Lecture 11: Exploratory Data Analysis

STAT 450, Fall 2021

This lecture is based on portions of Chapter 7 from R for Data Science

Exploratory Data Analysis (EDA) is an iterative process:

1. Generate questions about your data.
2. Search for answers by visualizing, transforming, and modeling your data.
3. Use what you learn to refine your questions and generate new questions.

EDA was promoted by statistician John W. Tukey, who wrote a book on the topic in the 1970s. He believed that statistical methodology, at the time, had placed to much emphasis on confirmatory data analysis (hypothesis testing).

Some terminology:

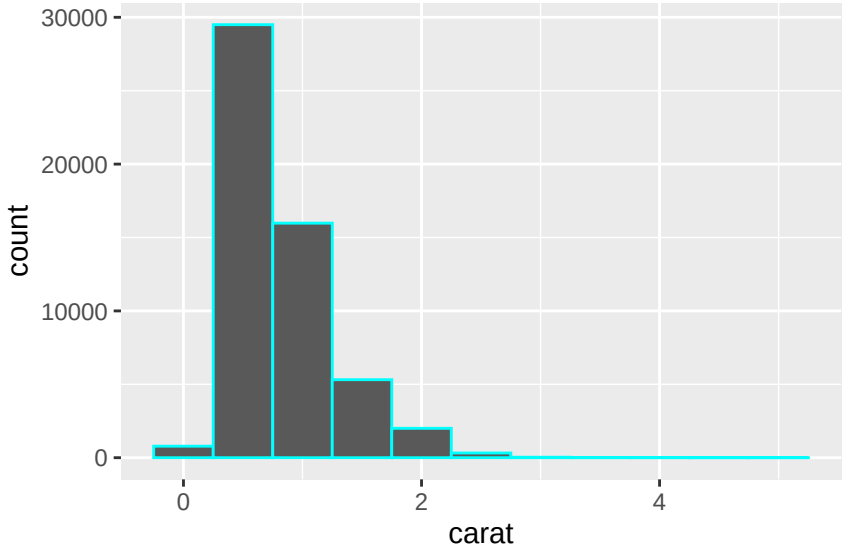
- **Variation** is the tendency of values of variable to change from measurement to measurement.
  - A histogram can be used to explore the variation of a numeric variable.
  - A bar plot can be used to explore the variation of a categorical variable.
- **Covariation** is the tendency of values of two or more variables to vary together in a related way.
  - Scatter plots can be used to explore covariation between two numeric variables.
  - Box plots can be used to explore covariation between a numeric and categorical variable.

Example: one numeric variable

Explore the distribution of variable `carat` (weight of diamond)[1]

```
library(tidyverse)
```

```
ggplot(data = diamonds, aes(x = carat)) +
  geom_histogram(binwidth = 0.5, color = "cyan")
```




```
diamonds %>%
  count(cut_width(carat, 0.5))
```

