# PRA2: Tipologia y Ciclo de Vida de los Datos

Oscar Pedreño y Daniel Villalobos

5/22/2022

## Contents



## Definición del proyecto

En el siguiente proyecto de la asignatura, Tipologia y Ciclo de Vida de los Datos, del máster de la Universitat Oberta de Catalunya, trataremos de predecir el resultado de una campanya de marketing de un banco portuges a través de una base de datos encontrada en la página web UCI Machine Learning Repository.

Se ha escogido esta temática ya que ambos integrantes del equipo procedemos de un area economica, ya sea de estudios, como en el ámbito laboral, por lo que esta base de datos, como veremos más adelante nos permitirá analizar ciertas variables economicas, mediante una regresión logística.

Como ya se ha comentado, se realizará un estudio de estos datos a través de un modelo de regresión logístico. Esto es debido a que nuestra variable respuesta, es una variable categórica con dos niveles, si la persona contactada no contrata o si la persona contactada contrata. Por lo tanto, creemos que la mejor manera de analizar estos datos es a través de un modelo logístico.

## Descripción del dataset

Las variables que se encuentran en este dataset son:

| Nombre | Tipo | Descripción | Valores |
|---|---|---|---|
| age | Numérica | Edat de la persona contactada | |
| job | Categorica nominal | Tipo de trabajo de la persona contactada | *Admin, blue-collar, entrepreneur, housemaid, management, retired, self-employed, services, Student, technician, unemployed* |
| marital | Categorica nominal | Estado civil de la persona contactada | *Divorced (divorciat/da o vidu/a), married, single* |
| education | Categorica nominal | Nivel educativo de la persona contactada | *Basic.9y, high.school, professional.course, university.degree* |
| housing | Categorica binaria | Indica si la persona contactada tiene una hipoteca contratada | *Yes, no* |
| loan | Categorica binaria | Indica si la persona contactada tiene un crédito personal | *Yes, no* |
| contact | Categorica binaria | Tipo de comunicación que se ha realitzado | *Cellular, telephone* |
| month | Categorica nominal | Mes en que se ha contactado por última vez | *Jan, feb, . . . , nov, dec* |
| day_of_week | Categorica nominal | Dia en que se ha contactat por última vez | *Mon, tue, wed, thu, fri.* |
| duration | Numérica contínua | Duración en segundos de el último contacto con la persona | |
| campaign | Numérica discreta | Número de veces que se ha contactado a la persona esta campaña | |
| previous | Numérica discreta | Número de veces que se ha contactado a la persona antes de esta campaña | |
| poutcome | Categorica nominal | Resultado de la campaña de marketing anterior | *Failure, nonexistent, success* |
| emp.var.rate | Numérica contínua | Tasa de varicación de la ocupación del momento en que se ha contactado (Indicador trimestral) | |
| cons.price.idx | Numérica contínua | Indice de precio del consumidor (Indicador mensual) | |
| cons.conf.idx | Numérica contínua | Indice de confianza del consumidor (Indicador mensual) | |
| euribor3m | Numérica contínua | Euribor a 3 meses en el dia del contacto (Indicador diario) | |
| nr.employed | Numérica discreta | Número de trabajadores en la entidad bancaria en el momento del contacto | |
| y | Categorica binaria | Indica si el cliente ha contratado un diposito bancario durante esta campaña | *Yes, no* |

Se han eliminado dos variables de la base de datos original (*default* y *pdays*), ya que, en el primer caso no se sabia interpretar el significado de la variable, y en el segundo caso porque no aportaba más información que la que ya aporta la variable *previous*.

Como ya se ha explicado, la variable respuesta que utilizaremos será, el resultado de la campaña, es decir, si un cliente contratará el crédito durante la campaña o no. En la base de datos esta información esta recogida en la variable *y*.

## Integración y selección de los datos de interés a analizar

Para empezar el análisis primero debemos realizar una lectura de los datos:

```
bd <- read.csv2("bank-additional-full.csv")
head(bd)
```

```
##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
## 3         93.994         -36.4     4.857        5191 no
## 4         93.994         -36.4     4.857        5191 no
## 5         93.994         -36.4     4.857        5191 no
## 6         93.994         -36.4     4.857        5191 no
```

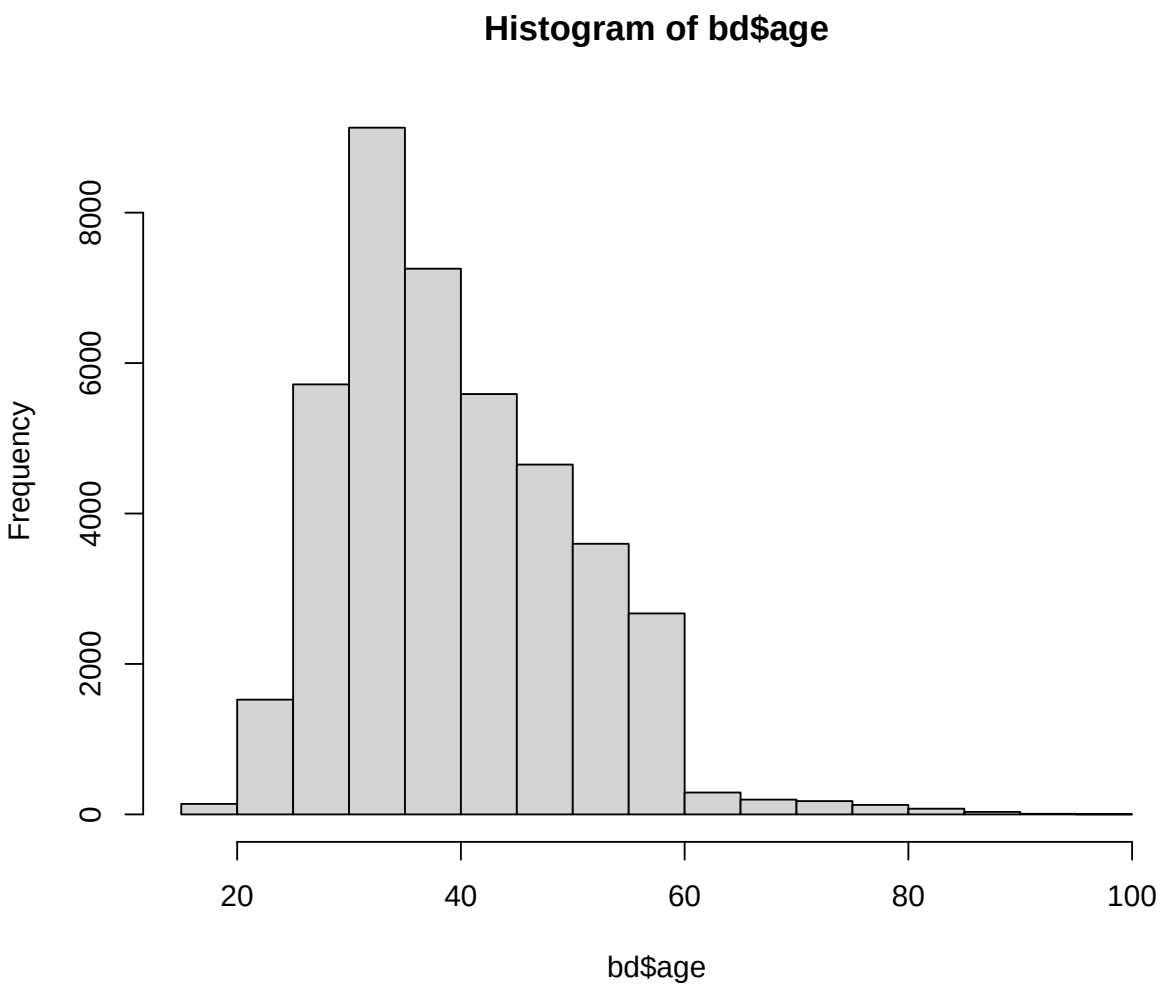
## Limpieza de los datos.

Seguidamente realizaremos un análisis univariante de las distintas variables para poder observar con que clase de valores estamos tratando.

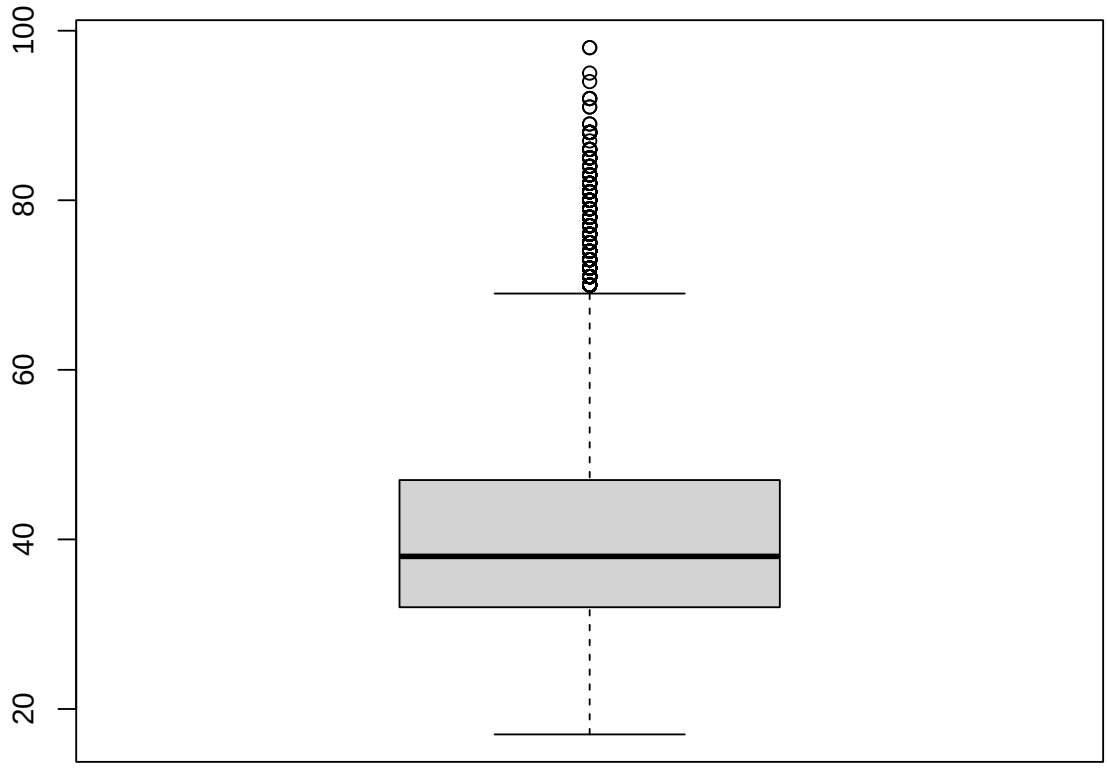
#### Representación gráfica

Como se puede observar la variable *age* contiene valores entre los 17 y los 98 años. La media de edad son 40.0240604. Se observan varios valores outliers en estos datos, aunque no creemos que estos valores vayan a influenciar en el análisis, ya que son valores de entre 65 años y 98, por lo tanto pueden ser valores de edad totalmente asumibles por una persona.

```
hist(bd$age)
```

