Homework 2

Ngaya Swai

Due 9/14/2021

Classmates/other resources consulted:

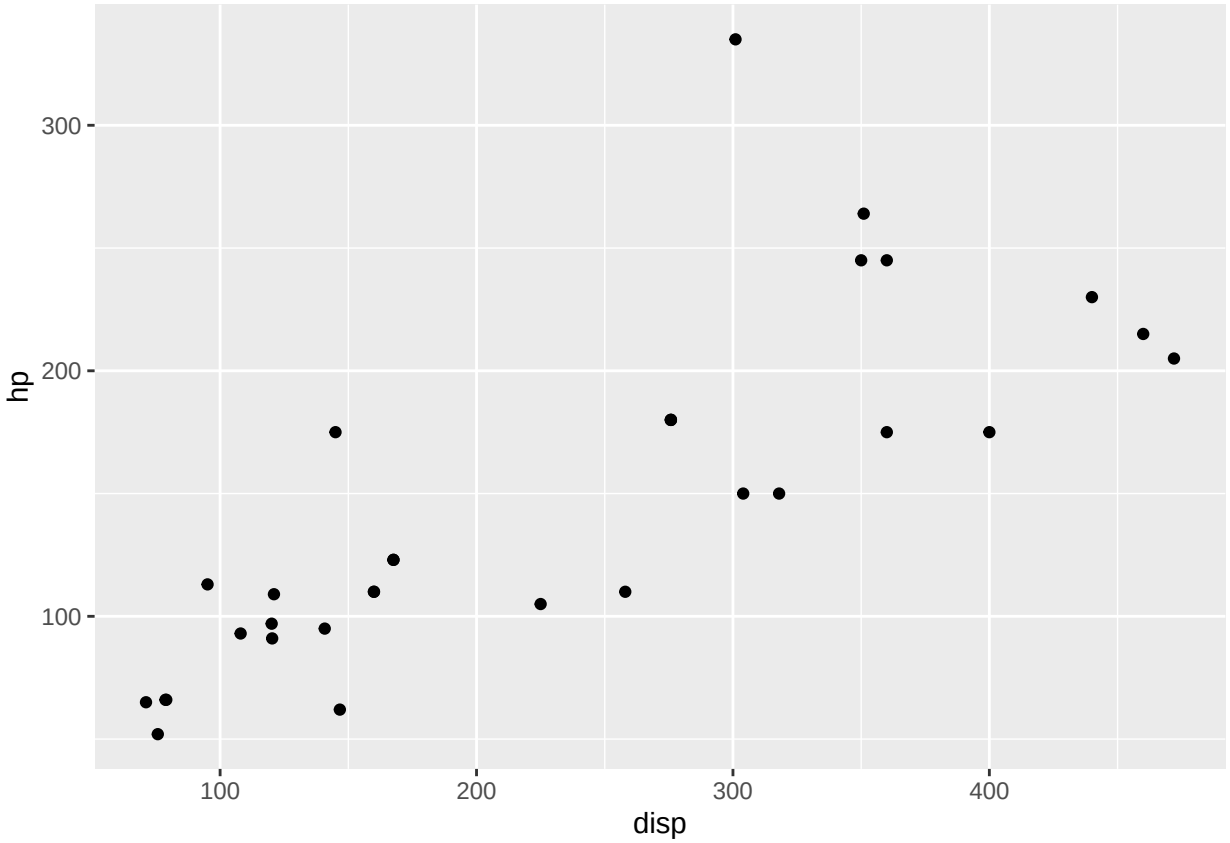
https://www.datacamp.com/community/tutorials/make-histogram-ggplot2 https://www.rdocumentation.org/packages/ggExtra/versions/0.9/topics/ggMarginal

```
library(ggplot2)
```

Question 1 (12 points)

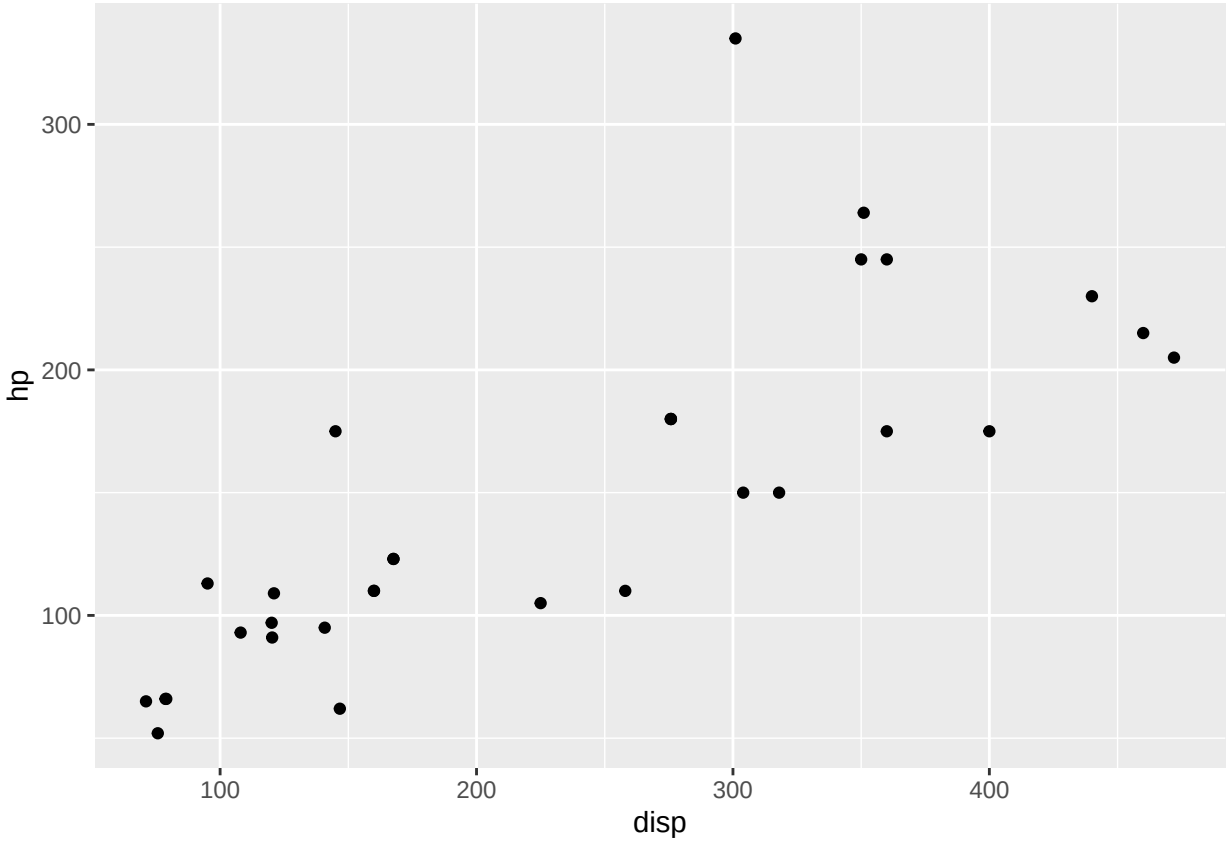
a. **Make a scatterplot of the Displacement (in cubic inches) vs. Gross horsepower of the cars in the mtcars data set.**

```
ggplot(data = mtcars) + geom_point(mapping = aes(x = disp, y = hp))
```

