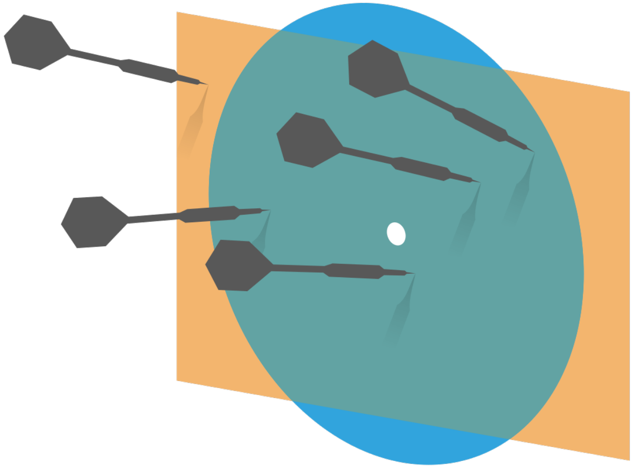
Probability and Statistics for Computer Science



Credit: wikipedia

"Unsupervised learning is arguably more typical of human and animal learning…"--- Kelvin Murphy, former professor at UBC

Hongye Liu, Teaching Assistant Prof, CS361, UIUC, 12.01.2020

Last time

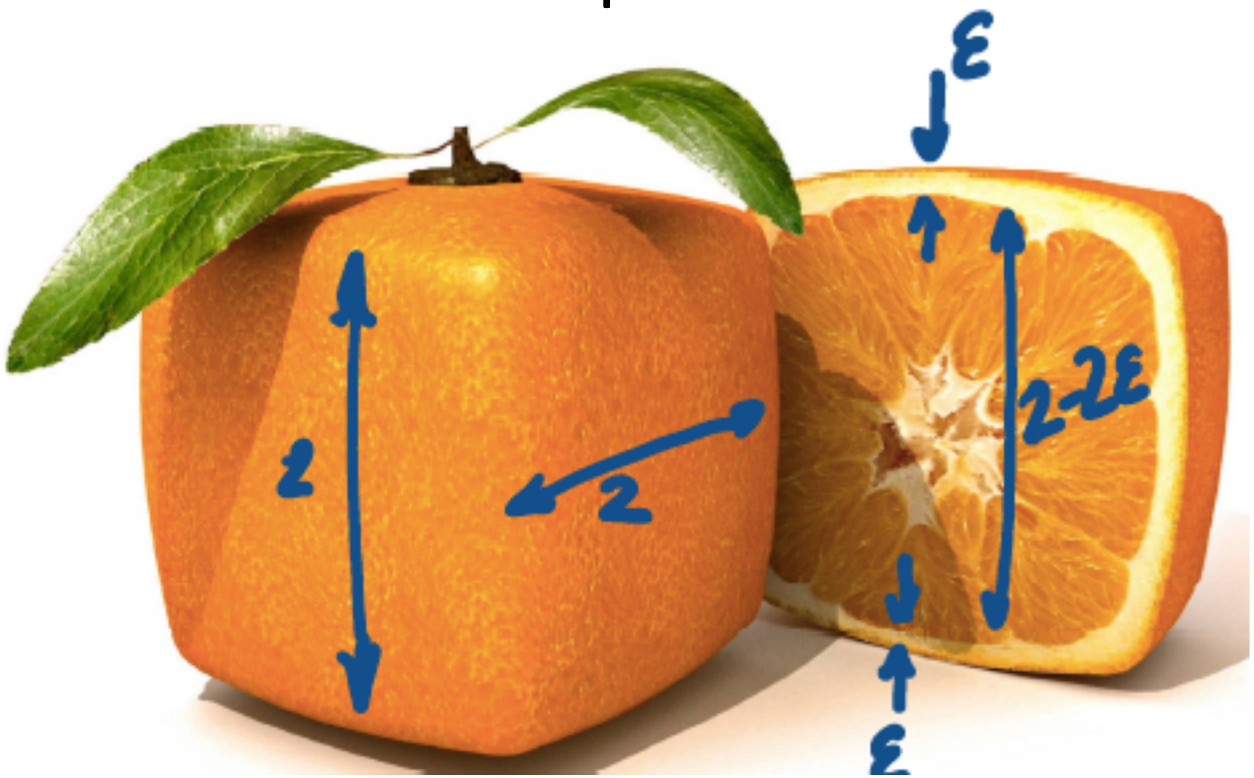
- Linear Regression (II)
- Nearest Neighbor Regression

Objectives

- The curse of dimensionality
- Multivariate normal distribution
- Concept of unsupervised learning
- Clustering (I)

First let's take a look at a 3D object

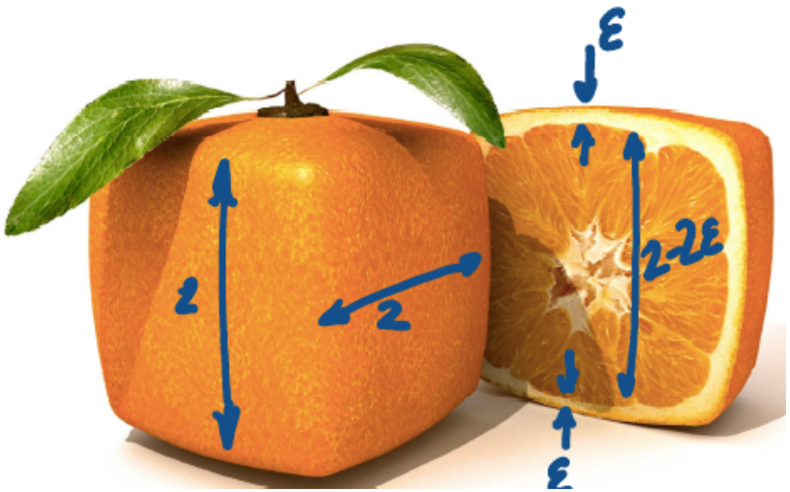
Is there more fruit than peel?



Credit: Prof. David Varodayan

First take a look at a 3D object

Is there more fruit or more peel?



Total Volume: 2^3

Vol. of fruit: $(2-2\varepsilon)^3$

Vol. of peel: $2^3-(2-2\varepsilon)^3$

Fraction of peel: $1-(1-\varepsilon)^3$

If $\varepsilon = 0.05$ fraction of peel ≈ 0.143

Credit: Prof. David Varodayan

What if we have a d-dimensional orange?

Is there always more fruit?

A. YES
B. NO

In arbitrary d-dimension

- Total amount of orange
- Amount of fruity part
- Fraction of orange that is peel

The curse of dimensions

- If a dataset is uniformly distributed in a high-dimensional cube (or other shape), majority of data is far from the origin.

- The above can be roughly proved by calculating the expected distance from the origin

The Expected distance from the origin in d-dimensional cube

$$E[\boldsymbol{x}^T\boldsymbol{x}] = E[\sum_{i=1}^{d} x_i^2] = \sum_{i=1}^{d} E[x_i^2]$$

$$= \sum_{i=1}^{d} \int_{cube} x_i^2 P(\boldsymbol{x}) d\boldsymbol{x}$$

The Expected distance from the origin in d-dimensional cube

$$E[\boldsymbol{x}^T\boldsymbol{x}] = E[\sum_{i=1}^{d} x_i^2] = \sum_{i=1}^{d} E[x_i^2]$$

$$= \sum_{i=1}^{d} \int_{cube} x_i^2 P(\boldsymbol{x}) d\boldsymbol{x}$$

Assuming the independence of each x_i

$$P(\boldsymbol{x}) = P(x_1)P(x_2)...P(x_d)$$

The Expected distance from the origin in d-dimensional cube

$$E[\boldsymbol{x}^T\boldsymbol{x}] = E[\sum_{i=1}^{d} x_i^2] = \sum_{i=1}^{d} E[x_i^2]$$

$$= \sum_{i=1}^{d} \int_{cube} x_i^2 P(\boldsymbol{x}) d\boldsymbol{x}$$

Assuming the independence of each x_i

$$P(\boldsymbol{x}) = P(x_1)P(x_2)...P(x_d)$$

$$\int_{-\infty}^{+\infty} P(x_i) dx_i = 1$$

The general law of continuous probability density

The Expected distance from the origin in d-dimensional cube

$$E[\boldsymbol{x}^T\boldsymbol{x}] = E[\sum_{i=1}^{d} x_i^2] = \sum_{i=1}^{d} E[x_i^2]$$

$$= \sum_{i=1}^{d} \int_{cube} x_i^2 P(\boldsymbol{x}) d\boldsymbol{x}$$

Assuming the independence of each x_i

$$P(\boldsymbol{x}) = P(x_1)P(x_2)...P(x_d)$$

$$\int_{-\infty}^{+\infty} P(x_i)dx_i = 1$$

The general law of continuous probability density

$$\Rightarrow \quad E[\boldsymbol{x}^T\boldsymbol{x}] = \sum_{i=1}^{d} \int_{-1}^{1} x_i^2 P(x_i) dx_i$$

