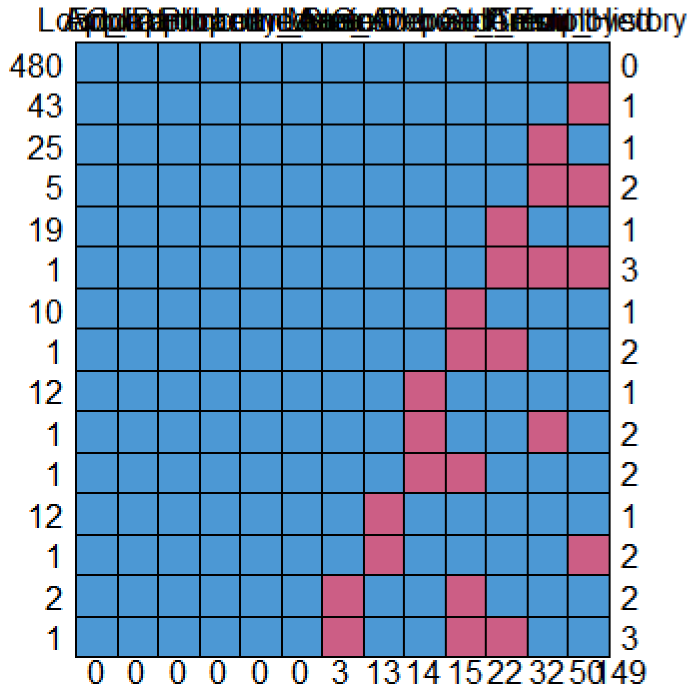
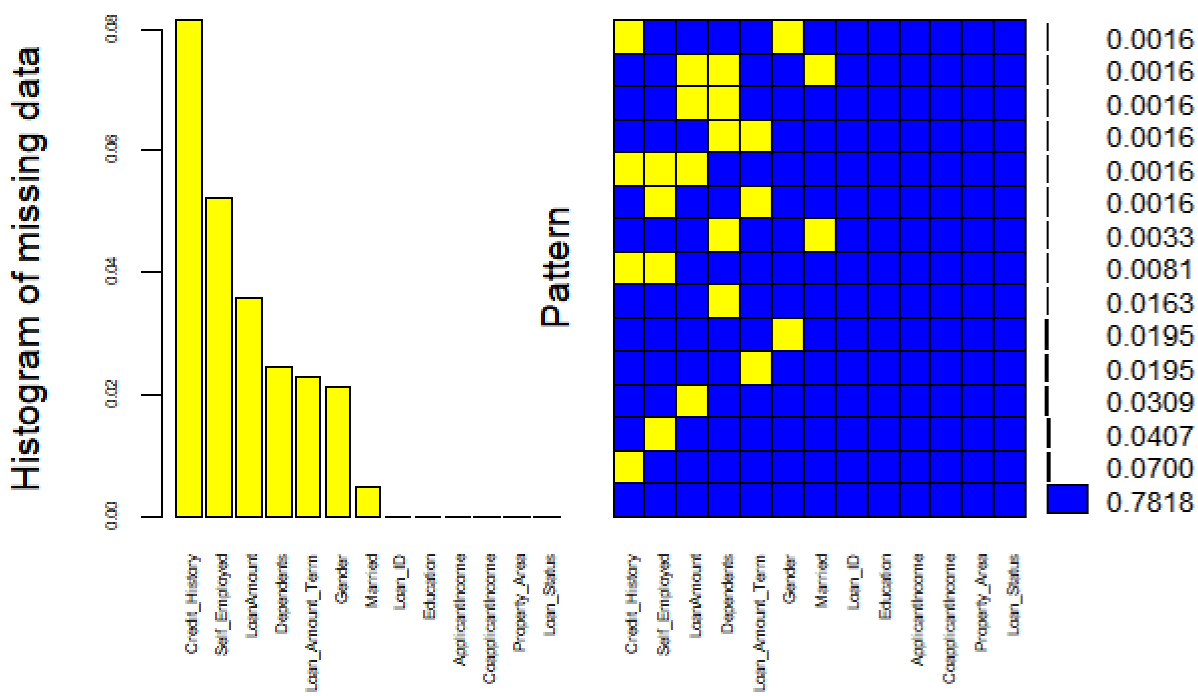
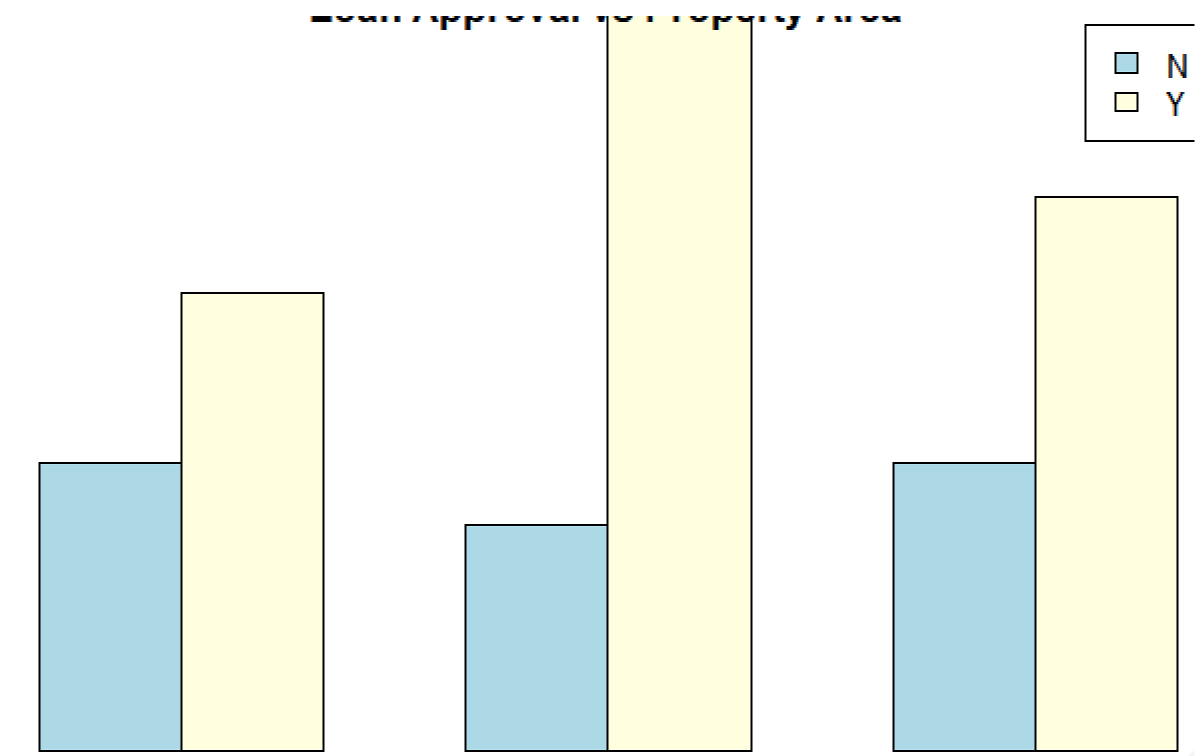
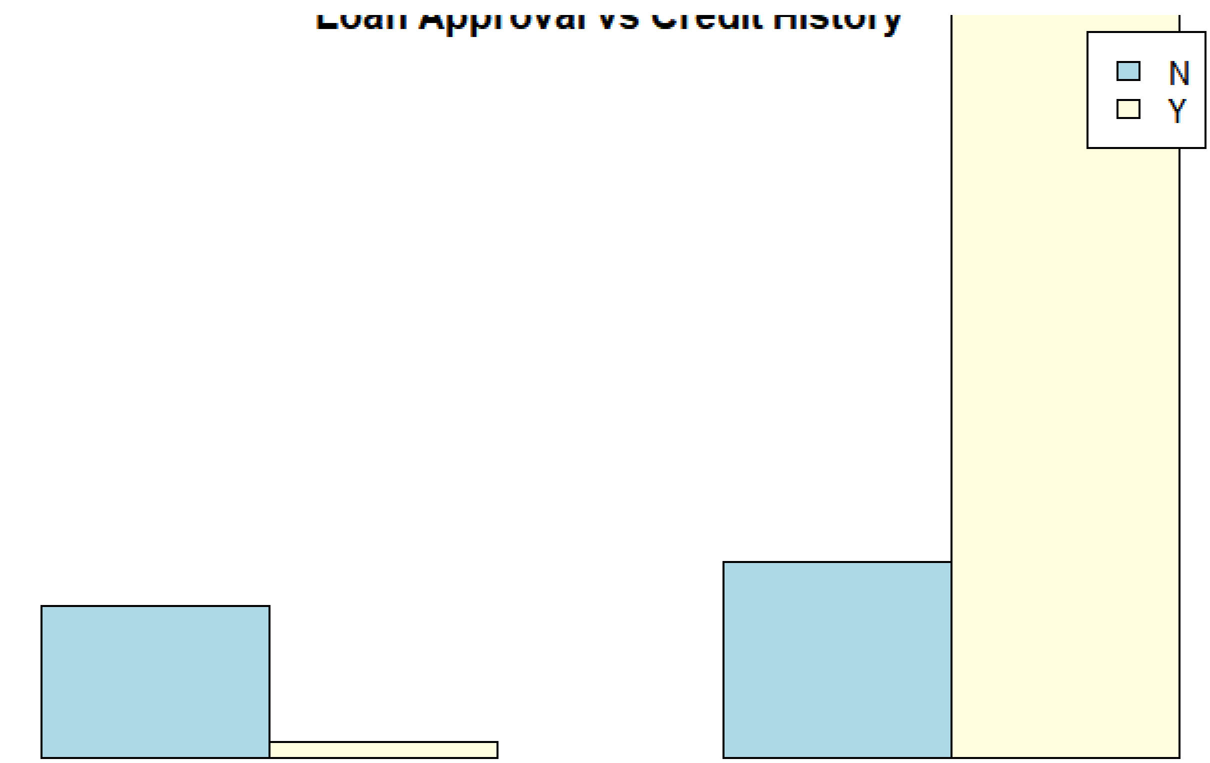
Graph Output

