Data Management With R: Exploratory Data Analysis

Matthias Haber

25 September 2017

Prerequisites

Packages

```
# install.packages("tidyverse")
library(tidyverse)
```

```
# install.packages("hexbin")
library(hexbin)
```

Data

We'll continue to work with the flights data and will also use the diamonds and mpg datasets. Make sure that you can access all of them.

```
# install.packages("nycflights13")
library(nycflights13)
str(flights)
str(diamonds)
str(mpg)
```

Last week's homework

Homework Question 1

```
# Which destination has the most carriers?
flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay)) %>%
  group_by(dest) %>%
  summarise(carriers = n_distinct(carrier)) %>%
  arrange(desc(carriers)) %>%
  head(n = 4)
```

```
## # A tibble: 4 x 2
##    dest carriers
##   <chr>    <int>
## 1   ATL        7
## 2   BOS        7
## 3   CLT        7
## 4   ORD        7
```

Homework Question 2

```
# Which destination has the largest spread (standard
# deviation) in terms of distance that planes traveled
# to get to it.
flights %>%
  group_by(dest) %>%
  summarise(spread = sd(distance)) %>%
  arrange(desc(spread)) %>%
  head(n = 1)
```

```
## # A tibble: 1 x 2
##    dest   spread
##   <chr>    <dbl>
## 1   EGE 10.54907
```

Homework Question 3

```
# What is the average (mean) departure delay of United
# Airlines? Round to the nearest integer.
flights %>%
  filter(carrier == "UA") %>%
  summarise(delay = round(mean(dep_delay, na.rm =TRUE)))
```

```
## # A tibble: 1 x 1
##   delay
##   <dbl>
## 1    12
```

Homework Question 4

```
# How many flights were delayed by at least an hour,
# but made up over 45 minutes in flight?
flights %>%
  filter(dep_delay >= 60, dep_delay-arr_delay > 45) %>%
  n_distinct()
```

```
## [1] 245
```

Homework Question 5

```
# At what time (minutes after midnight) did the first
# flight leave on September 18, 2013?
flights %>%
  filter(month == 9, day == 18) %>%
  mutate(dep_time2 = dep_time %/% 100 * 60 +
           dep_time %% 100) %>%
  select(dep_time2) %>%
  arrange(dep_time2) %>%
  head(n=1)
```

```
## # A tibble: 1 x 1
##   dep_time2
##       <dbl>
## 1       290
```

Homework Question 6

```
# How many flights left before 5am in September (including
# # of delayed flights from the previous day)?
flights %>%
  filter(!is.na(dep_delay)) %>%
  filter(month == 9, dep_time < 500) %>%
  n_distinct()
```

```
## [1] 66
```

Homework Question 7

```
# Which departure airport (FAA airport code) has the
# highest number of departure delays that are longer
# than 2 hours?
flights %>%
  filter(dep_delay > 120) %>%
  group_by(origin) %>%
  summarise(delay = n())
```

```
## # A tibble: 3 x 2
##   origin delay
##    <chr> <int>
## 1    EWR  3884
## 2    JFK  3048
## 3    LGA  2791
```

Homework Question 8

```
# Which departure airport (FAA airport code) has
# the longest mean departure delay in September?
flights %>%
  filter(month == 9) %>%
  group_by(origin) %>%
  summarise(delay = mean(dep_delay, na.rm =TRUE))
```

```
## # A tibble: 3 x 2
##   origin    delay
##    <chr>    <dbl>
## 1    EWR 7.290954
## 2    JFK 6.635776
## 3    LGA 6.207439
```

Homework Question 9

```
# Which carrier (two letter abbreviation) has the
# shortest average (mean) departure delay when you
# take into account the distance that carrier traveled?
flights %>%
  group_by(carrier) %>%
  mutate(delay = dep_delay / distance) %>%
  summarise(delay = mean(delay, na.rm =T)) %>%
  arrange(delay) %>%
  head(n=1)
```

```
## # A tibble: 1 x 2
##   carrier         delay
##     <chr>         <dbl>
## 1      HA 0.0009834607
```

Homework Question 10

```
# Which plane (tailnum) has the worst on-time
# record in terms of arrival delay?
flights %>%
  group_by(tailnum) %>%
  summarise(delay = mean(arr_delay, na.rm =T)) %>%
  arrange(desc(delay)) %>%
  head(n=1)
```

```
## # A tibble: 1 x 2
##   tailnum delay
##     <chr> <dbl>
## 1  N844MH   320
```

Exploratory Data Analysis

Goals

> *"There are no routine statistical questions, only questionable statistical routines." — Sir David Cox*
>
> *"Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise." — John Tukey*

Learn how to use visualization and transformation to systematically explore your data to answer or generate questions about your data.

1. What type of variation occurs within my variables?
2. What type of covariation occurs between my variables?

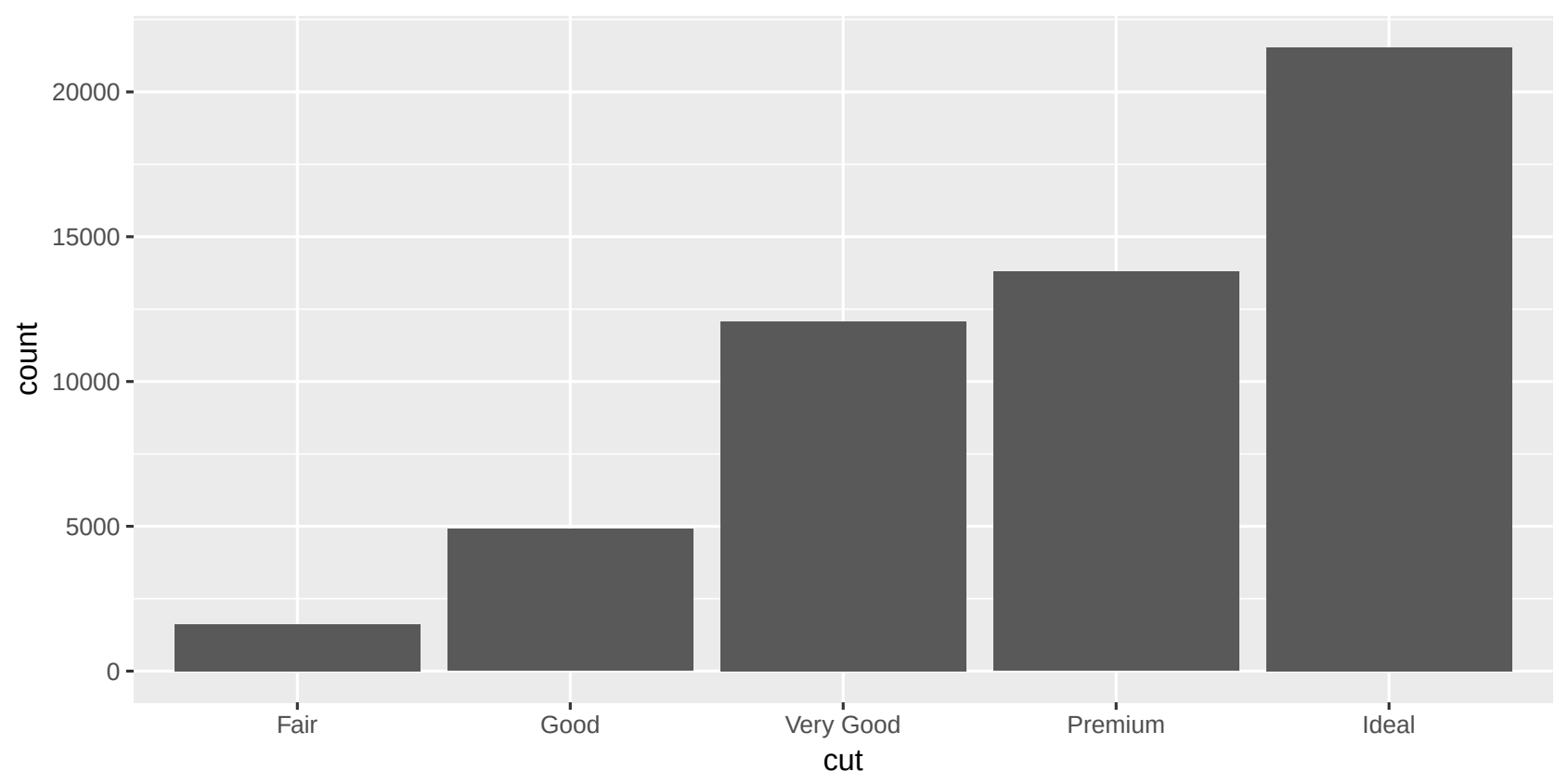
Variation

Variation

- Variation is the difference between expected output to observed output.
- Visualization of the distribution is different for categorical (`fctr`, `chr`) and continuous (`dbl`, `int`, `dttm`) variables

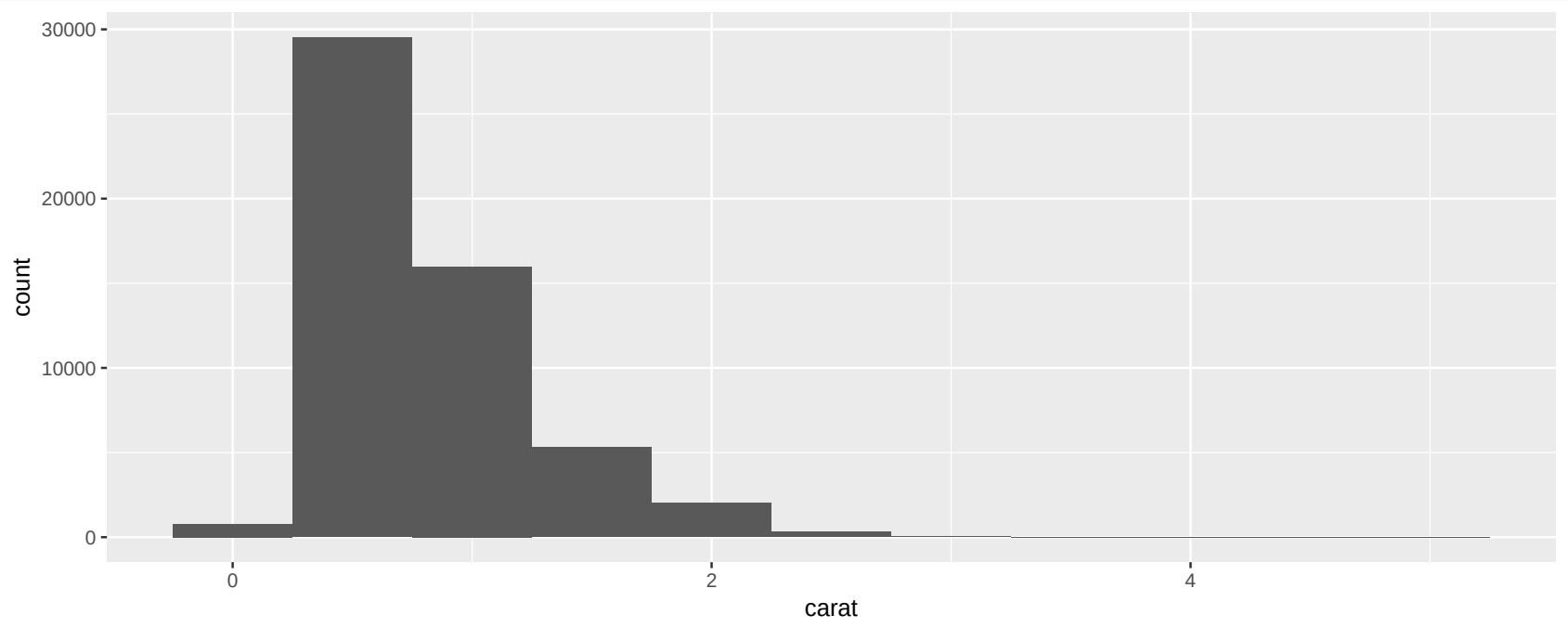
Distributions of categorical data

```
diamonds %>%
  ggplot() +
  geom_bar(mapping = aes(x = cut))
```



