Fernando Ferri

Práctica 2- MBD, Regresión logística

Parte 1) Problema a resolver

Uno de los elementos de la estrategia de Márketing de cualquier entidad financiera es la distribución, que trata de informar al usuario de los nuevos productos y servicios en unas condiciones que maximicen las opciones de adquisición. El desarrollo y aplicación de la segmentación de clientes, es una condición necesaria para competir adecuadamente en los diferentes ámbitos que la entidad financiera haya definido como mercado objetivo.

El telemarketing es una de las múltiples vías empleadas por las entidades financieras para ofrecer sus productos. Estas llamadas tienen unos costes de personal y de consumo que hacen primordial una buena selección de los individuos a los cuáles dirigir la venta. La entidad bancaria Just-Your-Bank (JYK) tiene el propósito de reducir dichos costes reduciendo el número de llamadas realizadas a aquellos clientes con pocas opciones de contractar el producto ofrecido.

El objetivo de esta práctica es predecir la probabilidad que un cliente al cual se realiza la llamada acepte el producto que se le ofrece.

Datos

Se entregarán dos conjuntos de datos con registros de clientes de JYK a los que se les realizó una llamada para vender un producto concreto:

1. Datos de entrenamiento. Contendrán la variable respuesta (contrató o no contrató el producto)
2. Datos test. No contendrán la variable respuesta y ésta deberá predecirse basándose en los datos históricos del conjunto de entrenamiento.

Las variables comunes que contienen ambos juegos son:

<u>Características del cliente</u>
id: identificador del cliente
age: edad
job: tipo de trabajo (admin., blue-collar, entrepreneur, housemaid, management, retired, self-employed, services, student, technician, unemployed, unknown)
marital: estado civil (divorced, married, single, unknown)
education: nivel de estudios (basic.4y, basic.6y, basic.9y, high.school, illiterate, professional.course, university.degree, unknown)
default: ¿es moroso? (no,yes,unknown)
housing: ¿tiene hipoteca? (no,yes,unknown)
loan: ¿tiene un préstamo personal? (no,yes,unknown)

<u>Características de la llamada</u>
contact: tipo de teléfono (cellular, telephone)
month: mes

day_of_week: día de la semana (mon, tue, wed, thu, fri)

Otros atributos
campaign: número de contactos realizados esta campaña para este cliente (incluyendo el actual)
pdays: número de días que han pasado desde que el cliente fue contactado por última vez para una campaña previa (999 significa que no fue contactado previamente)
previous: número de llamadas realizadas a este cliente antes de esta campaña
poutcome: resultado de la anterior campaña (failure, nonexistent, success)

Indicadores del contexto social y económico
emp.var.rate: indicador de la tasa de empleo (cuatrimestral)
cons.price.idx: IPC (mensual)
cons.conf.idx: Índice de confianza del consumidor (mensual)
euribor3m: euribor a 3 meses (diario)
nr.employed: número de empleados (cuadrimestral)

Variable respuesta (sólo en el juego de entrenamiento):
Y: ¿Se suscribió el cliente al depósito? (yes,no)

Evaluación

Se basará en el AUC (área bajo la curva ROC) del modelo aplicado sobre la prueba test.

$$\int_{\infty}^{-\infty} \mathrm{TPR}(T)\mathrm{FPR}'(T)\,dT$$

Donde:
TPR es la proporción de verdaderos positivos
FPR es la proporción de falsos positivos

Se obtendrá una puntuación más alta cuanto mayor sea este estadístico. Adicionalmente se valorará:

1. La parsimonia del modelo
2. Los criterios de selección del modelo
3. La presentación de los resultados

Parte 2: Estudio estadístico

Objetivo: Entrenar un modelo a partir de los datos de entrenamiento (training) con el objetivo predecir el valor de salida (y= exito de la campaña / adquisición del depósito) en los datos de test (testing – test validity and model accuracy).

2.1 Introducción conceptos estadísticos necesarios:

Para ello realizaremos un análisis estadístico basado en la regresion logística. Éste modelo predice a partir de un modelo diseñado con los coeficientes estimados para cada variable considerada en el modelo el valor de la variable salida o a predecir.

La variable salida predicha en la regresión logística no es normal, sino dicotómica ya que sólo puede tener dos valores posibles.

Función link

Transormación que se le aplica a la probabilidad de éxito (Π) para obtener valores entre -∞ y +∞.

- La probabilidad de éxito (**Π**) se mueve en el intervalo [0,1].
- **Odd**: (probabilidad éxito / probabilidad fracaso);
  - **Odd:** Π / (1- Π) = resultado entre [0, +∞].
- Aplicando el logaritmo obtenemos lo deseado:
  - **Función Link:** log(Π / (1- Π)) = [-∞, +∞].

a. Así pues, **f(x) = log(x/(1-x)) se** denomina función logit.
b. Al aplicar la función logit a una probabilidad el resultado se denomina **logodd** (logaritmo de odd)

Así pues, el modelo a ajustar será (en R se calcula con la función "glm" con el parámetro

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \cdots + \beta_p \cdot X_p$$

"family=binomial":

Los parámetros se estiman por el método de máxima verosimilitud, asumiendo que la variable respuesta sigue una distribución binomial, maximizando el valor de los parámetros que maximicen la probabilidad de haber observado esos datos.

En caso que se desee conocer la probabilidad para unas variables predicotras una vez estimado el modelo (con sus coeficientes), se tiene que deshacer la transformación:

$$PL = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \cdots + \beta_p \cdot X_p$$

$$\log\left(\frac{\pi}{1-\pi}\right) = PL \rightarrow \frac{\boldsymbol{\pi}}{\boldsymbol{1-\pi}} = \boldsymbol{odd} = \boldsymbol{e^{PL}} \rightarrow \pi = e^{PL} - \pi \cdot e^{PL} \rightarrow (1 + e^{PL}) \cdot \pi = e^{PL} \rightarrow \boldsymbol{\pi} = \frac{\boldsymbol{e^{PL}}}{\boldsymbol{1 + e^{PL}}}$$

Ejemplo externo de cómo interpretar los datos predichos por un modelo:

R call (Y → prediction → will payo r not the loan):

glm(formula=y ~ status+ duration+age…., family= binomial, data = datos)

```
Coefficients:
                                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                                     1.006e+00  1.176e+00   0.855 0.392390
status>= 200 DM                                 1.382e+00  4.272e-01   3.236 0.001213 **
status0 <=  and < 200 DM                        6.522e-01  2.533e-01   2.575 0.010035 *
statusno checking account                       2.086e+00  2.769e-01   7.535 4.90e-14 ***
duration                                       -2.832e-02  1.093e-02  -2.591 0.009556 **
```

- Variable "duration" **(numérica/meses)**
  - Deshacemos la transformación del valor "estimate" (estimado)
    e^(-0.0282) = 0.97 →

> **OR**_duration(months) → **ODD**_current value/**ODD**_previous value(ref) = 0.97 →
>
> → **ODD**_current_value = 0.97 * **ODD**_previous_value → -3% las probabilidades de pagar el crédito respecto al mes anterior.
>
> **ODD_y: Π** (pagar)/ **(1-Π)** no_pagar(ref)
>
> Categoría referencia para variable salida → Y: "pagar o no pagar".

- Variable "status >=200 DM" **(categórica):**
  - Deshacemos la transfomación del valor "estimate"
    e^(1.382) = 4.00 →

> Probabilidad de pagar es 4 veces superior a la categoria de referencia (cuenta en negativo) para cuentas con más de 200 DM. 4.00 = 400%
>
> **OR_status** = **ODD**_status_ge_200/**ODD**_status_negative (ref)= 4 (400%)

Evaluar capacidad predicción modelo:

El AUC (área bajo la curva ROC) del modelo aplicado sobre la prueba test.

$$\int_{\infty}^{-\infty} \mathrm{TPR}(T)\mathrm{FPR}'(T)\,dT$$

Donde:
TPR es la proporción de verdaderos positivos
FPR es la proporción de falsos positivos

Para curvas más abombadas, o sea, **para áreas bajo la curva más grandes → mejor capacidad predictiva** del modelo

Para mayores AUC, mejor será la predicción del modelo:

- AUC = 1 → diagnostico perfecto
- AUC = 0.5 → discriminación al azar
    - [0.5, 0.6)→ Discriminación mala.
    - [0.6, 0.75) → Discriminación regular.
    - [0.75, 0.9) → Discriminación buena.
    - [0.9, 1)→ Discriminación muy buena.

