

Challenge 7

Ariel Quek

2023-10-02

```
knitr::opts_chunk$set(echo = TRUE)
```

Challenge 7 Instructions

1. You are required to focus on the lecture videos and materials, owing to the break last week
2. The material taught this week is of significant importance for your final projects
3. The challenge this week will be your workbook: a document that you will generate by trying and experimenting with the code snippets taught in the lecture material.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.1
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr      1.1.0      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr   1.5.0
## ✓ ggplot2    3.4.3      ✓ tibble     3.1.8
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — Conflicts ————— tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the [8];http://conflicted.r-lib.org/[8]; to force all conflict
s to become errors
```

```
install.packages("palmerpenguins",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Ariel/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'palmerpenguins' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\Ariel\AppData\Local\Temp\RtmpwJTJTP\downloaded_packages
```

```
library(palmerpenguins)
```

```
## Warning: package 'palmerpenguins' was built under R version 4.2.3
```

```
View(penguins)
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel...
## $ island       <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse...
## $ bill_length_mm <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ...
## $ bill_depth_mm <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ...
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186...
## $ body_mass_g   <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ...
## $ sex          <fct> male, female, female, NA, female, male, female, male...
## $ year         <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007...
```

Recreating Plot for Palmer Penguins

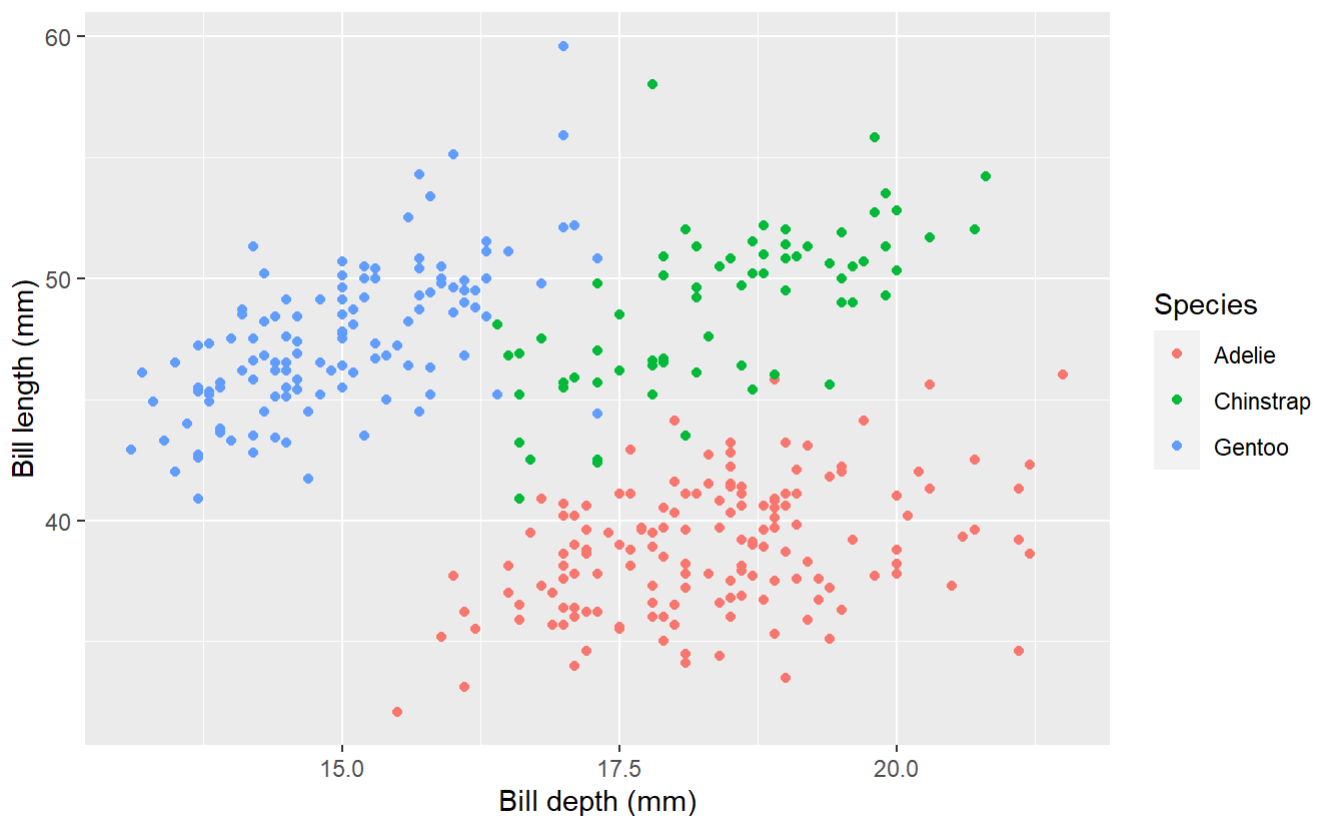
Plotting the bill depth against bill length of penguins

```
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species))+
geom_point() +
labs(title = "Bill depth and length",
     subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
     x = "Bill depth (mm)",
     y = "Bill length (mm)",
     colour = "Species",
     caption = "Source: Palmer Station LTER/ palmerpenguins package",
scale_colour_viridis_d())
```

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

Bill depth and length

Dimensions for Adelie, Chinstrap, and Gentoo Penguins

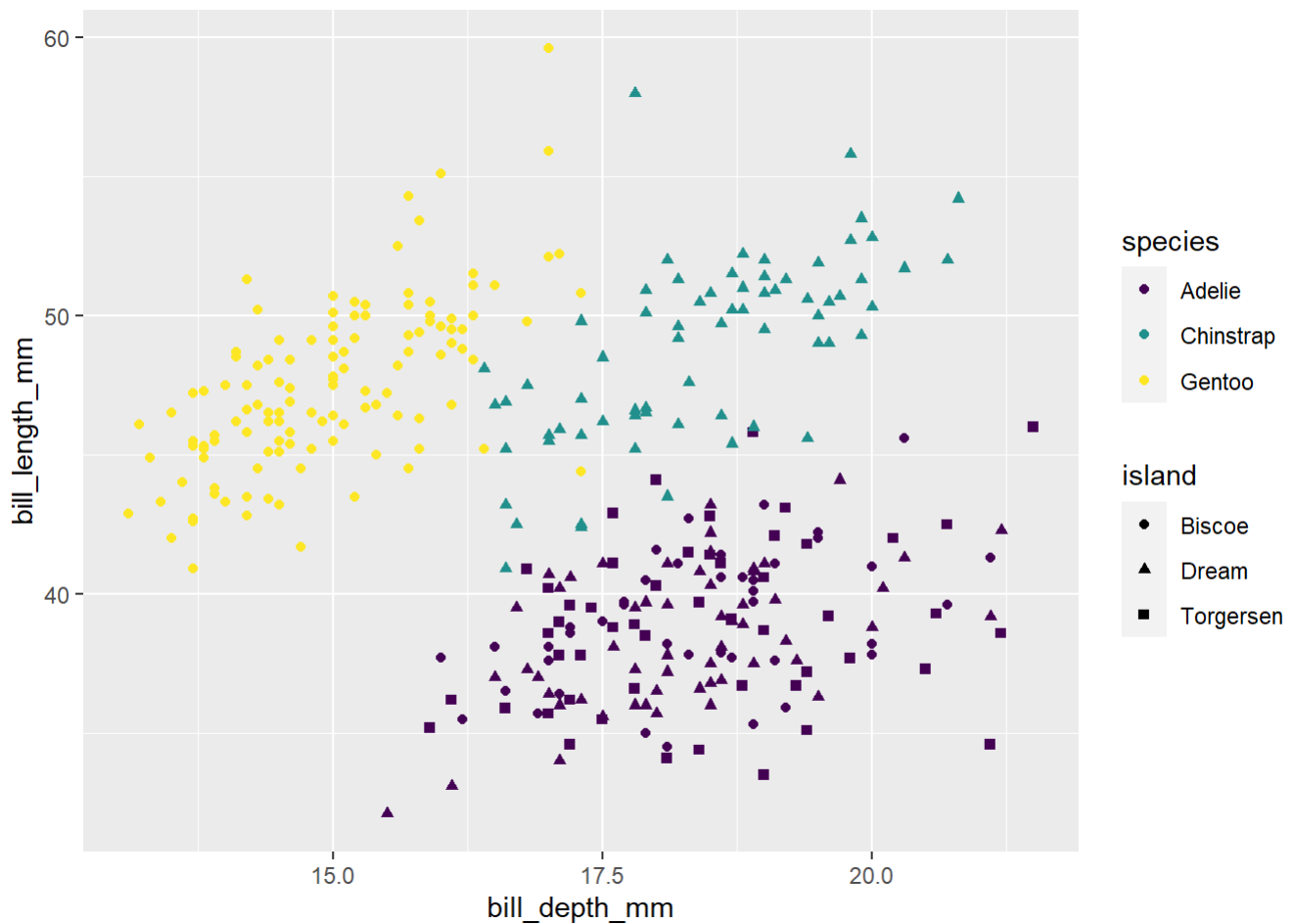


Source: Palmer Station LTER/ palmerpenguins package

Plotting bill depth against bill length with shaped data points:

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm, colour = species,
                    shape = island)) +
geom_point() + scale_colour_viridis_d()
```

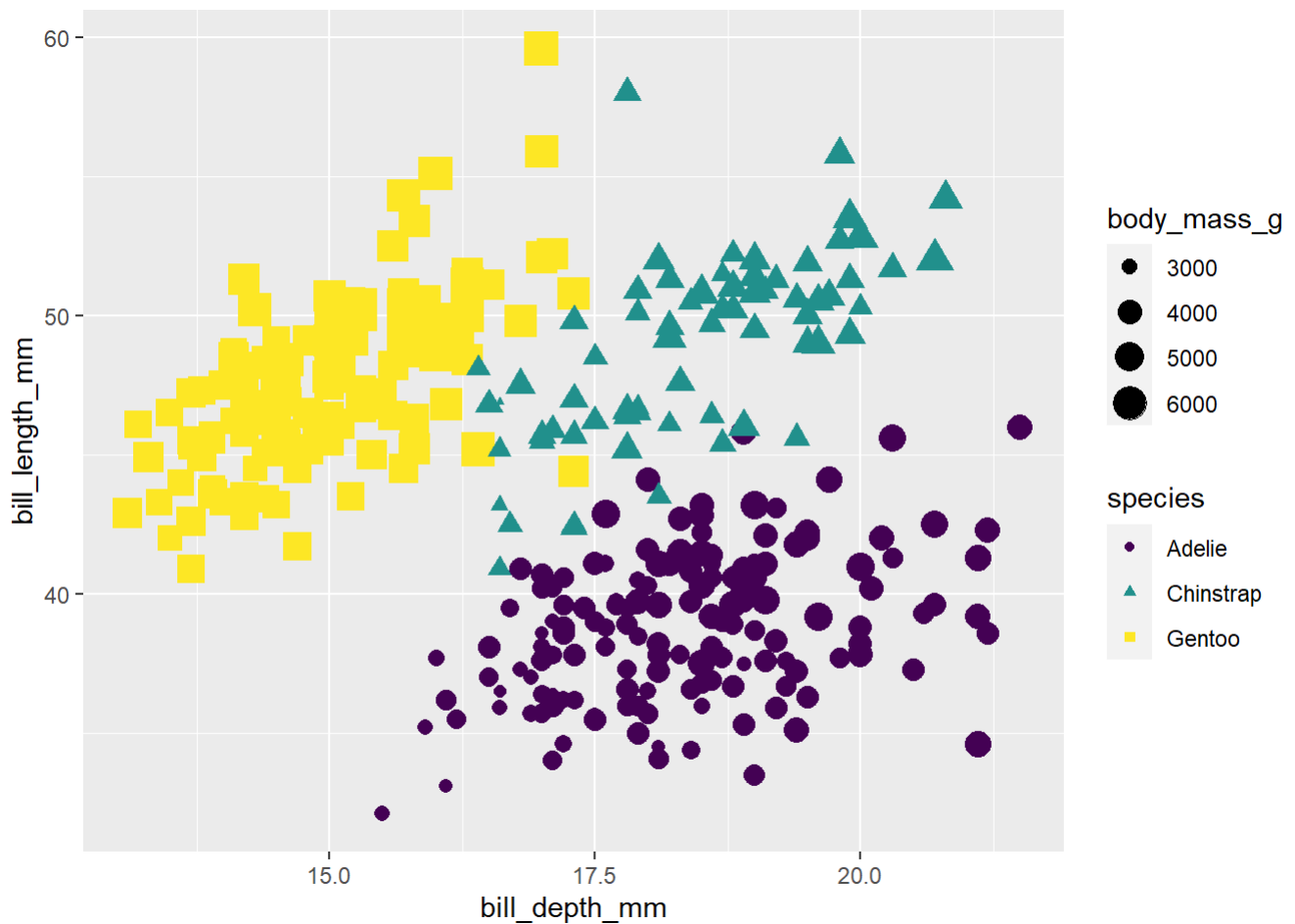
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



