Atividade 3 - Mineração de Dados

Rodrigo Malta Esteves

30/03/2021

```
library(readr) #Ler os dados
library(factoextra) #Análise de componentes principais
library(stringr) #Manipulação de strings
library("reshape2") #
library(tidyverse) #
library(dendextend) #Pacote para fazer dendograma
library(cluster)
```

```
nutr <- read_csv("C:\\Users\\malta\\Desktop\\Pós Graduação\\Mineração de Dados\\Trabalhos\\Atividade 3\\
nutr2 <- nutr[,c(2,4:8)]
attach(nutr2)
```

Questão 1

```
## Manipulando os dados
nutr3 <- nutr2[,-1]    # retirando a coluna de nomes
nomes <- nutr2[,1]   # pegando a primeira palavra
nutr3 <- as.data.frame(nutr3)
nomes <- as.data.frame(nomes)

## Nomeando as linhas do data frame
row.names(nutr3) <- nomes$Country
head(nutr3)
```

```
##                           Severe.Wasting   Wasting Overweight Stunting
## LEBANON                         2.250000  5.100000  18.750000 16.85000
## ETHIOPIA                        3.016667  9.957143   2.750000 47.84286
## GUINEA                          3.242857  9.425000   4.333333 34.23750
## NIGER (THE)                     4.330000 15.936364   1.520000 46.00000
## COMOROS (THE)                   5.400000 10.125000  12.466667 39.12500
## REPUBLIC OF MOLDOVA (THE)       1.450000  3.850000   6.900000  8.55000
##                           Underweight
## LEBANON                       3.85000
## ETHIOPIA                     31.05714
## GUINEA                       19.26250
## NIGER (THE)                  38.26364
## COMOROS (THE)                19.57500
## REPUBLIC OF MOLDOVA (THE)     2.70000
```

```
## Análise de componentes principais
acp <- prcomp(nutr3,scale=TRUE)   # scale padroniza as variáveis
```

i) Porcentagem da variabilidade explicada por cada componente.

Verificamos que as duas primeiras componentes foram suficientes para explicar 89.29% da variabilidade total. Nesse estudo, a primeira componente principal (CP1) representa 68.45% do total da variância e inclui as variáveis com os coeficientes de maior peso (autovetores) e contribuições (acp$rotations), representada por: Underweight(-0.7854,52.44%) e Stunting(-0.5745,46.31%). Já a segunda componente (CP2), a variável de maior autovetor e contribuição foi Overweight(0.6244,82.32%).

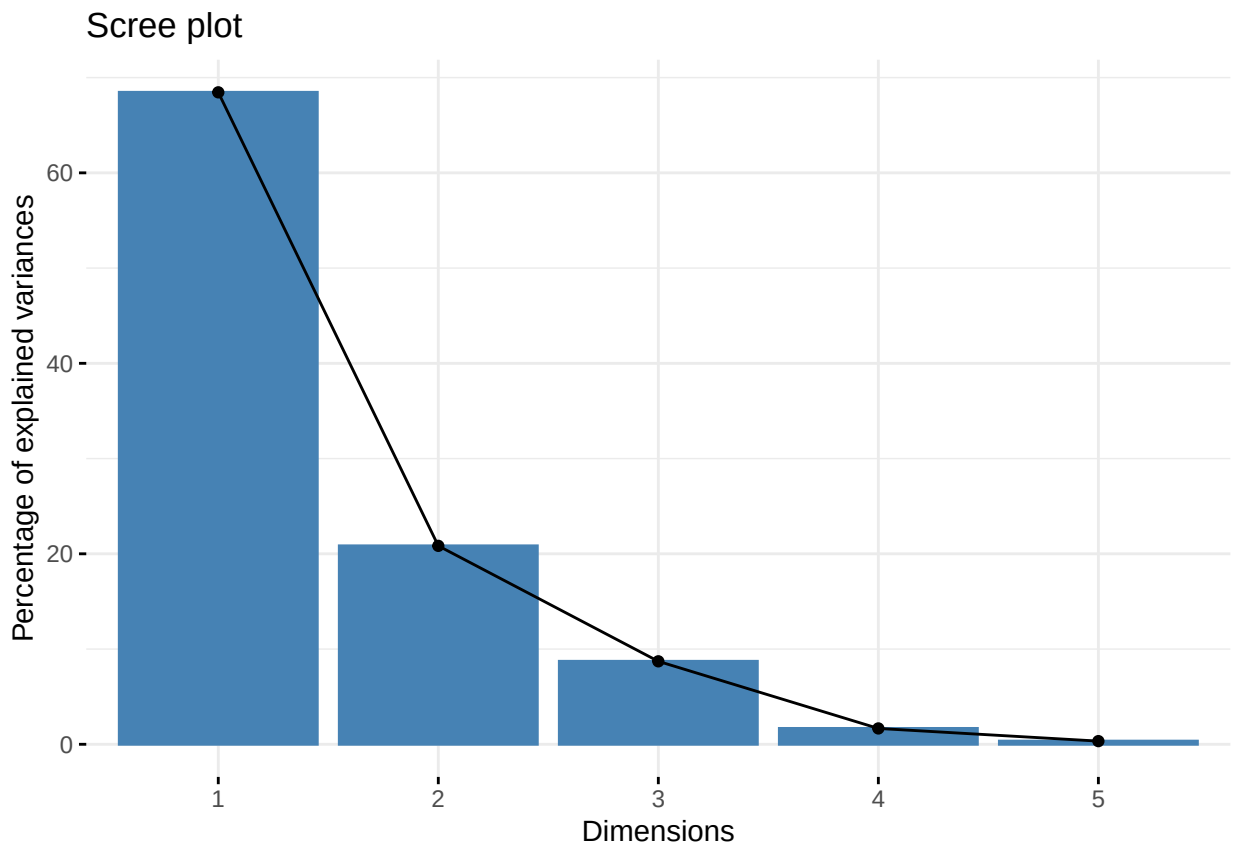
```
# Proporção da variabilidade explicada por cada componente
summary(acp)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     1.8500 1.0207 0.66000 0.28855 0.12922
## Proportion of Variance 0.6845 0.2084 0.08712 0.01665 0.00334
## Cumulative Proportion  0.6845 0.8929 0.98001 0.99666 1.00000
```

```
acp
```

```
## Standard deviations (1, .., p=5):
## [1] 1.8500351 1.0206917 0.6599988 0.2885523 0.1292196
##
## Rotation (n x k) = (5 x 5):
##                        PC1        PC2        PC3        PC4         PC5
## Severe.Wasting  0.4403885 -0.4956999  0.3359607 -0.5951612 -0.30537368
## Wasting         0.5060601 -0.2126939  0.3532750  0.4315907  0.62256783
## Overweight     -0.2456605 -0.8232197 -0.4316778  0.2737154 -0.02635352
## Stunting        0.4631854  0.1244338 -0.7390819 -0.3444426  0.32418008
## Underweight     0.5244725  0.1259703 -0.1724510  0.5157051 -0.64293777
```

```
## Scree plot
fviz_eig(acp) # Proporção da variabilidade explicada por cada componente
```

