“Loan Prediction Status of Customers”

This is to find ***Loan Defaulter***. Company wants to give loan to reliable customers so they have to classify customers based on the historical data.

Step 1. Import Data :- Reading CSV File in R

```
train_loan_data<-read.csv(file.choose())
test_loan_data<-read.csv(file.choose())
```

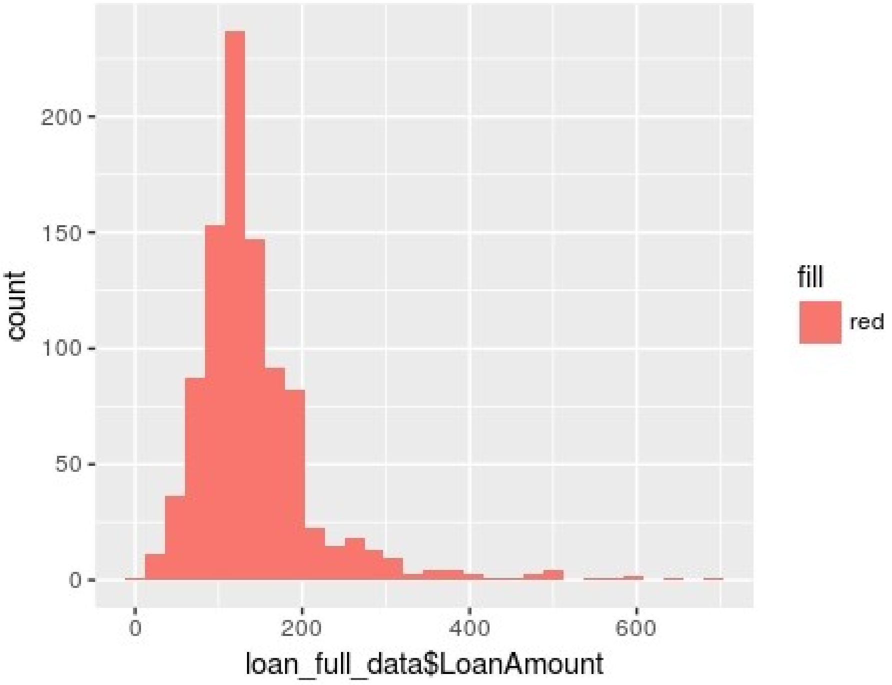
Step 2. EDA(Exploratory Data Analysis) :-

```
# Extraction of feature names in dataset
names(train_loan_data)
names(test_loan_data)

# Find dimensions of datasets
dim(train_loan_data)
dim(test_loan_data)

# Target variable analysis
barplot(table(train_loan_data$Loan_Status))

#combine both train and test data for analysis and preprocessing
test_loan_data$Loan_Status=NA
test_loan_data$IS_trainset=FALSE
train_loan_data$IS_trainset=TRUE
loan_full_data=rbind(train_loan_data,test_loan_data)
dim(loan_full_data)
head(loan_full_data)

# summary of all the features
summary(train_loan_data)
summary(test_loan_data)
summary(loan_full_data)
```

Here we saw which featuers having NA values. We have to clean this data.

