Week 7 Challenge

Deng Chunxi

2023-10-04

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 (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:

```
{r. echo=FALSE} library(tidyverse) library(palmerpenguins) glimpse(penguins)
```

Including Plots

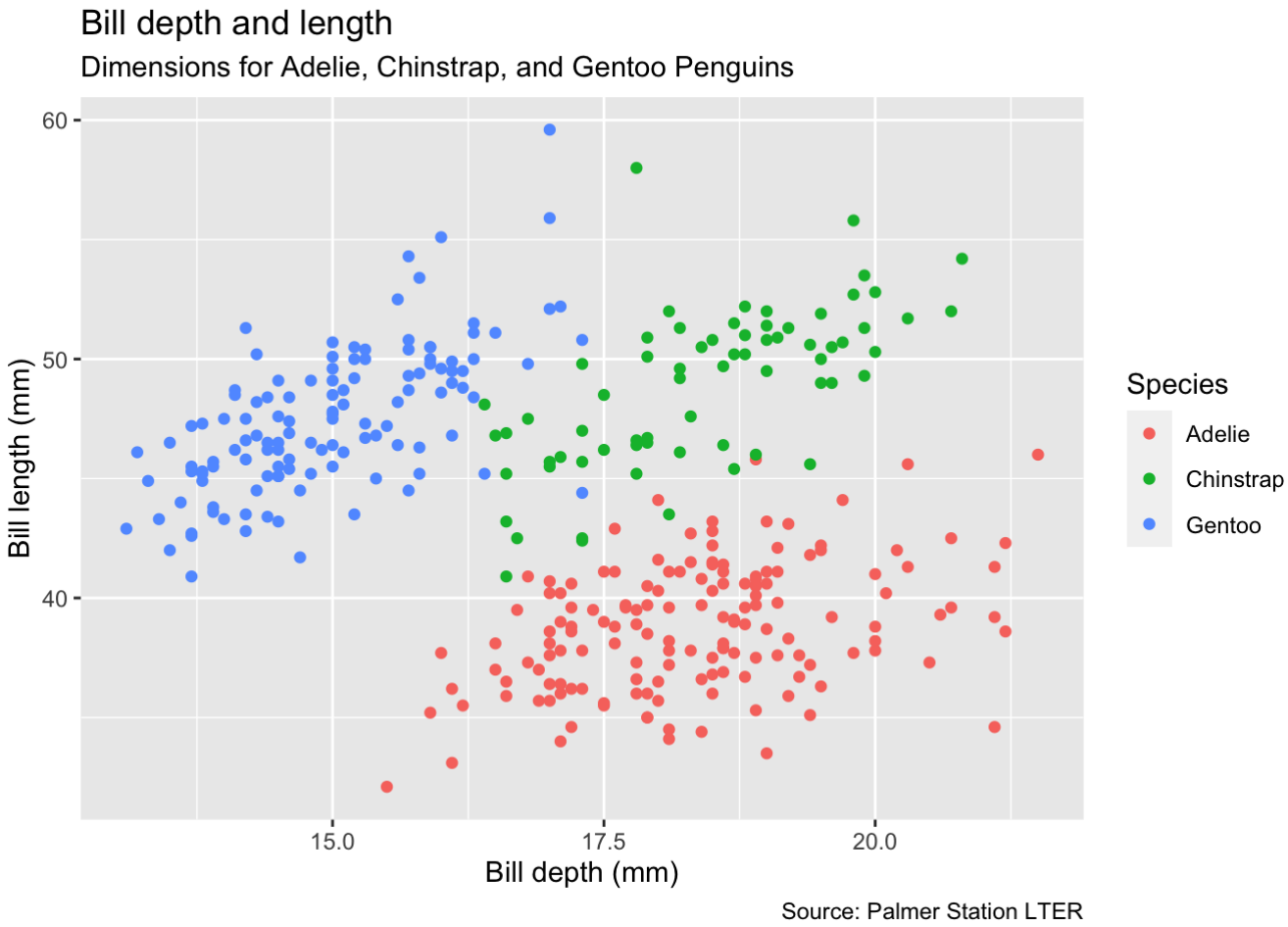
You can also embed plots, for example:

```
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

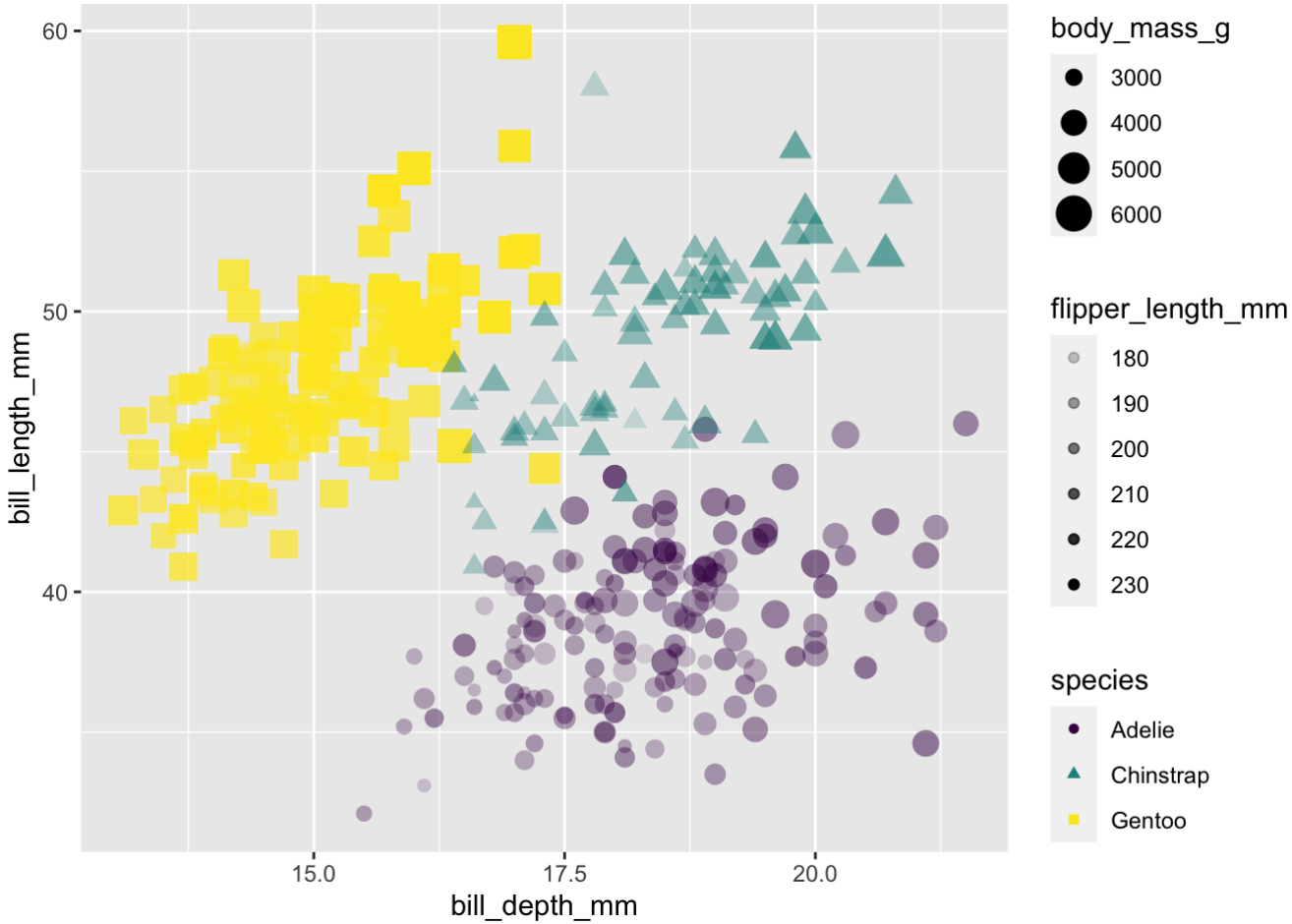
```
library(palmerpenguins)
ggplot(data = penguins, #Start with the penguins data frame
mapping = aes(x = bill_depth_mm, #Map bill depth to the x-axis
y = bill_length_mm, #Map bill length to the y-axis
colour = species)) + #Map species to the colour of each point
geom_point() + #Represent each observation with a point
labs(title = "Bill depth and length", #Title the plot "Bill depth and length"
subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins", #Add the subtitle "Dimensions for Adelie, Chi
nstrap, and Gentoo Penguins"
x = "Bill depth (mm)", y = "Bill length (mm)", #Label the x and y axes as "Bill depth (mm)" and "Bill length (m
m)", respectively
colour = "Species", #Label the legend "Species"
caption = "Source: Palmer Station LTER", #Add a caption for the data source
scale_colour_viridis_d()) #Use a discrete colour scale that is designed to be perceived by viewers with common fo
rms of colour blindness.
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```




```
ggplot(penguins, #Start with the penguins data frame
aes(x = bill_depth_mm, y = bill_length_mm, #Mapping the x and y axis
colour = species, #Separate each specie by colour
shape = species, #Separate each specie by shape
size = body_mass_g, #The size of the shape is determined by the mass
alpha = flipper_length_mm)) + #Modify transparency
geom_point() + #Represent each observation with a point
scale_colour_viridis_d() #Use a discrete colour scale that is designed to be perceived by viewers with common for
ms of colour blindness.
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```




