Loan Analysis for a Bank application

Prithviraj Guntha : 20BRS1188

2023-02-25

Performing data processing data manipulation, data modelling and statistical analysis for a loan dataset for a bank.

PART 1: we will now load the necessary packages

```
library(ggplot2)
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(caret)
```

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

```
library(rpart)
library(rpart.plot)
library(reshape2)
```

PART 2: Loading the dataset

```
data <- read.csv("/Users/prithviraj/Downloads/df1_loan.csv")
str(data)
```

```
## 'data.frame':	500 obs. of  15 variables:
##  $ X                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Loan_ID          : chr  "LP001002" "LP001003" "LP001005" "LP001006" ...
##  $ Gender           : chr  "Male" "Male" "Male" "Male" ...
##  $ Married          : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Dependents       : chr  "0" "1" "0" "0" ...
##  $ Education        : chr  "Graduate" "Graduate" "Graduate" "Not Graduate" ...
##  $ Self_Employed    : chr  "No" "No" "Yes" "No" ...
##  $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
##  $ CoapplicantIncome: num  0 1508 0 2358 0 ...
##  $ LoanAmount       : num  NA 128 66 120 141 267 95 158 168 349 ...
##  $ Loan_Amount_Term : num  360 360 360 360 360 360 360 360 360 360 ...
##  $ Credit_History   : num  1 1 1 1 1 1 1 0 1 1 ...
##  $ Property_Area    : chr  "Urban" "Rural" "Urban" "Urban" ...
##  $ Loan_Status      : chr  "Y" "N" "Y" "Y" ...
##  $ Total_Income     : chr  "$5849.0" "$6091.0" "$3000.0" "$4941.0" ...
```

PART 3: removing unwanted columns

```
data <- data[,-c(1,2,5,9,15)]
str(data)
```

```
## 'data.frame':	500 obs. of  10 variables:
##  $ Gender          : chr  "Male" "Male" "Male" "Male" ...
##  $ Married         : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Education       : chr  "Graduate" "Graduate" "Graduate" "Not Graduate" ...
##  $ Self_Employed   : chr  "No" "No" "Yes" "No" ...
##  $ ApplicantIncome : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
##  $ LoanAmount      : num  NA 128 66 120 141 267 95 158 168 349 ...
##  $ Loan_Amount_Term: num  360 360 360 360 360 360 360 360 360 360 ...
##  $ Credit_History  : num  1 1 1 1 1 1 1 0 1 1 ...
##  $ Property_Area   : chr  "Urban" "Rural" "Urban" "Urban" ...
##  $ Loan_Status     : chr  "Y" "N" "Y" "Y" ...
```

PART 4: Checking for missing values and removing them

before removing the missing values:

```
missing_values <- colSums(is.na(data))
missing_values
```

```
##           Gender          Married        Education    Self_Employed 
##                0                0                0                0 
##  ApplicantIncome       LoanAmount Loan_Amount_Term   Credit_History 
##                0               18               14               41 
##    Property_Area      Loan_Status 
##                0                0
```

```
data <- data[complete.cases(data),]
```

after removing the missing values

```
missing_values <- colSums(is.na(data))
missing_values
```

```
##          Gender           Married         Education     Self_Employed
##               0                 0                 0                 0
##   ApplicantIncome        LoanAmount Loan_Amount_Term    Credit_History
##               0                 0                 0                 0
##     Property_Area       Loan_Status
##               0                 0
```