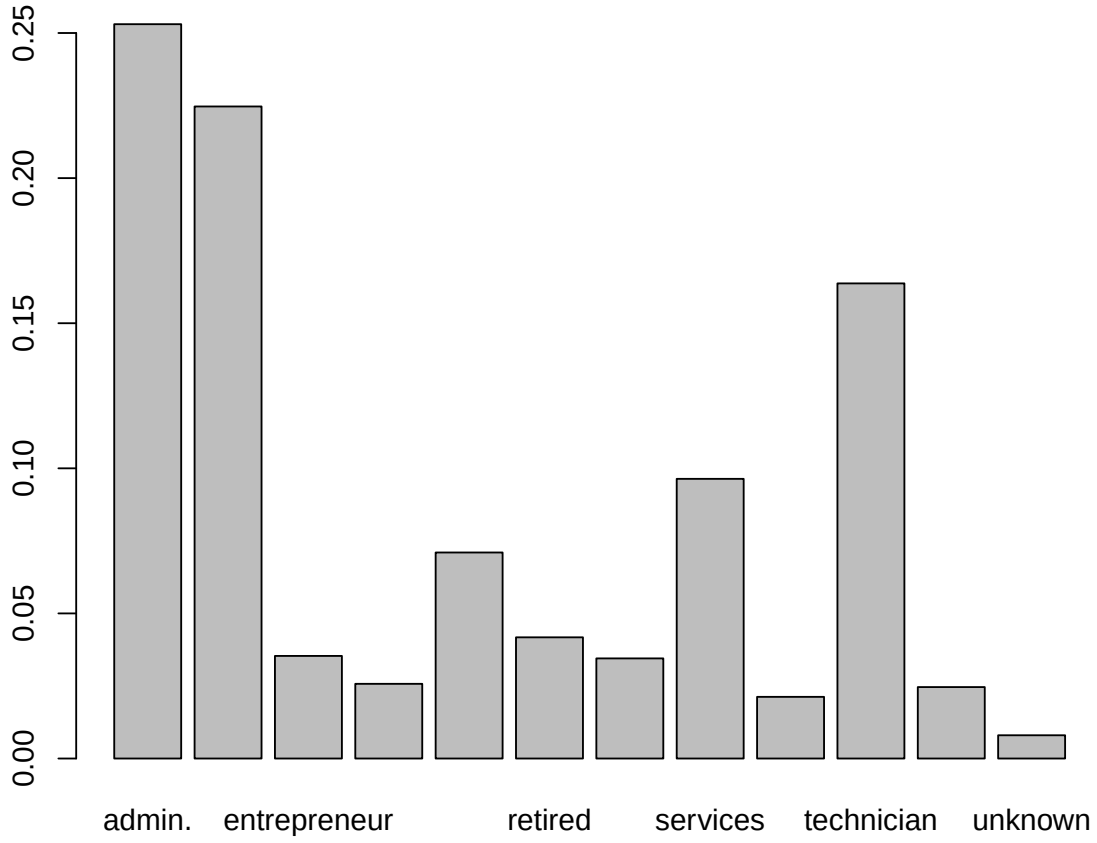



```
boxplot(bd$age)
```



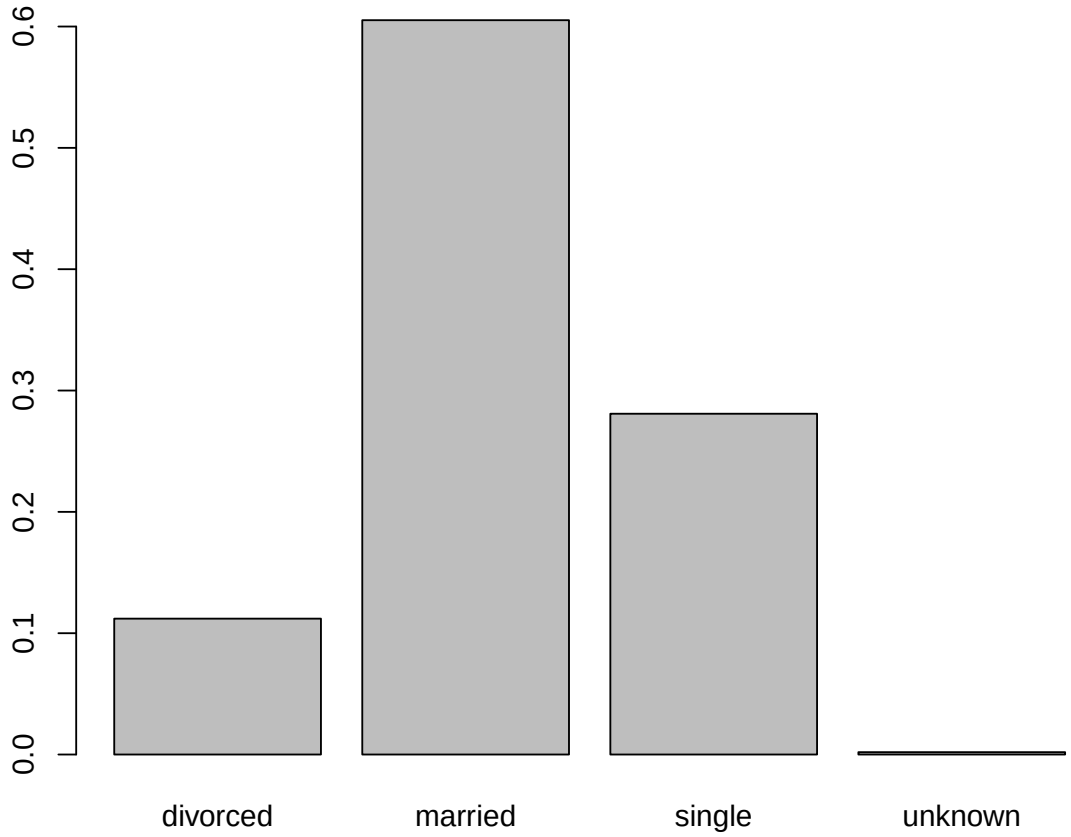

Observamos como las categorias predominantes, para la varible *job* son, *admin, blue-collar, technician*. Además observamos una pequeña sección de registros con un valor *unknown*, estos valores más adelante serán tratados como valores missing. La cantidad de estos valores es: 330.

```
barplot(prop.table(table(bd$job)))
```




En lo que refiere al estado civil de la persona contactada, observamos como el grupo dominante son personas casadas, además observamos una proporción de valores *unkwnon*, que como en el caso de la variable *job* serán tratados como valores faltantes. Tenemos 80 valores *unknown*

```
barplot(prop.table(table(bd$marital)))
```

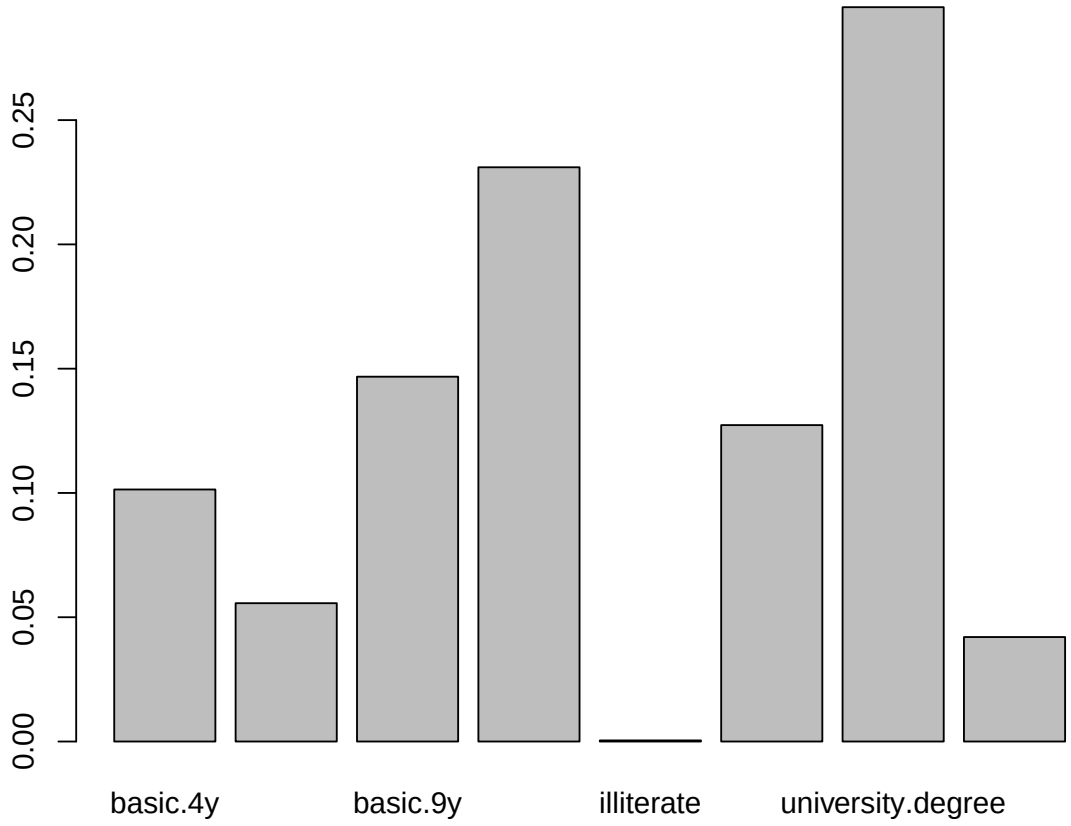



```
prop.table(table(bd$marital))
```

```
##
##    divorced     married      single     unknown
## 0.111974361 0.605224823 0.280858502 0.001942313
```

