Exercise Sheet 3

1. The following two-dimensional data set is given. Perform a K-means clustering with $K = 3$ using the Euclidean distance. Use the first three points as initial centroids. For each algorithm iteration, specify the distances between centroids and all points and calculate the changed centroids after each reassignment of the points.

<nbsp;>	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
x	2.0	2.0	2.0	2.5	2.5	3.0	4.0	4.0	4.5	4.5	4.5	4.5
y	1.0	1.5	2.0	1.0	2.0	4.0	1.0	2.5	1.0	1.5	2.5	3.0

```
dat <- tibble(
  x = c(2.0, 2.0, 2.0, 2.5, 2.5, 3.0, 4.0, 4.0, 4.5, 4.5, 4.5 , 4.5),
  y = c(1.0, 1.5, 2.0, 1.0, 2.0, 4.0, 1.0, 2.5, 1.0, 1.5, 2.5 , 3.0)
)

k <- 3
centroids <- dat %>% slice(1:k)
new_centroids <- centroids

iteration <- 1
while(TRUE) {
  print(str_c("Iteration ", iteration))
  cat('\n')
  dist_to_centroids <- as.matrix(dist(bind_rows(new_centroids, dat)))[1:k, (k+1):(nrow(dat)+k)]
  print(knitr::kable(dist_to_centroids))
  assignments <- dist_to_centroids %>% map_dbl(~which.min(.)) %>% as_tibble() %>% mutate(row =
  print(knitr::kable(assignments))
  cat('\n')
  new_centroids <- nest(assignments, row)$data %>% map(~colMeans(dat[.$row,])) %>% do.call("rbi
  print(knitr::kable(new_centroids))
  cat('\n')

  dat_clu <- dat %>%
    bind_cols(tibble(cluster = factor(assignments$value)))


  print(ggplot(dat_clu, aes(x, y, color = cluster, fill = cluster)) +
          geom_point(pch = 16, size = 10) +
          geom_point(data = dat_clu %>%
                       group_by(cluster) %>%
                       summarize_all(mean), shape = "+", size = 10) +
          guides(fill = FALSE, color = FALSE) +
          labs(title = str_c("Iteration", iteration)) +
          theme_bw())
```

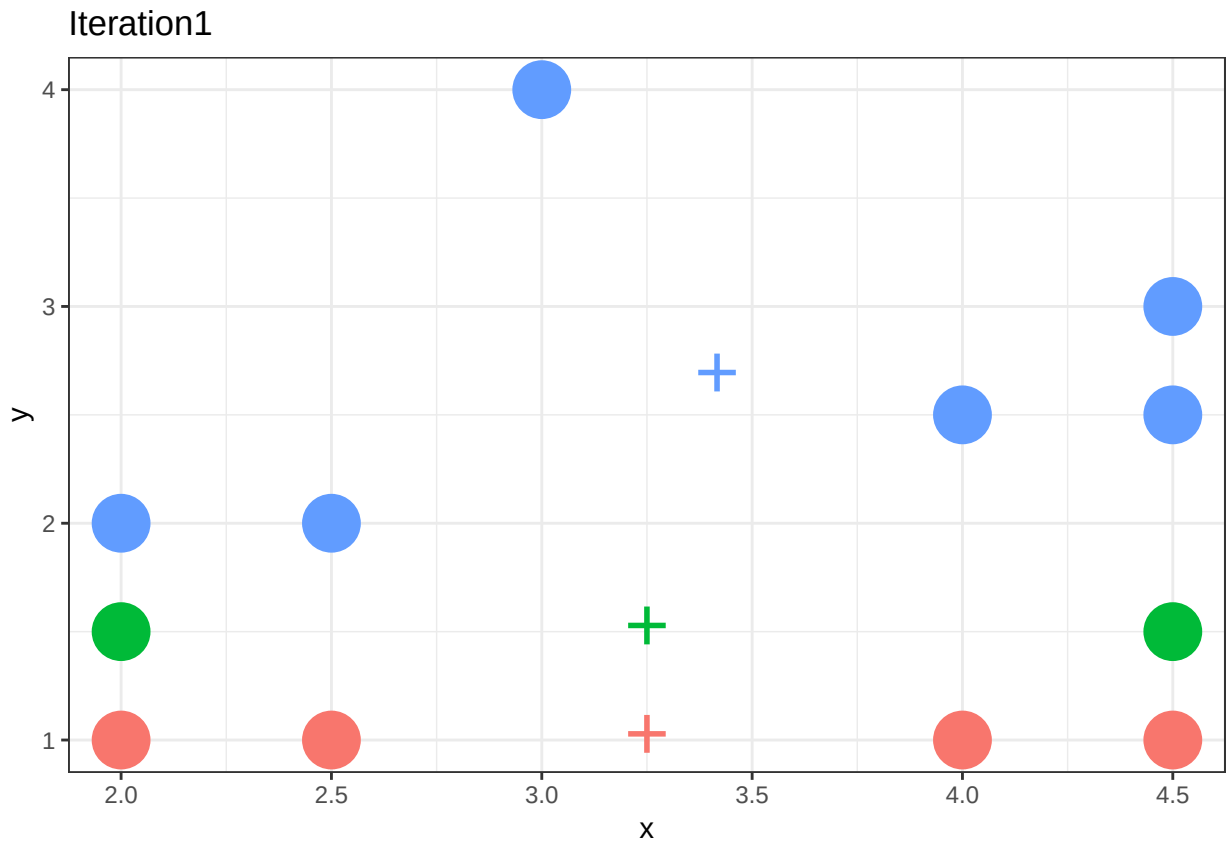
```
  dat <- tibble(
  x = c(7.25, 5.25, 2.8, 4.25, 5.1, 5.75, 2.30, 1.10, 4.0, 2.1, 3.8 , 5.9),
  y = c(1.00, 3.60, 3.8, 4.80, 3.8, 0.60, 1.65, 2.50, 4.0, 2.1, 1.6 , 1.4)
)

  print(ggplot(dat, aes(x, y)) +
          geom_point(pch = 16, size = 8) +
          guides(fill = FALSE, color = FALSE) +
          theme_bw())


  if(identical(centroids, new_centroids)) {
    break
  }
  centroids <- new_centroids
  iteration <- iteration + 1
}
```

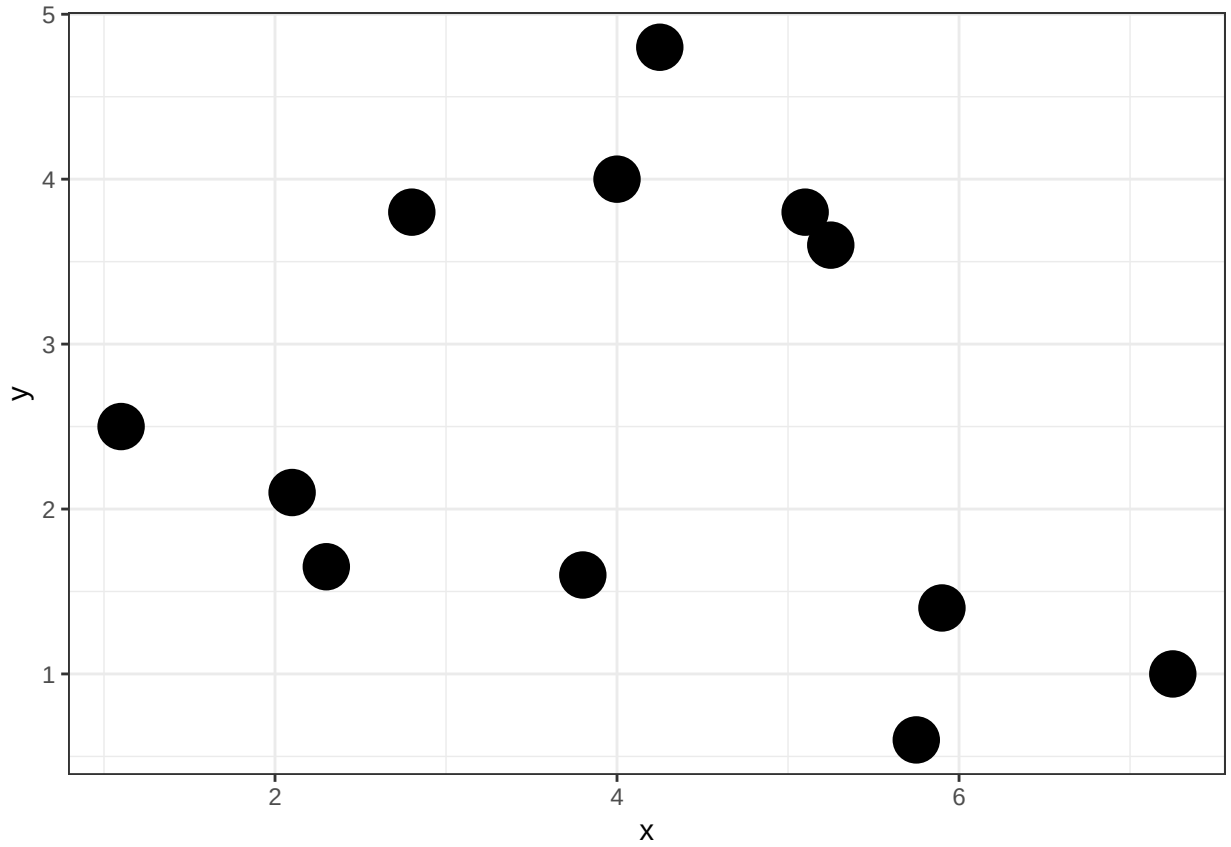
[1] "Iteration 1"

