Week 7 Challenge

Wong Zi Xin

2023-10-02

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:

II. Code to edit and execute using the Code-along-6.Rmd file

```
# Loading the package and dataset
# Enter code here
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
```

```
penguins <- read_csv("penguins.csv")
```

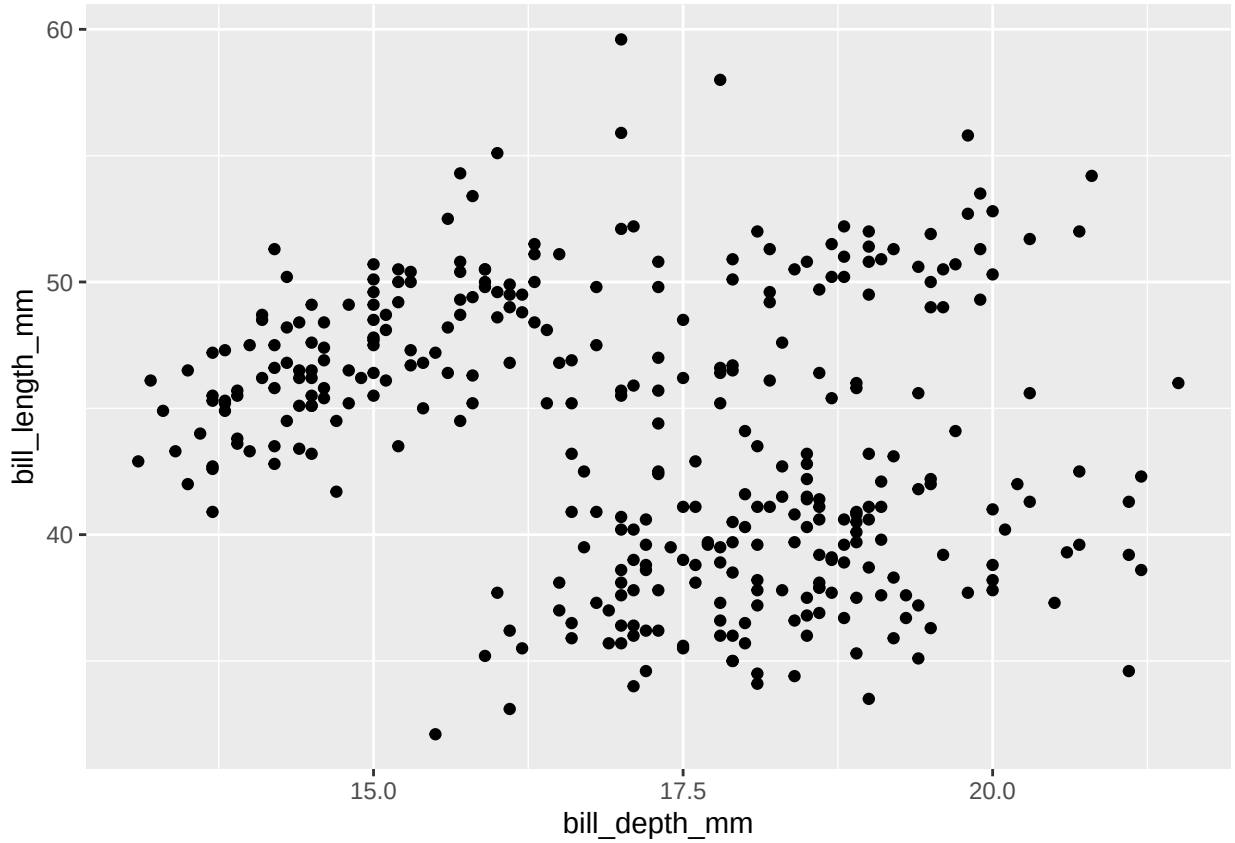
```
## Rows: 344 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): species, island, sex
## dbl (6): rowid, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(penguins)
```

```
## Rows: 344
## Columns: 9
## $ rowid             <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ species           <chr> "Adelie", "Adelie", "Adelie", "Adelie", "Adelie", "A~
## $ island            <chr> "Torgersen", "Torgersen", "Torgersen", "Torgersen", ~
## $ 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 <dbl> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <dbl> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <chr> "male", "female", "female", NA, "female", "male", "f~
## $ year              <dbl> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

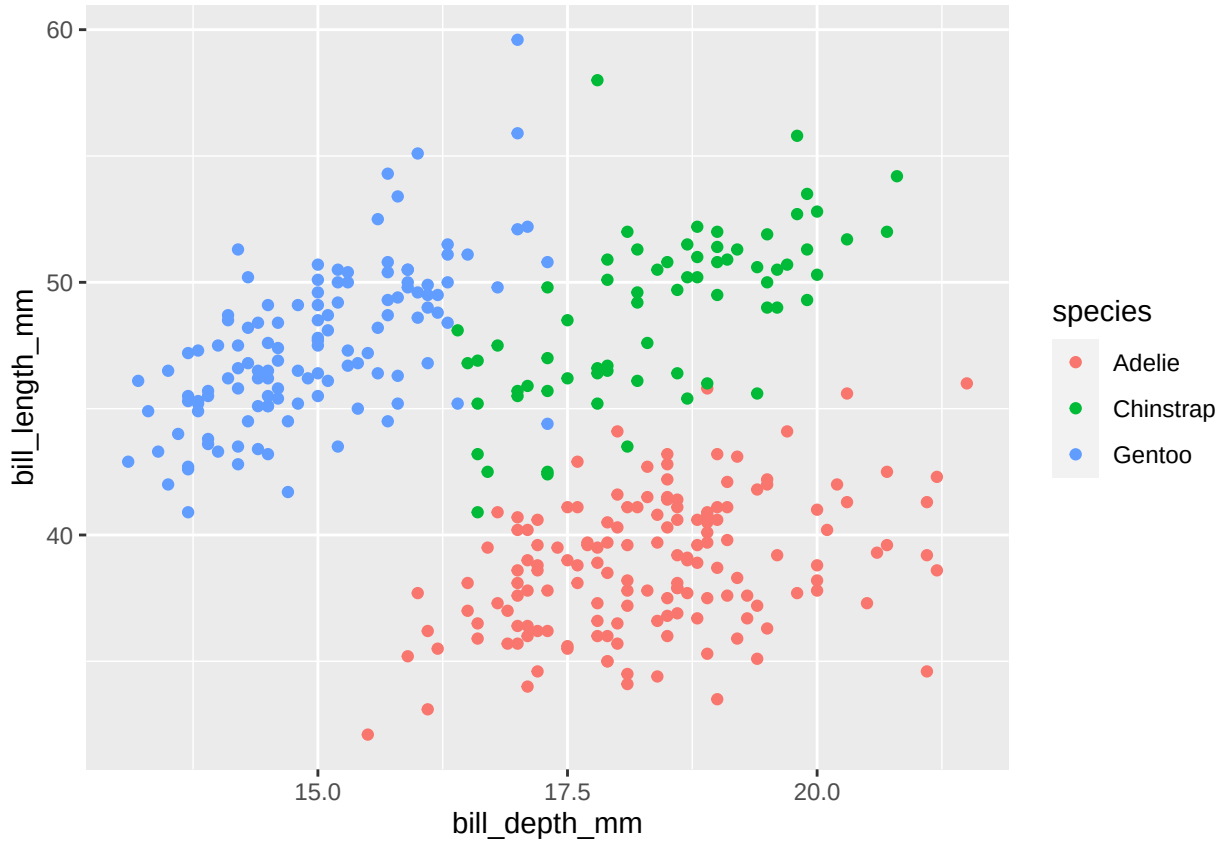
```
# plotting a graph to interpret the relationship between bill length and bill depth
# Enter code here
ggplot(data=penguins,
       mapping = aes(x=bill_depth_mm,
                     y=bill_length_mm)) +
  geom_point()
```

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



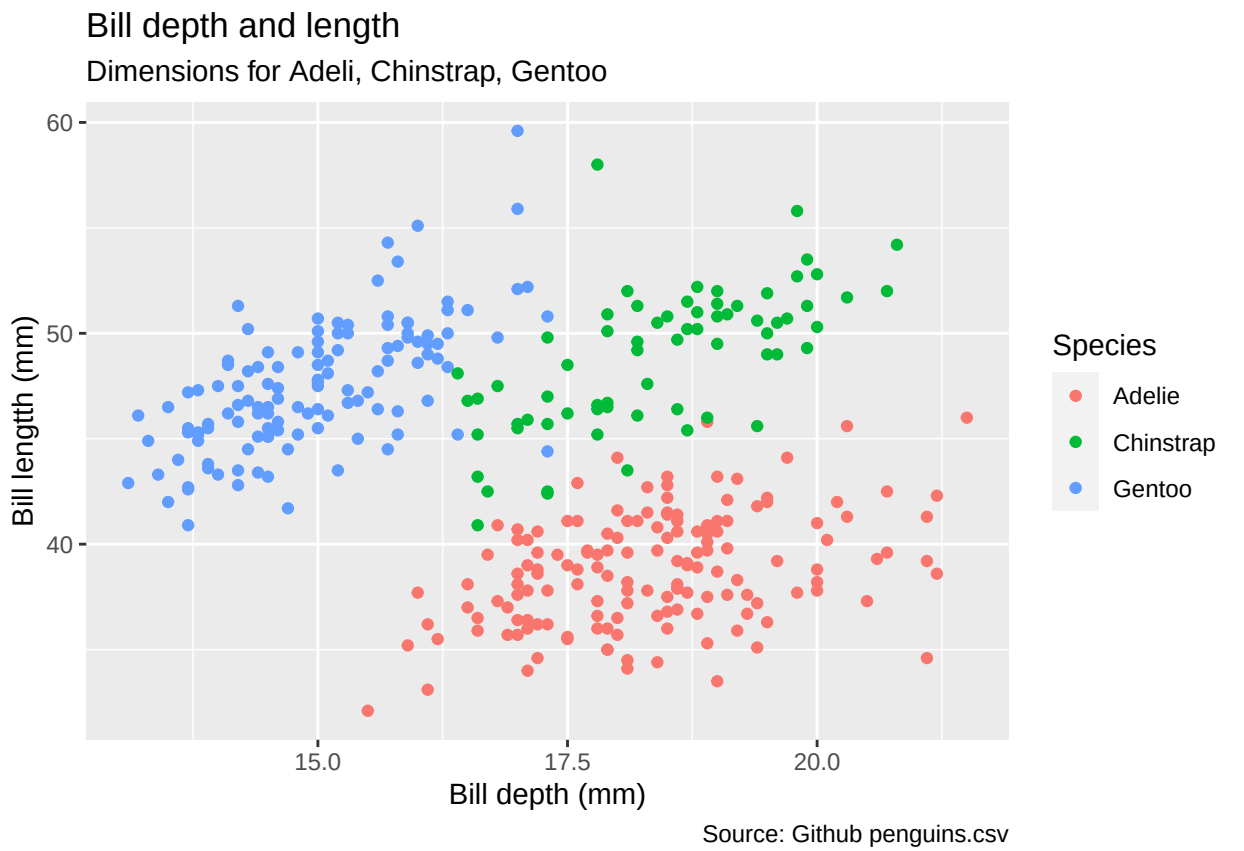
```
# plotting the same graph above but mapping species to the colour of each point
# Enter code here
ggplot(data=penguins,
       mapping = aes(x=bill_depth_mm,
                     y=bill_length_mm,
                     colour = species)) +
  geom_point()
```

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



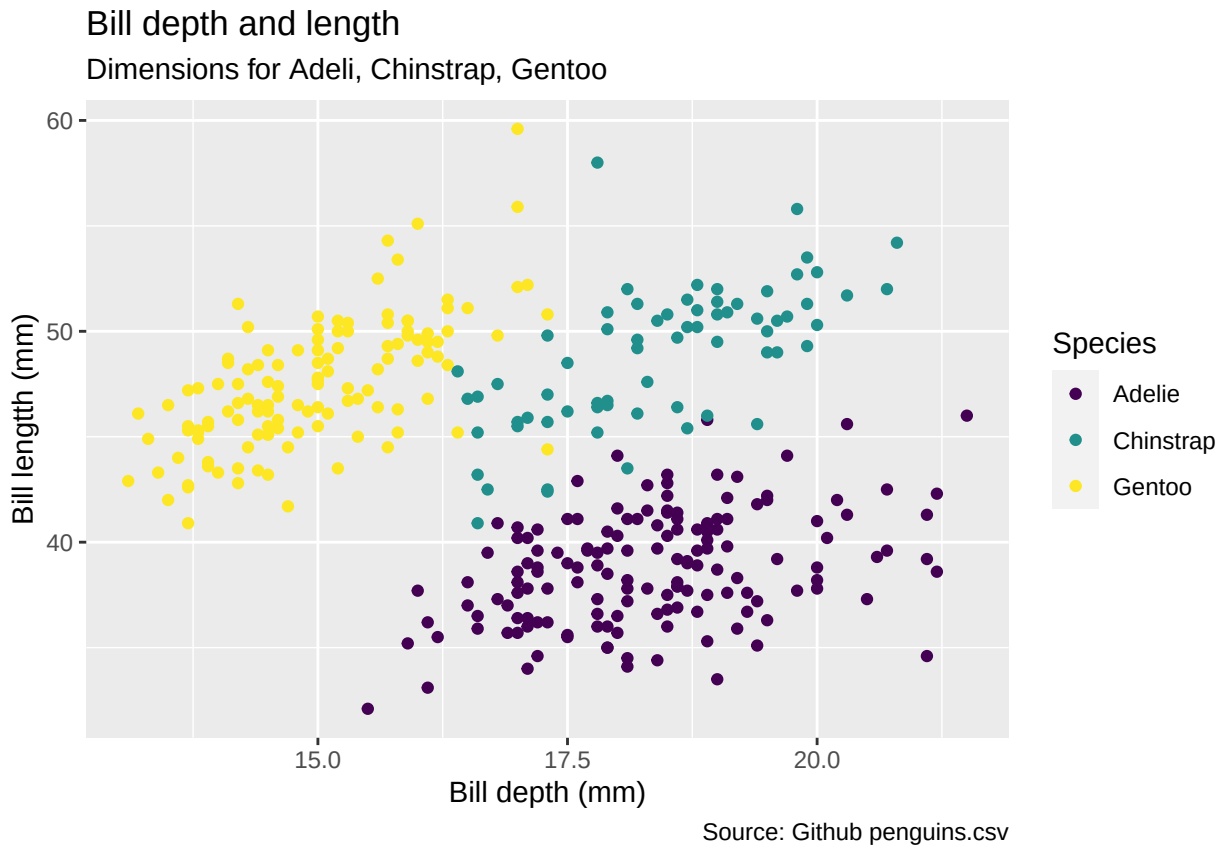

```
# Add title, subtitle, label axes, label legend, add caption
# Enter code here
ggplot(data=penguins,
       mapping = aes(x=bill_depth_mm,
                     y=bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adeli, Chinstrap, Gentoo",
       x = "Bill depth (mm)", y = "Bill lengtth (mm)",
       colour = "Species",
       caption = "Source: Github penguins.csv")
```

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



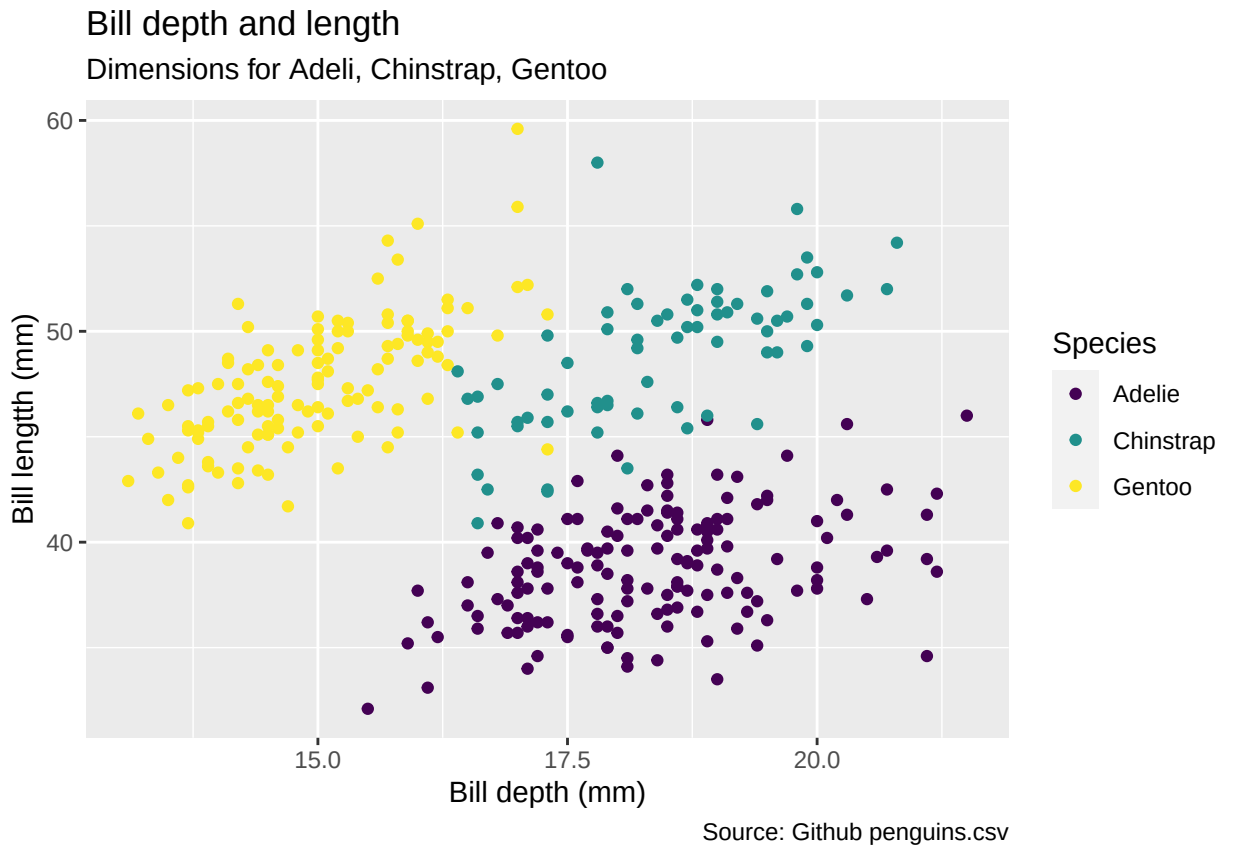

```
# Using a discrete colour scale
# Enter code here
ggplot(data=penguins,
       mapping = aes(x=bill_depth_mm,
                     y=bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adeli, Chinstrap, Gentoo",
       x = "Bill depth (mm)", y = "Bill lengtth (mm)",
       colour = "Species",
       caption = "Source: Github penguins.csv") +
scale_colour_viridis_d()
```

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



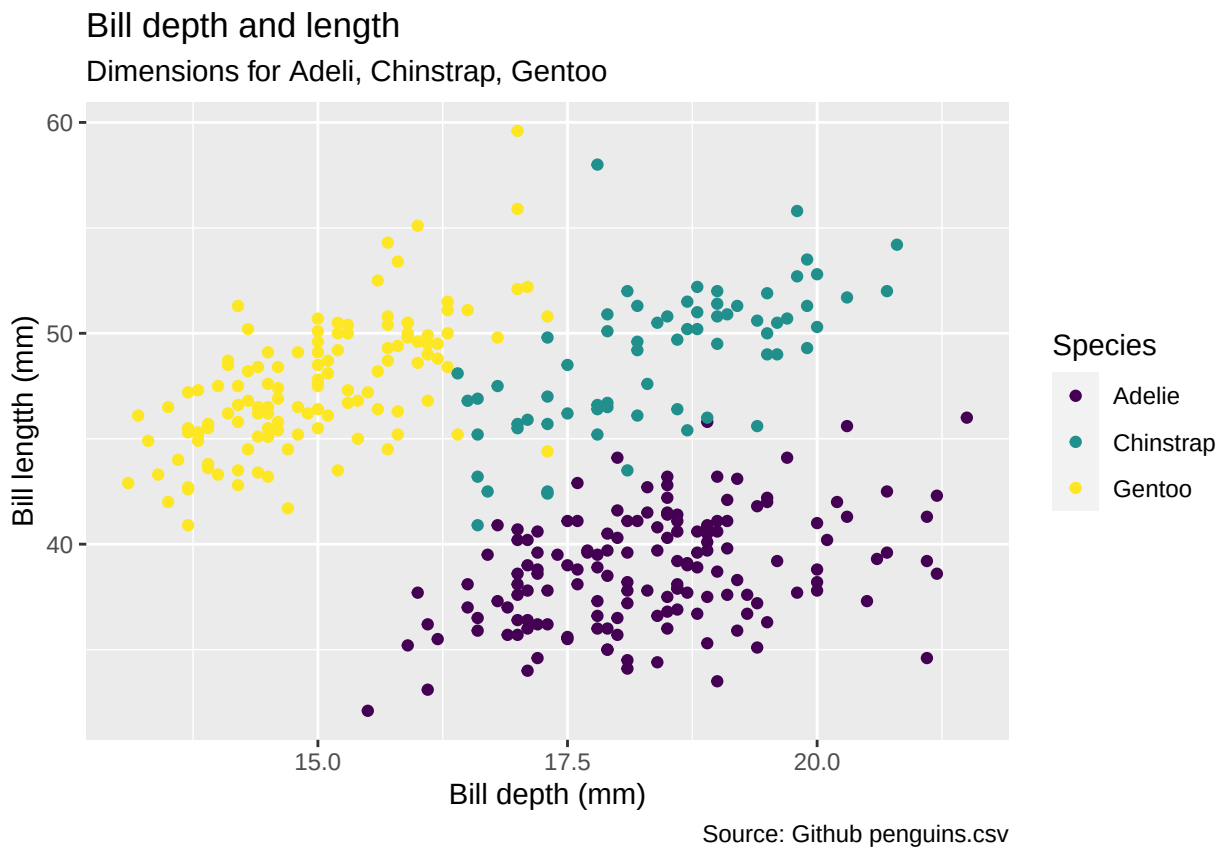

