Trabajo Clasificación Estadística del conjunto de datos 'Speedating'

Ruben

2023-04-27

Apertura de los datos y preprocesamiento estadístico

```
datos <- read.csv("C:\\Users\\Cosas\\OneDrive\\Escritorio\\ESTADISTICA\\MINERIA DE DATOS\\speeddating_p
head(datos,5)
```

```
##   age wave gender age_o d_age samerace importance_same_race
## 1  21    1 female    27     6        0                    2
## 2  21    1 female    22     1        0                    2
## 3  21    1 female    22     1        1                    2
## 4  21    1 female    23     2        0                    2
## 5  21    1 female    24     3        0                    2
##   importance_same_religion pref_o_attractive pref_o_sincere pref_o_intelligence
## 1                        4                35             20                  20
## 2                        4                60              0                   0
## 3                        4                19             18                  19
## 4                        4                30              5                  15
## 5                        4                30             10                  20
##   pref_o_funny pref_o_ambitious pref_o_shared_interests attractive_o sinsere_o
## 1           20                0                       5            6         8
## 2           40                0                       0            7         8
## 3           18               14                      12           10        10
## 4           40                5                       5            7         8
## 5           10               10                      20            8         7
##   intelligence_o funny_o ambitous_o shared_interests_o attractive_important
## 1              8       8          8                  6                   15
## 2             10       7          7                  5                   15
## 3             10      10         10                 10                   15
## 4              9       8          9                  8                   15
## 5              9       6          9                  7                   15
##   sincere_important intellicence_important funny_important ambtition_important
## 1                20                     20              15                  15
## 2                20                     20              15                  15
## 3                20                     20              15                  15
## 4                20                     20              15                  15
## 5                20                     20              15                  15
##   shared_interests_important attractive sincere intelligence funny ambition
## 1                         15          6       8            8     8        7
## 2                         15          6       8            8     8        7
## 3                         15          6       8            8     8        7
## 4                         15          6       8            8     8        7
## 5                         15          6       8            8     8        7
##   attractive_partner sincere_partner intelligence_partner funny_partner
```

```
## 1                  6               9                    7             7
## 2                  7               8                    7             8
## 3                  5               8                    9             8
## 4                  7               6                    8             7
## 5                  5               6                    7             7
##   ambition_partner shared_interests_partner sports tvsports exercise dining
## 1                6                        5      9        2        8      9
## 2                5                        6      9        2        8      9
## 3                5                        7      9        2        8      9
## 4                6                        8      9        2        8      9
## 5                6                        6      9        2        8      9
##   museums art hiking gaming clubbing reading tv theater movies concerts music
## 1       1   1      5      1        5       6  9       1     10       10     9
## 2       1   1      5      1        5       6  9       1     10       10     9
## 3       1   1      5      1        5       6  9       1     10       10     9
## 4       1   1      5      1        5       6  9       1     10       10     9
## 5       1   1      5      1        5       6  9       1     10       10     9
##   shopping yoga interests_correlate expected_happy_with_sd_people
## 1        8    1                0.14                             3
## 2        8    1                0.54                             3
## 3        8    1                0.16                             3
## 4        8    1                0.61                             3
## 5        8    1                0.21                             3
##   expected_num_interested_in_me expected_num_matches like guess_prob_liked met
## 1                             2                    4    7                6   0
## 2                             2                    4    7                5   1
## 3                             2                    4    7               NA   1
## 4                             2                    4    7                6   0
## 5                             2                    4    6                6   0
##   match
## 1     0
## 2     0
## 3     1
## 4     1
## 5     1
```

```
colSums(is.na(datos))
```

```
##                           age                          wave
##                            95                             0
##                        gender                         age_o
##                             0                           104
##                         d_age                      samerace
##                             0                             0
##          importance_same_race      importance_same_religion
##                            79                            79
##             pref_o_attractive               pref_o_sincere
##                            89                            89
##           pref_o_intelligence                 pref_o_funny
##                            89                            98
##              pref_o_ambitious       pref_o_shared_interests
##                           107                           129
##                  attractive_o                    sinsere_o
##                           212                           287
##                intelligence_o                       funny_o
```

```
##                           306                            360
##                    ambitous_o             shared_interests_o
##                           722                           1076
##          attractive_important              sincere_important
##                            79                             79
##        intellicence_important                funny_important
##                            79                             89
##           ambtition_important     shared_interests_important
##                            99                            121
##                    attractive                        sincere
##                           105                            105
##                  intelligence                          funny
##                           105                            105
##                      ambition             attractive_partner
##                           105                            202
##               sincere_partner           intelligence_partner
##                           277                            296
##                 funny_partner               ambition_partner
##                           350                            712
##      shared_interests_partner                         sports
##                          1067                             79
##                      tvsports                       exercise
##                            79                             79
##                        dining                        museums
##                            79                             79
##                           art                         hiking
##                            79                             79
##                        gaming                       clubbing
##                            79                             79
##                       reading                             tv
##                            79                             79
##                       theater                         movies
##                            79                             79
##                      concerts                          music
##                            79                             79
##                      shopping                           yoga
##                            79                             79
##           interests_correlate expected_happy_with_sd_people
##                           158                            101
## expected_num_interested_in_me           expected_num_matches
##                          6578                           1173
##                          like                guess_prob_liked
##                           240                            309
##                           met                          match
##                           375                              0
```

```
datos_filtrado <- datos[rowSums(is.na(datos)) <= ncol(datos) - 60, ]
# Eliminamos las instancias con más de 2 valores faltantes.
```

```
#expected_num_interested_in_me tiene 6074 datos faltantes y expected_num_matches tiene 1075, son
#variables que en principio no intervienen en que la persona tenga un match o no asi que los
#vamos a eliminar.
datos_filtrado2 <- datos_filtrado[,-c(57,58)]

#covertimos la variable género en binaria 0,1 ya que hay algunas herramientas que voy a
```

```r
#utilizar que sólo permiten valores numéricos, 0 implica femenino y 1 masculino.
datos_filtrado2$gender <- ifelse(datos_filtrado2$gender == "female", 0, 1)
```

```r
porc_perdidos <- function(x) sum(is.na(x))/length(x)
round(100*apply(datos_filtrado2, 2, porc_perdidos), 1)
```

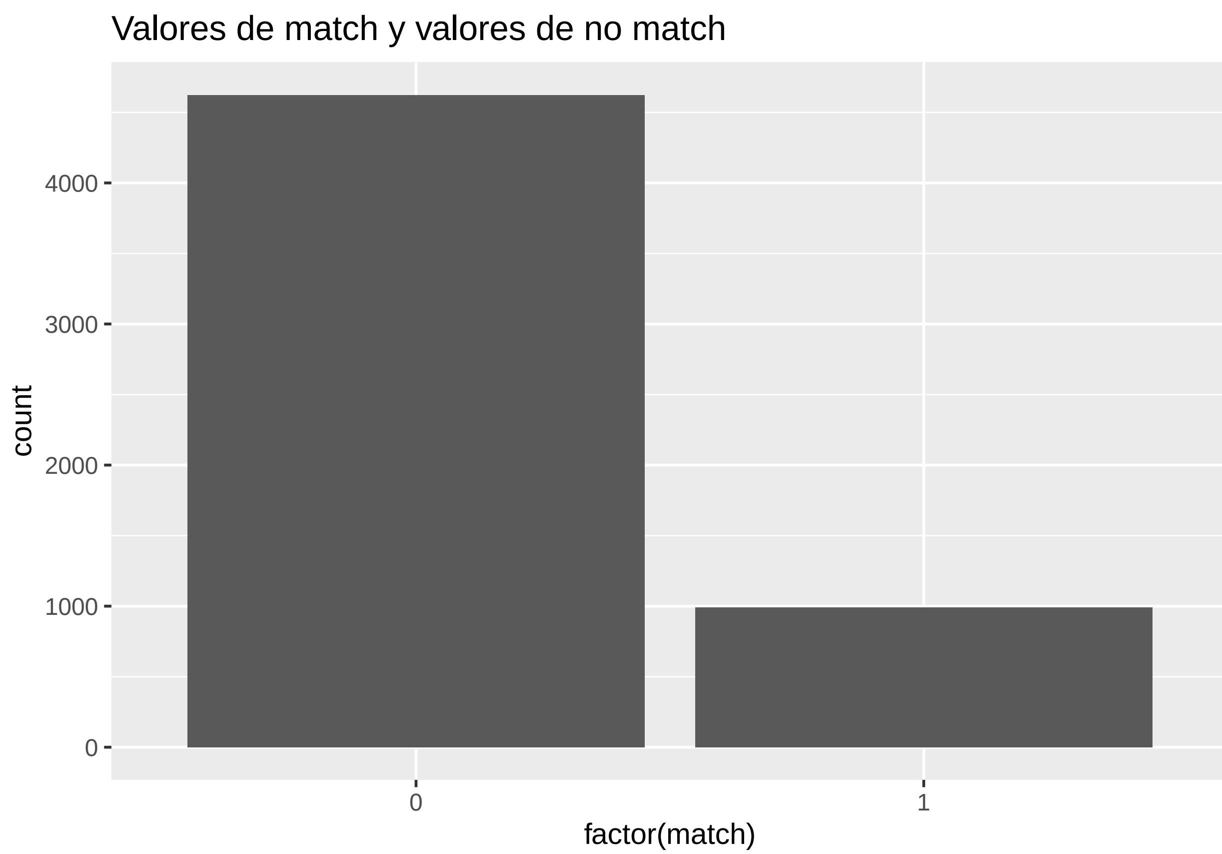
```
##                        age                       wave
##                        0.0                        0.0
##                     gender                      age_o
##                        0.0                        0.2
##                      d_age                   samerace
##                        0.0                        0.0
##        importance_same_race   importance_same_religion
##                        0.0                        0.0
##          pref_o_attractive            pref_o_sincere
##                        0.0                        0.0
##        pref_o_intelligence              pref_o_funny
##                        0.0                        0.0
##          pref_o_ambitious    pref_o_shared_interests
##                        0.0                        0.2
##               attractive_o                 sinsere_o
##                        0.1                        0.1
##             intelligence_o                    funny_o
##                        0.2                        0.3
##                 ambitous_o         shared_interests_o
##                        2.0                        5.8
##       attractive_important          sincere_important
##                        0.0                        0.0
##     intellicence_important            funny_important
##                        0.0                        0.0
##        ambtition_important  shared_interests_important
##                        0.0                        0.3
##                 attractive                    sincere
##                        0.0                        0.0
##               intelligence                      funny
##                        0.0                        0.0
##                   ambition         attractive_partner
##                        0.0                        0.1
##            sincere_partner       intelligence_partner
##                        0.1                        0.2
##              funny_partner           ambition_partner
##                        0.3                        2.2
##   shared_interests_partner                     sports
##                        5.9                        0.0
##                   tvsports                   exercise
##                        0.0                        0.0
##                     dining                    museums
##                        0.0                        0.0
##                        art                     hiking
##                        0.0                        0.0
##                     gaming                   clubbing
##                        0.0                        0.0
##                    reading                         tv
##                        0.0                        0.0
```

```
##                   theater                     movies
##                       0.0                        0.0
##                  concerts                      music
##                       0.0                        0.0
##                  shopping                       yoga
##                       0.0                        0.0
##       interests_correlate expected_happy_with_sd_people
##                       0.0                        0.0
##                      like            guess_prob_liked
##                       0.1                        0.3
##                       met                      match
##                       1.4                        0.0
```

```
#como el porcentaje mayor es alrededor de 6 y es una cifra relativamente pequeña, eliminamos las
#instancias con valores faltantes.

datos_final <- na.omit(datos_filtrado2)

write.csv(datos_final, file = 'speed_dating_final_data.csv')
```

Visualización de los datos. Gráficas de variables según match

```
library(ggplot2)
ggplot(datos_final, aes(x = factor(match))) +
  geom_bar() +
  ggtitle("Valores de match y valores de no match")
```



Los datos según match están muy desbalanceados, esto implicaría que un modelo con todos los valores de match predichos con 0 tendría muy alto nivel de precisión aunque en realidad no estaría prediciendo 'inteligentemente'.

```
#balanceamos los datos
datos_match_0 <- datos_final[datos_final$match == 0, ]
datos_match_1 <- datos_final[datos_final$match == 1, ]

n_match_0 <- nrow(datos_match_0)
n_match_1 <- nrow(datos_match_1)

set.seed(9202) # para que los resultados sean reproducibles
if (n_match_0 > n_match_1) {
  datos_match_0 <- datos_match_0[sample(1:nrow(datos_match_0), n_match_1), ]
} else {
  datos_match_1 <- datos_match_1[sample(1:nrow(datos_match_1), n_match_0), ]
}

datos_balanceados <- rbind(datos_match_0, datos_match_1)


# reordenamos aleatoriamente las filas
random_indices <- sample(nrow(datos_balanceados))
datos_balanceados <- datos_balanceados[random_indices,]


#Normalizamos los datos para que estén en la misma escala.
```

```
datos_final <- as.data.frame(datos_balanceados)

write.csv(datos_final, 'speed_dating_data_balanced.csv', row.names = TRUE)
datos_final_sc <- as.data.frame(scale(datos_balanceados))
```

HAY VARIAS FORMAS DE QUEDARNOS CON LAS CARACTERISTICAS MAS IMPORTANTES DEL DATASET:

SELECCION STEPWISE

SELECCION POR FUERZA BRUTA: COMO REGSUBSETS

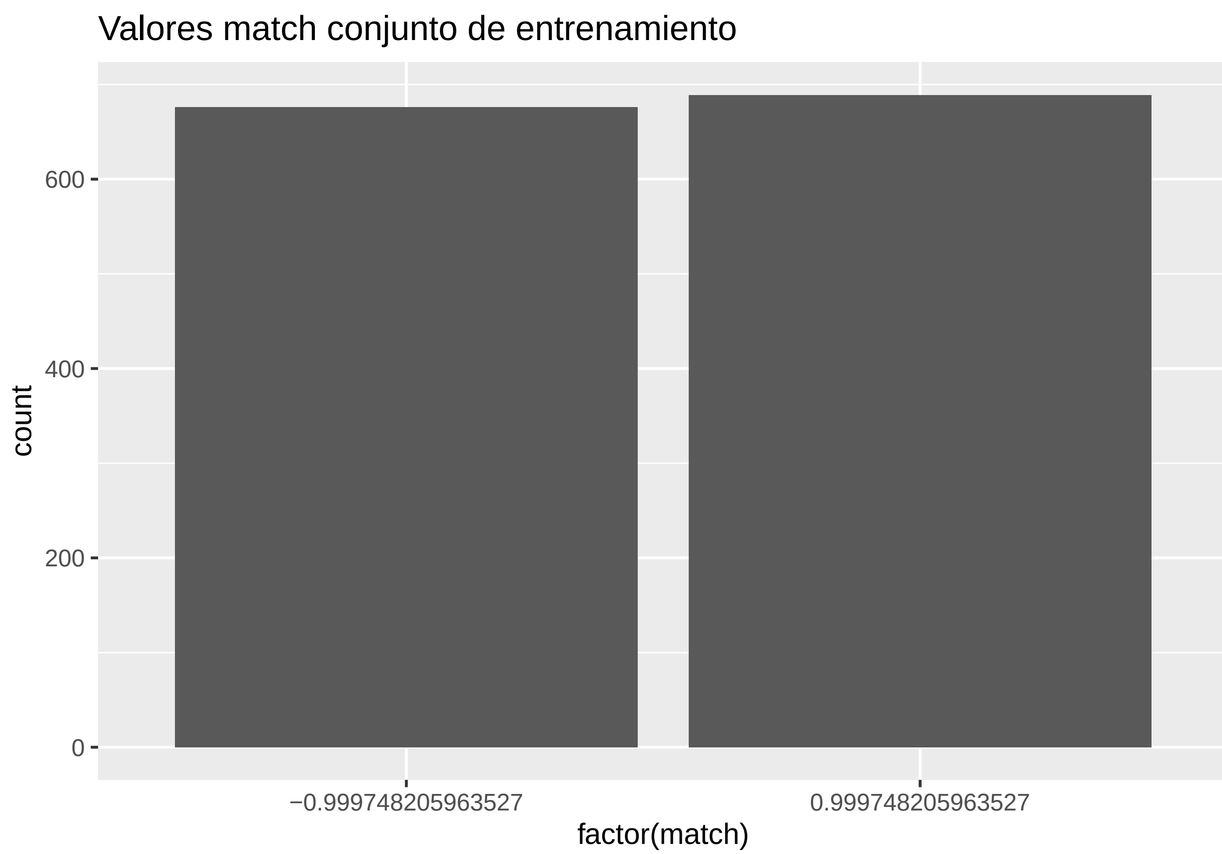
SELECCION POR MODELOS: COMO LASSO O RIDGE

