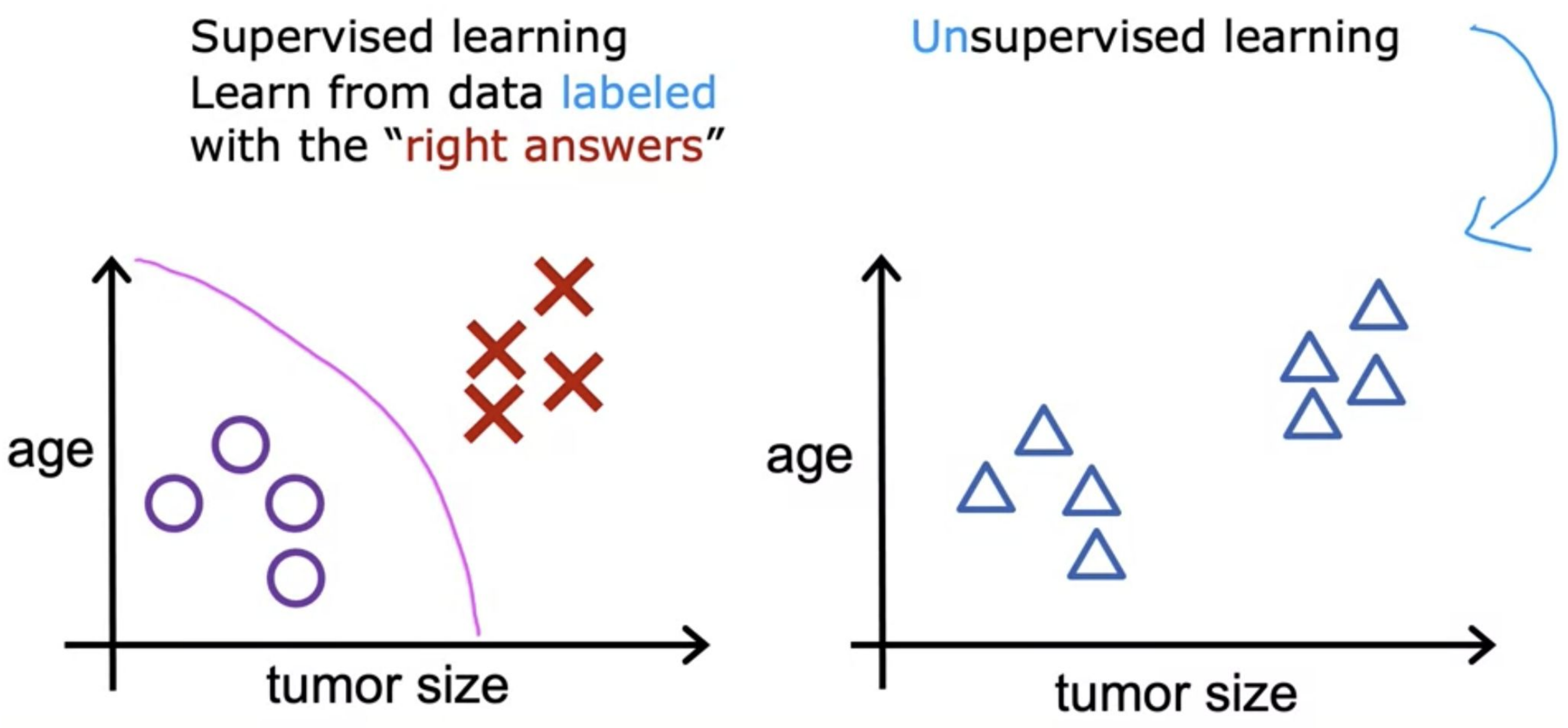
AIFS
AI INSTITUTE FOR FOOD SYSTEMS

AIBridge

AIBridge

Lecture 8

Introducing Unsupervised Learning



Supervised learning
Learn from data labeled
with the "right answers"

Unsupervised learning

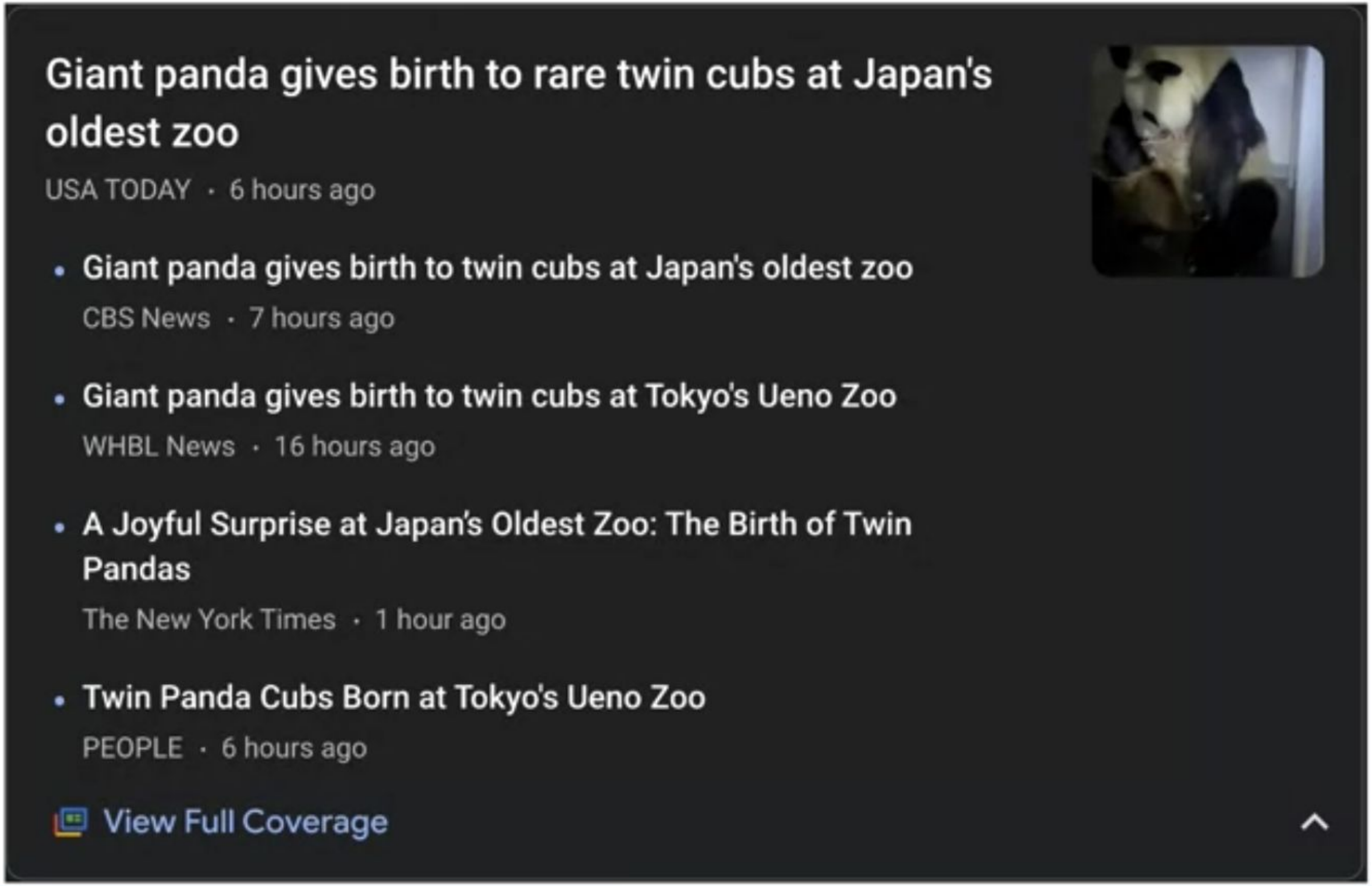
Credit: Andrew Ng, Machine Learning

Unsupervised Learning

Clustering

Dimension reduction

Clustering: Google news



Giant panda gives birth to rare twin cubs at Japan's oldest zoo

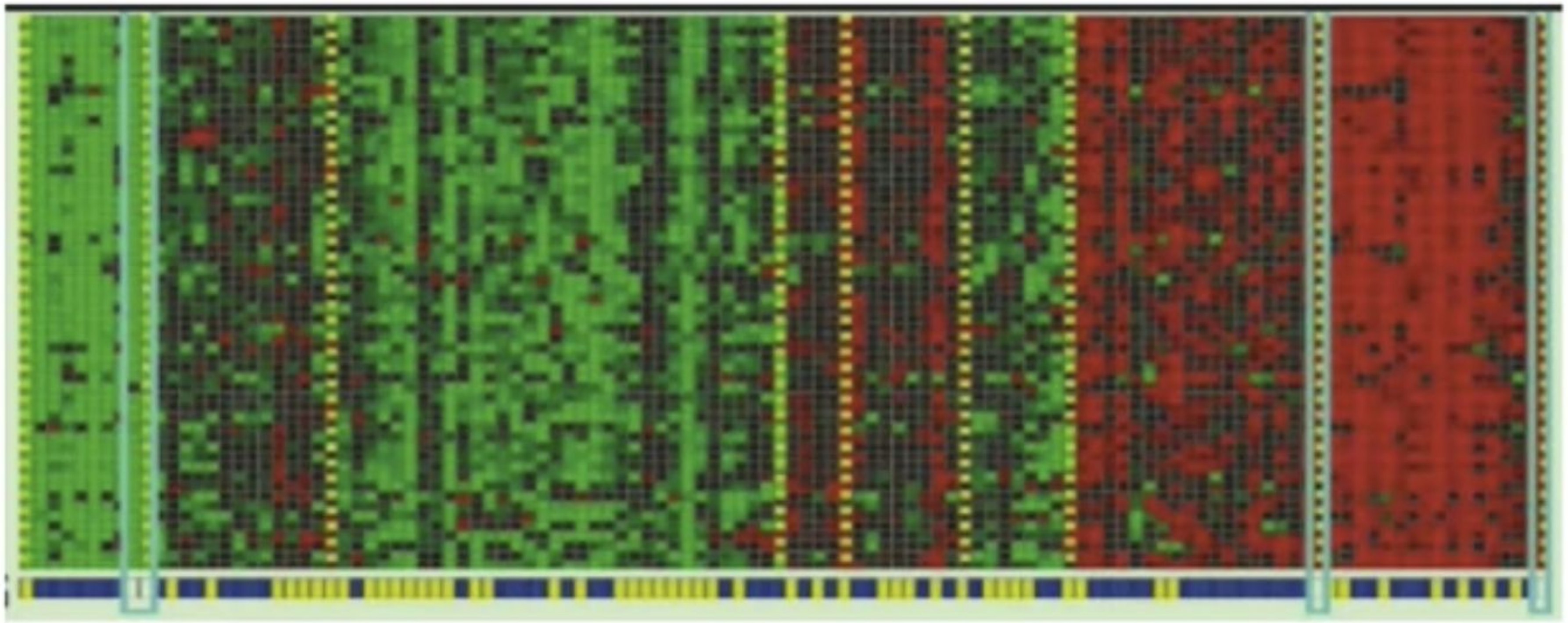
USA TODAY · 6 hours ago

- Giant panda gives birth to twin cubs at Japan's oldest zoo
  CBS News · 7 hours ago
- Giant panda gives birth to twin cubs at Tokyo's Ueno Zoo
  WHBL News · 16 hours ago
- A Joyful Surprise at Japan's Oldest Zoo: The Birth of Twin Pandas
  The New York Times · 1 hour ago
- Twin Panda Cubs Born at Tokyo's Ueno Zoo
  PEOPLE · 6 hours ago

Credit: Andrew Ng, Machine Learning

Clustering: DNA microarray

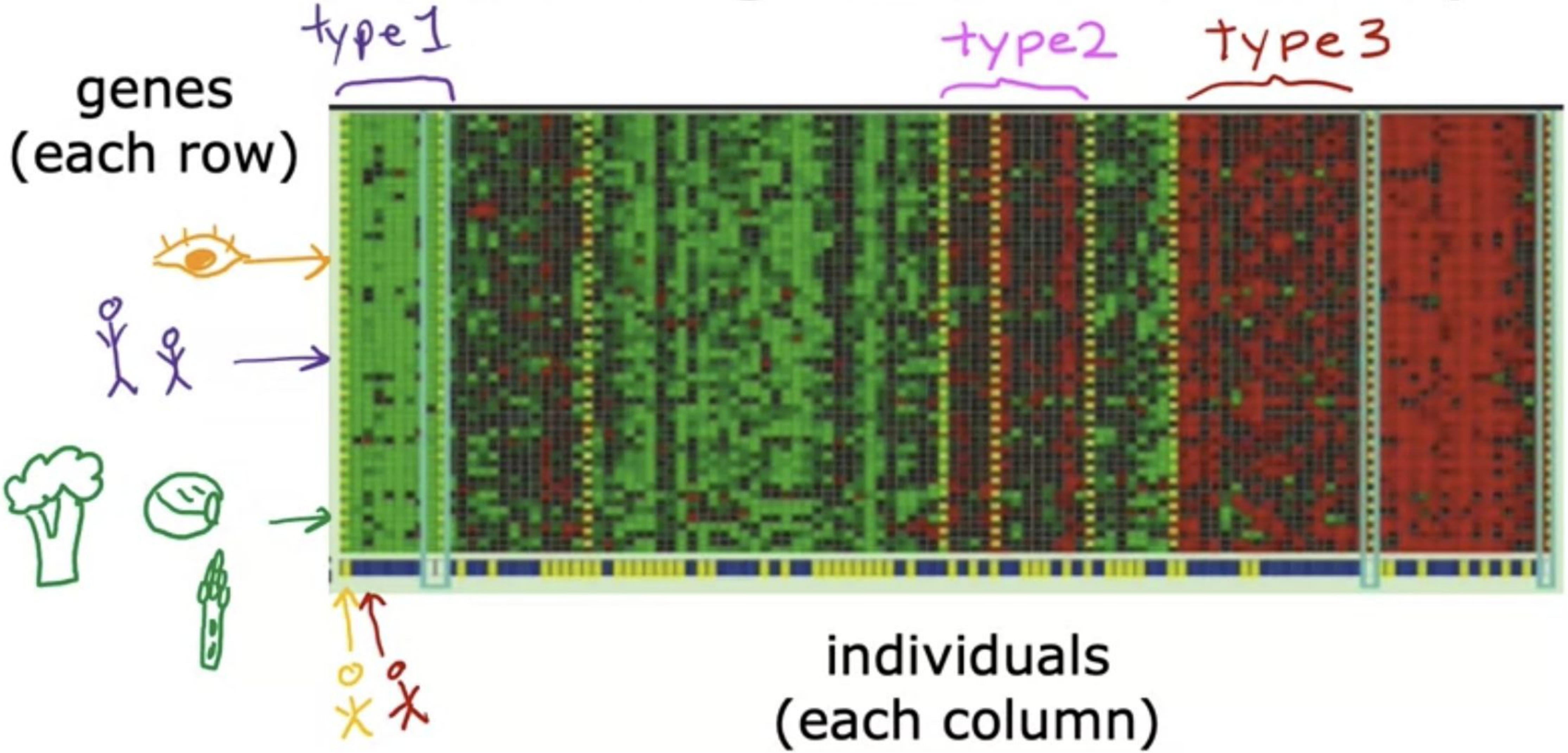
genes
(each row)



individuals
(each column)

Credit: Andrew Ng, Machine Learning

Clustering: DNA microarray

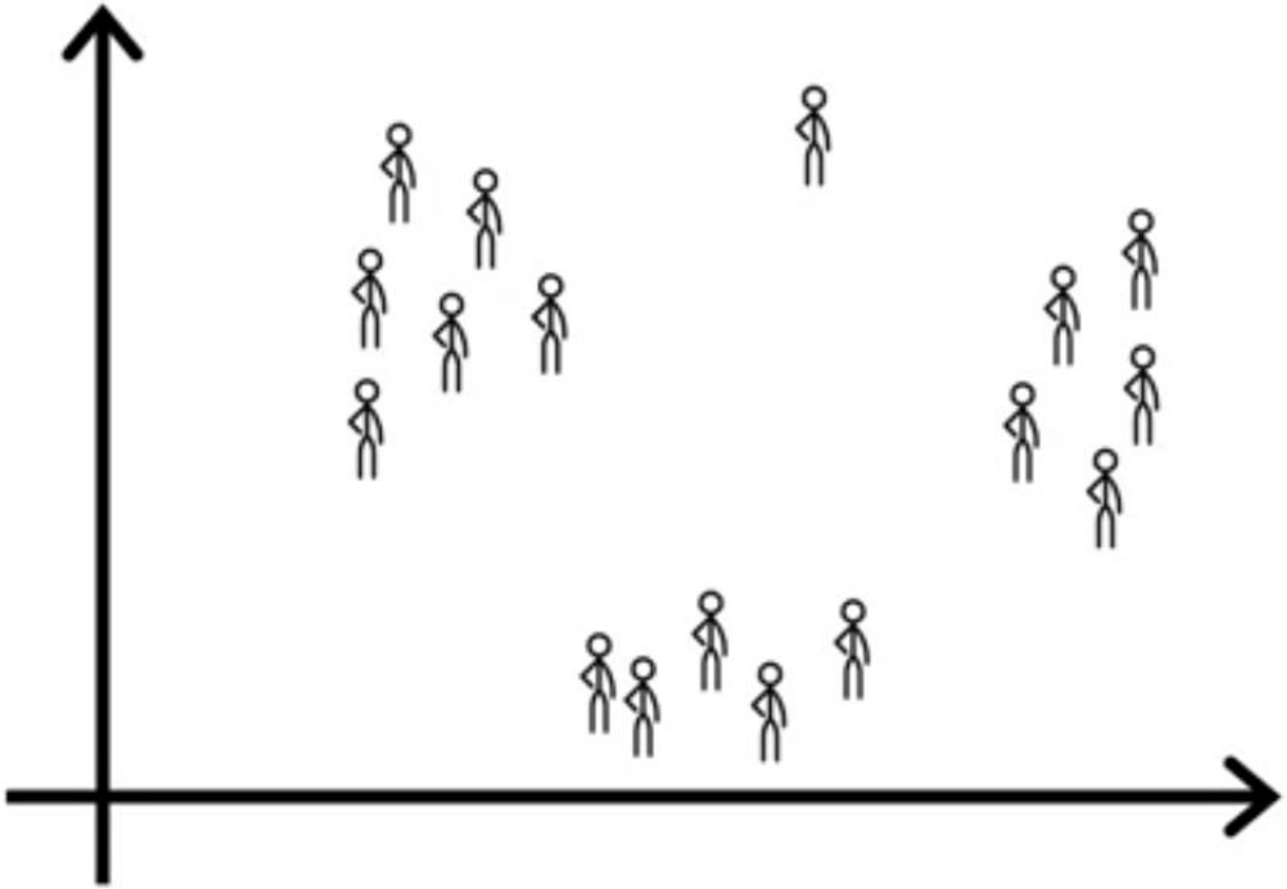


genes
(each row)

individuals
(each column)