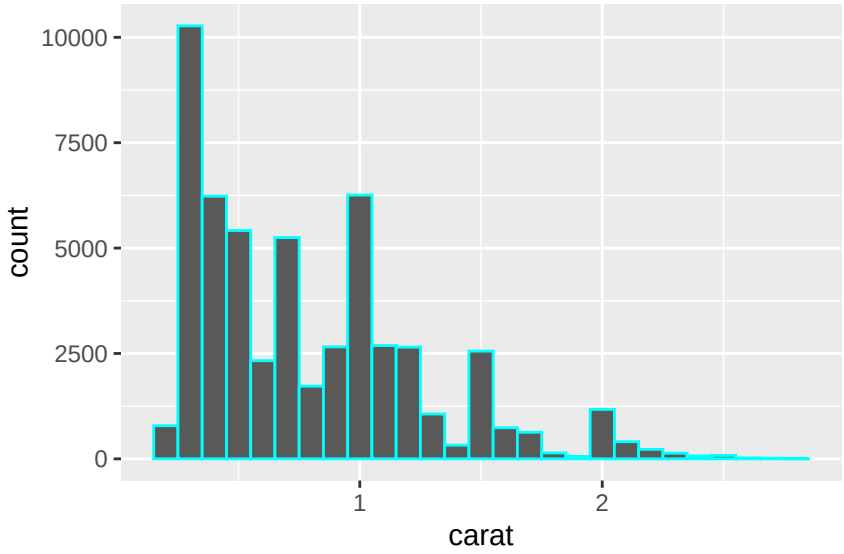
```
## # A tibble: 11 x 2
##    `cut_width(carat, 0.5)`     n
##    <fct>                   <int>
##  1 [-0.25,0.25]              785
##  2 (0.25,0.75]             29498
##  3 (0.75,1.25]             15977
##  4 (1.25,1.75]              5313
##  5 (1.75,2.25]              2002
##  6 (2.25,2.75]               322
##  7 (2.75,3.25]                32
##  8 (3.25,3.75]                 5
##  9 (3.75,4.25]                 4
## 10 (4.25,4.75]                 1
## 11 (4.75,5.25]                 1
```

[1] https://www.americangemsociety.org/page/diamondcarat

Histogram for diamonds with a size of less than three carats.

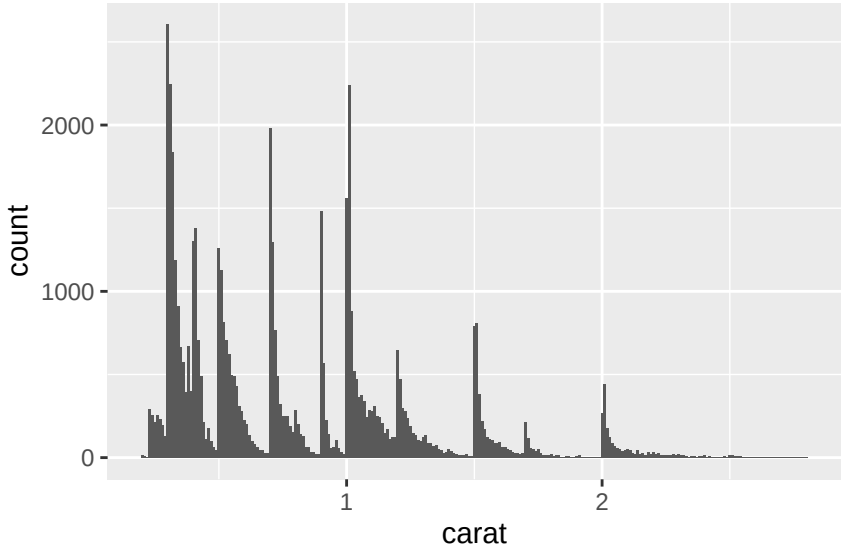
```
diamonds_small <- diamonds %>% filter(carat < 3)

ggplot(data = diamonds_small, aes(x = carat)) +
 geom_histogram(binwidth = 0.1, color = "cyan")
```



Using a smaller bin width:

```
ggplot(data = diamonds_small, aes(x = carat)) +
 geom_histogram(binwidth = 0.01)
```



