K-Means, proceso 1.3.2

David Ruiz

2025-09-04

Activación librerías

```
source('R-libraries-ICdD.r')
```

Importación la información

```
df<- read_csv("Sleep_health_and_lifestyle.csv") %>% data.frame()
```

La variable Blood Preassure se debe ajustar porque es una medida cuantitativa que se cargó como carácter. Hagamos la separación de los registros como: Sistólica y Diastólica para incluirla de manera adecuada en la reducción de Dimensionalidad y verifiquemos que hizo adecuadamente la separación

```
df <- df %>%
  separate(Blood.Pressure, into = c("sistolica_bp", "diastolica_bp"), sep
= "/", convert = TRUE)
df |> head()
```

```
##   Gender Age           Occupation Sleep.Duration Quality.of.Sleep
## 1   Male  27     Software Engineer            6.1                6
## 2   Male  28                Doctor            6.2                6
## 3   Male  28                Doctor            6.2                6
## 4   Male  28 Sales Representative             5.9                4
## 5   Male  28 Sales Representative             5.9                4
## 6   Male  28     Software Engineer            5.9                4
##   Physical.Activity.Level Stress.Level BMI.Category sistolica_bp
diastolica_bp
## 1                      42            6   Overweight          126
83
## 2                      60            8       Normal          125
80
## 3                      60            8       Normal          125
80
## 4                      30            8        Obese          140
90
## 5                      30            8        Obese          140
90
## 6                      30            8        Obese          140
90
##   Heart.Rate Daily.Steps Sleep.Disorder
## 1         77        4200           None
## 2         75       10000           None
## 3         75       10000           None
## 4         85        3000    Sleep Apnea
```

```
## 5          85        3000     Sleep Apnea
## 6          85        3000        Insomnia
```

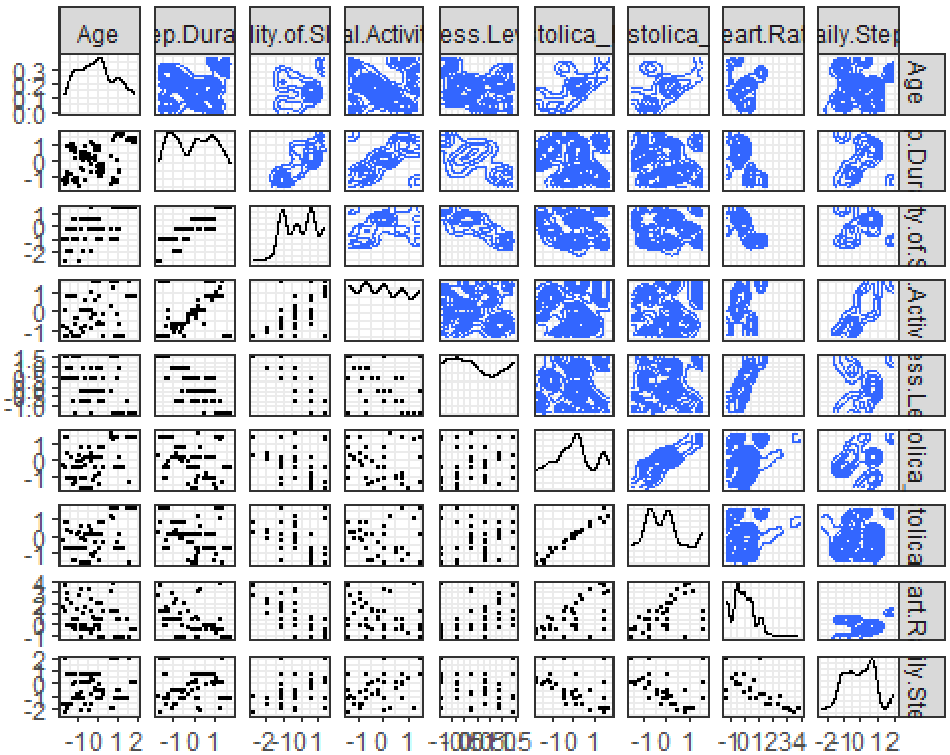
Camino 1.3.1

Separación de Variables

```
base_recipe <-
  recipe(Sleep.Disorder ~ ., data = df) %>%
  step_dummy(all_nominal_predictors()) %>%      # Variables categóricas a dummies
  step_normalize(all_numeric_predictors()) #%>%  # Escalar/normalizar y centrar numéricas

prepped_data <- prep(base_recipe) %>% bake(new_data = NULL)

df_C132=prepped_data %>% select(-Sleep.Disorder)

dim(df_C132)

## [1] 374  23

head(df_C132)

## # A tibble: 6 × 23
##     Age Sleep.Duration Quality.of.Sleep Physical.Activity.Level Stress.Level
##   <dbl>          <dbl>            <dbl>                   <dbl>        <dbl>
## 1 -1.75          -1.30            -1.10                  -0.824        0.347
## 2 -1.64          -1.17            -1.10                   0.0398       1.47
## 3 -1.64          -1.17            -1.10                   0.0398       1.47
## 4 -1.64          -1.55            -2.77                  -1.40         1.47
## 5 -1.64          -1.55            -2.77                  -1.40         1.47
## 6 -1.64          -1.55            -2.77                  -1.40         1.47
## # ℹ 18 more variables: sistolica_bp <dbl>, diastolica_bp <dbl>,
## #   Heart.Rate <dbl>, Daily.Steps <dbl>, Gender_Male <dbl>,
## #   Occupation_Doctor <dbl>, Occupation_Engineer <dbl>,
## #   Occupation_Lawyer <dbl>, Occupation_Manager <dbl>, Occupation_Nurse <dbl>,
## #   Occupation_Sales.Representative <dbl>, Occupation_Salesperson <dbl>,
## #   Occupation_Scientist <dbl>, Occupation_Software.Engineer <dbl>,
## #   Occupation_Teacher <dbl>, BMI.Category_Normal.Weight <dbl>, …
```

1. Exploracion del numero de grupos (kMedias)

```r
ggpairs(df_C132[,1:9],
        progress = FALSE,
upper = list(continuous = "density"),
lower = list(continuous = wrap("points", size = 0.5)),
diag = list(continuous = "densityDiag")) +
theme_bw()
```



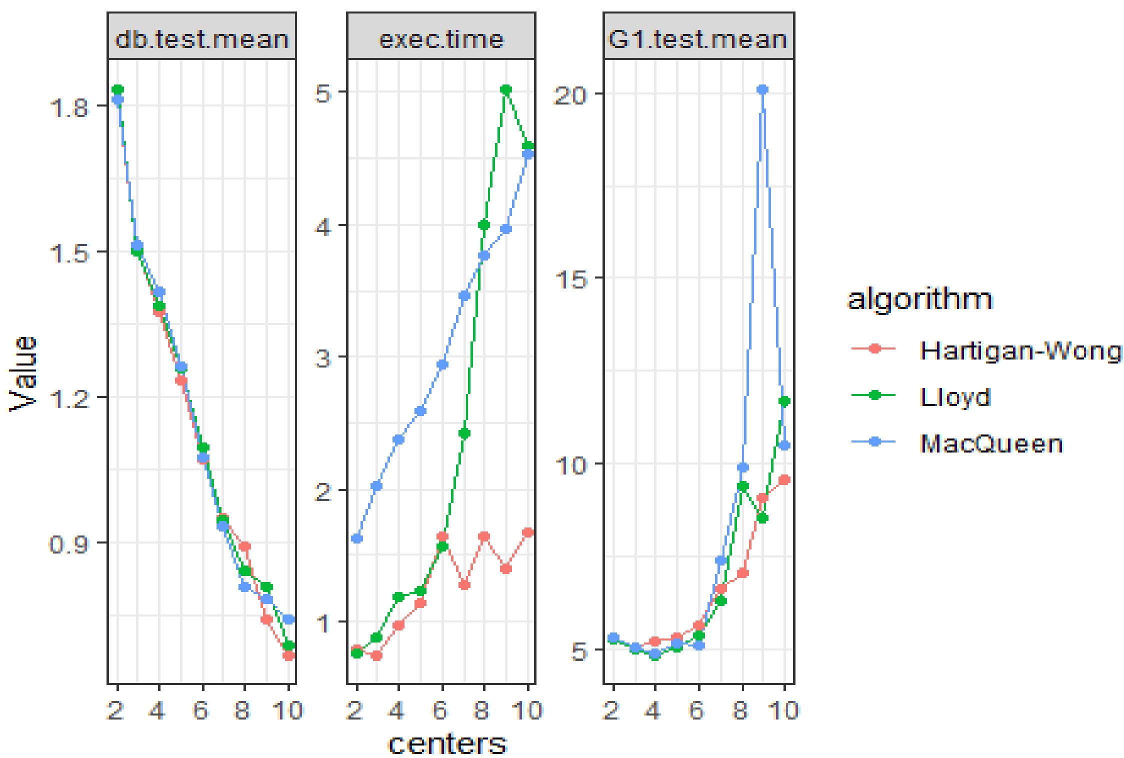

```r
set.seed(1234)

dfTask <- makeClusterTask(data = df_C132)
listLearners("cluster")$class

##  [1] "cluster.cmeans"          "cluster.Cobweb"
##  [3] "cluster.dbscan"          "cluster.EM"
##  [5] "cluster.FarthestFirst"   "cluster.kkmeans"
##  [7] "cluster.kmeans"          "cluster.MiniBatchKmeans"
##  [9] "cluster.SimpleKMeans"    "cluster.XMeans"

kMeans <- makeLearner("cluster.kmeans",par.vals = list(iter.max = 1000,
nstart = 38)) #nstart tomo el 10% del tamaño de las observaciones
kMeans

## Learner cluster.kmeans from package stats,clue
## Type: cluster
## Name: K-Means; Short name: kmeans
## Class: cluster.kmeans
## Properties: numerics,prob
```

```
## Predict-Type: response
## Hyperparameters: centers=2,iter.max=1e+03,nstart=38
```

Declaración de proceso

```
kMeansParamSpace <- makeParamSet(
makeDiscreteParam("centers", values = 2:10), #Consideremos máximo 8
grupos (2x4). Que es lo que veo
makeDiscreteParam("algorithm",
values = c("Hartigan-Wong", "Lloyd", "MacQueen")))
gridSearch <- makeTuneControlGrid()
kFold <- makeResampleDesc("CV", iters = 20)

set.seed(123)
tunedK <- tuneParams(kMeans, task = dfTask,
resampling = kFold,
par.set = kMeansParamSpace,
control = gridSearch,
measures = list(db, G1))
# debemos buscar el min db.test
tunedK

## Tune result:
## Op. pars: centers=10; algorithm=Hartigan-Wong
## db.test.mean=0.6651873,G1.test.mean=9.5849806

set.seed(1237)

kMeansTuningData <- generateHyperParsEffectData(tunedK)
kMeansTuningData$data

##    centers     algorithm db.test.mean G1.test.mean iteration exec.time
## 1        2 Hartigan-Wong    1.8161960     5.288435         1      0.78
## 2        3 Hartigan-Wong    1.5041067     5.027136         2      0.74
## 3        4 Hartigan-Wong    1.3763950     5.165722         3      0.97
## 4        5 Hartigan-Wong    1.2315576     5.311520         4      1.14
## 5        6 Hartigan-Wong    1.0707787     5.635000         5      1.64
## 6        7 Hartigan-Wong    0.9511661     6.629886         6      1.28
## 7        8 Hartigan-Wong    0.8908311     7.057153         7      1.64
## 8        9 Hartigan-Wong    0.7417761     9.055023         8      1.39
## 9       10 Hartigan-Wong    0.6651873     9.584981         9      1.67
## 10       2         Lloyd    1.8374917     5.230543        10      0.75
## 11       3         Lloyd    1.4997895     4.986219        11      0.87
## 12       4         Lloyd    1.3863929     4.812976        12      1.18
## 13       5         Lloyd    1.2602399     5.051003        13      1.23
## 14       6         Lloyd    1.0943853     5.335652        14      1.57
## 15       7         Lloyd    0.9458891     6.298377        15      2.42
## 16       8         Lloyd    0.8394862     9.391513        16      3.99
## 17       9         Lloyd    0.8071079     8.527463        17      5.03
## 18      10         Lloyd    0.6863253    11.674218        18      4.59
## 19       2      MacQueen    1.8161960     5.288435        19      1.62
```

```
## 20        3       MacQueen     1.5144990      5.003489          20        2.03
## 21        4       MacQueen     1.4192010      4.847606          21        2.38
## 22        5       MacQueen     1.2608201      5.158941          22        2.59
## 23        6       MacQueen     1.0758086      5.087126          23        2.94
## 24        7       MacQueen     0.9309144      7.361258          24        3.46
## 25        8       MacQueen     0.8076431      9.894033          25        3.77
## 26        9       MacQueen     0.7802679     20.132005          26        3.96
## 27       10       MacQueen     0.7404481     10.504346          27        4.53

gatheredTuningData <- gather(kMeansTuningData$data,
key = "Metric",
value = "Value",
c(-centers, -iteration, -algorithm))

ggplot(gatheredTuningData, aes(centers, Value, col = algorithm)) +
facet_wrap(~ Metric, scales = "free_y") +
geom_line() +
geom_point() +
theme_bw()
```



No existe como tal un mínimo, conforme incrementas el número de Clusters, baja el estadístico db, mientras que en los otros dos indicadores no hay un máximo com otal en ningún algoritmo, solo con excepción de MacQueen en G1 aquí si se logra marcar un máximo en con 9 centros.

La recomendación de este método es tener 10 Grupos ¿qué pasa si le hacemos caso?

```
set.seed(1237)

tunedKMeans <- setHyperPars(kMeans, par.vals = tunedK$x)
tunedKMeansModel <- train(tunedKMeans, dfTask)
kMeansModelData <- getLearnerModel(tunedKMeansModel)
kMeansModelData$iter

## [1] 4

tunedKMeans

## Learner cluster.kmeans from package stats,clue
## Type: cluster
## Name: K-Means; Short name: kmeans
## Class: cluster.kmeans
## Properties: numerics,prob
## Predict-Type: response
## Hyperparameters:
centers=10,iter.max=1e+03,nstart=38,algorithm=Hartigan-Wong

df_C132_1 <- mutate(df_C132,
kMeansCluster = as.factor(kMeansModelData$cluster))

df_C132_1=mutate(df_C132_1,
Sleep.Disorder = as.factor(df$Sleep.Disorder))

head(df_C132_1)

## # A tibble: 6 × 25
##     Age Sleep.Duration Quality.of.Sleep Physical.Activity.Level
Stress.Level
##   <dbl>          <dbl>            <dbl>                   <dbl>
<dbl>
## 1 -1.75          -1.30            -1.10                  -0.824
0.347
## 2 -1.64          -1.17            -1.10                   0.0398
1.47
## 3 -1.64          -1.17            -1.10                   0.0398
1.47
## 4 -1.64          -1.55            -2.77                  -1.40
1.47
## 5 -1.64          -1.55            -2.77                  -1.40
1.47
## 6 -1.64          -1.55            -2.77                  -1.40
1.47
## # ℹ 20 more variables: sistolica_bp <dbl>, diastolica_bp <dbl>,
## #   Heart.Rate <dbl>, Daily.Steps <dbl>, Gender_Male <dbl>,
## #   Occupation_Doctor <dbl>, Occupation_Engineer <dbl>,
```

```
## #   Occupation_Lawyer <dbl>, Occupation_Manager <dbl>,
Occupation_Nurse <dbl>,
## #   Occupation_Sales.Representative <dbl>, Occupation_Salesperson
<dbl>,
## #   Occupation_Scientist <dbl>, Occupation_Software.Engineer <dbl>,
## #   Occupation_Teacher <dbl>, BMI.Category_Normal.Weight <dbl>, …
```

```
table(df_C132_1$kMeansCluster)
```

```
##
##  1  2  3  4  5  6  7  8  9 10
## 65 73 65  9 60  4 41  4 19 34
```

```
table(df_C132_1$Sleep.Disorder)
```

```
##
##    Insomnia        None Sleep Apnea
##          77         219          78
```

Por lo tanto, hacerle caso al método de aplicar 10 grupos pulveriza la información cuándo se compara con los grupos de la variable respuesta que teníamos en la información original; posiblemente los grupos 1,2,3 podrían asociarse a las observaciones con individuos con Insomia o Apnea.

