HW 2

Ashe Underwood

10/12/2020

7.3.4 Exercises

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.)

```
## ── Attaching packages ─────────────────── tidyverse 1.3.0 ──
```

```
## ✓ tibble  3.0.3     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
## ✓ purrr   0.3.4
```

```
## ── Conflicts ────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

First, we need to use library to bring ggplot2 and tidyverse functions into our rmarkdown document. Despite using {r echo = FALSE} when calling these packages to our document, tidyverse has decided it will display itself loading into the document regardless.

With that taken care of, the diamonds data frame can be selected and arranged by price to provide a view of its distribution.

```
diamonds
```

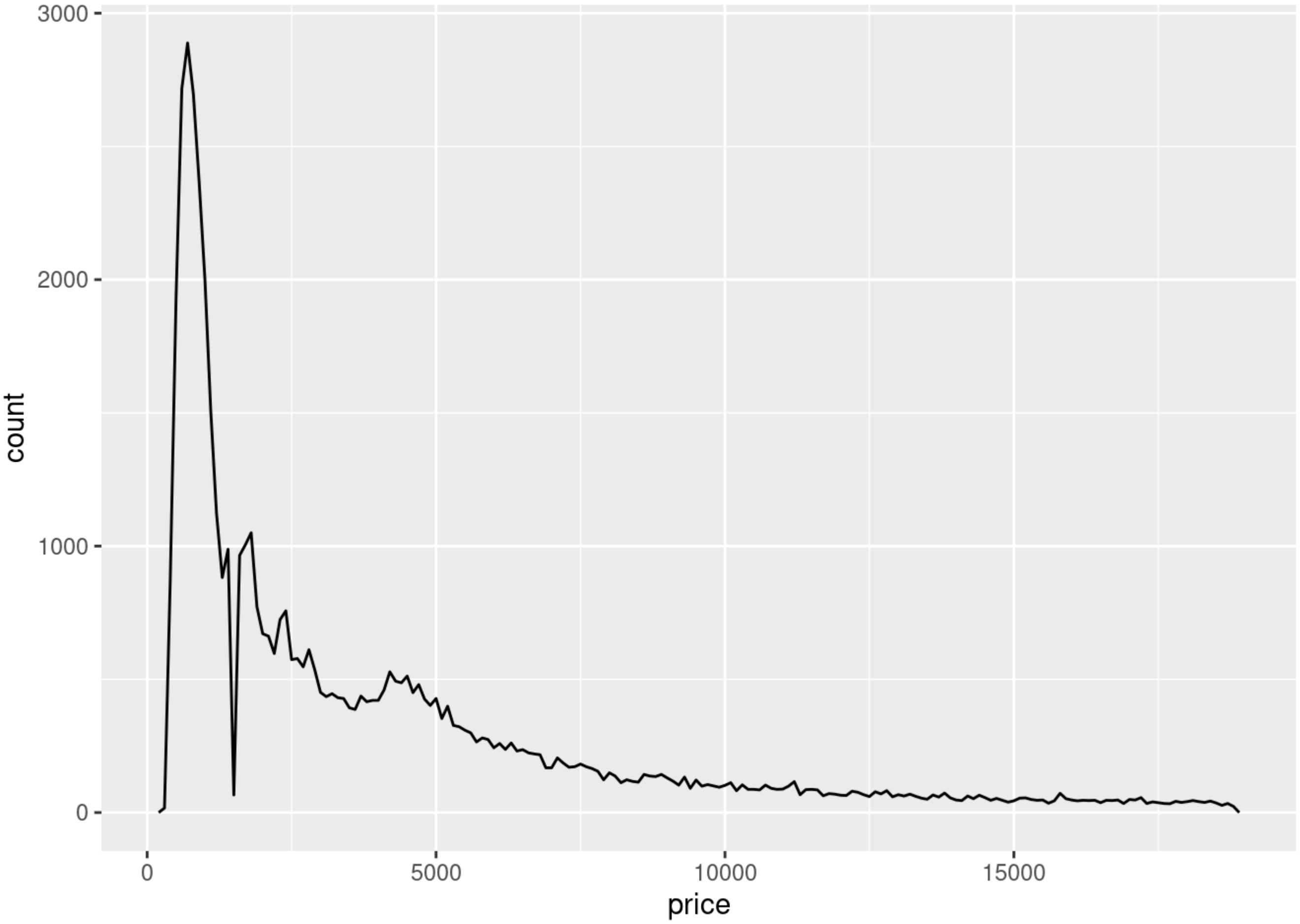
```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2 0.21  Premium   E     SI1      59.8    61   326  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
```

This tibble provides a basis for which variables to explore, but the ones that provide the most meaningful information to the average person are cut and carat, as they directly correspond to how well the diamond was shaped as well as how big it is, providing a vague sense of what diamonds should be valuable.

