Challenge-7

Lew Bo Cong

2023-10-02

II. Code to edit and execute using the Code-along-7.Rmd file

A. All about ggplot2 package

1. The Palmer Penguins (Slide #6)

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     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 error
```

```
library(palmerpenguins)
glimpse(penguins)
```

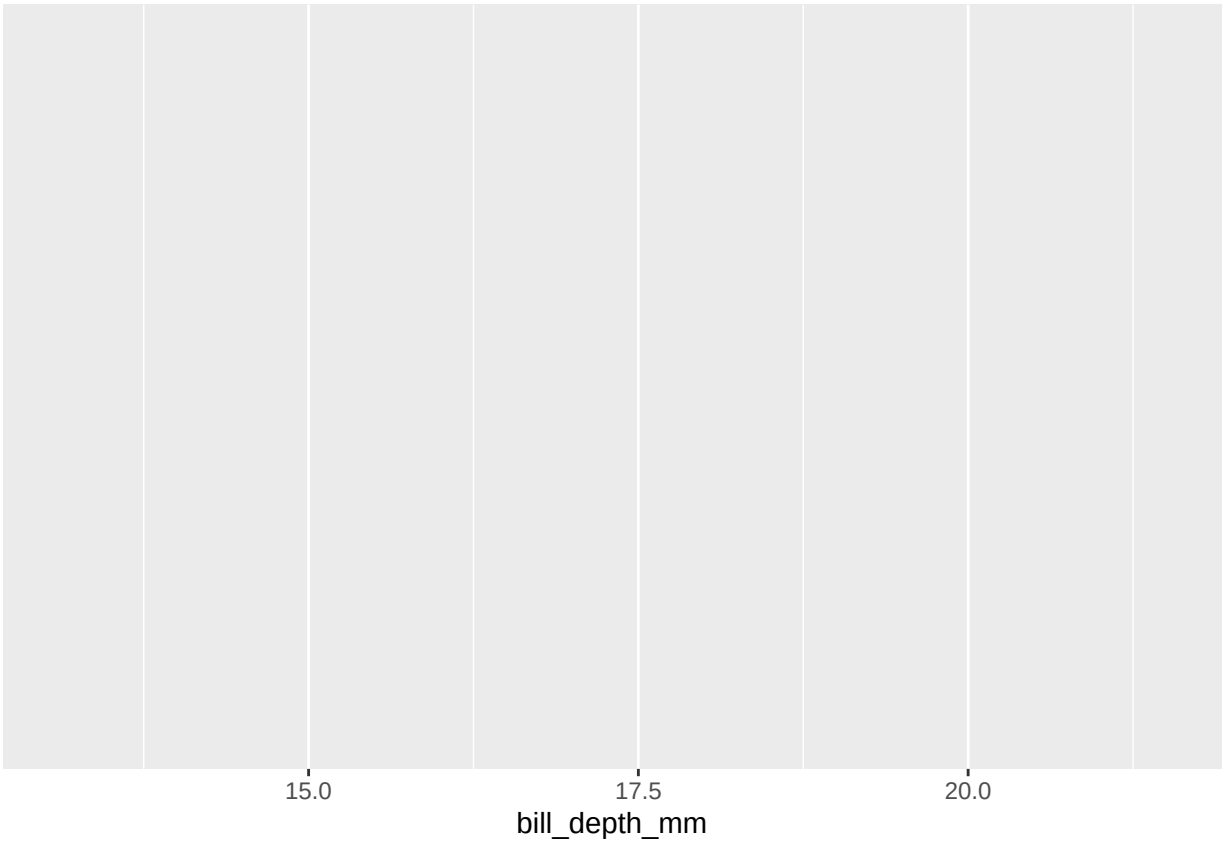
```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <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~
```

2. Palmer Penguins: Plot recreation (Slide #8-18)

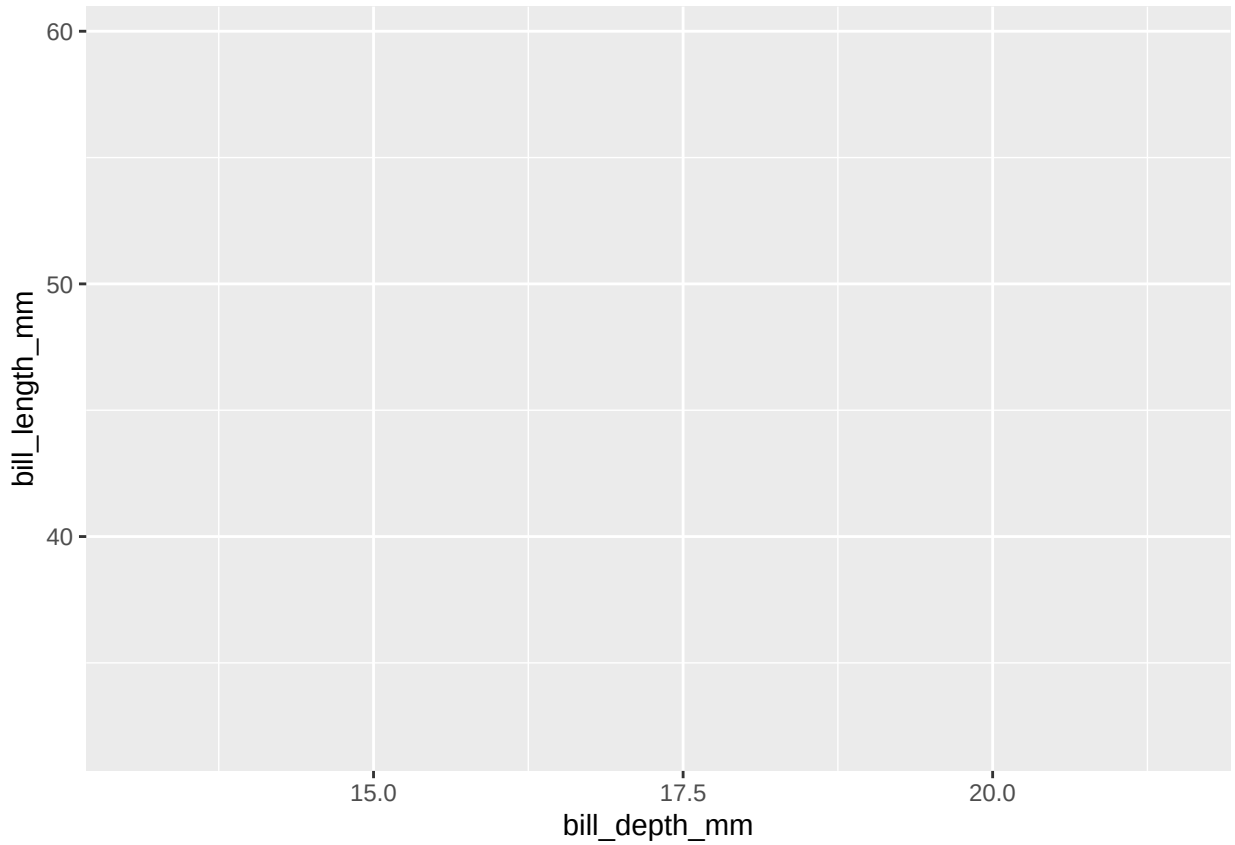
```r
# Step 1
ggplot(data = penguins)
```



```r
# Step 2
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm))
```

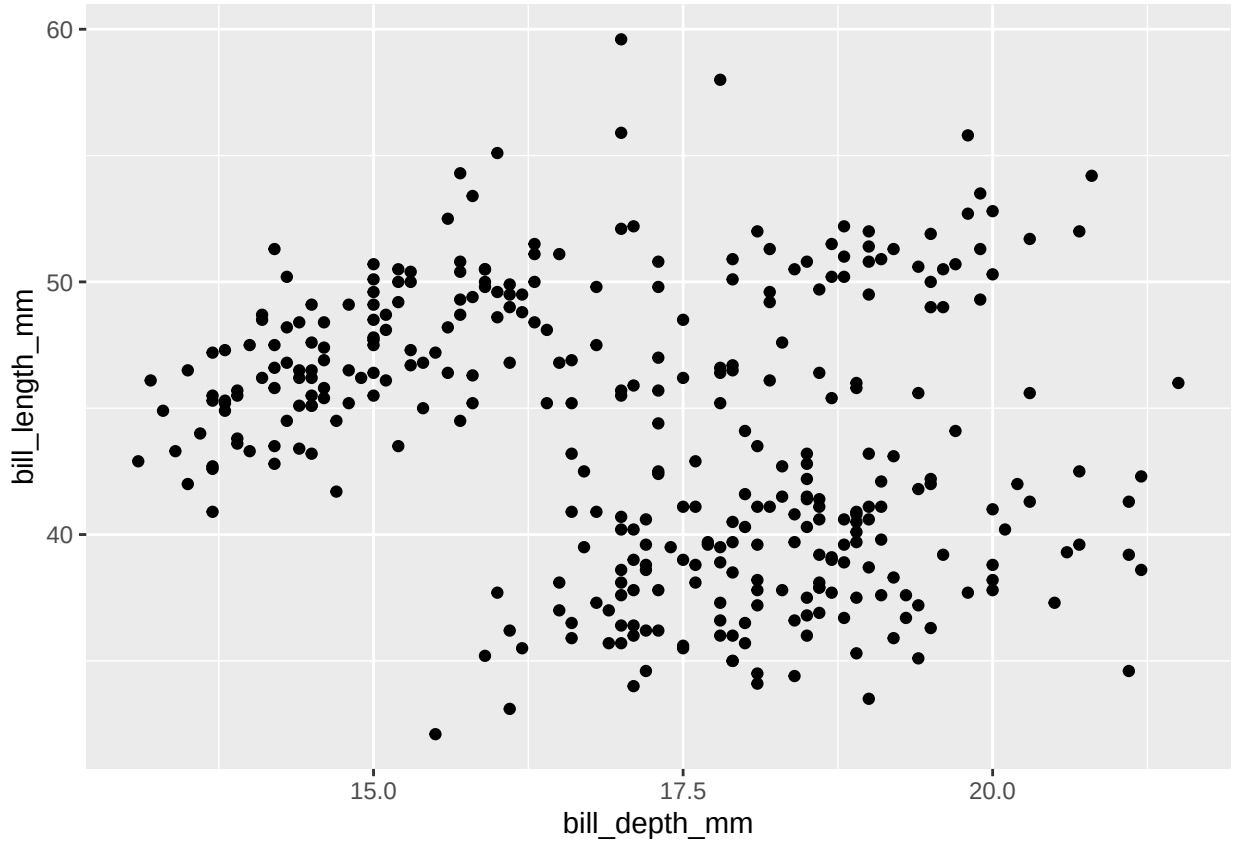



```
# Step 3
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm))
```



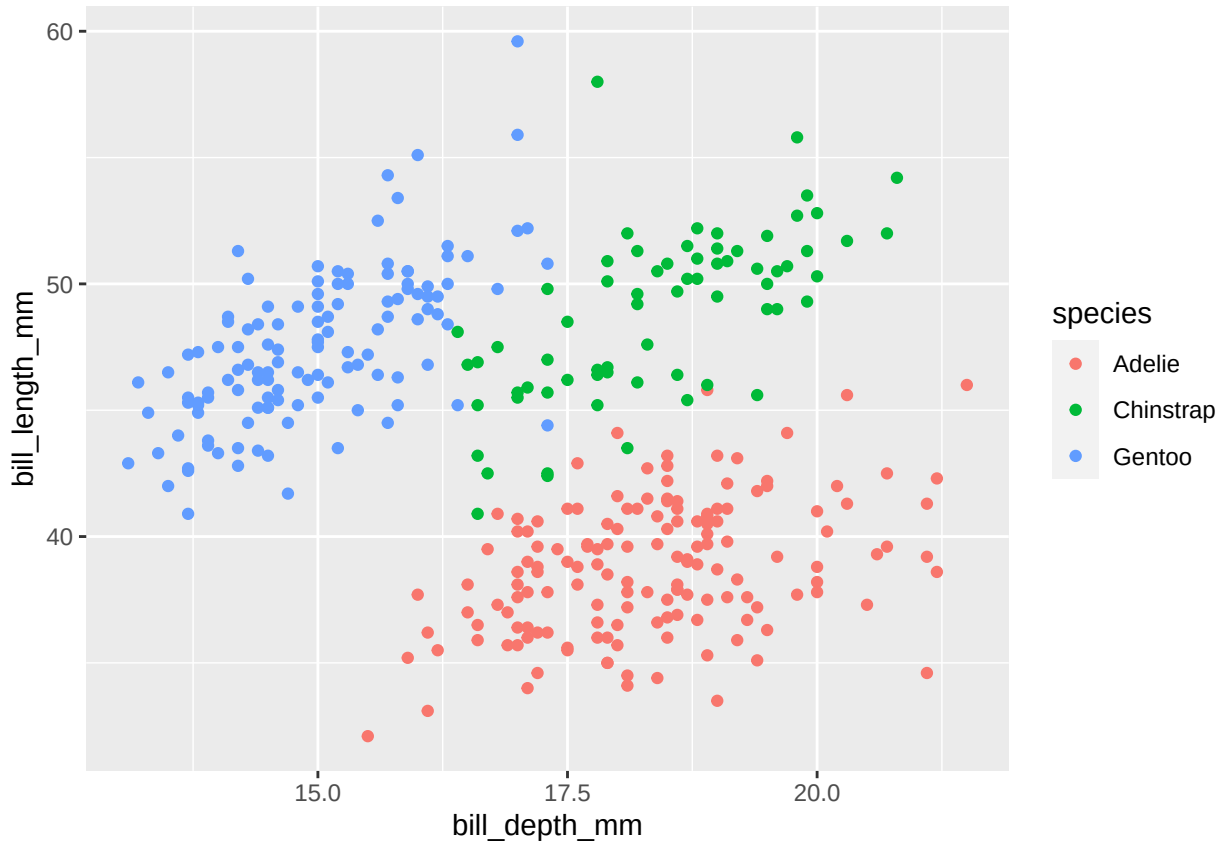

```
# Step 4
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm)) +
  geom_point()
```

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



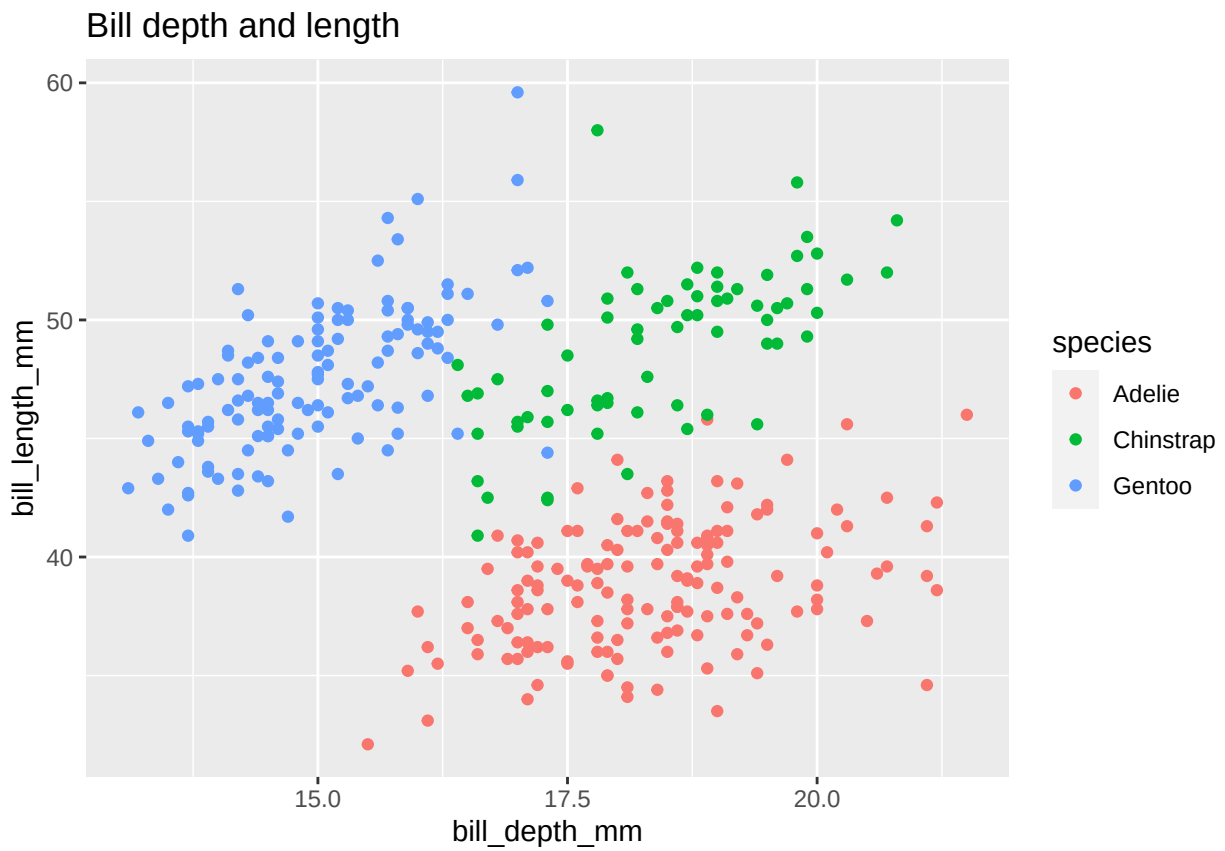

```
# Step 5
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point()
```

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



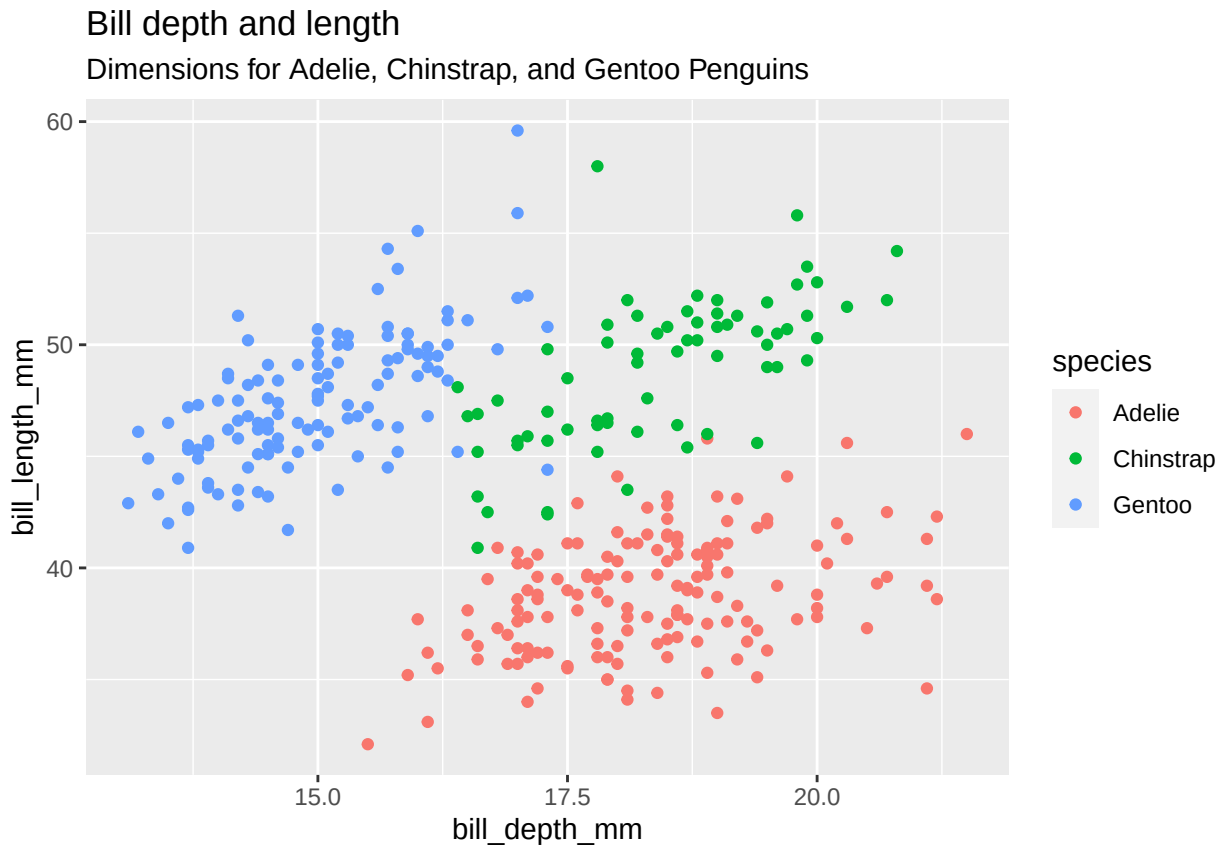

```
# Step 6
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length")
```

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



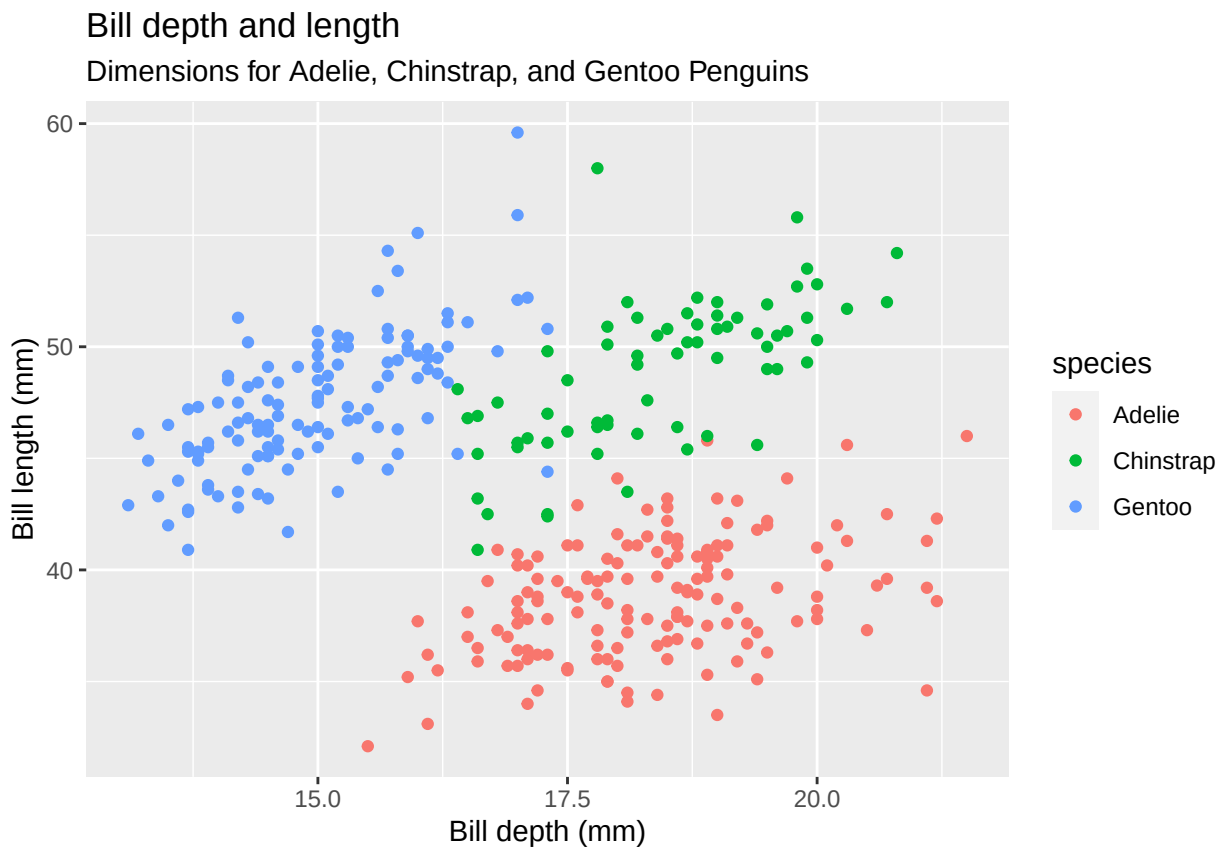

