Assignment 2

Austin Dollar

9/21/2021

Question 1

What is an ordinal variable? Identify the ordinal variables in the diamonds data set and specify their rankings.

An ordinal variable is a type of categorical variable, specifically such that they are ordered.

As displayed in the summary call

```
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  0.29 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
```

```
summary(diamonds)
```

```
##      carat               cut        color        clarity          depth      
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065   Min.   :43.00  
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258   1st Qu.:61.00  
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :61.80  
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171   Mean   :61.75  
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50  
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.   :79.00  
##                                     J: 2808   (Other): 2531                  
##      table           price             x                y        
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720  
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710  
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735  
```

```
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900
##
##        z
##  Min.   : 0.000
##  1st Qu.: 2.910
##  Median : 3.530
##  Mean   : 3.539
##  3rd Qu.: 4.040
##  Max.   :31.800
##
```

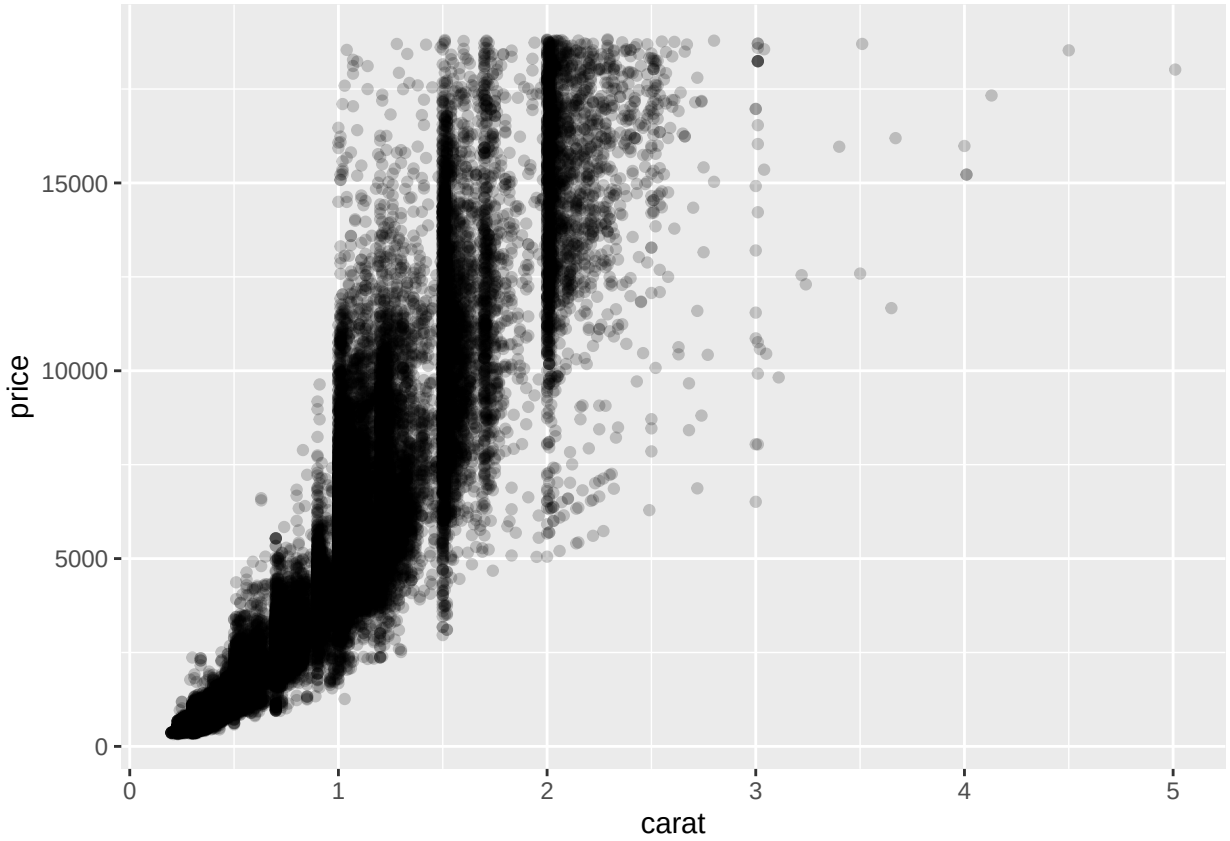
As shown above with simply calling the diamonds dataset, it tells us which variables are ordinal. These variables include cut, color, and clarity. Then, we call the summary(diamonds) function, which gives us detailed information on these variables, including their rankings. For cut, the ranking, is Fair, Good, Very Good, Premium, and finally Ideal. For Color, the order of ranking is as follows: D,E,F„H,I,J. For clarity, the order is: SI1,VS2,SI2,VS1,VVS2,VVS1, and Other.

Question 2

Generate a scatterplot with carat on the x-axis and price on the y-axis. Set alpha, a parameter governing opaqueness, for these point to be 0.2. Do you notice any interesting patterns with respect to the distribution of carats?

```
diamondframe <- data.frame(diamonds)

ggplot(diamondframe, aes(carat, price))+
  geom_point(alpha=0.2)
```



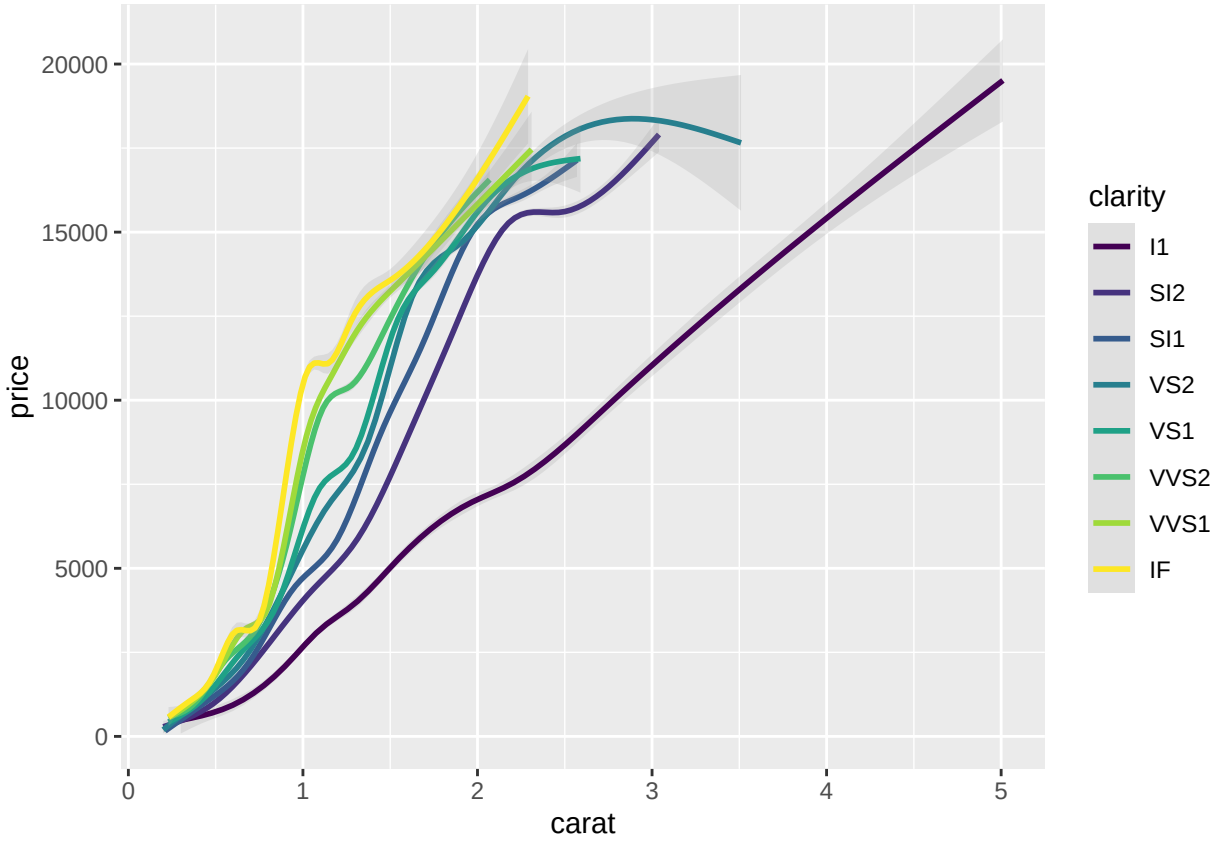
I noticed how by setting opaqueness to a smaller variable, we can tell where there is overlap and a higher concentration of variables. We can see that there is a high concentration on each low end of the carat value, with fewer values as you travel left in between carats, meaning that most are rated at a whole number for the carat.

