Stat. 450 Section 1 or 2: Homework 5

Prof. Eric A. Suess

So how should you complete your homework for this class?

- First thing to do is type all of your information about the problems you do in the text part of your R Notebook.
- Second thing to do is type all of your R code into R chunks that can be run.
- If you load the tidyverse in an R Notebook chunk, be sure to include the "message = FALSE" in the {r}, so {r message = FALSE}.
- Last thing is to spell check your R Notebook. Edit > Check Spelling... or hit the F7 key.

Homework 5:

```
 Read: Chapter 7
 Do 7.3.4 Exercises 1, 2, 3, 4
 Do 7.4.1 Exercises 1, 2
 Do 7.5.1.1 Exercises 2, 3, 4, 5, 6
```

```r
library(tidyverse)
```

7.3.4

1.

All are skewed to the right. The distributions of x and y are very similar. The distributions of z looks to be less spread out. All look to be bimodal.

I think x and y are the length and width, and z is the depth.

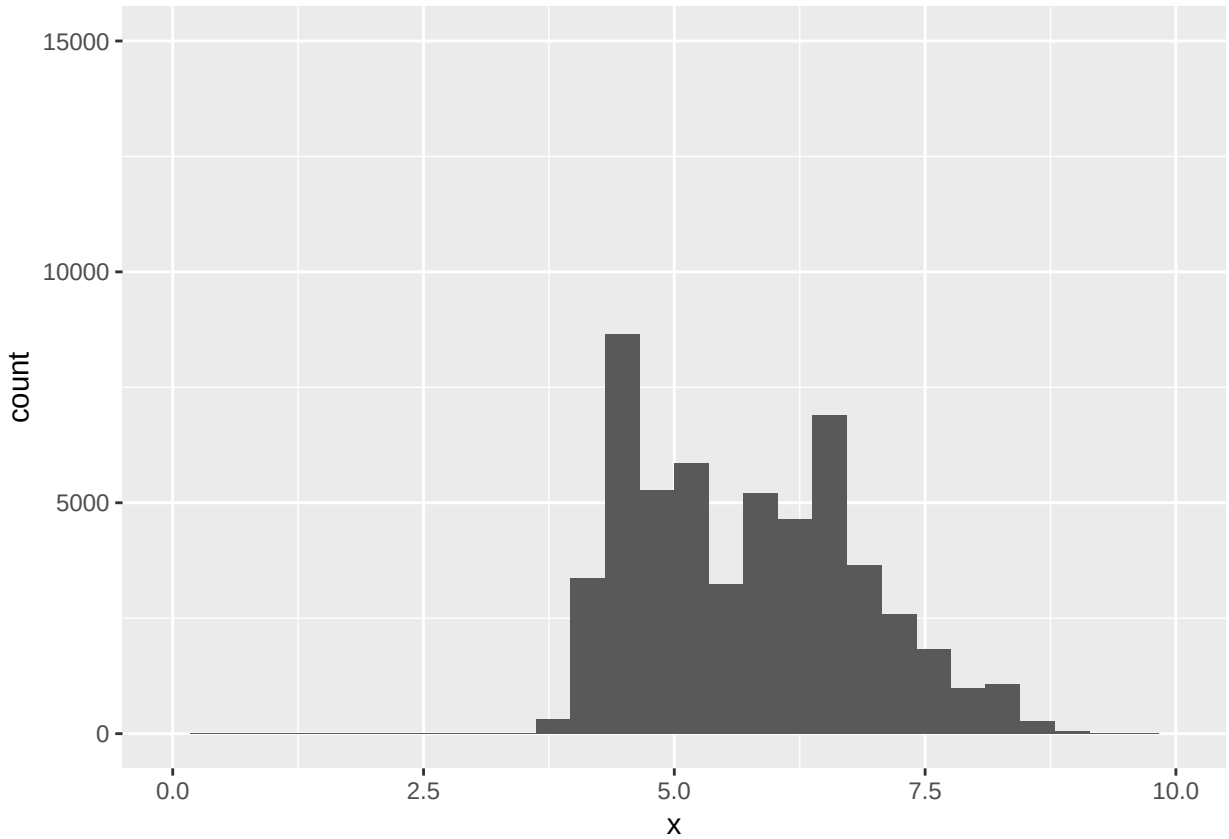
```r
diamonds
```

```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
##  3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
##  4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
##  5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
##  6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
##  7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
##  8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
##  9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

```r
diamonds %>% select(x,y,z) %>%
  ggplot(aes(x = x )) +
  geom_histogram() +
  scale_x_continuous(limits=c(0, 10)) +
  scale_y_continuous(limits=c(0, 15000))
```

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

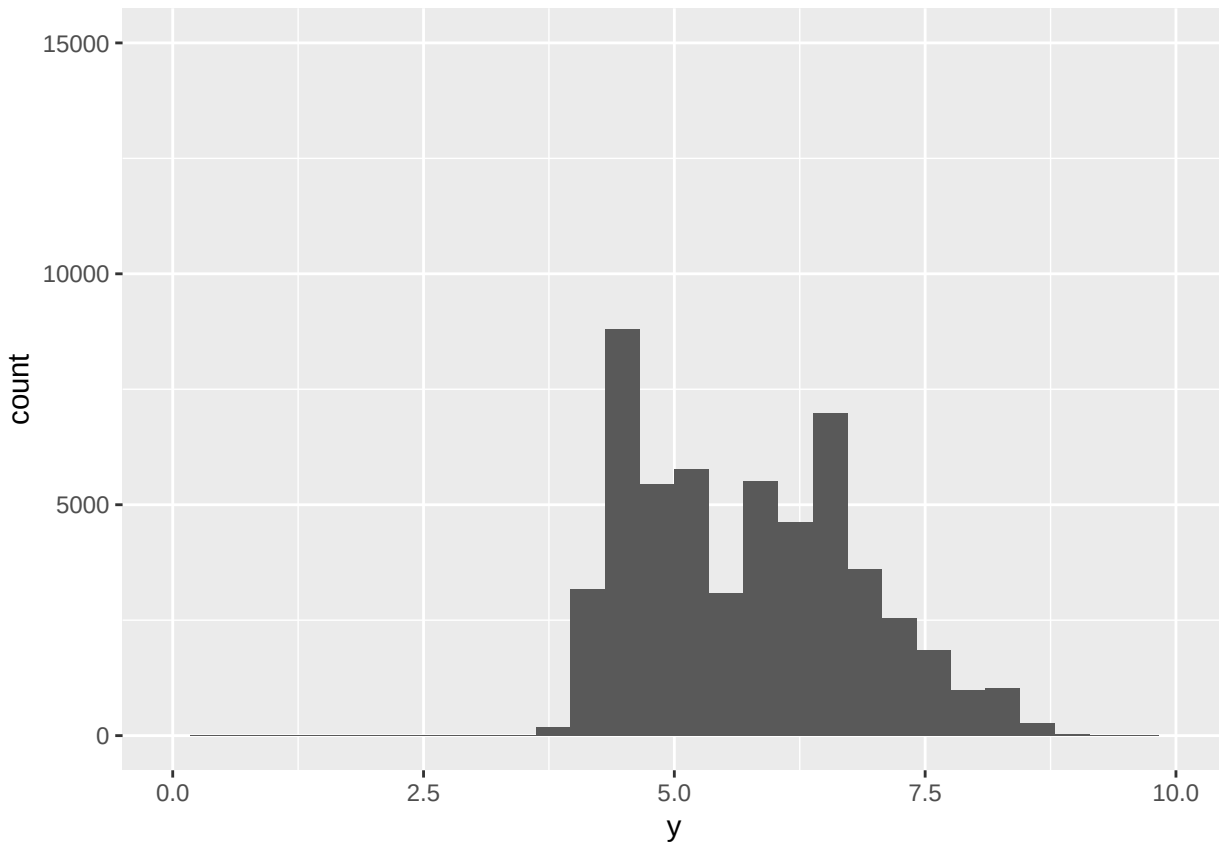
```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```




```
diamonds %>% select(x,y,z) %>%
  ggplot(aes(x = y )) +
  geom_histogram() +
  scale_x_continuous(limits=c(0, 10)) +
  scale_y_continuous(limits=c(0, 15000))
```

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

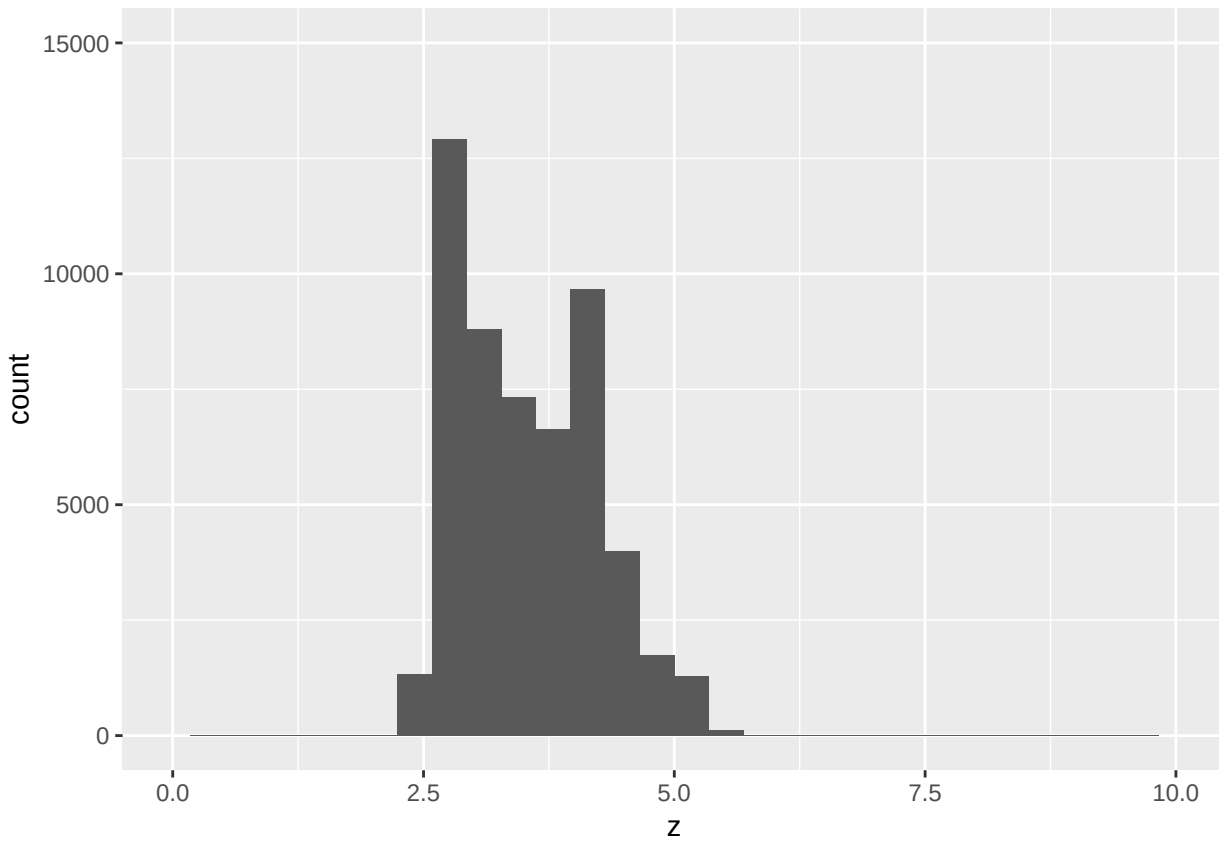
```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```