PART 5: # Convert “Gender”, “Married”, “Education”, “Self_Employed”, “Property_Area”, and “Loan_Status” columns to factors

```
data$Gender <- as.factor(data$Gender)
data$Married <- as.factor(data$Married)
data$Education <- as.factor(data$Education)
data$Self_Employed <- as.factor(data$Self_Employed)
data$Property_Area <- as.factor(data$Property_Area)
data$Loan_Status <- as.factor(data$Loan_Status)
str(data)
```

```
## 'data.frame':    428 obs. of  10 variables:
##  $ Gender          : Factor w/ 3 levels "","Female","Male": 3 3 3 3 3 3 3 3 3 3
...
##  $ Married         : Factor w/ 3 levels "","No","Yes": 3 3 3 2 3 3 3 3 3 3 ...
##  $ Education       : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 2 1 1 2 1 1
1 1 ...
##  $ Self_Employed   : Factor w/ 3 levels "","No","Yes": 2 3 2 2 3 2 2 2 2 2 ...
##  $ ApplicantIncome : int  4583 3000 2583 6000 5417 2333 3036 4006 12841 3200 ...
##  $ LoanAmount      : num  128 66 120 141 267 95 158 168 349 70 ...
##  $ Loan_Amount_Term: num  360 360 360 360 360 360 360 360 360 360 ...
##  $ Credit_History  : num  1 1 1 1 1 1 0 1 1 1 ...
##  $ Property_Area   : Factor w/ 3 levels "Rural","Semiurban",..: 1 3 3 3 3 3 2 3 2
3 ...
##  $ Loan_Status     : Factor w/ 2 levels "N","Y": 1 2 2 2 2 2 1 2 1 2 ...
```

PART 6: splitting the data intro train and test sets

```
set.seed(123)
trainIndex <- createDataPartition(data$Loan_Status, p = 0.8, list = FALSE)
train_data <- data[trainIndex, ]
test_data <- data[-trainIndex, ]
```

PART 7: Building data models

1. using a logistic regression model

building the model

```
loan_model <- glm(Loan_Status ~ ., data = train_data, family = binomial)
```

making predictions using the model

```
test_pred <- predict(loan_model, newdata = test_data, type = "response")
test_pred_class <- ifelse(test_pred > 0.5, "Y", "N")
```

getting the confusion matrix and model statistics

```
Loan_Statuss <- ifelse(test_data$Loan_Status == "Y", 1, 0)
test_pred_class <- ifelse(test_pred_class == "Y", 1, 0)
test_pred_class <- unname(test_pred_class)
confusionMatrix(as.factor(test_pred_class), as.factor(Loan_Statuss))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 13  2
##          1 13 57
##
##                Accuracy : 0.8235
##                  95% CI : (0.7257, 0.8977)
##     No Information Rate : 0.6941
##     P-Value [Acc > NIR] : 0.005003
##
##                   Kappa : 0.5287
##
##  Mcnemar's Test P-Value : 0.009823
##
##             Sensitivity : 0.5000
##             Specificity : 0.9661
##          Pos Pred Value : 0.8667
##          Neg Pred Value : 0.8143
##              Prevalence : 0.3059
##          Detection Rate : 0.1529
##    Detection Prevalence : 0.1765
##       Balanced Accuracy : 0.7331
##
##        'Positive' Class : 0
##
```

