R Notebook

ABSTRACT

En este proyecto de AADC he tratado de realizar las siguientes tareas:

1. Preprocesar los datos usando Caret para poder aplicar algoritmos de aprendizaje Supervisado, algoritmos de aprendizaje no supervisado y para visualizar patrones de los datos mediantes matrices de correlación.
2. Caracterizar y representar los datos mediantes diagramas de barras
3. Encontrar patrones en los datos mediante matrices de correlación
4. Realizar apredizaje no supervisado aplicando los algoritmos de kmeans y clustering jerárquico.
5. Clasificar los datos utilizando algoritmos de aprendizaje supervisado.
6. Comparar los rendimientos de la clasificación usando redeucción de dimensialidad mediante PCA, elección de variables mediante Wrapped y clasificación con todas las variables.

- ÍNDICE

1. Introducción a los datos

2. Exploración de los datos

3. Visualización y caracterización de los datos 1

4. Preprocesado de datos con CARET

5. Visualización y caracterización de los datos 2

5.1 Matriz Correlación

5.2 Clustering Jerárquico y Kmeans

6. Clasificación -> Ansiedad

7. Comparativa de modelos predictivos

8. Clasificación con Selección de predictores - Wrapped

9. Clasificación con Reducción de dimensionalidad- PCA

10. Comparativa Clasificación Final

- Introducción a los datos

En la actualidad, la salud mental es una de las mayores preocupaciones sanitarias para la sociedad. En especial para los jóvenes, donde el suicidio se llega a situar entre las principales causas de muerte.

En la última década se ha destapado la realidad sobre las patologías relacionadas con este ámbito y se han empezado a cuestionar los taboos que hasta entonces estas enfermedades llevaban consigo, lo que provocaba que antes se las tratasen como secundarias.

Todo esto ha permitido que se le dee por parte de los ciudadanos la importancia que requieren, y que a día de hoy se considere a estas patologías como a otras cualquiera.

El conjunto de datos utilizado en este proyecto contiene 11 características que describen la salud mental de 101 estudiantes de distintas carreras, cursos y edades.

El objetivo de nuestro proyecto será :

1. Tratar de caracterizar y encontrar respuestas sobre los problemas de salud mental existentes en los estudiantes universitarios. (a través de la aplicación de técnicas de aprendizaje no supervisado).
2. Tratar de predecir, y por tanto evitar de forma temprana, posibles cuadros de ansiedad en estudiantes con alto riesgo de peadecerlos. Lo que permitiría la aplicación de medidas preventivas que permitieran revertir la situación.

Las variables del conjunto de datos son:

1. fechaMedida - fecha en la que se recogió la muestra
2. genero - género del estudiante
3. edad - edad del estudiante (en el momento de la medición)
4. titulacion - carrera universitaria que cursa
5. año - año académico que cursa
6. calificacion - calificación obtenida durante su transcurso en la tirulació
7. estadoCivil - si está en alguna relación sentimental
8. depresion - si padece depresión
9. ansiedad - si padece ansiedad
10. ataquesPanico - si padece frecuentemente ataques de pánico
11. tratamiento - si toma algún tipo de medicamento para alguna patología relaciona con la salud mental

- Exploración de los datos

Carga de librerías

```
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.1     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.2.1
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
#install.packages('plotly', 'repos=http://cran.rstudio.com/ 295', dependencies=TRUE)

library(plotly)
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rsample)
library(recipes)
```

```
##
## Attaching package: 'recipes'
##
## The following object is masked from 'package:stringr':
##
##     fixed
##
## The following object is masked from 'package:stats':
##
##     step
```

```
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
#install.packages("C50")
library(C50)

#install.packages("ranger")

library(ranger)
```

Lectura de datos

```
#Importamos los datos y los estudiamos

datosMental<-read.csv("dataset_MentalHealth.csv")
```

Cambio de nombre de columnas

```
names(datosMental) <- c('fechaMedida', 'genero', 'edad', 'titulacion', 'año', 'calificacion',
'estadoCivil', 'depresion', 'ansiedad', 'ataquesPanico', 'tratamiento')
```

Resumen

```
str(datosMental)
```

```
## 'data.frame':    101 obs. of  11 variables:
##  $ fechaMedida  : chr  "8/7/2020 12:02" "8/7/2020 12:04" "8/7/2020 12:05" "8/7/2020 12:06"
...
##  $ genero       : chr  "Female" "Male" "Male" "Female" ...
##  $ edad         : int  18 21 19 22 23 19 23 18 19 18 ...
##  $ titulacion   : chr  "Engineering" "Islamic education" "BIT" "Laws" ...
##  $ año          : chr  "year 1" "year 2" "Year 1" "year 3" ...
##  $ calificacion : chr  "3.00 - 3.49" "3.00 - 3.49" "3.00 - 3.49" "3.00 - 3.49" ...
##  $ estadoCivil  : chr  "No" "No" "No" "Yes" ...
##  $ depresion    : chr  "Yes" "No" "Yes" "Yes" ...
##  $ ansiedad     : chr  "No" "Yes" "Yes" "No" ...
##  $ ataquesPanico: chr  "Yes" "No" "Yes" "No" ...
##  $ tratamiento  : chr  "No" "No" "No" "No" ...
```

```
head(datosMental)
```

	fechaMedida <chr>	gen… <chr>	e… <int>	titulacion <chr>	año <chr>	calificacion <chr>	estadoCivil <chr>	depr <chr>
1	8/7/2020 12:02	Female	18	Engineering	year 1	3.00 - 3.49	No	Yes
2	8/7/2020 12:04	Male	21	Islamic education	year 2	3.00 - 3.49	No	No
3	8/7/2020 12:05	Male	19	BIT	Year 1	3.00 - 3.49	No	Yes
4	8/7/2020 12:06	Female	22	Laws	year 3	3.00 - 3.49	Yes	Yes
5	8/7/2020 12:13	Male	23	Mathemathics	year 4	3.00 - 3.49	No	No
6	8/7/2020 12:31	Male	19	Engineering	Year 2	3.50 - 4.00	No	No

6 rows | 1-10 of 12 columns

```
dim(datosMental)
```

```
## [1] 101  11
```

Búsqueda y tratamiento de datos faltantes

```
nrow(datosMental)
```

```
## [1] 101
```

```
nrow(na.omit(datosMental))
```

```
## [1] 100
```

```
# Contar el número total de valores faltantes
sum(is.na(datosMental))
```

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

```
#2-> por columna
#Busco la columna que tiene algún valor faltante -> EDAD
indx <- apply(datosMental, 2, function(x) any(is.na(x)))
indx
```

```
##   fechaMedida        genero          edad    titulacion           año
##         FALSE         FALSE          TRUE         FALSE         FALSE
##  calificacion   estadoCivil     depresion      ansiedad ataquesPanico
##         FALSE         FALSE         FALSE         FALSE         FALSE
##   tratamiento
##         FALSE
```

```
#numero de la muestra (fila) que tiene el valor perdido
which(is.na(datosMental$edad))
```

```
## [1] 44
```

Tratamiento MissingValue -> Imputación

Hay un individuo con un valor perdido en la columna edad. Como nuestro dataset solo tiene 100 muestras, creo que es mejor la imputación por la media que eliminar el individuo.

```
datosMental[44,'edad'] <- median(datosMental$edad, na.rm =T)
sum(is.na(datosMental))
```

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

- Visualización y caracterización de los datos 1

Ideas

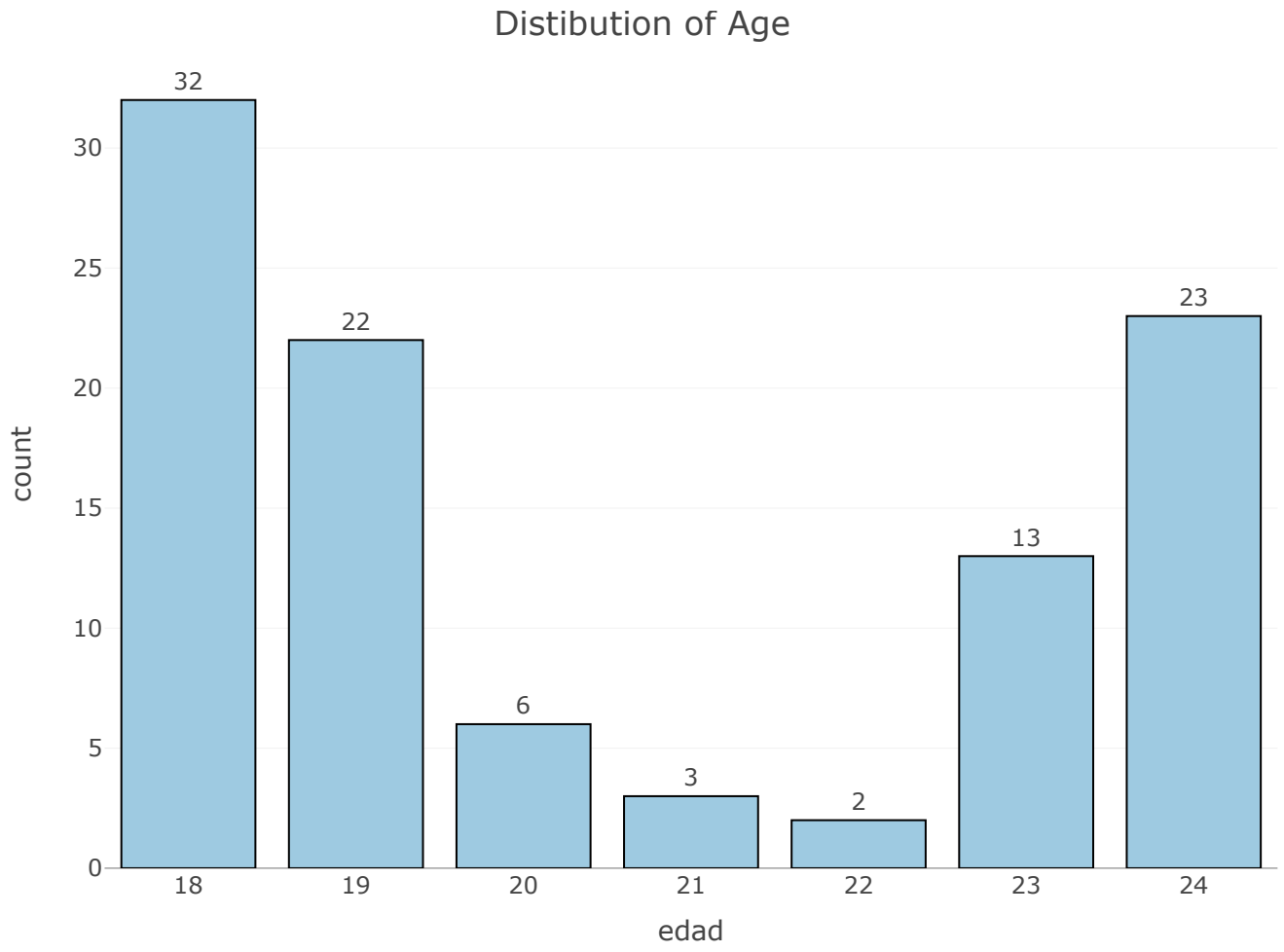
Qué porcentaje de los estudiantes que padecen depresión tienen también ansiedad?

Se tiene más ansiedad en los primeros años de carrera?

Qué porcentaje de estudiantes piden ayuda médica por edad?

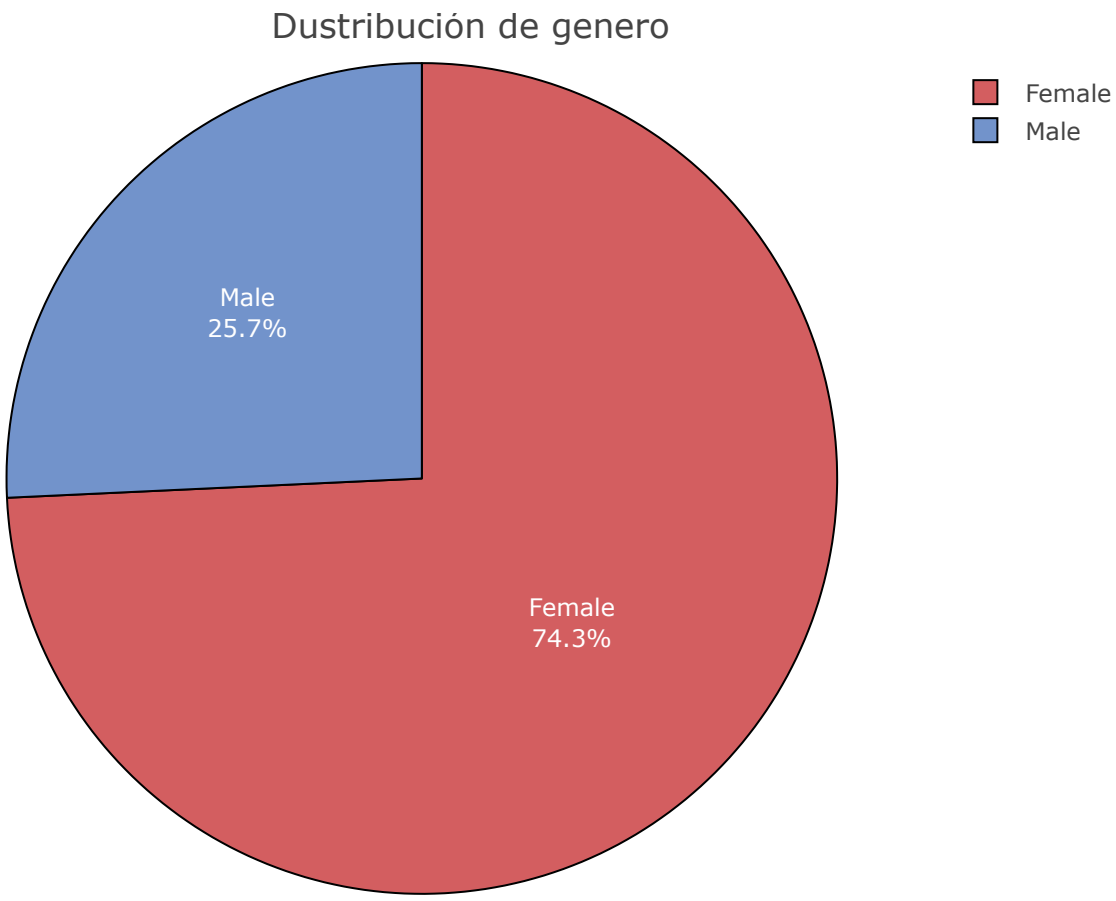
Distribución de edad

```
datosMental %>%
  group_by(edad) %>%
  summarize(count = n()) %>%
  plot_ly(x =~edad, y=~count, type = 'bar',
        text = ~count,
        textposition = 'outside',
        marker = list(color = 'rgb(158,202,225)',
          line = list(color = 'black',
                 width = 1.0))) %>%
layout(title = 'Distibution of Age')
```



