Model 3

SITTY AZQUIA M. CAMAMA

2022-12-15

Clustering Techniques

In this document, we will perform and compare the following clustering techniques results such as K-Means, Hierarchical and Model based clustering without considering the binary output and categorical variables in the data. In these models, **radiomics data** is utilized.

1. K-Means Clustering

K-Means Clustering is one of the most well-known and commonly used clustering algorithms for partitioning observations into a set of k groups.

Load Helper Packages

```
library(dplyr)       # for data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)     # for data visualization
library(stringr)     # for string functionality
library(gridExtra)   # for manipulating the grid
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(bestNormalize)
```

Load Modeling Packages

```
library(tidyverse)   # data manipulation
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v purrr   0.3.4
```

```
## v tidyr   1.2.1     v forcats 0.5.2
## v readr   2.1.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x gridExtra::combine() masks dplyr::combine()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
```

```
library(cluster)     # for general clustering algorithms
library(factoextra)  # for visualizing cluster results
```

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

```
library(mclust)      # for fitting clustering algorithms
```

```
## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
##
## Attaching package: 'mclust'
##
## The following object is masked from 'package:purrr':
##
##     map
```

Load Data Sets

Radiomics data contains 197 rows and 431 columns: **Failure.binary**: binary property to predict

```
radiomicsdata <- read.csv("~/R CLASS/FINAL PROJECT/radiomics_completedata.csv")
View(radiomicsdata)
```

Data Pre-Processing

Check for null and missing values

Using **anyNA()** function, We can determine if any missing values in our data. The result shows either **TRUE** or **FALSE**. If true, omit the missing values using **na.omit().** Hence, our data has no missing values.

```
anyNA(radiomicsdata)
```

```
## [1] FALSE
```

Check for normality

The **Shapiro-Wilk's Test** is used to check the normality of the data. The null hypothesis states that data are normally distributed. Before, we test the normality, remove the categorical and binary variable.

```
rd <- radiomicsdata%>%select_if(is.numeric)
rd <- rd[,-1]
test <- apply(rd,2,function(x){shapiro.test(x)})
```

unlist() function is used to convert a list to vector, so we can have the list of p-value of all variables.

```
pvalue_list <- unlist(lapply(test, function(x) x$p.value))
```

Compute the sum of total variable with p-value less than 0.05 alpha. Thus, we have 428 variables that are not normally distributed and Entropy_cooc.W.ADC is normally distributed.

```
sum(pvalue_list<0.05)  # not normally distributed
```

```
## [1] 428
```

```
sum(pvalue_list>0.05)  # normally distributed
```

```
## [1] 1
```

```
test$Entropy_cooc.W.ADC
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.98903, p-value = 0.135
```

To normalized the data, remove first the categorical, binary and Entropy__cooc.W.ADC variable and use **orderNorm()** function. The **x.t** is the elements of orderNorm() function transformed original data.

```
rdnorm=radiomicsdata[,c(3,5:length(names(radiomicsdata)))]
rdnorm=apply(rdnorm,2,orderNorm)
rdnorm=lapply(rdnorm, function(x) x$x.t)
rdnorm=rdnorm%>%as.data.frame()
```

Test again using shapiro-wilk's test.

```
test2=apply(rdnorm,2,shapiro.test)
pvalue_list2=unlist(lapply(test2, function(x) x$p.value))
```

Compute the sum of total variable with p-value less than 0.05 alpha and more than 0.05 alpha. Finally, our data is normally distributed.

```
sum(pvalue_list2<0.05)   # not normally distributed
```

```
## [1] 0
```

```
sum(pvalue_list2>0.05)   # normally distributed
```

```
## [1] 428
```

Create new data with the **Entropy__cooc.W.ADC**, and **rdnorm** variables.

```
keep = select(radiomicsdata, c("Entropy_cooc.W.ADC"))
df = cbind(keep,rdnorm)
View(df)
```

Apply K-Means Clustering Algorithm

The main goal of k-means clustering is to **create clusters** with a total within-cluster variation that is minimized. So, perform K-means clustering with 3 clusters, 100 maximum number of iterations, and 100 nstart.