```
diamonds %>% select(x,y,z) %>%
  ggplot(aes(x = z )) +
  geom_histogram() +
  scale_x_continuous(limits=c(0, 10)) +
  scale_y_continuous(limits=c(0, 15000))
```

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

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



2.

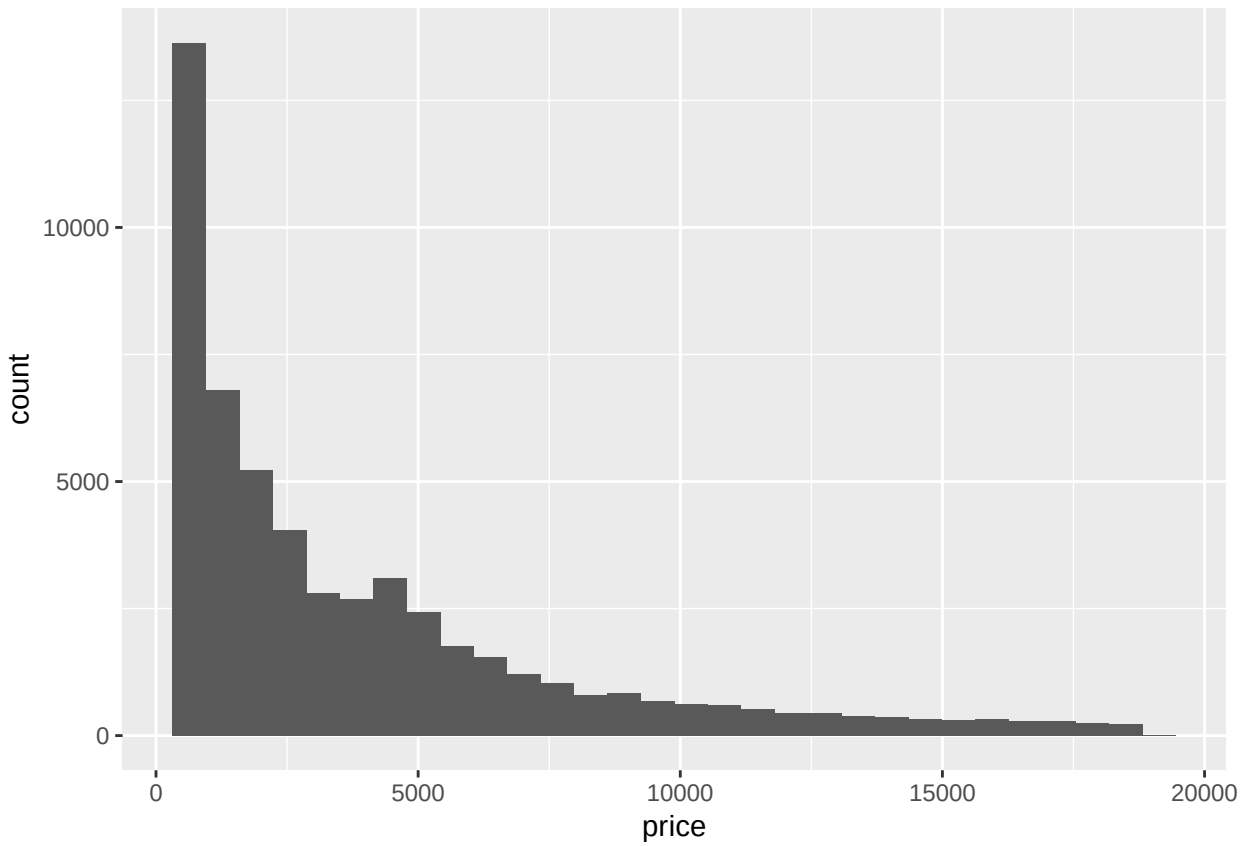
There are not prices around $1500.

The mode of the distributions is around $750.

At the lowest levels there are spikes in the prices where the diamonds are actually priced.

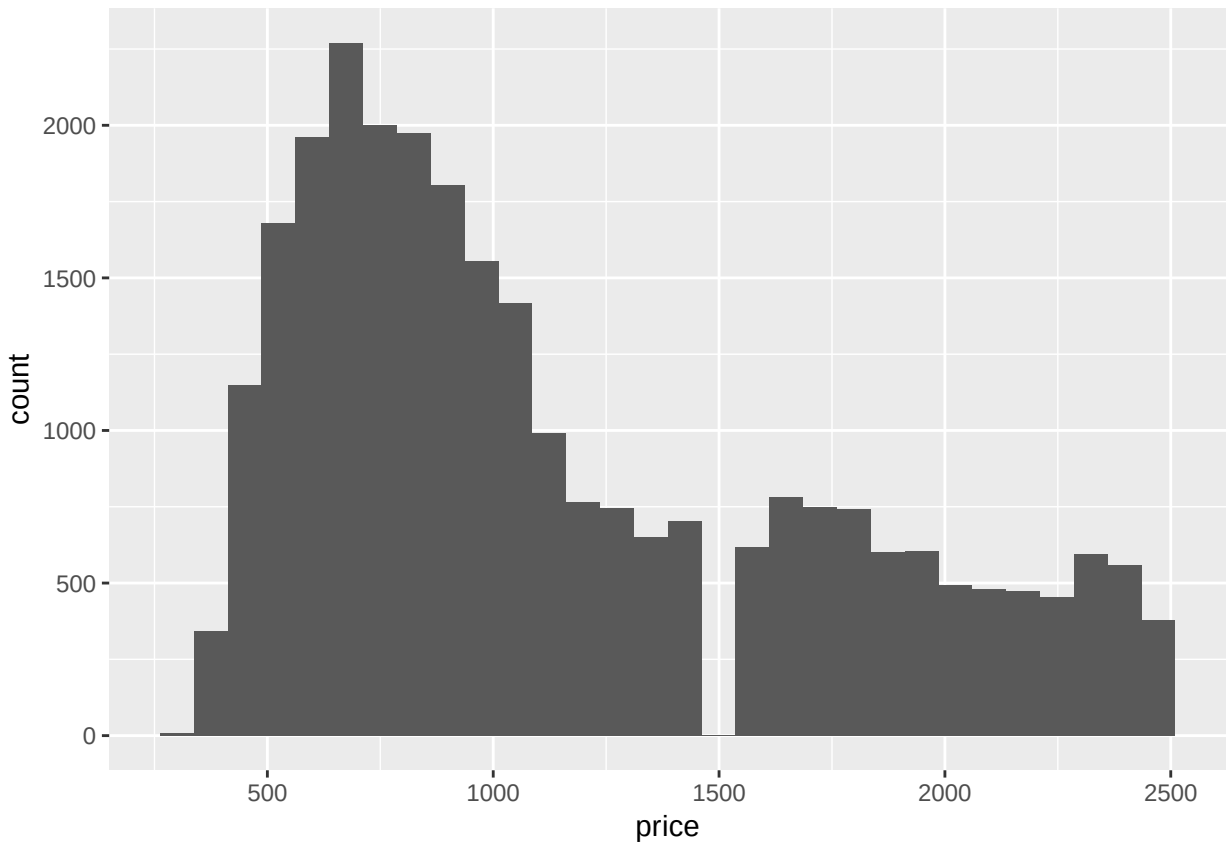
```
diamonds %>% select(price) %>%
  ggplot(aes(x = price )) +
  geom_histogram()
```

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

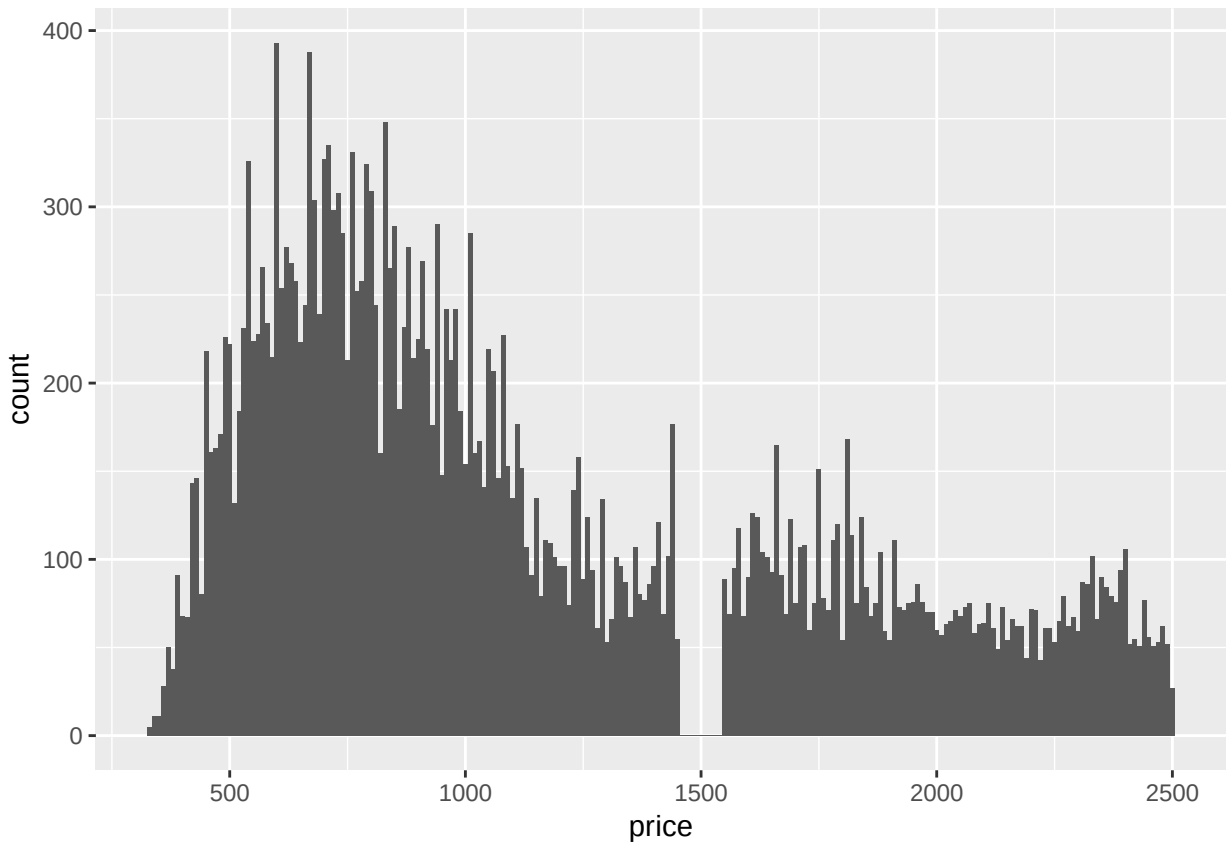



```
diamonds %>% select( price ) %>%
  filter(price < 2500) %>%
  ggplot(aes(x = price )) +
  geom_histogram()
```

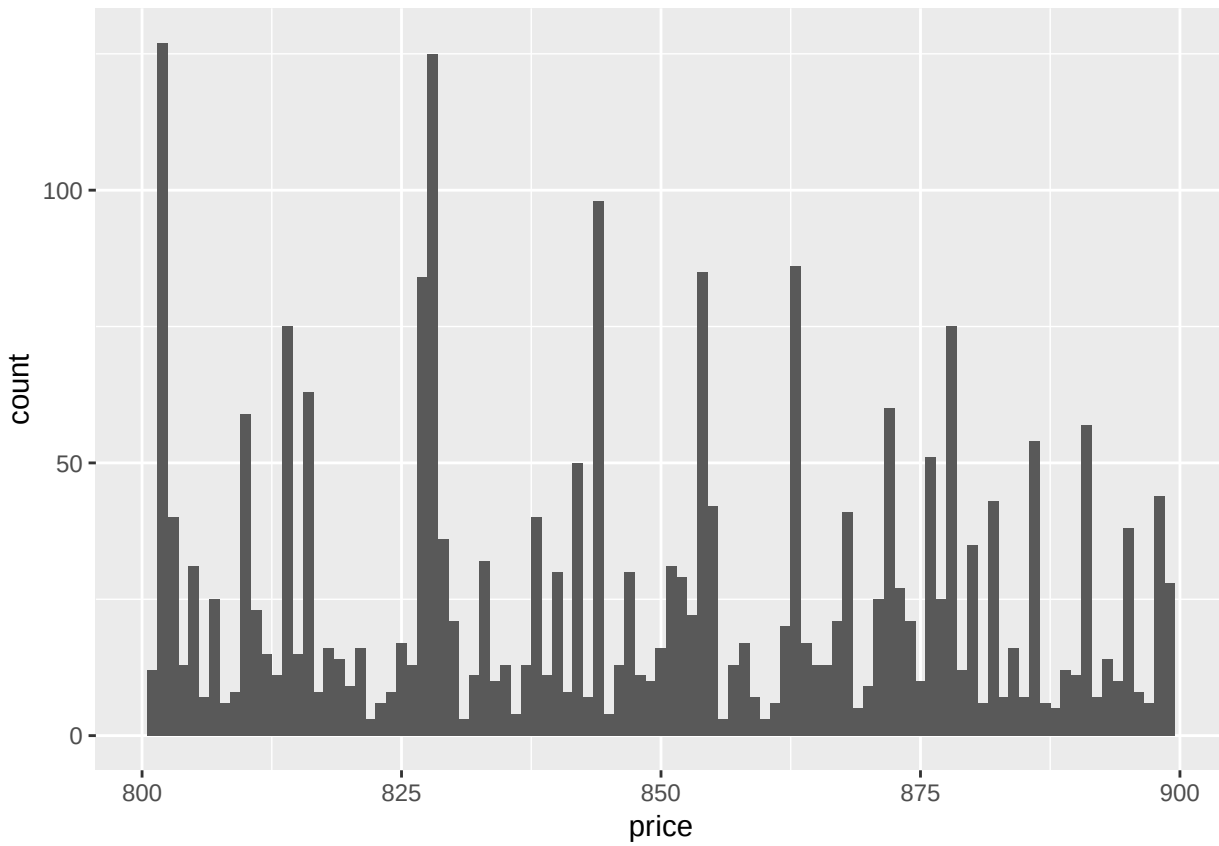
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
diamonds %>% select( price ) %>%
  filter(price < 2500) %>%
  ggplot(aes(x = price )) +
  geom_histogram(binwidth = 10, center = 0 )
```




```
diamonds %>% select( price ) %>%
  filter(price < 900, price > 800) %>%
  ggplot(aes(x = price )) +
  geom_histogram(binwidth = 01, center = 0 )
```



3.