Distribución de genero

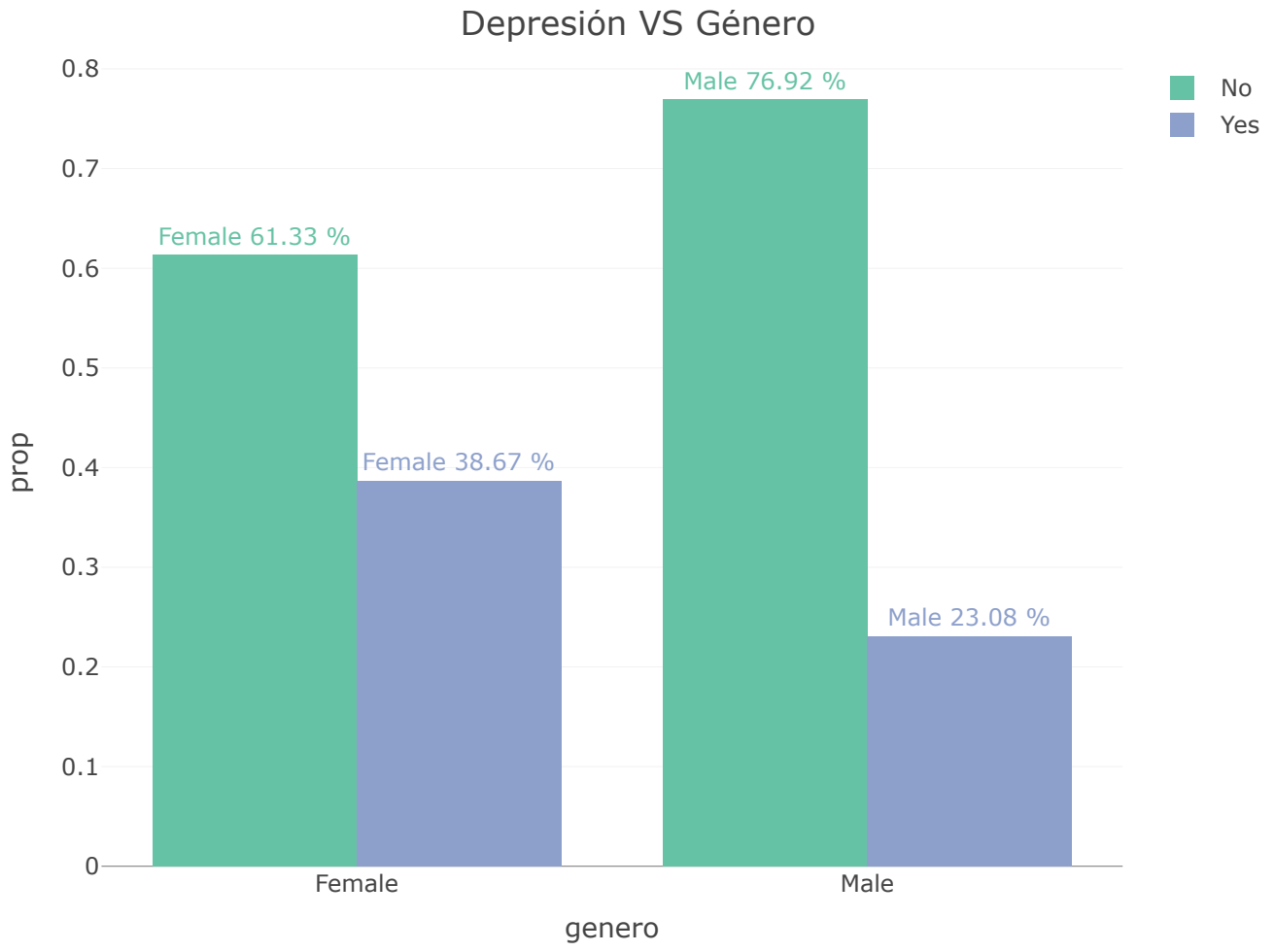
```
dis_gen <- datosMental %>%
  group_by(genero) %>%
  summarise(count = n(),
            percentage = round((n()/ nrow(datosMental)), digits = 4))


colors <- c('rgb(211,94,96)','rgb(114,147,203)')
Gender_PieChart <- plot_ly(data = dis_gen, labels = ~genero, values = ~percentage,
                type = 'pie', sort = F,
                textposition = 'inside',
                textinfo = 'label+percent',
                insidetextfont = list(color = 'White'),
                hoverinfo = 'text',
                text = ~count,
                marker = list(colors = colors,
                line = list(color = 'Black', width = 1)),
                showlegend = TRUE)
Gender_PieChart <- Gender_PieChart %>% layout(title = 'Dustribución de genero')
Gender_PieChart
```



Depresión VS Género

```
datosMental %>%
  count(genero, depresion, sort = F) %>%
  group_by(genero) %>%
  mutate(prop = round((n / sum(n)),digits = 4)) %>%
  plot_ly(x = ~genero, y=~prop, color = ~depresion, type = "bar",
          text = ~paste(genero, prop*100 ,'%'),
          textposition = 'outside') %>%
    layout(barmode = 'Stacked',
           title = 'Depresión VS Género')
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels

## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels
```

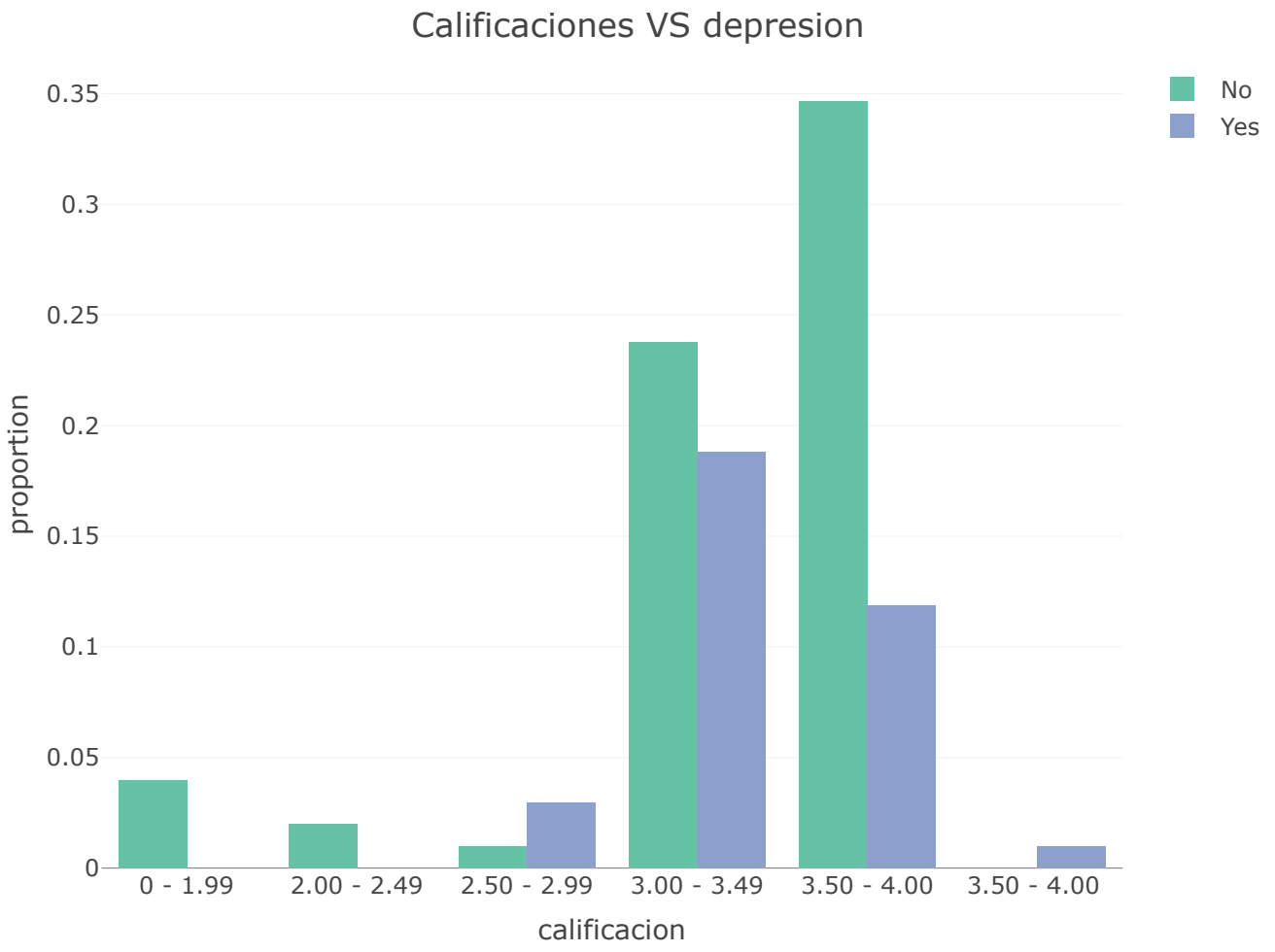


Se observa que las mujeres tienen un índice de depresión bastante más elevado que los hombres.

Calificaciones VS Depresion

```
datosMental %>%
  count(calificacion, depresion, sort = F) %>%
  mutate(proportion = round((n/sum(n)),digits=4)) %>%
  plot_ly(x =~calificacion, y=~proportion, color = ~depresion, type = 'bar') %>%
  layout(barmode = 'Group',
         title = 'Calificaciones VS depresion')
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels

## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels
```



Se puede observar que existe una especie de correlación entre la calificación de los estudiantes y la presencia de depresión. A medida que crecen las calificaciones aumenta la presencia de esta enfermedad.

Titulaciones VS depresión

Titulaciones en las que haya más de dos encuestados

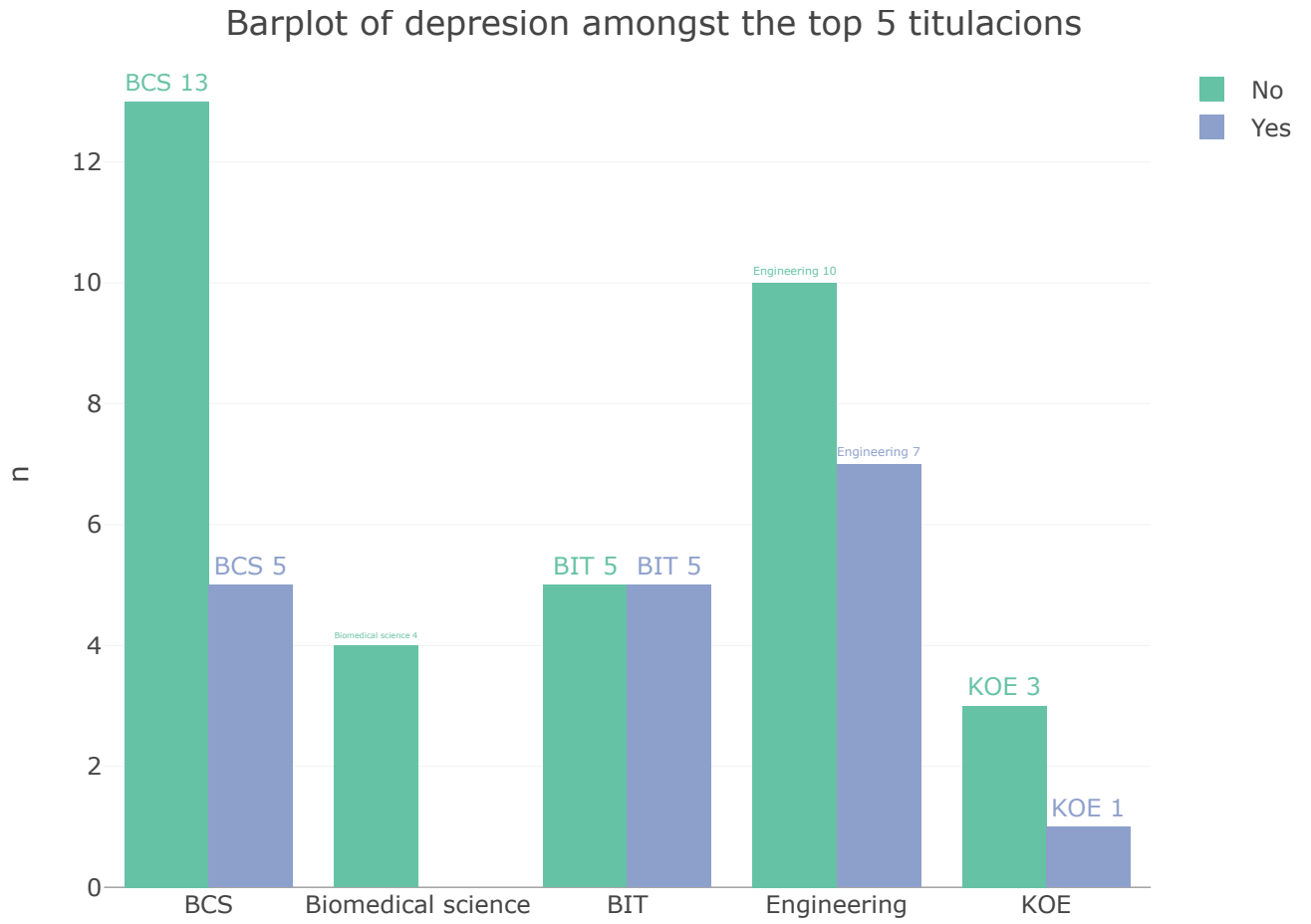
```
datosMental %>%
  group_by(titulacion) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  filter(count >2)
```

titulacion <chr>	count <int>
BCS	18
Engineering	17
BIT	10
Biomedical science	4
KOE	4

5 rows

```
datosMental %>%
  filter(grepl('BIT|KOE|BCS|Engineering|Biomedical science', titulacion)) %>%
  count(titulacion, depresion, sort = T) %>%
  group_by(titulacion) %>%
  mutate(prop = round((n / sum(n)),digits = 4)) %>%
  plot_ly(x = ~titulacion, y=~n, color = ~depresion, type = "bar",
          text = ~paste(titulacion, n),
          textposition = 'outside') %>%
  layout(barmode = 'Stacked',
         title = 'Barplot of depresion amongst the top 5 titulacions')
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels

## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels
```



titulacion

Observamos que la titulación que tiene un mayor porcentaje de estudiantes con depresión es BIT (Bachelor of Information Technology) seguido del resto de ingenierías.