2.2 Diseño del modelo (training) y su testeado (testing)

2.2.1 PARTE 1: construcción del modelo (training)

1. Borrar objetos en memoria memoria

```
rm(list=ls())rm(list=ls())
```

2. Cargar los datos que se utilizaran para entrenar el modelo (ca 80%)

```
# directorio de trabajo
setwd('/Users/Fer/Development/MASTERBD/ESTADISTICA/D_REGRESIO_LOGISTICA/practica
_reg_logistica')

# lectura de los datos , como veremos a continuación podemos contar 999 como NA.
datos2 <- read.table('p2_train.csv',header=TRUE,sep=";",na.strings=c("NA",999))
```

3. Hacer descriptiva para observar si hay que limpiar algún dato (i.e: eliminar campo no útil o tratar valores NA

```
View(head(datos2,100))   # vemos unos cuantos registros para hacernos una idea

summary(datos2)

datos2$id <-NULL        # Eliminamos id que simplemente identifica una observación.
```

4. Después de leer 999 como NA, vemos que pdays tiene el 96% de NA y la eliminamos

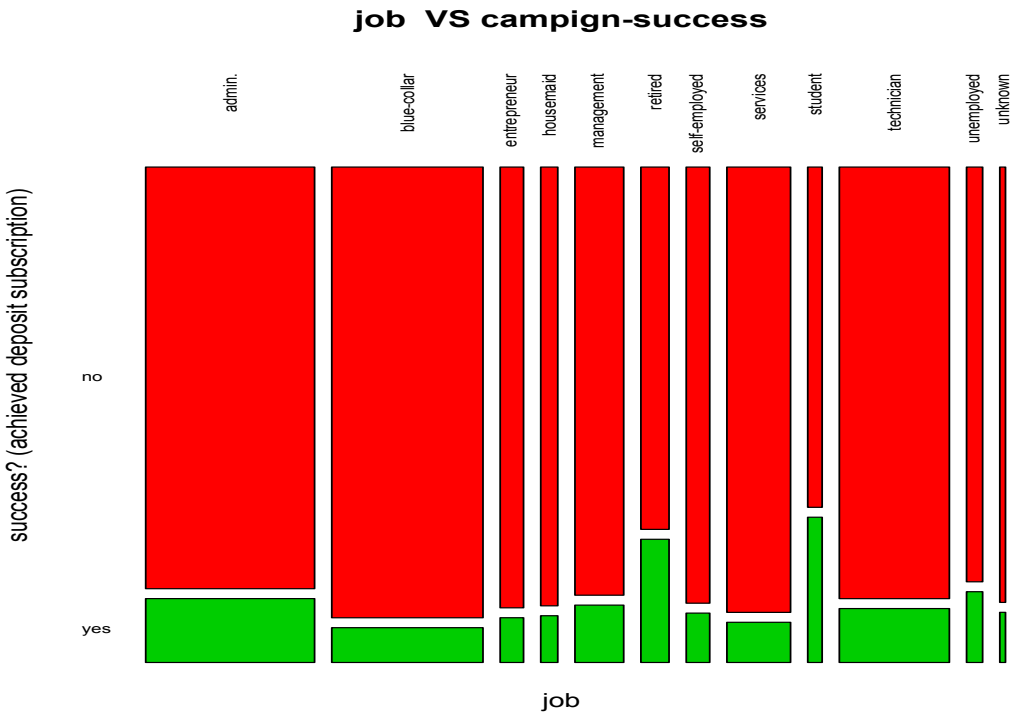
```
howmany_na_variables <- apply(apply(datos2,2,is.na),2,sum)

howmanyPdays <-howmany_na_variables["pdays"]/nrow(datos2)

howmanyPdays

          pdays

       0.9633793

datos2$pdays <- NULL
```

5. **Hacemos una descriptiva bivariante para las variables categóricas y como se relacionan con el valor salida (seleccionamos la posición de las variables categóricas)**

```
sapply(datos2,class)                                    # Clase de las variables
var.cat <- which(sapply(datos2,class)=="factor" & names(datos2)!="y") # Se seleccionan los factores y no sea la respuesta
```

6. **Plot con las variables categóricas y la respuesta**

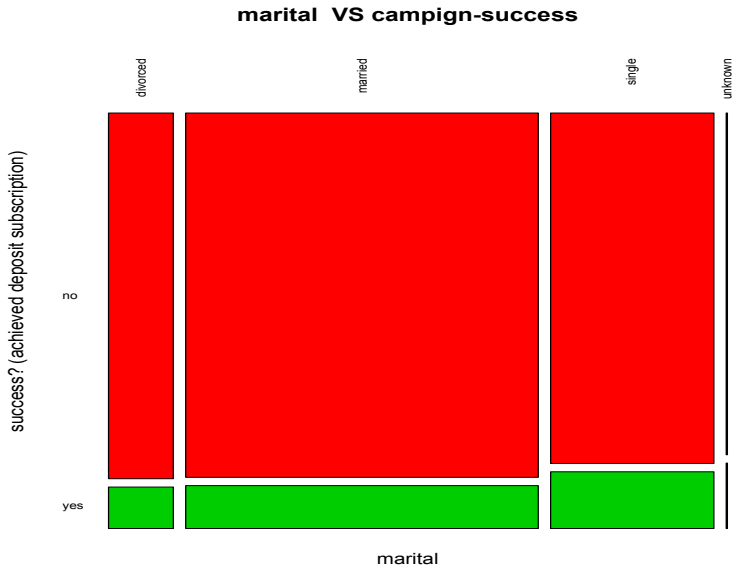
```
quartz() #ventana grande en MAC-OS

for(vc in var.cat){ mosaicplot(datos2[,vc]~datos2$y,main=paste(names(datos2)[vc]," VS campign-success"),col=2:3,las=2,xlab=names(datos2)[vc],ylab="success? (achieved deposit subscription)")

}
```



Observando como se pueden agrupar las variables, las agrupamos en el menor número de variables posibles dentro de lo razonable:

7. A partir de este mosaicplot agrupamos los trabajos en el menor número de categorías razonables, creamos una variable job2 con solo 2 categorias:

```
datos2$job2 <- factor(ifelse(datos2$job %in% c("retired","student"),"not_active","active"))

datos2$job <- NULL
```

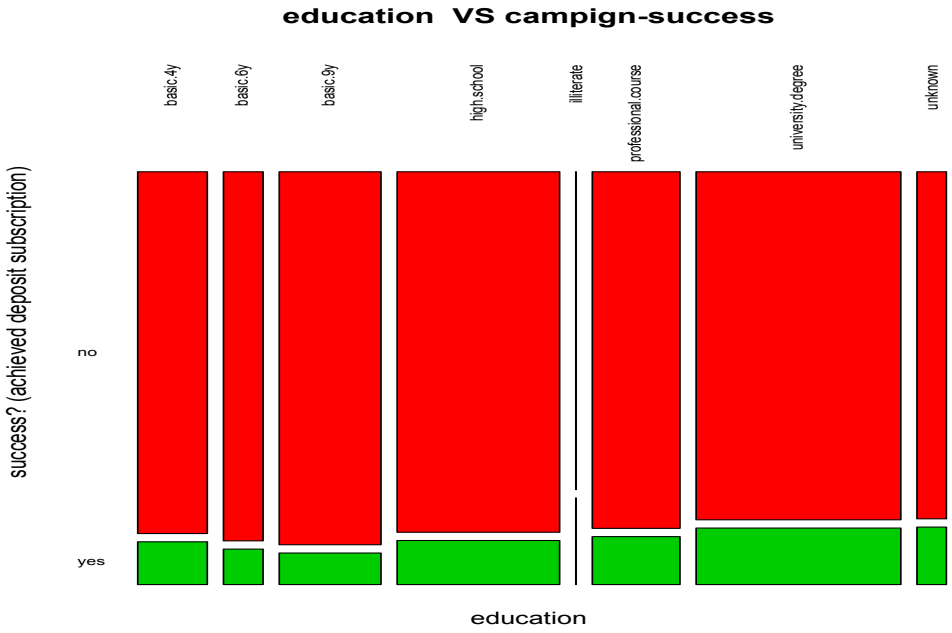


#Juntamos divorced y married al parecer tener la misma probabilidad de venta

#table(datos2$marital,datos2$y)

datos2$marital2 <- factor(ifelse(datos2$marital %in%
c("divorced","married"),'have_had_commitment','other'))

