Loan Prediction

Rohit Dixit

## Problem Statement

Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. Here they have provided a partial data set.

## Data Glimpse

**Variable** **Description**  
Loan\_ID -Unique Loan ID  
Gender -Male/ Female  
Married -Applicant married (Y/N)  
Dependents -Number of dependents  
Education -Applicant Education (Graduate/ Under Graduate)  
Self\_Employed -Self employed (Y/N)  
ApplicantIncome -Applicant income  
CoapplicantIncome -Coapplicant income  
LoanAmount -Loan amount in thousands  
Loan\_Amount\_Term -Term of loan in months  
Credit\_History -credit history meets guidelines  
Property\_Area -Urban/ Semi Urban/ Rural  
Loan\_Status Loan -approved (Y/N)

## R Code

**Importing Library**

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: replacing previous import by 'plyr::ddply' when loading 'caret'

## Warning: replacing previous import by 'rlang::expr' when loading 'recipes'

## Warning: replacing previous import by 'rlang::f\_lhs' when loading 'recipes'

## Warning: replacing previous import by 'rlang::is\_empty' when loading  
## 'recipes'

## Warning: replacing previous import by 'rlang::names2' when loading  
## 'recipes'

## Warning: replacing previous import by 'rlang::quos' when loading 'recipes'

library(mlbench)  
library(ggplot2)  
library(ggthemes)  
library(plyr)  
library(RANN)  
library(gridExtra)

**Data Importing**

#data<-read.csv(url('https://datahack-prod.s3.ap-south-1.amazonaws.com/train\_file/train\_u6lujuX\_CVtuZ9i.csv'))  
train\_data <-read.csv("train.csv")  
test\_data <-read.csv("test.csv")

**Data Exploration**

summary(train\_data)

## Loan\_ID Gender Married Dependents Education   
## LP001002: 1 : 13 : 3 : 15 Graduate :480   
## LP001003: 1 Female:112 No :213 0 :345 Not Graduate:134   
## LP001005: 1 Male :489 Yes:398 1 :102   
## LP001006: 1 2 :101   
## LP001008: 1 3+: 51   
## LP001011: 1   
## (Other) :608   
## Self\_Employed ApplicantIncome CoapplicantIncome LoanAmount   
## : 32 Min. : 150 Min. : 0 Min. : 9.0   
## No :500 1st Qu.: 2878 1st Qu.: 0 1st Qu.:100.0   
## Yes: 82 Median : 3812 Median : 1188 Median :128.0   
## Mean : 5403 Mean : 1621 Mean :146.4   
## 3rd Qu.: 5795 3rd Qu.: 2297 3rd Qu.:168.0   
## Max. :81000 Max. :41667 Max. :700.0   
## NA's :22   
## Loan\_Amount\_Term Credit\_History Property\_Area Loan\_Status  
## Min. : 12 Min. :0.0000 Rural :179 N:192   
## 1st Qu.:360 1st Qu.:1.0000 Semiurban:233 Y:422   
## Median :360 Median :1.0000 Urban :202   
## Mean :342 Mean :0.8422   
## 3rd Qu.:360 3rd Qu.:1.0000   
## Max. :480 Max. :1.0000   
## NA's :14 NA's :50

summary(test\_data)