Por lo que podemos deducir que los estudiantes de carreras del ámbito tecnológico tienen mayores problemas de salud mental

- Preprocesado de datos con CARET

La imputación de los datos ya la hemos hecho previamente en la exploración de los datos

Creación del objeto Reciped -> Para predecir la ansiedad

Reciped es un tipo de objeto que se define en Caret para el procesamiento y al que se le aplican diferentes métodos en las diferentes fases del preprocesado.

```
# No incluyo la fecha de medición
objeto_recipe_Ansiedad <- recipe(formula = ansiedad ~ genero + edad + titulacion + tratamient
o + calificacion + estadoCivil + depresion + ataquesPanico,
                         data =  datosMental)
objeto_recipe_Ansiedad
```

```
##
```

```
## ── Recipe ──────────────────────────────────────────────────────────────────────
```

```
##
```

```
## ── Inputs
```

```
## Number of variables by role
```

```
## outcome:   1
## predictor: 8
```

Normalización de los datos entre [0,1] -> No se hacerlo con caret

```
# se lo aplico a todas las var numéricas
# objeto_recipe_Ansiedad <- objeto_recipe_Ansiedad %>% step_scale(all_numeric(), -all_outcome
s())

objeto_recipe_Ansiedad <- objeto_recipe_Ansiedad %>% step_center(all_numeric())
objeto_recipe_Ansiedad <- objeto_recipe_Ansiedad %>% step_scale(all_numeric())
```

Binarización de variable categóricas -> One hot Encoding

```
# se lo aplico a todas las var categóricas
objeto_recipe_Ansiedad <- objeto_recipe_Ansiedad %>% step_dummy(all_nominal(), -all_outcomes
())
```

Entrenamiento del objeto Reciped

```
trained_recipe <- prep(objeto_recipe_Ansiedad, training = datosMental)
trained_recipe
```

```
##
```

```
## ── Recipe ──────────────────────────────────────────────────────────────────────
```

```
##
```

```
## ── Inputs
```

```
## Number of variables by role
```

```
## outcome:   1
## predictor: 8
```

```
##
```

```
## ── Training information
```

```
## Training data contained 101 data points and no incomplete rows.
```

```
##
```

```
## ── Operations
```

```
## • Centering for: edad | Trained
```

```
## • Scaling for: edad | Trained
```

```
## • Dummy variables from: genero, titulacion, tratamiento, ... | Trained
```

Aplicación a nuestro dataset

```
# no está la fecha de medida
datosMental_Limpios <- bake(trained_recipe, new_data = datosMental)

datosMental_Limpios
```

edad <dbl>	ansiedad <fct>	genero_Male <dbl>	titulacion_ALA <dbl>	titulacion_Banking.Studies <dbl>	titula
-1.0106183	No	0	0	0	
0.1949618	Yes	1	0	0	
-0.6087583	Yes	1	0	0	
0.5968218	No	0	0	0	
0.9986818	No	1	0	0	
-0.6087583	No	1	0	0	
0.9986818	No	0	0	0	
-1.0106183	Yes	0	0	0	
-0.6087583	No	0	0	0	
-1.0106183	Yes	1	0	0	

1-10 of 101 rows | 1-6 of 60 columns Previous **1** 2 3 4 5 6 … 11 Next

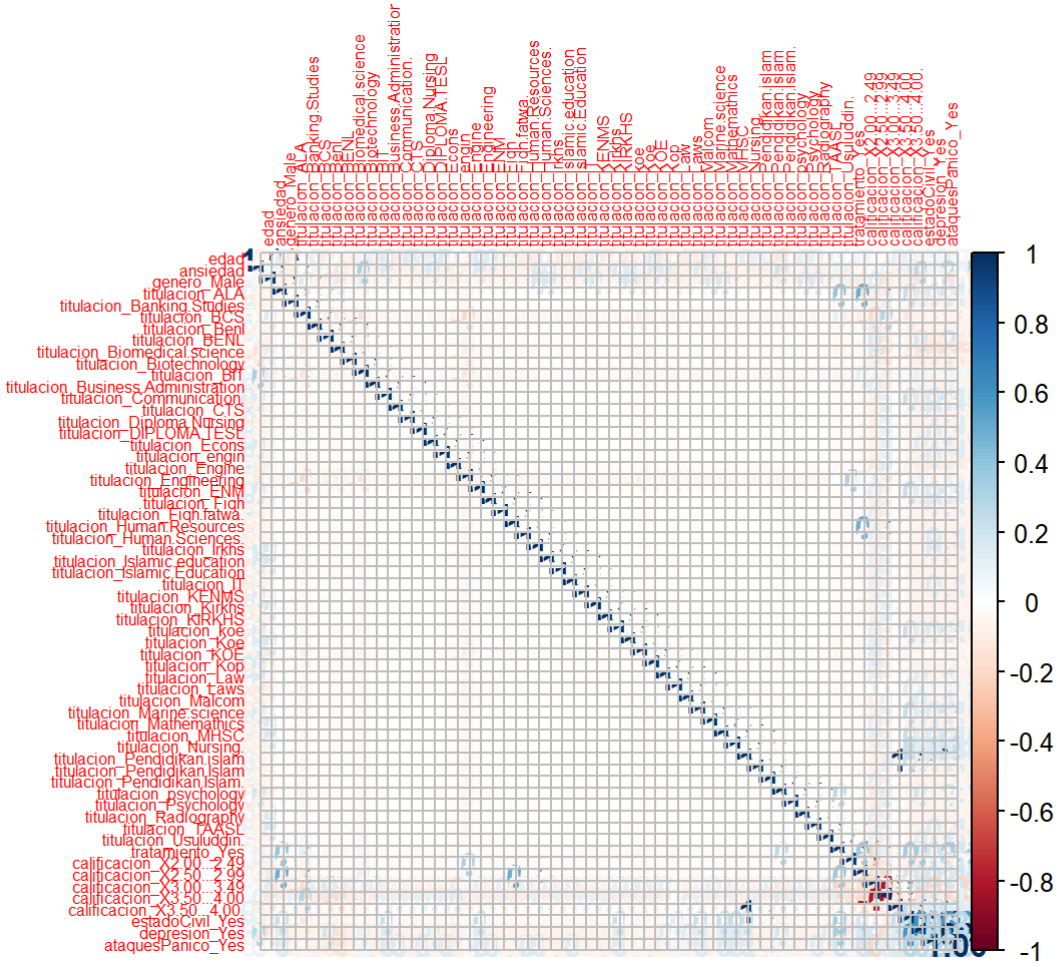
```
objeto_preprocesamiento <- preProcess(datosMental_Limpios, method = c("range"), range = c(0,
1))

# Aplicar el preprocesamiento a los datos
datosMental_Limpios <- predict(objeto_preprocesamiento, datosMental_Limpios)

# Ver los datos escalados
print(datosMental_Limpios)
```

```
## # A tibble: 101 × 60
##     edad ansiedad genero_Male titulacion_ALA titulacion_Banking.Studies
##    <dbl> <fct>          <dbl>          <dbl>                      <dbl>
##  1 0     No                 0              0                          0
##  2 0.5   Yes                1              0                          0
##  3 0.167 Yes                1              0                          0
##  4 0.667 No                 0              0                          0
##  5 0.833 No                 1              0                          0
##  6 0.167 No                 1              0                          0
##  7 0.833 No                 0              0                          0
##  8 0     Yes                0              0                          0
##  9 0.167 No                 0              0                          0
## 10 0     Yes                1              0                          0
## # ℹ 91 more rows
## # ℹ 55 more variables: titulacion_BCS <dbl>, titulacion_Benl <dbl>,
## #   titulacion_BENL <dbl>, titulacion_Biomedical.science <dbl>,
## #   titulacion_Biotechnology <dbl>, titulacion_BIT <dbl>,
## #   titulacion_Business.Administration <dbl>, titulacion_Communication. <dbl>,
## #   titulacion_CTS <dbl>, titulacion_Diploma.Nursing <dbl>,
## #   titulacion_DIPLOMA.TESL <dbl>, titulacion_Econs <dbl>, …
```

- Visualización y caracterización de los datos 2

Matriz de correlación

```
#Creo un nuevo dataset igual que el anterior pero donde voy a pasar la variable ansiedad a nu
mérica para crear la matriz de correlación

datos_matrizCor <- datosMental_Limpios

#paso la variable ansiedad a numérica y la meto en el nuevo dataset

datos_matrizCor$ansiedad <- as.numeric(datosMental_Limpios$ansiedad)

# Genero los coeficientes de correlación

matriz_correlacion <- cor(datos_matrizCor)

#Lo represento en una matriz de correlación con todas las variables
#PROBLEMA -> es demasiada grande (no se ve bien)

corrplot(matriz_correlacion, method = "number", tl.cex = 0.5)
```



No se ve apenas nada ya que hay demasiadas variables. Pero si se puede llegar a apreciar que existe una mayor correlación entre las últimas filas y las columnas de la matriz.

Por lo que voy a hacer otra matriz de correlación pero esta vez solo con un subconjunto de variables que me permitan ampliar y ver de forma más exacta la zona más correlacionada.

```
variables <- names(datos_matrizCor)
print(variables)
```

```
##  [1] "edad"                              "ansiedad"
##  [3] "genero_Male"                       "titulacion_ALA"
##  [5] "titulacion_Banking.Studies"        "titulacion_BCS"
##  [7] "titulacion_Benl"                   "titulacion_BENL"
##  [9] "titulacion_Biomedical.science"     "titulacion_Biotechnology"
## [11] "titulacion_BIT"                    "titulacion_Business.Administration"
## [13] "titulacion_Communication."         "titulacion_CTS"
## [15] "titulacion_Diploma.Nursing"        "titulacion_DIPLOMA.TESL"
## [17] "titulacion_Econs"                  "titulacion_engin"
## [19] "titulacion_Engine"                 "titulacion_Engineering"
## [21] "titulacion_ENM"                    "titulacion_Fiqh"
## [23] "titulacion_Fiqh.fatwa."            "titulacion_Human.Resources"
## [25] "titulacion_Human.Sciences."        "titulacion_Irkhs"
## [27] "titulacion_Islamic.education"      "titulacion_Islamic.Education"
## [29] "titulacion_IT"                     "titulacion_KENMS"
## [31] "titulacion_Kirkhs"                 "titulacion_KIRKHS"
## [33] "titulacion_koe"                    "titulacion_Koe"
## [35] "titulacion_KOE"                    "titulacion_Kop"
## [37] "titulacion_Law"                    "titulacion_Laws"
## [39] "titulacion_Malcom"                 "titulacion_Marine.science"
## [41] "titulacion_Mathemathics"           "titulacion_MHSC"
## [43] "titulacion_Nursing."               "titulacion_Pendidikan.islam"
## [45] "titulacion_Pendidikan.Islam"       "titulacion_Pendidikan.Islam."
## [47] "titulacion_psychology"             "titulacion_Psychology"
## [49] "titulacion_Radiography"            "titulacion_TAASL"
## [51] "titulacion_Usuluddin."             "tratamiento_Yes"
## [53] "calificacion_X2.00...2.49"         "calificacion_X2.50...2.99"
## [55] "calificacion_X3.00...3.49"         "calificacion_X3.50...4.00"
## [57] "calificacion_X3.50...4.00."        "estadoCivil_Yes"
## [59] "depresion_Yes"                     "ataquesPanico_Yes"
```

