Lending Club - Predicting Defaults on Loans

Josh Janda

September 4, 2019

Introduction

For my project, I will be evaluating the Lending Club loan dataset. In specific, I will be evaluating loans from the 2017 Quarter 1 period. This dataset includes information of all loans given by LendingClub.com. The goal of this project is to be able to predict the probability of default of a loan given a certain set of features. Some of the most important features (in my opinion), are:

- *annual_inc*: The self-reported annual income provided by the borrower during registration.
- *chargeoff_within_12_mths*: Number of charge-offs within 12 months
- *emp_length*: Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
- *grade*: Lending Club assigned loan grade
- *sub_grade*: Lending Club assigned loan subgrade
- *home_ownership*: The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
- *int_rate*: Interest Rate on the loan
- *loan_amt*: The listed amount of the loan applied for by the borrower.
- *pub_rec_bankruptcies*: Number of public record bankruptcies
- *totalAcc*: The total number of credit lines currently in the borrower's credit file

While there are many more variables, 145 to be exact, I believe these are some of the most important features that will help predict whether or not a loan will default.

Moving onto our target variable, we will be trying to predict:

- *loan_status*: Current status of the loan

Data Loading and Exploration

Now that we have introduced the data and our goal for it, we will want to load the data. But first, let's start with importing all needed libraries to run this project.

Now, let's load the data.

```
lending_data = vroom('loan.csv', delim = ",", na = "")
```

```
## Observations: 2,260,668
## Variables: 145
## chr [ 36]: term, grade, sub_grade, emp_title, emp_length, home_ownership, verification_...
## dbl [106]: loan_amnt, funded_amnt, funded_amnt_inv, int_rate, installment, annual_inc, ...
## lgl [  3]: id, member_id, url
##
## Call `spec()` for a copy-pastable column specification
## Specify the column types with `col_types` to quiet this message
```

We can now take a full look at all of the variables included in this data and the total number of observations and variables in this dataset.

```
colnames(lending_data)
```

```
##   [1] "id"
##   [2] "member_id"
```

```
##   [3] "loan_amnt"
##   [4] "funded_amnt"
##   [5] "funded_amnt_inv"
##   [6] "term"
##   [7] "int_rate"
##   [8] "installment"
##   [9] "grade"
##  [10] "sub_grade"
##  [11] "emp_title"
##  [12] "emp_length"
##  [13] "home_ownership"
##  [14] "annual_inc"
##  [15] "verification_status"
##  [16] "issue_d"
##  [17] "loan_status"
##  [18] "pymnt_plan"
##  [19] "url"
##  [20] "desc"
##  [21] "purpose"
##  [22] "title"
##  [23] "zip_code"
##  [24] "addr_state"
##  [25] "dti"
##  [26] "delinq_2yrs"
##  [27] "earliest_cr_line"
##  [28] "inq_last_6mths"
##  [29] "mths_since_last_delinq"
##  [30] "mths_since_last_record"
##  [31] "open_acc"
##  [32] "pub_rec"
##  [33] "revol_bal"
##  [34] "revol_util"
##  [35] "total_acc"
##  [36] "initial_list_status"
##  [37] "out_prncp"
##  [38] "out_prncp_inv"
##  [39] "total_pymnt"
##  [40] "total_pymnt_inv"
##  [41] "total_rec_prncp"
##  [42] "total_rec_int"
##  [43] "total_rec_late_fee"
##  [44] "recoveries"
##  [45] "collection_recovery_fee"
##  [46] "last_pymnt_d"
##  [47] "last_pymnt_amnt"
##  [48] "next_pymnt_d"
##  [49] "last_credit_pull_d"
##  [50] "collections_12_mths_ex_med"
##  [51] "mths_since_last_major_derog"
##  [52] "policy_code"
##  [53] "application_type"
##  [54] "annual_inc_joint"
##  [55] "dti_joint"
##  [56] "verification_status_joint"
```

```
##  [57] "acc_now_delinq"
##  [58] "tot_coll_amt"
##  [59] "tot_cur_bal"
##  [60] "open_acc_6m"
##  [61] "open_act_il"
##  [62] "open_il_12m"
##  [63] "open_il_24m"
##  [64] "mths_since_rcnt_il"
##  [65] "total_bal_il"
##  [66] "il_util"
##  [67] "open_rv_12m"
##  [68] "open_rv_24m"
##  [69] "max_bal_bc"
##  [70] "all_util"
##  [71] "total_rev_hi_lim"
##  [72] "inq_fi"
##  [73] "total_cu_tl"
##  [74] "inq_last_12m"
##  [75] "acc_open_past_24mths"
##  [76] "avg_cur_bal"
##  [77] "bc_open_to_buy"
##  [78] "bc_util"
##  [79] "chargeoff_within_12_mths"
##  [80] "delinq_amnt"
##  [81] "mo_sin_old_il_acct"
##  [82] "mo_sin_old_rev_tl_op"
##  [83] "mo_sin_rcnt_rev_tl_op"
##  [84] "mo_sin_rcnt_tl"
##  [85] "mort_acc"
##  [86] "mths_since_recent_bc"
##  [87] "mths_since_recent_bc_dlq"
##  [88] "mths_since_recent_inq"
##  [89] "mths_since_recent_revol_delinq"
##  [90] "num_accts_ever_120_pd"
##  [91] "num_actv_bc_tl"
##  [92] "num_actv_rev_tl"
##  [93] "num_bc_sats"
##  [94] "num_bc_tl"
##  [95] "num_il_tl"
##  [96] "num_op_rev_tl"
##  [97] "num_rev_accts"
##  [98] "num_rev_tl_bal_gt_0"
##  [99] "num_sats"
## [100] "num_tl_120dpd_2m"
## [101] "num_tl_30dpd"
## [102] "num_tl_90g_dpd_24m"
## [103] "num_tl_op_past_12m"
## [104] "pct_tl_nvr_dlq"
## [105] "percent_bc_gt_75"
## [106] "pub_rec_bankruptcies"
## [107] "tax_liens"
## [108] "tot_hi_cred_lim"
## [109] "total_bal_ex_mort"
## [110] "total_bc_limit"
```

```
## [111] "total_il_high_credit_limit"
## [112] "revol_bal_joint"
## [113] "sec_app_earliest_cr_line"
## [114] "sec_app_inq_last_6mths"
## [115] "sec_app_mort_acc"
## [116] "sec_app_open_acc"
## [117] "sec_app_revol_util"
## [118] "sec_app_open_act_il"
## [119] "sec_app_num_rev_accts"
## [120] "sec_app_chargeoff_within_12_mths"
## [121] "sec_app_collections_12_mths_ex_med"
## [122] "sec_app_mths_since_last_major_derog"
## [123] "hardship_flag"
## [124] "hardship_type"
## [125] "hardship_reason"
## [126] "hardship_status"
## [127] "deferral_term"
## [128] "hardship_amount"
## [129] "hardship_start_date"
## [130] "hardship_end_date"
## [131] "payment_plan_start_date"
## [132] "hardship_length"
## [133] "hardship_dpd"
## [134] "hardship_loan_status"
## [135] "orig_projected_additional_accrued_interest"
## [136] "hardship_payoff_balance_amount"
## [137] "hardship_last_payment_amount"
## [138] "disbursement_method"
## [139] "debt_settlement_flag"
## [140] "debt_settlement_flag_date"
## [141] "settlement_status"
## [142] "settlement_date"
## [143] "settlement_amount"
## [144] "settlement_percentage"
## [145] "settlement_term"
```

```
total_obs1 = nrow(lending_data)
total_vars1 = ncol(lending_data)
```

Total observations in this dataset: 2260668

Total variables in this dataset: 145

Target Variable Exploration

So, as stated above, there are a lot of variables given to help predict whether or not a person will default on their loan. Let's explore our target variable, *loan_status*, more closely to get an idea of what possible values this variable can take.

