

Exploratory data analysis

Visualising numerical data

The following content is based on Mine Çetinkaya-Rundel's excellent book Data Science in a Box

Terminology

Number of variables involved

- Univariate data analysis - distribution of **single** variable
- Bivariate data analysis - relationship between **two** variables
- Multivariate data analysis - relationship between **many** variables at once, usually focusing on the relationship between two while conditioning for others

Types of variables

Numerical variables

- **Numerical variables** can be classified as **continuous** or **discrete**
- Based on whether or not the variable can take on an infinite number of values (continuous) or only non-negative whole numbers (discrete), respectively.

Categorical variables

- If the variable is **categorical**, we can determine if it is **nominal** or **ordinal**, based on whether or not the levels have a natural ordering.

Data

Data: Lending Club

- Thousands of loans made through the Lending Club, which is a platform that allows individuals to lend to other individuals
- Not all loans are created equal -- ease of getting a loan depends on (apparent) ability to pay back the loan
- Data includes loans *made*, these are not loan applications



Take a peek at data

```
library(openintro)
glimpse(loans_full_schema)
```

```
## Rows: 10,000
## Columns: 55
## $ emp_title           <chr> "global config engine...
## $ emp_length          <dbl> 3, 10, 3, 1, 10, NA, ...
## $ state               <fct> NJ, HI, WI, PA, CA, K...
## $ homeownership       <fct> MORTGAGE, RENT, RENT,...
## $ annual_income        <dbl> 90000, 40000, 40000, ...
## $ verified_income      <fct> Verified, Not Verifie...
## $ debt_to_income       <dbl> 18.01, 5.04, 21.15, 1...
## $ annual_income_joint  <dbl> NA, NA, NA, NA, 57000...
## $ verification_income_joint <fct> , , , , Verified, , N...
## $ debt_to_income_joint <dbl> NA, NA, NA, NA, 37.66...
## $ delinq_2y            <int> 0, 0, 0, 0, 0, 1, 0, ...
## $ months_since_last_delinq <int> 38, NA, 28, NA, NA, 3...
## $ earliest_credit_line <dbl> 2001, 1996, 2006, 200...
## $ inquiries_last_12m   <int> 6, 1, 4, 0, 7, 6, 1, ...
## $ total_credit_lines   <int> 28, 30, 31, 4, 22, 32...
## $ open_credit_lines    <int> 10, 14, 10, 4, 16, 12...
...

```

Selected variables

```
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, ...  
## $ interest_rate  <dbl> 14.07, 12.61, 17.09, 6.72, 14.07, 6.72,...  
## $ term           <dbl> 60, 36, 36, 36, 36, 36, 60, 60, 36, 36,...  
## $ grade          <ord> C, C, D, A, C, A, C, B, C, A, C, B, C, ...  
## $ state          <fct> NJ, HI, WI, PA, CA, KY, MI, AZ, NV, IL,...  
## $ annual_income  <dbl> 90000, 40000, 40000, 30000, 35000, 3400...  
## $ homeownership  <fct> MORTGAGE, RENT, RENT, RENT, RENT, OWN, ...  
## $ debt_to_income <dbl> 18.01, 5.04, 21.15, 10.16, 57.96, 6.46,...
```


Selected variables

variable	description
loan_amount	Amount of the loan received, in US dollars
interest_rate	Interest rate on the loan, in an annual percentage
term	The length of the loan, which is always set as a whole number of months
grade	Loan grade, which takes a values A through G and represents the quality of the loan and its likelihood of being repaid
state	US state where the borrower resides
annual_income	Borrower's annual income, including any second income, in US dollars
homeownership	Indicates whether the person owns, owns but has a mortgage, or rents
debt_to_income	Debt-to-income ratio

Variable types

variable	type
loan_amount	numerical, continuous
interest_rate	numerical, continuous
term	numerical, discrete
grade	categorical, ordinal
state	categorical, not ordinal
annual_income	numerical, continuous
homeownership	categorical, not ordinal
debt_to_income	numerical, continuous

Visualizing numerical data

Describing shapes of numerical distributions

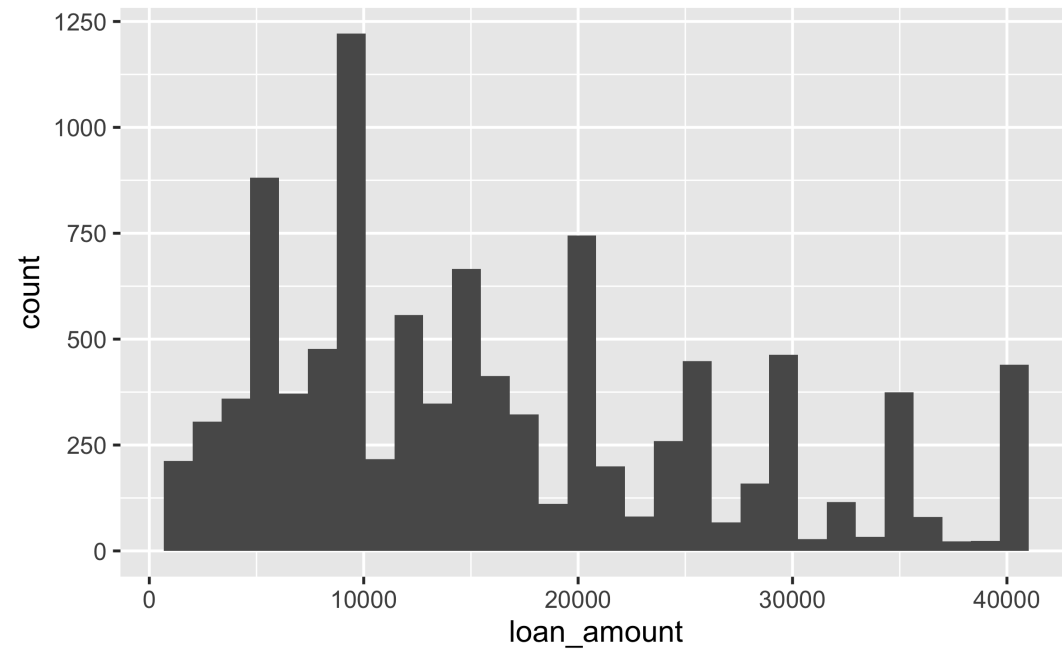
- shape:
 - skewness: right-skewed, left-skewed, symmetric (skew is to the side of the longer tail)
 - modality: unimodal, bimodal, multimodal, uniform
- center: mean (`mean`), median (`median`), mode (not always useful)
- spread: range (`range`), standard deviation (`sd`), inter-quartile range (`IQR`)
- unusual observations