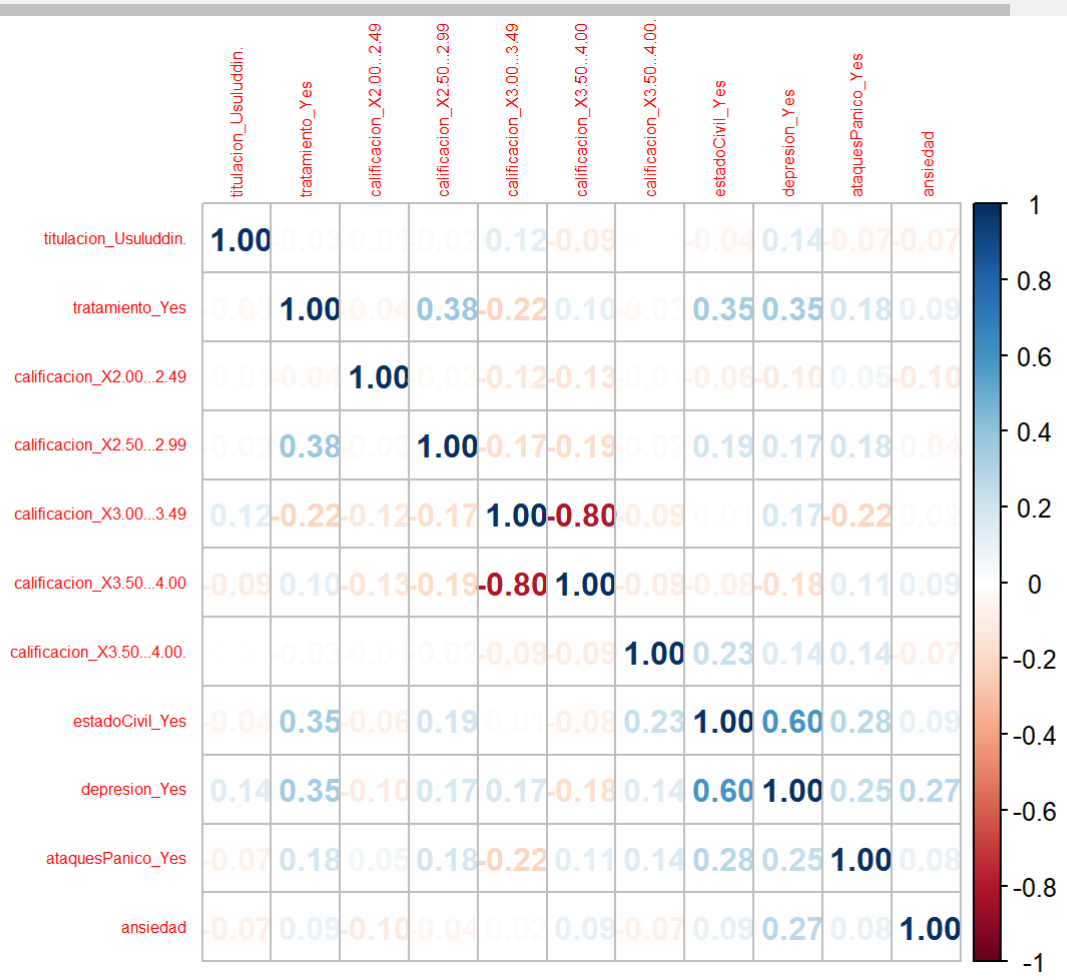
```
# voy a quedarme solo con las últimas 10 variables


# Selecciono las 10 últimas variables
subconjunto_cor_list <- tail(variables, 10)

# Crear el nuevo dataset con las 10 últimas variables
subconjunto_Cor <- datos_matrizCor[, subconjunto_cor_list]

# meto también la ansiedad
subconjunto_Cor$ansiedad <- datos_matrizCor$ansiedad

# Imprimir el nuevo dataset
subconjunto_Cor
```

titulacion_Usuluddin. <dbl>	tratamiento_Yes <dbl>	calificacion_X2.00...2.49 <dbl>	calificacio
0	0	0	
0	0	0	
0	0	0	

titulacion_Usuluddin. <dbl>	tratamiento_Yes <dbl>	calificacion_X2.00...2.49 <dbl>	calificacio
0	0	0	
0	0	0	
0	0	0	
0	0	0	
0	0	0	
0	0	0	
0	0	0	

1-10 of 101 rows | 1-4 of 11 columns Previous 1 2 3 4 5 6 ... 11 Next

```
subconjunto_Cor_Matriz <- cor(subconjunto_Cor)

#Lo represento en una matriz de correlación con todas las variables
#PROBLEMA -> es demasiada grande (no se ve bien)

corrplot(subconjunto_Cor_Matriz, method = "number", tl.cex = 0.5)
```



Se puede apreciar que parece que existe una más que existente correlación entre el estado civil de los estudiantes y la presencia de depresión

Intento de clustering K-means

```
# Aplicamos el algoritmo k-means con tres centros de masa y el parámetro nstar igual a 20
km_puntos <- kmeans(datos_matrizCor, centers=2, nstart=20)
km_puntos
```

```
## K-means clustering with 2 clusters of sizes 47, 54
##
## Cluster means:
##       edad ansiedad genero_Male titulacion_ALA titulacion_Banking.Studies
## 1 0.4255319 1.382979   0.1914894     0.00000000                 0.0212766
## 2 0.4135802 1.296296   0.3148148     0.01851852                 0.0000000
##   titulacion_BCS titulacion_Benl titulacion_BENL titulacion_Biomedical.science
## 1     0.29787234      0.00000000      0.00000000                   0.00000000
## 2     0.07407407      0.01851852      0.03703704                   0.07407407
##   titulacion_Biotechnology titulacion_BIT titulacion_Business.Administration
## 1              0.00000000     0.06382979                          0.00000000
## 2              0.01851852     0.12962963                          0.01851852
##   titulacion_Communication. titulacion_CTS titulacion_Diploma.Nursing
## 1                 0.0212766      0.0212766                 0.0212766
## 2                 0.0000000      0.0000000                 0.0000000
##   titulacion_DIPLOMA.TESL titulacion_Econs titulacion_engin titulacion_Engine
## 1              0.0212766        0.0212766        0.0212766        0.04255319
## 2              0.0000000        0.0000000        0.0000000        0.00000000
##   titulacion_Engineering titulacion_ENM titulacion_Fiqh titulacion_Fiqh.fatwa.
## 1              0.1276596     0.00000000      0.00000000             0.00000000
## 2              0.2037037     0.01851852      0.01851852             0.01851852
##   titulacion_Human.Resources titulacion_Human.Sciences. titulacion_Irkhs
## 1                 0.00000000                 0.00000000        0.0212766
## 2                 0.01851852                 0.01851852        0.0000000
##   titulacion_Islamic.education titulacion_Islamic.Education titulacion_IT
## 1                   0.00000000                   0.0212766    0.00000000
## 2                   0.01851852                   0.0000000    0.01851852
##   titulacion_KENMS titulacion_Kirkhs titulacion_KIRKHS titulacion_koe
## 1        0.0212766         0.0212766         0.0212766     0.00000000
## 2        0.0000000         0.0000000         0.0000000     0.01851852
##   titulacion_Koe titulacion_KOE titulacion_Kop titulacion_Law titulacion_Laws
## 1     0.00000000     0.02127660     0.00000000     0.00000000      0.02127660
## 2     0.01851852     0.05555556     0.01851852     0.01851852      0.01851852
##   titulacion_Malcom titulacion_Marine.science titulacion_Mathemathics
## 1         0.0212766                0.0212766              0.00000000
## 2         0.0000000                0.0000000              0.01851852
##   titulacion_MHSC titulacion_Nursing. titulacion_Pendidikan.islam
## 1      0.00000000           0.0212766                  0.00000000
## 2      0.01851852           0.0000000                  0.01851852
##   titulacion_Pendidikan.Islam titulacion_Pendidikan.Islam.
## 1                   0.0212766                   0.00000000
## 2                   0.0000000                   0.01851852
##   titulacion_psychology titulacion_Psychology titulacion_Radiography
## 1            0.04255319             0.0212766             0.00000000
## 2            0.00000000             0.0000000             0.01851852
##   titulacion_TAASL titulacion_Usuluddin. tratamiento_Yes
## 1        0.0212766            0.00000000      0.08510638
## 2        0.0000000            0.01851852      0.03703704
##   calificacion_X2.00...2.49 calificacion_X2.50...2.99 calificacion_X3.00...3.49
## 1                0.00000000                0.00000000                 0.0000000
## 2                0.03703704                0.07407407                 0.7962963
##   calificacion_X3.50...4.00 calificacion_X3.50...4.00. estadoCivil_Yes
## 1                         1                 0.00000000       0.1276596
## 2                         0                 0.01851852       0.1851852
##   depresion_Yes ataquesPanico_Yes
```

```
## 1     0.2553191         0.3829787
## 2     0.4259259         0.2777778
##
## Clustering vector:
##   [1] 2 2 2 2 2 1 2 1 2 1 1 1 2 2 1 1 2 2 1 1 2 2 1 2 1 1 1 2 1 1 2 2 1 1 2 2 2
##  [38] 2 1 2 2 1 2 2 1 1 1 1 2 1 2 1 2 2 1 2 1 2 1 2 2 1 1 1 1 1 2 2 1 2 2 1 2 2
##  [75] 1 1 2 2 2 1 1 1 2 2 2 1 2 1 2 2 1 2 2 2 2 1 1 2 1 1 2
##
## Within cluster sum of squares by cluster:
## [1]  97.36288 129.43004
##  (between_SS / total_SS =  16.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
km_puntos$cluster
```

```
##   [1] 2 2 2 2 2 1 2 1 2 1 1 1 2 2 1 1 2 2 1 1 2 2 1 2 1 1 1 2 1 1 2 2 1 1 2 2 2
##  [38] 2 1 2 2 1 2 2 1 1 1 1 2 1 2 1 2 2 1 2 1 2 1 2 2 1 1 1 1 1 2 2 1 2 2 1 2 2
##  [75] 1 1 2 2 2 1 1 1 2 2 2 1 2 1 2 2 1 2 2 2 2 1 1 2 1 1 2
```

```
#n strat es la aleatoriedad del algoritmo

# los vectores de abajo será el punto numero 1 cae en el claster 2
```

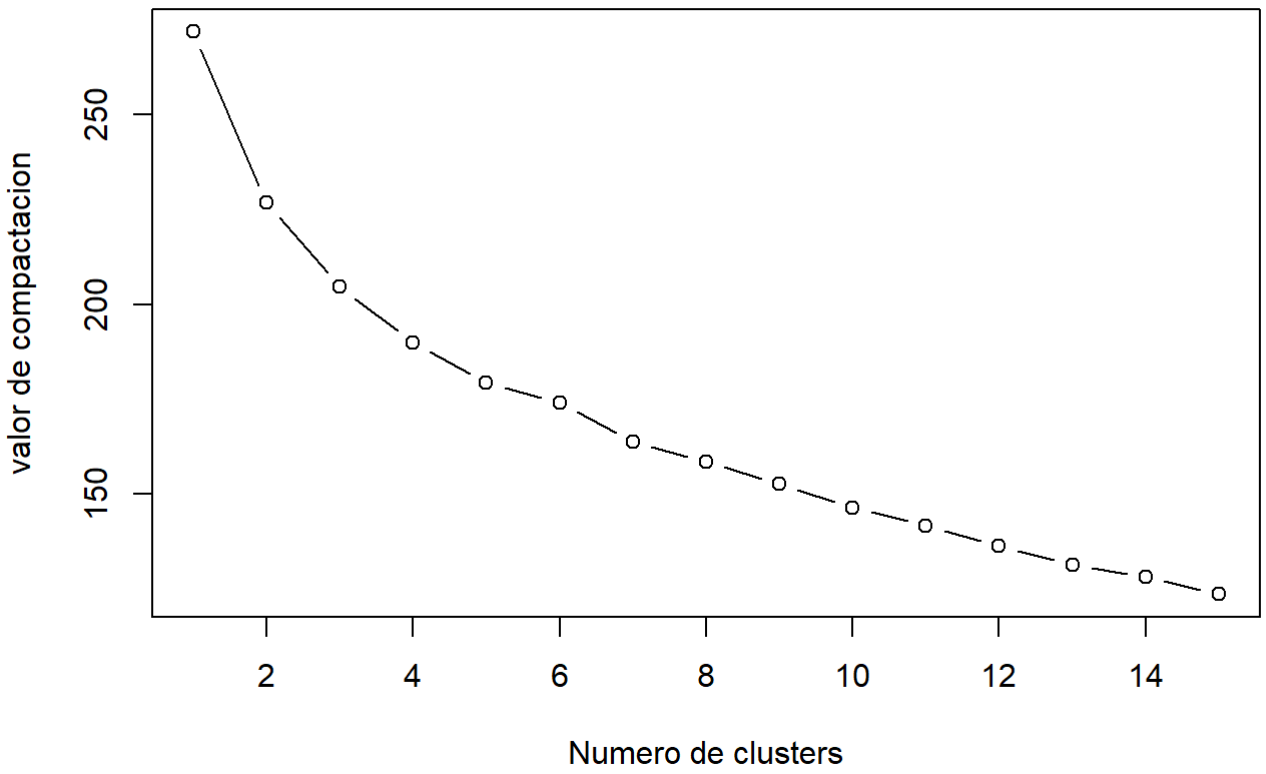
```
#No me deja representarlos graficamente por el tamaño de la representación -> he intentado ar
reglarlo pero me ha sido imposible.


#par(mar = c(1, 1, 1, 1)) # Ajustar los márgenes a valores más pequeños

#plot(datos_matrizCor, col=km_puntos$cluster) #se obtienen tres clusters
```

Dispersión de clusters

Esto nos permitirá observar cual es el número óptimo de clústers para nuestro conjunto de datos.

