Loans

Tina Hajinejad

2023-04-19

Importing the necessary libraries:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(skimr)
library(abind)
```

Question 1: Describe and justify two different topics or approaches you might want to consider for this dataset and task. (Mandatory)

1. Loan Approval Prediction: One approach could be to build a machine learning model that predicts whether a loan application will be approved or not based on the given variables. This can be framed as a binary classification problem where "Loan_Status" is the target variable. Various classification algorithms such as logistic regression, decision trees,… can be trained on the dataset to predict the likelihood of a loan application being approved. Model Selection techniques can also be applied to extract more meaningful information from the given variables.
2. Loan Amount Estimation: Another approach could be to analyze the factors that affect the loan amount and build a regression model to estimate the loan amount based on the given variables. This can be framed as a regression problem where "LoanAmount" is the target variable. The impact of various factors such as "ApplicantIncome", "CoapplicantIncome", "Loan_Amount_Term", "Credit_History", and "Property_Area" on the loan amount can be explored through visualizations and statistical analysis.

Question 2: Describe and show the code used to clean and collect the data. (Optional)

The data file contains train and test sets. I wanted to merge them together to have as much data as possible for analysis but that is not possible because the Loan_Status column does not exist in the test set.

```
train <- read.csv("loan_sanction_train.csv")
test <- read.csv("loan_sanction_test.csv")
names(train)
```

```
##  [1] "Loan_ID"           "Gender"            "Married"
##  [4] "Dependents"        "Education"         "Self_Employed"
##  [7] "ApplicantIncome"   "CoapplicantIncome" "LoanAmount"
## [10] "Loan_Amount_Term"  "Credit_History"    "Property_Area"
## [13] "Loan_Status"
```

```
names(test)
```

```
##  [1] "Loan_ID"           "Gender"            "Married"
##  [4] "Dependents"        "Education"         "Self_Employed"
##  [7] "ApplicantIncome"   "CoapplicantIncome" "LoanAmount"
## [10] "Loan_Amount_Term"  "Credit_History"    "Property_Area"
```

```
#data <- rbind(train,test)
```

First let's take a look at a summary of data, to check for the data types, and see if we have any missing data.

```
skim(train)
```

Table 1: Data summary

Name	train
Number of rows	614
Number of columns	13
Column type frequency:	
character	8
numeric	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Loan_ID	0	1	8	8	0	614	0
Gender	0	1	0	6	13	3	0
Married	0	1	0	3	3	3	0
Dependents	0	1	0	2	15	5	0
Education	0	1	8	12	0	2	0
Self_Employed	0	1	0	3	32	3	0
Property_Area	0	1	5	9	0	3	0
Loan_Status	0	1	1	1	0	2	0

Variable type: numeric

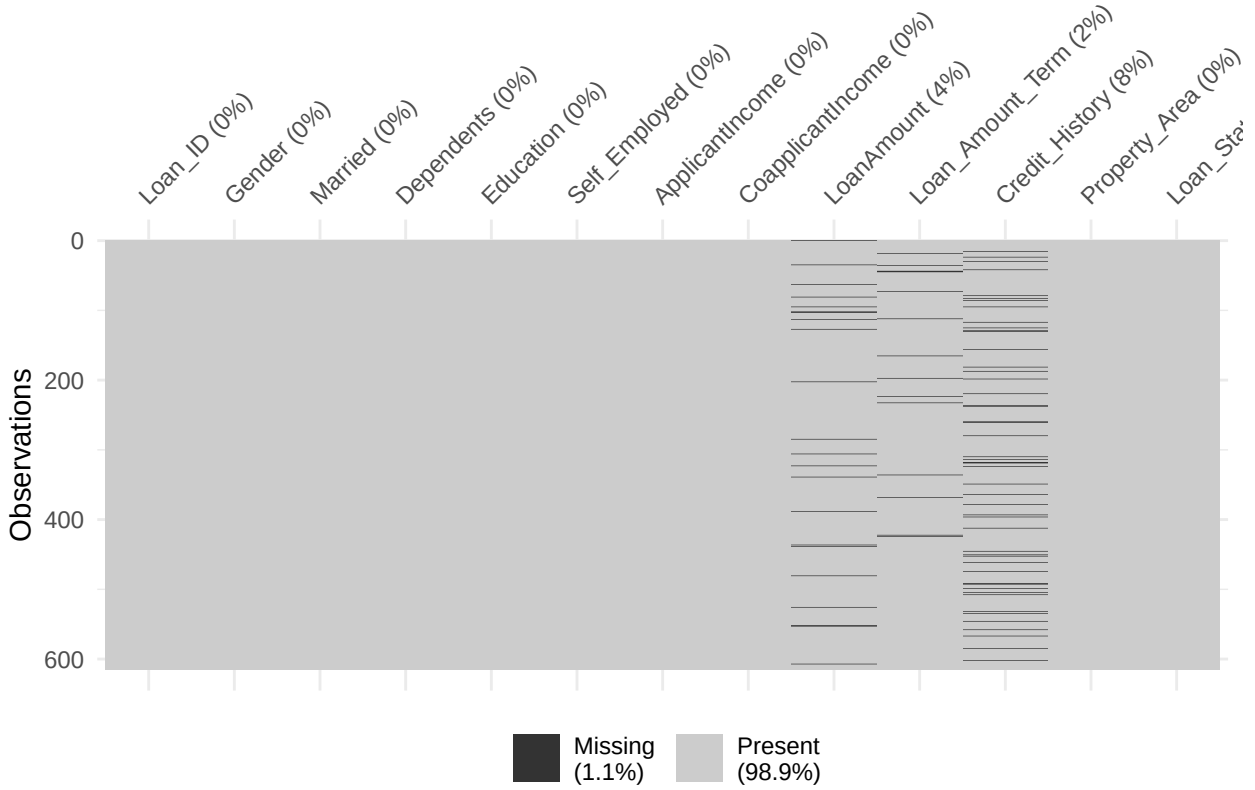
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ApplicantIncome	0	1.00	5403.46	6109.04	150	2877.5	3812.5	5795.00	81000	
CoapplicantIncome	0	1.00	1621.25	2926.25	0	0.0	1188.5	2297.25	41667	
LoanAmount	22	0.96	146.41	85.59	9	100.0	128.0	168.00	700	
Loan_Amount_Term	14	0.98	342.00	65.12	12	360.0	360.0	360.00	480	
Credit_History	50	0.92	0.84	0.36	0	1.0	1.0	1.00	1	