```
library(openintro)
```

```
## Loading required package: airports
```

```
## Loading required package: cherryblossom
```

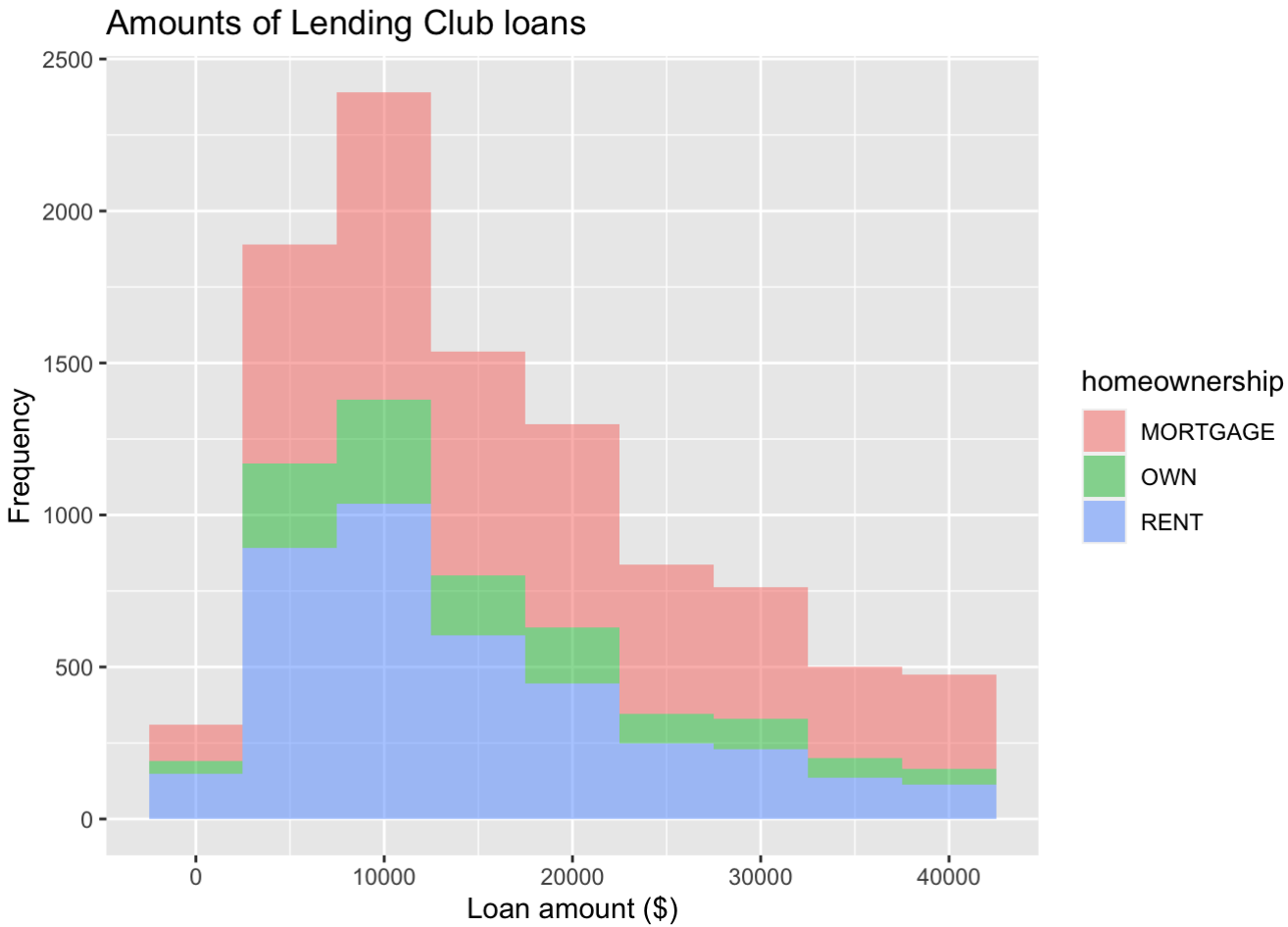
```
## Loading required package: usdata
```

