Lending Club Analysis

Siddhesh Aher

2025-12-13

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
## ========================================================
## Lending Club - Final Case Analysis (2012-2017)
## Data Science Tools
## ========================================================

## ------------------------------
## 1. Load Required Libraries
## ------------------------------
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(ggplot2)
library (usmap)
```

```
## Warning: package 'usmap' was built under R version 4.5.2
```

```
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.5.2
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(dplyr)
library(ggthemes)
library(stringr)
library(lubridate)
```

```
#--------------sETWD ACCOURING TO YOUR FILES

setwd("D:\\NEC\\OneDrive - New England College\\Data_Science_Tools -202547-CRN129\\Assignment 14\\Final

#------fILES LOADED

list.files()
```

```
##  [1] "Case Study.R"              "check.pdf"
##  [3] "check.Rmd"                 "check_files"
##  [5] "data2012.csv"              "data2013.csv"
##  [7] "data2014.csv"              "data2015.csv"
##  [9] "data2016.csv"              "data2017.csv"
## [11] "DataDictionary.csv"        "Final-Case-Analysis.pdf"
## [13] "Final-Case-Analysis.Rmd"   "Final-Case-Analysis_files"
## [15] "Final Case Analysis.docx"  "Final Case Analysis.Rmd"
## [17] "LCsample.csv"              "States.csv"
## [19] "states_regions.csv"
```

```
# run the below to see if the files are loaded if it says character(0) that means no data loaded
ls()
```

```
## character(0)
```

```
## ------------------------------
## 2. Import Lending Club Files. Stacking all six Lending Club files
```

```
## ------------------------------
# Update paths to where your files are stored
files <- list.files(pattern = "^data[0-9]+\\.csv$", full.names = TRUE)

loan_raw <- files %>%
  map_df(~ read_csv(.x, col_types = cols(.default = "c")))
```

```
## New names:
## New names:
## New names:
## New names:
## New names:
## New names:
## * `` -> `...1`
```

```
loan_raw <- clean_names(loan_raw)

##Then convert numeric columns:

loan_raw <- loan_raw %>%
  mutate(across(where(~ all(grepl("^[0-9 .-]*$", .x))), as.numeric))


##--Clean corrupted numeric values

loan_raw <- loan_raw %>%
  mutate(
    across(
      c(inq_last_6mths, pub_rec_bankruptcies),
      ~ ifelse(grepl("\\..*\\.", .x), NA, .x)
    )
  )



##-- Remove bad rows where the CSV header was duplicated

loan_raw <- loan_raw %>%
  filter(mths_since_last_delinq != "Unnamed: 28")


##--Run this to how the output: Should not be character(0)

str(loan_raw)
```

```
## tibble [668,348 x 32] (S3: tbl_df/tbl/data.frame)
##  $ x1                    : num [1:668348] 1 4 5 8 9 11 13 14 15 16 ...
##  $ loan_amnt             : chr [1:668348] "4500" "16950" "20675" "15000" ...
##  $ term                  : chr [1:668348] "36 months" "60 months" "36 months" "36 months" ...
##  $ int_rate              : chr [1:668348] "7.90%" "13.11%" "12.12%" "14.33%" ...
##  $ grade                 : chr [1:668348] "A" "B" "B" "C" ...
##  $ sub_grade             : chr [1:668348] "A4" "B4" "B3" "C1" ...
##  $ emp_title             : chr [1:668348] "acmc" "Wallauer Paint" "KJWW Engineering" "PPG Industries
```

