MVA_Assign3

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

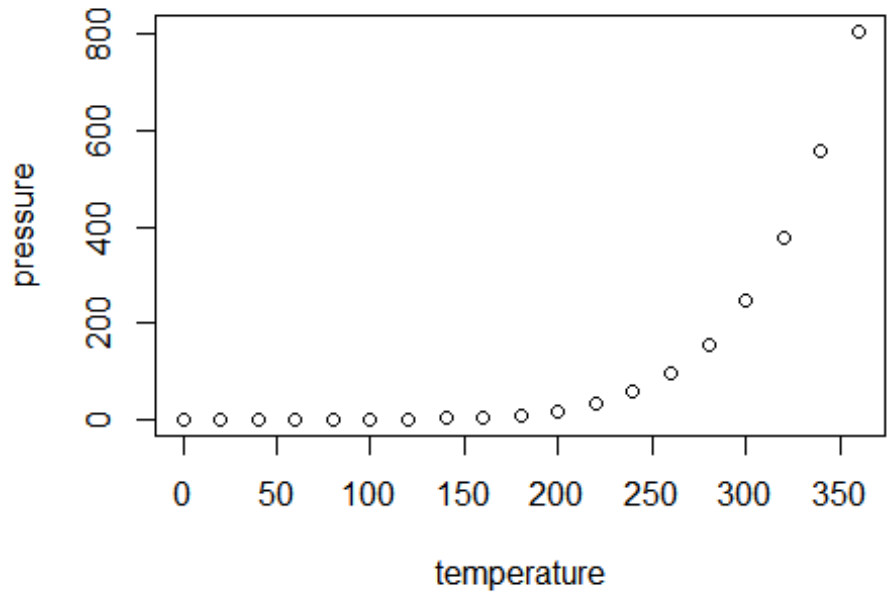
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library(data.table)
library(tidyverse)  # data manipulation

## -- Attaching packages ---------------------------------------
tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------
tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()

library(data.table) # fast file reading
library(gridExtra)  # arranging ggplot in grid

##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine

library(rmarkdown)
library(tinytex)
library(latexpdf)
library(latex2exp)
library(dplyr)

bank <- read.csv("C:/Users/Shamali/Desktop/Rutgers
Spring/multivariat/project/bank-marketing-dataset/bank.csv")
#Convert the data frame to data table
setDT(bank)
#Describe the columns and their data types
str(bank)

## Classes 'data.table' and 'data.frame':   11162 obs. of  17 variables:
##  $ age      : int  59 56 41 55 54 42 56 60 37 28 ...
##  $ job      : Factor w/ 11 levels "admin.","blue-collar",..: 1 1 10 8 1 5
5 6 10 8 ...
##  $ marital  : Factor w/ 3 levels "divorced","married",..: 2 2 2 2 2 3 2 1
2 3 ...
##  $ education: Factor w/ 3 levels "primary","secondary",..: 2 2 2 2 3 3 3 2
2 2 ...
##  $ default  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ balance  : int  2343 45 1270 2476 184 0 830 545 1 5090 ...
##  $ housing  : int  1 0 1 1 0 1 1 1 1 1 ...
##  $ loan     : int  0 0 0 0 0 1 1 0 0 0 ...
##  $ contact  : Factor w/ 2 levels "cellular","telephone": NA NA NA NA NA NA
NA NA NA NA ...
##  $ day      : int  5 5 5 5 5 5 6 6 6 6 ...
##  $ month    : Factor w/ 12 levels "0v","apr","aug",..: 10 10 10 10 10 10
10 10 10 10 ...
##  $ duration : int  1042 1467 1389 579 673 562 1201 1030 608 1297 ...
##  $ campaign : int  1 1 1 1 2 2 1 1 1 3 ...
##  $ 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 : Factor w/ 3 levels "failure","other",..: NA NA NA NA NA NA
NA NA NA NA ...
##  $ deposit  : int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>

#By head we get to know first n rows to get grasp of the data
head(bank)

##    age        job marital education default balance housing loan contact
day
## 1:  59     admin. married secondary       0    2343       1    0    <NA>
5
## 2:  56     admin. married secondary       0      45       0    0    <NA>
```

```
5
## 3:  41 technician married secondary       0    1270       1    0    <NA>
5
## 4:  55   services married secondary       0    2476       1    0    <NA>
5
## 5:  54     admin. married  tertiary       0     184       0    0    <NA>
5
## 6:  42 management  single  tertiary       0       0       1    1    <NA>
5
##    month duration campaign pdays previous poutcome deposit
## 1:   may     1042        1    -1        0     <NA>       1
## 2:   may     1467        1    -1        0     <NA>       1
## 3:   may     1389        1    -1        0     <NA>       1
## 4:   may      579        1    -1        0     <NA>       1
## 5:   may      673        2    -1        0     <NA>       1
## 6:   may      562        2    -1        0     <NA>       1
```

```
#Find NA In the data table.

table(is.na(bank))

##
##  FALSE   TRUE
## 178515  11239

#Find NA in Columns.
bank[is.na(age),NROW(age)]

## [1] 0

bank[is.na(job),NROW(job)]

## [1] 70

bank[is.na(education),NROW(education)]

## [1] 497

grep('NA',bank)

## [1]  2  4  9 16

#Find different elements in the column
unique(bank$job)

##  [1] admin.        technician    services      management    retired
##  [6] blue-collar   unemployed    entrepreneur  housemaid     <NA>
## [11] self-employed student
## 11 Levels: admin. blue-collar entrepreneur housemaid management ...
unemployed

unique(bank$marital)
```

```
## [1] married  single   divorced
## Levels: divorced married single

#Summary of dataset

summary(bank)

##       age                 job           marital         education
##  Min.   :18.00   management :2566   divorced:1293   primary  :1500
##  1st Qu.:32.00   blue-collar:1944   married :6351   secondary:5476
##  Median :39.00   technician :1823   single  :3518   tertiary :3689
##  Mean   :41.23   admin.     :1334                   NA's     : 497
##  3rd Qu.:49.00   services   : 923
##  Max.   :95.00   (Other)    :2502
##                  NA's       :  70
##     default            balance         housing            loan
##  Min.   :0.00000   Min.   :-6847   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:  122   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :  550   Median :0.0000   Median :0.0000
##  Mean   :0.01505   Mean   : 1529   Mean   :0.4731   Mean   :0.1308
##  3rd Qu.:0.00000   3rd Qu.: 1708   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :81204   Max.   :1.0000   Max.   :1.0000
##
##       contact           day             month         duration
##  cellular :8042   Min.   : 1.00   may    :2824   Min.   :   2
##  telephone: 774   1st Qu.: 8.00   aug    :1519   1st Qu.: 138
##  NA's     :2346   Median :15.00   jul    :1514   Median : 255
##                   Mean   :15.66   jun    :1222   Mean   : 372
##                   3rd Qu.:22.00   0v     : 943   3rd Qu.: 496
##                   Max.   :31.00   apr    : 923   Max.   :3881
##                                   (Other):2217
##     campaign          pdays            previous          poutcome
##  Min.   : 1.000   Min.   : -1.00   Min.   : 0.0000   failure:1228
##  1st Qu.: 1.000   1st Qu.: -1.00   1st Qu.: 0.0000   other  : 537
##  Median : 2.000   Median : -1.00   Median : 0.0000   success:1071
##  Mean   : 2.508   Mean   : 51.33   Mean   : 0.8326   NA's   :8326
##  3rd Qu.: 3.000   3rd Qu.: 20.75   3rd Qu.: 1.0000
##  Max.   :63.000   Max.   :854.00   Max.   :58.0000
##
##     deposit
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4738
##  3rd Qu.:1.0000
##  Max.   :1.0000
##

#bank1=bank.head(1000)
#bank1=tail(bank)
```

```r
#pairs(bank)

#Take sample of 1000 from the dataset.
bank12=bank[sample(.N,1000)]

#check for duplicate rows
sum(duplicated(bank))
```

```
## [1] 0
```

```r
#check for rows which contain missing data
sum(!complete.cases(bank))
```

```
## [1] 8487
```

```r
#Check for rows which have complete missing values in all columns
all.empty = rowSums(is.na(bank))==ncol(bank)
sum(all.empty)
```

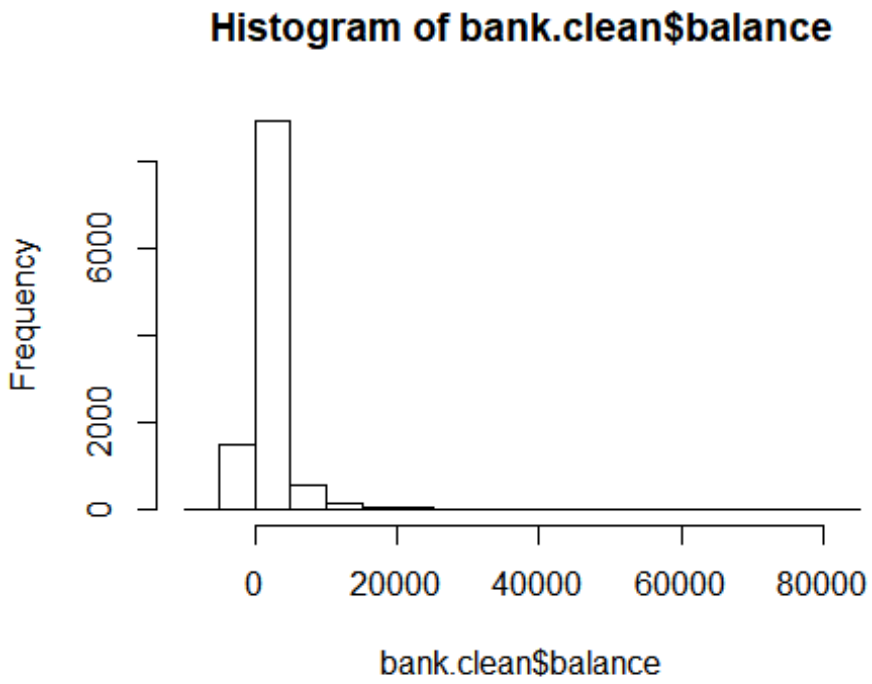
```
## [1] 0
```

