Lending Club Loan Data Analysis

Processing the Data

We load the data using read_csv() since it's faster.

```
path = "/Users/chenzheng/Downloads/Lending Club Loan Data/loan.csv"
LoanData = read_csv(path)
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   id = col_integer(),
##   member_id = col_integer(),
##   loan_amnt = col_double(),
##   funded_amnt = col_double(),
##   funded_amnt_inv = col_double(),
##   int_rate = col_double(),
##   installment = col_double(),
##   annual_inc = col_double(),
##   dti = col_double(),
##   delinq_2yrs = col_double(),
##   inq_last_6mths = col_double(),
##   mths_since_last_delinq = col_double(),
##   mths_since_last_record = col_double(),
##   open_acc = col_double(),
##   pub_rec = col_double(),
##   revol_bal = col_double(),
##   revol_util = col_double(),
##   total_acc = col_double(),
##   out_prncp = col_double(),
##   out_prncp_inv = col_double()
##   # ... with 11 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
## # A tibble: 6 x 74
##        id member_id loan_amnt funded_amnt funded_amnt_inv      term
##     <int>     <int>     <dbl>       <dbl>           <dbl>     <chr>
## 1 1077501   1296599      5000        5000            4975 36 months
## 2 1077430   1314167      2500        2500            2500 60 months
## 3 1077175   1313524      2400        2400            2400 36 months
## 4 1076863   1277178     10000       10000           10000 36 months
## 5 1075358   1311748      3000        3000            3000 60 months
## 6 1075269   1311441      5000        5000            5000 36 months
## # ... with 68 more variables: int_rate <dbl>, installment <dbl>,
## #   grade <chr>, sub_grade <chr>, emp_title <chr>, emp_length <chr>,
## #   home_ownership <chr>, annual_inc <dbl>, verification_status <chr>,
## #   issue_d <chr>, loan_status <chr>, pymnt_plan <chr>, url <chr>,
## #   desc <chr>, purpose <chr>, title <chr>, zip_code <chr>,
## #   addr_state <chr>, dti <dbl>, delinq_2yrs <dbl>,
## #   earliest_cr_line <chr>, inq_last_6mths <dbl>,
```