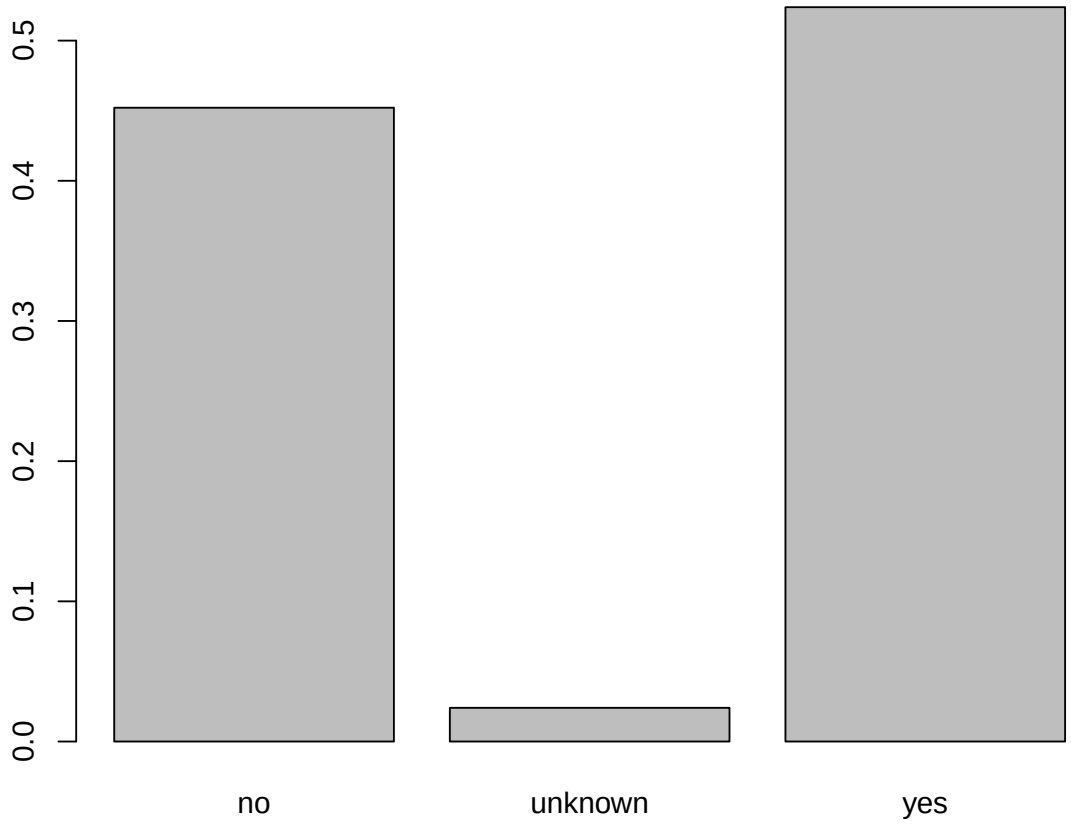
De las personas contactadas el gran grueso se clasifica como personas con un mínimo de estudios secundarios, ya que, las categroias de *high.school y university.degree* suman más del 50%. Se observan 1731 de valores *unknown*.

```
barplot(prop.table(table(bd$education)))
```



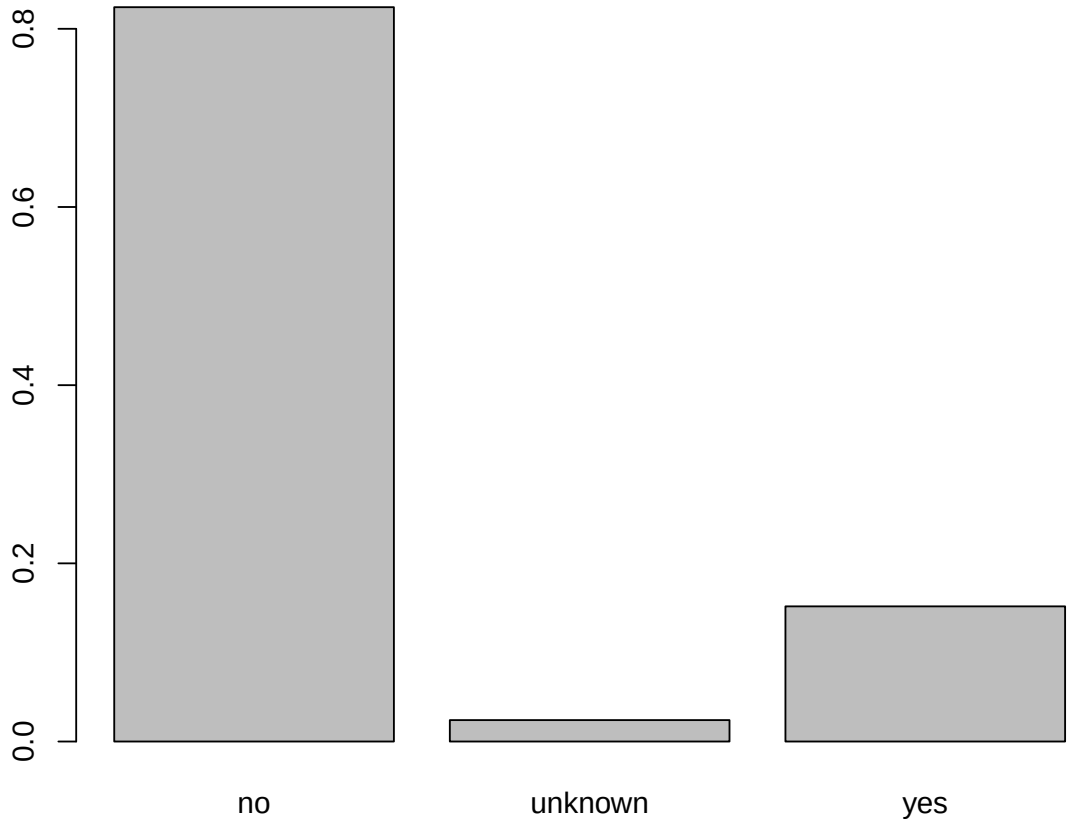

De las personas contratadas no se observa una gran diferencia entra las personas que ya tienen una hipoteca contratada o no, ya que ambos valores oscilan cerca del 50%. Podemos obervar como tenemos 990 de valores *unknown*.

```
barplot(prop.table(table(bd$housing)))
```



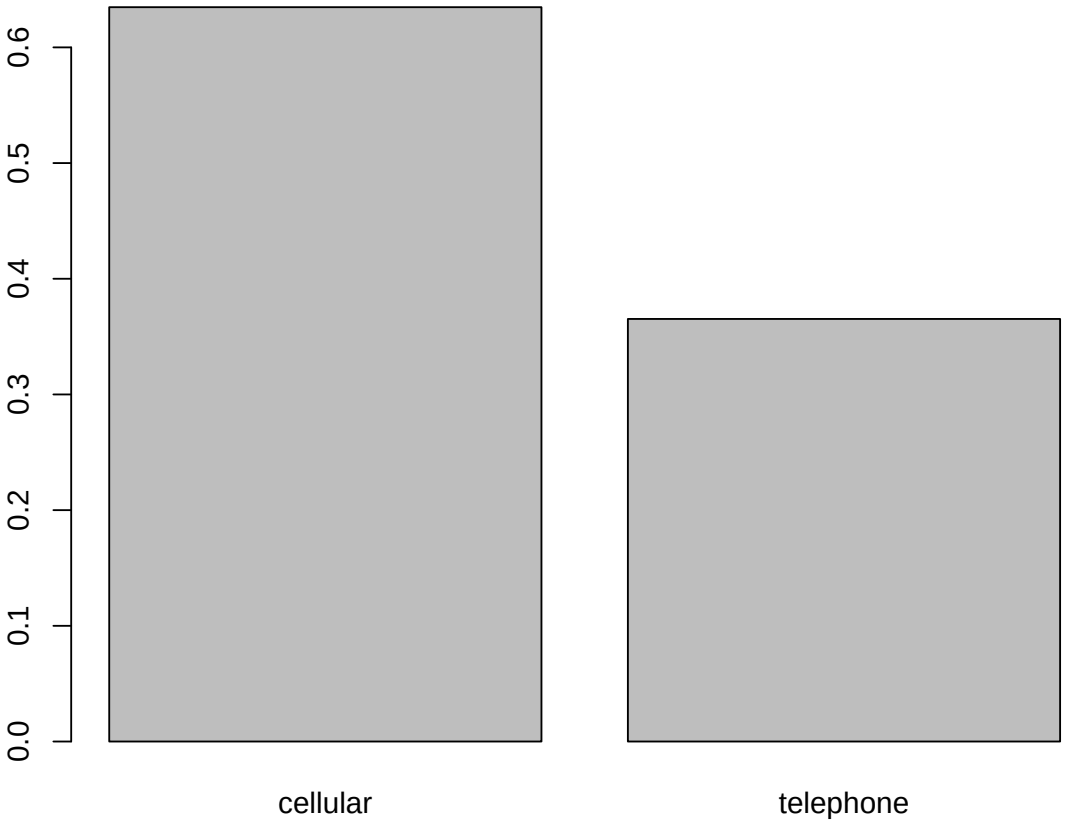

De las personas contactadas se observa como en su mayoria, algo más del 80%, no tienen contratado un crédito personal. Podemos obervar como tenemos 990 de valores *unknown*

```
barplot(prop.table(table(bd$loan)))
```