```r
#check for missing value by variable
sapply(bank, function(x) sum(is.na(x)))
```

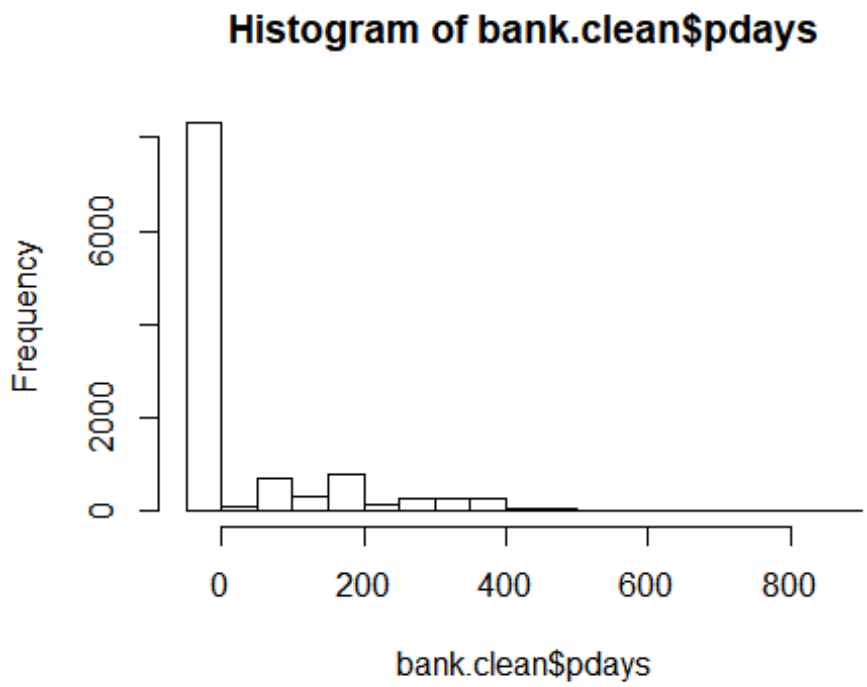
```
##       age       job   marital education   default   balance   housing
loan
##         0        70         0       497         0         0         0
0
##   contact       day     month  duration  campaign     pdays  previous
poutcome
##      2346         0         0         0         0         0         0
8326
##   deposit
##         0
```

```r
#Remove rows with all columns missing value
bank.clean = bank[!all.empty,]

#Create New Column To Indicate Missing Detection
bank.clean$missing = !complete.cases(bank.clean)

#Missing Numeric Value Treatment

#Replace with Average
bank.clean$age[is.na(bank.clean$age)] = mean(bank$age, na.rm=T)
bank.clean$day[is.na(bank.clean$day)] = mean(bank$day, na.rm=T)
bank.clean$duration[is.na(bank.clean$duration)] = mean(bank$duration,
na.rm=T)
bank.clean$previous[is.na(bank.clean$previous)] = mean(bank$previous,
na.rm=T)
bank.clean$campaign[is.na(bank.clean$campaign)] = mean(bank$campaign,
```

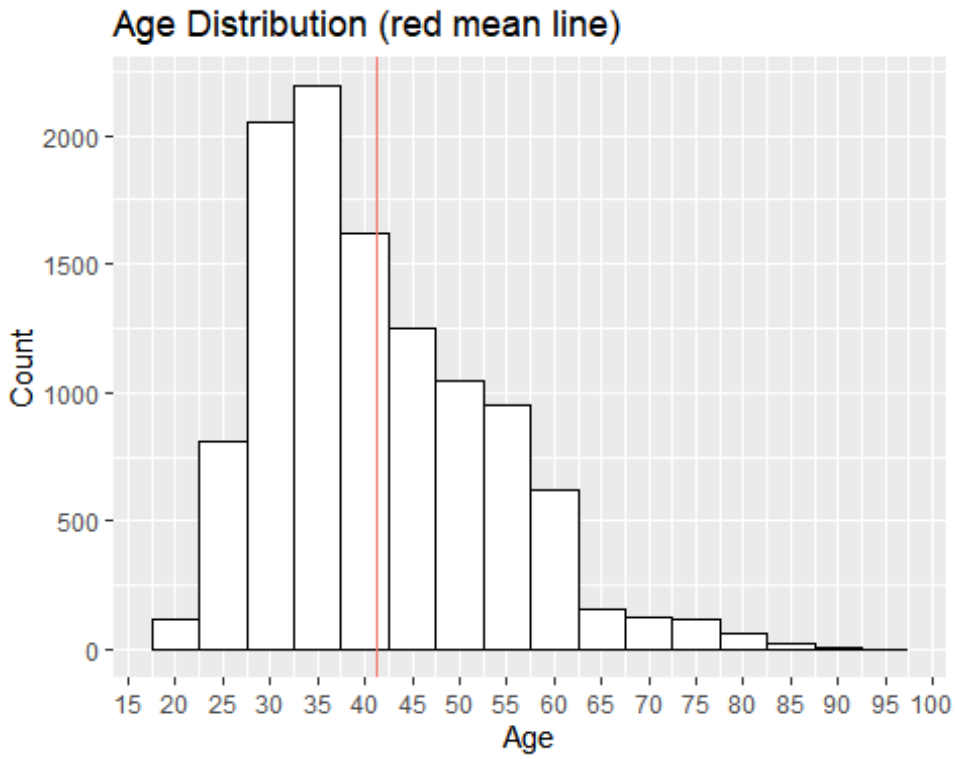
```
na.rm=T)

hist(bank.clean$balance)
```



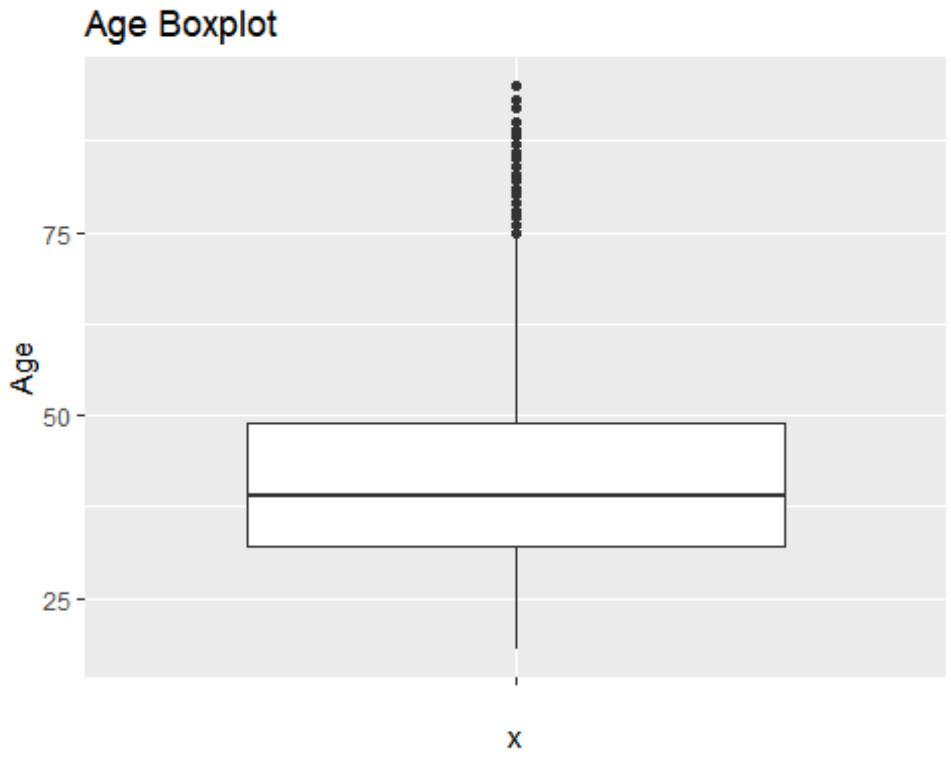

```
hist(bank.clean$pdays)
```



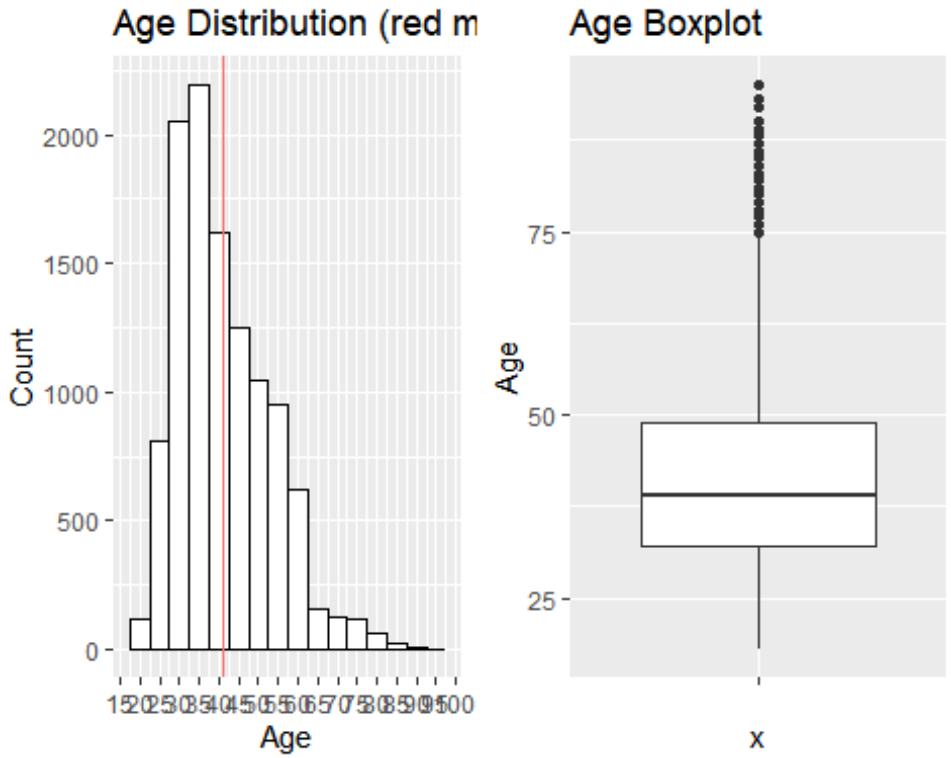

```
bank.clean$pdays[is.na(bank.clean$pdays)] = as.numeric(names(sort(-
table(bank$pdays)))[1])
bank.clean$balance[is.na(bank.clean$balance)] = as.numeric(names(sort(-
table(bank$balance)))[1])