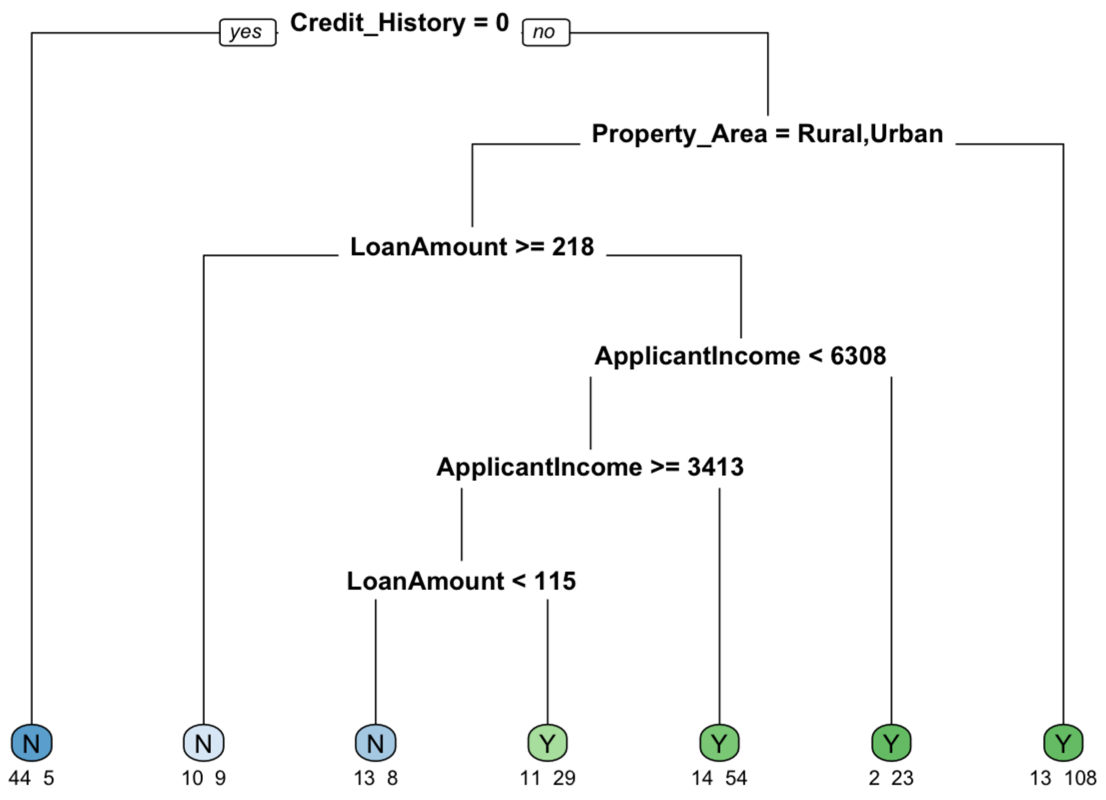
2. using a decision tree model

building the model

```
loan_model <- rpart(Loan_Status ~ ., data = train_data, method = "class")
```

plotting the model

```
rpart.plot(loan_model, type = 0, extra = 1, under = TRUE, varlen = 0, cex = 0.8)
```



making predictions using the model

```
test_pred <- predict(loan_model, newdata = test_data, type = "class")
```

