Exploratory analysis for the term deposits Bank marketing campaign

#install packages: # install.packages("ggplot2") # install.packages("dplyr") # install.packages("reshape") # install.packages("corrplot") # install.packages("rcompanion") # install.packages("car") # install.packages("lattice") # install.packages("vcd") # install.packages("cramer")

```
library(vcd)

## Loading required package: grid

library(corrplot)

## corrplot 0.92 loaded

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(reshape)

##
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':
##
##     rename

library(rcompanion)
library(lattice)
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

library(scales)
library(cramer)
```

```
## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit

## The following object is masked from 'package:lattice':
##
##     melanoma
```

#Upload the Dataset:

```
URL<-("https://raw.githubusercontent.com/MaramShriem/-Marketing-
Dataset/main/bank-full.csv")
Dataset<-read.csv(file=URL,header=T,sep=";",na.strings = c(" ","NA"))
```

#Checking NAs':

```
anyNA(Dataset)

## [1] FALSE

#no missing values.
```

#Check duplicates:

```
sum(duplicated(Dataset))

## [1] 0

#no duplicate values.
```

#Show the first 6 rows:

```
head(Dataset)

##   age          job marital education default balance housing loan contact
day
## 1  58   management married  tertiary      no    2143     yes   no unknown
5
## 2  44   technician  single secondary      no      29     yes   no unknown
5
## 3  33 entrepreneur married secondary      no       2     yes  yes unknown
5
## 4  47  blue-collar married   unknown      no    1506     yes   no unknown
5
## 5  33      unknown  single   unknown      no       1      no   no unknown
5
```

```
## 6  35   management married  tertiary      no     231     yes   no unknown
5
##   month duration campaign pdays previous poutcome  y
## 1   may      261        1    -1        0  unknown no
## 2   may      151        1    -1        0  unknown no
## 3   may       76        1    -1        0  unknown no
## 4   may       92        1    -1        0  unknown no
## 5   may      198        1    -1        0  unknown no
## 6   may      139        1    -1        0  unknown no
```

#What is the data type, number of columns and number of rows?

```
str(Dataset)
```

```
## 'data.frame':    45211 obs. of  17 variables:
##  $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : chr  "management" "technician" "entrepreneur" "blue-collar"
...
##  $ marital  : chr  "married" "single" "married" "married" ...
##  $ education: chr  "tertiary" "secondary" "secondary" "unknown" ...
##  $ default  : chr  "no" "no" "no" "no" ...
##  $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
##  $ housing  : chr  "yes" "yes" "yes" "yes" ...
##  $ loan     : chr  "no" "no" "yes" "no" ...
##  $ contact  : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ month    : chr  "may" "may" "may" "may" ...
##  $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ y        : chr  "no" "no" "no" "no" ...
```

#I have 10 categorical variables 7 numeric variables.

```
summary(Dataset)
```

```
##       age            job              marital           education
##  Min.   :18.00   Length:45211       Length:45211       Length:45211
##  1st Qu.:33.00   Class :character   Class :character   Class :character
##  Median :39.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.94
##  3rd Qu.:48.00
##  Max.   :95.00
##    default            balance         housing             loan
##  Length:45211       Min.   : -8019   Length:45211       Length:45211
##  Class :character   1st Qu.:    72   Class :character   Class :character
##  Mode  :character   Median :   448   Mode  :character   Mode  :character
##                     Mean   :  1362
##                     3rd Qu.:  1428
```

```
##                     Max.   :102127
##    contact               day           month              duration
##  Length:45211       Min.   : 1.00   Length:45211       Min.   :   0.0
##  Class :character   1st Qu.: 8.00   Class :character   1st Qu.: 103.0
##  Mode  :character   Median :16.00   Mode  :character   Median : 180.0
##                     Mean   :15.81                      Mean   : 258.2
##                     3rd Qu.:21.00                      3rd Qu.: 319.0
##                     Max.   :31.00                      Max.   :4918.0
##     campaign          pdays          previous          poutcome
##  Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000   Length:45211
##  1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000   Class :character
##  Median : 2.000   Median : -1.0   Median :  0.0000   Mode  :character
##  Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803
##  3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000
##  Max.   :63.000   Max.   :871.0   Max.   :275.0000
##       y
##  Length:45211
##  Class :character
##  Mode  :character
##
##
##
```

#What are the unique instances for the categorical variables?

```
unique(Dataset$job)
```

```
##  [1] "management"    "technician"    "entrepreneur"  "blue-collar"
##  [5] "unknown"       "retired"       "admin."        "services"
##  [9] "self-employed" "unemployed"    "housemaid"     "student"
```

```
unique(Dataset$marital)
```

```
## [1] "married"  "single"   "divorced"
```

```
unique(Dataset$education)
```

```
## [1] "tertiary"  "secondary" "unknown"   "primary"
```

```
unique(Dataset$default)
```

```
## [1] "no"  "yes"
```

```
unique(Dataset$housing)
```

```
## [1] "yes" "no"
```

```
unique(Dataset$loan)
```

```
## [1] "no"  "yes"
```

```
unique(Dataset$contact)
```

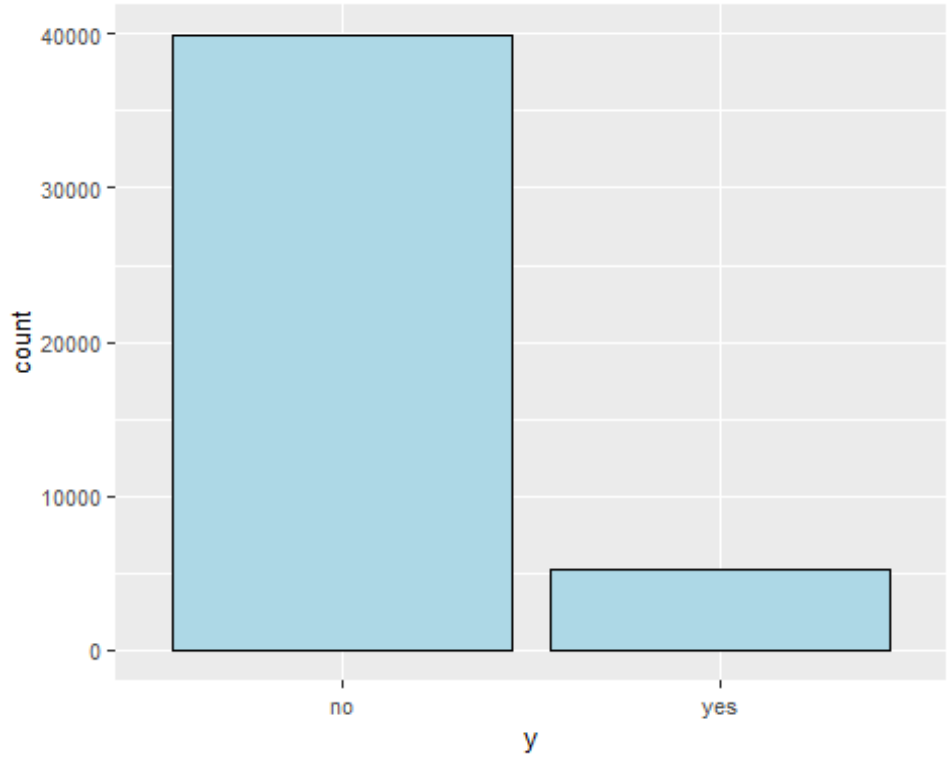
```
## [1] "unknown"   "cellular"  "telephone"

unique(Dataset$month)

##  [1] "may" "jun" "jul" "aug" "oct" "nov" "dec" "jan" "feb" "mar" "apr"
"sep"

unique(Dataset$poutcome)

## [1] "unknown" "failure" "other"   "success"

unique(Dataset$y)

## [1] "no"  "yes"
```

#What is the percentage of customers who will subscribe the product?

```
YesCust<-length(nrow(Dataset)[Dataset$y=="yes"])
round(YesCust/nrow(Dataset)*100,2)

## [1] 11.7
```

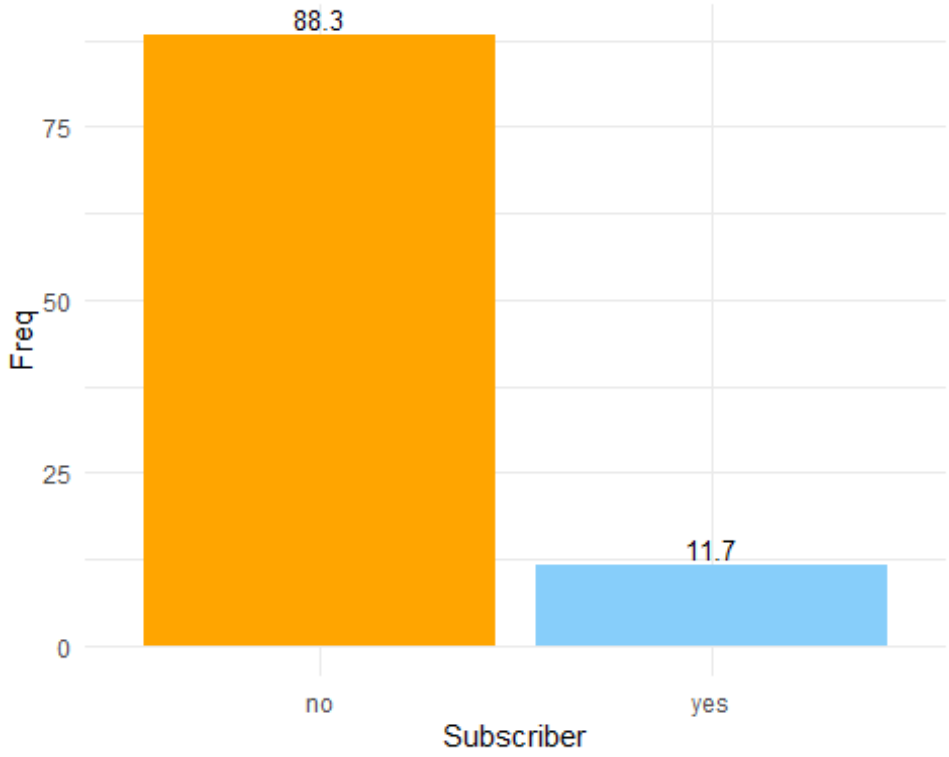
#Plot the count of the desired column y.

```
ycol<-ggplot(Dataset, aes(y))
ycol + geom_bar(color = "black",fill = "light blue") + theme(text =
element_text(size=10))
```



