

# r4ds Ex 7

MW

2019/06/12

## 7.3.4

### 1

Explore the distribution of each of the `x`, `y`, and `z` variables in `diamonds`. What do you learn? Think about a diamond and how you might decide which dimension is the length, width, and depth.

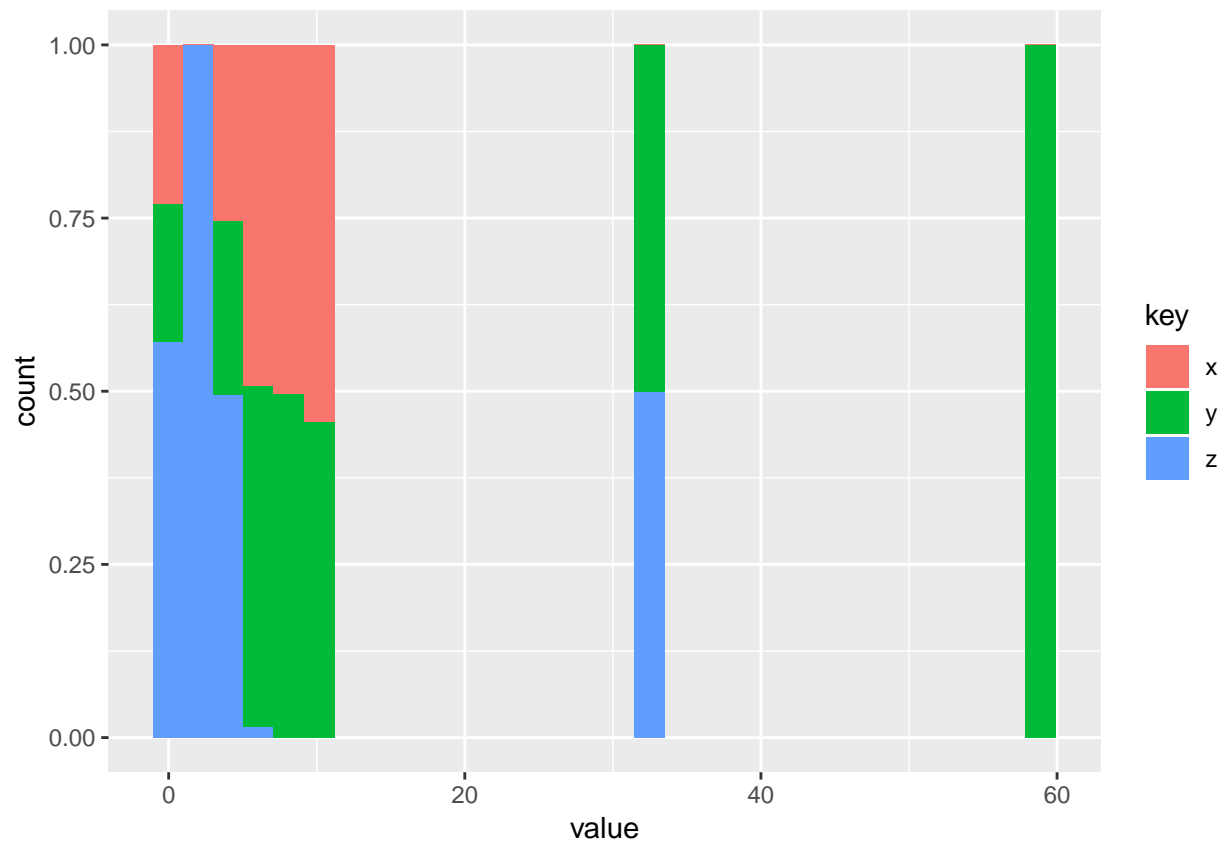
```
diamonds %>% select(x,y,z) %>% summary
```

```
##           x           y           z
##  Min.    : 0.000   Min.    : 0.000   Min.    : 0.000
## 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.700   Median : 5.710   Median : 3.530
##  Mean   : 5.731   Mean    : 5.735   Mean    : 3.539
## 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :10.740   Max.    :58.900   Max.    :31.800
```

```
diamonds %>% select(x,y,z) %>%
  gather() %>%
  ggplot(aes(x=value, fill = key)) +
  geom_histogram(position = "fill")
```

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

```
## Warning: Removed 66 rows containing missing values (geom_bar).
```



2

Explore the distribution of price. Do you discover anything unusual or surprising? (Hint: Carefully think about the `binwidth` and make sure you try a wide range of values.)

3

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

```
diamonds %>% filter(carat >= 0.99, carat <= 1) %>%
  count(carat)
```

```
## # A tibble: 2 x 2
##   carat     n
##   <dbl> <int>
## 1  0.99     23
## 2    1    1558
```

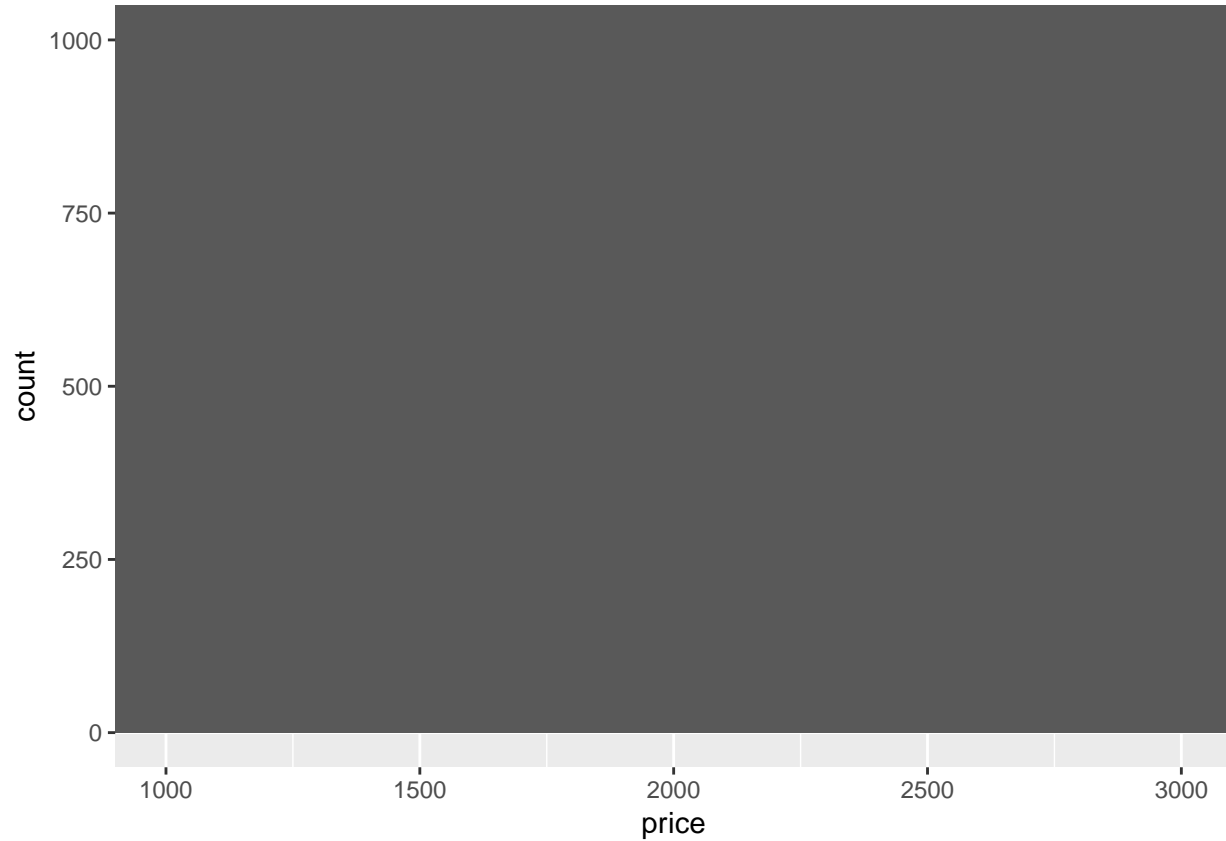
These results imply that many diamonds are rounded up.

