Clustering

Max Turgeon

DATA 2010–Tools and Techniques in Data Science

Lecture Objectives

- Compare and contrast supervised and unsupervised learning.
- Fit k-means and hierarchical clustering models in R.

Motivation

- So far we have focused on problems with a clear task:
  - Predict a house price, classify a tumour.
- Sometimes we don’t have a clear target for a prediction model, or we didn’t measure it.
- Sometimes we feel like two models on two parts of the data would work best.
- **Clustering** is another class of methods to add to your toolkit, which complements prediction models.

Unsupervised vs Supervised Learning

- **Supervised learning**: There is a target variable, and we are trying to predict it.
- **Unsupervised learning**: There is no clear target variable. We want to study the structure of the data.
- Methods include:
    - *Clustering*
    - Dimension reduction
    - Score functions

Clustering i

- **Clustering** is about grouping similar observations together.
  - The output will be a discrete label for each observation, with each label corresponding to a different cluster.
- Similarity can be measured in multiple ways, but is usually based on a notion of *distance*.
  - E.g. Euclidean distance, graph distance, Manhattan distance.
- We naturally think of clusters as spherical (i.e. lying inside a ball), but they could also be nested, or linear etc.
- Clustering is inherently ill-defined.
  - The clusters I see may be different than the ones you see!

Clustering ii

- As a result, we can't really say that a clustering of the data is correct, only that it's useful.
- There are 4 main applications of clustering:
    - Hypothesis development
    - Modelling over subsets
    - Data reduction
    - Outlier detection

K-means clustering i

- **K-means clustering** starts by randomly picking K points.
  - These will be the “centres” of each cluster.
- Then each observation is assigned to the cluster corresponding to the nearest centre.
- The centres are re-calculated by taking the centroid of each cluster.
  - When using the Euclidean distance, the centroid is the sample mean.
  - In general, the centroid is not actually an observation!

K-means clustering ii

- Repeat: reassign each observation to the cluster corresponding to the nearest centre, and recompute the centroids.
- The algorithm stops when the cluster assignments stop changing.
    - There is no convergence guarantee!
- The number of clusters K is a hyper-parameter. Ideally, we want to choose K not too large, but large enough that within-cluster similarity is greater than between-cluster similarity.

Example i

```r
library(dslabs)
dataset <- brca$x

results <- kmeans(dataset, centers = 5)
# How many observations per cluster?
results$size
```

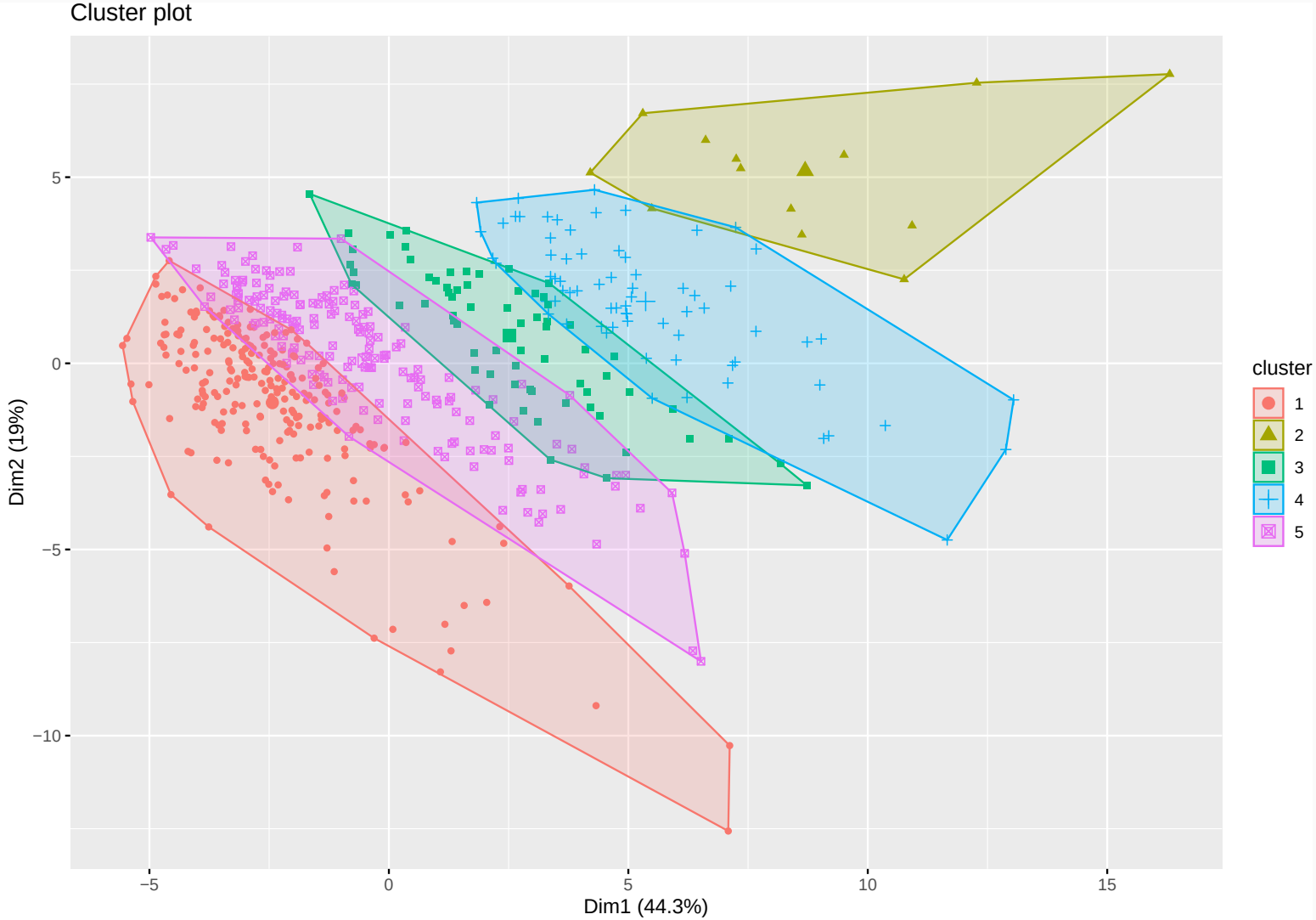
```
## [1] 237  13  69  71 179
```

Example ii

```
library(factoextra)

fviz_cluster(results, data = dataset,
             geom = "point")
```