Los que no se ve en ningún lado, son aquellos individuos que no sufren algún padecimiento, este grupo podría estar contaminando el resto de la información al poder contener más grupo o ninguno.

Con este método no tenemos conclusión

###Otra forma de explorar (kMeans)

Tomaremos la base previamente armada, solamente para evitar hacer ajustes en el código hacemos este recipe dummy para mantener el mismo nombre

```
 rec_df  <- recipe(~., data = df_C132)



 #rec_df <- recipe( ~ ., data = df) %>%
  # update_role(Sleep.Disorder,new_role = 'id') %>%
  #step_dummy(all_nominal_predictors()) %>%      # Variables categóricas
a dummies
  #step_normalize(all_numeric_predictors())
rec_df
```

Preparación de la Data

```
prepped_data <- prep(rec_df) %>% bake(new_data = NULL)
```

```
dim(prepped_data)
```

```
## [1] 374  23
```

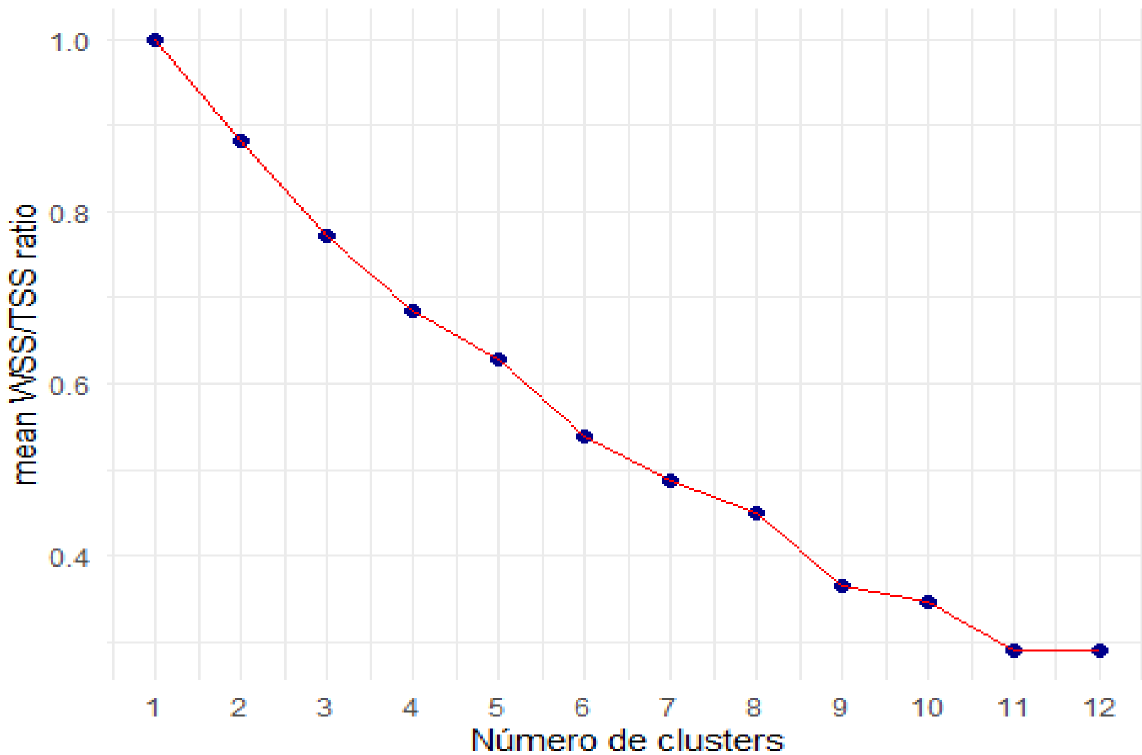
```
df1=prepped_data
```

Se realiza la corrida del ajuste hasta con 12 clusters con objetivo de validar si no hay un tope en los Clusters que puedes formar con la información actual

```
kmeans_spec <- k_means(num_clusters = tune())

kmeans_wf <- workflow(rec_df, kmeans_spec)

kmeans_wf <- kmeans_wf %>%
  update_model(kmeans_spec)

grid <- tibble(num_clusters = 1:12)

set.seed(123)
boots <- bootstraps(df1, times = 12)

res <- tune_cluster(
  kmeans_wf,
  resamples = boots,
  grid = grid,
  metrics = cluster_metric_set(sse_within_total, sse_total, sse_ratio)
)

res_metrics <- collect_metrics(res)%>% print(n=Inf)

## # A tibble: 36 × 7
##    num_clusters .metric          .estimator     mean     n   std_err
.config
##           <int> <chr>            <chr>         <dbl> <int>     <dbl>
<chr>
##  1            1 sse_ratio        standard      1        12   0
Preprocess…
##  2            1 sse_total        standard   8604.       12 152.
Preprocess…
##  3            1 sse_within_total standard   8604.       12 152.
Preprocess…
##  4            2 sse_ratio        standard      0.882    12   0.0118
Preprocess…
##  5            2 sse_total        standard   8604.       12 152.
Preprocess…
##  6            2 sse_within_total standard   7575.       12 121.
Preprocess…
##  7            3 sse_ratio        standard      0.772    12   0.00761
Preprocess…
##  8            3 sse_total        standard   8604.       12 152.
Preprocess…
##  9            3 sse_within_total standard   6642.       12 151.
Preprocess…
## 10            4 sse_ratio        standard      0.685    12   0.00946
```

```
Preprocess…
## 11           4 sse_total        standard   8604.       12 152.
Preprocess…
## 12           4 sse_within_total standard   5901.       12 155.
Preprocess…
## 13           5 sse_ratio        standard      0.628    12   0.0106
Preprocess…
## 14           5 sse_total        standard   8604.       12 152.
Preprocess…
## 15           5 sse_within_total standard   5412.       12 150.
Preprocess…
## 16           6 sse_ratio        standard      0.540    12   0.0131
Preprocess…
## 17           6 sse_total        standard   8604.       12 152.
Preprocess…
## 18           6 sse_within_total standard   4659.       12 179.
Preprocess…
## 19           7 sse_ratio        standard      0.487    12   0.0139
Preprocess…
## 20           7 sse_total        standard   8604.       12 152.
Preprocess…
## 21           7 sse_within_total standard   4186.       12 132.
Preprocess…
## 22           8 sse_ratio        standard      0.448    12   0.0149
Preprocess…
## 23           8 sse_total        standard   8604.       12 152.
Preprocess…
## 24           8 sse_within_total standard   3872.       12 179.
Preprocess…
## 25           9 sse_ratio        standard      0.365    12   0.0115
Preprocess…
## 26           9 sse_total        standard   8604.       12 152.
Preprocess…
## 27           9 sse_within_total standard   3140.       12 120.
Preprocess…
## 28          10 sse_ratio        standard      0.346    12   0.0144
Preprocess…
## 29          10 sse_total        standard   8604.       12 152.
Preprocess…
## 30          10 sse_within_total standard   2993.       12 166.
Preprocess…
## 31          11 sse_ratio        standard      0.290    12   0.00988
Preprocess…
## 32          11 sse_total        standard   8604.       12 152.
Preprocess…
## 33          11 sse_within_total standard   2499.       12 102.
Preprocess…
## 34          12 sse_ratio        standard      0.290    12   0.00874
Preprocess…
## 35          12 sse_total        standard   8604.       12 152.
```

```
Preprocess…
## 36          12 sse_within_total standard   2496.        12  89.7
Preprocess…

res_metrics %>%
  filter(.metric == "sse_ratio") %>%
  ggplot(aes(x = num_clusters, y = mean)) +
  geom_point(col="darkblue",size=2) +
  geom_line(col="red") +
  theme_minimal() +
  ylab("mean WSS/TSS ratio") +
  xlab("Número de clusters") +
  scale_x_continuous(breaks = 1:12)
```



De acuerdo a encontrar un SSE_Ratio bajo o que el decremento sea mínimo es con 12 Grupos, con 10 Clustres tambipen se ve una convergencia, pero vuelve a decrecer con 11; en número previo de grupos si se ve que decrece en "picada"

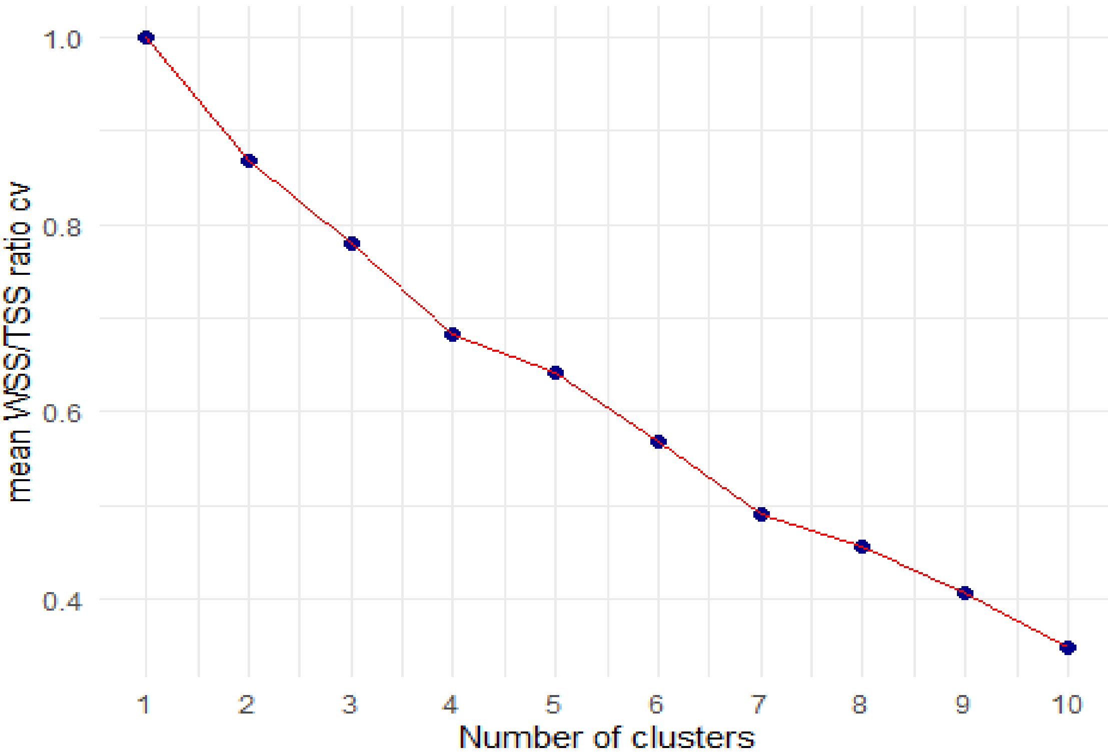
Validacion cruzada

```
df_cv <- vfold_cv(df1, v = 12) # lo divide en 10 segmentos (o "folds")

clust_num_grid <- grid_regular(num_clusters(),levels = 12)

#clust_num_grid

res1 <- tune_cluster(
```

```
  kmeans_wf,
  resamples = df_cv,
  grid = clust_num_grid,
  control = control_grid(save_pred = TRUE, extract = identity),
  metrics = cluster_metric_set(sse_within_total, sse_total, sse_ratio)
)

res1_metrics <- res1 %>% collect_metrics()%>% print(n=Inf)

## # A tibble: 30 × 7
##    num_clusters .metric          .estimator     mean     n   std_err
.config
##           <int> <chr>            <chr>         <dbl> <int>     <dbl>
<chr>
##  1            1 sse_ratio        standard      1        12   0
Preprocess…
##  2            1 sse_total        standard   7862.       12  53.7
Preprocess…
##  3            1 sse_within_total standard   7862.       12  53.7
Preprocess…
##  4            2 sse_ratio        standard      0.869    12   0.00901
Preprocess…
##  5            2 sse_total        standard   7862.       12  53.7
Preprocess…
##  6            2 sse_within_total standard   6835.       12  96.4
Preprocess…
##  7            3 sse_ratio        standard      0.779    12   0.0126
Preprocess…
##  8            3 sse_total        standard   7862.       12  53.7
Preprocess…
##  9            3 sse_within_total standard   6129.       12 108.
Preprocess…
## 10            4 sse_ratio        standard      0.683    12   0.00768
Preprocess…
## 11            4 sse_total        standard   7862.       12  53.7
Preprocess…
## 12            4 sse_within_total standard   5371.       12  85.7
Preprocess…
## 13            5 sse_ratio        standard      0.641    12   0.00862
Preprocess…
## 14            5 sse_total        standard   7862.       12  53.7
Preprocess…
## 15            5 sse_within_total standard   5042.       12  87.8
Preprocess…
## 16            6 sse_ratio        standard      0.569    12   0.00784
Preprocess…
## 17            6 sse_total        standard   7862.       12  53.7
Preprocess…
## 18            6 sse_within_total standard   4469.       12  67.7
Preprocess…
```

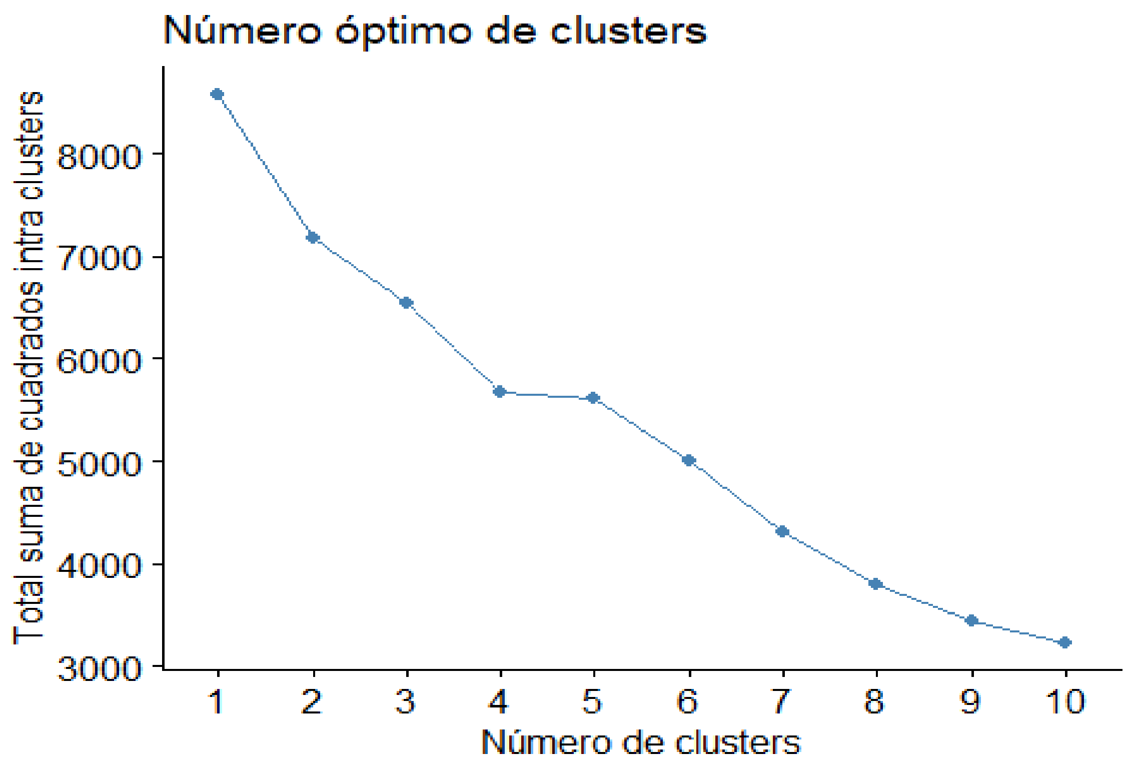
```
## 19          7 sse_ratio        standard      0.491    12   0.00715
Preprocess…
## 20          7 sse_total        standard   7862.       12  53.7
Preprocess…
## 21          7 sse_within_total standard   3861.       12  63.7
Preprocess…
## 22          8 sse_ratio        standard      0.456    12   0.00622
Preprocess…
## 23          8 sse_total        standard   7862.       12  53.7
Preprocess…
## 24          8 sse_within_total standard   3586.       12  53.8
Preprocess…
## 25          9 sse_ratio        standard      0.406    12   0.0104
Preprocess…
## 26          9 sse_total        standard   7862.       12  53.7
Preprocess…
## 27          9 sse_within_total standard   3195.       12  97.8
Preprocess…
## 28         10 sse_ratio        standard      0.349    12   0.00972
Preprocess…
## 29         10 sse_total        standard   7862.       12  53.7
Preprocess…
## 30         10 sse_within_total standard   2745.       12  87.1
Preprocess…
```

```
res1_metrics %>%
  filter(.metric == "sse_ratio") %>%
  ggplot(aes(x = num_clusters, y = mean)) +
  geom_point(col="darkblue",size=2) +
  geom_line(col="red") +
  theme_minimal() +
  ylab("mean WSS/TSS ratio cv") +
  xlab("Number of clusters") +
  scale_x_continuous(breaks = 1:12)
```