```
glimpse(loans_full_schema)
```

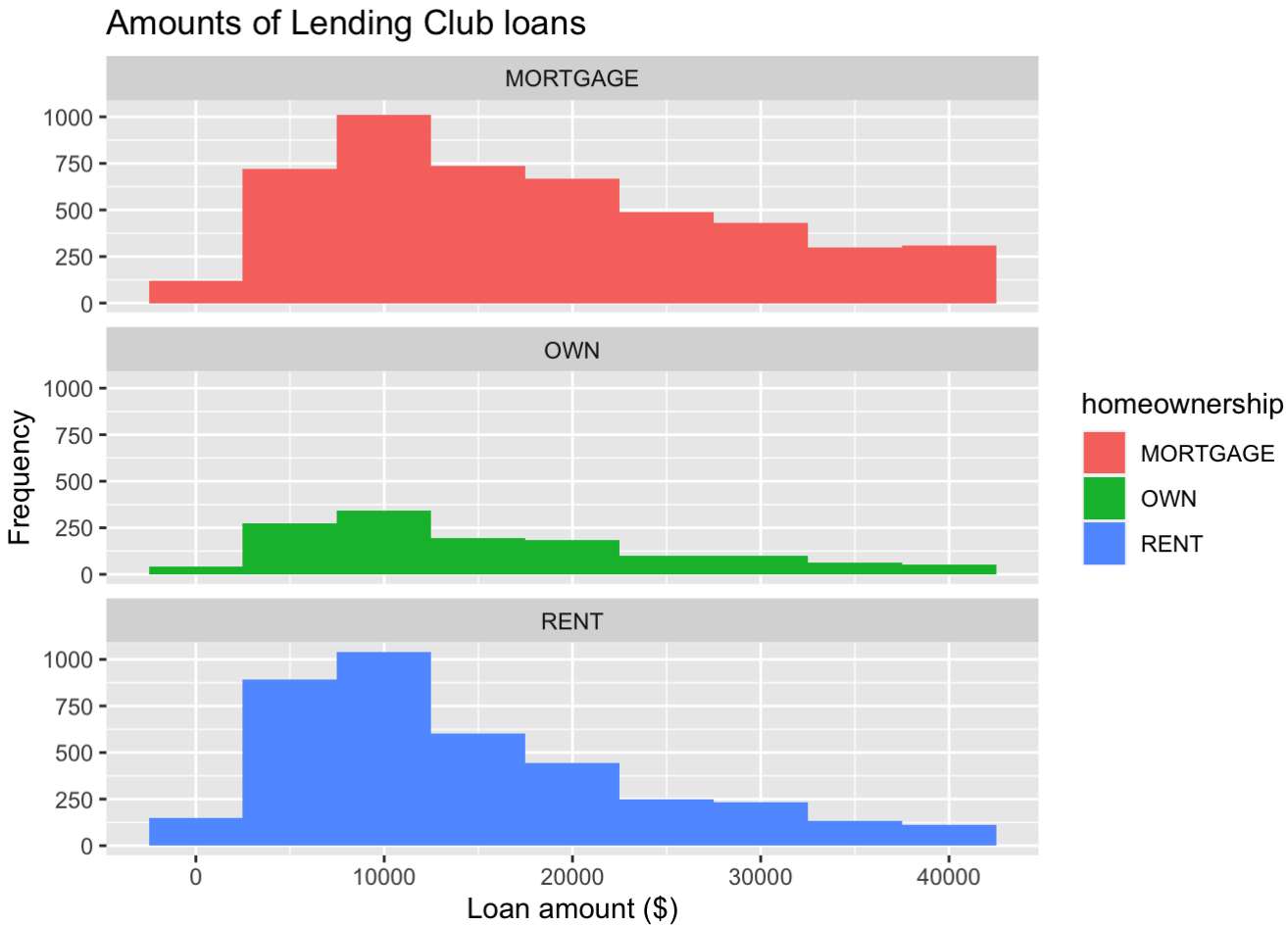
```
## Rows: 10,000
## Columns: 55
## $ emp_title                          <chr> "global config engineer ", "warehouse…
## $ emp_length                         <dbl> 3, 10, 3, 1, 10, NA, 10, 10, 10, 3, 1…
## $ state                              <fct> NJ, HI, WI, PA, CA, KY, MI, AZ, NV, I…
## $ homeownership                      <fct> MORTGAGE, RENT, RENT, RENT, RENT, OWN…
## $ annual_income                      <dbl> 90000, 40000, 40000, 30000, 35000, 34…
## $ verified_income                    <fct> Verified, Not Verified, Source Verifi…
## $ debt_to_income                     <dbl> 18.01, 5.04, 21.15, 10.16, 57.96, 6.4…
## $ annual_income_joint                <dbl> NA, NA, NA, NA, 57000, NA, 155000, NA…
## $ verification_income_joint          <fct> , , , , Verified, , Not Verified, , ,…
## $ debt_to_income_joint               <dbl> NA, NA, NA, NA, 37.66, NA, 13.12, NA,…
## $ delinq_2y                          <int> 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0…
## $ months_since_last_delinq           <int> 38, NA, 28, NA, NA, 3, NA, 19, 18, NA…
## $ earliest_credit_line               <dbl> 2001, 1996, 2006, 2007, 2008, 1990, 2…
## $ inquiries_last_12m                 <int> 6, 1, 4, 0, 7, 6, 1, 1, 3, 0, 4, 4, 8…
## $ total_credit_lines                 <int> 28, 30, 31, 4, 22, 32, 12, 30, 35, 9,…
## $ open_credit_lines                  <int> 10, 14, 10, 4, 16, 12, 10, 15, 21, 6,…
## $ total_credit_limit                 <int> 70795, 28800, 24193, 25400, 69839, 42…
## $ total_credit_utilized              <int> 38767, 4321, 16000, 4997, 52722, 3898…
## $ num_collections_last_12m           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ num_historical_failed_to_pay       <int> 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0…
## $ months_since_90d_late              <int> 38, NA, 28, NA, NA, 60, NA, 71, 18, N…
## $ current_accounts_delinq            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ total_collection_amount_ever       <int> 1250, 0, 432, 0, 0, 0, 0, 0, 0, 0, 0,…
## $ current_installment_accounts       <int> 2, 0, 1, 1, 1, 0, 2, 2, 6, 1, 2, 1, 2…
## $ accounts_opened_24m                <int> 5, 11, 13, 1, 6, 2, 1, 4, 10, 5, 6, 7…
## $ months_since_last_credit_inquiry   <int> 5, 8, 7, 15, 4, 5, 9, 7, 4, 17, 3, 4,…
## $ num_satisfactory_accounts          <int> 10, 14, 10, 4, 16, 12, 10, 15, 21, 6,…
## $ num_accounts_120d_past_due         <int> 0, 0, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, …
## $ num_accounts_30d_past_due          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ num_active_debit_accounts          <int> 2, 3, 3, 2, 10, 1, 3, 5, 11, 3, 2, 2,…
## $ total_debit_limit                  <int> 11100, 16500, 4300, 19400, 32700, 272…
## $ num_total_cc_accounts              <int> 14, 24, 14, 3, 20, 27, 8, 16, 19, 7, …
## $ num_open_cc_accounts               <int> 8, 14, 8, 3, 15, 12, 7, 12, 14, 5, 8,…
## $ num_cc_carrying_balance            <int> 6, 4, 6, 2, 13, 5, 6, 10, 14, 3, 5, 3…
## $ num_mort_accounts                  <int> 1, 0, 0, 0, 0, 3, 2, 7, 2, 0, 2, 3, 3…
## $ account_never_delinq_percent       <dbl> 92.9, 100.0, 93.5, 100.0, 100.0, 78.1…
```

