Clustering

```
library(mlbench)
data(Glass)
library(factoextra)
```

```
## Loading required package: ggplot2
```

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

```
library(dplyr)
```

```
##
## 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(tidyverse)
```

```
## —— Attaching packages ————————————————————————————————————————————————————————————————————————————————
———————————————————————————————————————————————————————————————————————————————————— tidyverse 1.3.0 ——
```

```
## ✓ tibble  3.0.0     ✓ purrr   0.3.4
## ✓ tidyr   1.0.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## —— Conflicts —————————————————————————————————————————————————————————————————————————————————————————
—————————————————————————————————————————————————————————————————————————————— tidyverse_conflicts() ——
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

K-means

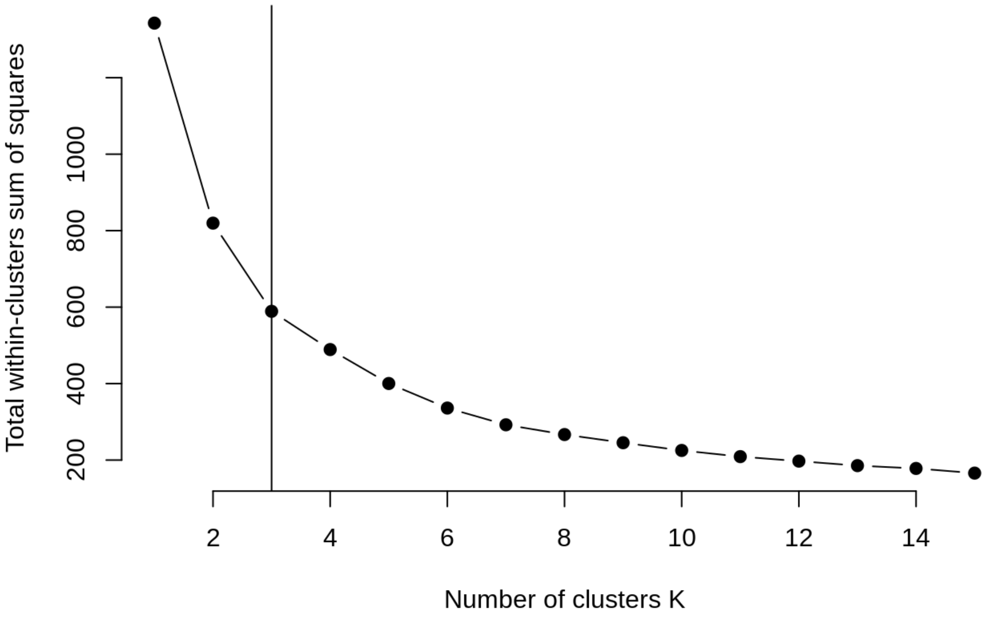
```
Glass$Type2 <- as.factor(c(rep('Window', 163), rep('Non-Window', 51)))
glass <- Glass[,-c(10,11)]
```

```
set.seed(42)

wss <- function(k) {
  kmeans(glass, k, nstart = 25)$tot.withinss
}

k.values <- 1:15
wss_values <-map(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
abline(v = 3)
```

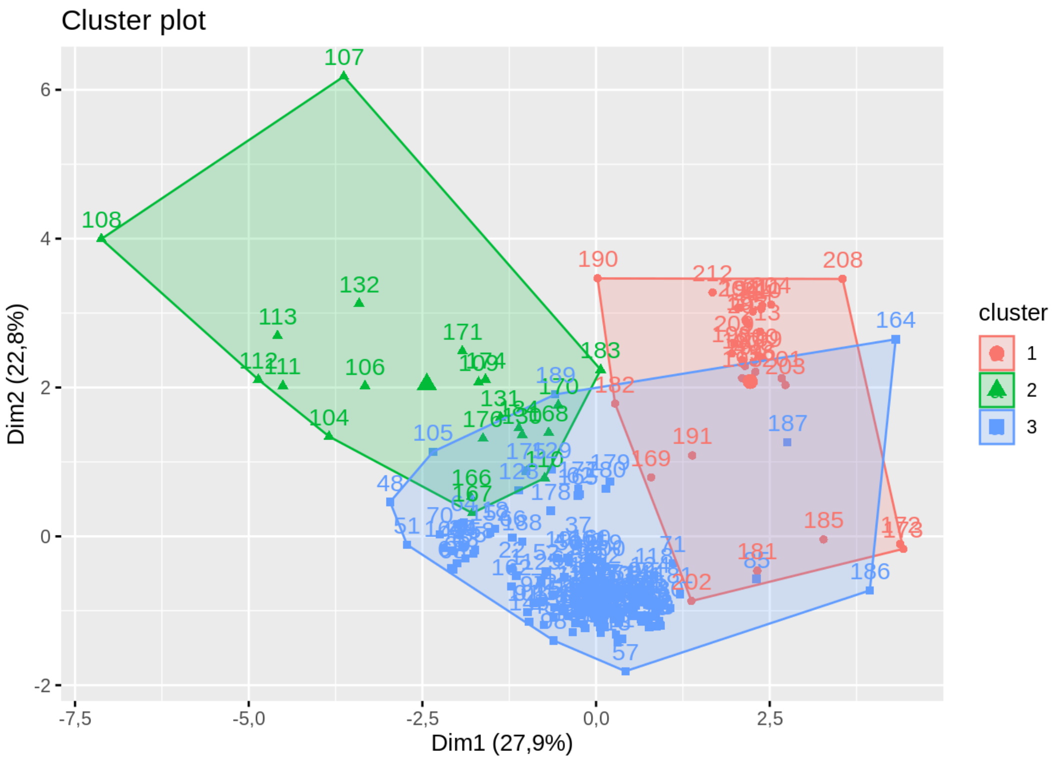


```
set.seed(42)
km.res <- kmeans(glass,3, nstart = 25)
```

```
km.res
```

```
## K-means clustering with 3 clusters of sizes 31, 21, 162
##
## Cluster means:
##         RI       Na        Mg       Al       Si         K        Ca         Ba
## 1 1,516358 14,45677 0,1977419 2,120968 73,12355 0,5883871  8,538387 0,88193548
## 2 1,523548 12,84524 0,4490476 1,305238 72,40524 0,2542857 12,383333 0,15000000
## 3 1,518078 13,28006 3,4501852 1,333642 72,59235 0,5110494  8,592901 0,04302469
##           Fe
## 1 0,01258065
## 2 0,07142857
## 3 0,06364198
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   3   3   3   2   3   2   2   2   2   2   2   2   2   3   3   3   3   3   3   3
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   3   3   3   3   3   3   3   3   3   2   2   2   3   3   3   3   3   3   3   3
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   3   3   3   3   3   2   2   2   1   2   2   1   1   2   3   2   3   3   3   3
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   1   1   2   2   1   3   3   3   3   1   1   1   1   1   1   1   1   1   1   1
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##
## Within cluster sum of squares by cluster:
## [1] 194,4349 138,9466 255,6500
##  (between_SS / total_SS =  56,1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
fviz_cluster(km.res, glass)
```



```
table(Glass$Type2, km.res$cluster)
```

```
##
##              1   2   3
##   Non-Window  31   9  11
##   Window       0  12 151
```

```
table('Glass_Type' = Glass$Type, 'Clusters' = km.res$cluster)
```

```
##           Clusters
## Glass_Type  1  2  3
##          1  0  0 70
##          2  0 12 64
##          3  0  0 17
##          5  3  7  3
##          6  3  2  4
##          7 25  0  4
```