Question 3

Generate a new figure with the same information as in the previous problem. Add to this figure a color aesthetic for clarity along with smoothed lines, also colored according to clarity. Do not include confidence intervals for these lines. What method was used to generate these lines? Give the full name, not just an acronym.

```
ggplot(diamondframe, aes(carat, price, color=clarity))+
  geom_smooth(alpha=0.2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

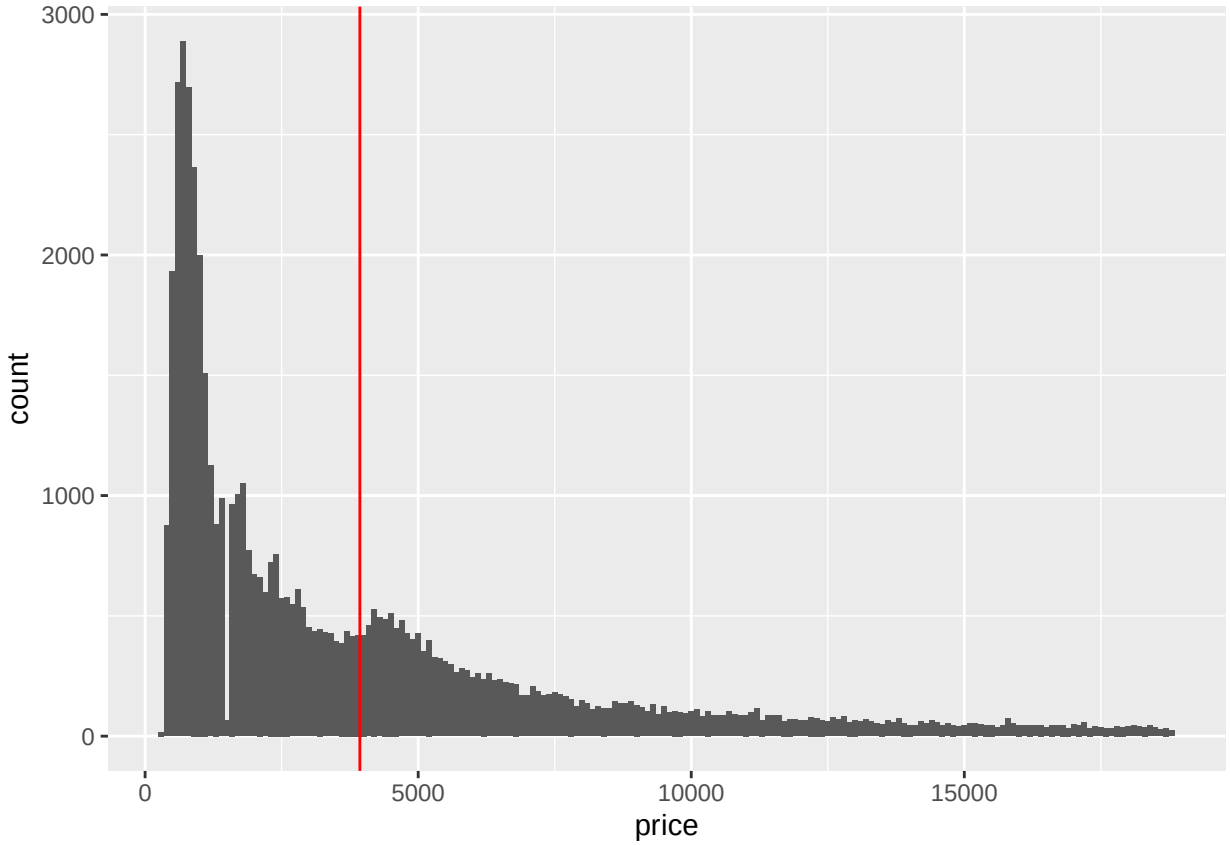


Above is the graph as described in the documentation. The method used to generate these lines was the generalized additive model.

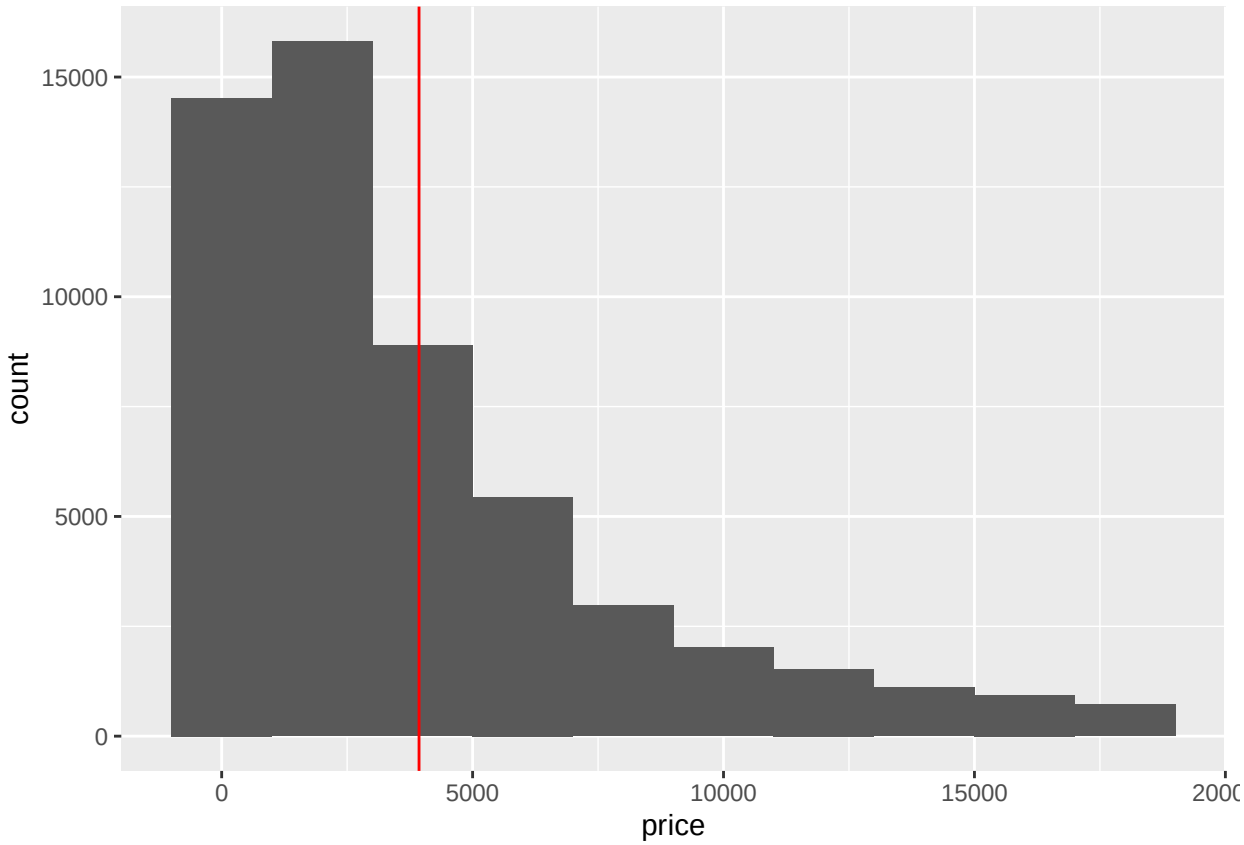
Question 4

