

# STAT406- Methods of Statistical Learning Lecture 19

Matias Salibian-Barrera

UBC - Sep / Dec 2019

# Unsupervised learning

- Unsupervised  $\neq$  Supervised
- High-“density” regions (w/o model)
- Agglomerative / hierarchical methods
- High-“density” regions (with a model)
  - EM-algorithm
- Dimension reduction (PCA, MDS, etc.)

# Clustering - Problem

- Data:  $p$  features / variables per “unit”

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

- $\mathbf{X}_1, \dots, \mathbf{X}_n$

# Clustering

- Goal: find **regions** where  $\mathbf{X}_i$ 's are “clustered”
- Goal: find **regions** where  $P(\mathbf{X})$  is relatively high
- These **regions** are sometimes modeled

# Clustering

- Lower dimensional subspaces (linear manifolds)

Principal Components

- Convex regions with high  $P(\mathbf{X})$

K-means / K-medoids – Hierarchical methods

# Clustering

- Intrinsically different from **classification**
- There is no clear performance measure to evaluate “success”
- Hence the name: “unsupervised learning”

# Clustering

## Example 1 – 9 Breweries - 26 attributes

```
> a
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [...]
V1  3.51  3.41  3.20  2.73  2.35  3.03  2.21  3.91  3.07 [...]
V2  4.43  4.05  3.66  5.25  3.88  4.23  3.27  2.71  4.08 [...]
V3  4.76  3.42  4.22  2.44  4.18  2.47  3.67  4.59  4.74 [...]
V4  3.68  3.78  3.07  2.75  2.78  3.12  2.49  3.91  3.34 [...]
V5  4.77  1.04  3.86  5.28  3.86  4.24  3.40  4.23  4.23 [...]
[...]
      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [...]
V1   3.07   3.45   2.53   3.12   2.93   2.24   2.41   3.32 [...]
V2   3.82   4.29   4.71   3.58   3.27   3.11   3.14   3.74 [...]
V3   4.17   4.44   4.53   4.10   4.13   4.12   3.43   4.32 [...]
V4   3.21   3.74   2.83   3.14   2.80   2.39   2.40   3.32 [...]
V5   3.94   4.47   4.83   3.82   3.46   3.39   3.22   4.01 [...]
[...]
```

# Clustering

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_9 \in \mathbb{R}^{26}$$

Do they appear grouped / clustered?



# UN Votes

- From <http://hdl.handle.net/1902.1/12379>
- UN, founded 1946, 193 members
- “important” votes (U.S. State Department)
- Votes: Yes (1), Abstain (2), No (3), Absent (8), Not a Member (9)
- 368 important votes, 77 countries voted  $\geq 95\%$  of these

# UN Votes

- Do voting patterns reflect political alignments?
- Do countries vote along known political blocks?
- Data:  $\mathbf{X}_i$ : votes for country  $i$

$$\mathbf{X}_i \in \mathbb{R}^{368}, \quad i = 1, \dots, 77 \text{ (countries)}$$

What groups are there?

# Cancer example

- From [HTF09], details in script
- Gene expression for 64 samples
- There are 6830 genes
- $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{64} \in \mathbb{R}^{6830}$
- We know tissue type for ea. sample
- Really: “**feature selection**”

# K-means / K-medoids

- Look for convex sets of relative high density
- The number of sets **K** is specified *a priori* (but we'll come back to this)
- Since “high density” is related to “closeness”

$$\min \sum_{r=1}^{\mathbf{K}} \sum_{i,j \in \mathcal{C}_r} d^2(\mathbf{x}_i, \mathbf{x}_j)$$