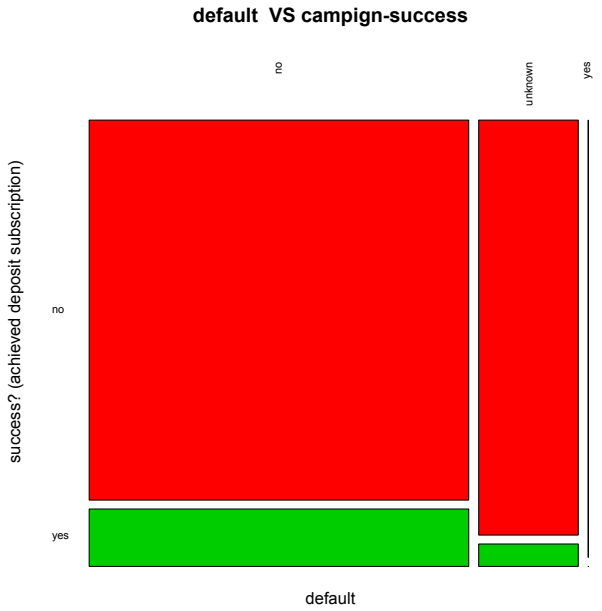
datos2$marital <- NULL



9. Miramos el modelo ajustado para la variable educación y agrupamos las que mayor similitud en efecto y semántica tienen: basic6y y basic.9y tienen un valor estimado parecido

```
mod.glm.edu <- glm(y~education,datos2,family=binomial)

summary(mod.glm.edu)

datos2$education2 <- factor(ifelse(datos2$education %in%
c("basic.6y","basic.9y"),"educationbasic",as.character(datos2$education)))

datos2$education <- NULL
```

```
Coefficients:

                    Estimate Std. Error z value Pr(>|z|)

(Intercept)              -2.13520    0.06075 -35.148  < 2e-16 ***

educationbasic.6y         -0.20714    0.10656  -1.944 0.051912 .

educationbasic.9y         -0.33430    0.08371  -3.994 6.51e-05 ***

educationhigh.school        0.03355    0.07232   0.464 0.642747

educationilliterate         0.83592    0.65417   1.278 0.201307

educationprofessional.course  0.12906    0.07963   1.621 0.105083

educationuniversity.degree    0.31760    0.06839   4.644 3.42e-06 ***

educationunknown            0.33764    0.10231   3.300 0.000966 ***

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 20398  on 28644  degrees of freedom. Residual deviance: 20260  on 28637  degrees of freedom. AIC: 20276
```



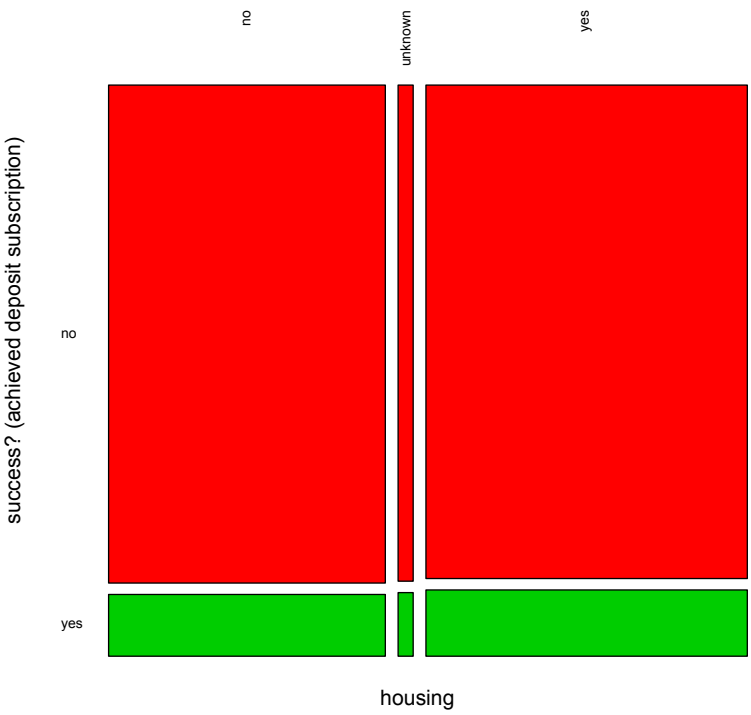
10. La variable default tiene una categoría sin individuos que hayan comprado. La juntamos esta categoría con otra obteniendo 2 categorias.

table(datos2$default,datos2$y)

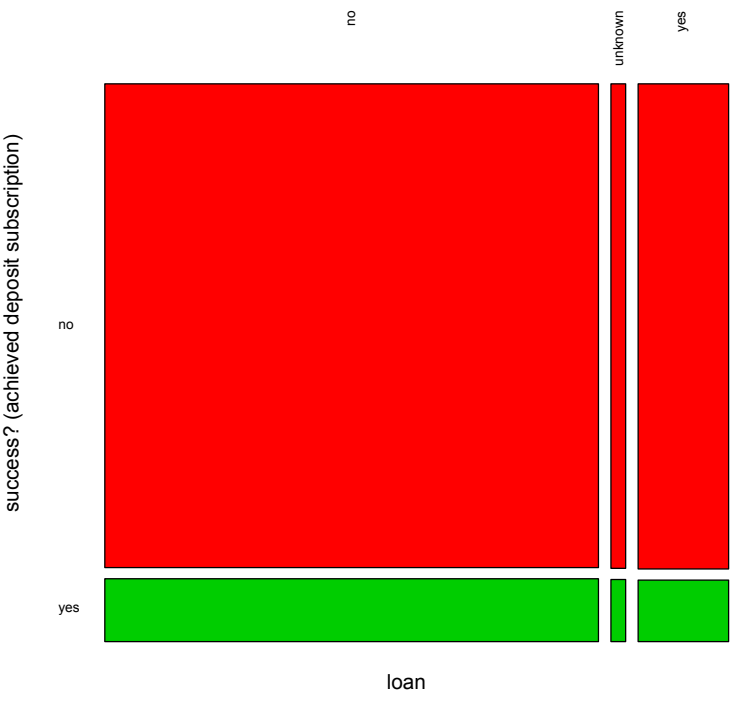
datos2$default2 <- factor(ifelse(datos2$default=='no','no','other'))

