WEEK 8: MACHINE LEARNING 2

SECU0057

BENNETT KLEINBERG

5 MAR 2020

Applied Data Science

WEEK 8: MACHINE LEARNING 2

TODAY

- unsupervised learning
- core algorithm in detail
- problems of unsupervised learning

UNSUPERVISED ML

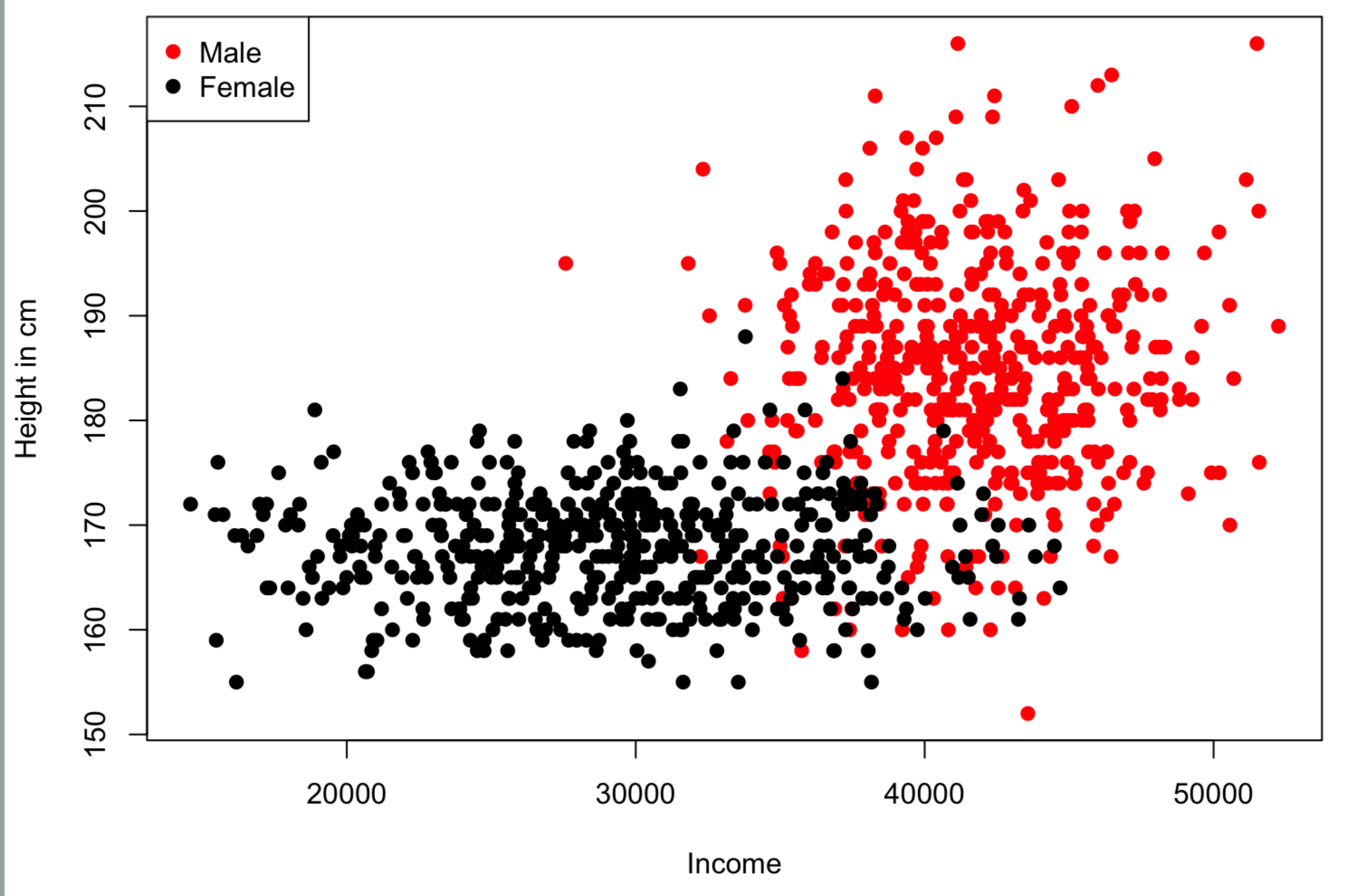
PROBLEM FOR SUPERVISED APPROACHES

- most of the time we don’t have labelled data
- sometimes there are no labels at all
- core idea: finding clusters in the data

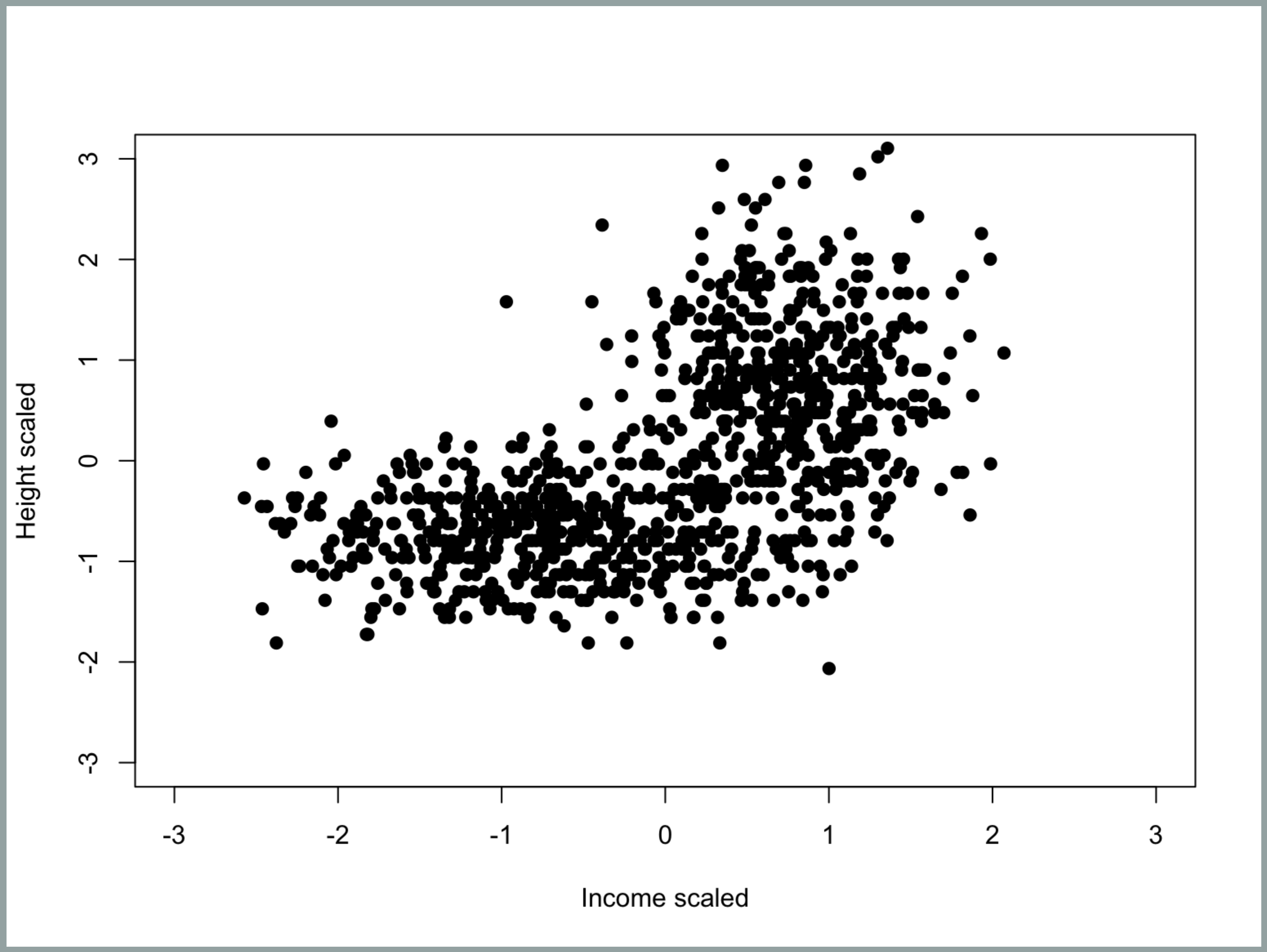
EXAMPLES

- grouping of online ads
- clusters in crime descriptions
- collections of texts without authors

Practically all interesting problems are unlabelled data problems.


Male
Female
Height in cm
150
160
170
180
190
200
210
20000
30000
40000
50000
Income

THE UNSUPERVISED CASE



AIM

- examining whether there are patterns (e.g. groups in the data)
- possibly: a ‘grouped’ underlying data generation process
- helpful because: reduces dimensions of the data

HOW TO TEST WHETHER THERE ARE PATTERNS?