Step 3. Data Cleaning or preprocessing :-

(1).Loan Amount having missing values so first plot histogram of this

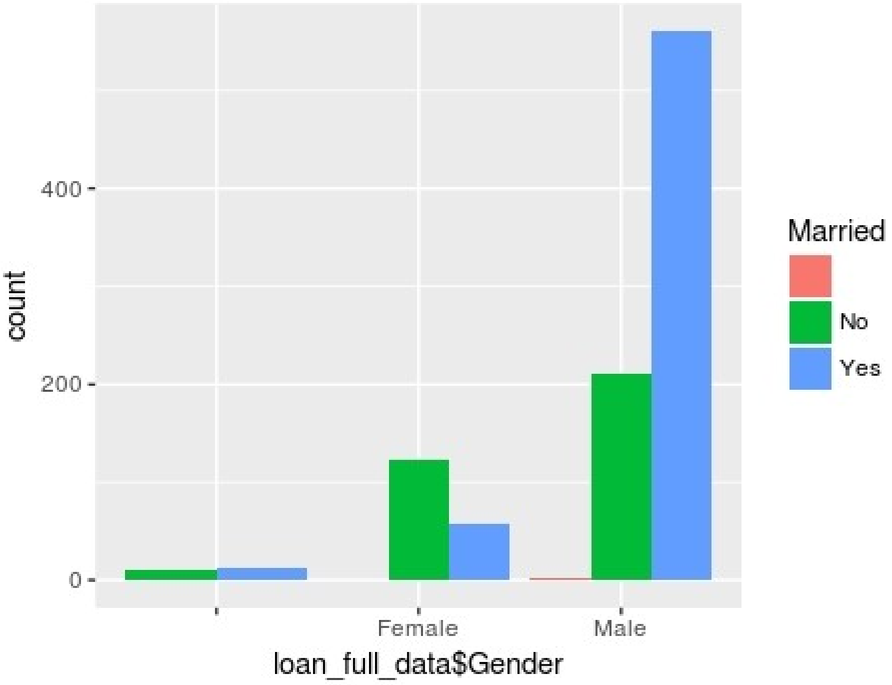


See in the picture, it is following normal distributuon so we will replace values with mean.

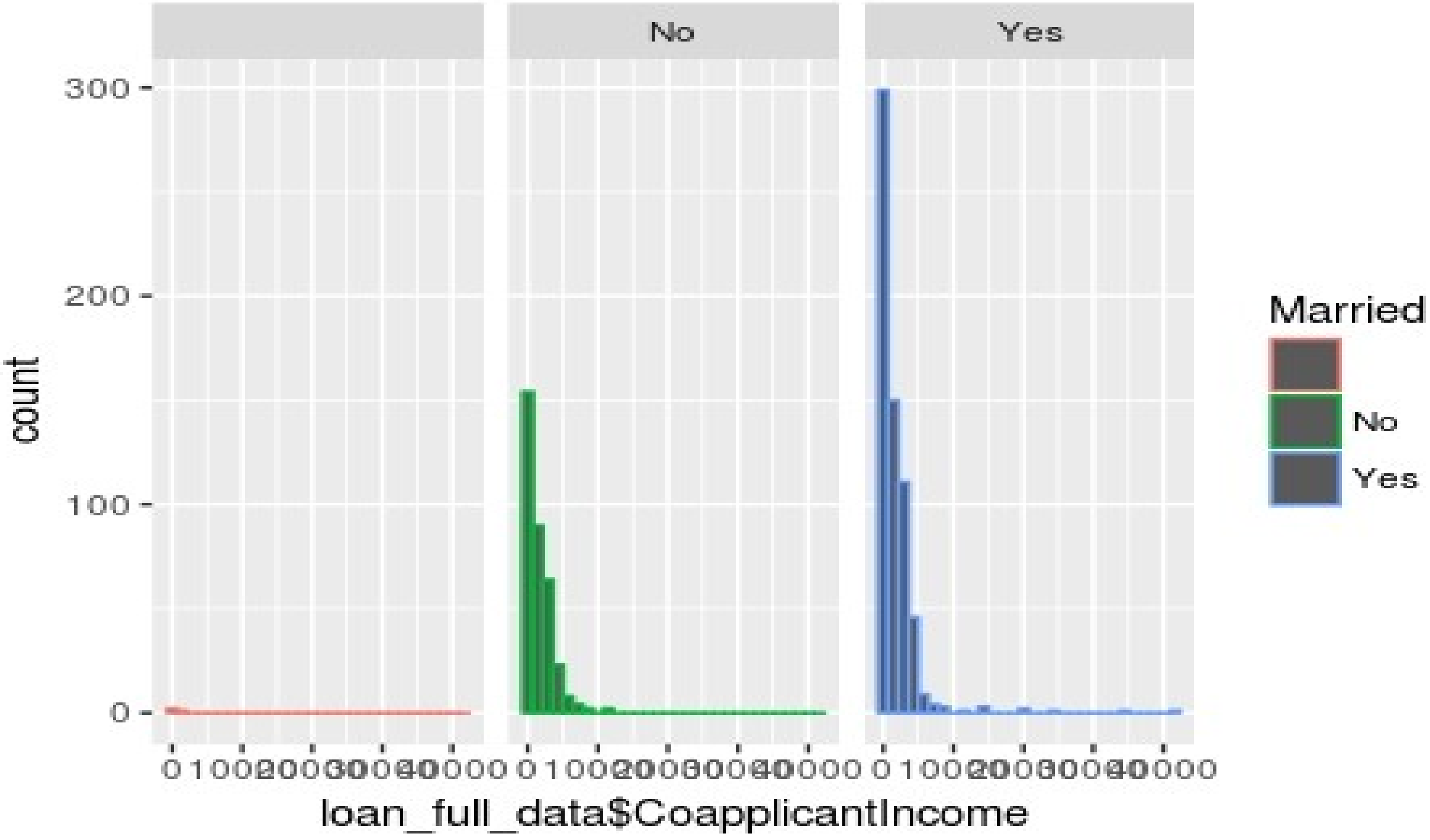
loan_full_data[is.na(loan_full_data$LoanAmount),"LoanAmount"]<-mean(loan_full_data$LoanAmount,na.rm=TRUE)