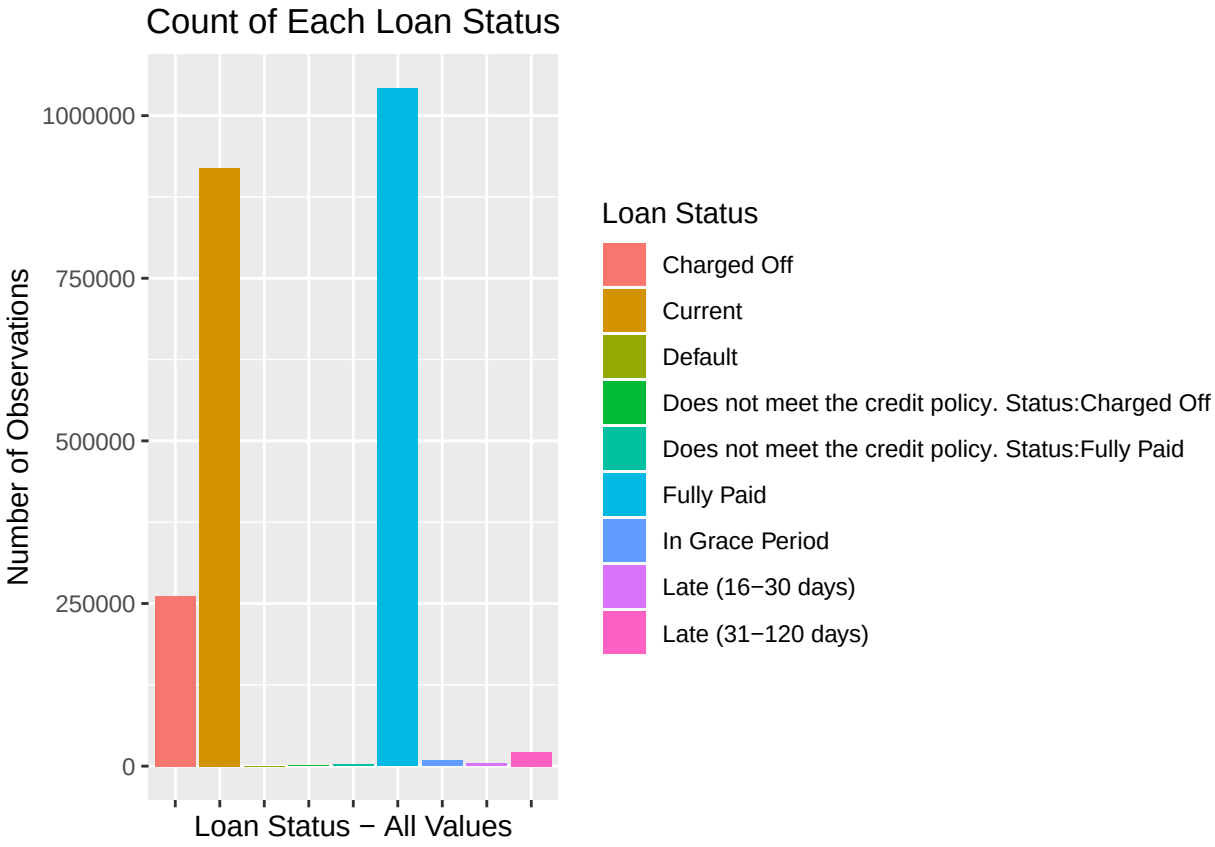
```
unique(lending_data$loan_status)
```

```
## [1] "Current"
## [2] "Fully Paid"
## [3] "Late (31-120 days)"
## [4] "In Grace Period"
## [5] "Charged Off"
## [6] "Late (16-30 days)"
```

```
## [7] "Default"
## [8] "Does not meet the credit policy. Status:Fully Paid"
## [9] "Does not meet the credit policy. Status:Charged Off"
```

There are a total of 9 different values that *loan_status* can take. Since we are interested in predicting whether someone will default on their loan or not, we will clean this variable later on so it only takes on values default or not default. Forn now, let's take a look at how many of each value we have using a bar graph.

```
ggplot(data = lending_data, aes(x = factor(loan_status), fill = loan_status)) +
  geom_bar() +
  theme(axis.text.x = element_blank(), plot.title = element_text(hjust = 0.5)) +
  ggtitle("Count of Each Loan Status") +
  xlab("Loan Status - All Values") +
  ylab("Number of Observations") +
  labs(fill = "Loan Status")
```



Looking at the bar graph above, we can see that the top three values that *loan_status* takes is "Charged Off", "Current", and "Fully Paid". All other values are very low in the number of observations.

Since we are interested in predicting whether or not someone will default on a loan, we can remove the values that will be not helpful in making this prediction. I will justify why I am removing each value from *loan_status* that I believe is not helpful.

- "Current": If you are current on your loan, this does not mean you will remain current. You can eventually default on the loan or pay it off, we do not know.
- "In Grace Period": If you are in the grace period, you can always go back to being current on your loan or eventually default. We do not know.
- "Late (16-30 days)": If you are late on paying your loan, you can always catch up on payments or eventually default. We do not know.

- "Late (31-120 days)": If you are late on paying your loan, you can always catch up on payments or eventually default. We do not know.

So, I will be keeping these values to eventually create a single variable that has the values "Default" or "Not Default"

- "Charged Off": This loan was defaulted on and sent to collections, and Lending Club has charged off the loan and considered it a loss.
- "Default": This loan was defaulted on.
- "Does not meet the credit policy. Status: Charged Off": This loan was defaulted on and sent to collections, and Lending Club has charged off the loan and considered it a loss.
- "Does not meet the credit policy. Status: Fully Paid": This loan was fully paid off and was not defaulted on.
- "Fully Paid": This loan was fully pauid off and was not defaulted on.

