Deliverable 3

Guillem Valls, Sergio Mazzariol

Table of Contents

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. #preguntar lo contrastamos con el akaike o nuestro resultado del modelo usamos akaike.

# 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: #preguntar diferencia summary y Anova, sabemos que el Anova nos dice para cada variable la probabilidad de que el modelo sea igual con o sin esta variable mediante el p-value, entonces que nos dice el summary de diferente?

# 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

Anova(m2)

## Anova Table (Type II tests)  
##   
## Response: duration  
## Sum Sq Df F value Pr(>F)   
## campaign 1095005 1 17.3600 3.145e-05 \*\*\*  
## pdays 297535 1 4.7171 0.02991 \*   
## previous 89459 1 1.4183 0.23375   
## Residuals 314498351 4986   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

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

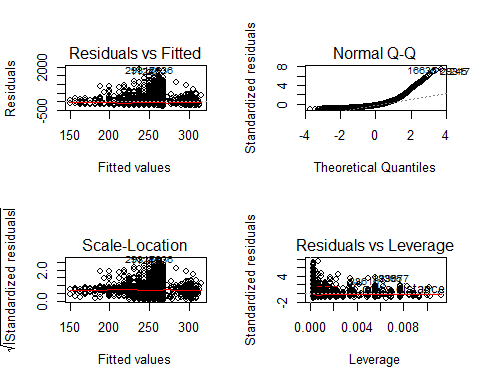
Anova(m3)

## Anova Table (Type II tests)  
##   
## Response: duration  
## Sum Sq Df F value Pr(>F)   
## campaign 1058016 1 16.7722 4.281e-05 \*\*\*  
## pdays 208524 1 3.3056 0.0691 .   
## Residuals 314587810 4987   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

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.

Ahora procederemos a comprobar el resultado usando la función step, conocida como Akaike o BIC.

# chunk 30  
m4<-step(m)

## Start: AIC=55153.4  
## duration ~ campaign + pdays  
##   
## Df Sum of Sq RSS AIC  
## <none> 314587810 55153  
## - pdays 1 208524 314796334 55155  
## - campaign 1 1058016 315645826 55168

Al aplicar el metodo step, vemos que no hace niguna criba más y se queda con el mismo modelo que ya nosotros habiamos cribado.

Si probamos con la versión bayesiana (del BIC):

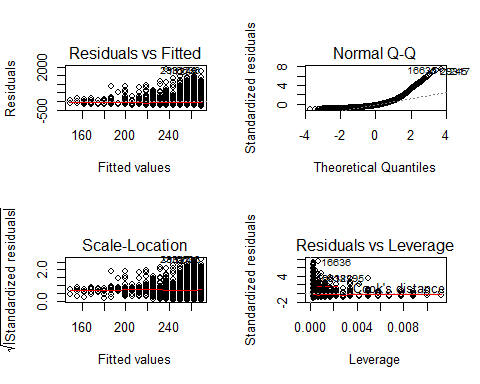
# chunk 40  
m5<-step(m,k=log(nrow(df)))

## 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))

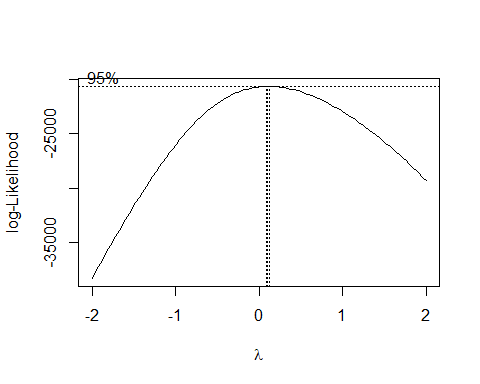
La versión bayesiana es conveniente usarla en casos de muestras grandes. En este caso vemos que se queda con una sola variable (campaign), ya que en el primer step del volcado vemos que sin la variable p-days el valor AIC, en este caso BIC, es menor.

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.

