LoanData

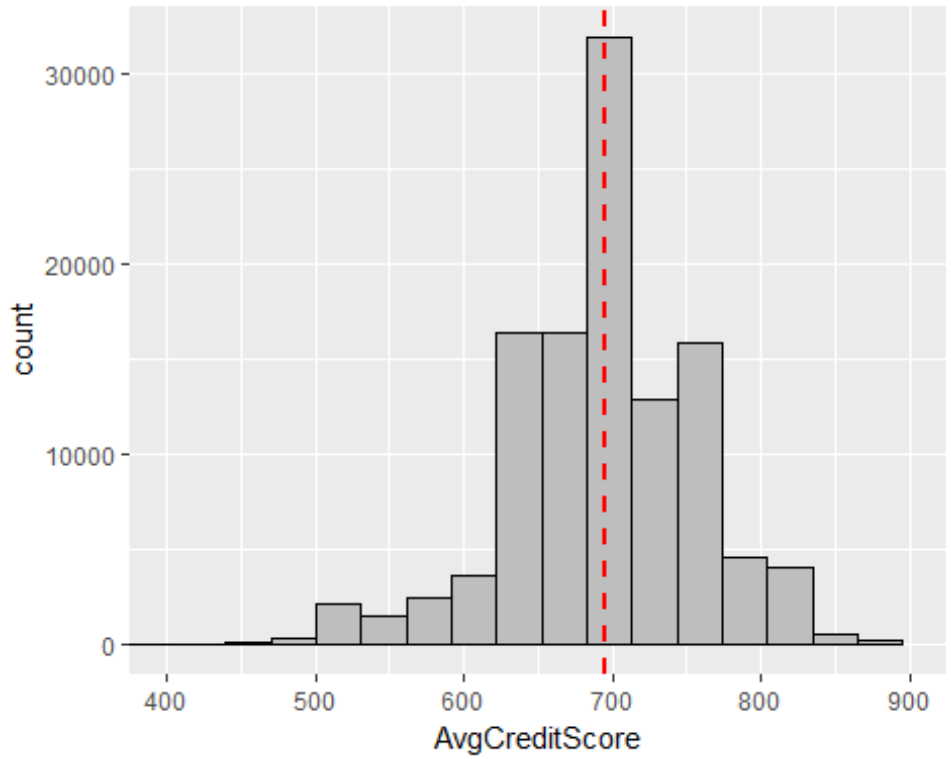
```
ld <- read.csv('prosperLoanData.csv')

suppressMessages(library(ggplot2))
suppressMessages(library(dplyr))
suppressMessages(library(reshape))
library(wesanderson)

ld <- subset(ld, LoanStatus != "Cancelled")

#Averaging lower and upper credit score rating to get the avg credit score
rating
ld$AvgCreditScore <- (ld$CreditScoreRangeUpper+ld$CreditScoreRangeLower) / 2

#Plotting the distribution of avg credit scores with a line in the middle for
mean
#Most traditional companies would want to know the credit score of the
portfolio as a measuring yardstick
ggplot(ld, aes(AvgCreditScore)) +geom_histogram(color='black', fill='gray') +
  coord_cartesian(xlim = c(400,900)) +
  geom_vline(aes(xintercept = mean(AvgCreditScore, na.rm = T)), #Ignore's NA
values for mean
             color = 'red', linetype='dashed', size=1)
```



```r
#In the documentation it says:
#Pre 2009 prosper generated grade (CreditGrade)
#Post 2009 (ProsperRating Alpha = AA-HR) and (ProsperScore 1-10 best being
10)

#Adding another column looking at just the date of the loan
ld$LoanDate <- as.Date(ld$LoanOriginationDate, "%Y-%m-%d")

#Making a dataframe of the pre 2009 data because it was judged by different
scale (CreditGrade)
ldpre2009 <- subset(ld, LoanDate < "2009-01-01")

