prosperLoanData EDA

Alonso Gutierrez

March 2, 2017

prosperLoanData EDA by Alonso Gutierrez

> **Tip**: One of the requirements of this project is that your code follows good formatting techniques, including limiting your lines to 80 characters or less. If you're using RStudio, go into Preferences > Code > Display to set up a margin line to help you keep track of this guideline!

```
## [1] "C:/Users/Alonso/Documents/Udacity/Data_Analysis/Exploratory_Data_Analysis/Data_Analysis_with_R/l
```

```
## [1] "es.Rmd"              "Final Project.Rmd"   "FInalProject2.Rmd"
## [4] "prosperLoanData.csv"
```

This data set contains 113,937 loans with 81 variables on each loan, including loan amount, borrower rate (or interest rate), current loan status, borrower income, borrower employment status, borrower credit history, and the latest payment information.

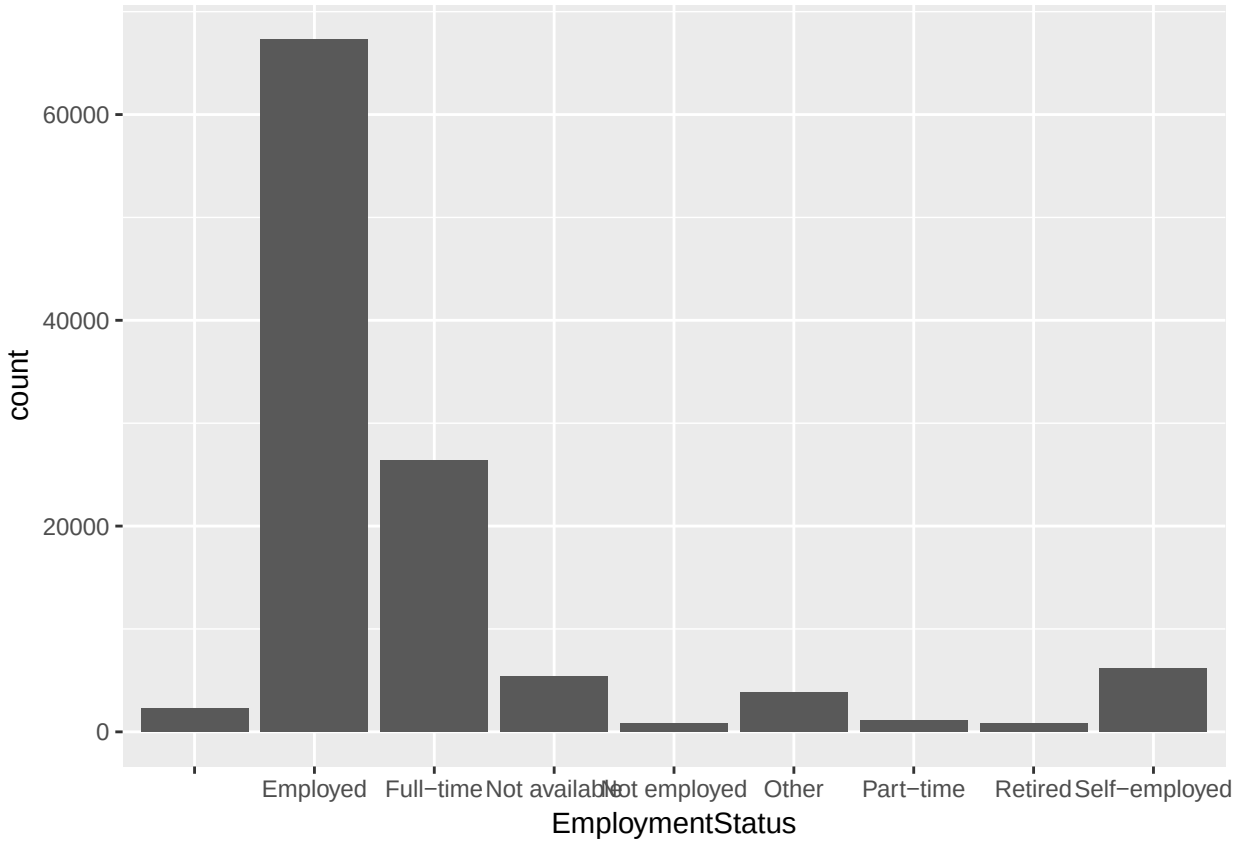
The idea here is to use exploratory data analysis in order to be able to show faulty borrowers in a better light, and be able to distinguish some or their most remarkable characteristics apart from those who do succeed to keep up on their payments, and finish.

By this data, does employment status at the time of listing have any visible correlation with loan status?

It is important to mention that the dataset distinguishes between borrowers that are 'Employed', 'Full time' and 'Part time'; this leads one to believe that this data maybe faulty since one category really is a set made up by the other two.

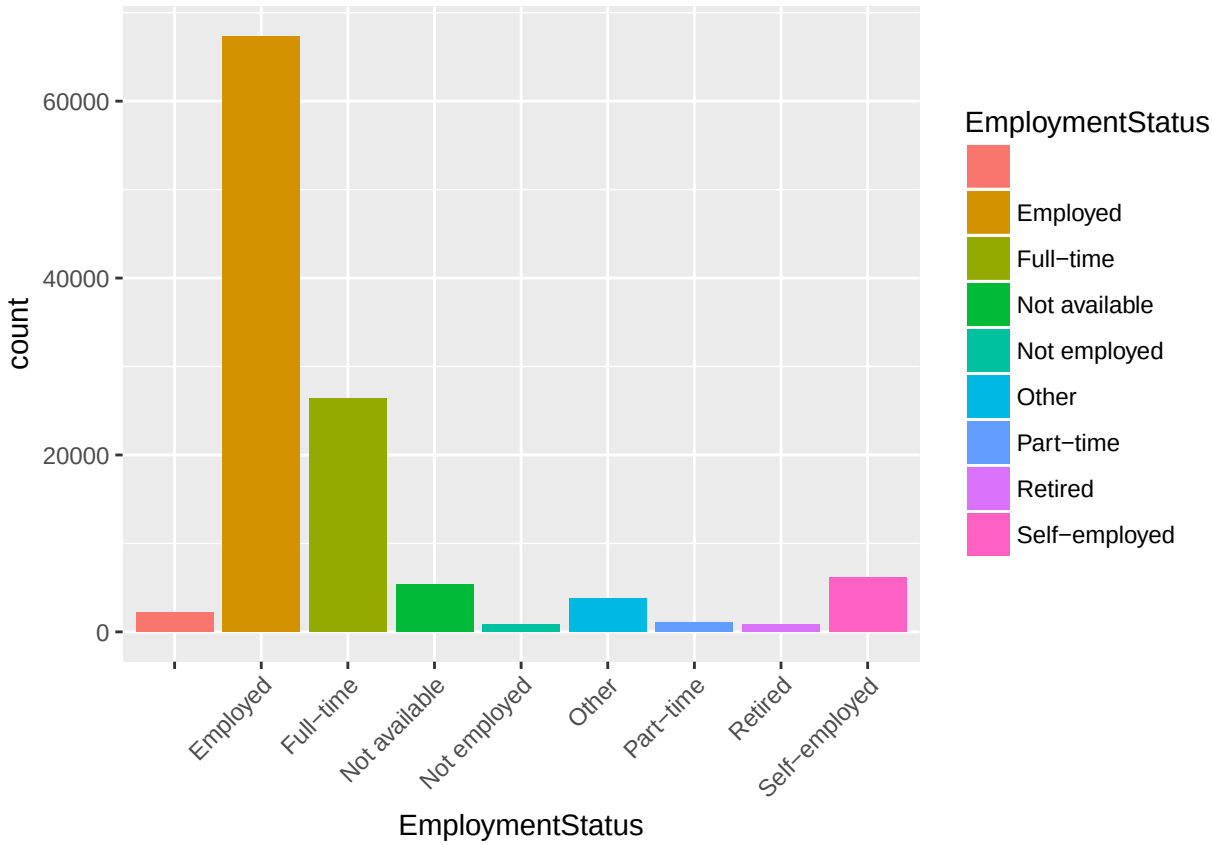
Univariate Plots

```
##  [1] "Cancelled"              "Chargedoff"
##  [3] "Completed"              "Current"
##  [5] "Defaulted"              "FinalPaymentInProgress"
##  [7] "Past Due (>120 days)"   "Past Due (1-15 days)"
##  [9] "Past Due (16-30 days)"  "Past Due (31-60 days)"
## [11] "Past Due (61-90 days)"  "Past Due (91-120 days)"
```



Here, it is easy to notice that the greater bulk of borrowers are made up of those who were employed at the time of loan listing. this group is made up of 'Employed', 'Full-time', 'Self-employed' in that order.

We can also see that that there are too many field nnames to fit on this small graph frame. we can use the theme function to tilt these labels, and the fill aesthetic for coloring like so:

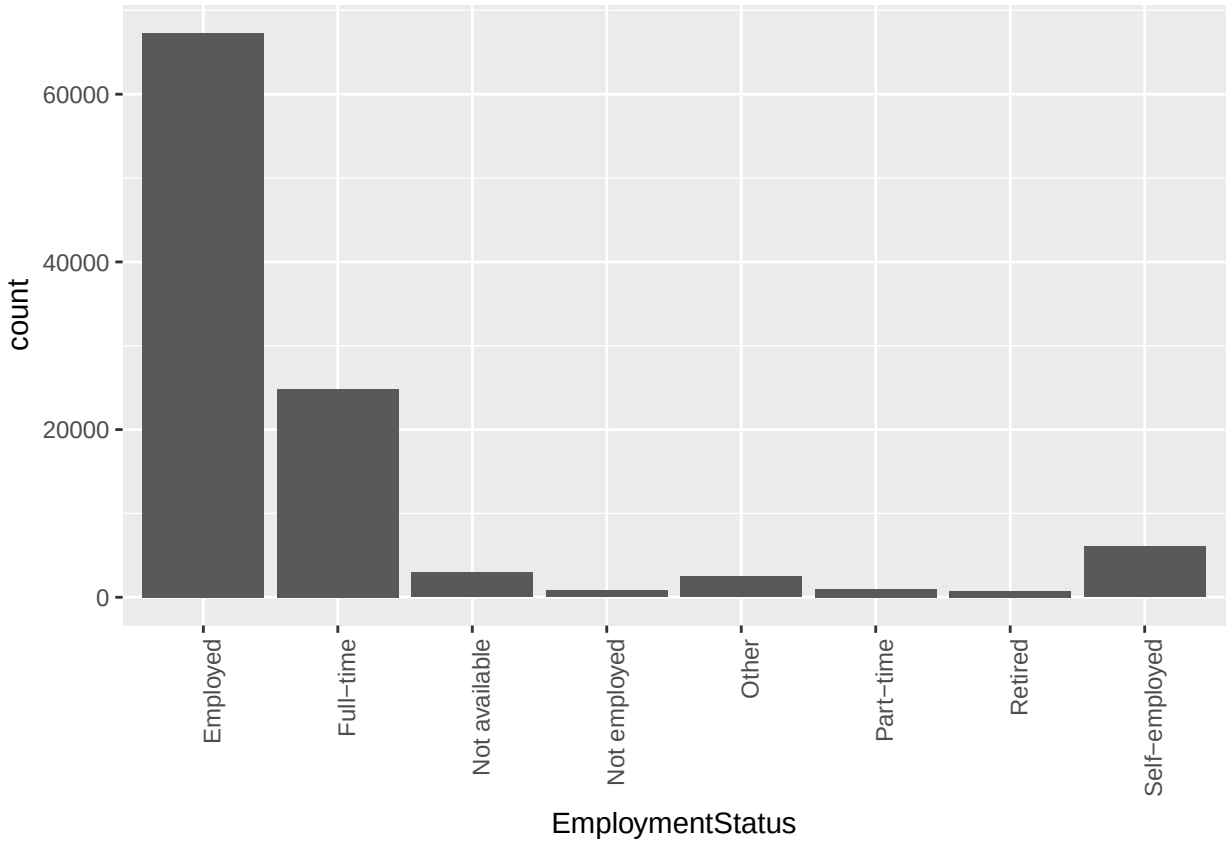


We notice here that there is an extra column in our histogram with no label. so we use the unique function to create a vector, named "c", which will contain all the unique labels of our rows.

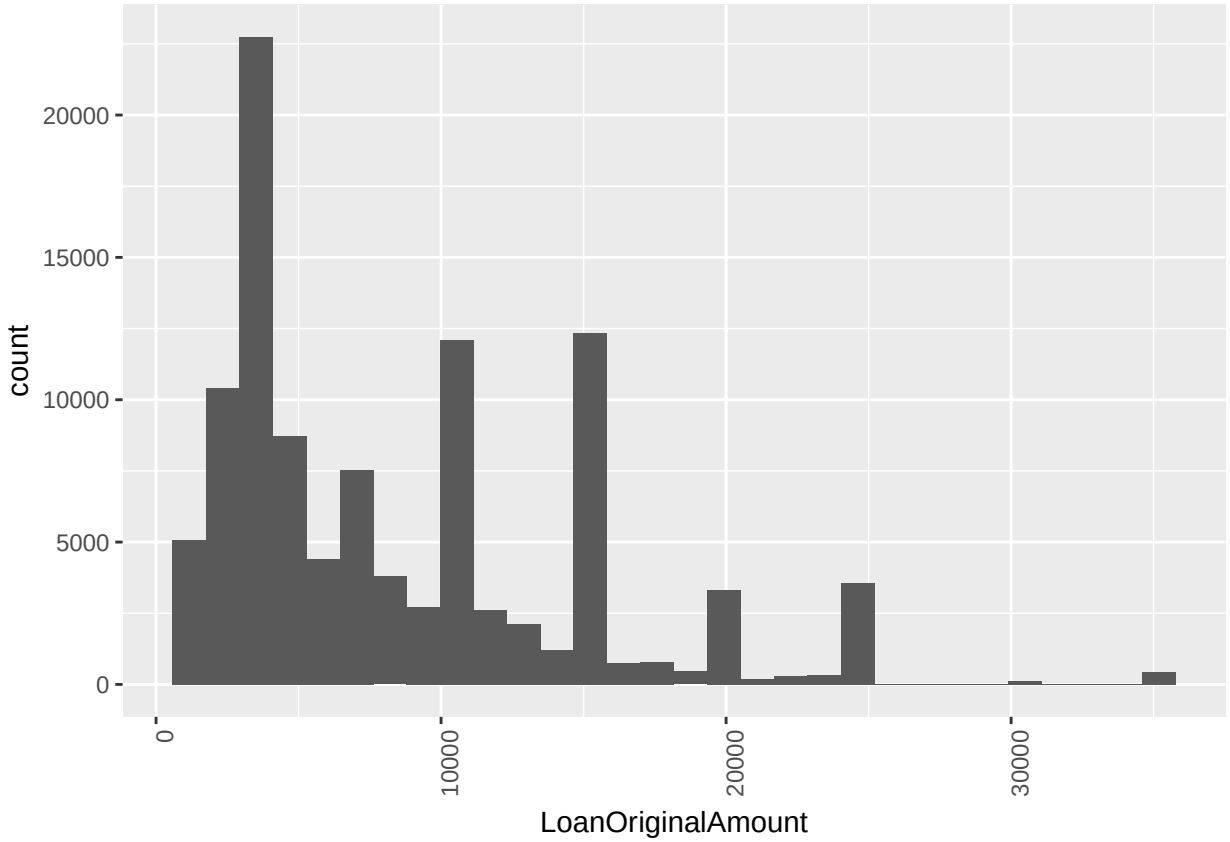
```
## [1] Self-employed Employed      Not available Full-time     Other
## [6]               Not employed  Part-time     Retired
## 9 Levels:  Employed Full-time Not available Not employed ... Self-employed
```

You can see that there is a graph column with no name (or ""). to get rid of this we save a subset of our initial data set to a variable, without these rows with values.