getting the confusion matrix and model statistics

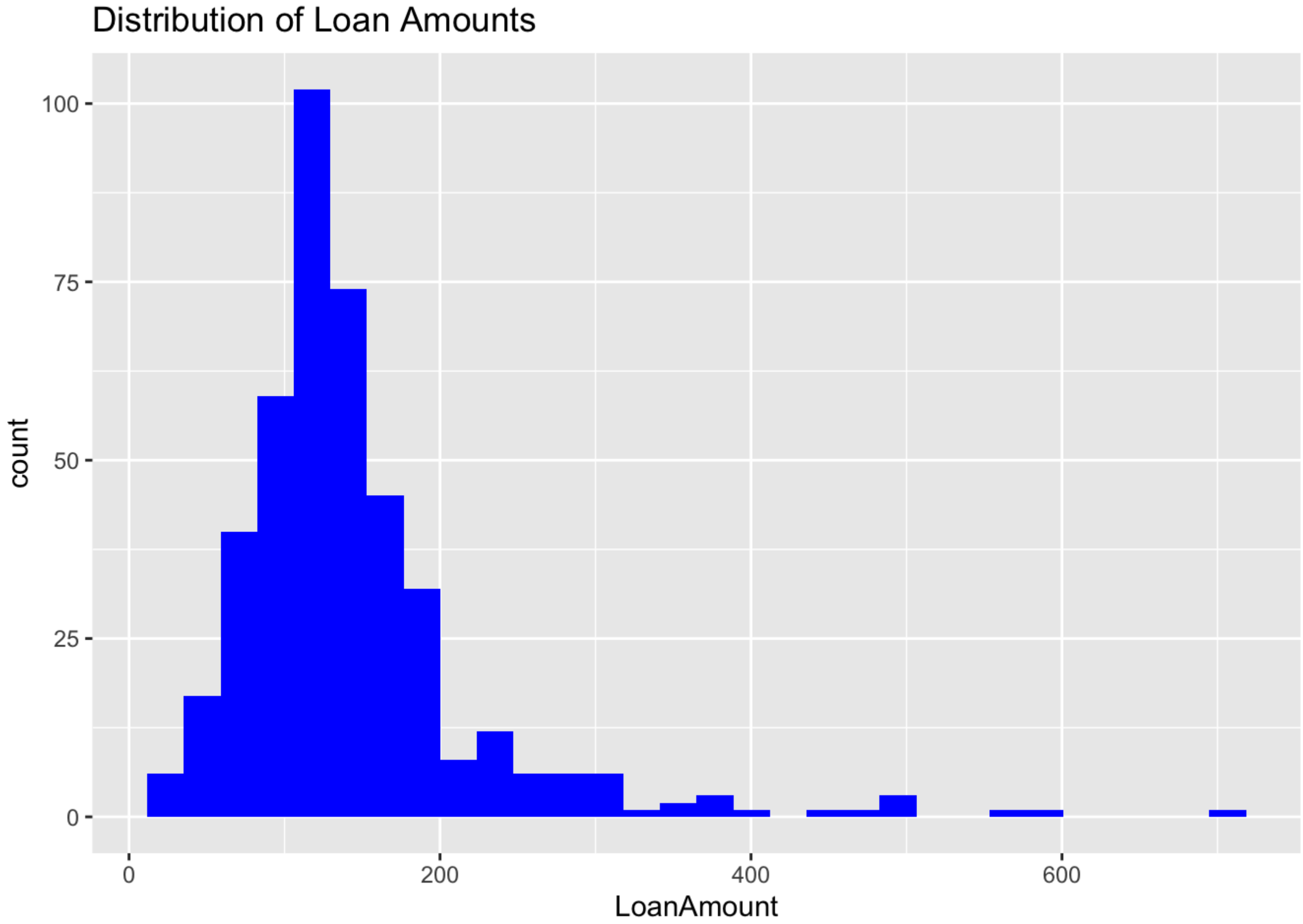
```
Loan_Statuss <- ifelse(test_data$Loan_Status == "Y", 1, 0)
test_pred <- ifelse(test_pred == "Y", 1, 0)
test_pred <- unname(test_pred)
confusionMatrix(as.factor(test_pred), as.factor(Loan_Statuss))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 13 11
##          1 13 48
##
##                Accuracy : 0.7176
##                  95% CI : (0.6096, 0.81)
##     No Information Rate : 0.6941
##     P-Value [Acc > NIR] : 0.3672
##
##                   Kappa : 0.3205
##
##  Mcnemar's Test P-Value : 0.8383
##
##             Sensitivity : 0.5000
##             Specificity : 0.8136
##          Pos Pred Value : 0.5417
##          Neg Pred Value : 0.7869
##              Prevalence : 0.3059
##          Detection Rate : 0.1529
##    Detection Prevalence : 0.2824
##       Balanced Accuracy : 0.6568
##
##        'Positive' Class : 0
##
```