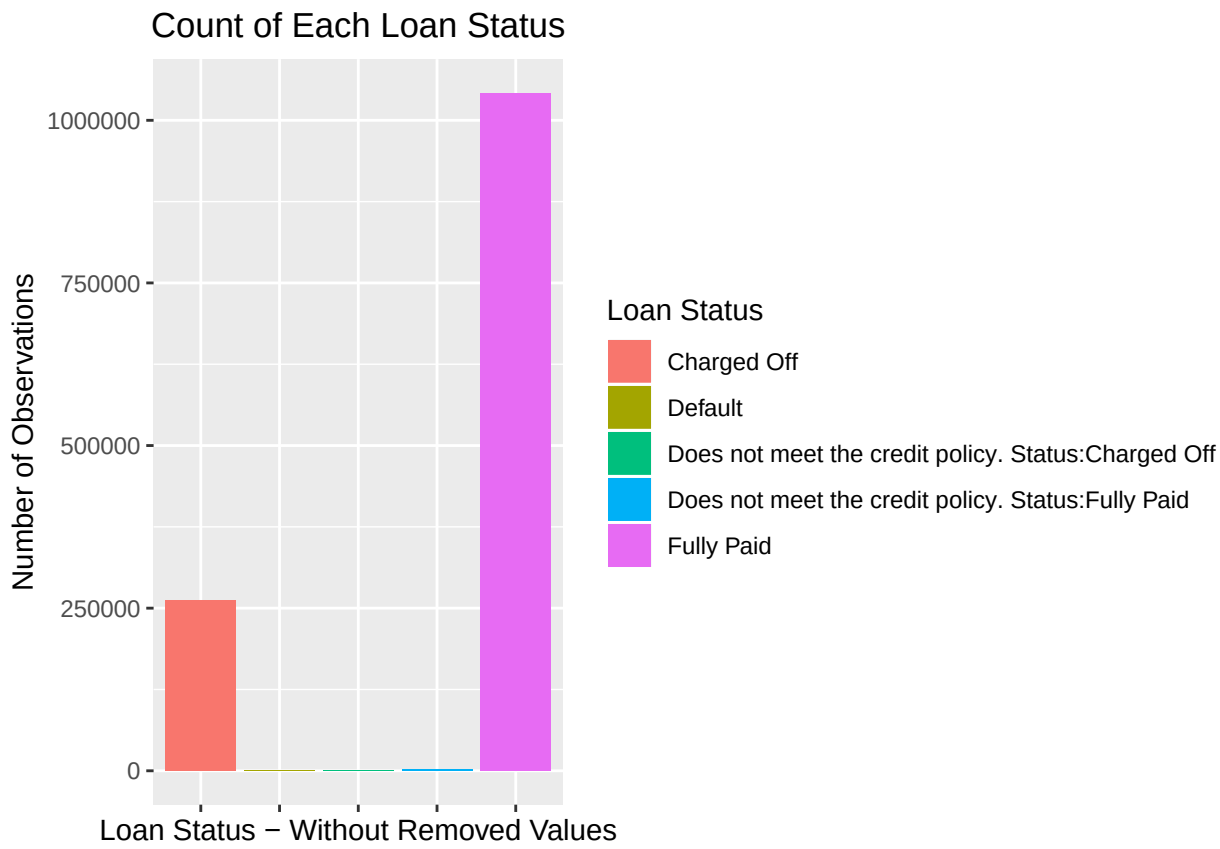
Let's clean the data a little bit now by removing all rows that have any of the four values for *loan_status* in order to remove these values.

```
'%ni%' = Negate('%in%') #create function that does the opposite of %in%

lending_data_target_clean = lending_data %>% filter(
  loan_status %ni% c("Current", "In Grace Period", "Late (16-30 days)", "Late (31-120 days)"))
total_obs2 = nrow(lending_data_target_clean)
```

With those values of *loan_status* removed, we have shrunken the data to only 1306387 rows, which is much smaller and more manageable. Let's take a look at a bar plot of *loan_status* once again.

```
ggplot(data = lending_data_target_clean, aes(x = factor(loan_status), fill = loan_status)) +
  geom_bar() +
  theme(axis.text.x = element_blank(), plot.title = element_text(hjust = 0.5)) +
  ggtitle("Count of Each Loan Status") +
  xlab("Loan Status - Without Removed Values") +
  ylab("Number of Observations") +
  labs(fill = "Loan Status")
```

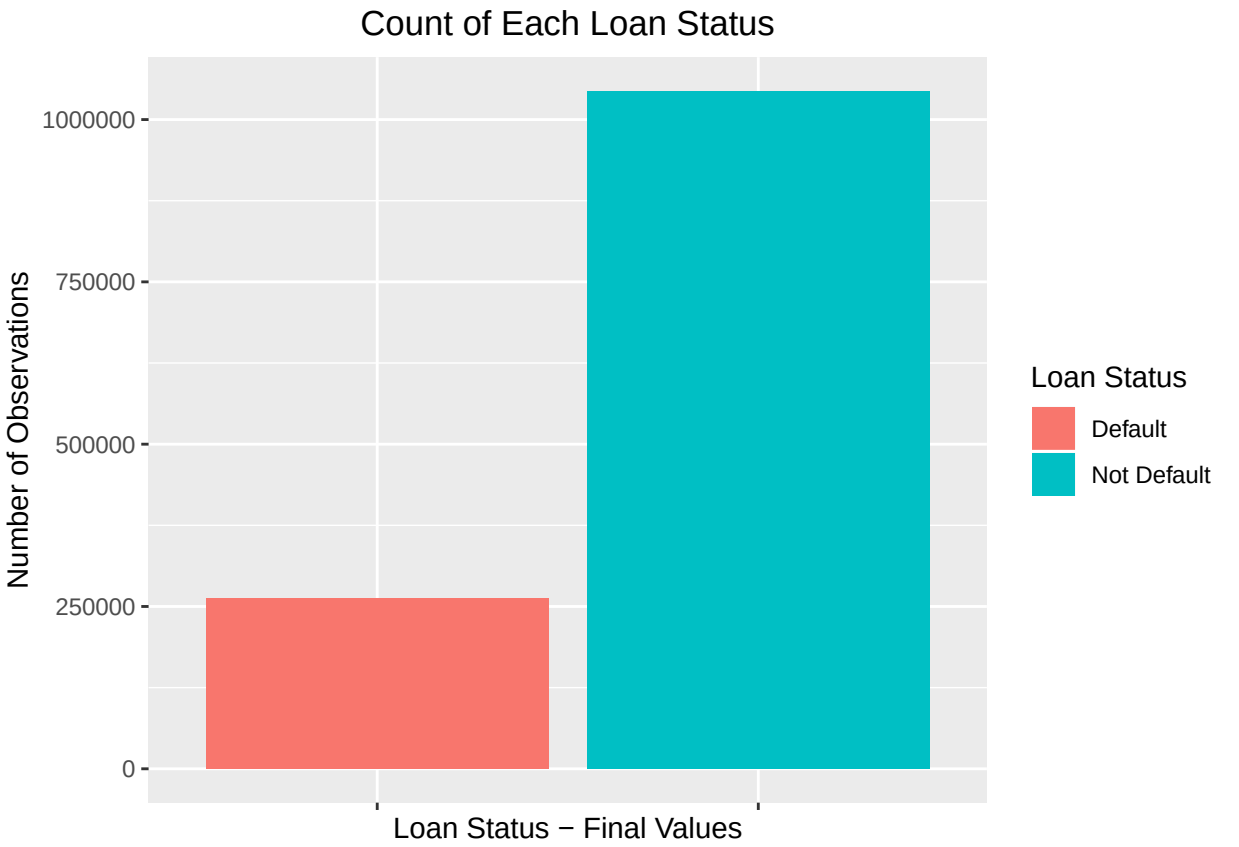


This plot looks much cleaner with the redudant levels of *loan_status* removed. Let's now create our final variable of *loan_status*, that we will use for model building. This variable will include levels of "Default" or "Not Default".

```
lending_data_target_clean$loan_status_final = ifelse(
  lending_data_target_clean$loan_status %in% c("Charged Off",
    "Default", "Does not meet the credit policy. Status:Charged Off"), "Default", "Not Default")
```

Let's take a look at the bar graph one more time for our created variable *loan_status_final*.

```
ggplot(data = lending_data_target_clean, aes(x = factor(loan_status_final), fill = loan_status_final)) +
  geom_bar() +
  theme(axis.text.x = element_blank(), plot.title = element_text(hjust = 0.5)) +
  ggtitle("Count of Each Loan Status") +
  xlab("Loan Status - Final Values") +
  ylab("Number of Observations") +
  labs(fill = "Loan Status")
```



With this variable created, we will now consider this to be our target variable that we will be using for model building and further visualization.