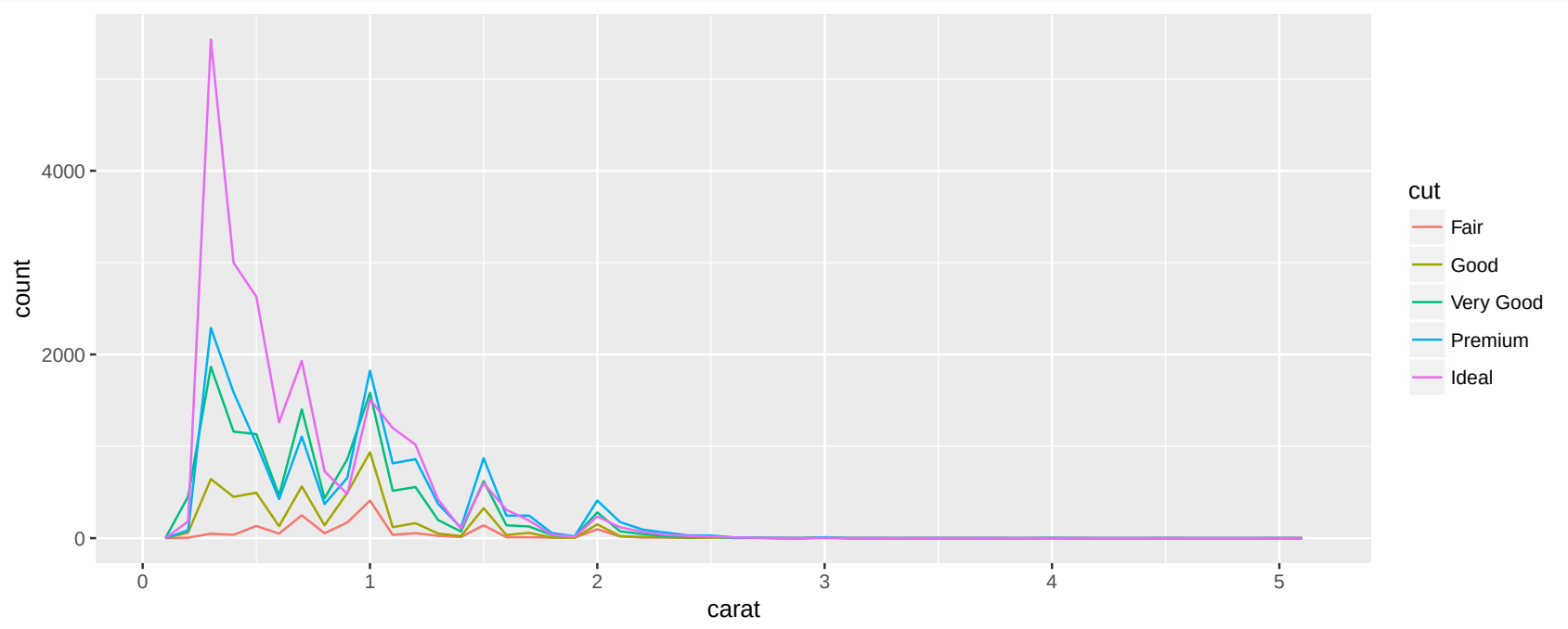
Distributions of continuous data

```
diamonds %>%
  ggplot() +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```



Overlaying multiple distributions

```
diamonds %>%
  ggplot(mapping = aes(x = carat, colour = cut)) +
  geom_freqpoly(binwidth = 0.1)
```



Typical values

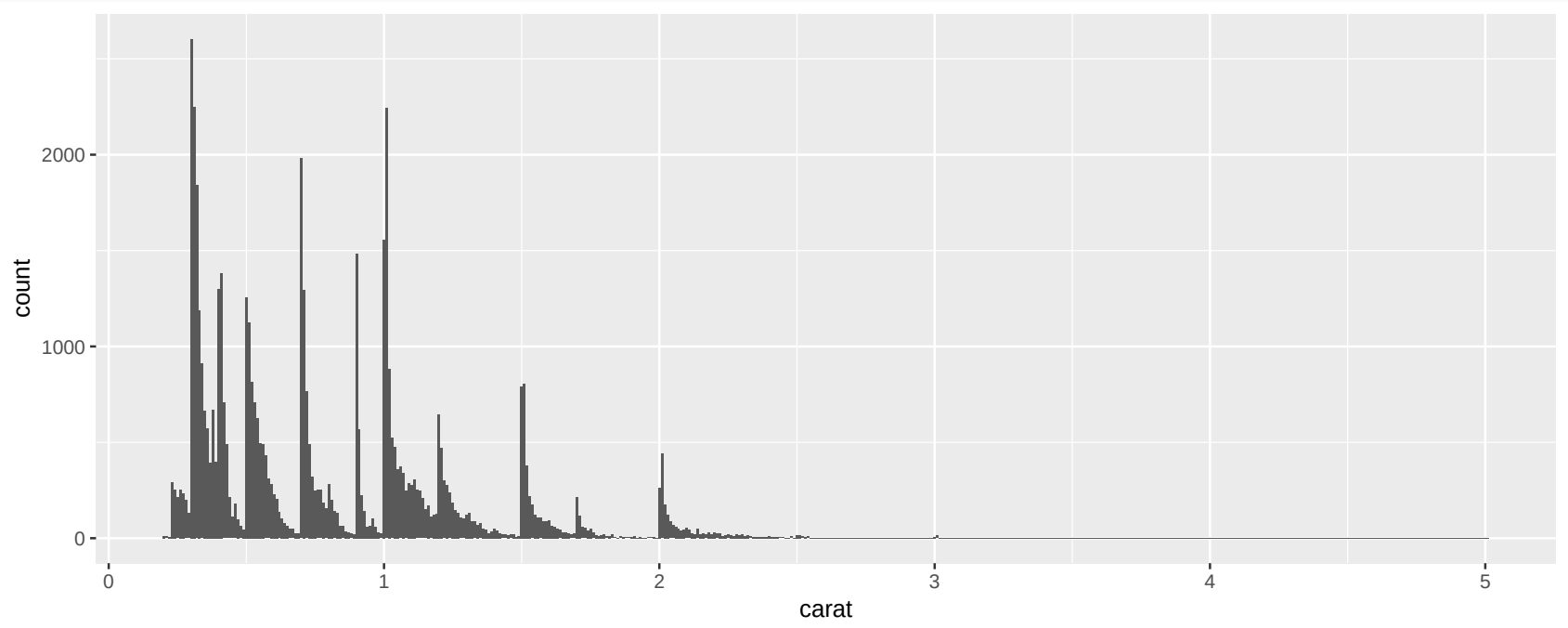
Look for anything unexpected:

- Which values are the most common? Why?
- Which values are rare? Why? Does that match your expectations?
- Can you see any unusual patterns? What might explain them?
- Do observations cluster together?
- How are the observations within each cluster similar to each other?

Typical values

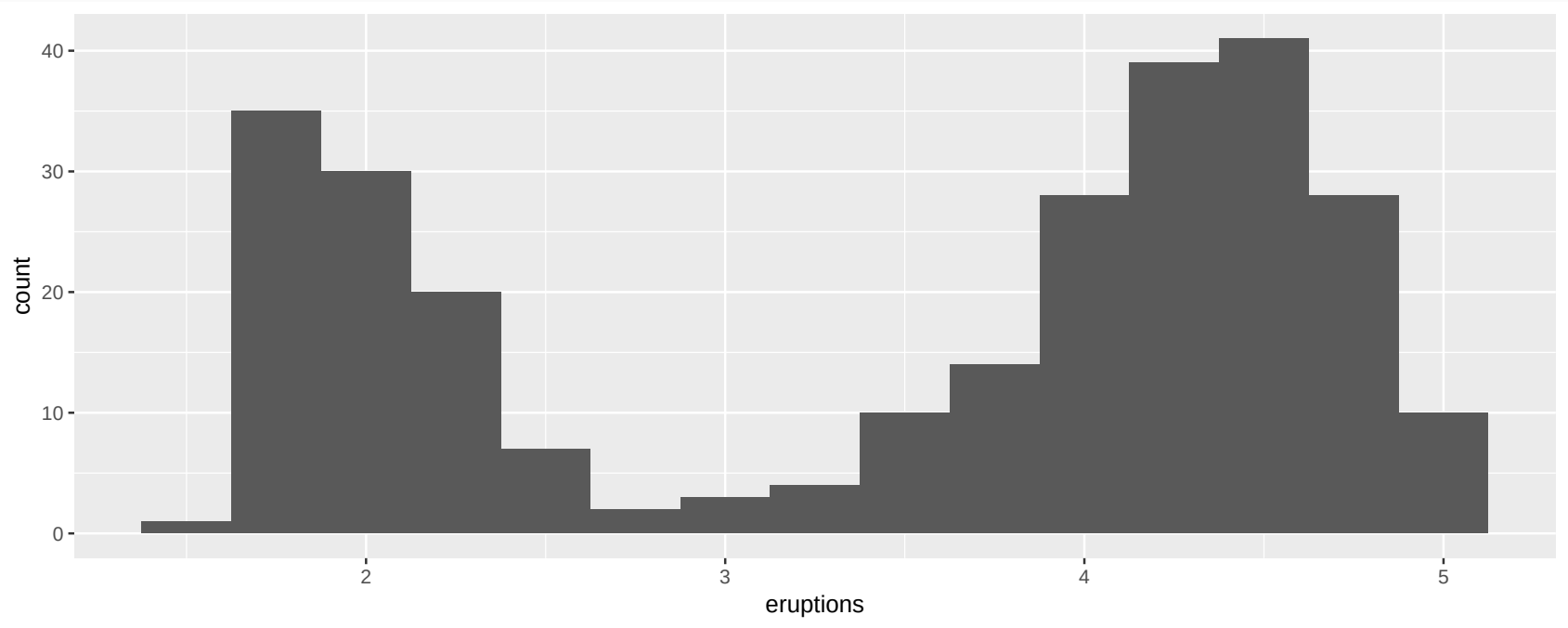
For example, what does this histogram tell you about typical values?

```
diamonds %>%
ggplot(mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.01)
```



Clusters of similar values

```
# Length of eruptions of the Old Faithful Geyser
faithful %>%
  ggplot(mapping = aes(x = eruptions)) +
  geom_histogram(binwidth = 0.25)
```



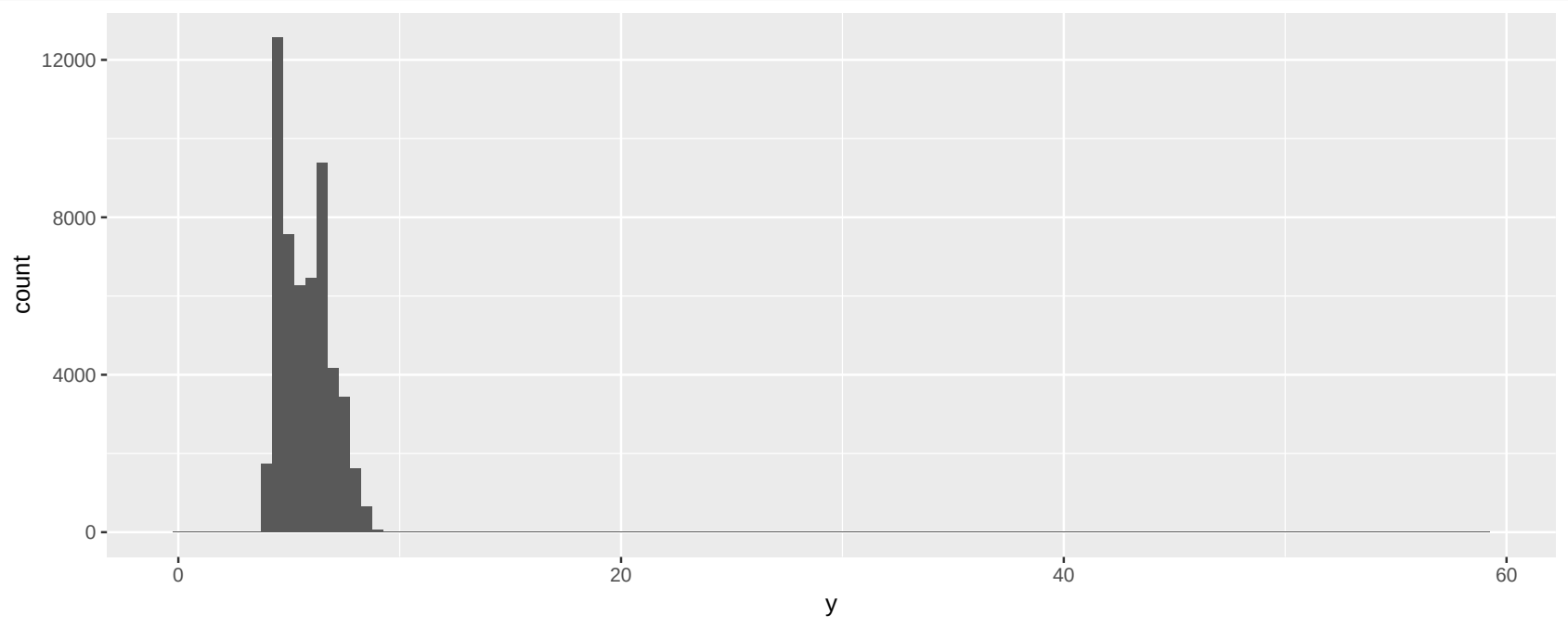
Outliers

Outliers are data points that do not seem to fit the pattern. They can have substantial effects on the results.

Outliers are difficult to detect in histograms so we may need to 'zoom in' on small values on the y or x-axis.

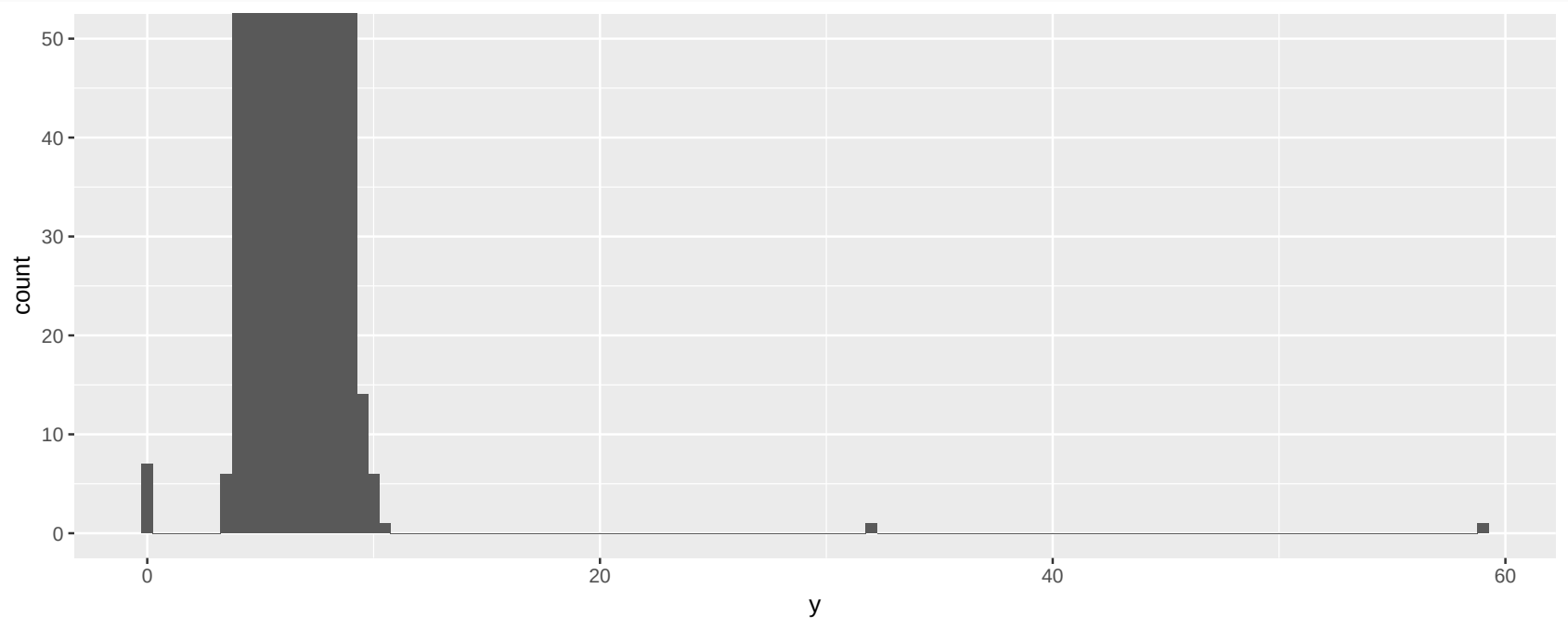
Outliers