```
# Alternative ways to plot graph
# Enter code here
ggplot(penguins,
       aes(x=bill_depth_mm,
                     y=bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adeli, Chinstrap, Gentoo",
       x = "Bill depth (mm)", y = "Bill lengtth (mm)",
       colour = "Species",
       caption = "Source: Github penguins.csv") +
scale_colour_viridis_d()
```

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



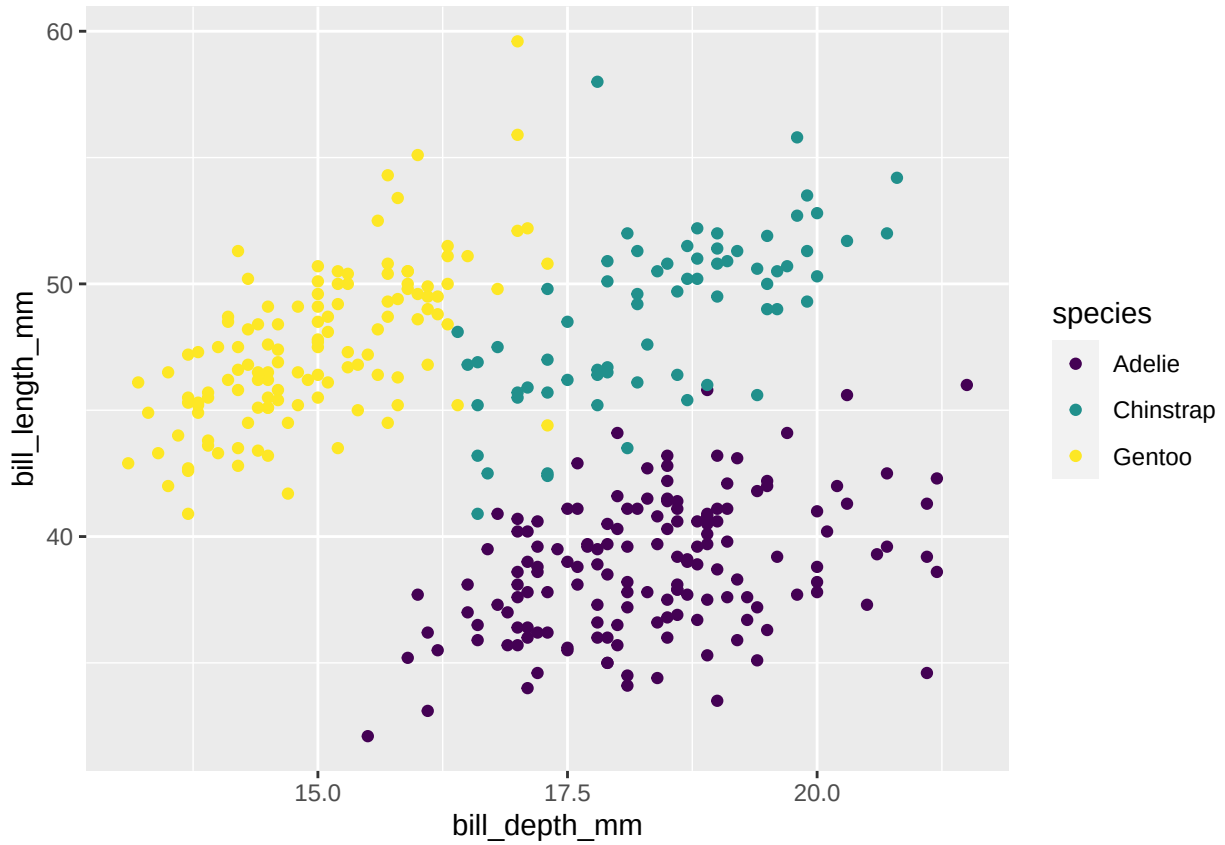

```
# Alternative ways to plot graph
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adeli, Chinstrap, Gentoo",
       x = "Bill depth (mm)", y = "Bill lengtth (mm)",
       colour = "Species",
       caption = "Source: Github penguins.csv") +
scale_colour_viridis_d()
```

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



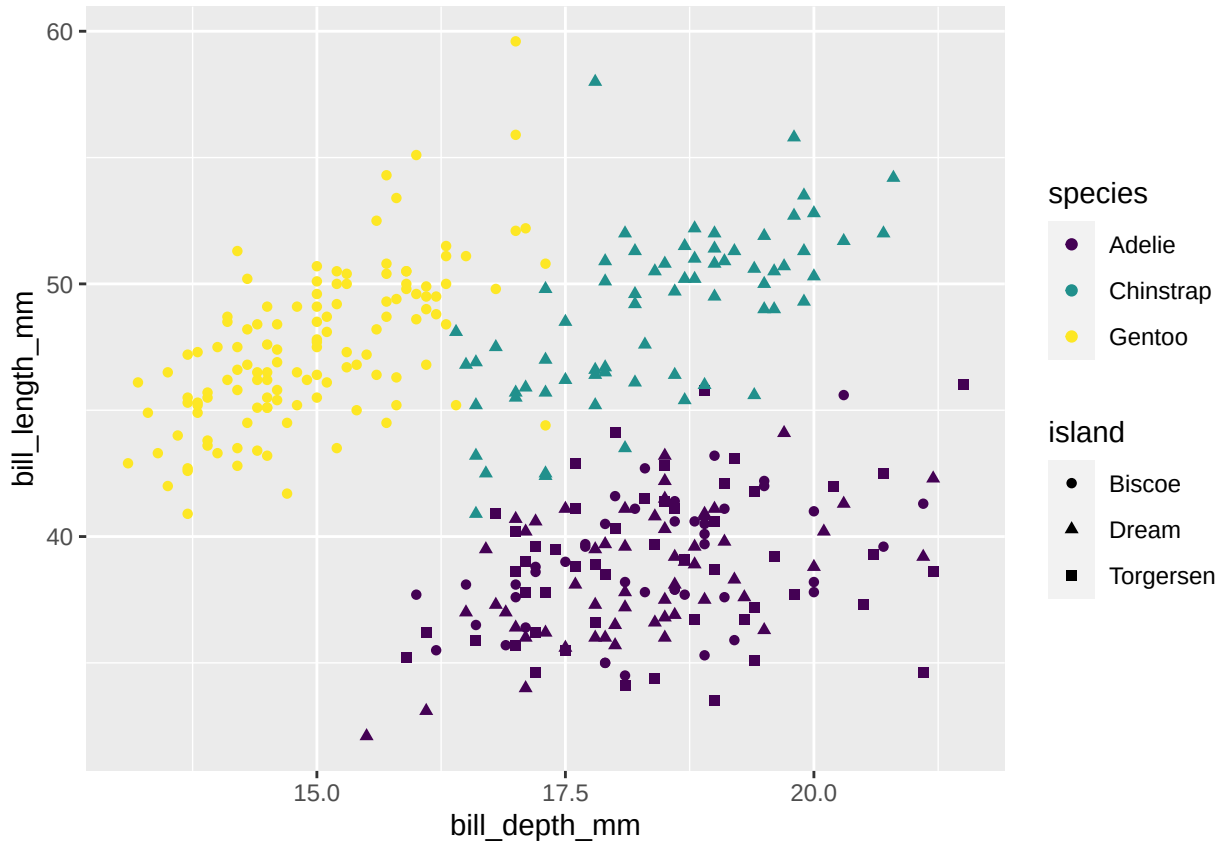

```
# Mapping colour to a specific variable
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species) +
  geom_point() + scale_colour_viridis_d()
```

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



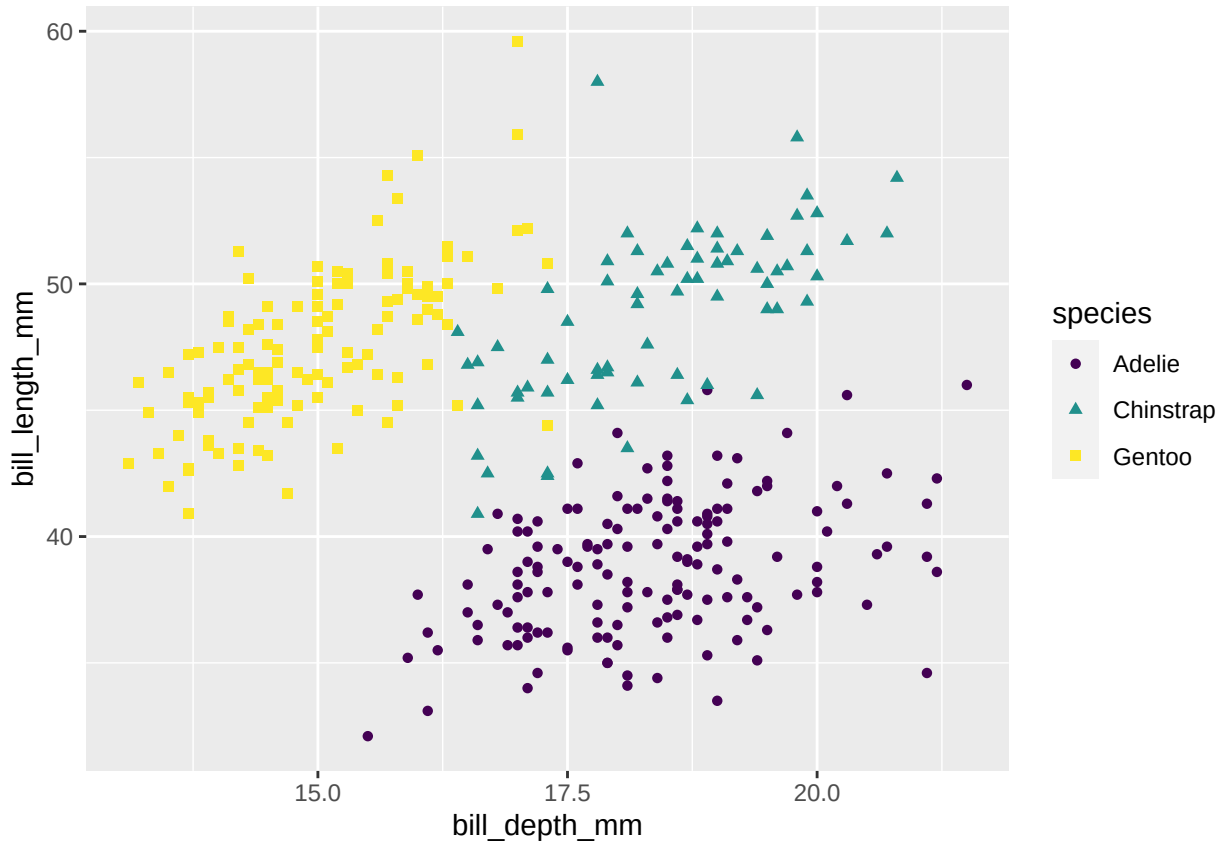

```
# Mapping shape to a specific variable
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species,
           shape = island) +
  geom_point() + scale_colour_viridis_d()
```

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



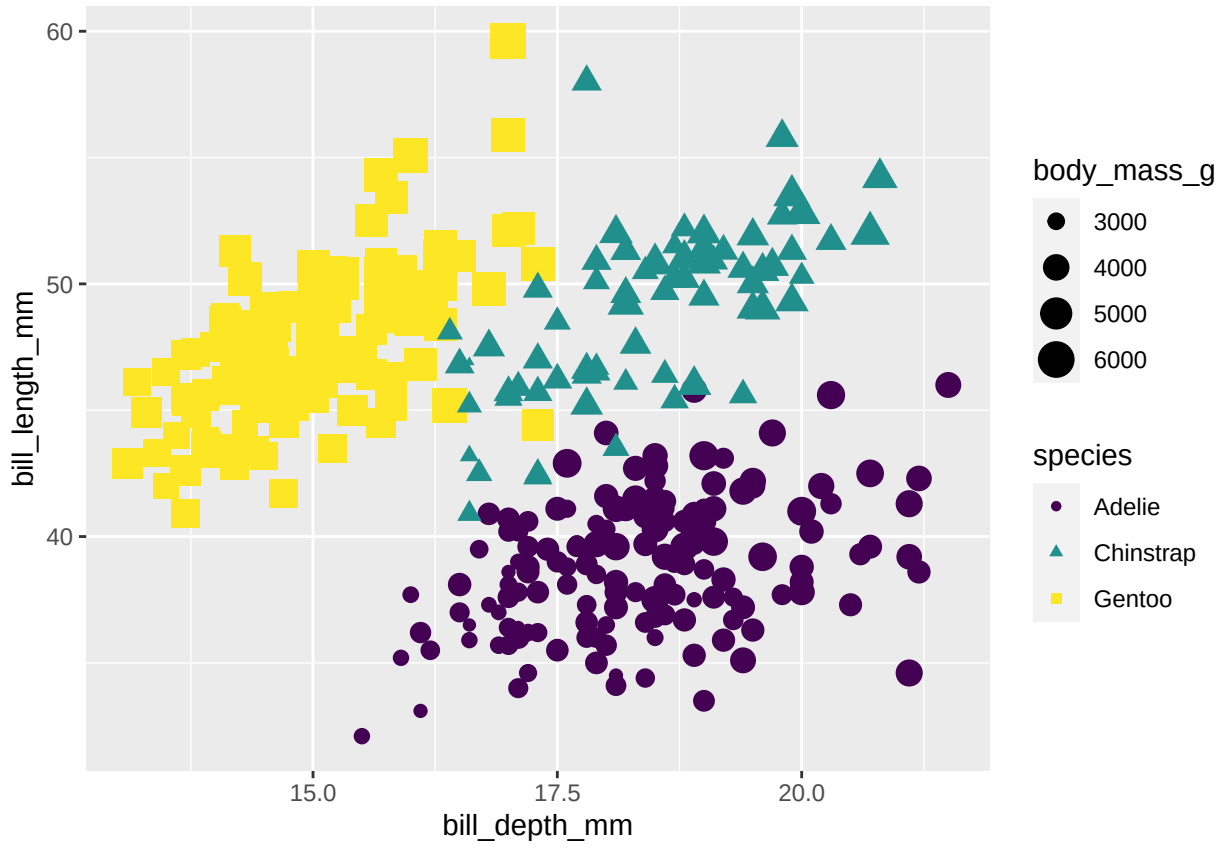

```
# Mapping shape to a specific variable
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species,
           shape = species) +
  geom_point() + scale_colour_viridis_d()
```

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



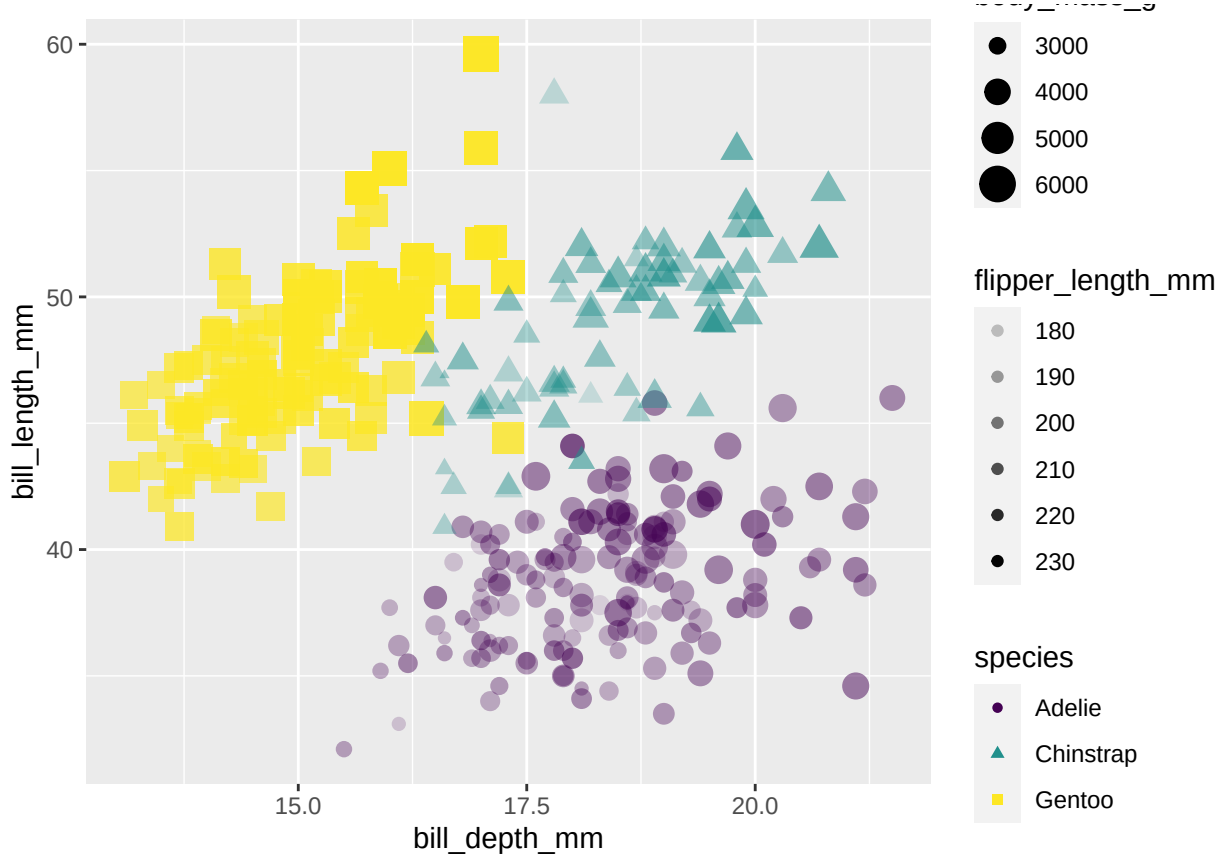

```
# Mapping size to a specific variable
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species,
           shape = species,
           size = body_mass_g) +
  geom_point() + scale_colour_viridis_d()
```

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



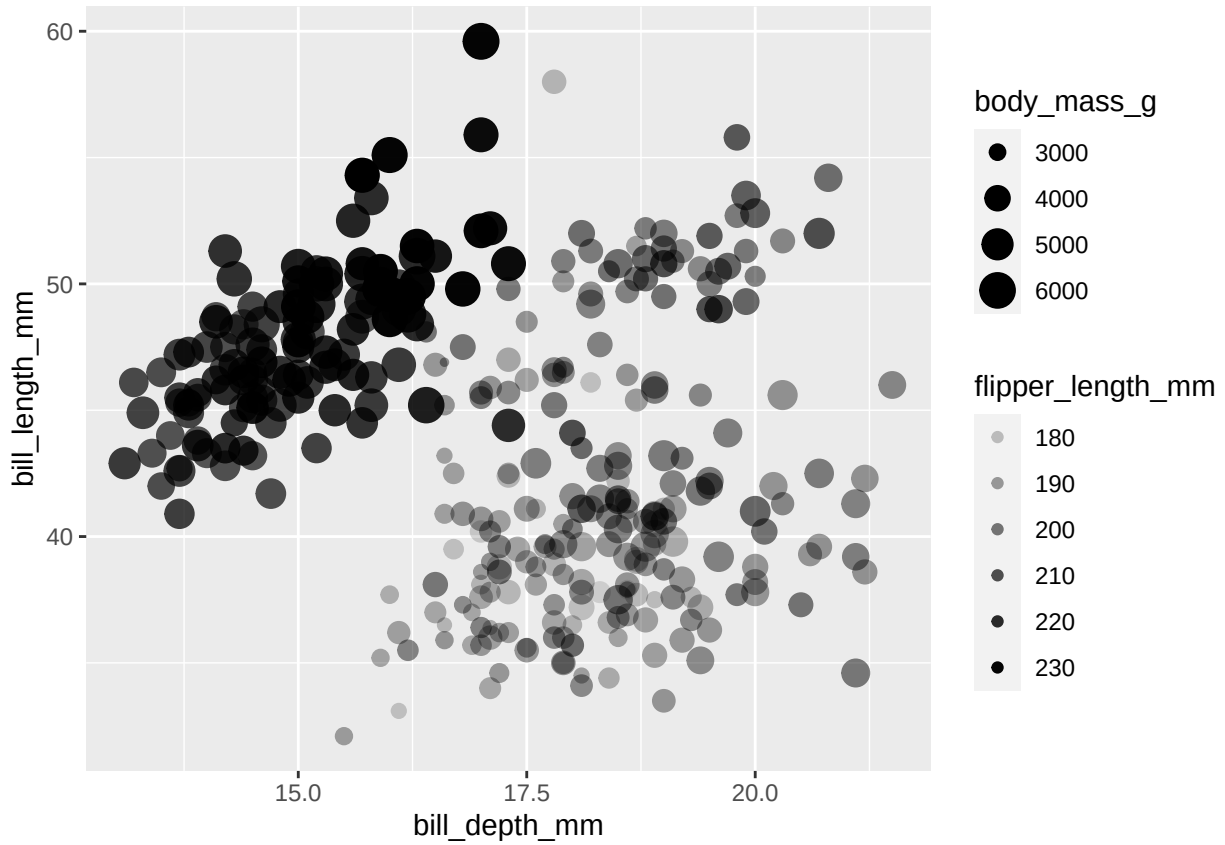