4

Compare and contrast `coord_cartesian()` vs `xlim()` or `ylim()` when zooming in on a histogram. What happens if you leave `binwidth` unset? What happens if you try and zoom so only half a bar shows?

```
diamonds %>% ggplot() +
  geom_histogram(mapping = aes(x = price)) +
  coord_cartesian(xlim = c(1000,3000), ylim = c(0, 1000))
```

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

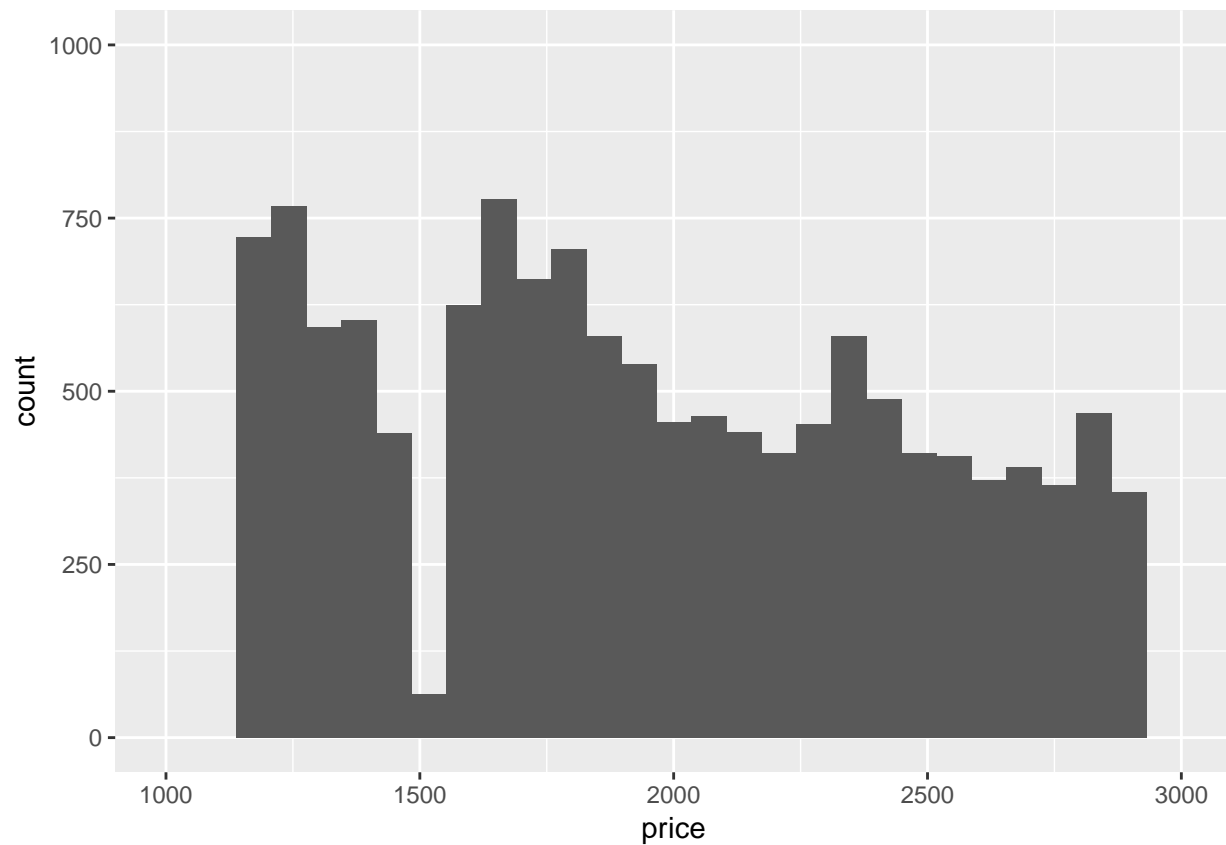


```
diamonds %>% ggplot() +
  geom_histogram(aes(x=price)) +
  xlim(1000,3000)+
  ylim(0,1000)
```

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

```
## Warning: Removed 38103 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



### 7.4.1

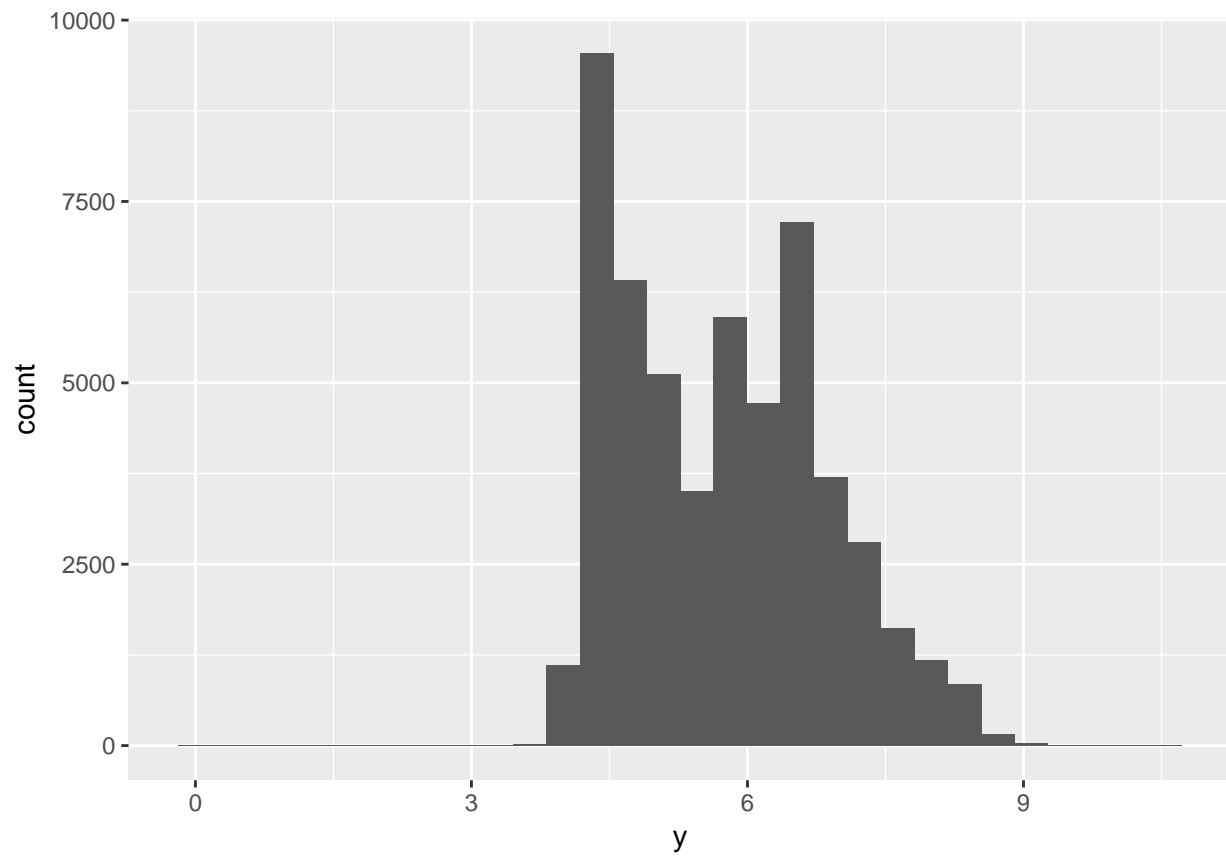
1

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

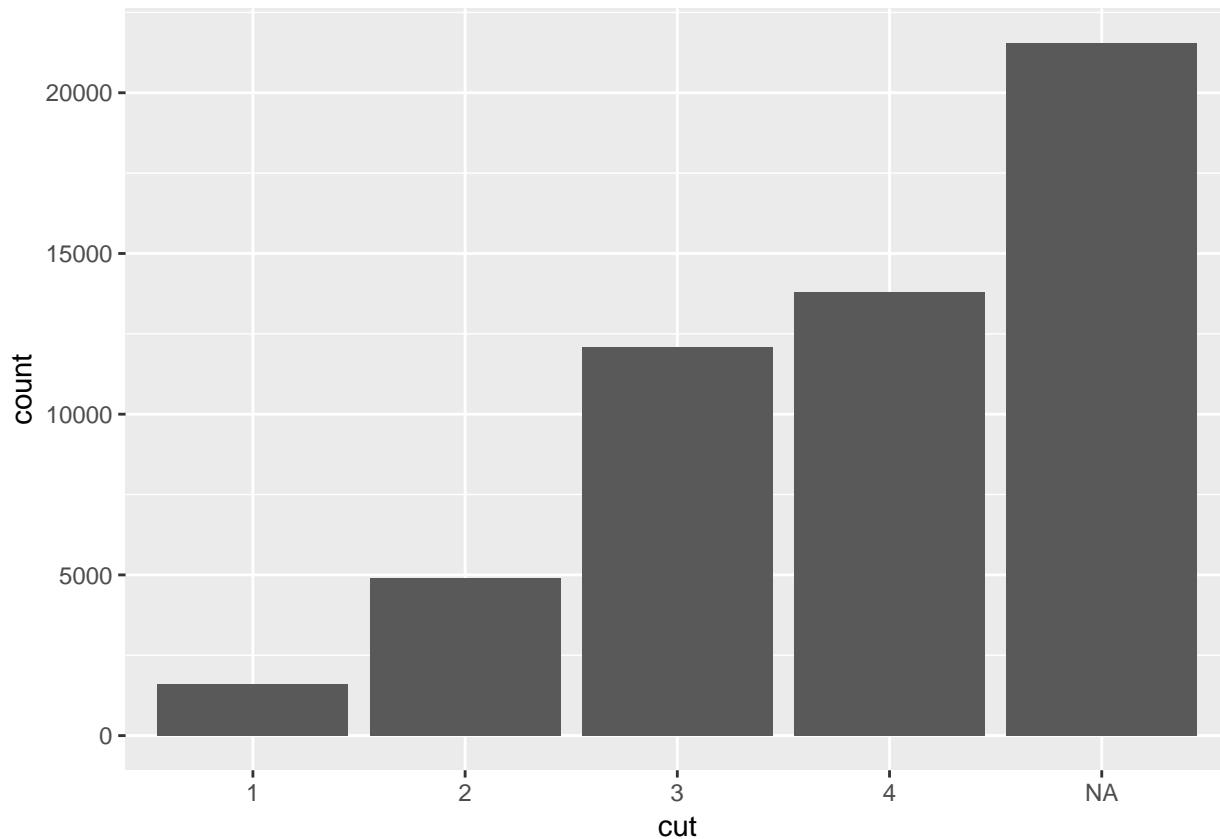
```
diamonds %>% mutate(y = ifelse(y > 20, NA_real_, y)) %>%
  ggplot(aes(x = y)) +
  geom_histogram()
```

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

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```



