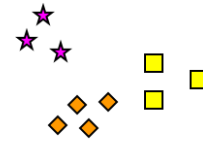
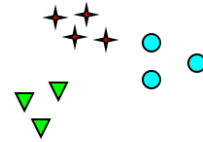
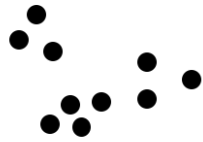
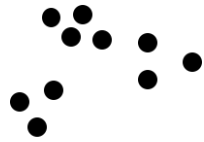


# ANOVA continued, PCA and clustering

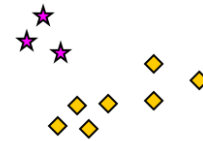
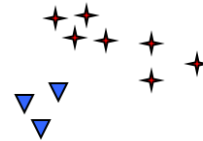
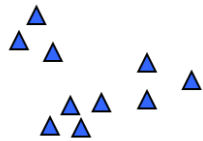
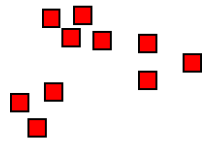


So tell me how many clusters do you see?



How many clusters?

Six Clusters



Two Clusters

Four Clusters

# Overview

## Review and continuation of ANOVAs

- Unbalanced data
- Repeated measures/block designs and random effects models

## Principal components analysis (PCA)

If there is time: Clustering

# ANOVA review

An Analysis of Variance (ANOVA) can be viewed as:

- A hypothesis test comparing multiple means
- A model for predicting means from categorical variables

In a **factorial ANOVA**, we model the response variable  $y$  as a function of **more than one** categorical predictor

For a two-way ANOVA we have:

$$y_{ijk} = \mu + \alpha_j + \beta_k + \gamma_{jk} + \varepsilon_{ijk}$$

$y_{ijk}$  is the  $i^{\text{th}}$  response when:

- factor A has level  $j$
- Factor B has level  $k$

$\mu$  is the Overall mean

$\alpha_j$  is the Main effect for factor A at level  $j$

$\beta_k$  is the Main effect for factor B at level  $k$

$\gamma_{jk}$  is the Specific interaction for  $j^{\text{th}}$  level of A and  $k^{\text{th}}$  level of B

$\varepsilon_{ijk}$  is the Random error for the  $ijk^{\text{th}}$  data point

# Two-way ANOVA hypotheses

## Main effect for A

$$H_0: \alpha_1 = \alpha_2 = \dots = \alpha_j = 0$$

$$H_A: \alpha_j \neq 0 \text{ for some } j$$

## Main effect for B

$$H_0: \beta_1 = \beta_2 = \dots = \beta_K = 0$$

$$H_A: \beta_k \neq 0 \text{ for some } k$$

## Interaction effect:

$$H_0: \text{All } \gamma_{jk} = 0$$

$$H_A: \gamma_{jk} \neq 0 \text{ for some } j, k$$

Where:

$\alpha_j$ : is the “effect” for factor A at level j

$\beta_k$ : is the “effect” for factor B at level k

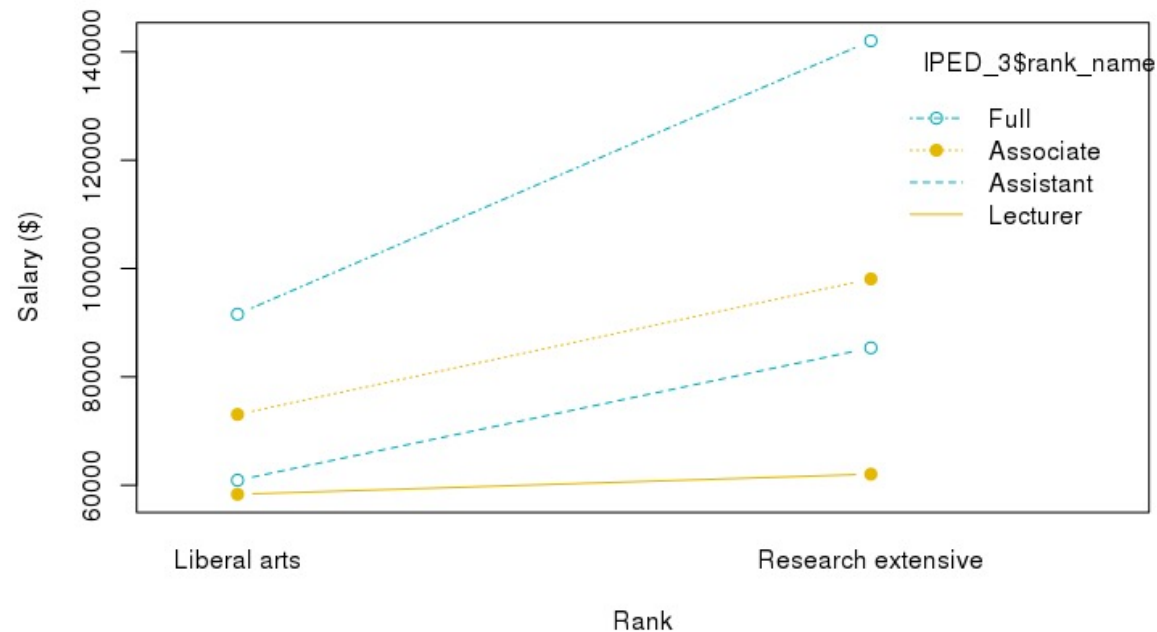
$\gamma_{jk}$ : is the interaction between level j of factor A, and level k of factor B.

$$y_{ijk} = \mu + \alpha_j + \beta_k + \gamma_{jk} + \varepsilon_{ijk}$$

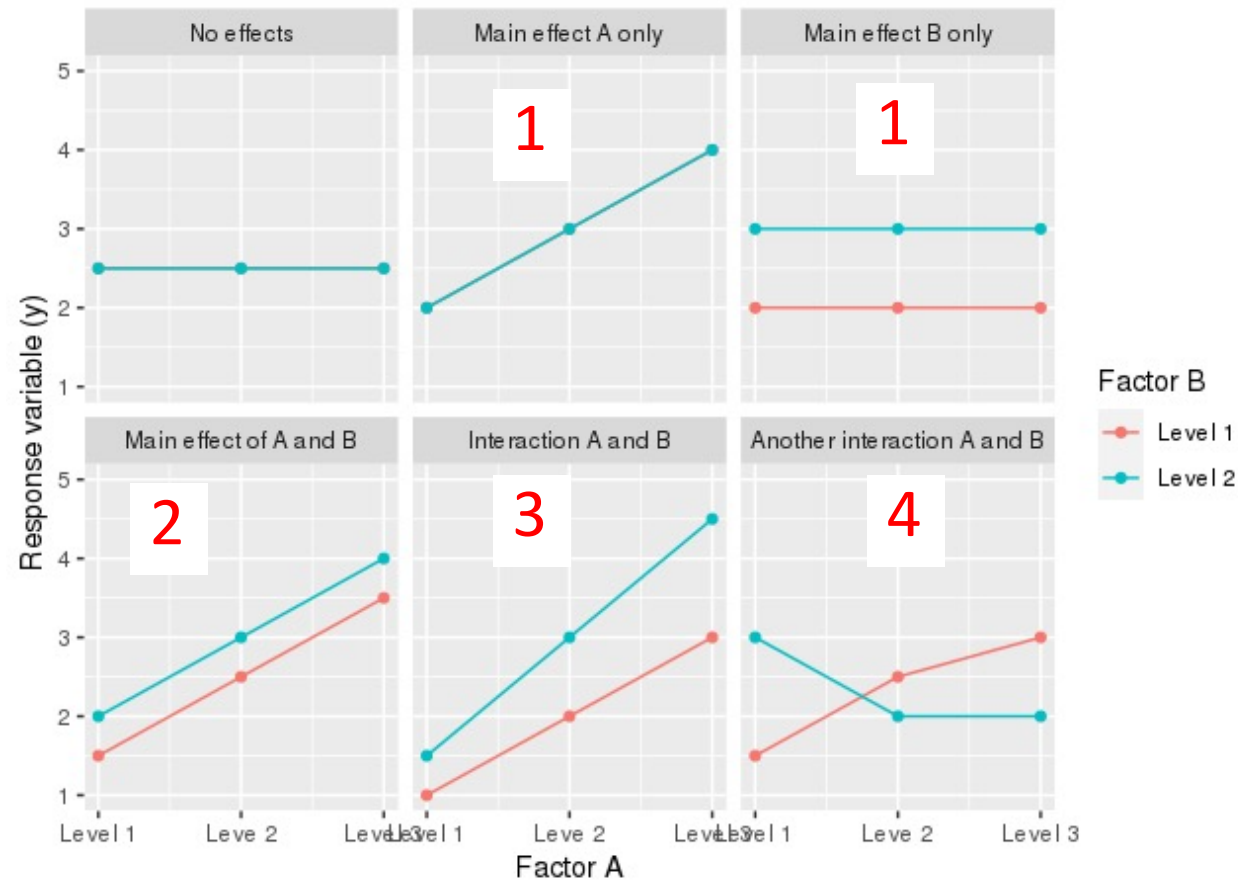
# Interaction plots

Interaction plots can help us visualize main effects and interactions

- Plot the levels of one of the factors on the x-axis
- Plot the levels of the other factor as separate lines

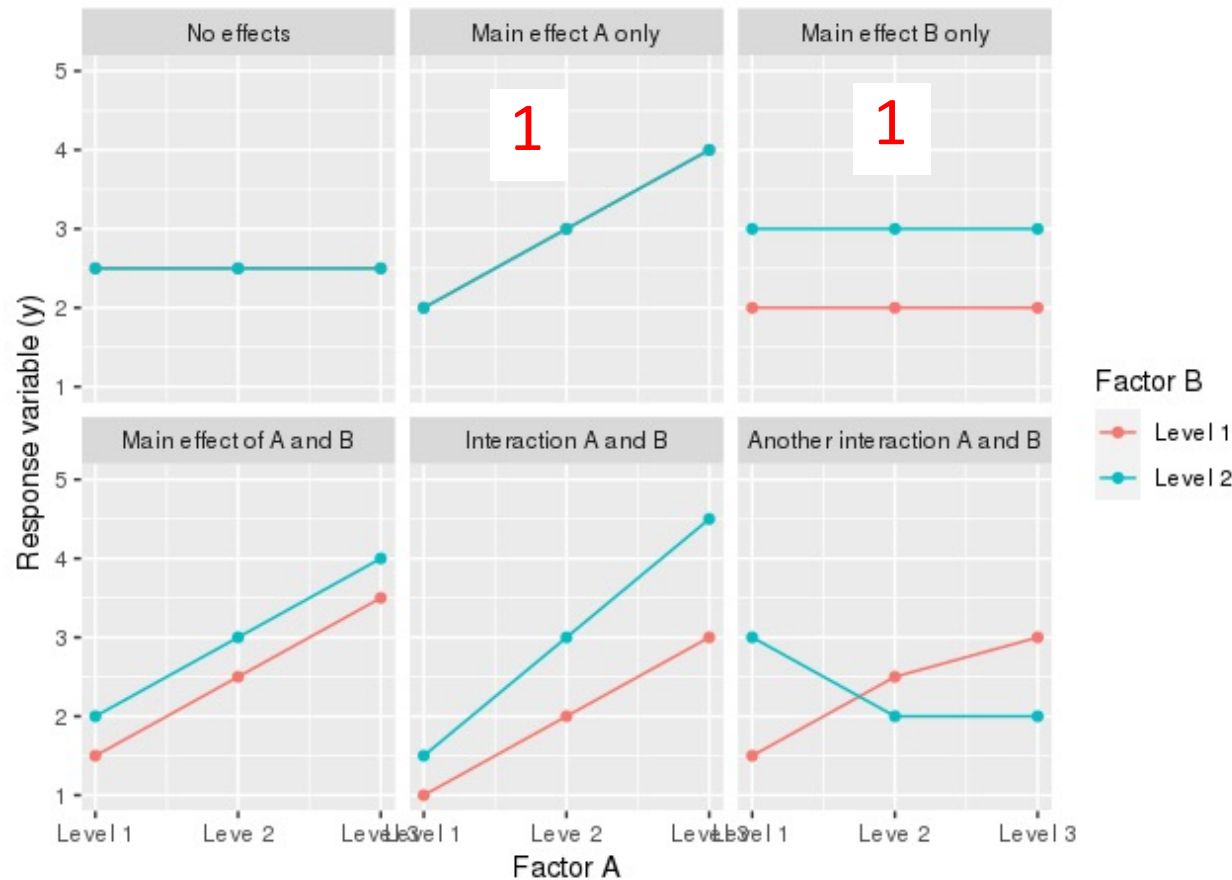


# Interpreting interaction through interaction plots



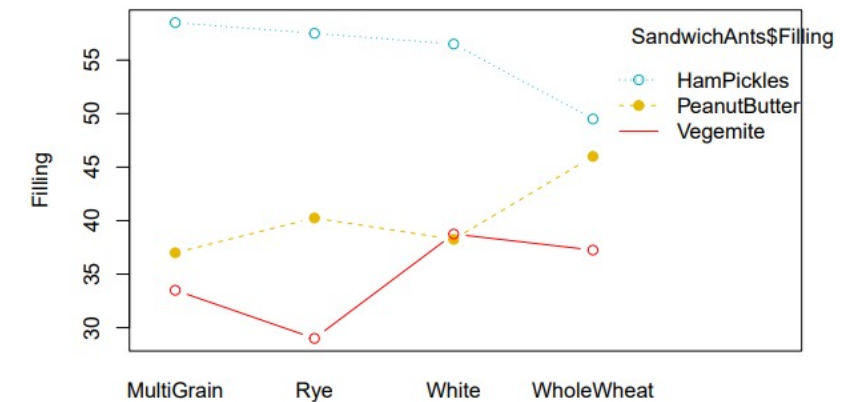
What are examples we have seen in class of the interactions in plots 1, 2, 3 and 4?