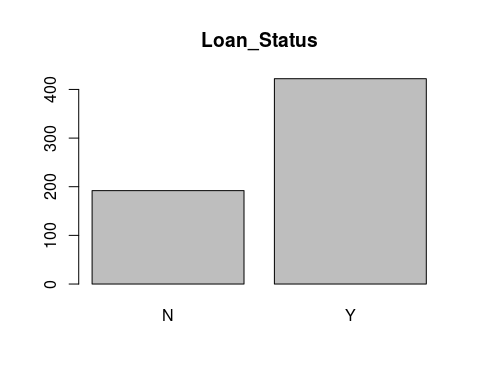
## Loan\_ID Gender Married Dependents Education   
## LP001015: 1 : 11 No :134 : 10 Graduate :283   
## LP001022: 1 Female: 70 Yes:233 0 :200 Not Graduate: 84   
## LP001031: 1 Male :286 1 : 58   
## LP001035: 1 2 : 59   
## LP001051: 1 3+: 40   
## LP001054: 1   
## (Other) :361   
## Self\_Employed ApplicantIncome CoapplicantIncome LoanAmount   
## : 23 Min. : 0 Min. : 0 Min. : 28.0   
## No :307 1st Qu.: 2864 1st Qu.: 0 1st Qu.:100.2   
## Yes: 37 Median : 3786 Median : 1025 Median :125.0   
## Mean : 4806 Mean : 1570 Mean :136.1   
## 3rd Qu.: 5060 3rd Qu.: 2430 3rd Qu.:158.0   
## Max. :72529 Max. :24000 Max. :550.0   
## NA's :5   
## Loan\_Amount\_Term Credit\_History Property\_Area  
## Min. : 6.0 Min. :0.0000 Rural :111   
## 1st Qu.:360.0 1st Qu.:1.0000 Semiurban:116   
## Median :360.0 Median :1.0000 Urban :140   
## Mean :342.5 Mean :0.8254   
## 3rd Qu.:360.0 3rd Qu.:1.0000   
## Max. :480.0 Max. :1.0000   
## NA's :6 NA's :29

The summary shows, we have NA values to handle let's explore our data more  
Checking our target variable- **Loan\_Status**

table(train\_data$Loan\_Status)

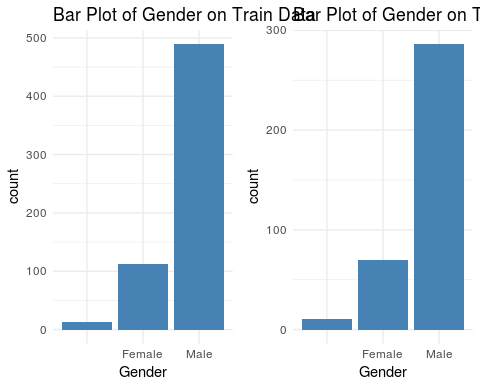
##   
## N Y   
## 192 422

barplot(table(train\_data$Loan\_Status),main= "Loan\_Status")



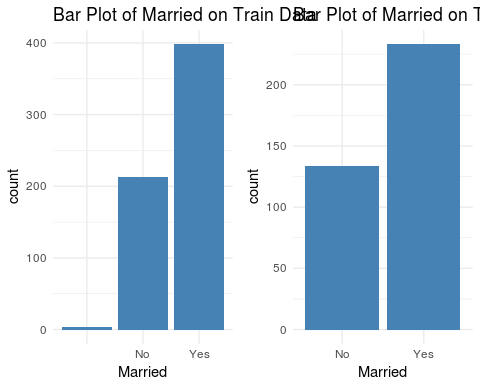
Lets's Explore our Independent variables-  
1. Gender

plot1 <- ggplot(data=train\_data, aes(train\_data$Gender))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Gender")+labs(title="Bar Plot of Gender on Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Gender))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Gender")+labs(title="Bar Plot of Gender on Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)



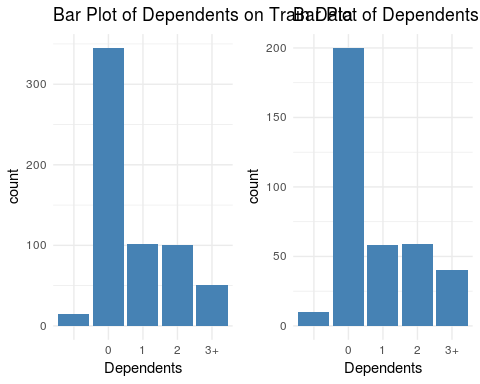
1. Married

plot1 <- ggplot(data=train\_data, aes(train\_data$Married))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Married")+labs(title="Bar Plot of Married on Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Married))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Married")+labs(title="Bar Plot of Married on Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)



