Project Part1: EDA

Code

Aaron Black, Thenmozhi Boopathy

2025-09-23

Introduction

This HTML file contains the data cleaning methods we have utilized to better predict whether someone will default on their loan or not based on a data set containing 237730 observations.

1. Data Overview and Summary Statistics

First, we are going to load the dataset into our RMarkdown file and get some basic information about our variables. This loaded data will be cleaned to help us eventually use the model for our holdout data. Our target variable is loan_default, meaning whether or not someone will default on a loan. Below is our starting code for the analysis.

Hide

```
#Read in the data
df = read.csv("train.csv", stringsAsFactors = TRUE)
str(df)
```

```
## 'data.frame':    237730 obs. of  36 variables:
##  $ loan_default          : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 1 2 1 ...
##  $ loan_amnt             : int  15175 7500 12000 12000 18200 7000 17775 21100 5000 2100
0 ...
##  $ term                  : Factor w/ 2 levels " 36 months"," 60 months": 1 1 2 1 1 1 1
2 1 1 ...
##  $ int_rate              : num  16.55 11.99 24.5 13.05 7.26 ...
##  $ installment           : num  538 249 349 405 564 ...
##  $ grade                 : Factor w/ 7 levels "A","B","C","D",..: 4 2 6 2 1 3 4 4 1 1
...
##  $ sub_grade             : Factor w/ 35 levels "A1","A2","A3",..: 17 8 28 10 4 11 18 20
4 4 ...
##  $ emp_title             : Factor w/ 112646 levels "","       NSA Industries llc",..: 5
8857 77473 78838 110463 62900 85034 99520 89406 66950 16560 ...
##  $ emp_length            : Factor w/ 12 levels "","< 1 year",..: 12 9 7 5 11 2 4 11 8 7
...
##  $ home_ownership        : Factor w/ 5 levels "MORTGAGE","NONE",..: 1 5 1 5 1 1 4 5 1 1
...
##  $ annual_inc            : num  33000 40000 70000 85000 78000 69000 72000 44000 35000 8
2000 ...
##  $ verification_status   : Factor w/ 3 levels "Not Verified",..: 2 1 2 3 2 2 3 2 1 1
...
##  $ issue_d               : Factor w/ 115 levels "Apr-2008","Apr-2009",..: 104 26 45 16
84 9 28 114 15 44 ...
##  $ purpose               : Factor w/ 14 levels "car","credit_card",..: 3 2 3 3 3 3 2 2
5 2 ...
##  $ title                 : Factor w/ 31659 levels "","'08 Rehab",..: 9604 8312 6855 763
1 9604 9604 7866 7866 22155 26858 ...
##  $ dti                   : num  23.6 29.5 18.5 33.4 14.3 ...
##  $ earliest_cr_line      : Factor w/ 659 levels "Apr-1955","Apr-1958",..: 652 648 200 4
72 92 647 154 593 479 321 ...
##  $ open_acc              : int  11 11 12 21 9 12 14 10 6 12 ...
##  $ pub_rec               : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ revol_bal             : num  11376 7113 11699 21133 11304 ...
##  $ revol_util            : num  46.1 56.5 83.6 58.7 62.8 44.7 50 46.6 40.9 69.7 ...
##  $ total_acc             : int  24 23 48 35 22 37 26 23 14 30 ...
##  $ initial_list_status   : Factor w/ 2 levels "f","w": 1 2 2 1 2 2 1 1 1 1 ...
##  $ application_type      : Factor w/ 3 levels "DIRECT_PAY","INDIVIDUAL",..: 2 2 2 2 2 2
2 2 2 2 ...
##  $ mort_acc              : int  2 0 6 2 3 0 0 0 1 1 ...
##  $ pub_rec_bankruptcies  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ address               : Factor w/ 236733 levels "000 Adam Station Apt. 329\nAshleybe
rg, AZ 22690",..: 76203 133532 44243 79606 222835 230929 193203 180306 174508 57888 ...
##  $ delinq_2yrs           : int  4 0 0 1 0 0 0 0 0 0 ...
##  $ fico_range_low        : int  680 680 665 665 695 670 690 705 715 695 ...
##  $ fico_range_high       : int  684 684 669 669 699 674 694 709 719 699 ...
##  $ inq_last_6mths        : int  1 0 3 0 0 1 0 1 0 0 ...
##  $ mths_since_last_delinq: int  5 43 69 20 53 59 NA NA NA 74 ...
##  $ last_credit_pull_d    : Factor w/ 137 levels "","Apr-2009",..: 11 124 45 103 94 94 3
3 126 45 94 ...
##  $ acc_now_delinq        : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ hardship_flag         : Factor w/ 1 level "N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ debt_settlement_flag  : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...
```

Notice how there are 18 character variables and 18 numeric variables. We will now split these data structures up to get more specific visuals with both structures.

	loan_amnt <int>	int_rate <dbl>	installment <dbl>	annual_inc <dbl>	dti <dbl>	open_… <int>	pub_r… <int>	revol_bal <dbl>	revo
1	15175	16.55	537.64	33000	23.56	11	0	11376	
2	7500	11.99	249.08	40000	29.49	11	0	7113	
3	12000	24.50	348.71	70000	18.50	12	0	11699	
4	12000	13.05	404.62	85000	33.36	21	0	21133	
5	18200	7.26	564.13	78000	14.28	9	0	11304	
6	7000	11.99	232.47	69000	18.63	12	1	12056	

6 rows | 1-10 of 19 columns

	loan_default <fct>	term <fct>	gr… <fct>	sub_grade <fct>	emp_title <fct>	emp_length <fct>	home_owr <fct>
1	No	36 months	D	D2	Maintenance	9 years	MORTGAG
2	Yes	36 months	B	B3	Professional	6 years	RENT
3	Yes	60 months	F	F3	Purchasing	4 years	MORTGAG
4	No	36 months	B	B5	Wells Fargo	2 years	RENT
5	No	36 months	A	A4	Merchandise Manager	8 years	MORTGAG
6	No	36 months	C	C1	Sales Manager	< 1 year	MORTGAG

6 rows | 1-8 of 19 columns

The data successfully split. Now, we can utilize different visualization methods to identify various anomalies, outliers, missing values, and where we can change the variable type to make our analysis the best it can be. Let's look at some preliminary observations.