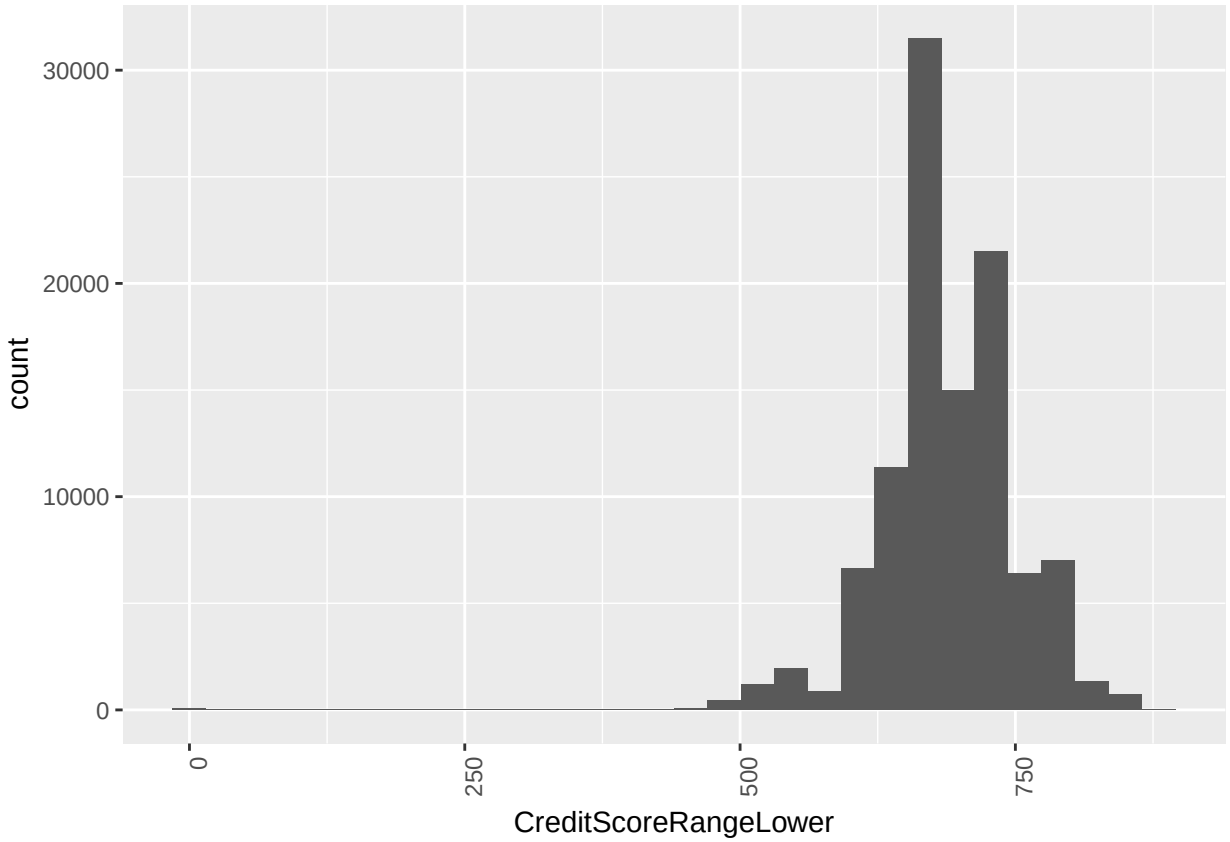
Now, let's Graph a few other variables



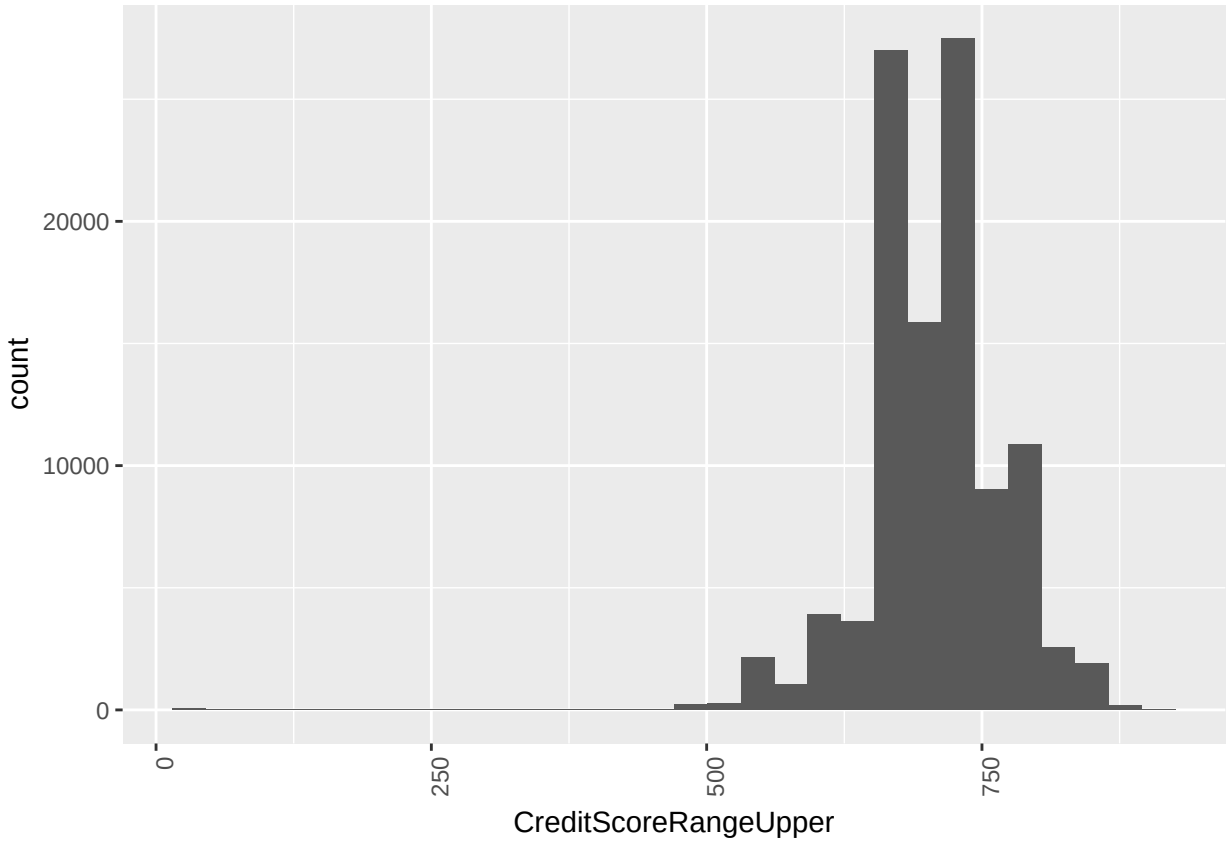

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



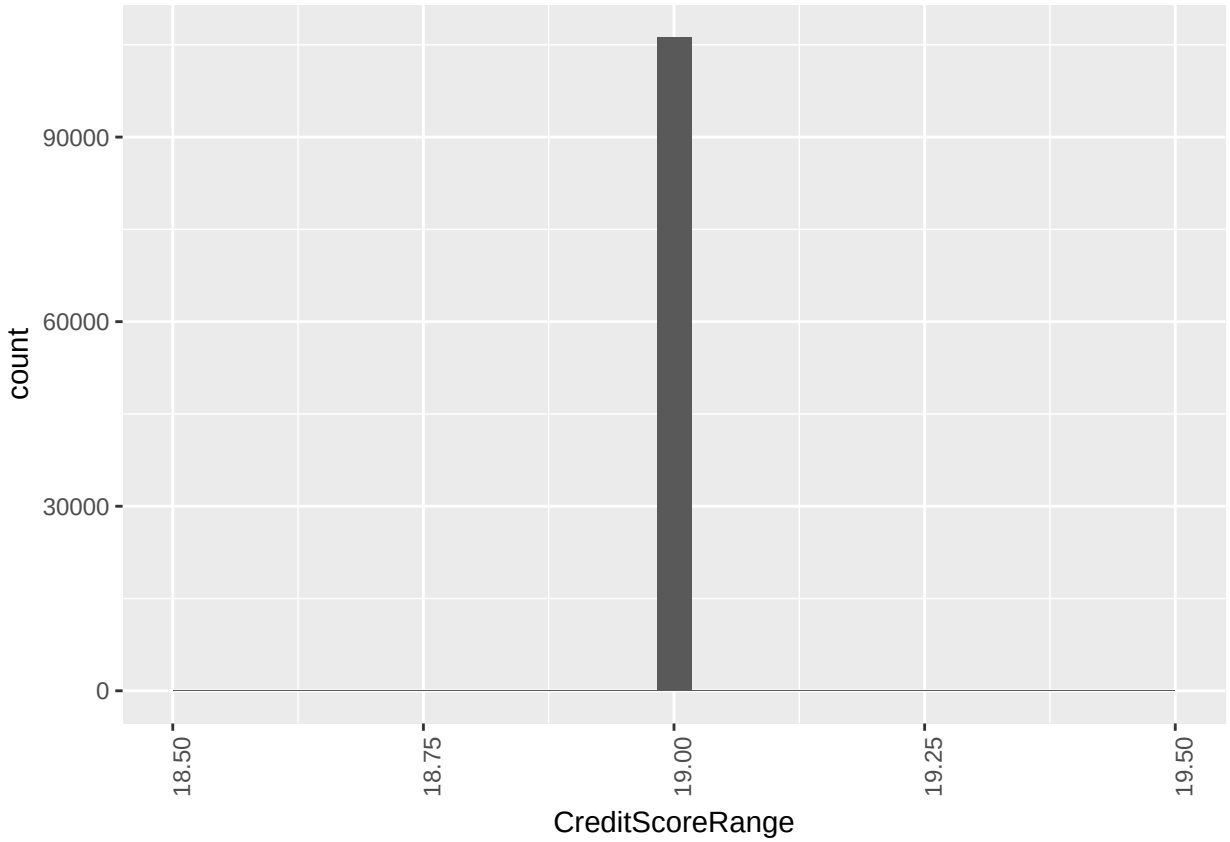

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

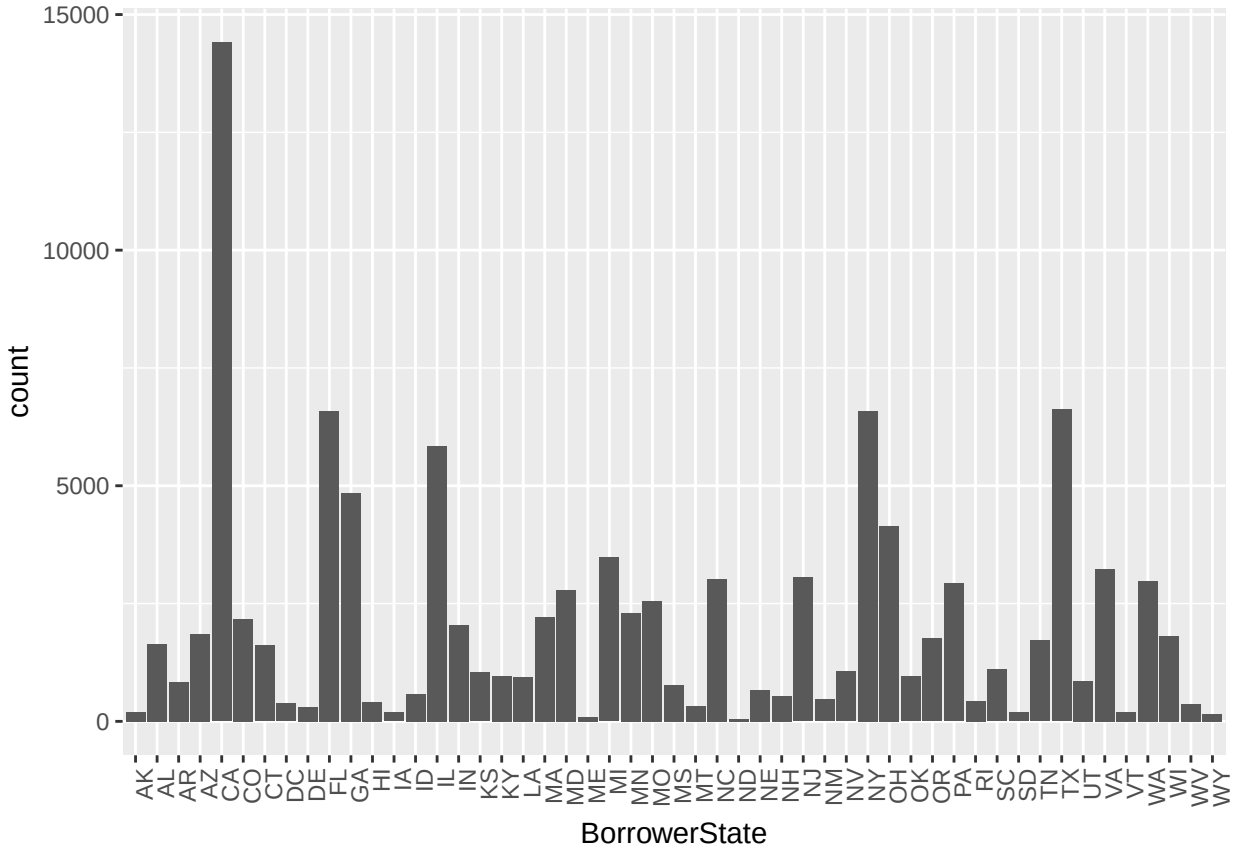


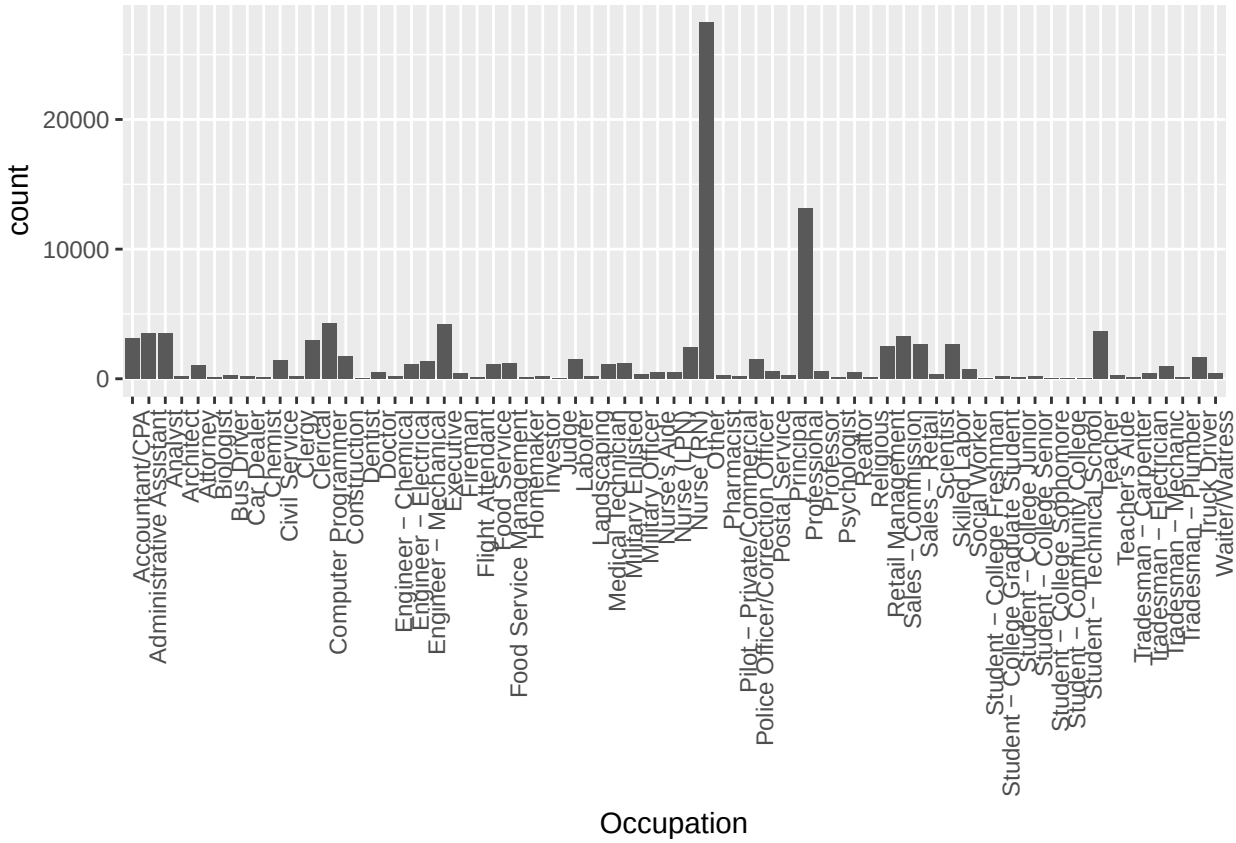

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

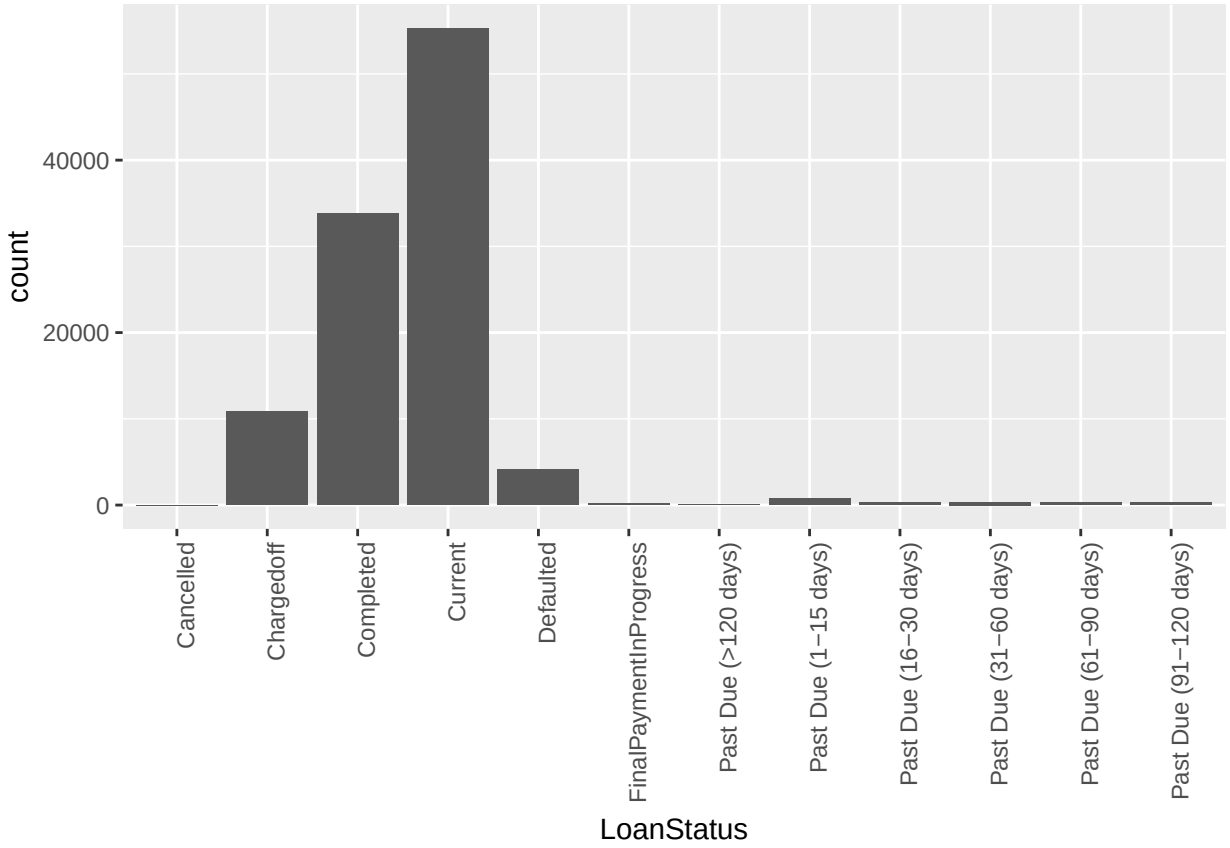



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


90000
60000
30000
0
count
18.50
18.75
19.00
19.25
19.50
CreditScoreRange


15000
10000
5000
0
count
AK
AL
AR
AZ
CA
CO
CT
DC
DE
FL
GA
HI
IA
ID
IL
IN
KS
KY
LA
MA
MD
ME
MI
MN
MO
MS
MT
NC
ND
NE
NH
NJ
NM
NV
NY
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VA
VT
WA
WI
WV
WY
BorrowerState


count
20000
10000
0
Accountant/CPA
Administrative Assistant
Analyst
Architect
Attorney
Biologist
Bus Driver
Car Dealer
Chemist
Civil Service
Clergy
Clerical
Computer Programmer
Construction
Dentist
Doctor
Engineer – Chemical
Engineer – Electrical
Engineer – Mechanical
Executive
Fireman
Flight Attendant
Food Service
Food Service Management
Homemaker
Investor
Judge
Laborer
Landscaping
Medical Technician
Military Enlisted
Military Officer
Nurse's Aide
Nurse (LPN)
Nurse (RN)
Other
Pharmacist
Pilot – Private/Commercial
Police Officer/Correction Officer
Postal Service
Principal
Professional
Professor
Psychologist
Realtor
Religious
Retail Management
Sales – Commission
Sales – Retail
Scientist
Skilled Labor
Social Worker
Student – College Freshman
Student – College Graduate Student
Student – College Junior
Student – College Senior
Student – College Sophomore
Student – Community College
Student – Technical School
Teacher
Teacher's Aide
Tradesman – Carpenter
Tradesman – Electrician
Tradesman – Mechanic
Tradesman – Plumber
Truck Driver
Waiter/Waitress
Occupation



Let's not forget that the fill aesthetic can be used to add color difference among levels of each factor variable on their respective graphs.

Univariate Analysis

What is the structure of your dataset?

My dataset is made up of 26 of the 81 total variables of the original dataset. I cut them short in the interest of time and memory space. Although more data is always better it seemed justifiable to cut this dataset to investigate a few variables at first since the discoveries in relations among so many variables could very well take a great amount of time to understand.

What is/are the main feature(s) of interest in your dataset?

My main features of interests are LoanStatus and EmploymentStatus, in order to find the best signs of borrowers defaulting on their payments, since logic follows that you will not have money to pay a loan if you do not have a job, which is most people's main source of income. Other features of interest may be: the type of job(Occupation) a borrower has, since different careers can make different amounts of money.Loan amount (LoanOriginalAmount) since it is harder to pay a bigger loan than it is a small one. Also, credit score, since it is a score system based on previous information of a borrower's likely of paying previous debt.

Did you create any new variables from existing variables in the dataset?

I created a new variable called CreditScoreRange which contains the range between the maximum credit score, and minimum credit score. I believe that there is a negative correlation between a borrower's likelyhood to pay off their loan and the magnitude of this number.

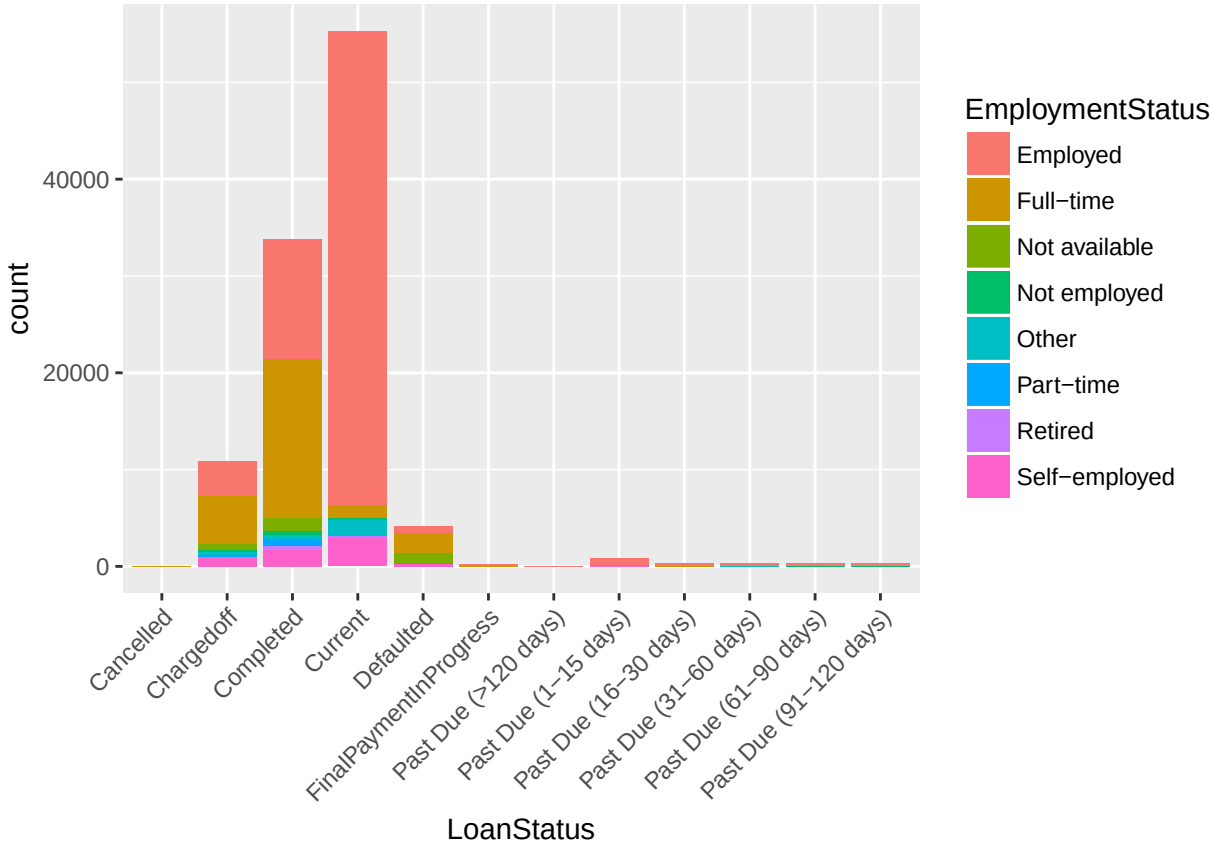
Of the features you investigated, were there any unusual distributions?
Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

As we move through these variables and graph their count per level, we noticed that certain categories in each graph are more likely to posess a higher count than others. For example, the employment graph demonstrates that most loans are given to those who were employed at the time the borrowers asked for the loan. The CreditScoreRangeLowre and CreditScoreRangeUpper categories show that most loan pertain to those borrower with credit scores roughly in the range of 500 to 750. the BorrowerState shows that California is the state with most borrowed loans. In the occupation variable graph we see that there is a high quantity of borrowers who wrote down 'Other' as their occupation, and the most popular loan Statuses are 'Chargedoff', 'Completed' and 'Current', as demonstrated in the LoanStatus graph.

Bivariate Plots

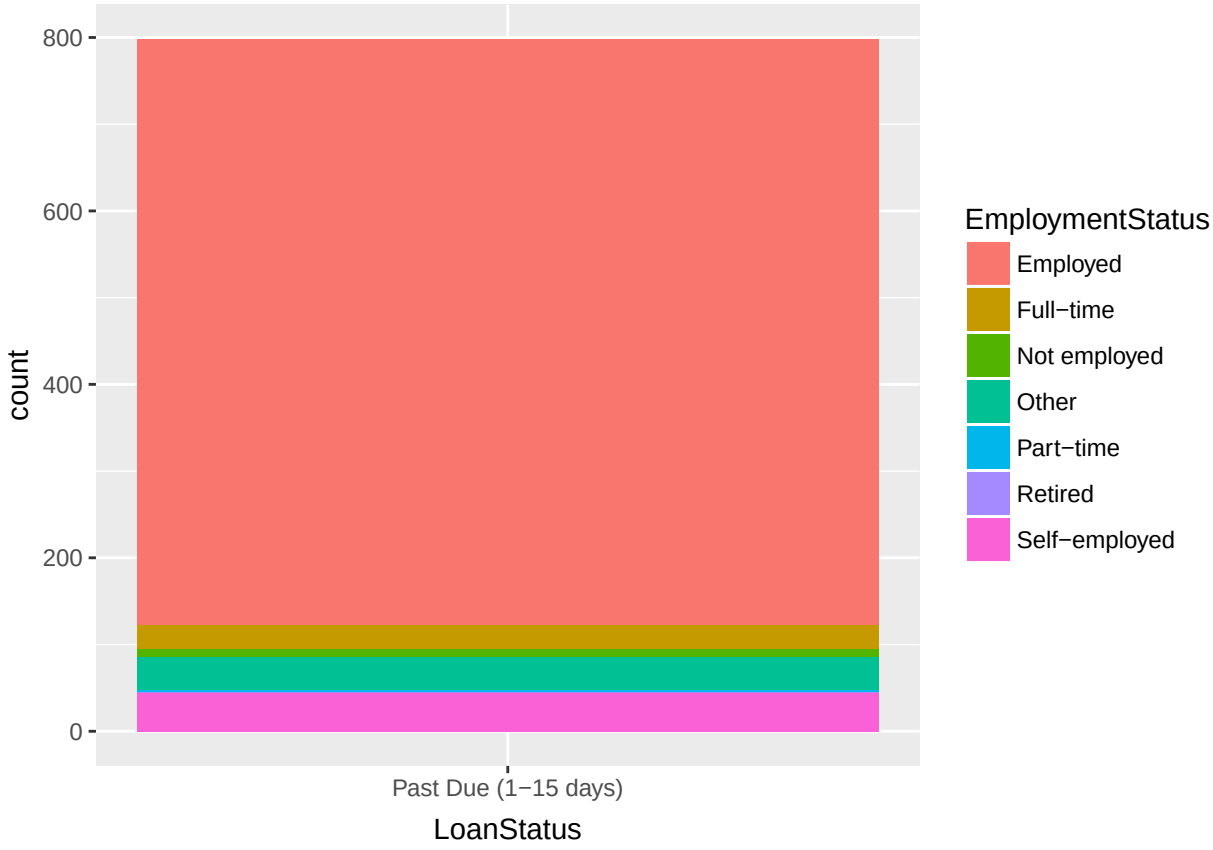
We can also use the fill aesthetic in order to distinguish among employment statuses in the count vs LoanStatus graph:

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

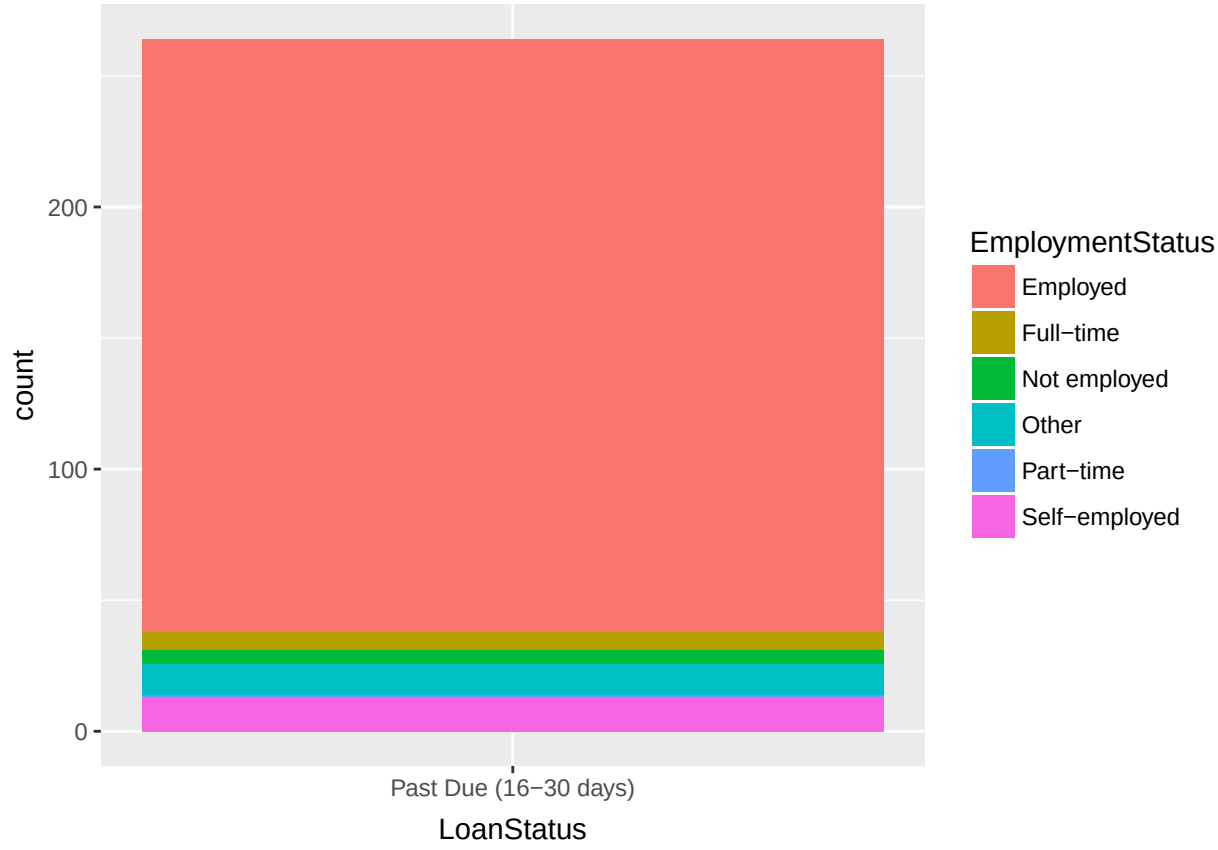


Let's take a look at those groups that have not comply with their payment deadlines. These include: "Defaulted" and all variations of "Past Due".