```
# Step 7
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins")
```

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



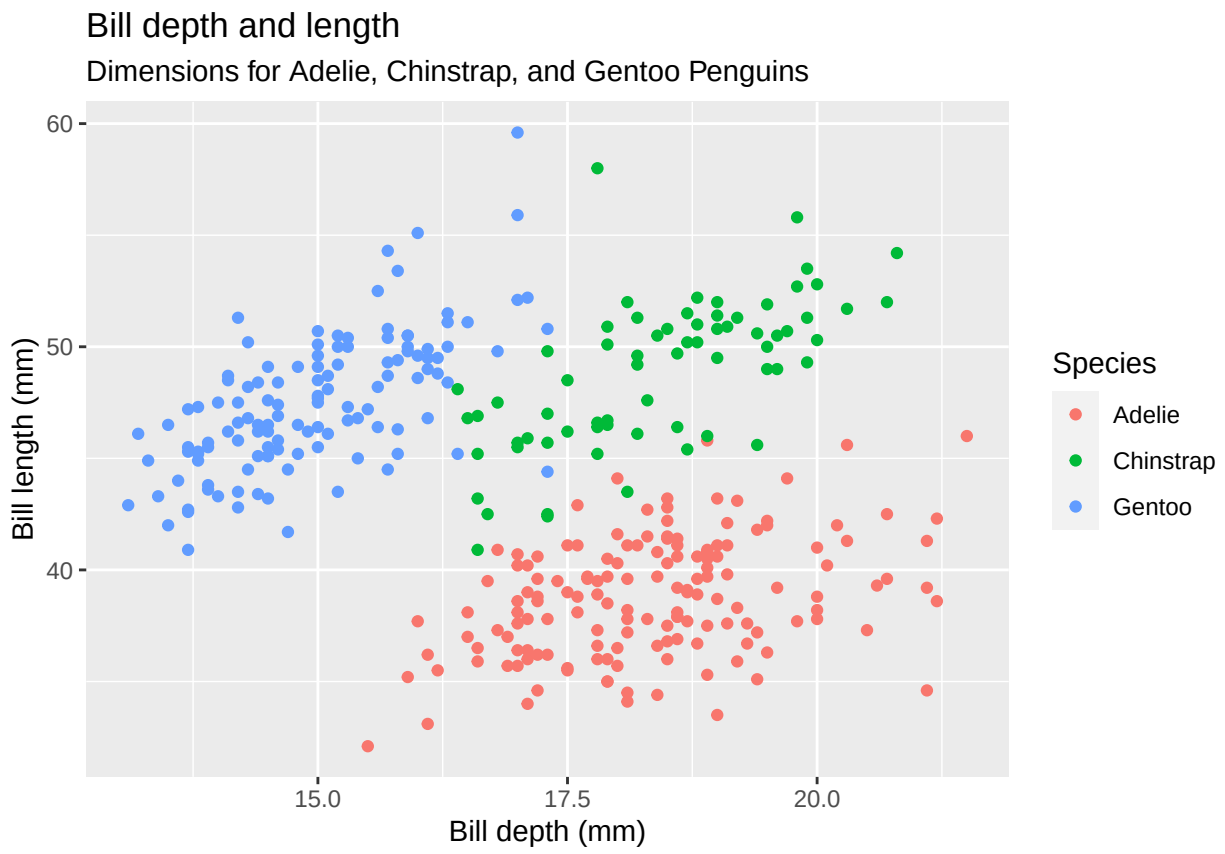

```
# Step 8
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
       x = "Bill depth (mm)",
       y = "Bill length (mm)")
```

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



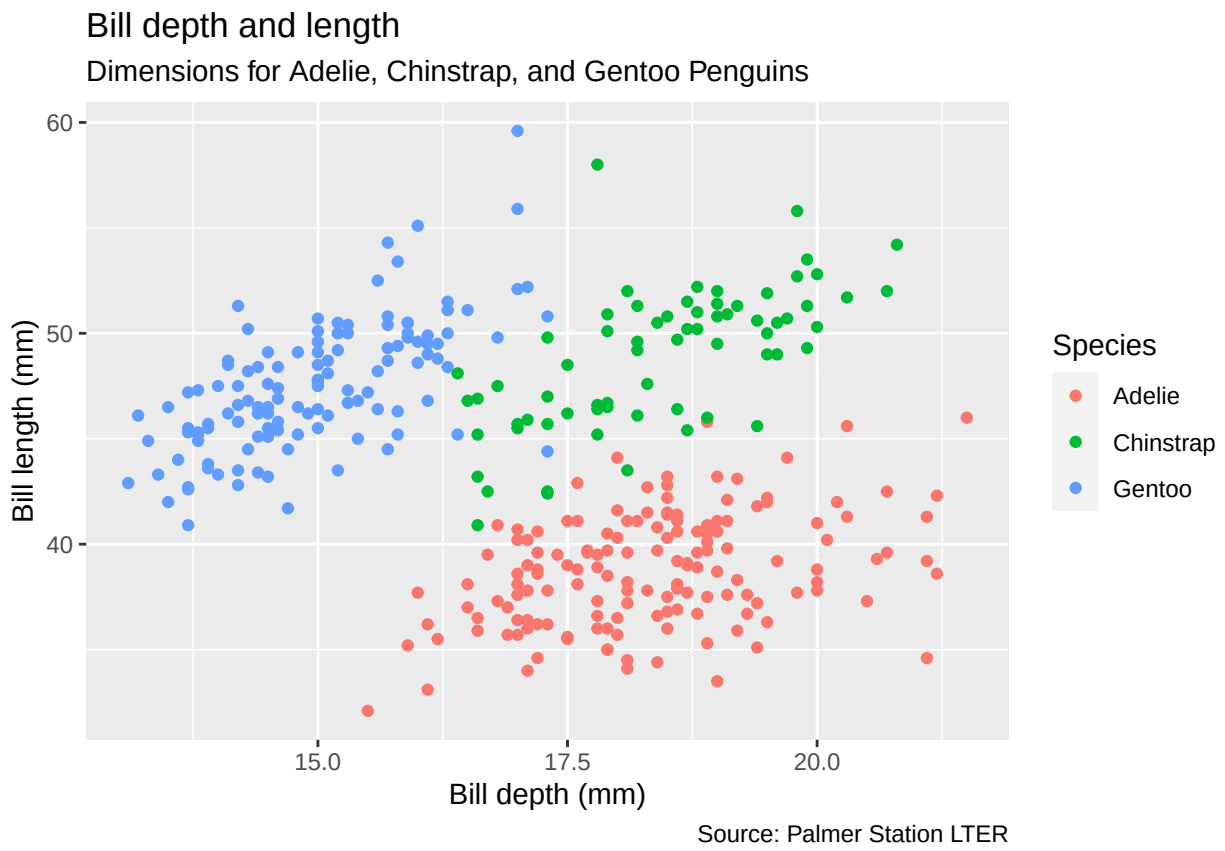

```
# Step 9
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
       x = "Bill depth (mm)",
       y = "Bill length (mm)",
       colour = "Species")
```

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



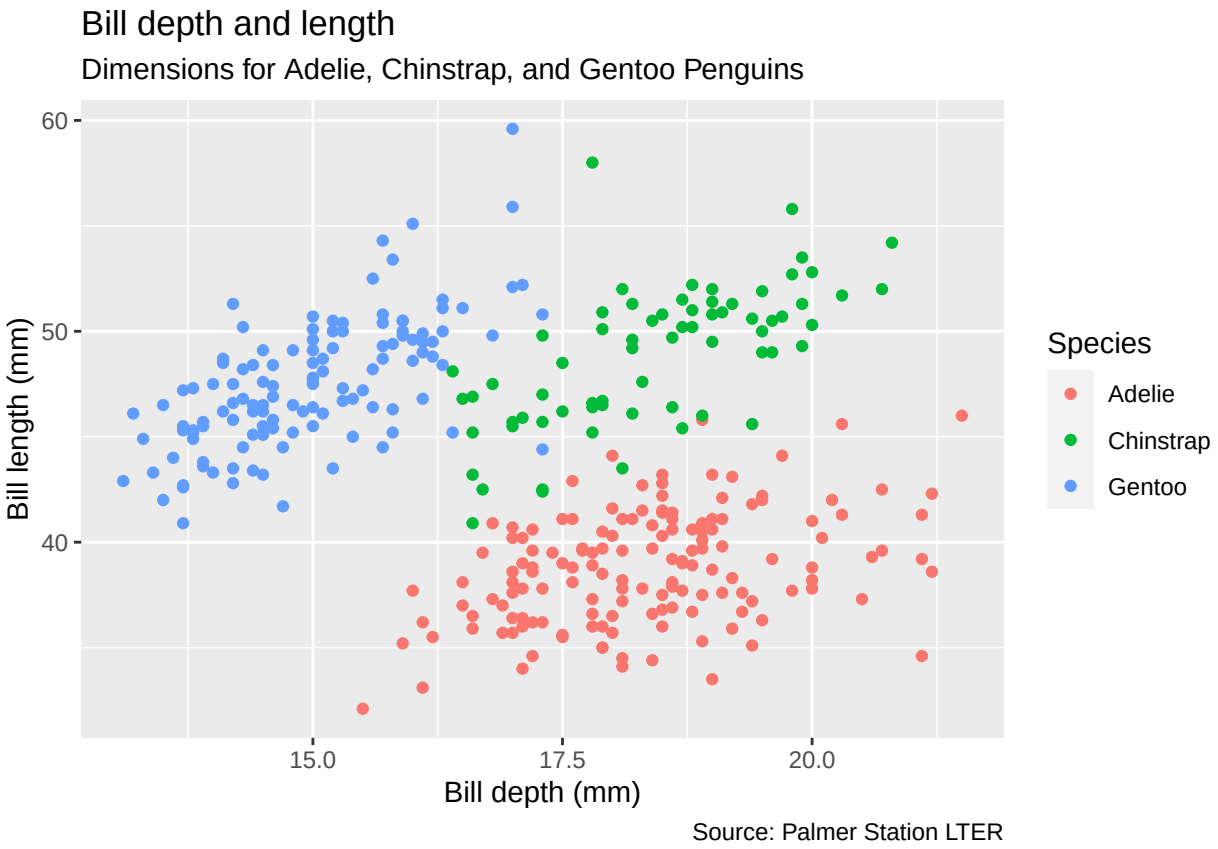

```
# Step 10
ggplot(data = penguins,
       mapping = aes(x = bill_depth_mm,
                     y = bill_length_mm,
                     colour = species)) +
  geom_point() +
  labs(title = "Bill depth and length",
       subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
       x = "Bill depth (mm)",
       y = "Bill length (mm)",
       colour = "Species",
       caption = "Source: Palmer Station LTER")
```

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




```
# Step 11
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",
       scale_colour_viridis_d())
```

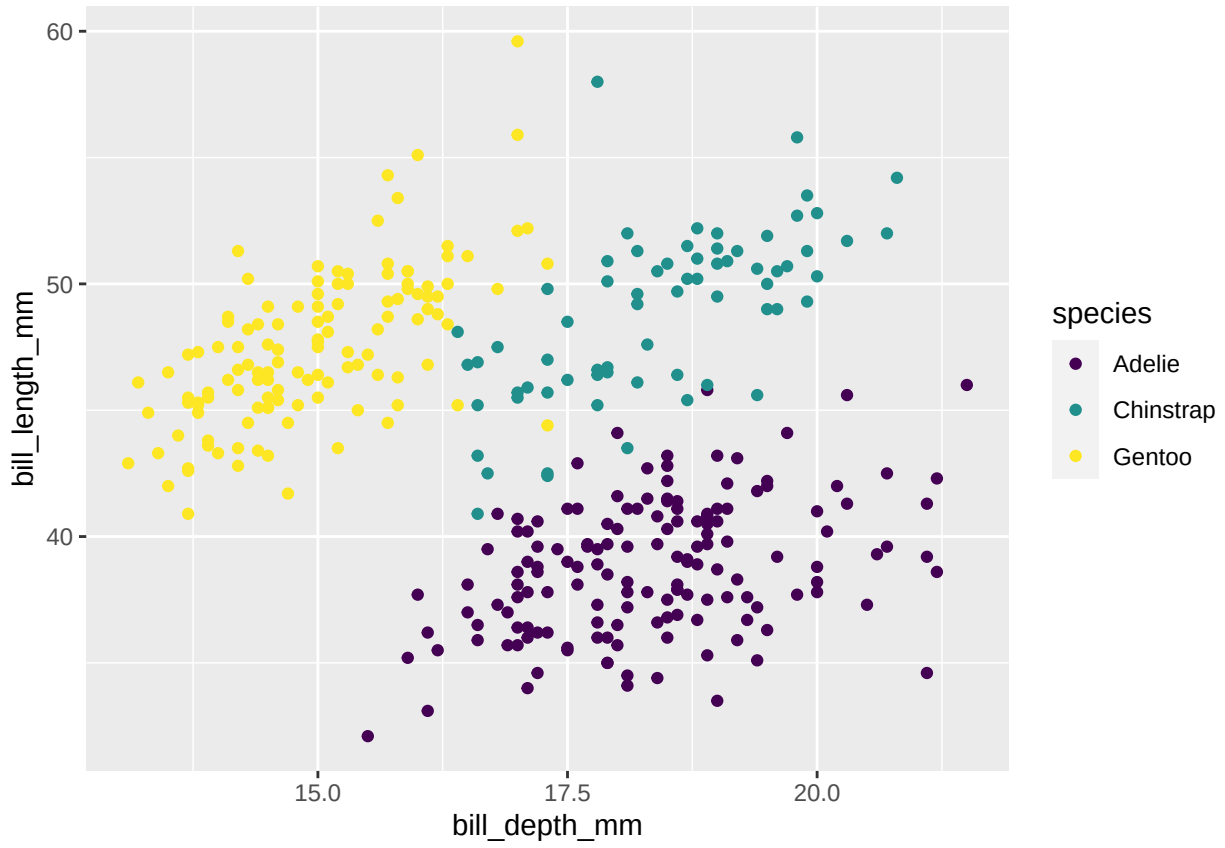
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



3. Palmer Penguins: Argument names (Slide #20)

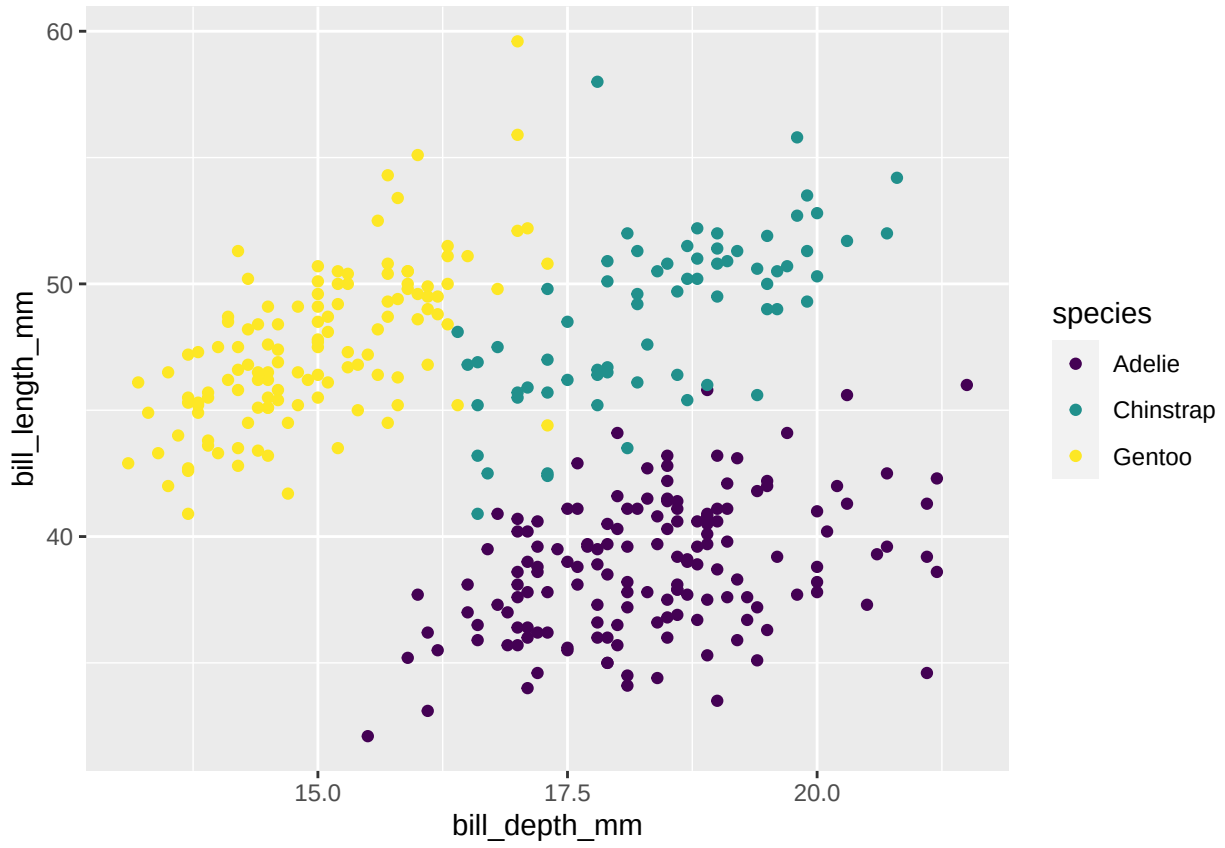
```
ggplot(penguins,
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species)) +
  geom_point() +
  scale_colour_viridis_d()
```

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




```
ggplot(penguins) + # Data layer
       aes(x = bill_depth_mm,
           y = bill_length_mm,
           colour = species) + # Aesthetics layer
              geom_point() + # Geometric layer
                scale_colour_viridis_d()
```

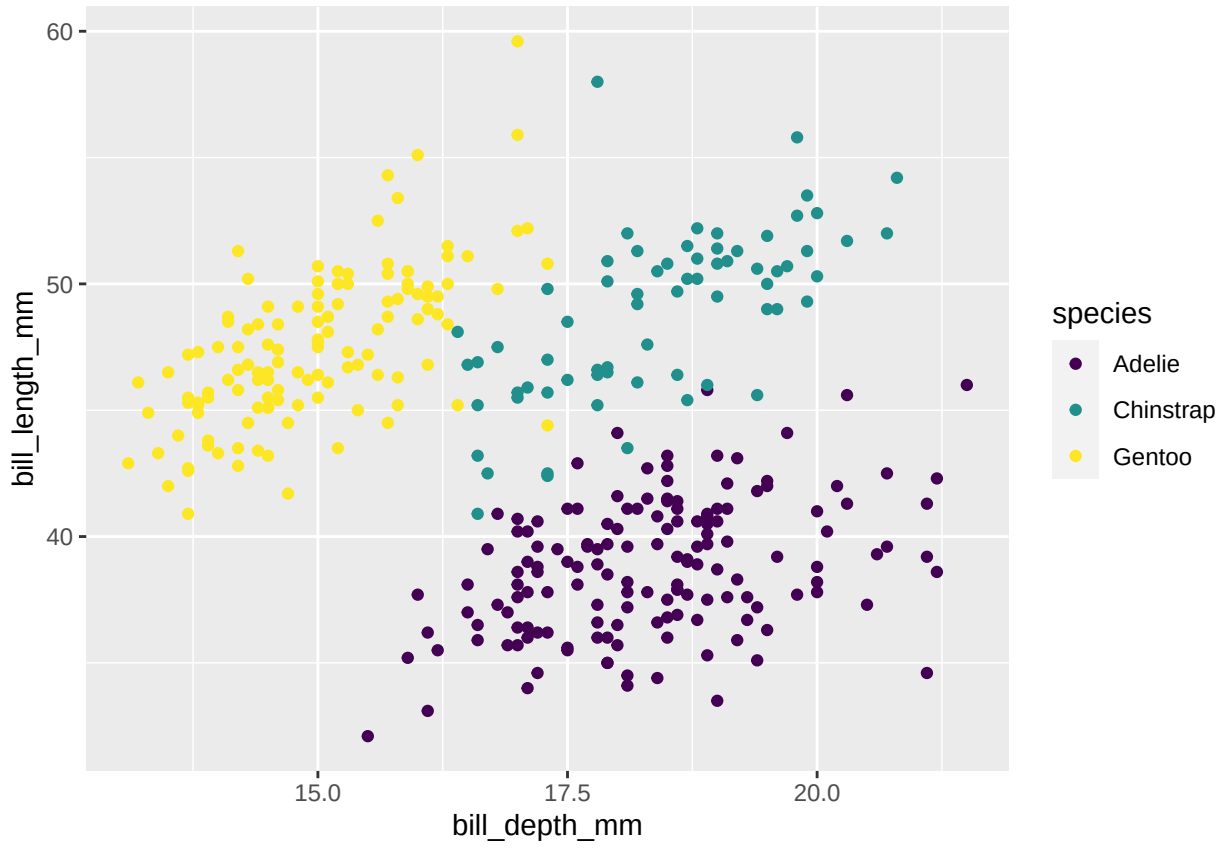
```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```