Parece que a partir de los 3 clúster la dispersión empieza a reducirse en menor medida.

```
# quiero hacer 15 ejecucuiones variando el numero d egrupos para comparar el valor de compact
acion

vector_compactacion <- 0
for (i in 1:15){
  km_puntos_aux2 <- kmeans(datos_matrizCor, centers=i, nstar=20)
  vector_compactacion[i] <- km_puntos_aux2$tot.withinss


}

# hacemos la graficas

par(mfrow= c(1,1))



plot(1:15, vector_compactacion, type= 'b', xlab= 'Numero de clusters',
     ylab= 'valor de compactacion')
```

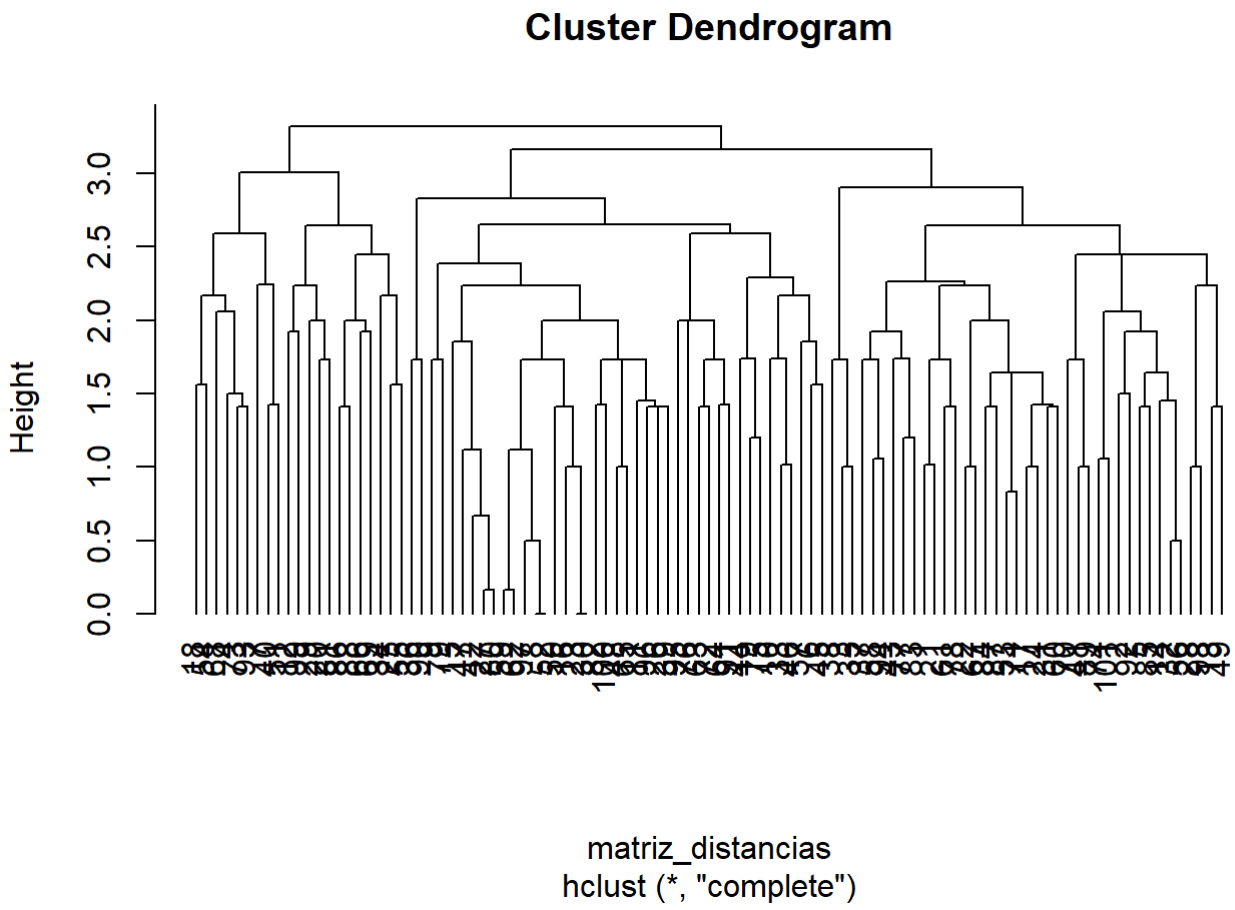


Clustering Jerárquico

```
#matriz de distancias
matriz_distancias <- dist(datos_matrizCor)

#se construye el dendograma
hclust_aux <- hclust(matriz_distancias)

summary(hclust_aux)
```

```
##             Length Class  Mode
## merge       200    -none- numeric
## height      100    -none- numeric
## order       101    -none- numeric
## labels        0    -none- NULL
## method        1    -none- character
## call          2    -none- call
## dist.method   1    -none- character
```

```
plot (hclust_aux, hang = -2)
```




```
# se puede costar según la altura del endograma
cutree (hclust_aux, h = 2.8)
```

```
##   [1] 1 1 2 3 1 4 3 4 4 4 4 5 1 1 4 4 1 3 4 5 1 1 4 1 5 4 4 1 5 4 1 1 4 5 2 6 1
##  [38] 2 4 3 1 4 1 4 4 4 4 4 1 4 3 4 1 3 5 1 4 1 4 1 1 4 4 4 4 5 1 3 5 1 1 4 3 1
##  [75] 4 4 1 1 4 4 5 4 1 1 1 5 1 5 1 6 4 1 3 1 4 4 4 1 5 4 1
```

```
# se puede cortar por el numero de cluster que quiero
cutree (hclust_aux, , k=3)
```

```
##   [1] 1 1 1 2 1 3 2 3 3 3 3 2 1 1 3 3 1 2 3 2 1 1 3 1 2 3 3 1 2 3 1 1 3 2 1 3 1
##  [38] 1 3 2 1 3 1 3 3 3 3 3 1 3 2 3 1 2 2 1 3 1 3 1 1 3 3 3 3 2 1 2 2 1 1 3 2 1
##  [75] 3 3 1 1 3 3 2 3 1 1 1 2 1 2 1 3 3 1 2 1 3 3 3 1 2 3 1
```

Visualizar -> Otra vez los problemas con los margenes

```
#plot(datos_matrizCor, col=cutree (hclust_aux, , k=3), main= 'Se obtienen 3 puntos')
```

- Clasificación -> Ansiedad

Vamos a realizar el entrenamiento y evaluación de modelos predictivos usando el paquete CARET.

Para ello, creo que la técnica de evaluación más confiable es la VALIDACIÓN CRUZADA.

Técnica de evaluación -> Validación Cruzada

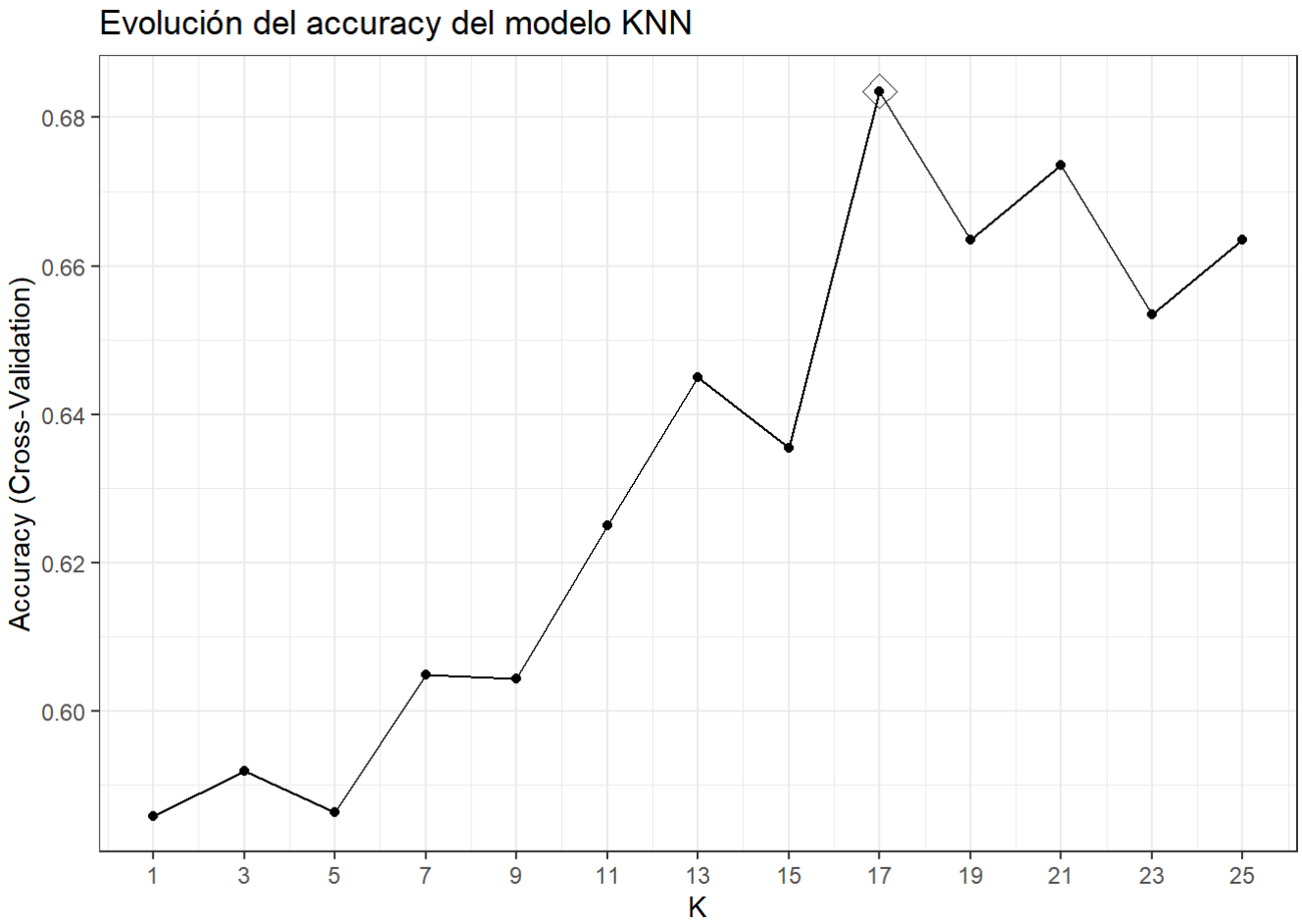
```
particiones  <- 5

set.seed(123)

control_train2 <- trainControl(method = "cv", #Validación cruzada repetida
                               number = particiones, #Dividimos conjunto en 5 partes
                               returnResamp = "final",
                               verboseIter = FALSE)
```

KNN

```
# Hiperparámetros
hiperparametros <- data.frame(k = c(1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25))

# AJUSTE DEL MODELO
# ============================================================================
set.seed(342)
modelo_knn <- train(ansiedad ~ ., data = datosMental_Limpios,
                    method = "knn",
                    tuneGrid = hiperparametros,
                    metric = "Accuracy",
                    trControl = control_train2)
modelo_knn
```

```
## k-Nearest Neighbors
##
## 101 samples
##  59 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 81, 81, 80, 82, 80
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.5858396   0.116765983
##    3  0.5918797   0.028996880
##    5  0.5863158  -0.051467417
##    7  0.6048371  -0.048533670
##    9  0.6043609  -0.050365497
##   11  0.6249373  -0.010119898
##   13  0.6449875   0.007797124
##   15  0.6354637  -0.010384694
##   17  0.6835589   0.096730147
##   19  0.6635088   0.017391304
##   21  0.6735589   0.060160761
##   23  0.6535088  -0.001786778
##   25  0.6635088   0.000000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 17.
```

```
ggplot(modelo_knn, highlight = TRUE) +
  scale_x_continuous(breaks = hiperparametros$k) +
  labs(title = "Evolución del accuracy del modelo KNN", x = "K") +
  theme_bw()
```