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

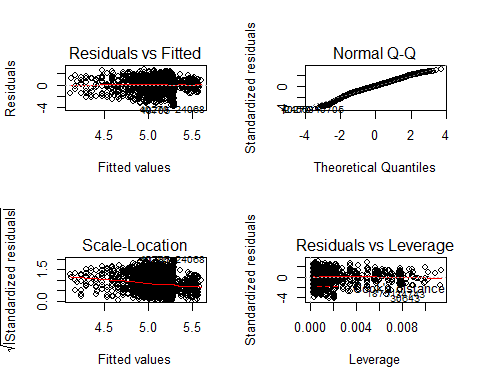
# chunk 70  
m7<-lm(log(duration)~campaign+pdays+nr.employed,data=df)  
summary(m7)

##   
## 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

par(mfrow=c(2,2))  
plot(m7)



par(mfrow=c(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

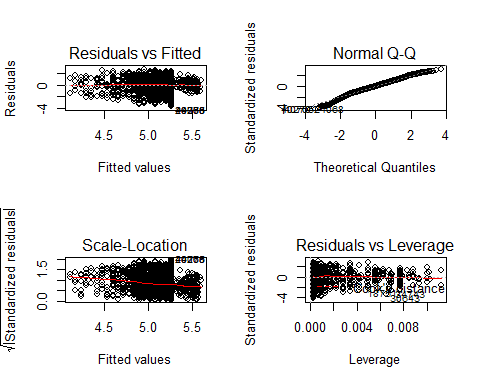
Anova(m9)

## 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

par(mfrow=c(2,2))  
plot(m9)



par(mfrow=c(1,1))

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.

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

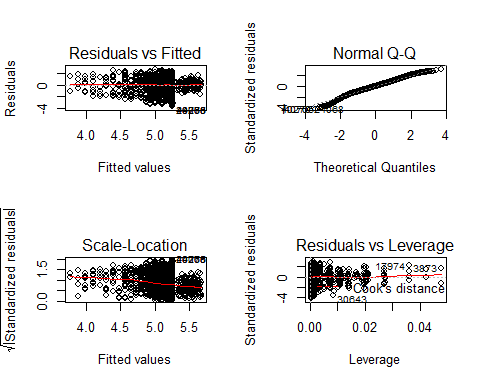
Anova(m20)

## 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

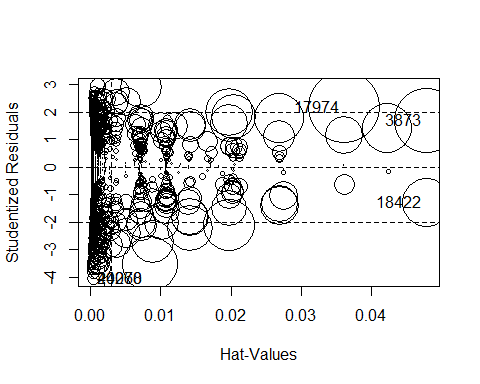
vif(m20)

## 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))  
influencePlot(m20)



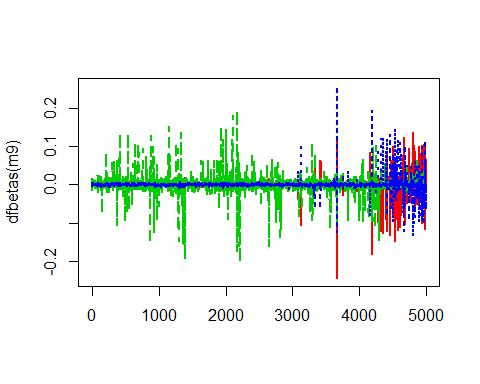
## StudRes Hat CookD  
## 3873 1.691165 0.0477252396 0.028656662  
## 17974 2.176520 0.0361042506 0.035461589  
## 18422 -1.290498 0.0477252396 0.016690616  
## 24068 -4.046059 0.0003678982 0.001201282  
## 40270 -4.046059 0.0003678982 0.001201282

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. En nuestro caso el resultado 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, es una recta bastante 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.  
- Residuals Vs Leverage. # 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. - 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 no hay niguna que sea excesivamente influente.