Feature Exploration

With the data cleaned and redudant rows removed, let's move on to exploring some of the important "X" variables, or features in this dataset. I will be taking a look at the three most important features, which I have picked.

- *annual_inc*: The self-reported annual income provided by the borrower during registration.
- *int_rate*: Interest Rate on the loan
- *loan_amt*: The listed amount of the loan applied for by the borrower.

I have picked *annual_inc* as I believe this will play a crucial factor on whether someone will default on their loan or not. My belief is that the higher the income someone has, the less chance they have of defaulting on the loan. Let's explore some summary statistics on this variable and also the distribution of it through the use of a histogram.

```
summary(lending_data_target_clean$annual_inc)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##        0    45760    65000    76149    90000 10999200        4
```

Looking at the summary statistics of *annual_inc*, we have some interesting results to make note of. First of all, we need to pay attention to the number of "NA's" in this variable, which is 4. For these NA's, we can either remove the rows or impute a value into them. We will make this decision later on. Our minimum *annual_inc* is zero, which seems off as I would not believe that a loan would be given to someone with zero income. Our maximum *annual_inc* is 10,999,200, which also seems off as that is a lot of income and I am doubtful someone would need a loan with that much income. The average *annual_inc* is 76,149, which is a

reasonable average for annual incomes on people requesting loans. It should be noted that *annual_inc* is a field that the loanee provides to Lending Club, so it is safe to believe that some people did not enter in their true annual income.

Let's now take a look at the summary of *annual_inc*, when grouped by whether or not they have defaulted on their loan. We will be ignoring the NA values for this summary.

```
lending_data_target_clean %>% group_by(loan_status_final) %>% summarize(
  MinInc = min(annual_inc, na.rm = TRUE),
  MeanInc = mean(annual_inc, na.rm = TRUE),
  MaxInc = max(annual_inc, na.rm = TRUE))
```

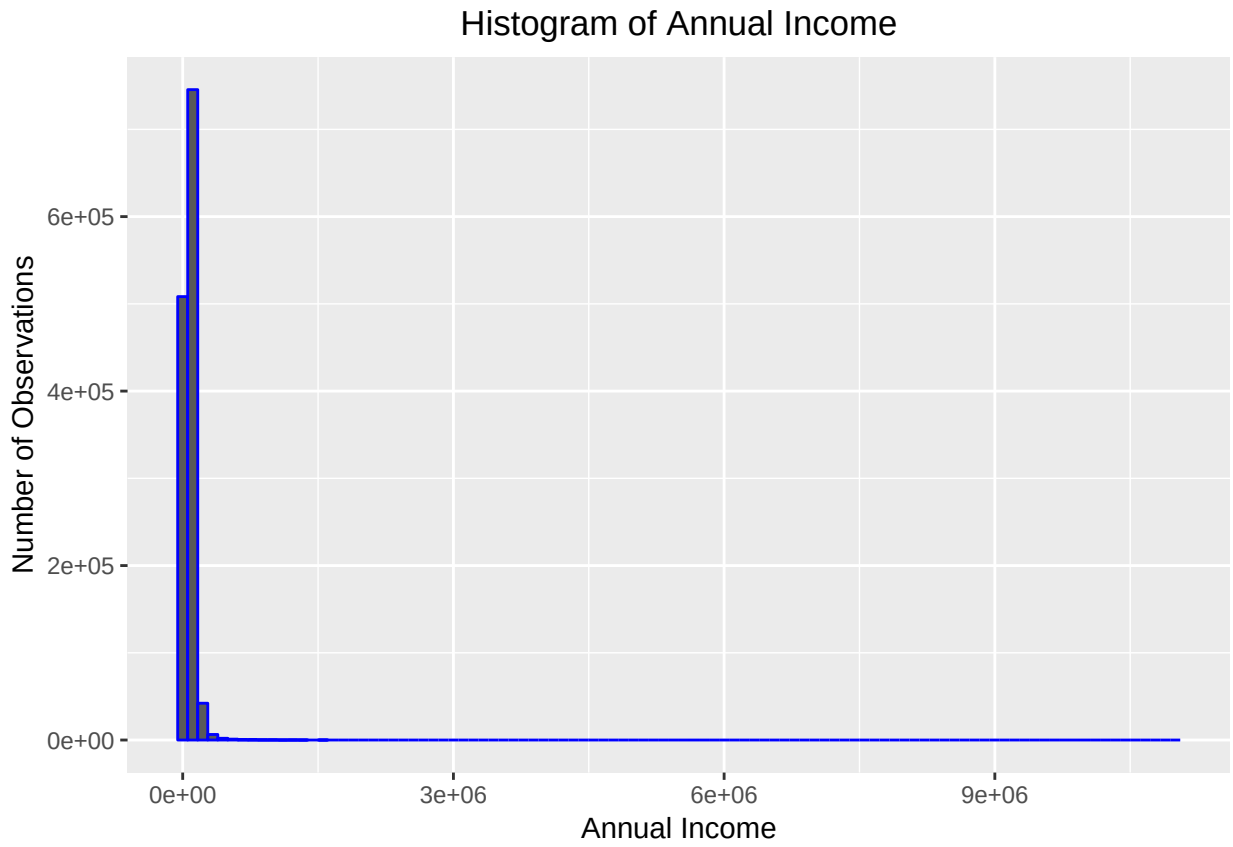
```
## # A tibble: 2 x 4
##   loan_status_final MinInc MeanInc   MaxInc
##   <chr>              <dbl>   <dbl>    <dbl>
## 1 Default                0  70325.  9500000
## 2 Not Default            0  77613. 10999200
```

Looking at the summary when grouped by whether or not the person has defaulted on their loan, we can see that the minimum income and maximum income really do not differ by much as each group has a minimum of zero and a maximum of a ridiculously high number. The mean incomes however differ by a good amount, about 7,000. This difference in means falls in limne with my belief that the higher someones income is the less chance they have of defaulting on their loan.