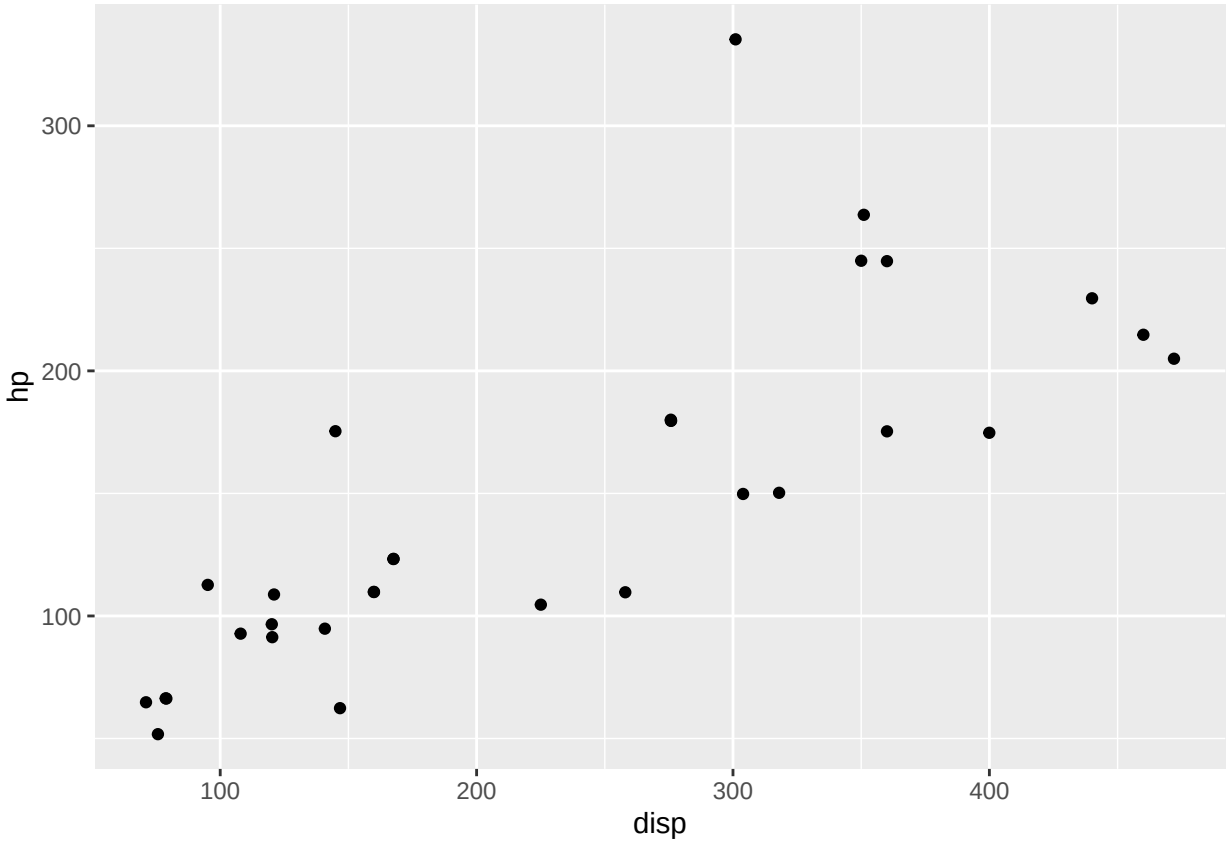


b. **Are there any data points in this data set that fall directly on top of each other? Explain how you know.**

```
ggplot(data = mtcars) + geom_point(mapping = aes(x = disp, y = hp))
```




```
ggplot(data = mtcars) + geom_point(mapping = aes(x = disp, y = hp), position = "jitter")
```

