به نام خدا

تمرین سوّم

مبانی داده‌کاوی

انواع روش‌های خوشه‌بندی و ارزیابی داخلی

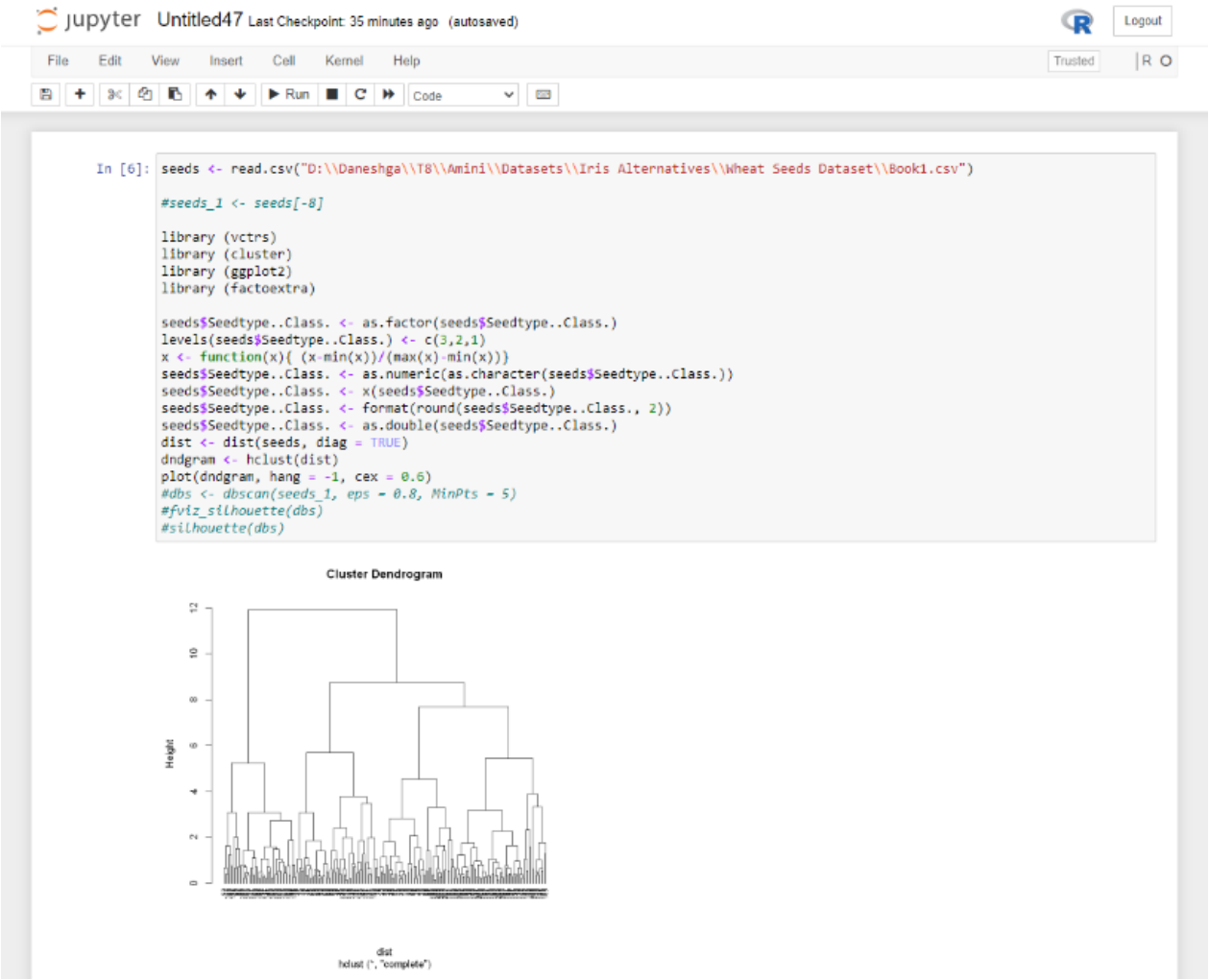
نام و نام خانوادگی: علی رضائی نژاد

شماره دانشجویی: ۹۶۰۱۸۴۱۵۶

مشخصه درس: ۹۱۳۵۱

نام استاد: خانم امینه امینی

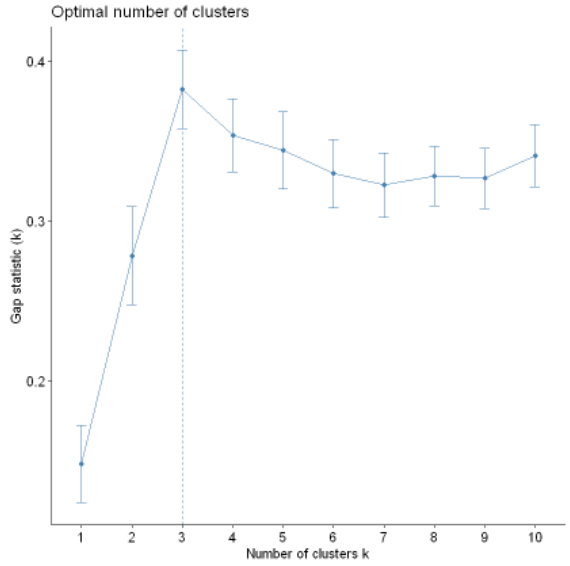
در تصاویر زیر خوشه‌بندی سلسله‌مراتبی و K-Means برای داده اعمال شده‌اند، با تابع clusGap تعداد ایده‌آل خوشه‌ها برای آنان مشخص و با روش سیلوئت با یکدیگر مقایسه شده‌اند. برای تفکیک بهتر به دلیل شلوغی گزارش‌کار، جزییات و روش‌های اضافه خوشه‌بندی (آگنس و مبتنی بر چگالی (DBSCAN)) در صفحات بعدیِ آن قابل ملاحظه هستند.



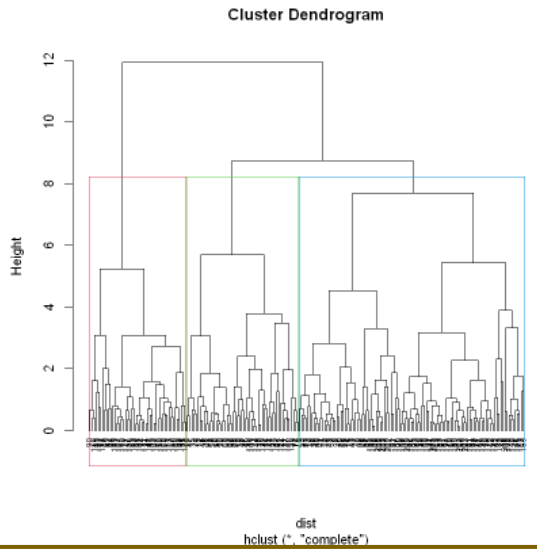
In [8]:
```
gap_stat <- clusGap(seeds, FUN = hcut, nstart = 25, K.max = 10, B = 60)
print(gap_stat)
```

```
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = seeds, FUNcluster = hcut, K.max = 10, B = 60, nstart = 25)
B=60 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstSEmax', SE.factor=1): 3
          logW   E.logW       gap     SE.sim
 [1,] 5.456081 5.604181 0.1481000 0.02393523
 [2,] 4.998009 5.276138 0.2781290 0.03065482
 [3,] 4.739087 5.121138 0.3820507 0.02465629
 [4,] 4.633549 4.987034 0.3534855 0.02287228
 [5,] 4.530358 4.874390 0.3440322 0.02411007
 [6,] 4.452203 4.781621 0.3294178 0.02131340
 [7,] 4.387688 4.710028 0.3223398 0.02018014
 [8,] 4.321201 4.649102 0.3279019 0.01879069
 [9,] 4.270001 4.596604 0.3266034 0.01881373
[10,] 4.209305 4.549946 0.3406407 0.01940048
```