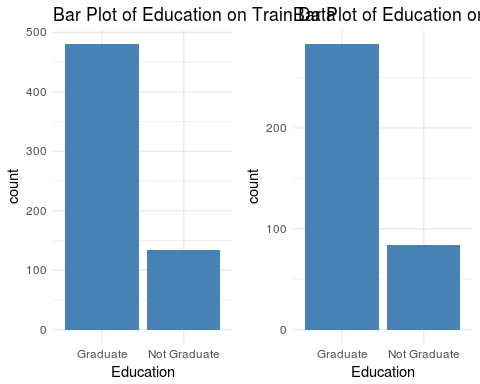
1. Dependents

plot1 <- ggplot(data=train\_data, aes(train\_data$Dependents))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Dependents")+labs(title="Bar Plot of Dependents on Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Dependents))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Dependents")+labs(title="Bar Plot of Dependents on Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)



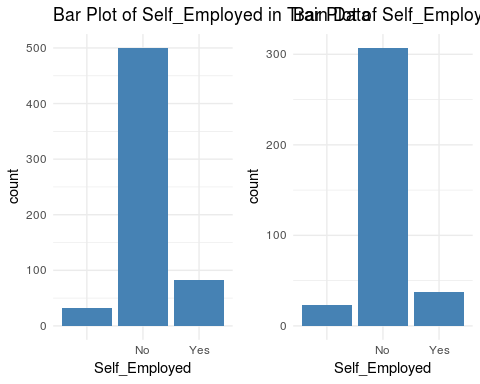
1. Education

plot1 <- ggplot(data=train\_data, aes(train\_data$Education))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Education")+labs(title="Bar Plot of Education on Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Education))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Education")+labs(title="Bar Plot of Education on Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)



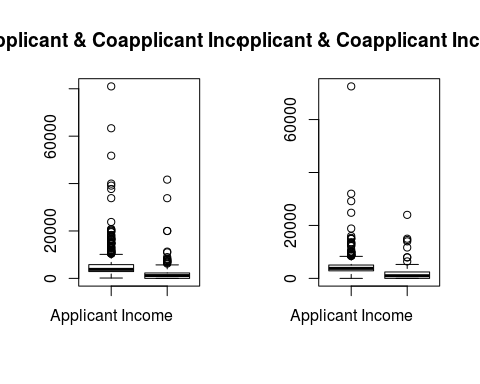
1. Self\_Employed

plot1 <- ggplot(data=train\_data, aes(train\_data$Self\_Employed))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Self\_Employed")+labs(title="Bar Plot of Self\_Employed in Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Self\_Employed))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Self\_Employed")+labs(title="Bar Plot of Self\_Employed in Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)



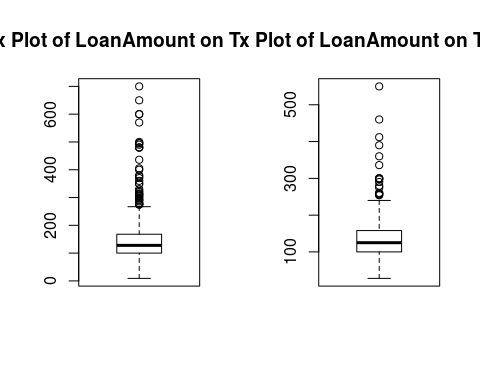
1. ApplicantIncome (Numeric) & CoapplicantIncome (Numeric)

par(mfrow=c(1,2))  
boxplot(train\_data$ApplicantIncome,train\_data$CoapplicantIncome,names=c("Applicant Income","Coapplicant Income"),main="Box Plot of Applicant & Coapplicant Income on Train Data")  
boxplot(test\_data$ApplicantIncome,test\_data$CoapplicantIncome,names=c("Applicant Income","Coapplicant Income"),main="Box Plot of Applicant & Coapplicant Income on Test Data")