```
##  $ emp_length            : chr [1:668348] "6 years" "10+ years" "1 year" "4 years" ...
##  $ home_ownership        : chr [1:668348] "MORTGAGE" "MORTGAGE" "RENT" "RENT" ...
##  $ annual_inc            : chr [1:668348] "115000" "115000" "120000" "65000" ...
##  $ issue_d               : chr [1:668348] "Jan-13" "Jan-13" "Jan-13" "Jan-13" ...
##  $ loan_status           : chr [1:668348] "Fully Paid" "Fully Paid" "Fully Paid" "Fully Paid" ...
##  $ purpose               : chr [1:668348] "home_improvement" "debt_consolidation" "debt_consolidatio
##  $ title                 : chr [1:668348] "home improvement" "Freedom" "Debt consolidation" "Debt co
##  $ addr_state            : chr [1:668348] "CA" "NY" "WI" "PA" ...
##  $ dti                   : chr [1:668348] "2.95" "30.9" "10.94" "17.75" ...
##  $ delinq_2yrs           : chr [1:668348] "1" "0" "1" "0" ...
##  $ inq_last_6mths        : chr [1:668348] "0" "2" "0" "0" ...
##  $ mths_since_last_delinq: chr [1:668348] "19" "46" "11" "34" ...
##  $ open_acc              : chr [1:668348] "12" "22" "14" "20" ...
##  $ total_pymnt           : chr [1:668348] "5069.003889" "23177.50107" "24902.25519" "16867.46853" ..
##  $ total_rec_int         : chr [1:668348] "569" "6227.5" "4197.26" "1867.47" ...
##  $ last_pymnt_d          : chr [1:668348] "Jan-16" "Oct-17" "Jan-16" "May-14" ...
##  $ last_pymnt_amnt       : chr [1:668348] "140.65" "1526.78" "687.72" "1145.66" ...
##  $ last_credit_pull_d    : chr [1:668348] "Dec-18" "Aug-18" "Mar-17" "Jun-19" ...
##  $ application_type      : chr [1:668348] "Individual" "Individual" "Individual" "Individual" ...
##  $ tot_cur_bal           : chr [1:668348] "723648" "599056" "42341" "62477" ...
##  $ acc_open_past_24mths  : chr [1:668348] "8" "1" "3" "14" ...
##  $ pub_rec_bankruptcies  : chr [1:668348] "0" "0" "1" "0" ...
##  $ orig_index            : num [1:668348] 133797 133920 133927 133997 134001 ...
##  $ issue_month           : chr [1:668348] "Jan" "Jan" "Jan" "Jan" ...
##  $ issue_year            : chr [1:668348] "2013" "2013" "2013" "2013" ...
```

```
colnames(loan_raw)
```

```
##  [1] "x1"                    "loan_amnt"             "term"
##  [4] "int_rate"              "grade"                 "sub_grade"
##  [7] "emp_title"             "emp_length"            "home_ownership"
## [10] "annual_inc"            "issue_d"               "loan_status"
## [13] "purpose"               "title"                 "addr_state"
## [16] "dti"                   "delinq_2yrs"           "inq_last_6mths"
## [19] "mths_since_last_delinq" "open_acc"             "total_pymnt"
## [22] "total_rec_int"         "last_pymnt_d"          "last_pymnt_amnt"
## [25] "last_credit_pull_d"    "application_type"      "tot_cur_bal"
## [28] "acc_open_past_24mths"  "pub_rec_bankruptcies"  "orig_index"
## [31] "issue_month"           "issue_year"
```

```
## ------------------------------
## 3. Import State Files
## ------------------------------
states <- read_csv("states.csv",show_col_types = FALSE) %>% clean_names()

colnames(states)
```

```
## [1] "geography"
## [2] "num_households"
## [3] "median_income_households"
## [4] "unemployment_rate_estimate_population_16_years_and_over"
## [5] "percent_below_poverty"
## [6] "population"
```

```
## [7] "males"
## [8] "females"
```

```
## -------------------------------
## 3. Import States Region Files
## -------------------------------

regions <- read_csv("states_regions.csv",show_col_types = FALSE) %>% clean_names()
```

```
## New names:
## * `` -> `...5`
```

```
colnames(regions)
```

```
## [1] "state"      "state_code" "region"     "division"   "x5"
```

```
##--Verify that files actually exist
list.files()
```

```
##  [1] "Case Study.R"              "check.pdf"
##  [3] "check.Rmd"                 "check_files"
##  [5] "data2012.csv"              "data2013.csv"
##  [7] "data2014.csv"              "data2015.csv"
##  [9] "data2016.csv"              "data2017.csv"
## [11] "DataDictionary.csv"        "Final-Case-Analysis.pdf"
## [13] "Final-Case-Analysis.Rmd"   "Final-Case-Analysis_files"
## [15] "Final Case Analysis.docx"  "Final Case Analysis.Rmd"
## [17] "LCsample.csv"              "States.csv"
## [19] "states_regions.csv"
```

```
## -------------------------------
## 4. Clean & Standardize State Names. Cleaning state codes