#EDA
summary(bank$age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   32.00   39.00   41.23   49.00   95.00

library(ggplot2)
gg = ggplot (bank)
graph1 = gg + geom_histogram(aes(x=age),color="black", fill="white", binwidth
= 5) +
  ggtitle('Age Distribution (red mean line)') +
  ylab('Count') +
  xlab('Age') +
  geom_vline(aes(xintercept = mean(age), color = "red")) +
  scale_x_continuous(breaks = seq(0,100,5)) +
  theme(legend.position = "none")
graph1
```



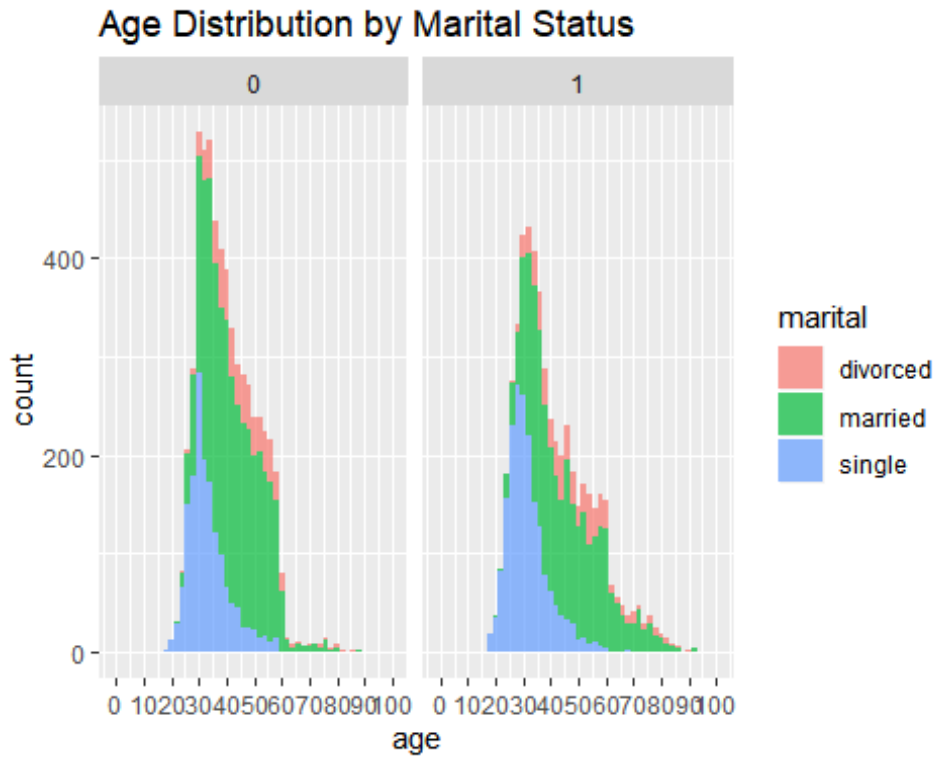

```
graph2 = gg + geom_boxplot(aes(x='', y=age)) +
  ggtitle('Age Boxplot') +
  ylab('Age')
graph2
```



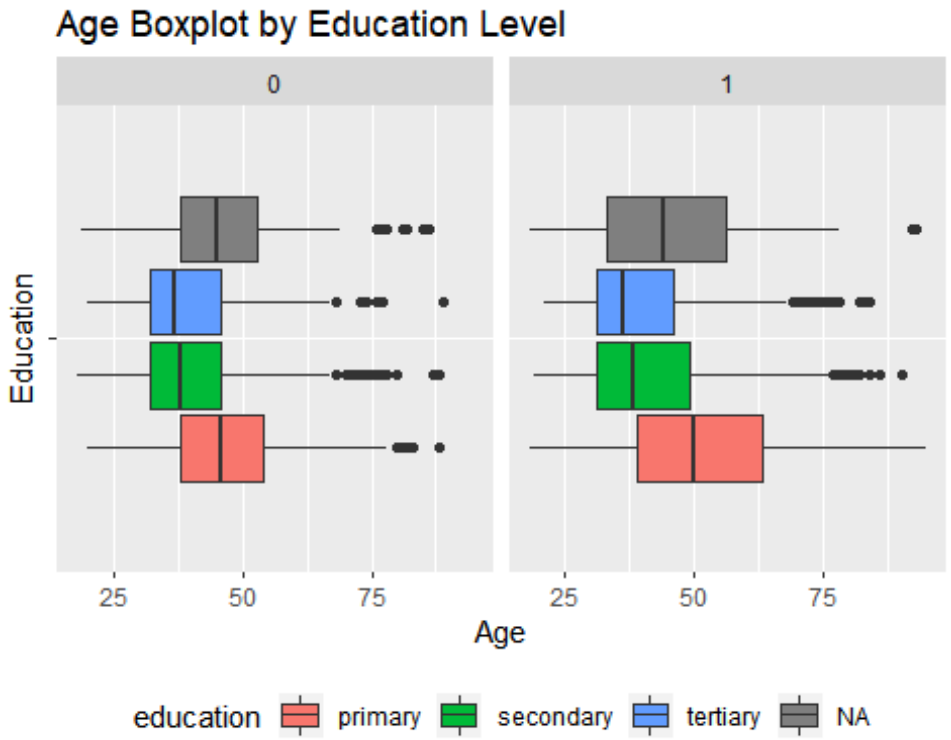

```
library(gridExtra)
grid.arrange(graph1, graph2, ncol = 2)
```



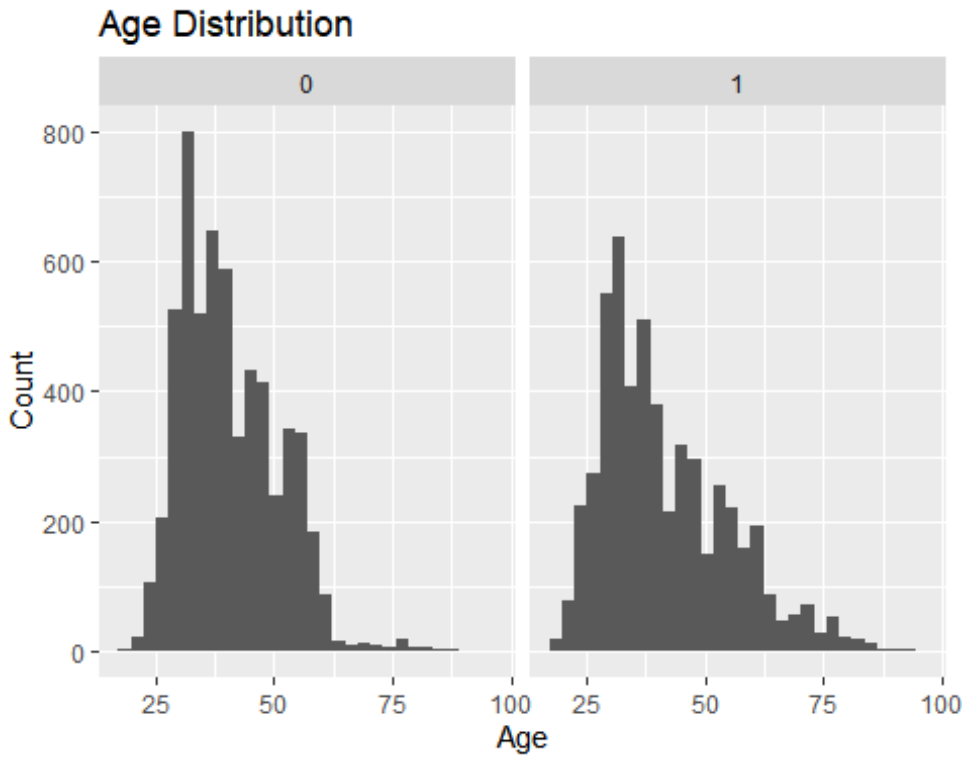
```r
#Age Distribution vs Marital Status That Subscribes Term Deposit

graph3 <- ggplot(bank, aes(x=age, fill=marital)) +
  geom_histogram(binwidth = 2, alpha=0.7) +
  facet_grid(cols = vars(deposit)) +
  expand_limits(x=c(0,100)) +
  scale_x_continuous(breaks = seq(0,100,10)) +
  ggtitle("Age Distribution by Marital Status")