Example iii



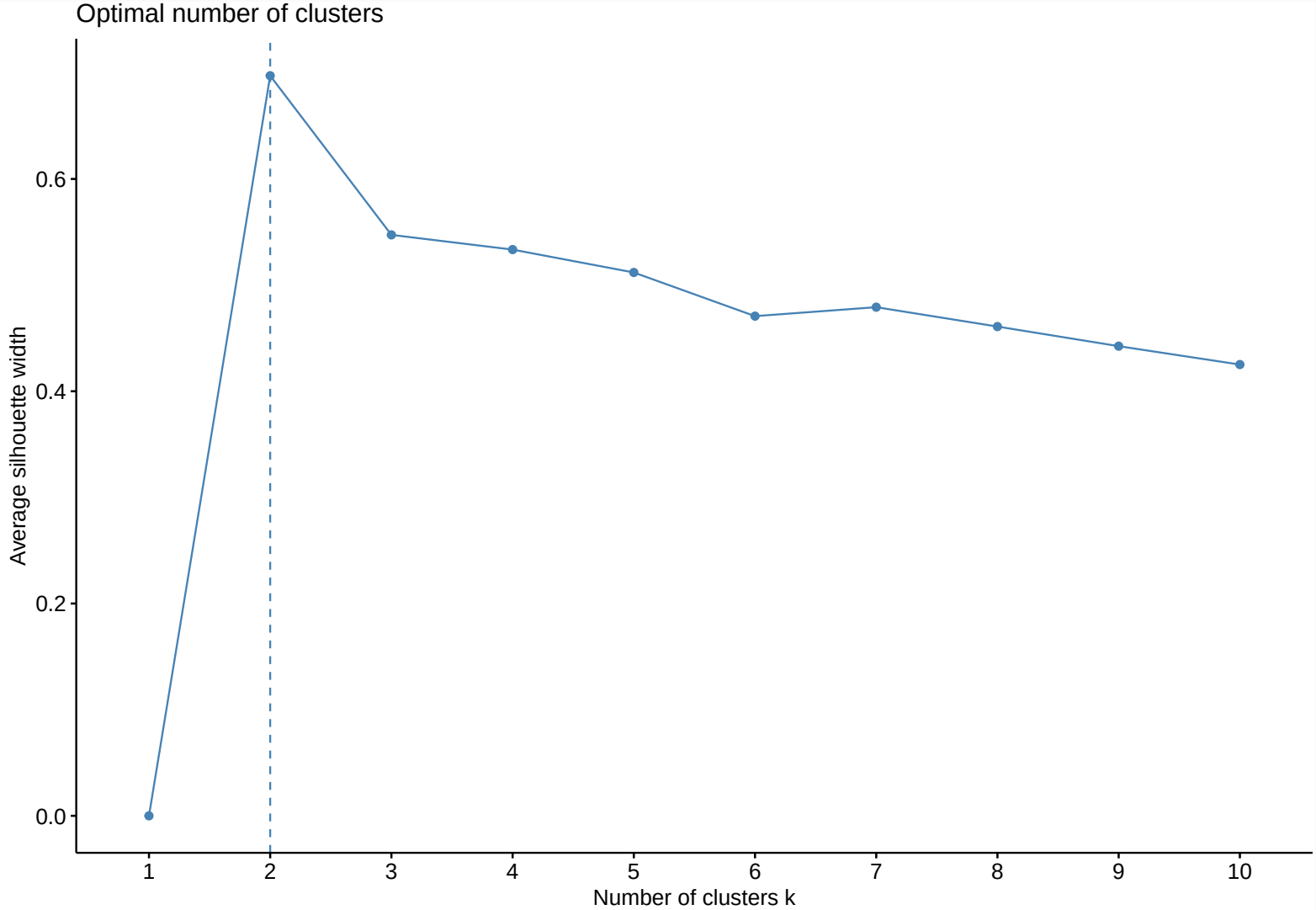
Example iv

- How many clusters should we pick? There are different approaches:
  - Elbow method
  - **Silhouette method**
  - Gap statistic
- We will use the silhouette method: the silhouette measures how well a particular observation fits within a cluster.
- We can take the average silhouette. A higher number means a better fit.

Example v

```
# This function computes the numbers for a range
# of values and then plots the results
fviz_nbclust(dataset, kmeans,
             method = "silhouette")
```