4. Aesthetics options (Slide #22-26)

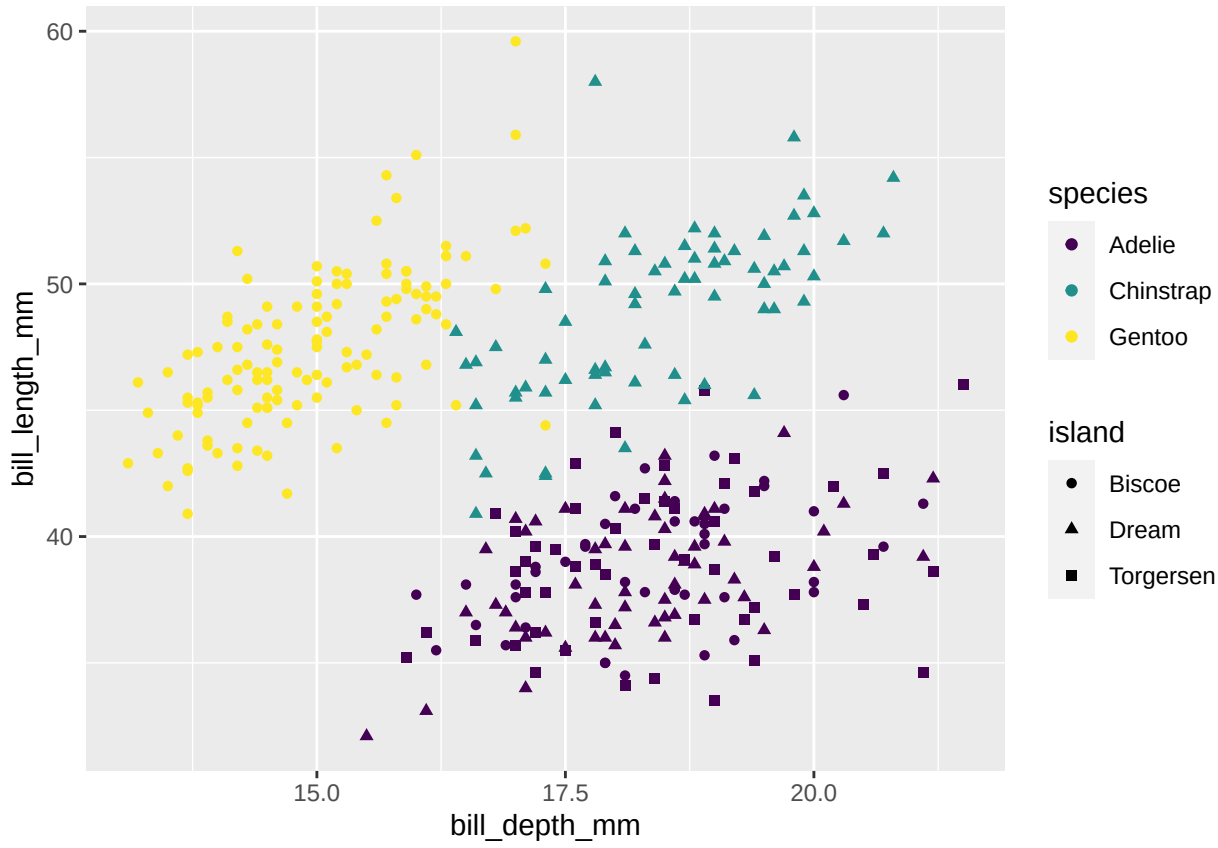
```
# Colour
ggplot(penguins) + aes(x = bill_depth_mm,
                       y = bill_length_mm,
                       colour = species) +
  geom_point() + scale_colour_viridis_d()
```

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



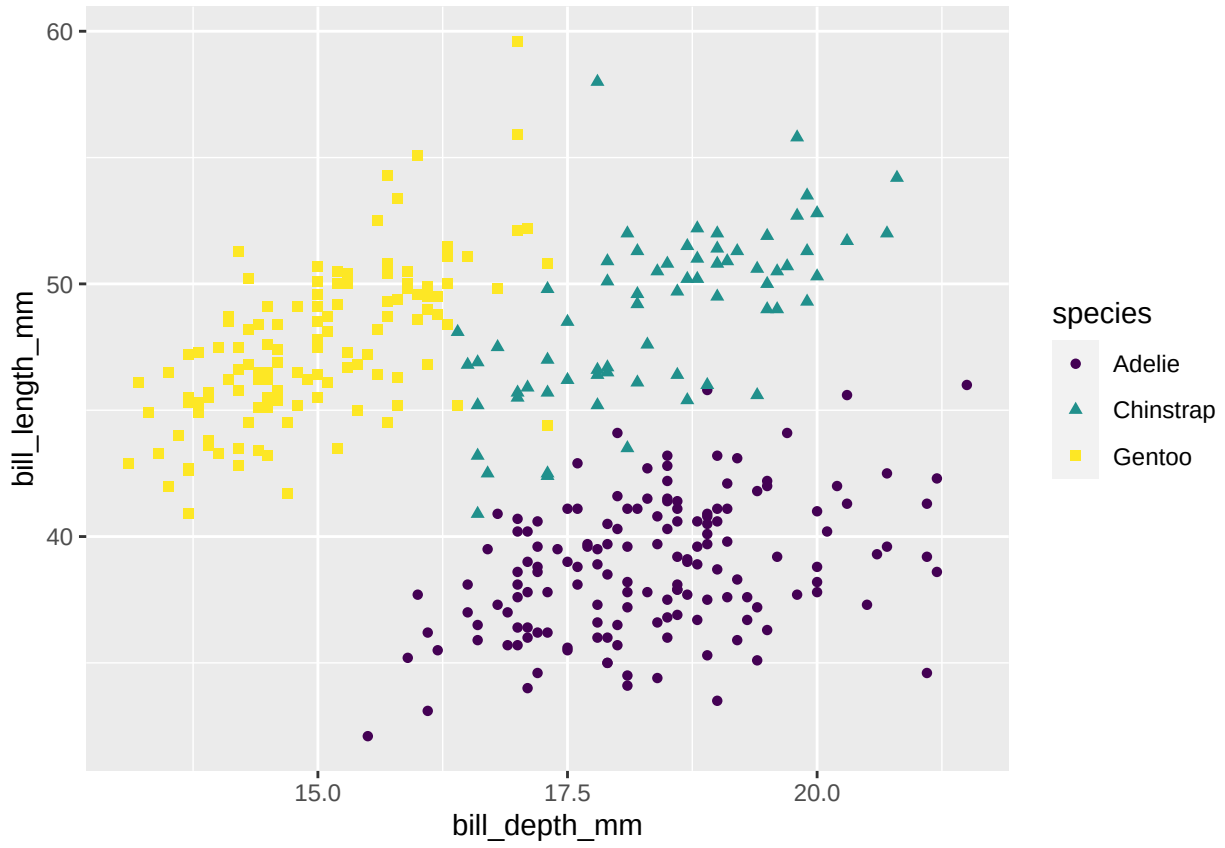

```
# Shape 1
ggplot(penguins) + aes(x = bill_depth_mm,
                       y = bill_length_mm,
                       colour = species,
                       shape = island) +
  geom_point() + scale_colour_viridis_d()
```

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



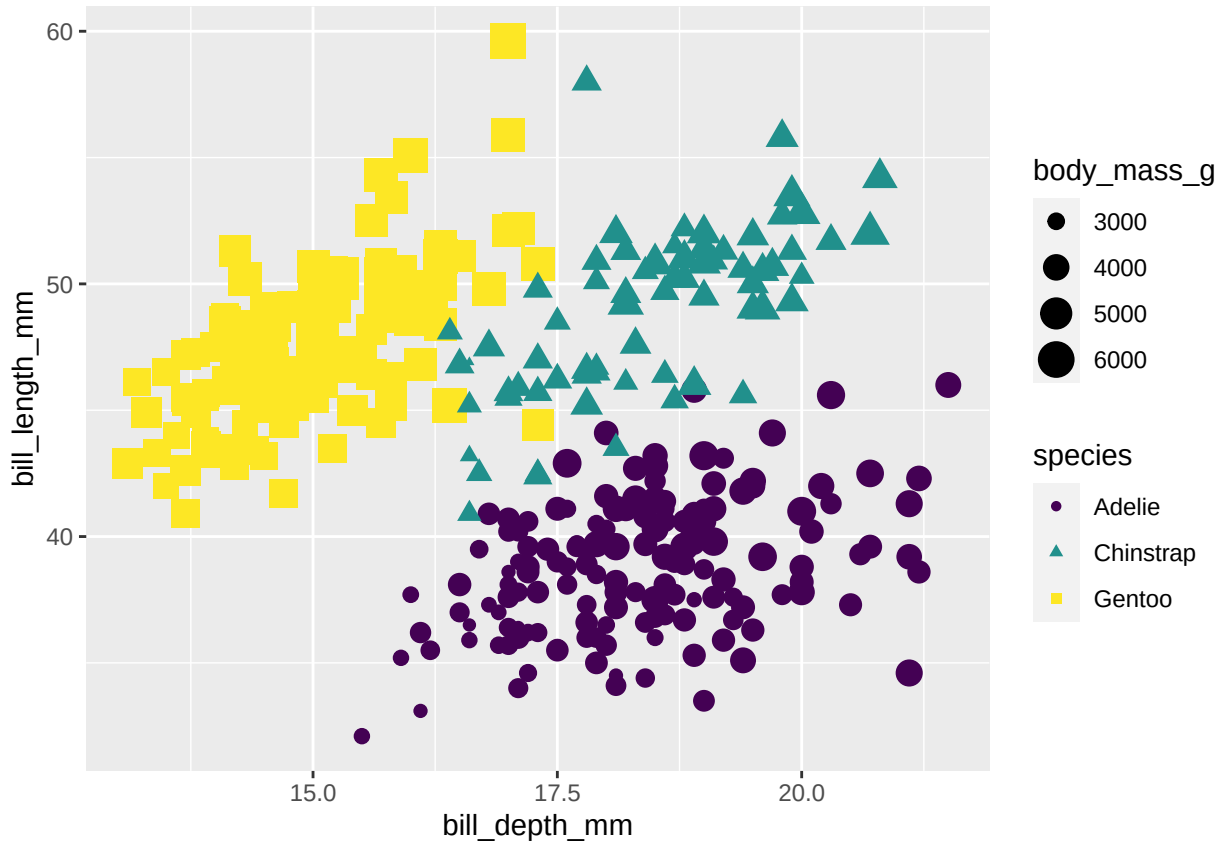

```
# Shape 2
ggplot(penguins) + aes(x = bill_depth_mm,
                       y = bill_length_mm,
                       colour = species,
                       shape = species) +
  geom_point() + scale_colour_viridis_d()
```

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



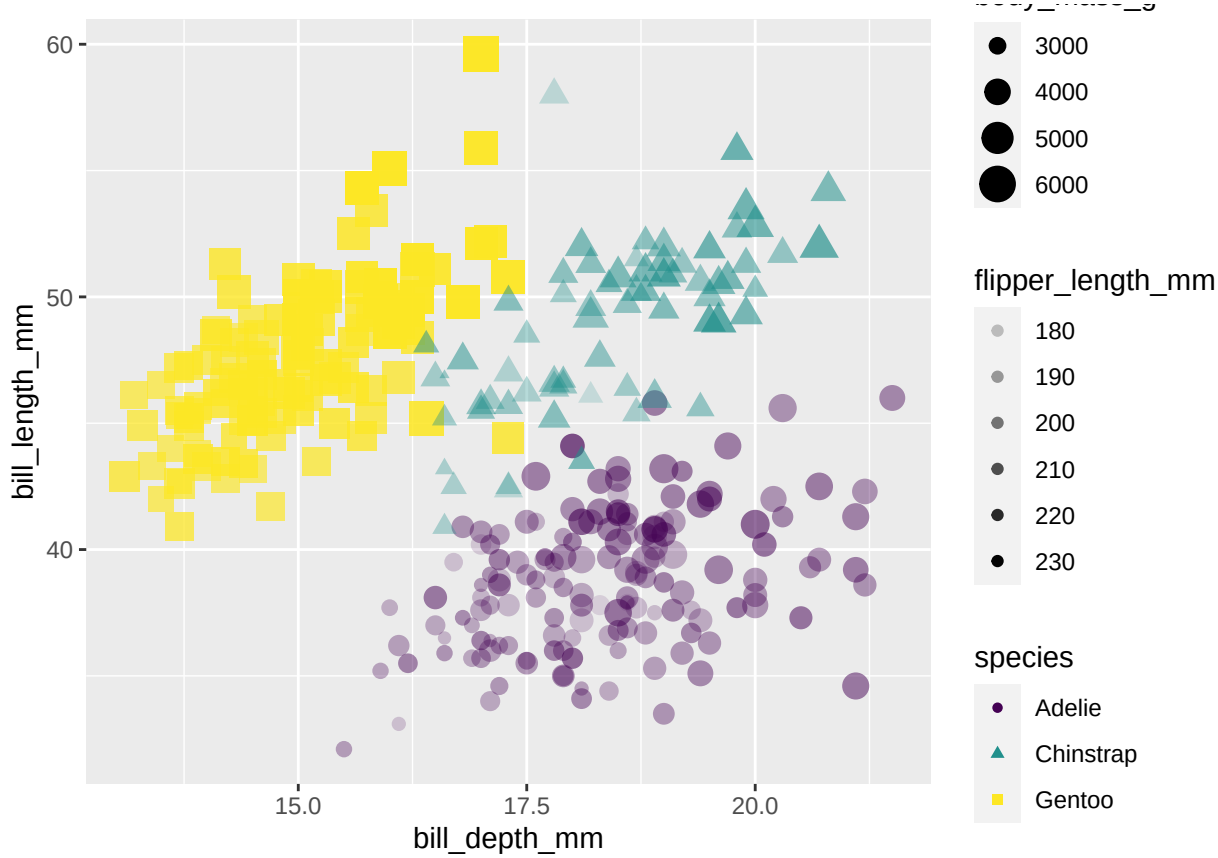

```
# Size
ggplot(penguins) + aes(x = bill_depth_mm,
                       y = bill_length_mm,
                       colour = species,
                       shape = species,
                       size = body_mass_g) +
  geom_point() + scale_colour_viridis_d()
```

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




```
# Alpha
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()
```

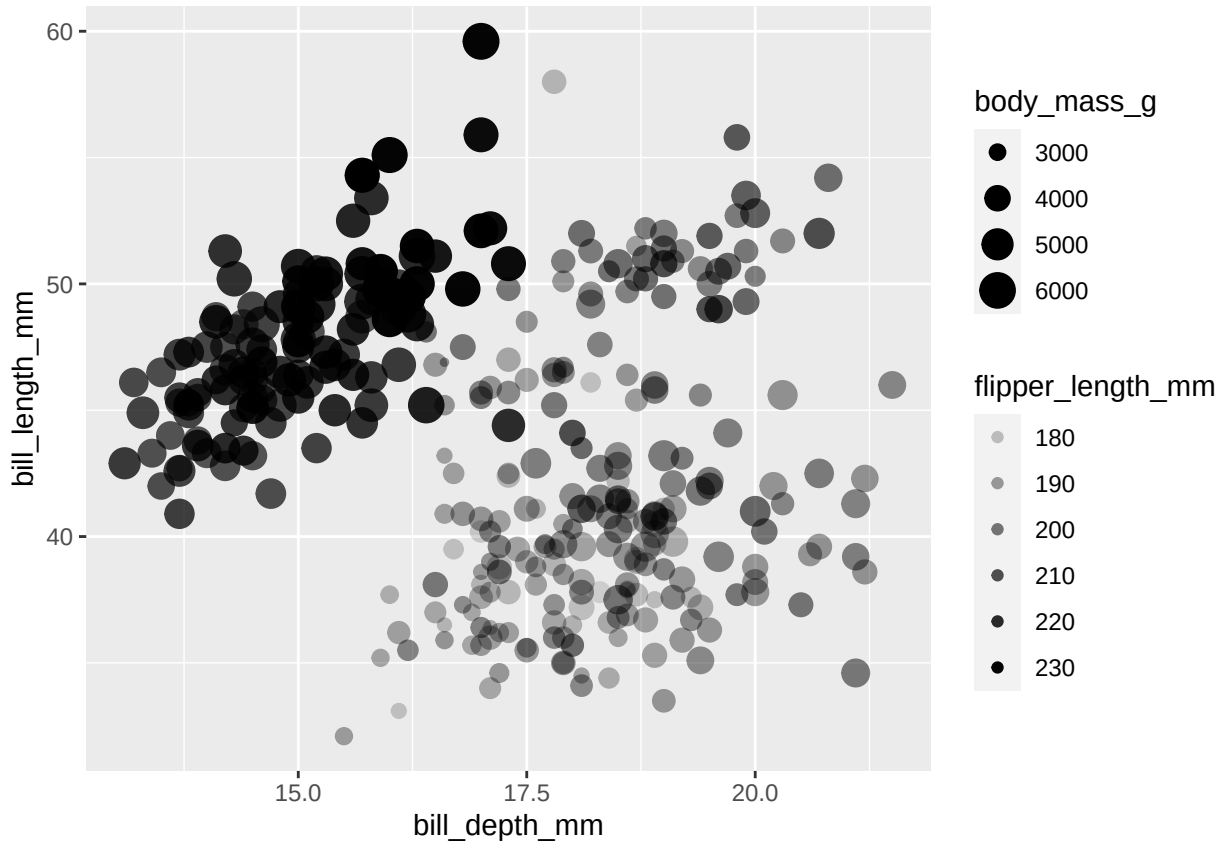
```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



5. Mapping vs Setting (Slide #28)

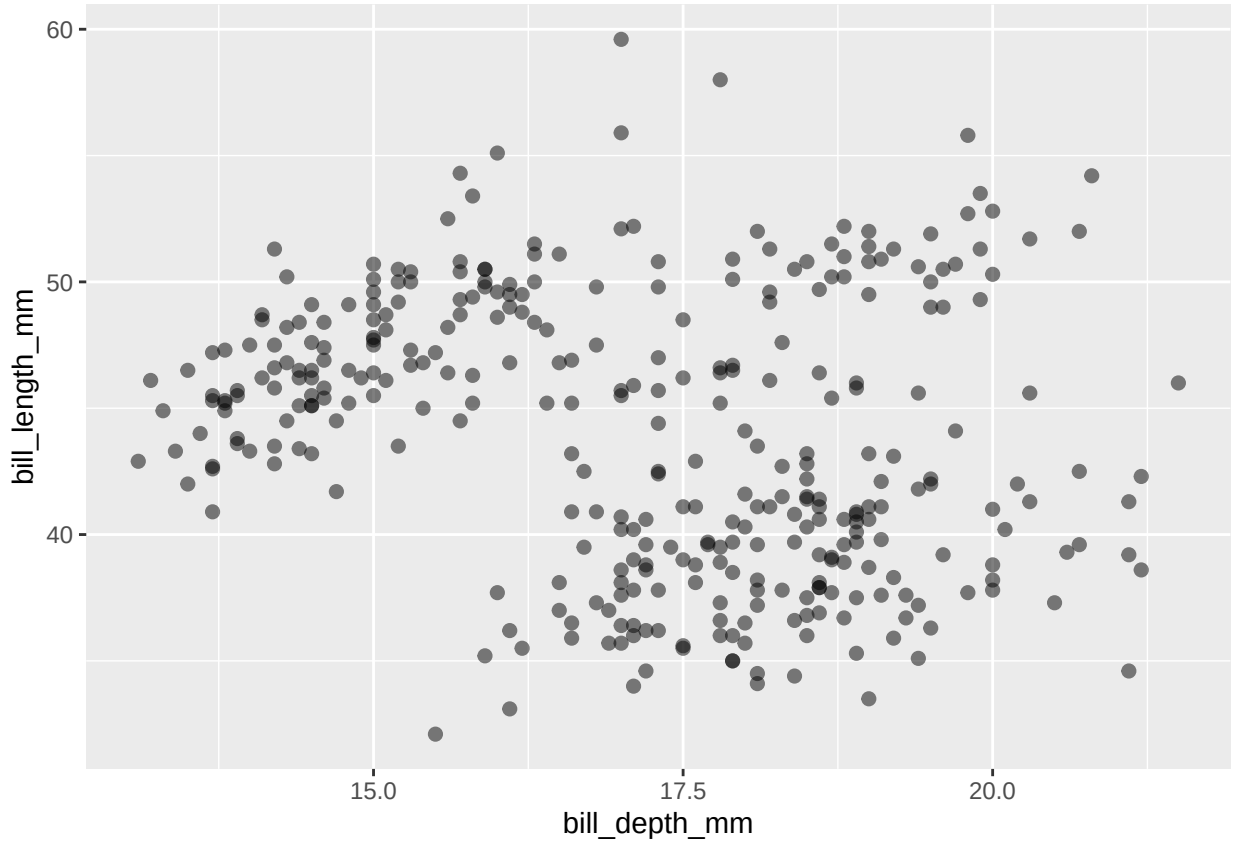
```
# Mapping
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm,
      size = body_mass_g,
      alpha = flipper_length_mm) +
  geom_point()
```

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




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

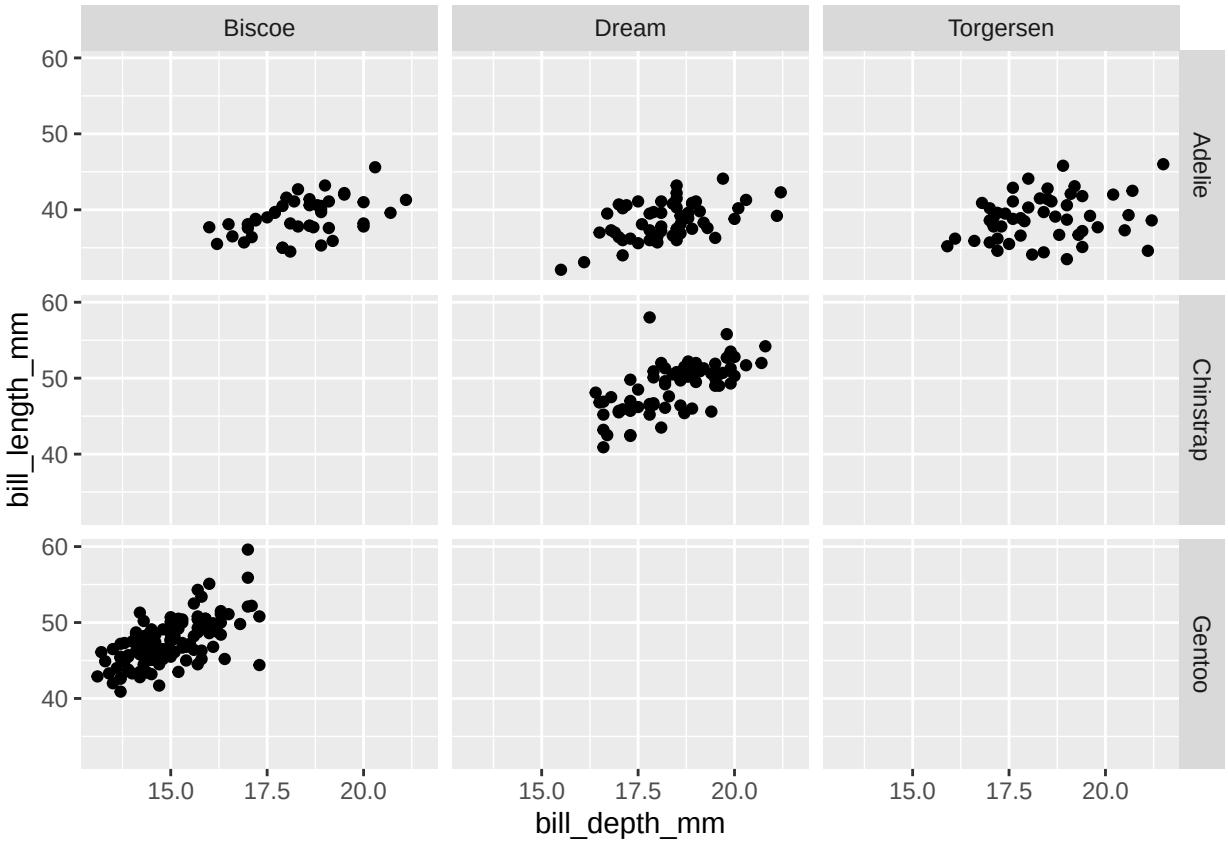
```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```