#or the class distribution

```
table<-data.frame(prop.table(table(Dataset$y)))
table$Freq<-round((table$Freq)*100,2)
names(table)[names(table) == 'Var1'] <- 'Subscriber'

ggplot(data=table, aes(x=Subscriber, y=Freq)) +
  geom_bar(stat="identity", fill=c("orange","light sky blue"))+
  geom_text(aes(label=Freq), vjust=-0.3, size=3.5)+
  theme_minimal()
```



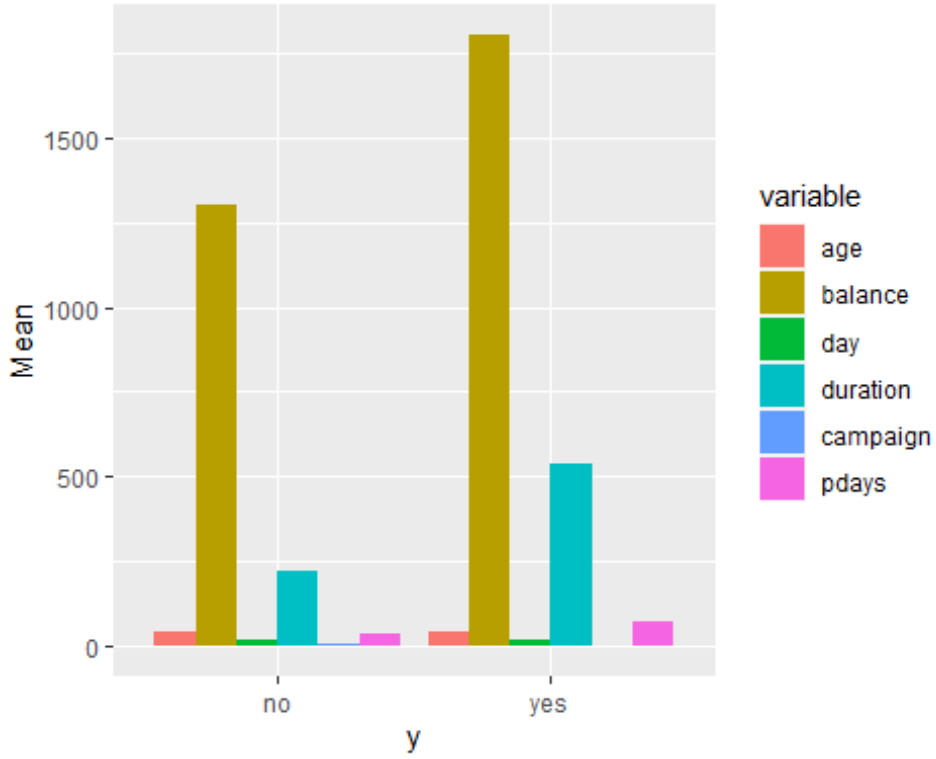

```
#That means the data is imbalanced.
```

```
      Analyzing the numeric variables ( we have 7 numeric attributes):
```

#What are the means and the means distribution for the numerical independent variables on the ys column:

```
NumIndVar<-Dataset[,c(1,6,10,12,13,14,15,17)]
MD<-data.frame(aggregate(.~y,mean,data=NumIndVar))
MD

##     y      age  balance      day duration campaign    pdays  previous
## 1  no 40.83899 1303.715 15.89229 221.1828 2.846350 36.42137 0.5021542
## 2 yes 41.67007 1804.268 15.15825 537.2946 2.141047 68.70297 1.1703536
```

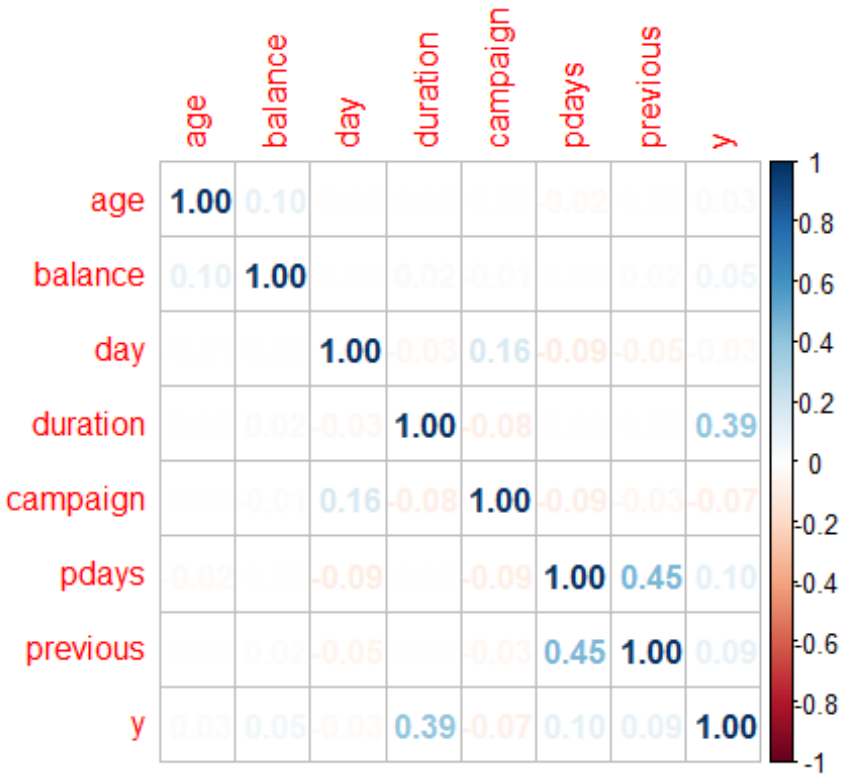
```
dfm<- melt(MD[,-c(8)], id.vars= 1)


ggplot(dfm,aes(x = y, y = value),ylab="Mean") +
  geom_bar(aes(fill = variable),stat = "identity",position = "dodge")+labs(y=
"Mean")
```




```
#balance, duration and pdays have changed obviously.
```

#Correlation for numeric variables:

```
DS<-Dataset
DS$y<-ifelse(DS$y=="yes",1,0)

vars  <-
c("age","balance","day","duration","campaign","pdays","previous","y")
m<-DS[vars]

corrplot(cor(m),method="number")
```




```
#in General the correlation between the variables is week except there is a
slight positive relationship between the y variable( dependent var) and the
duration.
#so let us take a deeper look in this relationship between variable y and the
duration.
```

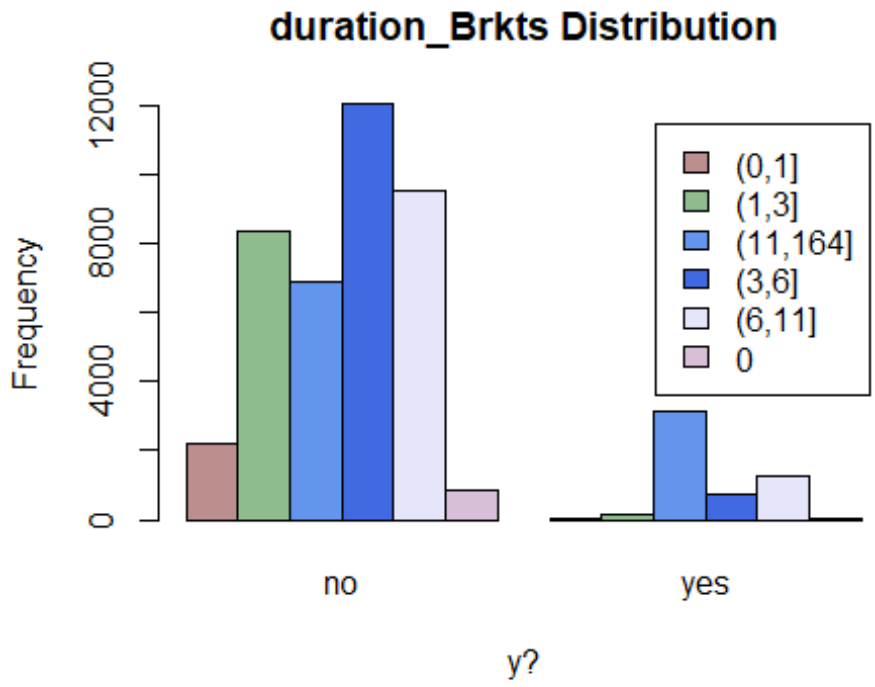
#Analyze the distribution of the duration variable over the y variable

```
#convert the duration seconds to minutes:
Dataset$durationMin<-  round(Dataset$duration/30)

summary(Dataset$durationMin)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   6.000   8.603  11.000 164.000

#based on the results i decided to create 5 brackets as i considered the
duration over 11 min as an outlier

Dataset$duration_Brkts <- cut(Dataset$durationMin,
                              breaks=c(-1,0,1,3,6,11,164))


Dataset$duration_Brkts<-as.character(Dataset$duration_Brkts)
unique(Dataset$duration_Brkts)

## [1] "(6,11]"   "(3,6]"    "(1,3]"    "(11,164]" "(0,1]"    "(-1,0]"
```

```
Dataset$duration_Brkts<-ifelse(Dataset$duration_Brkts=="(-
1,0]","0",Dataset$duration_Brkts)
unique(Dataset$duration_Brkts)
```

