Departemen Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor

IPB University
Bogor Indonesia

PEMODELAN KLASIFIKASI

PERTEMUAN #2

K-MEANS

Bagus Sartono
bagusco@apps.ipb.ac.id
2020

AGENDA

Pengenalan R

K-Means

PENGENALAN R

- Bekerja dengan dataframe
  - Import data
  - Menampilkan dataframe dan informasi umumnya (nama kolom, ukuran)
  - Mengakses kolom dan menggunakannya (deskripsi data)
  - Subsetting
  - Menambahkan kolom baru
- Basic programming

K-MEANS

Andaikan terdapat *n* buah amatan $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, …, $\boldsymbol{x}_n$.

Masing-masing amatan akan dikelompokkan ke dalam satu dari *k* buah kelompok. Besaran *k* umumnya jauh lebih kecil dibandingkan *n.*

Andaikan $\boldsymbol{c}_1$, …, $\boldsymbol{c}_k$ adalah centroid dari *k* buah kelompok.

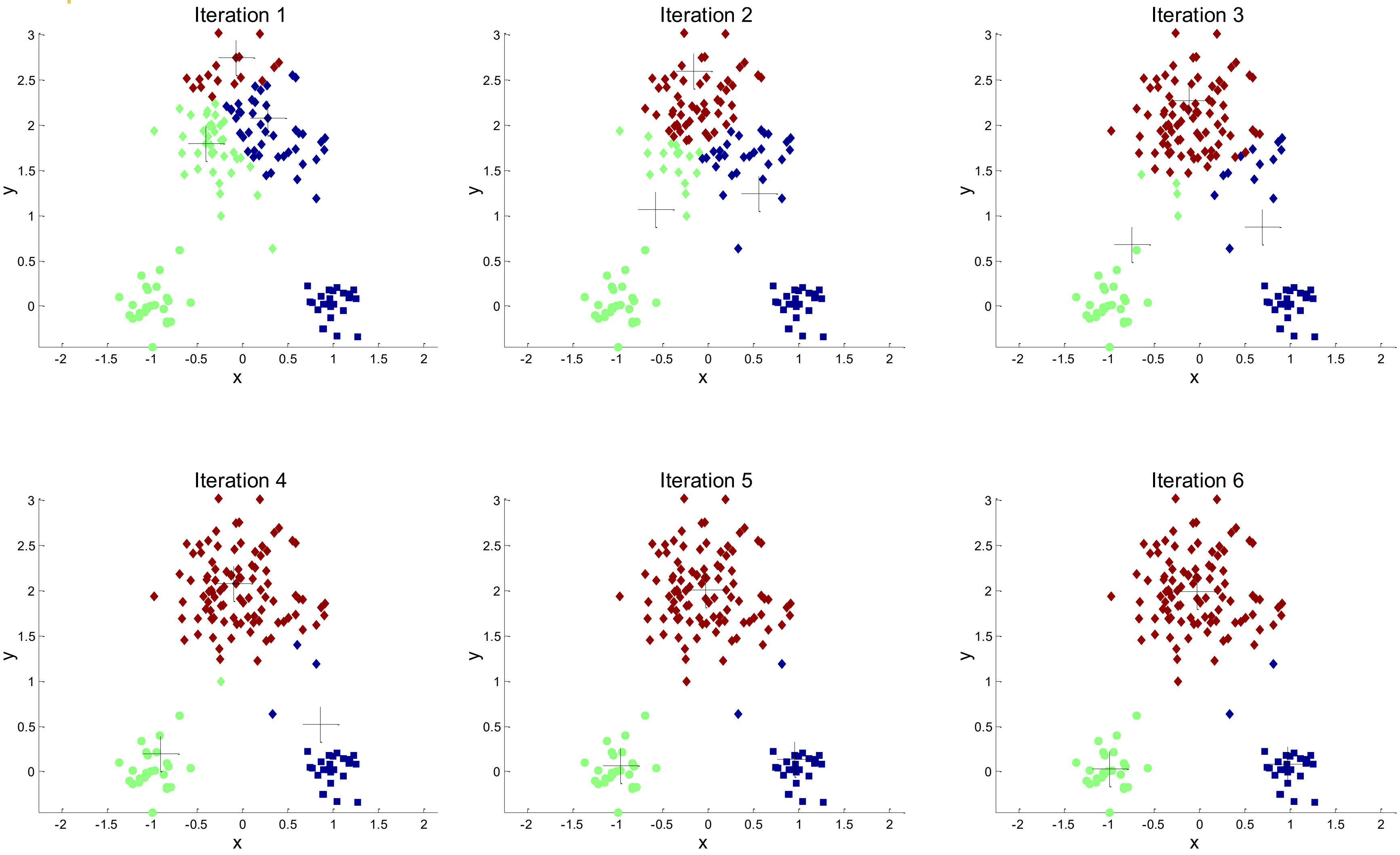
Intuisi dari pengelompokan adalah bahwa amatan akan dimasukkan ke kelompok *j* jika amatan tersebut memiliki jarak paling dekat dengan $\boldsymbol{c}_j$ dibandingkan dengan centroid lainnya.