```
## #   mths_since_last_delinq <dbl>, mths_since_last_record <dbl>,
## #   open_acc <dbl>, pub_rec <dbl>, revol_bal <dbl>, revol_util <dbl>,
## #   total_acc <dbl>, initial_list_status <chr>, out_prncp <dbl>,
## #   out_prncp_inv <dbl>, total_pymnt <dbl>, total_pymnt_inv <dbl>,
## #   total_rec_prncp <dbl>, total_rec_int <dbl>, total_rec_late_fee <dbl>,
## #   recoveries <dbl>, collection_recovery_fee <dbl>, last_pymnt_d <chr>,
## #   last_pymnt_amnt <dbl>, next_pymnt_d <chr>, last_credit_pull_d <chr>,
## #   collections_12_mths_ex_med <dbl>, mths_since_last_major_derog <chr>,
## #   policy_code <dbl>, application_type <chr>, annual_inc_joint <chr>,
## #   dti_joint <chr>, verification_status_joint <chr>,
## #   acc_now_delinq <dbl>, tot_coll_amt <chr>, tot_cur_bal <chr>,
## #   open_acc_6m <chr>, open_il_6m <chr>, open_il_12m <chr>,
## #   open_il_24m <chr>, mths_since_rcnt_il <chr>, total_bal_il <chr>,
## #   il_util <chr>, open_rv_12m <chr>, open_rv_24m <chr>, max_bal_bc <chr>,
## #   all_util <chr>, total_rev_hi_lim <chr>, inq_fi <chr>,
## #   total_cu_tl <chr>, inq_last_12m <chr>
```

```
## [1] "# of Rows in Dataframe: 887379"
```

```
## [1] "Dataframe Size: 714 Mb"
```

change the date convension for clearly data visualization

```
LoanData$issue_date = as.Date(gsub("^", '01-', LoanData$issue_d), format = "%d-%b-%Y")
```

Feature Engineering

```
LoanData$month = month(LoanData$issue_date)
```

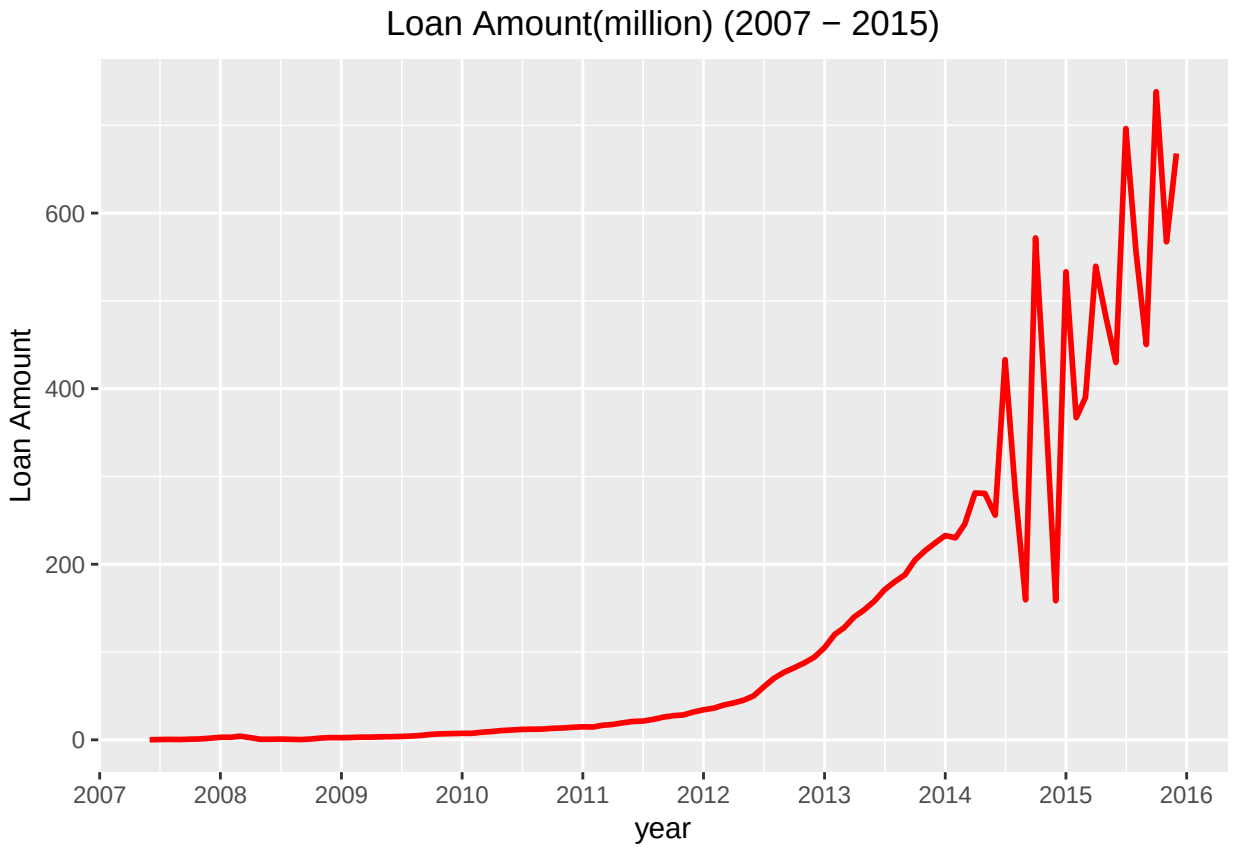
Create a chart of loan amount over time.