```
library(naniar)
```

```
##
## Attaching package: 'naniar'
```

```
## The following object is masked from 'package:skimr':
##
##     n_complete
```

```
## visualize missingness
vis_miss(train)
```



We neglect the credit history, as it does not mention on the data card if 1 and 0 are indication of existing credit history or do they indicate a good or a bad vredit history.We will try to impute the missing data from loan_amount term and loan_amount.

To see how we should impute Loan_Amount_Term, we will look at a table of frequencies.

```
table(train$Loan_Amount_Term)
```

```
##
##  12  36  60  84 120 180 240 300 360 480
##   1   2   2   4   3  44   4  13 512  15
```

In this table, we see that most of the loan terms are 360 months. Therefore 360 is a good value to impute the missingness.

```
train$Loan_Amount_Term[is.na(train$Loan_Amount_Term)] <- 360


skim(train)
```

Table 4: Data summary

Name	train
Number of rows	614
Number of columns	13
Column type frequency:	
character	8
numeric	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Loan_ID	0	1	8	8	0	614	0
Gender	0	1	0	6	13	3	0
Married	0	1	0	3	3	3	0
Dependents	0	1	0	2	15	5	0
Education	0	1	8	12	0	2	0
Self_Employed	0	1	0	3	32	3	0
Property_Area	0	1	5	9	0	3	0
Loan_Status	0	1	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ApplicantIncome	0	1.00	5403.46	6109.04	150	2877.5	3812.5	5795.00	81000	