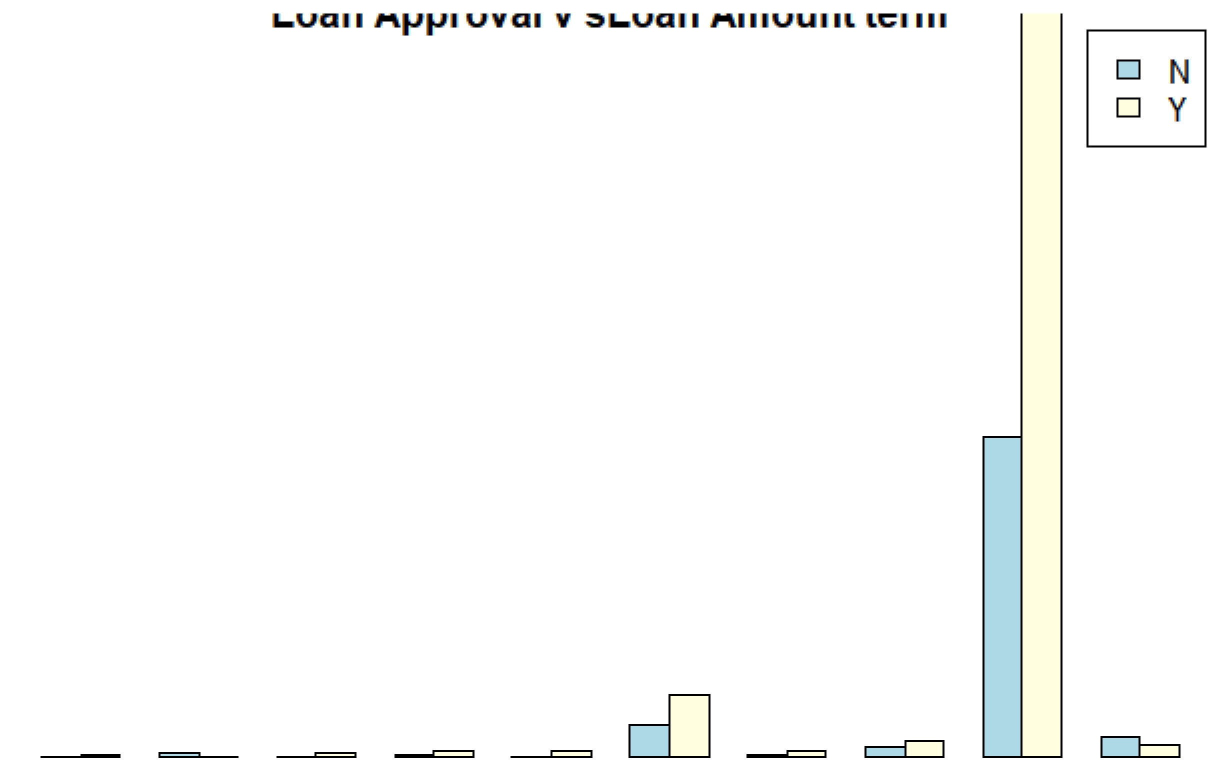


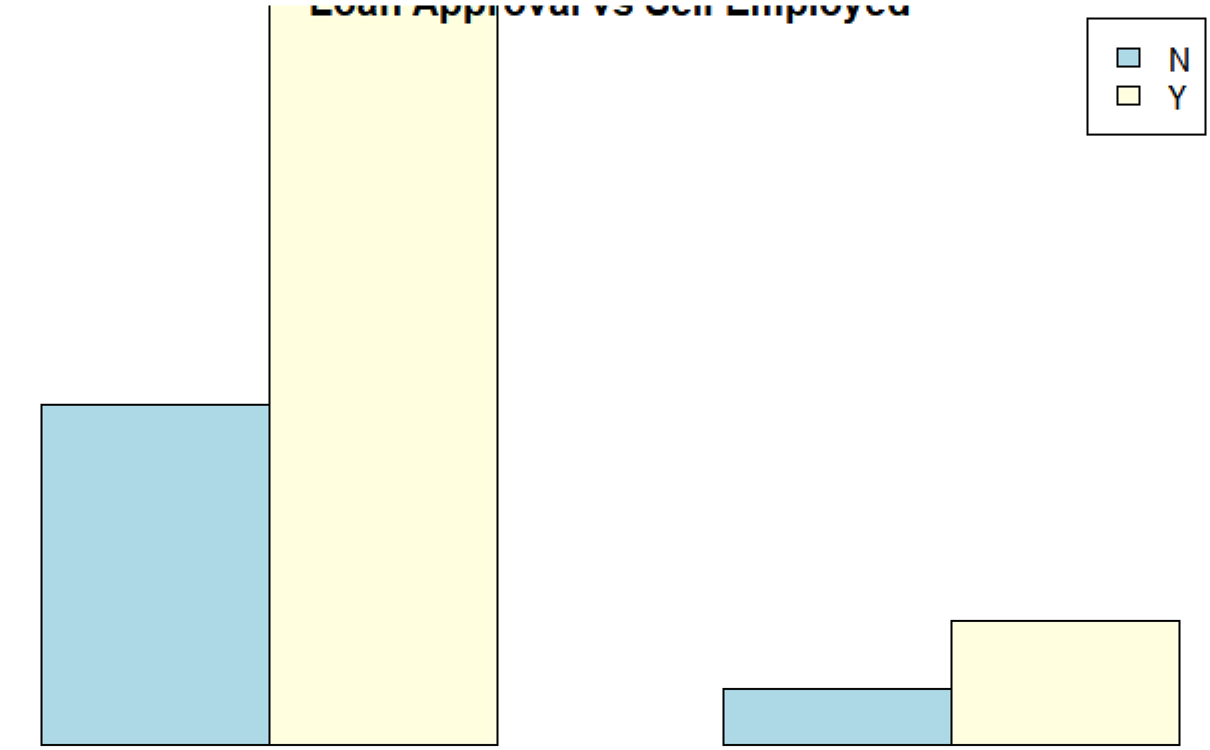


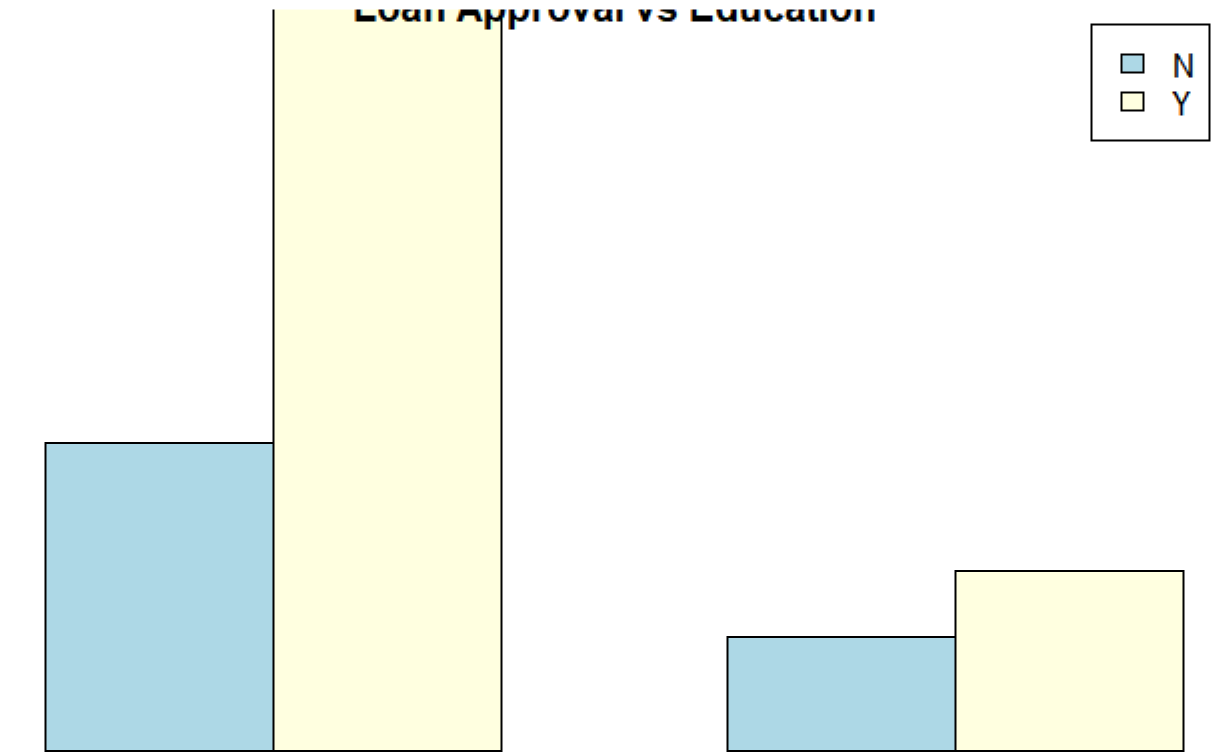


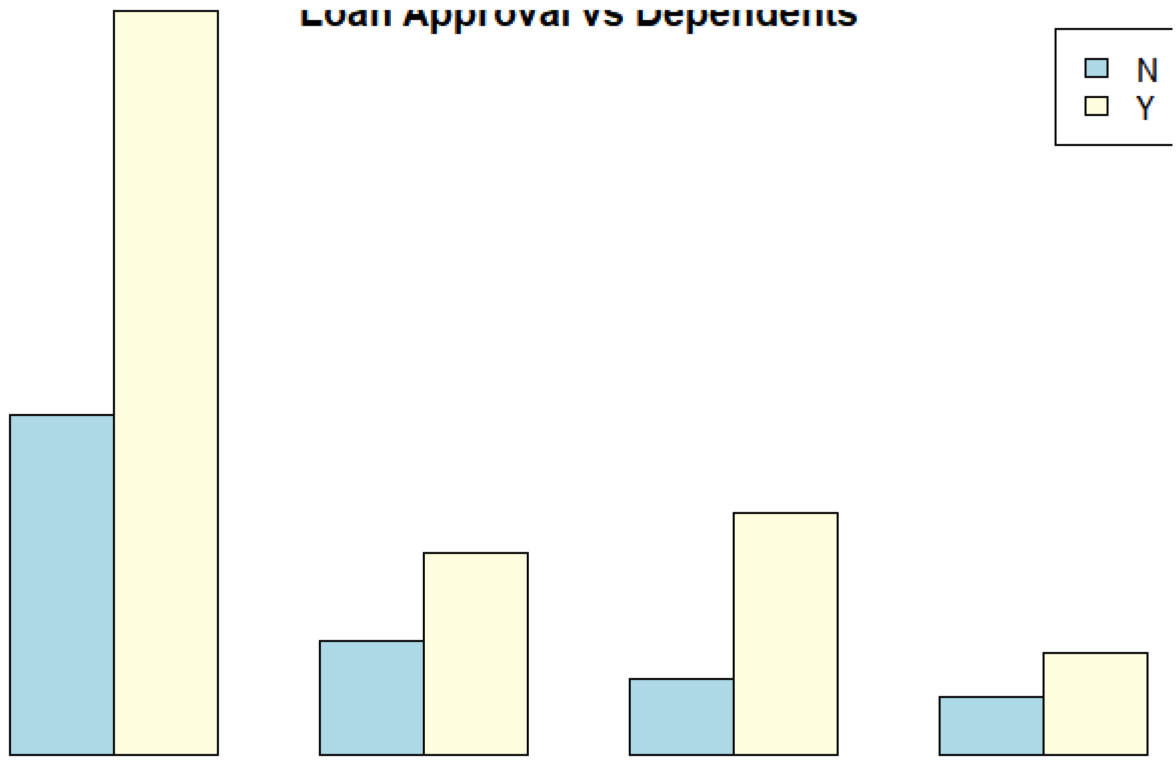