```
Loan_daily = LoanData %>% group_by(issue_date) %>% summarize(daily_amount = sum(loan_amnt)/1000000)

head(Loan_daily, nrow=10)
```

```
## # A tibble: 6 x 2
##   issue_date daily_amount
##       <date>        <dbl>
## 1 2007-06-01     0.091850
## 2 2007-07-01     0.348325
## 3 2007-08-01     0.515300
## 4 2007-09-01     0.372950
## 5 2007-10-01     0.753225
## 6 2007-11-01     1.008650
```

```
ggplot(Loan_daily, aes(x=issue_date, y=daily_amount)) + geom_line(color = "red", size = 1) +
  labs(x="year", y="Loan Amount", title = "Loan Amount(million) (2007 - 2015)") +
  theme(plot.title = element_text(hjust = 0.5))+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")
```



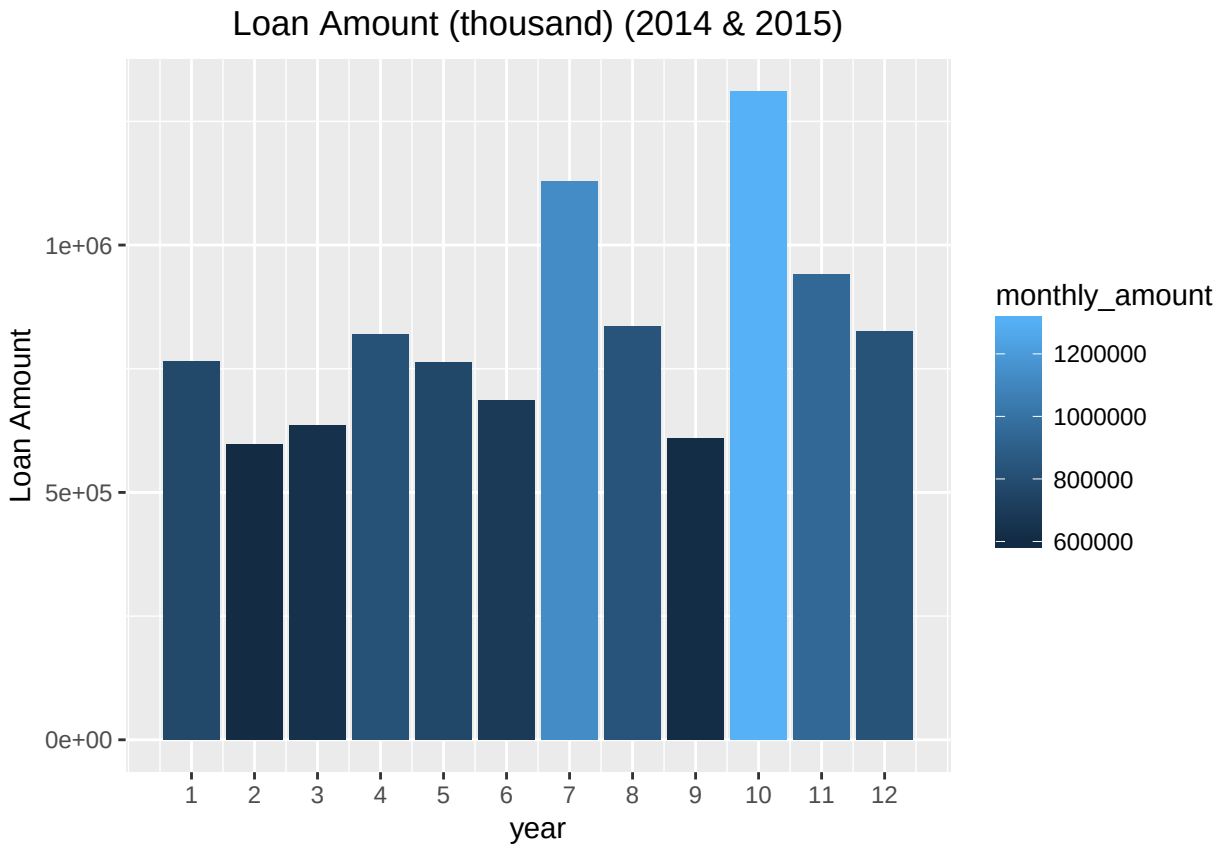
There are some daily variation between year 2014 and 2016. let's break it down by month of year in year of 2014 and 2015.