CoapplicantIncome	0	1.00	1621.25	2926.25	0	0.0	1188.5	2297.25	41667	
LoanAmount	22	0.96	146.41	85.59	9	100.0	128.0	168.00	700	
Loan_Amount_Term	0	1.00	342.41	64.43	12	360.0	360.0	360.00	480	
Credit_History	50	0.92	0.84	0.36	0	1.0	1.0	1.00	1	

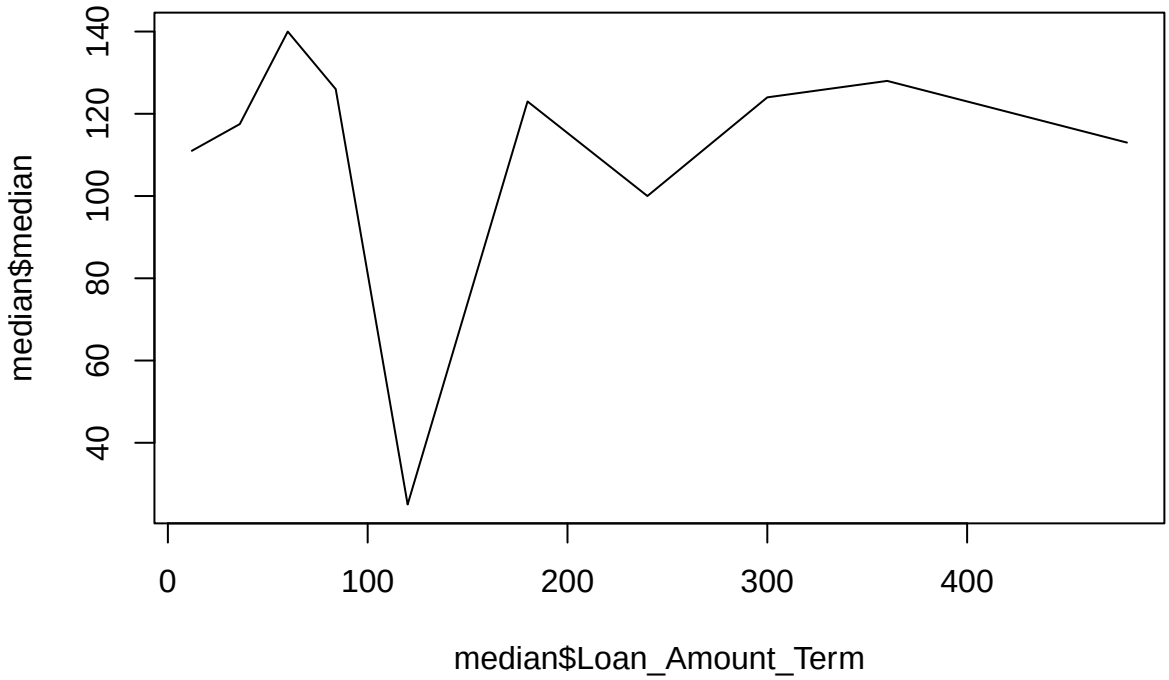
By looking at the skim, we see that the mean for Loan_Amount_Term changed only slightly, therefore the imputation was successful.

The data on loan amount is missing random and can be determined by loan length. we have to decide how to impute that. Two options are using the mean and the median. By plotting th mean and the median for each loan term, we can see if there is a meaningful relationship between mean and loan terms and/or median and loan terms.

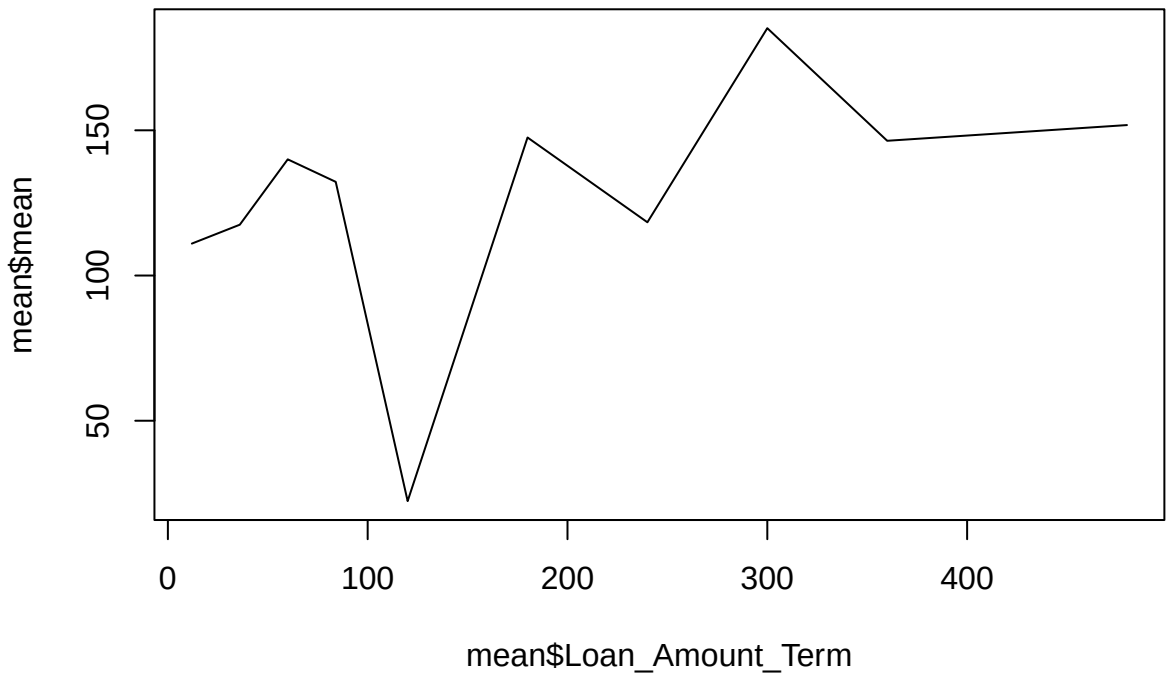
```
median<- ddply(train, "Loan_Amount_Term", summarise,
                      median = median(LoanAmount, na.rm = TRUE))
mean<- ddply(train, "Loan_Amount_Term", summarise,
                      mean= mean(LoanAmount, na.rm = TRUE))
```

let's make a plot and see which one makes more sense.

```
plot(median$Loan_Amount_Term,median$median,type="l")
```




```
plot(mean$Loan_Amount_Term,mean$mean,type="l")
```



Unfortunately, since the data on other loan terms are limited, we can not decide from the plots. We will try stochastic regression-based imputation.