Let's start at 2 clusters of sizes 144, 50 have Within cluster sum of squares of 42657.82, 13404.39, respectively.

```
k <-kmeans(df, centers = 2, iter.max = 100, nstart = 100)
k
```

```
## K-means clustering with 2 clusters of sizes 50, 147
##
## Cluster means:
##   Entropy_cooc.W.ADC     Failure GLNU_align.H.PET Min_hist.PET Max_hist.PET
## 1           12.32898  0.08209356      -0.09199414    0.8581268    0.8761984
## 2           12.26146 -0.02791162       0.03129052   -0.2918799   -0.2980267
##   Mean_hist.PET Variance_hist.PET Standard_Deviation_hist.PET Skewness_hist.PET
## 1     0.8696553         0.4852431                   0.8614179         0.7993239
```

```
## 2    -0.2958011        -0.1650487                  -0.2929993        -0.2718789
##   Kurtosis_hist.PET Energy_hist.PET Entropy_hist.PET AUC_hist.PET  H_suv.PET
## 1       -0.03993469       0.8193998        1.2518393    1.2601035  0.8782050
## 2        0.01358323      -0.2787074       -0.4257957   -0.4286066 -0.2987092
##   Volume.PET X3D_surface.PET ratio_3ds_vol.PET ratio_3ds_vol_norm.PET
## 1  0.5234776       0.5377336         0.9130014              0.9194300
## 2 -0.1780536      -0.1829026        -0.3105447             -0.3127313
##   irregularity.PET tumor_length.PET Compactness_v1.PET Compactness_v2.PET
## 1        1.2601035        1.0133370          1.0167104          0.6816948
## 2       -0.4286066       -0.3446723         -0.3458196         -0.2318686
##   Spherical_disproportion.PET Sphericity.PET Asphericity.PET Center_of_mass.PET
## 1                   0.9194300      0.8346426       0.9073671          0.7289097
## 2                  -0.3127313     -0.2838920      -0.3086283         -0.2479285
##   Max_3D_diam.PET Major_axis_length.PET Minor_axis_length.PET
## 1       0.8104716             0.8723031             1.0391691
## 2      -0.2756706            -0.2967017            -0.3534589
##   Least_axis_length.PET Elongation.PET Flatness.PET Max_cooc.L.PET
## 1             0.8915085       1.233600    1.1995823      0.8463483
## 2            -0.3032342      -0.419592   -0.4080212     -0.2878736
##   Average_cooc.L.PET Variance_cooc.L.PET Entropy_cooc.L.PET DAVE_cooc.L.PET
## 1          1.1824572           0.9546964          1.2601035       1.1285694
## 2         -0.4021963          -0.3247267         -0.4286066      -0.3838672
##   DVAR_cooc.L.PET DENT_cooc.L.PET SAVE_cooc.L.PET SVAR_cooc.L.PET
## 1       0.9883594       1.2601035       1.1821908       0.9910248
## 2      -0.3361767      -0.4286066      -0.4021057      -0.3370833
##   SENT_cooc.L.PET ASM_cooc.L.PET Contrast_cooc.L.PET Dissimilarity_cooc.L.PET
## 1       1.2601035      0.8094807           0.7982773                1.1285694
## 2      -0.4286066     -0.2753336          -0.2715229               -0.3838672
##   Inv_diff_cooc.L.PET Inv_diff_norm_cooc.L.PET IDM_cooc.L.PET
## 1           1.2476933                1.2601035       1.184250
## 2          -0.4243855               -0.4286066      -0.402806
##   IDM_norm_cooc.L.PET Inv_var_cooc.L.PET Correlation_cooc.L.PET
## 1           1.2601035          1.1893919              1.0050778
## 2          -0.4286066         -0.4045551             -0.3418632
##   Autocorrelation_cooc.L.PET Tendency_cooc.L.PET Shade_cooc.L.PET
## 1                  0.8960104           0.9910248        0.4629514
## 2                 -0.3047654          -0.3370833       -0.1574664
##   Prominence_cooc.L.PET IC1_.L.PET IC2_.L.PET Coarseness_vdif_.L.PET
## 1             0.7005870 -0.5707380  1.2562010              0.8110514
## 2            -0.2382949  0.1941286 -0.4272793             -0.2758677
##   Contrast_vdif_.L.PET Busyness_vdif_.L.PET Complexity_vdif_.L.PET
## 1            0.6724232            0.5626713              1.1186189
## 2           -0.2287154           -0.1913848             -0.3804826
##   Strength_vdif_.L.PET SRE_align.L.PET LRE_align.L.PET GLNU_align.L.PET
## 1            0.4987130       1.2601035       1.2601035        0.4467533
## 2           -0.1696303      -0.4286066      -0.4286066       -0.1519569
##   RLNU_align.L.PET RP_align.L.PET LGRE_align.L.PET HGRE_align.L.PET
## 1        0.4109397      1.2601035        0.9931653        0.9236862
## 2       -0.1397754     -0.4286066       -0.3378113       -0.3141790
##   LGSRE_align.L.PET HGSRE_align.L.PET LGHRE_align.L.PET HGLRE_align.L.PET
## 1         0.9975106         0.9215848         0.9631838         0.9364447
## 2        -0.3392893        -0.3134642        -0.3276135        -0.3185186
##   GLNU_norm_align.L.PET RLNU_norm_align.L.PET GLVAR_align.L.PET
## 1             1.0309673             1.2601035         0.9868800
```

```
## 2           -0.3506692            -0.4286066        -0.3356735
##   RLVAR_align.L.PET Entropy_align.L.PET SZSE.L.PET LZSE.L.PET LGLZE.L.PET
## 1         1.0169181           1.2601035  1.2601035  1.1491524   1.0086056
## 2        -0.3458905          -0.4286066 -0.4286066 -0.3908681  -0.3430631
##   HGLZE.L.PET SZLGE.L.PET SZHGE.L.PET LZLGE.L.PET LZHGE.L.PET GLNU_area.L.PET
## 1   0.9357872   1.0252793   0.9346034   0.8688938   0.8735299       0.4579289
## 2  -0.3182950  -0.3487345  -0.3178923  -0.2955421  -0.2971190      -0.1557581
##   ZSNU.L.PET  ZSP.L.PET GLNU_norm.L.PET ZSNU_norm.L.PET GLVAR_area.L.PET
## 1  0.4303770  1.2601035       1.0303448       1.2601035        1.0033531
## 2 -0.1463867 -0.4286066      -0.3504574      -0.4286066       -0.3412766
##   ZSVAR.L.PET Entropy_area.L.PET Max_cooc.H.PET Average_cooc.H.PET
## 1   0.8970301          1.2601035      0.5503672          1.2601035
## 2  -0.3051123         -0.4286066     -0.1871997         -0.4286066
##   Variance_cooc.H.PET Entropy_cooc.H.PET DAVE_cooc.H.PET DVAR_cooc.H.PET
## 1           1.2346999          1.1893755       1.2585499       1.2504238
## 2          -0.4199659         -0.4045495      -0.4280782      -0.4253142
##   DENT_cooc.H.PET SAVE_cooc.H.PET SVAR_cooc.H.PET SENT_cooc.H.PET
## 1       1.2062372       1.2601035       1.2395185       0.9803265
## 2      -0.4102848      -0.4286066      -0.4216049      -0.3334444
##   ASM_cooc.H.PET Contrast_cooc.H.PET Dissimilarity_cooc.H.PET
## 1      0.5645624           1.1982041                1.2585499
## 2     -0.1920280          -0.4075524               -0.4280782
##   Inv_diff_cooc.H.PET Inv_diff_norm_cooc.H.PET IDM_cooc.H.PET
## 1           1.0670858                1.2601035      0.9268130
## 2          -0.3629543               -0.4286066     -0.3152425
##   IDM_norm_cooc.H.PET Inv_var_cooc_.H.PET Correlation_cooc.H.PET
## 1           1.2601035           0.9398894              1.0155183
## 2          -0.4286066          -0.3196903             -0.3454144
##   Autocorrelation_cooc.H.PET Tendency_cooc.H.PET Shade_cooc.H.PET
## 1                  1.2468553           1.2152572       -0.6079343
## 2                 -0.4241004          -0.4133528        0.2067804
##   Prominence_cooc.H.PET IC1_d.H.PET IC2_d.H.PET Coarseness_vdif.H.PET
## 1             0.9302514 -0.17635749   1.1955390             0.8000853
## 2            -0.3164121  0.05998554  -0.4066459            -0.2721379
##   Contrast_vdif.H.PET Busyness_vdif.H.PET Complexity_vdif.H.PET
## 1           0.6221434           0.4186155             0.9603116
## 2          -0.2116134          -0.1423862            -0.3266366
##   Strength_vdif.H.PET SRE_align.H.PET LRE_align.H.PET RLNU_align.H.PET
## 1           0.3354073       1.2601035       1.0641749        0.4119913
## 2          -0.1140841      -0.4286066      -0.3619642       -0.1401331
##   RP_align.H.PET LGRE_align.H.PET HGRE_align.H.PET LGSRE_align.H.PET
## 1      1.2601035        0.8113038        1.2442073         0.8113038
## 2     -0.4286066       -0.2759537       -0.4231998        -0.2759537
##   HGSRE_align.H.PET LGHRE_align.H.PET HGLRE_align.H.PET GLNU_norm_align.H.PET
## 1         1.2601035         0.8135873         0.8807012             0.8207375
## 2        -0.4286066        -0.2767306        -0.2995582            -0.2791624
##   RLNU_norm_align.H.PET GLVAR_align.H.PET RLVAR_align.H.PET Entropy_align.H.PET
## 1             1.2601035         1.1949441         0.6236846           1.2601035
## 2            -0.4286066        -0.4064436        -0.2121376          -0.4286066
##   SZSE.H.PET LZSE.H.PET LGLZE.H.PET HGLZE.H.PET SZLGE.H.PET SZHGE.H.PET
## 1  1.2232981  0.3925204   0.8109439   1.2392449   0.8108628   1.1985268
## 2 -0.4160878 -0.1335103  -0.2758313  -0.4215119  -0.2758037  -0.4076622
##   LZLGE.H.PET LZHGE.H.PET GLNU_area.H.PET ZSNU.H.PET  ZSP.H.PET GLNU_norm.H.PET
## 1   0.4585734   0.3498507       0.4825644  0.3391641  1.0420572       0.8472273
```

```
## 2  -0.1559773  -0.1189968      -0.1641376 -0.1153619 -0.3544412      -0.2881725
##   ZSNU_norm.H.PET GLVAR_area.H.PET ZSVAR_H.PET Entropy_area.H.PET
## 1       1.0696557        1.1796629   0.3045160          1.2601035
## 2      -0.3638285       -0.4012459  -0.1035769         -0.4286066
##   Max_cooc.W.PET Average_cooc.W.PET Variance_cooc.W.PET Entropy_cooc.W.PET
## 1      0.6427321          0.8516085           0.4647485          1.2449097
## 2     -0.2186164         -0.2896627          -0.1580777         -0.4234387
##   DAVE_cooc.W.PET DVAR_cooc.W.PET DENT_cooc.W.PET SAVE_cooc.W.PET
## 1       0.8677561       0.4904128       1.2360752       0.8504316
## 2      -0.2951551      -0.1668071      -0.4204337      -0.2892624
##   SVAR_cooc.W.PET SENT_cooc.W.PET ASM_cooc.W.PET Contrast_cooc.W.PET
## 1       0.4522152       1.2579126      0.7028941           0.5126415
## 2      -0.1538147      -0.4278614     -0.2390795          -0.1743678
##   Dissimilarity_cooc.W.PET Inv_diff_cooc.W.PET Inv_diff_norm_cooc.W.PET
## 1                0.8677561           1.1660786                1.2601035
## 2               -0.2951551          -0.3966254               -0.4286066
##   IDM_cooc.W.PET IDM_norm_cooc.W.PET Inv_var_cooc.W.PET Correlation_cooc.W.PET
## 1      0.9819276           1.2601035          1.0699412              1.0064888
## 2     -0.3339890          -0.4286065         -0.3639256             -0.3423431
##   Autocorrelation_cooc.W.PET Tendency_cooc.W.PET Shade_cooc.W.PET
## 1                  0.4690750           0.4522152       0.19878284
## 2                 -0.1595493          -0.1538147      -0.06761321
##   Prominence_cooc.W.PET IC1_d.W.PET IC2_d.W.PET Coarseness_vdif.W.PET
## 1            0.23882962 -0.24662665   1.2462317             0.7396460
## 2           -0.08123456  0.08388662  -0.4238883            -0.2515803
##   Contrast_vdif.W.PET Busyness_vdif.W.PET Complexity_vdif.W.PET
## 1           0.7732209           0.4575578             0.3702140
## 2          -0.2630003          -0.1556319            -0.1259231
##   Strength_vdif.W.PET SRE_align.W.PET LRE_align.W.PET GLNU_align.W.PET
## 1           0.5676102       1.2601035       1.2126937        0.4934063
## 2          -0.1930647      -0.4286066      -0.4124808       -0.1678253
##   RLNU_align.W.PET RP_align.W.PET LGRE_align.W.PET HGRE_align.W.PET
## 1        0.4137762      1.2601035        0.7745228        0.4723914
## 2       -0.1407402     -0.4286066       -0.2634431       -0.1606774
##   LGSRE_align.W.PET HGSRE_align.W.PET LGHRE_align.W.PET HGLRE_align.W.PET
## 1         0.8179559         0.4620087         0.6245529         0.4926271
## 2        -0.2782163        -0.1571458        -0.2124329        -0.1675603
##   GLNU_norm_align.W.PET RLNU_norm_align.W.PET GLVAR_align.W.PET
## 1             0.8201921             1.2601035         0.4832532
## 2            -0.2789769            -0.4286066        -0.1643718
##   RLVAR_align.W.PET Entropy_align.W.PET SZSE.W.PET LZSE.W.PET LGLZE.W.PET
## 1         0.6953090           1.2601035  1.2601035  0.6184897   0.8061871
## 2        -0.2364997          -0.4286066 -0.4286066 -0.2103706  -0.2742133
##   HGLZE.W.PET SZLGE.W.PET SZHGE.W.PET LZLGE.W.PET LZHGE.W.PET GLNU_area.W.PET
## 1   0.4775674   0.9268940   0.4673123   0.3729370   0.5380657       0.4985108
## 2  -0.1624379  -0.3152701  -0.1589498  -0.1268493  -0.1830155      -0.1695615
##   ZSNU.W.PET  ZSP.W.PET GLNU_norm.W.PET ZSNU_norm.W.PET GLVAR_area.W.PET
## 1  0.3924770  1.2531151       0.8418332       1.2455332        0.4768281
## 2 -0.1334956 -0.4262296      -0.2863378      -0.4236507       -0.1621864
##   ZSVAR.W.PET Entropy_area.W.PET Min_hist.ADC Max_hist.ADC Mean_hist.ADC
## 1   0.3944796          1.2601035    0.5356019    1.2549000     1.2496503
## 2  -0.1341767         -0.4286066   -0.1803357   -0.4267104    -0.4250511
##   Variance_hist.ADC Standard_Deviation_hist.ADC Skewness_hist.ADC
## 1         0.6994483                    1.130847         0.4427724
```