# Interpreting interaction through interaction plots



What are examples we have seen in class of 1 (main effect only in one factor)?

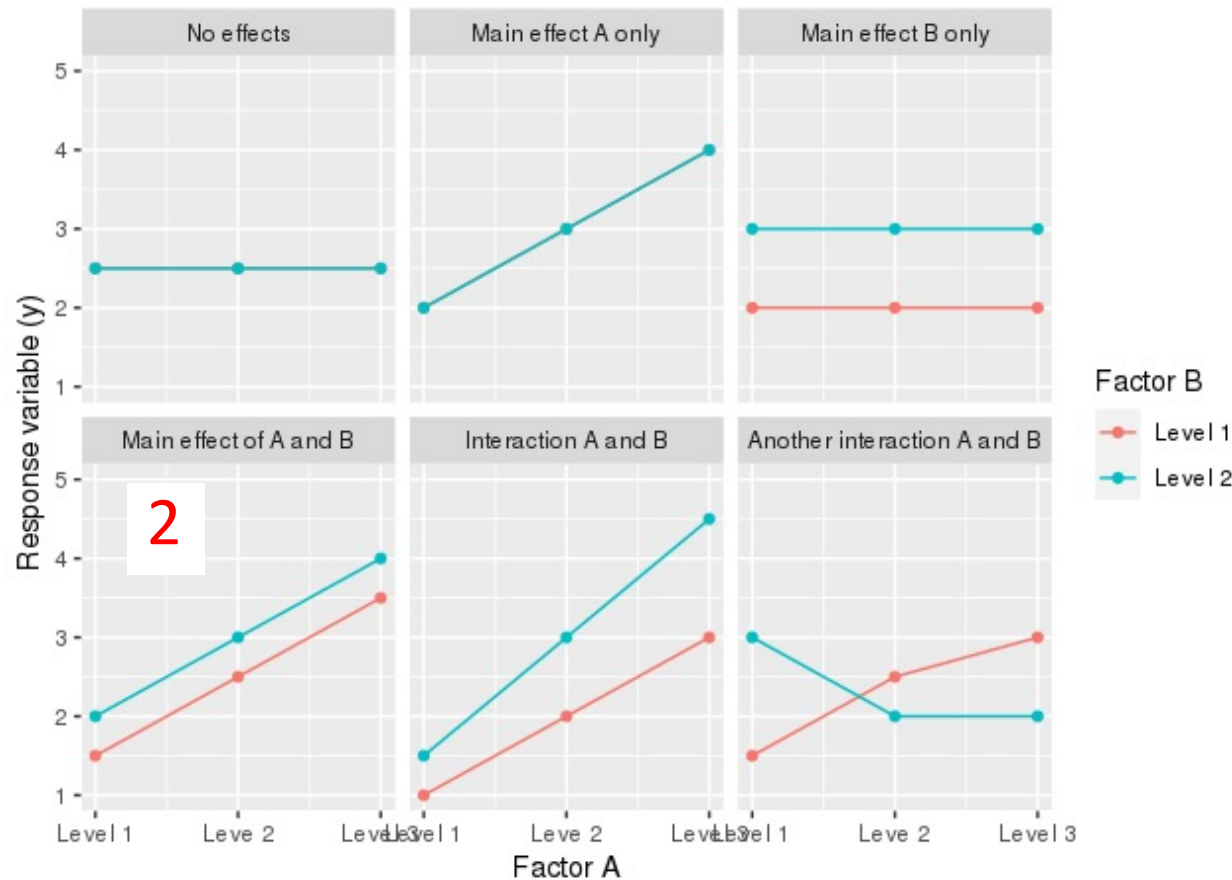
1: Ants sandwich preferences



$$y_{ijk} = \mu + \beta_k + \varepsilon_{ijk}$$

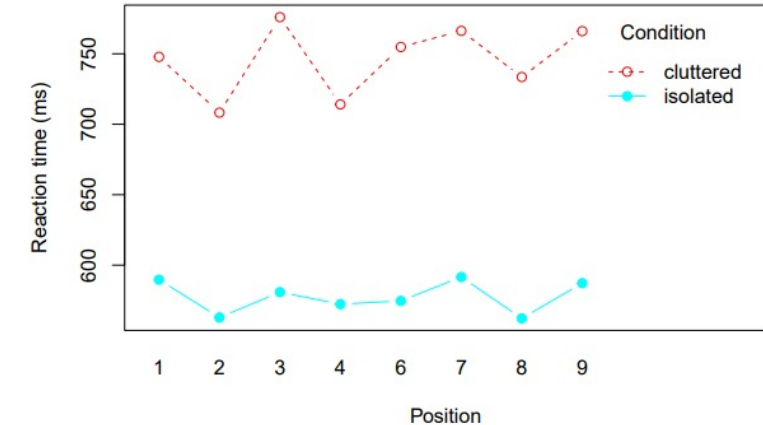
##		Df	Sum Sq	Mean Sq	F value	Pr(>F)
##	Bread	3	40.5	13.50	0.0754	0.9728619
##	Filling	2	3720.5	1860.25	10.3860	0.0002748 ***
##	Bread:Filling	6	577.0	96.17	0.5369	0.7765447
##	Residuals	36	6448.0	179.11		

# Interpreting interaction through interaction plots



What are examples we have seen in class of 2 (main effect in both factors, no interaction)?

## 2: Pop-out attention

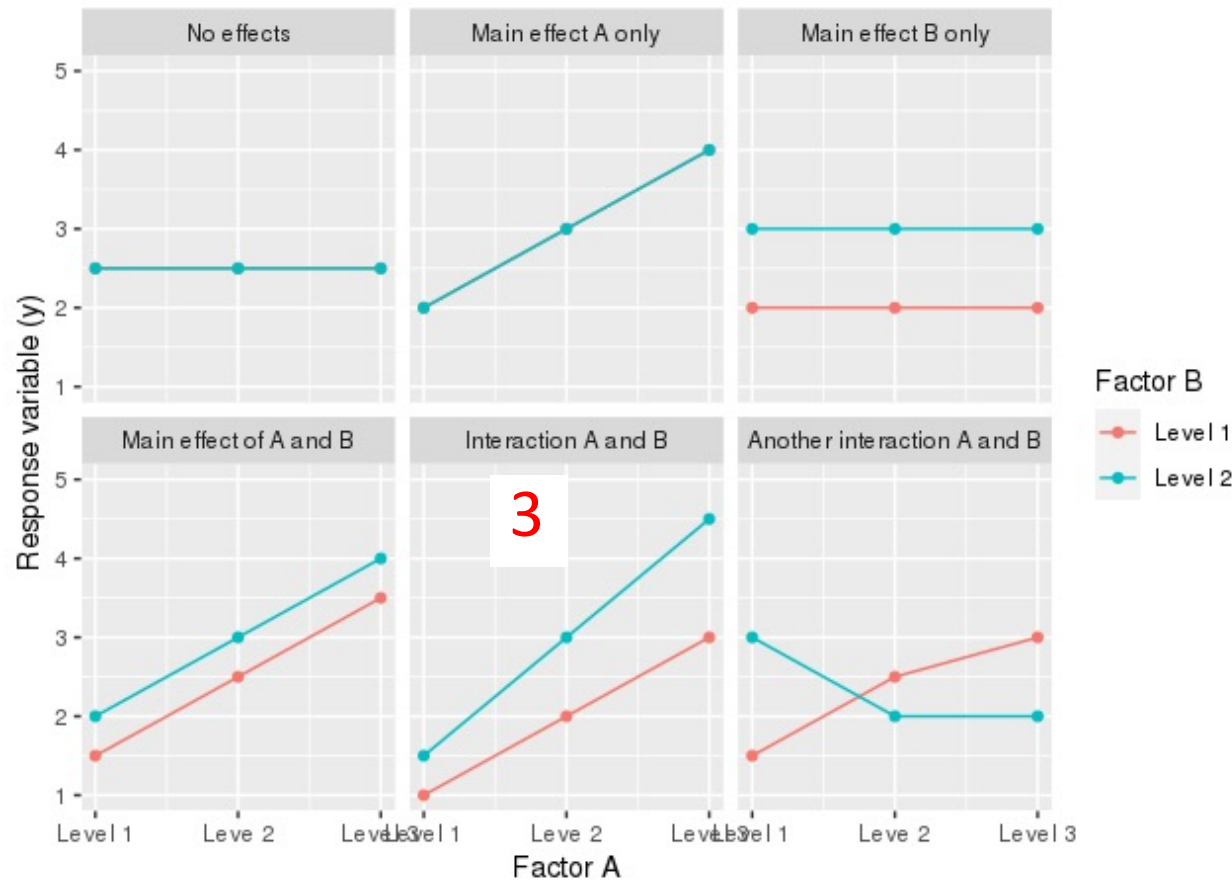


##	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## condition	1	77.63	77.63	1457.276	< 0.0000000000000002 ***
## position	7	2.83	0.40	7.579	0.00000000406 ***
## condition:position	7	0.56	0.08	1.504	0.161
## Residuals	5054	269.25	0.05		

$$y_{ijk} = \mu + \alpha_j + \beta_k + \varepsilon_{ijk}$$

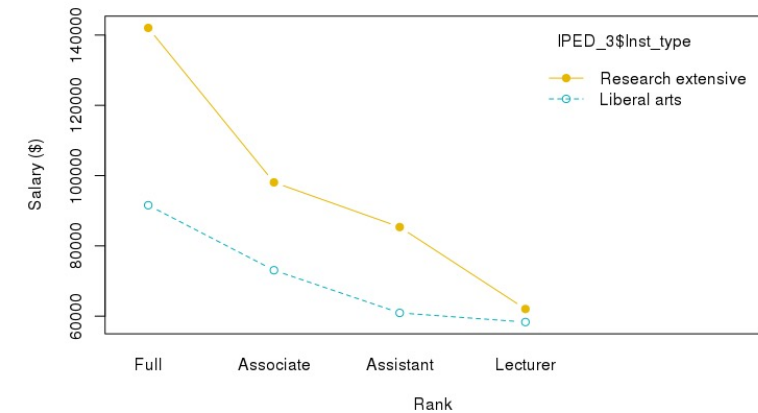


# Interpreting interaction through interaction plots



What are examples we have seen in class of 3 (main effects and interaction)?

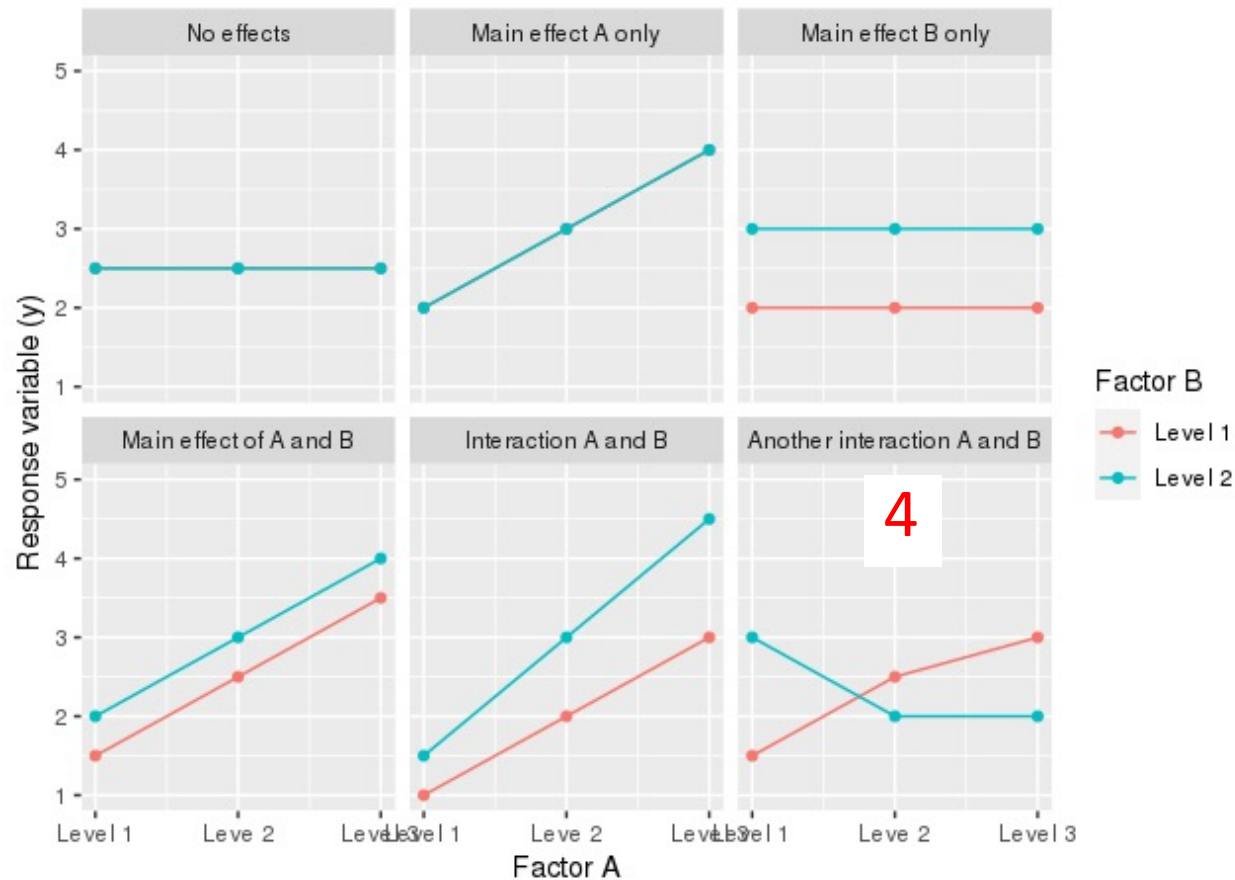
## 3: Faculty salaries



$$y_{ijk} = \mu + \alpha_j + \beta_k + \gamma_{jk} + \varepsilon_{ijk}$$

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	1752515009149	1	4422.319	< 0.00000000000000022 ***
rank_name	116444984138	3	97.946	< 0.00000000000000022 ***
Inst_type	223284242856	1	563.438	< 0.00000000000000022 ***
rank_name:Inst_type	71691991643	3	60.303	< 0.00000000000000022 ***
Residuals	496153389805	1252		