minimize over all partitions  $\mathcal{C}_1, \dots, \mathcal{C}_K$

# K-means / K-medoids

Note that

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^n d^2(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{r=1}^{\mathbf{K}} \sum_{i \in \mathcal{C}_r} \sum_{j=1}^n d^2(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{r=1}^{\mathbf{K}} \sum_{i \in \mathcal{C}_r} \left[ \sum_{j \in \mathcal{C}_r} d^2(\mathbf{x}_i, \mathbf{x}_j) + \sum_{j \notin \mathcal{C}_r} d^2(\mathbf{x}_i, \mathbf{x}_j) \right] \\ &= \sum_{r=1}^{\mathbf{K}} \sum_{i, j \in \mathcal{C}_r} d^2(\mathbf{x}_i, \mathbf{x}_j) + \sum_{r=1}^{\mathbf{K}} \sum_{i \in \mathcal{C}_r} \sum_{j \notin \mathcal{C}_r} d^2(\mathbf{x}_i, \mathbf{x}_j) \\ &\quad T = \quad W \quad + \quad B \end{aligned}$$

# K-means / K-medoids

When  $d^2(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2$

$$W = \sum_{r=1}^K \sum_{i,j \in \mathcal{C}_r} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_{r=1}^K \sum_{i \in \mathcal{C}_r} \|\mathbf{x}_i - \bar{\mathbf{x}}_r\|^2$$

- Given  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ , assign  $\mathbf{x}_i$  to the cluster  $\mathcal{C}_j$  with closest mean

$$\mathbf{x}_i \leftarrow \arg \min_{1 \leq j \leq K} \|\mathbf{x}_i - \bar{\mathbf{x}}_j\|^2$$

# K-means / K-medoids

- Note that

$$\bar{\mathbf{X}}_r = \hat{\boldsymbol{\mu}}_r = \arg \min_{\boldsymbol{\mu}} \sum_{i \in \mathcal{C}_r} \|\mathbf{x}_i - \boldsymbol{\mu}\|^2$$

- Given  $\hat{\boldsymbol{\mu}}_1, \dots, \hat{\boldsymbol{\mu}}_K$

$$\min_{\mathcal{C}_1, \dots, \mathcal{C}_K} \sum_{r=1}^k \sum_{i \in \mathcal{C}_r} \|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_r\|^2$$

is attained with

$$\mathbf{x}_i \leftarrow \arg \min_{1 \leq j \leq K} \|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j\|^2$$

# K-means / K-medoids

- And, given  $C_1, \dots, C_K$

$$\min_{\hat{\mu}_1, \dots, \hat{\mu}_K} \sum_{r=1}^k \sum_{i \in C_r} \|\mathbf{x}_i - \hat{\mu}_r\|^2$$

is attained with

$$\hat{\mu}_r \leftarrow \bar{\mathbf{x}}_r = \frac{1}{n_r} \sum_{i \in C_r} \mathbf{x}_i$$

**This suggests a simple iterative (and greedy) algorithm.**



# K-means / K-medoids

## Remarks

- Algorithm is greedy
- Answer depends on the initial configuration
- It needs to be started from **many initial configurations**

# Cancer data

```
> set.seed(31)
> nci.km <- kmeans(nci, centers=8)
> table(nci.km$cluster)
```

1	2	3	4	5	6	7	8
8	6	6	14	3	8	4	15

```
> set.seed(311)
> nci.km <- kmeans(nci, centers=8)
> table(nci.km$cluster)
```

1	2	3	4	5	6	7	8
4	12	6	9	4	8	19	2

# K-means / K-medoids

Need **more** starting points...

```
> set.seed(31)
> nci.km <- kmeans(nci, centers=8, iter.max = 5000,
  nstart=1000)
> table(nci.km$cluster)
```

1	2	3	4	5	6	7	8
3	8	5	14	6	15	9	4

```
> set.seed(311)
> nci.km <- kmeans(nci, centers=8, iter.max = 5000,
  nstart=1000)
> table(nci.km$cluster)
```

1	2	3	4	5	6	7	8
4	5	8	9	14	3	15	6

**These clusters are the same**

# K-means / K-medoids

Tissue	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	$C_8$
LEUKEMIA	1		5					
BREAST		2			1		2	2
RENAL		1			8			
COLON				1		6		
PROSTATE				2				
MELANOMA					1		7	
OVARIAN				5	1			
NSCLC				6	3			
OTHER	2	5			1			2