```
Glass$K_means <- km.res$cluster
```

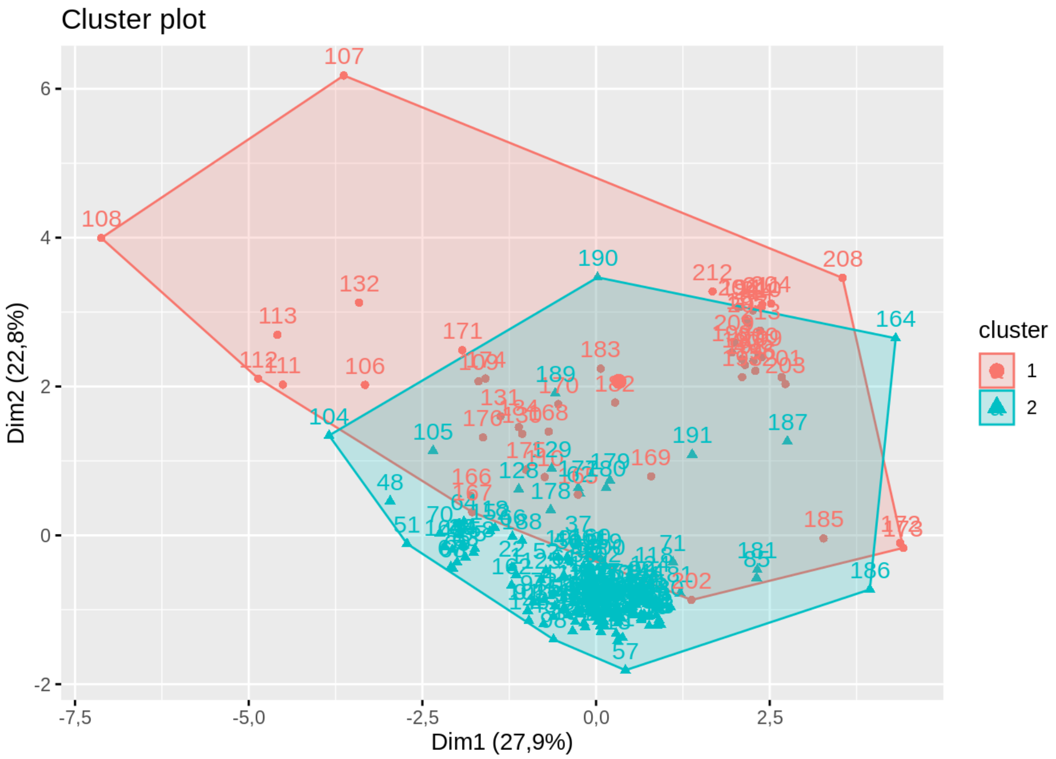
#K-means 2 and 6

```
set.seed(42)
km.res <- kmeans(glass,2, nstart = 25)
```

```
table(Glass$Type2, km.res$cluster)
```

```
##
##              1   2
##   Non-Window  39  12
##   Window      11 152
```

```
fviz_cluster(km.res, glass)
```

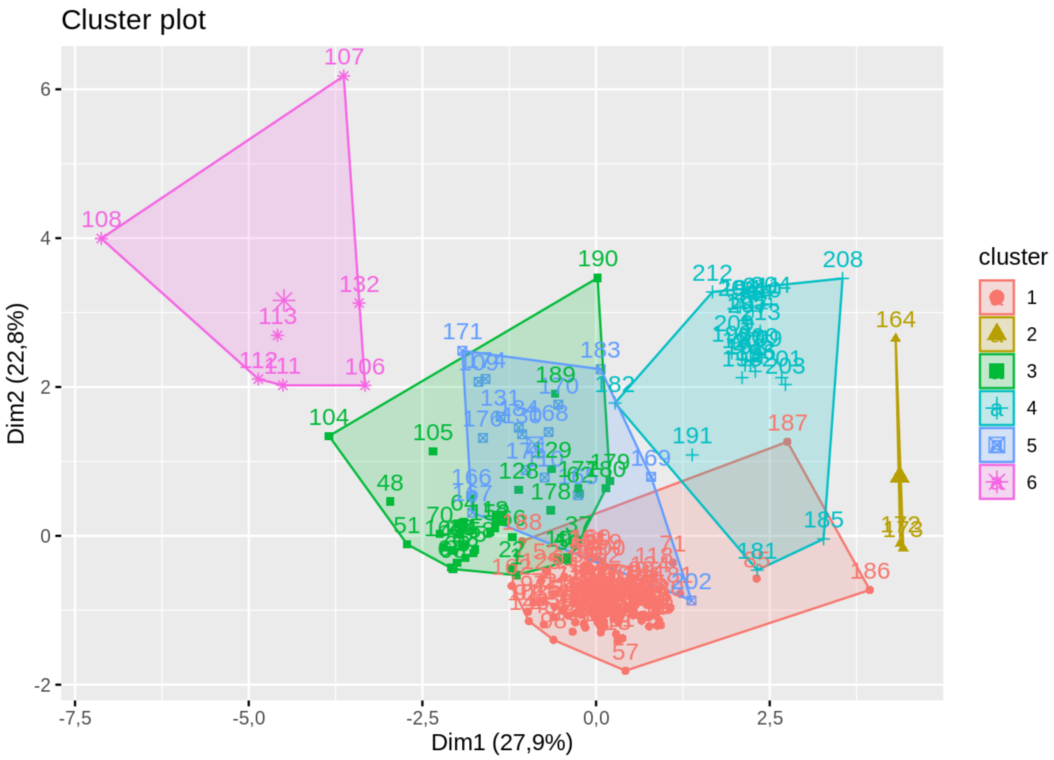


```
set.seed(42)
km.res <- kmeans(glass, 6, nstart = 25)
```

```
table('Glass_Type' = Glass$Type, 'Cluster' = km.res$cluster)
```

```
##           Cluster
## Glass_Type  1  2  3  4  5  6
##          1 48  0 22  0  0  0
##          2 61  0  4  0  4  7
##          3 14  0  3  0  0  0
##          5  0  3  0  0 10  0
##          6  0  0  4  3  2  0
##          7  3  0  2 23  1  0
```

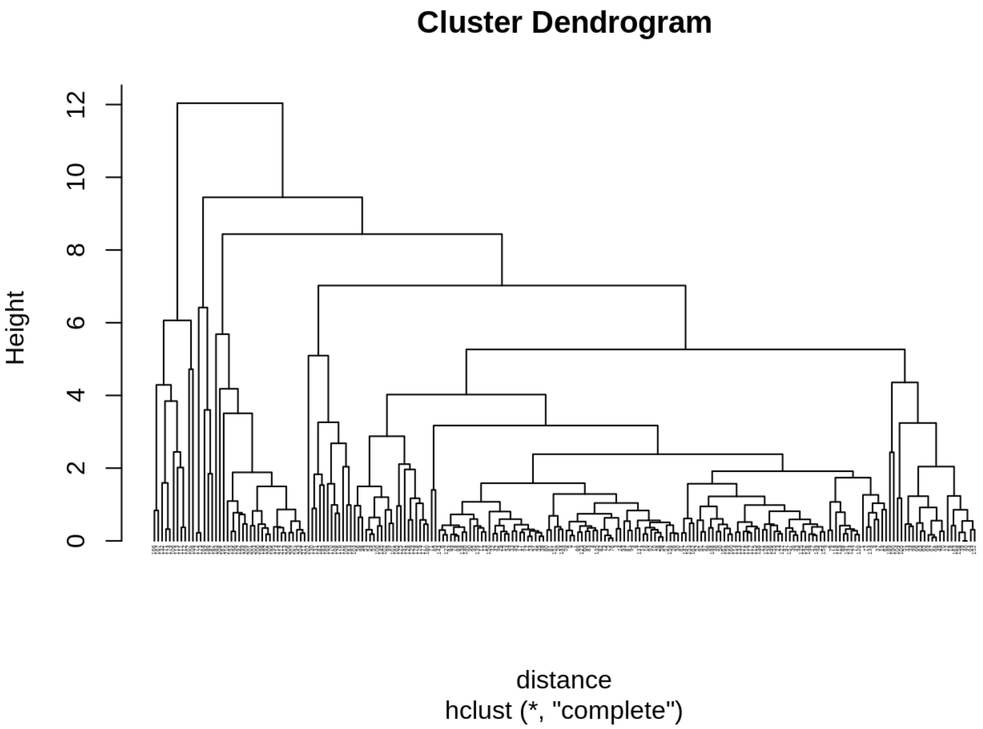
```
fviz_cluster(km.res, glass)
```



Hierarchical clustering

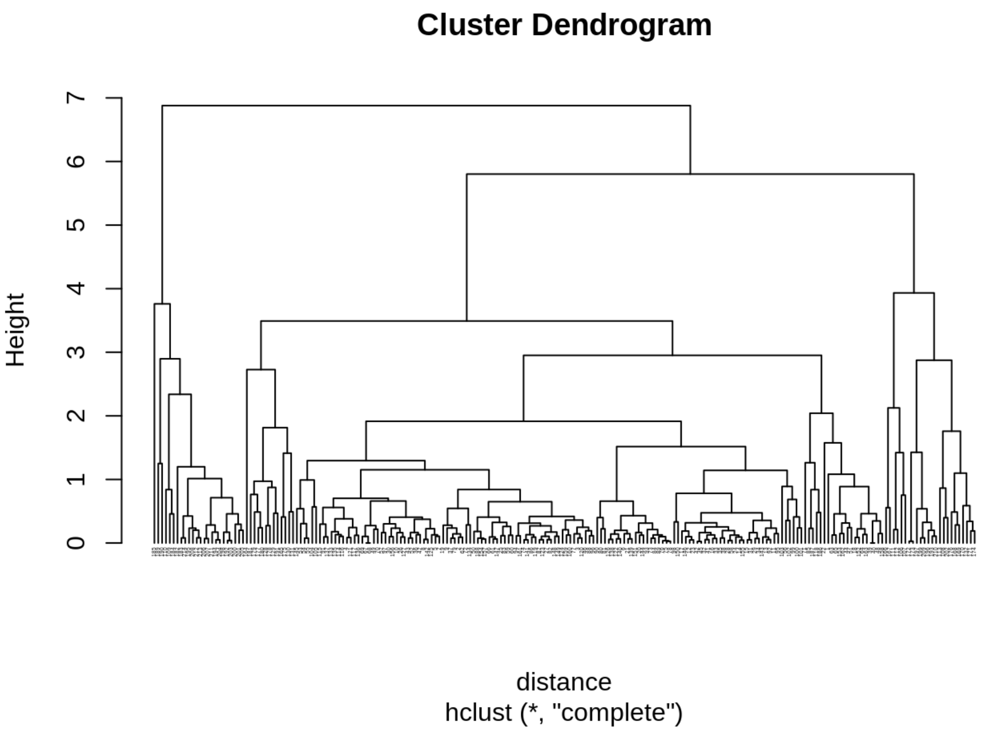
```
distance <- dist(glass , method = "euclidean")
hc_comp <- hclust(distance, method = "complete")
```