In [9]:
```
fviz_gap_stat(gap_stat)
```



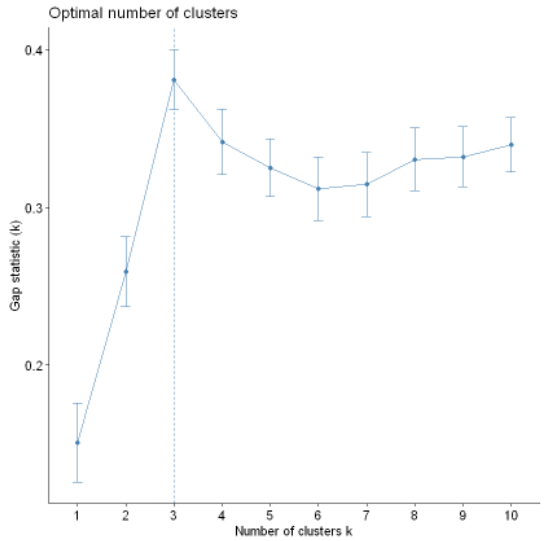
In [10]:
```
plot(dndgram, hang = -1, cex = 0.6)
rect.hclust(dndgram, k = 3, border = 2:5)
```



In [12]:
```
gap_stat <- clusGap(seeds, FUN = kmeans, nstart = 25, K.max = 10, B = 50)
print(gap_stat, method = "firstmax")
```

```
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = seeds, FUNcluster = kmeans, K.max = 10, B = 50, nstart = 25)
B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstmax'): 3
          logW   E.logW       gap     SE.sim
 [1,] 5.456081 5.606971 0.1508900 0.02486115
 [2,] 4.992403 5.251690 0.2592878 0.02218211
 [3,] 4.711451 5.092371 0.3809196 0.01921845
 [4,] 4.603905 4.945422 0.3415161 0.02074357
 [5,] 4.502382 4.827443 0.3250612 0.01817300
 [6,] 4.421860 4.733605 0.3117454 0.02004306
 [7,] 4.351476 4.666000 0.3145236 0.02035093
 [8,] 4.278530 4.608974 0.3304437 0.01984516
 [9,] 4.225917 4.557995 0.3320784 0.01910565
[10,] 4.173362 4.513025 0.3396633 0.01732672
```

In [13]:
```
fviz_gap_stat(gap_stat)
```



In [14]:
```
result <- kmeans( seeds, 3 , nstart = 25)
print(result)
```

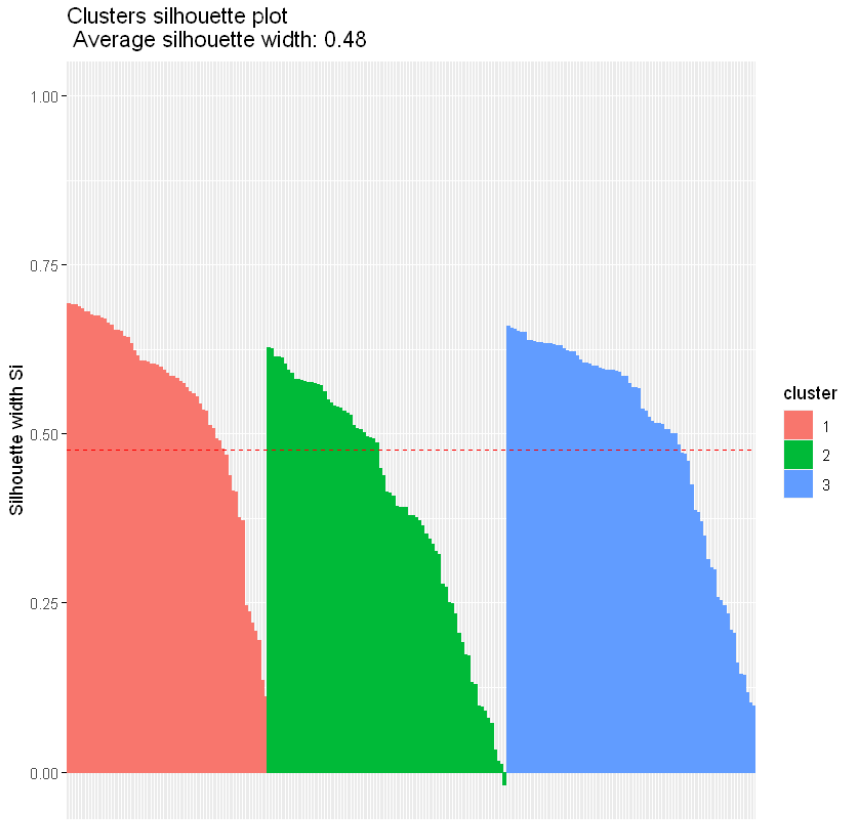
```
K-means clustering with 3 clusters of sizes 61, 73, 76

Cluster means:
      Area Perimeter Compactness Length.of.Kernel Width.of.Kernel
1 18.72180  16.29738   0.8850869         6.208934        3.722672
2 14.62616  14.45082   0.8789603         5.561466        3.274452
3 11.95053  13.26842   0.8520434         5.227105        2.870908
  Asymmetry.Coefficient Length.of.Kernel.Groove Seedtype..Class.
1              3.603590                6.066098         0.5081967
2              2.658852                5.187288         0.9041096
3              4.777987                5.091987         0.1052632

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 1 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 2 2 2 2 2 3 1 1 1 1
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
[112] 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 2 2 2 2 1 2 2 2 3 3 3 3 3 3 3 3
[149] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3
[186] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:
[1] 184.3545 215.1341 199.4653
 (between_SS / total_SS =  78.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

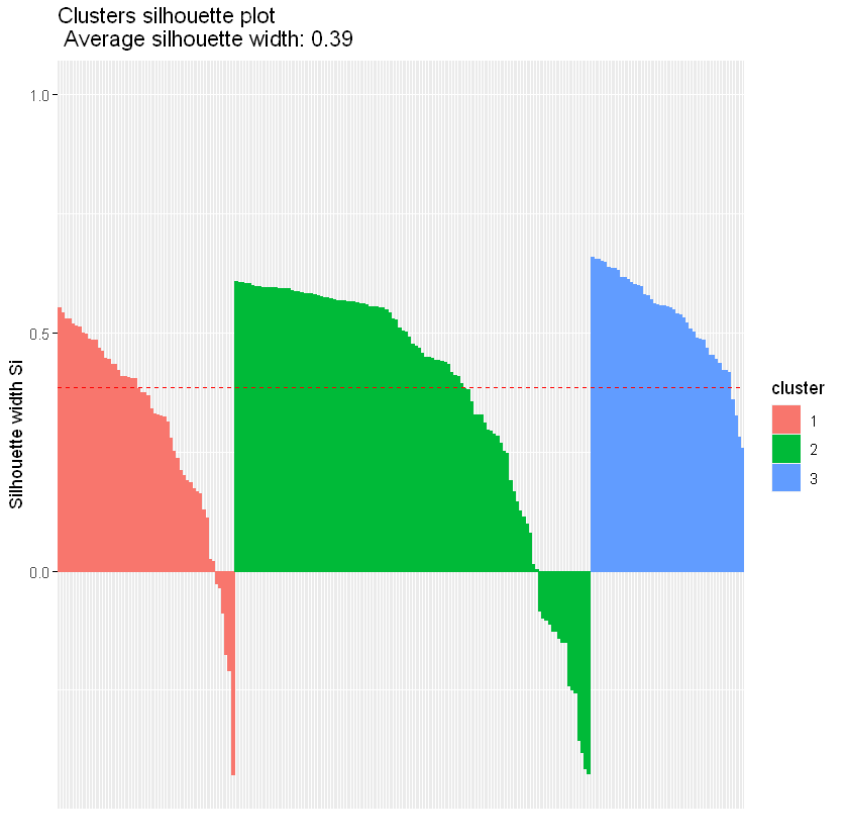
In [15]:
```
fviz_cluster(result, data = seeds)
```

اندازه‌گیری کیفیت خوشه‌بندی

In [18]:
```
s <- silhouette(result$cluster, dist)
fviz_silhouette(s)