There are 23 diamonds that are .99 carats and there are 1558 diamonds that are 1 carat. Rounding up is worth more money.

```
diamonds %>% select(carat) %>%
  count(carat == 0.99)
```

```
## # A tibble: 2 x 2
##   `carat == 0.99`     n
##   <lgl>           <int>
## 1 FALSE           53917
## 2 TRUE               23
```

```
diamonds %>% select(carat) %>%
  count(carat == 1)
```

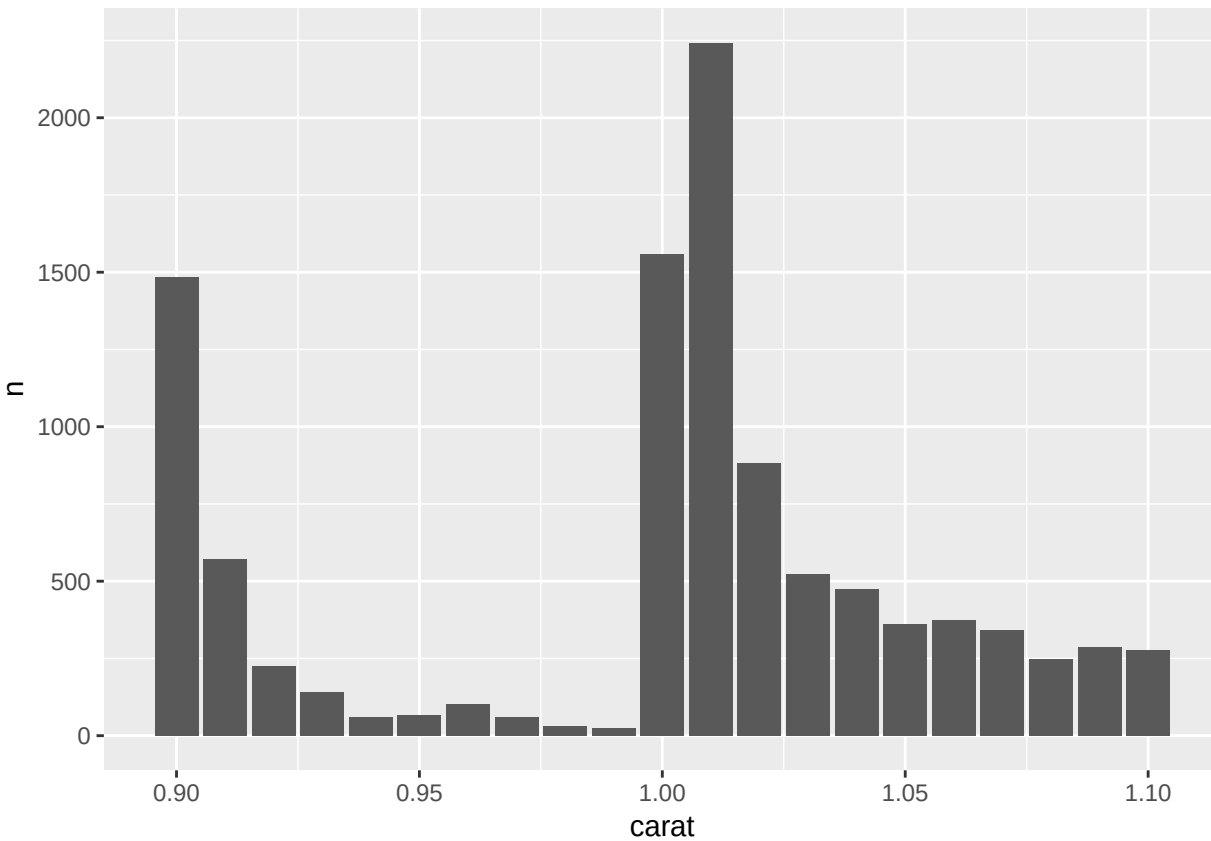
```
## # A tibble: 2 x 2
##   `carat == 1`     n
##   <lgl>        <int>
## 1 FALSE        52382
## 2 TRUE          1558
```

```
diamonds %>%
   filter(carat >= 0.9, carat <= 1.1) %>%
   count(carat)
```

```
## # A tibble: 21 x 2
```

```
##    carat     n
##    <dbl> <int>
##  1  0.9   1485
##  2  0.91   570
##  3  0.92   226
##  4  0.93   142
##  5  0.94    59
##  6  0.95    65
##  7  0.96   103
##  8  0.97    59
##  9  0.98    31
## 10  0.99    23
## # ... with 11 more rows
```

```
diamonds %>%
   filter(carat >= 0.9, carat <= 1.1) %>%
   count(carat) %>%
   ggplot(aes(x= carat, y = n)) +
   geom_col()
```



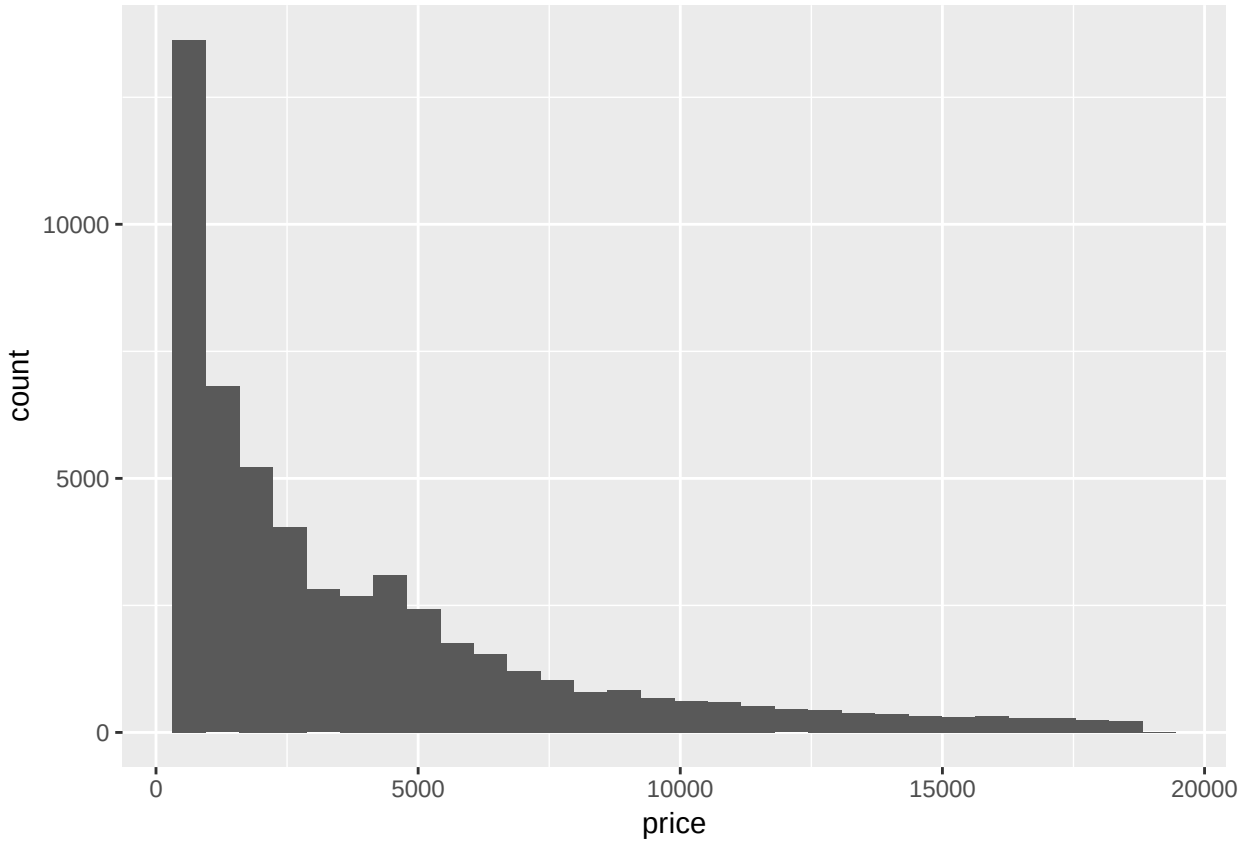
4.

The *cood_cartesian* function zooms in on the original histogram.

The *xlim* and *ylim* functions limits the range of the data before counting. So the histogram is made for a subset of the data.

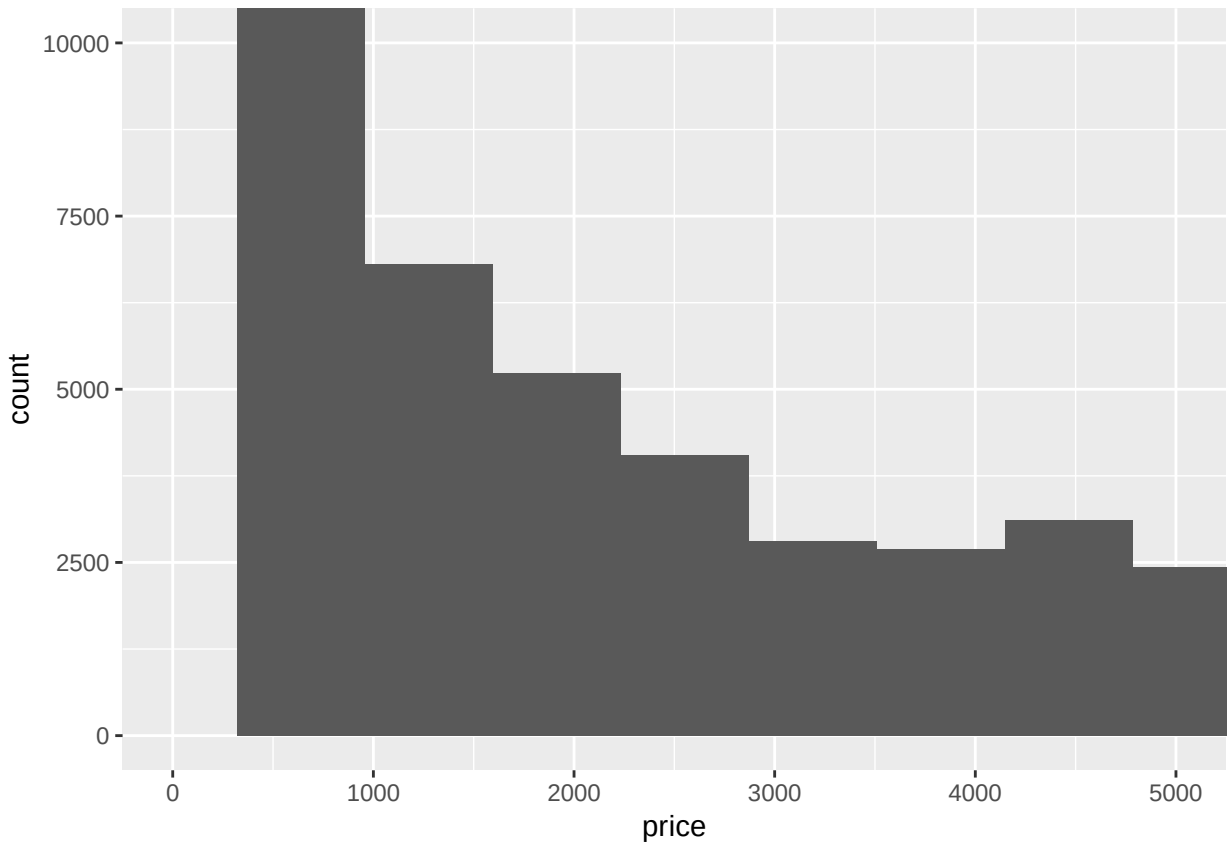
```r
diamonds %>% select(price) %>%
  ggplot(aes(x = price )) +
  geom_histogram()
```

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

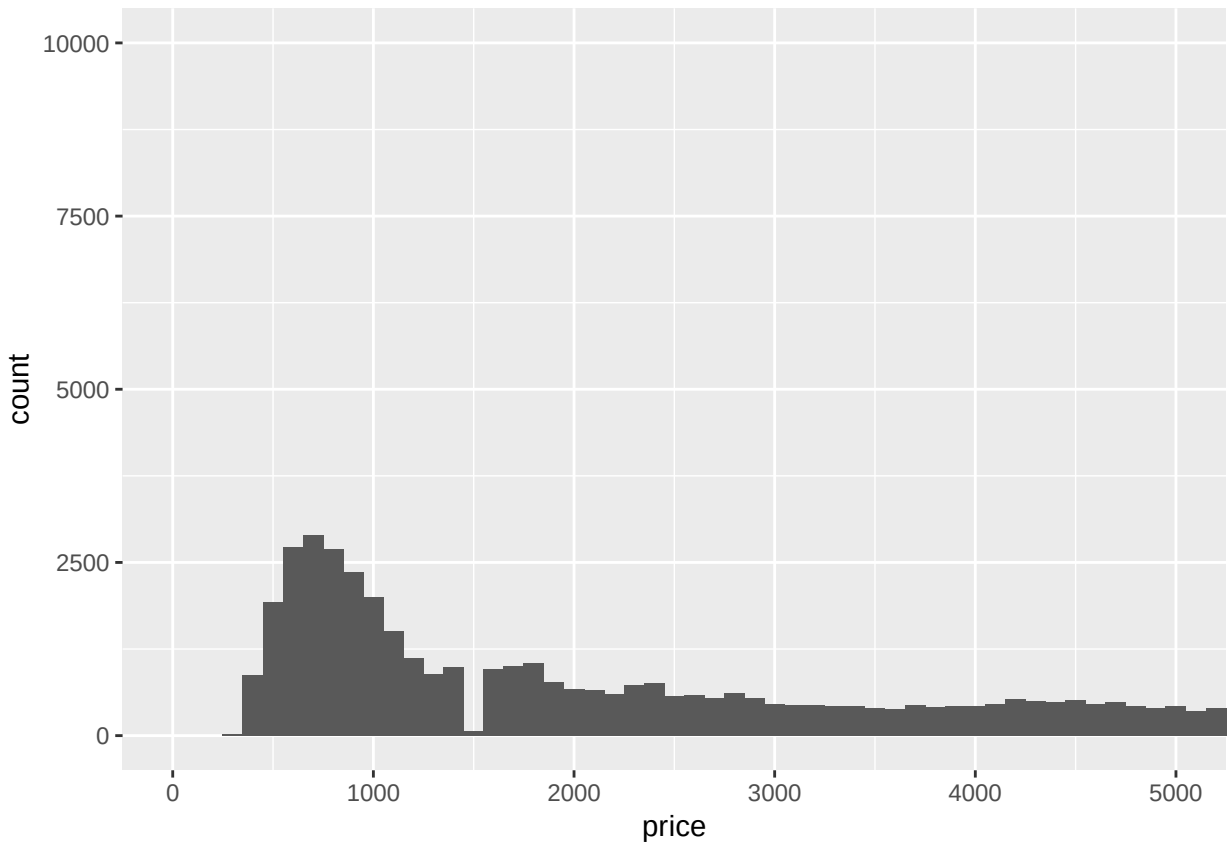



```r
diamonds %>% select(price) %>%
  ggplot(aes(x = price )) +
  geom_histogram() +
  coord_cartesian(xlim = c(0, 5000), ylim = c(0, 10000))
```