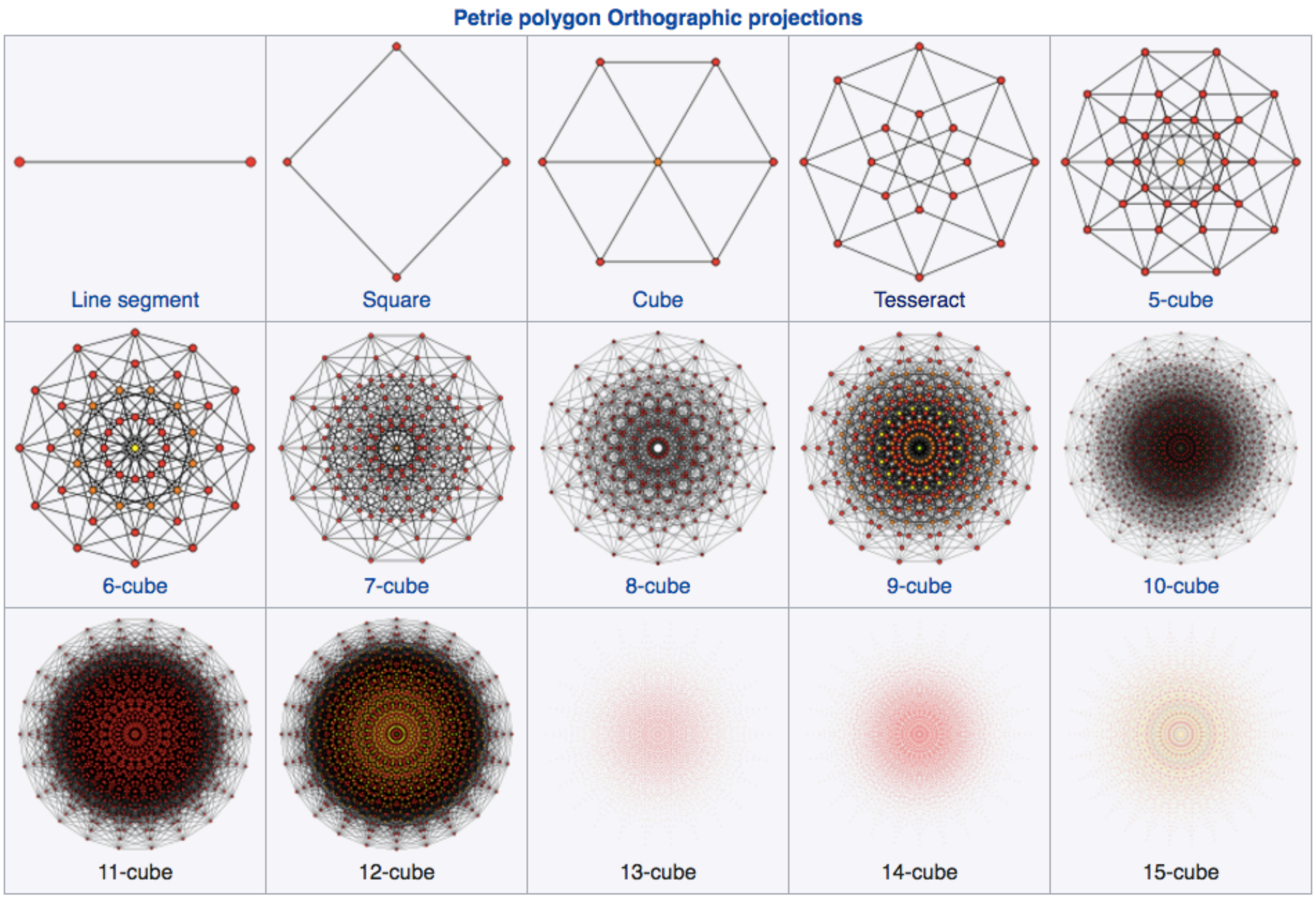
A lot of data is far from the origin.

- On average, data points are d/3 away from the origin (using square of distance)

$$
\begin{aligned}
E[\boldsymbol{x}^T\boldsymbol{x}] &= \sum_{i=1}^{d} \int_{-1}^{1} x_i^2 P(x_i) dx_i \\
&= \sum_{i=1}^{d} \frac{1}{2} \int_{-1}^{1} x_i^2 dx_i \\
&= \frac{d}{3}
\end{aligned}
$$

What do high-dimensional cubes look like?

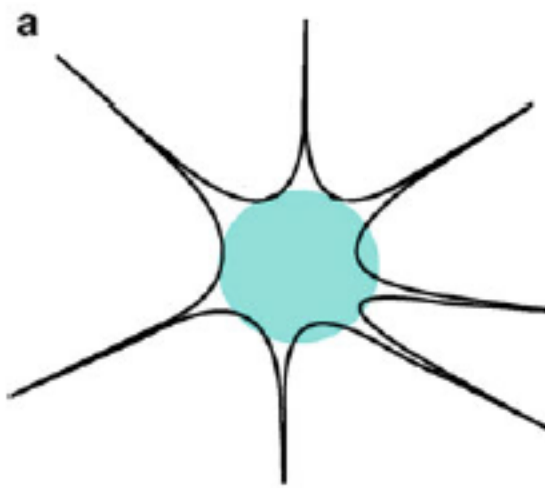
What do high-dimensional cubes look like?



Credit: Wiki

What does a convex object K in high dimensions look like?

The spikes are outliers in high dimension



A general convex set

Credit: G. Pfander editor, “Sampling theory, a Renaissance”

With this scaling, most of the volume of K is located around the Euclidean sphere of radius $\sqrt{n}$. Indeed, taking traces on both sides of the second equation in (1.2), we obtain

$$\mathbb{E}\,\|X\|_2^2 = n.$$

Therefore, by Markov’s inequality, at least 90% of the volume of K is contained in a Euclidean ball of size $O(\sqrt{n})$. Much more powerful concentration results are known—the bulk of K lies very near the sphere of radius $\sqrt{n}$ and the outliers have exponentially small volume. This is the content of the two major results in high-dimensional convex geometry, which we summarize in the following theorem.

Distance between points grows with increasing dimensions

$$E[d(\boldsymbol{u},\boldsymbol{v})^2] = E[(\boldsymbol{u}-\boldsymbol{v})^T(\boldsymbol{u}-\boldsymbol{v})]$$
$$= E[\boldsymbol{u}^T\boldsymbol{u}] + E[\boldsymbol{v}^T\boldsymbol{v}] - 2E[\boldsymbol{u}^T\boldsymbol{v}]$$

High dimensional histogram of a data set is unhelpful

- Most bins will be empty
- Some bins will have single data
- Very few will have more than one data point

Dealing with high dimensional data

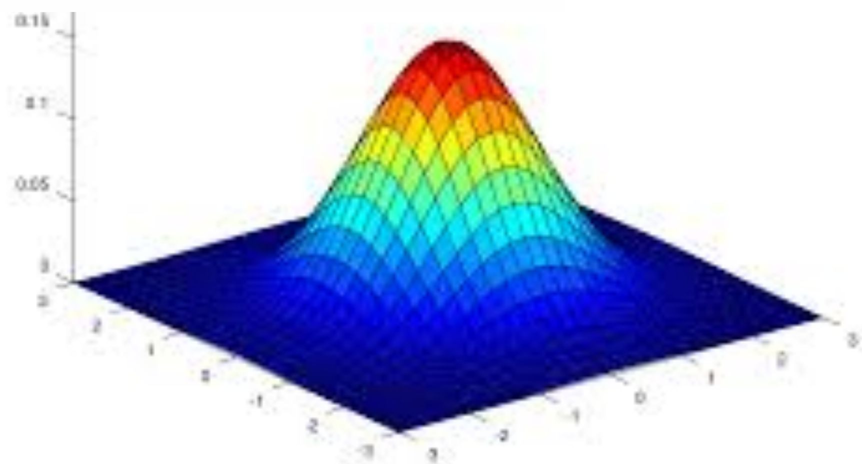
- Collect as much data as possible
- Cluster data into blobs/cluster
- Fit each blob with simple probability model

Multivariate normal distribution

- Extension of the normal distribution to multiple dimensions
- Bivariate normal distribution looks like this:

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]}$$

$-1 < \rho < 1$



Multivariate normal probability densitiy

- A multivariate normal random vector **X** of dimension d has this pdf:

$$P(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \left|\Sigma\right|}} exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}))$$

where

$\boldsymbol{\mu} = E[\boldsymbol{x}]$ is d-dimensional mean vector

$\Sigma = E[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]$ is the $d \times d$ positive definite covariance matrix

Multivariate MLE

- Given a d-dimensional data set ({x}) we can fit a multivariate normal distribution using MLE

$$P(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \left|\Sigma\right|}} exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}))$$

Unsupervised learning

- **Unsupervised learning** means knowledge discovery from the feature vectors **without labels**.
- Unsupervised learning may include:
  - Discovering **latent factors**
  - Discovering **clusters**
  - Discovering **graph structure**
  - Matrix completion

Q. Is this true?

- **Principal Component Analysis** is an unsupervised learning method.