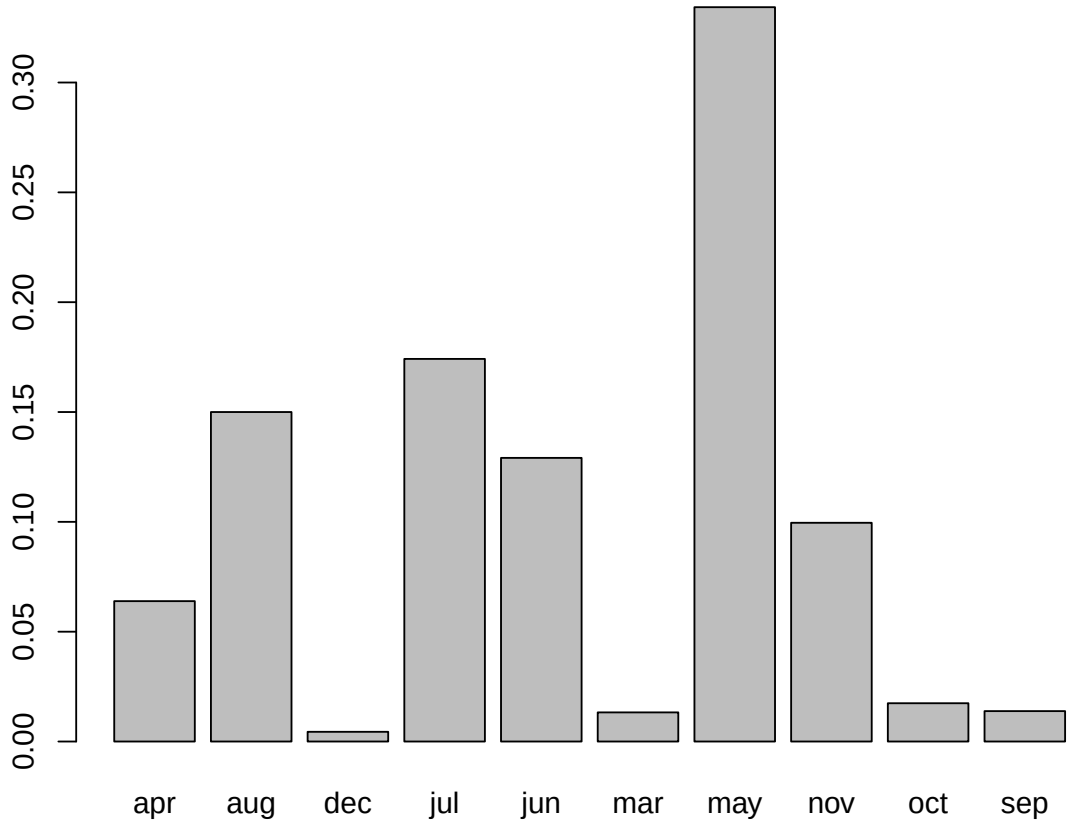

Se observa como el contacto de las personas se ha realizado o vía teléfono fijo o vía teléfono móvil. Se ve como la mayoria de llamadas han sido via teléfono móvil.

```
barplot(prop.table(table(bd$contact)))
```



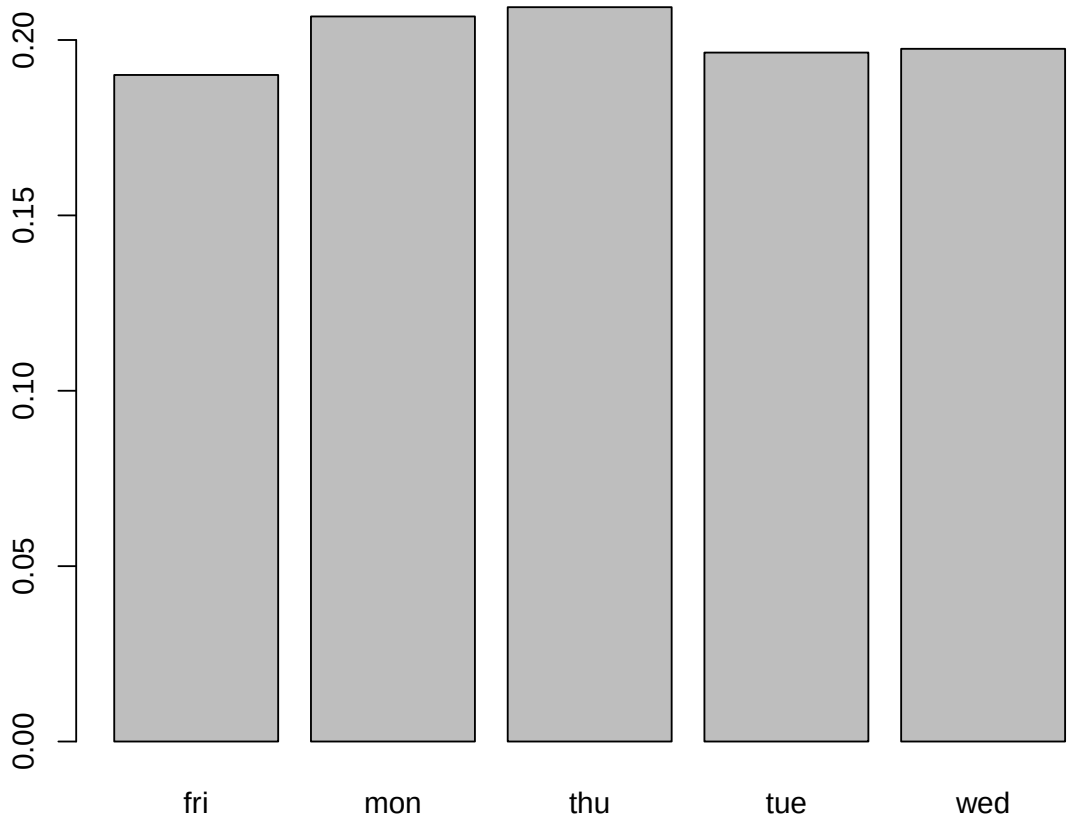

Se observa como la proporción de llamadas es mayor en mayo que en los demás meses, y si hacemos un poco de zoom hacia fuera, vemos como el gran grueso de llamadas se realiza en los meses de verano.

```
barplot(prop.table(table(bd$month)))
```



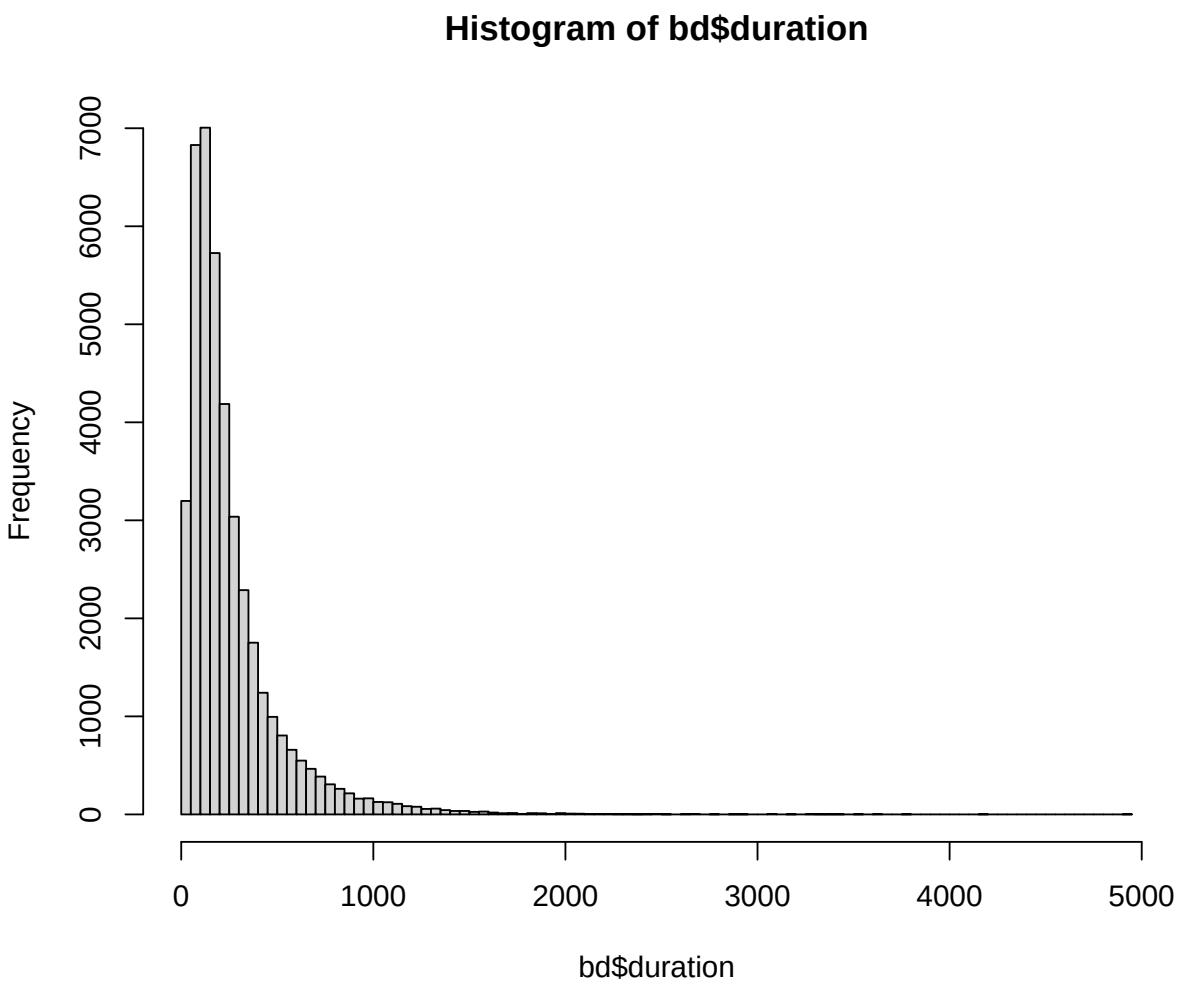

No se observan diferencias entre los distintos días de la semana, lo que si podemos observar es como no hay llamadas fuera de los horarios estandard, es decir no hay llamadas en fin de semana.

```
barplot(prop.table(table(bd$day_of_week)))
```