1. LoanAmount (Numeric)

par(mfrow=c(1,2))  
boxplot(train\_data$LoanAmount,main="Box Plot of LoanAmount on Train set")  
boxplot(test\_data$LoanAmount,main="Box Plot of LoanAmount on Test set")



1. Loan\_Amount\_Term (Numeric)

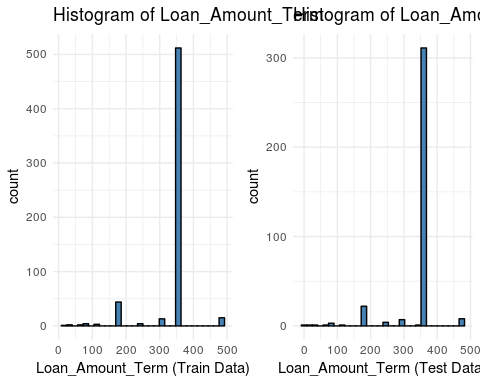
plot1 <- ggplot(data=train\_data, aes(train\_data$Loan\_Amount\_Term))+geom\_histogram(col="black",fill="steelblue",alpha=1)+theme\_minimal()+labs(x = "Loan\_Amount\_Term (Train Data)")+labs(title= "Histogram of Loan\_Amount\_Term")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Loan\_Amount\_Term))+geom\_histogram(col="black",fill="steelblue",alpha=1)+theme\_minimal()+labs(x = "Loan\_Amount\_Term (Test Data)")+labs(title= "Histogram of Loan\_Amount\_Term")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 14 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 6 rows containing non-finite values (stat\_bin).

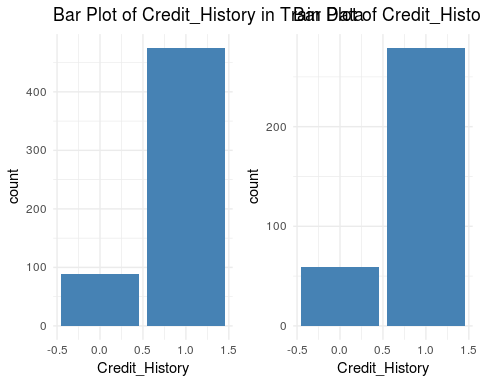


1. Credit\_History (Factor)

plot1 <- ggplot(data=train\_data, aes(train\_data$Credit\_History))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Credit\_History")+labs(title="Bar Plot of Credit\_History in Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Credit\_History))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Credit\_History")+labs(title="Bar Plot of Credit\_History in Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)

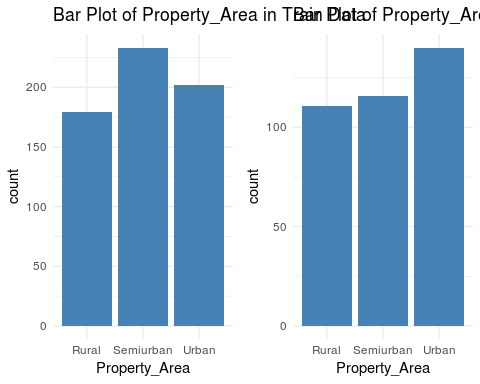
## Warning: Removed 50 rows containing non-finite values (stat\_count).