```
diamonds %>%
  ggplot() +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```



Outliers

```
diamonds %>%
  ggplot() +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  coord_cartesian(ylim = c(0, 50))
```



Exercises

1. Using the diamonds data

    1.1 explore the distribution of each of the `x`, `y`, and `z` variables. Can you see any patterns?

    1.2 Explore the distribution of `price`. Do you discover anything unusual or surprising?

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

Exercises solutions

1.1 Explore the distribution of `x`, `y`, and `z`.

- They are right skewed (lots of small diamonds but few very large ones).
- There is an outlier in y, and z.
- All three distributions are bimodal

1.2 Explore the distribution of `price`.

- There are no diamonds with a price of 1500

1.3 How many diamonds are 0.99 carat?

- 70 times more 1 carat diamonds than 0.99 carat diamond.

Missing values

Dealing with unusual values

You can deal with outliers and other unusual in several ways:

1. Drop the entire row containing outliers:

```
diamonds2 <- diamonds %>%
  filter(between(y, 3, 20))
```

```
+ Why might this be a bad idea?
```

Dealing with unusual values

2. Replace unusual values with missing values

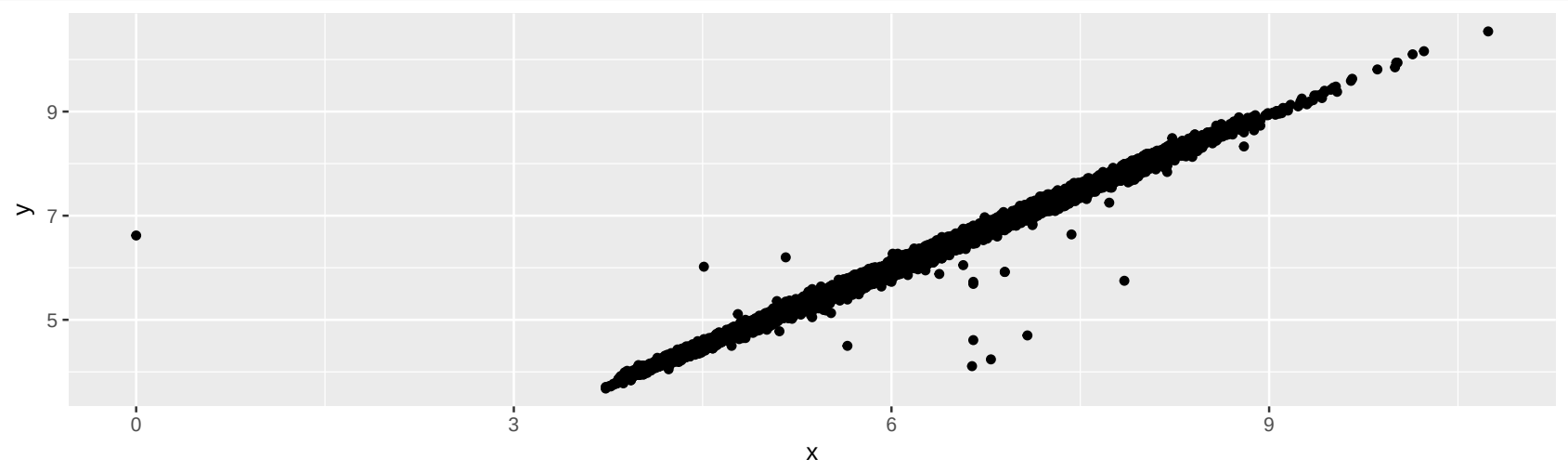
```
diamonds2 <- diamonds %>%
  mutate(y = ifelse(y < 3 | y > 20, NA, y))
```

ggplot2 way of plotting missing values

ggplot2 does not include missing values in the plot but displays a warning that they've been removed:

```
diamonds2 %>%
  ggplot(mapping = aes(x = x, y = y)) +
  geom_point()
```

```
## Warning: Removed 9 rows containing missing values (geom_
```



Exercises

2.1 What happens to missing values in a histogram? What happens to missing values in a bar chart? Why is there a difference?

Exercises solutions

2.1 What happens to missing values in a histogram / bar chart?

- In `geom_histogram` missing values are removed. In `geom_bar`, `NA` is treated as another category.

Covariation

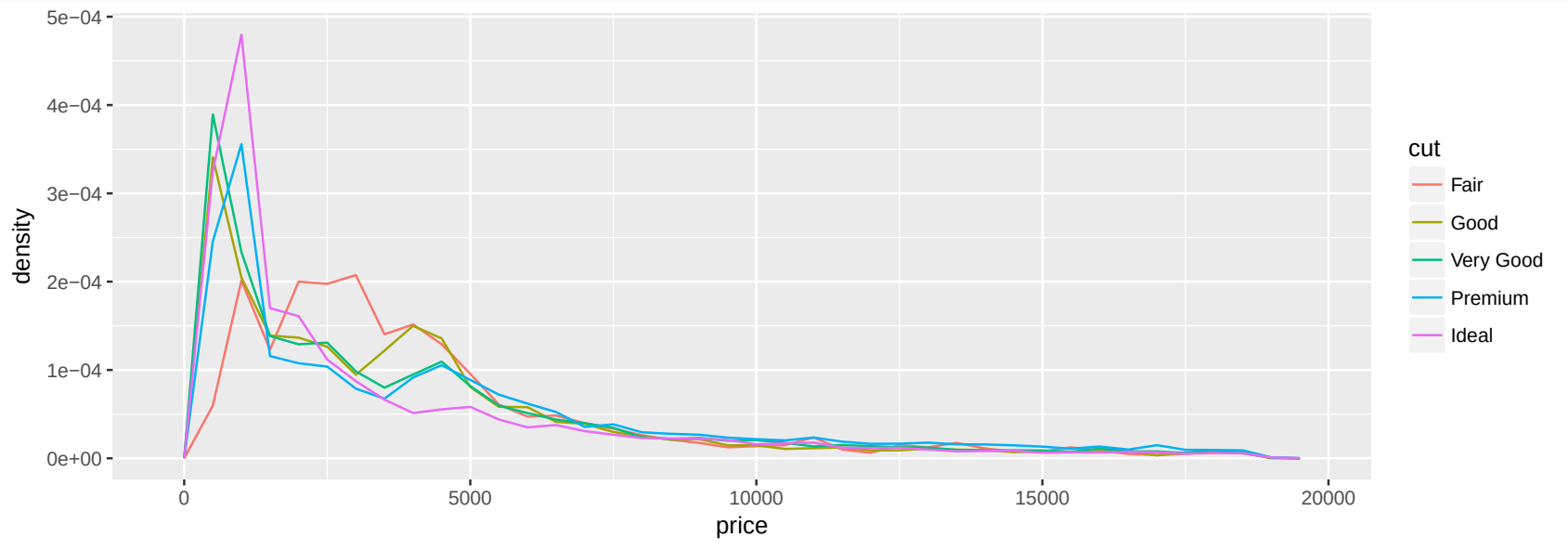
Covariation between continuous and categorical variables

Covariation is the tendency for the values of two or more variables to vary together in a related way.

For example, let look at the covariation between a continuous and categorical variable. We use `..density..`,which is the count standardised, to make the comparison easier.

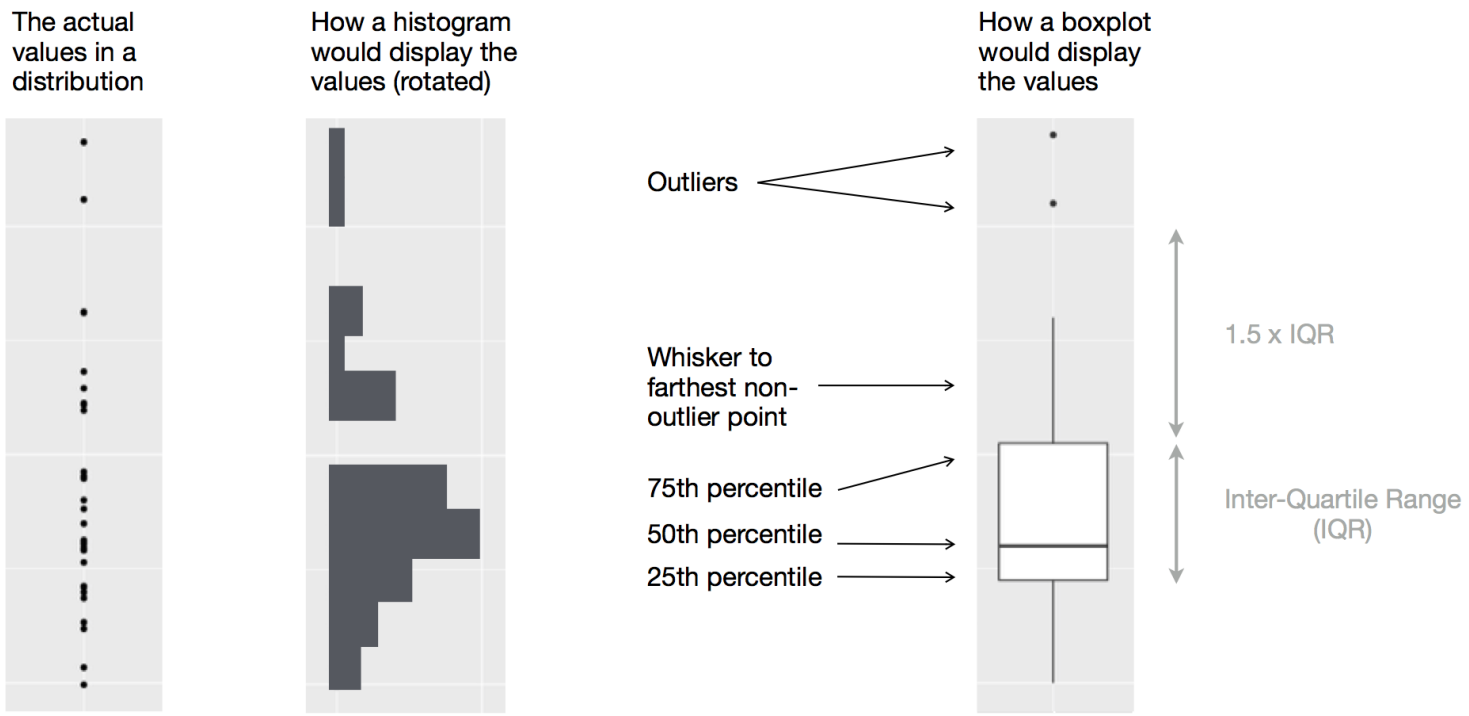
Covariation between continuous and categorical variables

```
diamonds %>%
  ggplot(mapping = aes(x = price, y = ..density..)) +
  geom_freqpoly(aes(colour = cut), binwidth = 500)
```



What do the distributions tell you?