```
## $ tax_liens                  <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ public_record_bankrupt     <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0…
## $ loan_purpose               <fct> moving, debt_consolidation, other, de…
## $ application_type           <fct> individual, individual, individual, i…
## $ loan_amount                <int> 28000, 5000, 2000, 21600, 23000, 5000…
## $ term                       <dbl> 60, 36, 36, 36, 36, 36, 60, 60, 36, 3…
## $ interest_rate              <dbl> 14.07, 12.61, 17.09, 6.72, 14.07, 6.7…
## $ installment                <dbl> 652.53, 167.54, 71.40, 664.19, 786.87…
## $ grade                      <fct> C, C, D, A, C, A, C, B, C, A, C, B, C…
## $ sub_grade                  <fct> C3, C1, D1, A3, C3, A3, C2, B5, C2, A…
## $ issue_month                <fct> Mar-2018, Feb-2018, Feb-2018, Jan-201…
## $ loan_status                <fct> Current, Current, Current, Current, C…
## $ initial_listing_status     <fct> whole, whole, fractional, whole, whol…
## $ disbursement_method        <fct> Cash, Cash, Cash, Cash, Cash, Cash, C…
## $ balance                    <dbl> 27015.86, 4651.37, 1824.63, 18853.26,…
## $ paid_total                 <dbl> 1999.330, 499.120, 281.800, 3312.890,…
## $ paid_principal             <dbl> 984.14, 348.63, 175.37, 2746.74, 1569…
## $ paid_interest              <dbl> 1015.19, 150.49, 106.43, 566.15, 754.…
## $ paid_late_fees             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
```

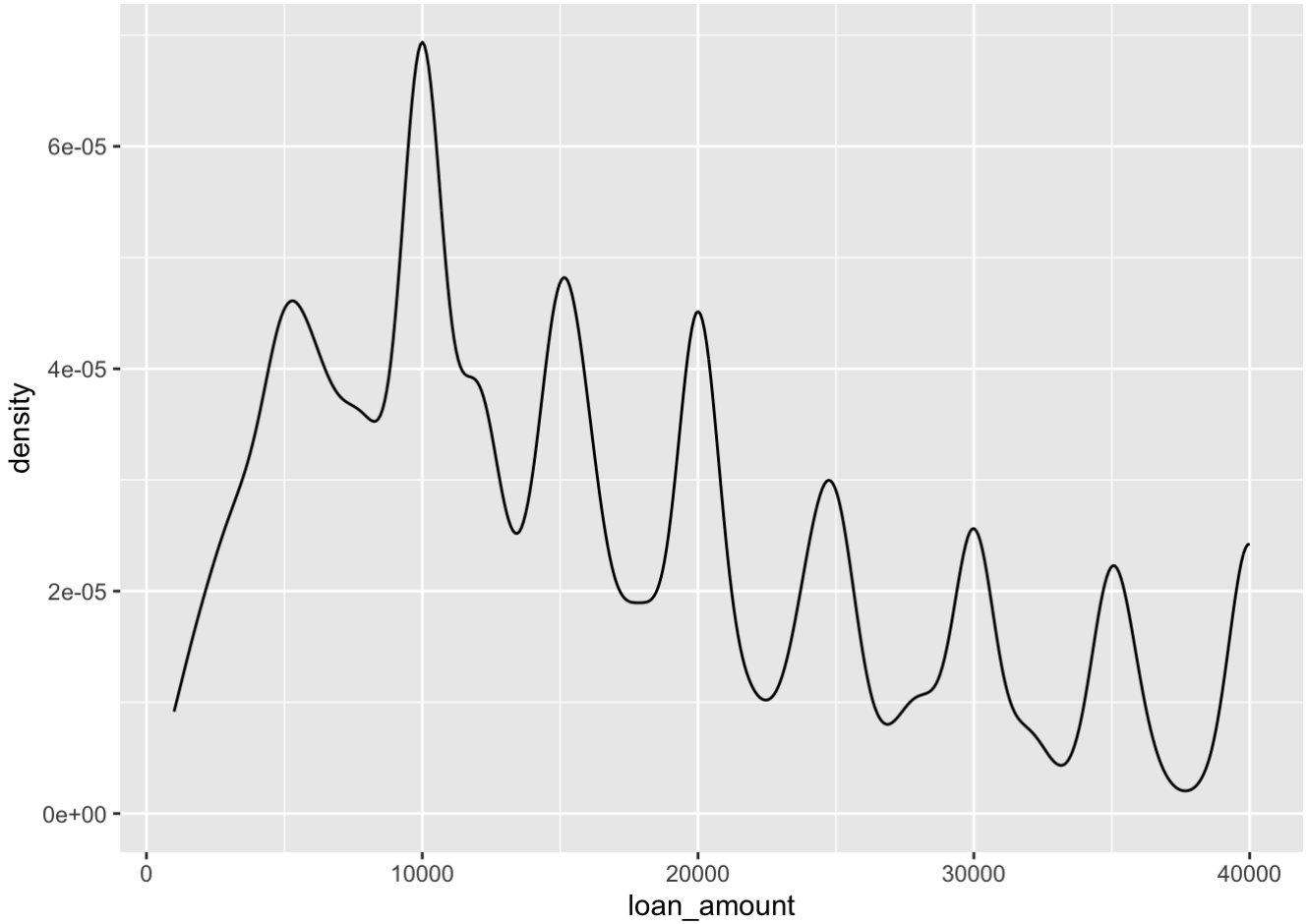
```
ggplot(loans_full_schema, #Start with the loans_full_scheme data frame
aes(x = loan_amount, #Map loan_amount to x-axis
fill = homeownership)) + #selecting the variable homeownership
geom_histogram(binwidth = 5000, alpha = 0.5) + #Setting the width and transparency of the histogram
labs(x = "Loan amount ($)", #Label the x axis
     y = "Frequency", #Label the y axis
     title = "Amounts of Lending Club loans") #Label the title
```



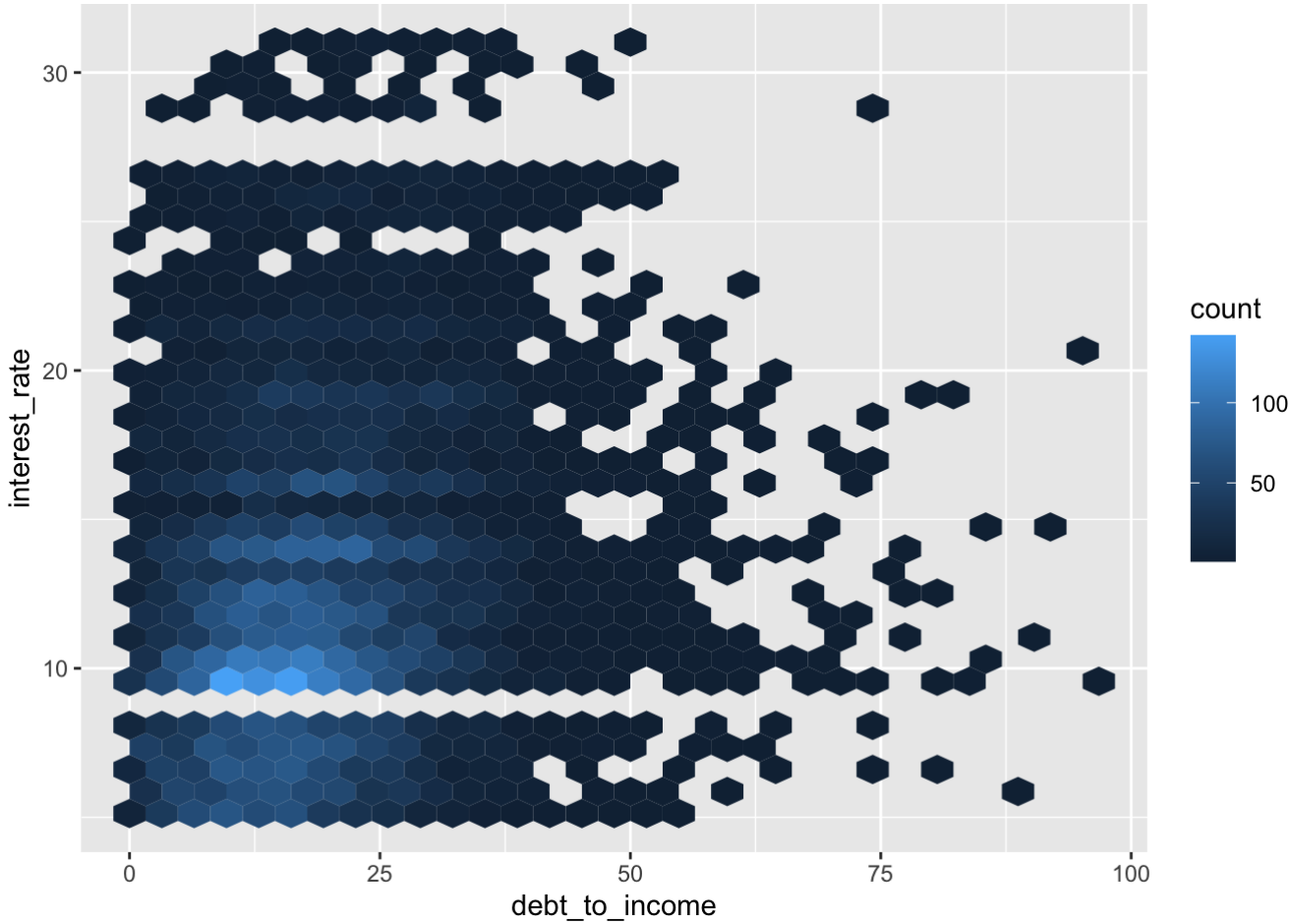