```
Loan_monthly = LoanData %>% filter(year(LoanData$issue_date) %in% c("2014", "2015")) %>%
  group_by(month) %>% summarise(monthly_amount = sum(loan_amnt)/1000)

head(Loan_monthly, nrow=10)
```

```
## # A tibble: 6 x 2
##   month monthly_amount
##   <dbl>          <dbl>
## 1     1       765847.9
## 2     2       596995.4
## 3     3       635766.1
## 4     4       820580.7
## 5     5       763851.1
## 6     6       685676.1
```

```
ggplot(Loan_monthly, aes(x = month, y = monthly_amount)) + geom_bar(stat = "identity", aes(fill=monthly
  scale_x_continuous(breaks = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  labs(x="year", y="Loan Amount", title = "Loan Amount (thousand) (2014 & 2015)") +
  theme(plot.title = element_text(hjust = 0.5))
```



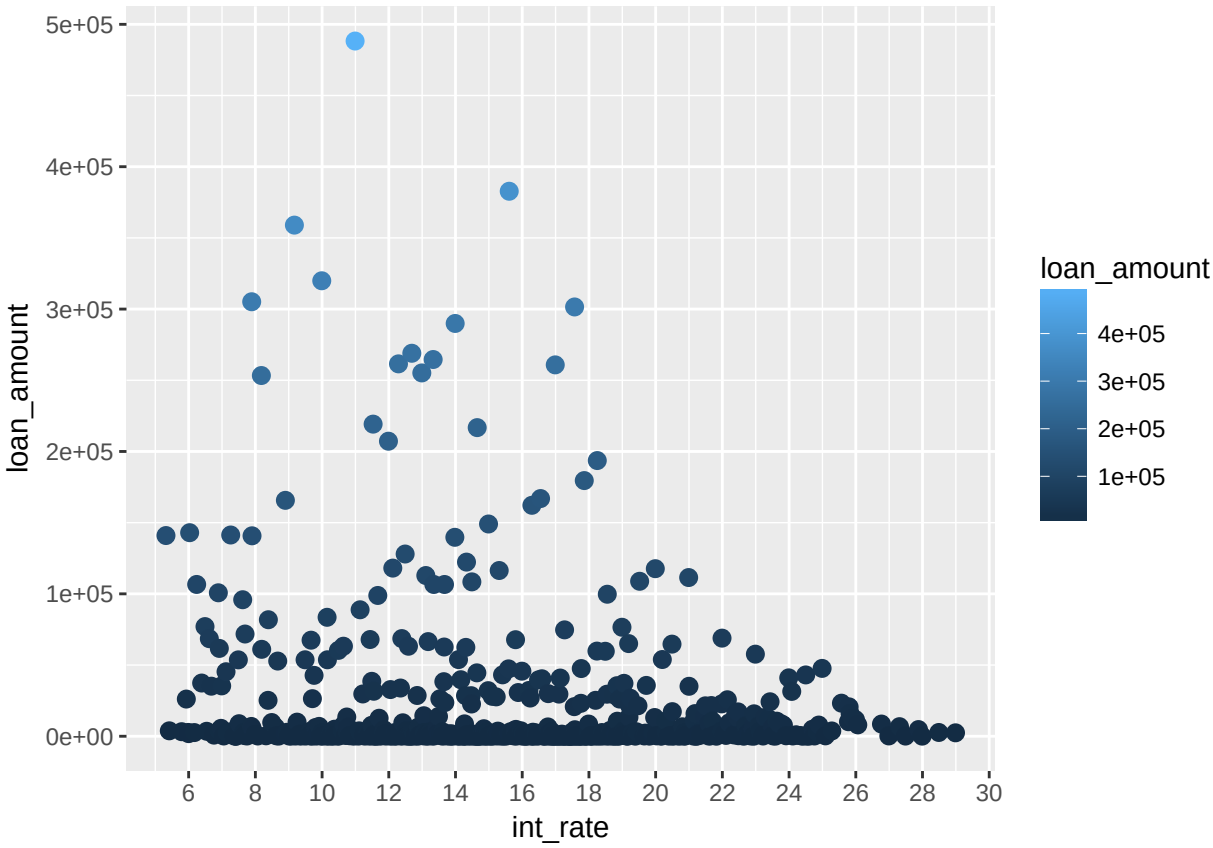
We can see the largest loan amount in July.