Create a histogram of diamond prices. Include a red vertical dashed line at the mean (the vline geometric object and this link might be helpful). Set the binwidth parameter to 100. Do you notice a slight bump right below 5000? Next set binwidth to 2000. Try a binwidth of 1. Why might binwidth or bins be important parameters when creating histograms?

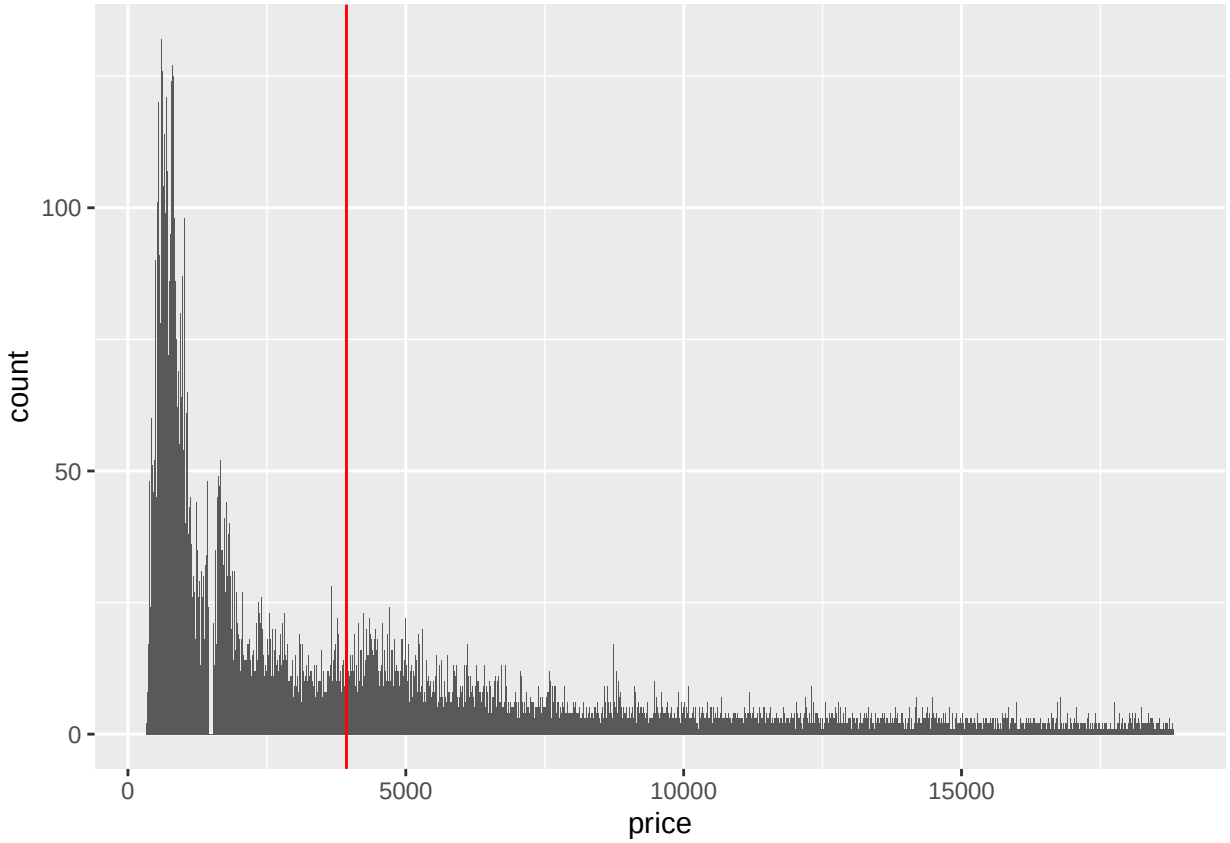
```
#binwidth 100
ggplot(diamonds, aes(price))+
  geom_histogram(binwidth = 100)+
  geom_vline(xintercept = mean(diamonds$price), color = 'red')
```




```
#binwidth 2000
ggplot(diamonds, aes(price))+
  geom_histogram(binwidth = 2000)+
  geom_vline(xintercept = mean(diamonds$price), color = 'red')
```