SELECCION POR UMBRAL DE CORRELACION

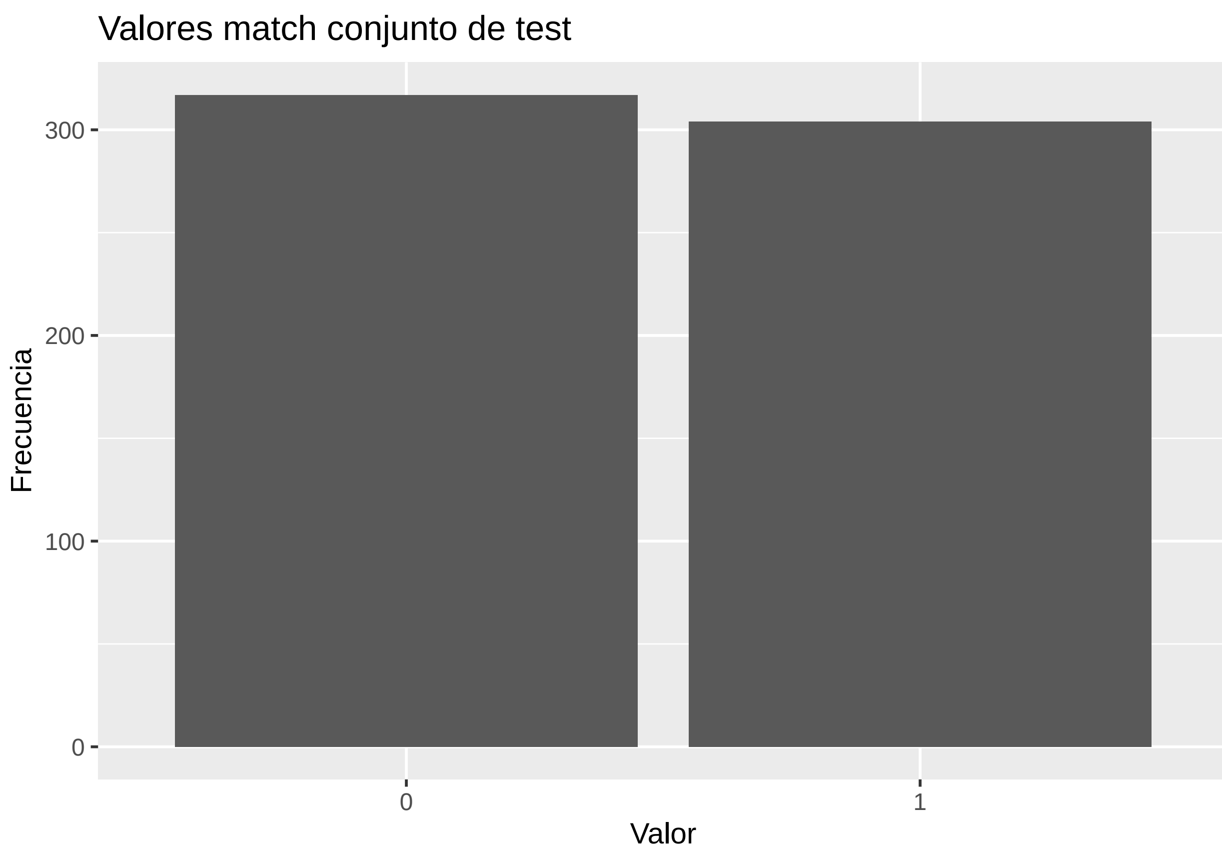
Técnicas de reducción de dimensionalidad

Método StepWise

```
#Dividimos el conjunto de datos para el entrenamiento y para la evaluación
set.seed(9202)
rand_num <- runif(nrow(datos_final_sc))

test <- datos_final_sc[rand_num>0.7,-60]
match_reales_test <- datos_final[rand_num>0.7, 'match']
match_train <- as.data.frame(datos_balanceados[rand_num<=0.7,'match'])
colnames(match_train) <- 'match'

datos_final <- datos_final[rand_num <=0.7, ]
datos_final_sc <- datos_final_sc[rand_num <= 0.7, ]

#Verificamos que los datasets de entrenamiento y validación están balanceados.
ggplot(datos_final_sc, aes(x = factor(match))) +
  geom_bar() +
  ggtitle("Valores match conjunto de entrenamiento")
```




```
tabla <- table(match_reales_test)
ggplot() +
  geom_col(data = as.data.frame(tabla), aes(x = match_reales_test, y = Freq)) +
  labs(x = "Valor", y = "Frecuencia") +
  ggtitle("Valores match conjunto de test")
```




```r
# SELECCION POR METODO STEPWISE
modelo <- lm(match ~ . , data = datos_final_sc)
summary(modelo)
```

```
##
## Call:
## lm(formula = match ~ ., data = datos_final_sc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87776 -0.63737  0.08386  0.63222  2.13616
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.0014180  0.0216325   0.066 0.947745
## age                           0.0054526  0.0253896   0.215 0.829991
## wave                         -0.0068193  0.0237725  -0.287 0.774267
## gender                       -0.0256702  0.0320305  -0.801 0.423028
## age_o                         0.0405917  0.0248352   1.634 0.102407
## d_age                        -0.0276485  0.0257184  -1.075 0.282552
## samerace                     -0.0424823  0.0224769  -1.890 0.058974 .
## importance_same_race         -0.0719847  0.0283147  -2.542 0.011127 *
## importance_same_religion      0.0346786  0.0269740   1.286 0.198801
## pref_o_attractive             0.0854558  0.1528085   0.559 0.576098
## pref_o_sincere                0.0302542  0.0863042   0.351 0.725980
## pref_o_intelligence           0.1130401  0.0836228   1.352 0.176678
```

```
## pref_o_funny                   0.0798718  0.0769011   1.039 0.299170
## pref_o_ambitious              -0.0036906  0.0758948  -0.049 0.961223
## pref_o_shared_interests        0.0593912  0.0756004   0.786 0.432248
## attractive_o                   0.1420419  0.0301853   4.706 2.80e-06 ***
## sinsere_o                     -0.0477924  0.0304039  -1.572 0.116213
## intelligence_o                 0.0328672  0.0338570   0.971 0.331845
## funny_o                        0.1804249  0.0344413   5.239 1.88e-07 ***
## ambitous_o                    -0.0287158  0.0301954  -0.951 0.341782
## shared_interests_o             0.1189556  0.0301081   3.951 8.20e-05 ***
## attractive_important          -0.1395848  0.1055384  -1.323 0.186201
## sincere_important             -0.1020767  0.0674139  -1.514 0.130222
## intellicence_important        -0.0599972  0.0625449  -0.959 0.337602
## funny_important               -0.0412729  0.0598211  -0.690 0.490356
## ambtition_important           -0.0415690  0.0552247  -0.753 0.451751
## shared_interests_important    -0.0622927  0.0588455  -1.059 0.289987
## attractive                    -0.0161124  0.0300832  -0.536 0.592330
## sincere                        0.0058634  0.0263901   0.222 0.824208
## intelligence                  -0.0494105  0.0300929  -1.642 0.100845
## funny                         -0.0223358  0.0267726  -0.834 0.404278
## ambition                       0.0143846  0.0285934   0.503 0.614998
## attractive_partner             0.1380314  0.0318247   4.337 1.55e-05 ***
## sincere_partner               -0.0303376  0.0319792  -0.949 0.342966
## intelligence_partner           0.0383194  0.0351952   1.089 0.276457
## funny_partner                  0.0943328  0.0355649   2.652 0.008089 **
## ambition_partner              -0.0948726  0.0312944  -3.032 0.002480 **
## shared_interests_partner       0.0445883  0.0330259   1.350 0.177218
## sports                        -0.0539343  0.0303299  -1.778 0.075594 .
## tvsports                      -0.0449238  0.0296780  -1.514 0.130342
## exercise                       0.0259705  0.0259439   1.001 0.317000
## dining                        -0.0006517  0.0279119  -0.023 0.981375
## museums                       -0.0414436  0.0480065  -0.863 0.388136
## art                            0.1106279  0.0461622   2.397 0.016692 *
## hiking                        -0.0059625  0.0261471  -0.228 0.819654
## gaming                         0.0245466  0.0262894   0.934 0.350629
## clubbing                       0.0157089  0.0242223   0.649 0.516755
## reading                        0.0473399  0.0247240   1.915 0.055745 .
## tv                             0.0595008  0.0301016   1.977 0.048290 *
## theater                       -0.0602236  0.0325388  -1.851 0.064420 .
## movies                        -0.0324689  0.0277784  -1.169 0.242677
## concerts                       0.0649172  0.0346704   1.872 0.061373 .
## music                         -0.0554700  0.0326429  -1.699 0.089502 .
## shopping                      -0.0866528  0.0301413  -2.875 0.004107 **
## yoga                           0.0027284  0.0258609   0.106 0.915992
## interests_correlate            0.0169313  0.0240198   0.705 0.481004
## expected_happy_with_sd_people  0.0462091  0.0247541   1.867 0.062165 .
## like                           0.1744593  0.0380203   4.589 4.89e-06 ***
## guess_prob_liked               0.1065369  0.0284957   3.739 0.000193 ***
## met                           -0.0184967  0.0272482  -0.679 0.497372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7944 on 1305 degrees of freedom
## Multiple R-squared:  0.3963, Adjusted R-squared:  0.3691
## F-statistic: 14.52 on 59 and 1305 DF,  p-value: < 2.2e-16
```

```r
r2_modelo1 <- summary(modelo)$adj.r
modelo_aic <- AIC(modelo)
modelo_bic <- BIC(modelo)
variables_modelo1 <- names(coef(modelo))[-1]
```

```r
#vemos como funciona el nuevo modelo optimizando el AIC
modelo2<- step(modelo, direction = 'both', trace = 0, k=2)
summary(modelo2)
```

```
##
## Call:
## lm(formula = match ~ age_o + samerace + importance_same_race +
##     importance_same_religion + pref_o_attractive + pref_o_intelligence +
##     pref_o_funny + pref_o_shared_interests + attractive_o + sinsere_o +
##     funny_o + shared_interests_o + attractive_important + sincere_important +
##     intelligence + attractive_partner + funny_partner + ambition_partner +
##     shared_interests_partner + sports + tvsports + art + reading +
##     tv + theater + concerts + music + shopping + expected_happy_with_sd_people +
##     like + guess_prob_liked, data = datos_final_sc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94485 -0.64237  0.07843  0.63610  2.10878
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.001581   0.021458   0.074  0.94129
## age_o                          0.040682   0.022494   1.809  0.07074 .
## samerace                      -0.045827   0.021992  -2.084  0.03737 *
## importance_same_race          -0.075979   0.026601  -2.856  0.00435 **
## importance_same_religion       0.042011   0.025727   1.633  0.10271
## pref_o_attractive              0.066998   0.030156   2.222  0.02647 *
## pref_o_intelligence            0.103594   0.026254   3.946 8.37e-05 ***
## pref_o_funny                   0.070512   0.025243   2.793  0.00529 **
## pref_o_shared_interests        0.046126   0.026969   1.710  0.08744 .
## attractive_o                   0.143367   0.028964   4.950 8.38e-07 ***
## sinsere_o                     -0.039933   0.025148  -1.588  0.11254
## funny_o                        0.188813   0.032389   5.830 6.96e-09 ***
## shared_interests_o             0.111057   0.028782   3.859  0.00012 ***
## attractive_important          -0.046361   0.025534  -1.816  0.06965 .
## sincere_important             -0.043094   0.024496  -1.759  0.07877 .
## intelligence                  -0.049195   0.024740  -1.988  0.04696 *
## attractive_partner             0.134305   0.030804   4.360 1.40e-05 ***
## funny_partner                  0.083207   0.033914   2.453  0.01428 *
## ambition_partner              -0.077183   0.027319  -2.825  0.00479 **
## shared_interests_partner       0.055241   0.032015   1.725  0.08467 .
## sports                        -0.044092   0.027273  -1.617  0.10618
## tvsports                      -0.041529   0.027503  -1.510  0.13129
## art                            0.075691   0.027805   2.722  0.00657 **
## reading                        0.041156   0.023102   1.781  0.07506 .
## tv                             0.052947   0.027641   1.916  0.05564 .
## theater                       -0.062830   0.028987  -2.168  0.03037 *
## concerts                       0.058616   0.032201   1.820  0.06893 .
## music                         -0.053727   0.030729  -1.748  0.08063 .
```

```
## shopping                       -0.074755   0.026363  -2.836  0.00464 **
## expected_happy_with_sd_people   0.038807   0.022822   1.700  0.08929 .
## like                            0.171624   0.036852   4.657 3.53e-06 ***
## guess_prob_liked                0.106895   0.027375   3.905 9.90e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7909 on 1333 degrees of freedom
## Multiple R-squared:  0.3887, Adjusted R-squared:  0.3745
## F-statistic: 27.34 on 31 and 1333 DF,  p-value: < 2.2e-16
```

```r
r2_modelo2 <- summary(modelo2)$adj.r

modelo2_aic <- AIC(modelo2)
modelo2_bic <- BIC(modelo2)
variables_modelo2 <- names(coef(modelo2))[-1] # se omite el primer elemento (Intercept)

#se ha quedado con unas 30 variables, la mitad del dataset original.
```

```r
modelo2$anova
```

```
##                            Step Df     Deviance Resid. Df Resid. Dev       AIC
## 1                               NA           NA      1305   823.4998 -569.7978
## 2                      - dining  1 0.0003440279      1306   823.5002 -571.7972
## 3             - pref_o_ambitious  1 0.0014890054      1307   823.5017 -573.7947
## 4                        - yoga  1 0.0065939564      1308   823.5083 -575.7838
## 5                         - age  1 0.0279104484      1309   823.5362 -577.7375
## 6                      - hiking  1 0.0235260341      1310   823.5597 -579.6985
## 7                     - sincere  1 0.0232254940      1311   823.5829 -581.6601
## 8                        - wave  1 0.0366100884      1312   823.6195 -583.5994
## 9                  - attractive  1 0.1525737292      1313   823.7721 -585.3465
## 10                   - ambition  1 0.1855281681      1314   823.9576 -587.0391
## 11                        - met  1 0.2677132716      1315   824.2254 -588.5957
## 12         - ambtition_important  1 0.2882299960      1316   824.5136 -590.1185
## 13             - funny_important  1 0.0369872690      1317   824.5506 -592.0572
## 14         - interests_correlate  1 0.2921158349      1318   824.8427 -593.5737
## 15                     - gender  1 0.3102919739      1319   825.1530 -595.0603
## 16                   - clubbing  1 0.3296955517      1320   825.4827 -596.5151
## 17     - intellicence_important  1 0.4051974647      1321   825.8879 -597.8452
## 18 - shared_interests_important  1 0.2236581312      1322   826.1115 -599.4756
## 19             - intelligence_o  1 0.5150366268      1323   826.6266 -600.6248
## 20                 - ambitous_o  1 0.2558895652      1324   826.8825 -602.2024
## 21                     - gaming  1 0.5786450416      1325   827.4611 -603.2475
## 22                    - museums  1 0.6612731826      1326   828.1224 -604.1571
## 23                   - exercise  1 0.6889138372      1327   828.8113 -605.0220
## 24                     - movies  1 0.7508730684      1328   829.5622 -605.7859
## 25             - sincere_partner  1 0.7795533847      1329   830.3417 -606.5038
## 26        - intelligence_partner  1 0.3611812258      1330   830.7029 -607.9102
## 27                      - funny  1 0.9349443499      1331   831.6378 -608.3748
## 28                      - d_age  1 1.1112161224      1332   832.7491 -608.5521
## 29               - pref_o_sincere  1 1.1552893549      1333   833.9043 -608.6597
```

```r
#ELIMINA TODAS ESTAS VARIABLES HASTA QUEDARSE CON EL MEJOR VALOR AIC
```

```r
#sustituimos el k=2 por log(n) para optimizar el BIC
modelo3<- step(modelo, direction = 'both', trace = 0, k=log(1365))
```

```r
summary(modelo3)
```

```
##
## Call:
## lm(formula = match ~ pref_o_intelligence + attractive_o + funny_o +
##     shared_interests_o + attractive_partner + sports + art +
##     shopping + like + guess_prob_liked, data = datos_final_sc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.24878 -0.66515  0.05455  0.65188  2.08600
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.0002408  0.0218122  -0.011 0.991193
## pref_o_intelligence  0.0721319  0.0216243   3.336 0.000874 ***
## attractive_o         0.1263450  0.0286240   4.414 1.10e-05 ***
## funny_o              0.1674483  0.0313936   5.334 1.13e-07 ***
## shared_interests_o   0.1312180  0.0284188   4.617 4.26e-06 ***
## attractive_partner   0.1525777  0.0299202   5.099 3.89e-07 ***
## sports              -0.0658613  0.0222787  -2.956 0.003168 **
## art                  0.0666448  0.0231864   2.874 0.004112 **
## shopping            -0.0715114  0.0228574  -3.129 0.001794 **
## like                 0.2023741  0.0322184   6.281 4.51e-10 ***
## guess_prob_liked     0.1108817  0.0262028   4.232 2.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8052 on 1354 degrees of freedom
## Multiple R-squared:  0.3566, Adjusted R-squared:  0.3518
## F-statistic: 75.03 on 10 and 1354 DF,  p-value: < 2.2e-16
```

```r
r2_modelo3 <- summary(modelo3)$adj.r
modelo3_aic <- AIC(modelo3)
modelo3_bic <- BIC(modelo3)
variables_modelo3 <- names(coef(modelo3))[-1]
```

Regresión de mejores subconjuntos

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.2.3
```

```r
modelo_subsets <- regsubsets(match ~ ., data = datos_final_sc, nvmax=10, really.big = T)
```

```r
#esta salida nos da los mejores modelos de cada tamaño.
set.seed(9202)
subsets_summary <- summary(modelo_subsets)
```