Hide

```
#Summary Statistics for numeric Variables
summary(bank_numeric)
```

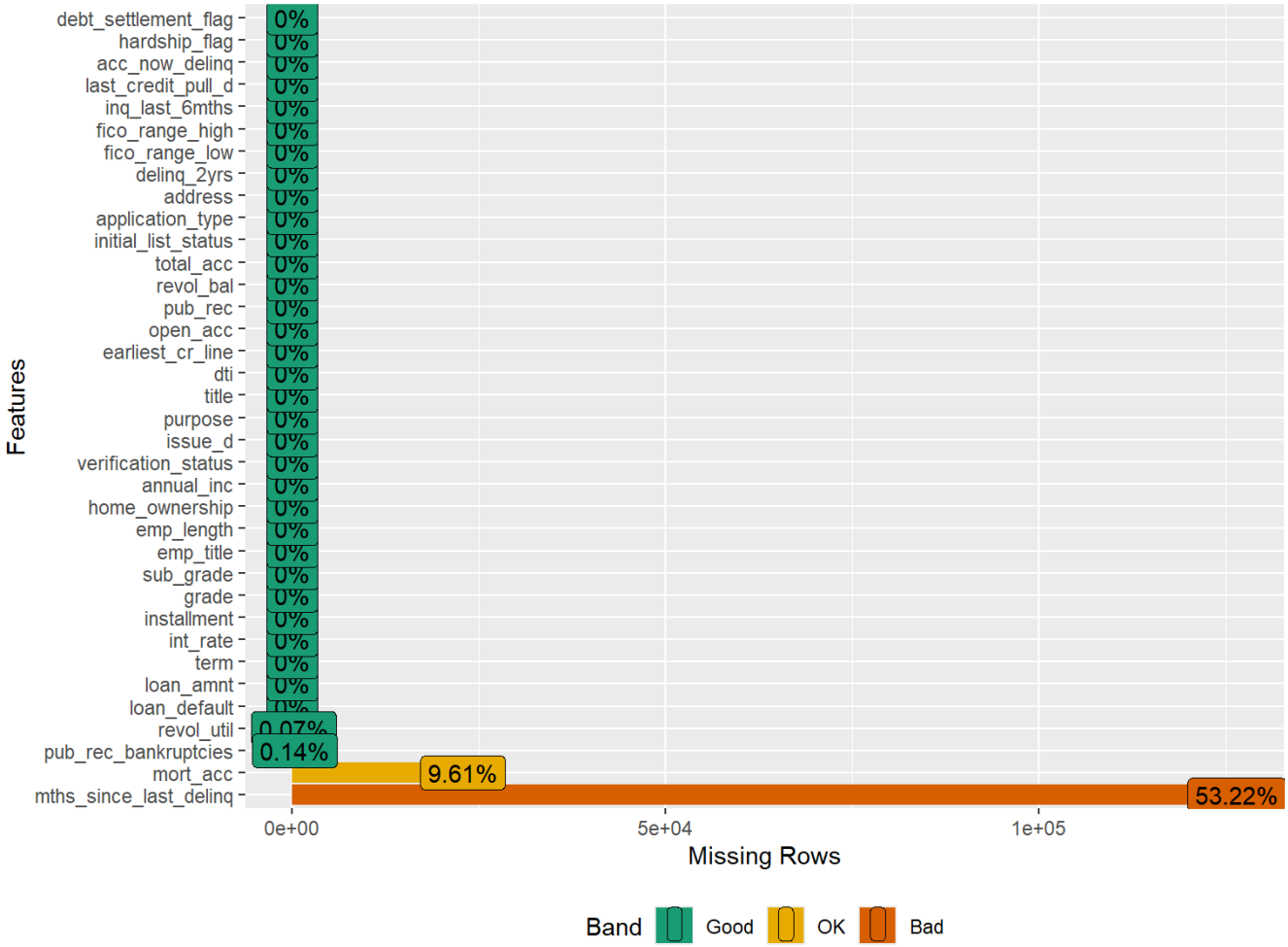
```
##    loan_amnt        int_rate      installment        annual_inc
##  Min.   :  500   Min.   : 5.32   Min.   :  16.25   Min.   :      0
##  1st Qu.: 8000   1st Qu.:10.39   1st Qu.: 250.33   1st Qu.:  45000
##  Median :12000   Median :13.33   Median : 375.49   Median :  64000
##  Mean   :14107   Mean   :13.63   Mean   : 431.69   Mean   :  74248
##  3rd Qu.:20000   3rd Qu.:16.49   3rd Qu.: 567.34   3rd Qu.:  90000
##  Max.   :40000   Max.   :30.99   Max.   :1533.81   Max.   :8706582
##
##       dti             open_acc        pub_rec          revol_bal
##  Min.   :   0.00   Min.   : 0.00   Min.   : 0.0000   Min.   :      0
##  1st Qu.:  11.29   1st Qu.: 8.00   1st Qu.: 0.0000   1st Qu.:   6015
##  Median :  16.90   Median :10.00   Median : 0.0000   Median :  11182
##  Mean   :  17.40   Mean   :11.31   Mean   : 0.1775   Mean   :  15786
##  3rd Qu.:  23.00   3rd Qu.:14.00   3rd Qu.: 0.0000   3rd Qu.:  19619
##  Max.   :9999.00   Max.   :90.00   Max.   :86.0000   Max.   :1298783
##
##    revol_util       total_acc         mort_acc     pub_rec_bankruptcies
##  Min.   :  0.00   Min.   :  2.00   Min.   : 0.00   Min.   :0.0000
##  1st Qu.: 35.80   1st Qu.: 17.00   1st Qu.: 0.00   1st Qu.:0.0000
##  Median : 54.70   Median : 24.00   Median : 1.00   Median :0.0000
##  Mean   : 53.75   Mean   : 25.41   Mean   : 1.81   Mean   :0.1211
##  3rd Qu.: 72.80   3rd Qu.: 32.00   3rd Qu.: 3.00   3rd Qu.:0.0000
##  Max.   :892.30   Max.   :150.00   Max.   :32.00   Max.   :8.0000
##  NA's   :177                       NA's   :22854   NA's   :336
##   delinq_2yrs      fico_range_low  fico_range_high inq_last_6mths
##  Min.   : 0.0000   Min.   :630.0   Min.   :634.0   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.:670.0   1st Qu.:674.0   1st Qu.:0.0000
##  Median : 0.0000   Median :690.0   Median :694.0   Median :0.0000
##  Mean   : 0.2824   Mean   :696.6   Mean   :700.6   Mean   :0.7796
##  3rd Qu.: 0.0000   3rd Qu.:710.0   3rd Qu.:714.0   3rd Qu.:1.0000
##  Max.   :29.0000   Max.   :845.0   Max.   :850.0   Max.   :8.0000
##
##  mths_since_last_delinq acc_now_delinq
##  Min.   :  0.00         Min.   :0.00000
##  1st Qu.: 16.00         1st Qu.:0.00000
##  Median : 32.00         Median :0.00000
##  Mean   : 34.71         Mean   :0.00424
##  3rd Qu.: 51.00         3rd Qu.:0.00000
##  Max.   :159.00         Max.   :6.00000
##  NA's   :126517
```

Each variable gives important values such as the mean and median with the min and max values helping identify outliers. We can also find early indicators of numeric variables that could potentially be manipulated into factor variables or factor lumped to help our prediction model.

Let’s remember to note the missing values:

Hide

```
plot_missing(df)
```



- revol_util = 177 missing values
- mort_acc = 22854 missing values
- pub__rec_bankruptcies = 336 missing values
- mths_since_last_delinq = 126517 missing

Hide

```
# Look at structure and ensure these are all factor variables
str(bank_factor)
```

```
## 'data.frame':    237730 obs. of  18 variables:
##  $ loan_default        : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 1 2 1 ...
##  $ term                : Factor w/ 2 levels " 36 months"," 60 months": 1 1 2 1 1 1 1 2
1 1 ...
##  $ grade               : Factor w/ 7 levels "A","B","C","D",..: 4 2 6 2 1 3 4 4 1 1 ...
##  $ sub_grade           : Factor w/ 35 levels "A1","A2","A3",..: 17 8 28 10 4 11 18 20 4
4 ...
##  $ emp_title           : Factor w/ 112646 levels "","       NSA Industries llc",..: 588
57 77473 78838 110463 62900 85034 99520 89406 66950 16560 ...
##  $ emp_length          : Factor w/ 12 levels "","< 1 year",..: 12 9 7 5 11 2 4 11 8 7
...
##  $ home_ownership      : Factor w/ 5 levels "MORTGAGE","NONE",..: 1 5 1 5 1 1 4 5 1 1
...
##  $ verification_status : Factor w/ 3 levels "Not Verified",..: 2 1 2 3 2 2 3 2 1 1 ...
##  $ issue_d             : Factor w/ 115 levels "Apr-2008","Apr-2009",..: 104 26 45 16 84
9 28 114 15 44 ...
##  $ purpose             : Factor w/ 14 levels "car","credit_card",..: 3 2 3 3 3 3 2 2 5
2 ...
##  $ title               : Factor w/ 31659 levels "","'08 Rehab",..: 9604 8312 6855 7631
9604 9604 7866 7866 22155 26858 ...
##  $ earliest_cr_line    : Factor w/ 659 levels "Apr-1955","Apr-1958",..: 652 648 200 472
92 647 154 593 479 321 ...
##  $ initial_list_status : Factor w/ 2 levels "f","w": 1 2 2 1 2 2 1 1 1 1 ...
##  $ application_type    : Factor w/ 3 levels "DIRECT_PAY","INDIVIDUAL",..: 2 2 2 2 2 2 2
2 2 2 ...
##  $ address             : Factor w/ 236733 levels "000 Adam Station Apt. 329\nAshleyber
g, AZ 22690",..: 76203 133532 44243 79606 222835 230929 193203 180306 174508 57888 ...
##  $ last_credit_pull_d  : Factor w/ 137 levels "","Apr-2009",..: 11 124 45 103 94 94 33
126 45 94 ...
##  $ hardship_flag       : Factor w/ 1 level "N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ debt_settlement_flag: Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...
```

As we can see, we have successfully identified our factor variables. Making a table wouldn't be useful in this case because certain variables have a significant amount of levels such as address, title, and emp_tile. Please note that we will make changes to the data in our main dataframe, we are just using the split datasets to look deeper into the data.