Changing the size of data points on the plot

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm, colour = species, shape = species,
size = body_mass_g)) +
  geom_point() + scale_colour_viridis_d()
```

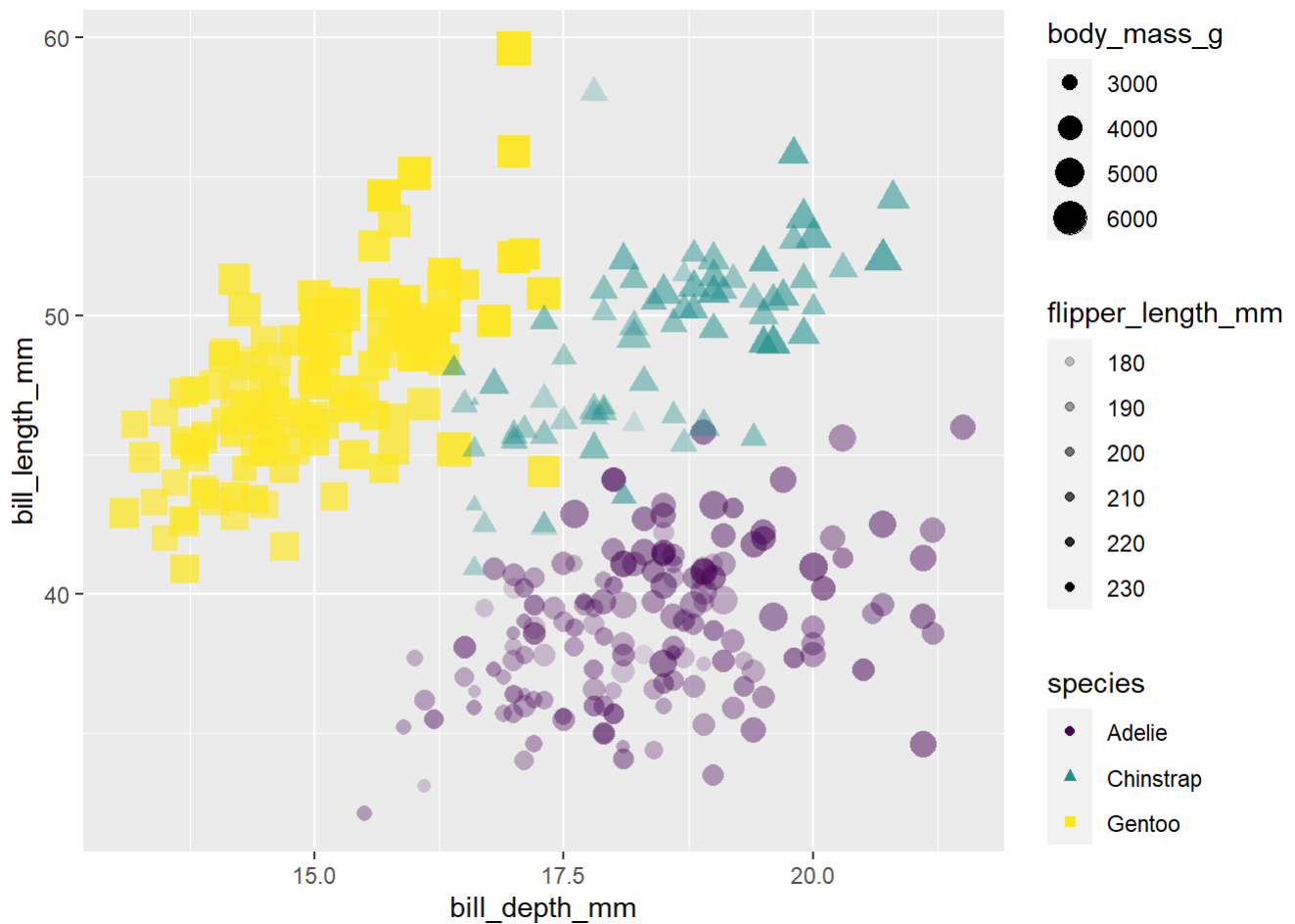
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



Manipulating alpha (opacity) of data points in the plot

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm, colour = species,
  shape = species, size = body_mass_g, alpha = flipper_length_mm)) +
  geom_point() + scale_colour_viridis_d()
```

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

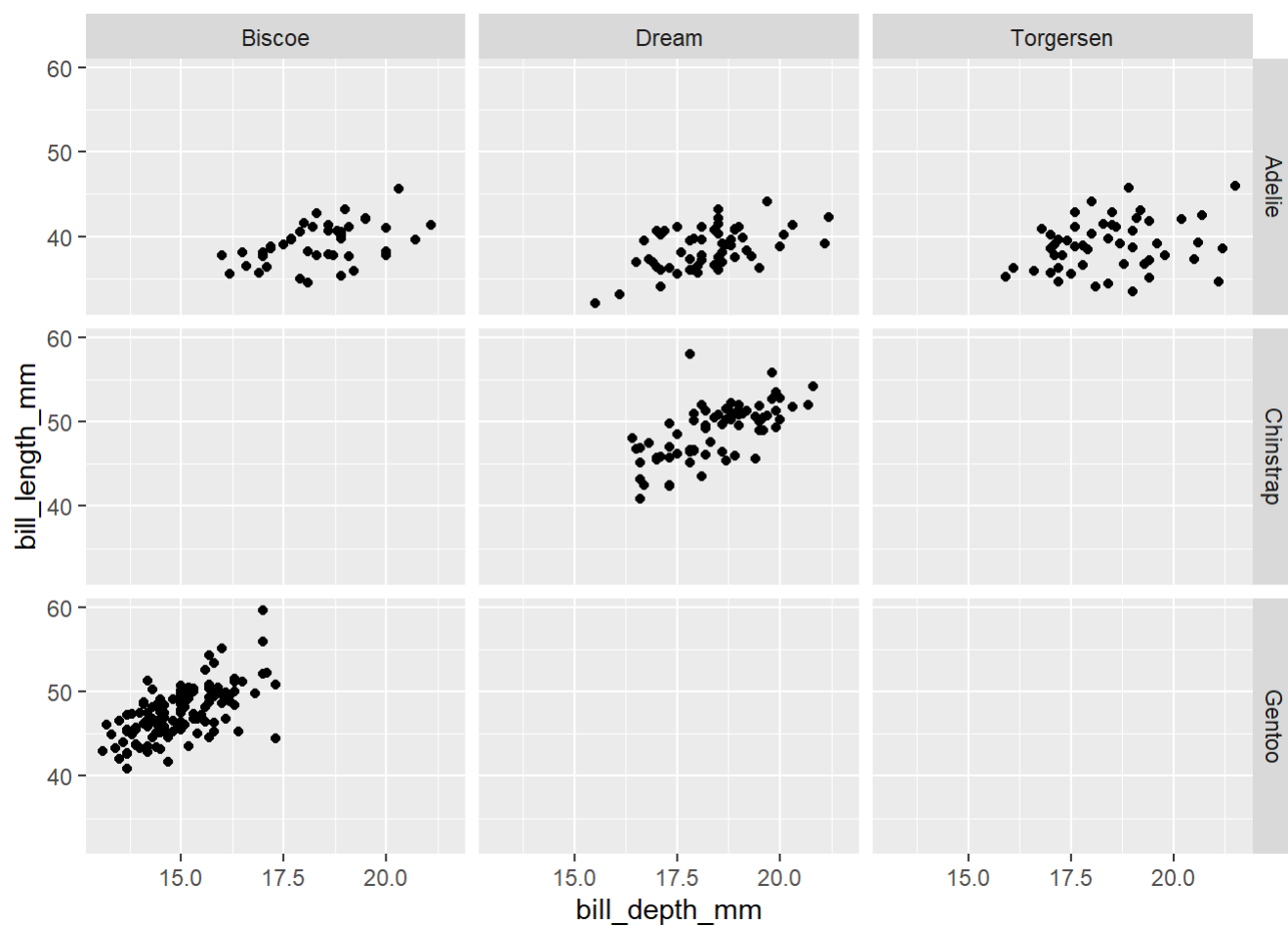


Faceting: creating smaller plots that display different subsets of the data

1. Plotting the bill length and bill depth of penguins based on the species and the island

```
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_grid(species ~ island)
```

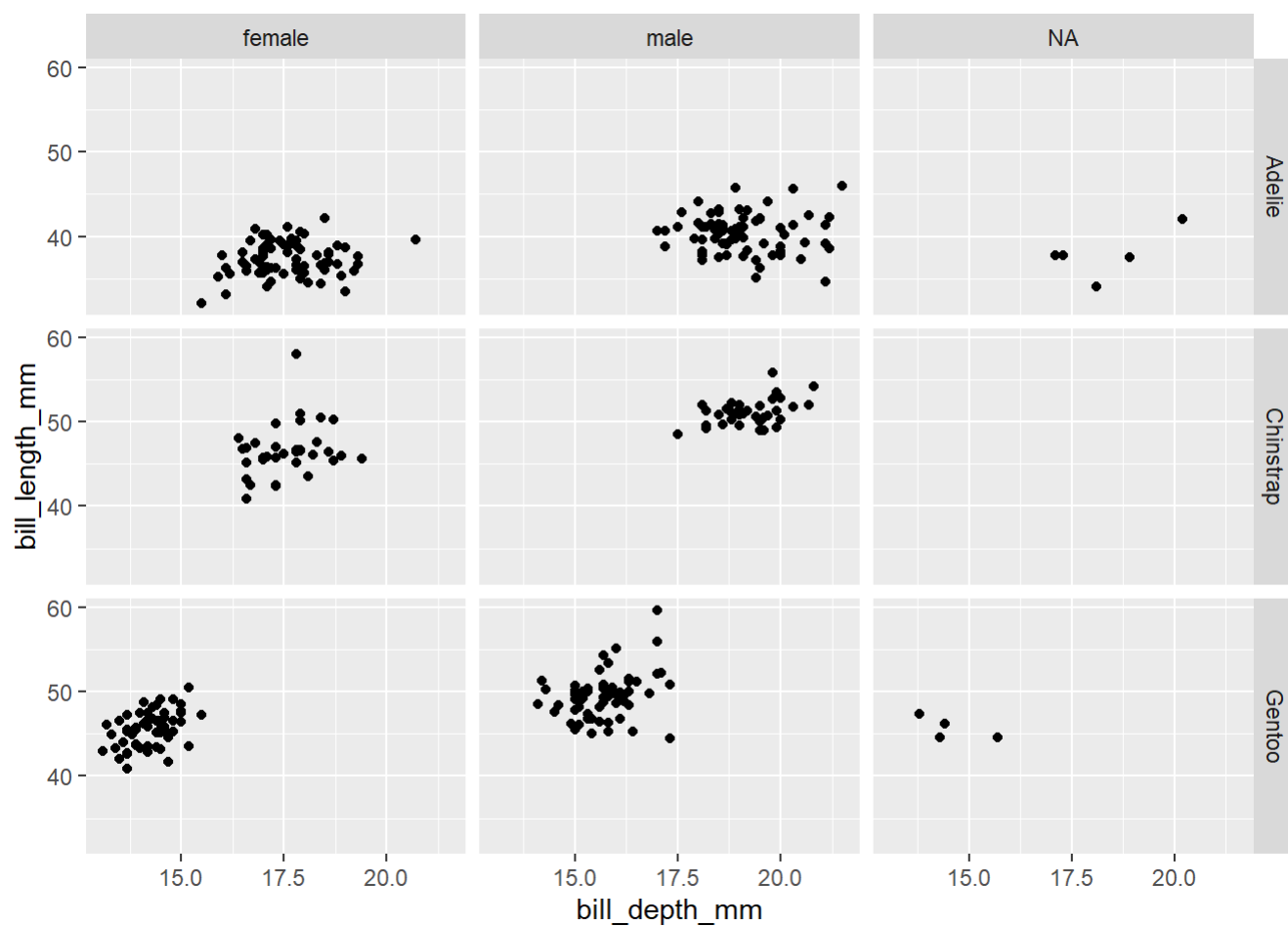
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



2. Plotting bill length and bill depth of penguins based on sex against species

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) + geom_point() +  
facet_grid(species ~ sex)
```

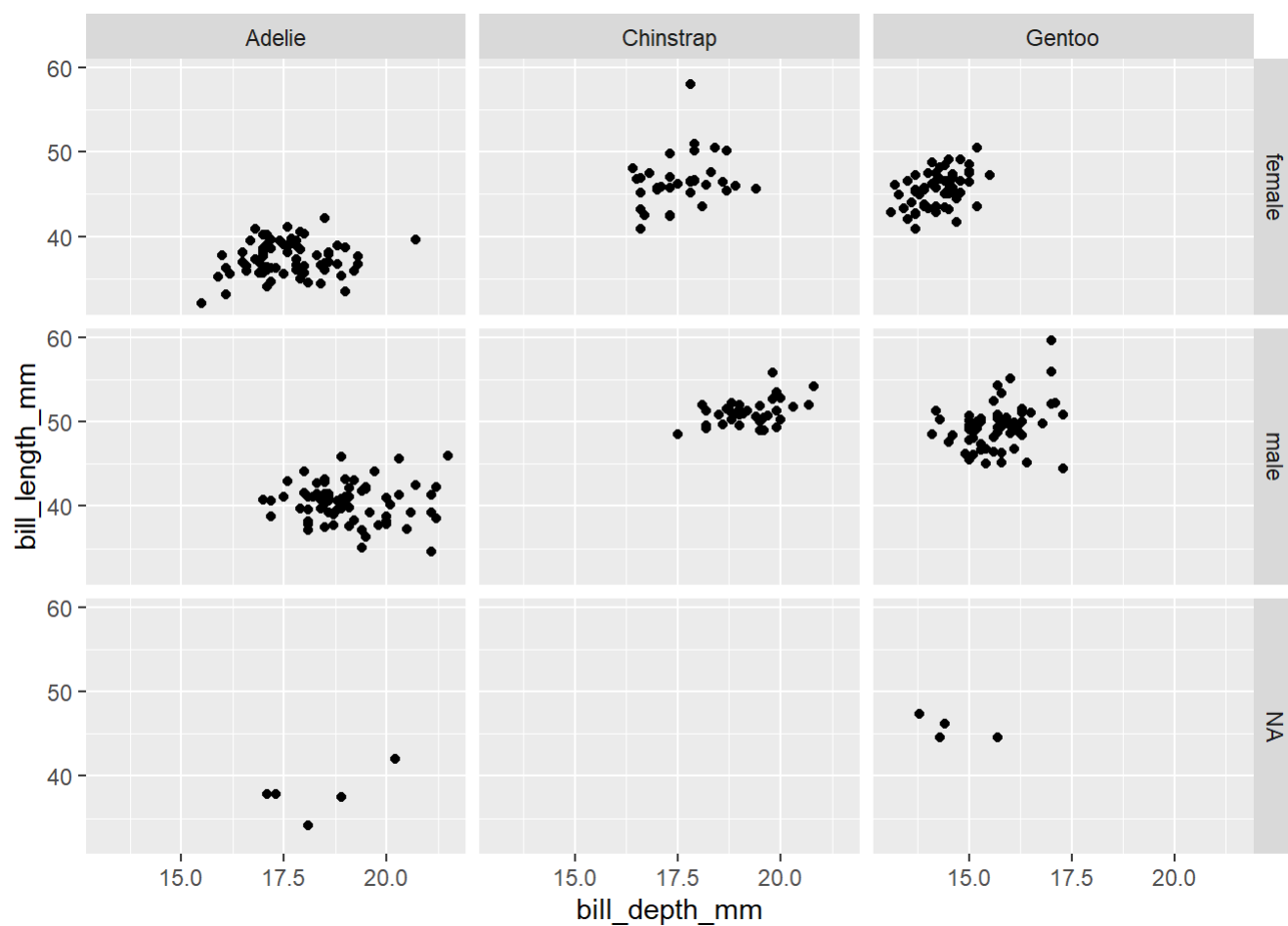
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



3. Alternatively, plotting bill length and bill depth of penguins based on species against sex

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) + geom_point() +  
facet_grid(sex ~ species)
```