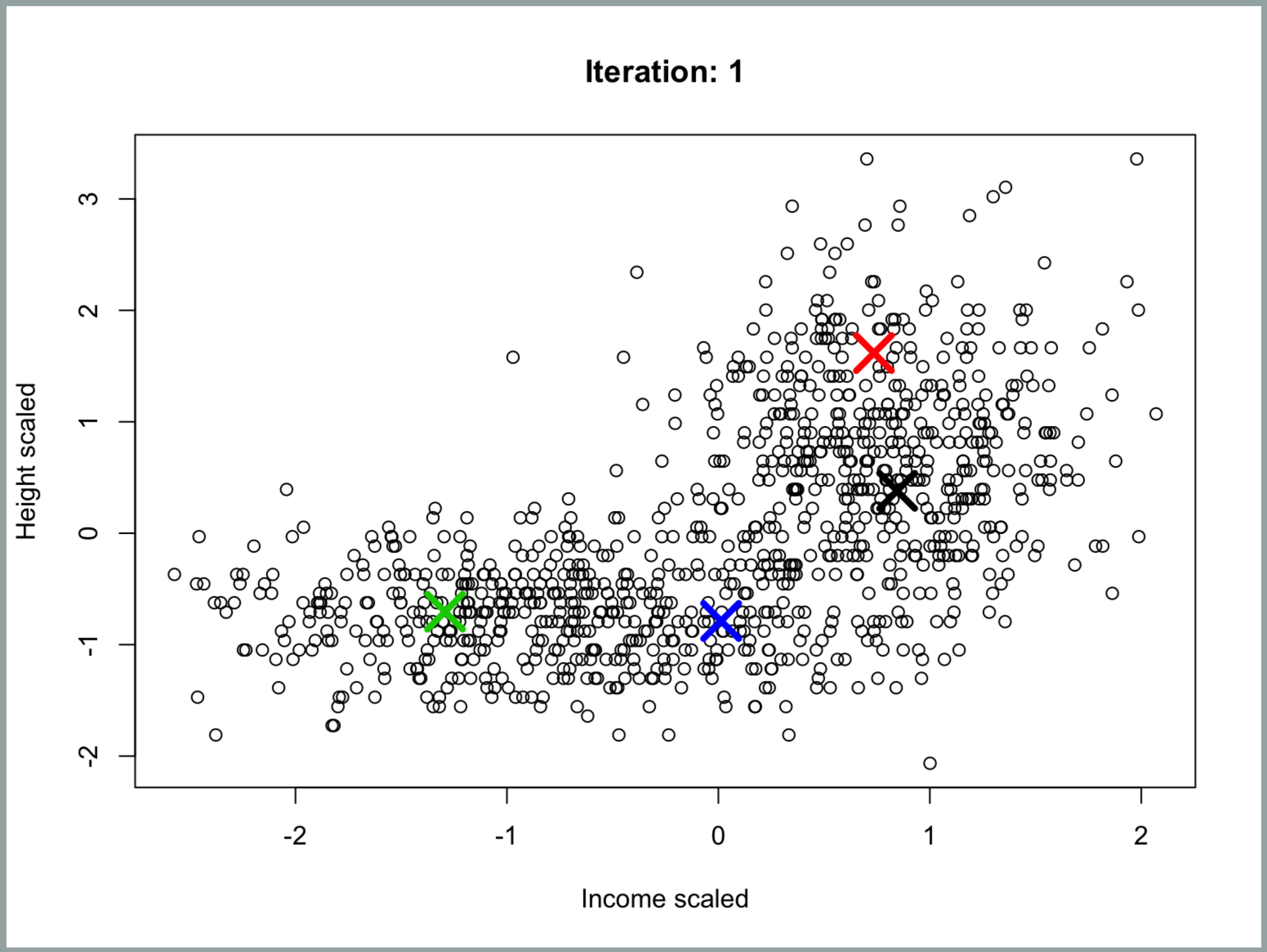
1. separate data into a set number of clusters
2. find the best cluster assignment of observations

Common method: **k-means algorithm**

1. SETTING K

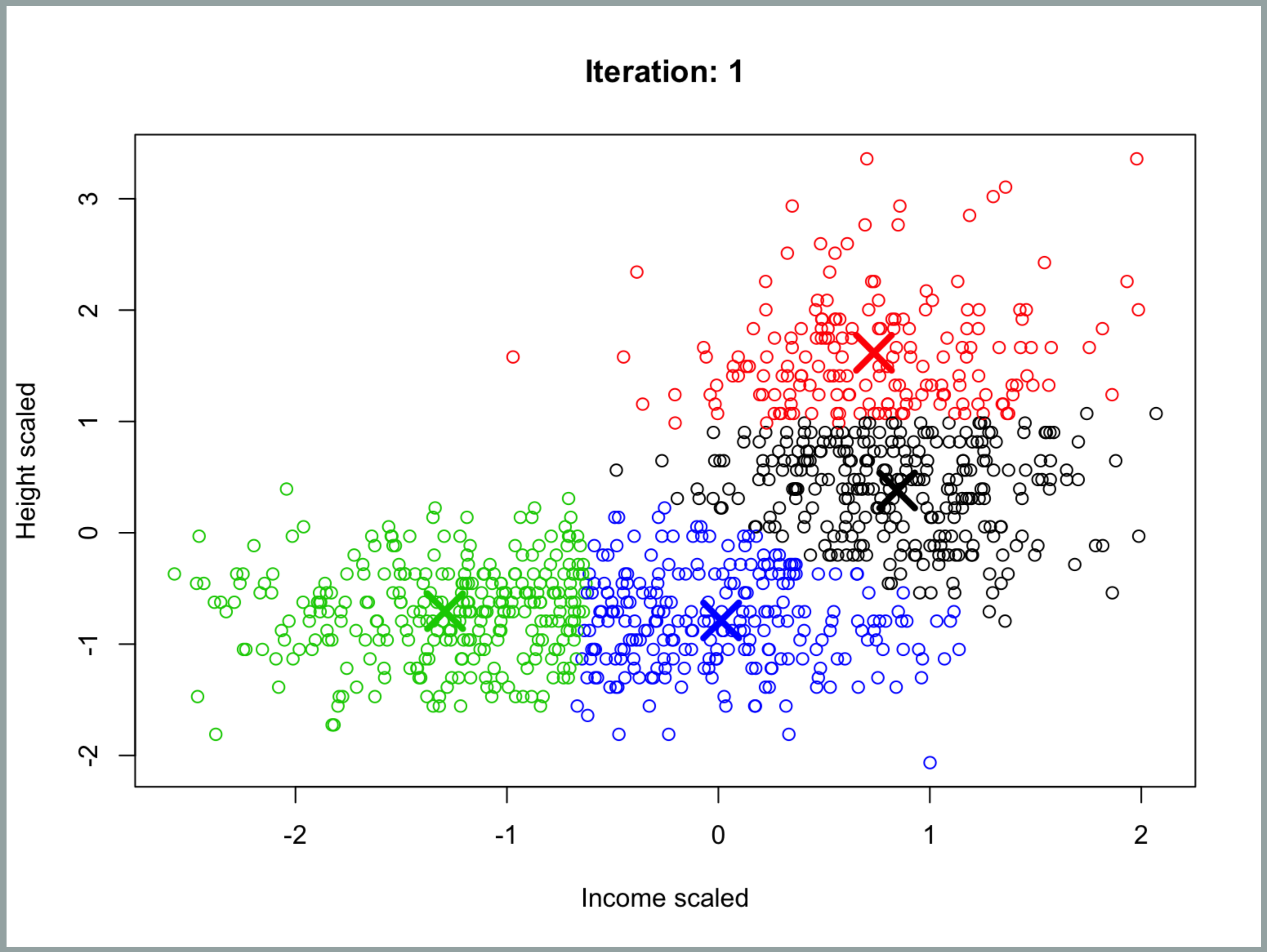
Let's take $k = 4$.

```
unsup_model_1 = kmeans(data4
                       , centers = 4
                       , nstart = 10
                       , iter.max = 10)
```



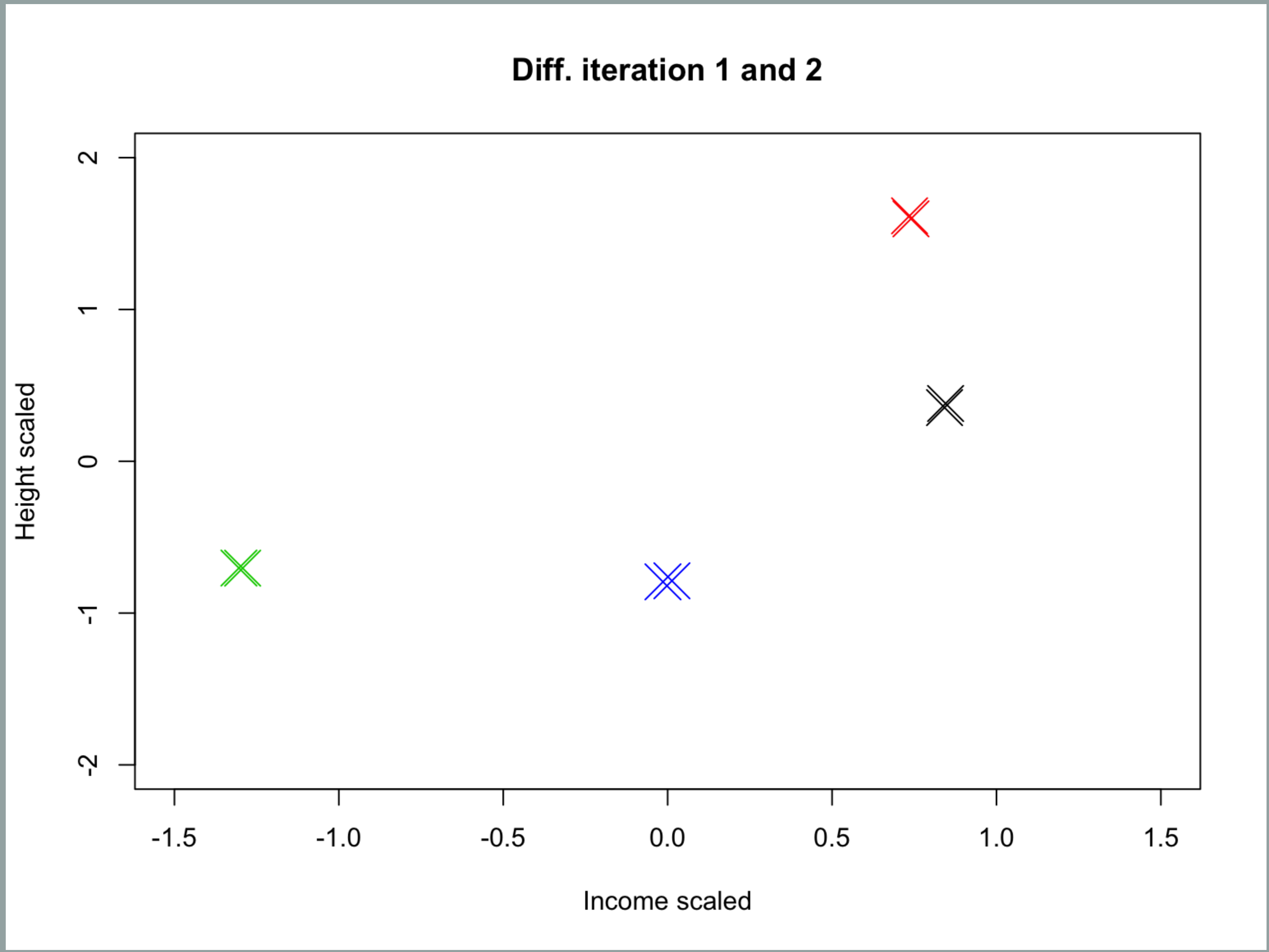
Iteration: 1
Height scaled
Income scaled
-2
-1
0
1
2
3