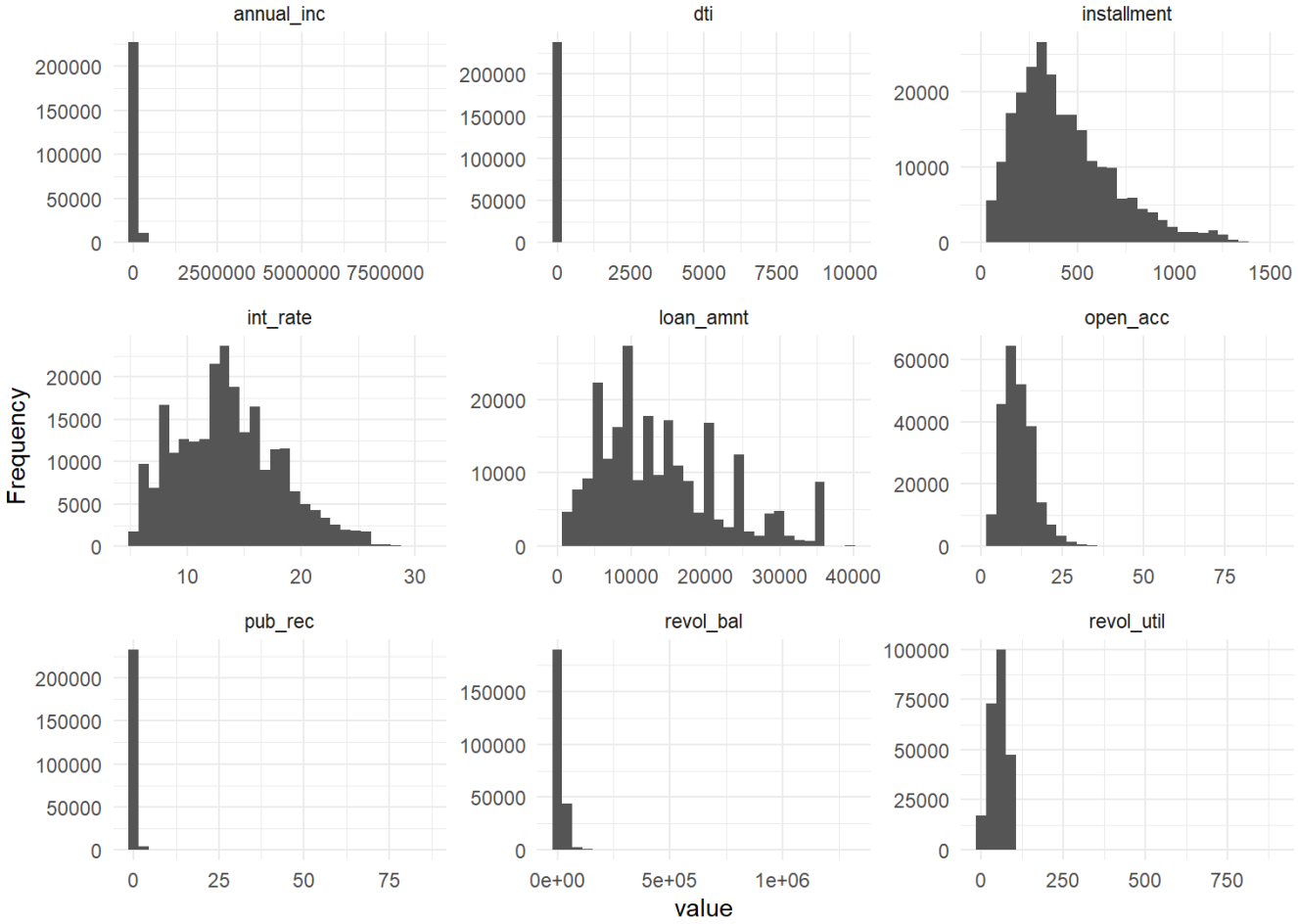
2. Data visualization

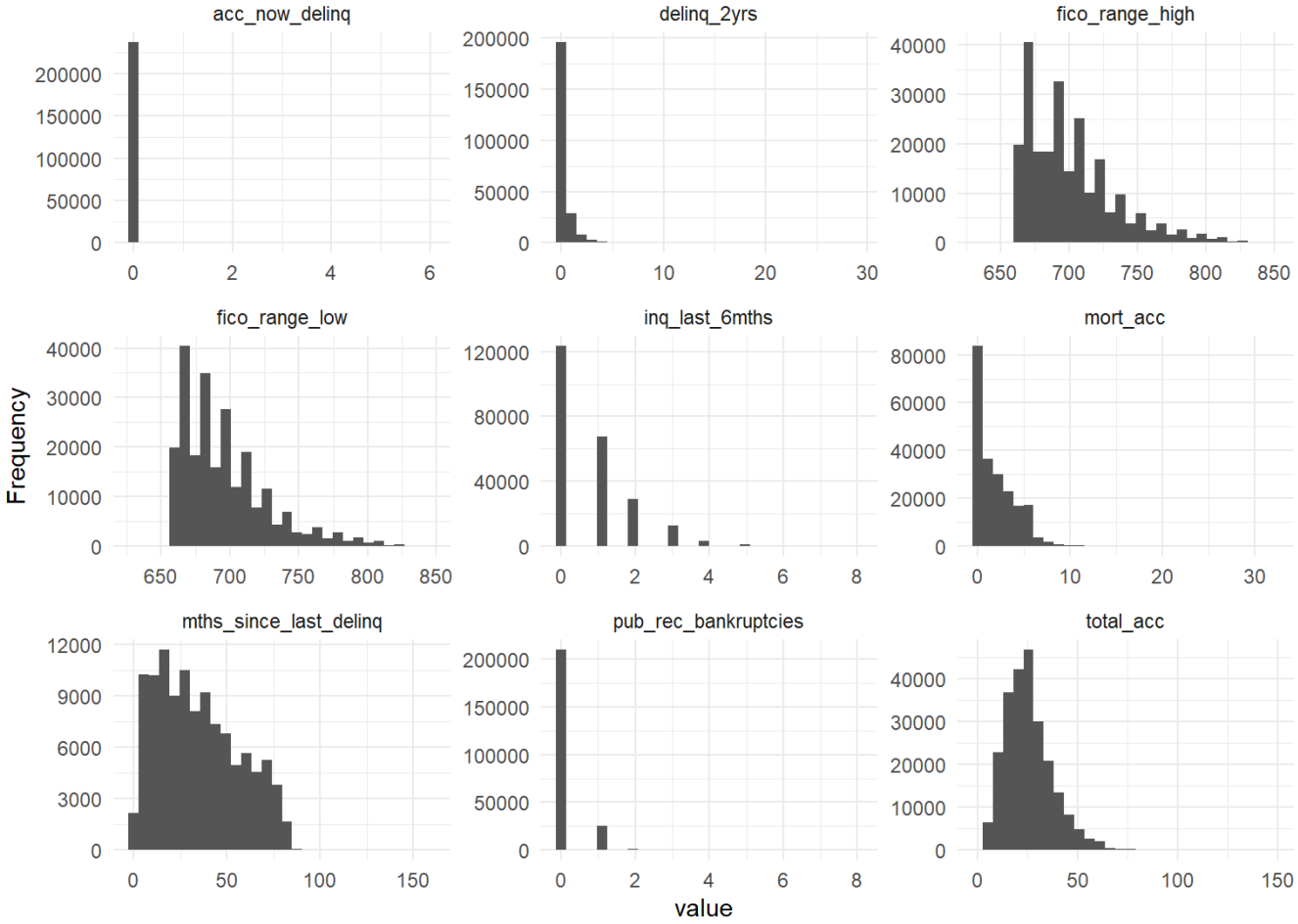
Now, we can dig further into our data and identify what we need to do with each specific variable based on our output. It is important to figure out whether to remove, alter, or keep each specific variables so we can have the best data to eventually train a champion model.

Hide

```
# Histograms for numeric variables

DataExplorer::plot_histogram(bank_numeric,
                             ggtheme=theme_minimal(),
                             ncol=3,
                             nrow=3)
```


annual_inc
dti
installment
int_rate
loan_amnt
open_acc
pub_rec
revol_bal
revol_util
Frequency
value
0
2500000
5000000
7500000
2500
5000
7500
10000
500
1000
1500
10
20
30
10000
20000
30000
40000
25
50
75
0e+00
5e+05
1e+06
250
500
750