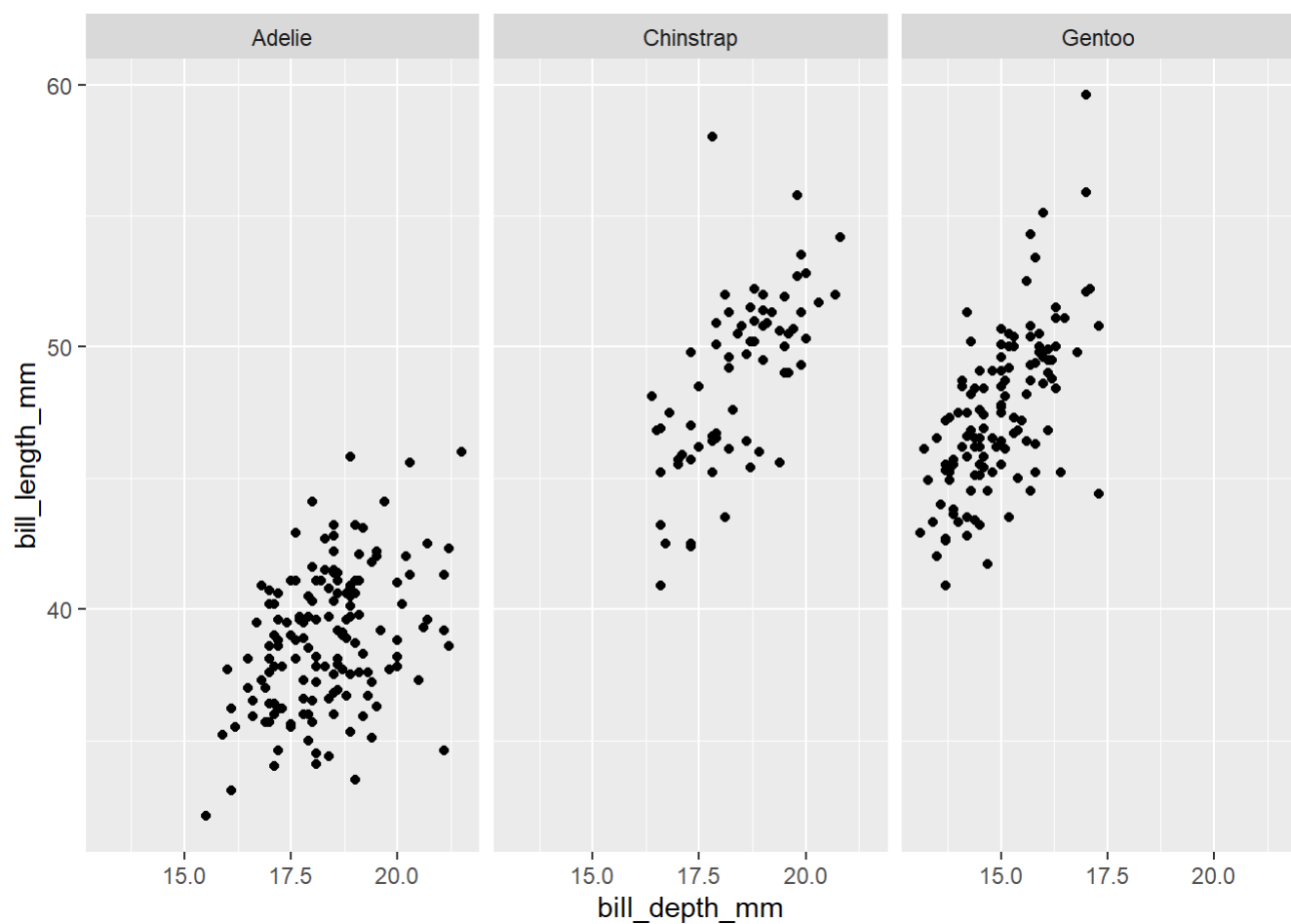
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```

4. Plotting bill length against bill depth only by species

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) + geom_point() +  
facet_wrap(~ species)
```

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

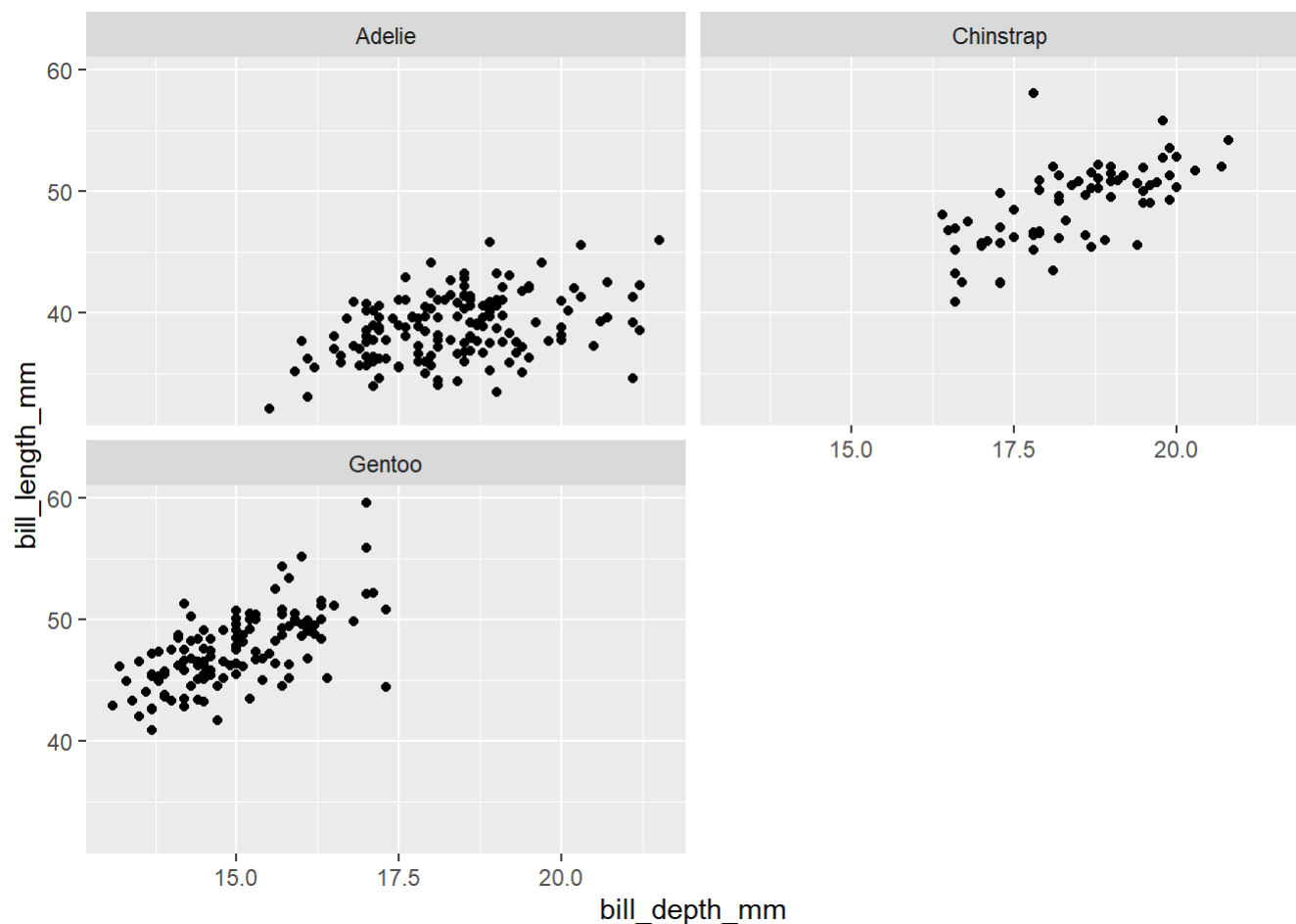


5.

Plotting bill length against bill depth by species, in separate grids

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm)) + geom_point()+  
facet_wrap(~ species, ncol =2)
```

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

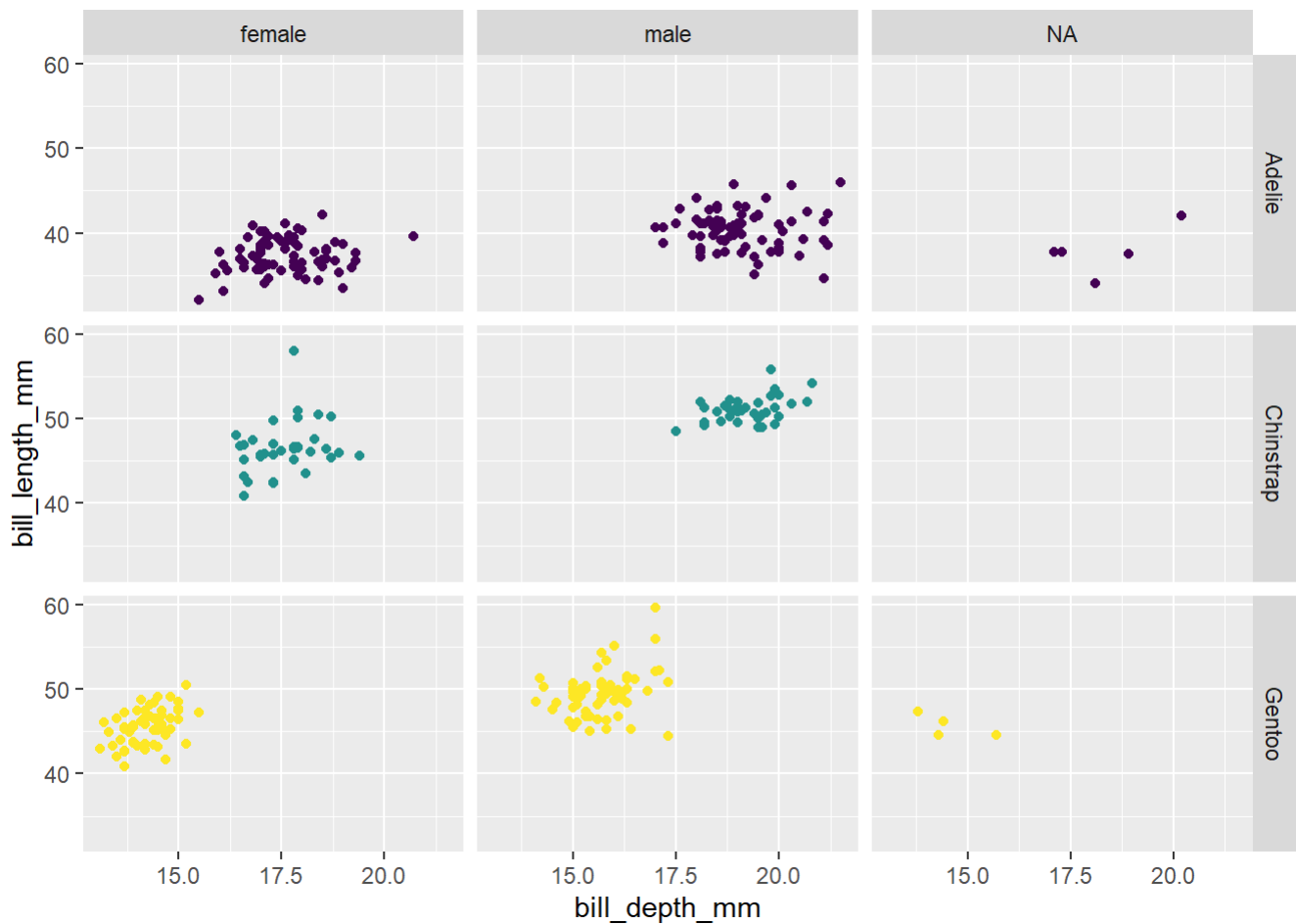


6.