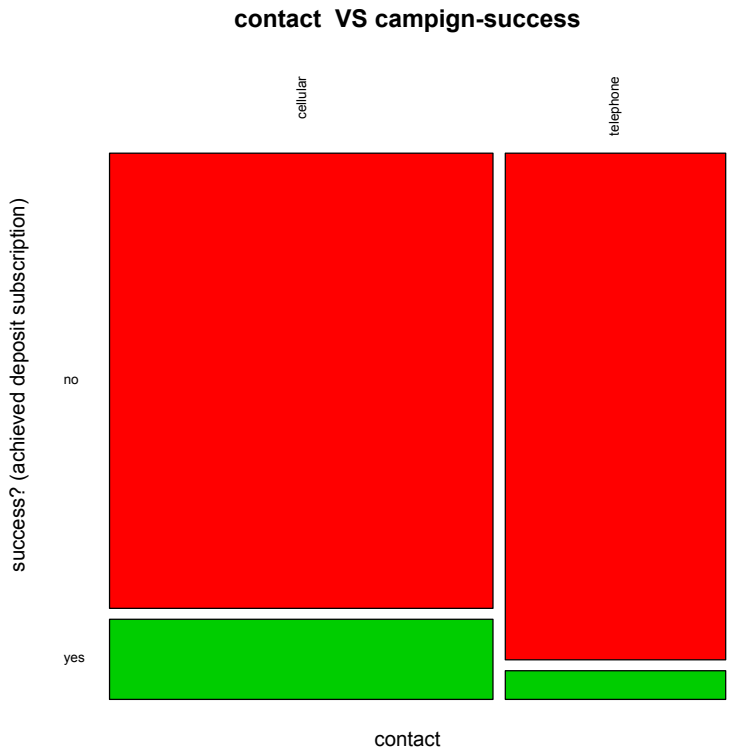
datos2$default <- NULL

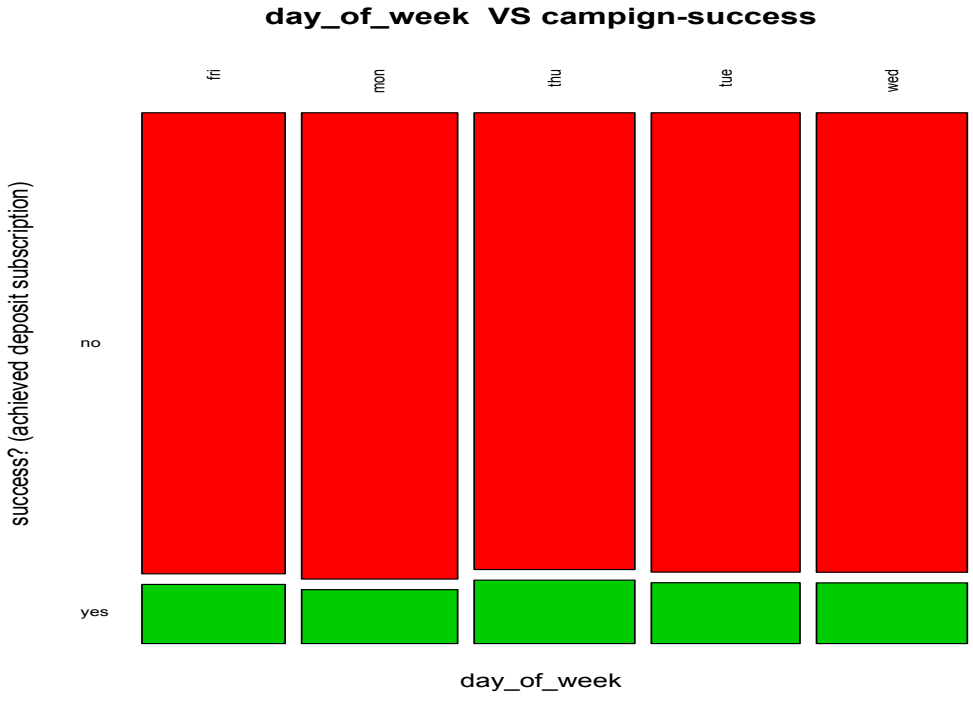
housing VS campign-success

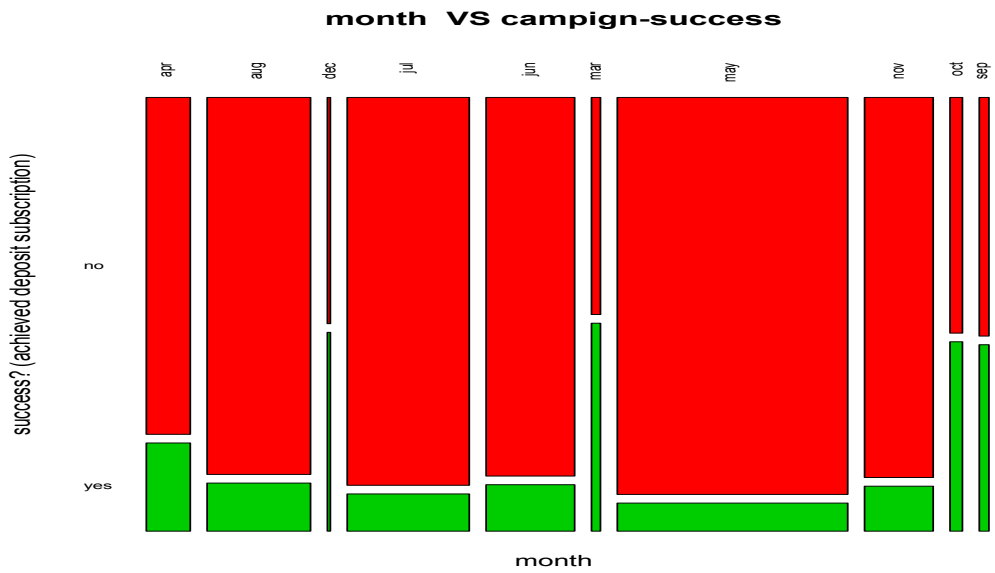


loan VS campign-success



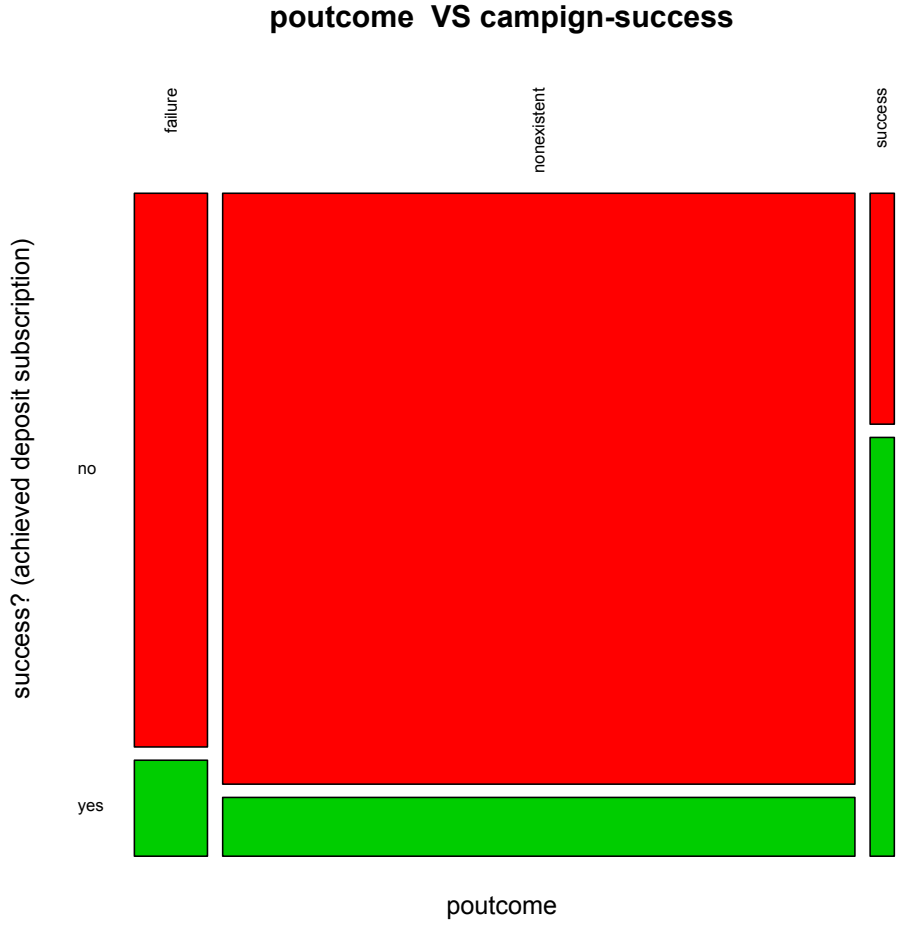

contact VS campign-success
cellular
telephone
success? (achieved deposit subscription)
no
yes
contact


day_of_week VS campign-success
fri
mon
thu
tue
wed
success? (achieved deposit subscription)
no
yes
day_of_week



8. Creamos otra variable month2 para agrupar en 2 grupos similares

```
datos2$month2 <- factor(ifelse(datos2$month %in%
c("dec","mar","oct","sep"),"higherOffer","lowerOffer"))

datos2$month <- NULL
```


poutcome VS campign-success
failure
nonexistent
success
success? (achieved deposit subscription)
no
yes
poutcome

11. Descriptiva bivariante para variables numericas . Vamos a analizar como influye la variabilidad de las variables numéricas en la aceptar o no el deposito.

Creamos un vector (var.num) que contenga la posicion de todas las variables numéricas o enteras

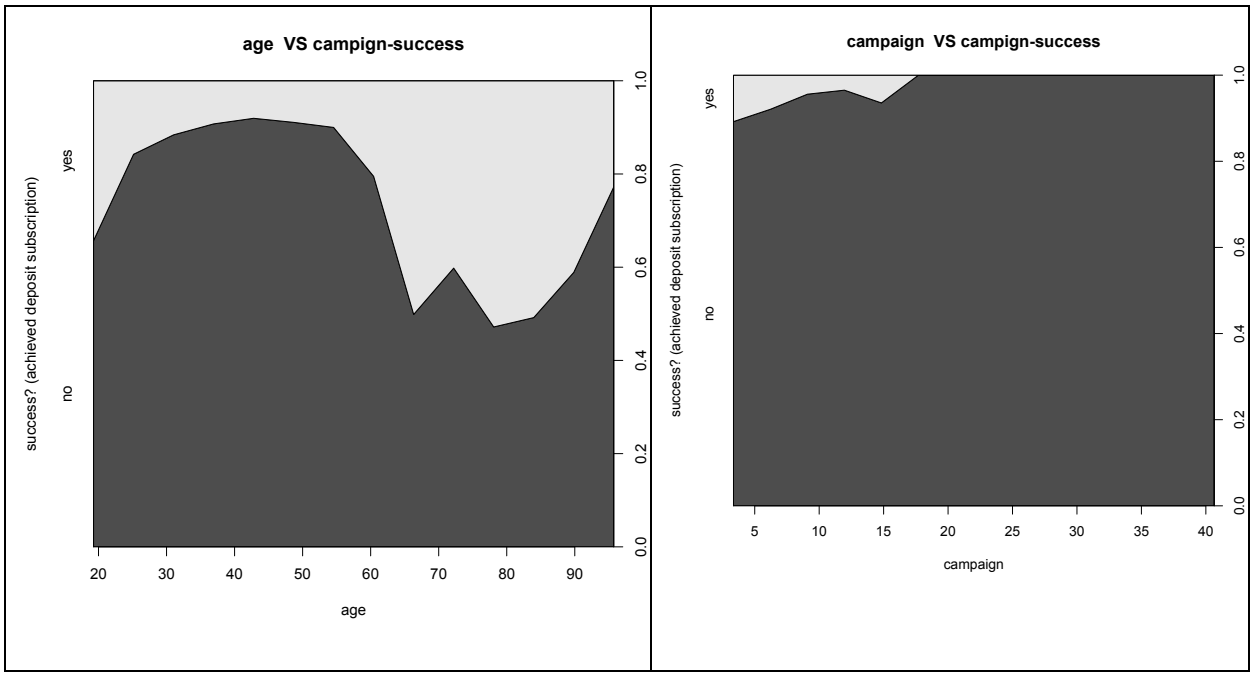
```
var.num <- which(sapply(datos2,class) %in% c("numeric","integer"))
```

