Bank Loan Logistic Reg

Piyush Jain

8/11/2020

```
# Hello Data Scientists

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(funModeling)
```

```
## Loading required package: Hmisc
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
##  / Now in Spanish: librovivodecienciadedatos.ai
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:Hmisc':
##
##     describe
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(ggplot2)
library(ggpubr)
library(ggthemes)
library(psych)
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```r
library(ROCR)
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```r
# Reading the Data set
bankloan <- read.csv("C:/Users/Admin/Desktop/Data Science/Imarticus/Projects/Bank Loan/bankloan.csv")
#View(bankloan)

# Check the dimension of the data frame
dim(bankloan)
```

```
## [1] 700   9
```

```r
# View all the column names
names(bankloan)
```

```
## [1] "age"      "ed"       "employ"   "address"  "income"   "debtinc"  "creddebt"
## [8] "othdebt"  "default"
```

```
str(bankloan)
```

```
## 'data.frame':    700 obs. of  9 variables:
##  $ age     : int  41 27 40 41 24 41 39 43 24 36 ...
##  $ ed      : int  3 1 1 1 2 2 1 1 1 1 ...
##  $ employ  : int  17 10 15 15 2 5 20 12 3 0 ...
##  $ address : int  12 6 14 14 0 5 9 11 4 13 ...
##  $ income  : int  176 31 55 120 28 25 67 38 19 25 ...
##  $ debtinc : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
##  $ creddebt: num  11.36 1.36 0.86 2.66 1.79 ...
##  $ othdebt : num  5.01 4 2.17 0.82 3.06 ...
##  $ default : int  1 0 0 0 1 0 0 0 1 0 ...
```

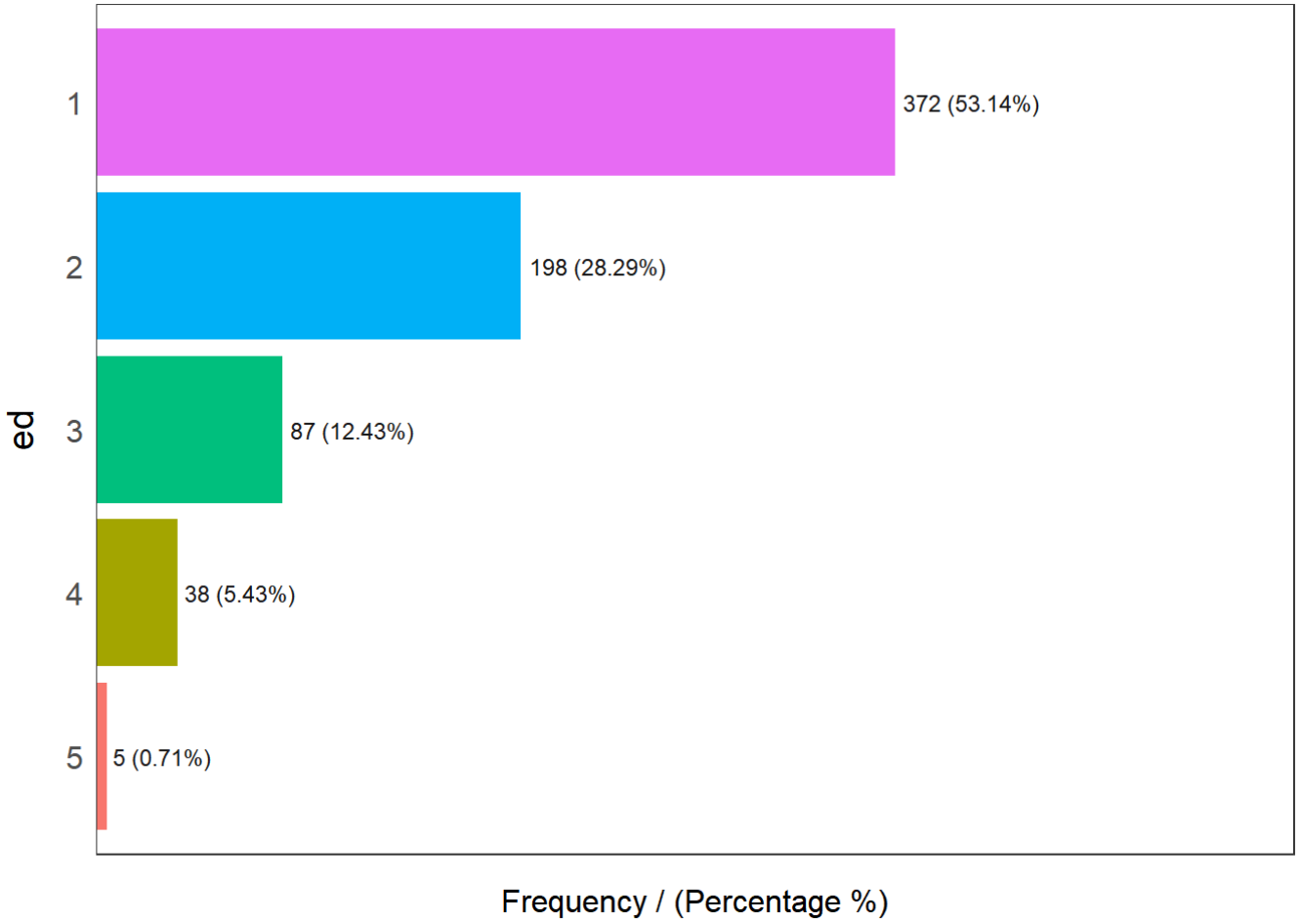
```
# ed and default are coming as integers but they are factors.
# Lets convert them into factors.
bankloan$ed <- as.factor(bankloan$ed)
bankloan$default <- as.factor(bankloan$default)

str(bankloan)
```

```
## 'data.frame':    700 obs. of  9 variables:
##  $ age     : int  41 27 40 41 24 41 39 43 24 36 ...
##  $ ed      : Factor w/ 5 levels "1","2","3","4",..: 3 1 1 1 2 2 1 1 1 1 ...
##  $ employ  : int  17 10 15 15 2 5 20 12 3 0 ...
##  $ address : int  12 6 14 14 0 5 9 11 4 13 ...
##  $ income  : int  176 31 55 120 28 25 67 38 19 25 ...
##  $ debtinc : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
##  $ creddebt: num  11.36 1.36 0.86 2.66 1.79 ...
##  $ othdebt : num  5.01 4 2.17 0.82 3.06 ...
##  $ default : Factor w/ 2 levels "0","1": 2 1 1 1 2 1 1 1 2 1 ...
```

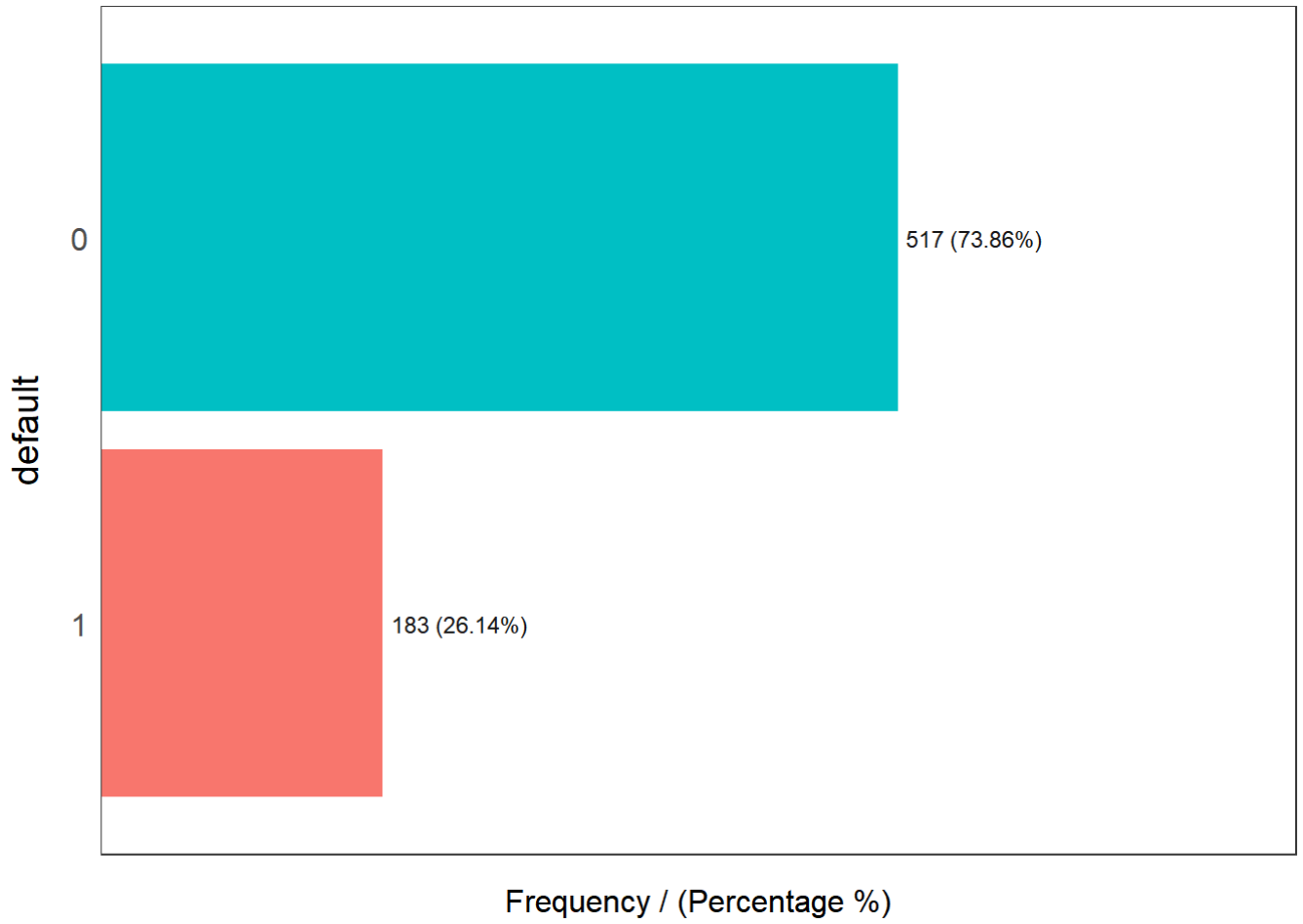
```
# ed and default are converted into Factors

status(bankloan)
```

```
##   variable q_zeros    p_zeros q_na p_na q_inf p_inf    type unique
## 1      age       0 0.00000000    0    0     0     0 integer     37
## 2       ed       0 0.00000000    0    0     0     0  factor      5
## 3   employ      62 0.08857143    0    0     0     0 integer     32
## 4  address      50 0.07142857    0    0     0     0 integer     31
## 5   income       0 0.00000000    0    0     0     0 integer    114
## 6  debtinc       0 0.00000000    0    0     0     0 numeric    231
## 7 creddebt       0 0.00000000    0    0     0     0 numeric    310
## 8  othdebt       0 0.00000000    0    0     0     0 numeric    429
## 9  default     517 0.73857143    0    0     0     0  factor      2
```

```
freq(bankloan)
```