graph3
```



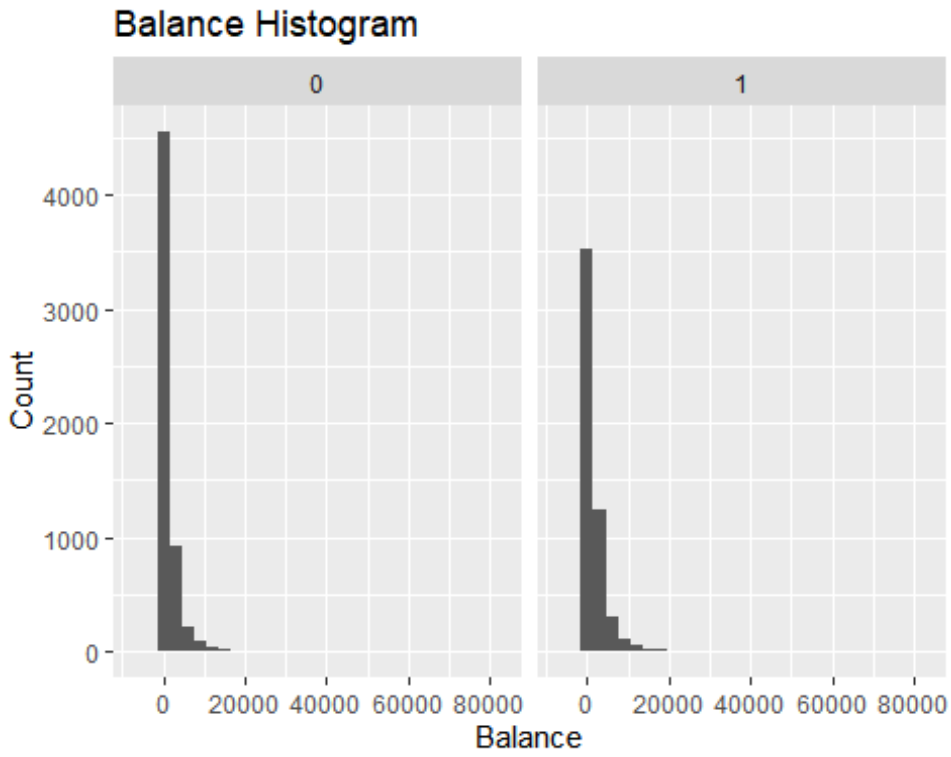

```r
#Age Boxplot vs Education Level That Subscribes Term Deposit

graph4 <- ggplot(bank, aes(x='', y=age, fill=education)) +
  geom_boxplot() +
  facet_grid(cols = vars(deposit)) +
  coord_flip() +
  ggtitle("Age Boxplot by Education Level") +
  ylab("Age") +
  xlab("Education") +
  theme(legend.position = "bottom")
graph4
```



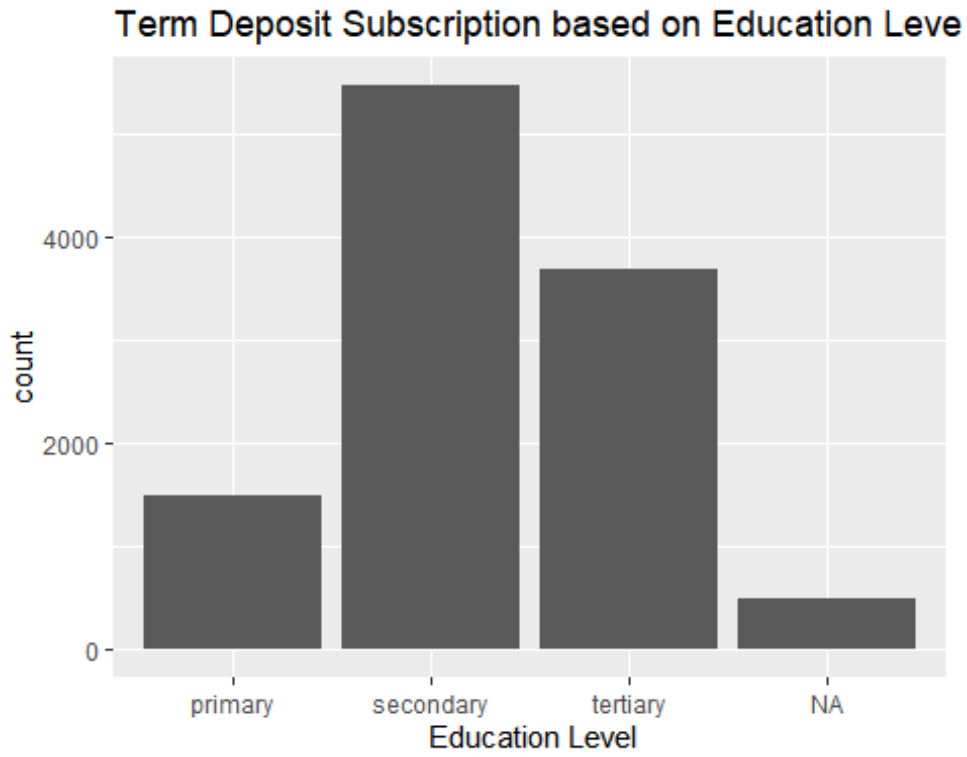

```
#Subscription Analysis

#Age vs Subscription
ggplot (bank, aes(x=age)) + geom_histogram() +
  facet_grid(cols=vars(deposit)) +
  ggtitle('Age Distribution') + ylab('Count') + xlab('Age')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



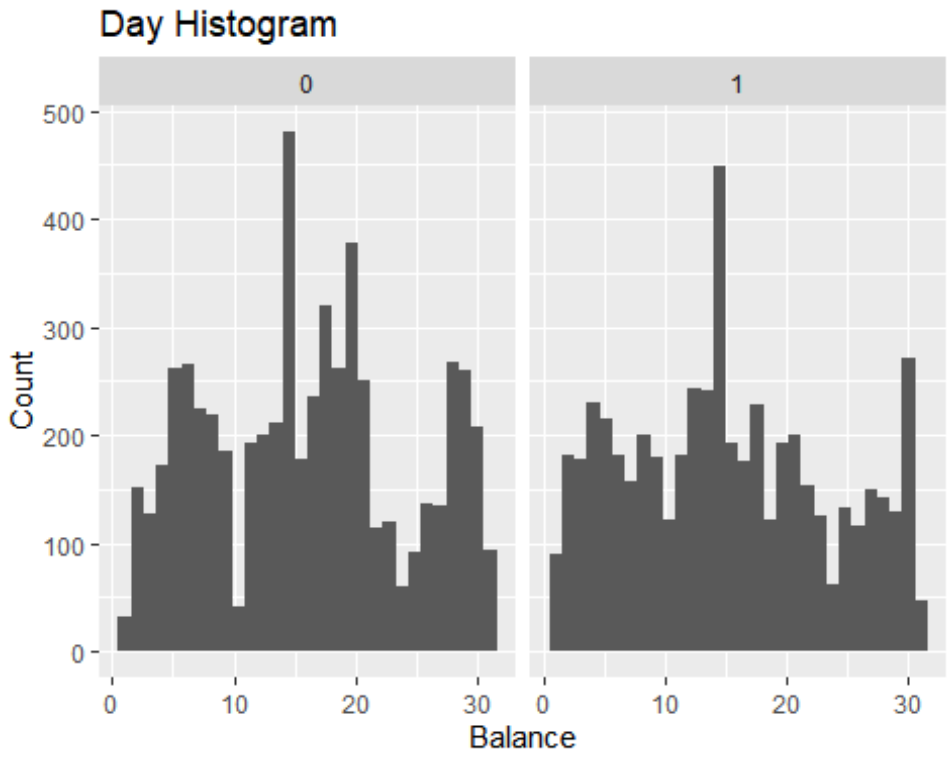

```
#Balance vs Subscription
ggplot (bank, aes(x=balance)) + geom_histogram() +
  facet_grid(cols=vars(deposit)) +
  ggtitle('Balance Histogram') + ylab('Count') + xlab('Balance')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



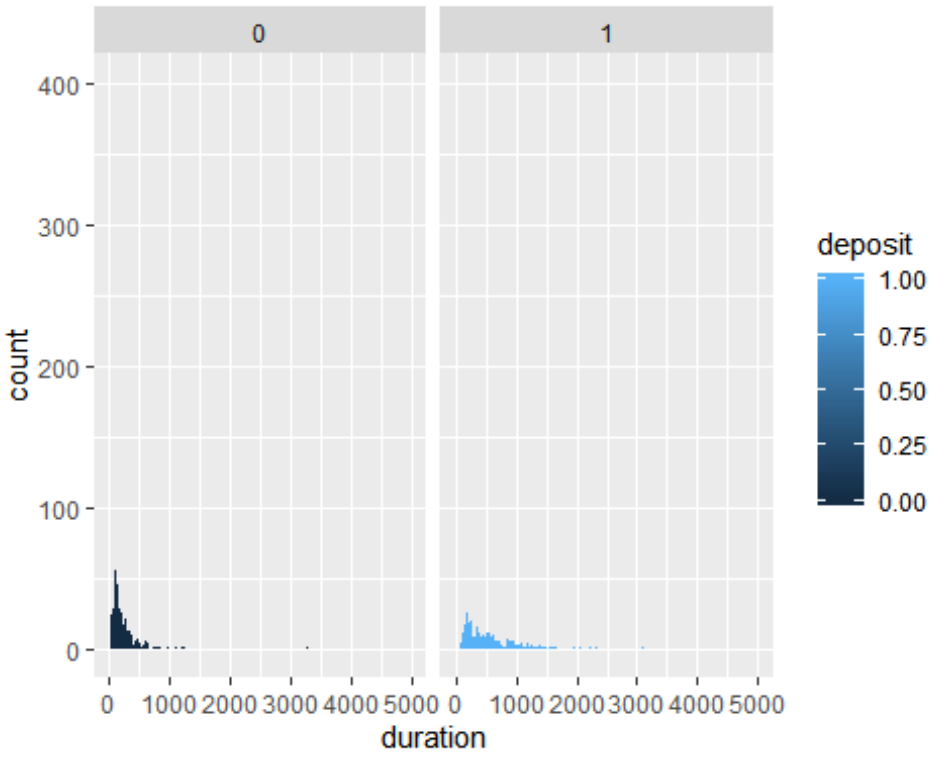

```r
#Education vs Subscription
ggplot(data = bank, aes(x=education, fill=deposit)) +
  geom_bar() +
  ggtitle("Term Deposit Subscription based on Education Level") +
  xlab(" Education Level") +
  guides(fill=guide_legend(title="Subscription of Term Deposit"))
```



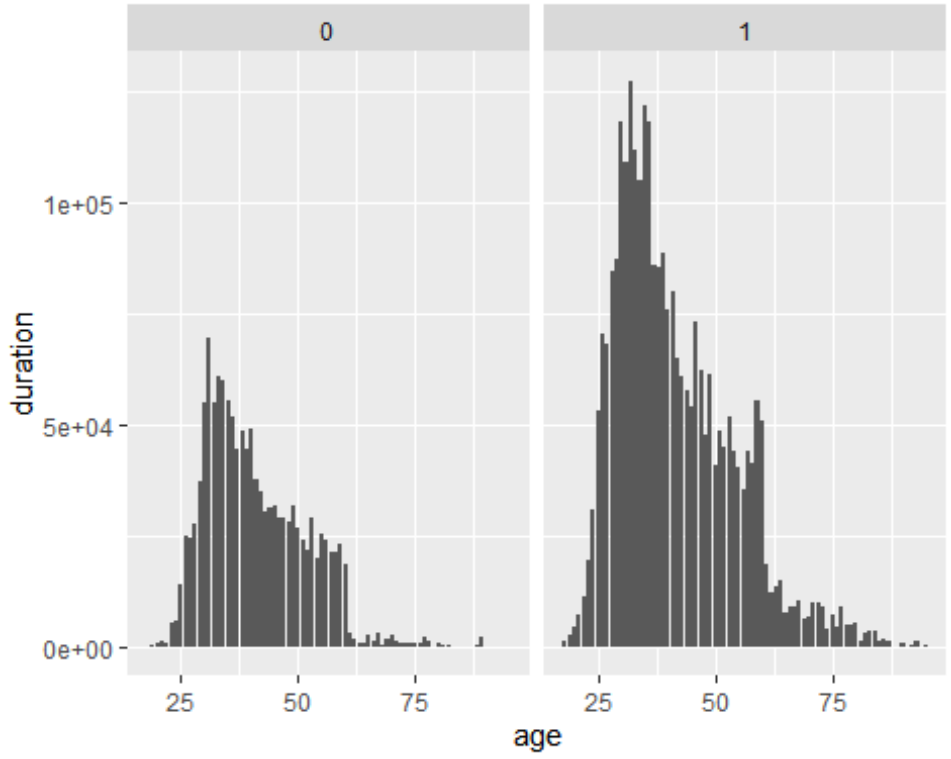