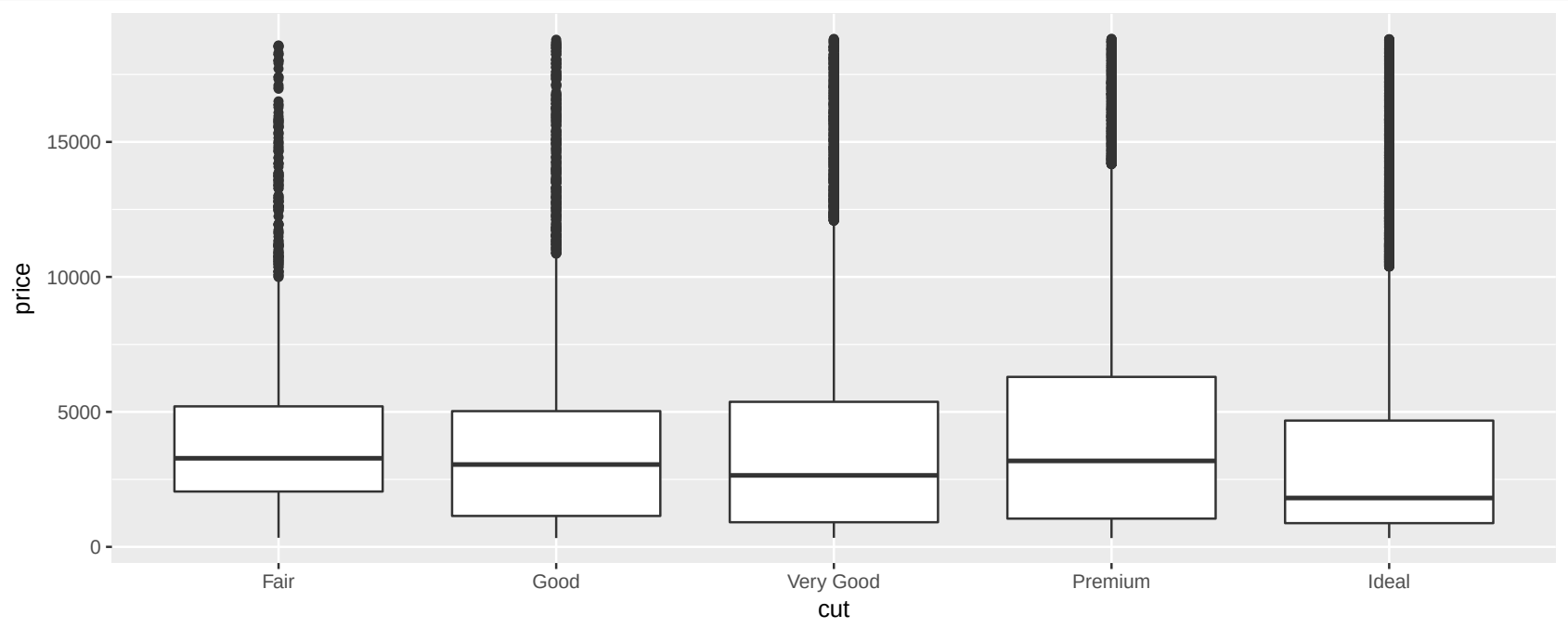
Boxplots


The actual values in a distribution
How a histogram would display the values (rotated)
How a boxplot would display the values
Outliers
Whisker to farthest non-outlier point
75th percentile
50th percentile
25th percentile
1.5 x IQR
Inter-Quartile Range (IQR)

Boxplots examples

We can use `geom_boxplot()` to plot the distribution of price by cut:

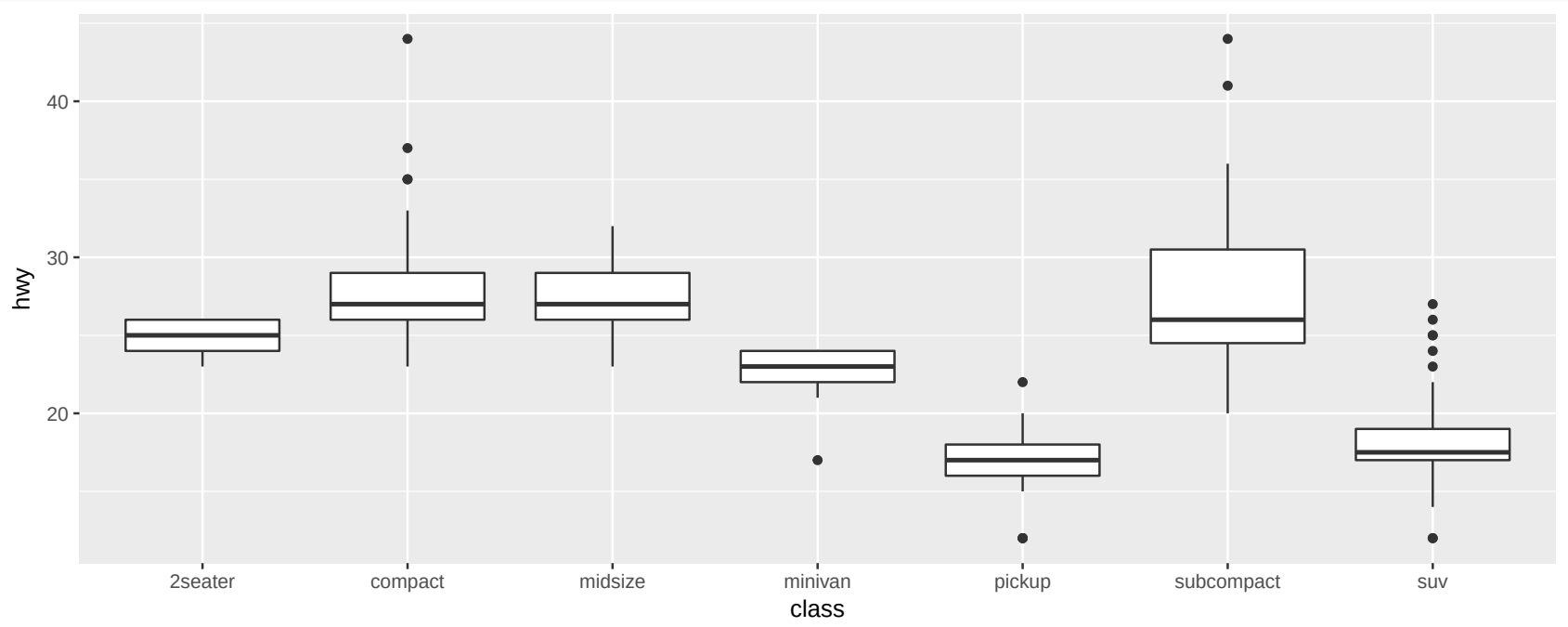
```
diamonds %>%
  ggplot(mapping = aes(x = cut, y = price)) +
  geom_boxplot()
```



Boxplots examples

Similarly, we can plot the distribution of `class` variable in `mpg`:

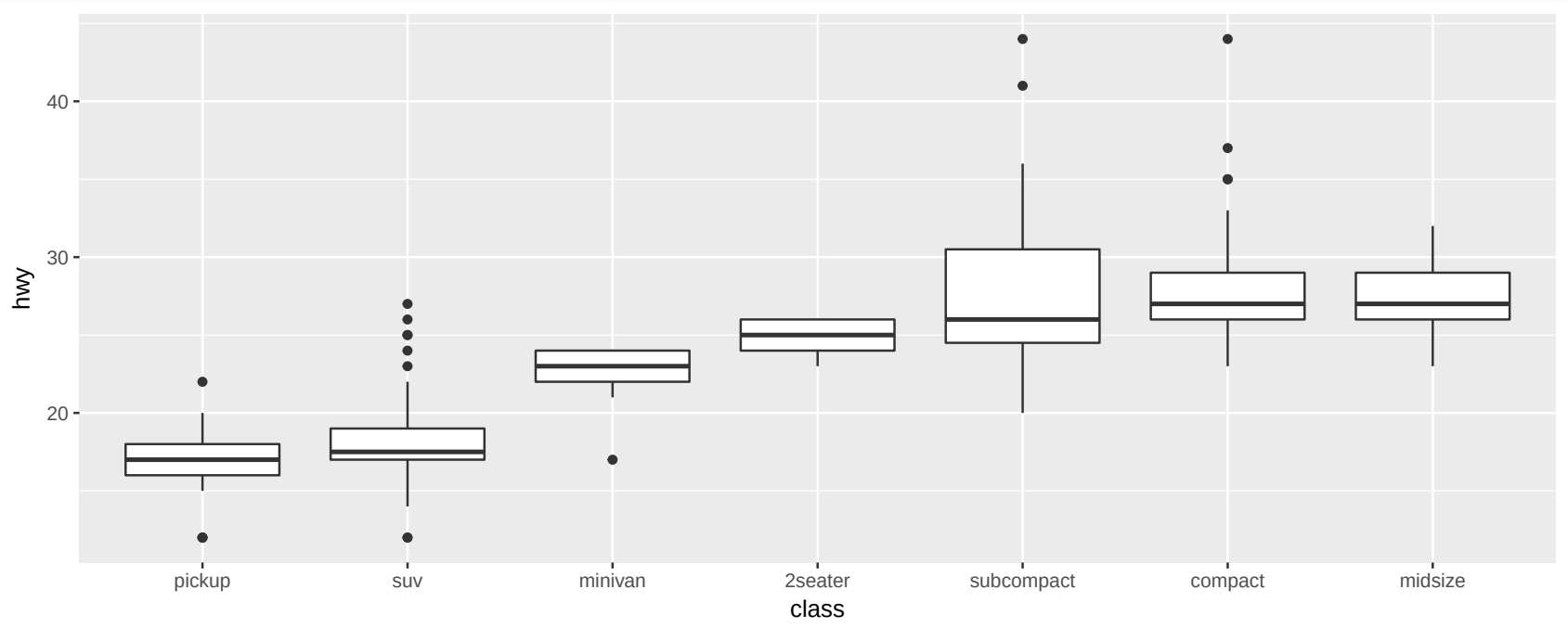
```
mpg %>%
  ggplot(mapping = aes(x = class, y = hwy)) +
  geom_boxplot()
```



Boxplots examples

Reorder `class` based on the median value of `hwy`:

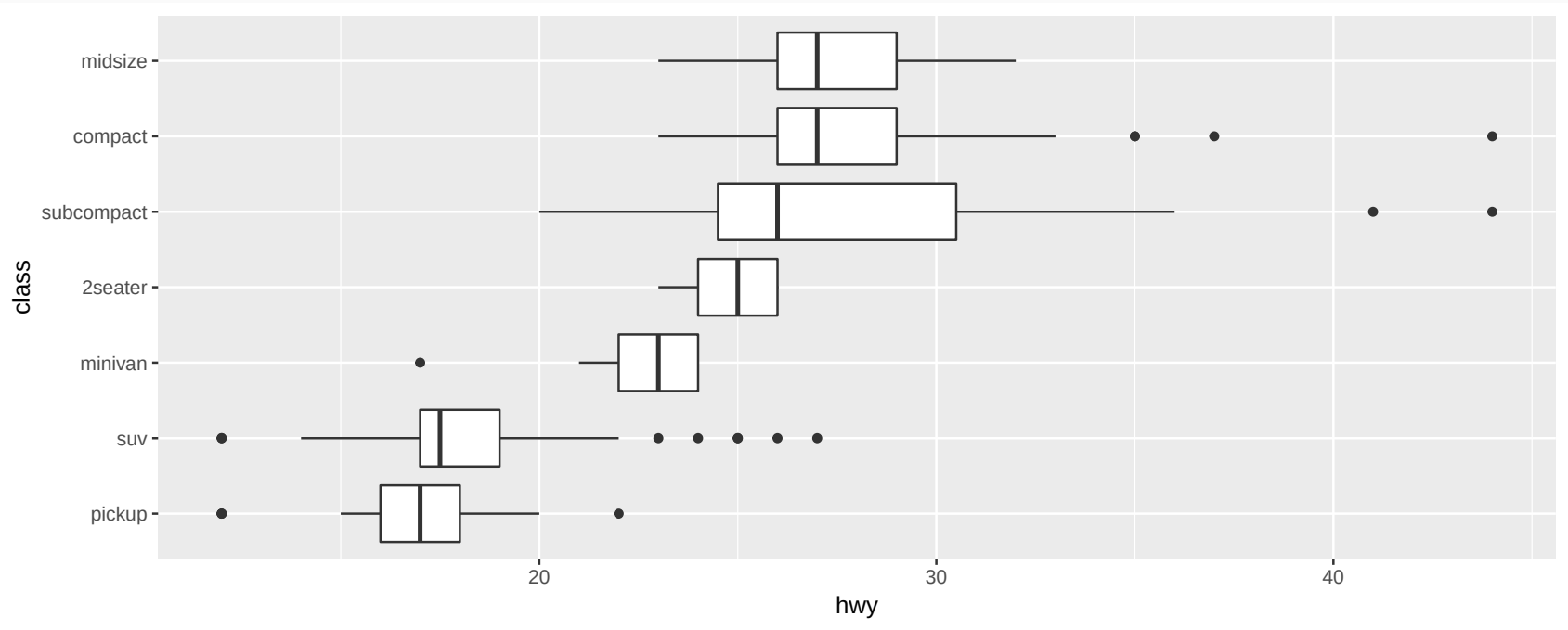
```
mpg %>%
  ggplot(mapping = aes(x = class, y = hwy)) +
  geom_boxplot(aes(x = reorder(class, hwy, FUN = median),
                   y = hwy))
```



Boxplots examples

Rotate the plot by 90°:

```
mpg %>%
  ggplot(mapping = aes(x = class, y = hwy)) +
  geom_boxplot(aes(x = reorder(class, hwy, FUN = median),
                   y = hwy)) +
  coord_flip()
```



Exercises

3.1 Create a boxplot to visualize the departure times of cancelled vs. non-cancelled flights.

3.2 Compare and contrast `geom_violin()` with a facetted `geom_histogram()`, or a coloured `geom_freqpoly()`. What are the pros and cons of each method?