Naive Bayes

```
#
#  # AJUSTE DEL MODELO
#  # ============================================================================
  set.seed(342)
  modelo_nb <- train(ansiedad ~ ., data = datosMental_Limpios,
                     method = "nb",
                    metric = "Accuracy",
                    trControl = control_train2)
```

```
## Warning: model fit failed for Fold1: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.d
efault(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: titulacion_ALA, titulacion_Banking.S
tudies, titulacion_Benl, titulacion_Biomedical.science, titulacion_Biotechnology, titulacion_
Business.Administration, titulacion_Communication., titulacion_CTS, titulacion_Diploma.Nursin
g, titulacion_DIPLOMA.TESL, titulacion_Econs, titulacion_engin, titulacion_Engine, titulacion
_ENM, titulacion_Fiqh, titulacion_Fiqh.fatwa., titulacion_Human.Resources, titulacion_Human.S
ciences., titulacion_Irkhs, titulacion_Islamic.education, titulacion_Islamic.Education, titul
acion_IT, titulacion_KENMS, titulacion_Kirkhs, titulacion_KIRKHS, titulacion_koe, titulacion_
Koe, titulacion_KOE, titulacion_Kop, titulacion_Law, titulacion_Laws, titulacion_Malcom, titu
lacion_Marine.science, titulacion_Mathemathics, titulacion_MHSC, titulacion_Nursing., titulac
ion_Pendidikan.islam, titulacion_Pendidikan.Islam, titulacion_Pendidikan.Islam., titulacion_p
sychology, titulacion_Psychology, titulacion_Radiography, titulacion_TAASL, titulacion_Usulud
din., calificacion_X2.00...2.49, calificacion_X3.50...4.00.
```

```
## Warning: model fit failed for Fold2: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.d
efault(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: titulacion_ALA, titulacion_Banking.S
tudies, titulacion_Benl, titulacion_BENL, titulacion_Biomedical.science, titulacion_Biotechno
logy, titulacion_Business.Administration, titulacion_Communication., titulacion_CTS, titulaci
on_Diploma.Nursing, titulacion_DIPLOMA.TESL, titulacion_Econs, titulacion_engin, titulacion_E
ngine, titulacion_ENM, titulacion_Fiqh, titulacion_Fiqh.fatwa., titulacion_Human.Resources, t
itulacion_Human.Sciences., titulacion_Irkhs, titulacion_Islamic.education, titulacion_Islami
c.Education, titulacion_IT, titulacion_KENMS, titulacion_Kirkhs, titulacion_KIRKHS, titulacio
n_koe, titulacion_Koe, titulacion_Kop, titulacion_Law, titulacion_Laws, titulacion_Malcom, ti
tulacion_Marine.science, titulacion_Mathemathics, titulacion_MHSC, titulacion_Nursing., titul
acion_Pendidikan.islam, titulacion_Pendidikan.Islam, titulacion_Pendidikan.Islam., titulacion
_psychology, titulacion_Psychology, titulacion_Radiography, titulacion_TAASL, titulacion_Usul
uddin., calificacion_X2.00...2.49, calificacion_X3.50...4.00.
```

```
## Warning: model fit failed for Fold3: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.d
efault(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: titulacion_ALA, titulacion_Banking.S
tudies, titulacion_Benl, titulacion_Biomedical.science, titulacion_Biotechnology, titulacion_
Business.Administration, titulacion_Communication., titulacion_CTS, titulacion_Diploma.Nursin
g, titulacion_DIPLOMA.TESL, titulacion_Econs, titulacion_engin, titulacion_Engine, titulacion
_ENM, titulacion_Fiqh, titulacion_Fiqh.fatwa., titulacion_Human.Resources, titulacion_Human.S
ciences., titulacion_Irkhs, titulacion_Islamic.education, titulacion_Islamic.Education, titul
acion_IT, titulacion_KENMS, titulacion_Kirkhs, titulacion_KIRKHS, titulacion_koe, titulacion_
Koe, titulacion_Kop, titulacion_Law, titulacion_Laws, titulacion_Malcom, titulacion_Marine.sc
ience, titulacion_Mathemathics, titulacion_MHSC, titulacion_Nursing., titulacion_Pendidikan.i
slam, titulacion_Pendidikan.Islam, titulacion_Pendidikan.Islam., titulacion_psychology, titul
acion_Psychology, titulacion_Radiography, titulacion_TAASL, titulacion_Usuluddin., calificaci
on_X2.00...2.49, calificacion_X3.50...4.00.
```

```
## Warning: model fit failed for Fold4: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.d
efault(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: titulacion_ALA, titulacion_Banking.S
tudies, titulacion_Benl, titulacion_BENL, titulacion_Biomedical.science, titulacion_Biotechno
logy, titulacion_Business.Administration, titulacion_Communication., titulacion_CTS, titulaci
on_Diploma.Nursing, titulacion_DIPLOMA.TESL, titulacion_Econs, titulacion_engin, titulacion_E
ngine, titulacion_ENM, titulacion_Fiqh, titulacion_Fiqh.fatwa., titulacion_Human.Resources, t
itulacion_Human.Sciences., titulacion_Irkhs, titulacion_Islamic.education, titulacion_Islami
c.Education, titulacion_IT, titulacion_KENMS, titulacion_Kirkhs, titulacion_KIRKHS, titulacio
n_koe, titulacion_Koe, titulacion_Kop, titulacion_Law, titulacion_Laws, titulacion_Malcom, ti
tulacion_Marine.science, titulacion_Mathemathics, titulacion_MHSC, titulacion_Nursing., titul
acion_Pendidikan.islam, titulacion_Pendidikan.Islam, titulacion_Pendidikan.Islam., titulacion
_psychology, titulacion_Psychology, titulacion_Radiography, titulacion_TAASL, titulacion_Usul
uddin., calificacion_X2.00...2.49, calificacion_X2.50...2.99, calificacion_X3.50...4.00.
```

```
## Warning: model fit failed for Fold5: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.d
efault(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: titulacion_ALA, titulacion_Banking.S
tudies, titulacion_Benl, titulacion_Biomedical.science, titulacion_Biotechnology, titulacion_
Business.Administration, titulacion_Communication., titulacion_CTS, titulacion_Diploma.Nursin
g, titulacion_DIPLOMA.TESL, titulacion_Econs, titulacion_engin, titulacion_Engine, titulacion
_ENM, titulacion_Fiqh, titulacion_Fiqh.fatwa., titulacion_Human.Resources, titulacion_Human.S
ciences., titulacion_Irkhs, titulacion_Islamic.education, titulacion_Islamic.Education, titul
acion_IT, titulacion_KENMS, titulacion_Kirkhs, titulacion_KIRKHS, titulacion_koe, titulacion_
Koe, titulacion_Kop, titulacion_Law, titulacion_Laws, titulacion_Malcom, titulacion_Marine.sc
ience, titulacion_Mathemathics, titulacion_MHSC, titulacion_Nursing., titulacion_Pendidikan.i
slam, titulacion_Pendidikan.Islam, titulacion_Pendidikan.Islam., titulacion_psychology, titul
acion_Psychology, titulacion_Radiography, titulacion_TAASL, titulacion_Usuluddin., calificaci
on_X2.00...2.49, calificacion_X3.50...4.00.
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
  modelo_nb
```

```
## Naive Bayes
##
## 101 samples
##  59 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 81, 81, 80, 82, 80
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE            NaN  NaN
##    TRUE      0.6635088    0
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```

Definición Técnica de evaluación y Árbol de

decisión RandomForest

```
# Hiperparámetros
# min.node.size -> minimo de nodos por hoja
hiperparametros <- expand.grid(mtry = c(3, 4, 5, 7),
                               min.node.size = c(2, 3, 4, 5, 10, 15, 20, 30, 35, 40),
                               splitrule = "gini")

# AJUSTE DEL MODELO
# ========================================================================
set.seed(342)
modelo_rf <- train(ansiedad ~ ., data = datosMental_Limpios,
                   method = "ranger",
                   tuneGrid = hiperparametros,
                   metric = "Accuracy",
                   trControl = control_train2)
modelo_rf
```

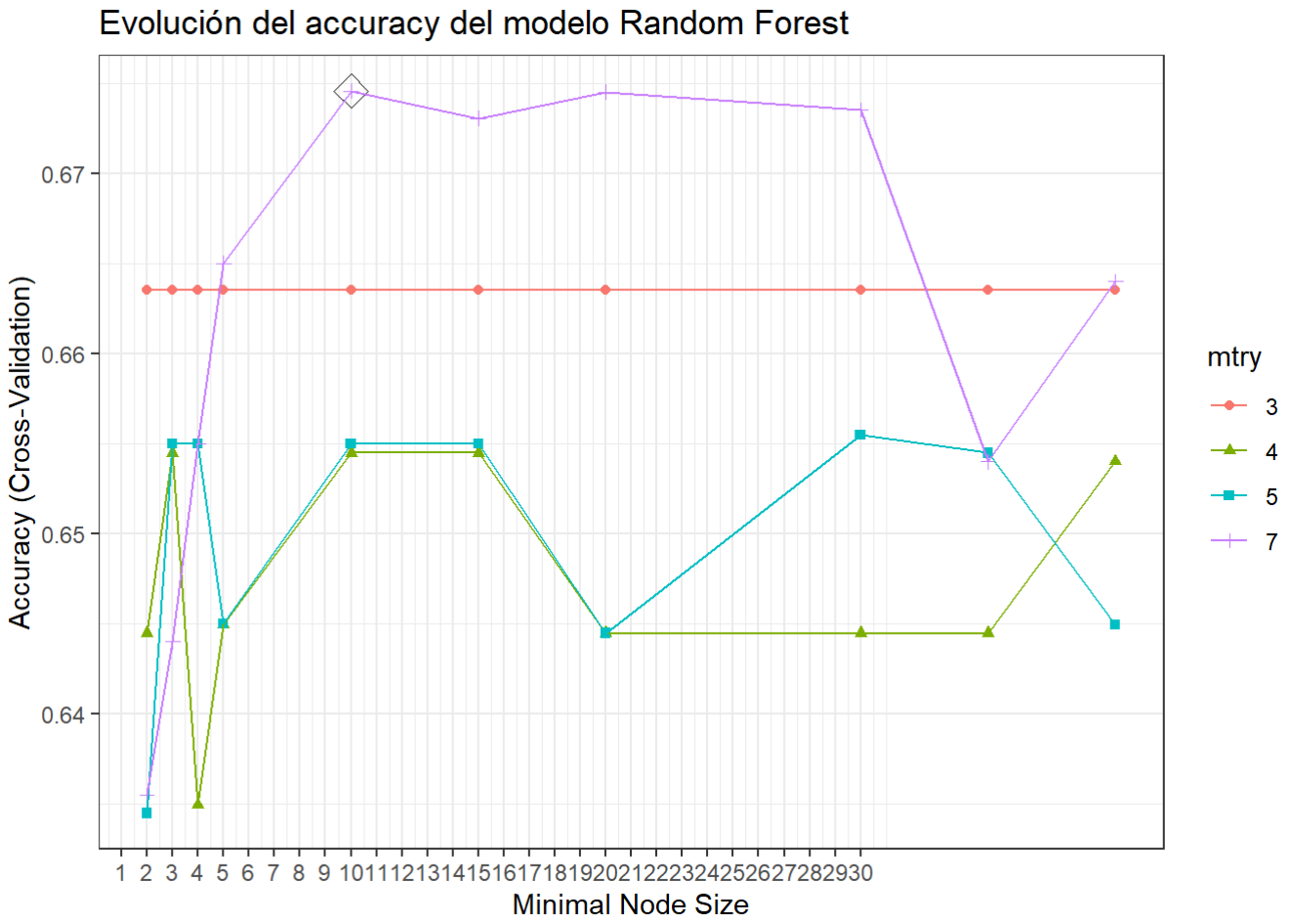
```
## Random Forest
##
## 101 samples
##  59 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 81, 81, 80, 82, 80
## Resampling results across tuning parameters:
##
##   mtry  min.node.size  Accuracy   Kappa
##   3      2             0.6635088   0.0000000000
##   3      3             0.6635088   0.0000000000
##   3      4             0.6635088   0.0000000000
##   3      5             0.6635088   0.0000000000
##   3     10             0.6635088   0.0000000000
##   3     15             0.6635088   0.0000000000
##   3     20             0.6635088   0.0000000000
##   3     30             0.6635088   0.0000000000
##   3     35             0.6635088   0.0000000000
##   3     40             0.6635088   0.0000000000
##   4      2             0.6444612  -0.0347826087
##   4      3             0.6544612   0.0008338297
##   4      4             0.6349373  -0.0500000000
##   4      5             0.6449373  -0.0143835616
##   4     10             0.6544612   0.0008338297
##   4     15             0.6544612   0.0008338297
##   4     20             0.6444612  -0.0347826087
##   4     30             0.6444612  -0.0347826087
##   4     35             0.6444612  -0.0347826087
##   4     40             0.6539850  -0.0181818182
##   5      2             0.6344612  -0.0161680092
##   5      3             0.6549875   0.0451283167
##   5      4             0.6549875   0.0451283167
##   5      5             0.6449875   0.0268071974
##   5     10             0.6549875   0.0427646803
##   5     15             0.6549875   0.0451283167
##   5     20             0.6444612  -0.0002105263
##   5     30             0.6554637   0.0285916450
##   5     35             0.6544612   0.0008338297
##   5     40             0.6449373  -0.0143835616
##   7      2             0.6354637   0.0644362793
##   7      3             0.6439850   0.0711328281
##   7      4             0.6549875   0.0979384576
##   7      5             0.6649875   0.1327430934
##   7     10             0.6745614   0.1919298603
##   7     15             0.6730326   0.1226143065
##   7     20             0.6745113   0.1312393340
##   7     30             0.6735088   0.0873646901
##   7     35             0.6539850   0.0311740891
##   7     40             0.6639850   0.0510010537
##
## Tuning parameter 'splitrule' was held constant at a value of gini
## Accuracy was used to select the optimal model using the largest value.
```

```
## The final values used for the model were mtry = 7, splitrule = gini
##  and min.node.size = 10.
```

```
modelo_rf$finalModel
```

```
## Ranger result
##
## Call:
##  ranger::ranger(dependent.variable.name = ".outcome", data = x,      mtry = min(param$mtr
y, ncol(x)), min.node.size = param$min.node.size,      splitrule = as.character(param$splitru
le), write.forest = TRUE,      probability = classProbs, ...)
##
## Type:                             Classification
## Number of trees:                  500
## Sample size:                      101
## Number of independent variables:  59
## Mtry:                             7
## Target node size:                 10
## Variable importance mode:         none
## Splitrule:                        gini
## OOB prediction error:             28.71 %
```

```
# REPRESENTACIÓN GRÁFICA
# ==============================================================================
ggplot(modelo_rf, highlight = TRUE) +
  scale_x_continuous(breaks = 1:30) +
  labs(title = "Evolución del accuracy del modelo Random Forest") +
  guides(color = guide_legend(title = "mtry"),
         shape = guide_legend(title = "mtry")) +
  theme_bw()
```



Comparativa de modelos predictivos con datasets original (preprocesado)