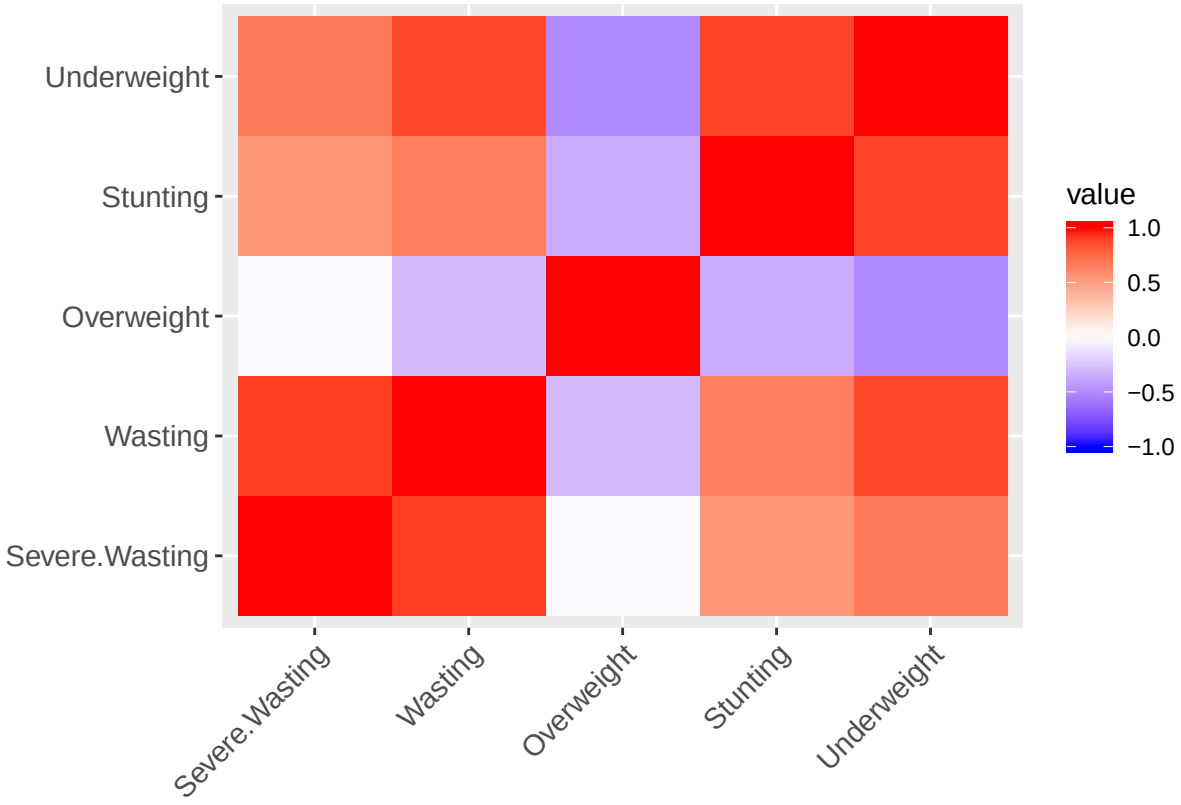



```r
# Matriz de covariância
Sigma <- cov(nutr3)

# Autovalores e autovetores
e <- eigen(Sigma)
Gamma <- e$vectors   # autovetores
Lambda <- e$values   # autovalores
e
```

```
## eigen() decomposition
## $values
## [1] 350.3236312  29.0168386  19.0649359   2.3571660   0.2244724
##
## $vectors
##             [,1]        [,2]       [,3]       [,4]        [,5]
## [1,] -0.05840832 -0.04556953 -0.2527578  0.4477484  0.85448627
## [2,] -0.18383546 -0.29684403 -0.4982167  0.6170583 -0.49910643
## [3,]  0.12513245  0.62448221 -0.7216572 -0.2695410 -0.03037122
## [4,] -0.78549250  0.52994207  0.2544577  0.1872986 -0.04830570
## [5,] -0.57457497 -0.48886645 -0.3199309 -0.5576982  0.13225027
```

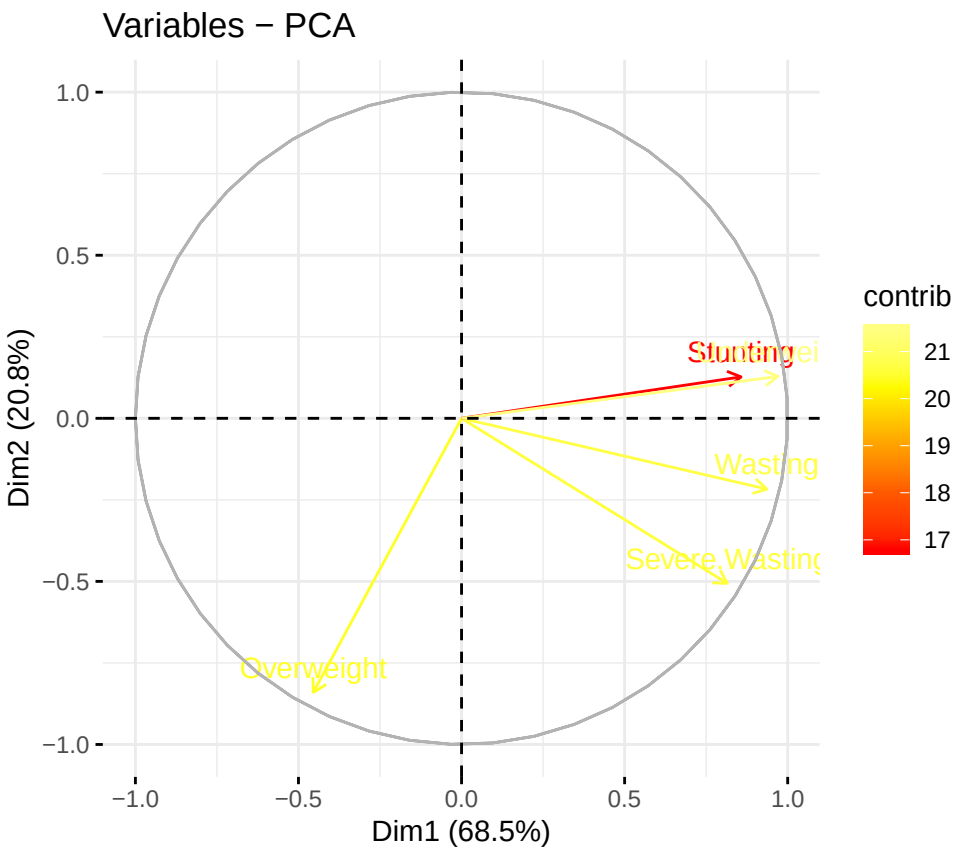
```r
# Reshape
corrm <- cor(nutr3)
corrm2 <- melt(corrm)
```

```r
# Visualização da matriz de correlação
ggplot(corrm2,aes(x=Var1,y=Var2,fill=value)) + geom_tile() +
  labs(x="",y="") + theme(axis.text.x=element_text(angle=45,hjust=1,size=11),axis.text.y=element_text(s
  scale_fill_gradient2(low="blue",high="red",mid="white",limit=c(-1,1))
```