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

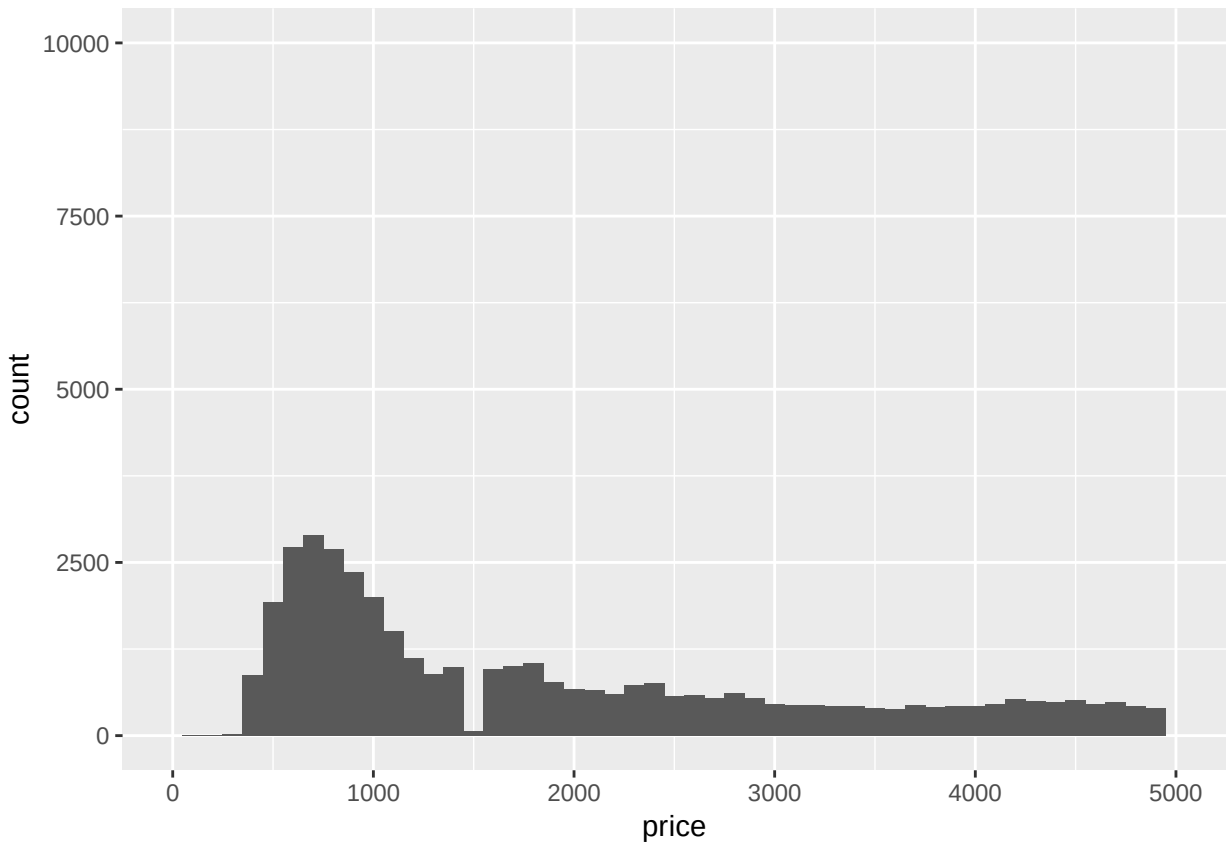



```
diamonds %>% select(price) %>%
  ggplot(aes(x = price )) +
  geom_histogram(binwidth = 100) +
  coord_cartesian(xlim = c(0, 5000), ylim = c(0, 10000))
```




```
diamonds %>% select(price) %>%
  ggplot(aes(x = price )) +
  geom_histogram(binwidth = 100) +
  xlim(0, 5000) +
  ylim(0, 10000)
```

```
## Warning: Removed 14714 rows containing non-finite values (stat_bin).
```



7.4.1

1.

For histograms missing data is removed.

For bargraphs the NAs are considered another category.

2.

The option *na.rm* in the *mean* and *sum* functions remove the NAs before the values of the functions are computed. NAs are not numeric values so they cannot be included in a sum calculation.

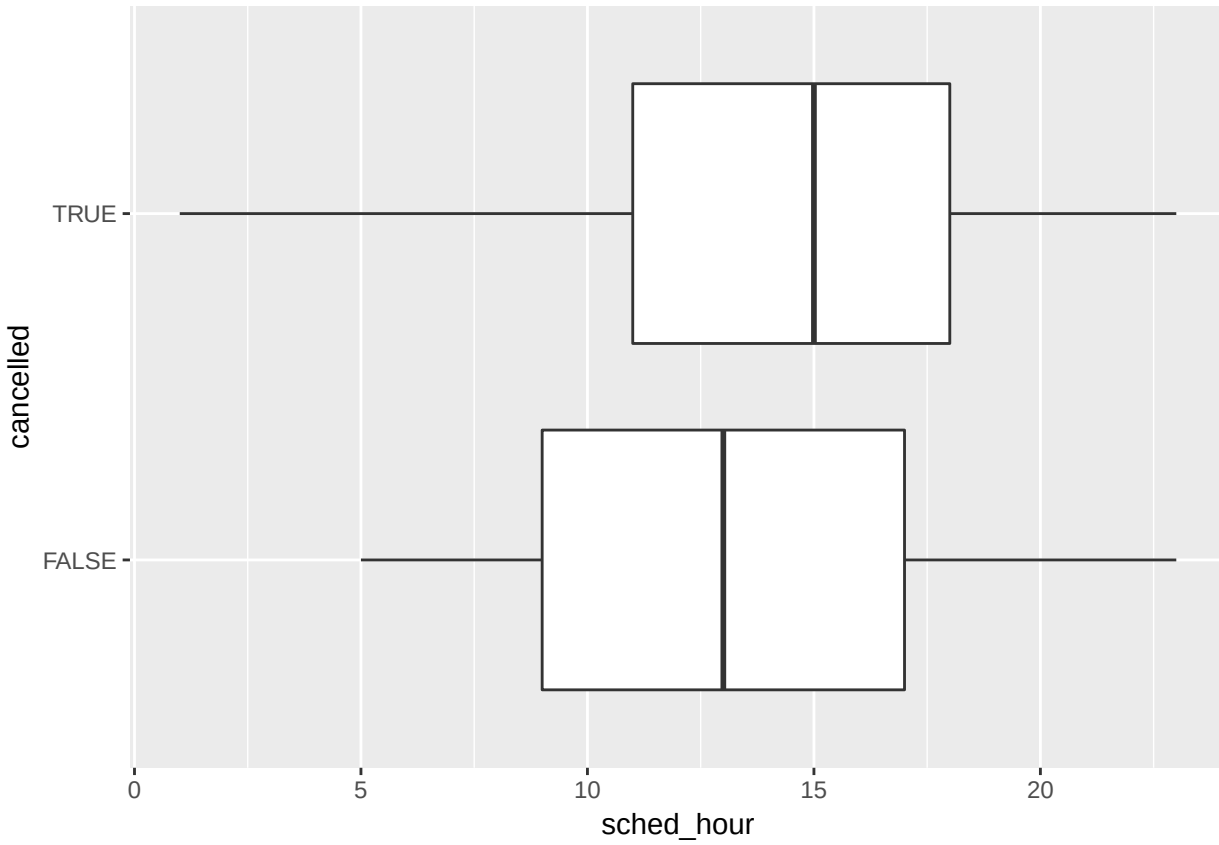
7.5.1.1

1.

The cancelled flights tend to occur later in the day, but have a wider range of scheduled departure hour.

```
library(nycflights13)

flights %>% mutate(
  cancelled = is.na(dep_time),
```

```r
  sched_hour = sched_dep_time %/% 100,
  sched_min  = sched_dep_time %% 100
  ) %>%
  ggplot(aes(x = cancelled, y = sched_hour)) +
  geom_boxplot() +
  coord_flip()
```



2.

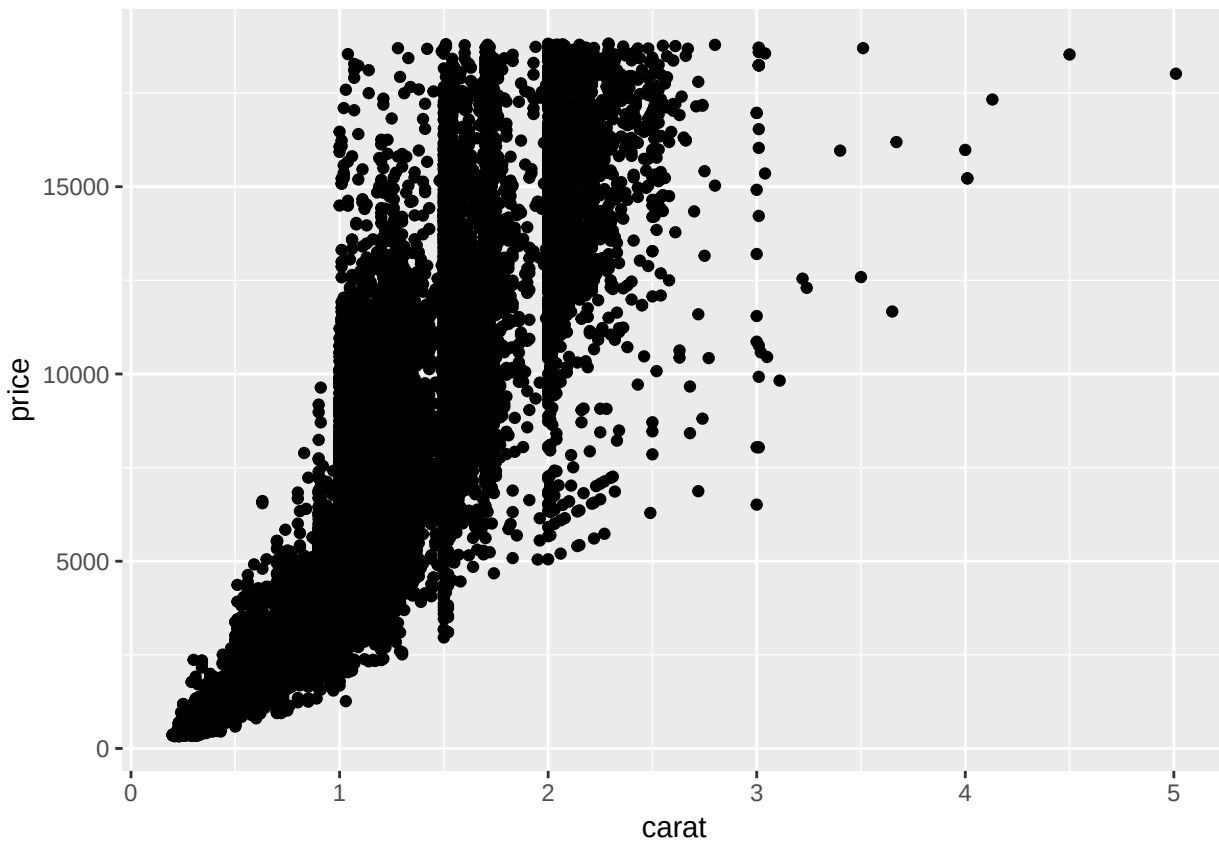
The most important variable is carat.

```r
diamonds %>% select (price, carat, depth, table, x, y , z) %>%
  cor()
```

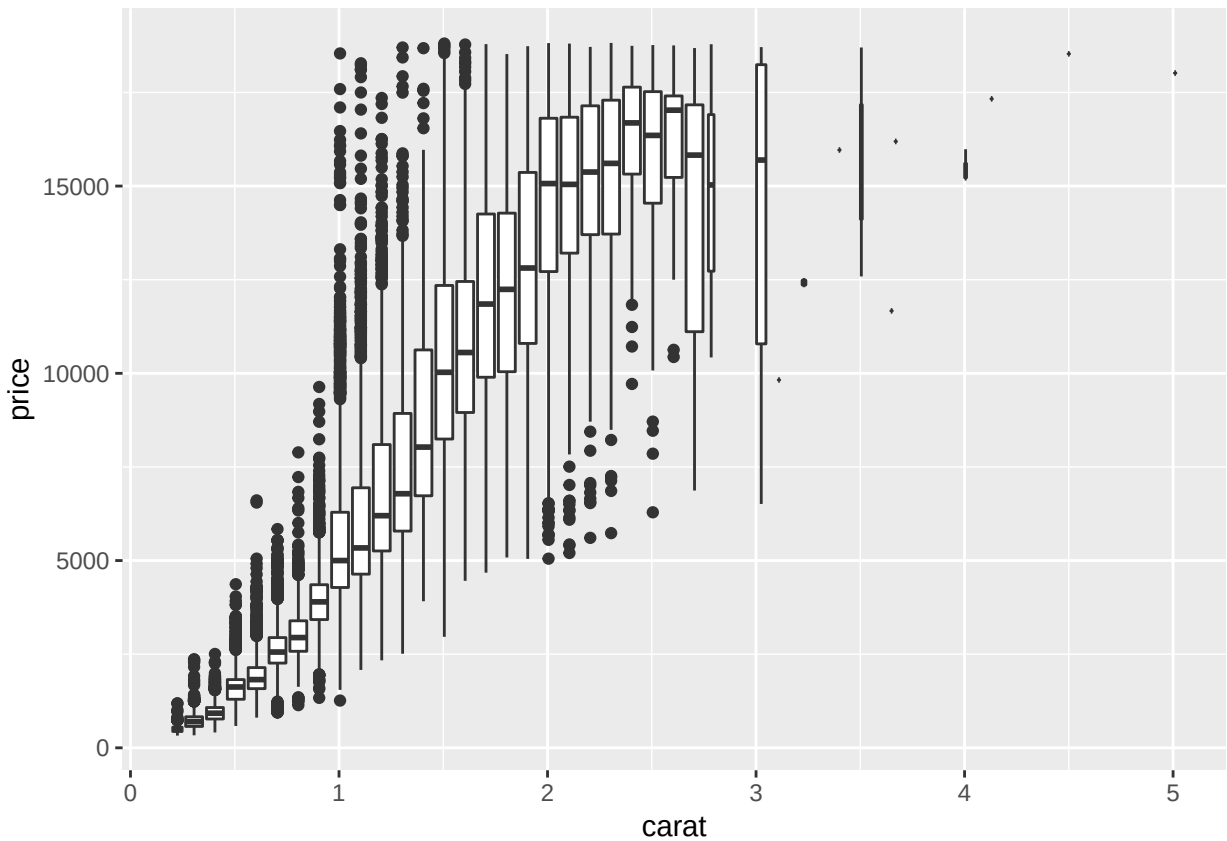
```
##            price      carat       depth      table           x           y
## price  1.0000000 0.92159130 -0.01064740  0.1271339  0.88443516  0.86542090
## carat  0.9215913 1.00000000  0.02822431  0.1816175  0.97509423  0.95172220
## depth -0.0106474 0.02822431  1.00000000 -0.2957785 -0.02528925 -0.02934067
## table  0.1271339 0.18161755 -0.29577852  1.0000000  0.19534428  0.18376015
## x      0.8844352 0.97509423 -0.02528925  0.1953443  1.00000000  0.97470148
## y      0.8654209 0.95172220 -0.02934067  0.1837601  0.97470148  1.00000000
## z      0.8612494 0.95338738  0.09492388  0.1509287  0.97077180  0.95200572
##               z
## price 0.86124944
## carat 0.95338738
## depth 0.09492388
```

```
## table 0.15092869
## x     0.97077180
## y     0.95200572
## z     1.00000000
```

```
diamonds %>% select (price, carat, depth, table, x, y , z) %>%
  ggplot(aes(x = carat, y = price)) +
  geom_point()
```