```
# Mapping alpha to a specific variable
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species,
           shape = species,
           size = body_mass_g,
           alpha = flipper_length_mm) +
  geom_point() + scale_colour_viridis_d()
```

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



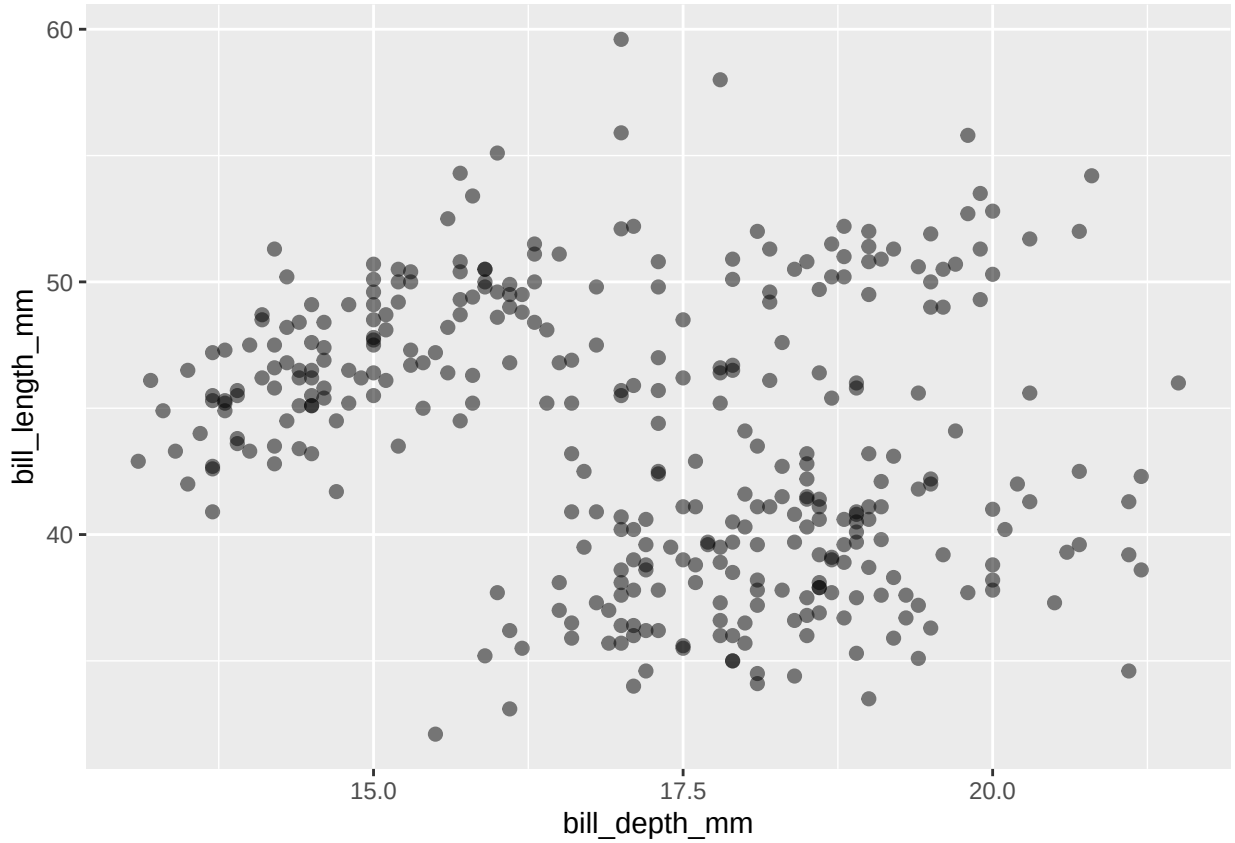

```
# Mapping VS Setting
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           size = body_mass_g,
           alpha = flipper_length_mm) +
  geom_point()
```

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



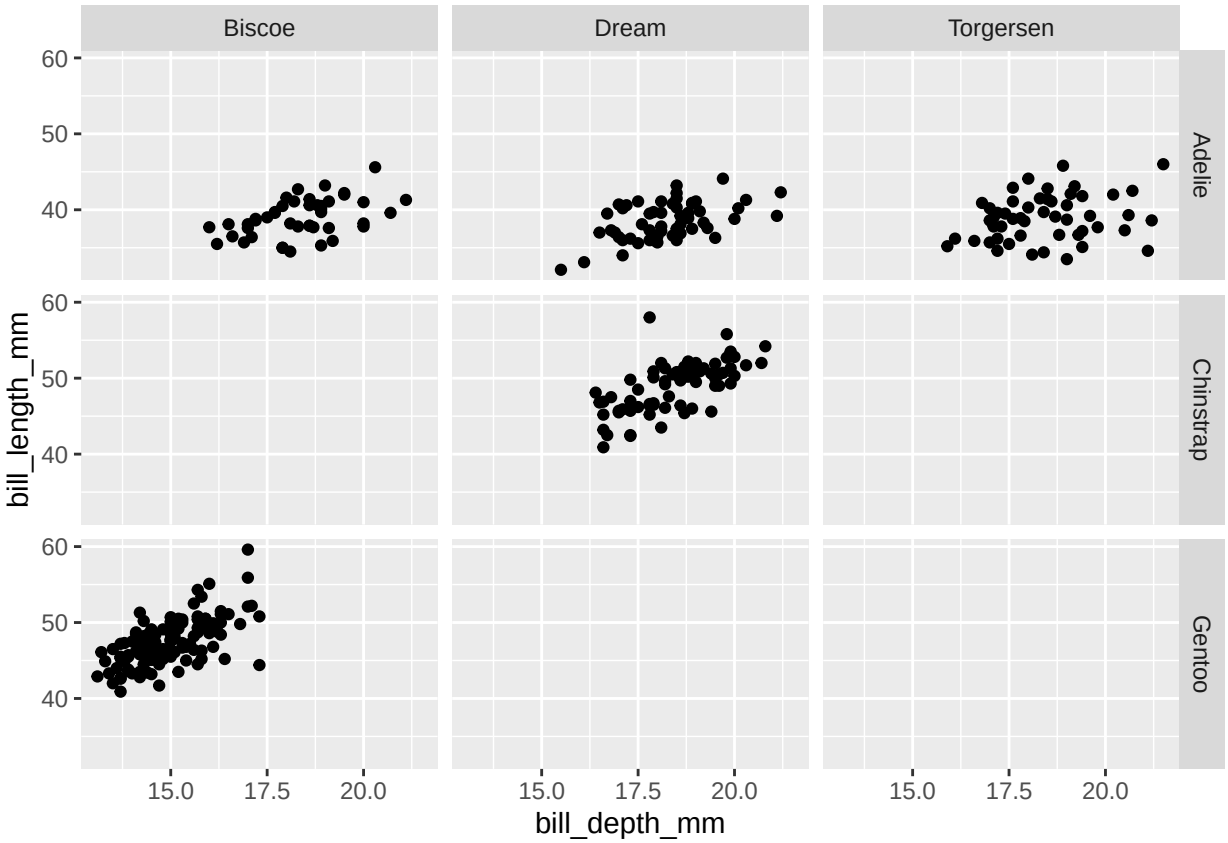

```
# Mapping VS Setting
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point(size = 2, alpha = 0.5)
```

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



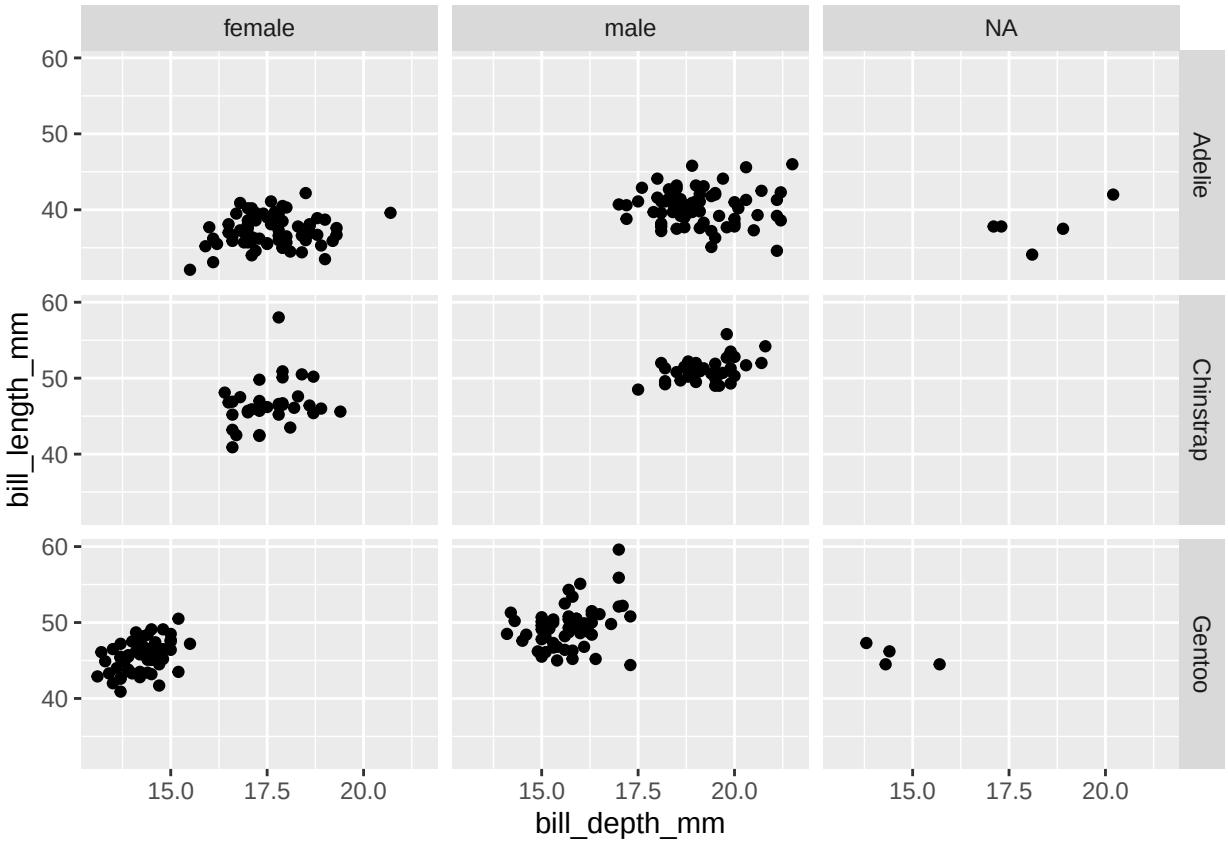

```
# Faceting
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point() +
  facet_grid(species ~ island)
```

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



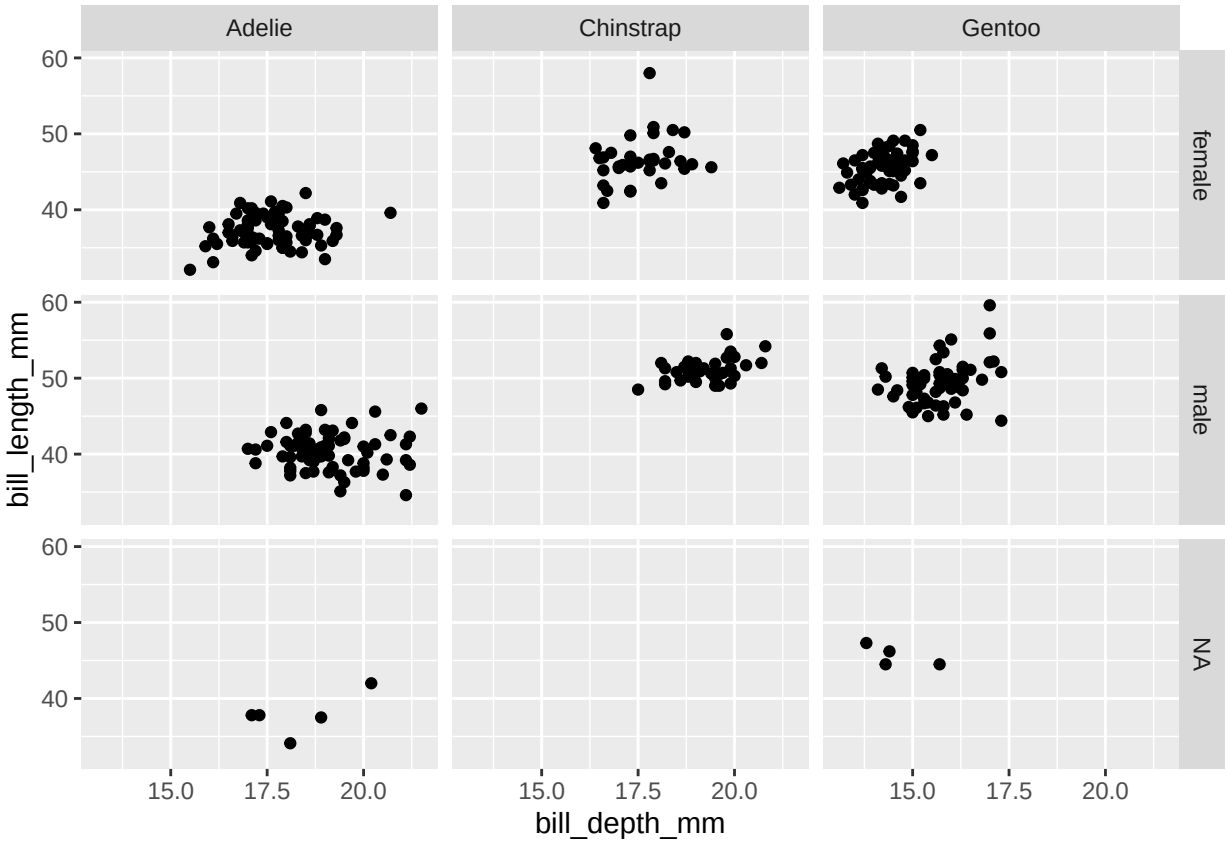

```
# Facet w
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point() +
  facet_grid(species ~ sex)
```

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



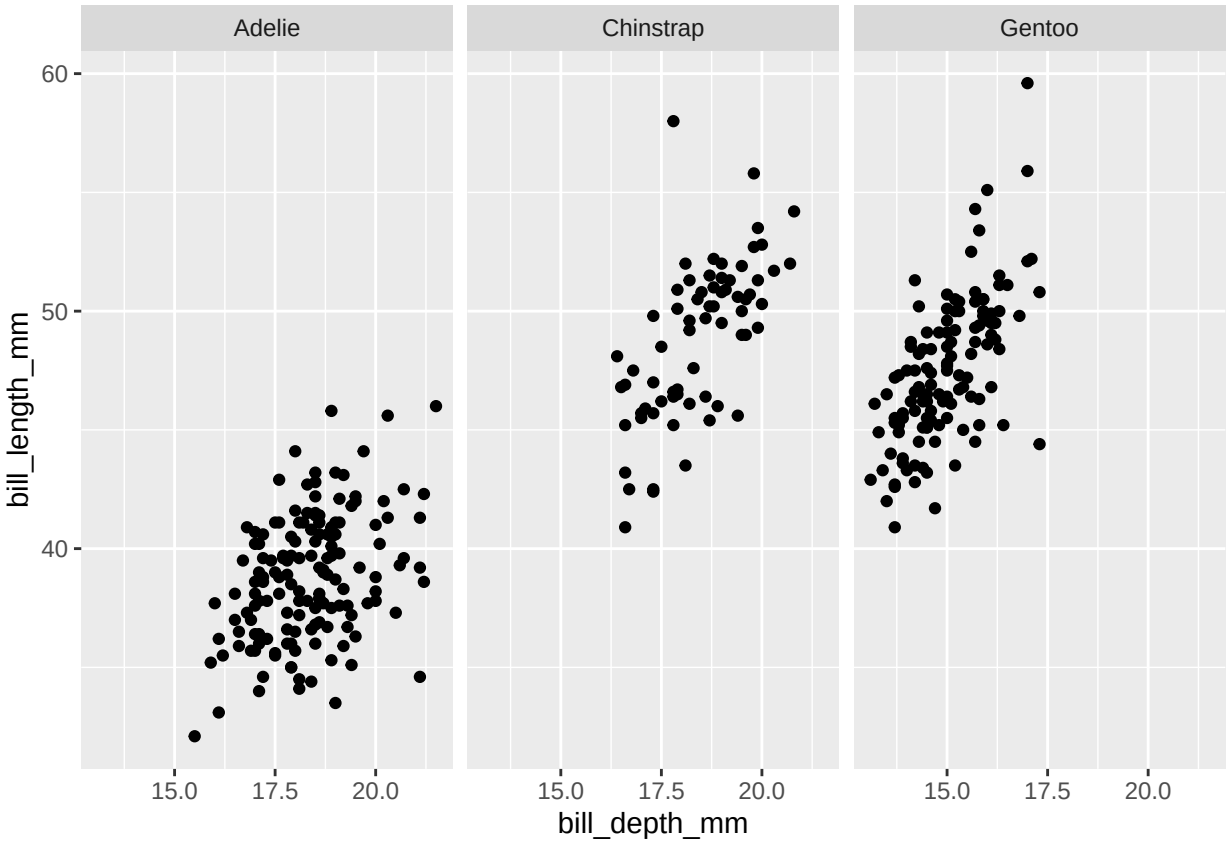

```
# Facet 3
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point() +
  facet_grid(sex ~ species)
```

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



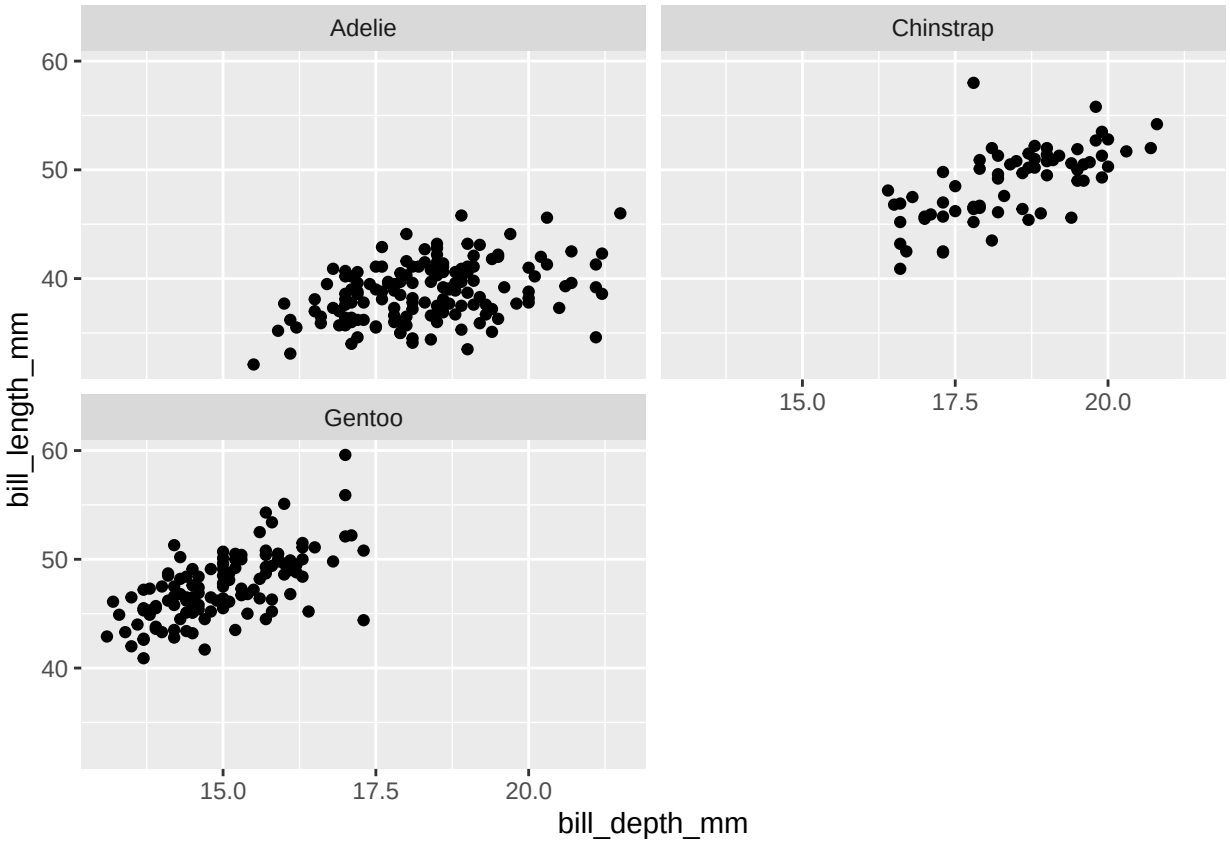