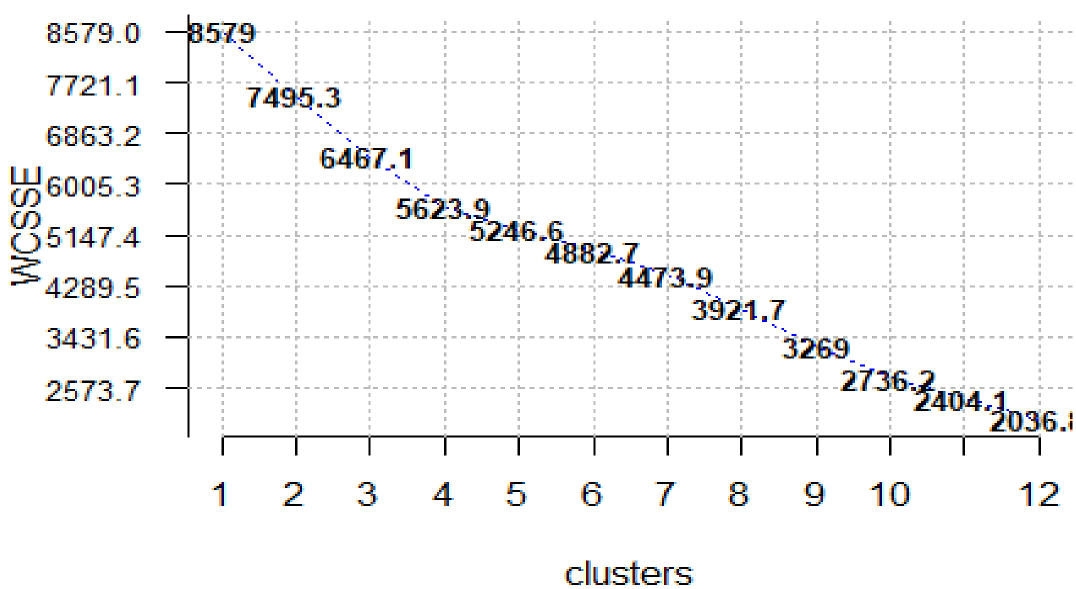


Aún con validación cruzada tenemos una caída acelerada del sse_ratio, no podemos concluir.

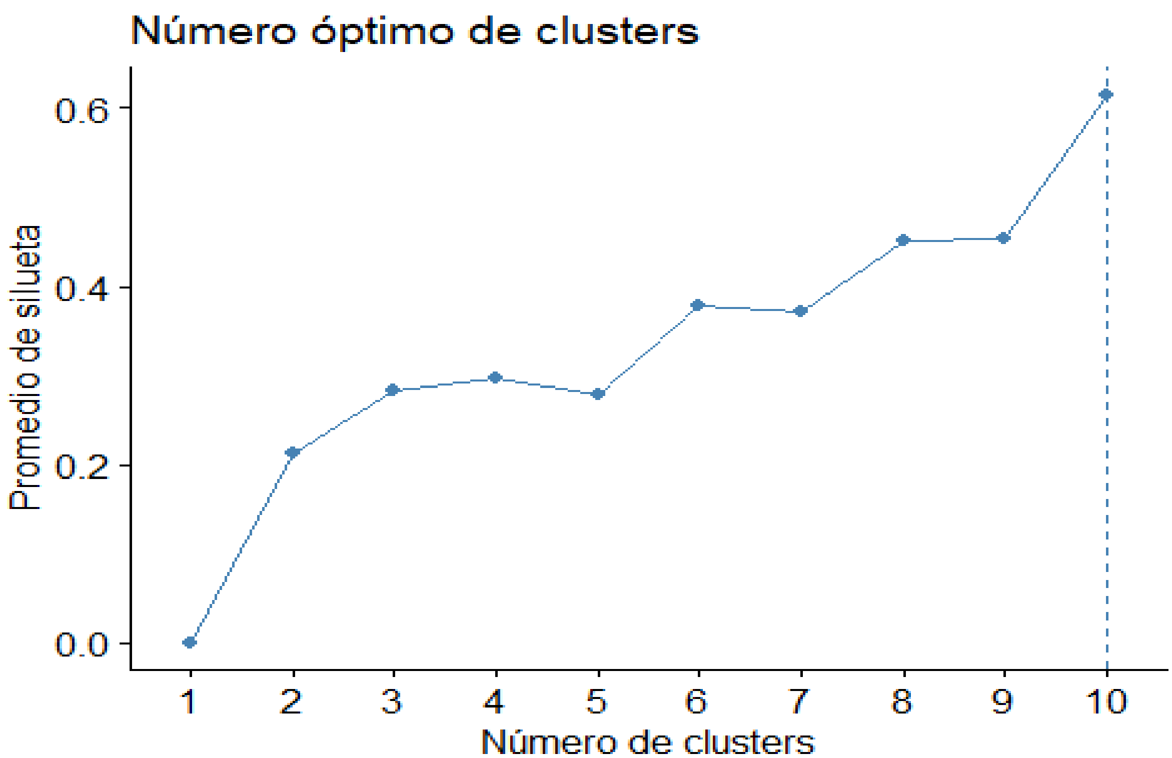
```
fviz_nbclust(df1, kmeans, method = "wss")+labs(x ="Número de
clusters")+labs(y="Total suma de cuadrados intra clusters")+labs(title =
"Número óptimo de clusters")
```



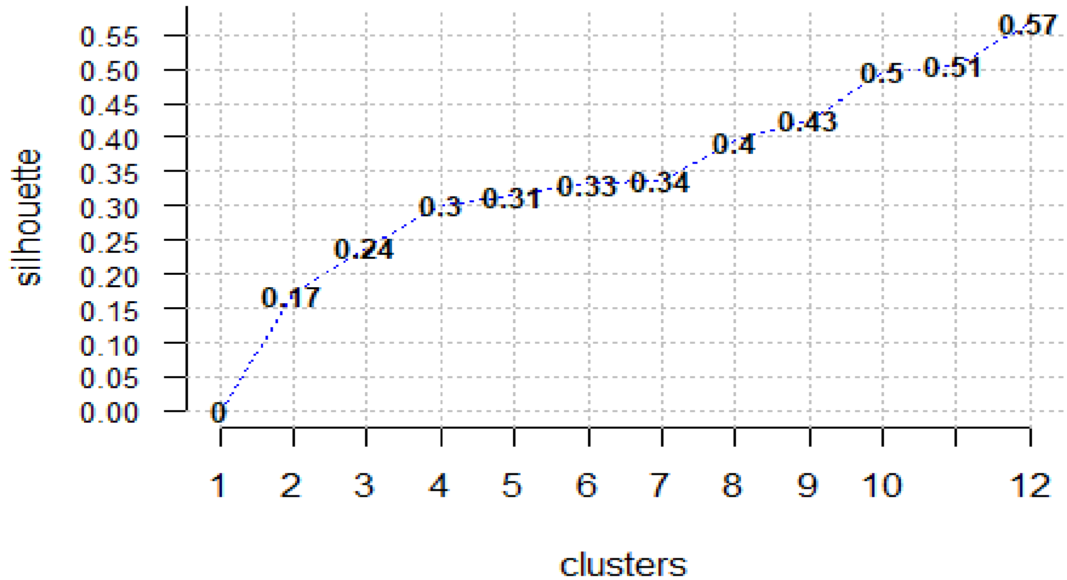

```
opt<-Optimal_Clusters_KMeans(df1, max_clusters=12,plot_clusters =
TRUE,criterion="WCSSE")
```



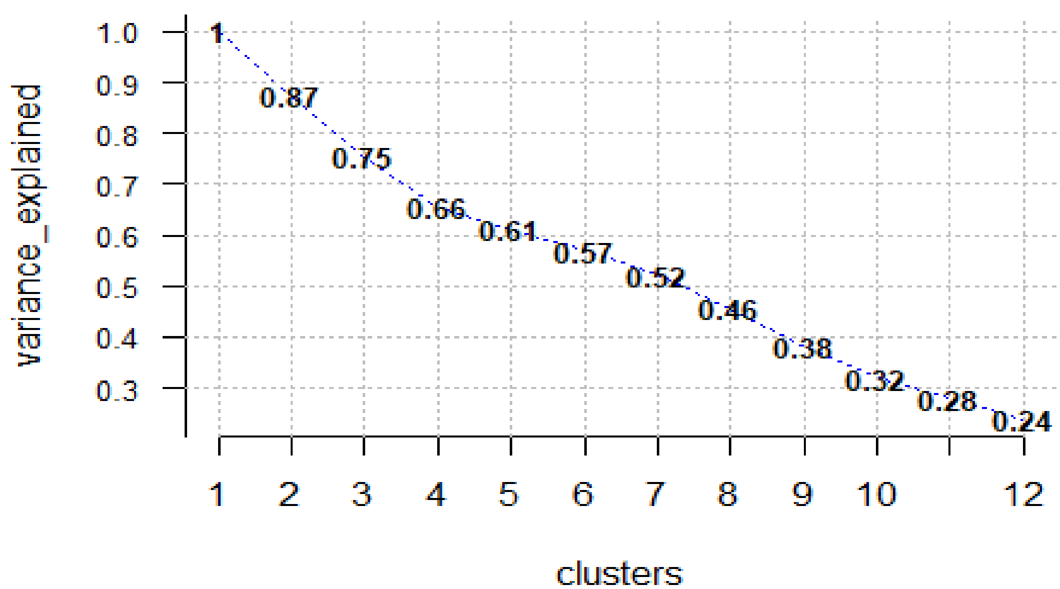
```r
fviz_nbclust(df1, kmeans, method = "silhouette")+labs(x ="Número de clusters")+labs(y="Promedio de silueta")+labs(title = "Número óptimo de clusters")
```



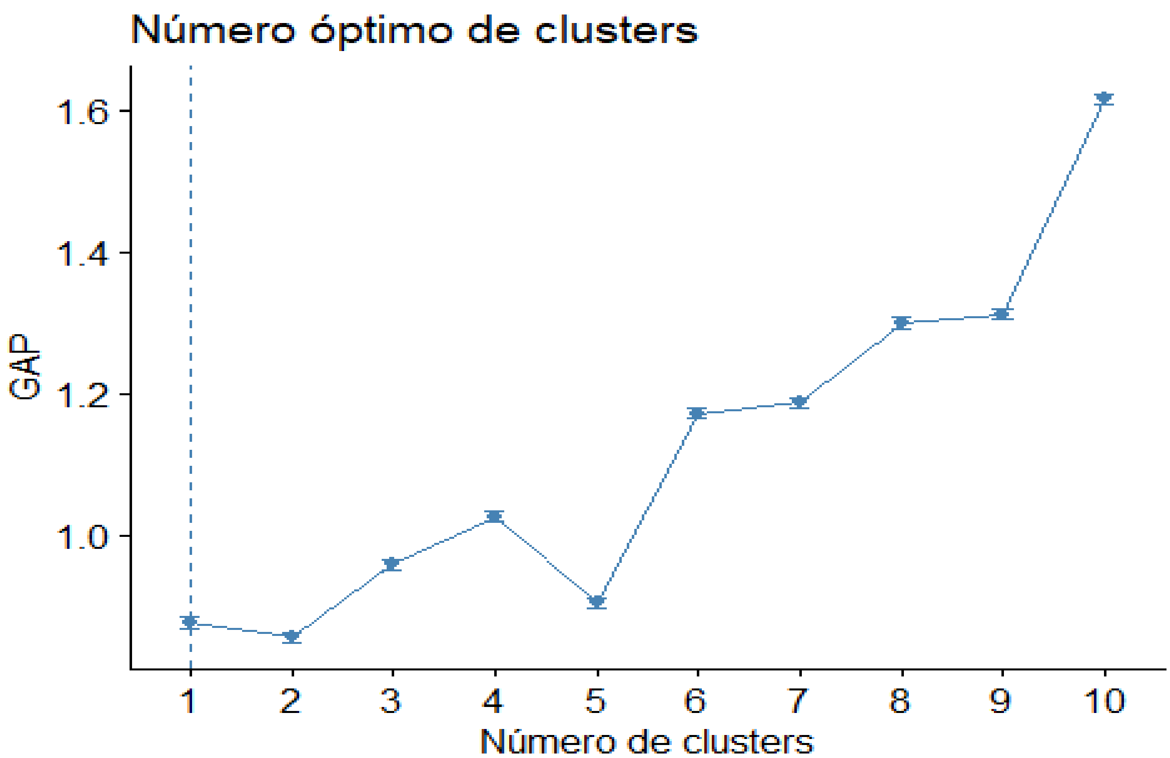

```r
opt1<-Optimal_Clusters_KMeans(df1, max_clusters=12, plot_clusters = TRUE, criterion="silhouette")
```




```
opt2<-Optimal_Clusters_KMeans(df1, max_clusters=12, plot_clusters = TRUE,
criterion = "variance_explained",fK_threshold = 0.90)
```

