Deliverable 3

Guillem Valls, Sergio Mazzariol

Table of Contents

Modelización con target numérico

Modelización con variables explicativas numéricas

Modelo simple

El primer paso es decidir con cuantas variables contamos para el modelo. Si tuviéramos muchas variables explicativas podríamos utilizar el resultado del condes para saber cuáles de ellas utilizar, aunque también sería posible seleccionarlas a partir del análisis de componentes principales. Dado que tenemos poca cantidad de variables usamos todas.

Empezamos utilizando **lm** para crear un modelo inicial del cual podemos ir descartando aquellas variables explicativas que nos parecen irrelevantes. Después contrastaremos nuestra selección usando el método Akaike o BIC, que en una sucesión de pasos va descartando variables.

```r
m1<-lm(duration~.,data=df[,c("duration",vars_num)])
summary(m1)
```

```
##
## Call:
## lm(formula = duration ~ ., data = df[, c("duration", vars_num)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -328.23 -154.46  -82.08   61.30 1842.65
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     777.28481 2613.90120   0.297   0.7662
## age               0.03205    0.34459   0.093   0.9259
## campaign         -6.21960    1.53172  -4.061 4.97e-05 ***
## pdays            -2.37020    1.40614  -1.686   0.0919 .
## previous        -17.62769    9.52959  -1.850   0.0644 .
## emp.var.rate      3.48261   13.07499   0.266   0.7900
## cons.price.idx   11.61303   15.53269   0.748   0.4547
## cons.conf.idx    -0.51158    1.24917  -0.410   0.6822
## euribor3m         3.62210   16.39663   0.221   0.8252
## nr.employed      -0.30339    0.28145  -1.078   0.2811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 251.1 on 4980 degrees of freedom
## Multiple R-squared:  0.006393,   Adjusted R-squared:  0.004597
## F-statistic:  3.56 on 9 and 4980 DF,  p-value: 0.0002021
```

```r
Anova(m1)
```

```
## Anova Table (Type II tests)
##
## Response: duration
##                   Sum Sq   Df F value    Pr(>F)
## age                  545    1  0.0087   0.92589
## campaign          1039241    1 16.4879 4.971e-05 ***
## pdays              179087    1  2.8413   0.09193 .
## previous           215671    1  3.4217   0.06440 .
## emp.var.rate         4472    1  0.0709   0.78998
## cons.price.idx      35233    1  0.5590   0.45471
## cons.conf.idx       10571    1  0.1677   0.68216
## euribor3m            3076    1  0.0488   0.82518
## nr.employed         73240    1  1.1620   0.28111
## Residuals      313891375 4980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Viendo este volcado, vemos que todas las variables menos, campaign tienen un p-value superior al 0.05, sin embargo, pdays y previous están por debajo de 0.1 lo que

podríamos llegar a incorporarlas al modelo. El r-square es de 0.006393 lo que nos dice que nuestro modelo no se ajusta bien.

Al ver el resultado de Anova, podemos ver resultados muy parecidos.

Ahora probaremos seleccionando las variables a partir de la criba anterior:

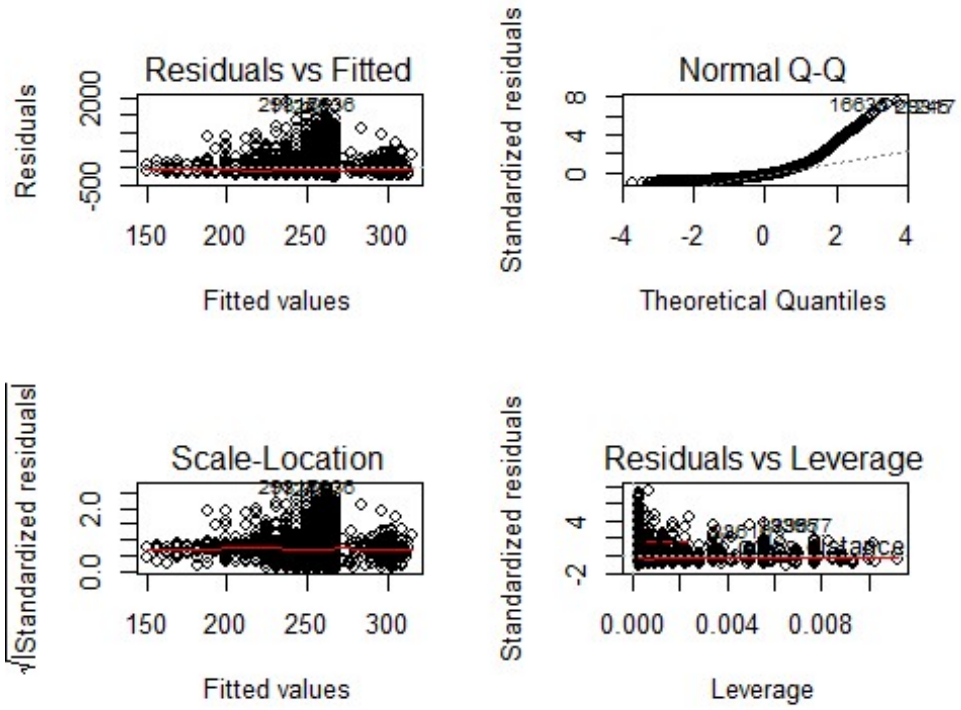
```
m2<-lm(duration~campaign+pdays+previous,data=df)
summary(m2)
```

```
##
## Call:
## lm(formula = duration ~ campaign + pdays + previous, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.50 -156.27  -82.24   61.80 1840.02
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  344.591     32.204  10.700  < 2e-16 ***
## campaign      -6.304      1.513  -4.167 3.14e-05 ***
## pdays         -2.991      1.377  -2.172   0.0299 *
## previous     -10.391      8.726  -1.191   0.2337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 251.1 on 4986 degrees of freedom
## Multiple R-squared:  0.004472,   Adjusted R-squared:  0.003873
## F-statistic: 7.465 on 3 and 4986 DF,  p-value: 5.52e-05
```

```
m3<-lm(duration~campaign+pdays,data=df)
summary(m3)
```

```
##
## Call:
## lm(formula = duration ~ campaign + pdays, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -263.02 -156.25  -82.58   60.87 1840.89
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  321.114     25.467  12.609  < 2e-16 ***
## campaign      -6.183      1.510  -4.095 4.28e-05 ***
## pdays         -2.040      1.122  -1.818   0.0691 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 251.2 on 4987 degrees of freedom
```

```
## Multiple R-squared:  0.004189,   Adjusted R-squared:  0.003789
## F-statistic: 10.49 on 2 and 4987 DF,  p-value: 2.848e-05

vif(m3)

## campaign    pdays
## 1.003138 1.003138

par(mfrow=c(2,2))
plot(m3)
```




```
par(mfrow=c(1,1))
m=m3;
```

Viendo el resultado del lm con estas variables, podemos ver que previous da por encima de 0.2, por lo que también descartamos esta variable. También podemos ver que el r-square sigue siendo muy bajo.

Al realizar nuevamente el lm con estas dos variables restantes, vemos que su p-value es inferior al 0.1, por lo que daríamos por concluida la criba.

Finalmente hacemos el análisis de residuos con vif, el cual nos dice si existen problemas de colinealidad es decir si existen variables que pueden explicar a otras. Si nos da valores por debajo de 3 son buenos y por encima de 5 que las variables elegidas tienen redundancia y que inflará las varianzas. En nuestro caso, el resultado de las dos variables es inferior a 3.

Viendo el plot de la normal Q-Q, vemos que los valores distan mucho de la recta de referencia, con que podemos decir que su distribución no es para nada normal.

Para quitar las variables redundantes probamos con la versión bayesiana del step (del BIC):

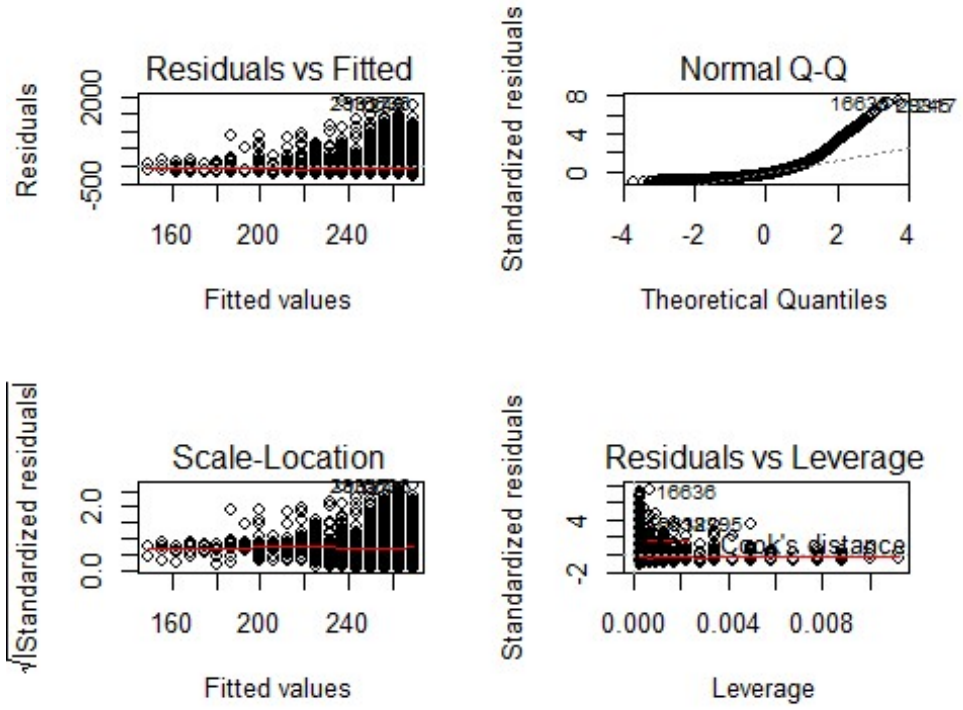
```
m5<-step(m,k=log(nrow(df)))
```

```
## Start:  AIC=55172.94
## duration ~ campaign + pdays
##
##            Df Sum of Sq       RSS   AIC
## - pdays     1    208524 314796334 55168
## <none>                  314587810 55173
## - campaign  1   1058016 315645826 55181
##
## Step:  AIC=55167.73
## duration ~ campaign
##
##            Df Sum of Sq       RSS   AIC
## <none>                  314796334 55168
## - campaign  1   1114698 315911032 55177
```

```
summary(m5)
```

```
##
## Call:
## lm(formula = duration ~ campaign, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.45 -156.69  -82.45   61.14 1840.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  275.786      5.199  53.051  < 2e-16 ***
## campaign      -6.336      1.508  -4.203 2.68e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 251.2 on 4988 degrees of freedom
## Multiple R-squared:  0.003529,   Adjusted R-squared:  0.003329
## F-statistic: 17.66 on 1 and 4988 DF,  p-value: 2.684e-05
```