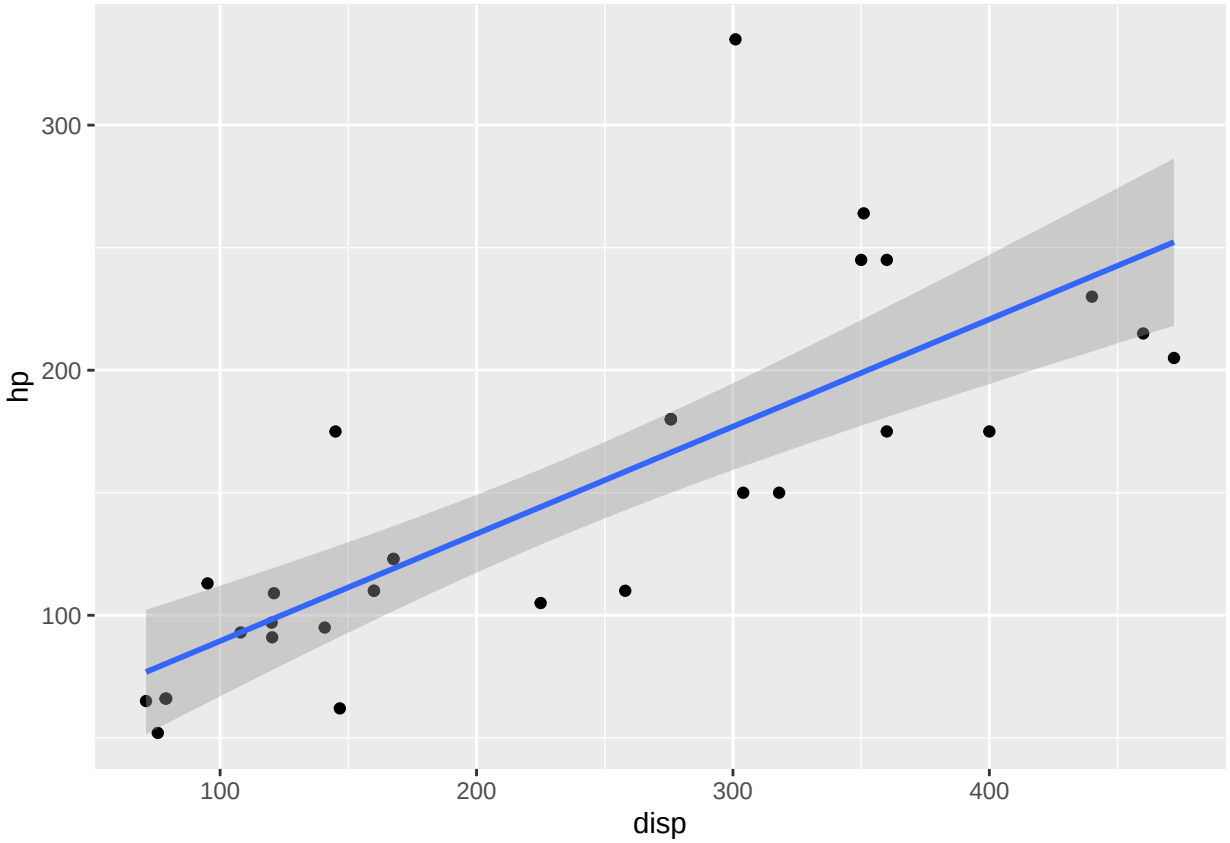


We know because when ussing the jitter position there are no additional data points shown.

c. **Add another layer to your plot from (c); this new layer should consist of a curve that best fits the data**

```
ggplot(data = mtcars, aes(x= disp, y = hp)) +
  geom_point() +
  geom_smooth(method = "lm")
```

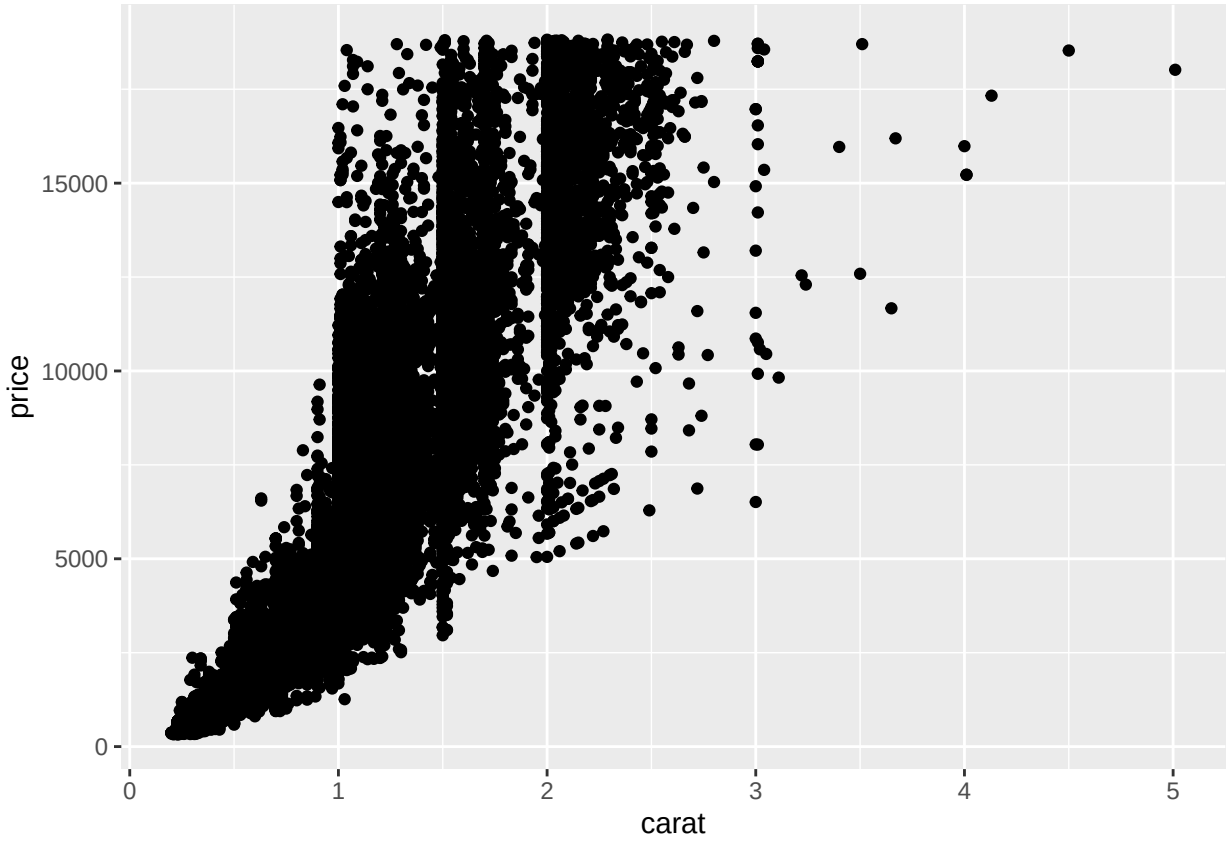
```
## `geom_smooth()` using formula 'y ~ x'
```


300
200
100
hp
100
200
300
400
disp

Question 2 (12 points)

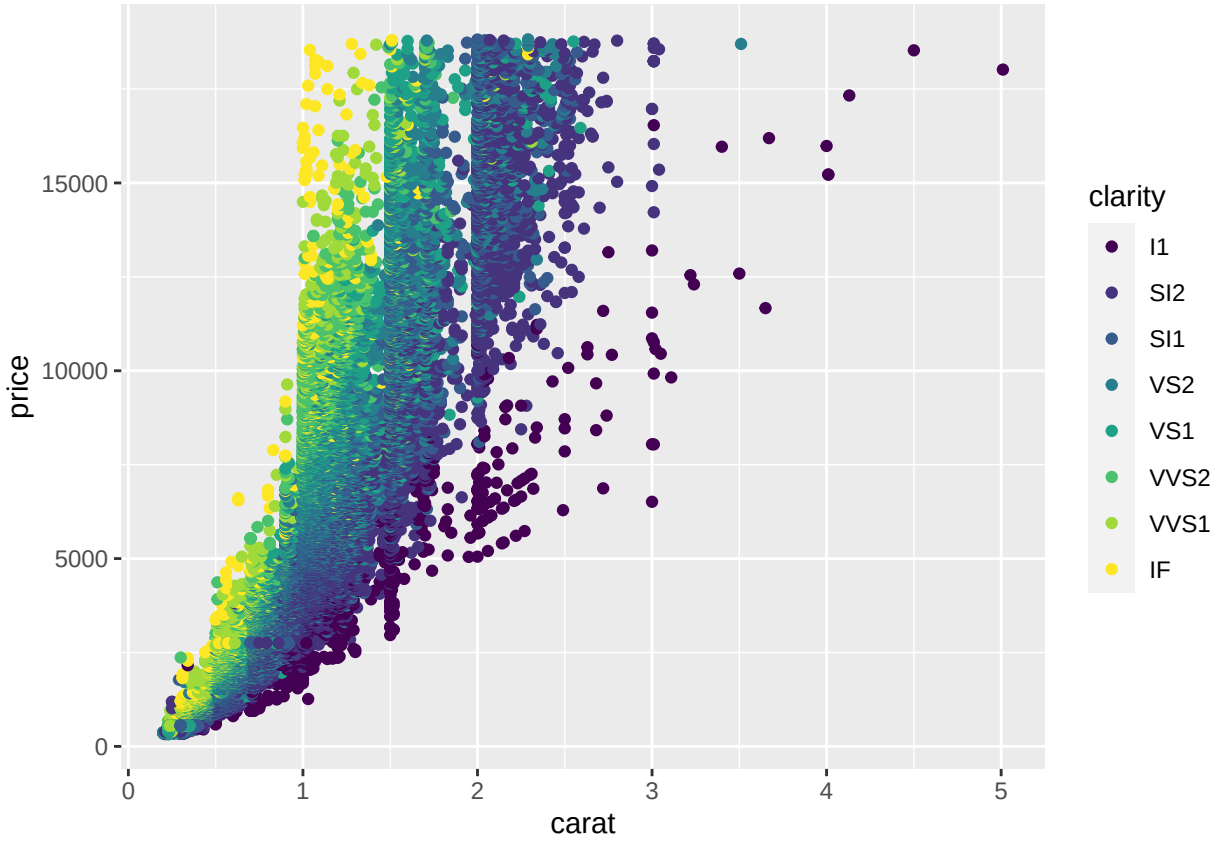
Consider the following scatterplot of price vs. carat of diamonds