```
diamonds %>% mutate(cut = ifelse(cut=="Ideal", NA_character_, cut)) %>%  
  ggplot(aes(x = cut)) +  
  geom_bar()
```



2

What does `na.rm = TRUE` do in `mean()` and `sum()`?

```
mean(c(0, 1, 2, NA), na.rm = TRUE)
```

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

```
sum(c(0, 1, 2, NA), na.rm = TRUE)
```

```
## [1] 3
```

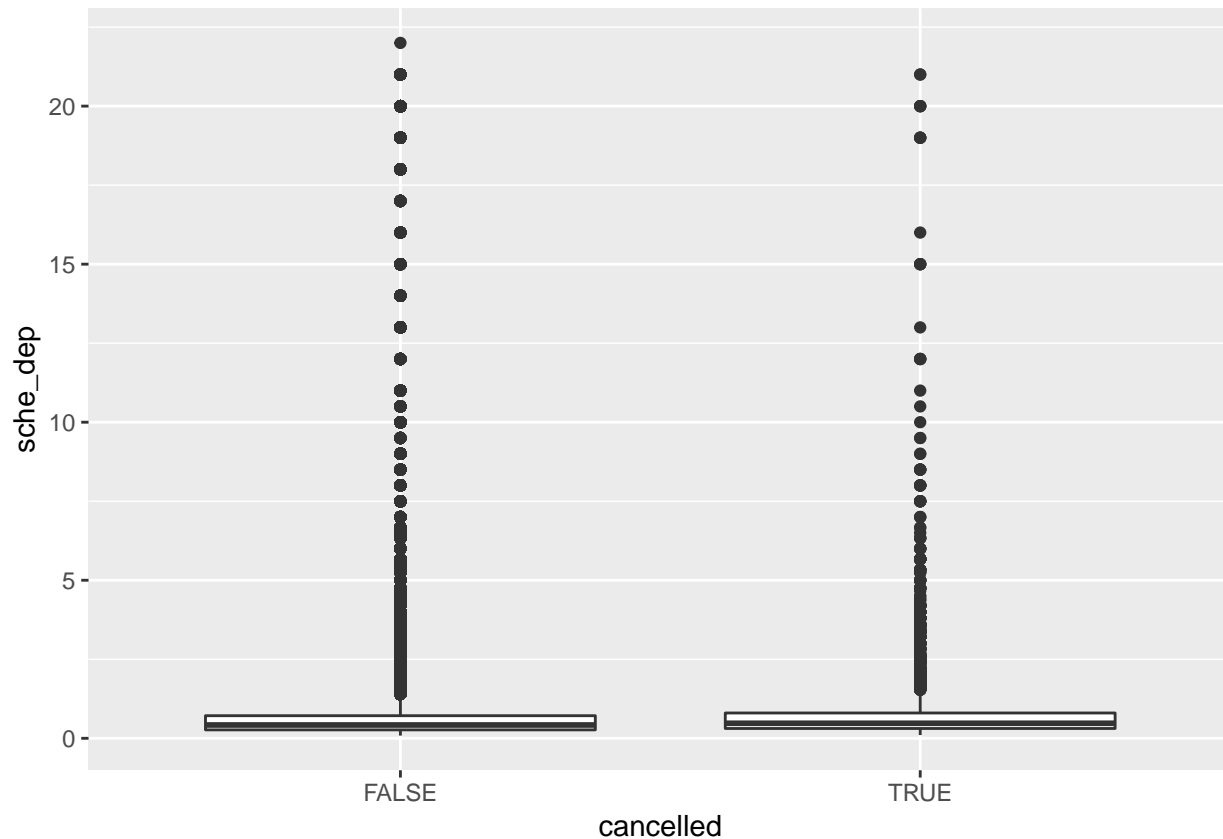
## 7.5.1

1

Use what you've learned to improve the visualization of the departure times of cancelled vs. non-cancelled flights.

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

```
## Warning: Removed 60696 rows containing non-finite values (stat_boxplot).
```



2

What variable in the diamonds dataset is most important for predicting the price of a diamond?  
 How is that variable correlated with cut? Why does the combination of those two relationships  
 lead to lower quality diamonds being more expensive?