PART 8: Statistical analysis

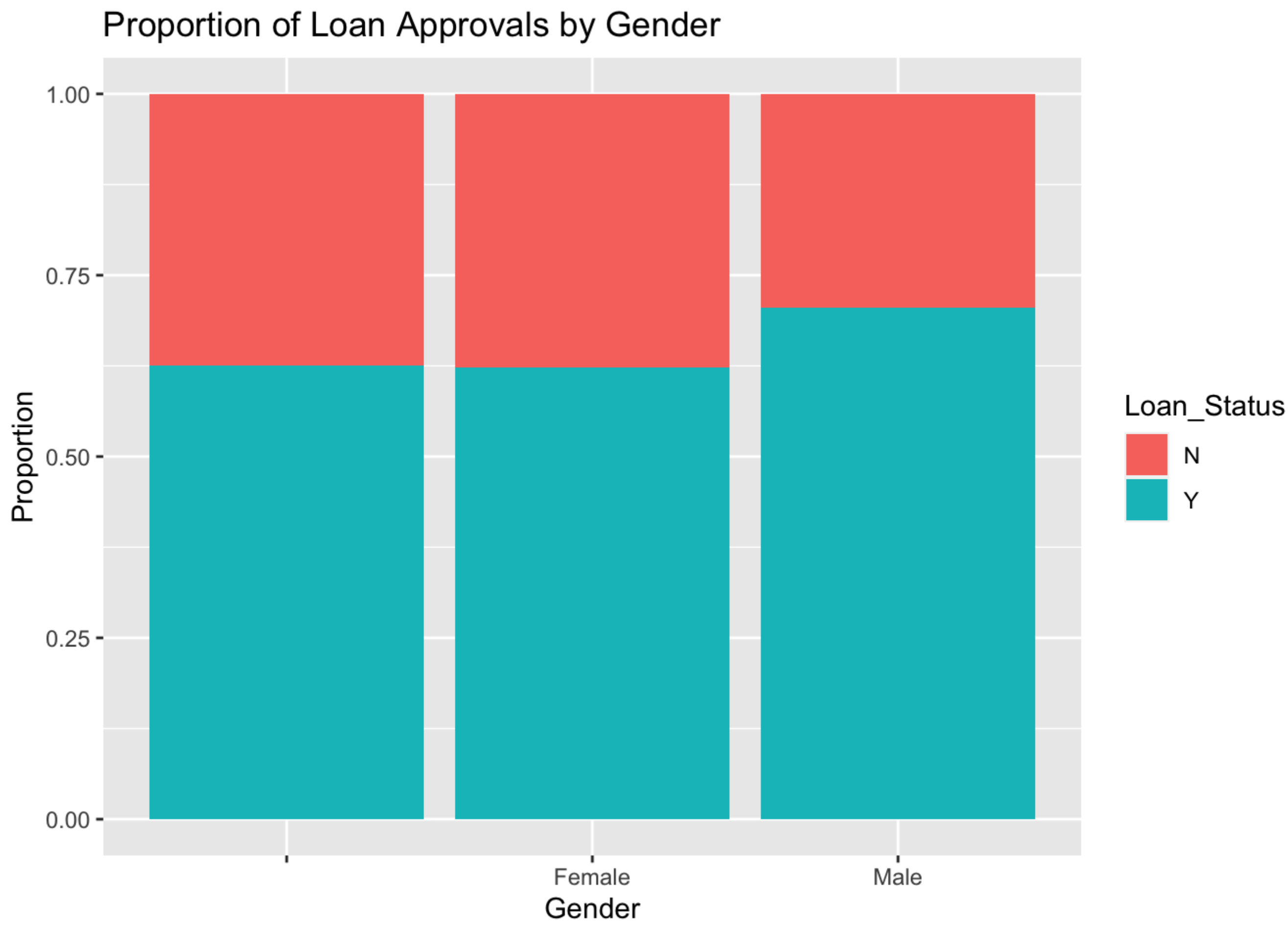
Plotting the distribution of Loan Amounts

```
ggplot(data = data, aes(x = LoanAmount)) +
  geom_histogram(fill = "blue", bins = 30) +
  labs(title = "Distribution of Loan Amounts")
```