4	5	6	7	8	9	10	11	12	13	14	
0.0	0.5	1.0	0.5000000	1.1180340	3.162278	2.000000	2.500000	2.500000	2.54951	2.915476	3.2015
0.5	0.0	0.5	0.7071068	0.7071068	2.692582	2.061553	2.236068	2.549510	2.50000	2.692582	2.9154
1.0	0.5	0.0	1.1180340	0.5000000	2.236068	2.236068	2.061553	2.692582	2.54951	2.549510	2.6925

value	row
1	1
2	2
3	3
1	4
3	5
3	6
1	7
3	8
1	9
2	10
3	11
3	12

x	y
3.250000	1.000000
3.250000	1.500000
3.416667	2.666667


Iteration1
4
3
2
1
y
2.0
2.5
3.0
3.5
4.0
4.5
x

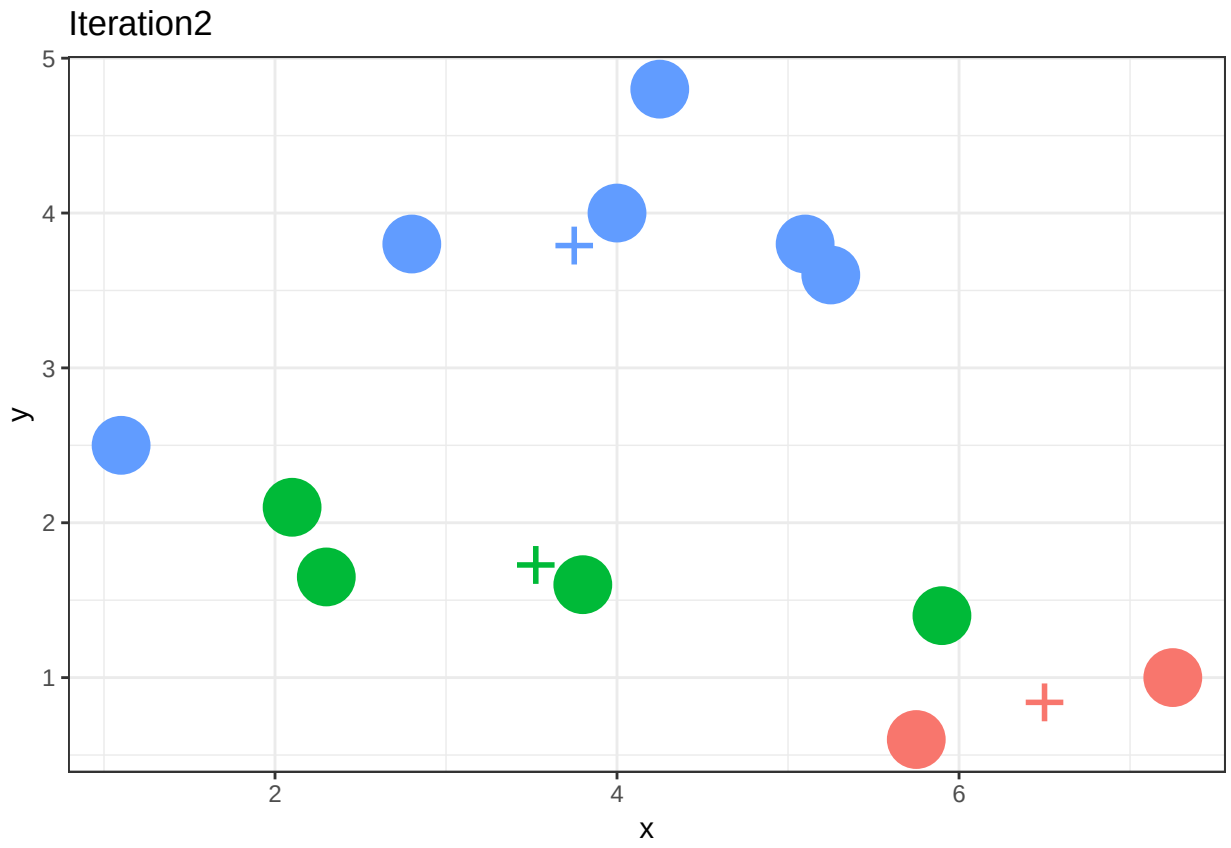


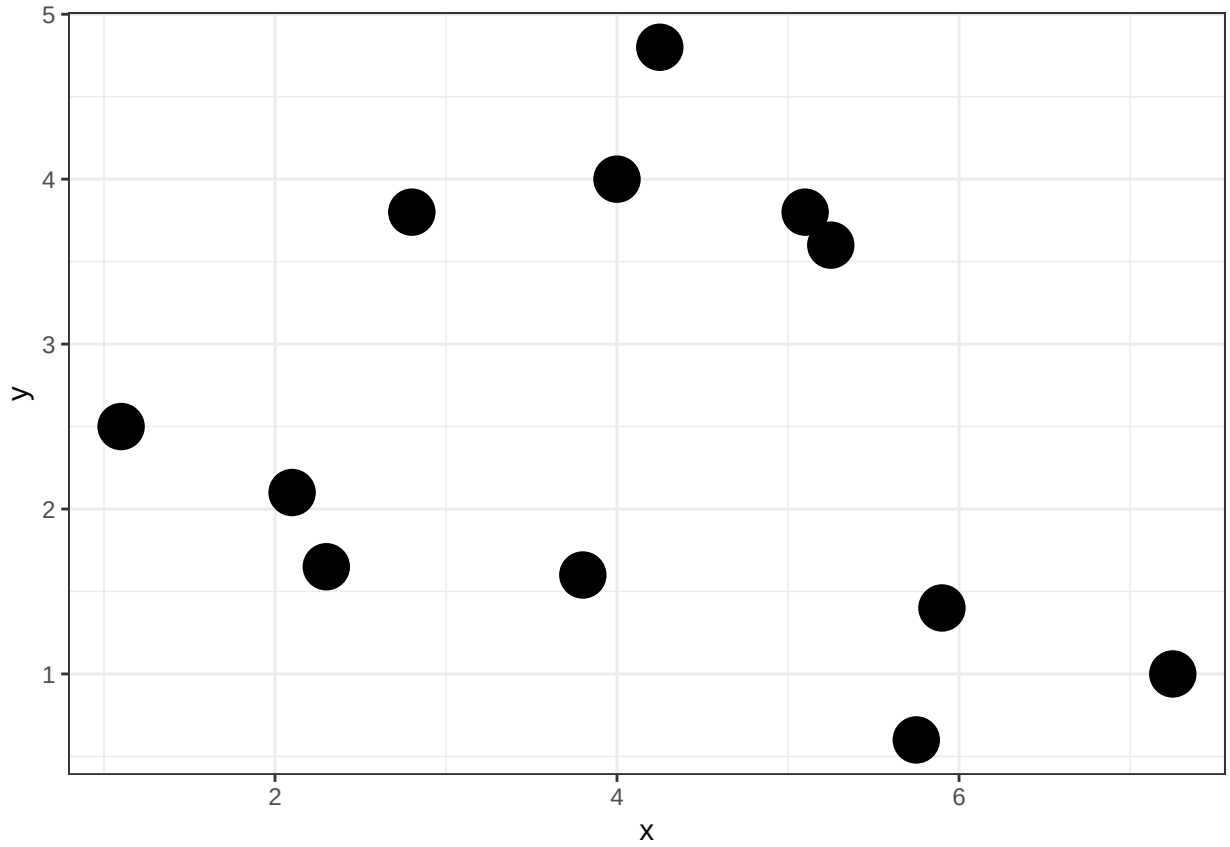
[1] “Iteration 2”