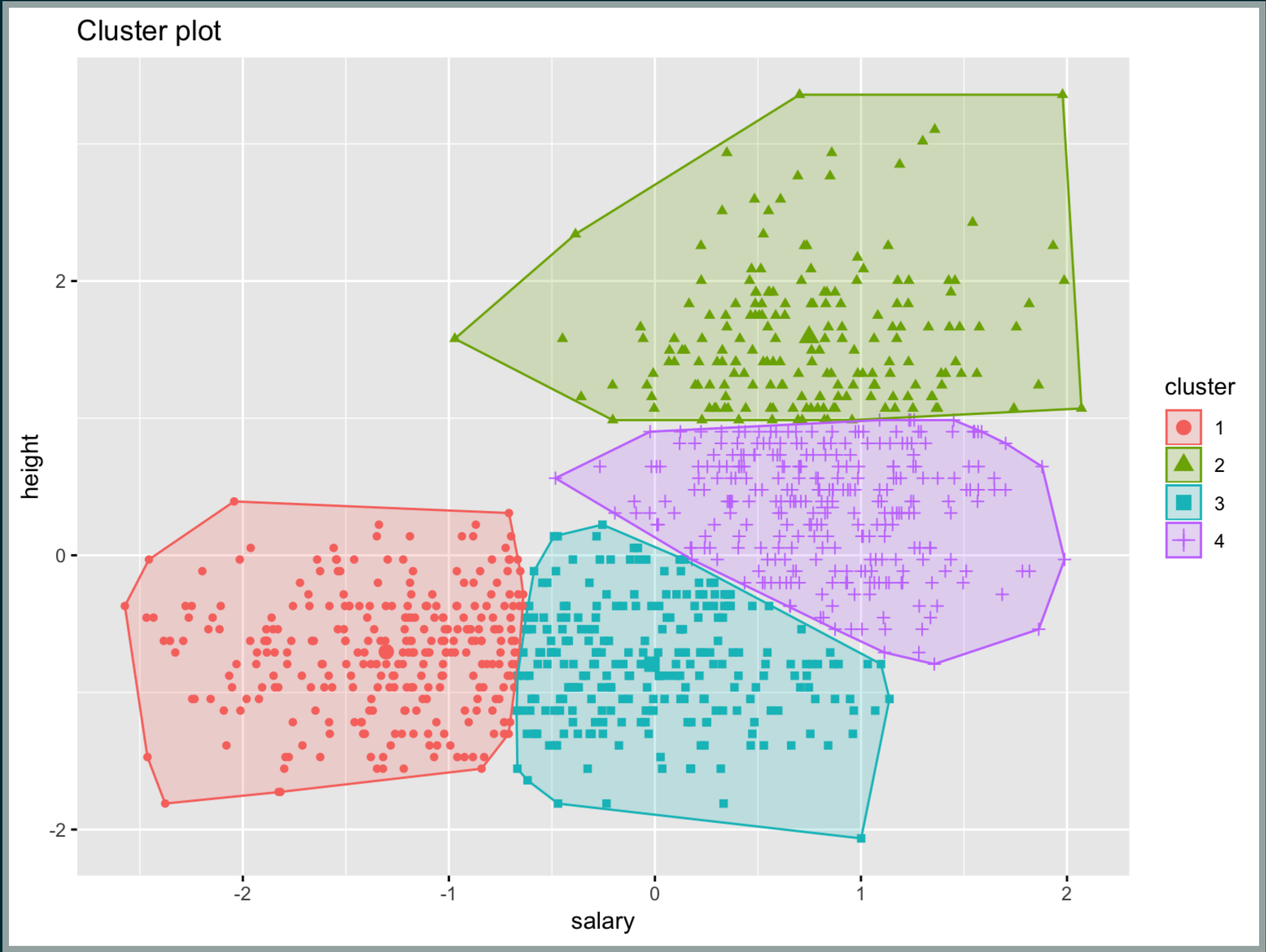
ASSIGNING CLUSTER MEMBERSHIP



ITERATIVE ALGORITHM

WHAT HAPPENED IN THE ITERATIONS?



