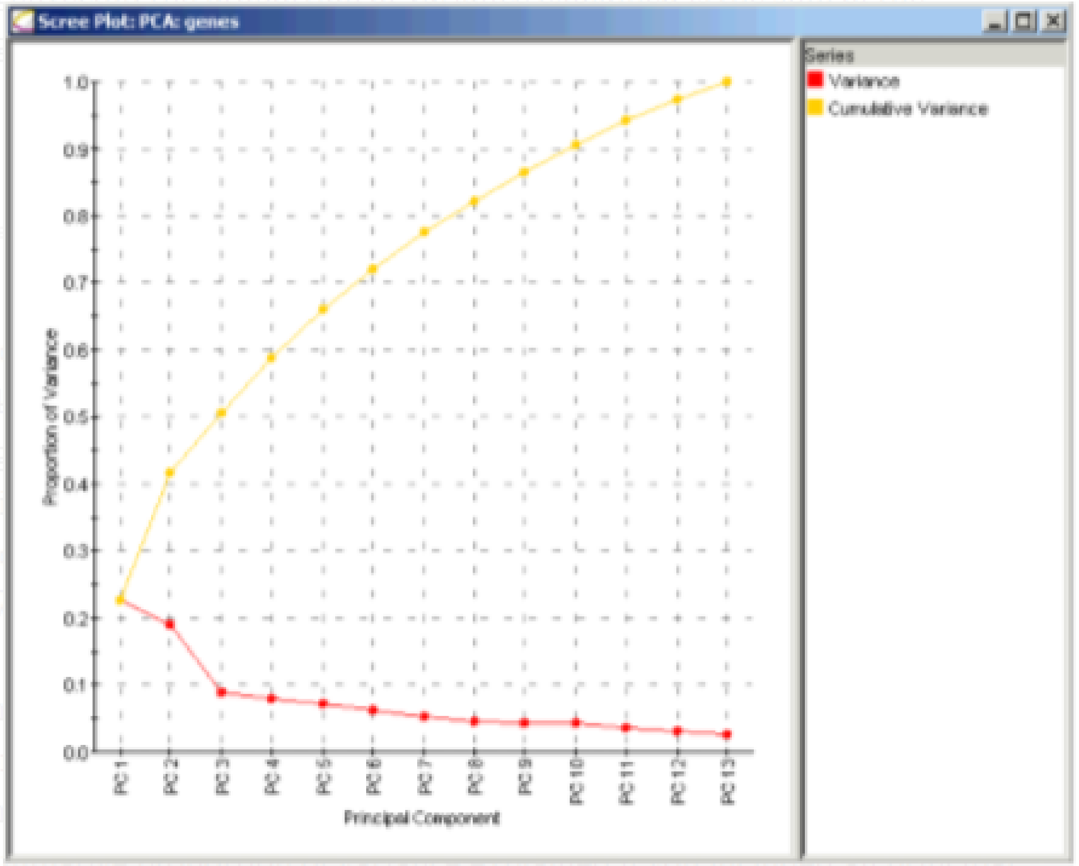
- Understanding PCA through animation.

- Each blue dot on the plot represents a point from data given by its x & y coordinate.
- A line P (red line) is drawn from the center of the dataset i.e. from the mean of x & y.
- Every point on the graph is projected on this line shown by two sets of points red & green.
- The spread or variance of data along line p is given by the distance between the two big red points.
- As the line p rotates the distance between the two red points changes according to the angle created by line p with the x-axis.
- The purple lines which join a point and its projection represent the error which arises when we approximate a point by its projection.

Principal Component Analysis

- The approximation error should be small, when the new variables closely approximate the old variables.
- The squared sum of the lengths of all purple lines gives the total error in approximation.
- The angle which minimizes the squared sum of errors also maximizes the distance between the red points.
- The direction of maximum spread is called the ***principal axis***.
- We apply the same procedure to find the next principal axis, which must be orthogonal to the other principal axes.
- Once, we get all the principal axes, the dataset is projected onto these axes. The columns in the projected or transformed dataset are called ***principal components***.

Selecting number of principal components



Consider this scree plot for genetic data. The red line indicates the proportion of variance explained by each feature, which is calculated by taking that principal component's eigenvalue divided by the sum of all eigenvalues. The proportion of variance explained by including only principal component 1 is $\lambda_1/(\lambda_1 + \lambda_2 + \ldots + \lambda p)$, which is about 23%. The proportion of variance explained by including only principal component 2 is $\lambda_2/(\lambda_1 + \lambda_2 + \ldots + \lambda p)$, or about 19%.

The proportion of variance explained by including both principal components 1 and 2 is $(\lambda_1 + \lambda_2)/(\lambda_1 + \lambda_2 + \ldots + \lambda p)$, which is about 42%. This is where the yellow line comes in; the yellow line indicates the cumulative proportion of variance explained if you included all principal components up to that point. For example, the yellow dot above PC2 indicates that including principal components 1 and 2 will explain about 42% of the total variance in the model.

While PCA is a very technical method relying on in-depth linear algebra algorithms, it’s a relatively intuitive method when you think about it:

1) First, the covariance matrix $\boldsymbol{Z}^T\boldsymbol{Z}$ is a matrix that contains estimates of how every variable in $\boldsymbol{Z}$ relates to every other variable in $\boldsymbol{Z}$. Understanding how one variable is associated with another is quite powerful.

2) Second, eigenvalues and eigenvectors are important. Eigenvectors represent directions. Think of plotting your data on a multidimensional scatterplot. Then one can think of an individual eigenvector as a particular “direction” in your scatterplot of data. Eigenvalues represent magnitude, or importance. Bigger eigenvalues correlate with more important directions.

3) Finally, we make an assumption that more variability in a particular direction correlates with explaining the behavior of the dependent variable. Lots of variability usually indicates signal, whereas little variability usually indicates noise. Thus, the more variability there is in a particular direction is, theoretically, indicative of something important we want to detect.

When should you use PCA?

- Reducing the dimensionality of the dataset reduces the size.
- If your learning algorithm is too slow because the input dimension is too high, then using PCA to speed it up.
-

Limitations of PCA

- If the number of variables is large, it becomes hard to interpret the principal components.
- PCA is most suitable when variables have a linear relationship among them.
- PCA is influenced to big outliers.