```
# Facet 4
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point() +
  facet_wrap(~ species)
```

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



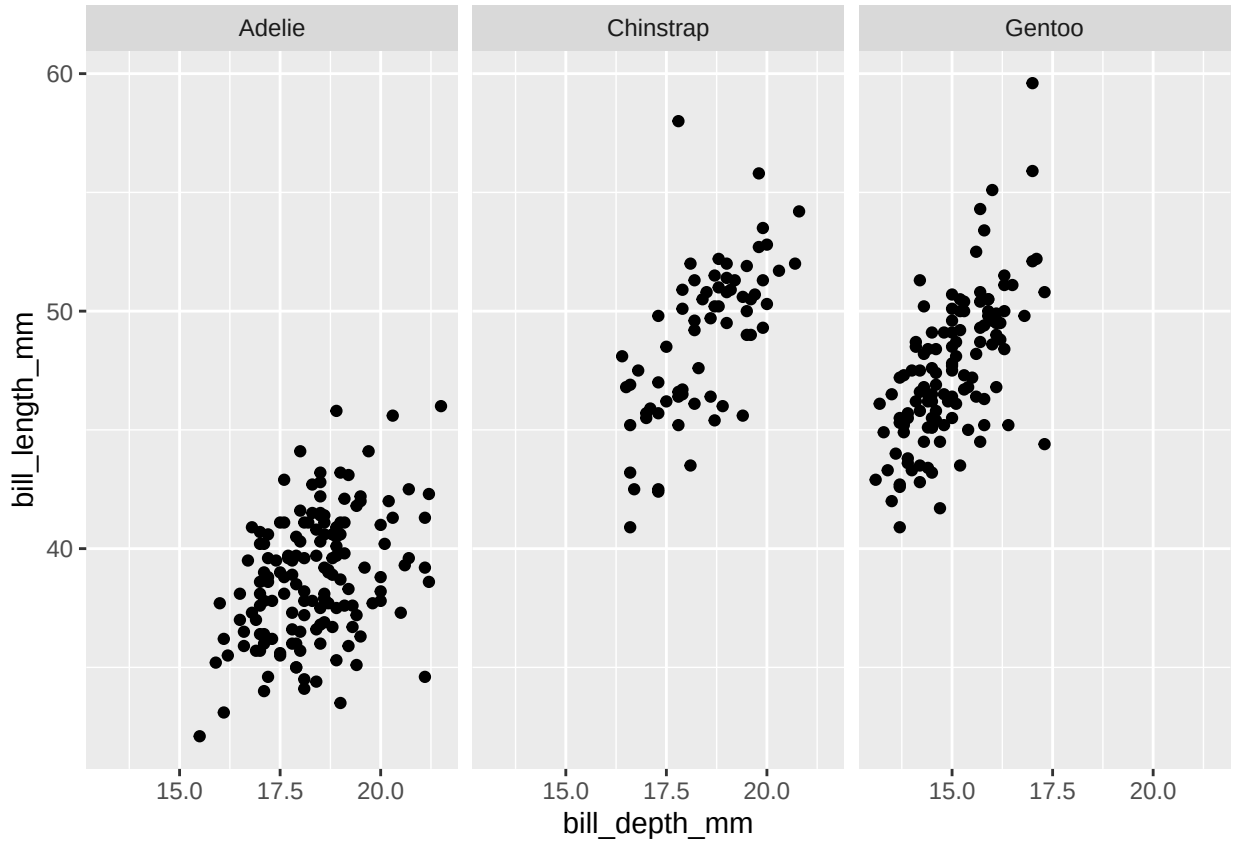

```
# Facet 5
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point() +
  facet_wrap(~ species, ncol = 2)
```

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



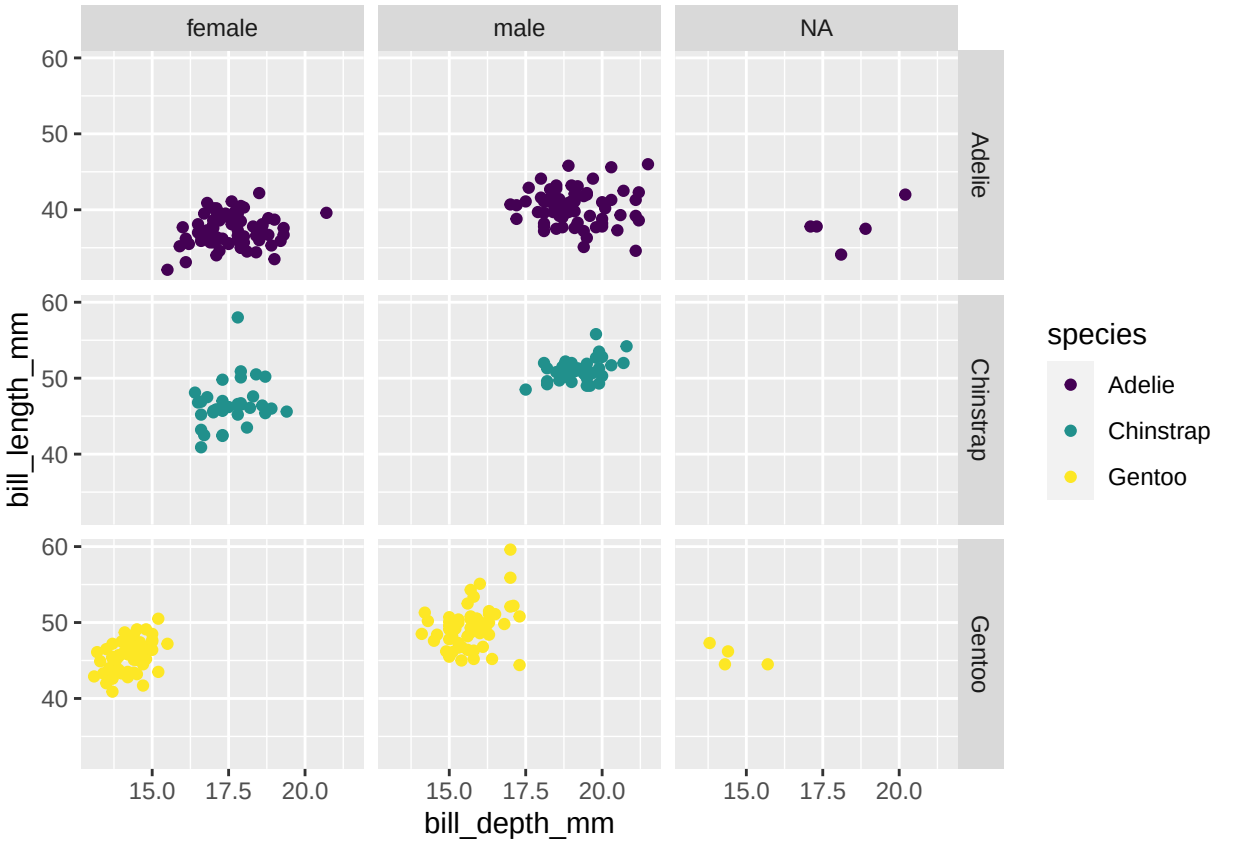

```
# Facet 6
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm) +
  geom_point() +
  facet_grid(. ~ species)
```

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



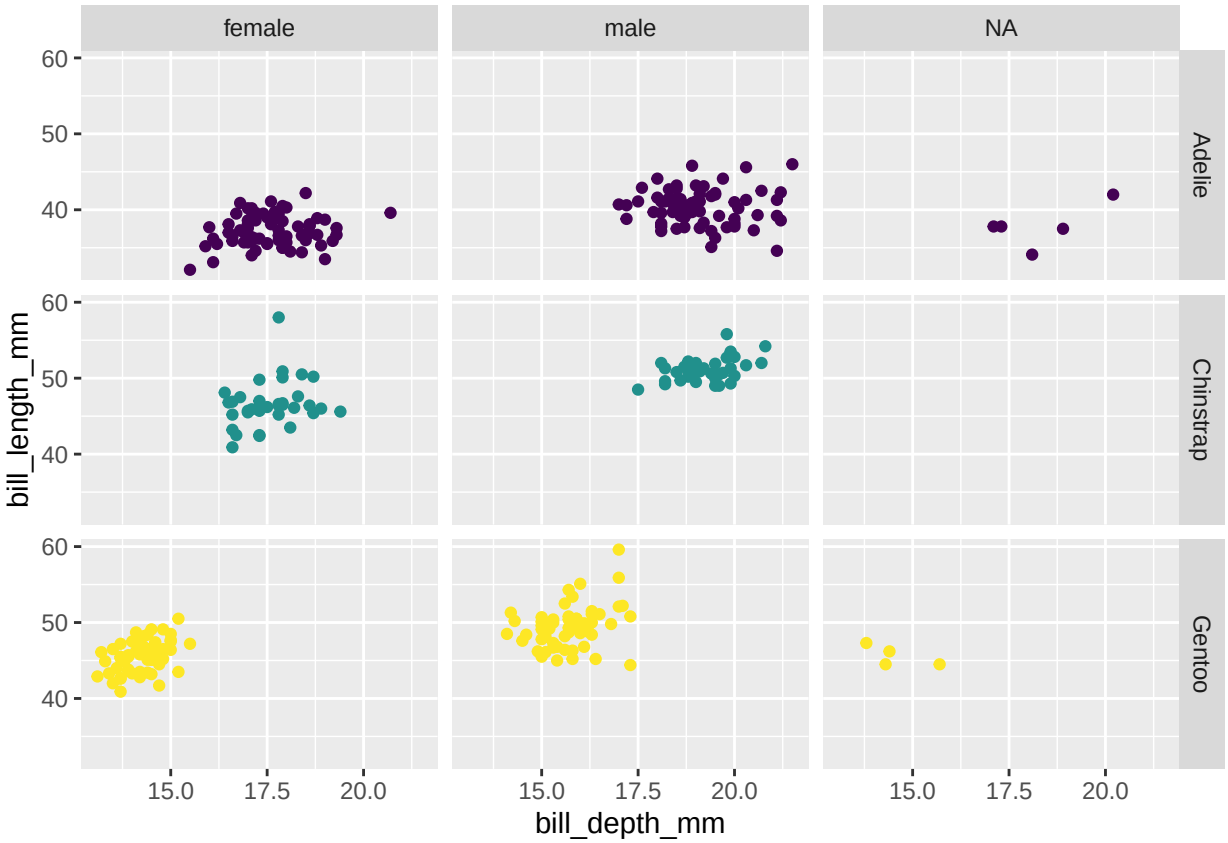

```
# Facet and Colour
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species) +
  geom_point() +
  facet_grid(species ~ sex) + scale_colour_viridis_d()
```

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




```
# Facet and Colour without Legend
# Enter code here
ggplot(penguins) +
       aes(x=bill_depth_mm,
           y=bill_length_mm,
           colour = species) +
  geom_point() +
  facet_grid(species ~ sex) + scale_colour_viridis_d() +
  guides(colour = "none")
```

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




```
# Load package and dataset
# Enter code here
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~
```

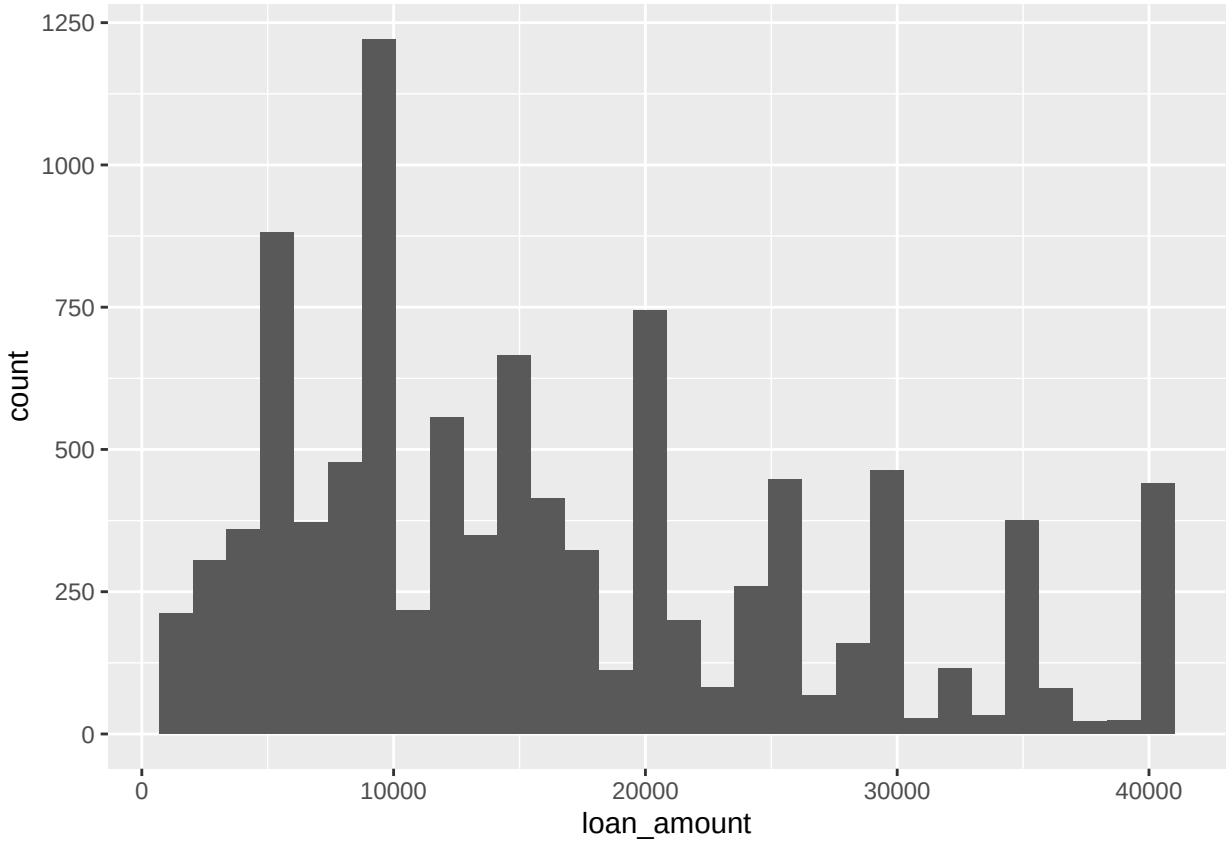
```
# Select variables
# Enter code here
loans <- loans_full_schema %>%
  select(loan_amount, interest_rate, term, grade,
         state, annual_income, homeownership, debt_to_income)
glimpse(loans)
```

```
## Rows: 10,000
```

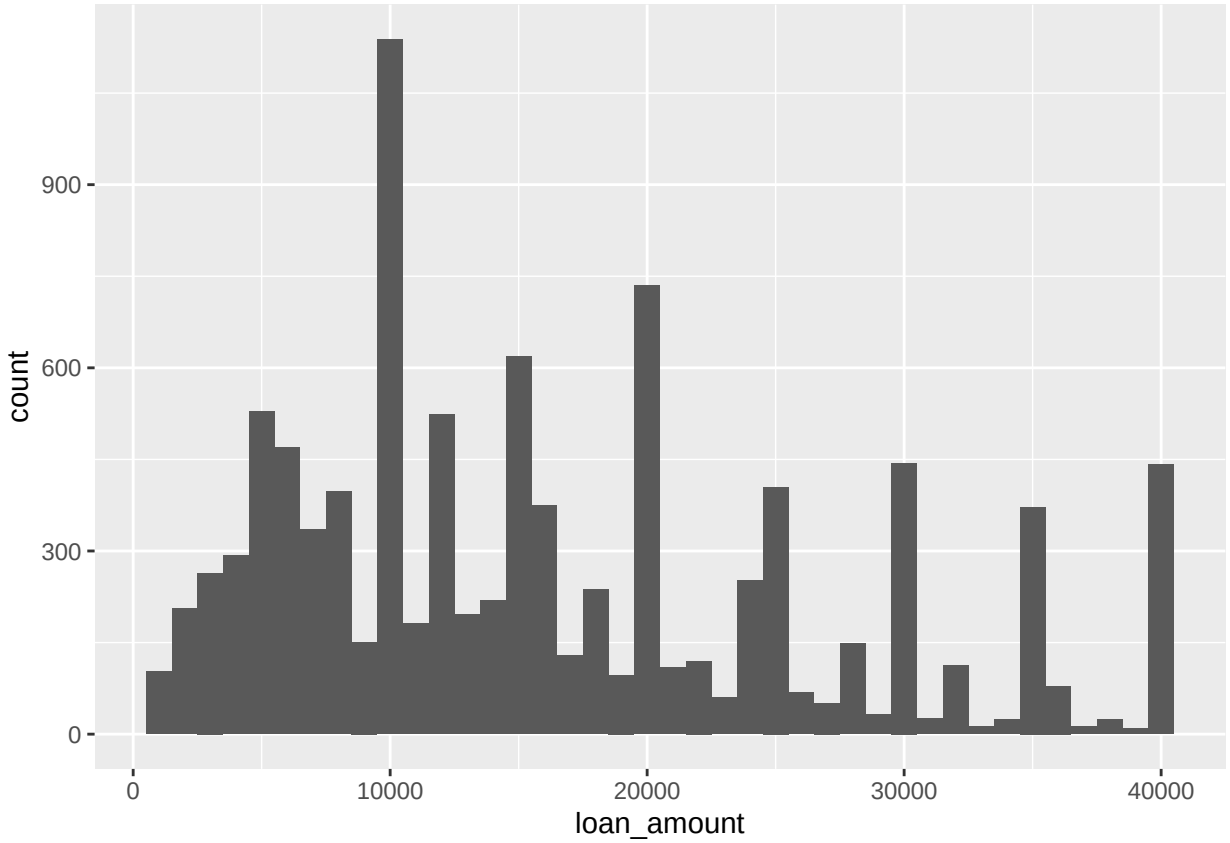
```
## Columns: 8
## $ loan_amount    <int> 28000, 5000, 2000, 21600, 23000, 5000, 24000, 20000, 20~
## $ interest_rate  <dbl> 14.07, 12.61, 17.09, 6.72, 14.07, 6.72, 13.59, 11.99, 1~
## $ term           <dbl> 60, 36, 36, 36, 36, 36, 60, 60, 36, 36, 60, 60, 36, 60,~
## $ grade          <fct> C, C, D, A, C, A, C, B, C, A, C, B, C, B, D, D, D, F, E~
## $ state          <fct> NJ, HI, WI, PA, CA, KY, MI, AZ, NV, IL, IL, FL, SC, CO,~
## $ annual_income  <dbl> 90000, 40000, 40000, 30000, 35000, 34000, 35000, 110000~
## $ homeownership  <fct> MORTGAGE, RENT, RENT, RENT, RENT, OWN, MORTGAGE, MORTGA~
## $ debt_to_income <dbl> 18.01, 5.04, 21.15, 10.16, 57.96, 6.46, 23.66, 16.19, 3~
```