Example vi

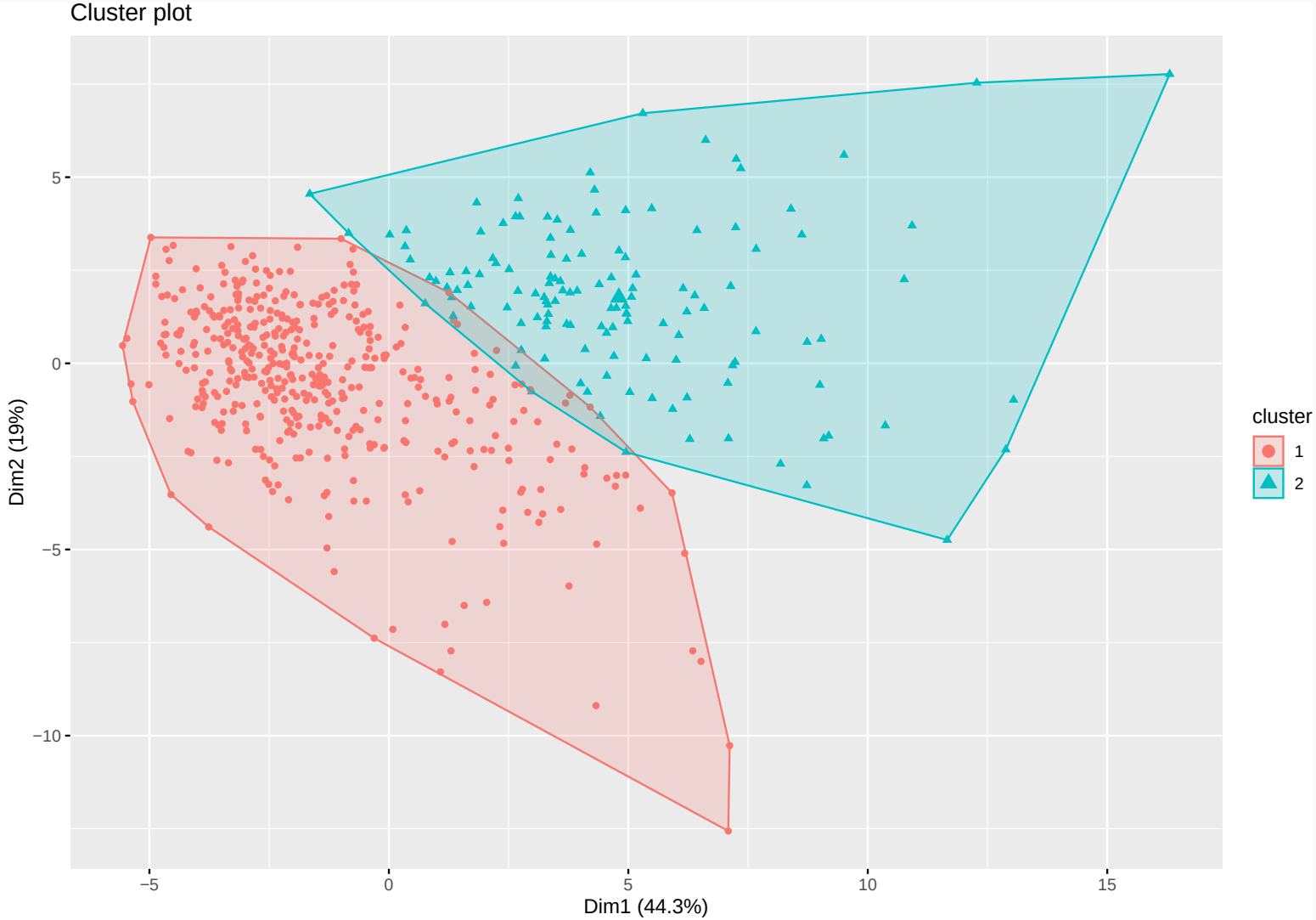

Optimal number of clusters
Average silhouette width
0.0
0.2
0.4
0.6
1
2
3
4
5
6
7
8
9
10
Number of clusters k

Example vii

- As we can see, the optimal K is 2. Let's refit K-means and visualize the new clusters.

```
results <- kmeans(dataset, centers = 2)
fviz_cluster(results, data = dataset,
             geom = "point")
```

Example viii



Exercise

The dataset `tissue_gene_expression` from the `dslabs` package is actually a list containing two slots:

- `x`: Gene expression measurements for 500 genes.
- `y`: A vector representing the tissue type of each sample.

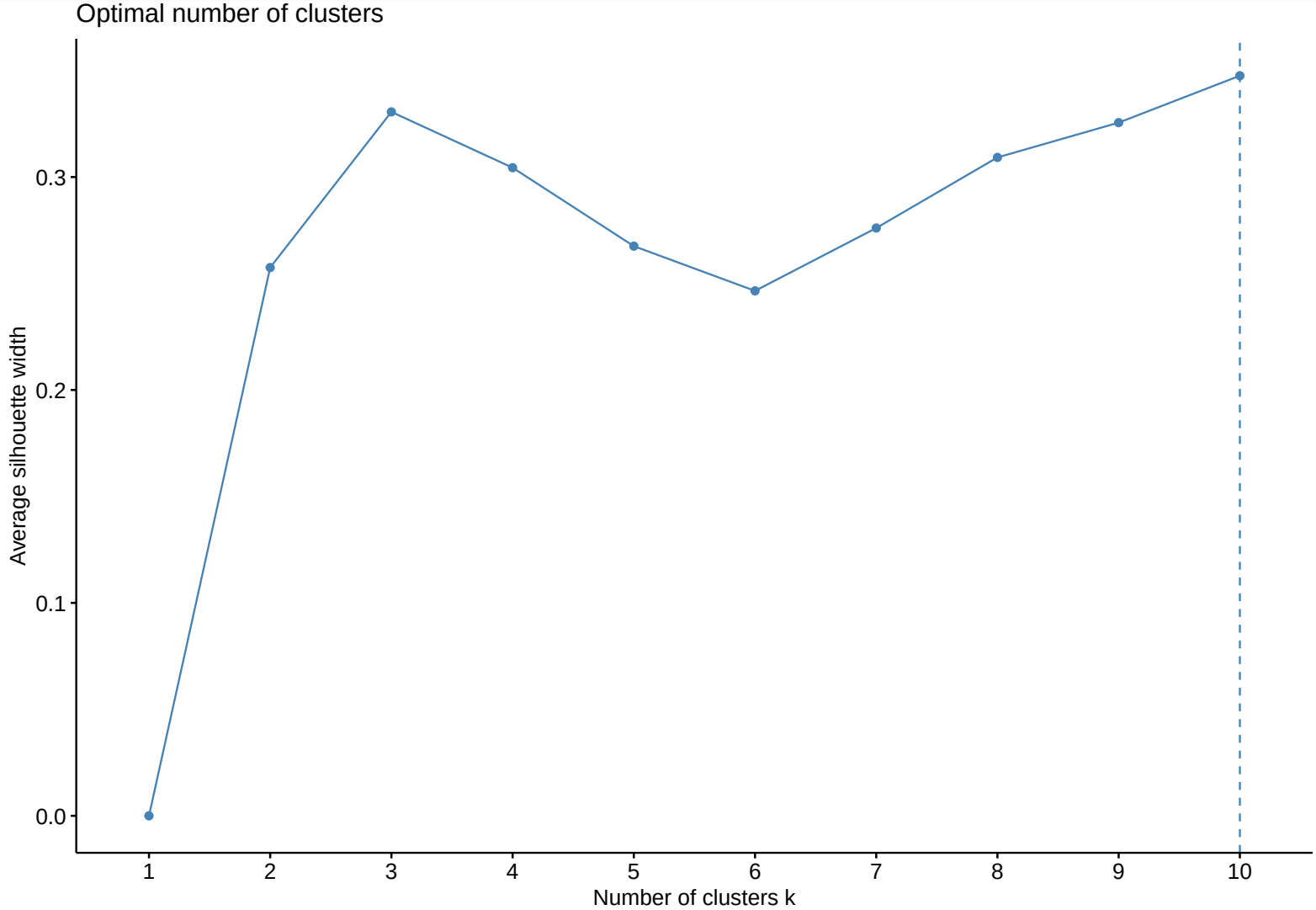
Using the data in slot `x`, find the optimal number of clusters.

Bonus: How much agreement is there between the clusters you found and tissue type?