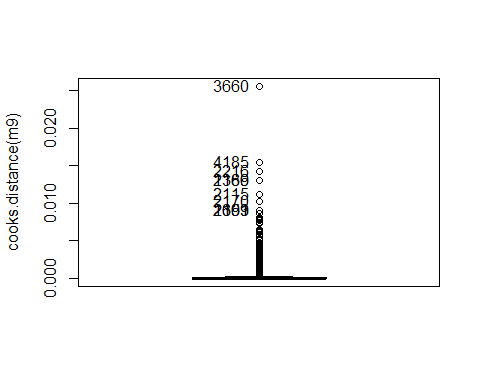
# preguntar practica en influent data, para estar seguros que el tema del diagnostico, hacer diagnotistico, residual plot, marginal plot, influence plot, porque en algun moemtno hay que hacerlo, se ha explicado esta semana.

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

# chunk 100  
matplot(dfbetas(m9), type="l", col=2:4,lwd=2)



Boxplot(cooks.distance(m9)) #preguntar nos sale el 3660 pero en el grafico anterior no sale, porque? y si hay que quitarlo al final, porque al final?



## [1] 3660 4185 2216 1389 2166 2115 2170 1361 2199 2653

Consideramos que hay un individuo que repercute demasiado en los datos, aún así no lo eliminaremos hasta el final.

# chunk 110  
vars\_cat\_total = c(vars\_cat, names(df[,22:29])) # creamos una vars\_cat con las categorias factorizadas  
condes(df[,c("duration",vars\_cat\_total)],1, proba= 0.01)

## $quali  
## R2 p.value  
## f.duration 0.621168787 0.000000e+00  
## f.campaign 0.003783221 7.858324e-05  
## month 0.004450289 8.185248e-03  
##   
## $category  
## Estimate p.value  
## f.duration-(300,2.1e+03] 310.35106 0.000000e+00  
## f.campaign-(1,2] 23.01041 3.895001e-05  
## month.apr 35.25783 4.865526e-03  
## f.season.Mar-May 13.19170 6.782891e-03  
## month.aug -25.22225 7.943838e-03  
## f.campaign-(2,20] -17.35164 3.316706e-03  
## f.duration-(180,300] -20.50721 3.927333e-04  
## f.duration-(120,180] -106.75355 5.404997e-53  
## f.duration-[5,120] -183.09030 1.278559e-312

Al hacer condes, con todas las variables categóricas, contemplamos el uso de f.campaign, 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.

Contrastamos un modelo con campaign o con f.campaign

# chunk 130  
m22<-lm(log(duration)~campaign+f.pdays+month,data=df)  
m23<-lm(log(duration)~f.pdays+f.campaign+month,data=df)  
BIC(m23,m22)

## df BIC  
## m23 14 13254.01  
## m22 13 13191.59

# 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 99.5 1 123.2361 < 2.2e-16 \*\*\*  
## f.pdays 15.3 1 18.9525 1.367e-05 \*\*\*  
## month 23.5 9 3.2361 0.0006331 \*\*\*  
## Residuals 4018.8 4978   
## ---  
## 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.

# chunk 140  
m30<-lm(log(duration)~(campaign+f.pdays+month)^2,data=df) # preguntar que mejora se tiene con este modelo?  
summary(m30)