```
par(mfrow=c(2,2))
plot(m5)
```




```
par(mfrow=c(1,1))
```

La versión bayesiana es conveniente usarla en casos de muestras grandes. En este caso vemos que se queda con una sola variable (campaign), ya que en el primer step del volcado vemos que sin la variable p-days el valor AIC, en este caso BIC, es menor.

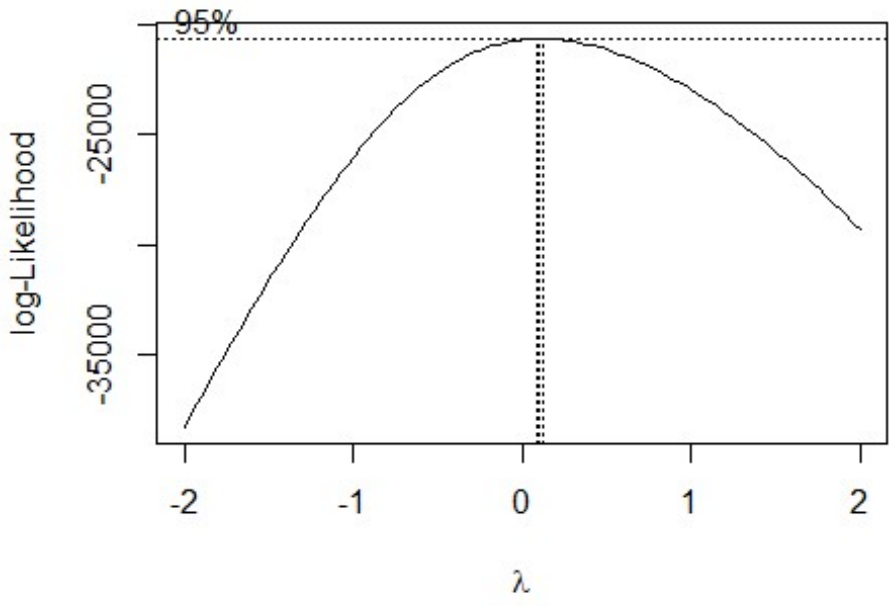
En este caso no podemos hacer el análisis de residuos con vif porque solo tenemos 1 variable.

Al igual que en nuestro caso nos da una plot Q-Q totalmente desviada de las dist normal.

Modelo con transformaciones

Mediante la función boxcox descartamos la posibilidad de elevar el target al cuadrado, pero sí contemplamos aplicarle el logaritmo, pues el pico de la curva está entre 0 y 1, bastante cerca del 0.

```
boxcox(m,data=df)
```



Ahora procedemos a la transformación polinómica.

Como solo tenemos una variable explicativa podemos empezar desde cero, pero si tuviéramos ya un modelo no volveríamos a empezar.

```
m6<-lm(log(duration)~.,data=df[,c("duration",vars_num)])
Anova(m6)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##                 Sum Sq   Df  F value  Pr(>F)
## age                0.1    1   0.1176 0.73162
## campaign          97.4    1 120.3195 < 2e-16 ***
## pdays              4.0    1   4.9361 0.02635 *
## previous           0.2    1   0.1873 0.66523
## emp.var.rate       0.2    1   0.1976 0.65665
## cons.price.idx     0.4    1   0.4944 0.48201
## cons.conf.idx      0.1    1   0.1082 0.74227
## euribor3m          1.6    1   1.9413 0.16359
## nr.employed        2.7    1   3.3650 0.06666 .
## Residuals       4030.4 4980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Viendo el resultado del Anova, procedemos a descartar las variables cuyo valor de Pr es mayor a 0.1

```r
m7<-lm(log(duration)~campaign+pdays+nr.employed,data=df)
summary(m7)
```

```
##
## Call:
## lm(formula = log(duration) ~ campaign + pdays + nr.employed,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6815 -0.5509 -0.0106  0.5858  2.6860
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.3887200  0.9445802   6.764  1.5e-11 ***
## campaign    -0.0598301  0.0054664 -10.945  < 2e-16 ***
## pdays       -0.0135538  0.0042873  -3.161  0.00158 **
## nr.employed -0.0001463  0.0001888  -0.775  0.43843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9007 on 4986 degrees of freedom
## Multiple R-squared:  0.02798,    Adjusted R-squared:  0.0274
## F-statistic: 47.84 on 3 and 4986 DF,  p-value: < 2.2e-16
```

```r
Anova(m7)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##             Sum Sq   Df  F value  Pr(>F)
## campaign      97.2    1 119.7932 < 2e-16 ***
## pdays          8.1    1   9.9945 0.00158 **
## nr.employed    0.5    1   0.6005 0.43843
## Residuals   4044.5 4986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Viendo los p-values, nos encontramos que la variable nr.employed es mayor a 0.1, por lo que procedemos a eliminarla de nuestro modelo.

Relativo al gráfico, podemos ver como la Normal Q-Q ha mejorado bastante acercándose a la recta ideal.

Ahora procedemos a quitar nr.employed.

```r
m9<-lm(log(duration)~campaign+pdays,data=df)
summary(m9)
```

```
##
## Call:
## lm(formula = log(duration) ~ campaign + pdays, data = df)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6522 -0.5521 -0.0090  0.5858  2.6797
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.660184   0.091319  61.982  < 2e-16 ***
## campaign    -0.060418   0.005413 -11.161  < 2e-16 ***
## pdays       -0.014703   0.004023  -3.655  0.00026 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9006 on 4987 degrees of freedom
## Multiple R-squared:  0.02786,    Adjusted R-squared:  0.02747
## F-statistic: 71.47 on 2 and 4987 DF,  p-value: < 2.2e-16
```

```
Anova(m9)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##           Sum Sq   Df F value    Pr(>F)
## campaign   101.0    1  124.57 < 2.2e-16 ***
## pdays       10.8    1   13.36 0.0002597 ***
## Residuals 4045.0 4987
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(m9)
```

```
## campaign    pdays
## 1.003138 1.003138
```

Viendo el valor final del r-square, podemos ver que este no es un buen modelo. También los que no puede decir es que las variables no representan a nuestro target, esto ya lo pudimos ver en el deliverable2.

El resultado del vif nos da valores aceptables, diciendo que no hay colinealidad entre variables.

Modelo de regresión polinómica

Ahora podemos probar con las versiones cuadráticas de las variables explicativas, partiendo de nuestro mejor modelo:

```
m20<-lm(log(duration)~poly(campaign,2)+poly(pdays,2),data=df)
summary(m20)
```

```
##
## Call:
```

```
## lm(formula = log(duration) ~ poly(campaign, 2) + poly(pdays,
##     2), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6353 -0.5534 -0.0100  0.5842  2.6431
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.17868    0.01274 406.451  < 2e-16 ***
## poly(campaign, 2)1 -10.03807    0.90154 -11.134  < 2e-16 ***
## poly(campaign, 2)2  -1.79572    0.90036  -1.994 0.046158 *
## poly(pdays, 2)1     -3.34605    0.90176  -3.711 0.000209 ***
## poly(pdays, 2)2     -1.90923    0.90014  -2.121 0.033968 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9 on 4985 degrees of freedom
## Multiple R-squared:  0.02951,    Adjusted R-squared:  0.02873
## F-statistic: 37.89 on 4 and 4985 DF,  p-value: < 2.2e-16
```

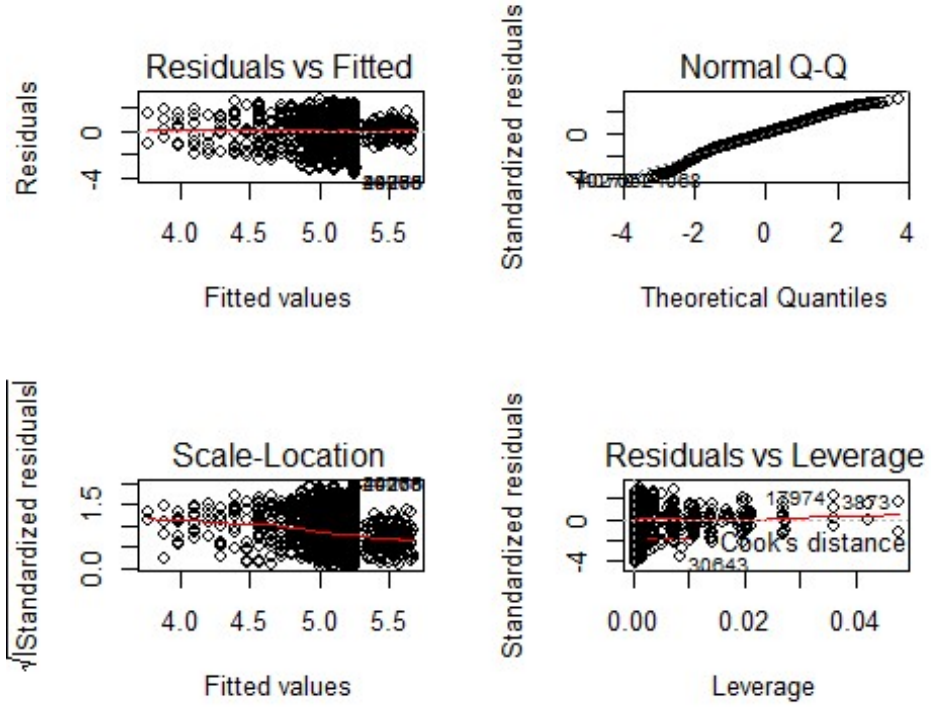
```
Anova(m20)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##                   Sum Sq   Df F value    Pr(>F)
## poly(campaign, 2)  103.7    2 64.0104 < 2.2e-16 ***
## poly(pdays, 2)      14.8    2  9.1263 0.0001106 ***
## Residuals         4038.2 4985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(m20)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## poly(campaign, 2) 1.004062  2        1.001014
## poly(pdays, 2)    1.004062  2        1.001014
```

```
par(mfrow=c(2,2))
plot(m20)
```




```
par(mfrow=c(1,1))
```

Modelización con variables explicativas numéricas y categóricas

Creamos una variable que contiene las variables categóricas y categóricas factorizadas además de las categóricas.

```
vars_cat_total = c(vars_cat, names(df[,22:29]))
condes(df[,c("duration",vars_cat_total)],1, proba= 0.05)
```

```
## $quali
##                       R2      p.value
## f.duration 0.621168787 0.000000e+00
## f.campaign 0.003783221 7.858324e-05
## month      0.004450289 8.185248e-03
## poutcome   0.001675246 1.528736e-02
## f.pdays    0.001120416 1.805086e-02
## f.season   0.001469523 2.555430e-02
##
## $category
##                                Estimate       p.value
## f.duration-(300,2.1e+03]  310.351061  0.000000e+00
## f.campaign-(1,2]           23.010415  3.895001e-05
## month.apr                  35.257830  4.865526e-03
## f.season.Mar-May           13.191703  6.782891e-03
## poutcome.success           38.426383  1.231875e-02
## f.pdays-[0,22]             22.891544  1.805086e-02
```

```
## day_of_week.wed             14.619928  3.788283e-02
## job.retired                 34.239467  3.904250e-02
## marital.divorced           -15.444147  4.653367e-02
## f.season.Jun-Aug            -6.204726  4.445499e-02
## f.pdays-(22,23]            -22.891544  1.805086e-02
## month.aug                  -25.222251  7.943838e-03
## f.campaign-(2,20]          -17.351641  3.316706e-03
## f.duration-(180,300]       -20.507215  3.927333e-04
## f.duration-(120,180]      -106.753548  5.404997e-53
## f.duration-[5,120]        -183.090298 1.278559e-312
```

Al hacer condes, con todas las variables categóricas, contemplamos el uso de f.campaign y month para nuestro modelo, ya que la probabilidad de que no tengan relación con el target está por debajo del 0.01. Como nos sale la versión categórica de campaign que también nos sale en el modelo numérico, debemos elegir entre una u otra, pero nunca las dos a la vez.

En vista de que la variable numérica pdays aporta una información errante ya que aquellos que no fueron contactados tienen asignados un valor que no les corresponde, decidimos utilizar f.pdays porque contiene una información más rigurosa, ya que se clasifican entre contactados y no contactados.

Debido a que la variable month es una variable con muchos niveles y eso no es bueno para la modelización, decidimos reagruparla.