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



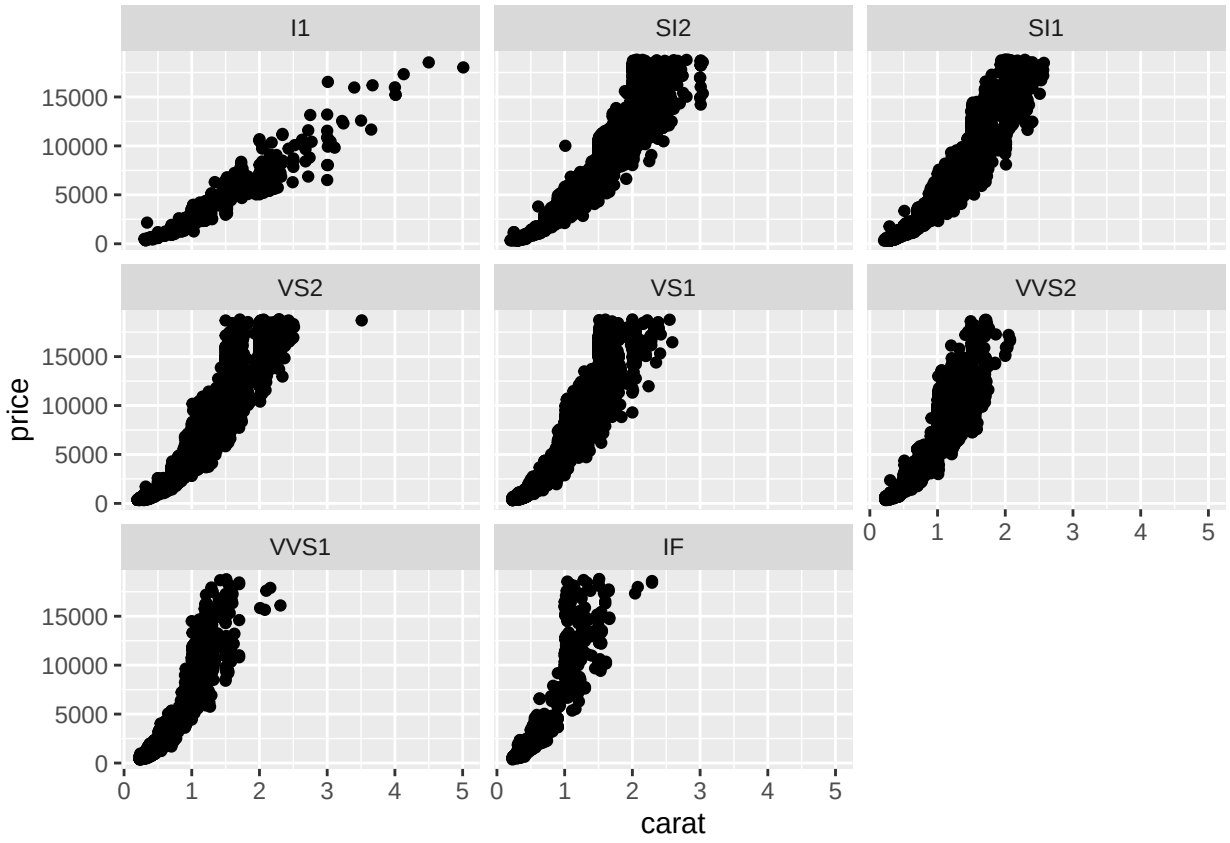
a. **Distinguish your data points based on the clarity of the diamonds, using whatever method you'd like (that is, the data points corresponding to one clarity level should look different than the data points corresponding to another clarity level, etc.)**

```
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price, color = clarity))
```



b. **Separate your data points into several distinct plots, one for each different clarity level**

```
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price)) +
  facet_wrap(~clarity)
```



c. **For each of the following questions, say whether the plot from (a) or the plot from (b) would be preferable for answering that question (you do not need to actually answer the questions, just say which plot would be more helpful for answering it):**

i. **What is the general trend of price vs. carats for the data set as a whole?**

Plot from A

ii. **Do all different clarity levels show similar trends for carats vs. price?**

Plot from B

iii. **For a given number of carats, what clarity level tends to be most expensive?**