```
#binwidth 1
ggplot(diamonds, aes(price))+
  geom_histogram(binwidth = 1)+
  geom_vline(xintercept = mean(diamonds$price), color = 'red')
```

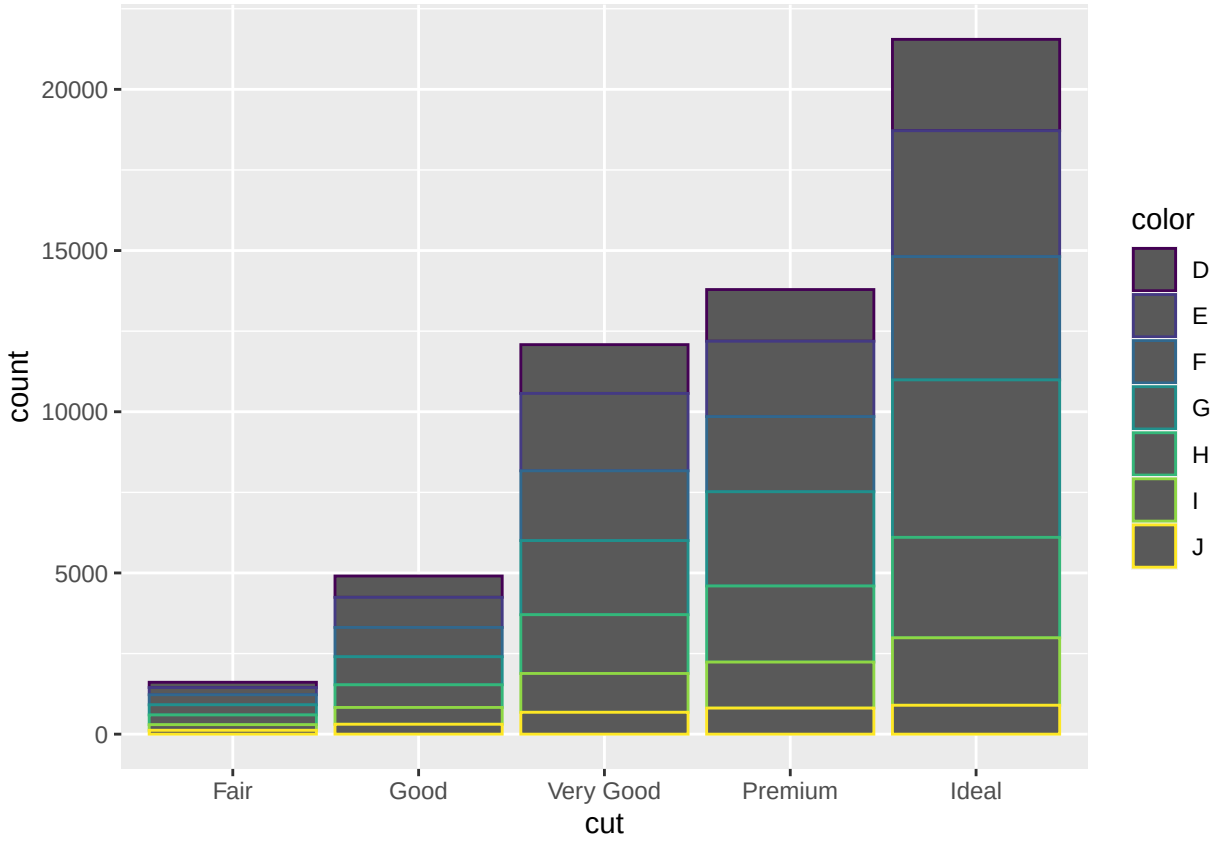


As seen above, we can see a slight dip in the graph just before the 5000mark, which is important data. However, this is only visible with binwidth set to 100. Having a binwidth too large or too small, as evidenced by the histograms with binwidth 2000 and 1, respectively, have the possibility of making the data not readable, as you cannot see that dip when viewing the histogram at those settings.

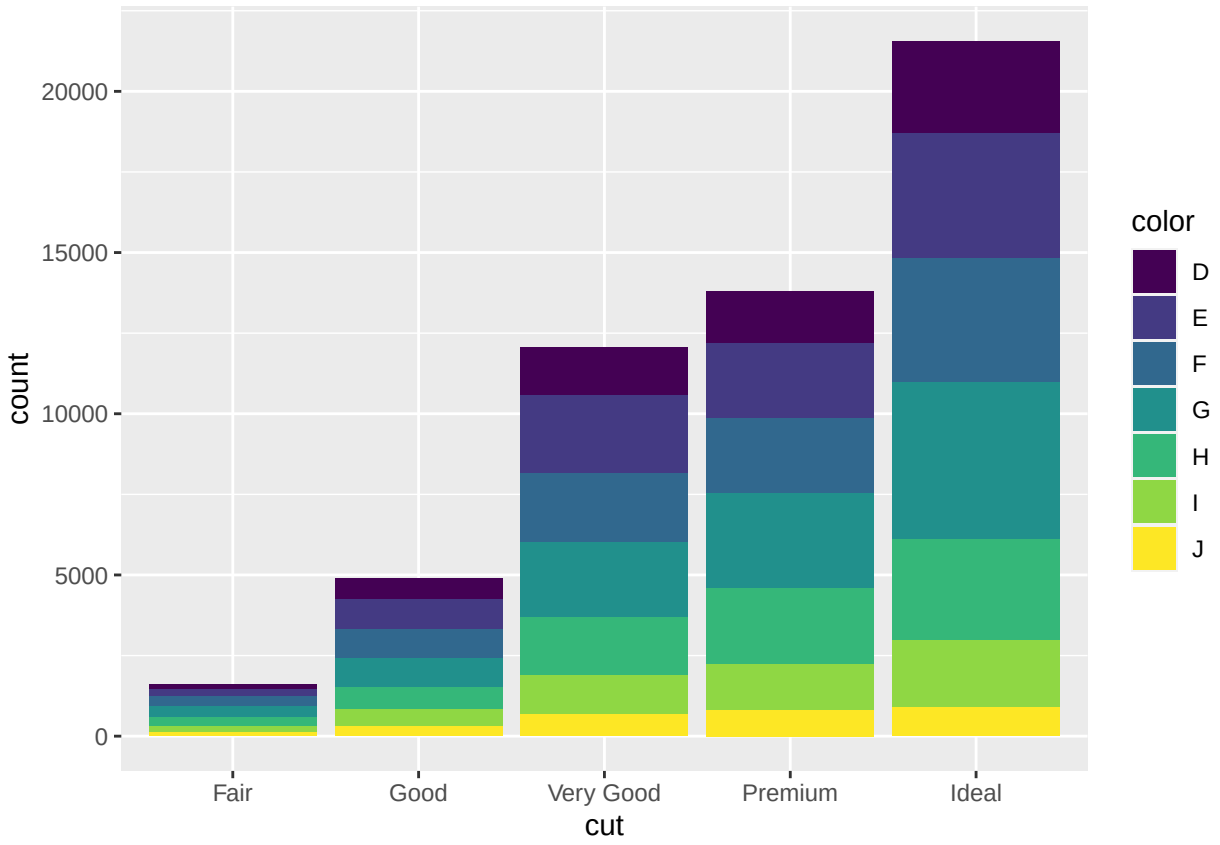
Question 5

Create a bar chart of diamond cuts. Add a color aesthetic for diamond color. Does this improve the visualization at all? Try adding a fill aesthetic for diamond color instead. Is this any better? Finally, set the position parameter to "dodge". How does the figure change?

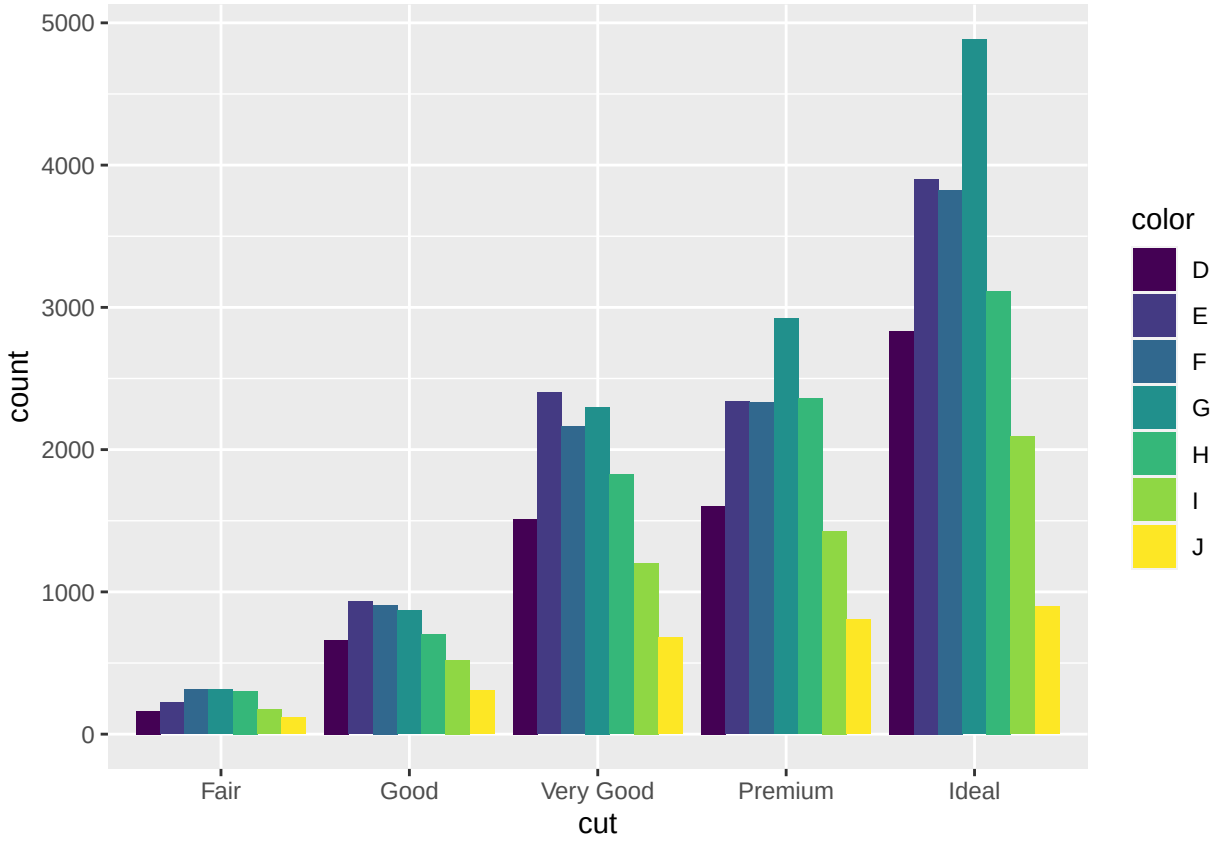
```
#color
ggplot(diamonds, aes(cut, color=color))+
  geom_bar()
```