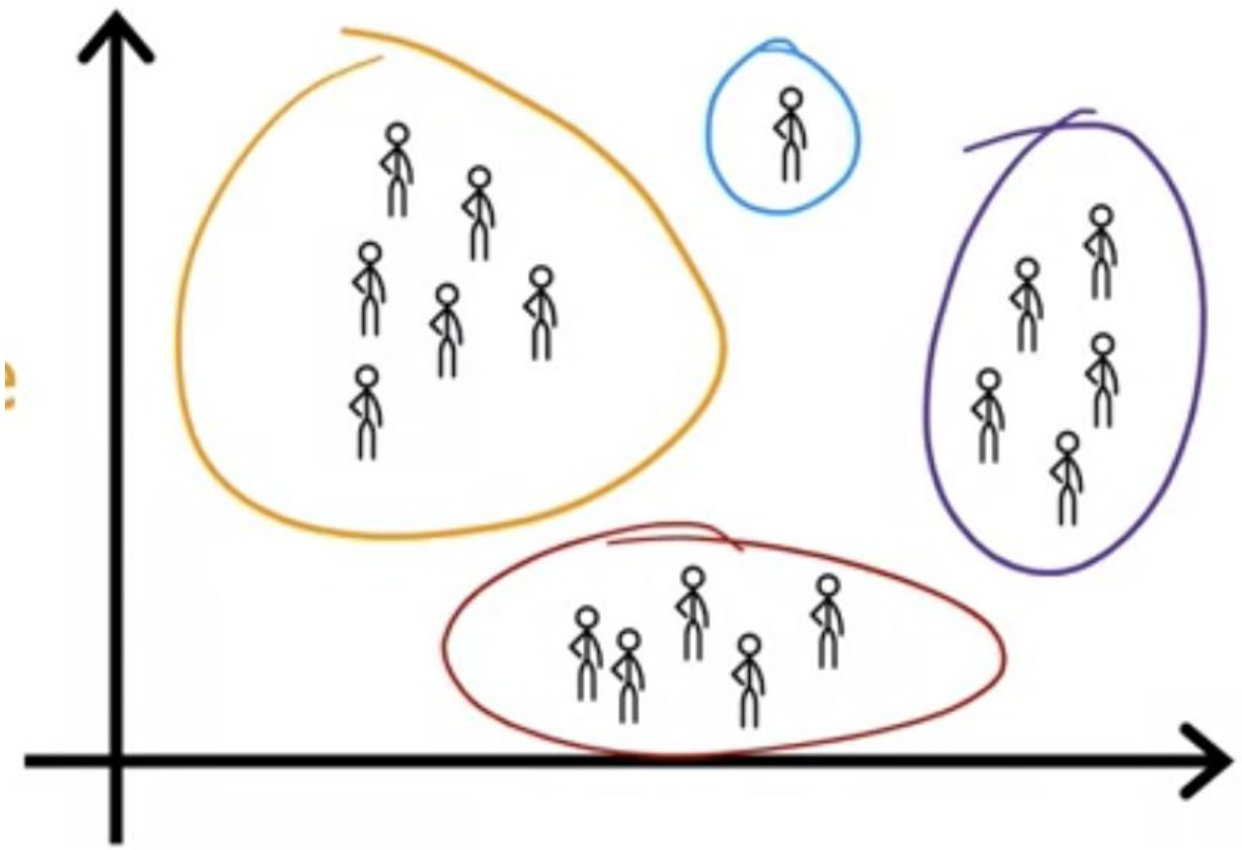
Credit: Andrew Ng, Machine Learning

Clustering: Grouping customers



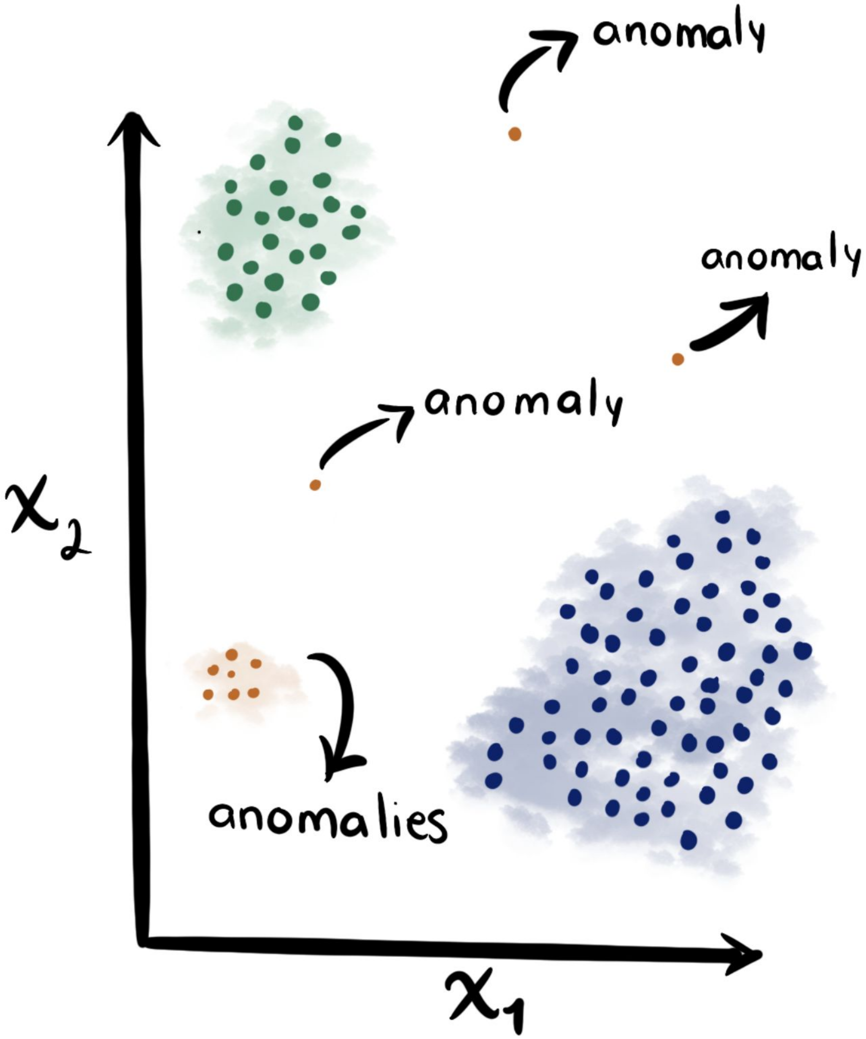
Credit: Andrew Ng, Machine Learning

Grouping Customers



Credit: Andrew Ng, Machine Learning

Anomaly Detection



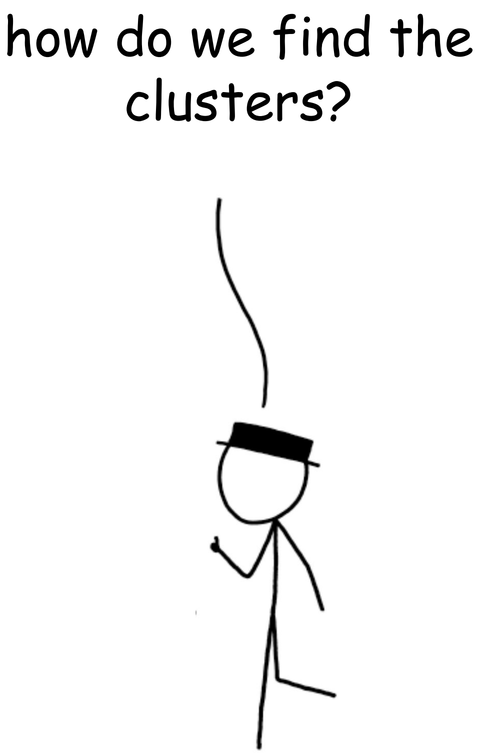
Credit: Anomaly Detection

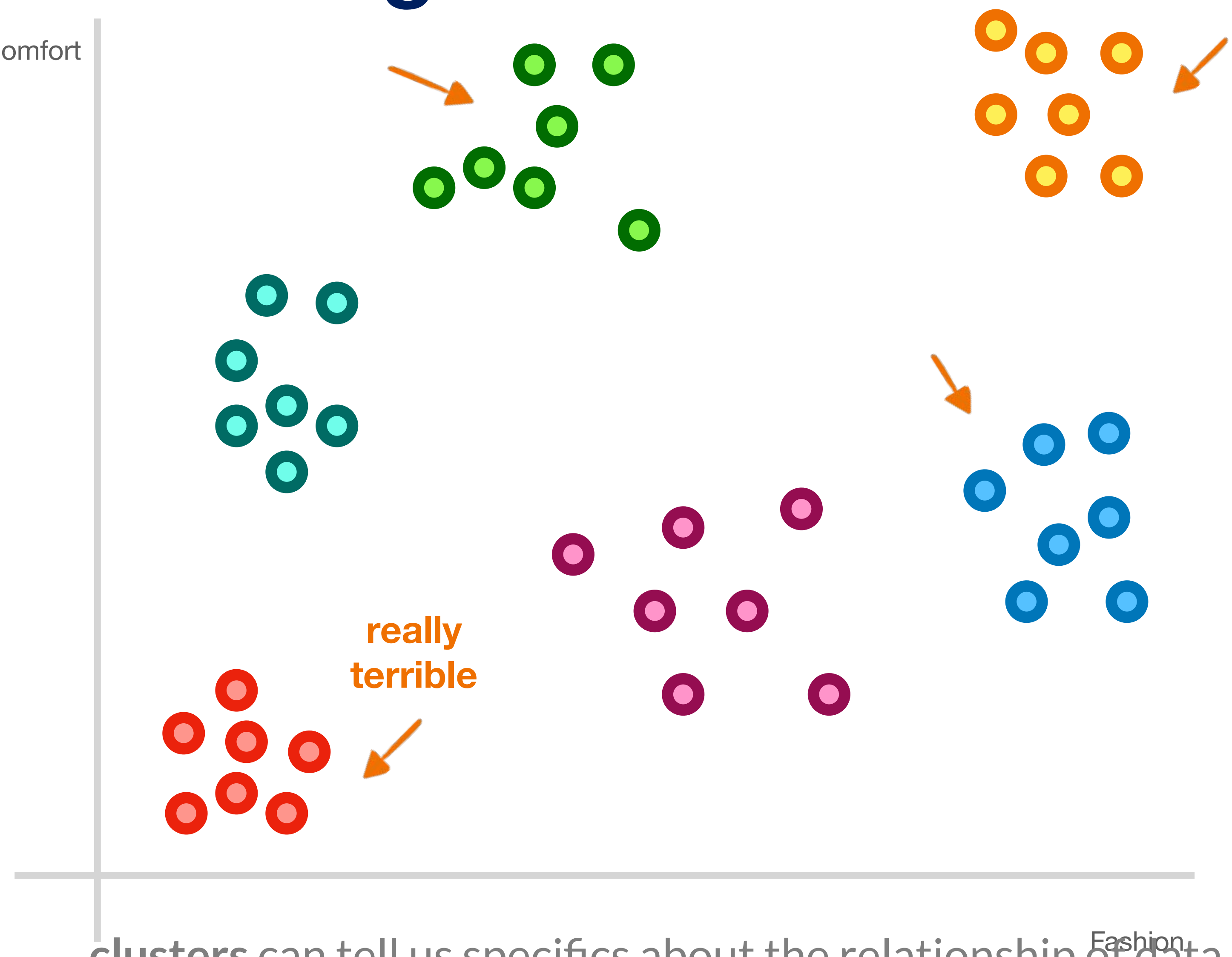
Unsupervised Learning

Clustering

Dimension reduction

K-means clustering

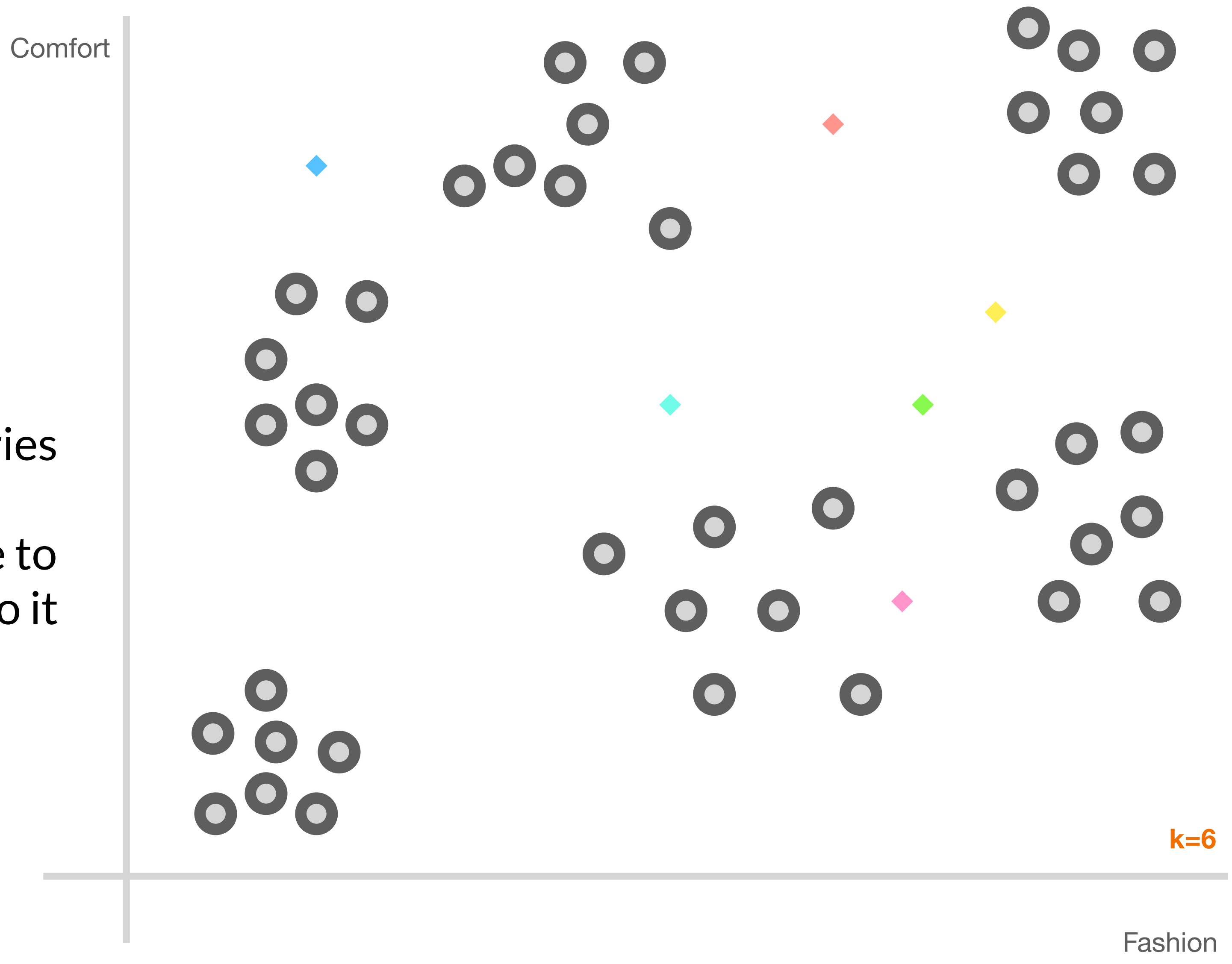




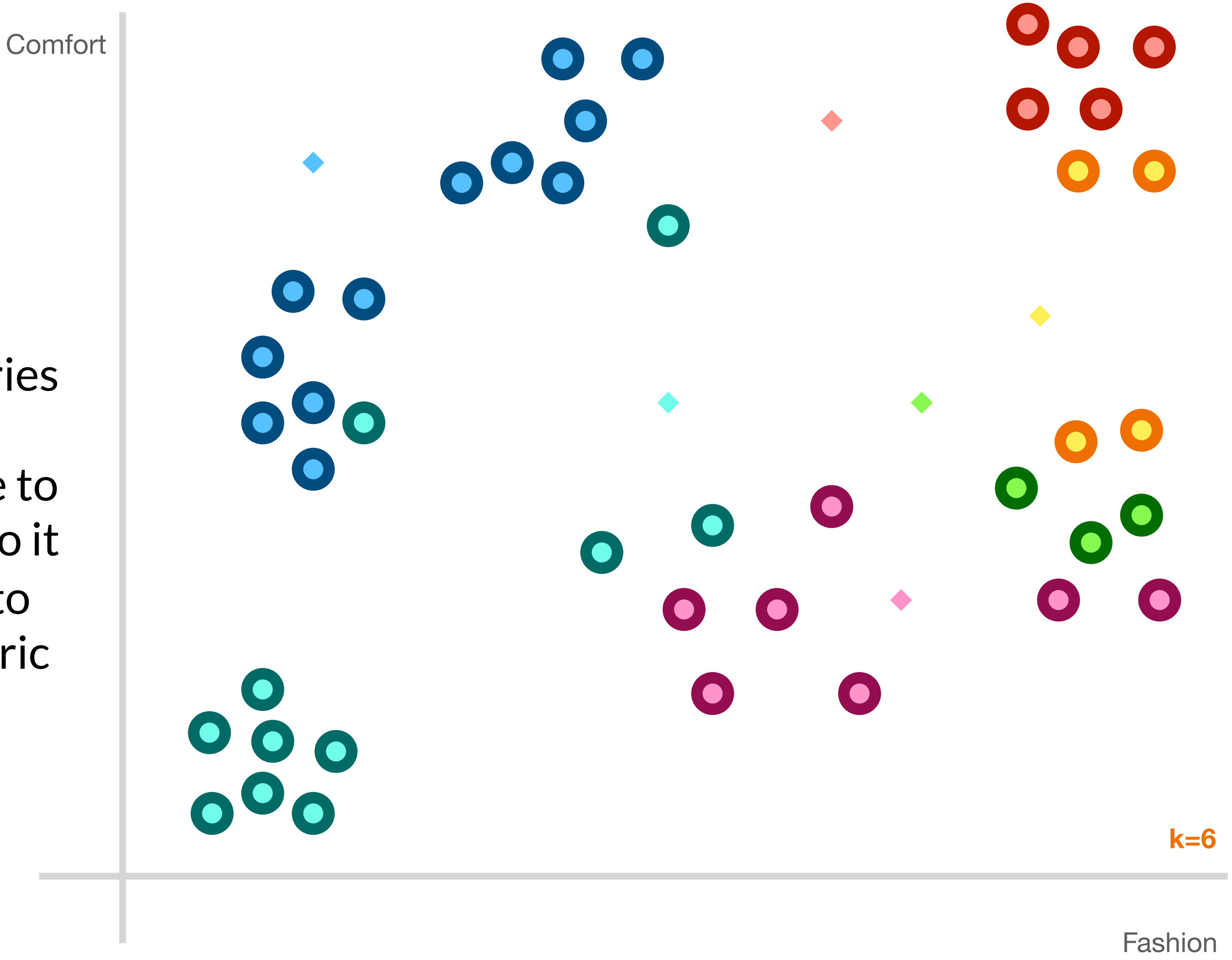
clusters can tell us specifics about the relationship of data

...even if they are unlabeled! → **unsupervised learning!**

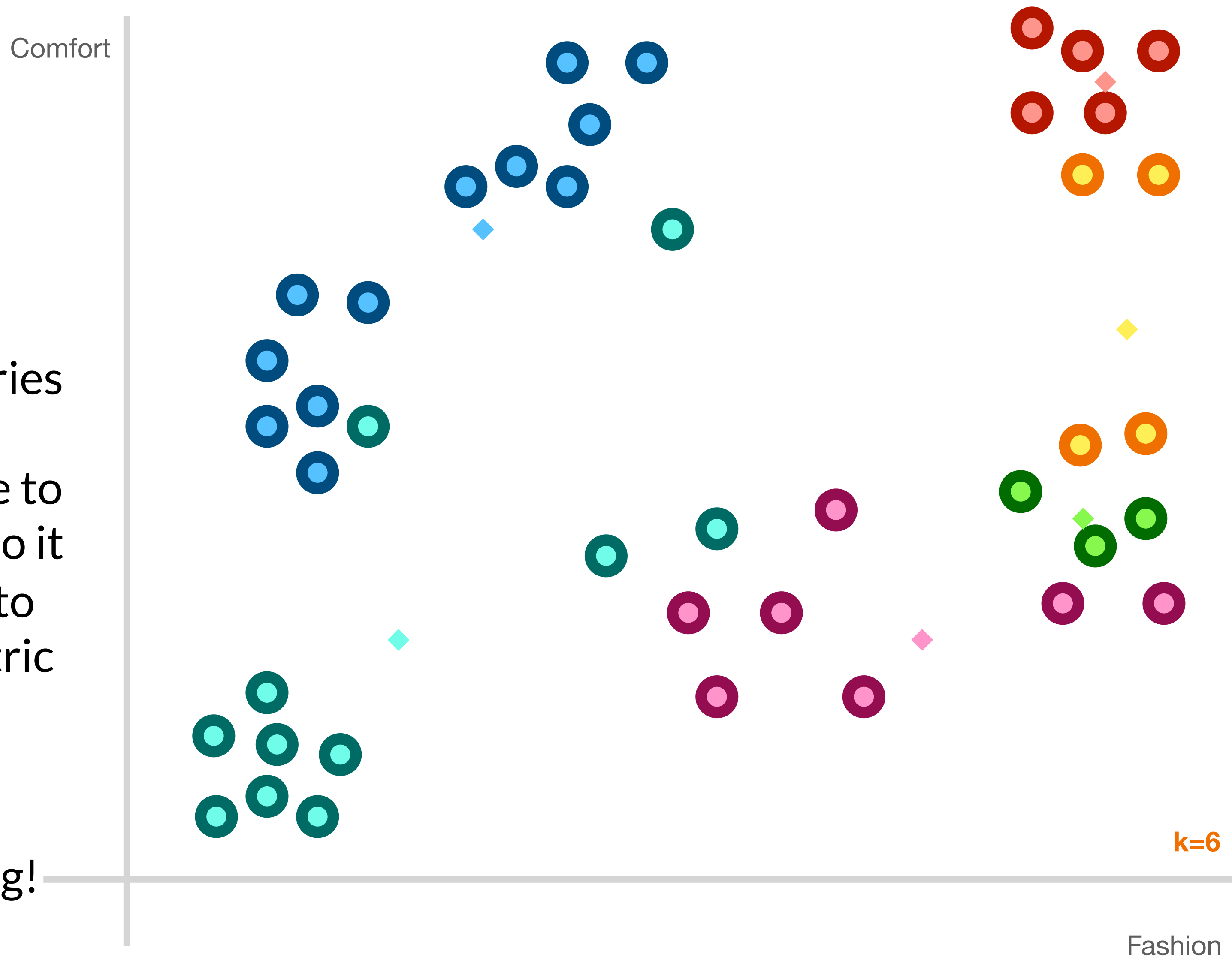
1. pick a K-number of clusters
2. randomly pick a series of “**centroids**”
3. assign each particle to the **centroid** closest to it



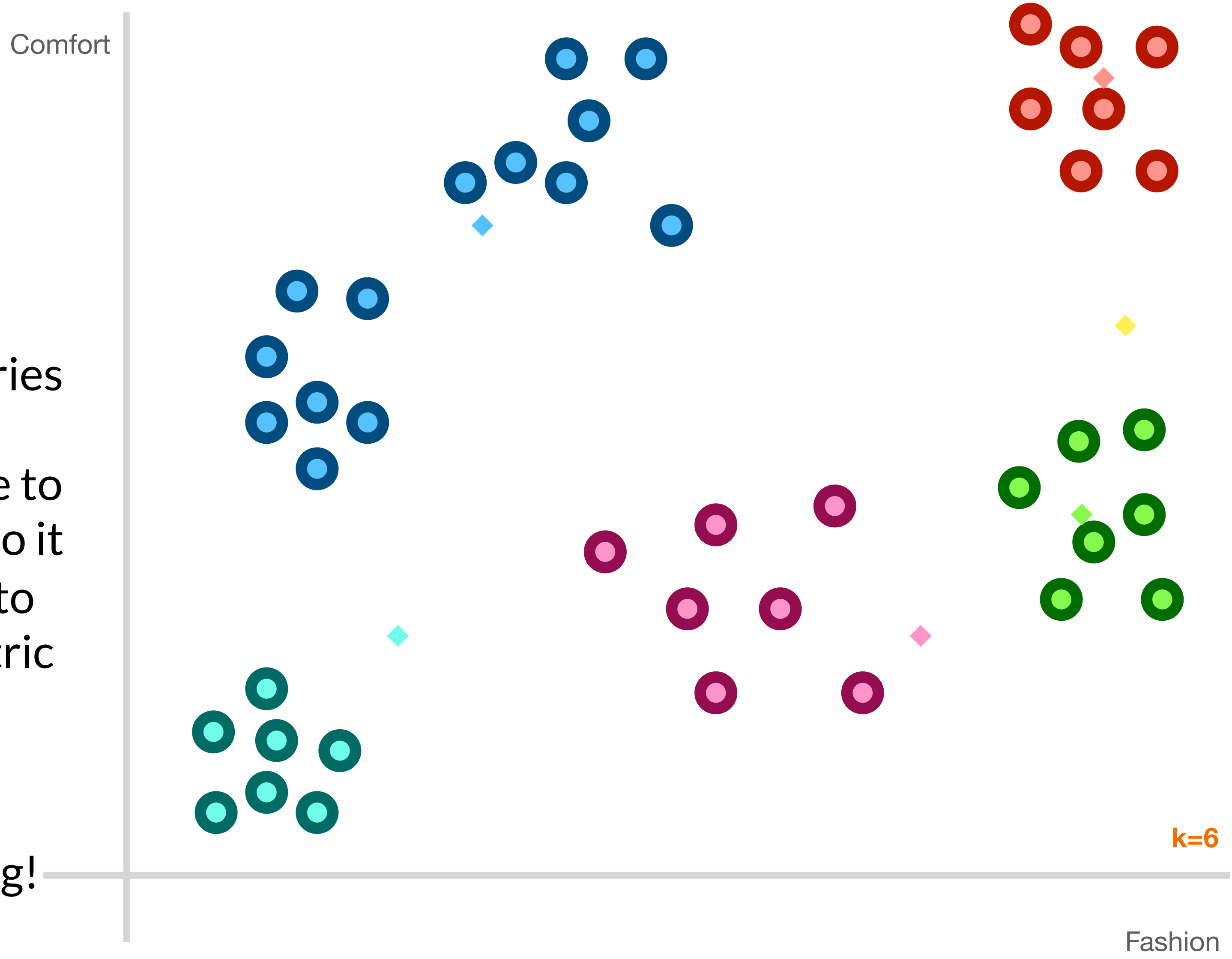
1. pick a K-number of clusters
2. randomly pick a series of “**centroids**”
3. assign each particle to the **centroid** closest to it
4.move the **centroid** to the weighted geometric center of samples assigned to it



1. pick a K-number of clusters
2. randomly pick a series of “**centroids**”
3. assign each particle to the **centroid** closest to it
4.move the **centroid** to the weighted geometric center of samples assigned to it
5. Repeat 3-4 until centroids stop moving!