```
broom::tidy(glm(diamonds$price ~ diamonds$carat + diamonds$cut + diamonds$color + diamonds$clarity))
```

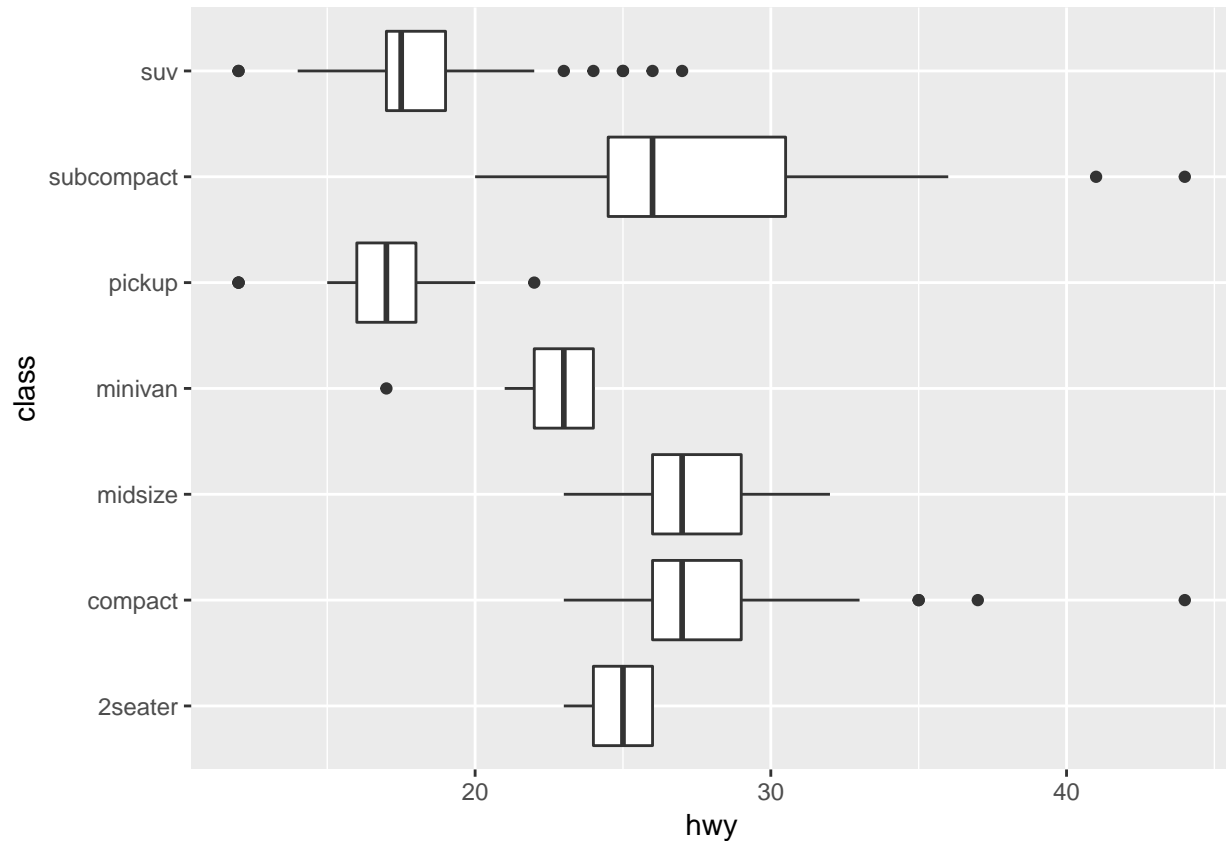
```
## # A tibble: 19 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)       -3711.    14.0   -265.     0.
## 2 diamonds$carat     8886.    12.0    738.     0.
## 3 diamonds$cut.L       699.    20.3    34.4  4.29e-256
## 4 diamonds$cut.Q     -328.    17.9   -18.3  1.52e- 74
## 5 diamonds$cut.C      181.    15.6    11.6  4.13e- 31
## 6 diamonds$cut^4      -1.21    12.5   -0.0969 9.23e- 1
## 7 diamonds$color.L   -1910.    17.7  -108.     0.
## 8 diamonds$color.Q    -628.    16.1   -39.0     0.
## 9 diamonds$color.C    -172.    15.1   -11.4  4.01e- 30
## 10 diamonds$color^4    21.7    13.8    1.57  1.17e- 1
## 11 diamonds$color^5   -85.9    13.1   -6.57  5.00e- 11
## 12 diamonds$color^6   -50.0    11.9   -4.20  2.62e- 5
## 13 diamonds$clarity.L  4218.    30.8   137.     0.
## 14 diamonds$clarity.Q -1832.    28.8   -63.6     0.
## 15 diamonds$clarity.C   923.    24.7    37.4  2.00e-302
## 16 diamonds$clarity^4 -362.    19.7   -18.3  6.82e- 75
## 17 diamonds$clarity^5  217.    16.1    13.4  3.76e- 41
```

```
## 18 diamonds$clarity^6      2.11      14.0      0.150 8.81e- 1
## 19 diamonds$clarity^7     110.       12.4      8.91  5.24e- 19
```

### 3

Install the ggstance package, and create a horizontal box plot. How does this compare to using `coord_flip()`?

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



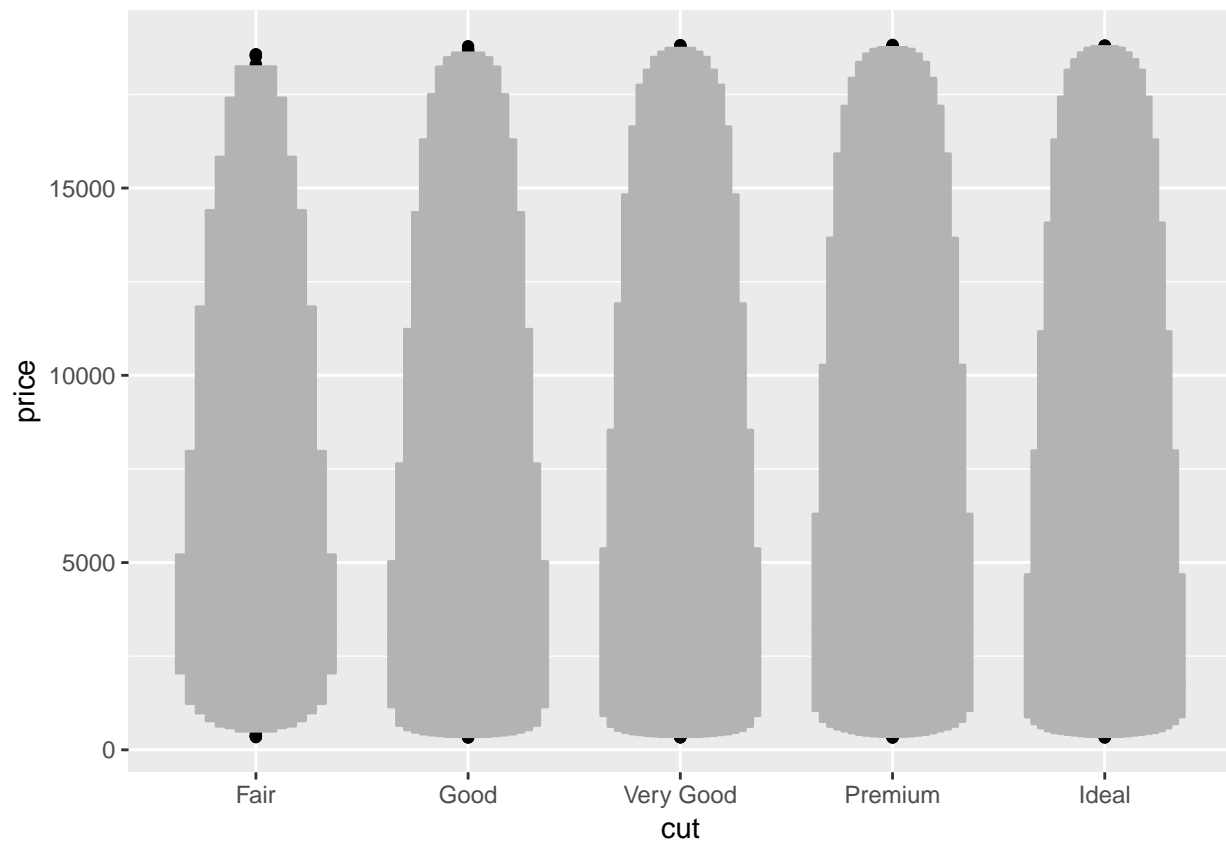
### 4

One problem with box plots is that they were developed in an era of much smaller datasets and tend to display a prohibitively large number of “outlying values”. One approach to remedy this problem is the letter value plot. Install the lvplot package, and try using `geom_lv()` to display the distribution of price vs cut. What do you learn? How do you interpret the plots?