```
plot(hc_comp, cex = 0.2, hang = -1)
```

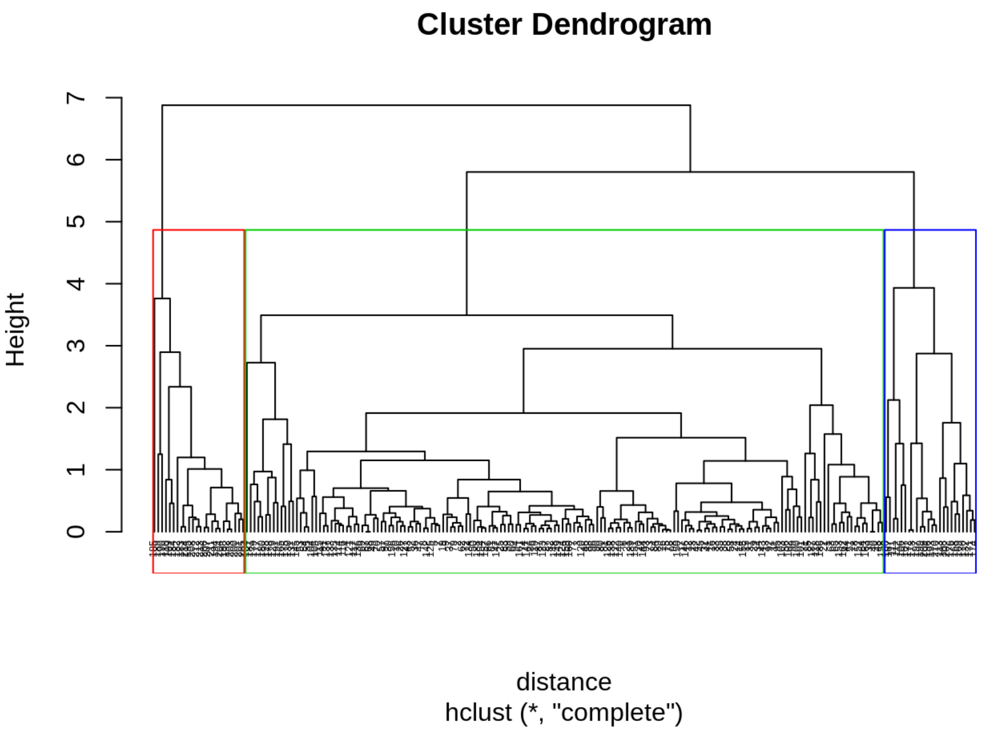


```
#Al,Mg, Ba
glass <- glass[,c(2,3,4)]
```

```
distance <- dist(glass , method = "euclidean")
hc_comp <- hclust(distance, method = "complete")
plot(hc_comp, cex = 0.2, hang = -1)
```



```
plot(hc_comp, cex = 0.4, hang = -1)
rect.hclust(hc_comp, k = 3, border = 2:5)
```



```
Glass$HC <- cutree(hc_comp, 3)
```

```
table('K_means' = Glass$K_means,'HC' = Glass$HC)
```

```
##        HC
## K_means   1   2   3
##       1   2  10  19
##       2   3  14   4
##       3 161   0   1
```

PCA

```
apply(Glass[,1:9], 2, mean)
```

```
##          RI          Na          Mg          Al          Si           K 
##  1,51836542 13,40785047  2,68453271  1,44490654 72,65093458  0,49705607 
##          Ca          Ba          Fe 
##  8,95696262  0,17504673  0,05700935 
```

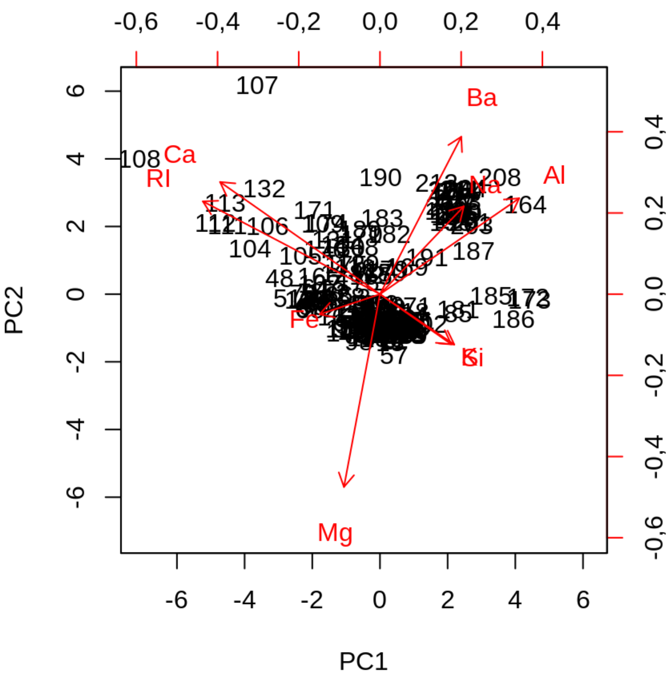
```
apply(Glass[,1:9], 2, sd)
```

```
##          RI          Na          Mg          Al          Si           K 
## 0,003036864 0,816603556 1,442407845 0,499269646 0,774545795 0,652191846 
##          Ca          Ba          Fe 
## 1,423153487 0,497219261 0,097438701 
```

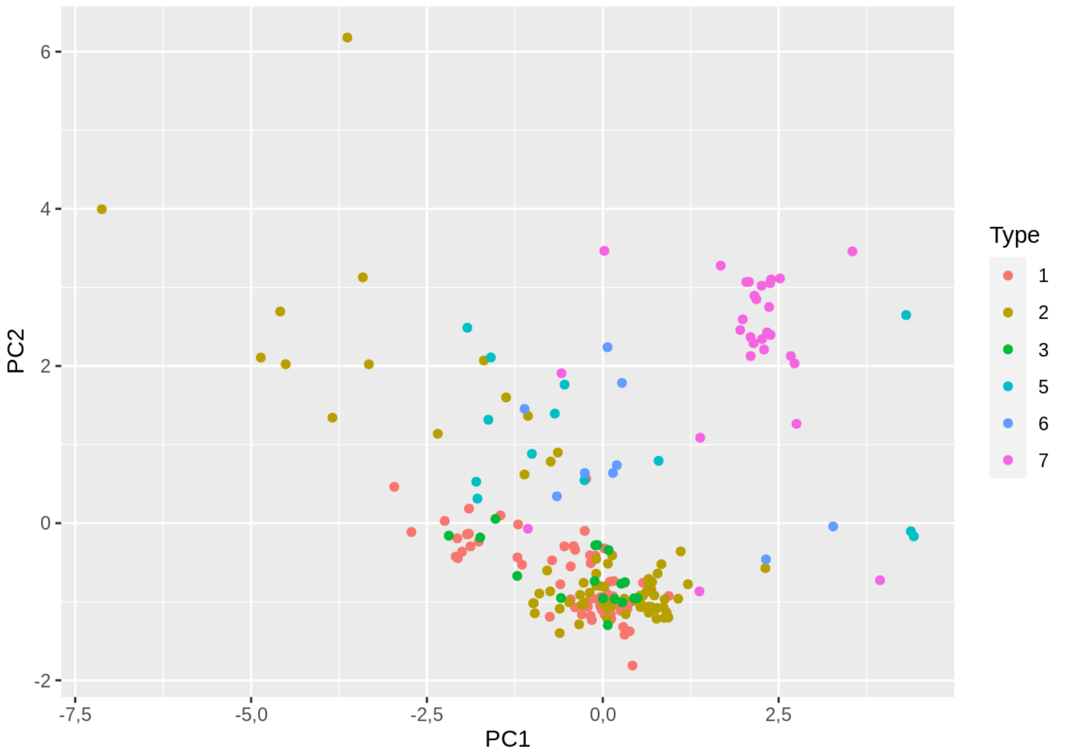
```
prc <- prcomp(x = Glass[,1:9], scale = TRUE)
prc
```