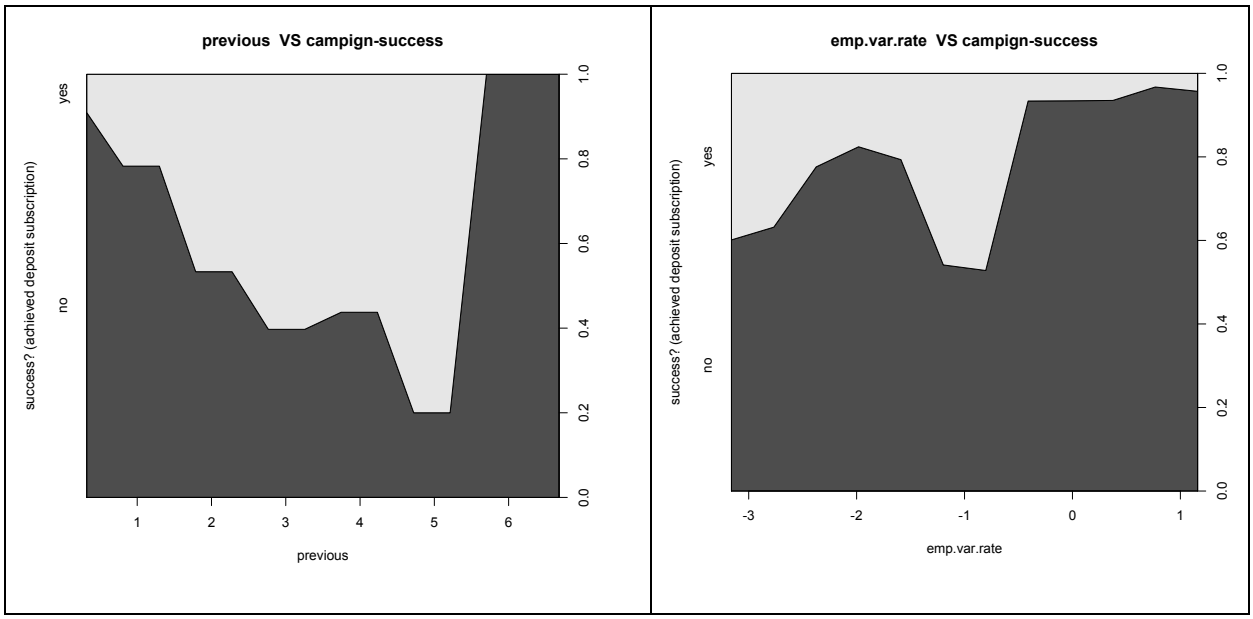
12. Usamos la función cdplot para dibujar las densidades de todas las variables numéricas con la respuesta

```
for(vn in var.num){

 cdplot(datos2$y~datos2[,vn],main=names(datos2)[vn],n=16)

}
```

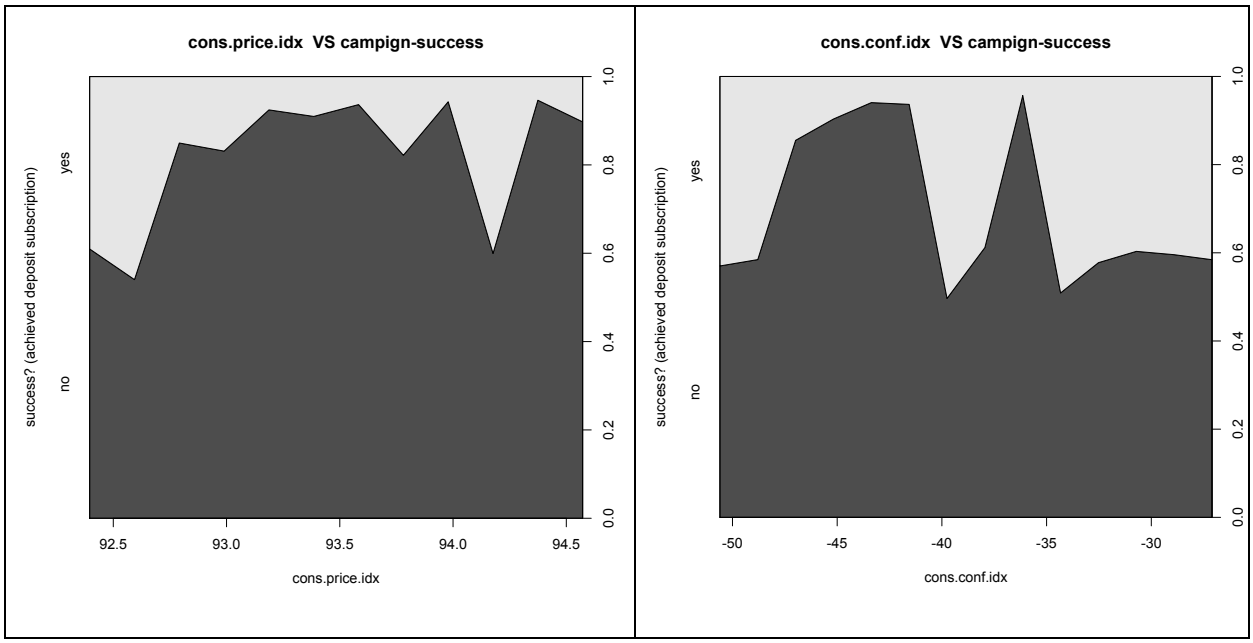
Conclusiones:

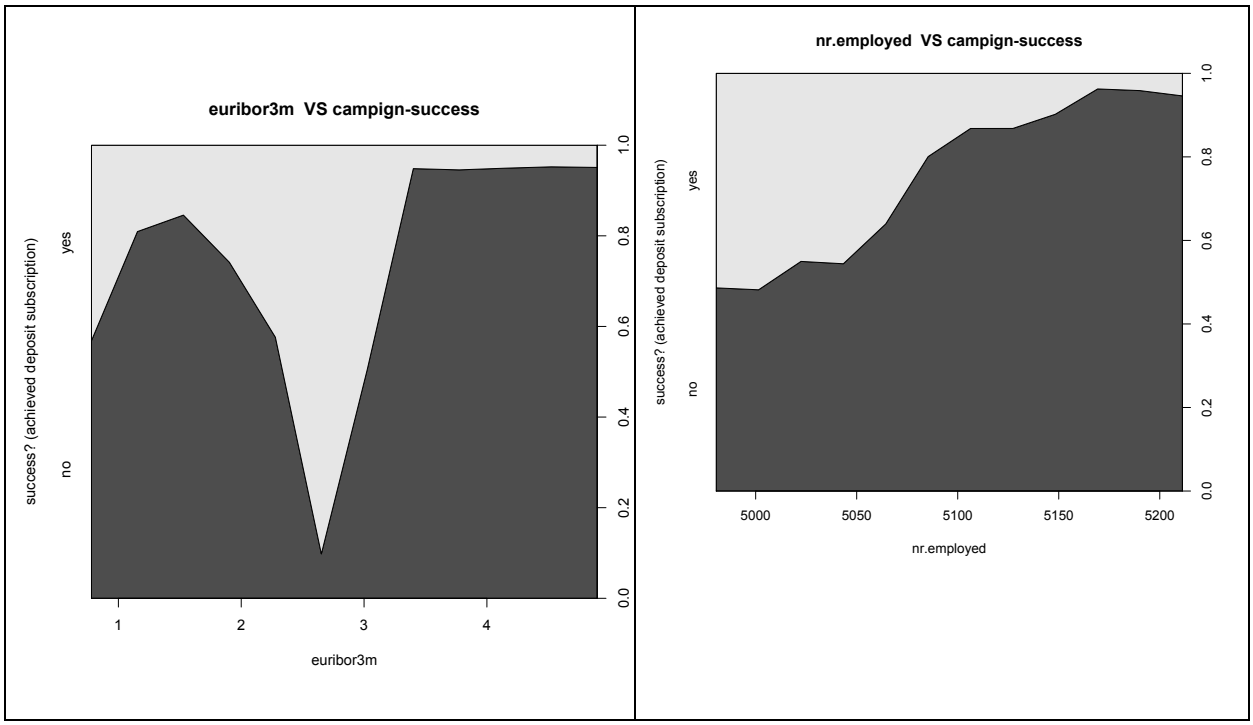
- **Age:** entre 25 y 55 años la probabilidad de no adquisición (rechazo) esta por encima del 80%. En cambio, entre 60 y 90 se encuentra la probabilidad de aceptación incluso hasta el 50% en algunos tramos.
- **Number of campaingns:** vemos que la probabilidad más alta de aceptación están en las primeras campañas. Cuentas más campañas se realizan menor es la probabilidad incluso prácticamente nula si el número de campañas es superior a 17, por lo que para ahorrar costes no se debería intentar más pasadas 17 veces.