Adding colour (without a legend)

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)) +
  geom_point() + facet_grid(species ~ sex) + scale_color_viridis_d() +
  guides(color = "none")
)
```

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



Visualizing numeric variables using Lending Club Data set

Retrieving dataset

```
install.packages("openintro", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Ariel/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'openintro' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\Ariel\AppData\Local\Temp\RtmpwJTJTP\downloaded_packages
```

```
library(openintro)
```

```
## Warning: package 'openintro' was built under R version 4.2.3
```

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

```
## Warning: package 'airports' was built under R version 4.2.3
```

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

```
## Warning: package 'cherryblossom' was built under R version 4.2.3
```

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

```
## Warning: package 'usdata' was built under R version 4.2.3
```

```
library(tidyverse)  
glimpse(loans_full_schema)
```

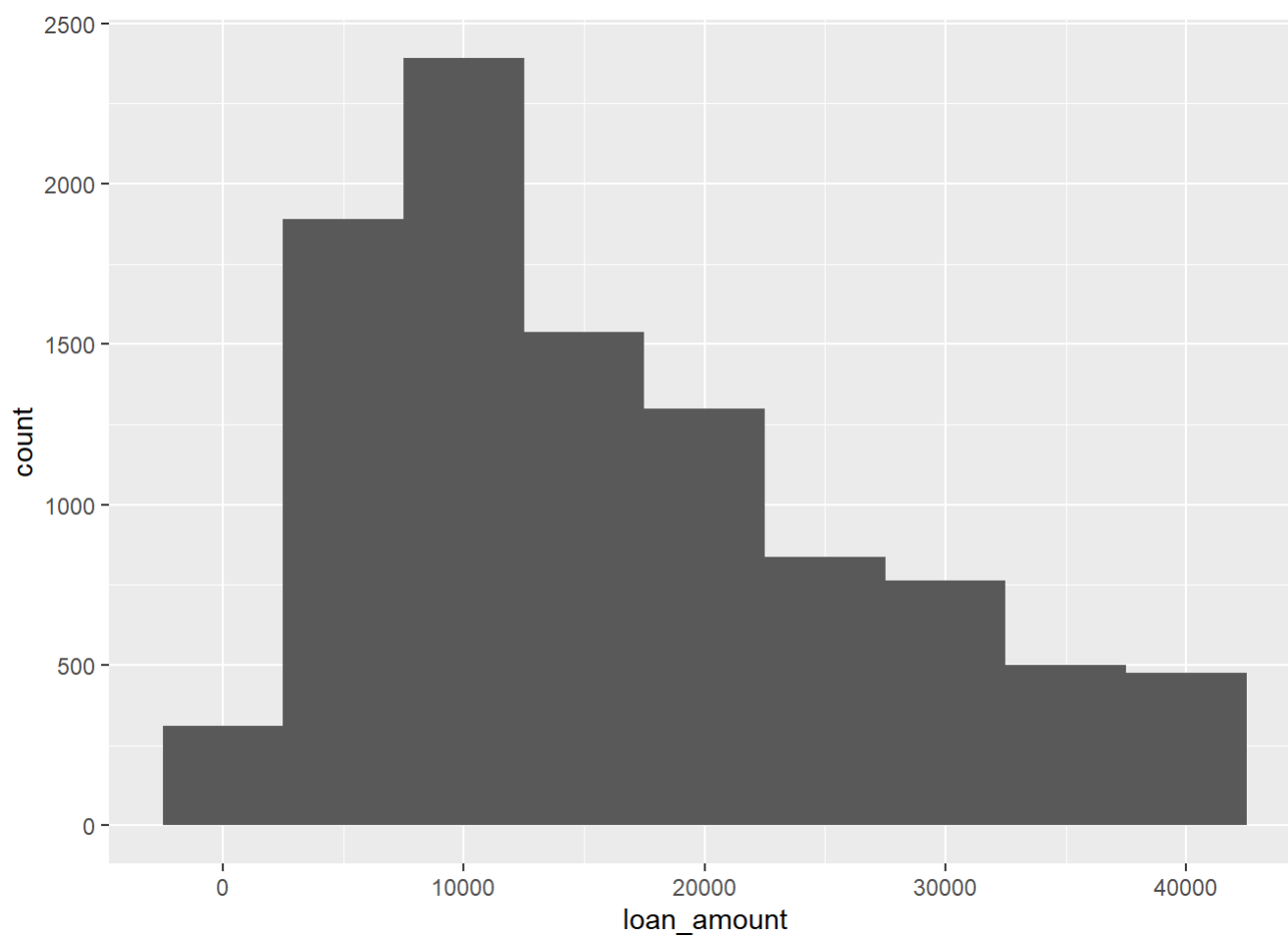
```

## 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...

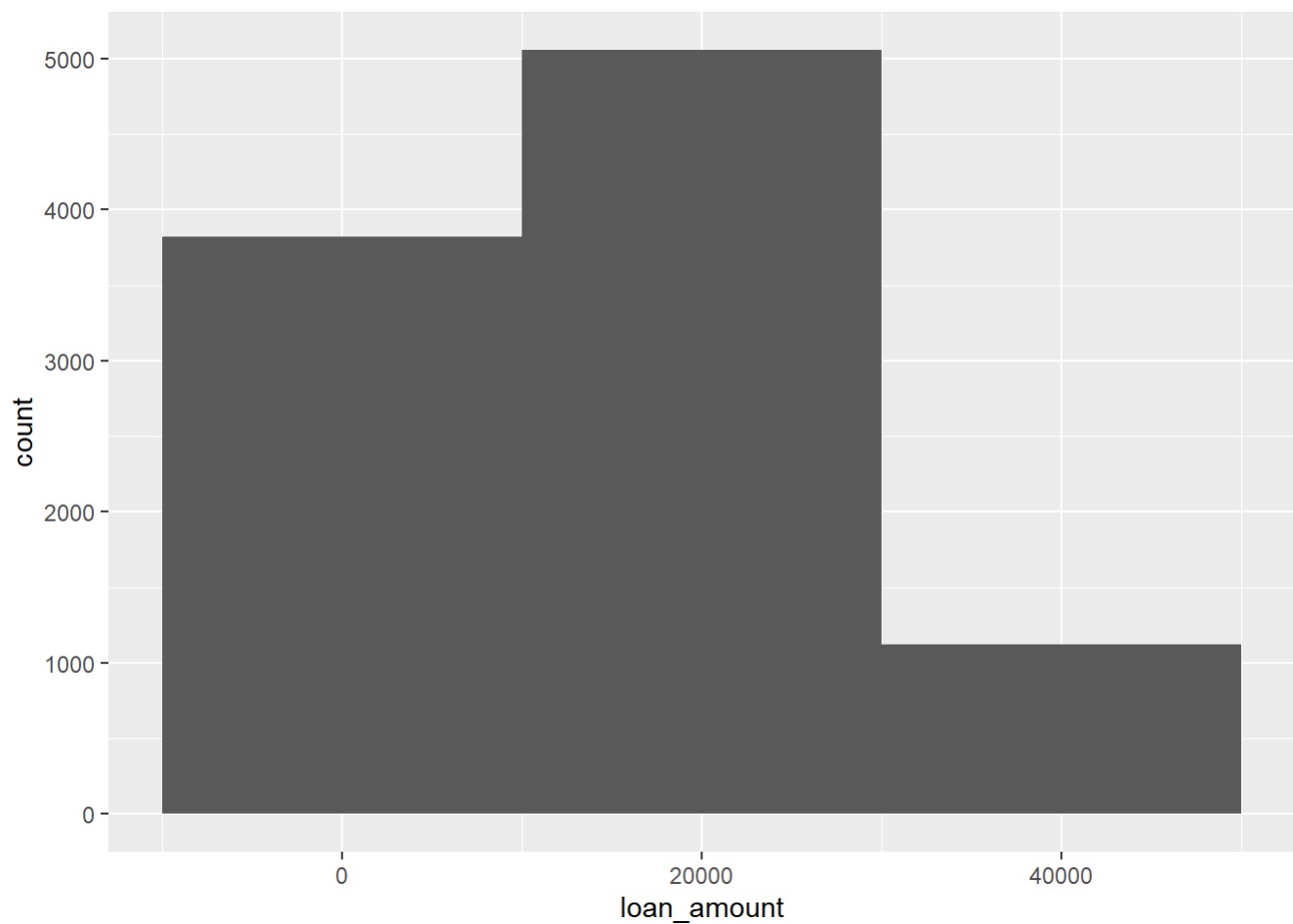
```


2. *binwidth = 5000*

```
ggplot(loans) + aes(x = loan_amount) +  
geom_histogram(binwidth = 5000)
```

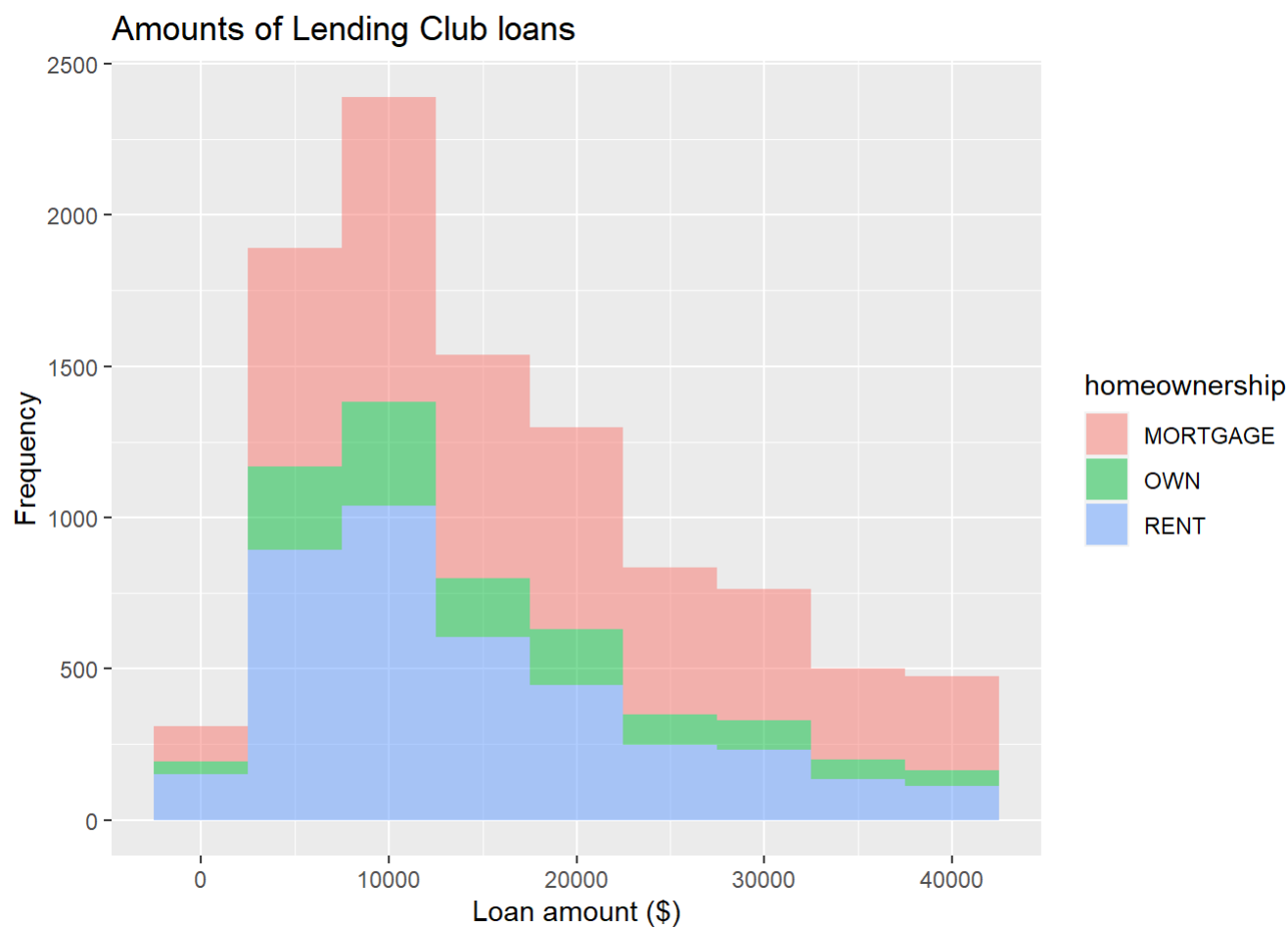
**3. *binwidth = 20000***

```
ggplot(loans) + aes(x = loan_amount) +  
geom_histogram(binwidth = 20000)
```

Customizing histograms, with colors and labels

```
ggplot(loans, aes(x = loan_amount, fill = homeownership)) +  
  geom_histogram(binwidth = 5000, alpha = 0.5) + labs(x = "Loan amount ($)", y = "Frequency", title =  
  "Amounts of Lending Club loans")
```



Creating facet plot of histograms with categorical variable