Plot from A

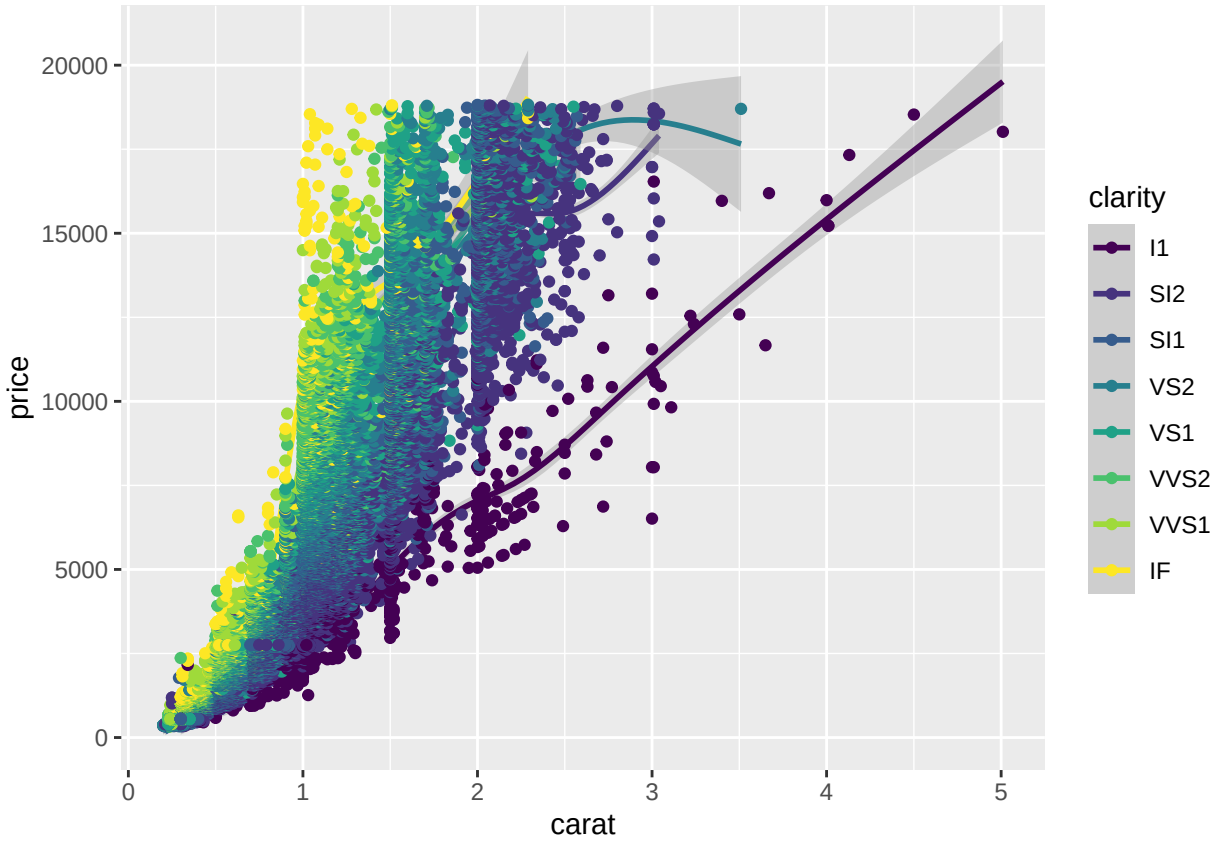
iv. **For what clarity level is the relationship between carats and price strongest?**

Plot from B

d. **Make a plot that shows the relationship between carats and price by giving a collection of smooth curves, one for each clarity level.**

```
ggplot(data = diamonds, mapping = aes (x = carat, y = price, color = clarity)) +
  geom_smooth() +
  geom_point()
```

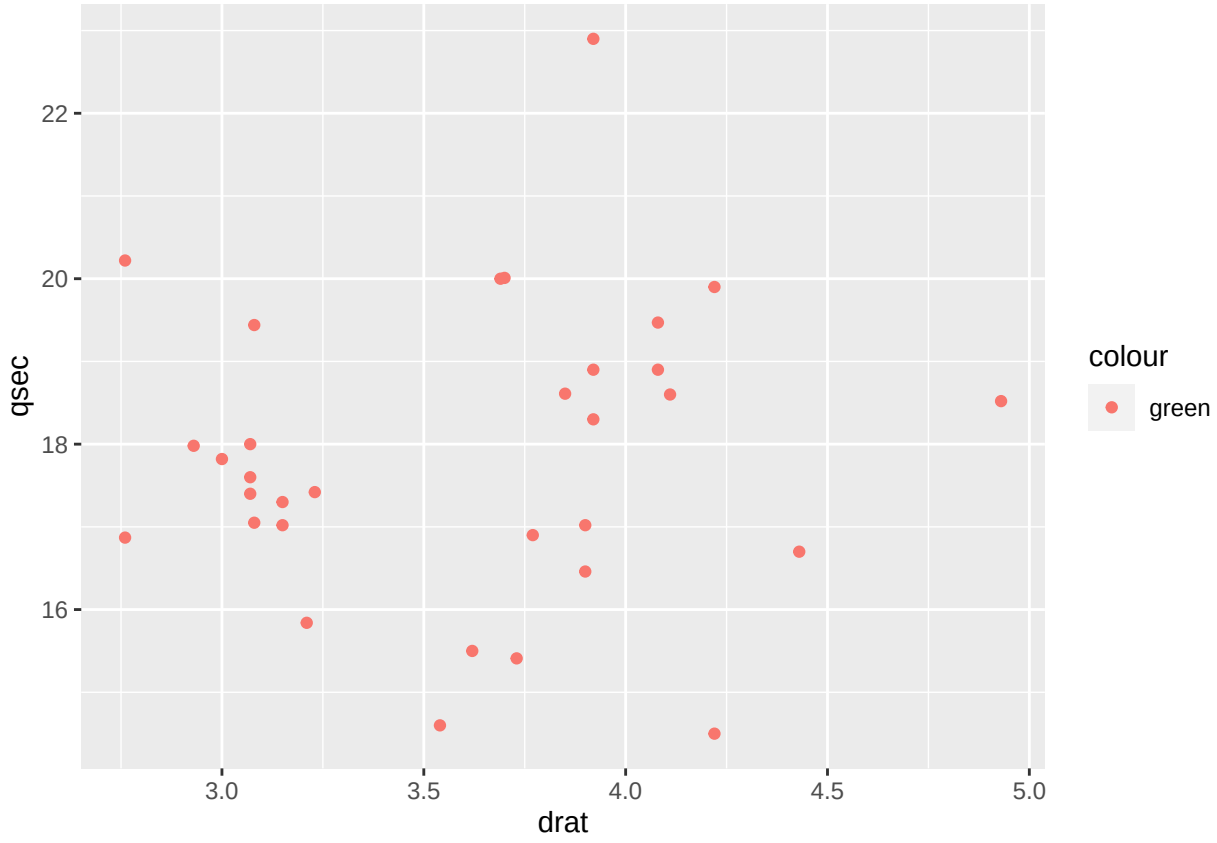
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Question 3 (4 points)