Sensitivity: Internal & Restricted

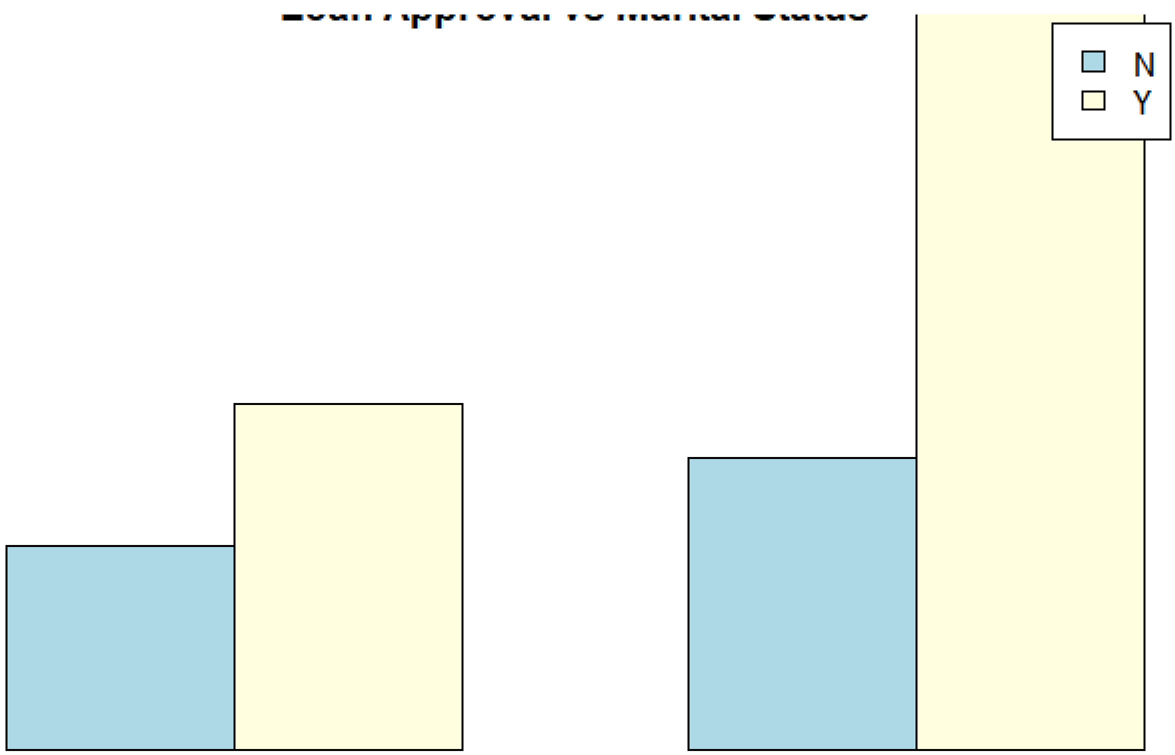

N
Y


N
Y

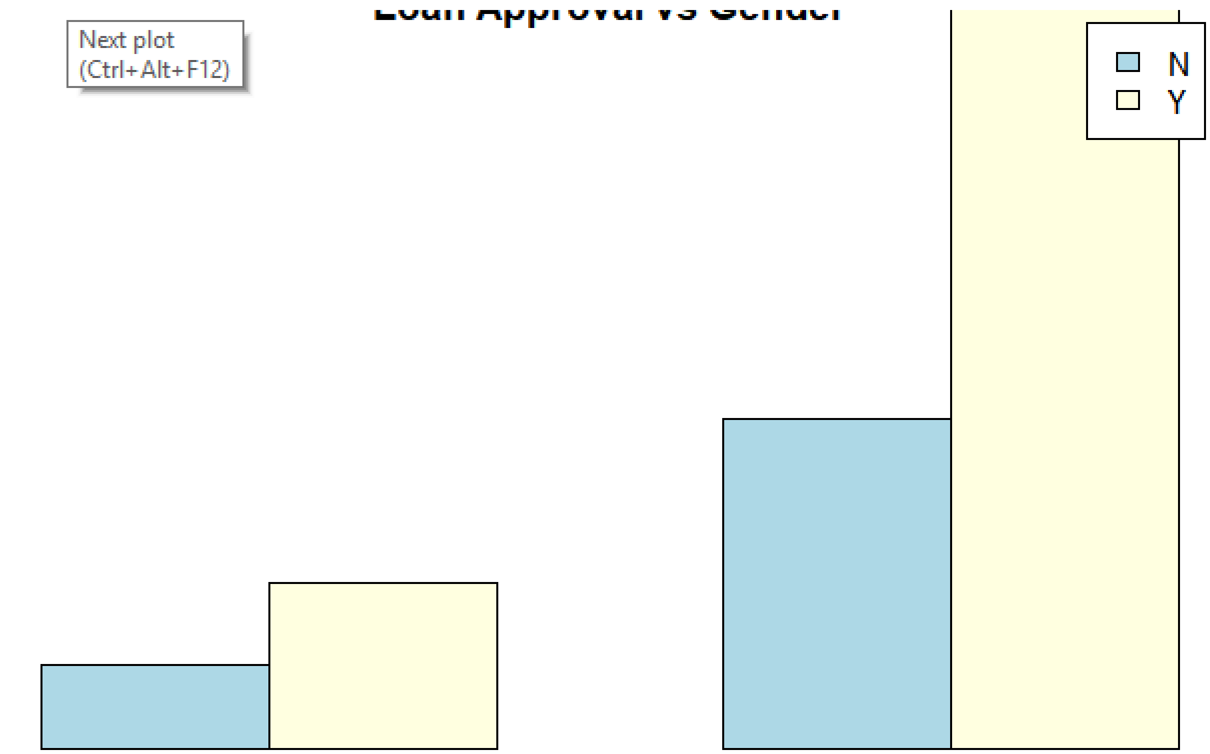

N
Y

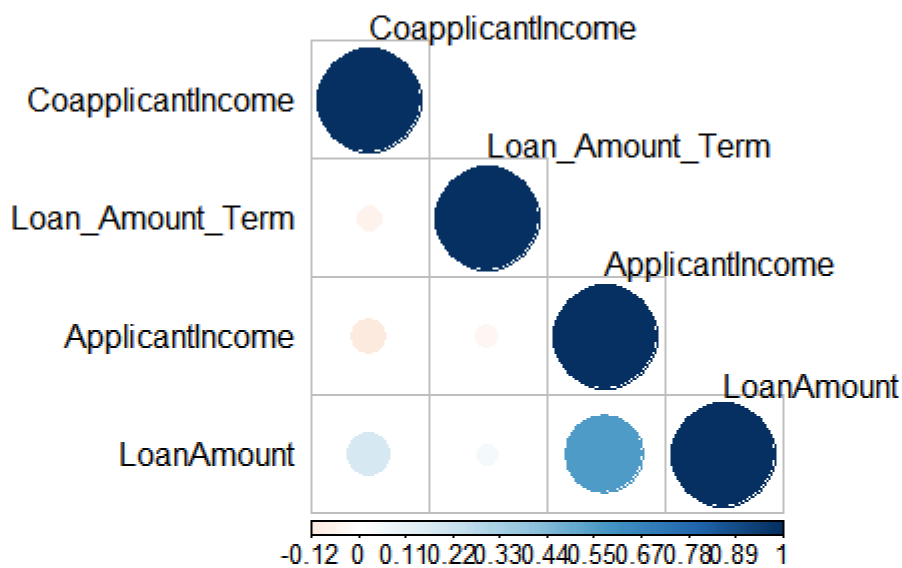