## Warning: Removed 29 rows containing non-finite values (stat\_count).

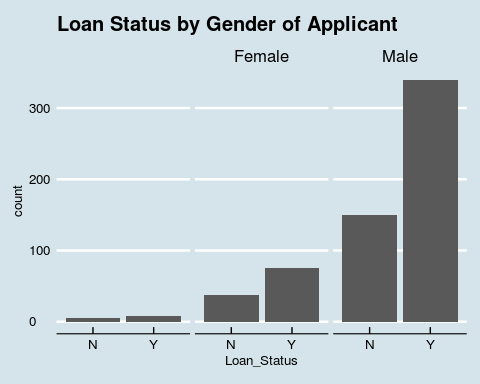


1. Property\_Area

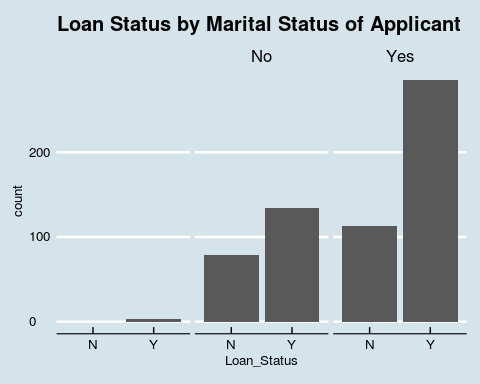
plot1 <- ggplot(data=train\_data, aes(train\_data$Property\_Area))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Property\_Area")+labs(title="Bar Plot of Property\_Area in Train Data")  
plot2 <- ggplot(data=test\_data, aes(test\_data$Property\_Area))+geom\_bar(fill="steelblue")+theme\_minimal()+labs(x = "Property\_Area")+labs(title="Bar Plot of Property\_Area in Test Data")  
grid.arrange(plot1, plot2, nrow=1, ncol=2)



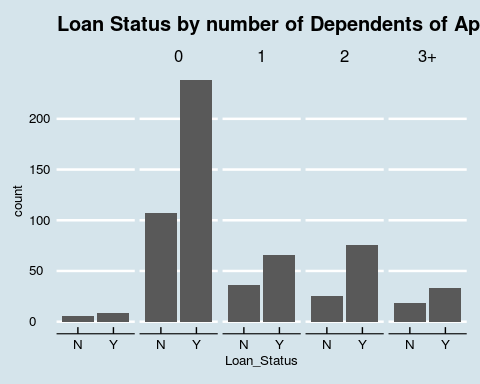
theme\_set(theme\_economist())  
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Gender)+ggtitle("Loan Status by Gender of Applicant"))



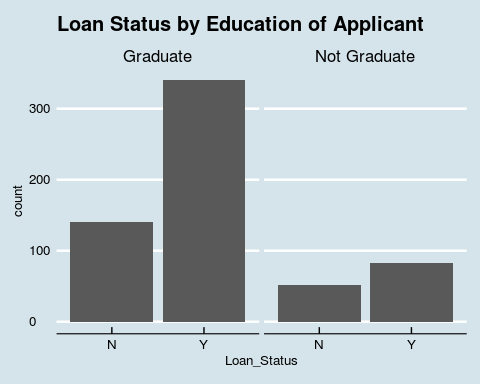
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Married)+ggtitle("Loan Status by Marital Status of Applicant"))



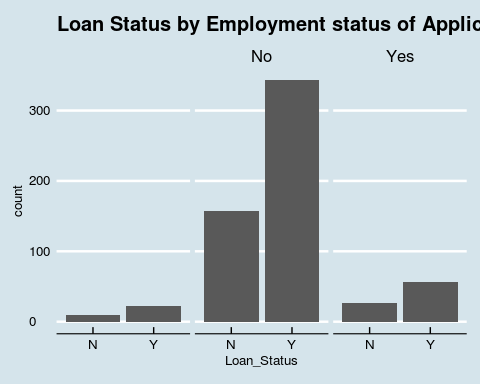
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Dependents)+ggtitle("Loan Status by number of Dependents of Applicant"))



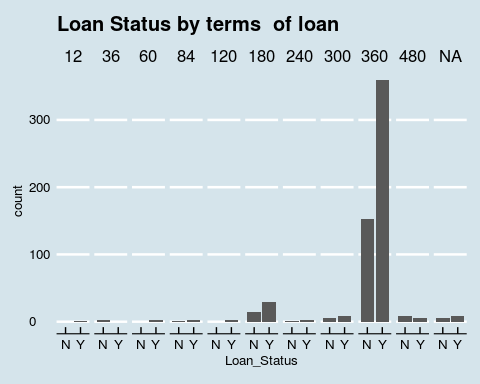
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Education)+ggtitle("Loan Status by Education of Applicant"))