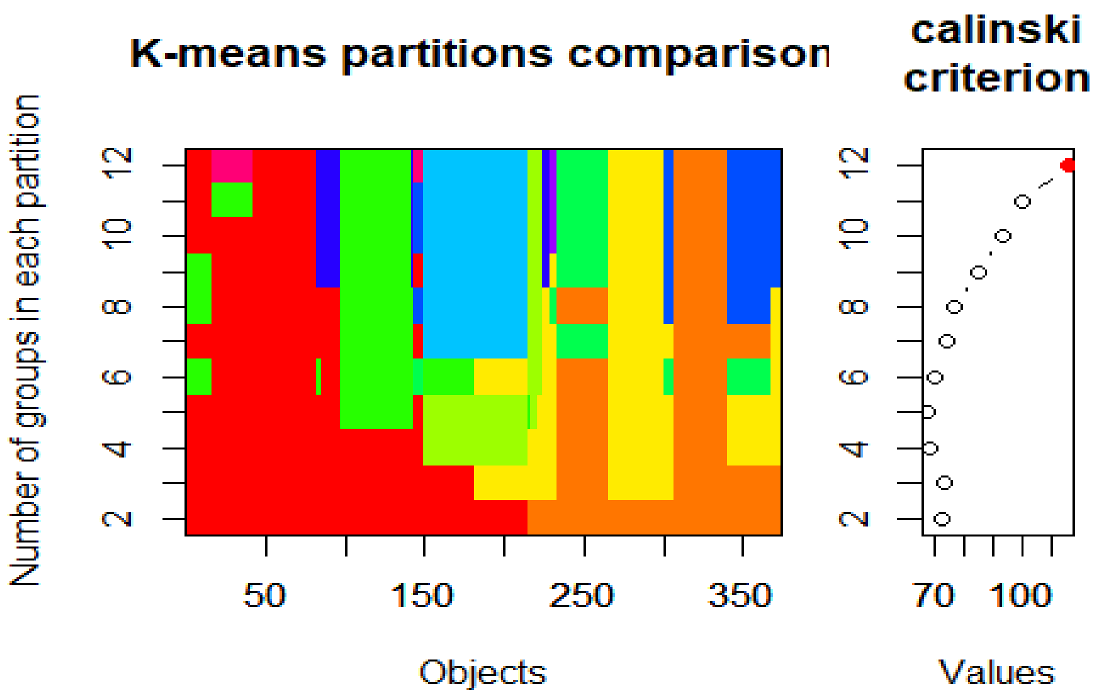


```
fviz_nbclust(df1, kmeans, method = "gap_stat")+labs(x ="Número de
clusters")+labs(y="GAP")+labs(title = "Número óptimo de clusters")
```

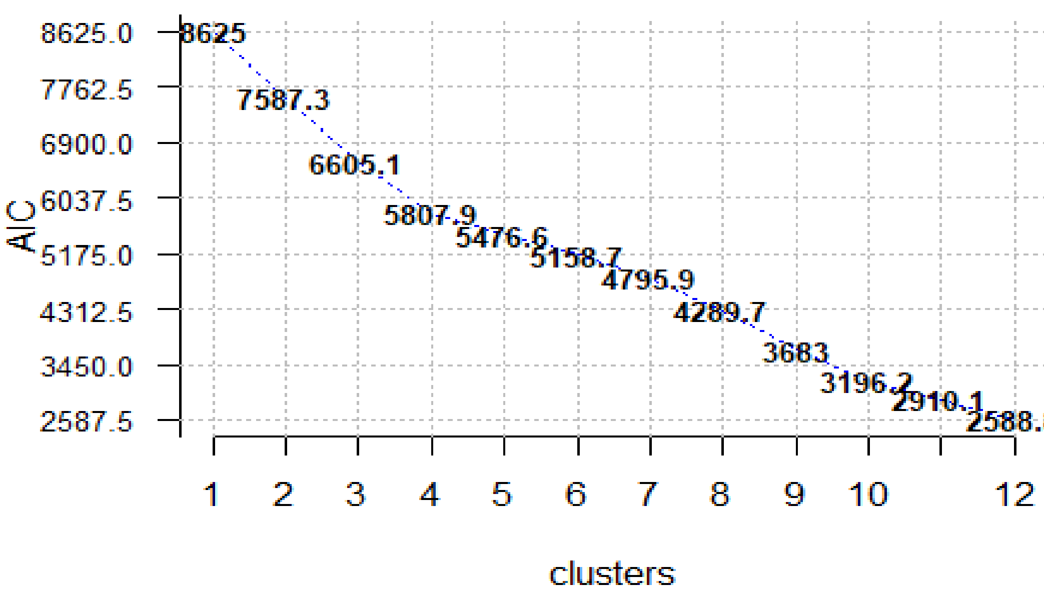


Sigue habiendo decrementos agresivos, sin embargo sucede entre la formación de 4-6 grupos ya que en ese rango se estabiliza los errores, sin embargo el promedio de Silueta recomienda 10 clusters... GAP dice que 1. No hay suficiente claridad

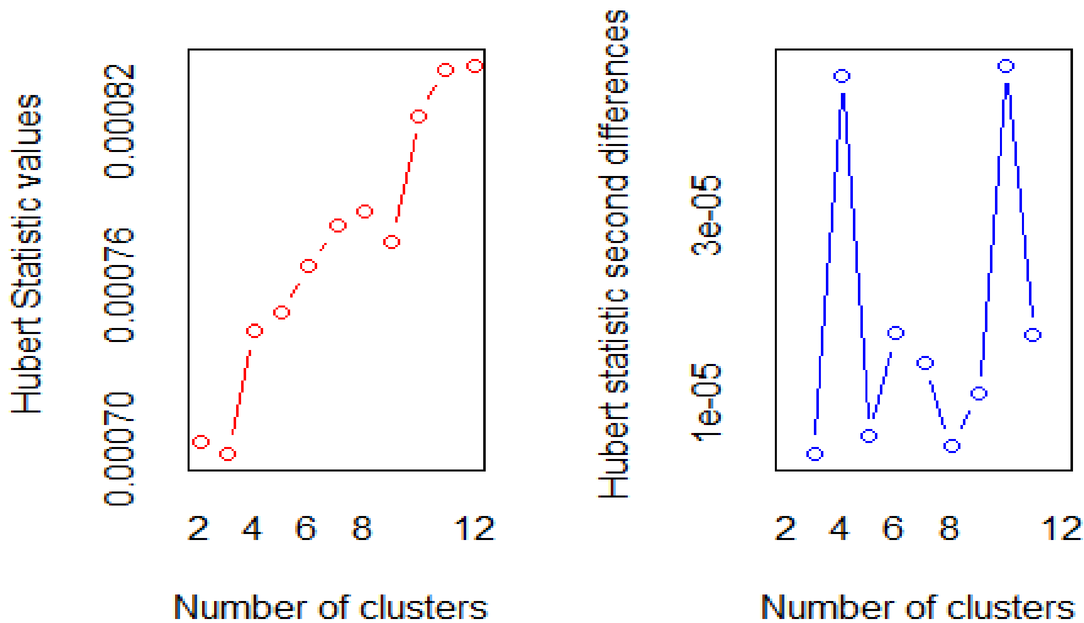
```
fit <- cascadeKM(df1, 2, 12, iter = 500)
plot(fit, sortg = TRUE, grpmts.plot = TRUE)
```



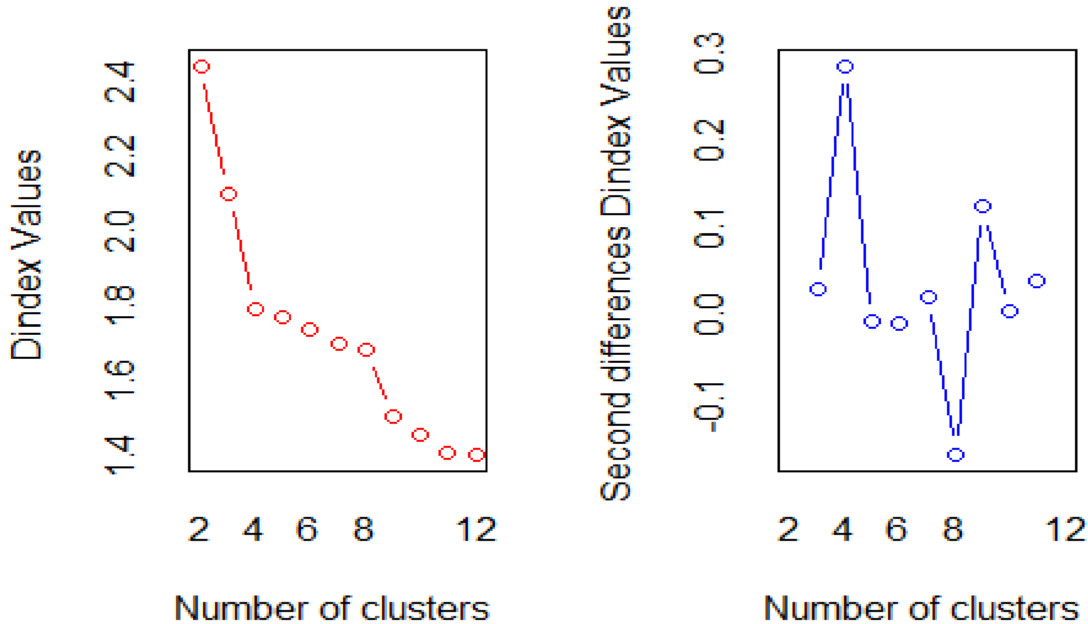

```
opt_aic<-Optimal_Clusters_KMeans(df1, 12, 'euclidean',
plot_clusters=TRUE,criterion="AIC")
```



```r
nb <- NbClust(df1[,1:9], distance = "euclidean", min.nc = 2, max.nc = 12,
method = "single", index ="all")
```




```
## *** : The Hubert index is a graphical method of determining the number
of clusters.
##                 In the plot of Hubert index, we seek a significant
knee that corresponds to a
##                 significant increase of the value of the measure i.e
the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                  In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                  second differences plot) that corresponds to a significant increase of the value of
##                  the measure.
##
## *******************************************************************
## * Among all indices:
## * 3 proposed 2 as the best number of clusters
## * 2 proposed 3 as the best number of clusters
## * 11 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 3 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 1 proposed 12 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *******************************************************************

names(nb)
```

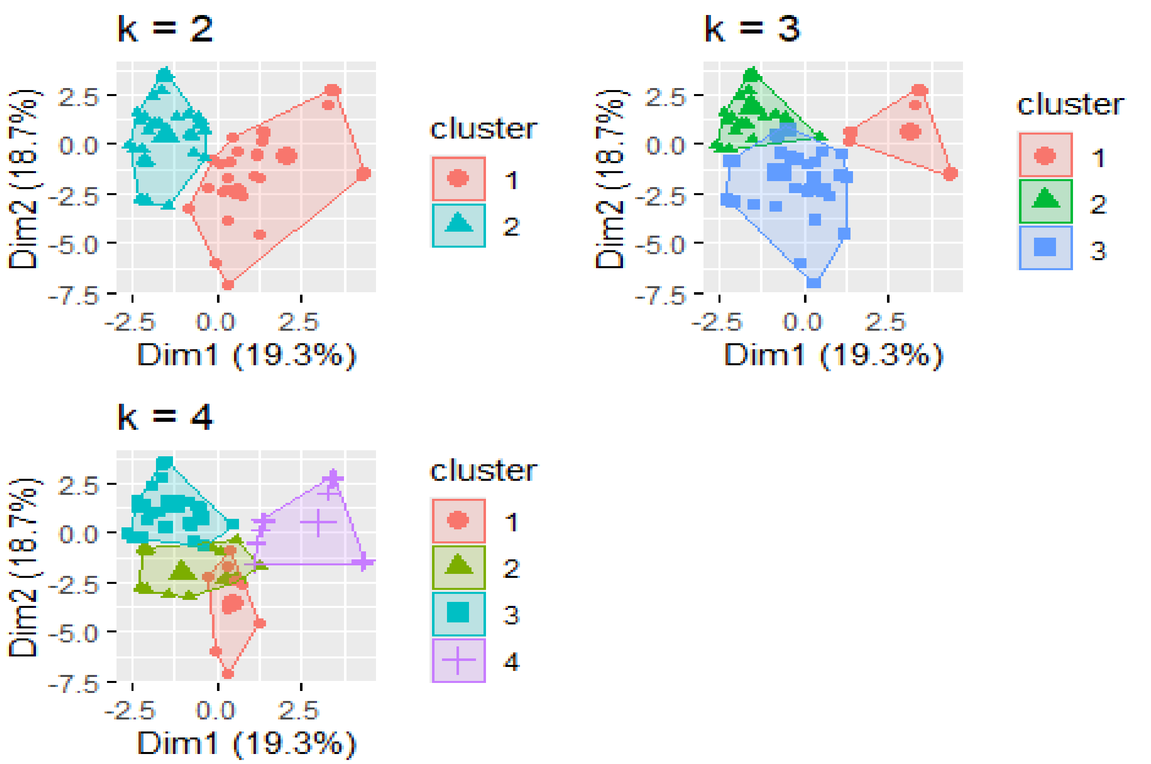
```
## [1] "All.index"          "All.CriticalValues" "Best.nc"
## [4] "Best.partition"
```

Las Cascada y AIC recomienda 12 clusters

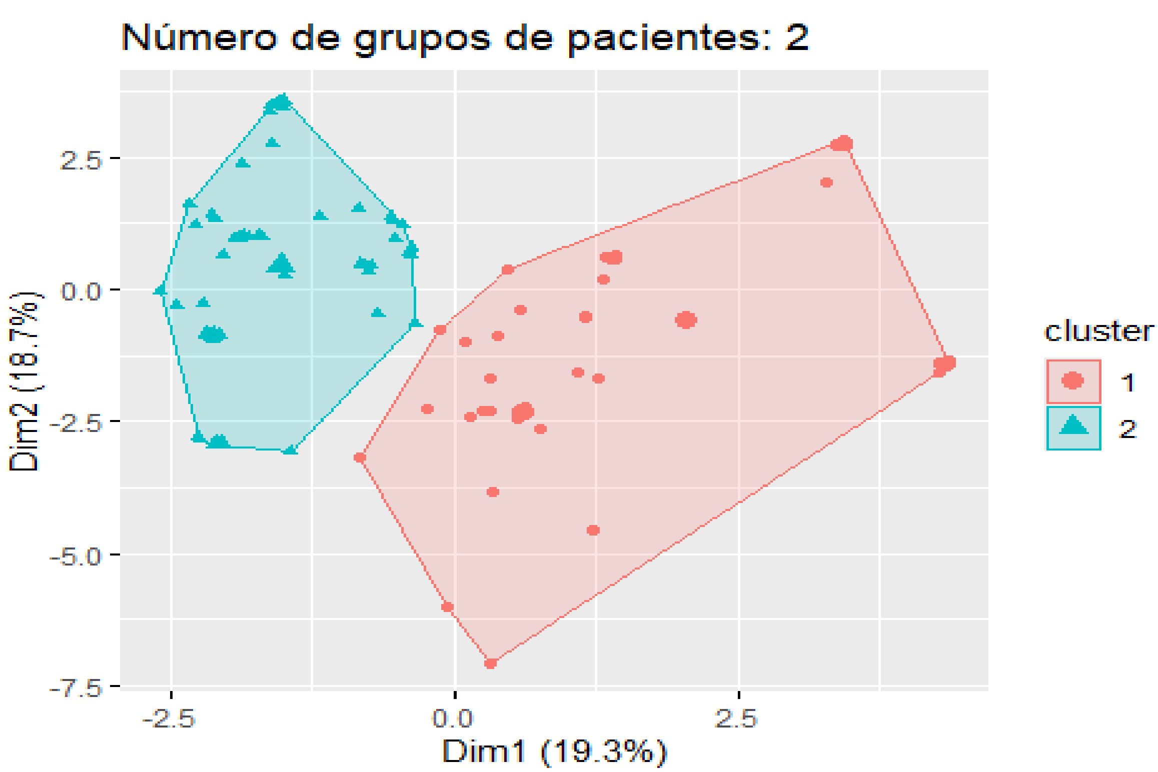
De acuerdo al método NB Clus, hay 11 propuestas con 4 Clusters, por lo tanto la recomendación es tener 4 Clusters

Si bien, la recomendación es tener 4 Clusters, exploremos como se ve con 2 y 3 con la intención de comparar la recomendación con la realidad y considerando