```
## 2       -0.2379076                  -0.384642        -0.1506029
##   Kurtosis_hist.ADC Energy_hist.ADC Entropy_hist.ADC AUC_hist.ADC Volume.ADC
## 1         0.2948765       0.8087608        1.2601035    1.2601035  0.5074105
## 2        -0.1002981      -0.2750884       -0.4286066   -0.4286063 -0.1725886
##   X3D_surface.ADC ratio_3ds_vol.ADC ratio_3ds_vol_norm.ADC irregularity.ADC
## 1       0.6201594         1.0660761              1.2601035        1.2601035
## 2      -0.2109386        -0.3626111             -0.4286066       -0.4286066
##   Compactness_v1.ADC Compactness_v2.ADC Spherical_disproportion.ADC
## 1          1.0616896          1.1019793                   1.2601035
## 2         -0.3611189         -0.3748229                  -0.4286066
##   Sphericity.ADC Asphericity.ADC Center_of_mass.ADC Max_3D_diam.ADC
## 1      1.2601035       1.1338612          0.4515334       0.9501040
## 2     -0.4286066      -0.3856671         -0.1535828      -0.3231646
##   Major_axis_length.ADC Minor_axis_length.ADC Least_axis_length.ADC
## 1             1.0970495             0.9851300             0.9153330
## 2            -0.3731461            -0.3350782            -0.3113378
##   Elongation.ADC Flatness.ADC Max_cooc.L.ADC Average_cooc.L.ADC
## 1      1.2566904    1.2282154      0.9028064          1.2528828
## 2     -0.4274457   -0.4177604     -0.3070770         -0.4261506
##   Variance_cooc.L.ADC Entropy_cooc.L.ADC DAVE_cooc.L.ADC DVAR_cooc.L.ADC
## 1           0.8853611          1.2601035       1.1650991       0.8752124
## 2          -0.3011432         -0.4286066      -0.3962922      -0.2976913
##   DENT_cooc.L.ADC SAVE_cooc.L.ADC SVAR_cooc.L.ADC SENT_cooc.L.ADC
## 1       1.2601035       1.2528828       0.8583253       1.0157352
## 2      -0.4286066      -0.4261506      -0.2919474      -0.3454882
##   ASM_cooc.L.ADC Contrast_cooc.L.ADC Dissimilarity_cooc.L.ADC
## 1      0.8370442           0.8134852                1.1650991
## 2     -0.2847126          -0.2766957               -0.3962922
##   Inv_diff_cooc.L.ADC Inv_diff_norm_cooc.L.ADC IDM_cooc.L.ADC
## 1           1.2578935                1.2601024      1.2115347
## 2          -0.4278549               -0.4286063     -0.4120866
##   IDM_norm_cooc.L.ADC Inv_var_cooc.L.ADC Correlation_cooc.L.ADC
## 1           1.2601035          1.2193186              1.0207723
## 2          -0.4286066         -0.4147342             -0.3472015
##   Autocorrelation_.L.ADC Tendency_cooc.L.ADC Shade_.L.ADC Prominence_cooc.L.ADC
## 1              1.0565602           0.8583253   0.23784081             0.5363148
## 2             -0.3593742          -0.2919474  -0.08089823            -0.1824200
##   IC1_.L.ADC IC2_.L.ADC Coarseness_vdif_.L.ADC Contrast_vdif_.L.ADC
## 1 -0.5804176  1.2488160              0.7057617            0.6734613
## 2  0.1974209 -0.4247674             -0.2400560           -0.2290685
##   Busyness_vdif_.L.ADC Complexity_vdif_.L.ADC Strength_vdif_.L.ADC
## 1            0.6863390               1.138812            0.3999373
## 2           -0.2334486              -0.387351           -0.1360331
##   SRE_align.L.ADC LRE_align.L.ADC GLNU_align.L.ADC RLNU_align.L.ADC
## 1       1.2601016       1.2601035        0.4874284        0.5054985
## 2      -0.4286066      -0.4286066       -0.1657920       -0.1719383
##   RP_align.L.ADC LGRE_align.L.ADC HGRE_align.L.ADC LGSRE_align.L.ADC
## 1      1.2601035        0.7915610        1.1312349         0.7928837
## 2     -0.4286066       -0.2692384       -0.3847738        -0.2696881
##   HGSRE_align.L.ADC LGHRE_align.L.ADC HGLRE_align.L.ADC GLNU_norm_align.L.ADC
## 1         1.1331911         0.7785402         1.1307935             1.1639080
## 2        -0.3854391        -0.2648096        -0.3846236            -0.3958871
##   RLNU_norm_align.L.ADC GLVAR_align.L.ADC RLVAR_align.L.ADC Entropy_align.L.ADC
## 1             1.2601035         0.9358820         1.0768529           1.2601035
```

```
## 2          -0.4286067        -0.3183272        -0.3662759          -0.4286066
##   SZSE.L.ADC LZSE.L.ADC LGLZE.L.ADC HGLZE.L.ADC SZLGE.L.ADC SZHGE.L.ADC
## 1  1.2601035  1.2004418   0.8011667   1.1481754   0.8030950   1.1433289
## 2 -0.4286067 -0.4083135  -0.2725057  -0.3905358  -0.2731619  -0.3888874
##   LZLGE.L.ADC LZHGE.L.ADC GLNU_area.L.ADC ZSNU.L.ADC  ZSP.L.ADC GLNU_norm.L.ADC
## 1   0.6995465   1.0756933       0.4929578  0.5099565  1.2601035       1.1575302
## 2  -0.2379409  -0.3658821      -0.1676727 -0.1734546 -0.4286066      -0.3937176
##   ZSNU_norm.L.ADC GLVAR_area.L.ADC ZSVAR.L.ADC Entropy_area.L.ADC
## 1       1.2601012        0.9490632   0.7539999          1.2601035
## 2      -0.4286064       -0.3228106  -0.2564626         -0.4286066
##   Max_cooc.H.ADC Average_cooc.H.ADC Variance_cooc.H.ADC Entropy_cooc.H.ADC
## 1      0.8206280          1.2601035           1.2601035          1.2601035
## 2     -0.2791249         -0.4286066          -0.4286066         -0.4286066
##   DAVE_cooc.H.ADC DVAR_cooc.H.ADC DENT_cooc.H.ADC SAVE_cooc.H.ADC
## 1       1.2601035       1.2575931       1.2600955       1.2601035
## 2      -0.4286066      -0.4277528      -0.4286066      -0.4286066
##   SVAR_cooc.H.ADC SENT_cooc.H.ADC ASM_cooc.H.ADC Contrast_cooc.H.ADC
## 1       1.2601035       1.2601035      0.8094090           1.2119605
## 2      -0.4286066      -0.4286066     -0.2753277          -0.4122315
##   Dissimilarity_cooc.H.ADC Inv_diff_cooc.H.ADC Inv_diff_norm_cooc.H.ADC
## 1                1.2601035           1.2597865                1.2601035
## 2               -0.4286066          -0.4284984               -0.4286066
##   IDM_cooc.H.ADC IDM_norm_cooc.H.ADC Inv_var_cooc.H.ADC Correlation_cooc.H.ADC
## 1      1.2386408           1.2601035          1.2416511               1.017390
## 2     -0.4213064          -0.4286066         -0.4223303              -0.346051
##   Autocorrelation_cooc.H.ADC Tendency_cooc.H.ADC Shade_cooc.H.ADC
## 1                  1.2601035           1.2601035        0.3894511
## 2                 -0.4286066          -0.4286066       -0.1324663
##   Prominence_cooc.H.ADC IC1_d.H.ADC IC2_d.H.ADC Coarseness_vdif.H.ADC
## 1             1.2601035  -0.4665213   1.2573571             0.7047403
## 2            -0.4286066   0.1586805  -0.4276725            -0.2397074
##   Contrast_vdif.H.ADC Busyness_vdif.H.ADC Complexity_vdif.H.ADC
## 1           1.2597865           0.6093025             1.2542493
## 2          -0.4284988          -0.2072457            -0.4266154
##   Strength_vdif.H.ADC SRE_align.H.ADC LRE_align.H.ADC GLNU_align.H.ADC
## 1           0.3532876       1.2601035       1.2601035        0.5141974
## 2          -0.1201658      -0.4286065      -0.4286066       -0.1748971
##   RLNU_align.H.ADC RP_align.H.ADC LGRE_align.H.ADC HGRE_align.H.ADC
## 1        0.5165398      1.2601035        1.0414292        1.2601035
## 2       -0.1756938     -0.4286067       -0.3542342       -0.4286066
##   LGSRE_align.H.ADC HGSRE_align.H.ADC LGHRE_align.H.ADC HGLRE_align.H.ADC
## 1         1.0333211         1.2601035         1.1034864         1.2601035
## 2        -0.3514752        -0.4286066        -0.3753355        -0.4286066
##   GLNU_norm_align.H.ADC RLNU_norm_align.H.ADC GLVAR_align.H.ADC
## 1             0.9864517             1.2601035         1.2601035
## 2            -0.3355290            -0.4286067        -0.4286066
##   RLVAR_align.H.ADC Entropy_align.H.ADC SZSE.H.ADC LZSE.H.ADC LGLZE.H.ADC
## 1         1.0733846           1.2601035  1.2601035  1.2601035    1.025167
## 2        -0.3650968          -0.4286067 -0.4286069 -0.4285691   -0.348696
##   HGLZE.H.ADC SZLGE.H.ADC SZHGE.H.ADC LZLGE.H.ADC LZHGE.H.ADC GLNU_area.H.ADC
## 1   1.2601035   0.9944530   1.2601035   1.0429835   1.2591423       0.5142307
## 2  -0.4286066  -0.3382493  -0.4286066  -0.3547563  -0.4282797      -0.1749084
##   ZSNU.H.ADC  ZSP.H.ADC GLNU_norm.H.ADC ZSNU_norm.H.ADC GLVAR_area.H.ADC
## 1  0.5168978  1.2601035       0.9860426       1.2601035        1.2601035
```

```
## 2 -0.1757982 -0.4286069      -0.3355441      -0.4286069       -0.4286066
##   ZSVAR.H.ADC Entropy_area.H.ADC Max_cooc.W.ADC Average_cooc.W.ADC
## 1   0.8192230          1.2601035      0.8102563          1.0308583
## 2  -0.2786455         -0.4286066     -0.2756021         -0.3506321
##   Variance_cooc.W.ADC DAVE_cooc.W.ADC DVAR_cooc.W.ADC DENT_cooc.W.ADC
## 1           0.6697559       1.1715010       0.7247077       1.2601035
## 2          -0.2278081      -0.3984697      -0.2464992      -0.4286066
##   SAVE_cooc.W.ADC SVAR_cooc.W.ADC SENT_cooc.W.ADC ASM_cooc.W.ADC
## 1       1.0322287       0.6197202       0.9901887      0.8094624
## 2      -0.3510982      -0.2107892      -0.3367989     -0.2753303
##   Contrast_cooc.W.ADC Dissimilarity_cooc.W.ADC Inv_diff_cooc.W.ADC
## 1           0.7416618                1.1715010           1.1925307
## 2          -0.2522659               -0.3984697          -0.4056228
##   Inv_diff_norm_cooc.W.ADC IDM_cooc.W.ADC IDM_norm_cooc.W.ADC
## 1                1.2601035      1.1987330           1.2601035
## 2               -0.4286066     -0.4077323          -0.4286066
##   Inv_var_cooc.W.ADC Correlation_cooc.W.ADC Autocorrelation_cooc.W.ADC
## 1           1.190806              1.0204847                  0.7149876
## 2          -0.405036             -0.3471036                 -0.2431931
##   Tendency_cooc.W.ADC Shade_cooc.W.ADC Prominence_cooc.W.ADC IC1_d.W.ADC
## 1           0.6197202       0.18619581             0.3040448  -0.5969332
## 2          -0.2107892      -0.06333191            -0.1034179   0.2030385
##   IC2_d.W.ADC Coarseness_vdif.W.ADC Contrast_vdif.W.ADC Busyness_vdif.W.ADC
## 1   1.2601035             0.7345515           0.6702456           0.9869673
## 2  -0.4286066            -0.2498473          -0.2279747          -0.3357031
##   Complexity_vdif.W.ADC Strength_vdif.W.ADC SRE_align.W.ADC LRE_align.W.ADC
## 1             0.4927289           0.5797181       1.2601035       1.2601035
## 2            -0.1675949          -0.1971830      -0.4286066      -0.4286066
##   GLNU_align.W.ADC RLNU_align.W.ADC RP_align.W.ADC LGRE_align.W.ADC
## 1        0.5692557        0.5040911      1.2601035        0.7918681
## 2       -0.1936244       -0.1714596     -0.4286066       -0.2693426
##   HGRE_align.W.ADC LGSRE_align.W.ADC HGSRE_align.W.ADC LGHRE_align.W.ADC
## 1        0.7331058         0.7942434         0.7328334          0.778014
## 2       -0.2493557        -0.2701505        -0.2492631         -0.264635
##   HGLRE_align.W.ADC GLNU_norm_align.W.ADC RLNU_norm_align.W.ADC
## 1         0.7398451             0.9383067             1.2601035
## 2        -0.2516480            -0.3191516            -0.4286067
##   GLVAR_align.W.ADC RLVAR_align.W.ADC Entropy_align.W.ADC SZSE.W.ADC LZSE.W.ADC
## 1         0.7098564         1.0097017           1.2601035  1.2601035  1.2601035
## 2        -0.2414478        -0.3434379          -0.4286066 -0.4286066 -0.4286066
##   LGLZE.W.ADC HGLZE.W.ADC SZLGE.W.ADC SZHGE.W.ADC LZLGE.W.ADC LZHGE.W.ADC
## 1   0.7975267   0.7333844   0.8019663   0.7316883   0.7154832   0.7564613
## 2  -0.2712675  -0.2494505  -0.2727780  -0.2488736  -0.2433616  -0.2572998
##   GLNU_area.W.ADC ZSNU.W.ADC  ZSP.W.ADC GLNU_norm.W.ADC ZSNU_norm.W.ADC
## 1       0.5728375  0.4925395  1.2601035       0.9198583       1.2601035
## 2      -0.1948427 -0.1675304 -0.4286066      -0.3128771      -0.4286066
##   GLVAR_area.W.ADC ZSVAR.W.ADC Entropy_area.W.ADC
## 1        0.7168983   1.0225936          1.2601035
## 2       -0.2438430  -0.3478209         -0.4286066
##
## Clustering vector:
##   [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
##  [38] 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
##  [75] 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
```

```
## [112] 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 1
## [149] 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
## [186] 1 1 1 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 13404.39 42657.82
##  (between_SS / total_SS =  33.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

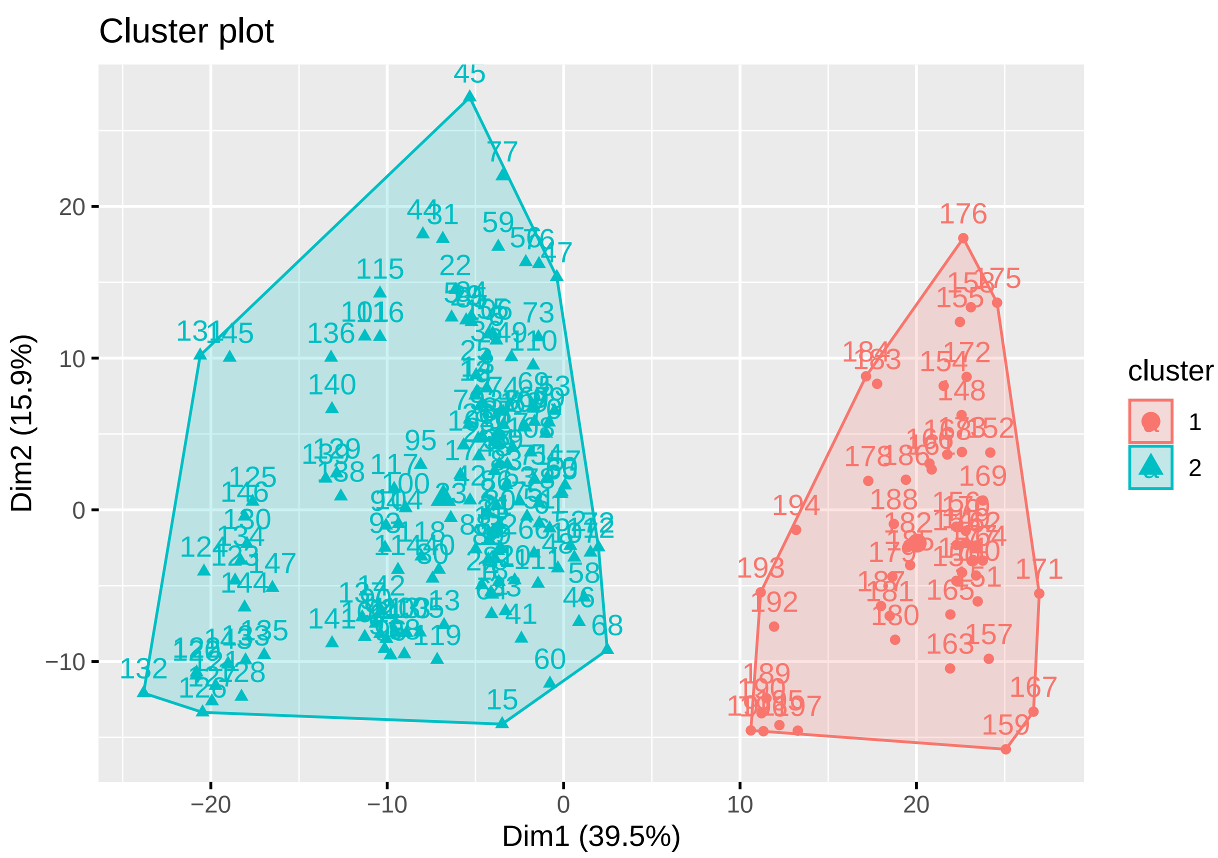
The quality of the k-means partition is measured by the **SSwithin**, and we want it to be as little as feasible. Thus, we have 33.2%.