```
#chunk 115
# Months to groups
df$f.influentMonth <- 3
# 1 level - mar-may
aux<-which(df$month %in% c("month.apr","month.jun","month.aug"))
df$f.influentMonth[aux] <-1

# 2 level - jun-ago
aux<-which(df$month %in% c("month.sep","month.may","month.jul"))
df$f.influentMonth[aux] <-2

# 3 level - aug-feb
aux<-which(df$month %in% c("month.mar","month.dec","month.oct","month.nov
"))
df$f.influentMonth[aux] <-3

df$f.influentMonth<-factor(df$f.influentMonth,levels=1:3,labels=c("apr-ju
n-aug","sep-may-jul","mar-dec-oct-nov"))
levels(df$f.influentMonth)<-paste0("f.influentMonth.",levels(df$f.influen
tMonth)) # Hacemos las etiquetas m?s informativas
summary(df$f.influentMonth)

##     f.influentMonth.apr-jun-aug     f.influentMonth.sep-may-jul
##                            1701                            2615
```

```
## f.influentMonth.mar-dec-oct-nov
##                             674
```

Contrastamos un modelo con campaign o con f.campaign para ver cuál es mejor.

```
m22<-lm(log(duration)~campaign+f.pdays+f.influentMonth,data=df)
m23<-lm(log(duration)~f.pdays+f.campaign+f.influentMonth,data=df)
BIC(m23,m22)
```

```
##     df      BIC
## m23  7 13214.68
## m22  6 13150.71
```

```
# Ya que nos quedamos con el modelo m22
Anova(m22)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##                 Sum Sq   Df  F value    Pr(>F)
## campaign         102.8    1 126.9775 < 2.2e-16 ***
## f.pdays           15.2    1  18.7951 1.484e-05 ***
## f.influentMonth    8.4    2   5.1938  0.005581 **
## Residuals       4033.9 4985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Haciendo BIC para comparar modelos, podemos ver que el que da un menor BIC es m22, por lo que decidimos quedarnos con este modelo. Viendo el resultado del Anova, podemos ver que los p-values son inferiores a 0.1

Interacciones

```
m30<-lm(log(duration)~(campaign+f.pdays+f.influentMonth)^2,data=df)
Anova(m30)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##                           Sum Sq   Df  F value    Pr(>F)
## campaign                   103.1    1 127.5736 < 2.2e-16 ***
## f.pdays                     15.0    1  18.5584  1.68e-05 ***
## f.influentMonth              8.5    2   5.2517  0.005268 **
## campaign:f.pdays             2.2    1   2.7306  0.098506 .
## campaign:f.influentMonth     5.2    2   3.1929  0.041136 *
## f.pdays:f.influentMonth      1.2    2   0.7427  0.475884
## Residuals                 4025.4 4980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vemos que la interacción entre campaign y nuestra nueva variable factor month es significativa, por lo tanto, creamos un nuevo modelo m31 con esa interacción. Por otro

lado, aunque f.pdays con f.influentMoth tiene un p-value muy alto de 0.4, realizamos la interacción porque lo pide el enunciado.

```
#chunk 140
m31<-lm(log(duration)~(f.influentMonth*campaign+f.pdays),data=df)
Anova(m31)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##                           Sum Sq   Df  F value    Pr(>F)
## f.influentMonth             8.4    2   5.1981  0.005557 **
## campaign                  102.8    1 127.0831 < 2.2e-16 ***
## f.pdays                    15.0    1  18.5535 1.684e-05 ***
## f.influentMonth:campaign    5.0    2   3.0728  0.046377 *
## Residuals                4028.9 4983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

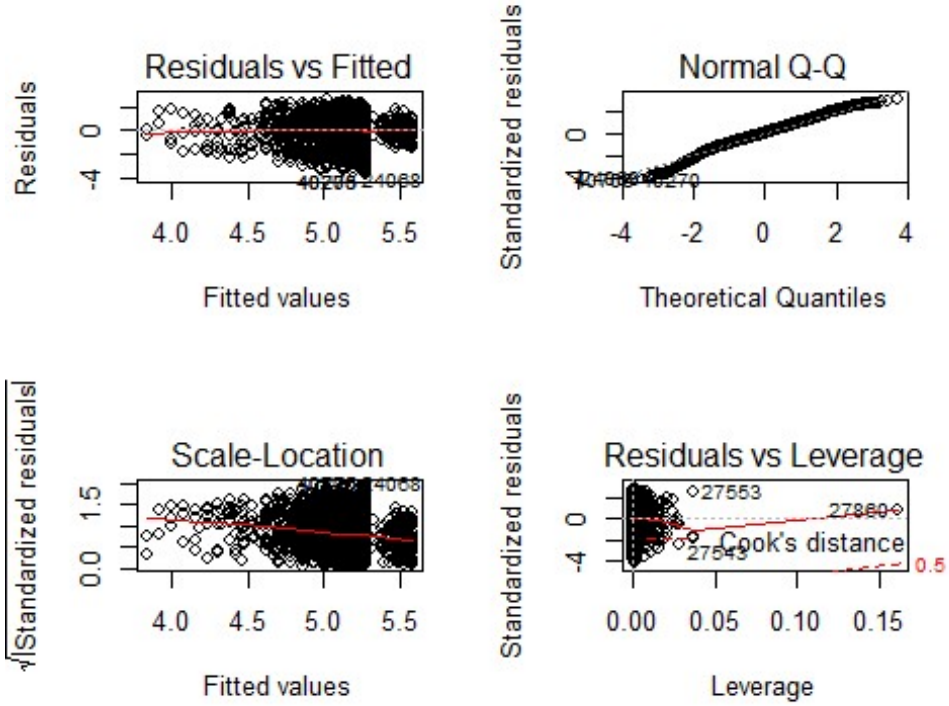
```
m32<-lm(log(duration)~(f.influentMonth*f.pdays+campaign),data=df)
Anova(m32)
```

```
## Anova Table (Type II tests)
##
## Response: log(duration)
##                          Sum Sq   Df  F value    Pr(>F)
## f.influentMonth            8.4    2   5.1932  0.005584 **
## f.pdays                   15.2    1  18.7930 1.486e-05 ***
## campaign                 103.1    1 127.4200 < 2.2e-16 ***
## f.influentMonth:f.pdays    1.2    2   0.7228  0.485455
## Residuals               4032.7 4983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vemos que el modelo 31 es aceptable, sus p-values son aceptables, mientras como ya era previsible el modelo m32 lo descartamos.

Validación

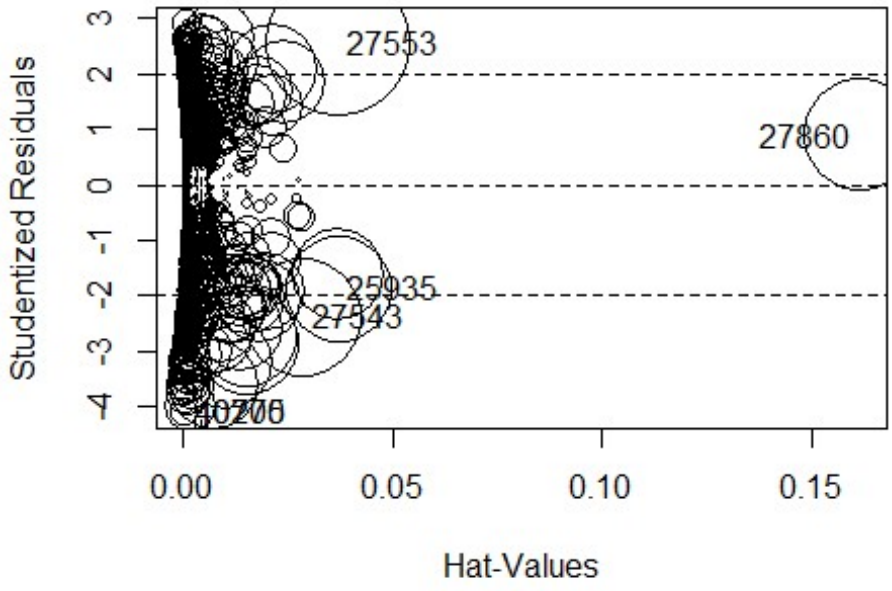
```
par(mfrow=c(2,2))
plot(m31)
```




```
par(mfrow=c(1,1))
```

Analizando los gráficos:
- Residual VS Fitted. En este gráfico muestra los residuos de los valores predichos. Lo deseable es que los puntos estén uniformemente dispersos, para poderlo contrastar el gráfico está provisto de una recta smoother que conviene que sea horizontal, y uniforme. A pesar de que podemos ver un patrón en el gráfico, podemos decir que el resultado no es aceptable. - Normal Q-Q. Este plot nos muestra la tendencia a una distribución normal de los residuos, esta provista de una recta diagonal de referencia en la que se espera que los residuos se ajusten lo máximo posible. En nuestro caso, apreciamos ciertas desviaciones en los extremos de la recta, aunque si lo comparamos con plots anteriores, se acerca más a la normal, pero sigue siendo poco aceptable.
- Scale-Location. Este plot hace referencia a la varianza de los valores de la predicción, si se mantiene constante implica homocedasticidad, de lo contrario heterocedasticidad que se vería reflejada en una nube de puntos en forma de cono. Para nuestro caso, podemos ver que el gráfico tiene una tendencia a cono que además se evidencia con la desviación de la smoother line. Pero es una heterocedasticidad que es imposible de corregir de manera fácil, es una réplica del primer plot. - Residuals Vs Leverage. Vemos que hay un individuo con mucho leverage, el 27860. Utilizaremos el influencePlot para poder ver con más detalles los individuos influyentes.

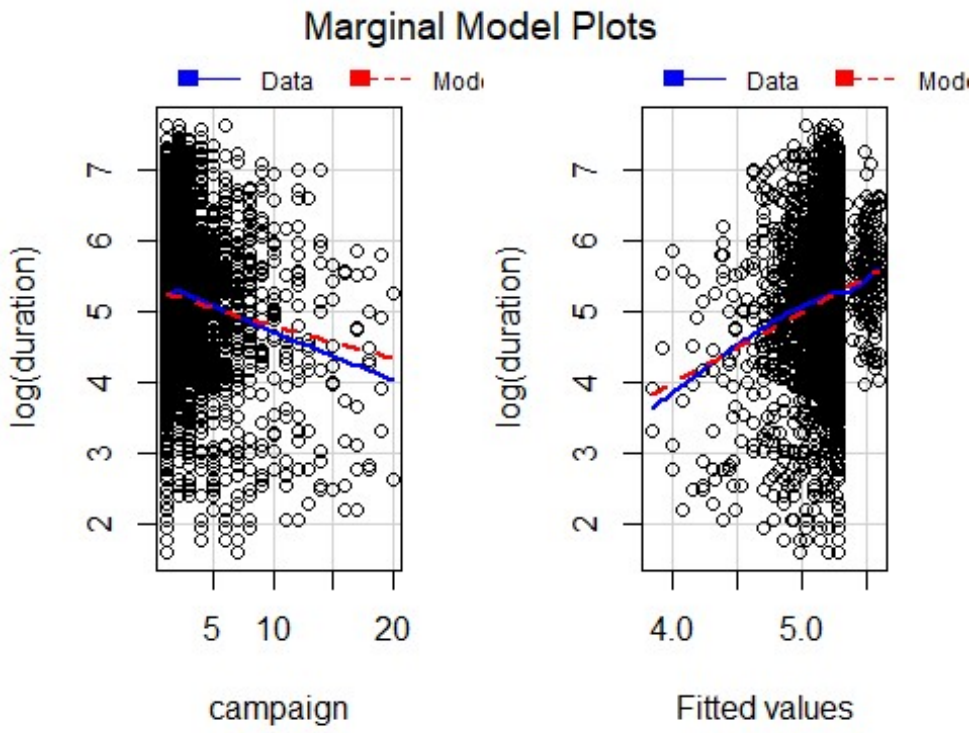
```
#chunk 150
influencePlot(m31)
```