Histogram

Histogram

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

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



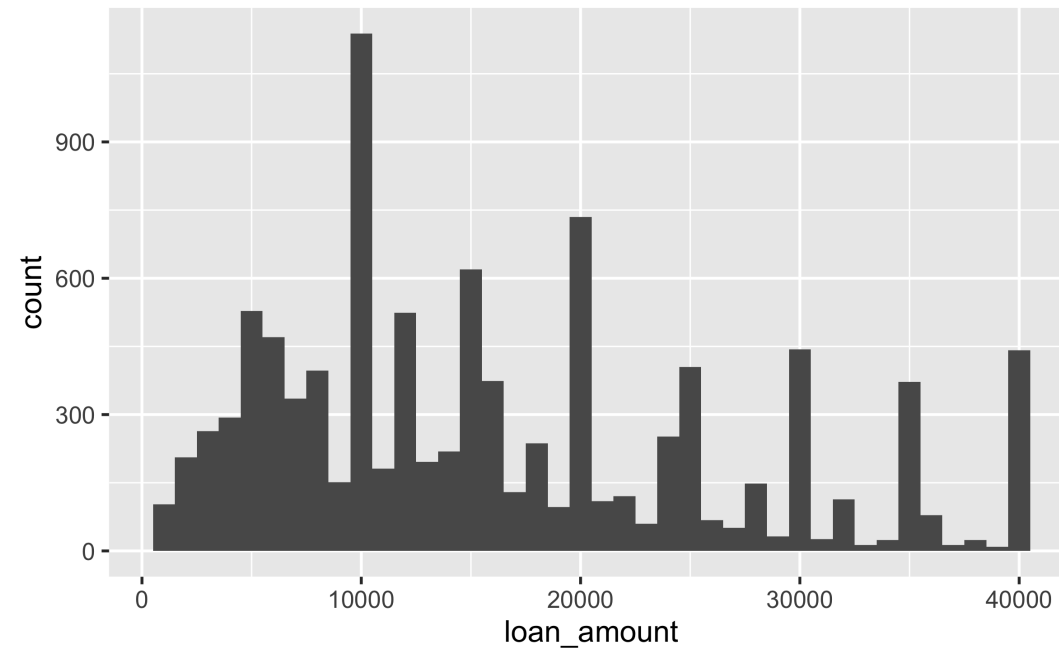
Histograms and binwidth

binwidth = 1000

binwidth = 5000

binwidth = 20000

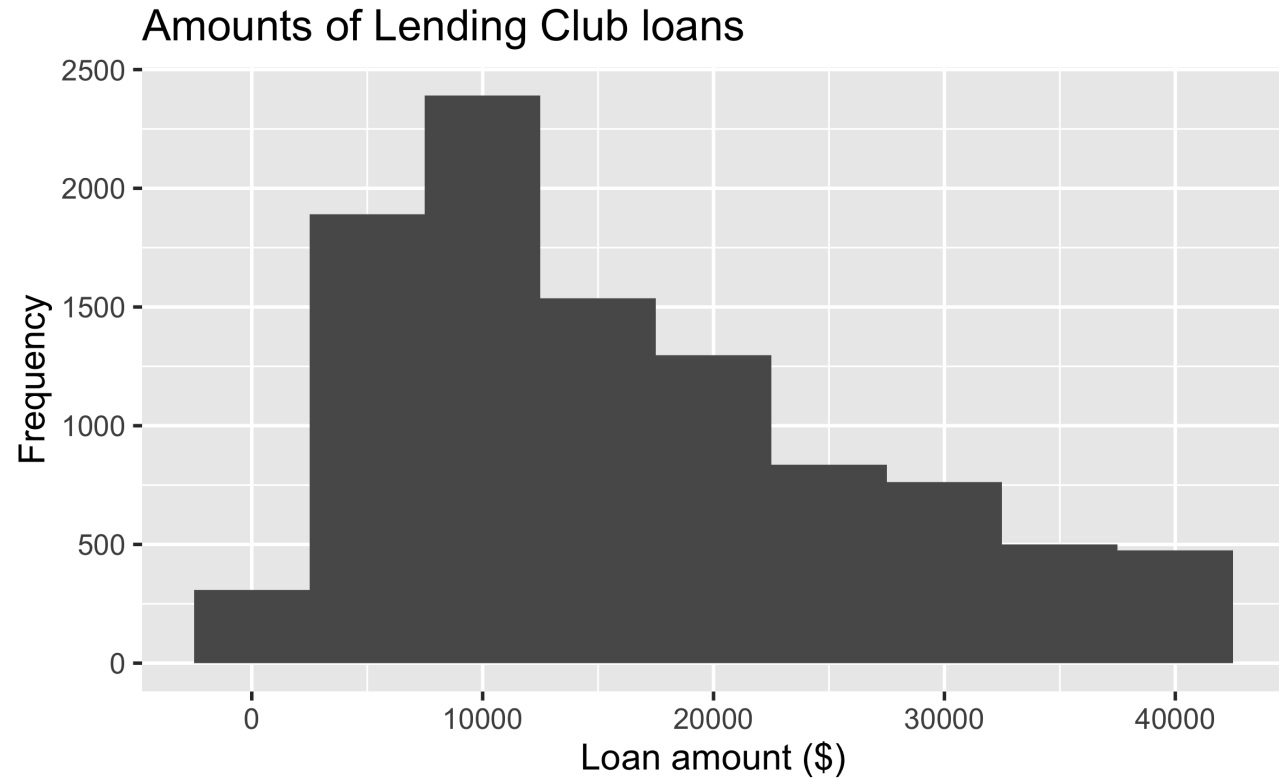
```
ggplot(loans, aes(x = loan_amount)) +  
  geom_histogram(binwidth = 1000)
```