```
library(mice)
```

```
##
```

```
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
#Dropping the Credit History column
train_n = subset(train, select = -c(Credit_History) )

#Using normal distribution, non Bayesian
imp = mice(train_n, method="norm.nob", m=1, seed=12345)
```

```
##
##  iter imp variable
##   1   1  LoanAmount
##   2   1  LoanAmount
##   3   1  LoanAmount
##   4   1  LoanAmount
##   5   1  LoanAmount
```

```
## Warning: Number of logged events: 8
```

```
#Imputing using the complete function

com = complete(imp, action=1)
```

Let's check to see if imputation changed the mean and median of loans with 360 months loan duration.

```
median2<- ddply(com, "Loan_Amount_Term", summarise,
                      median = median(LoanAmount, na.rm = TRUE))
mean2<- ddply(com, "Loan_Amount_Term", summarise,
                     mean= mean(LoanAmount, na.rm = TRUE))

median2$median[9]  #After imputation
```

```
## [1] 128
```

```
median$median[9] #before imputation
```

```
## [1] 128
```

The median stayed the same! Great.

```
mean2$mean[9]  #After imputation
```

```
## [1] 146.2506
```

```
mean$mean[9] #before imputation
```

```
## [1] 146.3886
```

It changes slightly. Which means imputation was good. Now let's check to see if there are any more missing values or not. Remember we dropped the column with credit history.

```
skim(com)
```

Table 7: Data summary

Name	com
Number of rows	614
Number of columns	12
Column type frequency:	
character	8
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Loan_ID	0	1	8	8	0	614	0
Gender	0	1	0	6	13	3	0
Married	0	1	0	3	3	3	0
Dependents	0	1	0	2	15	5	0
Education	0	1	8	12	0	2	0
Self_Employed	0	1	0	3	32	3	0
Property_Area	0	1	5	9	0	3	0
Loan_Status	0	1	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ApplicantIncome	0	1	5403.46	6109.04	150.00	2877.5	3812.5	5795.00	81000	
CoapplicantIncome	0	1	1621.25	2926.25	0.00	0.0	1188.5	2297.25	41667	
LoanAmount	0	1	146.58	85.50	4.38	100.0	128.0	171.50	700	
Loan_Amount_Term	0	1	342.41	64.43	12.00	360.0	360.0	360.00	480	

Next, I have to address the issue that "Dependents" is a categorical variable. let's see how:

```
table(com$Dependents)
```

```
##
##       0   1   2  3+
##  15 345 102 101  51
```

This shows that some NA's were not recognized by R. So We fill them with NA's. This issue probably exist with some other variables.

```
com[com == ""] <- NA
skim(com)
```

Table 10: Data summary

Name	com
Number of rows	614
Number of columns	12

Table 10: Data summary

Column type frequency:	
character	8
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Loan_ID	0	1.00	8	8	0	614	0
Gender	13	0.98	4	6	0	2	0
Married	3	1.00	2	3	0	2	0
Dependents	15	0.98	1	2	0	4	0
Education	0	1.00	8	12	0	2	0
Self_Employed	32	0.95	2	3	0	2	0
Property_Area	0	1.00	5	9	0	3	0
Loan_Status	0	1.00	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ApplicantIncome	0	1	5403.46	6109.04	150.00	2877.5	3812.5	5795.00	81000	
CoapplicantIncome	0	1	1621.25	2926.25	0.00	0.0	1188.5	2297.25	41667	
LoanAmount	0	1	146.58	85.50	4.38	100.0	128.0	171.50	700	
Loan_Amount_Term	0	1	342.41	64.43	12.00	360.0	360.0	360.00	480	

We will move forward with handling missing data if neccessary as we proceed.

Question 3: Give a ggpairs plot of what you think are the six most important variables. At least one must be categorical, and one continuous. Explain your choice of variables and the trends between them. (Mandatory)

First let's take a look at the names.