```
##          StudRes           Hat       CookD
## 25935 -1.866796 0.0369674028 0.019101054
## 27543 -2.380400 0.0286710547 0.023871117
## 27553  2.548824 0.0369674028 0.035586127
## 27860  0.882873 0.1613511871 0.021424449
## 40270 -4.092113 0.0005533216 0.001320216
## 40705 -4.092113 0.0005533216 0.001320216
```

```
marginalModelPlots(m31)
```

```
## Warning in mmps(...): Interactions and/or factors skipped
```




```
which(row.names(df)==27860)
```

```
## [1] 3329
```

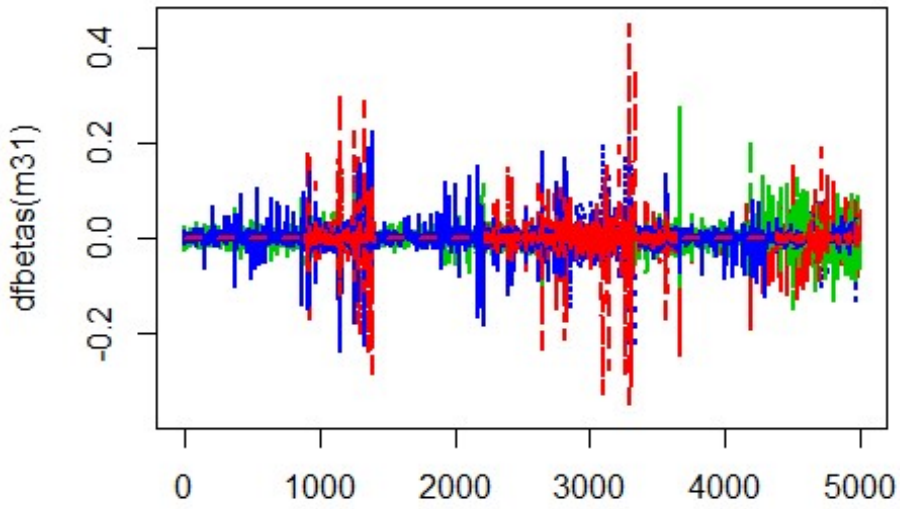
```
which(row.names(df)==27553)
```

```
## [1] 3293
```

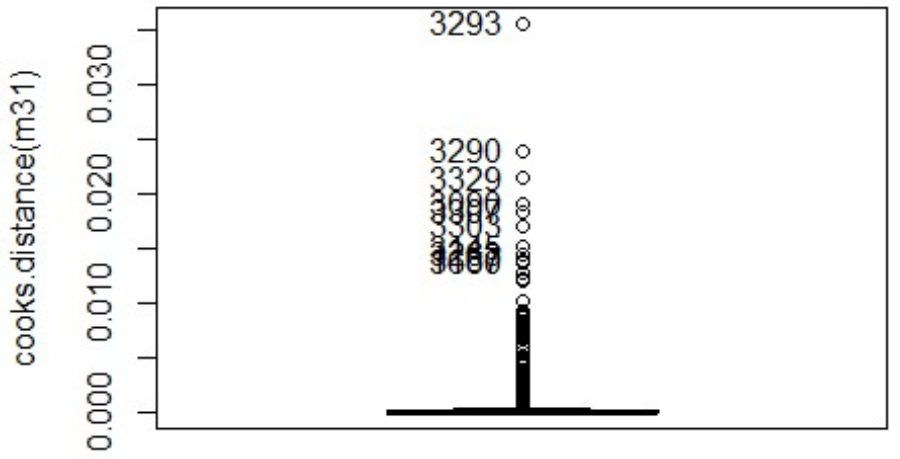
- InfluenPLot. Nos muestra los individuos más influentes, esto se puede ver gráficamente a través del radio de las circunferencias. En nuestro caso, viendo el gráfico podemos ver que hay individuos bastante influyentes, el 3329 y 3293 que para nuestra muestra serían los individuos.
- MarginalModelPlot. Nos muestra las discrepancias entre las predicciones de nuestro modelo y los resultados reales de nuestras observaciones desglosado por variables, utiliza dos líneas de soporte, una roja para la tendencia del modelo y otra azul referente a cada variable. Podemos ver que, para nuestro modelo, las líneas tienen un poco de desviación entre ellas, pero nada muy relevante.

Trabajamos con el mejor modelo obtenido, y vemos que individuos influyen más en nuestros datos para saber si están afectando nuestro resultado.

```
matplot(dfbetas(m31), type="l", col=2:4,lwd=2)
```



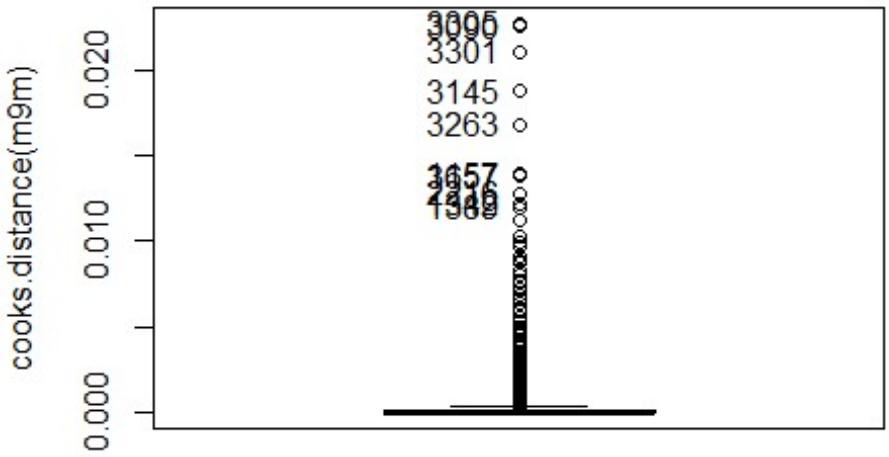

```
Boxplot(cooks.distance(m31))
```




```
##  [1] 3293 3290 3329 3090 3307 3303 3145 3263 1157 3660
```

Consideramos que hay un individuo que repercute demasiado en los datos (3293), aun así, no lo eliminaremos.

```
m9m<-lm(log(duration)~(f.influentMonth*campaign+f.pdays),data=df[c(-3293,
-3290,-3329),])

 Boxplot(cooks.distance(m9m))
```




```
##  [1] 3305 3090 3301 3145 3263 1157 3657 2216 1342 1389

summary(m9m)

##
## Call:
## lm(formula = log(duration) ~ (f.influentMonth * campaign + f.pdays),
##     data = df[c(-3293, -3290, -3329), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6728 -0.5449 -0.0105  0.5877  2.6092
##
## Coefficients:
##                                                        Estimate
## (Intercept)                                            5.631270
## f.influentMonthf.influentMonth.sep-may-jul             0.001571
## f.influentMonthf.influentMonth.mar-dec-oct-nov        -0.057482
## campaign                                              -0.078799
## f.pdaysf.pdays-(22,23]                                -0.300195
```

```
## f.influentMonthf.influentMonth.sep-may-jul:campaign      0.028354
## f.influentMonthf.influentMonth.mar-dec-oct-nov:campaign  0.011485
##                                                        Std. Error t v
alue
## (Intercept)                                              0.073735  76
.372
## f.influentMonthf.influentMonth.sep-may-jul               0.040830   0
.038
## f.influentMonthf.influentMonth.mar-dec-oct-nov           0.066125  -0
.869
## campaign                                                 0.008971  -8
.784
## f.pdaysf.pdays-(22,23]                                   0.069722  -4
.306
## f.influentMonthf.influentMonth.sep-may-jul:campaign      0.011481   2
.470
## f.influentMonthf.influentMonth.mar-dec-oct-nov:campaign  0.025573   0
.449
##                                                        Pr(>|t|)
## (Intercept)                                             < 2e-16 ***
## f.influentMonthf.influentMonth.sep-may-jul               0.9693
## f.influentMonthf.influentMonth.mar-dec-oct-nov           0.3847
## campaign                                                < 2e-16 ***
## f.pdaysf.pdays-(22,23]                                  1.7e-05 ***
## f.influentMonthf.influentMonth.sep-may-jul:campaign      0.0136 *
## f.influentMonthf.influentMonth.mar-dec-oct-nov:campaign  0.6534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8983 on 4980 degrees of freedom
## Multiple R-squared:  0.03195,    Adjusted R-squared:  0.03078
## F-statistic: 27.39 on 6 and 4980 DF,  p-value: < 2.2e-16

summary(m31)

##
## Call:
## lm(formula = log(duration) ~ (f.influentMonth * campaign + f.pdays),
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6728 -0.5452 -0.0098  0.5884  2.6092
##
## Coefficients:
##                                                         Estimate
## (Intercept)                                             5.631659
## f.influentMonthf.influentMonth.sep-may-jul              0.001578
## f.influentMonthf.influentMonth.mar-dec-oct-nov         -0.077716
## campaign                                               -0.078797
```

```
## f.pdaysf.pdays-(22,23]                                        -0.300603
## f.influentMonthf.influentMonth.sep-may-jul:campaign            0.028353
## f.influentMonthf.influentMonth.mar-dec-oct-nov:campaign        0.022602
##                                                              Std. Error t v
alue
## (Intercept)                                                    0.073805  76
.304
## f.influentMonthf.influentMonth.sep-may-jul                     0.040870   0
.039
## f.influentMonthf.influentMonth.mar-dec-oct-nov                 0.063074  -1
.232
## campaign                                                       0.008980  -8
.775
## f.pdaysf.pdays-(22,23]                                         0.069788  -4
.307
## f.influentMonthf.influentMonth.sep-may-jul:campaign            0.011493   2
.467
## f.influentMonthf.influentMonth.mar-dec-oct-nov:campaign        0.022954   0
.985
##                                                               Pr(>|t|)
## (Intercept)                                                   < 2e-16 ***
## f.influentMonthf.influentMonth.sep-may-jul                     0.9692
## f.influentMonthf.influentMonth.mar-dec-oct-nov                 0.2180
## campaign                                                      < 2e-16 ***
## f.pdaysf.pdays-(22,23]                                       1.68e-05 ***
## f.influentMonthf.influentMonth.sep-may-jul:campaign            0.0137 *
## f.influentMonthf.influentMonth.mar-dec-oct-nov:campaign        0.3248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8992 on 4983 degrees of freedom
## Multiple R-squared:  0.03173,    Adjusted R-squared:  0.03057
## F-statistic: 27.22 on 6 and 4983 DF,  p-value: < 2.2e-16
```

Podemos ver que el nuevo modelo sin los individuos influyentes tiene una mejora en el r-square, aunque este sigue siendo muy bajo.

Modelización con target binario

Empezamos dividiendo nuestra muestra en una muestra de trabajo y una muestra de testeo, para ello seleccionaremos aleatoriamente el 25% de la muestra para crear la muestra de testeo.

```
set.seed(19101990)
sam <-sample(1:nrow(df),0.75*nrow(df))

dfw<-df[sam,]
dft<-df[-sam,]
```

Modelización con variables explicativas numéricas

Modelo simple

Para empezar, hacemos un catdes con todas las variables numéricas para ver cuáles son las que están más relacionadas con nuestro target. Las utilizamos para hacer un modelo lineal general con variables explicativas numéricas. Este modelo es de la familia binomial ya que nuestro target es binario.