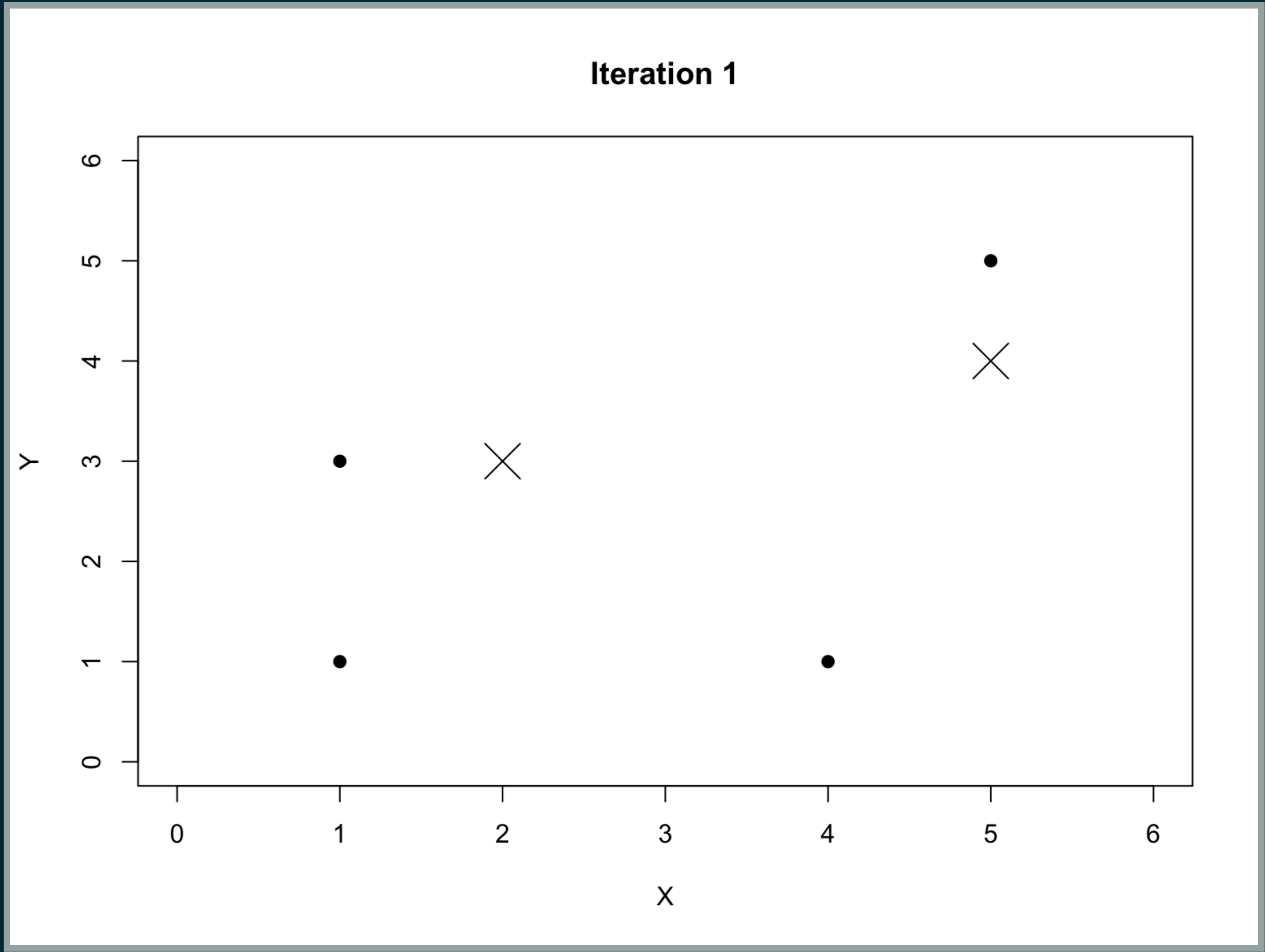


Cluster plot
height
salary
2
0
-2
-2
-1
0
1
2
cluster
1
2
3
4

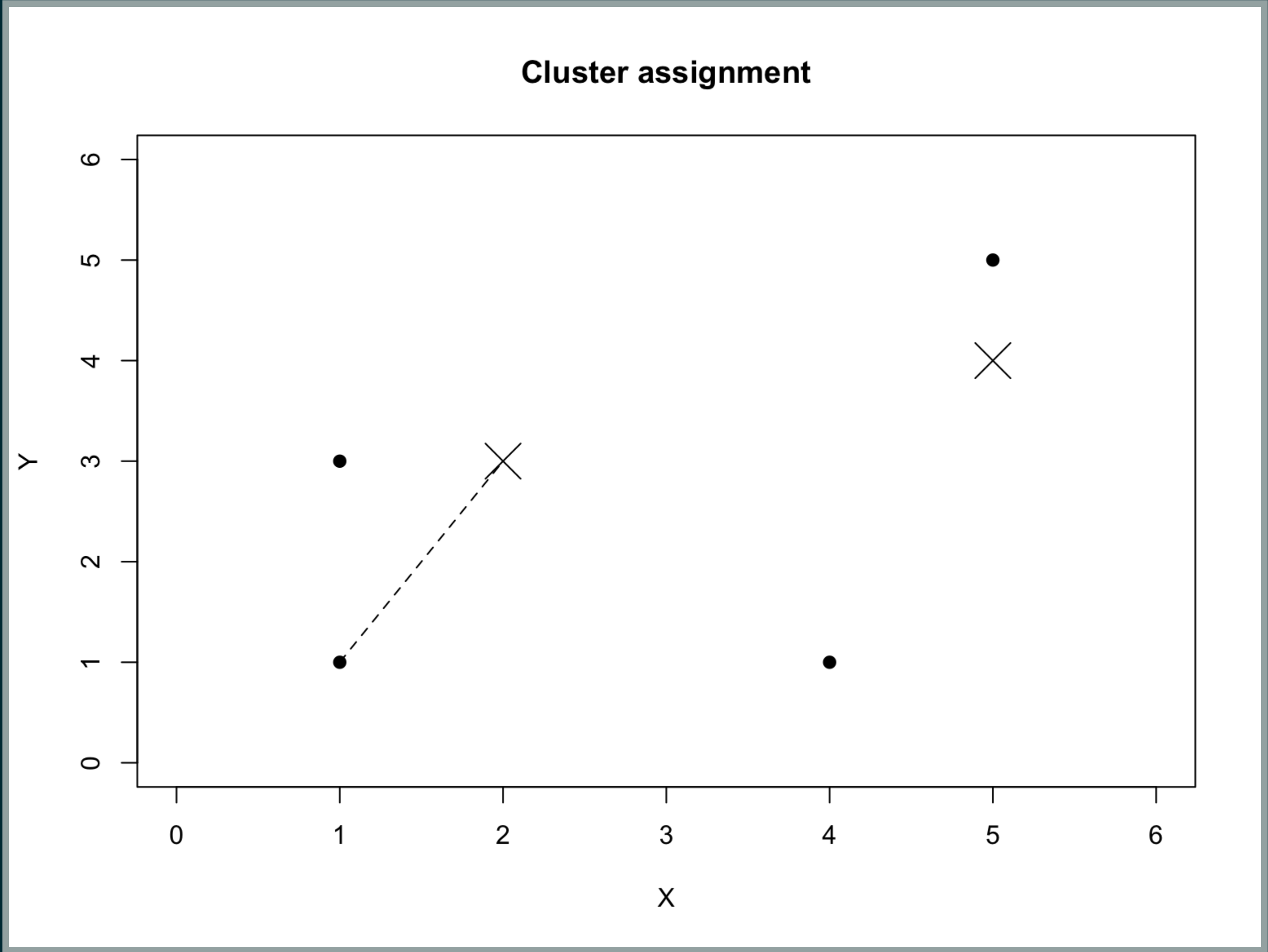
THE K-MEANS ALGORITHM IN DETAIL

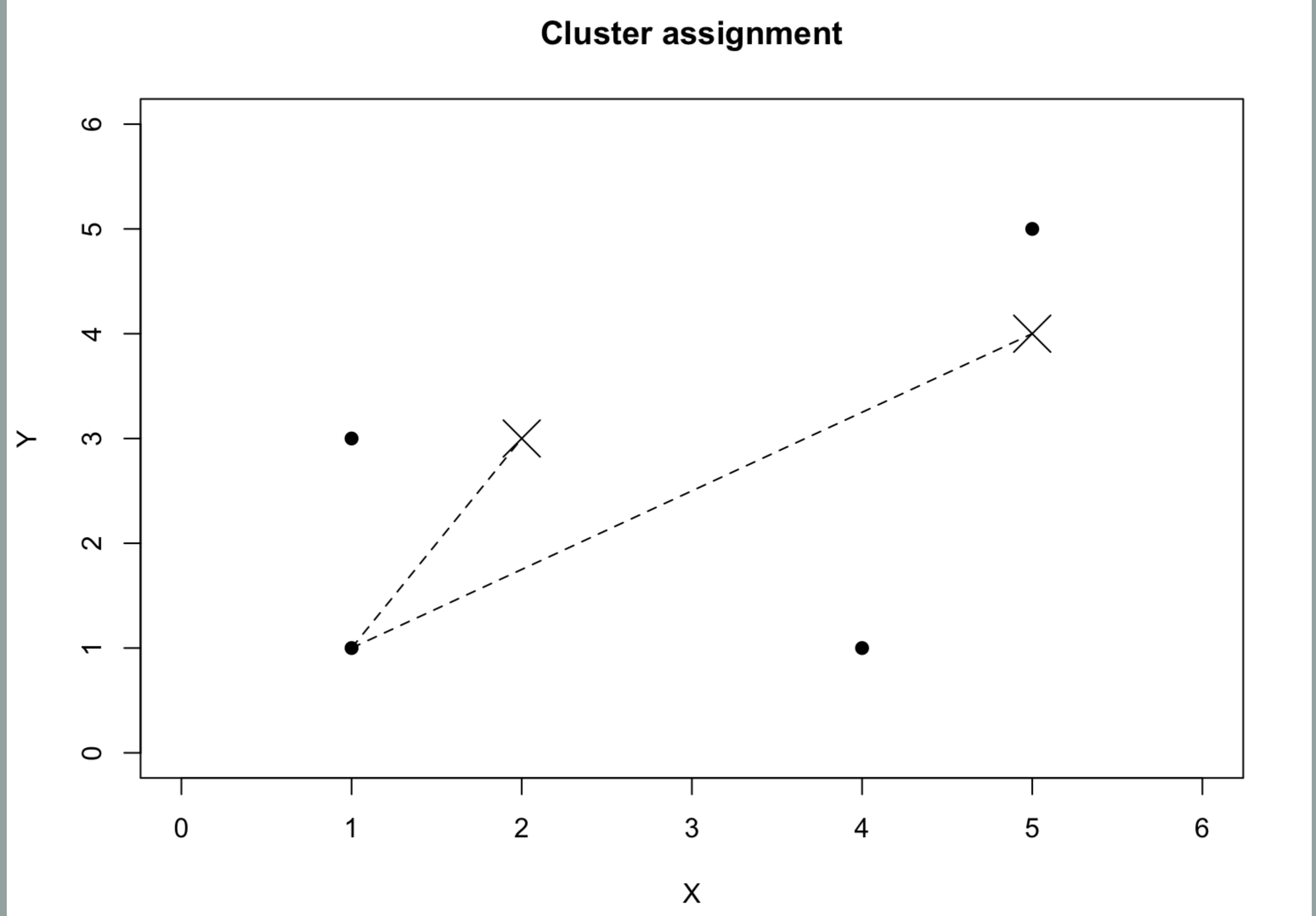
- set random centroids in n-dimensional space
- assign each observation to its closest centroid
- find new centroids
- re-assign the observations
- (iterative approach)

ASSIGNING CLUSTER MEMBERSHIP



OBTAINING DISTANCES (ERRORS)




Cluster assignment
Y
0
1
2
3
4
5
6
X
0
1
2
3
4
5
6

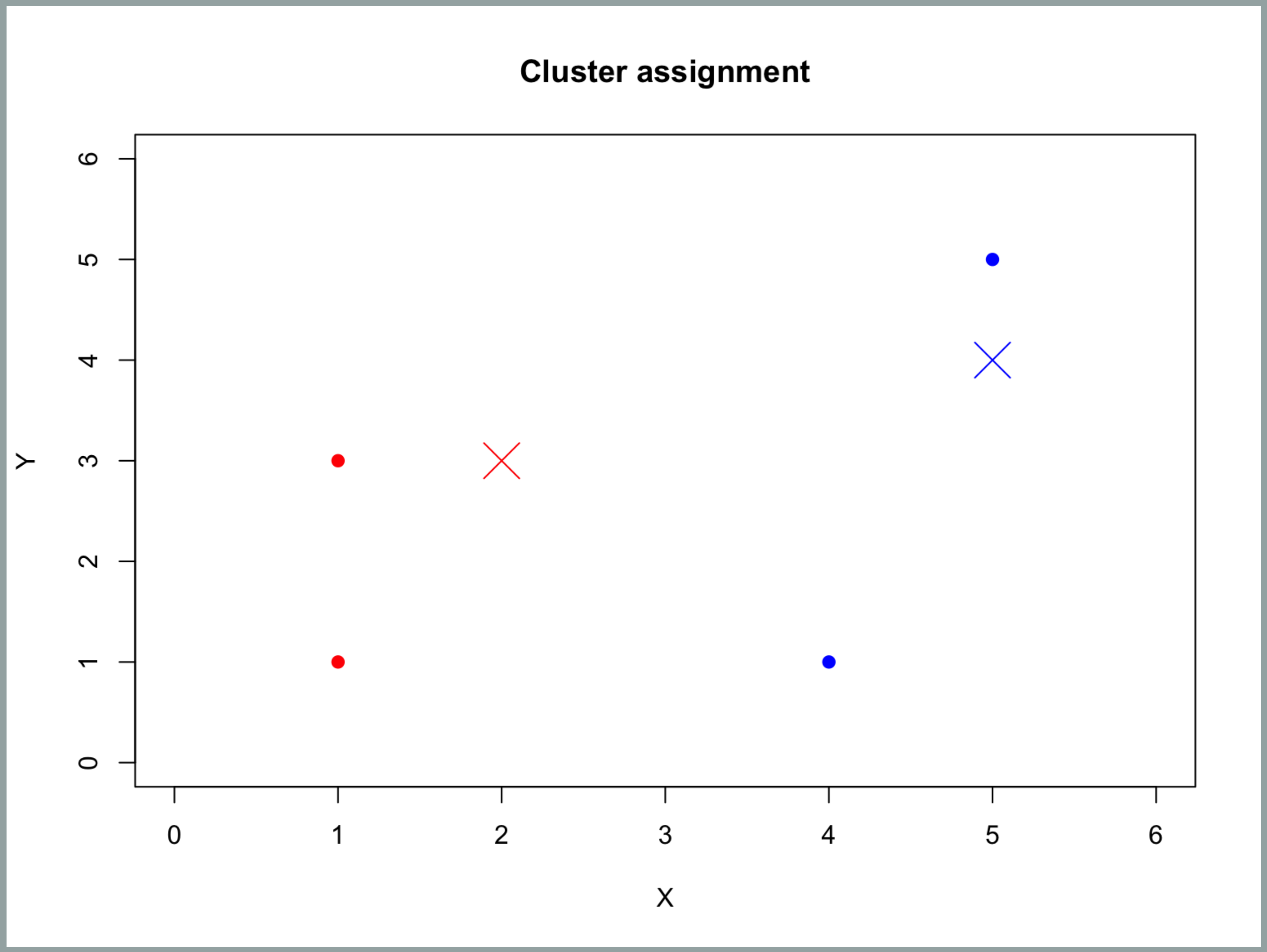
DISTANCE METRIC

- typically: Euclidean distance
- $dist(p, c) = \sqrt{(p_1 - c_1)^2 + (p_2 - c_2)^2}$