- **Previous**: indica numero de contactos realizados a un cliente previos a esta campaña. Vemos que entre 2 y 5 contactos obtienen la mayor tasa de éxito.
- **Emp.var.rate:** en valores negativos de la tasa de empleo se obtiene mayor éxito. Para valores positivos o cercanos a cero la tasa de éxito es casi nula, así que una tasa negativa seria beneficiosa para obtener clientes que compren el deposito.
- **Price index:** Tasas entre[92.5, 93] y [94, 94.3] obtienen más éxito.
- **Consumer confidence index:** Parece ser que valores más cercanos al 0 tienen mejor éxito.





- **Euribor**: entre 2 y 3 obtiene más éxito.
- **Nr.employed:** cuanto más grande, menos probabilidad , o sea, una cantidad menor beneficia para obtener el objetivo. Inversamente proporcional a nuestro objetivo.

13. Estimamos el modelo con todas las variables

mod.glm0 <- glm(y~., datos2 ,family=binomial)

summary(mod.glm0)

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-1.792e+02	1.877e+01	-9.550	< 2e-16	***
age	2.892e-03	2.148e-03	1.346	0.178148	
maritalmarried	9.047e-02	7.122e-02	1.270	0.203980	
maritalsingle	1.220e-01	8.082e-02	1.509	0.131179	
maritalunknown	3.624e-01	4.299e-01	0.843	0.399257	
housingunknown	-7.291e-02	1.382e-01	-0.527	0.597899	
housingyes	-3.324e-02	4.248e-02	-0.782	0.433997	
loanunknown	NA	NA	NA	NA	
loanyes	-1.558e-02	5.852e-02	-0.266	0.790081	

contacttelephone -9.222e-01 6.653e-02 -13.860 < 2e-16 ***

day_of_weekmon -1.791e-01 6.749e-02 -2.654 0.007952 **

day_of_weekthu 7.333e-02 6.516e-02 1.125 0.260406

day_of_weektue 4.785e-02 6.717e-02 0.712 0.476246

day_of_weekwed 8.222e-02 6.693e-02 1.228 0.219265

campaign -4.949e-02 1.111e-02 -4.456 8.36e-06 ***

previous 6.925e-02 6.481e-02 1.069 0.285243

poutcomenonexistent 5.699e-01 1.034e-01 5.513 3.52e-08 ***

poutcomesuccess 1.744e+00 9.330e-02 18.692 < 2e-16 ***

emp.var.rate -7.519e-01 7.142e-02 -10.529 < 2e-16 ***

cons.price.idx 1.519e+00 1.175e-01 12.926 < 2e-16 ***

cons.conf.idx 5.572e-02 6.472e-03 8.610 < 2e-16 ***

euribor3m -3.273e-01 9.487e-02 -3.450 0.000561 ***

nr.employed 7.437e-03 1.782e-03 4.174 2.99e-05 ***

job2not_active 3.131e-01 7.349e-02 4.261 2.04e-05 ***

month2lowerOffer -6.574e-01 8.495e-02 -7.738 1.01e-14 ***

education2educationbasic -1.544e-02 9.005e-02 -0.171 0.863865

education2high.school 8.052e-02 8.675e-02 0.928 0.353269

education2illiterate 8.235e-01 7.235e-01 1.138 0.254984

education2professional.course 1.718e-01 9.409e-02 1.826 0.067824 .

education2university.degree 2.500e-01 8.373e-02 2.986 0.002822 **

education2unknown 1.668e-01 1.197e-01 1.393 0.163588

default2other -3.378e-01 6.812e-02 -4.959 7.09e-07 ***

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 20398 on 28644 degrees of freedom

Residual deviance: 16333 on 28614 degrees of freedom

AIC: 16 395. Number of Fisher Scoring iterations: 6

14. Selecciona automáticamente las variables relevantes con step

Mirar si hay NA

apply(apply(datos2,2,is.na),2,sum)

#Se ejecuta la selección automática del modelo con step que nos debería estimar los parámetros del modelo.

mod.glm1 <- step(mod.glm0)

summary(mod.glm1)

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	*-1.794e+02*	*1.876e+01*	*-9.567*	*< 2e-16*	*****
contacttelephone	*-9.291e-01*	*6.628e-02*	*-14.017*	*< 2e-16*	*****
day_of_weekmon	*-1.784e-01*	*6.746e-02*	*-2.645*	*0.008174*	****
day_of_weekthu	*7.135e-02*	*6.513e-02*	*1.096*	*0.273288*	
day_of_weektue	*4.631e-02*	*6.716e-02*	*0.689*	*0.490528*	
day_of_weekwed	*8.037e-02*	*6.690e-02*	*1.201*	*0.229655*	
campaign	*-4.958e-02*	*1.111e-02*	*-4.460*	*8.18e-06*	*****
poutcomenonexistent	*4.840e-01*	*6.465e-02*	*7.486*	*7.10e-14*	*****
poutcomesuccess	*1.759e+00*	*9.247e-02*	*19.024*	*< 2e-16*	*****
emp.var.rate	*-7.515e-01*	*7.137e-02*	*-10.530*	*< 2e-16*	*****
cons.price.idx	*1.529e+00*	*1.174e-01*	*13.028*	*< 2e-16*	*****
cons.conf.idx	*5.680e-02*	*6.450e-03*	*8.806*	*< 2e-16*	*****
euribor3m	*-3.272e-01*	*9.478e-02*	*-3.452*	*0.000556*	*****
nr.employed	*7.357e-03*	*1.776e-03*	*4.143*	*3.42e-05*	*****
job2not_active	*3.415e-01*	*6.996e-02*	*4.882*	*1.05e-06*	*****
month2lowerOffer	*-6.598e-01*	*8.480e-02*	*-7.781*	*7.19e-15*	*****
education2educationbasic	*-3.141e-02*	*8.800e-02*	*-0.357*	*0.721090*	
education2high.school	*6.088e-02*	*8.356e-02*	*0.729*	*0.466268*	
education2illiterate	*8.192e-01*	*7.248e-01*	*1.130*	*0.258359*	
education2professional.course	*1.554e-01*	*9.231e-02*	*1.683*	*0.092350*	*.*
education2university.degree	*2.352e-01*	*8.094e-02*	*2.906*	*0.003664*	****
education2unknown	*1.599e-01*	*1.185e-01*	*1.350*	*0.177070*	
default2other	*-3.295e-01*	*6.771e-02*	*-4.867*	*1.13e-06*	*****

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 20398 on 28644 degrees of freedom

Residual deviance: 16338 on 28622 degrees of freedom

AIC: 16 384

Number of Fisher Scoring iterations: 6

Así pues, tras la selección automática, los coeficientes del modelo quedan así para cada variable:

coef(mod.glm1)

(Intercept)	***contacttelephone***	***day_of_weekmon***
-1.794255e+02	*-9.290683e-01*	*-1.784039e-01*
day_of_weekthu	***day_of_weektue***	***day_of_weekwed***
7.135095e-02	*4.630586e-02*	*8.036635e-02*
campaign	***poutcomenonexistent***	***poutcomesuccess***
-4.958009e-02	*4.839639e-01*	*1.759048e+00*
emp.var.rate	***cons.price.idx***	***cons.conf.idx***
-7.514796e-01	*1.528800e+00*	*5.680256e-02*
euribor3m	***nr.employed***	***job2not_active***
-3.271954e-01	*7.357258e-03*	*3.415211e-01*
month2lowerOffer	***education2educationbasic***	***education2high.school***
-6.598153e-01	*-3.141477e-02*	*6.088059e-02*
education2illiterate	***education2professional.course***	***education2university.degree***
8.191920e-01	*1.553787e-01*	*2.351877e-01*
education2unknown	***default2other***	
1.599008e-01	*-3.295361e-01*	

Vemos que muchas de las variables han sido eliminadas del modelo , por lo que las contamos como no relevantes para la predicción. I.e: age, y loan (ésta parecía no ser relevante al realizar el análisis bivariante con el grafico anteriormente, ya que todos sus valores obtenían resultado similar).