1. pick a K-number of clusters
2. randomly pick a series of "**centroids**"
3. assign each particle to the **centroid** closest to it
4.move the **centroid** to the weighted geometric center of samples assigned to it
5. Repeat 3-4 until centroids stop moving!



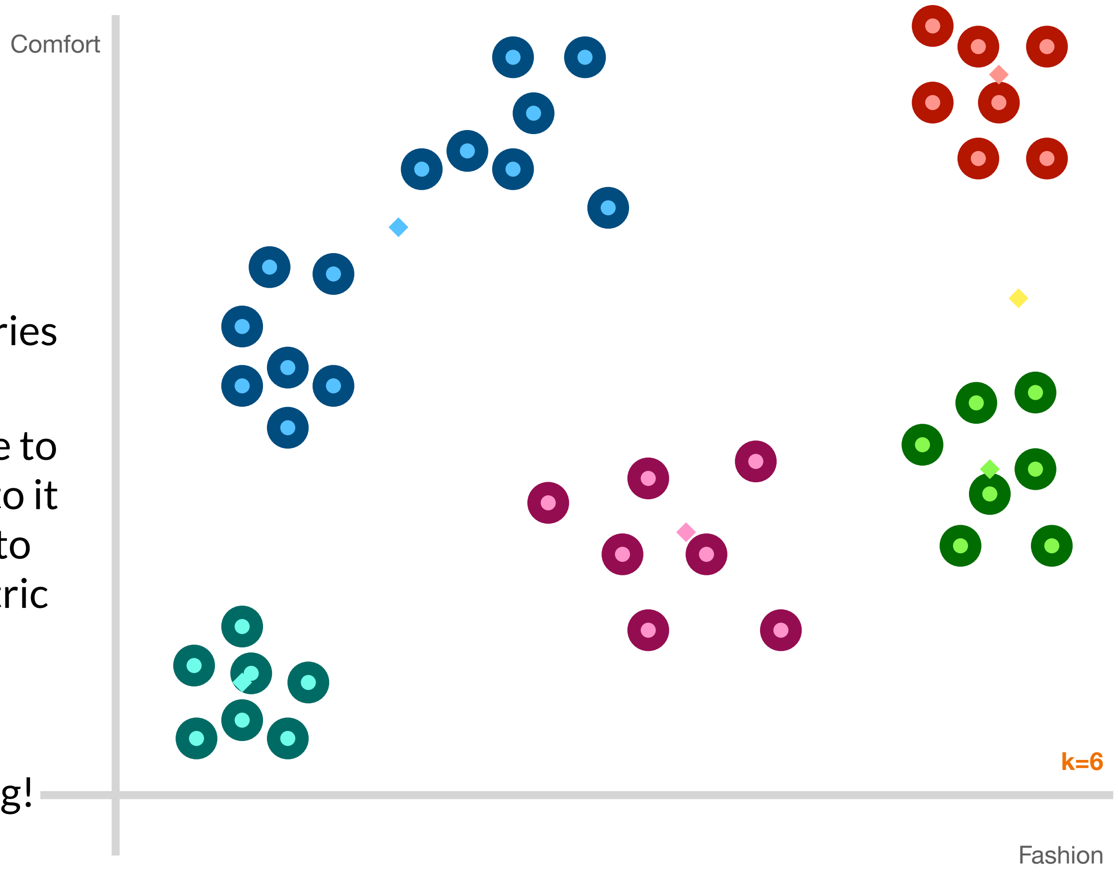
1. pick a K-number of clusters

2. randomly pick a series of "**centroids**"

3. assign each particle to the **centroid** closest to it

4.move the **centroid** to the weighted geometric center of samples assigned to it

5. Repeat 3-4 until centroids stop moving!



Did we get back the same clusters?

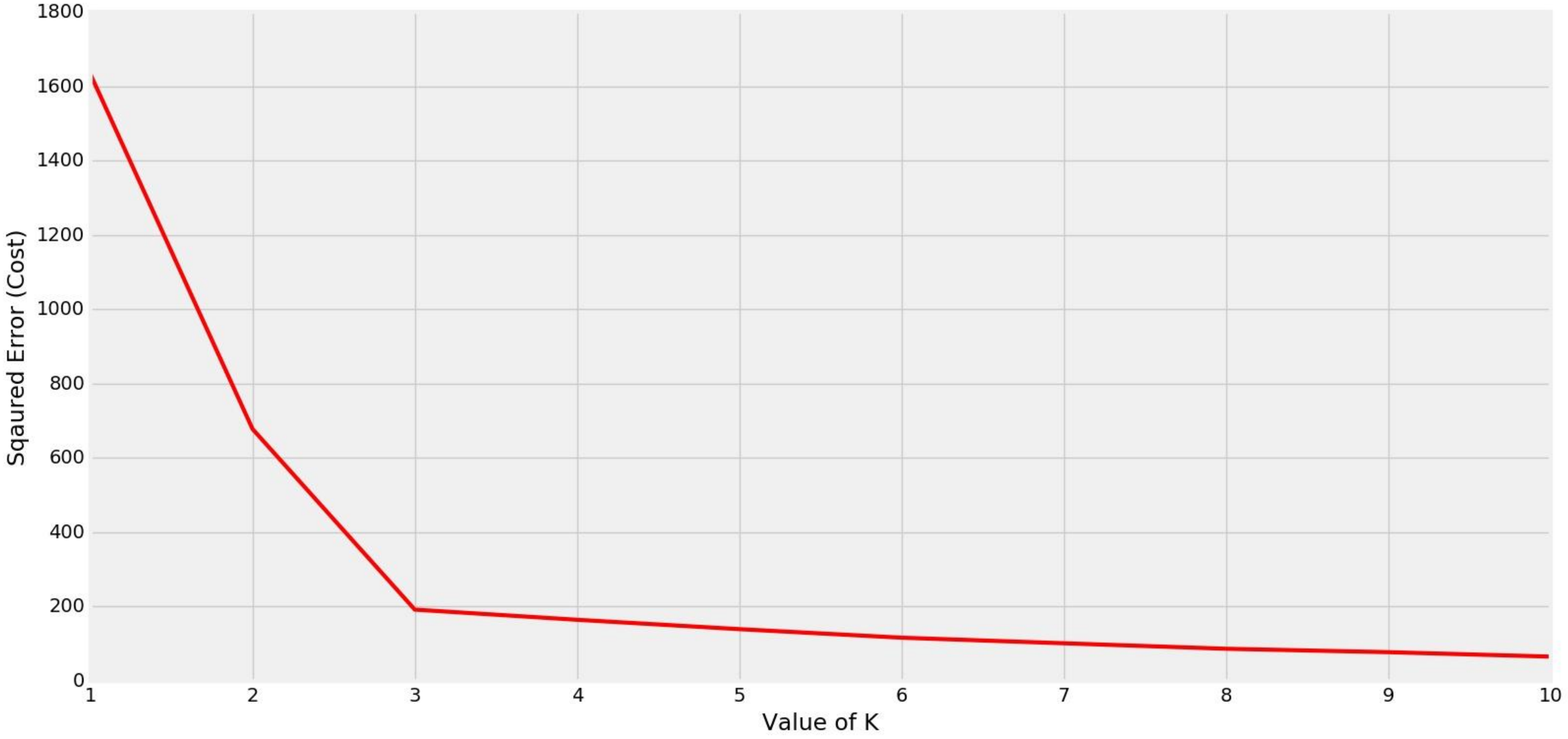
Nope. And that's OK.

Did we get back the same clusters?

Nope. And that’s OK.

K-means is an *indeterministic* algorithm—it has built-in randomness

Evaluation and Choosing K



Unsupervised Learning

Clustering

Dimension reduction

Unsupervised Learning

Why Dimension reduction?

Motivation for Dimension Reduction

Complex systems often must be modeled with large datasets, having dozens to millions of columns.

Often, several columns can be adding similar information to the model. So, there is a certain level of *redundancy.*

A

Individual	Height (cm)	Weight (kg)	Income ($)	Number of Children
Person A	165	65	60,000	2
Person B	168	63	100,000	5
Person C	159	82	50,000	1
Person D	183	68	90,000	4
Person E	187	87	110,000	5
Person F	189	89	95,000	4

Four dimensions; can't even be graphed!

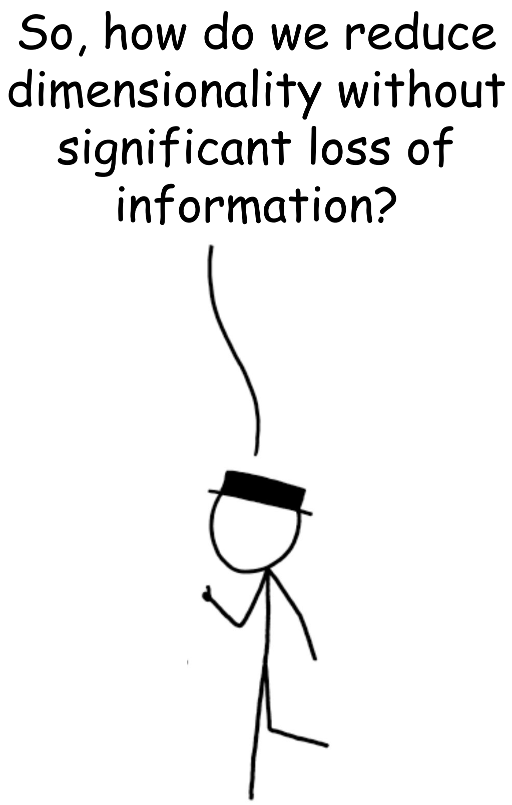
Motivation for Dimension Reduction

Individual	Height (cm)	Weight (kg)	Income ($)	Number of Children
Person A	165	65	60,000	2
Person B	168	63	100,000	5
Person C	159	82	50,000	1

What if I have a lot of features, but not a lot of samples?

Motivation for Dimension Reduction

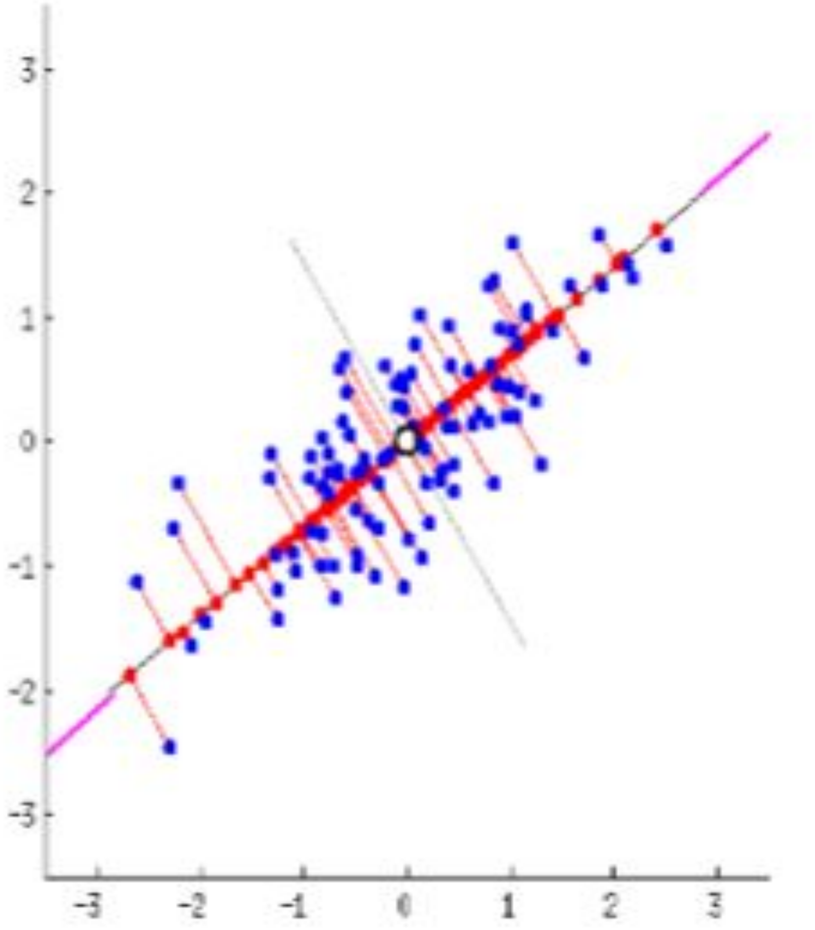
Individual	Height (cm)	Weight (kg)	Income ($)	Number of Children
Person A	165	65	60,000	2
Person B	168	63	100,000	5
Person C	159	82	50,000	1
Person D	183	68	90,000	4
Person E	187	87	110,000	5
Person F	189	89	95,000	4


So, how do we reduce dimensionality without significant loss of information?

Enter…

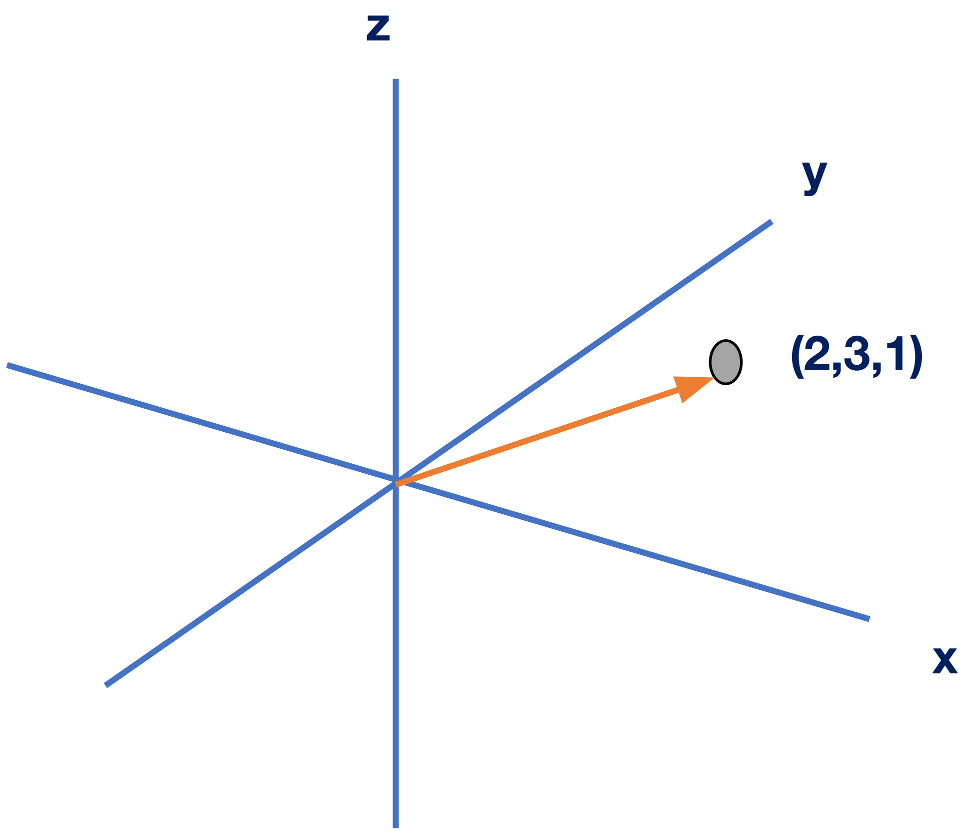
Principal Component Analysis

Principle Component Analysis



transform high-dimensional data into a **new coordinate** system, where the new features (principal components) are **orthogonal** (uncorrelated) and sorted in decreasing order of **variance**.

Exploring Dimensions and Basis Vectors

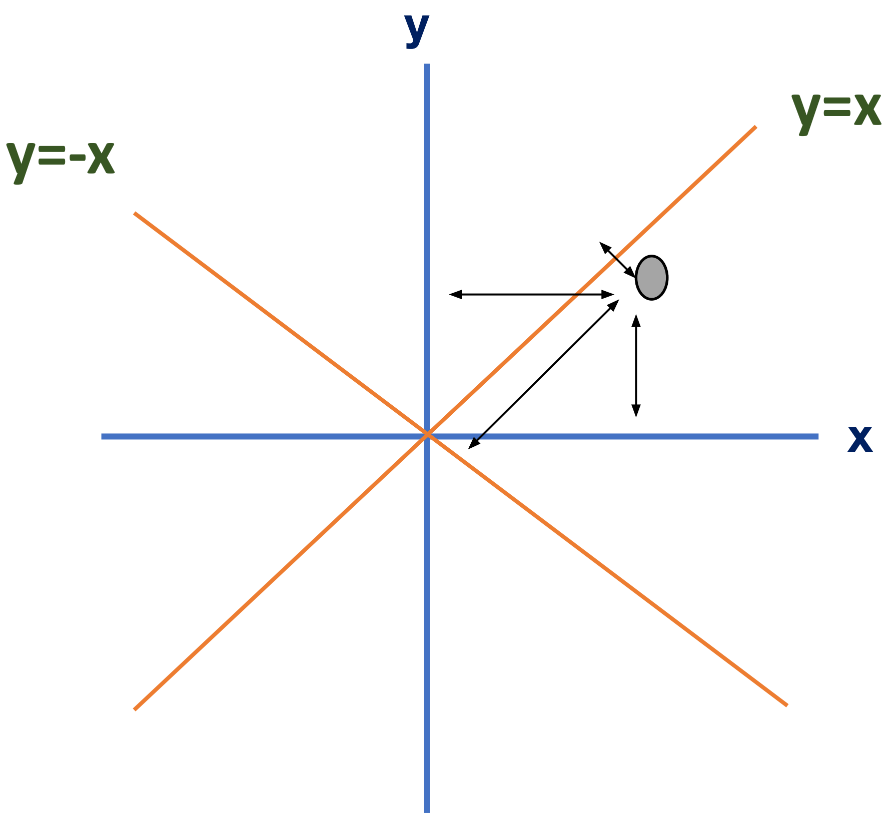


(2,3,1) is a datapoint.

$\begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$ is the vector to said datapoint.

Dimension = # of features

Exploring Dimensions and Basis Vectors

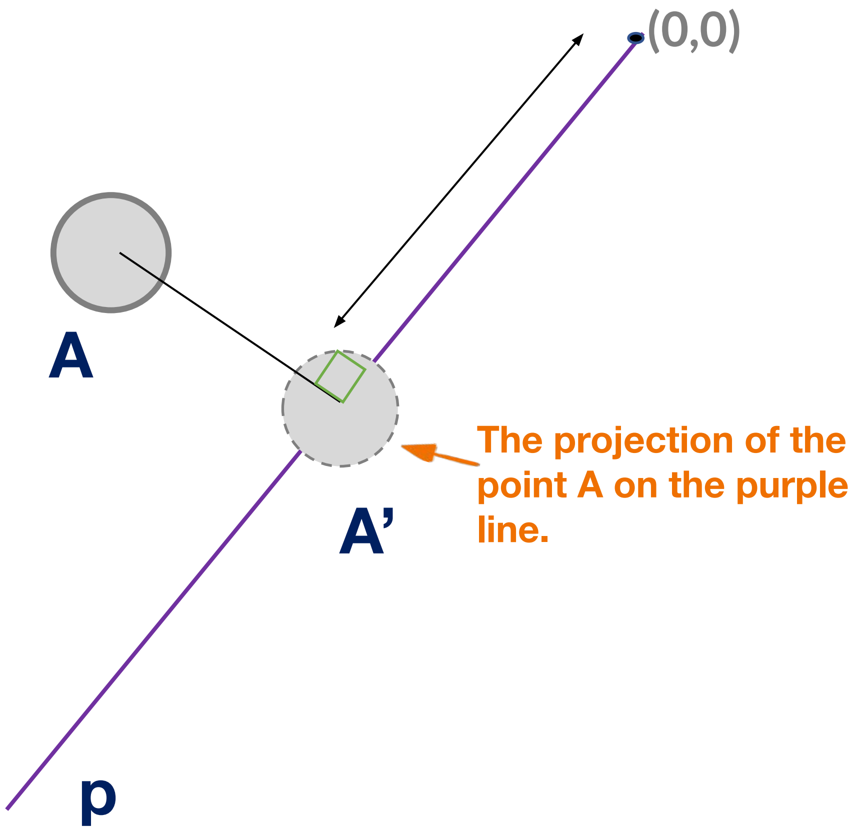


This gray point can be expressed as 3 blocks on x axis and 2 blocks on the y axis.