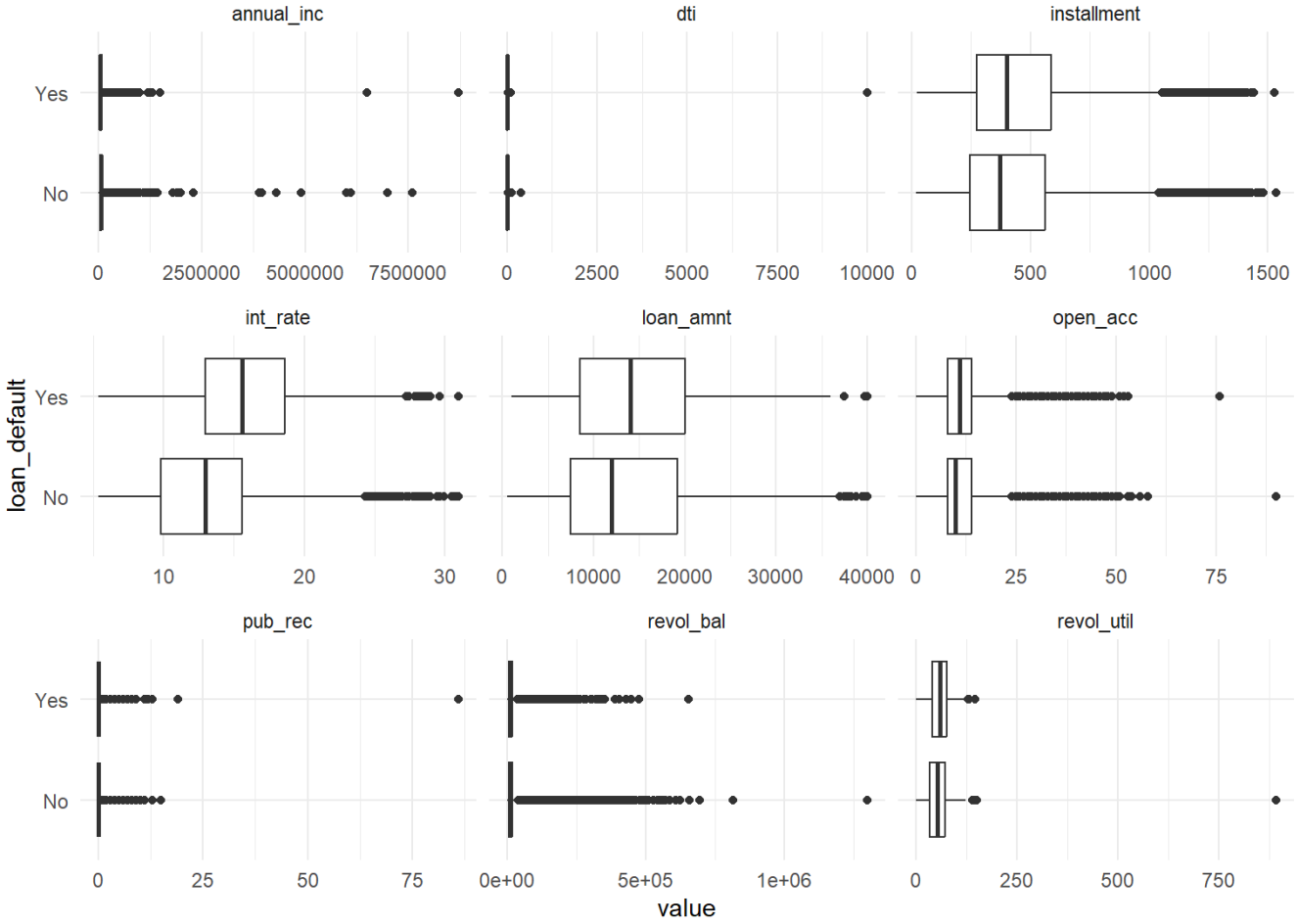


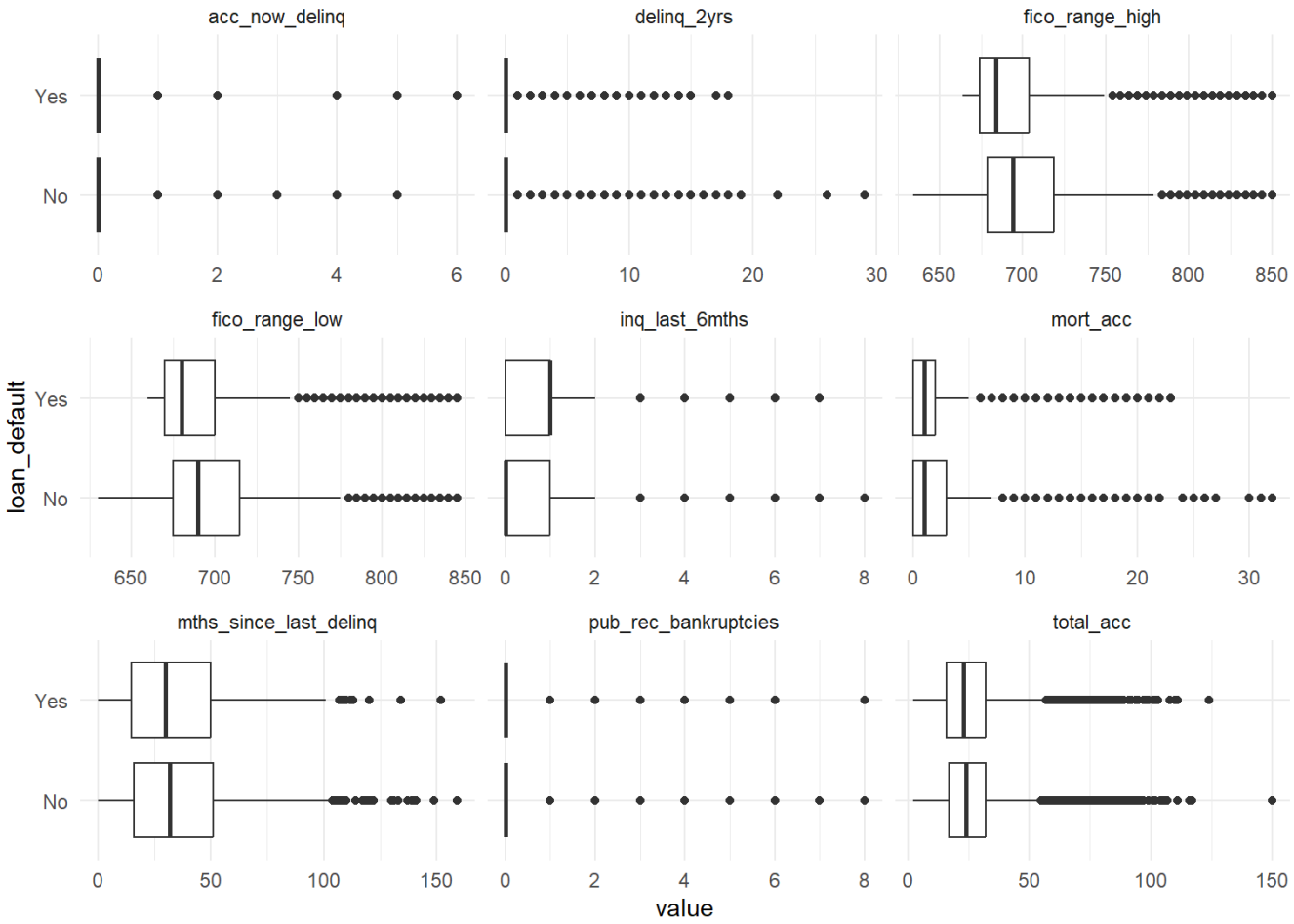
These variables can easily be seen for their skew, symmetry, and range. If the range is large from the models, like pub_rec, revol_bal, dti, and annual_inc, this indicates outliers in the data.

Lets look at the box plots for the numeric variables to confirm our suspicions on outliers and see if there are important variables to keep in our dataset based on loan_default.

Hide

```
# Boxplots based off the loan defaulting or not
bank_numeric$loan_default <- df$loan_default
DataExplorer::plot_boxplot(bank_numeric, by="loan_default",
                           ggtheme = theme_minimal(),
                          ncol=3,
                          nrow=3)
```

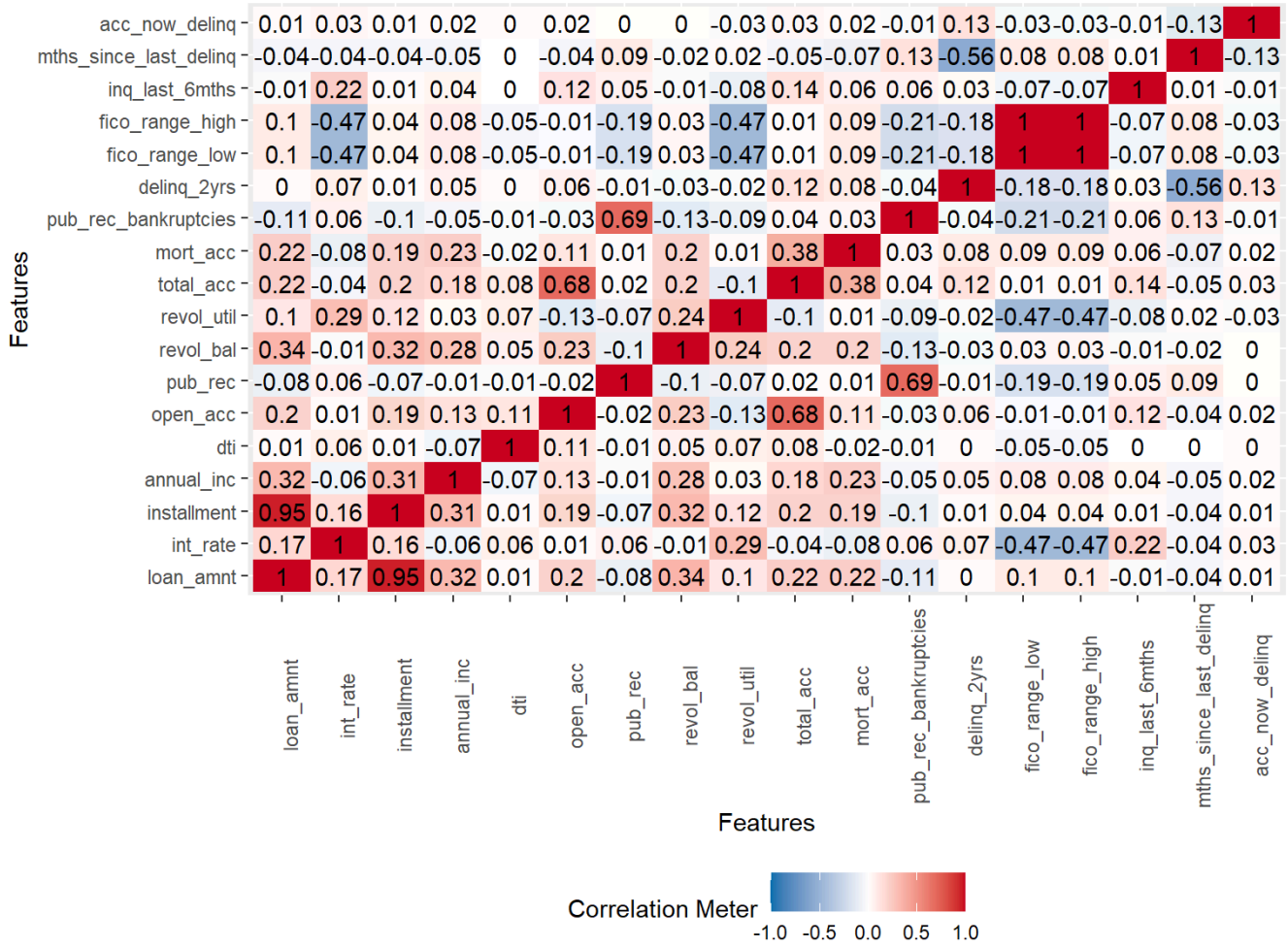

annual_inc
dti
installment
int_rate
loan_amnt
open_acc
pub_rec
revol_bal
revol_util
loan_default
Yes
No
0
2500000
5000000
7500000
2500
5000
7500
10000
500
1000
1500
10
20
30
10000
20000
30000
40000
25
50
75
0e+00
5e+05
1e+06
250
500
750
value



Based on these results, we can confirm outliers in the data and deal with them accordingly. Also, we see that these numeric variables can have correlation to one defaulting on a loan. Lets look at a heat map to finalize some of the cleaning techniques we will perform.