```
modelos <- list(KNN = modelo_knn,
                rf = modelo_rf,
                nb= modelo_nb)

resultados_resamples <- resamples(modelos)
```

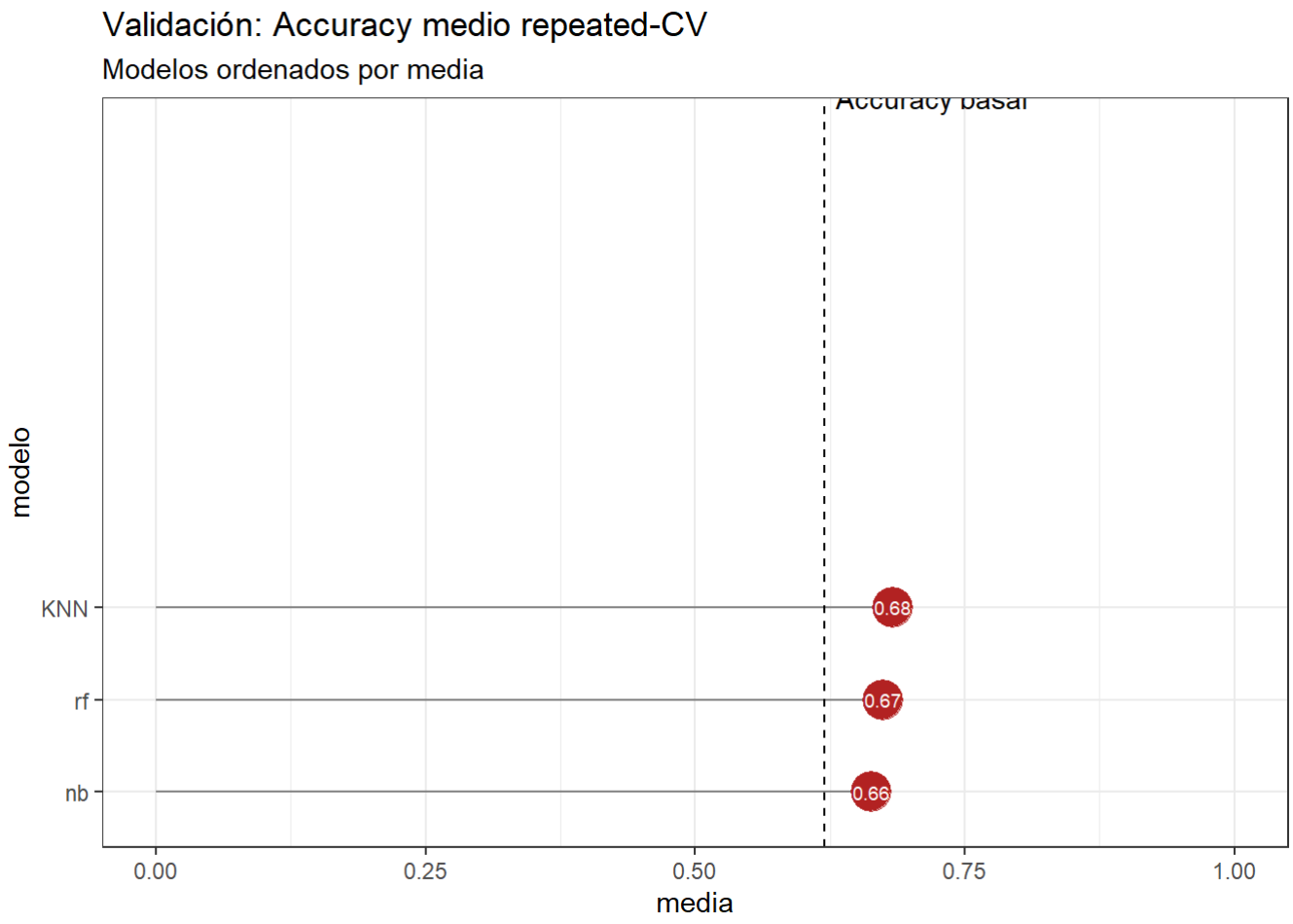
```
metricas_resamples <- resultados_resamples$values %>%
                        gather(key = "modelo", value = "valor", -Resample) %>%
                        separate(col = "modelo", into = c("modelo", "metrica"),
                                 sep = "~", remove = TRUE)
metricas_resamples %>% head()
```

	Resample <chr>	modelo <chr>	metrica <chr>	valor <dbl>
1	Fold1	KNN	Accuracy	0.6500000
2	Fold2	KNN	Accuracy	0.6500000
3	Fold3	KNN	Accuracy	0.7142857
4	Fold4	KNN	Accuracy	0.7368421
5	Fold5	KNN	Accuracy	0.6666667

	Resample	modelo	metrica	valor
	<chr>	<chr>	<chr>	<dbl>
6	Fold1	KNN	Kappa	0.0000000

6 rows

```
metricas_resamples %>%
  filter(metrica == "Accuracy") %>%
  group_by(modelo) %>%
  summarise(media = mean(valor)) %>%
  ggplot(aes(x = reorder(modelo, media), y = media, label = round(media, 2))) +
    geom_segment(aes(x = reorder(modelo, media), y = 0,
                     xend = modelo, yend = media),
                     color = "grey50") +
    geom_point(size = 7, color = "firebrick") +
    geom_text(color = "white", size = 2.5) +
    scale_y_continuous(limits = c(0, 1)) +
    # Accuracy basal
    geom_hline(yintercept = 0.62, linetype = "dashed") +
    annotate(geom = "text", y = 0.72, x = 8.5, label = "Accuracy basal") +
    labs(title = "Validación: Accuracy medio repeated-CV",
         subtitle = "Modelos ordenados por media",
         x = "modelo") +
    coord_flip() +
    theme_bw()
```



Selección de predictores -> Wrapped

Probamos la opción del uso de la técnica de wrapped para ver si se puede mejorar el rendimiento calculado por los algoritmos anteriormente, cambiando la combinación de variables predictoras y buscando la mejor combinación posible.

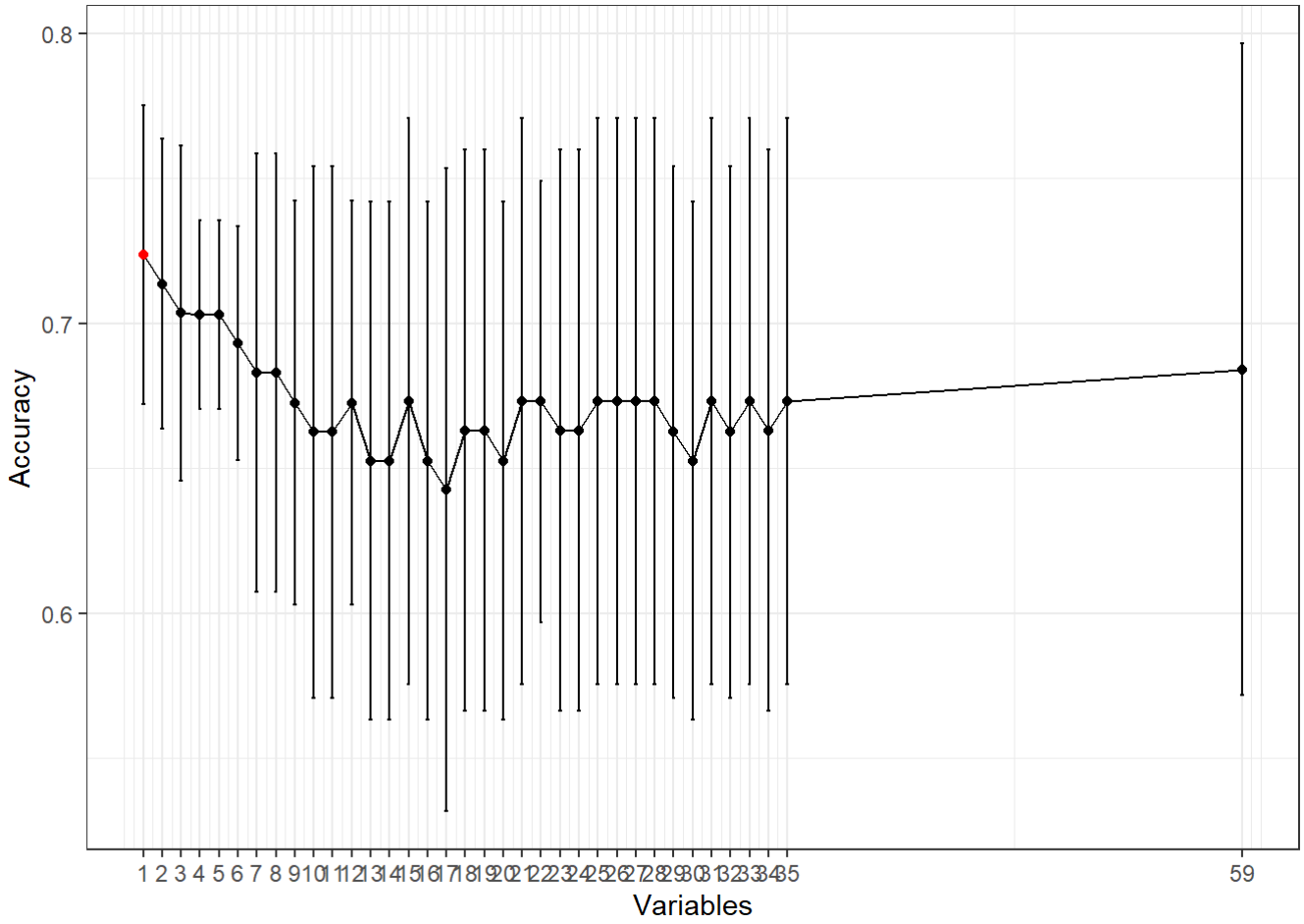
Wrapped usando Random Forest y validación cruzada

```
#Definimos el algoritmo que utilizamos en la técnica wrapped (Random Forest) con el # parámet
ro functions y la técnica
ctrl_rfe <- rfeControl(functions = rfFuncs, method = "cv", number = 5,
                       returnResamp = "all", verbose = FALSE)

# Se ejecuta la eliminación recursiva de predictores
set.seed(342)
rf_rfe <- rfe(ansiedad ~ ., data = datosMental_Limpios,
              sizes = c(1:35), #tamaño de los conjuntos de predictores analizados
              metric = "Accuracy",
              # El accuracy es la proporción de clasificaciones correctas
              rfeControl = ctrl_rfe,
              ntree = 500)
# Dentro de rfe() se pueden especificar argumentos para el modelo empleado, por
# ejemplo, el hiperparámetro ntree=500.
```

```
rf_rfe
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (5 fold)
##
## Resampling performance over subset size:
##
##  Variables Accuracy   Kappa AccuracySD KappaSD Selected
##          1   0.7236 0.24782    0.05154 0.14371        *
##          2   0.7136 0.22859    0.04997 0.12513
##          3   0.7036 0.18556    0.05777 0.13366
##          4   0.7031 0.20660    0.03247 0.09114
##          5   0.7031 0.20660    0.03247 0.09114
##          6   0.6931 0.17704    0.04039 0.07879
##          7   0.6831 0.18626    0.07556 0.12986
##          8   0.6831 0.17554    0.07556 0.12061
##          9   0.6726 0.17575    0.06963 0.12954
##         10   0.6626 0.15121    0.09172 0.14745
##         11   0.6626 0.13569    0.09172 0.18068
##         12   0.6726 0.16644    0.06963 0.11565
##         13   0.6526 0.10614    0.08931 0.15908
##         14   0.6526 0.10614    0.08931 0.15908
##         15   0.6731 0.14479    0.09765 0.18579
##         16   0.6526 0.12165    0.08931 0.12470
##         17   0.6426 0.09143    0.11086 0.19178
##         18   0.6631 0.12596    0.09676 0.17828
##         19   0.6631 0.12596    0.09676 0.17828
##         20   0.6526 0.10614    0.08931 0.15908
##         21   0.6731 0.15552    0.09765 0.19405
##         22   0.6731 0.14174    0.07606 0.14525
##         23   0.6631 0.12596    0.09676 0.17828
##         24   0.6631 0.12596    0.09676 0.17828
##         25   0.6731 0.15552    0.09765 0.19405
##         26   0.6731 0.15552    0.09765 0.19405
##         27   0.6731 0.15552    0.09765 0.19405
##         28   0.6731 0.15552    0.09765 0.19405
##         29   0.6626 0.13569    0.09172 0.18068
##         30   0.6526 0.10614    0.08931 0.15908
##         31   0.6731 0.15552    0.09765 0.19405
##         32   0.6626 0.13569    0.09172 0.18068
##         33   0.6731 0.15552    0.09765 0.19405
##         34   0.6631 0.12596    0.09676 0.17828
##         35   0.6731 0.14479    0.09765 0.18579
##         59   0.6841 0.18607    0.11242 0.25007
##
## The top 1 variables (out of 1):
##    titulacion_BIT
```

```
ggplot(data = rf_rfe$results, aes(x = Variables, y = Accuracy)) +
  geom_line() +
  scale_x_continuous(breaks  = unique(rf_rfe$results$Variables)) +
  geom_point() +
  geom_errorbar(aes(ymin = Accuracy - AccuracySD, ymax = Accuracy + AccuracySD),
                width = 0.2) +
  geom_point(data = rf_rfe$results %>% slice(which.max(Accuracy)),
             color = "red") +
  theme_bw()
```

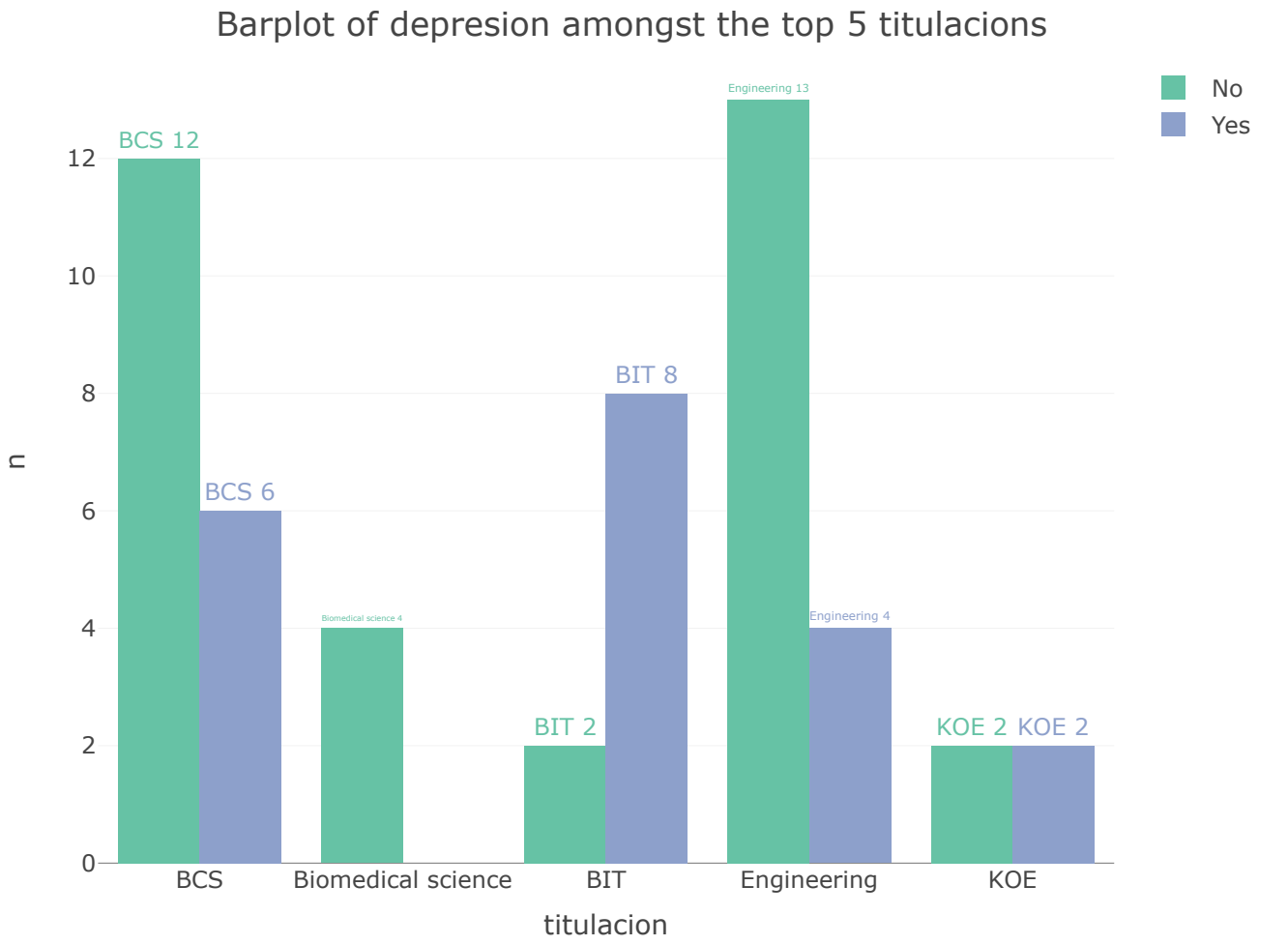