It can also be expressed as 1 block on y = -x and 3 blocks on y=x

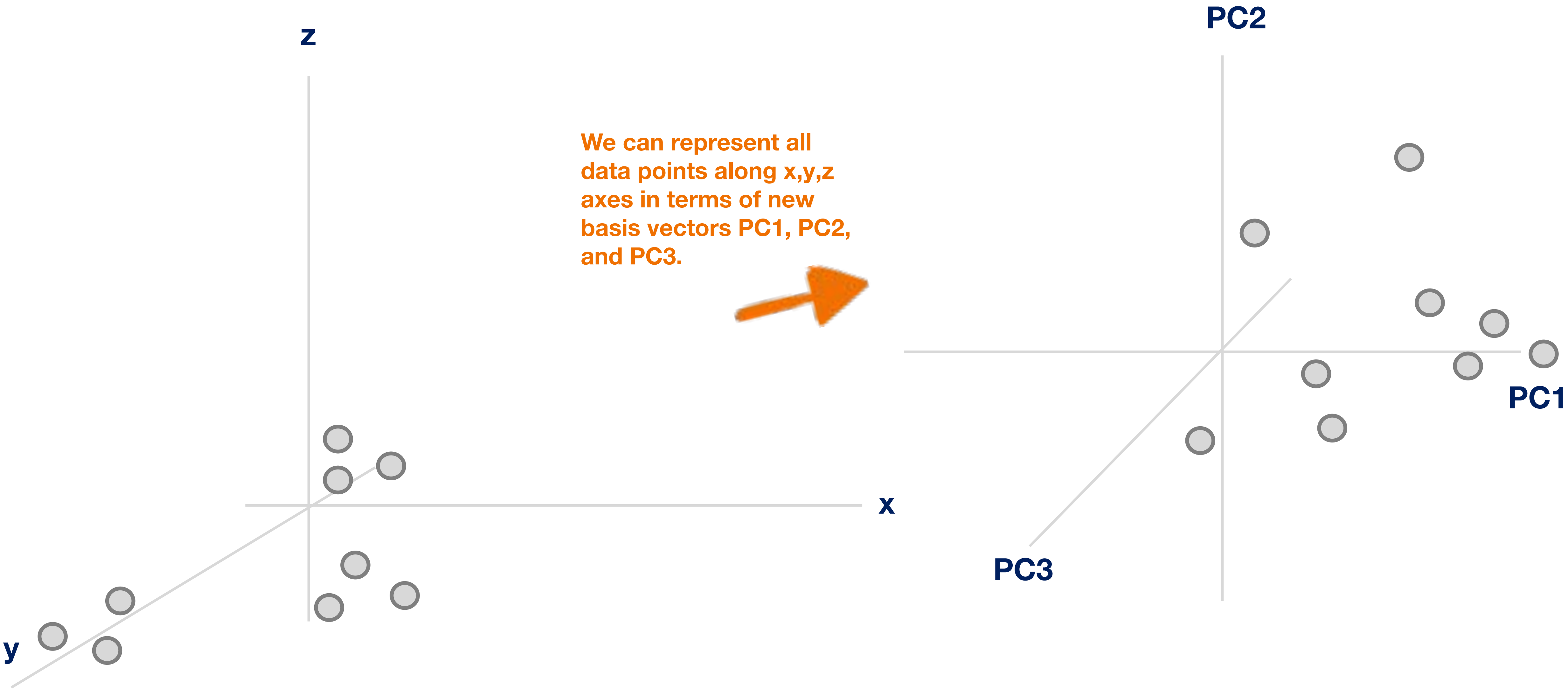
Projection

The projection (A’) of a point A on a particular line p is the point such that the line AA’ is perpendicular to p.



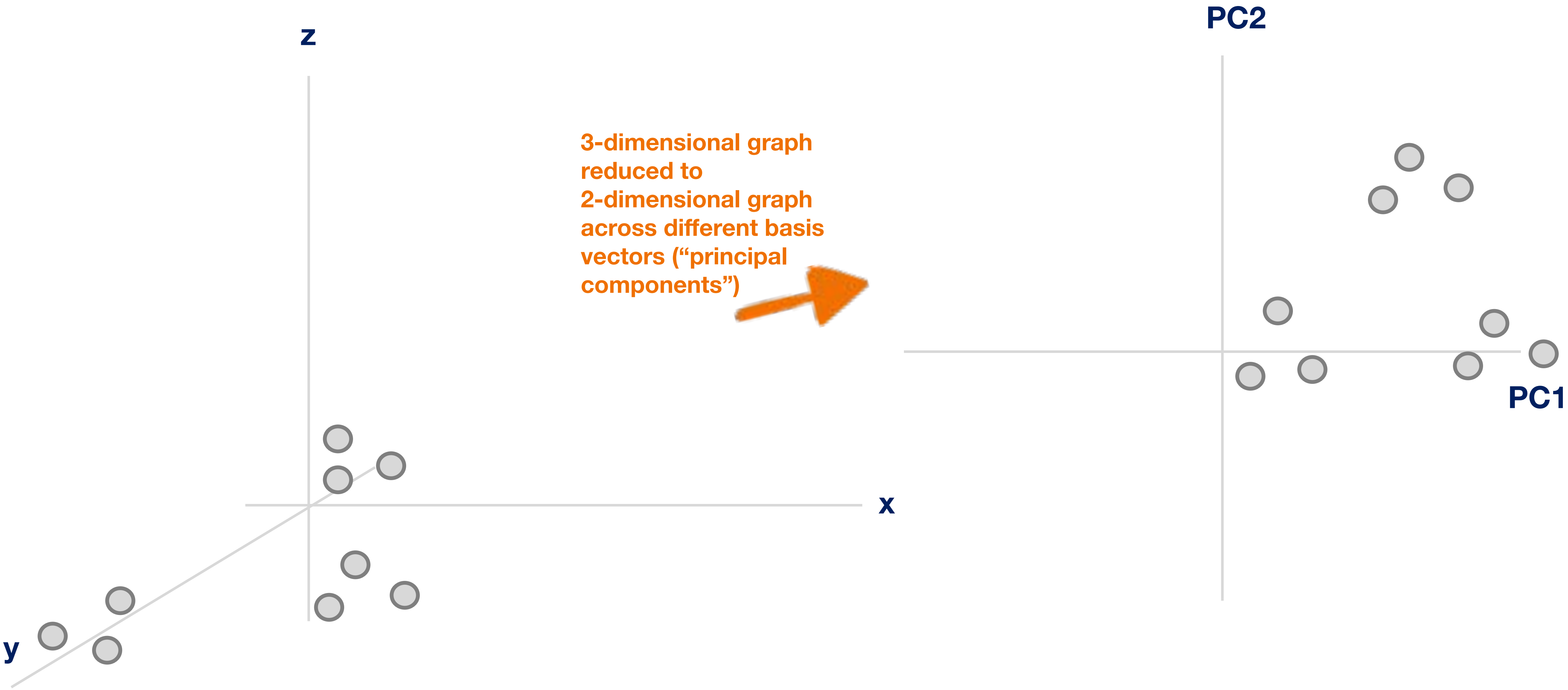
The projection of the point A on the purple line.

Principal Component Analysis



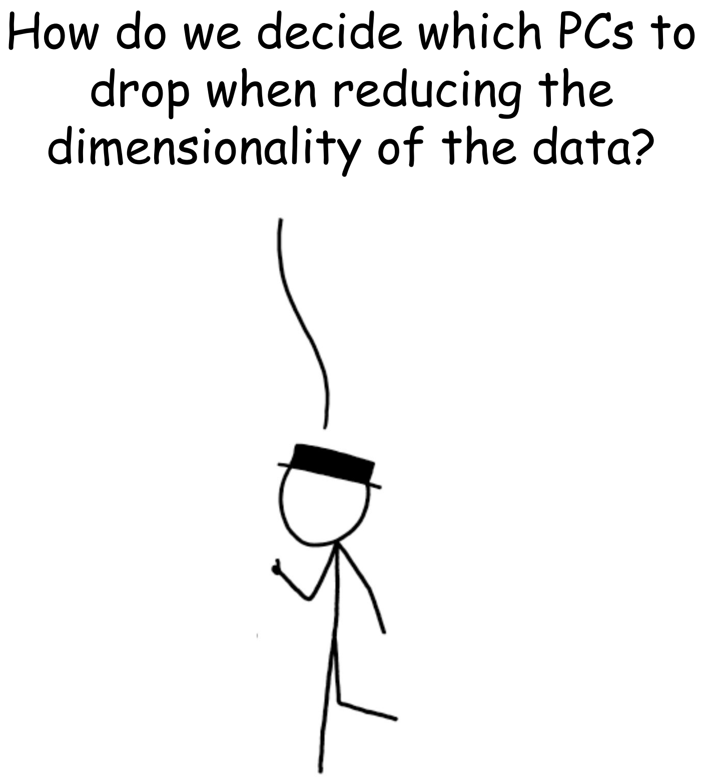
We can represent all data points along x,y,z axes in terms of new basis vectors PC1, PC2, and PC3.

Principal Component Analysis



3-dimensional graph reduced to 2-dimensional graph across different basis vectors (“principal components”)

Principal Component Analysis


How do we decide which PCs to drop when reducing the dimensionality of the data?

Principal Components

Think of these as new axes that we are orienting our data across.

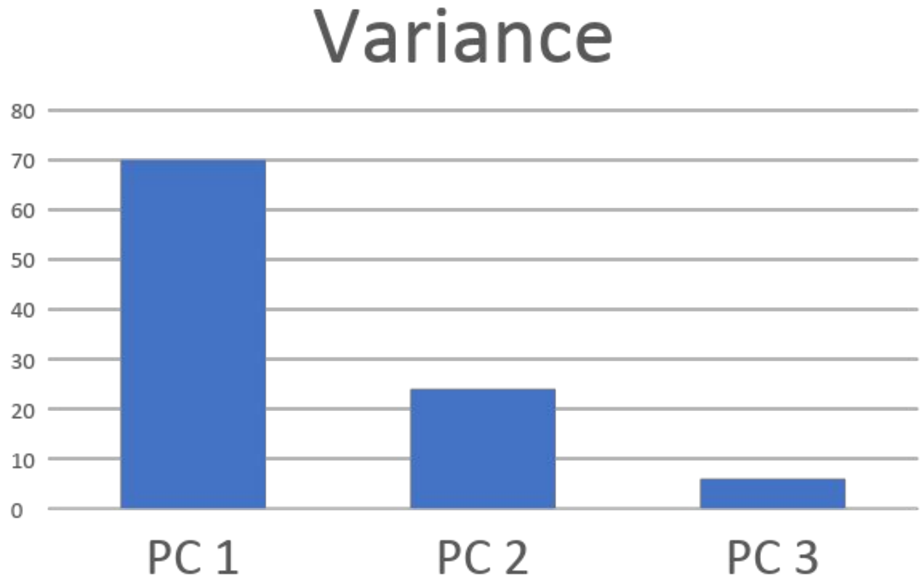
So instead of x,y, z, rather some linear combination of them.

They are done such that each principal component is uncorrelated with the others, so that translation across each component indicates different information. **So, they represent directions of maximal variance.**

This allows differences between data points to become more prominent

How do we decide which PCs to remove when reducing the dimensionality of the data?

Represents percentage of variance for each PC. Notice how PC1 has the most and it drops after that.

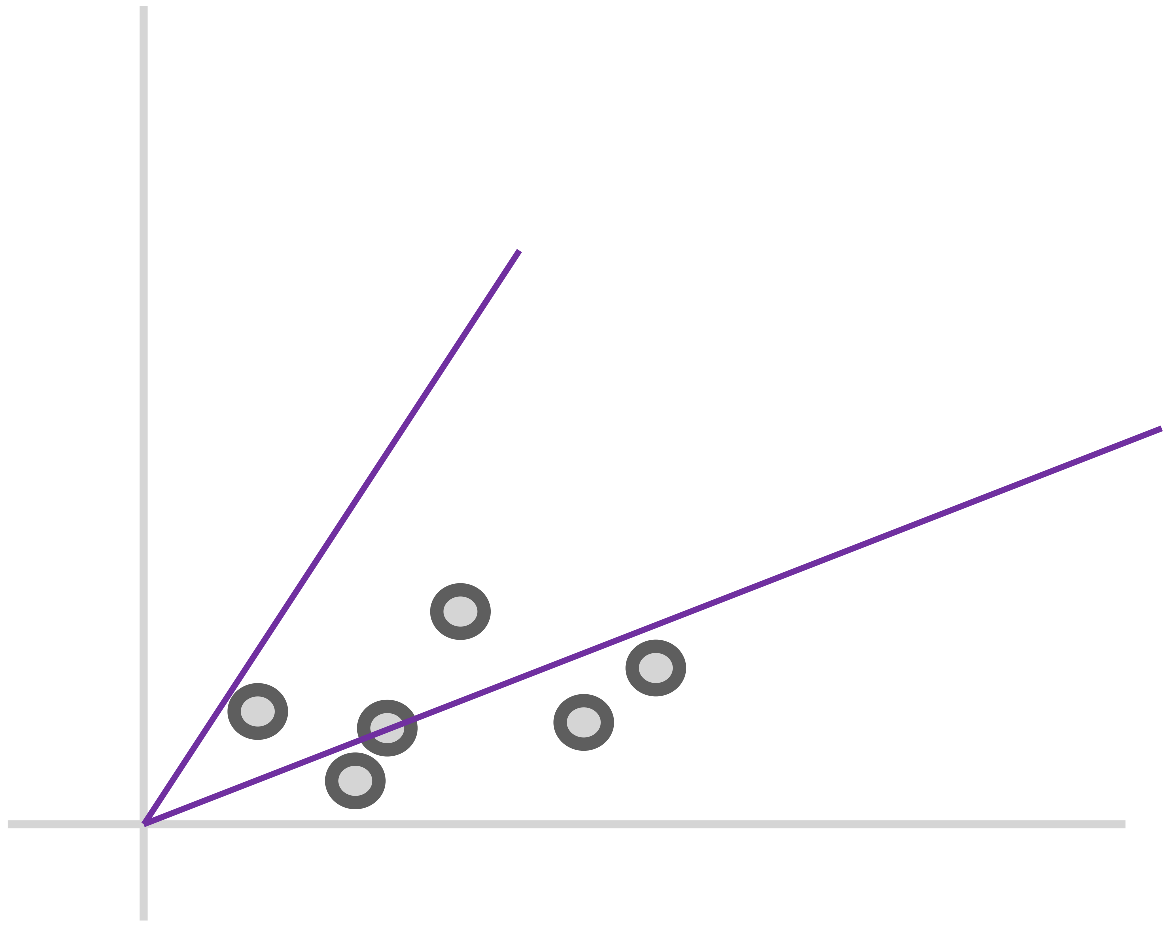


Since PC3 accounts for a very small percentage of overall variance, we can remove it. This is how PCA reduces dimensionality

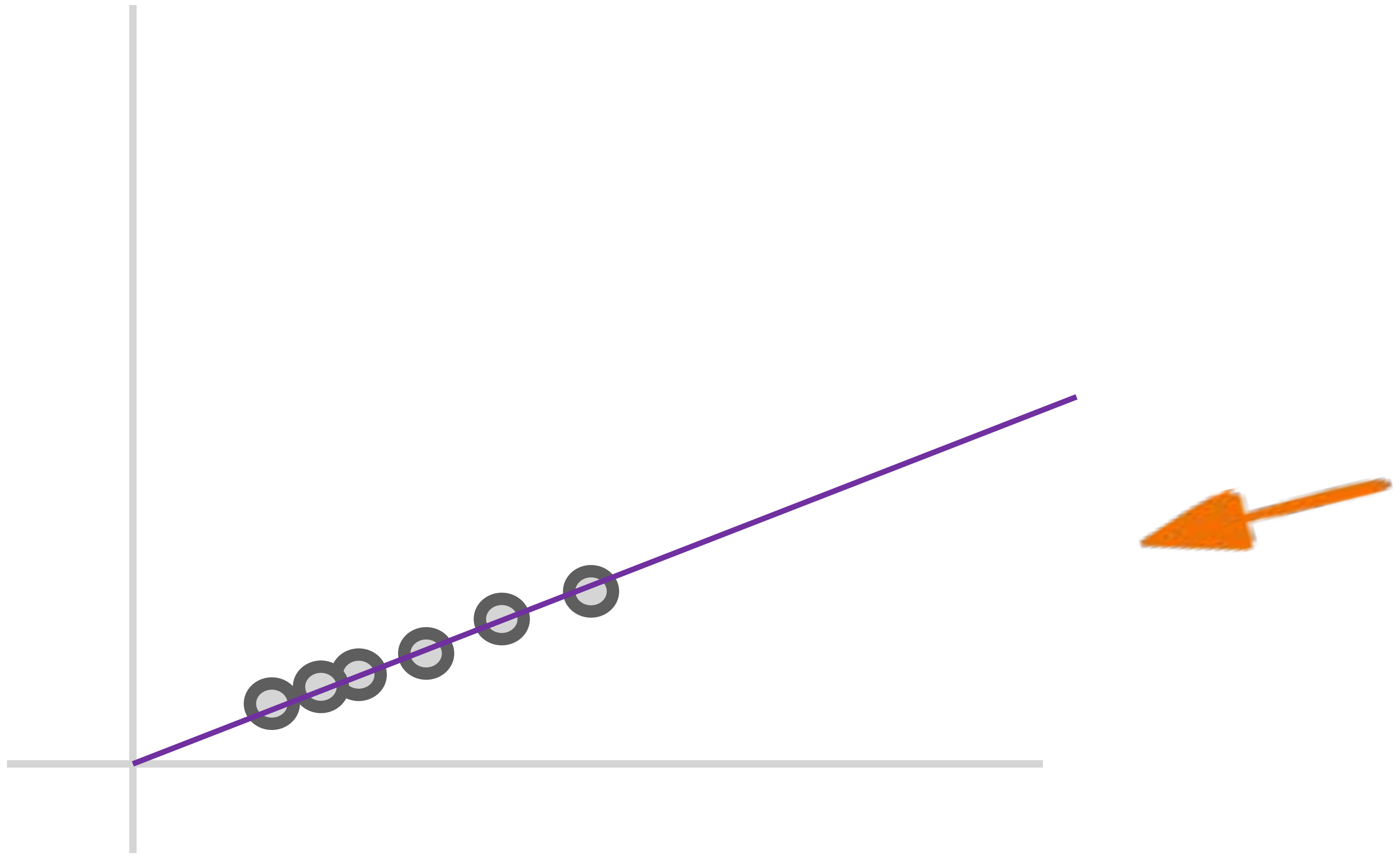
Principal Component Analysis

How do we decide the PCs?