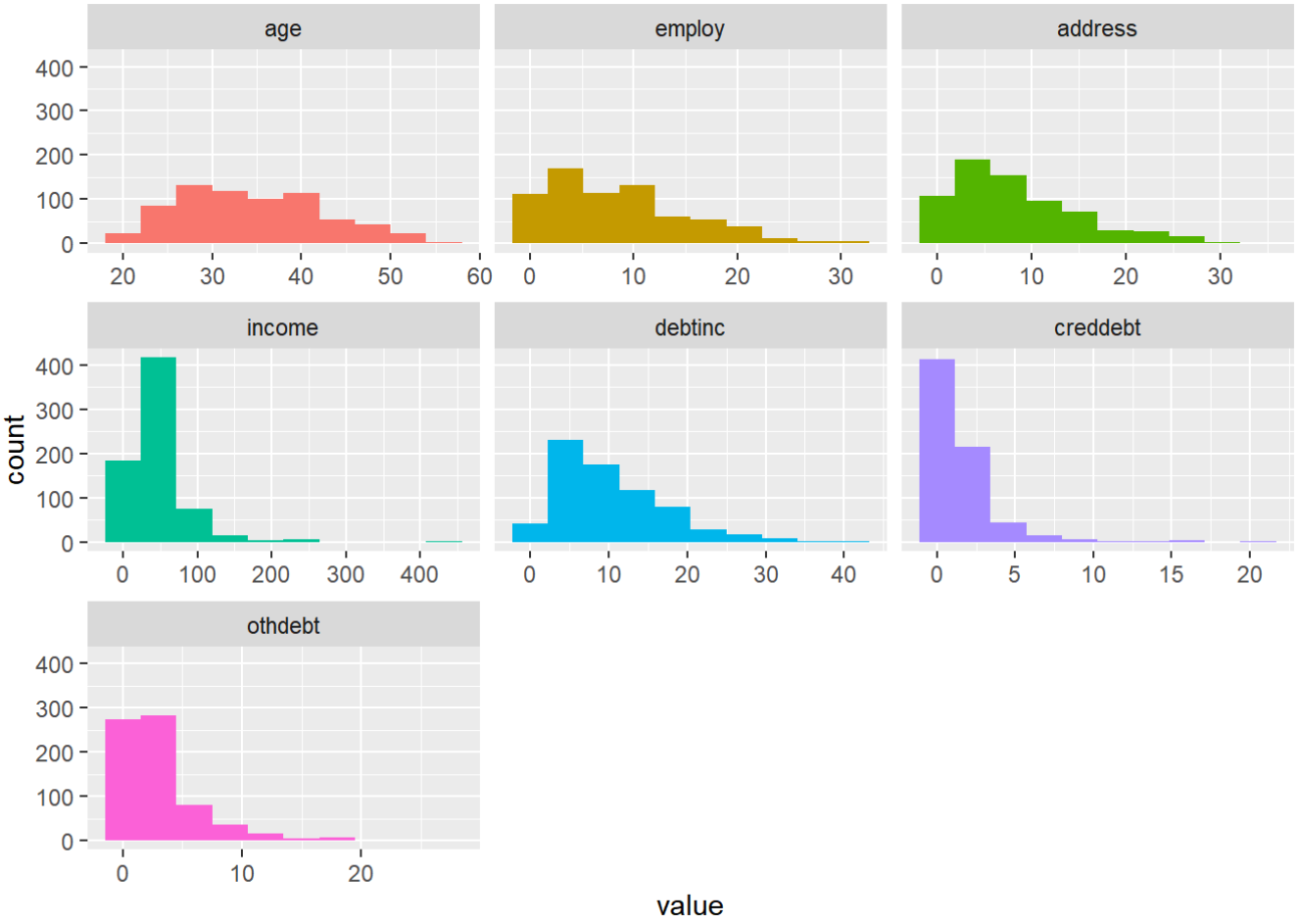

```
##   ed frequency percentage cumulative_perc
## 1  1       372      53.14           53.14
## 2  2       198      28.29           81.43
## 3  3        87      12.43           93.86
## 4  4        38       5.43           99.29
## 5  5         5       0.71          100.00
```




```
##   default frequency percentage cumulative_perc
## 1       0       517      73.86           73.86
## 2       1       183      26.14          100.00
```

```
## [1] "Variables processed: ed, default"
```

```
plot_num(bankloan) # Individual variable colorful graph
```




```
summary(bankloan)
```

```
##       age         ed          employ          address          income
##  Min.   :20.00   1:372   Min.   : 0.000   Min.   : 0.000   Min.   : 14.0
##  1st Qu.:29.00   2:198   1st Qu.: 3.000   1st Qu.: 3.000   1st Qu.: 24.0
##  Median :34.00   3: 87   Median : 7.000   Median : 7.000   Median : 34.0
##  Mean   :34.86   4: 38   Mean   : 8.389   Mean   : 8.279   Mean   : 45.6
##  3rd Qu.:40.00   5:  5   3rd Qu.:12.000   3rd Qu.:12.000   3rd Qu.: 55.0
##  Max.   :56.00           Max.   :31.000   Max.   :34.000   Max.   :446.0
```

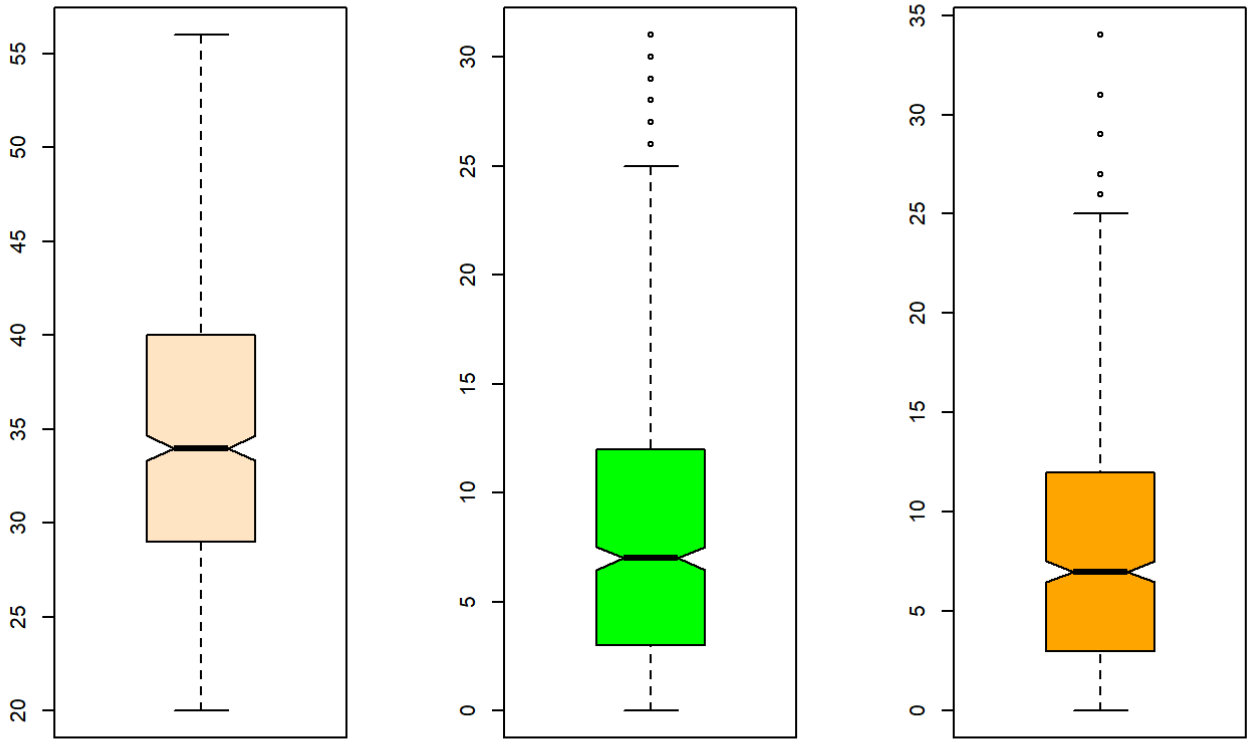
```
##     debtinc         creddebt         othdebt       default
##  Min.   : 0.40   Min.   : 0.010   Min.   : 0.050   0:517
##  1st Qu.: 5.00   1st Qu.: 0.370   1st Qu.: 1.048   1:183
##  Median : 8.60   Median : 0.855   Median : 1.985
##  Mean   :10.26   Mean   : 1.553   Mean   : 3.058
##  3rd Qu.:14.12   3rd Qu.: 1.905   3rd Qu.: 3.928
##  Max.   :41.30   Max.   :20.560   Max.   :27.030
```

```
# We could observe that there are no missing values.

describe(bankloan)
```