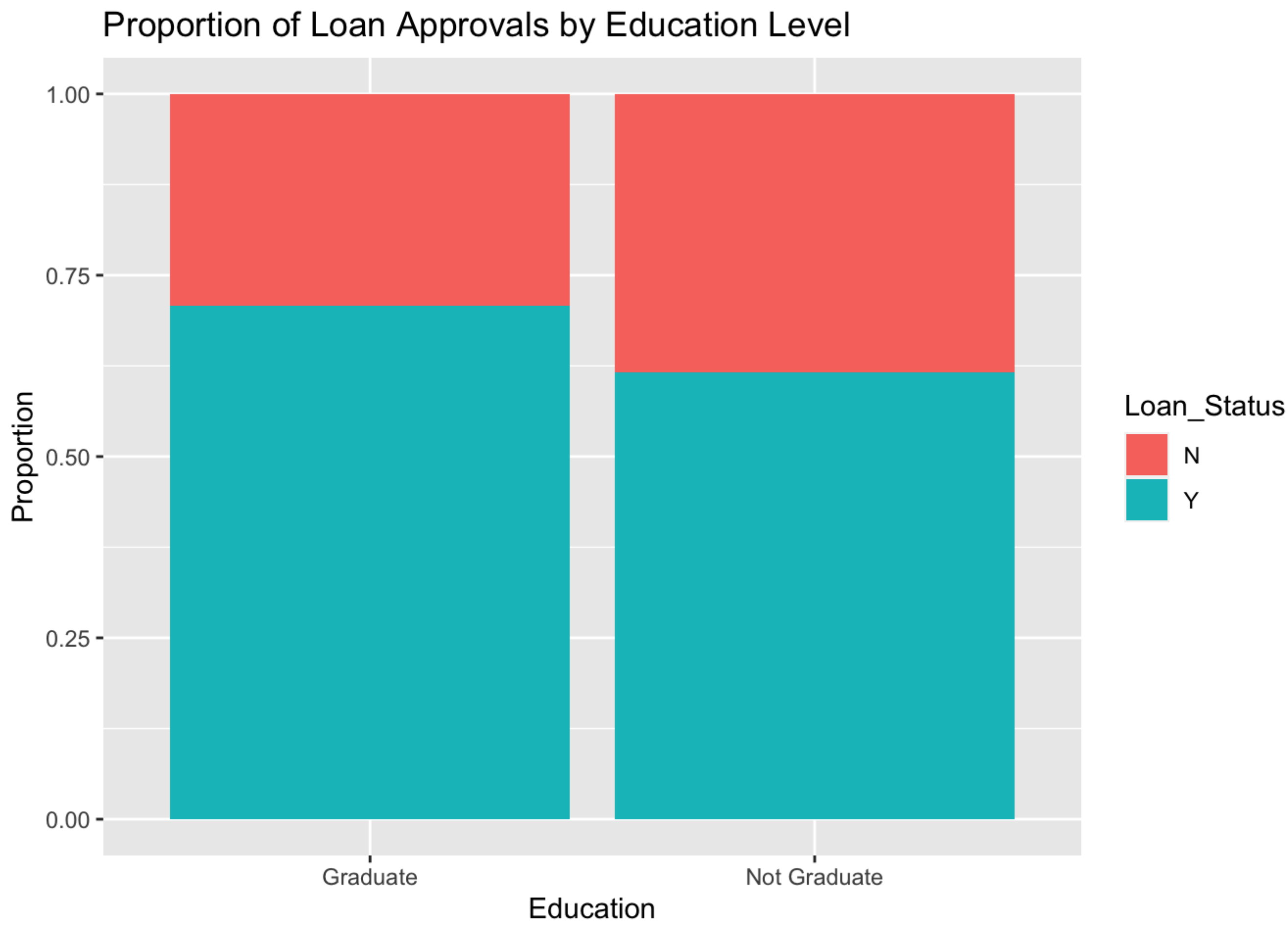
Plotting the proportion of loan approvals by gender

```
ggplot(data = data, aes(x = Gender, fill = Loan_Status)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of Loan Approvals by Gender", y = "Proportion")
```