Principal Component Analysis

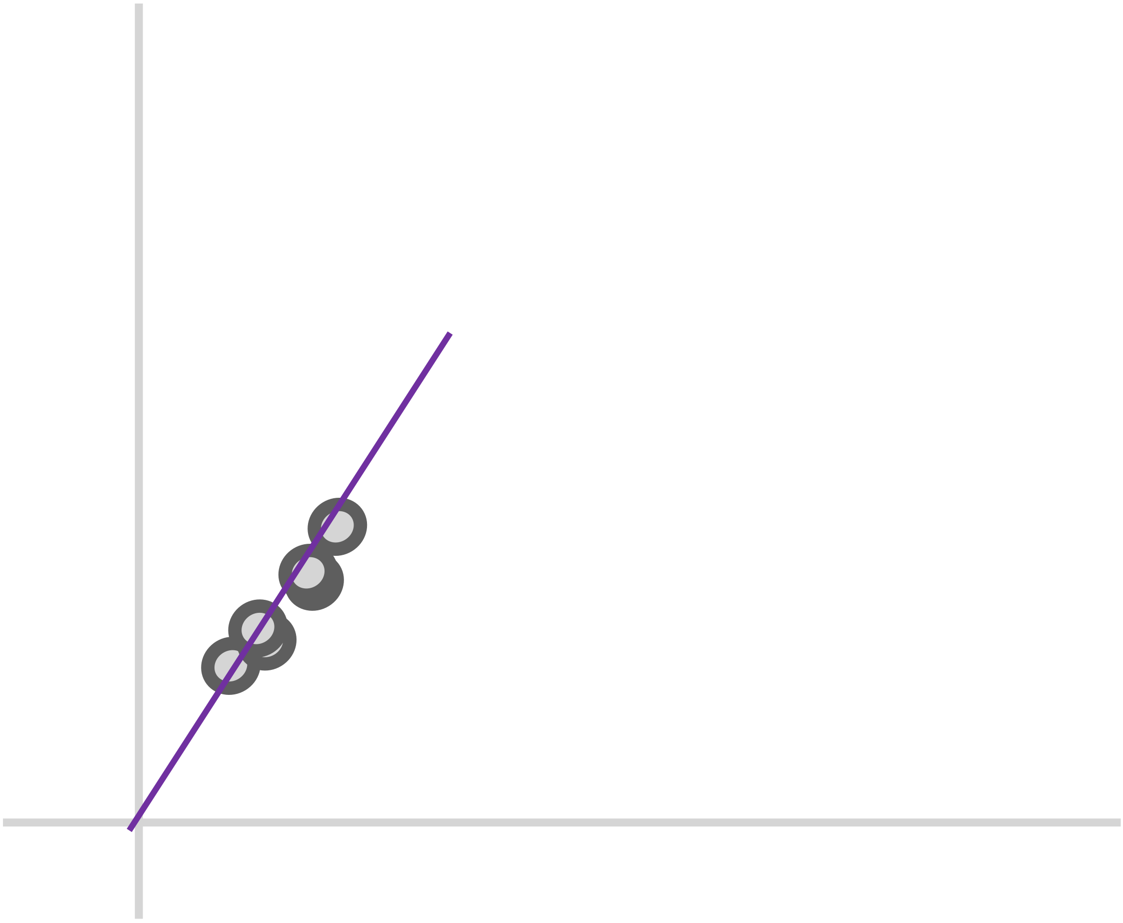


Principal Component Analysis



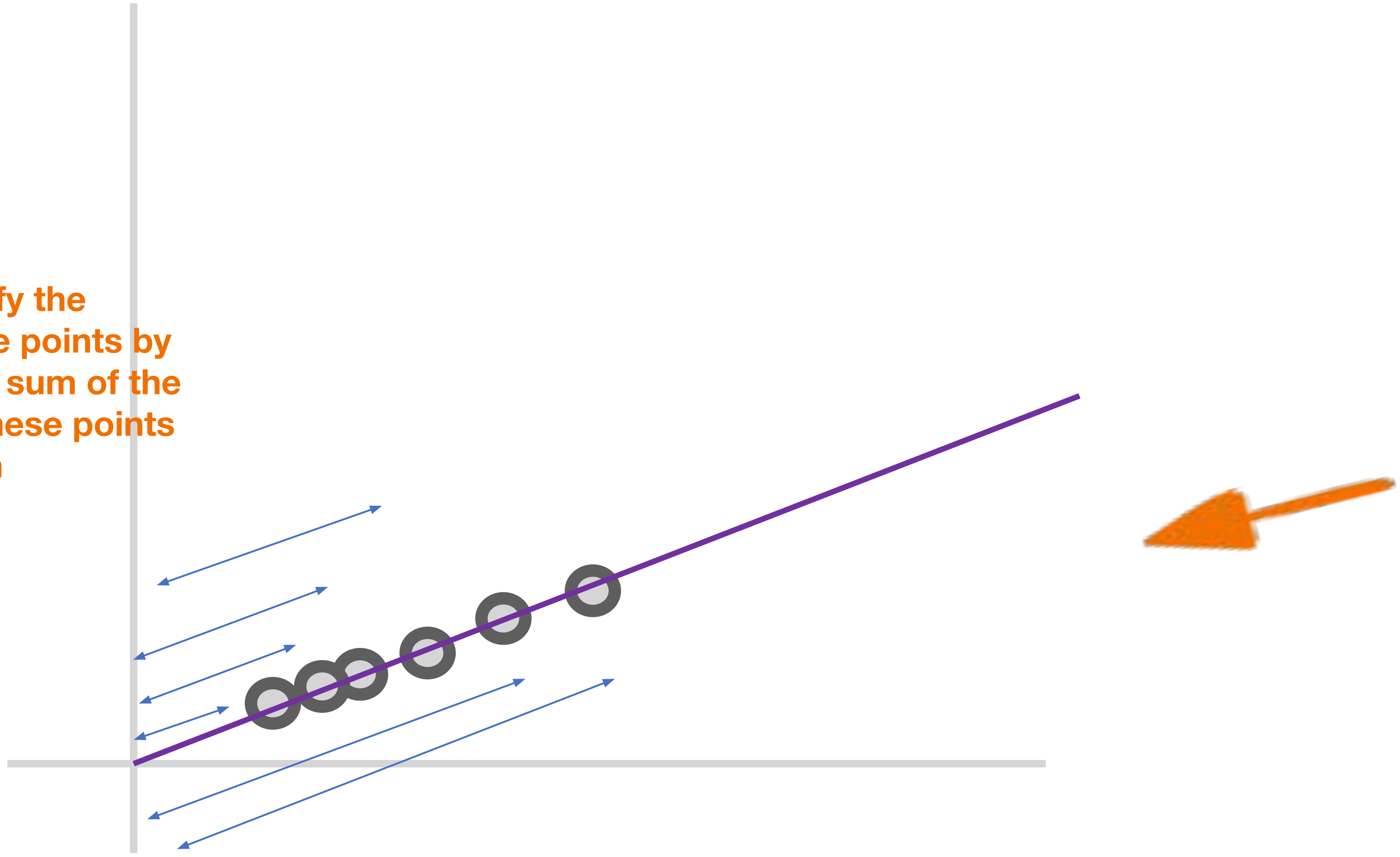
Notice how the points spread out from each other and from the origin.

Principal Component Analysis



Principal Component Analysis

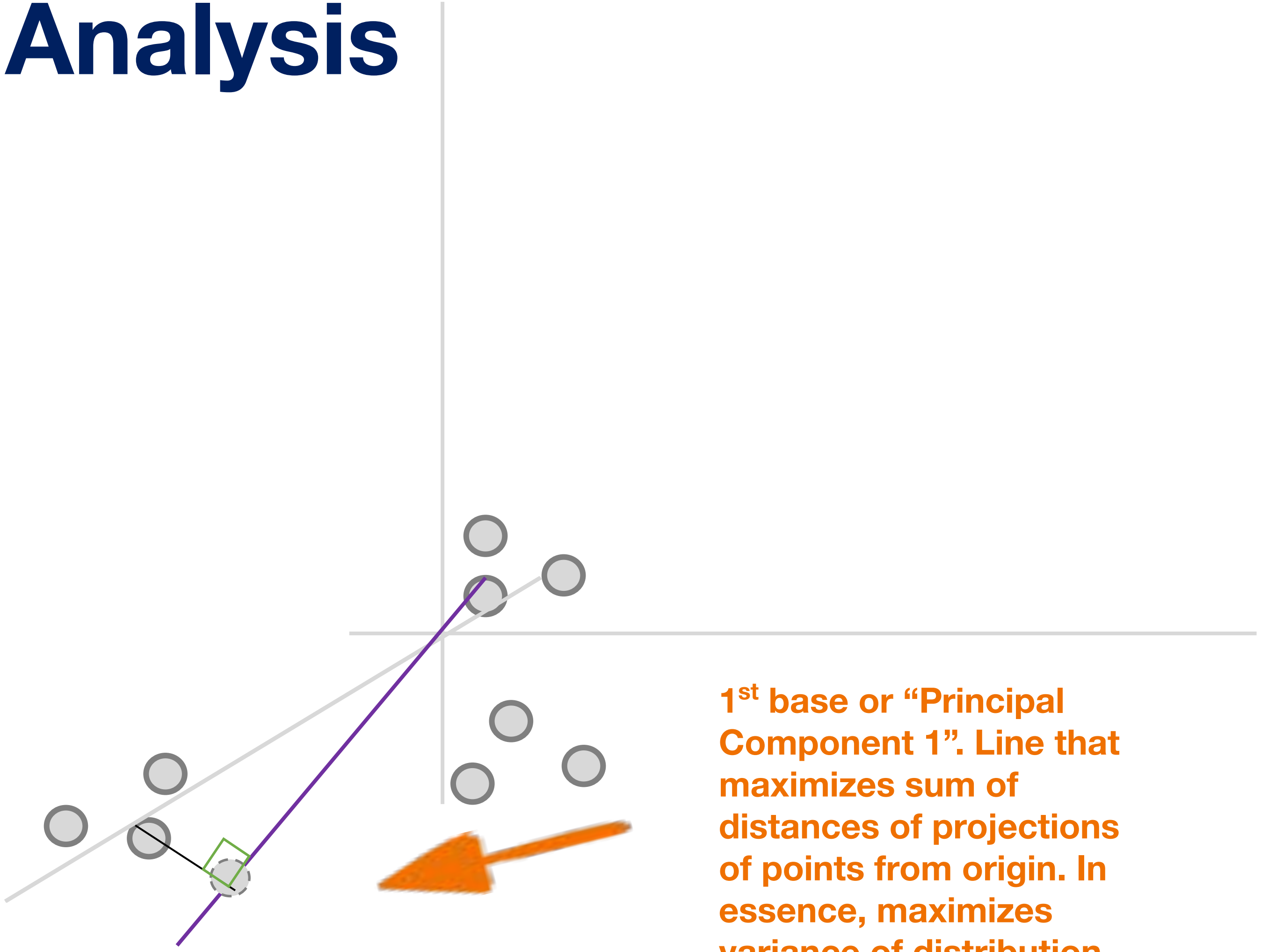
We can quantify the "spread" of the points by measuring the sum of the distances of these points from the origin



1st base or "Principal Component 1". Line that maximizes sum of distances of projections of points from origin. In essence, maximizes variance of distribution.

Principal Component Analysis

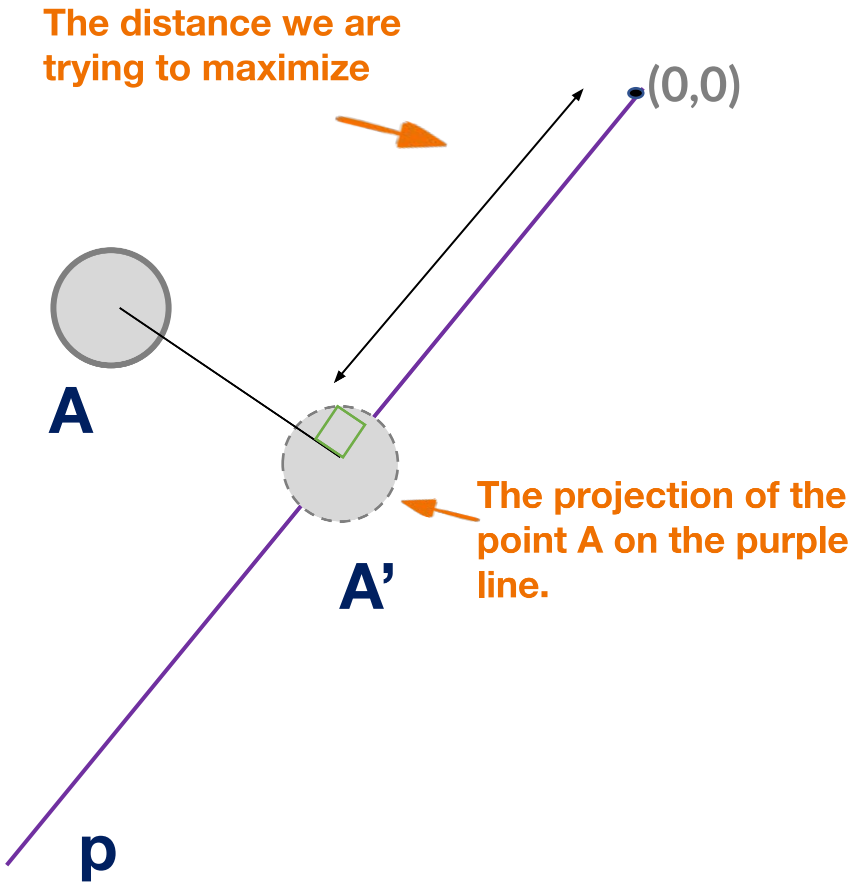
The degree to which a base aligns with the variance represents the amount of information separations along that basis can convey.



1st base or “Principal Component 1”. Line that maximizes sum of distances of projections of points from origin. In essence, maximizes variance of distribution.

Principal Component Analysis

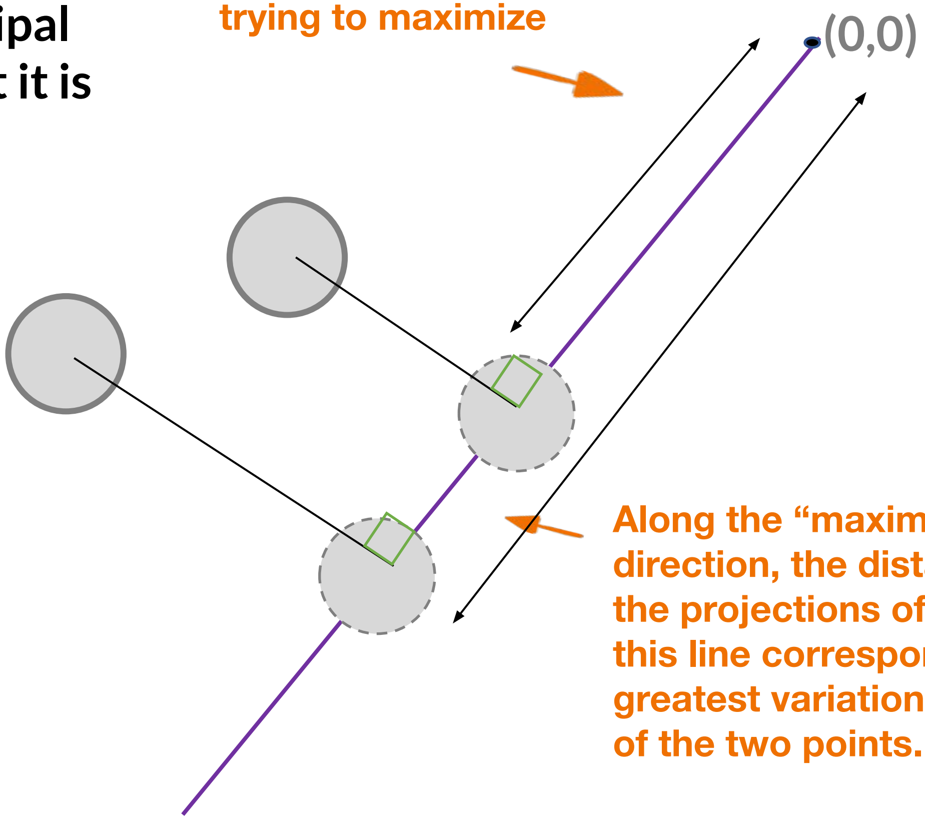
The projection (A’) of a point A on a particular line p is the point such that the line AA’ is perpendicular to p.



The distance we are trying to maximize
(0,0)
A
A’
The projection of the point A on the purple line.
p

Principal Component Analysis

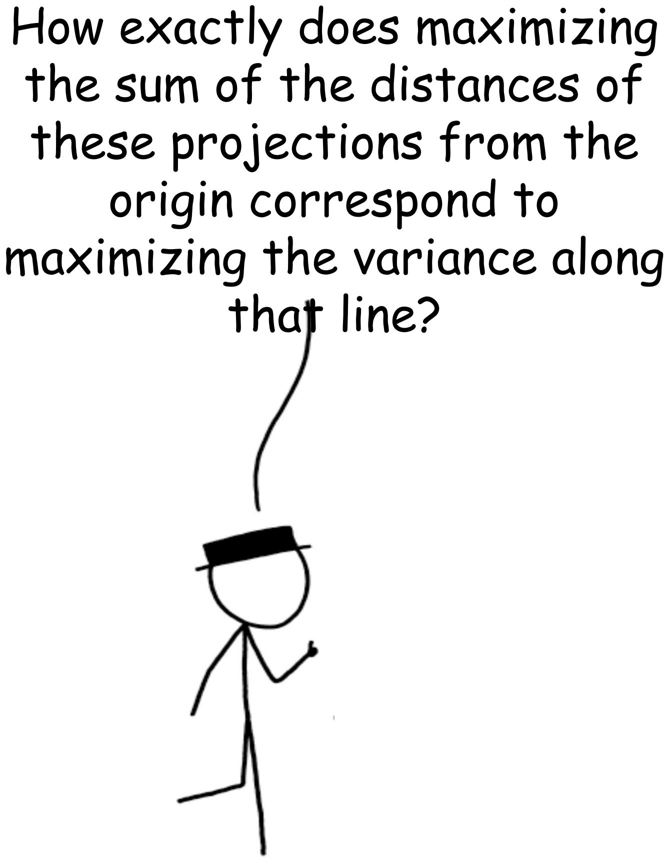
Idea behind this principal component line is that it is an axis along the “maximally variant” direction.



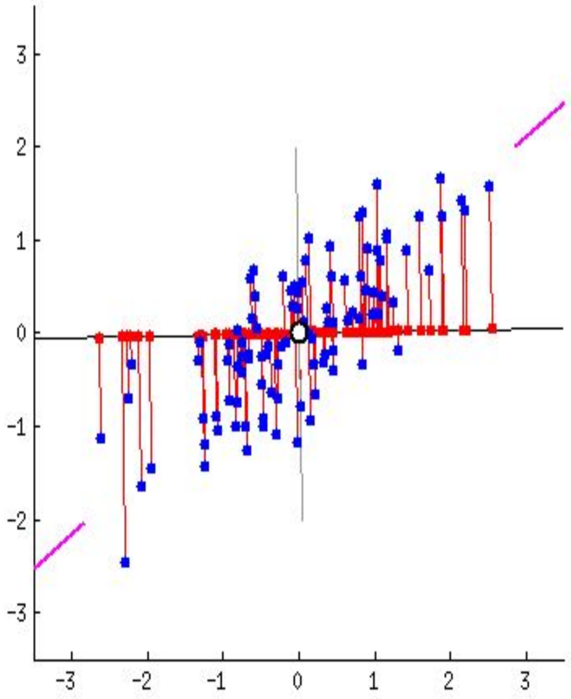
The distance we are trying to maximize

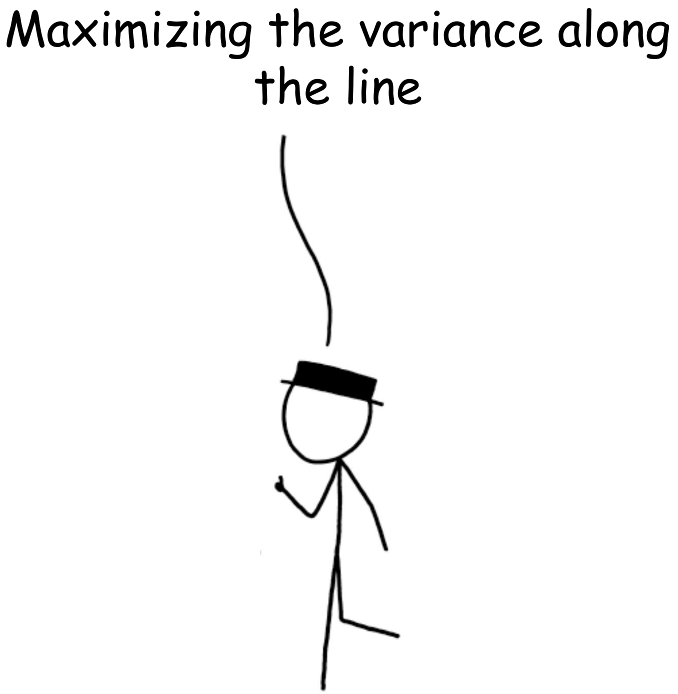
Along the “maximally variant” direction, the distance between the projections of two points on this line corresponds to the greatest variation in the values of the two points.

Principal Component Analysis


How exactly does maximizing the sum of the distances of these projections from the origin correspond to maximizing the variance along that line?