```
ggplot(loans_full_schema, #start with the data frame
       aes(x = loan_amount, fill = homeownership)) + #map loan_amount to x-axis and fill with homeownership
  geom_histogram(binwidth = 5000) + #adjust binwidth to 5000
labs(x = "Loan amount ($)",y = "Frequency",title = "Amounts of Lending Club loans") + #Label the graph
facet_wrap(~ homeownership, nrow = 3) #Number of rows
```



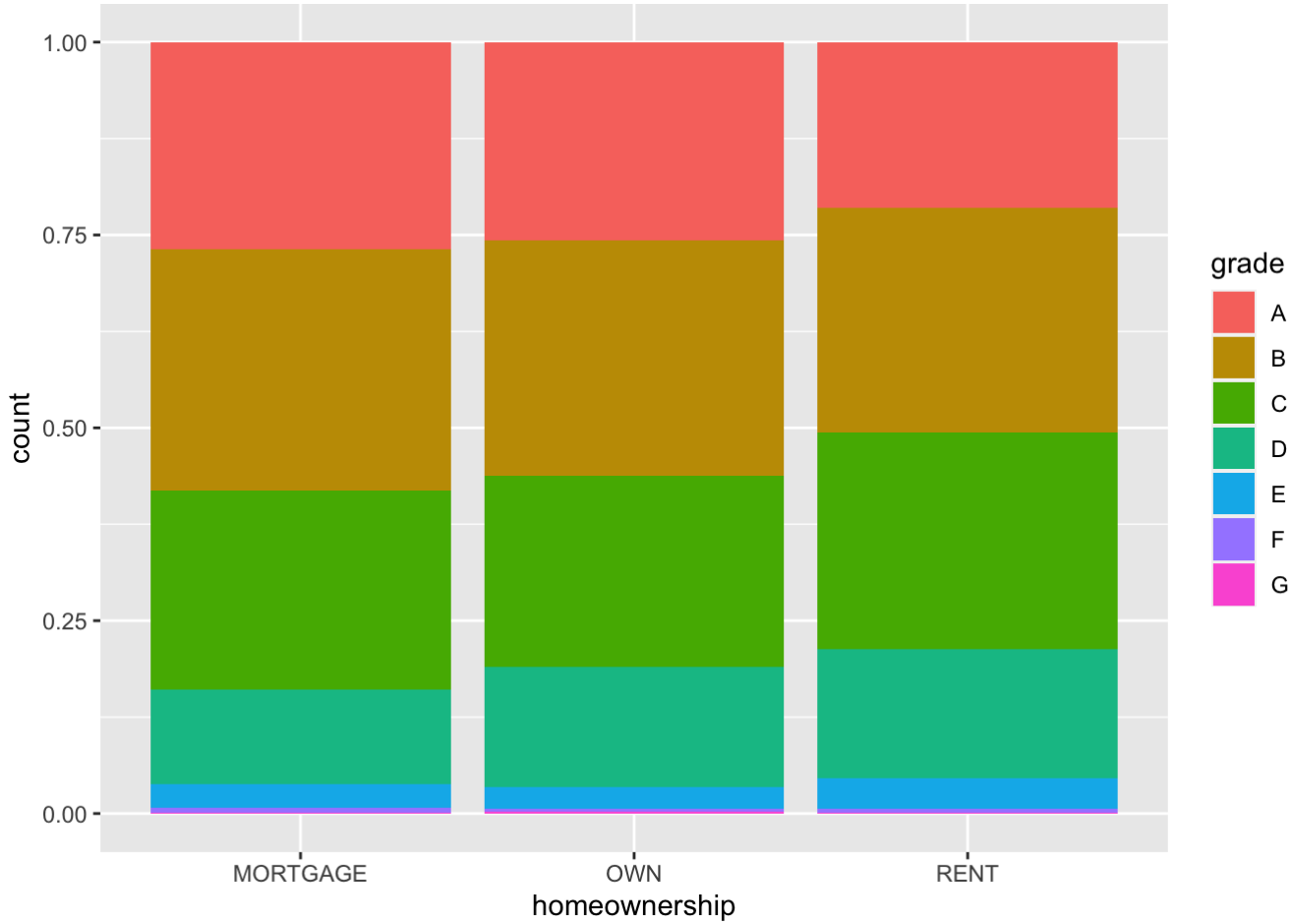

```
ggplot(loans_full_schema, #Start with data frame
       aes(x = loan_amount)) + #Map loan_amount to x-axis
geom_density(adjust = 0.5) #adjudt binwidth to 0.5
```



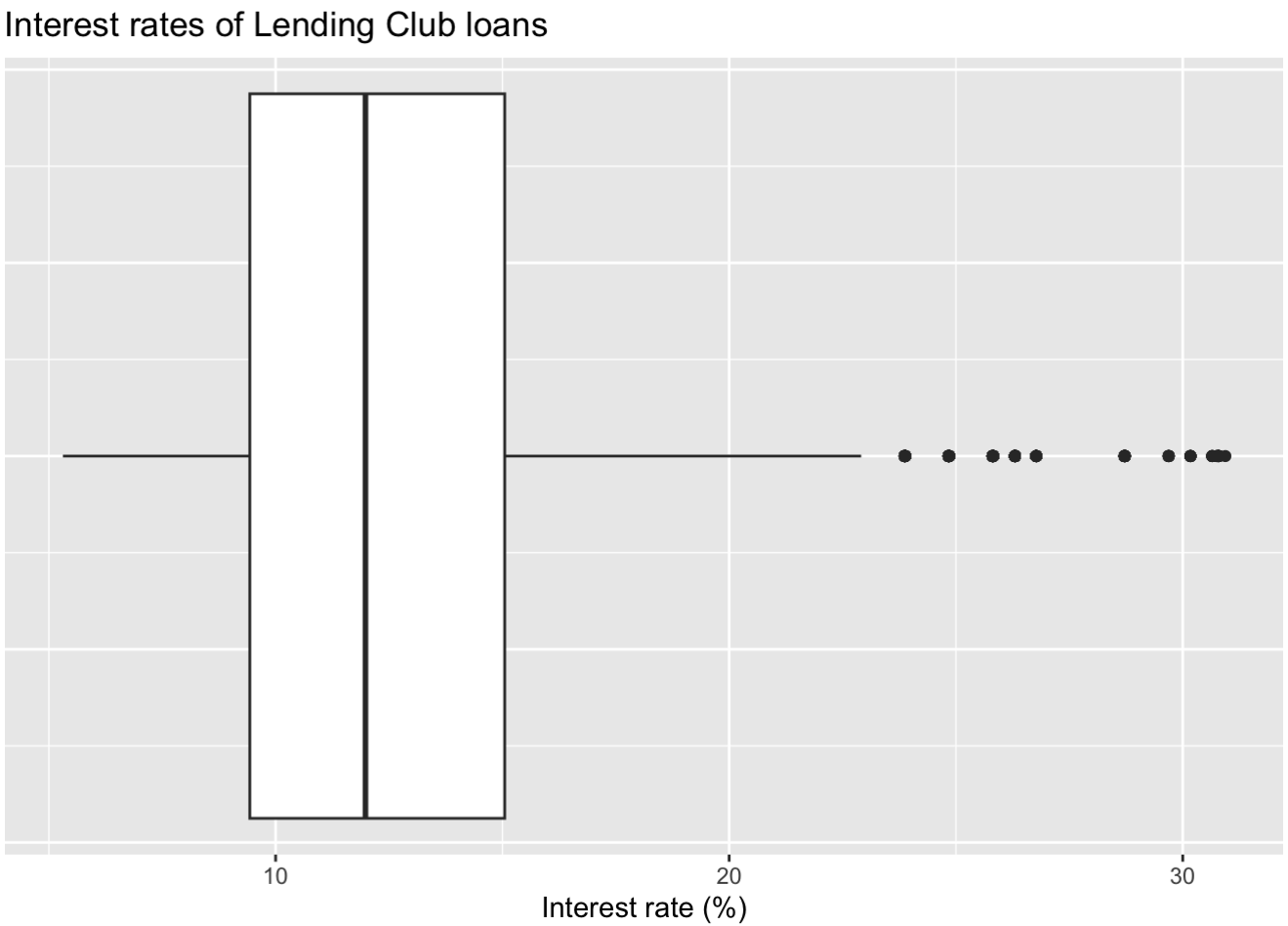