##   
## Call:  
## lm(formula = log(duration) ~ (campaign + f.pdays + month)^2,   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.6828 -0.5449 -0.0128 0.5826 2.5527   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 5.912653 0.279144 21.181  
## campaign -0.225906 0.065339 -3.457  
## f.pdaysf.pdays-(22,23] -0.316299 0.279207 -1.133  
## monthmonth.aug -0.247035 0.308347 -0.801  
## monthmonth.dec -0.747663 0.801963 -0.932  
## monthmonth.jul -0.076056 0.414552 -0.183  
## monthmonth.jun -0.205770 0.389347 -0.529  
## monthmonth.mar 0.221230 0.395412 0.559  
## monthmonth.may -0.251007 0.308110 -0.815  
## monthmonth.nov -0.135551 0.322511 -0.420  
## monthmonth.oct -0.201971 0.370164 -0.546  
## monthmonth.sep -0.215399 0.388286 -0.555  
## campaign:f.pdaysf.pdays-(22,23] 0.091302 0.055811 1.636  
## campaign:monthmonth.aug 0.056515 0.038798 1.457  
## campaign:monthmonth.dec 0.255706 0.167304 1.528  
## campaign:monthmonth.jul 0.093448 0.037956 2.462  
## campaign:monthmonth.jun 0.070730 0.038691 1.828  
## campaign:monthmonth.mar 0.128904 0.057745 2.232  
## campaign:monthmonth.may 0.073203 0.037731 1.940  
## campaign:monthmonth.nov 0.060346 0.044161 1.366  
## campaign:monthmonth.oct 0.110221 0.120385 0.916  
## campaign:monthmonth.sep 0.218898 0.093449 2.342  
## f.pdaysf.pdays-(22,23]:monthmonth.aug -0.046059 0.308513 -0.149  
## f.pdaysf.pdays-(22,23]:monthmonth.dec 0.147362 0.732583 0.201  
## f.pdaysf.pdays-(22,23]:monthmonth.jul -0.186906 0.414799 -0.451  
## f.pdaysf.pdays-(22,23]:monthmonth.jun -0.142056 0.390318 -0.364  
## f.pdaysf.pdays-(22,23]:monthmonth.mar -0.547080 0.408547 -1.339  
## f.pdaysf.pdays-(22,23]:monthmonth.may 0.005922 0.307503 0.019  
## f.pdaysf.pdays-(22,23]:monthmonth.nov -0.186400 0.320947 -0.581  
## f.pdaysf.pdays-(22,23]:monthmonth.oct -0.315532 0.347332 -0.908  
## f.pdaysf.pdays-(22,23]:monthmonth.sep -0.431189 0.366706 -1.176  
## Pr(>|t|)   
## (Intercept) < 2e-16 \*\*\*  
## campaign 0.00055 \*\*\*  
## f.pdaysf.pdays-(22,23] 0.25733   
## monthmonth.aug 0.42308   
## monthmonth.dec 0.35123   
## monthmonth.jul 0.85444   
## monthmonth.jun 0.59718   
## monthmonth.mar 0.57585   
## monthmonth.may 0.41530   
## monthmonth.nov 0.67429   
## monthmonth.oct 0.58535   
## monthmonth.sep 0.57910   
## campaign:f.pdaysf.pdays-(22,23] 0.10192   
## campaign:monthmonth.aug 0.14528   
## campaign:monthmonth.dec 0.12648   
## campaign:monthmonth.jul 0.01385 \*   
## campaign:monthmonth.jun 0.06760 .   
## campaign:monthmonth.mar 0.02564 \*   
## campaign:monthmonth.may 0.05242 .   
## campaign:monthmonth.nov 0.17185   
## campaign:monthmonth.oct 0.35994   
## campaign:monthmonth.sep 0.01920 \*   
## f.pdaysf.pdays-(22,23]:monthmonth.aug 0.88133   
## f.pdaysf.pdays-(22,23]:monthmonth.dec 0.84059   
## f.pdaysf.pdays-(22,23]:monthmonth.jul 0.65230   
## f.pdaysf.pdays-(22,23]:monthmonth.jun 0.71591   
## f.pdaysf.pdays-(22,23]:monthmonth.mar 0.18060   
## f.pdaysf.pdays-(22,23]:monthmonth.may 0.98464   
## f.pdaysf.pdays-(22,23]:monthmonth.nov 0.56141   
## f.pdaysf.pdays-(22,23]:monthmonth.oct 0.36369   
## f.pdaysf.pdays-(22,23]:monthmonth.sep 0.23971   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8981 on 4959 degrees of freedom  
## Multiple R-squared: 0.03867, Adjusted R-squared: 0.03286   
## F-statistic: 6.65 on 30 and 4959 DF, p-value: < 2.2e-16