A. TRUE

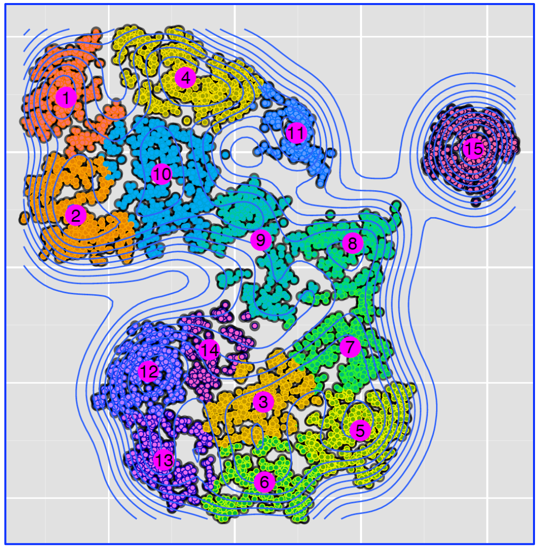
B. FALSE

Dimension Reduction is unsupervised learning

- For example in **Principal Component Analysis**, no labels are assumed about the data.
- PCA discovers the latent factors--- the important eigenvectors of the covariance matrix

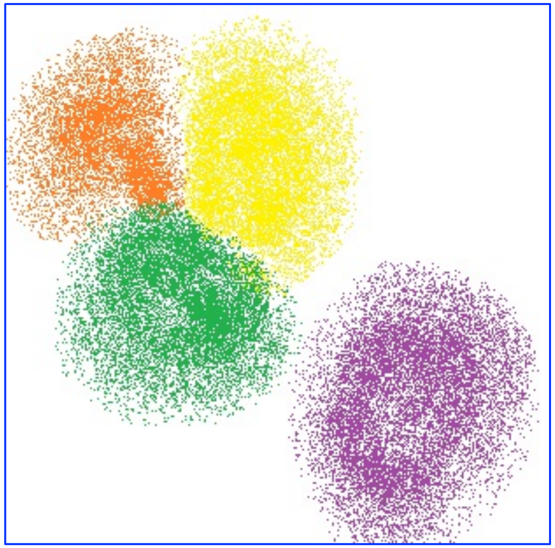
The family of unsupervised learning

Dimension reduction



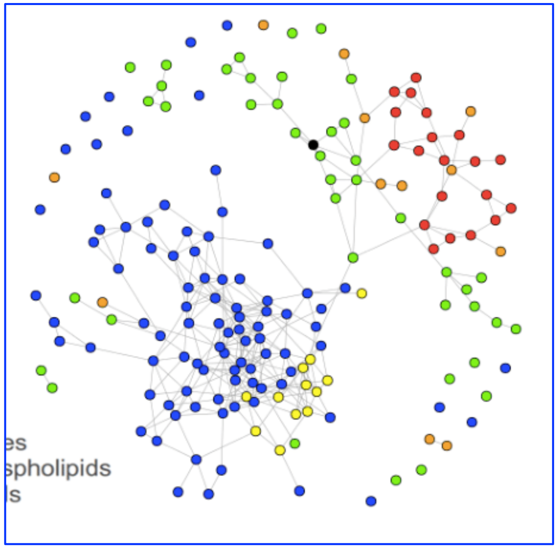
t-SNE

Clustering



K-means

Graph structure



Gaussian Graph model

……

Clustering as an unsupervised learning method

- Clustering identifies specific structure called **clusters**.
- In clustering **data is not labeled**. By identifying clusters, the method assigns cluster membership labels to data.
- A cluster is formed so that
  - Items within a cluster are “close” to each other
  - Items in different clusters are “far” from each other
  - Distance metric is important in clustering

Types of clustering method

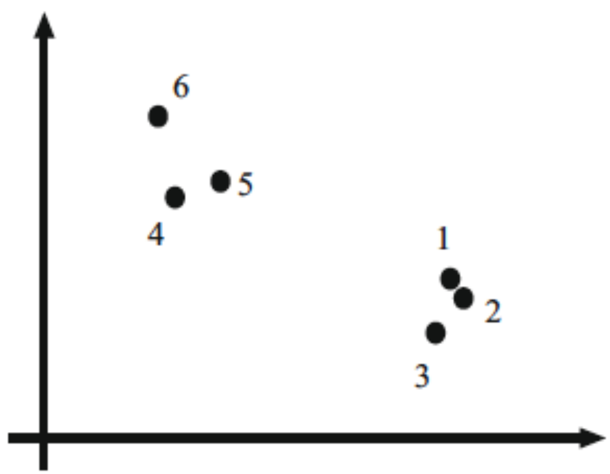
- By input type:
  - **Similarity based clustering**: input is N x N similarity/ distance matrix
  - **Feature based clustering:** input is N x D feature matrix
- By output type:
  - **Hierarchical clustering**
    - Top-down (divisive)
    - Bottom-up (agglomerative)
  - **Flat clustering**:
    - Mixture models, K-means clustering, Spectral clustering…

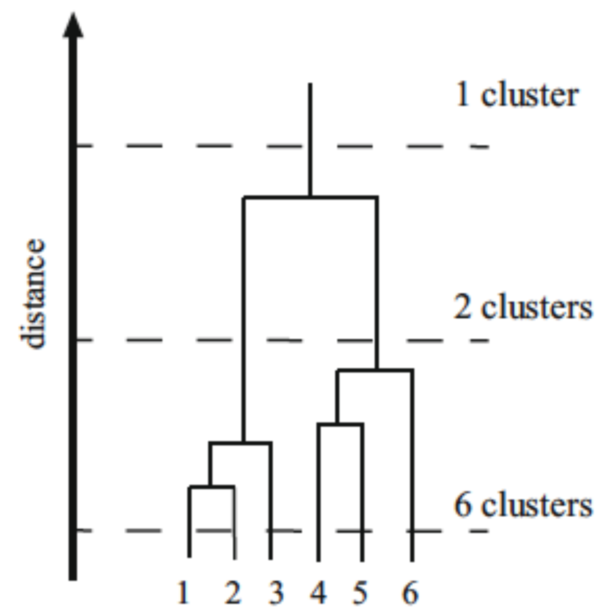
Hierarchical Clustering (I)

- Divisive clustering
  - Treat the whole dataset as a single cluster
  - Then split the data set recursively until you get a satisfactory clustering

Hierarchical Clustering (II)

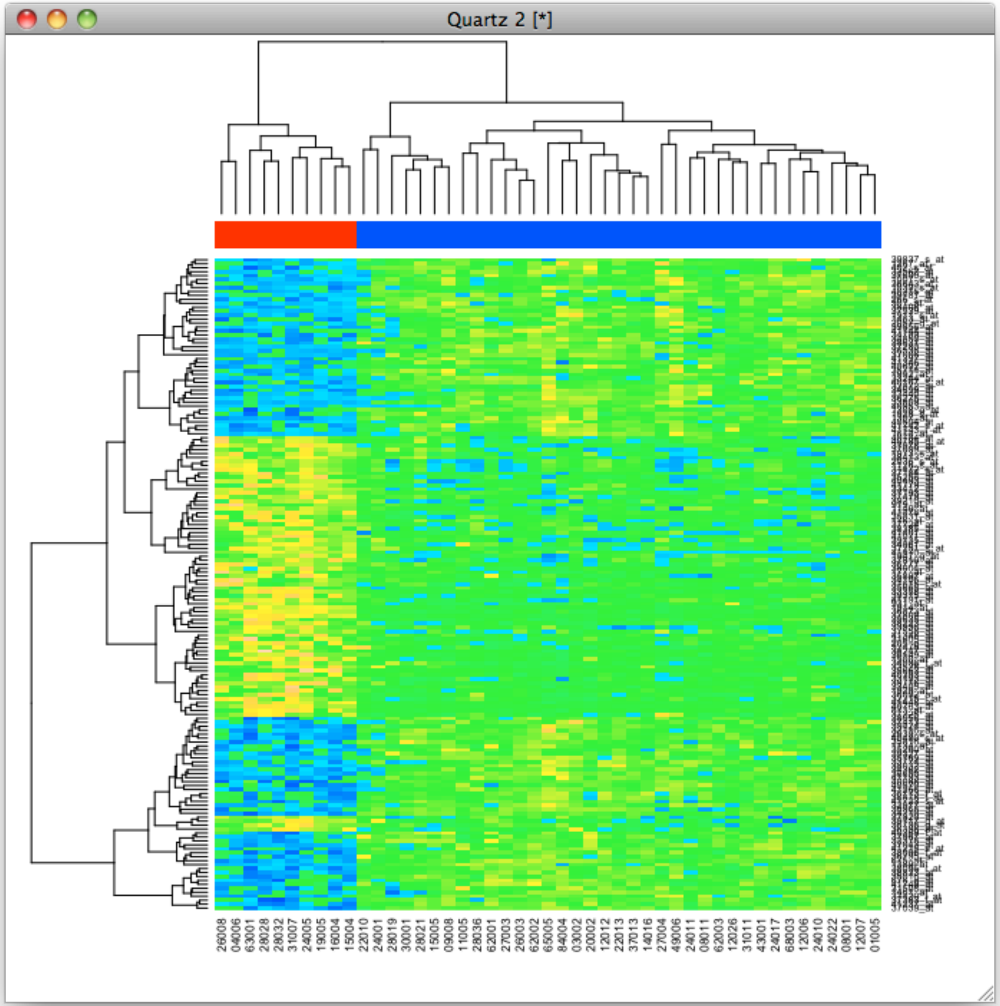
- Agglomerative clustering
  - Treat each data item as its own cluster
  - Then merge clusters until you get a satisfactory clustering
  - A "dendrogram" is created





