

Challenge-7

Wu Xingyi (Kyra)

2023-10-02

Questions

Question-1: Uploading the Palmer Penguin Data Solutions:

```
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
```

```
#install.packages("palmerpenguins")
library(palmerpenguins)
glimpse(penguins)
```

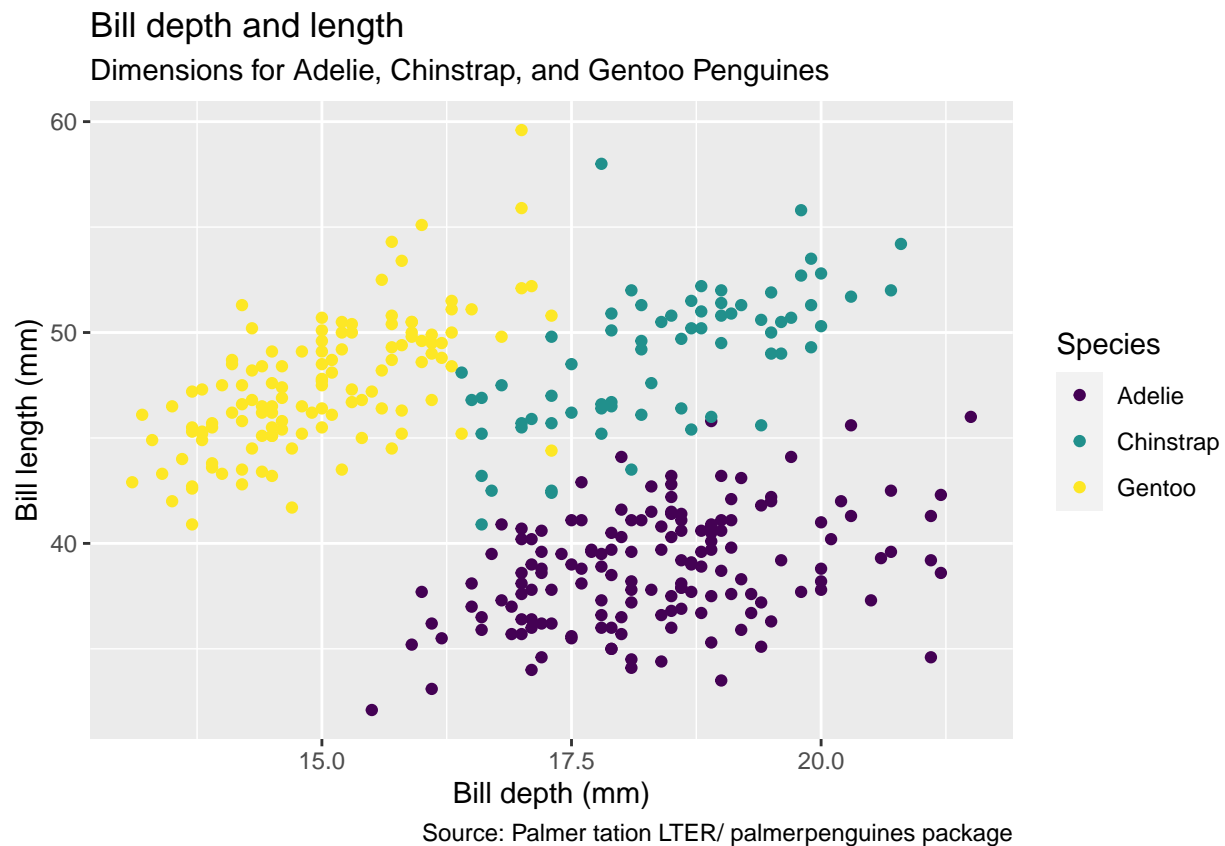
```
## Rows: 344
## Columns: 8
## $ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island       <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm <dbl> 18.7, 17.4, 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~
```

Question-2: Plots: Palmer Penguins Solutions:

```
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 tation LTER/ palmerpenguins",
      scale_colour_viridis_d())
```

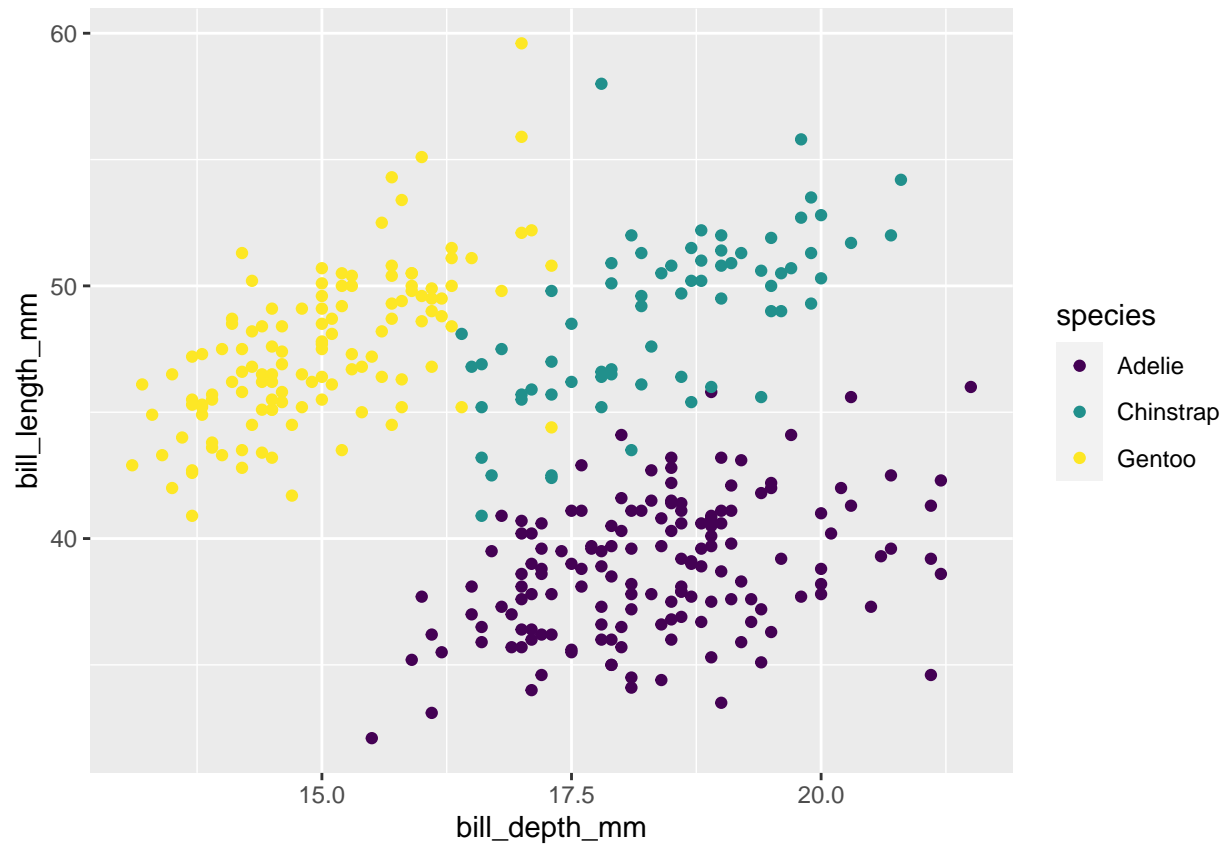
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-3: Argument Names; Colour Solutions:

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

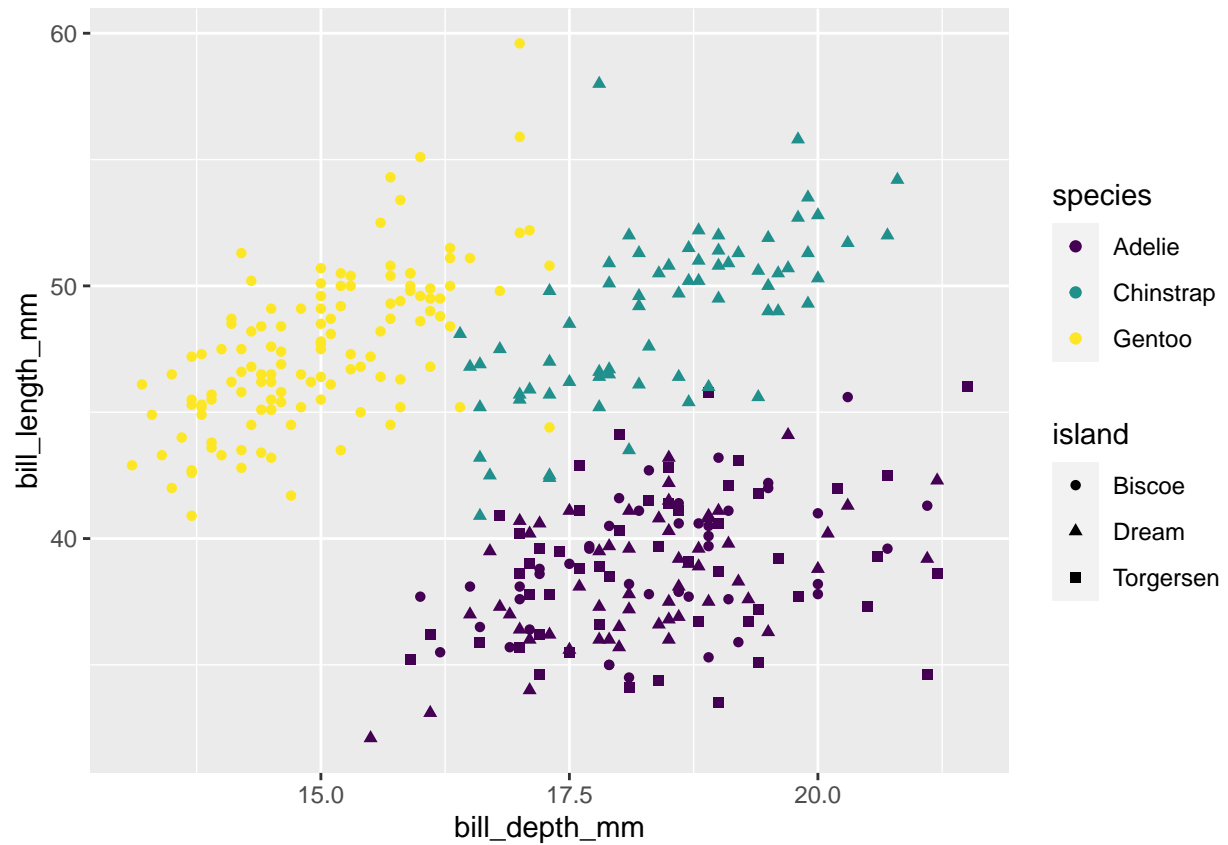
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-4: Argument Names; Shape = Island Solutions:

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

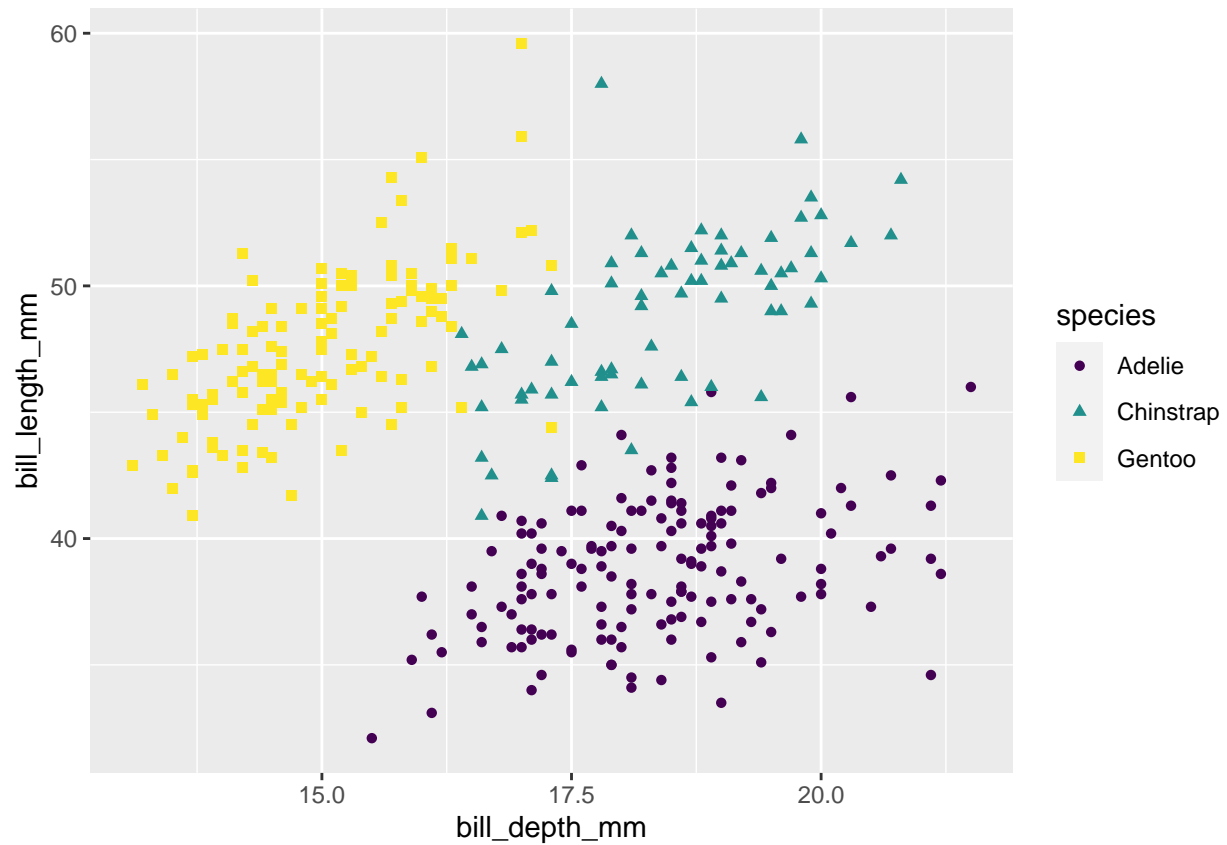
```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```



Question-5: Argument Names; Shape = Species Solutions:

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

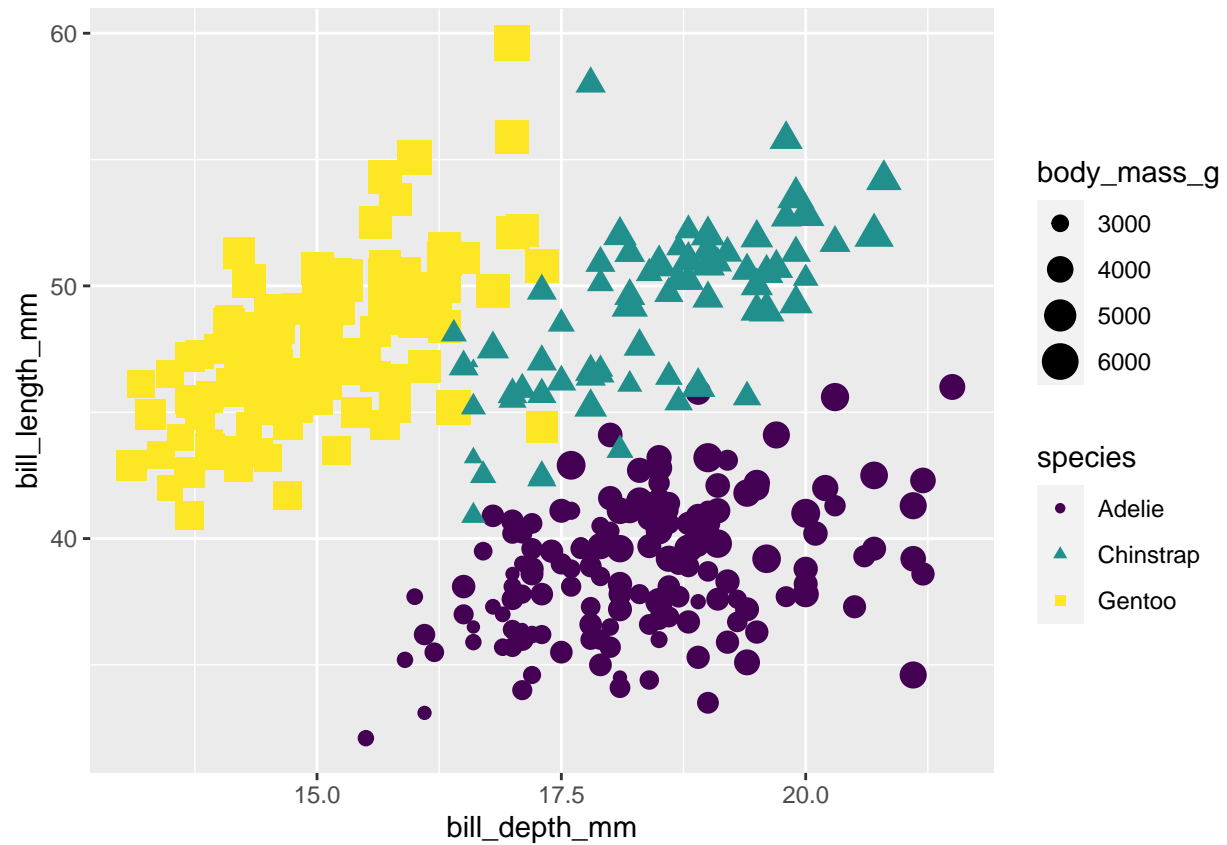
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-6: Argument Names; Size Solutions:

```
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()
```

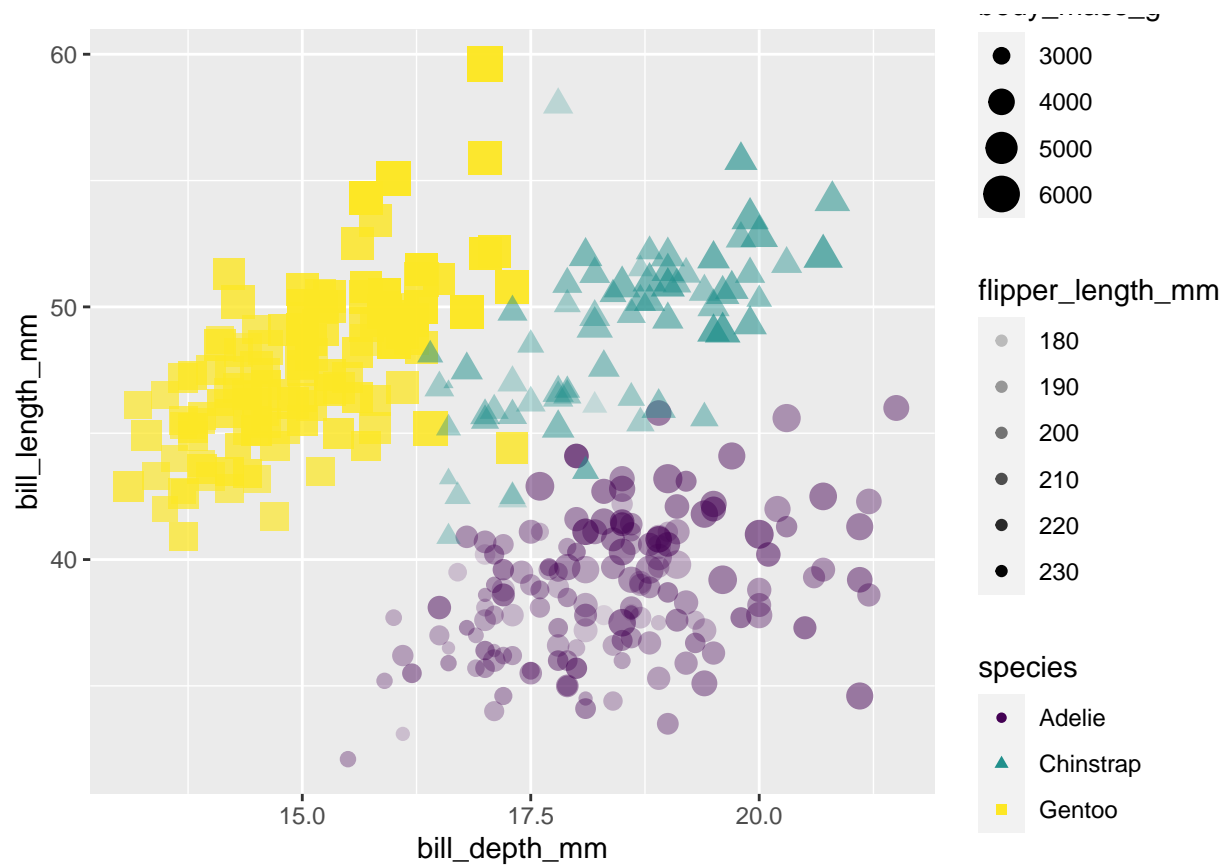
```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```



Question-7: Alpha Solutions:

```
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()
```

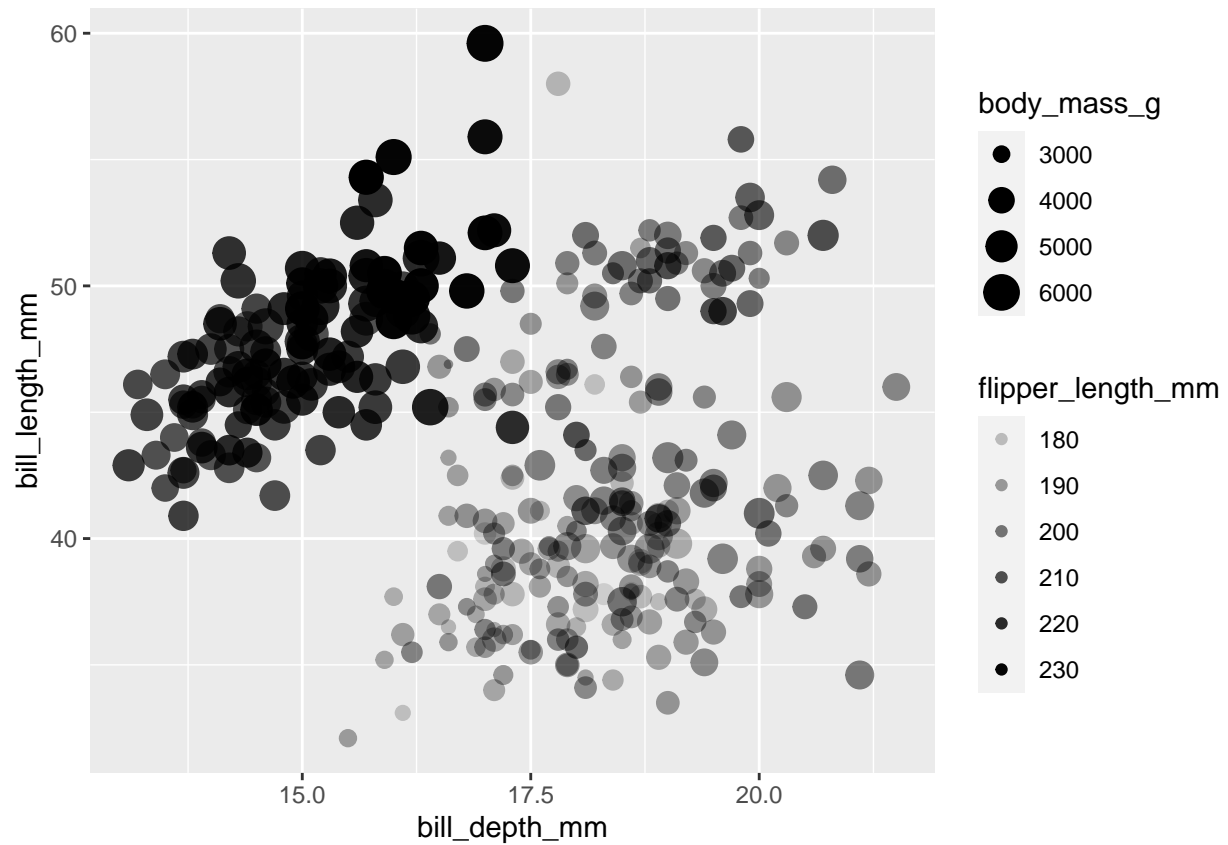
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-8 Mapping Solutions:

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

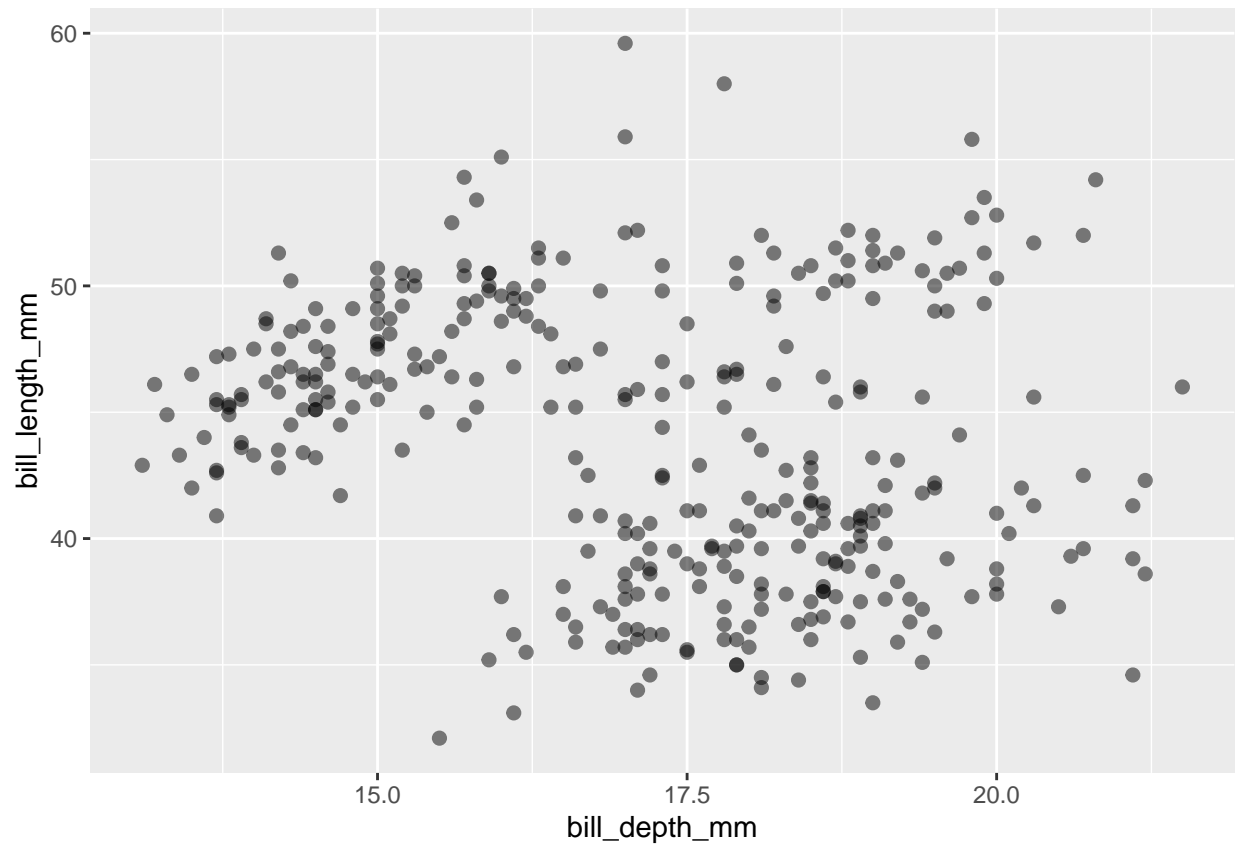
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-9: Setting Solutions:

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

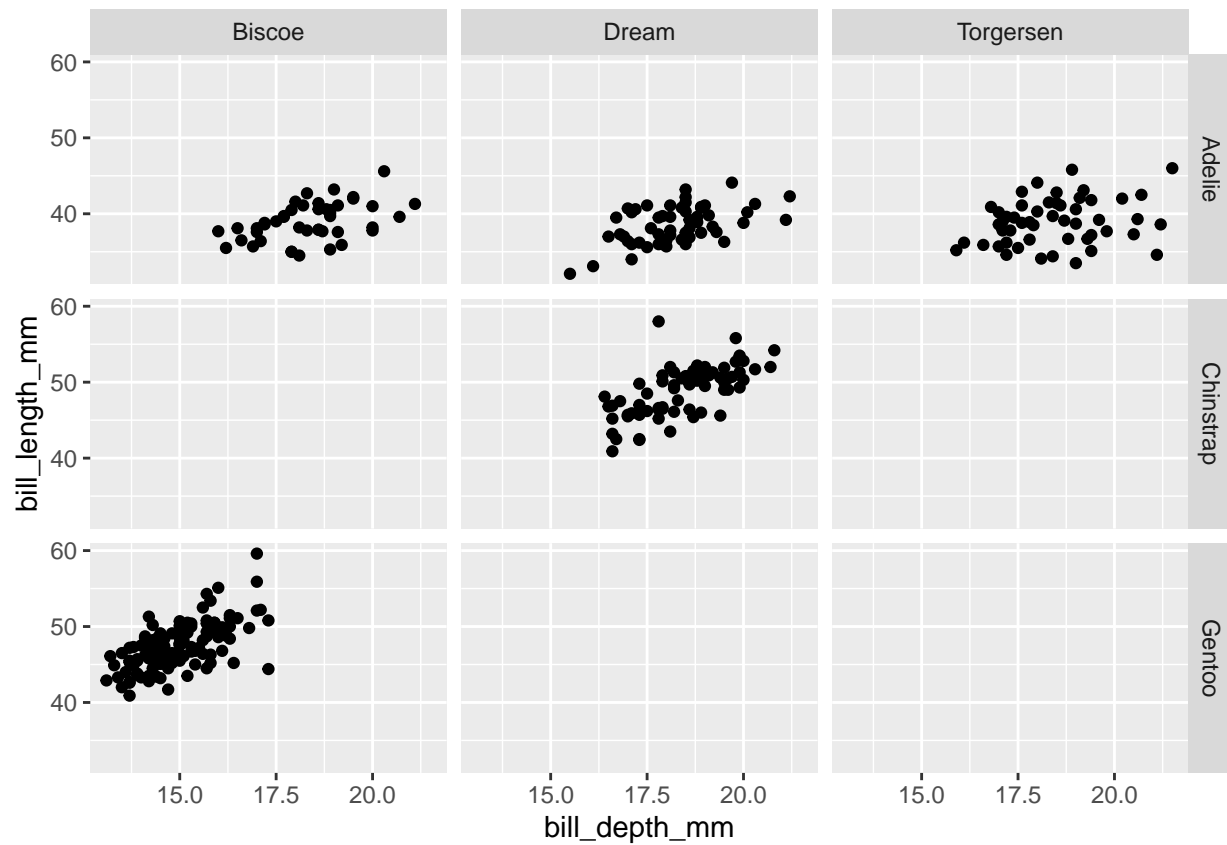
```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```

Question-10: Faceting Solutions:

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

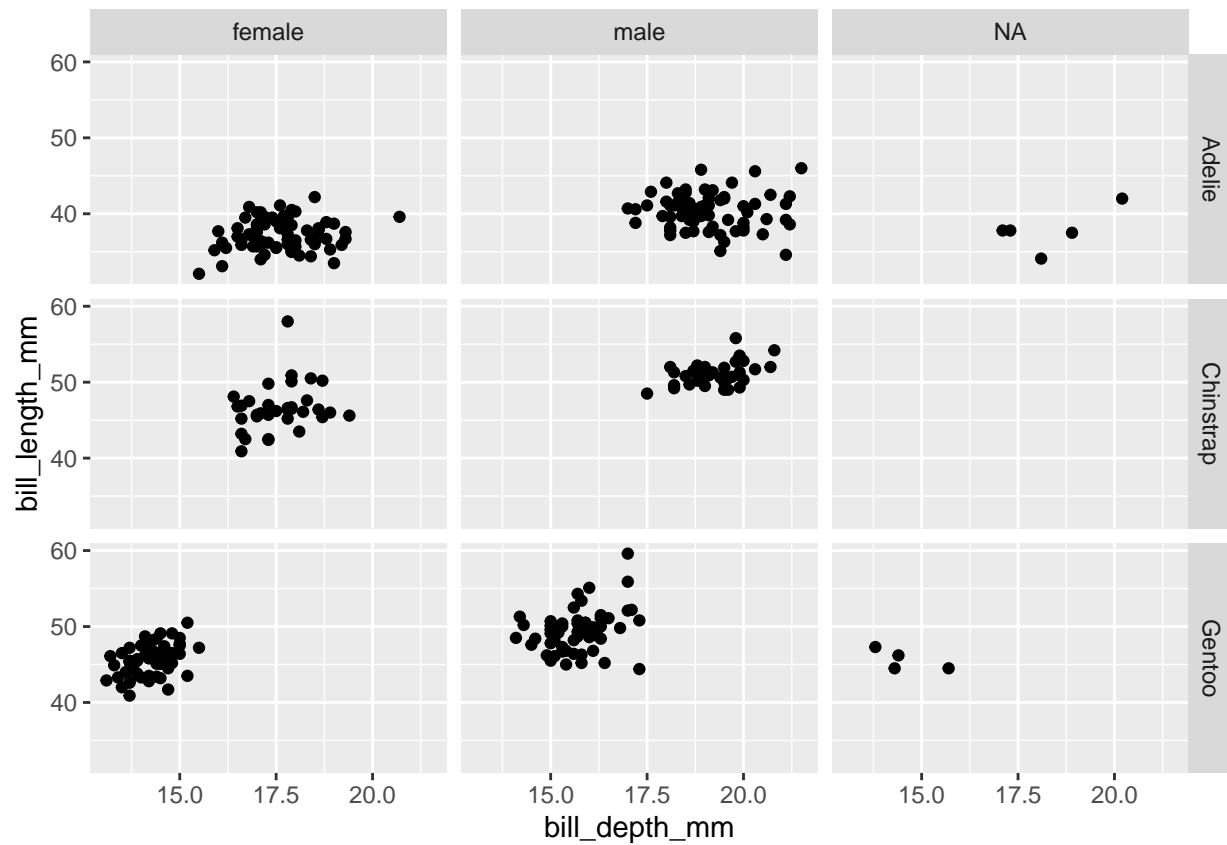
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-11: Faceting 2 Solutions:

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

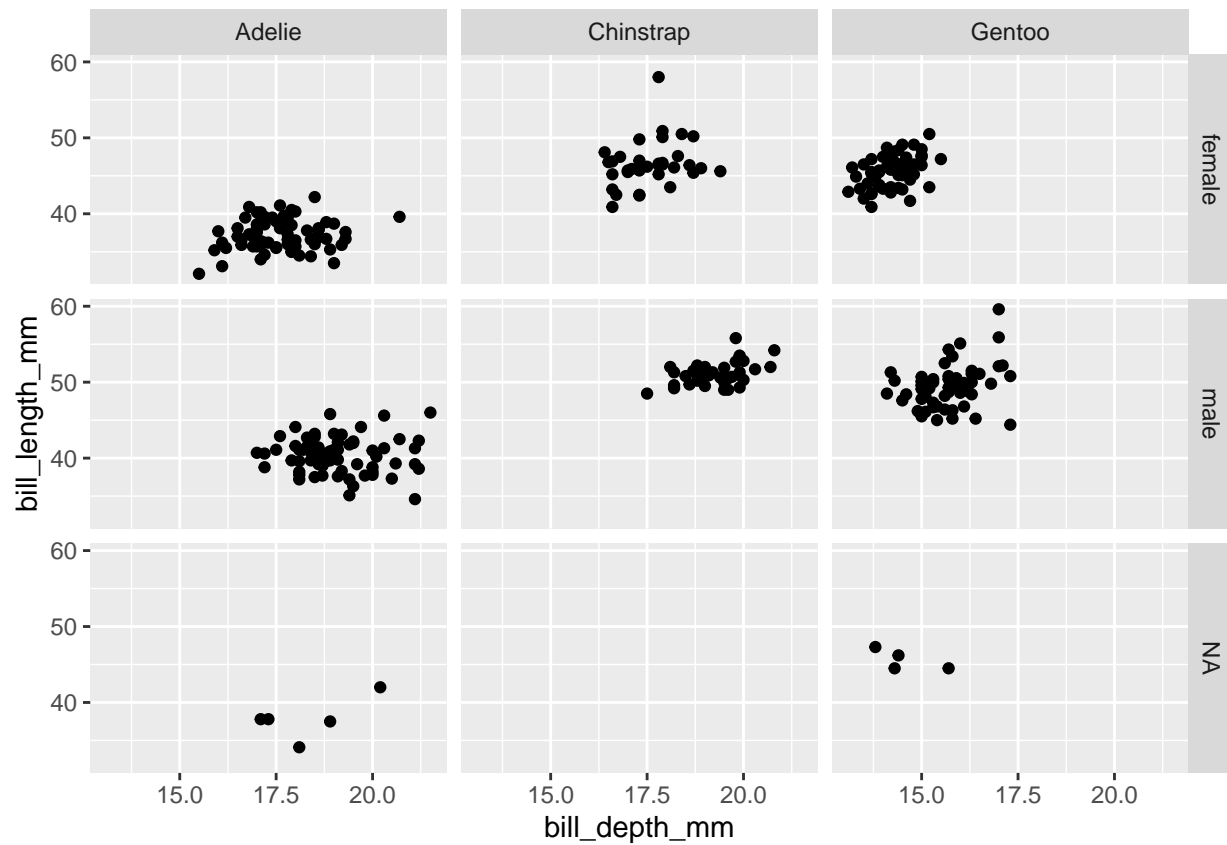
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-12: Faceting 3 Solutions:

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

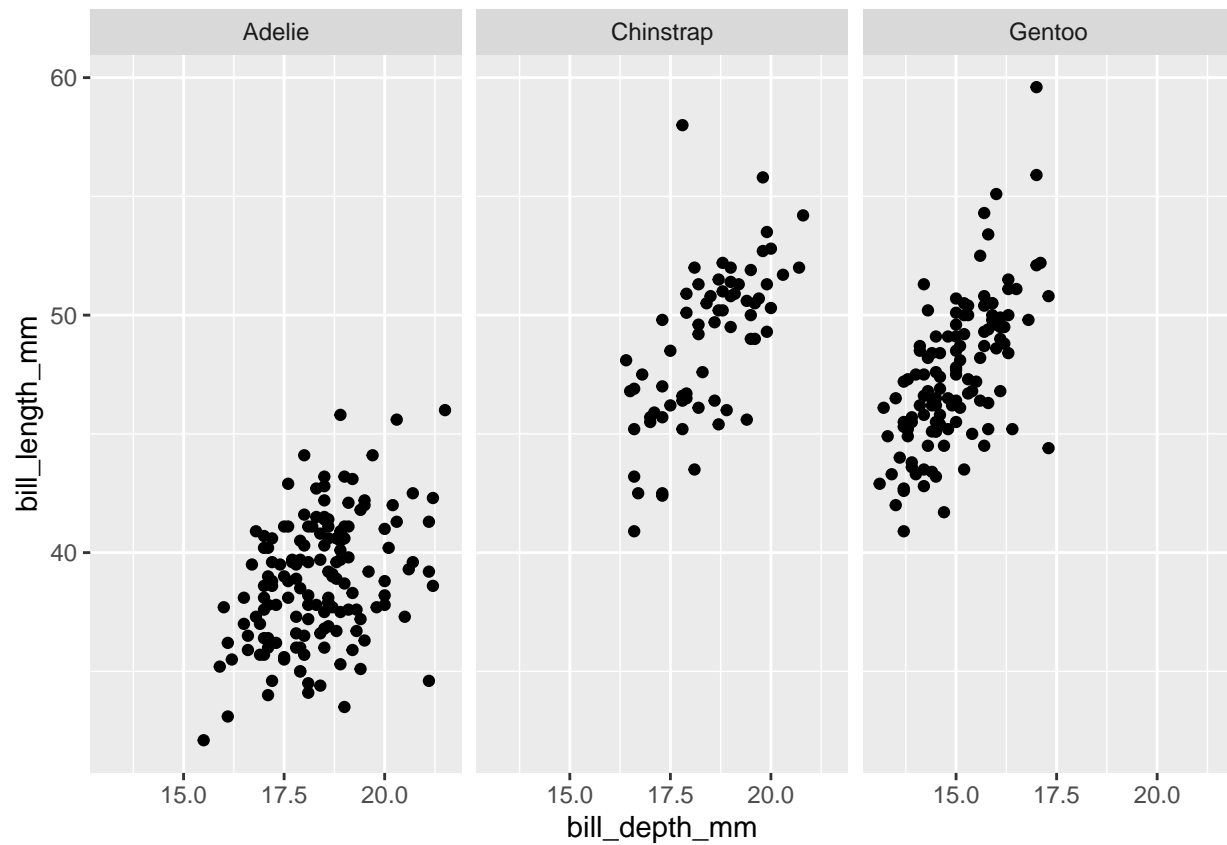
```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```



Question-13:Faceting 4 Solutions:

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

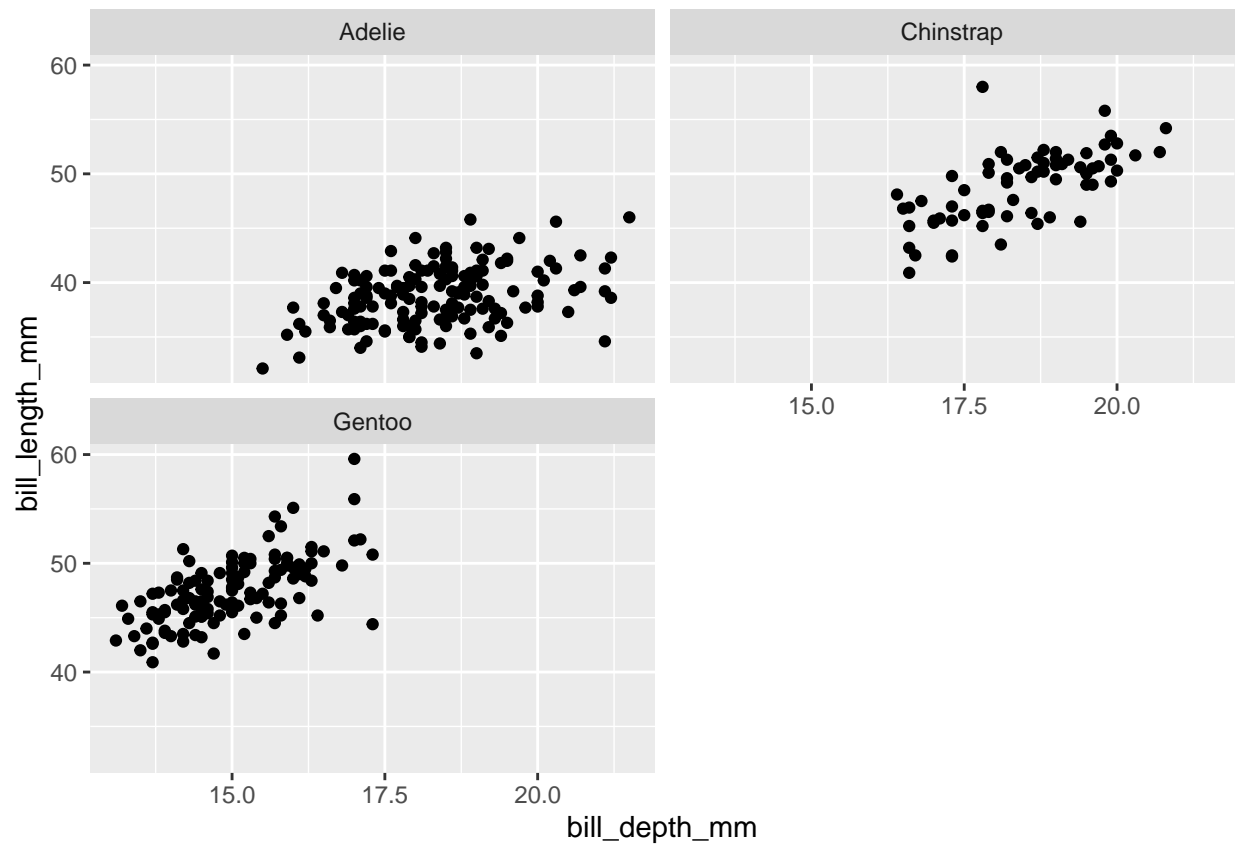
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-14:Faceting 5 Solutions:

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

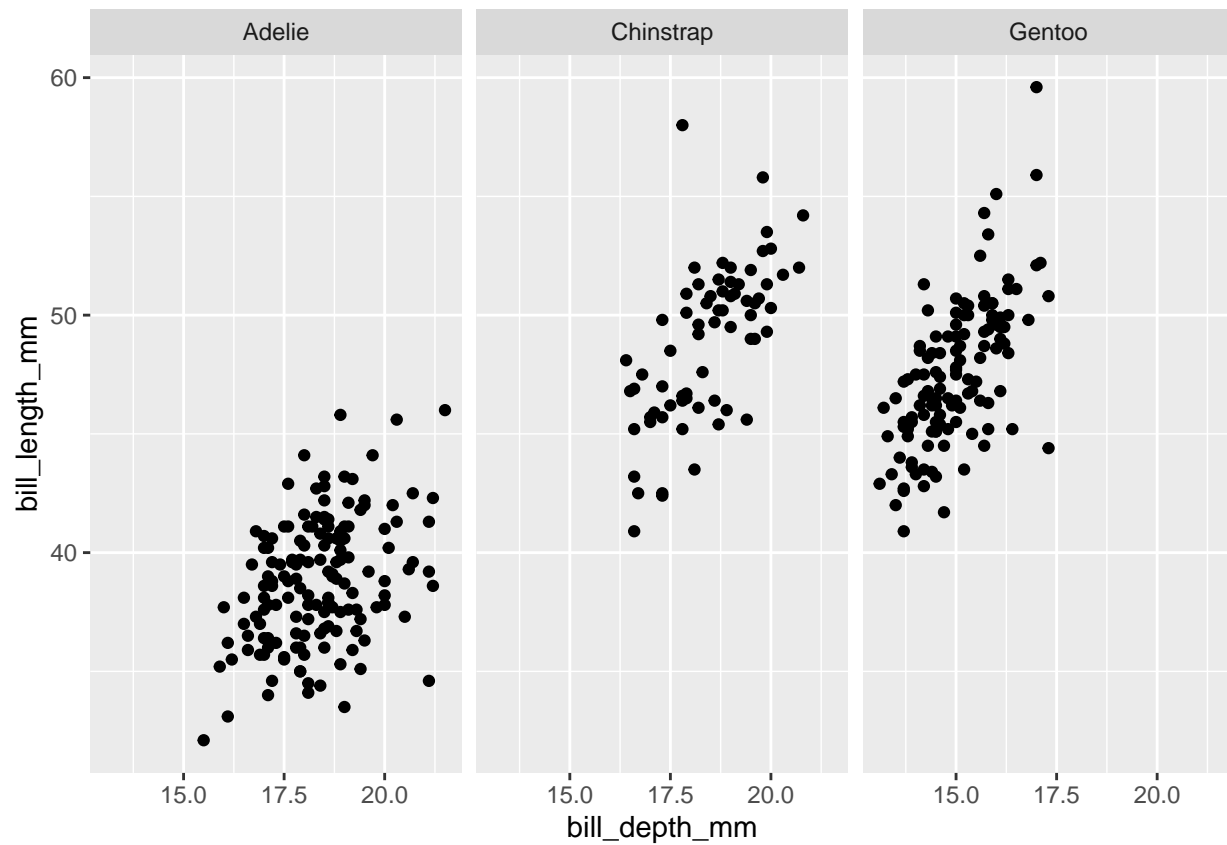
```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```



Question-15:Faceting 6 Solutions:

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

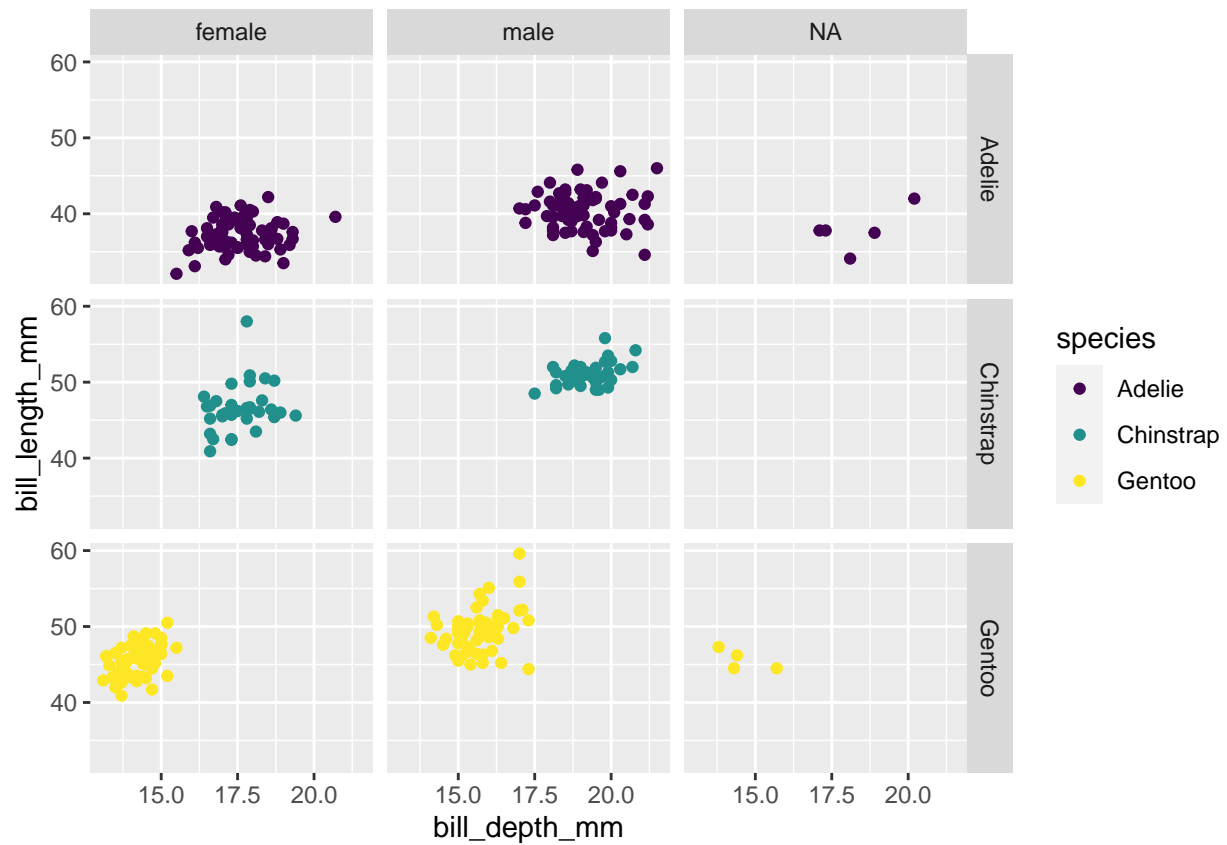
```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```