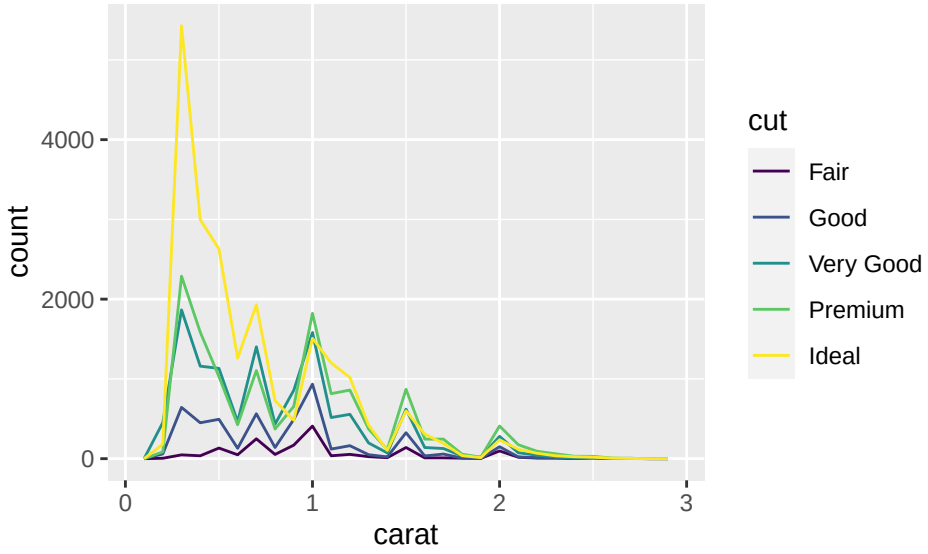
Exercise 1: Explore the distribution of `price`. Do you discover anything unusual or surprising? (Hint: Try adjusting the `bindwidth` of the histogram)

Exercise 2: How many diamonds are 0.99 carat? How many are 1 carat? What do you think is the cause of the difference?

Example: one categorical and one numeric variable

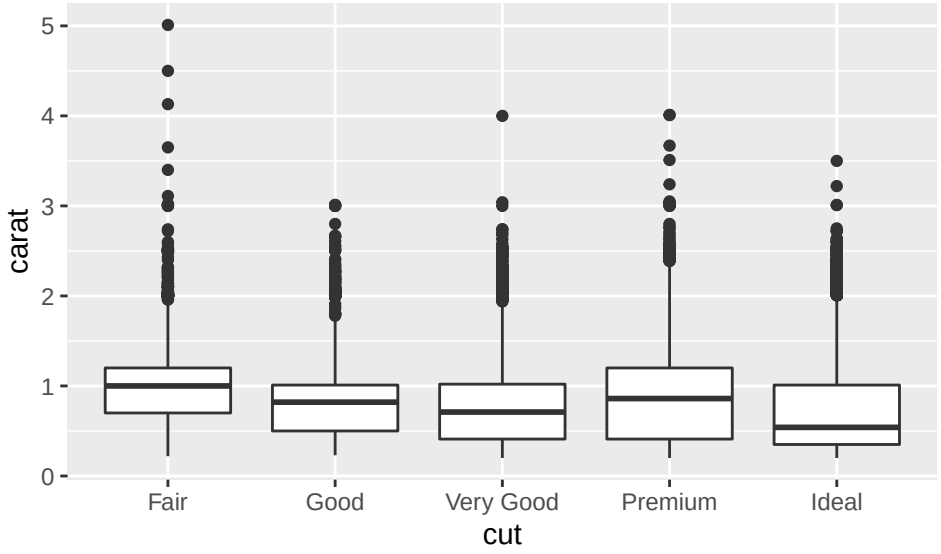
Use `geom_freqpoly()` to overlay multiple histograms on the same plot. `geom_freqpoly()` uses lines to display the counts, instead of bars.

```
ggplot(data = diamonds_small, aes(x = carat, color = cut)) +
 geom_freqpoly(binwidth = 0.1)
```



Box plots are also useful when exploring the relationship between a numeric and categorical variable.

```
ggplot(data = diamonds, aes(x = cut, y = carat)) +
  geom_boxplot()
```