Customizing histograms

Plot

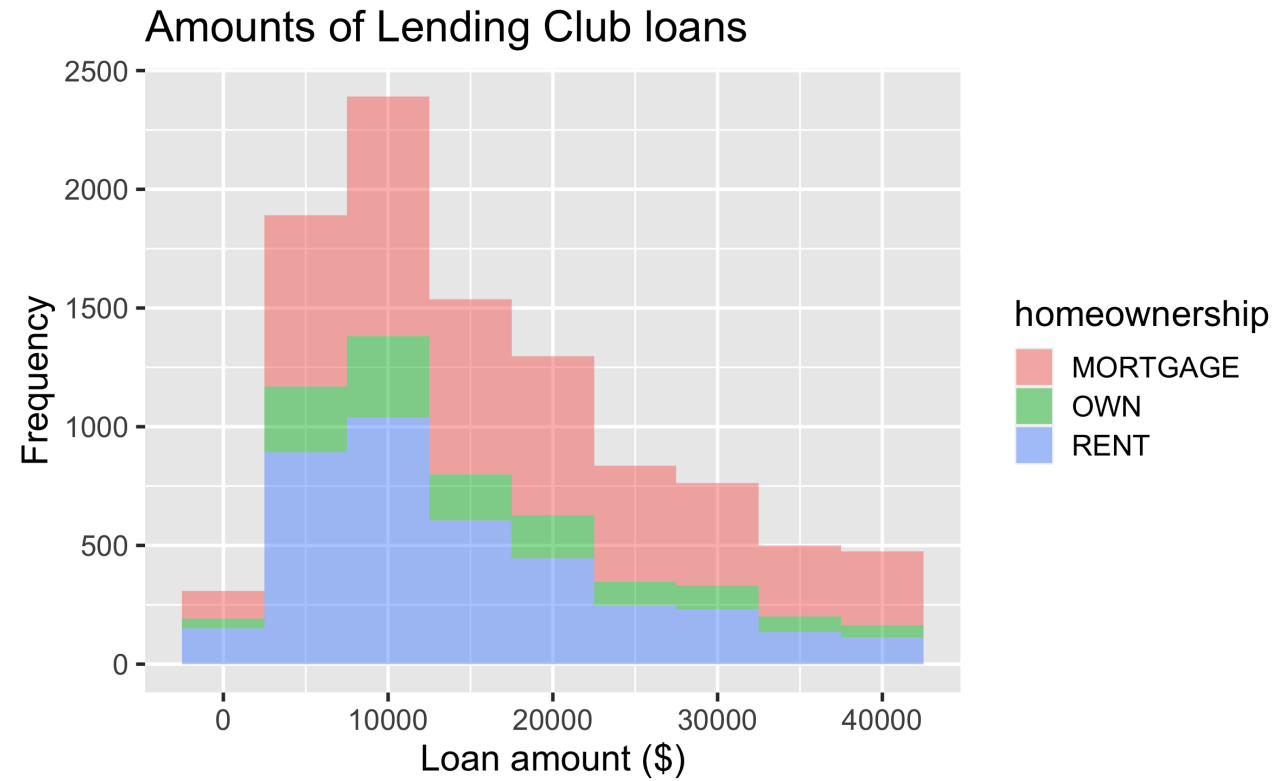
Code



Fill with a categorical variable

Plot

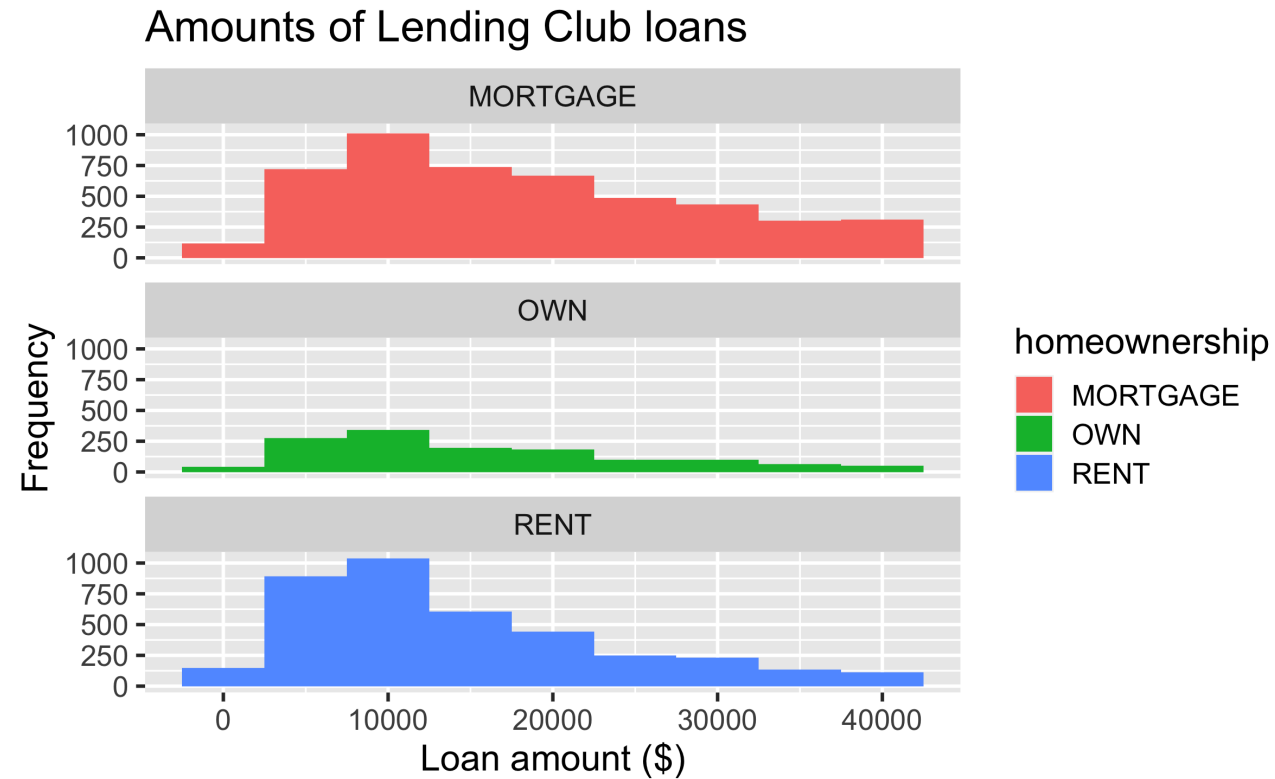
Code



Facet with a categorical variable

Plot

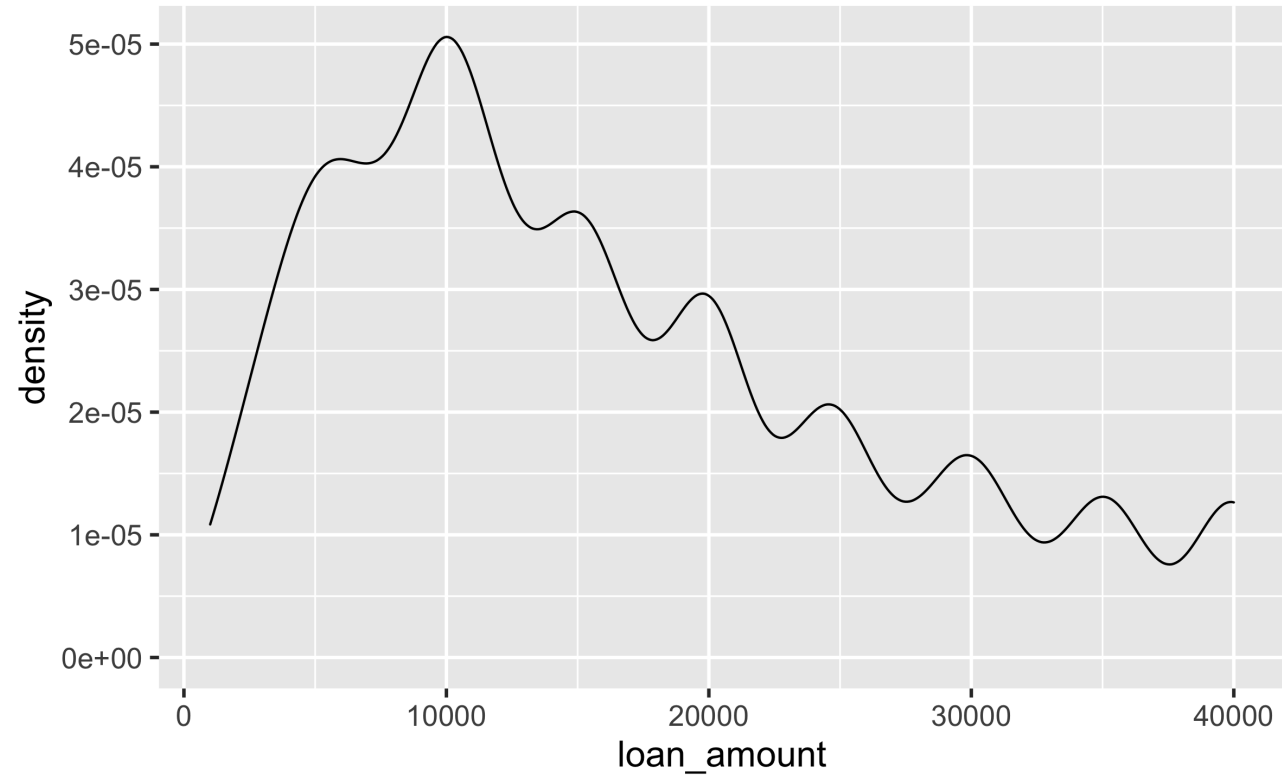
Code



Density plot

Density plot