ALGORITMA

1. Tentukan secara acak $\mathbf{c}_1, \ldots, \mathbf{c}_k$
2. Hitung jarak dari setiap $\mathbf{x}_i$ ke $\mathbf{c}_j$
3. Masukkan amatan ke-*i* ke dalam kelompok ke-*j* jika d_{ij} adalah yang paling kecil dibandingkan $d_{ij'}$
4. Perbarui $\mathbf{c}_1, \ldots, \mathbf{c}_k$ dengan menghitung rata-rata dari semua $\mathbf{x}_i$ yang menjadi anggota kelompok masing-masing
5. Kembali ke tahap 2 sampai konvergen

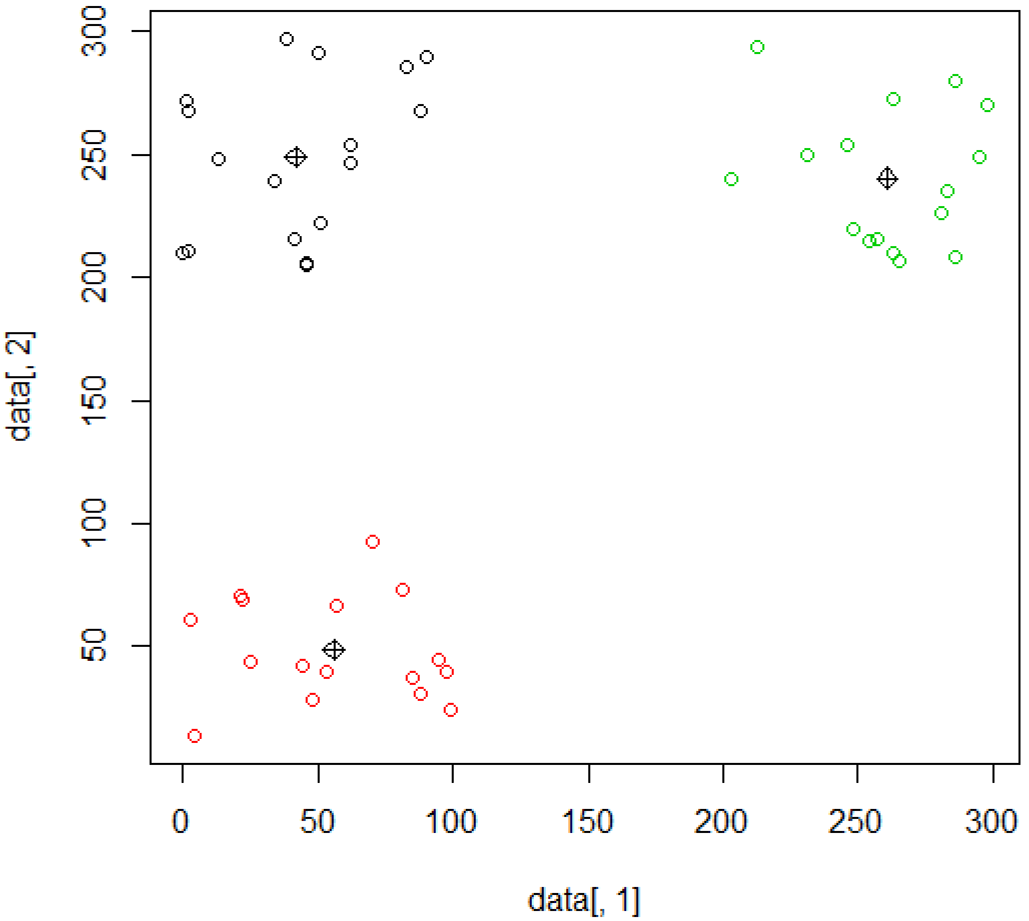
How do we decide when to stop?