```
k$betweenss/k$totss
```

```
## [1] 0.3324189
```

To plot the 2 clusters, use **fviz_cluster()** function.

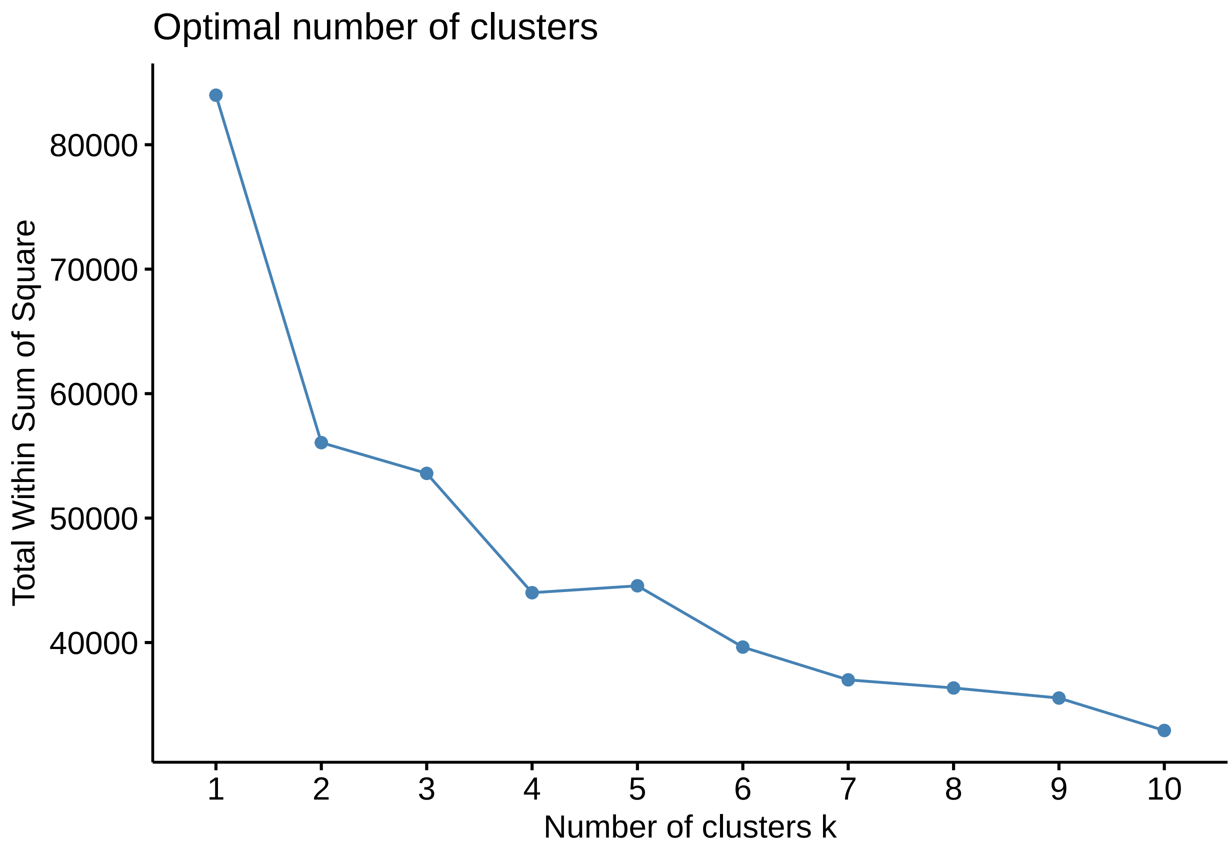
```
fviz_cluster(k, data = df)
```



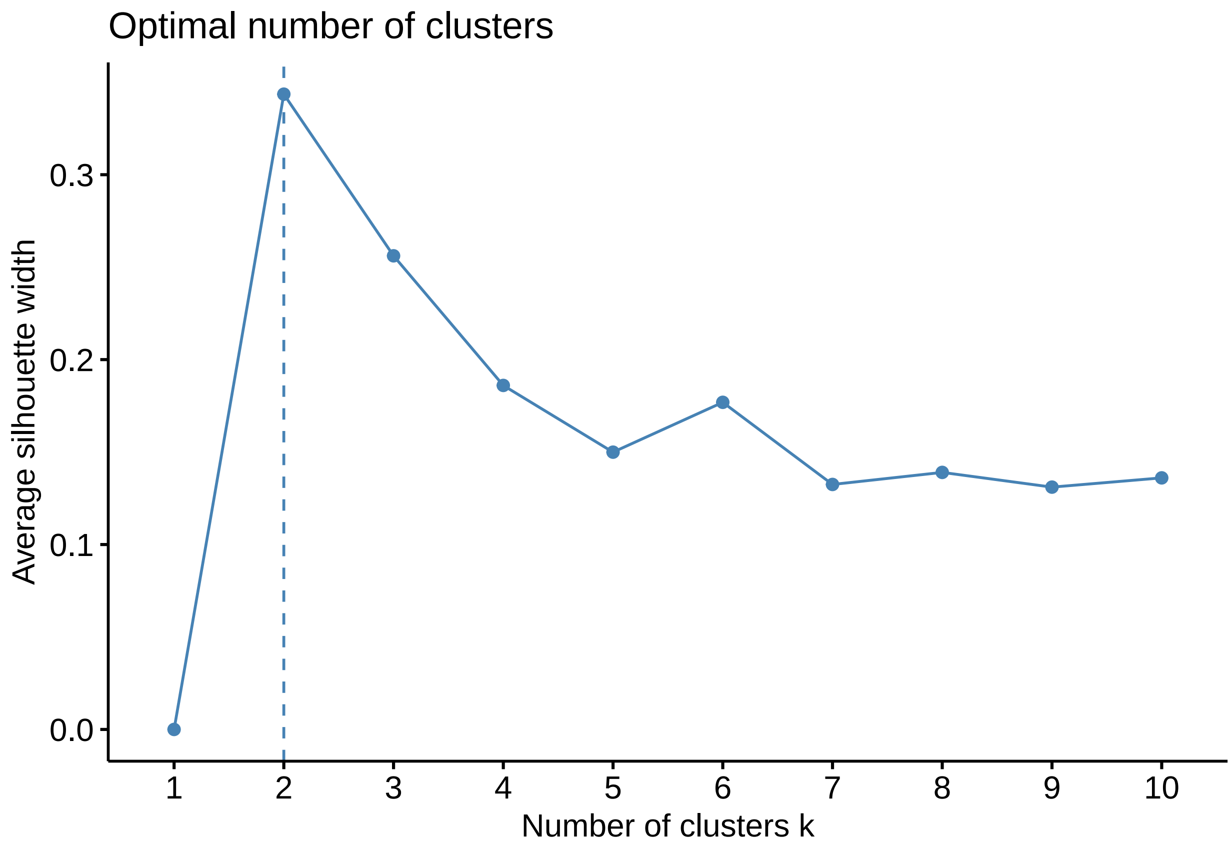
Determining Optimal Clusters

Using **Within Sum of Squares**, **Silhouette** and **gap_stat** plots, are another method to determine the optimal value of K number of clusters. It suggest with 2 clusters.

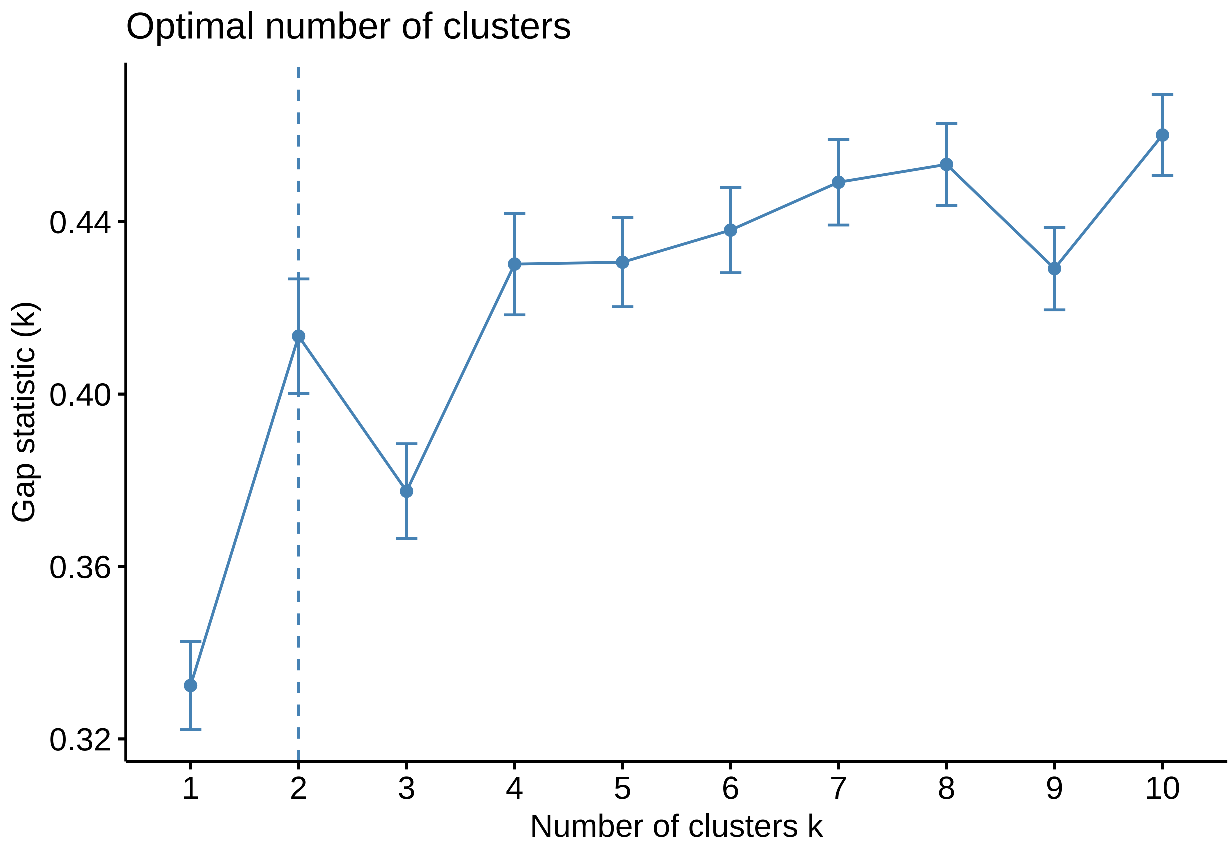
```
fviz_nbclust(df, kmeans, method = "wss")
```