6. Faceting (Slide #29-36)

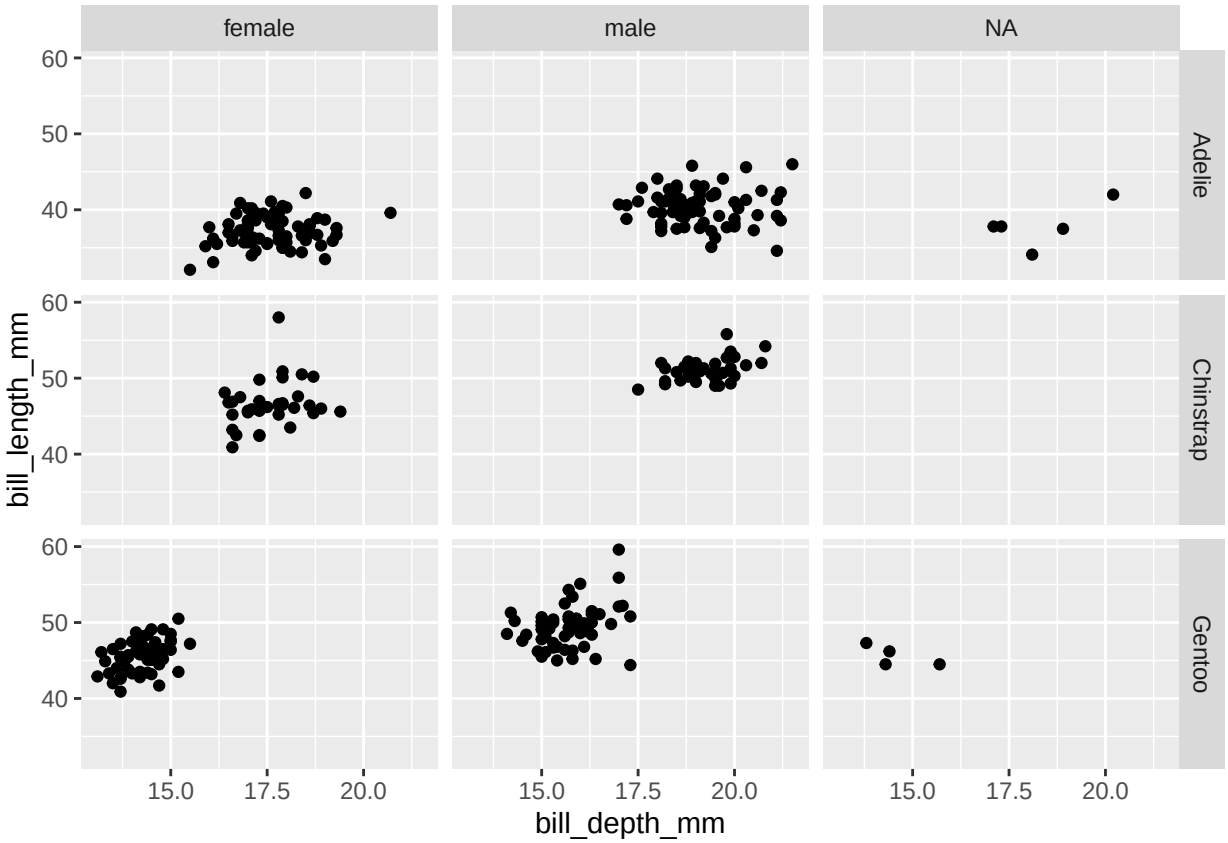
```
# Facet 1
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_grid(species ~ island) # species along the rows, island along the columns
```

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



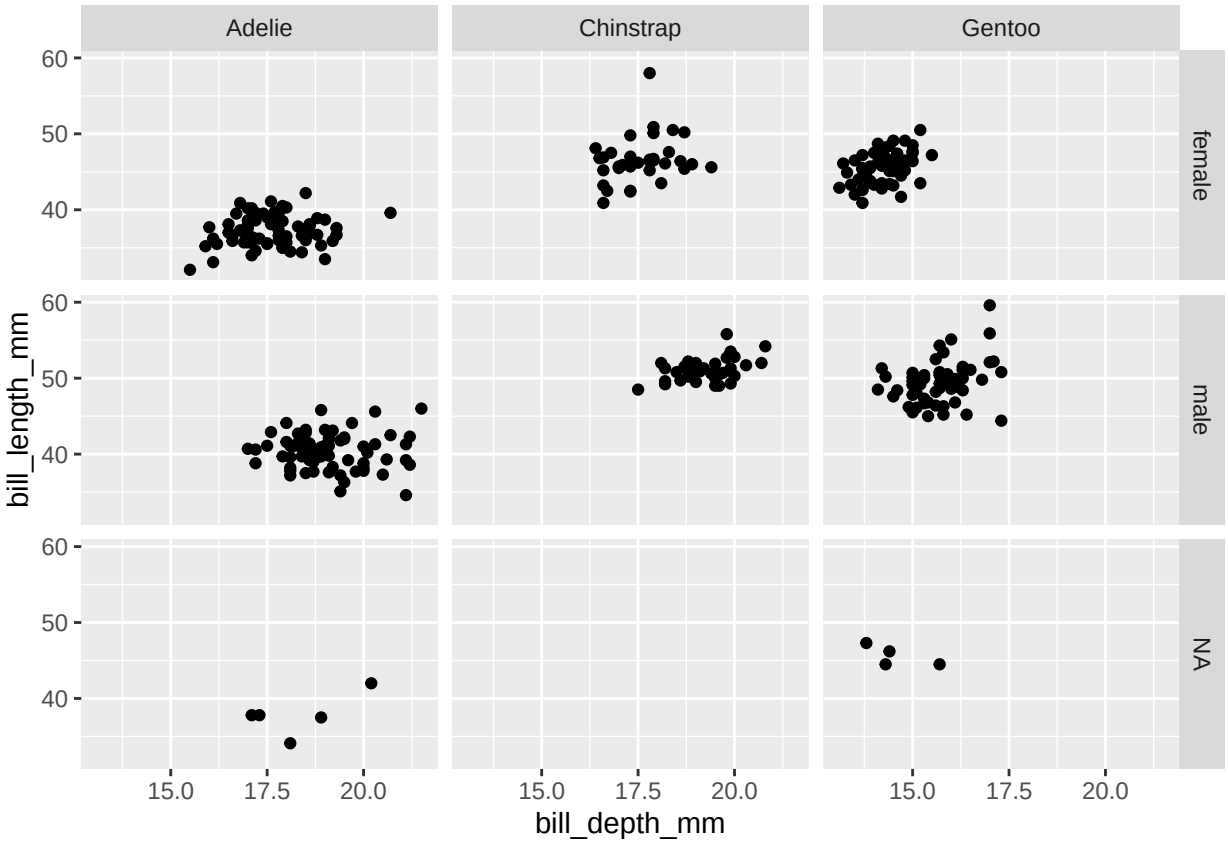

```
# Facet 2
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_grid(species ~ sex) # NA means certain obs with gender not tabulated
```

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



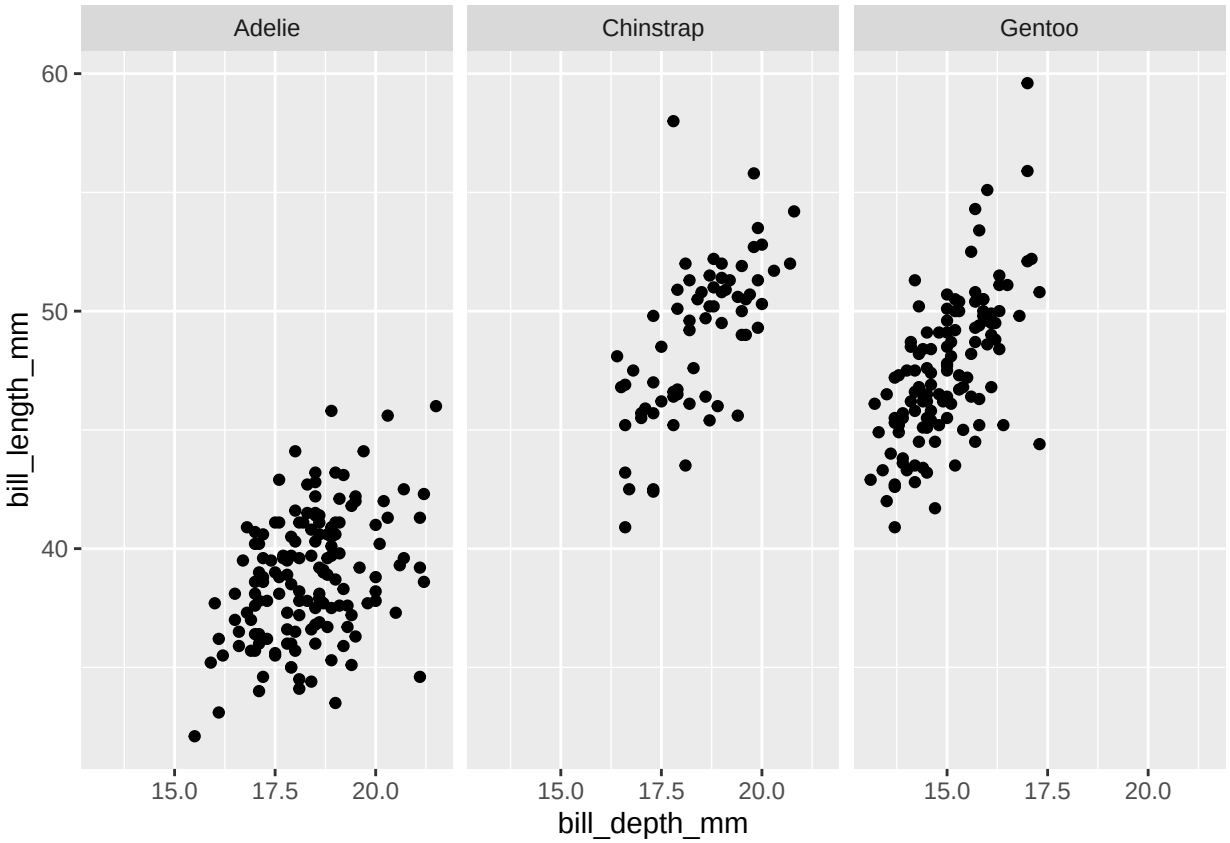

```
# Facet 3
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_grid(sex ~ species)
```

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



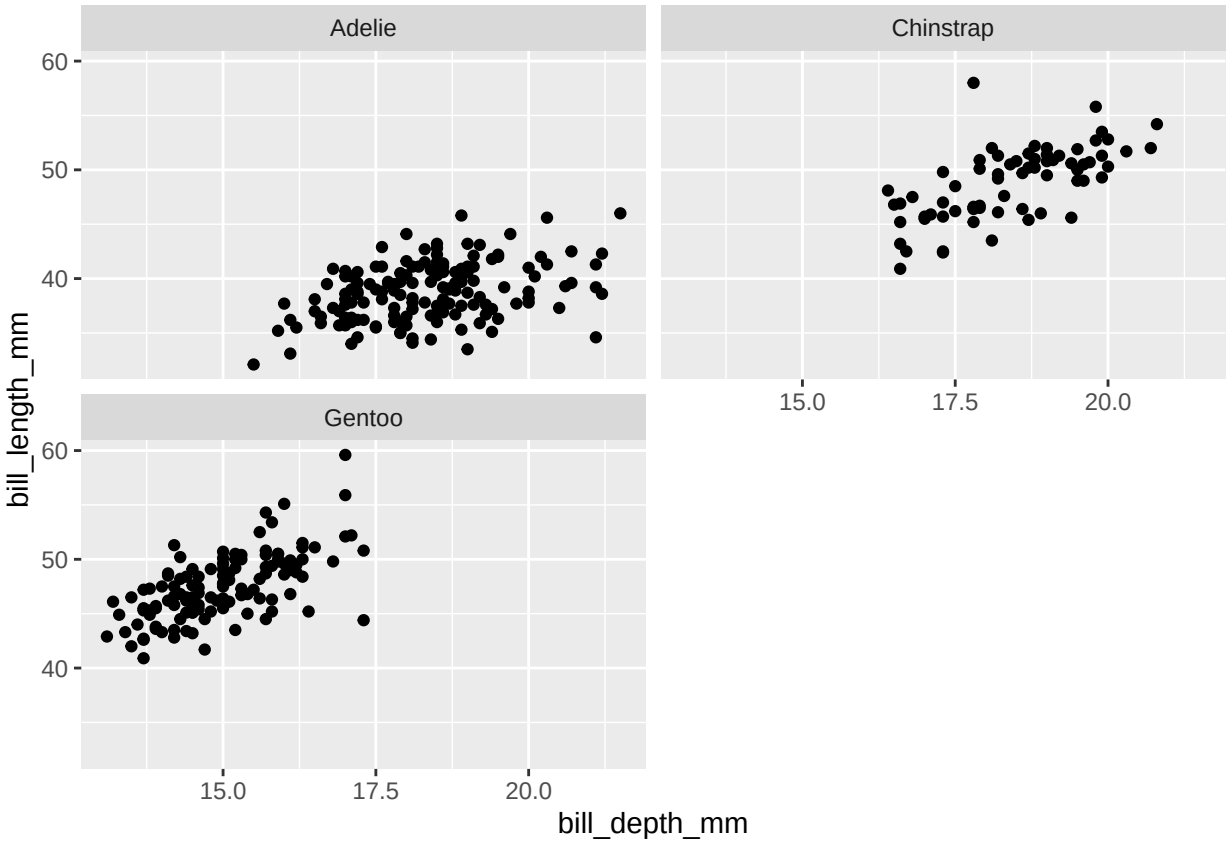

```
# Facet 4
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_wrap(~ species) # only sort by 1 variable
```

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



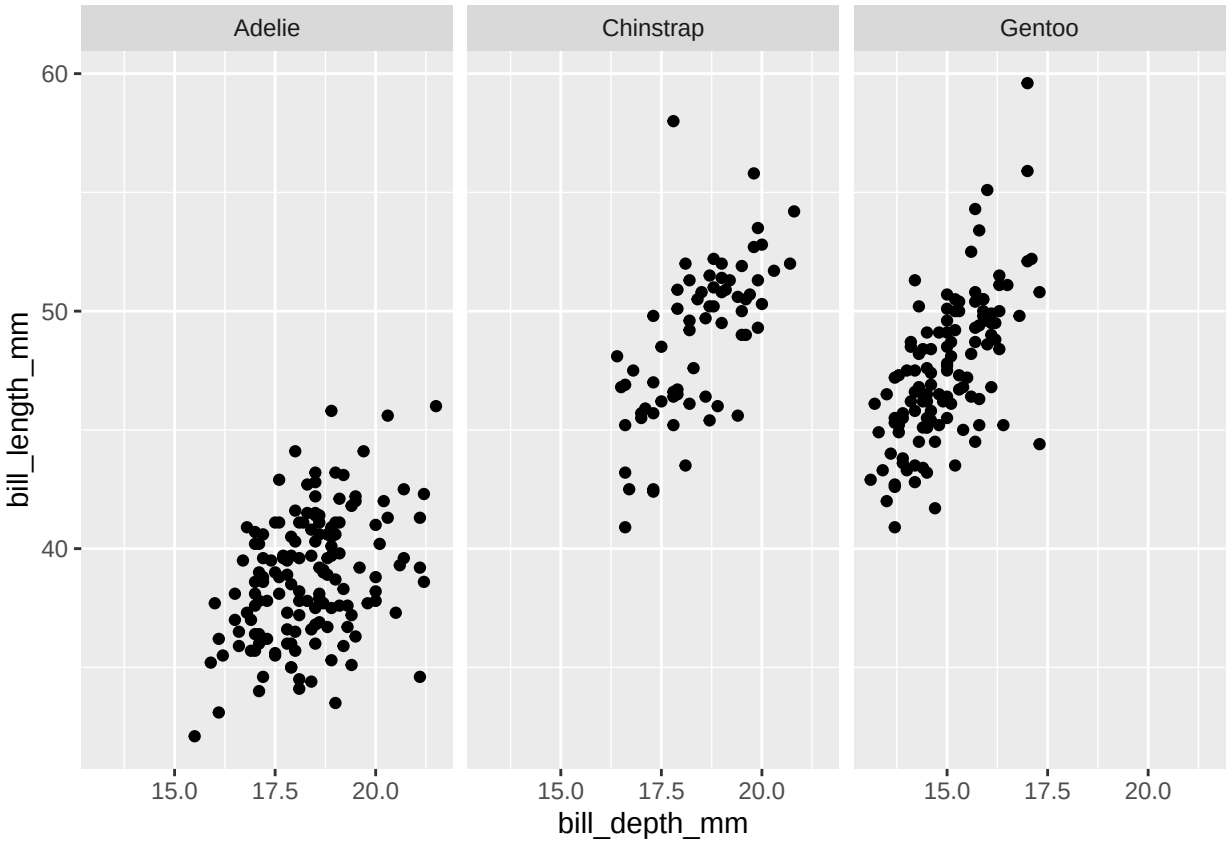

```
# Facet 5
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_wrap(~ species, ncol = 2)
```

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




```
# Facet 6
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm) +
  geom_point() +
  facet_grid(. ~ species) # grid needs variable for row & column,
```

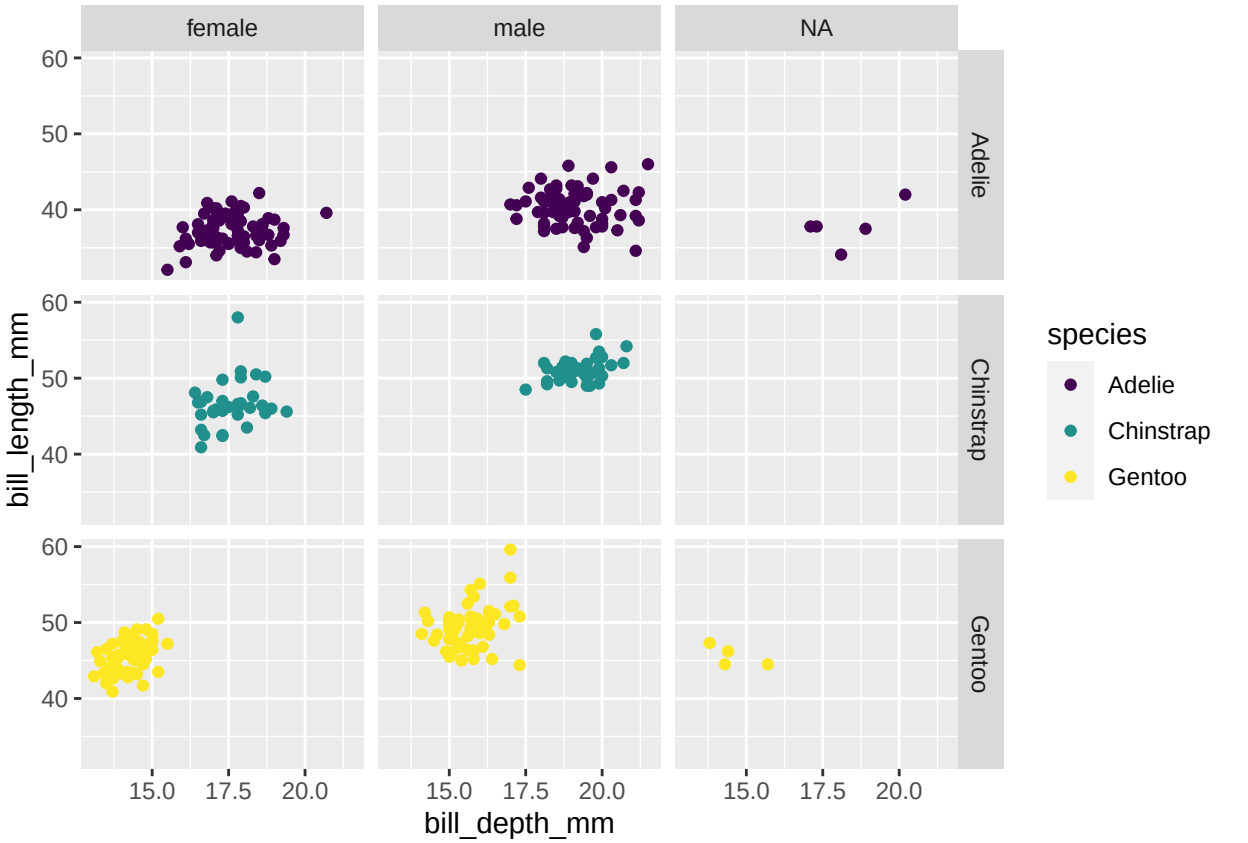
```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```




```
                    # use '.' if don't want to specify one of them
```

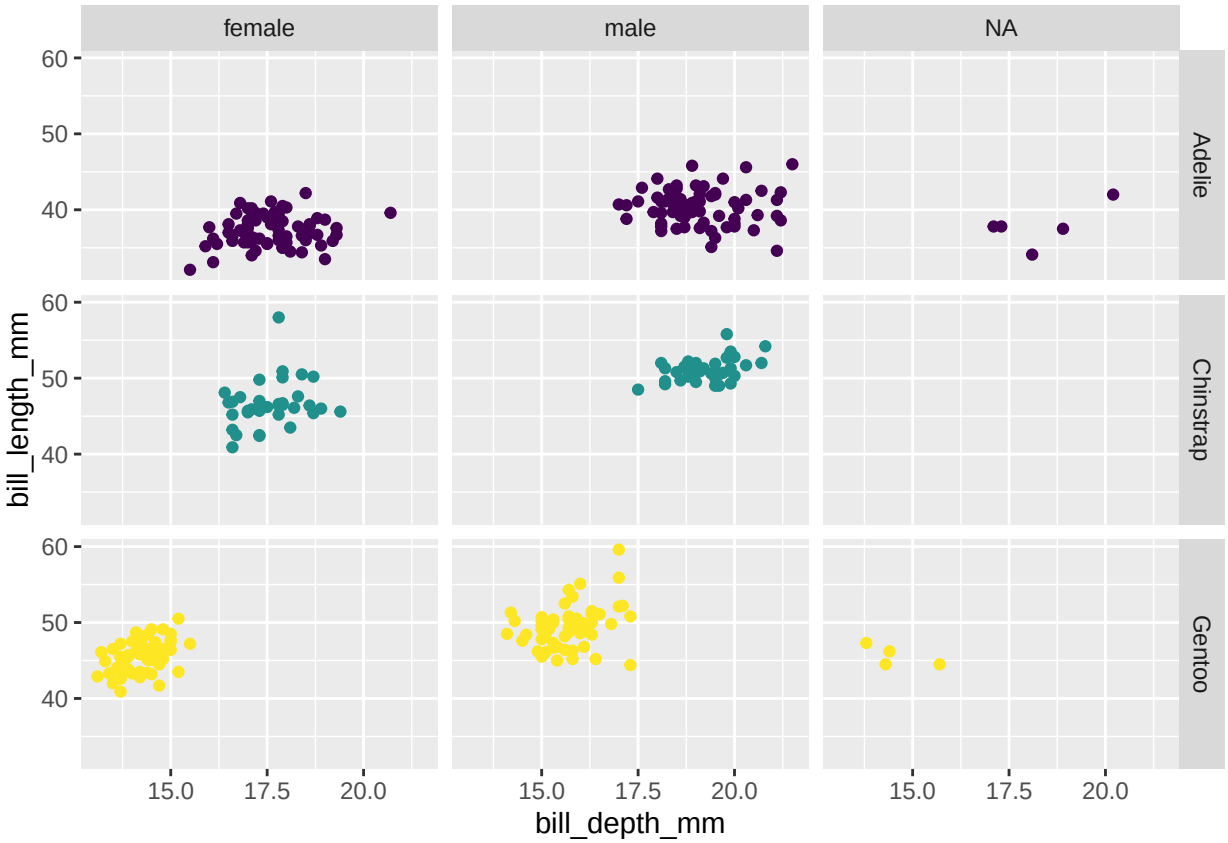
```
# Facet & Colour
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm,
      colour = species) +
  geom_point() +
  facet_grid(species ~ sex) +
  scale_colour_viridis_d()
```

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