```
# Plotting a histogram
# Enter code here
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram()
```

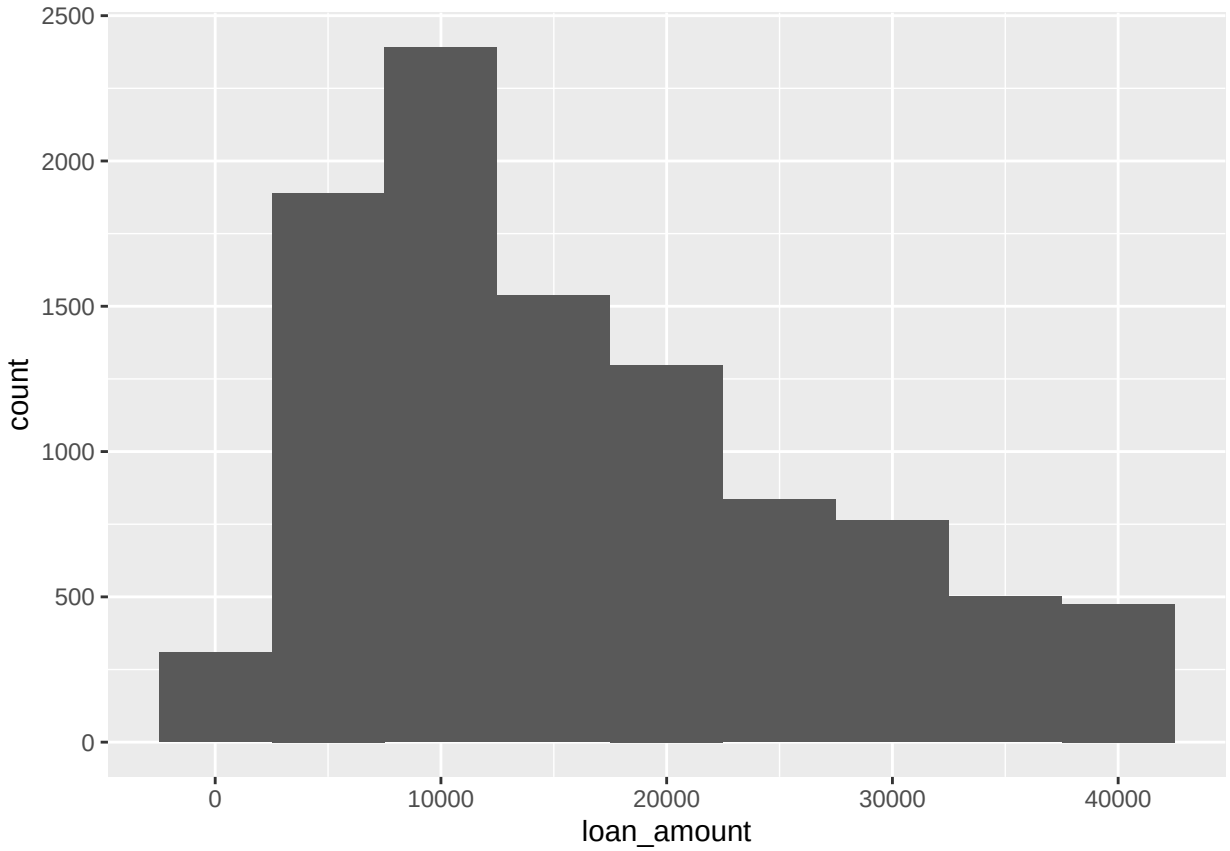
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



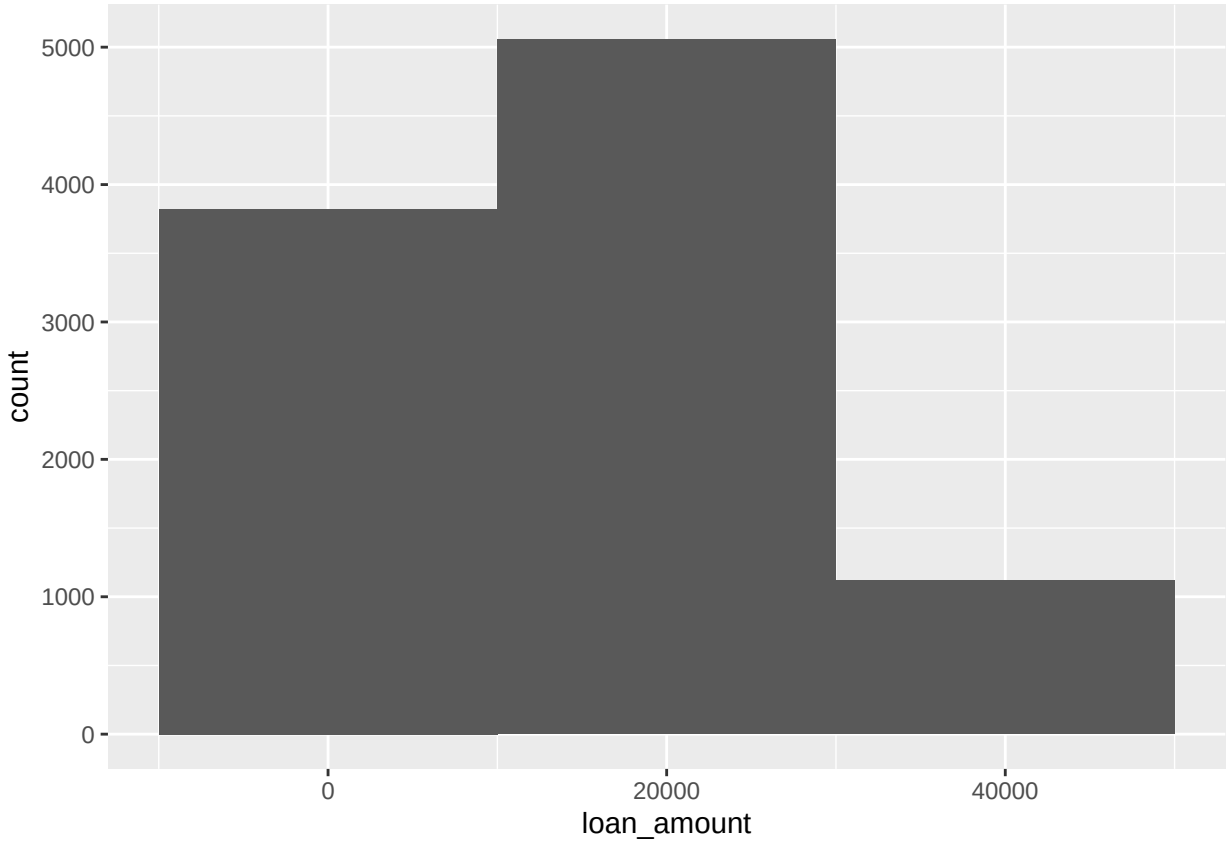

```
# Adjusting binwidth=1000
# Enter code here
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram(binwidth = 1000)
```



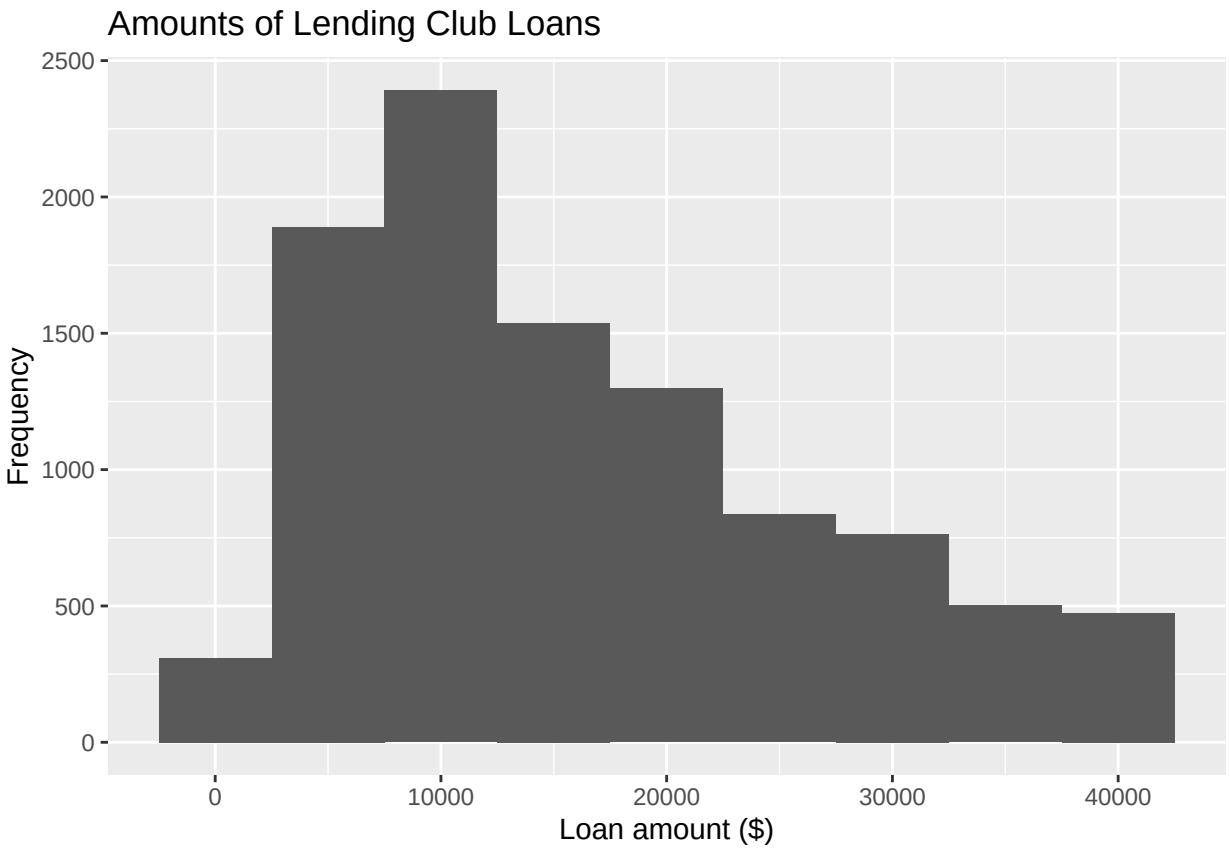

```
# Adjusting binwidth=5000
# Enter code here
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram(binwidth = 5000)
```



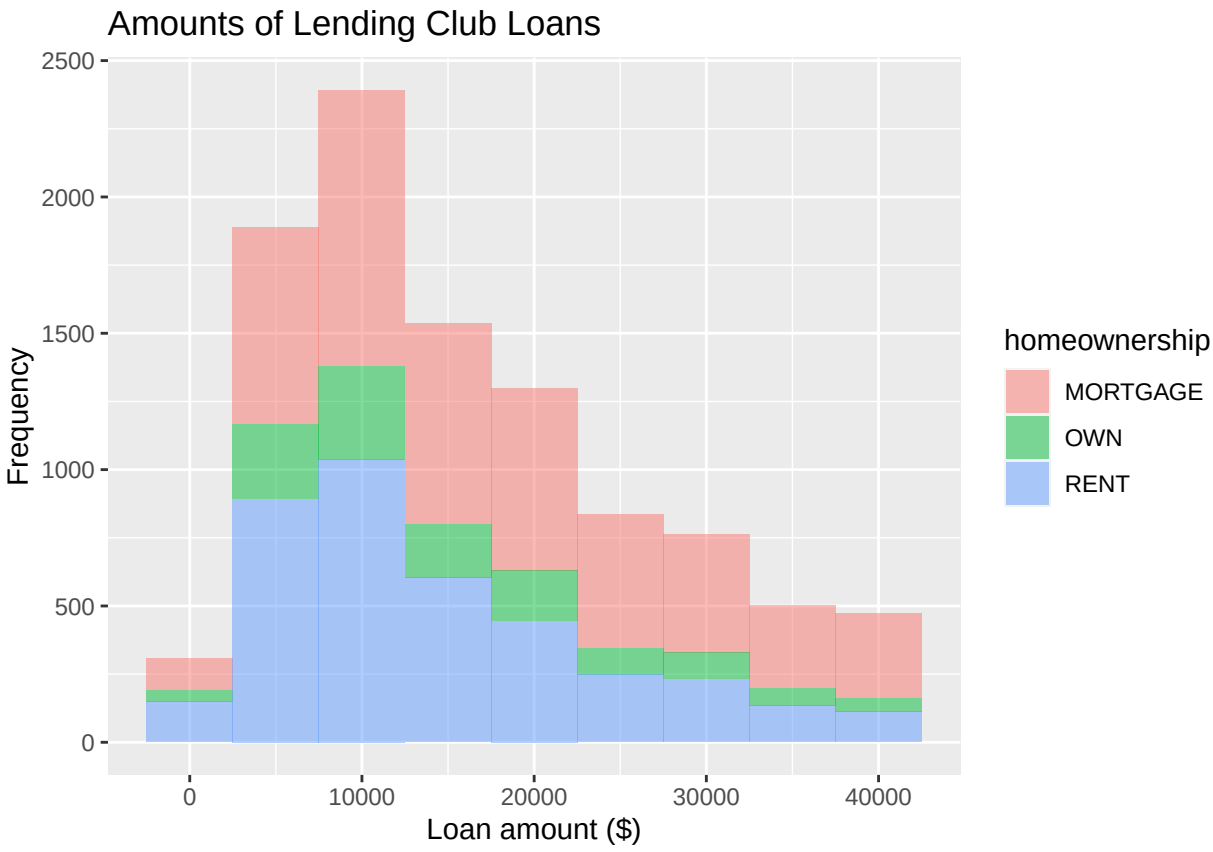

```
# Adjusting binwidth=20000
# Enter code here
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram(binwidth = 20000)
```



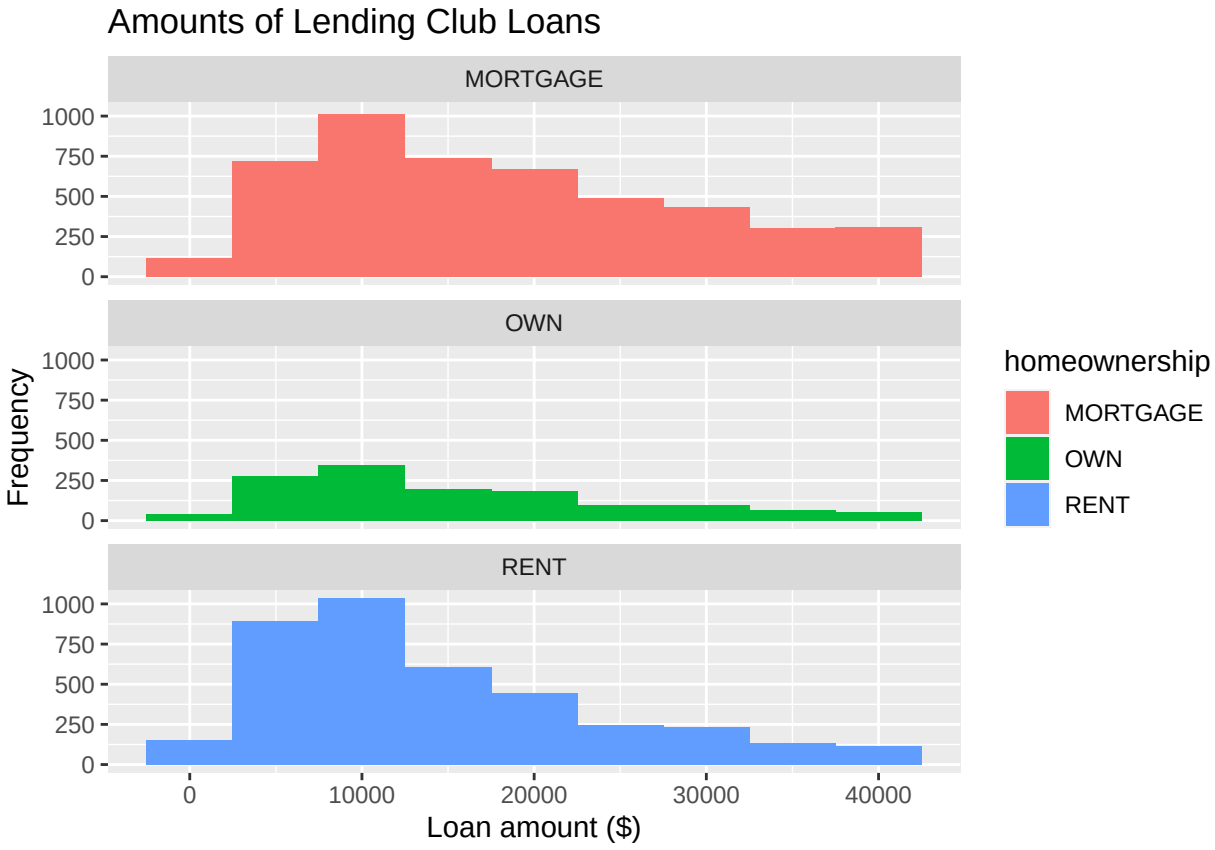

```
# Customising Histograms
# Enter code here
ggplot(loans, aes(x = loan_amount)) + geom_histogram(binwidth = 5000) +
  labs(x = "Loan amount ($)", y = "Frequency", title = "Amounts of Lending Club Loans")
```



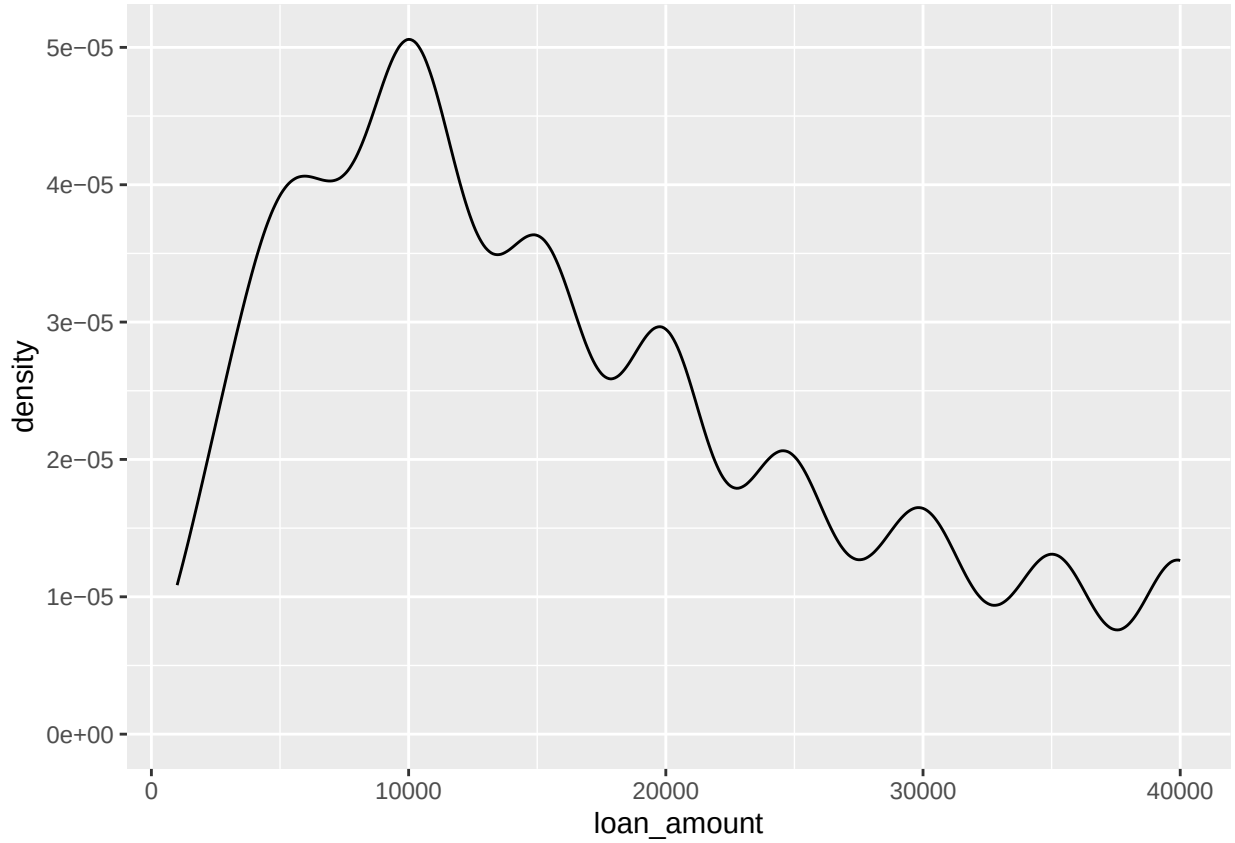

