

R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
 Copyright (C) 2024 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
 You are welcome to redistribute it under certain conditions.
 Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
 Type 'contributors()' for more information and
 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
 'help.start()' for an HTML browser interface to help.
 Type 'q()' to quit R.

[Previously saved workspace restored]

```
> install.packages("palmerpenguins")
Installing package into 'C:/Users/Simon/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://lib.stat.cmu.edu/R/CRAN/bin/windows/contrib/4.4/palmerpenguins_0.1.1.zip'
Content type 'application/zip' length 3005488 bytes (2.9 MB)
downloaded 2.9 MB
```

package 'palmerpenguins' successfully unpacked and MD5 sums checked

```
The downloaded binary packages are in
  C:\Users\Simon\AppData\Local\Temp\Rtmp4IM1Me\downloaded_packages
> install.packages("fpc")
Installing package into 'C:/Users/Simon/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
trying URL 'https://lib.stat.cmu.edu/R/CRAN/bin/windows/contrib/4.4/fpc_2.2-13.zip'
Content type 'application/zip' length 865063 bytes (844 KB)
downloaded 844 KB
```

package 'fpc' successfully unpacked and MD5 sums checked

```
The downloaded binary packages are in
  C:\Users\Simon\AppData\Local\Temp\Rtmp4IM1Me\downloaded_packages
> install.packages("cluster")
Installing package into 'C:/Users/Simon/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
trying URL 'https://lib.stat.cmu.edu/R/CRAN/bin/windows/contrib/4.4/cluster_2.1.8.1.zip'
Content type 'application/zip' length 607933 bytes (593 KB)
downloaded 593 KB
```

package 'cluster' successfully unpacked and MD5 sums checked

```
The downloaded binary packages are in
  C:\Users\Simon\AppData\Local\Temp\Rtmp4IM1Me\downloaded_packages
> library(cluster)
Warning message:
package 'cluster' was built under R version 4.4.3
> library(fpc)
Warning message:
package 'fpc' was built under R version 4.4.3
> seeds<-read.table("/Users/Simon/Downloads/seeds_original (1).csv", header=T, sep=",")
> dim(seeds)
[1] 210 8
> seeds<-seeds[sample(1:nrow(seeds), size = 210), ]
> head(seeds)
   Area Perimeter Compactness LengthKernel WidthKernel AsymmetryCoefficient
195 12.11    13.27    0.8639      5.236      2.975      4.132
 74 19.11    16.26    0.9081      6.154      3.930      2.936
 64 13.22    13.84    0.8680      5.395      3.070      4.157
```

```

201 12.38      13.44      0.8609      5.219      2.989      5.472
102 17.99      15.86      0.8992      5.890      3.694      2.068
5   16.14      14.99      0.9034      5.658      3.562      1.355
  LengthKernelGroove      Class
195      5.012 Canadian
74      6.079      Rosa
64      5.088      Kama
201      5.045 Canadian
102      5.837      Rosa
5      5.175      Kama
> create_input<- function(input_data, class, in_attr) {
+
+   input_data<-na.omit(input_data) # listwise deletion of missing
+   input_data$type<-input_data[,class] # put class label in column type
+   input_data[,class]<-NULL
+   input_data<-input_data[,c(in_attr,"type")]
+
+ }
> input_data<-data.frame(seeds)
> class<-"Class"
> in_attr<-c("Area", "Perimeter", "Compactness", "LengthKernel", "WidthKernel", "AsymmetryCoefficient", "LengthKernelGroove")
> input_data<-create_input(input_data, class, in_attr)
> dist <- dist(scale(input_data[,in_attr]), method = "euclidean")
> mydata<-input_data[,in_attr]
> wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
> for (i in 2:15) wss[i] <- sum(kmeans(mydata,
+   centers=i)$withinss)
> plot(1:15, wss, type="b", xlab="Number of Clusters",
+   ylab="Within groups sum of squares")
> for (i in 2:10) {
+
+   # K-Means Cluster Analysis
+   fit <- kmeans(mydata,i) # i cluster solution
+
+   # get cluster means
+   aggregate(mydata,by=list(fit$cluster),FUN=mean)
+
+   # append cluster assignment
+   mydata <- data.frame(mydata, fit$cluster)
+   print ("No. of Clusters=")
+   print (i)
+   print(
+ cluster.stats(dist, as.numeric(as.factor(t(input_data$type))), mydata[,ncol(mydata)], compareonly=T)
+ )
+
+ }
[1] "No. of Clusters="
[1] 2
$corrected.rand
[1] 0.4725286

$vi
[1] 0.8110976

[1] "No. of Clusters="
[1] 3
$corrected.rand
[1] 0.7166199

$vi
[1] 0.6689058

[1] "No. of Clusters="
[1] 4
$corrected.rand
[1] 0.662972

$vi