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

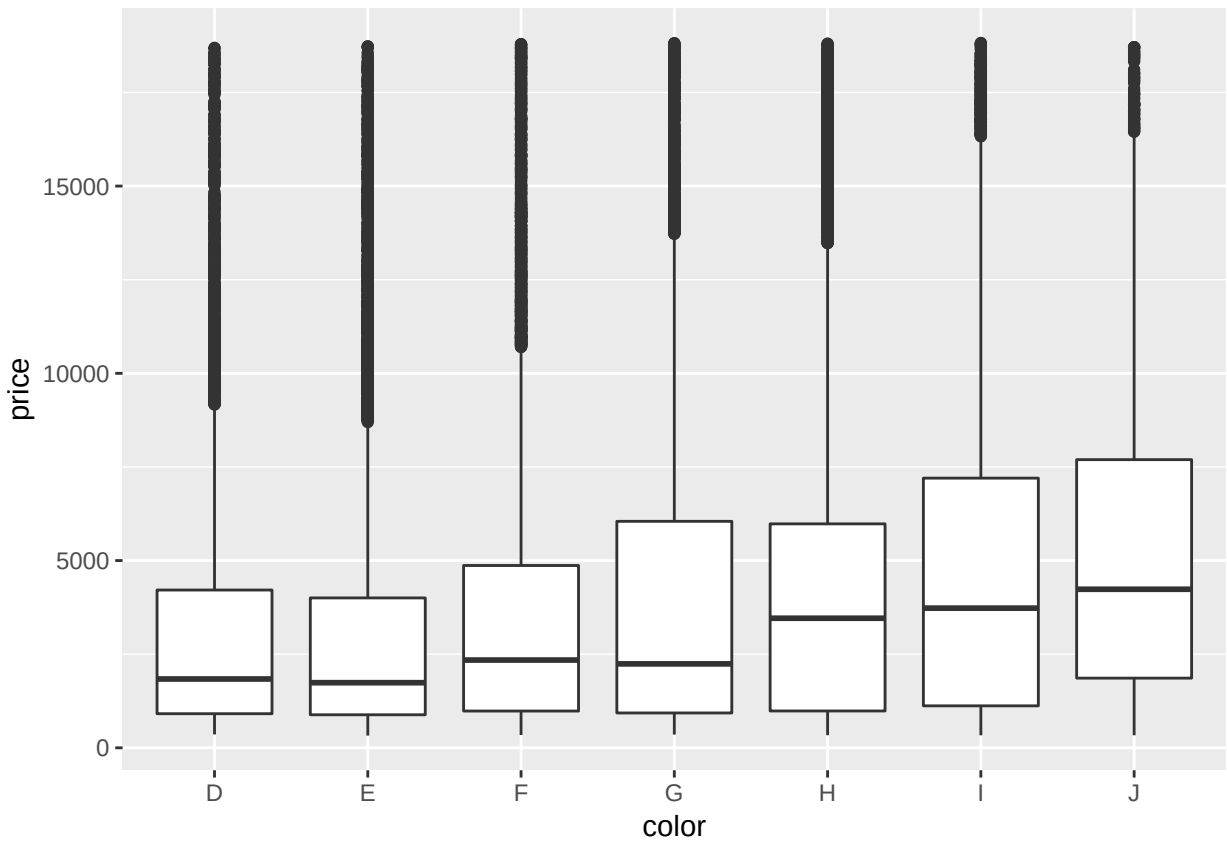


Examining the caregorical variables.

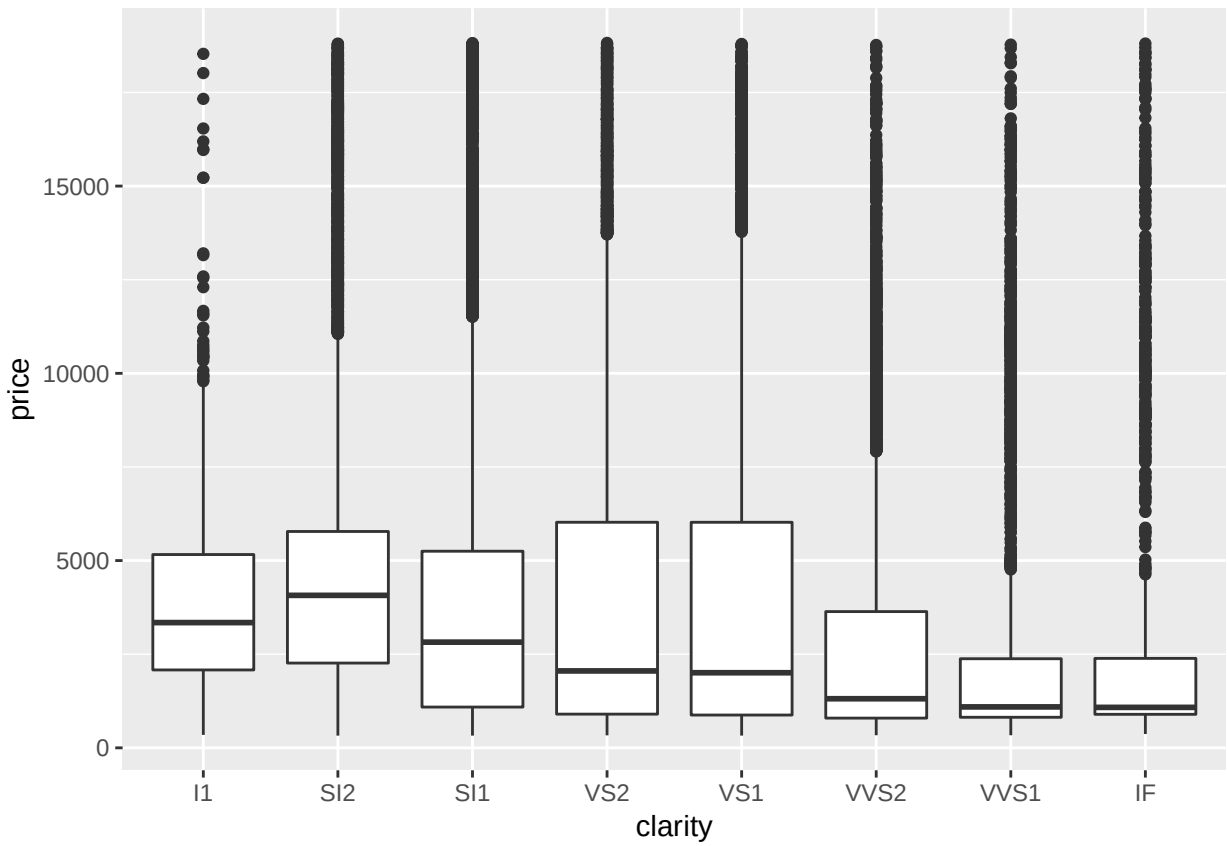
Weak positive relationship of price with color.

Weak neagative relationship of price with clarity and cut.

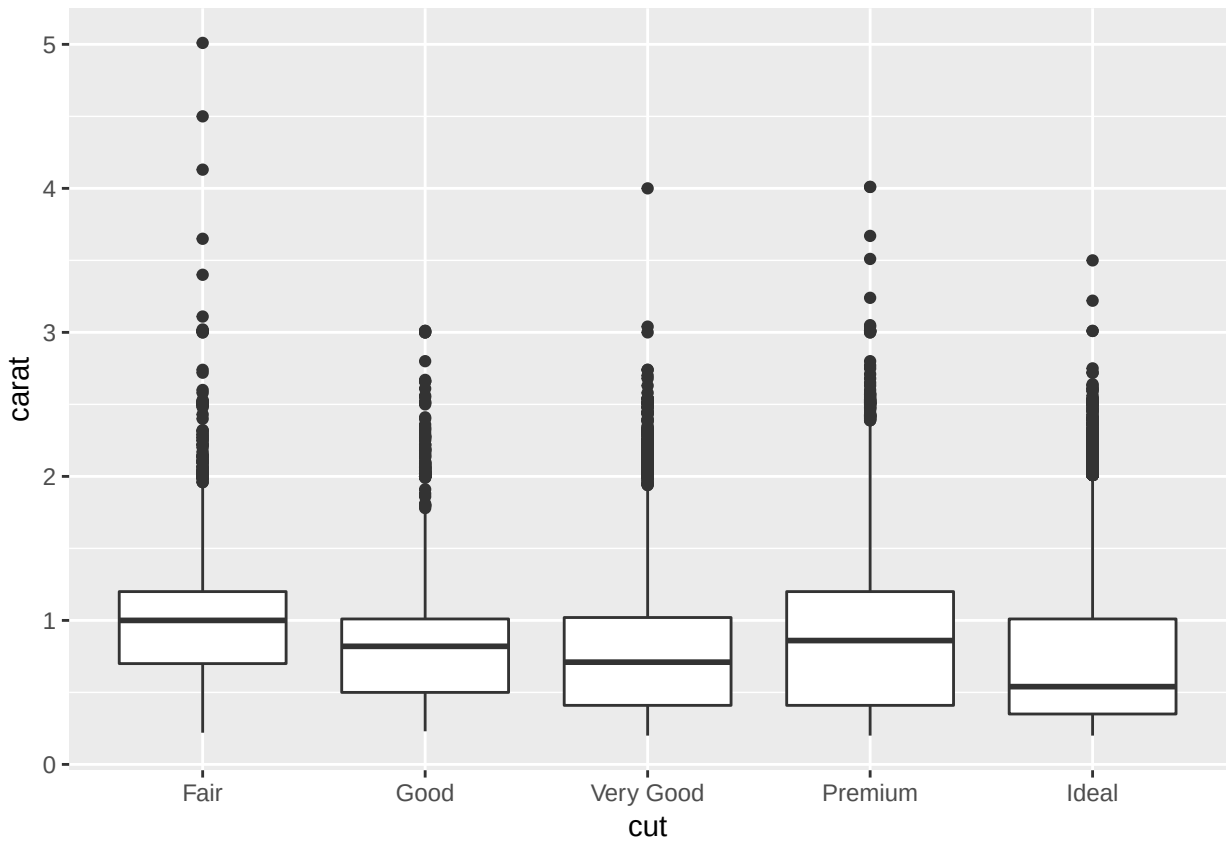
```
diamonds %>% ggplot( aes(x = color, y = price)) +
  geom_boxplot()
```




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



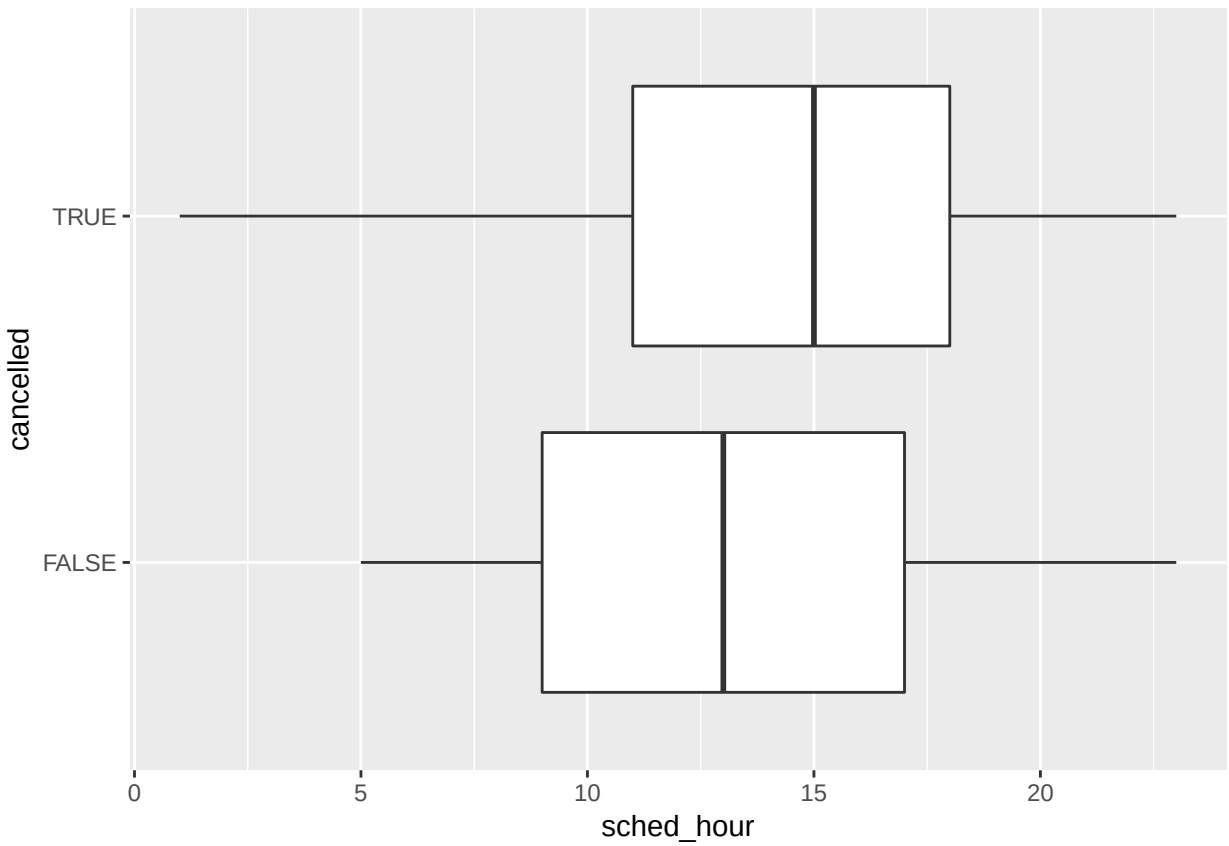

```r
ggplot(diamonds, aes(x = cut, y = carat)) +
  geom_boxplot()
```



3.