```r
subsets_summary
```

```
## Subset selection object
## Call: regsubsets.formula(match ~ ., data = datos_final_sc, nvmax = 10,
##     really.big = T)
## 59 Variables  (and intercept)
```

```
##                              Forced in Forced out
## age                              FALSE      FALSE
## wave                             FALSE      FALSE
## gender                           FALSE      FALSE
## age_o                            FALSE      FALSE
## d_age                            FALSE      FALSE
## samerace                         FALSE      FALSE
## importance_same_race             FALSE      FALSE
## importance_same_religion         FALSE      FALSE
## pref_o_attractive                FALSE      FALSE
## pref_o_sincere                   FALSE      FALSE
## pref_o_intelligence              FALSE      FALSE
## pref_o_funny                     FALSE      FALSE
## pref_o_ambitious                 FALSE      FALSE
## pref_o_shared_interests          FALSE      FALSE
## attractive_o                     FALSE      FALSE
## sinsere_o                        FALSE      FALSE
## intelligence_o                   FALSE      FALSE
## funny_o                          FALSE      FALSE
## ambitous_o                       FALSE      FALSE
## shared_interests_o               FALSE      FALSE
## attractive_important             FALSE      FALSE
## sincere_important                FALSE      FALSE
## intellicence_important           FALSE      FALSE
## funny_important                  FALSE      FALSE
## ambtition_important              FALSE      FALSE
## shared_interests_important       FALSE      FALSE
## attractive                       FALSE      FALSE
## sincere                          FALSE      FALSE
## intelligence                     FALSE      FALSE
## funny                            FALSE      FALSE
## ambition                         FALSE      FALSE
## attractive_partner               FALSE      FALSE
## sincere_partner                  FALSE      FALSE
## intelligence_partner             FALSE      FALSE
## funny_partner                    FALSE      FALSE
## ambition_partner                 FALSE      FALSE
## shared_interests_partner         FALSE      FALSE
## sports                           FALSE      FALSE
## tvsports                         FALSE      FALSE
## exercise                         FALSE      FALSE
## dining                           FALSE      FALSE
## museums                          FALSE      FALSE
## art                              FALSE      FALSE
## hiking                           FALSE      FALSE
## gaming                           FALSE      FALSE
## clubbing                         FALSE      FALSE
## reading                          FALSE      FALSE
## tv                               FALSE      FALSE
## theater                          FALSE      FALSE
## movies                           FALSE      FALSE
## concerts                         FALSE      FALSE
## music                            FALSE      FALSE
## shopping                         FALSE      FALSE
```

```
## yoga                             FALSE      FALSE
## interests_correlate              FALSE      FALSE
## expected_happy_with_sd_people    FALSE      FALSE
## like                             FALSE      FALSE
## guess_prob_liked                 FALSE      FALSE
## met                              FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           age wave gender age_o d_age samerace importance_same_race
## 1  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 2  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 3  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 4  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 5  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 6  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 7  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 8  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 9  ( 1 )  " " " "  " "    " "   " "   " "      " "
## 10  ( 1 ) " " " "  " "    " "   " "   " "      "*"
##           importance_same_religion pref_o_attractive pref_o_sincere
## 1  ( 1 )  " "                      " "               " "
## 2  ( 1 )  " "                      " "               " "
## 3  ( 1 )  " "                      " "               " "
## 4  ( 1 )  " "                      " "               " "
## 5  ( 1 )  " "                      " "               " "
## 6  ( 1 )  " "                      " "               " "
## 7  ( 1 )  " "                      " "               " "
## 8  ( 1 )  " "                      " "               " "
## 9  ( 1 )  " "                      " "               " "
## 10  ( 1 ) " "                      " "               " "
##           pref_o_intelligence pref_o_funny pref_o_ambitious
## 1  ( 1 )  " "                 " "          " "
## 2  ( 1 )  " "                 " "          " "
## 3  ( 1 )  " "                 " "          " "
## 4  ( 1 )  " "                 " "          " "
## 5  ( 1 )  " "                 " "          " "
## 6  ( 1 )  " "                 " "          " "
## 7  ( 1 )  " "                 " "          " "
## 8  ( 1 )  "*"                 " "          " "
## 9  ( 1 )  "*"                 " "          " "
## 10  ( 1 ) "*"                 " "          " "
##           pref_o_shared_interests attractive_o sinsere_o intelligence_o funny_o
## 1  ( 1 )  " "                     " "          " "       " "            " "
## 2  ( 1 )  " "                     " "          " "       " "            "*"
## 3  ( 1 )  " "                     " "          " "       " "            "*"
## 4  ( 1 )  " "                     "*"          " "       " "            "*"
## 5  ( 1 )  " "                     "*"          " "       " "            "*"
## 6  ( 1 )  " "                     "*"          " "       " "            "*"
## 7  ( 1 )  " "                     "*"          " "       " "            "*"
## 8  ( 1 )  " "                     "*"          " "       " "            "*"
## 9  ( 1 )  " "                     "*"          " "       " "            "*"
## 10  ( 1 ) " "                     "*"          " "       " "            "*"
##           ambitous_o shared_interests_o attractive_important sincere_important
## 1  ( 1 )  " "        " "                " "                  " "
```

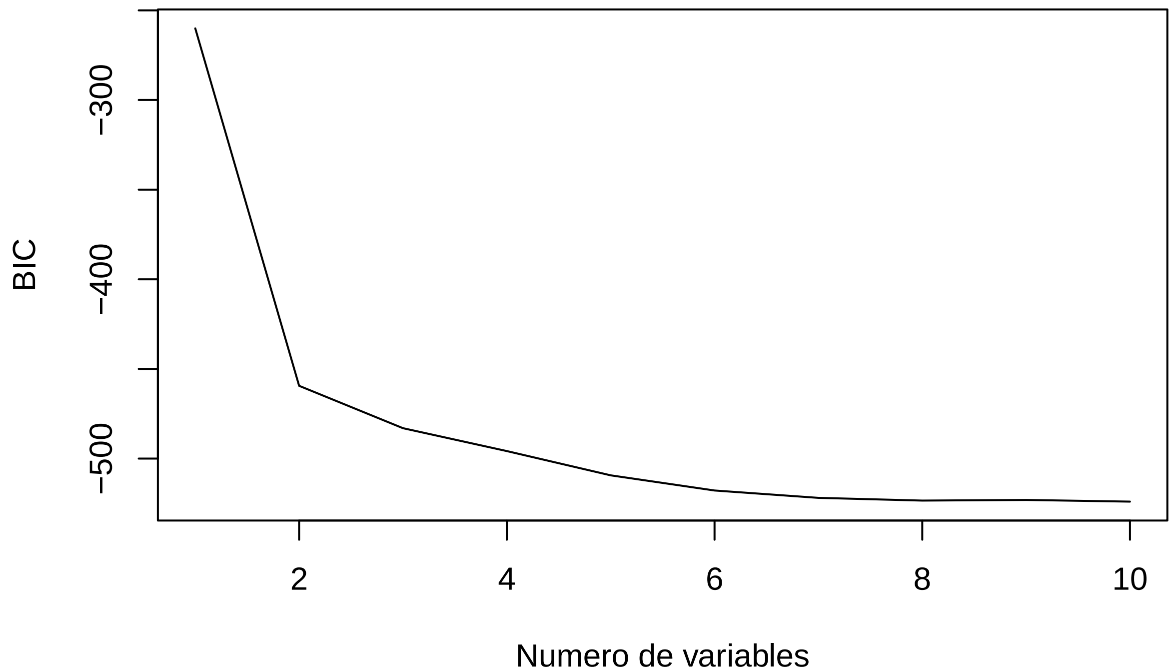
```
## 2  ( 1 )  " "        " "               " "                 " "
## 3  ( 1 )  " "        "*"               " "                 " "
## 4  ( 1 )  " "        "*"               " "                 " "
## 5  ( 1 )  " "        "*"               " "                 " "
## 6  ( 1 )  " "        "*"               " "                 " "
## 7  ( 1 )  " "        "*"               " "                 " "
## 8  ( 1 )  " "        "*"               " "                 " "
## 9  ( 1 )  " "        "*"               "*"                 " "
## 10  ( 1 ) " "        "*"               " "                 " "
##           intellicence_important funny_important ambtition_important
## 1  ( 1 )  " "                    " "             " "
## 2  ( 1 )  " "                    " "             " "
## 3  ( 1 )  " "                    " "             " "
## 4  ( 1 )  " "                    " "             " "
## 5  ( 1 )  " "                    " "             " "
## 6  ( 1 )  " "                    " "             " "
## 7  ( 1 )  " "                    " "             " "
## 8  ( 1 )  " "                    " "             " "
## 9  ( 1 )  " "                    " "             " "
## 10  ( 1 ) " "                    " "             " "
##           shared_interests_important attractive sincere intelligence funny
## 1  ( 1 )  " "                        " "        " "     " "          " "
## 2  ( 1 )  " "                        " "        " "     " "          " "
## 3  ( 1 )  " "                        " "        " "     " "          " "
## 4  ( 1 )  " "                        " "        " "     " "          " "
## 5  ( 1 )  " "                        " "        " "     " "          " "
## 6  ( 1 )  " "                        " "        " "     " "          " "
## 7  ( 1 )  " "                        " "        " "     " "          " "
## 8  ( 1 )  " "                        " "        " "     " "          " "
## 9  ( 1 )  " "                        " "        " "     " "          " "
## 10  ( 1 ) " "                        " "        " "     "*"          " "
##           ambition attractive_partner sincere_partner intelligence_partner
## 1  ( 1 )  " "      " "                " "             " "
## 2  ( 1 )  " "      " "                " "             " "
## 3  ( 1 )  " "      " "                " "             " "
## 4  ( 1 )  " "      " "                " "             " "
## 5  ( 1 )  " "      "*"                " "             " "
## 6  ( 1 )  " "      "*"                " "             " "
## 7  ( 1 )  " "      "*"                " "             " "
## 8  ( 1 )  " "      "*"                " "             " "
## 9  ( 1 )  " "      "*"                " "             " "
## 10  ( 1 ) " "      "*"                " "             " "
##           funny_partner ambition_partner shared_interests_partner sports
## 1  ( 1 )  " "           " "              " "                      " "
## 2  ( 1 )  " "           " "              " "                      " "
## 3  ( 1 )  " "           " "              " "                      " "
## 4  ( 1 )  " "           " "              " "                      " "
## 5  ( 1 )  " "           " "              " "                      " "
## 6  ( 1 )  " "           " "              " "                      " "
## 7  ( 1 )  " "           " "              " "                      " "
## 8  ( 1 )  " "           " "              " "                      " "
## 9  ( 1 )  " "           " "              " "                      " "
## 10  ( 1 ) " "           " "              " "                      " "
##           tvsports exercise dining museums art hiking gaming clubbing reading
```

```
## 1  ( 1 )  " "      " "      " "    " "     " " " "    " "    " "      " "
## 2  ( 1 )  " "      " "      " "    " "     " " " "    " "    " "      " "
## 3  ( 1 )  " "      " "      " "    " "     " " " "    " "    " "      " "
## 4  ( 1 )  " "      " "      " "    " "     " " " "    " "    " "      " "
## 5  ( 1 )  " "      " "      " "    " "     " " " "    " "    " "      " "
## 6  ( 1 )  " "      " "      " "    " "     " " " "    " "    " "      " "
## 7  ( 1 )  "*"      " "      " "    " "     " " " "    " "    " "      " "
## 8  ( 1 )  "*"      " "      " "    " "     " " " "    " "    " "      " "
## 9  ( 1 )  "*"      " "      " "    " "     " " " "    " "    " "      " "
## 10  ( 1 ) " "      " "      " "    " "     "*" " "    " "    " "      " "
##           tv  theater movies concerts music shopping yoga interests_correlate
## 1  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 2  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 3  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 4  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 5  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 6  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 7  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 8  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 9  ( 1 )  " " " "     " "    " "      " "   " "      " "  " "
## 10  ( 1 ) " " " "     " "    " "      " "   " "      " "  " "
##           expected_happy_with_sd_people like guess_prob_liked met
## 1  ( 1 )  " "                           "*"  " "              " "
## 2  ( 1 )  " "                           "*"  " "              " "
## 3  ( 1 )  " "                           "*"  " "              " "
## 4  ( 1 )  " "                           "*"  " "              " "
## 5  ( 1 )  " "                           "*"  " "              " "
## 6  ( 1 )  " "                           "*"  "*"              " "
## 7  ( 1 )  " "                           "*"  "*"              " "
## 8  ( 1 )  " "                           "*"  "*"              " "
## 9  ( 1 )  " "                           "*"  "*"              " "
## 10  ( 1 ) " "                           "*"  "*"              " "
```

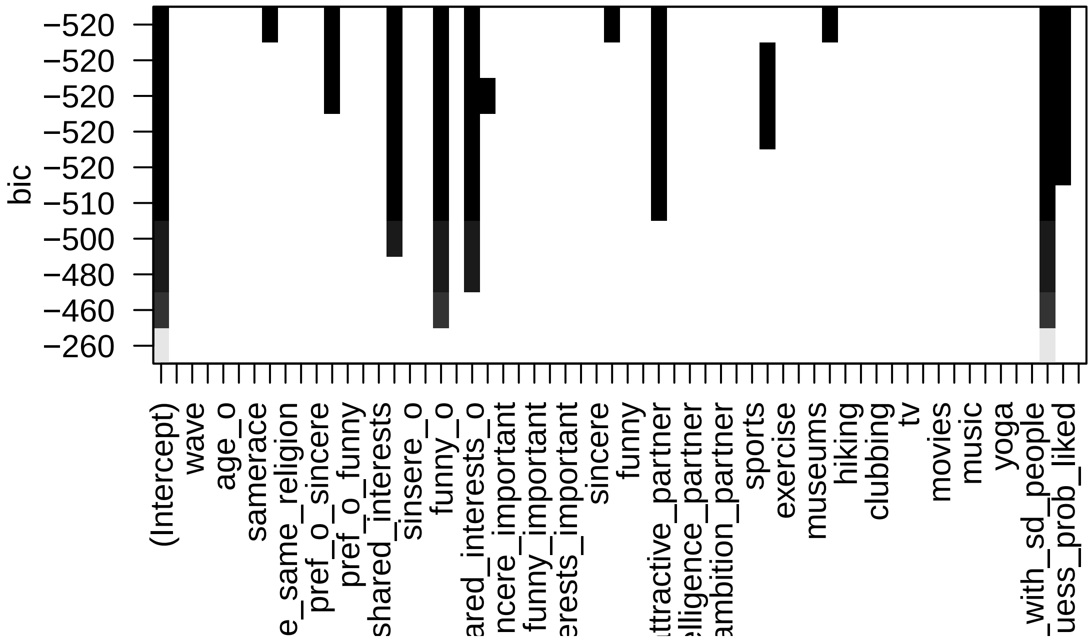
```r
which.min(subsets_summary$bic)
```

```
## [1] 10
```

```r
plot(subsets_summary$bic, xlab = 'Numero de variables', ylab = 'BIC', type = 'l')
```




```
plot(modelo_subsets, scale = 'bic')
```



```r
variables_fuerza <- names(coef(modelo_subsets, 10))[-1]
print('Las variables más importantes según este método son: ')
```

```
## [1] "Las variables más importantes según este método son: "
```

```r
variables_fuerza
```

```
##  [1] "importance_same_race" "pref_o_intelligence"  "attractive_o"
##  [4] "funny_o"              "shared_interests_o"   "intelligence"
##  [7] "attractive_partner"   "art"                  "like"
## [10] "guess_prob_liked"
```

```r
#habría sido conveniente ver como influiria en este análisis haber puesto como máximo de variables
#todas menos 'match' pero debido a la potencia de mi portátil tardaba demasiado.
```

Selección de variables por LASSO

```r
library('glmnet')
```

```
## Warning: package 'glmnet' was built under R version 4.2.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```r
x = as.matrix(datos_final[,1:59])
y =  datos_final[,60]

modelo_lasso <- glmnet(x,y, family= 'binomial', alpha=1)
```

```r
lasso_coefs <- coef(modelo_lasso, s=0)
sorted_indices <- order(abs(lasso_coefs), decreasing = TRUE)
```

```
## <sparse>[ <logic> ]: .M.sub.i.logical() maybe inefficient
```

```r
variables_lasso <- head(names(datos_final_sc)[sorted_indices-1],15)
#15 variables mas importantes en lasso
print('Las variables más importantes según el metodo de Lasso son: ')
```

```
## [1] "Las variables más importantes según el metodo de Lasso son: "
```

```r
variables_lasso
```

```
##  [1] "like"                 "funny_o"              "samerace"
##  [4] "attractive_o"         "attractive_partner"   "ambition_partner"
##  [7] "shared_interests_o"   "interests_correlate"  "guess_prob_liked"
## [10] "gender"               "funny_partner"        "art"
## [13] "intelligence"         "intelligence_partner" "music"
```

```r
#observamos que la regresion lasso elimina la variable funny important
tail(modelo_lasso$dev.ratio, 1)
```

```
## [1] 0.3725636
```

```r
#de momento es el r2 mas alto al que hemos llegado.
```

Selección de variables por método RIDGE

```r
#PROBEMOS CON LA REGRESION RIDGE

modelo_ridge <- glmnet(x,y, family= 'binomial', alpha=0)
coef(modelo_ridge, s=0)
```