$$dist(p[1, 1], c[2, 3]) = \sqrt{(1 - 2)^2 + (1 - 3)^2} = \sqrt{5} = 2.24$$

Objective: $\arg\min D(p_i, c_j)$

AFTER DISTANCE-BASED ASSIGNMENT



NEW CENTROIDS: K-MEANS

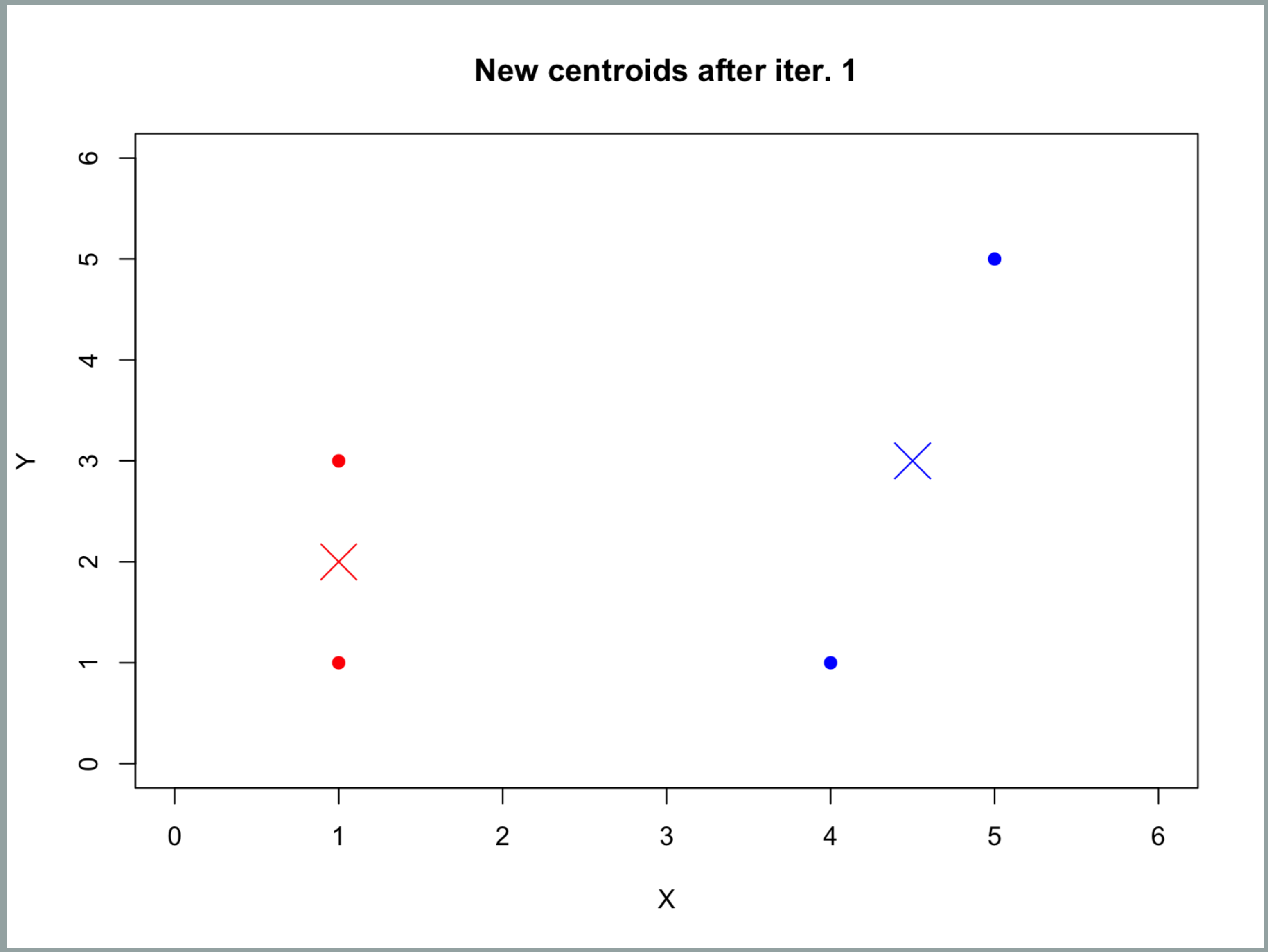
X	Y	Cluster
1	1	red
1	3	red
4	1	blue
5	5	blue

$$Mx_{red} = \frac{1+1}{2} = 1$$

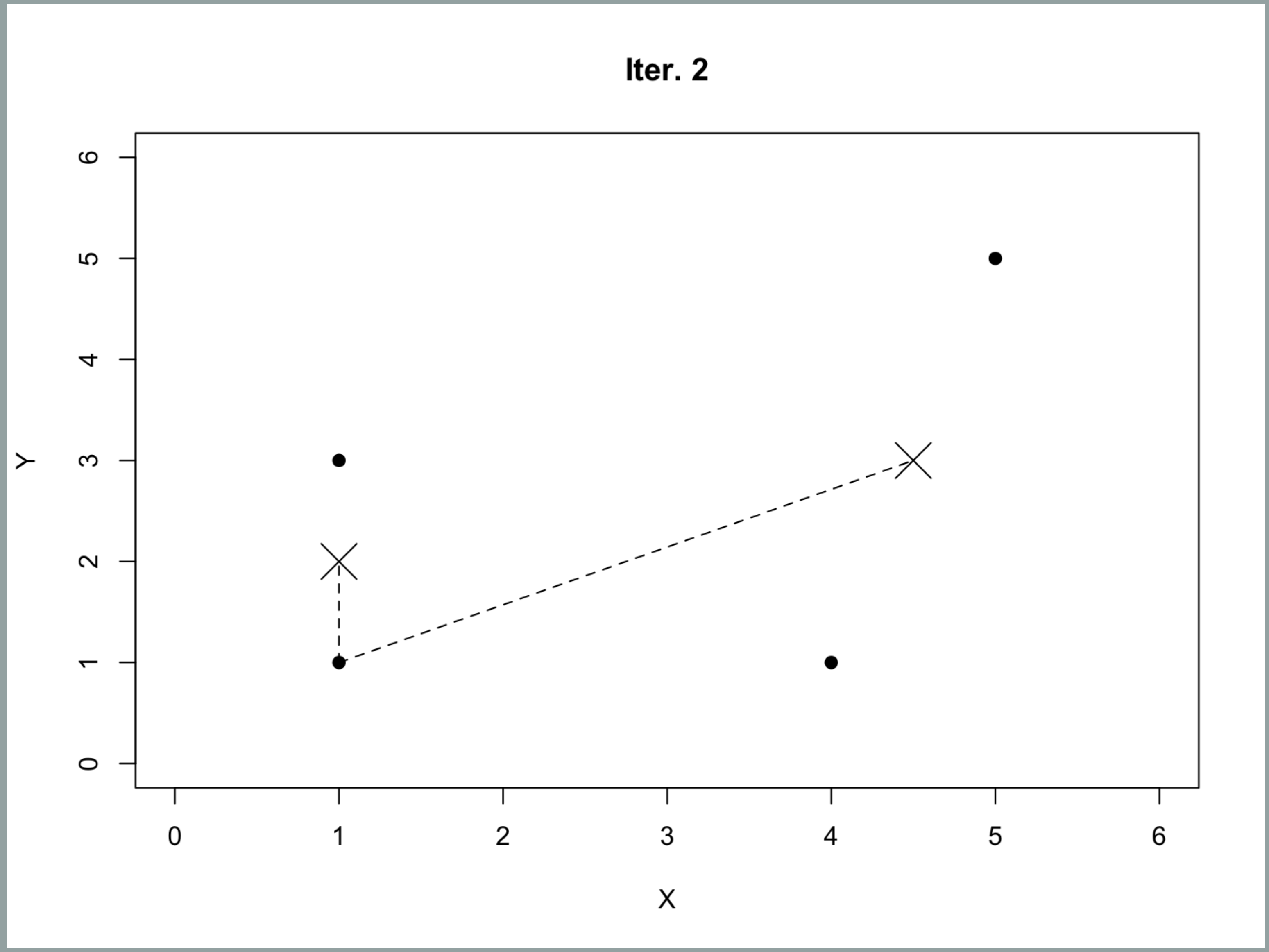
$$My_{red} = \frac{1+3}{2} = 2$$

$$M_{red} = [1, 2]$$

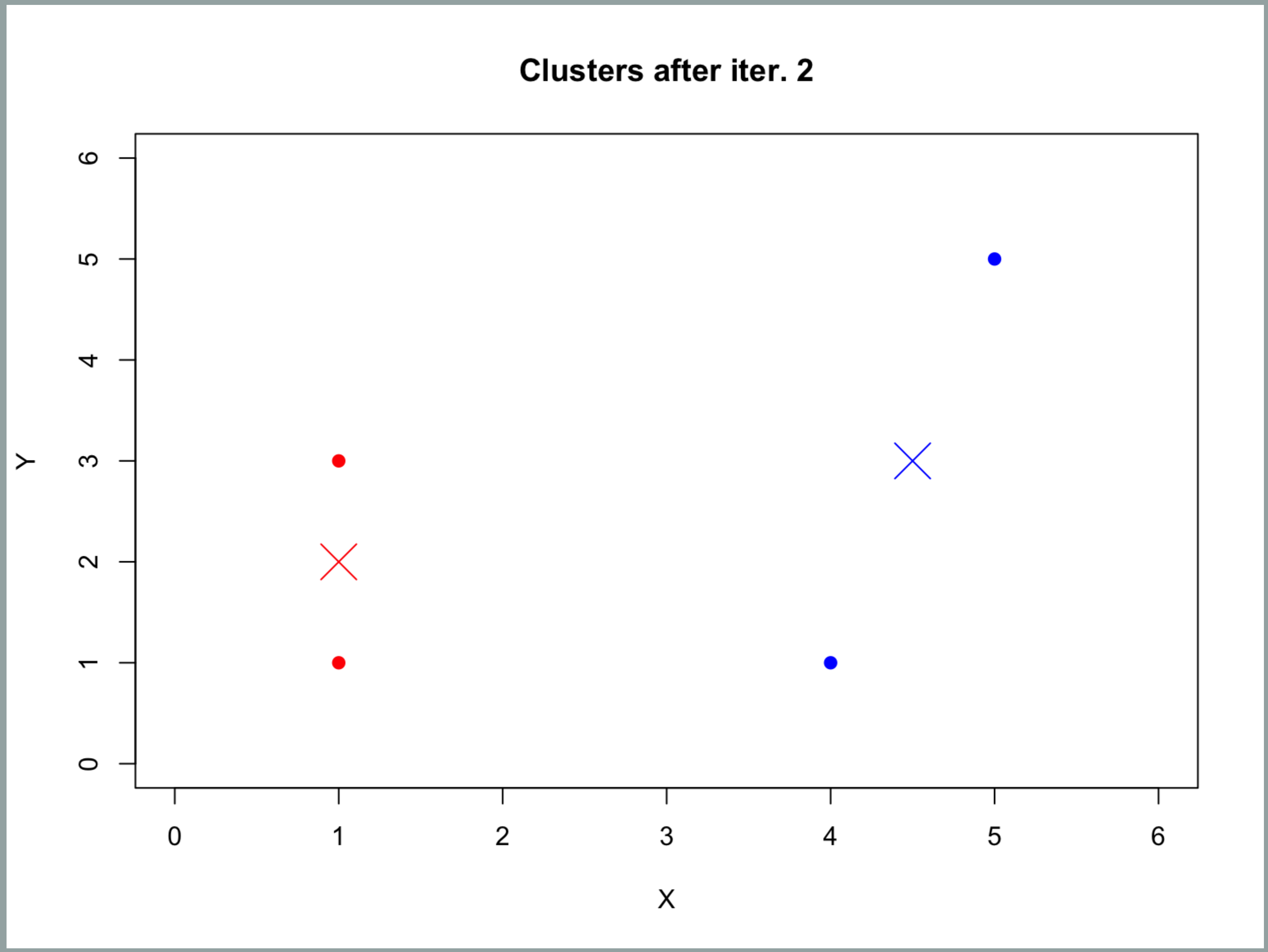
NEW CENTROIDS



ITERATION AFTER ITERATION



CLUSTER MEMBERSHIP AFTER ITERATION 2



STOPPING RULE

If any of these apply:

- convergence (i.e. no points change cluster membership)
- max. number of iterations (`iter.max = ...`)
- distance threshold reached

WHAT’S STRANGE ABOUT OUR APPROACH?

