kmeans Clustering 101

Bert Gollnick

kmeans Clustering

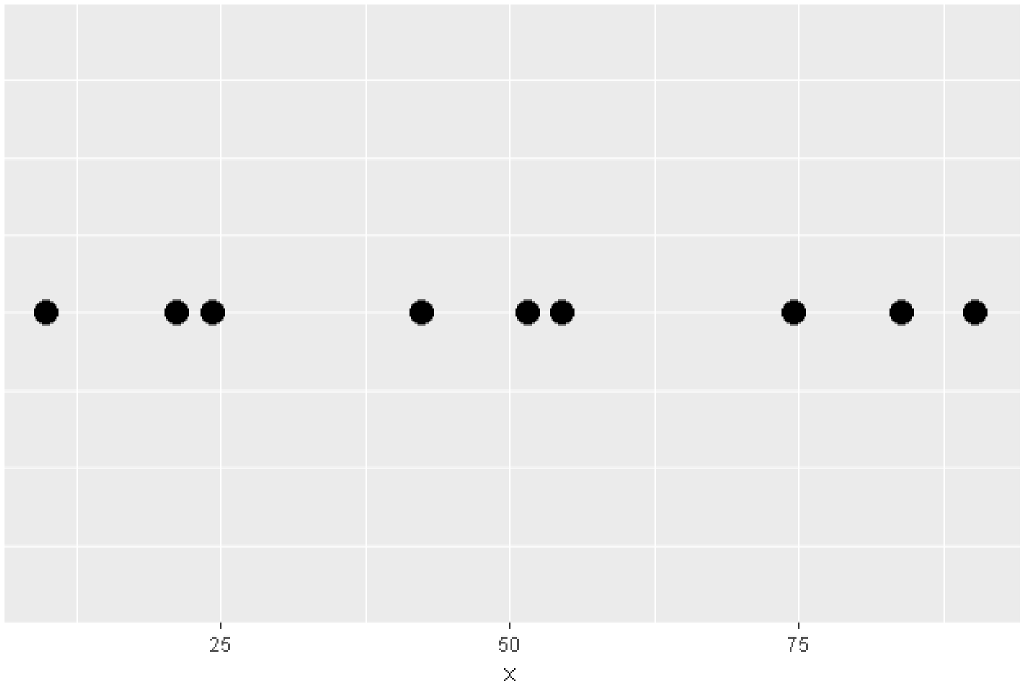
Introduction

- Clustering technique
- Most commonly used technique
- Similar to k nearest neighbor algorithm
- Assigns all observations to clusters
- Cluster number needs to be defined by user in advance
- Algorithm
  - minimizes differences within clusters and
  - maximizes differences between clusters
- Uses heuristics to find optimum (depending on starting points; adds randomness)