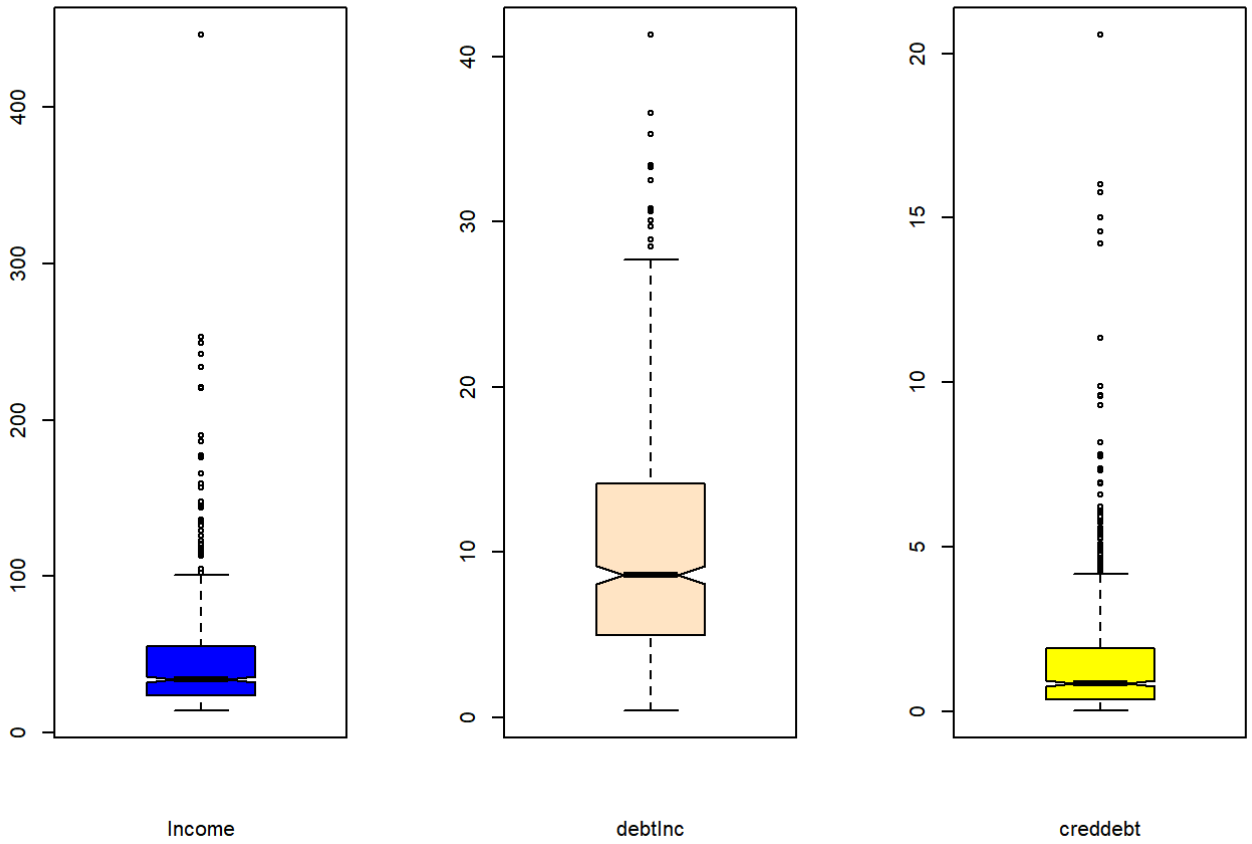
```
##          vars   n  mean    sd median trimmed   mad   min    max  range skew
## age         1 700 34.86  8.00  34.00   34.49  8.90 20.00  56.00  36.00 0.36
## ed*         2 700  1.72  0.93   1.00    1.57  0.00  1.00   5.00   4.00 1.20
## employ      3 700  8.39  6.66   7.00    7.72  7.41  0.00  31.00  31.00 0.83
## address     4 700  8.28  6.82   7.00    7.47  7.41  0.00  34.00  34.00 0.93
## income      5 700 45.60 36.81  34.00   38.82 17.79 14.00 446.00 432.00 3.84
## debtinc     6 700 10.26  6.83   8.60    9.48  6.23  0.40  41.30  40.90 1.09
## creddebt    7 700  1.55  2.12   0.86    1.13  0.88  0.01  20.56  20.55 3.88
## othdebt     8 700  3.06  3.29   1.98    2.41  1.66  0.05  27.03  26.98 2.72
## default*    9 700  1.26  0.44   1.00    1.20  0.00  1.00   2.00   1.00 1.08
##          kurtosis   se
## age         -0.62 0.30
## ed*          0.72 0.04
## employ       0.21 0.25
## address      0.30 0.26
## income      25.89 1.39
## debtinc      1.19 0.26
## creddebt    21.74 0.08
## othdebt     10.21 0.12
## default*    -0.83 0.02
```

```
# Income and Othdebt are highly skewed
```

```
########################## Univariant Analysis  ####################


# Box plot of each variable to detect outliers
par(mfrow=c(1,3))
boxplot(bankloan[,c(1)] , col = "bisque" ,notch=T , outline = T , xlab = "Age" )
boxplot(bankloan[,c(3)] , col = "green" ,notch=T , outline = T , xlab = "Employnment year")
boxplot(bankloan[,c(4)] , col = "orange" ,notch=T , outline = T , xlab = "Address year")
```




```
par(mfrow=c(1,3))
boxplot(bankloan[,c(5)] , col = "blue",notch=T , outline = T , xlab = "Income" )
boxplot(bankloan[,c(6)] , col = "bisque",notch=T , outline = T , xlab = "debtInc" )
boxplot(bankloan[,c(7)] , col = "yellow",notch=T , outline = T , xlab = "creddebt" )
```



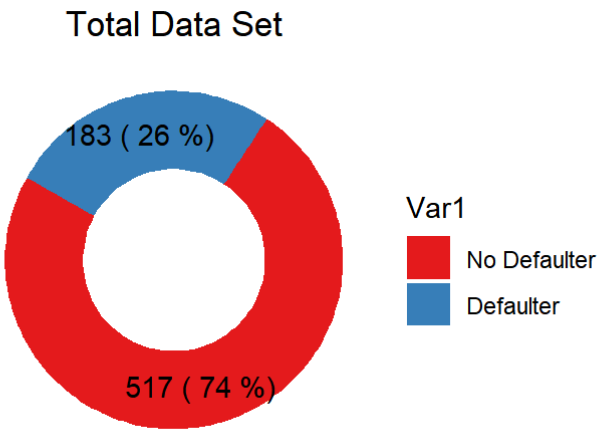

```
# Doghnut Plot of Distribution of Defaulters in Total Data
tdd <- table(bankloan$default)
names(tdd) = c("No Defaulter", "Defaulter")
tdd <- data.frame(tdd)
tdd$percent <- round((tdd$Freq/nrow(bankloan))*100 , digits = 0)

 total <-ggplot(data = tdd,
       aes(x = 2, y = Freq, fill = Var1))+
  geom_bar(stat = "identity")+
```

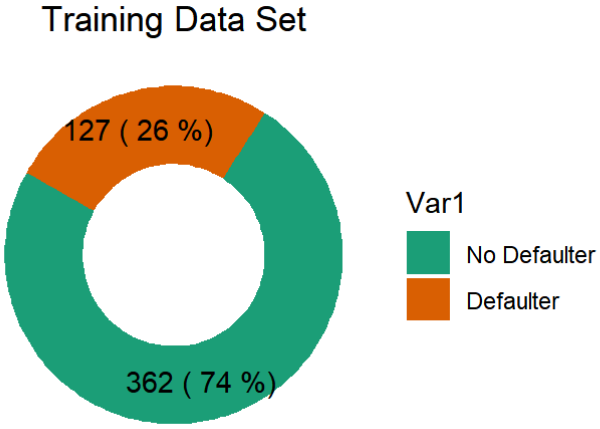
```
  coord_polar( "y",start = 200) +
  geom_text(aes(label = paste(Freq , "(",percent,"%)" , spe = "")), col = "black" ,  position = position_stack(vj
ust = 0.5)) +
  theme_void() +
  scale_fill_brewer(palette = "Set1")+
  xlim(.5,2.5) + labs(title ="Total Data Set") +
   theme(plot.title = element_text(hjust = .5))

 tdd
```