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



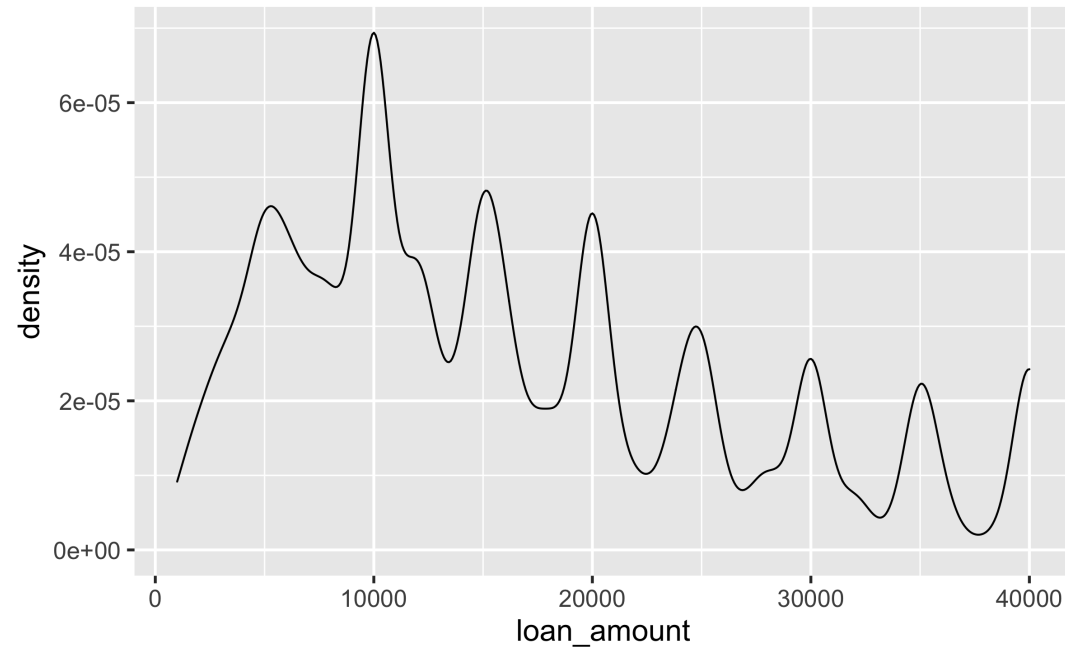
Density plots and adjusting bandwidth

adjust = 0.5

adjust = 1

adjust = 2

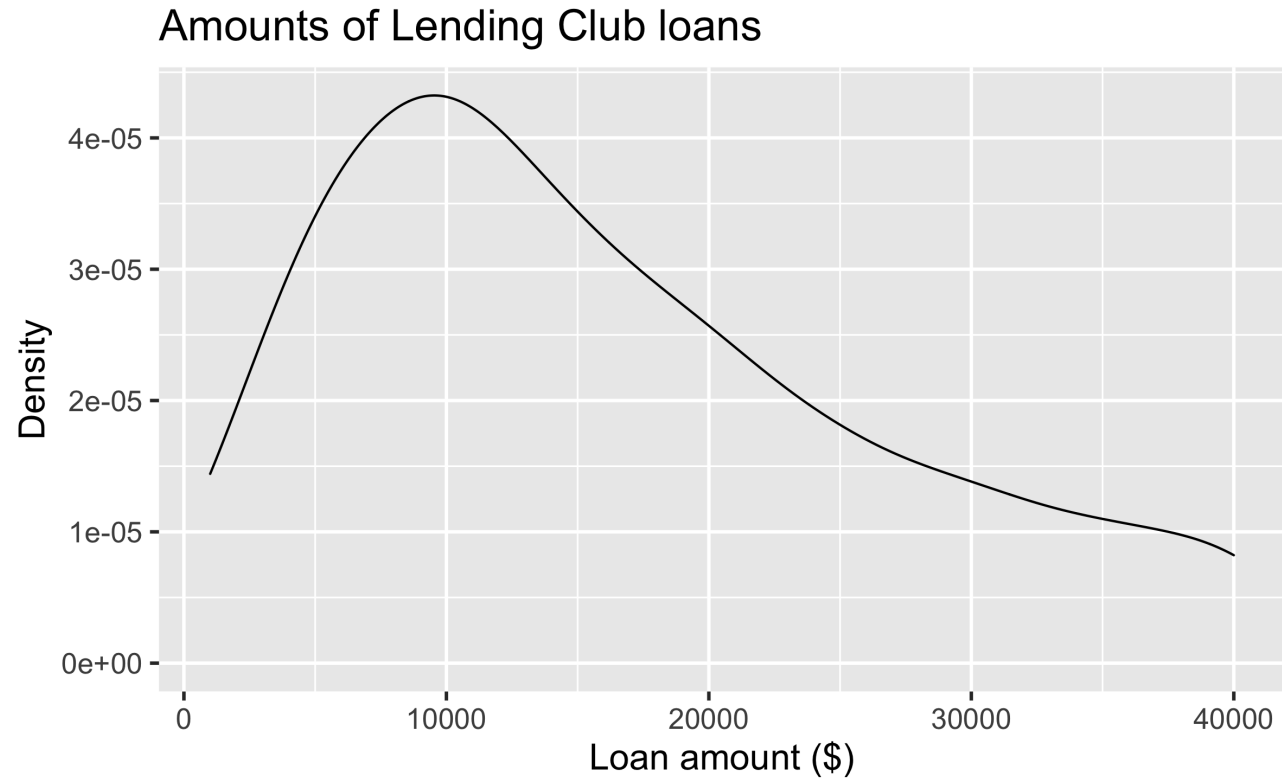
```
ggplot(loans, aes(x = loan_amount)) +  
  geom_density(adjust = 0.5)
```