```
catdes(dfw[,c("y",vars_num,"duration")],1)
```

```
##
## Link between the cluster variable and the quantitative variables
## ================================================================
##                        Eta2       P-value
## duration       0.176628371 4.526217e-160
## nr.employed    0.107146172  3.620826e-94
## pdays          0.098200453  4.721537e-86
## euribor3m      0.077763912  8.384377e-68
## emp.var.rate   0.072681695  2.518903e-63
## previous       0.043535295  4.410367e-38
## cons.price.idx 0.012743864  4.345713e-12
## campaign       0.006955344  3.241195e-07
## age            0.004418712  4.712507e-05
## cons.conf.idx  0.003847937  1.464648e-04
##
## Description of each cluster by quantitative variables
## ======================================================
## $y.no
##                    v.test Mean in category Overall mean sd in category
## nr.employed     20.020835     5174.1172673 5165.6679316     66.5132140
## pdays           19.166844       22.7627628   22.4142170      2.0015446
## euribor3m       17.056225        3.7481483    3.5760190      1.6726437
## emp.var.rate    16.489458        0.1990691    0.0485302      1.5112677
## cons.price.idx   6.904694       93.5892261   93.5660259      0.5683953
## campaign         5.100975        2.5990991    2.5299305      2.4284226
## cons.conf.idx   -3.794092      -40.6458859  -40.5445216      4.4151245
## age             -4.065760       39.8219219   40.0652058      9.8012900
## previous       -12.761878        0.1405405    0.1774452      0.4123568
## duration       -25.705383      219.4867868  255.9438803    196.7693288
##                 Overall sd       p.value
## nr.employed     73.3850752  3.625944e-89
## pdays            3.1621071  7.003107e-82
## euribor3m        1.7548478  3.142184e-65
## emp.var.rate     1.5874850  4.368514e-61
## cons.price.idx   0.5842716  5.031186e-12
## campaign         2.3578875  3.379091e-07
## cons.conf.idx    4.6456264  1.481848e-04
## age             10.4049243  4.787623e-05
## previous         0.5028450  2.676679e-37
```

```
## duration       246.6183107 1.017624e-145
##
## $y.yes
##                    v.test Mean in category Overall mean sd in category
## duration        25.705383      550.6092233  255.9438803     376.687736
## previous        12.761878        0.4757282    0.1774452       0.906767
## age              4.065760       42.0315534   40.0652058      14.230300
## cons.conf.idx    3.794092      -39.7252427  -40.5445216       6.140692
## campaign        -5.100975        1.9708738    2.5299305       1.574717
## cons.price.idx  -6.904694       93.3785097   93.5660259       0.670650
## emp.var.rate   -16.489458       -1.1682039    0.0485302       1.662956
## euribor3m      -17.056225        2.1847791    3.5760190       1.783746
## pdays          -19.166844       19.5970874   22.4142170       7.036851
## nr.employed    -20.020835     5097.3759709 5165.6679316      88.965074
##                  Overall sd       p.value
## duration       246.6183107 1.017624e-145
## previous         0.5028450  2.676679e-37
## age             10.4049243  4.787623e-05
## cons.conf.idx    4.6456264  1.481848e-04
## campaign         2.3578875  3.379091e-07
## cons.price.idx   0.5842716  5.031186e-12
## emp.var.rate     1.5874850  4.368514e-61
## euribor3m        1.7548478  3.142184e-65
## pdays            3.1621071  7.003107e-82
## nr.employed     73.3850752  3.625944e-89
```

```
gm1<-glm( y ~
            duration +
            nr.employed +
            pdays +
            euribor3m +
            emp.var.rate +
            previous +
            cons.price.idx +
            campaign +
            age +
            cons.conf.idx, family = binomial, data = dfw)
summary(gm1)
```

```
##
## Call:
## glm(formula = y ~ duration + nr.employed + pdays + euribor3m +
##     emp.var.rate + previous + cons.price.idx + campaign + age +
##     cons.conf.idx, family = binomial, data = dfw)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6937  -0.3319  -0.1897  -0.1238   2.9794
##
## Coefficients:
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.530e+01  5.242e+01  -0.673  0.50070
## duration        5.008e-03  2.518e-04  19.887  < 2e-16 ***
## nr.employed    -5.899e-03  4.944e-03  -1.193  0.23281
## pdays          -1.205e-01  1.702e-02  -7.079 1.45e-12 ***
## euribor3m       3.016e-02  2.839e-01   0.106  0.91542
## emp.var.rate   -6.405e-01  2.289e-01  -2.797  0.00515 **
## previous       -3.306e-01  1.316e-01  -2.512  0.01201 *
## cons.price.idx  6.955e-01  3.308e-01   2.102  0.03553 *
## campaign       -1.280e-01  4.381e-02  -2.922  0.00348 **
## age             1.356e-02  5.361e-03   2.530  0.01141 *
## cons.conf.idx   3.157e-02  1.987e-02   1.589  0.11208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2594.9  on 3741  degrees of freedom
## Residual deviance: 1584.3  on 3731  degrees of freedom
## AIC: 1606.3
##
## Number of Fisher Scoring iterations: 6
```

```
Anova(gm1)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                LR Chisq Df Pr(>Chisq)
## duration         563.62  1  < 2.2e-16 ***
## nr.employed        1.44  1   0.230622
## pdays             55.07  1  1.161e-13 ***
## euribor3m          0.01  1   0.915436
## emp.var.rate       7.77  1   0.005314 **
## previous           6.63  1   0.010024 *
## cons.price.idx     4.29  1   0.038251 *
## campaign           9.81  1   0.001740 **
## age                6.38  1   0.011547 *
## cons.conf.idx      2.52  1   0.112437
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Viendo el resultado de summary, podemos ver variables que tienen el p-value mayor a 0.1 (cons.cinf.idx, euribor3m), por lo que procedemos a quitarlas de nuestro modelo. Podemos ver que el deviance es inferior al null deviance.

```
gm2<-glm( y ~
            duration +
            nr.employed +
            pdays +
            emp.var.rate +
```

```
            previous +
            campaign +
            age +
            cons.conf.idx, family = binomial, data = dfw)
vif(gm2)
```

```
##      duration   nr.employed         pdays  emp.var.rate      previous
##      1.283771      3.979725      1.829567      3.518890      2.048827
##      campaign           age cons.conf.idx
##      1.029761      1.037190      1.057214
```

Haciendo vif podemos ver que emp.var.rate tiene un valor mayor a 3, por lo que decidimos sacarla de nuestro modelo.

```
gm3<-glm( y ~
            duration +
            nr.employed +
            pdays +
            previous +
            campaign +
            age +
            cons.conf.idx, family = binomial, data = dfw)
vif(gm3)
```

```
##      duration   nr.employed         pdays      previous      campaign
##      1.241533      1.496925      1.820608      2.031438      1.021478
##           age cons.conf.idx
##      1.034254      1.056345
```

Modelo de regresión polinómica

Hacemos un tanteo aplicando una transformación polinómica de segundo grado a cada una de las variables.

```
gm4<-glm(y~
           poly(duration,2) +
            poly(nr.employed,2) +
            poly(pdays,2) +
            poly(previous,2) +
            poly(campaign,2) +
            poly(age,2) +
            poly(cons.conf.idx,2), family = binomial, data = dfw
           )
summary(gm4)
```

```
##
## Call:
## glm(formula = y ~ poly(duration, 2) + poly(nr.employed, 2) +
##     poly(pdays, 2) + poly(previous, 2) + poly(campaign, 2) +
##     poly(age, 2) + poly(cons.conf.idx, 2), family = binomial,
##     data = dfw)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8033  -0.3187  -0.1672  -0.1044   2.9358
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -3.3334     0.1132 -29.449  < 2e-16 ***
## poly(duration, 2)1        82.4000     3.9367  20.931  < 2e-16 ***
## poly(duration, 2)2       -24.3172     3.0845  -7.884 3.18e-15 ***
## poly(nr.employed, 2)1    -61.5257     5.4420 -11.306  < 2e-16 ***
## poly(nr.employed, 2)2      4.9535     3.9118   1.266 0.205411
## poly(pdays, 2)1          -26.0379     3.5679  -7.298 2.92e-13 ***
## poly(pdays, 2)2           -0.9880     2.5784  -0.383 0.701581
## poly(previous, 2)1       -16.9490     4.7079  -3.600 0.000318 ***
## poly(previous, 2)2         8.2807     3.2481   2.549 0.010790 *
## poly(campaign, 2)1       -12.3096     5.9479  -2.070 0.038493 *
## poly(campaign, 2)2        10.7221     5.9999   1.787 0.073929 .
## poly(age, 2)1              5.7959     3.6339   1.595 0.110721
## poly(age, 2)2              6.4103     3.3367   1.921 0.054714 .
## poly(cons.conf.idx, 2)1    4.8461     3.7665   1.287 0.198225
## poly(cons.conf.idx, 2)2    5.7061     3.7847   1.508 0.131638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2594.9  on 3741  degrees of freedom
## Residual deviance: 1523.1  on 3727  degrees of freedom
## AIC: 1553.1
##
## Number of Fisher Scoring iterations: 7
```

En vista del summary, podemos omitir el termino cuadrático de las variables nr.employed, pdays, age, con.conf.idx.

```
gm5<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
summary(gm5)
```

```
##
## Call:
## glm(formula = y ~ poly(duration, 2) + nr.employed + pdays + poly(previ
ous,
```

```
##     2) + poly(campaign, 2) + age + cons.conf.idx, family = binomial,
##     data = dfw)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8321  -0.3202  -0.1672  -0.1022   2.9323
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         79.166461   5.222892  15.158  < 2e-16 ***
## poly(duration, 2)1  82.458219   3.923723  21.015  < 2e-16 ***
## poly(duration, 2)2 -23.887690   3.100519  -7.704 1.31e-14 ***
## nr.employed         -0.015261   0.001019 -14.975  < 2e-16 ***
## pdays               -0.133120   0.017975  -7.406 1.30e-13 ***
## poly(previous, 2)1 -15.732038   4.526575  -3.475  0.00051 ***
## poly(previous, 2)2   8.049464   3.229608   2.492  0.01269 *
## poly(campaign, 2)1 -12.349248   5.936024  -2.080  0.03749 *
## poly(campaign, 2)2  10.937572   6.001919   1.822  0.06840 .
## age                  0.011624   0.005552   2.094  0.03629 *
## cons.conf.idx        0.028578   0.011999   2.382  0.01724 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2594.9  on 3741  degrees of freedom
## Residual deviance: 1530.6  on 3731  degrees of freedom
## AIC: 1552.6
##
## Number of Fisher Scoring iterations: 7
```

```
Anova(gm5)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                    LR Chisq Df Pr(>Chisq)
## poly(duration, 2)   612.00  2  < 2.2e-16 ***
## nr.employed         255.09  1  < 2.2e-16 ***
## pdays                59.71  1  1.102e-14 ***
## poly(previous, 2)    16.12  2  0.0003158 ***
## poly(campaign, 2)    12.48  2  0.0019487 **
## age                   4.38  1  0.0364587 *
## cons.conf.idx         5.67  1  0.0172915 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

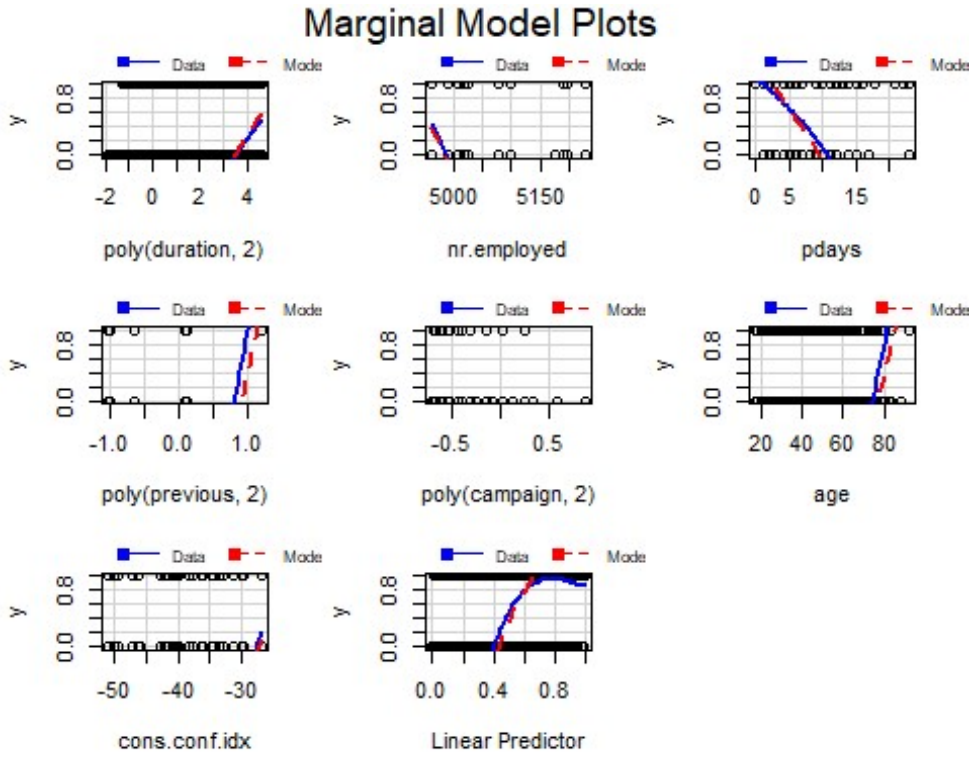
```
vif(gm5)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## poly(duration, 2) 1.377914  2        1.083442
```