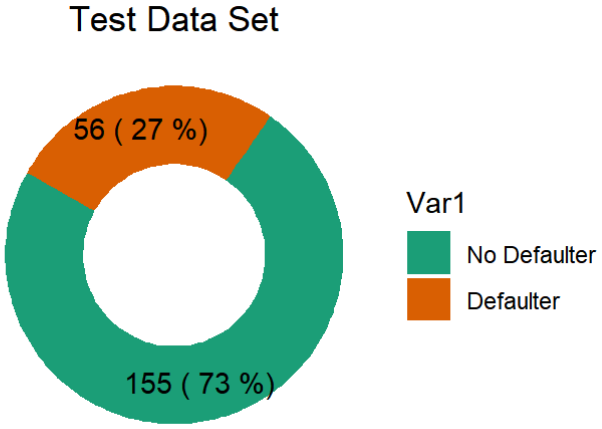
```
##           Var1 Freq percent
## 1 No Defaulter  517      74
## 2    Defaulter  183      26
```

```
## We could see that total defaulters are 183 and Non defaulters are 517

# Splitting the data into Train and Test
set.seed(42)
default_idx = sample(nrow(bankloan), 0.7*nrow(bankloan))
default_trn = bankloan[default_idx, ]   # Training Data
default_tst = bankloan[-default_idx, ]  # Testing Data


# Doghnut Plot of Distribution of Defaulters in Training Data
tab_trn = xtabs(~default_trn$default , data = default_trn)

names(tab_trn) = c("No Defaulter", "Defaulter")
tab_trn <- data.frame(tab_trn)
tab_trn$percent <- round((tab_trn$Freq/nrow(default_trn))*100 , digits = 0)

train <-ggplot(data = tab_trn, aes(x = 2, y = Freq, fill = Var1))+
  geom_bar(stat = "identity")+
  coord_polar( "y",start = 200) +
  geom_text(aes(label = paste(Freq , "(",percent,"%)" , spe = "")), col = "black" ,  position = position_stack(vj
ust = 0.5)) +
  theme_void() +
```

```
  scale_fill_brewer(palette = "Dark2")+
  xlim(.5,2.5) +  labs(title ="Training Data Set") +
  theme(plot.title = element_text(hjust = .5))


# Doghnut Plot of Distribution of Defaulters in Test Data
tab_tst <- xtabs(~default_tst$default , data = default_tst)

names(tab_tst) = c("No Defaulter", "Defaulter")
tab_tst <- data.frame(tab_tst)
tab_tst$percent <- round((tab_tst$Freq/nrow(default_tst))*100 , digits = 0)

test <-ggplot(data = tab_tst, aes(x = 2, y = Freq, fill = Var1))+
  geom_bar(stat = "identity")+
  coord_polar( "y",start = 200) +
  geom_text(aes(label = paste(Freq , "(",percent,"%)" , spe = "")), col = "black" ,  position = position_stack(vj
ust = 0.5)) +
  theme_void() +
  scale_fill_brewer(palette = "Dark2")+
  xlim(.5,2.5) +  labs(title ="Test Data Set") +
  theme(plot.title = element_text(hjust = .5))

#install.packages("ggpubr")
library(ggpubr)
ggarrange(total,
          ggarrange(train, test, ncol = 2),
          nrow = 2 )
```

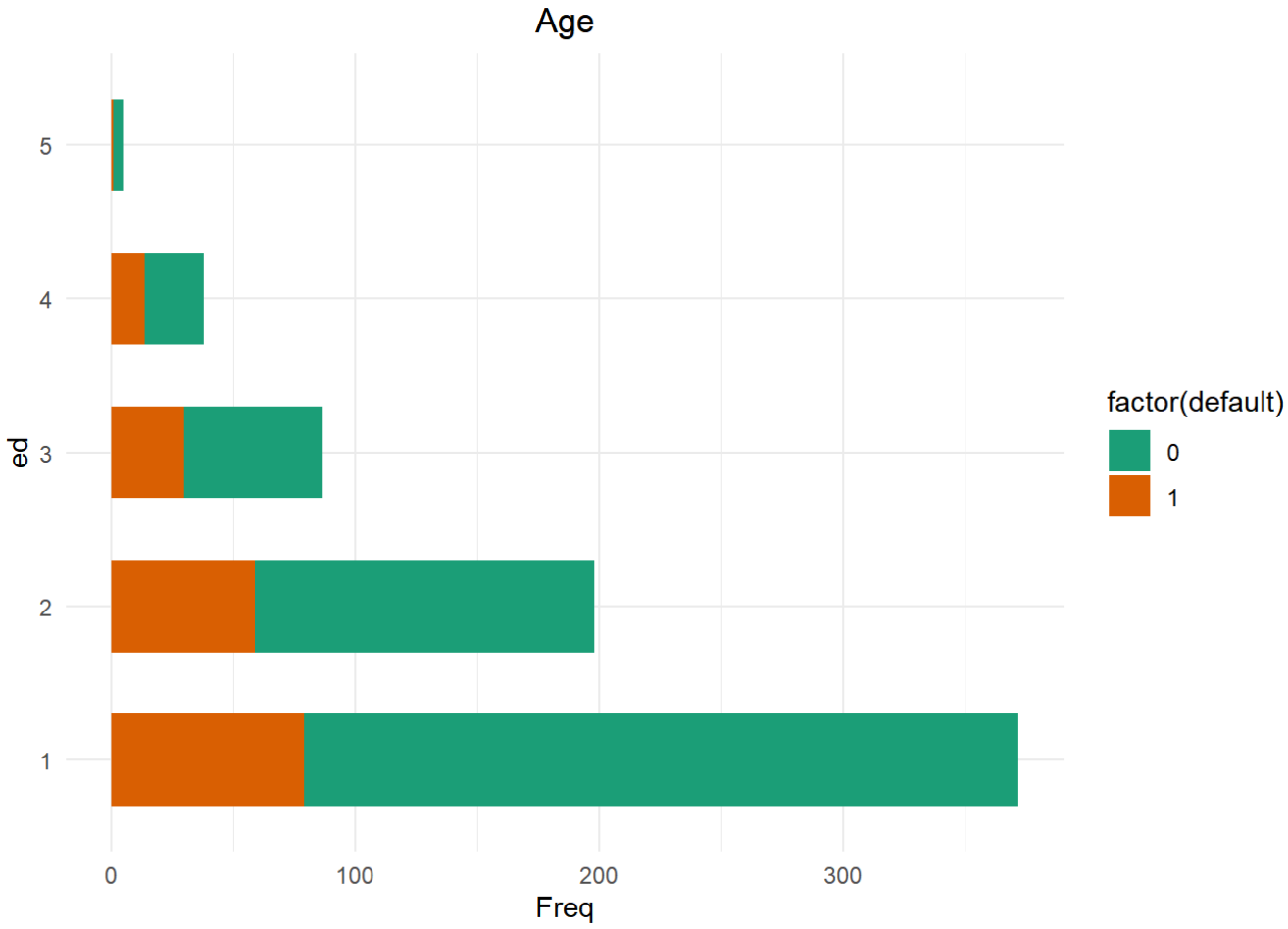






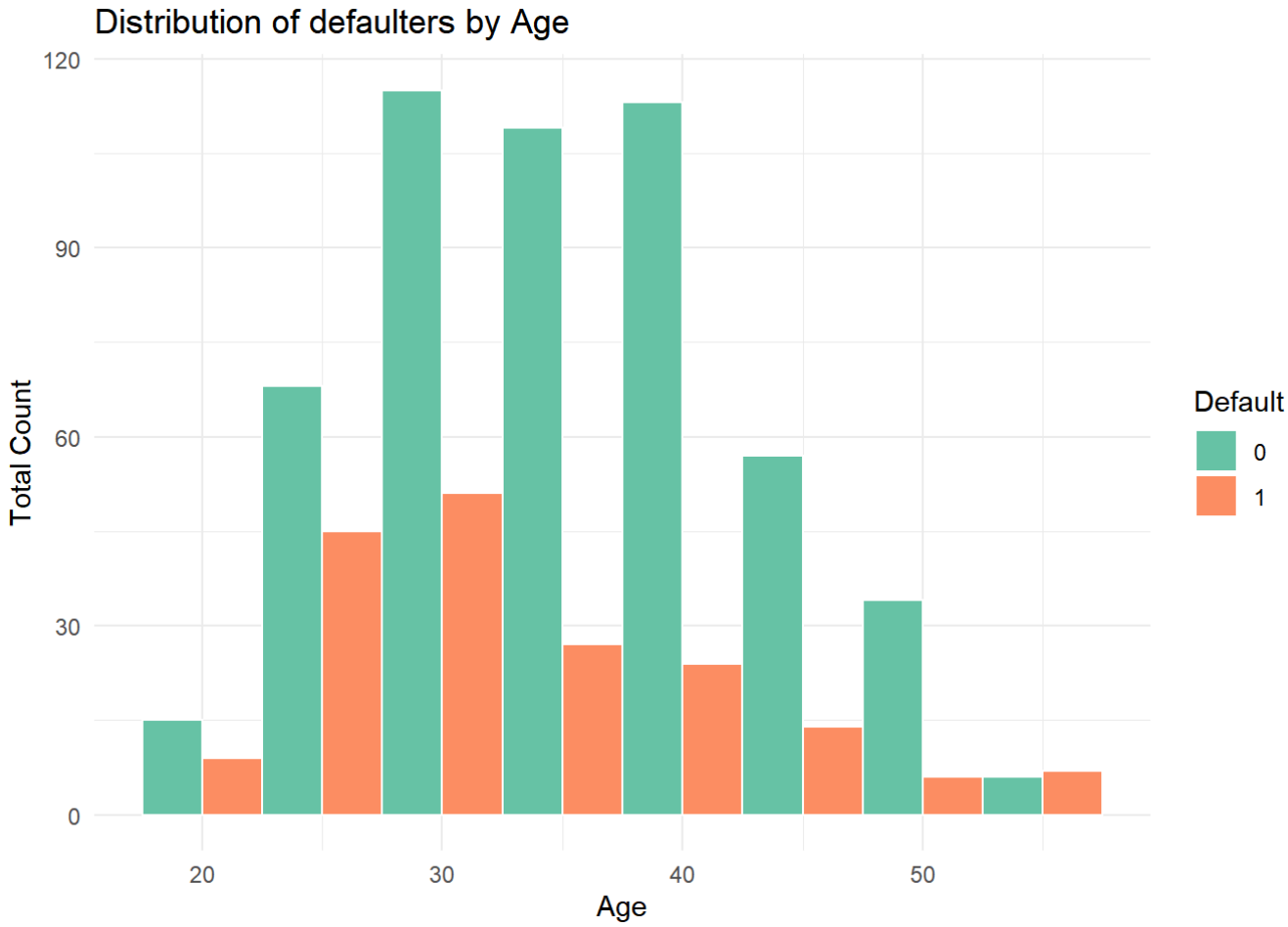

```
# Bar graph of Education and Defaulter

tab_edu <-xtabs(~ ed+default , data = bankloan)
tab_edu <- data.frame(tab_edu)

  ggplot(data = tab_edu , aes( x = ed  , y= Freq , fill = factor(default))) +
  geom_bar(stat = "identity", width = .6) +
  coord_flip() +
  labs(title="Age") +
```

```
  theme_minimal() +
theme(plot.title = element_text(hjust = .5), axis.ticks = element_blank()) +
scale_fill_brewer(palette = "Dark2")
```