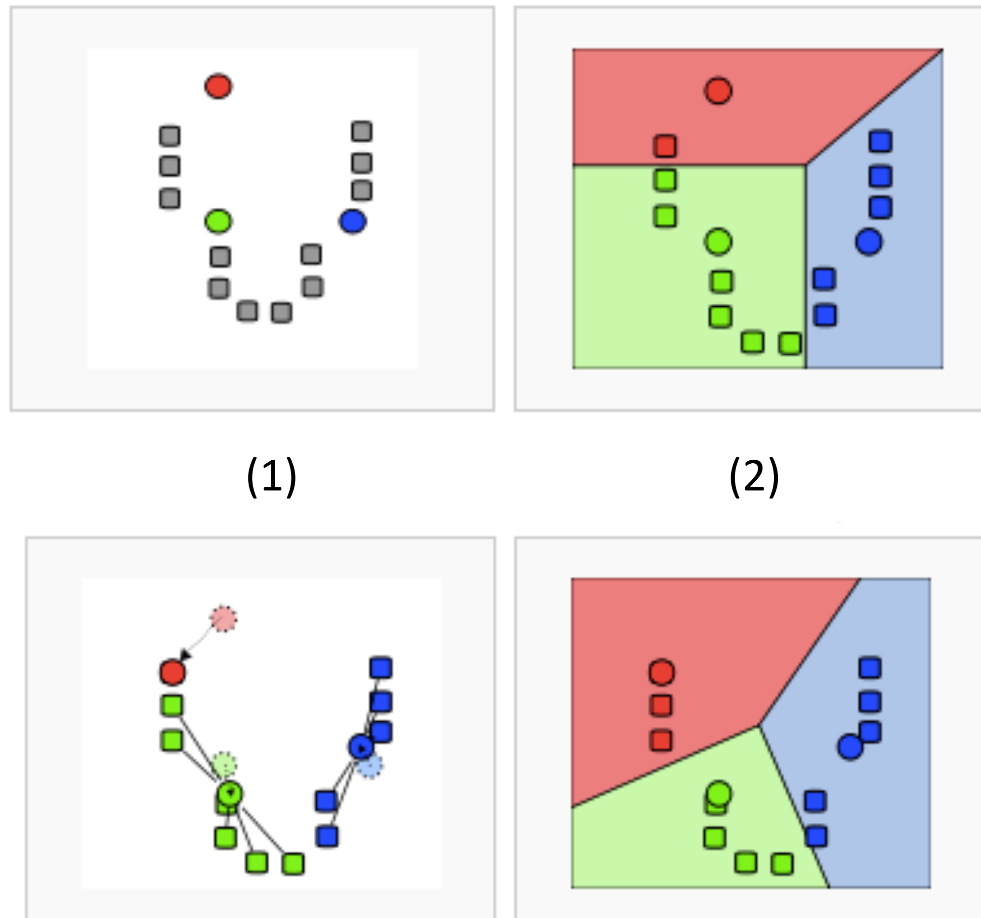
Hierarchical Clustering example

- Agglomerative clustering of matrix of gene-tissue pairs of human samples.
- Columns are tissues; rows are genes
- Clustering is done for both directions



K-means clustering

- Pick a value **k** as the number of clusters
- Select **k** random cluster centers
- Iterate until convergence:
  - Assign each data to the nearest center
  - Update the center within the cluster



(1) (2)

(3) Source:wikipedia (4)

Q. What are the values of c1 and c2?

- Given a dataset {0,2,4,6,24,26}, initialize the k - means clustering algorithm with 2 cluster centers c1= 3 and c2 = 4. What are the values of c1 and c2 after **one** iteration of k-means?

Q. What are the values of c_1 and c_2?

- Given a dataset {0,2,4,6,24,26}, initialize the k - means clustering algorithm with 2 cluster centers $c1= 3$ and $c2 = 4$. What are the values of c1 and c2 after **two** iterations of k-means?

What does k-means do mathematically?

- It's an minimization of a cost function

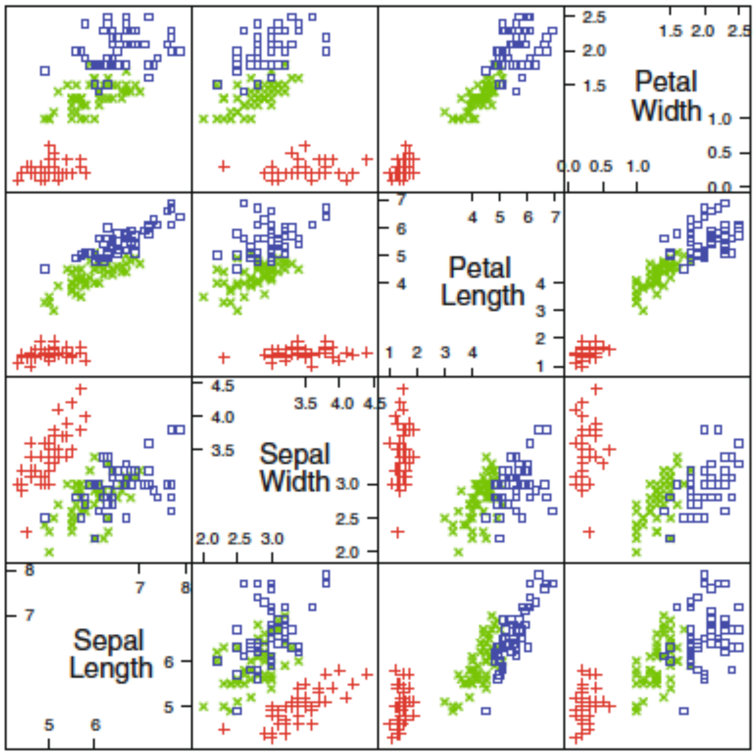
$$\boldsymbol{\phi}(\delta, \boldsymbol{c}) = \sum_{i,j} \delta_{i,j}[(\boldsymbol{x}_i - \boldsymbol{c}_j)^T(\boldsymbol{x}_i - \boldsymbol{c}_j)]$$

$$= \sum_{i}^{N} \sum_{j}^{k} \delta_{i,j} \left\| \boldsymbol{x}_i - \boldsymbol{c}_j \right\|^2 \quad \delta_{i,j} = \begin{cases} 1 & if \ \boldsymbol{x}_i \in cluster \ j \\ 0 & otherwise \end{cases}$$

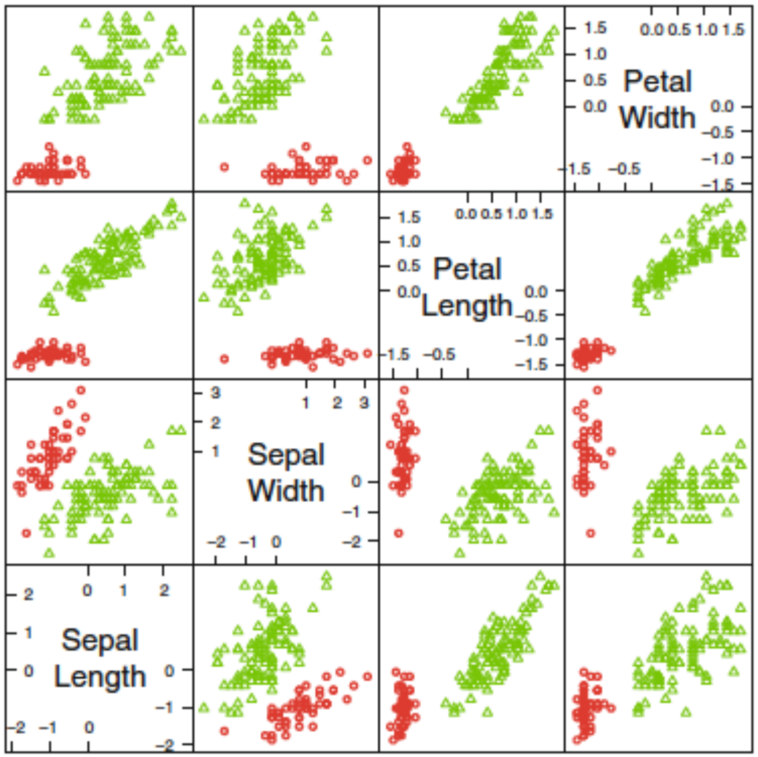
- Cost is defined by the sum of squared distances of each data point from its cluster center