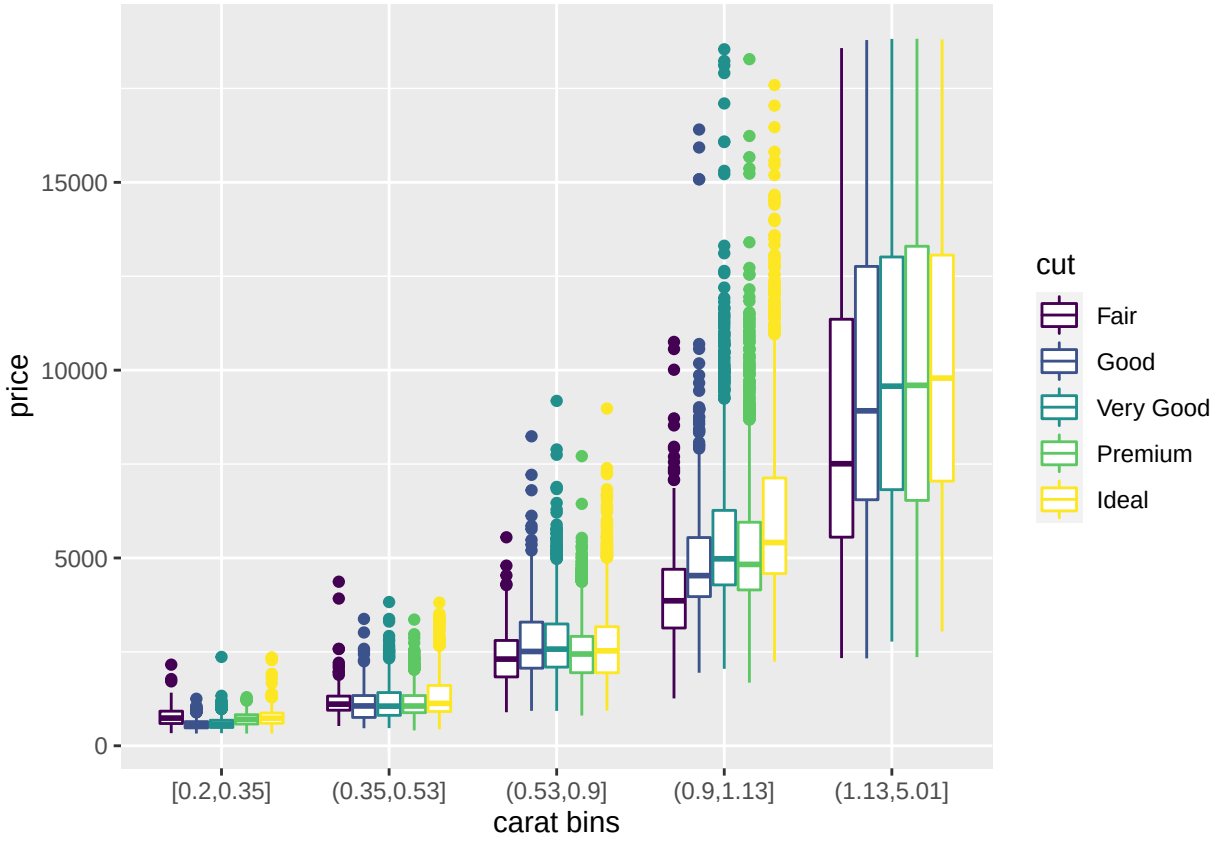


Exercise 3: Use the `dplyr` functions `group_by()` and `summarise()` to compute the median of `carat` for each category of `cut`. Also compute the median of `price` for each category of `cut`.

Exercise 4: [from Section 7.5] Visualize the combined distribution of `cut`, `carat`, and `price`.