```
## 60 x 1 sparse Matrix of class "dgCMatrix"
##                                         s1
## (Intercept)                  -9.106808e+00
## age                           2.592897e-05
## wave                          5.419800e-04
## gender                       -1.463659e-01
## age_o                         2.597199e-02
## d_age                        -2.000512e-02
## samerace                     -2.390510e-01
## importance_same_race         -6.961092e-02
## importance_same_religion      3.509879e-02
## pref_o_attractive            -4.930623e-04
## pref_o_sincere               -1.238405e-02
## pref_o_intelligence           2.540433e-02
## pref_o_funny                  2.116111e-02
## pref_o_ambitious             -2.261013e-02
## pref_o_shared_interests       5.475045e-03
## attractive_o                  2.163350e-01
## sinsere_o                    -4.533770e-02
## intelligence_o                5.881075e-02
## funny_o                       2.600207e-01
## ambitous_o                   -1.832925e-03
## shared_interests_o            1.632789e-01
```

```
## attractive_important          -8.515396e-03
## sincere_important             -1.210543e-02
## intellicence_important         1.873535e-03
## funny_important                1.286026e-02
## ambtition_important           -2.510418e-03
## shared_interests_important     2.319639e-03
## attractive                    -6.841733e-03
## sincere                        9.399014e-03
## intelligence                  -9.947537e-02
## funny                         -8.316958e-02
## ambition                       1.330358e-02
## attractive_partner             2.191958e-01
## sincere_partner               -1.511357e-02
## intelligence_partner           8.693331e-02
## funny_partner                  1.367442e-01
## ambition_partner              -1.245667e-01
## shared_interests_partner       8.312328e-02
## sports                        -5.509370e-02
## tvsports                      -3.856896e-02
## exercise                       2.034173e-02
## dining                         2.033210e-02
## museums                       -1.466183e-02
## art                            8.821651e-02
## hiking                         1.696084e-03
## gaming                         2.751251e-02
## clubbing                       1.676180e-02
## reading                        6.504012e-02
## tv                             4.479387e-02
## theater                       -6.213165e-02
## movies                        -5.476730e-02
## concerts                       5.932666e-02
## music                         -6.959913e-02
## shopping                      -8.596744e-02
## yoga                           1.128897e-02
## interests_correlate            1.591050e-01
## expected_happy_with_sd_people  6.024719e-02
## like                           2.837488e-01
## guess_prob_liked               1.469019e-01
## met                            8.580355e-02
```

```r
tail(modelo_ridge$dev.ratio, 1)
```

```
## [1] 0.3640352
```

```r
#tiene un menor r^2 por lo que tomaremos en cuenta el conjunto de variables más importantes en lasso
```

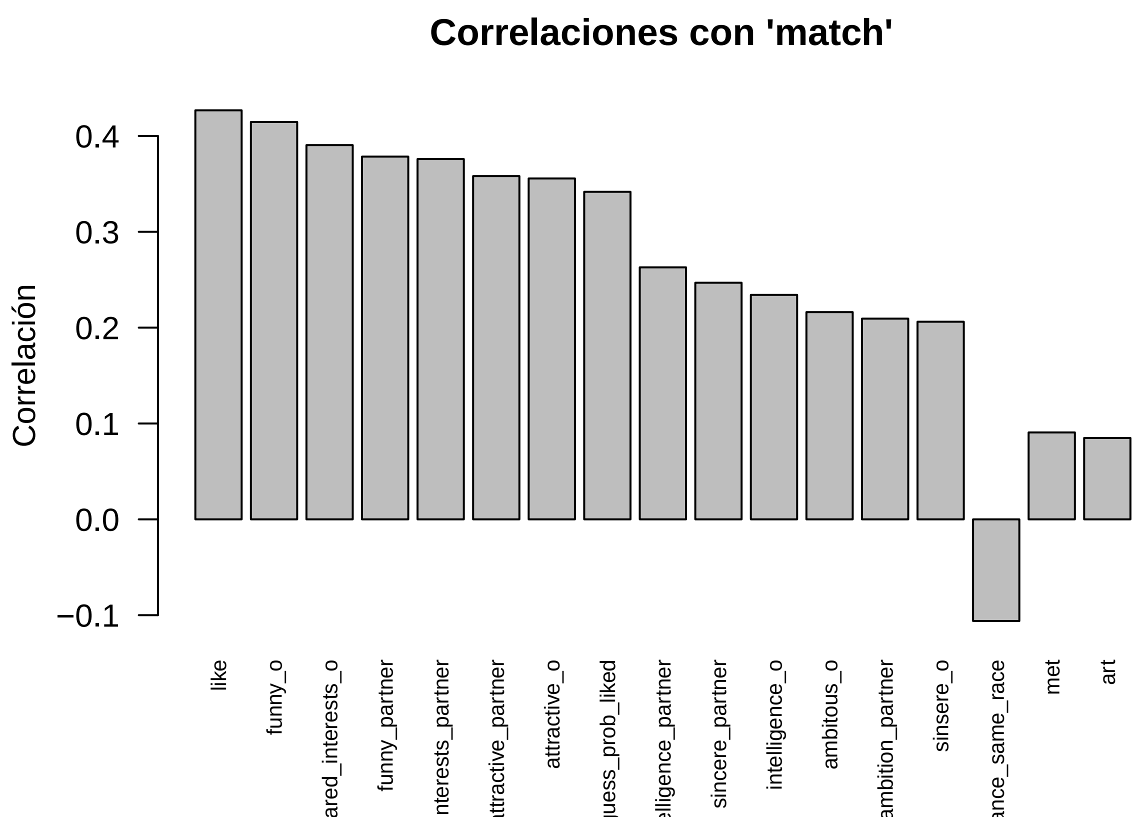
Método mediante umbral de correlaciones

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
correlaciones <- cor(datos_final, method = "pearson")
```

```r
# Seleccionar las variables con correlación mayor a 0.075
variables_seleccionadas <- correlaciones["match", ][abs(correlaciones["match", ]) > 0.075]
orden <- order(abs(variables_seleccionadas), decreasing = TRUE)
variables_seleccionadas <- variables_seleccionadas[orden]
variables_corr <- names(variables_seleccionadas)[-1] #18 variables con mas correlacion

# Crear un gráfico de barras para las variables seleccionadas
barplot(as.vector(variables_seleccionadas[-1]),
        names.arg = names(variables_seleccionadas[-1]),
        main = "Correlaciones con 'match'",
        ylab = "Correlación",
        cex.names = 0.7,
        las = 2)
```



Recopilamos métricas según conjunto de datos utilizado

```r
datos_fuerza <- datos_final_sc[,variables_fuerza]
datos_fuerza <- cbind(datos_fuerza, datos_final_sc['match'])
modelo_fuerza_ <- lm(match ~ ., data = datos_fuerza)
modelo_fuerza_aic <- AIC(modelo_fuerza_)
modelo_fuerza_bic <- BIC(modelo_fuerza_)
r2_modelo_fuerza <- summary(modelo_fuerza_)$adj.r


datos_lasso <- cbind(datos_final_sc[variables_lasso], datos_final_sc['match'])
```

```r
modelo_lasso_ <- lm(match ~., data = datos_lasso)
modelo_lasso_aic <- AIC(modelo_lasso_)
modelo_lasso_bic <- BIC(modelo_lasso_)
r2_modelo_lasso <- summary(modelo_lasso_)$adj.r

datos_corr <- cbind(datos_final_sc[variables_corr], datos_final_sc['match'])
modelo_corr_ <- lm(match~., data = datos_corr)
modelo_corr_aic <- AIC(modelo_corr_)
modelo_corr_bic <- BIC(modelo_corr_)
r2_modelo_corr <- summary(modelo_corr_)$adj.r


AIC_s <- c(modelo_aic, modelo2_aic, modelo3_aic, modelo_fuerza_aic, modelo_lasso_aic, modelo_corr_aic)
which.min(AIC_s)
```

```
## [1] 2
```

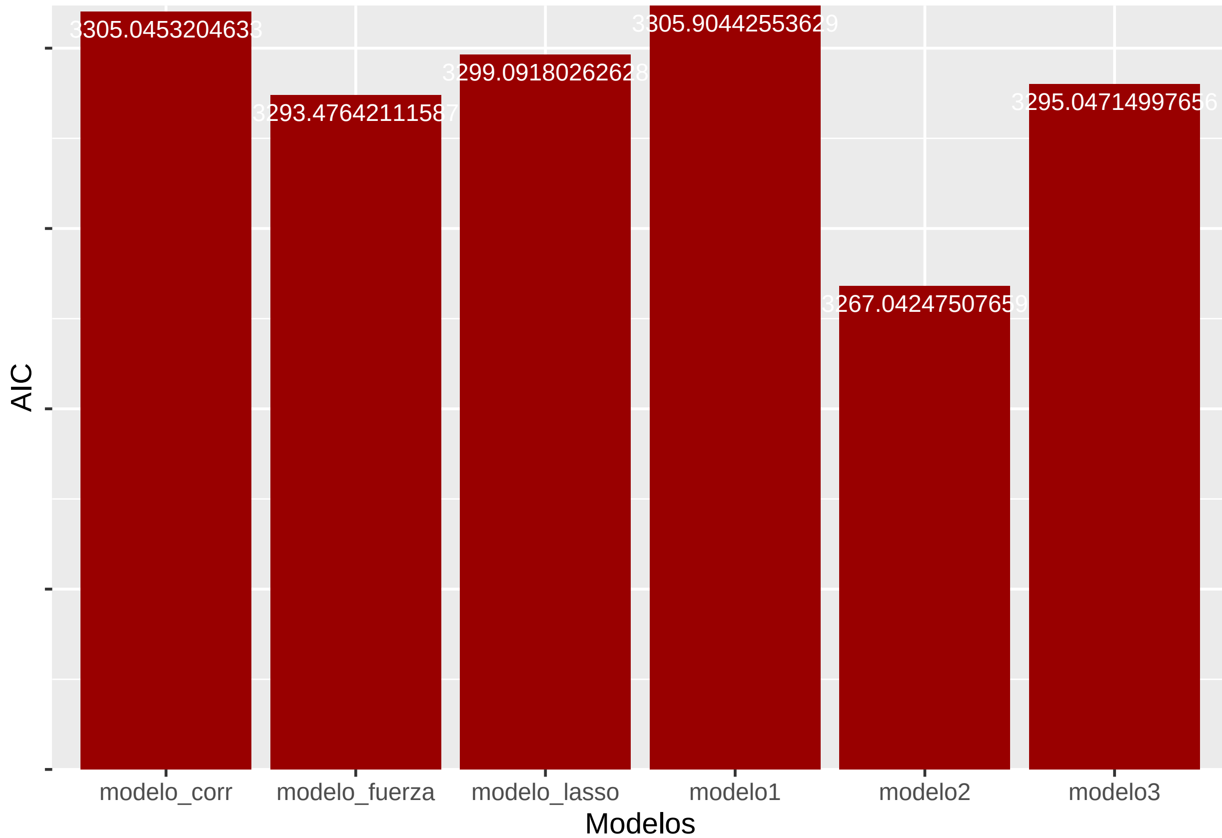
```r
BIC_s <- c(modelo_bic, modelo2_bic, modelo3_bic, modelo_fuerza_bic, modelo_lasso_bic, modelo_corr_bic)
which.min(BIC_s)
```

```
## [1] 4
```

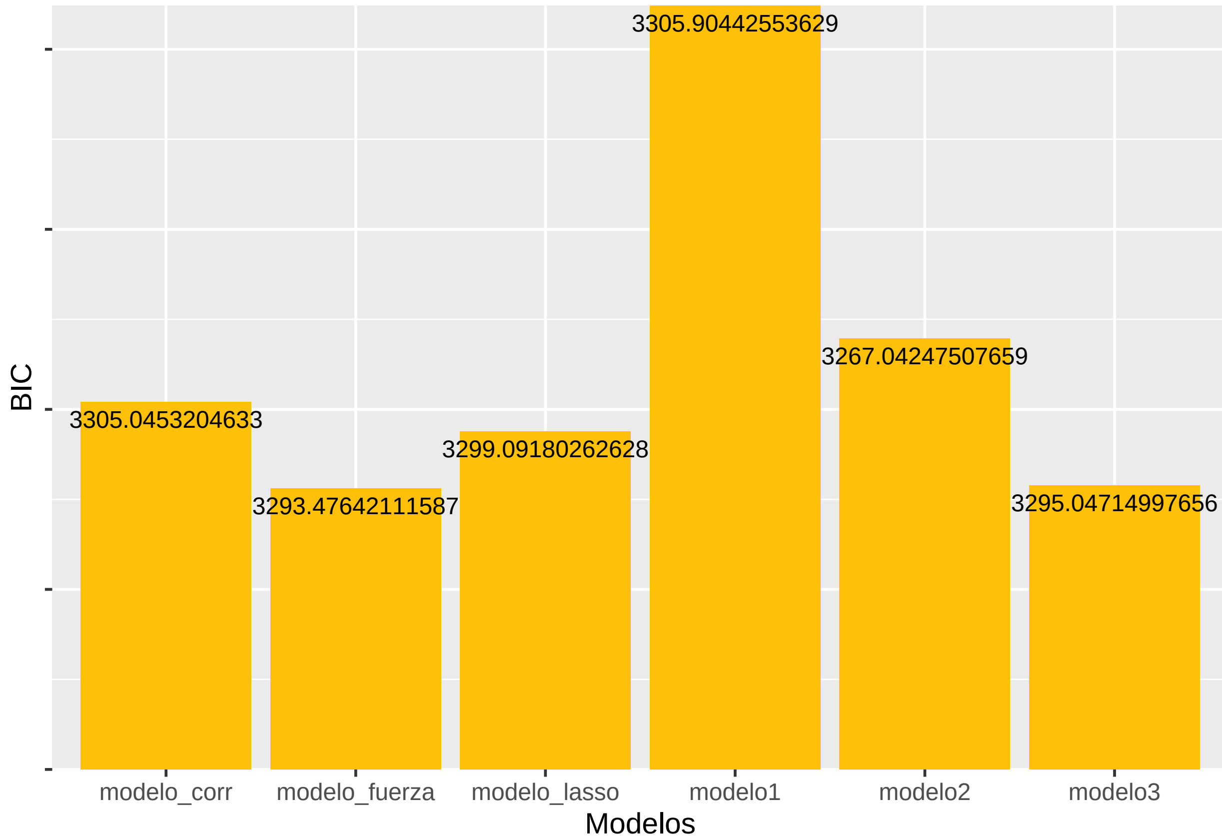
```r
R2 <- c(r2_modelo1, r2_modelo2, r2_modelo3, r2_modelo_fuerza, r2_modelo_lasso, r2_modelo_corr)


nmodelos <- c("modelo1", "modelo2", "modelo3", "modelo_fuerza", "modelo_lasso", "modelo_corr")
numero_vars <- c(length(variables_modelo1), length(variables_modelo2),
                 length(variables_modelo3), length(variables_fuerza), length(variables_lasso),
                 length(variables_corr))
comparacion <- data.frame(nmodelos, numero_vars, AIC_s, BIC_s, R2)
comparacion
```

```
##        nmodelos numero_vars    AIC_s    BIC_s        R2
## 1       modelo1          59 3305.904 3624.258 0.3690530
## 2       modelo2          31 3267.042 3439.266 0.3745020
## 3       modelo3          10 3295.047 3357.674 0.3517979
## 4 modelo_fuerza          10 3293.476 3356.103 0.3525434
## 5  modelo_lasso          15 3299.092 3387.813 0.3522277
## 6   modelo_corr          17 3305.045 3404.205 0.3503368
```