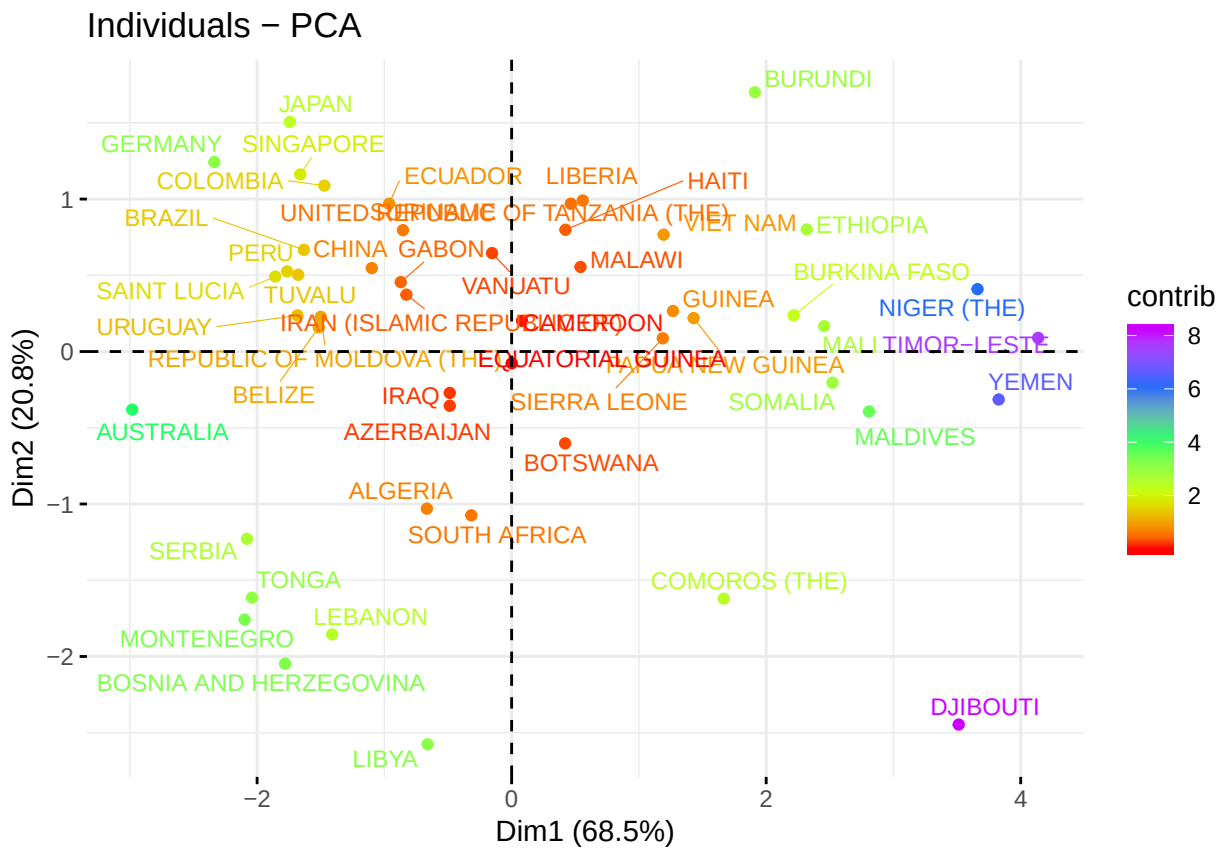


ii) Variáveis, países e biplot.

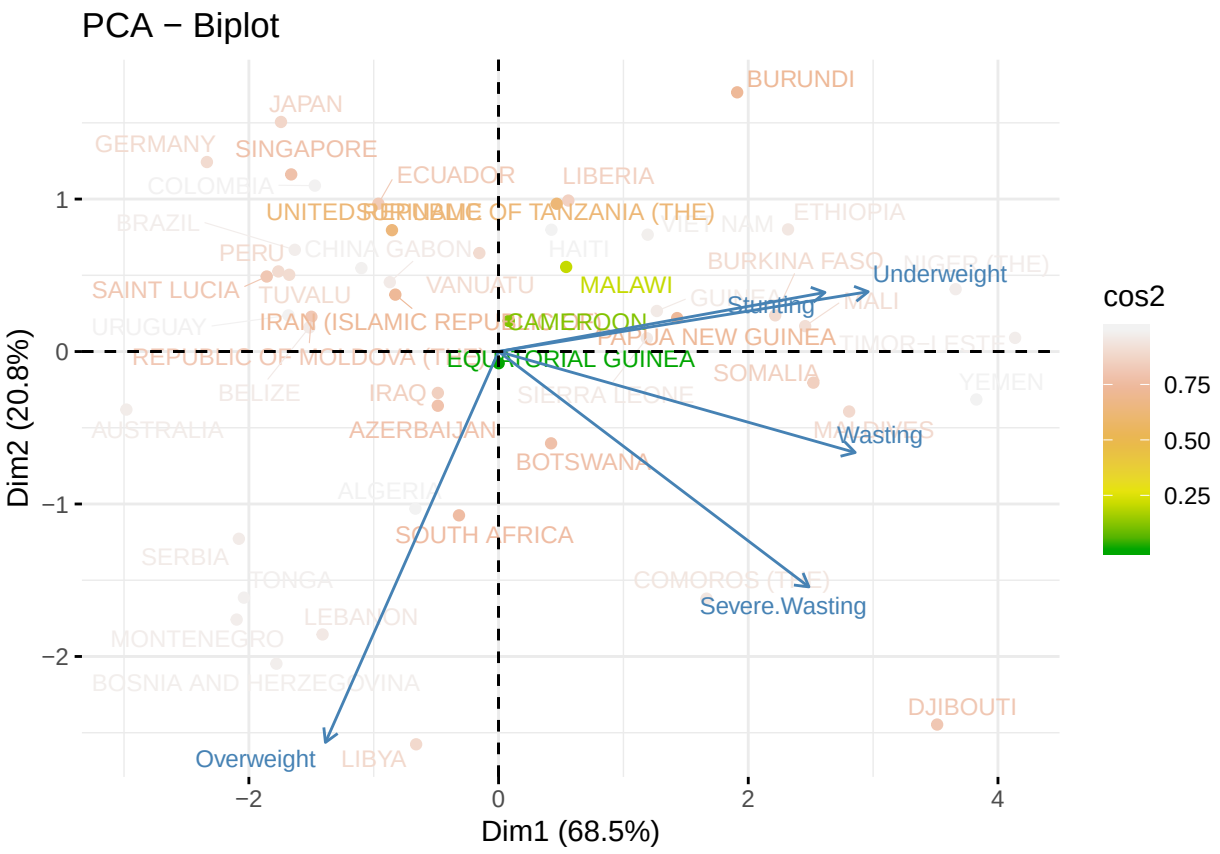
```r
## Variables - PCA -> relação entre as componentes principais e as variáveis originais.
fviz_pca_var(acp,col.var="contrib",gradient.cols=heat.colors(5))
```




```
## Individuals - PCA -> elementos do banco de dados em termos das componentes principais.
fviz_pca_ind(acp,repel=TRUE,col.ind="contrib",gradient.cols=rainbow(5),labelsize = 3)
```