```
#Day vs Subscription

ggplot (bank, aes(x=day)) + geom_histogram() +
  facet_grid(cols=vars(deposit)) +
  ggtitle('Day Histogram') + ylab('Count') + xlab('Balance')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



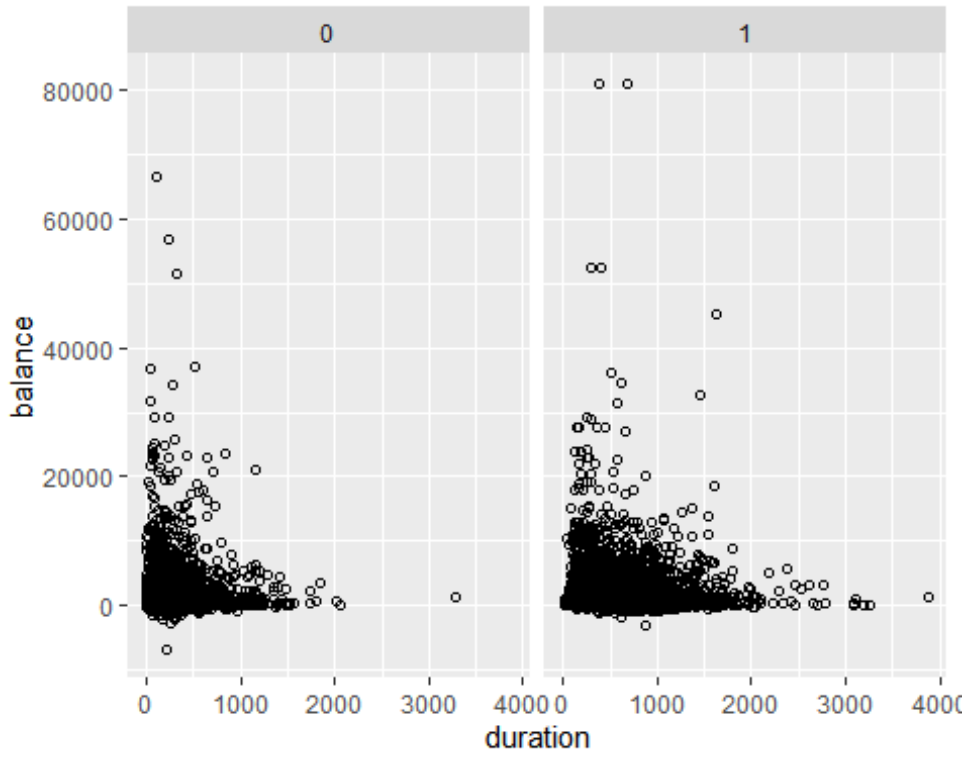

```
#duration


graph5 <- ggplot(bank, aes(x=duration, fill = deposit)) +
  geom_histogram(binwidth = 2) +
  facet_grid(cols = vars(deposit)) +
  coord_cartesian(xlim = c(0,5000), ylim = c(0,400))
graph5
```



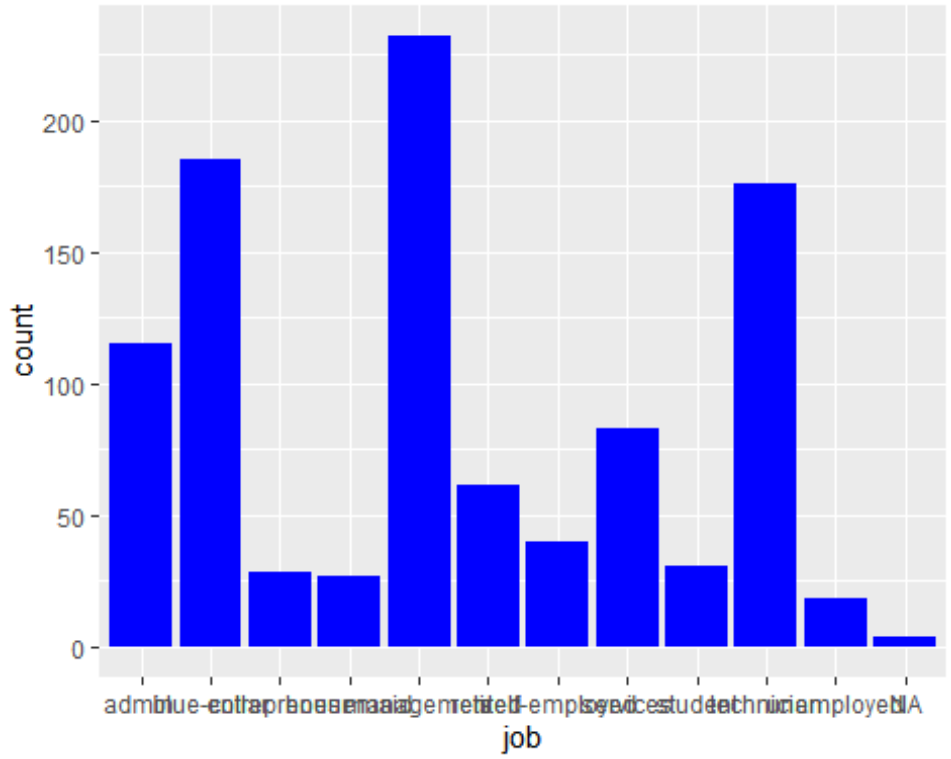

```r
#Barplot of Duration by Age
ggplot(bank, aes(age, duration)) +
  geom_col() +
  facet_grid(cols = vars(deposit))
```



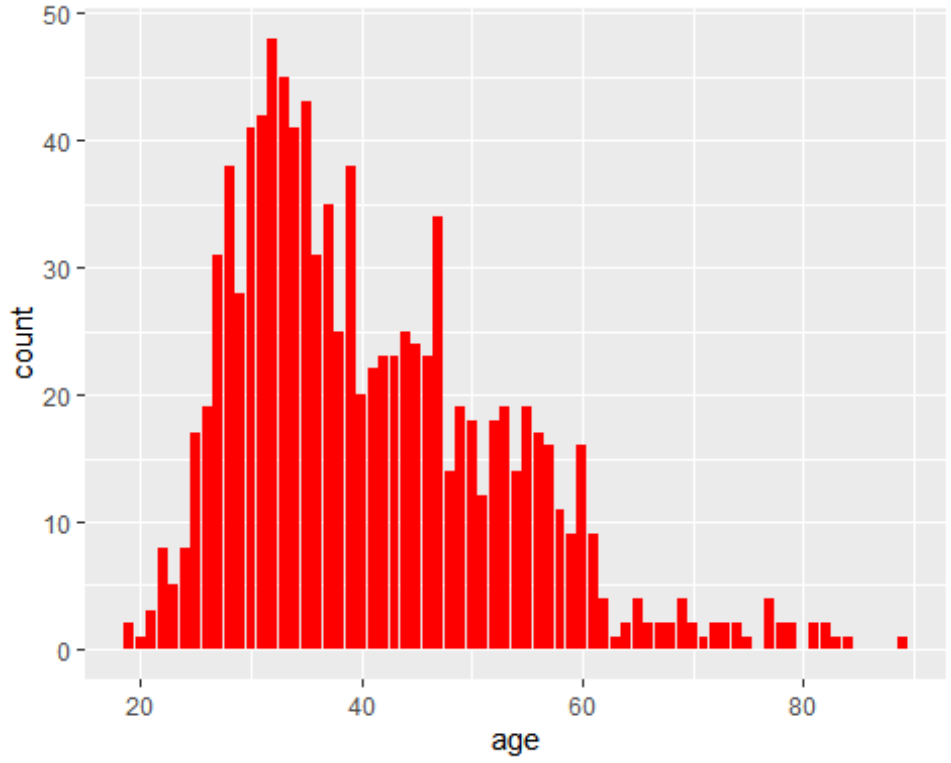

```
#Scatterplot of Duration s Balance

ggplot(bank, aes(x=duration, y=balance)) +
  facet_grid(cols = vars(deposit)) +
  geom_point(shape=1)
```