```
# Facet & Colour, no legend
ggplot(penguins) +
  aes(x = bill_depth_mm,
      y = bill_length_mm,
      colour = species) +
  geom_point() +
  facet_grid(species ~ sex) +
  scale_color_viridis_d() +
  guides(color = "none")
```

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



B. Visualising numeric variables

7. The Lending Club (Slides #39-40)

```
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~
```

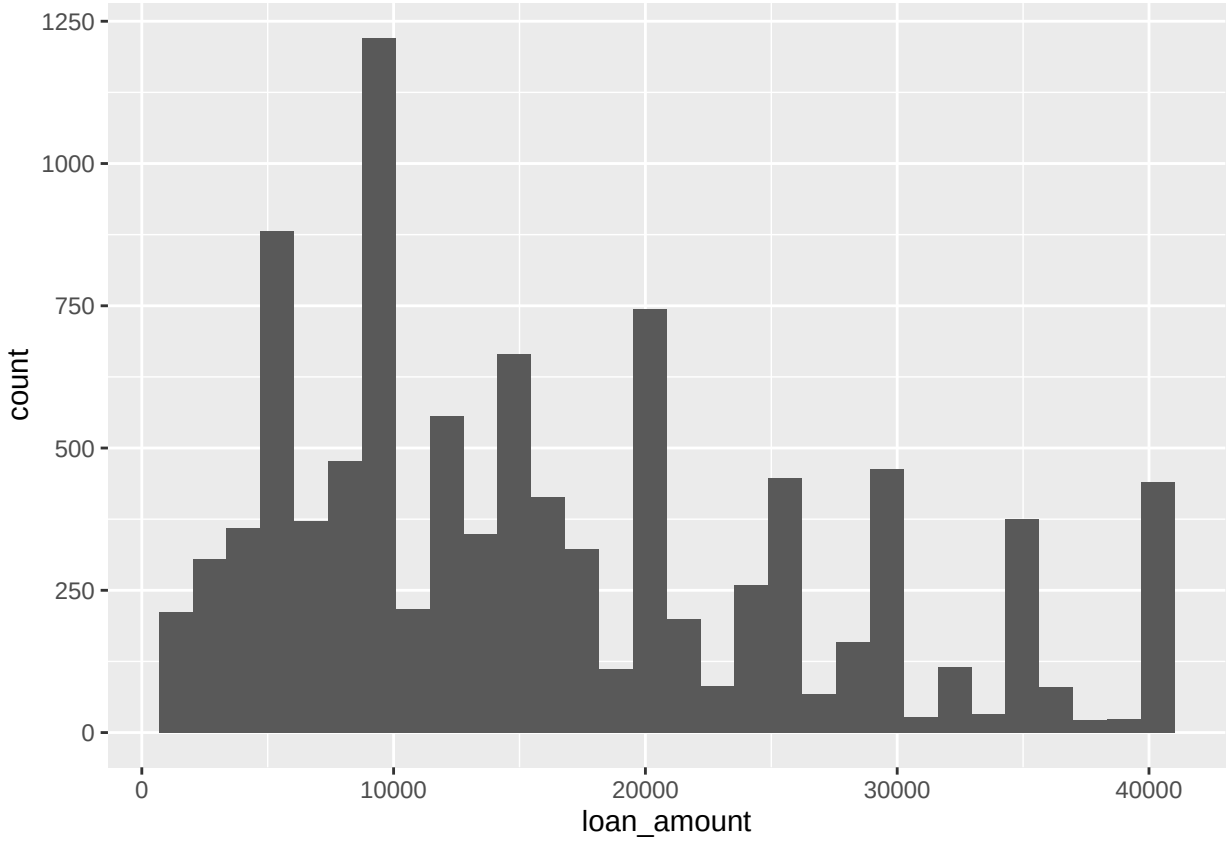
```
loans <- loans_full_schema %>%
  select(loan_amount, interest_rate, term, grade,
         state, annual_income, homeownership, debt_to_income)
glimpse(loans)
```

```
## 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~
```

8a. Histogram (Slides #46)

```
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram() # to know how many times a certain value appears in a variable
```

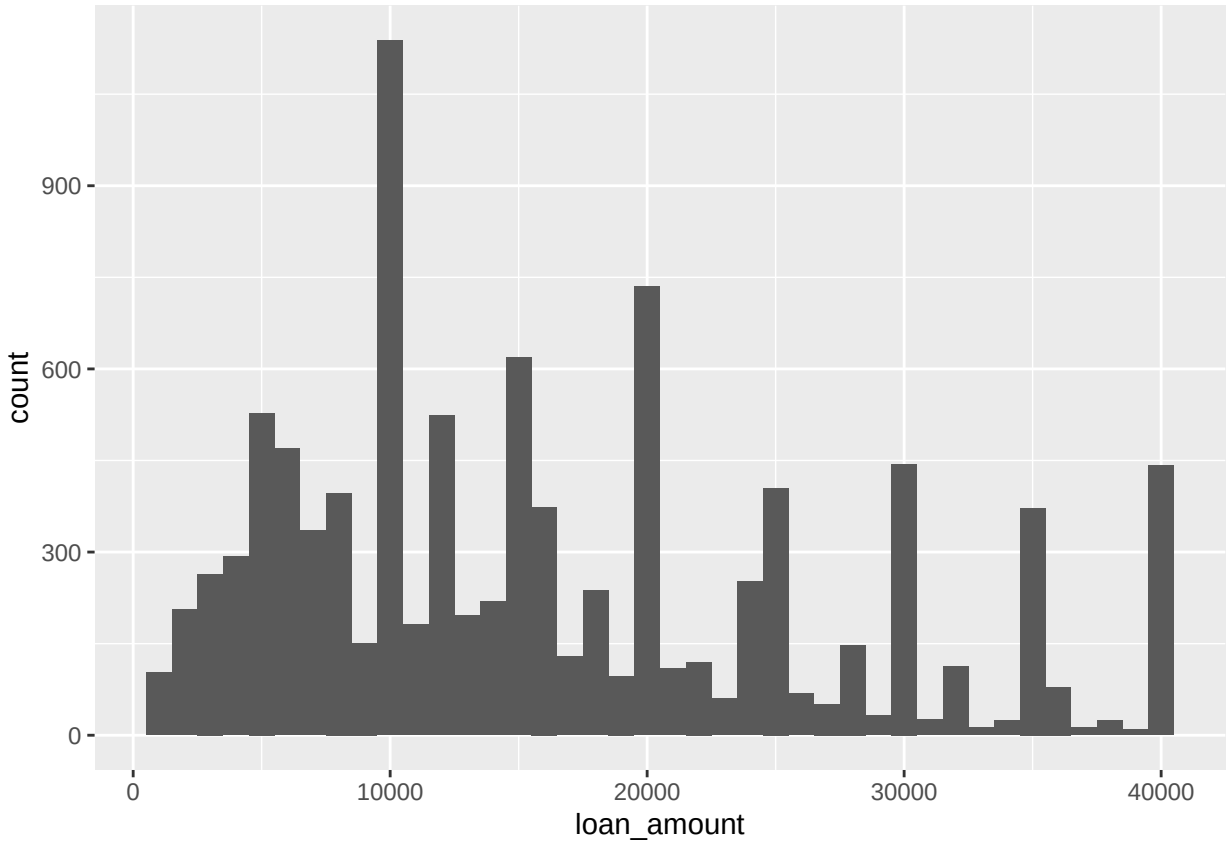
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



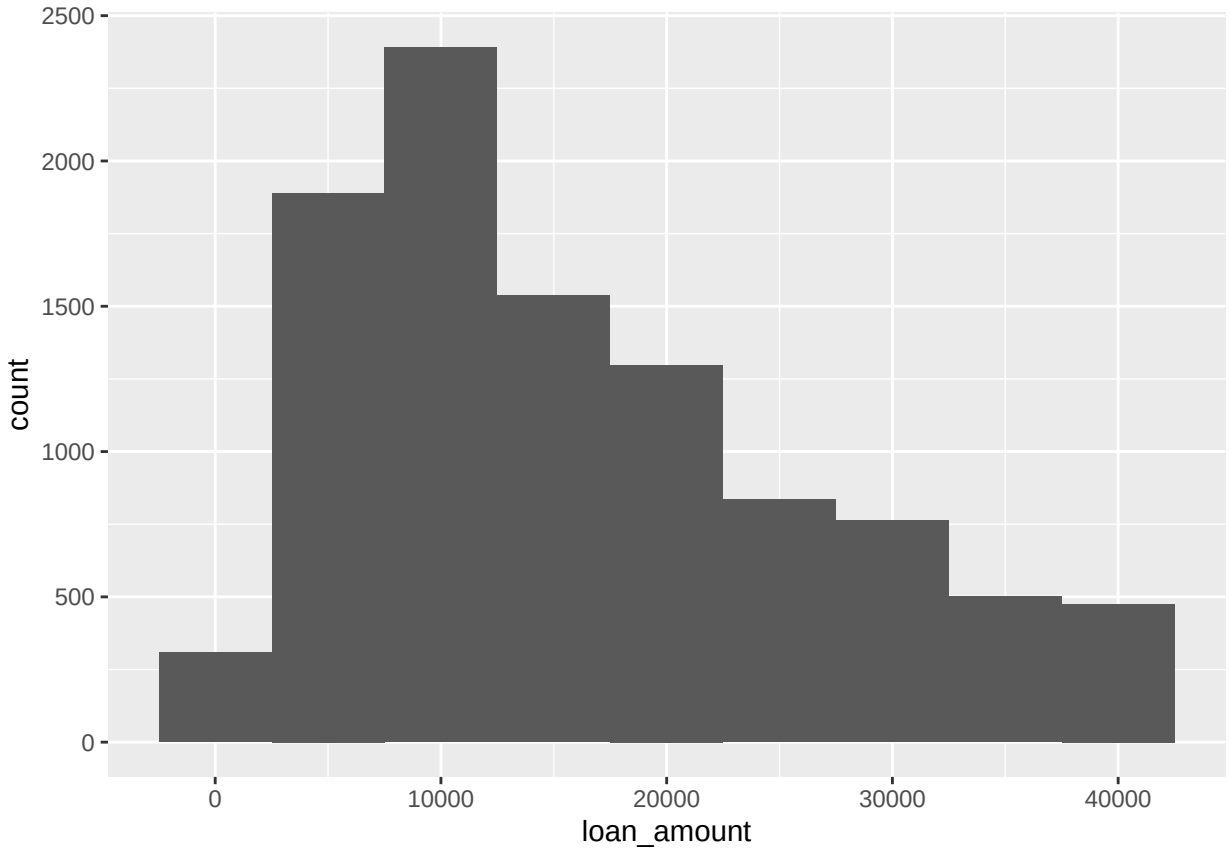

```
# or in a distribution (frequency of value)
```

8b. Histogram - varying binwidths (Slides #47-49)

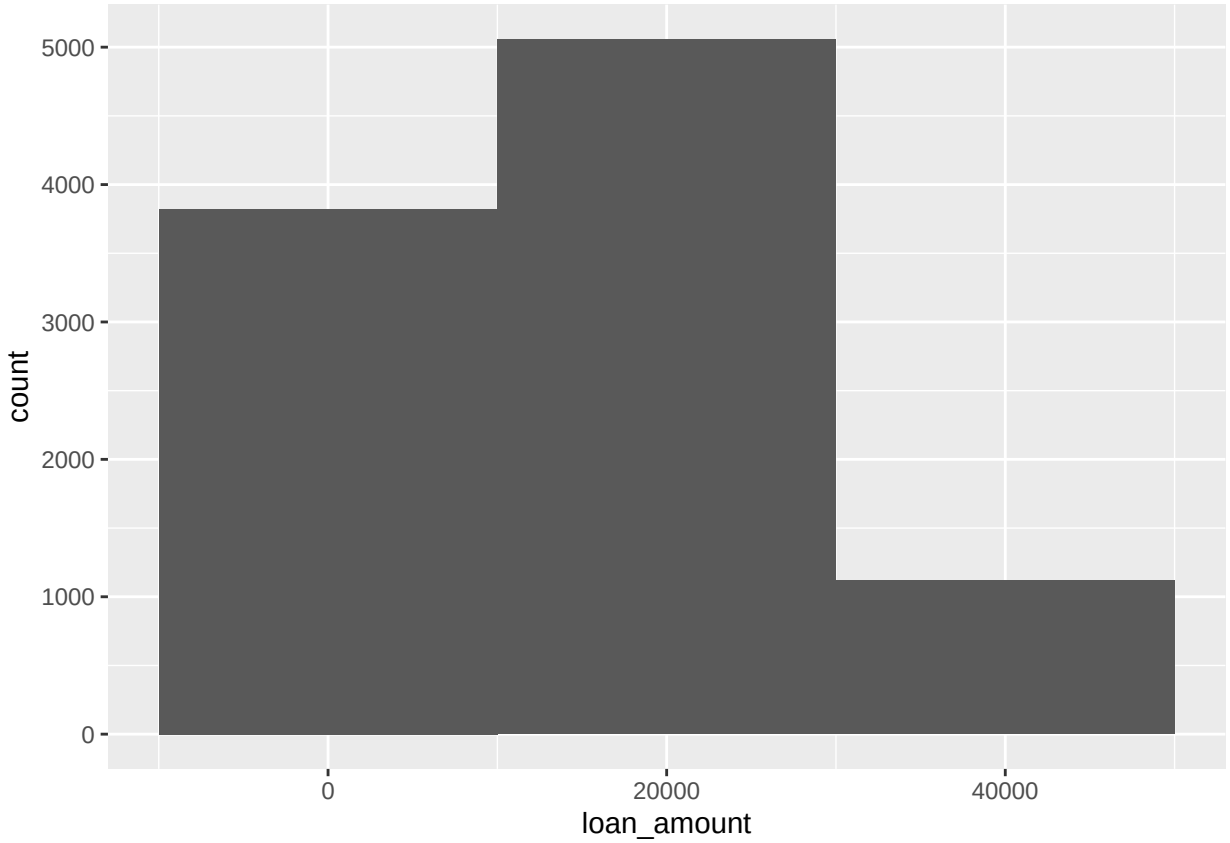
```
# binwidth = 1000
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram(binwidth = 1000)
```




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

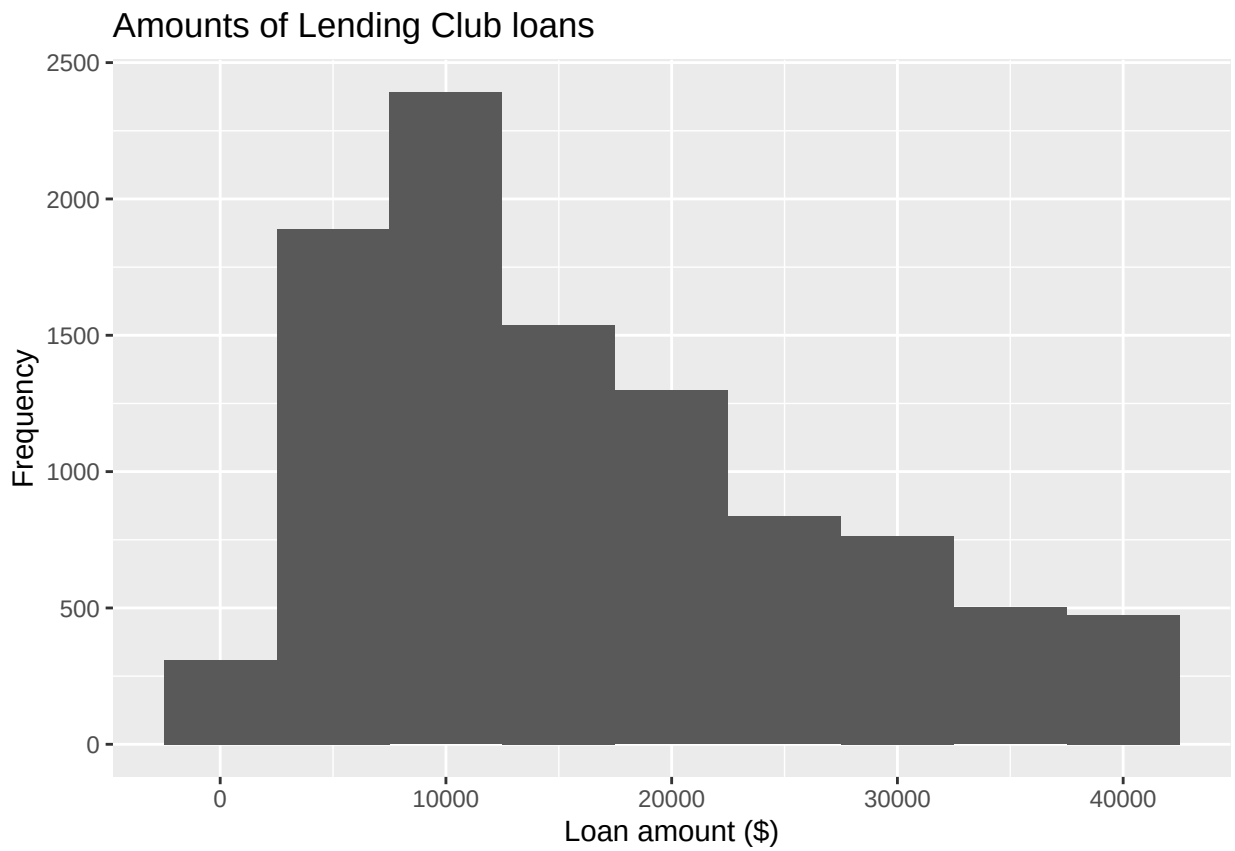



```
# binwidth = 20000
ggplot(loans) + aes(x = loan_amount) +
  geom_histogram(binwidth = 20000)
```

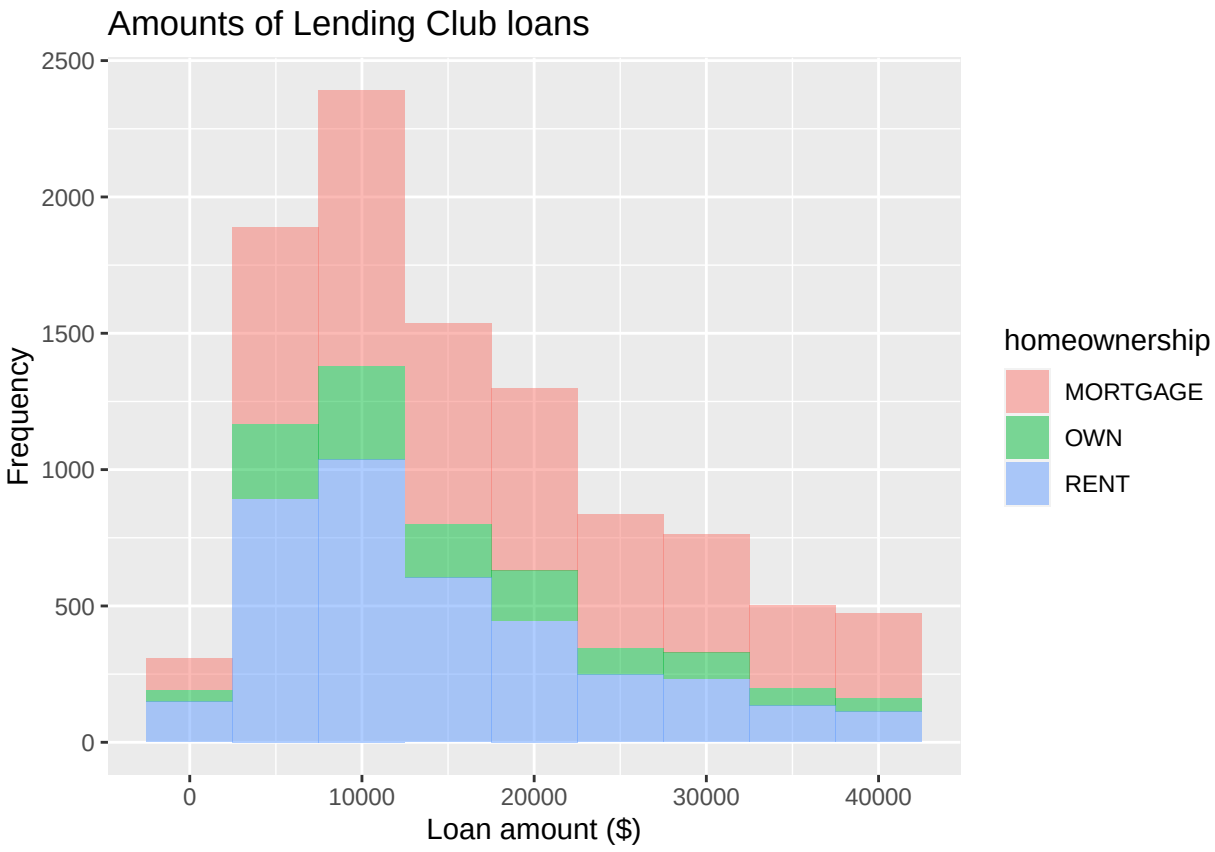


8c. Histogram - customisations (Slides #50-52)

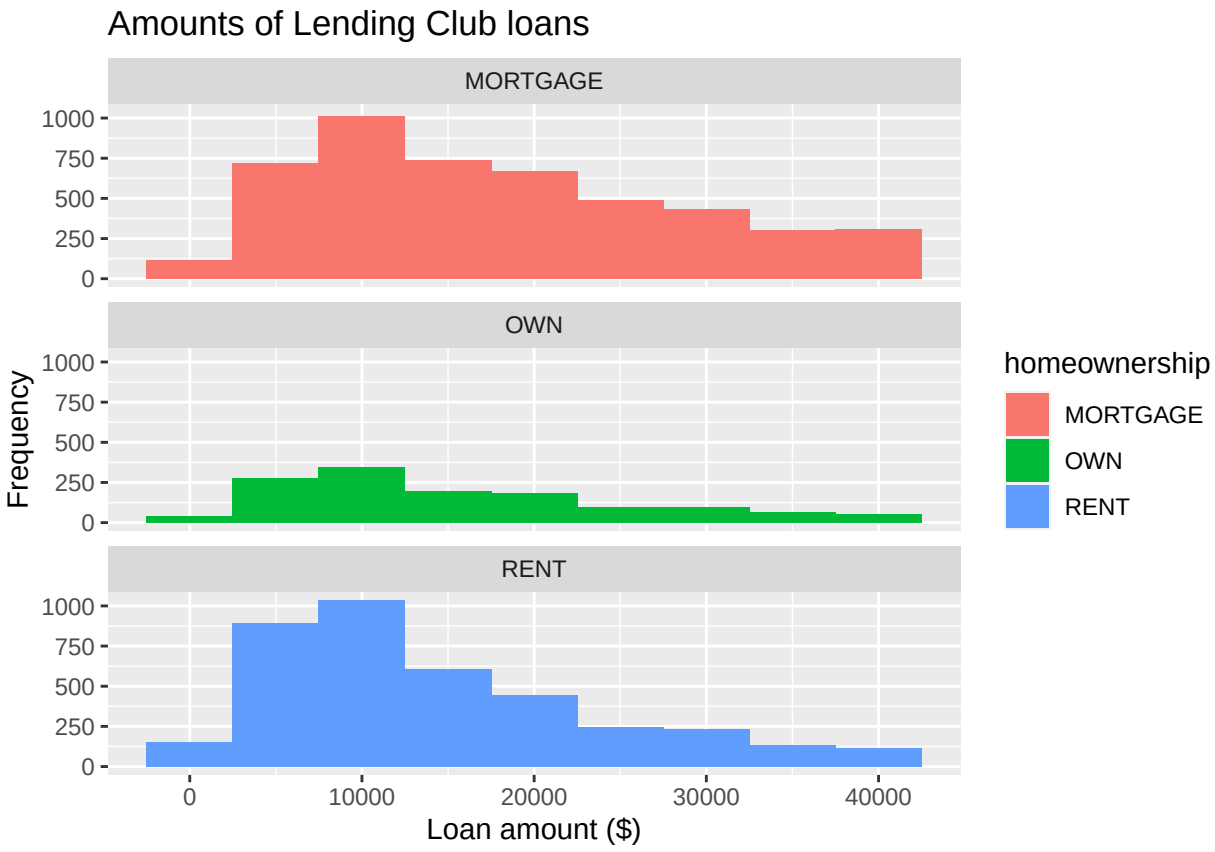
```
ggplot(loans) +
  aes(x = loan_amount) +
    geom_histogram(binwidth = 5000) +
      labs(x = "Loan amount ($)",
           y = "Frequency",
           title = "Amounts of Lending Club loans")
```