```
price_info <- diamonds%>%
  select(price, cut, carat)%>%
  arrange(price)
price_info
```

```
## # A tibble: 53,940 x 3
##    price cut       carat
##    <int> <ord>     <dbl>
##  1   326 Ideal     0.23
##  2   326 Premium   0.21
##  3   327 Good      0.23
##  4   334 Premium   0.290
##  5   335 Good      0.31
##  6   336 Very Good 0.24
##  7   336 Very Good 0.24
##  8   337 Very Good 0.26
##  9   337 Fair      0.22
## 10   338 Very Good 0.23
## # … with 53,930 more rows
```

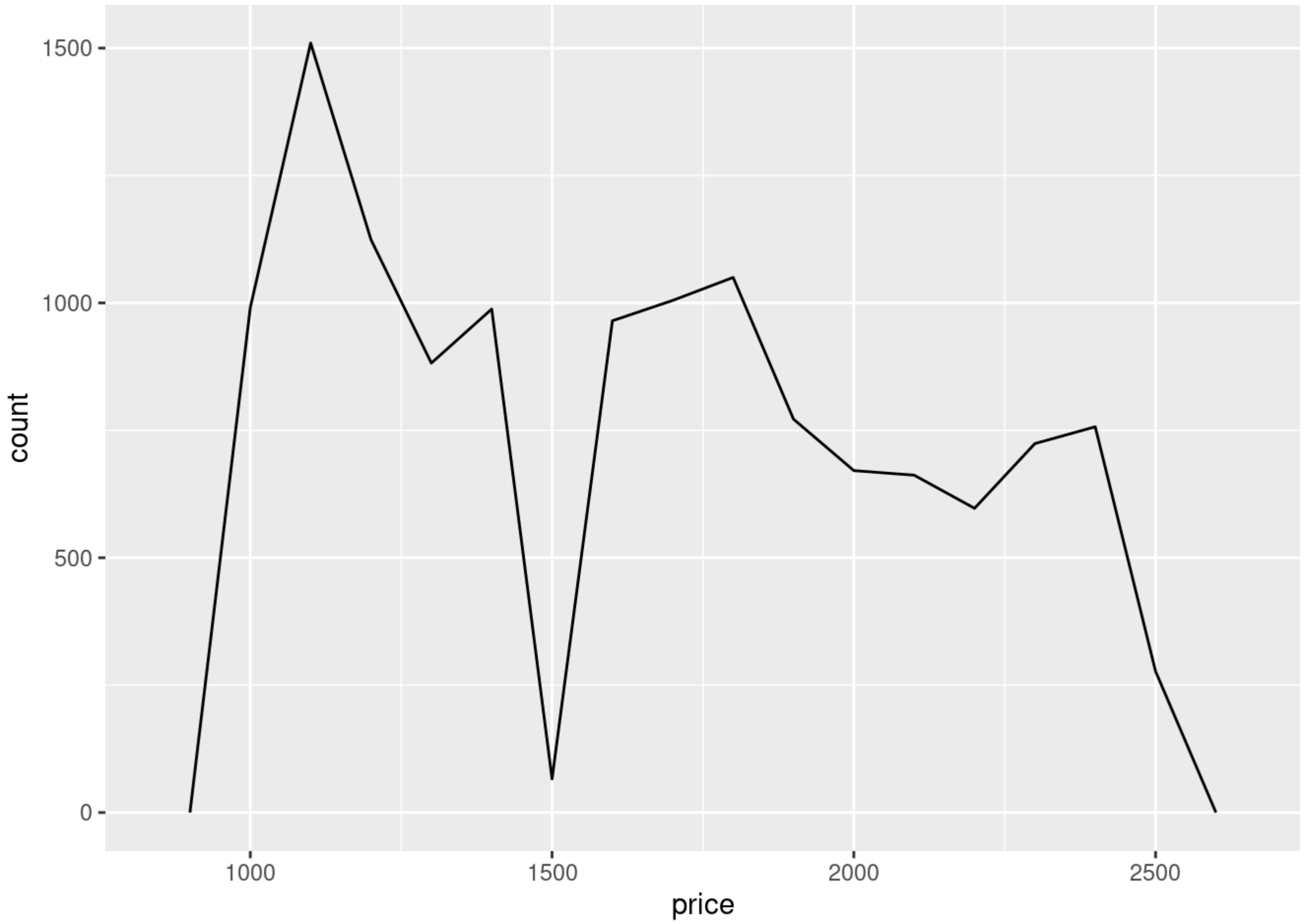
```
ggplot(diamonds)+
  geom_freqpoly(mapping = aes(x = price), binwidth = 100)
```



The price_info tibble reveals some weird values. When arranged from lowest to highest price values, the cut quality of the diamond does not seem to correlate very much with the diamond's eventual price. However, the graph below shows a generally logical progression, with there being a larger quantity of less expensive diamonds than more expensive. Additionally, the diamond price is fairly evenly influenced by its carat, an amount that dictates price more than cut quality in the case of this data frame.

Another piece of this graph that is strange is the dip to almost 0 diamonds at what appears to the 1000-1200 price mark.