With summary statistics aside, let's take a look at the histogram of *annual_inc* to get an idea of the distribution.

```
ggplot(lending_data_target_clean, aes(x = annual_inc)) +
  geom_histogram(stat = "bin", bins = 100, color = "blue") +
  xlab("Annual Income") +
  ylab("Number of Observations") +
  ggtitle("Histogram of Annual Income") +
  theme(plot.title = element_text(hjust = 0.50))
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

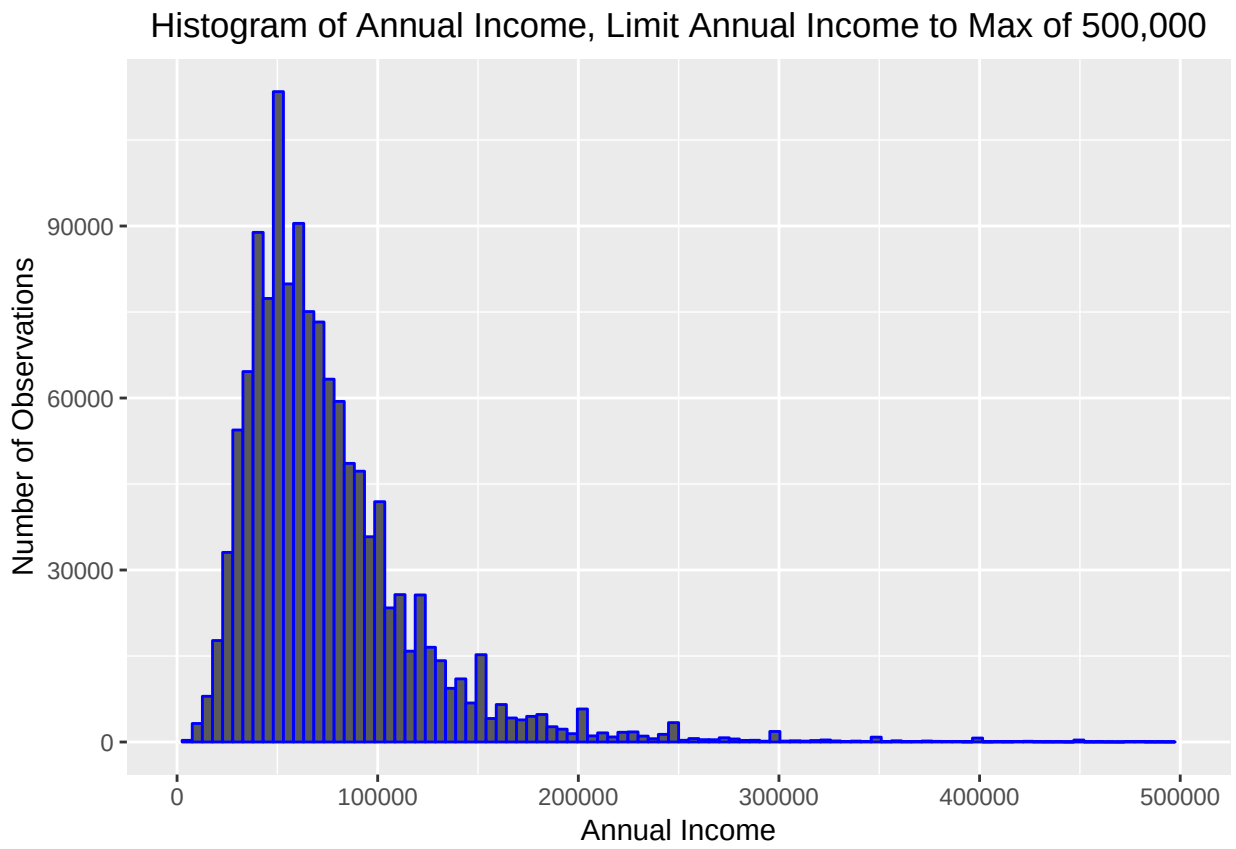


Looking at the histogram above, we don't really get a good idea of the distribution of *annual_inc* when allowing the range of the x-axis to span all of the way to the maximum value of 10,999,200. I am going to also include a histogram of *annual_inc* with an xrange of (0, 500,000) to get a better idea of the distribution. I have chosen 500,000 as that seems to be around the flatline of this histogram, also diving deeper into the data there are only 1675 observations greater than 500,000 which is only 0.1282162% of the data which is very minimal.

```
options(scipen = 10)
ggplot(lending_data_target_clean, aes(x = annual_inc)) +
  geom_histogram(stat = "bin", bins = 100, color = "blue") +
  xlim(0, 500000) +
  xlab("Annual Income") +
  ylab("Number of Observations") +
  ggtitle("Histogram of Annual Income, Limit Annual Income to Max of 500,000") +
  theme(plot.title = element_text(hjust = 0.50))
```

```
## Warning: Removed 1679 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Looking at the histogram of *annual_inc* with a limit of 500,000, we can see that most incomes surround the mean as the histogram has a very high peak. This distribution looks normal-ish if you remove outliers, however does have a fatter right tail as incomes can continue to increase while they cannot go below zero (I hope).

Moving on from *annual_inc*, let's now take a *int_rate*. I believe that this variable plays a crucial role in predicting the target variable of whether or not someone will default on their loan as interest rates are tied directly to how risky the loanee is. The more risky it is to borrow someone money (less chance that they will pay back the loan), the higher the interest rate you would charge them. Therefore, the higher the interest rate the higher the chance of someone defaulting on a loan. Let's explore some summary statistics on this variable and also the distribution of it through the use of a histogram.

```
summary(lending_data_target_clean$int_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.31    9.75   12.74   13.26   15.99   30.99
```

Looking at the summary statistics of *int_rate*, we have some interesting results to make note of. For the case of this variable, we have no NA values which is fantastic. Our minimum *int_rate* is 5.31%, which is a good interest rate for a loan. Our maximum *int_rate* is 30.99%, which is a very high interest rate on a loan and therefore this person was most likely very risk to give a loan to. The average *int_rate* is 13.26%, which is a reasonable average interest rate for loans given to all types of people.

Let's now take a look at the summary of *int_rate*, when grouped by whether or not they have defaulted on their loan.

```
lending_data_target_clean %>% group_by(loan_status_final) %>% summarize(
  MinIntRate = min(int_rate, na.rm = TRUE),
  MeanIntRate = mean(int_rate, na.rm = TRUE),
```

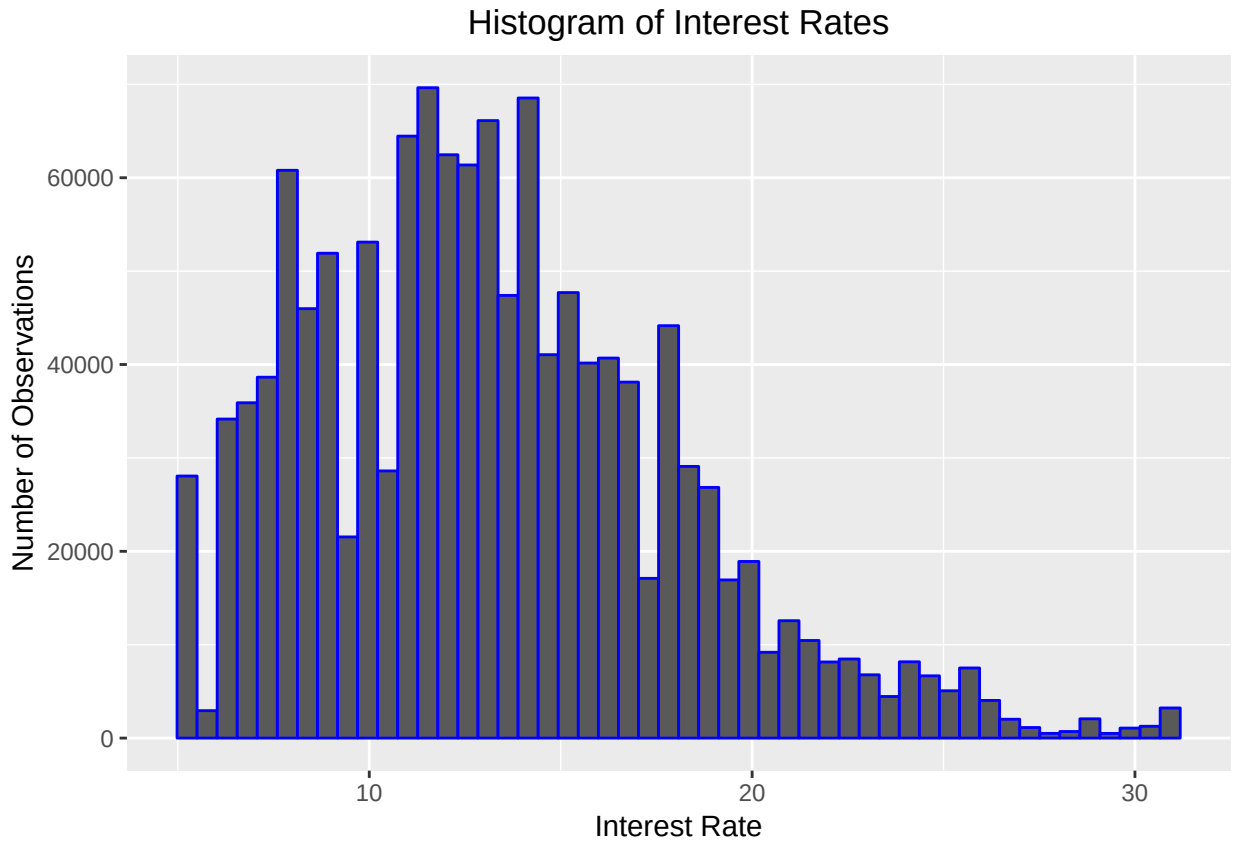
```
  MaxIntRate = max(int_rate, na.rm = TRUE))
```

```
## # A tibble: 2 x 4
##   loan_status_final MinIntRate MeanIntRate MaxIntRate
##   <chr>                  <dbl>       <dbl>      <dbl>
## 1 Default                 5.31        15.7       31.0
## 2 Not Default             5.31        12.6       31.0
```