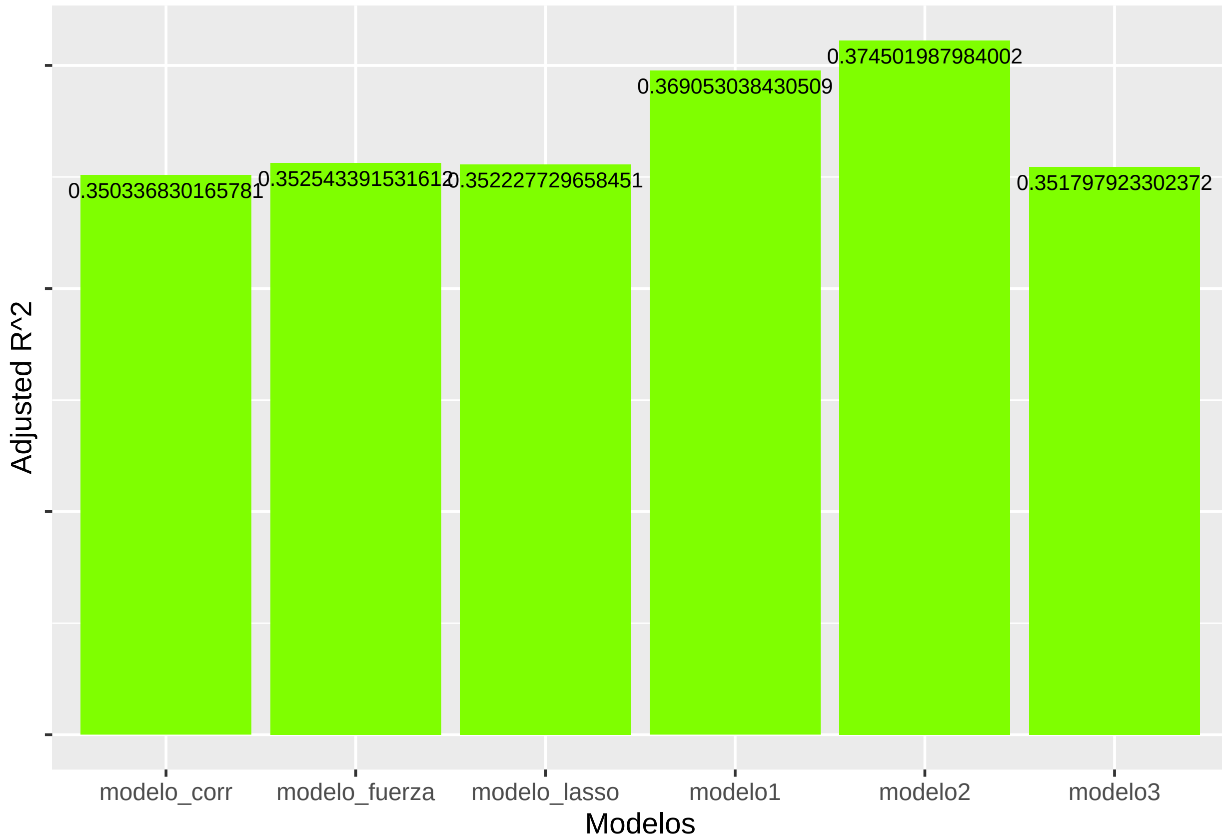
```r
library(ggplot2)
ggplot(comparacion, aes(x = nmodelos, y = AIC_s-3200)) +
  geom_bar(stat = 'identity', fill = '#990000') +
  labs(x = 'Modelos', y = 'AIC') +
  scale_y_continuous(expand = c(0,0)) +
  geom_text(aes(label = AIC_s), size = 3, vjust = 1.5,  color = 'white') +
  theme(axis.text.y = element_blank())
```




```r
ggplot(comparacion, aes(x = nmodelos, y = BIC_s-3200)) +
  geom_bar(stat = 'identity', fill = '#FFC107') +
  labs(x = 'Modelos', y = 'BIC') +
  scale_y_continuous(expand = c(0,0)) +
  geom_text(aes(label = AIC_s), size = 3, vjust = 1.5) +
  theme(axis.text.y = element_blank())
```




```
ggplot(comparacion, aes(x = nmodelos, y = R2-0.25)) +
  geom_bar(stat = 'identity', fill = '#7FFF00') +
  labs(x = 'Modelos', y = 'Adjusted R^2') +
  geom_text(aes(label = R2), size = 2.75, vjust = 1.5) +
  theme(axis.text.y = element_blank())
```



Según estás gráficas el mejor modelo en cuanto a AIC y a R^2 se refiere es el modelo 2 (usando stepwise de ambas direcciones), en cuanto a BIC es el modelo por regresión de subsets.

Regresión logística y cálculo de predicciones con la partición de test.

```
modelo2_log <- glm(match ~., data = as.data.frame(apply(cbind(datos_final_sc[variables_modelo2], match_t
summary(modelo2_log)
```

```
##
## Call:
## glm(formula = match ~ ., family = "binomial", data = as.data.frame(apply(cbind(datos_final_sc[variab
##     match_train), 2, function(x) (x - min(x))/(max(x) - min(x)))))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.62613  -0.67381   0.08969   0.70944   2.86313
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -11.8713     1.1564 -10.266  < 2e-16 ***
## age_o                            0.8560     0.5484   1.561 0.118521
## samerace                        -0.3581     0.1512  -2.368 0.017899 *
## importance_same_race            -0.9653     0.3141  -3.074 0.002114 **
## importance_same_religion         0.5601     0.2792   2.006 0.044832 *
```

```
## pref_o_attractive              2.3232     0.8131   2.857 0.004273 **
## pref_o_intelligence            2.8000     0.6757   4.144 3.42e-05 ***
## pref_o_funny                   2.5468     0.7071   3.602 0.000316 ***
## pref_o_shared_interests        0.9249     0.4482   2.063 0.039070 *
## attractive_o                   2.7661     0.5203   5.316 1.06e-07 ***
## sinsere_o                     -0.7099     0.5260  -1.350 0.177166
## funny_o                        3.5371     0.5766   6.134 8.56e-10 ***
## shared_interests_o             1.7304     0.4469   3.872 0.000108 ***
## attractive_important          -1.3132     0.7329  -1.792 0.073141 .
## sincere_important             -0.9815     0.5470  -1.794 0.072776 .
## intelligence                  -0.9858     0.4489  -2.196 0.028087 *
## attractive_partner             2.5492     0.5605   4.548 5.42e-06 ***
## funny_partner                  1.5889     0.6264   2.537 0.011194 *
## ambition_partner              -1.4931     0.5111  -2.921 0.003484 **
## shared_interests_partner       0.7859     0.5120   1.535 0.124795
## sports                        -0.4734     0.3237  -1.462 0.143632
## tvsports                      -0.4475     0.2979  -1.502 0.133020
## art                            1.0691     0.3740   2.859 0.004255 **
## reading                        0.9999     0.4956   2.018 0.043641 *
## tv                             0.6599     0.3381   1.952 0.050976 .
## theater                       -1.0106     0.3956  -2.555 0.010631 *
## concerts                       0.9708     0.4534   2.141 0.032258 *
## music                         -1.0938     0.5097  -2.146 0.031865 *
## shopping                      -0.8272     0.3113  -2.657 0.007877 **
## expected_happy_with_sd_people  0.6749     0.3952   1.707 0.087738 .
## like                           3.3717     0.6564   5.137 2.79e-07 ***
## guess_prob_liked               1.7441     0.4452   3.918 8.94e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1892.2  on 1364  degrees of freedom
## Residual deviance: 1204.6  on 1333  degrees of freedom
## AIC: 1268.6
##
## Number of Fisher Scoring iterations: 5
```

```
summary(modelo2_log)
```

```
##
## Call:
## glm(formula = match ~ ., family = "binomial", data = as.data.frame(apply(cbind(datos_final_sc[variab
##     match_train), 2, function(x) (x - min(x))/(max(x) - min(x)))))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.62613  -0.67381   0.08969   0.70944   2.86313
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -11.8713     1.1564 -10.266  < 2e-16 ***
## age_o                         0.8560     0.5484   1.561 0.118521
## samerace                     -0.3581     0.1512  -2.368 0.017899 *
## importance_same_race         -0.9653     0.3141  -3.074 0.002114 **
```

```
## importance_same_religion        0.5601     0.2792   2.006 0.044832 *
## pref_o_attractive               2.3232     0.8131   2.857 0.004273 **
## pref_o_intelligence             2.8000     0.6757   4.144 3.42e-05 ***
## pref_o_funny                    2.5468     0.7071   3.602 0.000316 ***
## pref_o_shared_interests         0.9249     0.4482   2.063 0.039070 *
## attractive_o                    2.7661     0.5203   5.316 1.06e-07 ***
## sinsere_o                      -0.7099     0.5260  -1.350 0.177166
## funny_o                         3.5371     0.5766   6.134 8.56e-10 ***
## shared_interests_o              1.7304     0.4469   3.872 0.000108 ***
## attractive_important           -1.3132     0.7329  -1.792 0.073141 .
## sincere_important              -0.9815     0.5470  -1.794 0.072776 .
## intelligence                   -0.9858     0.4489  -2.196 0.028087 *
## attractive_partner              2.5492     0.5605   4.548 5.42e-06 ***
## funny_partner                   1.5889     0.6264   2.537 0.011194 *
## ambition_partner               -1.4931     0.5111  -2.921 0.003484 **
## shared_interests_partner        0.7859     0.5120   1.535 0.124795
## sports                         -0.4734     0.3237  -1.462 0.143632
## tvsports                       -0.4475     0.2979  -1.502 0.133020
## art                             1.0691     0.3740   2.859 0.004255 **
## reading                         0.9999     0.4956   2.018 0.043641 *
## tv                              0.6599     0.3381   1.952 0.050976 .
## theater                        -1.0106     0.3956  -2.555 0.010631 *
## concerts                        0.9708     0.4534   2.141 0.032258 *
## music                          -1.0938     0.5097  -2.146 0.031865 *
## shopping                       -0.8272     0.3113  -2.657 0.007877 **
## expected_happy_with_sd_people   0.6749     0.3952   1.707 0.087738 .
## like                            3.3717     0.6564   5.137 2.79e-07 ***
## guess_prob_liked                1.7441     0.4452   3.918 8.94e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1892.2  on 1364  degrees of freedom
## Residual deviance: 1204.6  on 1333  degrees of freedom
## AIC: 1268.6
##
## Number of Fisher Scoring iterations: 5
```

```r
library('DescTools')
```

```
## Warning: package 'DescTools' was built under R version 4.2.3
```

```r
efron_model2 <- PseudoR2(modelo2_log, "Efron")
nagel_model2 <- PseudoR2(modelo2_log, "Nagelkerke")
aic_model2 <-AIC(modelo2_log)
```

Predicciones modelo 2 (31 variables por metodo stepwise ambas direcciones)

```r
library('pROC')
```

```
## Warning: package 'pROC' was built under R version 4.2.3
```

```
## Type 'citation("pROC")' for a citation.
```

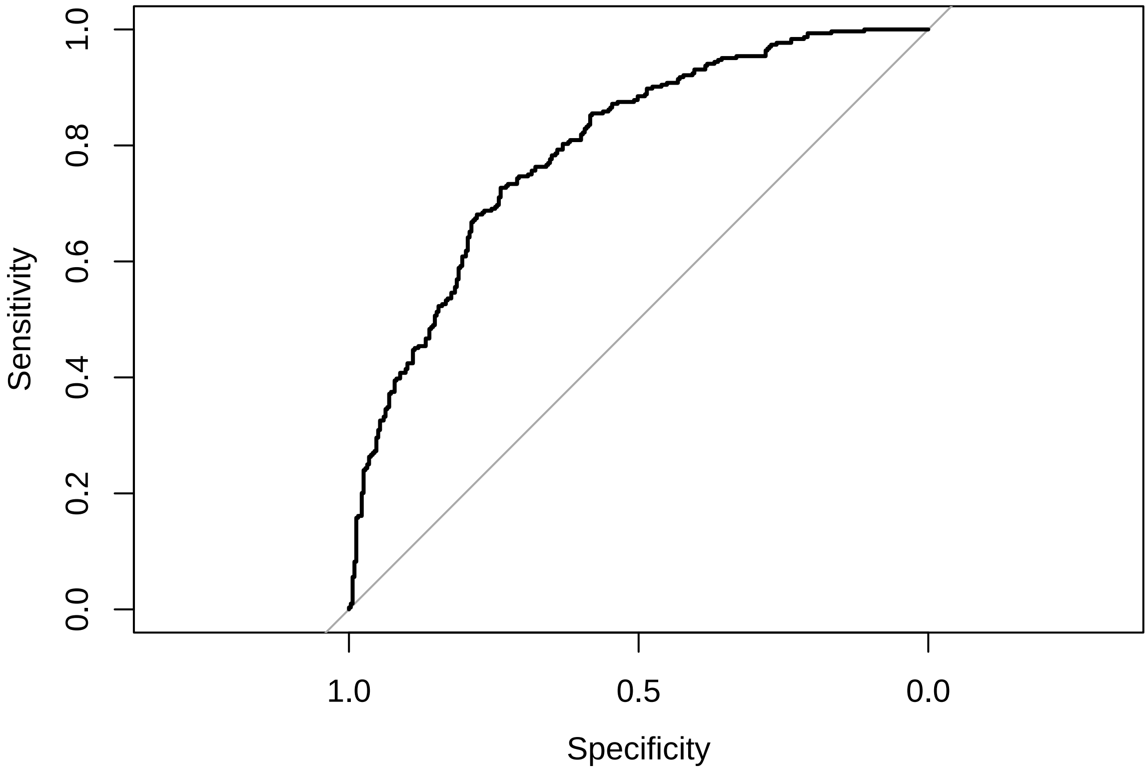
```
##
```

```
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
predicciones_model2 = predict(modelo2_log, newdata = as.data.frame(apply(test[variables_modelo2], 2, fun
tabla_model2 <- table(match_reales_test, ifelse(predicciones_model2>= 0.5,1,0))
curva_roc_modelo2 <- roc(match_reales_test, predicciones_model2)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
plot(curva_roc_modelo2)
```




```
porcentaje_aciertos_model2 <- sum(diag(tabla_model2))/sum(tabla_model2)*100
porcentaje_aciertos_model2
```

```
## [1] 70.20934
```

```
predicciones_model2 = ifelse(predict(modelo2_log, newdata = as.data.frame(apply(test[variables_modelo2]
```

```
data_model1 = as.data.frame(apply(datos_final_sc, 2, function(x) (x - min(x)) / (max(x) - min(x))))
modelo1_log <- glm(match ~., data = data_model1, family = 'binomial' )
summary(modelo1_log)
```

```
##
## Call:
## glm(formula = match ~ ., family = "binomial", data = data_model1)
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.57730  -0.67478   0.07446   0.71423   2.91936
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -10.809694   5.462212  -1.979 0.047816 *
## age                            0.041474   0.934270   0.044 0.964592
## wave                          -0.006084   0.265239  -0.023 0.981699
## gender                        -0.156221   0.217693  -0.718 0.472991
## age_o                          0.853462   0.599579   1.423 0.154610
## d_age                         -0.653435   0.678205  -0.963 0.335308
## samerace                      -0.328329   0.155515  -2.111 0.034752 *
## importance_same_race          -0.956962   0.336777  -2.842 0.004490 **
## importance_same_religion       0.519462   0.293151   1.772 0.076396 .
## pref_o_attractive              3.955732   3.907243   1.012 0.311342
## pref_o_sincere                 1.161362   1.873923   0.620 0.535423
## pref_o_intelligence            3.525585   2.014046   1.750 0.080032 .
## pref_o_funny                   3.283991   1.996955   1.644 0.100073
## pref_o_ambitious               0.308255   1.212470   0.254 0.799312
## pref_o_shared_interests        1.489224   1.188144   1.253 0.210059
## attractive_o                   2.724717   0.543556   5.013 5.37e-07 ***
## sinsere_o                     -0.934334   0.638967  -1.462 0.143671
## intelligence_o                 0.744807   0.797585   0.934 0.350393
## funny_o                        3.429027   0.628291   5.458 4.82e-08 ***
## ambitous_o                    -0.320241   0.597136  -0.536 0.591755
## shared_interests_o             1.814598   0.472331   3.842 0.000122 ***
## attractive_important          -4.285799   3.606292  -1.188 0.234667
## sincere_important             -2.265961   1.738904  -1.303 0.192541
## intellicence_important        -1.786623   1.871209  -0.955 0.339681
## funny_important               -0.808172   1.828876  -0.442 0.658565
## ambtition_important           -1.771733   1.969593  -0.900 0.368364
## shared_interests_important    -0.824057   1.088643  -0.757 0.449075
## attractive                    -0.016176   0.601508  -0.027 0.978546
## sincere                        0.248297   0.505460   0.491 0.623264
## intelligence                  -1.147085   0.551584  -2.080 0.037561 *
## funny                         -0.777839   0.621570  -1.251 0.210785
## ambition                       0.300199   0.427334   0.702 0.482372
## attractive_partner             2.663660   0.579425   4.597 4.28e-06 ***
## sincere_partner               -0.663606   0.674324  -0.984 0.325064
## intelligence_partner           1.262473   0.763177   1.654 0.098080 .
## funny_partner                  1.681929   0.657799   2.557 0.010561 *
## ambition_partner              -1.991370   0.595777  -3.342 0.000830 ***
## shared_interests_partner       0.652321   0.531769   1.227 0.219935
## sports                        -0.699995   0.370709  -1.888 0.058991 .
## tvsports                      -0.464956   0.328031  -1.417 0.156362
## exercise                       0.342169   0.326576   1.048 0.294756
## dining                         0.378297   0.499015   0.758 0.448399
## museums                       -0.594867   0.724155  -0.821 0.411382
## art                            1.478108   0.626080   2.361 0.018231 *
## hiking                        -0.043755   0.338642  -0.129 0.897193
## gaming                         0.456207   0.474046   0.962 0.335864
## clubbing                       0.151737   0.289001   0.525 0.599555
```

```
## reading                          1.168627   0.523895   2.231 0.025704 *
## tv                               0.659229   0.371553   1.774 0.076021 .
## theater                         -0.980721   0.452391  -2.168 0.030169 *
## movies                          -0.561781   0.453528  -1.239 0.215460
## concerts                         1.104099   0.492969   2.240 0.025111 *
## music                           -1.240954   0.555164  -2.235 0.025398 *
## shopping                        -1.008561   0.357327  -2.823 0.004765 **
## yoga                             0.027573   0.317106   0.087 0.930709
## interests_correlate              0.307984   0.466257   0.661 0.508903
## expected_happy_with_sd_people    0.752156   0.428048   1.757 0.078887 .
## like                             3.546674   0.676519   5.243 1.58e-07 ***
## guess_prob_liked                 1.703197   0.466736   3.649 0.000263 ***
## met                              0.807866   1.895806   0.426 0.670011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1892.2  on 1364  degrees of freedom
## Residual deviance: 1186.4  on 1305  degrees of freedom
## AIC: 1306.4
##
## Number of Fisher Scoring iterations: 5
```

```r
efron_model1 <- PseudoR2(modelo1_log, "Efron")
nagel_model1 <- PseudoR2(modelo1_log, "Nagelkerke")
aic_model1 <- AIC(modelo1_log)
```