K-means clustering example: Iris

True labels

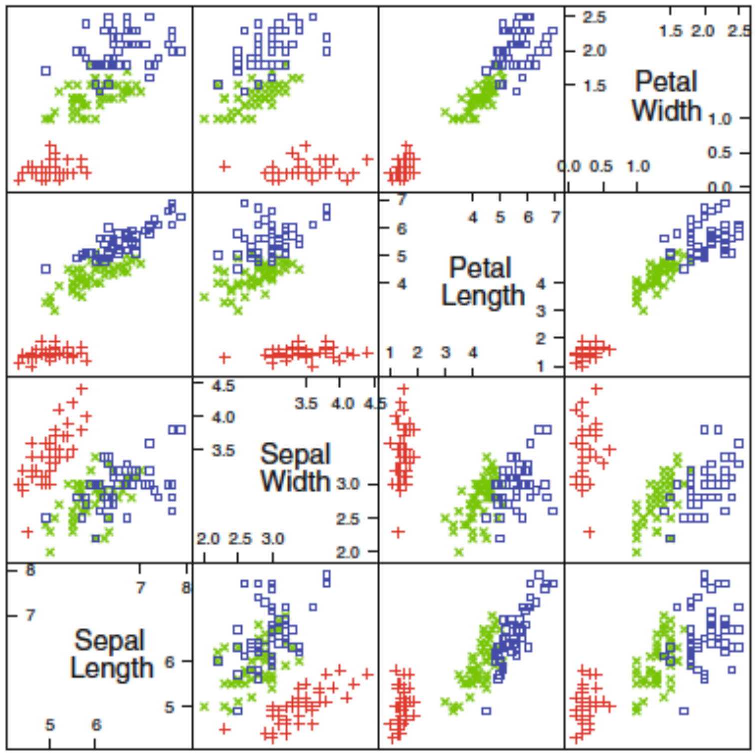


2 clusters

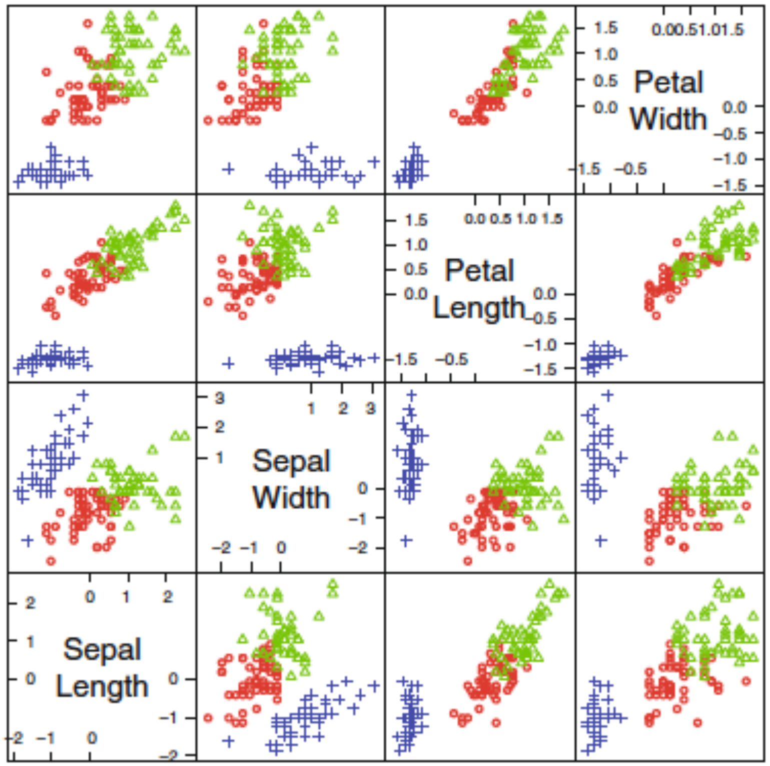


K-means clustering example: Iris

True labels

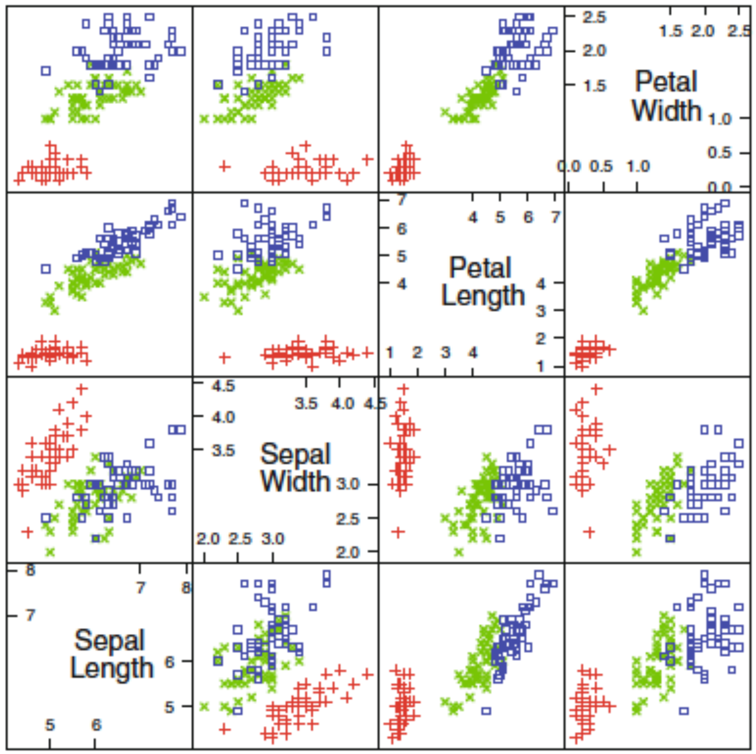


3 clusters

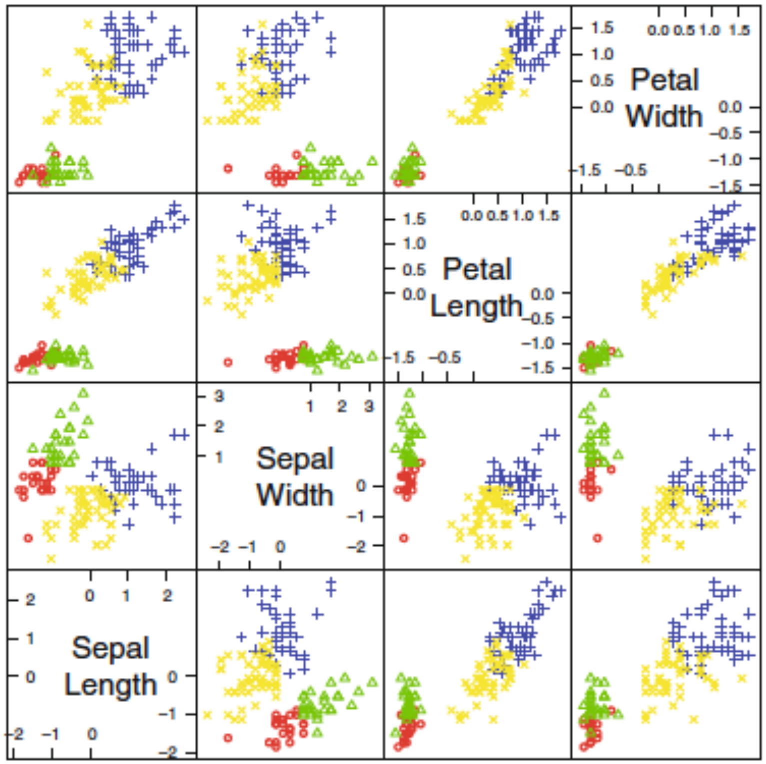


K-means clustering example: Iris

True labels



4 clusters



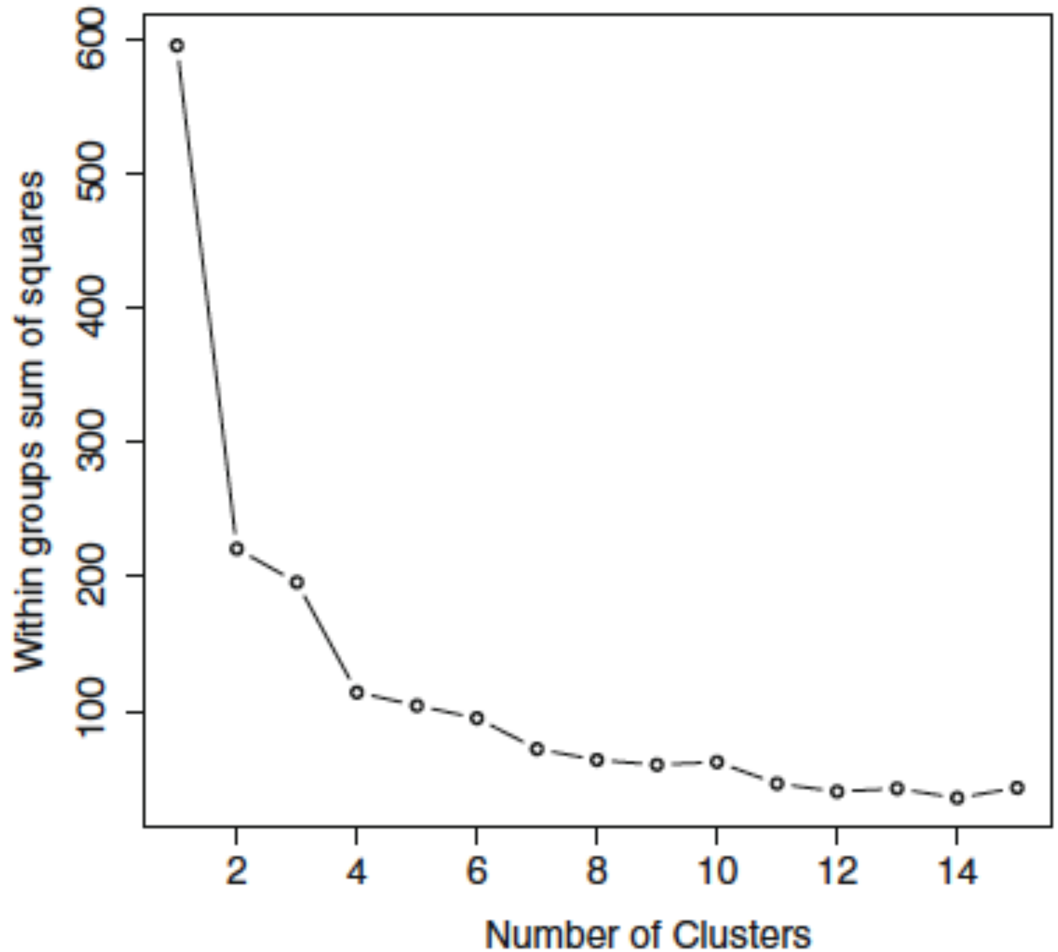
How to choose the value of k?