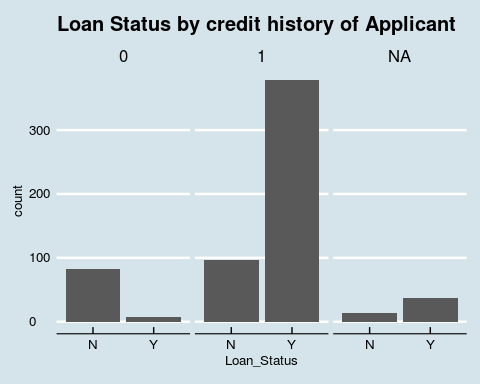
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Self\_Employed)+ggtitle("Loan Status by Employment status of Applicant"))



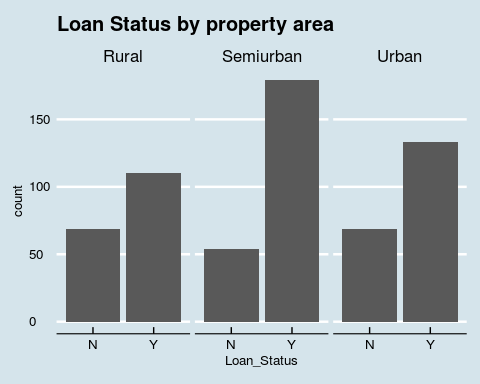
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Loan\_Amount\_Term)+ggtitle("Loan Status by terms of loan"))



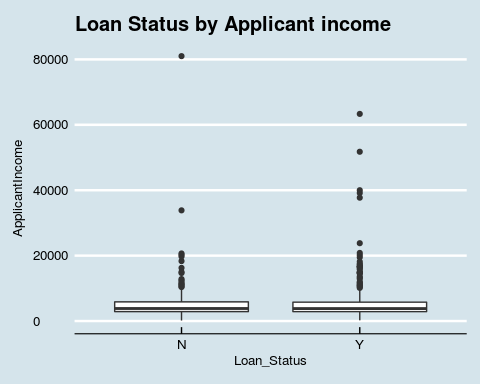
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Credit\_History)+ggtitle("Loan Status by credit history of Applicant"))



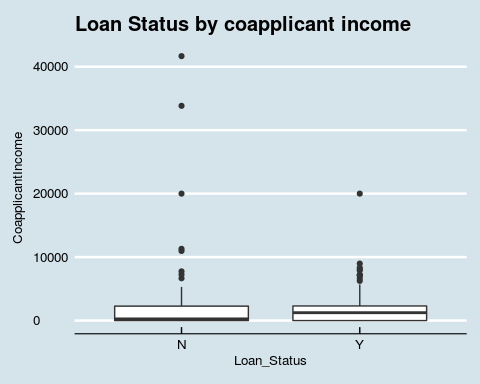
print(ggplot(train\_data, aes(x=Loan\_Status))+geom\_bar()+facet\_grid(.~Property\_Area)+ggtitle("Loan Status by property area"))



print(ggplot(train\_data, aes(x=Loan\_Status,y=ApplicantIncome))+geom\_boxplot()+ggtitle("Loan Status by Applicant income"))

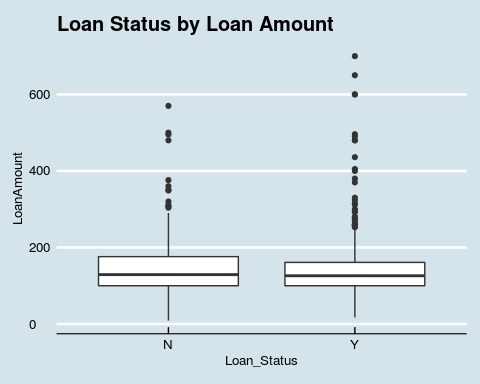


print(ggplot(train\_data, aes(x=Loan\_Status,y=CoapplicantIncome))+geom\_boxplot()+ggtitle("Loan Status by coapplicant income"))



print(ggplot(train\_data, aes(x=Loan\_Status,y=LoanAmount))+geom\_boxplot()+ggtitle("Loan Status by Loan Amount"))