HOW DO WE KNOW k?

Possible approach:

- run it for n combinations: $k = 1, k = 2, \ldots k = n$
- assess how good k is

What does “good” mean?

DETERMINING *K*

WSS = within (cluster) sum of squares

- take difference between each point x_i in cluster c_j
- remember: c_j is now the mean of all points $x_{i,j}$
- so: we square the difference

$$\arg\min \sum_{x_{i,j}, c_j} (x_{i,j} - c_j)^2$$

CLUSTER DETERMINATION

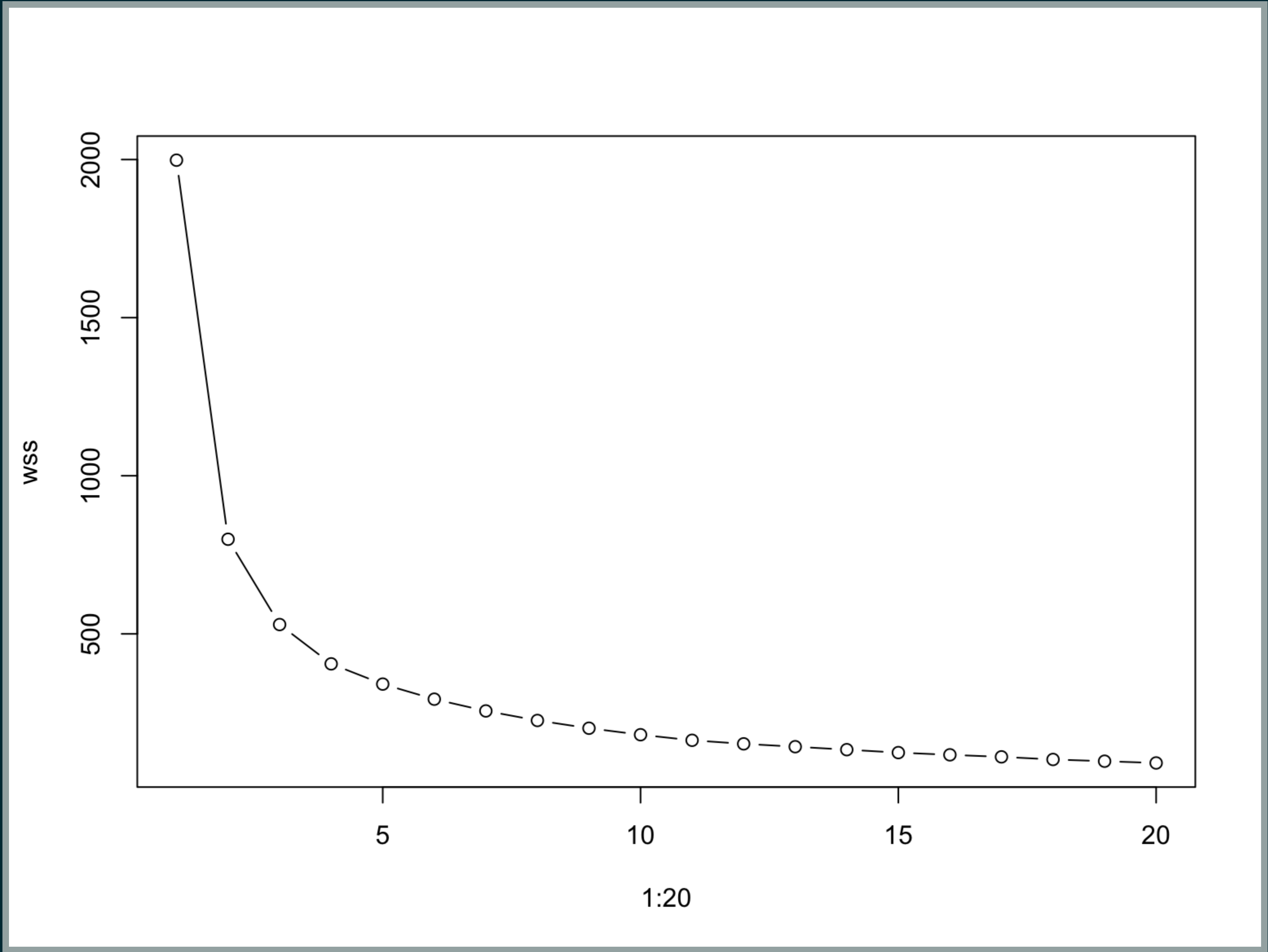
```
wss = numeric()
for(i in 1:20){
  kmeans_model = kmeans(data4, centers = i, iter.max = 20, nstart = 10)
  wss[i] = kmeans_model$tot.withinss
}
```

FOR $k = 1..k = 20$

```
wss
```

```
##  [1] 1998.00000  799.23145  529.42464  405.14898  341.16308  293.44305
##  [7]  256.25549  226.13568  201.62530  181.03906  163.43303  152.20691
## [13]  143.17168  133.78717  124.50437  117.49929  111.04724  102.77820
## [19]   97.30524   91.73814
```

SCREE PLOT (= THE ELBOW METHOD)