- Sometimes we have the knowledge from the data set.
- Sometimes we have some other natural way to choose k.
- Otherwise given the cost function, we may perform clustering for many k values and choose k from the knee of the cost function empirically.

Choose k from the cost function curve

Which is best?
Still depends on the application

Usually we want fewer clusters.



Some variants of k-means clustering

- Soft assignment allows some data items to belong to multiple clusters with weights associated with each cluster
- Hierarchical k-means speeds up clustering for very large datasets
- K-medioids allows clustering of data that cannot be averaged

Q. What is different between a hierarchical clustering (hc) and k-means?

A. HC produces dendrogram while k-means results in only flat clusters.
B. HC doesn't need to choose number of clusters while k-means needs that step.
C. HC has higher order time complexity than k-means
D. All the above.

K-means clustering example: Portugal consumers

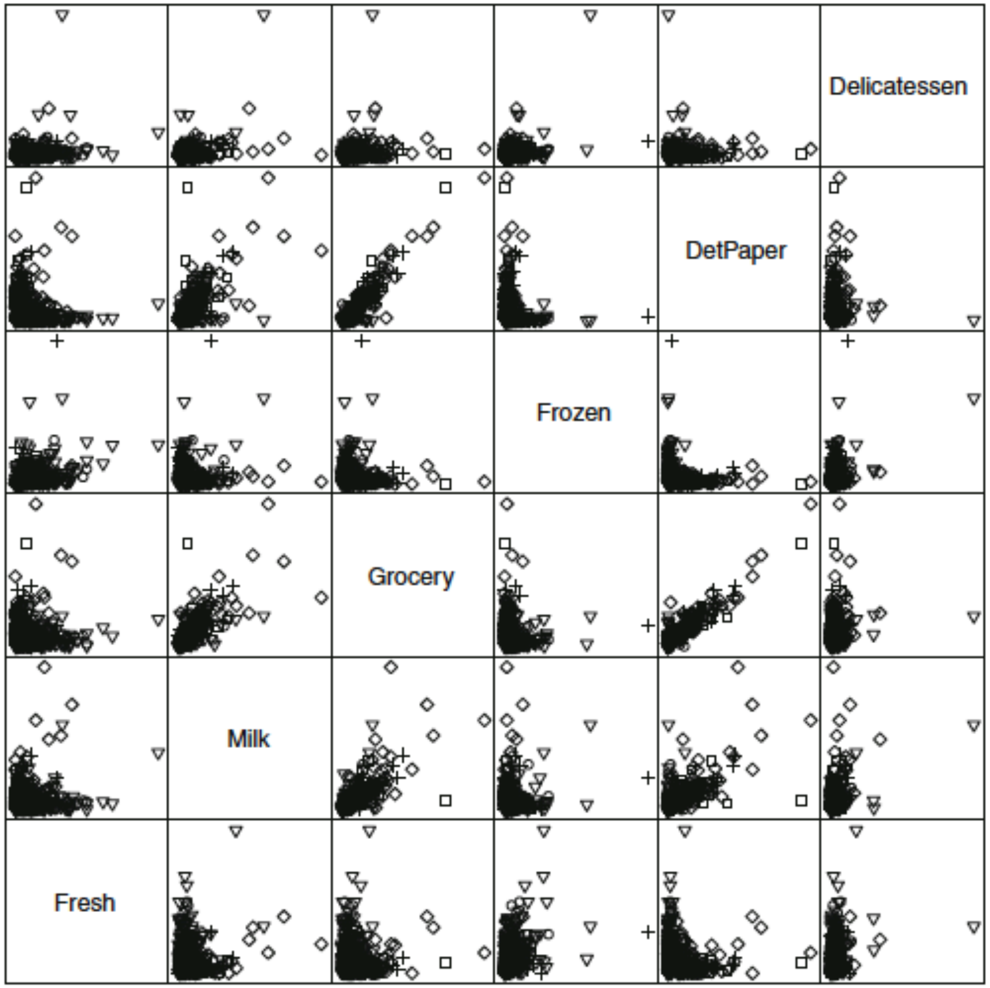
- The dataset consists of the annual grocery spending of 440 customers
- Each customer's spending is recorded in 6 features:
  - fresh food, milk, grocery, frozen, detergents/paper, delicatessen
- Each customer is labeled by: 6 labels in total
  - Channel (Channel 1 & 2) (Horeca 298, Retail 142)
  - Region (Region 1, 2 &3) (Lisbon 77, Oporto 47, Other 316)

Lisbon, Portugal

Oporto, Portugal

Visualization of the data

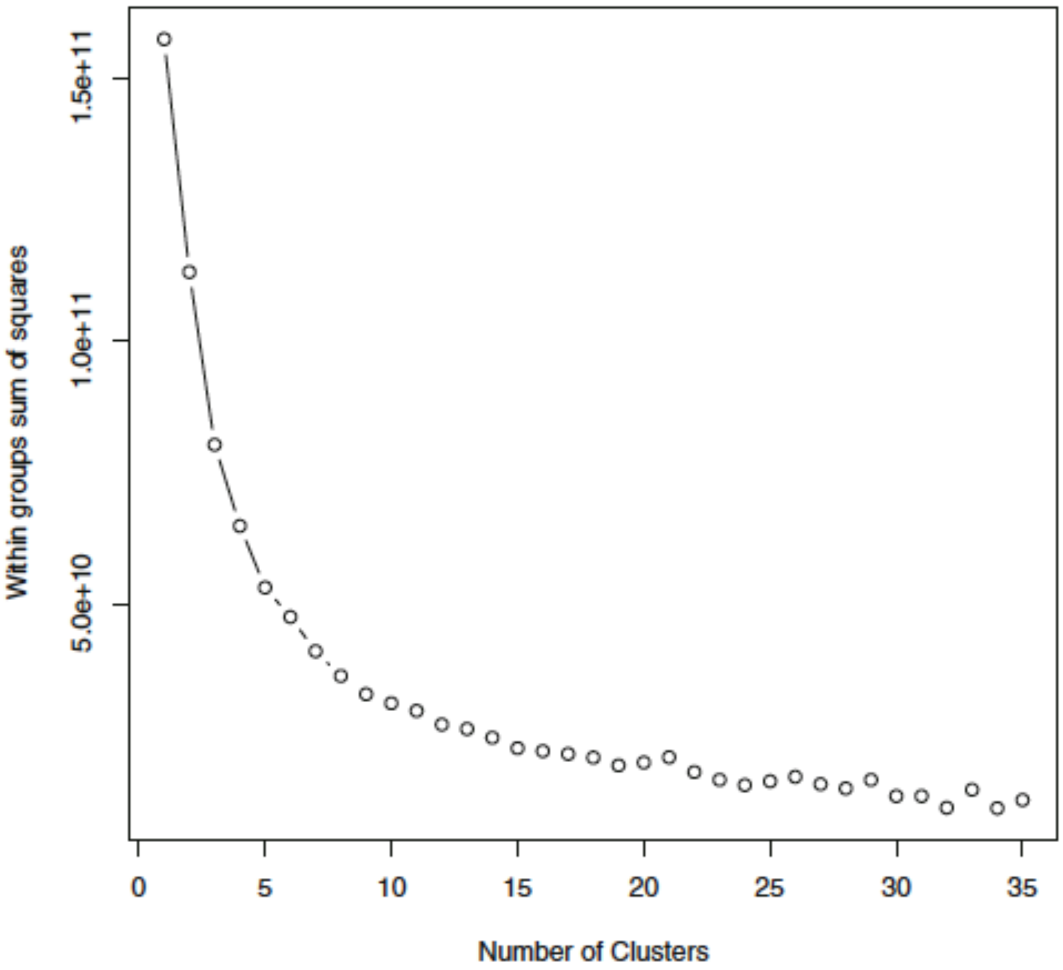
- Visualize the data with scatter plots
- We do see that some features are correlated.
- But overall we do not see significant structure or groups in the data.