## Warning: Removed 22 rows containing non-finite values (stat\_boxplot).

 \*\* Filling NA values\*\* 1. Usually while applying for loan total family income is calculated So we can add Applicant and Co\_Applicant income, but before that we need to combine our dataset

Complete\_Data <- rbind(train\_data[,2:12],test\_data[,2:12])  
Complete\_Data <- mutate(Complete\_Data,TotalIncome=ApplicantIncome+CoapplicantIncome)

When there is "No" co-applicant income assuming as Unmarried and Married otherwise

Complete\_Data$Gender <- as.character(Complete\_Data$Gender)  
Complete\_Data$Married <- as.character(Complete\_Data$Married)  
Complete\_Data$Self\_Employed <- as.character(Complete\_Data$Self\_Employed)  
Complete\_Data$Married[Complete\_Data$Married=="" & Complete\_Data$CoapplicantIncome==0]<-"No"  
Complete\_Data$Married[Complete\_Data$Married==""]<- "Yes"

Plots show that if gender is male its income is more than female so

Complete\_Data$Dependents <- as.character(Complete\_Data$Dependents)  
Complete\_Data$Gender[Complete\_Data$Gender=="" & Complete\_Data$Dependents==""] <- "Male"

When Dependents is unknown but not married then assuming no dependents

Complete\_Data$Dependents[Complete\_Data$Dependents=="" & Complete\_Data$Married=="No"]<- "0"

Most of the loan term is 360, so filling NA as 360 in loan amount and renaming 350 as 360 and 6 as 60 since their feq is low and m,ight be due to tying error while entering data

Complete\_Data$Loan\_Amount\_Term[is.na(Complete\_Data$Loan\_Amount\_Term)]<-"360"  
library(car)  
Complete\_Data$Loan\_Amount\_Term <- recode(Complete\_Data$Loan\_Amount\_Term,"'350'='360';'6'='60'")

Assuming "" empty factor in self employed. As most are not self employed

Complete\_Data$Self\_Employed[Complete\_Data$Self\_Employed==""] <- "No"

Assuming person with no credit history as another catrgory

Complete\_Data$Credit\_History<-recode(Complete\_Data$Credit\_History,"NA=2")  
Complete\_Data$Gender <- as.factor(Complete\_Data$Gender)  
Complete\_Data$Married <- as.factor(Complete\_Data$Married)  
Complete\_Data$Dependents <- as.factor(Complete\_Data$Dependents)  
Complete\_Data$Self\_Employed <- as.factor(Complete\_Data$Self\_Employed)  
Complete\_Data$Loan\_Amount\_Term <- as.factor(Complete\_Data$Loan\_Amount\_Term)

To predict Remaining Gender by (Mode Imputation) & Dependents

levels(Complete\_Data$Gender)[levels(Complete\_Data$Gender)==""] <- "Male"  
levels(Complete\_Data$Dependents)[levels(Complete\_Data$Dependents)==""] <- "0"

We will predict Loan Amount using K-Nearrest neighbours

preProcValues <- preProcess(Complete\_Data, method = c("knnImpute","center","scale"))  
Complete\_data\_processed <- predict(preProcValues, Complete\_Data)

Splitting Training and Test Data set back

trainSet <- Complete\_data\_processed[ 1:614,]  
trainSet$Loan\_Status <- train\_data$Loan\_Status  
testSet <- Complete\_data\_processed[615:981,]

Creating Train control

fitControl <- trainControl(method = "cv", number = 7, savePredictions = 'final', classProbs = T)

Creating Random Forest

#Training the random forest model  
model\_rf<-train(Loan\_Status~.,data=trainSet,method='rf',trControl=fitControl,tuneLength=3)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

#Predicting using random forest model  
testSet$pred\_rf<-predict(object = model\_rf,testSet[,-13])

This was the uploaded to kaggle and output accuracy was 81.1% after tuning random forest.