```
ggplot(loans, aes(x = loan_amount, fill = homeownership)) + geom_histogram(binwidth = 5000) +  
labs(x = "Loan amount ($)", y =  
"Frequency", title = "Amounts of Lending Club loans") + facet_wrap(~ homeownership, nrow = 3)
```

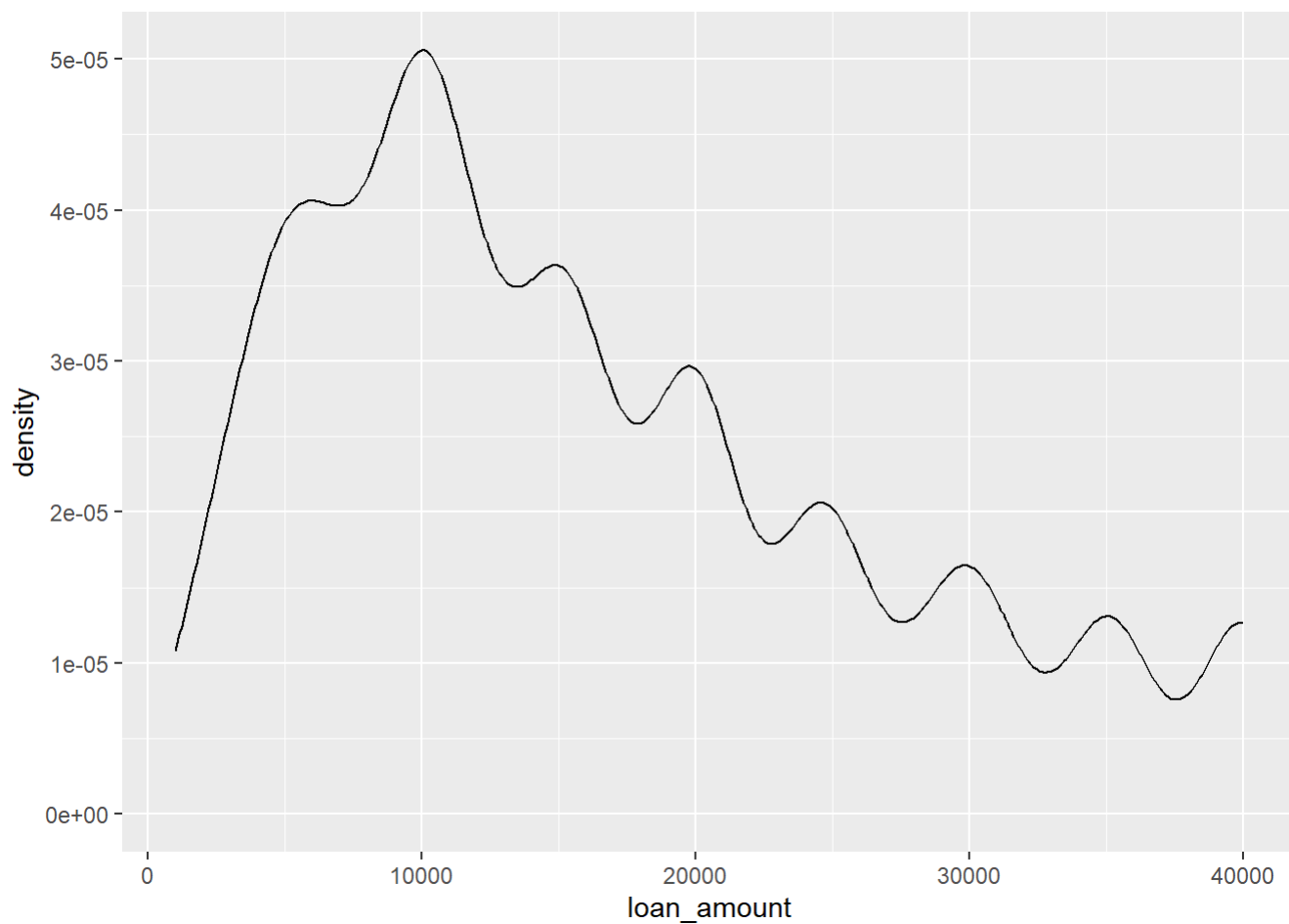
Amounts of Lending Club loans



Creating Density Plot with customizable bandwidth

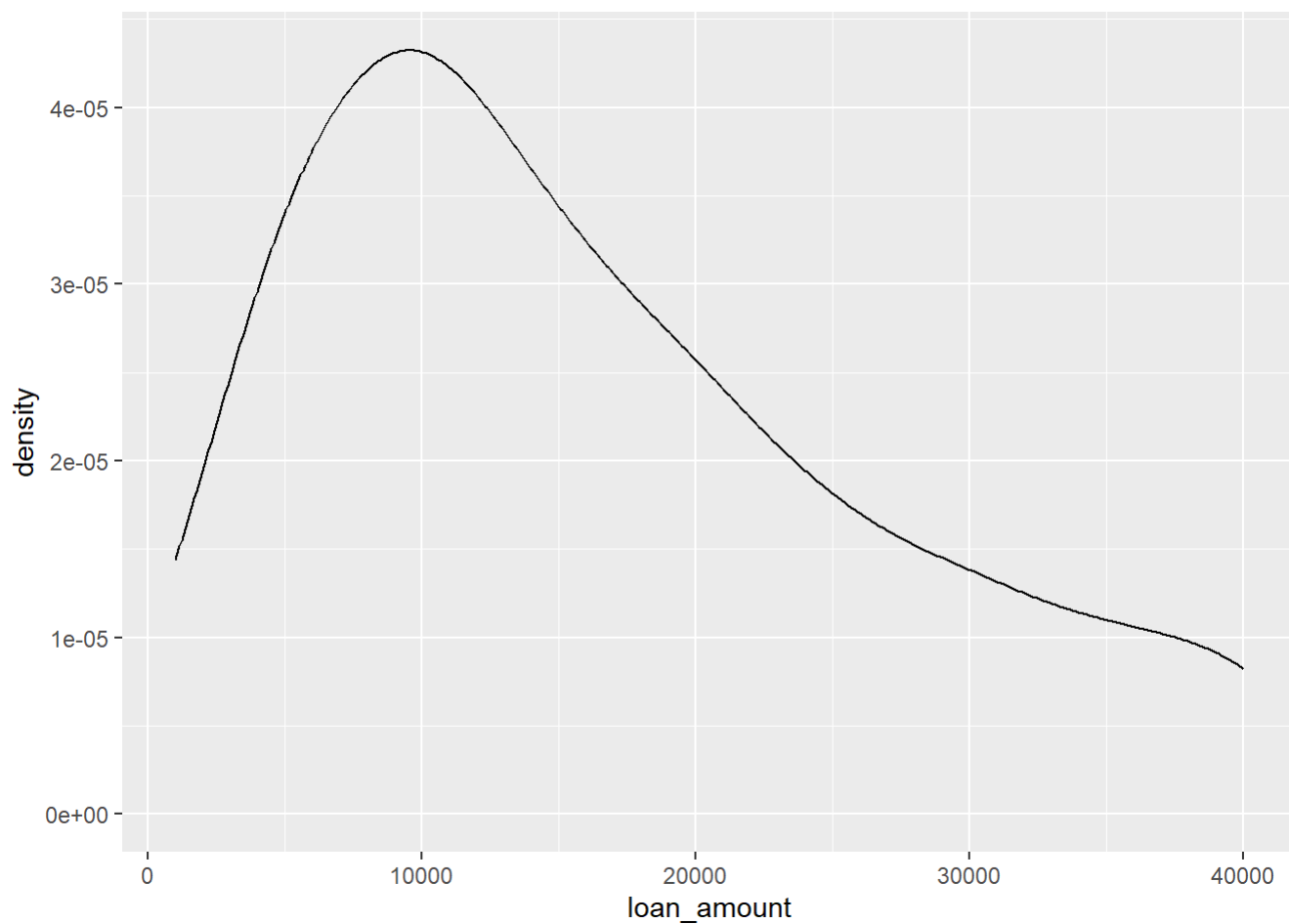
1. Default bandwidth

```
ggplot(loans, aes(x = loan_amount)) +  
  geom_density(adjust=1)
```



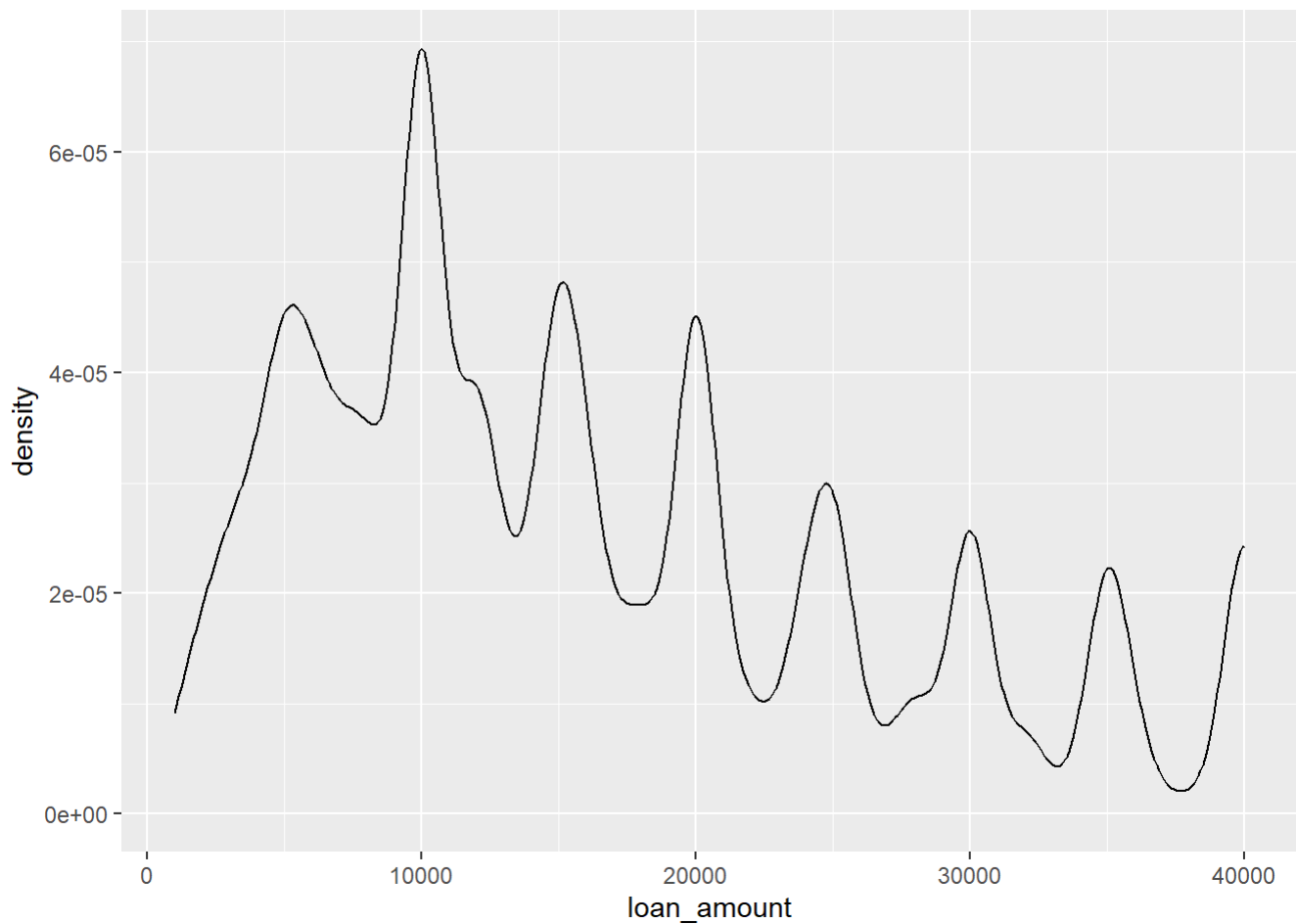
2. Increased bandwidth (2)

```
ggplot(loans, aes(x = loan_amount)) +  
  geom_density(adjust=2)
```



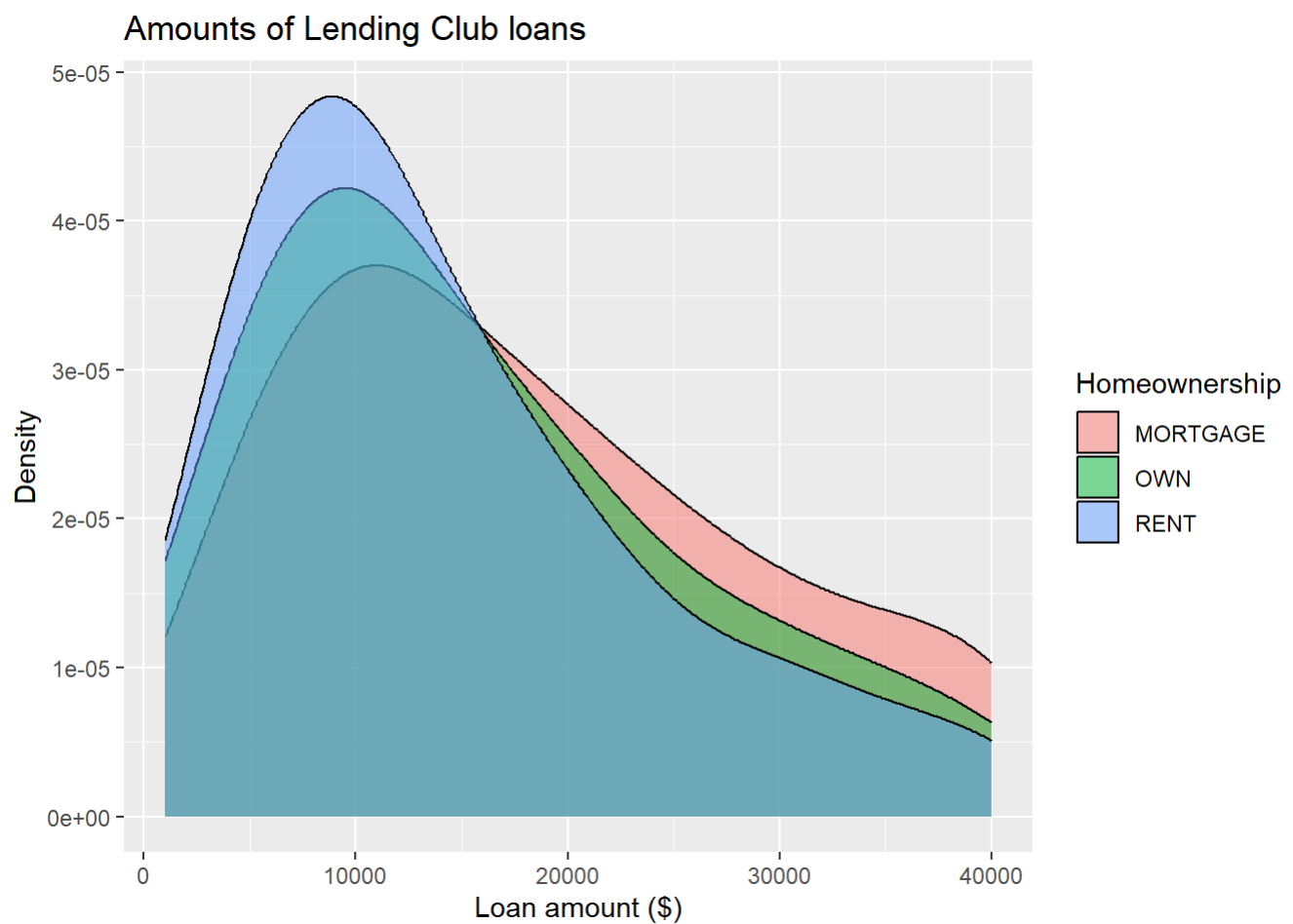
3. Decreased bandwidth (0.5)