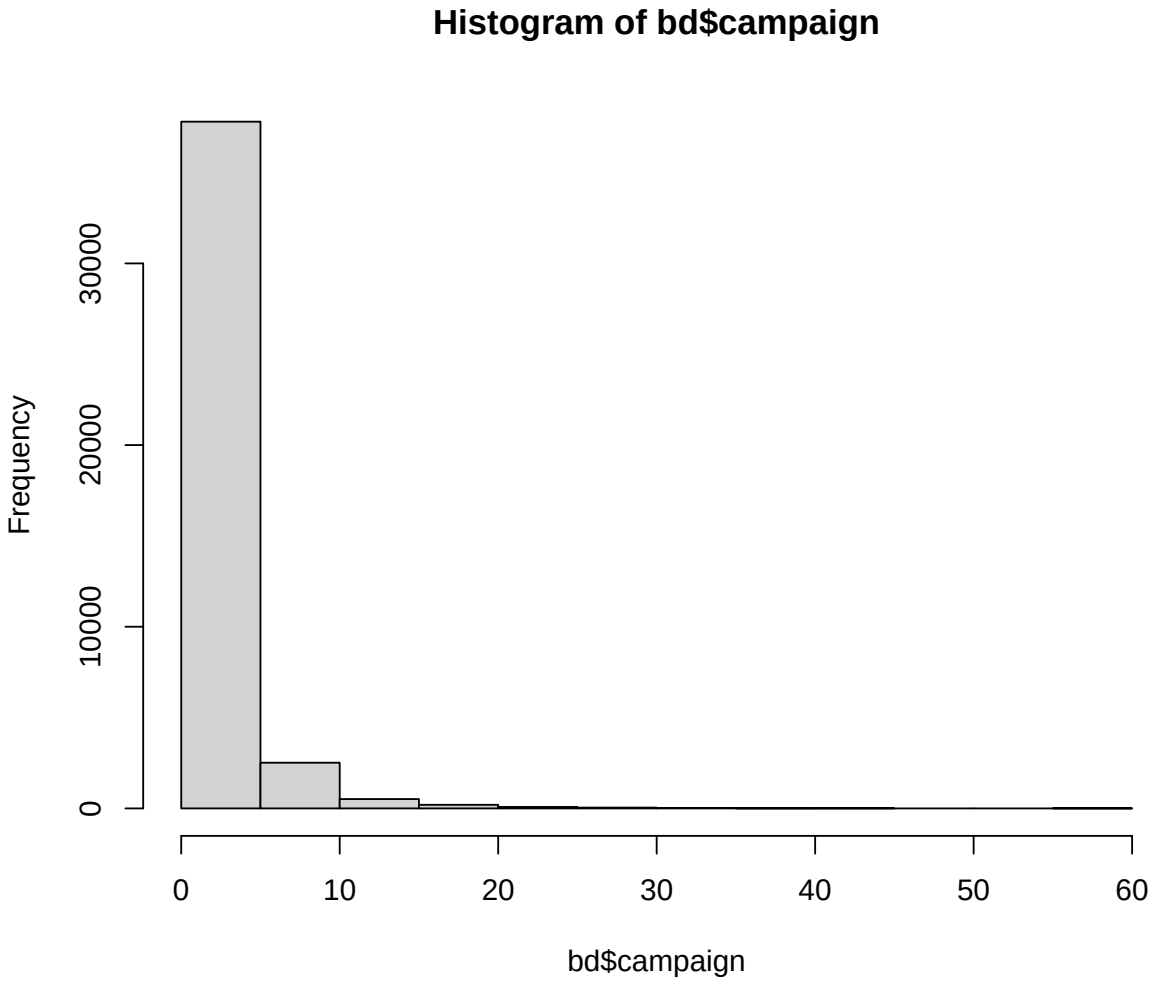

Observamos comom la gran mayoria de llamadas tienen una duración corta, de media tienen una duración de 258.2850102 segundos, pero en este caso tiene más sentido mirar la mediana, ya que valores elevados pueden influenciar sobre el valor de la media, en cuanto a la mediana observamos que las llamadas duran 180. Por lo que no son llamadas muy largas.

```
hist(bd$duration, breaks= 100)
```




Se observa como la mediana de esta variable es 2 por lo que se puede obervar que en su gran mayoria, las personas de esta campaña no han sido contactadas muchas veces.

```
hist(bd$campaign)
```

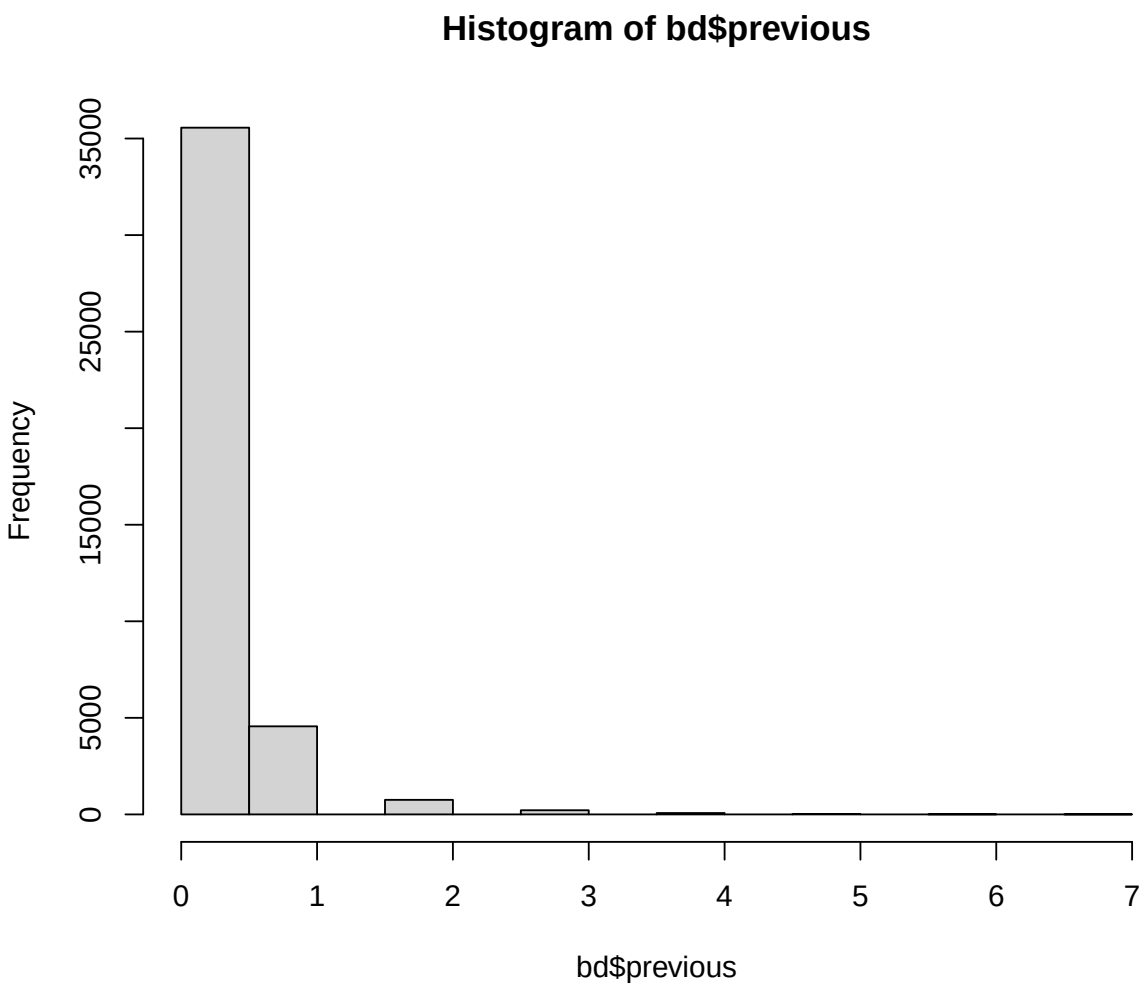



```
summary(bd$campaign)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.568   3.000  56.000
```

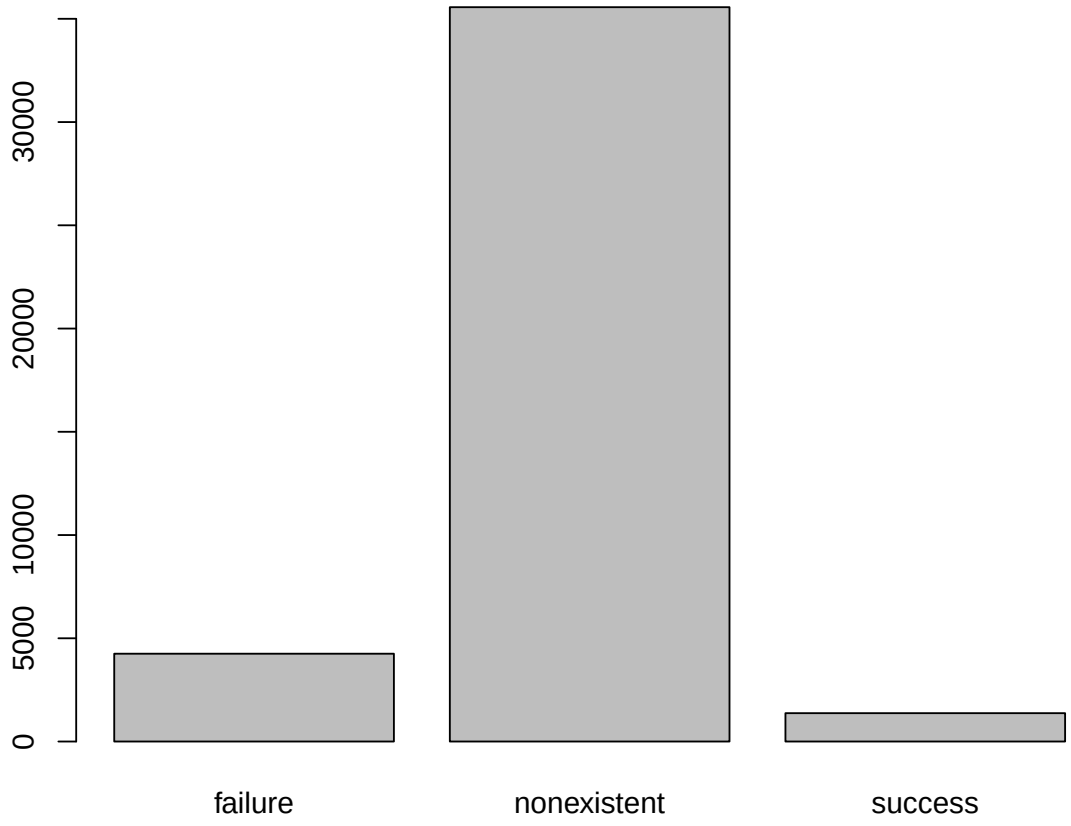
Se observa como la mediana de esta variable es 0 por lo que se puede obervar que en su gran mayoria, las personas de esta campaña no habian sido contactadas con anterioridad.

```
hist(bd$previous)
```