# UN Votes example

- Not all countries voted each time

```
> dim(X)
[1] 368 77
> sum( complete.cases(X) )
[1] 145
```

- Only use resolutions with full votes

```
X2 <- X[complete.cases(X), ]
```

- Use `kmeans` in R

# UN Votes example

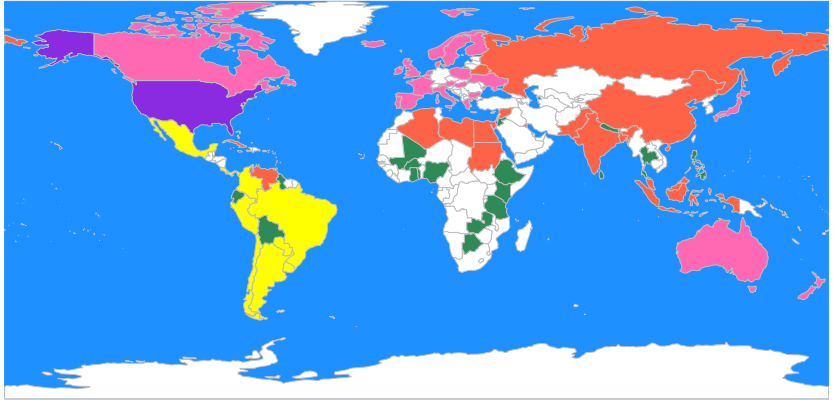
```
> set.seed(123)
> b <- kmeans(t(X2), centers=5,
               iter.max=20, nstart=1)
> table(b$cluster)
```

```
 1  2  3  4  5
18  2  7 19 31
```

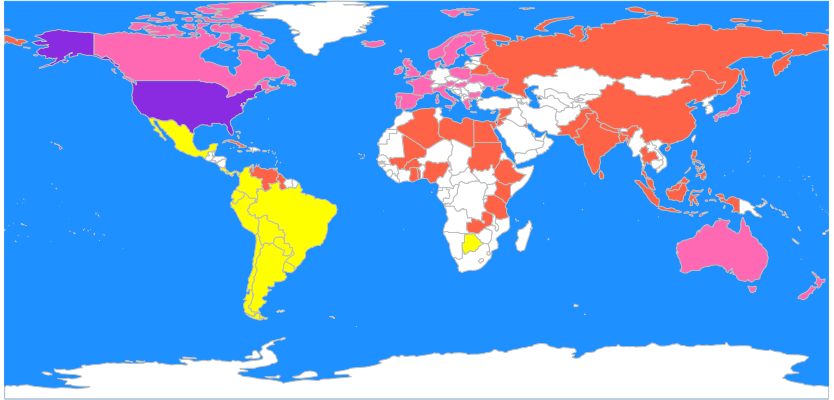
```
> b <- kmeans(t(X2), centers=5,
               iter.max=20, nstart=1)
> table(b$cluster)
```

```
 1  2  3  4  5
27 12 13  7 18
```

# UN Votes example - $K=5$

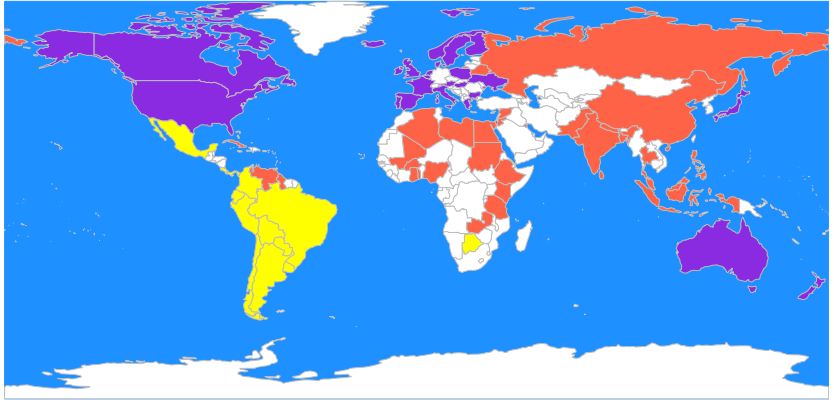


# UN Votes example - $K=4$





# UN Votes example - $K=3$



# K-means++

- A cleverly chosen set of initial centres
- K-means++

- Pick a centre  $\mathbf{c}_1$  at random (from data)