Customizing density plots

Plot

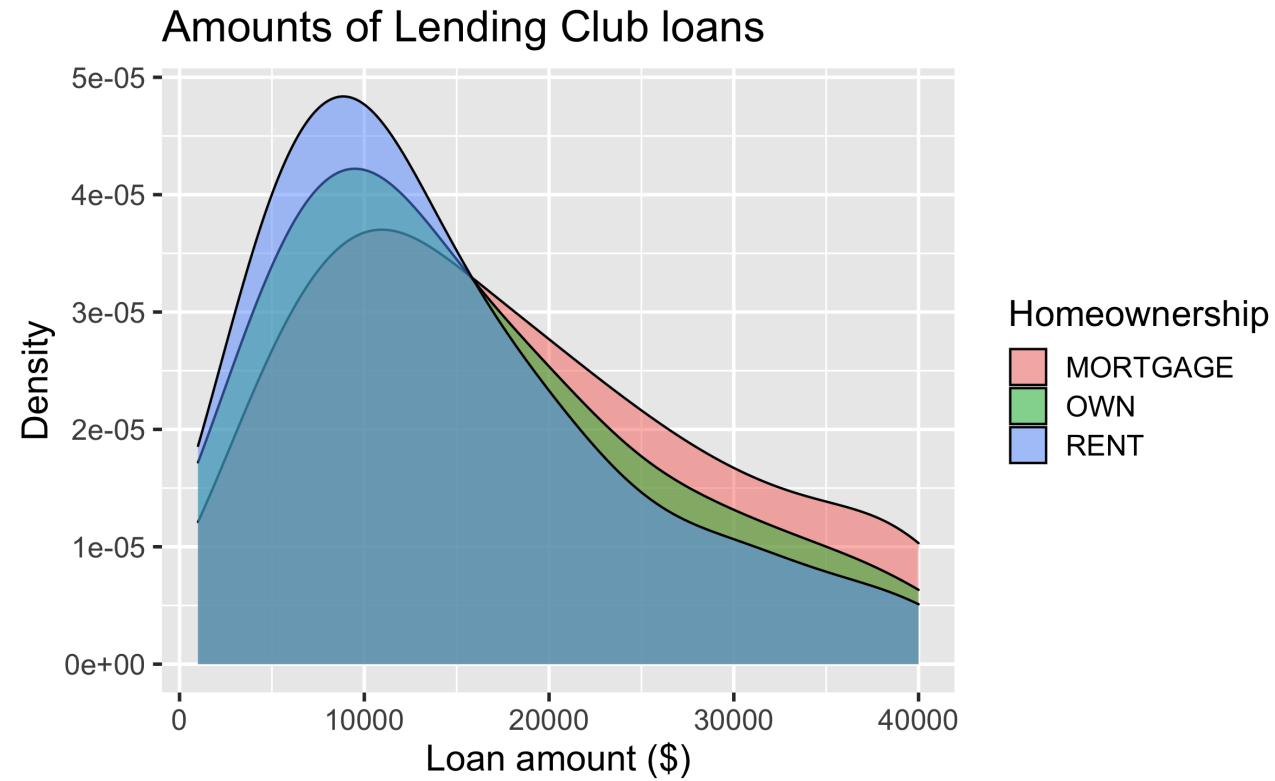
Code



Adding a categorical variable

Plot

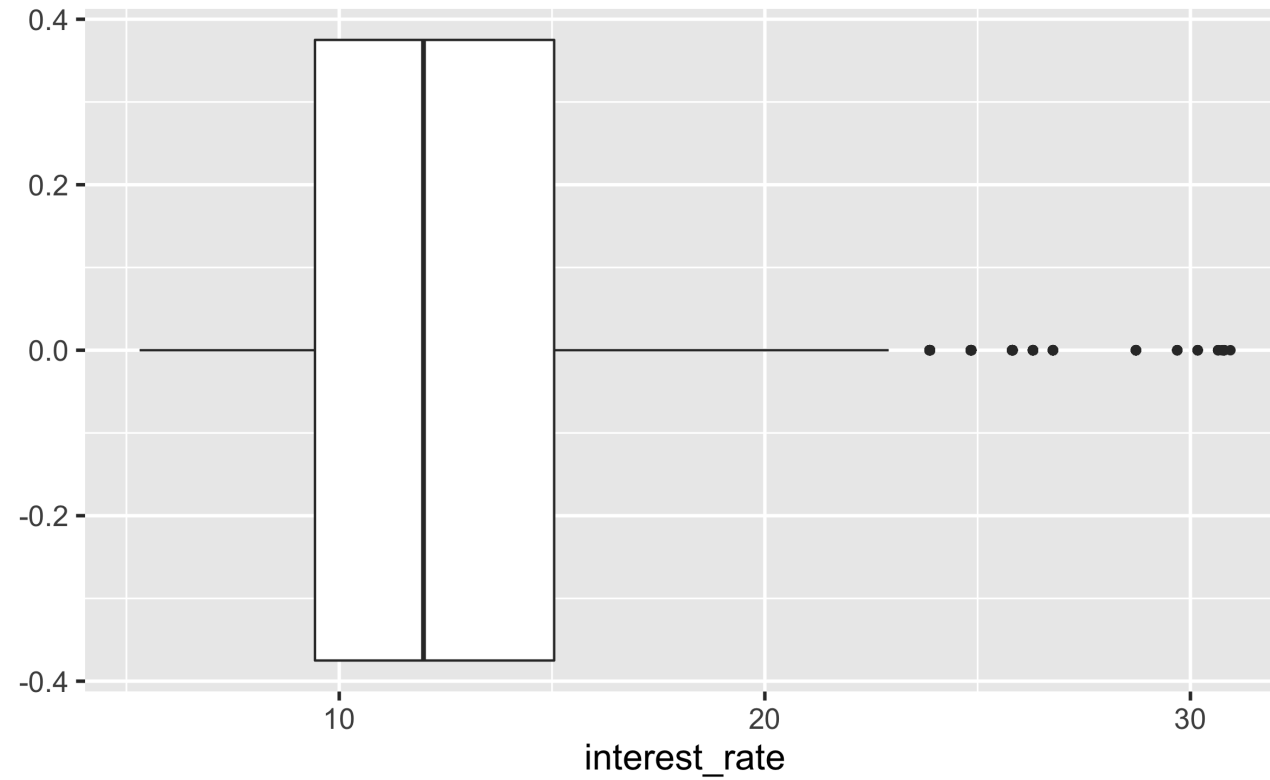
Code



Box plot

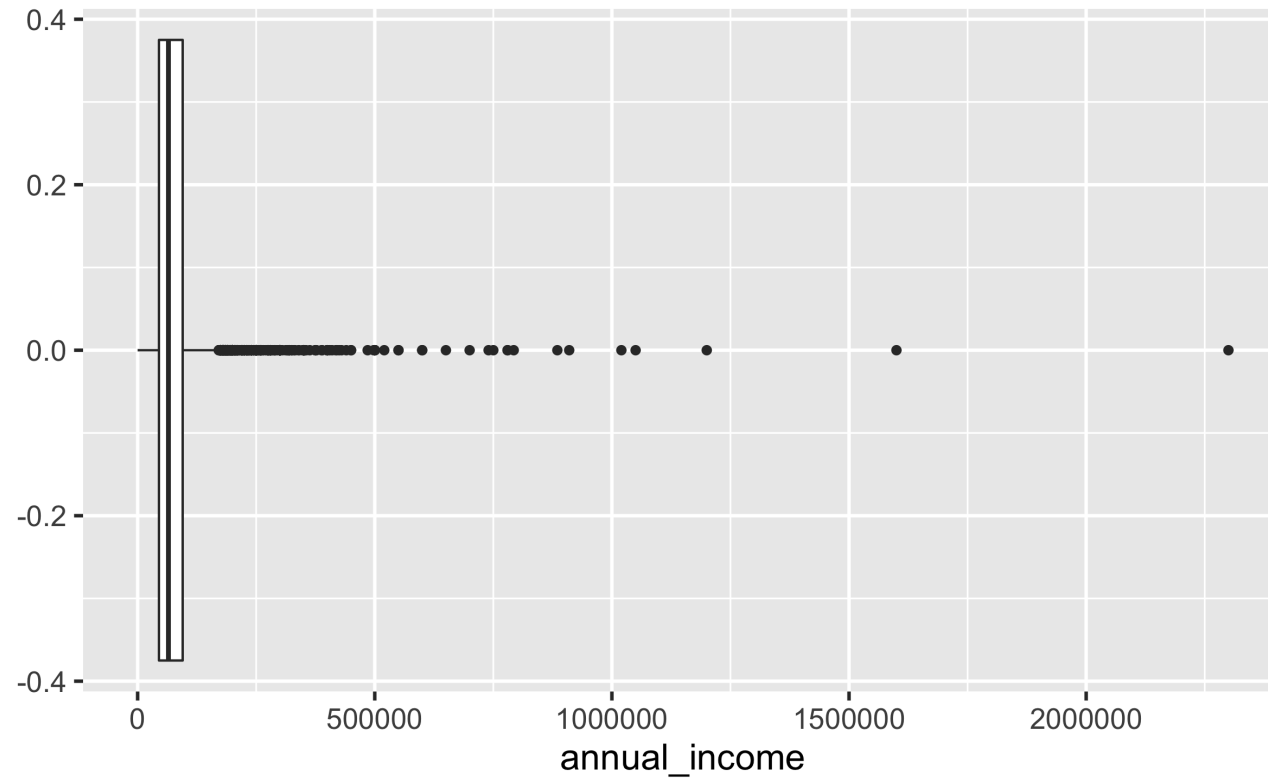