Looks the same, but x and y need to be switched for the boxploth()

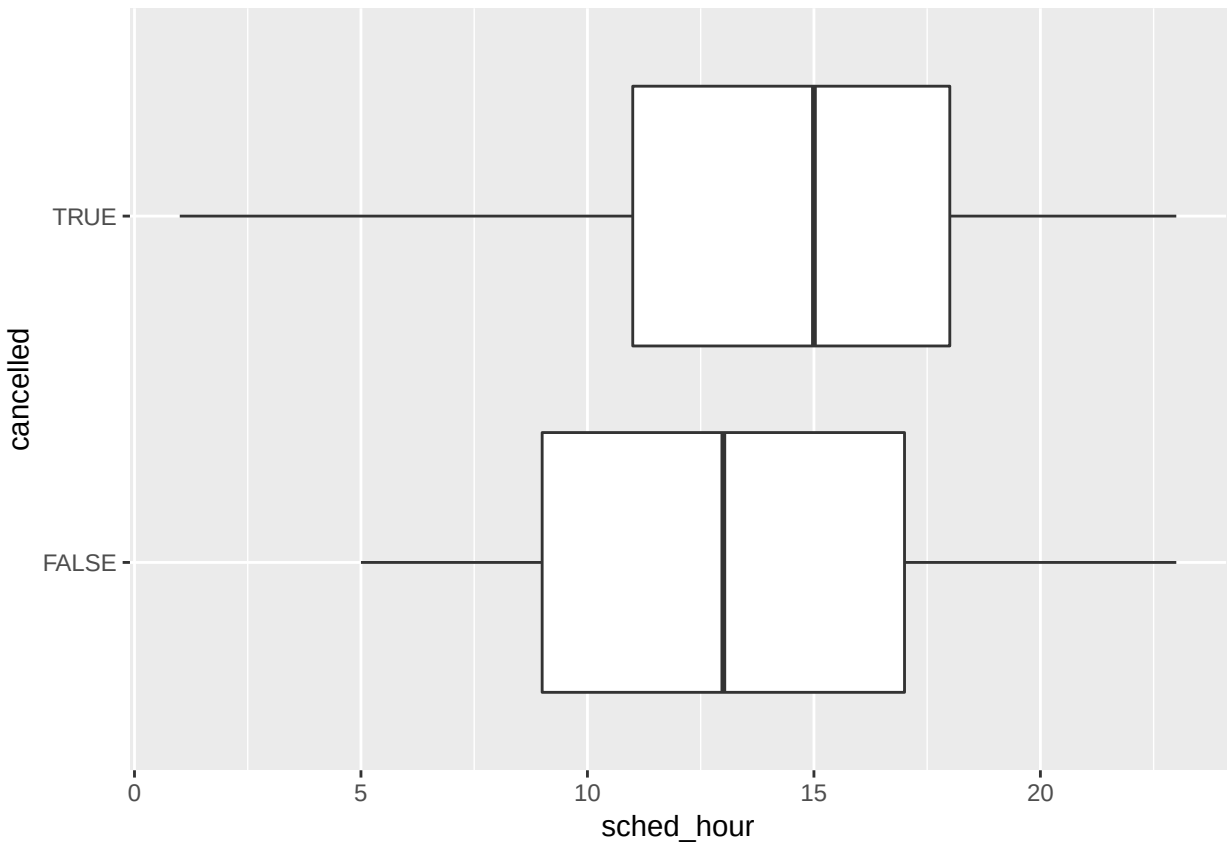
```
flights %>% mutate(
  cancelled = is.na(dep_time),
  sched_hour = sched_dep_time %/% 100,
  sched_min  = sched_dep_time %% 100
  ) %>%
  ggplot(aes(x = cancelled, y = sched_hour)) +
  geom_boxplot() +
  coord_flip()
```




```r
library(ggstance)
```

```
##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```r
flights %>% mutate(
  cancelled = is.na(dep_time),
  sched_hour = sched_dep_time %/% 100,
  sched_min  = sched_dep_time %% 100
  ) %>%
  ggplot(aes(y = cancelled, x = sched_hour)) +
  geom_boxploth()
```

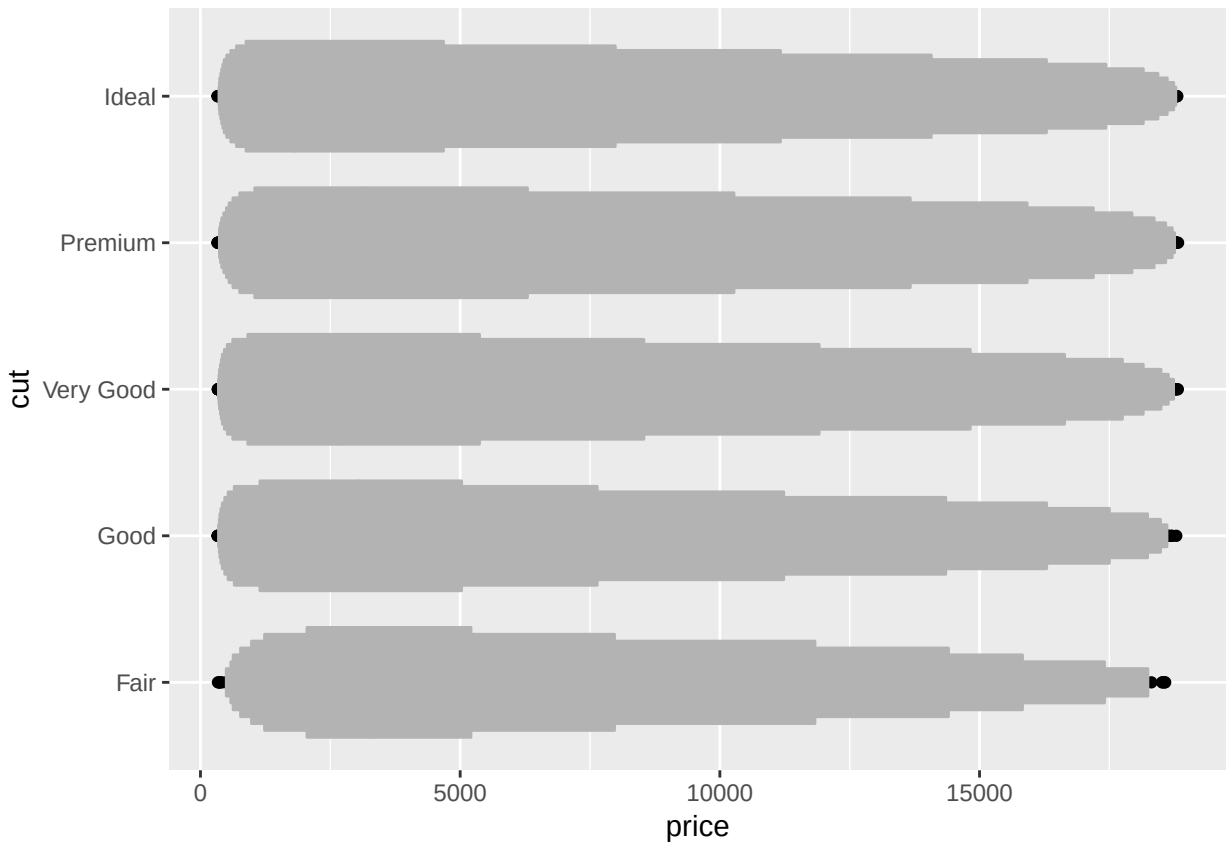


4.

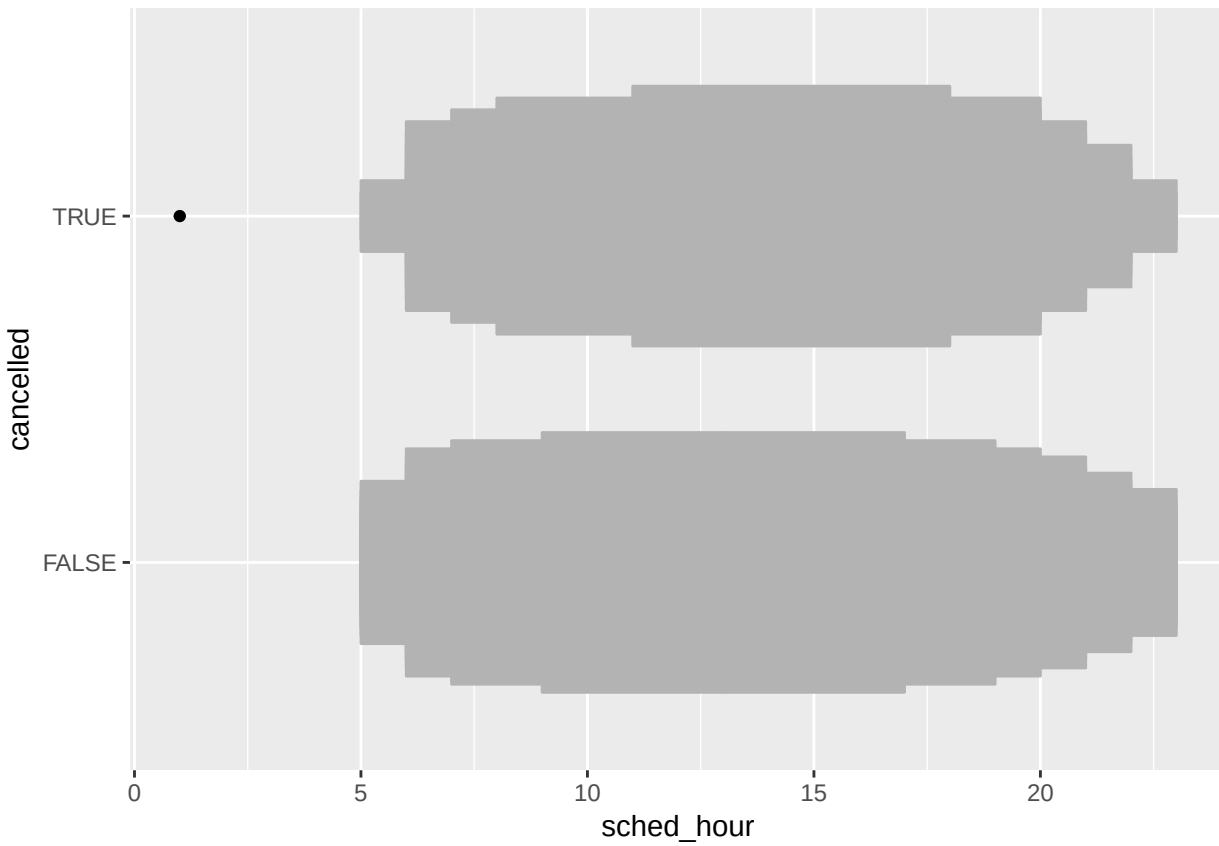
The boxes in the lvplot correspond to percentiles, every 10%.