```
## nr.employed       1.641804  1        1.281329
## pdays             1.808186  1        1.344688
## poly(previous, 2) 2.079898  2        1.200910
## poly(campaign, 2) 1.044713  2        1.010996
## age               1.037165  1        1.018413
## cons.conf.idx     1.060481  1        1.029796
```

```
marginalModelPlots(gm5)
```

```
## Warning in mmps(...): Splines and/or polynomials replaced by a fitted
## linear combination
```



Podemos ver que los p-values son inferiores a 0.1 para todas las variables, también vemos que el resultado del vif no presenta colinealidad.

Generalmente podemos ver que el modelo no se acerca tanto a los valores reales.

Modelización con variables explicativas numéricas y categóricas

```
# duration y f.duration
gm6<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
```

```r
gm7<-glm(y~
           f.duration +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
BIC(gm7,gm6)
```

```
##     df      BIC
## gm7 12 1847.037
## gm6 11 1621.054
```

```r
# pdays y f.pdays
gm8<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
gm9<-glm(y~
           poly(duration,2) +
            nr.employed +
            f.pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
BIC(gm9,gm8)
```

```
##     df      BIC
## gm9 11 1620.968
## gm8 11 1621.054
```

```r
# previous y f.previous
gm10<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
gm11<-glm(y~
```

```
           poly(duration,2) +
            nr.employed +
            pdays +
            f.previous +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
BIC(gm11,gm10)
```

```
##      df      BIC
## gm11 11 1621.220
## gm10 11 1621.054
```

```
# campaign vs f.campaign
gm12<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
gm13<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            f.campaign +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
BIC(gm13,gm12)
```

```
##      df      BIC
## gm13 11 1624.202
## gm12 11 1621.054
```

```
# age vs f.age
gm14<-glm(y~
           poly(duration,2) +
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
gm15<-glm(y~
           poly(duration,2) +
```

```
            nr.employed +
            pdays +
            poly(previous,2) +
            poly(campaign,2) +
            f.age +
            cons.conf.idx, family = binomial, data = dfw
           )
BIC(gm15,gm14)
```

```
##      df      BIC
## gm15 13 1630.939
## gm14 11 1621.054
```

A partir de los resultados de los BICs, nos quedamos con las versiones de las variables numéricas o de factores cuyo valor de BIC es menor.

Con el resultado obtenido anteriormente, creamos un nuevo modelo.

```
gm16<-glm(y~
           poly(duration,2) +
            nr.employed +
            f.pdays +
            poly(previous,2) +
            poly(campaign,2) +
            age +
            cons.conf.idx, family = binomial, data = dfw
           )
summary(gm16)
```

```
##
## Call:
## glm(formula = y ~ poly(duration, 2) + nr.employed + f.pdays +
##     poly(previous, 2) + poly(campaign, 2) + age + cons.conf.idx,
##     family = binomial, data = dfw)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8290  -0.3203  -0.1678  -0.1034   2.9250
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             77.867008   5.208078  14.951  < 2e-16 ***
## poly(duration, 2)1      82.135385   3.916635  20.971  < 2e-16 ***
## poly(duration, 2)2     -23.637806   3.097309  -7.632 2.32e-14 ***
## nr.employed             -0.015156   0.001018 -14.892  < 2e-16 ***
## f.pdaysf.pdays-(22,23]  -2.326819   0.314062  -7.409 1.27e-13 ***
## poly(previous, 2)1     -16.405010   4.606619  -3.561 0.000369 ***
## poly(previous, 2)2       9.381292   3.275937   2.864 0.004187 **
## poly(campaign, 2)1     -12.215658   5.905733  -2.068 0.038599 *
## poly(campaign, 2)2      10.952908   5.964041   1.836 0.066285 .
## age                      0.012244   0.005557   2.203 0.027560 *
```

```
## cons.conf.idx            0.028649   0.011975   2.392 0.016738 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2594.9  on 3741  degrees of freedom
## Residual deviance: 1530.5  on 3731  degrees of freedom
## AIC: 1552.5
##
## Number of Fisher Scoring iterations: 7
```

```
Anova(gm16)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                    LR Chisq Df Pr(>Chisq)
## poly(duration, 2)   608.02  2  < 2.2e-16 ***
## nr.employed         251.42  1  < 2.2e-16 ***
## f.pdays              59.79  1  1.055e-14 ***
## poly(previous, 2)    17.95  2  0.0001263 ***
## poly(campaign, 2)    12.34  2  0.0020925 **
## age                   4.85  1  0.0276839 *
## cons.conf.idx         5.72  1  0.0167883 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(gm16)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## poly(duration, 2) 1.374363  2        1.082743
## nr.employed       1.632340  1        1.277631
## f.pdays           1.911480  1        1.382563
## poly(previous, 2) 2.167088  2        1.213303
## poly(campaign, 2) 1.044674  2        1.010986
## age               1.037314  1        1.018486
## cons.conf.idx     1.057644  1        1.028418
```

Comprobamos el resultado y son correctos.

Ahora añadimos el resto de factores, utilizamos un catdes para ver cuales están más relacionadas con nuestro target.

```
catdes(dfw[,c("y",vars_cat)],1)
```

```
##
## Link between the cluster variable and the categorical variables (chi-s
quare test)
## ================================================================
===========
##                p.value df
```

```
## poutcome  8.905412e-84  2
## month     9.664852e-42  9
## job       5.444613e-17 10
## contact   2.379692e-14  1
## default   1.290603e-09  1
## marital   8.221254e-04  2
## housing   4.148800e-03  1
## education 8.424425e-03  6
##
## Description of each cluster by the categories
## =============================================
## $y.no
##                                          Cla/Mod    Mod/Cla     Global
## poutcome=poutcome.nonexistent          90.93168 87.9279279 86.0502405
## contact=contact.telephone              94.20074 38.0480480 35.9433458
## default=default.unknown                94.96855 22.6726727 21.2453234
## month=month.may                        92.54210 34.6546547 33.3244254
## job=job.blue-collar                    93.39513 24.2042042 23.0625334
## job=job.services                       94.67456  9.6096096  9.0326029
## marital=marital.married                90.30817 60.7207207 59.8343132
## education=education.basic.9y           92.32026 16.9669670 16.3548904
## housing=housing.no                     90.57259 47.0270270 46.2052378
## month=month.dec                        66.66667  0.3003003  0.4008552
## housing=housing.yes                    87.63040 52.9729730 53.7947622
## education=education.university.degree 86.43042 30.0300300 30.9192945
## marital=marital.single                 85.95506 27.5675676 28.5408872
## month=month.apr                        80.08850  5.4354354  6.0395510
## job=job.student                        70.65217  1.9519520  2.4585783
## month=month.sep                        62.74510  0.9609610  1.3629075
## job=job.retired                        74.56647  3.8738739  4.6231962
## month=month.mar                        54.16667  0.7807808  1.2827365
## default=default.no                     87.37699 77.3273273 78.7546766
## month=month.oct                        54.83871  1.0210210  1.6568680
## contact=contact.cellular               86.06592 61.9519520 64.0566542
## poutcome=poutcome.success              33.33333  1.1711712  3.1266702
##                                            p.value     v.test
## poutcome=poutcome.nonexistent          1.033863e-17   8.570110
## contact=contact.telephone              1.723437e-15   7.959781
## default=default.unknown                6.533397e-11   6.530995
## month=month.may                        4.627585e-07   5.041143
## job=job.blue-collar                    7.948561e-07   4.936626
## job=job.services                       1.615234e-04   3.772649
## marital=marital.married                1.803002e-03   3.120898
## education=education.basic.9y           2.923747e-03   2.975643
## housing=housing.no                     4.058668e-03   2.873566
## month=month.dec                        2.219854e-02  -2.286953
## housing=housing.yes                    4.058668e-03  -2.873566
## education=education.university.degree 9.941775e-04  -3.292169
## marital=marital.single                 2.417762e-04  -3.670817
## month=month.apr                        5.137216e-05  -4.049295
```

```
## job=job.student                           1.134551e-06  -4.866738
## month=month.sep                           8.387879e-07  -4.926119
## job=job.retired                           4.689017e-08  -5.462717
## month=month.mar                           9.910614e-10  -6.110843
## default=default.no                        6.533397e-11  -6.530995
## month=month.oct                           7.066133e-12  -6.856311
## contact=contact.cellular                  1.723437e-15  -7.959781
## poutcome=poutcome.success                 7.526687e-49 -14.689500
##
## $y.yes
##                                             Cla/Mod   Mod/Cla     Global
## poutcome=poutcome.success                 66.666667 18.932039  3.1266702
## contact=contact.cellular                  13.934084 81.067961 64.0566542
## month=month.oct                           45.161290  6.796117  1.6568680
## default=default.no                        12.623006 90.291262 78.7546766
## month=month.mar                           45.833333  5.339806  1.2827365
## job=job.retired                           25.433526 10.679612  4.6231962
## month=month.sep                           37.254902  4.611650  1.3629075
## job=job.student                           29.347826  6.553398  2.4585783
## month=month.apr                           19.911504 10.922330  6.0395510
## marital=marital.single                    14.044944 36.407767 28.5408872
## education=education.university.degree     13.569576 38.106796 30.9192945
## housing=housing.yes                       12.369598 60.436893 53.7947622
## month=month.dec                           33.333333  1.213592  0.4008552
## housing=housing.no                         9.427415 39.563107 46.2052378
## education=education.basic.9y               7.679739 11.407767 16.3548904
## marital=marital.married                    9.691827 52.669903 59.8343132
## job=job.services                           5.325444  4.368932  9.0326029
## job=job.blue-collar                        6.604867 13.834951 23.0625334
## month=month.may                            7.457899 22.572816 33.3244254
## default=default.unknown                    5.031447  9.708738 21.2453234
## contact=contact.telephone                  5.799257 18.932039 35.9433458
## poutcome=poutcome.nonexistent              9.068323 70.873786 86.0502405
##                                               p.value    v.test
## poutcome=poutcome.success                 7.526687e-49 14.689500
## contact=contact.cellular                  1.723437e-15  7.959781
## month=month.oct                           7.066133e-12  6.856311
## default=default.no                        6.533397e-11  6.530995
## month=month.mar                           9.910614e-10  6.110843
## job=job.retired                           4.689017e-08  5.462717
## month=month.sep                           8.387879e-07  4.926119
## job=job.student                           1.134551e-06  4.866738
## month=month.apr                           5.137216e-05  4.049295
## marital=marital.single                    2.417762e-04  3.670817
## education=education.university.degree     9.941775e-04  3.292169
## housing=housing.yes                       4.058668e-03  2.873566
## month=month.dec                           2.219854e-02  2.286953
## housing=housing.no                        4.058668e-03 -2.873566
## education=education.basic.9y              2.923747e-03 -2.975643
## marital=marital.married                   1.803002e-03 -3.120898
```