```
names(com)
```

```
##  [1] "Loan_ID"           "Gender"            "Married"
##  [4] "Dependents"        "Education"         "Self_Employed"
##  [7] "ApplicantIncome"   "CoapplicantIncome" "LoanAmount"
## [10] "Loan_Amount_Term"  "Property_Area"     "Loan_Status"
```

The choice of "Loan_Status", "ApplicantIncome","CoapplicantIncome","LoanAmount","Loan_Amount_Term", and "Property_Area" seems reasonable.

Fortunately, they are not among the variables that we care about for now.

I will divide plots based on property_Are, too see it's effect separately.

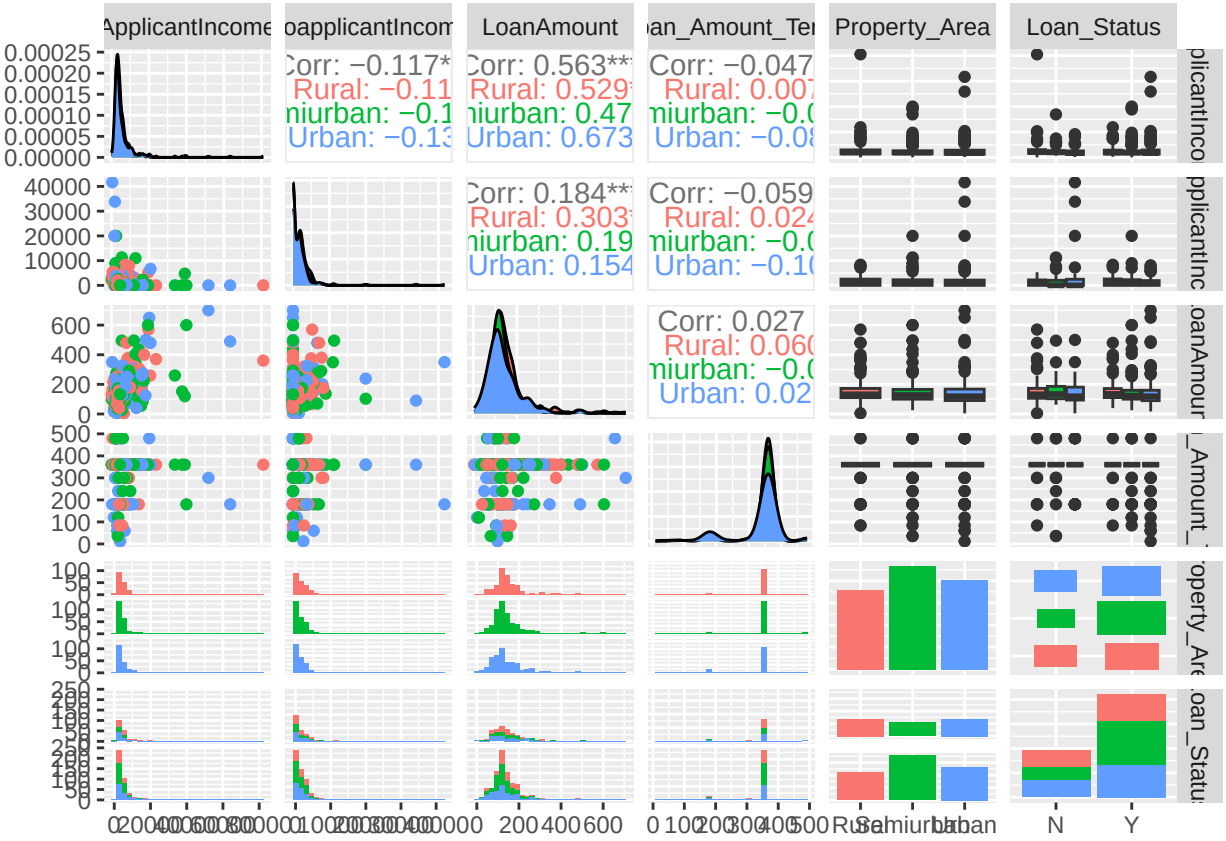
```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
pm2 <- ggpairs(com,
               mapping = aes(color = Property_Area),
               columns = c(7, 8, 9, 10, 11, 12))


print(pm2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



I chose these variables because intuitively, the goal of such analysis is that to see how different variables and characteristics of a person and a property influence if they get a loan or not. Therefore:

1. Loan_Status: This parameter should exist because it determines if somebody will get the loan or not (Response Variable).
2. ApplicantIncome: Since this is a loan, the loan giver wants to know if the person getting the loan has the ability to pay their debts.
3. CoapplicantIncome: If an application has more than one applicant, then the other person is also responsible for paying their loans. So their income, which gives a sense of the ability to pay back the loan, is an important variable.
4. LoanAmount: The amount of the loan is absolutely of importance because depending on this number, and the applicant income, and … the decision on whether the person get the loan or not varies.
5. Loan_Amount_Term: This variable is chosen because in relation to the other variables, this variable can show if a person can actually pay their debts or not.
6. Property_Area: Depending on the property area, the amount of loan can change.

Now I have separated the plots based on property type, because the loans for each area can be different from the other loans, as the property price changes.