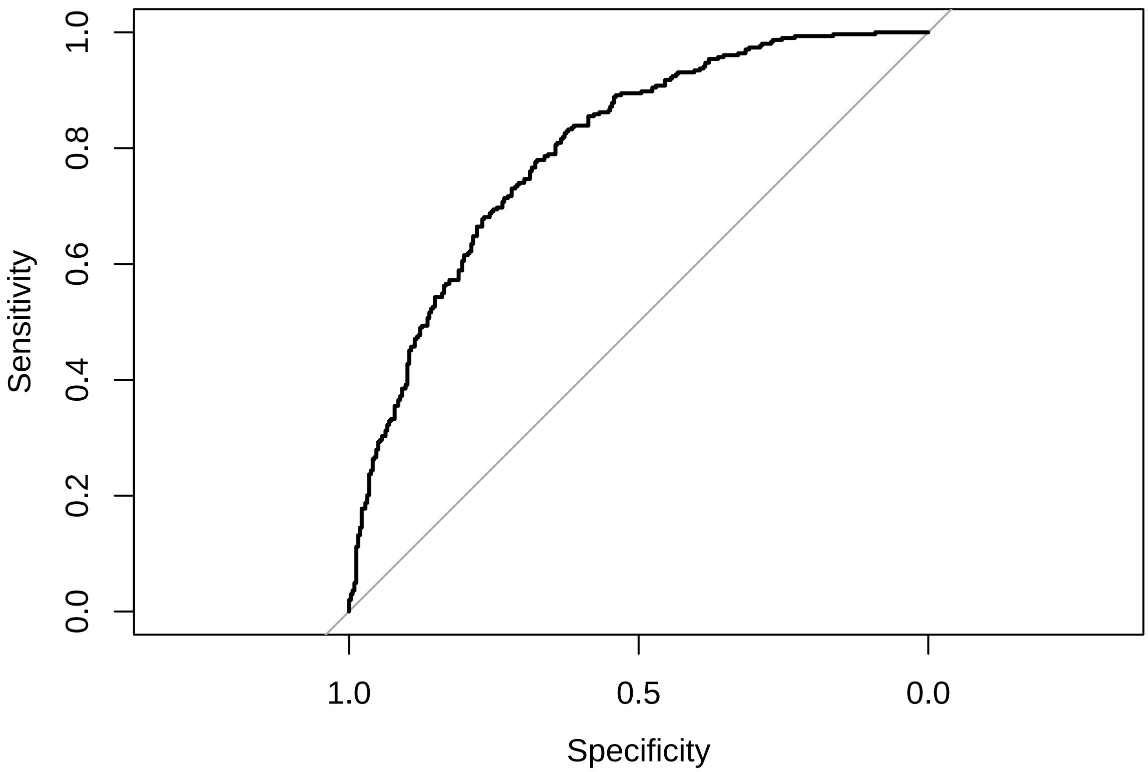
Predicciones modelo 1 (todas las variables)

```r
predicciones_model1 = predict(modelo1_log, newdata = as.data.frame(apply(test, 2, function(x) (x - min(
tabla_model1 <- table(match_reales_test, ifelse(predicciones_model1>=0.5,1,0))
curva_roc_modelo1 <- roc(match_reales_test, predicciones_model1)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(curva_roc_modelo1)
```




```
porcentaje_aciertos_model1 <- sum(diag(tabla_model1))/sum(tabla_model1)*100
porcentaje_aciertos_model1
```

```
## [1] 68.438
```

```
modelo_lasso_log <- glm(match ~., data = as.data.frame(apply(datos_lasso, 2, function(x) (x - min(x)) /
summary(modelo_lasso_log)
```

```
##
## Call:
## glm(formula = match ~ ., family = "binomial", data = as.data.frame(apply(datos_lasso,
##     2, function(x) (x - min(x))/(max(x) - min(x)))))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7256  -0.7516   0.1055   0.7406   2.6241
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -9.1823     0.6972 -13.171  < 2e-16 ***
## like                    3.2519     0.6166   5.274 1.33e-07 ***
## funny_o                 3.0000     0.5363   5.594 2.22e-08 ***
## samerace               -0.3227     0.1437  -2.245  0.02474 *
## attractive_o            2.3318     0.4932   4.728 2.27e-06 ***
## attractive_partner      2.7533     0.5354   5.143 2.71e-07 ***
## ambition_partner       -1.6685     0.5443  -3.065  0.00217 **
## shared_interests_o      1.7319     0.4182   4.141 3.45e-05 ***
```

```
## interests_correlate    0.6127     0.4043   1.515  0.12969
## guess_prob_liked       1.8747     0.4145   4.522 6.11e-06 ***
## gender                -0.2980     0.1502  -1.985  0.04715 *
## funny_partner          1.3193     0.5968   2.211  0.02705 *
## art                    0.6662     0.3133   2.127  0.03346 *
## intelligence          -1.2617     0.4113  -3.067  0.00216 **
## intelligence_partner   0.7465     0.6221   1.200  0.23015
## music                 -0.5368     0.3730  -1.439  0.15004
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1892.2  on 1364  degrees of freedom
## Residual deviance: 1272.2  on 1349  degrees of freedom
## AIC: 1304.2
##
## Number of Fisher Scoring iterations: 5
```

```r
efron_model_lasso <- PseudoR2(modelo_lasso_log, "Efron")
nagel_model_lasso <- PseudoR2(modelo_lasso_log, "Nagelkerke")
aic_model_lasso <- AIC(modelo_lasso_log)
```

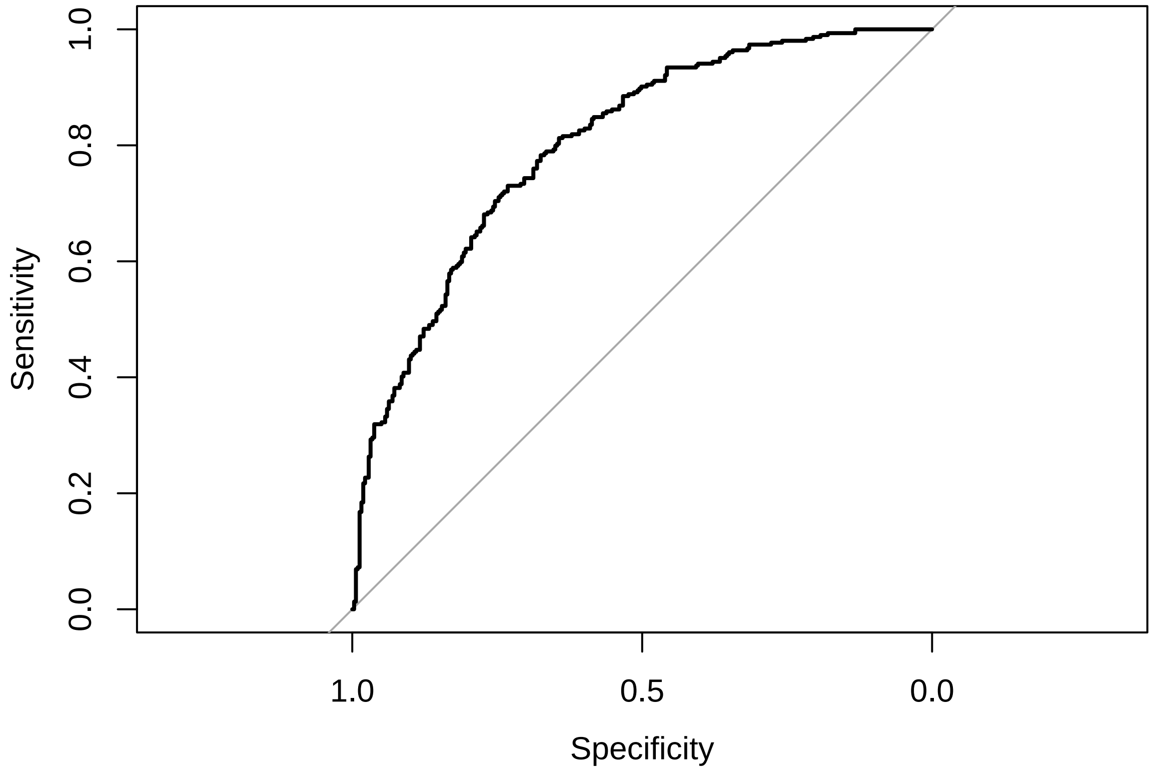
Predicciones modelo lasso (15 variables)

```r
predicciones_model_lasso = predict(modelo_lasso_log, newdata = as.data.frame(apply(test[variables_lasso]
tabla_model_lasso <- table(match_reales_test, ifelse(predicciones_model_lasso>=0.5, 1,0))
curva_roc_model_lasso <- roc(match_reales_test, predicciones_model_lasso)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(curva_roc_model_lasso)
```




```
porcentaje_aciertos_model_lasso <- sum(diag(tabla_model_lasso))/sum(tabla_model_lasso)*100
porcentaje_aciertos_model_lasso
```

```
## [1] 71.65862
```

```
modelo_corr_log <- glm(match ~., data = as.data.frame(apply(datos_corr, 2, function(x) (x - min(x)) / (m
summary(modelo_corr_log)
```

```
##
## Call:
## glm(formula = match ~ ., family = "binomial", data = as.data.frame(apply(datos_corr,
##     2, function(x) (x - min(x))/(max(x) - min(x)))))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6966  -0.7354   0.1153   0.7603   2.8171
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -9.1613     0.6840 -13.395  < 2e-16 ***
## like                        3.2814     0.6257   5.244 1.57e-07 ***
## funny_o                     3.1471     0.5801   5.425 5.79e-08 ***
## shared_interests_o          1.8738     0.4328   4.329 1.50e-05 ***
## funny_partner               1.1139     0.6131   1.817 0.069247 .
## shared_interests_partner    0.7457     0.4855   1.536 0.124533
## attractive_partner          2.5571     0.5267   4.855 1.20e-06 ***
## attractive_o                2.4248     0.4929   4.920 8.66e-07 ***
```

```
## guess_prob_liked           1.2016     0.4158   2.890 0.003852 **
## intelligence_partner       1.2578     0.6999   1.797 0.072316 .
## sincere_partner           -0.9080     0.6289  -1.444 0.148788
## intelligence_o             0.4502     0.7436   0.605 0.544893
## ambitous_o                -0.8644     0.5487  -1.575 0.115213
## ambition_partner          -1.8121     0.5437  -3.333 0.000859 ***
## sinsere_o                 -0.6058     0.5915  -1.024 0.305740
## importance_same_race      -0.7031     0.2460  -2.858 0.004261 **
## met                        0.3845     1.5458   0.249 0.803561
## art                        0.7289     0.2899   2.514 0.011922 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1892.2  on 1364  degrees of freedom
## Residual deviance: 1278.1  on 1347  degrees of freedom
## AIC: 1314.1
##
## Number of Fisher Scoring iterations: 5
```

```
efron_model_corr <- PseudoR2(modelo_corr_log, "Efron")
nagel_model_corr <- PseudoR2(modelo_corr_log, "Nagelkerke")
aic_model_corr <- AIC(modelo_corr_log)
```

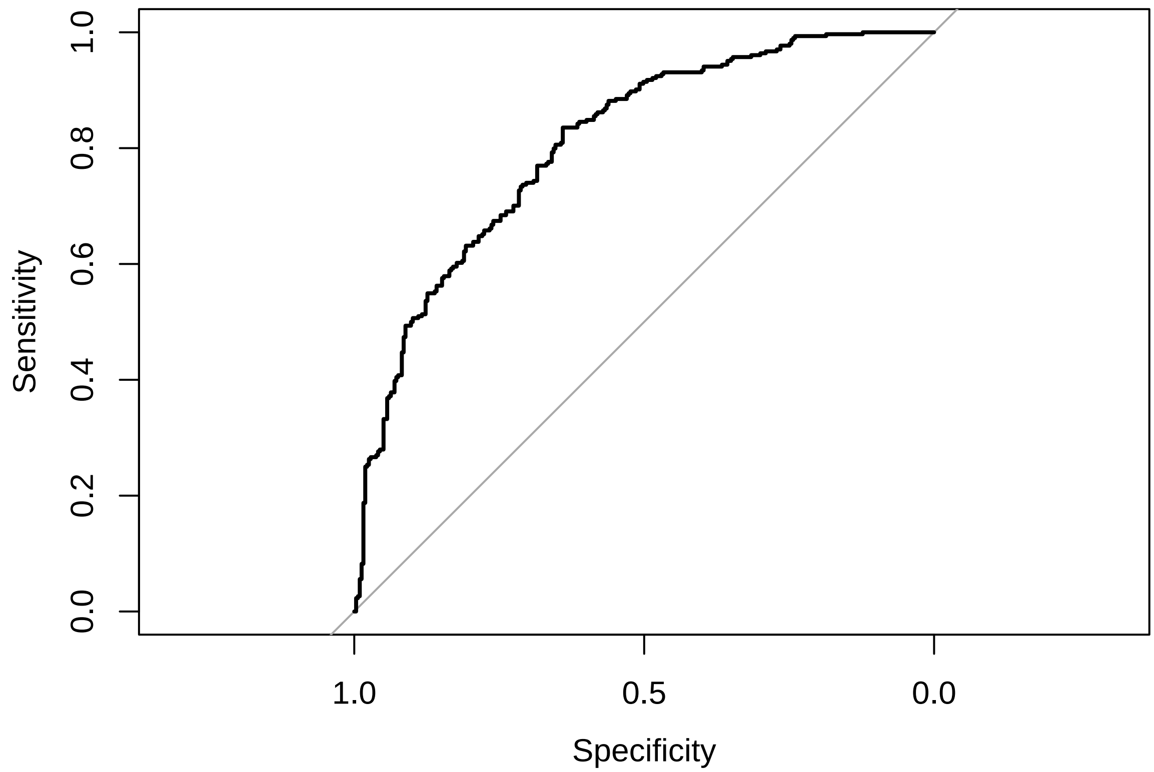
Predicciones modelo correlaciones (17 variables)

```
predicciones_model_corr = predict(modelo_corr_log, newdata = as.data.frame(apply(test[variables_corr],
tabla_model_corr <- table(match_reales_test, ifelse(predicciones_model_corr>=0.5,1,0))
curva_roc_model_corr <- roc(match_reales_test, predicciones_model_corr)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(curva_roc_model_corr)
```




```r
porcentaje_aciertos_model_corr <- sum(diag(tabla_model_corr))/sum(tabla_model_corr)*100
porcentaje_aciertos_model_corr
```

```
## [1] 71.65862
```

```r
modelo_fuerza_log <- glm(match ~., data = as.data.frame(apply(datos_fuerza, 2, function(x) (x - min(x))
summary(modelo_fuerza_log)
```

```
##
## Call:
## glm(formula = match ~ ., family = "binomial", data = as.data.frame(apply(datos_fuerza,
##     2, function(x) (x - min(x))/(max(x) - min(x)))))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8619  -0.7448   0.1085   0.7471   2.5845
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -9.6619     0.6718 -14.383  < 2e-16 ***
## importance_same_race    -0.7231     0.2439  -2.964 0.003035 **
## pref_o_intelligence      1.4413     0.5281   2.729 0.006350 **
## attractive_o             2.2733     0.4849   4.688 2.75e-06 ***
## funny_o                  2.9128     0.5265   5.533 3.15e-08 ***
## shared_interests_o       1.8786     0.4146   4.531 5.88e-06 ***
## intelligence            -1.3244     0.3946  -3.357 0.000788 ***
## attractive_partner       2.8671     0.5149   5.568 2.58e-08 ***
```

```
## art                      0.9348     0.2905   3.218 0.001291 **
## like                     3.4503     0.5459   6.320 2.61e-10 ***
## guess_prob_liked         1.6627     0.4101   4.055 5.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1892.2  on 1364  degrees of freedom
## Residual deviance: 1279.6  on 1354  degrees of freedom
## AIC: 1301.6
##
## Number of Fisher Scoring iterations: 5
```

```r
efron_model_fuerza <- PseudoR2(modelo_fuerza_log, "Efron")
nagel_model_fuerza <- PseudoR2(modelo_fuerza_log, "Nagelkerke")
aic_model_fuerza <- AIC(modelo_fuerza_log)
```