```
## job=job.services                          1.615234e-04 -3.772649
## job=job.blue-collar                       7.948561e-07 -4.936626
## month=month.may                           4.627585e-07 -5.041143
## default=default.unknown                   6.533397e-11 -6.530995
## contact=contact.telephone                 1.723437e-15 -7.959781
## poutcome=poutcome.nonexistent             1.033863e-17 -8.570110
```

Viendo el resultado del catdes, obtenemos que las variables que están más relacionadas son outcome, month, job, contact, default, marital, housing y education. Como month tiene muchos niveles decidimos usar el month factorizado.

```
gm17<-glm(y~poly(duration,2) +nr.employed +f.pdays +poly(previous,2) +pol
y(campaign,2) +age +cons.conf.idx+poutcome+ f.influentMonth + job+ contac
t+ default+ marital+ housing+ education, family = binomial, data = dfw)
Anova(gm17)

## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                     LR Chisq Df Pr(>Chisq)
## poly(duration, 2)     625.07  2  < 2.2e-16 ***
## nr.employed           177.78  1  < 2.2e-16 ***
## f.pdays                 0.02  1   0.895857
## poly(previous, 2)       1.21  2   0.546061
## poly(campaign, 2)       9.00  2   0.011102 *
## age                     6.51  1   0.010722 *
## cons.conf.idx           1.54  1   0.214577
## poutcome                9.22  2   0.009954 **
## f.influentMonth        12.92  2   0.001564 **
## job                    12.51 10   0.252449
## contact                 3.60  1   0.057727 .
## default                 5.75  1   0.016536 *
## marital                 4.60  2   0.100507
## housing                 3.43  1   0.063919 .
## education               5.17  6   0.522155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Seguimos cribando dado el resultado del Anova

```
gm18<-glm(y~poly(duration,2) +nr.employed +poly(campaign,2) +age +poutcom
e+ f.influentMonth + contact+ default+ housing, family = binomial, data =
dfw)
Anova(gm18)

## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                     LR Chisq Df Pr(>Chisq)
## poly(duration, 2)     623.25  2  < 2.2e-16 ***
## nr.employed           213.89  1  < 2.2e-16 ***
```

```
## poly(campaign, 2)     9.65  2   0.008020 **
## age                   8.49  1   0.003566 **
## poutcome             70.85  2  4.116e-16 ***
## f.influentMonth      20.46  2  3.614e-05 ***
## contact               3.80  1   0.051222 .
## default               8.33  1   0.003909 **
## housing               3.21  1   0.073099 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(gm18)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## poly(duration, 2) 1.447940  2        1.096952
## nr.employed       1.721167  1        1.311932
## poly(campaign, 2) 1.056114  2        1.013743
## age               1.033641  1        1.016681
## poutcome          1.296327  2        1.067035
## f.influentMonth   1.106236  2        1.025562
## contact           1.123197  1        1.059810
## default           1.075544  1        1.037084
## housing           1.010610  1        1.005291
```

Ahora las variables nos dan aceptables, con p-values menores a 0.1 y sin colinealidad.

Ahora hacemos un step con el criterio bayesiano, para validar el modelo

```
gm19<-step(gm18,k=log(nrow(dfw)))
```

```
## Start:  AIC=1601.84
## y ~ poly(duration, 2) + nr.employed + poly(campaign, 2) + age +
##     poutcome + f.influentMonth + contact + default + housing
##
##                      Df Deviance    AIC
## - poly(campaign, 2)  2   1496.3 1595.0
## - housing            1   1489.9 1596.8
## - contact            1   1490.5 1597.4
## <none>                   1486.7 1601.8
## - default            1   1495.0 1601.9
## - age                1   1495.1 1602.1
## - f.influentMonth    2   1507.1 1605.8
## - poutcome           2   1557.5 1656.2
## - nr.employed        1   1700.5 1807.5
## - poly(duration, 2)  2   2109.9 2208.6
##
## Step:  AIC=1595.03
## y ~ poly(duration, 2) + nr.employed + age + poutcome + f.influentMonth
+
##     contact + default + housing
##
##                      Df Deviance    AIC
```

```
## - housing            1   1499.3 1589.8
## - contact            1   1500.8 1591.3
## <none>                   1496.3 1595.0
## - age                1   1505.1 1595.6
## - default            1   1505.2 1595.7
## - f.influentMonth    2   1518.2 1600.4
## - poutcome           2   1567.5 1649.7
## - nr.employed        1   1717.9 1808.4
## - poly(duration, 2)  2   2117.1 2199.4
##
## Step:  AIC=1589.77
## y ~ poly(duration, 2) + nr.employed + age + poutcome + f.influentMonth
+
##     contact + default
##
##                     Df Deviance    AIC
## - contact            1   1504.1 1586.4
## <none>                   1499.3 1589.8
## - age                1   1507.6 1589.8
## - default            1   1508.4 1590.7
## - f.influentMonth    2   1522.0 1596.1
## - poutcome           2   1570.1 1644.2
## - nr.employed        1   1721.4 1803.7
## - poly(duration, 2)  2   2120.4 2194.5
##
## Step:  AIC=1586.4
## y ~ poly(duration, 2) + nr.employed + age + poutcome + f.influentMonth
+
##     default
##
##                     Df Deviance    AIC
## - age                1   1512.3 1586.3
## <none>                   1504.1 1586.4
## - default            1   1514.4 1588.5
## - f.influentMonth    2   1529.8 1595.6
## - poutcome           2   1573.6 1639.5
## - nr.employed        1   1747.3 1821.3
## - poly(duration, 2)  2   2128.2 2194.0
##
## Step:  AIC=1586.33
## y ~ poly(duration, 2) + nr.employed + poutcome + f.influentMonth +
##     default
##
##                     Df Deviance    AIC
## <none>                   1512.3 1586.3
## - default            1   1520.5 1586.3
## - f.influentMonth    2   1539.5 1597.1
## - poutcome           2   1582.0 1639.6
## - nr.employed        1   1767.7 1833.5
## - poly(duration, 2)  2   2136.3 2193.9
```

```
summary(gm19)
```

```
##
## Call:
## glm(formula = y ~ poly(duration, 2) + nr.employed + poutcome +
##     f.influentMonth + default, family = binomial, data = dfw)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6811  -0.3085  -0.1597  -0.1003   3.0735
##
## Coefficients:
##                                                Estimate Std. Error
## (Intercept)                                    75.15851    5.20488
## poly(duration, 2)1                             83.77127    3.98039
## poly(duration, 2)2                            -24.63315    3.12149
## nr.employed                                    -0.01528    0.00103
## poutcomepoutcome.nonexistent                    0.89824    0.22100
## poutcomepoutcome.success                        2.40968    0.30331
## f.influentMonthf.influentMonth.sep-may-jul     -0.68808    0.15291
## f.influentMonthf.influentMonth.mar-dec-oct-nov  0.06799    0.19438
## defaultdefault.unknown                         -0.57789    0.20843
##                                               z value Pr(>|z|)
## (Intercept)                                    14.440  < 2e-16 ***
## poly(duration, 2)1                             21.046  < 2e-16 ***
## poly(duration, 2)2                             -7.891 2.99e-15 ***
## nr.employed                                   -14.831  < 2e-16 ***
## poutcomepoutcome.nonexistent                    4.064 4.81e-05 ***
## poutcomepoutcome.success                        7.945 1.95e-15 ***
## f.influentMonthf.influentMonth.sep-may-jul     -4.500 6.80e-06 ***
## f.influentMonthf.influentMonth.mar-dec-oct-nov  0.350  0.72651
## defaultdefault.unknown                         -2.773  0.00556 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2594.9  on 3741  degrees of freedom
## Residual deviance: 1512.3  on 3733  degrees of freedom
## AIC: 1530.3
##
## Number of Fisher Scoring iterations: 7
```

Hay que ver que todos los coeficientes sean calculables y que no tengamos ningún NA en el summary, en nuestro caso no tenemos ninguno.