De las personas que anteriormente habian sido contactadas la gran mayoria no contrataron el producto bancario, pero se observa como de las personas de la campaña actual, un gran grueso de estos individuos son nuevos.

```
barplot(table(bd$poutcome))
```




Las siguientes varibles representan distintos indicadores económicos, como la variación mensual del trabajo o el IPC mensual. Los englobaremos todos en la categroria de **Situación Economica actual**. De las cuales por el momento no hay gran cosa a destacar.

```
table(bd$emp.var.rate) #VARIACIÓN TRABAJO TRIMESTRAL
```

```
##
##  -0.1  -0.2  -1.1  -1.7  -1.8  -2.9    -3  -3.4   1.1   1.4
##  3683    10   635   773  9184  1663   172  1071  7763 16234
```

```
table(bd$cons.price.idx) #IPC MENSUAL
```

```
##
## 92.201 92.379 92.431 92.469 92.649 92.713 92.756 92.843 92.893 92.963 93.075
##    770    267    447    178    357    172     10    282   5794    715   2458
##   93.2 93.369 93.444 93.749 93.798 93.876 93.918 93.994 94.027 94.055 94.199
##   3616    264   5175    174     67    212   6685   7763    233    229    303
## 94.215 94.465 94.601 94.767
##    311   4374    204    128
```

```
table(bd$cons.conf.idx) # INDICE DE CONFIANZA CONSUMIDOR
```

```
##
## -26.9 -29.8 -30.1 -31.4   -33 -33.6 -34.6 -34.8 -36.1 -36.4 -37.5 -38.3 -39.8
##   447   267   357   770   172   178   174   264  5175  7763   303   233   229
##   -40 -40.3 -40.4 -40.8 -41.8   -42 -42.7 -45.9 -46.2 -47.1 -49.5   -50 -50.8
##   212   311    67   715  4374  3616  6685    10  5794  2458   204   282   128
```

```
table(bd$euribor3m) #EURIBOR A TRES MESES POR DIA
```

```
##
## 0.634 0.635 0.636 0.637 0.638 0.639  0.64 0.642 0.643 0.644 0.645 0.646 0.649
##     8    43    14     6     7    16    10    35    23    38    26    49    10
##  0.65 0.651 0.652 0.653 0.654 0.655 0.659 0.663 0.668 0.672 0.677 0.682 0.683
##    12     7    35    19    20    34    15    11    18    16    12    39    14
## 0.684 0.685 0.688  0.69 0.692 0.695 0.697 0.699   0.7 0.701 0.702 0.704 0.706
##    16     9     3     3    10    10    23    28     5    11    22     7    18
## 0.707 0.708 0.709  0.71 0.711 0.712 0.713 0.714 0.715 0.716 0.717 0.718 0.719
##    26     6     8     9     4    10     9   139   135    54    18    18    30
##  0.72 0.721 0.722 0.723 0.724 0.727 0.728 0.729  0.73 0.731 0.732 0.733 0.735
##    78     6    74    10    23     5    35    13    24    15     6    18    21
## 0.737 0.739  0.74 0.741 0.742 0.743 0.744 0.748 0.749  0.75 0.752 0.753 0.754
##    25    82    45    27    68    17    17    12     2     7     5     7    44
## 0.755 0.761 0.762 0.766 0.767 0.768  0.77 0.771 0.773 0.778 0.781 0.782 0.788
##     7    22     4     3    17    13    20     6    22     7    21    11    20
##  0.79 0.793 0.797 0.802 0.803 0.809  0.81 0.813 0.819 0.821 0.822 0.825 0.827
##    11     9    38     7    31    21    19     6    22    24    11    27    24
## 0.829 0.834 0.835 0.838  0.84 0.843 0.846 0.849 0.851 0.854 0.859 0.861 0.869
##    13    13    20    29    18    16    21    35    27    30    35    65    54
##  0.87 0.873 0.876 0.877 0.878 0.879  0.88 0.881 0.882 0.883 0.884 0.885 0.886
##    13    82    31    20    33   180    20    79    25   124   128    10    48
## 0.888 0.889  0.89 0.891 0.893 0.894 0.895 0.896 0.898 0.899   0.9 0.903 0.904
##     5    17     8     4    13     3     3    37    39    50    27    12    60
## 0.905 0.908 0.914 0.921 0.927 0.933 0.937 0.942 0.944 0.953 0.956 0.959 0.965
##    17    16     3     2     2     1     2     7     3     2     1    16     5
## 0.969 0.972 0.977 0.979 0.982 0.985 0.987 0.993 0.996     1 1.007 1.008 1.016
##     1    17    21     3    15     7    19     5     1    18     3     5     9
## 1.018 1.025 1.028 1.029  1.03 1.031 1.032 1.035 1.037 1.039  1.04 1.041 1.043
##     3    14     9    44     6     8    16     7     6     9    10     9     9
## 1.044 1.045 1.046 1.047 1.048 1.049  1.05 1.059 1.072 1.085 1.099 1.206 1.215
##    37     1    15     1    22    13    21    23    34     7    11     9    20
## 1.224 1.235 1.244  1.25 1.252 1.259  1.26 1.262 1.264 1.266 1.268  1.27 1.281
##     7     9   422   587    26    70   252   145    87   820    95   110   637
## 1.286 1.291 1.299 1.313 1.327 1.334 1.344 1.354 1.365 1.372 1.384 1.392   1.4
##    16   544   520   492   538   482   395   215   303    10     9    21    13
## 1.405 1.406  1.41 1.415 1.423 1.435 1.445 1.453 1.466 1.479 1.483 1.498  1.51
##  1169    25   254    98    87    81   103    81    57    62    50    35    11
##  1.52 1.531 1.538 1.548 1.556  1.56 1.574 1.584 1.602 1.614 1.629  1.64  1.65
##    17    29    17     6    12     8     1     3     8    13    10    10     8
## 1.663 1.687 1.703 1.726 1.757 1.778 1.799 1.811 3.053 3.282 3.329 3.428 3.488
##    20    22     8    11    20     3    14    31     1     1     1     1     1
## 3.563 3.669 3.743 3.816 3.853 3.879 3.901 4.021 4.076  4.12 4.153 4.191 4.223
##     2     1     1     1     1     2     1   676   822   756   690   610     4
```

```
## 4.245 4.286 4.343 4.406 4.474 4.592 4.663   4.7 4.733  4.76 4.794 4.827 4.855
##     9     7     5     7     3     4     9     8     2     3     5     5   840
## 4.856 4.857 4.858 4.859  4.86 4.864 4.865 4.866 4.912 4.918 4.921 4.936 4.947
##  1210  2868   733   788   892  1044   373   340     7     4     3     6    98
## 4.955 4.956 4.957 4.958 4.959  4.96 4.961 4.962 4.963 4.964 4.965 4.966 4.967
##   103    23   537   581   895  1013  1902  2613  2487  1175  1071   622   643
## 4.968  4.97     5 5.045
##   992   172     7     9
```

```
table(bd$nr.employed) #TREBALLADOR PER TRIMESTRE
```

```
##
## 4963.6 4991.6 5008.7 5017.5 5023.5 5076.2 5099.1 5176.3   5191 5195.8 5228.1
##    635    773    650   1071    172   1663   8534     10   7763   3683  16234
```

### Conteo de missings i eliminación

Realizaremos un breve análisis de los valores missings en cada uno de los registros, esto nos dara un poco más de información de como esta estructurada la base de datos.

```
for (i in 1:nrow(bd)){

  bd$na_count[i] <- sum(bd[i,] == 'unknown')

}
table(bd$na_count)
```

```
##
##     0     1     2     3     4     5
## 30488  9034  1338   306    20     2
```

Como se muestra en la tabla de arriba, hay registros con hasta 5 valores missing, de los cuales, aquellos que tengan más de 3 valores faltantes por registro, al ser una cantidad bastante elevada de missings los eliminaremos de la base de datos. Además eliminaremos de la base de datos la variable default y pdays, la primera por su dificil interpretación, y la segunda, ya que nos aporta la misma información que la variable *previous*.

```
bd <- bd[bd$na_count < 3, ] # Eliminamos si tiene más de 3 unkowns
bd <- bd[, -5] # Eliminamos default
bd <- bd[, -12] # Eliminamos pdays
```

Siguiendo con el preproceso de los datos se ha decidido juntar categorias de la variable education, ya que de esta manera reducimos el número de categorias y se puede realizar un análisis más sencillo. Se han juntado aquellas categroias que tenian educación básica de 9 años o inferior en la categoria *basic.9y*.