```
# Fill with a categorical variable
# Enter code here
ggplot(loans, aes(x = loan_amount, fill = homeownership)) +
  geom_histogram(binwidth = 5000, alpha = 0.5) +
  labs(x = "Loan amount ($)", y = "Frequency", title = "Amounts of Lending Club Loans")
```



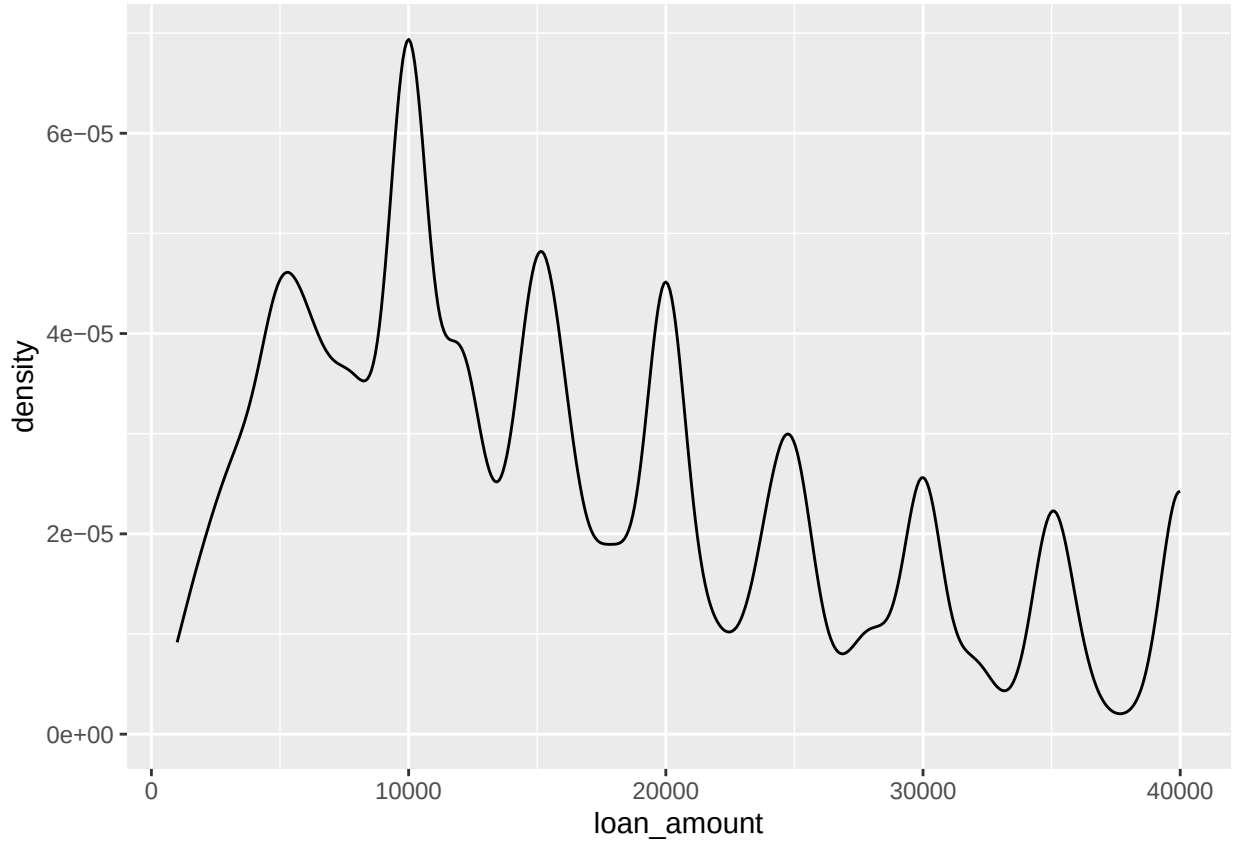

```
# Facet with a categorical variable
# Enter code here
ggplot(loans, aes(x = loan_amount, fill = homeownership)) +
  geom_histogram(binwidth = 5000) +
  labs(x = "Loan amount ($)", y = "Frequency", title = "Amounts of Lending Club Loans") +
  facet_wrap(~ homeownership, nrow = 3)
```



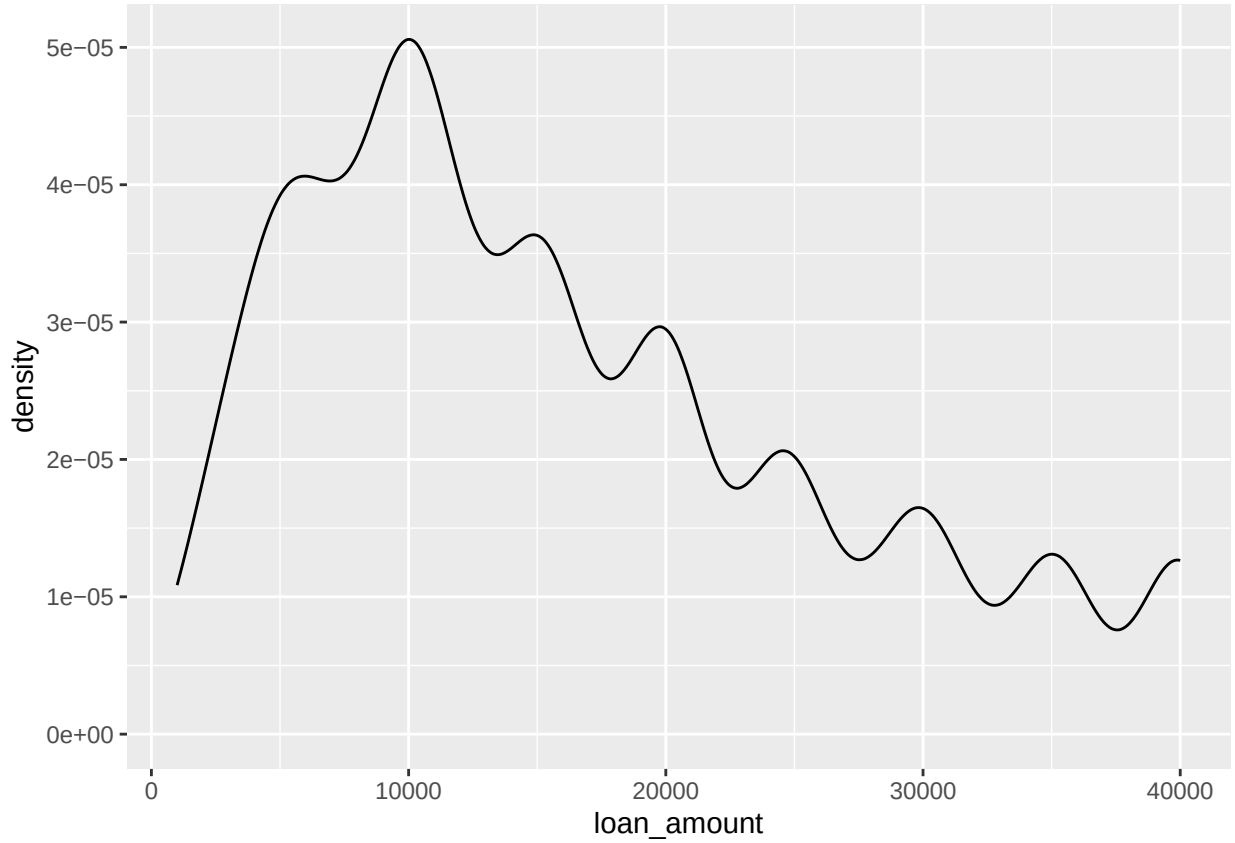

```
# Density Plot
# Enter code here
ggplot(loans, aes(x = loan_amount)) +
  geom_density()
```



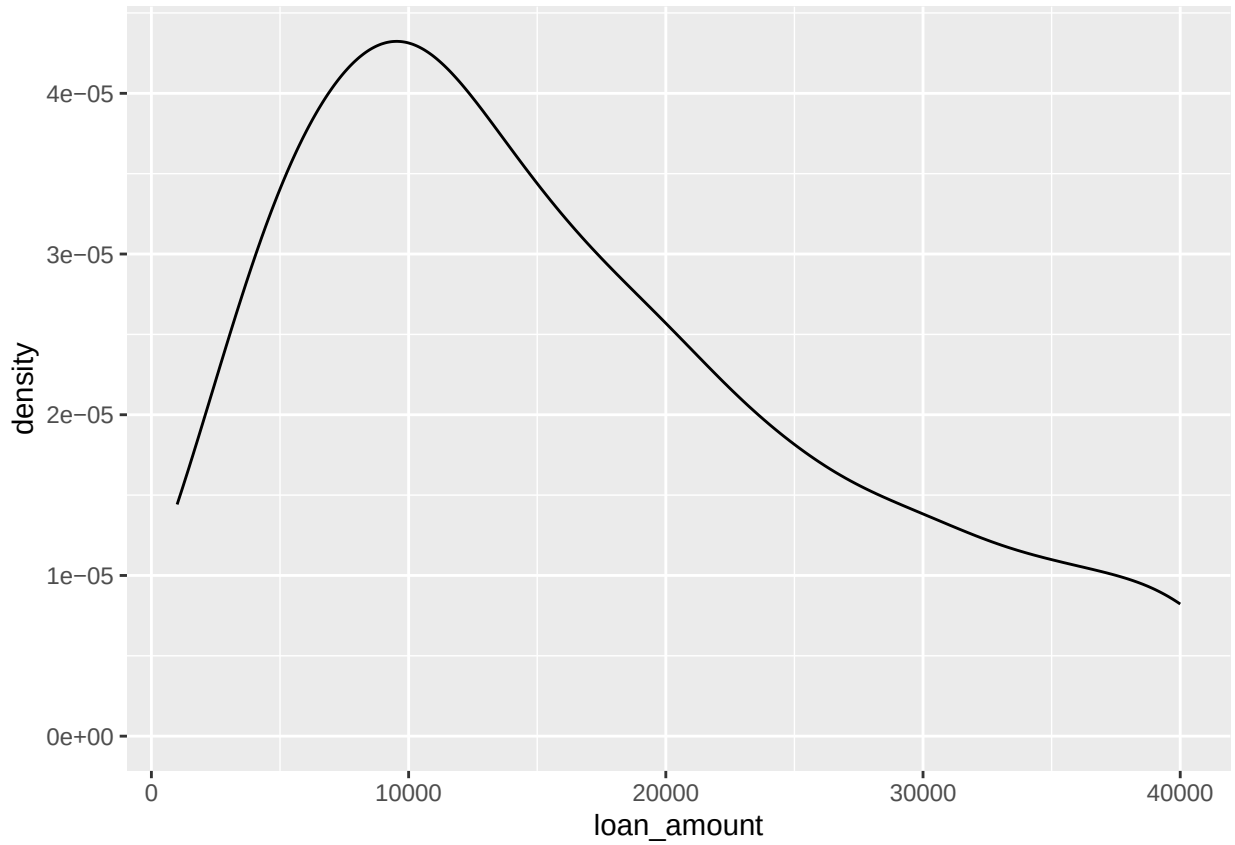

```
# Density Plot and adjusting binwidth
# Enter code here
ggplot(loans, aes(x = loan_amount)) +
  geom_density(adjust = 0.5)
```



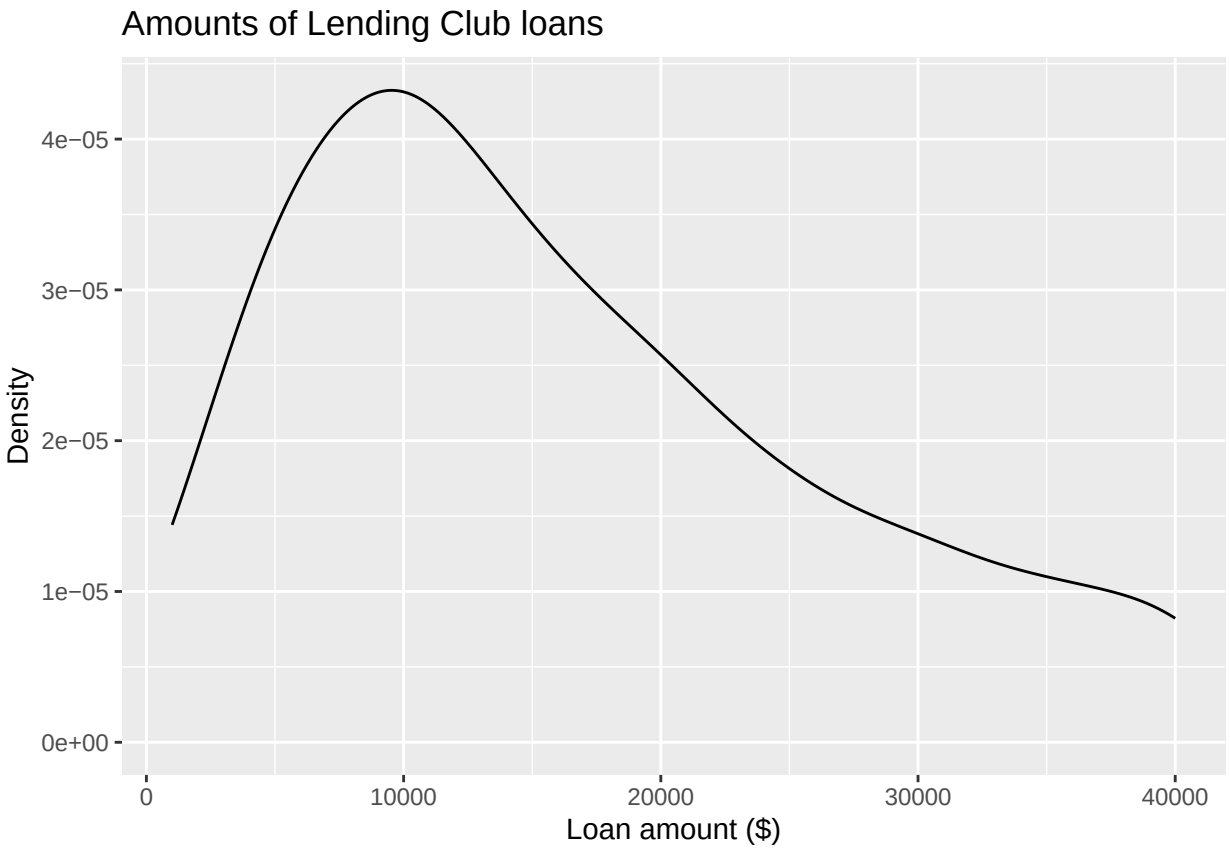

```
# Density Plot and adjusting binwidth
# Enter code here
ggplot(loans, aes(x = loan_amount)) +
  geom_density(adjust = 1)
```



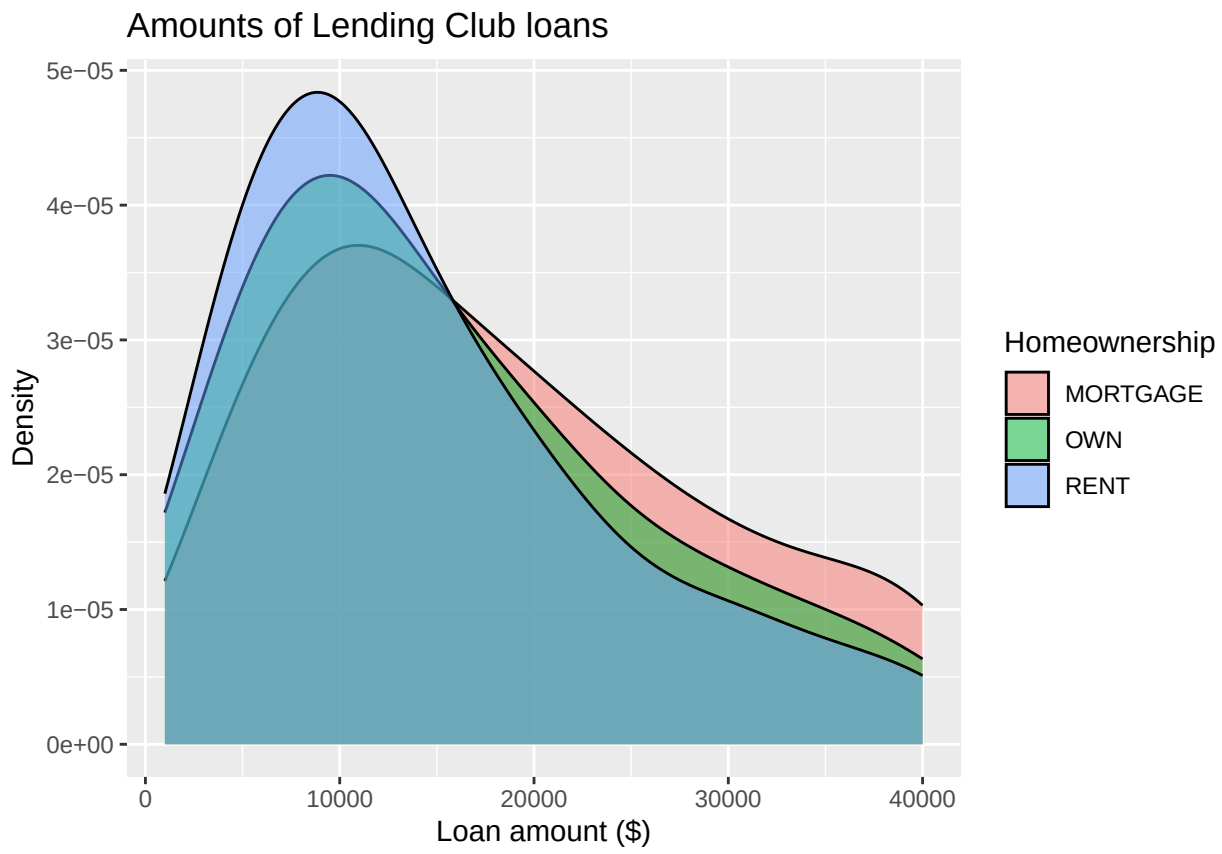

```
# Density Plot and adjusting binwidth
# Enter code here
ggplot(loans, aes(x = loan_amount)) +
  geom_density(adjust = 2)
```



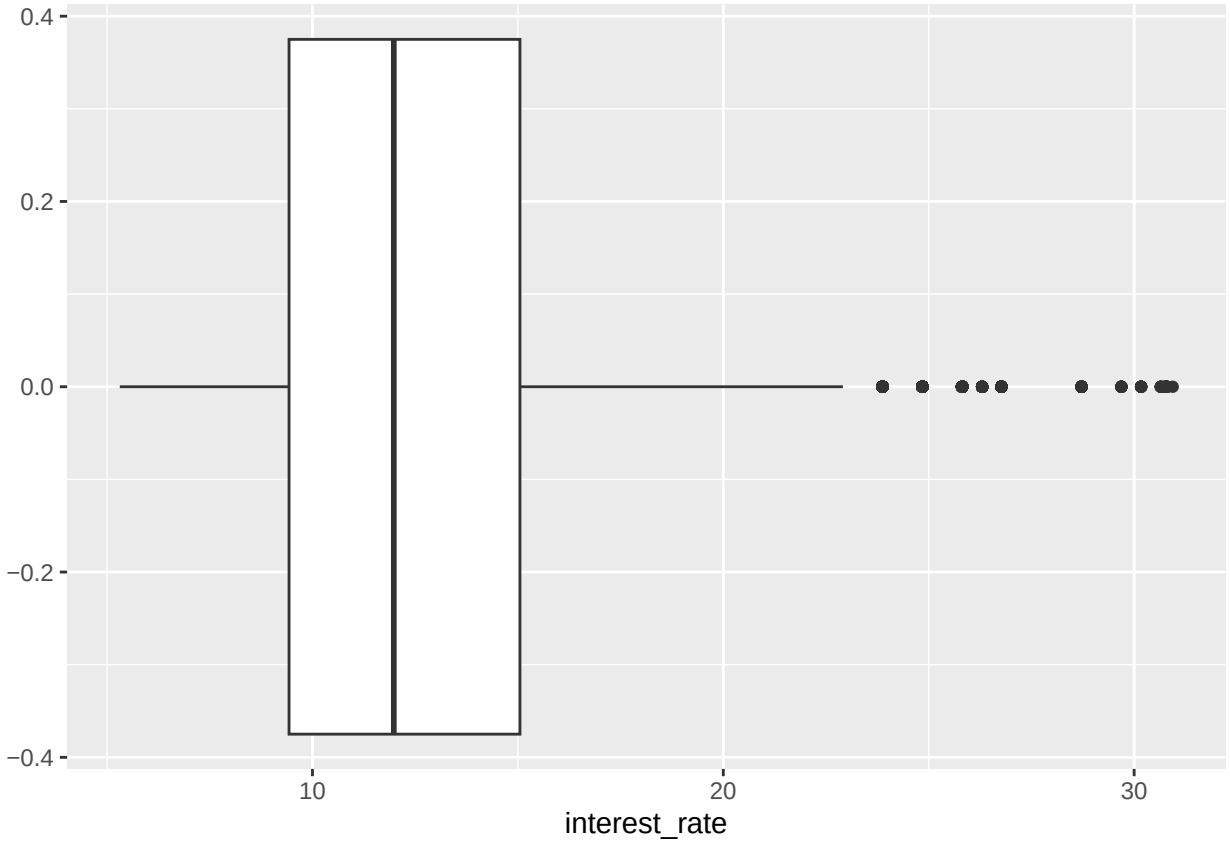