coef(m30) # preguntar que hace el coef

## (Intercept)   
## 5.912653137   
## campaign   
## -0.225906350   
## f.pdaysf.pdays-(22,23]   
## -0.316298985   
## monthmonth.aug   
## -0.247035382   
## monthmonth.dec   
## -0.747663024   
## monthmonth.jul   
## -0.076056314   
## monthmonth.jun   
## -0.205769839   
## monthmonth.mar   
## 0.221229884   
## monthmonth.may   
## -0.251006930   
## monthmonth.nov   
## -0.135550598   
## monthmonth.oct   
## -0.201970567   
## monthmonth.sep   
## -0.215398873   
## campaign:f.pdaysf.pdays-(22,23]   
## 0.091302471   
## campaign:monthmonth.aug   
## 0.056515403   
## campaign:monthmonth.dec   
## 0.255705643   
## campaign:monthmonth.jul   
## 0.093447954   
## campaign:monthmonth.jun   
## 0.070730159   
## campaign:monthmonth.mar   
## 0.128904421   
## campaign:monthmonth.may   
## 0.073202782   
## campaign:monthmonth.nov   
## 0.060345875   
## campaign:monthmonth.oct   
## 0.110221261   
## campaign:monthmonth.sep   
## 0.218898359   
## f.pdaysf.pdays-(22,23]:monthmonth.aug   
## -0.046059266   
## f.pdaysf.pdays-(22,23]:monthmonth.dec   
## 0.147361727   
## f.pdaysf.pdays-(22,23]:monthmonth.jul   
## -0.186906070   
## f.pdaysf.pdays-(22,23]:monthmonth.jun   
## -0.142055763   
## f.pdaysf.pdays-(22,23]:monthmonth.mar   
## -0.547079553   
## f.pdaysf.pdays-(22,23]:monthmonth.may   
## 0.005922296   
## f.pdaysf.pdays-(22,23]:monthmonth.nov   
## -0.186400016   
## f.pdaysf.pdays-(22,23]:monthmonth.oct   
## -0.315532317   
## f.pdaysf.pdays-(22,23]:monthmonth.sep   
## -0.431188964

m31<-step(m30,k=log(nrow(df)))

## Start: AIC=-839.45  
## log(duration) ~ (campaign + f.pdays + month)^2  
##   
## Df Sum of Sq RSS AIC  
## - f.pdays:month 9 4.3590 4004.4 -910.65  
## - campaign:month 9 12.2844 4012.3 -900.79  
## - campaign:f.pdays 1 2.1587 4002.2 -845.27  
## <none> 4000.1 -839.45  
##   
## Step: AIC=-910.65  
## log(duration) ~ campaign + f.pdays + month + campaign:f.pdays +   
## campaign:month  
##   
## Df Sum of Sq RSS AIC  
## - campaign:month 9 12.232 4016.7 -972.07  
## - campaign:f.pdays 1 2.403 4006.8 -916.18  
## <none> 4004.4 -910.65  
##   
## Step: AIC=-972.07  
## log(duration) ~ campaign + f.pdays + month + campaign:f.pdays  
##   
## Df Sum of Sq RSS AIC  
## - month 9 23.513 4040.2 -1019.58  
## - campaign:f.pdays 1 2.139 4018.8 -977.93  
## <none> 4016.7 -972.07  
##   
## Step: AIC=-1019.58  
## log(duration) ~ campaign + f.pdays + campaign:f.pdays  
##   
## Df Sum of Sq RSS AIC  
## - campaign:f.pdays 1 2.1386 4042.3 -1025.5  
## <none> 4040.2 -1019.6  
##   
## Step: AIC=-1025.46  
## log(duration) ~ campaign + f.pdays  
##   
## Df Sum of Sq RSS AIC  
## <none> 4042.3 -1025.46  
## - f.pdays 1 13.564 4055.9 -1017.26  
## - campaign 1 100.363 4142.7 -911.59