Below we divide `carat` into 5 bins (intervals), and then use box plots to investigate the relationship between `cut` and `price` within each `carat` bin. Generally, we see that as `carat` increase, `price` increases (positive association). Additionally, we see that for diamonds within each `carat` bin (i.e., diamonds of similar weight), `price` tends to increase as the quality of the `cut` improves.

```
ggplot(data = diamonds, aes(x = cut_number(carat, 5), y = price, color = cut)) +
  geom_boxplot() + xlab("carat bins")
```



Type `help(cut_number)` into the console to learn more about this and related functions. When using `cut_number()` each bin has roughly the same number of observations:

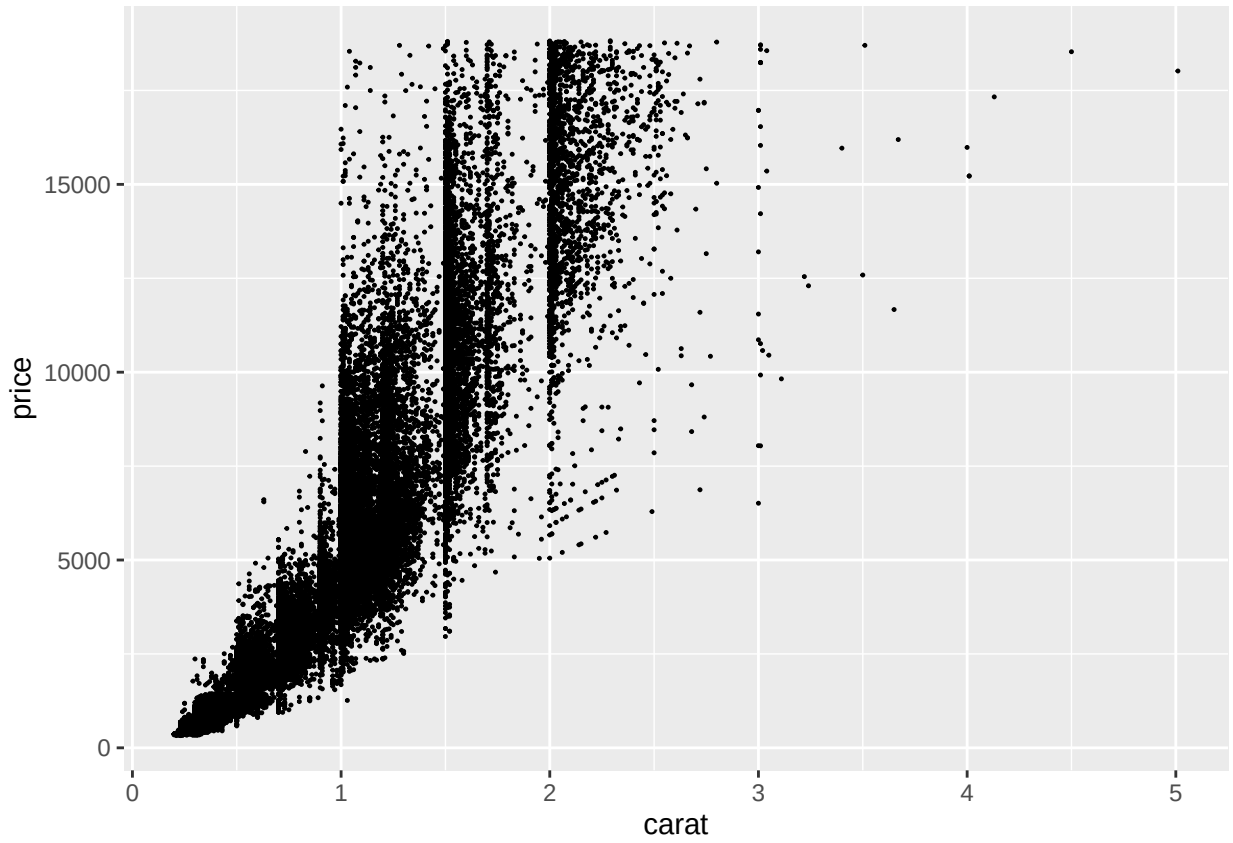
```
diamonds %>%
  count(cut_number(carat, 5))
```

```
## # A tibble: 5 x 2
##   `cut_number(carat, 5)`     n
##   <fct>                  <int>
## 1 [0.2,0.35]             11058
## 2 (0.35,0.53]            10527
## 3 (0.53,0.9]             12017
## 4 (0.9,1.13]              9651
## 5 (1.13,5.01]            10687
```