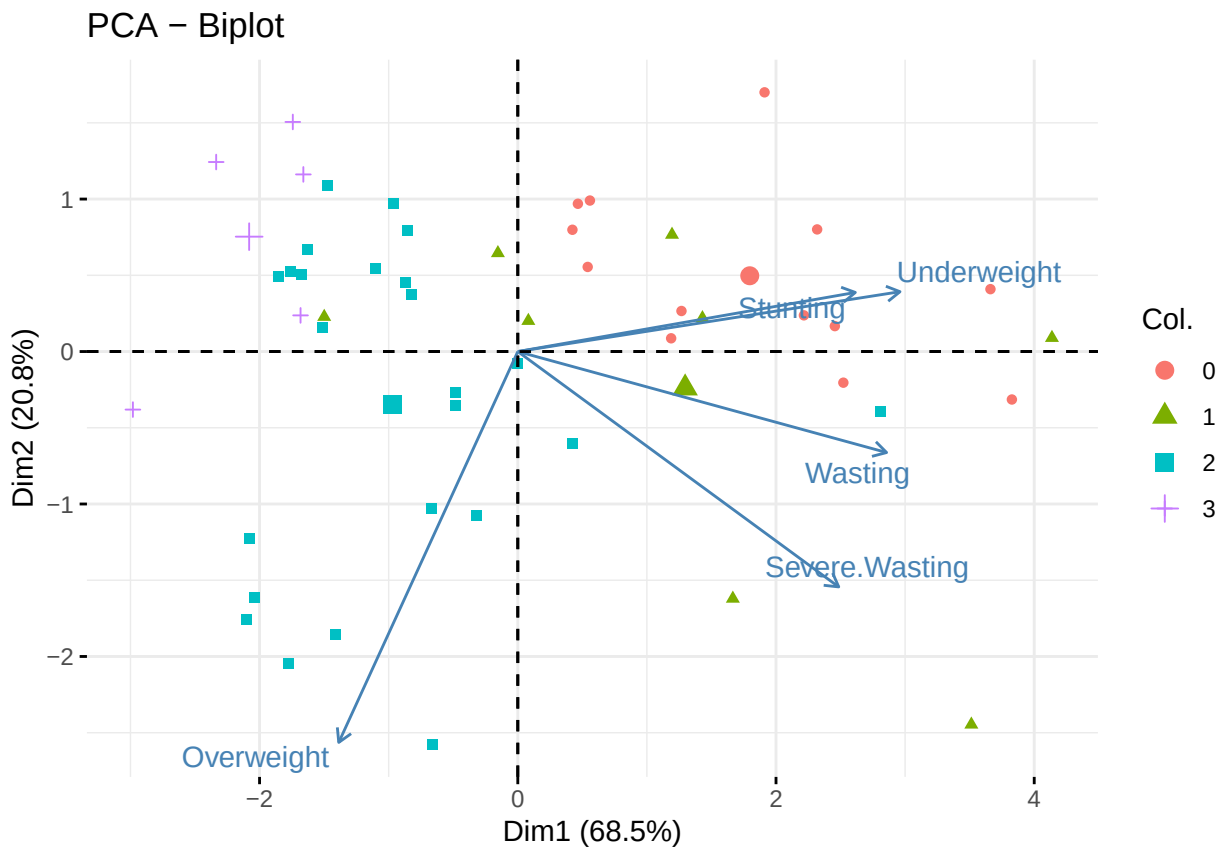

```
## PCA - Biplot -> combinação dos dois plots anteriores
fviz_pca_biplot(acp,repel=TRUE,col.ind="cos2",gradient.cols=terrain.colors(5),labelsize = 3)
```



Questão 2

Países com rendas “0” e “1” tem uma tendência muito maior a ter problemas relacionadas a subnutrição uma vez que estão localizados nos primeiro e quarto quadrantes do biplot. Os paises afetados por problema de sobrepeso estão mais relacionados com rendas maiores “2” e “3”.

```
# Biplot dos paises pela renda
fviz_pca_biplot(acp,col.ind=as.factor(nutr$Income.Classification), repel=TRUE,geom="point")
```