```
## Standard deviations (1, .., p=9):
## [1] 1,58466518 1,43180731 1,18526115 1,07604017 0,95603465 0,72638502 0,60741950
## [8] 0,25269141 0,04011007
## 
## Rotation (n x k) = (9 x 9):
##           PC1         PC2           PC3         PC4          PC5         PC6
## RI -0,5451766  0,28568318 -0,0869108293 -0,14738099  0,073542700 -0,11528772
## Na  0,2581256  0,27035007  0,3849196197 -0,49124204 -0,153683304  0,55811757
## Mg -0,1108810 -0,59355826 -0,0084179590 -0,37878577 -0,123509124 -0,30818598
## Al  0,4287086  0,29521154 -0,3292371183  0,13750592 -0,014108879  0,01885731
## Si  0,2288364 -0,15509891  0,4587088382  0,65253771 -0,008500117 -0,08609797
## K   0,2193440 -0,15397013 -0,6625741197  0,03853544  0,307039842  0,24363237
## Ca -0,4923061  0,34537980  0,0009847321  0,27644322  0,188187742  0,14866937
## Ba  0,2503751  0,48470218 -0,0740547309 -0,13317545 -0,251334261 -0,65721884
## Fe -0,1858415 -0,06203879 -0,2844505524  0,23049202 -0,873264047  0,24304431
##            PC7         PC8        PC9
## RI -0,08186724 -0,75221590 -0,02573194
## Na -0,14858006 -0,12769315  0,31193718
## Mg  0,20604537 -0,07689061  0,57727335
## Al  0,69923557 -0,27444105  0,19222686
## Si -0,21606658 -0,37992298  0,29807321
## K  -0,50412141 -0,10981168  0,26050863
## Ca  0,09913463  0,39870468  0,57932321
## Ba -0,35178255  0,14493235  0,19822820
## Fe -0,07372136 -0,01627141  0,01466944
```

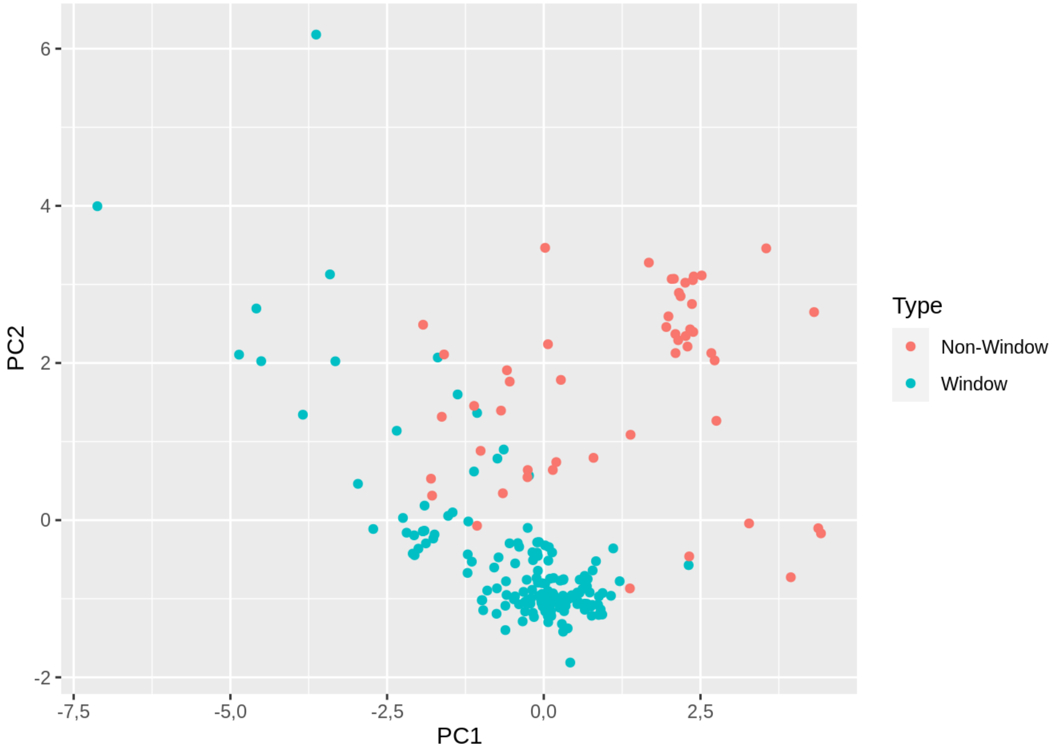
```
biplot(prc, scale = 0)
```



```
prc_adj <- data.frame(prc$x, Type = Glass$Type)
ggplot(data = prc_adj, aes(x = PC1, y = PC2, color=Type)) +
  geom_point()
```



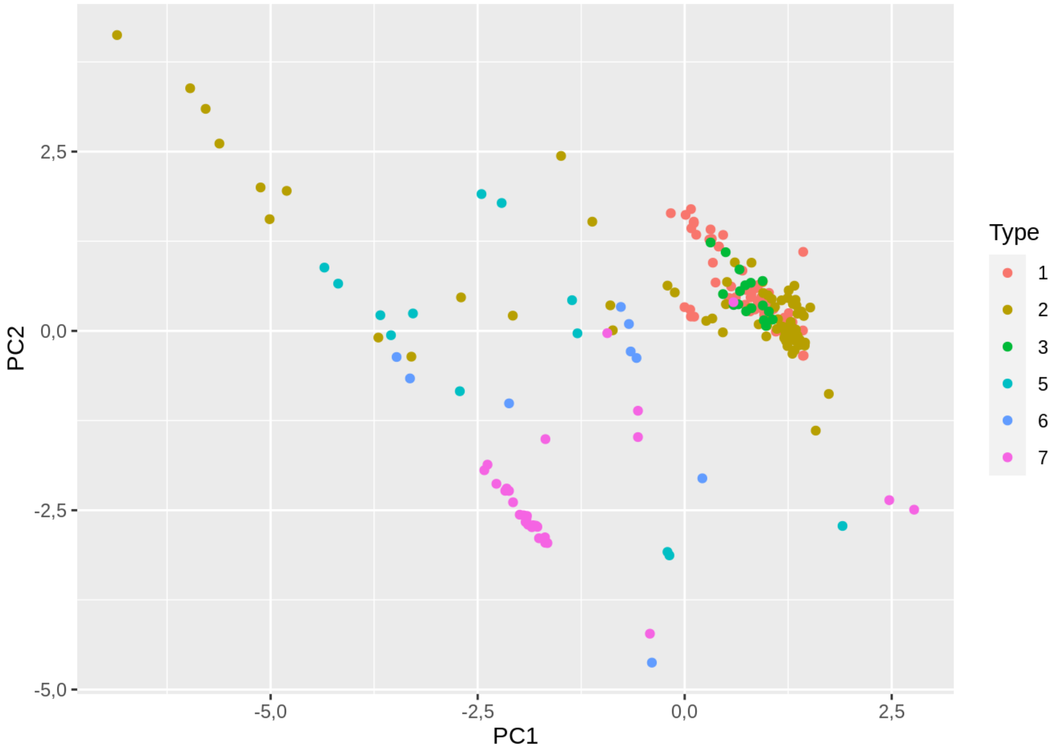
```
prc_adj <- data.frame(prc$x, Type = Glass$Type2)
ggplot(data = prc_adj, aes(x = PC1, y = PC2, color=Type)) +
  geom_point()
```



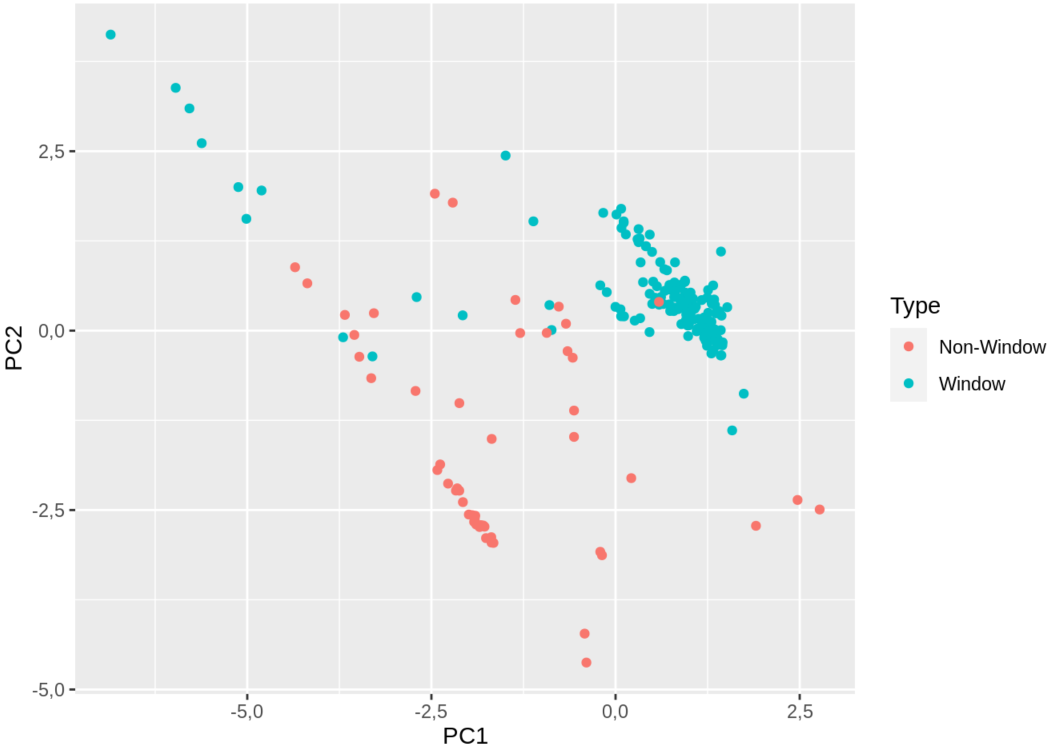
#Unscaled variant