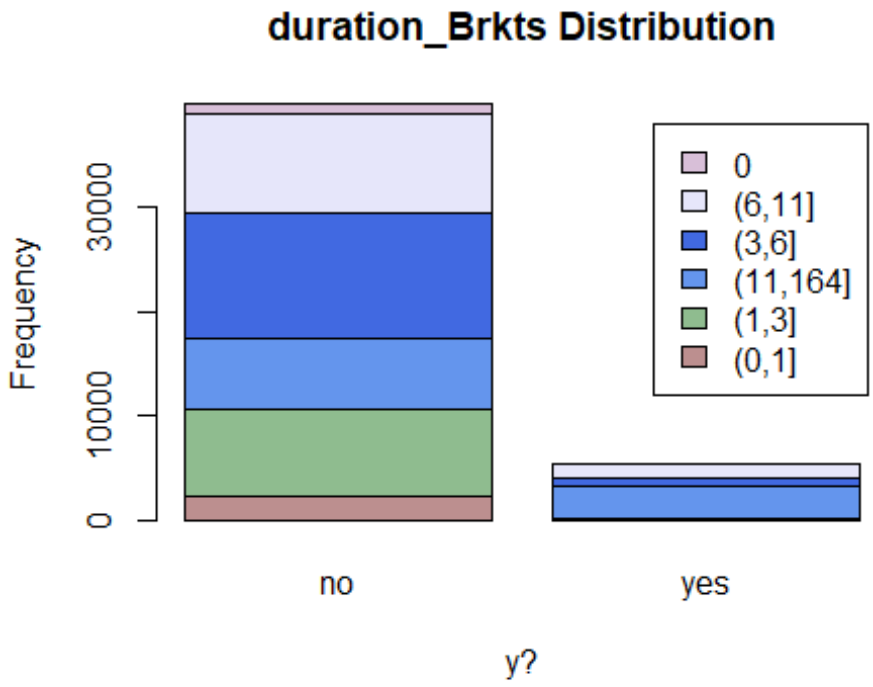
```
## [1] "(6,11]"   "(3,6]"    "(1,3]"    "(11,164]" "(0,1]"    "0"
```

```
#getwd()
#write.csv(Dataset,file="Dataset.csv")

#bar plot to display the distribution of duration_Brkts over the class
variable.

Dataset$y <- factor(Dataset$y)

other_table<-table(Dataset$duration_Brkts,Dataset$y)

barplot(other_table,
        main = "duration_Brkts Distribution",
        xlab = "y?", ylab = "Frequency",
        col = c("rosybrown", "darkseagreen",
"cornflowerblue","royalblue","lavender","thistle"),
        legend.text = rownames(other_table),
        beside = TRUE) # Grouped bars
```




```
#or
barplot(other_table,
        main = "duration_Brkts Distribution",
```

```
        xlab = "y?", ylab = "Frequency",
        col = c("rosybrown", "darkseagreen",
"cornflowerblue","royalblue","lavender","thistle"),
        legend.text = rownames(other_table),
        beside = FALSE) # Stacked bars (default)
```




```
##results:
#call duration =0 and (0,1])  means they will not subscribe the product
#most clients who decided to purchase the  product had a call duration
between 11-164 minutes.
```

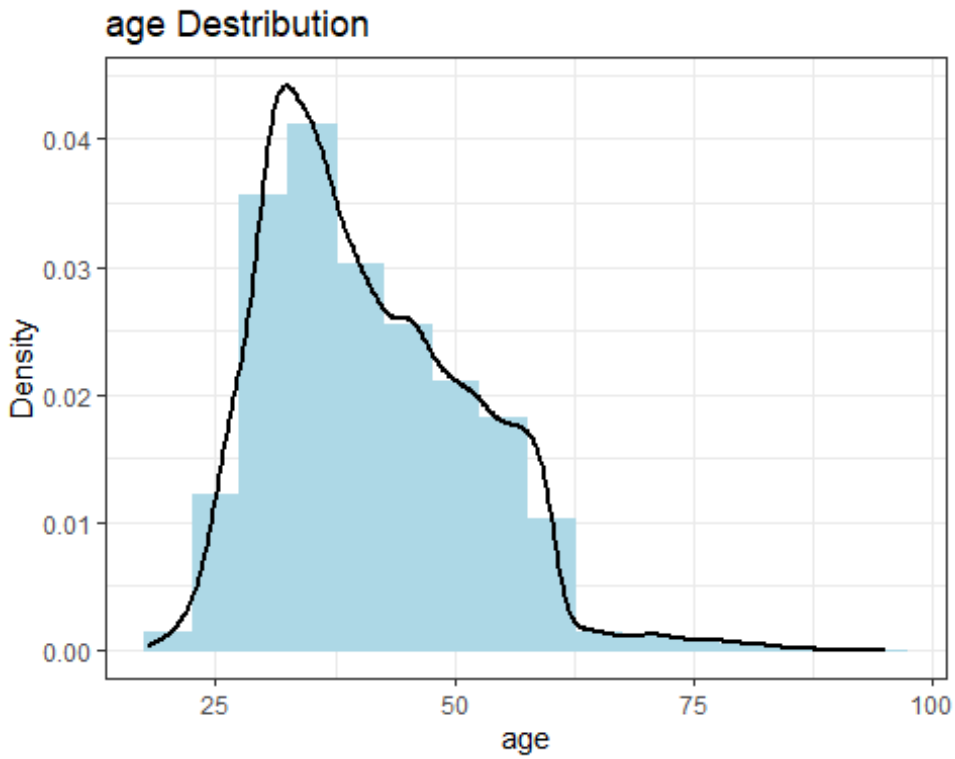
#Correlation test between y and duration variable:

```
cor.test(DS$y,DS$duration)

##
##  Pearson's product-moment correlation
##
## data:  DS$y and DS$duration
## t = 91.289, df = 45209, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3867095 0.4022759
## sample estimates:
##      cor
## 0.394521
```

```
#the P-value is less than 0.05 which means the relationship is statistically
significant.
```

#Lets check the age distribution:

```
ggplot(Dataset, aes(x=age))+
  ggtitle("age Destribution")+
  xlab("age")+
  ylab("Density")+
  theme_bw()+#to make the background in a white color
  geom_histogram(aes(y=..density..),binwidth=5,color="light blue",fill='light
blue')+
  geom_density(linetype="solid",color="black",adjust=1,size=1)
```

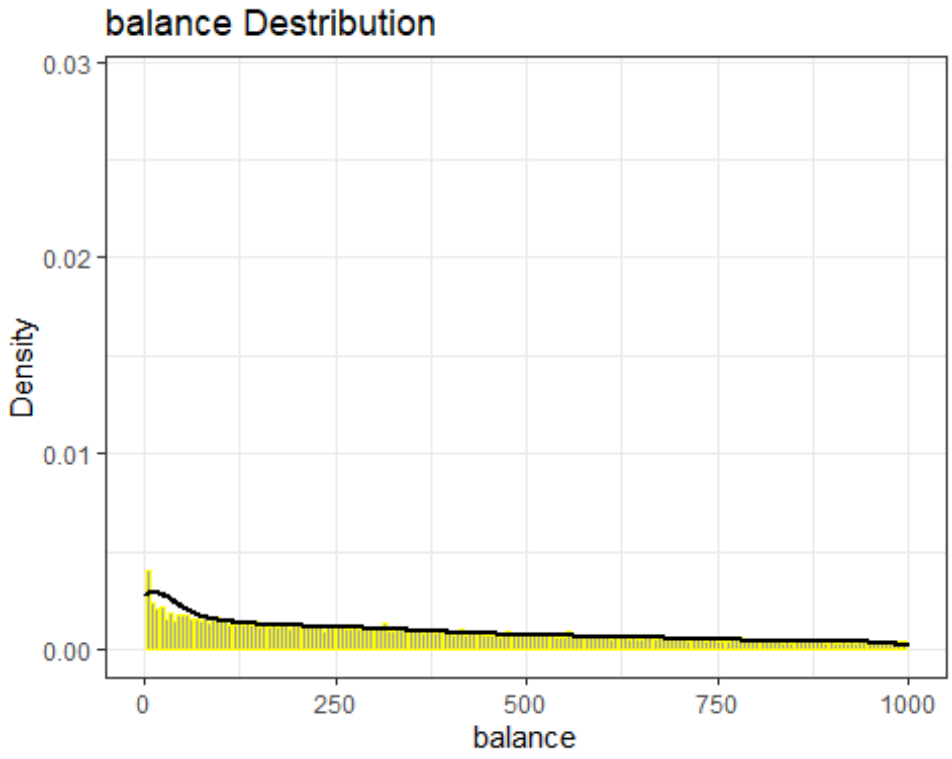



```
#test age normality:

#PS: Shapiro function is to test normality of the variable
#(if the the distribution is normal the  P-Value should be greater than 0.05)

set.seed(10)
x<-sample(Dataset$age,5000)
shapiro.test(x)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  x
## W = 0.96409, p-value < 2.2e-16

#age: not normal distribution and the age is between 30 and 45
```

#Lets check the balance distribution:

```
ggplot(Dataset, aes(x=balance) )+
  ggtitle("balance Destribution")+
  xlab("balance")+
  ylab("Density")+
  xlim(0,1000)+
  theme_bw()+#to make the background in a white color

geom_histogram(aes(y=..density..),binwidth=5,color="yellow",fill='#A4A4A4')+
  geom_density(linetype="solid",color="black",adjust=1,size=1)

## Warning: Removed 18397 rows containing non-finite values (stat_bin).

## Warning: Removed 18397 rows containing non-finite values (stat_density).

## Warning: Removed 2 rows containing missing values (geom_bar).
```

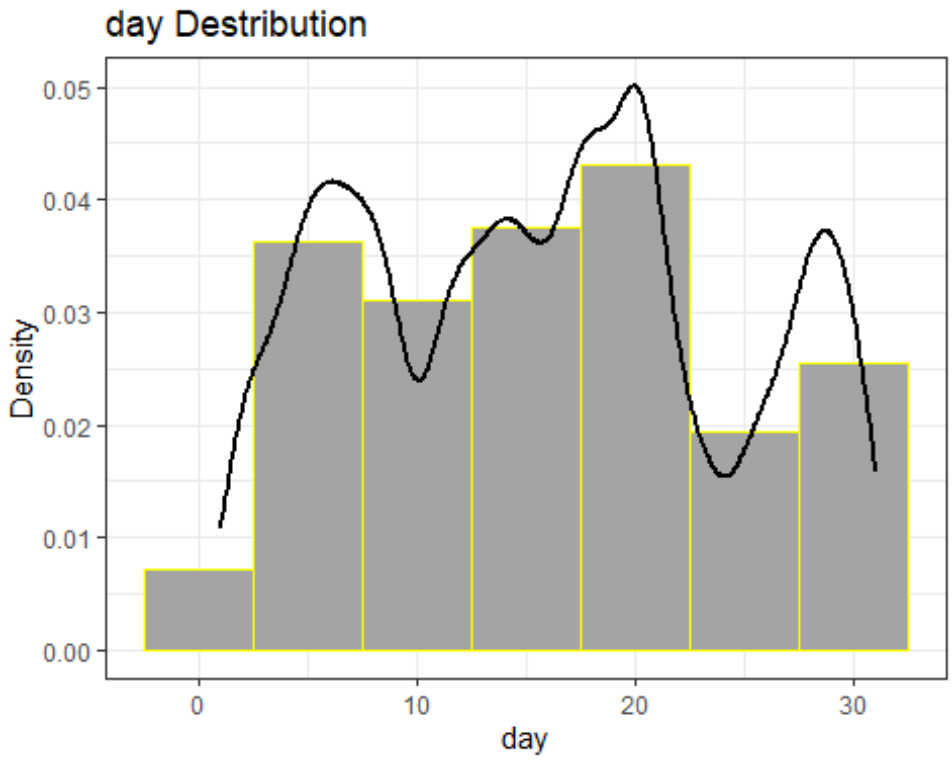



```
#test age normality:

#PS: Shapiro function is to test normality of the variable
#(if the the distribution is normal the  P-Value should be greater than 0.05)
```

```
set.seed(10)
x<-sample(Dataset$balance,5000)
shapiro.test(x)

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.4445, p-value < 2.2e-16

#balance: not normal distribution.
```