4	5	6	7	8	9	10	11	12	13	
4.000000	3.280244	2.835930	3.929377	3.355965	2.531798	1.1510864	2.621545	3.092329	1.591383	0
4.031129	2.900000	2.343608	3.448188	2.951694	2.657066	0.9617692	2.371181	2.610077	1.297112	0
4.179979	2.057237	1.290241	2.290318	2.029299	3.116979	1.5101508	2.322654	1.455354	1.433430	1

value	row
1	1
3	2
3	3
3	4
3	5
1	6
2	7
3	8
3	9
2	10
2	11
2	12

x	y
6.500	0.8000
3.750	3.7500
3.525	1.6875



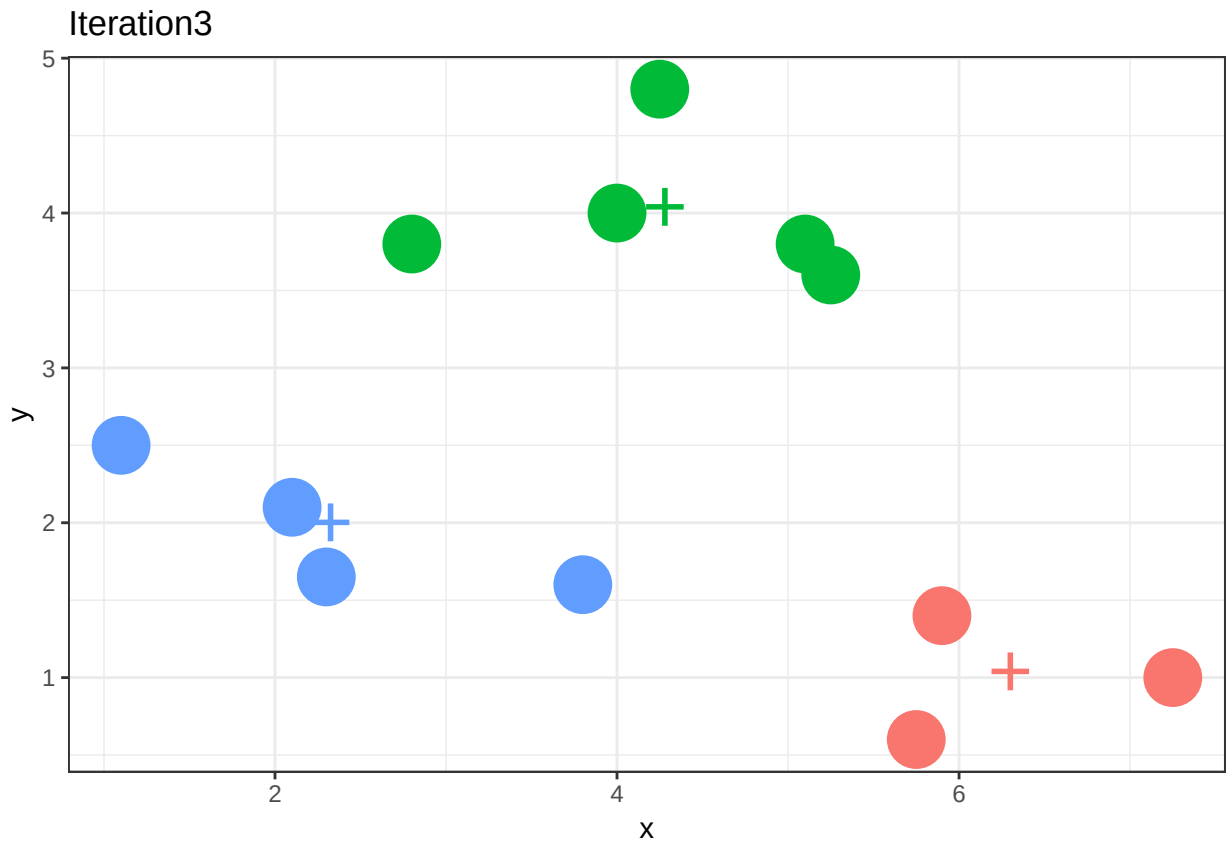


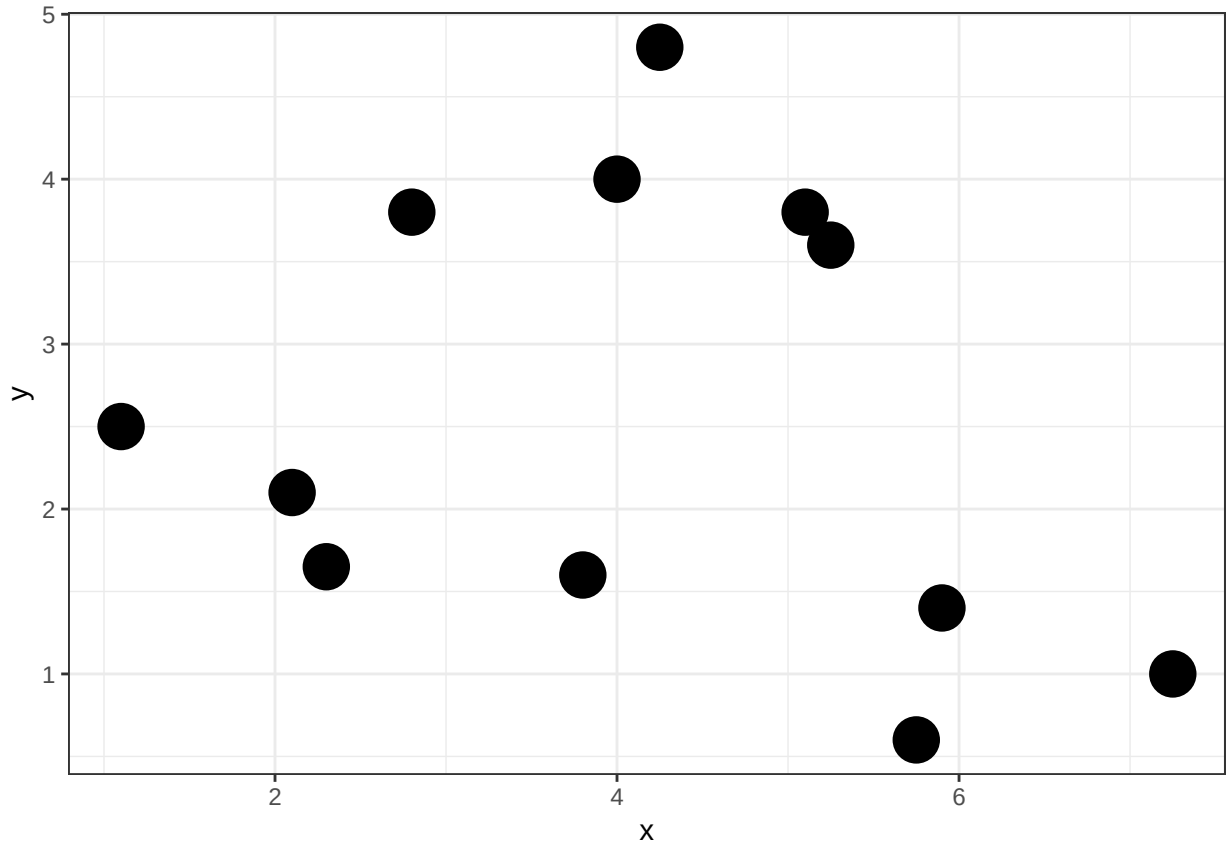
[1] “Iteration 3”

4	5	6	7	8	9	10	11	12	13
0.7762087	3.066350	4.7634021	4.589390	3.310589	0.7762087	4.285149	5.661272	4.0607881	4.588028
4.4511235	1.507481	0.9513149	1.162970	1.350926	3.7312866	2.551960	2.930017	0.3535534	2.333452
3.7879125	2.575516	2.2334460	3.195822	2.635011	2.4765462	1.225574	2.557495	2.3607798	1.483503