N
Y

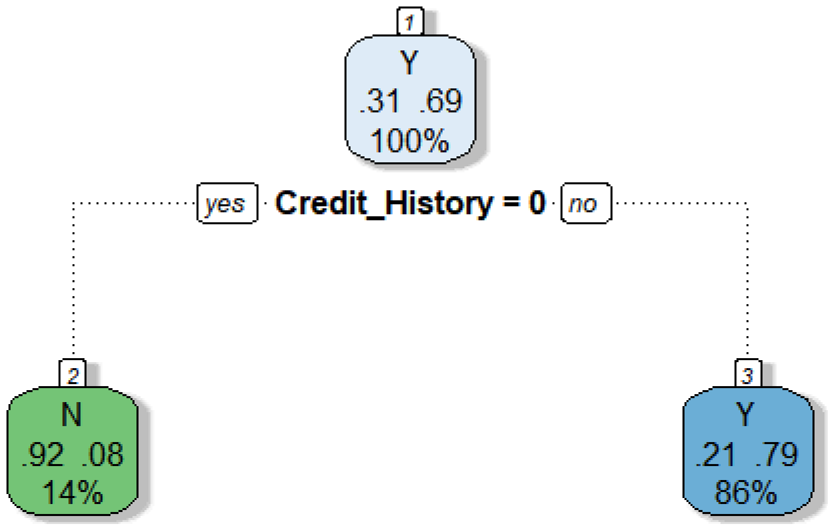


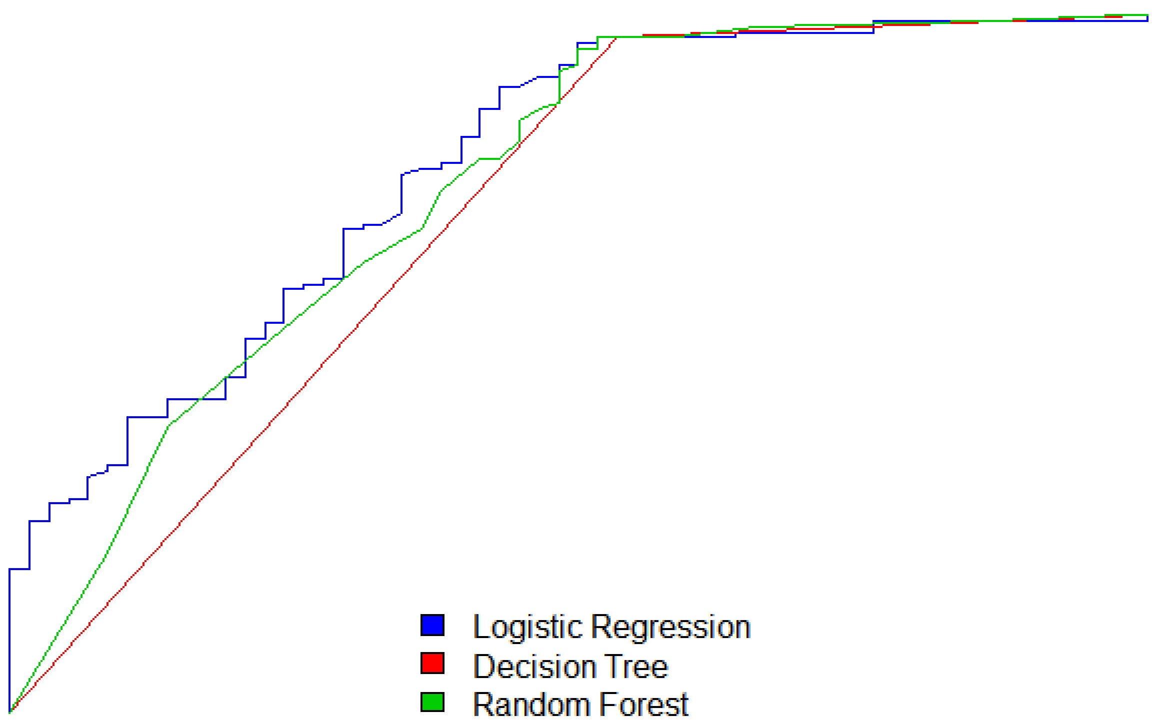


Sensitivity: Internal & Restricted


Next plot
(Ctrl+Alt+F12)
N
Y


CoapplicantIncome
Loan_Amount_Term
ApplicantIncome
LoanAmount
CoapplicantIncome
Loan_Amount_Term
ApplicantIncome
LoanAmount
-0.12 0 0.11 0.22 0.33 0.44 0.55 0.67 0.78 0.89 1