```
# Fill with categorical variable
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")
```

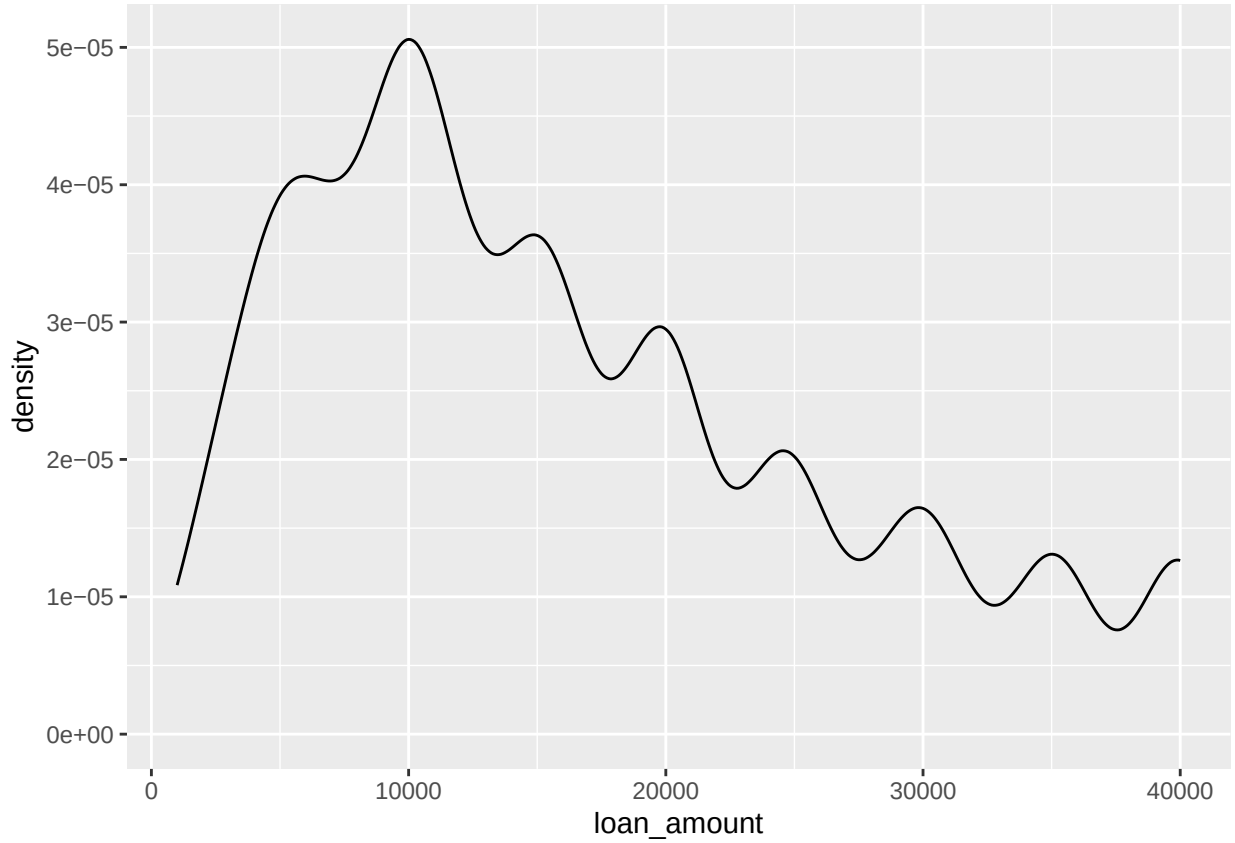



```
# Facet with categorical variable
ggplot(loans) +
  aes(x = loan_amount,
      fill = homeownership) +
    geom_histogram(binwidth = 5000) +
      labs(x = "Loan amount ($)",
           y = "Frequency",
           title = "Amounts of Lending Club loans") +
        facet_wrap(~ homeownership, nrow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



9a. Density plot (Slides #53)

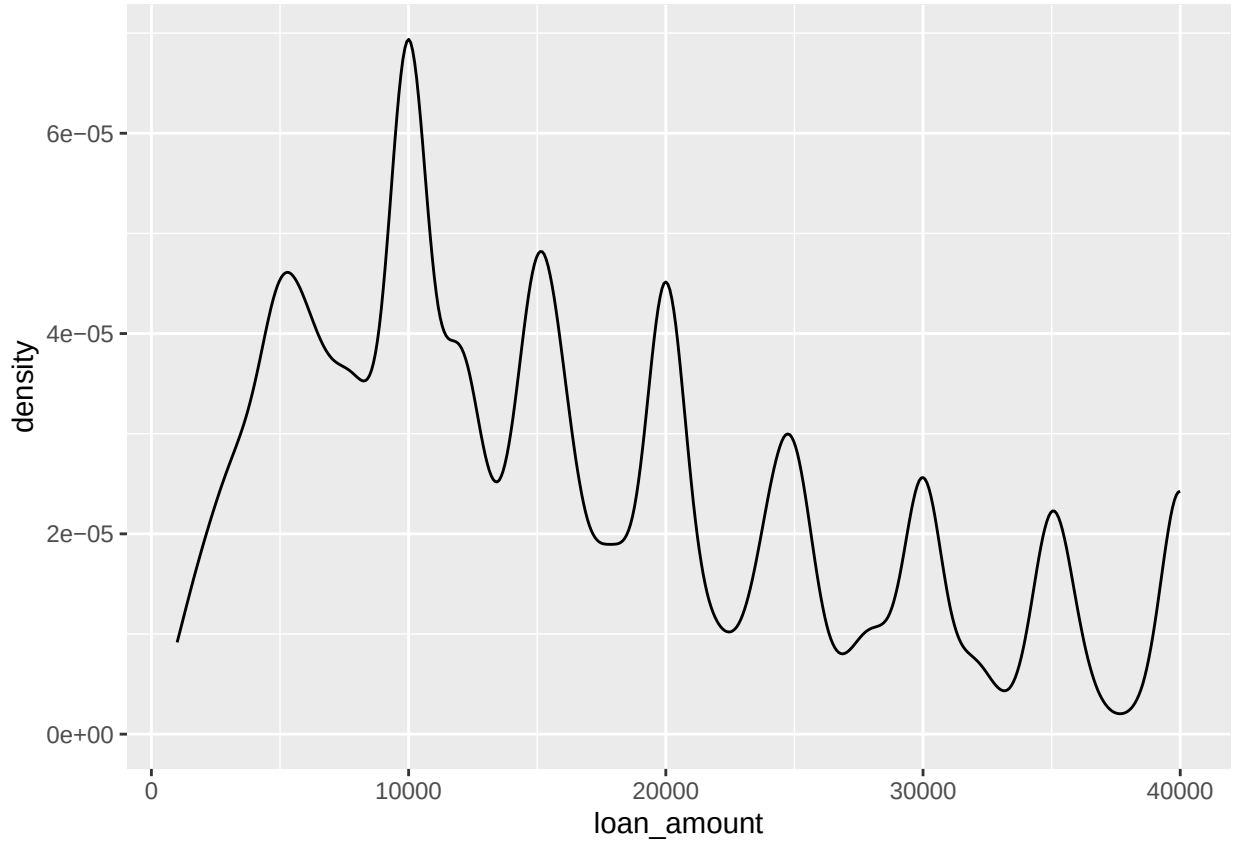
```
ggplot(loans) + aes(x = loan_amount) +
  geom_density() # just like histograms but represented by smooth curves instead;
```



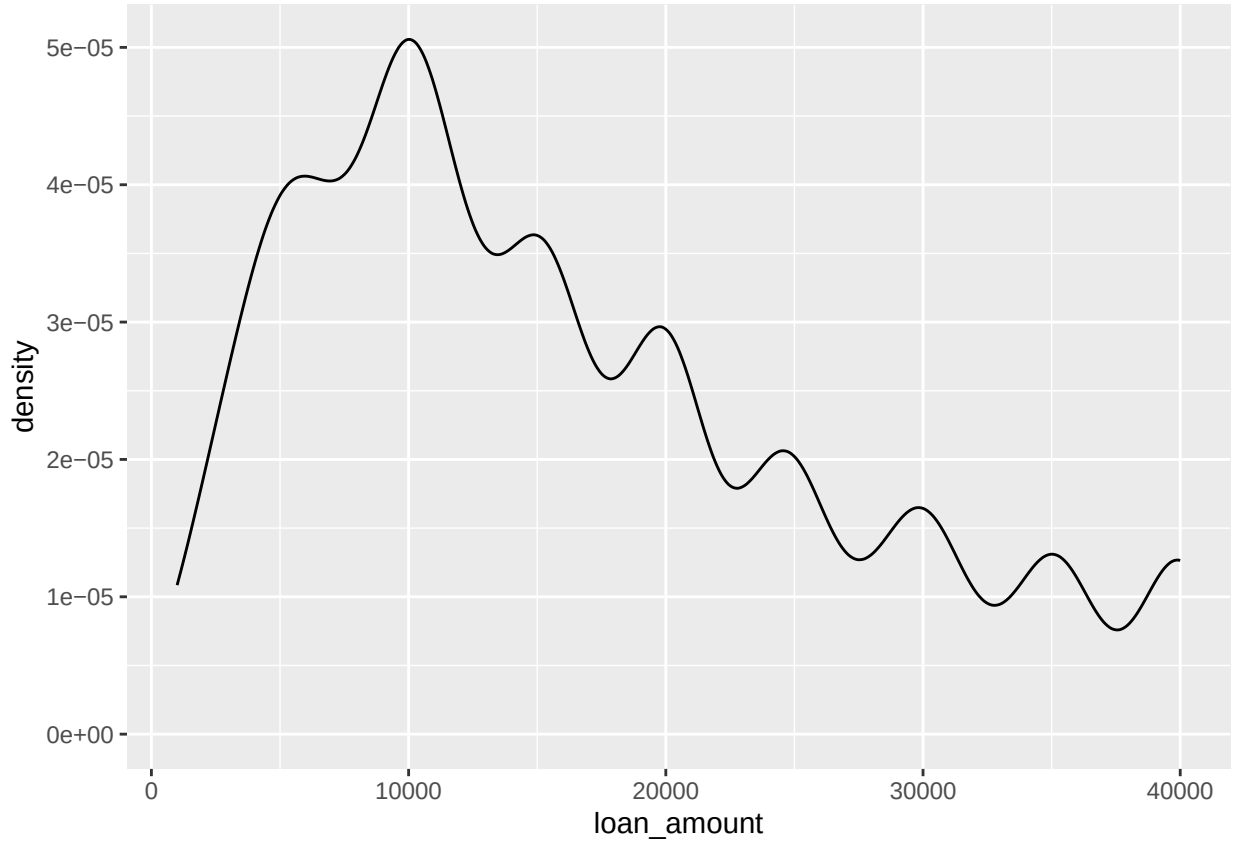

```
                # gives probability density of data in each range instead of count
                # (no. of times values in certain range occur over
                # total no. of values in the variable)
```

9b. Density plot - varying bandwidths (Slides #54-56)

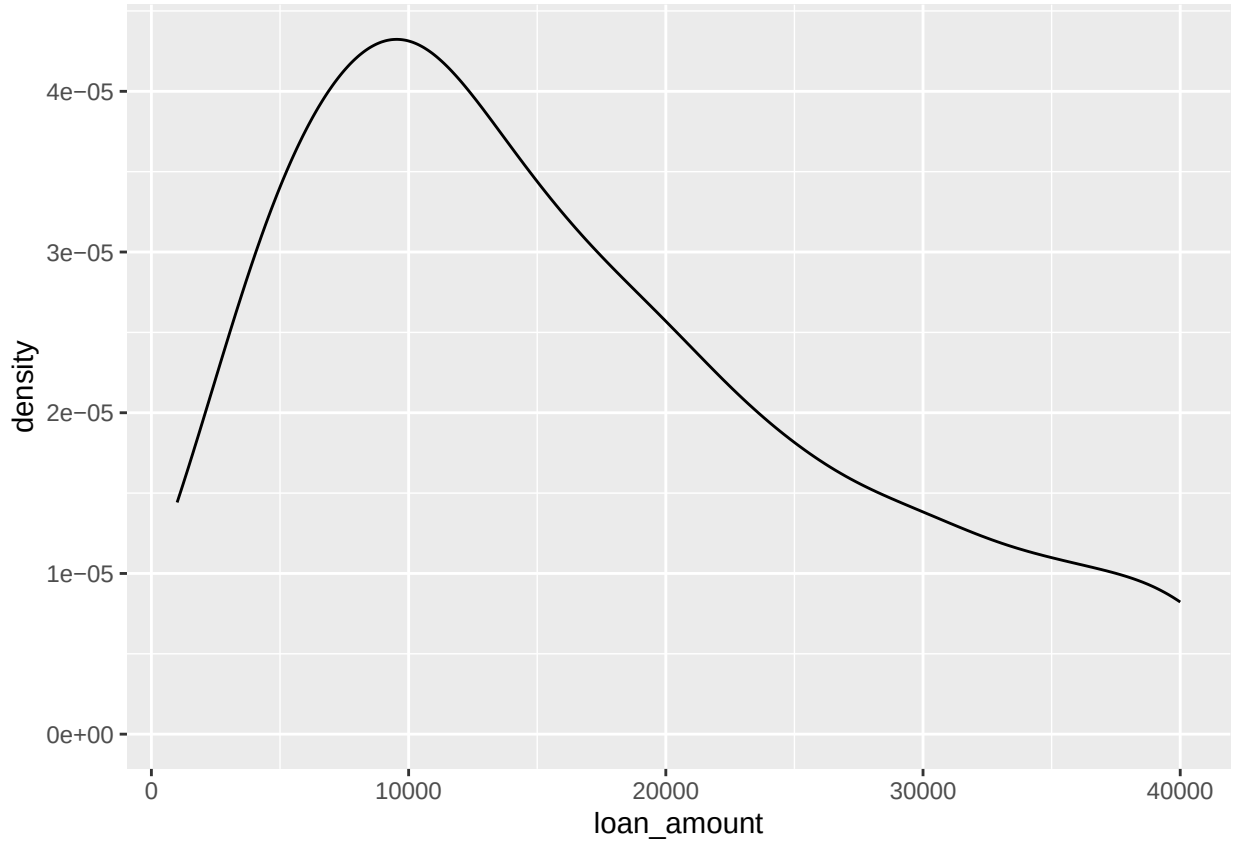
```
ggplot(loans) + aes(x = loan_amount) +
  geom_density(adjust = 0.5)
```




```
ggplot(loans) + aes(x = loan_amount) +
  geom_density(adjust = 1) # default bandwidth
```

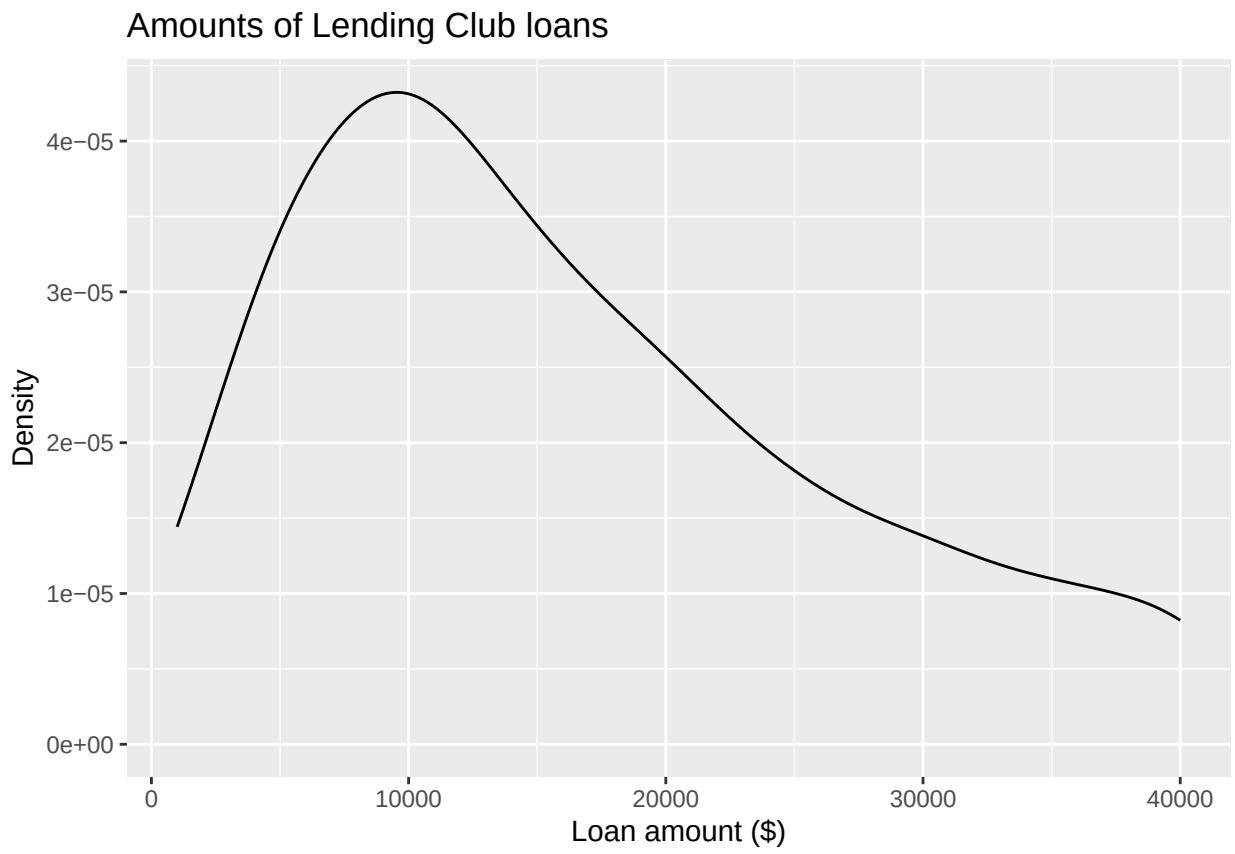



```
ggplot(loans) + aes(x = loan_amount) +
  geom_density(adjust = 2)
```

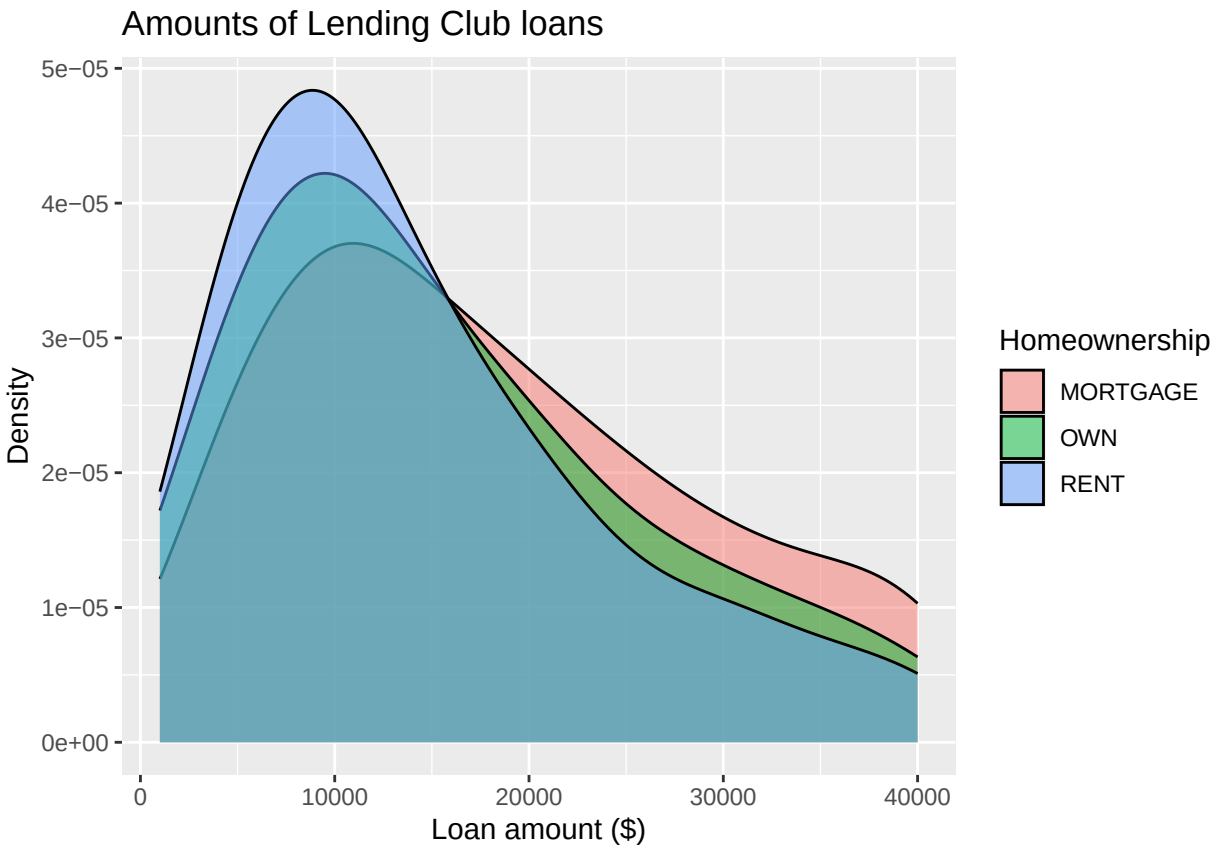


9c. Density plot - customisations (Slides #57-58)