s <- silhouette(cutree(dndgram, 3), dist)
fviz_silhouette(s)
```

```
  cluster size ave.sil.width
1       1   61          0.54
2       2   73          0.40
3       3   76          0.50
  cluster size ave.sil.width
1       1   54          0.30
2       2  109          0.36
3       3   47          0.53
```


Clusters silhouette plot
Average silhouette width: 0.48
Silhouette width Si
1.00
0.75
0.50
0.25
0.00
cluster
1
2
3


Clusters silhouette plot
Average silhouette width: 0.39
Silhouette width Si
1.0
0.5
0.0
cluster
1
2
3

ابتدا از علاوه بر بزرگ‌تر بودن ضریب سیلوئت در حاصل خوشه‌بندی K-Means، میزان کمترِ سیلوئت منفی در نمودار آن نسبت به خوشه‌بندی سلسله‌مراتبی نیز بیانگر برتری این روش برای داده‌ی ما است.

خوشه‌بندی سلسله‌مراتبی Hierarchical Clustering

ابتدا از روش خوشه‌بندی سلسله‌مراتبی را با روش پیش‌فرض (متد complete) کتابخانه‌ی stats و سپس از روش Agglomerative Nesting (AGNES) به عنوان یک روش برای خوشه‌بندی سلسله‌مراتبی تجمعی (AHC) استفاده می‌کنیم. کتابخانه‌های cluster و stats هر دو توابعی برای رسم دندوگرام در اختیار ما قرار می‌دهند. از توابع plot و pltree برای این منظور استفاده شده است.

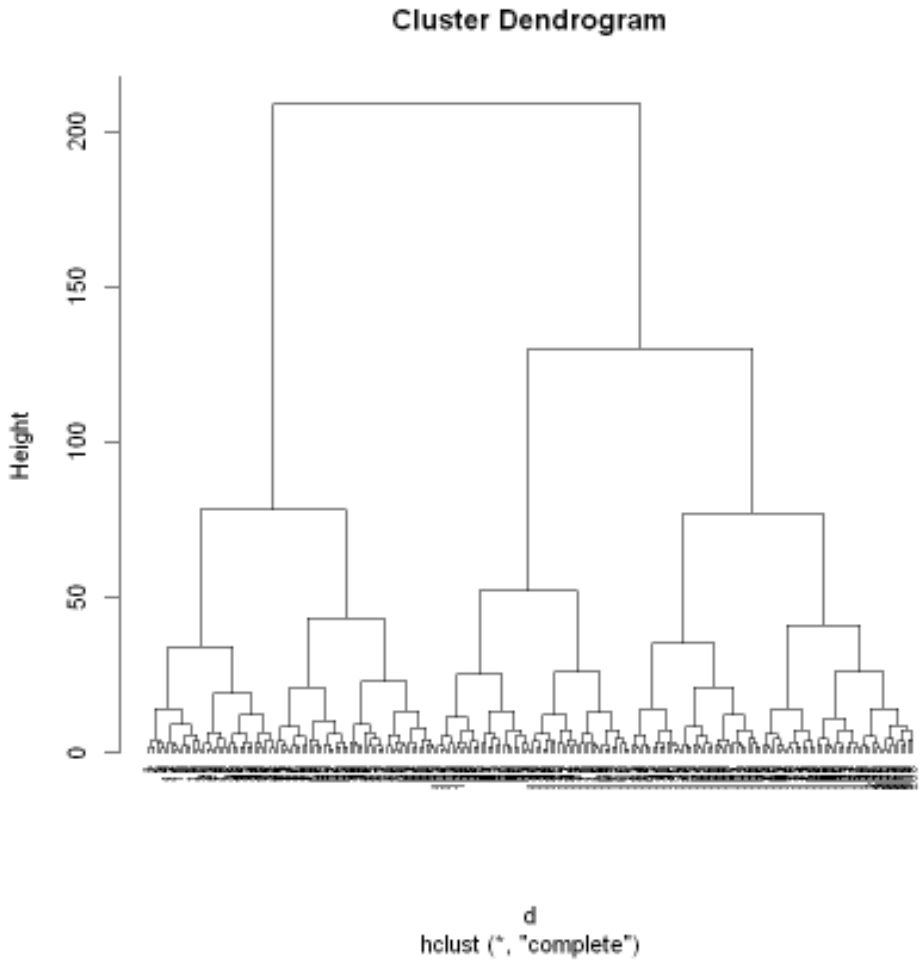
در آغاز، (همان‌گونه که در کامنت‌های کد مشخص شده) ماتریس عدم شباهت را با فاصله‌ی اقلیدسی محاسبه کردم زیرا به عنوان ورودی برای الگوریتم‌های خوشه‌بندی ضروری می‌باشد. گرچه در قسمت بعدی (AGNES) خواهیم دید که R به ما قابلیت ترسیم ماتریس عدم شباهت را از روی داده‌ی خام و در دستور خوشه‌بندی را می‌دهد.

(نیاز به استفاده از لایبرری factoextra در ادامه توضیح داده شده است.)

In [1]:
```
seeds <- read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
library(cluster)
library(purrr)
library(ggplot2)
library(factoextra)
library(stats)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

In [2]:
```
# Dissimilarity matrix
d <- dist(seeds, method = "euclidean")

# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```



کتابخانه کلاستر برای خوشه‌بندی AGNES چهار روش متفاوت را در اختیار ما قرار می‌دهد، در ادامه با محاسبه ضریب Agglomerative، بهترین روش (ward) را انتخاب کردم. همانطور که مشاهده می‌شود این مقدار را برای هر روش بررسی کردم که در نهایت روش ward منجر به بهترین نتیجه بود.
برای رجوع به داکیومنتیشنِ این کتابخانه از لینک استفاده شد.

cluster.pdf (r-project.org)

- برای استفاده از تابع map_dbl نیاز به کتابخانه‌ی purrr نیز بود که آن را در ابتدا (عکس اول) ایمپورت کردم.

```
In [19]: # Compute with agnes
hc2 <- agnes(seeds, method = "complete")

# Agglomerative coefficient
hc2$ac
```

0.990771966520607

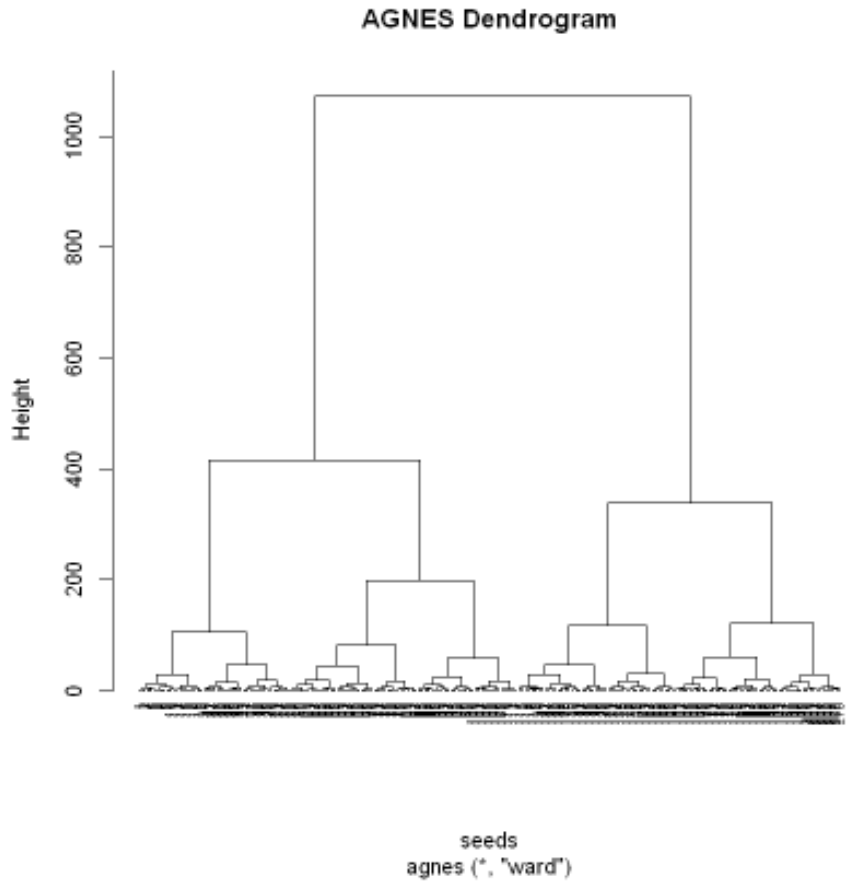
```
In [4]: # methods to assess
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient
ac <- function(x) {
  agnes(seeds, method = x)$ac
}