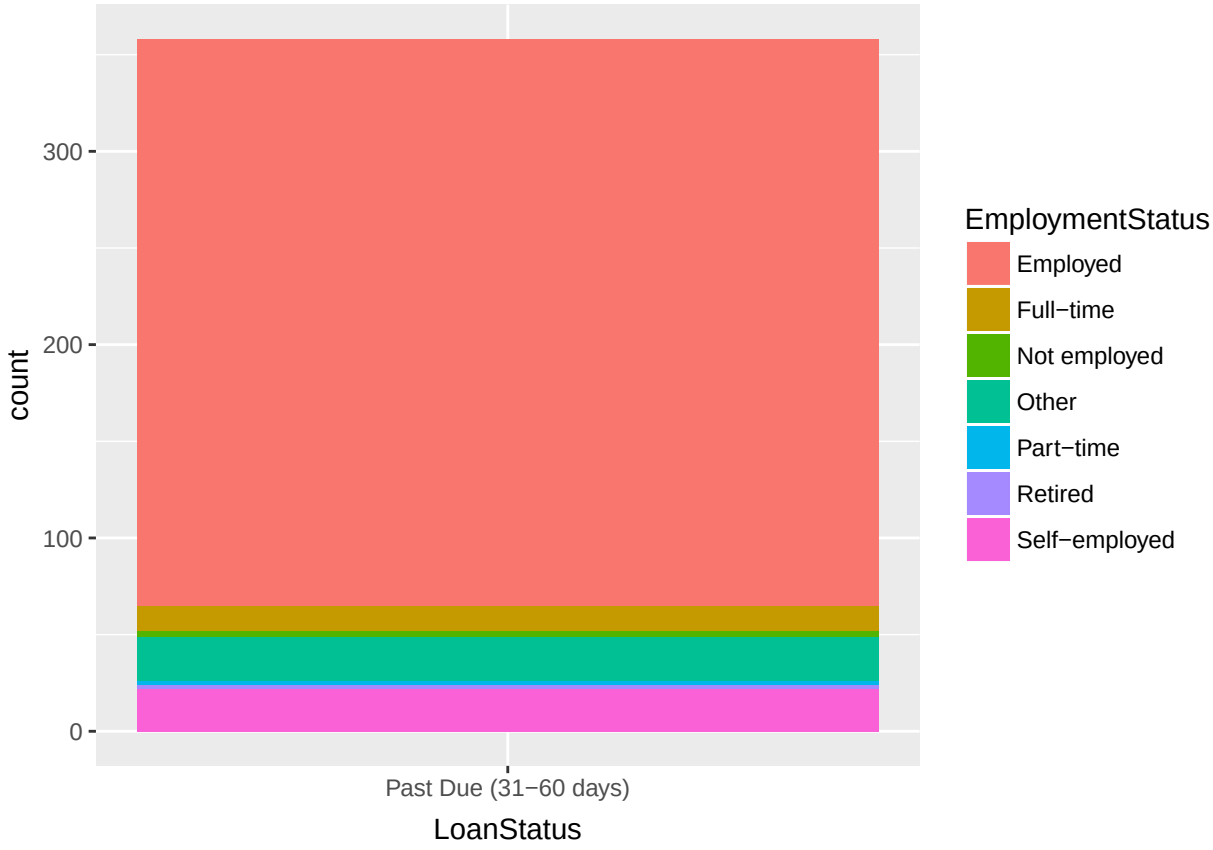
```
## Warning: Removed 105423 rows containing non-finite values (stat_count).
```



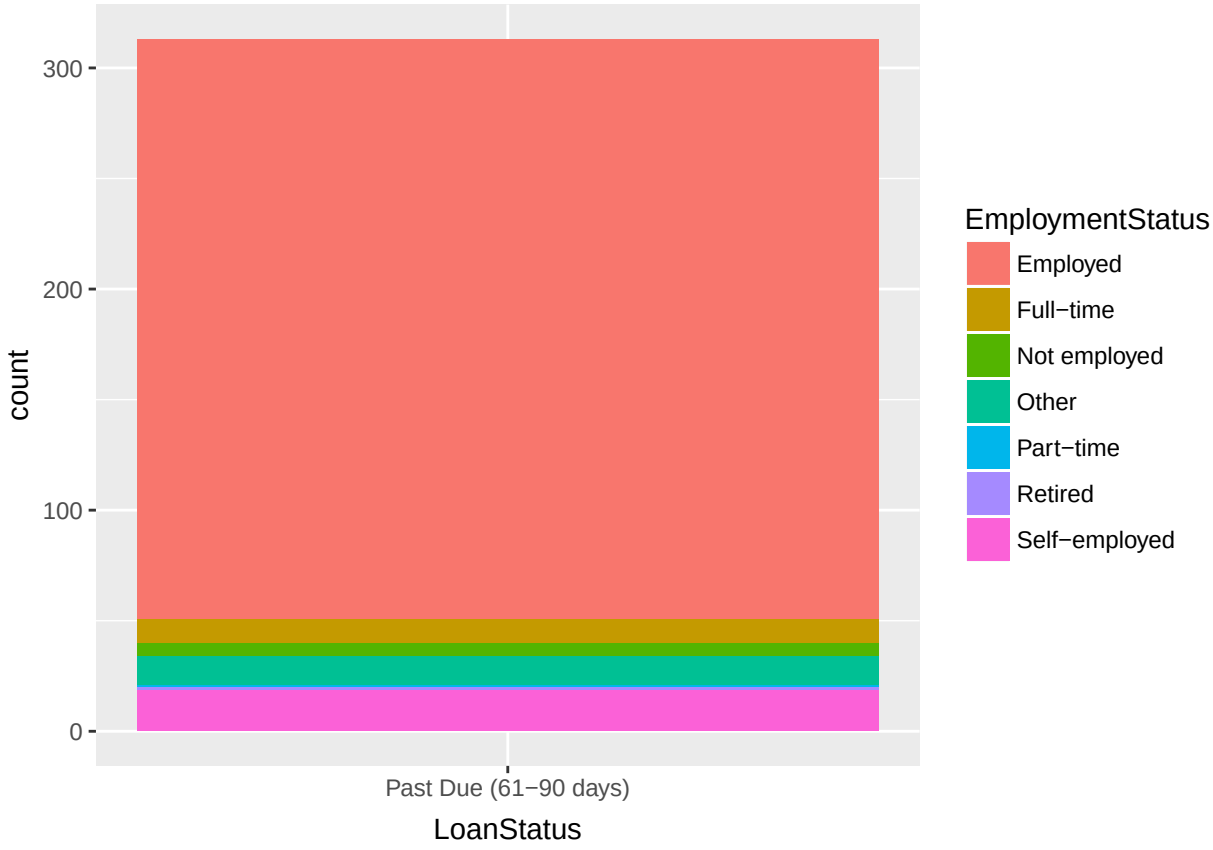

```
## Warning: Removed 105957 rows containing non-finite values (stat_count).
```



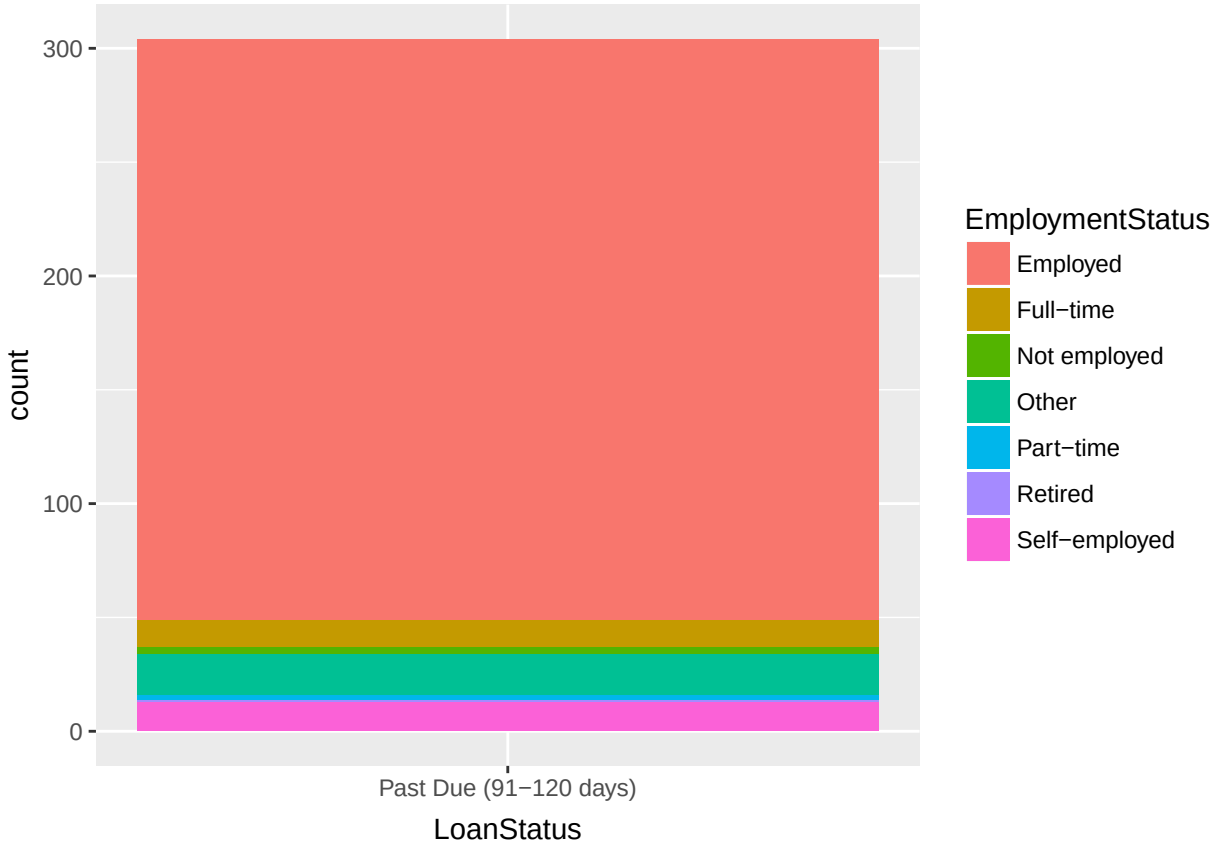

```
## Warning: Removed 105863 rows containing non-finite values (stat_count).
```



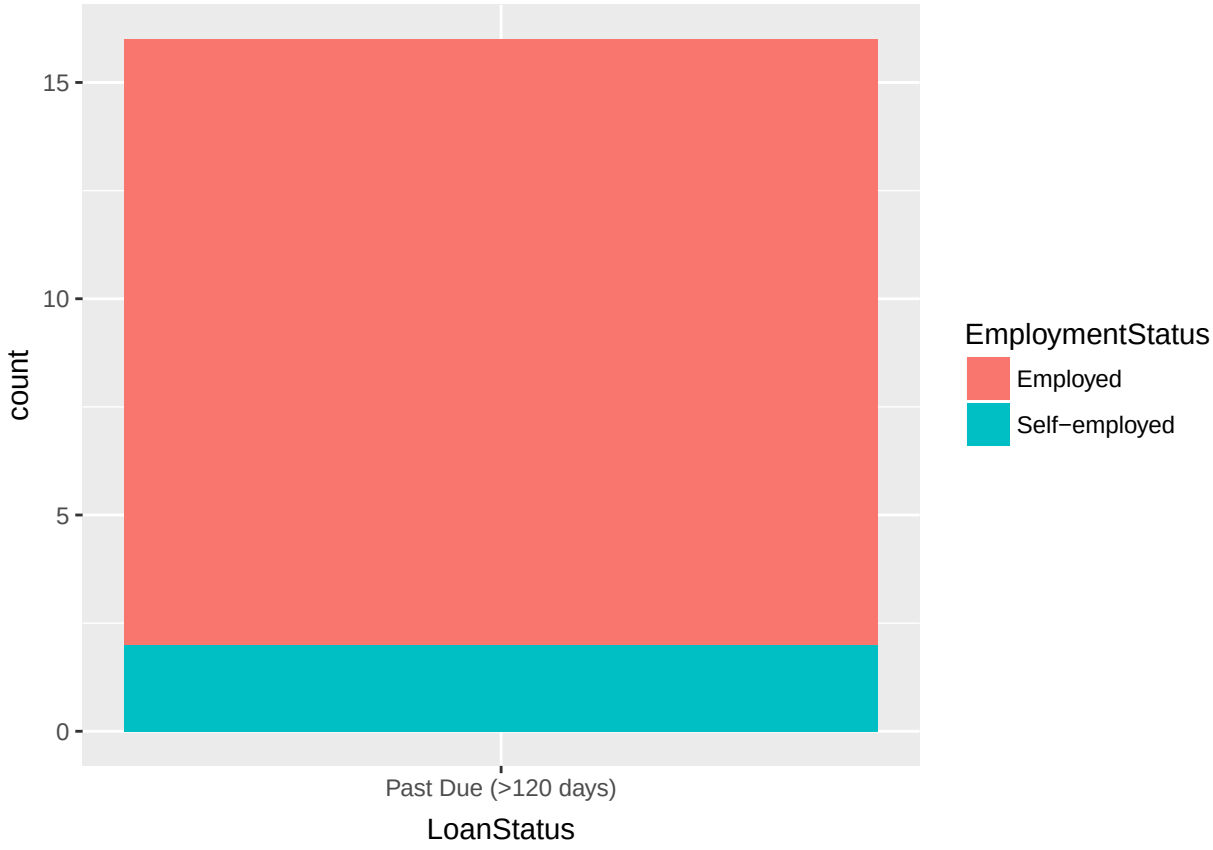

```
## Warning: Removed 105908 rows containing non-finite values (stat_count).
```



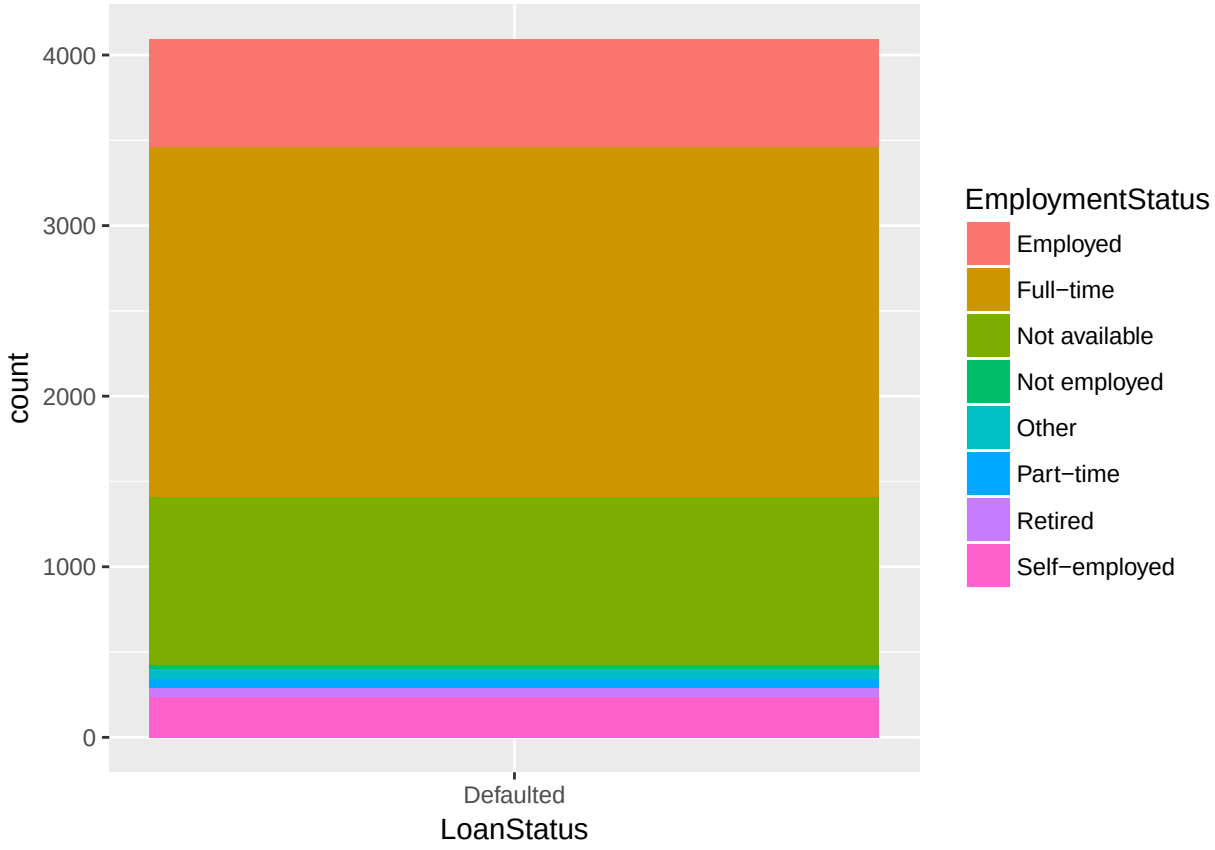

```
## Warning: Removed 105917 rows containing non-finite values (stat_count).
```




```
## Warning: Removed 106205 rows containing non-finite values (stat_count).
```

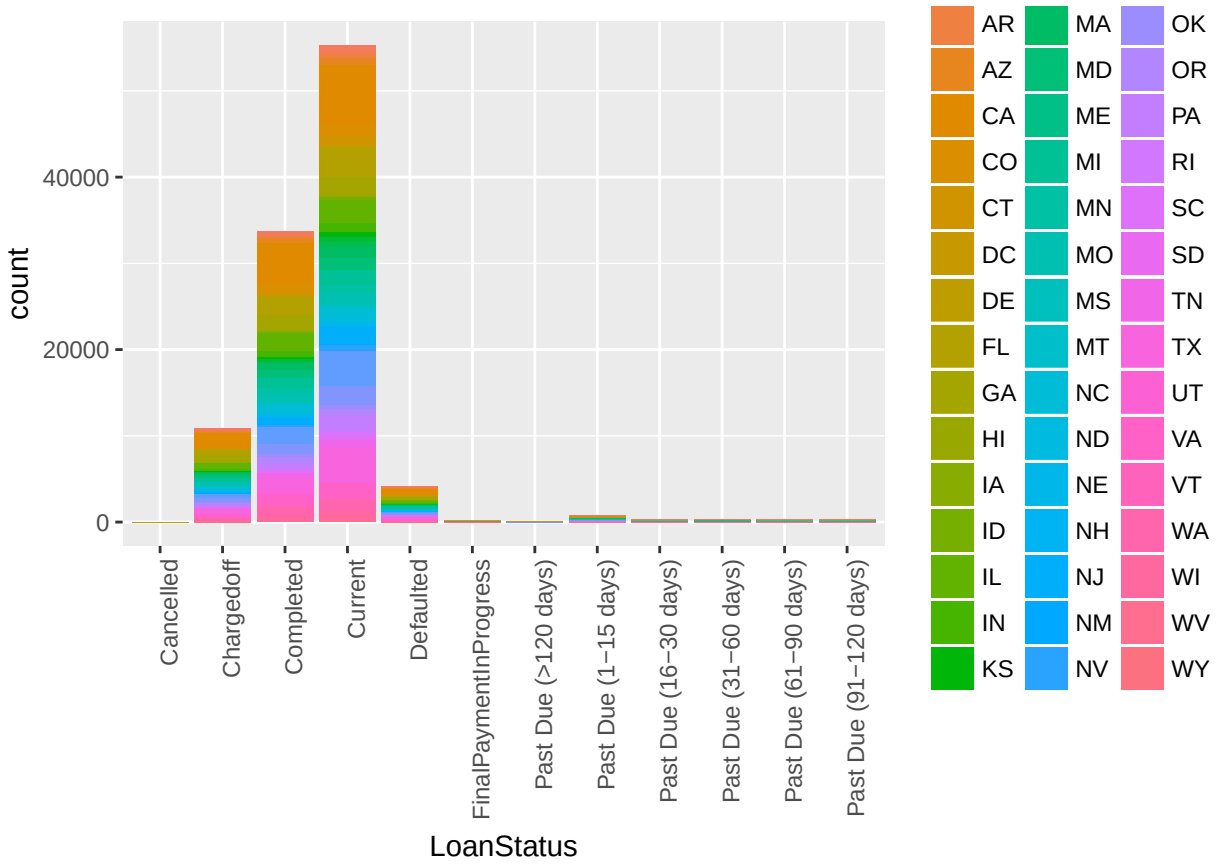



```
## Warning: Removed 102130 rows containing non-finite values (stat_count).
```



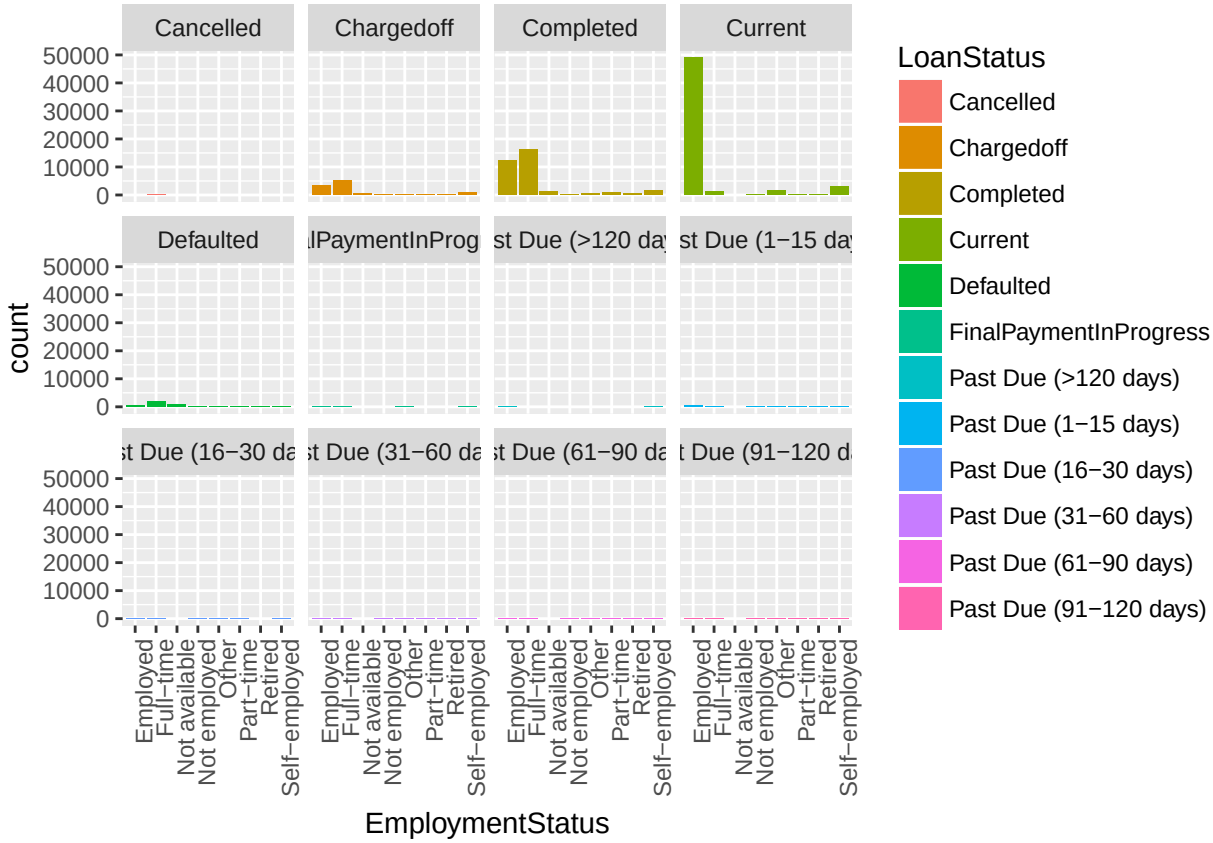
We can see that Employed borrowers make up the majority of all "Past Due" bars. Whereas "Full-time" and "Not Available" make up the majority of the "Defaulted" category.