#Lets check the day distribution:

```
ggplot(Dataset, aes(x=day) )+
  ggtitle("day Destribution")+
  xlab("day")+
  ylab("Density")+
  theme_bw()+#to make the background in a white color

geom_histogram(aes(y=..density..),binwidth=5,color="yellow",fill='#A4A4A4')+
  geom_density(linetype="solid",color="black",adjust=1,size=1)
```

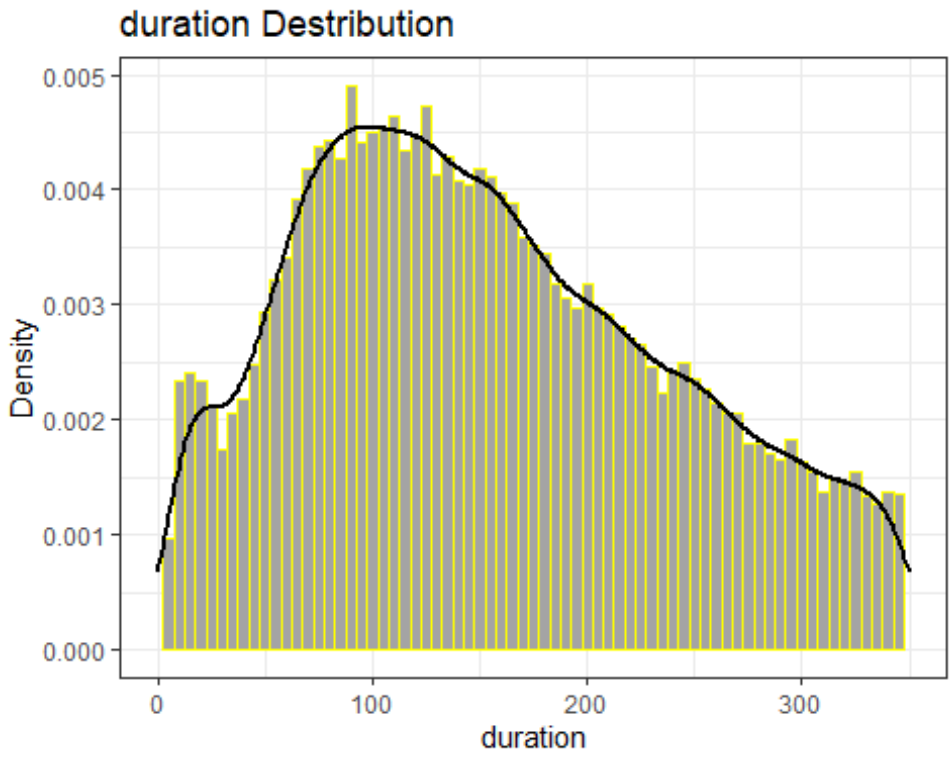



```
#test age normality:

#PS: Shapiro function is to test normality of the variable
#(if the the distribution is normal the  P-Value should be greater than 0.05)
```

```
set.seed(10)
x<-sample(Dataset$day,5000)
shapiro.test(x)

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.96, p-value < 2.2e-16

#day: not normal distribution.
```

#Lets check the duration distribution:(last call duration)

```
ggplot(Dataset, aes(x=duration) )+
  ggtitle("duration Destribution")+
  xlab("duration")+
  ylab("Density")+
  xlim(0,350)+
  theme_bw()+#to make the background in a white color

geom_histogram(aes(y=..density..),binwidth=5,color="yellow",fill='#A4A4A4')+
  geom_density(linetype="solid",color="black",adjust=1,size=1)

## Warning: Removed 9772 rows containing non-finite values (stat_bin).

## Warning: Removed 9772 rows containing non-finite values (stat_density).

## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
#test age normality:

#PS: Shapiro function is to test normality of the variable
#(if the the distribution is normal the  P-Value should be greater than 0.05)

set.seed(15)
x<-sample(Dataset$duration,5000)
shapiro.test(x)

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.71308, p-value < 2.2e-16

#duration: not normal distribution.
```

#Lets check the campaign distribution:(campaign is the number of call during this campaign)

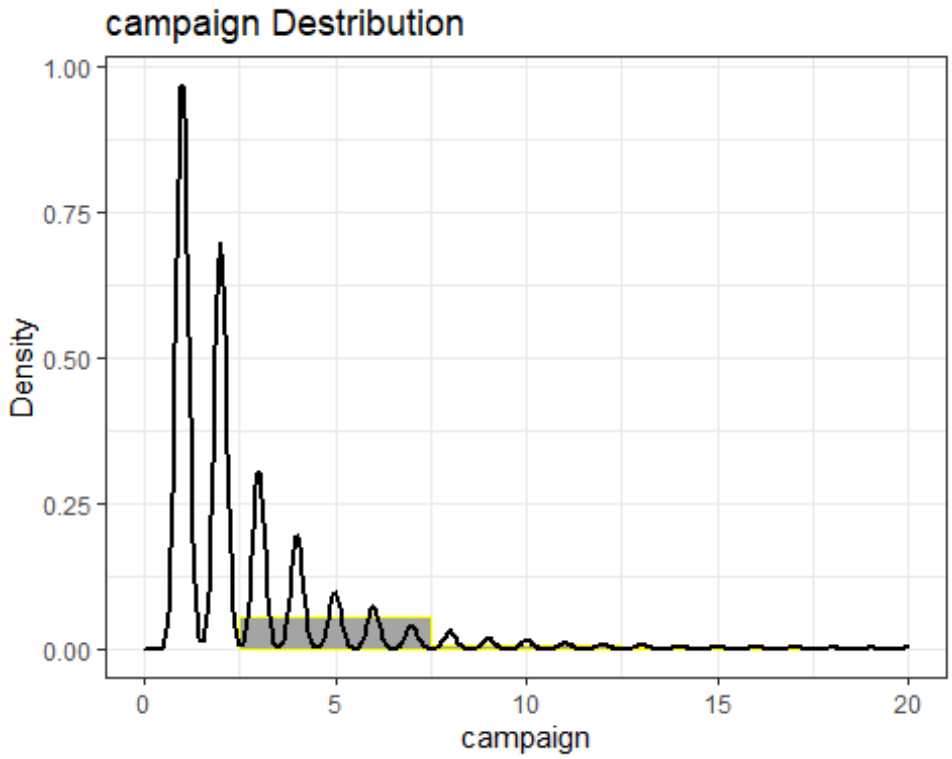
```
ggplot(Dataset, aes(x=campaign) )+
  ggtitle("campaign Destribution")+
  xlab("campaign")+
  ylab("Density")+
  xlim(0,20)+
  theme_bw()+#to make the background in a white color
```

```
geom_histogram(aes(y=..density..),binwidth=5,color="yellow",fill='#A4A4A4')+
  geom_density(linetype="solid",color="black",adjust=1,size=1)
```

```
## Warning: Removed 244 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 244 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
#test age normality:

#PS: Shapiro function is to test normality of the variable
#(if the the distribution is normal the  P-Value should be greater than 0.05)

set.seed(20)
x<-sample(Dataset$campaign,5000)
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.56475, p-value < 2.2e-16
```

```
#campaign: not normal distribution.
```

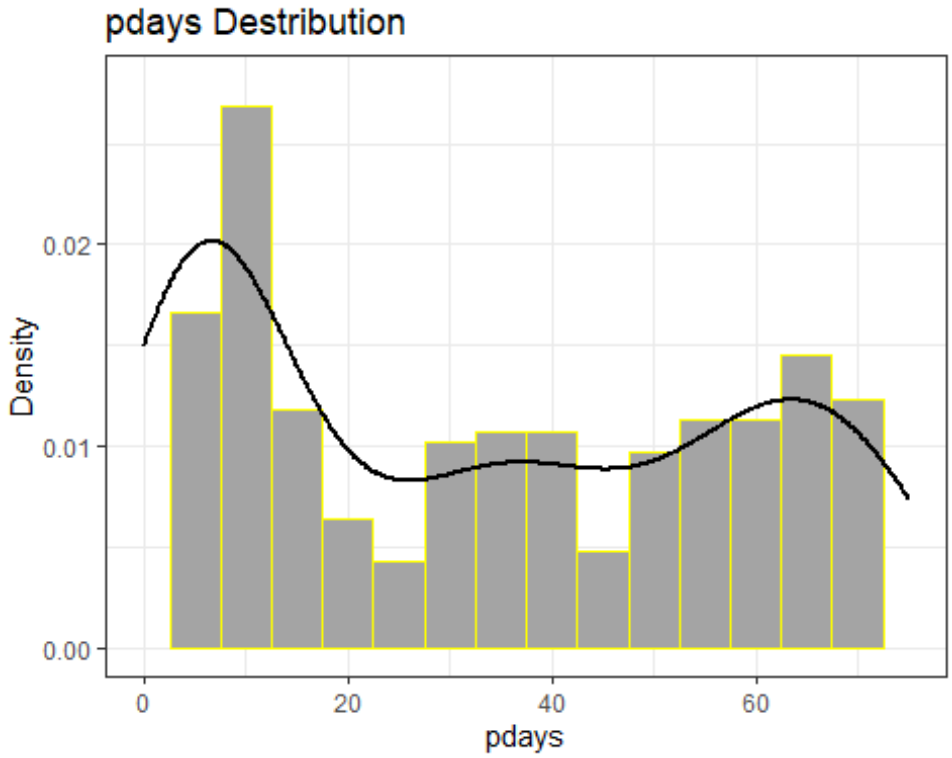
#Lets check the pdays distribution:(pdays is the number of days that passed by after the client was last contacted from a previous campaign)

```
ggplot(Dataset, aes(x=pdays) )+
  ggtitle("pdays Destribution")+
  xlab("pdays")+
  ylab("Density")+
  xlim(0,75)+
  theme_bw()+#to make the background in a white color

geom_histogram(aes(y=..density..),binwidth=5,color="yellow",fill='#A4A4A4')+
  geom_density(linetype="solid",color="black",adjust=1,size=1)
```

```
## Warning: Removed 44839 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 44839 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



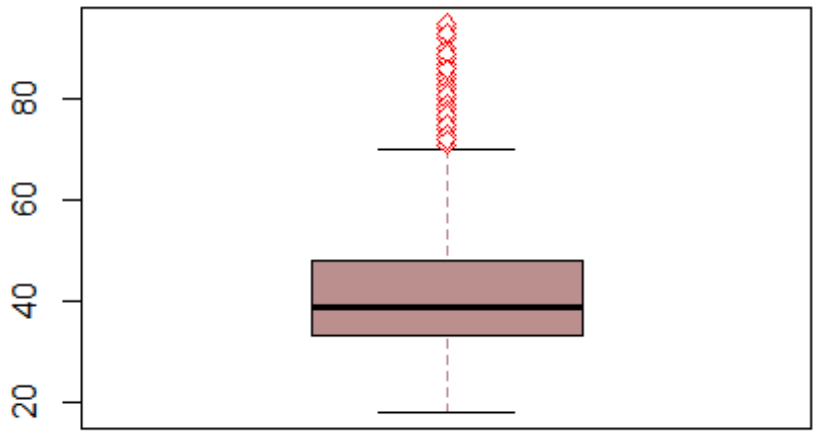