- Then for  $j$  in  $2:k$

- Compute weights

$$w_i = \min \left( d^2(\mathbf{x}_i, \mathbf{c}_1), \dots, d^2(\mathbf{x}_i, \mathbf{c}_{j-1}) \right) ,$$

- Pick next centre  $\mathbf{c}_j$  from data with prob  $\propto w_i$

- Implemented in `flexclust::kcca`

# Choosing $K$

For each cluster  $\mathcal{C}_r$ , let

$$W(\mathcal{C}_r) = \sum_{j, j \in \mathcal{C}_r} d^2(\mathbf{x}_i, \mathbf{x}_j) \quad r = 1, \dots, \mathbf{K}$$

and

$$W_{\mathbf{K}} = \sum_{j=1}^{\mathbf{K}} W(\mathcal{C}_r)$$

# Choosing $K$

- Note that selecting  $K$  to minimize  $W_K$  does not generally work
- $W_K$  typically decreases with  $K$
- A simple example follows

# Selecting the number $K$ of clusters

For each cluster  $C_r$ , let

$$W(C_r) = \sum_{j, i \in C_r} d(\mathbf{x}_i, \mathbf{x}_j) \quad r = 1, \dots, \mathbf{K}$$

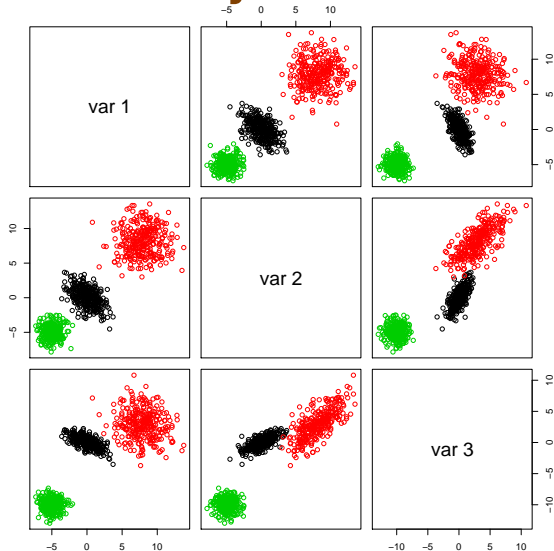
and

$$W_{\mathbf{K}} = \sum_{j=1}^{\mathbf{K}} W(C_r)$$

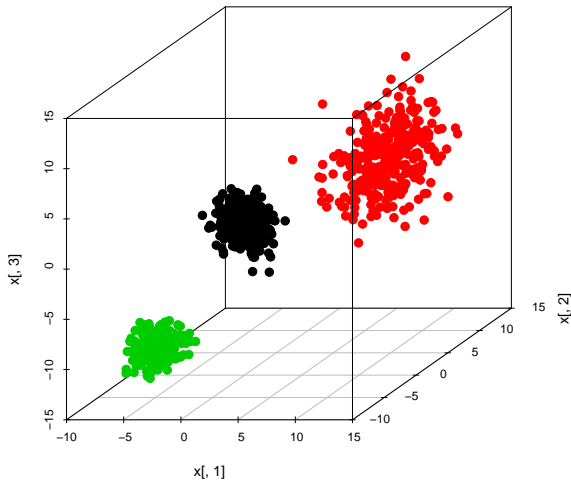
# Selecting the number $K$ of clusters

- Note that selecting  $K$  to minimize  $W_K$  does not generally work
- $W_K$  typically decreases with  $K$
- A simple example follows