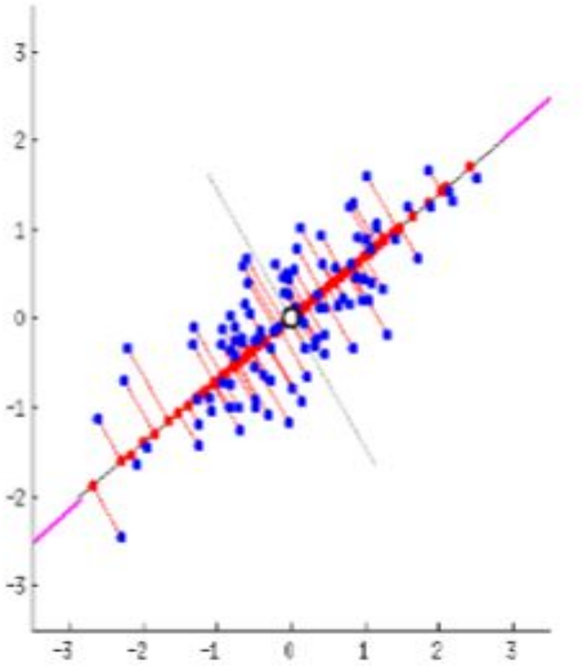
Principal Component Analysis

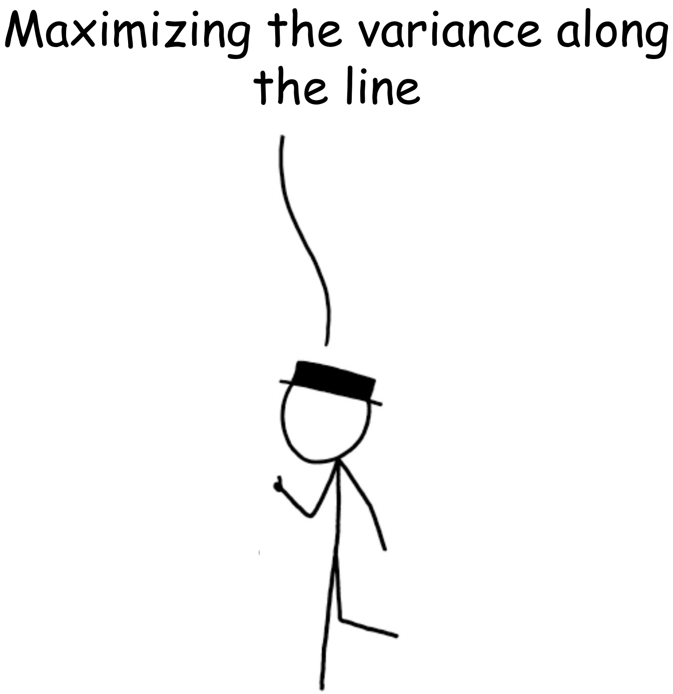




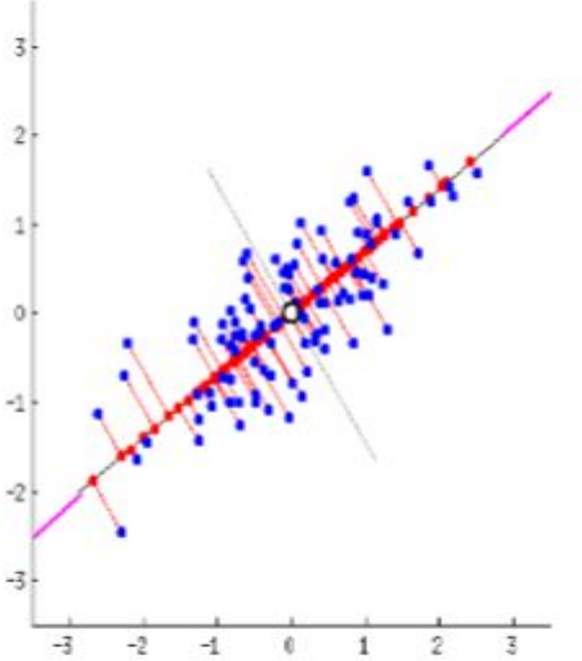
Built using https://gist.github.com/anonymous/7d888663c6ec679ea65428715b99bfdd

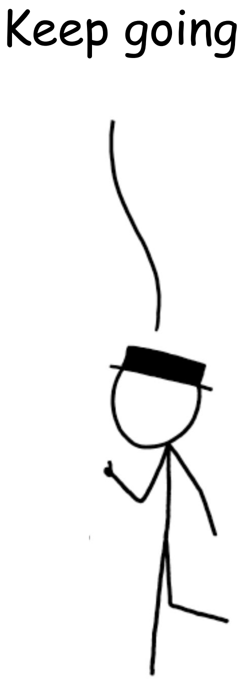
Principal Component Analysis





Principal Component Analysis





Principal Component Analysis

Standardization

Covariance Matrix Calculation

Eigenvector Calculation

Form Principal Components and Build Graph

Standardization

Individual	Height (cm)	Weight (kg)	Income ($)	Number of Children
Person A	165	65	60,000	2
Person B	168	63	100,000	5
Person C	159	82	50,000	1
Person D	183	68	90,000	4
Person E	187	87	110,000	5
Person F	189	89	95,000	4

Compare the data of each of the 4 columns. How do they differ numerically?

Standardization

Individual	Height (cm)	Weight (kg)	Income ($)	Number of Children
Person A	165	65	60,000	2
Person B	168	63	100,000	5
Person C	159	82	50,000	1
Person D	183	68	90,000	4
Person E	187	87	110,000	5
Person F	189	89	95,000	4
Range	159-189	63-89	50,000-110,000	1-5
Variance	161.76	135.87	564166666	2.7

Compare the data of each of the 4 columns. How do they differ numerically?

Their range varies drastically. Consequently, their variances are very different.

Standardization

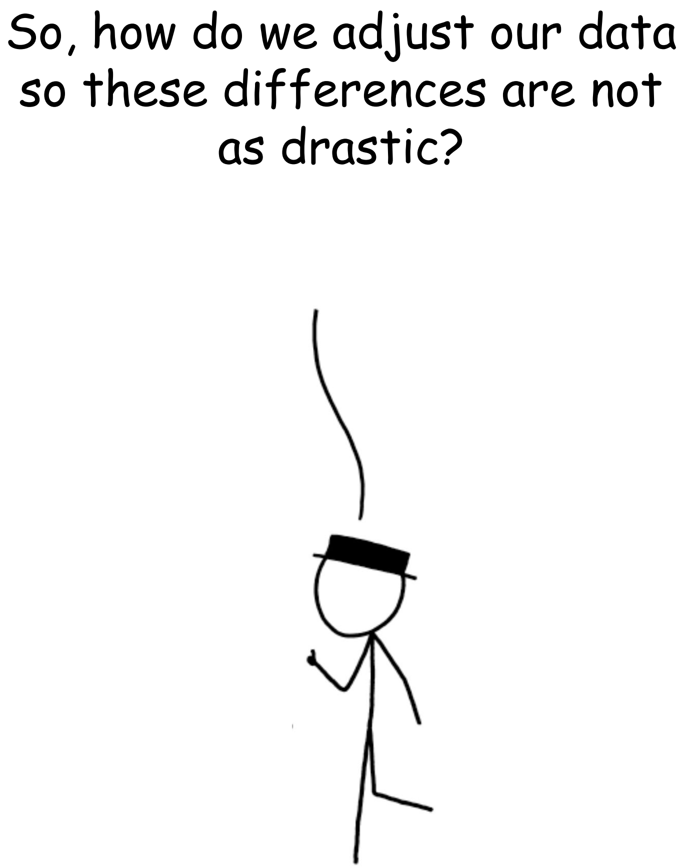
Individual	Height (cm)	Weight (kg)	Income ($)	Number of Children
Person A	165	65	60,000	2
Person B	168	63	100,000	5
Person C	159	82	50,000	1
Person D	183	68	90,000	4
Person E	187	87	110,000	5
Person F	189	89	95,000	4
Range	159-189	63-89	50k-100k	1-5
Variance	161.76	135.87	564166670	2.7

If this is not addressed, some of the feature columns will **dominate** over the other ones.

This can bias the results and final principal component analysis; making it difficult to view differences between values in one column compared to another.

So final graph may have the differences between the weights of various persons be miniscule.

Standardization


So, how do we adjust our data so these differences are not as drastic?

Standardization

Recap: we want to put different variables on the same scale.

This can mean many things from giving them the same mean and standard deviation, to keeping the range consistent, and so on.

Here, we will use a method called **z-scoring.**

$$z = \frac{value - mean}{standard\ deviation}$$

The rescaled distribution will have a mean of 0 and standard deviation of 1

Note: it does not mean the new data follow Normal distribution

Principal Component Analysis

Standardization

Covariance Matrix Calculation

Eigenvector Calculation

Form Principal Components and Build Graph

Covariance Matrix Calculation

Covariance is really just a measure of how **correlated** two variables/features are.

If your covariance is positive, that means there’s a **positive correlation.**

If your covariance is negative, that means there’s a **negative correlation.**

$$Cov(x,y) = \sum \frac{(x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

Covariance Matrix Calculation

Review: Lecture 7; feature engineering

Make new features with high variance.

Pick new features with low correlation to other features.

What should our new features look like?

Covariance Matrix Calculation

Can measure this correlation using **covariance.** If covariance is **positive,** then features are correlated in the sense they both increase together. If covariance is **negative,** then features are inversely correlated.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

$$Cov(x,y) = \sum \frac{(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

Principal Component Analysis

Standardization

Covariance Matrix Calculation

Eigenvector Calculation

Form Principal Components and Build Graph

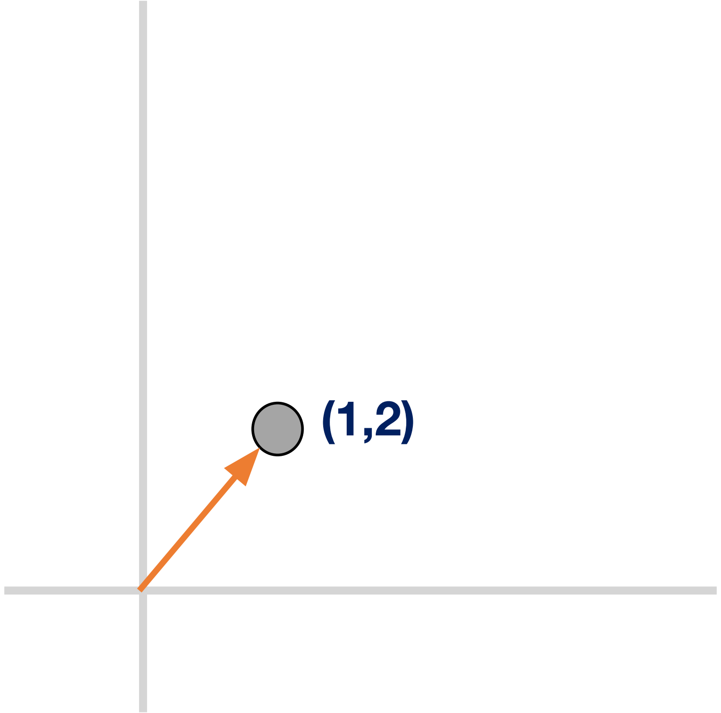
Eigenvector Calculation

We can think of matrices as **transformations** of vectors.

When you multiply a matrix with a vector; two things happen:

1. It **scales** the vector.
2. It **rotates** the vector

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$$



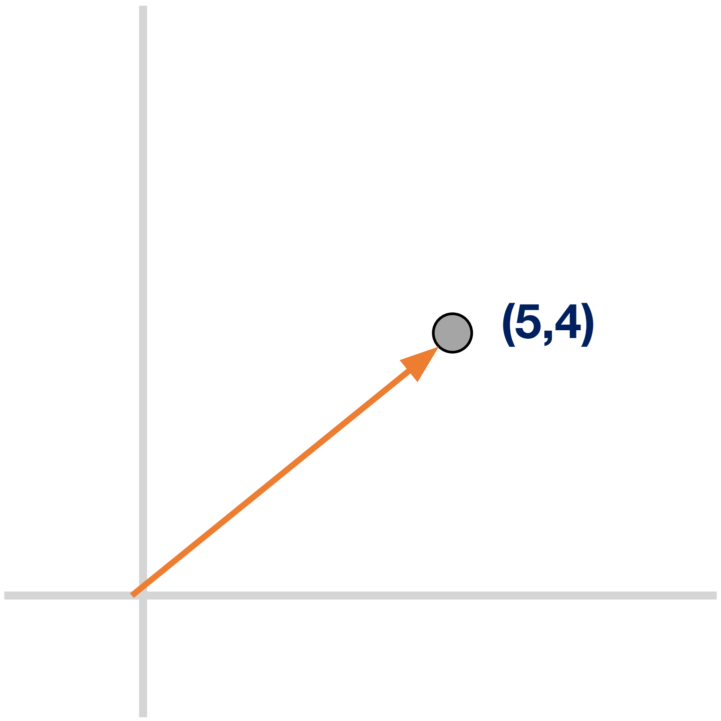
Eigenvector Calculation

We can think of matrices as **transformations** of vectors.

When you multiply a matrix with a vector; two things happen:

1. It **scales** the vector.
2. It **rotates** the vector

$$\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix} = \begin{bmatrix}5\\ 4\end{bmatrix}$$

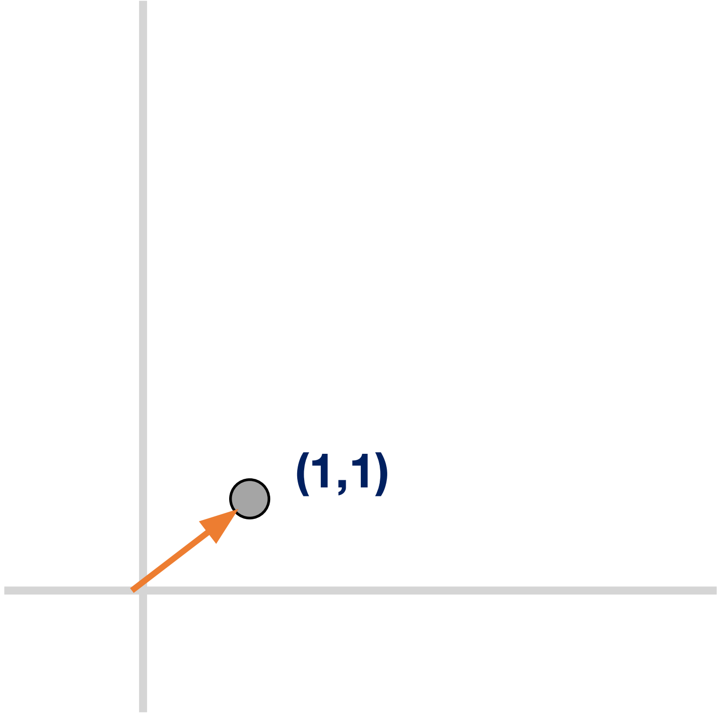


Eigenvector Calculation

Eigenvectors are characteristic vectors specific to a matrix or transformation.

Graphically speaking, when you multiply a matrix with its specific eigenvectors, the eigenvectors **don't get rotated, only scaled.**

$$\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}3\\3\end{bmatrix} = 3\begin{bmatrix}1\\1\end{bmatrix}$$



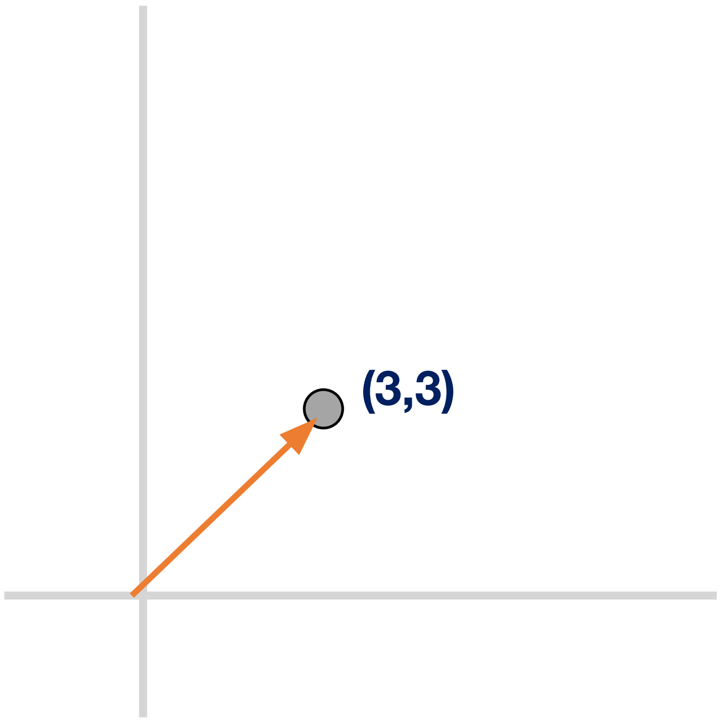
Eigenvector Calculation

Eigenvectors are characteristic vectors specific to a matrix or transformation.

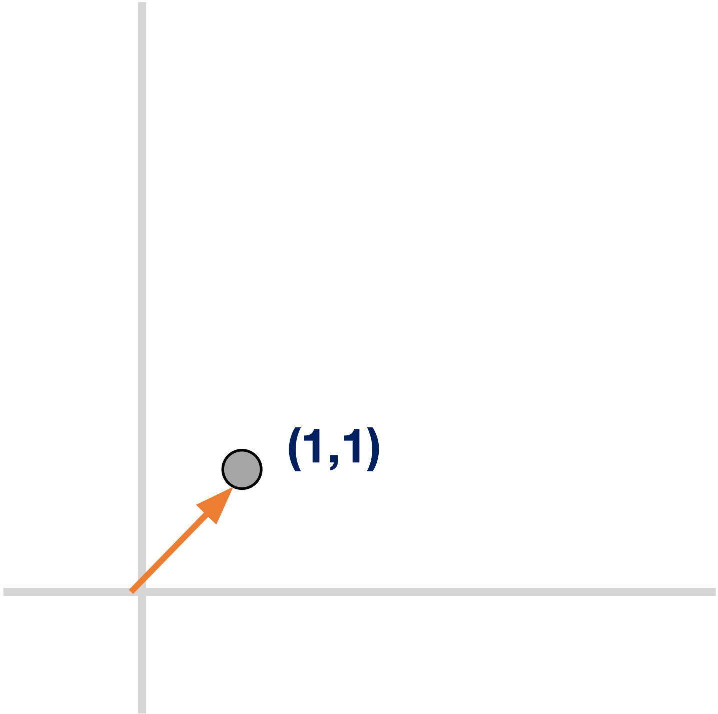
Graphically speaking, when you multiply a matrix with its specific eigenvectors, the eigenvectors **don't get rotated, only scaled.**

The factor by which an eigenvector is scaled is called its eigenvalue

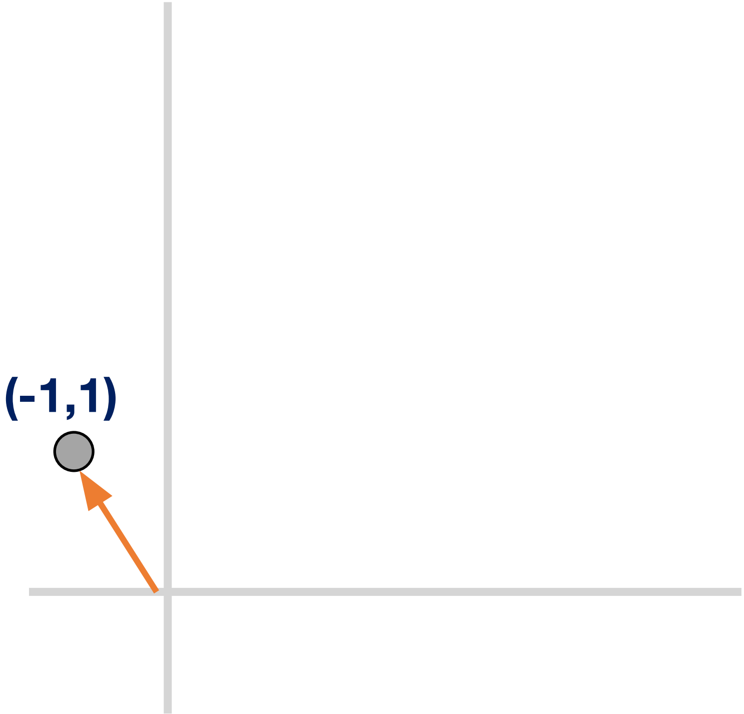
$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



Eigenvector Calculation

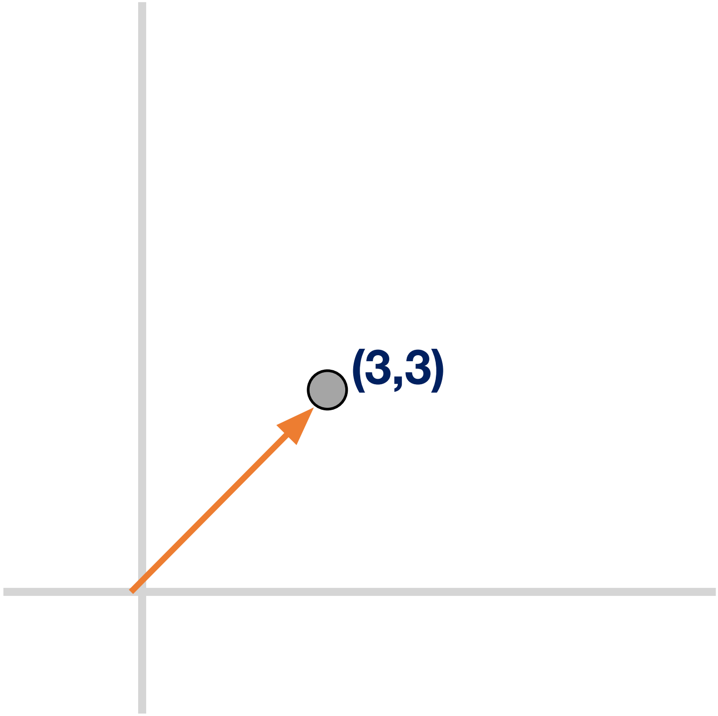


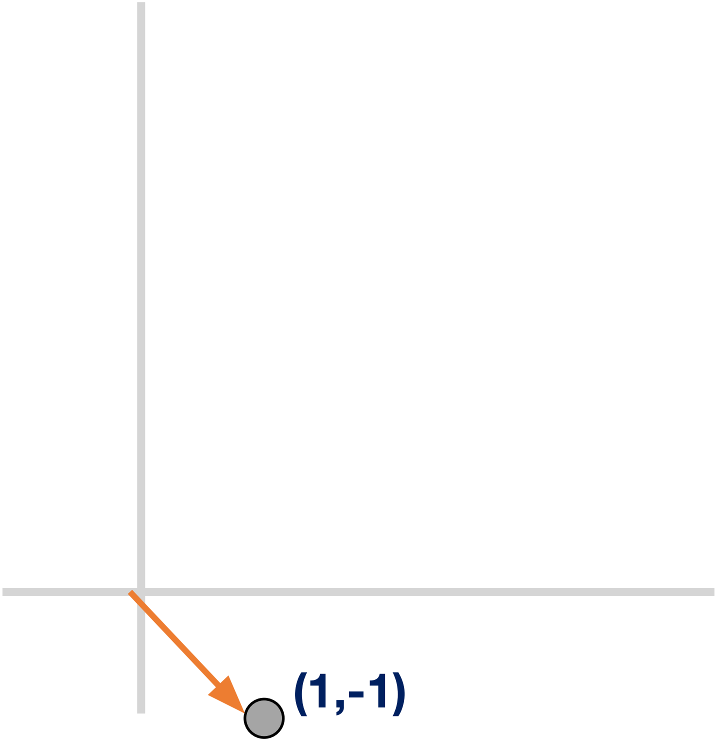
$$\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}3\\3\end{bmatrix} = 3\begin{bmatrix}1\\1\end{bmatrix}$$



$$\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} = \begin{bmatrix}1\\-1\end{bmatrix} = -1\begin{bmatrix}-1\\1\end{bmatrix}$$