Looking at the statistics of *int_rate* when grouped by whether or not someone has defaulted on their loan, once again we see that the minimum and maximum values are similar (equal in this case). I believe they are equal due to this being the minimum and maximum interest rates offered by Lending Club. However, when looking at the mean *int_rate* between groups, we can see that those who did not default on their loan have a lower mean interest rate. This falls in line with my belief that those with lower interest rates are less risky and therefore have less chance to default on their loan.

With summary statistics aside, let's take a look at the histogram of *int_rate* to get an idea of the distribution.

```
ggplot(lending_data_target_clean, aes(x = int_rate)) +
  geom_histogram(stat = "bin", bins = 50, color = "blue") +
  xlab("Interest Rate") +
  ylab("Number of Observations") +
  ggtitle("Histogram of Interest Rates") +
  theme(plot.title = element_text(hjust = 0.50))
```



Looking at the histogram of *int_rate*, it appears to have a normal-ish distribution with most interest rates being near the mean. However, it is right-skewed as more risky people receive higher interest rates therefore causing the distribution to have a fatter right tail.

Lastly, let's take a look at *loan_amt*. I believe that this variable also plays a crucial role in predicting whether or not someone will default on their loan as the larger the loan amount someone takes out I believe there is less chance that they plan on (and are capable of), repaying the loan. However, it should be noted that this is my belief and there could be cases where someone with more income needs a larger loan and is therefore capable of paying it back. Let's explore some summary statistics on this variable and also the distribution of it through the use of a histogram.

```
summary(lending_data_target_clean$loan_amnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     500    8000   12000   14406   20000   40000
```

Looking at the summary statistics of *loan_amnt*, we have some interesting results to make note of. For the case of this variable, we have no NA values which is fantastic. Our minimum *loan_amnt* is 500, which is a rather small loan and is likely to be paid back. Our maximum *loan_amnt* is 40,000, which is a larger-than-average loan and therefore carries more risk of being defaulted on. The average *loan_amnt* is 14,406, which seems high for an average of the given loan amounts.

Let's now take a look at the summary of *loan_amnt*, when grouped by whether or not they have defaulted on their loan.

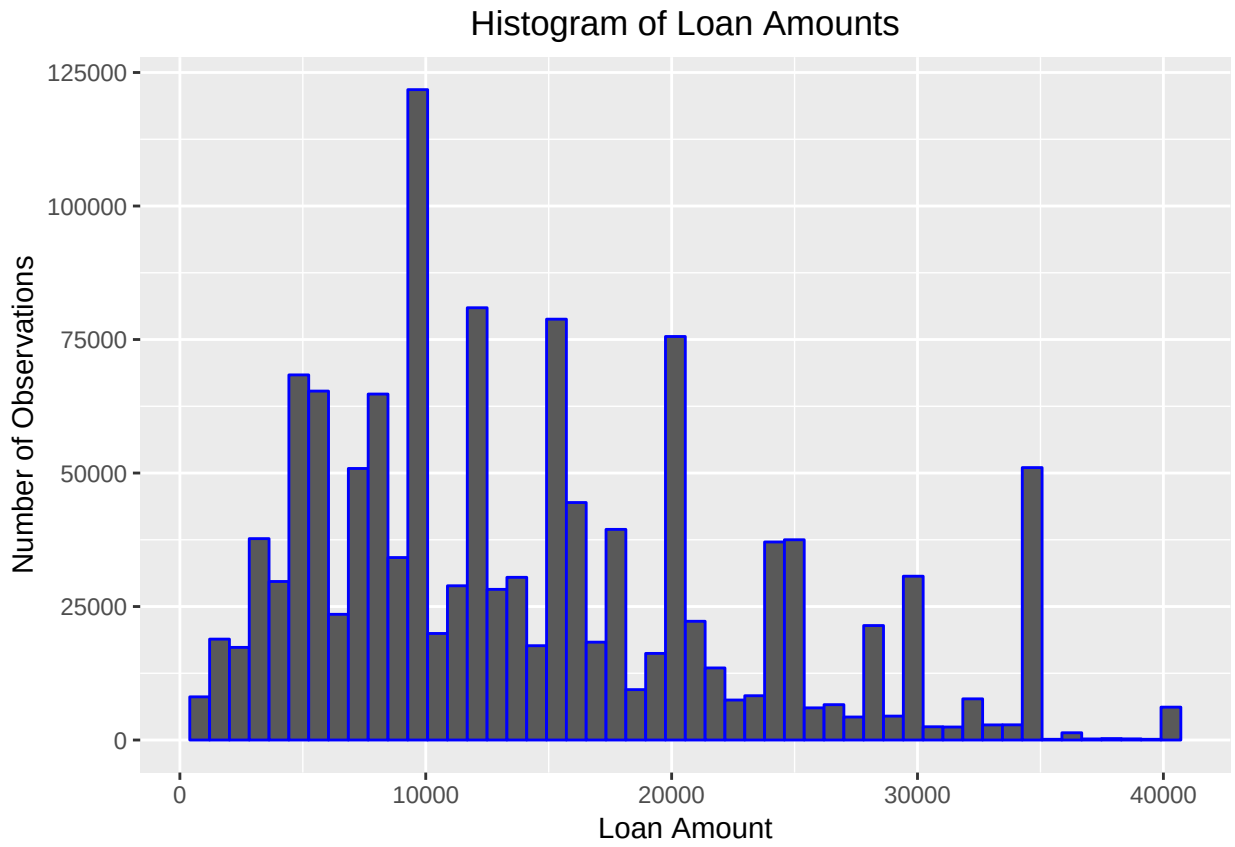
```
lending_data_target_clean %>% group_by(loan_status_final) %>% summarize(
  MinLoanAmnt = min(loan_amnt, na.rm = TRUE),
  MeanLoanAmnt = mean(loan_amnt, na.rm = TRUE),
  MaxLoanAmnt = max(loan_amnt, na.rm = TRUE))
```

```
## # A tibble: 2 x 4
##   loan_status_final MinLoanAmnt MeanLoanAmnt MaxLoanAmnt
##   <chr>                   <dbl>        <dbl>       <dbl>
## 1 Default                   500       15532.       40000
## 2 Not Default               500       14122.       40000
```

Looking at the statistics of *loan_amnt* when grouped by whether or not someone has defaulted on their loan, once again we see that the minimum and maximum values are similar (equal in this case). I believe they are equal due to this being the minimum and maximum loan amounts offered by Lending Club. However, when looking at the mean *loan_amnt* between groups, we can see that those who did not default on their loan have a lower mean loan amount. This falls in line with my belief that those with a lower loan amount have less chance of defaulting on their loan.