```
prc_un <- prcomp(x = Glass[,1:9], scale = FALSE)
```

```
prc_adj_un <- data.frame(prc_un$x, Type = Glass$Type)
ggplot(data = prc_adj_un, aes(x = PC1, y = PC2, color=Type)) +
  geom_point()
```



```
prc_adj_un <- data.frame(prc_un$x, Type = Glass$Type2)
ggplot(data = prc_adj_un, aes(x = PC1, y = PC2, color=Type)) +
  geom_point()
```



PVE

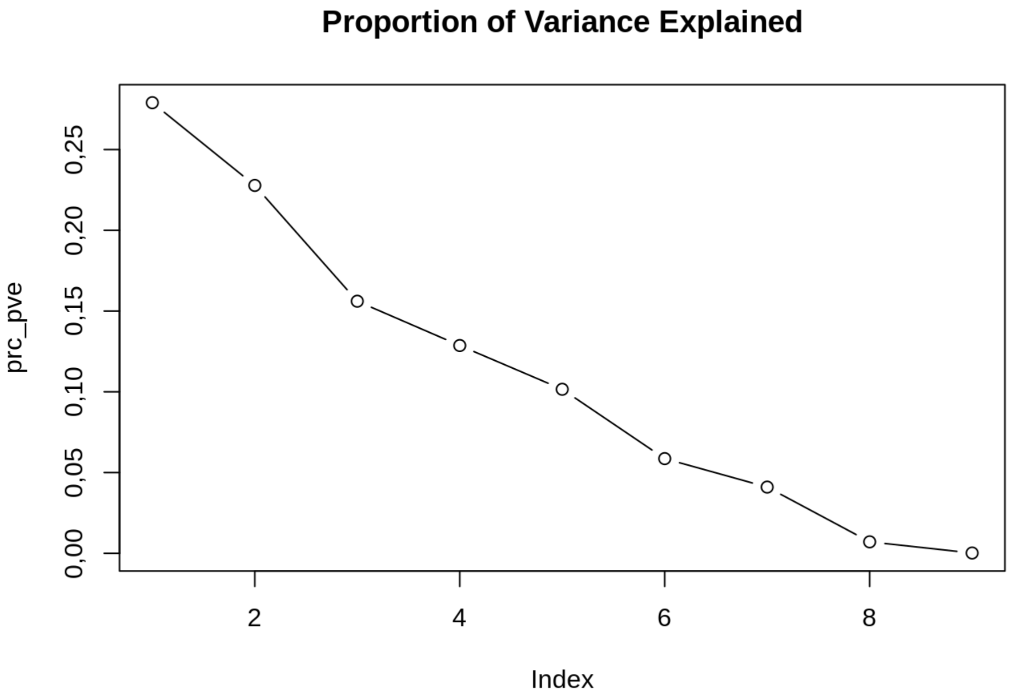
```
prc_var <- prc$sdev^2
prc_pve <- prc_var / sum(prc_var)
prc_pve
```

```
## [1] 0,2790181918 0,2277857983 0,1560937771 0,1286513829 0,1015558052
## [6] 0,0586261325 0,0409953826 0,0070947720 0,0001787575
```

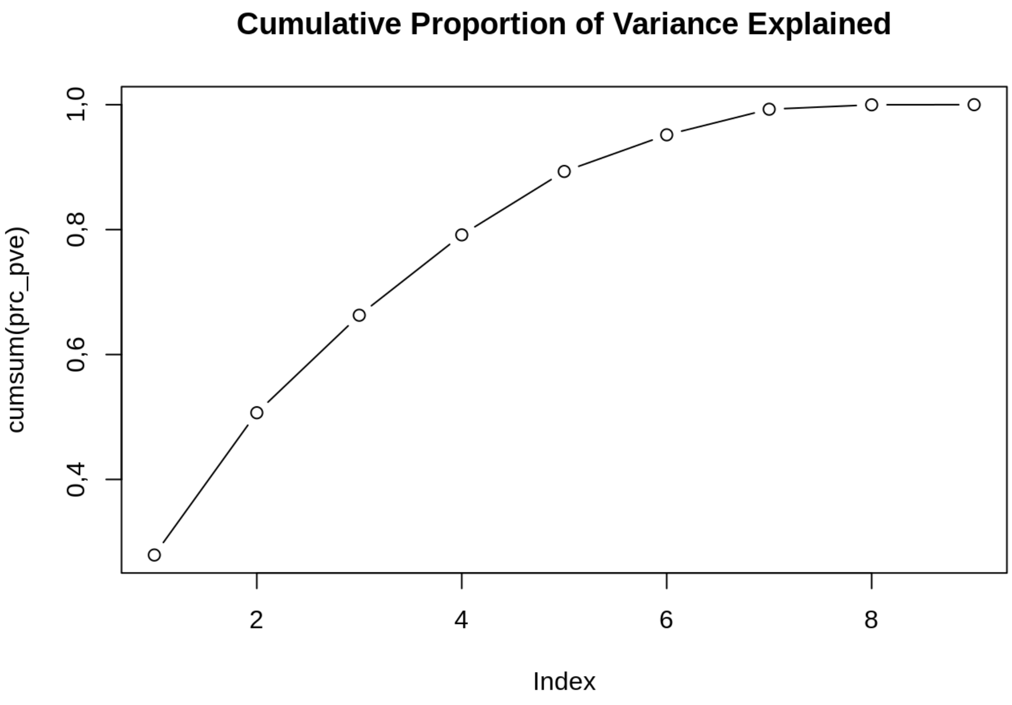
```
cumsum(prc_pve)
```

```
## [1] 0,2790182 0,5068040 0,6628978 0,7915492 0,8931050 0,9517311 0,9927265
## [8] 0,9998212 1,0000000
```

```
plot(prc_pve, type = "b", main = "Proportion of Variance Explained")
```



```
plot(cumsum(prc_pve), type = "b", main = "Cumulative Proportion of Variance Explained")
```



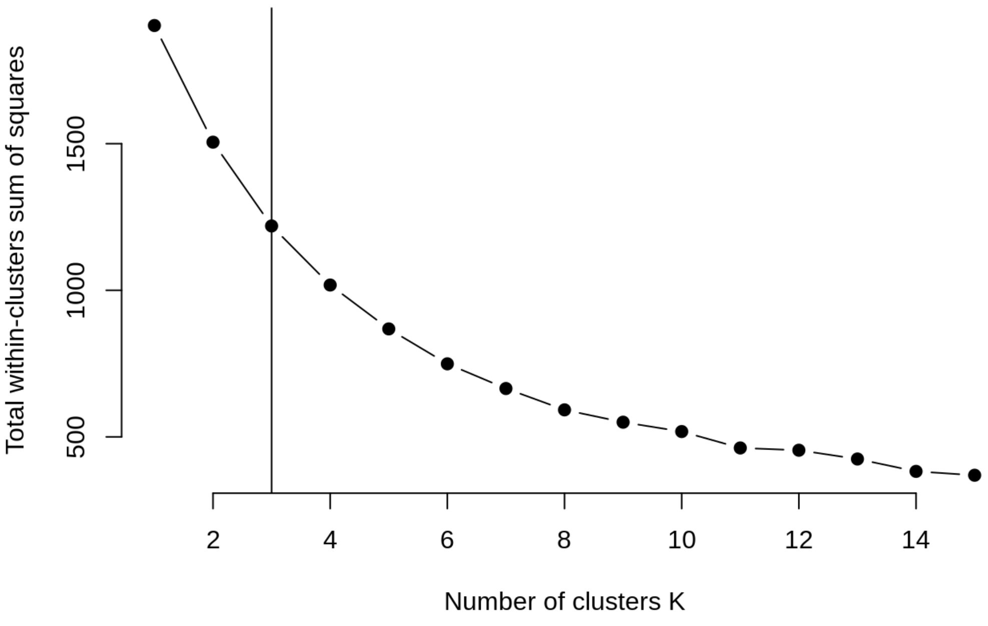
K-means