## -------------------------------

loan_clean <- loan_raw %>%
  mutate(
    addr_state = str_trim(addr_state),
    addr_state = str_to_upper(addr_state)
  ) %>%
  rename(state = addr_state)


## -------------------------------
## 5. Merge Lending Data With States & Regions
## -------------------------------

##loan_clean$state (abbreviations). To merge with states.csv, we need full state names:

state_abbrev <- c(
  AL="Alabama", AK="Alaska", AZ="Arizona", AR="Arkansas", CA="California",
  CO="Colorado", CT="Connecticut", DE="Delaware", FL="Florida", GA="Georgia",
```

```
  HI="Hawaii", ID="Idaho", IL="Illinois", IN="Indiana", IA="Iowa",
  KS="Kansas", KY="Kentucky", LA="Louisiana", ME="Maine", MD="Maryland",
  MA="Massachusetts", MI="Michigan", MN="Minnesota", MS="Mississippi",
  MO="Missouri", MT="Montana", NE="Nebraska", NV="Nevada", NH="New Hampshire",
  NJ="New Jersey", NM="New Mexico", NY="New York", NC="North Carolina",
  ND="North Dakota", OH="Ohio", OK="Oklahoma", OR="Oregon", PA="Pennsylvania",
  RI="Rhode Island", SC="South Carolina", SD="South Dakota", TN="Tennessee",
  TX="Texas", UT="Utah", VT="Vermont", VA="Virginia", WA="Washington",
  WV="West Virginia", WI="Wisconsin", WY="Wyoming"
)

loan_clean <- loan_clean %>%
  mutate(state_full = state_abbrev[state])

##Merge with states.csv using full state name

loan_states <- loan_clean %>%
  left_join(states, by = c("state_full" = "geography"))

## regions$State is full name.

loan_states_regions <- loan_states %>%
  left_join(regions %>% select(state, region, division), by = c("state_full" = "state"))


## colnames(states) -- check for column names

## ------------------------------
## 6. Remove Missing or Invalid States
## ------------------------------
# Check how many rows have missing population
sum(is.na(loan_states_regions$population))
```

```
## [1] 1581
```

```
# Remove rows with missing population
loan_states_regions <- loan_states_regions %>%
  filter(!is.na(population))

# Inspect the cleaned dataframe
loan_states_regions
```

```
## # A tibble: 666,767 x 42
##       x1 loan_amnt term      int_rate grade sub_grade emp_title       emp_length
##    <dbl> <chr>     <chr>     <chr>    <chr> <chr>     <chr>           <chr>
##  1     1 4500      36 months 7.90%    A     A4        acmc            6 years
##  2     4 16950     60 months 13.11%   B     B4        Wallauer Paint  10+ years
##  3     5 20675     36 months 12.12%   B     B3        KJWW Engineeri~ 1 year
##  4     8 15000     36 months 14.33%   C     C1        PPG Industries~ 4 years
##  5     9 14950     36 months 14.33%   C     C1        Accsys Technol~ 5 years
##  6    11 10000     36 months 10.16%   B     B1        Trilogy Health~ 5 years
##  7    13 15250     36 months 15.80%   C     C3        Mystic Medical~ 10+ years
##  8    14 30000     36 months 18.75%   D     D3        Sotera Defense~ 3 years
```

```
##  9    15 12000     36 months 12.12%   B     B3        Loftco Constru~ 2 years
## 10    16 12000     36 months 12.12%   B     B3        City of San Di~ 5 years
## # i 666,757 more rows
## # i 34 more variables: home_ownership <chr>, annual_inc <chr>, issue_d <chr>,
## #   loan_status <chr>, purpose <chr>, title <chr>, state <chr>, dti <chr>,
## #   delinq_2yrs <chr>, inq_last_6mths <chr>, mths_since_last_delinq <chr>,
## #   open_acc <chr>, total_pymnt <chr>, total_rec_int <chr>, last_pymnt_d <chr>,
## #   last_pymnt_amnt <chr>, last_credit_pull_d <chr>, application_type <chr>,
## #   tot_cur_bal <chr>, acc_open_past_24mths <chr>, ...
```

```
## ========================================================
## ANALYSIS SECTION
## ========================================================

## ------------------------------
## A. Distribution of loans by state
## ------------------------------
loans_by_state <- loan_states_regions %>%
  group_by(state_full) %>%
  summarise(num_loans = n()) %>%
  arrange(desc(num_loans))