```
ggplot(loans_full_schema %>% #Start with the loans_full_scheme data frame
filter(debt_to_income < 100), #Filter the variable debt_to_income which is less than 100
aes(x = debt_to_income, #Map debt_to_income to x-axis
y = interest_rate)) + #Map interest_rate to y-axis
geom_hex() #Represent each observation with a hex
```



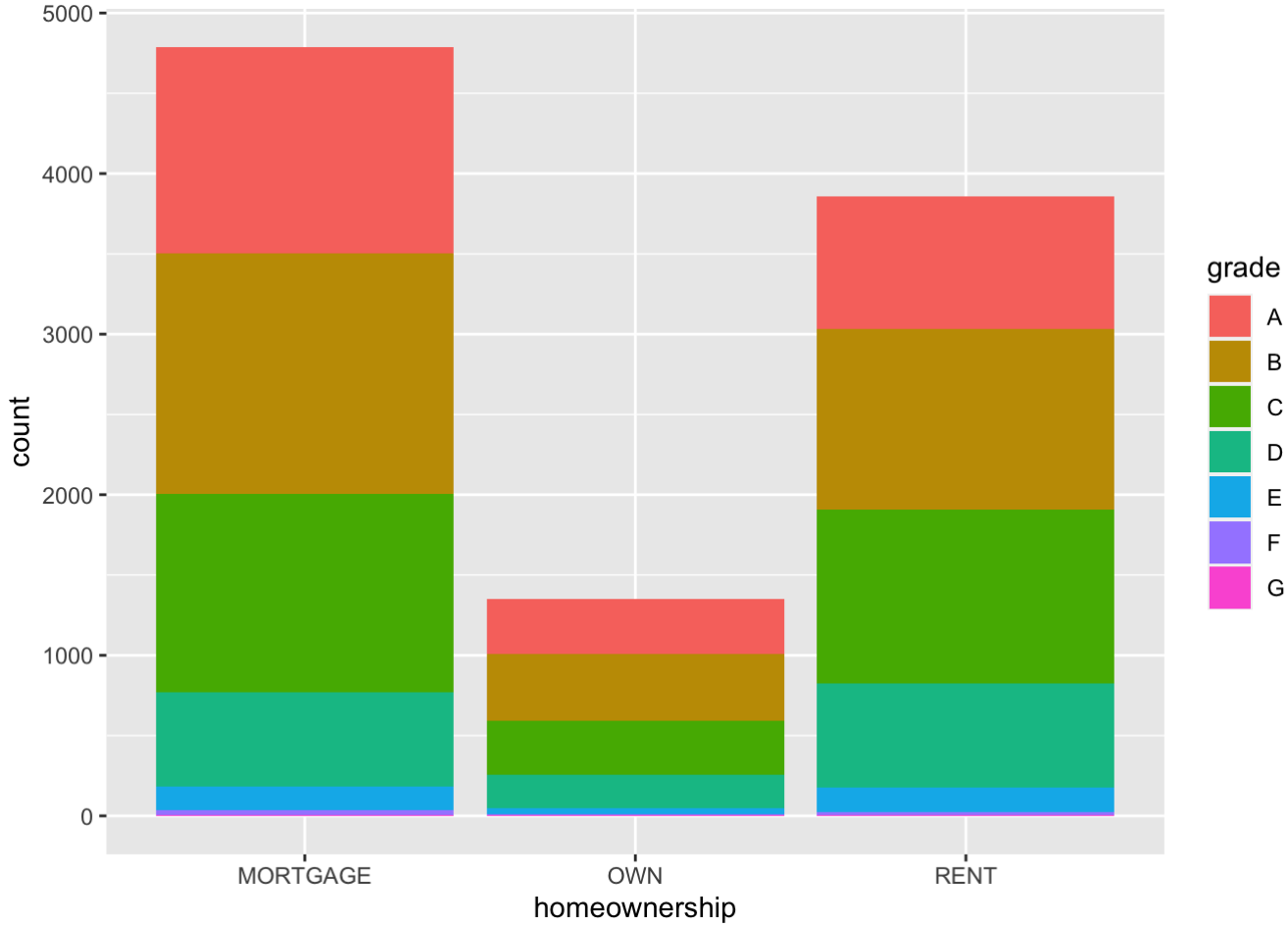

```
ggplot(loans_full_schema, #Start with the loans_full_schema data frame
       aes(x = homeownership, fill = grade)) + # #Map homeownership to x-axis and fill with grade
geom_bar(position = "fill") #Calculate the proportion by percentage
```




```
ggplot(loans_full_schema, aes(x = interest_rate)) + #giving x-axis a frame
 geom_boxplot() + #type of chart
 labs(x = "Interest rate (%)",y = NULL, #mapping to the x-axis and y-axis
 title = "Interest rates of Lending Club loans") + #giving a title to the graph
 theme( axis.ticks.y = element_blank(), axis.text.y = element_blank() ) #customising the non-data components of t
he graph
```

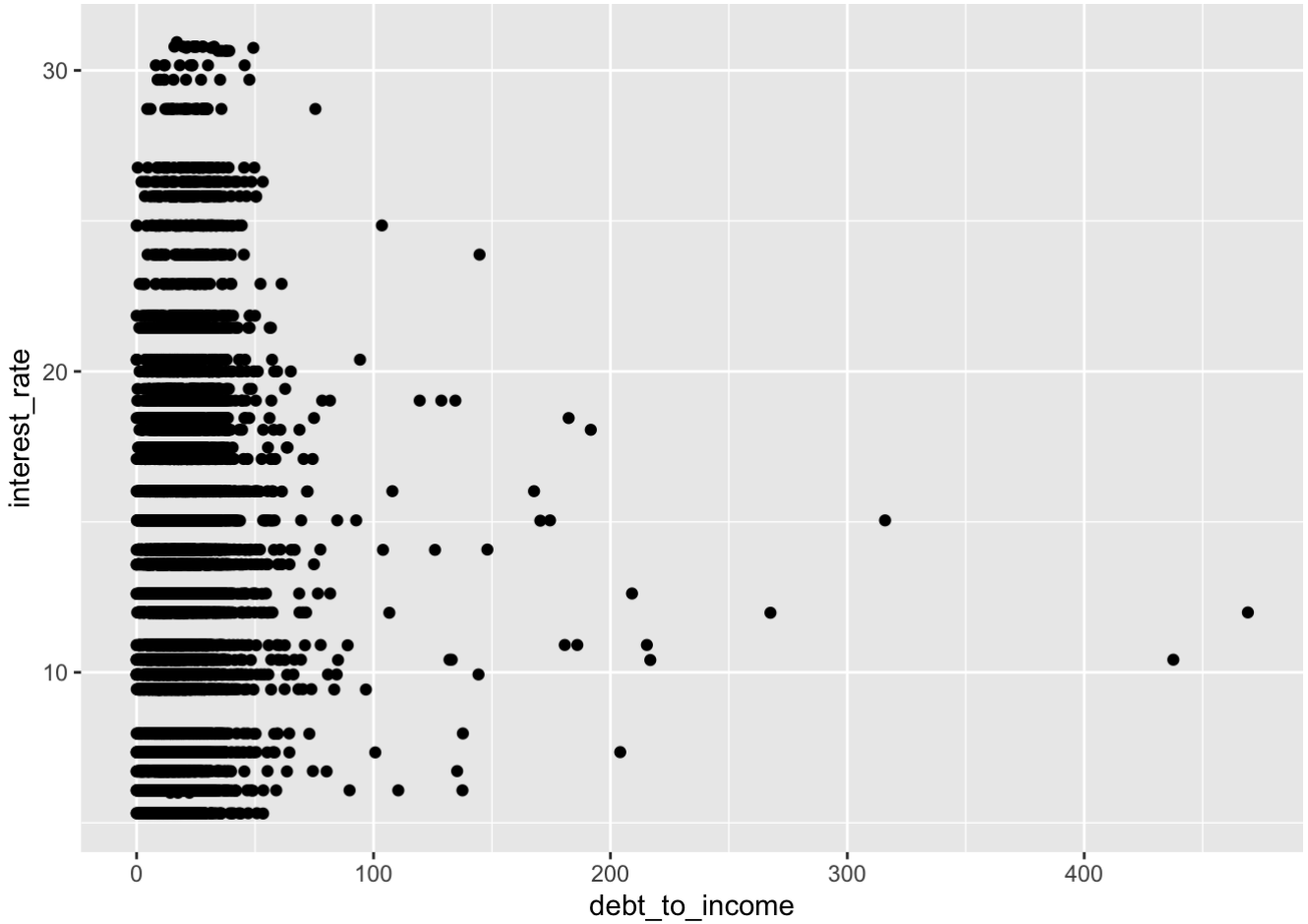