```
weirdness <- price_info %>%
  filter(price >= 1000, price < 2500) %>%
  arrange(price)
ggplot(weirdness)+
  geom_freqpoly(mapping = aes(x = price), binwidth = 100)
```



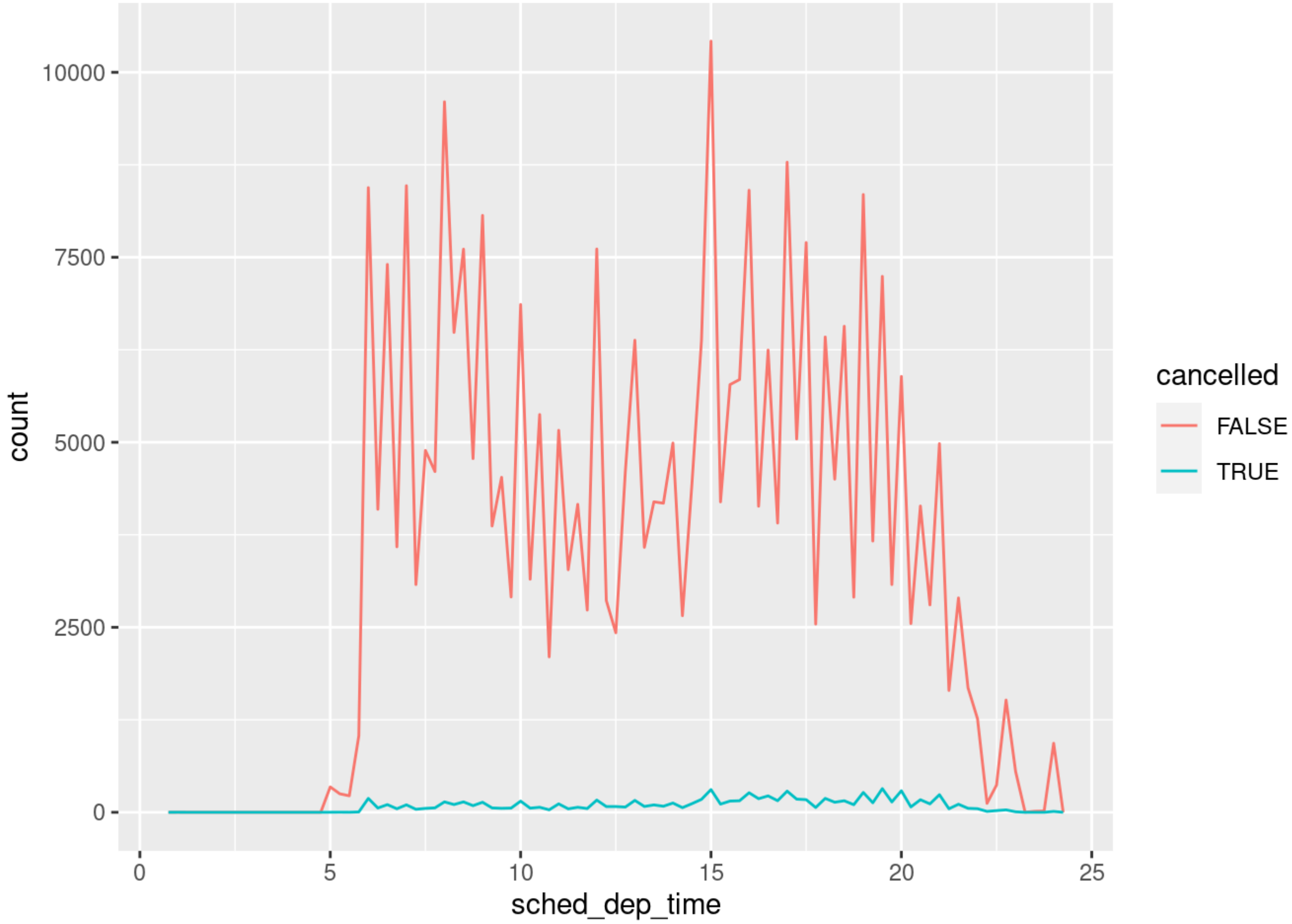
A closer look reveals that there are very few diamonds with a price value of 1500, which causes the strange visual dip in the graph.

7.5.1.1 Exercises

1. Use what you’ve learned to improve the visualisation of the departure times of cancelled vs. non-cancelled flights.

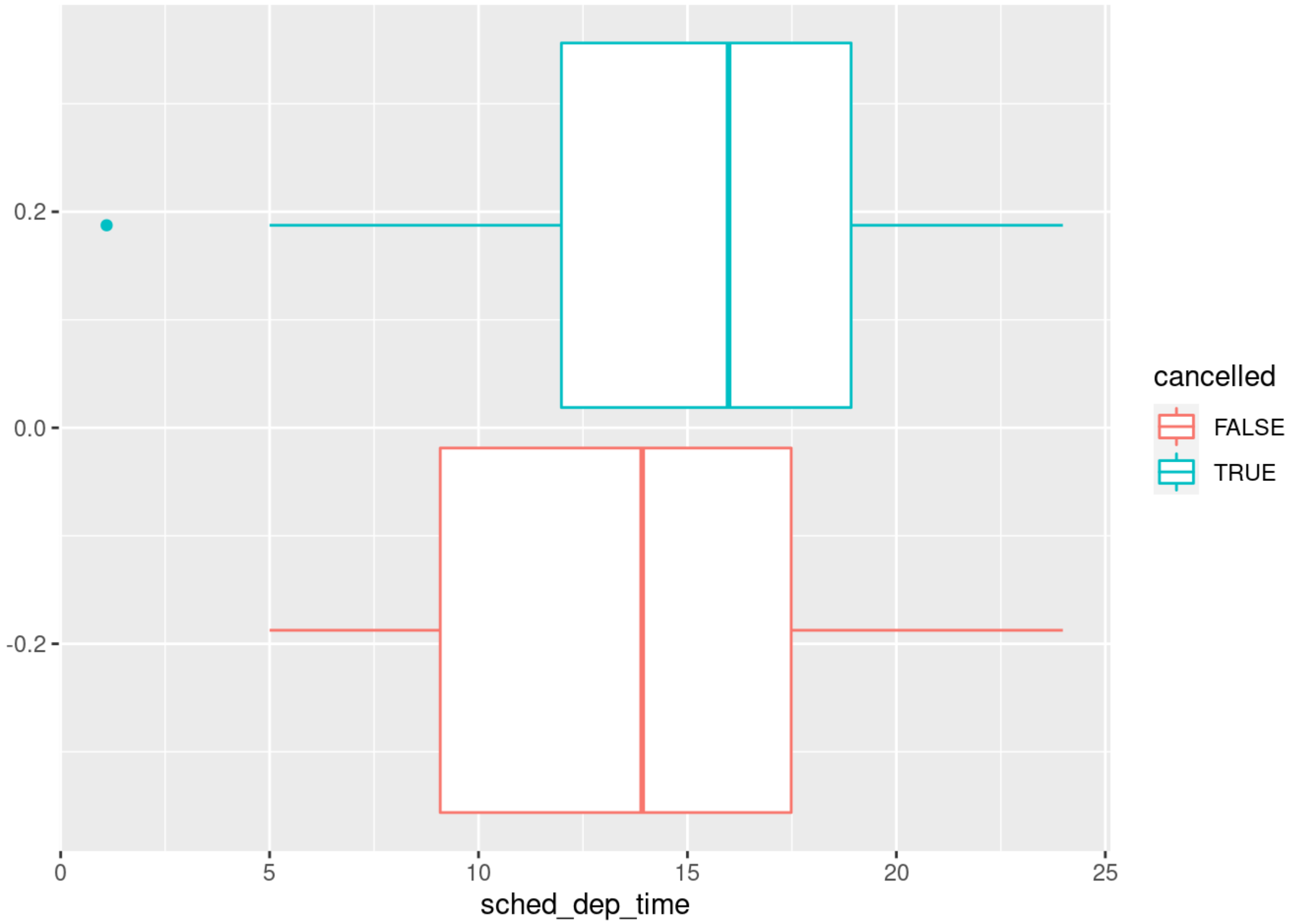
first step is to show the first visualization that is provided in section 7.4

```
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(mapping = aes(sched_dep_time)) +
    geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4)
```



This visual is not great, because it tracks the number of total flgihts cancelled or non-cancelled on the y-axis, information that is not particularly useful to seeing the departure time comparison or to seeing any values for cancelled flights at all really, as the count difference is extreme enough that the cancelled line is hardly visible. A better visualization would be to use divide these two seperate categorical values up into distinct visual representations. As the task here is to compare the distribution of departure times in a comparitive way, the simplest soultion is to make two box plots.

```
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(mapping = aes(sched_dep_time))+
  geom_boxplot(mapping = aes(colour = cancelled))
```



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?

a. While it would make sense that the variable that is the most important for predicting price would be carat, due to the consistency with which it rises with price over the course of the the diamonds tibble. carat is also positively correlated with cut, such that price is determined primarily by carat as said before, and therefore secondarily by cut. A high quality diamond may end up being less expensive due to a lower grade cut quality despite a higher grade carat.