Outliers are in the direction of the thinner percentiles.

```
library(lvplot)

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



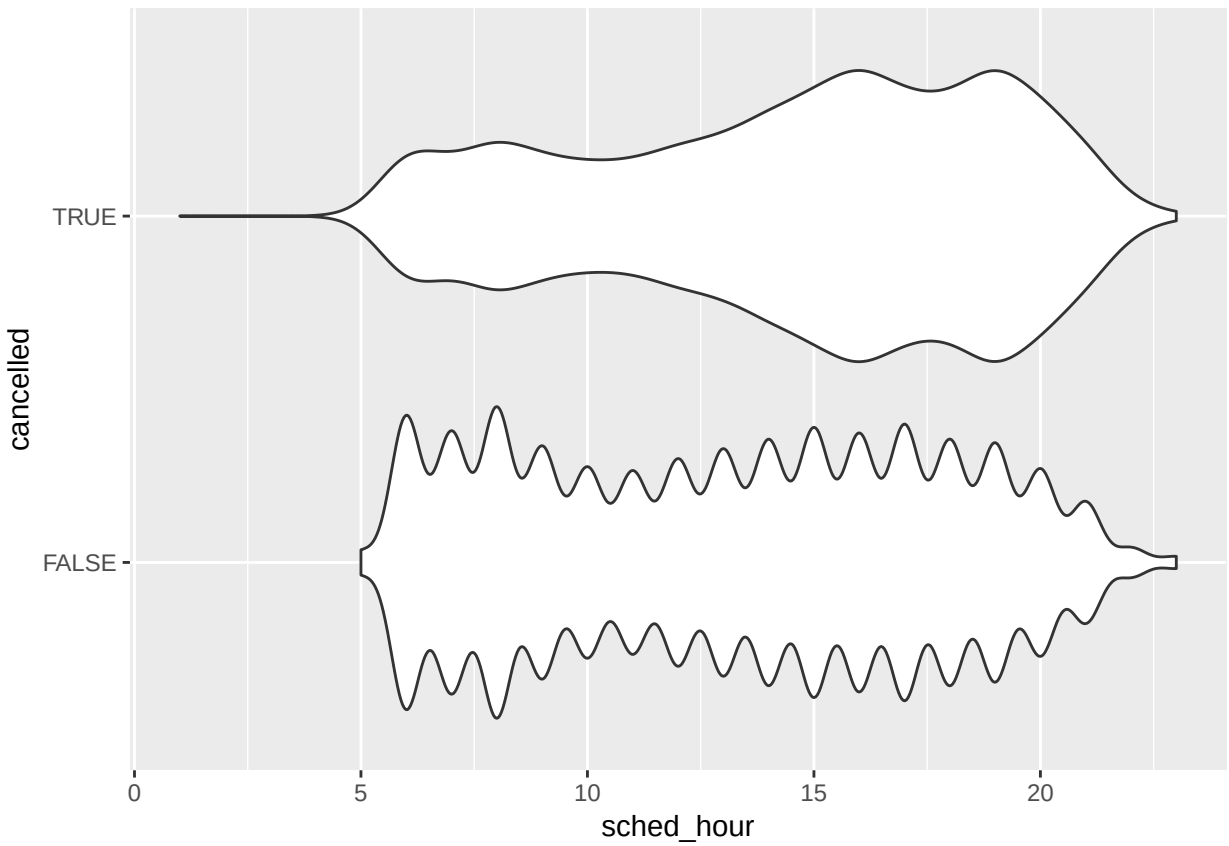

```
flights %>% mutate(
  cancelled = is.na(dep_time),
  sched_hour = sched_dep_time %/% 100,
  sched_min  = sched_dep_time %% 100
  ) %>%
  ggplot(aes(x = cancelled, y = sched_hour)) +
  geom_lv() +
  coord_flip()
```



5.

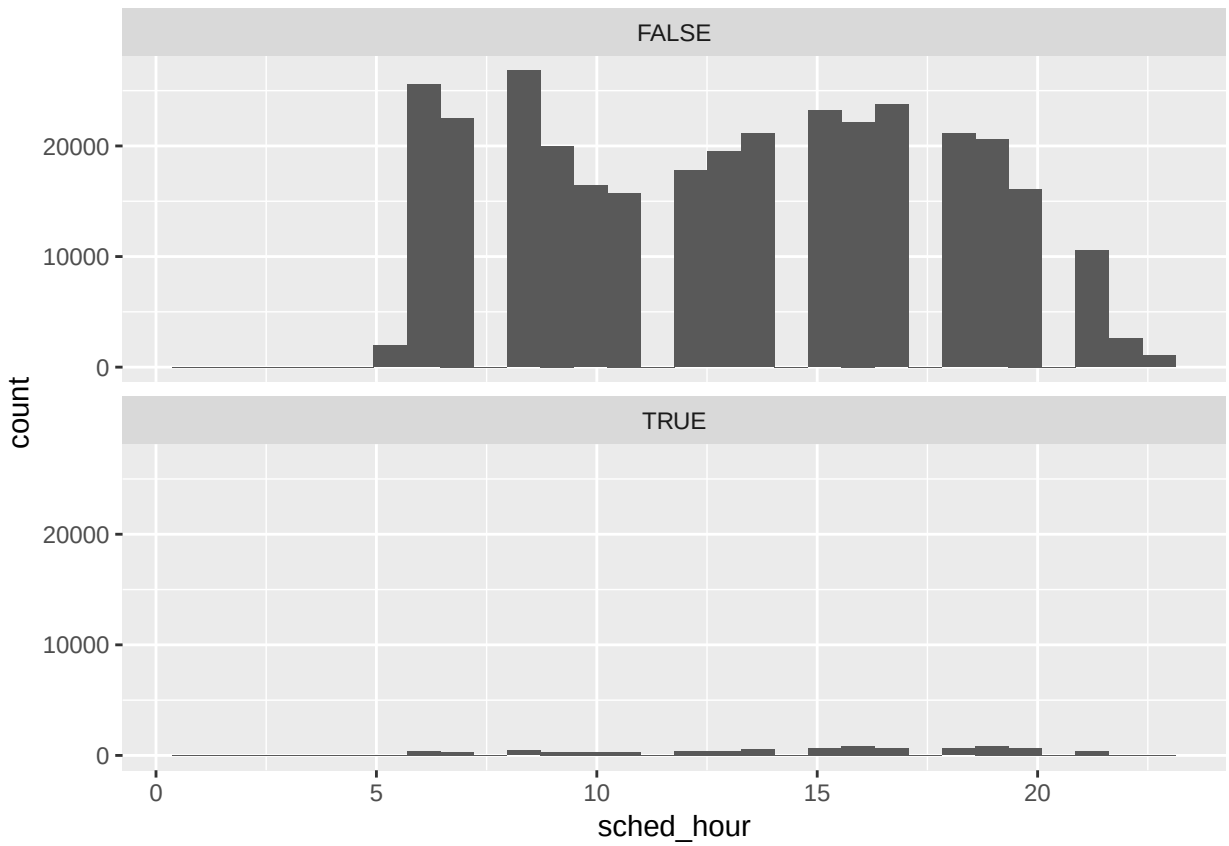
The facted histograms are printed in the reverse order of the violin plots. I would be good to have the vertical scales the same.

```
flights %>% mutate(
  cancelled = is.na(dep_time),
  sched_hour = sched_dep_time %/% 100,
  sched_min  = sched_dep_time %% 100
  ) %>%
  ggplot(aes(x = cancelled, y = sched_hour)) +
  geom_violin() +
  coord_flip()
```




```
flights %>% mutate(
  cancelled = is.na(dep_time),
  sched_hour = sched_dep_time %/% 100,
  sched_min  = sched_dep_time %% 100
  ) %>%
  ggplot(aes( x = sched_hour )) +
  geom_histogram() +
  facet_wrap(~ cancelled, nrow = 2)
```

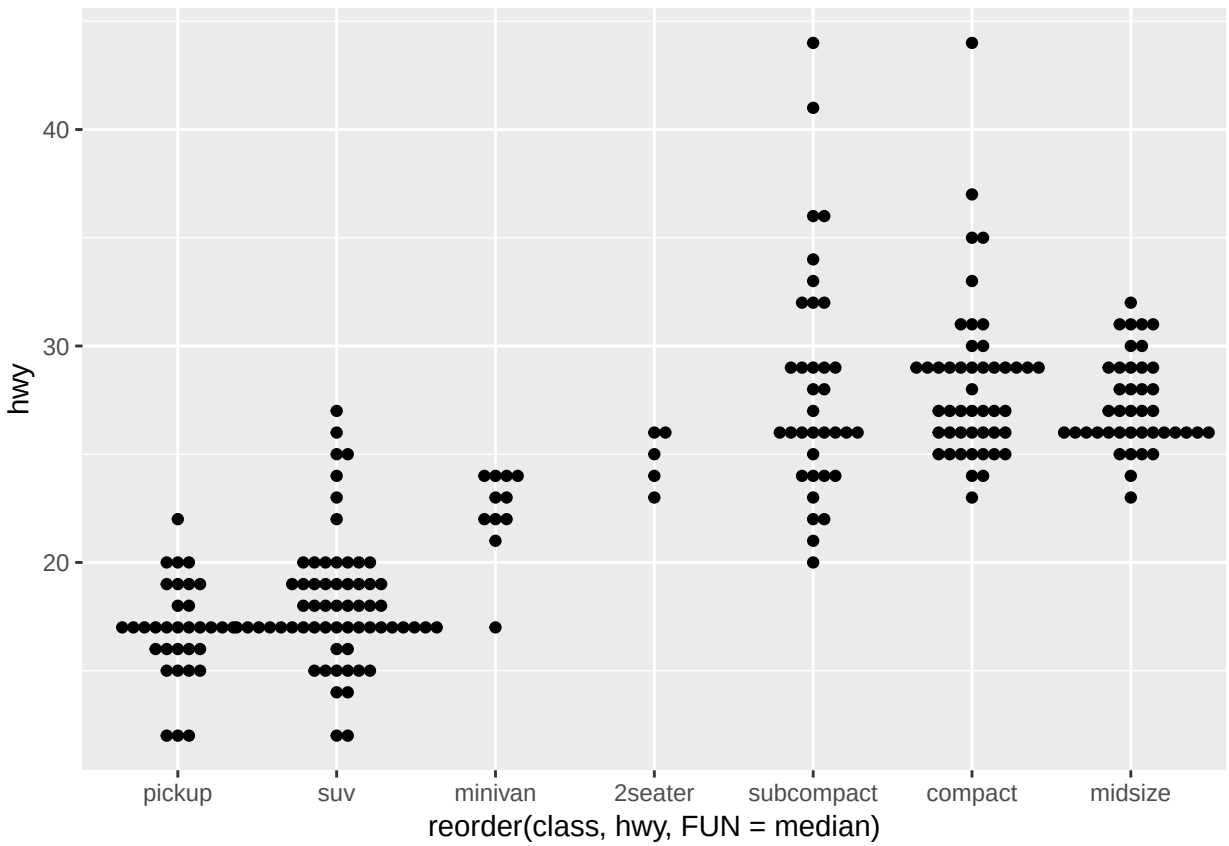
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



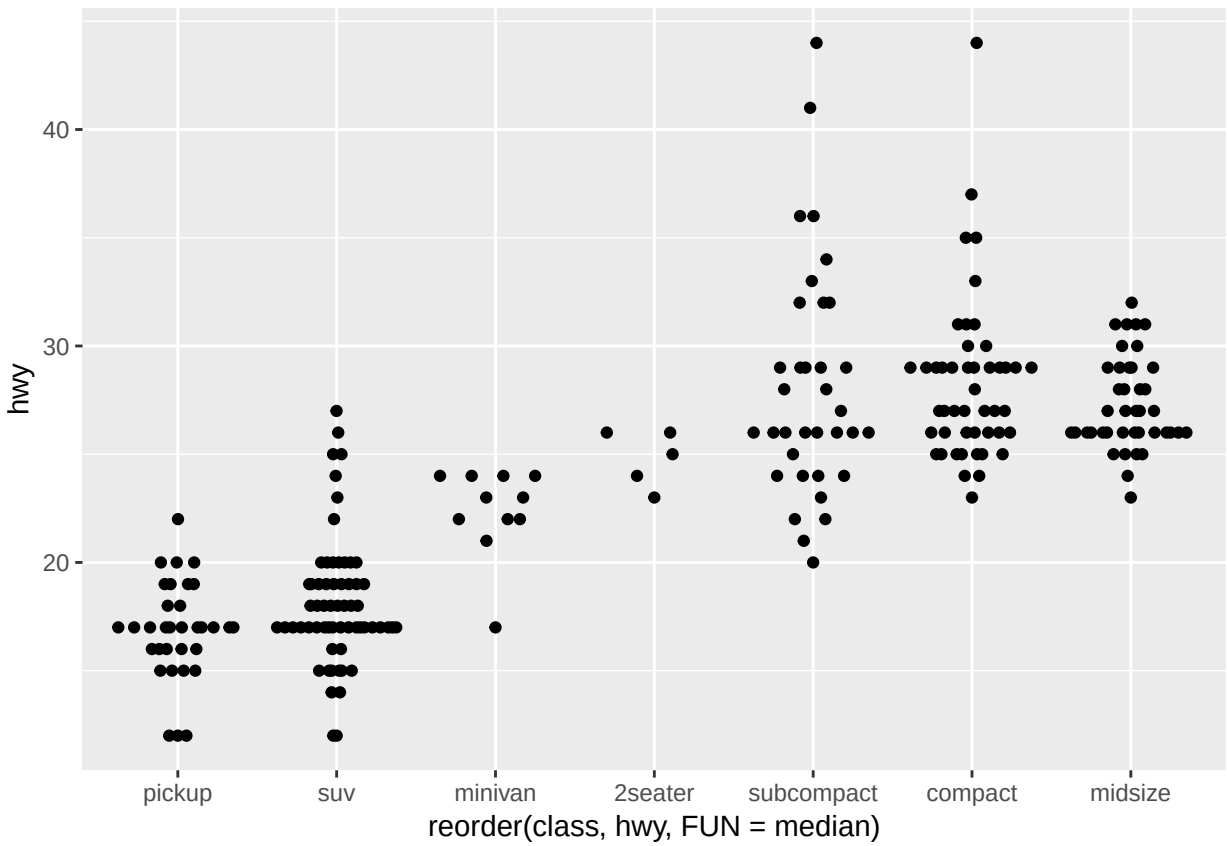
6.

method	description
default	jitters the point horizontally
tukey	jitters more
tukeyDense	jitters but less than tukey
frowney	jitters downward
smiley	jitters upward

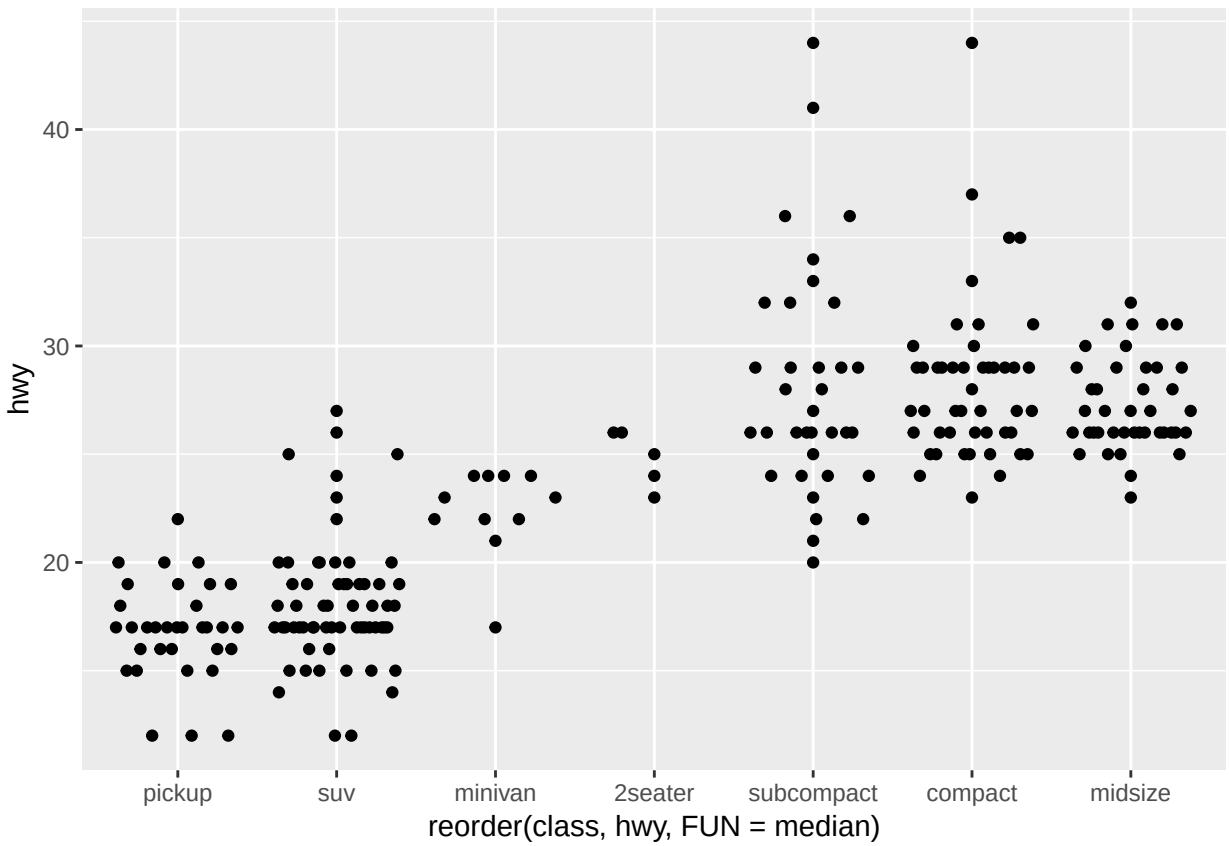
```
library(ggbeeswarm)