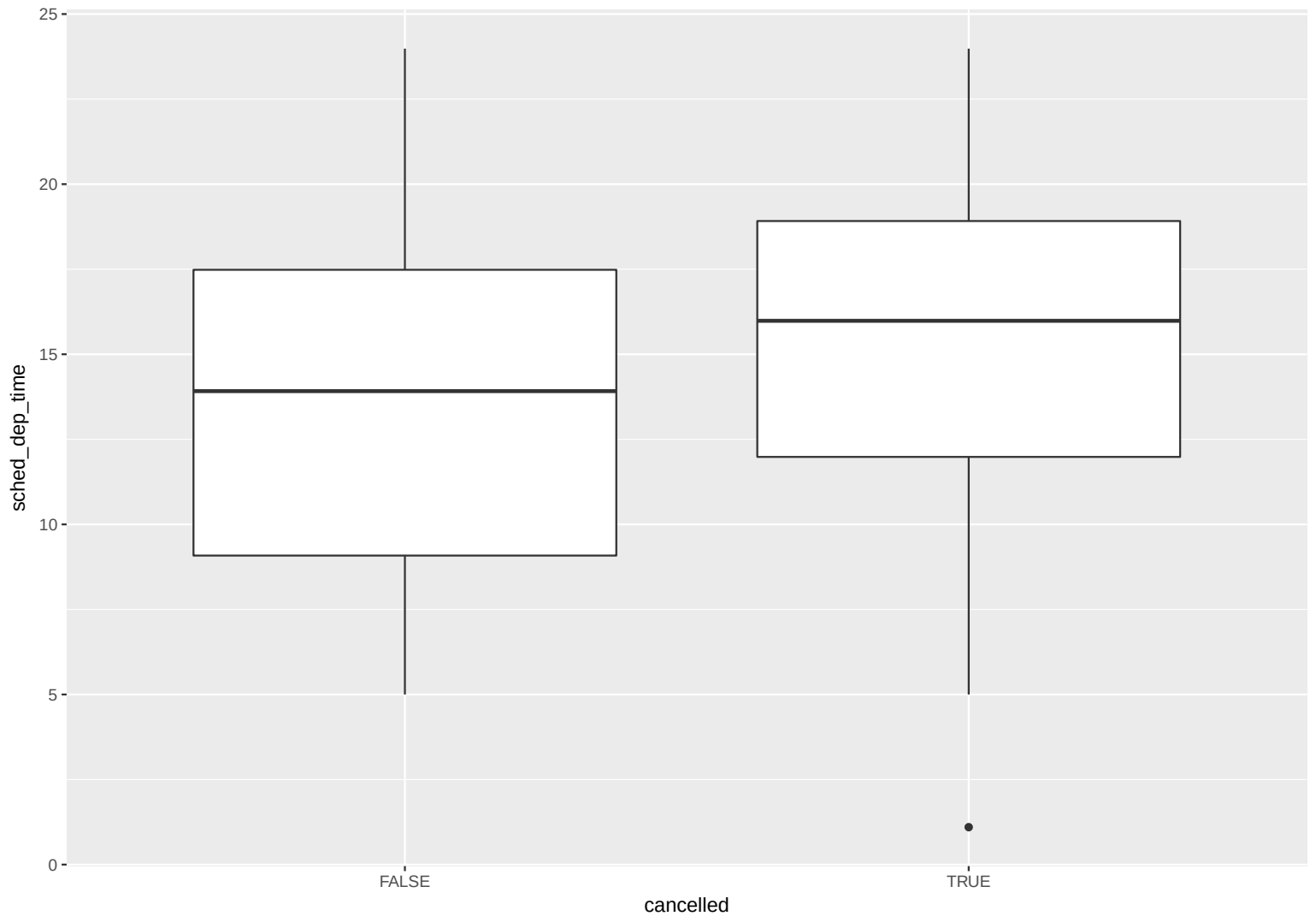
Exercises solutions

3.1 Departure times of cancelled vs. non-cancelled flights

```
nycflights13::flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %/% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + sched_min / 60
  ) %>%
  ggplot() +
    geom_boxplot(mapping = aes(y = sched_dep_time,
                               x = cancelled))
```

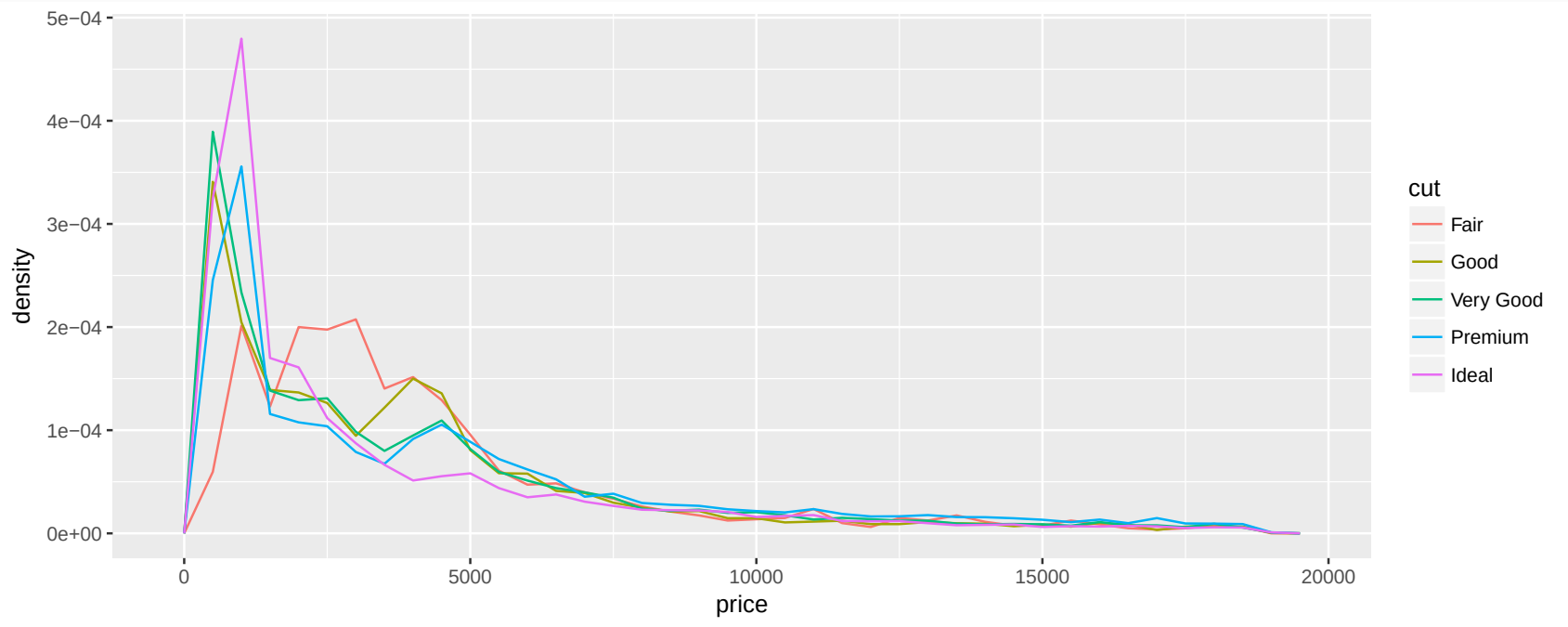
Exercises solutions

3.1 Departure times of cancelled vs. non-cancelled flights



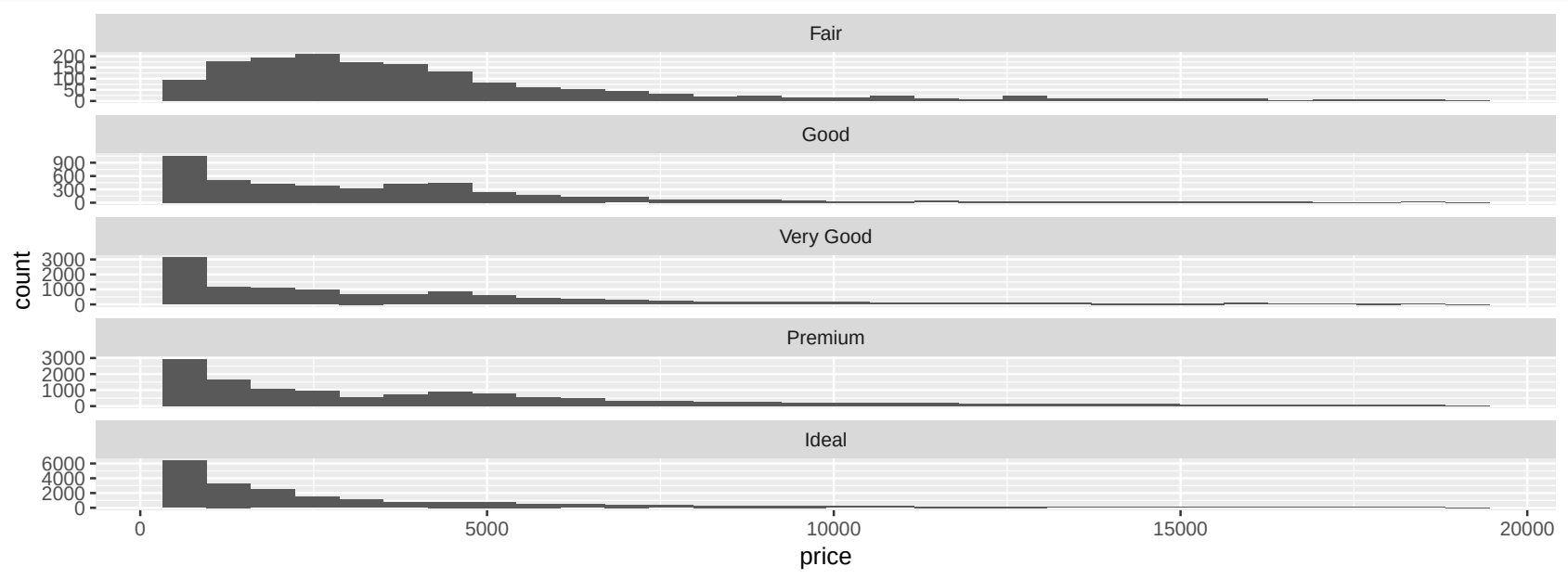
Exercises solutions

```
# `geom_freqpoly()`
diamonds %>%
  ggplot(mapping = aes(x = price, y = ..density..)) +
  geom_freqpoly(aes(colour = cut), binwidth = 500)
```