kmeans Clustering

Example

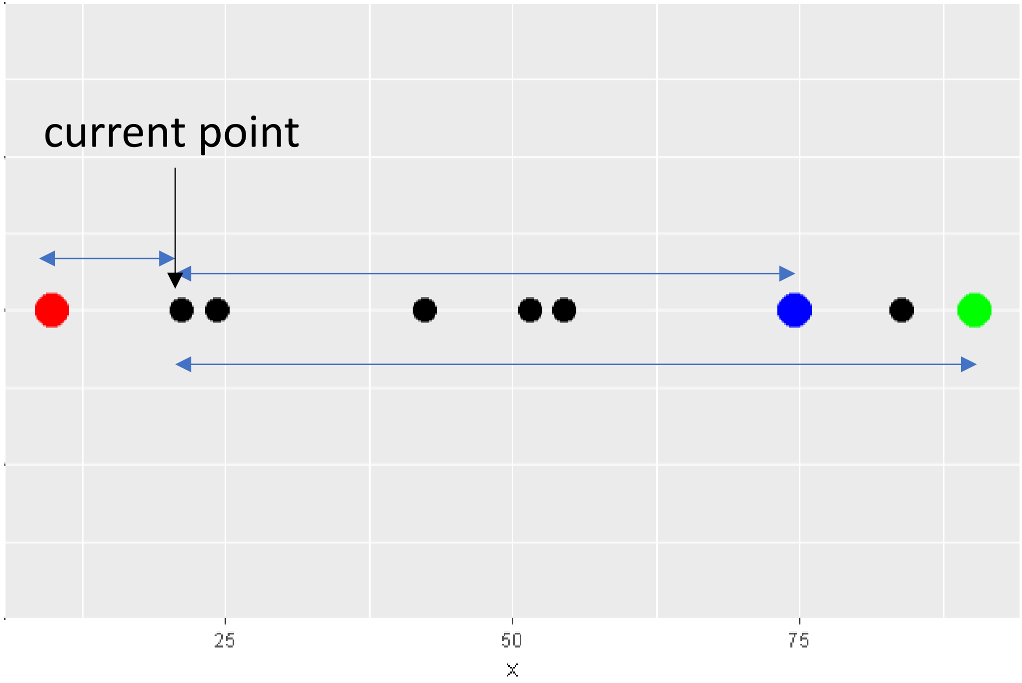
- Points in one-dimension shall be clustered
- Say: three clusters



kmeans Clustering

Example

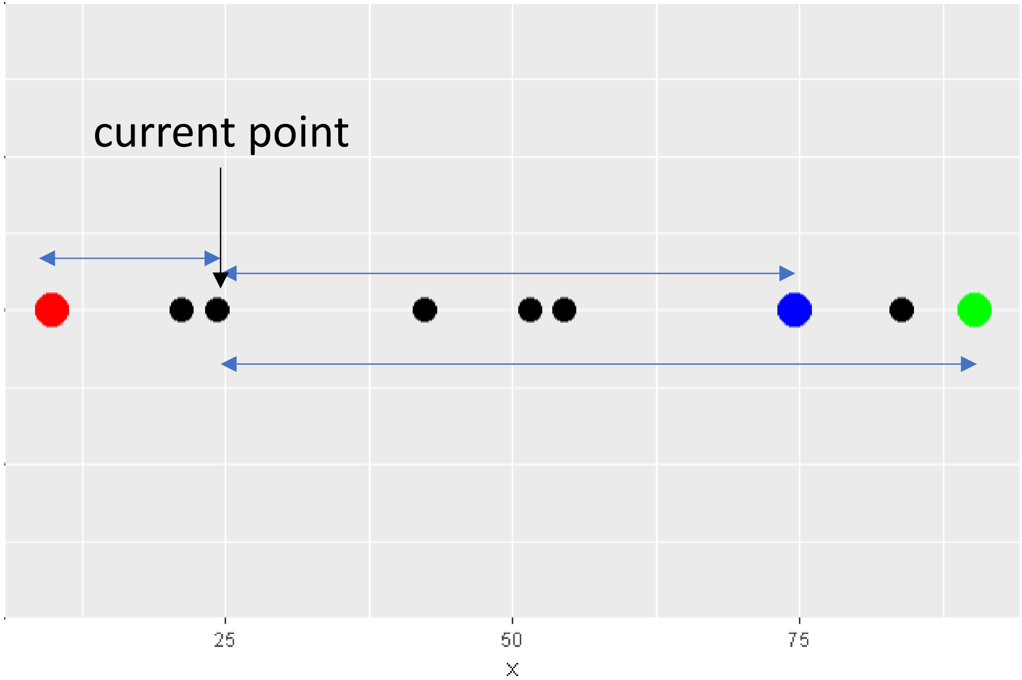
- Randomly choose three points as first cluster centers
- Calculate distance from each point to each cluster