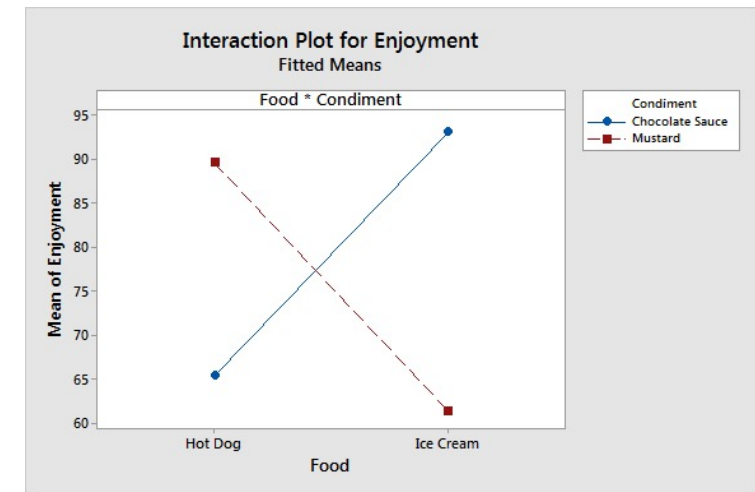
# Interpreting interaction through interaction plots



4

What are examples we have seen in class of 4 (reverse ordering of effects with interaction)?

## 4: Foods and condiments



```
> summary(aov(Enjoyment ~ Food*Condiment, data = condiments_food ))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Food	1	2	2	0.064	0.80136
Condiment	1	278	278	11.071	0.00135 **
Food:Condiment	1	15696	15696	626.153	< 0.0000000000000002 ***
Residuals	76	1905	25		

$$y_{ijk} = \mu + \alpha_j + \beta_k + \gamma_{jk} + \varepsilon_{ijk}$$

# Complete and balanced designs

**Complete factorial design:** at least one measurement for each possible combination of factor levels

- E.g., in a two-way ANOVA for factors A and B, if there are K levels for factor A, and J levels for factor B, then there needs to be at least one measurement for each of the KJ levels

**Balanced design:** the sample size is the same for all combination of factor levels

- E.g., there are the same number of samples in each of the KJ level combinations.
- The computations and interpretations for non-balanced designs are a bit harder.

# Unbalanced designs

For unbalanced designs, there are different ways to compute the sum of squares, and hence one can get different p-values

- The problem is analogous to multicollinearity. If two explanatory variables are correlated either can account for the variability in the response data.

**Type I sum of squares**, (also called sequential sum of squares) the order that terms are entered in the model matters.

- `anova(lm(y ~ A*B))` gives different results than using `anova(lm(y ~ B*A))`
- $SS(A)$  is taken into account before  $SS(B)$  is considered etc.

**Type III sum of squares**, the order that that terms are entered into the model does not matter.

- `Car::Anova(lm(y ~ A*B) , type = "III")` is the same as `car::Anova(lm(y ~ B*A) , type = "III")`
- For each factor,  $SS(A)$ ,  $SS(B)$ ,  $SS(AB)$  is taken into account after all other factors are added

# Repeated measures ANOVA

In a **repeated measures ANOVA**, the same case/observational units are measured at each factor level.

Example: Do people prefer chocolate, butterscotch or caramel sauce?

**Between subjects experiment:** different people rate chocolate, butterscotch or caramel sauce.

- Run a between subjects ANOVA (as we have done before)

**Within subjects experiment:** each person in the experiment gives ratings for all three toppings.

- Run a repeated measures ANOVA

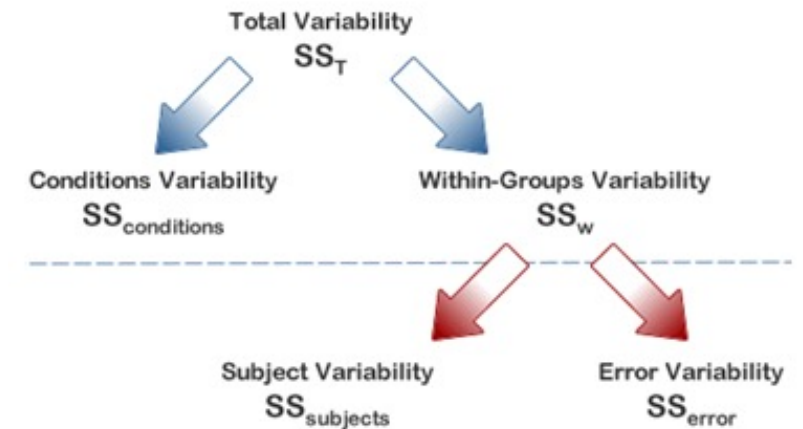
# Repeated measures ANOVA

The advantages of a repeated measures ANOVA is that we can potentially reduce a lot of the variability between the cases

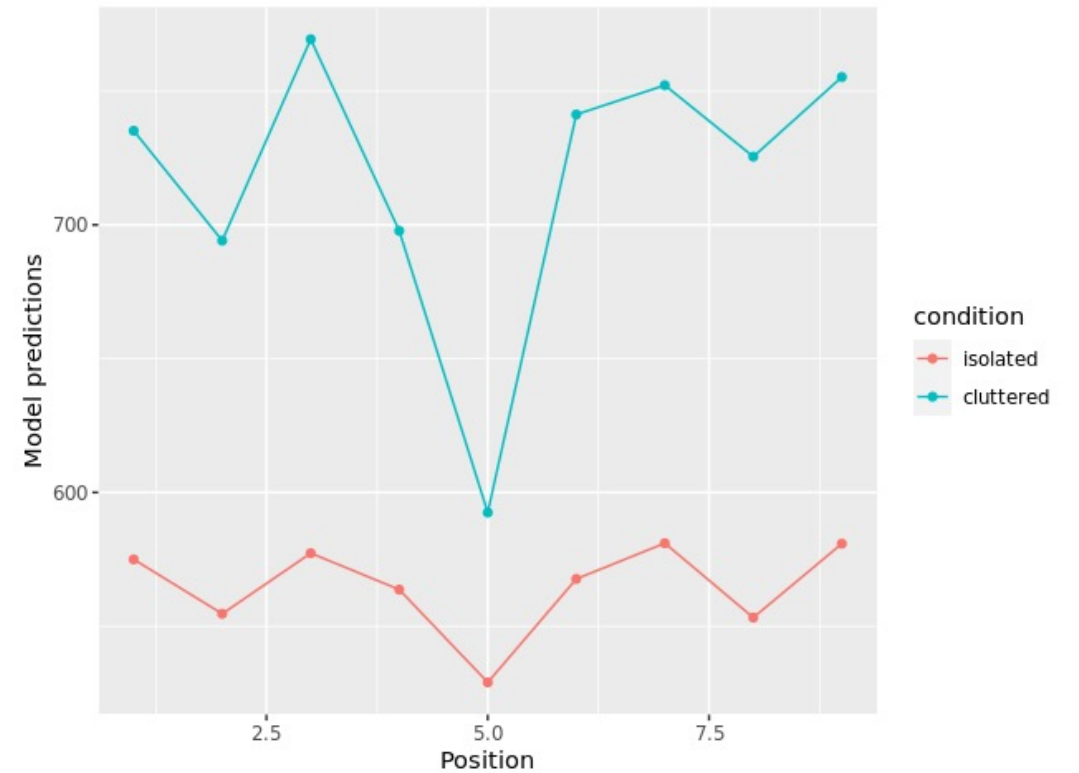
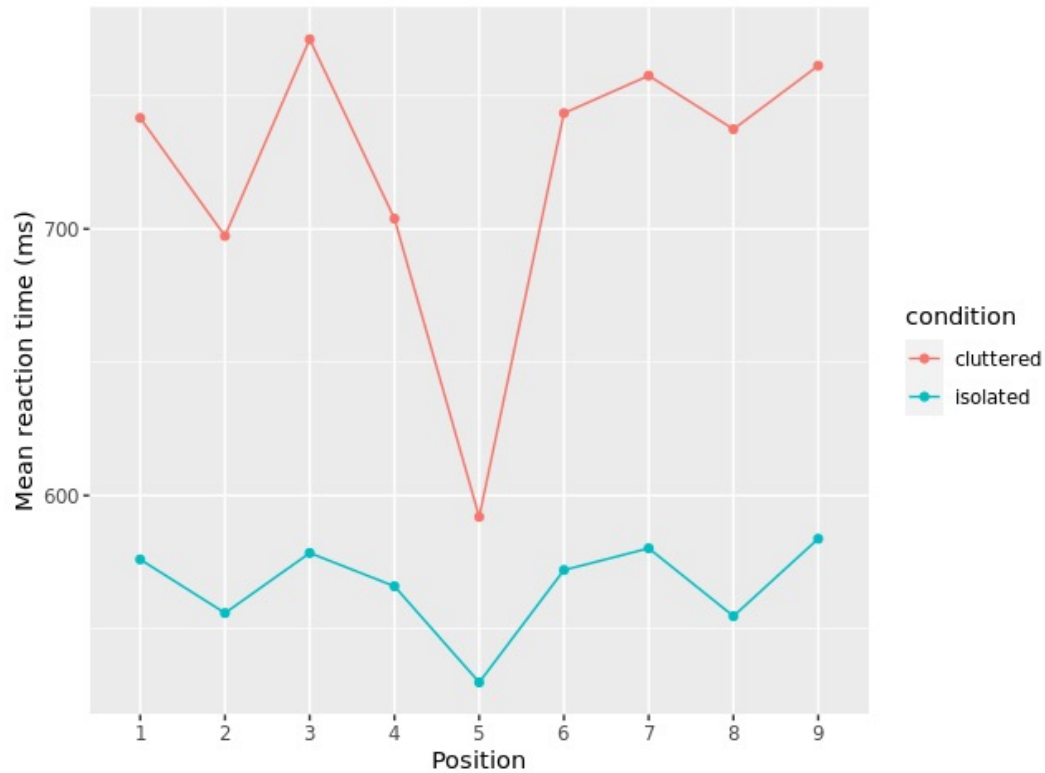
- This is a generalization of a paired t-test to more than two population means

To run a repeated measures ANOVA, we use a factor called ID that has a unique value for each observational unit