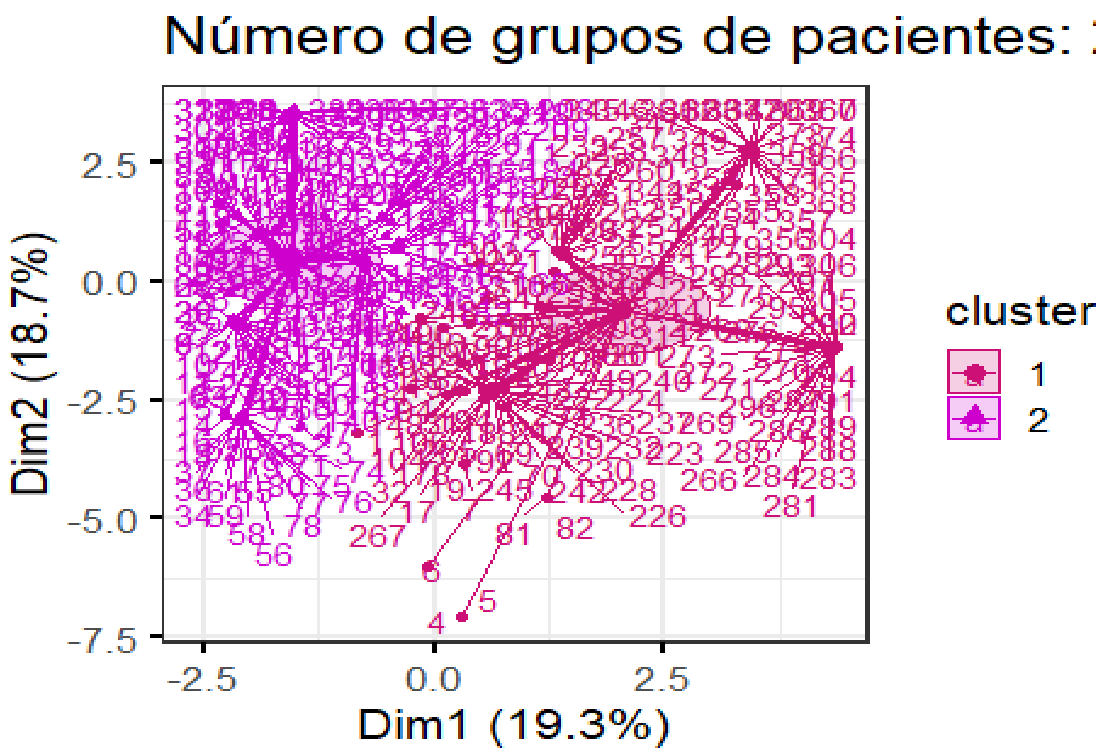
```
k2 <- kmeans(df1, centers = 2, nstart = 25)
k3 <- kmeans(df1, centers = 3, nstart = 25)
k4 <- kmeans(df1, centers = 4, nstart = 25)

k2$tot.withinss/k2$totss; k3$tot.withinss/k3$totss;
k4$tot.withinss/k4$totss

## [1] 0.836569

## [1] 0.7359433

## [1] 0.656108

p2 <- fviz_cluster(k2, geom = "point", data = df1)+ ggtitle("k = 2")
p3 <- fviz_cluster(k3, geom = "point",  data = df1) + ggtitle("k = 3")
p4 <- fviz_cluster(k4, geom = "point",  data = df1) + ggtitle("k = 4")



grid.arrange(p2, p3, p4,nrow=2, ncol = 2)
```



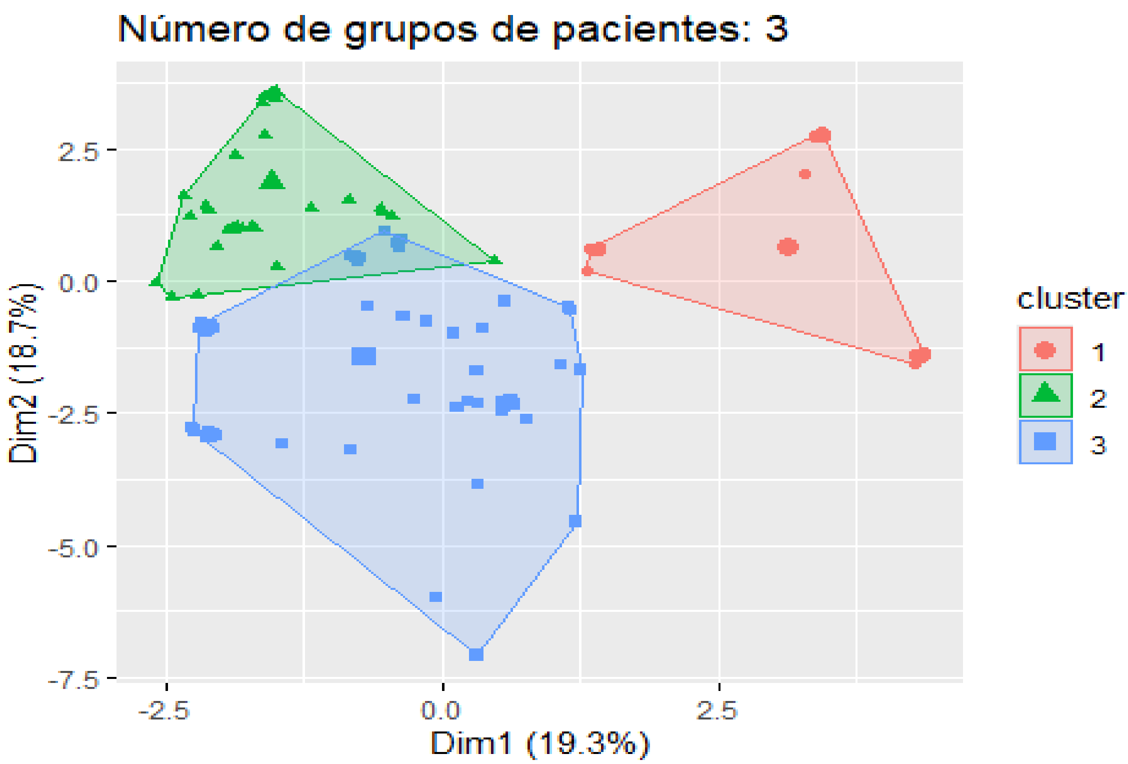

```r
fviz_cluster(k2, geom = "point",  data = df1) + ggtitle("Número de grupos de pacientes: 2")
```



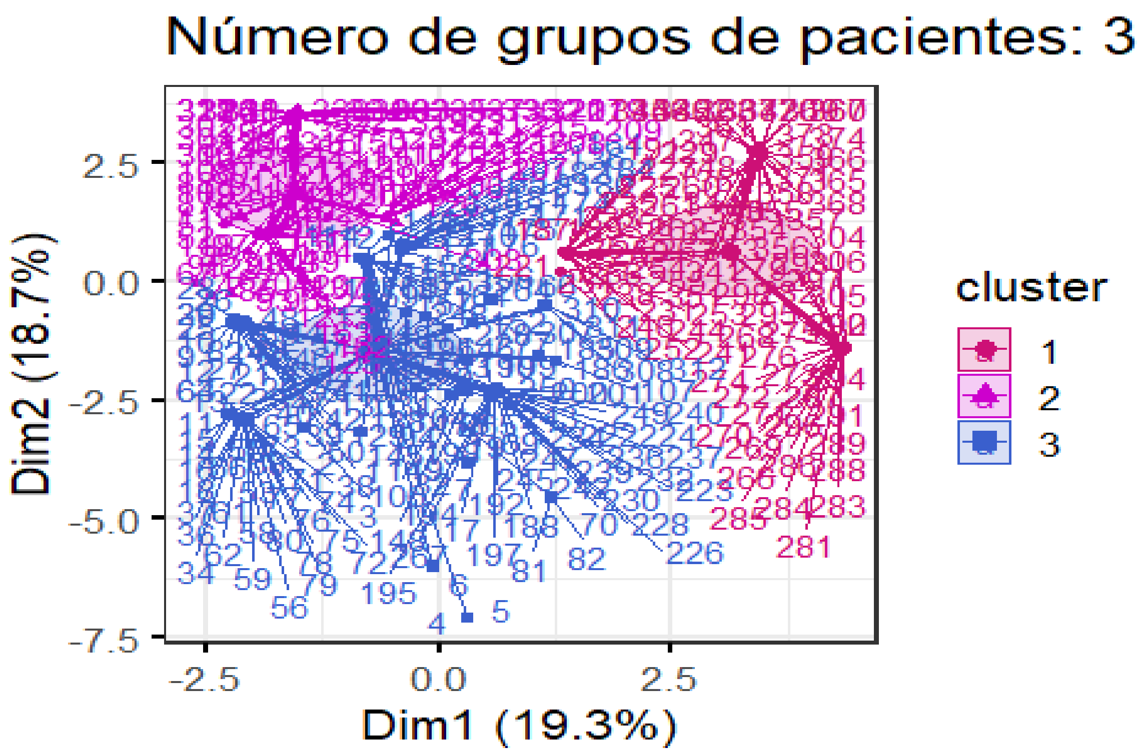
```
fviz_cluster(k2, data = df1,
             palette=c("deeppink3", "magenta3"),
             ellipse.type = "euclid",
             star.plot = T,
             repel = T,
             ggtheme = theme())+ ggtitle("Número de grupos de pacientes: 2")
```



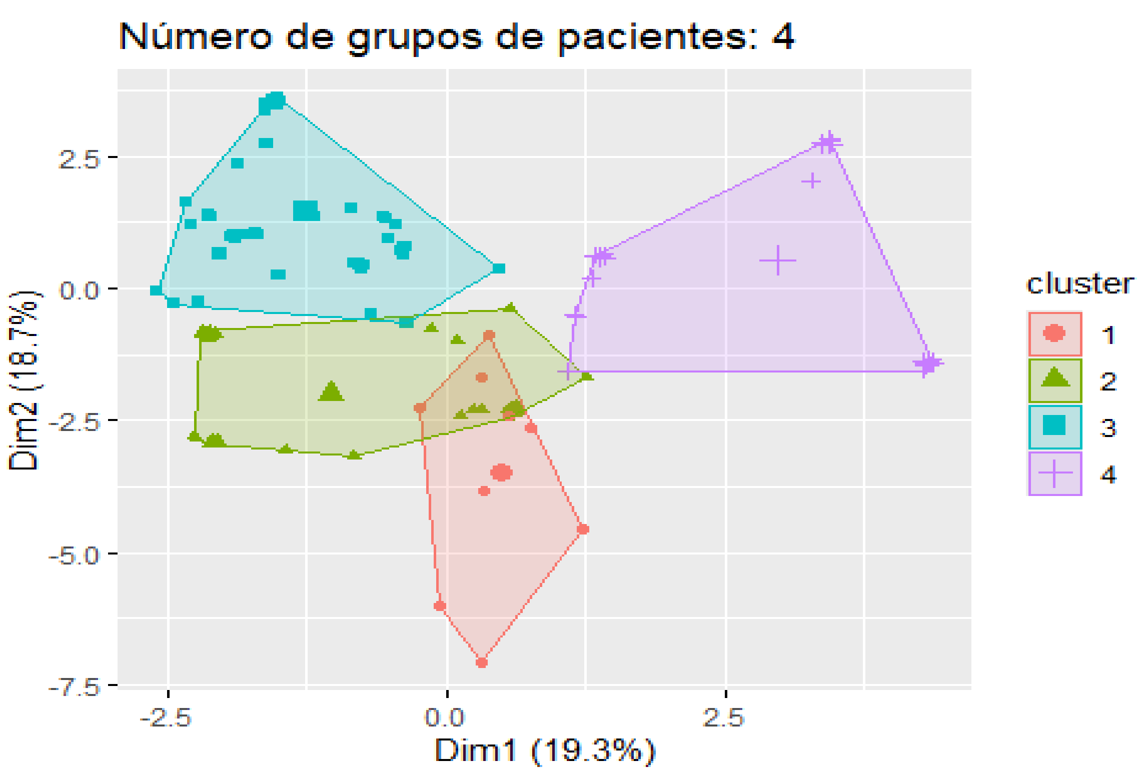

```
require(tibble)

fviz_cluster(k3, geom = "point",  data = df1) + ggtitle("Número de grupos de pacientes: 3")
```



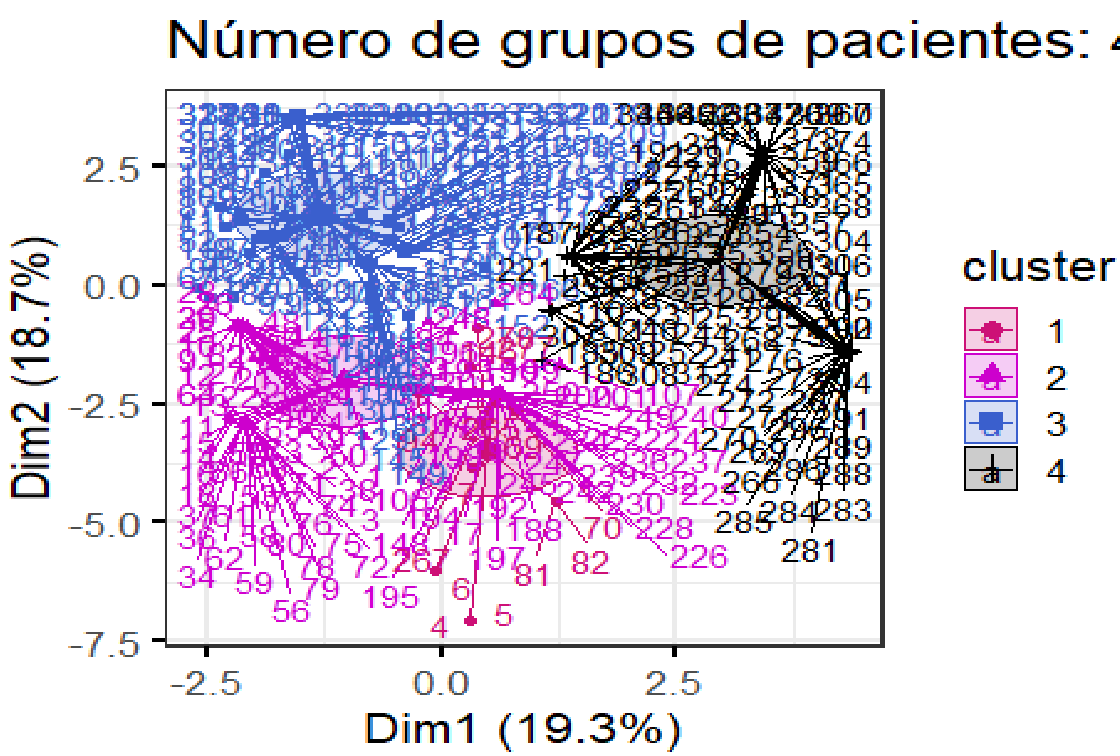

```
fviz_cluster(k3, data = df1,
             palette=c("deeppink3", "magenta3", "royalblue3"),
             ellipse.type = "euclid",
             star.plot = T,
             repel = T,
             ggtheme = theme())+ ggtitle("Número de grupos de pacientes:
3")
```




```r
require(tibble)

fviz_cluster(k4, geom = "point",  data = df1) + ggtitle("Número de grupos de pacientes: 4")
```