```
# Customising Density Plot
# Enter code here
ggplot(loans, aes(x = loan_amount)) +
  geom_density(adjust = 2) +
  labs(x = "Loan amount ($)", y = "Density", title = "Amounts of Lending Club loans")
```



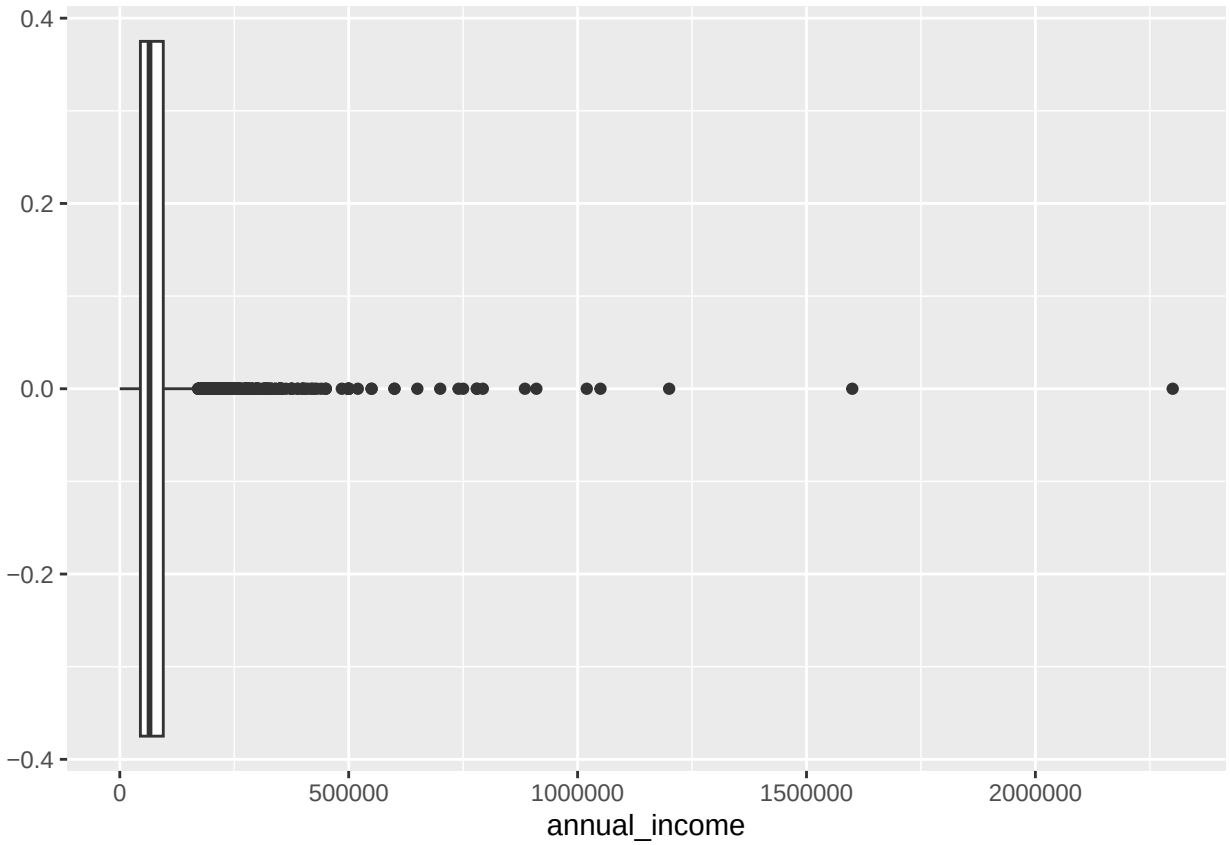

```
# Adding a categorical variable
# Enter code here
ggplot(loans, aes(x = loan_amount, fill = homeownership)) +
  geom_density(adjust = 2, alpha = 0.5) +
  labs(x = "Loan amount ($)", y = "Density", title = "Amounts of Lending Club loans", fill = "Homeowners
```



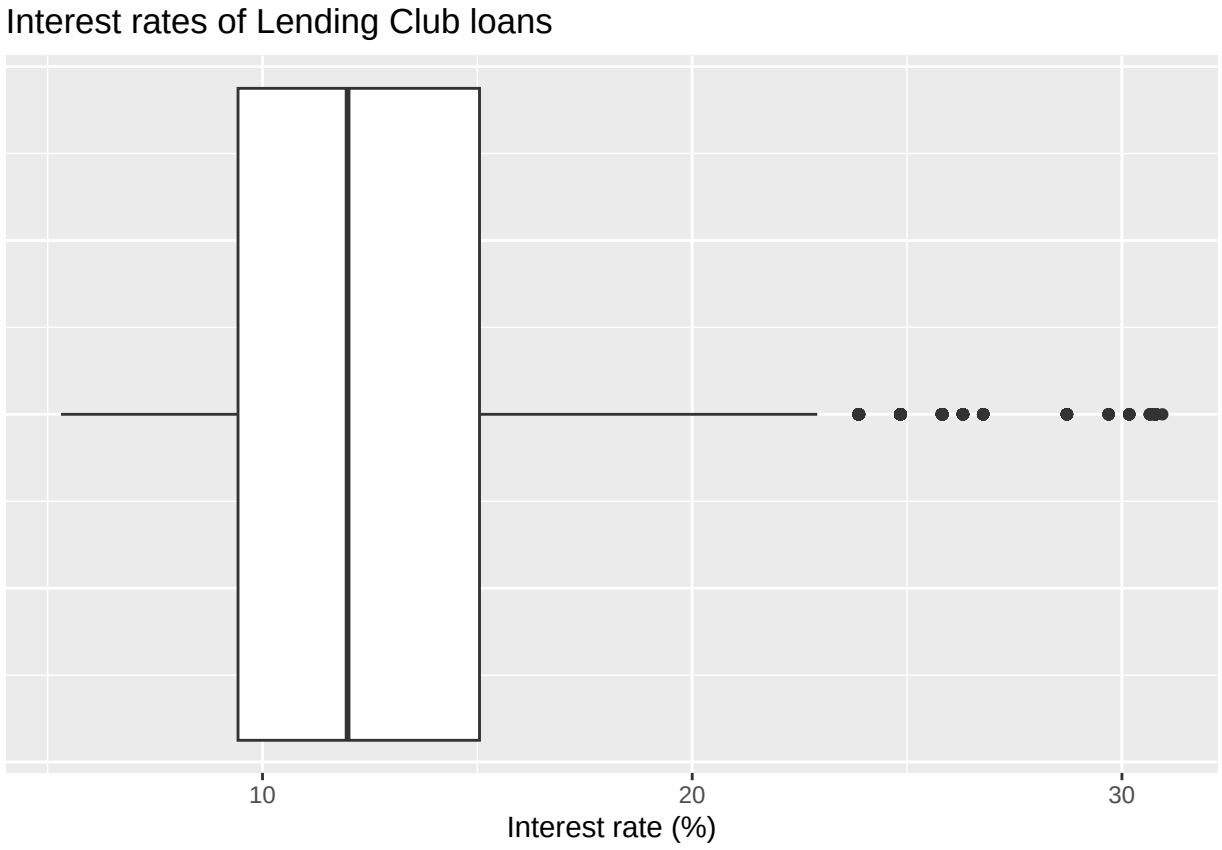

```
# Boxplot
# Enter code here
ggplot(loans, aes(x = interest_rate)) +
  geom_boxplot()
```



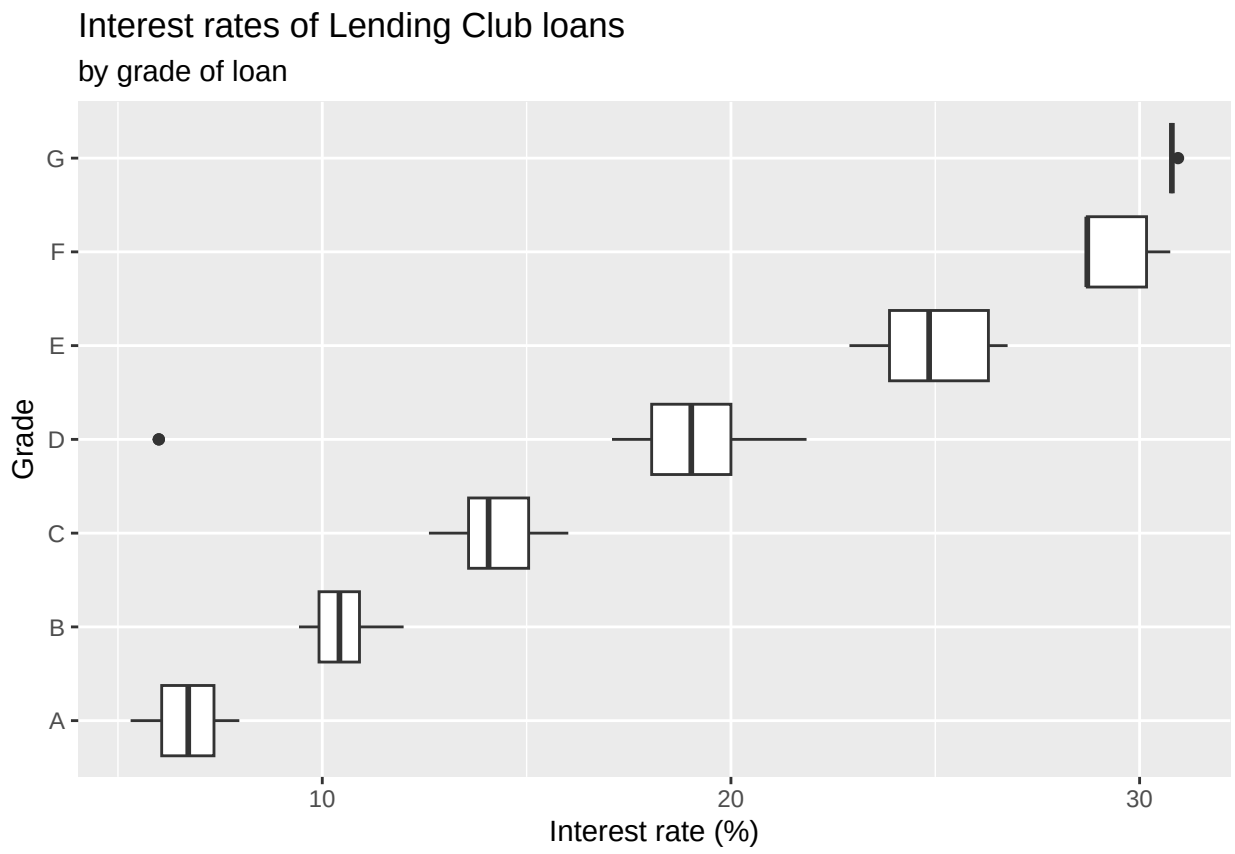

```
# Boxplot and outliers
# Enter code here
ggplot(loans, aes(x = annual_income)) +
  geom_boxplot()
```




```
# Customising boxplots
# Enter code here
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() )
```

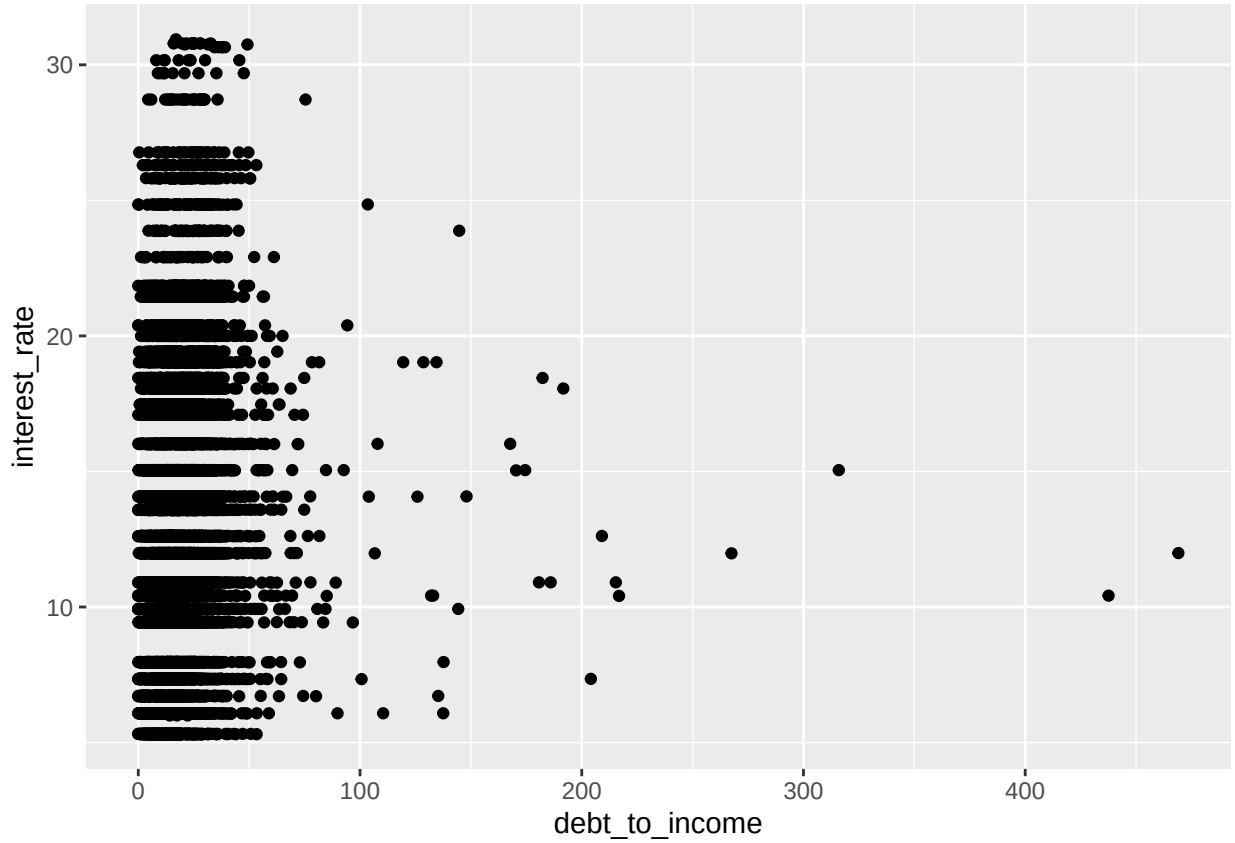



```
# Adding a categoric variable
# Enter code here
ggplot(loans, aes(x = interest_rate,
y = grade)) +
  geom_boxplot() +
  labs(x = "Interest rate (%)",y = "Grade",title = "Interest rates of Lending Club loans",subtitle = "b
```



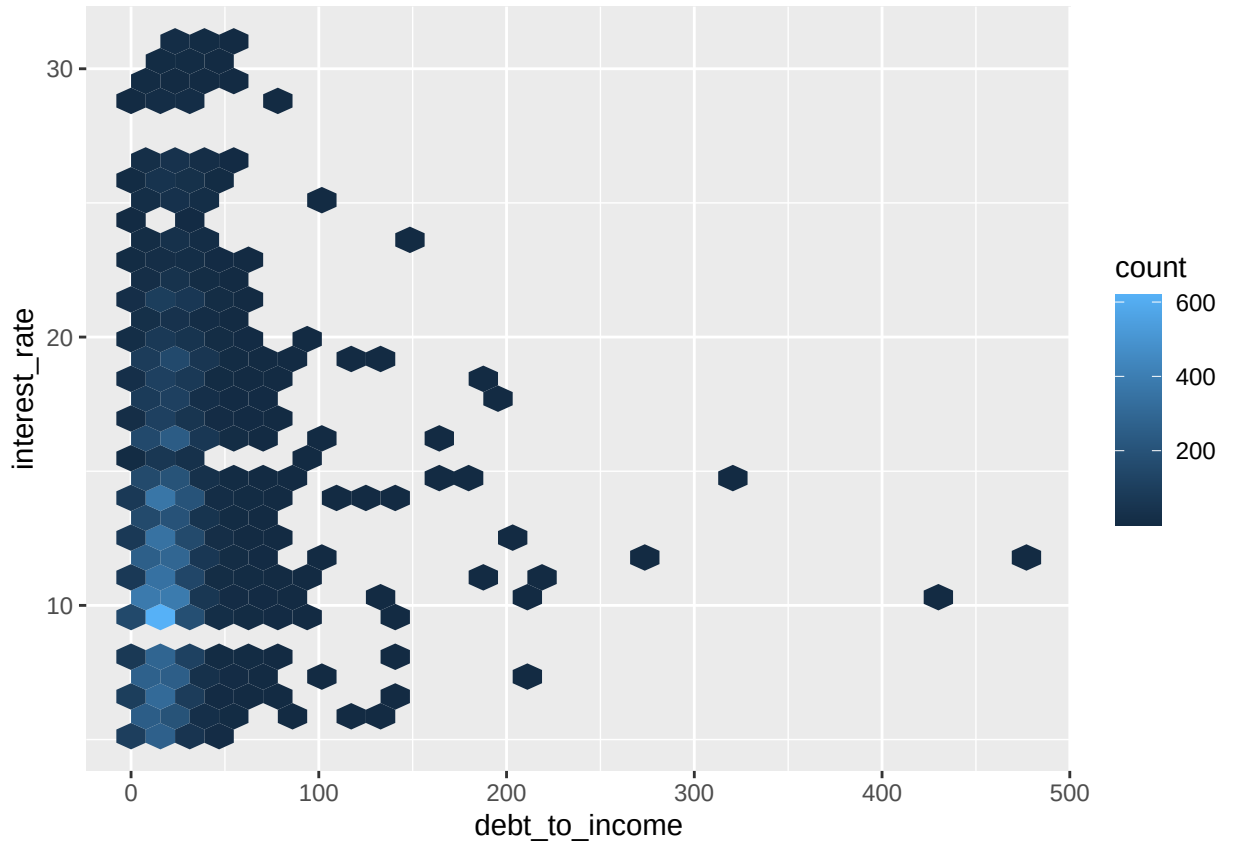

```
# Scatterplot
# Enter code here
ggplot(loans, aes(x = debt_to_income, y = interest_rate)) +
  geom_point()
```

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

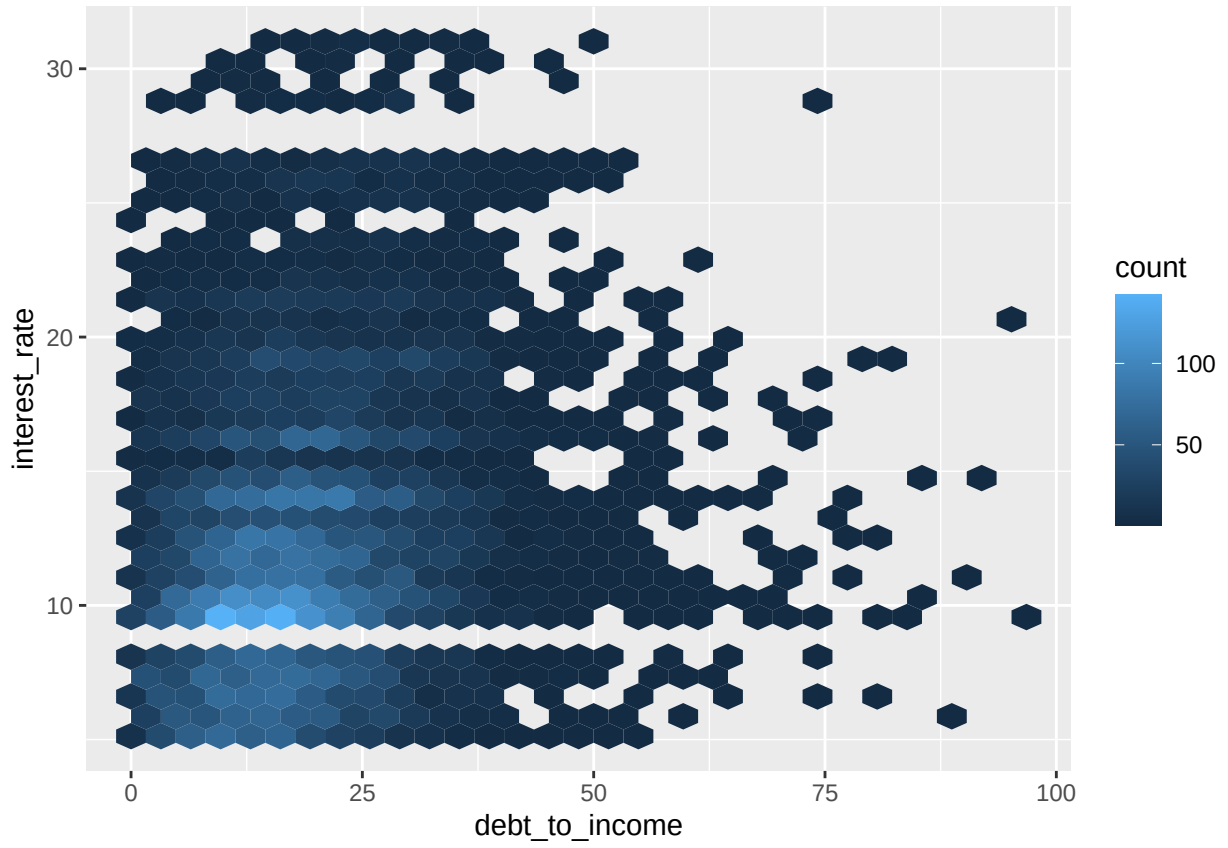