(2) Gender having missing values plot the graph. It is showing mode pattern in graph more number of male customers so we will replace NA values with mode which is MALE

loan_full_data[loan_full_data$Gender=='',"Gender"]<-'Male'

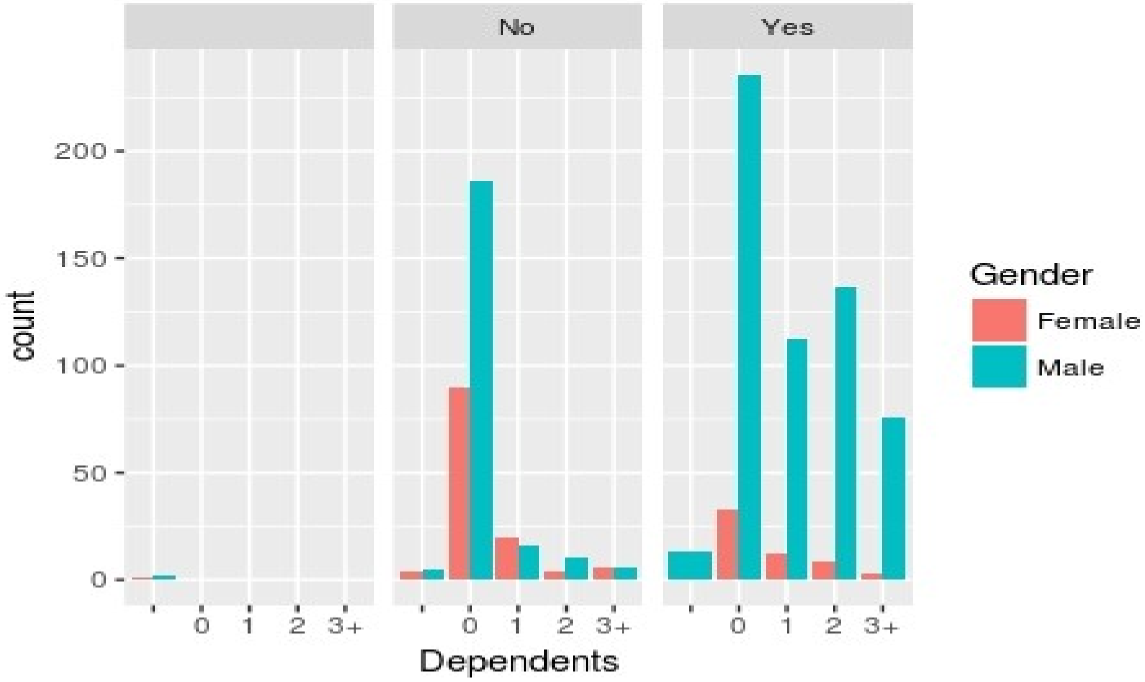


(3).Married also having NA values. In a graph we have seen which is having coapplicantIncome less is not married so we will fill NA values like this.



```
loan_full_data$Married[is.na(loan_full_data$Married) &
loan_full_data$CoapplicantIncome==0]<-"No"
loan_full_data$Married[is.na(loan_full_data$Married)]<- "Yes"
```