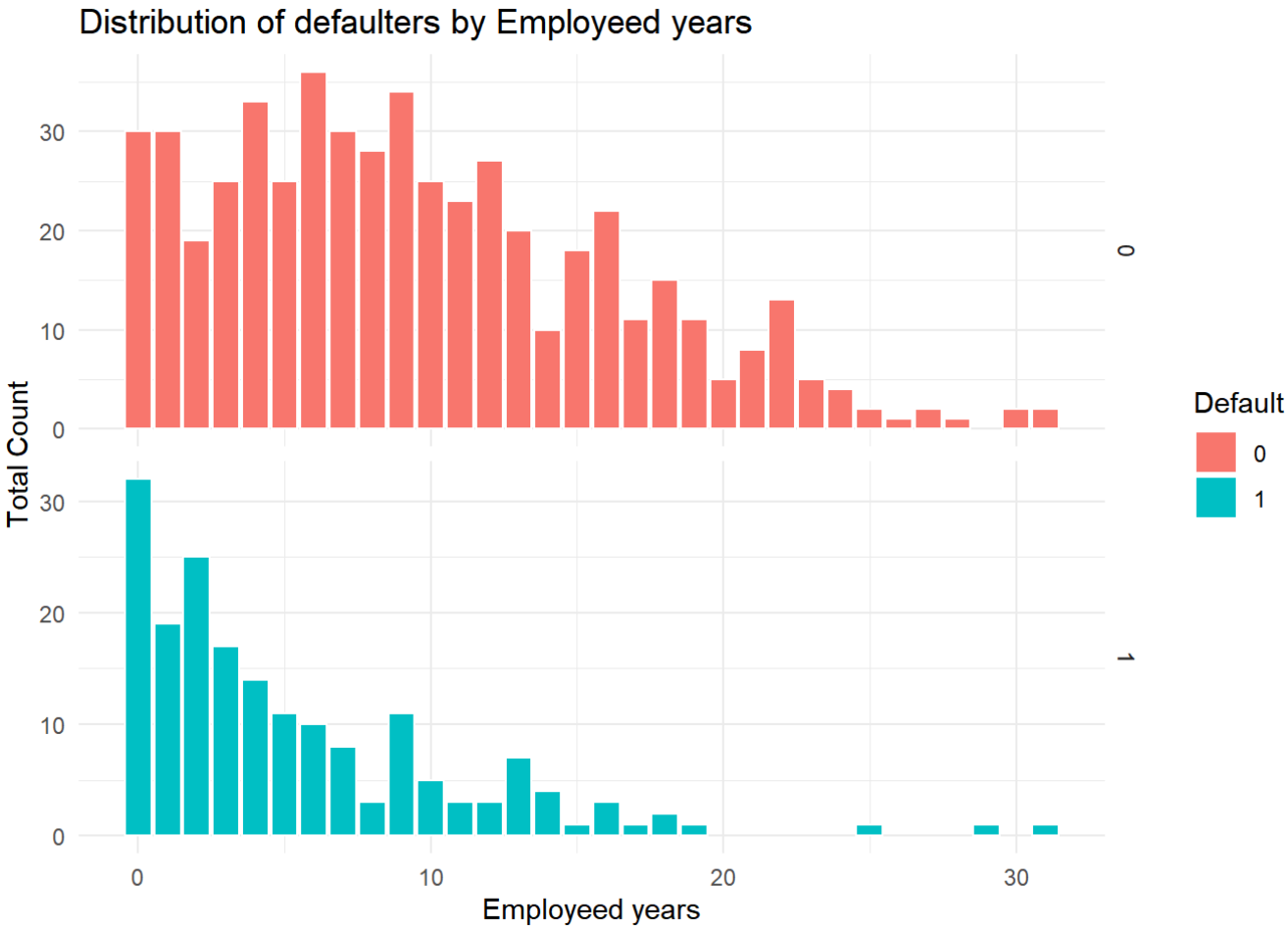

```
## We can observe that with higher education there are less defaulters.


# Histogram of Age and Defaulter
ggplot(data = bankloan , aes( x = age , fill = factor(default)))+
```

```
  geom_histogram(binwidth = 5 ,position = "dodge", color ="white" ) + theme_minimal() +
  labs(y = "Total Count",
     fill = "Default",
     x = "Age",
     title = "Distribution of defaulters by Age") +
  scale_fill_brewer(palette="Set2")
```

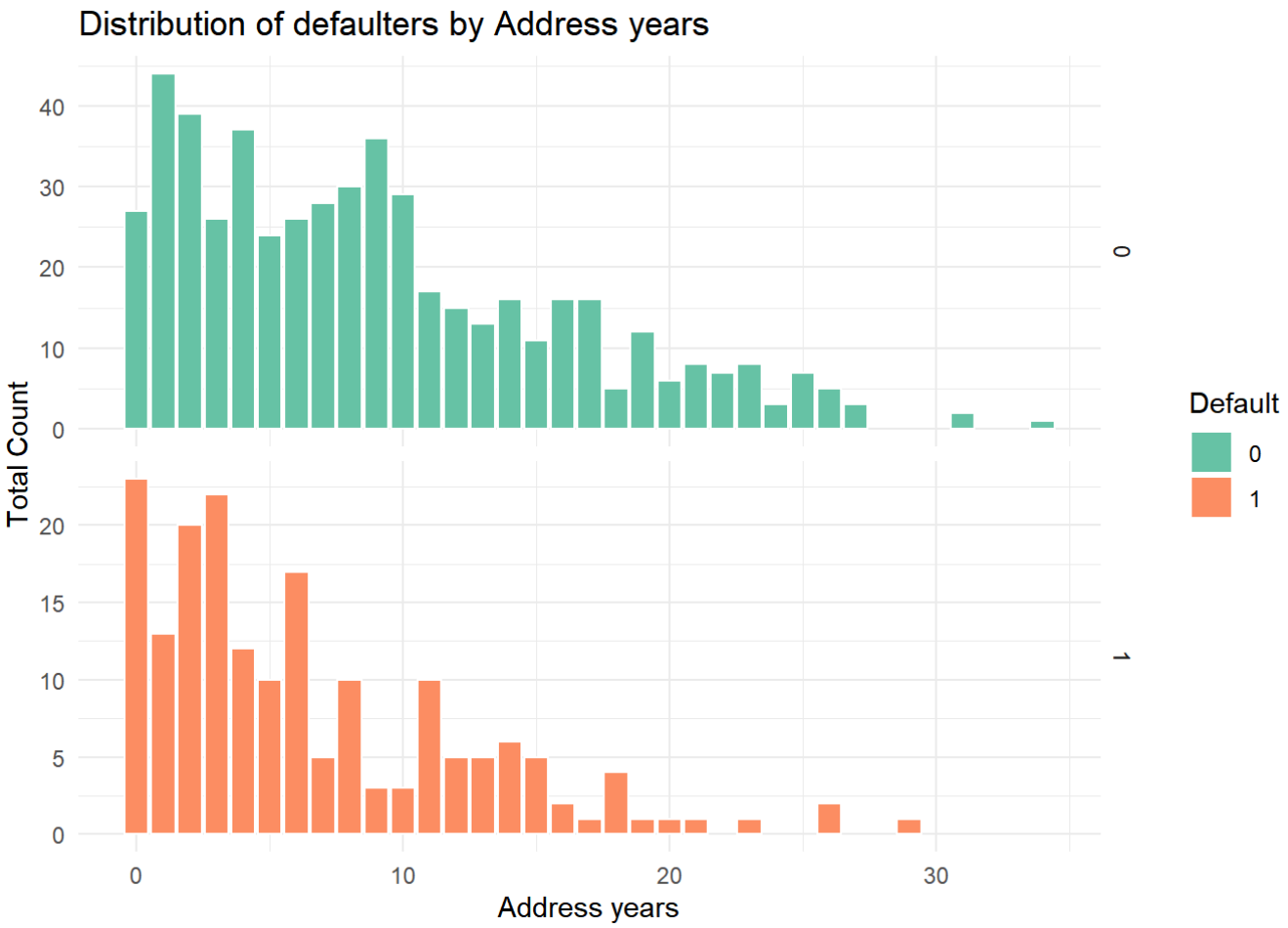



```
## We can observe that between age group of 25 - 45 the default rate is high
```

```r
# Histogram of Employeed year and Defaulter
ggplot(data = bankloan , aes( x = employ , fill = factor(default)))+
  geom_bar(color="white") + theme_minimal() +
  labs(y = "Total Count",
       fill = "Default",
       x = "Employeed years",
       title = "Distribution of defaulters by Employeed years") +
    facet_grid(default ~., scales = "free")
```



```
## We can observe that in employed years less than 10 there is higher chance of being defaulter.

# Histogram of Address year and Defaulter
ggplot(data = bankloan , aes( x = address ,fill = factor(default)))+
  geom_bar(color="white") + theme_minimal() +
  labs(y = "Total Count",
       fill = "Default",
       x = "Address years",
       title = "Distribution of defaulters by Address years") +
  scale_fill_brewer(palette="Set2")+
  facet_grid(default ~., scales = "free")
```