Eigenvector Calculation



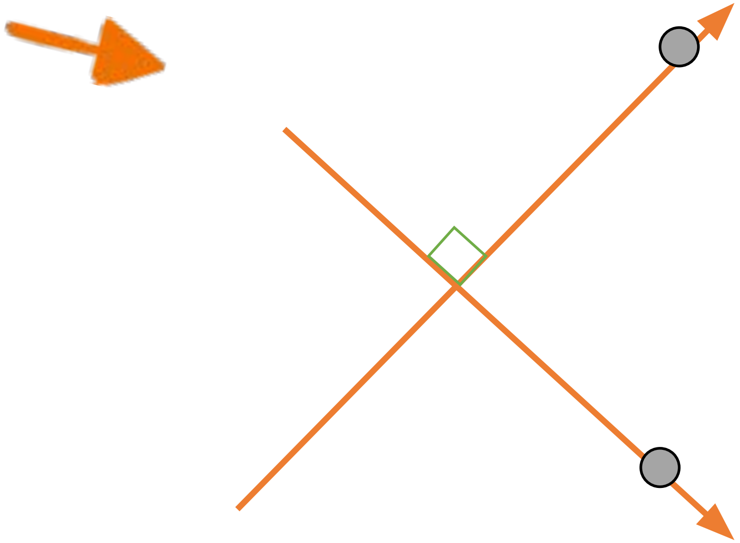


$$\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}3\\3\end{bmatrix} = 3\begin{bmatrix}1\\1\end{bmatrix}$$

$$\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} = \begin{bmatrix}1\\-1\end{bmatrix} = -1\begin{bmatrix}-1\\1\end{bmatrix}$$

Eigenvector Calculation

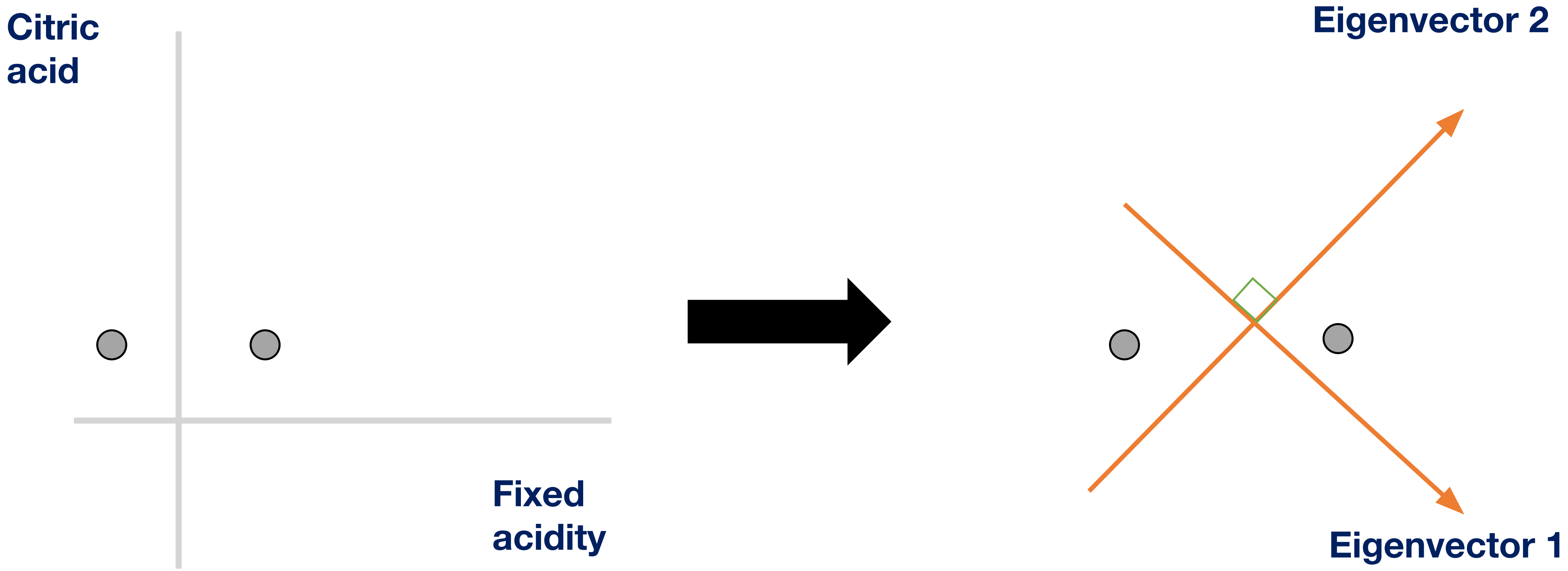
The two eigenvectors are perpendicular to each other!



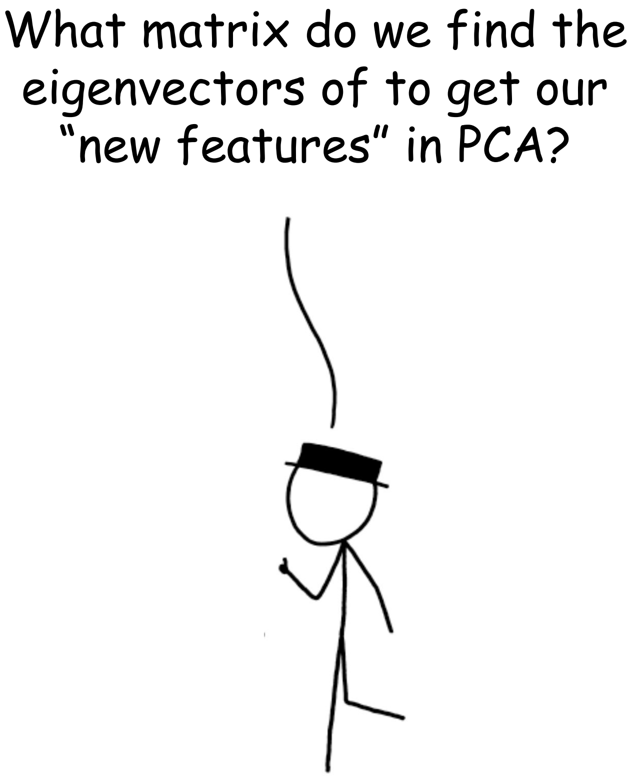
Eigenvectors act as basis vectors!

Every point in 2-D can be expressed as some combination of (1,1) and (-1,1).

Eigenvector Calculation



Eigenvector Calculation


What matrix do we find the eigenvectors of to get our "new features" in PCA?

Eigenvector Calculation

By calculating the eigenvectors of the covariance matrix, we can get our **principal components.**

We use the eigenvectors to create a basis for the graph. These basis vectors represent the principal components.

Since these are eigenvectors of the **covariance matrix**, they represent **directions of maximal variance.**

$$A = \begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

$$Av = \lambda v$$

v is the eigenvector and lambda is the eigenvalue

Eigenvector Calculation

$$Av = \lambda v$$

$$Av - \lambda v = 0$$

$$(A - \lambda)v = 0$$

$$|A - \lambda| = 0$$

$$A = \begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

When you find the root of the resulting polynomial, you will find all the possible eigenvalues. For each eigenvalue, plug it into the original equation to find the corresponding eigenvector v.

Principal Component Analysis

Standardization

Covariance Matrix Calculation

Eigenvector Calculation

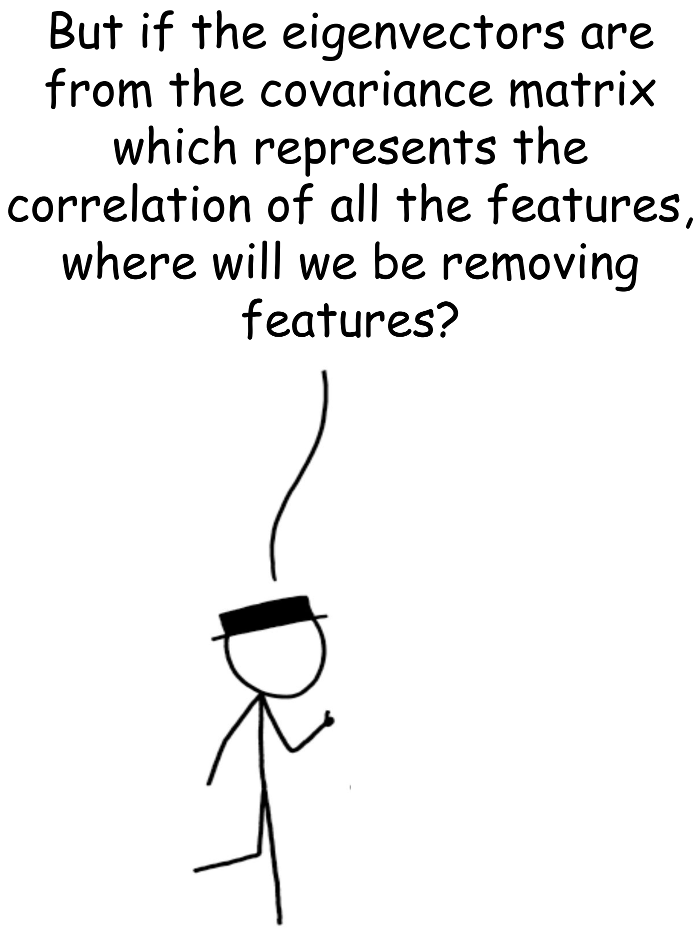
Form Principal Components and Build Graph

Form Principal Components and Build Graph

Let the three eigenvalues of the three eigenvectors $v_1, v_{2,}, v_3$ be $\lambda_1, \lambda_2, \lambda_3$ such that $\lambda_1 >= \lambda_2 >= \lambda_3$

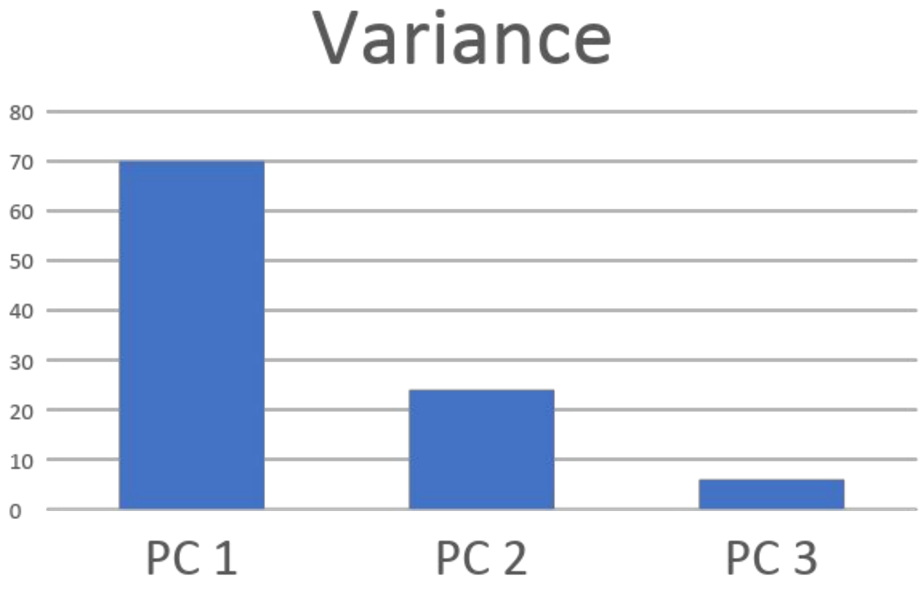
Then, the **principal components** will be $v_1, v_{2,}, v_3$ and the **variances** they carry are in the ratio of $\lambda_1, \lambda_2, \lambda_3$

Form Principal Components and Build Graph


But if the eigenvectors are from the covariance matrix which represents the correlation of all the features, where will we be removing features?

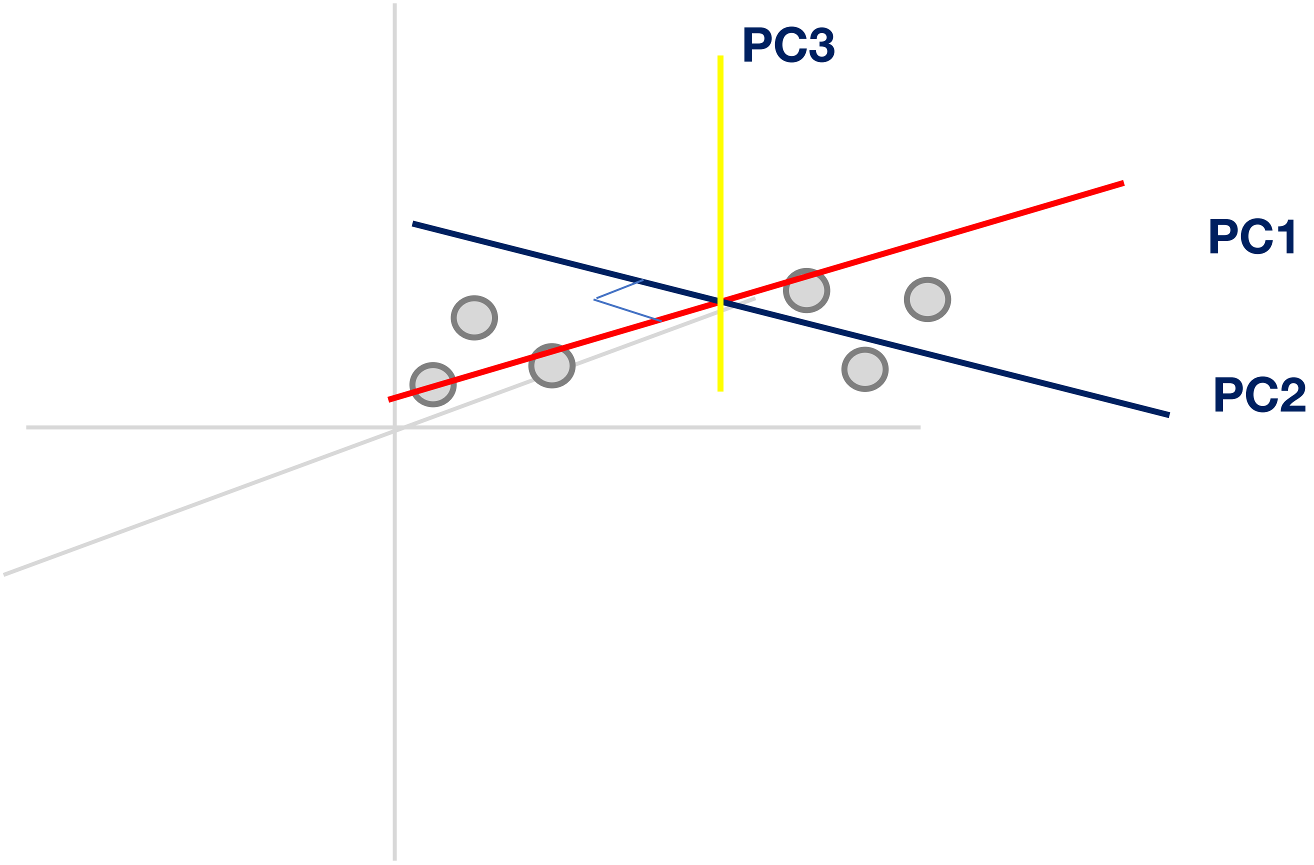
If the percentage of variance of a particular principal component is small enough, discard it. You've now removed a dimension! Form a new matrix which only has the eigenvectors/principal components you've selected.

Let this matrix be called your **Feature Vector.**



Now, it’s time to reorient the original data along these new axes

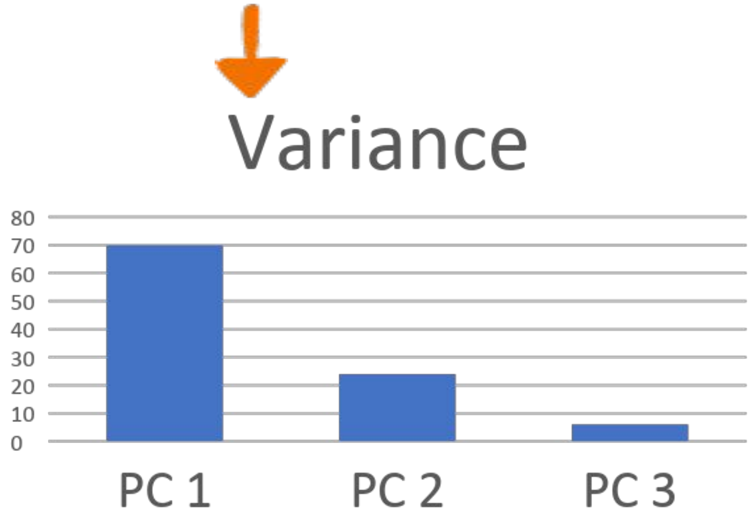
Form Principal Components and Build Graph

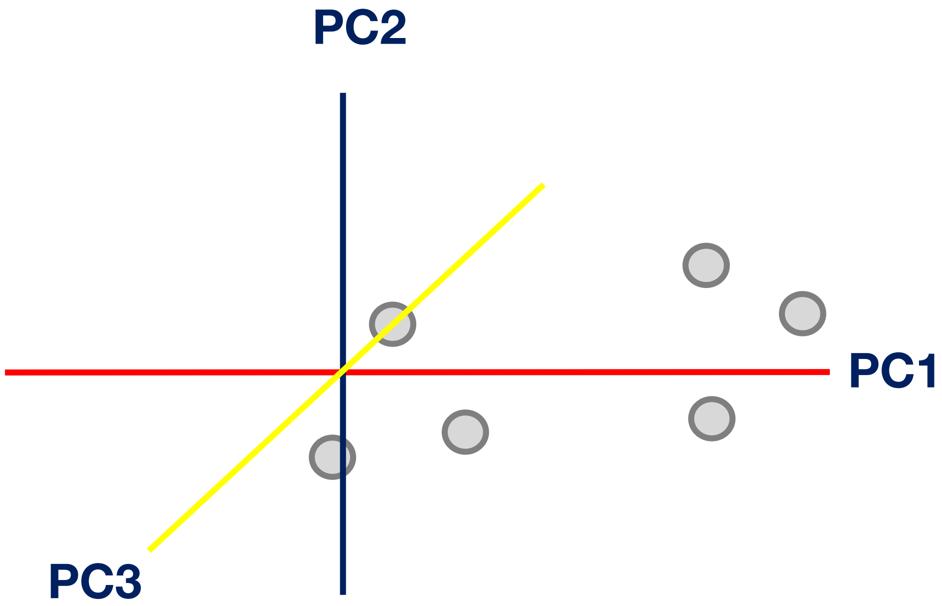


$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Form Principal Components and Build Graph

The third dimension was removed as it did not contribute much in terms of variance





$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Principal Component Analysis

Linear transformation of existing features

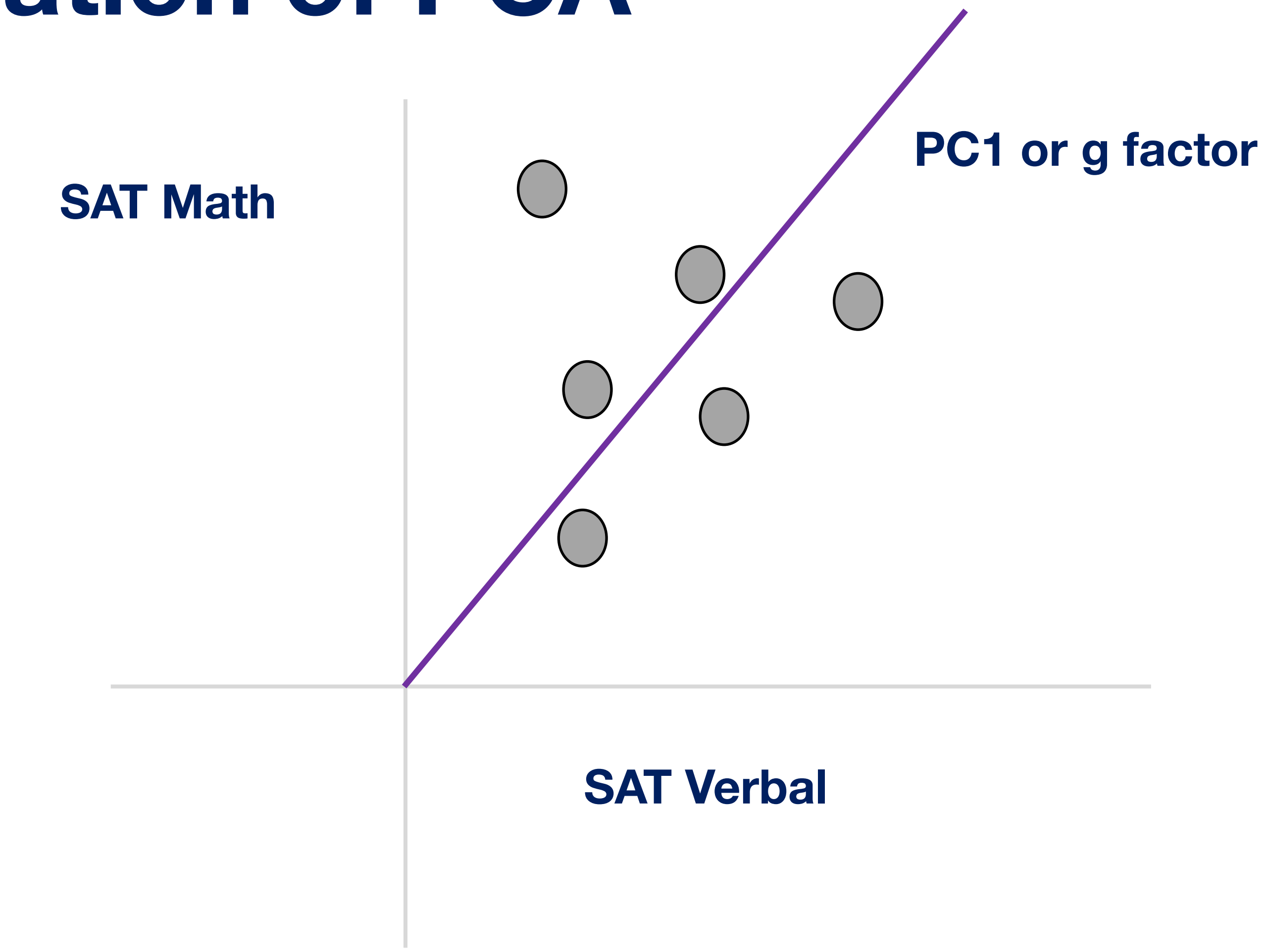
Each PC is selected to maximize the explained variance in this direction of the remain data