```
ggplot(loans_full_schema, aes(x = homeownership, #using loans data frame, mapping x-axis
 fill = grade)) + #filling with a categorical variable
 geom_bar() #type of graph
```

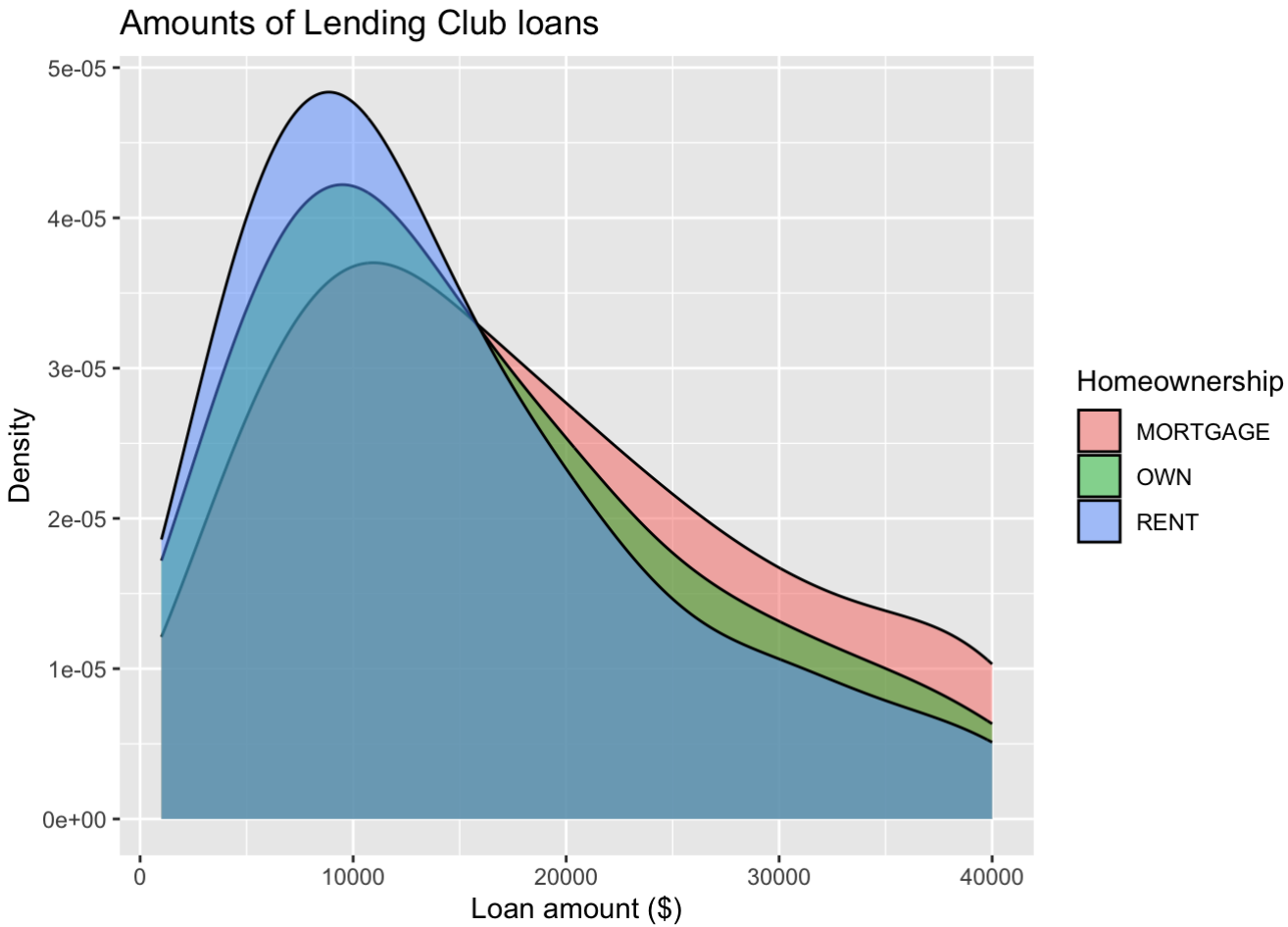