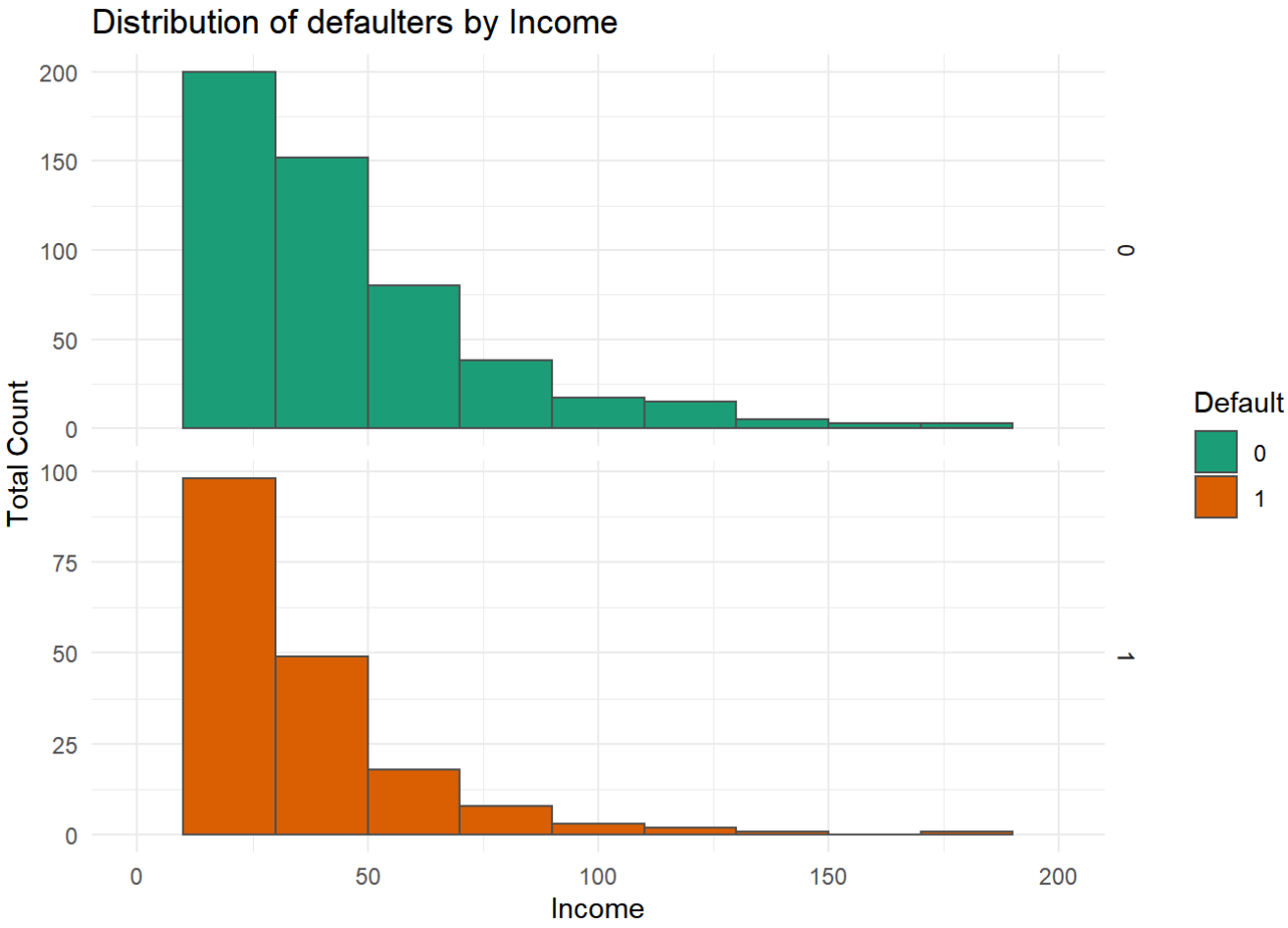



```
## We can observe that in address years less than 10 there is higher chance of being defaulter.

# Histogram of Income and Defaulter
ggplot(data = bankloan , aes( x = income ,fill = factor(default)))+
  geom_histogram(binwidth = 20, color = "grey30")+  xlim(0,200) + theme_minimal() +
  labs(y = "Total Count",
       fill = "Default",
       x = "Income",
       title = "Distribution of defaulters by Income") +
```

```
  scale_fill_brewer(palette="Dark2")+
  facet_grid (default ~., scales = "free")
```

```
## Warning: Removed 7 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



```
## For income less than 50 chances of being a defaulter is high


## T test
ttd=t.test(bankloan$age~bankloan$default,var.equal = TRUE,alternative = "two.sided")
ttd
```

```
##
##  Two Sample t-test
##
## data:  bankloan$age by bankloan$default
## t = 3.6718, df = 698, p-value = 0.0002592
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.164881 3.842275
## sample estimates:
## mean in group 0 mean in group 1
##        35.51451        33.01093
```

```
ttd2=t.test(bankloan$employ~bankloan$default,var.equal = TRUE,alternative = "two.sided")
ttd2
```

```
##
##  Two Sample t-test
##
## data:  bankloan$employ by bankloan$default
## t = 7.7948, df = 698, p-value = 2.347e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3.205433 5.363888
## sample estimates:
## mean in group 0 mean in group 1
##        9.508704        5.224044
```

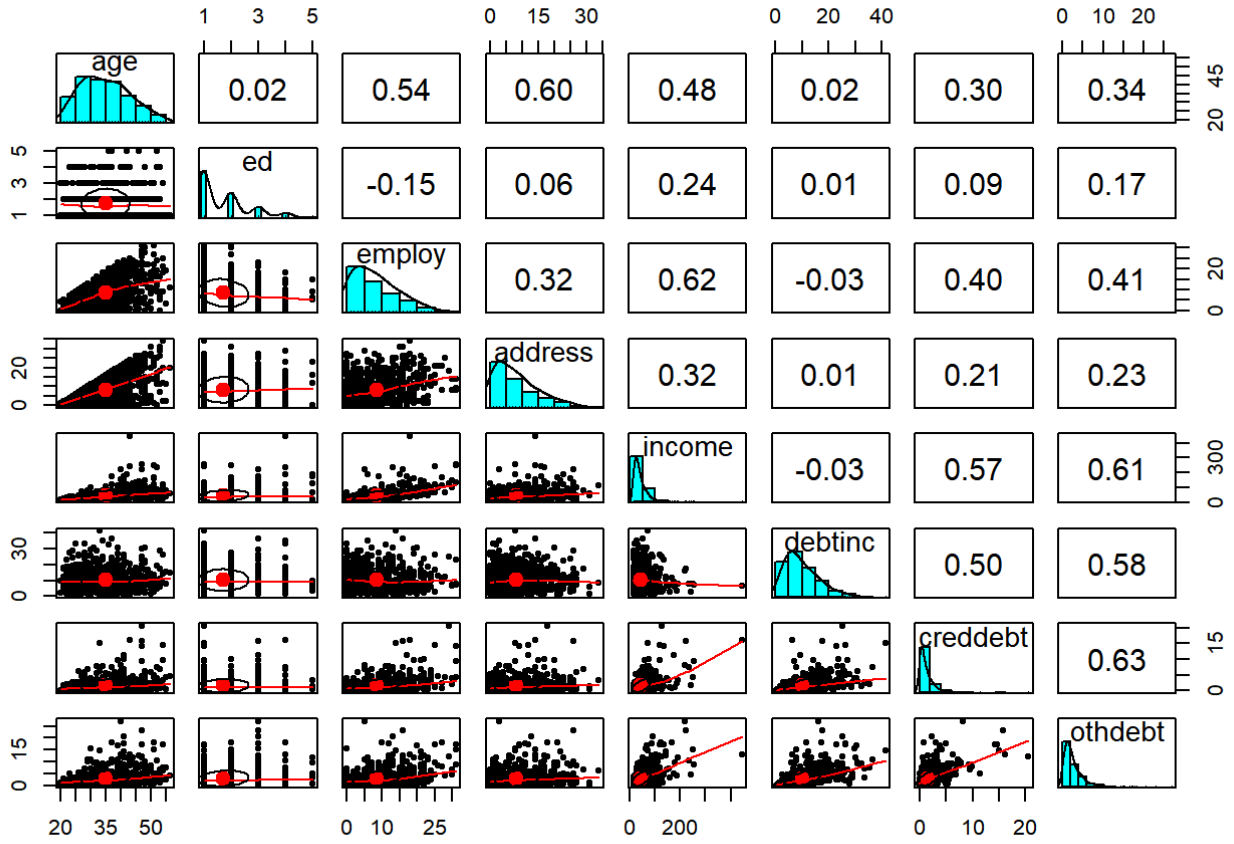
```
ttd3=t.test(bankloan$address~bankloan$default,var.equal = TRUE,alternative = "two.sided")
ttd3
```