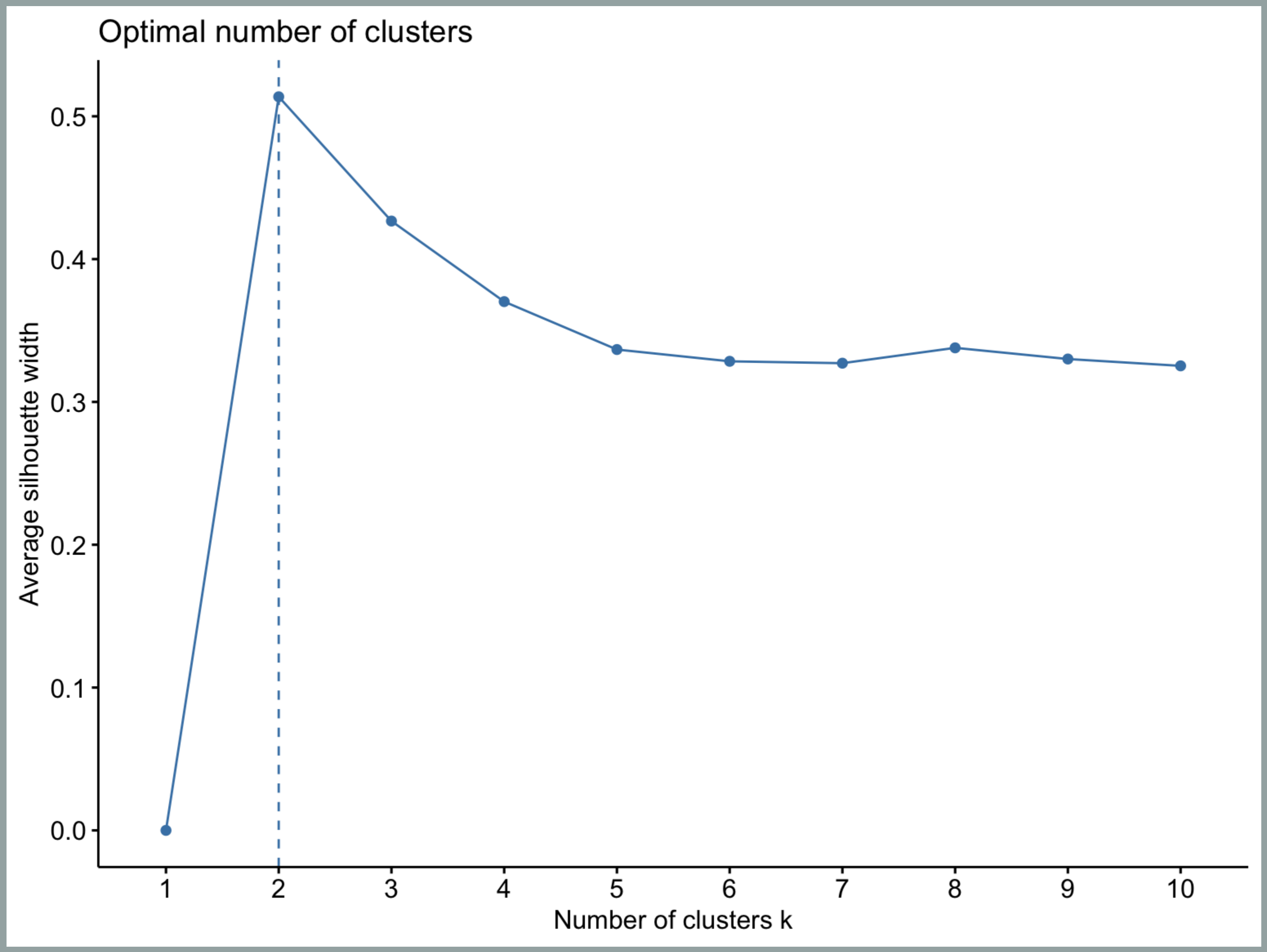


OTHER METHODS TO ESTABLISH K

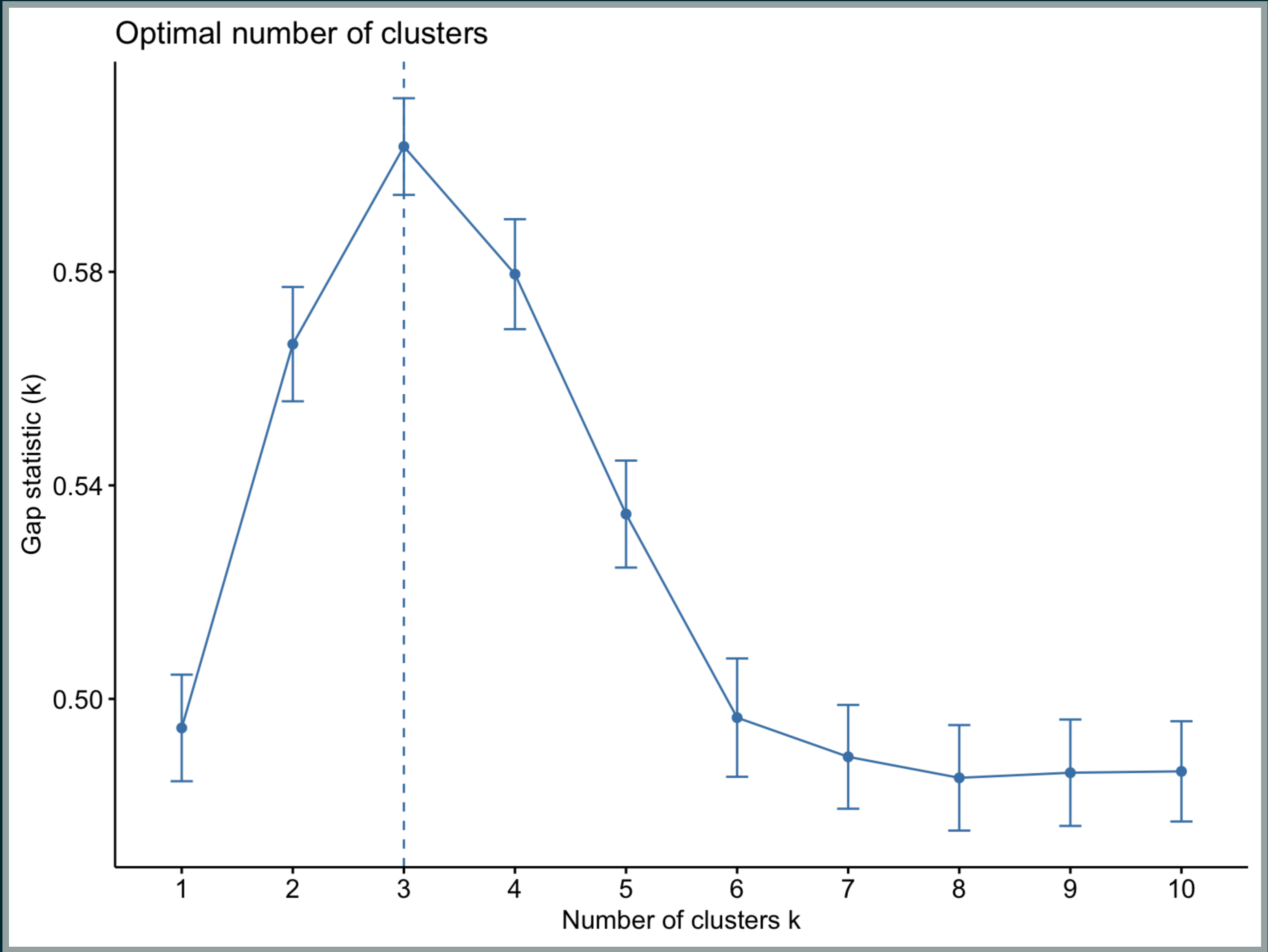
- Silhoutte method (cluster fit)
- Gap statistic

See also this tutorial.

SILHOUETTE METHOD



GAP STATISTIC

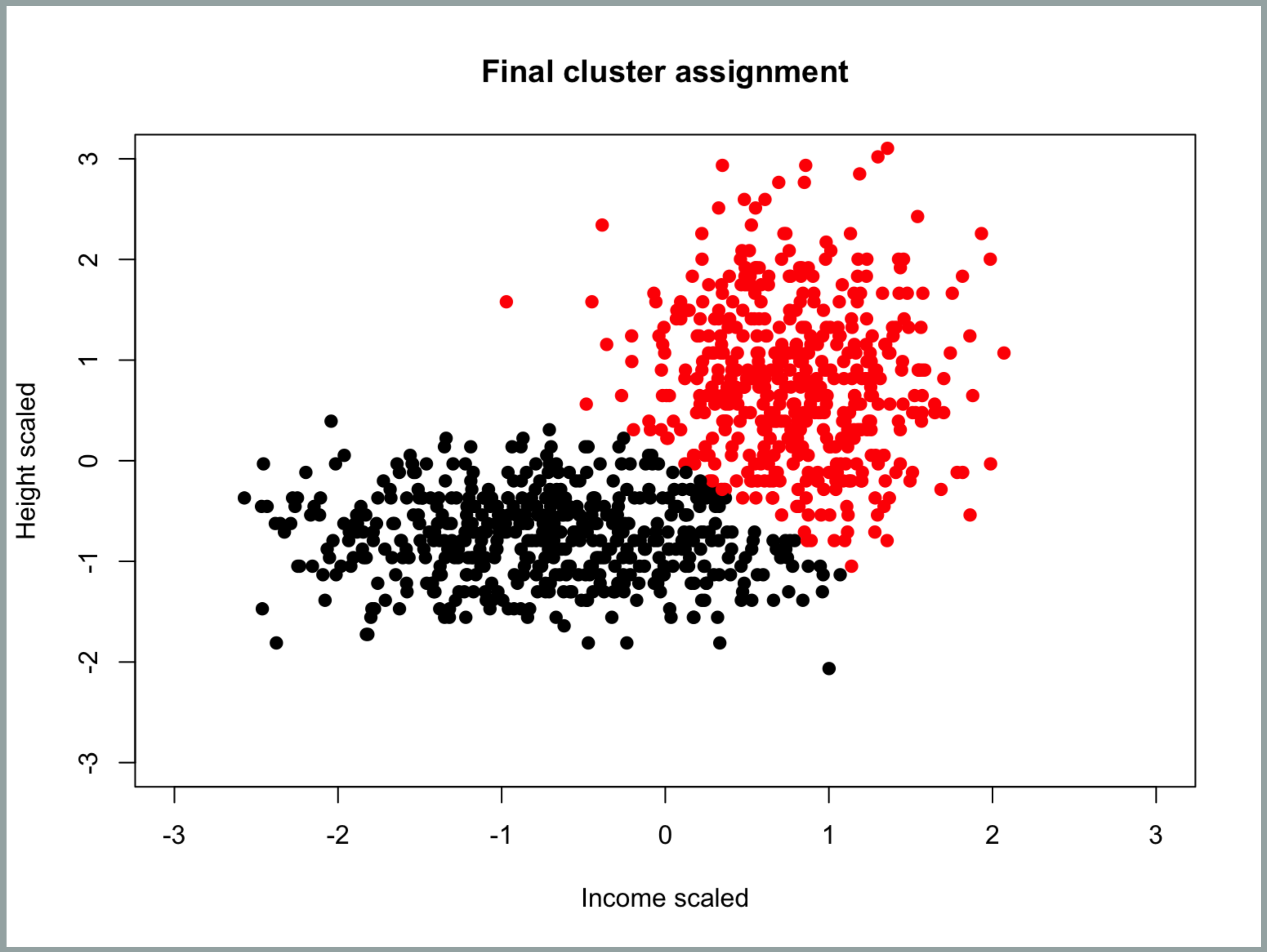


APPLYING K-MEANS CLUSTERING

We settle for $k = 2$

```
unsup_model_final = kmeans(data4
                        , centers = 2
                        , nstart = 10
                        , iter.max = 10)
```

PLOT THE CLUSTER ASSIGNMENT



OTHER UNSUPERVISED METHODS

- k-means (today)
- hierarchical clustering
- density clustering

ISSUES WITH UNSUPERVISED LEARNING

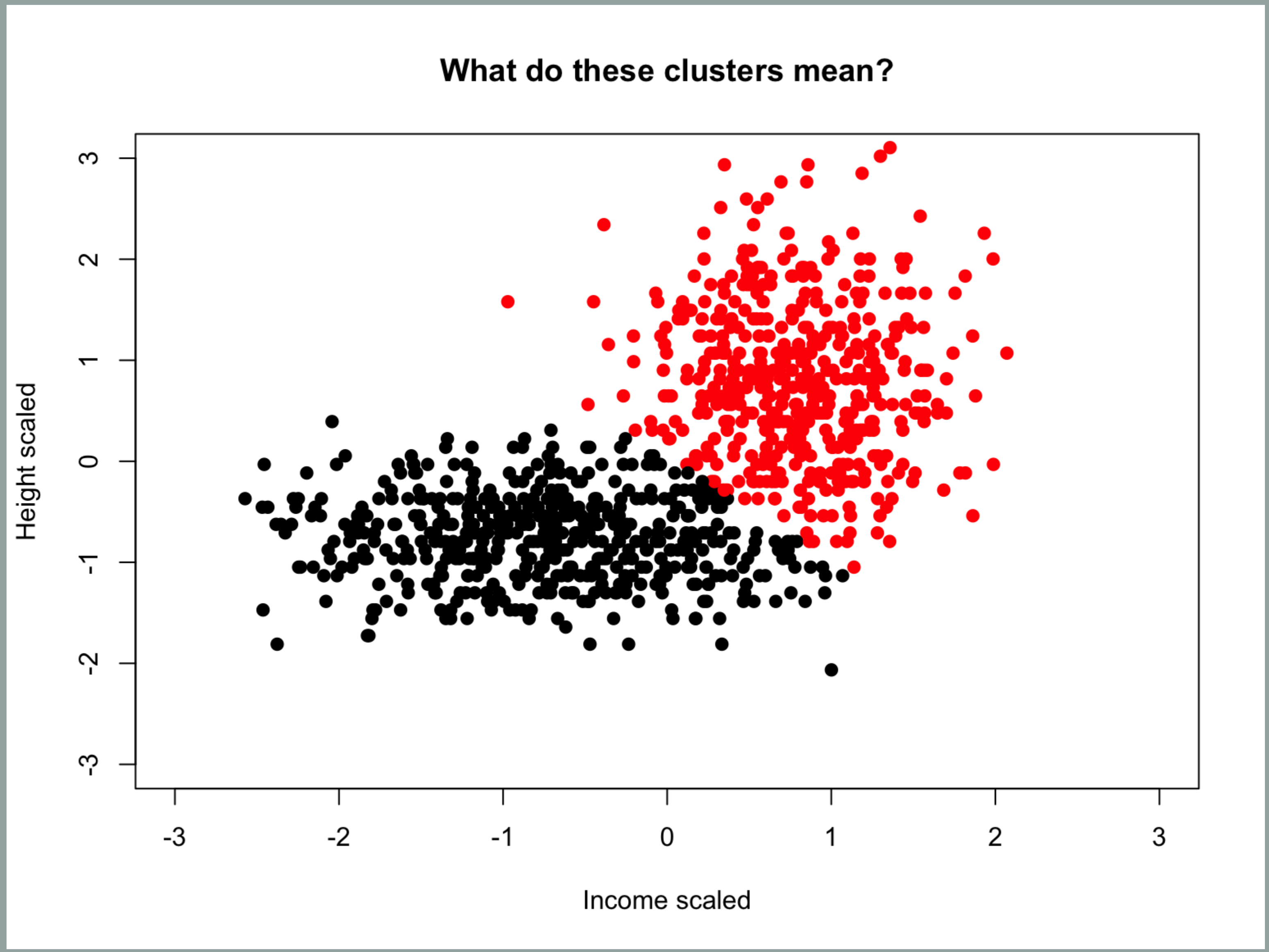
What’s lacking?