Let's use the same strategy to contrast more than one supporting variable at a time. Here we did not use all of the variables that we graphed previously since some of these variable are continuos, integer data.

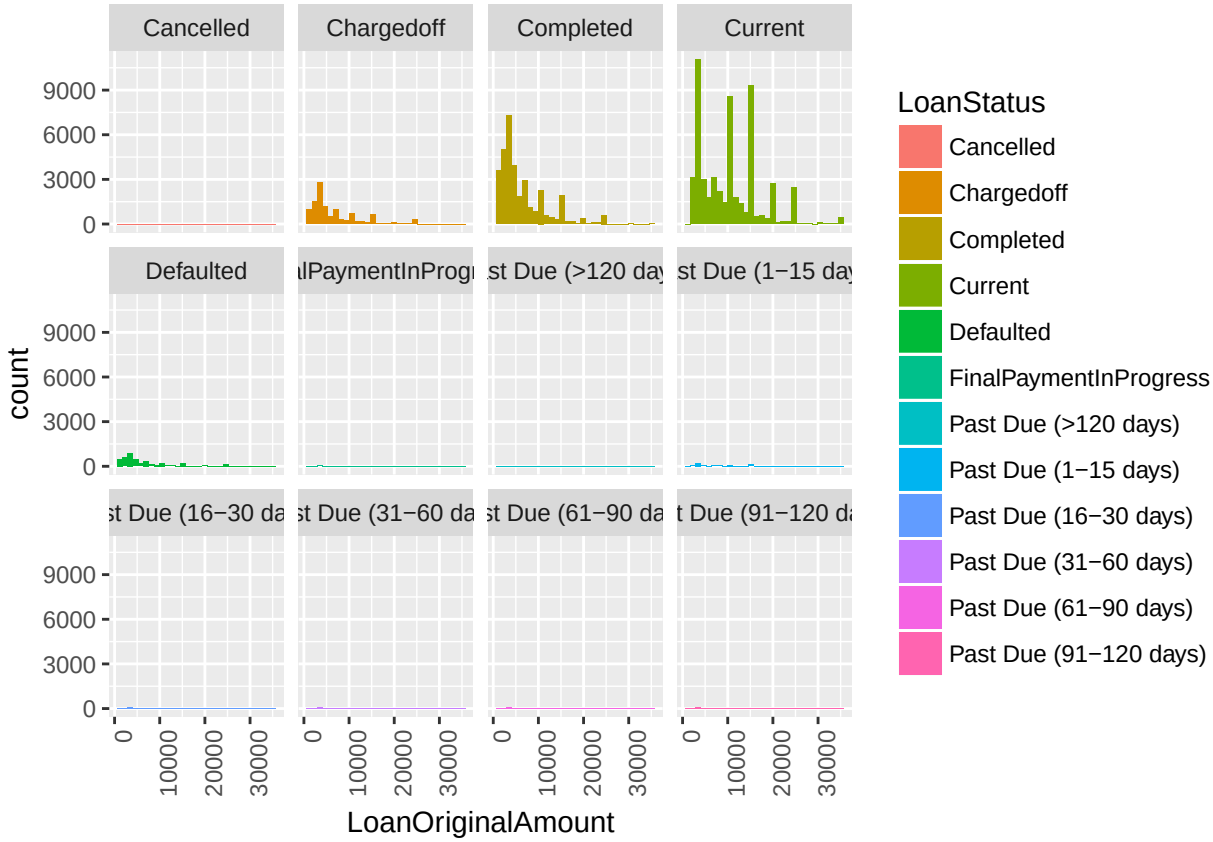


This is a very simple way to find the spread of categories, like those of "EmploymentStatus" accross the body of another variable's magnitude, LoanStatus vs Count.

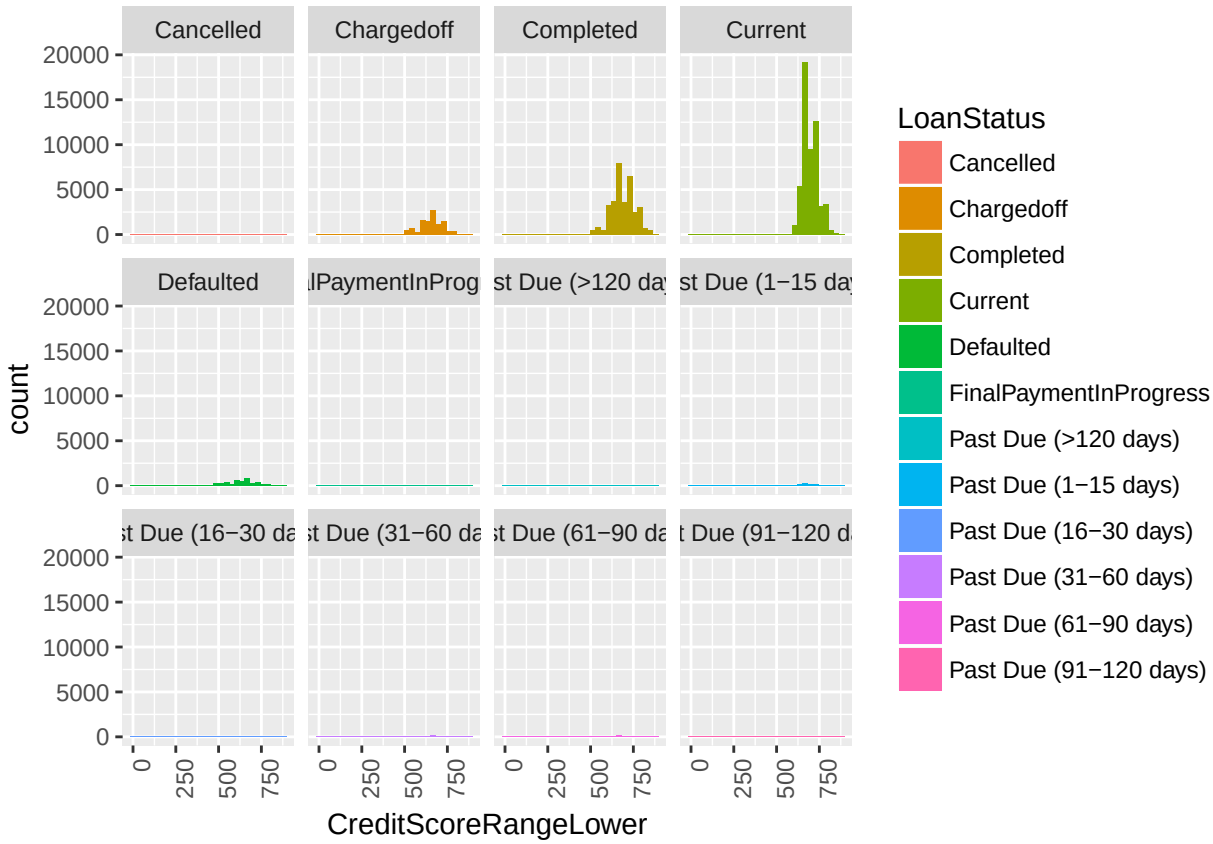
We could also employ ggplot's facet_wrap in order to make different graphs of any categorical variable vs Count accross all levels of "LoanStatus"



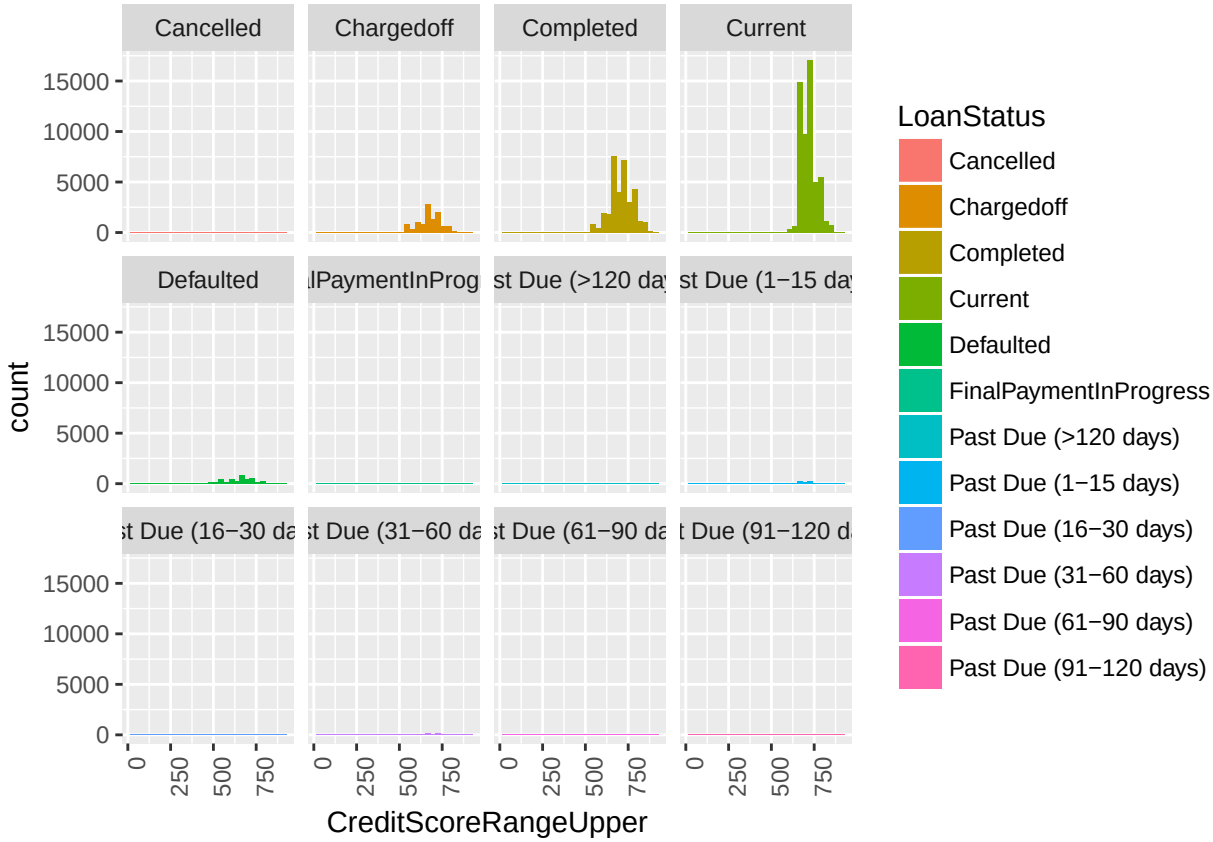

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



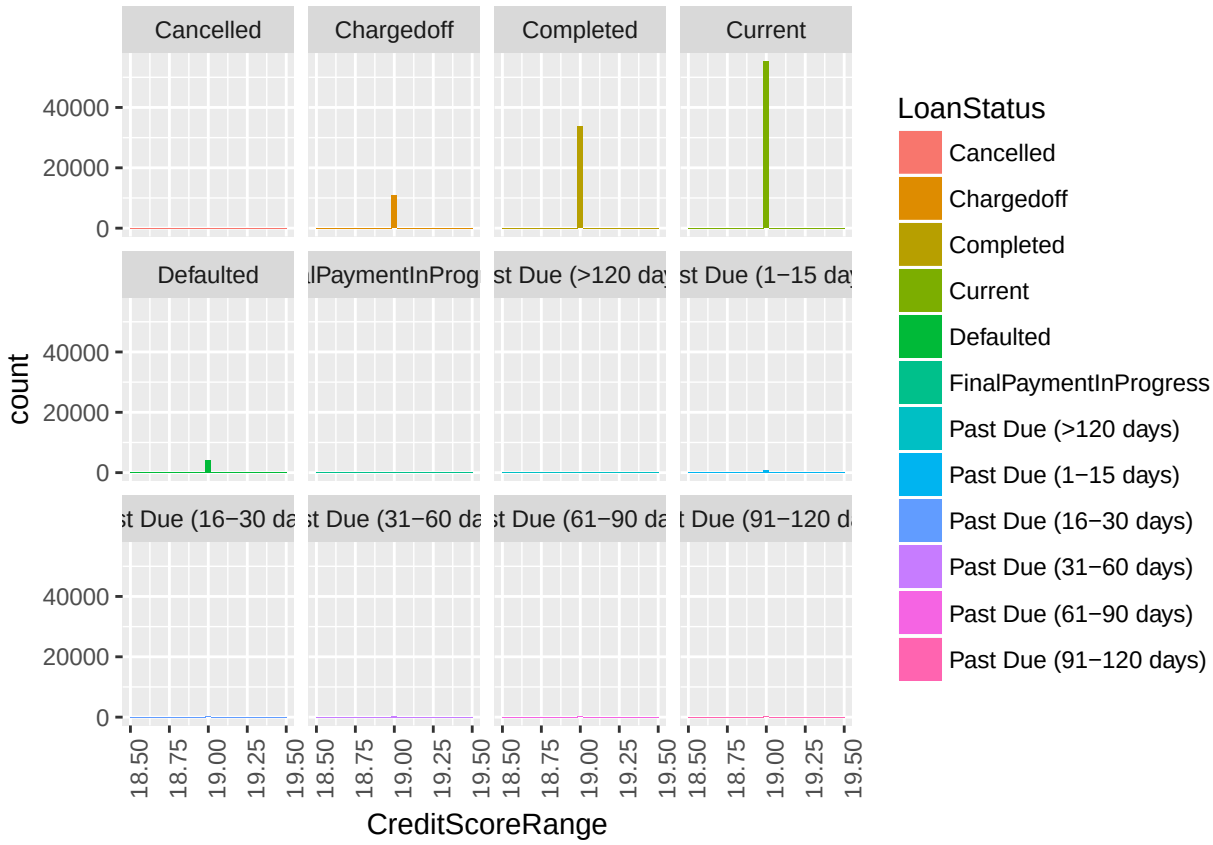

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

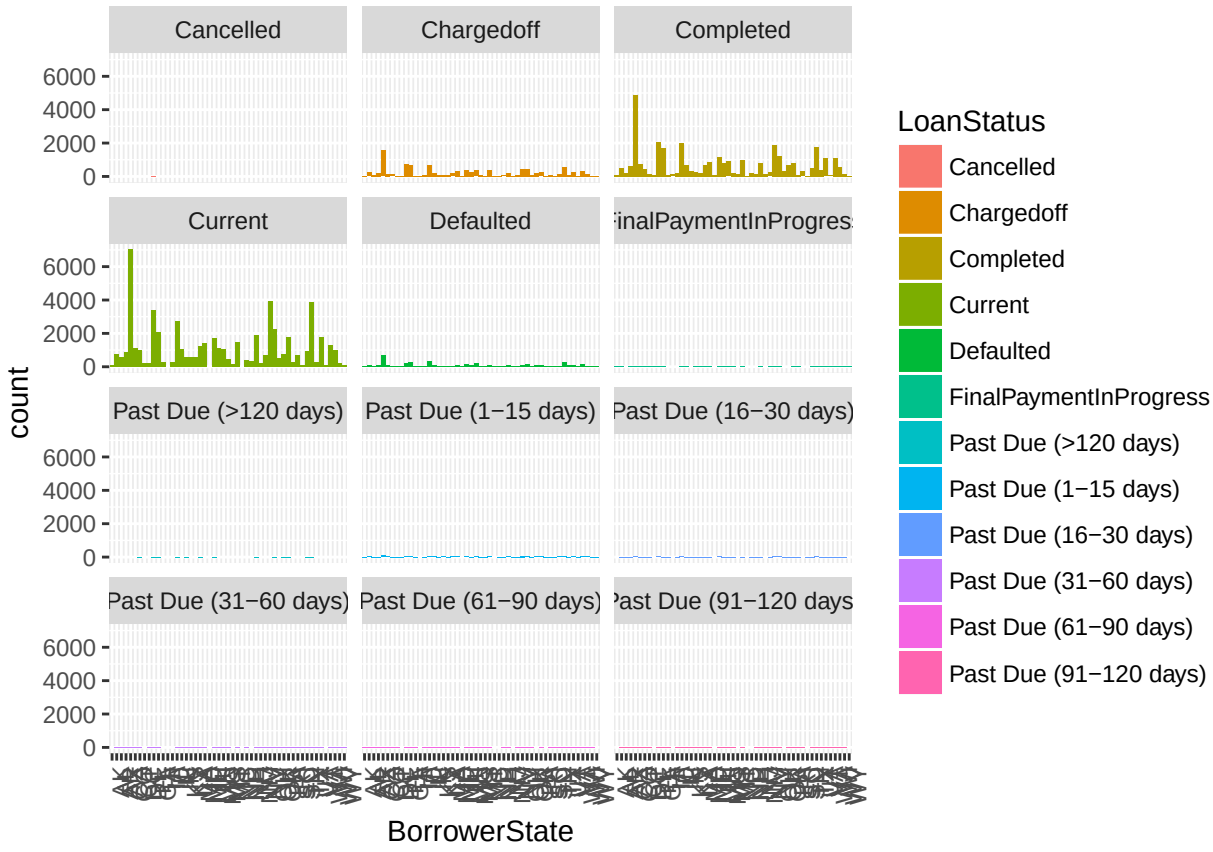


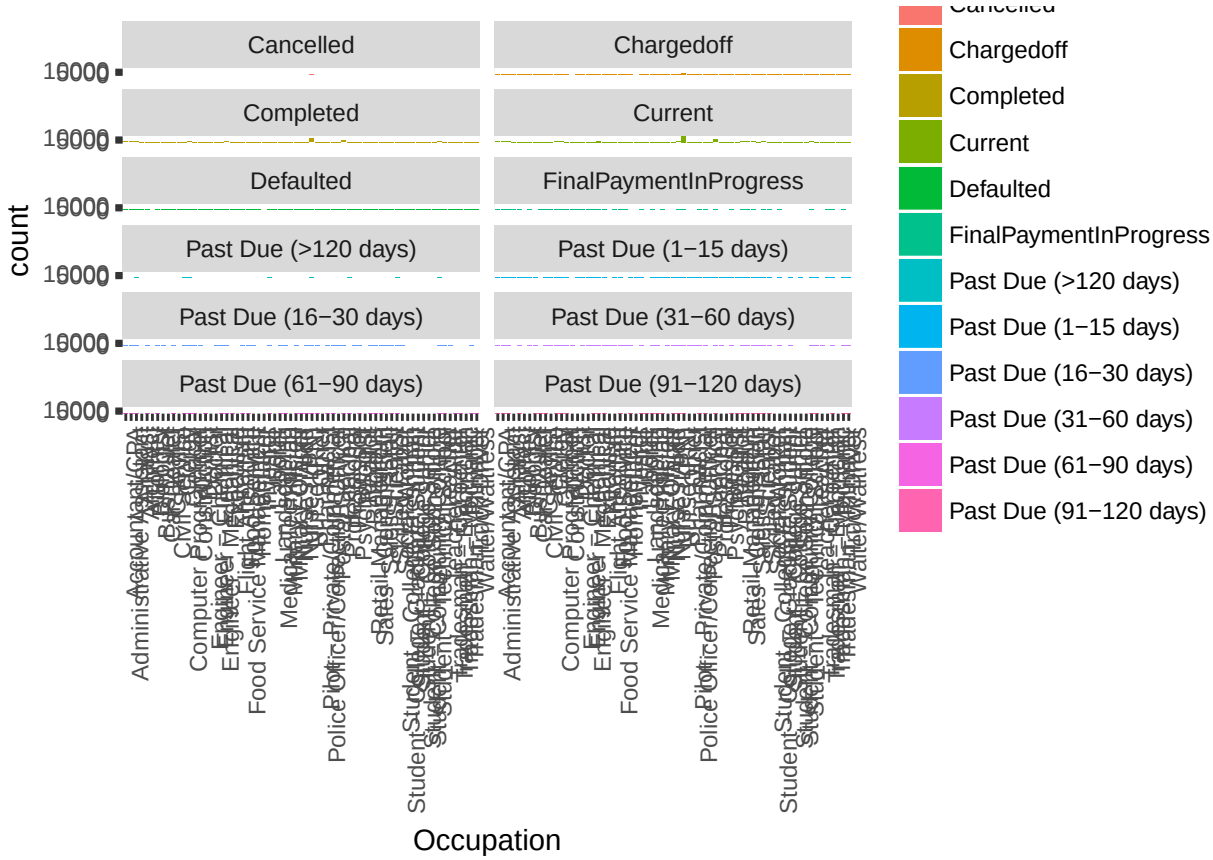

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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


Cancelled
Chargedoff
Completed
Current
Defaulted
lPaymentInProg
st Due (>120 da
st Due (1−15 da
t Due (16−30 da
t Due (31−60 da
t Due (61−90 da
t Due (91−120 d
count
40000
20000
0
18.50
18.75
19.00
19.25
19.50
CreditScoreRange
LoanStatus
Cancelled
Chargedoff
Completed
Current
Defaulted
FinalPaymentInProgress
Past Due (>120 days)
Past Due (1−15 days)
Past Due (16−30 days)
Past Due (31−60 days)
Past Due (61−90 days)
Past Due (91−120 days)


Cancelled
Chargedoff
Completed
Current
Defaulted
FinalPaymentInProgress
Past Due (>120 days)
Past Due (1−15 days)
Past Due (16−30 days)
Past Due (31−60 days)
Past Due (61−90 days)
Past Due (91−120 days)
count
6000
4000
2000
0
BorrowerState
LoanStatus



Now here, all graph groups are similar, except for the Occupation vs LoanStatus, and the Borrowerstate vs LoanStatus, whose facet_wrap argument contains the ncol argument in order to provide enough space for all categories marked at the bottom.