```
library(lvplot)

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

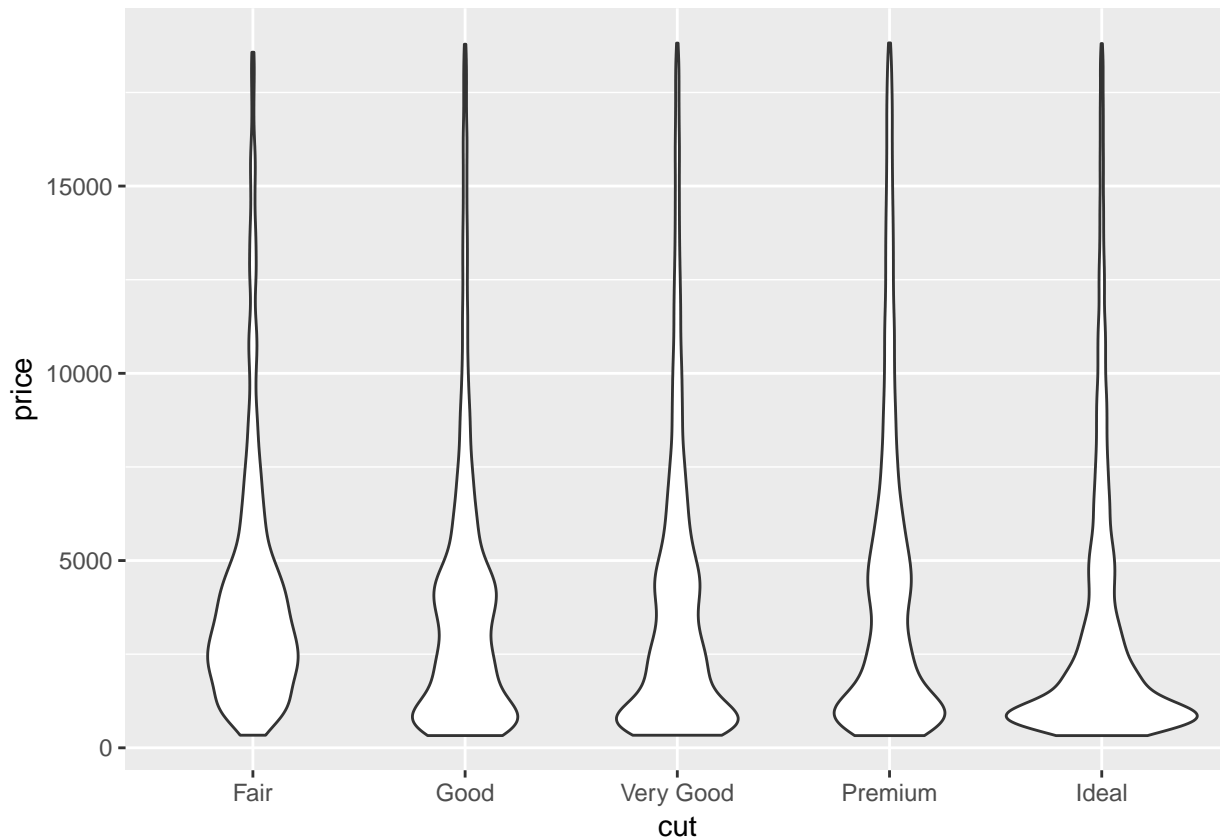




5

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

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



`geom_violin` has a merit that distribution is easy to understand.

## 6

If you have a small dataset, it's sometimes useful to use `geom_jitter()` to see the relationship between a continuous and categorical variable. The `ggbeeswarm` package provides a number of methods similar to `geom_jitter()`. List them and briefly describe what each one does.

## 7.5.2

### 1

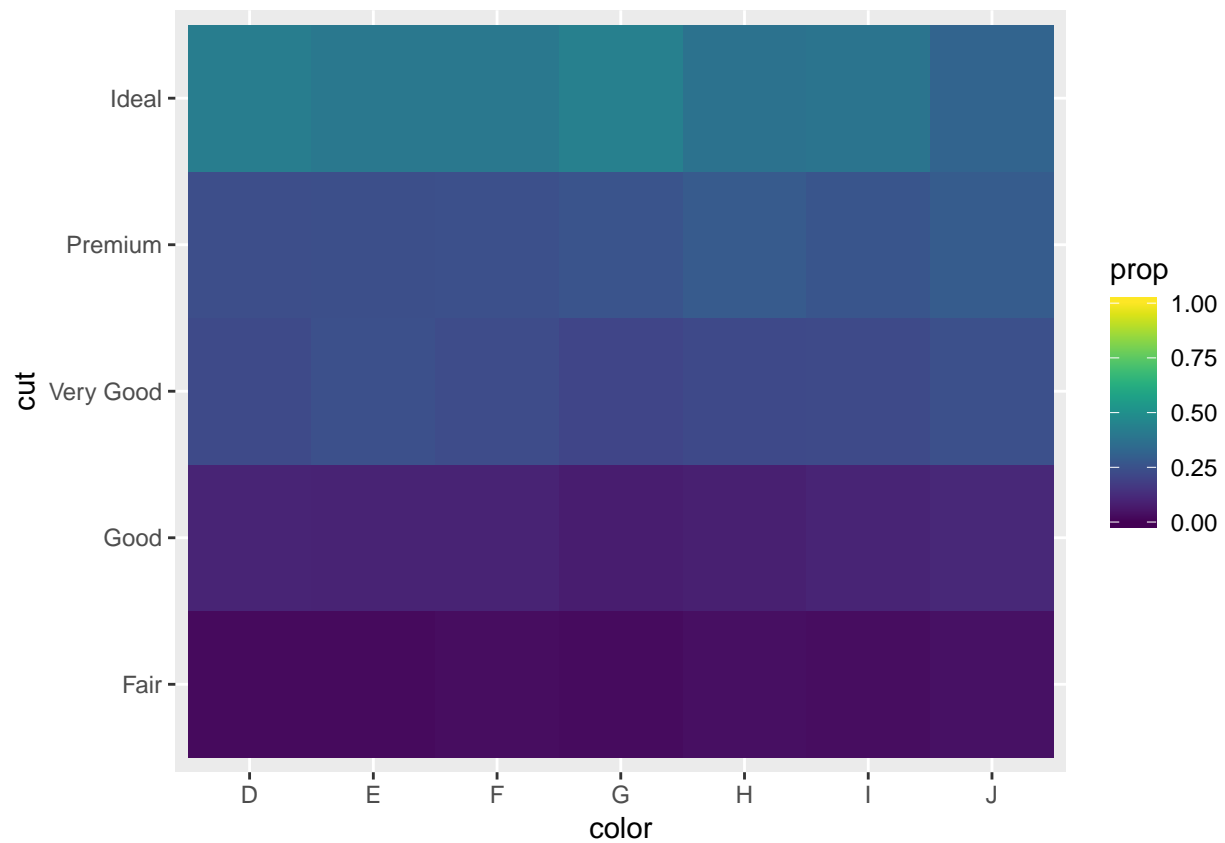
How could you rescale the count dataset above to more clearly show the distribution of cut within color, or color within cut?