Question-16:Facet and Colour Solutions:

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

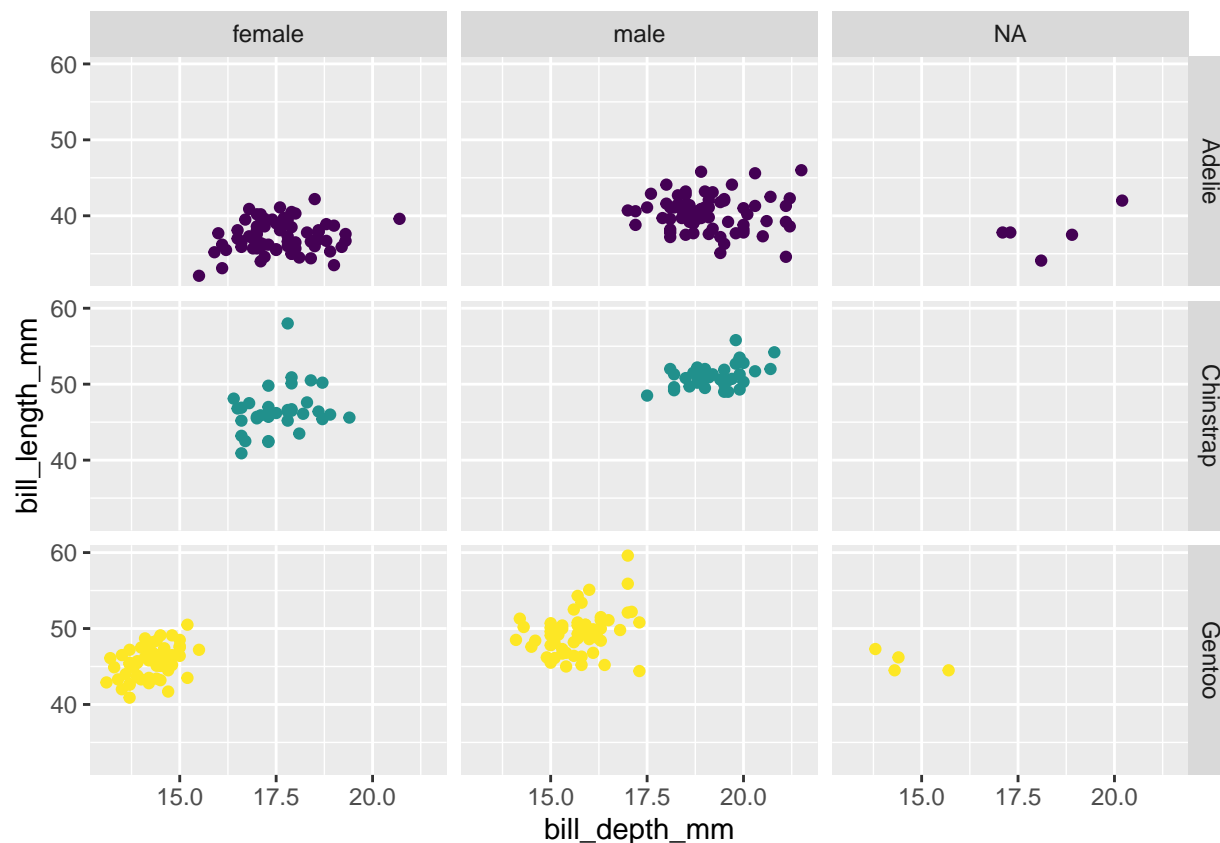
Warning: Removed 2 rows containing missing values ('geom_point()').



Question-17:Face and Colour, No Legend Solutions:

```
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()').



Question-18: Uploading the Lending Club Data Solutions:

```
# install.packages("openintro")
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~

```

Question-19: Select Variables Solutions:

```

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

```

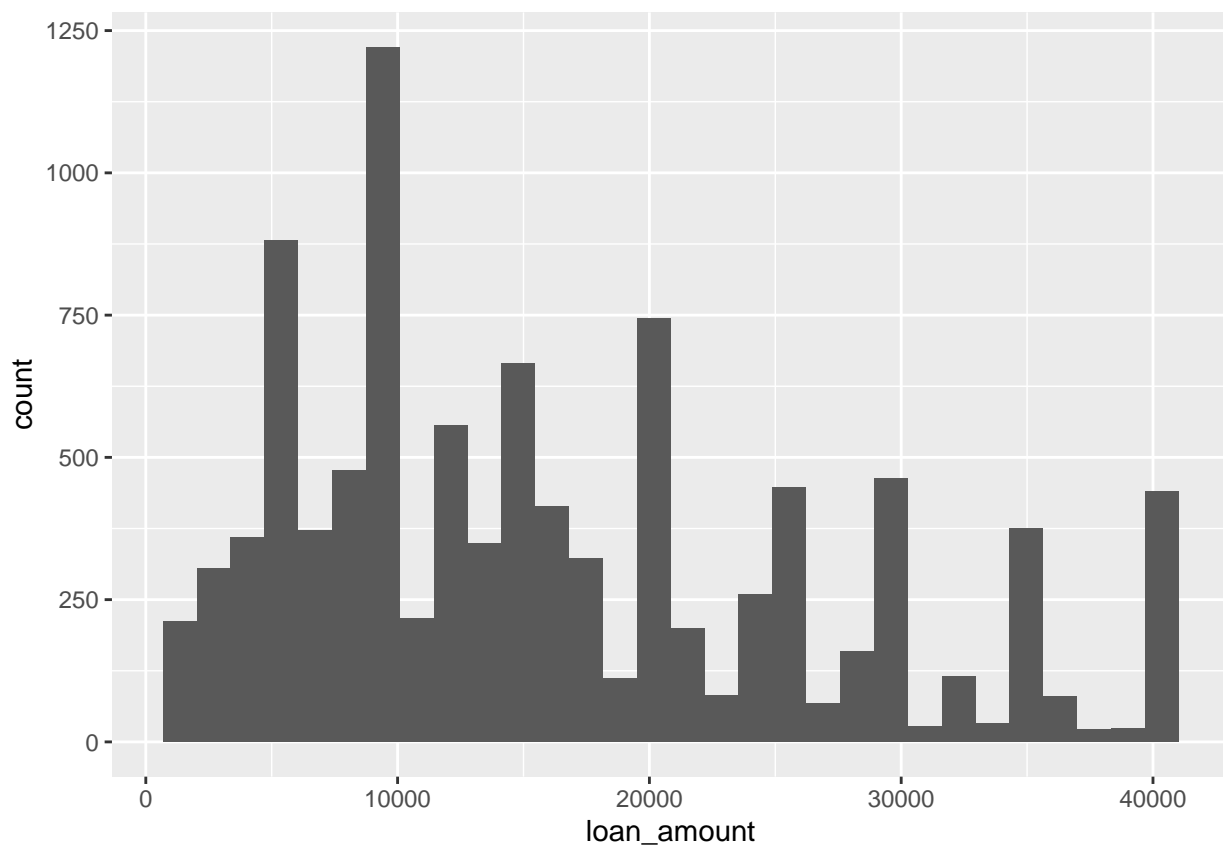
```
glimpse(loans)
```