```
pca_res <- prc$x
prc <- prc$x[,1:7]
set.seed(42)

wss <- function(k) {
  kmeans(prc, k, nstart = 25)$tot.withinss
}

k.values <- 1:15
wss_values <-map(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
abline(v = 3)
```

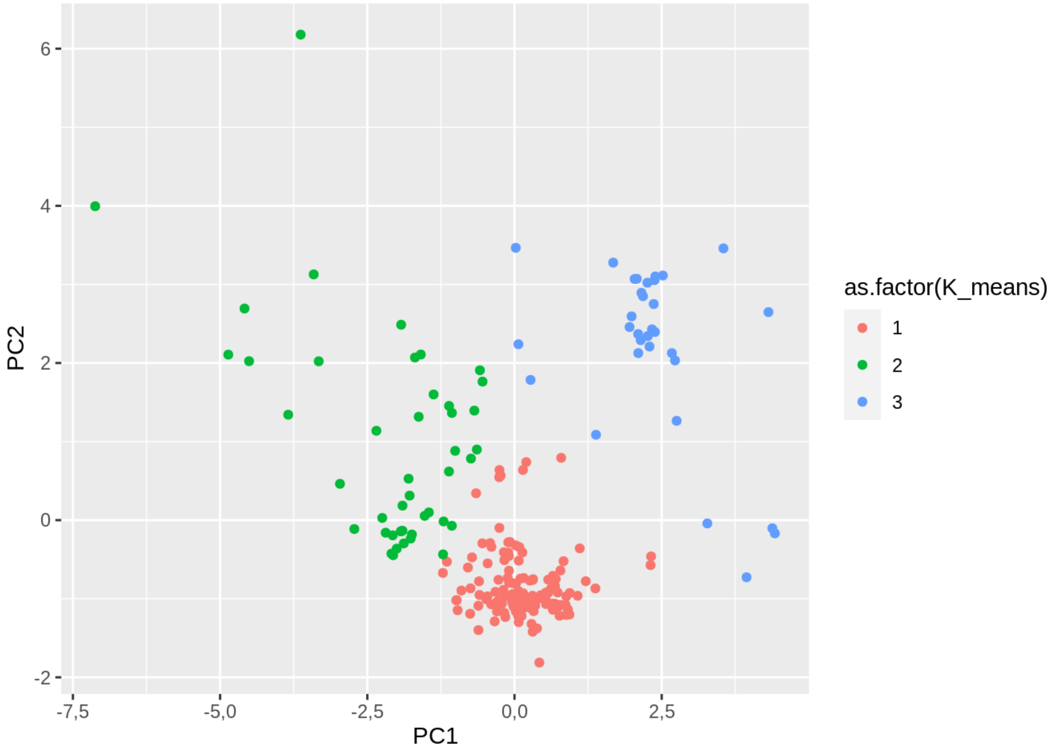


```
set.seed(42)
km.res <- kmeans(prc,3, nstart = 25)
```

```
km.res
```

```
## K-means clustering with 3 clusters of sizes 137, 45, 32
## 
## Cluster means:
##          PC1        PC2          PC3         PC4         PC5        PC6
## 1  0,1414358 -0,8325299  0,008226861  0,07806163 -0,01529622 -0,1052881
## 2 -2,1058788  0,9684670 -0,082629603 -0,17190774  0,14847033  0,2518977
## 3  2,3558700  2,2023619  0,080976631 -0,09245609 -0,14329946  0,0965335
##           PC7
## 1  0,07344303
## 2 -0,12911682
## 3 -0,13285745
## 
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   1   1 
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 
##   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   2 
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
##   1   1   1   2   1   1   1   2   2   1   2   1   1   1   1   1   1   1   1   1 
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 
##   1   1   2   2   2   2   2   2   2   2   1   1   1   1   1   1   1   1   1   1 
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 
##   1   1   1   2   2   2   2   2   2   2   2   2   2   1   1   1   1   1   1   1 
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 
##   1   1   1   1   1   1   1   2   2   2   2   2   1   1   1   1   1   1   1   1 
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 
##   1   1   1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   2   1   1 
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 
##   1   1   2   3   1   2   2   2   1   2   2   3   3   2   2   2   1   1   1   1 
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 
##   1   3   3   2   3   3   3   3   2   2   3   3   3   3   3   3   3   3   3   3 
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 
##   3   1   3   3   3   3   3   3   3   3   3   3   3   3 
## 
## Within cluster sum of squares by cluster:
## [1] 331,3635 455,9302 432,1088
##  (between_SS / total_SS =  35,9 %)
## 
## Available components:
## 
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
df <- as.data.frame(prc)
df$K_means <- km.res$cluster
ggplot(df,aes(x=PC1,y=PC2,color= as.factor(K_means))) + geom_point()
```



```
table('Glass_Type' = Glass$Type, 'Cluster' = km.res$cluster)
```

```
##           Cluster
## Glass_Type  1  2  3
##          1 54 16  0
##          2 61 15  0
##          3 14  3  0
##          5  2  8  3
##          6  5  1  3
##          7  1  2 26
```

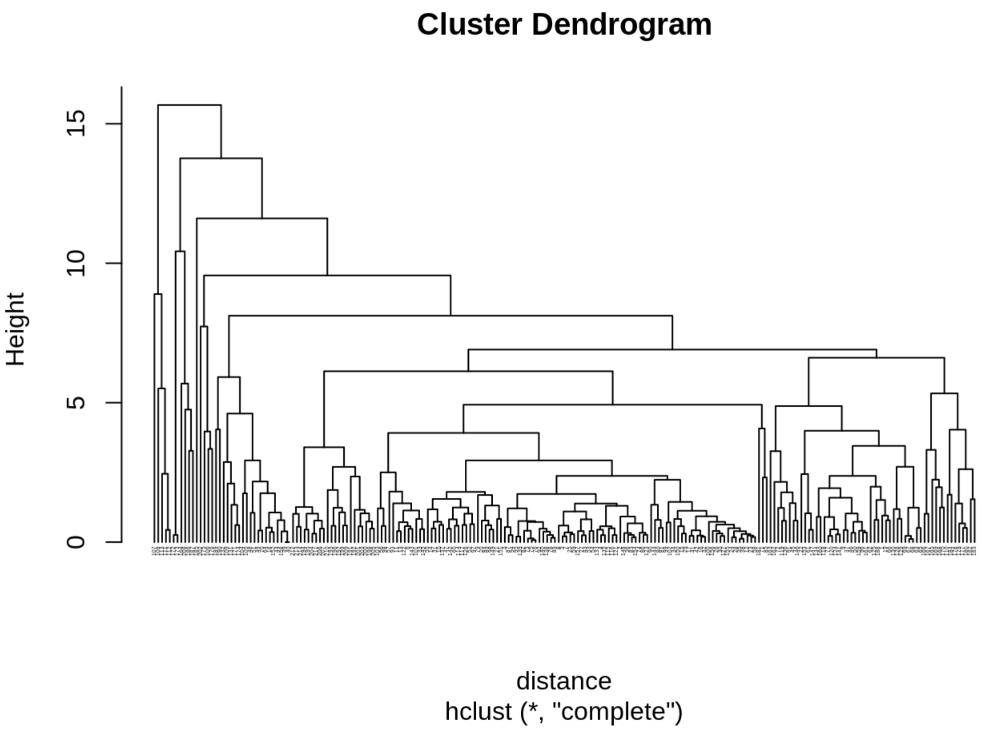
```
table('Glass_Type' = Glass$Type2, 'Cluster' = km.res$cluster)
```