Example: two numeric variables

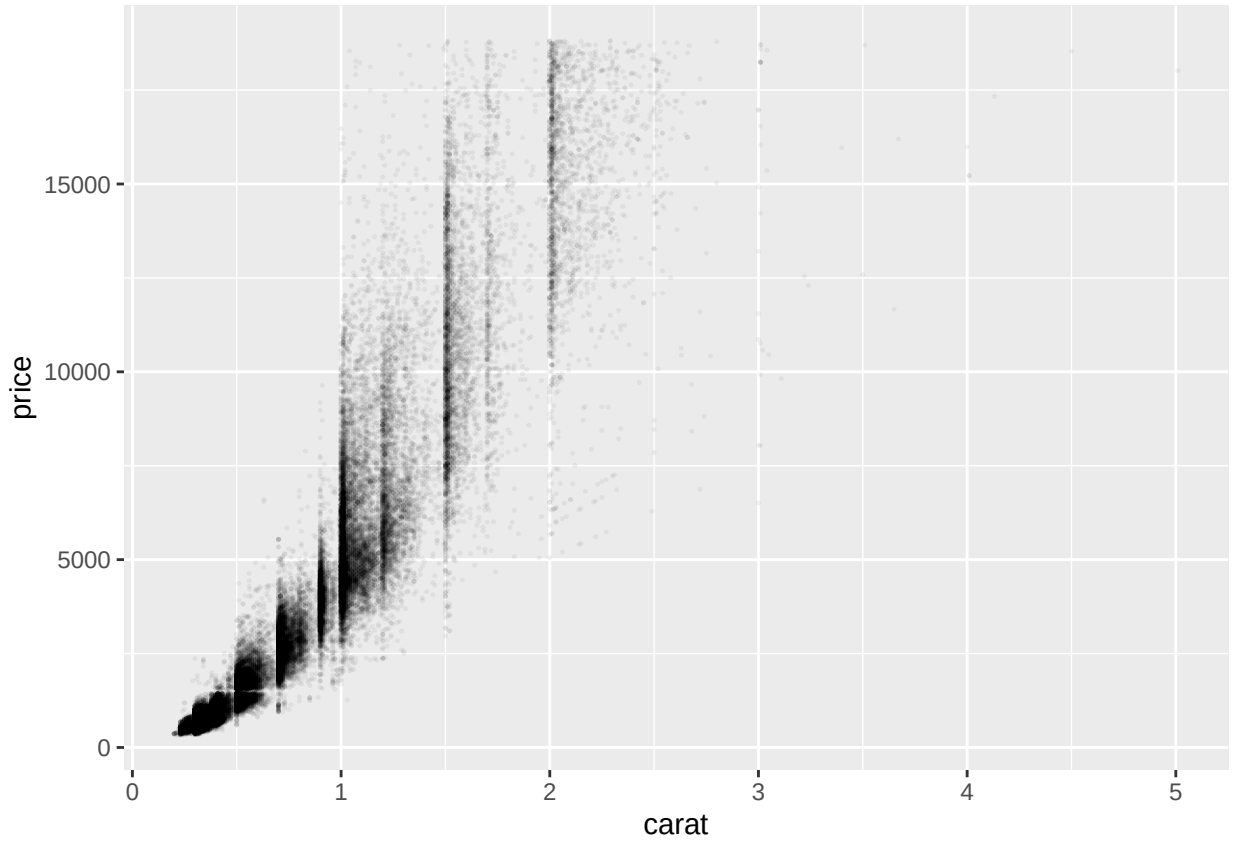
Explore the association (covariation) between `carat` and `price`. We see an increasing, exponential relationship.

```
ggplot(data = diamonds, aes(x = carat, y = price)) +
  geom_point(size = 0.2)
```