map_dbl(m, ac)
```

average: 0.982241531667919 **single:** 0.665399341125739 **complete:** 0.990771966520607 **ward:** 0.998179129705023

در اولین cell، مقدار ضریب Agglomerative، به تنهایی و برای روش complete محاسبه شد و در قسمت دوّم هر چهار مقدار برای مقایسه راحت‌تر در یک بخش مشخص شده‌اند.

در نهایت، نتیجه خوشه‌بندی تحت عنوان **hc3** ذخیره، و نمودار درختی آن با تابع **pltree** (صفحه‌ی ۶۱ - داکیومنتیشنِ کتابخانه **cluster**) رسم شد.

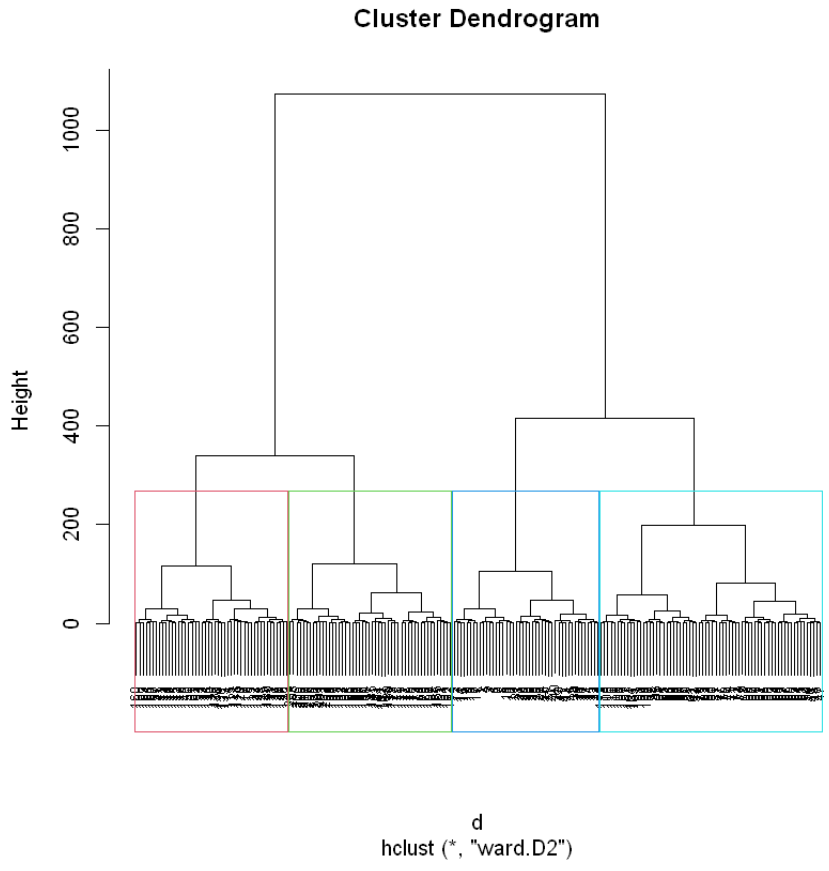
```
In [23]: hc3 <- agnes(seeds, method = "ward")
         pltree(hc3, cex = 0.6, hang = -1, main = "AGNES Dendrogram")
```



برای مشخص کردن واضح تر خوشه‌ها بر روی دندوگرام به طریق زیر میتوان آنان را مشخص کرد. تعداد خوشه‌ها (k) برابر چهار در نظر گرفته شده است. از آرگیومنت **border** برای تخصیص رنگ هر مستطیل استفاده می‌شود.

In [7]:
```
# Ward's method
hc5 <- hclust(d, method = "ward.D2" )

# Cut tree into 4 groups
sub_grp <- cutree(hc5, k = 4)
```

In [9]:
```
plot(hc5, cex = 0.6)
rect.hclust(hc5, k = 4, border = 2:5)
```

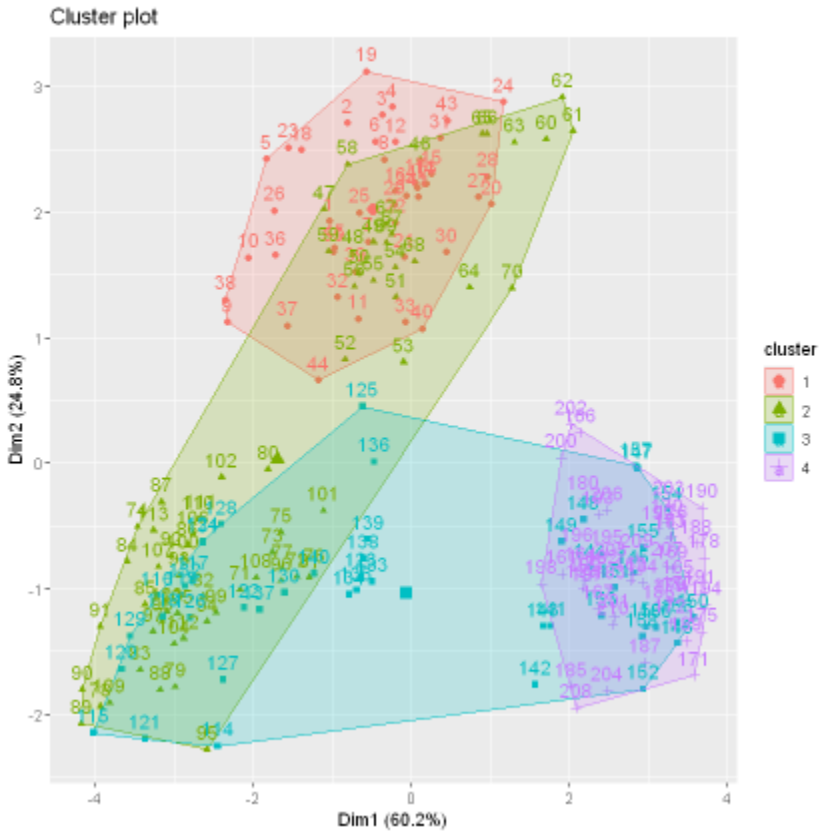


برای نمایش خوشه‌ها (در این‌جا چهار خوشه) در قالب اسکتررپلات و در یک دستگاه دکارتی از تابع **fviz_cluster** که در کتابخانه‌ی **factoextra** تعریف شده استفاده شد.

(برای نصب کتابخانه **factoextra** از فرمان زیر در ترمینال مامبا استفاده شد.)

mamba install -c conda-forge r-factoextra

```
In [10]: fviz_cluster(list(data = seeds, cluster = sub_grp))
```



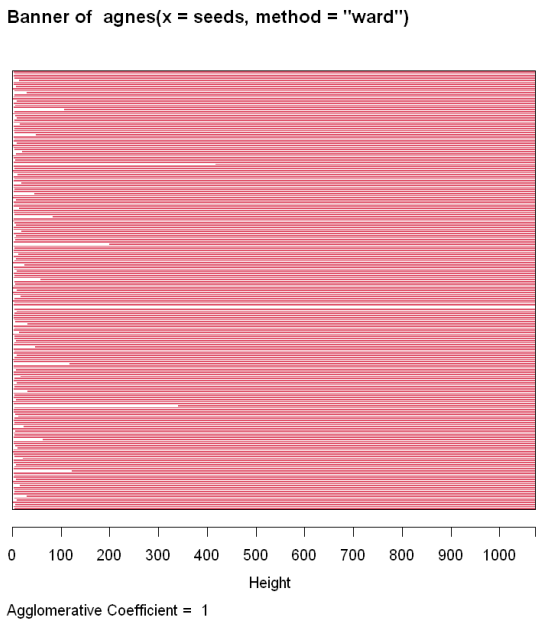
با تابع **cutree** نیز می‌توان خوشه هر یک از ۲۱۰ دانه گندم سنجیده شده را مشاهده کرد:

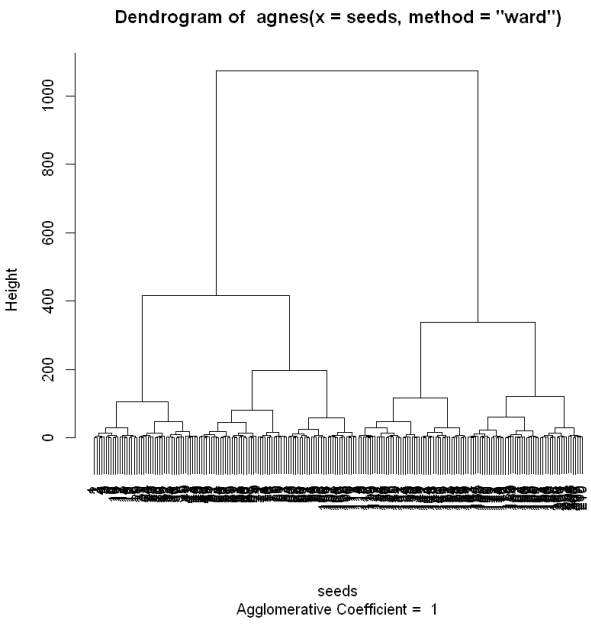
```
In [11]: hc_a <- agnes(seeds, method = "ward")
         cutree(as.hclust(hc_a), k = 4)
```