Code output

```
> set.seed(108)
> work_path = "D:/vaibhav/trend nxt/topgear/R Community/Predictive Model Base
d Logistic Regression-LoanData"
> setwd(work_path)
```

```
> # Some NAs are coded as empty strings , hence converting them to NA
> data <- read.csv("Loan_data.csv", header=T, na.strings=c(""," ","NA"))
> original_data=data
> head(data)
   Loan_ID Gender Married Dependents    Education Self_Employed ApplicantInco
me
1 LP001002   Male      No          0     Graduate            No            58
49
2 LP001003   Male     Yes          1     Graduate            No            45
83
3 LP001005   Male     Yes          0     Graduate           Yes            30
00
4 LP001006   Male     Yes          0 Not Graduate            No            25
83
5 LP001008   Male      No          0     Graduate            No            60
00
6 LP001011   Male     Yes          2     Graduate           Yes            54
17
  CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History Property_Area
1                 0         NA              360              1         Urban
2              1508        128              360              1         Rural
3                 0         66              360              1         Urban
4              2358        120              360              1         Urban
5                 0        141              360              1         Urban
6              4196        267              360              1         Urban
  Loan_Status
1           Y
2           N
3           Y
4           Y
5           Y
6           Y
> summary(data)
     Loan_ID          Gender     Married     Dependents        Education    Self_Em
ployed
 LP001002:  1   Female:112   No  :213    0   :345    Graduate    :480   No  :50
0
 LP001003:  1   Male  :489   Yes :398    1   :102    Not Graduate:134   Yes : 8
2
 LP001005:  1   NA's  : 13   NA's:  3    2   :101                       NA's: 3
2
 LP001006:  1                            3+  : 51
 LP001008:  1                            NA's: 15
 LP001011:  1
 (Other) :608
 ApplicantIncome CoapplicantIncome   LoanAmount    Loan_Amount_Term
 Min.   :  150   Min.   :    0     Min.   :  9.0   Min.   : 12
 1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.0   1st Qu.:360
 Median : 3812   Median : 1188     Median :128.0   Median :360
 Mean   : 5403   Mean   : 1621     Mean   :146.4   Mean   :342
 3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:168.0   3rd Qu.:360
 Max.   :81000   Max.   :41667     Max.   :700.0   Max.   :480
                                   NA's   :22      NA's   :14
 Credit_History     Property_Area Loan_Status
 Min.   :0.0000   Rural    :179   N:192
 1st Qu.:1.0000   Semiurban:233   Y:422
 Median :1.0000   Urban    :202
 Mean   :0.8422
 3rd Qu.:1.0000
 Max.   :1.0000
 NA's   :50
> str(data)
'data.frame':   614 obs. of  13 variables:
```

```
 $ Loan_ID          : Factor w/ 614 levels "LP001002","LP001003",..: 1 2 3 4
5 6 7 8 9 10 ...
 $ Gender           : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2
...
 $ Married          : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 2 2 2 2 2 ...
 $ Dependents       : Factor w/ 4 levels "0","1","2","3+": 1 2 1 1 1 3 1 4 3
2 ...
 $ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1 2 1
1 2 1 1 1 ...
 $ Self_Employed    : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 1 1 1 ...
 $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841
...
 $ CoapplicantIncome: num  0 1508 0 2358 0 ...
 $ LoanAmount       : int  NA 128 66 120 141 267 95 158 168 349 ...
 $ Loan_Amount_Term : int  360 360 360 360 360 360 360 360 360 360 ...
 $ Credit_History   : int  1 1 1 1 1 1 1 0 1 1 ...
 $ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3 3 3
3 2 3 2 ...
 $ Loan_Status      : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 1 2 1 ...
> nrow(data)
[1] 614
> ncol(data)
[1] 13
> any(is.na(data))
[1] TRUE
> sum(is.na(data))
[1] 149
> ### HAving a look at missing data
> md.pattern(data)
    Loan_ID Education ApplicantIncome CoapplicantIncome Property_Area Loan_St
atus
480       1         1               1                 1             1
1
43        1         1               1                 1             1
1
25        1         1               1                 1             1
1
5         1         1               1                 1             1
1
19        1         1               1                 1             1
1
1         1         1               1                 1             1
1
10        1         1               1                 1             1
1
1         1         1               1                 1             1
1
12        1         1               1                 1             1
1
1         1         1               1                 1             1
1
1         1         1               1                 1             1
1
12        1         1               1                 1             1
1
1         1         1               1                 1             1
1
2         1         1               1                 1             1
1
1         1         1               1                 1             1
1
          0         0               0                 0             0
0
    Married Gender Loan_Amount_Term Dependents LoanAmount Self_Employed
```

Sensitivity: Internal & Restricted

```
480        1      1                 1          1          1             1
43         1      1                 1          1          1             1
25         1      1                 1          1          1             0
5          1      1                 1          1          1             0
19         1      1                 1          1          0             1
1          1      1                 1          1          0             0
10         1      1                 1          0          1             1
1          1      1                 1          0          0             1
12         1      1                 0          1          1             1
1          1      1                 0          1          1             0
1          1      1                 0          0          1             1
12         1      0                 1          1          1             1
1          1      0                 1          1          1             1
2          0      1                 1          0          1             1
1          0      1                 1          0          0             1
           3     13                14         15         22            32
    Credit_History
480              1   0
43               0   1
25               1   1
5                0   2
19               1   1
1                0   3
10               1   1
1                1   2
12               1   1
1                1   2
1                1   2
12               1   1
1                0   2
2                1   2
1                1   3
                50 149
> aggr_plot <- aggr(data, col=c('Blue','Yellow'), numbers=TRUE, sortVars=TRUE
, labels=names(data), cex.axis=.5,cex.numbers=.9, gap=1, ylab=c("Histogram of
missing data","Pattern"))

 Variables sorted by number of missings:
         Variable       Count
    Credit_History 0.081433225
     Self_Employed 0.052117264
        LoanAmount 0.035830619
        Dependents 0.024429967
  Loan_Amount_Term 0.022801303
            Gender 0.021172638
           Married 0.004885993
           Loan_ID 0.000000000
         Education 0.000000000
   ApplicantIncome 0.000000000
 CoapplicantIncome 0.000000000
     Property_Area 0.000000000
       Loan_Status 0.000000000
> # dropping the ID column
> Loan_ID = data$Loan_ID
> data$Loan_ID = NULL
> data$Credit_History = factor(data$Credit_History)
> # Exploratory Analysis (Univariate)
> p1= qplot(Gender,data = data,geom="auto")
> p2 = qplot(Married, data=data,geom="auto")
> p3 = qplot(Dependents,data = data,geom="auto")
> p4 = qplot(Education,data = data,geom="auto")
> p5 = qplot(Self_Employed,data = data,geom="auto")
> p6 = qplot(Credit_History,data = data,geom="auto")
> p7 = qplot(Property_Area,data = data,geom="auto")
```

```
> p8 = qplot(Loan_Status,data = data,geom="auto")
> grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8,nrow=3,ncol=3)
> # Imputation
> NAsubset = data[c("Gender","Married","Dependents","Self_Employed","LoanAmou
nt","Loan_Amount_Term","Credit_History")]
> summary(NAsubset)
    Gender     Married    Dependents Self_Employed   LoanAmount     Loan_Amount
_Term
 Female:112   No  :213   0   :345   No  :500      Min.   :  9.0   Min.   : 12
 Male  :489   Yes :398   1   :102   Yes : 82      1st Qu.:100.0   1st Qu.:360
 NA's  : 13   NA's:  3   2   :101   NA's: 32      Median :128.0   Median :360
                         3+  : 51                 Mean   :146.4   Mean   :342
                         NA's: 15                 3rd Qu.:168.0   3rd Qu.:360
                                                  Max.   :700.0   Max.   :480
                                                  NA's   :22      NA's   :14
 Credit_History
 0   : 89
 1   :475
 NA's: 50
```

```
> set.seed(108)
> imputed=complete(mice(NAsubset))

 iter imp variable
  1   1  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  1   2  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  1   3  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  1   4  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  1   5  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  2   1  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  2   2  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  2   3  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  2   4  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  2   5  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  3   1  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  3   2  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  3   3  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  3   4  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  3   5  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  4   1  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  4   2  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  4   3  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
```