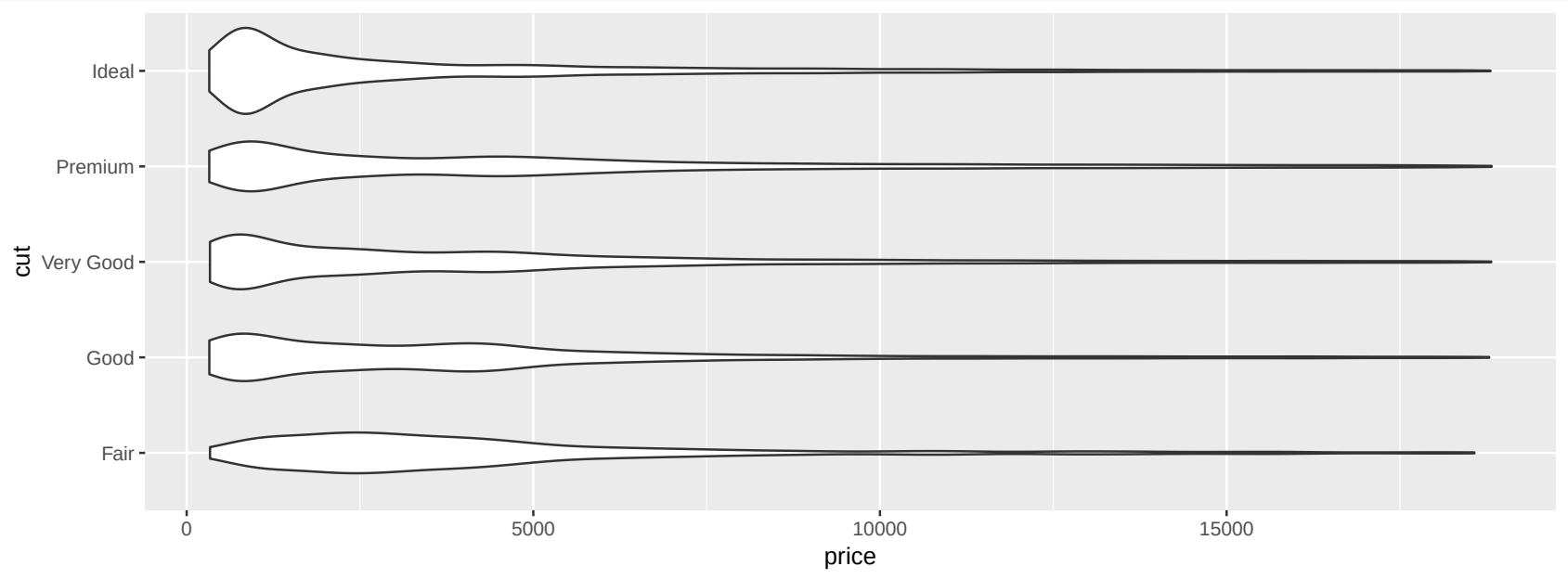
Exercises solutions

```
# `geom_violin`
diamonds %>%
  ggplot(mapping = aes(x = price)) +
  geom_histogram() +
  facet_wrap(~ cut, ncol = 1, scales = "free_y")
```



Exercises solutions

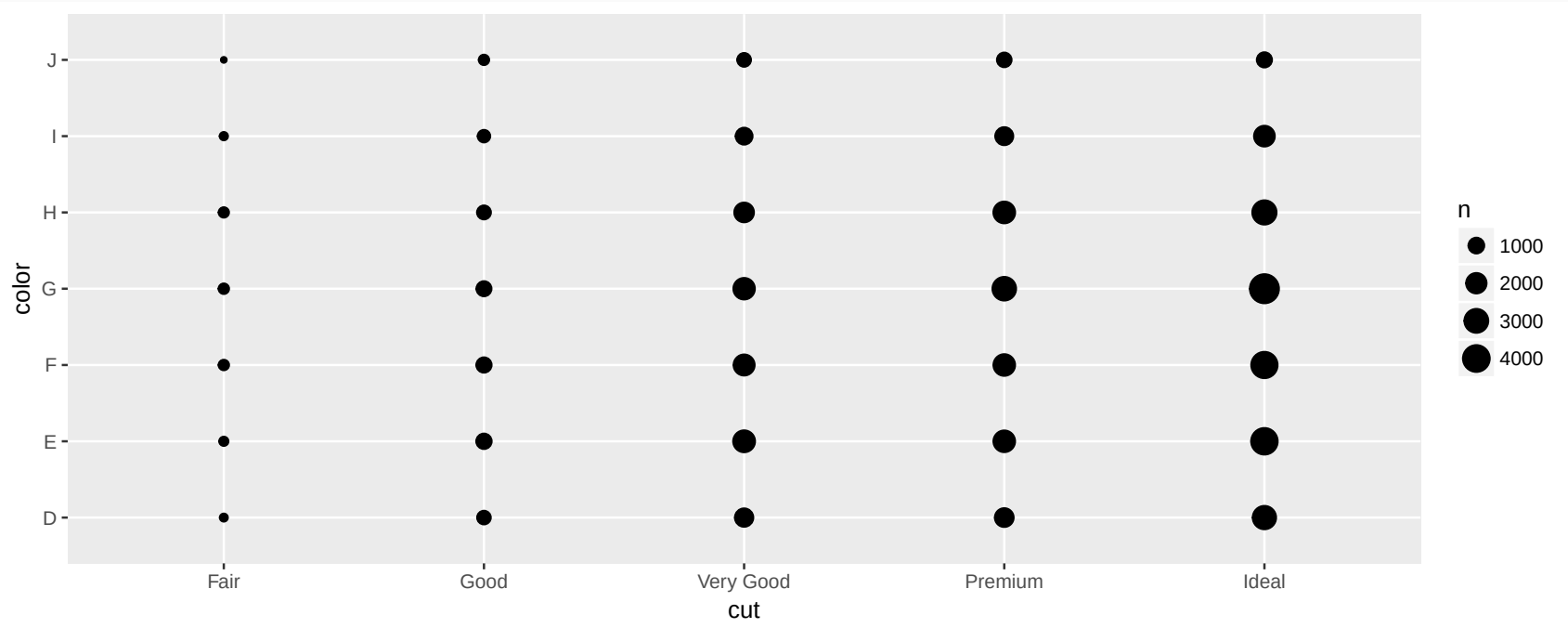
```
# facetted `geom_histogram`
diamonds %>%
  ggplot(mapping = aes(x = cut, y = price)) +
  geom_violin() +
  coord_flip()
```



Covariation between two categorical variables

To visualise the covariation between categorical variables, we'll need to count the number of observations for each combination.

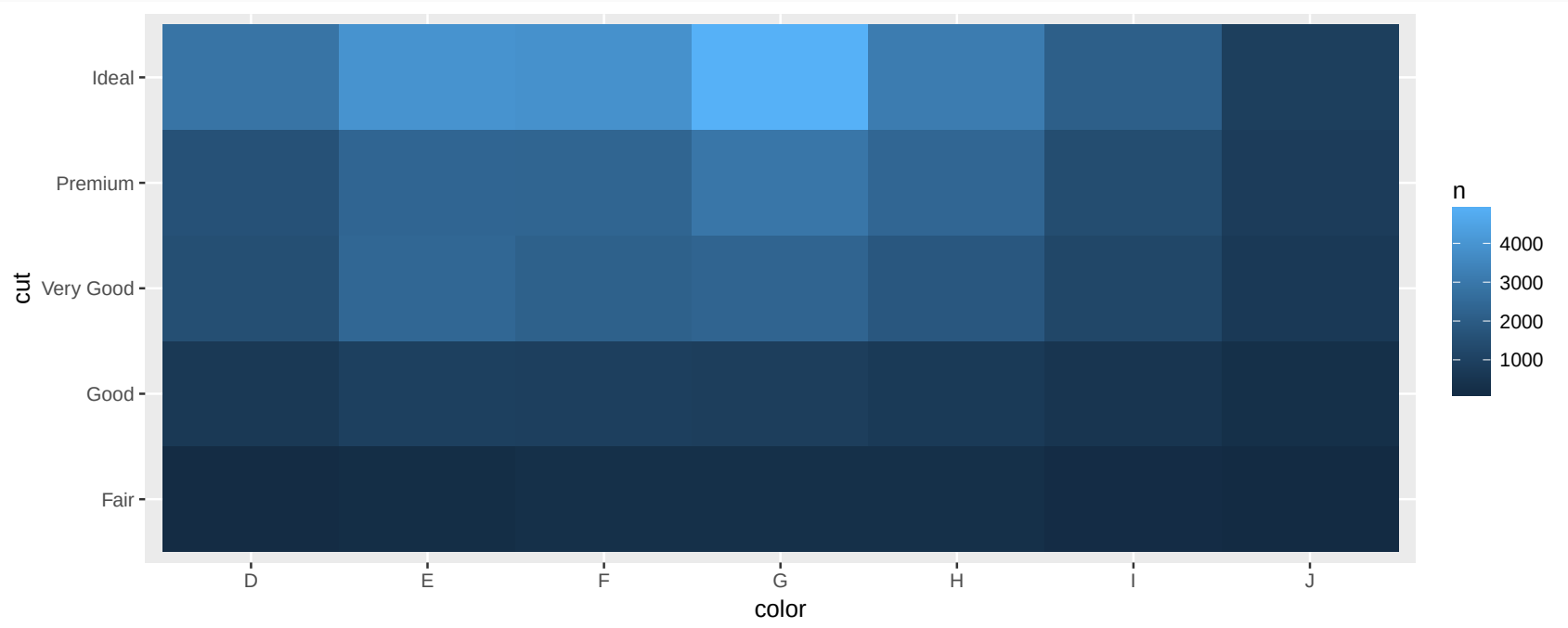
```
diamonds %>%
  ggplot() +
  geom_count(mapping = aes(x = cut, y = color))
```



Covariation between two categorical variables

Or use `dplyr()` and plot with `geom_tile`:

```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
    geom_tile(mapping = aes(fill = n))
```



Exercises

4.1 Why is it slightly better to use `aes(x = color, y = cut)` rather than `aes(x = cut, y = color)` in the example above?

4.2 Use `geom_tile()` together with dplyr to explore how average flight delays vary by destination and month of year. What makes the plot difficult to read? How could you improve it?

Exercises solutions

4.1. Why is it better to use `aes(x = color, y = cut)`

- Better to plot the categorical variable with a larger number of categories or the longer labels on the y-axis
- Plot becomes easier to read

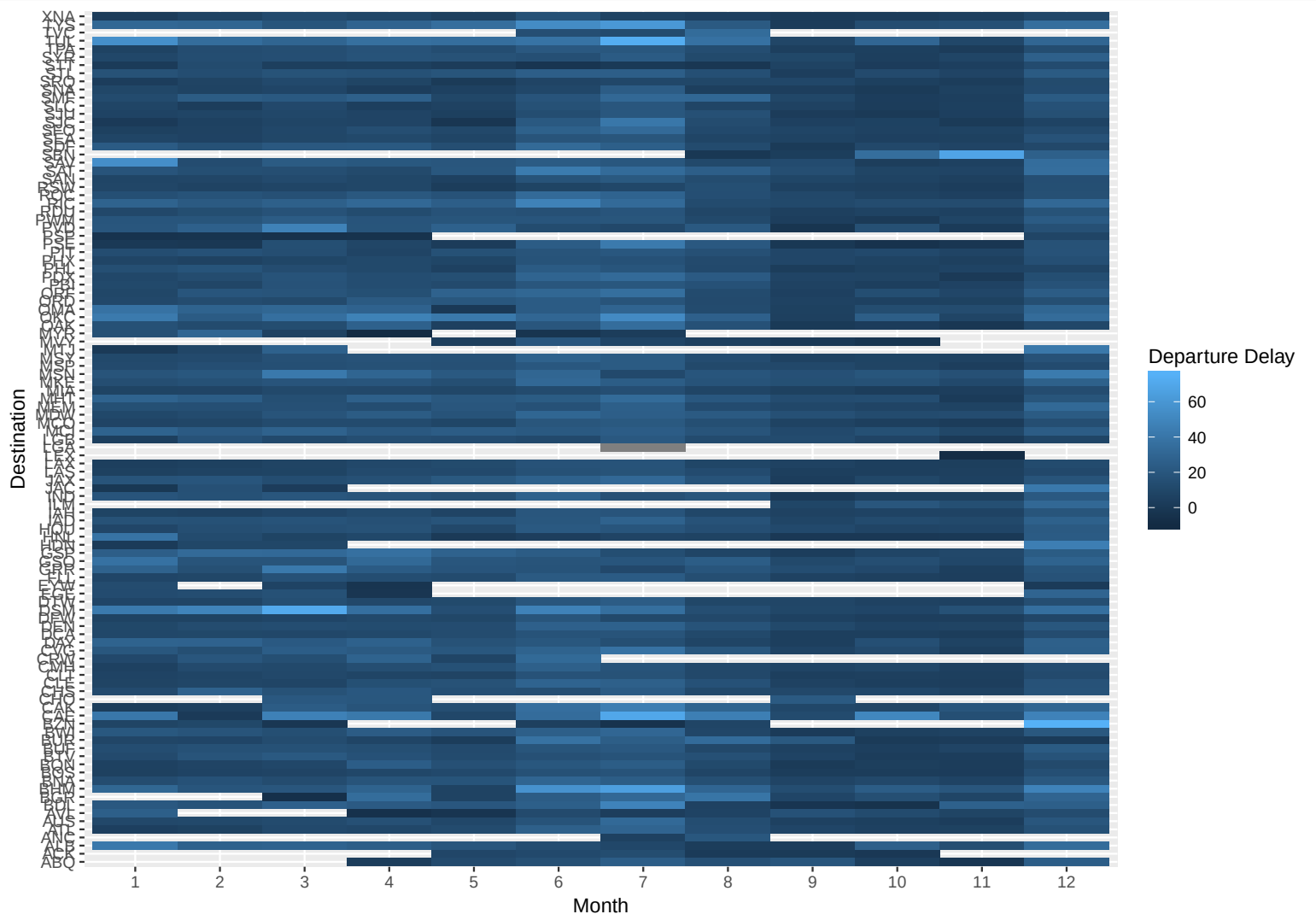
Exercises solutions

4.2 Explore how average flight delays vary by destination and month.

```
flights %>%
  group_by(month, dest) %>%
  summarise(dep_delay = mean(dep_delay, na.rm = TRUE)) %>%
  ggplot(aes(x = factor(month), y = dest,
             fill = dep_delay)) +
  geom_tile() +
  labs(x = "Month", y = "Destination",
       fill = "Departure Delay")
```