15. Validación. Instalamos la siguiente librería

```
install.packages('ResourceSelection')

library(ResourceSelection)
```

16. Realiza el test de Hosmer-Lemeshow

```
valores.reales <- mod.glm1$y

valores.predichos <- fitted(mod.glm1)

#?hoslem.test

hoslem.test(valores.reales ,valores.predichos)
```

Hosmer and Lemeshow goodness of fit (GOF) test

data: valores.reales, valores.predichos

X-squared = 43.65, df = 8, p-value = 6.625e-07

Observamos que el ***test de Hosmer and Lemeshow falla***. No obstante era de esperar en volúmenes de datos grandes.

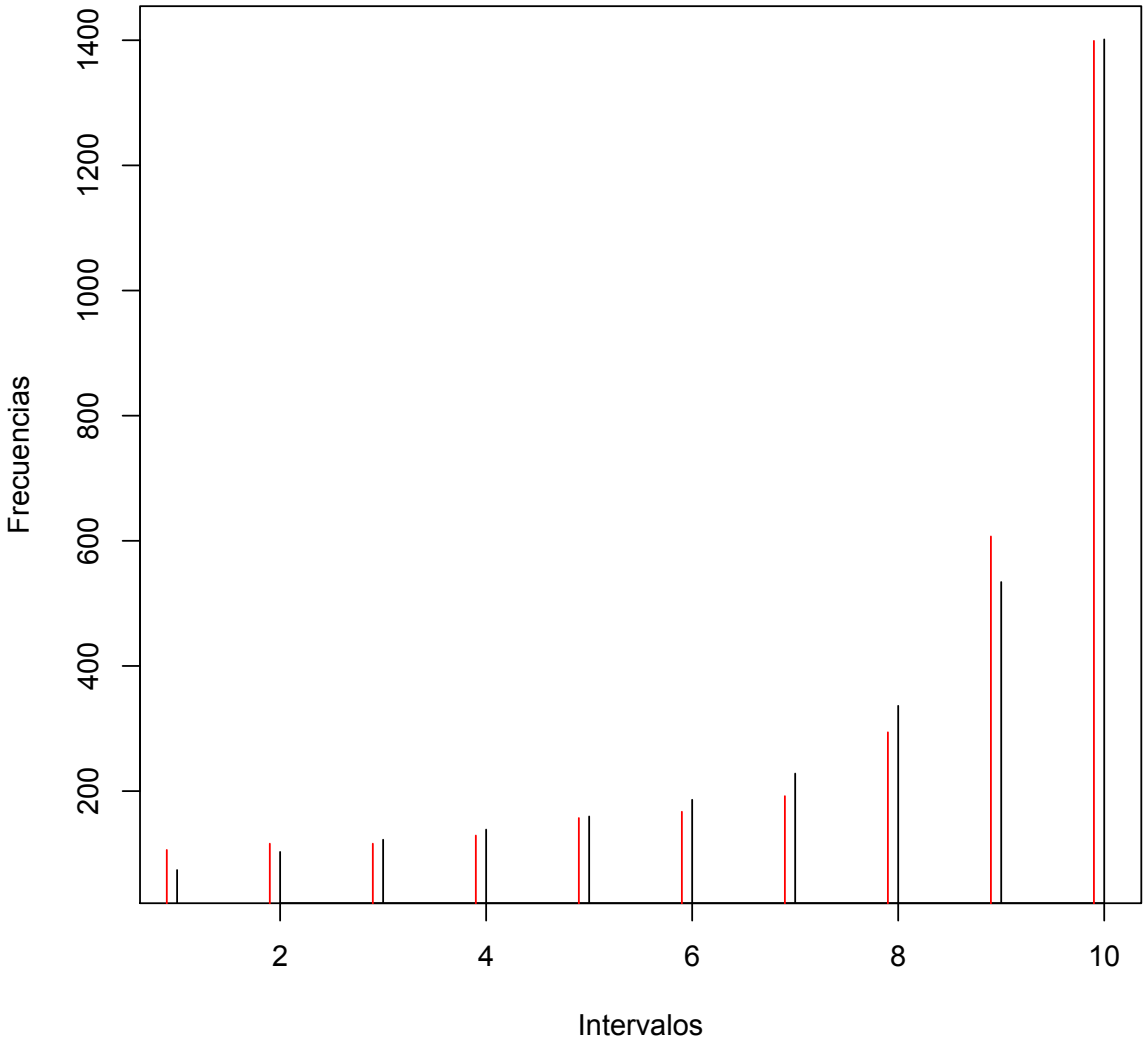
Así pues, **vamos a realizar una validación visual** para validar el modelo.

17. Dividimos los valores predichos en deciles

```
br <- quantile(valores.predichos,seq(0,1,0.1))
```

18. Dibujamos los efectivos predichos vs los esperados en cada cuantil

```
quartz()

int <- cut(fitted(mod.glm1),br)

obs <- tapply(mod.glm1$y,int,sum)

exp <- tapply(fitted(mod.glm1),int,sum)

plot(1:10,exp,type='h',xlab="Intervalos",ylab="Frecuencias")

lines(1:10-0.1,obs,type='h',col=2)
```



Tras dividir los datos test en diez grupos, el análisis visual sobre los valores predichos / *expected* (barras negras) y los reales/ *observed* extraídos de los datos test (barra negra), vemos que la predicción se acerca mucho al valor real básicamente para todos los diez grupos en que se han dividido las observaciones.

19,20. Calculamos el OR de contactar al teléfono (contact) fijo respecto a hacerlo por móvil

OR fijo_móvil = ODD_fijo/ ODD_movil (ref)

- celular (movil) → variable referencia (1ª por orden alfabético)
- telephone (fijo)

Proceso:

- Obtenemos el coeficiente para "fijo": mod.glm1$coefficients["contacttelephone"] → -0.9290683
- Ha sido calculado respecto la ODD del móvil (referencia) en el modelo.
- Deshacer la transformación: e^(-0.9290683) = **0.3949215**
- **R code:** exp(mod.glm1$coefficients["campaign"])

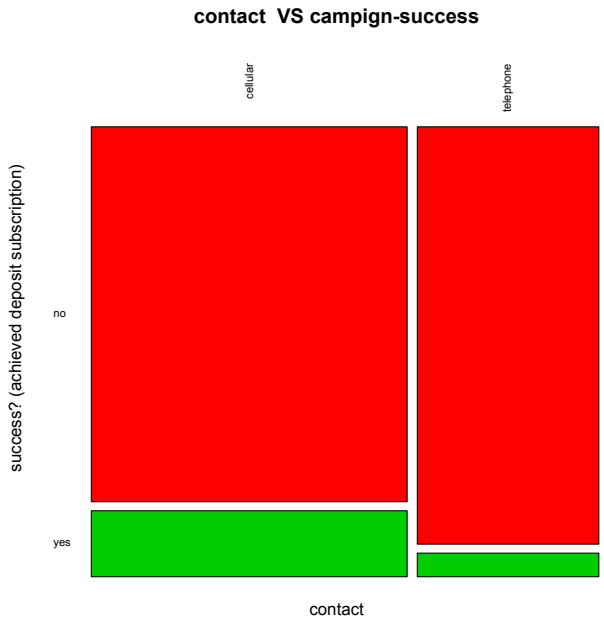
ODD_fijo_movil = ***ODD_fijo/ ODD_movil (ref)*** →

exp(mod.glm1$coefficients["contacttelephone"]) → **0.3949215**

Resultado →

ODD_fijo = 0.39 * **ODD**_movil → En fijo tiene -60% las probabilidades de vender el crédito respecto al movil.

Si recordamos el gráfico anterior vemos que era lo esperado en el análisis.



21. Calculamos los intervalos de confianza del 95% para todos los ORs

Cuál puede ser la variable más relevante a la hora de tener éxito en la venta?

- Para esto calculamos los intervalos de confianza y confluiremos que la variable con más influencia será la que tiene el intervalo de confianza más estrecho.