```
aov(reaction_time ~ condition * position + participant,  
    data = popout_log_data)
```

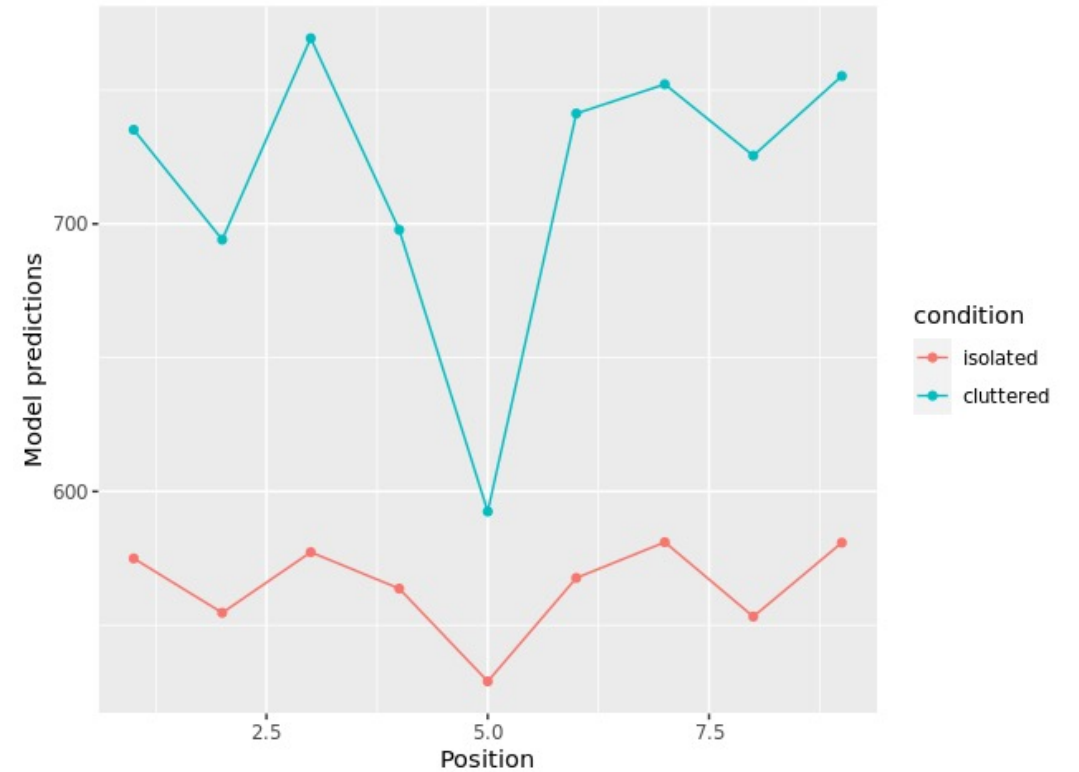
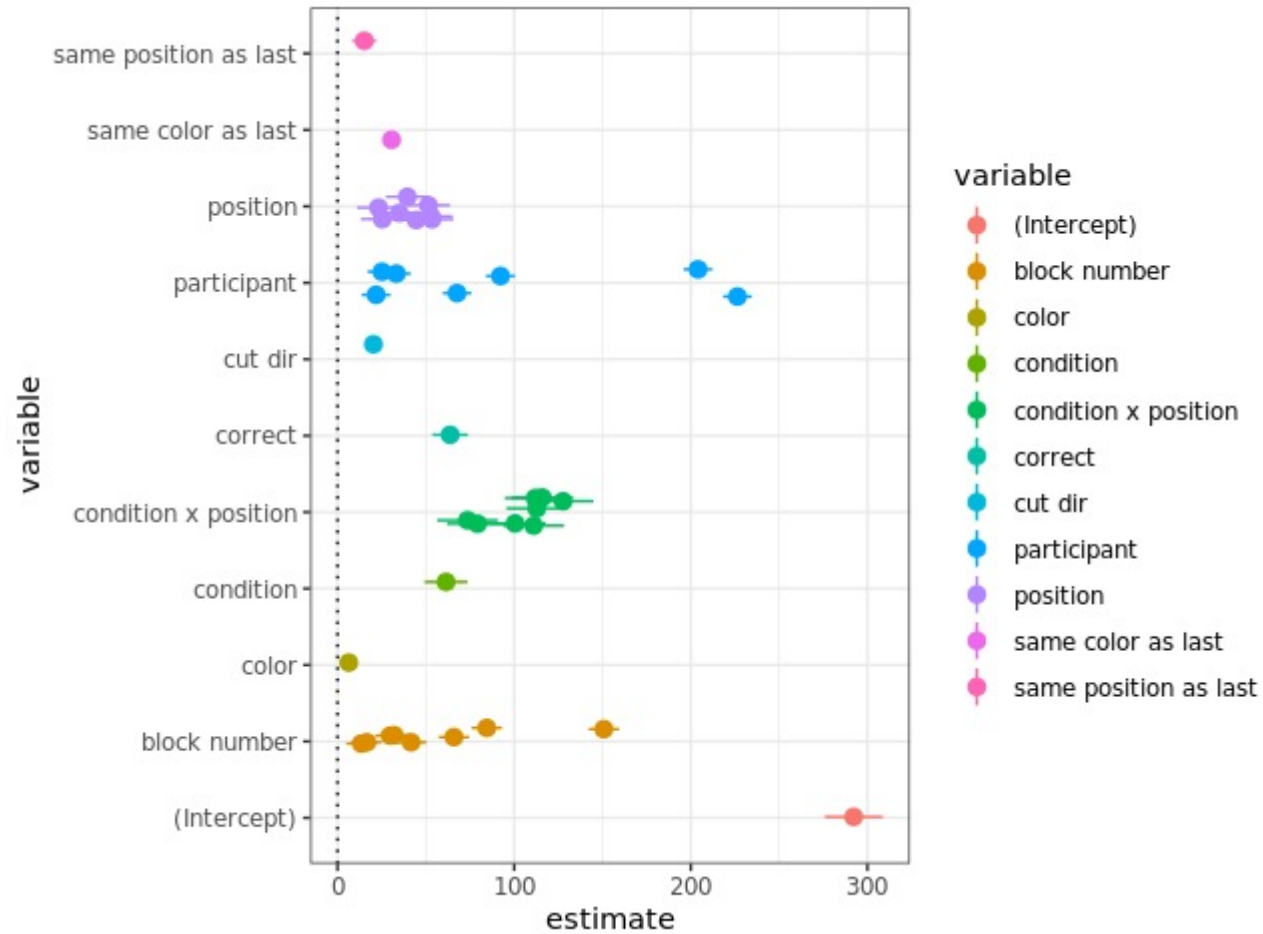


# The ANOVA model – popout data