#Making dataframe of the post 2009 data because it was judged by comperable
but diff scale (ProsperRating Alpha)
ldpost2009 <- subset(ld, LoanDate >= "2009-01-01")

ld <- ld %>%
mutate(ProsperGradeAll = ifelse(ProsperRating..Alpha. != '',
as.character(ProsperRating..Alpha.), as.character(CreditGrade)))

ld <- ld %>%
mutate(ProsperGradeAll = ifelse(ProsperGradeAll != '',
as.character(ProsperGradeAll), 'NG'))

ld <- ld %>%
mutate(ProsperGradeAll = ifelse(ProsperGradeAll != 'NC',
as.character(ProsperGradeAll), 'NG'))

ld <- subset(ld, ProsperGradeAll != 'NG')
```

```r
ld$ProsperGradeAll <- factor(ld$ProsperGradeAll, levels = c('AA','A','B',
'C', 'D', 'E', 'HR'))
summary(ld$ProsperGradeAll)
```

```
##    AA     A     B     C     D     E    HR
##  8881 17865 19970 23993 19427 13084 10440
```

```r
#Looking at the distributions of the pre 2009 credit grades
prop.table(table(ldpre2009$CreditGrade))
```

```
##
##                     A          AA           B           C           D
## 0.000000000 0.114463453 0.121237256 0.151581130 0.195023328 0.178019699
##           E          HR          NC
## 0.113668567 0.121133575 0.004872991
```

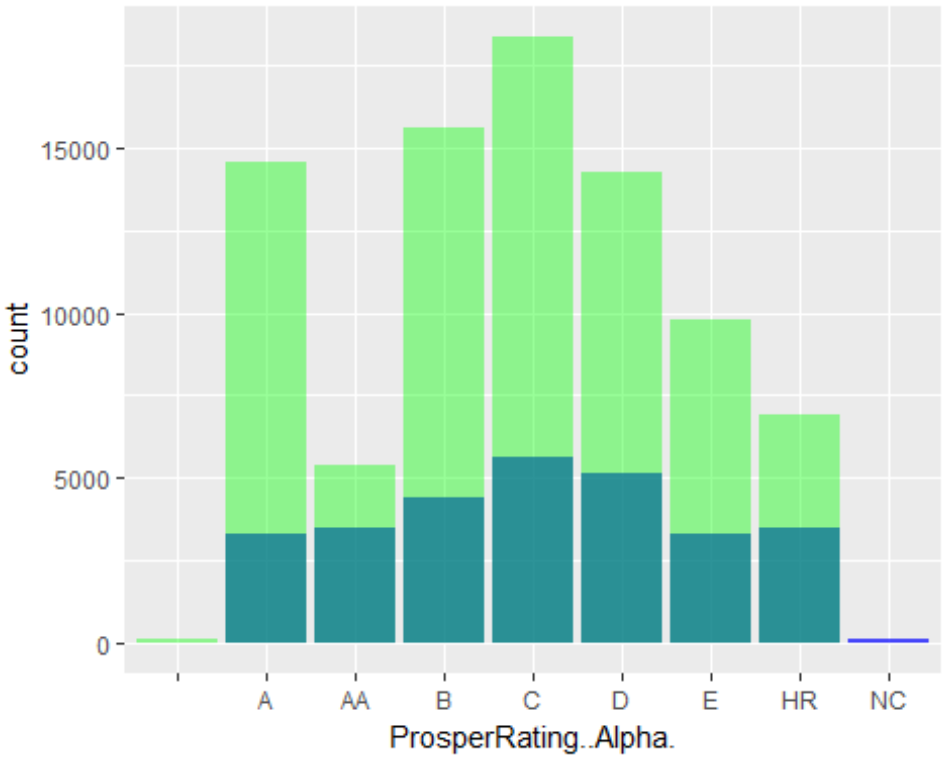
```r
#We can see that it looked like a fairly normal distribution prior to 2009

#Looking at the distribution of the post 2009 prosper ratings
prop.table(table(ldpost2009$ProsperRating..Alpha.))