## Per capita loans
loans_per_capita <- loan_states_regions %>%
  group_by(state_full, population) %>%
  summarise(num_loans = n()) %>%
  mutate(loans_per_capita = num_loans / population)
```

```
## `summarise()` has grouped output by 'state_full'. You can override using the
## `.groups` argument.
```

```
## Distribution by region / division
loans_by_region <- loan_states_regions %>%
  group_by(region) %>%
  summarise(num_loans = n())

loans_by_division <- loan_states_regions %>%
  group_by(division) %>%
  summarise(num_loans = n())

head(loans_per_capita, 5)
```

```
## # A tibble: 5 x 4
## # Groups:   state_full [5]
##   state_full population num_loans loans_per_capita
##   <chr>           <dbl>     <int>            <dbl>
## 1 Alabama       4850771      8494          0.00175
## 2 Alaska         738565      1614          0.00219
## 3 Arizona       6809946     16029          0.00235
## 4 Arkansas      2977944      4980          0.00167
## 5 California   38982847     91928          0.00236
```

```r
head(loans_by_region, 5)
```

```
## # A tibble: 4 x 2
##   region    num_loans
##   <chr>         <int>
## 1 Midwest      115685
## 2 Northeast    140046
## 3 South        241964
## 4 West         169072
```

```r
head(loans_by_division, 5)
```

```
## # A tibble: 5 x 2
##   division           num_loans
##   <chr>                  <int>
## 1 East North Central     83817
## 2 East South Central     28858
## 3 Middle Atlantic       103728
## 4 Mountain               50880
## 5 New England            36318
```

```r
##Missing states check
all_states <- unique(states$geography)
observed_states <- unique(loan_states_regions$state_full)
missing_states <- setdiff(all_states, observed_states)
missing_states
```

```
## [1] "District of Columbia" "Puerto Rico"
```

```r
## -------------------------------
## B. Average loan amounts by state / division
## -------------------------------


##Before using mean() or other numeric operations, always check the type:
str(loan_states_regions$loan_amnt)
```

```
##  chr [1:666767] "4500" "16950" "20675" "15000" "14950" "10000" "15250" ...
```

```r
##Convert loan_amnt to numeric

loan_states_regions <- loan_states_regions %>%
  mutate(loan_amnt = as.numeric(loan_amnt))

##Average by State
avg_loan_state <- loan_states_regions %>%
  group_by(state_full) %>%
  summarise(avg_loan_amount = mean(loan_amnt, na.rm = TRUE))

head(avg_loan_state, 5)
```

```
## # A tibble: 5 x 2
##   state_full avg_loan_amount
##   <chr>                <dbl>
## 1 Alabama             14316.
## 2 Alaska              16892.
## 3 Arizona             14442.
## 4 Arkansas            13805.
## 5 California          15070.
```

```
##Average by division

avg_loan_division <- loan_states_regions %>%
  group_by(division) %>%
  summarise(avg_loan_amount = mean(loan_amnt, na.rm = TRUE))

head(avg_loan_division, 5)
```

```
## # A tibble: 5 x 2
##   division           avg_loan_amount
##   <chr>                        <dbl>
## 1 East North Central          14423.
## 2 East South Central          14327.
## 3 Middle Atlantic             14881.
## 4 Mountain                    14578.
## 5 New England                 15089.
```

```
## -------------------------------
## C. Loan Grade - Avg Interest Rate & Amount
## -------------------------------

##--Before using mean() or other numeric operations, always check the type:
str(loan_states_regions$int_rate)
```

```
##  chr [1:666767] "7.90%" "13.11%" "12.12%" "14.33%" "14.33%" "10.16%" ...
```

```
# Convert int_rate to numeric

loan_states_regions <- loan_states_regions %>%
  mutate(int_rate = as.numeric(str_remove(int_rate, "%")))



grade_summary <- loan_states_regions %>%
  group_by(grade) %>%
  summarise(
    avg_interest = mean(int_rate, na.rm = TRUE),
    avg_loan_amount = mean(loan_amnt, na.rm = TRUE),
    count = n()
  )

head(grade_summary, 5)
```

```
## # A tibble: 5 x 4
##   grade avg_interest avg_loan_amount  count
##   <chr>        <dbl>           <dbl>  <int>
## 1 A             7.19          14057.  89430
## 2 B            10.6           13245. 199663
## 3 C            13.9           14641. 203226
## 4 D            17.5           15855. 102764
## 5 E            20.8           18285.  50370
```

```
## -------------------------------
## D. Yearly frequency distribution (2012-2017)
## -------------------------------

# Convert issue_d to date (first day of the month)
loan_states_regions <- loan_states_regions %>%
  mutate(issue_date = my(issue_d))


yearly_summary <- loan_states_regions %>%
  mutate(issue_year = year(issue_date)) %>%
  group_by(state, issue_year) %>%
  summarise(
    loans = n(),
    avg_amount = mean(loan_amnt, na.rm = TRUE),
    avg_interest = mean(int_rate, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```
head(yearly_summary, 5)
```

```
## # A tibble: 5 x 5
## # Groups:   state [1]
##   state issue_year loans avg_amount avg_interest
##   <chr>      <dbl> <int>      <dbl>        <dbl>
## 1 AK          2012    88     16343.         16.2
## 2 AK          2013   170     17032.         15.0
## 3 AK          2014   293     16792.         14.5
## 4 AK          2015   494     17403.         13.2
## 5 AK          2016   375     16279.         13.3
```

```
## -------------------------------
## E. Relationship between population & avg loan amount
## -------------------------------
pop_relationship <- loan_states_regions %>%
  group_by(state, population) %>%
  summarise(avg_loan = mean(loan_amnt, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```r
cor(pop_relationship$population, pop_relationship$avg_loan, use = "complete.obs")
```

```
## [1] 0.1454316
```

```r
head(pop_relationship, 5)
```

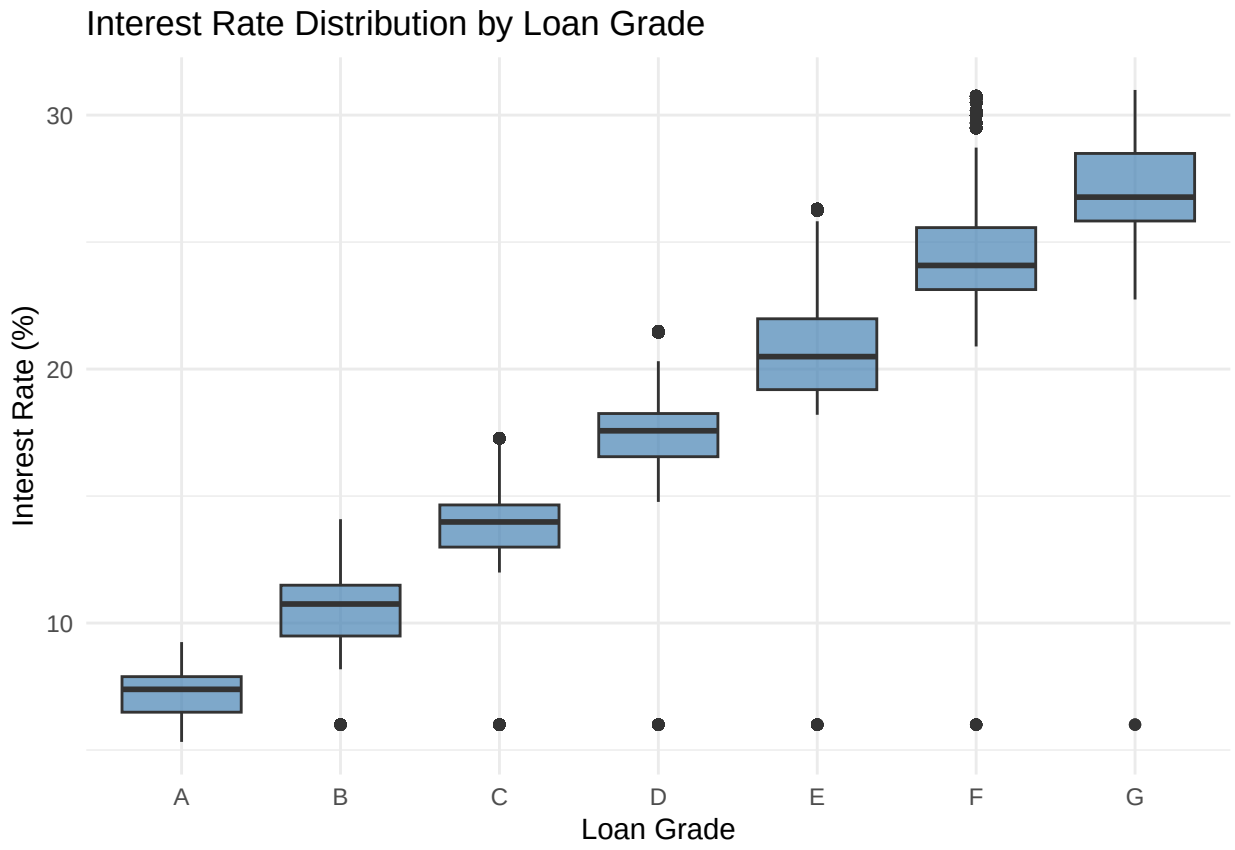
```
## # A tibble: 5 x 3
## # Groups:   state [5]
##   state population avg_loan
##   <chr>      <dbl>    <dbl>
## 1 AK        738565   16892.
## 2 AL       4850771   14316.
## 3 AR       2977944   13805.
## 4 AZ       6809946   14442.
## 5 CA      38982847   15070.
```