```
fviz_nbclust(df, kmeans, method = "silhouette")
```

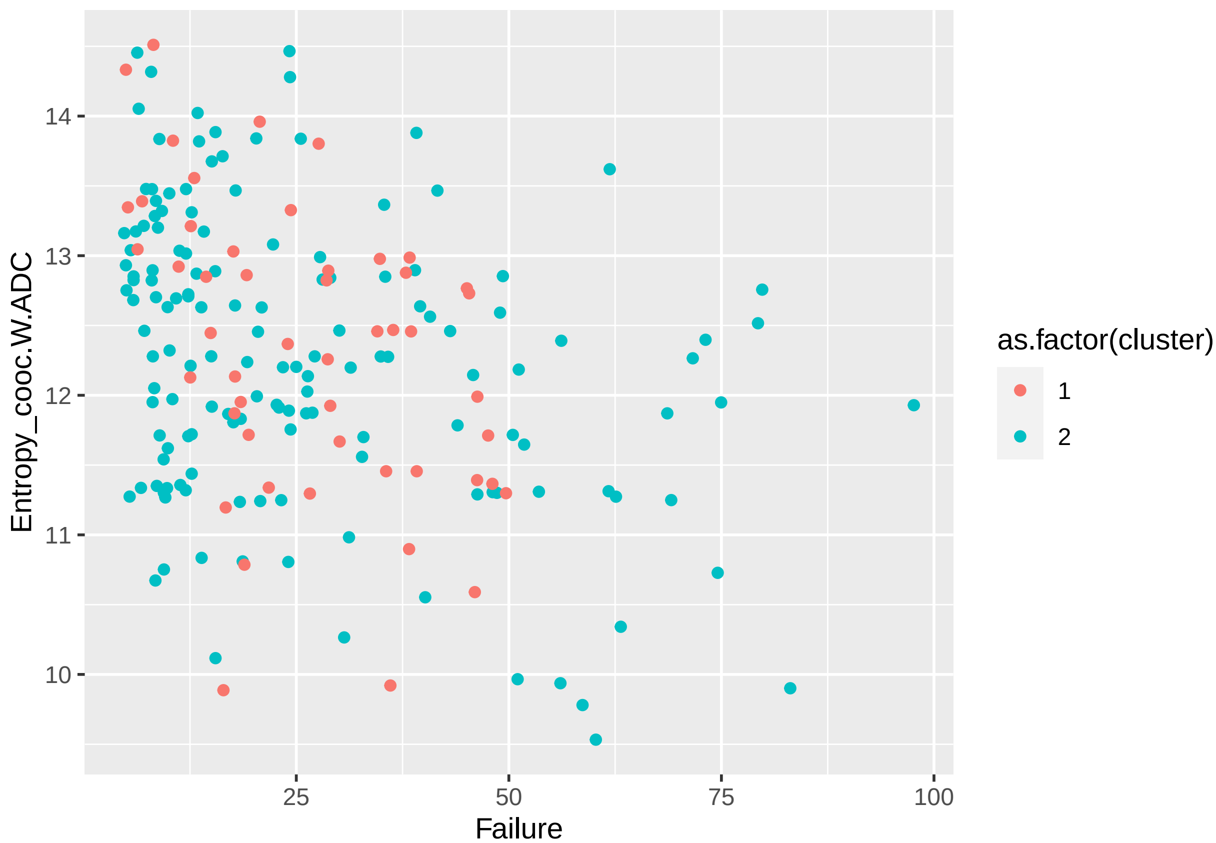



```
fviz_nbclust(df, kmeans, method = "gap_stat")
```



Visualize clusters using the original variables where **x is Failure** and **y is Entropy_cooc.W.ADC**

```
radiomicsdata <- radiomicsdata |> mutate(cluster = k$cluster)
radiomicsdata |> ggplot(aes(x = Failure, y = Entropy_cooc.W.ADC, col = as.factor(cluster))) + geom_poin
```



2. Heirarchical Clustering

An alternate method to k-means clustering for identifying groupings in a data set is hierarchical clustering. Unlike kmeans,the number of clusters does not need to be predetermined because in this method will build a hierarchy of clusters.

Standardize Data

Before building a clustering model, standardization of data is required.