cut within color

```
library(viridis)
```

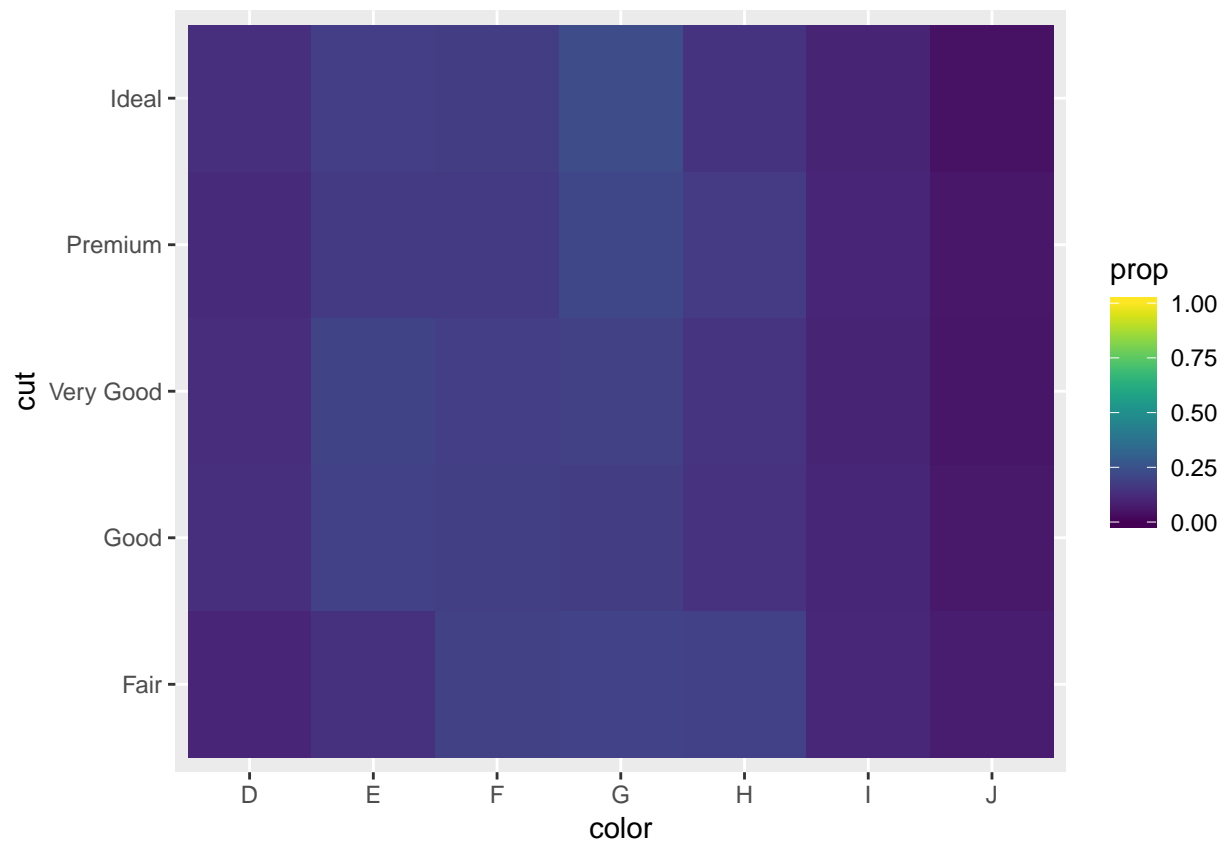
```
## Loading required package: viridisLite
```

```
diamonds %>% count(color, cut) %>%
  group_by(color) %>%
  mutate(prop=n/sum(n)) %>%
  ggplot()+
  geom_tile(aes(x=color, y=cut, fill=prop))+
  scale_fill_viridis(limits=c(0,1))
```



color within cut

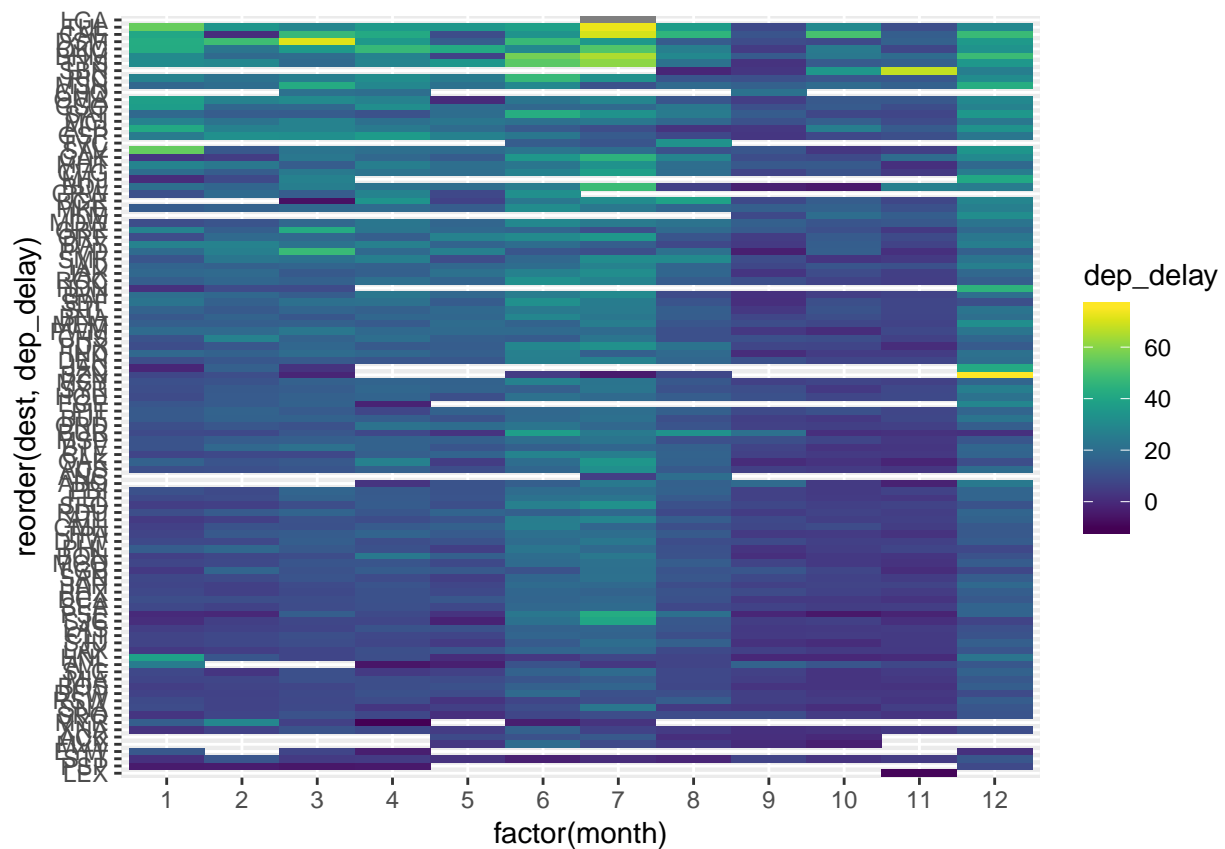
```
diamonds %>% count(color, cut) %>%
  group_by(cut) %>%
  mutate(prop=n/sum(n)) %>%
  ggplot()+
  geom_tile(aes(x=color, y=cut, fill=prop))+
  scale_fill_viridis(limits=c(0,1))
```



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?

```
nycflights13::flights %>%
  group_by(month, dest) %>%
  summarize(dep_delay = mean(dep_delay, na.rm = TRUE)) %>%
  ggplot(aes(x=factor(month), y=reorder(dest, dep_delay), fill=dep_delay)) +
  geom_tile() +
  scale_fill_viridis()
```