Sensitivity: Internal & Restricted

```
  4   4  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  4   5  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  5   1  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  5   2  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  5   3  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  5   4  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
  5   5  Gender  Married  Dependents  Self_Employed  LoanAmount  Loan_Amount_
Term  Credit_History
> data$Gender=imputed$Gender
> data$Married=imputed$Married
> data$Dependents=imputed$Dependents
> data$Self_Employed=imputed$Self_Employed
> data$LoanAmount=imputed$LoanAmount
> data$Loan_Amount_Term=imputed$Loan_Amount_Term
> data$Credit_History=imputed$Credit_History
> summary(data)
    Gender     Married    Dependents        Education     Self_Employed Applicant
Income
 Female:116   No :213    0 :355     Graduate     :480    No :527        Min.    :
150
 Male  :498   Yes:401    1 :103     Not Graduate:134    Yes: 87        1st Qu.:
2878
                         2 :104                                        Median :
3812
                         3+: 52                                        Mean    :
5403
                                                                       3rd Qu.:
5795
                                                                       Max.    :8
1000
 CoapplicantIncome   LoanAmount    Loan_Amount_Term Credit_History   Property
_Area
 Min.   :    0     Min.   :  9.0   Min.   : 12      0: 93          Rural    :
179
 1st Qu.:    0     1st Qu.:100.0   1st Qu.:360      1:521          Semiurban:
233
 Median : 1188     Median :128.0   Median :360                     Urban    :
202
 Mean   : 1621     Mean   :146.9   Mean   :342
 3rd Qu.: 2297     3rd Qu.:168.0   3rd Qu.:360
 Max.   :41667     Max.   :700.0   Max.   :480
 Loan_Status
 N:192
 Y:422
```

```
> plottable1=table(data$Loan_Status,data$Property_Area)
> barplot(plottable1, main="Loan Approval vs Property Area", xlab="Property A
rea",col=c("LightBlue","LightYellow"),legend=rownames(plottable1),beside = TR
UE)
> plottable2=table(data$Loan_Status,data$Credit_History)
> barplot(plottable2, main="Loan Approval vs Credit History", xlab="Credit Hi
story",col=c("LightBlue","LightYellow"),legend=rownames(plottable2),beside =
TRUE)
> plottable3=table(data$Loan_Status,data$Loan_Amount_Term)
```

Sensitivity: Internal & Restricted

```
> barplot(plottable3, main="Loan Approval v sLoan Amount term", xlab="Loan am
ount term",col=c("LightBlue","LightYellow"),legend=rownames(plottable3),besid
e = TRUE)
> plottable4=table(data$Loan_Status,data$Self_Employed)
> barplot(plottable4, main="Loan Approval vs Self Employed", xlab="Self_Emplo
yed",col=c("LightBlue","LightYellow"),legend=rownames(plottable4),beside = TR
UE)
> plottable5=table(data$Loan_Status,data$Education)
> barplot(plottable5, main="Loan Approval vs Education", xlab="Education",col
=c("LightBlue","LightYellow"),legend=rownames(plottable5),beside = TRUE)
> plottable6=table(data$Loan_Status,data$Dependents)
> barplot(plottable6, main="Loan Approval vs Dependents", xlab="Dependents",c
ol=c("LightBlue","LightYellow"),legend=rownames(plottable6),beside = TRUE)
> plottable7=table(data$Loan_Status,data$Married)
> barplot(plottable7, main="Loan Approval vs Marital Status", xlab="Marriage"
,col=c("LightBlue","LightYellow"),legend=rownames(plottable7),beside = TRUE)
> plottable8=table(data$Loan_Status,data$Gender)
> barplot(plottable8, main="Loan Approval vs Gender", xlab="Gender",col=c("Li
ghtBlue","LightYellow"),legend=rownames(plottable8),beside = TRUE)
> # Correlation Analysis
> numeric_features= data[c("ApplicantIncome","CoapplicantIncome","LoanAmount"
,"Loan_Amount_Term")]
> corTable=cor(numeric_features)
> corTable
                  ApplicantIncome CoapplicantIncome LoanAmount Loan_Amount_Te
rm
ApplicantIncome        1.00000000       -0.11660458 0.56421656       -0.048646
77
CoapplicantIncome     -0.11660458        1.00000000 0.17931270       -0.064713
35
LoanAmount             0.56421656        0.17931270 1.00000000        0.040639
87
Loan_Amount_Term      -0.04864677       -0.06471335 0.04063987        1.000000
00
> corrplot( cor(as.matrix(numeric_features), method = "pearson", use = "compl
ete.obs") ,is.corr = FALSE, type = "lower", order = "hclust", tl.col = "black
", tl.srt = 360)
> # applicant income and loan amount correlated
> # Feature Engineering
> # Add a new feature has a coapplicant
> coAppIn=data$CoapplicantIncome
> for(i in data$CoapplicantIncome){
+   data$CoapplicantIncome[data$CoapplicantIncome!=0.00] = 1.00
+ }
> data$Coapplicant= as.factor(data$CoapplicantIncome)
> # data$CoapplicantIncome= coAppIn
> ## Training & Testing Set
> set.seed(108)
> split=sample.split(data$Loan_Status,SplitRatio = .7)
> train=subset(data,split==T)
> test=subset(data,split==F)
> ## Model Building & CV Using GLM
> Status=glm(Loan_Status~Married+LoanAmount+Credit_History+Property_Area,data
=train,family="binomial")
> predGlm=predict(Status,type="response",newdata=test)
> summary(Status)

Call:
glm(formula = Loan_Status ~ Married + LoanAmount + Credit_History +
    Property_Area, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1290  -0.3714   0.5506   0.7211   2.3792
```

```
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)            -3.042930   0.573079  -5.310 1.10e-07 ***
MarriedYes              0.630318   0.261820   2.407  0.01606 *  
LoanAmount             -0.002024   0.001290  -1.570  0.11652    
Credit_History1         3.853596   0.492311   7.828 4.97e-15 ***
Property_AreaSemiurban  0.857523   0.312383   2.745  0.00605 ** 
Property_AreaUrban      0.429623   0.303043   1.418  0.15628    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 532.79  on 428  degrees of freedom
Residual deviance: 396.57  on 423  degrees of freedom
AIC: 408.57

Number of Fisher Scoring iterations: 5

> # Computing accuracy
> table(test$Loan_Status,predGlm>.5)
   
    FALSE TRUE
  N    27   31
  Y     4  123
> (27+123)/(27+123+31+4)
[1] 0.8108108
> set.seed(108)
> # Decision Tree Model
> numFolds = trainControl( method = "cv", number = 10 )
> cpGrid = expand.grid( .cp = seq(0.01,0.5,0.01))
> train(Loan_Status~Married+LoanAmount+Credit_History+Property_Area,data=trai
n,method="rpart",trControl=numFolds,tuneGrid=cpGrid)
CART 

429 samples
  4 predictor
  2 classes: 'N', 'Y' 

No pre-processing
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 386, 386, 385, 386, 386, 387, ... 
Resampling results across tuning parameters:

  cp    Accuracy   Kappa    
  0.01  0.8088719  0.4785112
  0.02  0.8088719  0.4785112
  0.03  0.8088719  0.4785112
  0.04  0.8088719  0.4785112
  0.05  0.8088719  0.4785112
  0.06  0.8088719  0.4785112
  0.07  0.8088719  0.4785112
  0.08  0.8088719  0.4785112
  0.09  0.8088719  0.4785112
  0.10  0.8088719  0.4785112
  0.11  0.8088719  0.4785112
  0.12  0.8088719  0.4785112
  0.13  0.8088719  0.4785112
  0.14  0.8088719  0.4785112
  0.15  0.8088719  0.4785112
  0.16  0.8088719  0.4785112
  0.17  0.8088719  0.4785112
  0.18  0.8088719  0.4785112
```