value	row
1	1
2	2
2	3
2	4
2	5
1	6
3	7
3	8
2	9
3	10
3	11
1	12

x	y
6.300	1.0000
4.280	4.0000
2.325	1.9625



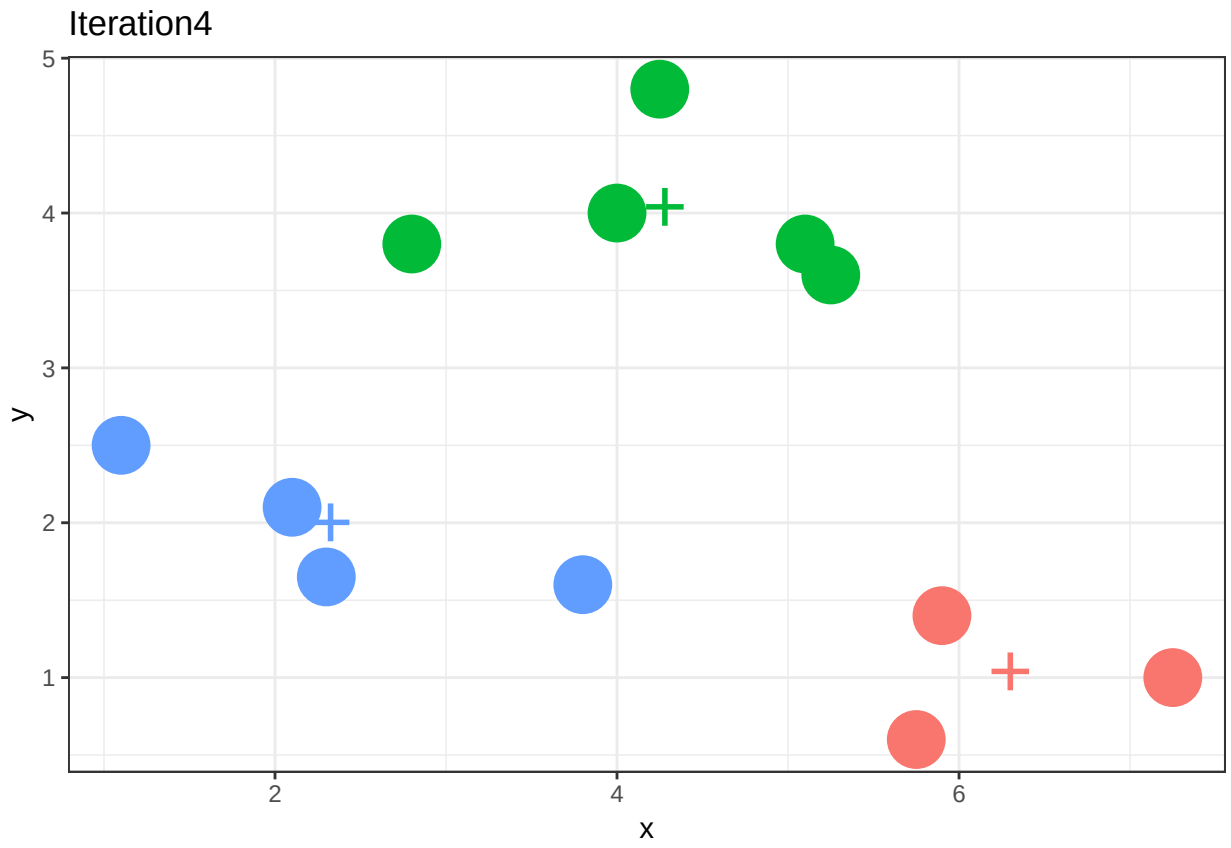


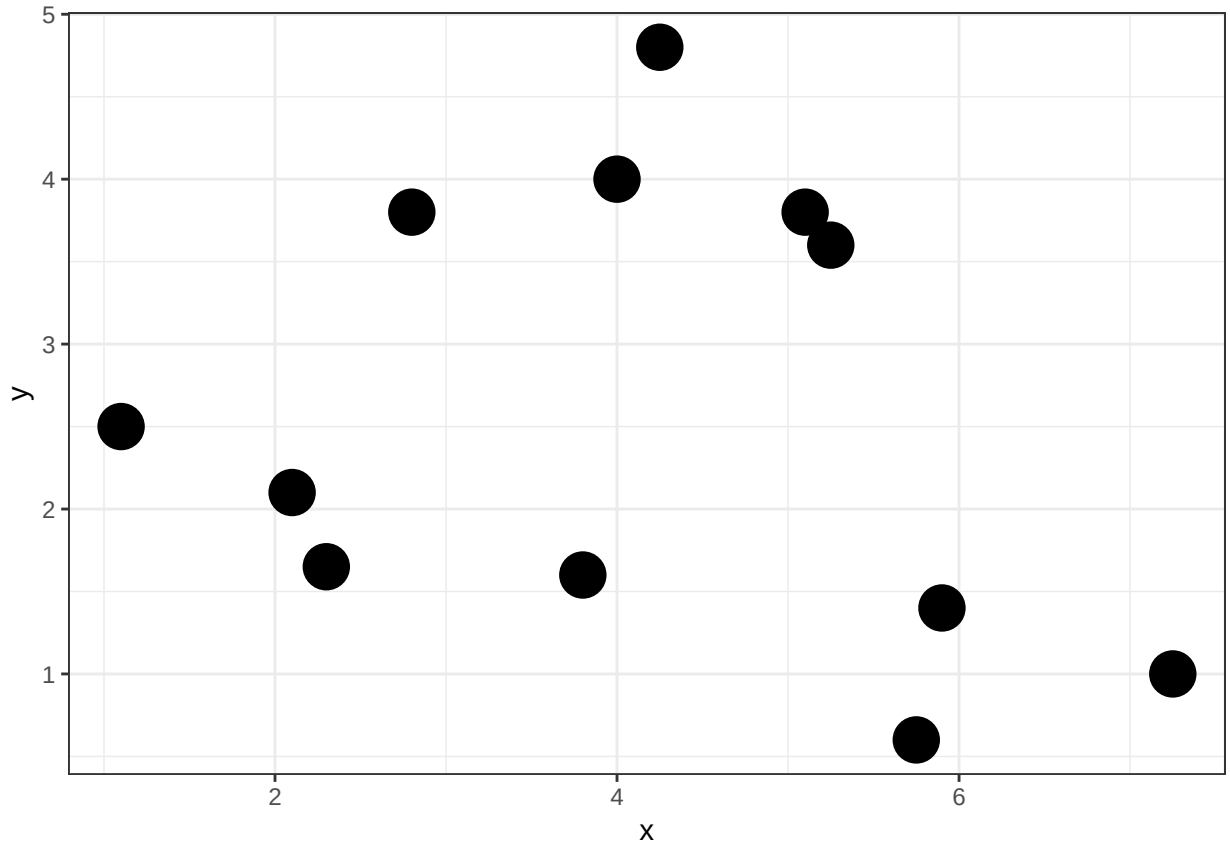
[1] “Iteration 4”

4	5	6	7	8	9	10	11	12	1
0.950000	2.804015	4.482187	4.3176961	3.0463092	0.6800735	4.0524684	5.412024	3.780212	4.341658
4.221481	1.049238	1.493452	0.8005623	0.8440379	3.7041733	3.0729302	3.516021	0.280000	2.891781
5.018170	3.352168	1.897902	3.4288528	3.3282174	3.6860590	0.3134984	1.337734	2.637618	0.263687

value	row
1	1
2	2
2	3
2	4
2	5
1	6
3	7
3	8
2	9
3	10
3	11
1	12

x	y
6.300	1.0000
4.280	4.0000
2.325	1.9625





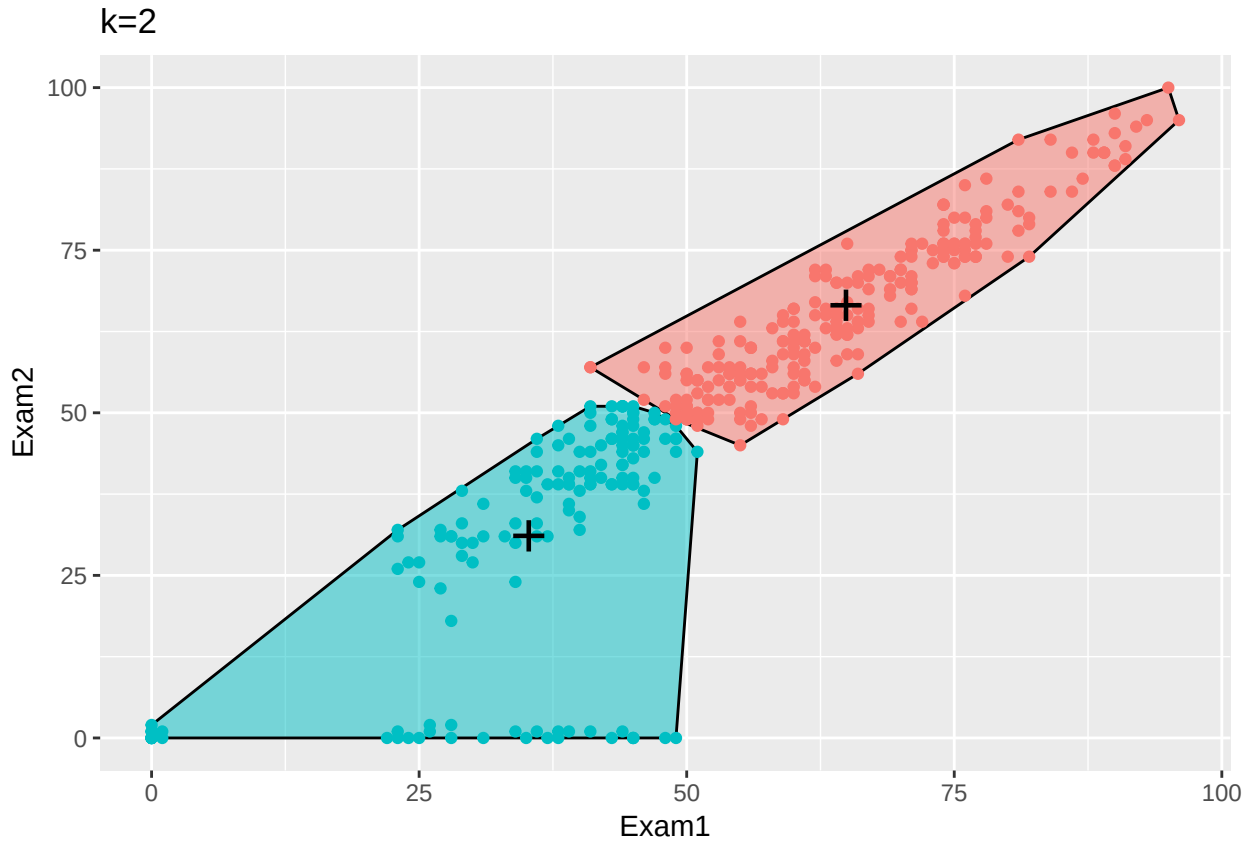

```
# plot(dat, col = kmeans(dat, centers = dat[1:3,])$cluster)
```