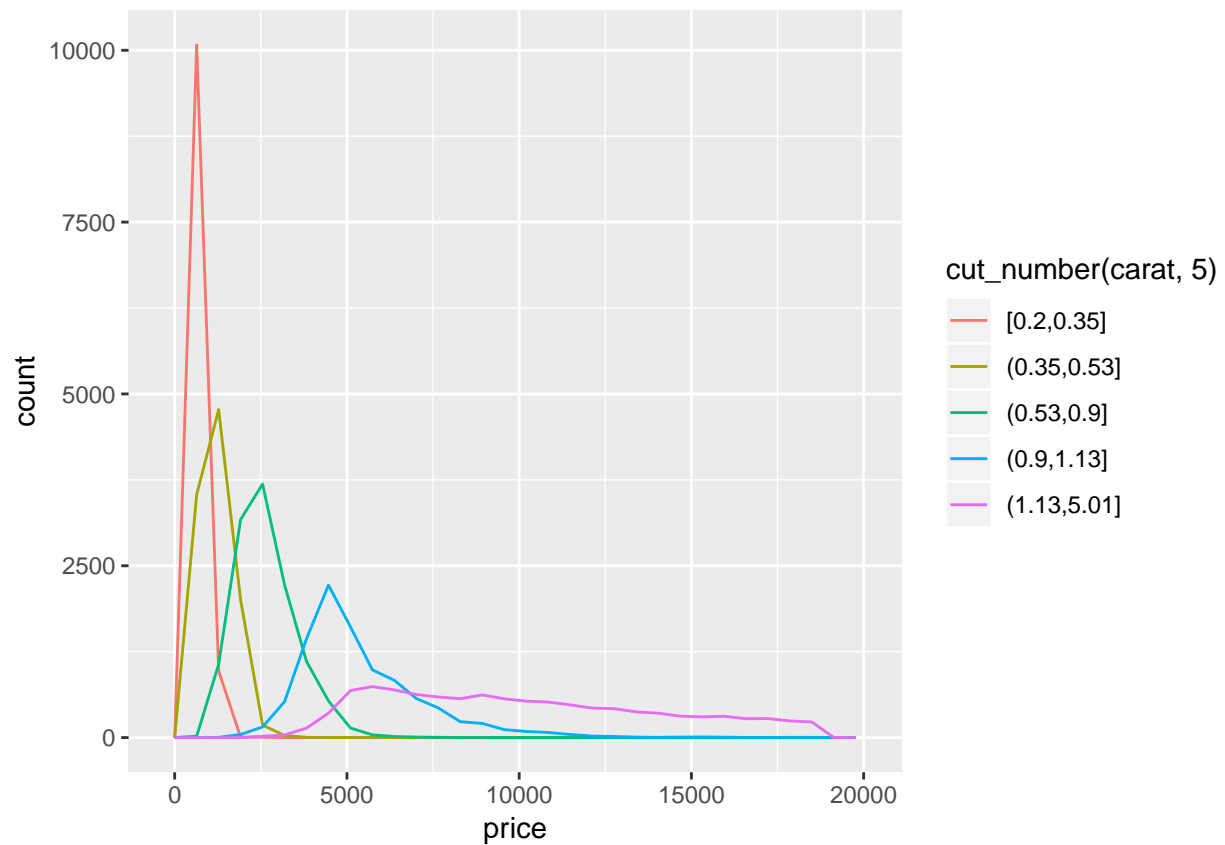


7.5.3 ## 1 > > Why is it slightly better to use `aes(x = color, y = cut)` rather than `aes(x = cut, y = color)` in the example above? >

cut\_number

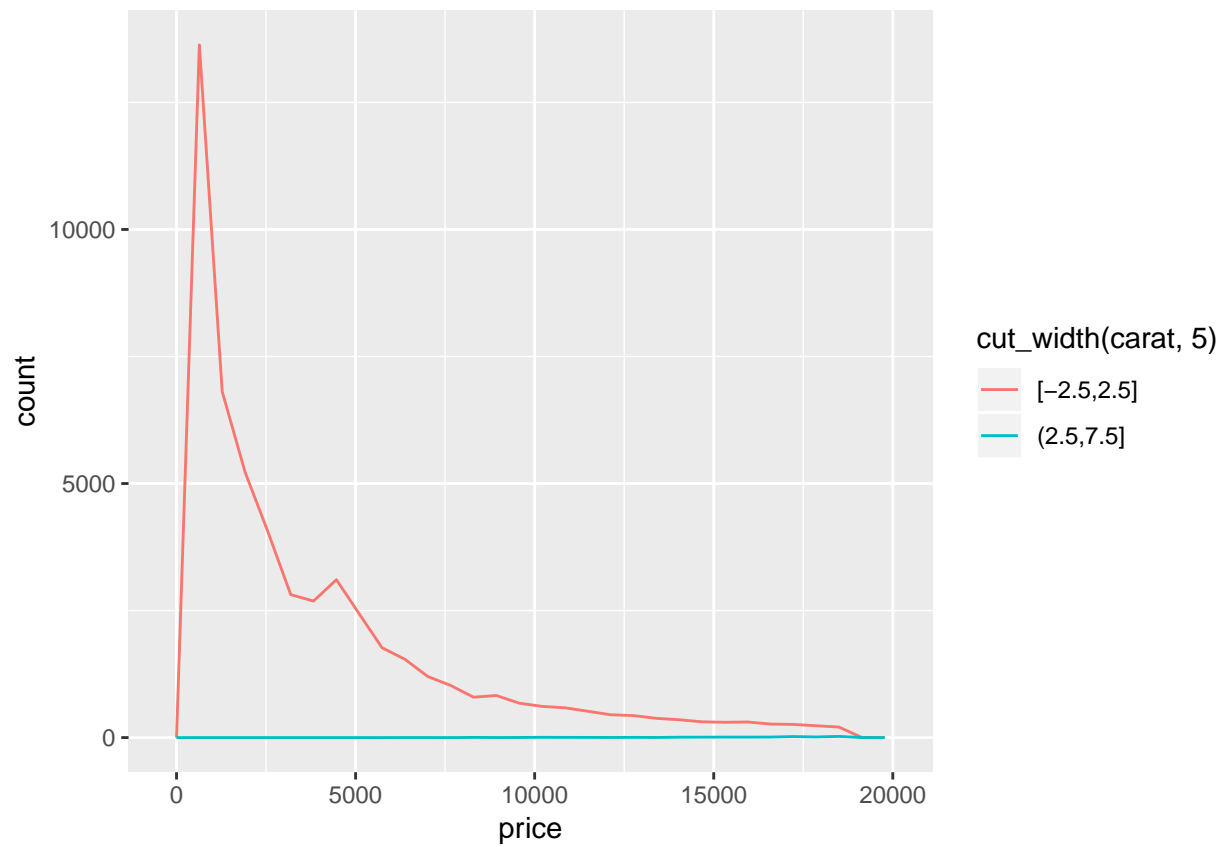
```
diamonds %>% ggplot(aes(color=cut_number(carat, 5), x=price)) +
  geom_freqpoly()
```