kmeans Clustering

Example

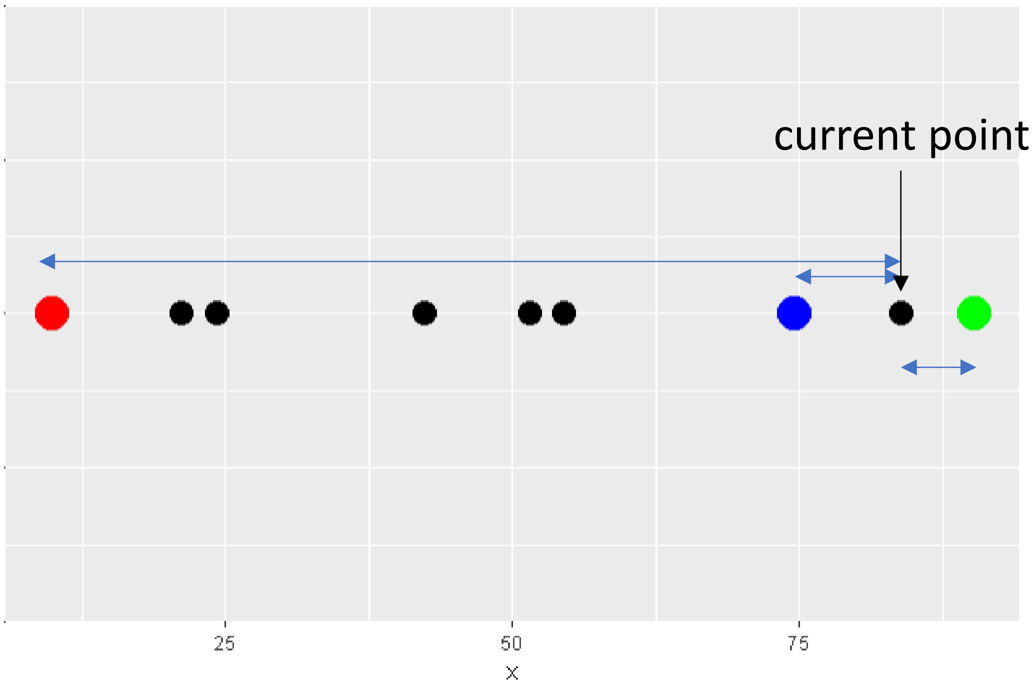
- Randomly choose three points as first cluster centers
- Calculate distance from each point to each cluster



kmeans Clustering

Example

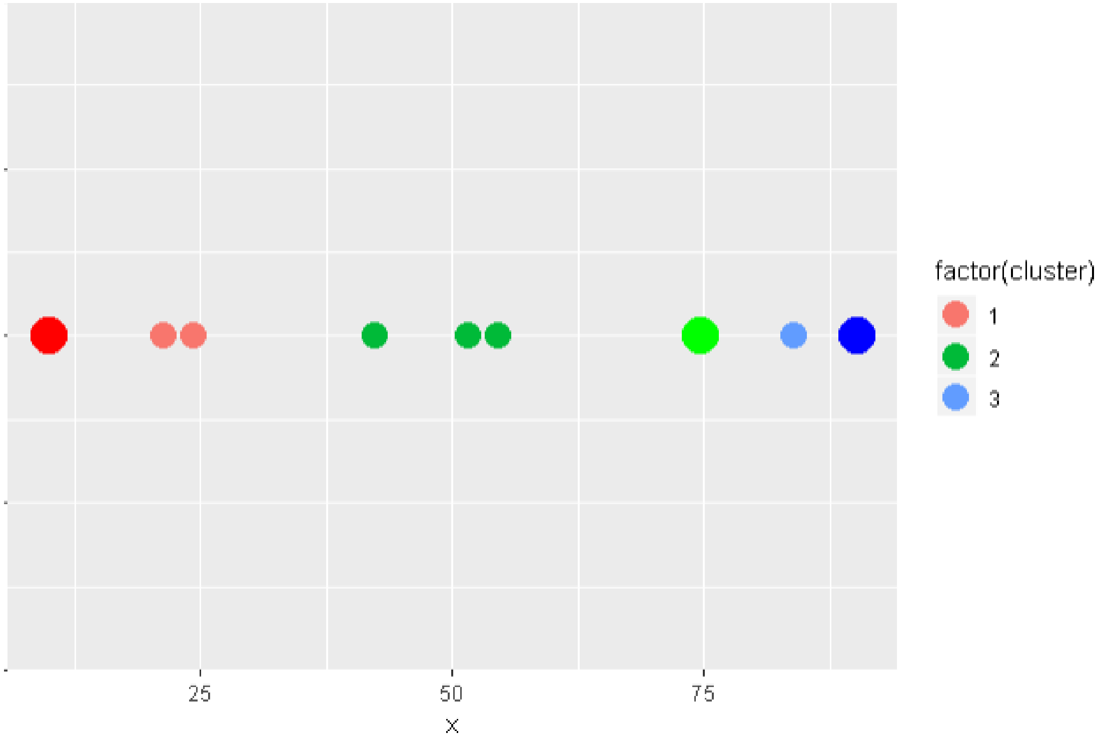
- Randomly choose three points as first cluster centers
- Calculate distance from each point to each cluster