Solution i

```
data_gene <- tissue_gene_expression$x
fviz_nbclust(data_gene, kmeans,
             method = "silhouette")
```

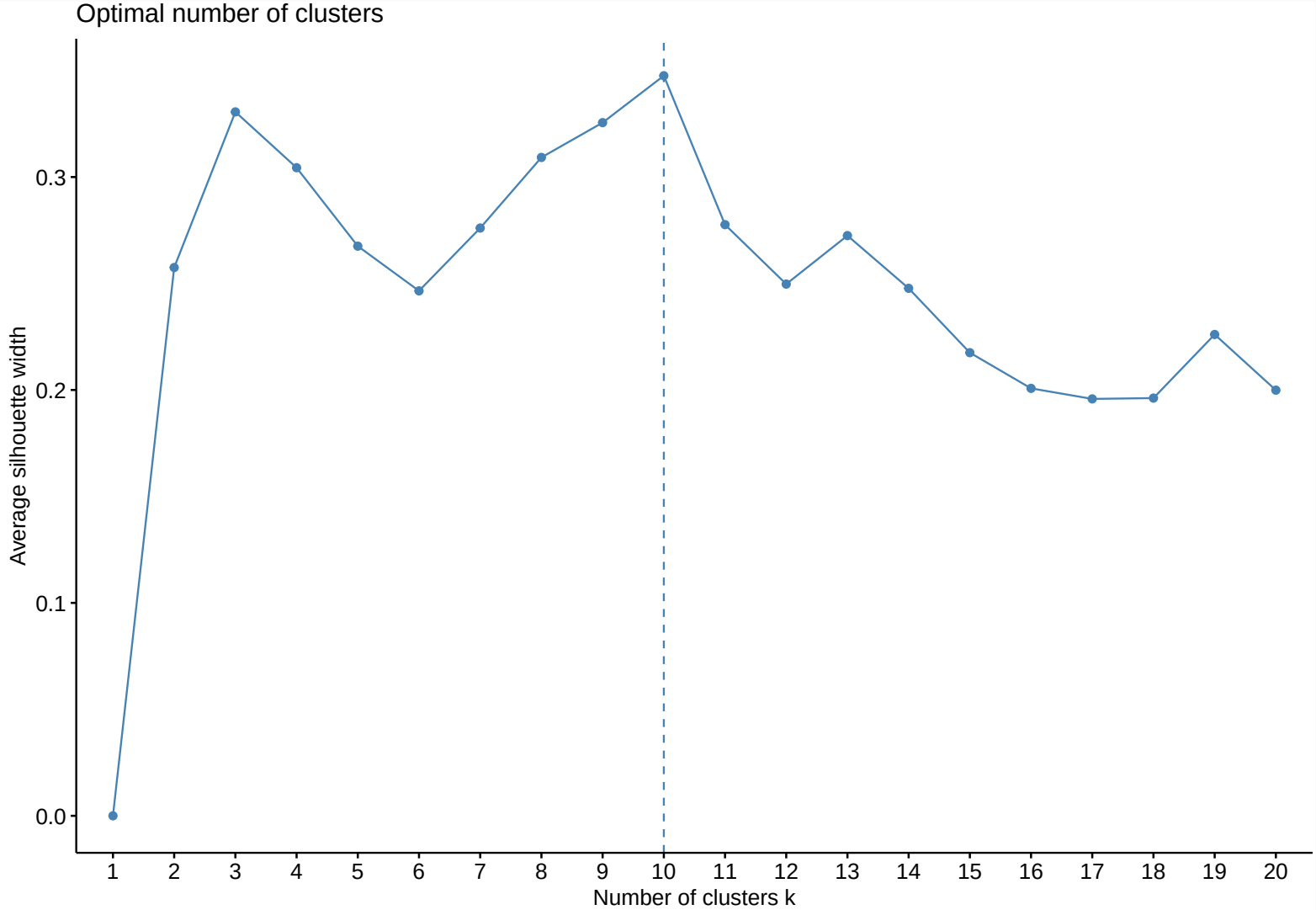
Solution ii


Optimal number of clusters
Average silhouette width
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7
8
9
10
Number of clusters k

Solution iii

```
# Since the optimal value is on the boundary
# it's a good idea to repeat with a larger k.max
fviz_nbclust(data_gene, kmeans,
             method = "silhouette",
             k.max = 20)
```