IC <- confint(mod.glm1) # Intervalos de confianza para los coeficiente.

round(exp(IC),2) # Intervalos de confianza para los ORs

	row.names	2.5 %	97.5 %	dif
1	(Intercept)	0.00	0.00	0.00 (EXCLUIR)
2	contacttelephone	0.35	0.45	0.10
3	day_of_weekmon	0.73	0.95	0.22
4	day_of_weekthu	0.95	1.22	0.27
5	day_of_weektue	0.92	1.19	0.27
6	day_of_weekwed	0.95	1.24	0.29
7	campaign	0.93	0.97	0.04
8	poutcomenonexistent	1.43	1.84	0.41
9	poutcomesuccess	4.85	6.97	2.12
10	emp.var.rate	0.41	0.54	0.13
11	cons.price.idx	3.66	5.80	2.14
12	cons.conf.idx	1.05	1.07	0.02
13	euribor3m	0.60	0.87	0.27
14	**nr.employed**	**1.00**	**1.01**	**0.01**
15	job2not_active	1.23	1.61	0.38
16	month2lowerOffer	0.44	0.61	0.17
17	education2educationbasic	0.82	1.15	0.33
18	education2high.school	0.90	1.25	0.35
19	education2illiterate	0.45	8.37	7.92
20	education2professional.course	0.98	1.40	0.42
21	education2university.degree	1.08	1.48	0.40
22	education2unknown	0.93	1.48	0.55
23	default2other	0.63	0.82	0.19

Así pues, “nr. employed” es la variable que tiene más influencia.

Estimacion de las probabilidades de adquirir el producto:

22. Estima las probabilidades predichas por el modelo para todos los individuos

pr <- predict(mod.glm1,datos2,type="response")

pr

23. Identificamos a los individuos que el modelo predice con mayor y menor probabilidad de comprar el producto y verifica si realmente lo adquirieron o no

ind.max <- which.max(pr)

datos2$y[ind.max]

→ [1] yes, Levels: no yes. Así pues verificamos que para el individuo con mayor probabilidad de comprar el producto ha sido correcta la predicción y si que lo ha comprado.

ind.min <- which.min(pr)

datos2$y[ind.min]

→ [1] no, Levels: no yes. Verificamos que el modelo ha predicho que no lo compraría y la comprobación nos dice que no lo ha comprado.

24. Escogemos un idividuo al azar y miramos la probabilidad predicha y si realmente escogio el producto

#Metodo 1

#ind <- datos2[1000,] # Posicion del individuo

#ind$y

#Metodo 2

ind <- 1000 # Posicion del individuo

datos2$y[ind] → [1] yes (si lo ha comprado)

pr[ind] # Probabilidad de escogerlo → 0.7266956

25. Curva ROC y AUC

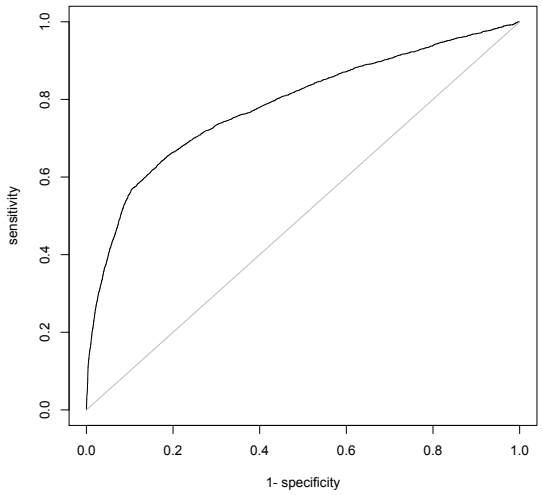
#install.packages("AUC") # Instalar paquete

library("AUC") # Cargar paquete

26. La instruccion roc prepara los datos para luego hacer el grafico de la curva ROC y para calcular el AUC

roc.curve <- roc(pr,datos2$y)

plot(roc.curve)



Recordemos que cuando más abombada sea la curva, mejor será la predicción.

***auc(roc.curve)** = 0.7850414*

- *AUC* [0.75, 0.9) → Discriminación buena.

27. A partir de ahora la empresa decide unicamente llamar a aquellos que tengan una probabilidad mayor de 0.2 de adquirir el producto.

cut.point <- which(roc.curve$cutoffs<0.2)[1] # Posicion del punto de corte para p=0.2

¿Que porcentaje se espera de las llamadas que adquieran el producto?

 # True Positive Rate (TPR) --> De los que realmente hubieran adquirido el producto a que porcentaje llamaremos?

roc.curve$tpr[cut.point]→ 0.5083765

¿Que porcentaje de los que no llamamos hubiesen adquirido el producto? False Positive Rate (FPR) --> De los que realmente NO hubieran adquirido el producto, a que porcentaje llamaremos?

roc.curve$fpr[cut.point] → 0.08082959

2.2.2 PARTE 2: testear resultados (testing)

28. Leemos los datos del conjunto test (testing)

test <- read.table('p2_test (con variable respuesta).csv',header=TRUE,sep=';')

29. Realizamos las transformaciones realizadas en el modelo (conjunto de entrenamiento)

test$id <- NULL **# Remove ID**

test$pdays <- NULL **# Remove PDAYS**

test$job2 <- factor(ifelse(test$job %in% c("retired","student"),"not_active","active"))

test$job <- NULL **#Remove job and grup it in job2**

test$month2 <- factor(ifelse(test$month %in%
c("dec","mar","oct","sep"),"higherOffer","lowerOffer"))

test$month <- NULL **# grup month in 2 categories and remove the default month**

test$education2 <- factor(ifelse(test$education %in%
c("basic.6y","basic.9y"),"educationbasic",as.character(test$education)))

test$education <- NULL **# group education**

test$default2 <- factor(ifelse(test$default=='no','no','other'))

test$default <- NULL **# group default**

#-- Juntamos divorced y married

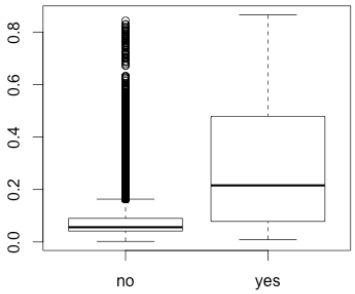
test$marital2 <- factor(ifelse(test$marital %in%
c("divorced","married"),'have_had_commitment','other'))

test$marital <- NULL

30. Representamos las predicciones en función de la respuesta con un boxplot

*pr <- predict(mod.glm1,test**, type="response"**)*

boxplot(pr~test$y)

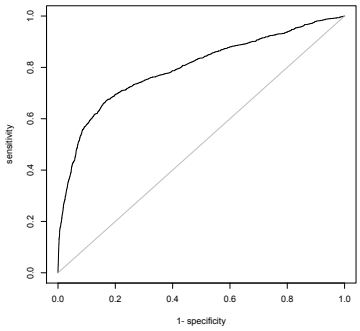


En este gráfico se observa como era de esperar que las probabilidades más bajas (entre 0 y 0.2) la predicción fue correcta y no se tuvo éxito en la venta. Se observa algunos valores de error extremos por encima del tercer cuartil. En cambio en la segunda caja observamos predicciones más altas de éxito correspondiendo al valor real de éxito en la venta.

31. Hacemos la curva ROC y calcula el AUC

roc.curve <- roc(pr,test$y)

plot(roc.curve)



auc(roc.curve) → *0.7962037* → *AUC* [0.75, 0.9) → Discriminación buena

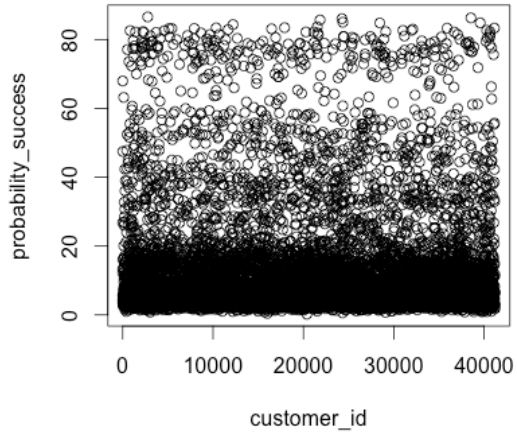
2.3 Guardar probabilidades predichas para cada cliente

#Exportamos en un fichero txt los ids de los usuarios y la probabilidad de éxito.

*df.txt <- data.frame("customer_id"=test$id,"probability_success" = pr*100)*

write.table(df.txt,file="probabilities.txt",sep=",")

**note: in %(probability *100)*



En este grafico observamos como se distribuyen las probabilidades.

Se aprecia que ha un mayor número de probabilidades por debajo del 20%, por lo que podría concluirse que la mayoría de las llamadas no tendran éxito.