With summary statistics aside, let's take a look at the histogram of *loan_amnt* to get an idea of the distribution.

```
ggplot(lending_data_target_clean, aes(x = loan_amnt)) +
  geom_histogram(stat = "bin", bins = 50, color = "blue") +
  xlab("Loan Amount") +
  ylab("Number of Observations") +
  ggtitle("Histogram of Loan Amounts") +
  theme(plot.title = element_text(hjust = 0.50))
```



Looking at the histogram of *loan_amnt*, I cannot see any real distribution underlying the data. I believe that this could be due to people needing different loan amounts for different needs, therefore leaving no real pattern and distribution to follow.

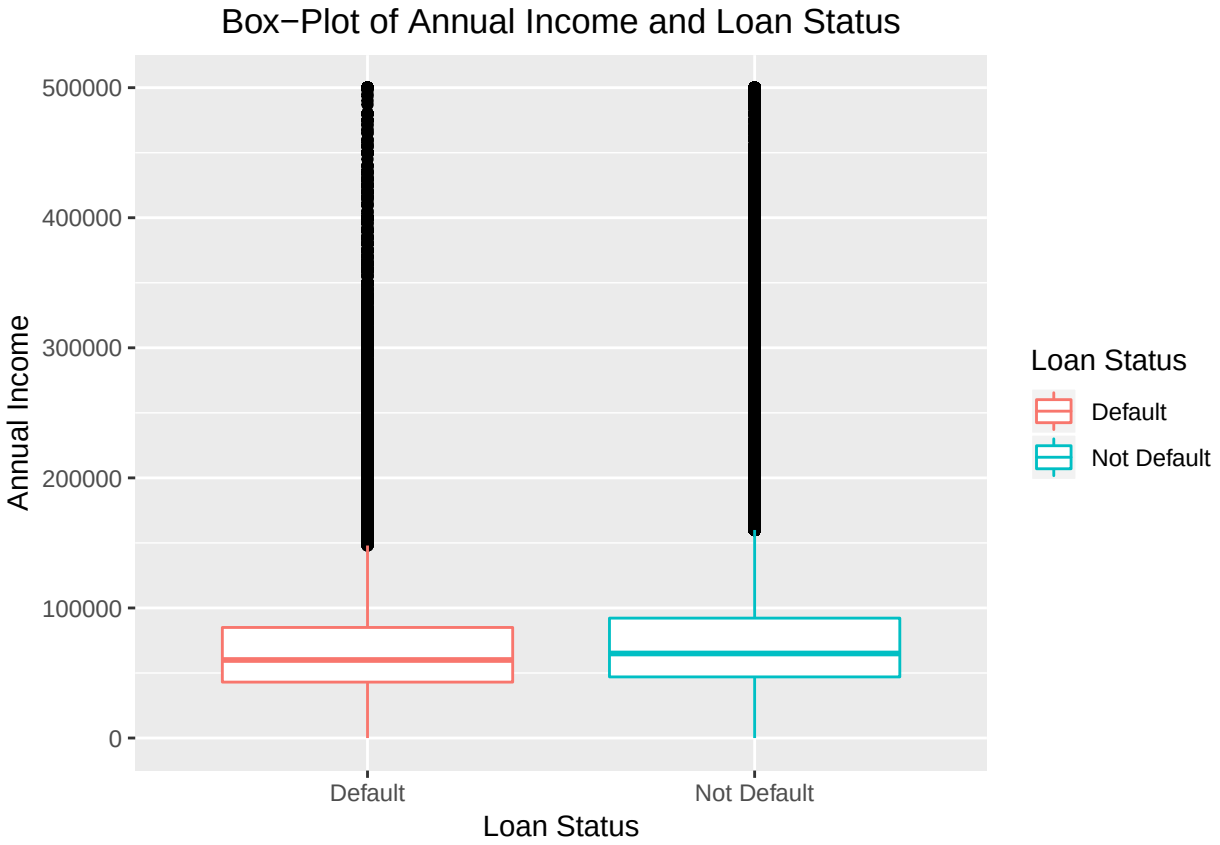
With the exploration of our target variable as well as three of my believed important variables, we are now going to take a look at the box-plot between each of these three variables, *annual_inc*, *int_rate*, and *loan_amnt*, and the output variable, *loan_status*. The box-plot will provide us a graph that will differentiate between each level of *loan_status* to take a look at the difference in mean and variance and overall density of each level.

Comparing X's vs Y

Let's first take a look at the box-plot between *annual_inc* and *loan_status*. Note that I will be once again limiting the annual income to a maximum of 500,000 to disregard outliers and obvious incorrect incomes.

```
ggplot(lending_data_target_clean, aes(x = loan_status_final, y = annual_inc, color = loan_status_final))
  geom_boxplot(outlier.colour = "black") +
  ylim(0, 500000) +
  labs(x = "Loan Status",
       y = "Annual Income",
       title = "Box-Plot of Annual Income and Loan Status",
       color = "Loan Status") +
  theme(plot.title = element_text(hjust = 0.50))
```