```
hdf <- radiomicsdata %>%
  select_if(is.numeric) %>%  # select numeric columns
  select(-Failure.binary) %>%    # remove target column
  mutate_all(as.double) %>%  # coerce to double type
  scale()
```

Apply Heirarchical Clustering Algorithm

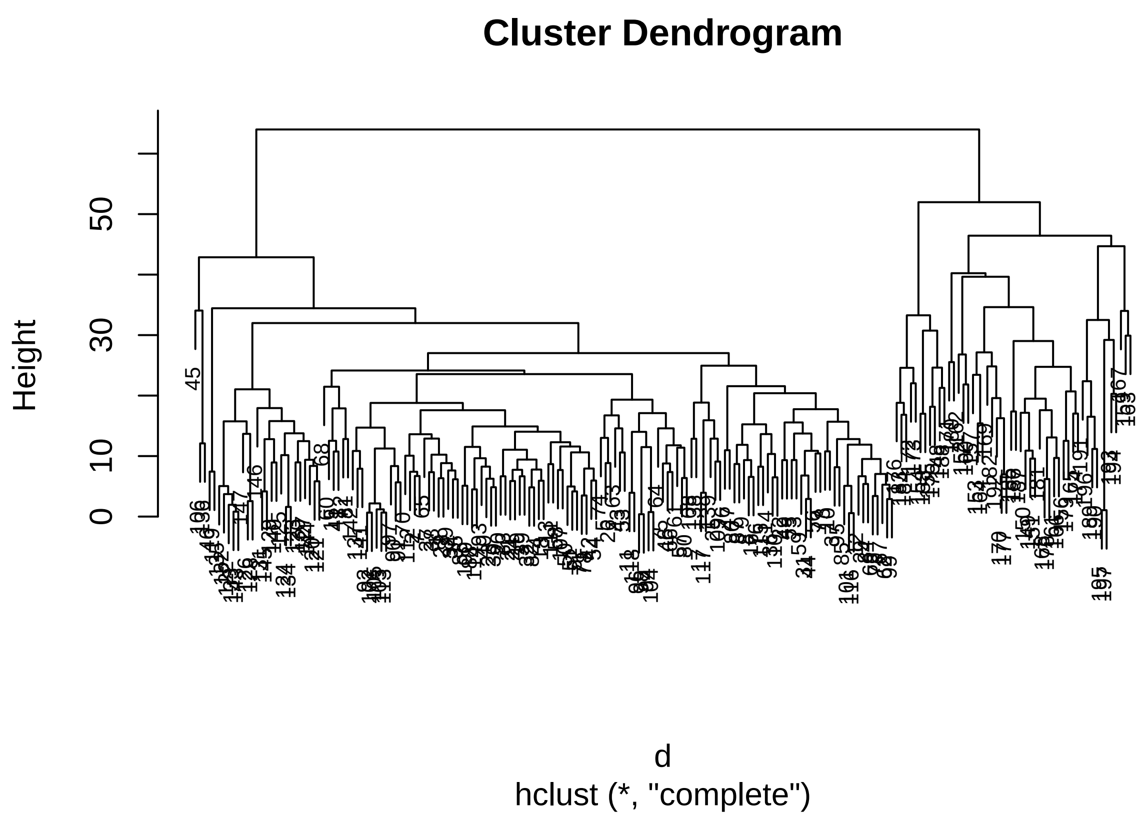
Similar to k-means, we compute first the dissimilarity of observations using distance measures to get the agglomerative coefficient (AC). Using **hclust() function**, we can feed these values and specify the agglomeration method to be used either **"complete", "average", "single", or "ward.D2"**

```
#Dissimilarity matrix
d <- dist(hdf, method = "euclidean")

## Hierarchical clustering using Complete Linkage
h1 <- hclust(d, method = "complete")
```

```r
sub_grp1 <- cutree(h1, k = 8) # Cut tree into 8 groups
table(sub_grp1)  # Number of members in each cluster
```

```
## sub_grp1
##   1   2   3   4   5   6   7   8
## 144   3  11  23   3   3   2   8
```

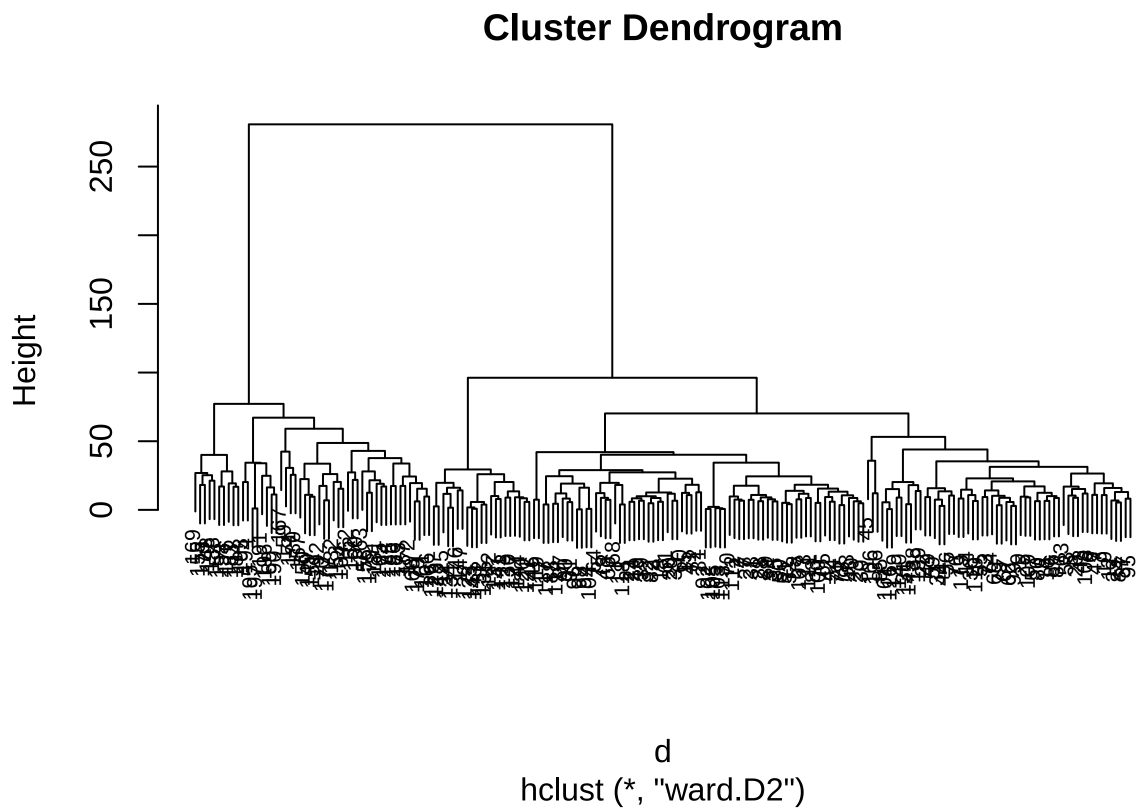
```r
plot(h1, cex=0.7)
```




```r
# Using Ward's method
h2 <- hclust(d, method = "ward.D2" )
sub_grp <- cutree(h2, k = 8)
table(sub_grp)
```

```
## sub_grp
##  1  2  3  4  5  6  7  8
## 70 53  3 21 10 28  4  8
```

```r
plot(h2, cex=0.7)
```



Using Agglomerative Hierarchical Clustering

We can also use the agnes() function as alternative way to get the agglomerative coefficient (AC), which measures the amount of clustering structure found.

```
set.seed(123)
h3 <- agnes(hdf, method = "complete")

#agglomerative coefficient
h3$ac
```

```
## [1] 0.8490083
```

```
# another way to compute coefficient
ac <- function(x) {
  agnes(hdf, method = x)$ac
}

# methods to assess
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# get agglomerative coefficient for each linkage method
purrr::map_dbl(m, ac)
```

```
##   average    single  complete      ward
## 0.7620641 0.7098672 0.8490083 0.9655596
```

Using Divisive Hierarchical Clustering

Aside from agglomeration method, we can also perform divisive hierarchical clustering which **diana() function** allows us to perform. However, there is no agglomerative coefficient to give but divisive coefficient (DC).

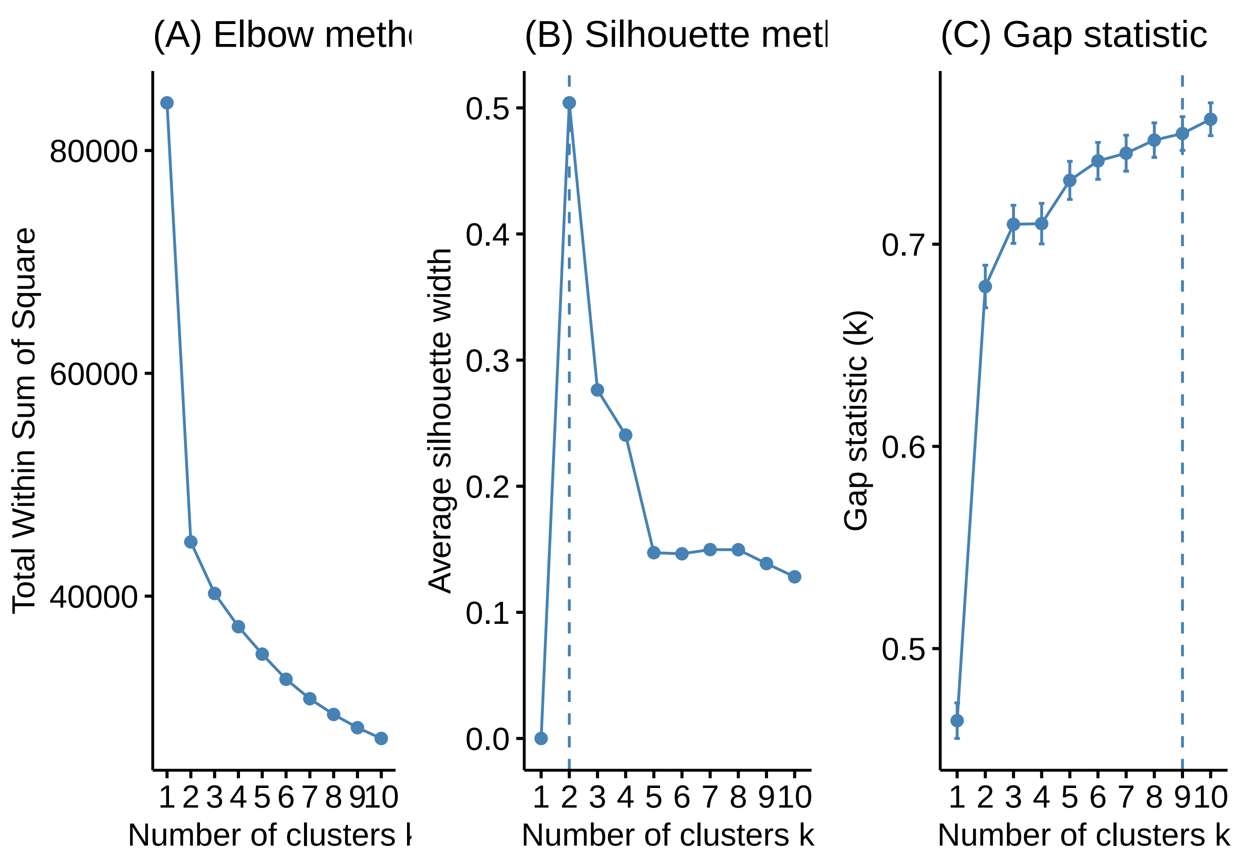
```
h4 <- diana(hdf)

#Divisive coefficient
h4$dc
```

```
## [1] 0.8429389
```

Determining Optimal Clusters

Determining optimal clusters using **Elbow method**, **Silhouette** and **gap_stat** plots. It reveals that in elbow method and silhoutte suggest 2 clusters while 9 clusters in gap statistic.