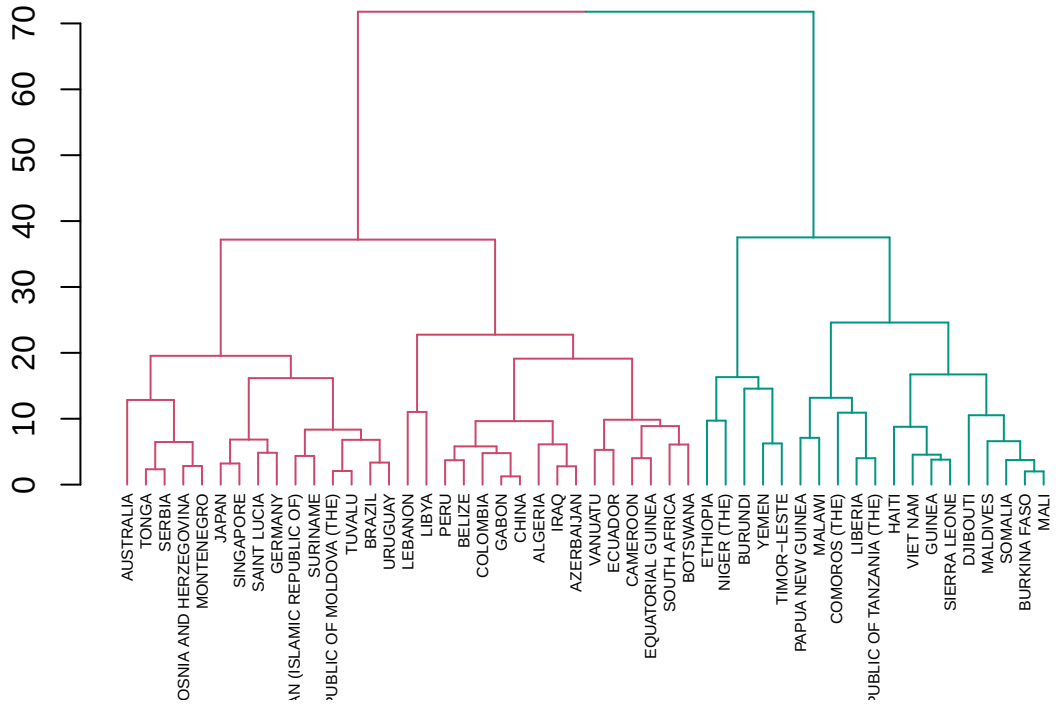
Questão 3

Nesse caso, podemos usar para definir os clusters a variável referente a renda (Income.Classification)

```
dist_customers <- dist(nutr3)
hc_customers <- hclust(dist_customers)

#Calculate the mean for each category
#segment_customers %>%
#  group_by(cluster) %>%
#  summarise_all(list(mean))

hc_players <- hclust(dist_customers,method="complete")
dend_players <- as.dendrogram(hc_players)

clust_customers2 <- cutree(hc_customers,k=2)
segment_customers2 <- mutate(nutr3, cluster = clust_customers2)
count(segment_customers2,cluster)
```

```
## # A tibble: 2 x 2
##   cluster     n
##     <int> <int>
## 1       1    31
## 2       2    19
```

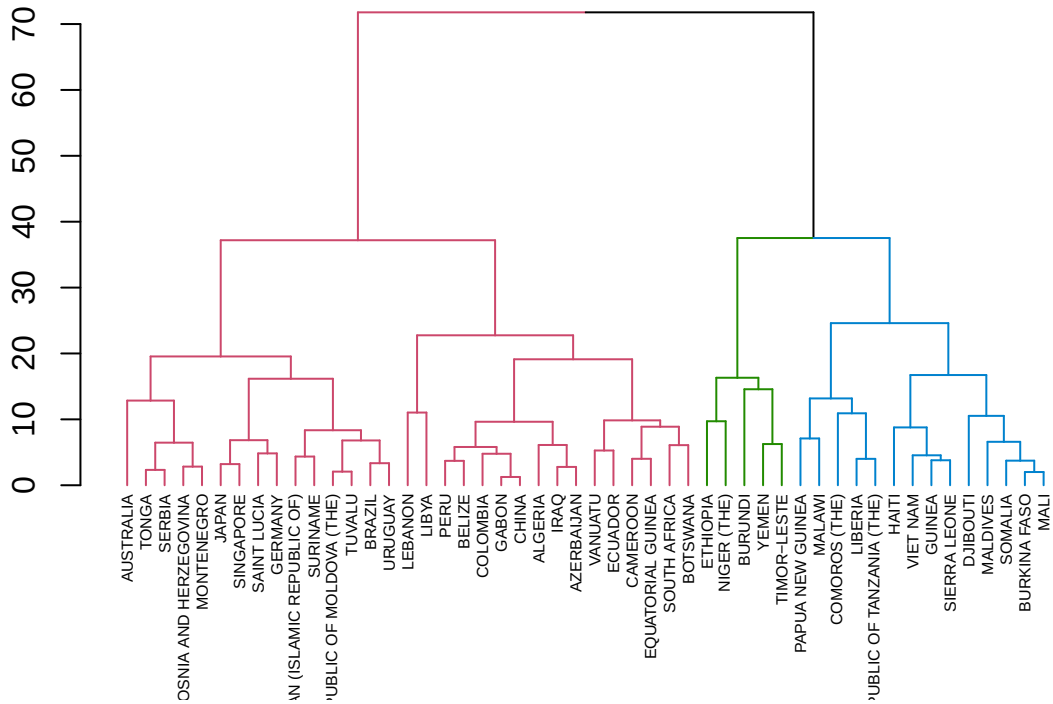
```
dend_color_2 <- color_branches(dend_players,k=2)
dend_color_2 <- set(dend_color_2, "labels_cex", .5)
plot(dend_color_2)
```




```
clust_customers3 <- cutree(hc_customers,k=3)
segment_customers3 <- mutate(nutr3, cluster = clust_customers3)
count(segment_customers3,cluster)
```

```
## # A tibble: 3 x 2
##   cluster     n
##     <int> <int>
## 1       1    31
## 2       2     5
## 3       3    14
```

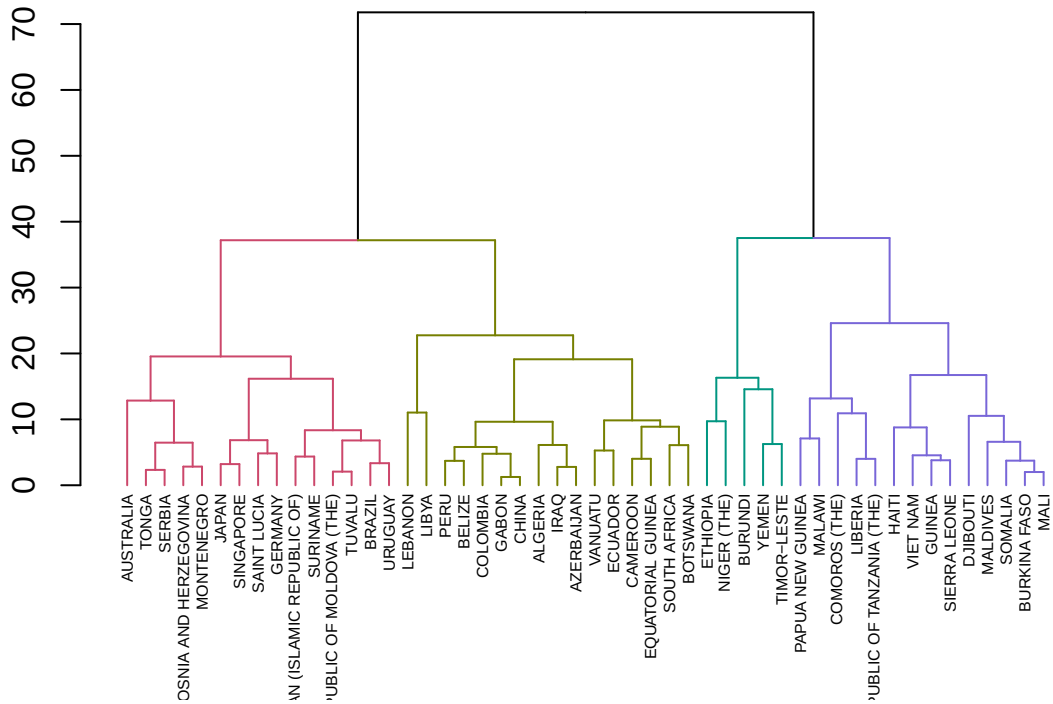
```
dend_color_3 <- color_branches(dend_players,k=3)
dend_color_3 <- set(dend_color_3, "labels_cex", .5)
plot(dend_color_3)
```




```
clust_customers4 <- cutree(hc_customers,k=4)
segment_customers4 <- mutate(nutr3, cluster = clust_customers4)
count(segment_customers4,cluster)
```

```
## # A tibble: 4 x 2
##   cluster     n
##     <int> <int>
## 1       1    16
## 2       2     5
## 3       3    14
## 4       4    15
```