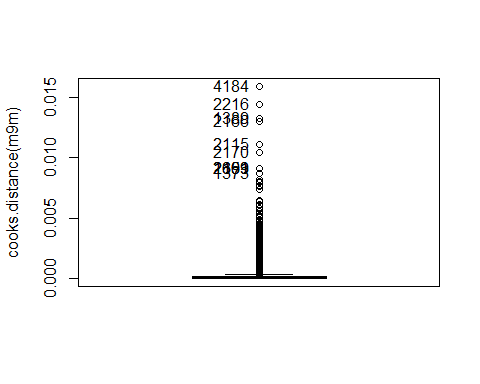
Anova(m31)

## Anova Table (Type II tests)  
##   
## Response: log(duration)  
## Sum Sq Df F value Pr(>F)   
## campaign 100.4 1 123.818 < 2.2e-16 \*\*\*  
## f.pdays 13.6 1 16.734 4.369e-05 \*\*\*  
## Residuals 4042.3 4987   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(m31,m30) # Fisher test - Priority to BIC criteria

## Analysis of Variance Table  
##   
## Model 1: log(duration) ~ campaign + f.pdays  
## Model 2: log(duration) ~ (campaign + f.pdays + month)^2  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 4987 4042.3   
## 2 4959 4000.1 28 42.243 1.8704 0.003607 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# chunk 150  
# Sin el individuo que más afecta  
 m9m<-lm(log(duration)~campaign+pdays,data=df[-3660,]) # 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] 4184 2216 1389 2166 2115 2170 2199 2653 1361 1373

coef(m9m)

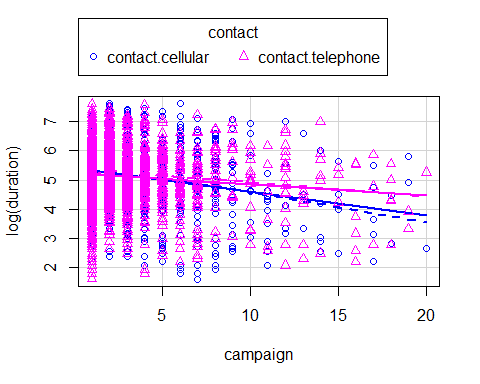
## (Intercept) campaign pdays   
## 5.68249440 -0.05984131 -0.01573543

# Interactions between numeric variables and factors

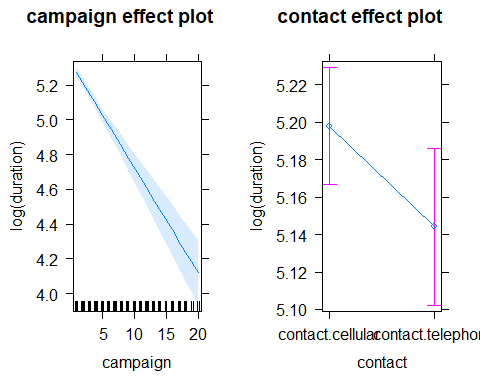
# Exemple adhoc: Y ~X+A  
m40<-lm(log(duration)~campaign+contact,data=df)  
summary(m40)

##   
## Call:  
## lm(formula = log(duration) ~ campaign + contact, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.6263 -0.5618 -0.0040 0.5899 2.6520   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.350108 0.020410 262.131 <2e-16 \*\*\*  
## campaign -0.060490 0.005434 -11.131 <2e-16 \*\*\*  
## contactcontact.telephone -0.053867 0.026737 -2.015 0.044 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9015 on 4987 degrees of freedom  
## Multiple R-squared: 0.02605, Adjusted R-squared: 0.02566   
## F-statistic: 66.7 on 2 and 4987 DF, p-value: < 2.2e-16

