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.

# chunk 10  
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

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:

# chunk 20  
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

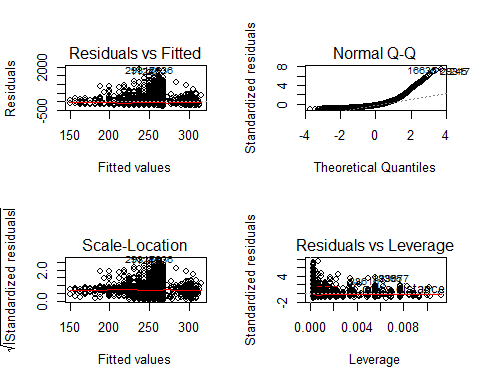
m3<-lm(duration~campaign+pdays,data=df)  
summary(m3)

##   
## 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):

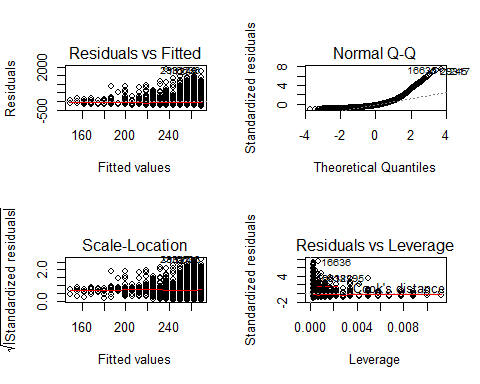
# chunk 40  
m5<-step(m,k=log(nrow(df)))

## 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.

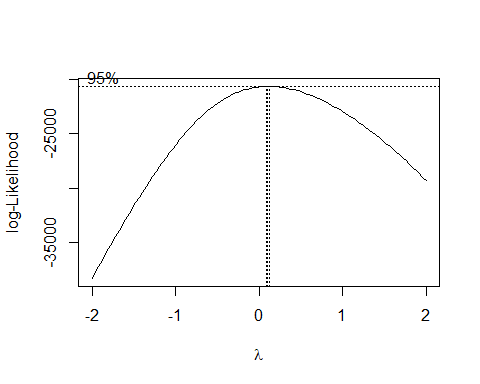
En este caso no podemos hacer el análisis de residuos con vif porque solo tenemos 1 variable.

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.

# chunk 50  
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.

# chunk 60  
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

# chunk 70  
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

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.

# chunk 80  
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:

# chunk 90  
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

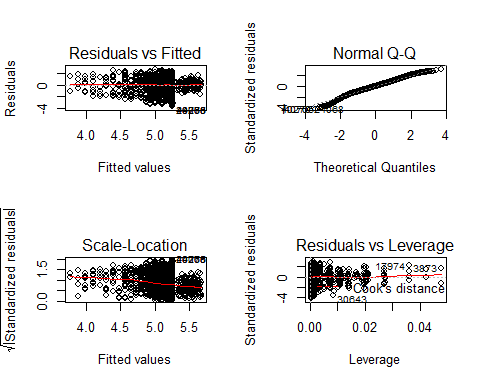
Anova(m20)

## 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.

# chunk 110  
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 númerico, 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-jun-aug","sep-may-jul","mar-dec-oct-nov"))  
levels(df$f.influentMonth)<-paste0("f.influentMonth.",levels(df$f.influentMonth)) # 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 cual es mejor.

# chunk 120  
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

# chunk 130  
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

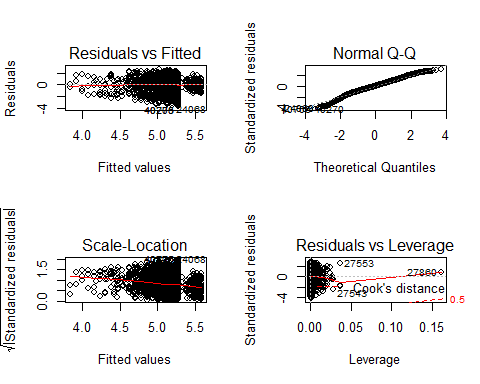
m32<-lm(log(duration)~(f.influentMonth\*f.pdays+campaign),data=df)  
Anova(m32)

## 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

# chunk 140  
par(mfrow=c(2,2))  
plot(m31)

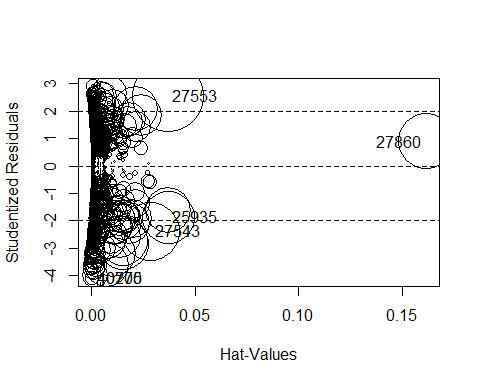


par(mfrow=c(1,1))

Analizando los gráficos:  
- Residual VS Fitted. En este gráfico muestra los residuos de los valores predecidos. Lo deseable es que los puntos estén uniformemente dispersos, para poderlo contrastar el gráfico esta 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 replica del primer plot.