```
ggplot(loans, aes(x = loan_amount)) +  
  geom_density(adjust=0.5)
```



Customizing density plot!

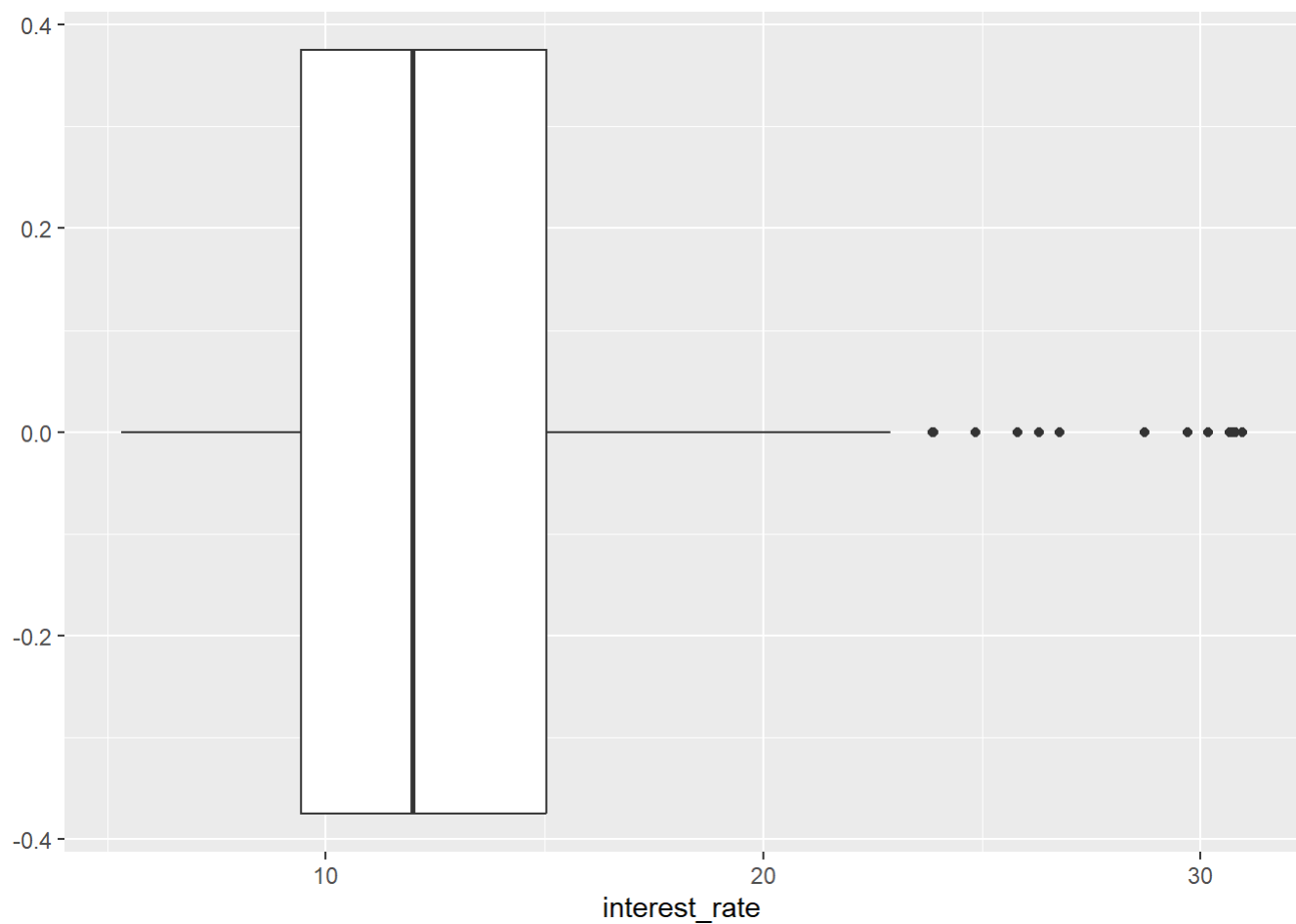
```
ggplot(loans, aes(x = loan_amount, fill = homeownership)) +  
  geom_density(adjust = 2, alpha = 0.5) + labs(x = "Loan amount ($)", y = "Density", title = "Amounts  
of Lending Club loans", fill = "Homeownership")
```



Plotting box plots

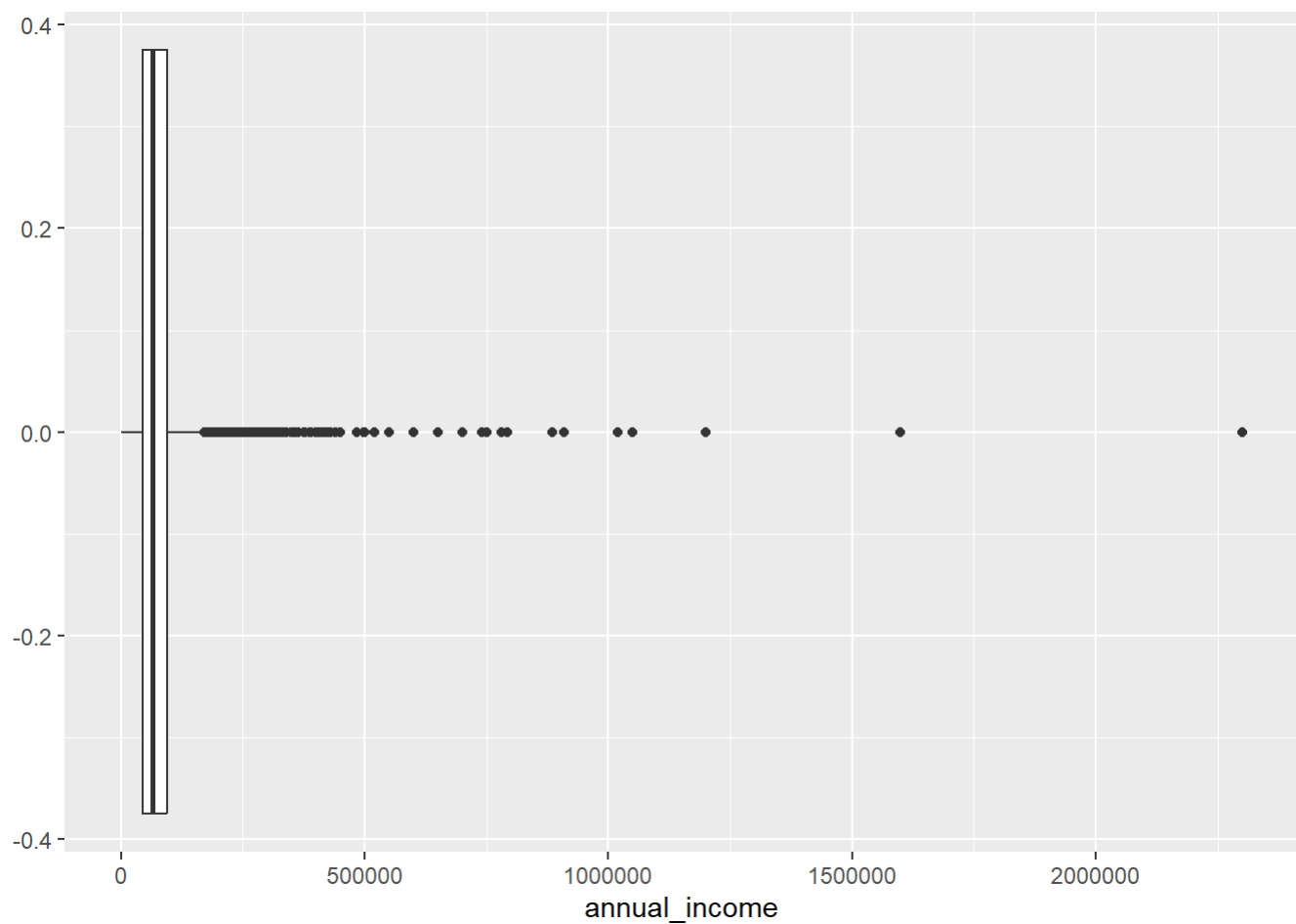
Plotting interest rate of loans

```
ggplot(loans, aes(x = interest_rate)) +geom_boxplot()
```



Plotting annual income against loans

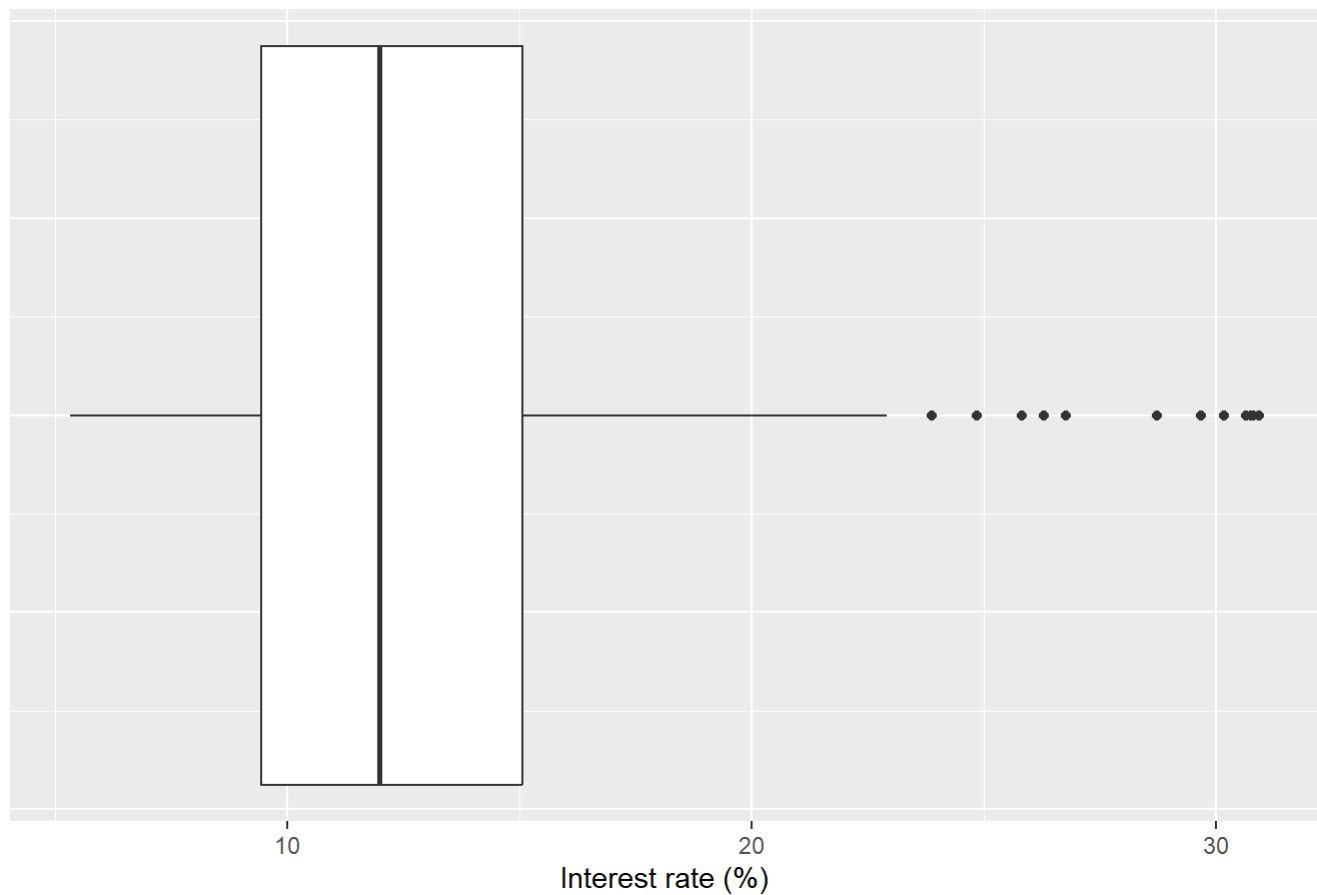
```
ggplot(loans, aes(x = annual_income)) +  
geom_boxplot()
```

Customizing box plots