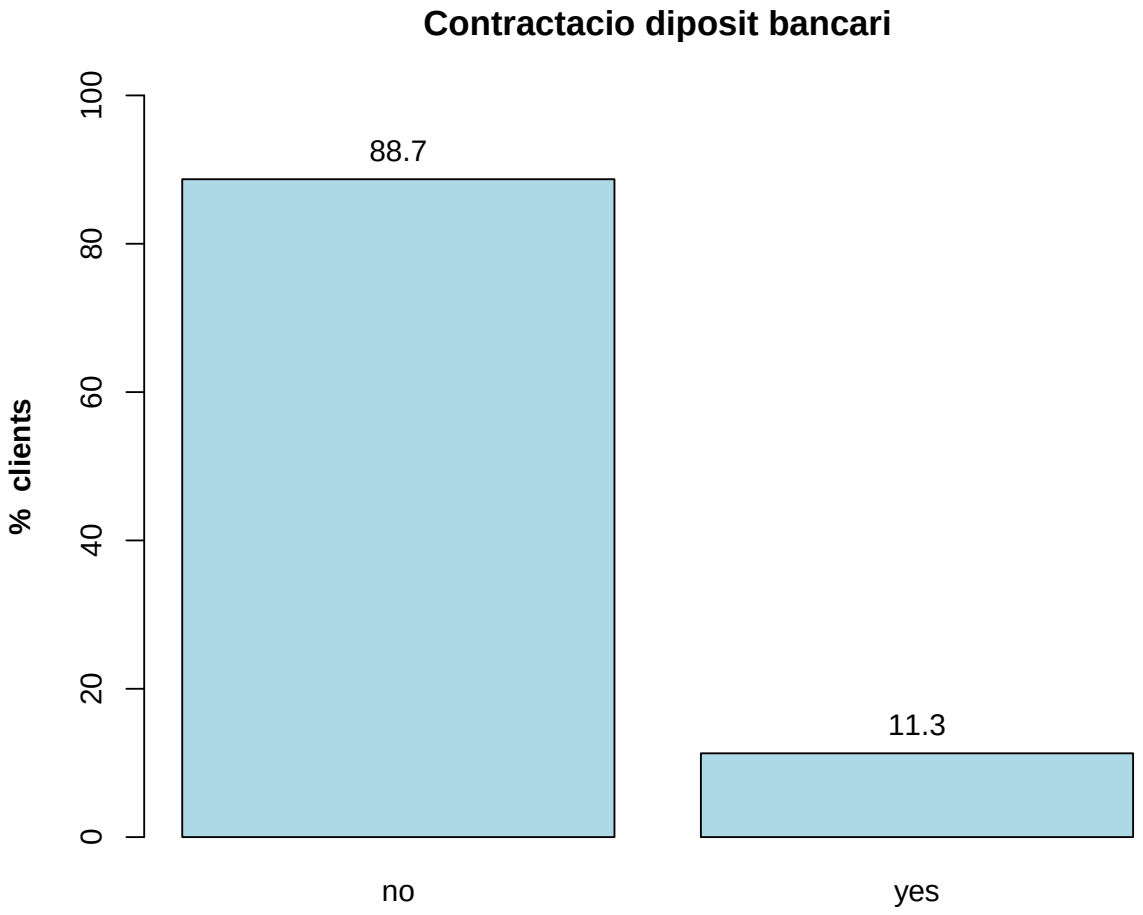
```
for (k in 1:nrow(bd)){
  if(bd$education[k] == 'basic.4y' | bd$education[k] == 'basic.6y' | bd$education[k] == 'basic.9y' | bd$
    bd$education[k] <- 'basic.9y'
  }
}
```

**Representación variable respuesta**

Seguidamente mostraremos un gráfico donde se muestra, en porcentaje, la cantidad de respuestas para cada una de las dos categorias.

```
fig <-  barplot(100*prop.table(table(bd$y)), col = "lightblue", main = "Contractacio diposit bancari", y
                font.lab = 2, ylim = c(0,100), width = 0.5)

text(fig, 100*prop.table(table(bd$y)),  round(100*prop.table(table(bd$y)),3), cex= 1, pos = 3)
```




Se puede observar como la categoria *no* tiene un cantidad muy elevada de registros respecto los registros de la categoria *yes*

Profundizando en el análisis podemos observar como las personas que si han aceptado el credito bancario tienen de media una duración de llamada más elevada que las personas que no han contratado este producto.

```
aggregate(duration~y, data= bd, mean)
```

```
##     y duration
## 1  no 220.9312
## 2 yes 552.7509
```

### Imputación missings

Para realizar la imputación de los valores faltantes, lo primero que hemos realizado, ha sido sobre las variables categoricas que son aquellas que para nuestra base de datos alberga valores missings, pero estos no estan bien formateados, por lo tanto para aquellos registros con una label *unknown* les hemos asignado un valor *NA*.

```
for(i in 1:nrow(bd)){
  for (j in 2:6){
    if(bd[i,j] == 'unknown'){
    bd[i,j] <- NA
    }
    }
}
```

Seguidamente hemos convertido estas variables en factores, para más adelante poder implementar un algoritmo de imputación de valores faltantes.

```
bd[,(2:6)] <- lapply(bd[,2:6], as.factor)

summary(bd)
```

```
##       age                 job            marital
##  Min.   :17.00   admin.     :10391   divorced: 4586
##  1st Qu.:32.00   blue-collar: 9160   married :24699
##  Median :38.00   technician : 6712   single  :11503
##  Mean   :39.98   services   : 3948   NA's    :   72
##  3rd Qu.:47.00   management : 2912
##  Max.   :98.00   (Other)    : 7487
##                  NA's       :  250
##                education     housing        loan         contact
##  basic.9y           :12414   no  :18594   no  :33879   Length:40860
##  high.school        : 9481   yes :21527   yes : 6242   Class :character
##  professional.course: 5223   NA's:  739   NA's:  739   Mode  :character
##  university.degree  :12128
##  NA's               : 1614
##
##
##     month           day_of_week          duration         campaign
##  Length:40860       Length:40860       Min.   :   0.0   Min.   : 1.000
##  Class :character   Class :character   1st Qu.: 102.0   1st Qu.: 1.000
##  Mode  :character   Mode  :character   Median : 180.0   Median : 2.000
##                                        Mean   : 258.4   Mean   : 2.567
##                                        3rd Qu.: 320.0   3rd Qu.: 3.000
##                                        Max.   :4918.0   Max.   :43.000
##
##     previous        poutcome         emp.var.rate       cons.price.idx
##  Min.   :0.0000   Length:40860       Length:40860       Length:40860
##  1st Qu.:0.0000   Class :character   Class :character   Class :character
##  Median :0.0000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.1737
##  3rd Qu.:0.0000
##  Max.   :7.0000
```

```
##
##  cons.conf.idx       euribor3m         nr.employed             y
##  Length:40860       Length:40860       Length:40860       Length:40860
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     na_count
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2866
##  3rd Qu.:1.0000
##  Max.   :2.0000
##
```

Finalmente se ha realizado una imputación de los valores con el algoritmo de *mice*, con el método de *polyreg* que es el adecuado para utilizar variables categóricas. Finalmente asignamos los valores imputados a la base de datos y de esta manera ya no tenemos valores faltantes.

```
imp_mice <- mice(bd[,2:6], m=1, meth = "polyreg" )
```

```
##
##  iter imp variable
##   1   1  job  marital  education  housing  loan
##   2   1  job  marital  education  housing  loan
##   3   1  job  marital  education  housing  loan
##   4   1  job  marital  education  housing  loan
##   5   1  job  marital  education  housing  loan
```

```
summary(imp_mice)
```

```
## Class: mids
## Number of multiple imputations:  1
## Imputation methods:
##       job   marital education   housing      loan
## "polyreg" "polyreg" "polyreg" "polyreg" "polyreg"
## PredictorMatrix:
##           job marital education housing loan
## job         0       1         1       1    1
## marital     1       0         1       1    1
## education   1       1         0       1    1
## housing     1       1         1       0    1
## loan        1       1         1       1    0
```

```
summary(complete(imp_mice))
```

```
##           job            marital                    education     housing
##  admin.     :10451   divorced: 4597   basic.9y           :13010   no :18931
##  blue-collar: 9244   married :24748   high.school        : 9888   yes:21929
```

```
##  technician : 6737   single  :11515   professional.course: 5413
##  services   : 3969                    university.degree  :12549
##  management : 2923
##  retired    : 1718
##  (Other)    : 5818
##   loan
##  no :34509
##  yes: 6351
##
##
##
##
##
```

```
imp_mice2 <- complete(imp_mice)

bd[,2:6] <- imp_mice2
```

## Análisis de los datos.

Antes de poder realizar cualquier tipo de análisis vamos a realizar una *factorización* de las variables categóricas para así poder analizar mejor los datos.

```
levels(bd$y) <- c(0,1)
bd$y <- as.factor(bd$y) # convertimos la varible respuesta a factor

bd[sapply(bd, is.character)] <- lapply(bd[sapply(bd, is.character)],
                                       as.factor)

bd$month <- factor(bd$month, levels = c("mar","apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "
bd$day_of_week <- factor(bd$day_of_week, levels = c("mon", "tue", "wed", "thu", "fri"))
bd$age <- as.numeric(bd$age)
```

Seguidamente para poder realizar un buen estudio separaremos nuestra base de datos en dos conjuntos, uno de entrenamiento y otro de validación. Con el primero lo que haremos será crear el modelo predictivo y conseguir el mejor ajuste posible, y seguidamente con el conjunto de validación comprobaremos que tan bueno es el ajuste.

```
n <- nrow(bd)

set.seed(23531)
learn <- sample(1:n, round(0.75*n))


nlearn <- length(learn)
ntest <- n - nlearn
set.seed(23531)
valid <- sample(1:nlearn, round(0.25*nlearn))
train <- learn[-valid]
```

Seguidamente creamos un primer modelo, este modelo lo que pretende encontrar es la relación de las variables *age, job, marital, education, housing, loan, contact, month, day_of_week, duration, campaign, previous, poutcome.*