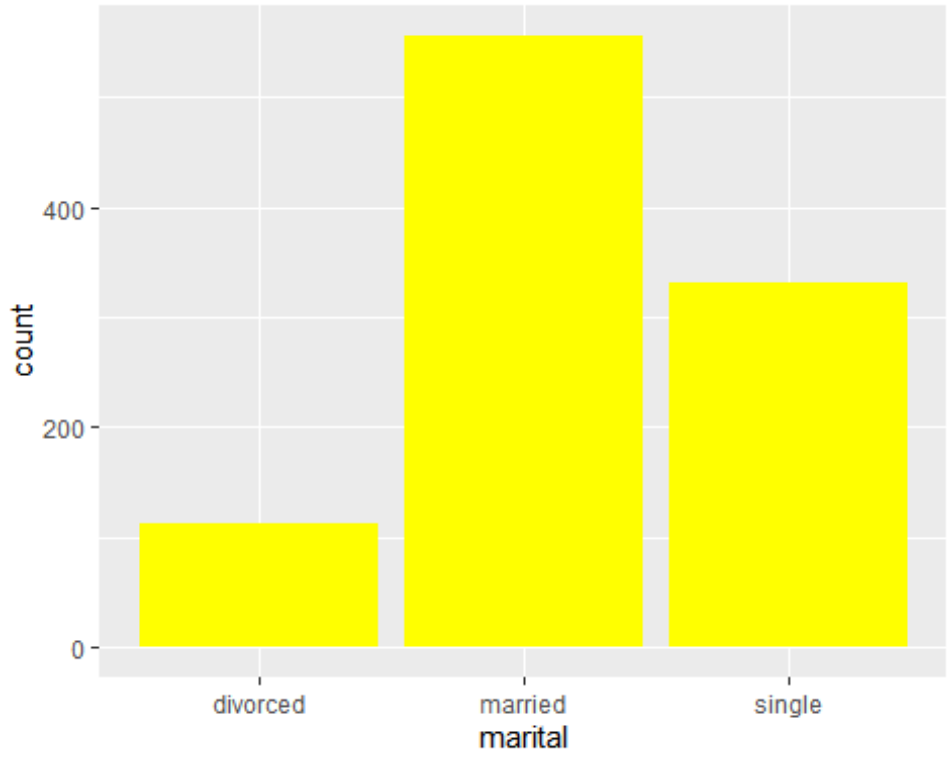

```
ggplot (bank12 ,aes (x=job)) + geom_bar(fill='blue')
```



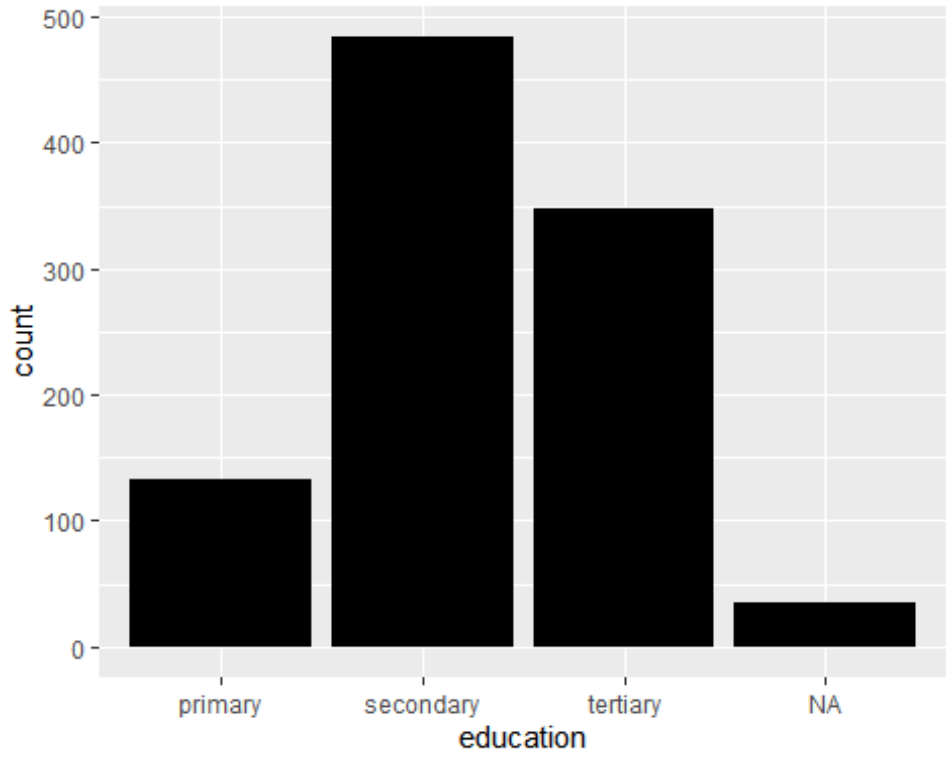

```
ggplot (bank12 ,aes (x=age)) + geom_bar(fill = 'red')
```



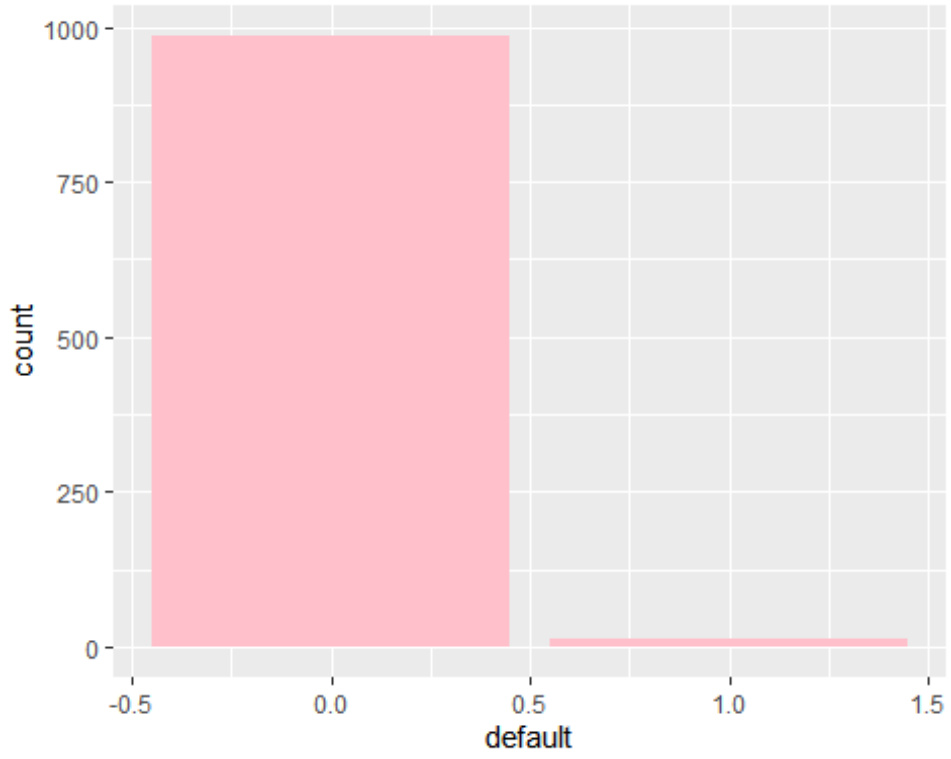

```
ggplot (bank12 ,aes (x=marital)) + geom_bar(fill = 'yellow')
```



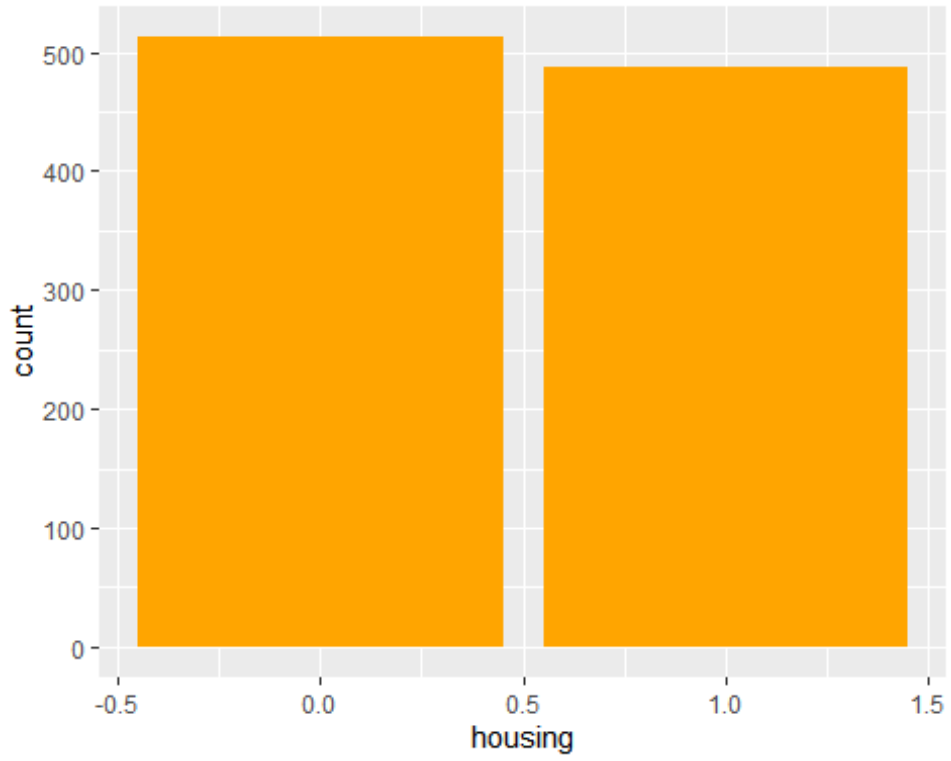

```
ggplot (bank12 ,aes (x=education)) + geom_bar(fill = 'black')
```



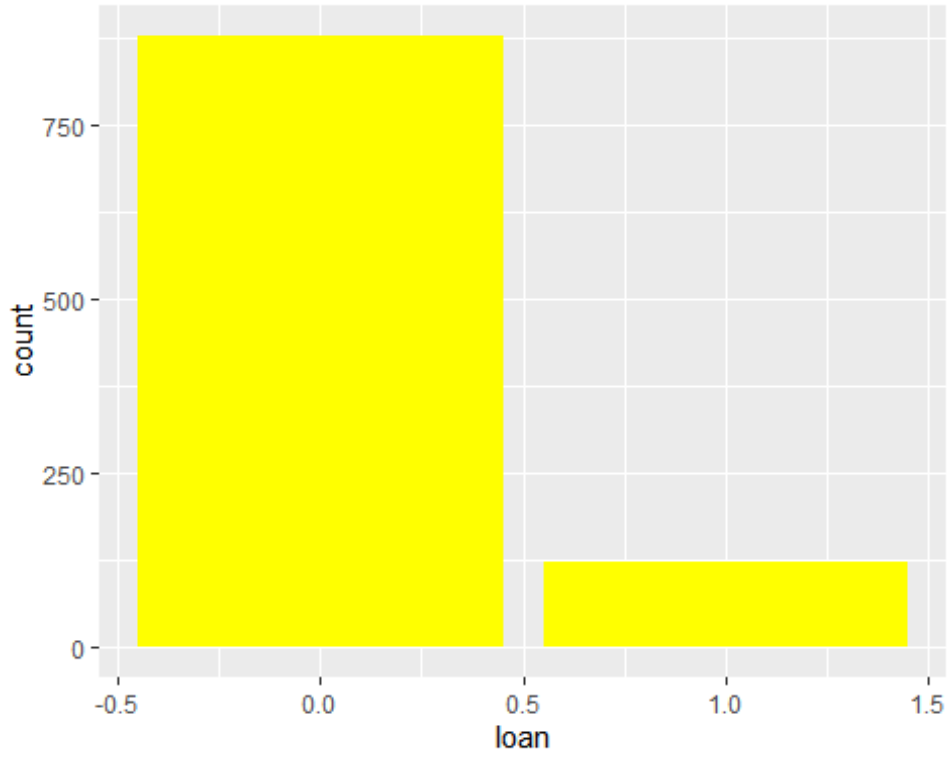

```
ggplot (bank12 ,aes (x=default)) + geom_bar(fill = 'pink')
```