```
#test age normality:

#PS: Shapiro function is to test normality of the variable
#(if the the distribution is normal the  P-Value should be greater than 0.05)

set.seed(20)
x<-sample(Dataset$pdays,5000)
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
```

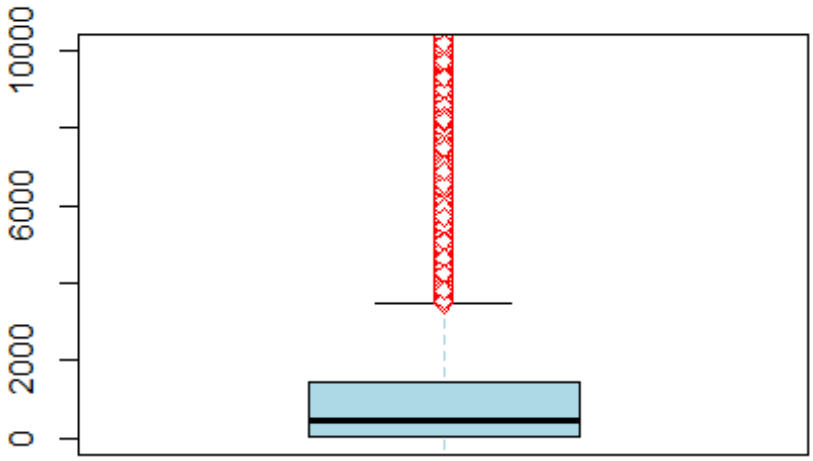
```
##
## data:  x
## W = 0.4707, p-value < 2.2e-16

#pdays: not normal distribution.
```

#Boxplot to check outliars:

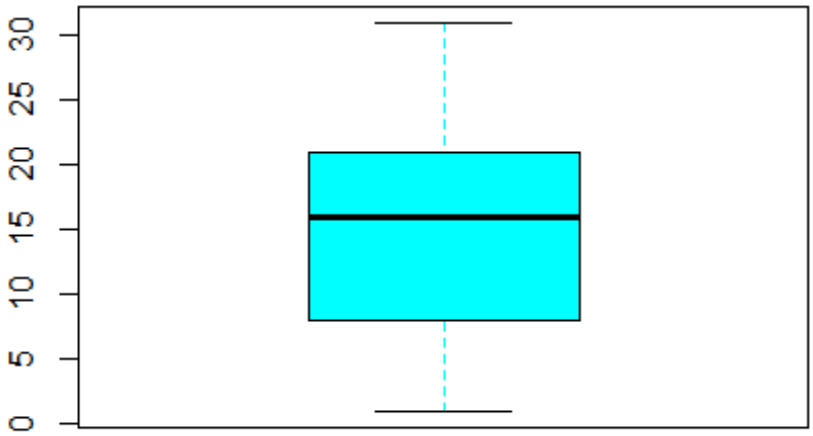
```
boxplot(Dataset$age,outcol="red",pch=23,whiskcol="rosybrown",col="rosybrown",
names = "age")
```



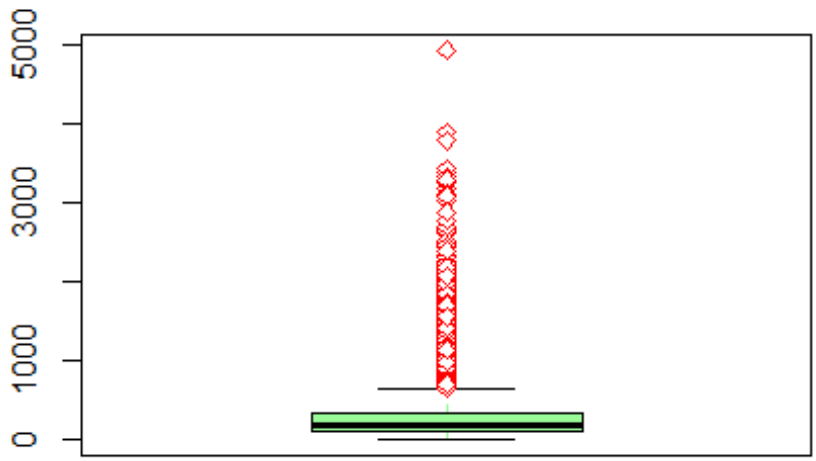

```
boxplot(Dataset$balance,outcol="red",pch=23,whiskcol="lightblue",col="lightbl
ue",names = "balance",ylim= c(0, 10000))
```



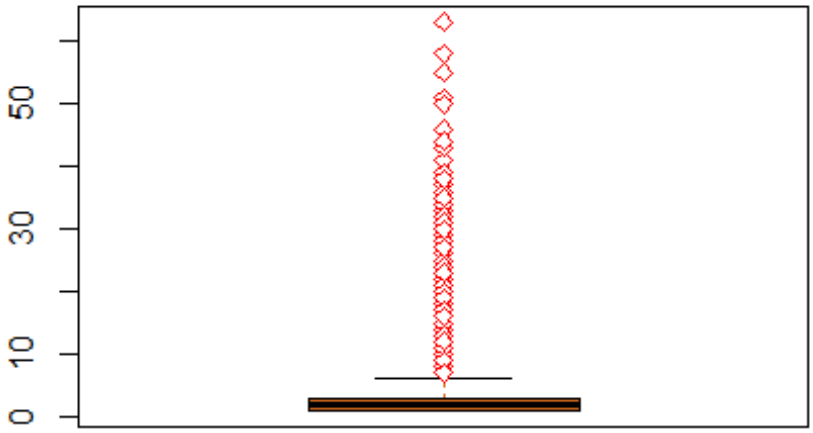

```
boxplot(Dataset$day,outcol="red",pch=23,whiskcol="cyan",col="cyan",names =
"day")
```



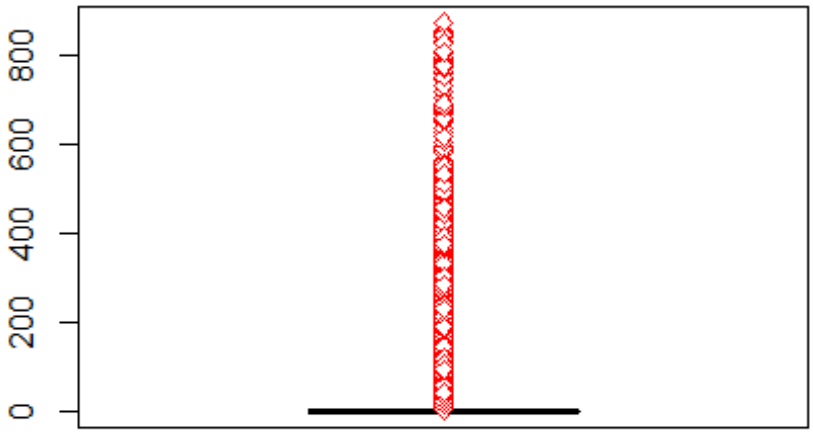
```
boxplot(Dataset$duration,outcol="red",pch=23,whiskcol="palegreen",col="palegr
een",names = "duration")
```



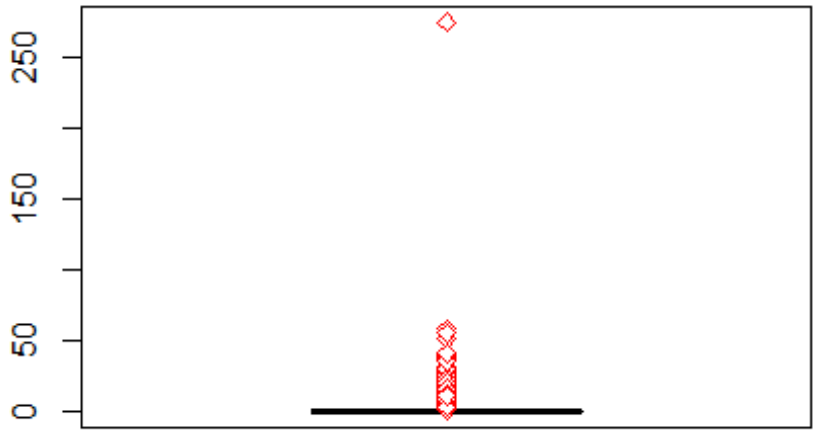

```
boxplot(Dataset$campaign,outcol="red",pch=23,whiskcol="chocolate",col="chocol
ate",names = "campaign")
```




```
boxplot(Dataset$pdays,outcol="red",pch=23,whiskcol="seagreen",col="seagreen",
names = "pdays")
```



```
boxplot(Dataset$previous,outcol="red",pch=23,whiskcol="seagreen",col="seagree
n",names = "previous")
```



#Categorical variable correlation Matrix# #Cramer V is Used to calculate the correlation/association between nominal categorical variables.