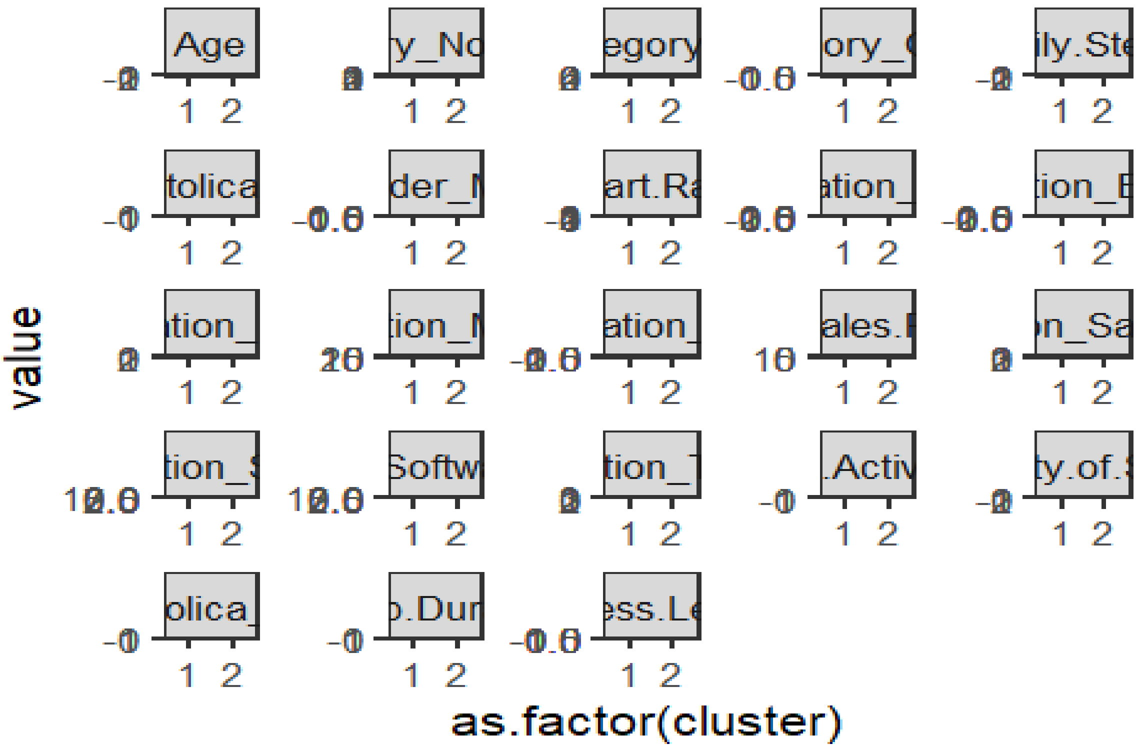


```
fviz_cluster(k4, data = df1,
             palette=c("deeppink3", "magenta3", "royalblue3", "black"),
             ellipse.type = "euclid",
             star.plot = T,
             repel = T,
             ggtheme = theme())+ ggtitle("Número de grupos de pacientes: 
4")
```



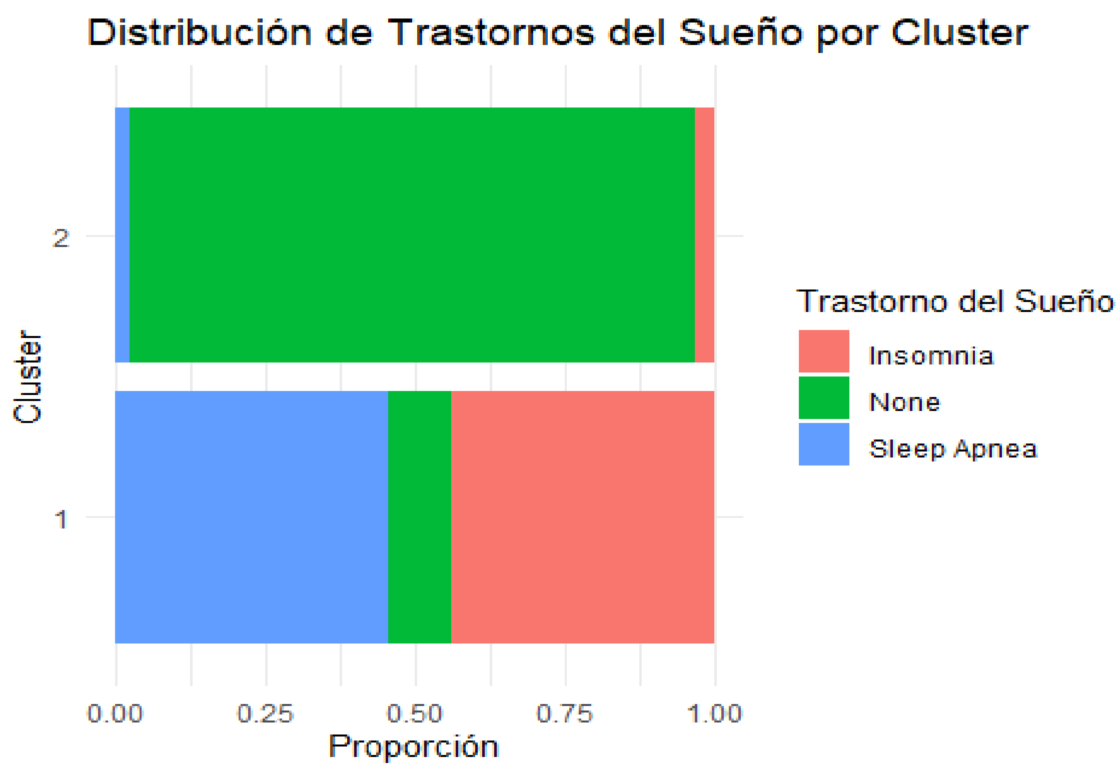

```
require(tibble)

k2 %>%
  extract_centroids()%>% as_tibble() %>% print(width=Inf)

## # A tibble: 2 × 24
##   .cluster     Age Sleep.Duration Quality.of.Sleep 
Physical.Activity.Level
##   <fct>      <dbl>          <dbl>            <dbl>                   
<dbl>
## 1 Cluster_1  0.549         -0.459           -0.418                  
0.0497
## 2 Cluster_2 -0.410          0.343            0.313                  -
0.0372
##   Stress.Level sistolica_bp diastolica_bp Heart.Rate Daily.Steps 
Gender_Male
##          <dbl>        <dbl>         <dbl>      <dbl>       <dbl>       
<dbl>
## 1        0.216        0.878         0.902      0.442     -0.0884
```

```
-0.410
## 2       -0.162       -0.656        -0.674     -0.330      0.0661
0.307
##   Occupation_Doctor Occupation_Engineer Occupation_Lawyer
Occupation_Manager
##               <dbl>               <dbl>             <dbl>
<dbl>
## 1            -0.420              -0.416            -0.341
0.0692
## 2             0.314               0.311             0.255            -
0.0517
##   Occupation_Nurse Occupation_Sales.Representative
Occupation_Salesperson
##              <dbl>                           <dbl>
<dbl>
## 1            0.626                          0.0979
0.409
## 2           -0.468                         -0.0732                 -
0.305
##   Occupation_Scientist Occupation_Software.Engineer Occupation_Teacher
##                  <dbl>                        <dbl>              <dbl>
## 1                0.139                       0.0175              0.341
## 2               -0.104                      -0.0131             -0.255
##   BMI.Category_Normal.Weight BMI.Category_Obese
BMI.Category_Overweight
##                        <dbl>              <dbl>
<dbl>
## 1                     -0.108              0.221                  1.04
## 2                      0.0808            -0.166                 -
0.780

kmeans_clusters2 <-
  bind_cols(df1, cluster=k2$cluster)

kmeans_clusters2 %>%
  pivot_longer(-cluster) %>%
  ggplot(aes(x = as.factor(cluster), y = value, fill =
as.factor(cluster))) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(name), scales = "free")
```




```
kmeans_clusters2 %>%
  group_by(cluster) %>%
  summarise(num_users = n()) %>%
  mutate(pct_users = num_users / sum(num_users))

## # A tibble: 2 × 3
##   cluster num_users pct_users
##     <int>     <int>     <dbl>
## 1       1       160     0.428
## 2       2       214     0.572

table(df$Sleep.Disorder)

##
##    Insomnia        None Sleep Apnea
##          77         219          78

table(kmeans_clusters2$cluster)

##
##   1   2
## 160 214

cluster_assignments2= cbind(kmeans_clusters2$cluster,df$Sleep.Disorder)
%>% data.frame()

colnames(cluster_assignments2)=c('Cluster','Sleep.Disorder')
```

```r
summary_plot <- ggplot(cluster_assignments2, aes(x = Cluster, fill =
Sleep.Disorder)) +
  geom_bar(position = "fill") +
  labs(
    title = "Distribución de Trastornos del Sueño por Cluster",
    x = "Cluster",
    y = "Proporción",
    fill = "Trastorno del Sueño"
  ) +
  coord_flip() +
  theme_minimal()

print(summary_plot)
```

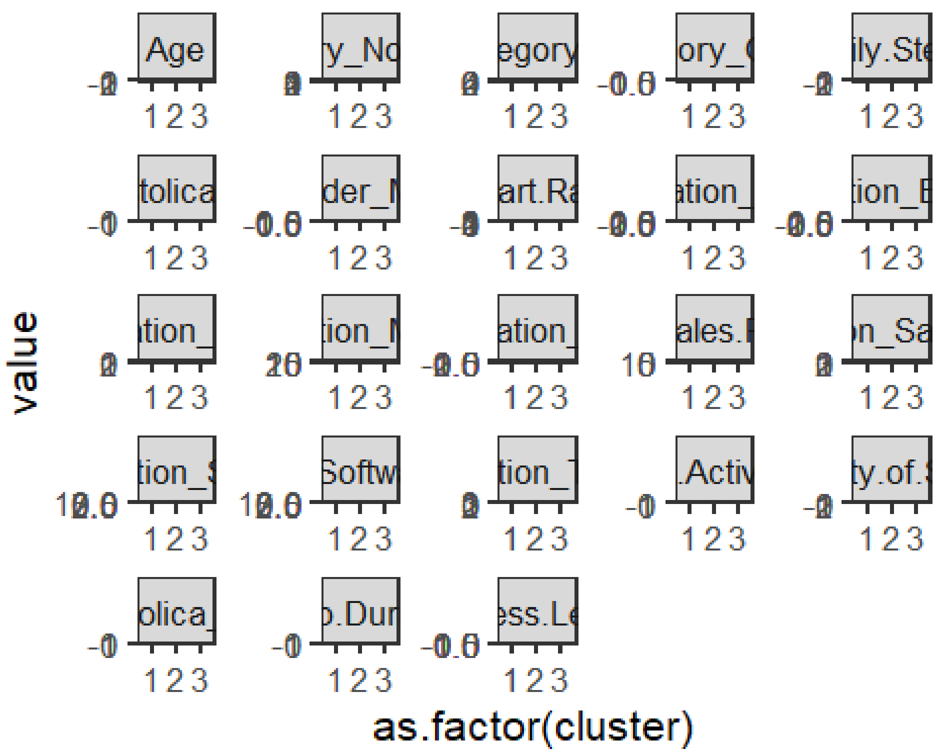


El grupo 2 se asocia con el grupo None, pero el grupo 1 tiene una mezcla de Insomnia y Disociación de sueño; podríamos decir que son personas con algún padecimiento y sin padecimiento.

```r
k3 %>%
  extract_centroids()%>% as_tibble() %>% print(width=Inf)

## # A tibble: 3 × 24
##   .cluster     Age Sleep.Duration Quality.of.Sleep
Physical.Activity.Level
##   <fct>      <dbl>          <dbl>            <dbl>
<dbl>
## 1 Cluster_1  1.03          -0.230           0.0474
0.572
```

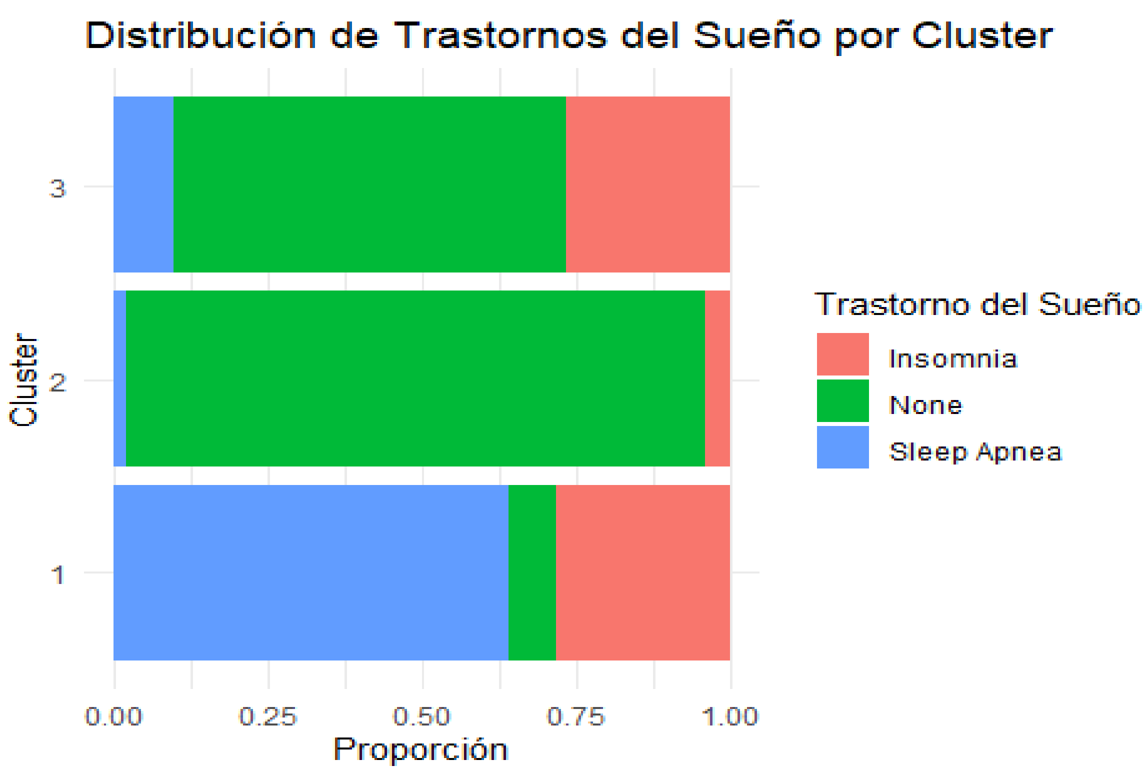
```
## 2 Cluster_2  0.103          0.739           0.837                   -
0.138
## 3 Cluster_3 -0.598         -0.319          -0.521                   -
0.216
##   Stress.Level sistolica_bp diastolica_bp Heart.Rate Daily.Steps
Gender_Male
##          <dbl>        <dbl>         <dbl>      <dbl>       <dbl>
<dbl>
## 1       -0.162        1.29           1.44      -0.127       0.550
-1.01
## 2       -0.818       -0.890        -0.926     -0.662      -0.180
-0.401
## 3        0.569       -0.141        -0.200      0.459      -0.179
0.762
##   Occupation_Doctor Occupation_Engineer Occupation_Lawyer
Occupation_Manager
##               <dbl>               <dbl>             <dbl>
<dbl>
## 1            -0.483              -0.449            -0.379            -
0.0517
## 2            -0.435               1.10             -0.379            -
0.0517
## 3             0.509              -0.419             0.421
0.0576
##   Occupation_Nurse Occupation_Sales.Representative
Occupation_Salesperson
##              <dbl>                           <dbl>
<dbl>
## 1             1.29                         -0.0732                  -
0.305
## 2           -0.420                         -0.0732                  -
0.305
## 3           -0.421                          0.0815
0.340
##   Occupation_Scientist Occupation_Software.Engineer Occupation_Teacher
##                  <dbl>                        <dbl>              <dbl>
## 1               -0.104                       -0.104              0.603
## 2               -0.104                        0.0811            -0.161
## 3                0.116                        0.00587           -0.218
##   BMI.Category_Normal.Weight BMI.Category_Obese
BMI.Category_Overweight
##                        <dbl>              <dbl>
<dbl>
## 1                     -0.244             -0.166                  1.23
## 2                      0.417             -0.166                  -
0.789
## 3                     -0.121              0.184                  -
0.174
```

```r
kmeans_clusters3 <-
  bind_cols(df1, cluster=k3$cluster)

kmeans_clusters3 %>%
  pivot_longer(-cluster) %>%
  ggplot(aes(x = as.factor(cluster), y = value, fill =
as.factor(cluster))) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(name), scales = "free")
```




```r
kmeans_clusters3 %>%
  group_by(cluster) %>%
  summarise(num_users = n()) %>%
  mutate(pct_users = num_users / sum(num_users))

## # A tibble: 3 × 3
##   cluster num_users pct_users
##     <int>     <int>     <dbl>
## 1       1        92     0.246
## 2       2       105     0.281
## 3       3       177     0.473

table(df$Sleep.Disorder)

##
##    Insomnia        None Sleep Apnea
##          77         219          78
```