kmeans Clustering

Example

- Assign points to closest cluster



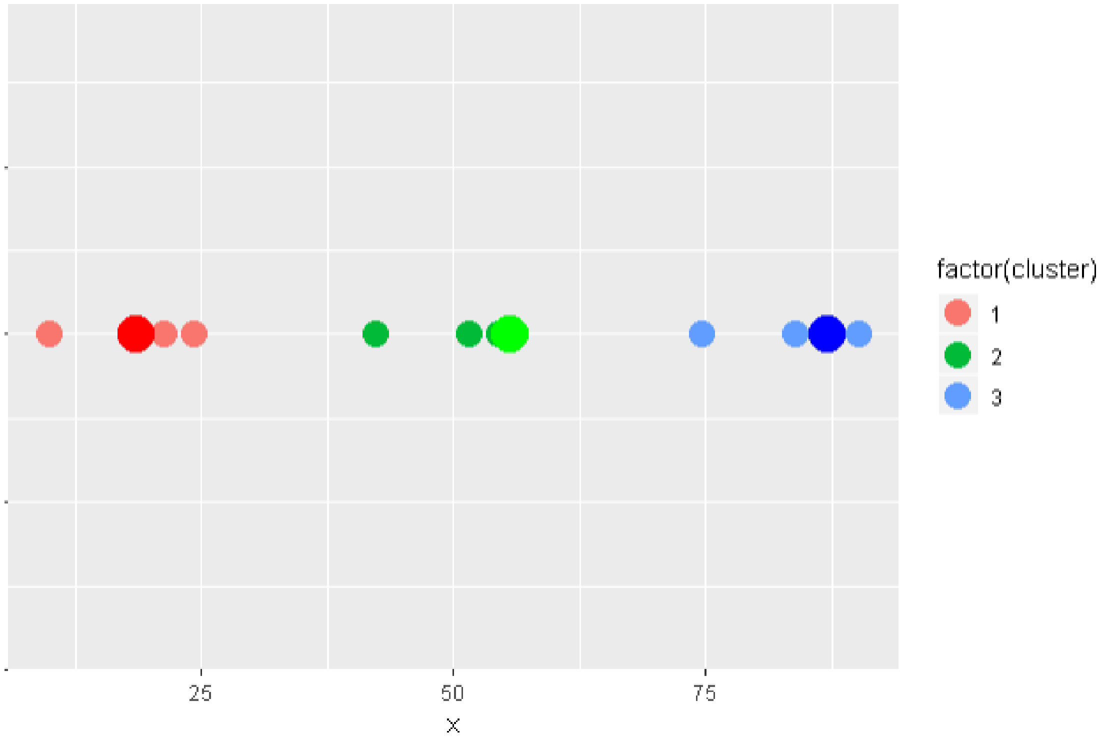
kmeans Clustering

Example