```
# 0: The variables are not associated
#- 1: The variables are perfectly associated
#- 0.25: The variables are weakly associated
#- .75: The variables are moderately associated

vars  <-
c("job","marital","education","default","housing","loan","contact","month","p
outcome","y")
df <- Dataset[vars]
# Initialize empty matrix to store coefficients
empty_m <- matrix(ncol = length(df),
                  nrow = length(df),
                  dimnames = list(names(df),
                                  names(df)))


# Function that accepts matrix for coefficients and data and returns a
correlation matrix
calculate_cramer <- function(m, df) {
  for (r in seq(nrow(m))){
    for (c in seq(ncol(m))){
```

```
      m[[r, c]] <- assocstats(table(df[[r]], df[[c]]))$cramer
    }
  }
  return(m)
}

cor_matrix <- calculate_cramer(empty_m ,df)

corrplot(cor_matrix,method="number")
```


```
#only poutcome and month have a weak association with the dependent variable
y.
#month and housing are a very correlated variables.
#education and job are a very correlated variables.
#contact and month are a very correlated variables.
```

#Merge the housing and the loan variables together and recheck the correlation:

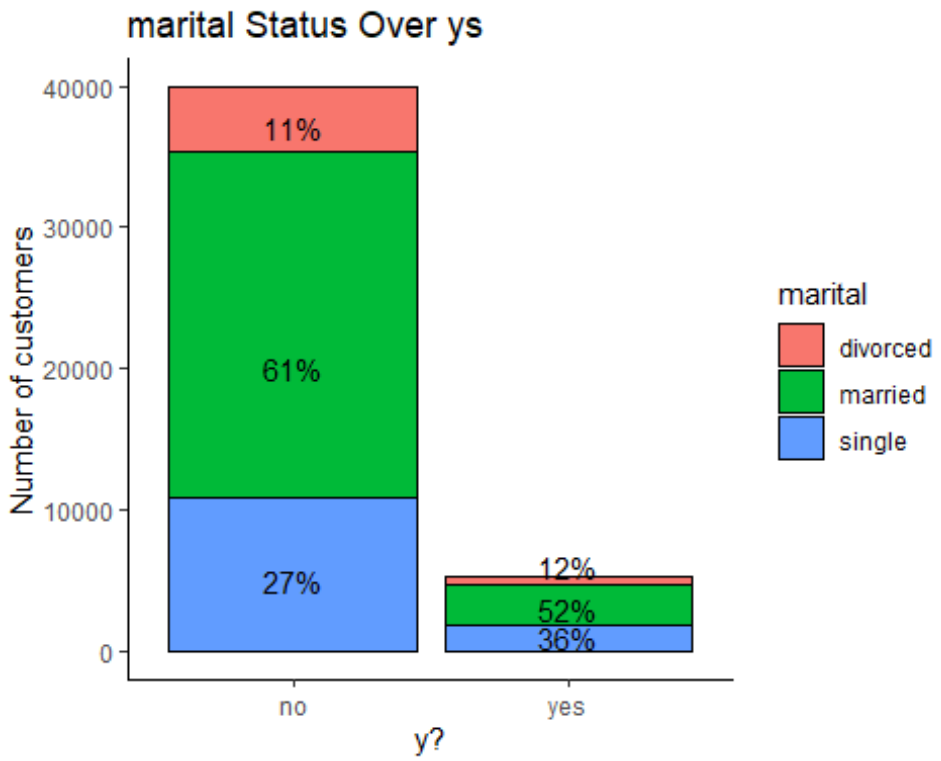
```
DS<-Dataset
DS$Totalloans<-
ifelse(DS$housing=="yes","yes",ifelse(DS$loan=="yes","yes","no"))
DS$y<-factor(DS$y)
cramerV(DS$Totalloans,DS$y)

## Cramer V
##   0.1591
```

```
Df<-DS
Df$Totalloans<-ifelse(Df$Totalloans=="yes",1,0)
Df$y<-ifelse(Df$y=="yes",1,0)
cor.test(Df$Totalloans,Df$y)

##
##  Pearson's product-moment correlation
##
## data:  Df$Totalloans and Df$y
## t = -34.263, df = 45209, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1680633 -0.1500943
## sample estimates:
##       cor
## -0.159092

#correlation is negative weak relationship but statistically significant (the
Pvalue is less than 0.05)
```

#Marital Status percentages over the desired variable y###

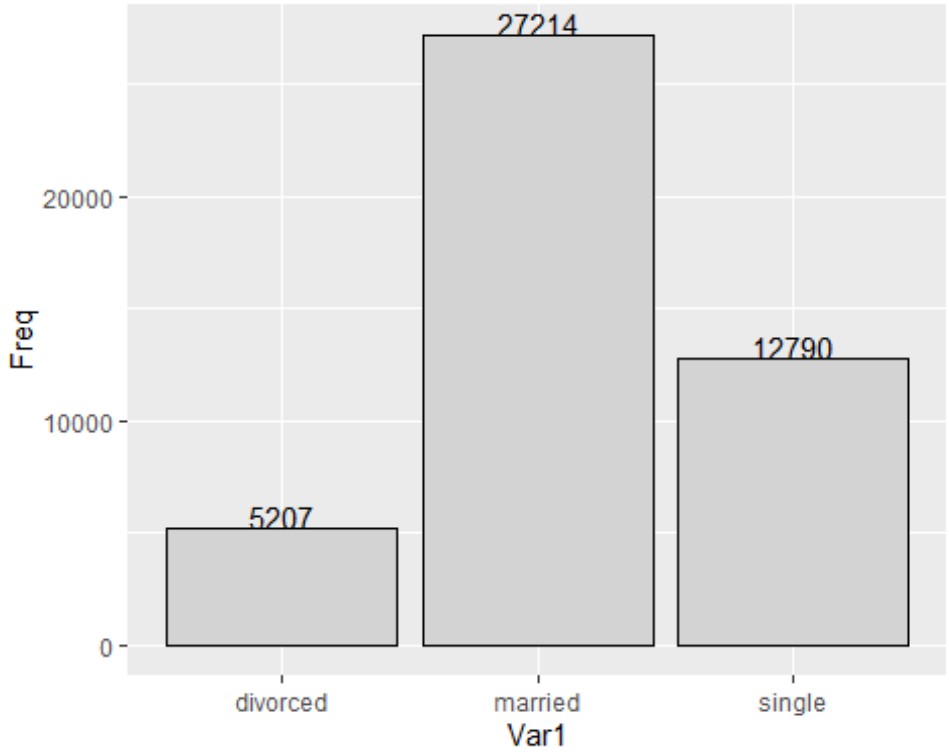
```
DS<-Dataset
DS$y<-ifelse(DS$y=="yes",1,0)

subDiv<-
round((aggregate(DS$y,by=list(DS$marital),sum)[1,2]/length(which(DS$y==1))),3
)
subMarr<-
round((aggregate(DS$y,by=list(DS$marital),sum)[2,2]/length(which(DS$y==1))),3
)
subsin<-
round((aggregate(DS$y,by=list(DS$marital),sum)[3,2]/length(which(DS$y==1))),3
)
subDiv<-label_percent()(subDiv)
subMarr<-label_percent()(subMarr)
subsin<-label_percent()(subsin)

NsubDiv<-round((length(which(DS$marital=="divorced"))-
aggregate(DS$y,by=list(DS$marital),sum)[1,2])/length(which(DS$y==0)),4)
NsubMarr<-round((length(which(DS$marital=="married"))-
aggregate(DS$y,by=list(DS$marital),sum)[2,2])/length(which(DS$y==0)),4)
Nsubsin<-round((length(which(DS$marital=="single"))-
aggregate(DS$y,by=list(DS$marital),sum)[3,2])/length(which(DS$y==0)),4)
NsubDiv<-label_percent()(NsubDiv)
NsubMarr<-label_percent()(NsubMarr)
Nsubsin<-label_percent()(Nsubsin)

ggplot(Dataset,aes(x=y,fill=marital))+
  geom_bar(colour="black",width = .9)+
```

```
  theme_classic(base_size = 11)+# background theme
  labs(y="Number of customers",x="y?",title ="marital Status Over ys" )+
  annotate(geom="text",x=2, y=1000, label=subsin)+
  annotate(geom="text",x=2, y=3000, label=subMarr)+
  annotate(geom="text",x=2, y=6000, label=subDiv)+
  annotate(geom="text",x=1, y=5000, label=Nsubsin)+
  annotate(geom="text",x=1, y=20000, label=NsubMarr)+
  annotate(geom="text",x=1, y=37000, label=NsubDiv)
```



#Distribution of the marital status:

```
m<-as.data.frame(table(Dataset$marital))
ycol<-ggplot(m, aes(x=Var1,y=Freq,fill=Var1))
ycol + geom_bar(color = "black",fill = "light gray",stat="identity")
+geom_text(aes(label=Freq),vjust=0)
```



#Poutcome attribute distribution:

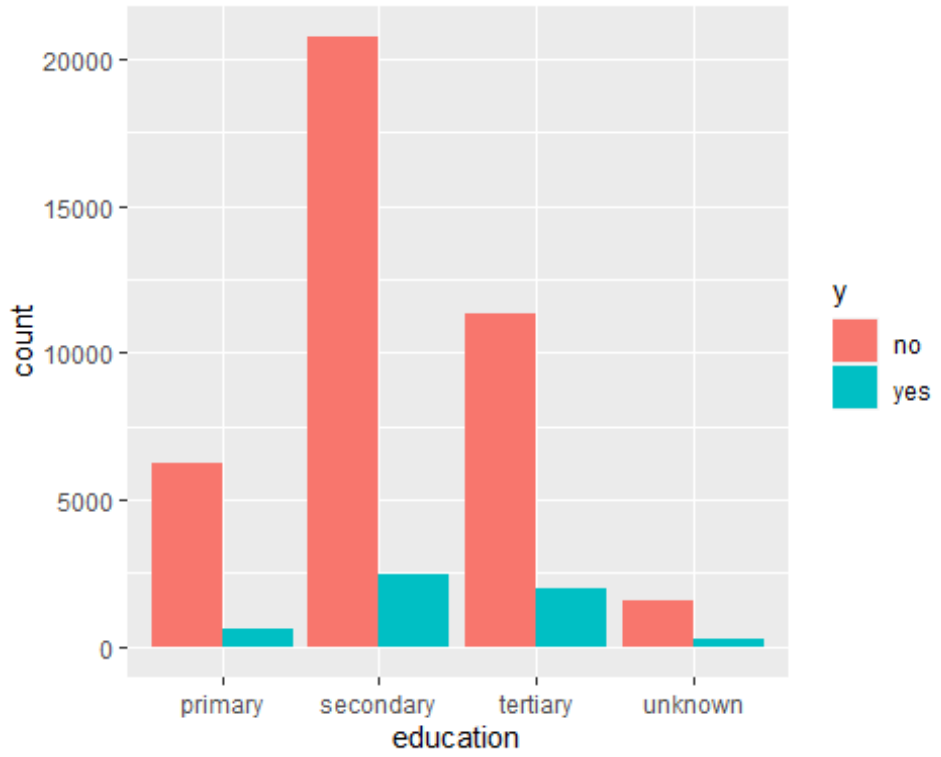
```
aggregate(Dataset$poutcome,list(Dataset$poutcome),FUN=length)
```