One criterion for stopping is if we observe the assignment functions in the two iterations are exactly the same. If the assignment function doesn't change anymore, then the centroid won't change either (and vice versa).


Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
y
3
2.5
2
1.5
1
0.5
0
x
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

```
setwd ("D:/bagusco/Kuliah S2 --- Pemodelan Klasifikasi/data")
data <- read.csv("ilustrasikm.csv")

cluster <- kmeans(data, 3)
plot(data[,1], data[,2], col=cluster$cluster)
points(cluster$centers, pch=9)
```



Penentuan banyaknya cluster

Dua alternatif

- Ditentukan oleh peneliti
- Didasarkan pada data

Pada studi segmentasi, banyaknya cluster melambangkan banyaknya segmen

Preferable approach: *"let the data speak"*

- Didasarkan pada statistik tertentu
- Memungkinkan terjadinya perdebatan dan ketidakpastian

Ilustrasi situasi dimana banyaknya cluster ditentukan oleh peneliti

A retailer wants to identify several shopping profiles in order to activate new and targeted retail outlets

The budget only allows him to open three types of outlets

A partition into three clusters follows naturally, although it is not necessarily the optimal one.

Fixed number of clusters and (*k*-means) non hierarchical approach

Ilustrasi situasi dimana biarkan data yang menentukan banyaknya cluster

Clustering of shopping profiles is expected to detect a new market niche.

For market segmentation purposes, it is less advisable to constrain the analysis to a fixed number of clusters

- A hierarchical procedure allows to explore all potentially valid numbers of clusters
- For each of them there are some statistical diagnostics to pinpoint the best partition.
- What is needed is a *stopping rule* for the hierarchical algorithm, which determines the number of clusters at which the algorithm should stop.

Statistical tests are not always univocal, leaving some room to the researcher's experience and arbitrariness

Statistical rigidities should be balanced with the knowledge gained from and interpretability of the final classification.

Determining the optimal number of cluster

Kriteria

- Within sum of squares → Elbow Method
- Silhouette coefficient → Maximum

Silhouette Coefficient

For each observation i, the *silhouette coef s(i)* is defined as follows:

Put a(i) = average dissimilarity between i and all other points of the cluster to which i belongs (if i is the *only* observation in its cluster, *s(i) := 0* without further calculations).

For all *other* clusters C, put *d(i,C)*= average dissimilarity of i to all observations of C.

The smallest of these *d(i,C)* is *b(i)* := min_C *d(i,C)*, and can be seen as the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does *not* belong.