What can you (not) say?

CAVEATS OF UNSUP. ML

- there is no “ground truth”
- interpretation/subjectivity
- cluster choice

INTERPRETATION OF FINDINGS



INTERPRETATION OF FINDINGS

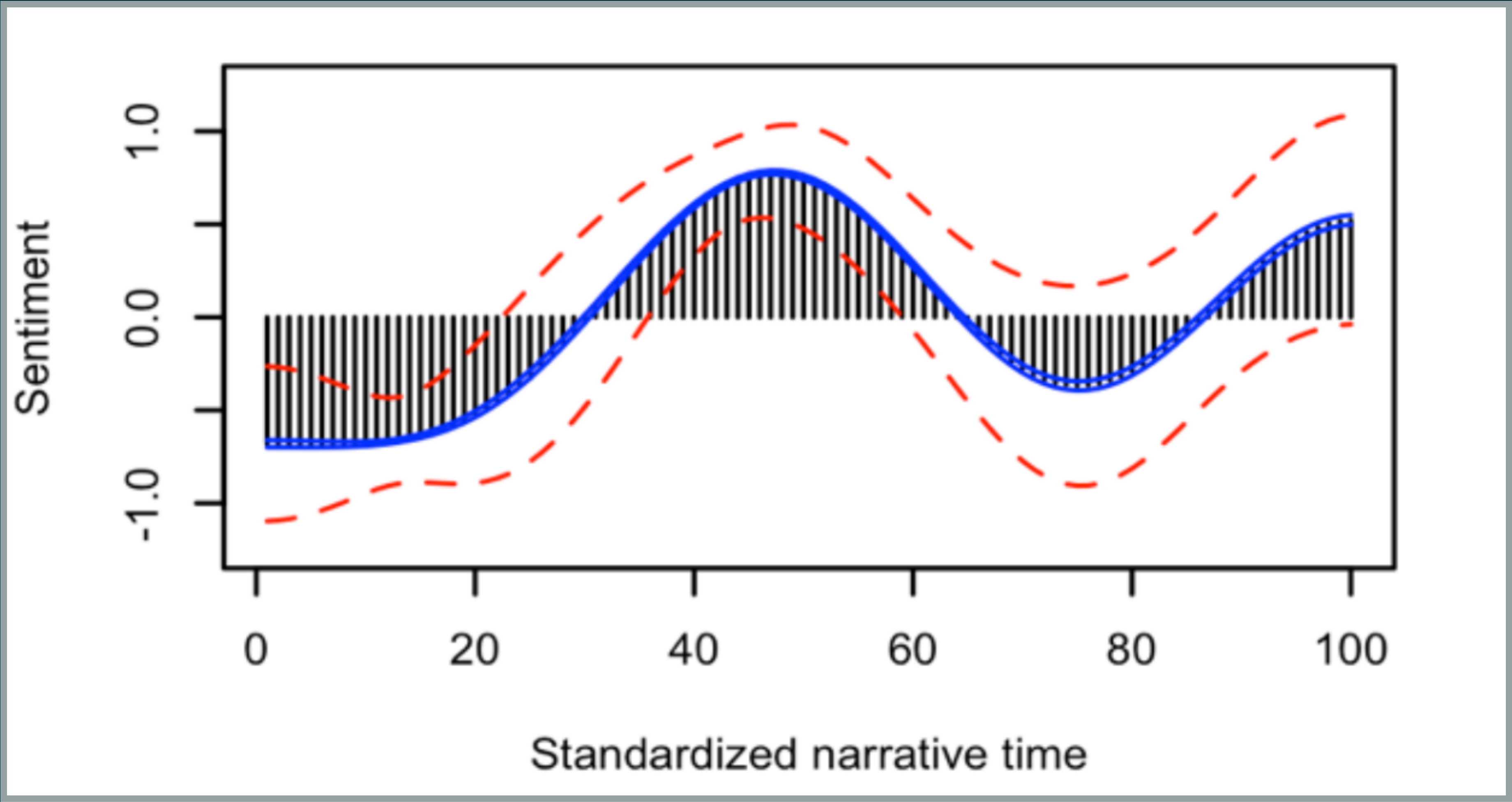
```
unsup_model_final$centers
```

```
##        salary     height
## 1 -0.7474895 -0.7551138
## 2  0.7937260  0.8018218
```

- Cluster 1: lower salary, shorter height
- Cluster 2: higher salary, larger height
- People in cluster 1 earn less and are shorter than those in cluster 2

We cannot say more than that!

INTERPRETATION OF FINDINGS

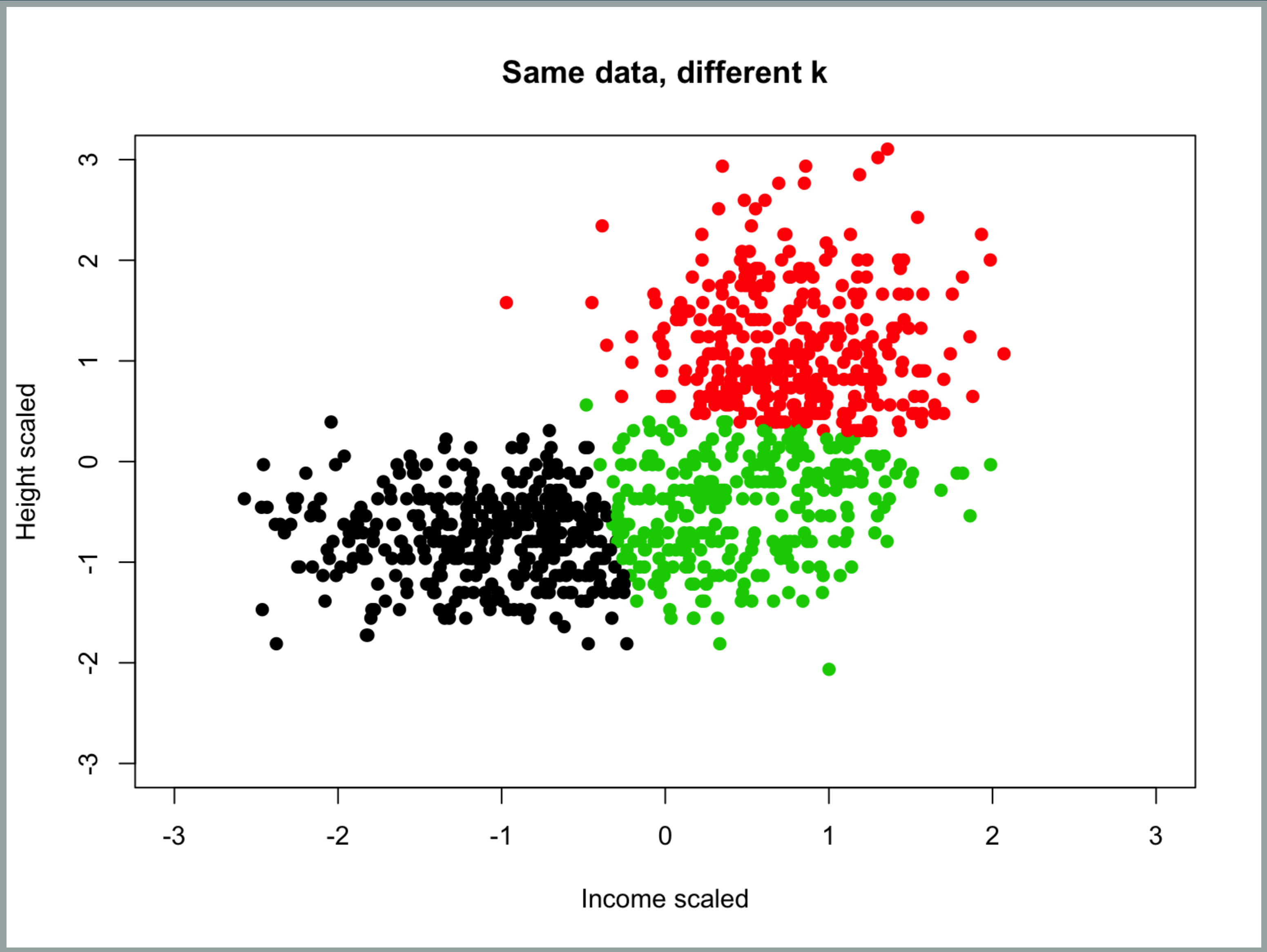


INTERPRETATION OF FINDINGS

- subjective
- labelling tricky
- researcher’s choice!
- be open about this

CLUSTER CHOICE

What if we chose $k = 3$?


Same data, different k
Height scaled
Income scaled
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

WHEN K CHANGES, THE INTERPRETATION CHANGES

```
km_3$centers
```

```
##        salary     height
## 1 -1.1253285 -0.7403048
## 2  0.7959880  1.1611042
## 3  0.4627853 -0.4561074
```

INTERPRETATION FOR K=3

- Cluster 1: avg-to-high salary, small
- Cluster 2: very low salary, small
- Cluster 3: high salary, very tall

CLUSTER CHOICE

- be open about it
- make all choices transparent
- always share code and data (“least vulnerable”" principle)

IMPORTANT

Note: we cannot say anything about accuracy.

See the k-NN model.

BIGGER PICTURE OF MACHINE LEARNING

- covered so far: supervised + unsupervised learning
- next week: neural networks

How do supervised and unsupervised learning relate to each other?

CASE EXAMPLE

- suppose you want to measure hate speech in the UK
- on Twitter
- and you have 10m Tweets of interest

POSSIBLE APPROACH

- you craft rules to determine hate speech vs non-hate speech
- problematic: might not capture all dynamics + costly

Better: supervised machine learning (text classification)

TEXT CLASSIFICATION APPROACH

- you annotate some data (typically crowdsourced)
- you build a supervised learning model
- with proper train-test splitting
- and assess the model with $Pr_{hatespeech}$

Suppose you have a good enough model.

REMEMBER

- the aim was to measure hate speech in the UK
- your model should now be good to annotate unlabelled data
- i.e. you can use the model on all Tweets
- and then answer the RQ

WHAT'S NEXT?

- Today's tutorial + homework: unsupervised learning in R

Next week: Machine Learning 3