```

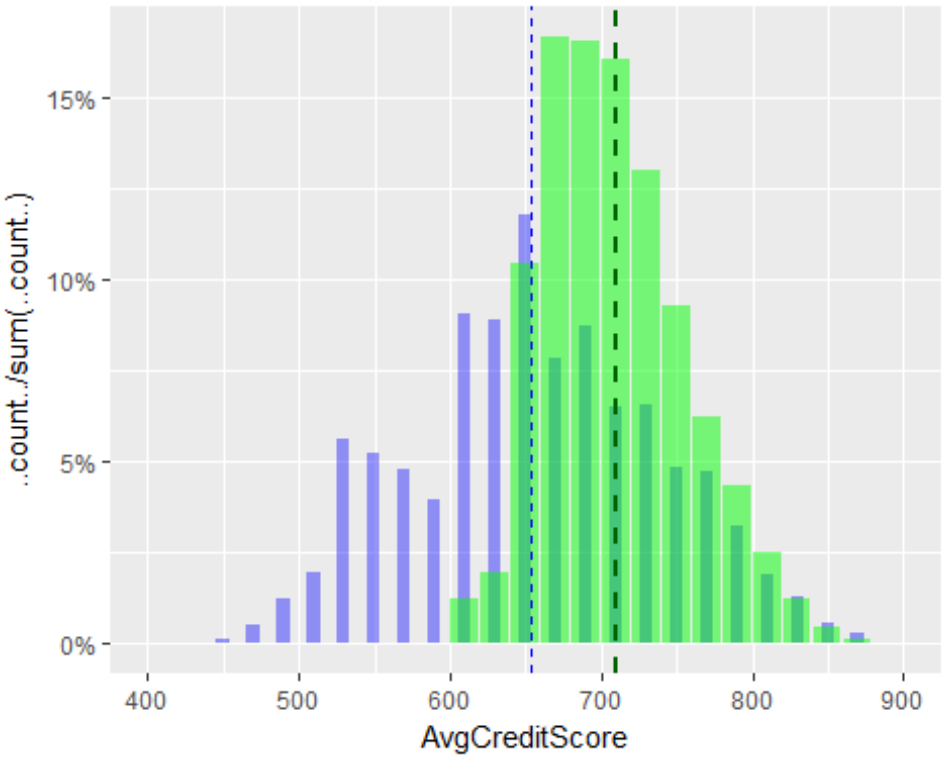
```
##
##                     A          AA           B           C           D
## 0.001694177 0.171194277 0.063202231 0.183312352 0.215831147 0.167935339
##           E          HR
## 0.115239361 0.081591115
```

```r
#We can see they spread out their loans more on A, B, and less on HR

#Plotting both to see how these distributions look
ggplot(ldpost2009, aes(ProsperRating..Alpha.)) +
  geom_histogram(data = ldpre2009, stat = 'count', aes(CreditGrade), fill =
'blue', alpha = 0.7) +
  geom_histogram(data = ldpost2009, stat = 'count',
aes(ProsperRating..Alpha.), fill = 'green', alpha = 0.4)
```

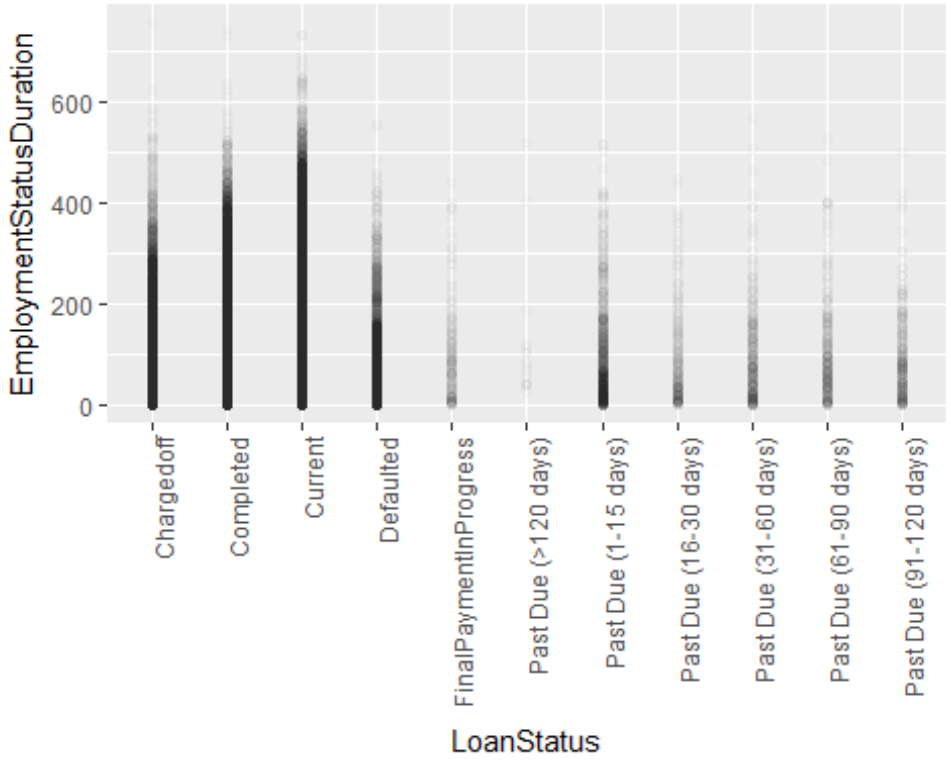



```
#Plot the pre and post 2009 distribution of credit score ratings
#We can see they have a more normal distribution after 2009, it looks like
they adjusted their risk in a good way
ggplot(ld, aes(x = AvgCreditScore)) +
  geom_histogram(data = ldpre2009, stat = 'count', aes(y = ..count.. /
sum(..count..)), fill = 'blue', alpha = 0.4) +
  geom_histogram(data = ldpost2009, stat = 'count', aes(y = ..count.. /
sum(..count..)), fill = 'green', alpha = 0.5)+
  scale_y_continuous(labels = scales::percent) +
  coord_cartesian(xlim = c(400, 900)) +
  geom_vline(data = ldpre2009, aes(xintercept = mean(AvgCreditScore, na.rm =
T)),
             color = 'blue', linetype='dashed', size=0.5) +
  geom_vline(data = ldpost2009, aes(xintercept = mean(AvgCreditScore, na.rm =
T)),
             color = 'dark green', linetype='dashed', size=1)