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

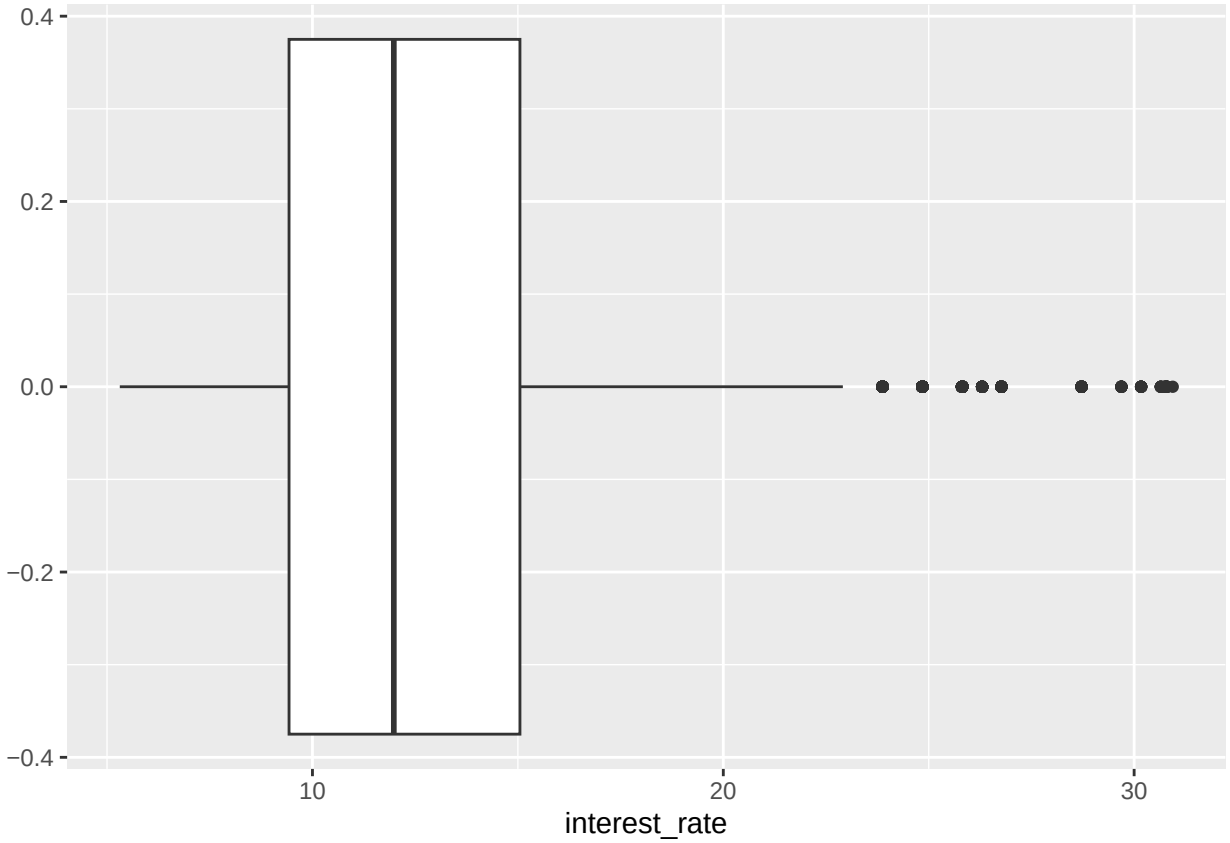



```
# Add categorical variable
ggplot(loans) + aes(x = loan_amount,
                    fill = homeownership) +
  geom_density(adjust = 2,
               alpha = 0.5) +
      labs(x = "Loan amount ($)",
           y = "Density",
           title = "Amounts of Lending Club loans",
           fill = "Homeownership")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



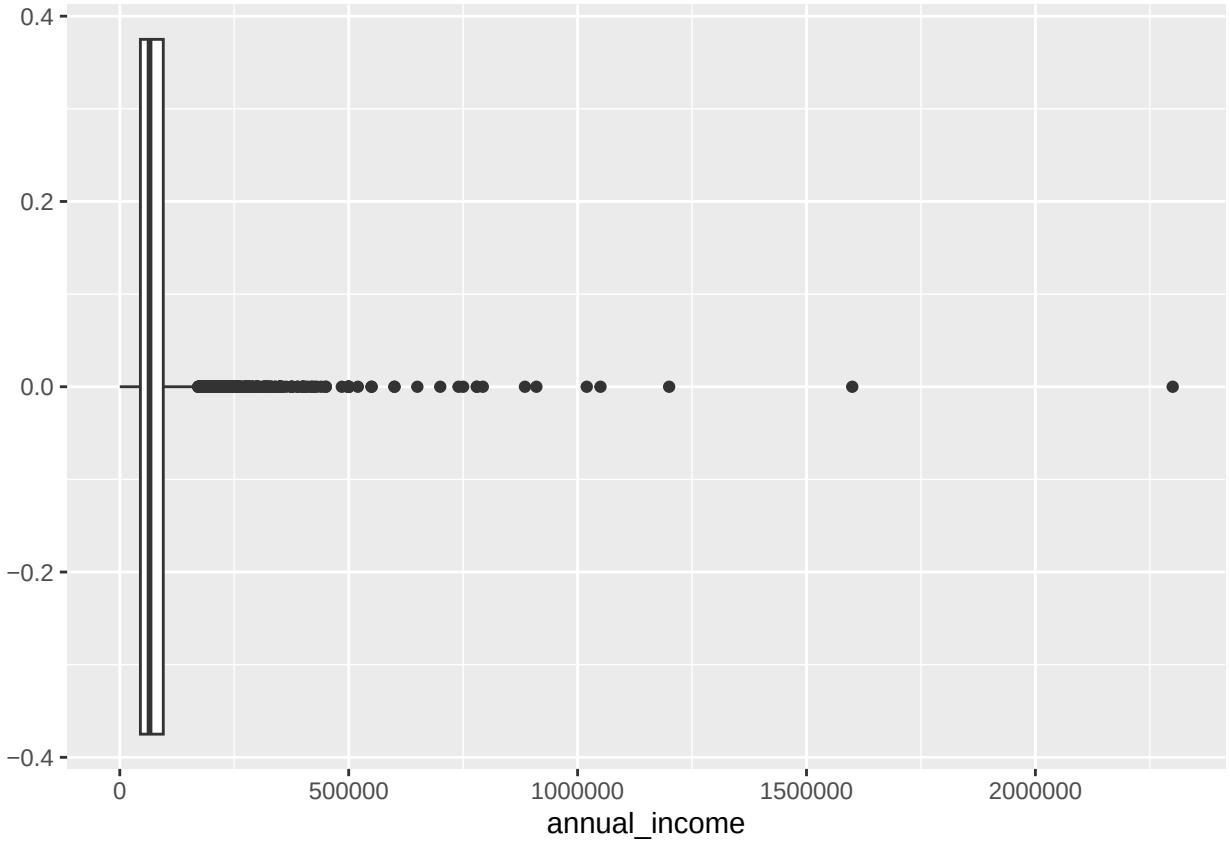
10a. Box plot (Slides #59-60)

```
ggplot(loans) + aes(x = interest_rate) +
  geom_boxplot() # vertical line in the middle = median, boundaries = IQR,
```



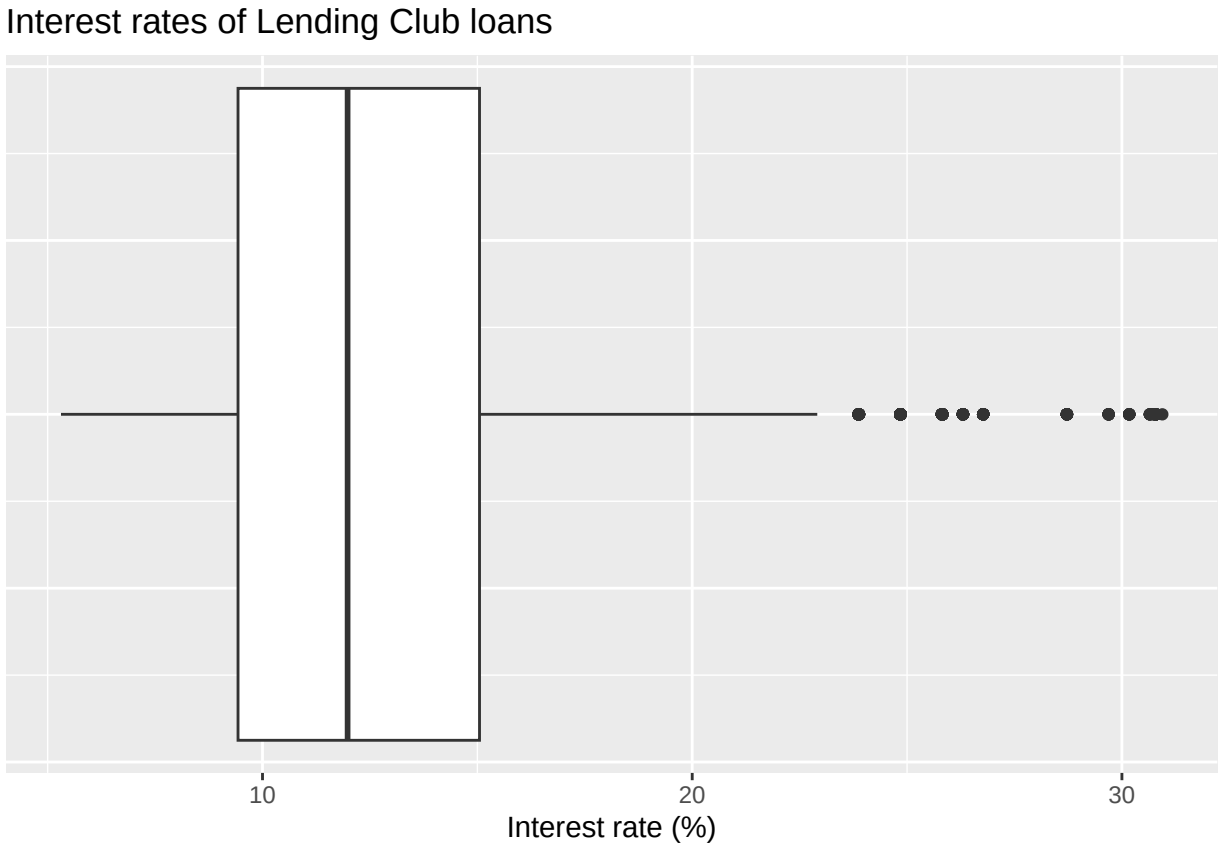

```
                # dots outside = outliers, tips of horizontal line = min & max
```

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

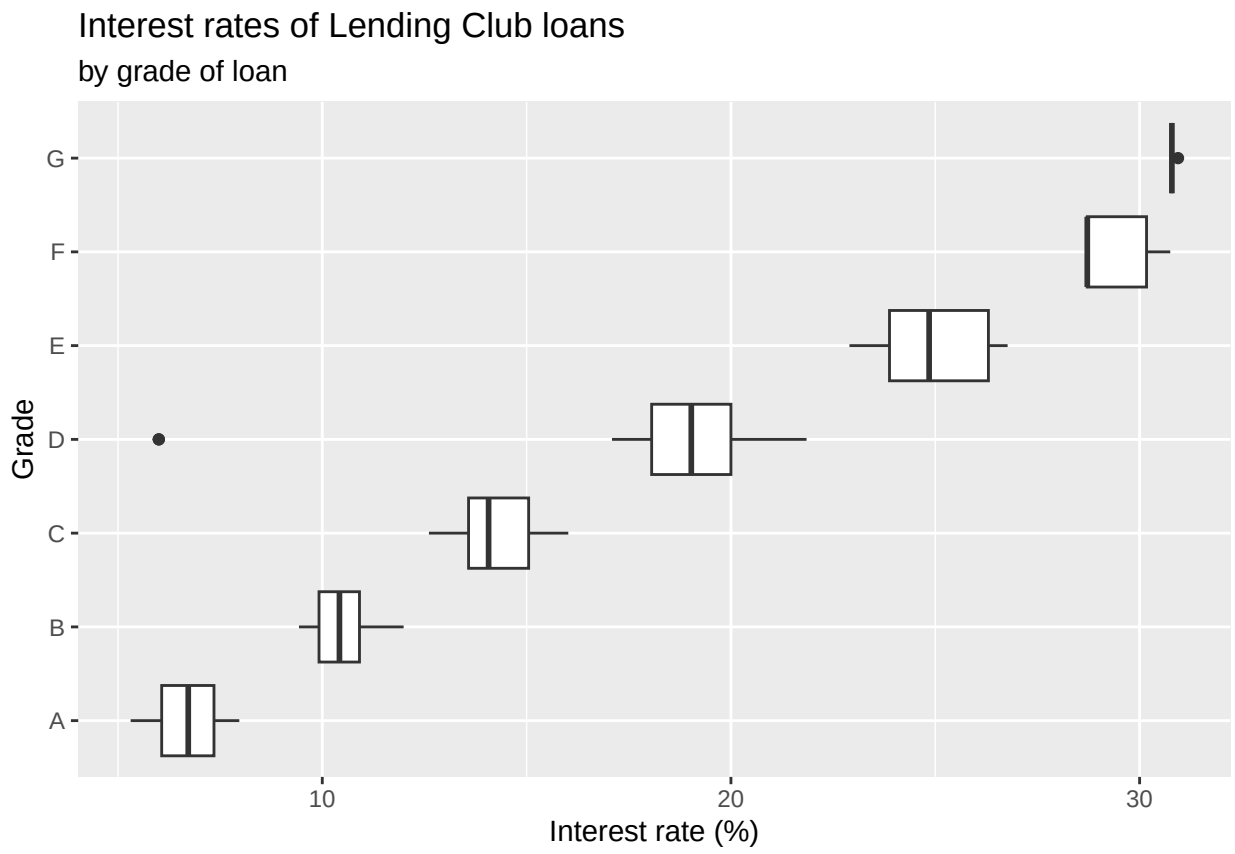


10b. Box plot - customisations (Slides #61-62)

```
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())
```



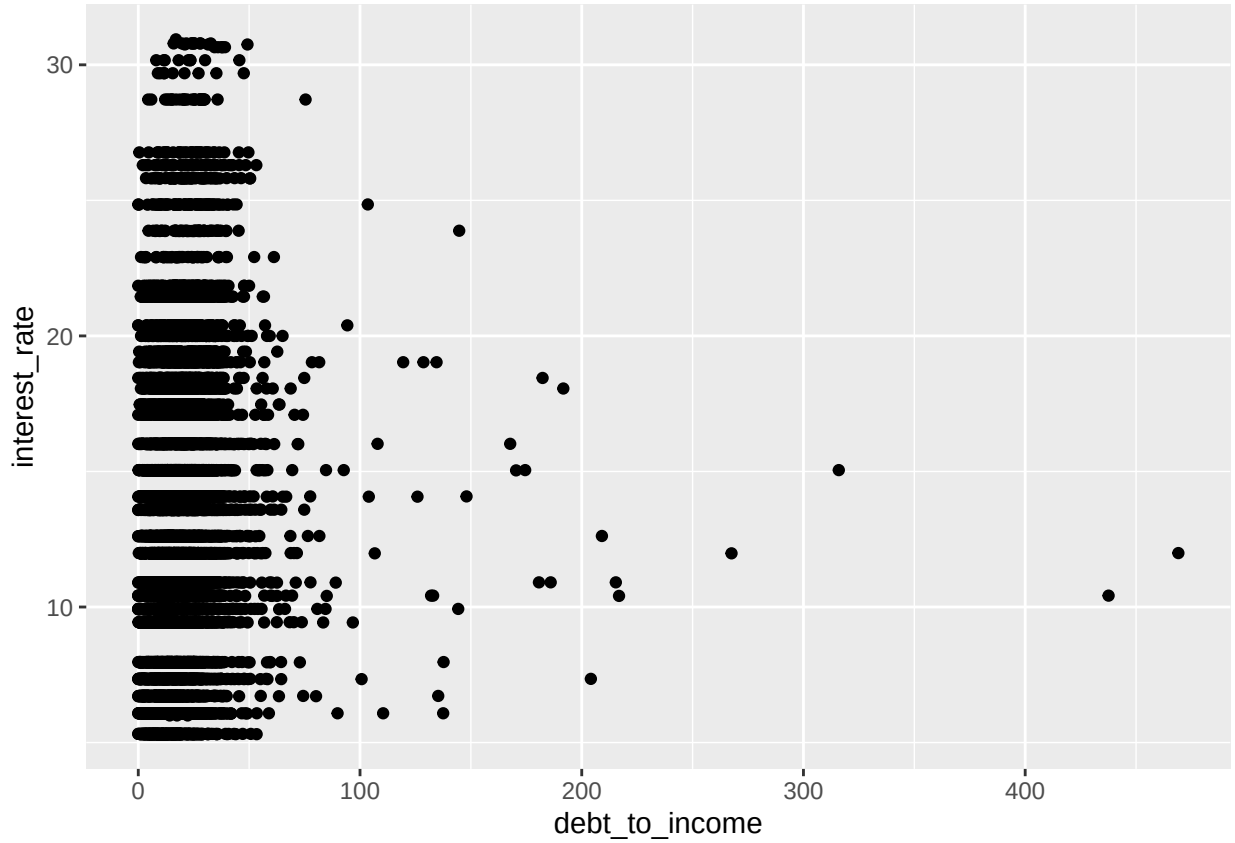

```
# adding categoric variable
ggplot(loans) + aes(x = interest_rate,
                    y = grade) +
  geom_boxplot() +
  labs(x = "Interest rate (%)",
       y = "Grade",
       title = "Interest rates of Lending Club loans",
       subtitle = "by grade of loan")
```



11. Scatterplot (Slides #63)

```
ggplot(loans) + aes(x = debt_to_income,
                    y = interest_rate) +
  geom_point() # but too concentrated on the left
```

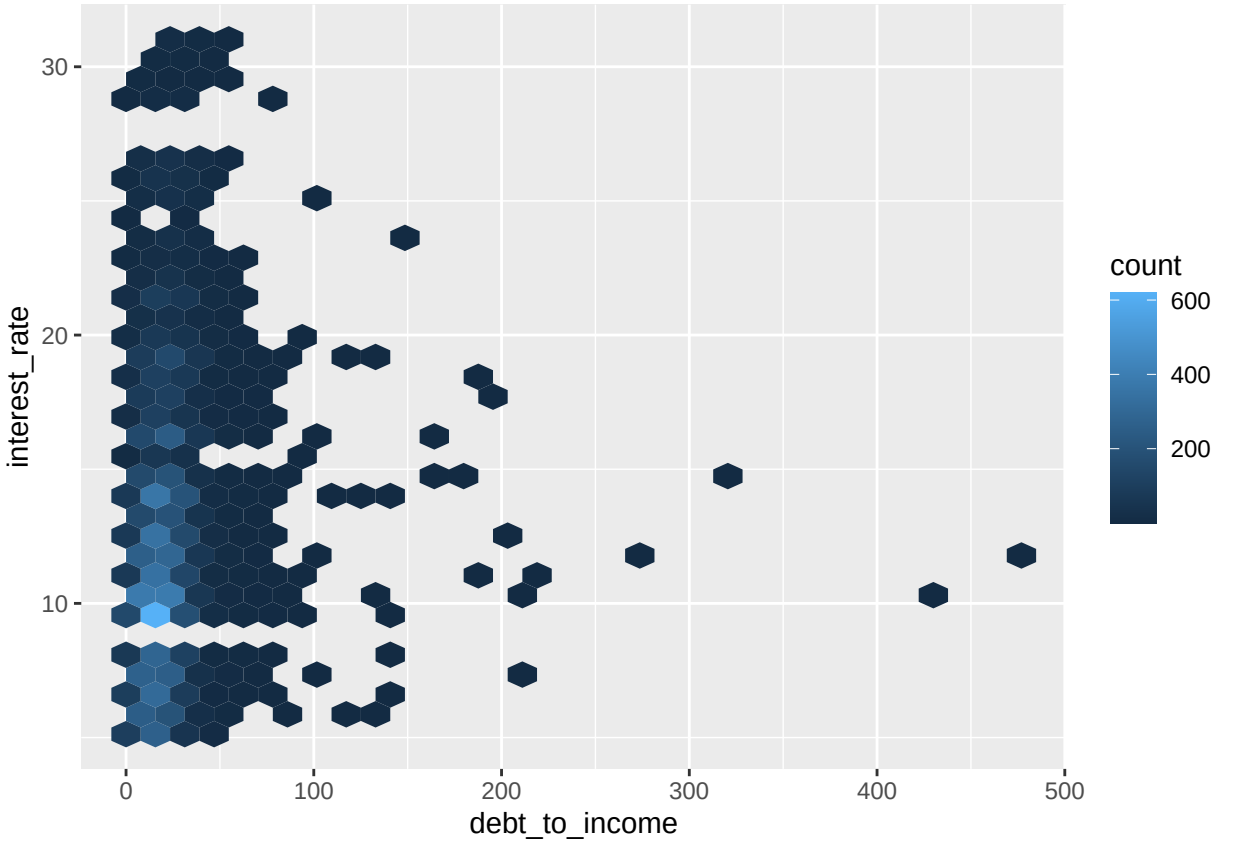
```
## Warning: Removed 24 rows containing missing values (‘geom_point()‘).
```



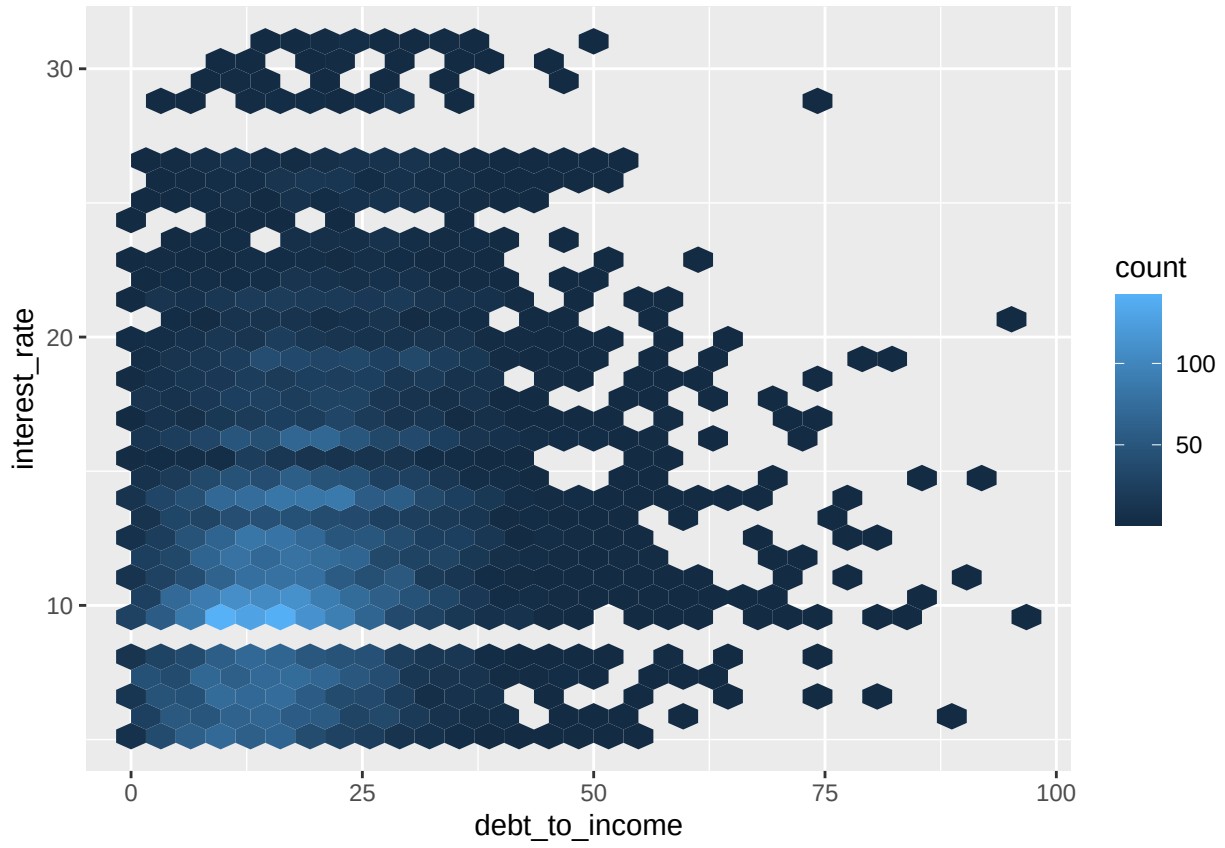
12. Hex plot (Slides #64-65)