```
##
##  Two Sample t-test
##
## data:  bankloan$address by bankloan$default
## t = 4.4047, df = 698, p-value = 1.226e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.414687 3.690111
## sample estimates:
## mean in group 0 mean in group 1
##        8.945841        6.393443
```

```
ttd4=t.test(bankloan$income~bankloan$default,var.equal = TRUE,alternative = "two.sided")
ttd4
```

```
##
##  Two Sample t-test
##
## data:  bankloan$income by bankloan$default
## t = 1.8797, df = 698, p-value = 0.06056
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.264352 12.147600
## sample estimates:
## mean in group 0 mean in group 1
##        47.15474        41.21311
```

```
library(psych)
pairs.panels(bankloan[,1:8])
```




```
# CREATE MODEL WITH TRAIN DATASET
model1<-glm(default~.,data=default_trn,family='binomial')
summary(model1)
```

```
##
## Call:
## glm(formula = default ~ ., family = "binomial", data = default_trn)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.8380  -0.6431  -0.3092   0.2493   2.7588
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.423939   0.736490  -3.291 0.000998 ***
## age           0.052312   0.022153   2.361 0.018208 *
## ed2           0.569352   0.301198   1.890 0.058719 .
## ed3           0.267572   0.401745   0.666 0.505395
## ed4          -0.189559   0.584449  -0.324 0.745683
## ed5         -11.815952 590.161416  -0.020 0.984026
## employ       -0.258482   0.039926  -6.474 9.54e-11 ***
## address      -0.110776   0.028849  -3.840 0.000123 ***
## income       -0.006336   0.008301  -0.763 0.445272
## debtinc       0.092850   0.036443   2.548 0.010839 *
## creddebt      0.546455   0.128356   4.257 2.07e-05 ***
## othdebt       0.067674   0.092029   0.735 0.462124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 560.16  on 488  degrees of freedom
## Residual deviance: 380.95  on 477  degrees of freedom
## AIC: 404.95
##
## Number of Fisher Scoring iterations: 13
```

```
## Based on output of model 1 we can observe that ed ,income, othdebt are
## in-significant data as per p-value.

# BUILDING MODEL 2 BY REMOVING INSIGNIFICANT PREDICTOS
model2<-glm(default~age+employ+address+debtinc+creddebt,data=default_trn,family='binomial')
summary(model2)
```

```
##
## Call:
## glm(formula = default ~ age + employ + address + debtinc + creddebt,
##     family = "binomial", data = default_trn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9398  -0.6619  -0.3150   0.2157   2.6769
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.35619    0.62978  -3.741 0.000183 ***
## age          0.04738    0.02139   2.215 0.026782 *
## employ      -0.25099    0.03593  -6.986 2.82e-12 ***
## address     -0.10606    0.02819  -3.762 0.000169 ***
## debtinc      0.11604    0.02271   5.110 3.21e-07 ***
## creddebt     0.49425    0.09728   5.081 3.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 560.16  on 488  degrees of freedom
## Residual deviance: 386.68  on 483  degrees of freedom
## AIC: 398.68
##
## Number of Fisher Scoring iterations: 6
```

```r
# PREDICTIONS
pred_log=predict(model2,default_tst,type='response')

# ROC CURVE WITH 2 ,3 THREASHOLDS
library(ROCR)
per_log1=prediction(pred_log,default_tst$default)
acc=performance(per_log1,"acc")
plot(acc,colorrize=T) # find threashold value = 1.0
```

```r
#Build confusion matrix with above determined threashold value to get maximum accuracy
table(default_tst$default, pred_log>0.4)
```

```
##
##     FALSE TRUE
##   0   135   20
##   1    21   35
```

```r
#     FALSE  TRUE
# 0    135    20
# 1     21    35


ROC_Curve =performance(per_log1,"tpr","fpr")
plot(ROC_Curve,colorize=T)
abline(a=0,b=1)
#We need to take value when TPR and FPR are in green shape here its 0.4 to 0.6 e.g 0.5 threshold

# AUC - AREA UNDER CURVE
auc=performance(per_log1,"auc")
auc=unlist(slot(auc,"y.values"))
auc
```

```
## [1] 0.8620968
```

```r
auc=round(auc,4)
auc
```

```
## [1] 0.8621
```

```r
legend(.6,.4, auc,title="AUC")
```

```r
get_logistic_pred = function(mod, data, res = "y", pos = 1, neg = 0, cut = 0.5) {
  probs = predict(mod, newdata = data, type = "response")
  ifelse(probs > cut, pos, neg)
}

test_pred_50 = get_logistic_pred(model2, data = default_tst, res = "default",
                                 pos = 1, neg = 0, cut = 0.5)

test_tab_50 = table(predicted = test_pred_50, actual = default_tst$default)

library(caret)
test_con_mat_50 = confusionMatrix(test_tab_50)
test_con_mat_50
```

```
## Confusion Matrix and Statistics
##
##          actual
## predicted   0   1
##         0 141  27
##         1  14  29
##
##                Accuracy : 0.8057
##                  95% CI : (0.7458, 0.8568)
##     No Information Rate : 0.7346
##     P-Value [Acc > NIR] : 0.01018
##
##                   Kappa : 0.4618
##
##  Mcnemar's Test P-Value : 0.06092
##
##             Sensitivity : 0.9097
##             Specificity : 0.5179
##          Pos Pred Value : 0.8393
##          Neg Pred Value : 0.6744
##              Prevalence : 0.7346
##          Detection Rate : 0.6682
##    Detection Prevalence : 0.7962
```

```
##       Balanced Accuracy : 0.7138
##
##        'Positive' Class : 0
##
```

```
# UPSAMPLING ON OVERALL DATASET
#install.packages("ROSE")
library(ROSE)   #RANDOMELY OVERSAMPLING EXAMPLES
table(bankloan$default)
```

```
##
##   0   1
## 517 183
```

```
# 0   1
# 517 183
over<- ovun.sample(default~.,data = bankloan,method = "over",N=1034)$data

table(over$default)
```

```
##
##   0   1
## 517 517
```

```
# 0   1
# 517 517

#BELOW IN N WE NEED TO PUT VALUE FOR 0 WE CAN SEE THERE ARE 363 ENTRIES FOR 0
#AND IF WE WANT SAME FOR ONE THEN PUT 363*2 IN N=726 TO GET EQUAL SAMPLE FOR 0 AND 1


# DATA SPLIT of BALANCED DATA
set.seed(123)
sample_data_Balanced=sample(2,nrow(over),replace = T,prob = c(.7,.3))
```

```
train_data_balanced=over[sample_data_Balanced==1,]
dim(train_data_balanced) #733   9
```

```
## [1] 733   9
```

```
test_data_balanced=over[sample_data_Balanced==2,]
dim(test_data_balanced) #301   9
```

```
## [1] 301   9
```

```
# Build a model
model_log2=glm(default~., data = train_data_balanced,family = 'binomial')
length(model_log2)
```

```
## [1] 30
```

```
summary(model_log2)
```

```
##
## Call:
## glm(formula = default ~ ., family = "binomial", data = train_data_balanced)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.79380  -0.77394  -0.06927   0.75969   2.55201
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.341033   0.537399  -0.635  0.52569
## age          0.031216   0.015990   1.952  0.05091 .
## ed2          0.225657   0.229204   0.985  0.32486
## ed3         -0.084408   0.304519  -0.277  0.78164
```

```
## ed4         -0.359318   0.410734  -0.875  0.38167
## ed5          0.309076   1.404364   0.220  0.82581
## employ      -0.242749   0.027432  -8.849  < 2e-16 ***
## address     -0.096051   0.020769  -4.625 3.75e-06 ***
## income      -0.008169   0.007598  -1.075  0.28229
## debtinc      0.075651   0.028517   2.653  0.00798 **
## creddebt     0.642721   0.101984   6.302 2.93e-10 ***
## othdebt     -0.030604   0.079532  -0.385  0.70039
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1016.0  on 732  degrees of freedom
## Residual deviance:  680.5  on 721  degrees of freedom
## AIC: 704.5
##
## Number of Fisher Scoring iterations: 5
```

```
#BUILD BEST FIT MODEL
model_log3=glm(default~employ+address+creddebt, data = train_data_balanced,family = 'binomial')
model_log3
```

```
##
## Call:  glm(formula = default ~ employ + address + creddebt, family = "binomial",
##     data = train_data_balanced)
##
## Coefficients:
## (Intercept)       employ      address     creddebt
##     1.04364     -0.26298     -0.08192      0.77093
##
## Degrees of Freedom: 732 Total (i.e. Null);  729 Residual
## Null Deviance:       1016
## Residual Deviance: 714.1     AIC: 722.1
```