```




```
#First lets adjust the close date to be just the date (since time is not in
it anyway)
#First Recorded credit line also goes up to date level
ld$ClosedDate <- as.Date(ld$ClosedDate, "%Y-%m-%d")
ld$FirstRecordedCreditLine <- as.Date(ld$FirstRecordedCreditLine, "%Y-%m-%d")

ggplot(ld, aes(LoanStatus, EmploymentStatusDuration)) + geom_point(alpha =
1/100) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
#Lets check what loan statuses we have
summary(ld$LoanStatus)
```

```
##               Cancelled             Chargedoff               Completed
##                       0                  11951                   37910
##                 Current               Defaulted FinalPaymentInProgress
##                   56576                    4951                     205
##   Past Due (>120 days)   Past Due (1-15 days)  Past Due (16-30 days)
##                      16                     806                     265
##  Past Due (31-60 days)  Past Due (61-90 days) Past Due (91-120 days)
##                     363                     313                     304
```

```
#We can see we have many categories, including 6 past due categories which we
could simplify
```

```r
#Lets check to see where we should categorize the final payment loans by
looking at their % funded
#Because they are 99.65% funded we'll categorize them as 'Complete'
summary(subset(ld, LoanStatus=='FinalPaymentInProgress')$PercentFunded)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7055  1.0000  1.0000  0.9965  1.0000  1.0000