```
##             Cluster
## Glass_Type     1   2   3
##   Non-Window   8  11  32
##   Window     129  34   0
```

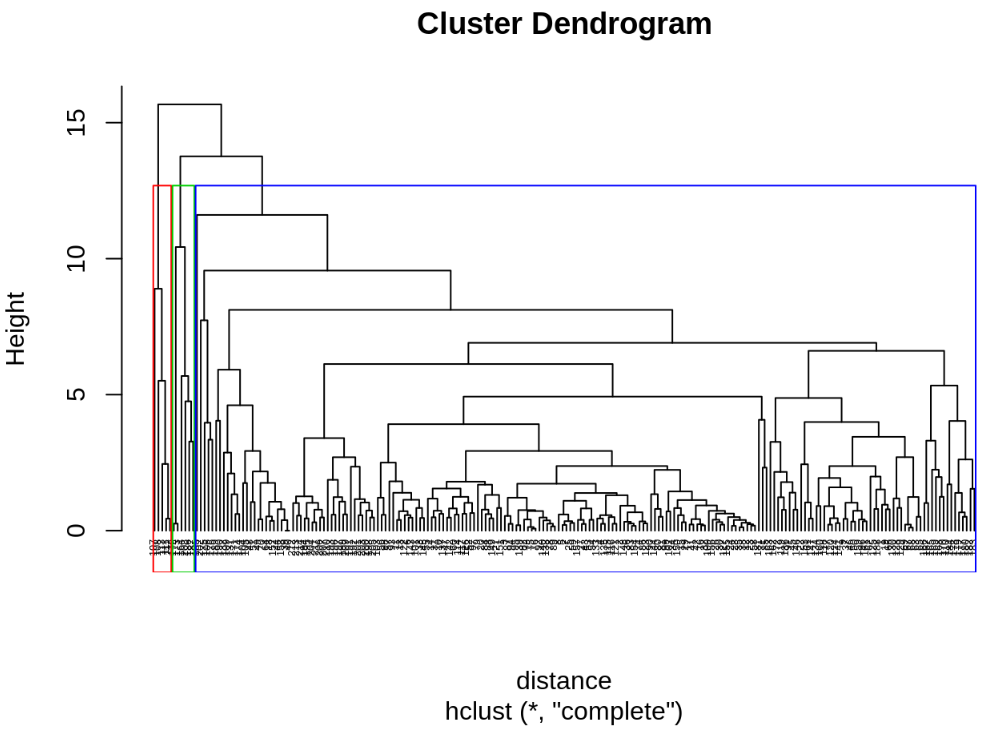
Hierarchical clustering

```
distance <- dist(prc, method = "euclidean")
hc_comp <- hclust(distance, method = "complete")
```

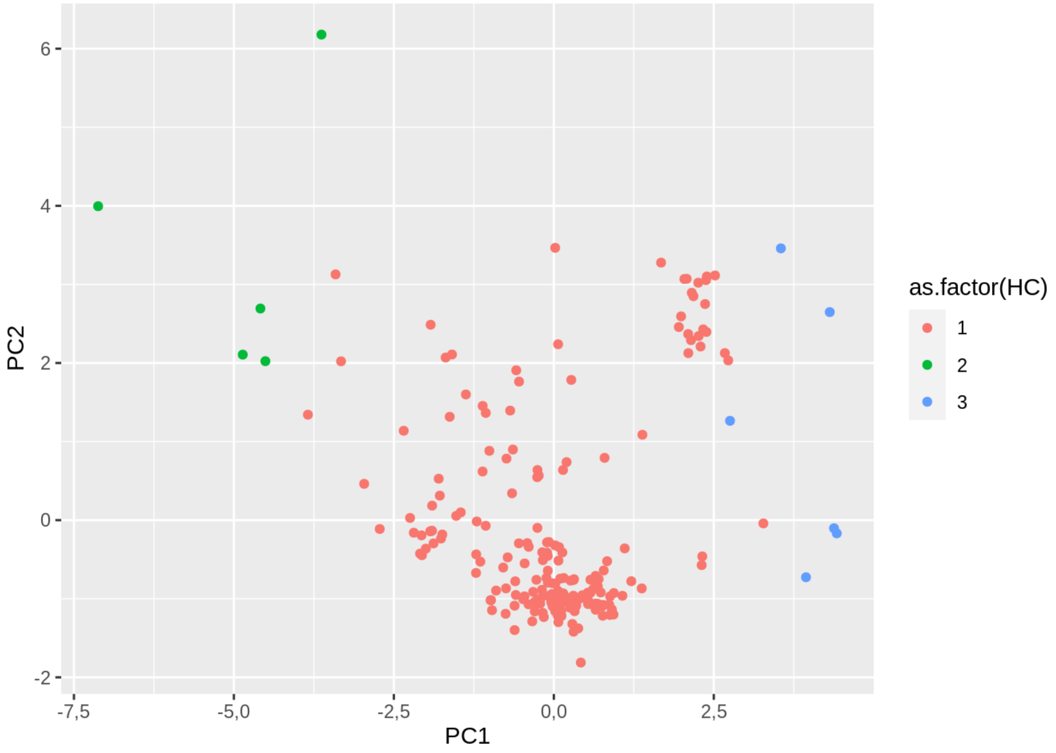
```
plot(hc_comp, cex = 0.2, hang = -1)
```



```
plot(hc_comp, cex = 0.4, hang = -1)
rect.hclust(hc_comp, k = 3, border = 2:5)
```



```
df$HC <- cutree(hc_comp, 3)
ggplot(df,aes(x=PC1,y=PC2,color= as.factor(HC))) + geom_point()
```



```
table('Glass_Type' = Glass$Type, 'Cluster' = km.res$cluster)
```

```
##           Cluster
## Glass_Type  1  2  3
##          1 54 16  0
##          2 61 15  0
##          3 14  3  0
##          5  2  8  3
##          6  5  1  3
##          7  1  2 26
```

```
table('Glass_Type' = Glass$Type2, 'Cluster' = km.res$cluster)
```

```
##             Cluster
## Glass_Type     1   2   3
##   Non-Window   8  11  32
##   Window     129  34   0
```

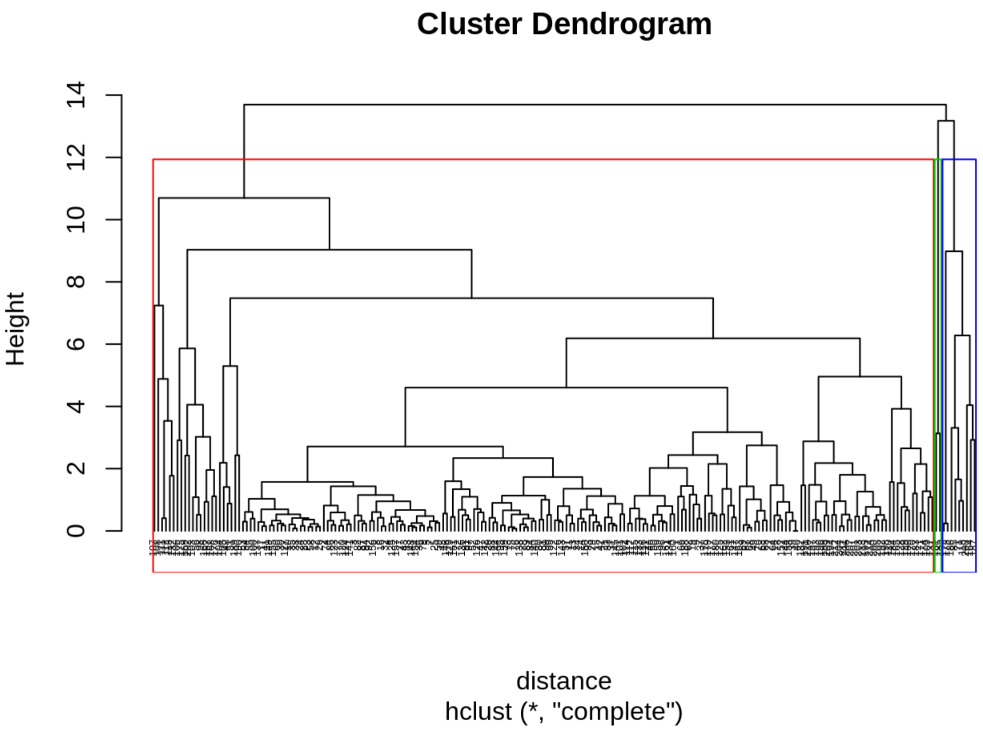
```
table('K_means' = df$K_means, 'HC' =  df$HC)
```