Here we get a more discrete representation of each supporting variables' description against the LoanStatus. It is easier to distinguish each column of count per supporting variable's level against the grouping by LoanStatus' levels than it was before. The fill aesthetic argument was introduced to make the bars more distinguishable.

Bivariate Analysis

Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

By simultaneously contrasting two supporting variables we can see their relationship with one another. When we analyzed individual columns of the LoanStatus vs EmploymentStatus we noticed that the main body of all "Past Due" variations are made up of employed borrower, and the "Defaulted" category is made up mainly of borrower who marke their employment status as "Full-time" and "Not Available".

Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

What was the strongest relationship you found?

Fast forward to the part we use ggplot and facet_wrap, we can see here that is easier to spot any spike in count of loans according to their loan status, but also contrasted against any of the other variables, and their respective levels. For example, in Count vs EmploymentStatus facet_wrapped around Loanstatus we can see that there are major spike for employed and fulltime borrowers under the categories of chargedoff, completed and current. Another example are the spikes in the Count vs Ocupation vs LoanStatus graph; these spikes represent the high count of borrowers under the "Other" Occupation status, in the "Chargedoff", "Current" and "Completed" categories of LoanStatus.

OVerall, these Bivariate graphing methods have served to better interpret the hidden relationships between variables. By classifying Data points further the addition of more discriminating factors we have brought a better understanding in our EDA rough draft, which may lead one to believe that consecutive addition of other variables will help understand this data set in new better ways.

Multivariate Plots

Since our subjects of interest are those which are past due on their payments, or have defaulted on their loans then we will go ahead and limit our data set to borrowers who fall under these categories.

```
csv_file.reassesedLS <-
  csv_file.reassesedEmp[which(
    csv_file.reassesedEmp$LoanStatus == 'Past Due (1-15 days)'|
    csv_file.reassesedEmp$LoanStatus =='Past Due (16-30 days)'|
    csv_file.reassesedEmp$LoanStatus =='Past Due (31-60 days)'|
    csv_file.reassesedEmp$LoanStatus =='Past Due (61-90 days)'|
    csv_file.reassesedEmp$LoanStatus =='Past Due (91-120 days)'|
    csv_file.reassesedEmp$LoanStatus =='Past Due (>120 days)'|
    csv_file.reassesedEmp$LoanStatus =='Defaulted'),]
```

We should notice the count reduction of borrowers from 106221 to 6144.

Let's make sure that only those levels of LoanStatus we are interested in have trickle down into our new dataset

```
unique(csv_file.reassesedLS$LoanStatus)
```

```
## [1] Past Due (1-15 days)   Defaulted              Past Due (16-30 days)
## [4] Past Due (61-90 days)  Past Due (31-60 days)  Past Due (91-120 days)
## [7] Past Due (>120 days)
## 12 Levels: Cancelled Chargedoff Completed Current ... Past Due (91-120 days)
```

LoanStatus and Employment:

Before we had notice that borrowers with a 'Employed' employment status made up most of the mass of past due loans (including all of its variations in past due time), as opposed to other employment statuses. We will also look at the body of employed borrowers with 'Defaulted' loans. Now let's take this information and contrast it against other variables.

Let's reassess our data set again to narrow it down to those employed borrowers that fall under this Loan statuses.

```
csv_file.reassesedLSandEmp <- subset(csv_file.reassesedLS,
                                     csv_file.reassesedLS$EmploymentStatus ==
                                       'Employed')
```

Since we have narrowed our dataset further down into loan listings of employed borrowers whose loan status is not favorable we will now investigate the relations of this new set to the other variables. Here we will make a boxplot of our loan amounts. We do so in order to look at the variability of loan amounts according to any given categorical variable that fall under 'Defaulted' and all 'Past Due' variations of status.Remember, these are all employed borrowers.

It would very handy to use GGally's ggpairs function in order to create a matrix of plots of all possible pairs within these variable.

let's reorganize our variables for a better visualization of our data.

```
csv_file.drops <- csv_file.reassesedLSandEmp[,c("LoanStatus",
                  "IncomeRange",
                  "IsBorrowerHomeowner",
                  "IncomeVerifiable",
                  "EmploymentStatus",
                  "FirstRecordedCreditLine",
                  "EmploymentStatusDuration",
                  "OpenCreditLines",
                  "DelinquenciesLast7Years",
                  "CreditScoreRange",
                  "CreditScoreRangeLower",
                  "CreditScoreRangeUpper",
                  "TotalCreditLinespast7years",
                  "TotalTrades",
                  "CurrentDelinquencies",
                  "TradesOpenedLast6Months",
                  "StatedMonthlyIncome",
                  "CurrentCreditLines",
                  "AmountDelinquent",
                  "DebtToIncomeRatio",
                  "LoanOriginalAmount")]
```

We reorganized all columns in csv.drops in order to better observe the relations within variables once we use ggpairs.

ggpairs

```
ggpairs(csv_file.drops)
```

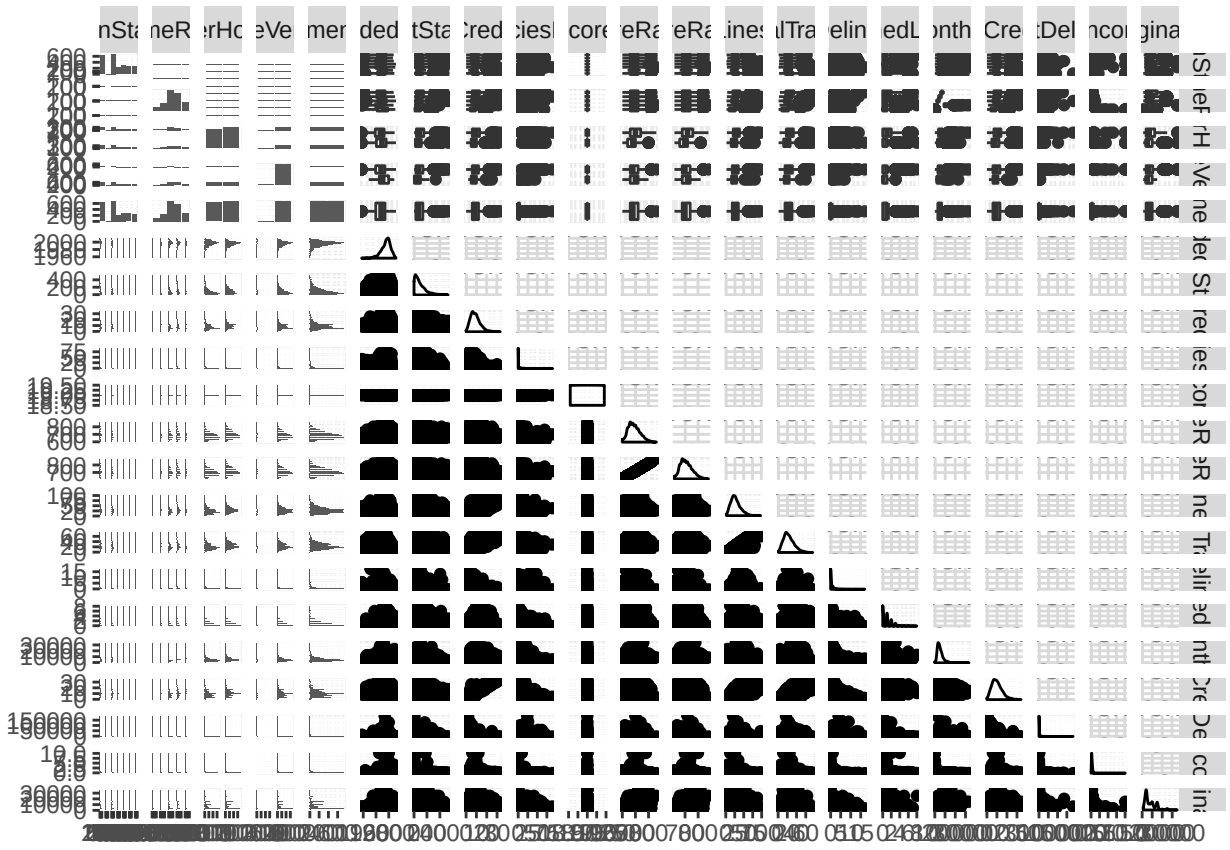
```
## Warning: Removed 85 rows containing non-finite values (stat_boxplot).

## Warning: Removed 85 rows containing non-finite values (stat_boxplot).

## Warning: Removed 85 rows containing non-finite values (stat_boxplot).

## Warning: Removed 85 rows containing non-finite values (stat_boxplot).

## Warning: Removed 85 rows containing non-finite values (stat_boxplot).

## `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`.

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero
```

```
## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

## `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`.

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values

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

## Warning: Removed 85 rows containing non-finite values (stat_bin).

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

## Warning: Removed 85 rows containing non-finite values (stat_bin).

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

## Warning: Removed 85 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 85 rows containing non-finite values (stat_bin).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 85 rows containing non-finite values (stat_bin).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing missing values (geom_point).
## Warning: Removed 85 rows containing non-finite values (stat_density).
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 85 rows containing missing values
## `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`.
## Warning: Removed 85 rows containing missing values (geom_point).
```



We payed particular attention to the correlation values of all continuous variables to spot the simplest relations to understand. It seem as if there aren't many of these which can be related.

GGally's ggduo

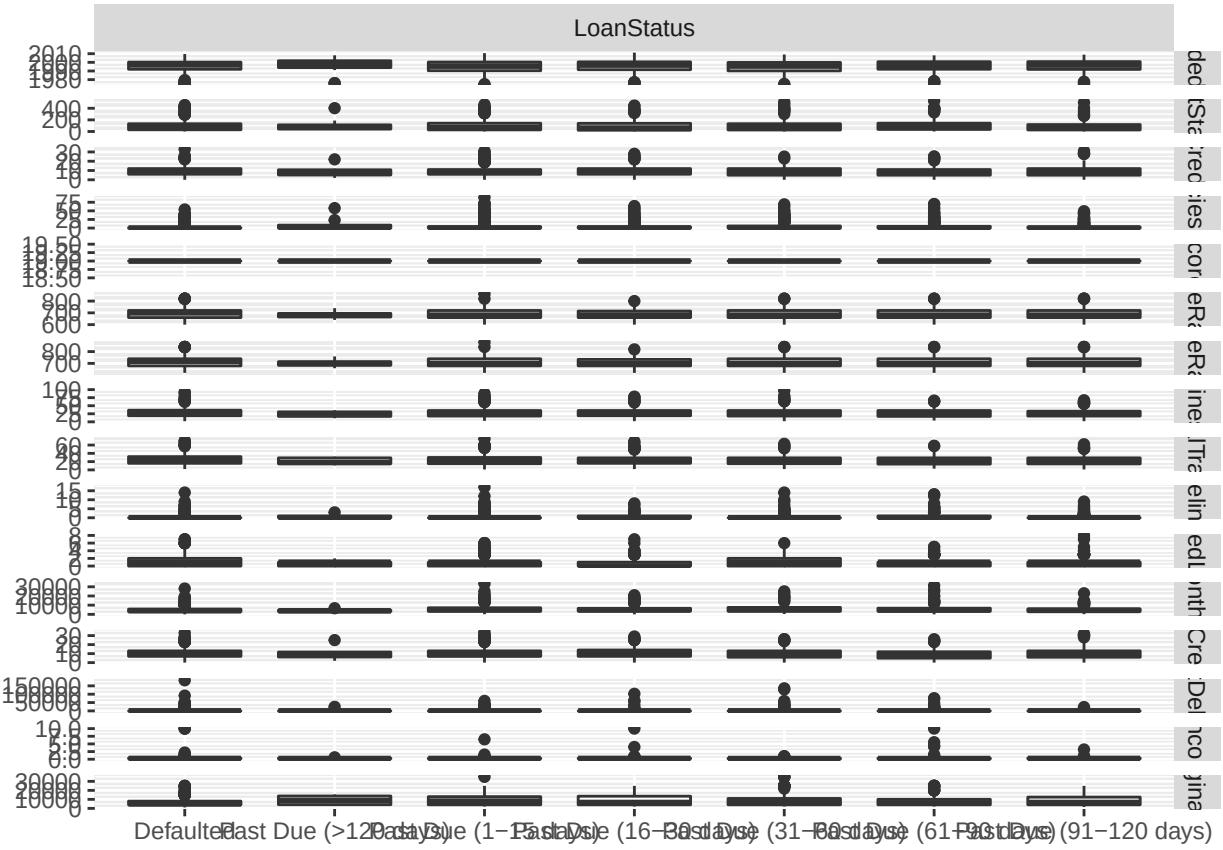
In this case, ggduo can narrow down some of ggpairs' work, this is because it can graph one varible of our dataset against all others simultanously. Here, we are comparing "LoanStatus" against the rest of the variables in separate chunks for better visualization.

```
drops <- c('Term',
           'ListingKey',
           'Occupation',
           'BorrowerState',
           'IsBorrowerHomeowner',
           'EmploymentStatus',
           'IncomeRange',
           'IncomeVerifiable')

csv_file.drops <- csv_file.drops[,!(names(csv_file.drops) %in% drops)]

ncol(csv_file.drops)
```

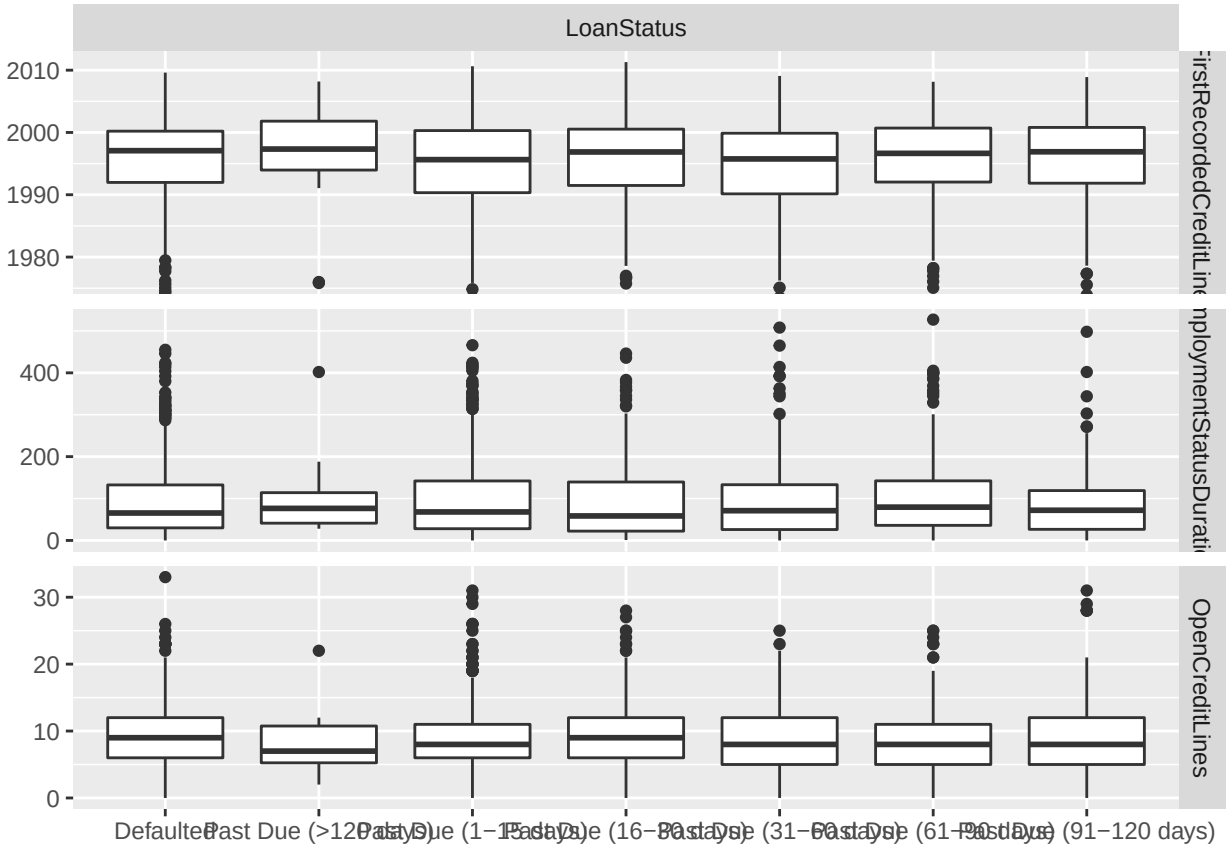
```
## [1] 17
```

```
ggduo(csv_file.drops, 1, 2:17)
```

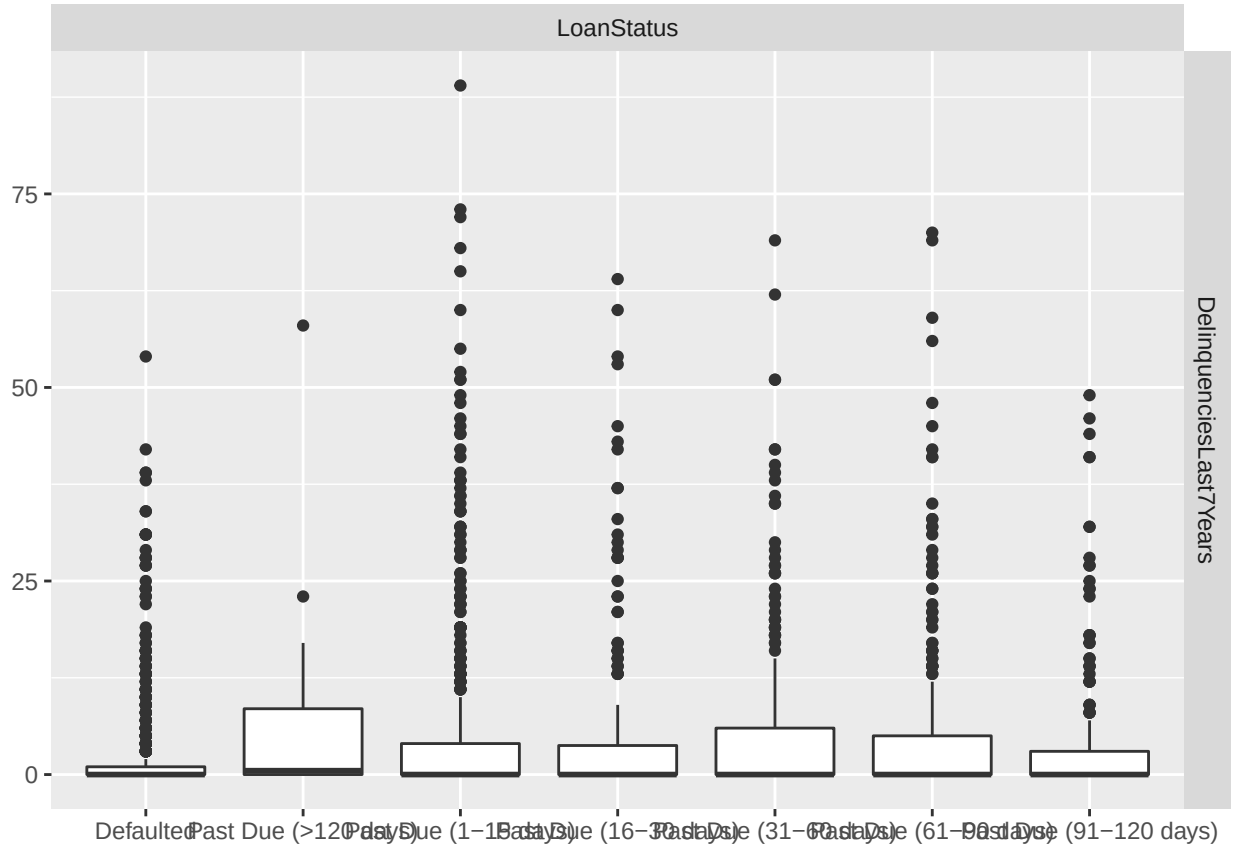
```
## Warning: Removed 85 rows containing non-finite values (stat_boxplot).
```



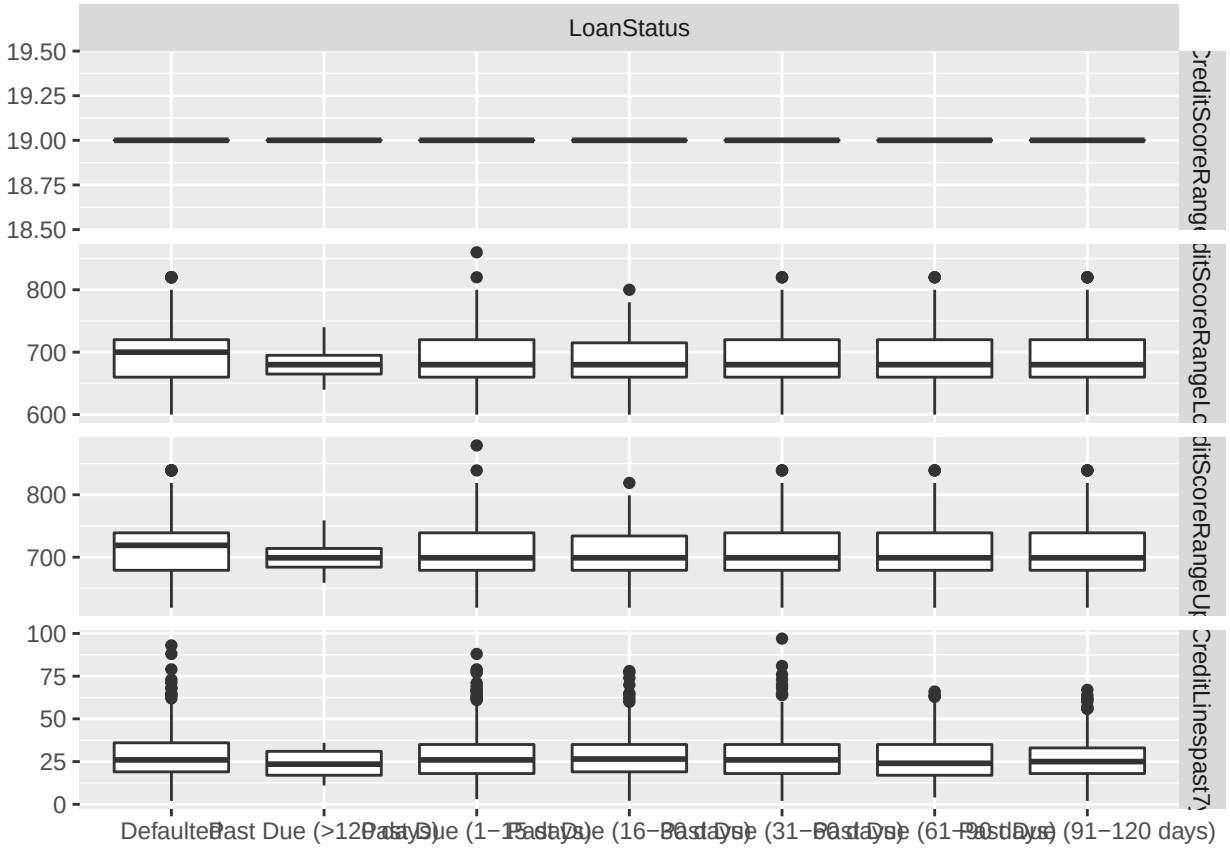

```
# Let's divide this group of graphs into multiple for a better look at the
# graphed data

ggduo(csv_file.drops, 1, 2:4)
```