```
ggplot(loans_full_schema, #Start with loan_full_schema date frame
       aes(x = debt_to_income, y = interest_rate)) + #Label the x and y axis
geom_point() #Represent each observation as a point
```

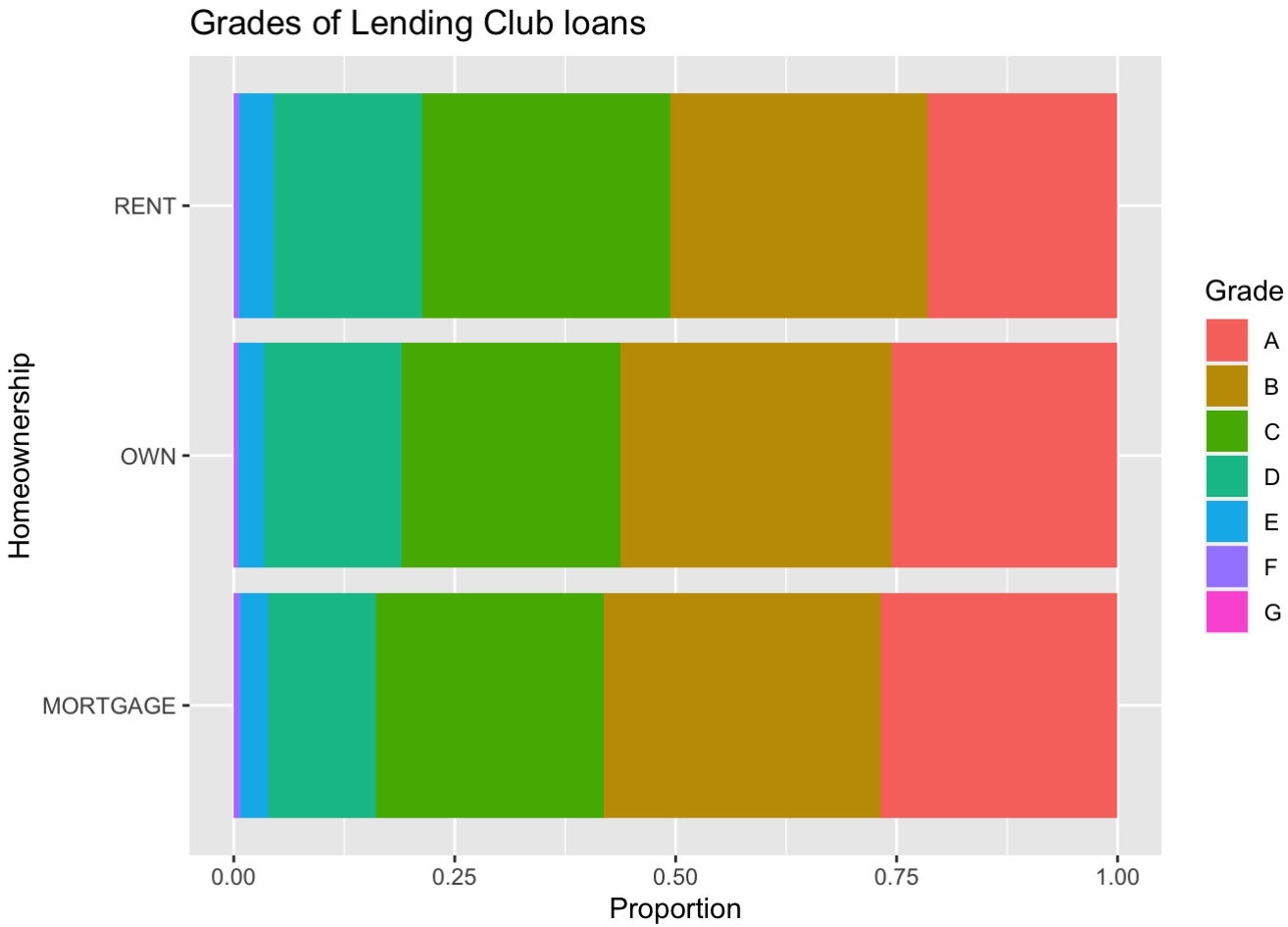
```
## Warning: Removed 24 rows containing missing values (`geom_point()`).
```




```
ggplot(loans_full_schema, #Start with the loans_full_schema data frame
      aes(x = loan_amount, fill = homeownership)) + #Map loan-amount to x-axis and fill with homeownership
geom_density(adjust = 2, alpha = 0.5) + #adjuste bandwidth to 2 and transparency to 0.5
labs(x = "Loan amount ($)", #Label x-axis
     y = "Density", #Label y-axis
     title = "Amounts of Lending Club loans", fill = "Homeownership") #Label title and fill with homeownership
```

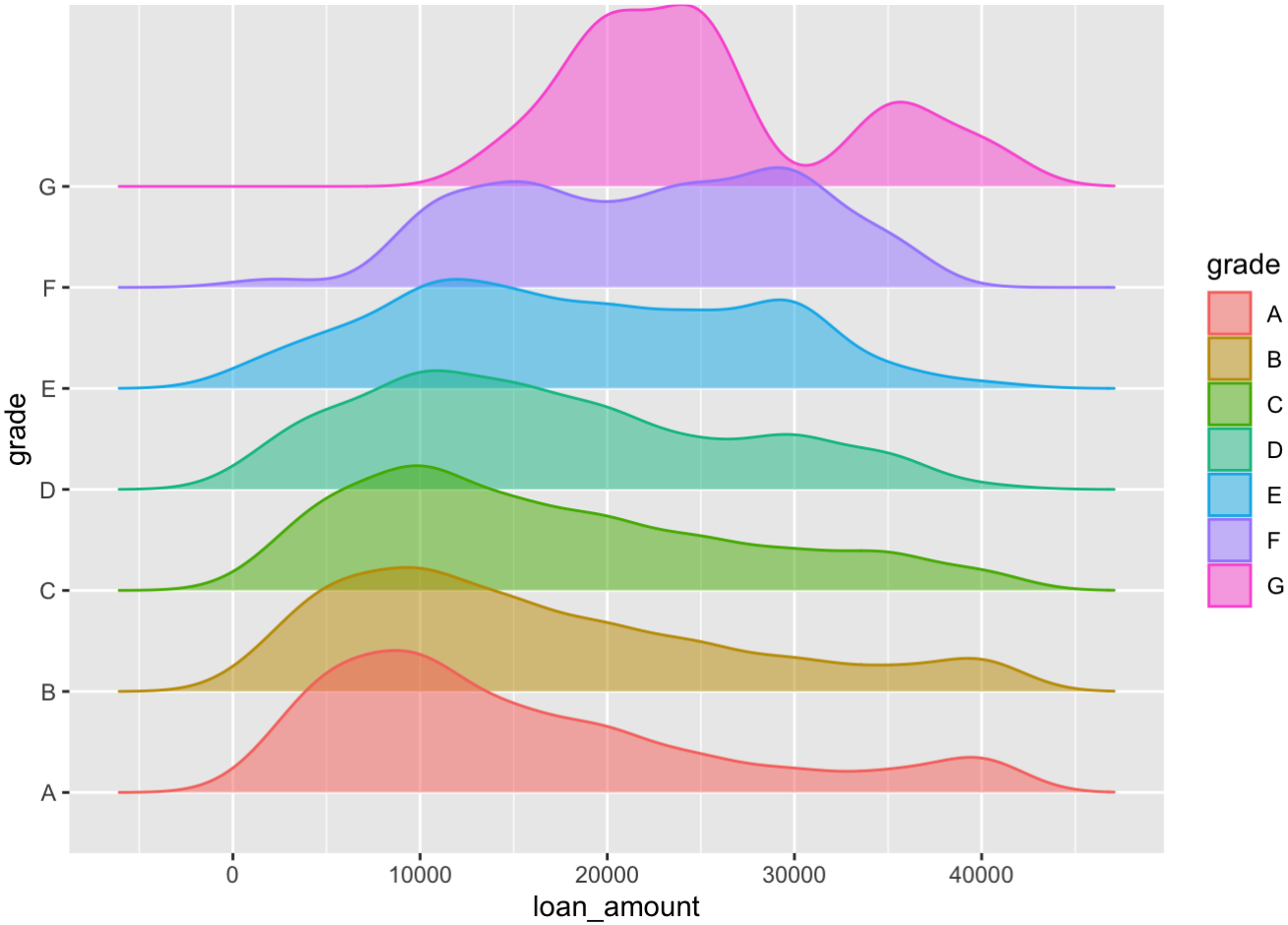



```
ggplot(loans_full_schema, #Start with the loans_full_scheme data frame
aes(y = homeownership, fill = grade)) + #Map homeownership to y-axis and fill with grade
geom_bar(position = "fill") + #Calculate the proportion for each grade
labs( x = "Proportion", #Label x-axis
      y = "Homeownership", #Label y-axis
      fill = "Grade", #Calculate the proportion for each grade
      title = "Grades of Lending Club loans") #Label title
```




```
library(ggridges)
ggplot(loans_full_schema, #Start with the loans_full_scheme data frame
       aes(x = loan_amount, #Map loan-amount to x-axis
           y = grade, #Map grade to y-axis
           fill = grade, #Represent each grade as proportion
           color = grade)) + #Separate each grade by colour
geom_density_ridges(alpha = 0.5) #Setting the transparency
```

```
## Picking joint bandwidth of 2360
```


grade
G
F
E
D
C
B
A
0
10000
20000
30000
40000
loan_amount
grade
A
B
C
D
E
F
G