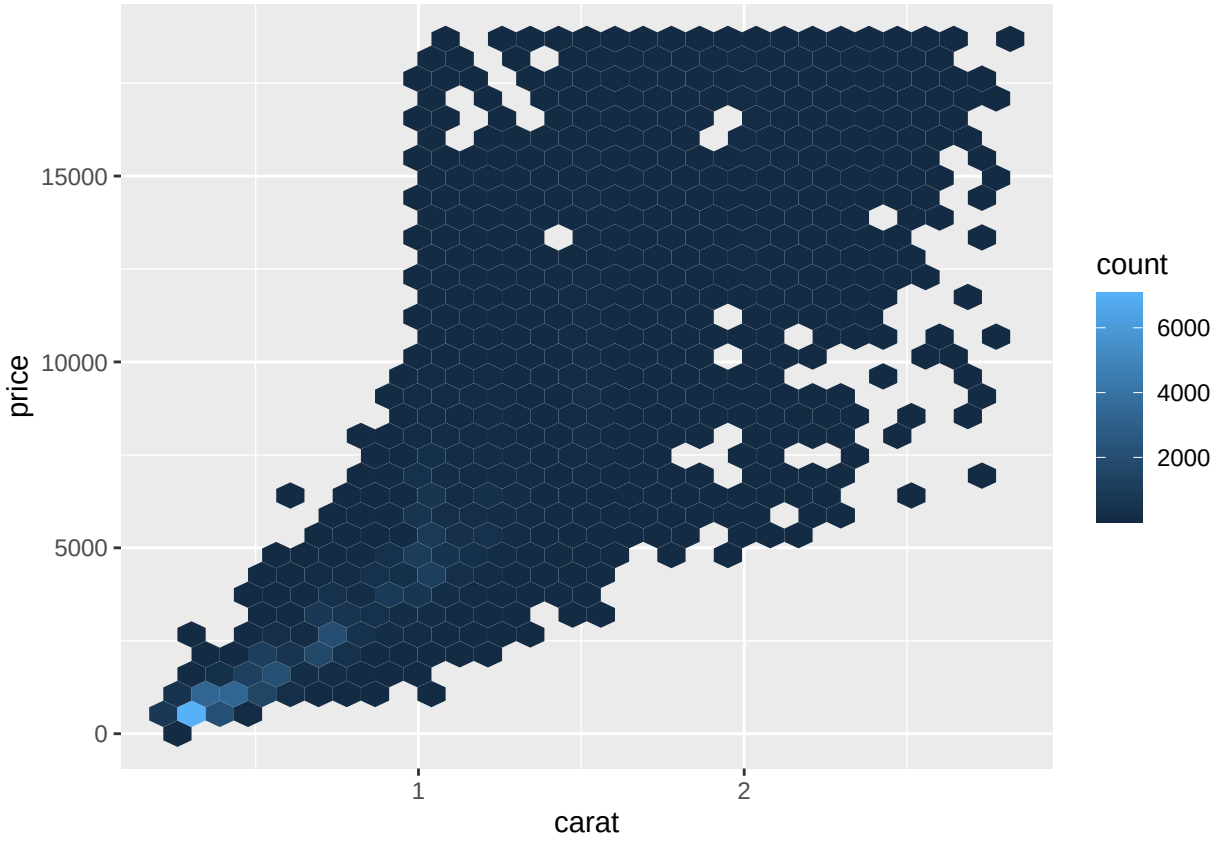
One problem with the previous scatter plots is that many of the points overlapped. One way to fix the problem is to use `alpha` to adjust the point transparency.

```
ggplot(data = diamonds, aes(x = carat, y = price)) +
  geom_point(size = 0.2, alpha = 0.05)
```



`geom_hex()` divides the plane into hexagonal bins and then uses a fill color to display how many points fall into each bin.

```
library(hexbin)
ggplot(data = diamonds_small, aes(x = carat, y = price)) +
  geom_hex()
```