```

```r
#Summarizing All Past Dues as one category
#Summarizing Charged Off & Defaulted as one category
#When I used the case_when, it does this as a character instead of a vector,
so changing to vector after case_when
ld <- mutate(ld, Status = case_when (LoanStatus == 'Current' ~ 'Current',
                                     LoanStatus == 'Completed' ~ 'Completed',
                                     LoanStatus == 'FinalPaymentInProgress' &
PercentFunded >= 0.95 ~ 'Completed',
                                     LoanStatus == 'FinalPaymentInProgress' &
PercentFunded < 0.95 ~ 'Past Due',
                                     LoanStatus %in% c('Chargedoff',
'Defaulted') ~ 'Defaulted',
                                     LoanStatus %in% c('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)', 'Past Due (>120 days)') ~ 'Past Due'))

ld$Status <- factor(ld$Status, levels = c('Completed', 'Current', 'Past Due',
'Defaulted'))
summary(ld$Status)
```

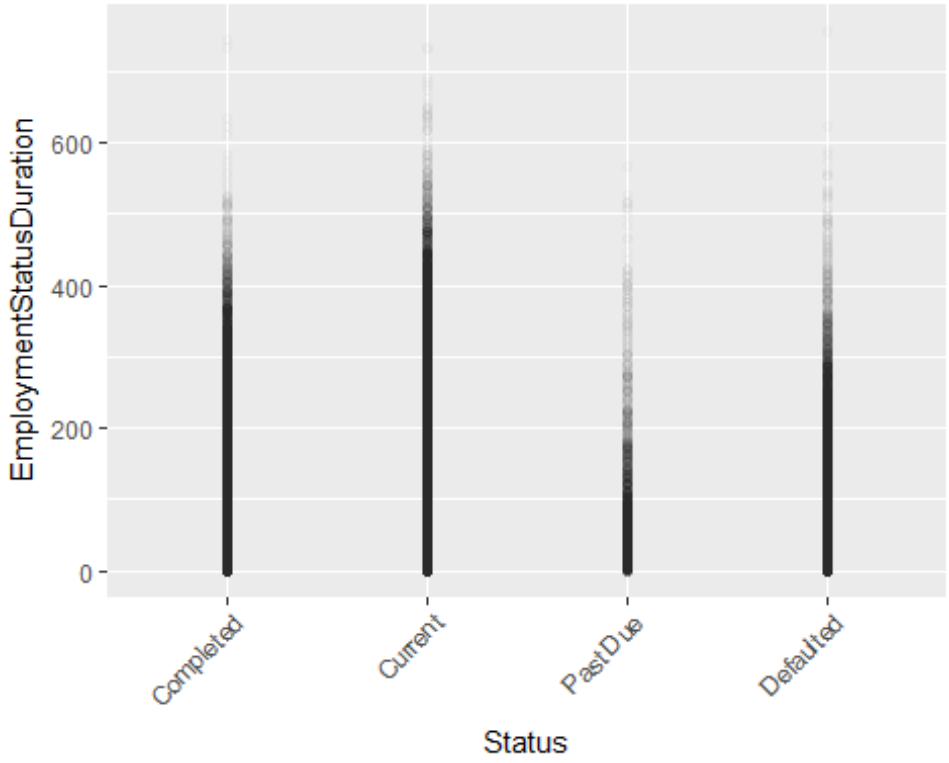
```
## Completed   Current  Past Due Defaulted
##     38112     56576      2070     16902
```

```r
summary(ld$Status)
```

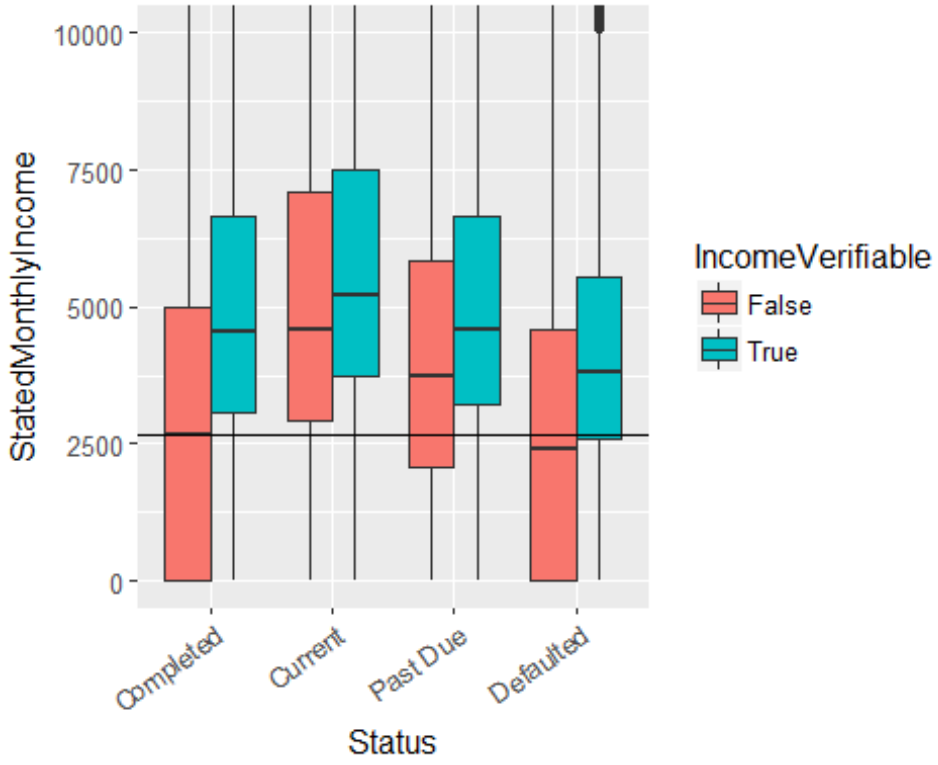
```
## Completed   Current  Past Due Defaulted
##     38112     56576      2070     16902
```

```
ggplot(ld, aes(Status, EmploymentStatusDuration)) + geom_point(alpha = 1/100)
+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 7480 rows containing missing values (geom_point).