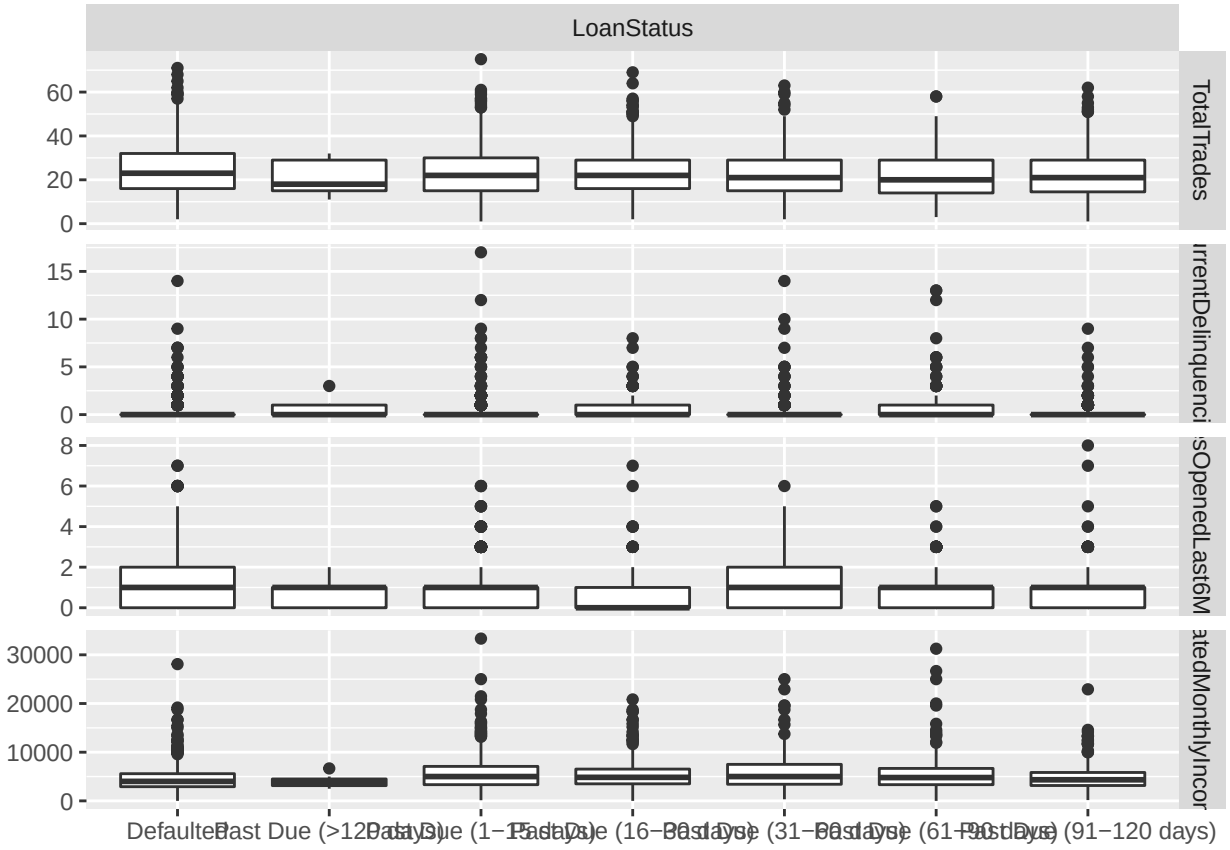

```
ggduo(csv_file.drops, 1, 5)
```




```
ggduo(csv_file.drops, 1, 6:9)
```

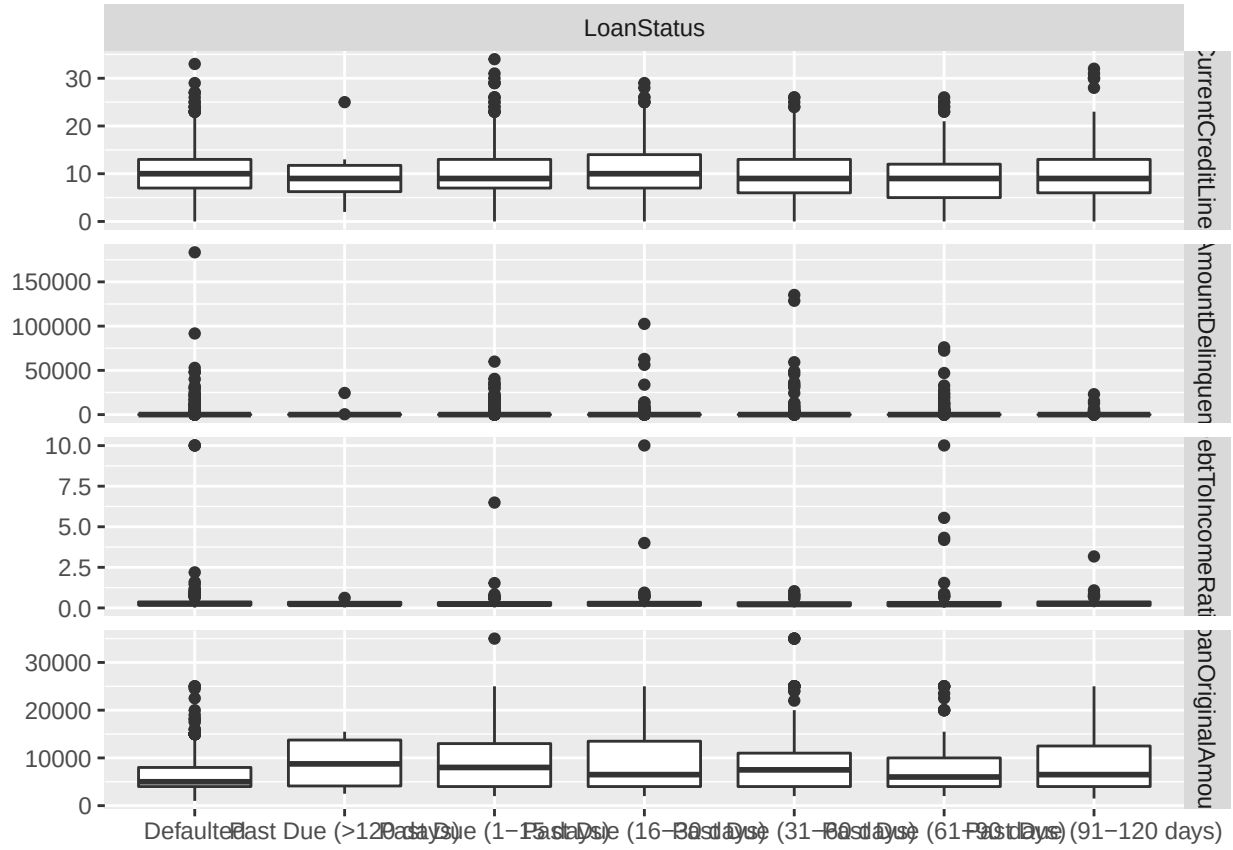



```
ggduo(csv_file.drops, 1, 10:13)
```




```
ggduo(csv_file.drops, 1, 14:17)
```

```
## Warning: Removed 85 rows containing non-finite values (stat_boxplot).
```



Furthermore, we can better describe the graphed ranges of each

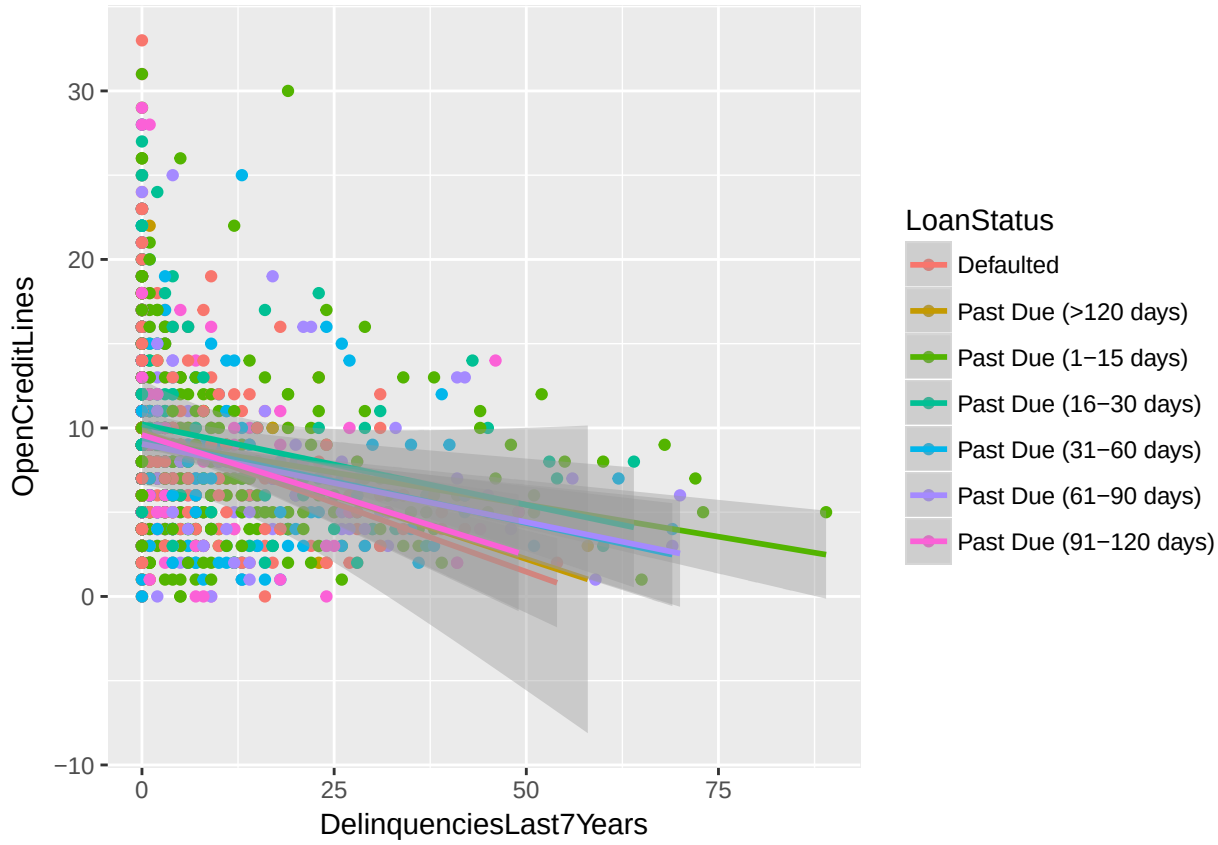
```
csv_file.drops <- csv_file.reassesedLSandEmp[,c("LoanStatus",
                  "IncomeRange",
                  "IsBorrowerHomeowner",
                  "IncomeVerifiable",
                  "EmploymentStatus",
                  "FirstRecordedCreditLine",
                  "EmploymentStatusDuration",
                  "OpenCreditLines",
                  "DelinquenciesLast7Years",
                  "CreditScoreRange",
                  "CreditScoreRangeLower",
                  "CreditScoreRangeUpper",
                  "TotalCreditLinespast7years",
                  "TotalTrades",
                  "CurrentDelinquencies",
                  "TradesOpenedLast6Months",
                  "StatedMonthlyIncome",
                  "CurrentCreditLines",
                  "AmountDelinquent",
                  "DebtToIncomeRatio",
                  "LoanOriginalAmount")]
```

We finally, redefine csv_file.drops to use next some of the columns that we previously got rid off.

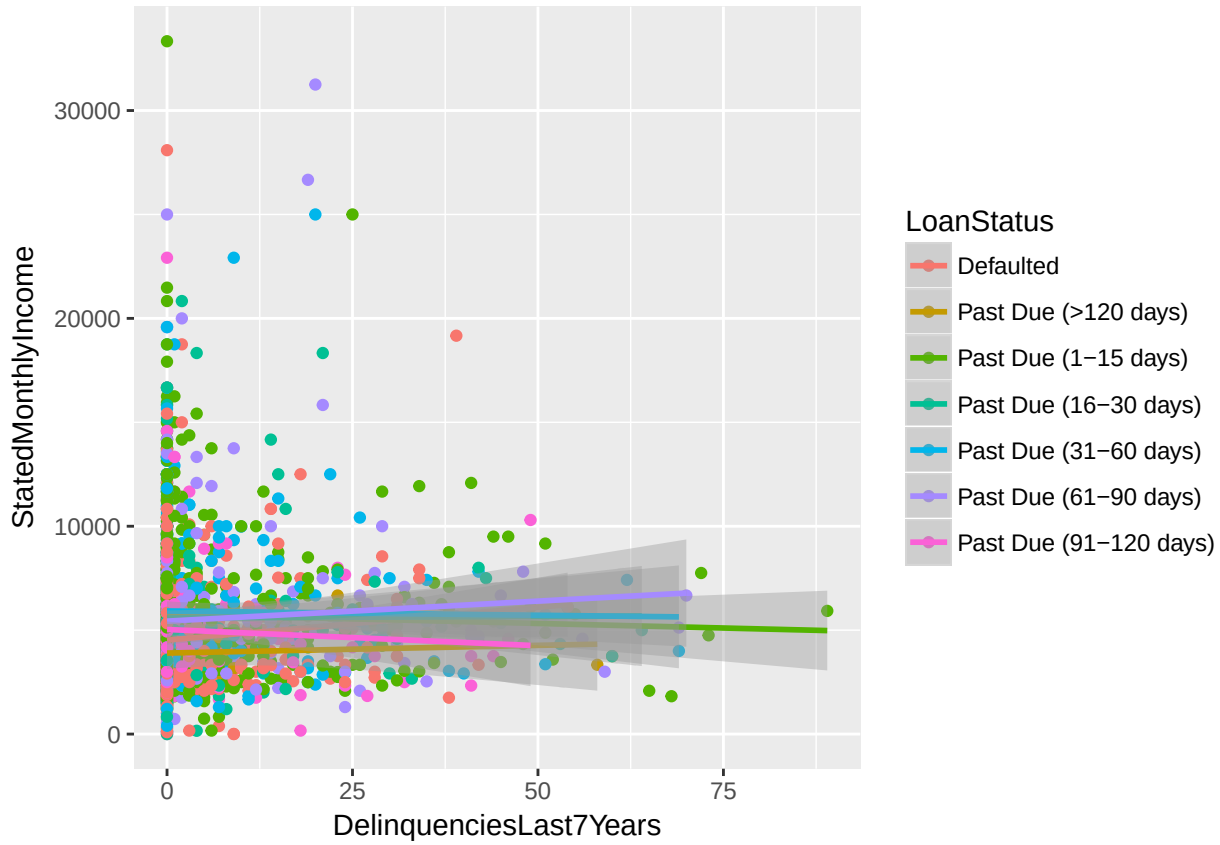
Here we will graph 3 varibles at once, controlling for employment status and loan status. We will also graph

linear models.

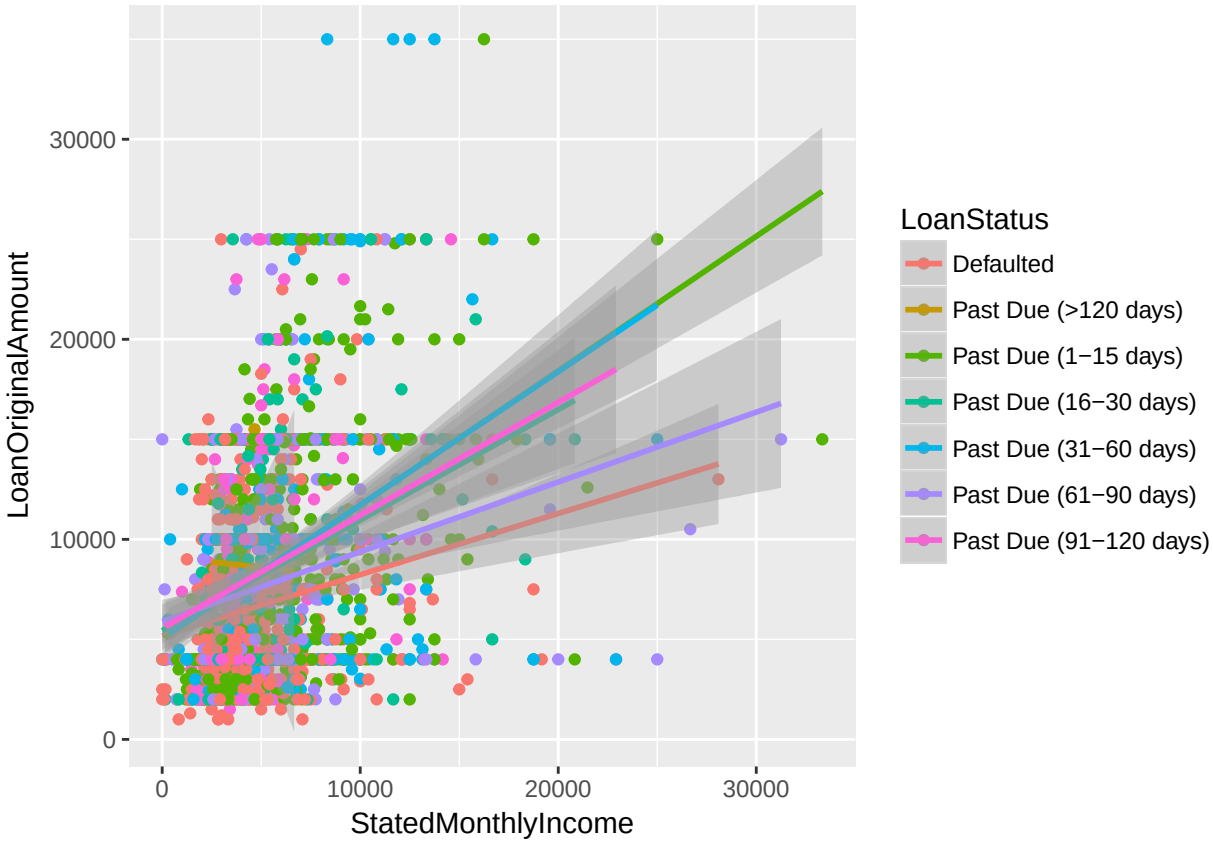
```
ggplot(aes(x = DelinquenciesLast7Years,
           y = OpenCreditLines,
           color = LoanStatus ),
       data = csv_file.drops)+
  geom_point()+
  stat_smooth(method = 'lm') #lm = linear model
```




```
ggplot(aes(x = DelinquenciesLast7Years,
           y = StatedMonthlyIncome,
           color =LoanStatus  ),
       data = csv_file.drops)+
  geom_point()+
  stat_smooth(method = 'lm') #lm = linear model
```




```
ggplot(aes(x = StatedMonthlyIncome,
           y = LoanOriginalAmount,
           color = LoanStatus),
       data = csv_file.drops)+
  geom_point()+
  stat_smooth(method = 'lm') #lm = linear model
```



In this graphs we notice a few characteristics like the negative trend between open credit lines and delinquencies in the last 7 years among borrowers, with all unfavorable levels of loan status, who are employed. Also, there is a positive trend between stated monthly income and loan original amount, with all unfavorable levels of loan status, for borrowers who are employed.

Multivariate Analysis

Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

Let's take a look at some interaction with a correlation value of higher than 0.3 (Significant):

"AmountDelinquent" and "TotalCreditLinespast7years" | cor 0.41

"StatedMonthlyIncome" and "LoanOriginalAmount" | cor 0.335

"LoanOriginalAmount" vs "CreditScoreRangeUpper(Lower)" | cor 0.323

The correlation between amount delinquent and total credit lines opened in the past 7 years may be so relatively highly correlated since it makes sense that the greater the net-credit (sum of all the given credit in every line) a borrower possess, the more room there is for a borrower to owe, even past a delinquent status.

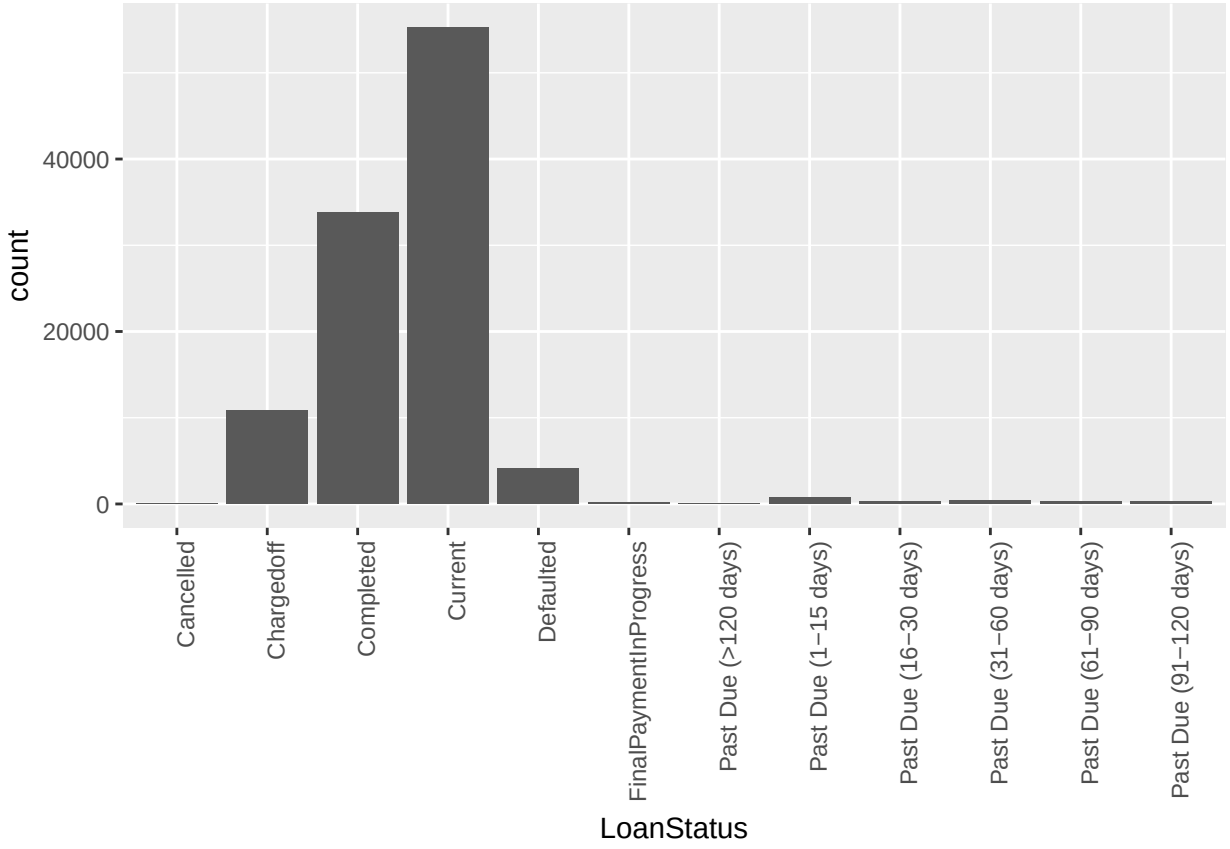
Were there any interesting or surprising interactions between features?