```
##        HC
## K_means   1   2   3
##       1 137   0   0
##       2  40   5   0
##       3  26   0   6
```

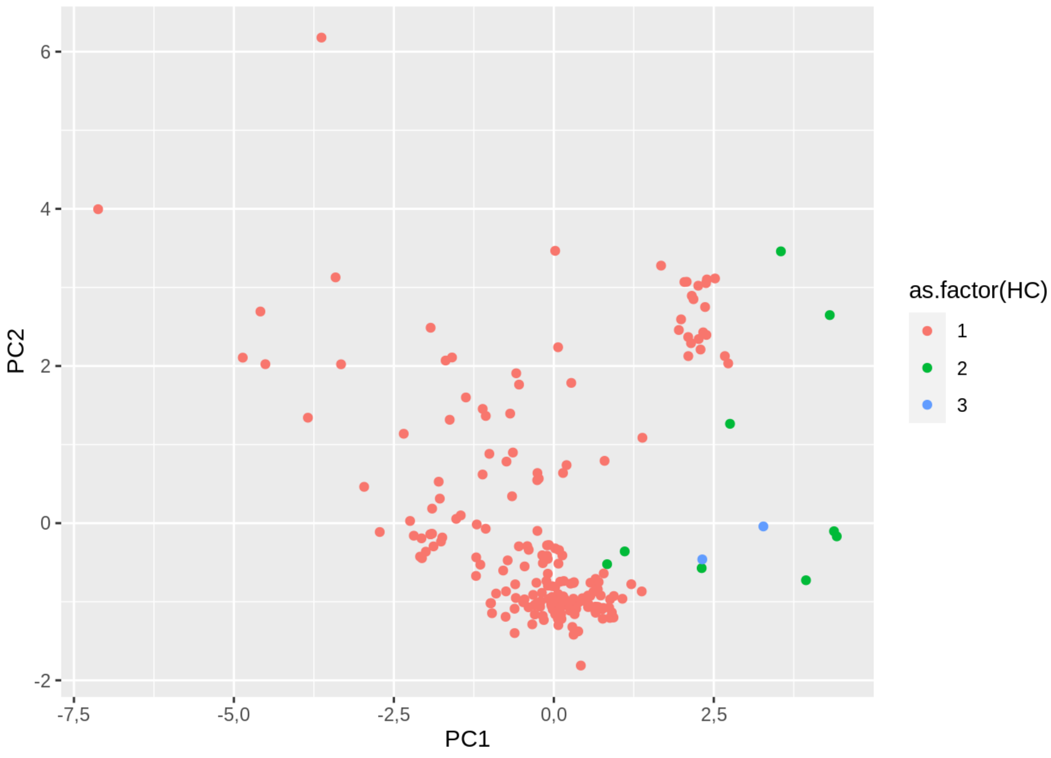
2nd variant

```
prc <- pca_res[,1:4]
distance <- dist(prc, method = "euclidean")
hc_comp <- hclust(distance, method = "complete")
```

```
plot(hc_comp, cex = 0.4, hang = -1)
rect.hclust(hc_comp, k = 3, border = 2:5)
```



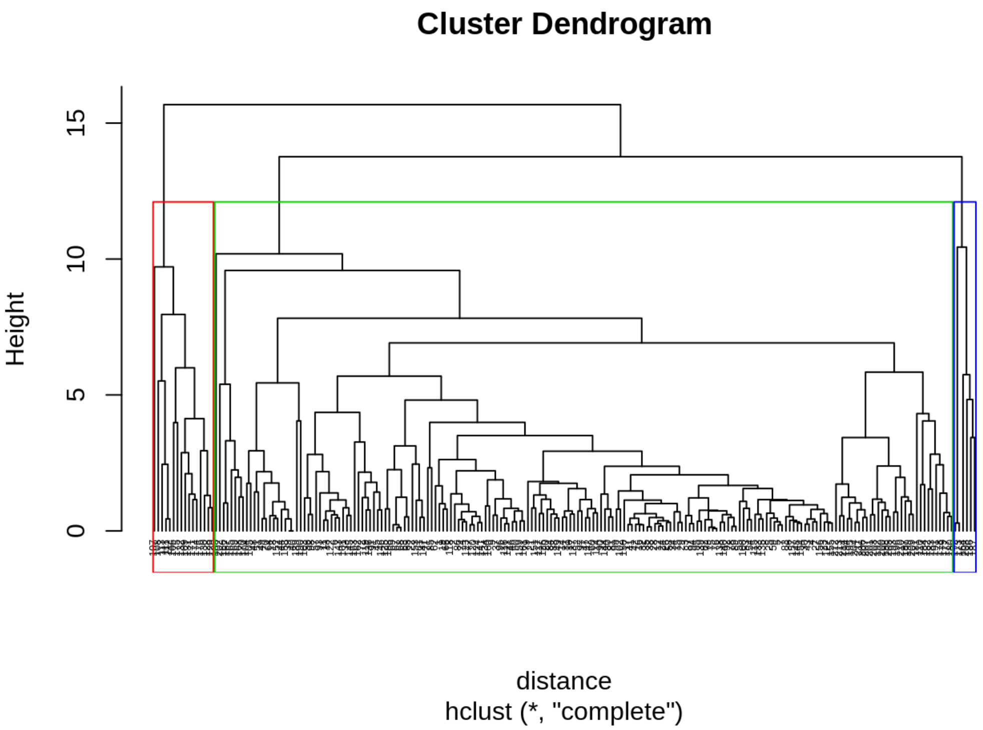
```
df$HC <- cutree(hc_comp, 3)
ggplot(df,aes(x=PC1,y=PC2,color= as.factor(HC))) + geom_point()
```



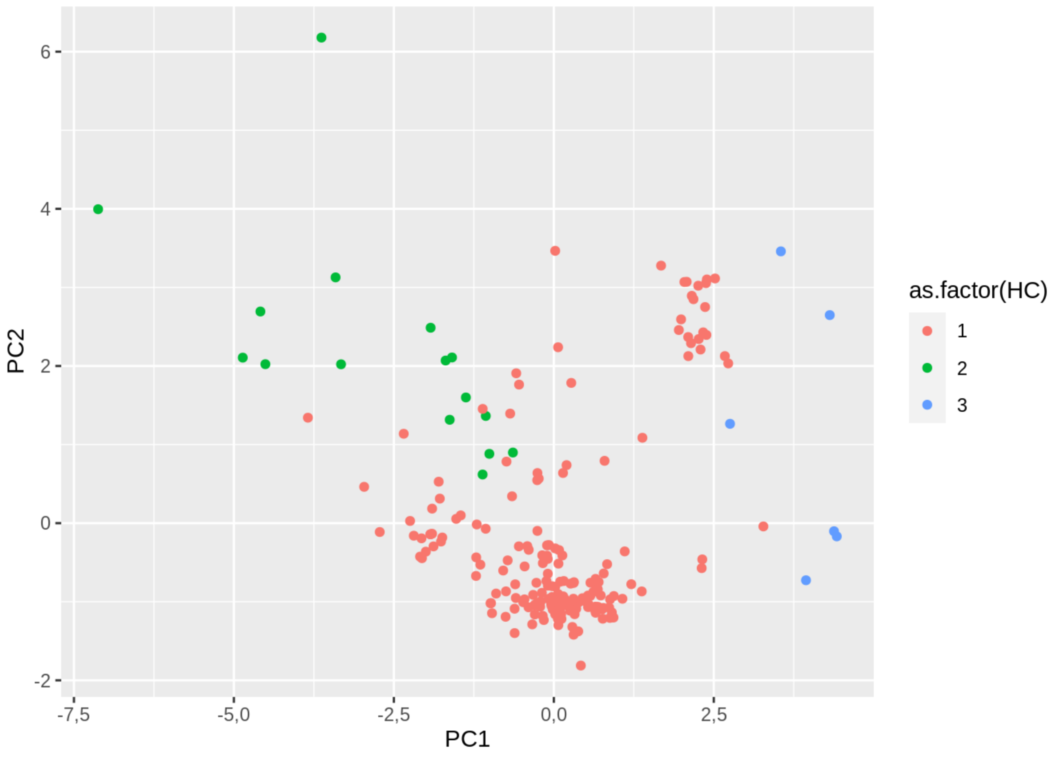
3rd variant

```
prc <- pca_res
distance <- dist(prc, method = "euclidean")
hc_comp <- hclust(distance, method = "complete")
```

```
plot(hc_comp, cex = 0.4, hang = -1)
rect.hclust(hc_comp, k = 3, border = 2:5)
```



```
df$HC <- cutree(hc_comp, 3)
ggplot(df,aes(x=PC1,y=PC2,color= as.factor(HC))) + geom_point()
```



```
table('Glass_Type' = Glass$Type, 'Cluster' = df$HC)
```

```
##          Cluster
## Glass_Type  1  2  3
##          1 70  0  0
##          2 64 12  0
##          3 17  0  0
##          5  6  4  3
##          6  9  0  0
##          7 26  0  3
```

```
table('Glass_Type' = Glass$Type2, 'Cluster' = df$HC)
```

```
##             Cluster
## Glass_Type     1   2   3
##   Non-Window  41   4   6
##   Window     151  12   0
```

```
table('K_means' = df$K_means, 'HC' =  df$HC)
```

```
##        HC
## K_means   1   2   3
##       1 137   0   0
##       2  29  16   0
##       3  26   0   6
```