```
#fill
ggplot(diamonds, aes(cut, fill=color))+
  geom_bar()
```




```
#dodge
ggplot(diamonds, aes(cut, fill=color))+
  geom_bar(position = 'dodge')
```

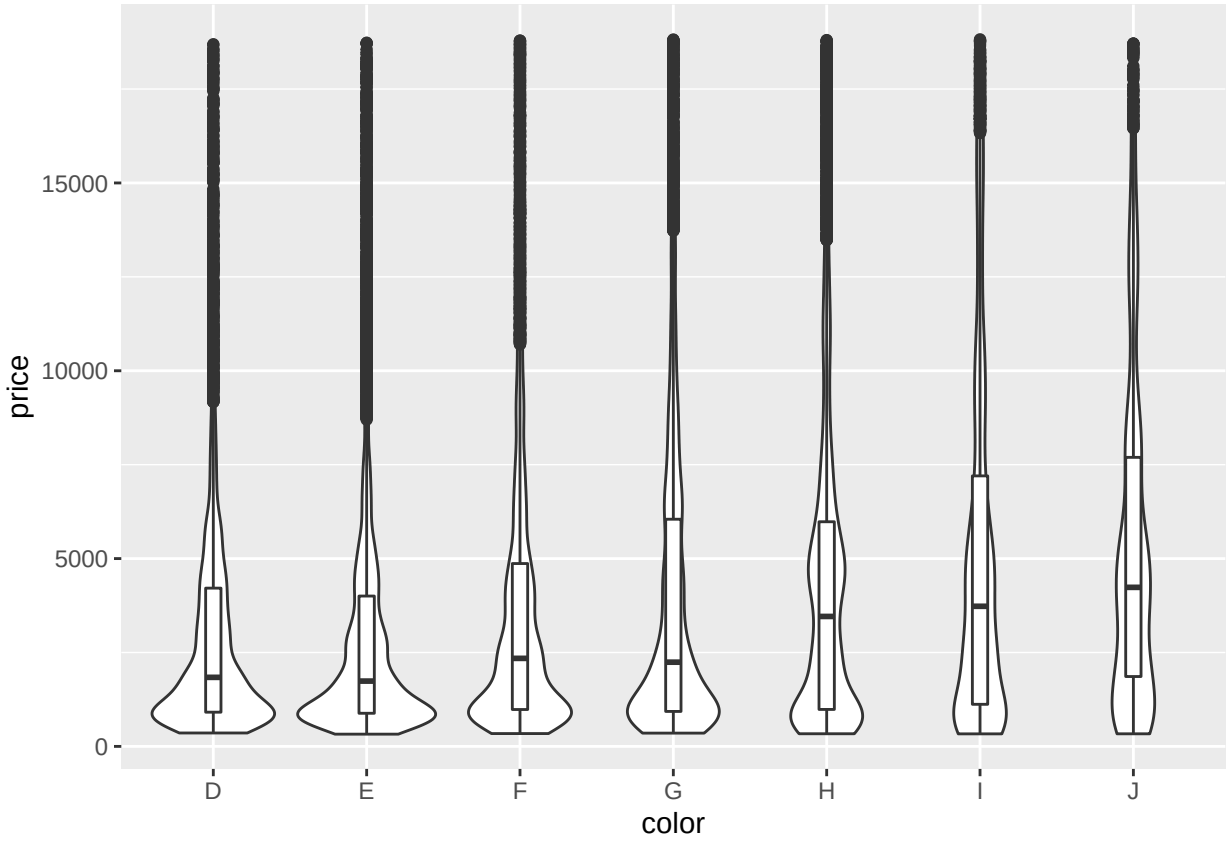


As seen above, setting color to color kind of shows us what we would want, but it really is not very readable, as the colors are just lines in the bar. Fill, however, separates the bars into distinct sections, which gives the chart much more readability and is better understood. Likewise, dodge further increases this understanding by offsetting the bars and their respective colors, so that we can see the counts sorted by both cut and color.

Question 6

Make a violin plot comparing diamond color and price. Add a boxplot on top of this figure comparing the same variables. Play with the boxplot width parameter so that the boxplots fit inside the violin plots.