```
rf_rfe$optVariables
```

```
## [1] "titulacion_BIT"
```

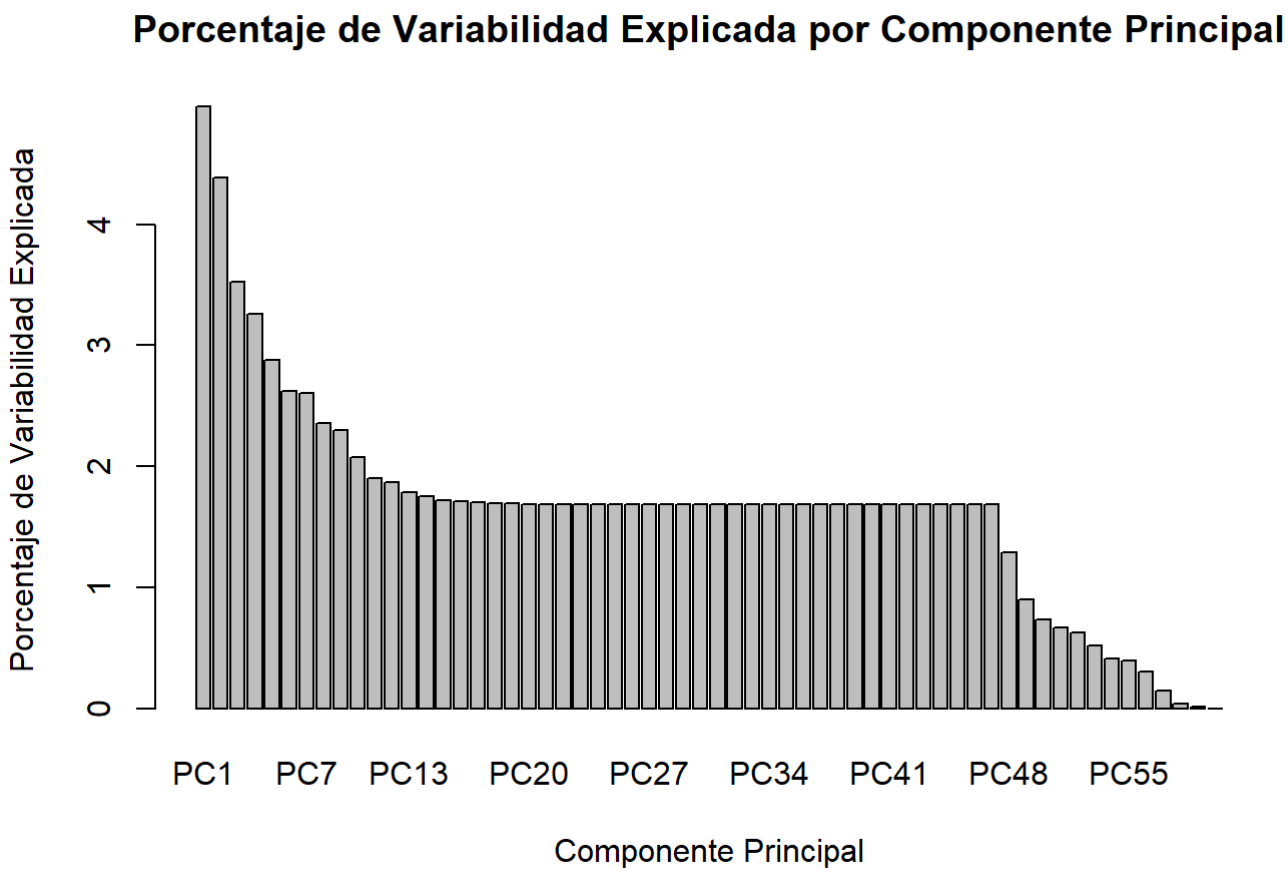
```
datosMental %>%
  filter(grepl('BIT|KOE|BCS|Engineering|Biomedical science', titulacion)) %>%
  count(titulacion, ansiedad, sort = T) %>%
  group_by(titulacion) %>%
  mutate(prop = round((n / sum(n)),digits = 4)) %>%
  plot_ly(x = ~titulacion, y=~n, color = ~ansiedad, type = "bar",
          text = ~paste(titulacion, n),
          textposition = 'outside') %>%
  layout(barmode = 'Stacked',
         title = 'Barplot of depresion amongst the top 5 titulacions')
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels

## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning reques
ted palette with 3 different levels
```



PCA - KNN

```
# Aplicar PCA a los datos
pca <- prcomp(datos_matrizCor, scale. = TRUE)

# Obtener el porcentaje de variabilidad explicada por cada componente
porcentaje_variabilidad <- pca$sdev^2 / sum(pca$sdev^2) * 100

# Crear el gráfico de barras
barplot(porcentaje_variabilidad, names.arg = paste0("PC", 1:length(porcentaje_variabilidad)),
        xlab = "Componente Principal", ylab = "Porcentaje de Variabilidad Explicada",
        main = "Porcentaje de Variabilidad Explicada por Componente Principal")
```




```
# Obtener las 13 mejores componentes principales
componentes_principales <- pca$x[, 1:13]

# Crear un nuevo dataset con las 13 mejores componentes principales
datasetPCA <- as.data.frame(componentes_principales)

# Añadir la variable objetivo al nuevo dataset, si corresponde
datasetPCA$ansiedad <- datosMental_Limpios$ansiedad
```

```
# Hiperparámetros
hiperparametros <- data.frame(k = c(1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25))

# AJUSTE DEL MODELO
# ============================================================================
set.seed(342)
modelo_knnPCA <- train(ansiedad ~ ., data = datasetPCA,
                       method = "knn",
                       tuneGrid = hiperparametros,
                       metric = "Accuracy",
                       trControl = control_train2)
modelo_knnPCA
```

```
## k-Nearest Neighbors
##
## 101 samples
##  13 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 81, 81, 80, 82, 80
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.8217043  0.5992674
##    3  0.8528070  0.6611777
##    5  0.8327569  0.5911317
##    7  0.8327569  0.5880289
##    9  0.8232331  0.5393680
##   11  0.8032331  0.4889246
##   13  0.7932331  0.4608295
##   15  0.7832331  0.4200816
##   17  0.7832331  0.4200816
##   19  0.7732331  0.3872770
##   21  0.7431830  0.2875356
##   23  0.7426566  0.2888465
##   25  0.7231328  0.2227953
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

COMPARATIVA FINAL

```
modelos <- list(KNN = modelo_knn,
                rf = modelo_rf,
                nb= modelo_nb,
                pcaKNN=modelo_knnPCA,
                wrapped= rf_rfe )

resultados_resamples <- resamples(modelos)
```

```
## Warning in resamples.default(modelos): 'wrapped' did not have
## 'returnResamp="final"; the optimal subset is used
```

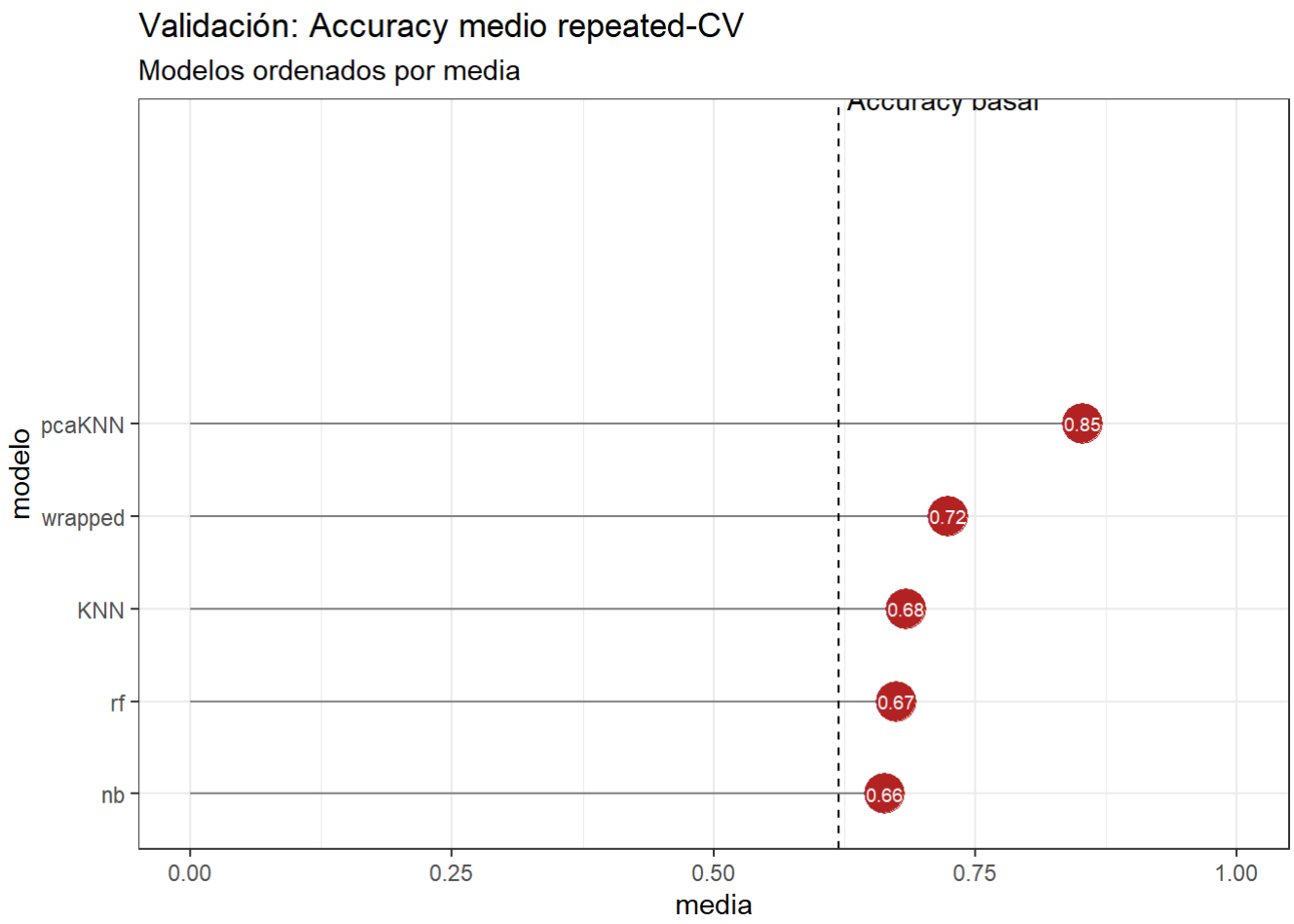
```
metricas_resamples <- resultados_resamples$values %>%
                        gather(key = "modelo", value = "valor", -Resample) %>%
                        separate(col = "modelo", into = c("modelo", "metrica"),
                                 sep = "~", remove = TRUE)
metricas_resamples %>% head()
```

	Resample	modelo	metrica	valor
	<chr>	<chr>	<chr>	<dbl>
1	Fold1	KNN	Accuracy	0.6500000

	Resample <chr>	modelo <chr>	metrica <chr>	valor <dbl>
2	Fold2	KNN	Accuracy	0.6500000
3	Fold3	KNN	Accuracy	0.7142857
4	Fold4	KNN	Accuracy	0.7368421
5	Fold5	KNN	Accuracy	0.6666667
6	Fold1	KNN	Kappa	0.0000000

6 rows

```
metricas_resamples %>%
  filter(metrica == "Accuracy") %>%
  group_by(modelo) %>%
  summarise(media = mean(valor)) %>%
  ggplot(aes(x = reorder(modelo, media), y = media, label = round(media, 2))) +
    geom_segment(aes(x = reorder(modelo, media), y = 0,
                     xend = modelo, yend = media),
                     color = "grey50") +
    geom_point(size = 7, color = "firebrick") +
    geom_text(color = "white", size = 2.5) +
    scale_y_continuous(limits = c(0, 1)) +
    # Accuracy basal
    geom_hline(yintercept = 0.62, linetype = "dashed") +
    annotate(geom = "text", y = 0.72, x = 8.5, label = "Accuracy basal") +
    labs(title = "Validación: Accuracy medio repeated-CV",
         subtitle = "Modelos ordenados por media",
         x = "modelo") +
    coord_flip() +
    theme_bw()
```


Validación: Accuracy medio repeated-CV
Modelos ordenados por media
Accuracy basal
modelo
pcaKNN
wrapped
KNN
rf
nb
0.85
0.72
0.68
0.67
0.66
0.00
0.25
0.50
0.75
1.00
media