Scatter Plot Matrix

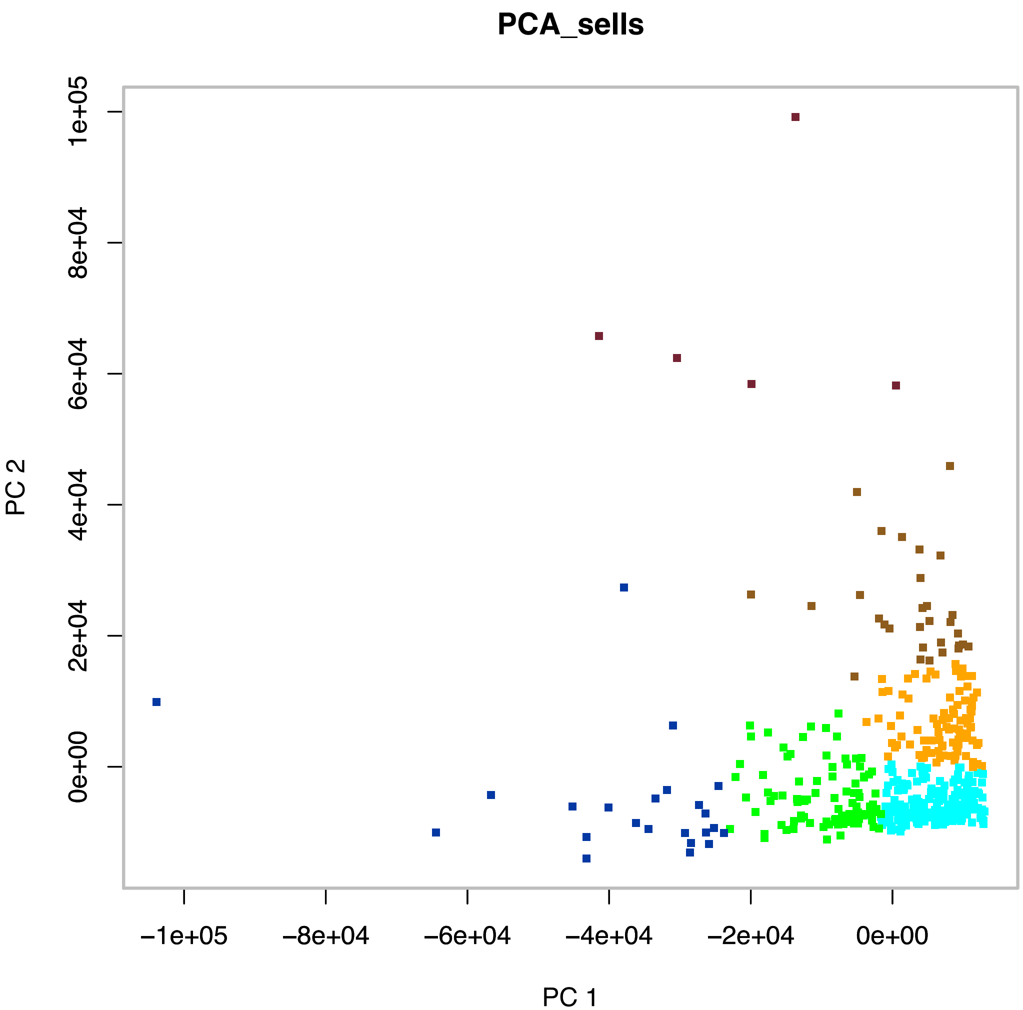
Do kmeans and choose k through the cost function

It's good to pick a **k** around the knee:
I choose 6 for it matches the number of labels



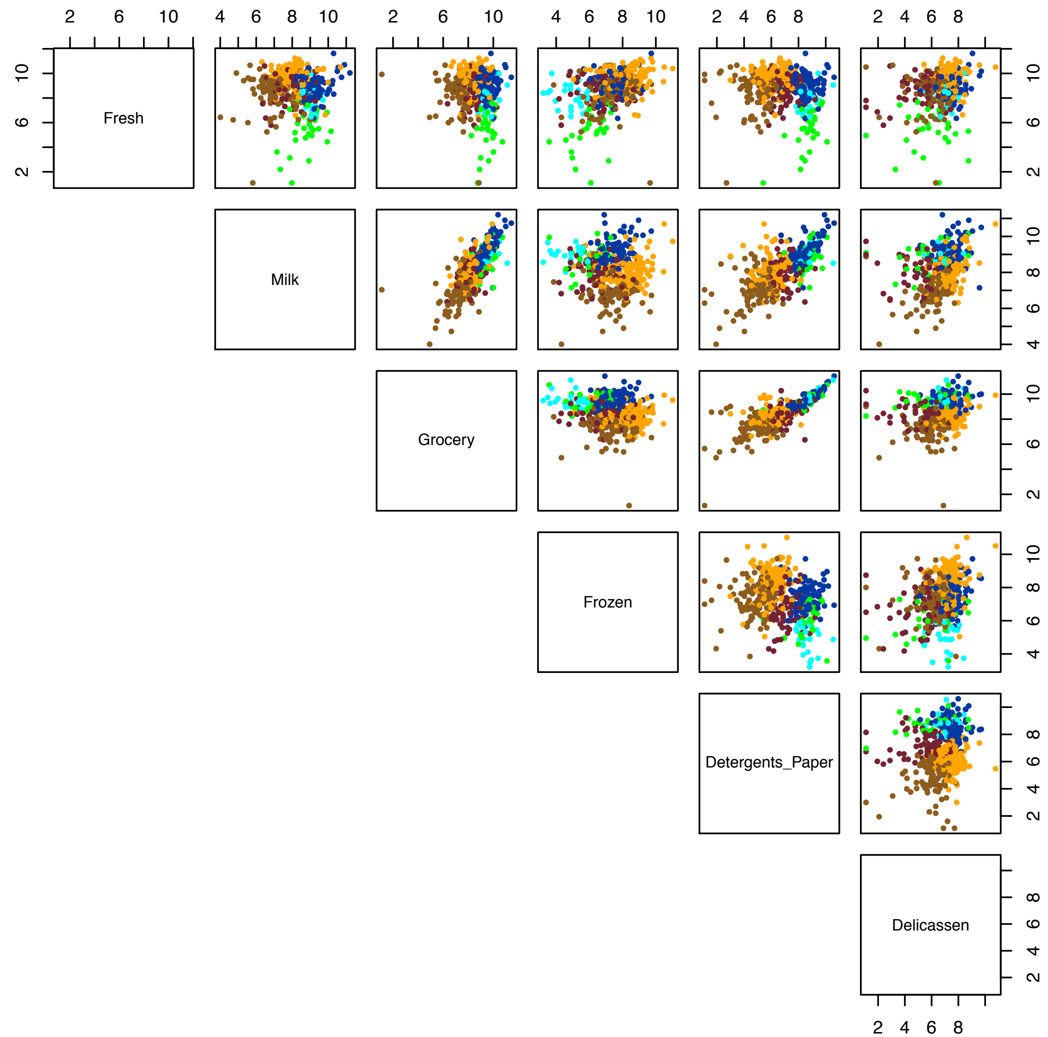
Visualization of the data (PCA)

- PCA does show some separation. **Colors are the clusters**
- Data points show large range of dynamics!