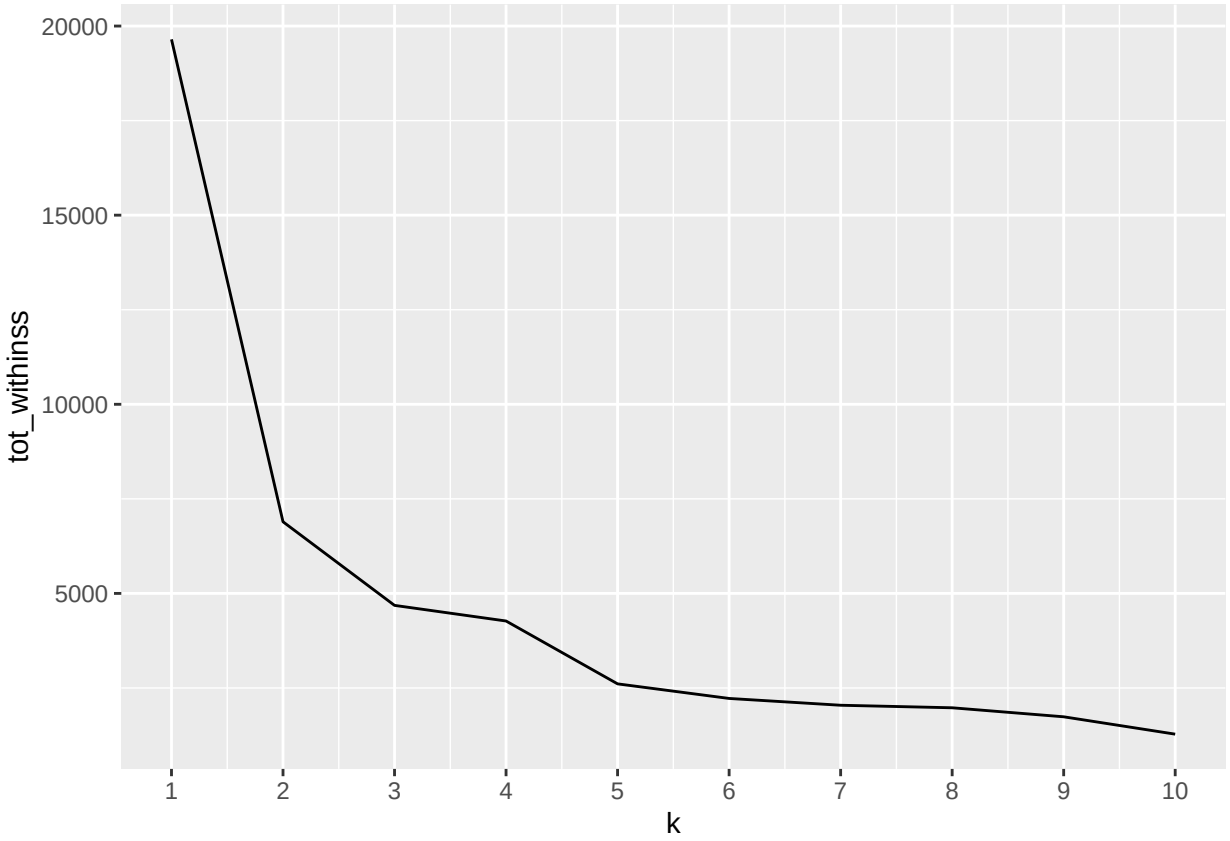
```
dend_color_4 <- color_branches(dend_players,k=4)
dend_color_4 <- set(dend_color_4, "labels_cex", .5)
plot(dend_color_4)
```



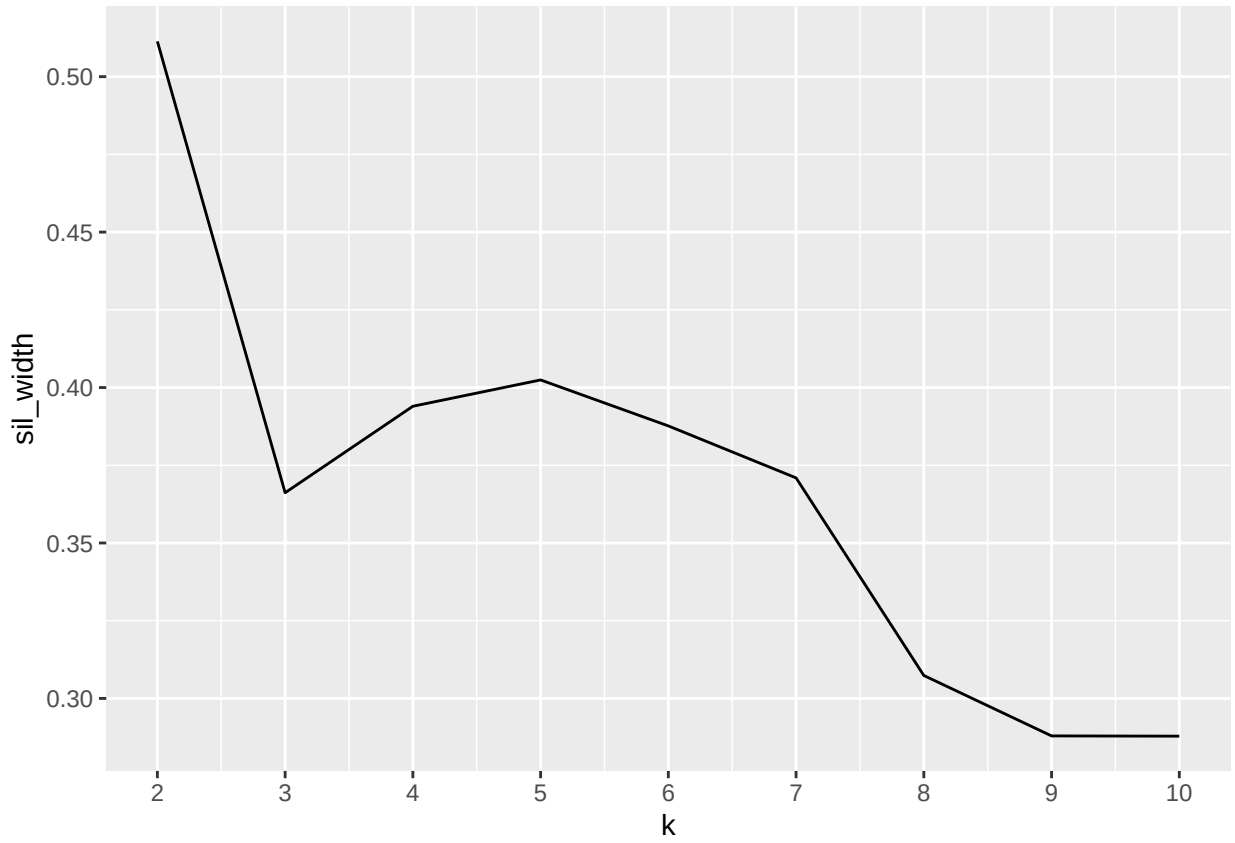
4.