Finally,

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

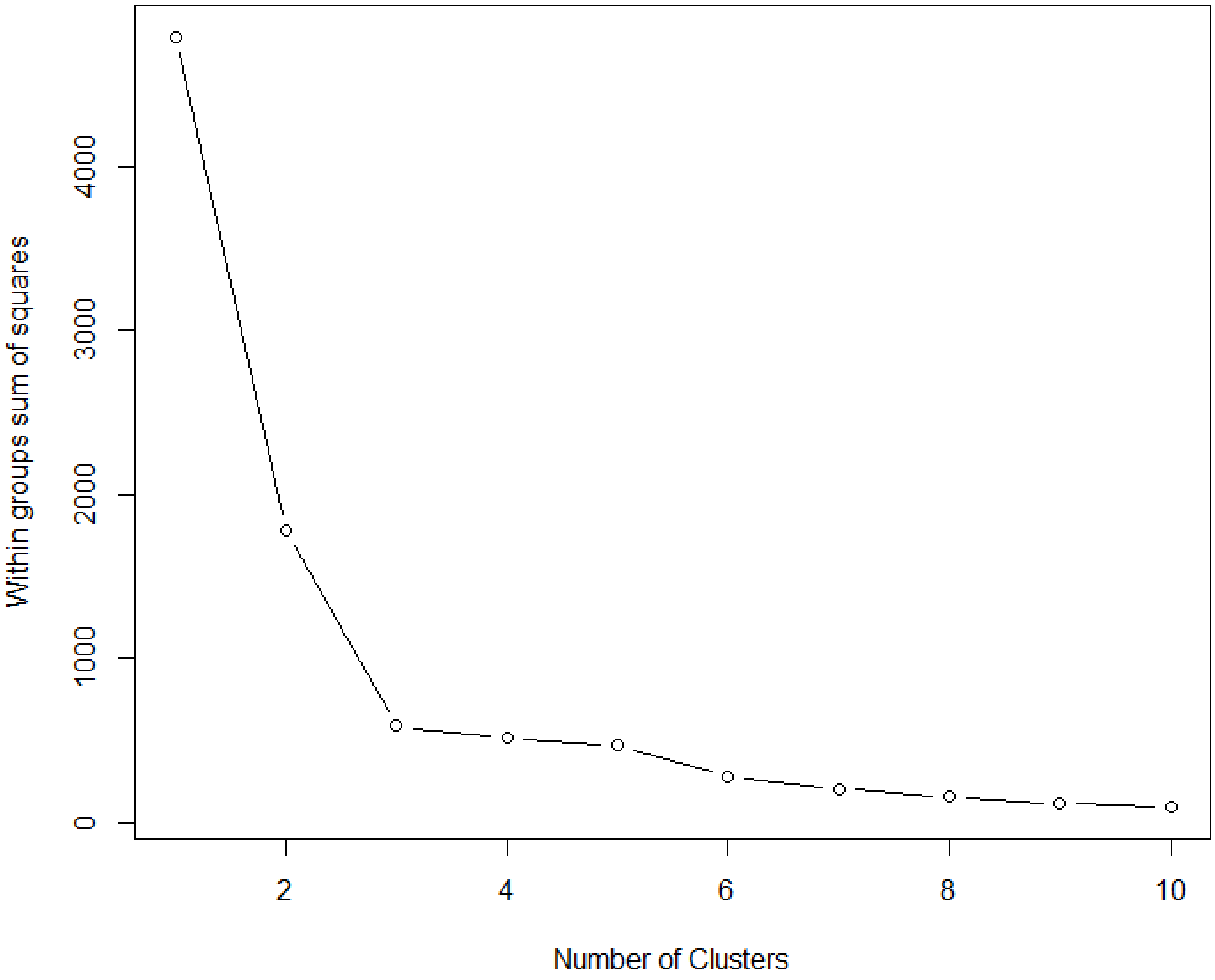
```
setwd ("D:/bagusco/Kuliah S2 --- Pemodelan Klasifikasi/data”)
data <- read.csv("ilustrasikm.csv")
```

```
hasilgerombol <- kmeans(data, centers=3, iter.max =10)
hasilgerombol$cluster

hasilgerombol$tot.withinss

wssplot <- function(dataku, nc=15, seed=1234){
  wss <- (nrow(dataku)-1)*sum(apply(dataku,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- kmeans(dataku, centers=i)$tot.withinss}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}

wssplot(data, nc=10)
```

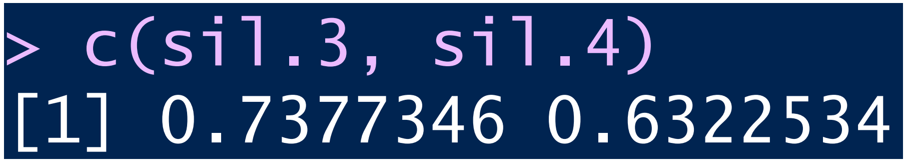

Within groups sum of squares
0
1000
2000
3000
4000
2
4
6
8
10
Number of Clusters

```
library("cluster")
jarak <- as.matrix(dist(data))

hasilgerombol <- kmeans(data, centers=3, iter.max =10)
sil.3 <-
mean(silhouette(hasilgerombol$cluster,dmatrix=jarak)[,3])

hasilgerombol <- kmeans(data, centers=4, iter.max =10)
sil.4 <-
mean(silhouette(hasilgerombol$cluster,dmatrix=jarak)[,3])

c(sil.3, sil.4)
```



```
> c(sil.3, sil.4)
[1] 0.7377346 0.6322534
```