Loan amount VS. interest rate

```
Loan_by_interestrate = LoanData %>% group_by(int_rate) %>% summarise(loan_amount = sum(loan_amnt)/1000)
head(Loan_by_interestrate)
```

```
## # A tibble: 6 x 2
##   int_rate loan_amount
##      <dbl>       <dbl>
## 1     5.32  140957.375
## 2     5.42    3797.300
## 3     5.79    3178.850
## 4     5.93   26116.800
## 5     5.99    2683.025
## 6     6.00    1810.850
```

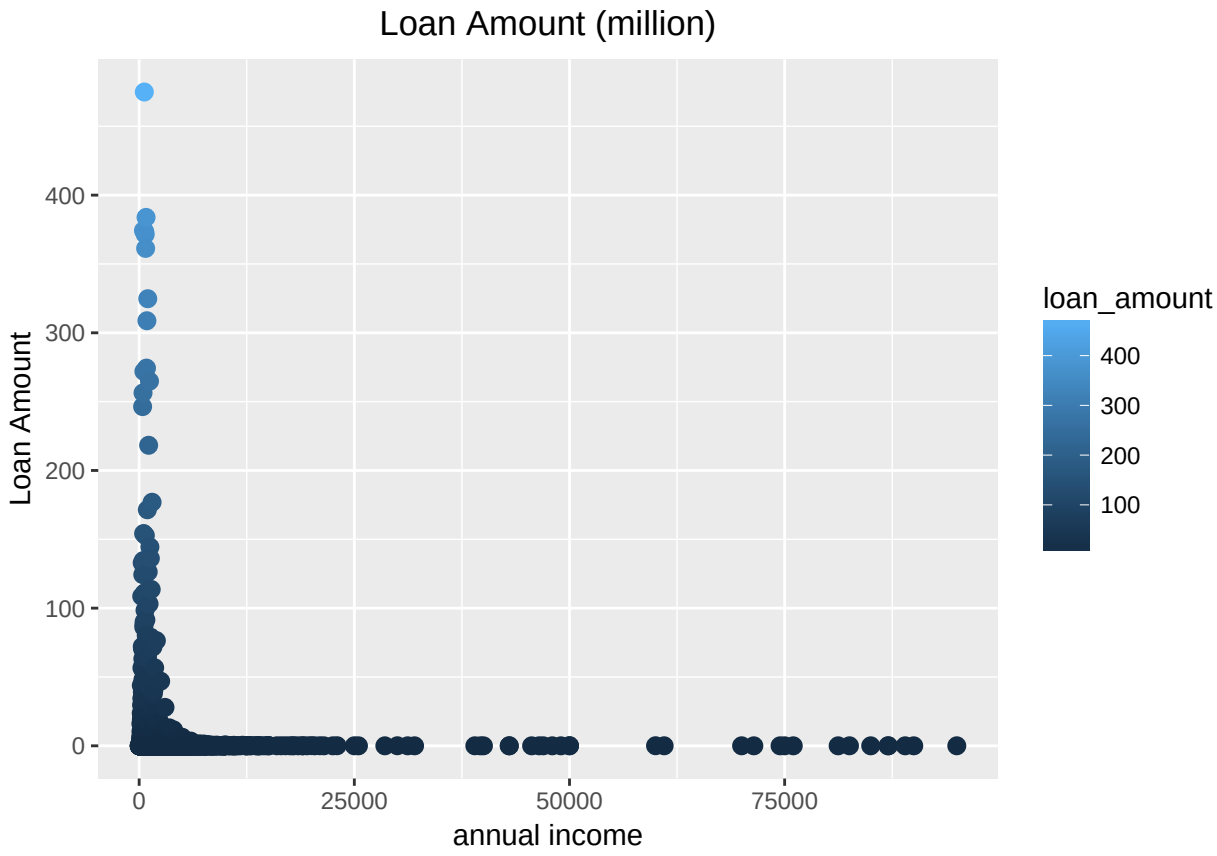
```
ggplot(Loan_by_interestrate, aes(x = int_rate, y=loan_amount, color = loan_amount)) +
  geom_point(shape = 16, size = 3) + scale_x_continuous(breaks = seq(0, 30, by=2))