```
ggplot(diamonds, aes(color, price))+
  geom_violin()+
  geom_boxplot(width = 0.1)
```



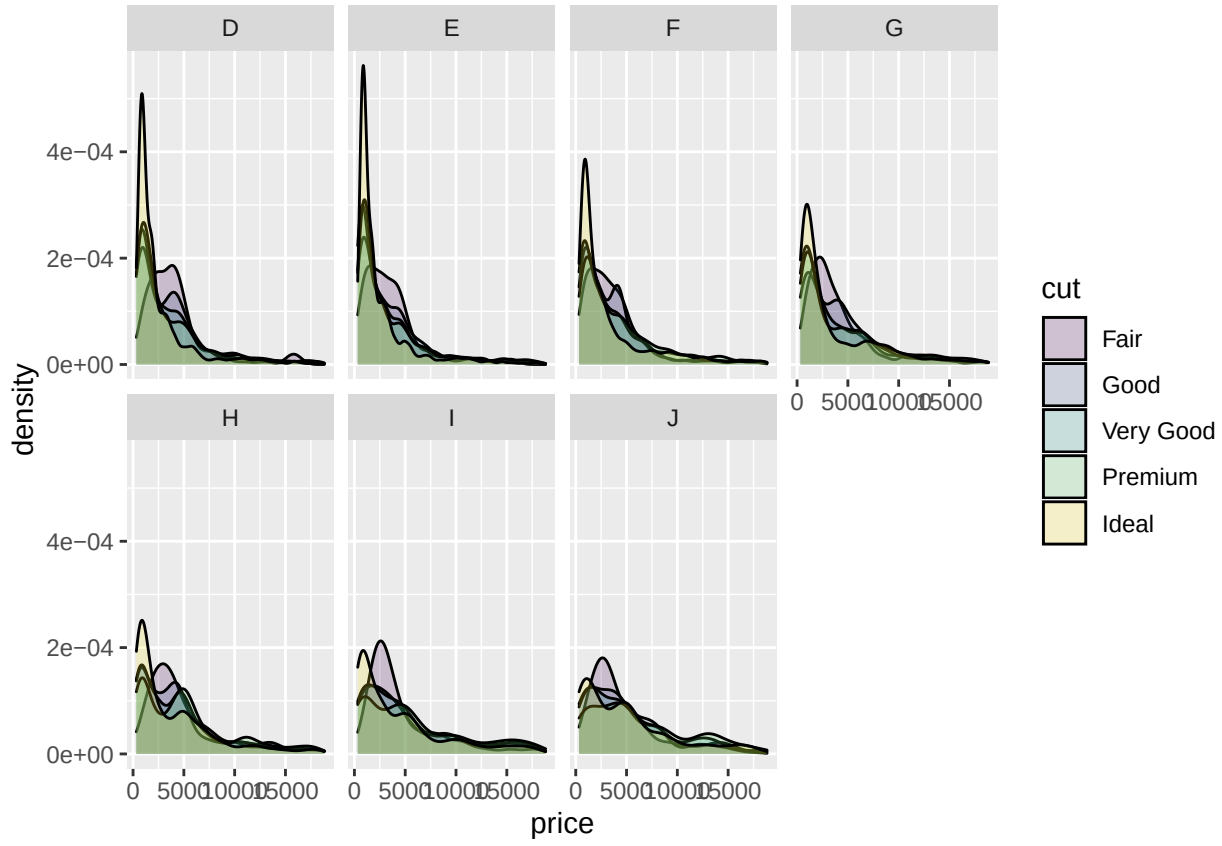
When creating the violin and boxplots, I had to shrink the width of the boxplots to a scale of 1/10, or .1, to get them to fit in the violins

Question 7

Write R code to reproduce the figure below. The density geometric object might be useful.

[Figure is in pdf documentation of lab assignment]

```
ggplot(diamonds, aes(price, fill = cut))+
  geom_density(alpha = .2)+
  facet_wrap(~color, nrow = 2)
```



Question 8

Import the data from billboard.csv as a tibble and make it tidy. Remember that, in tidy data, each variable has its own column, each observation has its own row, and each value has its own cell. Rename the artist.inverted column to artist and the track column to song. Make a tibble containing the columns year, artist, song, genre, and time along with information related to week on the chart and rank.

```
billboard<-read_csv("billboard.csv")
```

```
## Rows: 317 Columns: 83
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): artist.inverted, track, time, genre
## dbl  (66): year, x1st.week, x2nd.week, x3rd.week, x4th.week, x5th.week, x6th...
## lgl  (11): x66th.week, x67th.week, x68th.week, x69th.week, x70th.week, x71st...
## date  (2): date.entered, date.peaked
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#pivot to create the rank variable

tidy_board<-billboard %>%
  pivot_longer(
    x1st.week:x76th.week,
    names_to = "week",
    values_to = "rank"
  )

#rename variables artist, track

tidy_board<-setNames(tidy_board, replace(names(tidy_board), names(tidy_board) == 'artist.inverted', 'ar

tidy_board<-setNames(tidy_board, replace(names(tidy_board), names(tidy_board) == 'track', 'song'))