* Residuals Vs Leverage. #preguntar otra vez…. preguntar, no vemos las curvas de nivel en el grafico, y como se ven reflejadas las observaciones influyentes, y que significa que el leverage sea el factor de anclaje…. lo podemos mirar en las transparencias de teoria 5 4 o 3 veces la mediana del leverage, hay que dibujarlo o valorarlo de alguna maeraa , hacer el influence polot que nos dara la distancia de coolk(ya esta hecho)

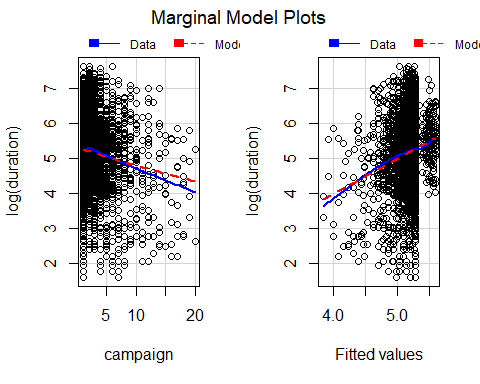
#chunk 150  
influencePlot(m31)



## 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

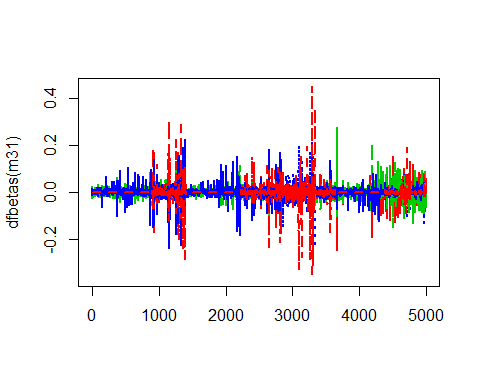
which(row.names(df)==27553)

## [1] 3293

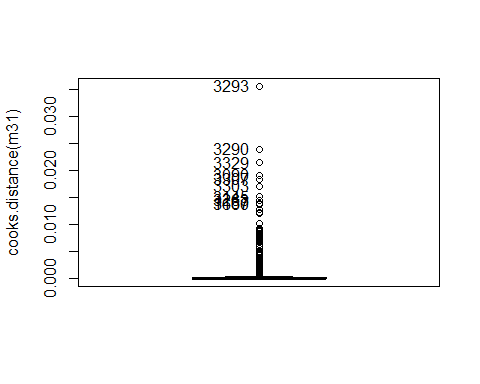
* InfluenPLot. Nos muestra las individuos más influentes, esto se puede ver gráficamente a través del radio de las circunferencias. En nuestro caso, viendo el gráfico podemos ver que hay individuos bastante influyentes, el 3329 y 3293 que para nuestra muestra serian los individuos.
* MarginalModelPlot. 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 nuestros dis gráficos #preguntar, solo se puede usar para variabls numericas?? podemos usarlo aqui donde la unica variable numerica es campaign??

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

# chunk 160  
matplot(dfbetas(m31), type="l", col=2:4,lwd=2)



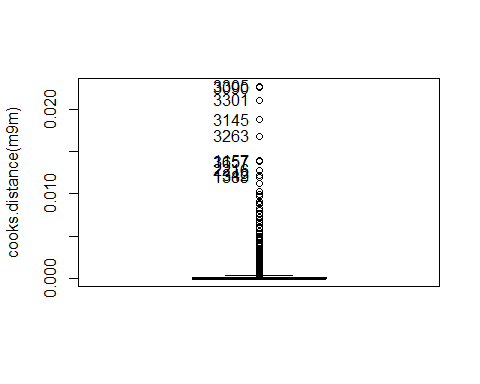
Boxplot(cooks.distance(m31))



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

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

#chunk 170   
# Sin el individuo que más afecta  
  
m9m<-lm(log(duration)~(f.influentMonth\*campaign+f.pdays),data=df[c(-3293,-3290,-3329),]) # ACORDARNOS DE MOVERLO AL FINAL - se deberia eliminar al final de todo no aqui, ya que al final tendremos todos los datos  
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 value  
## (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 value  
## (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.

# chunk 150  
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 númericas para ver cuales son las que están más relacionadas con nuestro target. Las ultilizamos para hacer un modelo lineal general con variables explicativas numéricas. Este modelo es de la familia binomial ya que nuestro target es binario.

# chunk 160  
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.

# chunk 170  
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 decidmos sacarla de nuestro modelo.

# chunk 180  
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 con transformaciones

# chunk 190

### Modelo de regresión polinómica

## Modelización con variables explicativas numéricas y categóricas

## Interacciones

## Validación

# hacer tablas de confusion para el targer binario, y las predicciones. Para el numerico no hay qu ehacer predicciones