- Calculate new cluster means

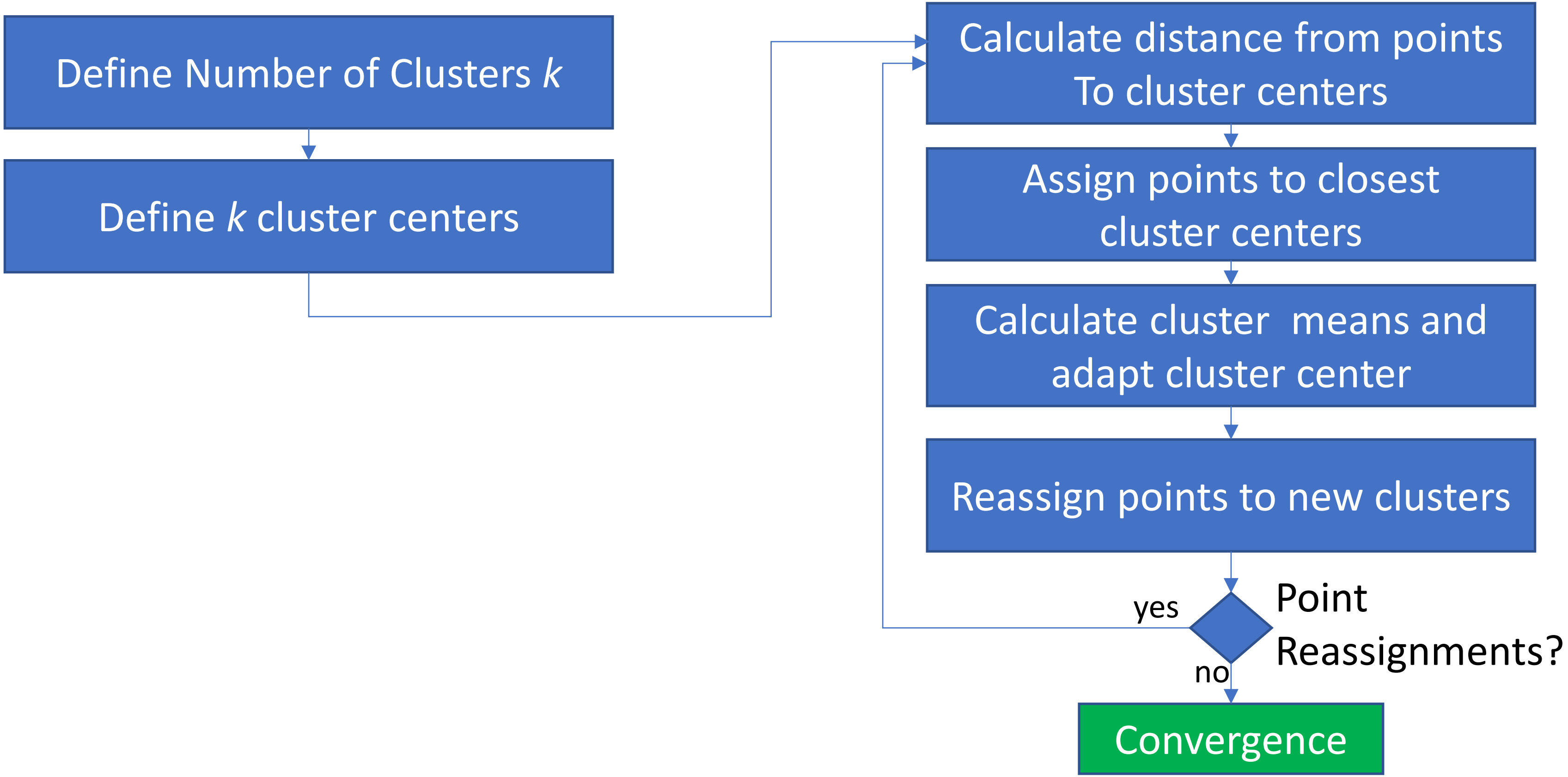
- Calculate distances from points to cluster centers
- Assign points to closest cluster