Interacciones

Primero probamos con todas las interacciones posibles de orden 2 para hacernos una idea de las interacciones que podemos usar de muestra.

```
gm20<-glm(y~ (poly(duration,2) +nr.employed +poly(campaign,2) +age +poutc
ome+ f.influentMonth + contact+ default+ housing)^2, family = binomial, d
ata = dfw)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
Anova(gm20)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                                      LR Chisq Df Pr(>Chisq)
## poly(duration, 2)                      606.44  2  < 2.2e-16 ***
## nr.employed                            219.61  1  < 2.2e-16 ***
## poly(campaign, 2)                       11.93  2  0.0025650 **
## age                                      7.41  1  0.0064886 **
## poutcome                                63.00  2  2.083e-14 ***
## f.influentMonth                         16.74  2  0.0002317 ***
## contact                                  6.26  1  0.0123735 *
## default                                  8.37  1  0.0038152 **
## housing                                  4.99  1  0.0255125 *
## poly(duration, 2):nr.employed           12.60  2  0.0018372 **
## poly(duration, 2):poly(campaign, 2)      2.17  4  0.7042833
## poly(duration, 2):age                    0.20  2  0.9040617
## poly(duration, 2):poutcome               3.63  4  0.4578341
## poly(duration, 2):f.influentMonth       16.14  4  0.0028348 **
## poly(duration, 2):contact                0.70  2  0.7038259
## poly(duration, 2):default                1.43  2  0.4886817
## poly(duration, 2):housing                6.20  2  0.0450133 *
## nr.employed:poly(campaign, 2)            2.43  2  0.2968249
## nr.employed:age                          0.00  1  0.9476946
## nr.employed:poutcome                     5.67  2  0.0588327 .
## nr.employed:f.influentMonth              0.84  2  0.6558761
## nr.employed:contact                      0.01  1  0.9381053
## nr.employed:default                      0.48  1  0.4868526
## nr.employed:housing                      0.02  1  0.8895965
## poly(campaign, 2):age                    8.90  2  0.0116869 *
## poly(campaign, 2):poutcome              13.38  4  0.0095456 **
## poly(campaign, 2):f.influentMonth       18.34  4  0.0010595 **
## poly(campaign, 2):contact                1.19  2  0.5509442
```

```
## poly(campaign, 2):default                0.06  2  0.9705383
## poly(campaign, 2):housing                3.48  2  0.1755176
## age:poutcome                             1.69  2  0.4304841
## age:f.influentMonth                      0.46  2  0.7937762
## age:contact                              0.15  1  0.6986240
## age:default                              0.83  1  0.3622380
## age:housing                              3.63  1  0.0567390 .
## poutcome:f.influentMonth                 7.12  4  0.1297656
## poutcome:contact                         3.67  2  0.1594200
## poutcome:default                         3.36  2  0.1866776
## poutcome:housing                         1.76  2  0.4153831
## f.influentMonth:contact                  9.26  2  0.0097760 **
## f.influentMonth:default                  4.77  2  0.0920652 .
## f.influentMonth:housing                  2.12  2  0.3465305
## contact:default                          0.64  1  0.4233159
## contact:housing                          0.30  1  0.5831223
## default:housing                          3.33  1  0.0678341 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Elegiremos una interacción factor por factor y factor por numérica de entre todas las interacciones, cuyos p-values son los más convincentes. Comprobamos la bondad de la interacción factor

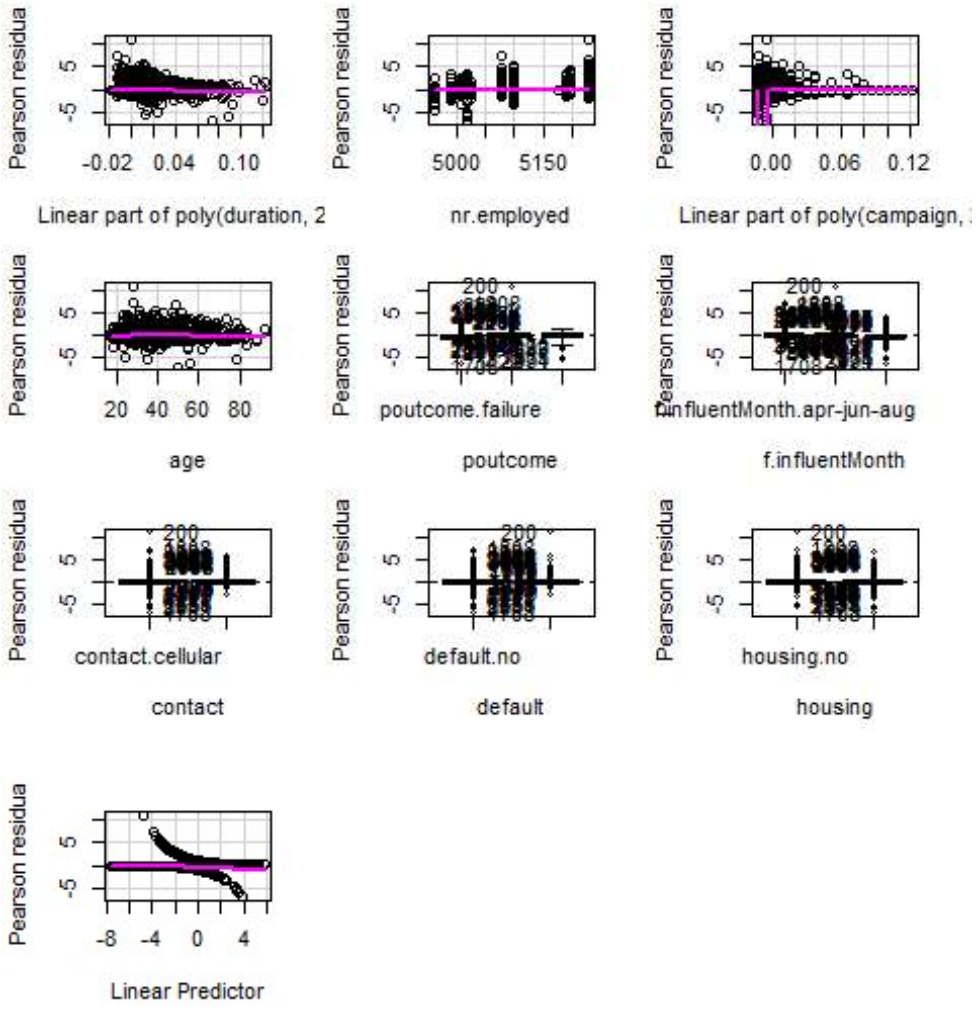
```
gm21<-glm(y~ poly(duration,2) +nr.employed +poly(campaign,2) +age +poutco
me+ f.influentMonth*contact+ default+ housing, family = binomial, data =
dfw)

gm22<-glm(y~ poly(duration,2) +nr.employed +poly(campaign,2)*f.influentMo
nth  +age +poutcome+ contact+ default+ housing, family = binomial, data =
dfw)

BIC(gm21,gm20)

##      df      BIC
## gm21 16 1613.523
## gm20 88 2052.909

BIC(gm22,gm20)

##      df      BIC
## gm22 18 1618.316
## gm20 88 2052.909
```

Vemos que los dos modelos con interacciones dan mejor que nuestro modelo, entre estos vemos que el de menor BIC es el de la interacción de f.influentMonth*contact.

Validación

Para la validación analizamos los gráficos

```
residualPlots(gm21)
```

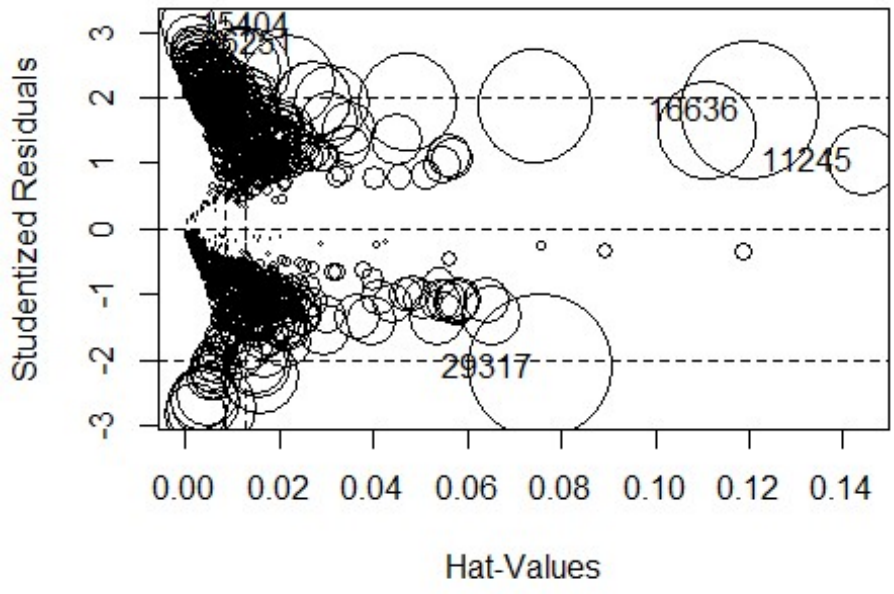



```
##                     Test stat Pr(>|Test stat|)
## poly(duration, 2)
```

```
## nr.employed           0.8698          0.35102
## poly(campaign, 2)
## age                   3.4946          0.06157 .
## poutcome
## f.influentMonth
## contact
## default
## housing
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Con el residualPlots, podemos ver que tenemos una observación en común que es muy influente, como ahora vamos a hacer el influencePlot, podremos determinar si efectivamente esta observación es demasiado influyente.

```
influencePlot(gm21)
```




```
##         StudRes          Hat       CookD
## 15404  3.116369 0.0006512188 0.004991063
## 36251  2.835739 0.0015232026 0.005018080
## 11245  1.043963 0.1440726950 0.007802044
## 29317 -2.103560 0.0758185513 0.033909413
## 16636  1.817313 0.1200968043 0.030955448
```

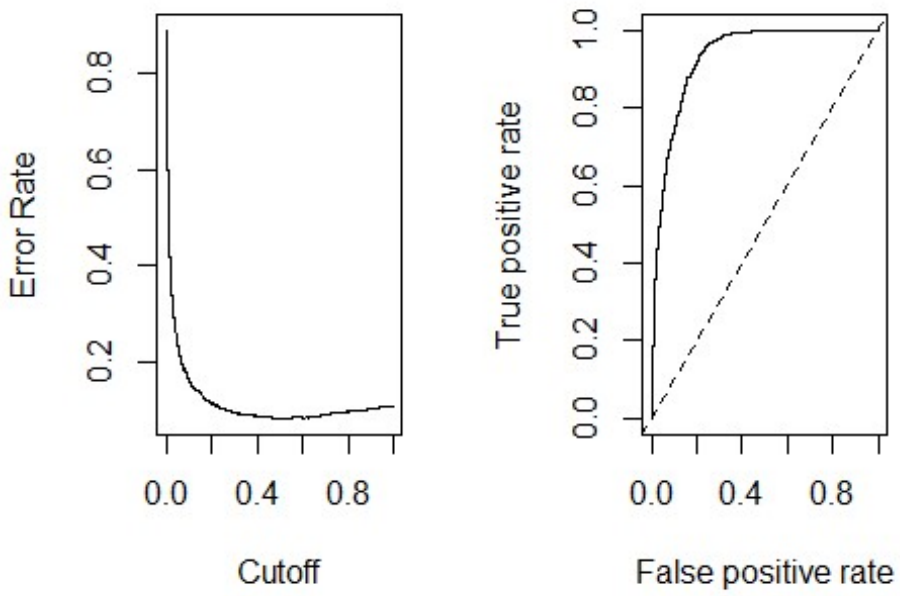
```
which(row.names(df)==11245)
```

```
## [1] 1317
```

```
which(row.names(df)==16636)

## [1] 1940

which(row.names(df)==29317)

## [1] 3498
```

Viendo el resultado del influentPlot, no vemos al individuo 200, que nos sale en la gráfica de residuos, lo que nos puede decir que no influye demasiado en nuestro modelo.

```
dataroc<-prediction(predict(gm21, type="response"),dfw$y)
par(mfrow=c(1,2))
plot(performance(dataroc,"err"))
plot(performance(dataroc,"tpr","fpr"))
abline(0,1,lty=2)
```



Estamos cogiendo las betas de este modelo y aplicándolos a las variables explicativas del dft, para así obtener las predicciones según nuestro modelo. Montamos una tabla con las predicciones y los datos reales a modo de matriz de confusión, del cual su diagonal nos indica la cantidad de aciertos.

```
p<-factor(ifelse(predict(gm21, dft, type = "response") < 0.4, 0, 1 ))
tabConfusion<-table(p, dft[,"y"])
```

Para calcular la capacidad predictiva del modelo, bastará con sumar la diagonal de la matriz de confusión y dividirla entre el número de observaciones.

```
capacidadPredictiva <- (tabConfusion[1,1] + tabConfusion[2,2])/nrow(dft)
```

Tenemos un 91,42% de aciertos con nuestro modelo.

Nos damos cuenta que por los datos que tenemos no es posible que tengamos una capacidad predictiva tan grande, por lo que decidimos comparar con el modelo null.

```
gmnull<-glm(y~ 1, family = binomial, data = dfw)
pnull<-factor(ifelse(predict(gmnull, dft, type = "response") < 0.4, 0, 1
))
tabConfusionNull<-table(pnull, dft[,"y"])
capacidadPredictivaNull <- (tabConfusionNull[1,1] + 0)/nrow(dft)
```

Con el modelo Null tenemos un 89,58% de aciertos, ahora viendo la diferencia entre nuestro modelo y el null tenemos que

```
MejoraModelo <- capacidadPredictiva - capacidadPredictivaNull
MejoraModelo*100

## [1] 1.842949
```

Tenemos que nuestro modelo es 1.84% mejor que el modelo más básico. El hecho de que la capacidad predictiva sea tan alta en ambos casos, es debido a que la gran mayoría de las observaciones tienen como valor de respuesta “no”, esto hace que cualquier modelo por tonto que sea tenga una buena capacidad predictiva.