```
summary(model_log3)
```

```
##
## Call:
## glm(formula = default ~ employ + address + creddebt, family = "binomial",
##     data = train_data_balanced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6775  -0.8298  -0.0869   0.8059   2.3123
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.04364    0.16593   6.290 3.18e-10 ***
## employ      -0.26298    0.02369 -11.100  < 2e-16 ***
## address     -0.08192    0.01666  -4.916 8.82e-07 ***
## creddebt     0.77093    0.07500  10.279  < 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: 1016.04  on 732  degrees of freedom
## Residual deviance:  714.08  on 729  degrees of freedom
## AIC: 722.08
##
## Number of Fisher Scoring iterations: 5
```

```
#PREDICT
pred_log3=predict(model_log3,test_data_balanced,type='response')

table(test_data_balanced$default)
```

```
##
##   0   1
## 146 155
```

```
# 0   1
# 146 155

# CREATE CONFUSION MATRIX
table(test_data_balanced$default, pred_log3>=0.5)
```

```
##
##     FALSE TRUE
##   0   109   37
##   1    32  123
```

```
accuracy_balanced =  (109+123)/(109+37+32+123)
accuracy_balanced
```

```
## [1] 0.7707641
```

```
true_positive_rate_balanced=(123)/(123+32) # Recall/Sensitity=TP/TP+FN Sesitivity shows how relevant model is in
 terms of positive result
true_positive_rate_balanced
```

```
## [1] 0.7935484
```

```
false_positive_rate_balanced=(37)/(37+109) #FP/FP+TN
false_positive_rate_balanced
```

```
## [1] 0.2534247
```

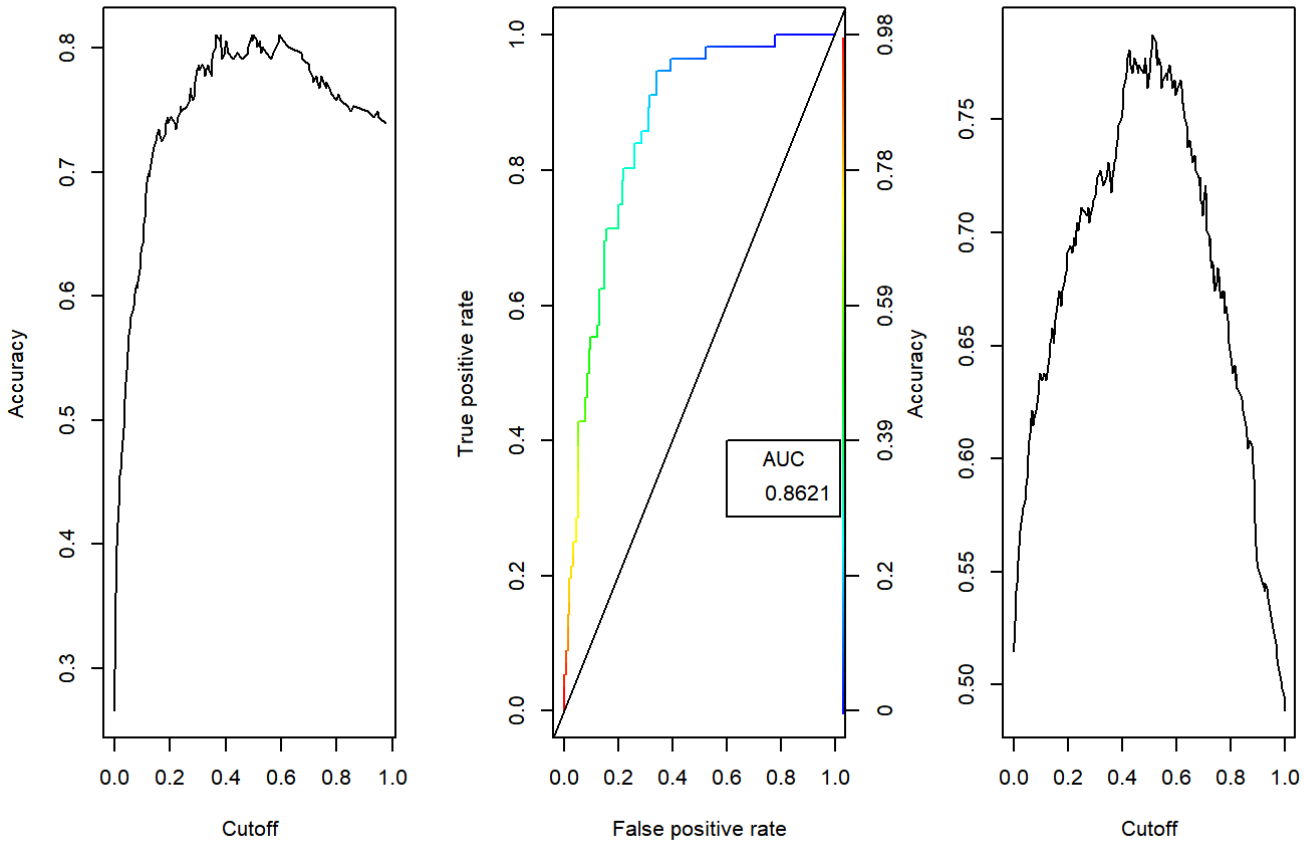
```
precision_balanced=(123)/(125+32)   # TP/TP+FP =predicted truly relevant result, among +ve predictors how many ar
e true
precision_balanced
```

```
## [1] 0.7834395
```

```
f2score_balanced=(2*true_positive_rate_balanced*precision_balanced)/
  (true_positive_rate_balanced+precision_balanced)
f2score_balanced      # mean between precision and recall,best measure of performance in situations with imbalanc
ed data
```

```
## [1] 0.7884615
```

```
# ROC CURVE
per_log3=prediction(pred_log3,test_data_balanced$default)
ROC_Curve3=performance(per_log3,"acc")
plot(ROC_Curve3,colorrize=T) # find threashold value = 1.0
```




```
#Build confusion matrix with above determined threashold value to get maximum accuracy
ROC_Curve3=performance(per_log3,"tpr","fpr")
# plot(ROC_Curve3,colorize=T)
plot(ROC_Curve3, colorize=T,main="ROC curve Best fit model",ylab="TPR(sensitivity)",xlab="FPR(1-specificity)")
abline(a=0,b=1)

# AUC - AREA UNDER CURVE
auc3=performance(per_log3,"auc")
```

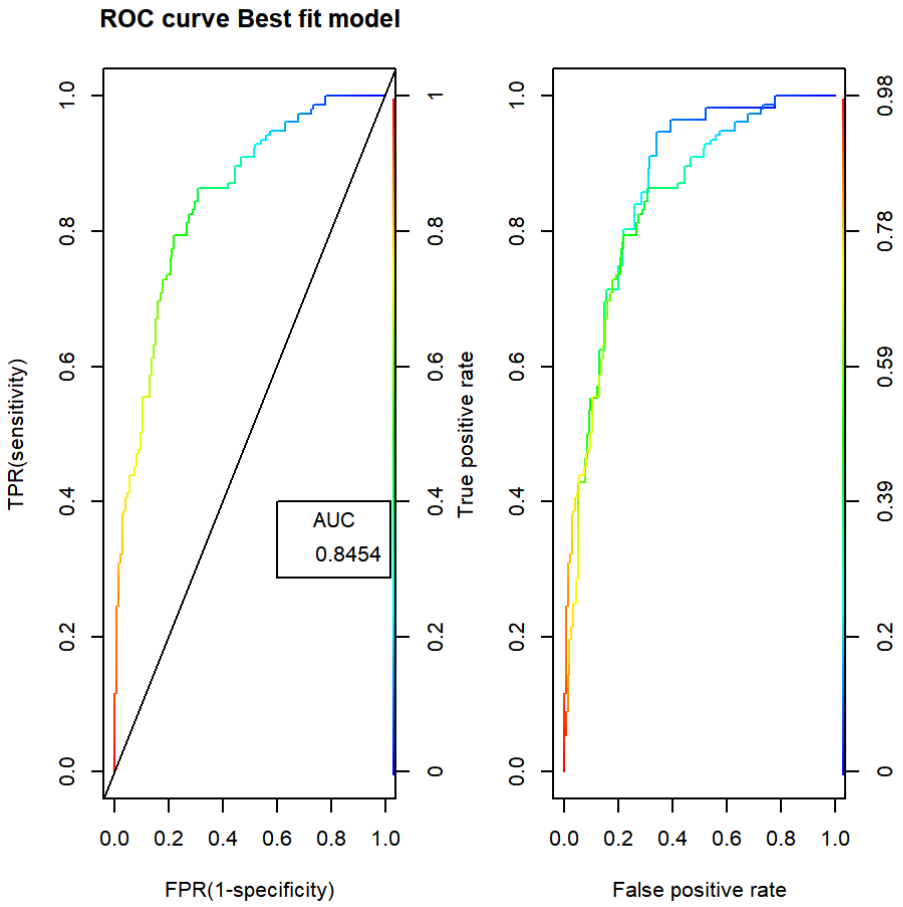
```
auc3=unlist(slot(auc3,"y.values"))
auc3     # 0.8402121
```

```
## [1] 0.8454264
```

```
auc3=round(auc3,4)
auc3 #0.8402
```

```
## [1] 0.8454
```

```
legend(.6,.4, auc3,title="AUC")


plot(ROC_Curve,colorize=T)
plot(ROC_Curve3, colorize=T,main="ROC curve Best fit model",ylab="TPR(sensitivity)",xlab="FPR(1-specificity)",add
 = TRUE)


#==== End of script thank you ====
```



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:

Including Plots

You can also embed plots, for example: