Data Visualization

Huyen Nguyen

Data Description

1. Lending Club: The Lending Club data is LendingClub_LoanStats_2011_v2.csv. The data dictionary is in LCDataDictionary.xlsx. Lending Club is an online, peer-to-peer marketplace that connects borrowers and investors. We will conduct some exploratory analysis on a dataset of past Lending Club loans, including loan details and information about the borrowers. Of particular interest are variables loan_amnt, which represents the amount of money requested by a borrower, and funded_amnt, which represents the amount of money actually funded by investors. A full data dictionary can be found in LCDataDictionary.xlsx.

2. Dow Jones: The Dow Jones data is DowJones_history.csv. This contains the daily Open, Close, High and Low stock price for the Dow Jones industrial available index from late June through late Sept 2020. The data source is Yahoo!Finance.

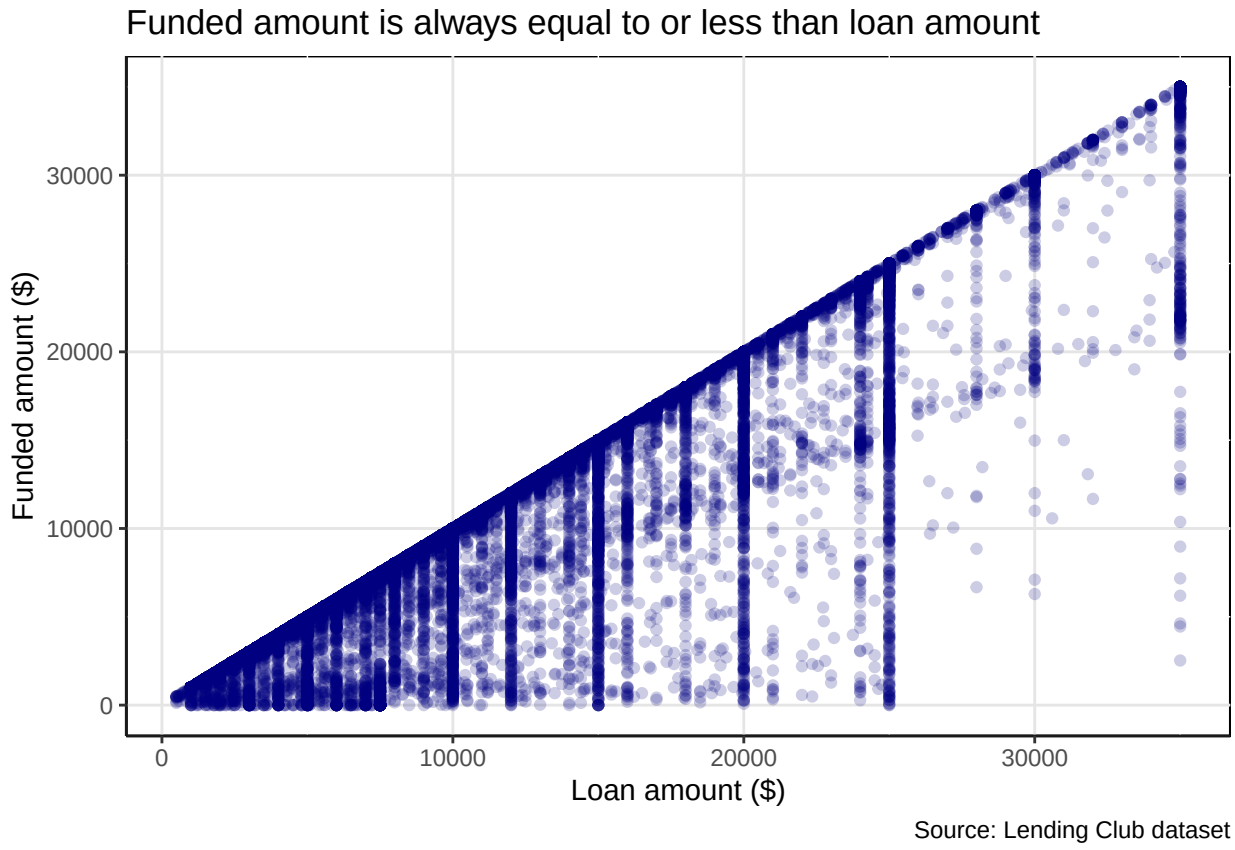
3. Thurman networks: Blake Thurman analyzed the employees at an office to understand how the network structure influenced power dynamics and success. There are two files: Thurman social networks and Thurman reporting structure. The social relationship data describes the informal relationship among the employees and the office data describes the reporting hierarchy. The paper is: In the office: Networks and coalitions.

Part 1: Scatter plots

a) Scatter plot Make a scatter plot relating the loan amount (the amount that each borrower requested) to the funded amount (the amount funded by investors).

```
  #CHART 1A CODE
lc %>%
# create a scatter plot, change color of data points to navyblue
#make each data point transparent by setting alpha < 1
  ggplot(aes(x=loan_amnt, y=funded_amnt_inv)) +
  geom_point(color="navyblue", alpha = 0.2) +

# add informative title, caption, x and y axis
  labs(title = "Funded amount is always equal to or less than loan amount",
       x = "Loan amount ($)", y = "Funded amount ($)", caption='Source: Lending Club dataset') +

#Change the background: change background color from grey to white
  theme(panel.background = element_rect(fill="white", colour = "black"),
        panel.grid.major = element_line(colour = "grey90"),#change gridline color to grey90
```

```
        #change color of axis lines to grey
        axis.line = element_line(color="grey10"),

        #put legend at the bottom of the chart
        legend.position = "bottom") +

        #provide sequential, diverging and qualitative colour schemes from ColorBrewer
        scale_color_brewer(palette="Dark2")
```

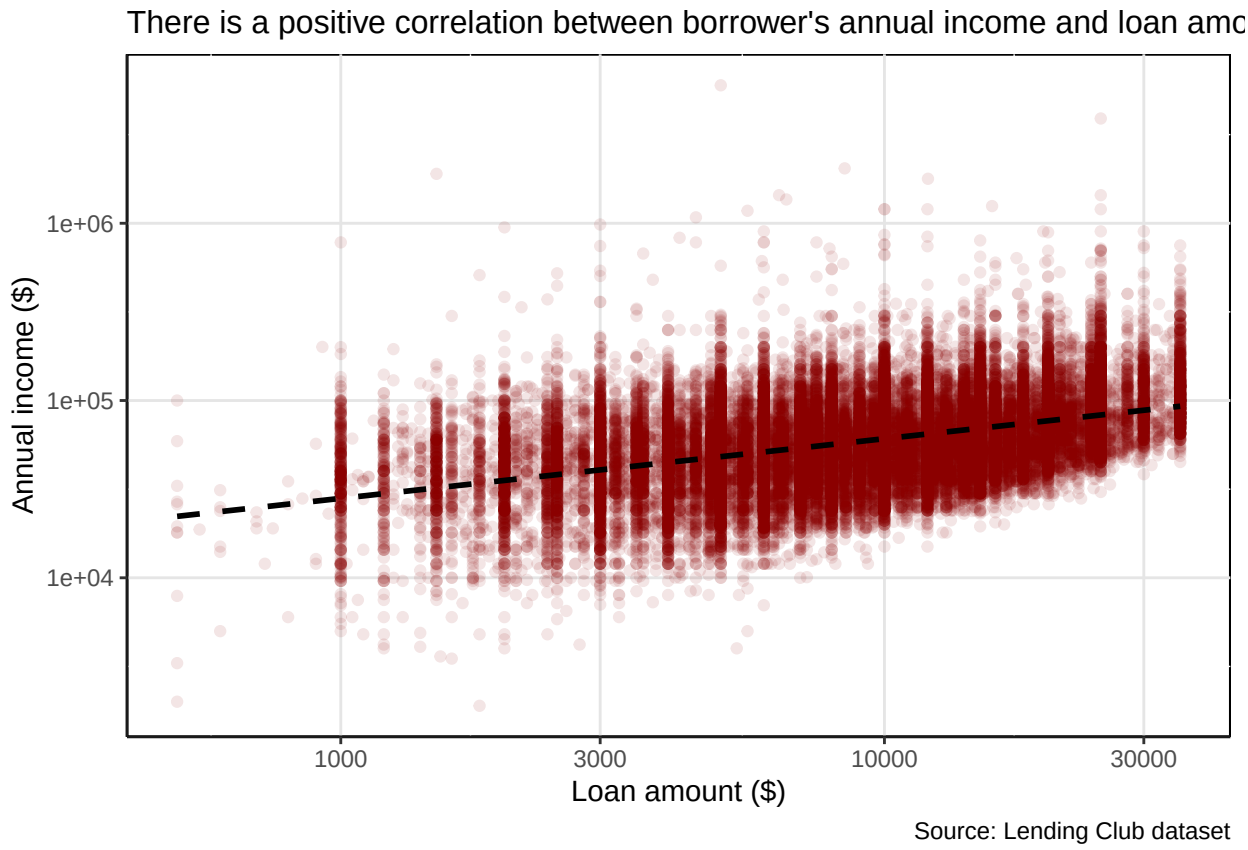


b) Make a plot exploring the relationship between annual income (the borrower's income) and loan amount (the amount that the borrower requests). Use the logarithmic scale.

```
#CHART 1B CODE
lc %>%
  # create a scatter plot, change color of data points to darkred
  ggplot(aes(x=loan_amnt, y=annual_inc)) +
  geom_point(color="darkred", alpha = 0.1) +

  #Use the logarithmic scale for both x and y axes
  scale_x_continuous(trans = 'log10') + scale_y_continuous(trans = 'log10')+
  labs(title = "There is a positive correlation between borrower's annual income and loan amount",
  x = "Loan amount ($)", y = "Annual income ($)", caption='Source: Lending Club dataset') +
  theme(panel.background = element_rect(fill="white", colour = "black"),
        panel.grid.major = element_line(colour = "grey90"),
        axis.line = element_line(color="grey10"),
```

```
        plot.title = element_text(size=12),
        legend.position = "bottom") +
        scale_color_brewer(palette="Dark2") +

  #add a trendline; 'lm': using smoothing method as linear regression,
  #se: standard error (display confidence interval around smooth?)
  geom_smooth(method="lm", color="black", linetype=2, se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

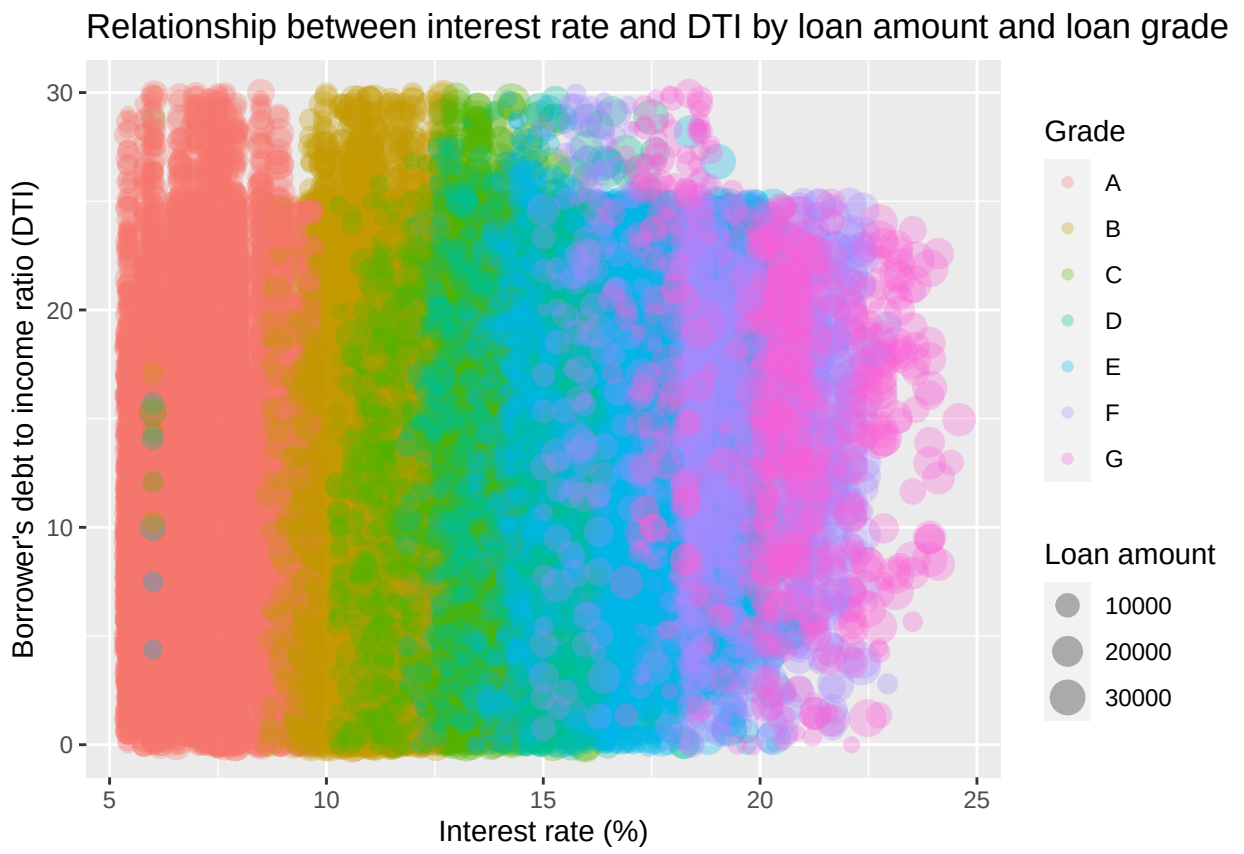


c) Bubble Chart and Colors

Make a plot relating interest rate to the borrower's debt to income ratio (dti). Incorporate the loan amount and loan grade.

```
#CHART 1C CODE
#preprocess data
lc$int_rate <- str_replace(lc$int_rate, "%", "") #remove percent side from each value
lc$int_rate_new <- as.numeric(lc$int_rate) #convert string to numeric
plot_data <- lc %>% drop_na(grade)  #drop NA

#make a plot
plot_data %>%
```

```
#use color to denote loan grade, and size of data points to denote loan amount.
ggplot(aes(x=int_rate_new, y=dti, color=grade, size=loan_amnt)) +
geom_point(alpha = 0.3) +
labs(title = "Relationship between interest rate and DTI by loan amount and loan grade",
     x = "Interest rate (%)", y = "Borrower's debt to income ratio (DTI)",
     size="Loan amount", color="Grade")
```

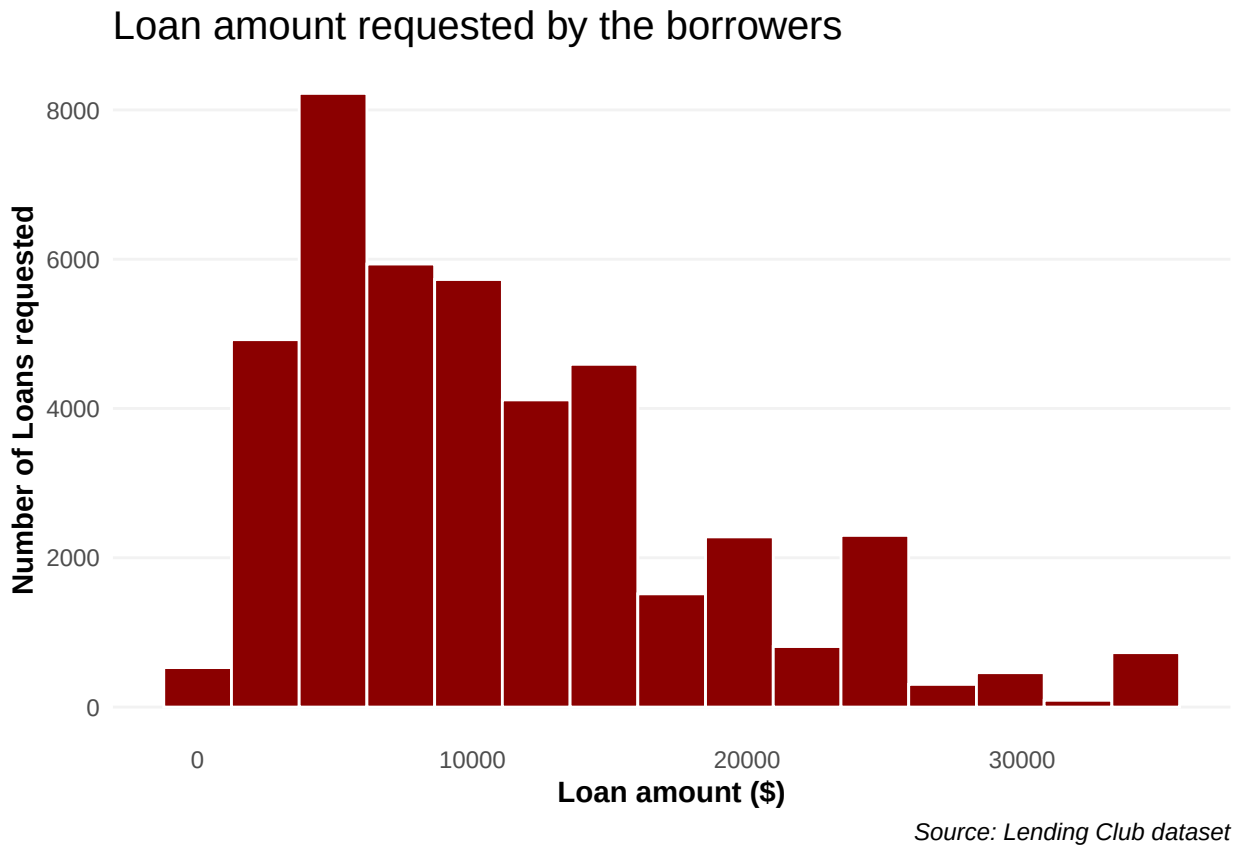


Part 2: Distributions

a) Use a histogram to show the distribution of the Loan Amount (the amount requested by the borrower). Make sure to change the border color of the bars and change the x and y-axis labels.

```
#CHART 2A CODE
lc %>% ggplot(aes(x=loan_amnt)) +

  #create a histogram with 15 bins
  geom_histogram(bins=15, fill="darkred", color="white") +
  labs(title="Loan amount requested by the borrowers",
       caption = "Source: Lending Club dataset",
       x="Loan amount ($)", y="Number of Loans requested") +

  theme_bw() +
  theme(legend.position = "bottom",
        axis.ticks = element_blank(),
```

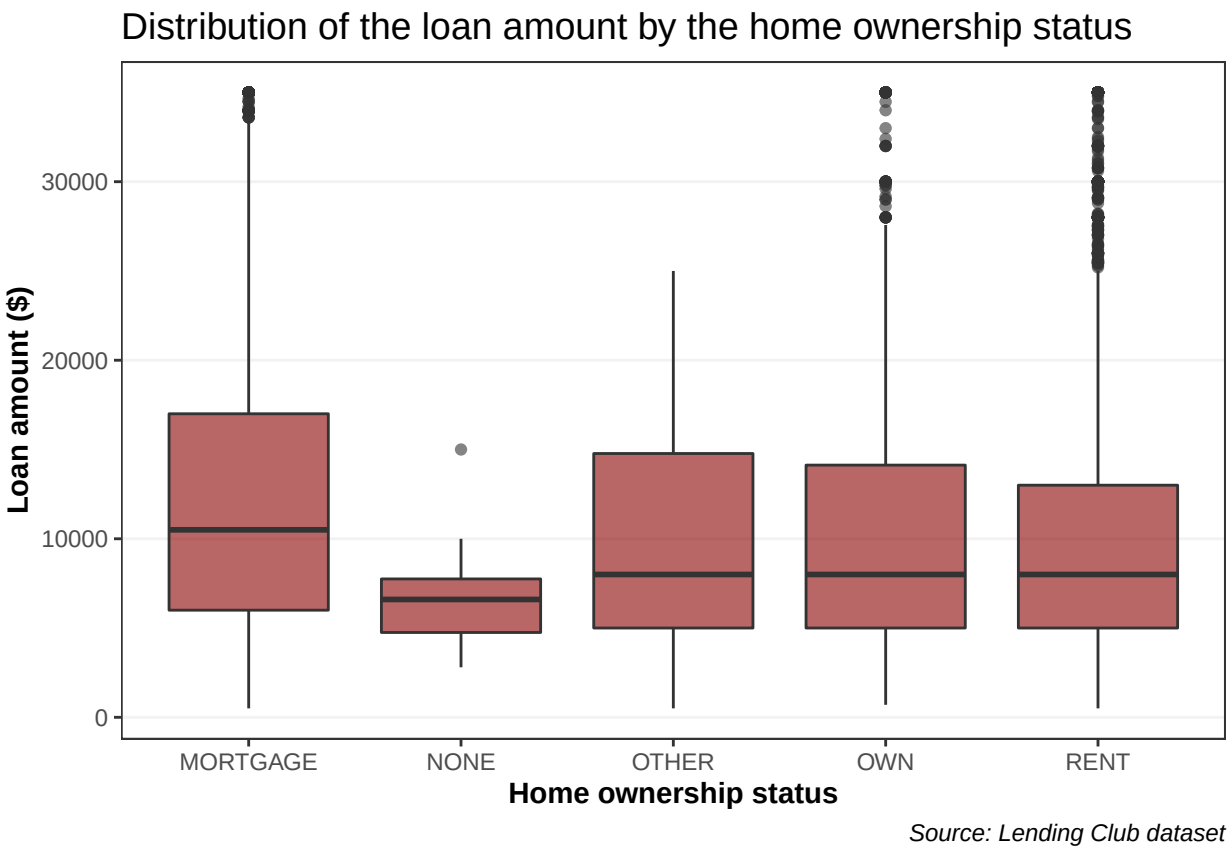
```
        plot.title = element_text(size=15),
        panel.border = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.y = element_line(color="grey95"),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_blank(),
        axis.title = element_text(face="bold"),
        plot.caption = element_text(face="italic"))
```



b) Compare the distribution of the loan amount by the homeownership status by creating a chart that shows some distribution information for each level of home ownership. In particular, make sure that the chart describes the median loan amount for each home ownership status.

```
#CHART 2B CODE
plot_data2 <- lc %>% drop_na(home_ownership)
plot_data2 %>%

  #create a boxplot with home ownership on the x axis and loan amount on the y axis
  ggplot(aes(x=home_ownership, y=loan_amnt)) +
  geom_boxplot(fill="darkred", alpha=0.6) +
  theme_bw() +
  theme(legend.position = "bottom",
        plot.title = element_text(size=14),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.y = element_line(color="grey95"),
```

```
        panel.grid.minor.x =element_blank(),
        panel.grid.major.x = element_blank(),
        axis.title = element_text(face="bold"),
        plot.caption = element_text(face="italic")) +
  labs(title="Distribution of the loan amount by the home ownership status",
       caption = "Source: Lending Club dataset",
       x="Home ownership status", y="Loan amount ($)")
```

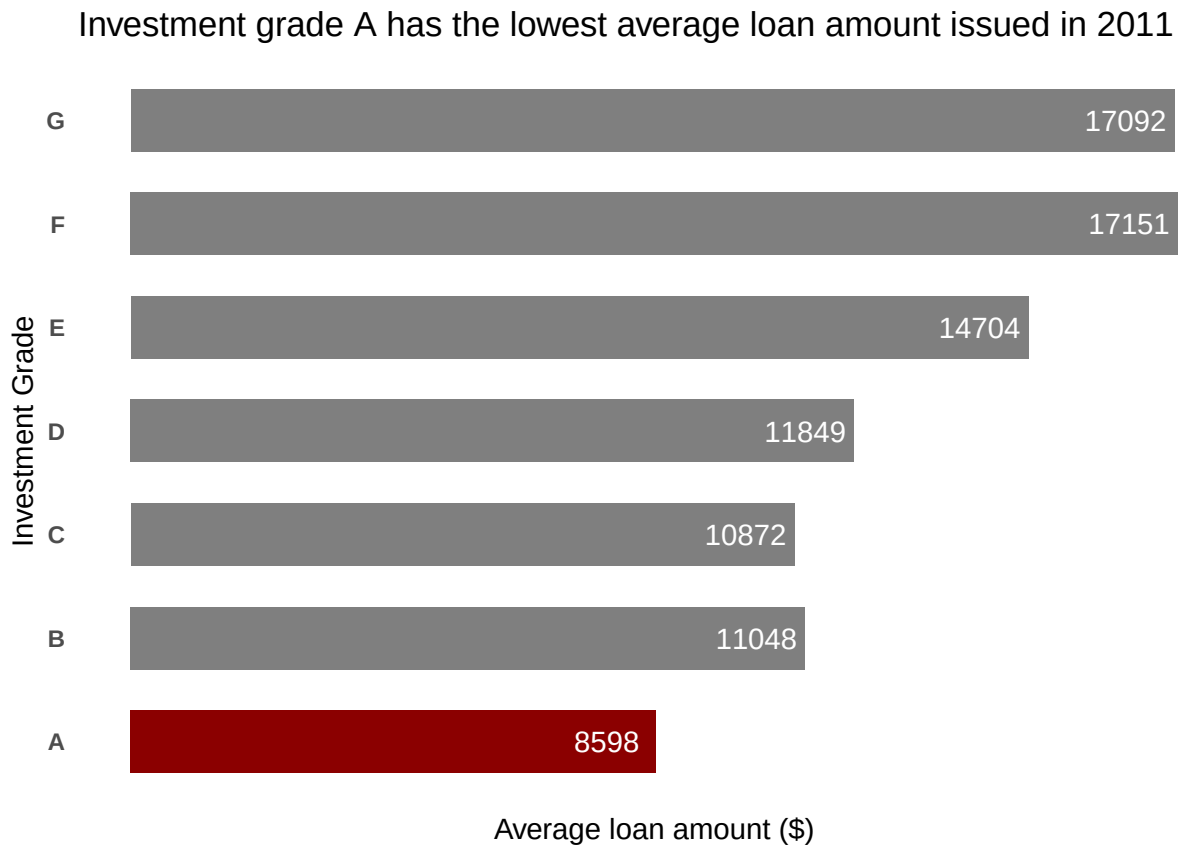


Part 3: Categorical plot

a) Create a bar chart with the average Loan Amount by Investment Grade for 2011. Ensure that readers can easily identify the Loan Grade with the lowest average Loan Amount.

```
#CHART 3A CODE

#preprocess data

#get only data from 2011, grouped by loan grade, and
#calculate the average of loan amount for each loan grade.
plot_data3 <- lc %>%filter(issue_year==2011)  %>%
        group_by(grade) %>%
        summarize(mean_loans = mean(loan_amnt, na.rm=TRUE))

#make a plot
```

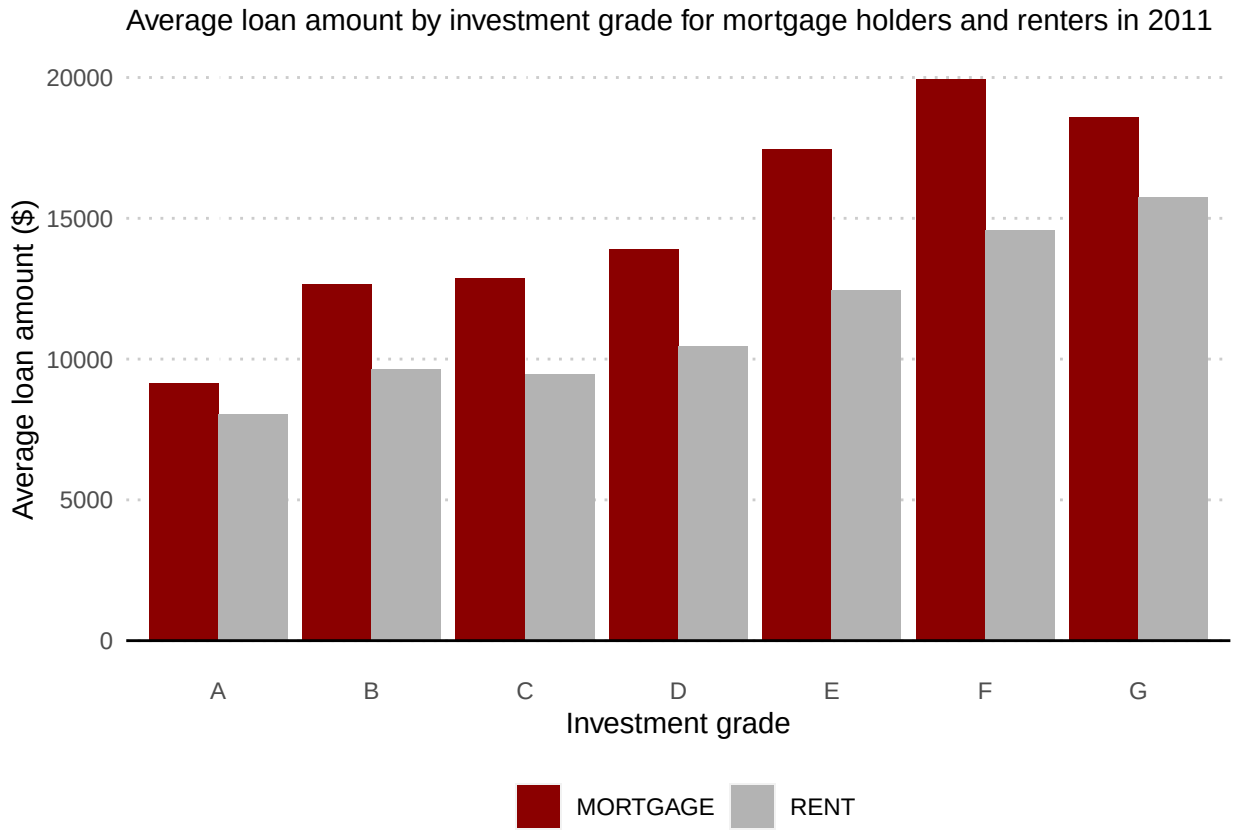
```r
plot_data3 %>% na.omit() %>%
  ggplot(aes(x=grade, y=mean_loans)) +
  geom_col(aes(fill=ifelse(mean_loans<10000,"Y","N")), width=0.6)+
  scale_fill_manual(name="",breaks=c("Y","N"),values=c("darkred","grey50"))+
  coord_flip() +
  labs(x="Investment Grade", y="Average loan amount ($)",
       title = "Investment grade A has the lowest average loan amount issued in 2011") +
  theme(panel.background = element_blank(),
        legend.title = element_blank(),
        legend.position = "none",
        axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        axis.text.y = element_text(face="bold", hjust=1)) +
        geom_text(aes(y=mean_loans-800,
                label=round(mean_loans, digits = 0)),
            color="white")
```



b) Show the average loan amount by investment grade for mortgage holders vs. renters, in 2011.

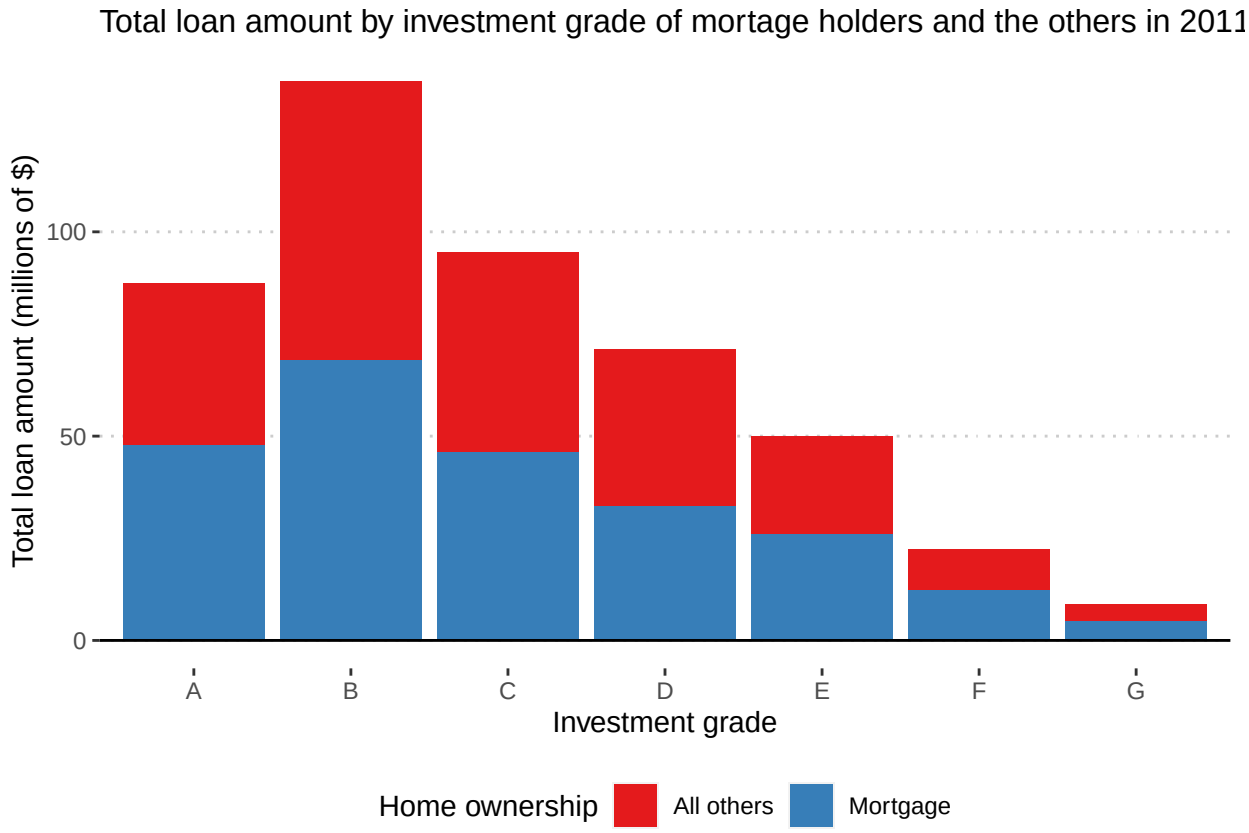
```r
#CHART 3B CODE

#preprocess data

#get only data from 2011 and for mortgage holders vs. renters,
#grouped by loan grade and home ownership,
```

```r
#and calculate the average of loan amount for each loan grade and home ownership.
plot_data4 <- lc %>%filter(issue_year==2011,
      home_ownership=='MORTGAGE'|home_ownership=='RENT')  %>%
      group_by(grade, home_ownership) %>%
      summarize(mean_loans = mean(loan_amnt, na.rm=TRUE))

#make a plot

#create a cluster chart with loan grade on the x axis and loan average on the y axis
#for mortgage holders vs. renters (home ownership)
plot_data4 %>%
  ggplot(aes(x=grade, y=mean_loans, group=home_ownership)) +
  geom_col(aes(fill=home_ownership), position="dodge") +
  labs(y="Average loan amount ($)", x="Investment grade",
title="Average loan amount by investment grade for mortgage holders and renters in 2011") +

  #manually fill two types of ownership with custom colors
  scale_fill_manual(name="",breaks=c("MORTGAGE","RENT"),
                    values=c("darkred", "grey70")) +
  geom_hline(yintercept = 0) +
  theme(panel.background = element_blank(),
        plot.title = element_text(size=11),
        legend.position = "bottom",
        axis.ticks = element_blank(),
        panel.grid.major.y = element_line(color="grey80", linetype = 3))
```



c) Show the total loan amount by investment grade in 2011. Break each grade into mortgage holders vs. all others.

```
#CHART 3C CODE

#preprocess data
plot_data5 <- lc %>%
  filter(issue_year==2011) %>%
mutate(`Home ownership`= ifelse(home_ownership == "MORTGAGE", "Mortgage","All others")) %>%
  group_by(grade, `Home ownership`) %>%
  summarize(total_loans = sum(loan_amnt, na.rm=TRUE)/1000000)

#make a plot
plot_data5 %>% na.omit() %>%

  #create a stacked bar showing the total loan amount by investment grade,
  #break each grade into mortgage holders vs. all others.
  ggplot(aes(x=grade, y=total_loans, group=`Home ownership`)) +
  geom_col(aes(fill=`Home ownership`)) +
  scale_fill_brewer(palette = "Set1") +
    labs(y="Total loan amount (millions of $)", x="Investment grade",
       title="Total loan amount by investment grade of mortage holders and the others in 2011") +
  geom_hline(yintercept = 0) +
  theme(panel.background = element_blank(),
        plot.title = element_text(size=12),
        legend.position = "bottom",
        panel.grid.major.y = element_line(color="grey80", linetype = 3))
```



Part 4: Pyramid and square area chart

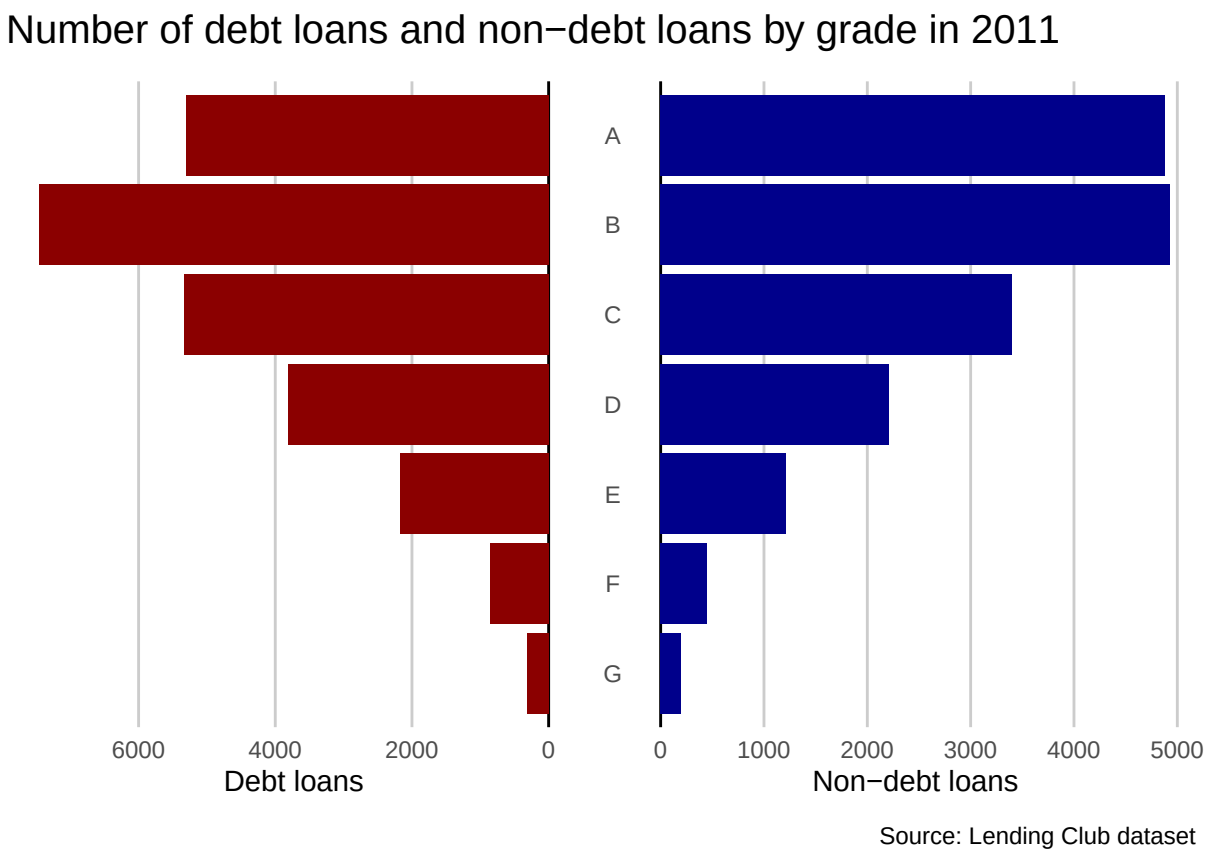
a.Pyramid Chart

Use patchwork to create a pyramid chart. Place a label for the Loan Grade in the center, and display the number of loans, organized by the purpose of the debt (group debt consolidation and credit cards together versus all other loan purposes).

Create one chart with the number of loans for debt (debt consolidation and credit cards), and create a second chart with the number of non-debt loans (all other loan purposes). Remember to change the levels of the Loan Grade factor so that A is at the top of the chart. Use patchwork to combine the two charts.

```
#preprocess data
data1 <- lc %>%

  #create new column called loan_purpose to categorize loans into debt loans
  #and non-debt loans
  mutate(loan_purpose= ifelse(
    purpose == "debt_consolidation"|purpose == "credit_card",
    "debt_loan","nondebt_loan")) %>%

  #calculate the total number of loans for each loan type
  select(grade, loan_purpose) %>% group_by(grade, loan_purpose) %>%
  summarise(num=n()) %>%
  na.omit() %>%
```

```r
  #create num_sign column to change the sign of number of debt-loans into negative numbers.
  mutate(num_sign= ifelse(loan_purpose == 'debt_loan', -num, num))

#create the first bar chart for non-debt loans
nondebt_chart <- data1 %>%
  filter(loan_purpose=='nondebt_loan') %>%
  ggplot(aes(y=grade, x=num_sign)) +
  geom_vline(xintercept = 0) +
  geom_col(fill="darkblue") +
  labs(x="Non-debt loans") +
  theme(axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major.x = element_line(color="grey80"),
        panel.background = element_rect(fill="white"),
        axis.text.y = element_text(hjust=0.5)) +

  #sort the loan grades on the y axis to make sure grade A is on the top of the chart.
  scale_y_discrete(limits=c('G', 'F', 'E', 'D', 'C', 'B', 'A'))

#create the second bar chart for debt loans
debt_chart <- data1 %>%
  filter(loan_purpose=='debt_loan') %>%
  ggplot(aes(y=grade, x=num_sign)) +
  geom_vline(xintercept = 0) +
  geom_col(fill="darkred") +
  labs(x="Debt loans") +
  theme(axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major.x = element_line(color="grey80"),
        panel.background = element_rect(fill="white")) +

  #change the scale of x axis to positive numbers
  scale_x_continuous(breaks=c(0,-2000,-4000,-6000),
                     labels = c(0,2000,4000,6000)) +
  scale_y_discrete(limits=c('G', 'F', 'E', 'D', 'C', 'B', 'A'))

#combine two charts using patchwork to create a pyramid chart
debt_chart + nondebt_chart +
  plot_annotation(
    title = 'Number of debt loans and non-debt loans by grade in 2011',
    caption = 'Source: Lending Club dataset',
    theme = theme(plot.title = element_text(size = 15))
  )
```

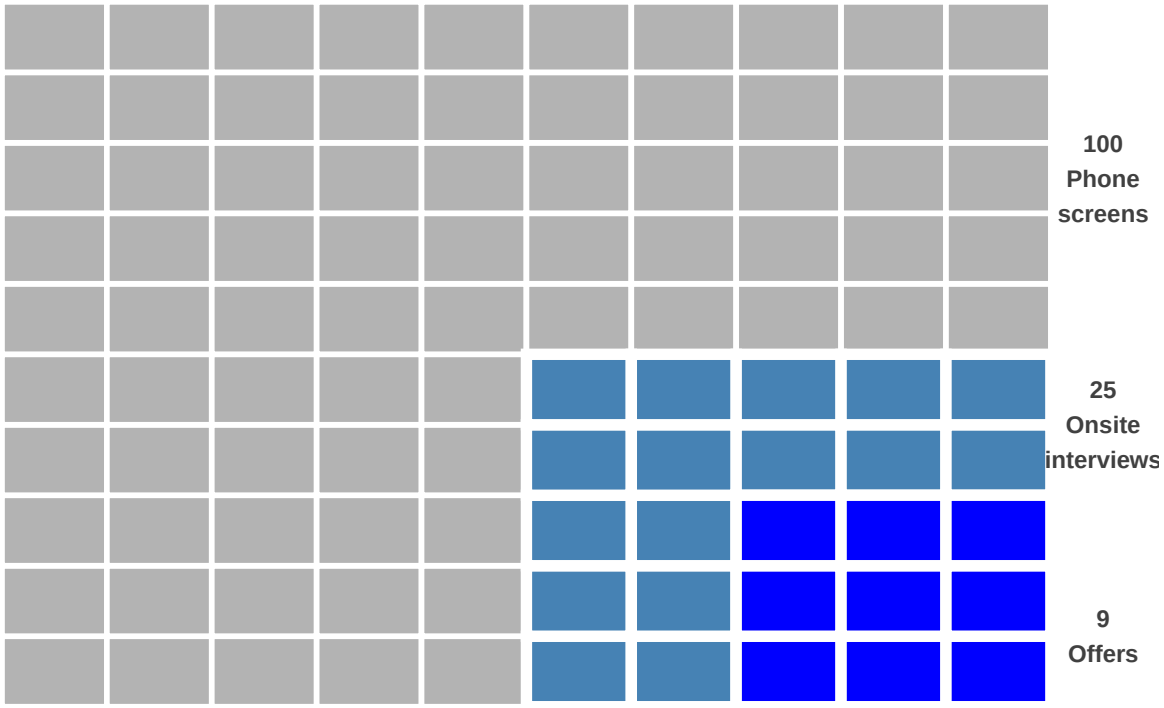


4b. Square Area Chart

Make a square area chart that shows that "Out of every 100 phone screens, we bring 25 candidates onsite, and extend 9 offers.

```
# make the grid of 100 squares (10 x 10 grid)
x_dim=10
y_dim=10
x = as.vector(sapply(1:x_dim, FUN=function(x) {rep(x,y_dim)}))
y = rep(1:y_dim, x_dim)

# 25 candidates onsite were chosen out of 100 phone screens
# make the small grid of 5 x 5
color_x_dim=5
color_y_dim=5
color_x = as.vector(sapply(seq(x_dim-color_x_dim+1,x_dim,1),
                           FUN=function(x) {rep(x,color_x_dim)}))
color_y = rep(1:color_y_dim, color_y_dim)

# Only 9 candidates receiving offers
# make the small grid of 3 x 3
color2_x_dim=3
color2_y_dim=3
color2_x = as.vector(sapply(seq(x_dim-color2_x_dim+1,x_dim,1),
                           FUN=function(x) {rep(x,color2_x_dim)}))
color2_y = rep(1:color2_y_dim, color2_y_dim)
```

```r
large_grid_data <- tibble(x=x, y=y)
color_data <- tibble(x=color_x, y=color_y)
color2_data <- tibble(x=color2_x, y=color2_y)

#make a plot
q2 <- ggplot() +
  geom_tile(data=large_grid_data, #create square area chart
            aes(x=x, y=y), fill="grey70",
            color="white", size=1) +
  geom_tile(data=color_data,
            aes(x=x, y=y), fill="steelblue", color="white",size=2) +
  geom_tile(data=color2_data,
            aes(x=x, y=y), fill="blue", color="white",size=2) +
  theme(panel.background = element_blank(),
        plot.caption = element_text(size=15, face="bold", hjust=0.5,
                                    color="steelblue"),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank()) +
  labs(caption = "Interview outcome", size=5) +
    geom_text(aes(x=11, y=8, label="100\nPhone\nscreens"),
            fontface="bold", colour="grey25", size=3.2) +
    geom_text(aes(x=11, y=4.5, label="25\nOnsite\ninterviews"),
            fontface="bold", colour="grey25", size=3.2) +
    geom_text(aes(x=11, y=1.5, label="9\nOffers"),
            fontface="bold", colour="grey25", size=3.2)
q2
```



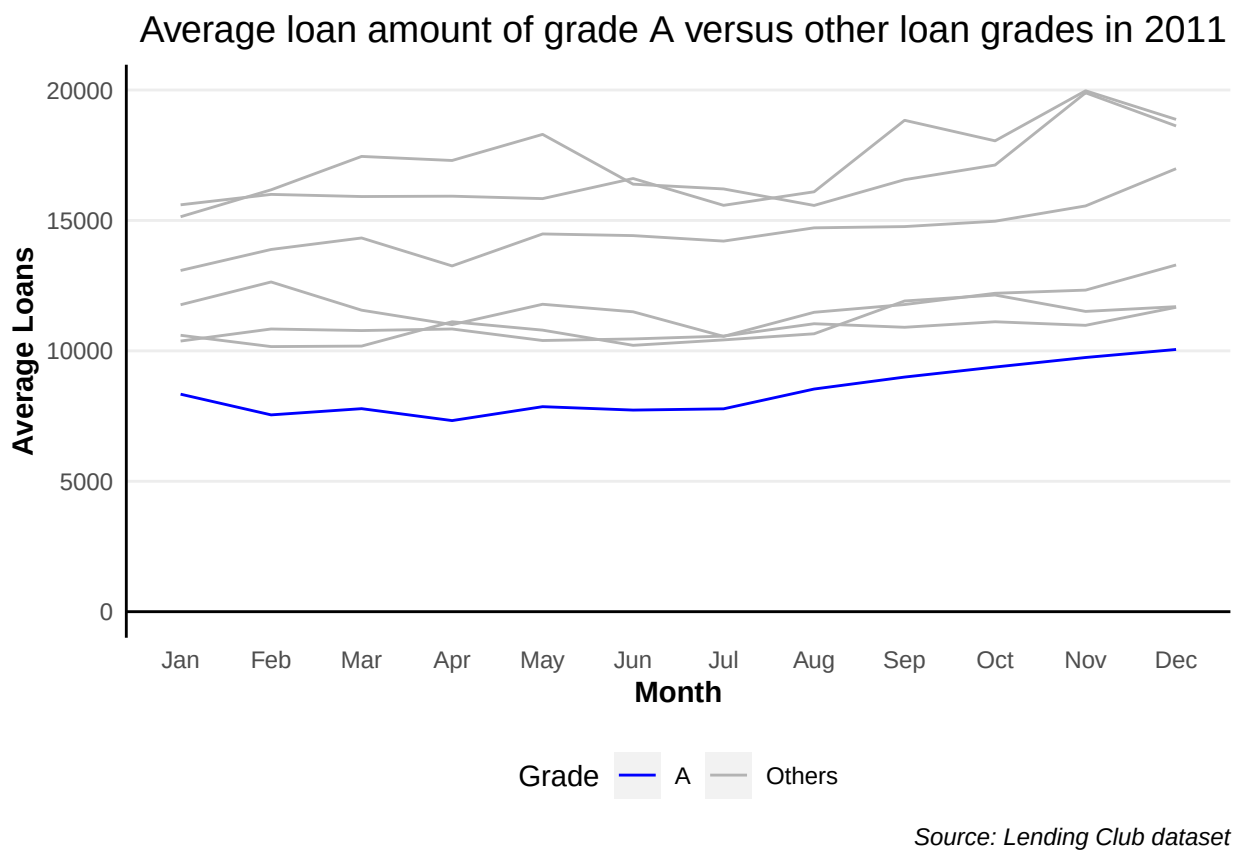
Part 5: Time Series (20 points)

5a. Multiple Line Chart

Create a line chart of the average loan amount in each of the loan grades over time. Highlight loan grade A and keep the other shades light grey.

```
#preprocess data
data2a <- lc %>%
  separate(issue_d, c('date', 'month'), sep='-')
  data2a <- data2a %>%
  group_by(grade, month) %>%
  summarize(mean_loan = mean(loan_amnt)) %>%

  #sort months manually from Jan to Dec, not by alphabetical order (by default).
  mutate(monthf = factor(month,
                         levels=c("Jan","Feb","Mar",
                                  "Apr","May","Jun",
                                  "Jul","Aug","Sep",
                                  "Oct","Nov","Dec"),
                         ordered=TRUE))

#separate loan grade A from others
data2a <- data2a %>%
  mutate(two_grades= ifelse(grade == "A", "A","Others"))
```

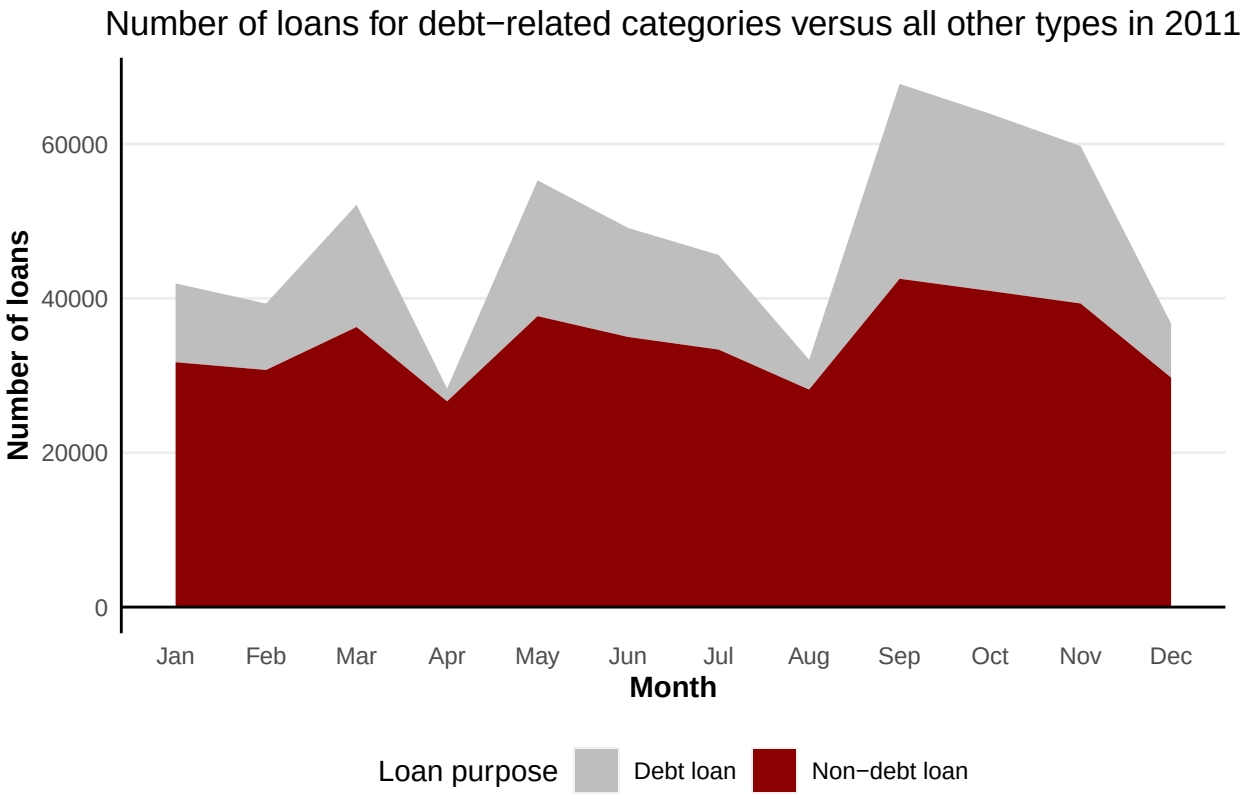
```r
#make a plot
data2a %>% na.omit() %>%
 ggplot(aes(x=monthf, y=mean_loan, group=grade)) +
 geom_line(aes(color=as.factor(two_grades))) +  #create line chart
labs(x = "Month", y="Average Loans",
     title = " Average loan amount of grade A versus other loan grades in 2011",
     caption="Source: Lending Club dataset", color = "Grade") +
  geom_hline(yintercept = 0) +
  theme(axis.ticks = element_blank(),
        legend.position = "bottom",
        plot.title = element_text(size=14, hjust=0.5),
        panel.border = element_blank(),
        panel.background = element_rect(fill="white"),
        panel.grid.minor.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line(color="grey93"),
        panel.grid.major.x = element_blank(),
        axis.title = element_text(face="bold"),
        axis.line.y = element_line(color="black"),
        plot.caption = element_text(face="italic")) +

  #highlight loan grade A and keep the other shades light grey.
   scale_color_manual(breaks = c("A", 'Others'),
                      values = c("blue", "grey70"))
```



5b. Area chart

Create an area chart showing the total number of loans for debt-related categories (credit cards and debt consolidation), versus all other types over time.

```
data2b <- lc %>%
  separate(issue_d, c('date', 'month'), sep='-') %>%
  mutate(loan_purpose= ifelse(
    purpose == "debt_consolidation"|purpose == "credit_card", "Debt loan","Non-debt loan"))

data2b <-data2b %>%
  group_by(loan_purpose, month) %>%
  summarise(num=n()) %>%

  #sort months manually from Jan to Dec, not by alphabetical order (by default).
  mutate(monthf = factor(month,
                         levels=c("Jan","Feb","Mar",
                                  "Apr","May","Jun",
                                  "Jul","Aug","Sep",
                                  "Oct","Nov","Dec"),
                         ordered=TRUE))

data2b %>% na.omit() %>%
  ggplot(aes(x=monthf, y=cumsum(num), group=loan_purpose)) +
  geom_area(aes(fill=loan_purpose)) +  #create area chart
  labs(x = "Month", y="Number of loans",
       title='Number of loans for debt-related categories versus all other types in 2011',
       caption="Source: Lending Club dataset") +
    geom_hline(yintercept = 0) +
  theme(legend.position = "bottom",
          axis.ticks = element_blank(),
          plot.title = element_text(size=13, hjust=0.5),
          panel.border = element_blank(),
          panel.background = element_rect(fill="white"),
          panel.grid.minor.y = element_blank(),
          panel.grid.minor.x = element_blank(),
          panel.grid.major.y = element_line(color="grey93"),
          panel.grid.major.x = element_blank(),
          axis.title = element_text(face="bold"),
          axis.line.y = element_line(color="black"),
          plot.caption = element_text(face="italic")) +
    scale_fill_manual(labels = c("Debt loan", "Non-debt loan"),
                      values = c('gray', 'darkred')) +
  guides(fill=guide_legend(title='Loan purpose'))
```



Source: Lending Club dataset

2c. Waterfall Chart

Create the waterfall chart for the entire DJI data. Include a stacked bar for the afterhours trading.

Data Dictionary: *Date*: The trading date (Monday through Friday) *Open*: The stock price when the market opens *Close*: The stock price when the market closes *High*: The highest stock price achieved during the day *Low*: The lowest stock price recorded during the day

```
#preprocess data
#make the data set: compute whether the price is up or down at the end of the day
data2c.tmp1 <- dji %>%
  select(Date, Open, Close) %>%
  mutate(Direction = ifelse(Close-Open>=0, "Up", "Down"),
         NextDate = lead(Date),
         NextOpen = lead(Open),
         AfterHours = 'N')

#create data to show afterhour trading
data2c.tmp2 <- data2c.tmp1 %>%
  select(Date, Open=Close, Close=NextOpen, NextDate) %>%
  mutate(Direction = ifelse(Close-Open>=0, "Up", "Down"),
         AfterHours = 'Y', NextOpen=Close)


data2c.tmp = rbind(data2c.tmp1, data2c.tmp2)
```

```
#create waterfall chart
waterfall <- ggplot(data2c.tmp,
                    aes(x = Date, fill = Direction, linetype=AfterHours,
                        colour=AfterHours, group=AfterHours)) +
  geom_rect(aes(xmin = Date - 0.4, # control bar gap width
                xmax = Date + 0.4,
                ymin = Open,
                ymax = Close),
            alpha=0.95) +
#add dotted lines between bars
  geom_segment(data = data2c.tmp[1:(nrow(data2c.tmp) -1),],aes(x = Date + 0.25,
                                                               xend = NextDate - 0.25,
                                                               y = Close,
                                                               yend = NextOpen),
                                                                linetype=3) +
  theme(panel.background = element_rect(fill="white"),
        legend.position = "bottom",
        panel.grid.major.y = element_line(color="grey80", linetype=2),
        panel.grid.minor.y = element_line(color="grey90", linetype=3)) +
  labs(x="Time", y='Price', title='Dow Jones stock prices Quarter 3, 2020') +
  scale_fill_manual(breaks = c("Down", "Up"),
                    values = c('red4', 'green')) +
  scale_y_continuous(breaks=seq(25000,29000,by=500))

waterfall
```

Part 6: Networks (10 points)

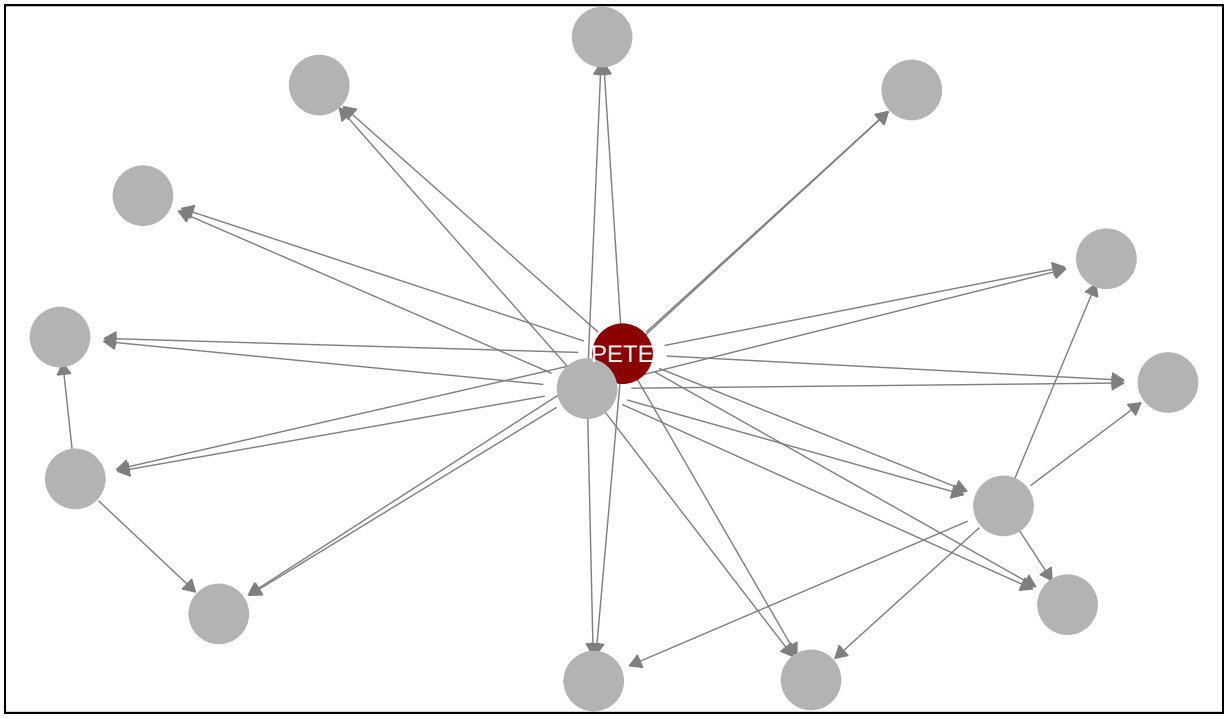
6a. Reporting Network

Use ggnet to create the hierarchy reporting network in the Thurman office data. Label the most central person in the advice network.

```
orgnet <- network(thurman.org, directed=T)

# calculate the degree
org <- sna::degree(orgnet)
# Label the person with the highest degree
org_label <- network.vertex.names(orgnet)[which.max(org)]

#color the node of people who are not the person with the highest degree
degree_color <- rep("grey70",length(org))

#color the node of the person with the highest degree
degree_color[which.max(org)] <- "darkred"

ggnet2(orgnet,
       label=org_label,
       size=10,
       node.color = degree_color,
       label.color='white',
       label.size=3,
       arrow.size = 5,
       arrow.gap = 0.04) +
  labs(title="Reporting network",
       subtitle = "Pete is the most central person in this reporting network.",
       caption = "Source: In the office: Networks and coalitions.") +
  theme(panel.border = element_rect(fill=NA,color="black"),
        plot.caption = element_text(face = "italic"))
```

Reporting network

Pete is the most central person in this reporting network.



Source: In the office: Networks and coalitions.

3b. Social network

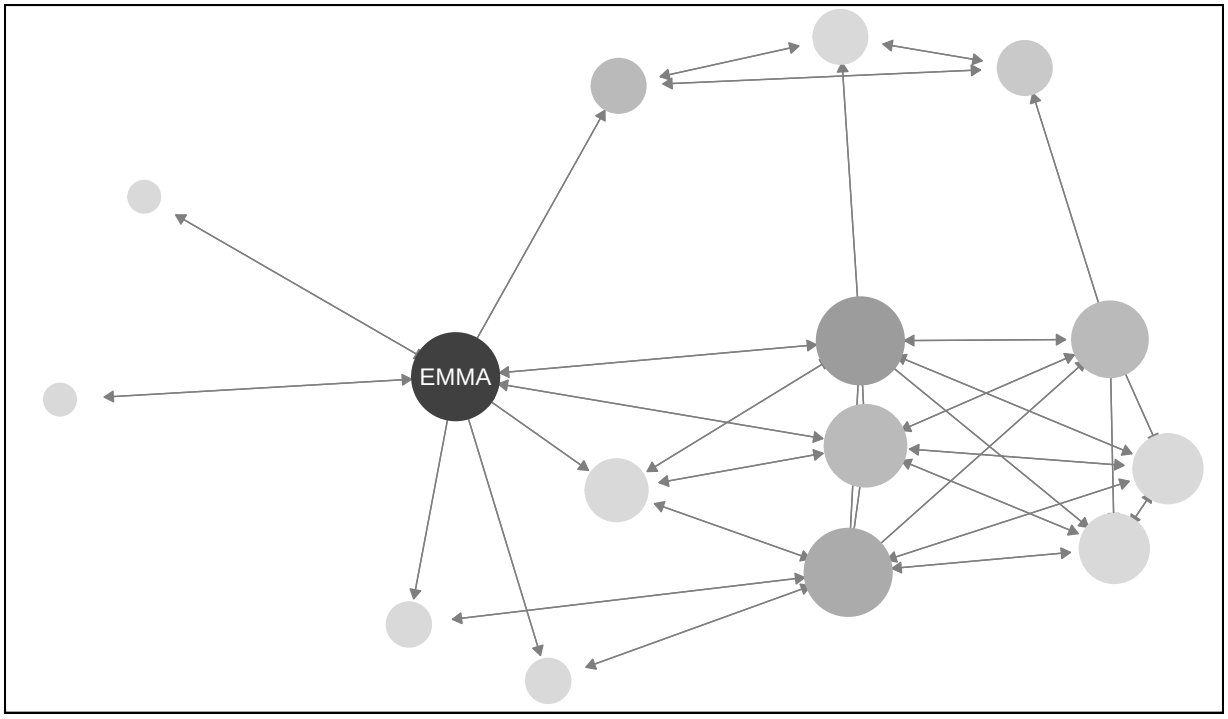
Create the social network in the Thurman data. Change the size of the nodes according to their total degree. Color the nodes varying shades of gray based on their betweenness centrality. Specifically label the node with the highest betweenness centrality. Include the name of this most central person in the graph title.

```
socnet <- network(thurman.soc, directed=T)

#calculate the betweenness
bc = betweenness(socnet)

# Create a pallete of shades of grey showing the betweenness
# Side note: grey values range from 1 (darkest) to 99 (lightest)
# The lowest degree (lightest shade) will have a grey value of 85
# The highest degree (darkest shade) will have a grey value of 25

# Label the person with the highest betweenness
bc_label <- network.vertex.names(socnet)[which.max(bc)]
bc_grey = 100 - round(bc/ max(bc), digits=1)*60 - 15
bc_node_colors = paste("grey", bc_grey,sep="")
bc_label_colors = ifelse(bc_grey<=25, "white", "black")


ggnet2(socnet, label=bc_label,
       size="degree", # size the degree
```

```
      max_size = 15, # maximum size
      node.color = bc_node_colors,  #color node by level of betweenness
      label.color=bc_label_colors,  #label node by level of betweenness
      label.size=3,
      arrow.size = 4,
      arrow.gap = 0.04) +

 labs(title="Social Network",
      subtitle = "Emma is the most central person in this social network.",
      caption = "Source: In the office: Networks and coalitions.") +
 theme(panel.border = element_rect(fill=NA,color="black"),
       plot.caption = element_text(face = "italic"),
       legend.position = "None")
```

Social Network

Emma is the most central person in this social network.



Source: In the office: Networks and coalitions.