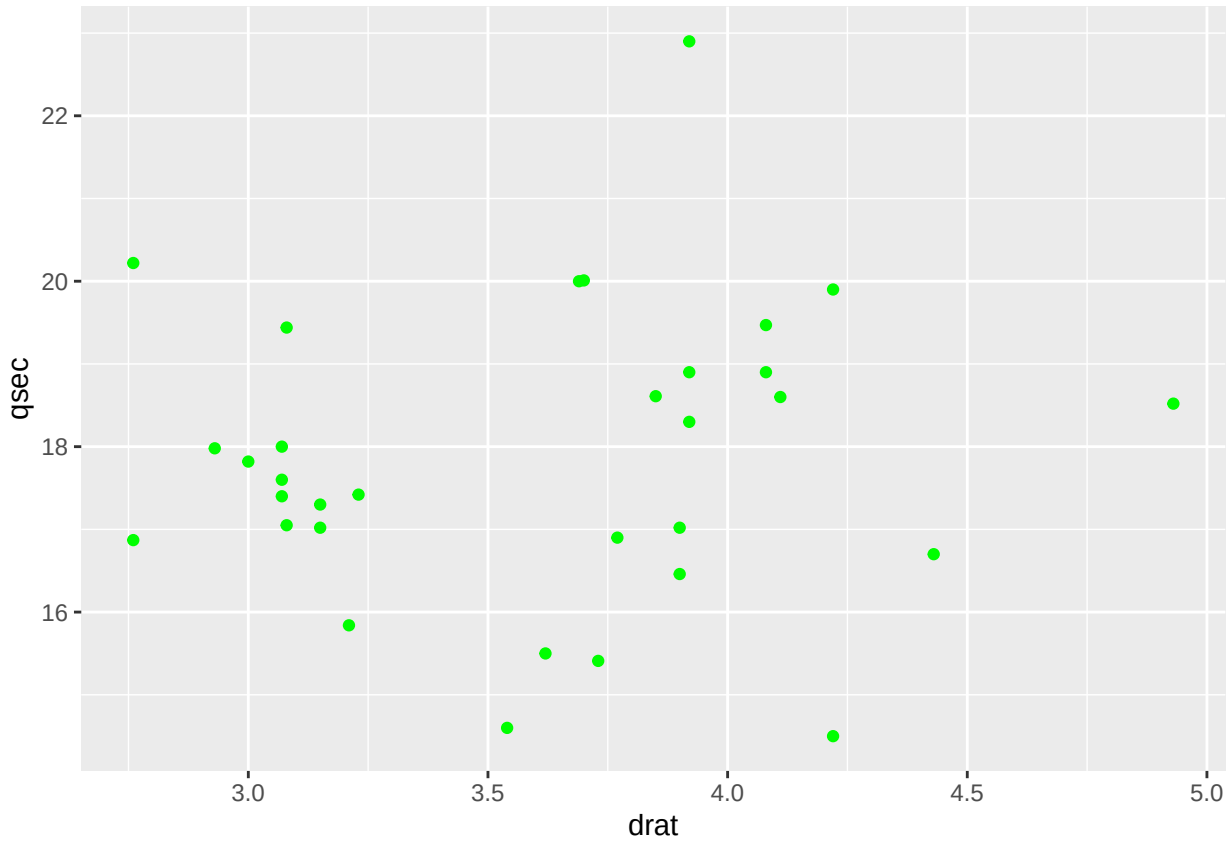
What has gone wrong with this plot, and how would you change it to make all points green?

```
ggplot(data = mtcars) + geom_point(mapping = aes(x = drat, y = qsec, color = "green"))
```



The color = green was inside the aesthetic so it won't change the color of all of the points.

```
ggplot(data = mtcars) + geom_point(mapping = aes(x = drat, y = qsec), color = "green")
```


22
20
18
16
qsec
3.0
3.5
4.0
4.5
5.0
drat

Question 4 (12 points)

a. **Will the following two commands produce the same output? Explain why or why not.**

ggplot(data = mpg) + geom_point(mapping = aes(x = hwy, y = cty))

ggplot(mpg) + geom_point(aes(x = hwy, y = cty))

They will produce the same output because the second command simply removed the data = and the mappin =, which are the defaults of the functions.

b. **Will the following two commands produce the same output? Explain why or why not.**

ggplot(data = mpg, mapping = aes(x = hwy, y = cty), size = 1.5) + geom_point()

ggplot(data = mpg) + geom_point(mapping = aes(x = hwy, y = cty), size = 1.5)

Yes, the formatting is just moved around to included the geom_point after aesthetic.

c. **Will the following two commands produce the same output? Explain why or why not.**

ggplot(data = mtcars) + geom_point(mapping = aes(x = drat, y = qsec), size = 1.5) ggplot(data = mtcars) + geom_count(mapping = aes(x = drat, y = qsec), size = 1.5)

Yes, even though the command uses a point instead of a count, it'll still produce the same output.

Question 5 (10 points)

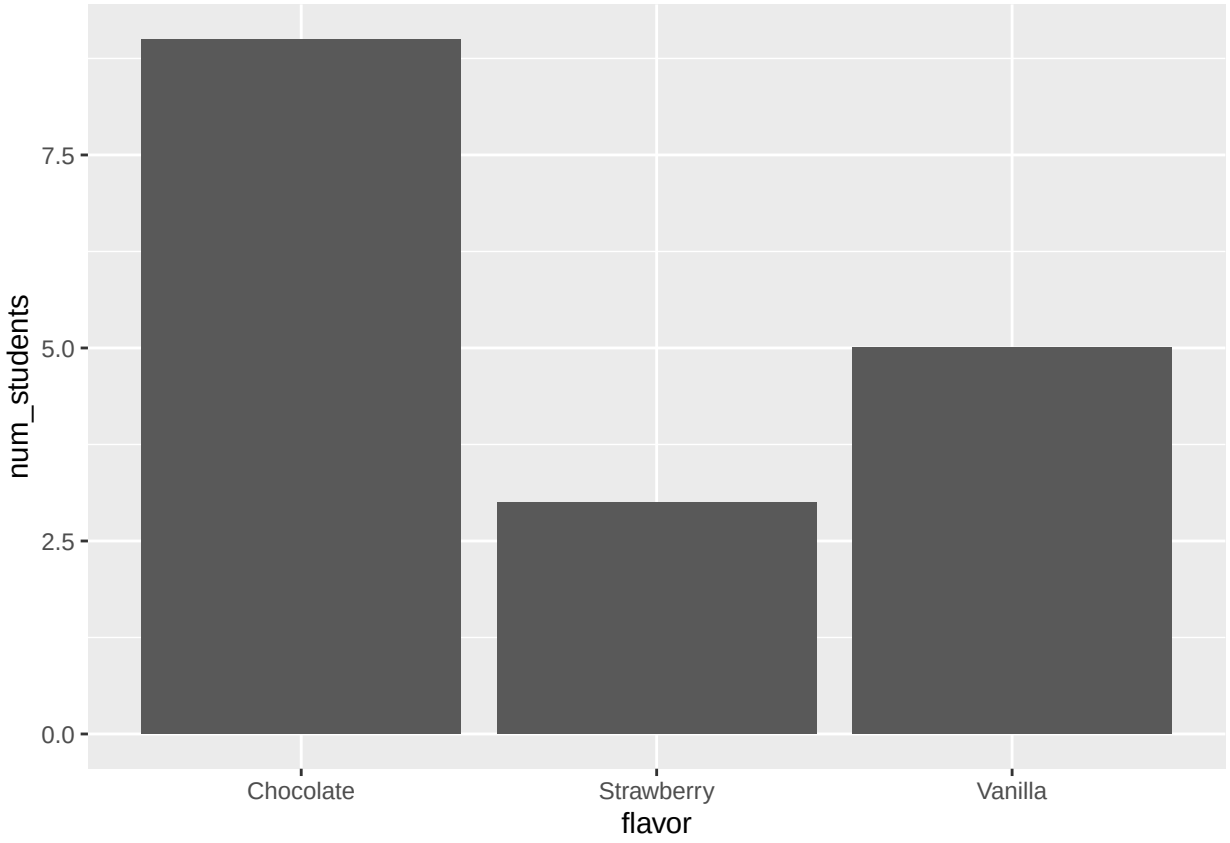
Suppose you have data about students' preferences for ice cream represented in a table, and you want to make a bar chart.