```
p1 <- fviz_nbclust(hdf, FUN = hcut, method = "wss",
                   k.max = 10) +
  ggtitle("(A) Elbow method")
p2 <- fviz_nbclust(hdf, FUN = hcut, method = "silhouette",
                   k.max = 10) +
  ggtitle("(B) Silhouette method")
p3 <- fviz_nbclust(hdf, FUN = hcut, method = "gap_stat",
                   k.max = 10) +
  ggtitle("(C) Gap statistic")
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```



3. Model-Based Clustering

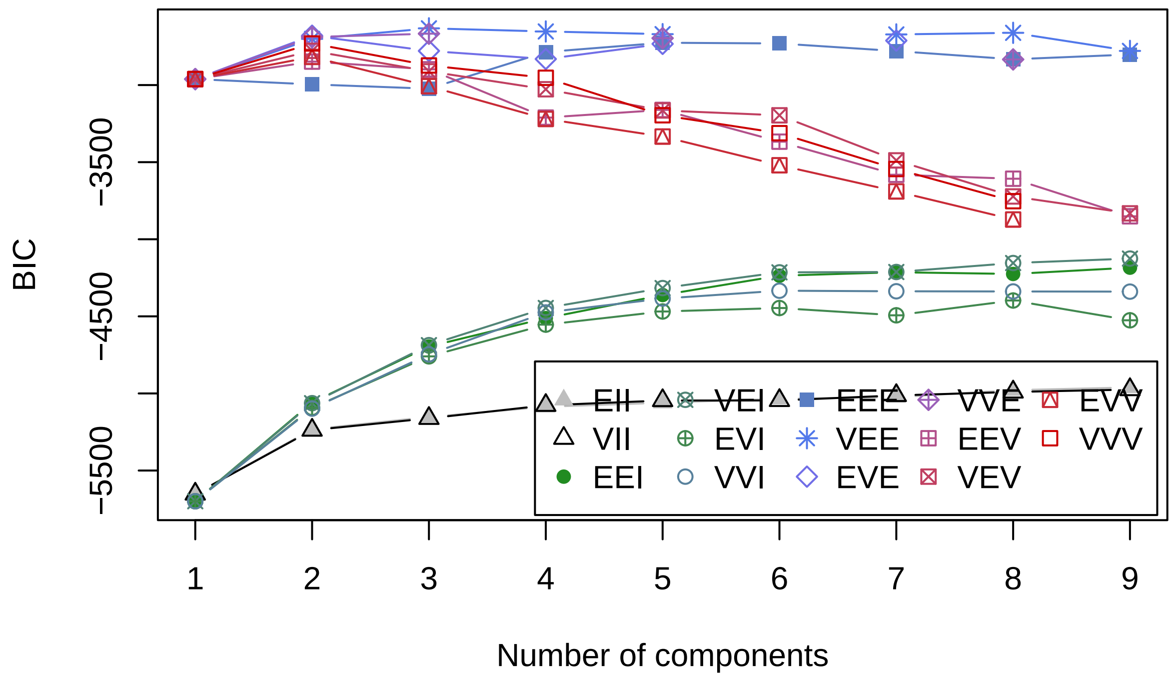
The advantage of model-based clustering over K-means and hierarchical clustering is that it automatically determines the ideal number of clusters. In this clustering, Gaussian mixture models is applied, which are one of the most popular model-based clustering approaches available. Using **df** values in k-means clustering since it is already standardized, we can use **Mclust() function**. Leaving **G = NULL** forces Mclust() to evaluate 1–9 clusters and select the optimal number of components based on BIC.

```
mb <- Mclust(df[,1:10], G=NULL)
summary(mb)
```

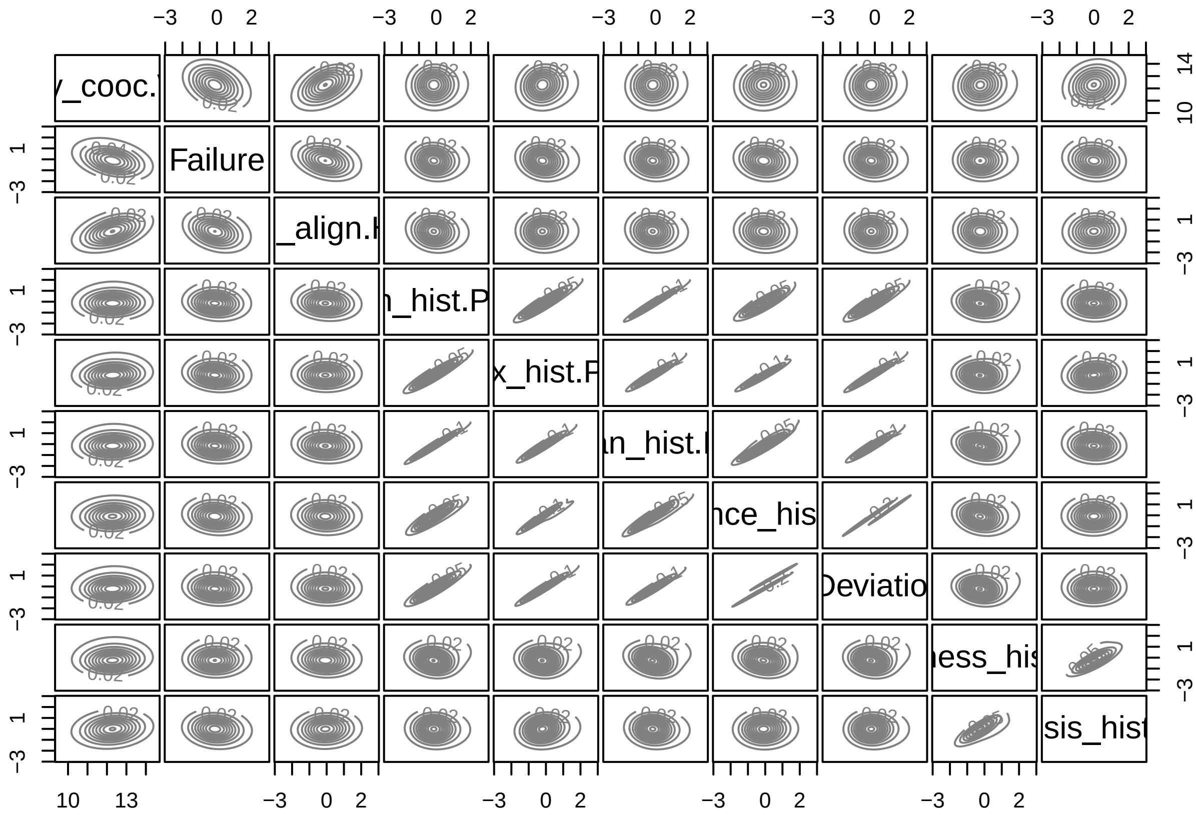
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEE (ellipsoidal, equal shape and orientation) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##           -1081 197 89 -2632.206 -2651.775
##
## Clustering table:
##   1   2   3
## 111  50  36
```

The result shows 3 optimal number of clusters with BIC -2632.206. A negative zone with the highest value indicates the preferred model, In general, the lower the BIC value, the better. Plot the results with BIC, density and uncertainty.

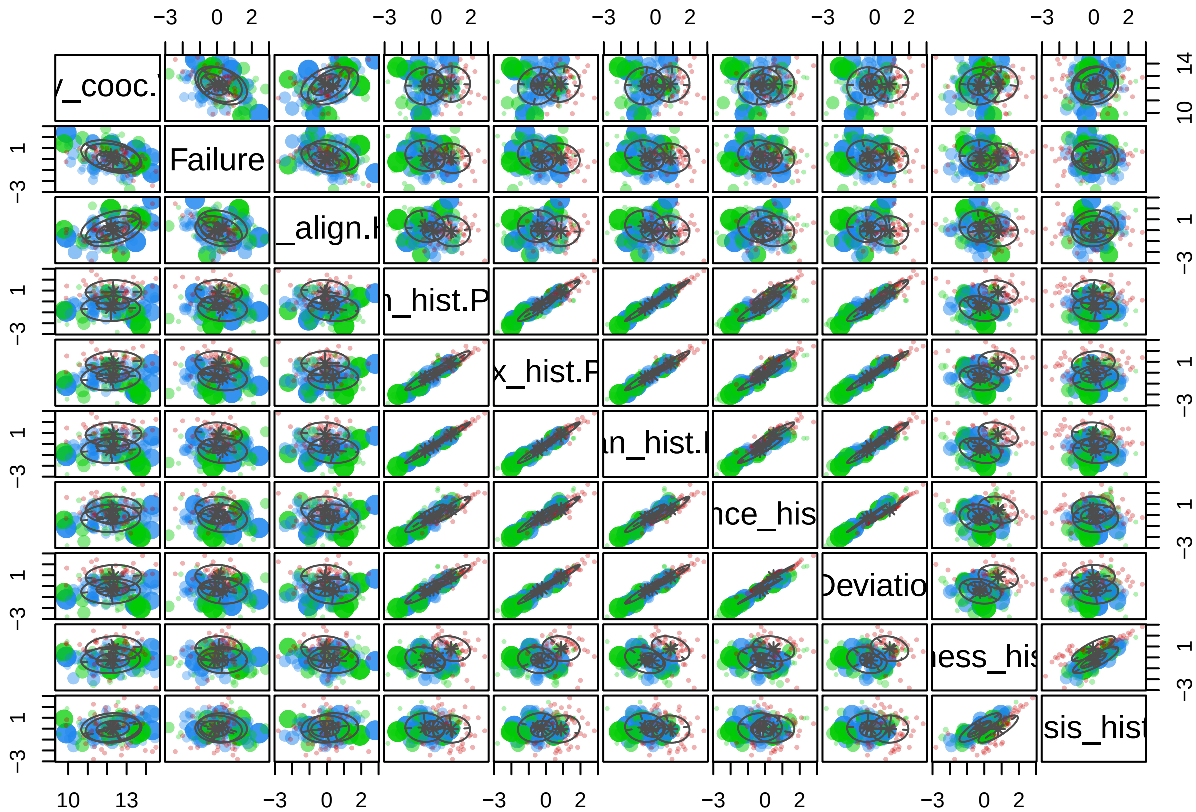
```
legend_args <- list(x = "bottomright", ncol = 5)
plot(mb, what = 'BIC', legendArgs = legend_args)
```