```
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
 2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
 2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
 3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  4  4  4  4  4
 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
 4  4  4  4  4  4  4  4  4  4  4  4
```

اجرای تابع **plot** برای **hc3** (حاصل خوشه‌بندی **AGNES**):

```
In [12]: plot(hc3)
```



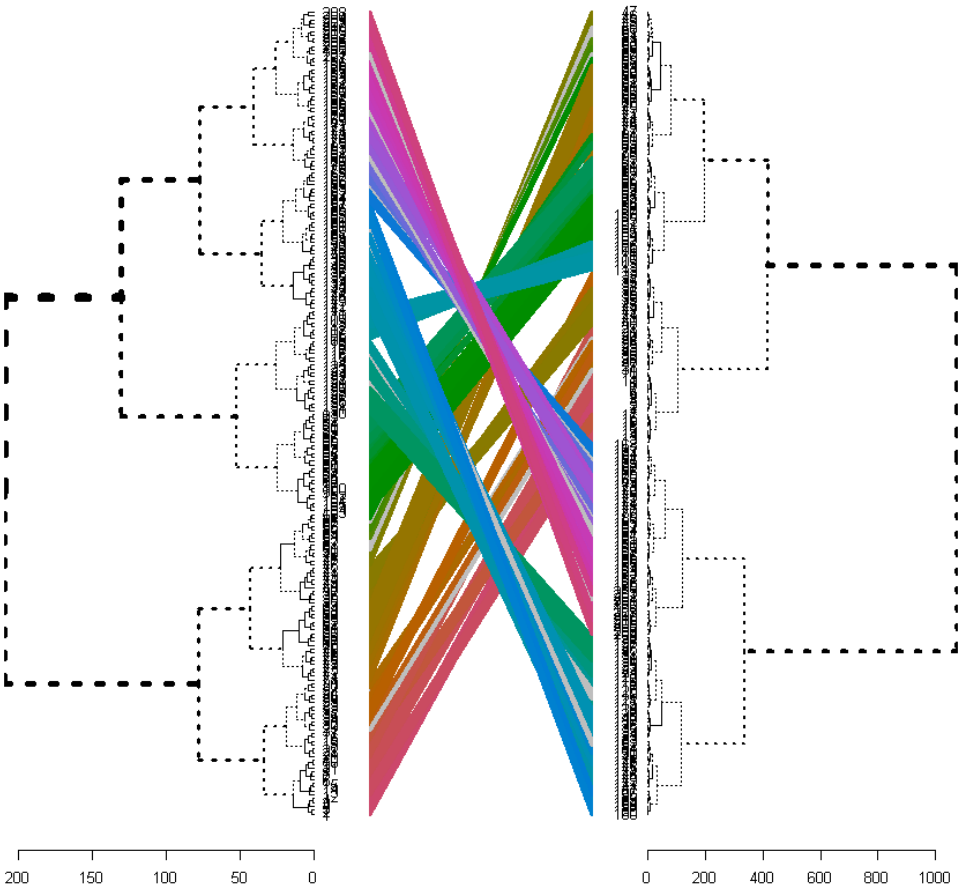


برای مقایسه دو درخت با تابع **tanglegram** کتابخانه **dendextend** را به طریق زیر نصب و استفاده کردم.

In [14]:
```
install.packages('dendextend')
```

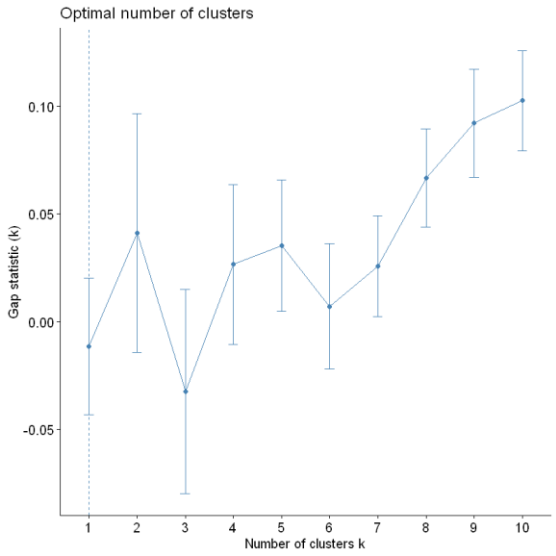
```
package 'dendextend' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Ali\AppData\Local\Temp\RtmpmwpnUF\downloaded_packages
```

In [16]:
```
library(dendextend)
# Compute distance matrix
res.dist <- dist(seeds, method = "euclidean")

# Compute 2 hierarchical clusterings
hc1 <- hclust(res.dist, method = "complete")
hc2 <- hclust(res.dist, method = "ward.D2")

# Create two dendrograms
dend1 <- as.dendrogram (hc1)
dend2 <- as.dendrogram (hc2)

tanglegram(dend1, dend2)
```



همانطور که از میزان درهم پیچیدگی نمودار صفحه گذشته و هم‌پوشانی خوشه‌ها نمایان بود «چهار» خوشه تعداد مناسبی نبود. تعداد ایده‌آل خوشه‌ها را به طریق زیر با **clusGap** و تعیین شیوه خوشه‌بندی به صورت سلسه‌مراتبی با **hcut**، بدست می‌آوریم.

```
In [17]: gap_stat <- clusGap(seeds, FUN = hcut, nstart = 25, K.max = 10, B = 50)
         fviz_gap_stat(gap_stat)
```




```
In [18]: sub_grp <- cutree(hc5, k = 1)
         fviz_cluster(list(data = seeds, cluster = sub_grp))
```

بدین ترتیب خوشه‌بندی را دوباره و فقط با یک خوشه عملی کردم.

خوشه‌بندی مبتنی بر چگالی DBSCAN

دو کتابخانه‌ی کاربردی برای خوشه‌بندی DBSCAN در R، **dbscan** و **fpc** نام دارند. در ادامه با استفاده از این دو، خوشه‌بندی را برای دیتابیس خود و دیتای **multishapes** که در کتابخانه **factoextra** برای آشنایی با DBSCAN وجود داشت، اعمال کردم.

مروری بر ساختار مجموعه داده

```
In [3]: seeds <- read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
str(seeds)
head(seeds)
#data(seeds)
```