Exercises solutions

4.2 Explore how average flight delays vary by destination and month.



Exercises solutions

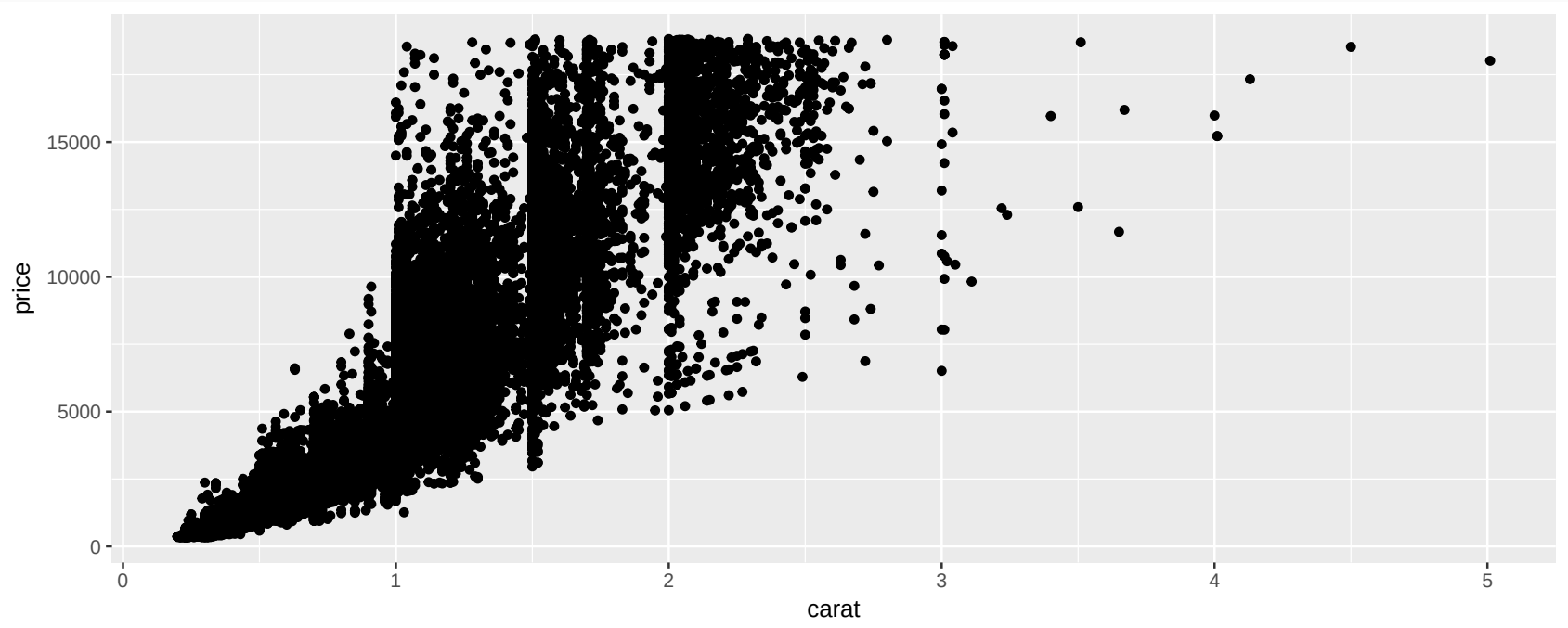
4.2 Explore how average flight delays vary by destination and month. What makes the plot difficult to read? How could you improve it?

- sort destinations by a meaningful quanity (e.g. distance)
- remove missing values
- better color scheme

Covariation between two continuous variables

Easiest way to visualize the covariation between two continuous variables is to draw a scatterplot with `geom_point()`.

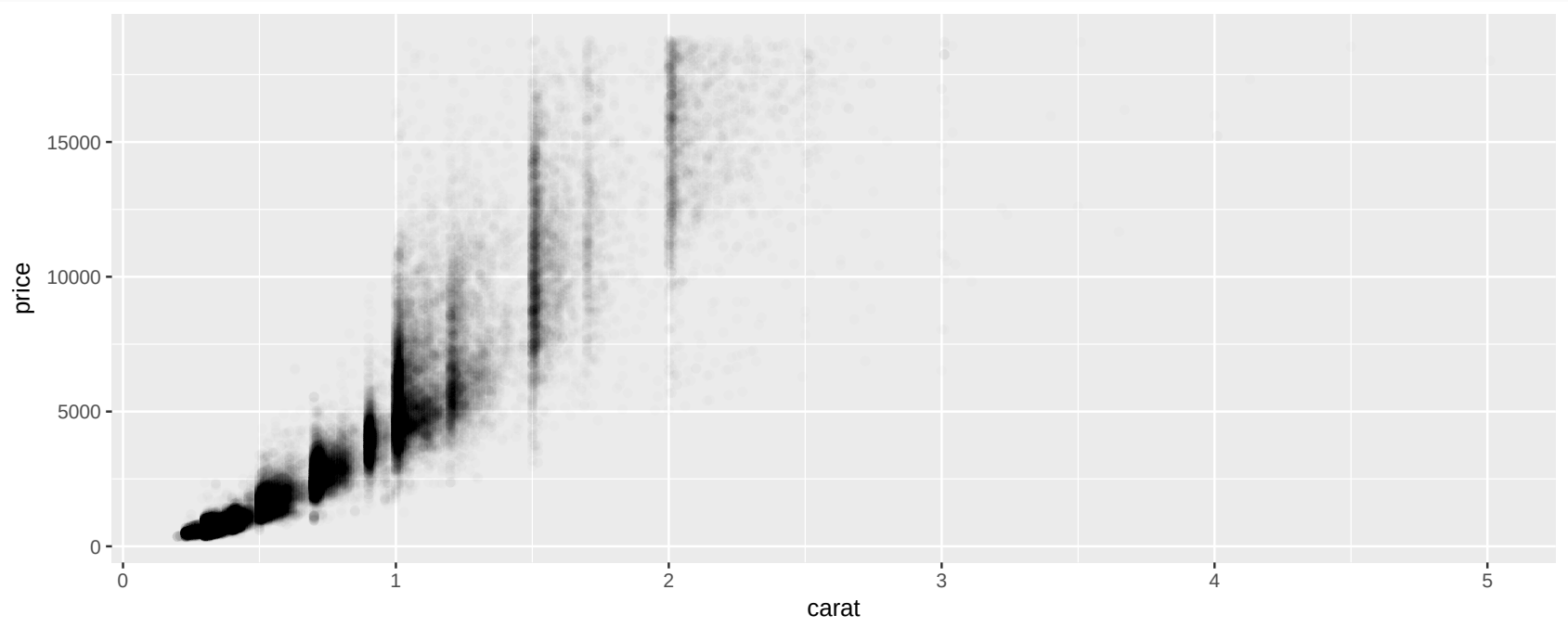
```
diamonds %>%
  ggplot() +
  geom_point(mapping = aes(x = carat, y = price))
```



Covariation between two continuous variables

Scatterplots become less useful as the size of the dataset grows because points begin to overplot. One solution is to increase transparency with `alpha`.

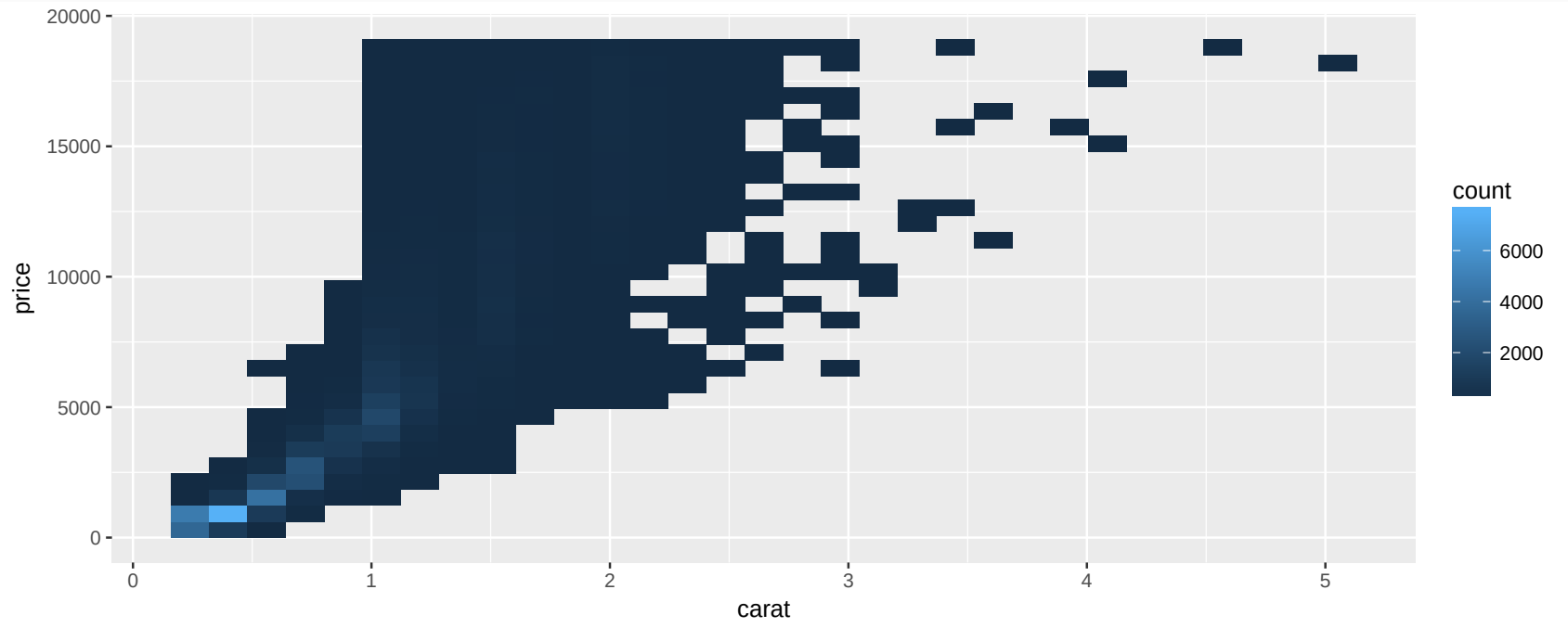
```
diamonds %>%
  ggplot() +
  geom_point(mapping = aes(x = carat, y = price), alpha = 1
```