```



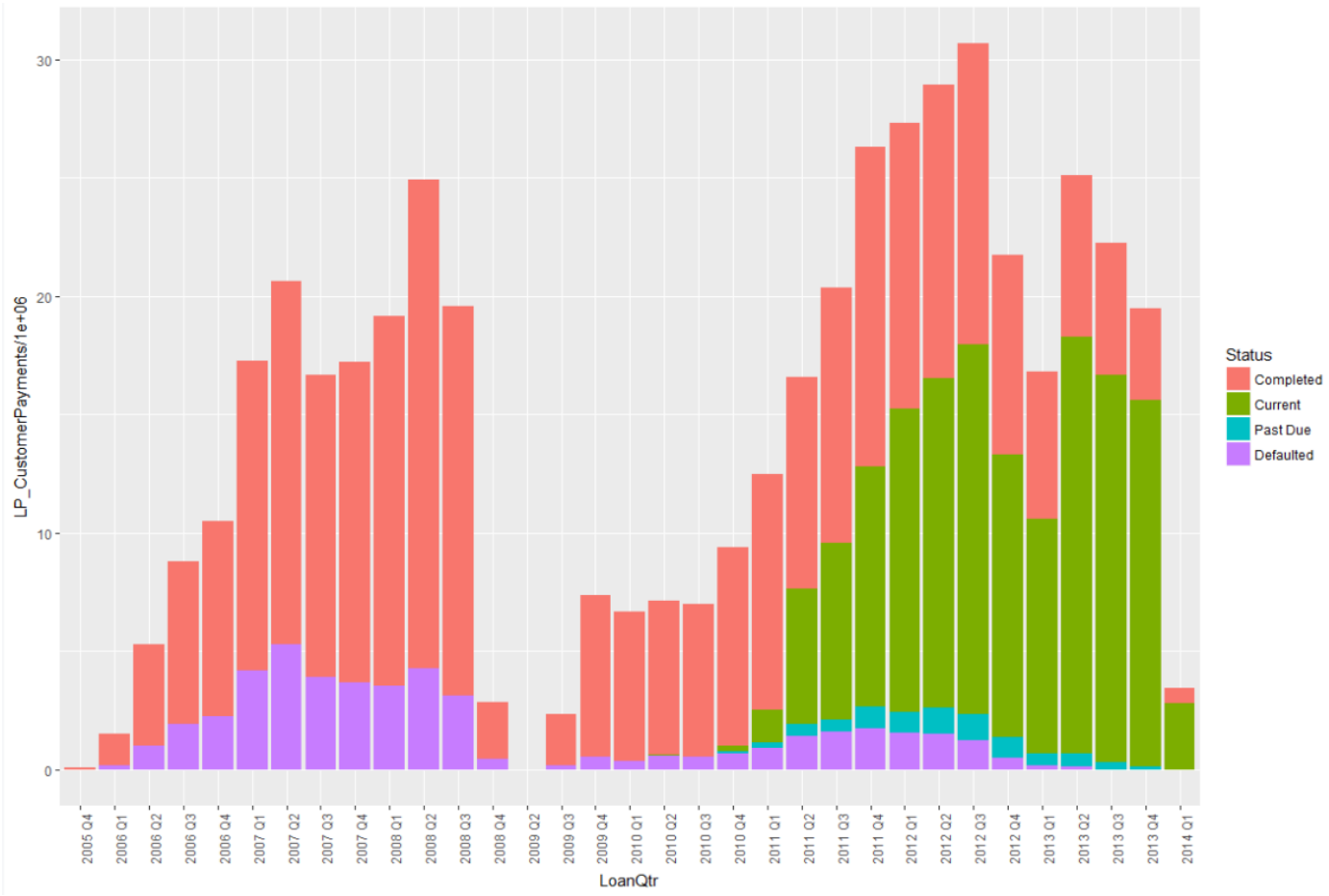
```
ggplot(ld, aes(Status, StatedMonthlyIncome, fill=IncomeVerifiable)) +
  geom_boxplot() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle =
35, hjust = 1)) +
  coord_cartesian(ylim = c(0,10000)) +
  geom_hline(yintercept=median(subset(ld, Status=='Completed' &
IncomeVerifiable=='False')$StatedMonthlyIncome))
```



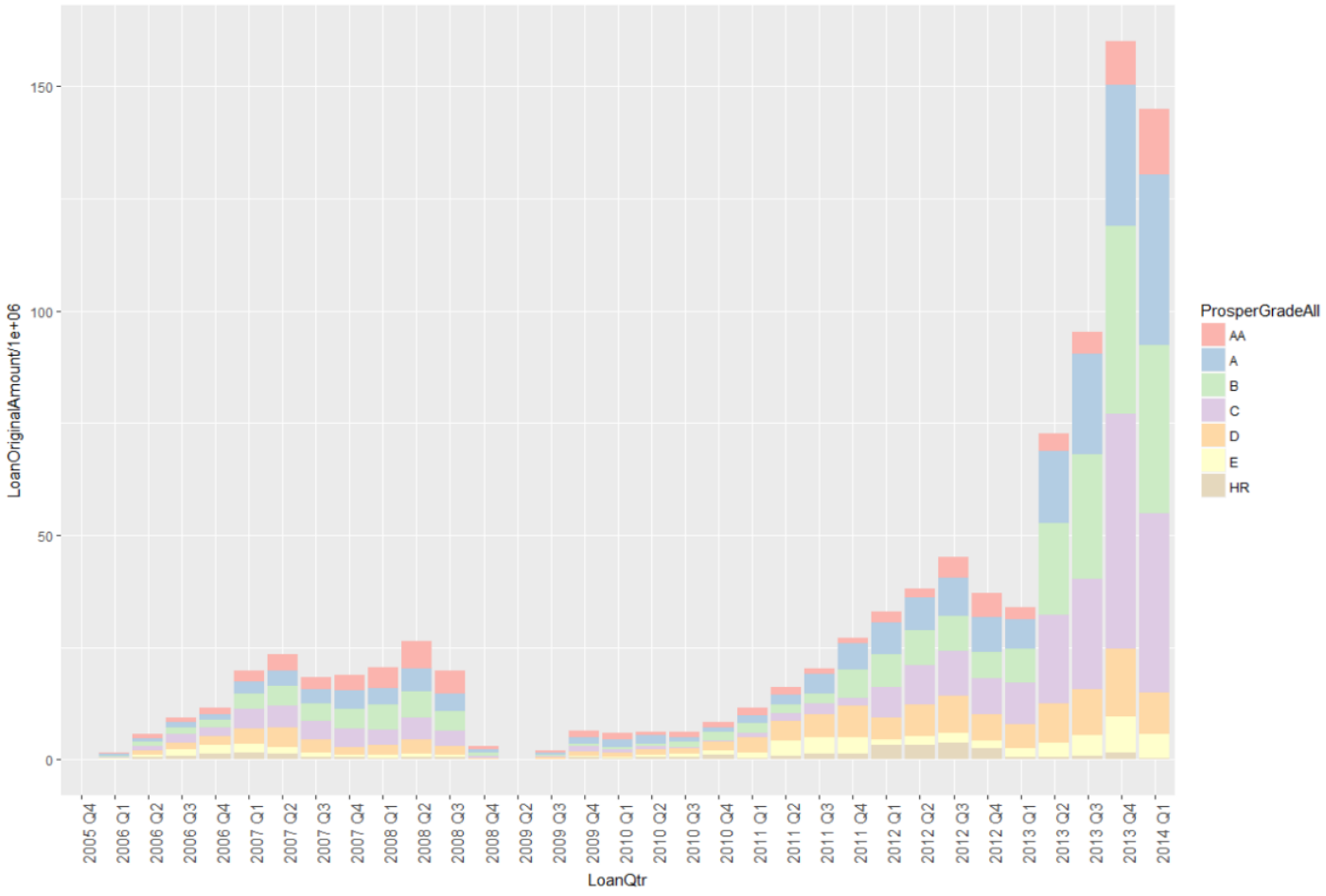

```
            #color = 'blue', linetype='dashed', size=0.5)
  #geom_vline(data = ldpre2009, aes(xintercept = mean(AvgCreditScore, na.rm =
T)),
  #scale_fill_gradient(low = "light blue", high = "dark red")

ld <- mutate(ld, LoanQtr = paste(substring(LoanOriginationQuarter, 4,8),
substring(LoanOriginationQuarter, 1,2)))
```

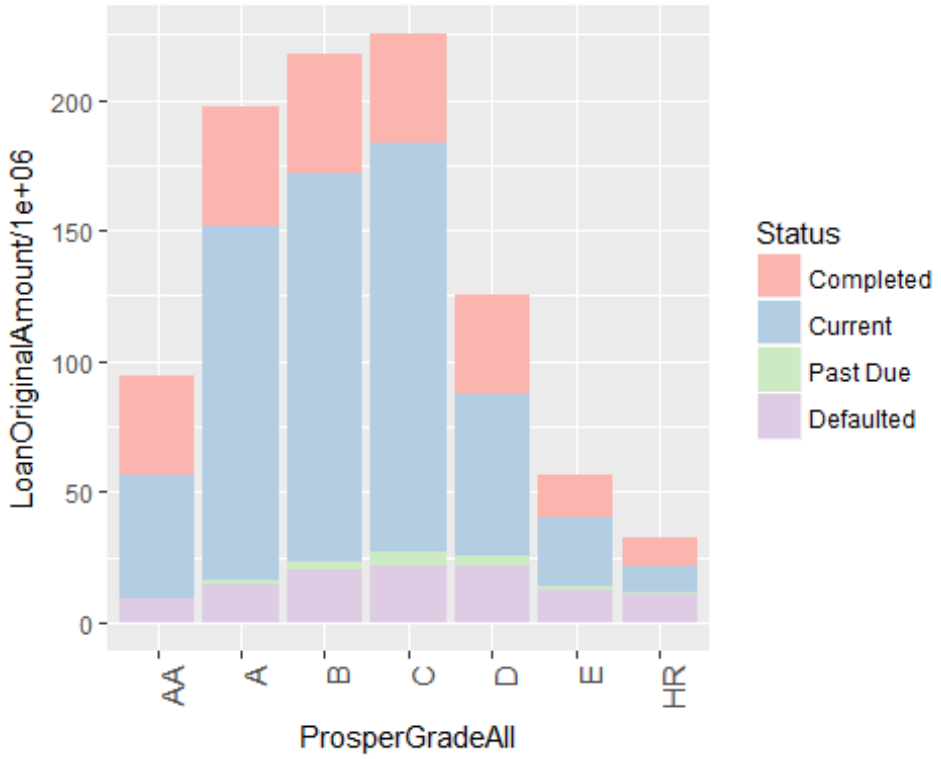
```