Although the correlation between "LoanOriginalAmount" vs "CreditScoreRangeUpper(Lower)" and "Stated-MonthlyIncome" vs "LoanOriginalAmount" are considered significant, They are not as highly correlated as I expected them to be. At first, under my logic, I suspected that these respectively compared variables would have a correlation of close to 1 since it made sense that the more money you make, the more you can ask for in a loan; By contrast, the lower a borrower's lowest credit score range, the least likely they are to get more money in their loan. But then I realized that I was not taking the other variables into account at the moment, and that I had over estimated the impact of these two variables on "LoanOriginalAmount".

Final Plots and Summary

> **Tip**: You've done a lot of exploration and have built up an understanding of the structure of and relationships between the variables in your dataset. Here, you will select three plots from all of your previous exploration to present here as a summary of some of your most interesting findings. Make sure that you have refined your selected plots for good titling, axis labels (with units), and good aesthetic choices (e.g. color, transparency). After each plot, make sure you justify why you chose each plot by describing what it shows.

Plot One

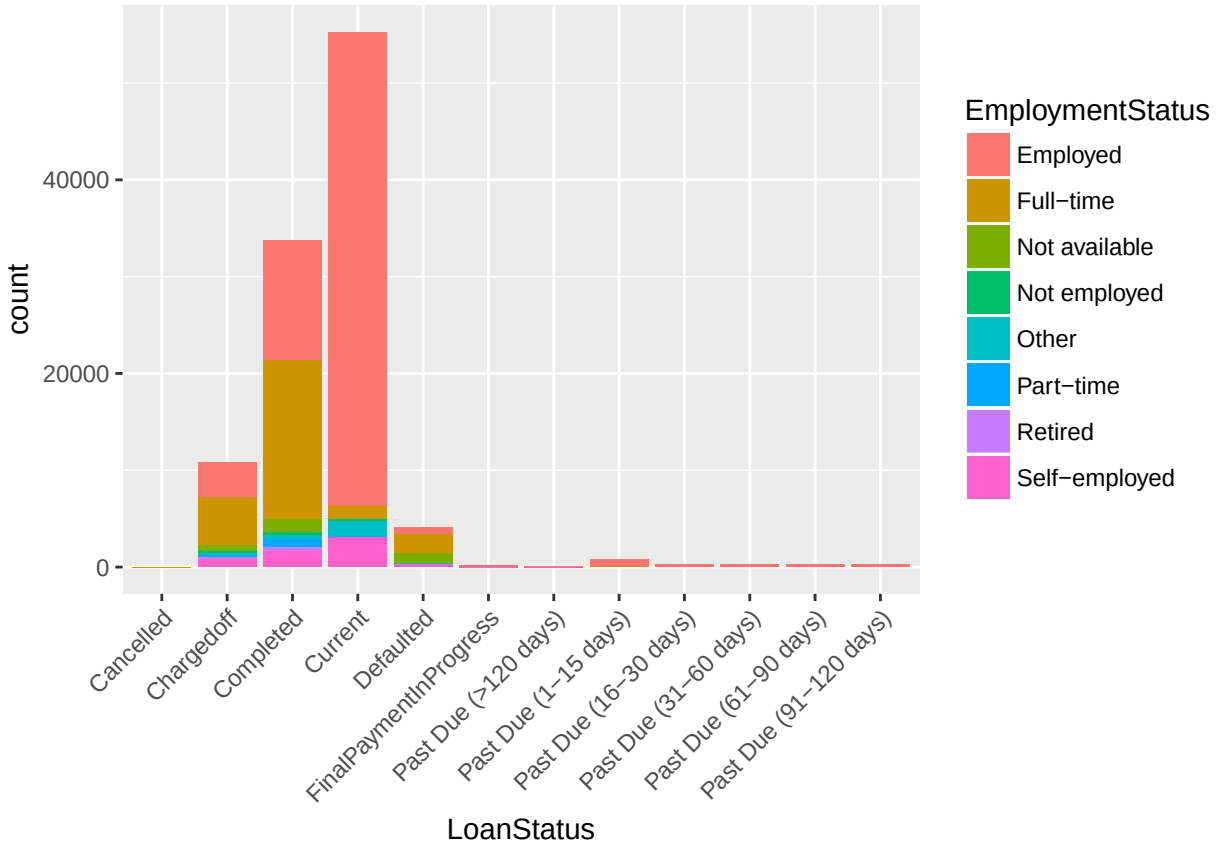


Description One

I believe that this is the best plot from our univariable plot section. Since our main variable of interest is the our "LoanStaus" variable, we can use this plot to clealy depict the magnitude of each of it's levels. We can see that although our unfavorable loan status are relatively uncommon, such loans still exist. From here we can better distinguish the rest of our dataset to learn from these individuals with unfavorable loan statuses.

Plot Two

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

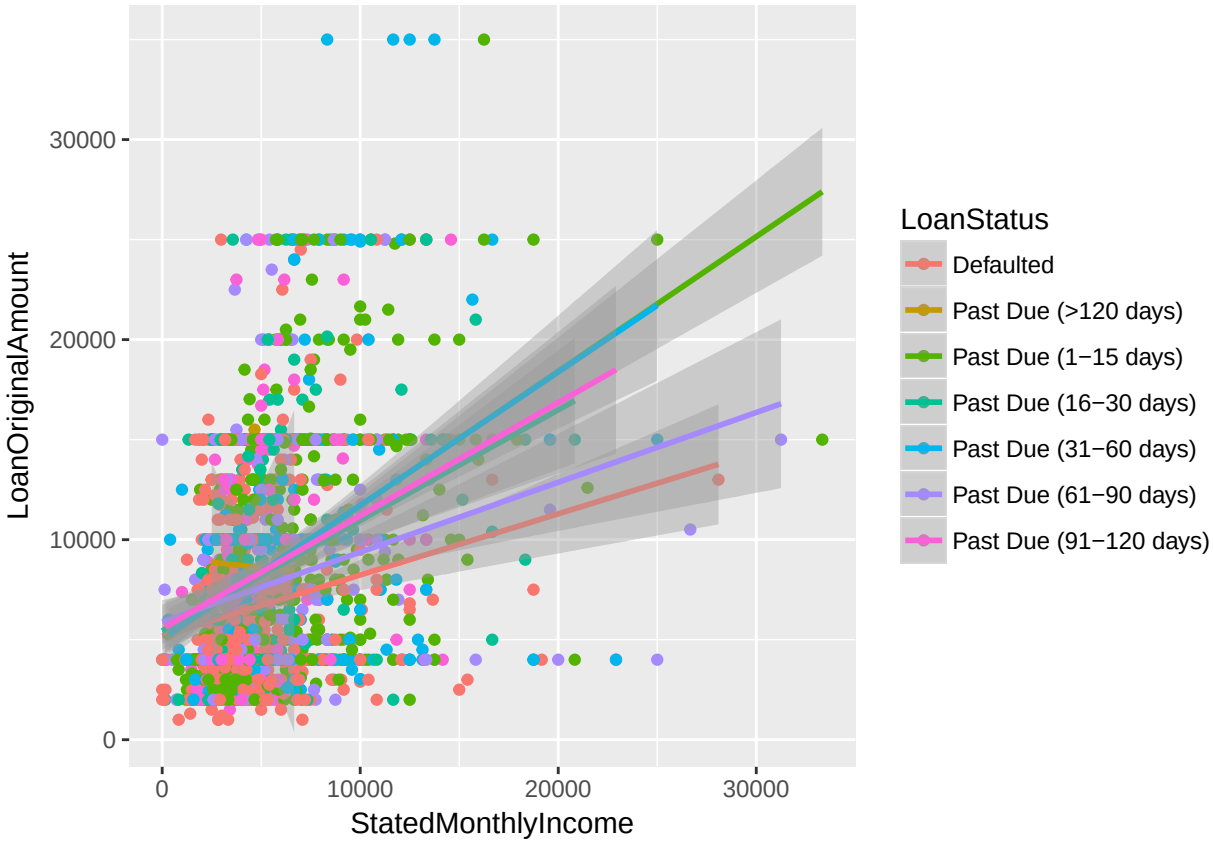


Description Two

Here, just like in the last plot we are showcasing the magnitudes of each possible loan status, but we are using the fill aes argument to visually separate the different employment statuses that any borrower could have noted at the time of listing. This addition serves to demonstrate the composition of each of this loan statuses in terms of employment statuses. we can also use the coord_cartesian wrap to zoom in onto a single loan status level at a time (or more), and the scale_y_continous wrap to zoom in vertically_wise.

I was able to singled out each column in this graph, which helped in getting a sense of percetage composition of each one of these columns by employment status. Later on I was able to futher filter my data to investigate that employment status which seem to make part of those Loan statuses of interest.

Plot Three



Description Three

Here we visualize all trends of each level of loan status against the amount of loan original amount vs the delinquencies in the last 7 years of each borrower who is employed. I believe it is important to highlight these relations since they demonstrate the increasing tendency in loan original amounts according to monthly income for all unfavorable loan status, but at different rates.

Reflection

> **Tip**: Here's the final step! Reflect on the exploration you performed and the insights you found. What were some of the struggles that you went through? What went well? What was surprising? Make sure you include an insight into future work that could be done with the dataset.

It was an interesting project to work on. It served me in helping me build a strategy with which to approach a data set for explorational purposes. Better than that is the fact that I got the chance to do so with a dataset of the financial sector, which peaks my interest.

Some of the struggles I faced at first were finding a purpose for any information I could extract, and selecting my variables carefully, since this is a large set of data. After that, I struggled with few bugs in Rstudio with the graph image resizing (which ultimately led me to discover the "theme" wrap for the functions qplot an ggplot, and jpeg image saving, although I did not use this one in this project). I also found another way to

apply a series of conditional filters to a whole dataset, which then led me to discover the "which"" argument in the multivariable plot section.

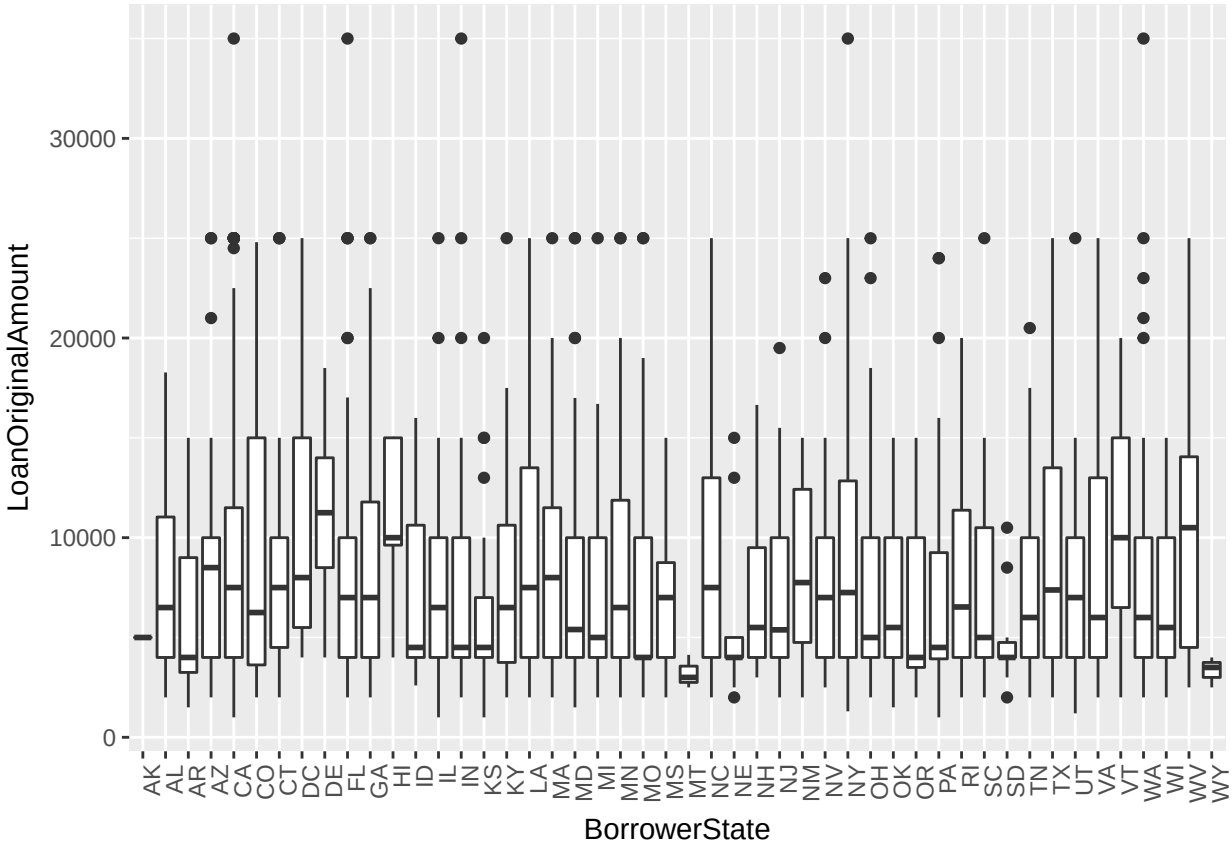
The biggest surprise to me was the plotted the composition of loans per loan status, in terms of employment status in the bivariate plot section. It is clearly visible that those unfavorable status loans were mostly made up of employed borrowers. Although a "past due" does not necessarily imply that a loan will be defaulted on, it does raise question as to why those who are employ tend be the majority of the crowd of those who miss payments, and subsequently may be predisposed to default on their loans. Another surprise was finding how similar most borrower with unfavorable loan statuses are to the rest (I found this with code that can be found in my brainstorm section below) Although not exactly similar to the rest of the borrowers, our borrowers in the csv_file.drops data set, share a lot of characteristics with their counter parts for example: their high and low credit score ranges had a similiar variation, their employment status durations were very similar and so was the spread of original loan amounts. There are even more statistical comparison that can be seen below.

Obviously, there is a lot more relationships that are hidden behind this data, and even more if one is to look at the entire original dataset. Some of the other variables, and impact on loan status, I would have liked to investigate are "BorrowerAPR" and "BorrowerRate"; these two could potentially be some of the biggest factors in predicting whether a loan will be defaulted on or not.

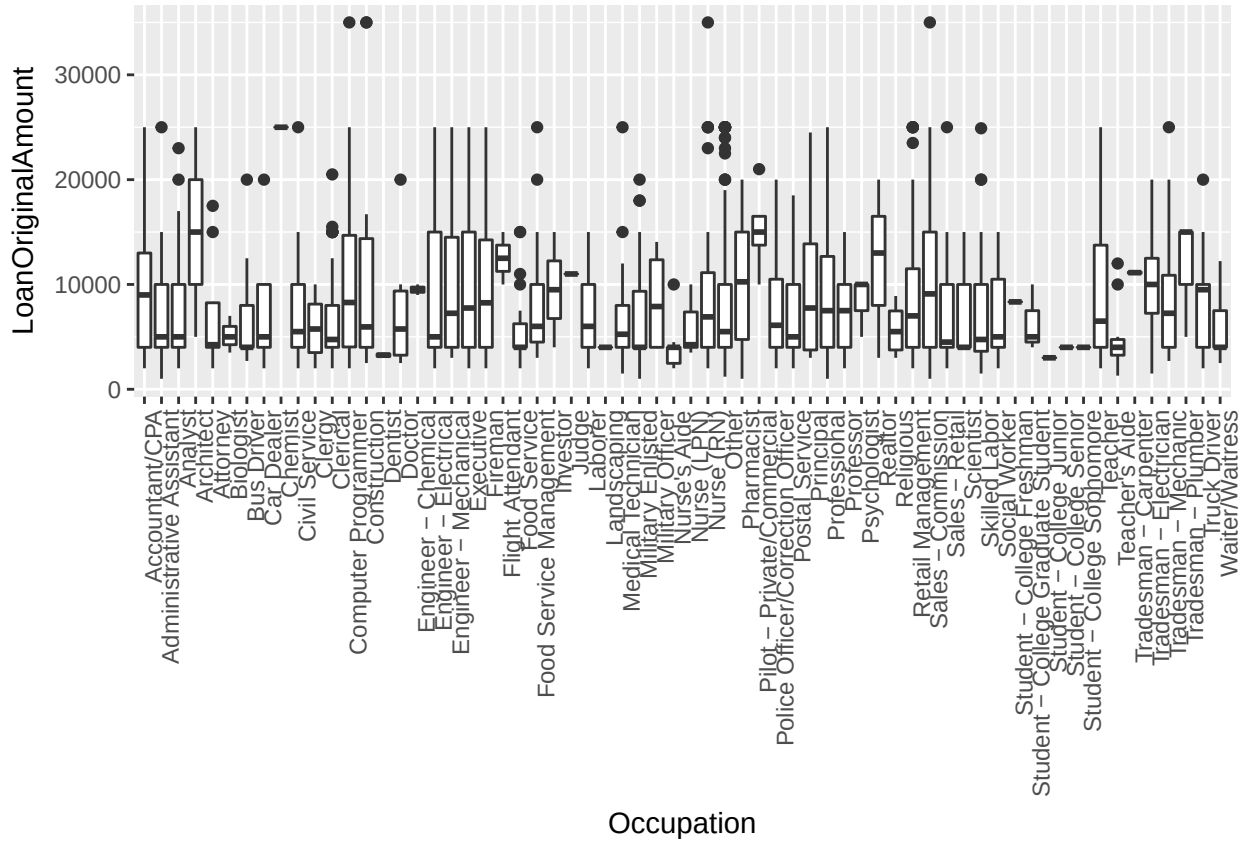
BRAINSTORM

Everything below this point was part of all roughs drafts I previously made. I kept all of this here in case it would come in handy later on, not just the code but the ideas as well.

```
ggplot(aes(x = BorrowerState, y = LoanOriginalAmount), data =
         csv_file.reassesedLSandEmp)+
  geom_boxplot()+
  theme(axis.text.x=element_text(angle=90,vjust=1,hjust=1))
```



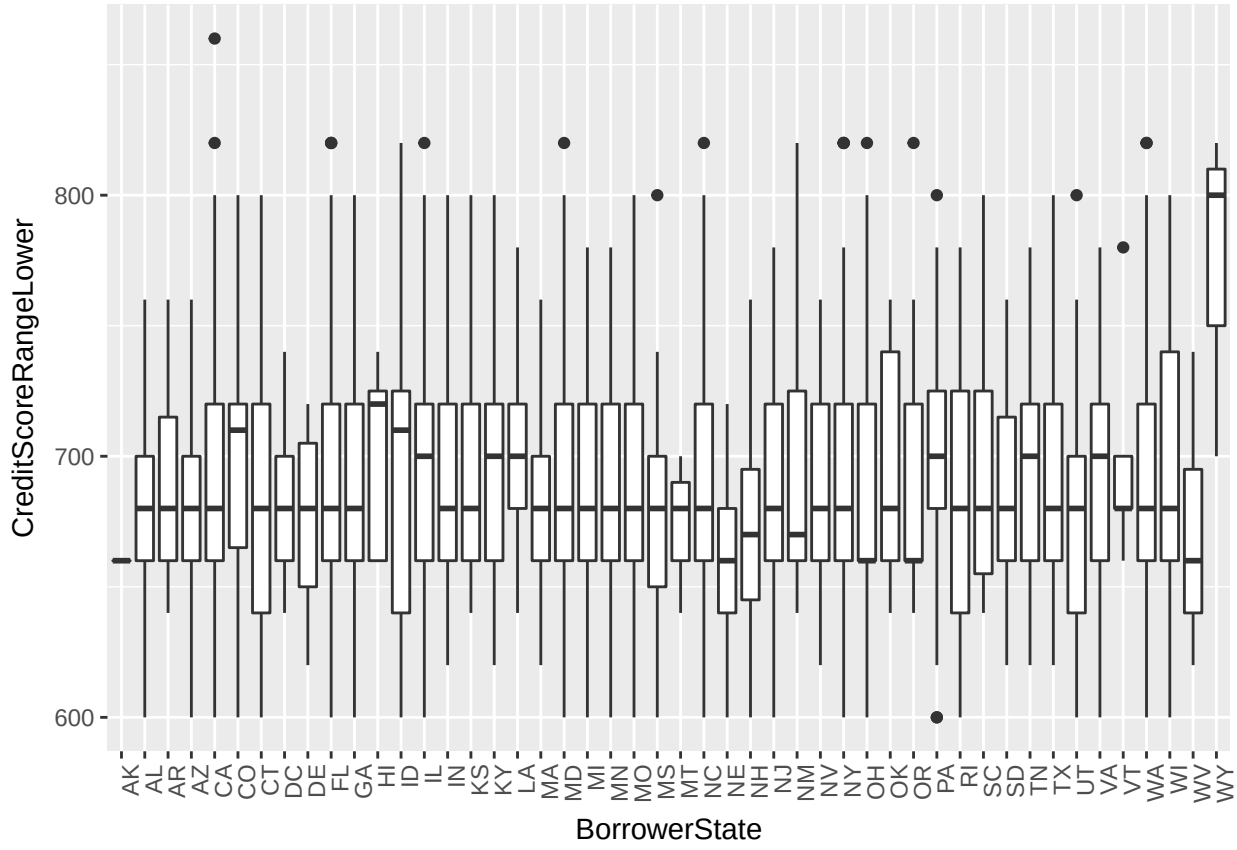

```
ggplot(aes(x = Occupation, y = LoanOriginalAmount), data =
         csv_file.reassesedLSandEmp)+
  geom_boxplot()+
  theme(axis.text.x=element_text(angle=90,vjust=1,hjust=1))
```



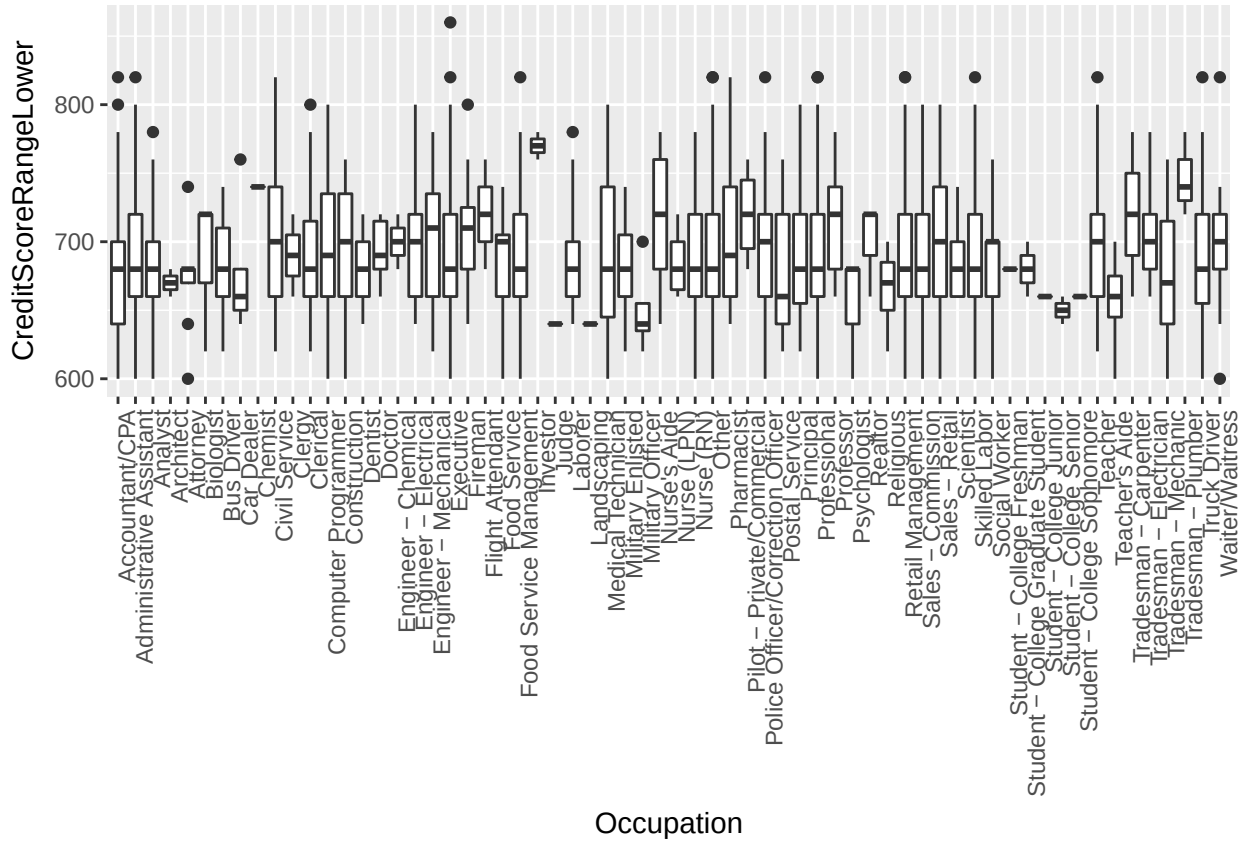
How about switching our continous variable with others?