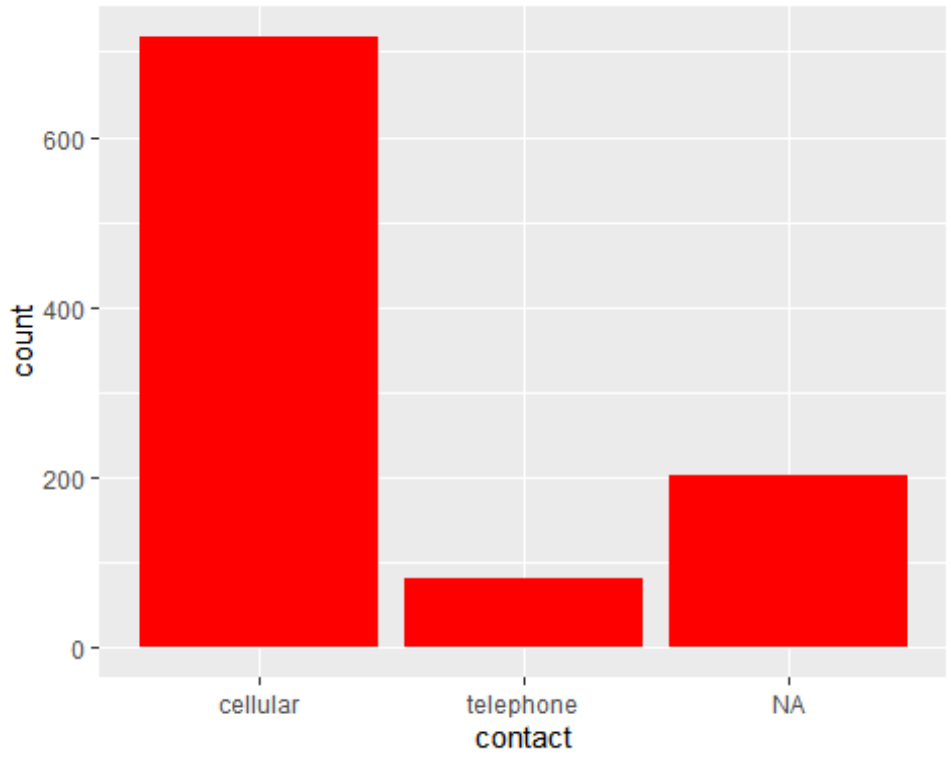

```
ggplot (bank12 ,aes (x=housing)) + geom_bar(fill = 'orange')
```



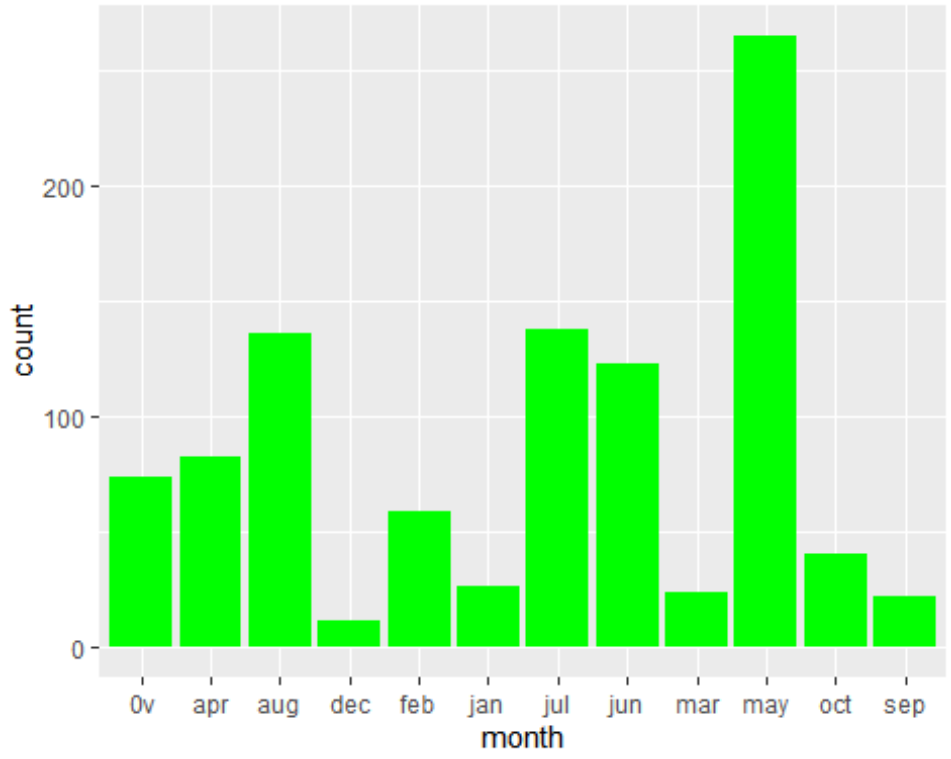

```
ggplot (bank12 ,aes (x=loan)) + geom_bar(fill = 'yellow')
```



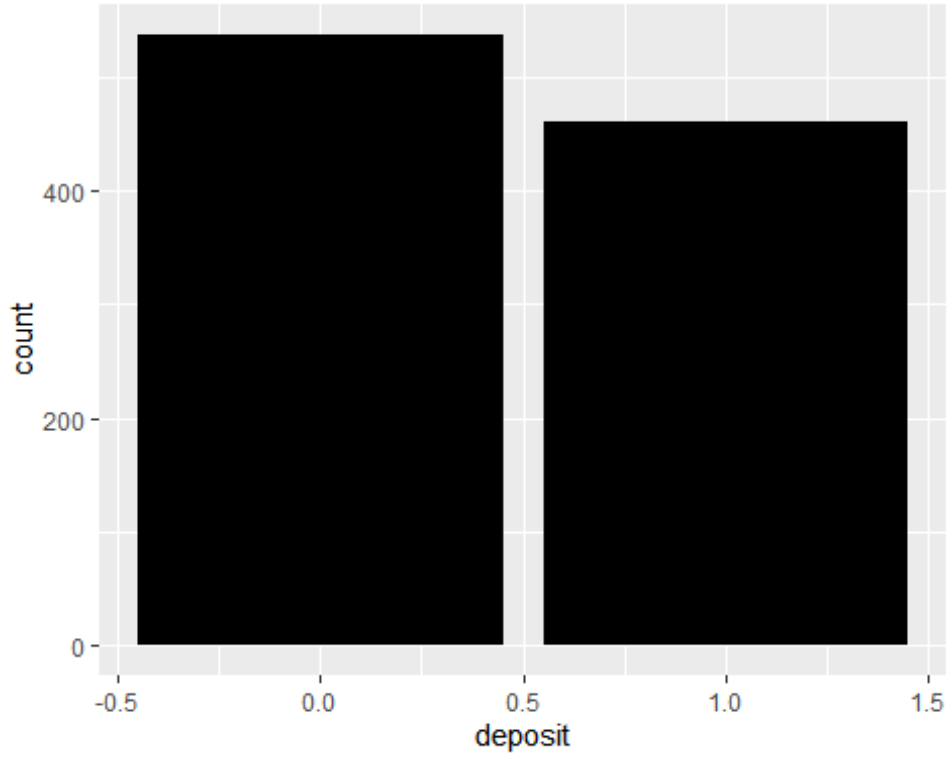

```
ggplot (bank12 ,aes (x=contact)) + geom_bar(fill = 'red')
```




```
ggplot (bank12 ,aes (x=month)) + geom_bar(fill = 'green')
```

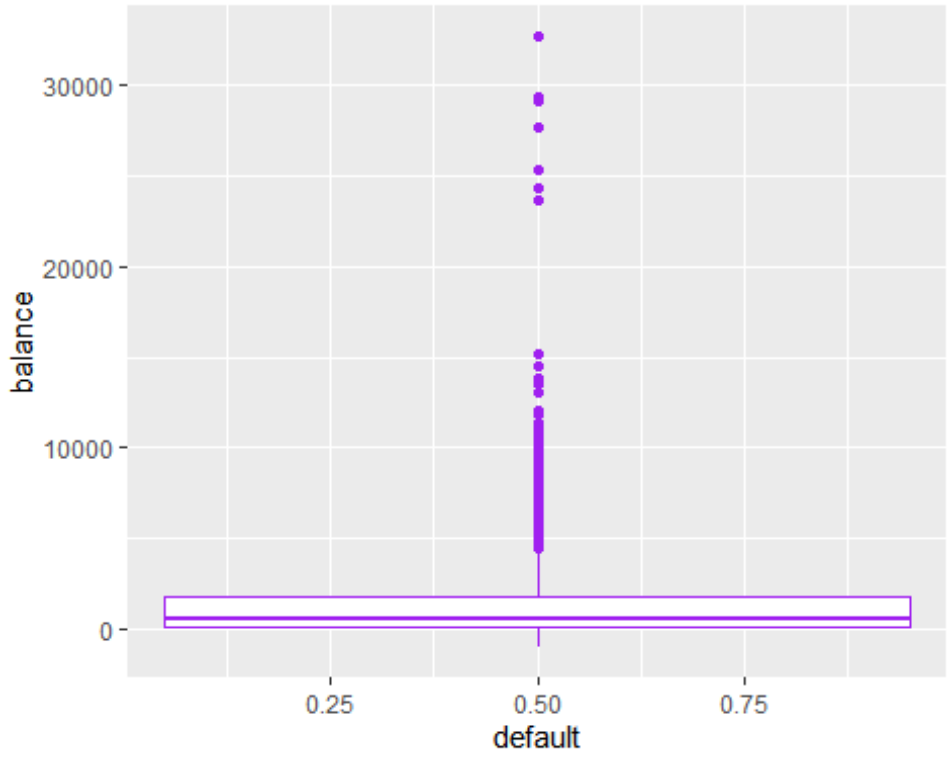



```
ggplot (bank12 ,aes (x=deposit)) + geom_bar(fill = 'black')
```