```
'data.frame':   210 obs. of  9 variables:
 $ Instance              : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Area                  : num  15.3 14.9 14.3 13.8 16.1 ...
 $ Perimeter             : num  14.8 14.6 14.1 13.9 15 ...
 $ Compactness           : num  0.871 0.881 0.905 0.895 0.903 ...
 $ Length.of.Kernel      : num  5.76 5.55 5.29 5.32 5.66 ...
 $ Width.of.Kernel       : num  3.31 3.33 3.34 3.38 3.56 ...
 $ Asymmetry.Coefficient : num  2.22 1.02 2.7 2.26 1.35 ...
 $ Length.of.Kernel.Groove: num  5.22 4.96 4.83 4.8 5.17 ...
 $ Seedtype..Class.      : int  1 1 1 1 1 1 1 1 1 1 ...
```

A data.frame: 6 × 9

	Instance	Area	Perimeter	Compactness	Length.of.Kernel	Width.of.Kernel	Asymmetry.Coefficient	Length.of.Kernel.Groove	Seedtype..Class.
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
1	1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	2	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
3	3	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	1
4	4	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
5	5	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1
6	6	14.38	14.21	0.8951	5.386	3.312	2.462	4.956	1

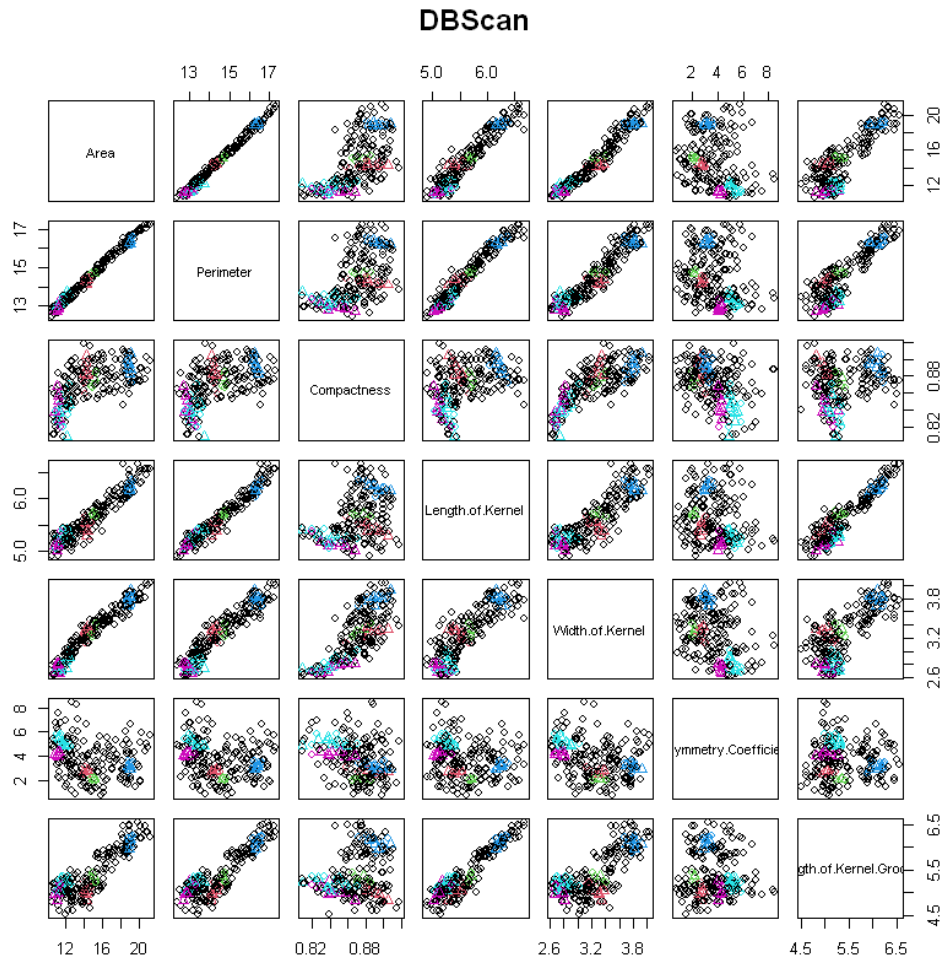
برای ادامه‌ی کار، دو فیلد **instance** و برچسب را از در محاسبات لحاظ نمی‌کنیم.

In [5]:
```r
# Installing Packages
install.packages("fpc")
# Loading package
library(fpc)

# Remove label form dataset
seeds_1 <- seeds[-9]
seeds_2 <- seeds_1[-1]

# Fitting DBScan clustering Model
# to training dataset
set.seed(220)  # Setting seed
Dbscan_cl <- dbscan(seeds_2, eps = 0.45, MinPts = 5)
Dbscan_cl

# Checking cluster
Dbscan_cl$cluster

# Table
table(Dbscan_cl$cluster, seeds$Seedtype..Class.)

# Plotting Cluster
plot(Dbscan_cl, seeds_2, main = "DBScan")
```

```
dbscan Pts=210 MinPts=5 eps=0.45
         0 1 2  3  4 5
border 163 3 4  5  7 1
seed     0 5 1  6  8 7
total  163 8 5 11 15 8
```

```
2 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0
0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 2 1 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 3 0 0
0 0 0 0 3 3 3 3 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0 0 0 0 0 3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 4 0 5 4 0 4 0 4 0 0 4 0 0
0 4 0 5 0 0 0 5 4 0 0 4 0 4 0 0 4 4 4 0 5 4 5 0 0 4 5 0 0 0 0 0 0
0 0 0 0 5 0 0 0 5 0 0 0
```

```
   1  2  3
0 57 59 47
1  8  0  0
2  5  0  0
3  0 11  0
4  0  0 15
5  0  0  8
```



خوشه‌بندی K-Means برای مقایسه، با پنج خوشه و پس از حذف دو ستون زائد (۲ از ۹) انجام شده است:

In [6]:
```
install.packages("dbscan")
```

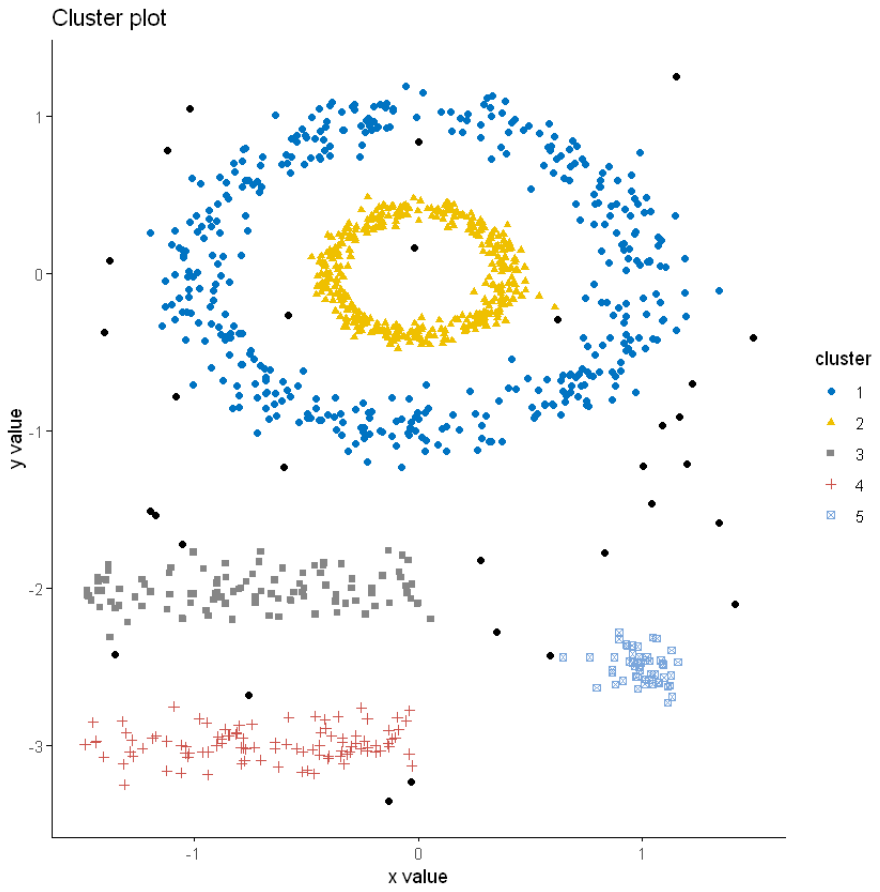
```
package 'dbscan' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Ali\AppData\Local\Temp\RtmpcXD3Oq\downloaded_packages
```

In [10]:
```
library(ggplot2)
library(factoextra)
data("seeds")
df <- seeds[, 2:8]
set.seed(123)
km.res <- kmeans(seeds, 5, nstart = 25)
fviz_cluster(km.res, seeds, frame = FALSE, geom = "point")
```

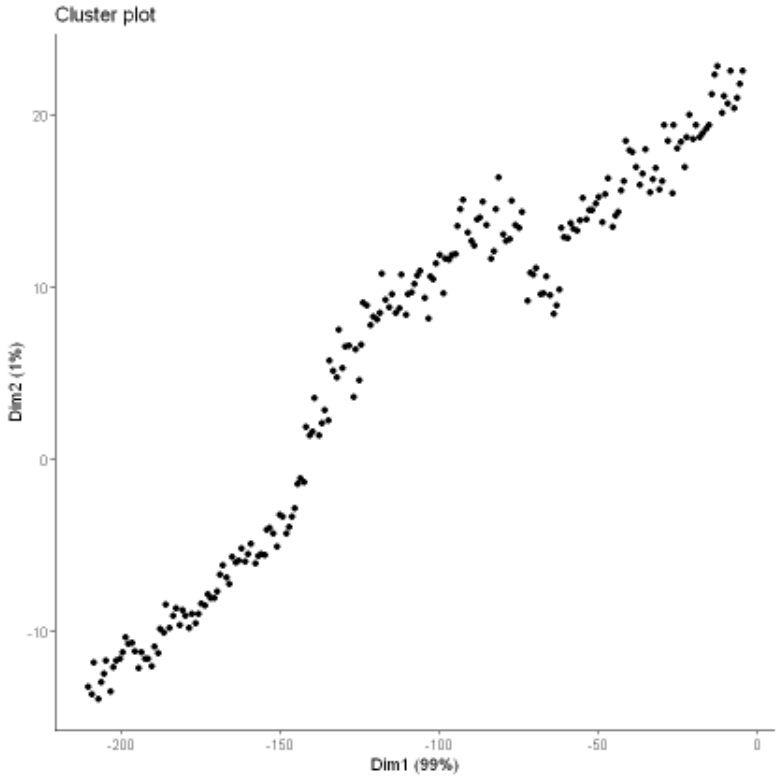
اجرای DBSCAN

تست:

In [14]:
```r
# Load the data
data("multishapes", package = "factoextra")
df <- multishapes[, 1:2]