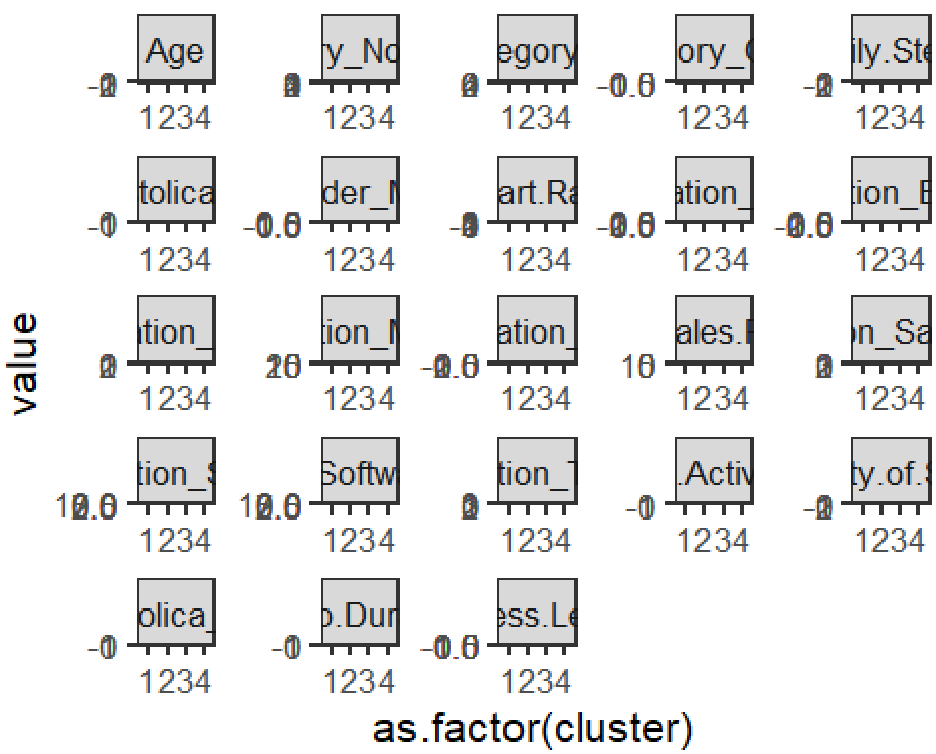
```
table(kmeans_clusters3$cluster)

##
##   1   2   3
##  92 105 177

cluster_assignments3= cbind(kmeans_clusters3$cluster,df$Sleep.Disorder)
%>% data.frame()

colnames(cluster_assignments3)=c('Cluster','Sleep.Disorder')

summary_plot <- ggplot(cluster_assignments3, aes(x = Cluster, fill =
Sleep.Disorder)) +
  geom_bar(position = "fill") +
  labs(
    title = "Distribución de Trastornos del Sueño por Cluster",
    x = "Cluster",
    y = "Proporción",
    fill = "Trastorno del Sueño"
  ) +
  coord_flip() +
  theme_minimal()

print(summary_plot)
```



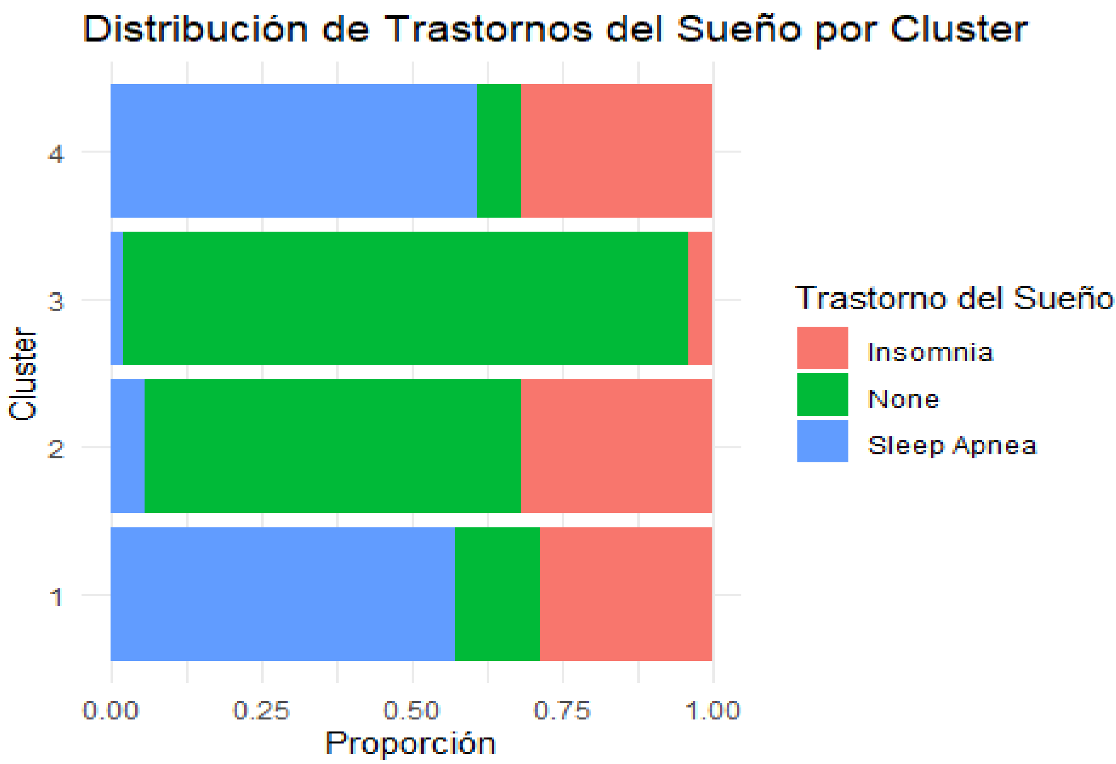
Con tres grupos, la mezcla del grupo 1 se mantiene con una merma en padecimiento de Insomio, pero los que no tienen padecimiento se separan en dos grupos

```
k4 %>%
  extract_centroids()%>% as_tibble() %>% print(width=Inf)

## # A tibble: 4 × 24
##   .cluster      Age Sleep.Duration Quality.of.Sleep
Physical.Activity.Level
##   <fct>       <dbl>          <dbl>            <dbl>
<dbl>
## 1 Cluster_1 -0.631          -0.561         -1.10                       -
0.392
## 2 Cluster_2 -0.819          -0.538         -0.877                      -
0.478
## 3 Cluster_3 -0.0190          0.622          0.741
0.0724
## 4 Cluster_4  1.02           -0.263          0.00599
0.472
##   Stress.Level sistolica_bp diastolica_bp Heart.Rate Daily.Steps
Gender_Male
##          <dbl>        <dbl>         <dbl>      <dbl>       <dbl>
<dbl>
## 1        0.387         1.02          0.683     3.02       -1.79
0.275
## 2        0.874        -0.433        -0.427     0.411      -0.275
0.879
## 3       -0.626        -0.577        -0.638    -0.548       0.0666
0.00263
## 4       -0.0761        1.20          1.33     -0.0522      0.453     -
1.01
##   Occupation_Doctor Occupation_Engineer Occupation_Lawyer
Occupation_Manager
##               <dbl>               <dbl>             <dbl>
<dbl>
## 1             0.244              -0.449            0.0518              -
0.0517
## 2             1.02               -0.401           -0.379
0.124
## 3            -0.449               0.636            0.525               -
0.0517
## 4            -0.483              -0.449           -0.379               -
0.0517
##   Occupation_Nurse Occupation_Sales.Representative
Occupation_Salesperson
##              <dbl>                           <dbl>
<dbl>
## 1           -0.492                            1.88                     -
0.305
## 2           -0.377                           -0.0732
0.733
## 3           -0.441                           -0.0732                   -
0.305
```

```
## 4           1.15                          -0.0732                   -
0.305
##   Occupation_Scientist Occupation_Software.Engineer Occupation_Teacher
##                  <dbl>                        <dbl>              <dbl>
## 1                2.67                         0.590             -0.115
## 2               -0.104                       -0.0156            -0.228
## 3               -0.104                        0.0256            -0.216
## 4               -0.104                       -0.104              0.591
##   BMI.Category_Normal.Weight BMI.Category_Obese
BMI.Category_Overweight
##                        <dbl>              <dbl>
<dbl>
## 1                     -0.244               4.26                 -0.225
## 2                     -0.0858             -0.166                -
0.0470
## 3                      0.248              -0.166                -0.767
## 4                     -0.244              -0.166                 1.23

kmeans_clusters4 <-
  bind_cols(df1, cluster=k4$cluster)

kmeans_clusters4 %>%
  pivot_longer(-cluster) %>%
  ggplot(aes(x = as.factor(cluster), y = value, fill =
as.factor(cluster))) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(name), scales = "free")
```



```r
kmeans_clusters4 %>%
  group_by(cluster) %>%
  summarise(num_users = n()) %>%
  mutate(pct_users = num_users / sum(num_users))

## # A tibble: 4 × 3
##   cluster num_users pct_users
##     <int>     <int>     <dbl>
## 1       1        14    0.0374
## 2       2       110    0.294
## 3       3       150    0.401
## 4       4       100    0.267

table(df$Sleep.Disorder)

##
##    Insomnia        None Sleep Apnea
##          77         219          78

table(kmeans_clusters4$cluster)

##
##   1   2   3   4
##  14 110 150 100

cluster_assignments4= cbind(kmeans_clusters4$cluster,df$Sleep.Disorder)
%>% data.frame()

colnames(cluster_assignments4)=c('Cluster','Sleep.Disorder')

summary_plot <- ggplot(cluster_assignments4, aes(x = Cluster, fill =
Sleep.Disorder)) +
  geom_bar(position = "fill") +
  labs(
    title = "Distribución de Trastornos del Sueño por Cluster",
    x = "Cluster",
    y = "Proporción",
    fill = "Trastorno del Sueño"
  ) +
  coord_flip() +
  theme_minimal()

print(summary_plot)
```

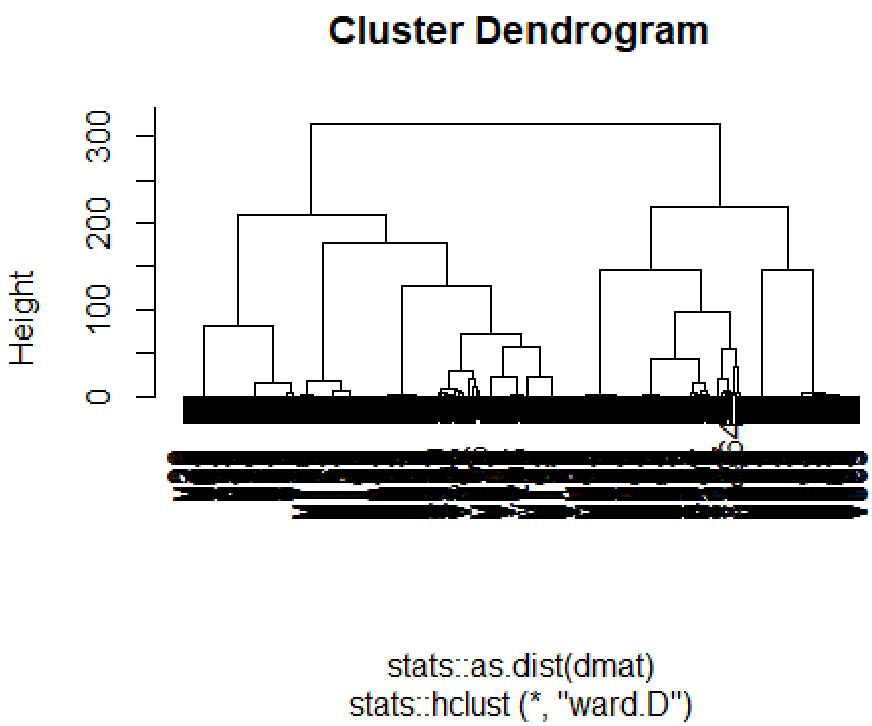


Con los datos proporcionados no hay similitudes para determinar si puede padecer algún transtorno ya que a medida que vamos separando la información los pacientes con algún pedecimiento se separan, lo que tratan de mantenerse son los que no tienen padecimiento.

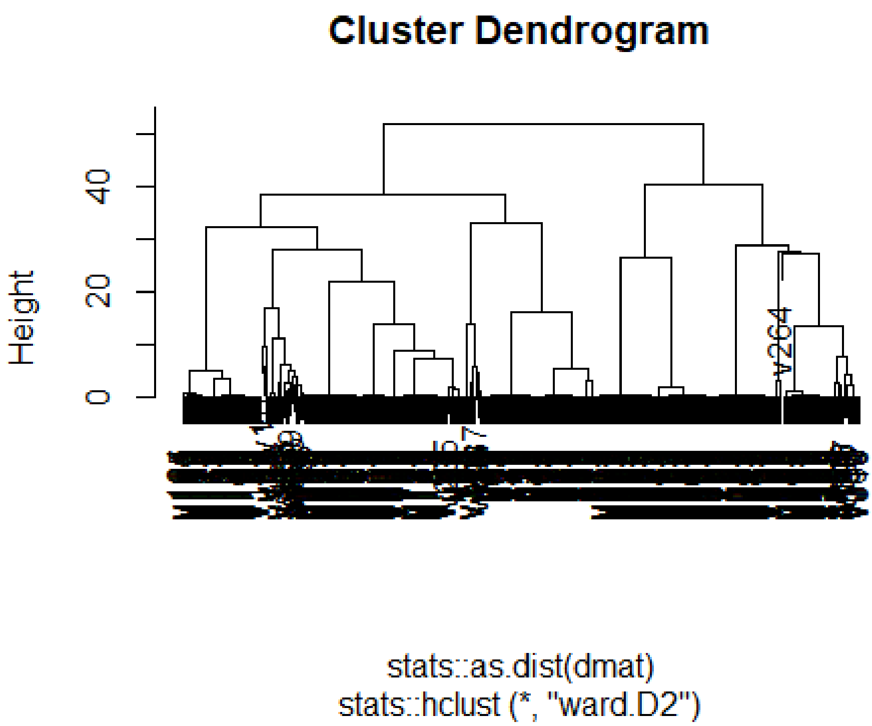
Ajustan PCA y reconstruyendo

Ponemos en recipe que sea al menos 80% de la varianza explicada como criterio del número Componentes