Dimensionality reduction is to remove dimensions with low variances

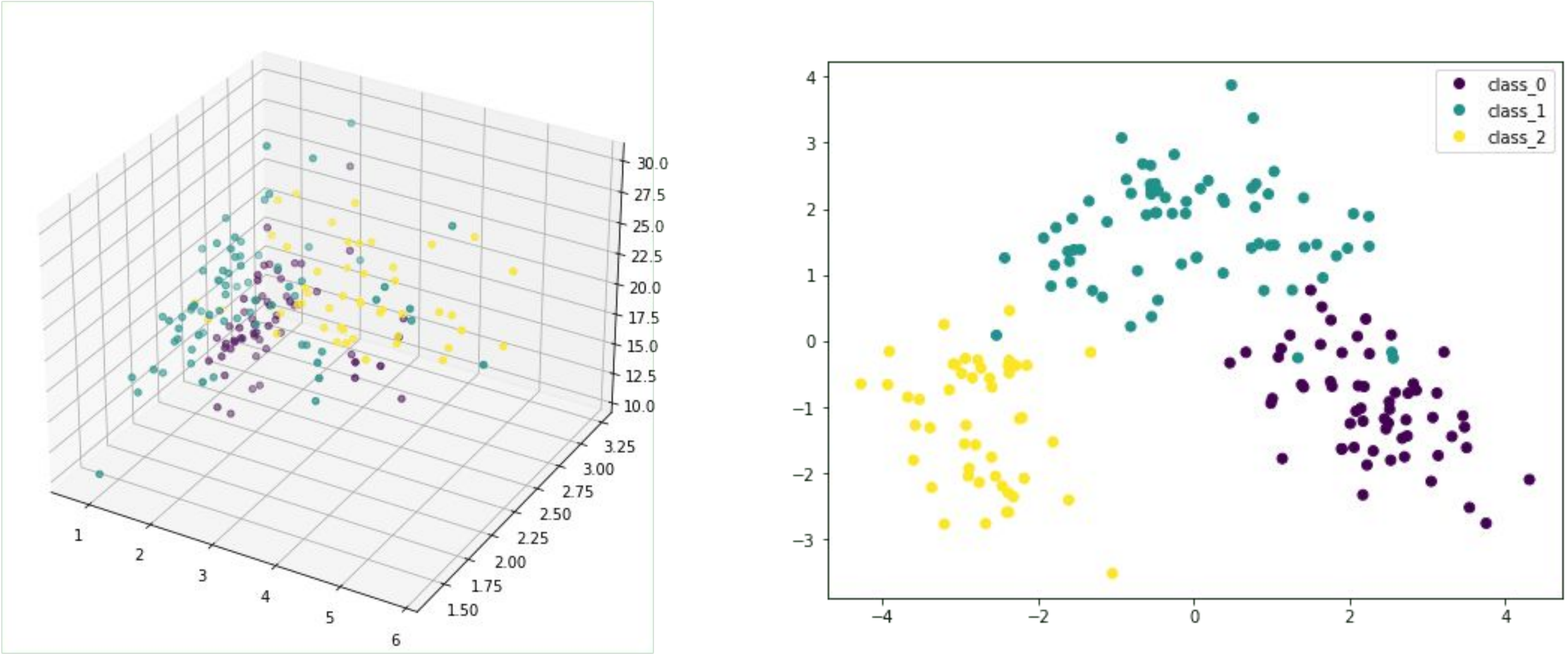
Tradeoff between dimensionality vs. info. loss

A real-world application of PCA

IQ testing!

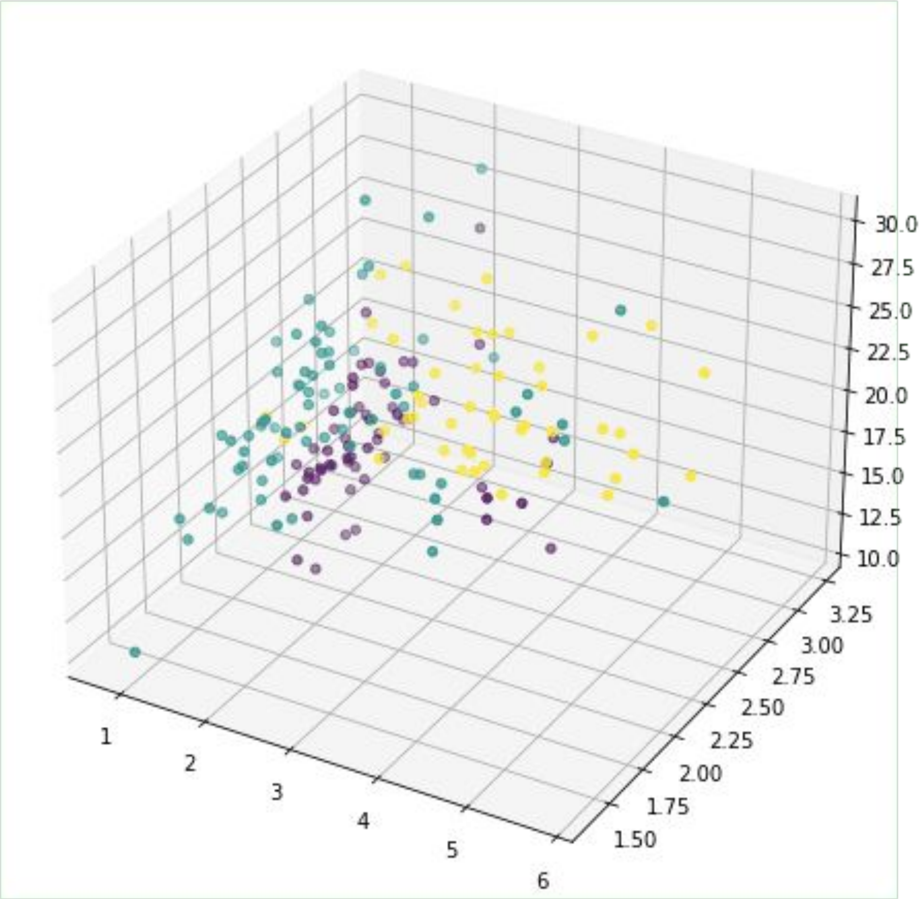


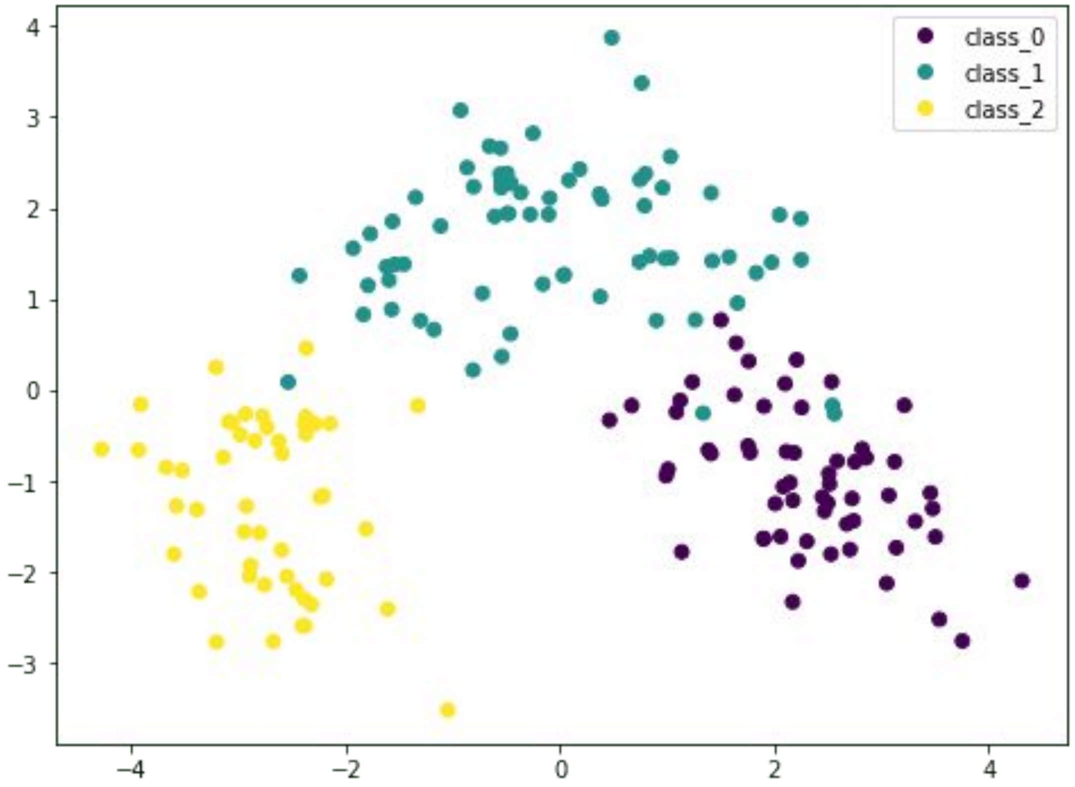
Data Visualization



The wine dataset with 13 features and 3 classes.
Source: https://machinelearningmastery.com/principal-component-analysis-for-visualization/

Feature Extraction



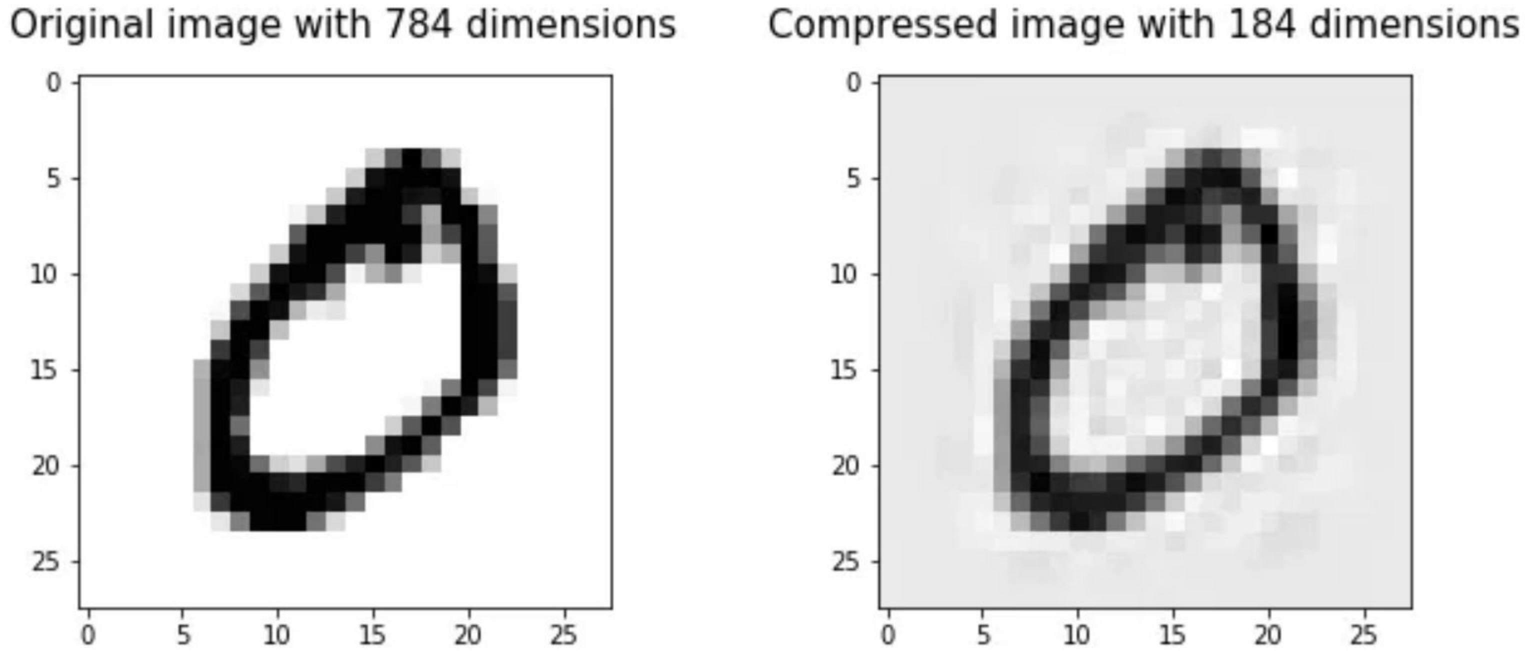


Q: feature extraction vs. feature selection?

The wine dataset with 13 features and 3 classes.

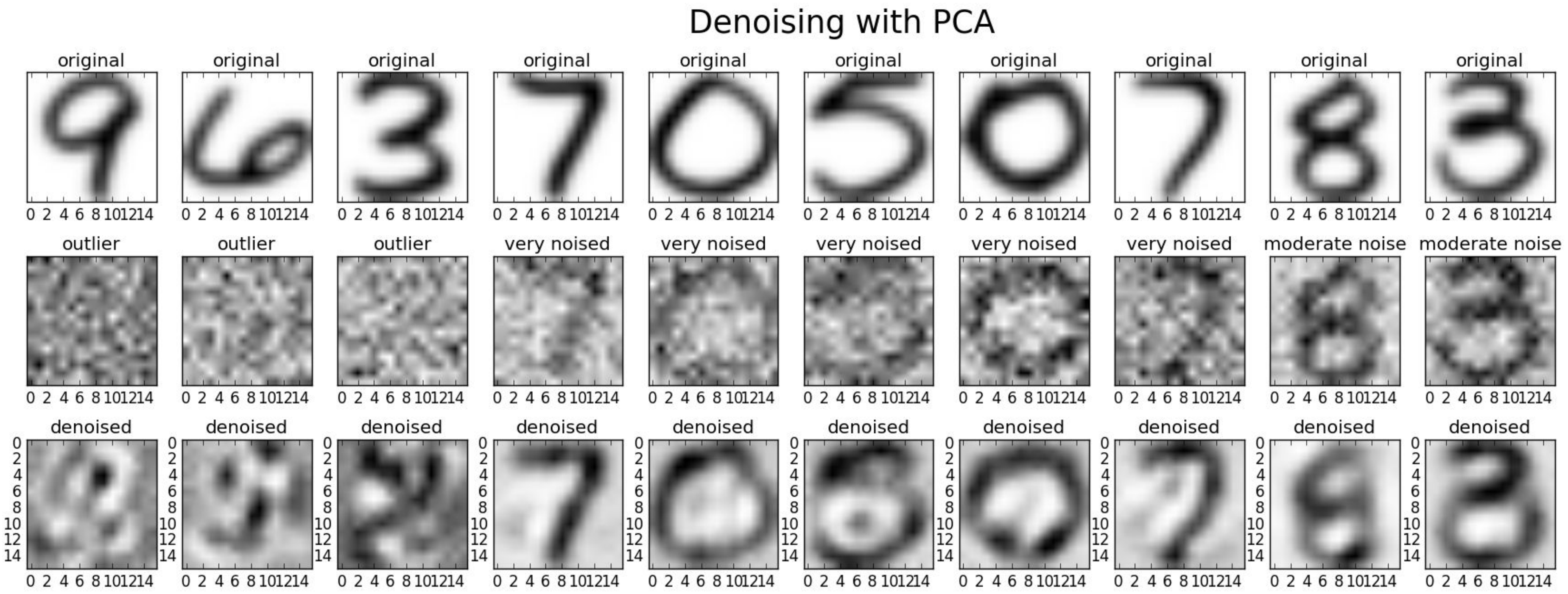
Source: https://machinelearningmastery.com/principal-component-analysis-for-visualization/

Image Compression



Source: https://towardsdatascience.com/image-compression-using-principal-component-analysis-pca-253f26740a9f

Noise Reduction

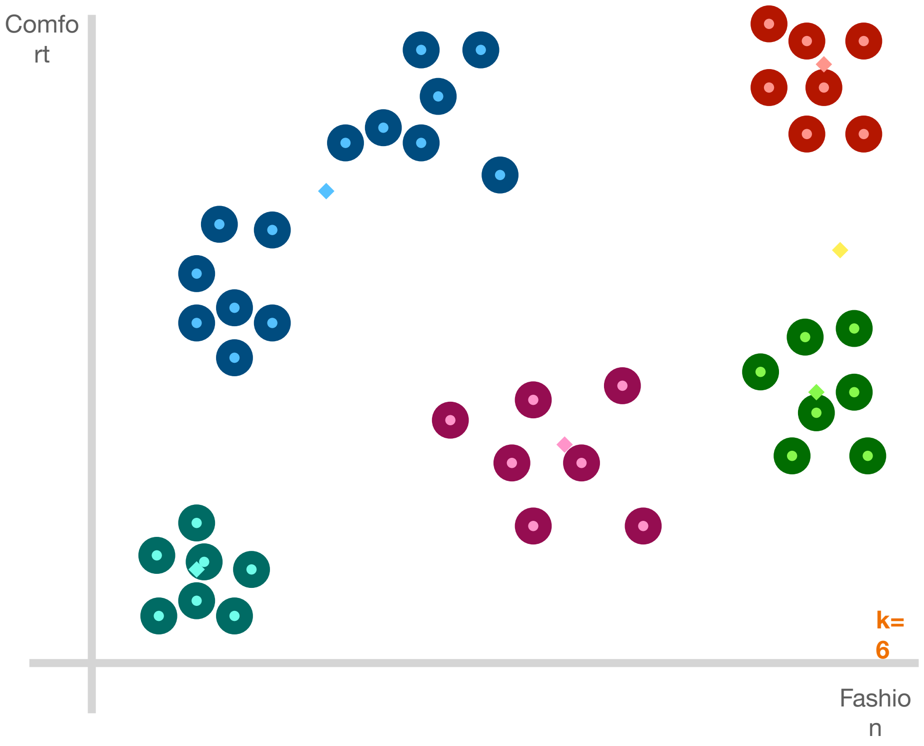


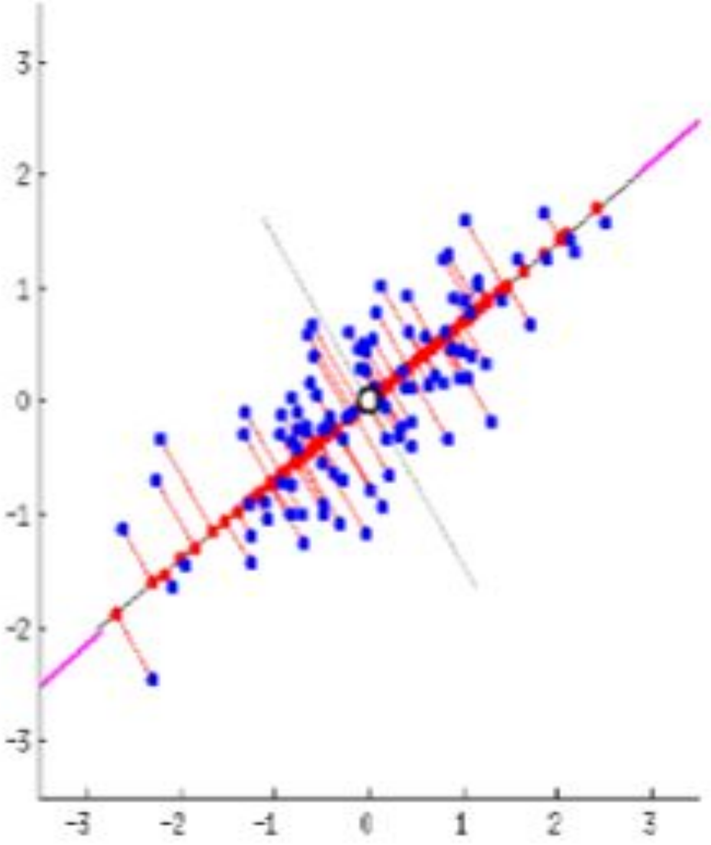
Source: https://stats.stackexchange.com/questions/247260/principal-component-analysis-eliminate-noise-in-the-data

Unsupervised Learning

Clustering

Dimension reduction





Project

Typical steps to apply ML

- Data preprocessing
- Trying different ML algorithms
  - Training set, validation set, test set
- Diagnostics
  - More training samples
  - Increase/decrease feature set
  - Increase/decrease regularization
- Loop back

A ML Project

- Why ML is a suitable approach
  - Do not use ML for the purpose of using ML
  - Evaluate existing approaches and room for improvement
- Problem abstraction and formulation
  - Set appropriate goals
  - Model complexity, data availability, evaluation
  - Domain knowledge critical
- Data collection and data cleaning
  - What, where, and how
- ML algorithms
  - This is often the “easy” part
- Evaluation, sanity check, interpretation
- Iterate the process