```
##   Group.1     x
## 1 failure  4901
## 2   other  1840
## 3 success  1511
## 4 unknown 36959
```

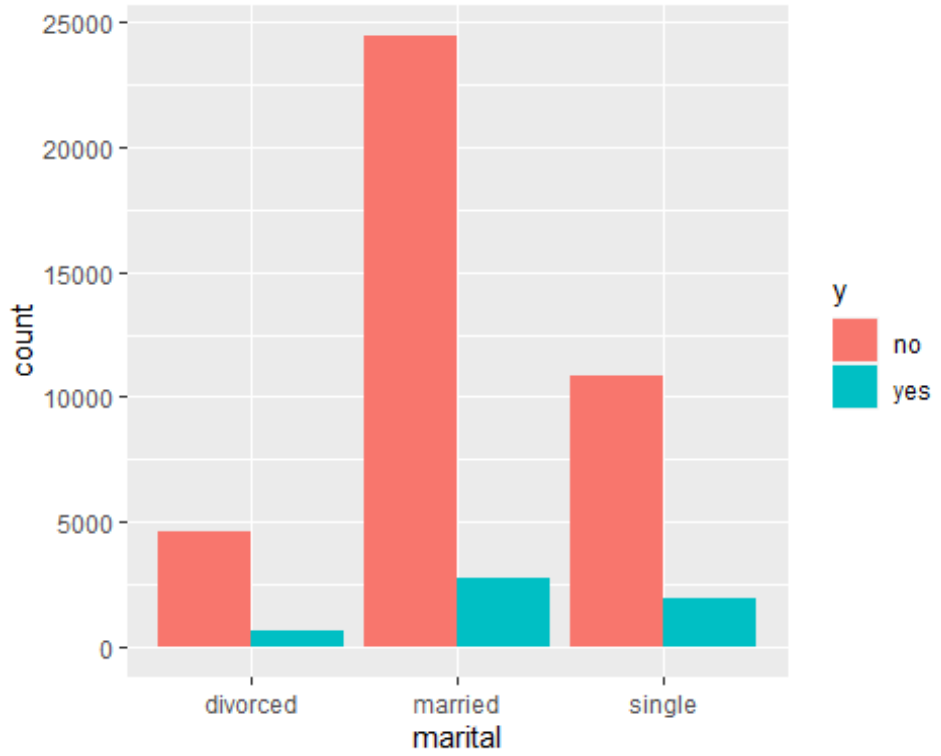
```
#the count of "unknown" value equals 36,959 (82%)
```

#Categorical distribution over the class y

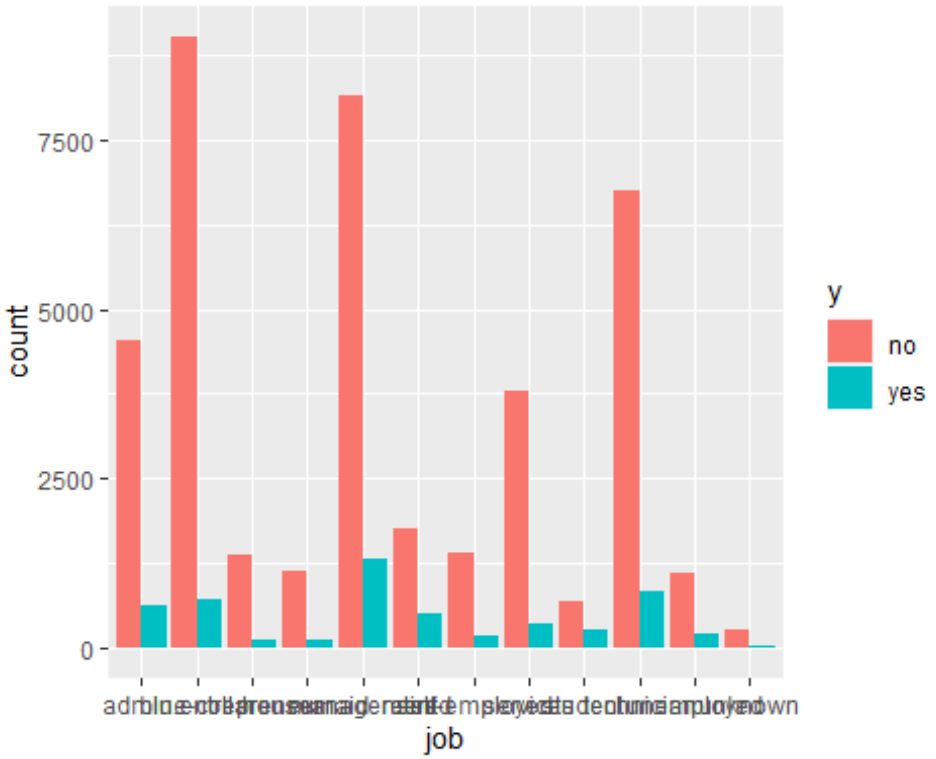
```
attach(Dataset)
ggplot(Dataset, aes(education, ..count..)) + geom_bar(aes(fill = y), position
= "dodge")
```



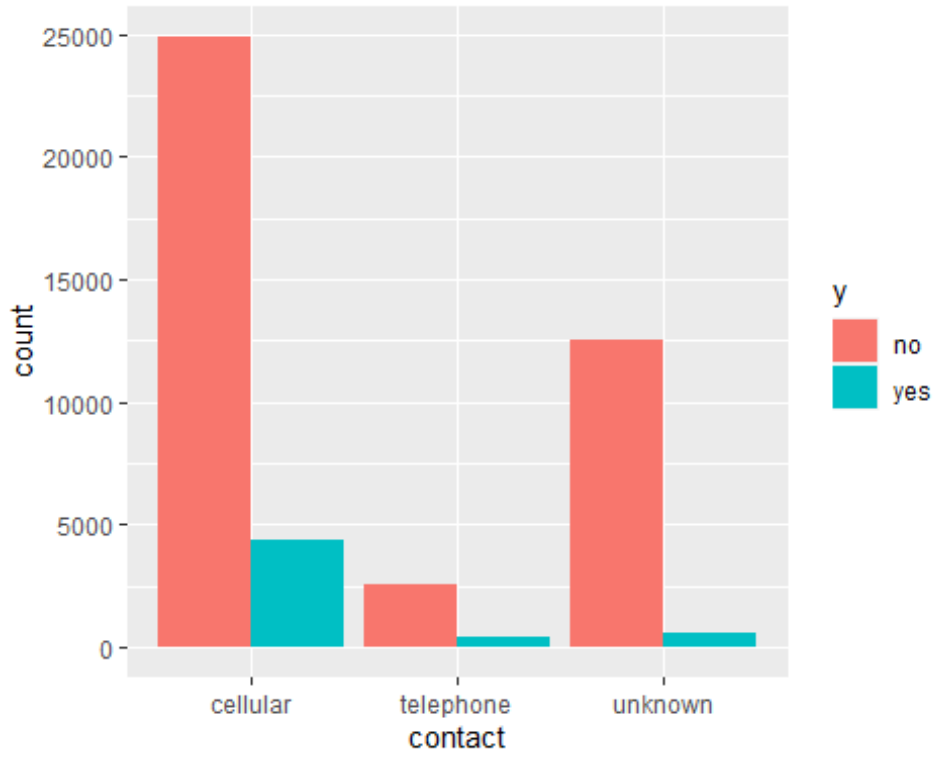

```
ggplot(Dataset, aes(marital, ..count..)) + geom_bar(aes(fill = y), position =
"dodge")
```



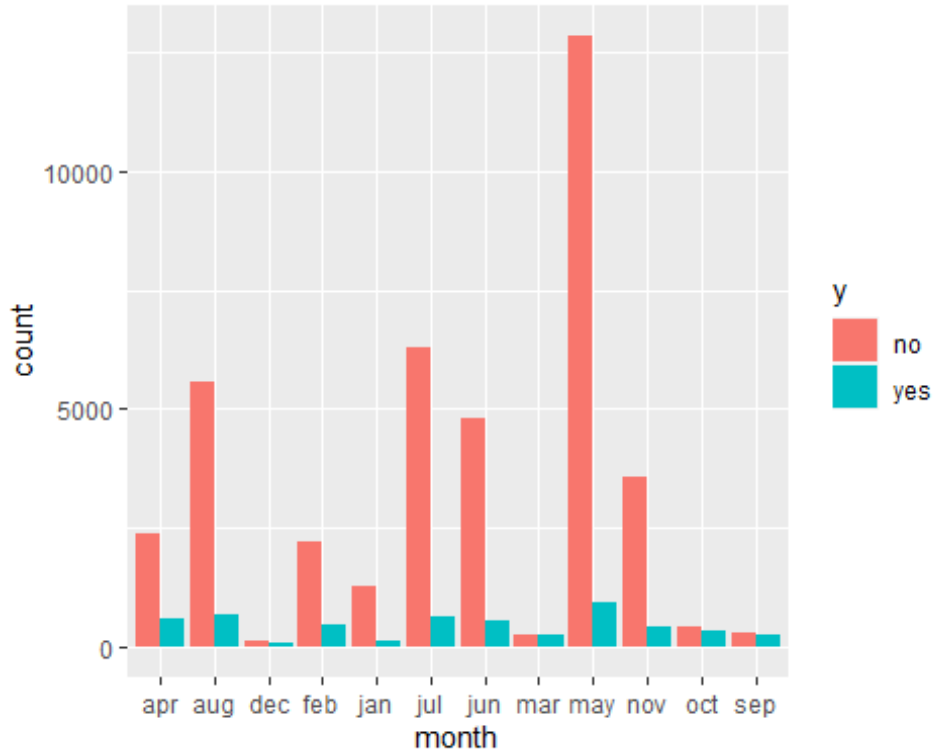
```
ggplot(Dataset, aes(job, ..count..)) + geom_bar(aes(fill = y), position =
"dodge")
```



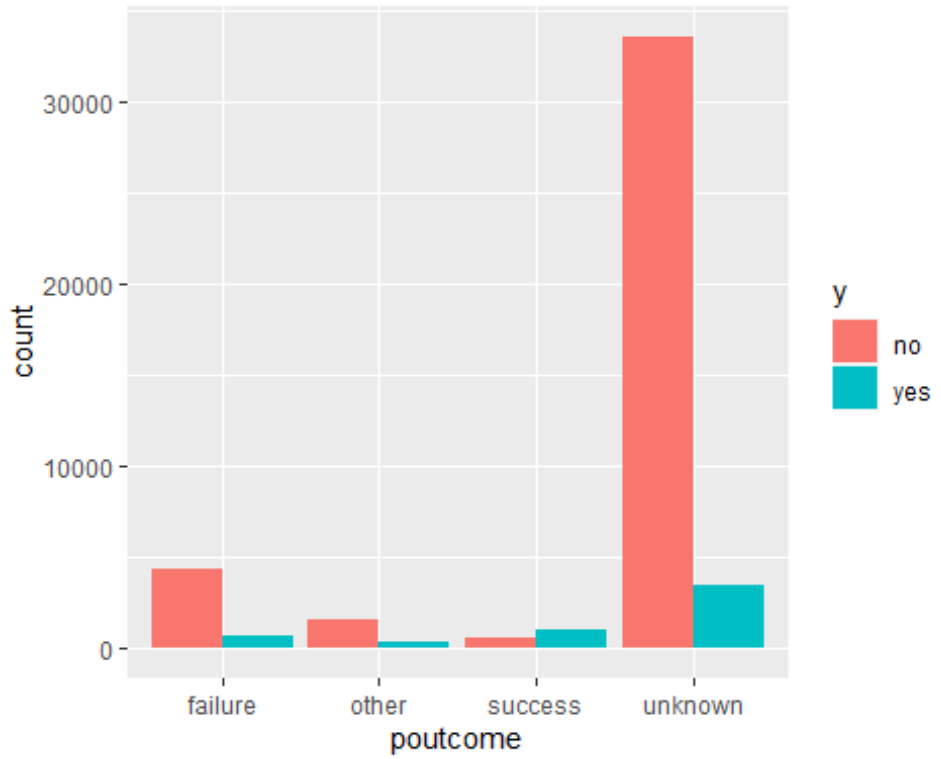