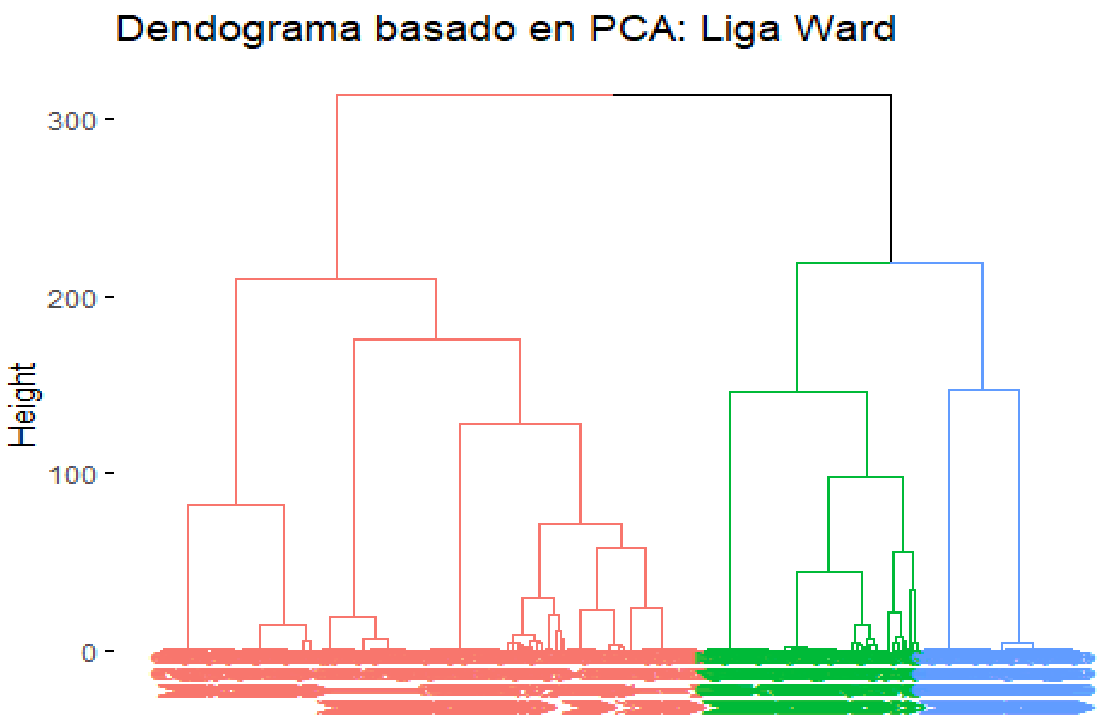
```
df_pca_rec <- recipe(~ ., data = df) %>%
  update_role(Sleep.Disorder, new_role = "id") %>%
  step_dummy(all_nominal_predictors()) %>%
  step_normalize(all_predictors()) %>%
  step_pca(all_predictors(), threshold = 0.80)

df_pca_wf <- workflow() %>%
  add_recipe(df_pca_rec)

df_pca_hier <- df_pca_wf %>%
  add_model(hier_clust(linkage_method = "ward.D")) %>%
  fit(data = df) %>%
  extract_fit_engine() %>%
  plot()
```




```r
df_pca_hier <- df_pca_wf %>%
  add_model(hier_clust(linkage_method = "ward.D2")) %>%
  fit(data = df) %>%
  extract_fit_engine() %>%
  plot()
```



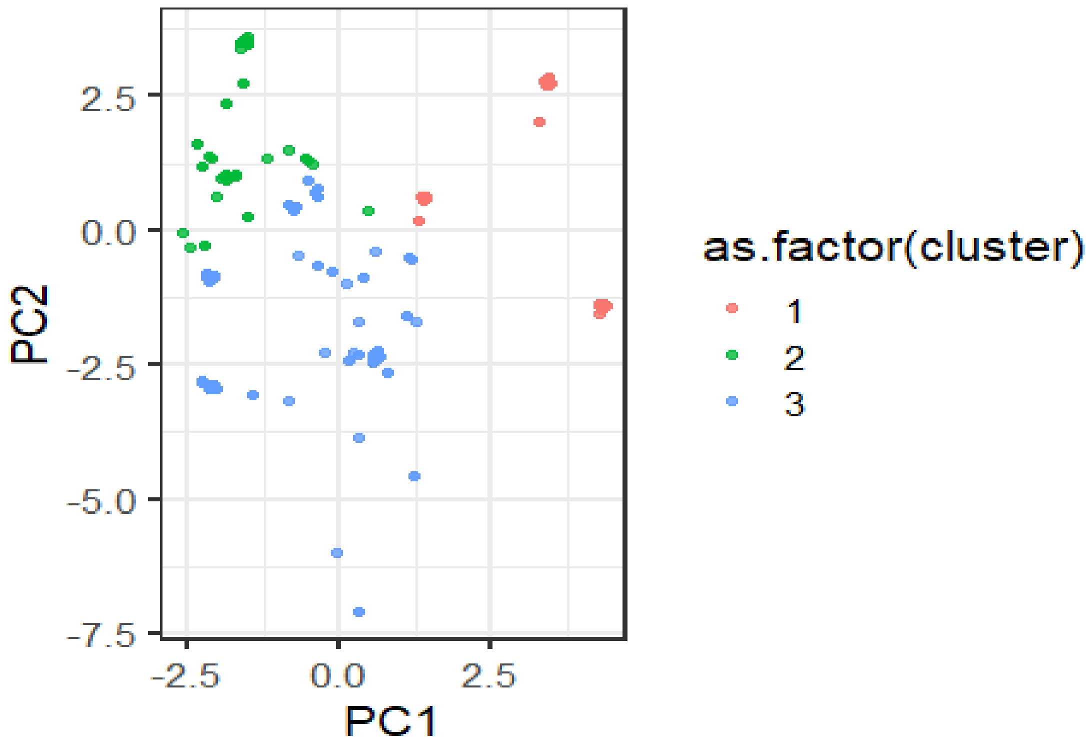

```
df_pca_hier <- df_pca_wf %>%
  add_model(hier_clust(linkage_method = "ward.D")) %>%
  fit(data = df) %>%
  extract_fit_engine() %>%
  fviz_dend(k = 3, main = "Dendograma basado en PCA: Liga Ward")%>%
  plot()

## Registered S3 method overwritten by 'dendextend':
##   method      from
##   rev.hclust vegan

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use
"none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the factoextra package.
##   Please report the issue at
<https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning
was
## generated.
```



Con el dendograma indica que efectivamente debemos quedarnos con **tres grupos**

kMeans con tres grupos de acuerdo a la recomendación del Dendograma

```
pca_df <-
  recipe(~ . , data = df) %>%
  update_role(Sleep.Disorder, new_role = "id") %>%
  step_dummy(all_nominal_predictors()) %>%
  step_normalize(all_predictors()) %>%
  step_pca(all_predictors(), threshold = 0.80) %>%
  prep(df) %>%
  bake(df)

pca_clusters2 <-
  bind_cols(pca_df, cluster=k3$cluster)

g2<-ggplot(pca_clusters2, aes(x = PC1, y = PC2, color =
as.factor(cluster))) +
  geom_point(alpha = 0.8, show.legend = TRUE)

g2
```

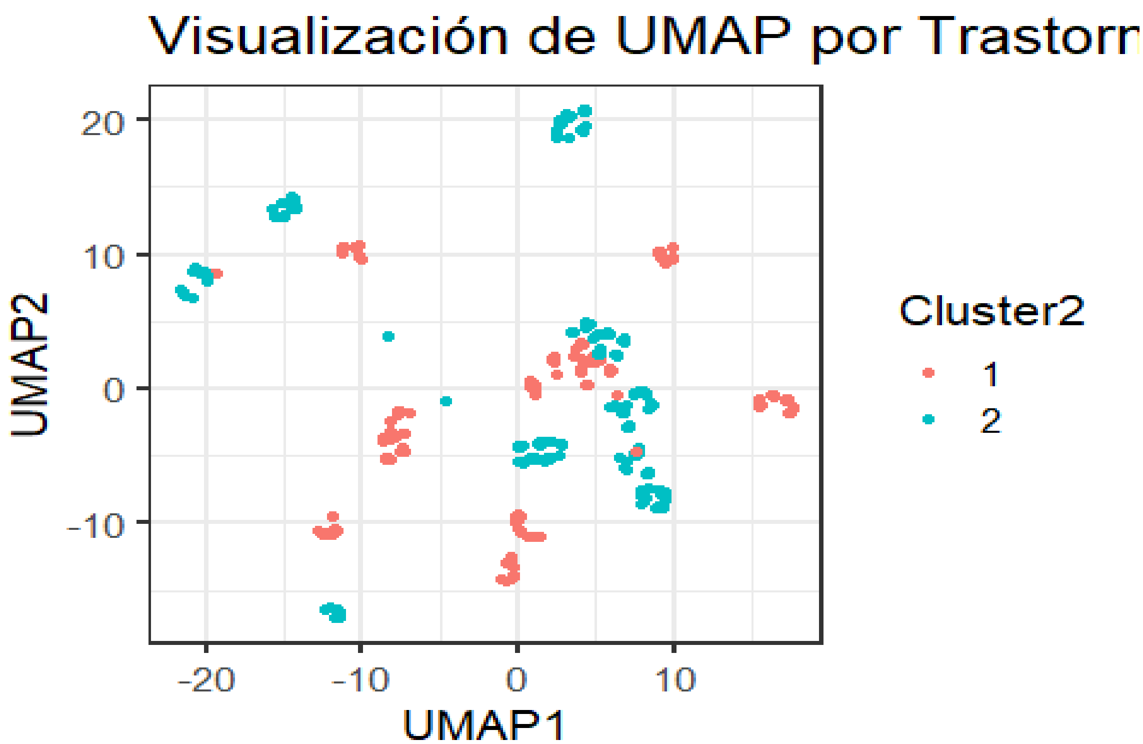


Al parecer con una reducción de dimensiones si existe una clara serpación de la información, solo recordar que este gráfico está construido solo con dos Componenete y estas explican el 38% de la variabilidad de acuerdo al ejercicio PCA entregado previamente

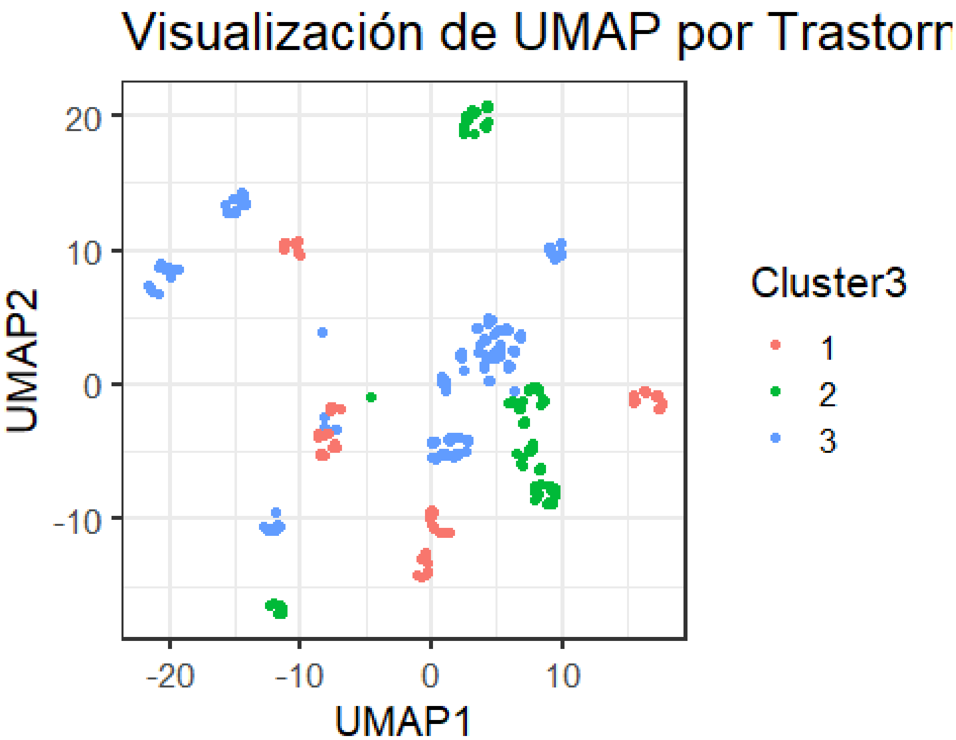
Creemos UMAP para gráficar. Los hiperparámetros que tomaremos son: neighbors=50 y min_dis=0.5 que fueron la conclusión del ejericio anterior

```
# Declaro Recipe

umap_rec <- recipe(~., data = df) %>%
  update_role(Sleep.Disorder, new_role = "id") %>%
  step_dummy(all_nominal_predictors()) %>%  ###Así trabajarán las
categóricas
  step_normalize(all_predictors()) %>%
  step_umap(
    all_predictors(),
    neighbors = 50,       # <-- Número de vecinos
    min_dist = 0.5,         # <-- Distancia mínima
    num_comp = 2            # <-- Número de componentes a generar
(opcional, por defecto es 2)
            )

umap_res <- prep(umap_rec)

umap_res=juice(umap_res)
```

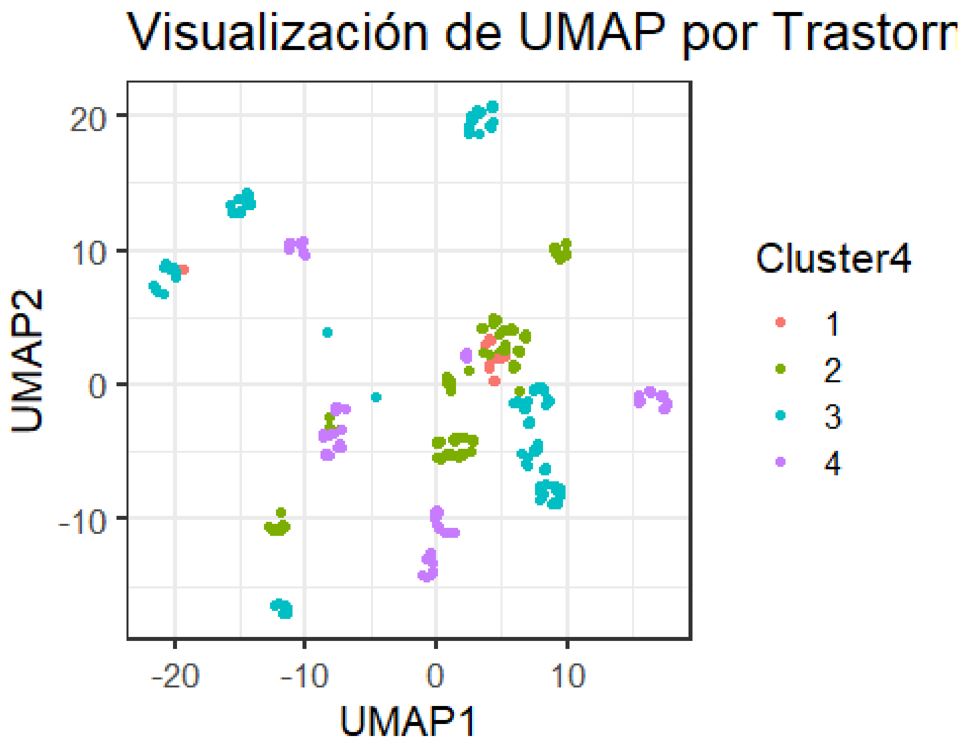
```
umap_res2 = umap_res %>% mutate(Cluster2 =
as.factor(kmeans_clusters2$cluster))

umap_res2%>%
  ggplot(aes(UMAP1, UMAP2)) +
  geom_point(aes(color = Cluster2), size = 1.5)+
  labs(title = "Visualización de UMAP por Trastorno del Sueño")
```




```
umap_res3 = umap_res %>% mutate(Cluster3 =
as.factor(kmeans_clusters3$cluster))

umap_res3%>%
  ggplot(aes(UMAP1, UMAP2)) +
  geom_point(aes(color = Cluster3), size = 1.5)+
  labs(title = "Visualización de UMAP por Trastorno del Sueño")
```




```
umap_res4 = umap_res %>% mutate(Cluster4 =
as.factor(kmeans_clusters4$cluster))

umap_res4%>%
  ggplot(aes(UMAP1, UMAP2)) +
  geom_point(aes(color = Cluster4), size = 1.5)+
  labs(title = "Visualización de UMAP por Trastorno del Sueño")
```



Al parece con UMAP releva porque algunas veces busca varios clusters ya que al ir migrando a una estructura más local hay características de la información que creará grupos con mayor particularidad.

Profundizar con está investigación, podría revelar padecimientos peculiares