tidy_board
```

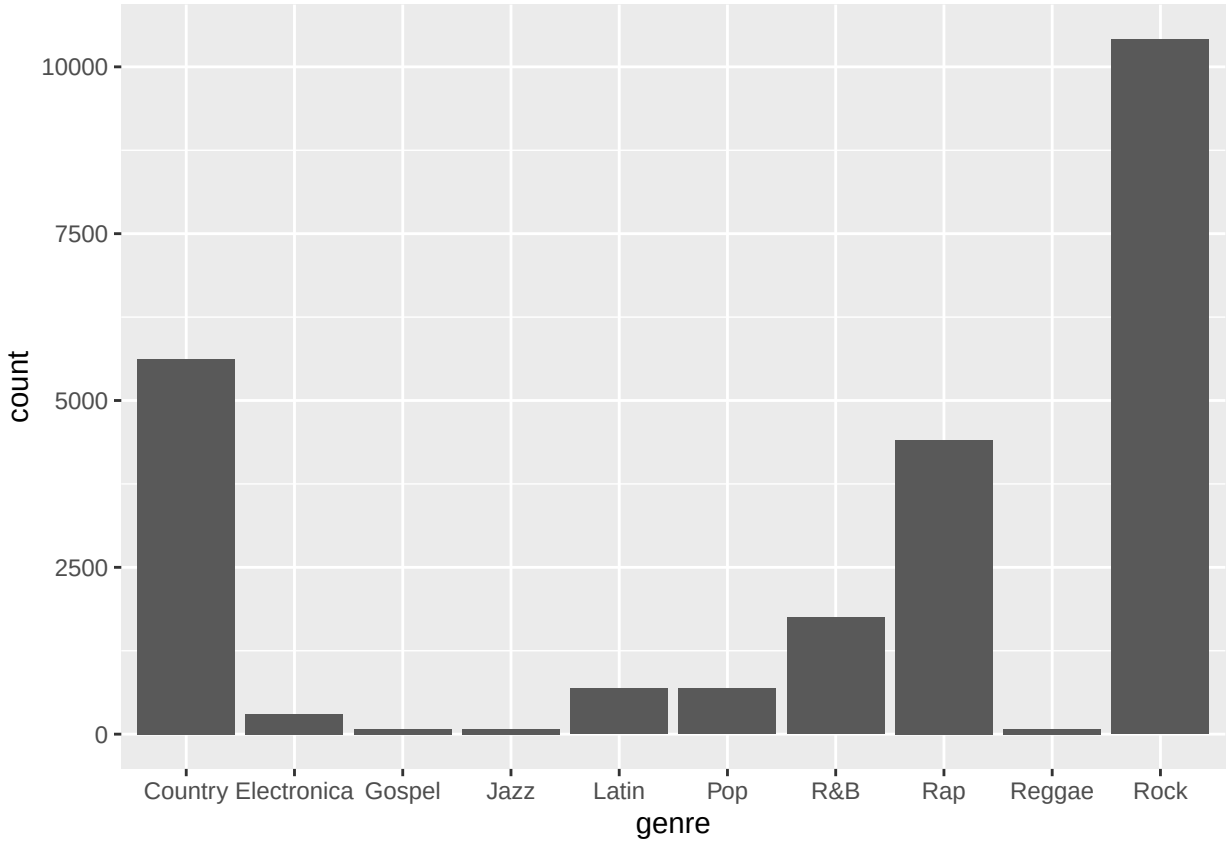
```
## # A tibble: 24,092 x 9
##     year artist          song   time  genre date.entered date.peaked week   rank
##    <dbl> <chr>           <chr>  <chr> <chr> <date>       <date>      <chr> <dbl>
##  1  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x1st~    78
##  2  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x2nd~    63
##  3  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x3rd~    49
##  4  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x4th~    33
##  5  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x5th~    23
##  6  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x6th~    15
##  7  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x7th~     7
##  8  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x8th~     5
##  9  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x9th~     1
## 10  2000 Destiny's Child Indep~ 3:38  Rock  2000-09-23   2000-11-18  x10t~     1
## # ... with 24,082 more rows
```

As seen above, all the necessary changes were made to billboard, and were put into a dataframe called "tidy_board"

Question 9

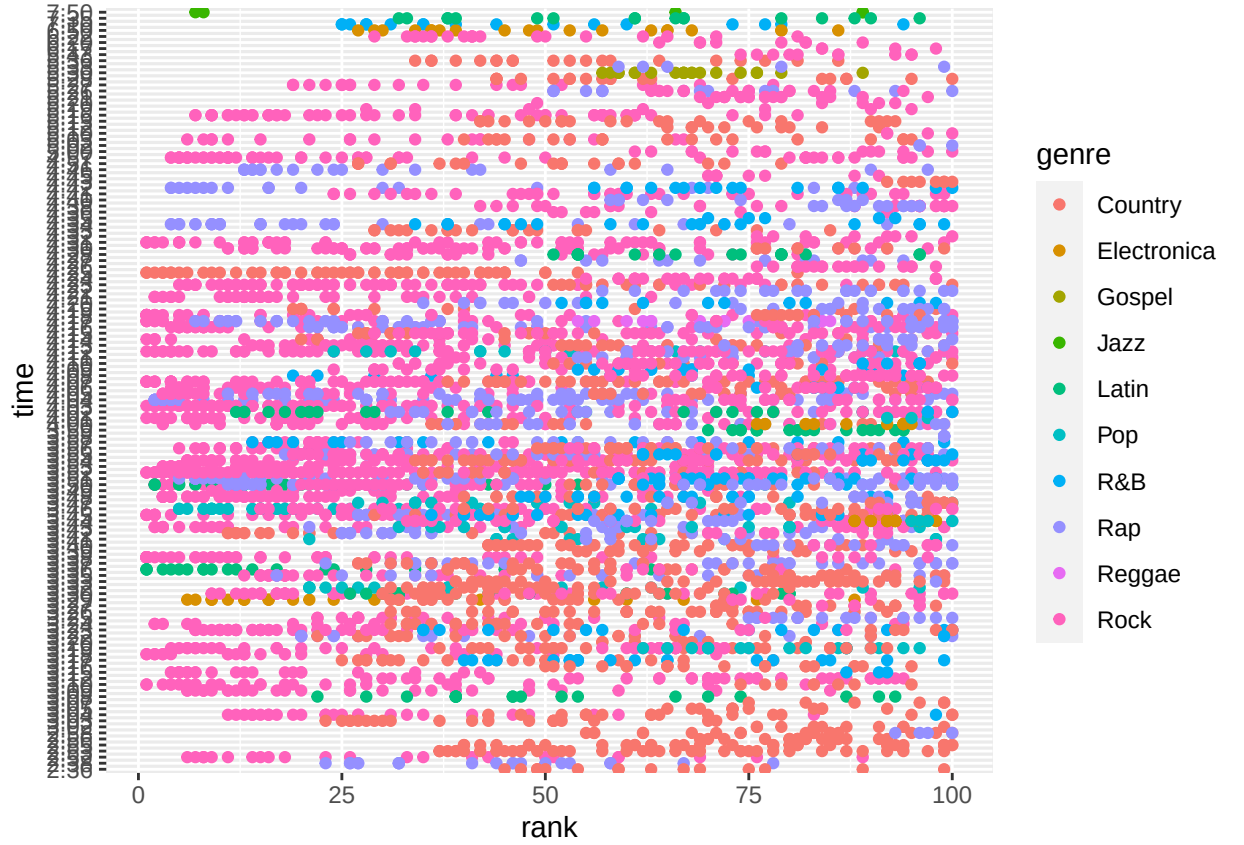
Create a bar chart of song genres. Create a separate scatterplot with rank on the x-axis, time on the y-axis, and a color aesthetic based on genre. What causes the horizontal lines of points in this second figure?