```
aov(reaction_time ~ condition + position + color + cut_dir + correct + block_number + participant +  
    same_position_as_last + same_color_as_last + position * condition, data = popout_data)
```

# The ANOVA model – popout data



`aov(reaction_time ~ condition + position + color + cut_dir + correct + block_number + participant + same_position_as_last + same_color_as_last + position * condition, data = popout_data)`



# Brief mention: random effects models

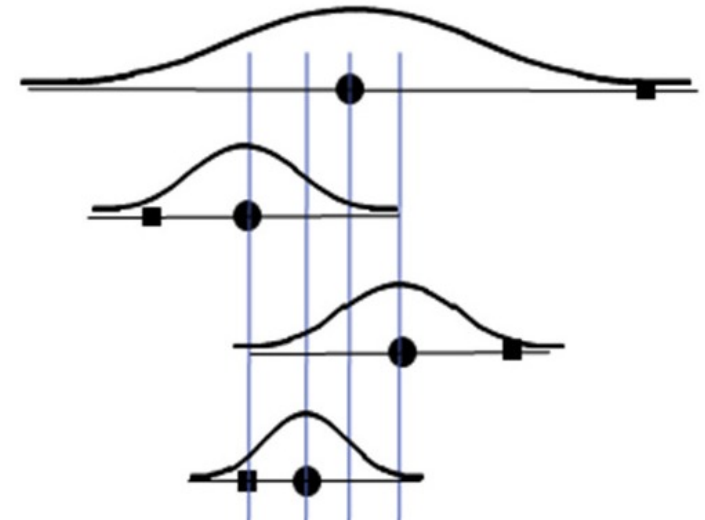
In a random effects ANOVA, factor levels are viewed as being randomly generated from an underlying distribution, rather than having a fixed number of levels.

For example, we could view participants in an experiment as being a random sample from participants in a population.

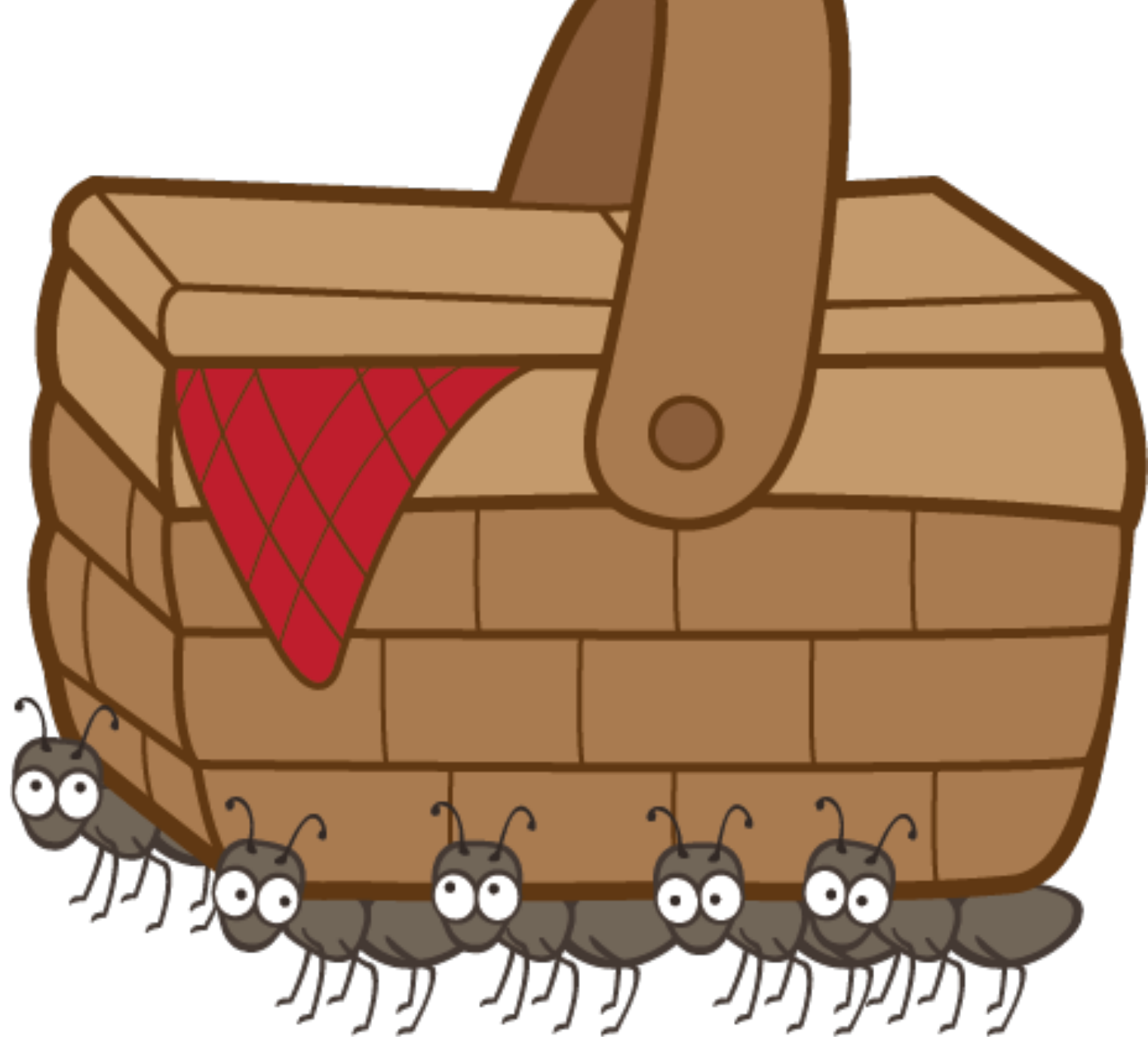
- We then just estimate a mean and standard deviation for the underlying population, rather a separately ID for each participant.
  - This leads to few parameters and hence more degrees of freedom.

You can run mixed effects models in R using the [lme4](#) package

- This is beyond what we will do in this class :/



Let's these  
topics it R...



# Principal Component Analysis

# Supervised learning and unsupervised learning

In **supervised learning** we have a response variable  $y$ , along with explanatory variables  $x_1, x_2, \dots, x_k$

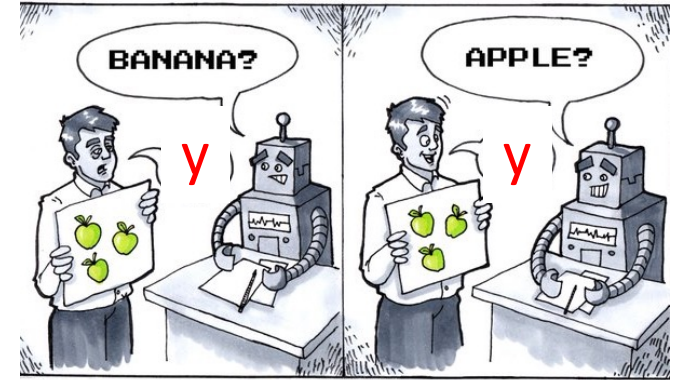
- For example, linear and logistic regression are supervised learning problems because we model a response variable  $y$  as a function of several explanatory variables,  $x_1, x_2, \dots, x_k$

In **unsupervised learning**, we have explanatory variables  $x_1, x_2, \dots, x_k$  but **no** response variable  $y$

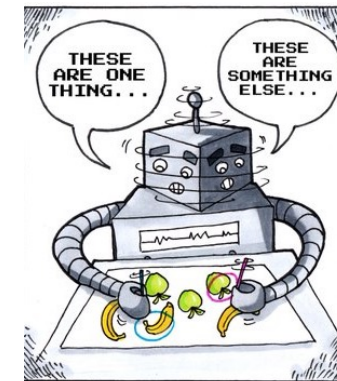
Unsupervised learning can be useful in order to find structure in the data and to visualize patterns

A key challenge in unsupervised learning is that there is no real ground truth response variable  $y$

- So we don't have measures like MSPE to see how well our model is fitting the data



**Supervised Learning**

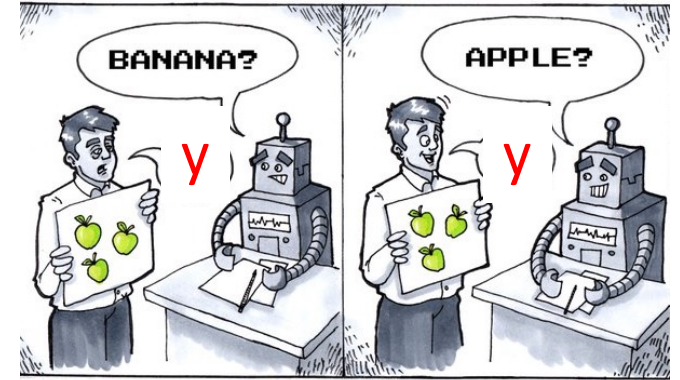


**Unsupervised Learning**

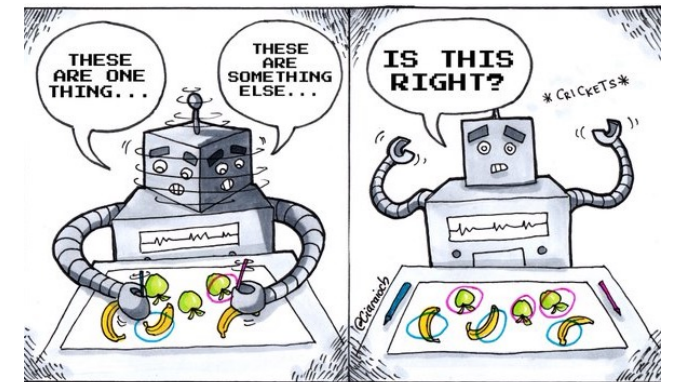
# Unsupervised learning

We will discuss two types of unsupervised learning:

1. **Dimensionality reduction** where we try to find a smaller set of features that captures most of the variability in the data
  - Principal component analysis (PCA)
2. **Clustering** where we try to group similar data points together



**Supervised Learning**



**Unsupervised Learning**

# Principal Component Analysis

**Dimensionality reduction methods** find a new smaller set of explanatory variables that capture key properties in the data:

- $f(x_1, x_2, \dots, x_k) \longrightarrow t_1, t_2, \dots, t_d$  where  $d \ll k$
- This can be useful for visualization if  $d$  is 2 or 3

The diagram illustrates the transformation of a data matrix. On the left, a matrix of size  $n \times k$  is shown, with rows representing observations and columns representing original features. The matrix is enclosed in large red brackets labeled  $n$  (vertical) and  $k$  (horizontal). The matrix elements are  $x_{11}, x_{12}, \dots, x_{1k}$  in the first row,  $x_{21}, x_{22}, \dots, x_{2k}$  in the second row,  $\vdots$  in the third row, and  $x_{n1}, x_{n2}, \dots, x_{nk}$  in the last row. A horizontal arrow points to the right, where a matrix of size  $n \times d$  is shown. This matrix is also enclosed in large red brackets labeled  $n$  (vertical) and  $d$  (horizontal). The matrix elements are  $t_{11}, t_{12}$  in the first row,  $t_{21}, t_{22}$  in the second row,  $\vdots$  in the third row, and  $t_{n1}, t_{n2}$  in the last row.

$$\begin{matrix} & \overbrace{\hspace{1.5cm}}^k \\ \underbrace{\hspace{1cm}}_n \left[ \begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{array} \right] & \longrightarrow & \underbrace{\hspace{1cm}}_n \left[ \begin{array}{cc} \overbrace{\hspace{1cm}}^d \\ t_{11} & t_{12} \\ t_{21} & t_{22} \\ \vdots & \vdots \\ t_{n1} & t_{n2} \end{array} \right] \end{matrix}$$

# Principal Component Analysis

**Principal Component Analysis** is a dimensionality method that tries to capture most of the variability in the original data

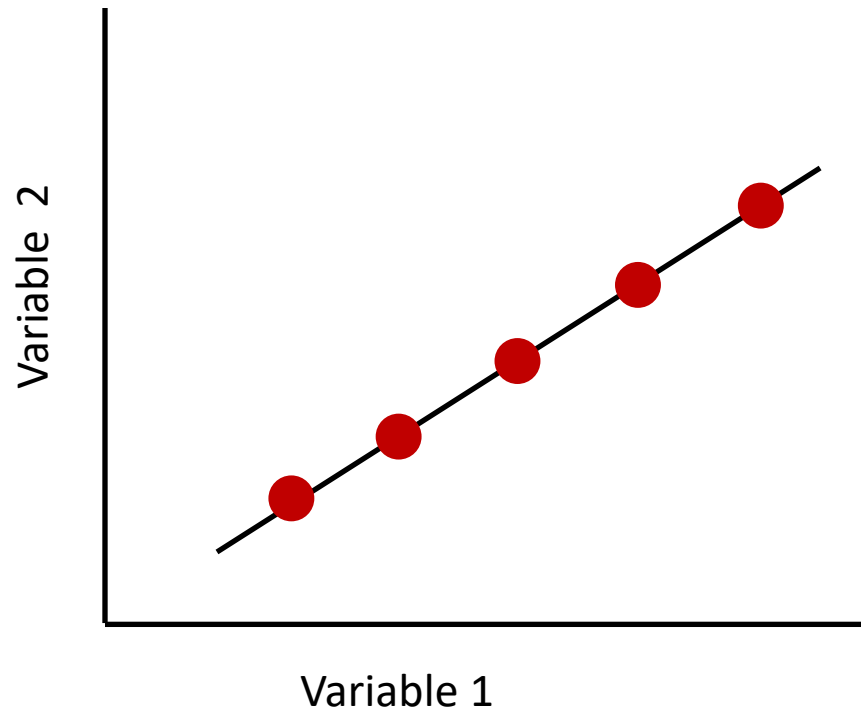