```
ggplot(loans, aes(x = interest_rate)) +geom_boxplot() +labs(x = "Interest rate (%)", y = NULL, title = "Interest rates of Lending Club loans") +theme( axis.ticks.y = element_blank(), axis.text.y = element_blank() )
```

Interest rates of Lending Club loans

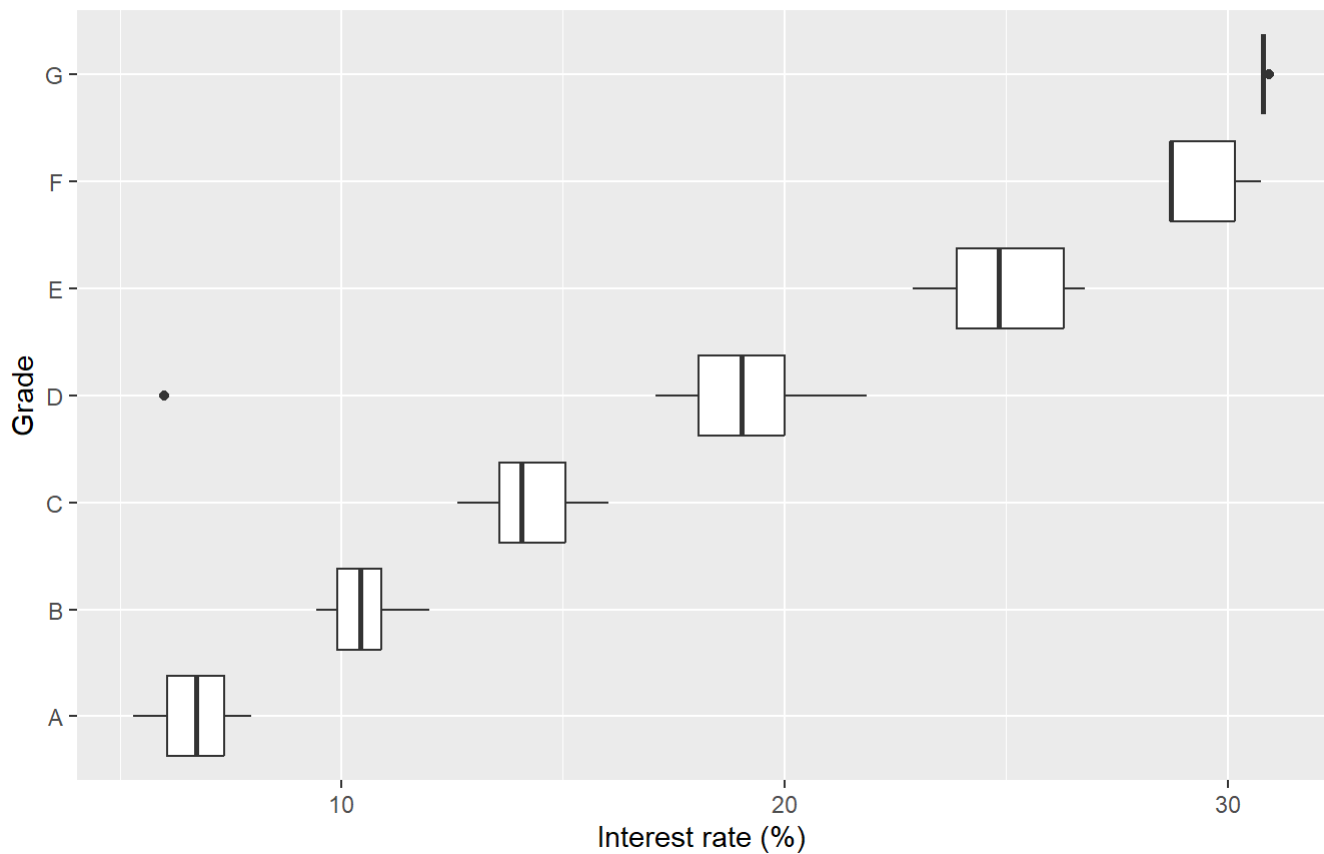


By grade of loan

```
ggplot(loans, aes(x = interest_rate, y = grade)) + geom_boxplot() + labs(x =  
"Interest rate (%)", y = "Grade", title = "Interest rates of Lending Club loans", subtitle = "by gr  
ade of loan")
```

Interest rates of Lending Club loans

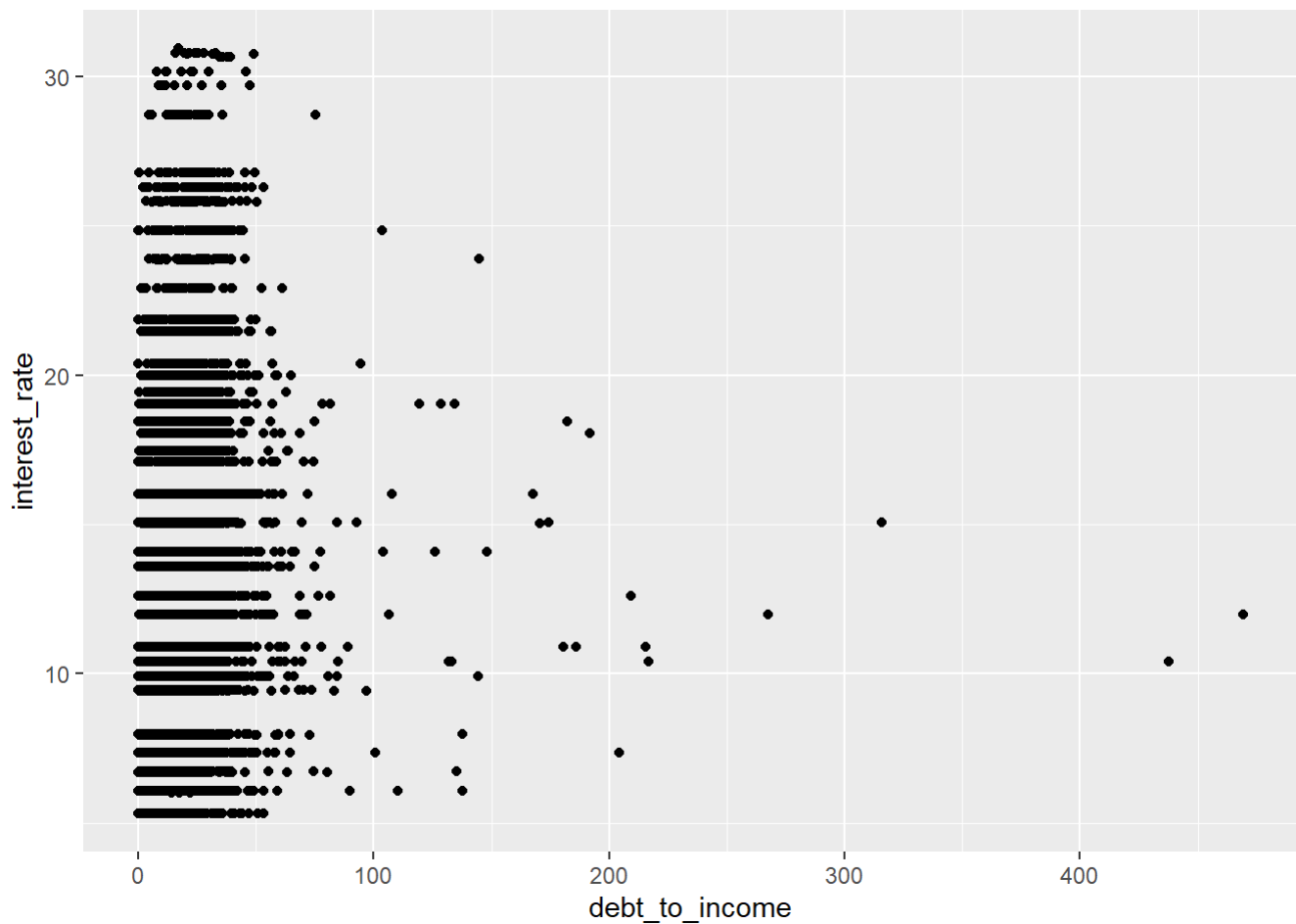
by grade of loan



Creating Scatterplot

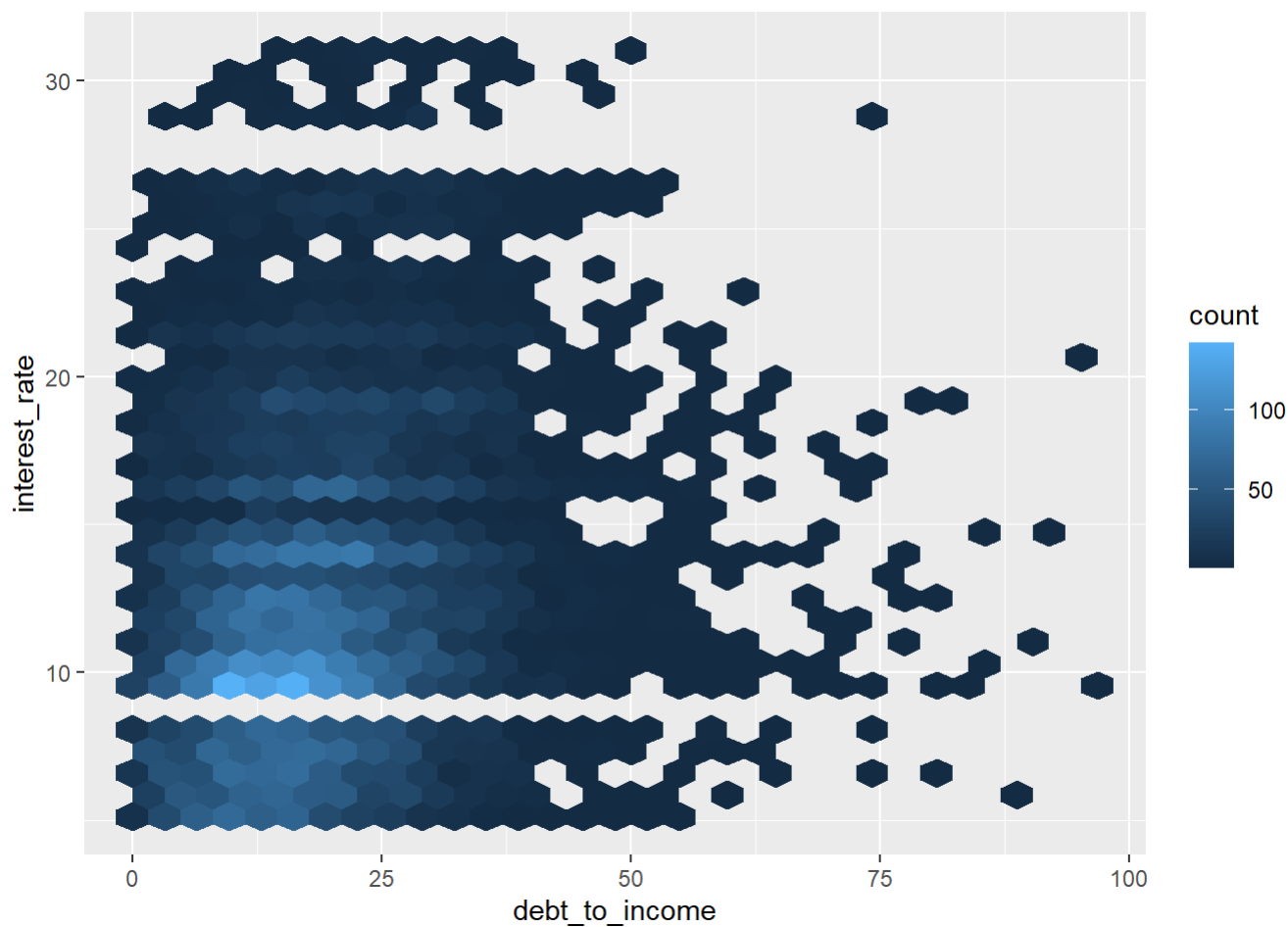
```
ggplot(loans, aes(x = debt_to_income, y = interest_rate)) +  
geom_point()
```

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



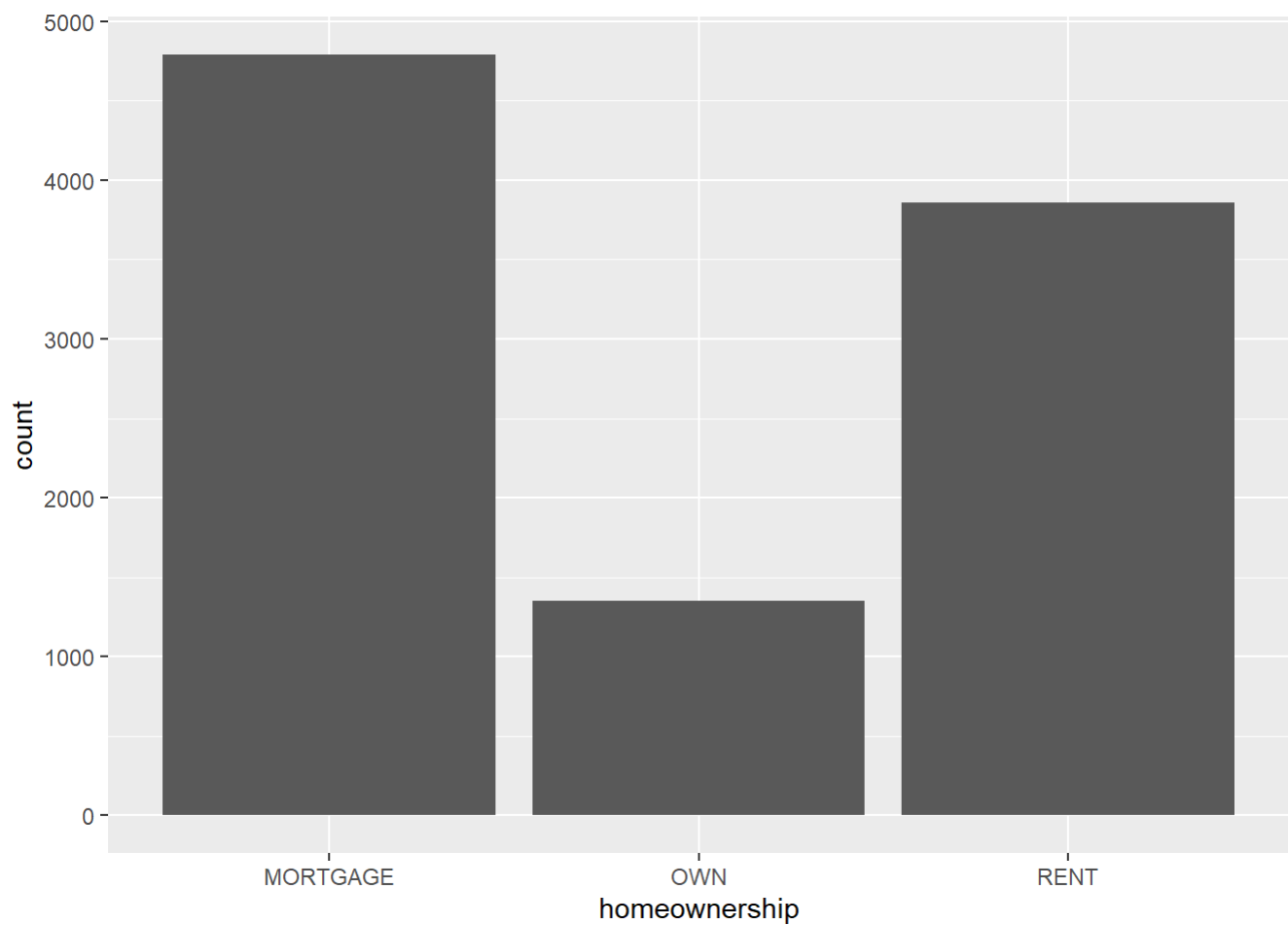
Creating Hex Plot