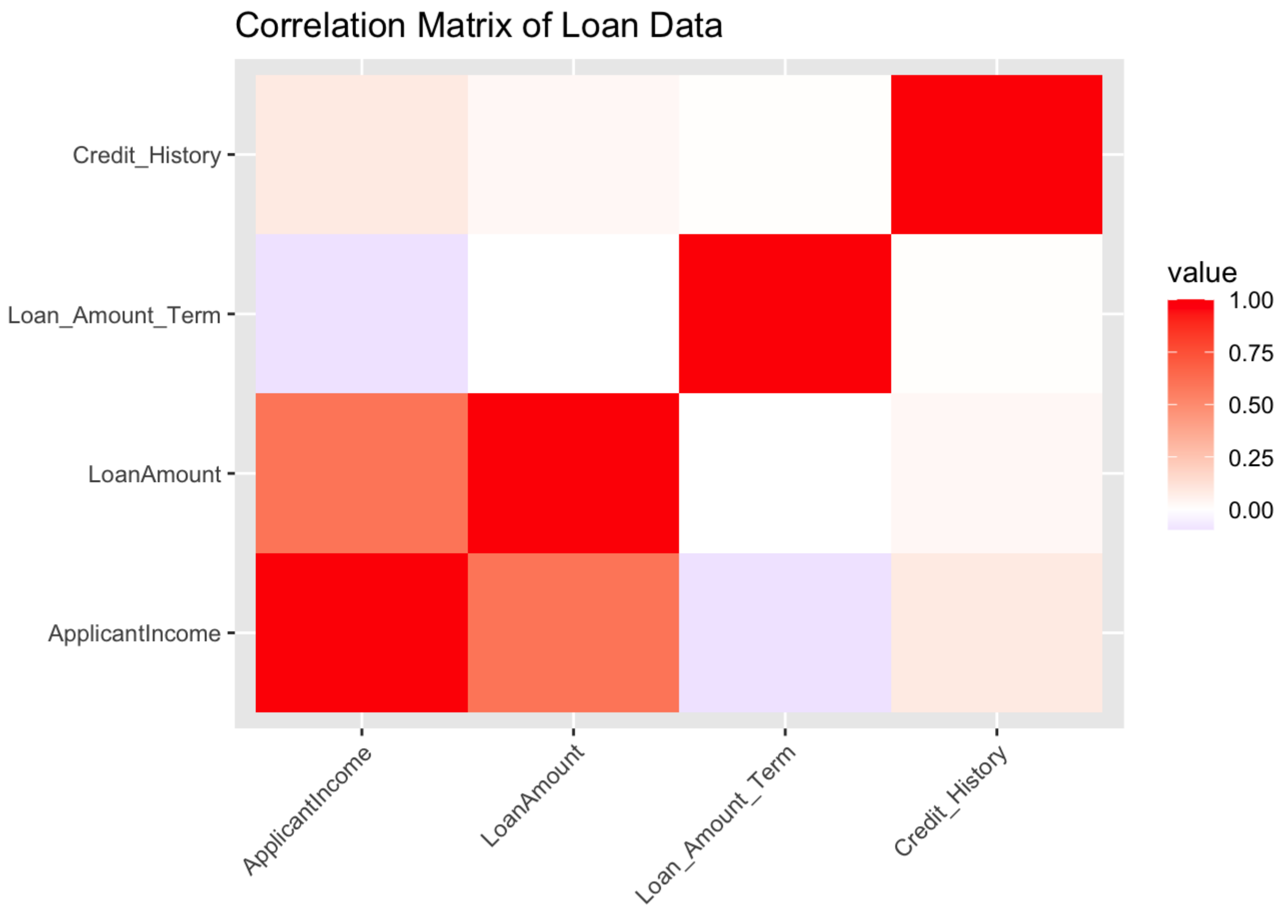
Plotting the proportion of loan approvals by education level

```
ggplot(data = data, aes(x = Education, fill = Loan_Status)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of Loan Approvals by Education Level", y = "Proportion")
```



PART 9: Correlation analysis

```
loan_cor <- cor(train_data[, c("ApplicantIncome", "LoanAmount", "Loan_Amount_Term",
"Credit_History")])
ggplot(data = melt(loan_cor), aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Correlation Matrix of Loan Data", x = "", y = "") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```


Correlation Matrix of Loan Data
Credit_History
Loan_Amount_Term
LoanAmount
ApplicantIncome
ApplicantIncome
LoanAmount
Loan_Amount_Term
Credit_History
value
1.00
0.75
0.50
0.25
0.00