The diagram illustrates the transformation of a data matrix in Principal Component Analysis. On the left, a matrix of size  $n \times k$  is shown, with rows representing observations and columns representing original features. The matrix is enclosed in large red curly braces labeled  $n$  (for rows) and  $k$  (for columns). The matrix elements are  $x_{11}, x_{12}, \dots, x_{1k}$  in the first row,  $x_{21}, x_{22}, \dots, x_{2k}$  in the second row,  $\vdots$  in the third row, and  $x_{n1}, x_{n2}, \dots, x_{nk}$  in the last row. A horizontal arrow points to the right, indicating the transformation. On the right, the transformed matrix of size  $n \times d$  is shown, with rows representing observations and columns representing principal components. It is also enclosed in large red curly braces labeled  $n$  (for rows) and  $d$  (for columns). The matrix elements are  $t_{11}, t_{12}$  in the first row,  $t_{21}, t_{22}$  in the second row,  $\vdots$  in the third row, and  $t_{n1}, t_{n2}$  in the last row.

$$\begin{matrix} & \overbrace{\hspace{1.5cm}}^k \\ \underbrace{\hspace{1cm}}_n \left[ \begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{array} \right] & \longrightarrow & \underbrace{\hspace{1cm}}_n \left[ \begin{array}{cc} \overbrace{\hspace{1cm}}^d \\ t_{11} & t_{12} \\ t_{21} & t_{22} \\ \vdots & \vdots \\ t_{n1} & t_{n2} \end{array} \right] \end{matrix}$$

# Principal Component Analysis

Suppose that two features are highly correlated.

We can summarize their joint values  $(x_1, x_2)$  using a single features  $t_1$



$$t_1 = \frac{1}{2} x_1 + \frac{1}{2} x_2$$

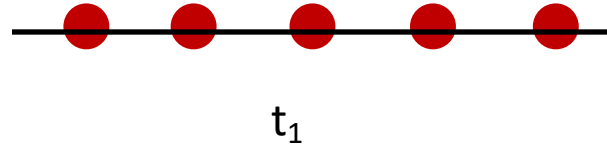


# Principal Component Analysis

Suppose that two features are highly correlated.

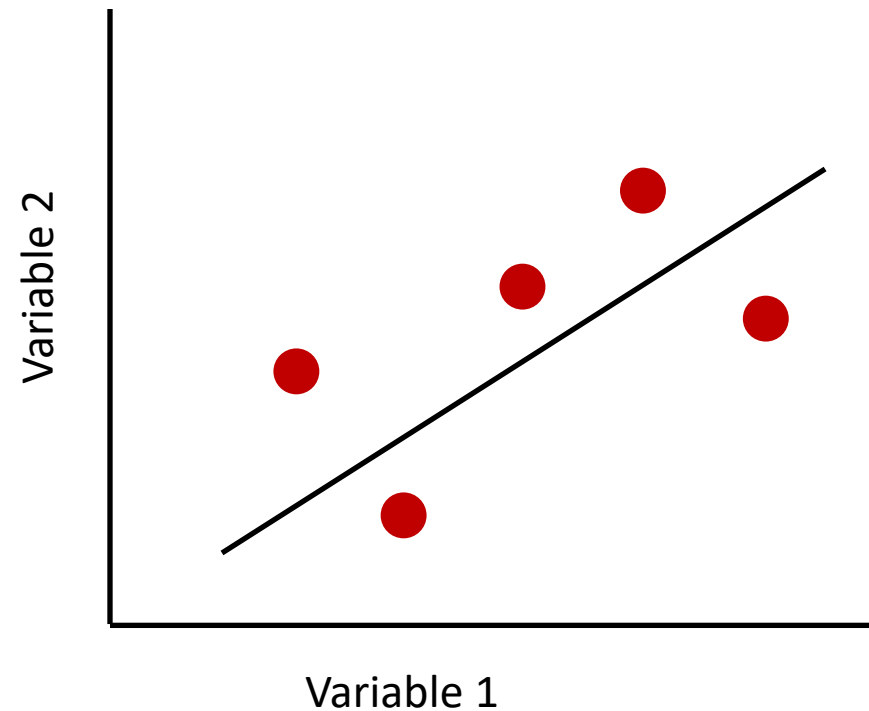
We can summarize their joint values  $(x_1, x_2)$  using a single features  $t_1$

$$t_1 = \frac{1}{2} x_1 + \frac{1}{2} x_2$$



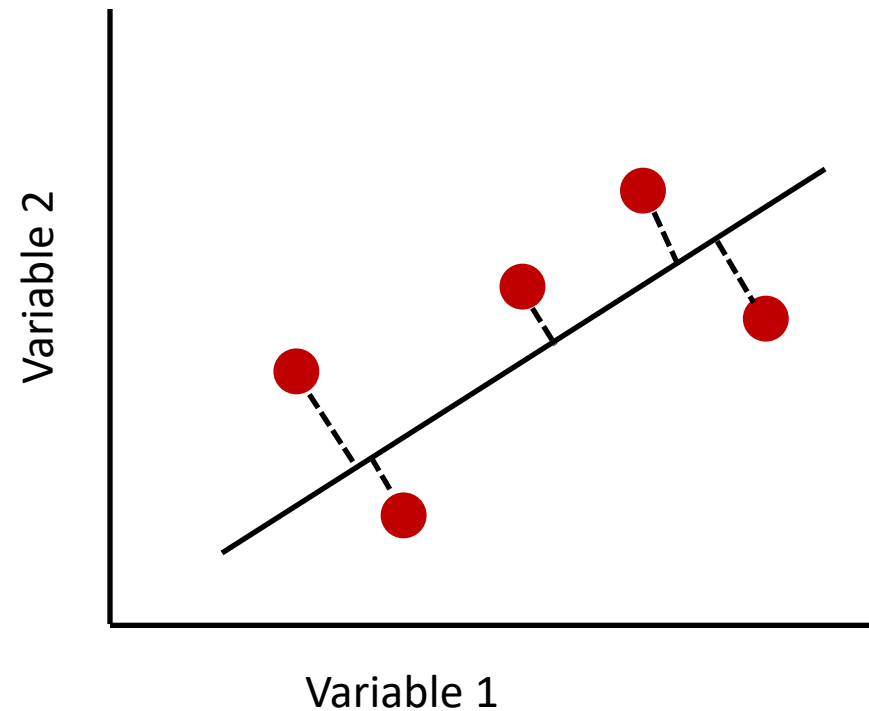
# Principal Component Analysis

We can also do this even if the correlation is not perfect



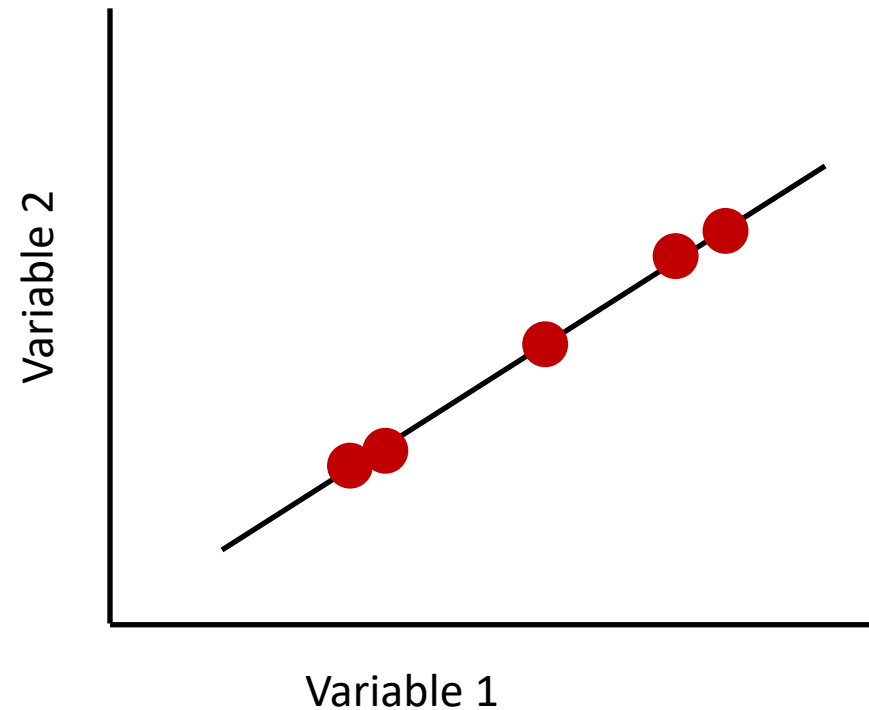
# Principal Component Analysis

We can also do this even if the correlation is not perfect



# Principal Component Analysis

We can also do this even if the correlation is not perfect



# Principal Component Analysis

Principal component **scores**  $t_i$ 's are linear combinations of the original variables  $x_{ij}$ 's:

$$t_{i1} = \alpha_{11}x_{i1} + \alpha_{21}x_{i2} + \dots + \alpha_{k1}x_{ik}$$

$\alpha_{j1}$  are the **loadings** for the first principal component

- The "norm" of the loadings is 1

$$\sum_{j=1}^k \alpha_{j1}^2 = 1$$

$$t_{11} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$

We can do this for each case in our data set we get values:  $t_{11}, \dots, t_{n1}$

# Principal Component Analysis

To run PCA, we start by centering each variable  $x_i$  so that it has a mean of 0

We also usually divide all variables by their standard deviation

- i.e., z-score transform the features before performing PCA
- We divide by the  $s_i$ 's so that variables with large variances don't dominate

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$
$$\begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_k \end{bmatrix}$$
$$\begin{bmatrix} s_1 & s_2 & \cdots & s_k \end{bmatrix}$$

# Principal Component Analysis

The loadings for the first principal component are found by finding the projection vector  $A_1 = (\alpha_{11}, \alpha_{21}, \alpha_{k1})$  such that the variance of the  $t_i$  is maximized

Find the  $\alpha$ 's that maximize:

$$\frac{1}{n-1} \sum_{i=1}^n t_i^2 =$$

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$

Subject to the constraint:

$$\sum_{j=1}^k \alpha_{j1}^2 = 1$$

# Principal Component Analysis

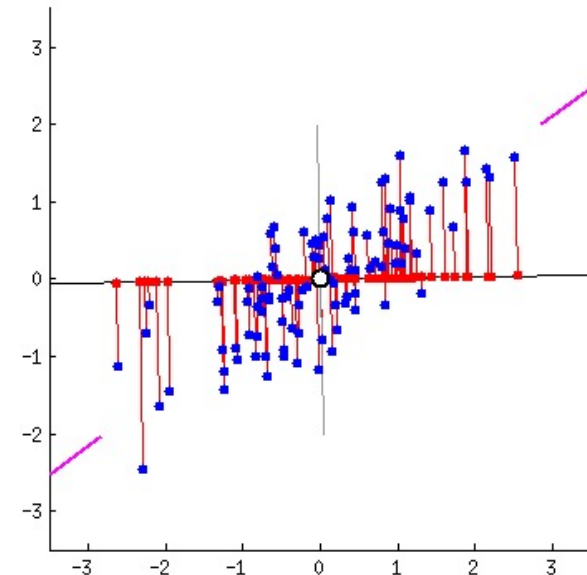
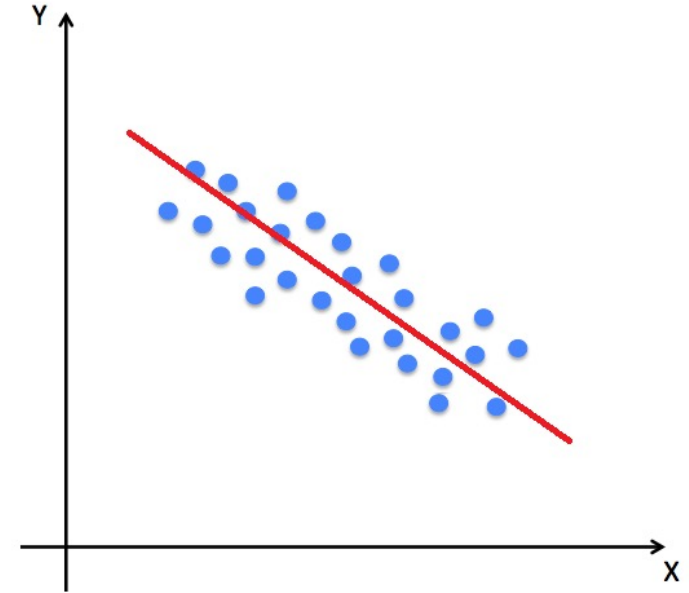
The loadings for the first principal component are found by finding the projection vector  $A_1 = (\alpha_{11}, \alpha_{21}, \alpha_{k1})$  such that the variance of the  $t_i$  is maximized

Find the  $\alpha$ 's that maximize:

$$\frac{1}{n-1} \sum_{i=1}^n t_i^2 = \frac{1}{n-1} \sum_{i=1}^n (\alpha_{11}z_{i1} + \alpha_{21}z_{i2} + \dots + \alpha_{k1}z_{ik})^2$$

Subject to the constraint:

$$\sum_{j=1}^k \alpha_{j1}^2 = 1$$





# The Second Principal Component

The second principal component scores  $t_{i2}$  is the linear combination of the  $z_1, z_2, \dots, z_k$  that has maximal variance and is **uncorrelated** with the first principal component scores  $t_{i1}$

- $t_{i2} = \alpha_{12}z_1 + \alpha_{22}z_2 + \dots + \alpha_{k2}z_k$