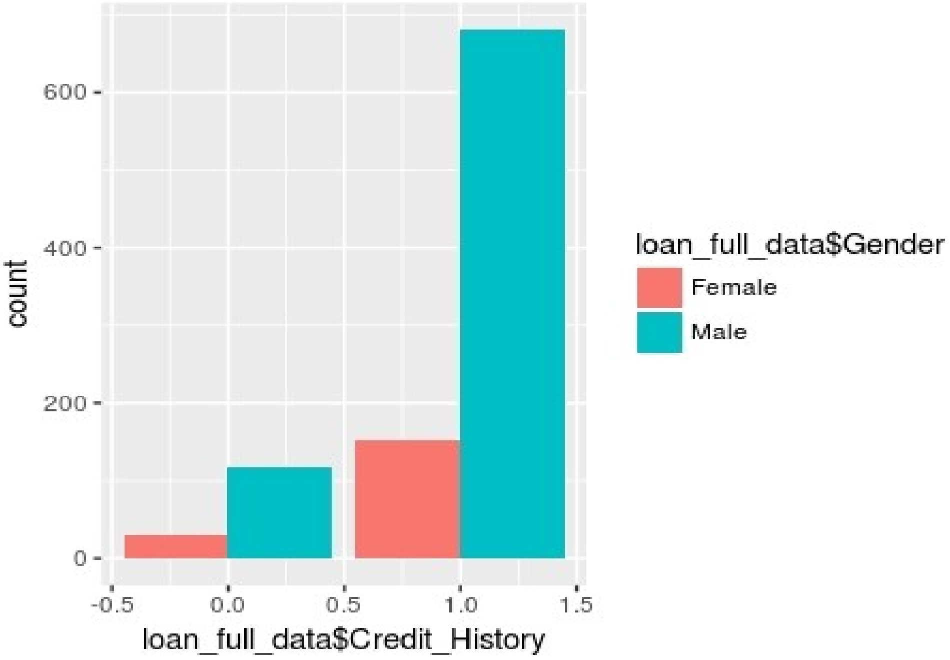
(4).Dependents having missing values so see the plot



so this is also showing most of the members having 0 dependents so we will fill mode of this.

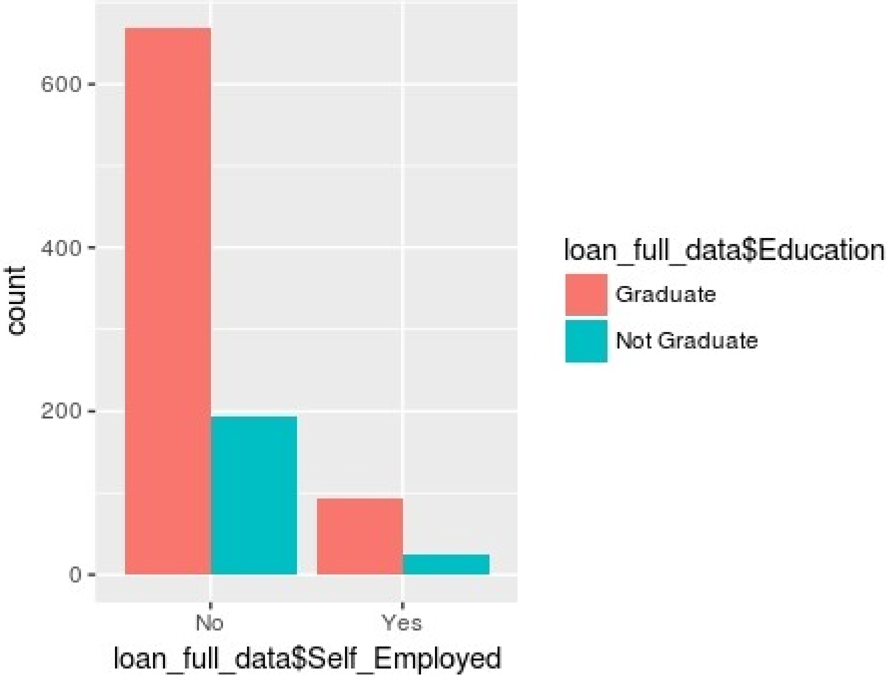
loan_full_data[is.na(loan_full_data$Dependents),"Dependents"]<-0