# Pairs plot - Easy case

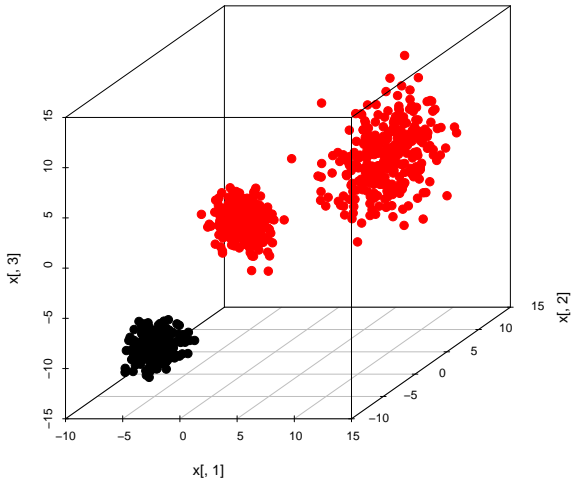


# Pairs plot - Easy case

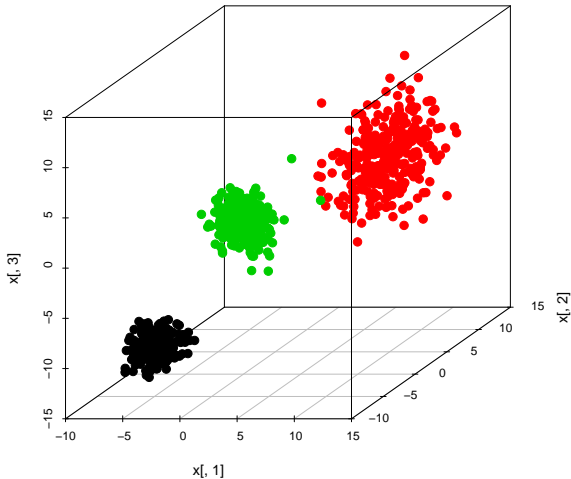




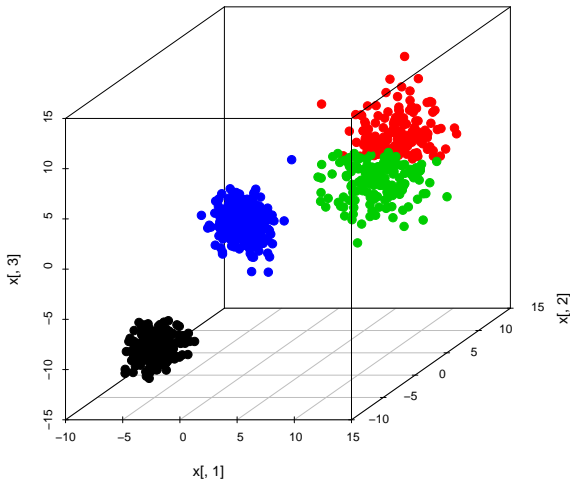
# Pairs plot - K-means - $K = 2$



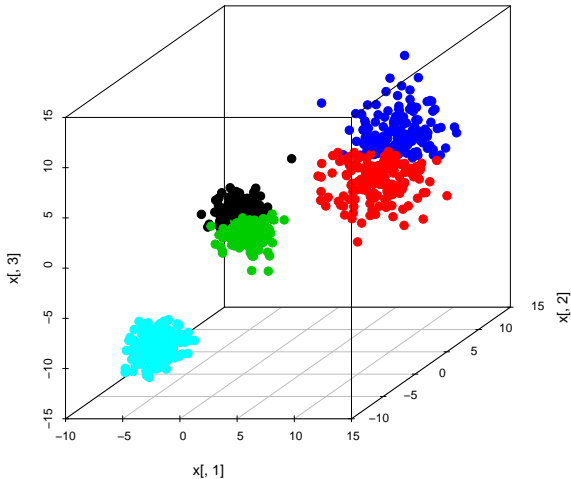
# Pairs plot - K-means - $K = 3$



# Pairs plot - K-means - $K = 4$

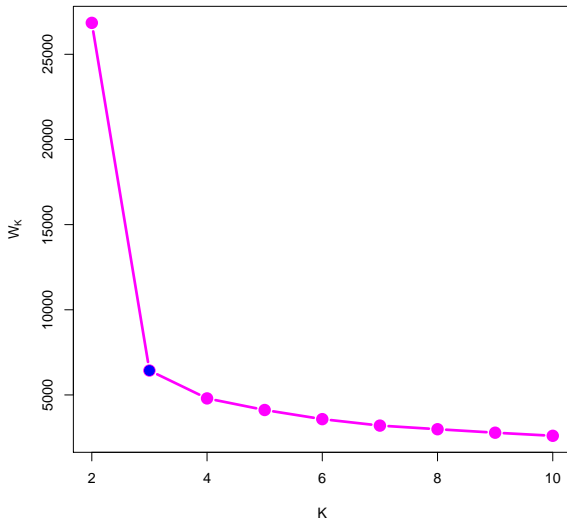


# Pairs plot - K-means - $K = 5$

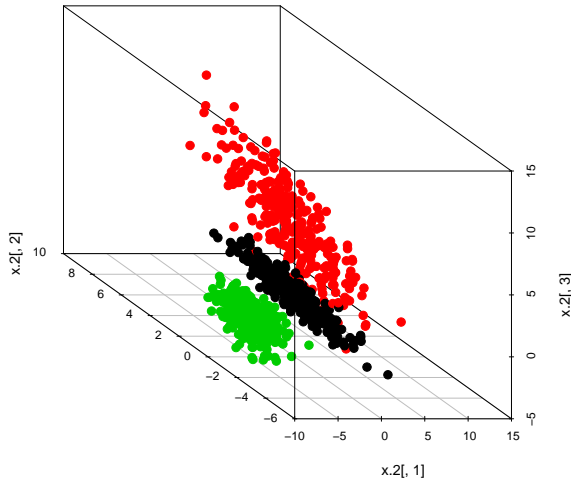


# Pairs plot - K-means - $W_K$

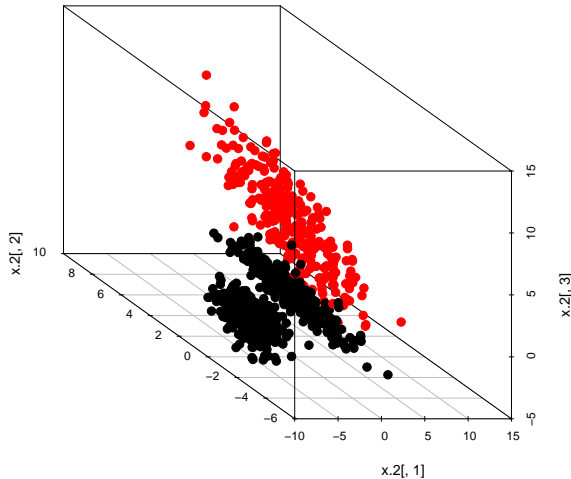
K-means -  $W_K$