- $\text{cor}(T_1, T_2) = 0$

This is equivalent of having  $A_1$  be orthogonal to  $A_2$

- $A_1^T A_2 = 0$ 
$$\sum_{j=1}^k \alpha_{j1} \cdot \alpha_{j2} = 0$$

First principal component

$$\begin{bmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{n1} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{k1} \end{bmatrix}$$

Second principal component

$$\begin{bmatrix} t_{12} \\ t_{22} \\ \vdots \\ t_{n2} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{12} \\ \alpha_{22} \\ \vdots \\ \alpha_{k2} \end{bmatrix}$$

# The Second Principal Component

The second principal component scores  $t_{i2}$  is the linear combination of the  $z_1, z_2, \dots, z_k$  that has maximal variance and is **uncorrelated** with the first principal component scores  $t_{i1}$

- $t_{i2} = \alpha_{12}z_1 + \alpha_{22}z_2 + \dots + \alpha_{k2}z_k$
- $\text{cor}(T_1, T_2) = 0$

This is equivalent of having  $A_1$  be orthogonal to  $A_2$

- $A_1^T A_2 = 0$ 
$$\sum_{j=1}^k \alpha_{j1} \cdot \alpha_{j2} = 0$$

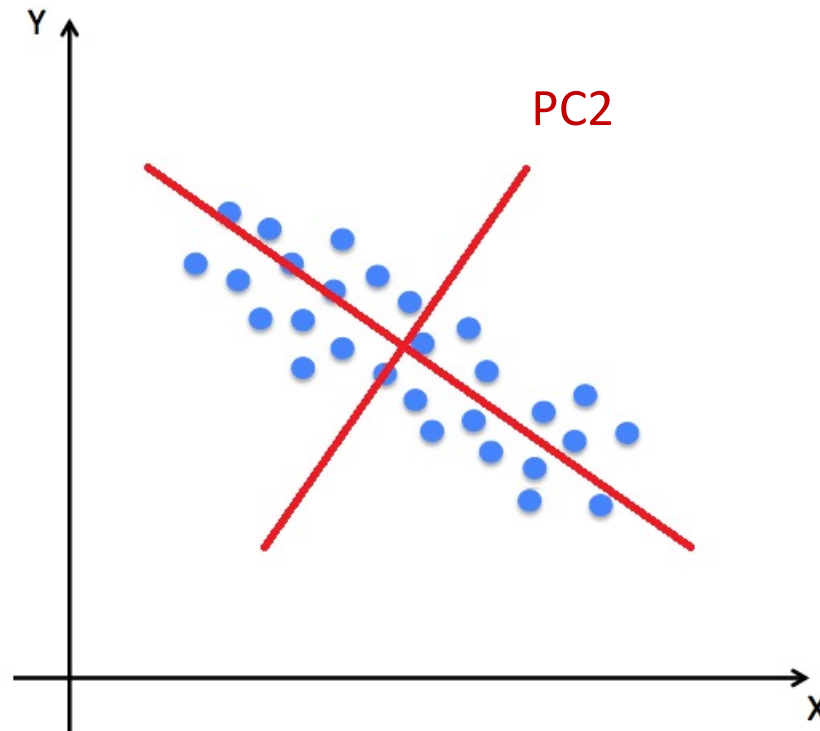
First and second principal components

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \\ \vdots & \vdots \\ t_{n1} & t_{n2} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \\ \vdots & \vdots \\ \alpha_{k1} & \alpha_{k2} \end{bmatrix}$$

# Geometric interpretation of the second PC

Find the direction that maximizes the variance of  $t_i$ 's

- Data projected on to the principal component is most spread out that is perpendicular (orthogonal) to the other PCs



# Higher Principal Components

We continue this process until we find all the principal component scores,  $T_1, T_2, \dots, T_d$

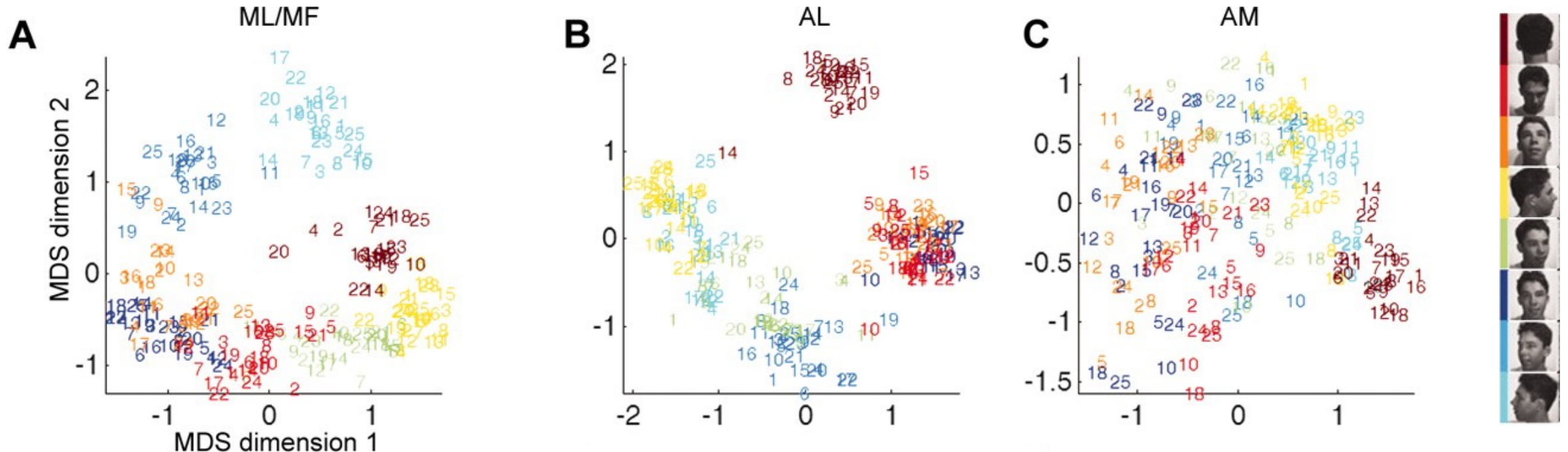
- The principal component scores are unique up to a sign flip  $T_i = -T_i$ 
  - To find the principal components what is really done is an eigenvalue decomposition of the covariance matrix.

All principal components

$$\begin{bmatrix} t_{11} & t_{12} & \dots & t_{1d} \\ t_{21} & t_{22} & \dots & t_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nd} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1d} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{k1} & \alpha_{k2} & \dots & \alpha_{kd} \end{bmatrix}$$

# Neuroscience example

Freiwald and Tsao (Science 2010) used dimensionality reduction to reduce the activity of a large population of neurons to two dimensions so that they could visualize how different brain regions represent faces.



# Proportion of Variance Explained

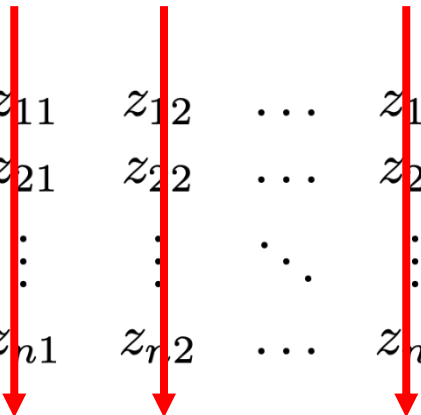
In order to know how many principal components to use, it is usual to assess the **proportion of variance explained** (PVE) by each PC

Total variance: 
$$\sum_{j=1}^k Var(z_j) = \sum_{j=1}^k \frac{1}{n-1} \sum_{i=1}^n (z_{ij})^2$$


Variance explained by  $m^{\text{th}}$  principal component:

$$Var(t_m) = \frac{1}{n-1} \sum_{i=1}^n t_{im}^2 = \frac{1}{n-1} \sum_{i=1}^n \left( \sum_{j=1}^k \alpha_{jm} z_{ij} \right)^2$$

$$\begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix}$$


  
 $Var(z_1)$

$$\begin{bmatrix} t_{11} & t_{12} & \dots & t_{1d} \\ t_{21} & t_{22} & \dots & t_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nd} \end{bmatrix}$$

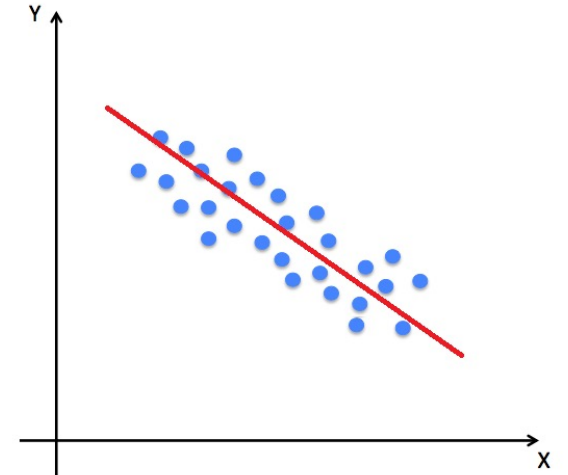

  
 $Var(t_m)$

# Proportion of Variance Explained

Proportion *of variance that is explained* by each PC:

$$PVE_{mv} = \frac{\text{Variance explained by } m^{\text{th}} \text{ principal component}}{\text{Total variance}}$$

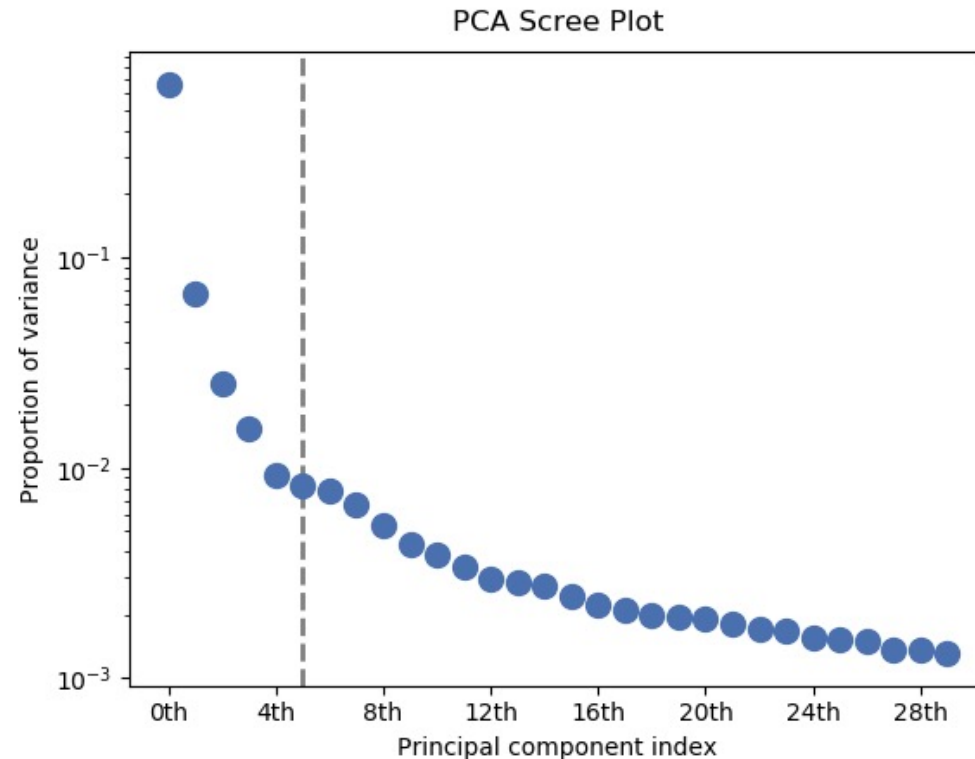
$$PVE_m = \frac{Var(t_m)}{Var(total)} = \frac{\sum_{i=1}^n (\sum_{j=1}^k \alpha_{jm} z_{ij})^2}{\sum_{j=1}^k \sum_{i=1}^n z_{ij}^2}$$



# Deciding how many PCs to use

A **scree plot** shows the PVE as a function of PC number

- The number of PCs chosen is often selected by looking for the “elbow” in this plot
  - i.e., point where PVE stop dramatically dropping and levels off





# PCA example: personality traits of fictional characters

The [Open-Source Psychometrics Project](#) conducted a survey where they got ratings of 235+ personality traits from 800 fictional characters.

Let's use PCA to assess:

- How to personality traits commonly covary
- Which fictional characters are most similar

If you want to find out which fictional character you are most similar you can take their [“Which Character” personality quiz](#)

Rate characters from Good Will Hunting:



Where does Will Hunting fall on this spectrum?