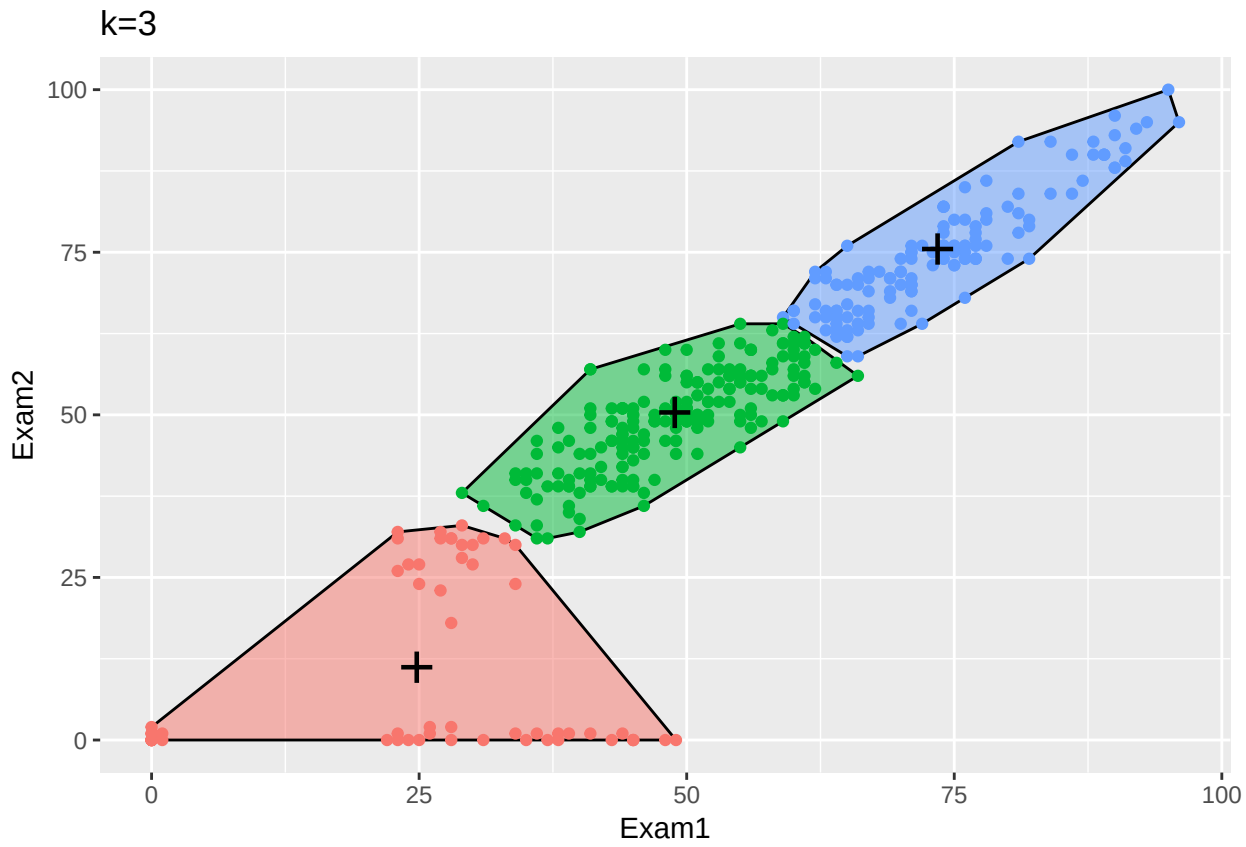
2. A school would like to group its pupils according to their performance at two intermediate examinations. It is assumed that there are at least 2 clusters of pupils. Load the file `clustering-student-mat.csv`. The file contains for each of the two exams the number of points scored for a total of 395 students. Perform a K-means-clustering for each $k \in \{2, 3, \ldots, 8\}$. Display the cluster assignments of the points in a scatter plot.

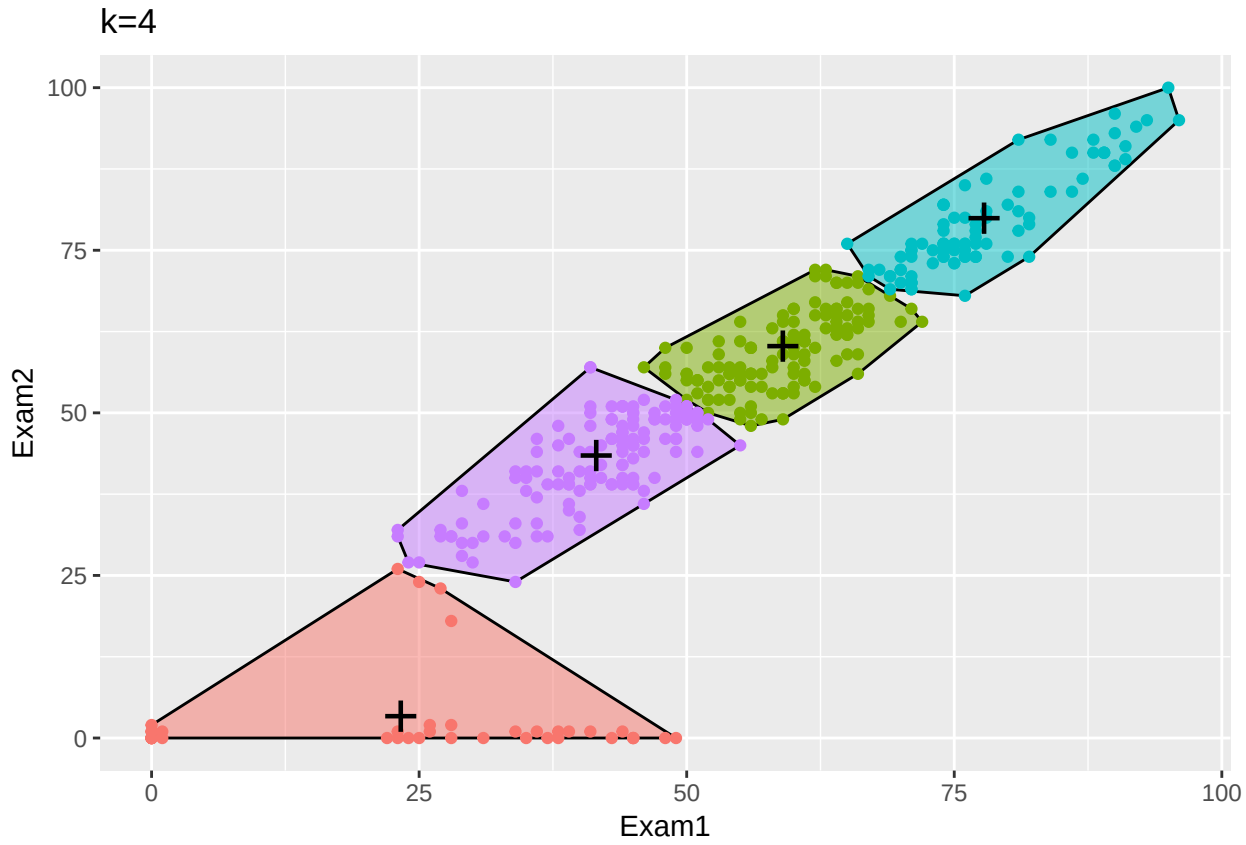
```
student <- read_csv("clustering-student-mat.csv")
```