Al tener una variable respuesta del tipo categórica binaria se realizará una regresión logística.

```
modelo_1 <- glm(y~
                age + job + marital + education + housing+
                loan + contact + month + day_of_week + duration
                + campaign + previous + poutcome
                , data = bd[train,], family = binomial(link = logit))

summary(modelo_1)
```

```
##
## Call:
## glm(formula = y ~ age + job + marital + education + housing +
##     loan + contact + month + day_of_week + duration + campaign +
##     previous + poutcome, family = binomial(link = logit), data = bd[train,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.5753  -0.3527  -0.2497  -0.1649   3.0550
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.568e+00  2.671e-01  -5.872 4.31e-09 ***
## age                            3.885e-04  3.117e-03   0.125 0.900818
## jobblue-collar                -4.775e-01  1.036e-01  -4.611 4.01e-06 ***
## jobentrepreneur               -4.190e-01  1.662e-01  -2.521 0.011708 *
## jobhousemaid                  -2.270e-03  1.814e-01  -0.013 0.990015
## jobmanagement                 -2.980e-01  1.128e-01  -2.642 0.008239 **
## jobretired                     7.098e-01  1.330e-01   5.338 9.41e-08 ***
## jobself-employed              -3.184e-01  1.513e-01  -2.105 0.035327 *
## jobservices                   -3.867e-01  1.115e-01  -3.469 0.000522 ***
## jobstudent                     6.350e-01  1.396e-01   4.548 5.42e-06 ***
## jobtechnician                 -1.773e-01  8.938e-02  -1.983 0.047325 *
## jobunemployed                 -8.708e-03  1.620e-01  -0.054 0.957126
## maritalmarried                 6.699e-02  8.867e-02   0.756 0.449927
## maritalsingle                  2.252e-01  1.004e-01   2.242 0.024992 *
## educationhigh.school           4.158e-02  8.762e-02   0.474 0.635159
## educationprofessional.course   1.595e-01  1.042e-01   1.532 0.125632
## educationuniversity.degree     1.960e-01  8.822e-02   2.222 0.026307 *
## housingyes                     3.172e-02  5.244e-02   0.605 0.545320
## loanyes                       -4.349e-02  7.284e-02  -0.597 0.550435
## contacttelephone              -1.150e+00  8.068e-02 -14.256  < 2e-16 ***
## monthapr                      -1.707e+00  1.573e-01 -10.852  < 2e-16 ***
## monthmay                      -2.638e+00  1.508e-01 -17.488  < 2e-16 ***
## monthjun                      -1.681e+00  1.617e-01 -10.392  < 2e-16 ***
## monthjul                      -2.769e+00  1.544e-01 -17.933  < 2e-16 ***
## monthaug                      -2.556e+00  1.527e-01 -16.735  < 2e-16 ***
## monthsep                      -9.763e-01  1.963e-01  -4.972 6.62e-07 ***
## monthoct                      -7.058e-01  1.853e-01  -3.809 0.000140 ***
## monthnov                      -2.557e+00  1.597e-01 -16.013  < 2e-16 ***
## monthdec                      -6.889e-01  2.845e-01  -2.422 0.015449 *
## day_of_weektue                 2.822e-01  8.316e-02   3.394 0.000689 ***
## day_of_weekwed                 2.418e-01  8.376e-02   2.887 0.003894 **
```

```
## day_of_weekthu                1.811e-01  8.149e-02   2.222 0.026276 *
## day_of_weekfri                1.486e-01  8.549e-02   1.739 0.082077 .
## duration                      4.139e-03  8.929e-05  46.352  < 2e-16 ***
## campaign                     -5.944e-02  1.452e-02  -4.094 4.23e-05 ***
## previous                      2.614e-01  7.679e-02   3.404 0.000663 ***
## poutcomenonexistent           2.690e-01  1.268e-01   2.121 0.033933 *
## poutcomesuccess               2.335e+00  1.136e-01  20.548  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16123  on 22983  degrees of freedom
## Residual deviance: 10572  on 22946  degrees of freedom
## AIC: 10648
##
## Number of Fisher Scoring iterations: 6
```

Del modelo que hemos construido podemos observar como la variable *duration*, que hemos analizado anteriormente tiene un efecto positivo sobre el resultado de la campaña, además se puede observar como el lunes, dia base, es el peor dia para realizar llamadas, y que en proporcion de aceptación de llamadas el mejor més es marzo. También se puede comprobar como las personas que han sido contactadas con anterioridad son más propensas a aceptar.

El valor de Akaike nos muestra que el modelo, al tener demasiados registros, nos muestra un valor bastante alejado, pero aun así se podría pensar por el momento de que el modelo recoge bastante información de los datos y que puede ser un buen predictor. Seguidamente realizaremos el análisis predictivo correspondiente para analizar la performance de este modelo.

## Resolución del problema

Finalmente valoraremos el ajuste de nuestro modelo calculando la matriz de confusión y el accuracy del modelo, además de una representación gráfica del ajuste del modelo, con la curva ROC.

```
pred <- predict.glm(modelo_1, newdata=bd[train,], type="response")
pred_train <- ifelse(pred > 0.5, 1, 0)
pred_train <- factor(pred_train, levels = c("0", "1"), labels = c("No Contrata", "Contrata"))
```

```
matrizConfusion <- table(bd[train,]$y, pred_train)
matrizConfusion
```

```
##      pred_train
##       No Contrata Contrata
##   no        19928      481
##   yes        1658      917
```

Para realizar el cálculo de la accuracy se debe realizar sobre la predicción realizada con el conjunto de datos de validación.

```
pred_valid <- predict(modelo_1, type = 'response', newdata = bd[valid,])
pred_valid <- ifelse(pred_valid > 0.5, 1, 0)
```

```
pred_valid <- factor(pred_valid, levels = c("0", "1"), labels = c("No Contrata", "Contrata"))
matrizConfusion <- table(bd[valid,]$y, pred_valid)
matrizConfusion
```

```
##      pred_valid
##       No Contrata Contrata
##   no         7088      105
##   yes         344      124
```

```
vp <- matrizConfusion[1,1]
fn <- matrizConfusion[1,2]
vn <- matrizConfusion[2,2]
fp <- matrizConfusion[2,1]
total <- (vp+vn+fn+fp)

accuracy <- (vp + vn)/total
error_rate <- (fp+fn)/total
recall <- vp/(vp+fp)
especificity <- vn/(vn+fn)
```

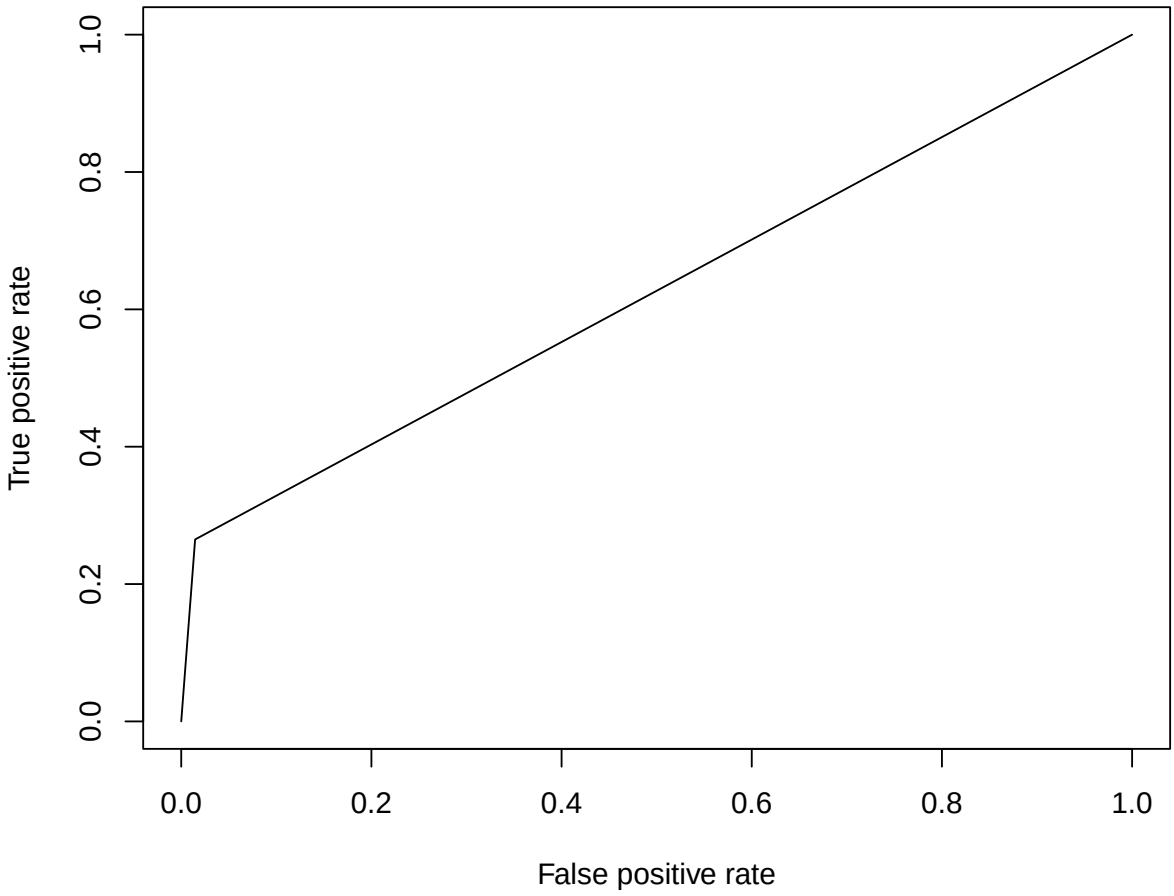
Las métricas obtenidas son las siguientes

| Metric | Value |
|---|---|
| Accuracy | 0.9413915 |
| Error rate | 0.0586085 |
| Recall | 0.9537137 |
| Especificity | 0.5414847 |

Como se puede observar el performance de todos las métricas es bastante bueno, por lo que a prioriri el ajuste del modelo parece ser bastante bueno. El único cálculo que vemos más alejado de tener unos buenos números es el de la Especificidad, la qual se puede observar como solo el 54.1484716% es clasificado como negativo cuando de verdad es negativo, en nuestro caso concreto, solo el 54.1484716% de las personas que contratan el crédito son clasificadas correctamente.

Seguidamente realizaremos un estudio con más profundidad a través de la curva ROC i así esclarecer si realmente el modelo presentado es un buen clasificador y, por lo tanto, predice con bastante exactitud nuestros datos.

```
pred1 <- prediction(as.numeric(pred_valid), as.numeric(bd[valid,]$y))
perf1 <- performance(pred1, "tpr", "fpr")
plot(perf1)
```




Como podemos observar la curva ROC no se aleja mucho de la diagonal, por lo que se puede concluir que pese a que tengamos un accuracy bastante elevado la predicción no es muy buena.