oppressed  privileged

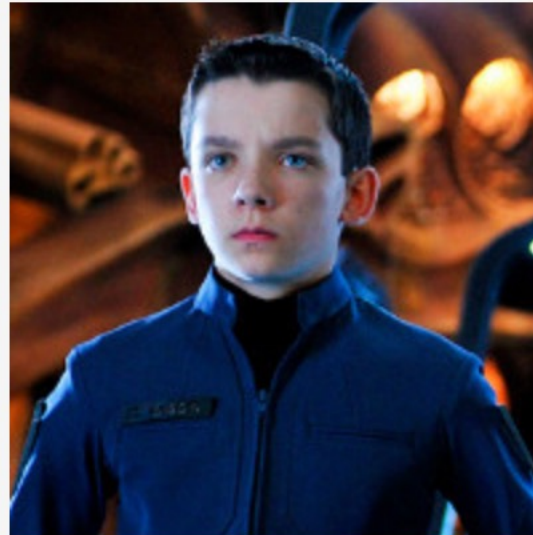
Answer

(don't know, skip)

19/25

# Let's try the PCA in R...

The best match between the self assessment you provided and the profile of a fictional character as rated by other people who have taken this survey is the character Ender Wiggin (Ender's Game).



**84% match**

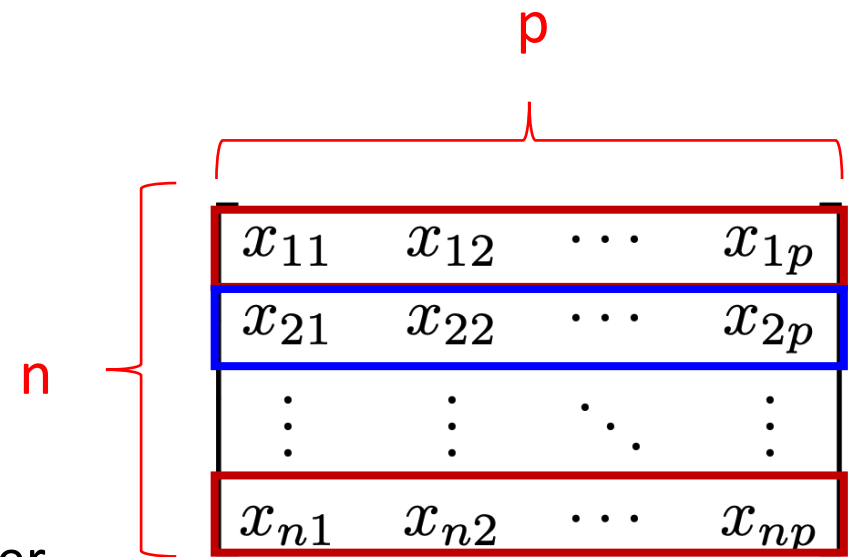
Your traits versus their traits are graphed below (click on points for labels).



# Clustering

Clustering divides  $n$  data points  $x_i$ 's into subgroups

- Data points in the same group are similar/homogeneous
- Data points in different groups are different from each other



Examples:

- Examining gene expression levels to group cancer types together
- Examining consumer purchasing behavior to perform market segmentation

Clustering can be:

- **Flat:** no structure beyond dividing points into groups
- **Hierarchical:** Population is divided into smaller and smaller groups (tree like structure)

# K-means clustering

K-means clustering partitions the data into ***K*** distinct, non-overlapping clusters

- i.e., each data point  $x_i$  belongs to exactly one cluster  $C_k$

The number of clusters, ***K***, needs to be specified prior to running the algorithm

The goal is to minimize the within-cluster variation for some measure  $W(C_k)$

$$\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K W(C_k) \right\}$$

# K-means clustering

A common within cluster similarity measure  $W(C_k)$  is the sum of the **Euclidean distance** between all pairs of points in a cluster:

$$\text{minimize}_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$

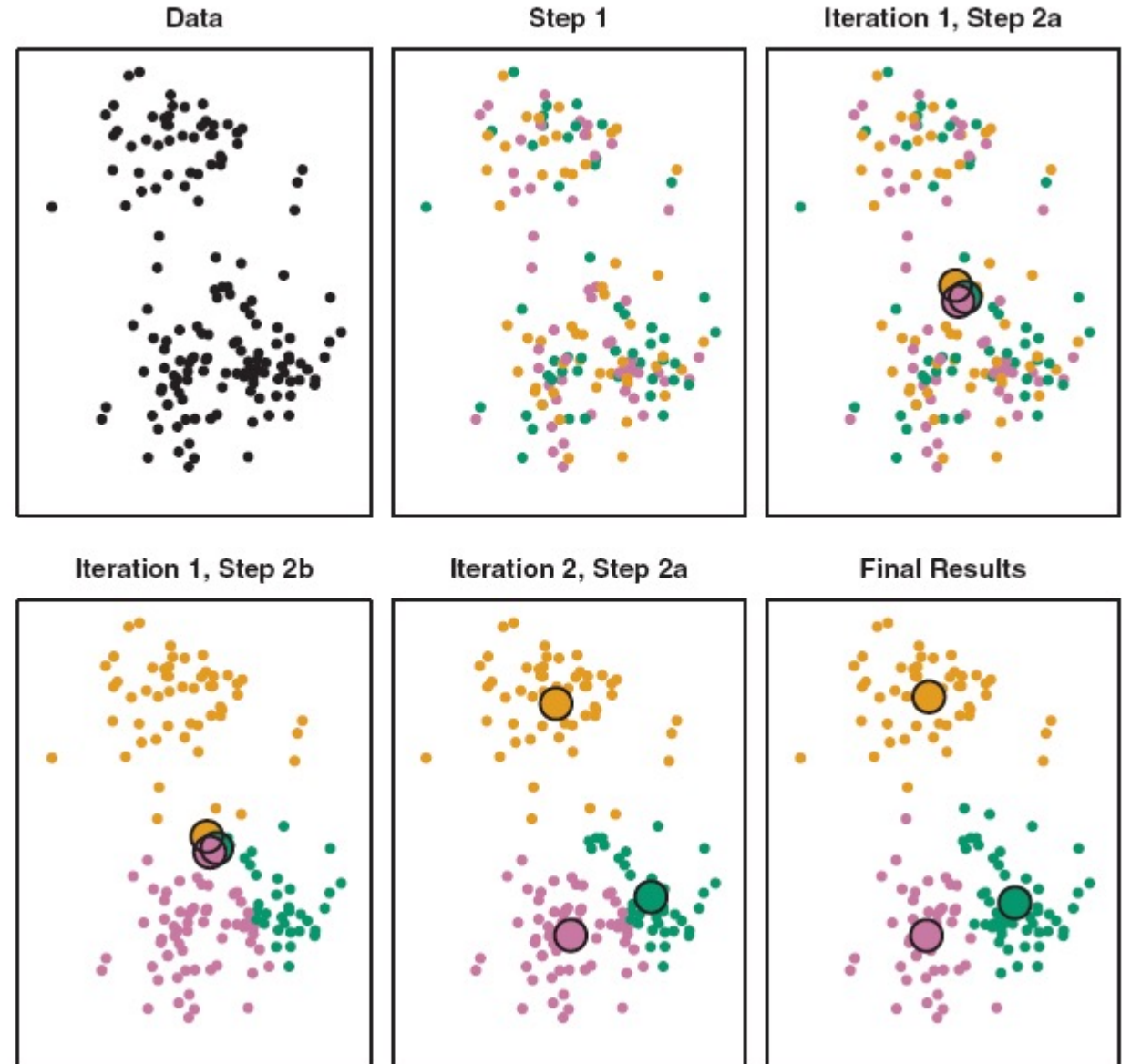
This is equivalent to minimizing:  $\frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2$

Finding the exact optimal solution is computationally intractable (there are  $k^n$  possible partitions), but a simple algorithm exists to find a local optimum which often works well in practice.

$x_{11}$	$x_{12}$	$\cdots$	$x_{1p}$
$x_{21}$	$x_{22}$	$\cdots$	$x_{2p}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x_{n1}$	$x_{n2}$	$\cdots$	$x_{np}$

# K-means clustering

1. Randomly assign points to clusters  $C_k$
2. Calculate cluster centers as means of points in each cluster
3. Assign points to the closest cluster center
4. Recalculate cluster center as the mean of points in each cluster
5. Repeat steps 3 and 4 until convergence



# K-means clustering

Because only a local minimum is found, different random initializations will lead to different solutions

- One should run the algorithm multiple times to get better solutions



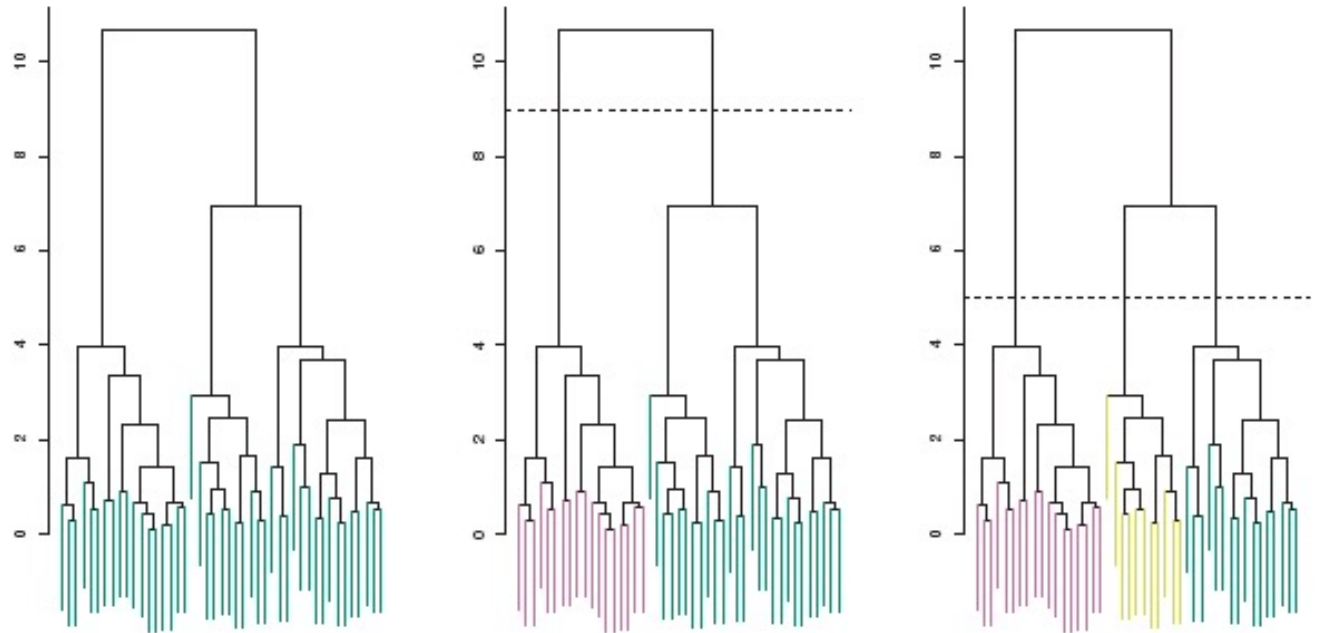


# Hierarchical clustering

In **hierarchical clustering** we create a dendrogram which is a tree-based representation of successively larger clusters.

We can cut the dendrogram at any point to create as many clusters as desired

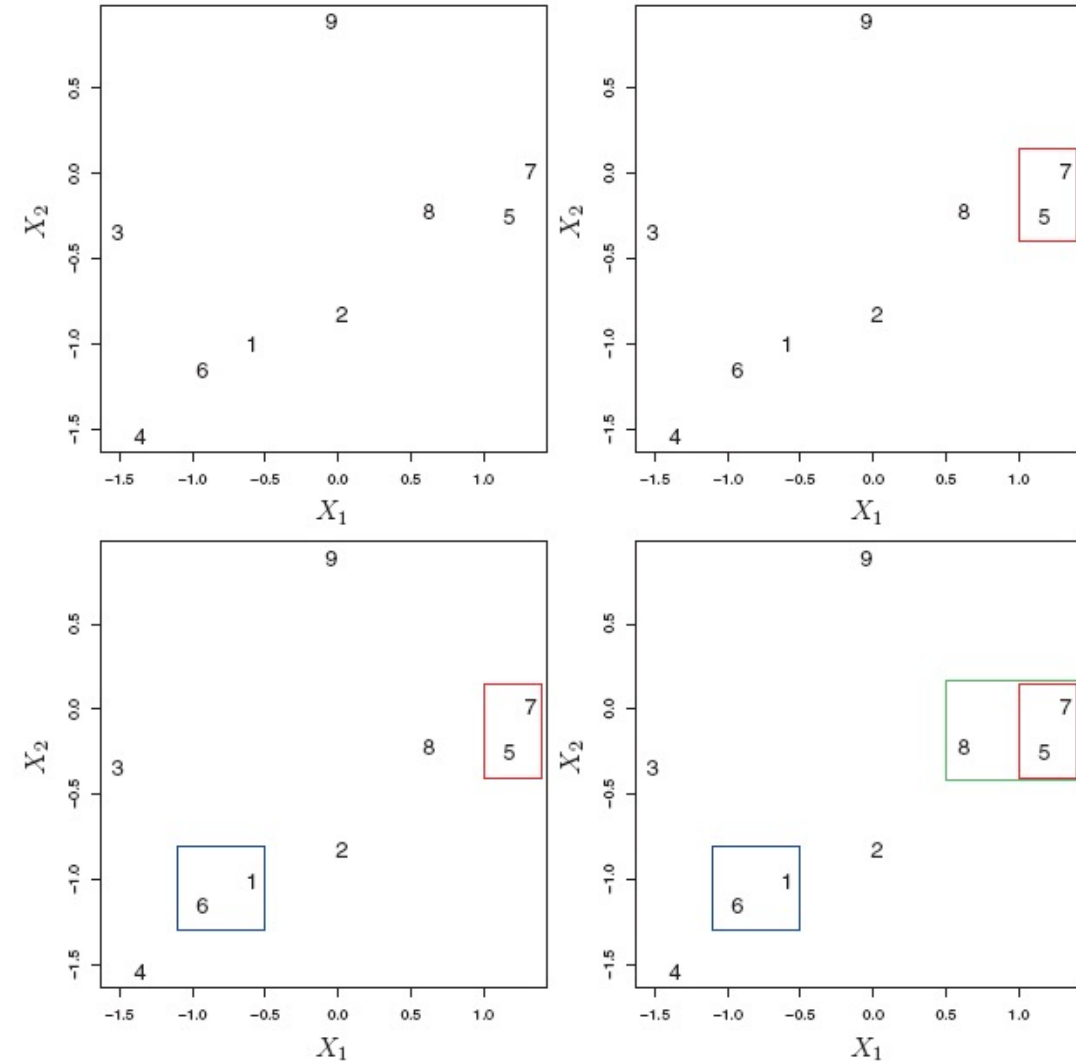
- i.e., don't need to specify the number of clusters,  $K$ , beforehand



# Hierarchical clustering

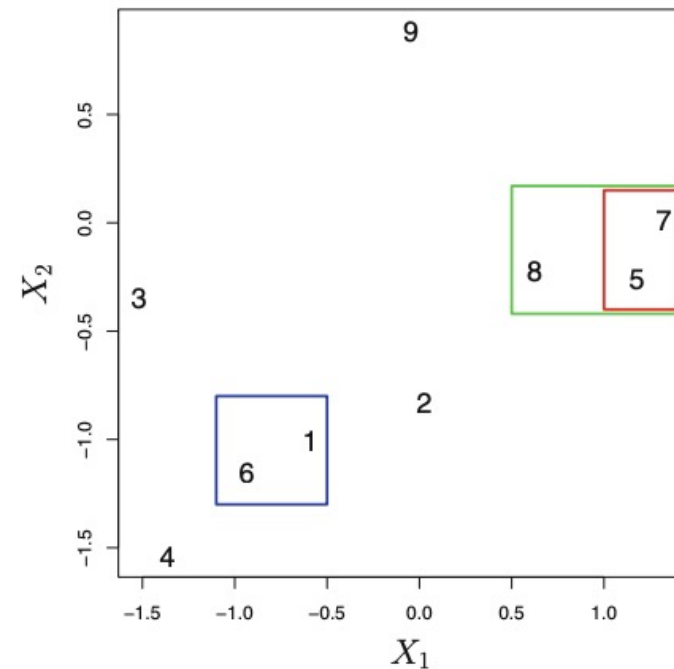
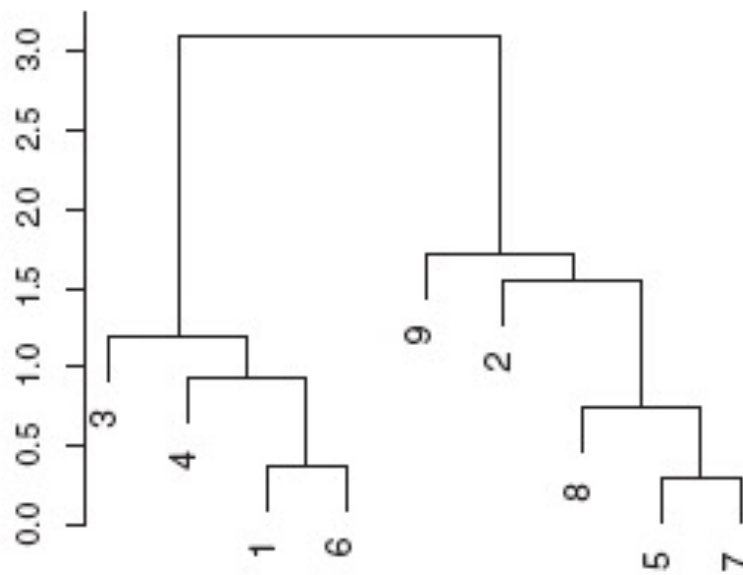
We can create a hierarchical clustering of the data using simple bottom-up agglomerative algorithm:

1. Choosing a (dis)similarity measure
  - E.g., The Euclidean distance
2. Initializing the clustering by treating each point as its own cluster
3. Successively merging the pair of clusters that are most similar
  - i.e., calculate the similarity between all pairs of clusters and merging the pair that is most similar
4. Stopping when all points have been merged into a single cluster



# Hierarchical clustering

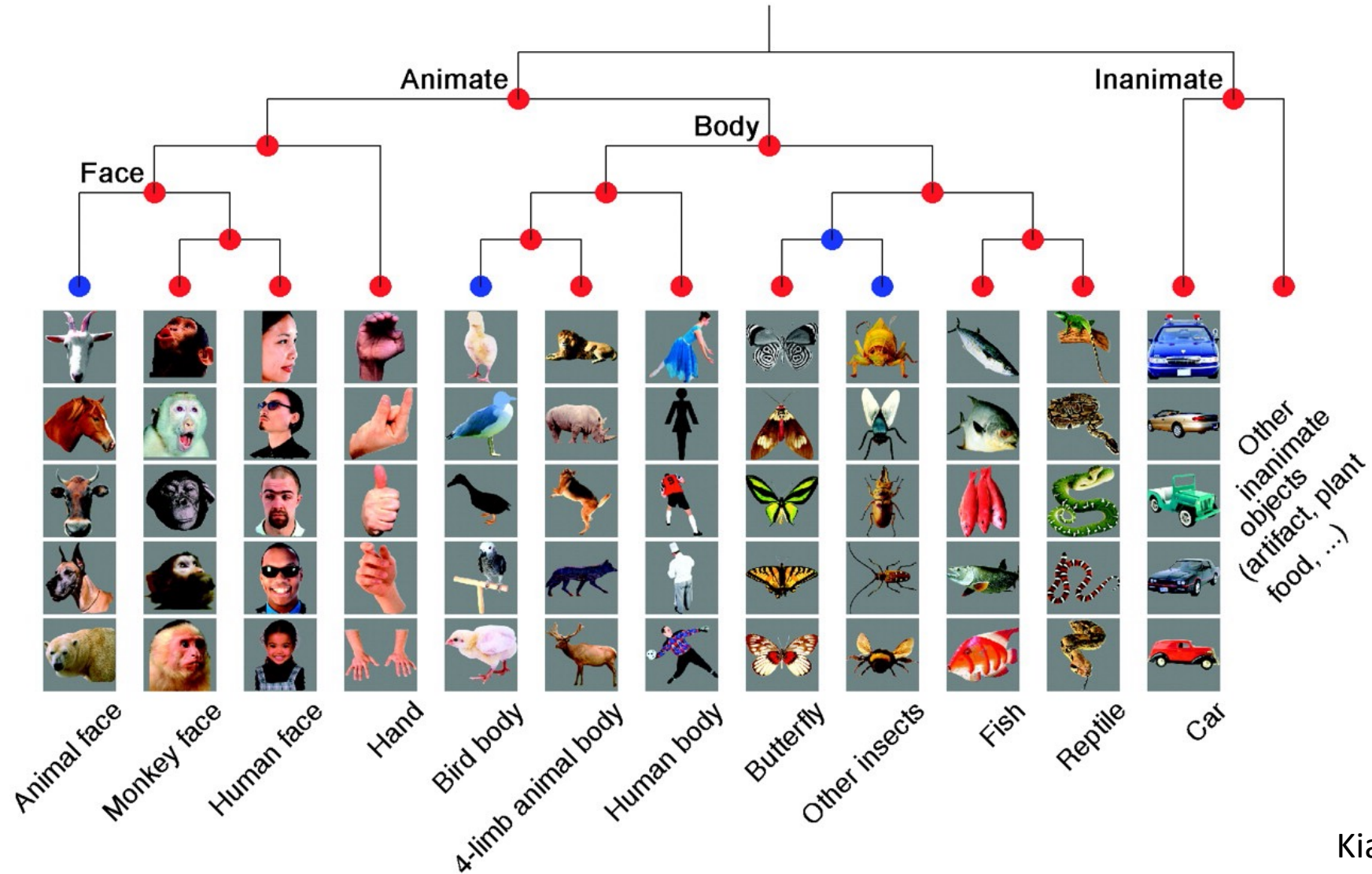
The vertical height that two clusters/points merge show how similar the two *clusters* are



Note: horizontal distance between *individual points* is not important:

- point 9 is considered as similar to point 2 as it is to point 7

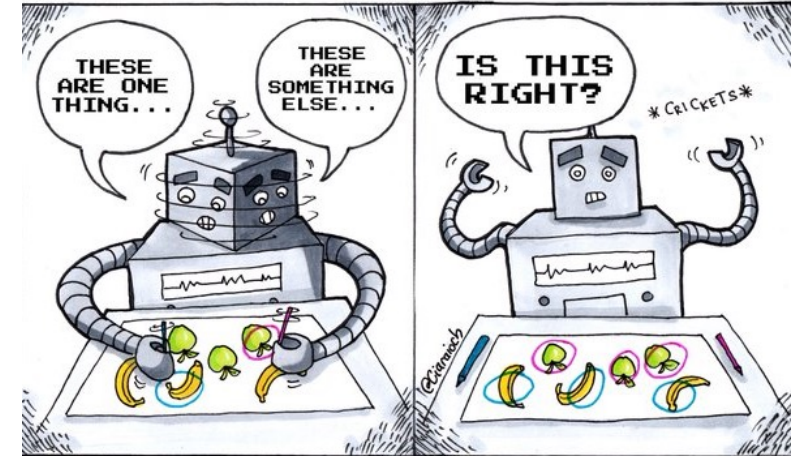
# Hierarchical clustering example



# Issues with clustering

Choices made can effect the results:

- Feature normalization and/or dissimilarity measure
- K-means: choice of K
- For hierarchical cluster: linkage and cut height



**Unsupervised Learning**

Potential approaches to deal with these issues:

- Try a few methods and see if one gives interesting/useful results
- Validate that you get similar results on a second set of data

Let's try clustering in R...