# Compute DBSCAN using fpc package
library("fpc")
set.seed(123)
dbsample <- fpc::dbscan(df, eps = 0.15, MinPts = 5)

# Plot DBSCAN results
library("factoextra")
fviz_cluster(dbsample, data = df, stand = FALSE,
             ellipse = FALSE, show.clust.cent = FALSE,
             geom = "point",palette = "jco", ggtheme = theme_classic())
```



```
In [24]: dbs <- fpc::dbscan(seeds, eps = 0.15, MinPts = 5)
fviz_cluster(dbs, data = seeds, stand = FALSE,
             ellipse = FALSE, show.clust.cent = FALSE,
             geom = "point",palette = "jco", ggtheme = theme_classic())
```



این خوشه‌بندی به دو دلیلِ کوچکی بیش از حد حداکثر شعاع همسایگی و حساب کردن فیلد شمارش و کلاس، اشتباه است. تمامی داده‌ها با نقطه‌های مشکی رنگ نمایش داده شده‌اند که نشانگر آن است که با این شعاع کوچک همه داده‌ها نویز محسوب می‌شوند.

در صفحه بعد، با اصلاح این اشکالات، خوشه‌بندی **DBSCAN** با هردو لایبرری نام برده شده اجرا شده و به دو صورت آماری (تعداد و وضعیت کلاسترها) و بصری نمایش داده شده است. مشاهده می‌شود که هر دو روش با شعاع ۰/۸ و حداقل تعداد نقاط در همسایگی (**MinPts**) برابر با ۵، سه خوشه تشکیل دادند. تصویر صفحه زیرین بیانگر موقعیت این خوشه‌ها در فضای دو بعدی است.

jupyter Untitled46 Last Checkpoint: 26 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Trusted | R

Run Code

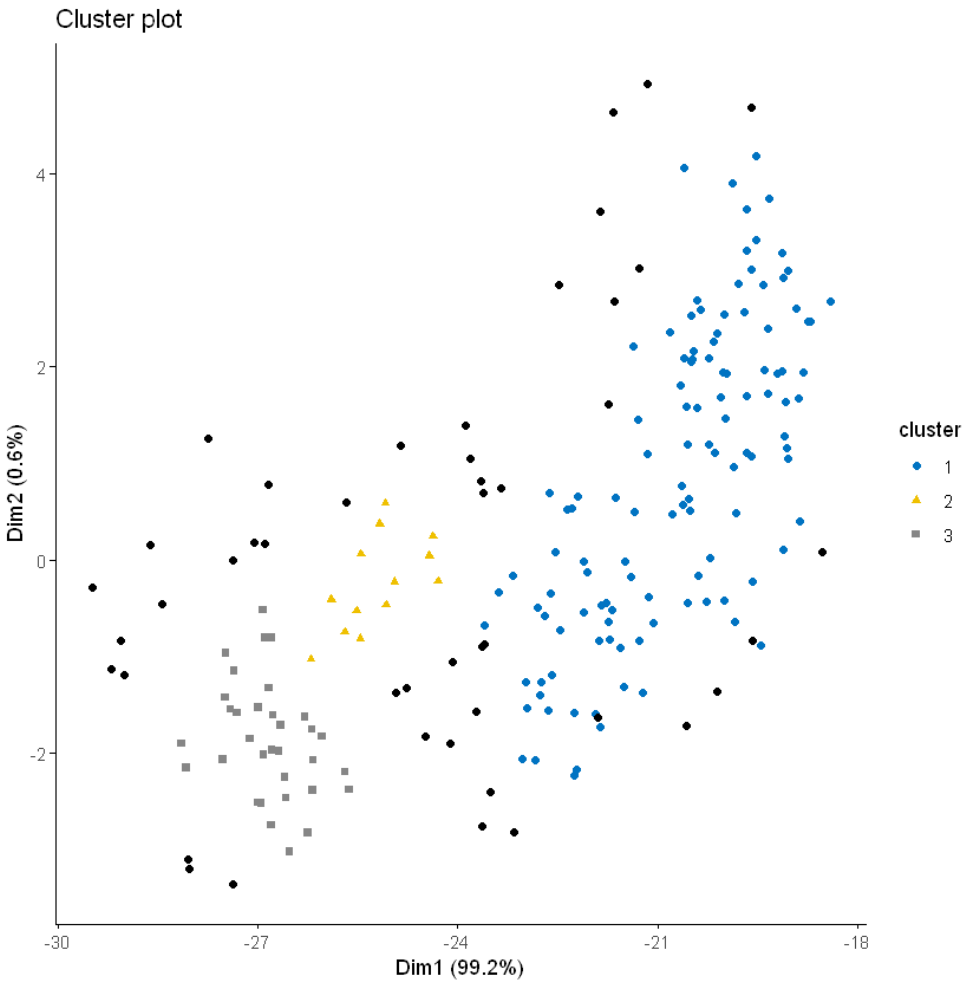
In [1]:
```
seeds <- read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Book1.csv")
```

In [23]:
```
seeds_1 <- seeds[-8]
library("dbscan")
```

In [25]:
```
dbscan(seeds_1, eps = 0.8, minPts = 5)
```

```
DBSCAN clustering for 210 objects.
Parameters: eps = 0.8, minPts = 5
The clustering contains 3 cluster(s) and 45 noise points.

  0   1   2   3
 45 118  14  33

Available fields: cluster, eps, minPts
```

In [27]:
```
# Compute DBSCAN using fpc package
library("fpc")
dbs <- fpc::dbscan(seeds_1, eps = 0.8, MinPts = 5)
fviz_cluster(dbs, data = seeds_1, stand = FALSE,
             ellipse = FALSE, show.clust.cent = FALSE,
             geom = "point",palette = "jco", ggtheme = theme_classic())
```

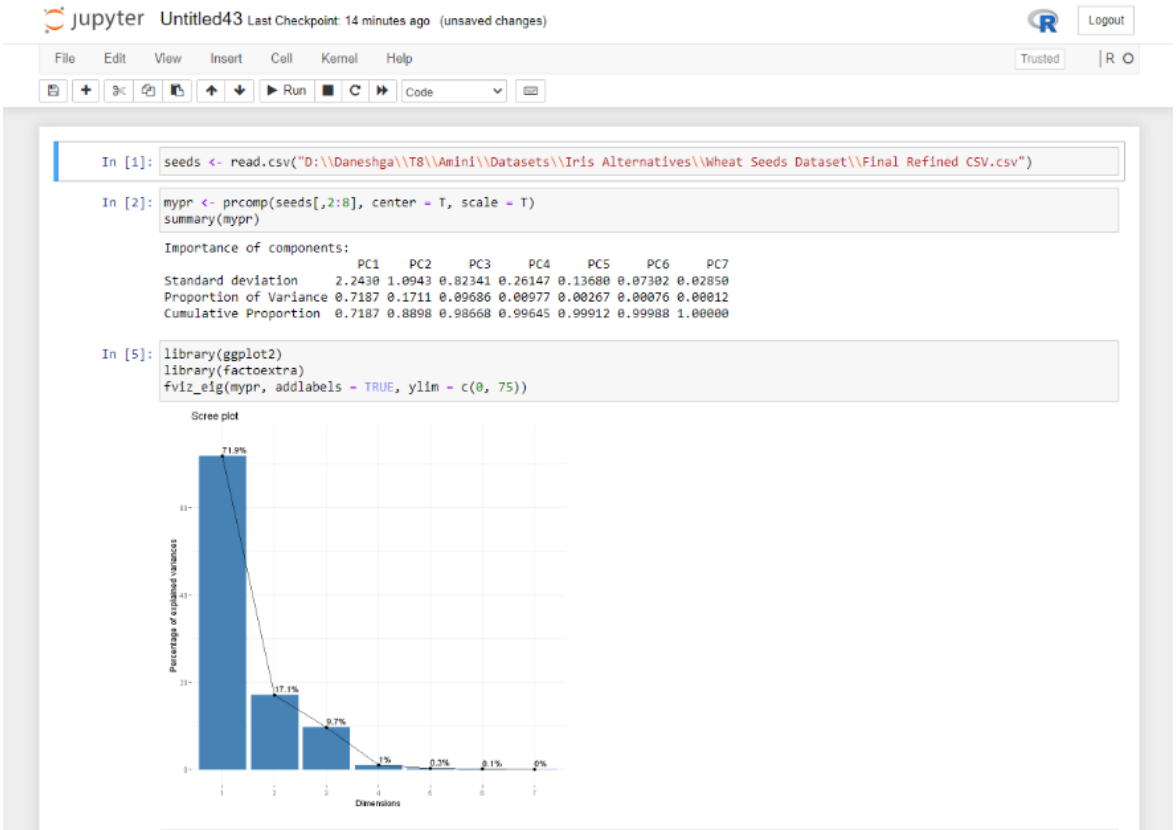


Silhouette Coefficient اندازه‌گیری کیفیت خوشه‌بندی

برای اندازه‌گیری کیفیت خوشه‌بندی های صورت گرفته از متد داخلی سیلوئت که با مقایسه میزان نزدیکی عناصر داده‌ای در یک خوشه و میزان تمایز یک خوشه از خوشه‌های دیگر عمل می‌کند، استفاده می‌کنیم. این سنجش کیفیت در صفحه‌ی ۵ و ۶ قابل ملاحظه است.

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

آنالیزهای متفرقه با خوشه‌بندی K-Means:



In [6]:
```
var <- get_pca_var(mypr)
var
```