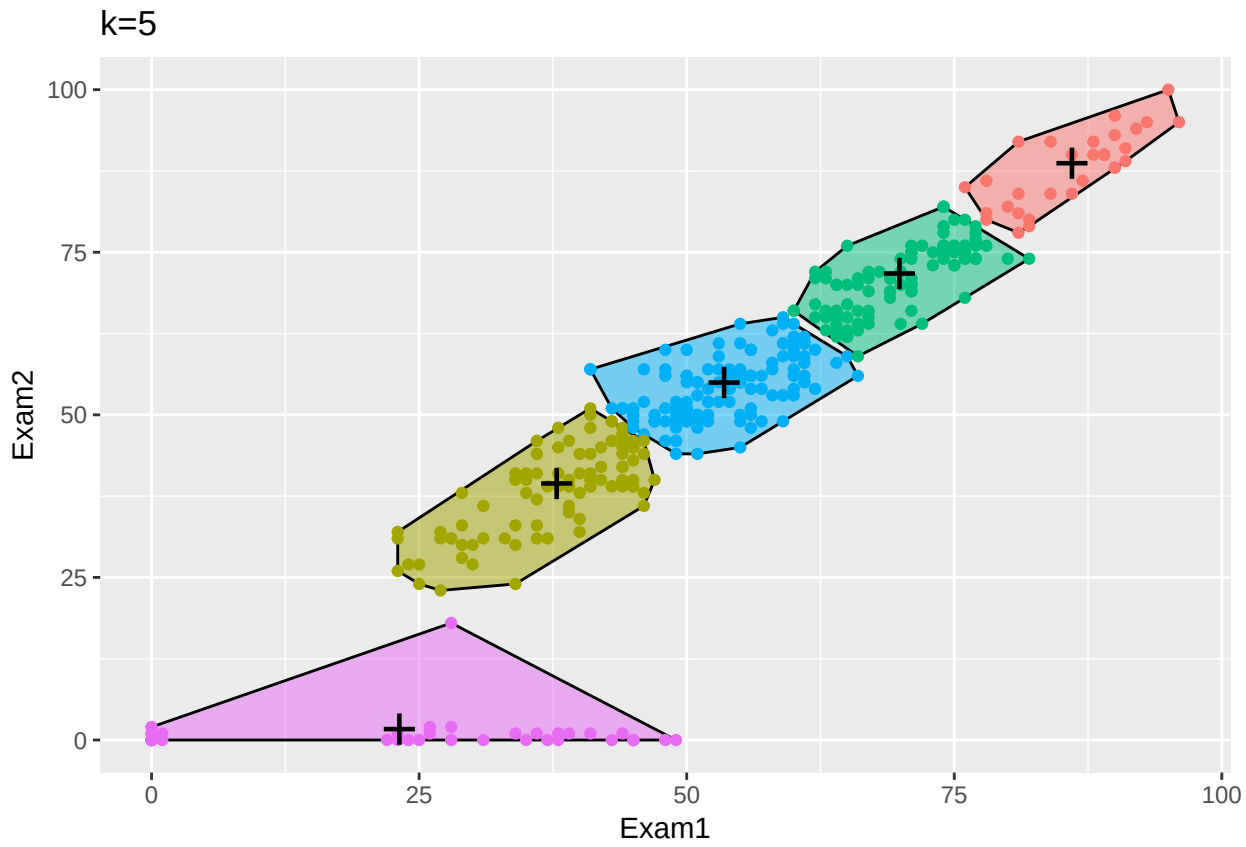
```
library(cluster)
K <- 2:8
list_clu_res <- vector("list", length(K))

list_clu_res <- map(K, ~kmeans(student %>% select(Exam1, Exam2), centers = .))

for(i in seq_along(K)) {
  student_clu <- student %>%
    bind_cols(tibble(cluster = list_clu_res[[i]]$cluster)) %>%
    mutate(cluster = factor(cluster))

  # Filter points that lie on a cluster's convex hull
  student_hull <- student_clu %>%
    split(.$cluster) %>%
```

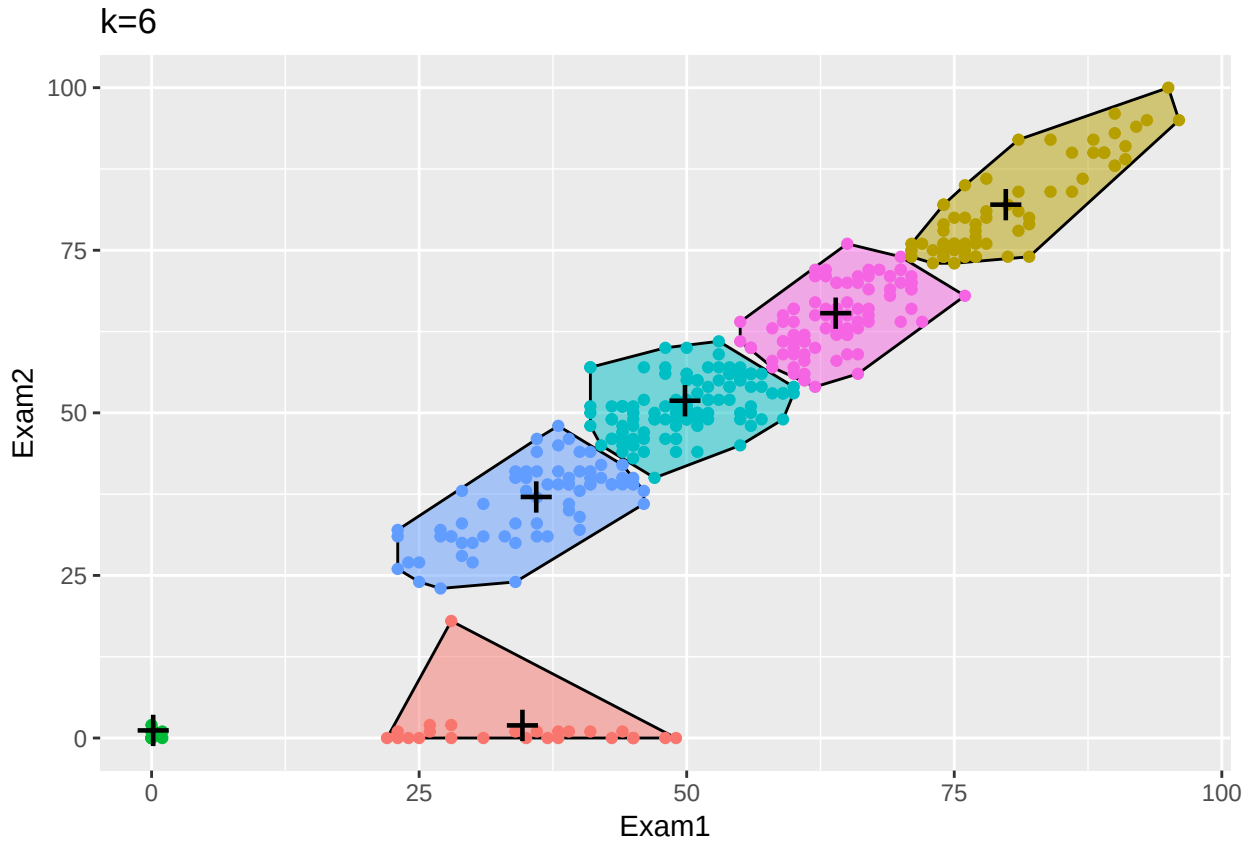
```
    map(~ slice(., chull(.$Exam1, .$Exam2))) %>%
    do.call("rbind", .)

  print(ggplot(student_clu, aes(Exam1, Exam2, color = cluster, fill = cluster)) +
          geom_polygon(data = student_hull, alpha = .5, color = "black") +
          geom_point(pch = 21) +
          geom_point(data = student_clu %>%
                       group_by(cluster) %>%
                       summarize_all(mean), shape = "+", color = "black",
                     size = 8) +
          guides(fill = FALSE, color = FALSE) +
          labs(title = str_c("k=", K[i])))
}
```