```



Loan amount VS. annual income

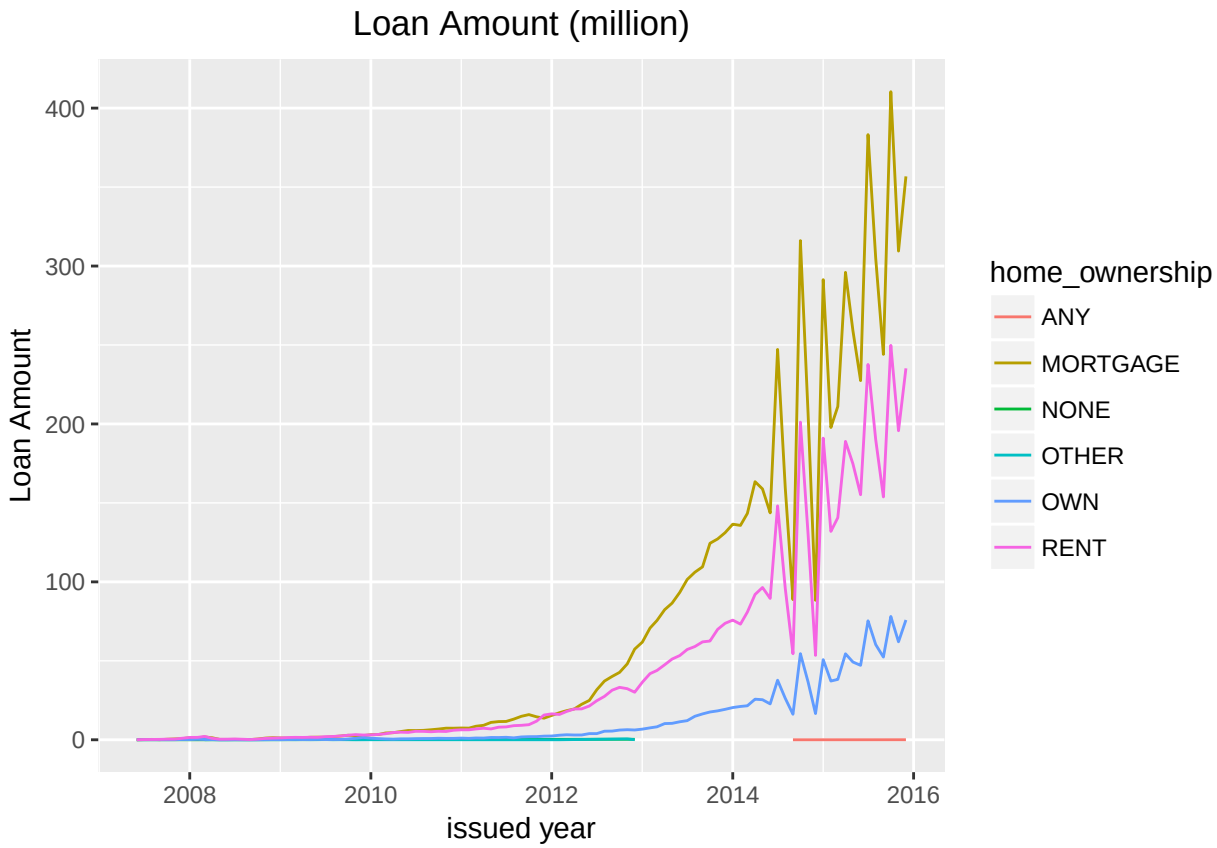
```
Loan_by_income = LoanData %>% group_by(annual_inc) %>% summarise(loan_amount = sum(loan_amnt, na.rm = TR
ggplot(Loan_by_income, aes(x = annual_inc/100, y=loan_amount, color = loan_amount)) +
  geom_point(shape = 16, size = 3) +
  labs(x="annual income", y="Loan Amount", title = "Loan Amount (million)") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



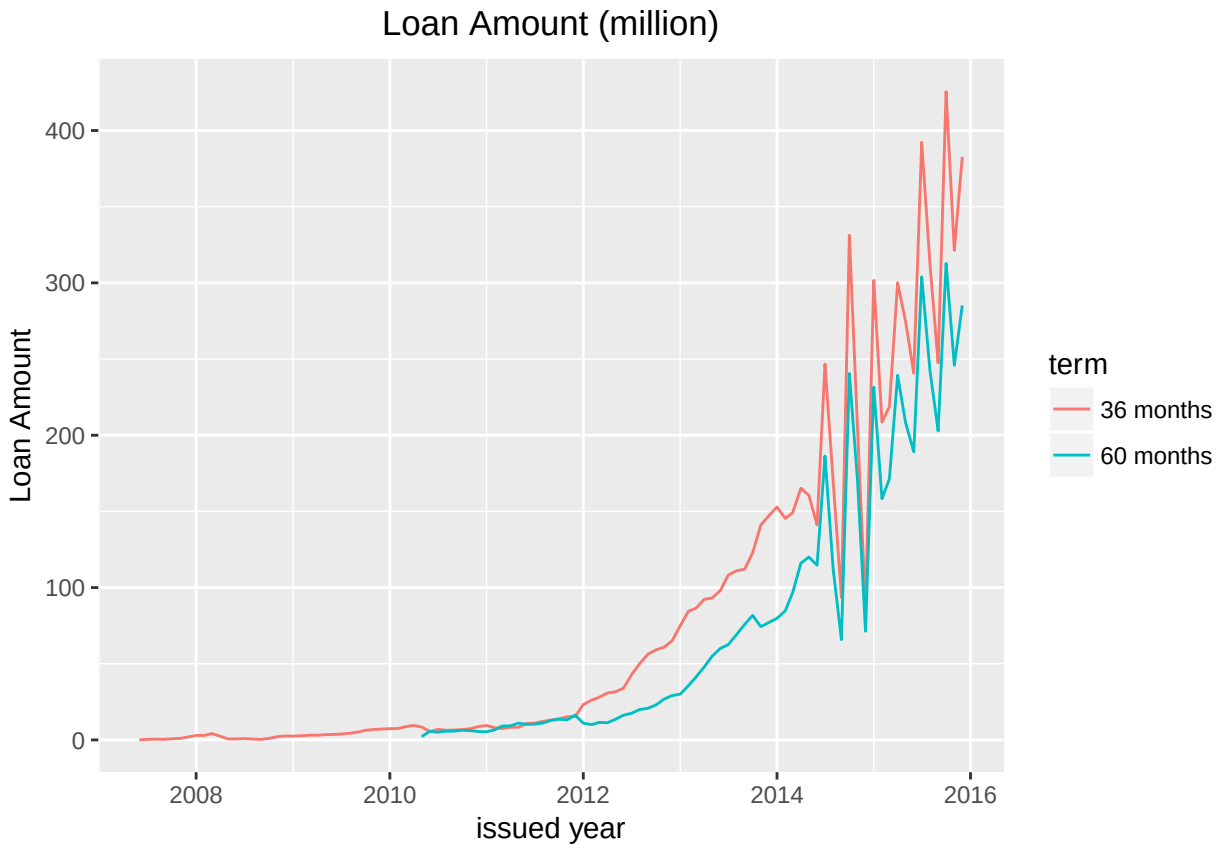
Loan amount by home ownership

```