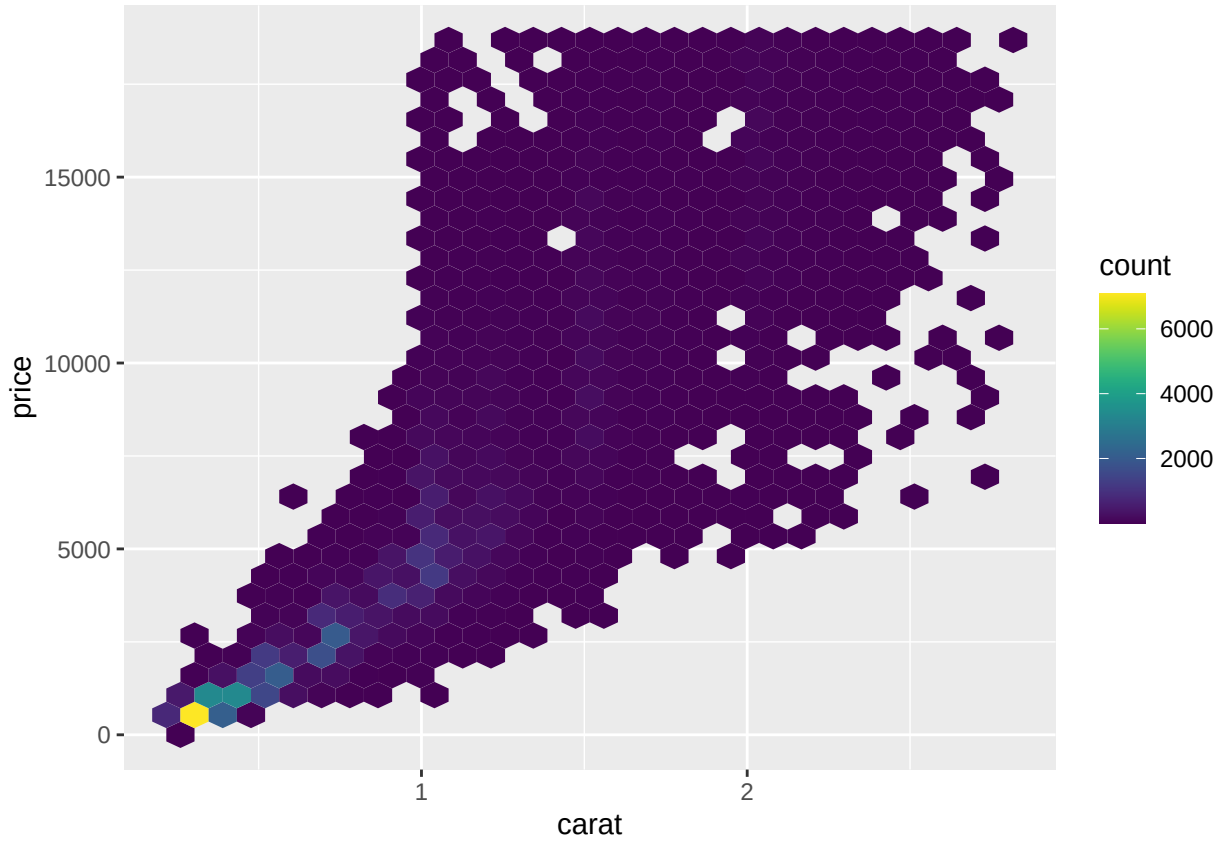


This plot uses the `viridis` color palette for the counts:

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
ggplot(data = diamonds_small, aes(x = carat, y = price)) +
  geom_hex() +
  scale_fill_viridis()
```



Counts displayed on logarithmic scale:

```
ggplot(data = diamonds_small, aes(x = carat, y = price)) +
  geom_hex() +
  scale_fill_viridis(trans = "log", breaks = c(0, 10, 100, 1000, 3000))
```