```
plot(mb, what = "density")
```



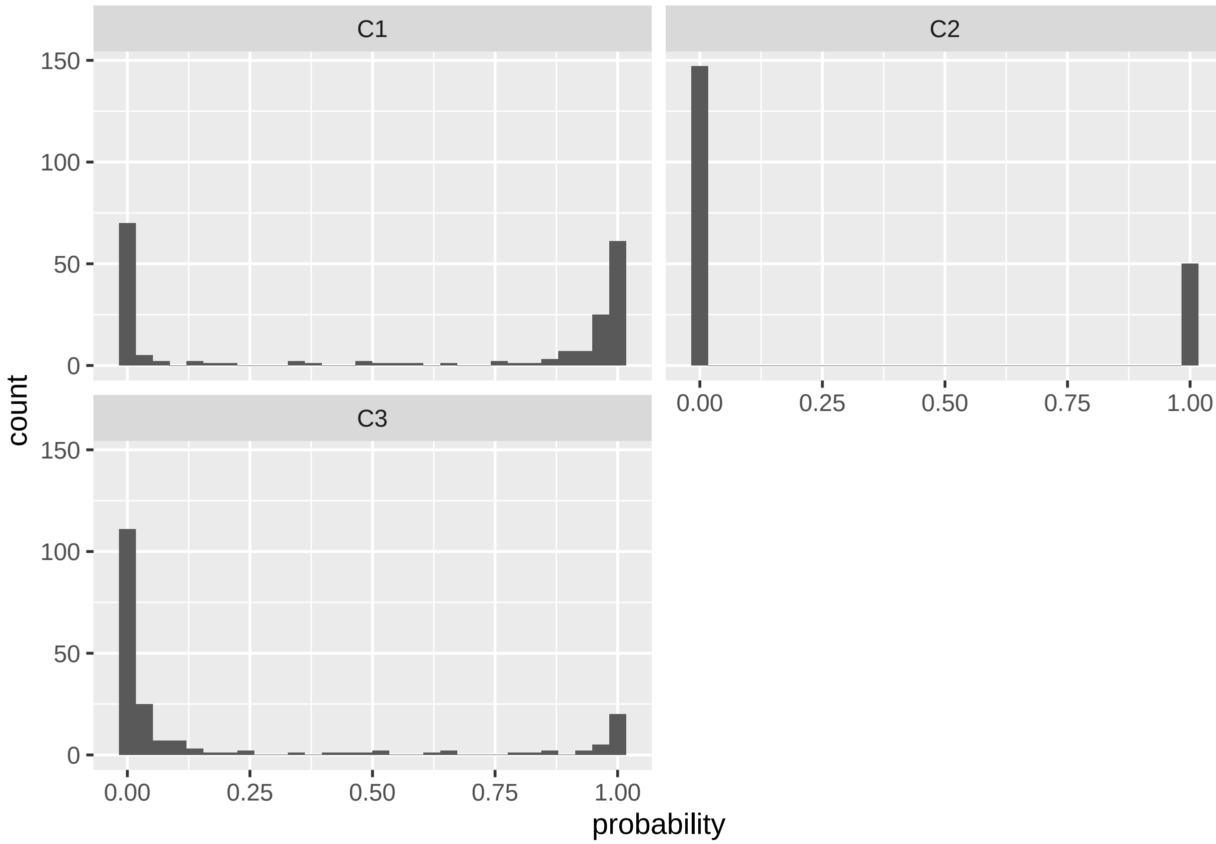
```
plot(mb, what = "uncertainty")
```



Plot the distribution of probabilities for all observations aligning to each of the 3 clusters. As clusters have more observations with middling levels of probability (i.e., 0.25–0.75), their clusters are usually less compact. Therefore, C3 is less compact than other clusters.

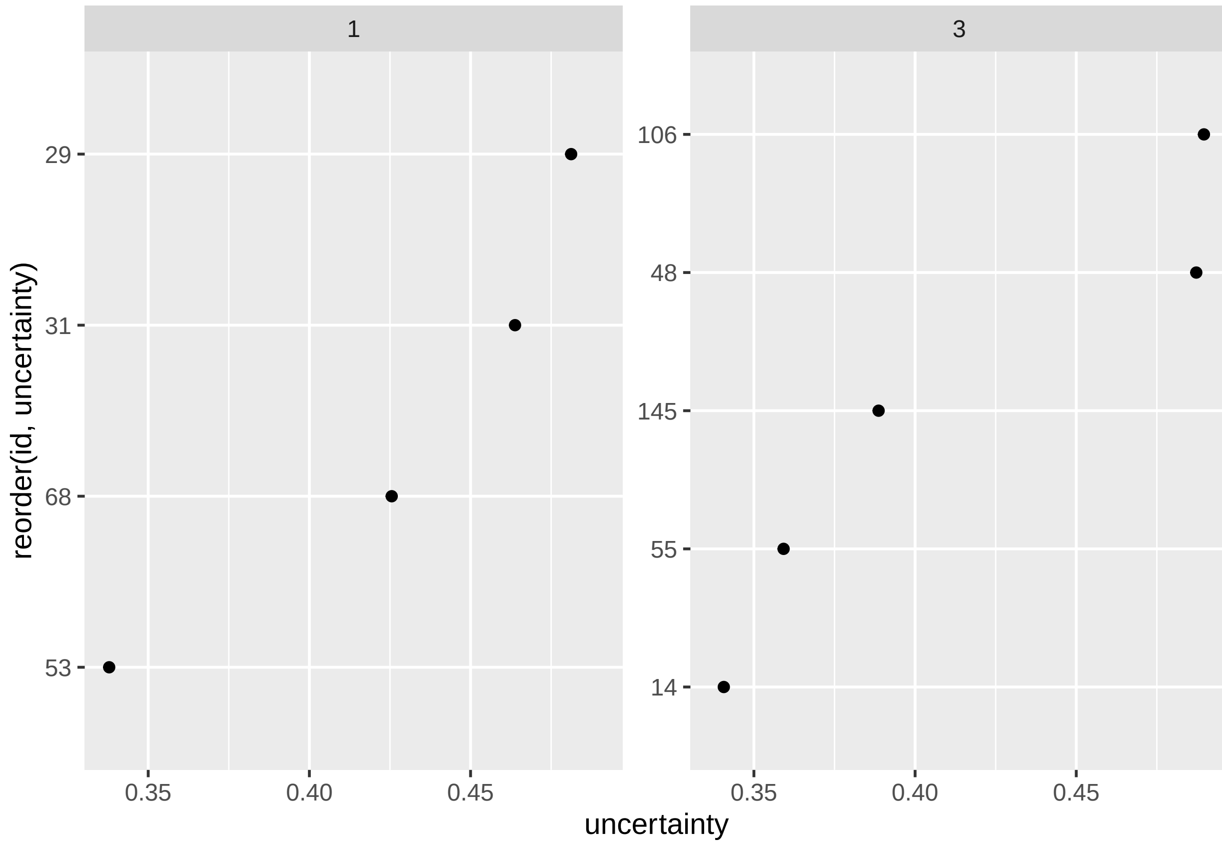
```
probabilities <- mb$z
colnames(probabilities) <- paste0('C', 1:3)

probabilities <- probabilities %>%
  as.data.frame() %>%
  mutate(id = row_number()) %>%
  tidyr::gather(cluster, probability, -id)

ggplot(probabilities, aes(probability)) +
  geom_histogram() +
  facet_wrap(~ cluster, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



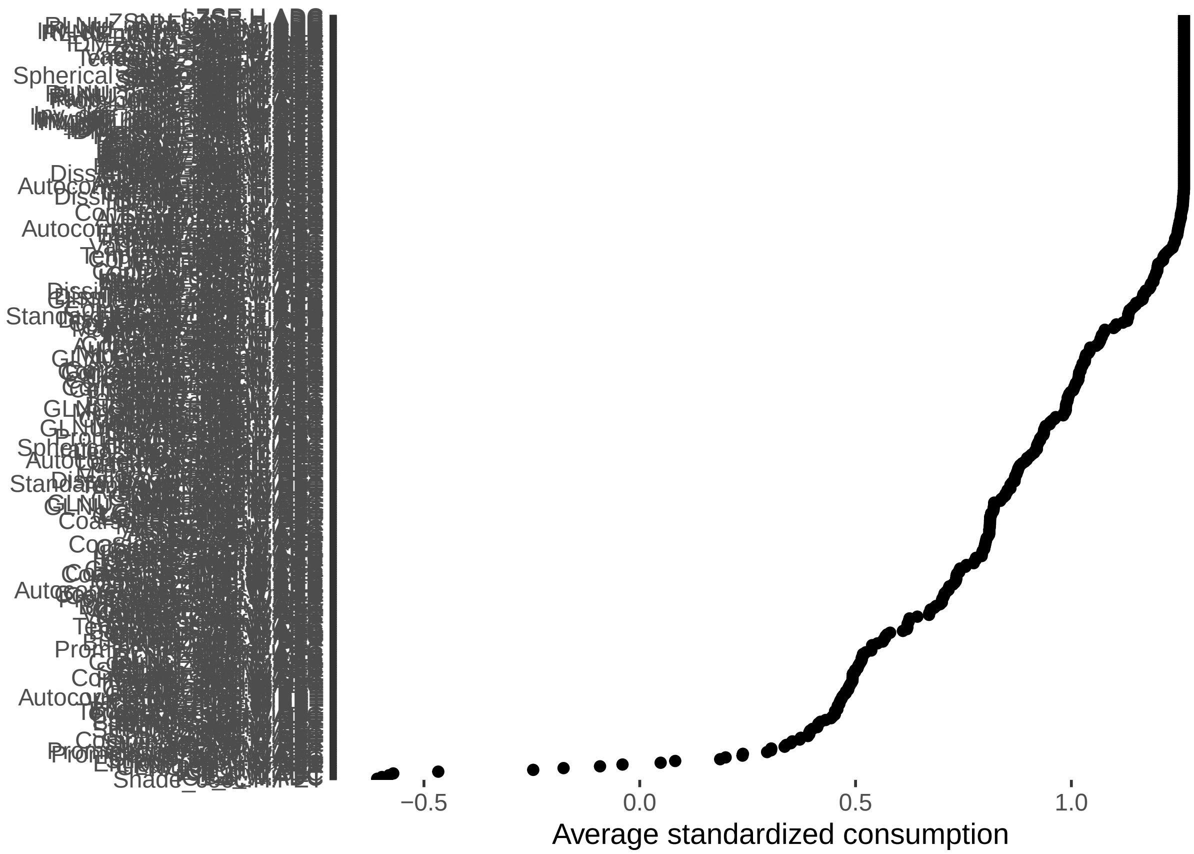
Plot the observations that are aligned to each cluster but their uncertainty of membership is greater than 0.25.

```
uncertainty <- data.frame(
  id = 1:nrow(df),
  cluster = mb$classification,
  uncertainty = mb$uncertainty
)

uncertainty %>%
  group_by(cluster) %>%
  filter(uncertainty > 0.25) %>%
  ggplot(aes(uncertainty, reorder(id, uncertainty))) +
  geom_point() +
  facet_wrap(~ cluster, scales = 'free_y', nrow = 1)
```



Plot the average standardized consumption for cluster 2 observations compared to all observations.

```
cluster2 <- df %>%
  scale() %>%
  as.data.frame() %>%
  mutate(cluster = mb$classification) %>%
  filter(cluster == 2) %>%
  select(-cluster)

cluster2 %>%
  tidyr::gather(product, std_count) %>%
  group_by(product) %>%
  summarize(avg = mean(std_count)) %>%
  ggplot(aes(avg, reorder(product, avg))) +
  geom_point() +
  labs(x = "Average standardized consumption", y = NULL)
```



−0.5
0.0
0.5
1.0
Average standardized consumption