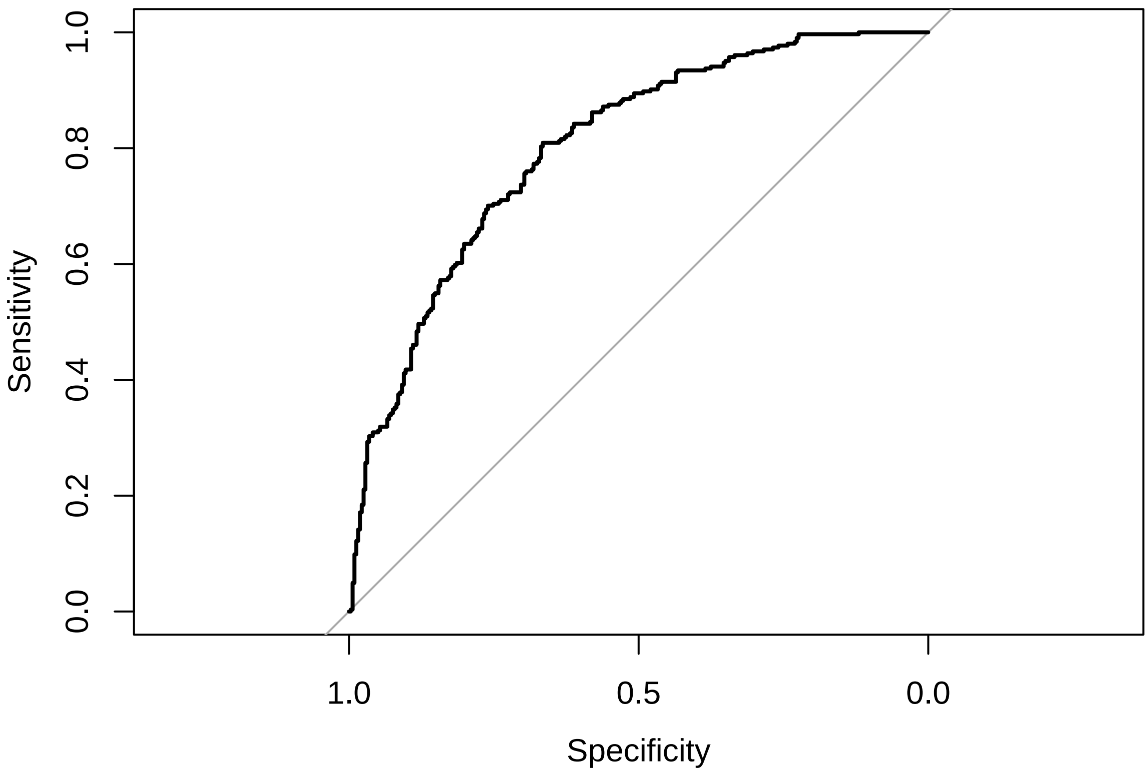
Predicciones modelo regresión de mejores subsets (10 variables)

```r
predicciones_model_fuerza = predict(modelo_fuerza_log, newdata = as.data.frame(apply(test[variables_fue
tabla_model_fuerza <- table(match_reales_test, ifelse(predicciones_model_fuerza >= 0.5,1,0))
curva_roc_model_fuerza <- roc(match_reales_test, predicciones_model_fuerza)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(curva_roc_model_fuerza)
```




```
porcentaje_aciertos_model_fuerza <- sum(diag(tabla_model_fuerza))/sum(tabla_model_fuerza)*100
porcentaje_aciertos_model_fuerza
```

```
## [1] 71.81965
```

Comparación de r^2, aic y precisión de los diferentes modelos de regresión logística

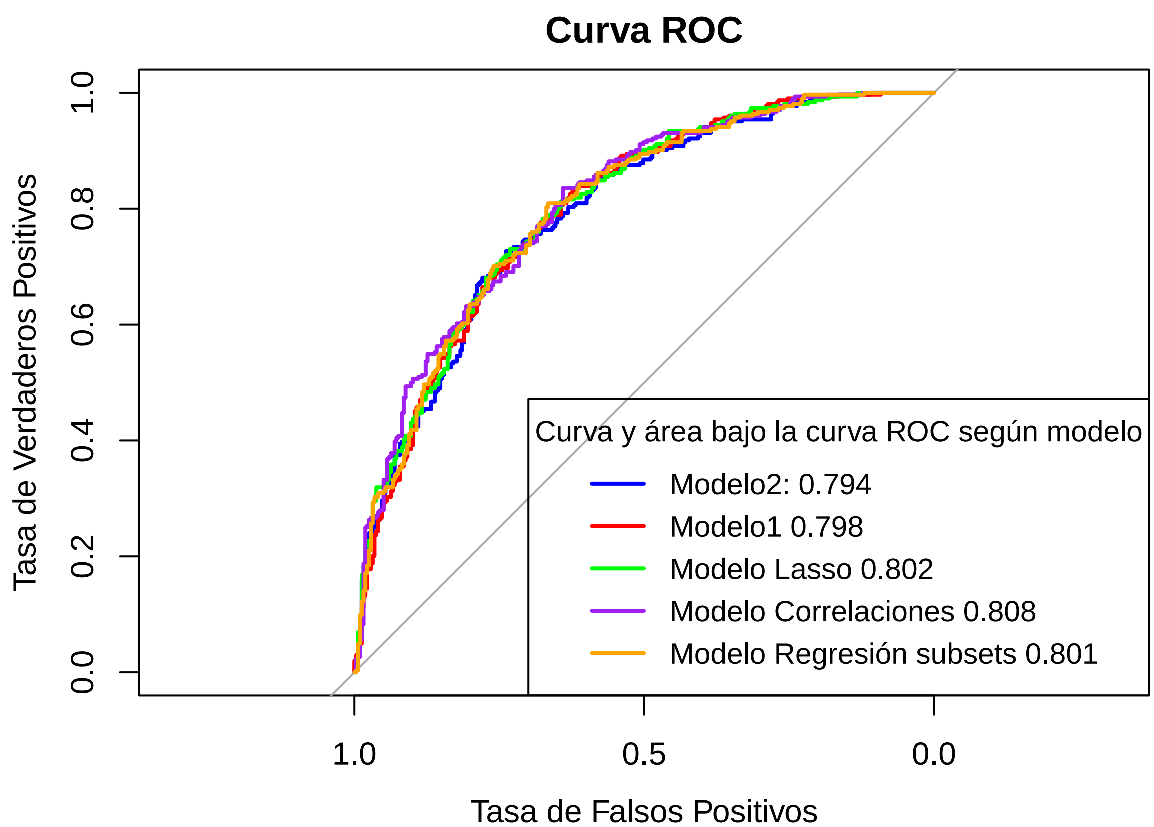
```
precision <- c(porcentaje_aciertos_model2, porcentaje_aciertos_model1, porcentaje_aciertos_model_lasso,

efron <- c(efron_model2, efron_model1, efron_model_lasso, efron_model_corr, efron_model_fuerza)

nagel <- c(nagel_model2, nagel_model1, nagel_model_lasso, nagel_model_corr, nagel_model_fuerza)

aic_log <- c(aic_model2, aic_model1, aic_model_lasso, aic_model_corr, aic_model_fuerza)

nmodelos <- c('modelo2', 'modelo1', 'modelo lasso', 'modelo corr', 'modelo fuerza')

comparacion2 <- data.frame(nmodelos, efron, nagel, aic_log, precision)
comparacion2
```

```
##        nmodelos     efron     nagel  aic_log precision
## 1       modelo2 0.4261223 0.5276441 1268.591  70.20934
## 2       modelo1 0.4353259 0.5382869 1306.441  68.43800
## 3  modelo lasso 0.3895787 0.4867410 1304.184  71.65862
```

```
## 4   modelo corr 0.3820614 0.4830714 1314.088  71.65862
## 5 modelo fuerza 0.3863797 0.4821303 1301.597  71.81965
```

El modelo con mejor porcentaje de predicción es el modelo por lasso, el cual tiene 15 variables y el modelo por correlaciones.

```
plot(curva_roc_modelo2, col = "blue", main = "Curva ROC", xlab = "Tasa de Falsos Positivos", ylab = "Ta
lines(curva_roc_modelo1, col='red', print.auc = TRUE)
lines(curva_roc_model_lasso, col='green')
lines(curva_roc_model_corr, col = 'purple')
lines(curva_roc_model_fuerza, col = 'orange')
legend('bottomright', title = 'Curva y área bajo la curva ROC según modelo',
                                 legend = c(paste('Modelo2:', round(curva_roc_modelo2$auc,3)),
                                 paste('Modelo1', round(curva_roc_modelo1$auc, 3)),
                                 paste('Modelo Lasso', round(curva_roc_model_lasso$auc,3)),
                                 paste('Modelo Correlaciones', round(curva_roc_model_corr$auc, 3)),
                                 paste('Modelo Regresión subsets', round(curva_roc_model_fuerza$auc,3))
                                 col = c('blue', 'red', 'green', 'purple', 'orange'),cex = 0.9, lty=1,
```



Naive Bayes Es conveniente usar este método ya que el número de variables es bastante grande. Vamos a ver su rendimiento con el conjunto de datos más conveniente devuelto por el modelo 1 y el modelo 2 ya que son los que tienen mayor número de variables

```
library('e1071')
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```
modelo_nb <- naiveBayes(match ~.,cbind(datos_final_sc[variables_modelo1], match_train))
modelo_nb
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.4952381 0.5047619
##
## Conditional probabilities:
##    age
## Y            [,1]      [,2]
##   0  0.018018168 1.0291933
##   1 -0.004104341 0.9472378
##
##    wave
## Y           [,1]     [,2]
##   0 0.071756270 1.007986
##   1 0.002252741 1.000121
##
##    gender
## Y           [,1]     [,2]
##   0  0.02051803 1.000092
##   1 -0.01921904 1.000464
##
##    age_o
## Y           [,1]      [,2]
##   0 0.005656487 0.9846191
##   1 0.024332346 0.9894163
##
##    d_age
## Y           [,1]      [,2]
##   0  0.03472640 1.0162134
##   1 -0.04586774 0.9345411
##
##    samerace
## Y           [,1]      [,2]
##   0  0.01496479 1.0032125
##   1 -0.01570866 0.9973653
##
##    importance_same_race
## Y           [,1]      [,2]
##   0  0.09963667 1.0233069
##   1 -0.11214082 0.9624911
##
##    importance_same_religion
## Y           [,1]      [,2]
##   0  0.03549610 0.9996628
##   1 -0.06000594 0.9801814
##
##    pref_o_attractive
## Y           [,1]      [,2]
```

```
##   0 0.003299894 0.9720996
##   1 0.031947103 1.0841447
##
##    pref_o_sincere
## Y           [,1]      [,2]
##   0  0.06052243 0.9983572
##   1 -0.08287144 1.0218582
##
##    pref_o_intelligence
## Y           [,1]      [,2]
##   0 -0.04955318 1.0517008
##   1  0.02849611 0.9946808
##
##    pref_o_funny
## Y           [,1]      [,2]
##   0 -0.06700968 0.9109109
##   1  0.05471344 1.0988989
##
##    pref_o_ambitious
## Y           [,1]     [,2]
##   0 0.026726728 1.000284
##   1 0.004241674 1.009906
##
##    pref_o_shared_interests
## Y           [,1]      [,2]
##   0  0.02193558 0.9567117
##   1 -0.07379913 1.0406235
##
##    attractive_o
## Y          [,1]      [,2]
##   0 -0.3710119 1.0362965
##   1  0.3474071 0.8448677
##
##    sinsere_o
## Y          [,1]     [,2]
##   0 -0.2025192 1.054529
##   1  0.2093797 0.896755
##
##    intelligence_o
## Y          [,1]      [,2]
##   0 -0.2218746 1.0605732
##   1  0.2464745 0.8779985
##
##    funny_o
## Y          [,1]     [,2]
##   0 -0.4290992 1.038916
##   1  0.4172037 0.807388
##
##    ambitous_o
## Y          [,1]      [,2]
##   0 -0.2054513 1.0461479
##   1  0.2328822 0.9323307
##
##    shared_interests_o
```

```
## Y          [,1]      [,2]
##   0 -0.4026299 0.9885114
##   1  0.3919324 0.8841241
##
##    attractive_important
## Y           [,1]      [,2]
##   0  0.02865025 1.0022910
##   1 -0.03793376 0.9997948
##
##    sincere_important
## Y           [,1]      [,2]
##   0  0.01839945 1.0345820
##   1 -0.03012725 0.9922474
##
##    intellicence_important
## Y            [,1]      [,2]
##   0 -0.029531949 1.0655772
##   1  0.008062987 0.9345575
##
##    funny_important
## Y           [,1]     [,2]
##   0 -0.03758024 0.989125
##   1  0.06951185 1.049835
##
##    ambtition_important
## Y           [,1]     [,2]
##   0 -0.03334432 1.062783
##   1  0.02864059 0.960597
##
##    shared_interests_important
## Y           [,1]     [,2]
##   0  0.05312481 1.007678
##   1 -0.02837771 1.010961
##
##    attractive
## Y           [,1]      [,2]
##   0 -0.08103280 1.0279745
##   1  0.02618693 0.9849214
##
##    sincere
## Y            [,1]      [,2]
##   0 -0.013178199 1.0128596
##   1  0.004360183 0.9903578
##
##    intelligence
## Y           [,1]      [,2]
##   0 -0.04420876 1.0335883
##   1  0.05729902 0.9629094
##
##    funny
## Y            [,1]     [,2]
##   0  0.002123418 1.039980
##   1 -0.048804455 1.017353
##
```

```
##    ambition
## Y            [,1]      [,2]
##   0 0.0005246497 1.0023243
##   1 0.0040207198 0.9838518
##
##    attractive_partner
## Y          [,1]      [,2]
##   0 -0.3176257 0.9949156
##   1  0.3730991 0.7980468
##
##    sincere_partner
## Y          [,1]      [,2]
##   0 -0.2462165 1.0286969
##   1  0.2418984 0.8846594
##
##    intelligence_partner
## Y          [,1]      [,2]
##   0 -0.2695298 1.0683129
##   1  0.2587791 0.8625569
##
##    funny_partner
## Y          [,1]      [,2]
##   0 -0.3795056 1.0050684
##   1  0.3703408 0.8226593
##
##    ambition_partner
## Y          [,1]      [,2]
##   0 -0.2069493 1.0104728
##   1  0.2074201 0.9246503
##
##    shared_interests_partner
## Y          [,1]      [,2]
##   0 -0.3684137 0.9587492
##   1  0.3789534 0.8838068
##
##    sports
## Y           [,1]      [,2]
##   0 -0.01626490 0.9985207
##   1 -0.02555759 1.0155141
##
##    tvsports
## Y           [,1]      [,2]
##   0  0.02064614 1.0168545
##   1 -0.02937276 0.9811854
##
##    exercise
## Y           [,1]     [,2]
##   0 -0.06717501 1.017354
##   1 -0.01299769 1.020946
##
##    dining
## Y           [,1]      [,2]
##   0 0.003084476 0.9905666
##   1 0.043088138 0.9982491
```

```
##
##    museums
## Y          [,1]      [,2]
##   0 -0.05056002 0.9764531
##   1  0.06952343 0.9875452
##
##    art
## Y          [,1]      [,2]
##   0 -0.07415212 0.9724649
##   1  0.09294971 0.9893408
##
##    hiking
## Y          [,1]      [,2]
##   0 -0.02126518 0.9766124
##   1  0.03093636 0.9935406
##
##    gaming
## Y          [,1]     [,2]
##   0 -0.01028722 0.983843
##   1  0.01541922 1.004627
##
##    clubbing
## Y          [,1]      [,2]
##   0 -0.06147156 0.9992232
##   1  0.04178317 1.0064914
##
##    reading
## Y          [,1]      [,2]
##   0 -0.06327711 1.0608223
##   1  0.03002764 0.9142332
##
##    tv
## Y           [,1]      [,2]
##   0  0.007956381 1.0051704
##   1 -0.012676160 0.9757215
##
##    theater
## Y           [,1]      [,2]
##   0 -0.006066056 0.9703566
##   1  0.002281742 1.0139289
##
##    movies
## Y          [,1]     [,2]
##   0  0.03467529 1.024398
##   1 -0.05295059 1.005078
##
##    concerts
## Y          [,1]      [,2]
##   0 -0.05482764 0.9927214
##   1  0.04090749 1.0115725
##
##    music
## Y           [,1]      [,2]
##   0 -0.020088518 0.9823813
```

```
##   1 -0.002220439 1.0030257
##
##    shopping
## Y           [,1]      [,2]
##   0  0.03695667 0.9824118
##   1 -0.04509594 0.9856052
##
##    yoga
## Y           [,1]      [,2]
##   0 -0.05461979 0.9768409
##   1  0.07568966 0.9908400
##
##    interests_correlate
## Y           [,1]      [,2]
##   0 -0.06016453 0.9899618
##   1  0.06733859 1.0074652
##
##    expected_happy_with_sd_people
## Y           [,1]      [,2]
##   0 -0.03336271 0.9932720
##   1  0.03786178 0.9914717
##
##    like
## Y          [,1]      [,2]
##   0 -0.4087194 1.0152439
##   1  0.4415018 0.7735341
##
##    guess_prob_liked
## Y          [,1]      [,2]
##   0 -0.3361117 0.9451217
##   1  0.3446245 0.9282603
##
##    met
## Y           [,1]      [,2]
##   0 -0.09851550 0.8527886
##   1  0.05201038 0.8011969
```

```r
preds_nb <- predict(modelo_nb, test[variables_modelo1], type = 'class')
preds_nb2 <- predict(modelo_nb, test[variables_modelo1], type = 'raw')[,1]
```

```r
tabla_model_nb <- table(match_reales_test, preds_nb)
porcentaje_aciertos_model_nb <- sum(diag(tabla_model_nb))/sum(tabla_model_nb)*100
porcentaje_aciertos_model_nb
```