Hide

```
# Create a heat map to check correlation between variables
DataExplorer::plot_correlation(bank_numeric,
                               type="continuous",
                               cor_args = list("use" = "pairwise.complete.obs")
                               )
```



Total_acc and open_acc has correlated metrics with an R^2 of .68, while pub_rec and pub_rec_bankrupcies have correlated features with an R^2 of 0.68. Installment has an extremely high correlation to loan amount, showing we can remove this variable.

After using some visualization techniques to look at our numeric variables, we have listed below our cleaning strategys for each one.

Keep the Same

These 7 variables below have key indicators that will help us predict loan_default based on the summary statistics, boxplot, heatmap, and bar plot visualizations.

- loan_amnt
- int_rate
- annual_inc
- open_acc
- total_acc
- fico_range_low
- fico_range_high
- inq_last_6mnths

Change and Recode

These numeric variables were flagged as having anomalies, outliers, or ranges that could be manipulated to better suit our model.

- dti: There is one outlier. We can figure out how to remove this extreme value.
- pub_rec: One outlier that we must examine.
- revol_bal: One outlier that we must examine.
- revol_util: Contains a massive outlier and missing values, we must analyze further.
- mort_acc: Many null values. We will analyze more in depth in the upcoming sections.
- pub_rec_bankruptcies: Null values will be further analyzed.

Variables we are Removing

- acc_now_delinq: We are removing this variable because most of the values are zero and it wouldn't give us any special indication of loan default.
- mths_since_last_delinq: We will remove this variable because there are too many null values(53.22%).Most people probably don't have any delinquency, so this variable can be disregarded.
- delinq_2yrs: Most of the values are zero. but we can also just get rid of it.
- installment: We are removing this variable because of its high correlation (0.95) to loan amount, meaning they provide the same information.

Hide

```
# Remove these variables
df<-df%>%
  select(-installment, -acc_now_delinq, -mths_since_last_delinq, -delinq_2yrs)
dim(df)
```

```
## [1] 237730     32
```

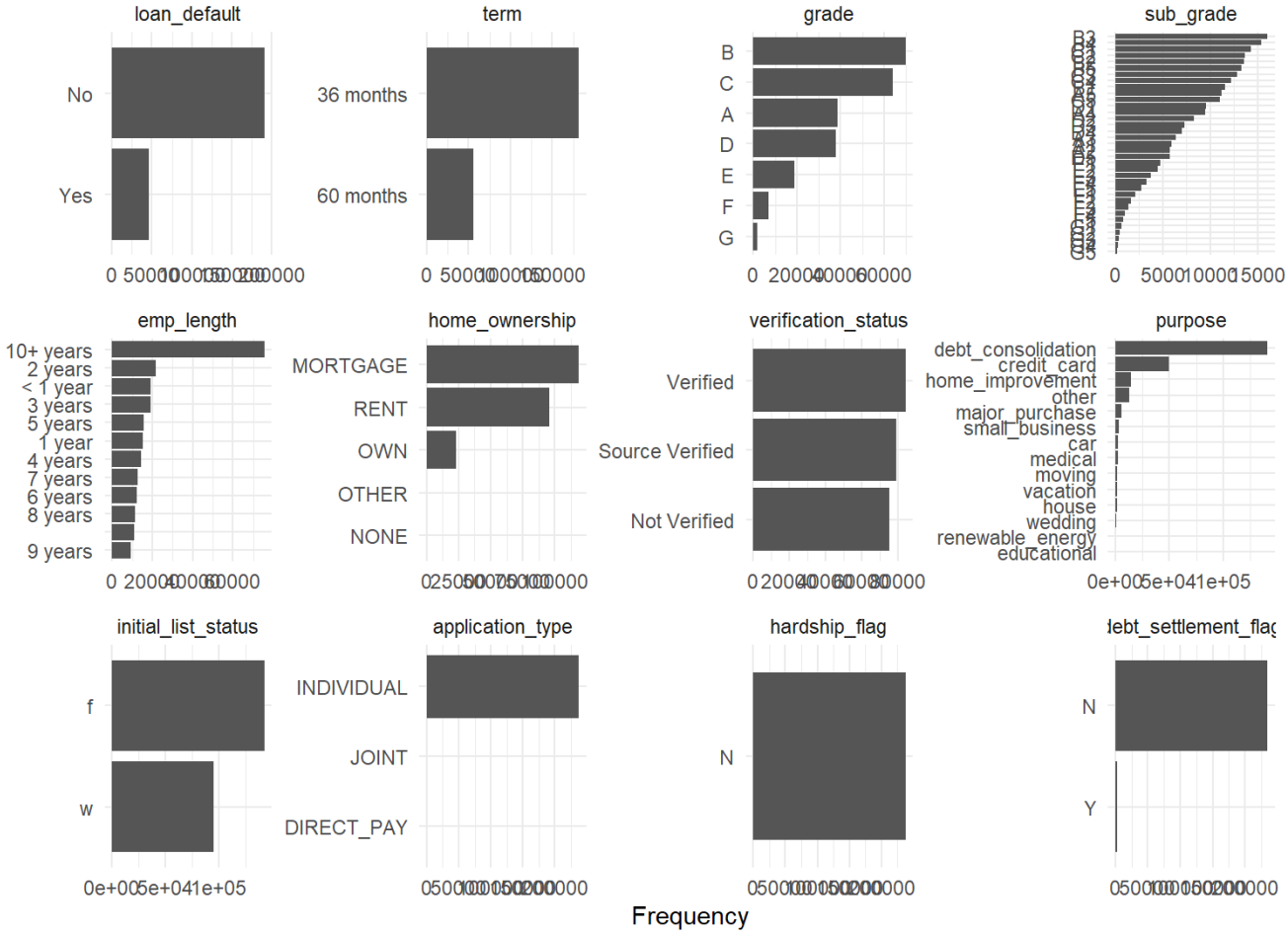
We now have only 32 variables in our original data frame. (36-4=32). None of these variables show major distribution between the 'yes' and 'no decision of a loan default based on the boxplot and amount of null values.

Cateogrical Variables:

Now, we will look at the factor variables and decide where to make changes based on the prediction power and levels. We will also recode and lump variables together if there are too many levels to the data or remove when there's too many levels in itself.

Hide

```
# Look at structure and levels for categorical variables
DataExplorer::plot_bar(bank_factor,
                       ggtheme = theme_minimal(),
                       ncol=4,
                       nrow=5)
```



There are 12 variables shown here. Notice how some variables are not included in the visual, such as emp_title, issue_d, earliest_cr_line, address, and last_credit_pull_d. These variables all have a significant amount of levels, meaning they could either be manipulated to a numeric variable or be removed because there would way too many dummy variables. Even some of the bar plots have too many levels that will need to be analyzed further later in our analysis.

Now, we will identify what we are doing with each variable. For the categorical variables we are keeping, we will specify changing the leels or leaving the data as it is.

Keep the Same

- Term: 2 levels, we will keep the same
- Grade: 7 levels, keep the same
- Verification_status: 2 levels, Keep the same
- Initial_list_status: 2 levels, Keep the same
- Debt_settlement_flag: a “yes” decision has a high correlation to “yes” to loan_default, keep the same

Change and Recode

- Emp_length: Use bins to cluster years together
- Home_ownership: Remove the “Other” and “None” variable
- Purpose: Keep “Debt consolidation,” cluster the rest into “Other”

Variables we are Removing

- Emp_title: Remove (too many levels)
- Issue_d: Remove (No significance to loan_default_

- Title: Remove (Too many levels)
- Application_type: Remove (Vast majority of values in “individual” value)
- Hardship_flag: Remove (only 1 level)
- Earliest_cr_line: Dates do not seem significant in predicting loan default
- Last_credit_pull_d: Dates do not seem significant in predicting loan default
- Address (zip code): Discriminatory variable, not significant in explaining loan default
- sub_grade: Has the same properties as grade, not needed in our model

Hide

```
# Remove variables that are not seen as needed
df<-df%>%
  select(-emp_title, -issue_d, -title, -application_type, -hardship_flag, -earliest_cr_lin
e, -last_credit_pull_d, -address, -sub_grade)
dim(df)
```

```
## [1] 237730     23
```

We now have 23 Variables in our model (32-8=23)

3. Handling Missing Data

Earlier, we identified some of the missing values in different variables. Although we removed the mnths_since_last_delinq variable, we still have three variables that are missing data:

Hide

```
#Checking missing Values
colSums(is.na(df))
```

```
##         loan_default            loan_amnt                 term
##                    0                    0                    0
##             int_rate                grade           emp_length
##                    0                    0                    0
##       home_ownership           annual_inc  verification_status
##                    0                    0                    0
##              purpose                  dti             open_acc
##                    0                    0                    0
##              pub_rec            revol_bal           revol_util
##                    0                    0                  177
##            total_acc  initial_list_status             mort_acc
##                    0                    0                22854
## pub_rec_bankruptcies       fico_range_low      fico_range_high
##                  336                    0                    0
##       inq_last_6mths debt_settlement_flag
##                    0                    0
```

1. mort_acc

There are 22854 missing values in the # of mortgage accounts row. In this case, we can assume that anyone who didn't input a value in this column does not want to disclose that they have a mortgage account. Therefore, we can impute these missing valuee. With the median being 1 and the mean being 1.81, we can perform median imputation on this variable.

Hide

```
# Median imputation on the mort_acc variable
df$mort_acc[is.na(df$mort_acc)] <- 1
sum(is.na(df$mort_acc))
```

```
## [1] 0
```

2. pub_rec_bankrupcies

With 336 missing values missing with this variable, we can either impute values of get rid of these rows of data. The vast majority of these values are 0, so lets perform median imputation and plug in 0 to these values.