```
ggplot(loans) + aes(x = debt_to_income,
                    y = interest_rate) +
  geom_hex() # similar to histogram;
```

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



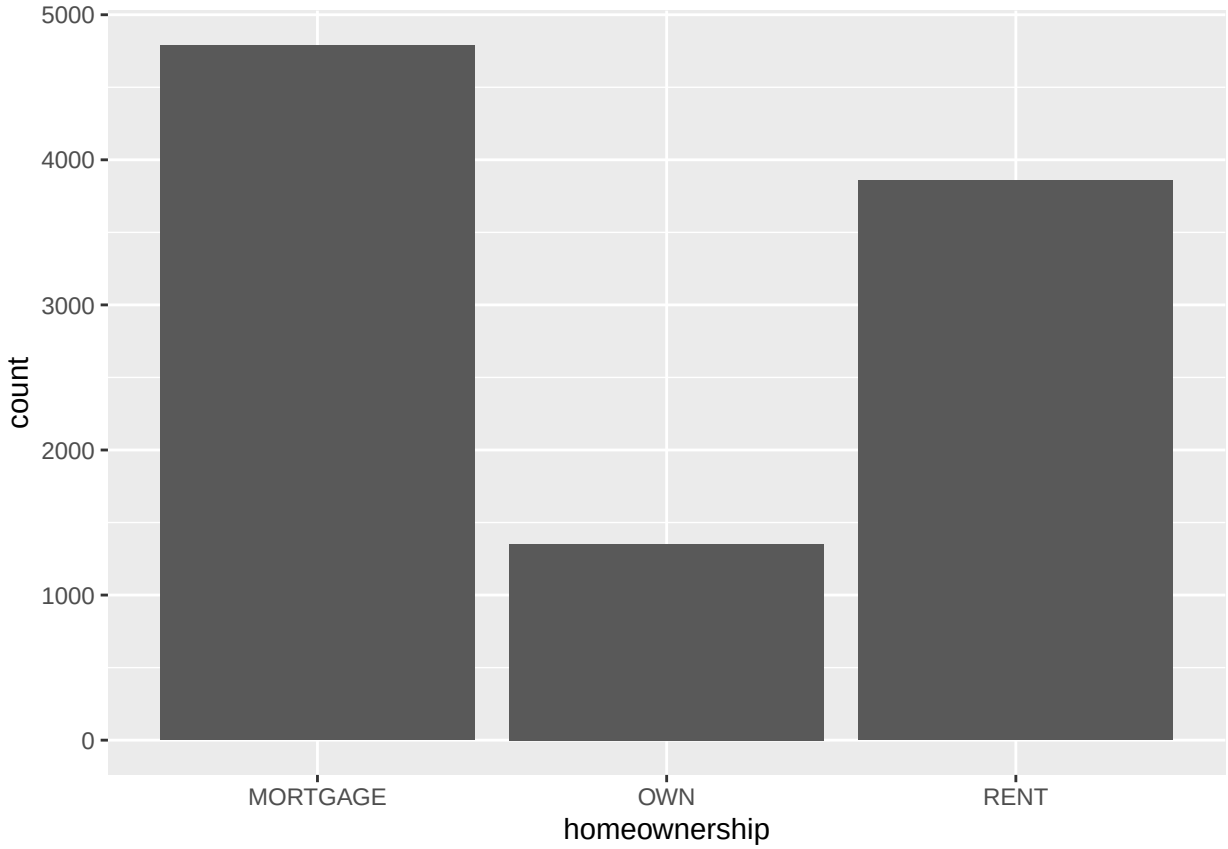

```
ggplot(loans %>% filter(debt_to_income < 100)) + aes(x = debt_to_income,
                    y = interest_rate) +
  geom_hex()
```



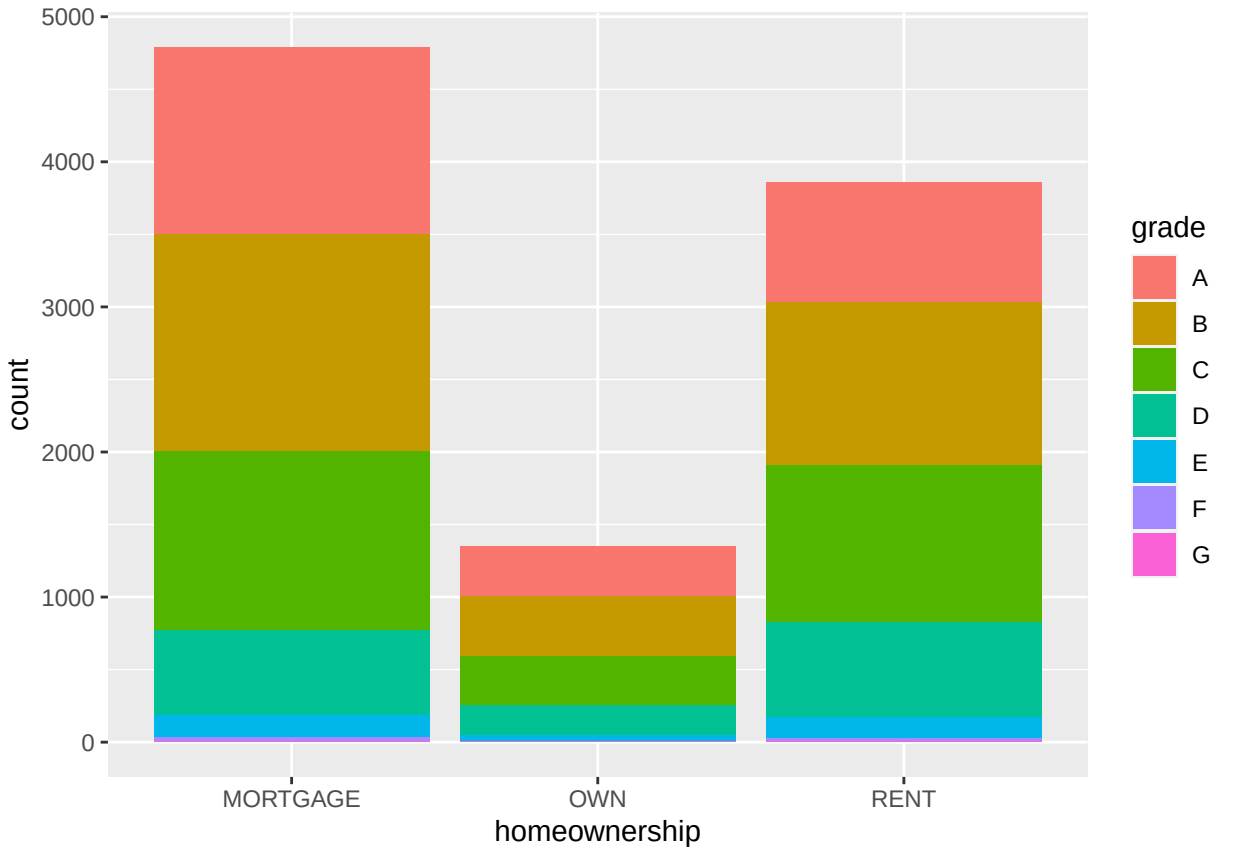
C. Visualising categoric variables

13. Bar plot (Slides #67-71)

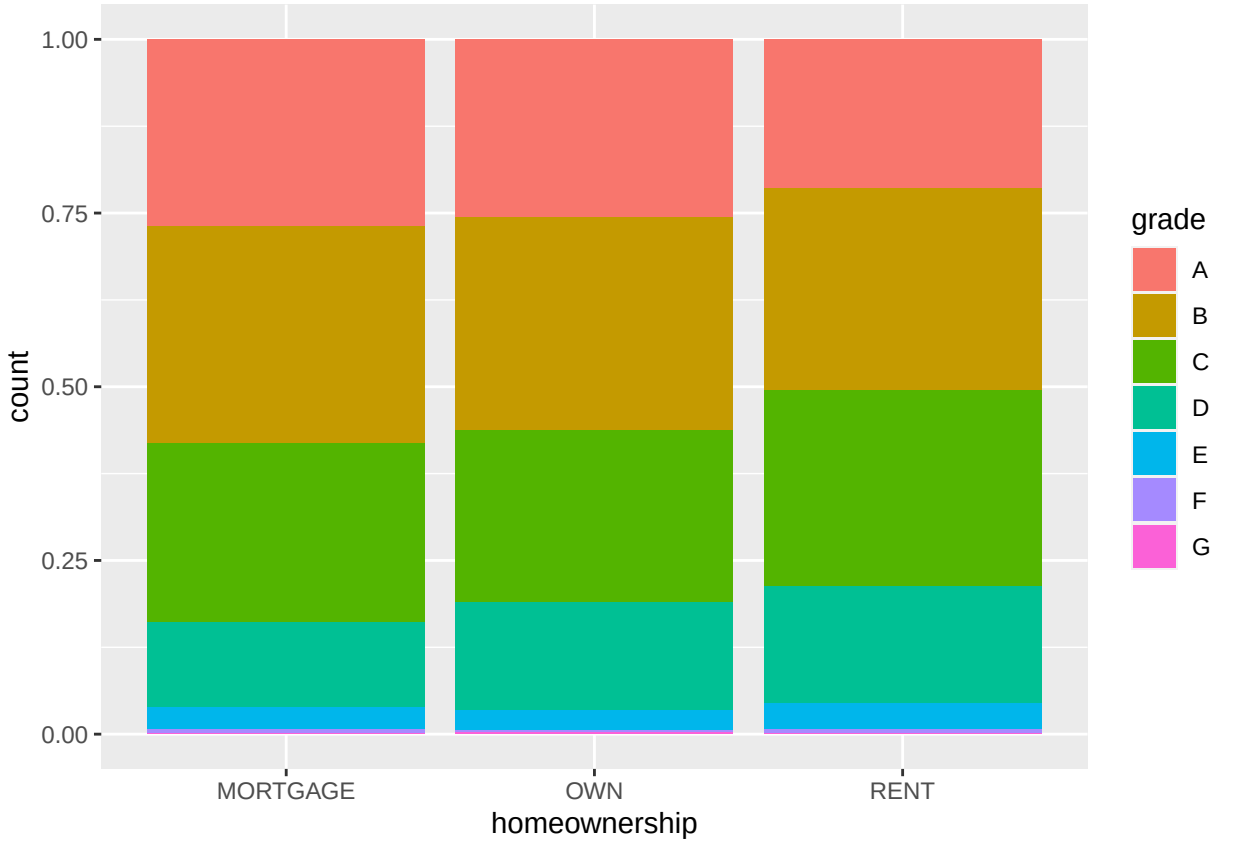
```
ggplot(loans) + aes(x = homeownership) +
  geom_bar()
```



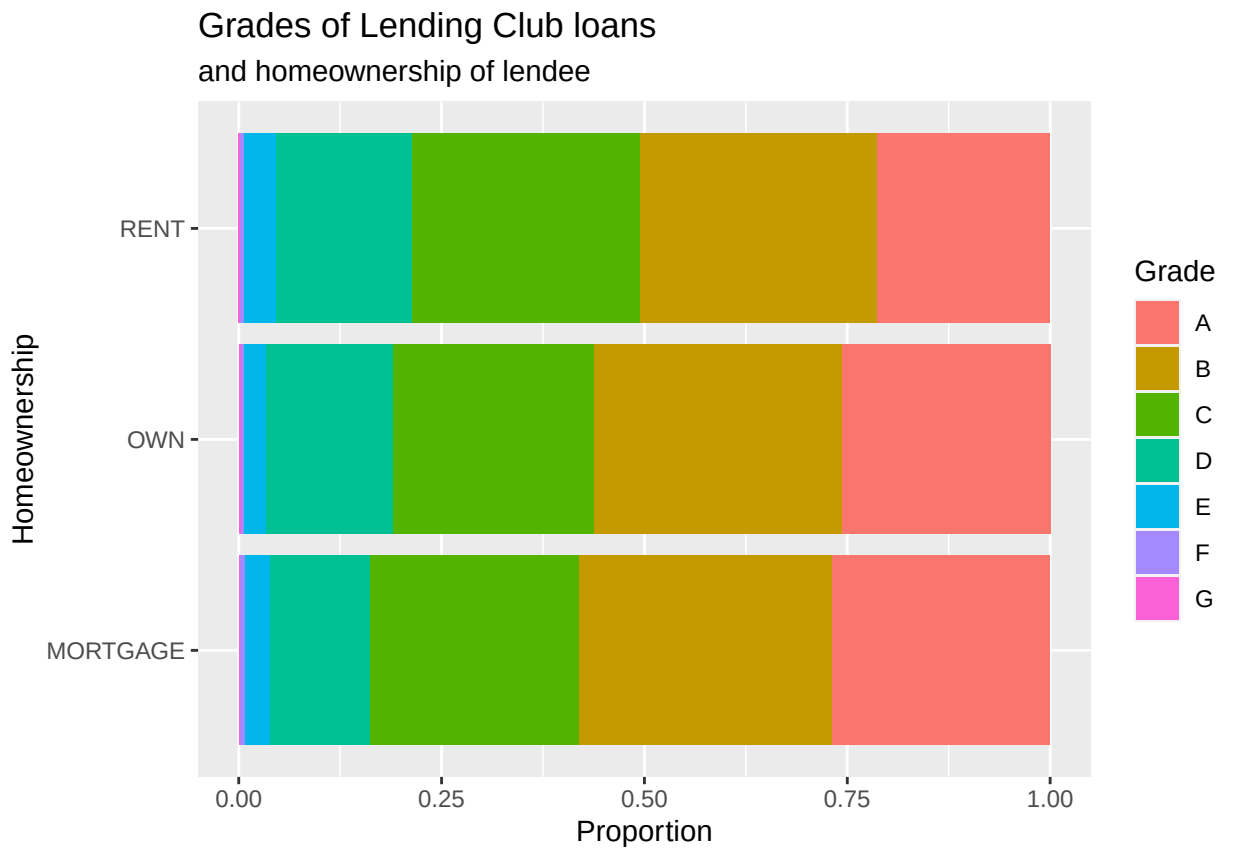

```
# segmented bar plot
ggplot(loans) + aes(x = homeownership,
                    fill = grade) +
  geom_bar()
```




```
# segmented bar plot
ggplot(loans) + aes(x = homeownership,
                    fill = grade) +
  geom_bar(position = "fill")
```



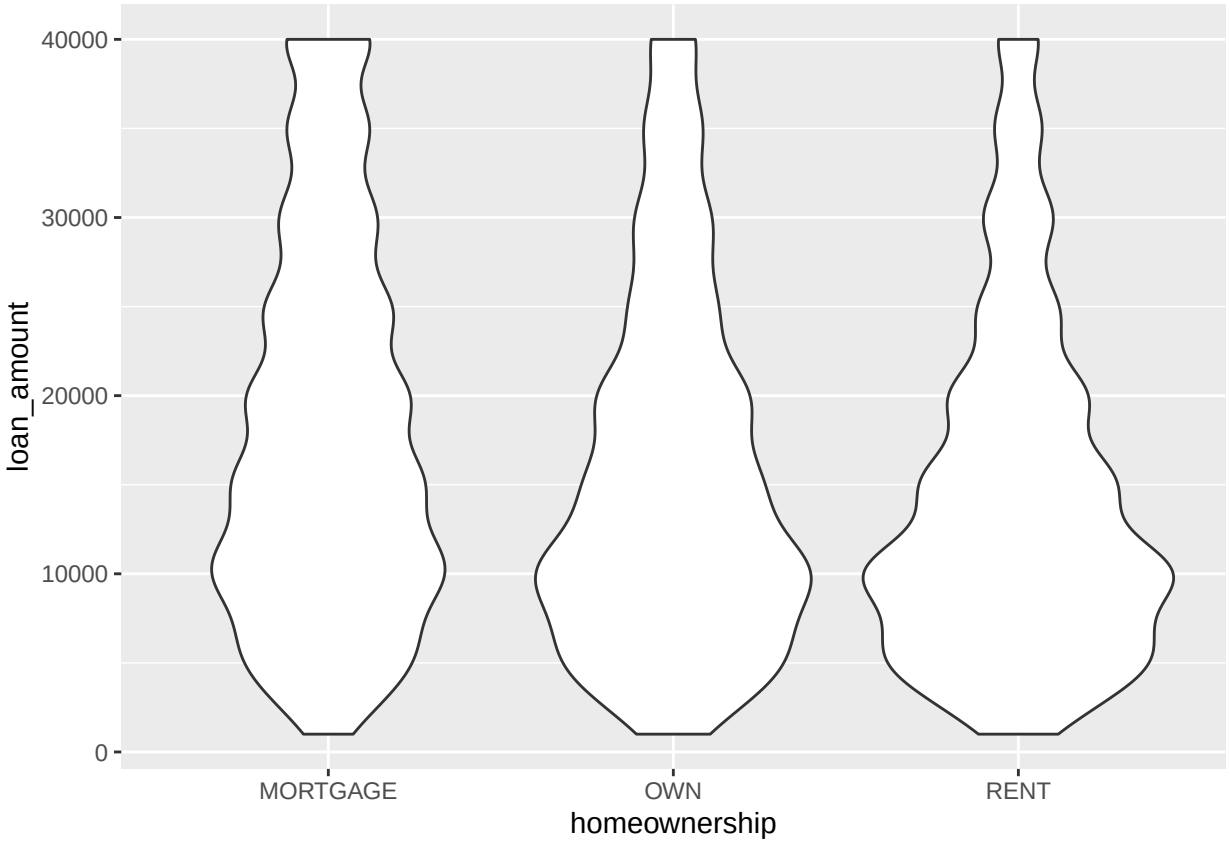

```
# customisation of bar plot
ggplot(loans) + aes(y = homeownership,
                    fill = grade) +
  geom_bar(position = "fill") +
  labs(x = "Proportion",
       y = "Homeownership",
       fill = "Grade",
       title = "Grades of Lending Club loans",
       subtitle = "and homeownership of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



D. Visualising variables of varied types

14. Violin plot (Slides #73)

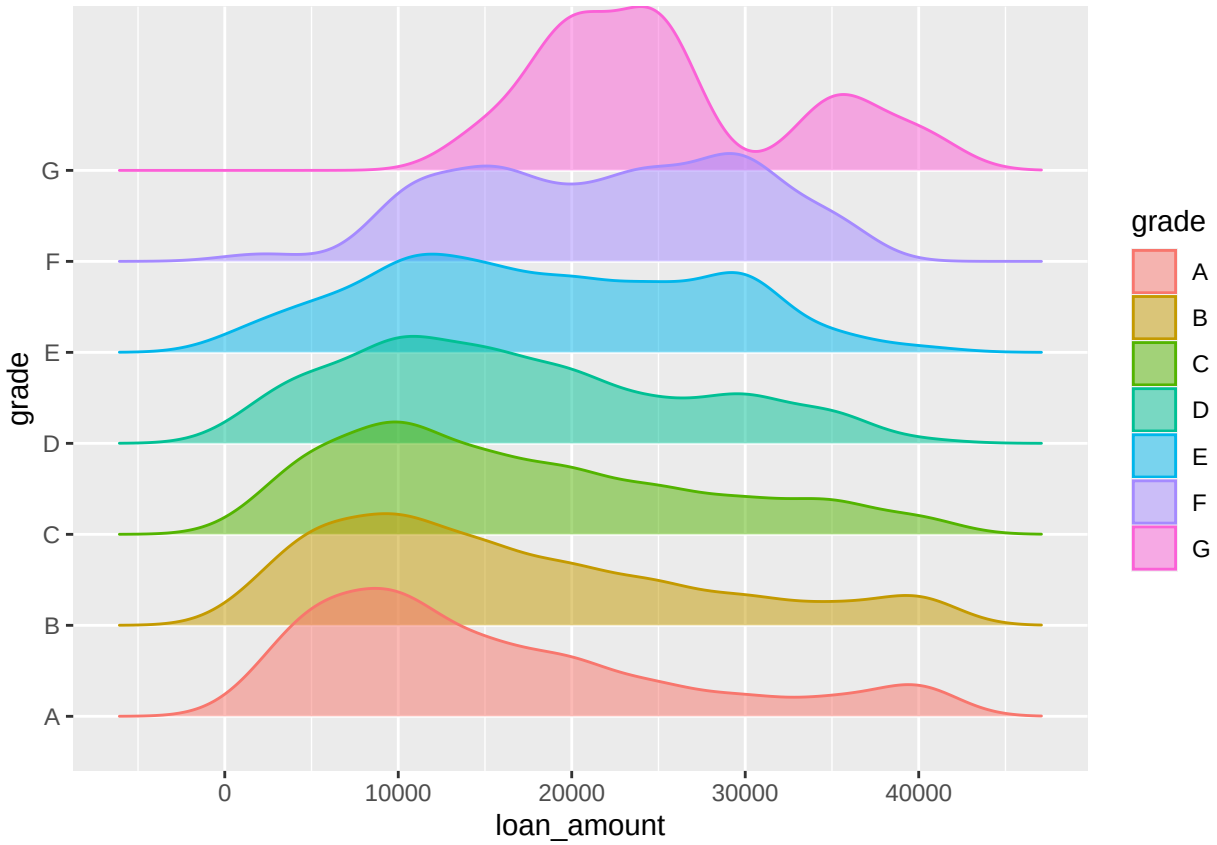
```
ggplot(loans) + aes(x = homeownership,
                    y = loan_amount) +
  geom_violin()
```



15. Ridge plot (Slides #74)

```
library(ggridges)
ggplot(loans) + aes(x = loan_amount,
                    y = grade,
                    fill = grade,
                    color = 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