a. **First, suppose your data table has two columns, with the type of ice cream in a column named 'flavor' and the number of students who say that flavor is their favorite in a column named 'num_students'. How would you make a bar chart that shows how many students prefer each type of ice cream?**

Here is a sample table if you want to try out your command:

```
library(tibble)
ice_cream_a = tibble(flavor = c("Vanilla", "Chocolate", "Strawberry"), num_students = c(5,9,3))

ggplot(data = ice_cream_a) + geom_bar(aes(x = flavor, y = num_students), stat = "identity")
```

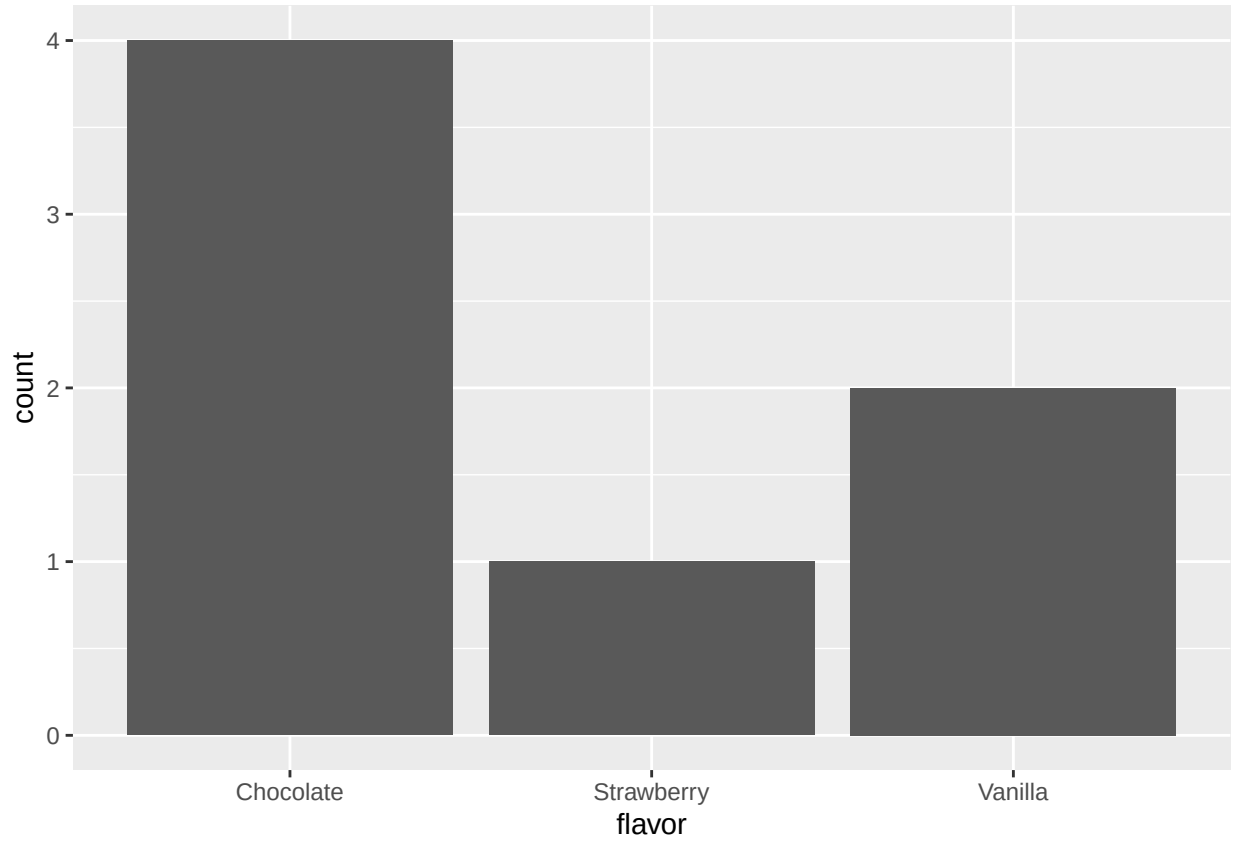


b. **Now, suppose your table has a row for each student. The first column, named 'student', has each student's name, and the second column, named 'flavor', has that student's favorite ice cream flavor. How would you make a bar chart that shows how many students prefer each type of ice cream?**

Here is a sample table if you want to try out your command:

```r
ice_cream_b = tibble( name = c("Student A", "Student B", "Student C",
                        "Student D", "Student E", "Student F", "Student G"),
                      flavor = c("Chocolate", "Vanilla", "Chocolate", "Strawberry",
                        "Vanilla", "Chocolate", "Chocolate"))

ggplot(ice_cream_b) + geom_bar(aes(x = flavor))
```



Question 6 (4 points)