```
# Hex plot
# Enter code here
ggplot(loans, aes(x = debt_to_income, y = interest_rate)) +
  geom_hex()
```

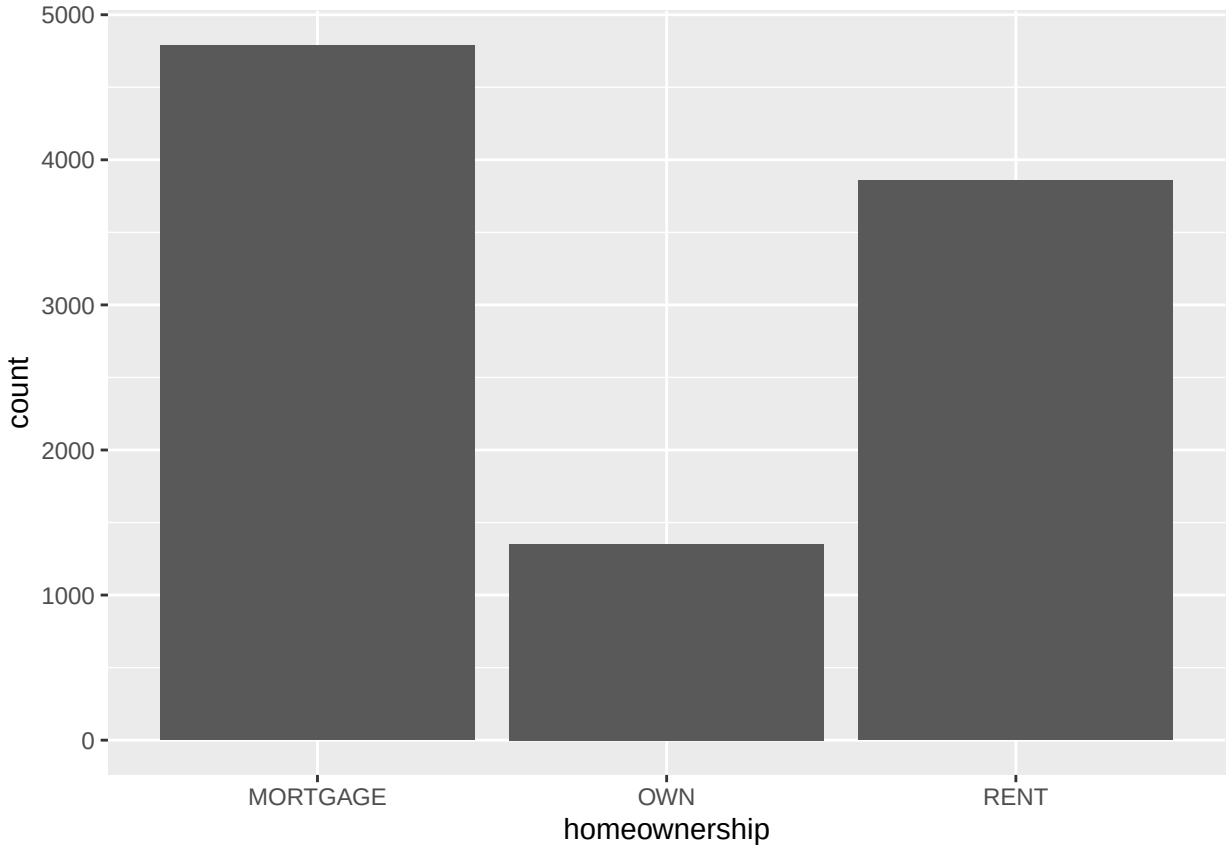
```
## Warning: Removed 24 rows containing non-finite values (‘stat_binhex()‘).
```



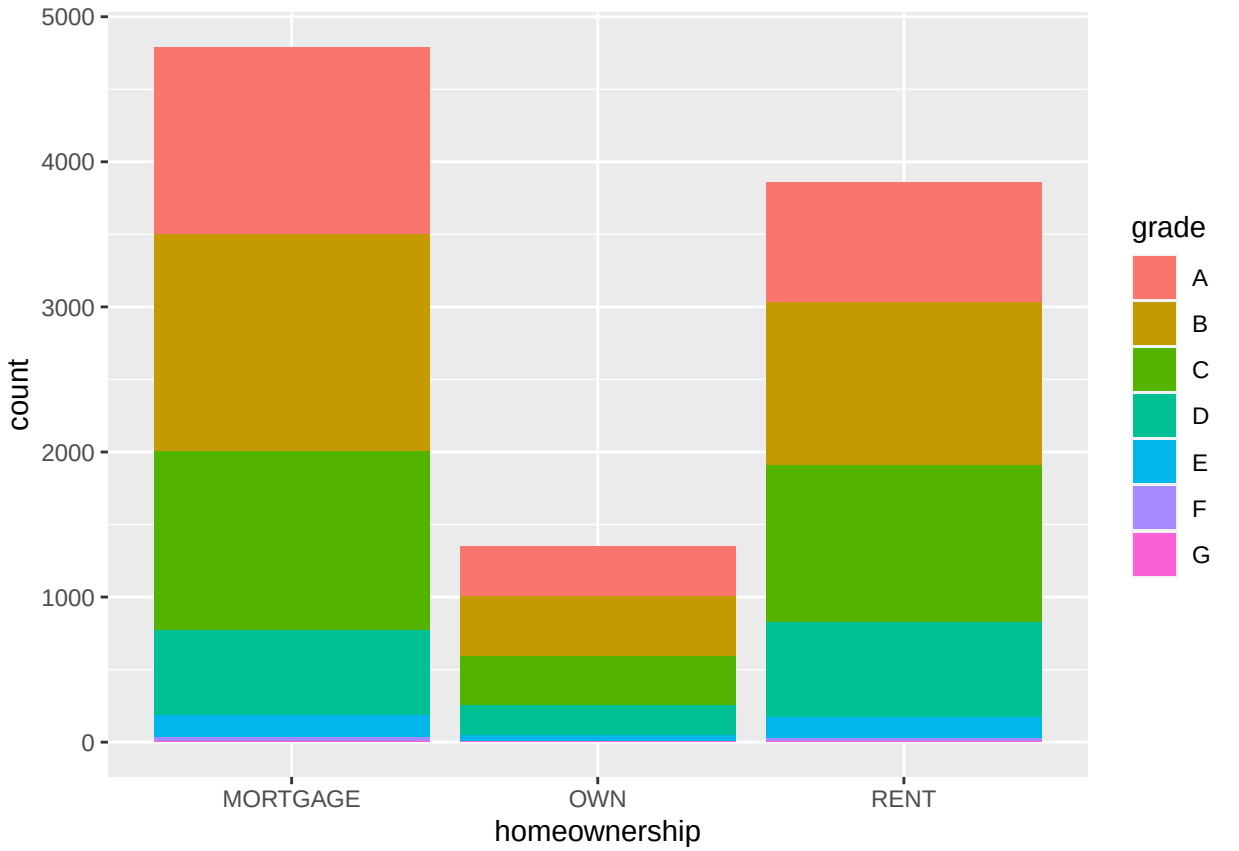

```
# Hex plot
# Enter code here
ggplot(loans %>% filter(debt_to_income < 100),
       aes(x = debt_to_income, y = interest_rate)) +
geom_hex()
```



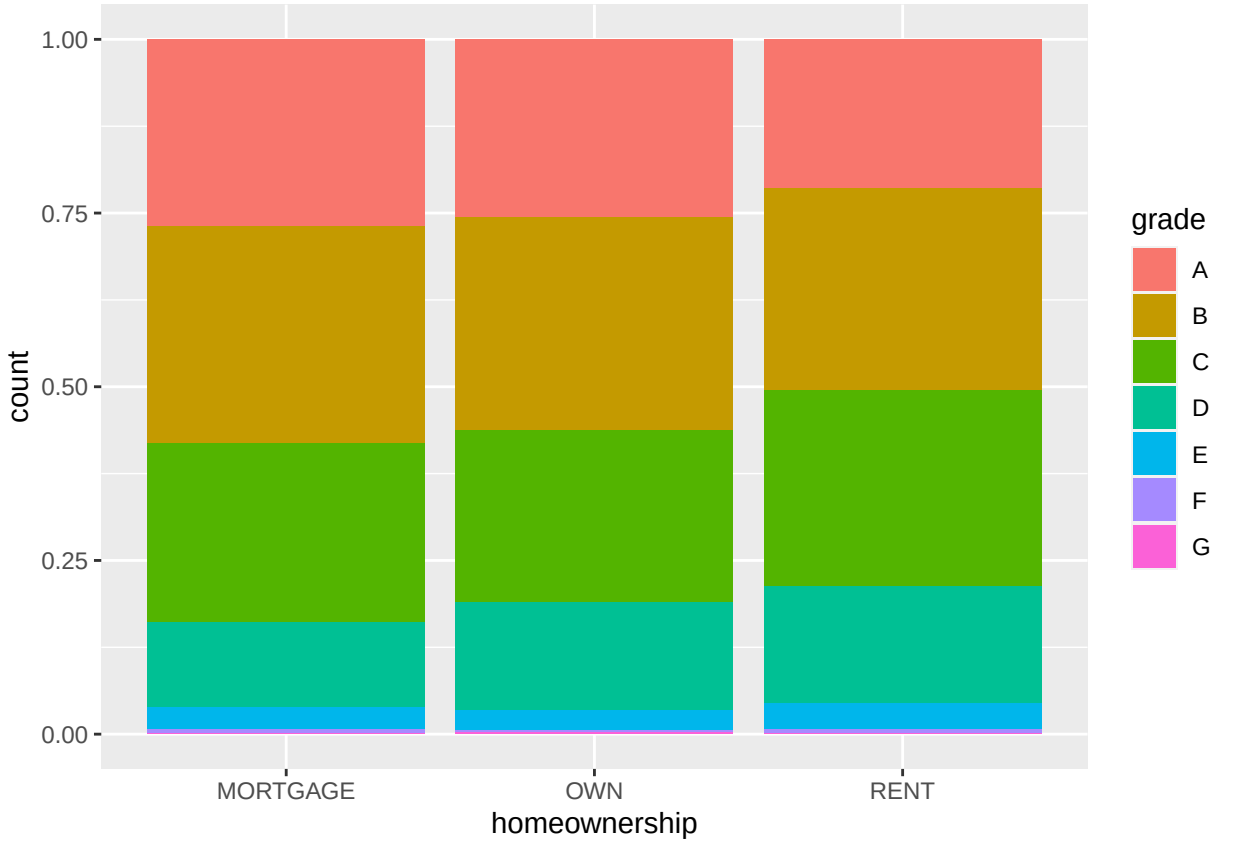

```
# Bar plot
# Enter code here
ggplot(loans, aes(x = homeownership)) +
  geom_bar()
```



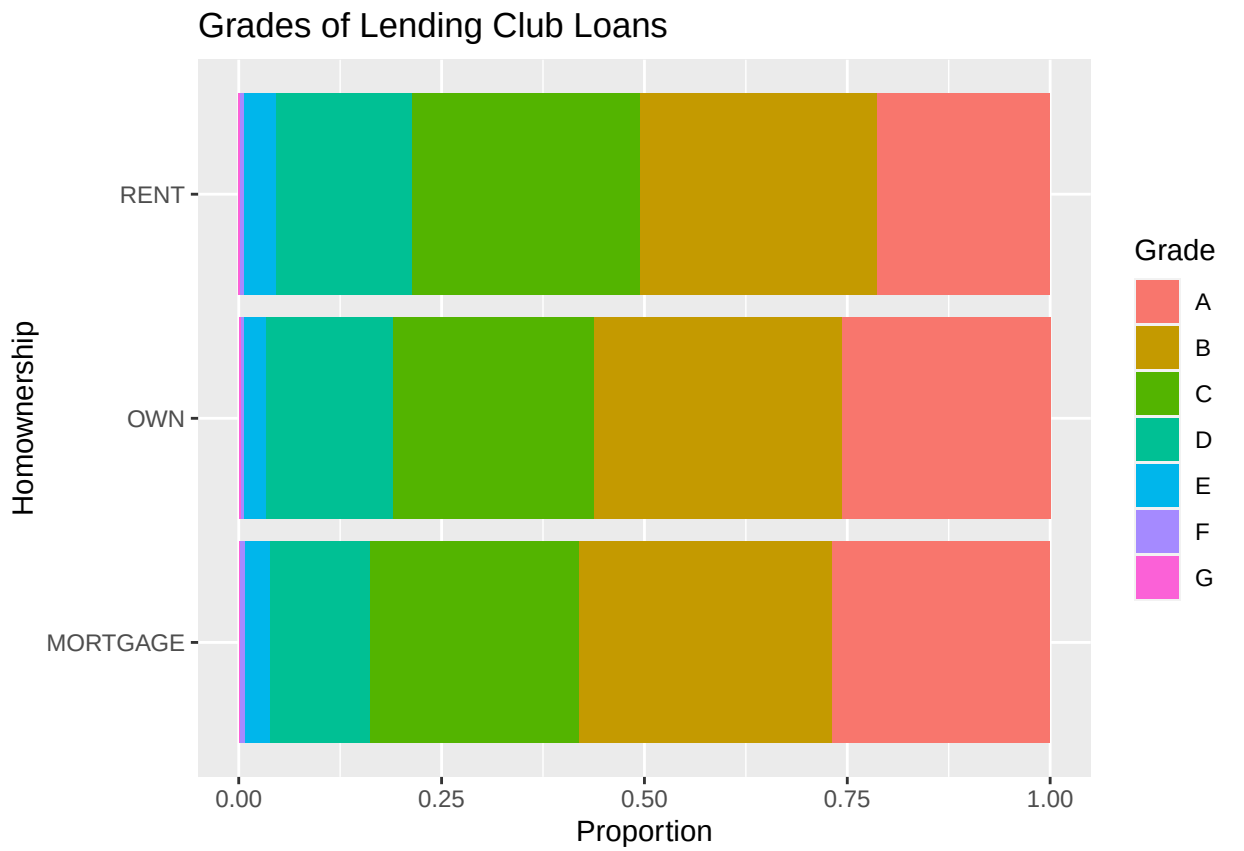

```
# Segmented Bar plot
# Enter code here
ggplot(loans, aes(x = homeownership,
                  fill = grade)) +
  geom_bar()
```



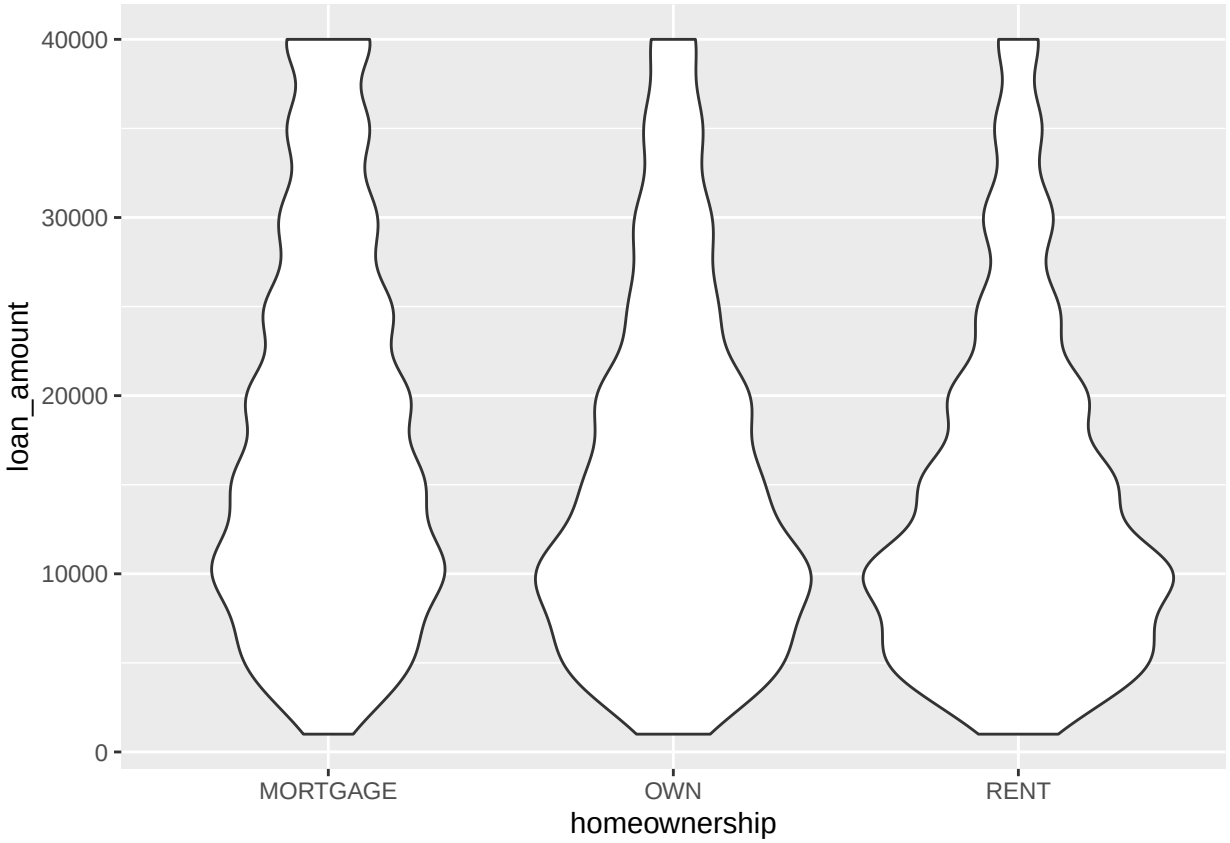

```
# Segmented Bar plot with equal heights for better comparison
# Enter code here
ggplot(loans, aes(x = homeownership, fill = grade)) +
  geom_bar(position = "fill")
```




```
# Customising bar plots
# Enter code here
ggplot(loans, aes(y = homeownership, fill = grade)) + geom_bar(position = "fill") +
  labs(x = "Proportion", y = "Homownership", fill = "Grade", title = "Grades of Lending Club Loans")
```

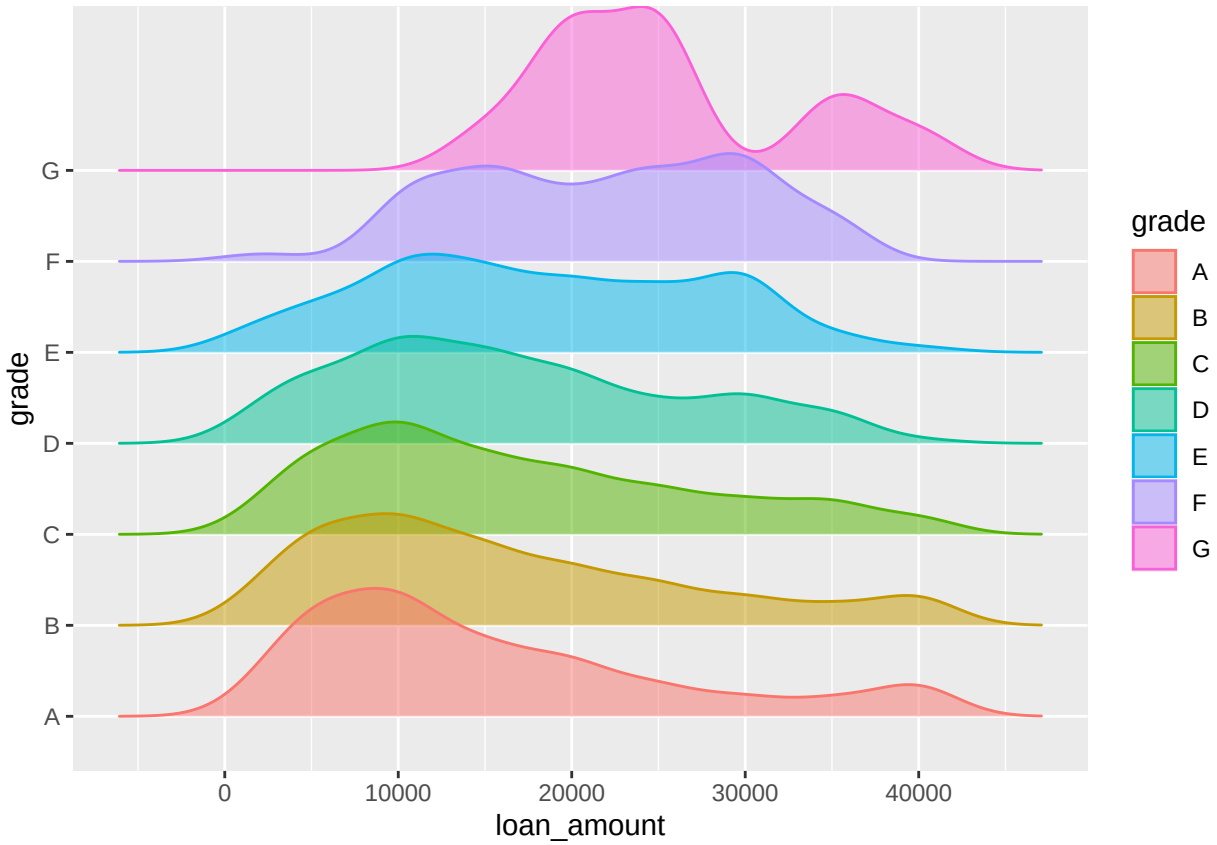



```
# Violin plots
# Enter code here
ggplot(loans, aes(x = homeownership, y = loan_amount)) +
  geom_violin()
```




```
# Ridge plots
# Enter code here
#install.packages("ggridges")
library(ggridges)
library(ggplot2)
ggplot(loans, aes(x = loan_amount, y = grade, fill = grade, colour = grade)) +
  geom_density_ridges(alpha = 0.5)
```

```
## 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