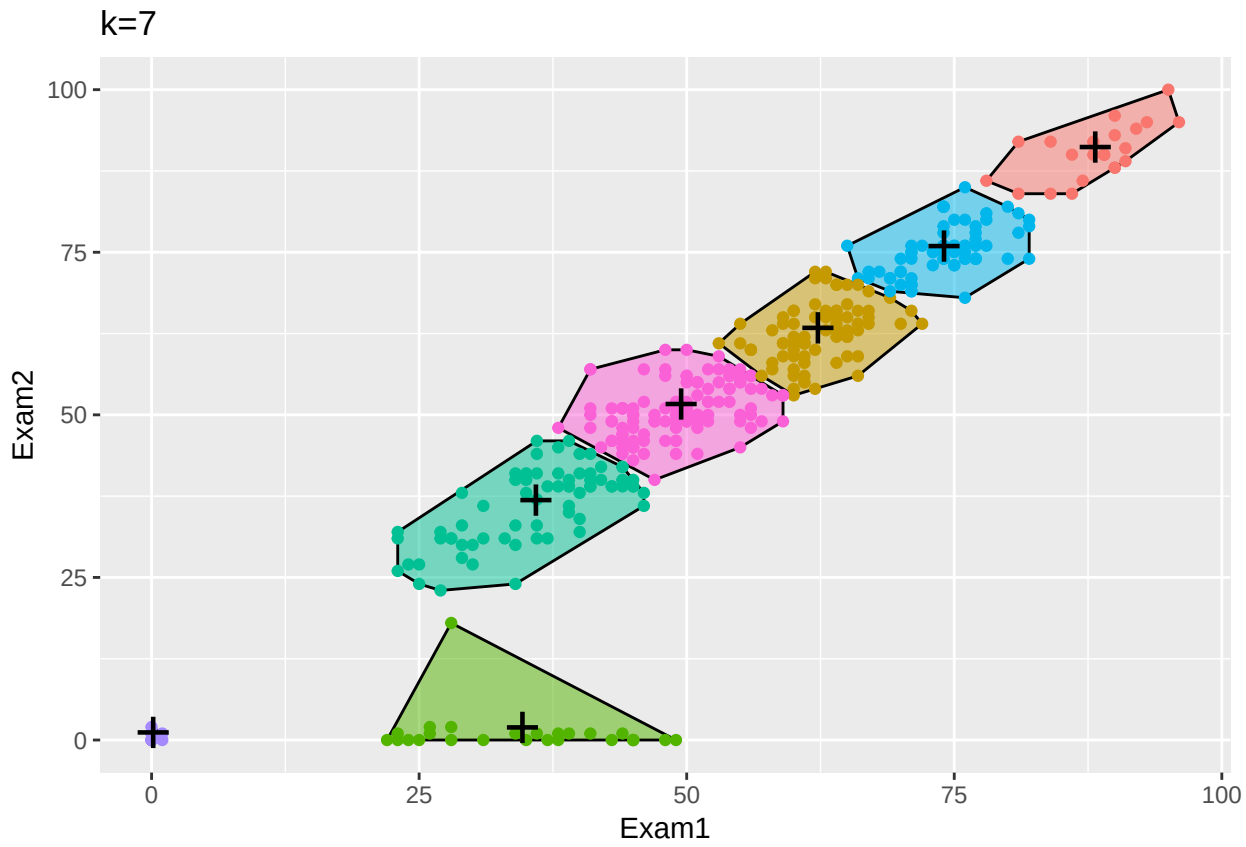


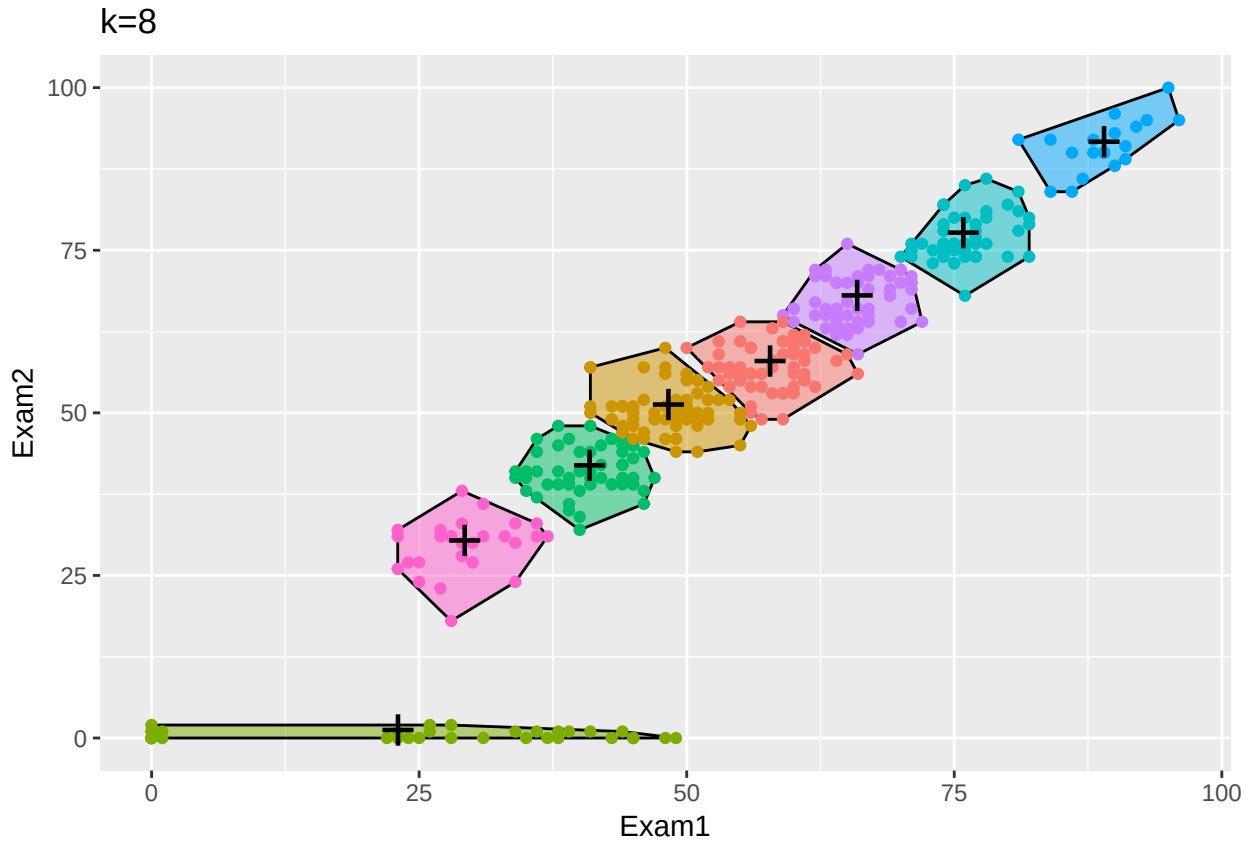

k=3
100
75
50
25
0
Exam2
0
25
50
75
100
Exam1


k=4
100
75
50
25
0
Exam2
0
25
50
75
100
Exam1


k=5
100
75
50
25
0
Exam2
0
25
50
75
100
Exam1

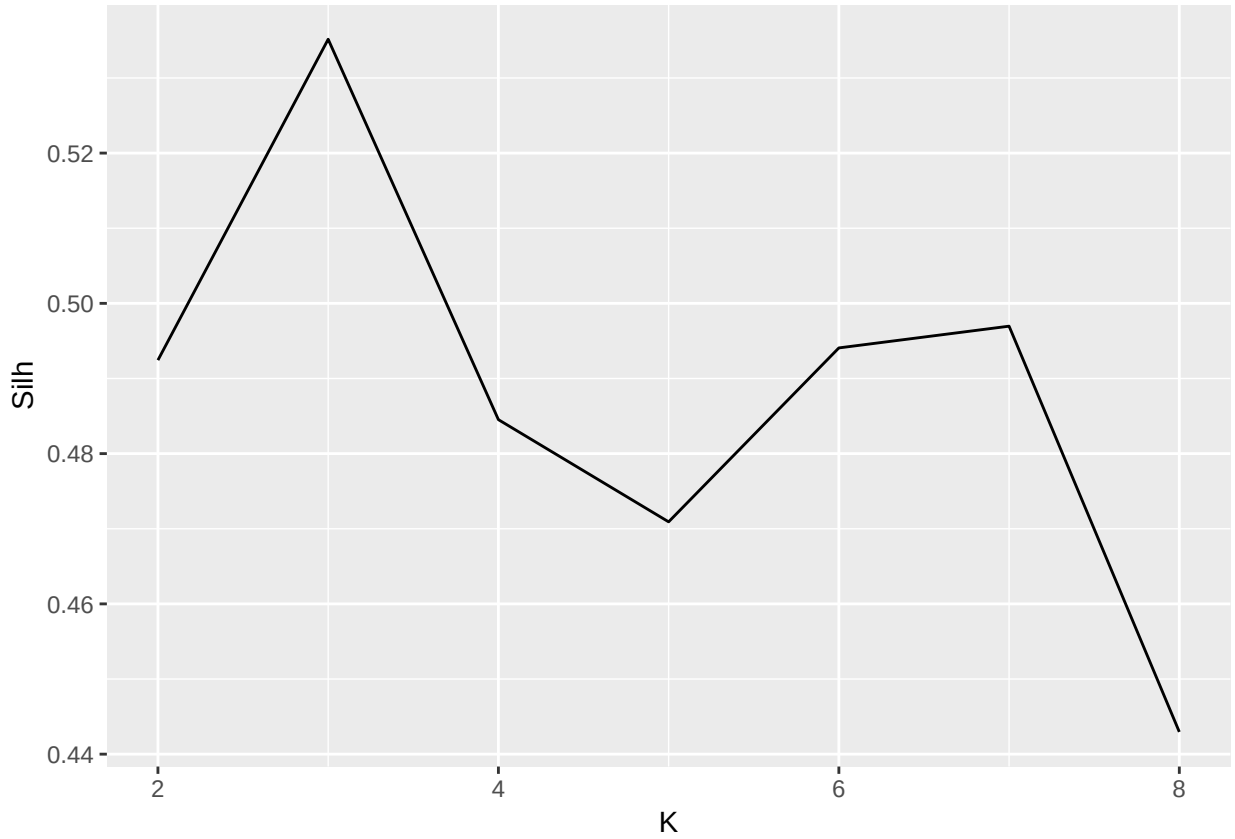

k=6
100
75
50
25
0
Exam2
0
25
50
75
100
Exam1


k=7
100
75
50
25
0
Exam2
0
25
50
75
100
Exam1



3. For the clustering in task 2, use the silhouette coefficient to find the optimal value for the number of clusters K. Evaluate the result for the representativeness of the centroids with respect to their cluster.

```
silh <- map_dbl(list_clu_res, ~mean(silhouette(.$cluster, dist(student %>% select(Exam1, Exam2)

student_silh <- tibble(K = K,
                       Silh = silh)

student_silh %>%
  ggplot(aes(K, Silh)) +
  geom_line(aes(group = factor(1)))
```




```
run_opt <- which.max(student_silh$Silh)
```

```
# k <- 6
# clu <- hclust(dist(df), "ward.D2")
# ct <- cutree(clu, k = k)
#
# df_clu <- df %>%
#   select(Exam1, Exam2) %>%
#   bind_cols(., tibble(Cluster = ct)) %>%
#   mutate(Cluster = factor(Cluster))
#
# df_hull <- df_clu  %>%
#     split(.$Cluster) %>%
#     map(~ slice(., chull(.$Exam1, .$Exam2))) %>%
#     do.call("rbind", .)