# Suport visual  
scatterplot(log(duration)~campaign|contact,data=df)



# Interpretation of models through effects library  
library(effects)  
plot(allEffects(m40))



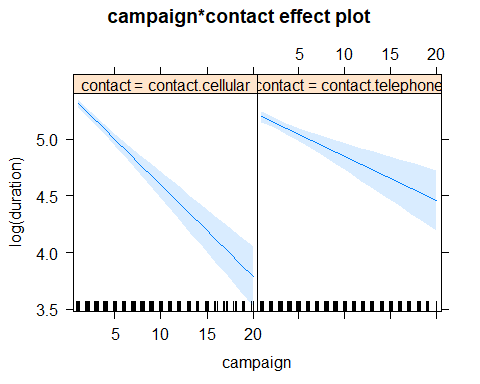
m41<-lm(log(duration)~campaign\*contact,data=df)  
summary(m41)

##   
## Call:  
## lm(formula = log(duration) ~ campaign \* contact, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.5870 -0.5627 -0.0022 0.5937 2.7245   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 5.396770 0.023758 227.153 < 2e-16  
## campaign -0.080359 0.007515 -10.694 < 2e-16  
## contactcontact.telephone -0.161462 0.038797 -4.162 3.21e-05  
## campaign:contactcontact.telephone 0.041526 0.010864 3.822 0.000134  
##   
## (Intercept) \*\*\*  
## campaign \*\*\*  
## contactcontact.telephone \*\*\*  
## campaign:contactcontact.telephone \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9002 on 4986 degrees of freedom  
## Multiple R-squared: 0.0289, Adjusted R-squared: 0.02831   
## F-statistic: 49.46 on 3 and 4986 DF, p-value: < 2.2e-16

# Are interactions significant?  
anova(m40,m41) # pvalue << 0.05 -> H0 Rejected -> m41 X\*A

## Analysis of Variance Table  
##   
## Model 1: log(duration) ~ campaign + contact  
## Model 2: log(duration) ~ campaign \* contact  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 4987 4052.6   
## 2 4986 4040.7 1 11.841 14.611 0.0001337 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(allEffects(m41))



# Outcome/Target : A binary response variable (Binary Target) will be the response variable for Binary Regression Models included in Statistical Modeling Part III.

Explicative Variables for modeling purposes are those available in dataset, exceptions will be indicated, if any.

Multivariant Analysis conducted in previous deliverables has to be used to select the initial model. Students have some degrees in freedom in model building, but the following conditions are requested:

\* Split the sample in work and test samples (consisting on a 80-20 split). Working data frame has to be used for model building purposes.  
  
\*At least two numerical variables have to be considered as explicative variables for initial steps in model building.  
\*Select the most significant factors according to feature selection as initial model factors. Put some reasonable limits to initial model complexity.  
\*You have to consider at least one interaction between a couple of factors and one interaction between factor and covariate.  
\*Diagnostics of the final model have to be undertaken. Lack of fit observations and influence data have to be selected and discussed (connections to multidimensional outliers in Multivariant Data Analysis is highly valuable).  
\*You have to predict Y (Binary Target) in the Working Data Frame vs the rest according to the best validated model that you can find and make a confusion matrix.  
\*Make a confusion matrix in the Testing Data Frame for Y (Binary Target) according to the best validated model found.

#en clase  
#adding factors  
gm10<-glm(y~pdays+emp.var.rate+poly(previous,2)+cons.price.idx+campaign+cons.conf.idx,family=binomial,data=df)

Confusion Matrix: When referring to the performance of a classification model, we are interested in the modelâs ability to correctly predict or separate the classes. When looking at the errors made by a classification model, the confusion matrix gives the full picture. Consider e.g. a three class problem with the classes A, and B. The confusion matrix shows how the predictions are made by the model. The rows correspond to the known class of the data, i.e. the labels in the data. The columns correspond to the predictions made by the model. The value of each of element in the matrix is the number of predictions made with the class corresponding to the column for examples with the correct value as represented by the row. Thus, the diagonal elements show the number of correct classifications made for each class, and the off-diagonal elements show the errors made.