A análise do "cotovelo" aponta para k=2 clusters como a melhor escolha possível, concordando com a análise de componentes principais. No entanto, assim como no dendrograma, o número de clusters ideal para ser usado dependerá de como os dados serão usados para análises posteriores.

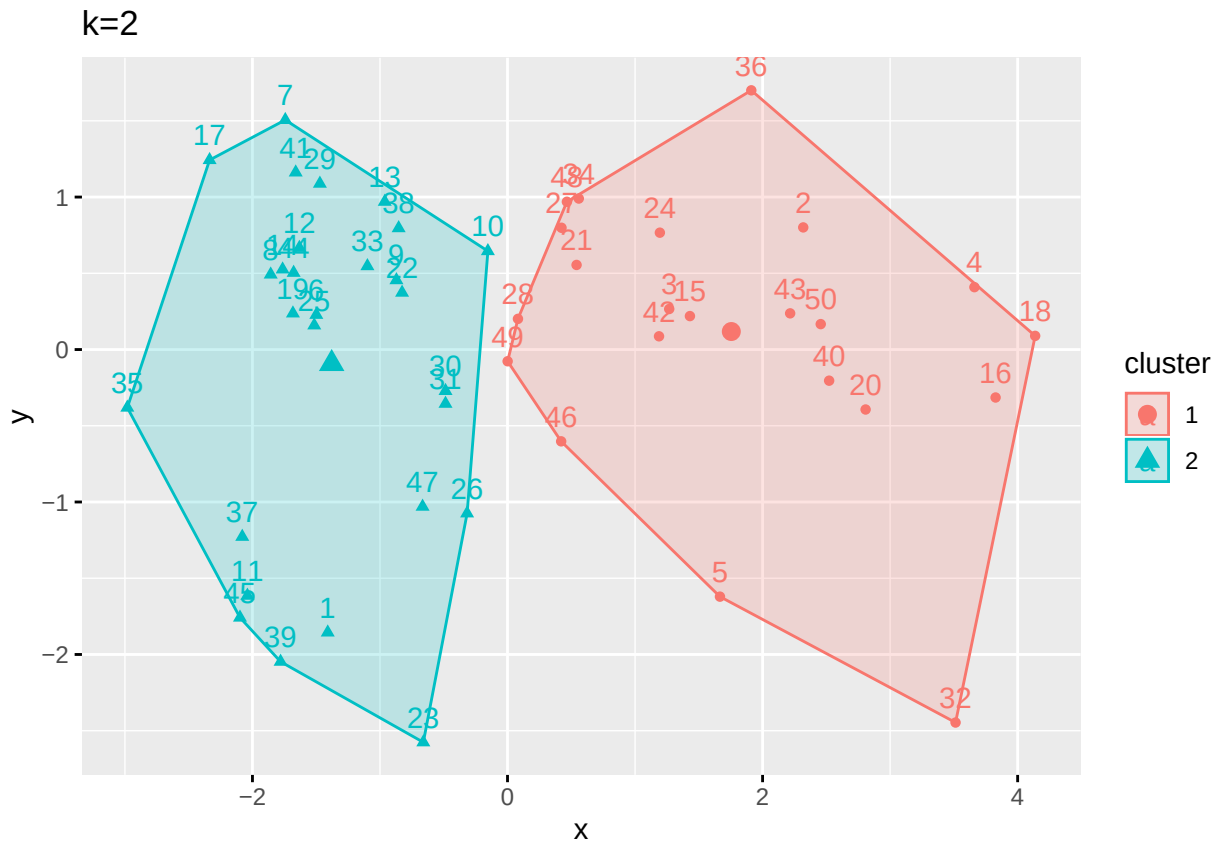
```
# Elbow Analysis
# Rodando modelos variando os k's
tot_withinss <- map_dbl(1:10,  function(k){
  model <- kmeans(x = nutr3, centers = k)
  model$tot.withinss
})
# Gerando um data frame com os k
elbow_df <- data.frame(
  k = 1:10,
  tot_withinss = tot_withinss
)
# Plot do "cotovelo"
ggplot(elbow_df, aes(x = k, y = tot_withinss)) +
  geom_line() +
  scale_x_continuous(breaks = 1:10)
```