1. The first thing that is observant from this plot is the correlation between loan amount and the applicant's income, and it is shown that this correlation is statistically significant. This was expected, since people try to request a loan they can pay back. It can also be seen the the co applicant's income is important for the loan amount, but by much smaller amount, This is because the primary applicant is the the first person responsible to pay the loans.
2. The Property Area has little effect on the loan status (approval or rejection), as the amount of loans approved or rejected are almost the same for all property types (the area of the red, green, and blue

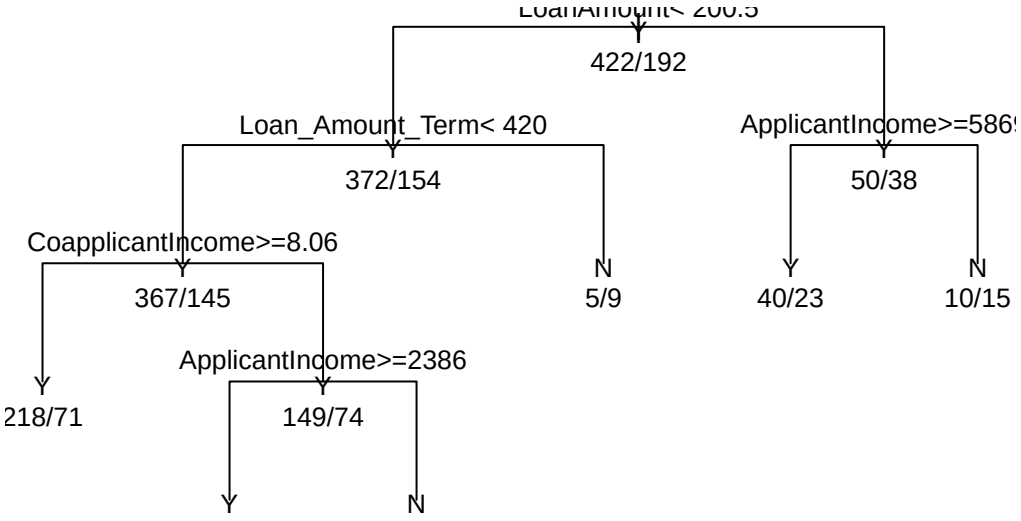
boxes), with the exception that semiurbun properties got approved more than the other two.

3. Most people requested lower amounts of loans, as visible from the normal shaped distribution for loan amount as the pick is closer to lower amounts.
4. Also, comparing loan status and applicant income, we can see that the people who got a "Yes" on their loan application have incomes more than the 3/2 of all incomes (excluding outliers). This is a very subtle relationship, as the income of the applicants that got Yes or No are about the same, and we can see that one applicant that had the highest salaries was rejected on their loan application.
5. It is easy to observe that most people choose 360 months as their Loan_Amount_Term, regardless of the Loan Amount.

These were only a few things that we could conclude from this plot.

Question 4: Build a classification tree of one of the six variables from the last part as a function of the other five, and any other explanatory variables you think are necessary. Show code, explain reasoning, and show the tree as a simple (ugly) plot. Show the confusion matrix. Give two example predictions and follow them down the tree. (Mandatory)

```
library(rpart)
com2 = com

com2$Loan_Status <- factor(com2$Loan_Status, levels=c("Y", "N"))
com2$Property_Area <- factor(com2$Property_Area, levels=c("Urbun", "Rural","Semiurbun"))

fit = rpart(Loan_Status ~ ApplicantIncome + CoapplicantIncome + LoanAmount +
              Loan_Amount_Term + Property_Area , data=com2)

plot(fit, uniform=TRUE)
text(fit, use.n=TRUE, all=TRUE, cex=0.8)
```



Out of 614 cases, 422 got the loan and 192 did not. It first asks if the loan amount is below 200.5, if yes, we go to the left. Since most of the loans are less than 200.5, it is obvious we have more branches in the left. If the amount of the loan is bigger than 200.5, the tree suddenly asks:"Is the applicant's income higher than 5869?", if yes, give them the loan. 40 people out of 63 that had an income higher than 5869 got the loan, 23 didn't. As we saw previously, the property area did not have much effect on the loan decision. Now going to the left of the tree, the next question is whether the loan amount term is less than 420 months. If it's more, the loan request will probably be rejected.

Next, it asks if the loan term is less than 420 months, is also asks if the coapplicant's income more than 8 dollars?! which is basically asking if the co applicant has any income at all or not. And if the co applicant has less than 8 dollars income, does the primary applicant have at least 2386 dollars income? If yes, the loan will be given to them. It is interesting that the question of "applicant's income" was one of the very last questions.

Now let's follow two examples from the test set and see of they get approved for a loan.

```
test[200,]
```

```
##      Loan_ID Gender Married Dependents Education Self_Employed ApplicantIncome
## 200 LP002102   Male     Yes          0  Graduate           Yes             1900
##     CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History Property_Area
## 200              1442         88              360              1         Rural
```

Starting from top of the tree, is the loan amount less than 200? In this case it's 88, so yes, and we go to the left of the tree. Next, asking if the loan amount term is less than 420 months. The asnwer is yes because it's 360 months. Moving forward to the left. The co applicant's is 1442, so yes it's larger than 8, and hence, this person receives the loan.