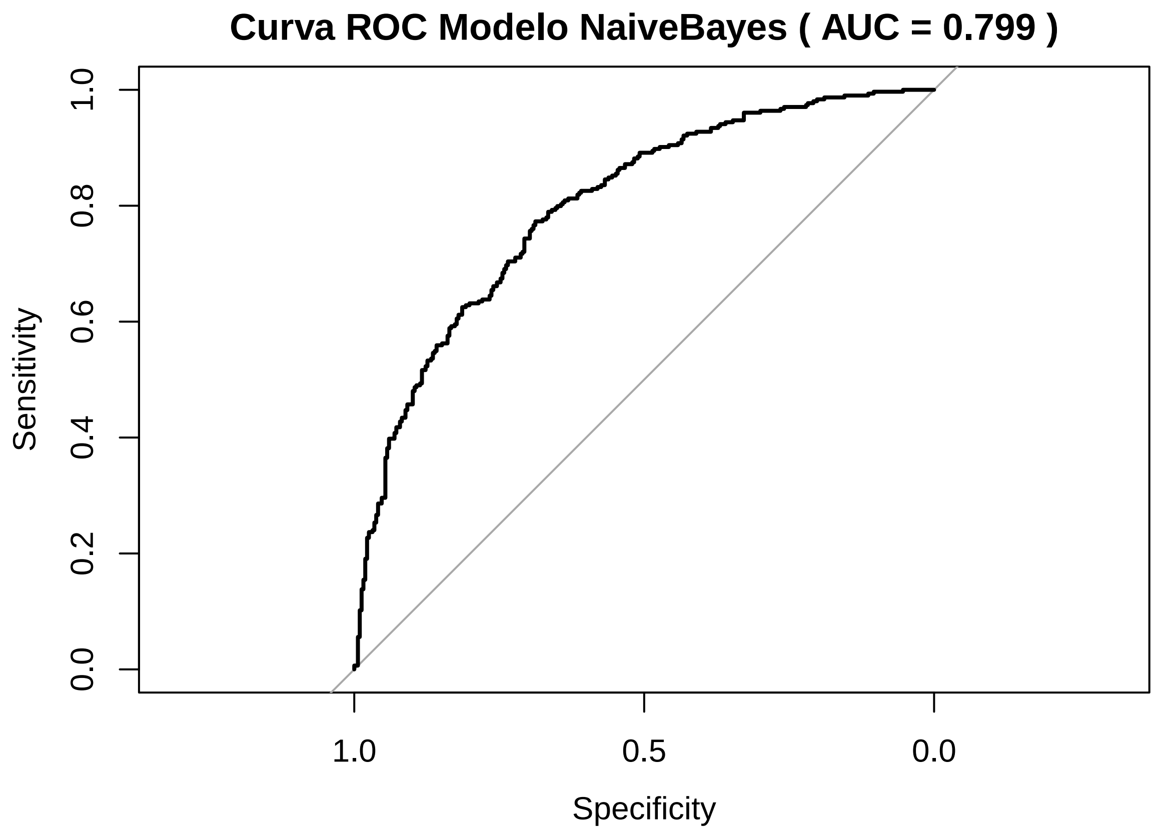
```
## [1] 72.6248
```

```r
curva_roc_model_nb <- roc(match_reales_test, preds_nb2)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls > cases
```

```r
plot(curva_roc_model_nb, main = paste("Curva ROC Modelo NaiveBayes ( AUC =", round(curva_roc_model_nb$au
```




```r
modelo_nb2 <- naiveBayes(match ~.,cbind(datos_final_sc[variables_modelo2], match_train))
modelo_nb2
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.4952381 0.5047619
##
## Conditional probabilities:
##    age_o
## Y           [,1]      [,2]
##   0 0.005656487 0.9846191
##   1 0.024332346 0.9894163
##
##    samerace
## Y           [,1]      [,2]
##   0  0.01496479 1.0032125
##   1 -0.01570866 0.9973653
##
##    importance_same_race
```

```
## Y           [,1]      [,2]
##   0  0.09963667 1.0233069
##   1 -0.11214082 0.9624911
##
##    importance_same_religion
## Y           [,1]      [,2]
##   0  0.03549610 0.9996628
##   1 -0.06000594 0.9801814
##
##    pref_o_attractive
## Y           [,1]      [,2]
##   0 0.003299894 0.9720996
##   1 0.031947103 1.0841447
##
##    pref_o_intelligence
## Y           [,1]      [,2]
##   0 -0.04955318 1.0517008
##   1  0.02849611 0.9946808
##
##    pref_o_funny
## Y           [,1]      [,2]
##   0 -0.06700968 0.9109109
##   1  0.05471344 1.0988989
##
##    pref_o_shared_interests
## Y           [,1]      [,2]
##   0  0.02193558 0.9567117
##   1 -0.07379913 1.0406235
##
##    attractive_o
## Y          [,1]      [,2]
##   0 -0.3710119 1.0362965
##   1  0.3474071 0.8448677
##
##    sinsere_o
## Y          [,1]     [,2]
##   0 -0.2025192 1.054529
##   1  0.2093797 0.896755
##
##    funny_o
## Y          [,1]     [,2]
##   0 -0.4290992 1.038916
##   1  0.4172037 0.807388
##
##    shared_interests_o
## Y          [,1]      [,2]
##   0 -0.4026299 0.9885114
##   1  0.3919324 0.8841241
##
##    attractive_important
## Y           [,1]      [,2]
##   0  0.02865025 1.0022910
##   1 -0.03793376 0.9997948
##
```

```
##    sincere_important
## Y           [,1]      [,2]
##   0  0.01839945 1.0345820
##   1 -0.03012725 0.9922474
##
##    intelligence
## Y           [,1]      [,2]
##   0 -0.04420876 1.0335883
##   1  0.05729902 0.9629094
##
##    attractive_partner
## Y          [,1]      [,2]
##   0 -0.3176257 0.9949156
##   1  0.3730991 0.7980468
##
##    funny_partner
## Y          [,1]      [,2]
##   0 -0.3795056 1.0050684
##   1  0.3703408 0.8226593
##
##    ambition_partner
## Y          [,1]      [,2]
##   0 -0.2069493 1.0104728
##   1  0.2074201 0.9246503
##
##    shared_interests_partner
## Y          [,1]      [,2]
##   0 -0.3684137 0.9587492
##   1  0.3789534 0.8838068
##
##    sports
## Y           [,1]      [,2]
##   0 -0.01626490 0.9985207
##   1 -0.02555759 1.0155141
##
##    tvsports
## Y           [,1]      [,2]
##   0  0.02064614 1.0168545
##   1 -0.02937276 0.9811854
##
##    art
## Y           [,1]      [,2]
##   0 -0.07415212 0.9724649
##   1  0.09294971 0.9893408
##
##    reading
## Y           [,1]      [,2]
##   0 -0.06327711 1.0608223
##   1  0.03002764 0.9142332
##
##    tv
## Y            [,1]      [,2]
##   0  0.007956381 1.0051704
##   1 -0.012676160 0.9757215
```

```
##
##    theater
## Y            [,1]      [,2]
##   0 -0.006066056 0.9703566
##   1  0.002281742 1.0139289
##
##    concerts
## Y           [,1]      [,2]
##   0 -0.05482764 0.9927214
##   1  0.04090749 1.0115725
##
##    music
## Y            [,1]      [,2]
##   0 -0.020088518 0.9823813
##   1 -0.002220439 1.0030257
##
##    shopping
## Y           [,1]      [,2]
##   0  0.03695667 0.9824118
##   1 -0.04509594 0.9856052
##
##    expected_happy_with_sd_people
## Y           [,1]      [,2]
##   0 -0.03336271 0.9932720
##   1  0.03786178 0.9914717
##
##    like
## Y          [,1]      [,2]
##   0 -0.4087194 1.0152439
##   1  0.4415018 0.7735341
##
##    guess_prob_liked
## Y          [,1]      [,2]
##   0 -0.3361117 0.9451217
##   1  0.3446245 0.9282603
```

```r
preds_nb_2 <- predict(modelo_nb2, test[variables_modelo2], type = 'class')
preds_nb2_2 <- predict(modelo_nb2, test[variables_modelo2], type = 'raw')[,1]
```

```r
tabla_model_nb2 <- table(match_reales_test, preds_nb_2)
porcentaje_aciertos_model_nb2 <- sum(diag(tabla_model_nb2))/sum(tabla_model_nb2)*100
porcentaje_aciertos_model_nb2
```

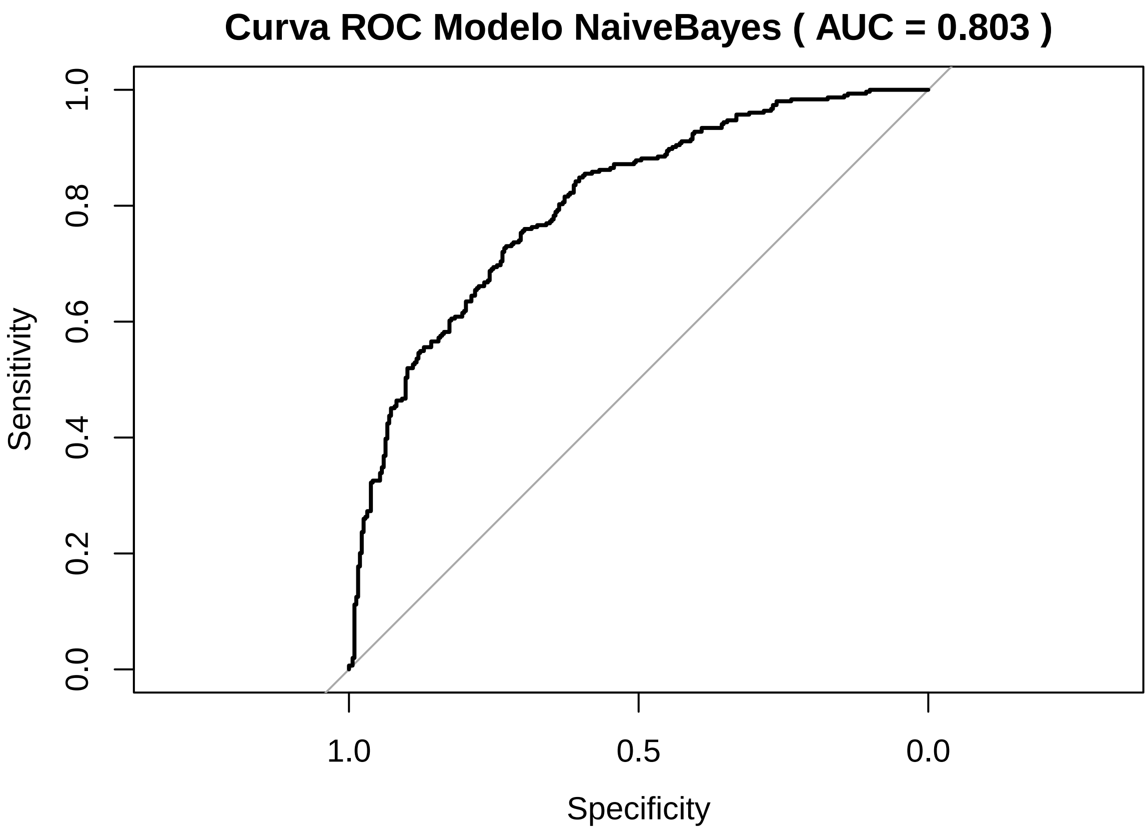
```
## [1] 71.81965
```

```r
curva_roc_model_nb2 <- roc(match_reales_test, preds_nb2_2)
```

```
## Setting levels: control = 0, case = 1
```

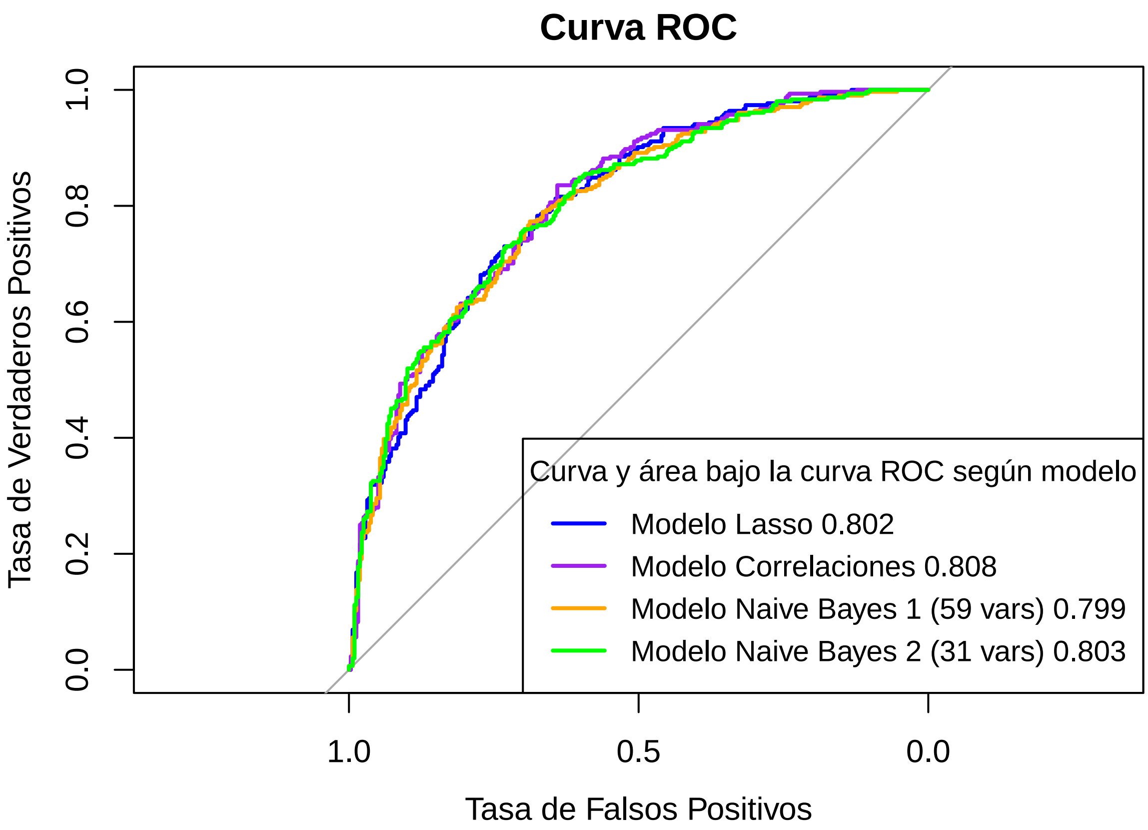
```
## Setting direction: controls > cases
```

```r
plot(curva_roc_model_nb2, main = paste("Curva ROC Modelo NaiveBayes ( AUC =", round(curva_roc_model_nb2
```



Comparación de modelos

```
plot(curva_roc_model_lasso, col = "blue", main = "Curva ROC", xlab = "Tasa de Falsos Positivos", ylab =
lines(curva_roc_model_corr, col = 'purple')
lines(curva_roc_model_nb, col = 'orange')
lines(curva_roc_model_nb2, col = 'green')
legend('bottomright', title = 'Curva y área bajo la curva ROC según modelo',
                                legend = c(paste('Modelo Lasso', round(curva_roc_model_lasso$auc,3)),
                                paste('Modelo Correlaciones', round(curva_roc_model_corr$auc, 3)),
                                paste('Modelo Naive Bayes 1 (59 vars)', round(curva_roc_model_nb$auc,3)
                                paste('Modelo Naive Bayes 2 (31 vars)', round(curva_roc_model_nb2$auc,3
                                col = c('blue', 'purple', 'orange', 'green'),cex = 0.9, lty=1, lwd = 2)
```



Según el área bajjo la curva ROC el mejor modelo es el modelo de regresión logística calculado con las 15 variables más correladas con la variable objetivo 'match', aunque le siguen de cerca los demás modelos.

```
modelo_usado <- c('Regresión Logística', 'Regresión Logística', 'Naive Bayes', 'Naive Bayes')
numero_vars <- c(length(variables_corr), length(variables_lasso), 59, length(variables_modelo2))
metodo_extraccion_vars <- c('Correlaciones', 'Lasso', 'Todas las variables', 'Stepwise')
precision_final <- c(porcentaje_aciertos_model_corr, porcentaje_aciertos_model_lasso, porcentaje_aciert
auc_final <- c(round(curva_roc_model_corr$auc,3), round(curva_roc_model_lasso$auc,3), round(curva_roc_mo

comparacion_final <- data.frame(modelo_usado, numero_vars, metodo_extraccion_vars, precision_final, auc_
comparacion_final
```

```
##          modelo_usado numero_vars metodo_extraccion_vars precision_final
## 1 Regresión Logística          17          Correlaciones        71.65862
## 2 Regresión Logística          15                  Lasso        71.65862
## 3         Naive Bayes          59    Todas las variables        72.62480
## 4         Naive Bayes          31               Stepwise        71.81965
##   auc_final
## 1     0.808
## 2     0.802
## 3     0.799
## 4     0.803
```

En este último resumen final, vemos que los 4 modelos funcionan de forma muy similar. Por el mayor rendimiento en el valor del área bajo la curva y al ser un modulo con un número razonable de variables, me quedaría con el modelo de regresión logística por correlaciones (ya que elige de forma intuitiva las variables y no es del todo complejo.)