(5).Credit History also having NA values



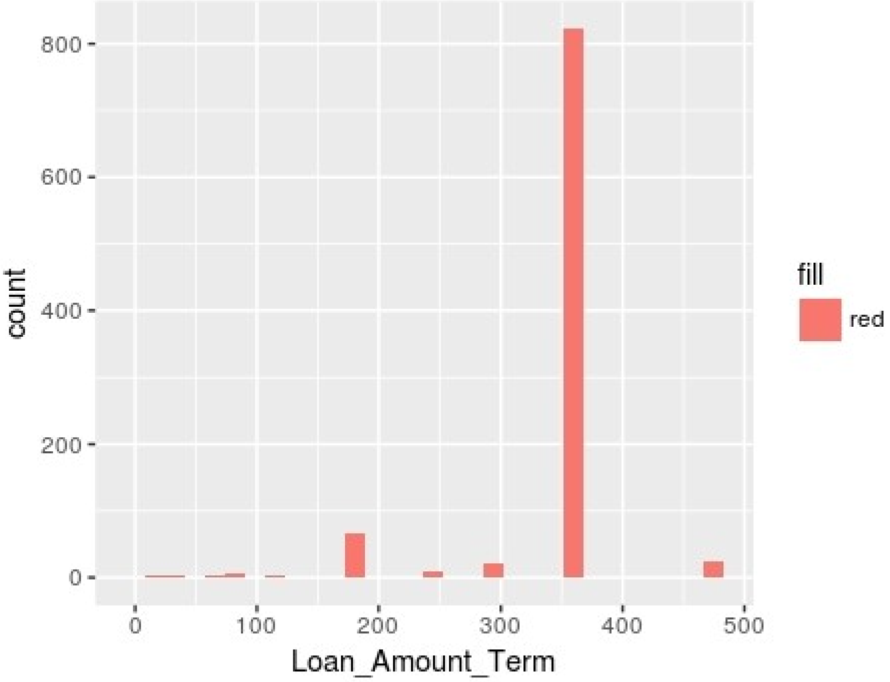
so it is also following mode pattern we will go with mode value and put na=1
loan_full_data[is.na(loan_full_data$Credit_History),"Credit_History"]<-1

(6).Self Employed having NA values



In this also we go with mode value
loan_full_data[loan_full_data$Self_Employed=='"",Self_Employed"]<-'No'

(7).Loan Amount term having NA values



loan amount term following mode pattern so we will replace this with mode value.
loan_full_data[is.na(loan_full_data$Loan_Amount_Term),"Loan_Amount_Term"]<-360

Step 4. Feature Extraction and Dummy Variables:-

we calculate two new feature variables :-
Total Income = Applicant Income + Coapplicant Income
and Loan Amount by Total Income = Loan Amount/Total Income

```
loan_full_data<mutate(loan_full_data,TotalIncome=ApplicantIncome+CoapplicantIncome)
loan_full_data$LoanAmountByTotIncome<-loan_full_data$LoanAmount/loan_full_data$TotalIncome
```

Step 5. Model Development and validation:-

```
# Divide training and test data set in a ratio of 70:30
train_mod=loan.train[1:429,]
test_mod=loan.train[429:614,]
```

1. Rpart Algorithm:-

```
Rec_model <- rpart(Loan_Status~., train_mod, method = "class")
pred <- predict(Rec_model,test_mod, type = "class")

##validation to testdata
conf <- table(test_mod$Loan_Status, pred)
print(conf)
accuracy <- sum(diag(conf))/sum(conf)
print(accuracy)
```

Accuracy we got in case of Rpart = 0.83 or 83%

2. Decision Tree Algorithm:-

```
library("party")
Dtree_model=ctree(train_mod$Loan_Status~.,data = train_mod)
plot(Dtree_model)

#Validation
test_mod$prid=predict(Dtree_model,test_mod)
score=prediction(as.numeric(test_mod$prid),as.numeric(test_mod$Loan_Status))
performance(score,"auc")
plot(performance(score,"tpr","fpr"),col="green")
```

Accuracy we got in case of Decision Tree = 0.73 or 73%

3. Logistic Regression model :-

```
model <- glm(Loan_Status~.,family=binomial(link='logit'),data=train_mod)
summary(model)
anova(model, test="Chisq")
prid=predict(model,test_mod)
install.packages("ROCR")
library("ROCR")
score=prediction(as.numeric(prid),as.numeric(test_mod$Loan_Status))
performance(score,"auc")
plot(performance(score,"tpr","fpr"),col="green")
```

Accuracy we got in case of Logistic regression = 0.75 ot 75%

Based on 3 models we conclude Rpart algorithm works good in this case and giving accuracy of 83 %.