Another example:

```
test[251,]
```

```
##      Loan_ID Gender Married Dependents Education Self_Employed ApplicantIncome
## 251 LP002383   Male     Yes         3+  Graduate            No            3242
##     CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History Property_Area
## 251               437        142              480              0         Urban
```

Again starting from top of the tree, is the loan amount less than 200? In this case it's 142, so yes, and we go to the left of the tree. Next, asking if the loan amount term is less than 420 months. The answer is no because it's 480 months. Moving forward to the right, this person will not receive the loan.

Finally the confusion matrix:

```
library(caret)
```

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

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

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
train$Loan_Status<- factor(train$Loan_Status, levels=c("Y", "N"))
train$Property_Area <- factor(train$Property_Area, levels=c("Urbun", "Rural","Semiurbun"))
# predict using the train set
fit2 = rpart(Loan_Status ~ ApplicantIncome + CoapplicantIncome + LoanAmount +
             Loan_Amount_Term + Property_Area , data=com2)
train$predicted <- predict(fit2, train, type="class")

# create confusion matrix
confusionMatrix(train$predicted, train$Loan_Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   Y   N
##          Y 405 163
##          N  17  29
##
##                Accuracy : 0.7068
##                  95% CI : (0.6691, 0.7426)
##     No Information Rate : 0.6873
##     P-Value [Acc > NIR] : 0.1584
##
##                   Kappa : 0.1397
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9597
##             Specificity : 0.1510
```

```
##          Pos Pred Value : 0.7130
##          Neg Pred Value : 0.6304
##              Prevalence : 0.6873
##          Detection Rate : 0.6596
##    Detection Prevalence : 0.9251
##       Balanced Accuracy : 0.5554
##
##        'Positive' Class : Y
##
```

Question: Build another model using one of the continuous variables from your six most important. This time use your model selection and dimension reduction tools, and include at least one non-linear term. (Mandatory)

Let's see if we can find the Applicant's income from the other variables. Here, I change Property_Area for Education because in predicting Income, Education is a better indicator.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

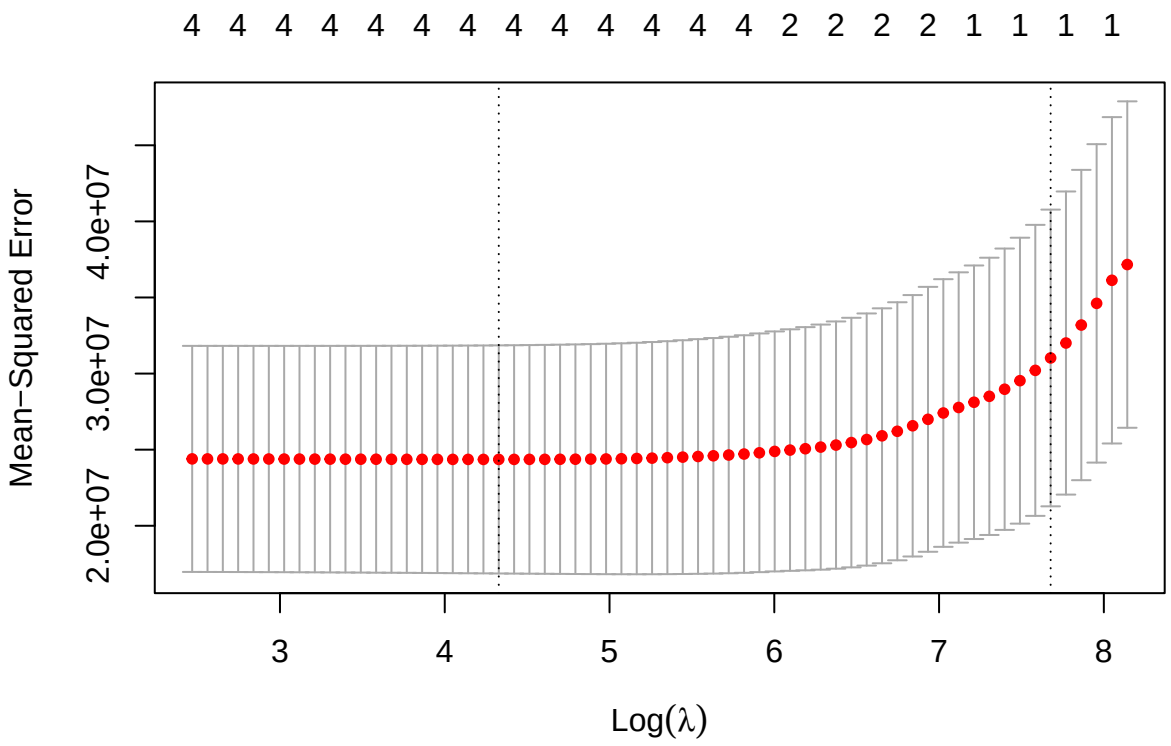
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-6
```