```r
## ------------------------------
## F. Grade vs median income
## ------------------------------

 income_grade_by_state <- loan_states_regions %>%
   group_by(state, grade) %>%
   summarise(
     num_loans = n(),
     .groups = "drop"
   ) %>%
   left_join(
     loan_states_regions %>%
       select(state, median_income_households) %>%
       distinct(),
     by = "state"
   ) %>%
   arrange(desc(median_income_households))

head(income_grade_by_state, 5)
```

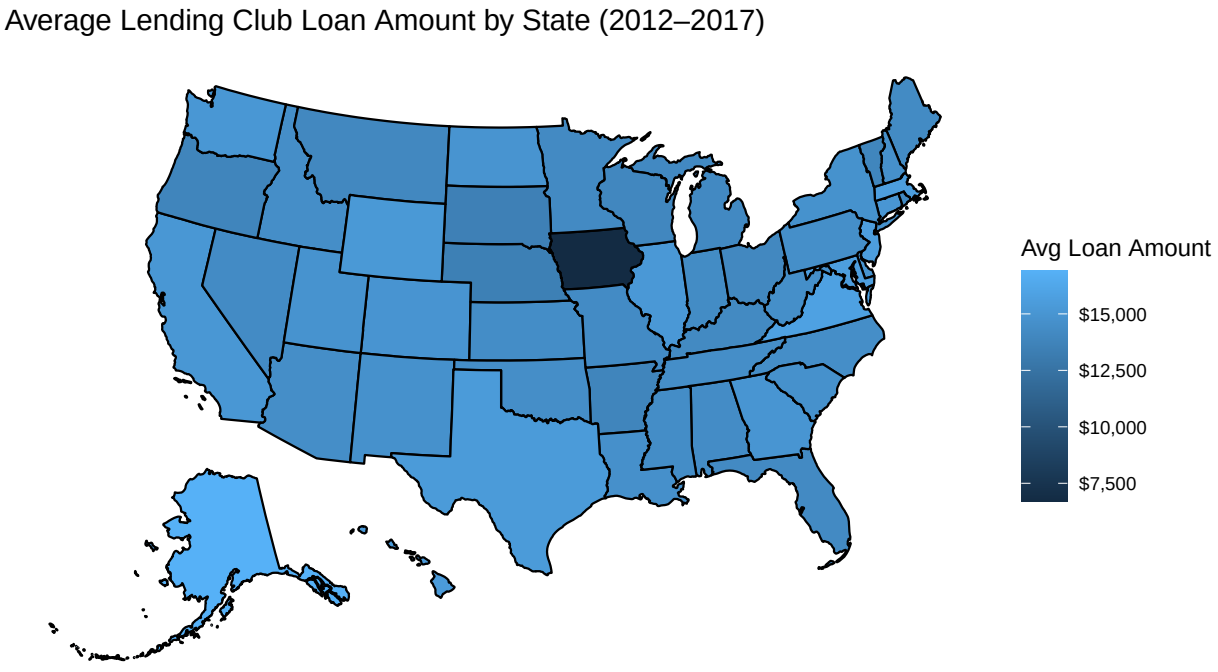
```
## # A tibble: 5 x 4
##   state grade num_loans median_income_households
##   <chr> <chr>     <int>                     <dbl>
## 1 MD    A          2184                     78916
## 2 MD    B          4971                     78916
## 3 MD    C          5152                     78916
## 4 MD    D          2654                     78916
## 5 MD    E          1422                     78916
```