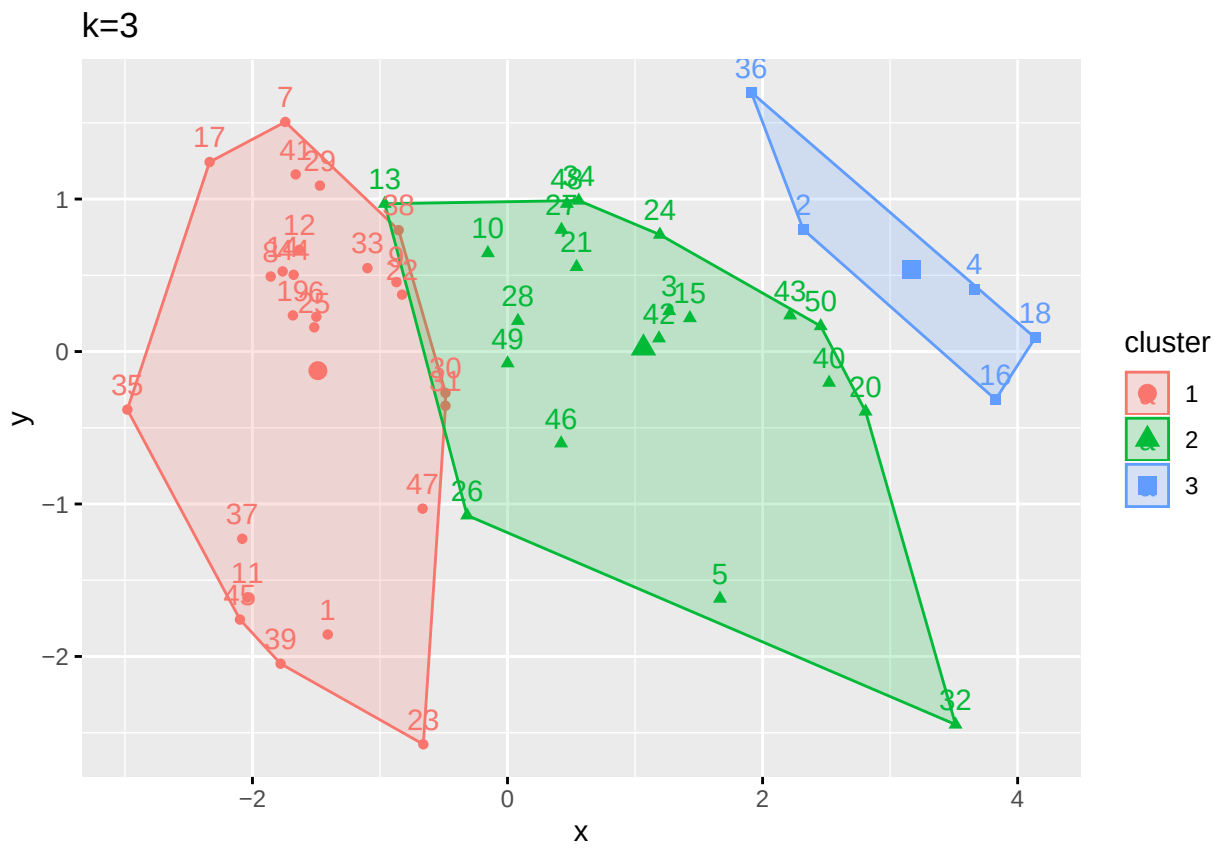

```r
# Average Silhouette Widths
# Use map_dbl to run many models with varying value of k
sil_width <- map_dbl(2:10,  function(k){
  model <- pam(nutr3, k = k)
  model$silinfo$avg.width
})
# Generate a data frame containing both k and sil_width
sil_df <- data.frame(
  k = 2:10,
  sil_width = sil_width
)
# Plot the relationship between k and sil_width
ggplot(sil_df, aes(x = k, y = sil_width)) +
  geom_line() +
  scale_x_continuous(breaks = 2:10)
```



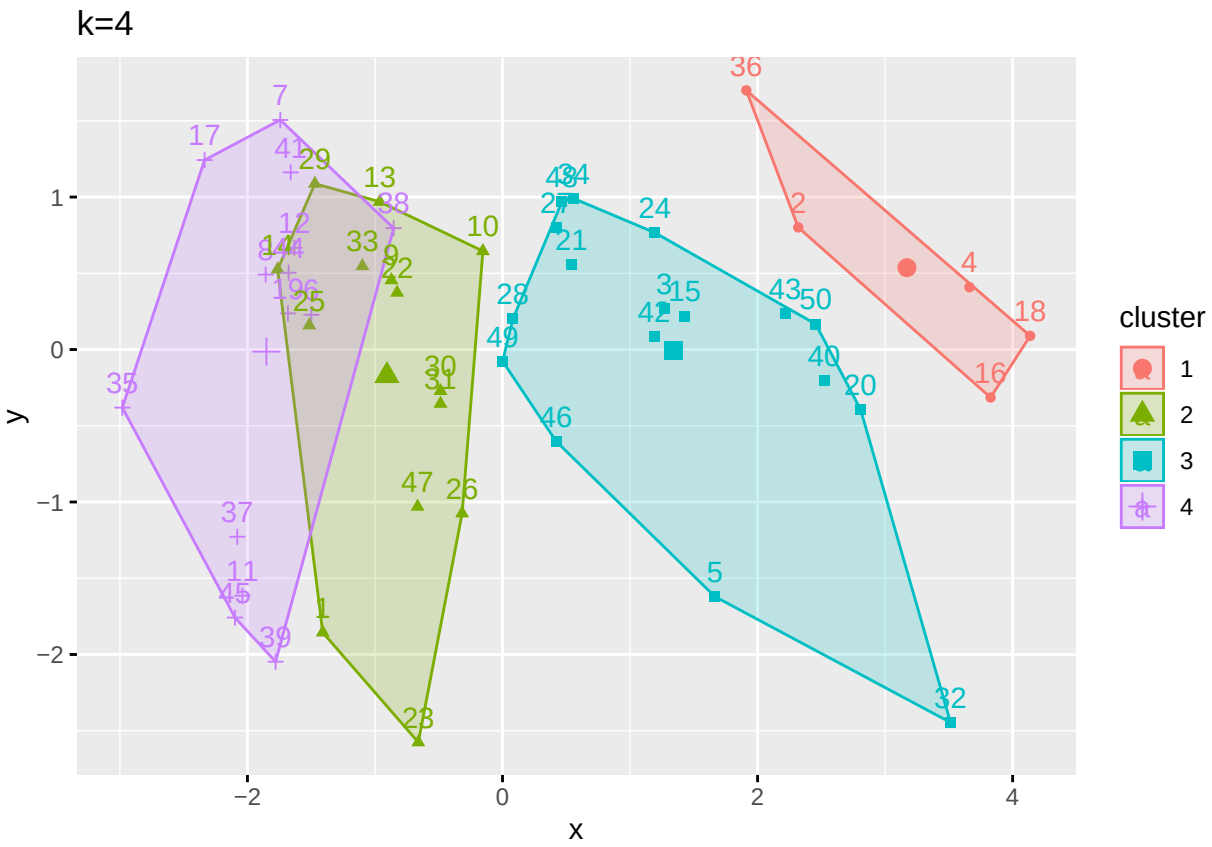

```
k2 <- kmeans(nutr[,4:8],2)
k3 <- kmeans(nutr[,4:8],3)
k4 <- kmeans(nutr[,4:8],4)

fviz_cluster(k2,data=nutr[,4:8]) + labs(x="x",y="y",title="k=2")
```




```
fviz_cluster(k3,data=nutr[,4:8]) + labs(x="x",y="y",title="k=3")
```




```
fviz_cluster(k4,data=nutr[,4:8]) + labs(x="x",y="y",title="k=4")
```


k=4
cluster
1
2
3
4
x
y
−2
0
2
4
1
0
−1
−2