- Iterate until assignments don't change any more



kmeans Clustering

Workflow

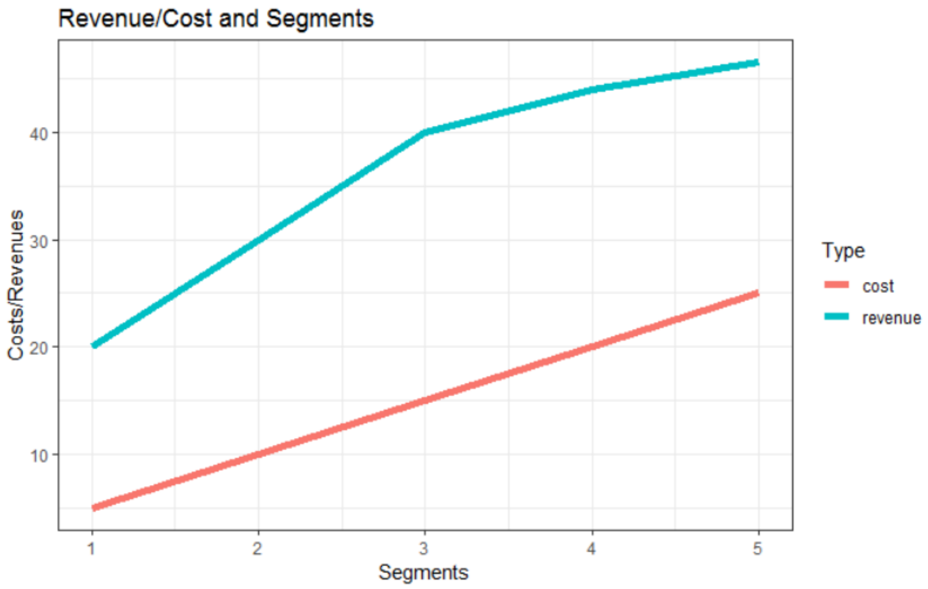


kmeans Clustering

Cluster Number k: Why is there an optimum?

- Why is there an optimum?
- What is the optimum number?

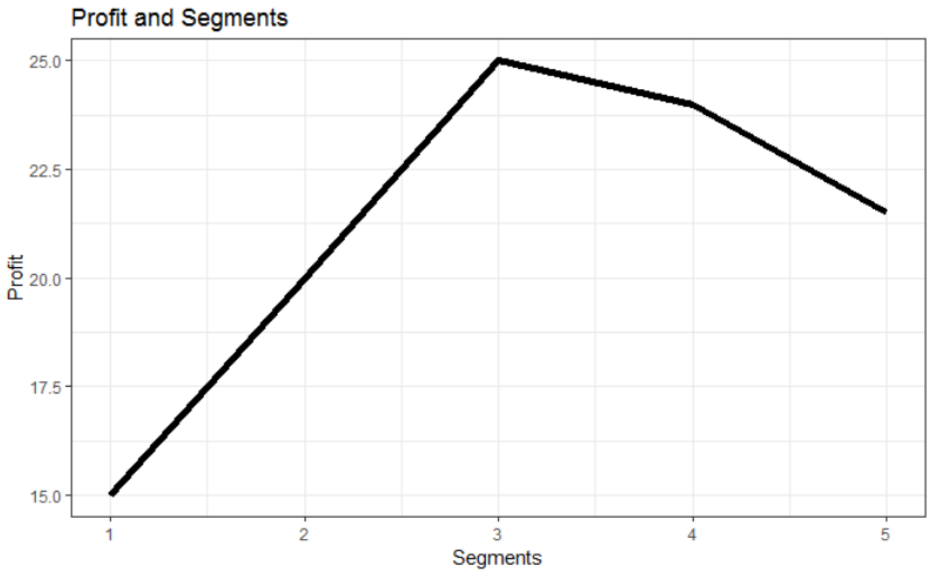
- Example: Marketing Campaign
  - Segment customers into groups
  - Prepare a newsletter specific to each group
  - Being more specific
    - increases revenues (asymptotic)
    - Increases cost (linear)



kmeans Clustering

Cluster Number k: Why is there an optimum?