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



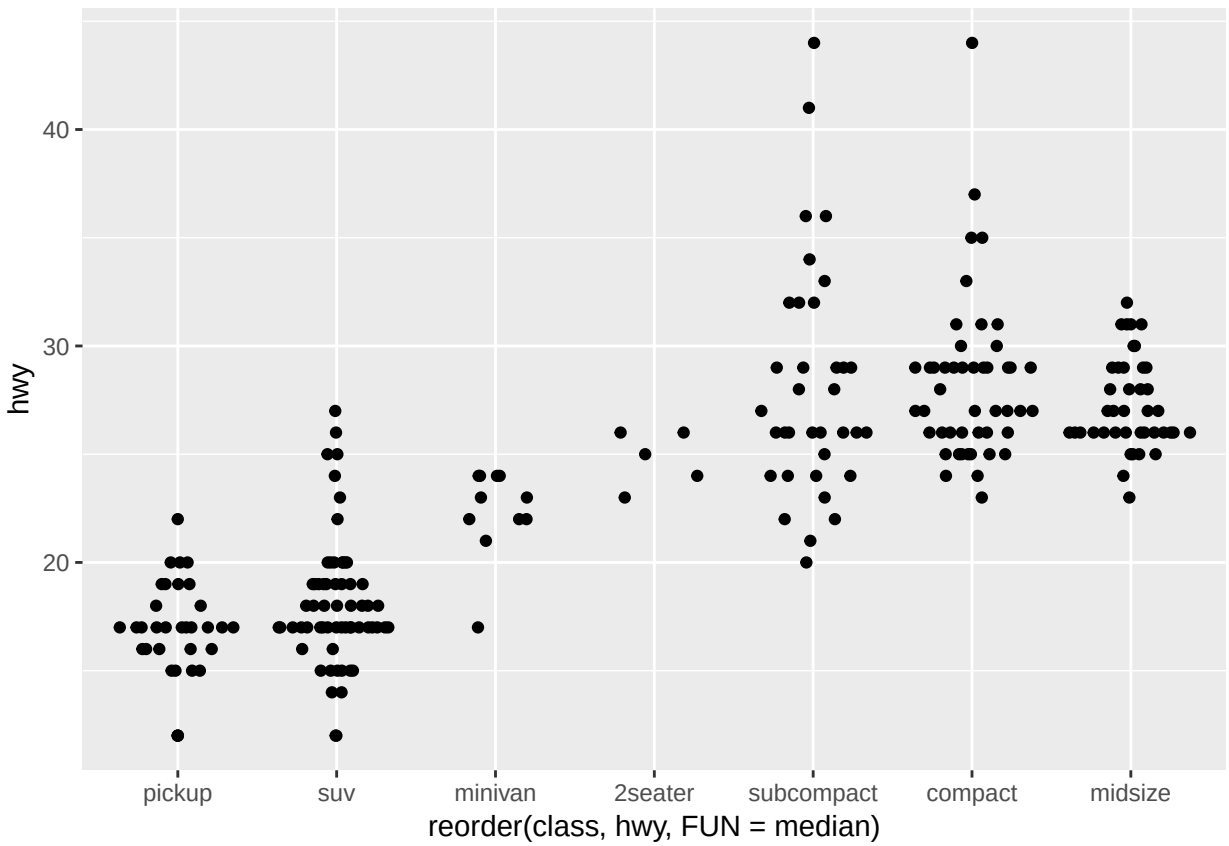

```
mpg %>% ggplot(aes(x = reorder(class, hwy, FUN = median),y = hwy) ) +
  geom_quasirandom()
```



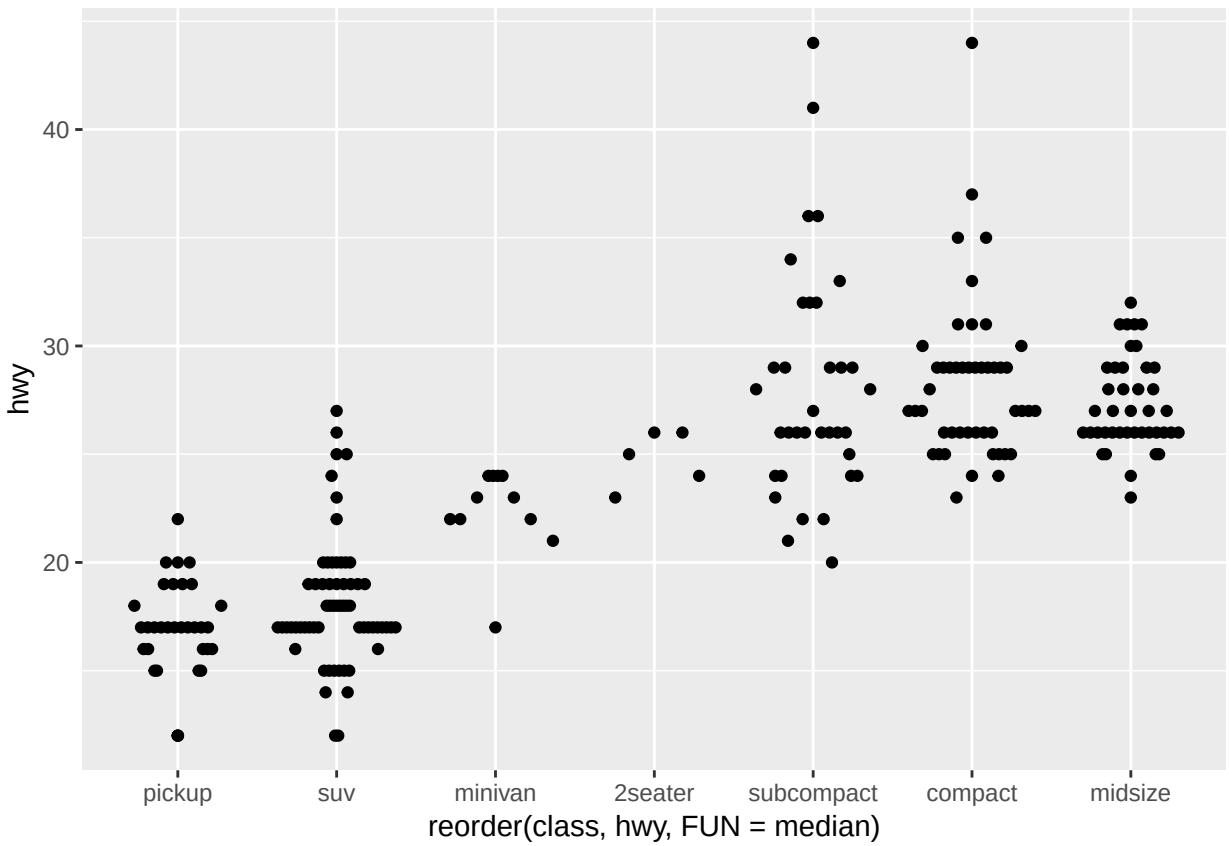

```
mpg %>% ggplot(aes(x = reorder(class, hwy, FUN = median),y = hwy) ) +
  geom_quasirandom(method = "tukey")
```



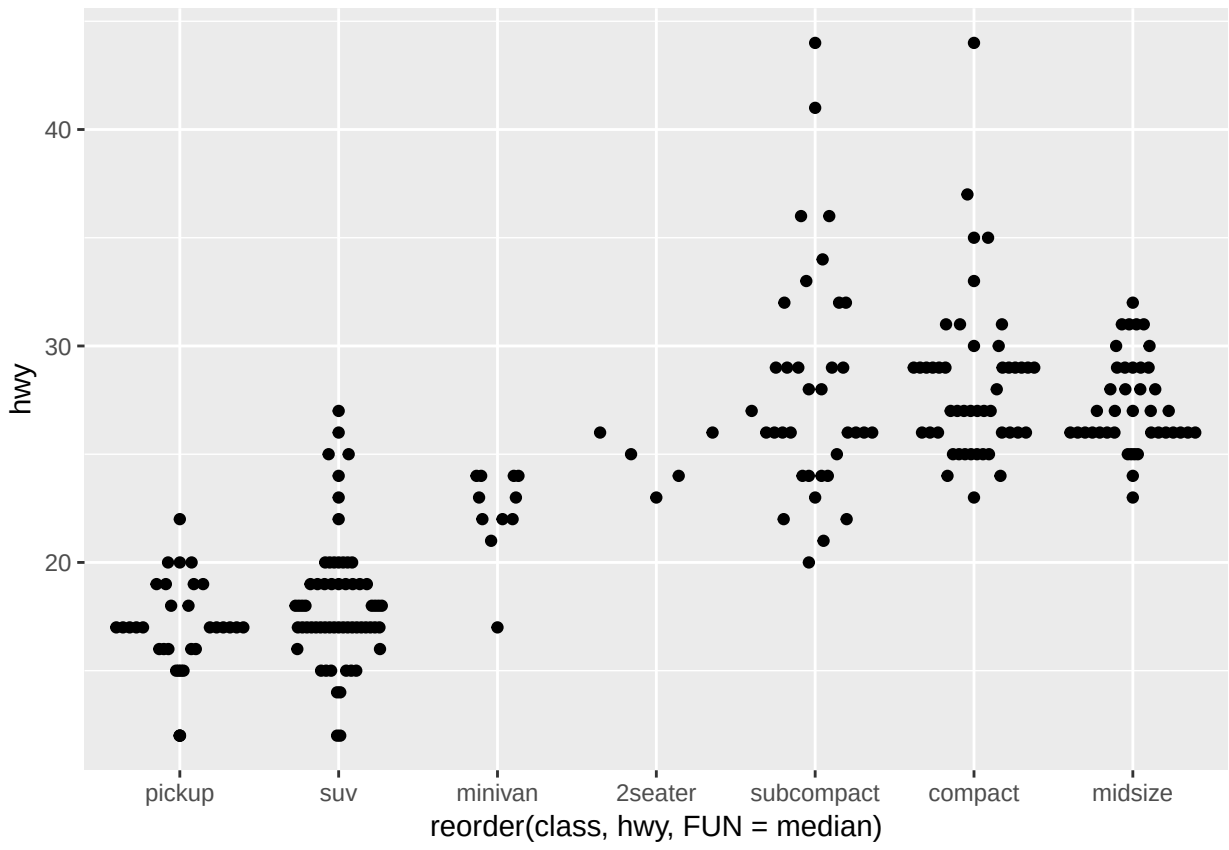

```
mpg %>% ggplot(aes(x = reorder(class, hwy, FUN = median),y = hwy) ) +
  geom_quasirandom(method = "tukeyDense")
```




```
mpg %>% ggplot(aes(x = reorder(class, hwy, FUN = median),y = hwy) ) +
  geom_quasirandom(method = "frowney")
```




```
mpg %>% ggplot(aes(x = reorder(class, hwy, FUN = median),y = hwy) ) +
  geom_quasirandom(method = "smiley")
```


40
30
20
hwy
pickup
suv
minivan
2seater
subcompact
compact
midsize
reorder(class, hwy, FUN = median)