Box plot

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



Box plot and outliers

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

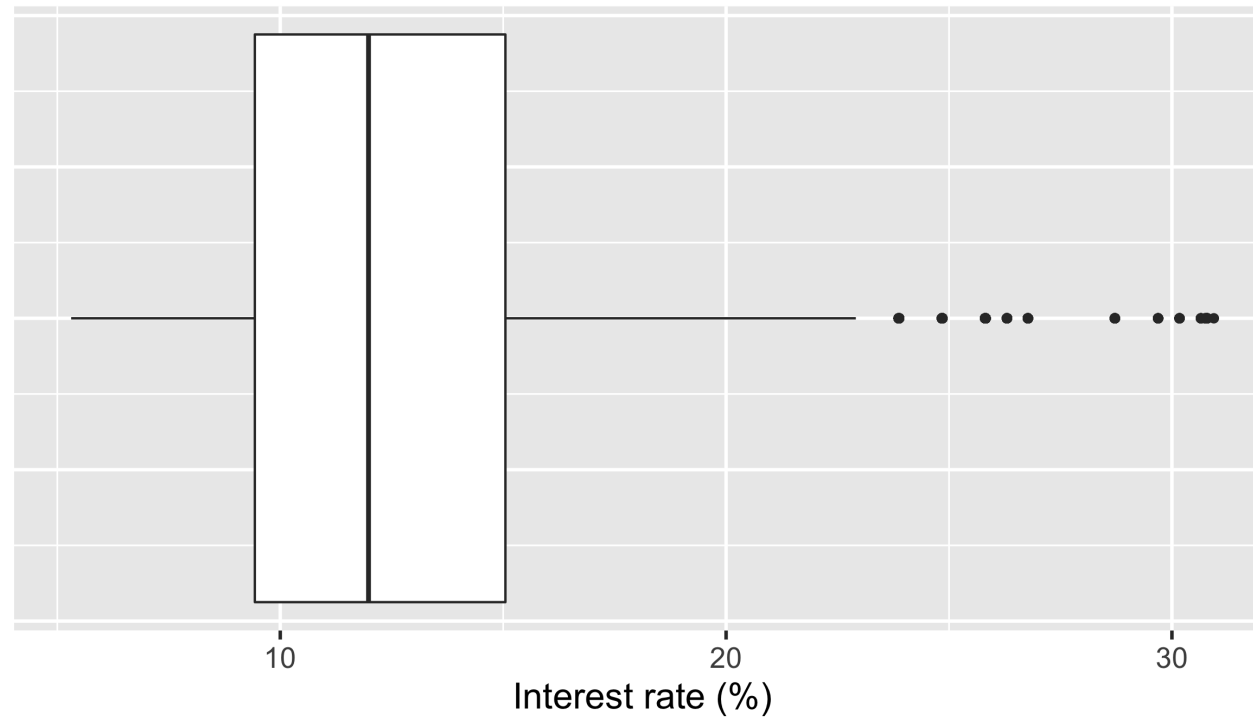


Customizing box plots

Plot

Code