```r
## ============================================================
## VISUALIZATION SECTION
## ============================================================

## ------------------------------
## 1. Plot: Interest Rates by Grade
```

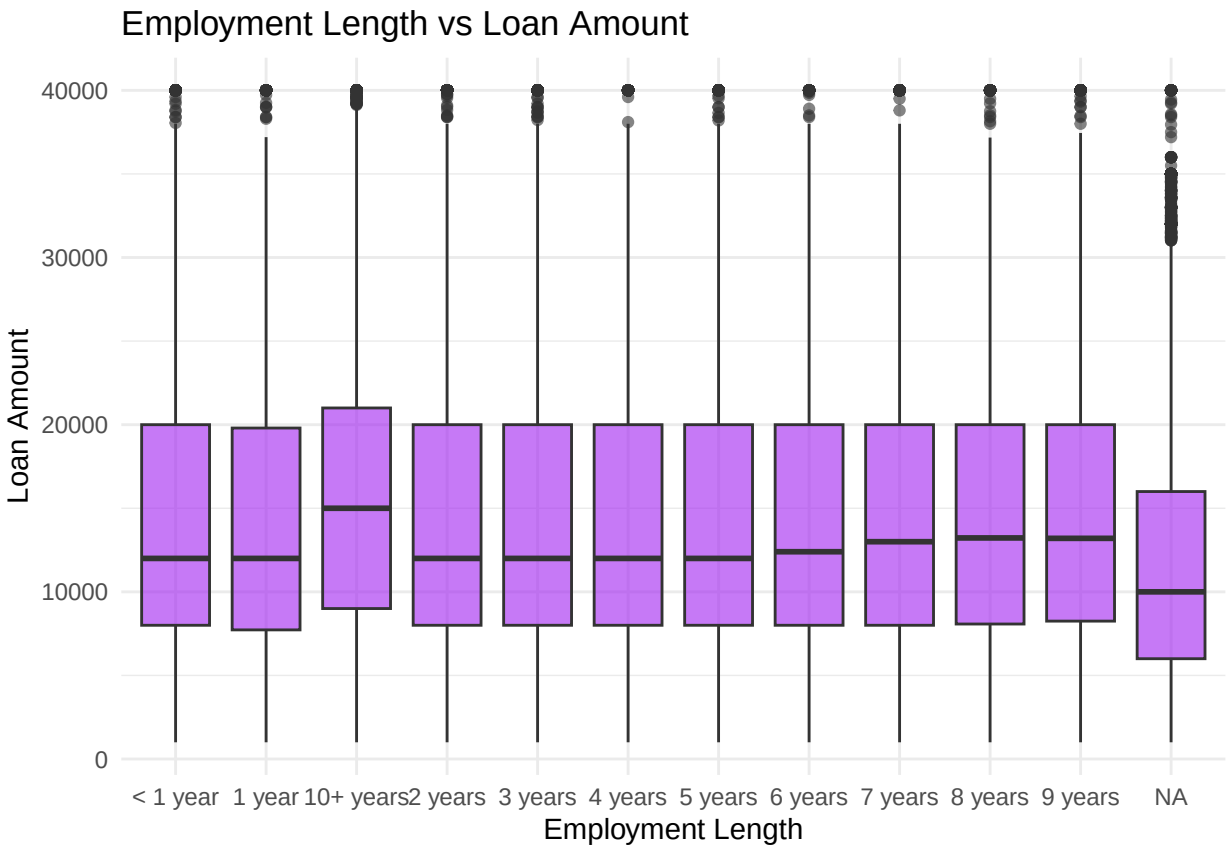
```r
## ------------------------------
ggplot(loan_states_regions, aes(x = grade, y = int_rate)) +
  geom_boxplot(fill = "steelblue", alpha = 0.7) +
  labs(
    title = "Interest Rate Distribution by Loan Grade",
    x = "Loan Grade",
    y = "Interest Rate (%)"
  ) +
  theme_minimal()
```