```

```

[1] 0.7620818

[1] "No. of Clusters="
[1] 5
$corrected.rand
[1] 0.546608

$vi
[1] 1.001136

[1] "No. of Clusters="
[1] 6
$corrected.rand
[1] 0.5252724

$vi
[1] 1.081817

[1] "No. of Clusters="
[1] 7
$corrected.rand
[1] 0.4846178

$vi
[1] 1.194291

[1] "No. of Clusters="
[1] 8
$corrected.rand
[1] 0.3717045

$vi
[1] 1.390718

[1] "No. of Clusters="
[1] 9
$corrected.rand
[1] 0.3527937

$vi
[1] 1.452764

[1] "No. of Clusters="
[1] 10
$corrected.rand
[1] 0.3126053

$vi
[1] 1.574404

> fit <- kmeans(scale(input_data[, in_attr]), 3)
> plot(input_data[,in_attr], col=fit$cluster, pch=20)
> clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE,
+   cex=1,
+   #labels=2,
+   lines=0)
> table(input_data$type, fit$cluster)

      1  2  3
Canadian 0 66 4
Kama      2  6 62
Rosa      65  0  5
> # apply the standardize transformation to make all attributes comparable
> mydata<-scale(mydata)
> wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
> for (i in 2:15) wss[i] <- sum(kmeans(mydata,
+   centers=i)$withinss)
> plot(1:15, wss, type="b", xlab="Number of Clusters",
+   ylab="Within groups sum of squares")
> for (i in 2:10) {

```

```
+  
+ # K-Means Cluster Analysis  
+ fit <- kmeans(mydata,i) # i cluster solution  
+  
+ # get cluster means  
+ aggregate(mydata,by=list(fit$cluster),FUN=mean)  
+  
+ # append cluster assignment  
+ mydata <- data.frame(mydata, fit$cluster)  
+ print ("No. of Clusters=")  
+ print (i)  
+ print(  
+ cluster.stats(dist, as.numeric(as.factor(t(input_data$type))), mydata[,ncol(mydata)], compareon  
ly=T)  
+ )  
+  
+ }  
[1] "No. of Clusters="   
[1] 2  
$corrected.rand  
[1] 0.1073122  
  
$vi  
[1] 1.473036  
  
[1] "No. of Clusters="   
[1] 3  
$corrected.rand  
[1] 0.7426936  
  
$vi  
[1] 0.6434285  
  
[1] "No. of Clusters="   
[1] 4  
$corrected.rand  
[1] 0.662972  
  
$vi  
[1] 0.7620818  
  
[1] "No. of Clusters="   
[1] 5  
$corrected.rand  
[1] 0.546608  
  
$vi  
[1] 1.001136  
  
[1] "No. of Clusters="   
[1] 6  
$corrected.rand  
[1] 0.5779558  
  
$vi  
[1] 0.9859217  
  
[1] "No. of Clusters="   
[1] 7  
$corrected.rand  
[1] 0.3884237  
  
$vi  
[1] 1.312177  
  
[1] "No. of Clusters="   
[1] 8  
$corrected.rand  
[1] 0.3718653
```

```
$vi
[1] 1.377782

[1] "No. of Clusters="
[1] 9
$corrected.rand
[1] 0.3483191

$vi
[1] 1.446834

[1] "No. of Clusters="
[1] 10
$corrected.rand
[1] 0.333138

$vi
[1] 1.52105

> fit <- kmeans(scale(input_data[, in_attr]), 3)
> plot(input_data[,in_attr], col=fit$cluster, pch=20)
> clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE,
+         cex=1,
+         #labels=2,
+         lines=0)
>
```