```
## 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~
```

Question-20: Histograms Solutions:

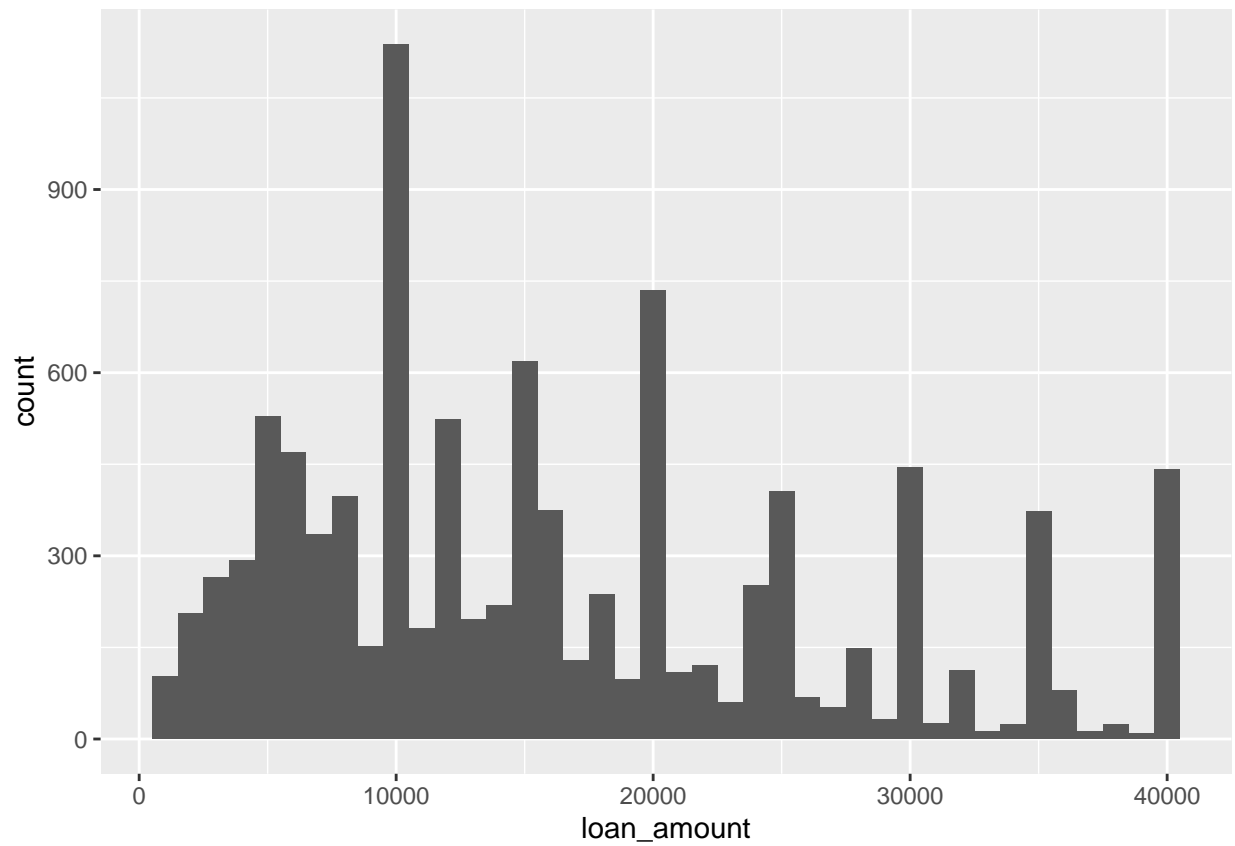
```
ggplot(loans) + aes(x = loan_amount) +  
  geom_histogram()
```

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



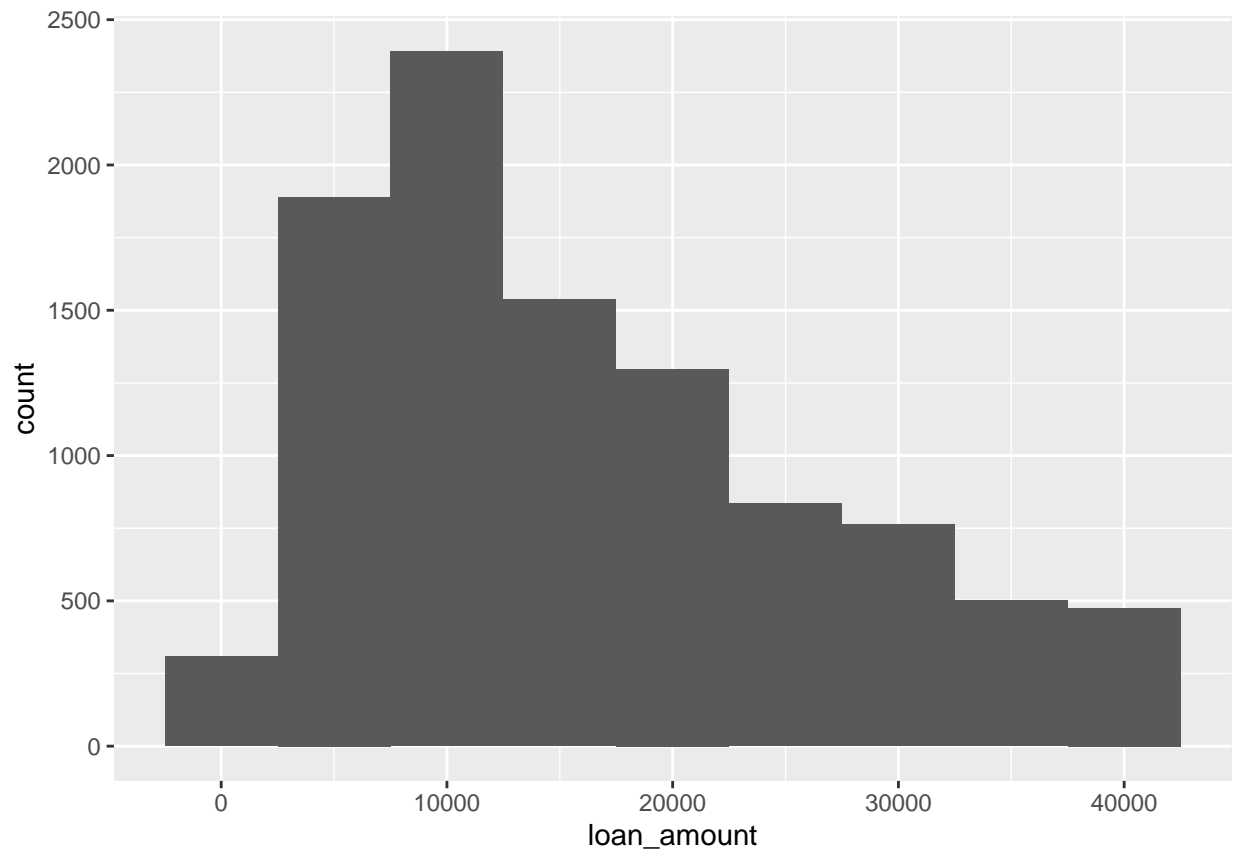
Question-21: Histograms and binwidth=1000 Solutions:

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



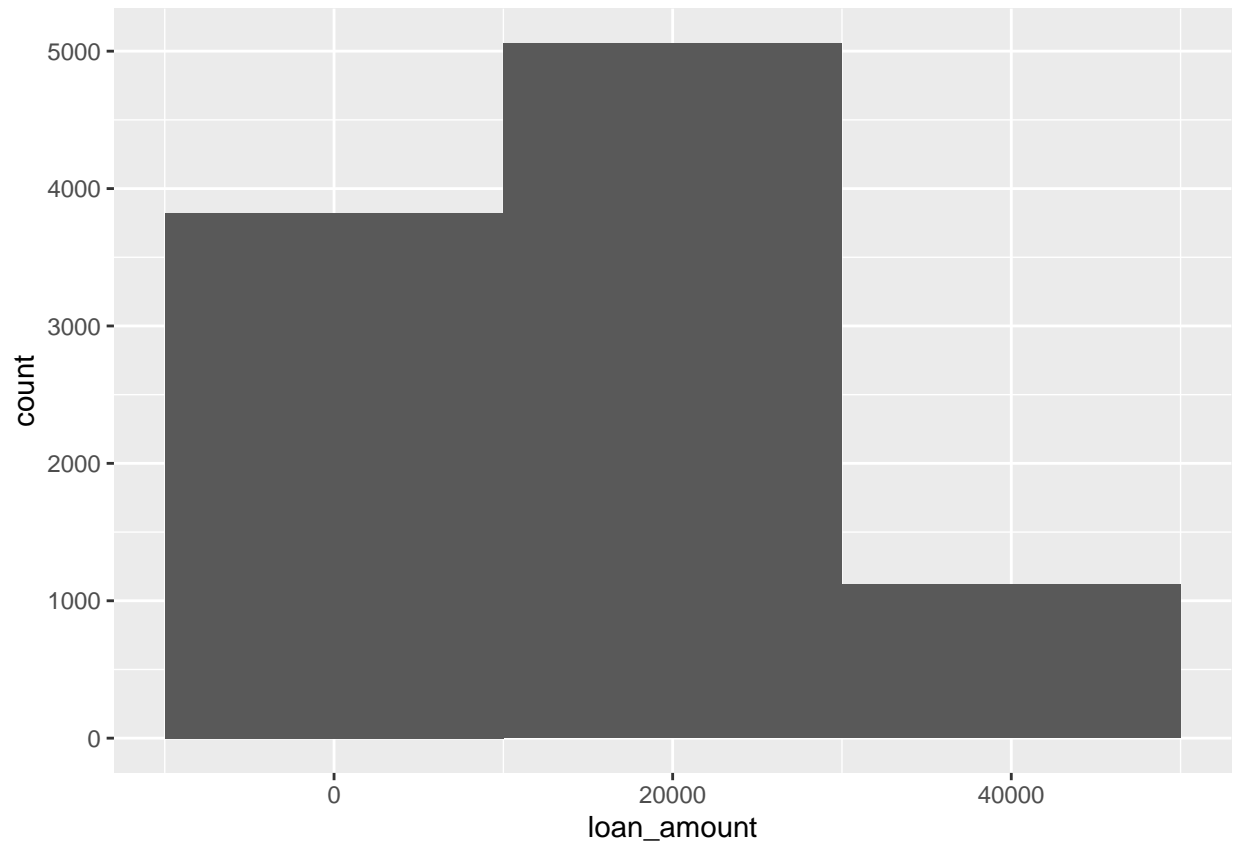
Question-22: Histograms and binwidth = 5000 Solutions:

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



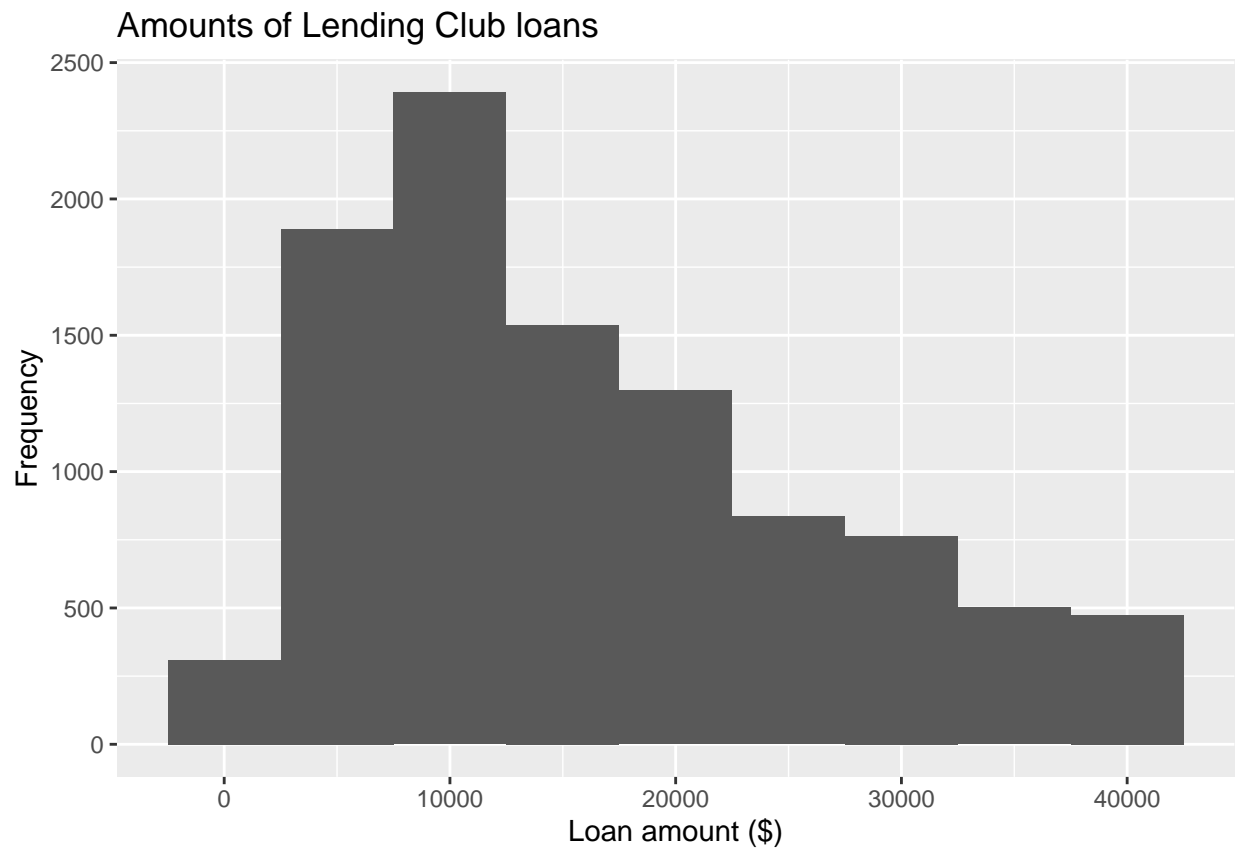
Question-23: Histograms and binwidth=20000 Solutions:

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



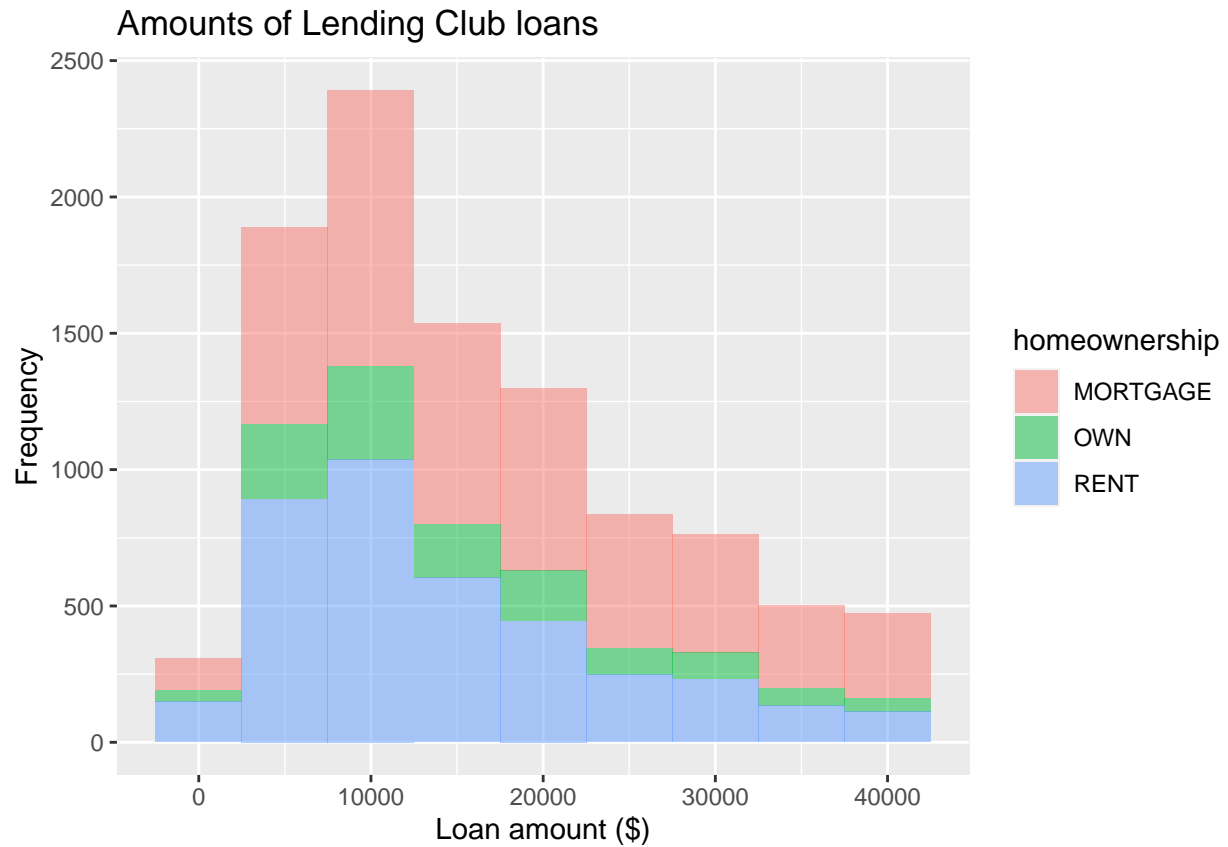
Question-24: Customizing Histograms Solutions:

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



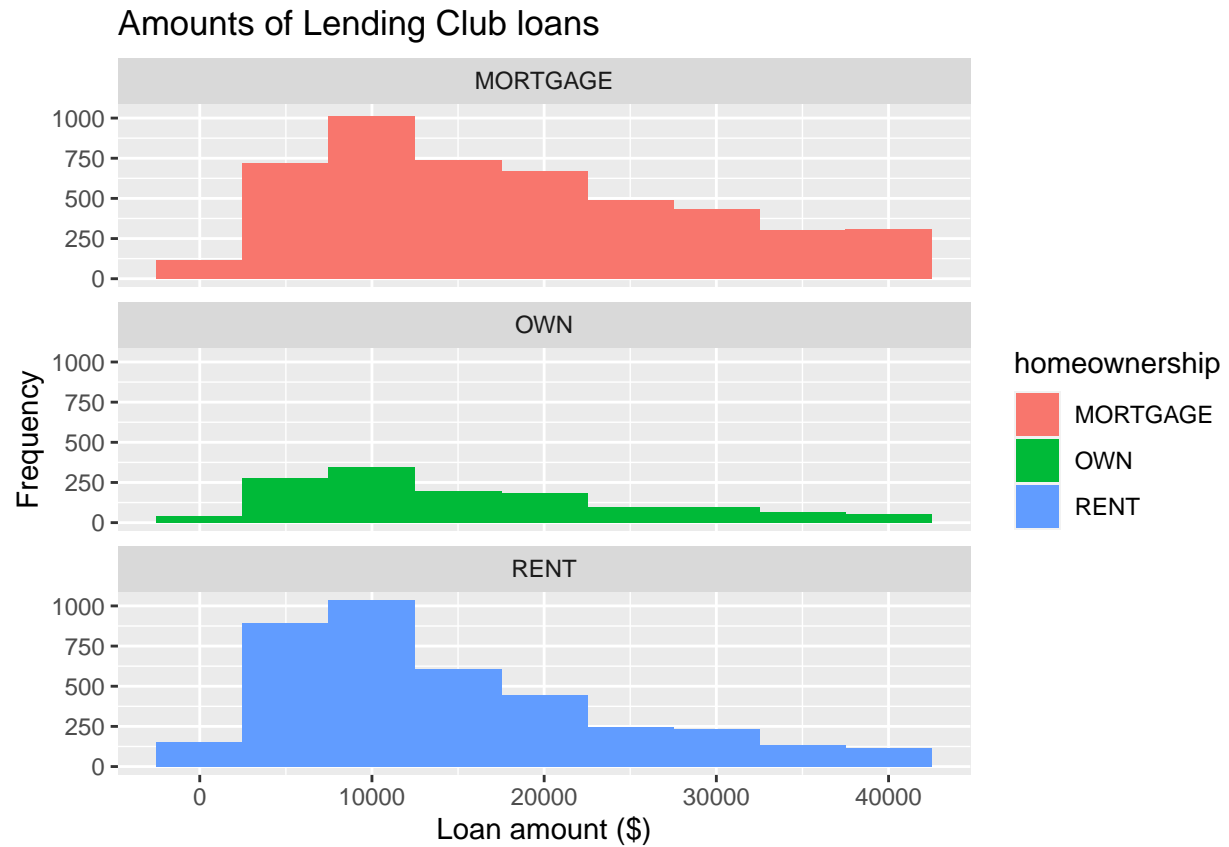
Question-25: Fill with a categorical variable Solutions:

```
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")
```



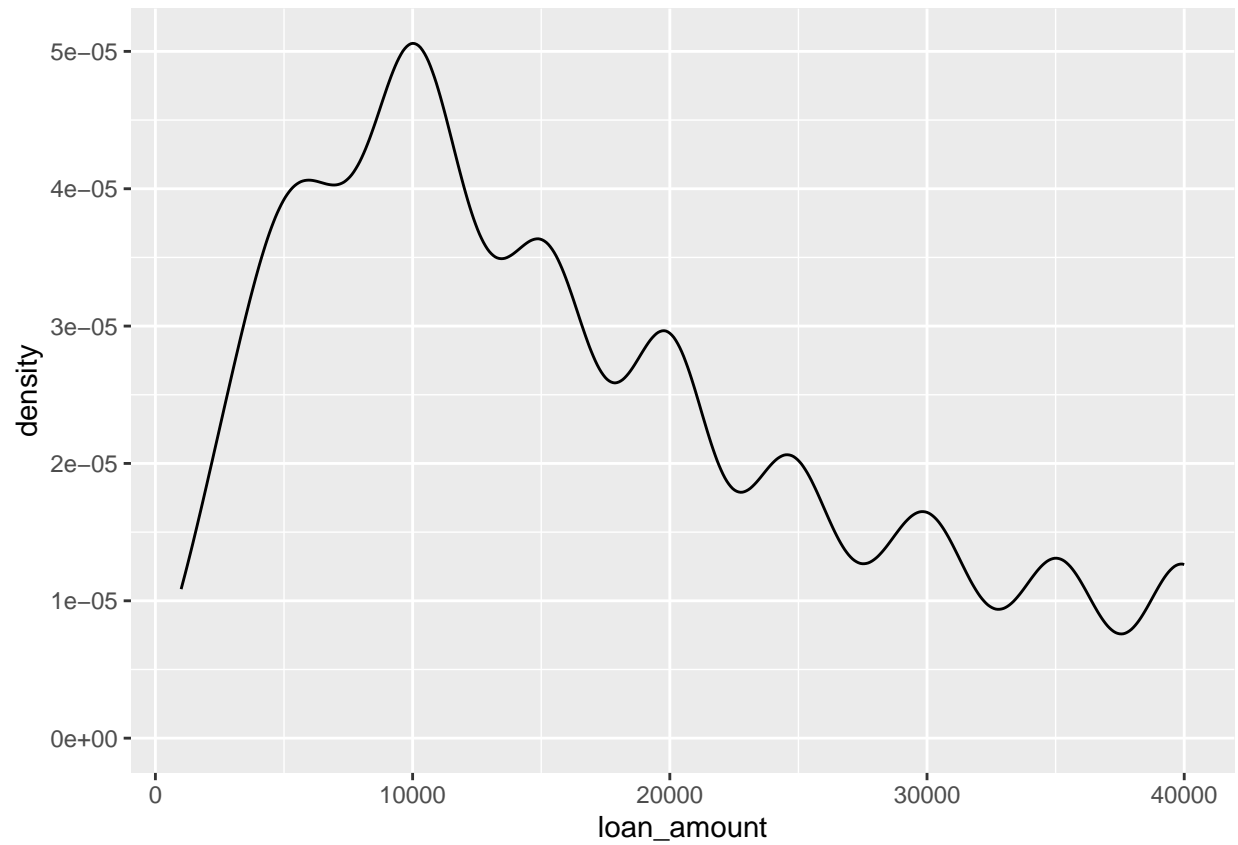
Question-26: Facet with a categorical variable Solutions:

```
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)
```

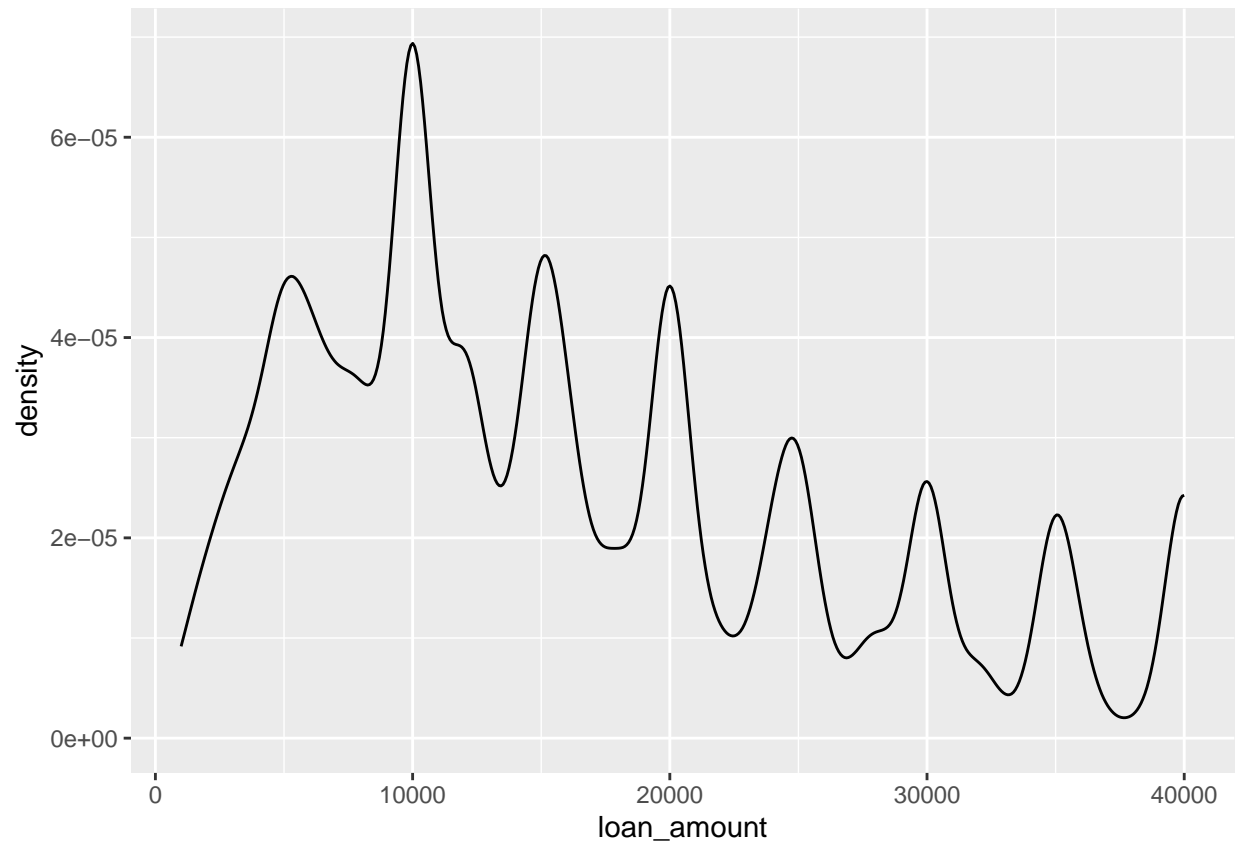
Question-27: Density plot Solutions:

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

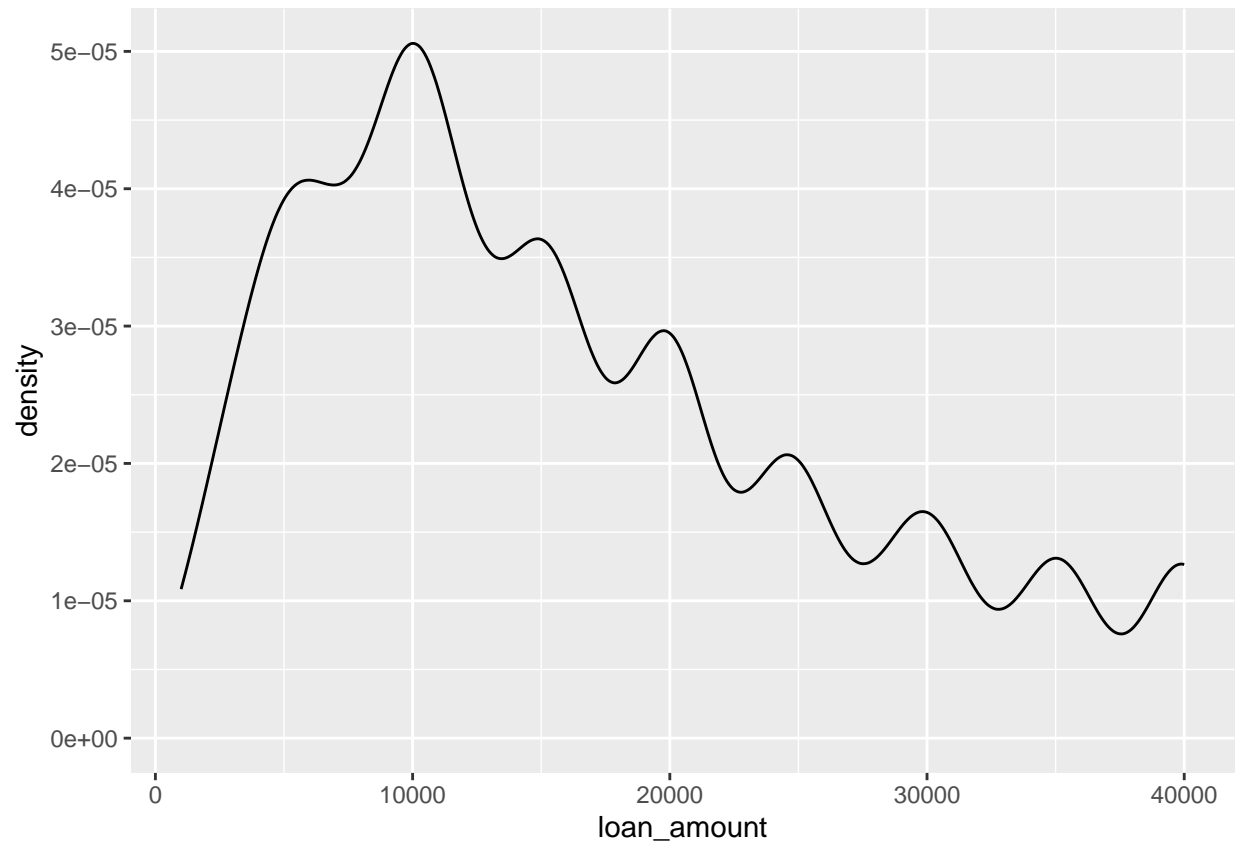


Question-28: Density plots and adjusting bandwidth Solutions:

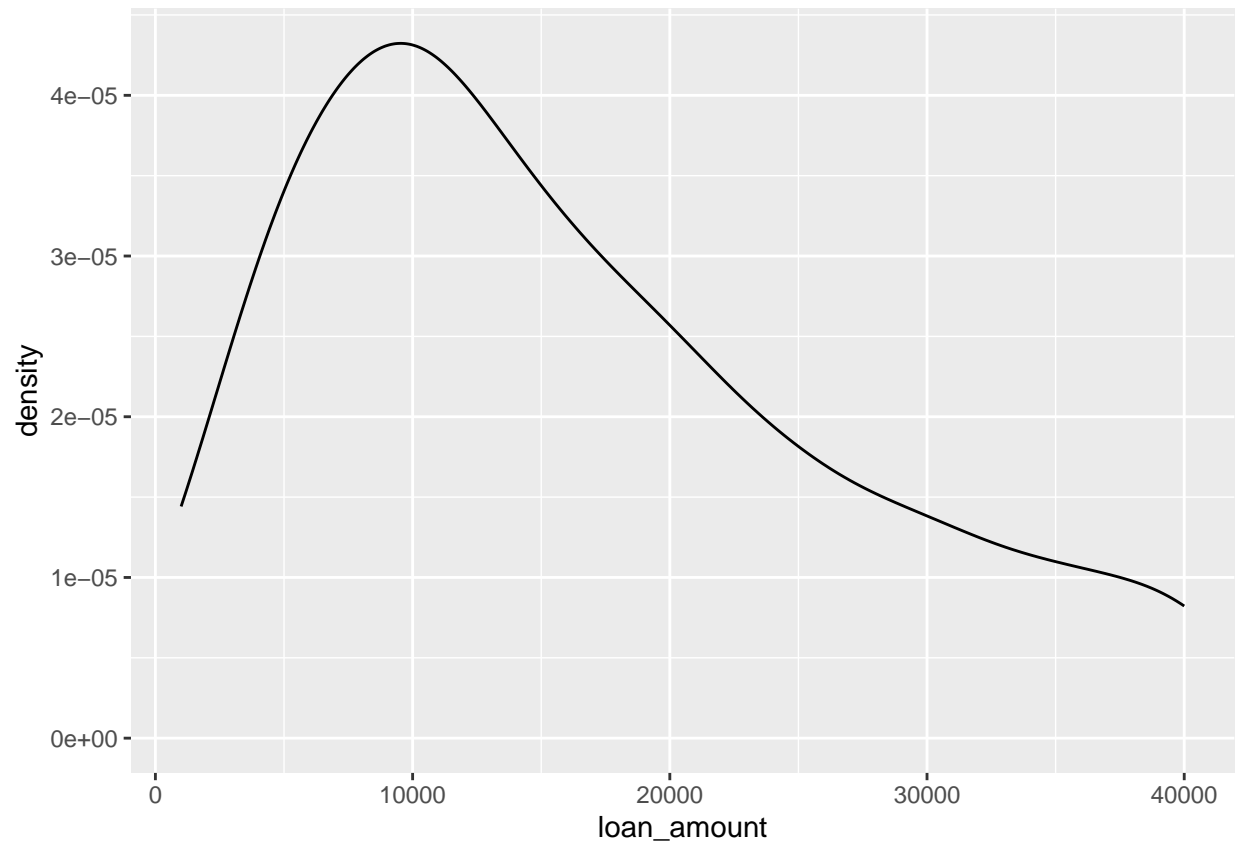
```
ggplot(loans, aes(x = loan_amount)) +  
  geom_density(adjust = 0.5)
```



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

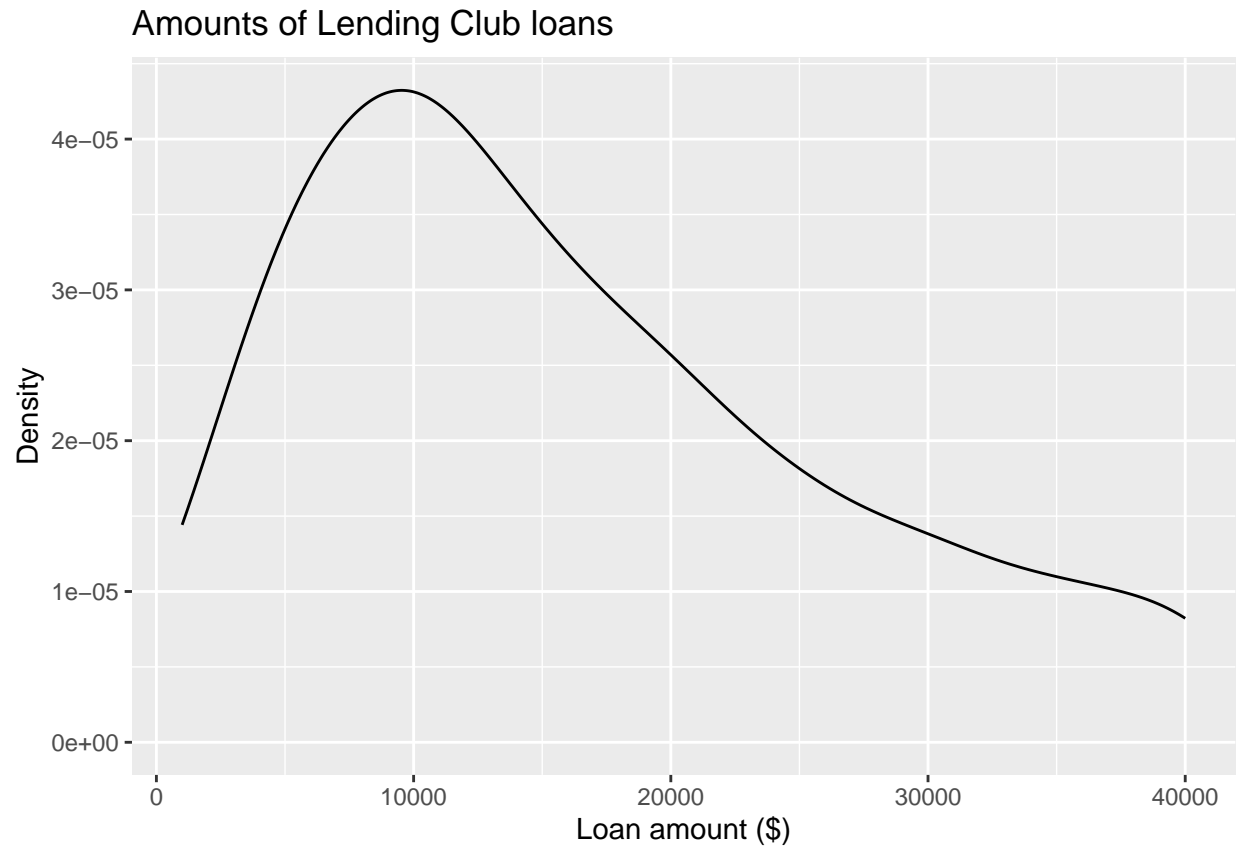


```
ggplot(loans, aes(x = loan_amount)) +  
  geom_density(adjust = 2)
```



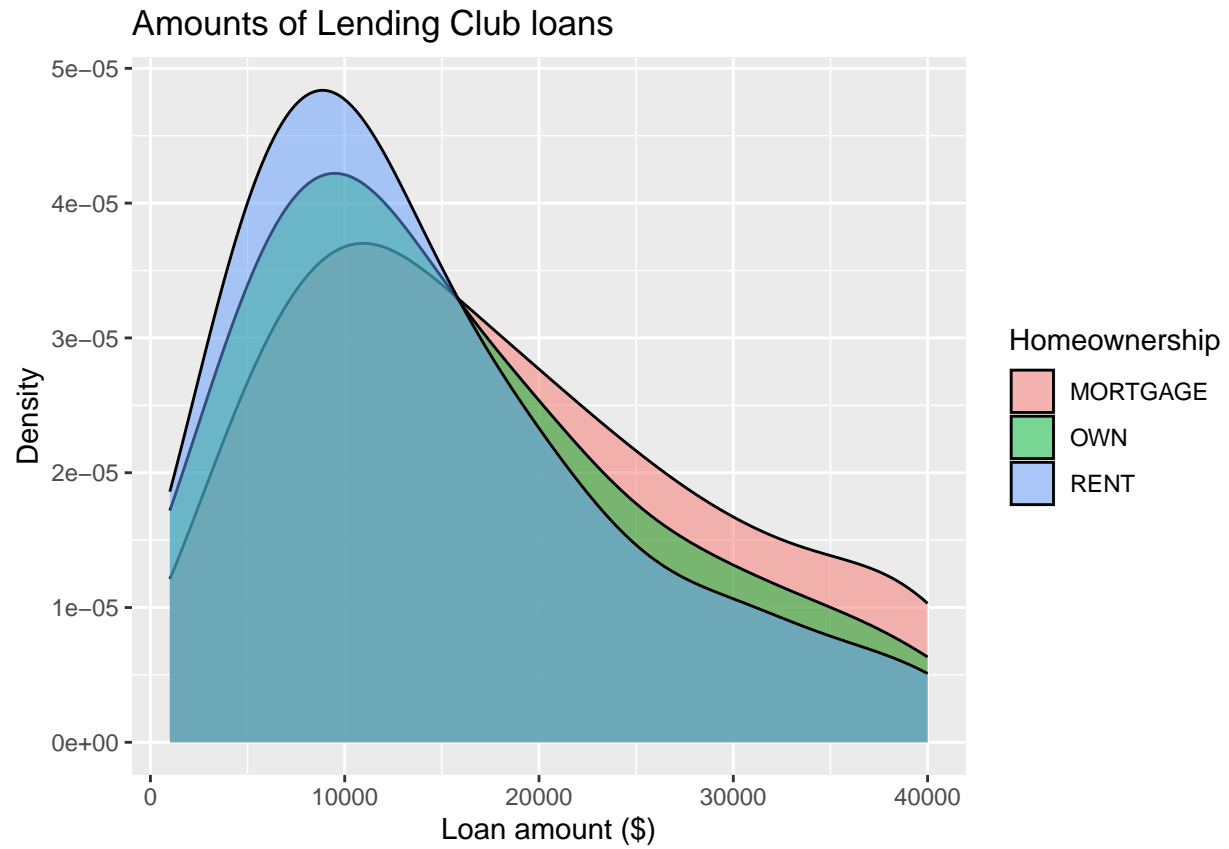
Question-29: Customizing density plots Solutions:

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



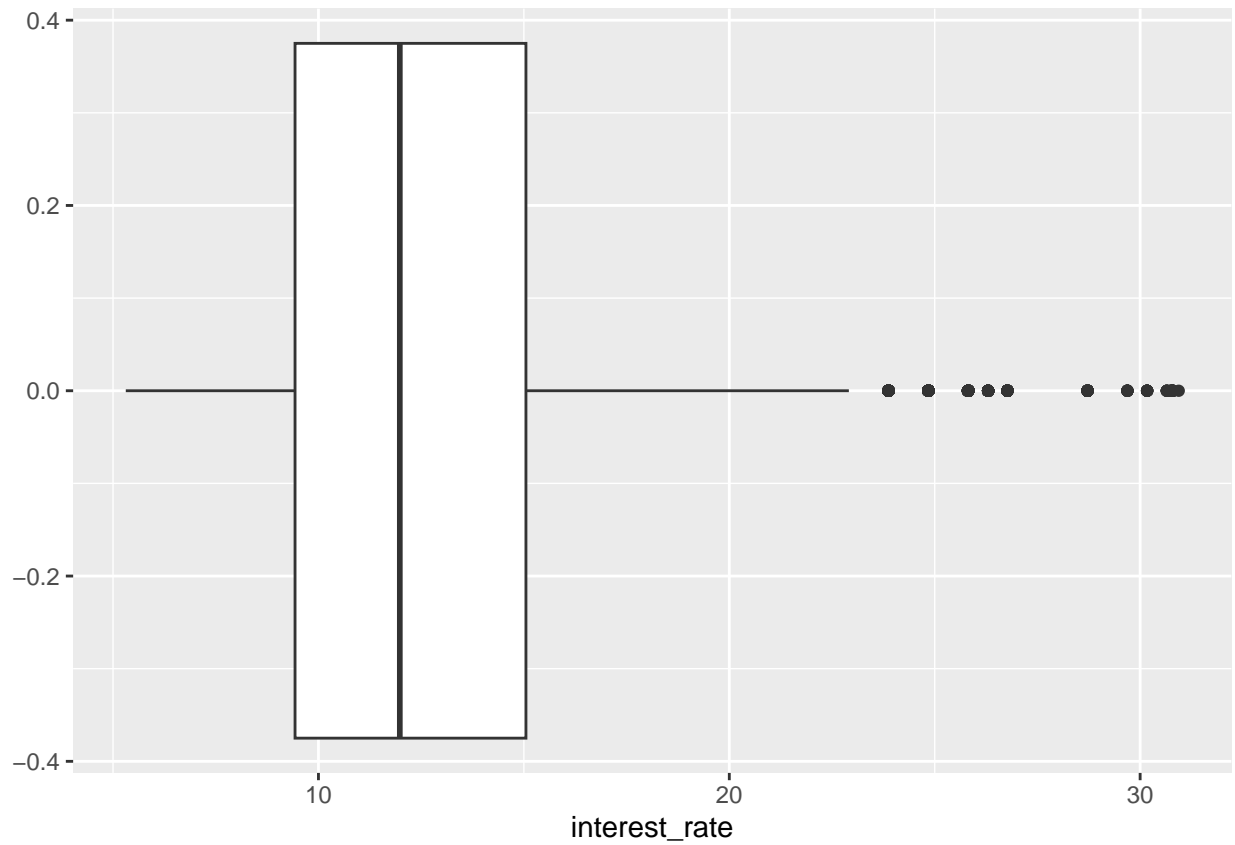
Question-30: Adding a categorical variable Solutions:

```
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")
```



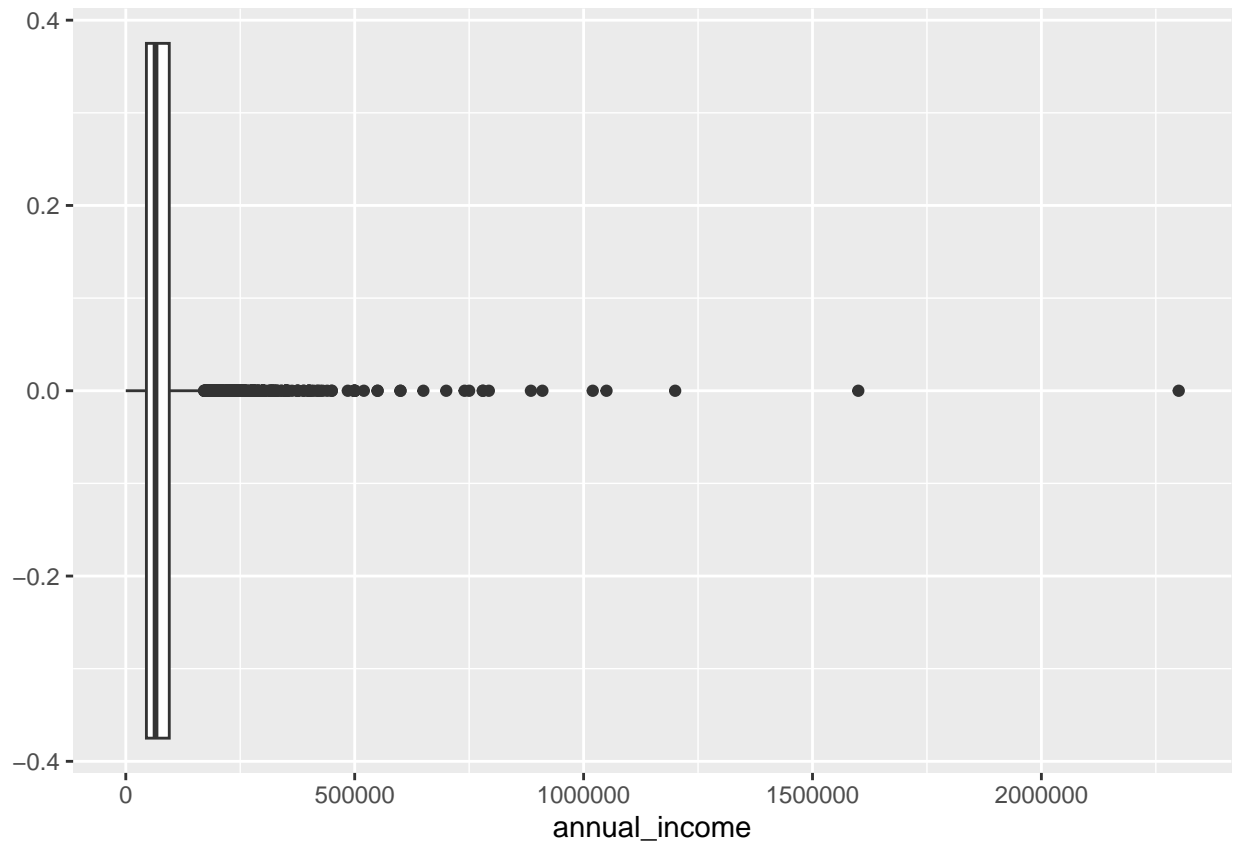
Question-31: Box plot Solutions:

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



Question-32: Box plot and outliers Solutions:

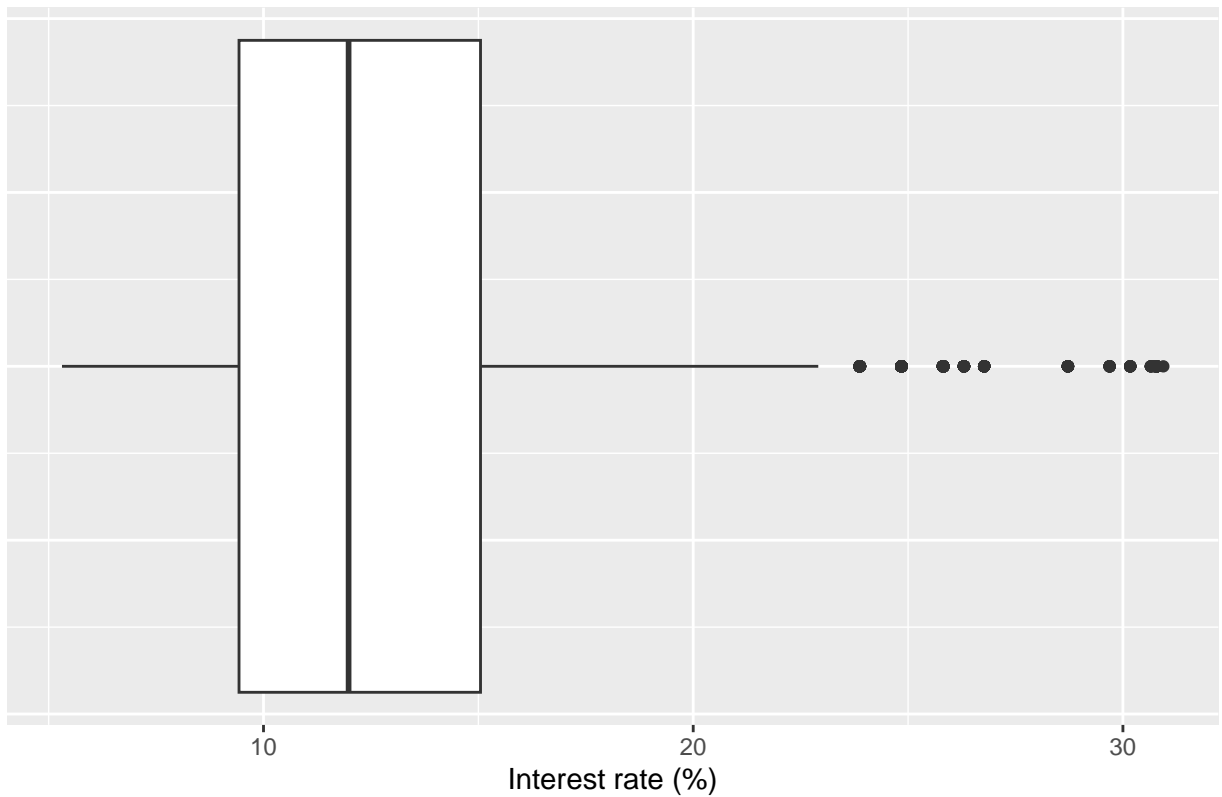
```
ggplot(loans, aes(x = annual_income)) +  
  geom_boxplot()
```

Question-33: Customizing box plots Solutions:

```
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() )
```

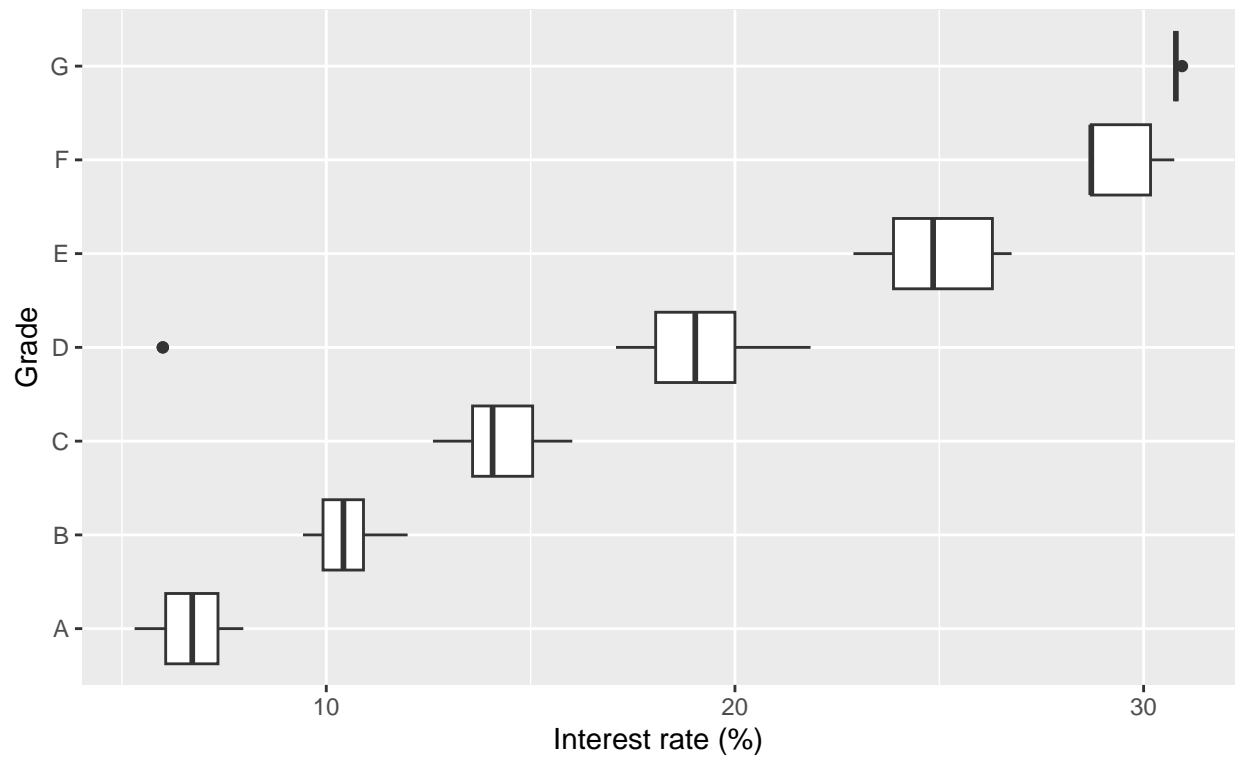
Interest rates of Lending Club loans



Question-34: Adding a categoric variable Solutions:

```
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")
```

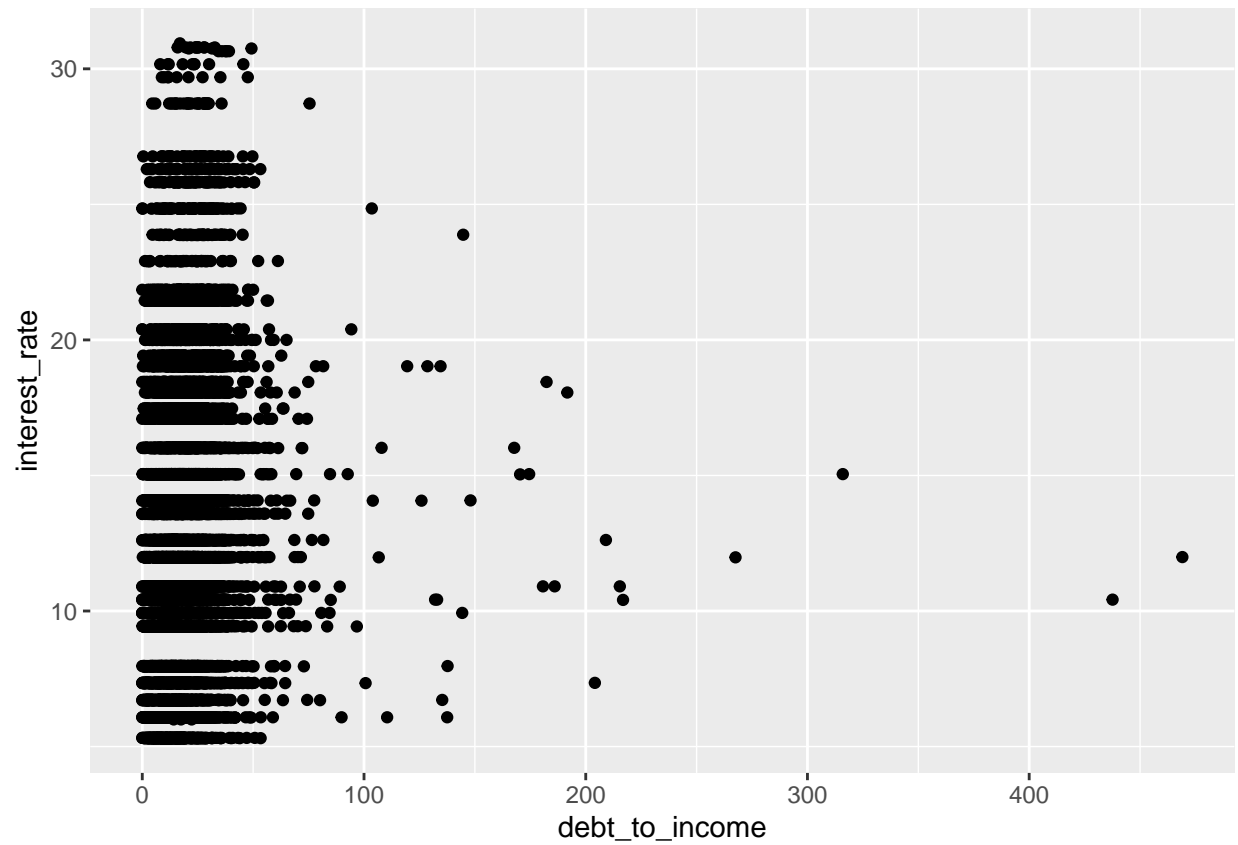
Interest rates of Lending Club loans by grade of loan



Question-35: Scatterplot Solutions:

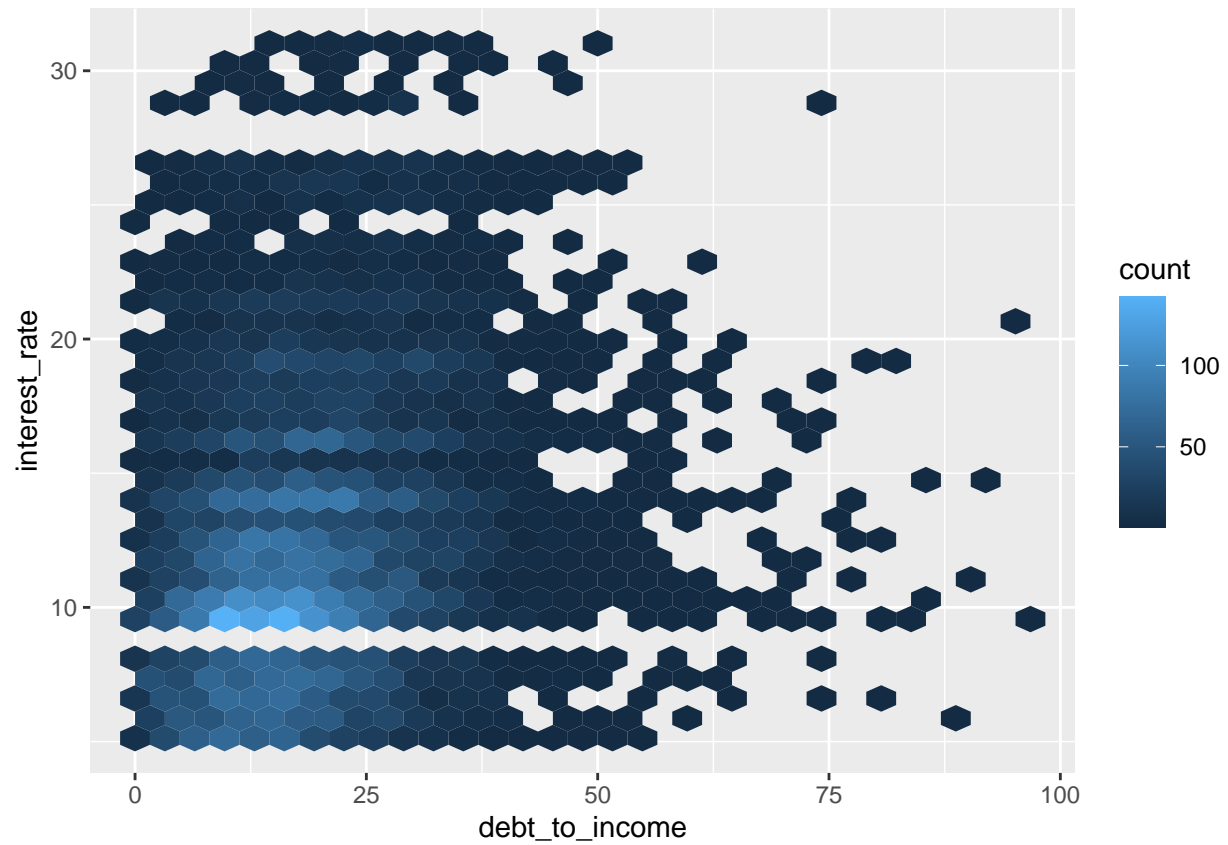
```
ggplot(loans, aes(x = debt_to_income, y = interest_rate)) +  
  geom_point()
```

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



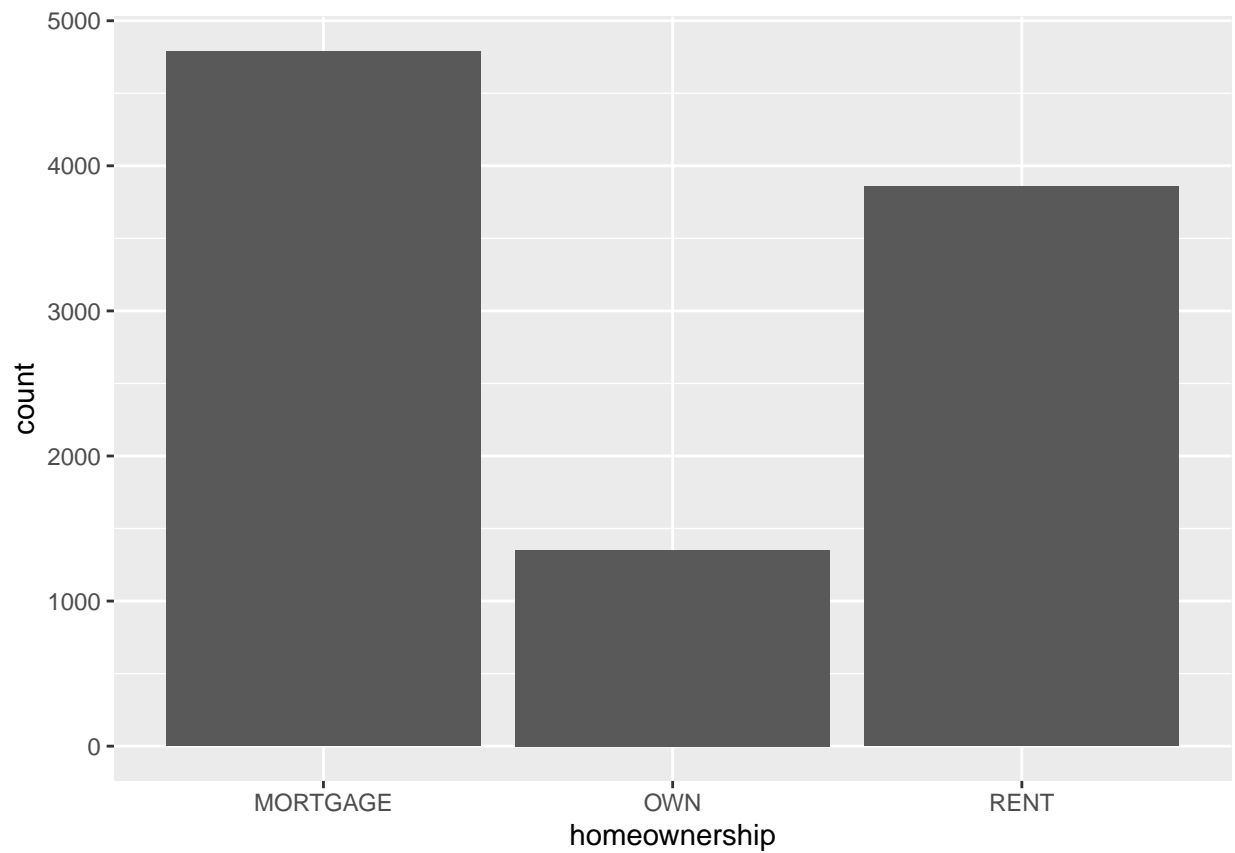
Question-36: Hex Plot Solutions:

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



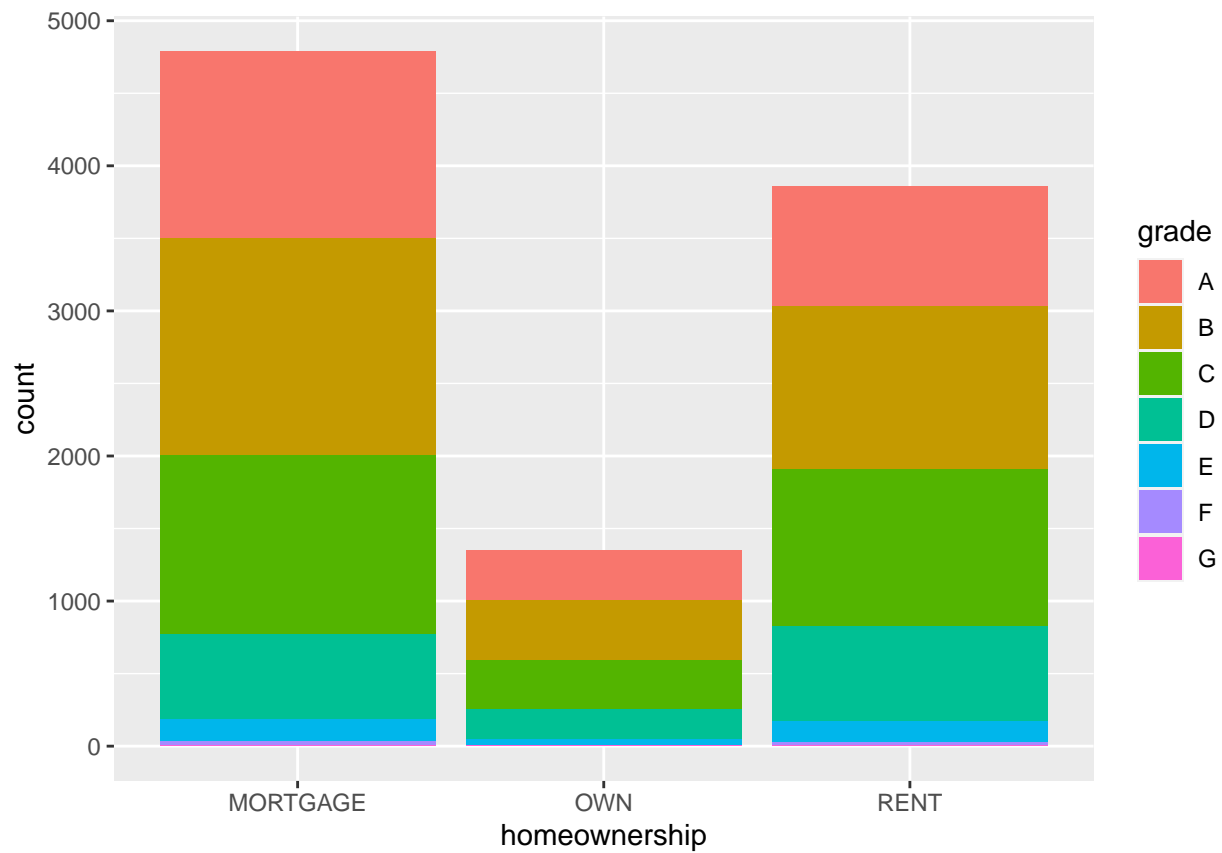
Question-37: Bar Plot Solutions:

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

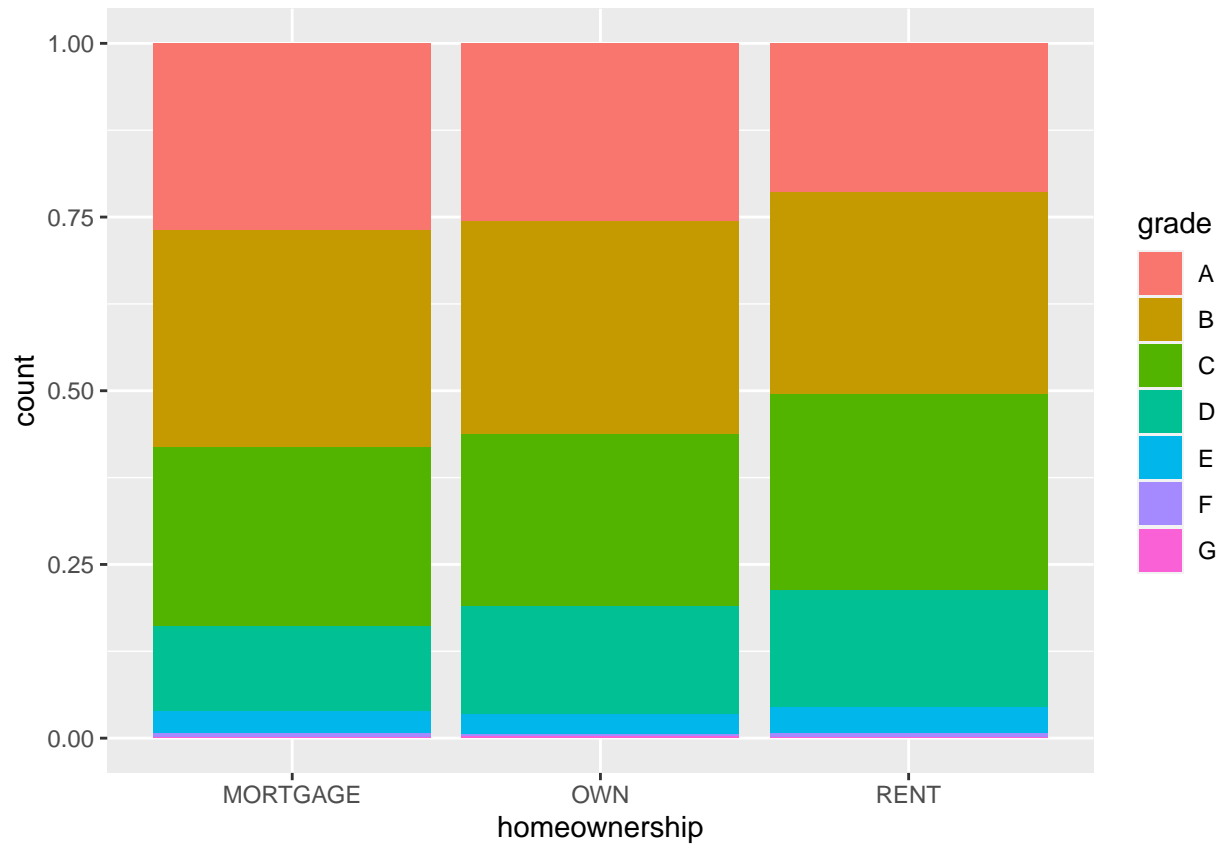


Question-38: Segmented Bar Plot Solutions:

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



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



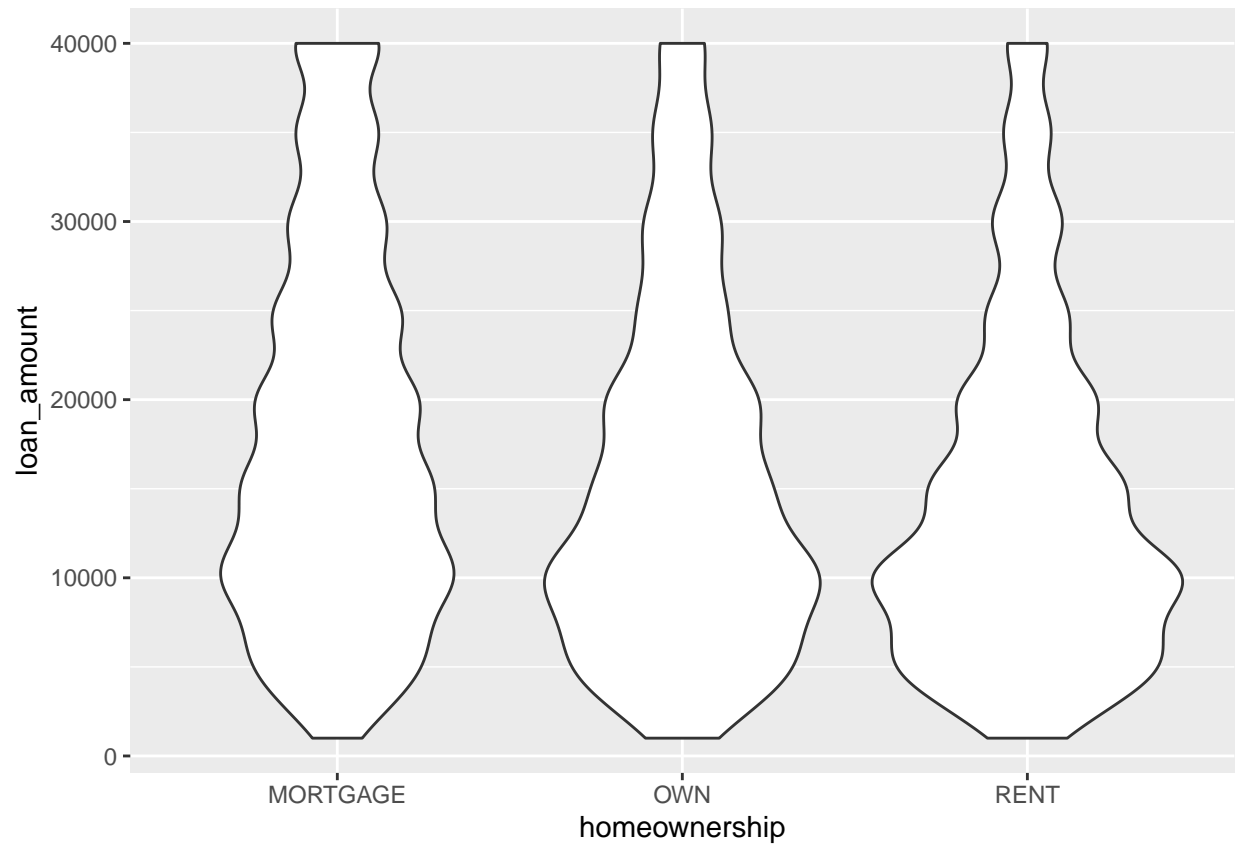
Question-39: Customizing bar plots Solutions:

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




Question-40: Violin Plots Solutions:

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



Question-41: Ridge Plots Solutions:

```
#install.packages("ggribes")  
library(ggribes)  
ggplot(loans, aes(x = loan_amount, y = grade, fill = grade, color = grade)) +  
  geom_density_ridges(alpha = 0.5)
```

Picking joint bandwidth of 2360