#
# ggplot(df_clu, aes(Exam1, Exam2, color = Cluster, fill = Cluster)) +
#   geom_polygon(data = df_hull %>% filter(!Cluster == "Noise"), alpha = .5, color = "black") +
#   geom_point(pch = 21) +
#   scale_fill_discrete(drop = F) +
#   scale_color_discrete(drop = F)
```

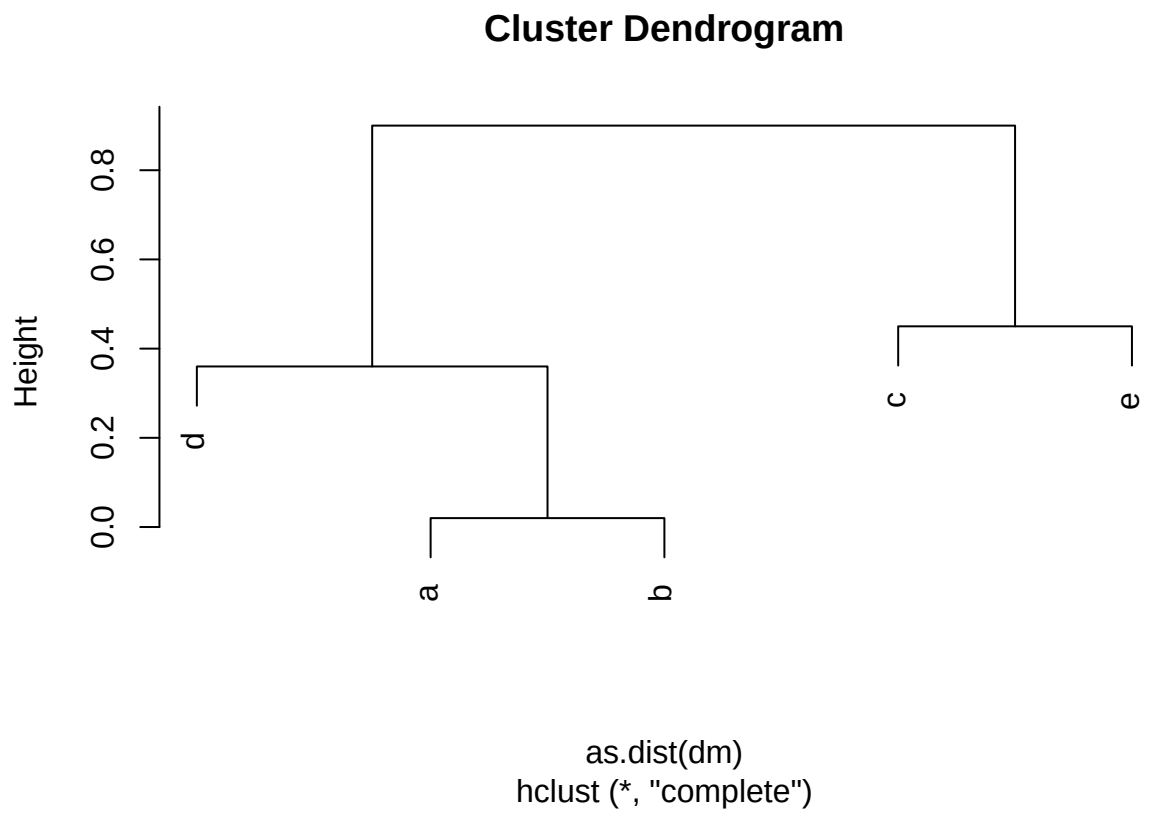
4. The following distance matrix is given. Perform agglomerative hierarchical clustering with *single* und *complete* linkage. Display the result in a dendrogram. The dendrogram should

represent the order in which the points are joined.

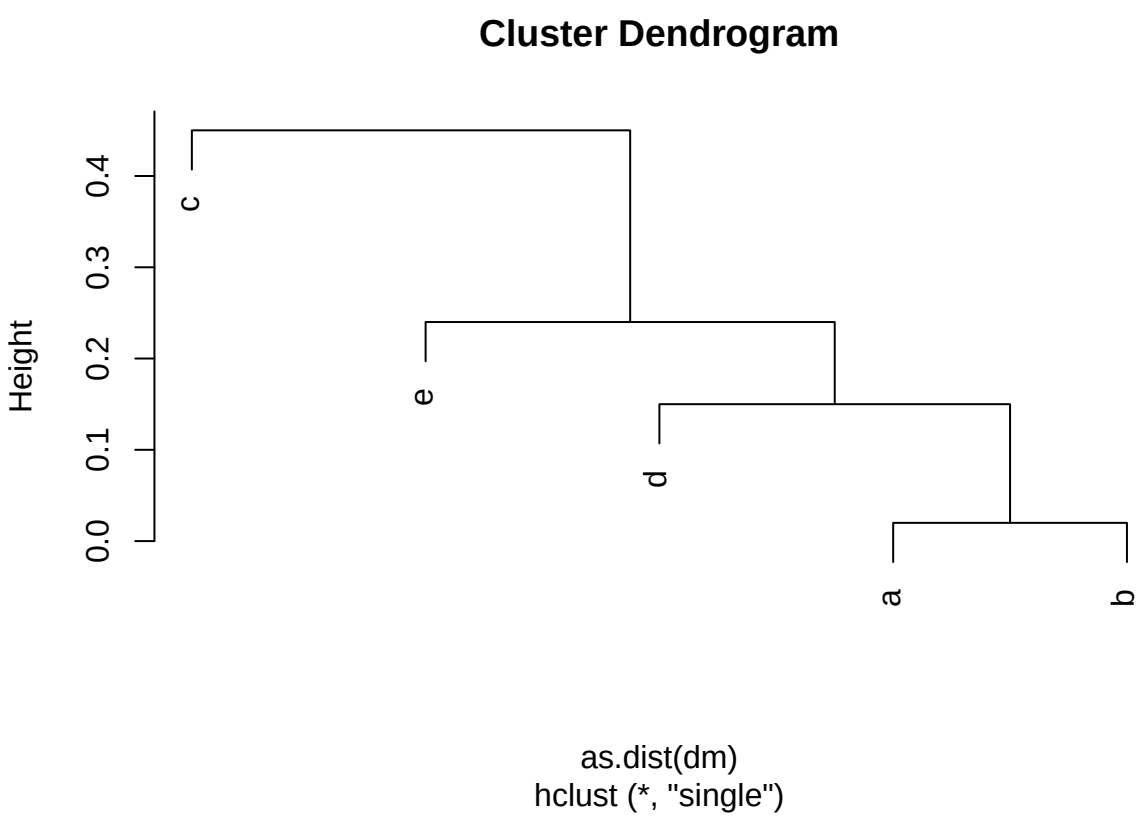
```
# S.563 Tan Steinbach Kumar
dm <- tribble(~p1,~p2,~p3,~p4,~p5,
              0.00, 0.02, 0.90, 0.36, 0.53,
              0.02, 0.00, 0.65, 0.15, 0.24,
              0.90, 0.65, 0.00, 0.59, 0.45,
              0.36, 0.15, 0.59, 0.00, 0.56,
              0.53, 0.24, 0.45, 0.56, 0.00) %>% as.matrix()
rownames(dm) <- letters[1:5]
colnames(dm) <- letters[1:5]
knitr::kable(dm)
```

	a	b	c	d	e
a	0.00	0.02	0.90	0.36	0.53
b	0.02	0.00	0.65	0.15	0.24
c	0.90	0.65	0.00	0.59	0.45
d	0.36	0.15	0.59	0.00	0.56
e	0.53	0.24	0.45	0.56	0.00

```
plot(hclust(as.dist(dm), "complete"))
```




```
plot(hclust(as.dist(dm), "single"))
```



Dataset for task 2: http://isgwww.cs.uni-magdeburg.de/cv/lehre/VisAnalytics/material/exercise/datasets/clustering-student-mat.csv