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



```
cut_width
diamonds %>% ggplot(aes(color=cut_width(carat, 5), x=price)) +
  geom_freqpoly()

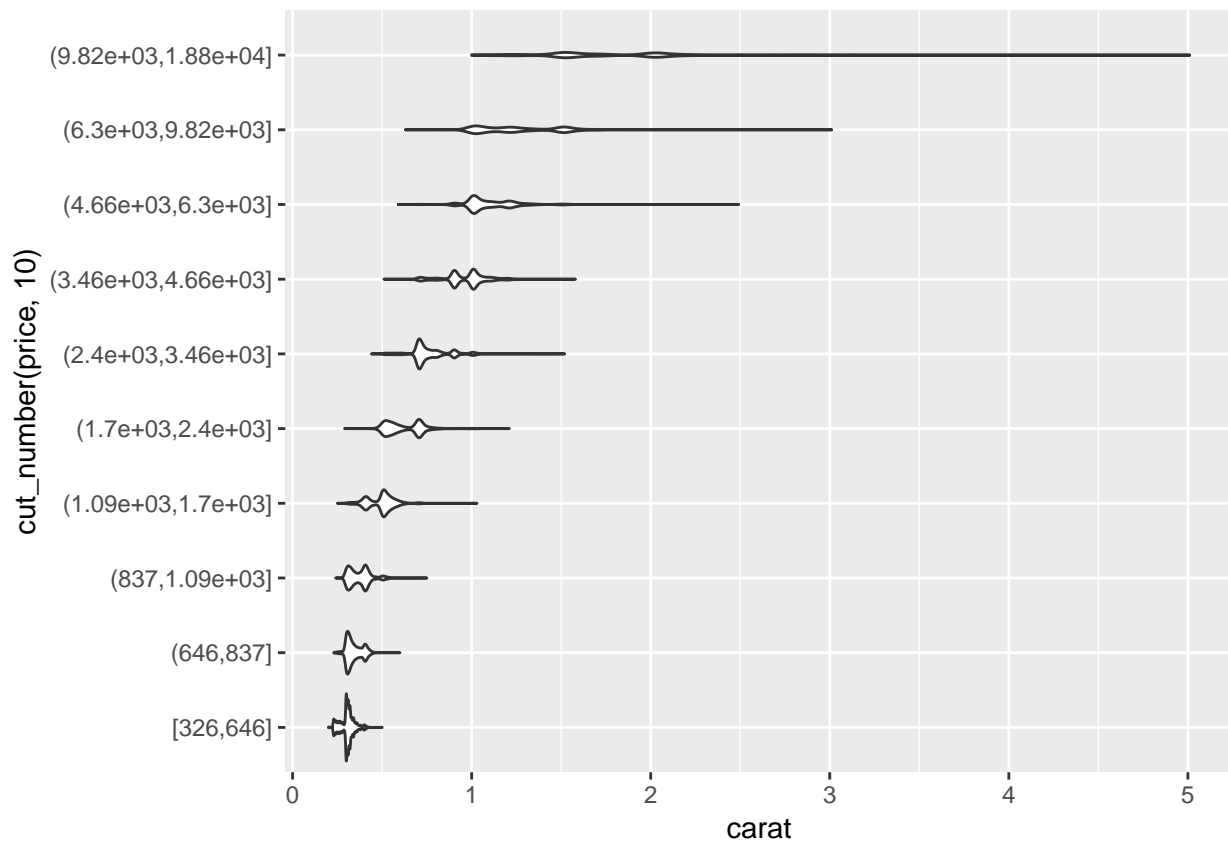
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2

Visualize the distribution of carat, partitioned by price.

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



3

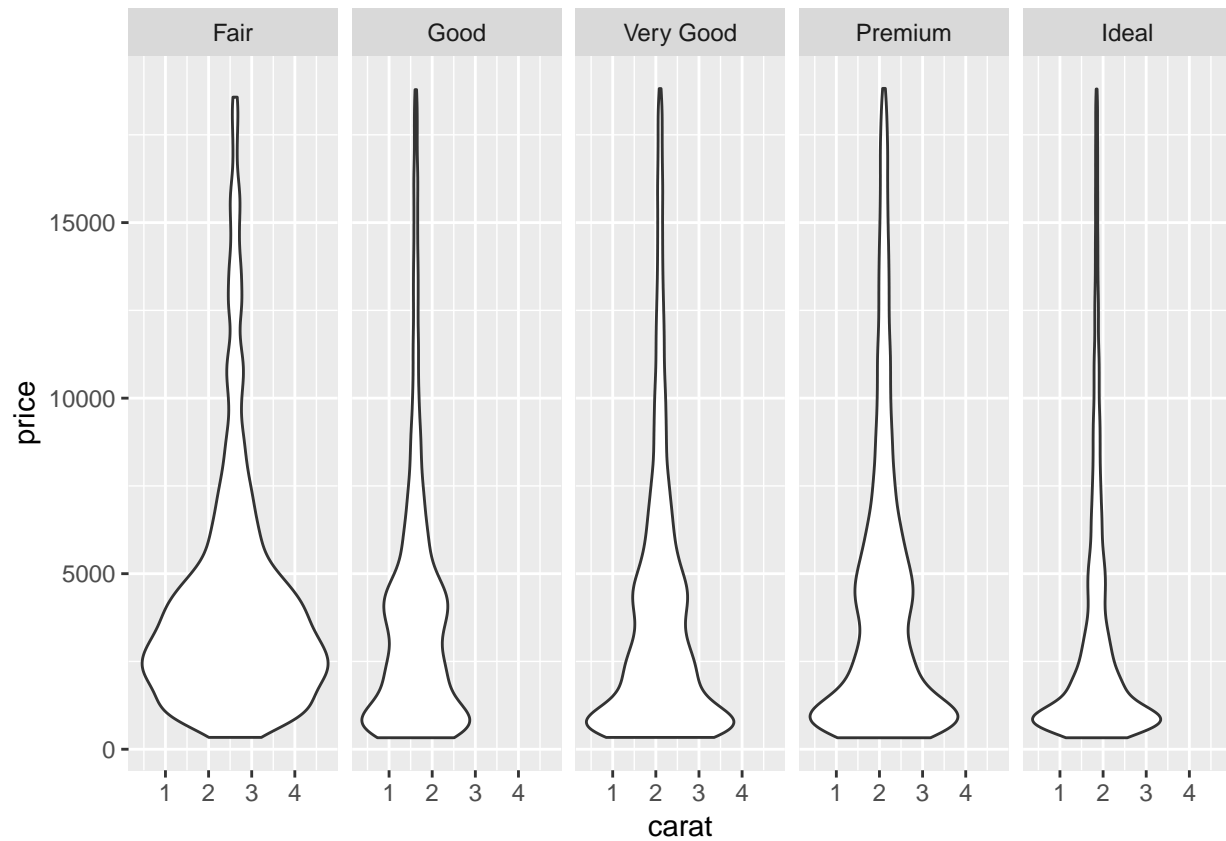
How does the price distribution of very large diamonds compare to small diamonds. Is it as you expect, or does it surprise you?

4

Combine two of the techniques you've learned to visualize the combined distribution of cut, carat, and price.

```
diamonds %>% ggplot(aes(x=carat, y=price)) +  
  geom_violin() +  
  facet_grid(~cut)
```

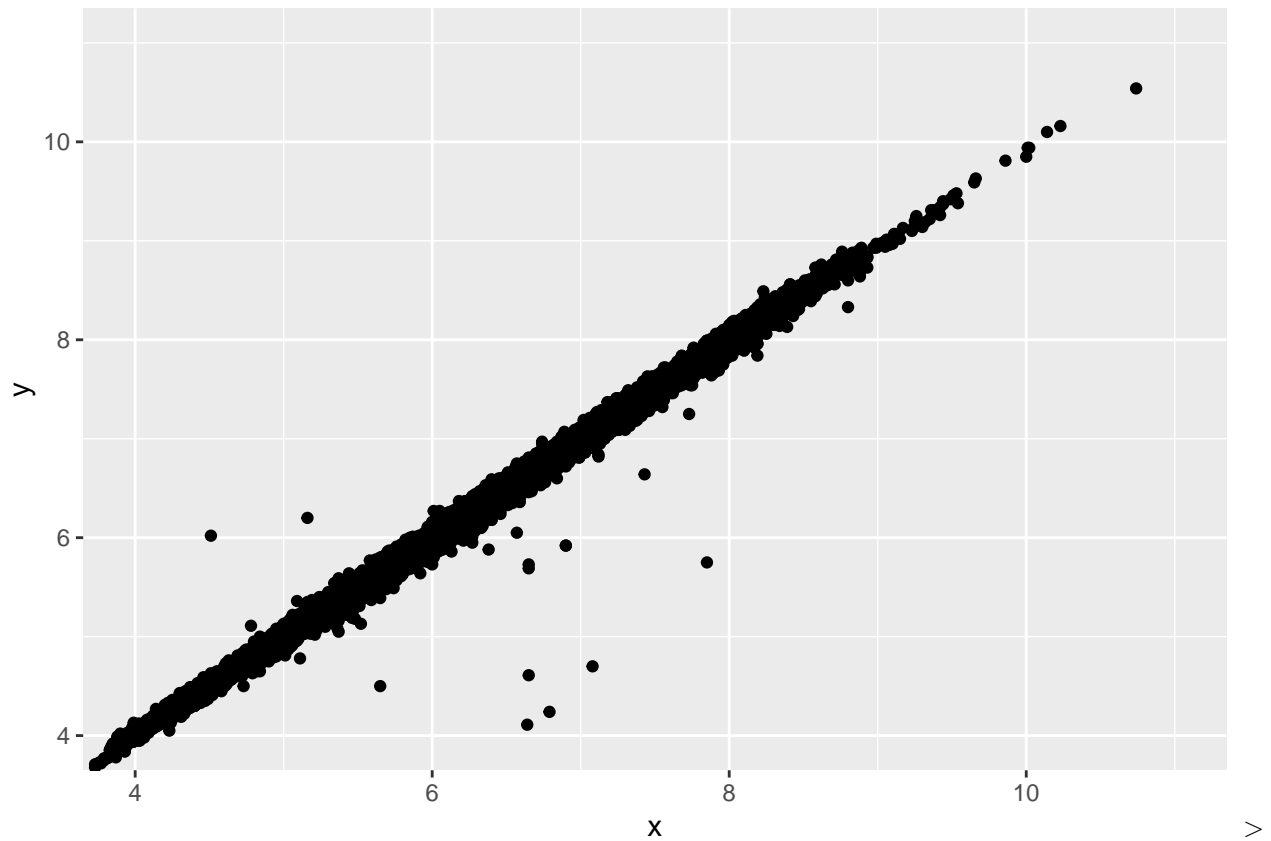




## 5

Two dimensional plots reveal outliers that are not visible in one dimensional plots. For example, some points in the plot below have an unusual combination of x and y values, which makes the points outliers even though their x and y values appear normal when examined separately.

```
ggplot(data = diamonds) +  
  geom_point(mapping = aes(x = x, y = y)) +  
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



> Why is a scatterplot a better display than a binned plot for this case? >

If there is a strong relationships between x and y, then you should use scatterplot.