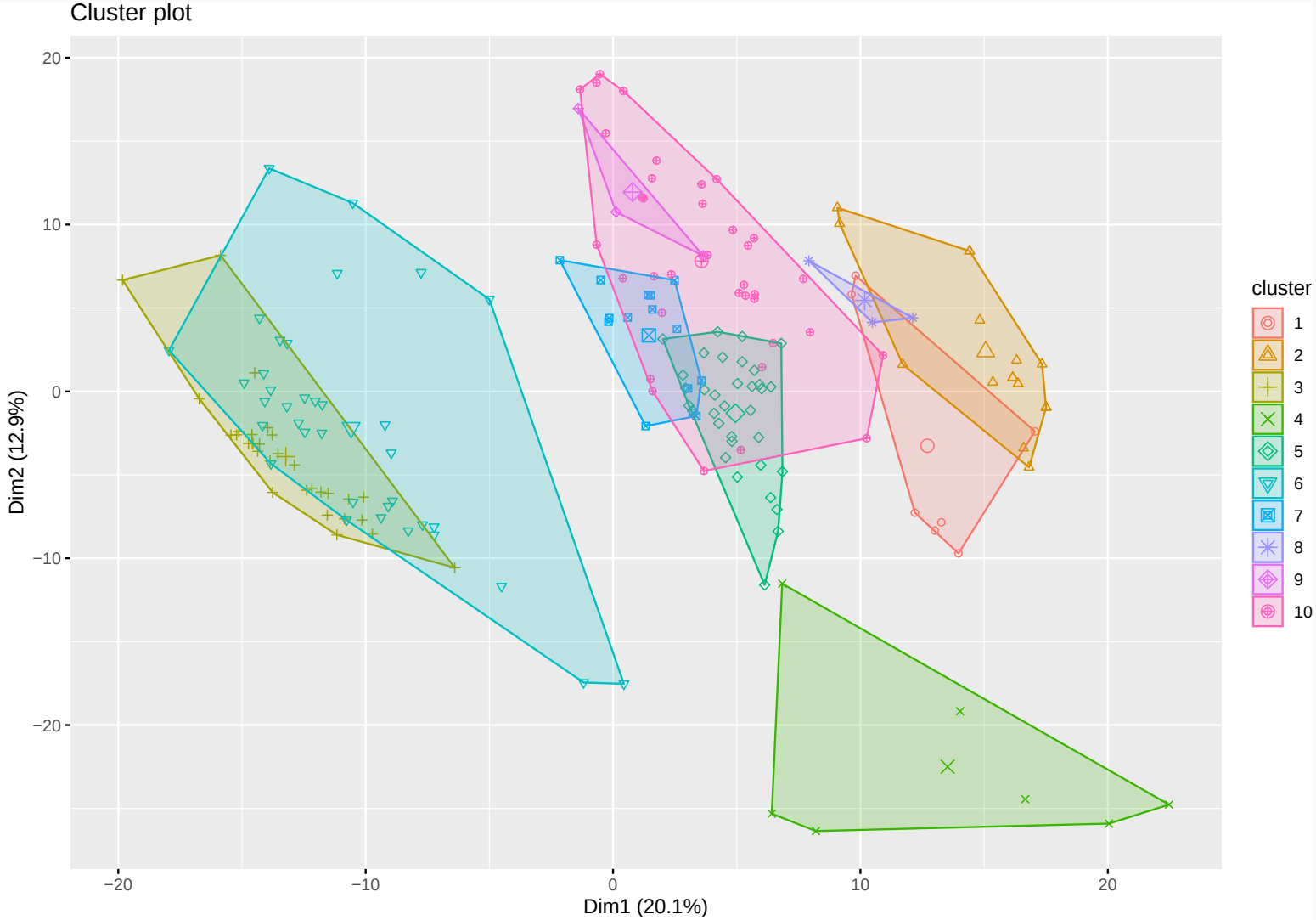
Solution iv


Optimal number of clusters
Average silhouette width
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Number of clusters k

Solution v

```
# Visualization
results <- kmeans(data_gene, centers = 10)
fviz_cluster(results, data = data_gene,
             geom = "point")
```

Solution vi


Cluster plot
Dim2 (12.9%)
Dim1 (20.1%)
20
10
0
−10
−20
−20
−10
0
10
20
cluster
1
2
3
4
5
6
7
8
9
10

Solution vii

```
# Bonus
table(results$cluster,
      tissue_gene_expression$y)
```

```
##
## cerebellum colon endometrium hippocampus
kidney liver placenta
## 1 0 0 0 0 0 7 0
## 2 0 0 0 0 0 17 0
## 3 31 0 0 0 0 0 0
## 4 2 0 0 0 3 2 0
```

Solution viii

```
## 5 0 34 0 0 0 0 0
## 6 5 0 0 31 0 0 0
## 7 0 0 15 0 0 0 0
## 8 0 0 0 0 0 0 3
## 9 0 0 0 0 0 0 3
## 10 0 0 0 0 36 0 0
```

Hierarchical clustering

- In **hierarchical** (or agglomerative) clustering, we create a hierarchy of cluster by successively merging clusters together.
- We start with every point being its own cluster.
- At each step, we identify a pair of clusters to merge.
- We stop when there is only a single cluster left.
- For a chosen K number of clusters, we "cut" the tree where $K + 1$ clusters became K clusters.
    - This gives us our clustering.

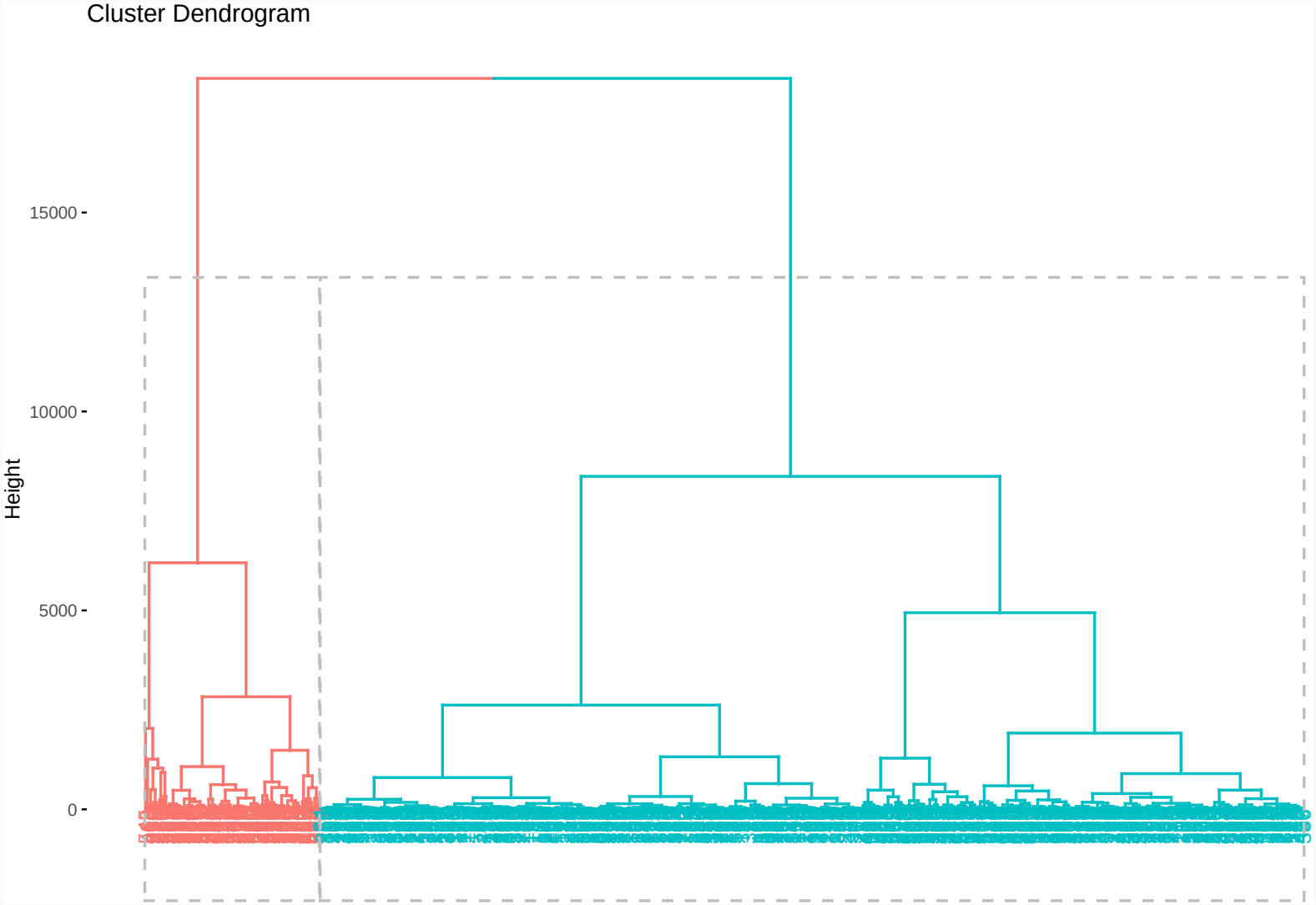
Linkage Criteria

- We can choose between many linkage criteria to determine which clusters to merge. They all essentially measure the distance between clusters, and we merge the “closest” clusters.
    - *Nearest neighbour*: minimum distance between pairs of points.
    - *Average link*: average distance between pairs of points.
    - *Nearest centroid*: compute the centroid of each cluster and measure distance between clusters as distance between centroids.
    - *Furthest link*: maximum distance between pairs of points.
- Note that for each criterion, we still need to choose a notion of distance between points
    - E.g. Euclidean, Manhattan, etc.