```
ggplot(loans %>% filter(debt_to_income <100), aes(x = debt_to_income, y = interest_rate)) + geom_hex()
```



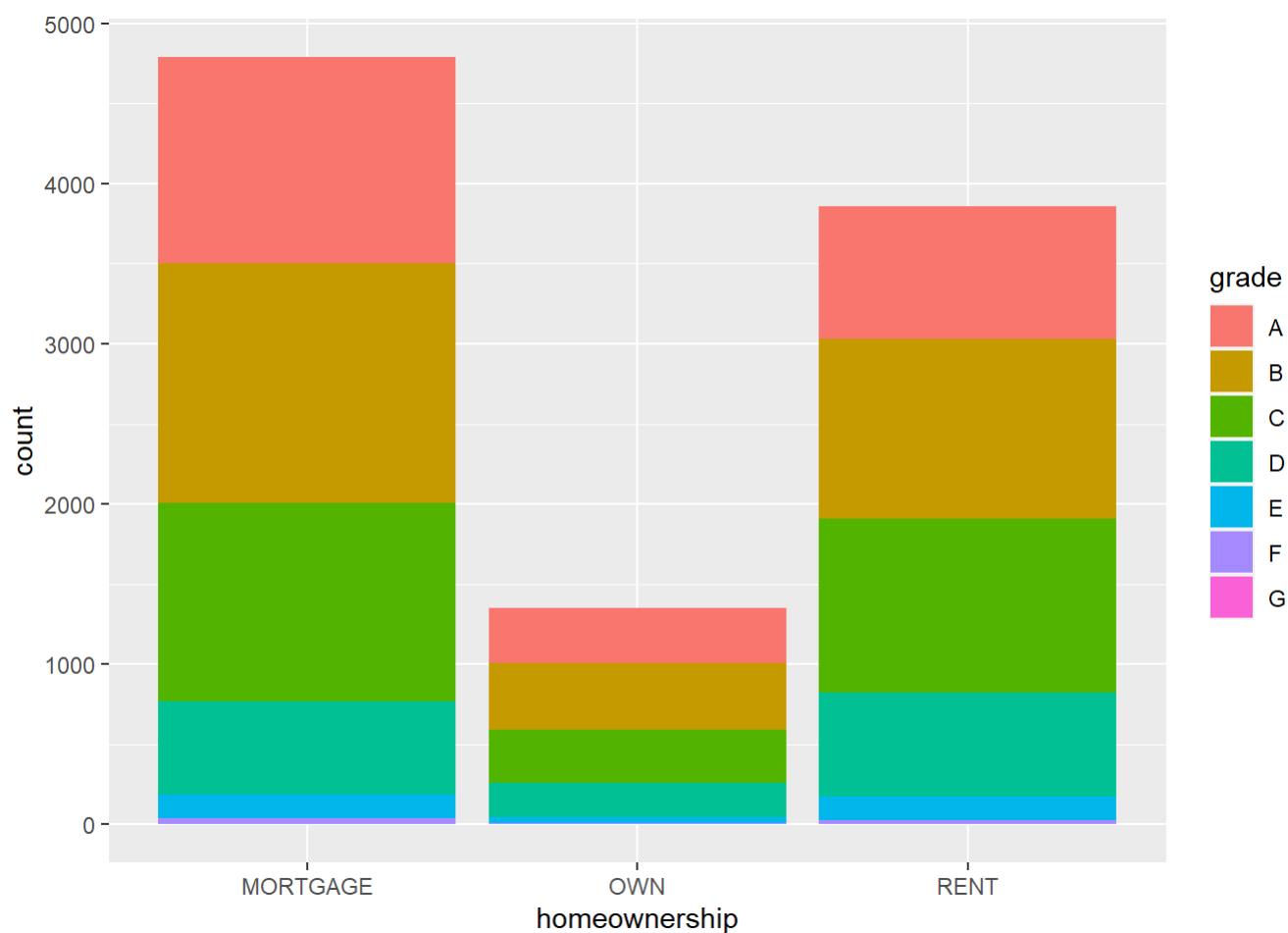
Creating Bar Plot

```
ggplot(loans, aes(x = homeownership)) +geom_bar()
```



Customizing segmented bar plot (by grade of homeownership)

```
ggplot(loans, aes(x = homeownership,  
fill = grade)) +  
geom_bar()
```



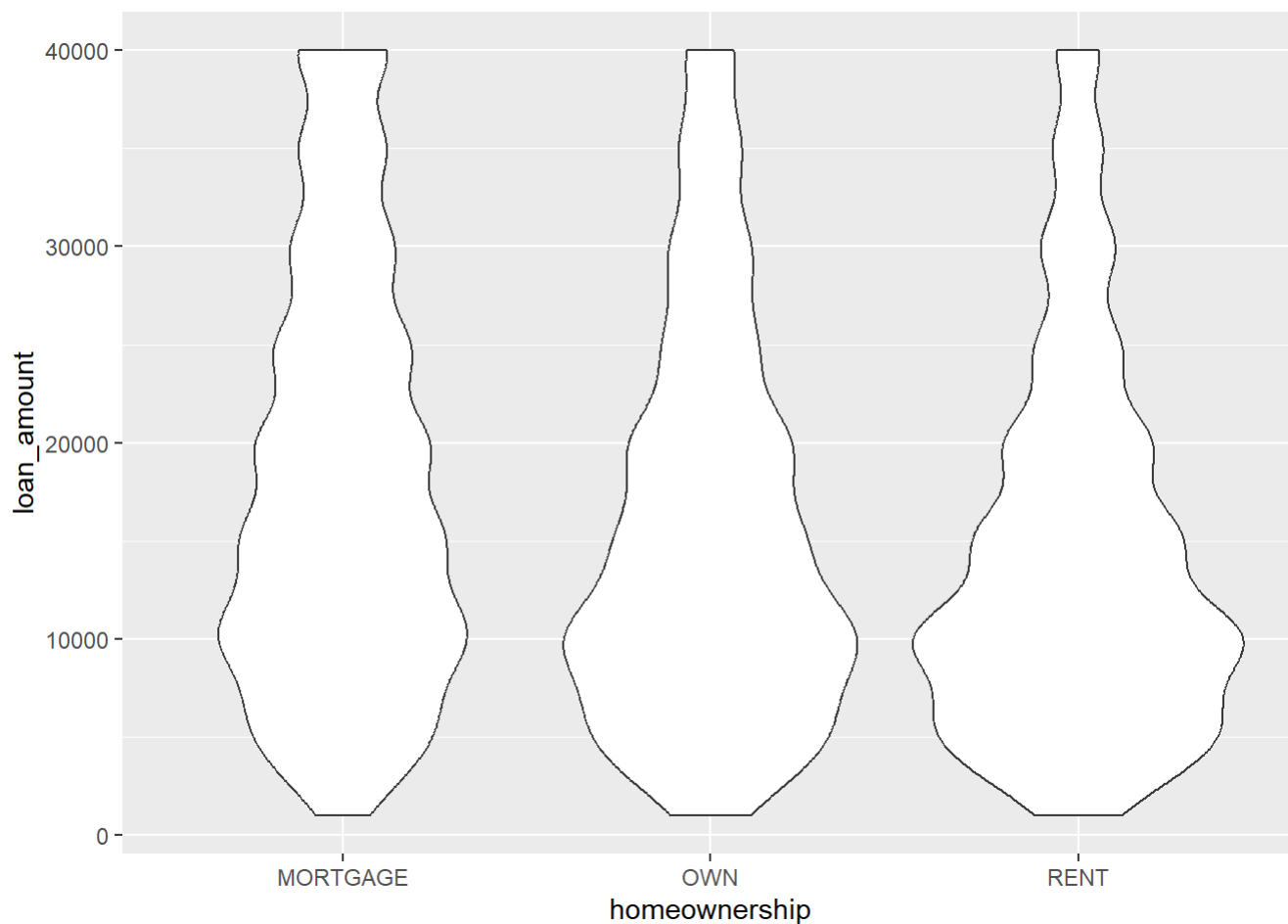
Customizing bar plots (by proportion)

```
ggplot(loans, aes(y = homeownership, fill = grade)) + geom_bar(position =  
"fill") +labs( x ="Proportion", y ="Homeownership", fill ="Grade", title ="Grades of Lending  
Club loans", subtitle="and homeownership of lendeer")
```



Creating violin plots

```
ggplot(loans, aes(x = homeownership, y = loan_amount)) +  
  geom_violin()
```

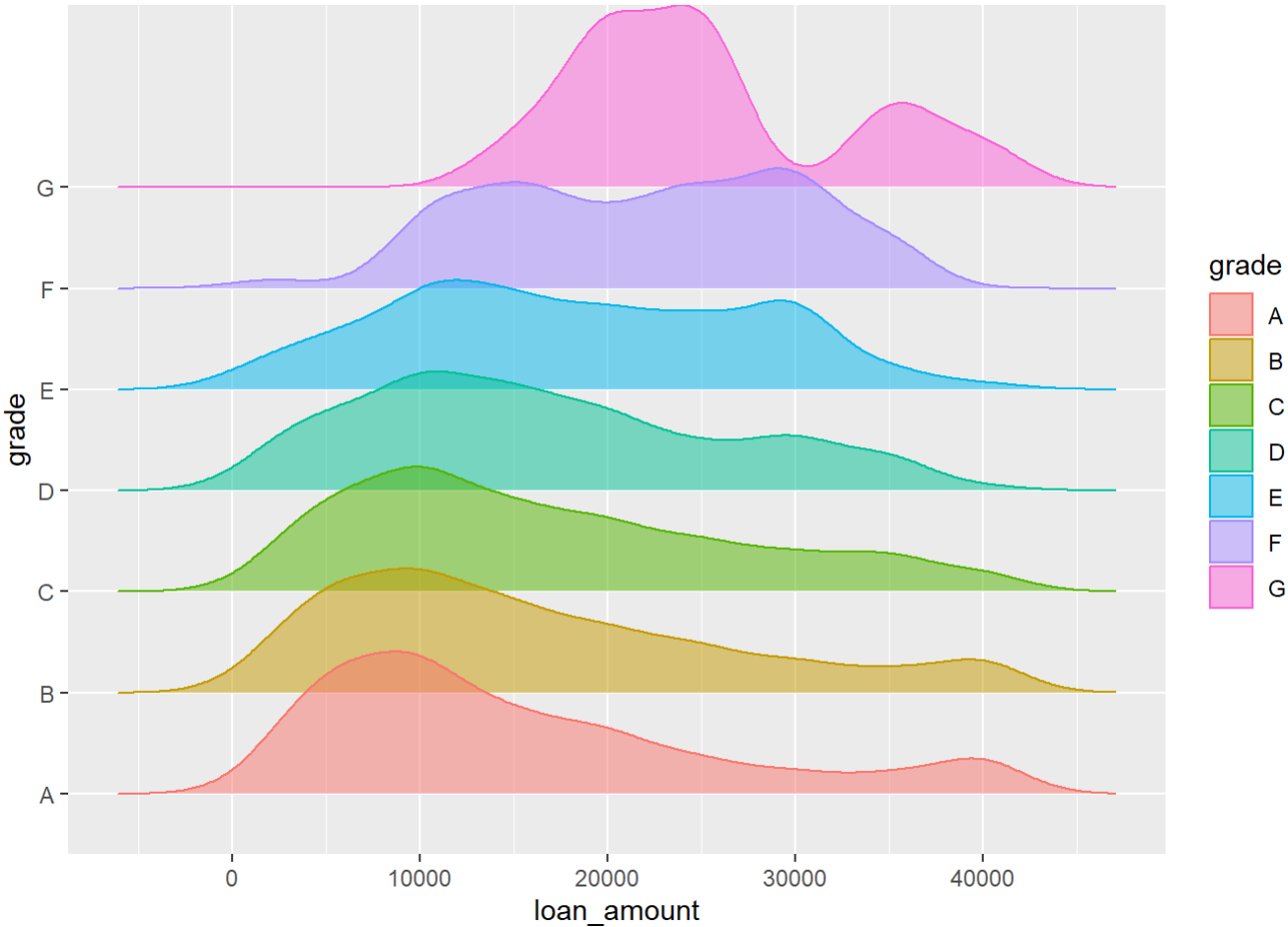
Creating ridge plots

```
library(ggribes)
```

```
## Warning: package 'ggribes' was built under R version 4.2.3
```

```
ggplot(loans, aes(x = loan_amount, y = grade, fill = grade, color = grade)) +geom_density_ridges(alpha =0.5)
```

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



Thank you!