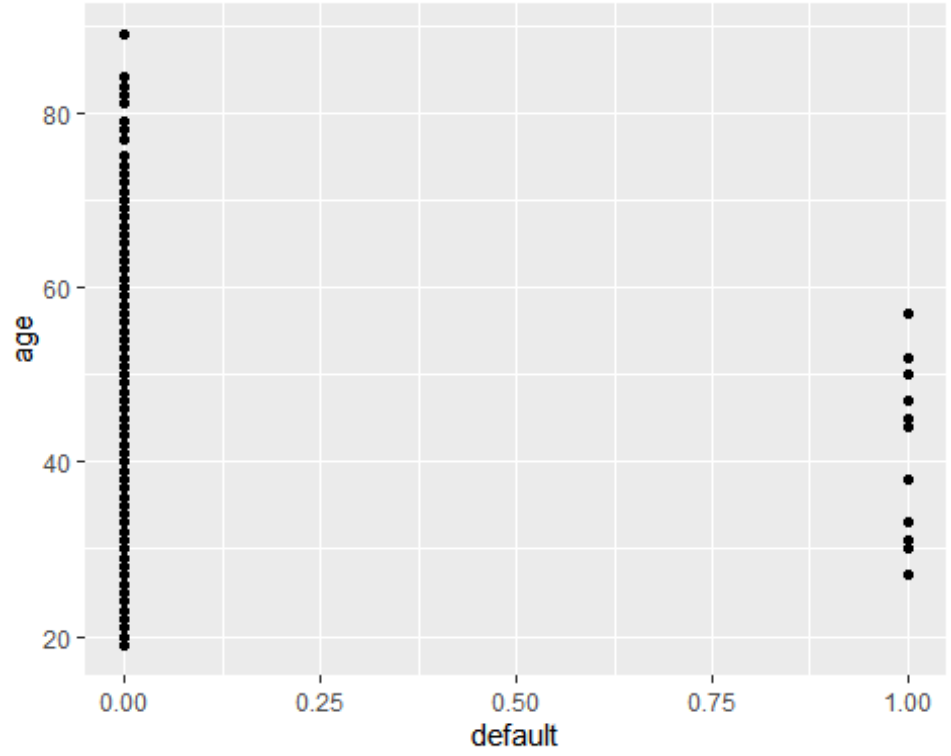



```
ggplot(bank12 ,aes(x=default ,y= balance))+geom_boxplot( color ='purple')
```

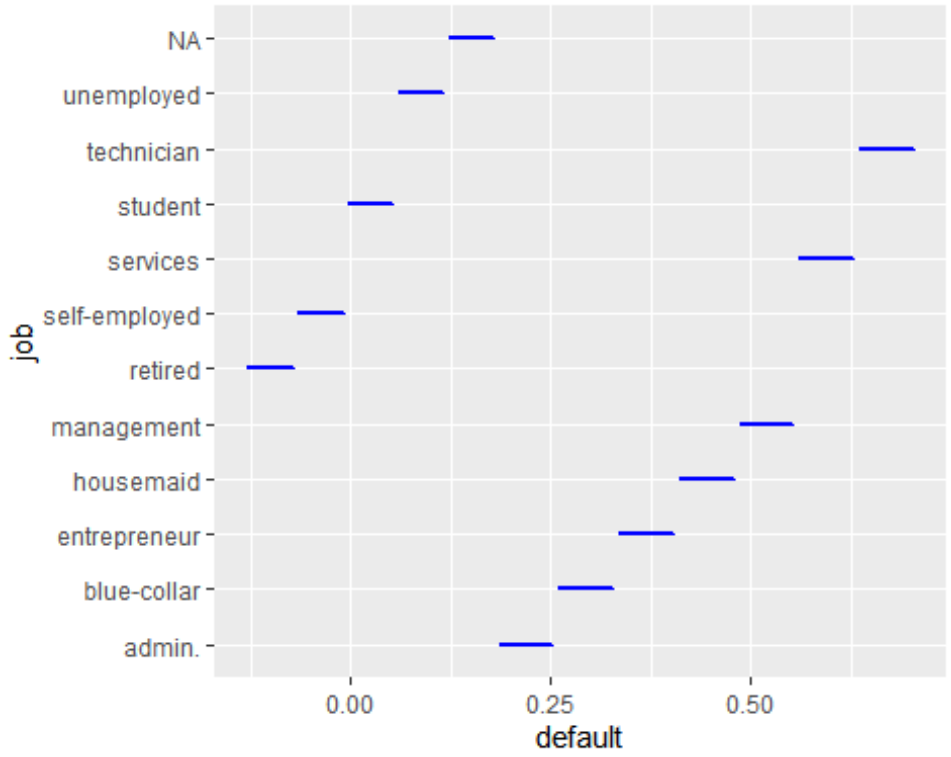
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



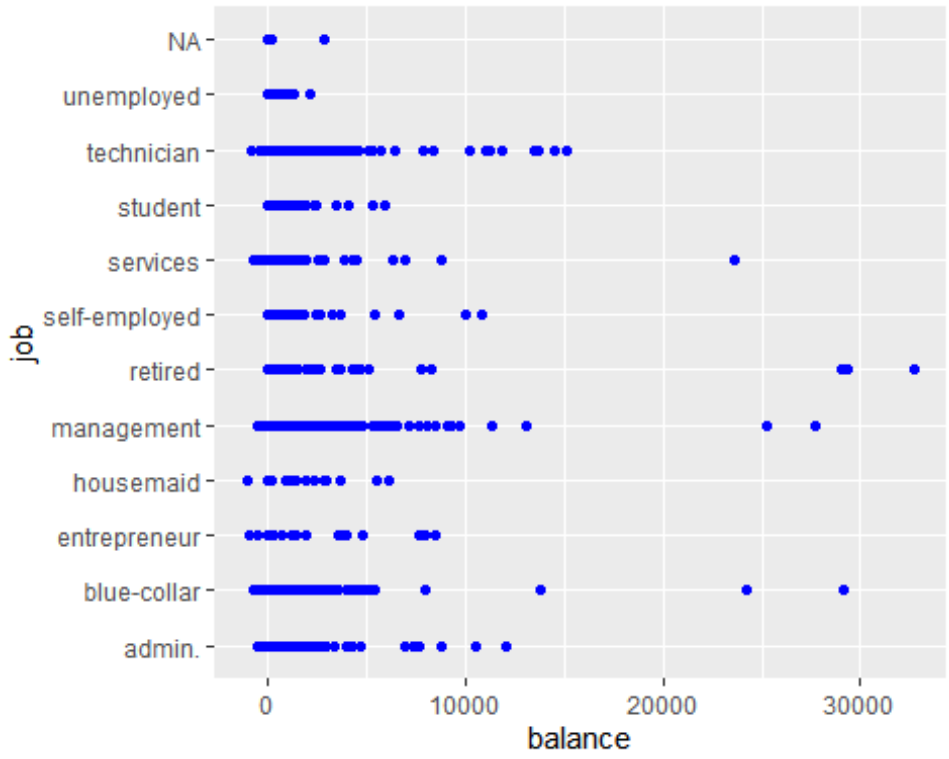
```
ggplot(bank12 ,aes(x=default ,y= age))+geom_point( color ='black')
```




```
ggplot(bank12 ,aes(x=default ,y= job))+geom_boxplot( color ='blue')
```



```
ggplot(bank12 ,aes(x=balance ,y= job))+geom_point( color ='blue')
```




```
#Correlation
setDF(bank)
corr_data<-data.frame(bank$default,bank$balance,bank$housing)
corr<-cor(corr_data)
corr

##              bank.default bank.balance bank.housing
## bank.default   1.00000000  -0.06095389   0.01107575
## bank.balance  -0.06095389   1.00000000  -0.07709205
## bank.housing   0.01107575  -0.07709205   1.00000000

#T-test

setDT(bank)
with(data=bank,t.test(age[default=="1"],age[default=="0"],var.equal=TRUE))

##
##  Two Sample t-test
##
## data:  age[default == "1"] and age[default == "0"]
## t = -1.207, df = 11160, p-value = 0.2275
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.9331657  0.6975263
## sample estimates:
```

```
## mean of x mean of y
##  40.13095  41.24877
```

```
with(data=bank,t.test(balance[default=="1"],balance[default=="0"],var.equal=T
RUE))
```

```
##
##  Two Sample t-test
##
## data:  balance[default == "1"] and balance[default == "0"]
## t = -6.4512, df = 11160, p-value = 1.155e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2105.247 -1124.041
## sample estimates:
##  mean of x  mean of y
##  -61.80357 1552.84064
```

```
with(data=bank,t.test(duration[default=="1"],duration[default=="0"],var.equal
=TRUE))
```

```
##
##  Two Sample t-test
##
## data:  duration[default == "1"] and duration[default == "0"]
## t = -1.0311, df = 11160, p-value = 0.3025
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -80.71927  25.07267
## sample estimates:
## mean of x mean of y
##  344.5893  372.4126
```

```
with(data=bank,t.test(housing[default=="1"],housing[default=="0"],var.equal=T
RUE))
```

```
##
##  Two Sample t-test
##
## data:  housing[default == "1"] and housing[default == "0"]
## t = 1.1701, df = 11160, p-value = 0.242
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.03066537  0.12150063
## sample estimates:
## mean of x mean of y
## 0.5178571 0.4724395
```