Hide

```
#Median imputation to the pub_rec_bankrupcies variable
df$pub_rec_bankruptcies[is.na(df$pub_rec_bankruptcies)] <- 0
sum(is.na(df$pub_rec_bankruptcies))
```

```
## [1] 0
```

3. revol_util

There are 177 missing values with this variables, indiciating either removing the rows with these missing values or imputing a value that won't vastly change the variable integrity. Since the summary statistics show the revolving utility to have a larger range, imputing data in these variables could slighty skew the data into a misleading direction. Therefore, it is best to delete these 177 rows of data.

Hide

```
# Removing rows where revol_util is NA
df <- df[!is.na(df$revol_util), ]
nrow(df)
```

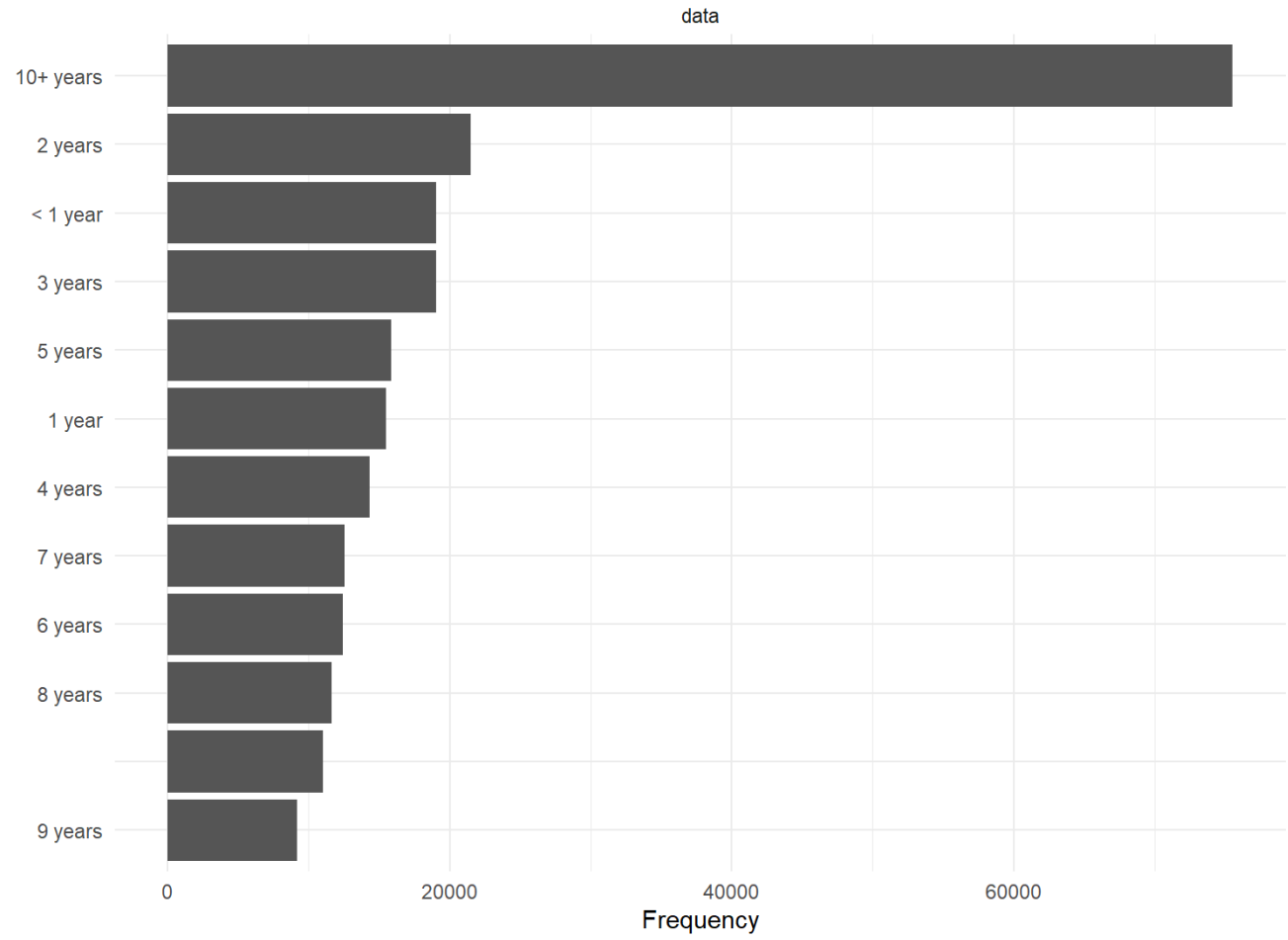
```
## [1] 237553
```

237730-237553 = 177 rows deleted from the data.

4. emp_length

Hide

```
DataExplorer::plot_bar(df$emp_length,
                       ggtheme = theme_minimal(),
                       ncol=4,
                       nrow=5)
```



Hide

```
table(df$emp_length)
```

```
##
##           < 1 year    1 year 10+ years   2 years   3 years   4 years   5 years
##     11031     19049     15480     75529     21514     19013     14329     15853
##   6 years   7 years   8 years   9 years
##     12448     12517     11600      9190
```

Hide

```
emp_numeric <- df$emp_length

emp_numeric <- gsub("< 1 year", "0", emp_numeric)
emp_numeric <- gsub("10\\+ years", "10", emp_numeric)
emp_numeric <- gsub(" years", "", emp_numeric)
emp_numeric <- gsub(" year", "", emp_numeric)
emp_numeric[emp_numeric == ""] <- NA
emp_numeric <- as.numeric(emp_numeric)
summary(emp_numeric)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   3.000   6.000   5.939  10.000  10.000   11031
```

Hide

```
emp_numeric[is.na(emp_numeric)] <- median(emp_numeric, na.rm = TRUE)

df$emp_length <- emp_numeric

table(df$emp_length)
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10
## 19049 15480 21514 19013 14329 15853 23479 12517 11600  9190 75529
```

4. Outlier/Anomaly Detection

Throughout our data cleaning process, we have identified numerous variables with outliers. Below, we will decide what to do with these variables. In deciding what to do, we must ask ourselves: Are these outliers realistic in context of the real world?

1. pub_rec

This variable has 1 extreme value at 86 when most values are 0. At that point of having that many public derogatory records, you are bound to default on your loan. Therefore, this value can remain in the data set.

2. revol_util

This singular outlier is extreme given the calculation for this value is credit used / total credit. This would mean that this individual would use an absurd amount of credit used. We will cap this variable to keep the integrity of the data realistic.

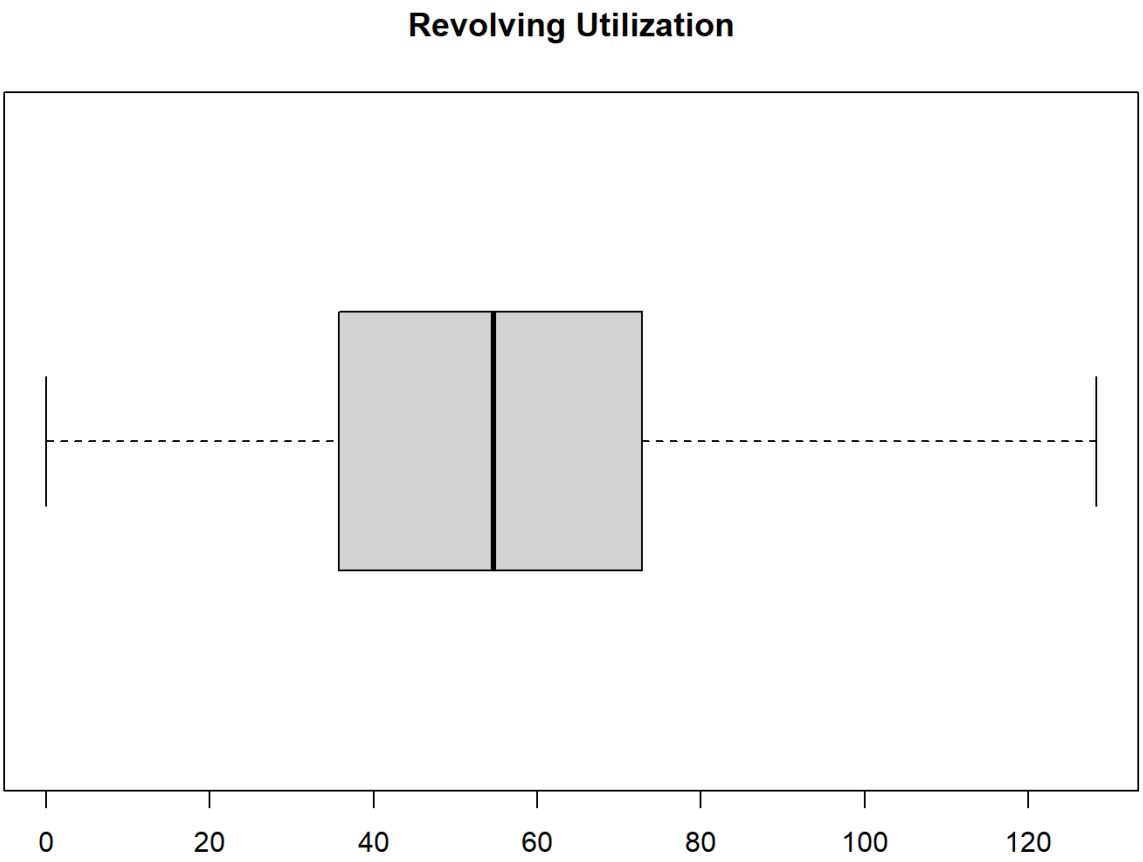
Hide

```
# Capping the revol_util variable
Q1 <- quantile(df$revol_util, 0.25, na.rm = TRUE)
Q3 <- quantile(df$revol_util, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
upper_bound <- Q3 + 1.5 * IQR
df$revol_util[df$revol_util > upper_bound] <- upper_bound
summary(df$revol_util)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   35.80   54.70   53.75   72.80  128.30
```