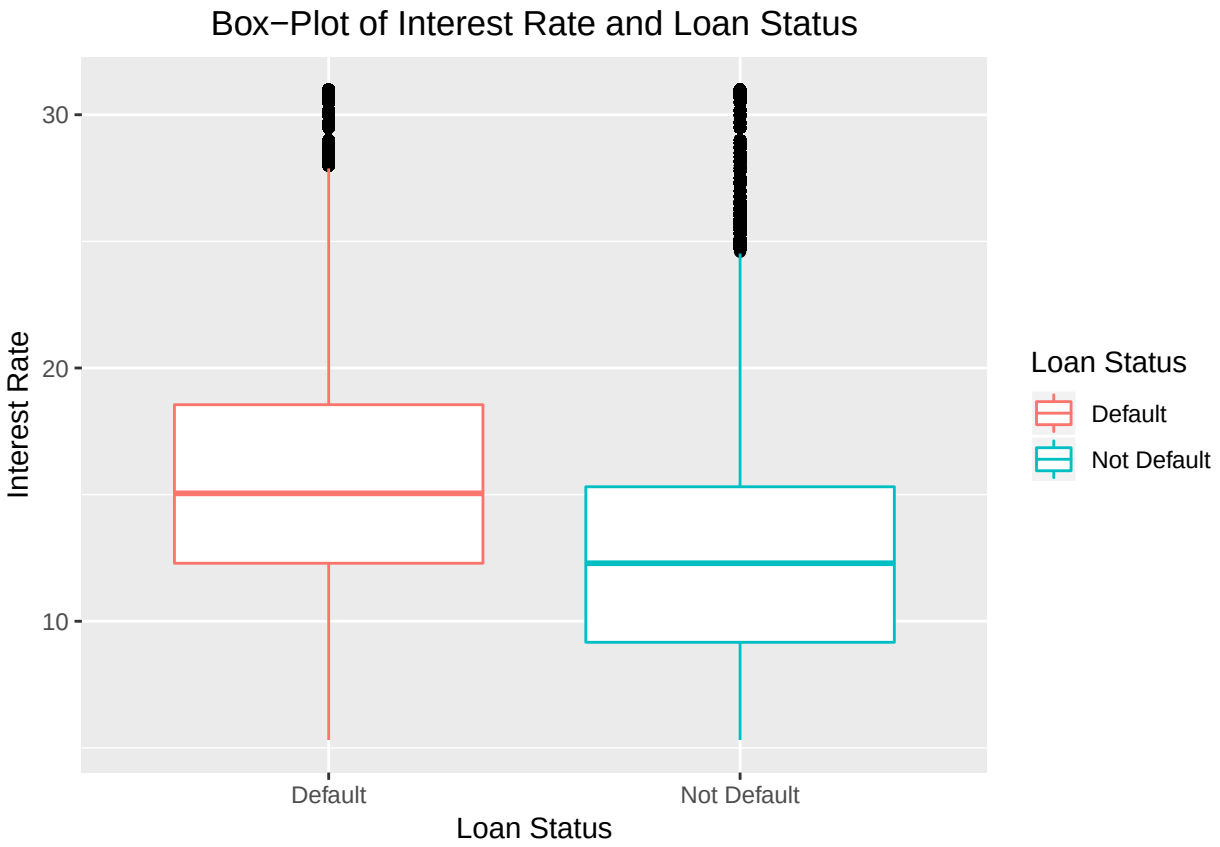
```
## Warning: Removed 1679 rows containing non-finite values (stat_boxplot).
```



Looking at the plot above, we can see that there is a very small difference between groups "Default" and "Not Default". This difference is about 7,000, as noted above when discussing the *annual_inc* variable. We can also see that there are many outliers in this variable, as noted by the observations that are black.

Let's now take a look at the box-plot between *int_rate* and *loan_status*.

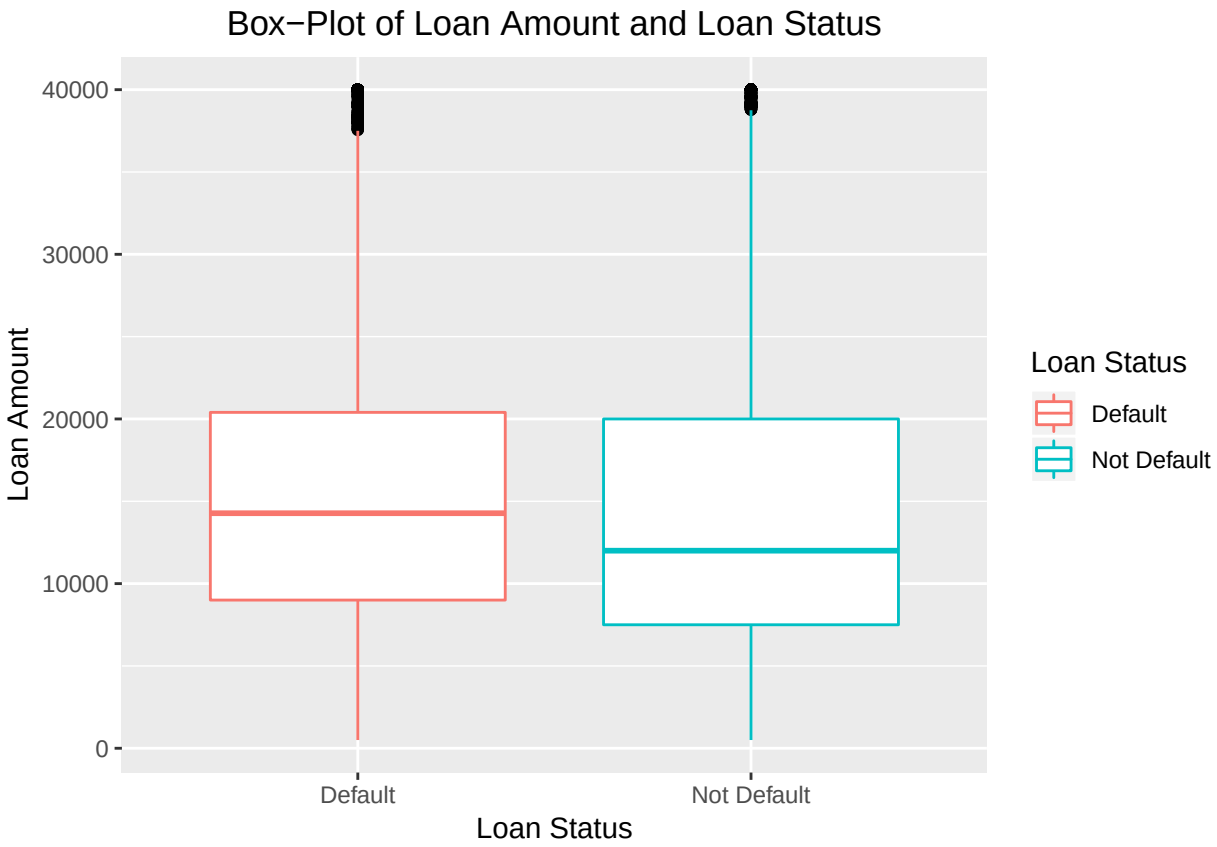
```
ggplot(lending_data_target_clean, aes(x = loan_status_final, y = int_rate, color = loan_status_final)) +
  geom_boxplot(outlier.colour = "black") +
  labs(x = "Loan Status",
       y = "Interest Rate",
       title = "Box-Plot of Interest Rate and Loan Status",
       color = "Loan Status") +
  theme(plot.title = element_text(hjust = 0.50))
```



Looking at the plot above, we can see that there is a large difference of interest rates between groups of "Default" and "Not Default". When looking at the outliers, we can see that the outliers of the "Default" group start at about an interest rate of 28% while then outliers of the "Not Default" group start at about an interest rate of 24.5%. This tells us that overall, the "Not Default" group has lower interest rates and a lower interest rate tends to point towards a loan that will not be defaulted on.

Lastly, let's take a look at the box-plot between *loan_amnt* and *loan_status*.

```
ggplot(lending_data_target_clean, aes(x = loan_status_final, y = loan_amnt, color = loan_status_final))
  geom_boxplot(outlier.colour = "black") +
  labs(x = "Loan Status",
       y = "Loan Amount",
       title = "Box-Plot of Loan Amount and Loan Status",
       color = "Loan Status") +
  theme(plot.title = element_text(hjust = 0.50))
```



Looking at the plot above, we can see that the loan amounts between each group does not have a huge difference. The outliers start at about the same amount of 37,000. However, the mean loan amount for the "Not Default" group is about 2,000 lower. The spread of the observations in the "Not Default" group is a little bit larger though.

All in all, I believe that these three variables will play an important role in predicting whether or not someone will default on their loan. This is shown through summary statistics, visualizations of variables, as well as box-plots between each variable and the target variable *loan_status*.