Sensitivity: Internal & Restricted

```
  0.19  0.8088719  0.4785112
  0.20  0.8088719  0.4785112
  0.21  0.8088719  0.4785112
  0.22  0.8088719  0.4785112
  0.23  0.8088719  0.4785112
  0.24  0.8088719  0.4785112
  0.25  0.8088719  0.4785112
  0.26  0.8088719  0.4785112
  0.27  0.8088719  0.4785112
  0.28  0.8088719  0.4785112
  0.29  0.8088719  0.4785112
  0.30  0.8088719  0.4785112
  0.31  0.8088719  0.4785112
  0.32  0.8088719  0.4785112
  0.33  0.8088719  0.4785112
  0.34  0.8088719  0.4785112
  0.35  0.8088719  0.4785112
  0.36  0.8088719  0.4785112
  0.37  0.8088719  0.4785112
  0.38  0.7929629  0.4208189
  0.39  0.7293416  0.1653267
  0.40  0.6876925  0.0000000
  0.41  0.6876925  0.0000000
  0.42  0.6876925  0.0000000
  0.43  0.6876925  0.0000000
  0.44  0.6876925  0.0000000
  0.45  0.6876925  0.0000000
  0.46  0.6876925  0.0000000
  0.47  0.6876925  0.0000000
  0.48  0.6876925  0.0000000
  0.49  0.6876925  0.0000000
  0.50  0.6876925  0.0000000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.37.
> decisionTreeModel=rpart(Loan_Status~Married+LoanAmount+Credit_History+Prope
rty_Area,data=train,method="class",cp=.37)
> rpart.plot(decisionTreeModel,extra=104, box.palette="GnBu",branch.lty=3, sh
adow.col="gray", nn=TRUE)
> ## CV with rpart
> predDT=predict(decisionTreeModel,newdata = test,type = "class")
> table(test$Loan_Status,predDT)
   predDT
      N   Y
  N  27  31
  Y   4 123
> # Accuracy
> (27+123)/(27+123+31+4)
[1] 0.8108108
> # RF Model
> set.seed(108)
> randomForestModel=randomForest(Loan_Status~Married+LoanAmount+Credit_Histor
y+Property_Area,data=train,ntree=50,nodesize=10)
> predictRF=predict(randomForestModel,newdata=test)
> table(test$Loan_Status,predictRF)
   predictRF
      N   Y
  N  28  30
  Y   4 123
> # Accuracy
> (29+122)/(29+122+29+5)
[1] 0.8162162
> # AUC Calculation
> glm_ROC=predict(Status,test,type="response")
```

```
> pred_glm=prediction(glm_ROC,test$Loan_Status)
> perf_glm=performance(pred_glm,"tpr","fpr")
> dt_ROC=predict(decisionTreeModel,test)
> pred_dt=prediction(dt_ROC[,2],test$Loan_Status)
> perf_dt=performance(pred_dt,"tpr","fpr")
> RF_ROC=predict(randomForestModel,test,type="prob")
> pred_RF=prediction(RF_ROC[,2],test$Loan_Status)
> perf_RF=performance(pred_RF,"tpr","fpr")
> auc_glm <- performance(pred_glm,"auc")
> auc_glm <- round(as.numeric(auc_glm@y.values),3)
> auc_dt <- performance(pred_dt,"auc")
> auc_dt <- round(as.numeric(auc_dt@y.values),3)
> auc_RF <- performance(pred_RF,"auc")
> auc_RF <- round(as.numeric(auc_RF@y.values),3)
> print(paste('AUC of Logistic Regression:',auc_glm))
[1] "AUC of Logistic Regression: 0.787"
> print(paste('AUC of Decision Tree:',auc_dt))
[1] "AUC of Decision Tree: 0.717"
> print(paste('AUC of Random Forest:',auc_RF))
[1] "AUC of Random Forest: 0.757"
> # ROC Curves
> plot(perf_glm, main = "ROC curves for the models", col='blue')
> plot(perf_dt,add=TRUE, col='red')
> plot(perf_RF, add=TRUE, col='green3')
> legend('bottom', c("Logistic Regression", "Decision Tree", "Random Forest")
, fill = c('blue','red','green3'), bty='n')
>
```