```r
## ------------------------------
## 2. U.S. Map - Average Loan Amount by State
## ------------------------------
state_avg_map <- loan_states_regions %>%
  group_by(state) %>%
  summarise(avg_loan = mean(loan_amnt, na.rm = TRUE))

plot_usmap(data = state_avg_map, values = "avg_loan", color = "black") +
  scale_fill_continuous(name = "Avg Loan Amount", label = dollar) +
  theme(legend.position = "right") +
  labs(title = "Average Lending Club Loan Amount by State (2012-2017)")
```



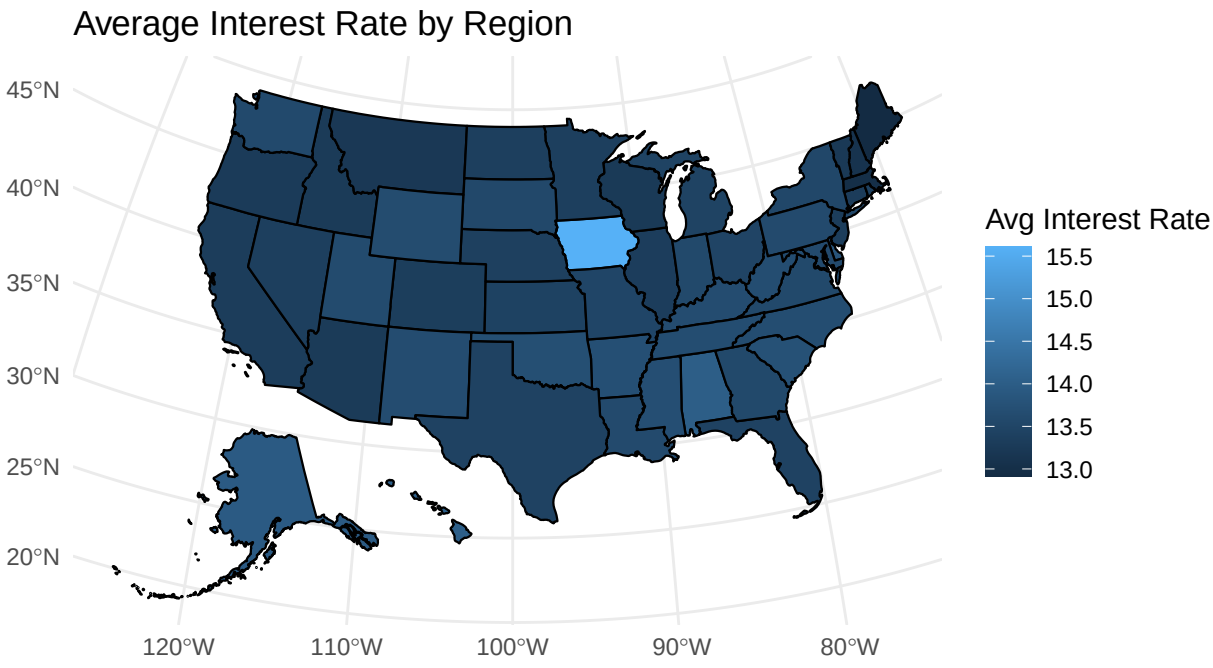

```
## ------------------------------
## 4. Employment Length vs Loan Amount
## ------------------------------
ggplot(loan_states_regions, aes(x = emp_length, y = loan_amnt)) +
  geom_boxplot(fill = "purple", alpha = 0.6) +
  labs(
    title = "Employment Length vs Loan Amount",
    x = "Employment Length",
    y = "Loan Amount"
  ) +
  theme_minimal()
```




```
## 5. Regional Map - Any Interesting Insight
## ------------------------------
region_map <- loan_states_regions %>%
  group_by(state, region) %>%
  summarise(avg_interest = mean(int_rate, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```
plot_usmap(data = region_map, values = "avg_interest", color = "black") +
  scale_fill_continuous(name = "Avg Interest Rate") +
  labs(title = "Average Interest Rate by Region") +
  theme_minimal()
```


Average Interest Rate by Region
45°N
40°N
35°N
30°N
25°N
20°N
120°W
110°W
100°W
90°W
80°W
Avg Interest Rate
15.5
15.0
14.5
14.0
13.5
13.0