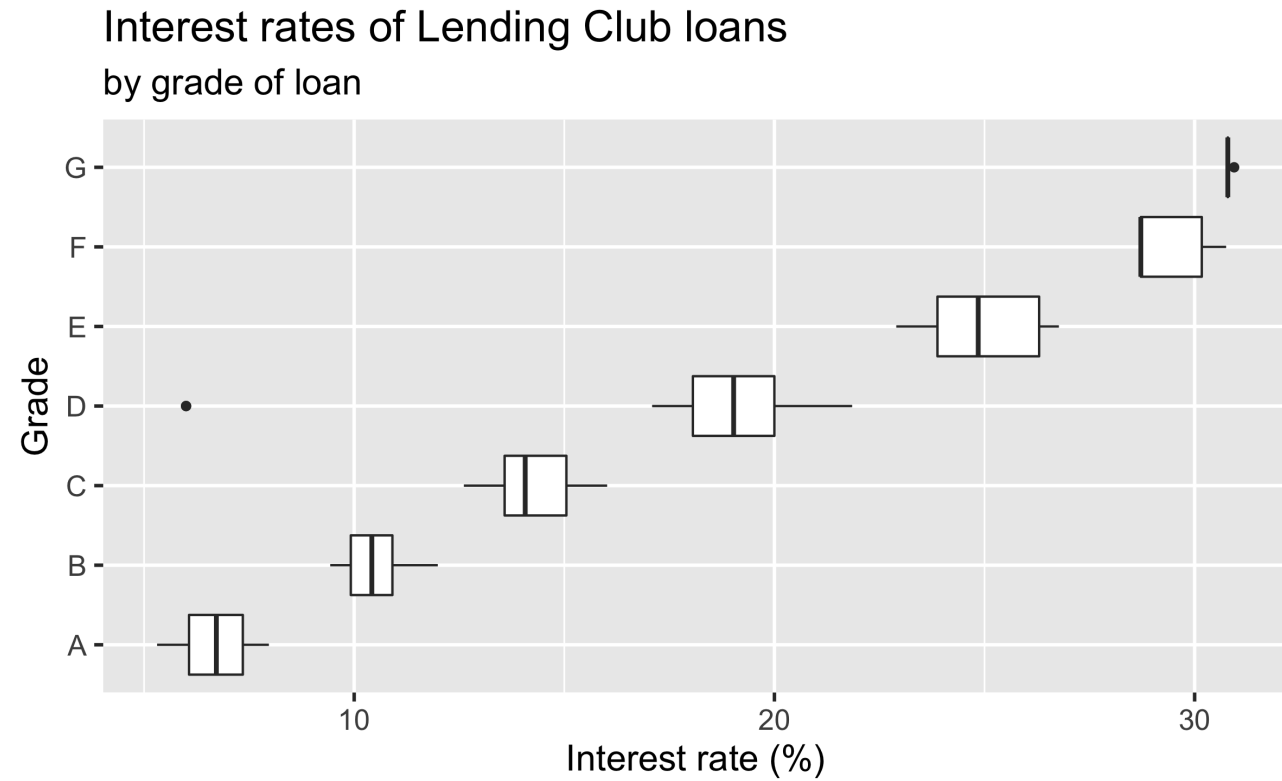
Interest rates of Lending Club loans



Adding a categorical variable

Plot

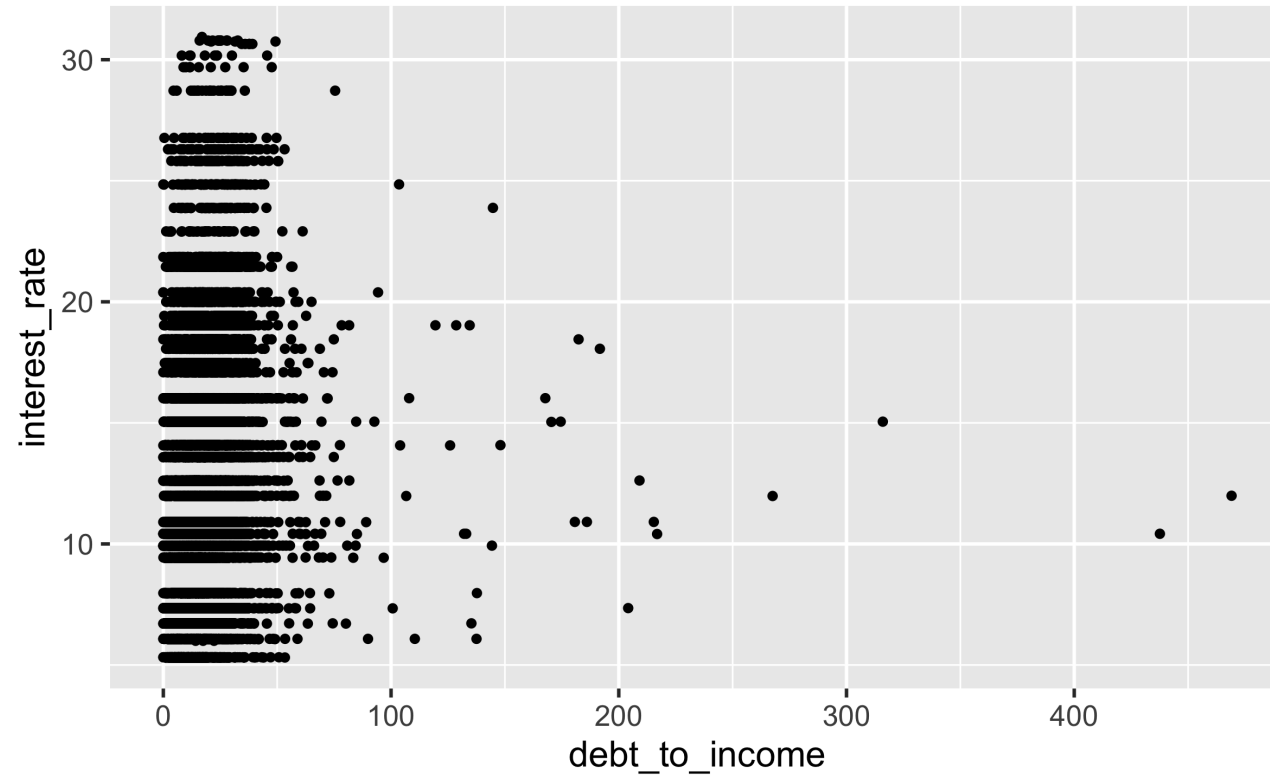
Code



Relationships numerical variables

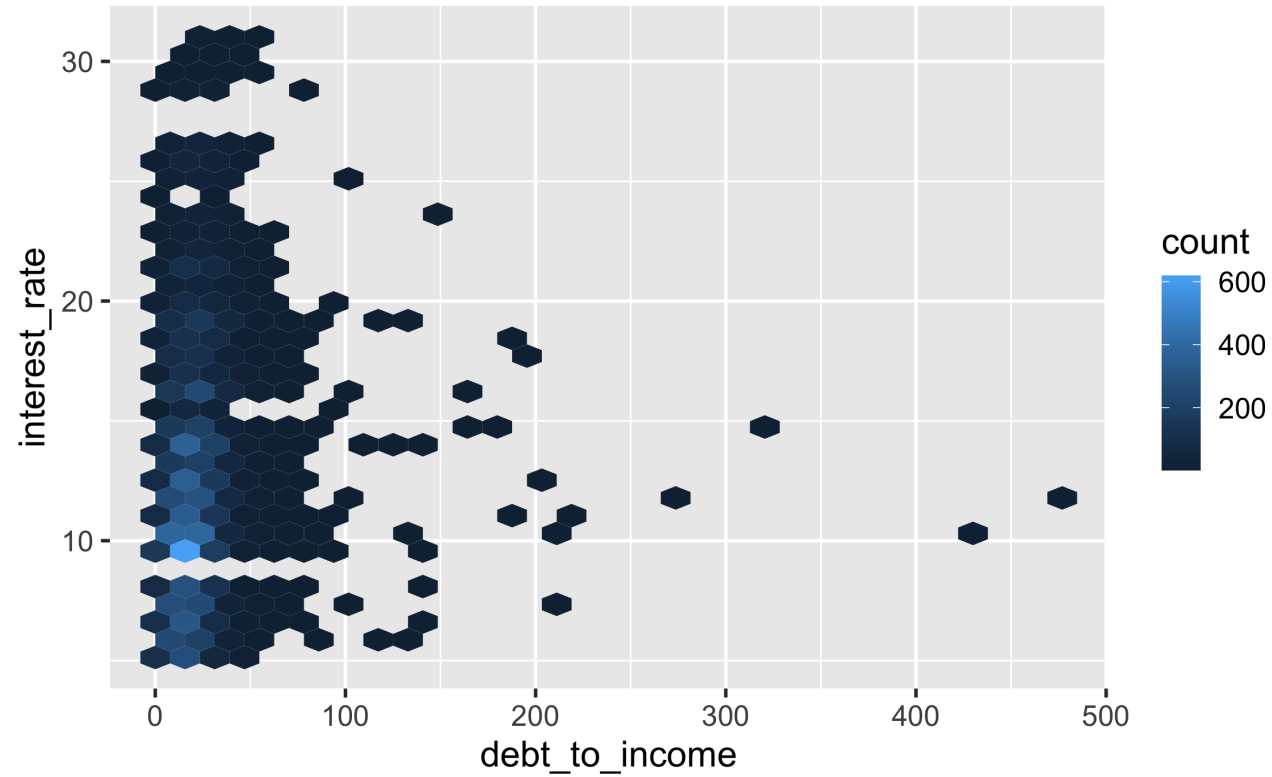
Scatterplot

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



Hex plot

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



Hex plot

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