Loan_by_ownership = LoanData %>% group_by(issue_date, home_ownership) %>% summarise(loan_amount = sum(lo
ggplot(Loan_by_ownership, aes(x = issue_date, y = loan_amount, colour = home_ownership)) + geom_line() +
  labs(x="issued year", y="Loan Amount", title = "Loan Amount (million)") +
  theme(plot.title = element_text(hjust = 0.5))
```



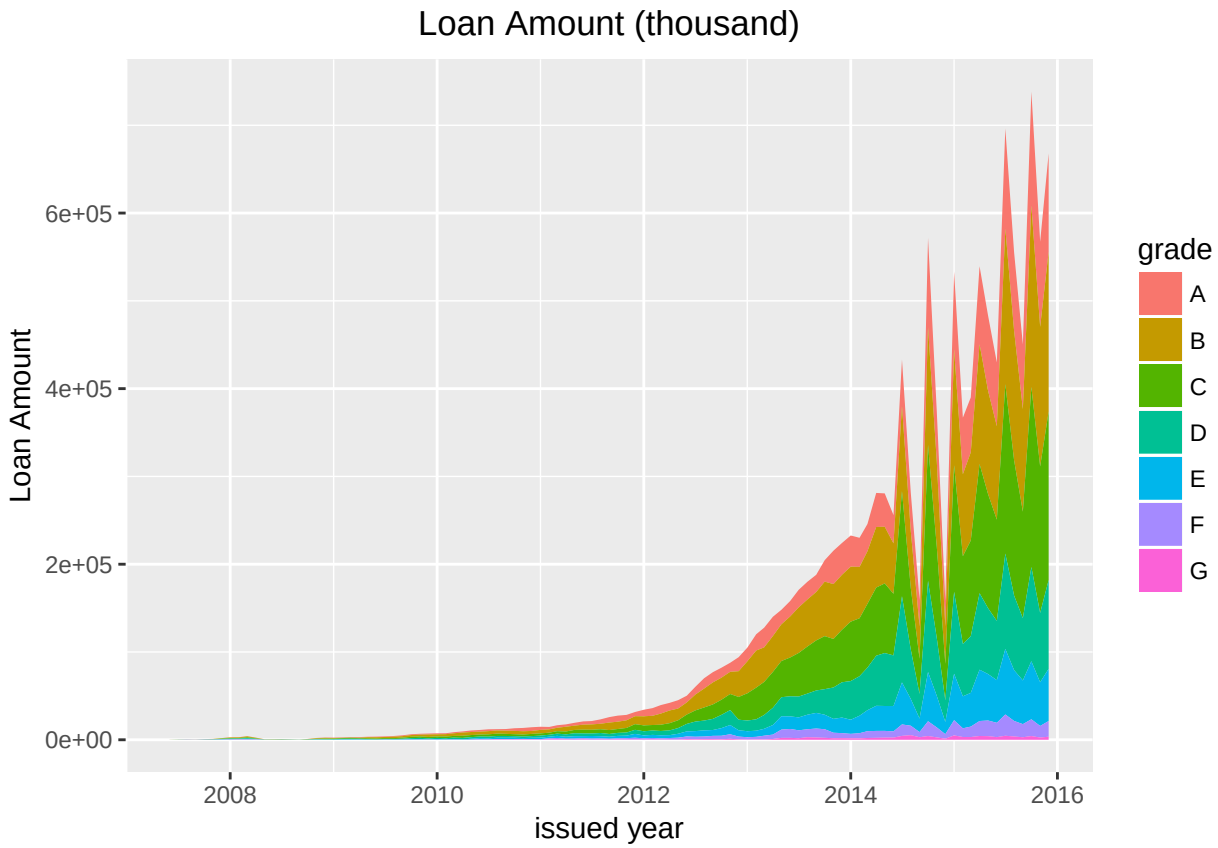
Loan amount by loan term

```
Loan_by_term = LoanData %>% group_by(issue_date, term) %>% summarise(loan_amount = sum(loan_amnt)/100000
ggplot(Loan_by_term, aes(x = issue_date, y = loan_amount, colour = term)) + geom_line() +
  labs(x="issued year", y="Loan Amount", title = "Loan Amount (million)") +
  theme(plot.title = element_text(hjust = 0.5))
```



Loan amount by grade

```
Loan_by_grade = LoanData %>% group_by(issue_date, grade) %>% summarise(loan_amount = sum(loan_amnt)/1000
ggplot(Loan_by_grade, aes(x=issue_date, y = loan_amount)) + geom_area(aes(fill=grade)) +
  labs(x="issued year", y="Loan Amount", title = "Loan Amount (thousand)") +
  theme(plot.title = element_text(hjust = 0.5))
```


Loan Amount (thousand)
Loan Amount
6e+05
4e+05
2e+05
0e+00
2008
2010
2012
2014
2016
issued year
grade
A
B
C
D
E
F
G