Hide

```
boxplot(df$revol_util, main = "Revolving Utilization", horizontal = TRUE)
```



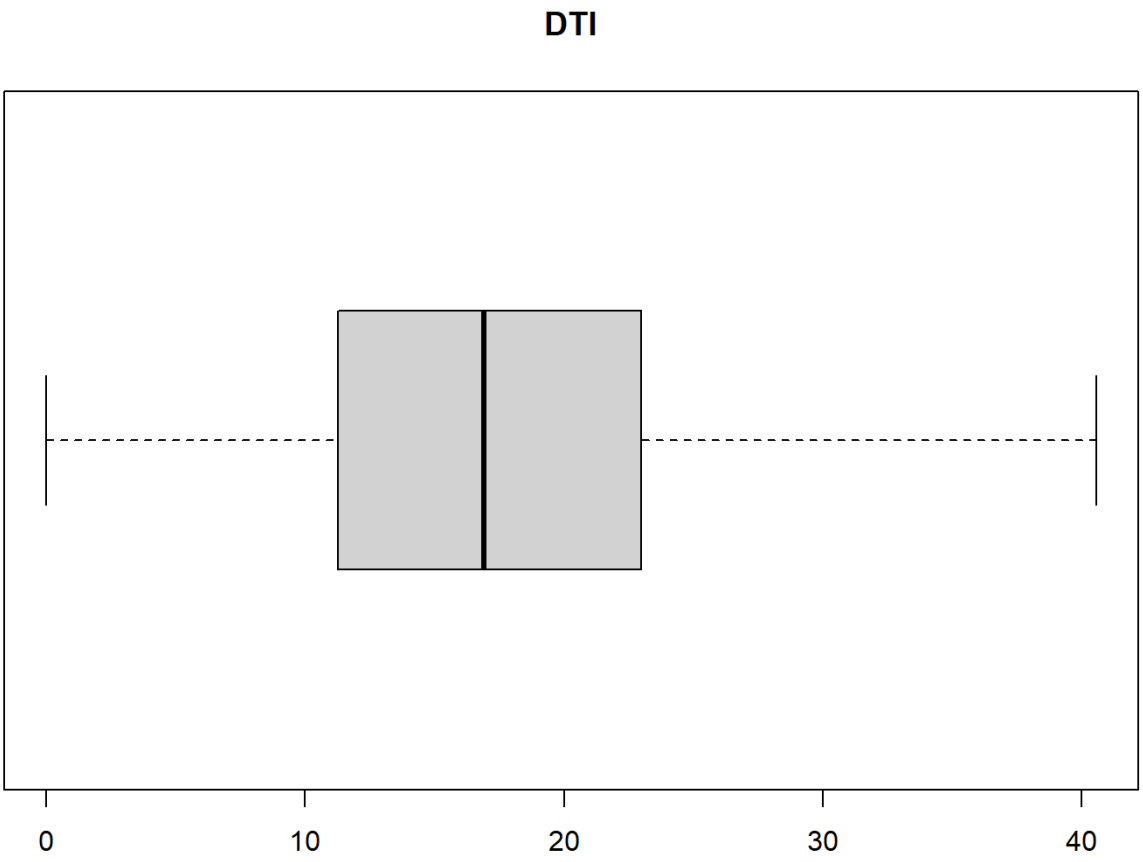
3. dti

There is 1 extreme outlier that should be referred to as an anomaly. The next highest value would be under 50, and having a debt-to-income ratio that high is unheard of. Let's cap this outlier.

Hide

```
# Capping the dti outlier
Q1 <- quantile(df$dti, 0.25, na.rm = TRUE)
Q3 <- quantile(df$dti, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1

upper_bound <- Q3 + 1.5 * IQR
df$dti[df$dti > upper_bound] <- upper_bound
boxplot(df$dti, main = "DTI", horizontal = TRUE)
```



4. annual_inc

There is one outlier that significantly larger than the rest. However, we will be keeping this variable because it is realistic for someone to make that much money annually. Think C-suite executives or professional athletes.

5. total_acc

There is one significant outlier, but the skew of the data is more intriguing than that outlier. We will keep this outlier.

6. open_acc

Two specific outliers in the data, we do not see any need to remove these outliers given it is one value of hundreds of thousands and there's one value that reflect a "yes" or "no" response in loan_default.

7. revol_bal

This outlier is realistic in the same sense that the annual income outlier is realistic. Some people have a significant amount of money and utilize these funds to start businesses.

5. Data Transformation

When deciding to transform certain variables in our data, we need to ensure there is a wide enough skew. Many variables, such as fico_range_high and fico_range_low have very obvious right skews in their data, driving up the mean and keeping the median in a lower value. Although this could be transformed, we do not want to alter the data to lose its integrity and predictability within our future training model.

6. Dimension Reduction

Dimension reduction could be an important part of training an accurate model, especially with variables that have an extreme amount of levels. Let's take subgrade for example. We could use the Principal Component Analysis Method (PCA) to capture the most important information on that variable while reducing the 35 levels in that variable alone. However, we argue that changing this variable to numeric can better capture its predicting power. Overall, performing the PCA method does not seem like our predictive power would increase for any categorical variable with multiple levels.

Feature Selection

For feature selection, we will be removing the "other" and "none" sections in the home_ownership variable.

Hide

```
#Removing the "other" and "none" values
df <- df[!df$home_ownership %in% c("OTHER", "NONE"), ]
df$home_ownership <- droplevels(df$home_ownership)
table(df$home_ownership)
```

```
##
## MORTGAGE      OWN     RENT
##   118722    22700    96040
```

With this, we removed the levels "Other" and "None" from this data set, giving the model a clearer set of three levels that have a good amount of values.

Reducing Dimensionality

The purpose variable shows many levels that have few values assigned to those levels. There are three main categories for this variable, so we will reduce the dimension and assign the rest of the values to the "other" dimension.

Hide

```
#Reduce dimensions in the purpose variable
df$purpose <- fct_other(df$purpose, keep = c("debt_consolidation", "credit_card", "home_im
provement"), other_level = "other")
table(df$purpose)
```

```
##
##        credit_card debt_consolidation   home_improvement              other
##              49782             140765              14365              32550
```

We also want to put emp_length into 5 main groups so we can dummy encode this variable later. We will put these dimension in 5 levels: 0-3 years, 4-6 years, 7-9 years, and 10+ years

Hide

```
df1=df
# Reducing dimensions on emp_length
df <- df %>%
  mutate(emp_length = case_when(
    emp_length %in% c("1","2", "3") ~ "0-3 years",
    emp_length %in% c("4", "5", "6") ~ "4-6 years",
    emp_length %in% c("7", "8", "9") ~ "7-9 years",
    emp_length == "10" ~ "10+ years"
  ))
df$emp_length <- factor(df$emp_length,
                                levels = c("0-3 years", "4-6 years", "7-9 years", "10+ yea
rs"),
                                ordered = TRUE)
df <- df %>%
  filter(!is.na(emp_length) & emp_length != "Unknown")
```

Overall, uninformative variables have been dropped and factors have been clustered. We can now focus on our current dimensions that have been kept.

7. Encoding Categorical Variables

Encoding these categorical variables are important for our logistic regression. Without dummy-encoding, the models won't be able to provide any significant predictors. Below, we will be using the fastdummies package to make our model useful for a logistic (glm) regression.

Hide

```
# Dummy encoding all remaining categorical variables
df <- fastDummies::dummy_cols(df,
                              select_columns = c("loan_default",
                                                 "term",
                                                 "home_ownership",
                                                 "verification_status",
                                                 "grade",
                                                 "purpose",
                                                 "initial_list_status",
                                                 "debt_settlement_flag",
                                                 "emp_length"),
                              remove_first_dummy = TRUE,      # avoids dummy variable trap
                              remove_selected_columns = TRUE) # removes original factor co
lumns
```

All of our factor variables are now dummy-encoded and ready to be train in a logistic regression model. We decided to dummy-encode all of these variables to be consistent across the board in our analysis.