- Example: Marketing Campaign
  - Profit = Revenue – Cost
  - There is a max Profit!
  - Choose nr. of segments for max profit



kmeans Clustering

Cluster Number k: Setting the optimum

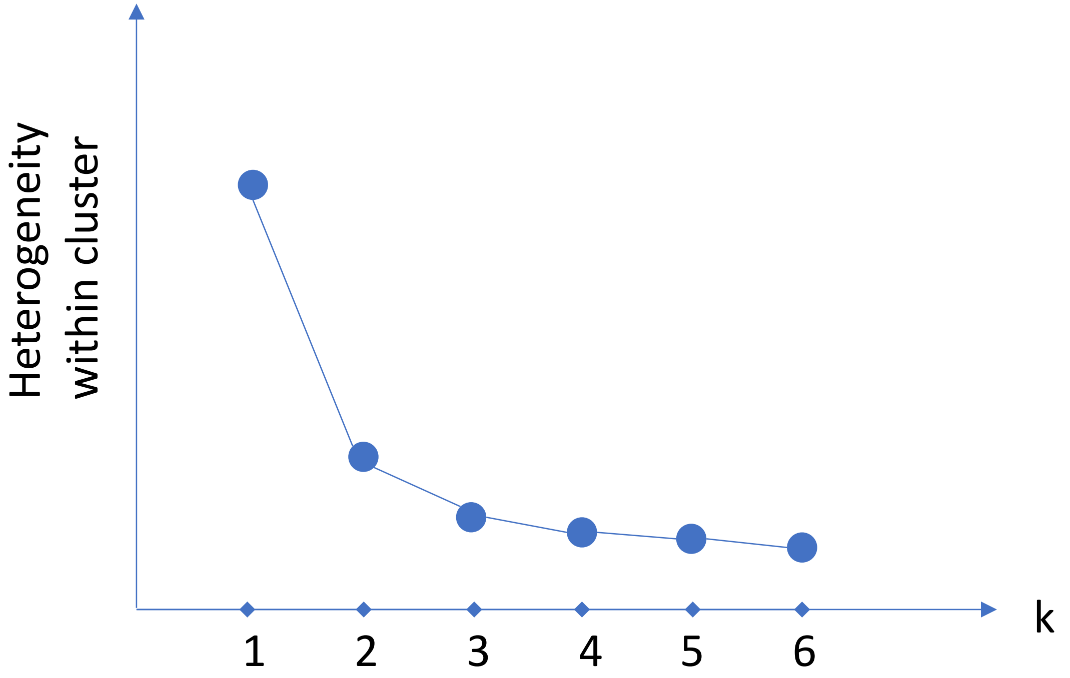
- Different clusters created with range of *k*
- *k* too large
  - Data homogeneous within clusters
  - risk of overfitting
  - Extreme: k = number of observations
- *k* too small
  - Variation within clusters too large
  - Risk of underfitting
- Ideally: domain knowledge on “good” *k*

kmeans Clustering

Cluster Number k: Setting the optimum

Elbow Method

- Analyses heterogeneity within clusters over *k*
- Goal: not minimizing
- Opt k:
  - strong decrease of hetero-geneity to this k
  - Larger k diminishing returns



kmeans

Advantages / Disadvantages

+

- Simple to understand
- Very flexible
- Performs well on many datasets

-

- „grandfather" of clustering – not as powerful as recent offsprings
- Inherits randomness → optimum not always found
- Requires „educated guess" on number of clusters