CreditScoreRangeLower with categorical variable variations

```
## CreditScoreRangeLower vs BorrowerStates
ggplot(aes(x = BorrowerState, y = CreditScoreRangeLower), data =
         csv_file.reassesedLSandEmp)+
  geom_boxplot()+
  theme(axis.text.x=element_text(angle=90,vjust=1,hjust=1))
```




```
## CreditScoreRangeLower vs Occupation
ggplot(aes(x = Occupation, y = CreditScoreRangeLower), data =
         csv_file.reassesedLSandEmp)+
  geom_boxplot()+
  theme(axis.text.x=element_text(angle=90,vjust=1,hjust=1))
```

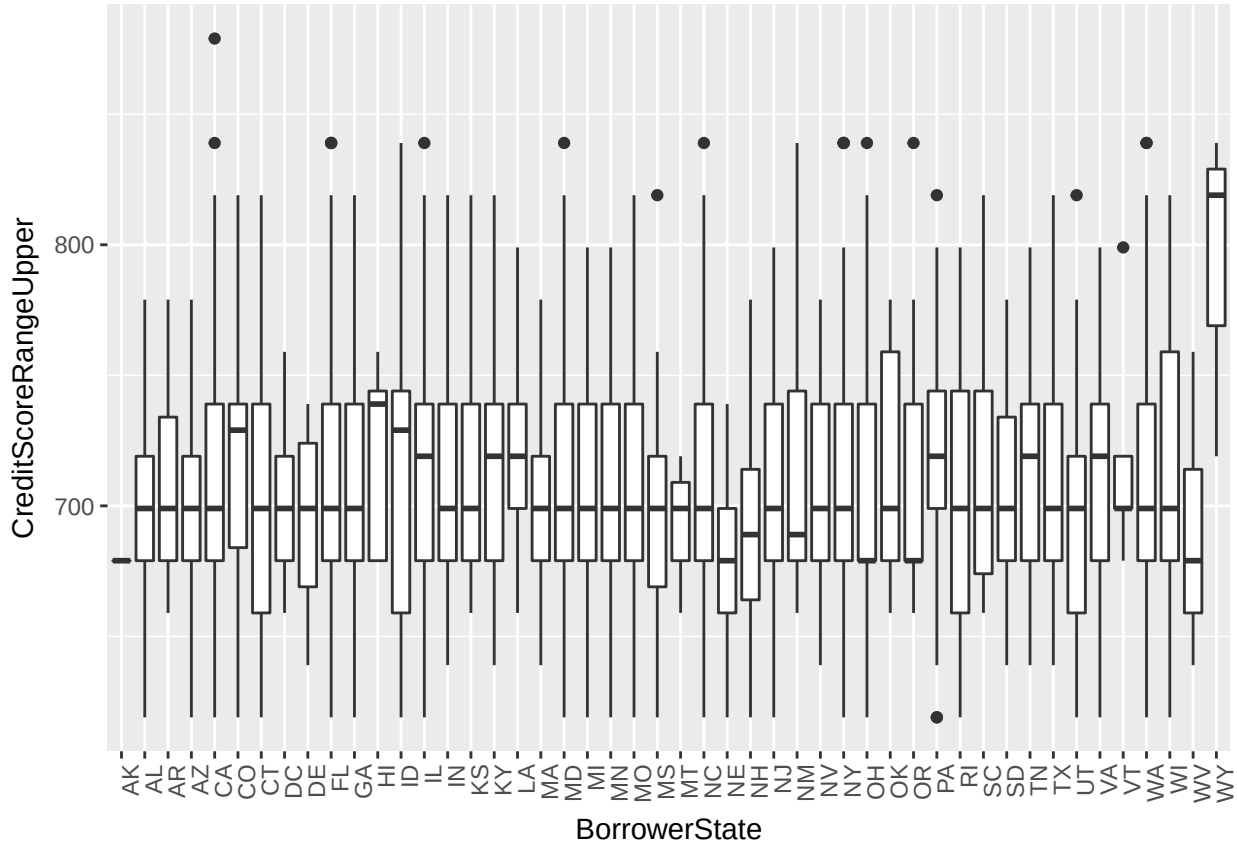


It is easy now to spot the variability of The lowest reported credit scores for all employed customers who have unfavorable loan statuses by their state of residence or by their profession. The 'CreditScoreRangeLower' vs 'BorrowerState' reveals that Idaho, Oklahoma and Wisconsin are some of the states with the largest variability. Alaska possess the lowest variability. Finally, Wyoming presents itself with the same variability of other states but at an overall higher range than others. In other words, Wyomings' borrower's credit scores tend to be higher than most others.

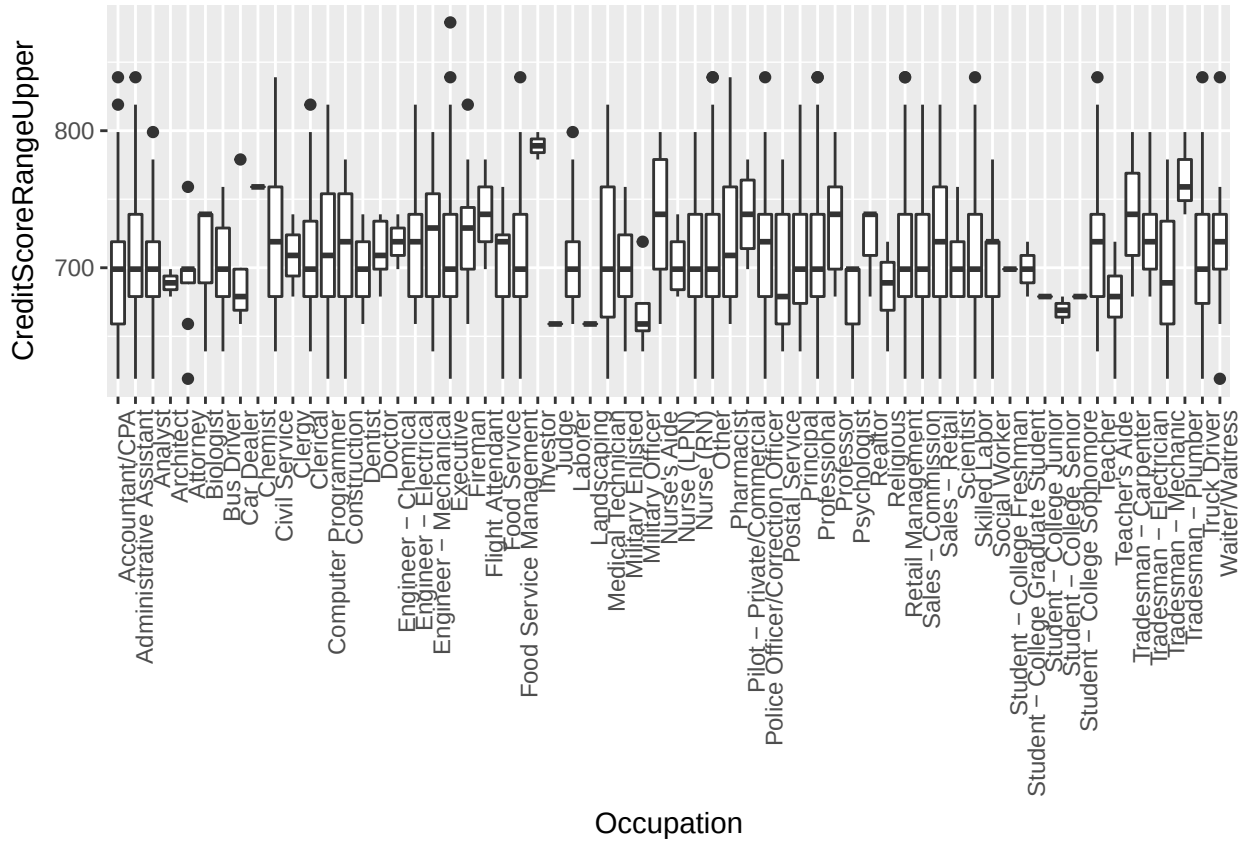
The "Occupation" vs "CreditScoreRangeLower" demonstrates that among those employed borrowers with unfavorable loan statuses, borrowers who are mdeical technicians have some of the highest spreads of Lower range credit scores. Whereas chemists, judges, borrower's who work in the landscaping business and college freshmen/junior/Sophomore are amongst those with the lowest spread.

CreditScoreRangeUpper with categorical variable variations

```
## CreditScoreRangeUpper vs BorrowerStates
ggplot(aes(x = BorrowerState, y = CreditScoreRangeUpper), data =
         csv_file.reassesedLSandEmp)+
  geom_boxplot()+
  theme(axis.text.x=element_text(angle=90,vjust=1,hjust=1))
```




```
## CreditScoreRangeUpper vs Occupation
ggplot(aes(x = Occupation, y = CreditScoreRangeUpper), data =
         csv_file.reassesedLSandEmp)+
  geom_boxplot()+
  theme(axis.text.x=element_text(angle=90,vjust=1,hjust=1))
```



Just like we saw before WY has some of the best credit scores, in this case, the upper range. Like we noticed before ID, OK and WI have some of the greatest spread, but unlike before we notice that CT also possess a higher than normal spread, in the upper range of course.

At this point we have seen some of the few combination of interaction among a few variables of the latest subset of our original dataset.

Comparison

csv_file.counterDrops will be a new subset of our orginal data set in which you can find all the rows that are NOT in the csv_file.drops subset, but it will only contain the same columns as does csv_file.drops, this will help compare our specific subjects of interest againt the larger majority that is not employed AND possess an unfavorable loan status.

```
# make csv_file.counterDrops

# we are going to write csv_file2, the file against which we will compare
# csv_file.drops in order to find those rows the belong in
# csv_file.counterdrops, but we will do so, in the same column order as
# csv_file.drops

csv_file.counterDrops <- csv_file[,c("LoanStatus",
                  "EmploymentStatus",
                  "FirstRecordedCreditLine",
                  "EmploymentStatusDuration",
                  "OpenCreditLines",
```

```
                  "DelinquenciesLast7Years",
                  "CreditScoreRange",
                  "CreditScoreRangeLower",
                  "CreditScoreRangeUpper",
                  "TotalCreditLinespast7years",
                  "TotalTrades",
                  "CurrentDelinquencies",
                  "TradesOpenedLast6Months",
                  "StatedMonthlyIncome",
                  "CurrentCreditLines",
                  "AmountDelinquent",
                  "DebtToIncomeRatio",
                  "LoanOriginalAmount")]

csv_file.drops <- csv_file.drops[,c("LoanStatus",
                  "EmploymentStatus",
                  "FirstRecordedCreditLine",
                  "EmploymentStatusDuration",
                  "OpenCreditLines",
                  "DelinquenciesLast7Years",
                  "CreditScoreRange",
                  "CreditScoreRangeLower",
                  "CreditScoreRangeUpper",
                  "TotalCreditLinespast7years",
                  "TotalTrades",
                  "CurrentDelinquencies",
                  "TradesOpenedLast6Months",
                  "StatedMonthlyIncome",
                  "CurrentCreditLines",
                  "AmountDelinquent",
                  "DebtToIncomeRatio",
                  "LoanOriginalAmount")]

csv_file.counterDrops <-
  csv_file.counterDrops[which(
    csv_file.counterDrops$EmploymentStatus != "Employed"&(
    csv_file.counterDrops$LoanStatus !='Past Due (1-15 days)'|
    csv_file.counterDrops$LoanStatus !='Past Due (16-30 days)'|
    csv_file.counterDrops$LoanStatus !='Past Due (31-60 days)'|
    csv_file.counterDrops$LoanStatus !='Past Due (61-90 days)'|
    csv_file.counterDrops$LoanStatus !='Past Due (91-120 days)'|
    csv_file.counterDrops$LoanStatus !='Past Due (>120 days)'|
    csv_file.counterDrops$LoanStatus !='Defaulted')),]

unique(csv_file.counterDrops$EmploymentStatus)
```

```
## [1] Self-employed Not available Full-time     Other
## [6] Not employed  Part-time     Retired
## 9 Levels:  Employed Full-time Not available Not employed ... Self-employed
```

```
unique(csv_file.counterDrops$LoanStatus)
```

```
##  [1] Completed              Defaulted              Current
##  [4] Chargedoff             Past Due (1-15 days)   Cancelled
```

```
##  [7] Past Due (16-30 days)  Past Due (31-60 days)  FinalPaymentInProgress
## [10] Past Due (61-90 days)  Past Due (91-120 days) Past Due (>120 days)
## 12 Levels: Cancelled Chargedoff Completed Current ... Past Due (91-120 days)
```

```
names(csv_file.counterDrops)
```

```
##  [1] "LoanStatus"                 "EmploymentStatus"
##  [3] "FirstRecordedCreditLine"    "EmploymentStatusDuration"
##  [5] "OpenCreditLines"            "DelinquenciesLast7Years"
##  [7] "CreditScoreRange"           "CreditScoreRangeLower"
##  [9] "CreditScoreRangeUpper"      "TotalCreditLinespast7years"
## [11] "TotalTrades"                "CurrentDelinquencies"
## [13] "TradesOpenedLast6Months"    "StatedMonthlyIncome"
## [15] "CurrentCreditLines"         "AmountDelinquent"
## [17] "DebtToIncomeRatio"          "LoanOriginalAmount"
```

```
summary(csv_file.counterDrops)
```

```
##                  LoanStatus         EmploymentStatus
##  Completed            :25742   Full-time    :26355
##  Chargedoff           : 8463   Self-employed: 6134
##  Current              : 7647   Not available: 5347
##  Defaulted            : 4388   Other        : 3806
##  Past Due (1-15 days) :  131                : 2255
##  Past Due (31-60 days):   70   Part-time    : 1088
##  (Other)              :  174   (Other)      : 1630
##  FirstRecordedCreditLine EmploymentStatusDuration OpenCreditLines
##  Min.   :1947-08-24      Min.   :  0.00           Min.   : 0.000
##  1st Qu.:1989-11-01      1st Qu.: 19.00           1st Qu.: 5.000
##  Median :1995-03-27      Median : 51.00           Median : 8.000
##  Mean   :1994-02-21      Mean   : 83.29           Mean   : 8.555
##  3rd Qu.:1999-08-19      3rd Qu.:114.00           3rd Qu.:11.000
##  Max.   :2012-02-14      Max.   :755.00           Max.   :51.000
##  NA's   :697             NA's   :7624             NA's   :7604
##  DelinquenciesLast7Years CreditScoreRange CreditScoreRangeLower
##  Min.   : 0.000          Min.   :19       Min.   :  0.0
##  1st Qu.: 0.000          1st Qu.:19       1st Qu.:620.0
##  Median : 0.000          Median :19       Median :660.0
##  Mean   : 4.784          Mean   :19       Mean   :665.4
##  3rd Qu.: 4.000          3rd Qu.:19       3rd Qu.:720.0
##  Max.   :99.000          Max.   :19       Max.   :880.0
##  NA's   :990             NA's   :591      NA's   :591
##  CreditScoreRangeUpper TotalCreditLinespast7years  TotalTrades
##  Min.   : 19.0         Min.   :  2.0              Min.   :  0.00
##  1st Qu.:639.0         1st Qu.: 15.0              1st Qu.: 13.00
##  Median :679.0         Median : 23.0              Median : 20.00
##  Mean   :684.4         Mean   : 25.1              Mean   : 21.71
##  3rd Qu.:739.0         3rd Qu.: 33.0              3rd Qu.: 29.00
##  Max.   :899.0         Max.   :136.0              Max.   :126.00
##  NA's   :591           NA's   :697                NA's   :7544
##  CurrentDelinquencies TradesOpenedLast6Months StatedMonthlyIncome
##  Min.   : 0.0000      Min.   : 0.000          Min.   :      0
##  1st Qu.: 0.0000      1st Qu.: 0.000          1st Qu.:   2550
##  Median : 0.0000      Median : 0.000          Median :   4000
##  Mean   : 0.9981      Mean   : 0.876          Mean   :   4842
```

```
##  3rd Qu.: 1.0000      3rd Qu.: 1.000          3rd Qu.:   5833
##  Max.   :83.0000      Max.   :17.000          Max.   :1750003
##  NA's   :697          NA's   :7544
##  CurrentCreditLines AmountDelinquent DebtToIncomeRatio LoanOriginalAmount
##  Min.   : 0.000     Min.   :     0   Min.   : 0.000    Min.   : 1000
##  1st Qu.: 6.000     1st Qu.:     0   1st Qu.: 0.130    1st Qu.: 2550
##  Median : 9.000     Median :     0   Median : 0.210    Median : 4500
##  Mean   : 9.782     Mean   :  1026   Mean   : 0.313    Mean   : 6233
##  3rd Qu.:13.000     3rd Qu.:     0   3rd Qu.: 0.320    3rd Qu.: 8000
##  Max.   :52.000     Max.   :444745   Max.   :10.010    Max.   :35000
##  NA's   :7604       NA's   :7622     NA's   :7128
```

```
summary(csv_file.drops)
```

```
##                   LoanStatus       EmploymentStatus
##  Past Due (1-15 days)  :675   Employed     :2355
##  Defaulted             :630                :   0
##  Past Due (31-60 days) :293   Full-time    :   0
##  Past Due (61-90 days) :262   Not available:   0
##  Past Due (91-120 days):255   Not employed :   0
##  Past Due (16-30 days) :226   Other        :   0
##  (Other)               : 14   (Other)      :   0
##  FirstRecordedCreditLine EmploymentStatusDuration OpenCreditLines
##  Min.   :1958-01-01      Min.   :  0.00           Min.   : 0.000
##  1st Qu.:1991-04-27      1st Qu.: 28.00           1st Qu.: 6.000
##  Median :1996-08-06      Median : 69.00           Median : 8.000
##  Mean   :1995-05-29      Mean   : 94.99           Mean   : 9.008
##  3rd Qu.:2000-05-24      3rd Qu.:135.50           3rd Qu.:12.000
##  Max.   :2011-04-19      Max.   :527.00           Max.   :33.000
##
##  DelinquenciesLast7Years CreditScoreRange CreditScoreRangeLower
##  Min.   : 0.000          Min.   :19       Min.   :600.0
##  1st Qu.: 0.000          1st Qu.:19       1st Qu.:660.0
##  Median : 0.000          Median :19       Median :680.0
##  Mean   : 4.409          Mean   :19       Mean   :689.8
##  3rd Qu.: 3.000          3rd Qu.:19       3rd Qu.:720.0
##  Max.   :89.000          Max.   :19       Max.   :860.0
##
##  CreditScoreRangeUpper TotalCreditLinespast7years  TotalTrades
##  Min.   :619.0         Min.   : 2.00              Min.   : 1.00
##  1st Qu.:679.0         1st Qu.:18.00              1st Qu.:15.00
##  Median :699.0         Median :26.00              Median :22.00
##  Mean   :708.8         Mean   :27.72              Mean   :23.33
##  3rd Qu.:739.0         3rd Qu.:35.00              3rd Qu.:30.00
##  Max.   :879.0         Max.   :97.00              Max.   :75.00
##
##  CurrentDelinquencies TradesOpenedLast6Months StatedMonthlyIncome
##  Min.   : 0.0000      Min.   :0.0000          Min.   :    0.25
##  1st Qu.: 0.0000      1st Qu.:0.0000          1st Qu.: 3166.67
##  Median : 0.0000      Median :1.0000          Median : 4583.33
##  Mean   : 0.4841      Mean   :0.9359          Mean   : 5320.30
##  3rd Qu.: 0.0000      3rd Qu.:1.0000          3rd Qu.: 6585.00
##  Max.   :17.0000      Max.   :8.0000          Max.   :33333.33
##
##  CurrentCreditLines AmountDelinquent DebtToIncomeRatio LoanOriginalAmount
```

```
##  Min.   : 0.0       Min.   :     0   Min.   : 0.0100   Min.   : 1000
##  1st Qu.: 6.0       1st Qu.:     0   1st Qu.: 0.1600   1st Qu.: 4000
##  Median : 9.0       Median :     0   Median : 0.2300   Median : 6000
##  Mean   :10.1       Mean   :  1417   Mean   : 0.3009   Mean   : 8051
##  3rd Qu.:13.0       3rd Qu.:     0   3rd Qu.: 0.3500   3rd Qu.:10000
##  Max.   :34.0       Max.   :183396   Max.   :10.0100   Max.   :35000
##                                      NA's   :85
```

The point of this last section was to compare our subjects of interest to their counterparts in a more detailed aspect. More information can be derived from these summaries alone.