Example i

```
# By default, use Ward's criterion for linkage
results <- hcut(dataset, k = 2)
fviz_dend(results, rect = TRUE)
```

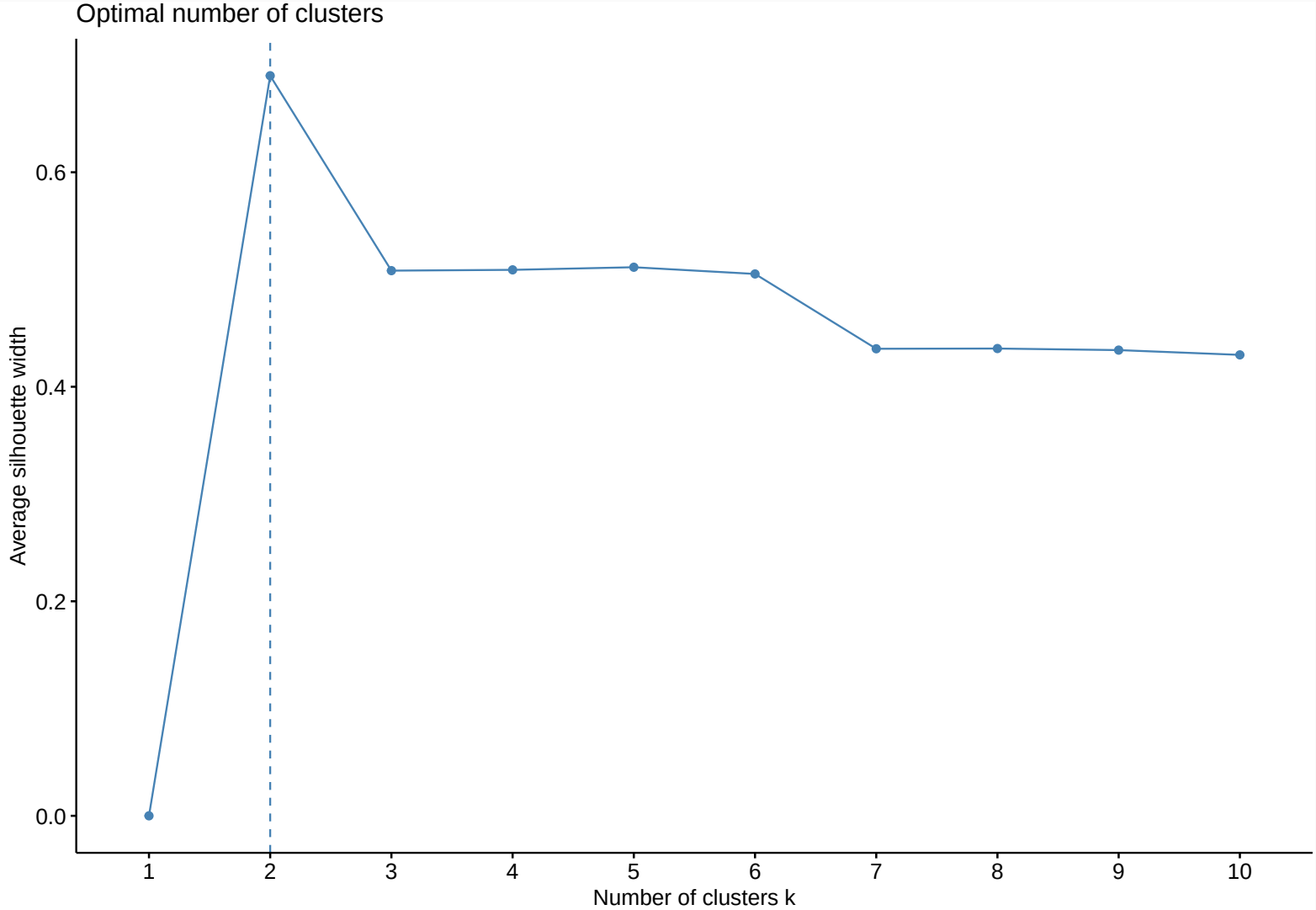
Example ii



Example iii

```
fviz_nbclust(dataset, hcut,
             method = "silhouette")
```

Example iv


Optimal number of clusters
Average silhouette width
0.0
0.2
0.4
0.6
1
2
3
4
5
6
7
8
9
10
Number of clusters k