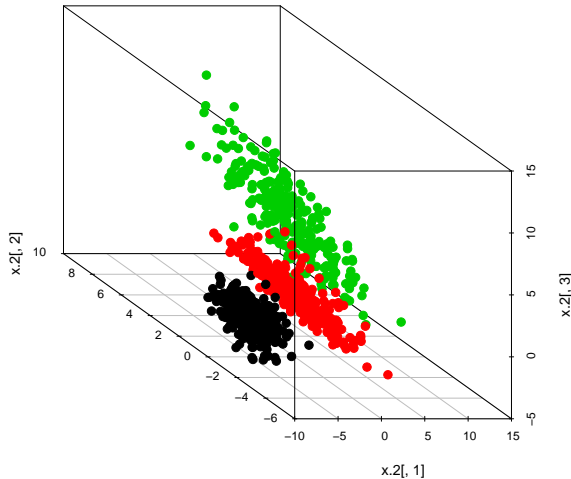
# Pairs plot - K-means



# Pairs plot - K-means - $K = 2$

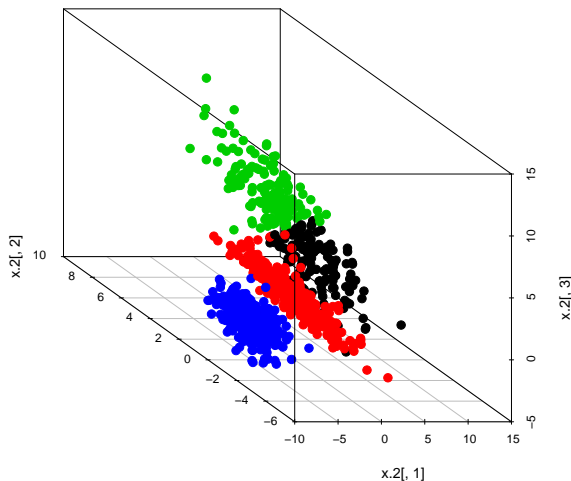


# Pairs plot - K-means - $K = 3$

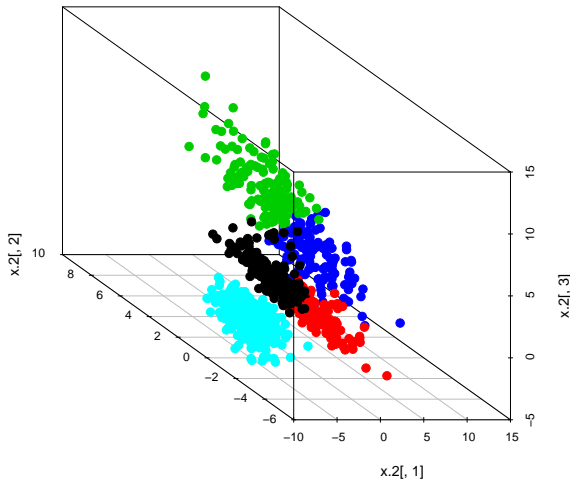




# Pairs plot - K-means - $K = 4$

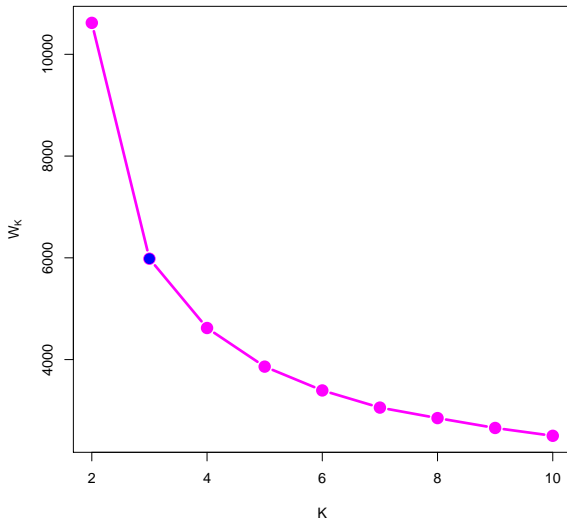


# Pairs plot - K-means - $K = 5$



# Pairs plot - K-means - $W_K$

K-means -  $W_K$



# GAP Statistic

GAP Statistic (Tibshirani, Walther and Hastie, 2001)

Consider

$$G(\mathbf{K}) = E[\log(W_{\mathbf{K}})] - \log(W_{\mathbf{K}})$$

where  $E[\log(W_{\mathbf{K}})]$  is the expected value under a certain reference distribution

# Clest algorithm

## Clest algorithm

Idea - select the value of **K** that produces classes that are best predicted by your favourite classification method.

Dudoit, Fridlyand, 2002, A prediction-based resampling method for estimating the number of clusters in a dataset, Genome Biology **3(7)** : research0036.1 - 0036.21

# Other approaches to select **K**

Dudoit, Fridlyand, 2003, Bagging to improve the accuracy of a clustering procedure, Bioinformatics, **19**, 1090-1099