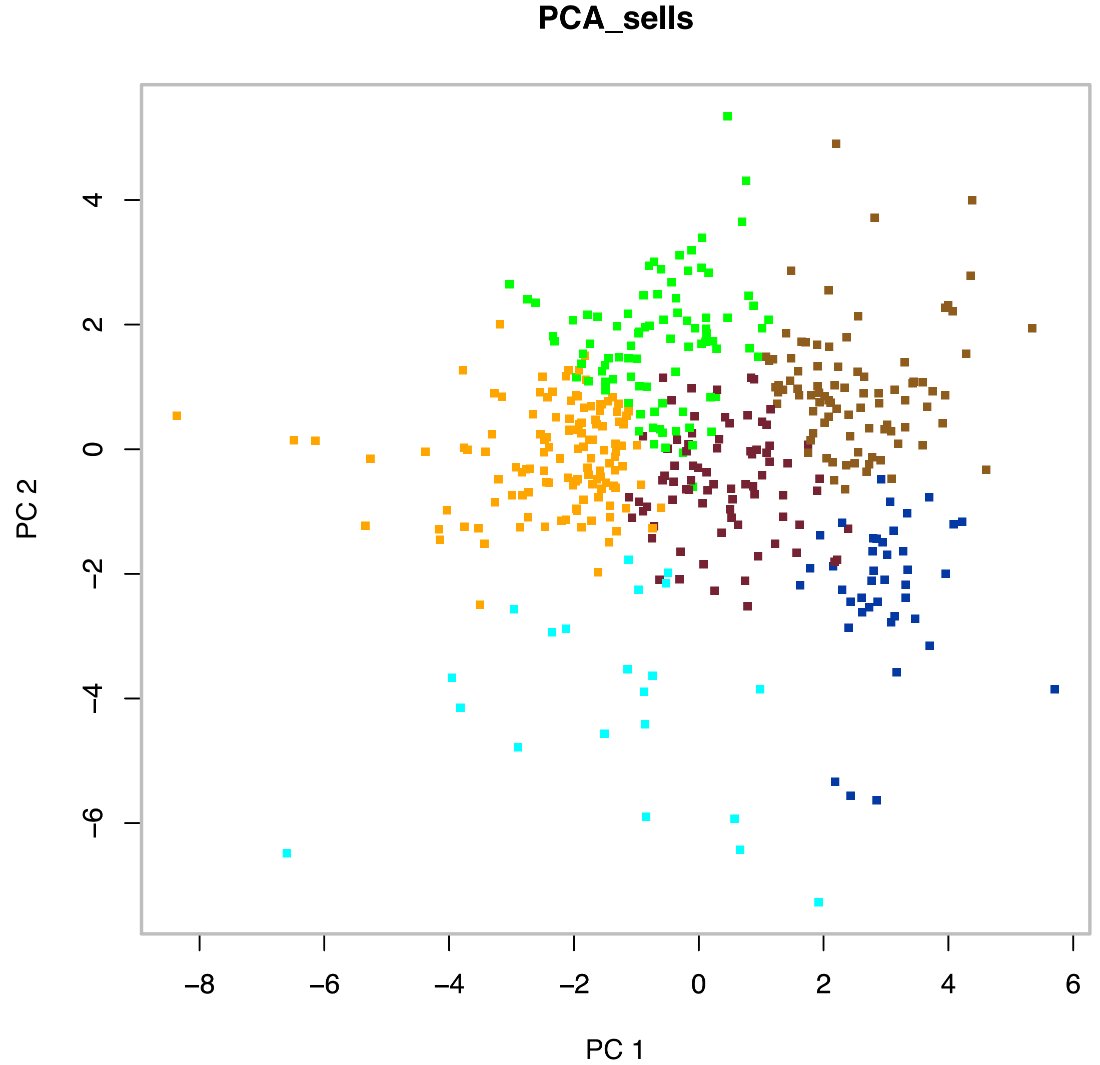
Do log transform of the data

- Log transform the data
- Do scatter plot matrix after the log transform
- Do the kmeans and color the clusters identified by k-means



PCA after log transformation: Clusters

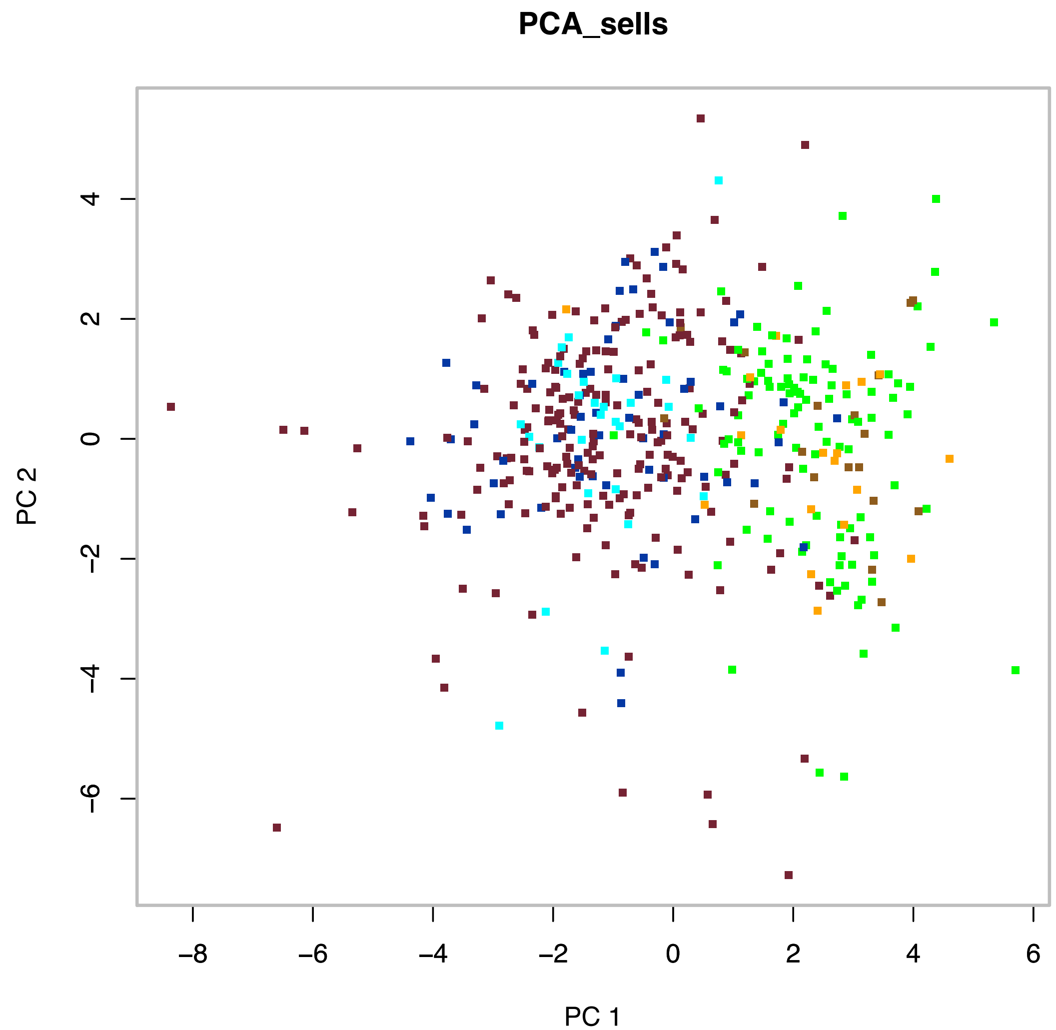
Colors show the **clusters** identified by k-means



PCA after log transformation

Colors show the **Channel-region labels**

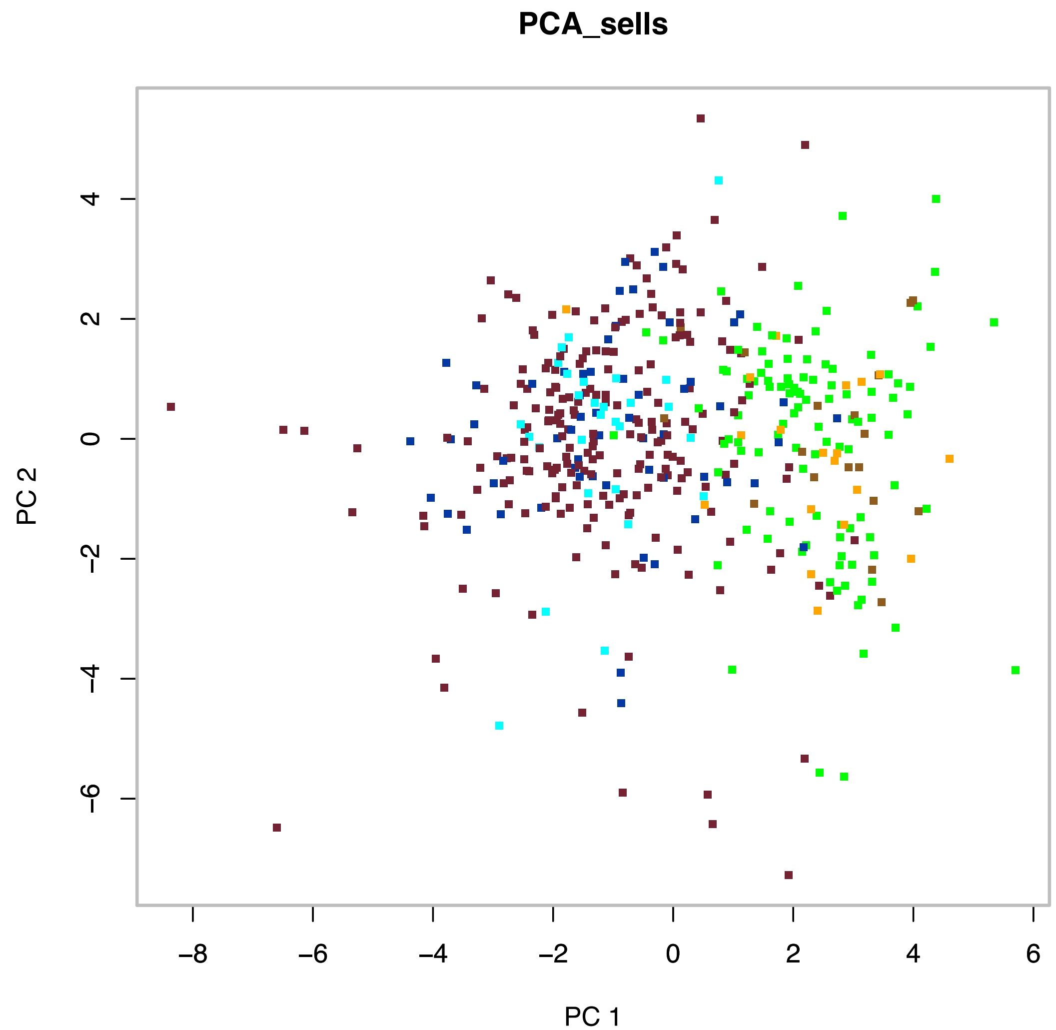
What does this tell us?



PCA after log transformation

Colors show the **Channel-region labels**

Channels differ a lot



Assignments

- Read Chapter 11 of the textbook
- Next time: Clustering (II) & intro. Of Markov Chain

Additional References

- Robert V. Hogg, Elliot A. Tanis and Dale L. Zimmerman. “Probability and Statistical Inference”
- Kelvin Murphy, “Machine learning, A Probabilistic perspective”

See you next time

See You!