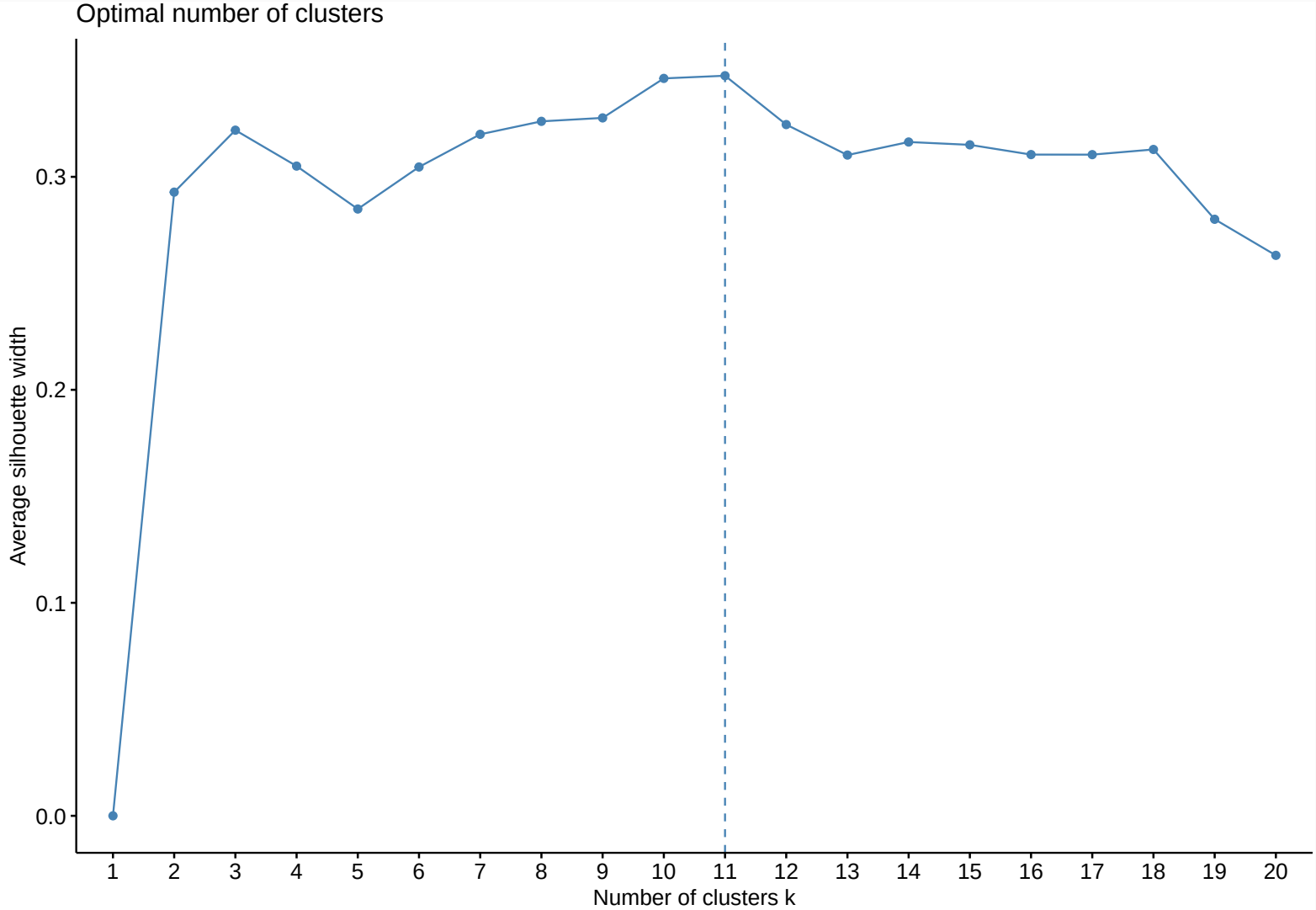
Exercise

Repeat the previous exercise using hierarchical clustering.

Solution i

```
fviz_nbclust(data_gene, hcut,
             method = "silhouette",
             k.max = 20)
```

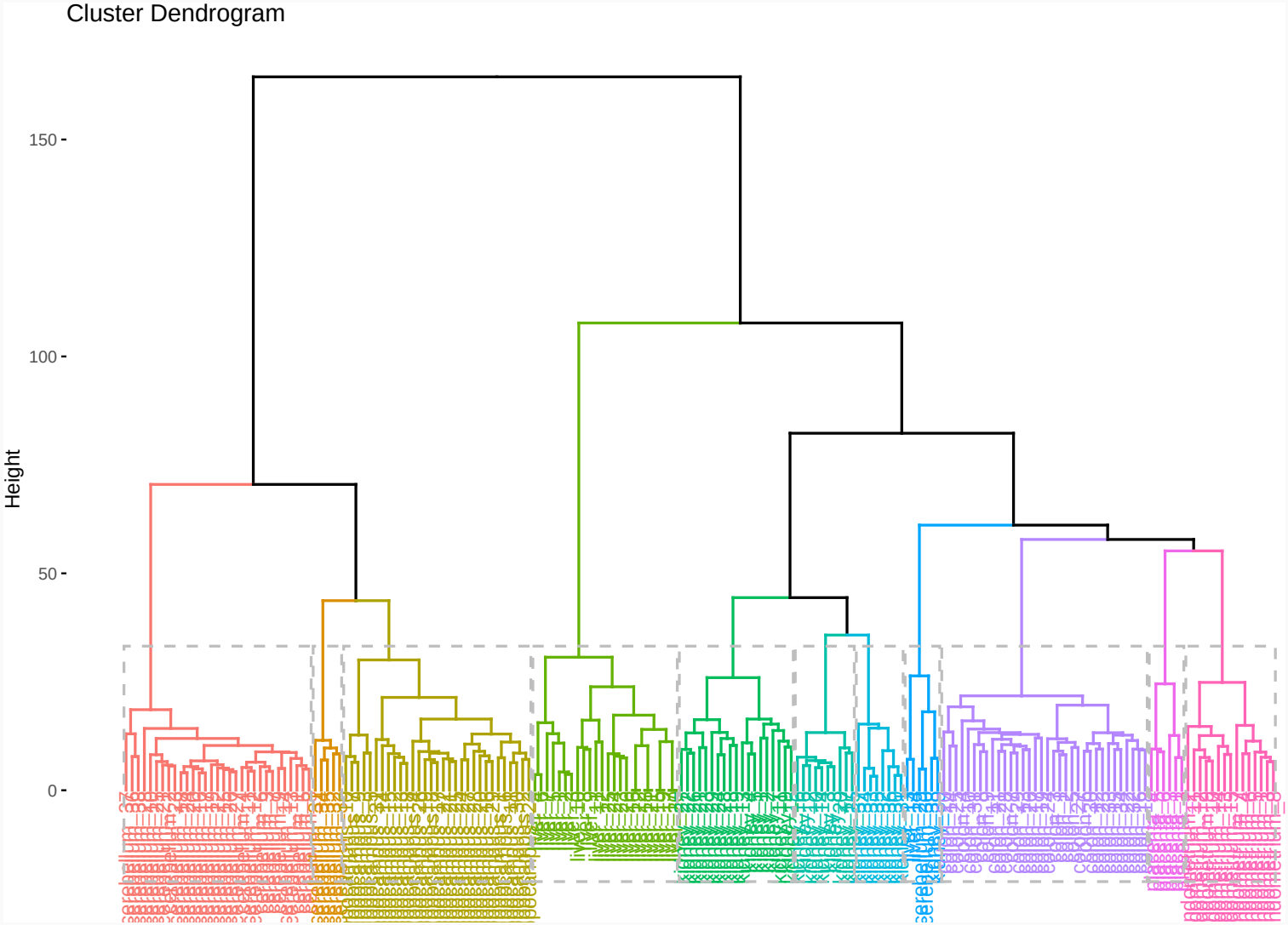
Solution ii


Optimal number of clusters
Average silhouette width
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Number of clusters k

Solution iii

```
results <- hcut(data_gene, k = 11)
fviz_dend(results, rect = TRUE)
```