```
ggplot(Dataset, aes(contact, ..count..)) + geom_bar(aes(fill = y), position =
"dodge")
```




```
ggplot(Dataset, aes(month, ..count..)) + geom_bar(aes(fill = y), position =
"dodge")
```

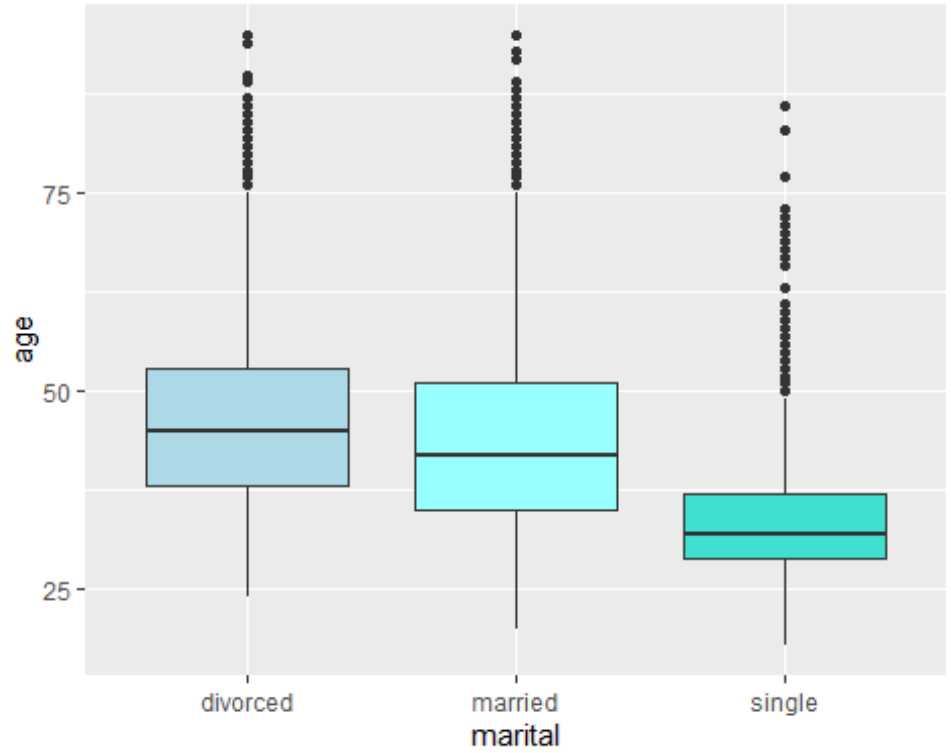


```
ggplot(Dataset, aes(poutcome, ..count..)) + geom_bar(aes(fill = y), position
= "dodge")
```

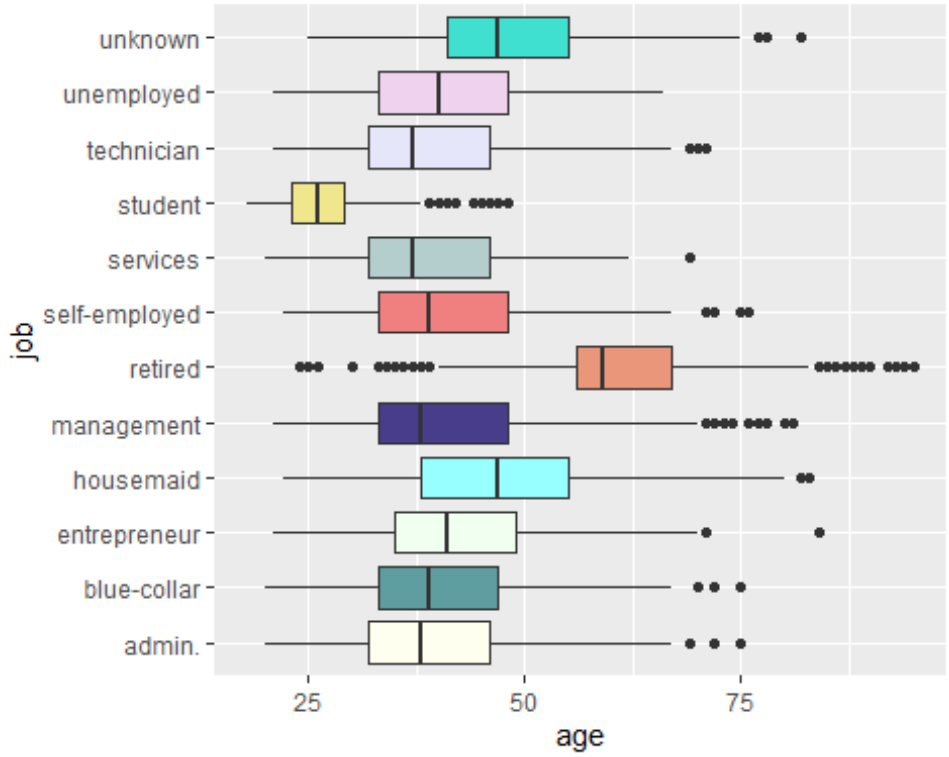


#EDA between some independent variables:

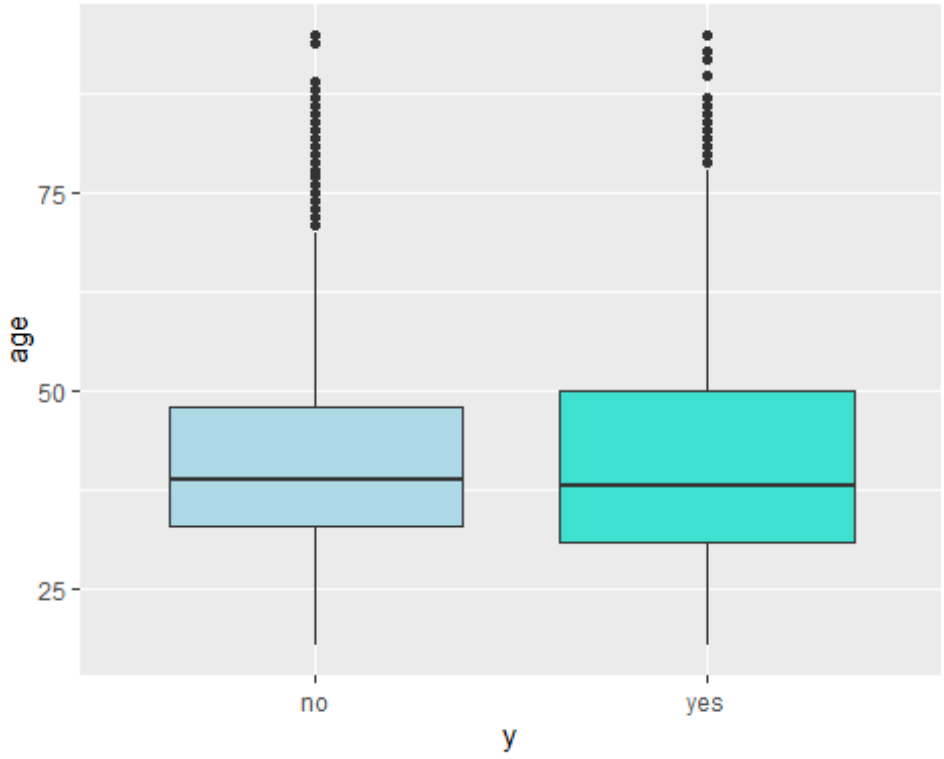
```
ggplot(Dataset, aes(x=marital, y=age)) + geom_boxplot(fill=c('light
blue','darkslategray1','turquoise'))
```




```
#The plot shows that the average age of unmarried clients is significantly
lower than that of the other clients.

ggplot(Dataset, aes(x=age, y=job),width =0.4) +
geom_boxplot(fill=c("ivory","cadetblue","honeydew","darkslategray1","darkslat
eblue","darksalmon","lightcoral","lightcyan3","khaki","lavender","thistle2","
turquoise"))
```




```
#the age of the most retired customers are between 60 and 27 for students

ggplot(Dataset, aes(x=y, y=age)) + geom_boxplot(fill=c('light
blue','turquoise'))
```


75
50
25
age
no
yes
y