Covariation between two continuous variables

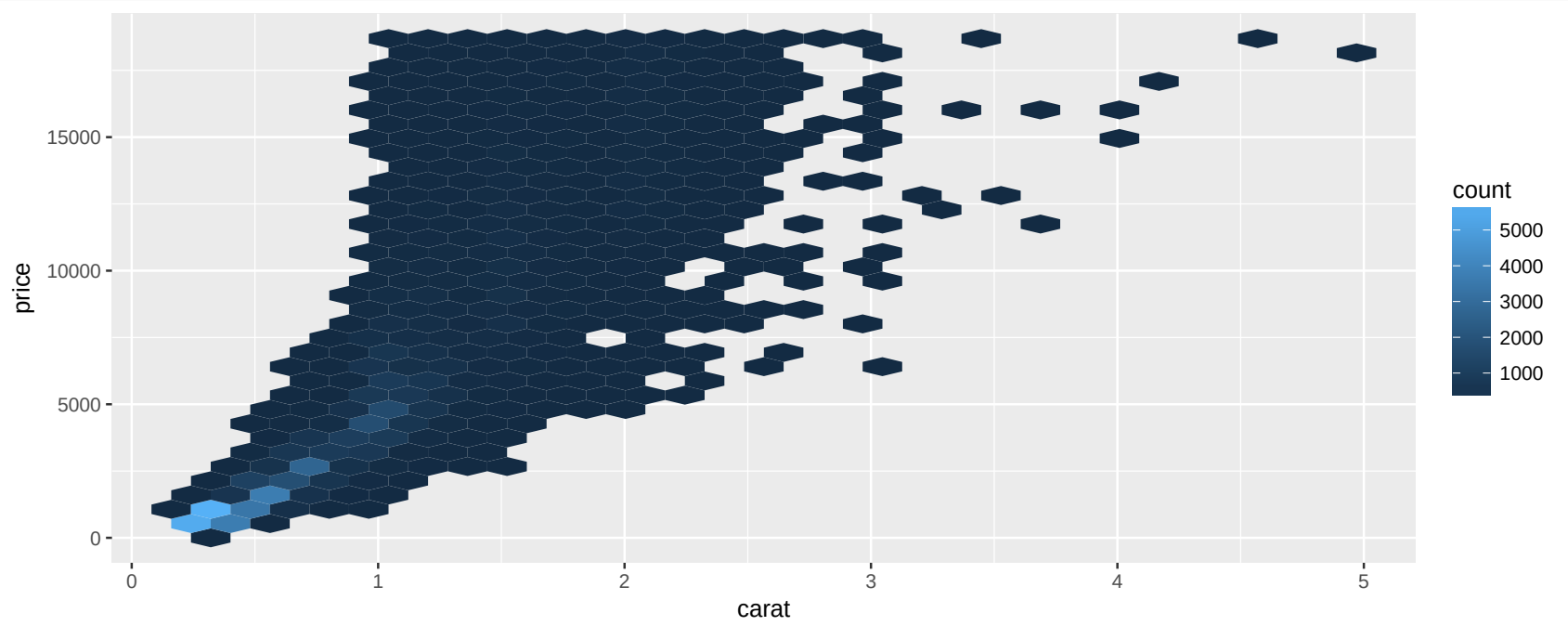
Another solution is to use `geom_bin2d()` and `geom_hex()` that divide the coordinate plane into 2d bins.

```
diamonds %>%
  ggplot() +
  geom_bin2d(mapping = aes(x = carat, y = price))
```



Covariation between two continuous variables

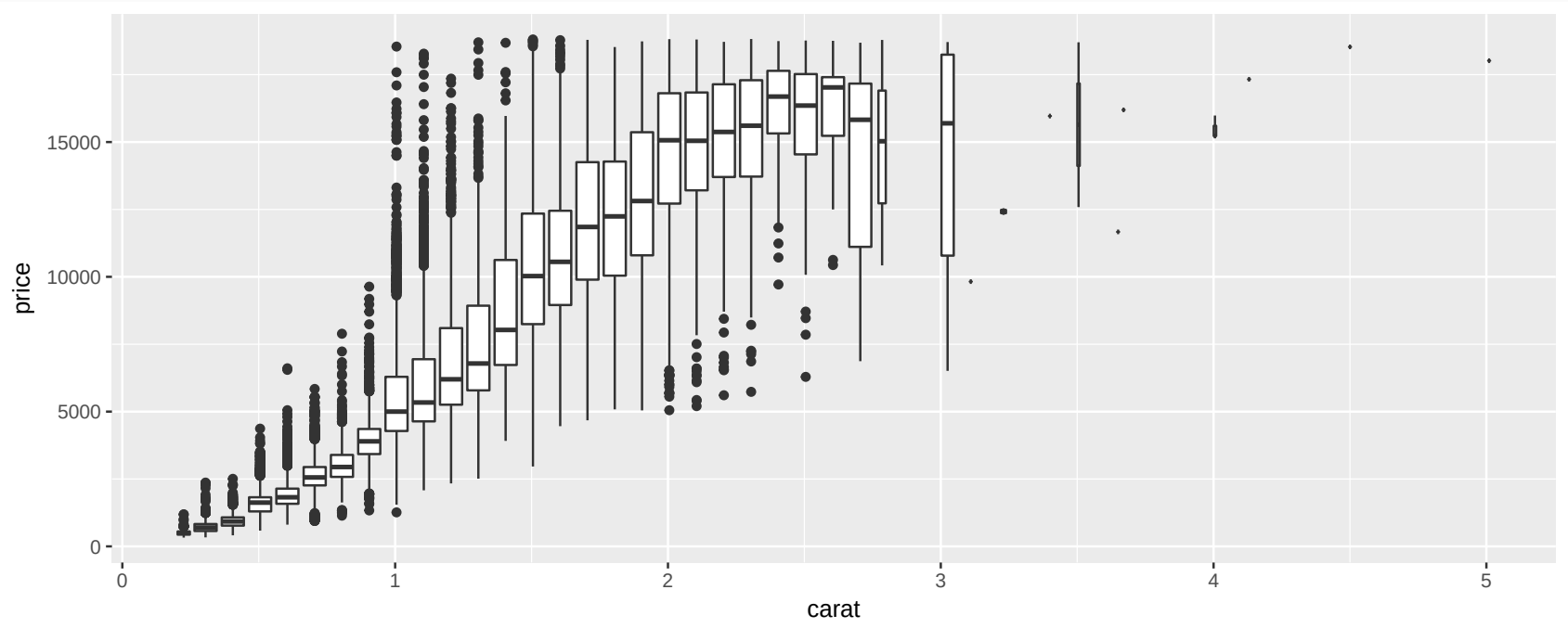
```
# install.packages("hexbin")
library(hexbin)
diamonds %>%
  ggplot() +
  geom_hex(mapping = aes(x = carat, y = price))
```



Covariation between two continuous variables

Another third option is to bin one continuous variable so it acts like a categorical variable.

```
diamonds %>%
  ggplot( mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.1))
```



Exercises

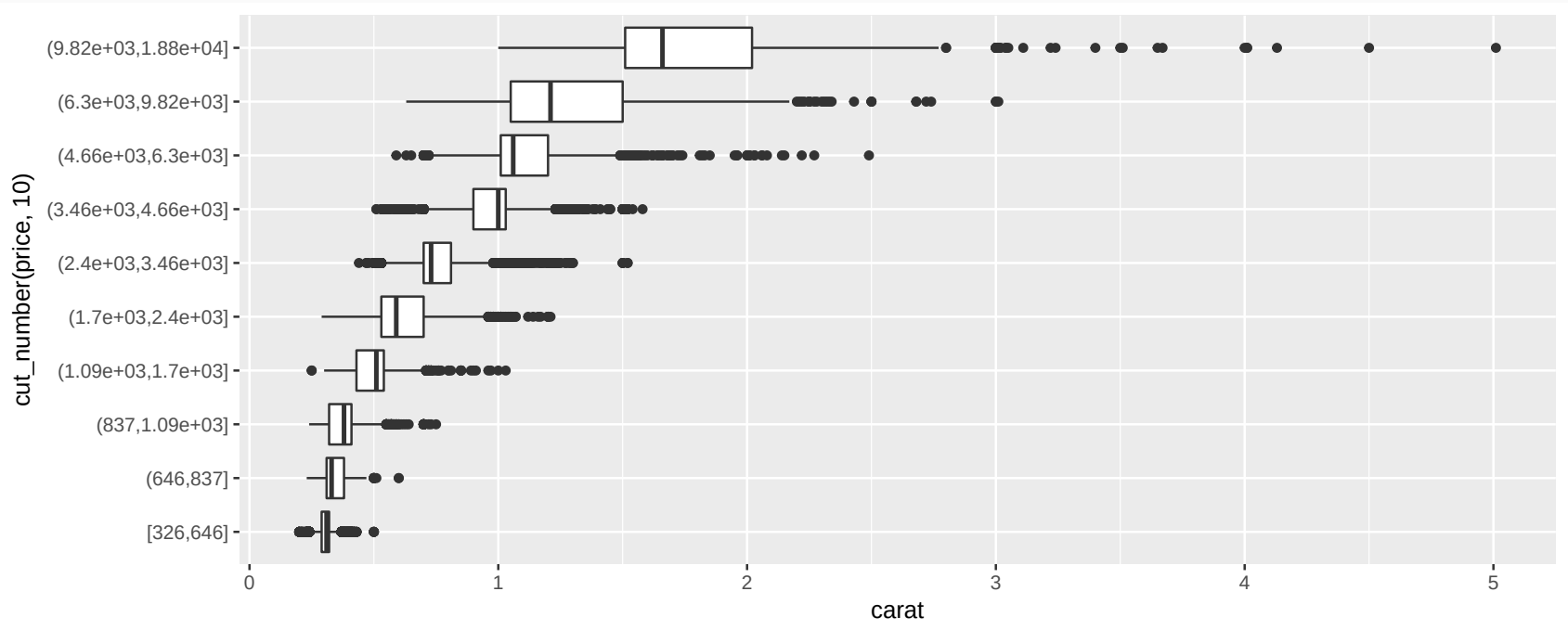
5.1 Visualise the distribution of carat, partitioned by price. How does the price distribution of very large diamonds compare to small diamonds (Hint: Look at the `cut_number` function).

5.2 Visualise the combined distribution of cut, carat, and price (Hint: Look at the `cut_number` function).

Exercises solutions

5.1 Visualise the distribution of carat, partitioned by price.

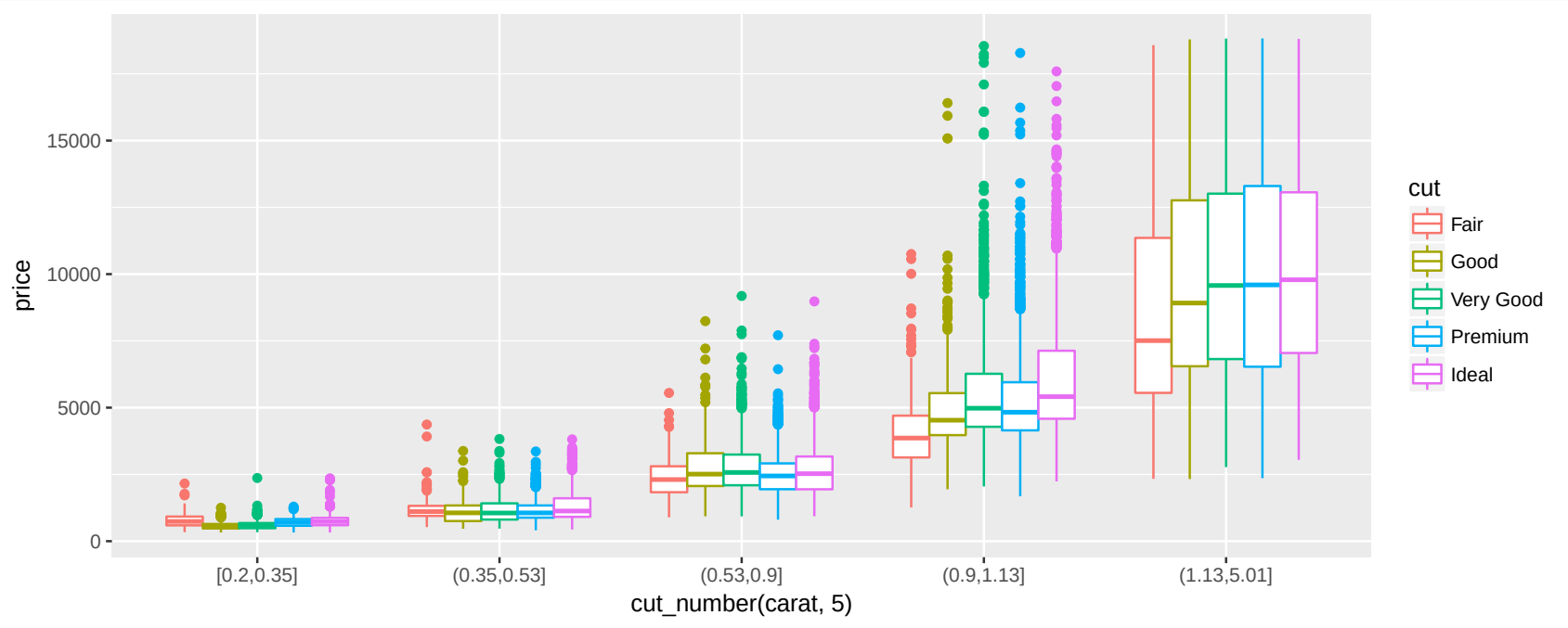
```
diamonds %>%
  ggplot(mapping = aes(x = cut_number(price, 10),
                       y = carat)) +
  geom_boxplot() +
  coord_flip()
```



Exercises solutions

5.2 Visualise the combined distribution of cut, carat, and price.

```
diamonds %>%
  ggplot(mapping = aes(x = cut_number(carat, 5),
                       y = price, color = cut)) +
  geom_boxplot()
```



Final Remarks about Patterns

Final Remarks about Patterns

Patterns provide clues about relationships because they reveal covariation. If you spot a pattern, ask yourself:

- Could this pattern be due to coincidence?
- How can you describe the relationship implied by the pattern?
- How strong is the relationship implied by the pattern?
- What other variables might affect the relationship?
- Does the relationship change if you look at individual subgroups of the data?

Homework Exercises

Homework Exercises

For this week's homework exercise go to DataCamp and complete the course on *Exploratory Data Analysis*. You have to create an account on DataCamp and join the *Data Management with R* group using the link in the email invitation that I sent a week ago.

Deadline: Sunday, October 1 before midnight.

That's it for today. Questions?