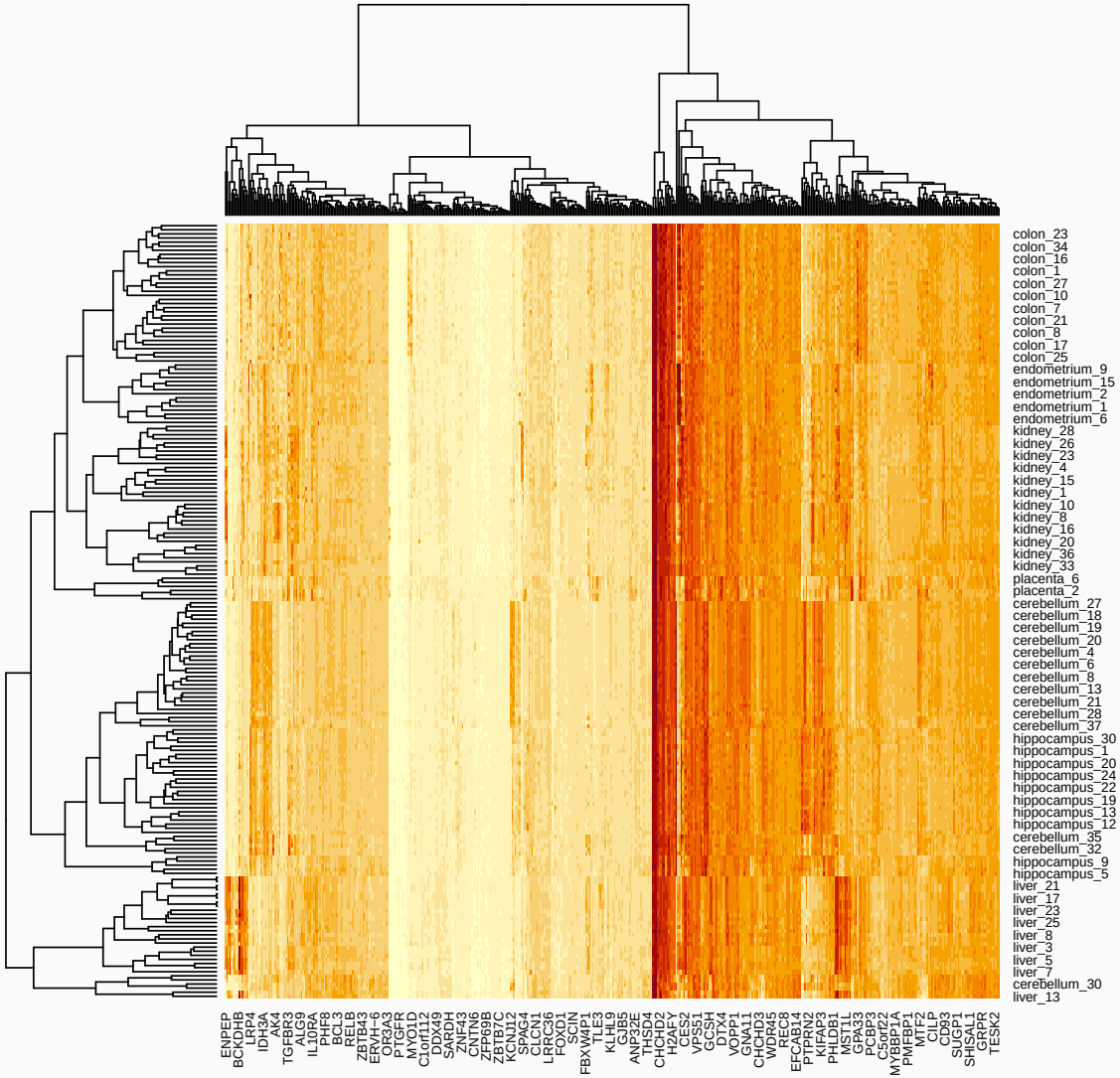
Solution iv



Solution v

```
# Gene expression data is often represented
# using a heatmap
heatmap(data_gene)
```

Solution vi


colon_23
colon_34
colon_16
colon_1
colon_27
colon_10
colon_7
colon_21
colon_8
colon_17
colon_25
endometrium_9
endometrium_15
endometrium_2
endometrium_1
endometrium_6
kidney_28
kidney_26
kidney_23
kidney_4
kidney_15
kidney_1
kidney_10
kidney_8
kidney_16
kidney_20
kidney_36
kidney_33
placenta_6
placenta_2
cerebellum_27
cerebellum_18
cerebellum_19
cerebellum_20
cerebellum_4
cerebellum_6
cerebellum_8
cerebellum_13
cerebellum_21
cerebellum_28
cerebellum_37
hippocampus_30
hippocampus_1
hippocampus_20
hippocampus_24
hippocampus_22
hippocampus_19
hippocampus_13
hippocampus_12
cerebellum_35
cerebellum_32
hippocampus_9
hippocampus_5
liver_21
liver_17
liver_23
liver_25
liver_8
liver_3
liver_5
liver_7
cerebellum_30
liver_13
ENPEP
ENDOD1
BCKDHB
LRP4
IDH3A
AK4
TGFBR3
ALG9
IL10RA
PHF8
RCL3
RELB
ZBTB43
ERVH-6
OR3A3
PTGFR
MYO1D
C1orf112
DDX49
SARDH
ZNF43
CNTN6
ZFP69B
ZBTB7C
KCNJ12
SPAG4
CLCN1
LRRC36
FOXD1
SCP1
FBXW4P1
TLE3
KLHL9
GJB5
ANP32E
THSD4
CHCHD2
H2AFY
CES2
VPS51
GCSH
DTX4
VOPP1
GNA11
CHCHD3
WDR45
REC8
EFCAB14
PTPRB
KIFAP3
PHLDB1
MST1L
GPA33
PCBP3
C5orf22
MYBBP1A
PMFBP1
MTF2
LIF
CD93
SUGP1
SHISAL1
GRPR
TESK2

Final remarks

- Clustering is often seen as exploratory.
  - We don't have a clear hypothesis.
- We can measure performance by comparing clusters.
  - Maximize within-cluster similarity, minimize between-cluster similarity.
  - E.g. Silhouette score, variance reduction.
- Other clustering methods can find non-spherical clusters.
  - E.g. DBSCAN, spectral clustering.
- Clustering is often used in conjunction with **dimension reduction**.
  - Project data into lower dimensional space.
  - Cluster the data.
- This is especially useful for *visualization*.