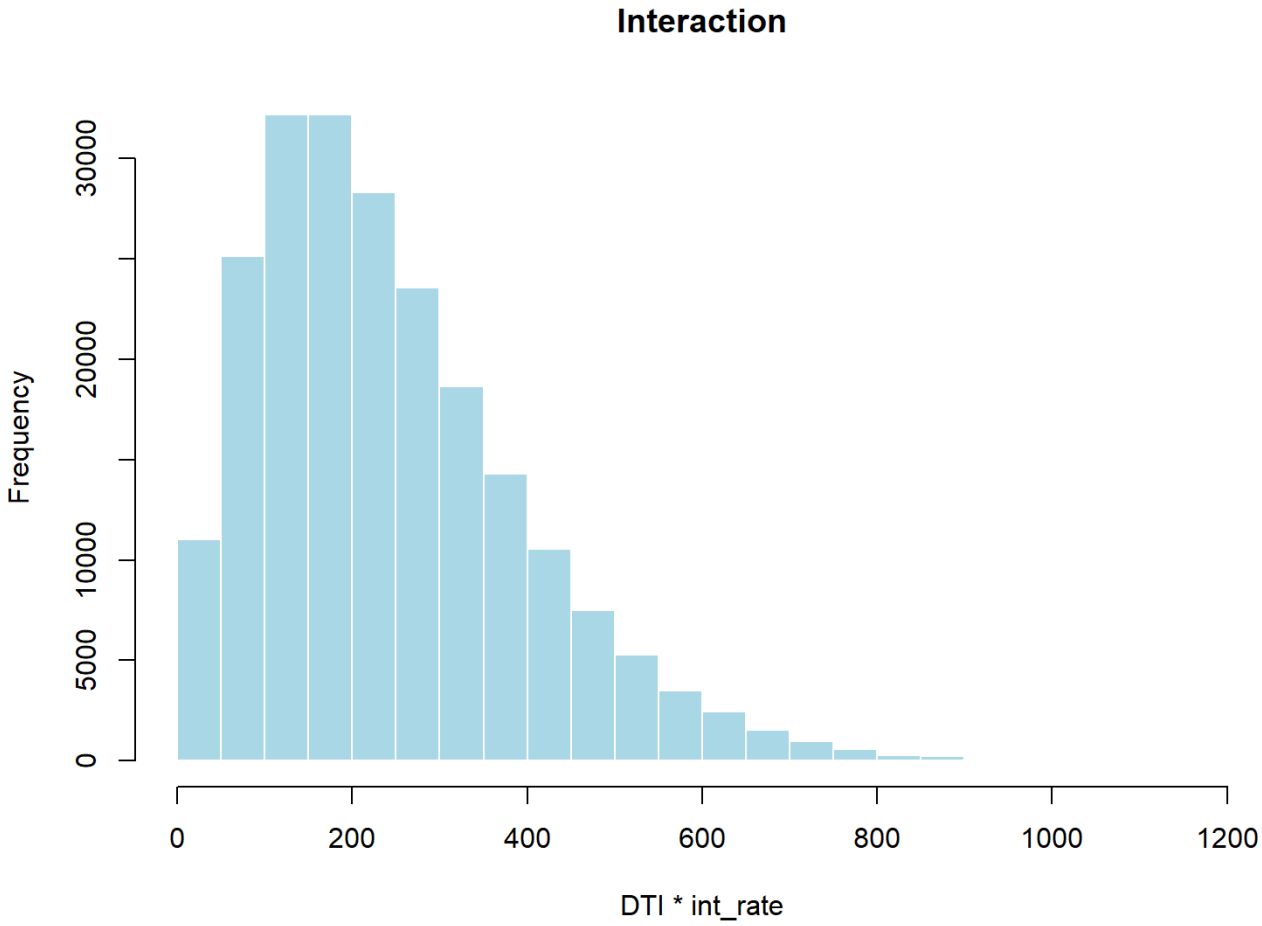
8. Feature Creation

There are many different feature creation methods that we could perform to potentially make our model the best in predicting loan default. That being said, these methods could hurt us as much as benefit our predicition accuracy if we perform something confusing or misleading.

For our feature creation method of choice, we found that the debt to income ratio and the interest rate together could indicate a higher probability to loan default. Therefore, we tried pefroming an interaction term:

Hide

```
# dti * int_rate interaction term
df$interaction_dti_interest <- df$dti * df$int_rate
# Show correlation
hist(df$interaction_dti_interest,
     main = "Interaction",
     xlab = "DTI * int_rate",
     col = "lightblue",
     border = "white")
```



There is a pretty significant right skew in this interaction term. For now, we will utilize this interaction term to help train our model in the upcoming sections.

Conclusion: Summary of our Cleaned Data

After performing all of these steps to create the best data frame to train a model, we have resulted in this dataset:

Hide

```
summary(df)
```

```
##    loan_amnt        int_rate       annual_inc           dti
##  Min.   :  500   Min.   : 5.32   Min.   :      0   Min.   : 0.00
##  1st Qu.: 8000   1st Qu.:10.49   1st Qu.:  45505   1st Qu.:11.36
##  Median :12000   Median :13.33   Median :  65000   Median :16.97
##  Mean   :14189   Mean   :13.63   Mean   :  74686   Mean   :17.40
##  3rd Qu.:20000   3rd Qu.:16.49   3rd Qu.:  90000   3rd Qu.:23.04
##  Max.   :40000   Max.   :30.99   Max.   :8706582   Max.   :40.56
##     open_acc        pub_rec          revol_bal        revol_util
##  Min.   : 1.00   Min.   : 0.0000   Min.   :     0   Min.   :  0.00
##  1st Qu.: 8.00   1st Qu.: 0.0000   1st Qu.:  6096   1st Qu.: 35.90
##  Median :10.00   Median : 0.0000   Median : 11294   Median : 54.80
##  Mean   :11.34   Mean   : 0.1817   Mean   : 15912   Mean   : 53.79
##  3rd Qu.:14.00   3rd Qu.: 0.0000   3rd Qu.: 19803   3rd Qu.: 72.80
##  Max.   :90.00   Max.   :86.0000   Max.   :814300   Max.   :128.30
##    total_acc         mort_acc      pub_rec_bankruptcies fico_range_low
##  Min.   :  2.00   Min.   : 0.000   Min.   :0.0000       Min.   :630.0
##  1st Qu.: 17.00   1st Qu.: 0.000   1st Qu.:0.0000       1st Qu.:670.0
##  Median : 24.00   Median : 1.000   Median :0.0000       Median :690.0
##  Mean   : 25.54   Mean   : 1.772   Mean   :0.1239       Mean   :696.5
##  3rd Qu.: 32.00   3rd Qu.: 3.000   3rd Qu.:0.0000       3rd Qu.:710.0
##  Max.   :150.00   Max.   :32.000   Max.   :8.0000       Max.   :845.0
##  fico_range_high inq_last_6mths   loan_default_Yes term_ 60 months
##  Min.   :634.0   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:674.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :694.0   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :700.5   Mean   :0.7795   Mean   :0.1952   Mean   :0.2389
##  3rd Qu.:714.0   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :850.0   Max.   :8.0000   Max.   :1.0000   Max.   :1.0000
##  home_ownership_OWN home_ownership_RENT verification_status_Source Verified
##  Min.   :0.00000    Min.   :0.0000      Min.   :0.0000
##  1st Qu.:0.00000    1st Qu.:0.0000      1st Qu.:0.0000
##  Median :0.00000    Median :0.0000      Median :0.0000
##  Mean   :0.09671    Mean   :0.3906      Mean   :0.3276
##  3rd Qu.:0.00000    3rd Qu.:1.0000      3rd Qu.:1.0000
##  Max.   :1.00000    Max.   :1.0000      Max.   :1.0000
##  verification_status_Verified    grade_B          grade_C
##  Min.   :0.0000               Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000               1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000               Median :0.0000   Median :0.0000
##  Mean   :0.3573               Mean   :0.2939   Mean   :0.2685
##  3rd Qu.:1.0000               3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000               Max.   :1.0000   Max.   :1.0000
##     grade_D          grade_E           grade_F           grade_G
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :0.000000
##  Mean   :0.1588   Mean   :0.07964   Mean   :0.02942   Mean   :0.007623
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000000
##  purpose_debt_consolidation purpose_home_improvement purpose_other
##  Min.   :0.0000             Min.   :0.00000          Min.   :0.0000
##  1st Qu.:0.0000             1st Qu.:0.00000          1st Qu.:0.0000
```

```
##  Median :1.0000             Median :0.00000          Median :0.0000
##  Mean   :0.5947             Mean   :0.06196          Mean   :0.1346
##  3rd Qu.:1.0000             3rd Qu.:0.00000          3rd Qu.:0.0000
##  Max.   :1.0000             Max.   :1.00000          Max.   :1.0000
##  initial_list_status_w debt_settlement_flag_Y emp_length_4-6 years
##  Min.   :0.0000        Min.   :0.00000        Min.   :0.0000
##  1st Qu.:0.0000        1st Qu.:0.00000        1st Qu.:0.0000
##  Median :0.0000        Median :0.00000        Median :0.0000
##  Mean   :0.4007        Mean   :0.01307        Mean   :0.2456
##  3rd Qu.:1.0000        3rd Qu.:0.00000        3rd Qu.:0.0000
##  Max.   :1.0000        Max.   :1.00000        Max.   :1.0000
##  emp_length_7-9 years emp_length_10+ years interaction_dti_interest
##  Min.   :0.0000       Min.   :0.0000       Min.   :   0.0
##  1st Qu.:0.0000       1st Qu.:0.0000       1st Qu.: 129.1
##  Median :0.0000       Median :0.0000       Median : 214.7
##  Mean   :0.1524       Mean   :0.3457       Mean   : 243.7
##  3rd Qu.:0.0000       3rd Qu.:1.0000       3rd Qu.: 328.9
##  Max.   :1.0000       Max.   :1.0000       Max.   :1176.0
```

This summary shows the finalized data set we will be using to train our model. We will now convert our data into an RDS file and continue building out our model. While there could be other actions performed, this will be a great start on our path to have the champion model!

Hide

```
# Convert into an rds file
saveRDS(df, file = "group5AA_Black-Boopathy_train.rds")
```