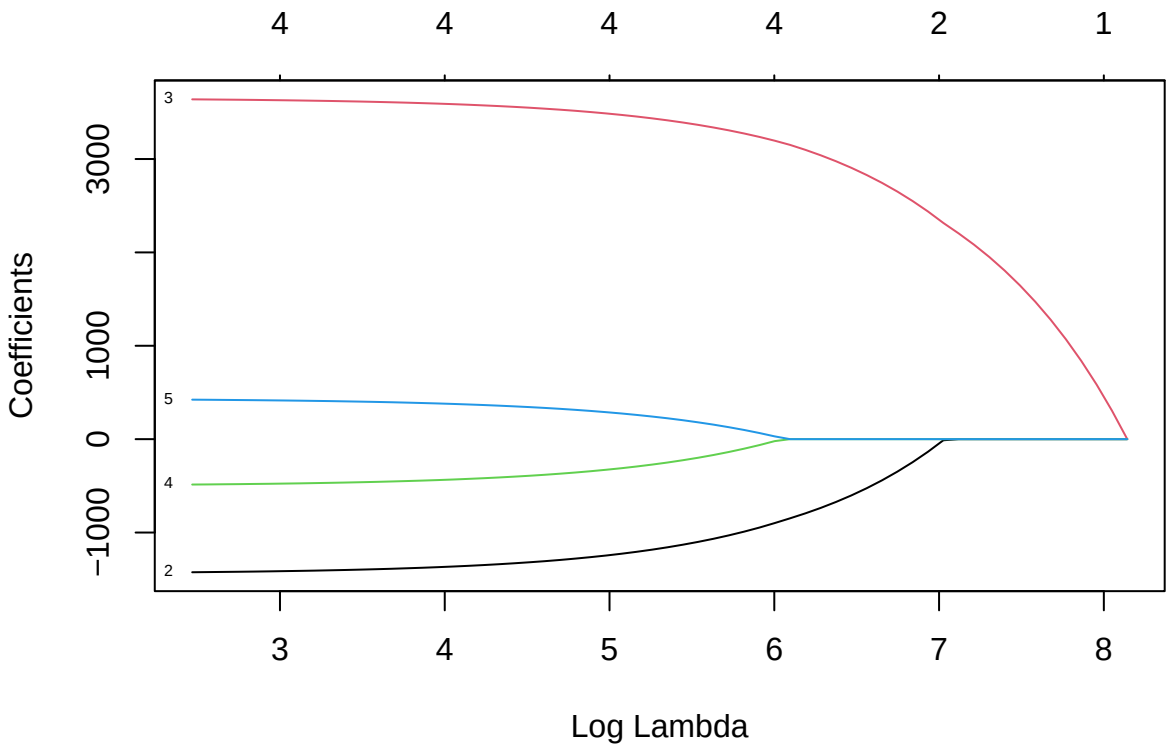
```
new_df <- com

#Changing loan_status: 1 for Y and 0 for N
new_df$binary <- ifelse(new_df$Loan_Status == 'Y', 1, 0)

#Changing Education: 1 for Graduated and 0 for Nor Graduated
new_df$grad <- ifelse(new_df$Education == 'Graduate', 1, 0)

x <- as.matrix(new_df[,c('binary','CoapplicantIncome',
                         'LoanAmount','Loan_Amount_Term','grad')])
y <- as.numeric(new_df$ApplicantIncome)

#Scaling all the variables:
x_norm <- scale(x)
#y_norm <- scale(y)

#Finally the model
cv_model <- cv.glmnet(x_norm, y, alpha = 1)

#Finding the minimum lambda (Penalty Parameter)
best_lambda <- cv_model$lambda.min
best_lambda
```

```
## [1] 75.78432
```

```
plot(cv_model)
```




```
plot(cv_model$glmnet.fit, "lambda", label=TRUE)
```



Now constructing the best model:

```
best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
coef(best_model)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                              s0
## (Intercept)       1533.3737974
## binary               .
## CoapplicantIncome   -0.4577511
## LoanAmount          41.7178244
## Loan_Amount_Term    -6.3698817
## grad               867.8992427
```

This shows that the best indicator of a person's income is their education status, and after that, from the amount of the loan they requested we can understand their income. As we also saw from the decision tree, the loan status and the applican't income do not have a meaningful relationship.

If we want a denser model, we can choose a lower lambda, and omit Coapplicantincome and Loan_amount_term from the variables.

Question: Discuss briefly any ethical concerns like residual disclosure that might arise from the use of your data set, possibly in combination with some additional data outside your dataset. (Option)

One the concerns this data set had was that with a little information out of this data set, you can actually pinpoint how these people are. For example, With the marriage status, number of dependents, education, and property type, plus the city that these people are applying from can narrow it down to one person. If this information is breached, their account history can be used for fraud, because some information on credit history is available. Furthermore, this information could be used to find out how much one applicant and the co applicant make, which is normally private information. Coupled with some other data set, many confidential information can be revealed.