ggplot(ld, aes(LoanQtr, LP_CustomerPayments/1000000, fill = Status))
+geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggplot(ld, aes(LoanQtr, LoanOriginalAmount/1000000, fill = ProsperGradeAll))
+geom_bar(stat='identity') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size=11)) +
  scale_fill_brewer(palette="Pastel1")
```

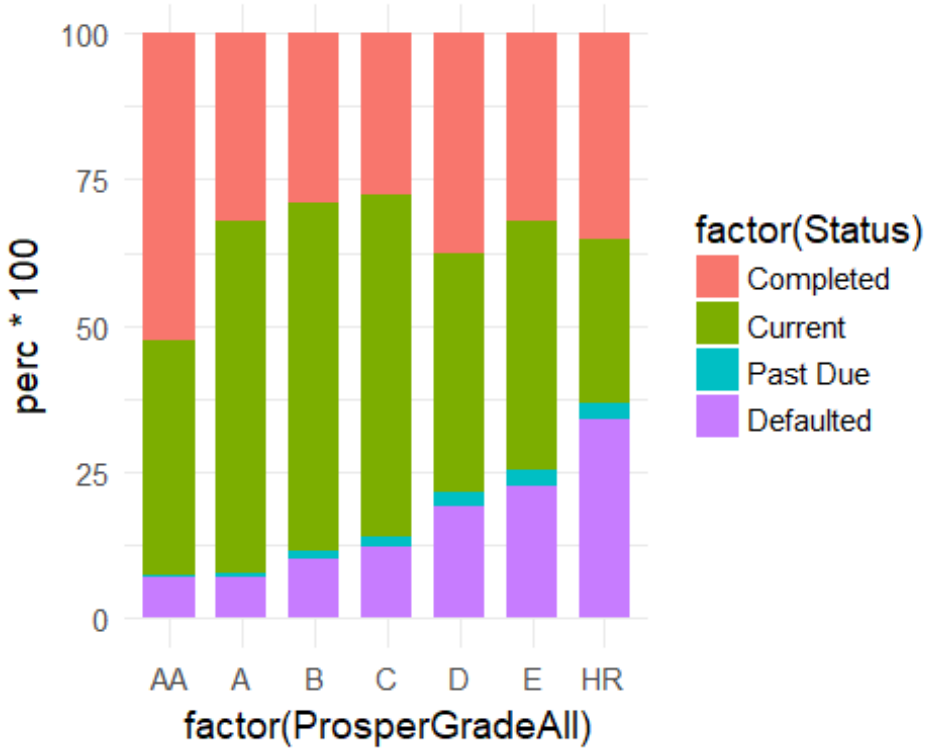


```
ggplot(ld, aes(ProsperGradeAll, LoanOriginalAmount/1000000, fill = Status))
+geom_bar(stat='identity') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size=11)) +
  scale_fill_brewer(palette="Pastel1")
```



```r
ld2 <- ld %>%
  group_by(ProsperGradeAll,Status) %>%
  summarise(count=n()) %>%
  mutate(perc=count/sum(count))


ggplot(ld2, aes(x = factor(ProsperGradeAll), y = perc*100, fill =
factor(Status))) +
  geom_bar(stat="identity", width = 0.7) +
  theme_minimal(base_size = 14)
```




```r
#Now lets make a dataframe of the completed loans
ldcomplete <- subset(ld, LoanStatus == 'Completed')

#ldcompletefact <- ldcomplete[,c('AvgCreditScore', 'ProsperScoreAll',
'EmploymentStatusDuration', 'IsBorrowerHomeowner', 'OpenCreditLines',
'TotalCreditLinespast7years', 'OpenRevolvingAccounts',
'InquiriesLast6Months', 'TotalInquiries', 'CurrentDelinquencies',
'DelinquenciesLast7Years', 'PublicRecordsLast10Years',
'PublicRecordsLast12Months', 'BankcardUtilization', 'TotalTrades',
'TradesNeverDelinquent..percentage.', 'TradesOpenedLast6Months',
'DebtToIncomeRatio', 'IncomeVerifiable', 'StatedMonthlyIncome')]

#Only using complete cases without any nulls
#ldcompletefact <- ldcompletefact[complete.cases(ldcompletefact),]
```