```
#barchart
ggplot(tidy_board, aes(genre))+
  geom_bar()
```




```
#scatterplot
ggplot(tidy_board, aes(rank, time, color=genre))+
  geom_point()
```

```
## Warning: Removed 18785 rows containing missing values (geom_point).
```



The cause of the massive amount of data points that form a vertical line is songs that are listed as 'N/A' on the charts, so they appear off to the side, effectively "below 0." There is many of these, and they have varying lengths, which is what the time variable measures.

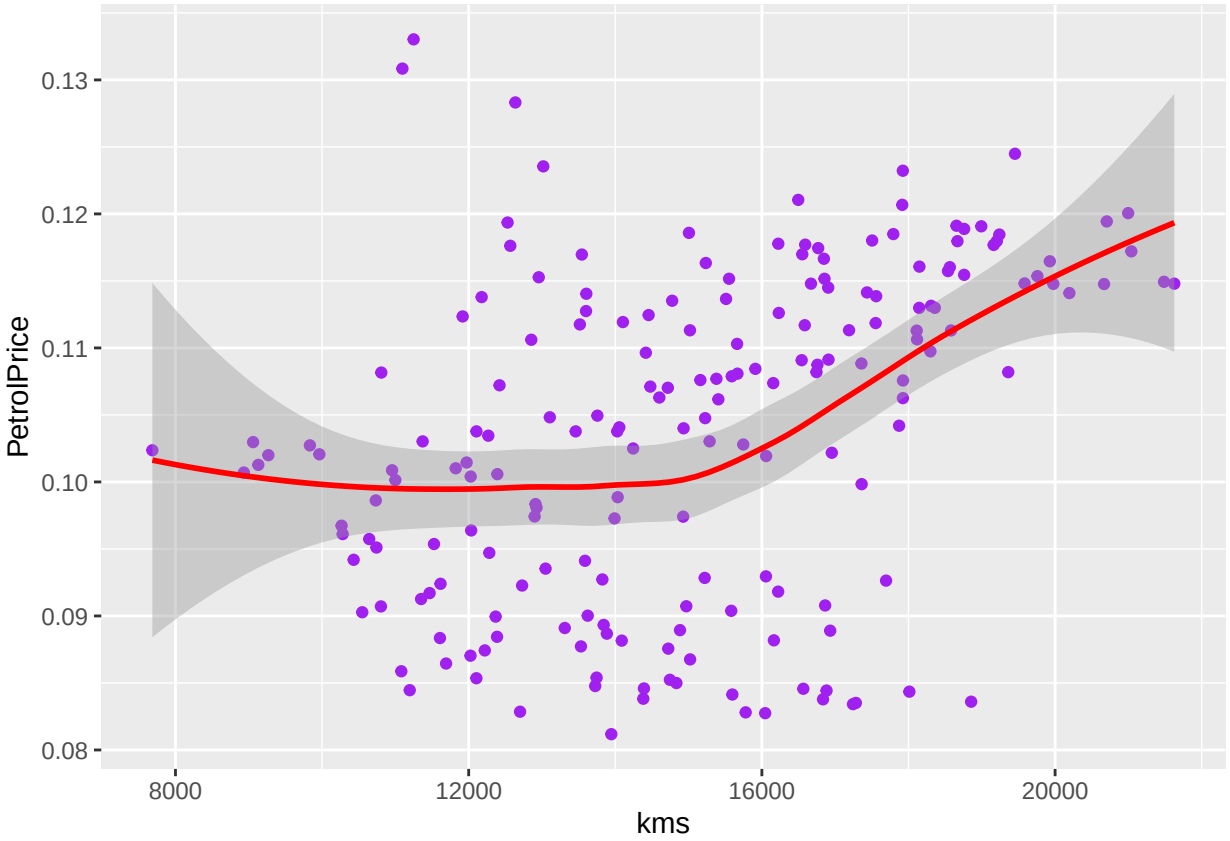
Question 10

Pick any data set from data() or from the tidyverse. Using this data set, make at least two figures. One should be a scatterplot with a smoothed curve. The other can be whatever you would like, be creative!

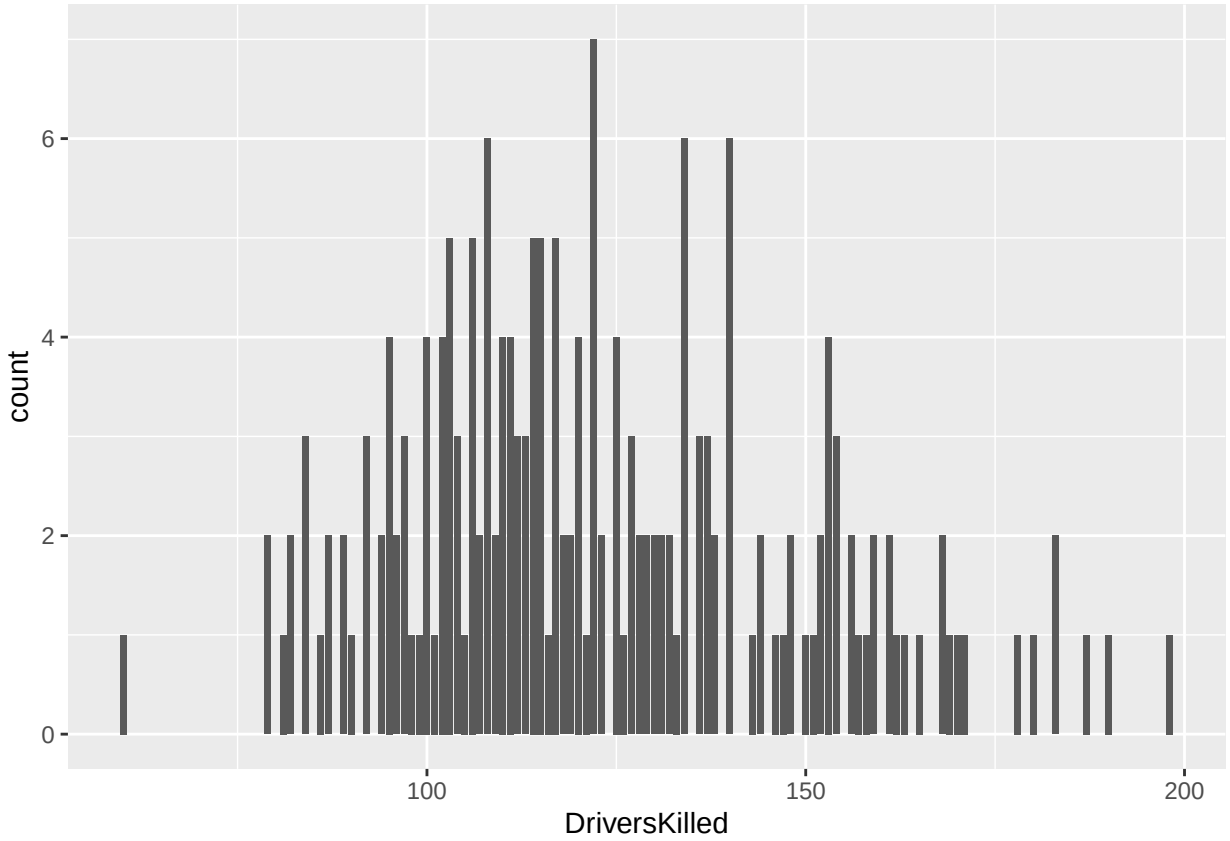
```
#using dataset "seatbelts"

seatdata <- data.frame(Seatbelts)
#scatterplot with smooth curve for petrol price vs distance driven
ggplot(seatdata, aes(kms, PetrolPrice))+
 geom_point(color = "purple")+
  geom_smooth(color = "red")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#bar chart for driver deaths
ggplot(seatdata, aes(DriversKilled))+
  geom_bar()
```


count
6
4
2
0
100
150
200
DriversKilled