```
Principal Component Analysis Results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the variables"
2 "$cor"     "Correlations between variables and dimensions"
3 "$cos2"    "Cos2 for the variables"
4 "$contrib" "contributions of the variables"
```

In [7]:
```
head(var$cos2, 4)
```

A matrix: 4 × 7 of type dbl

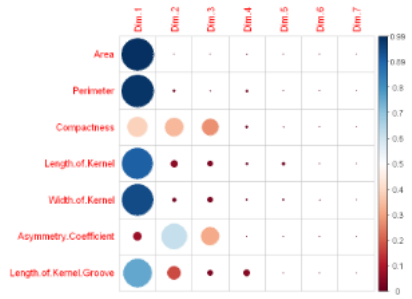
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7
Area	0.9939475	0.0008450341	0.0004537914	0.002563424	0.0007819319	0.0009696239	4.386450e-04
Perimeter	0.9810106	0.0084506414	0.0024277403	0.005967849	0.0005683714	0.0012093247	3.655035e-04
Compactness	0.3860875	0.3353216539	0.2688363174	0.007572511	0.0020708334	0.0001069541	4.276371e-06
Length.of.Kernel	0.9026272	0.0508079572	0.0304376025	0.004743335	0.0109831433	0.0003990721	1.739726e-06

In [9]:
```
install.packages('corrplot')
```

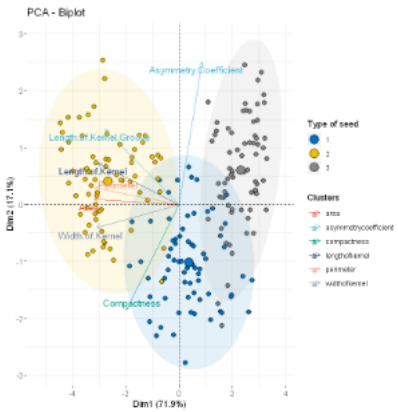
```
package 'corrplot' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Ali\AppData\Local\Temp\Rtmp8Y51EI\downloaded_packages
```

In [11]:
```
library(corrplot)
corrplot(var$cos2, is.corr=FALSE)
```

```
corrplot 0.89 loaded
```

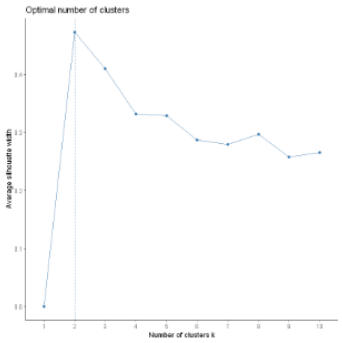


In [13]:
```
fviz_pca_biplot(mypr,
                # Fill individuals by groups
                geom.ind = as.factor("point"),
                addEllipses = TRUE, label = "var",
                pointshape = 21,
                pointsize = 2.5,
                fill.ind = as.factor(seeds$Seedtype..Class.),
                col.ind = as.factor("black"),
                # Color variable by groups
                col.var = factor(c("area", "perimeter", "compactness", "lengthofkernel",
                                   "widthofkernel", "asymmetrycoefficient", "asymmetrycoefficient")),


                legend.title = list(fill = "Type of seed", color = "Clusters"),
                repel = TRUE        # Avoid label overplotting
)+ggpubr::fill_palette("jco")+      # Individual fill color
  ggpubr::color_palette("npg")      # Variable colors
```



In [14]:
```
comp <- data.frame(mypr$x[,1:3])
```

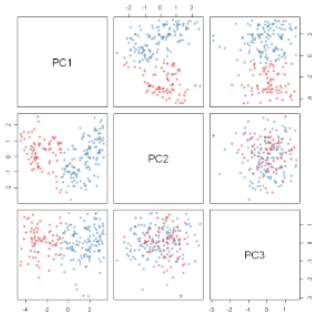
In [15]:
```
fviz_nbclust(comp, kmeans, method = "silhouette") + theme_classic()
```



In [17]:
```
install.packages("RColorBrewer")
```

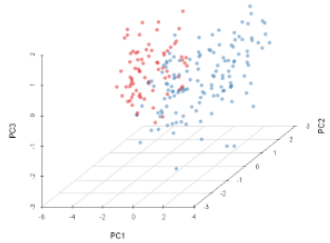
```
package 'RColorBrewer' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Ali\AppData\Local\Temp\RtmpBY51EI\downloaded_packages
```

In [18]:
```
library(RColorBrewer)
k <- kmeans(comp, 2, nstart=25, iter.max=1000)
palette(alpha(brewer.pal(9,'Set1'), 0.5))
plot(comp, col=k$clust, pch=16)
```



In [20]:
```
install.packages("scatterplot3d") # Install
library("scatterplot3d") # Load
scatterplot3d(comp[,1:3], pch=16, color = k$clust,
              grid = TRUE, box = FALSE)
```

```
package 'scatterplot3d' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Ali\AppData\Local\Temp\RtmpBY51EI\downloaded_packages
```



In [21]:
```
summary(k)
```

```
             Length Class  Mode
cluster      210    -none- numeric
centers        6    -none- numeric
totss          1    -none- numeric
withinss       2    -none- numeric
tot.withinss   1    -none- numeric
betweenss      1    -none- numeric
size           2    -none- numeric
iter           1    -none- numeric
ifault         1    -none- numeric
```

In [22]:
```
table(k$cluster)
```

```
 1  2
77 133
```

خلاصه‌ای از فرم دیتاست و مفهوم Compactness:

	A	P	C	L	W	AC	LG	Class
1	15.26	14.84	0.871	5.763	3.312	2.221	5.22	1
2	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
3	14.29	14.09	0.905	5.291	3.337	2.699	4.825	1
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
210	12.3	13.34	0.8684	5.243	2.974	5.637	5.063	3

1. A = Area مساحت
2. P = Perimeter محیط
3. C = Compactness دنسیتی/چگالی کل دانه
4. L = Length of Kernel طول هسته
5. W = Width of Kernel عرض هسته
6. AC = Asymmetry Coefficient ضریب عدم تقارن
7. LG = Length of Kernel Groove طول گروو دانه

Compactness به این صورت تعریف شده است:

$$C = \frac{4\pi A}{P^2}$$

a = Area مساحت (میلی‌متر مربع)

p = perimeter محیط (میلی‌متر)

π = 3.14