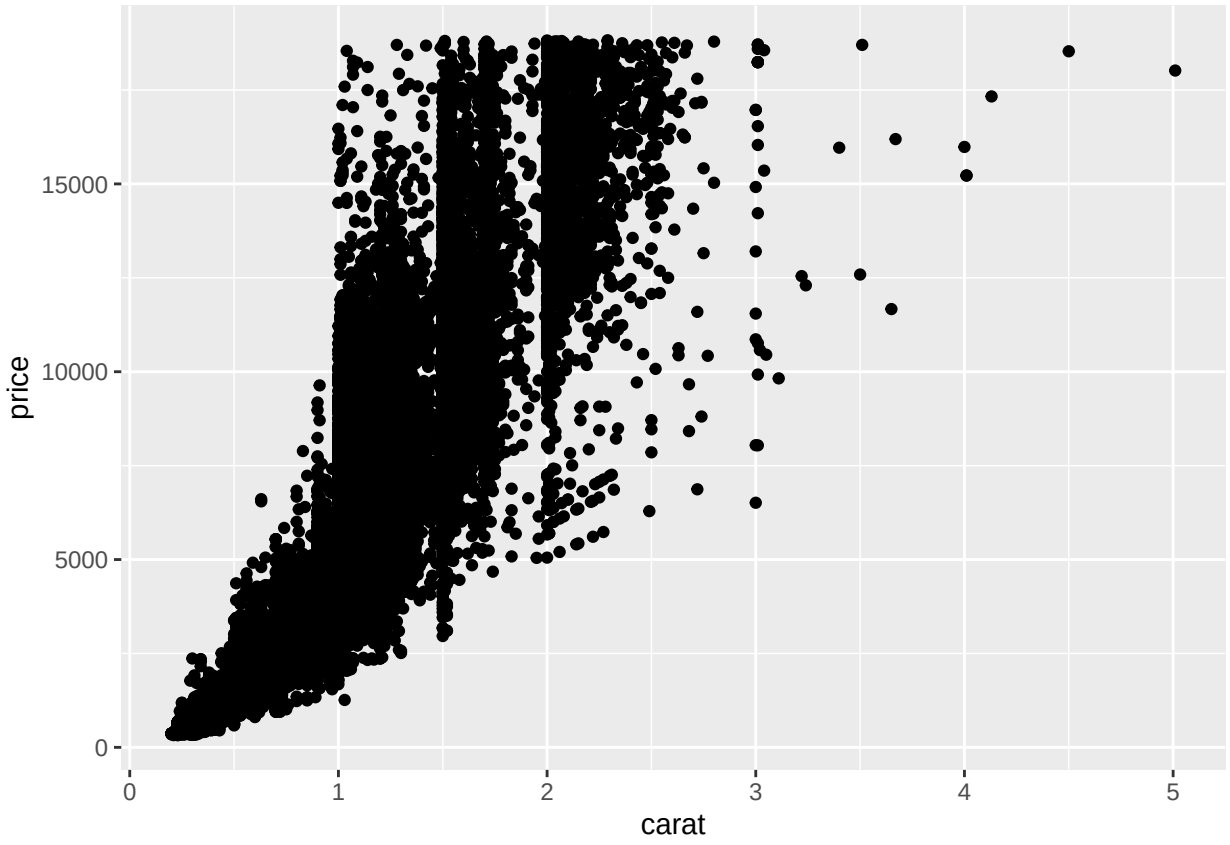
What does geom_col() do? How is it different from geom_bar()?

geom_col creates a table version of the data, requiring a y variable. Whereas, geom_bar can work simply off the count of the x variable.

Question 7 (8 points)

You can change the style of your plot by adding a theme. For each of the following plot themes, try it out and describe in words what it does. You can use the plot from earlier as an example if you'd like:

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



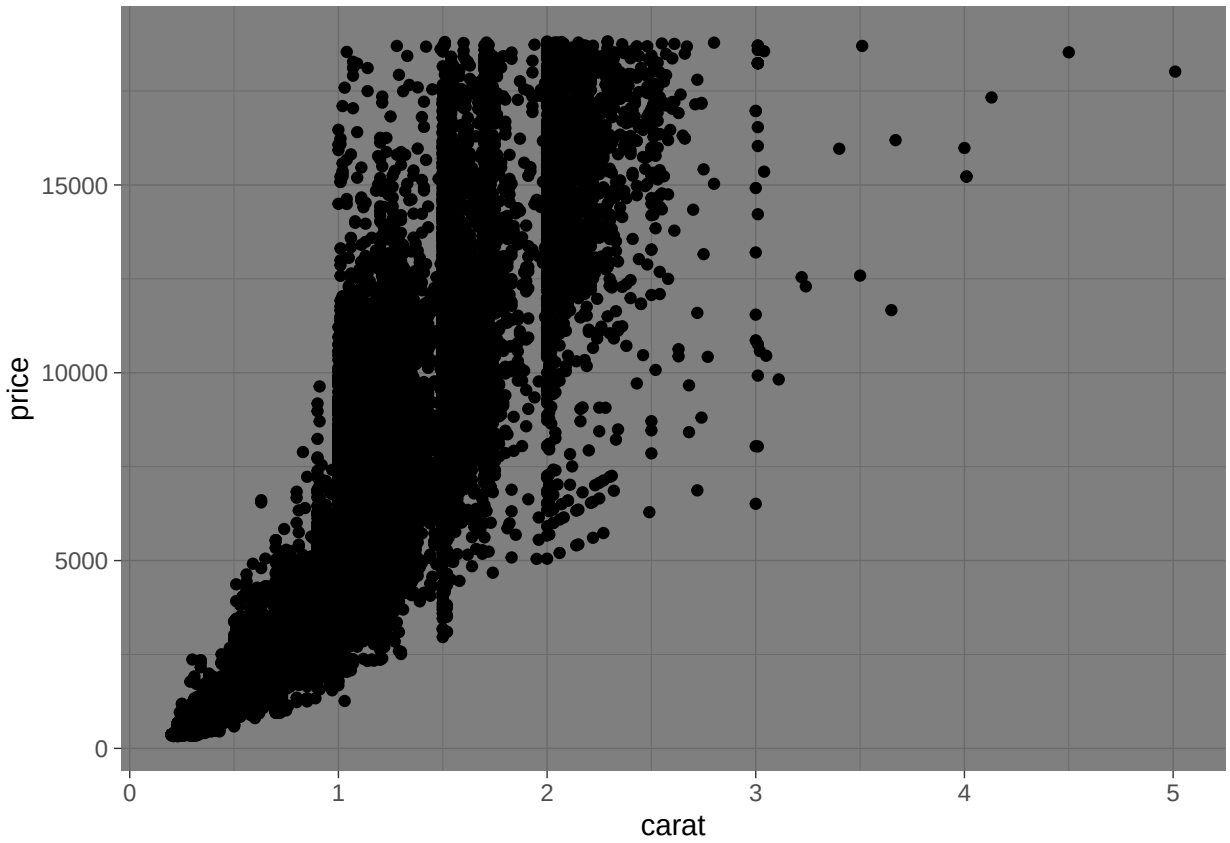
a. **theme_void()**

```
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price)) + theme_void()
```

This removes all of the background and labels, leaving just the raw data points.

b. **theme_dark()**

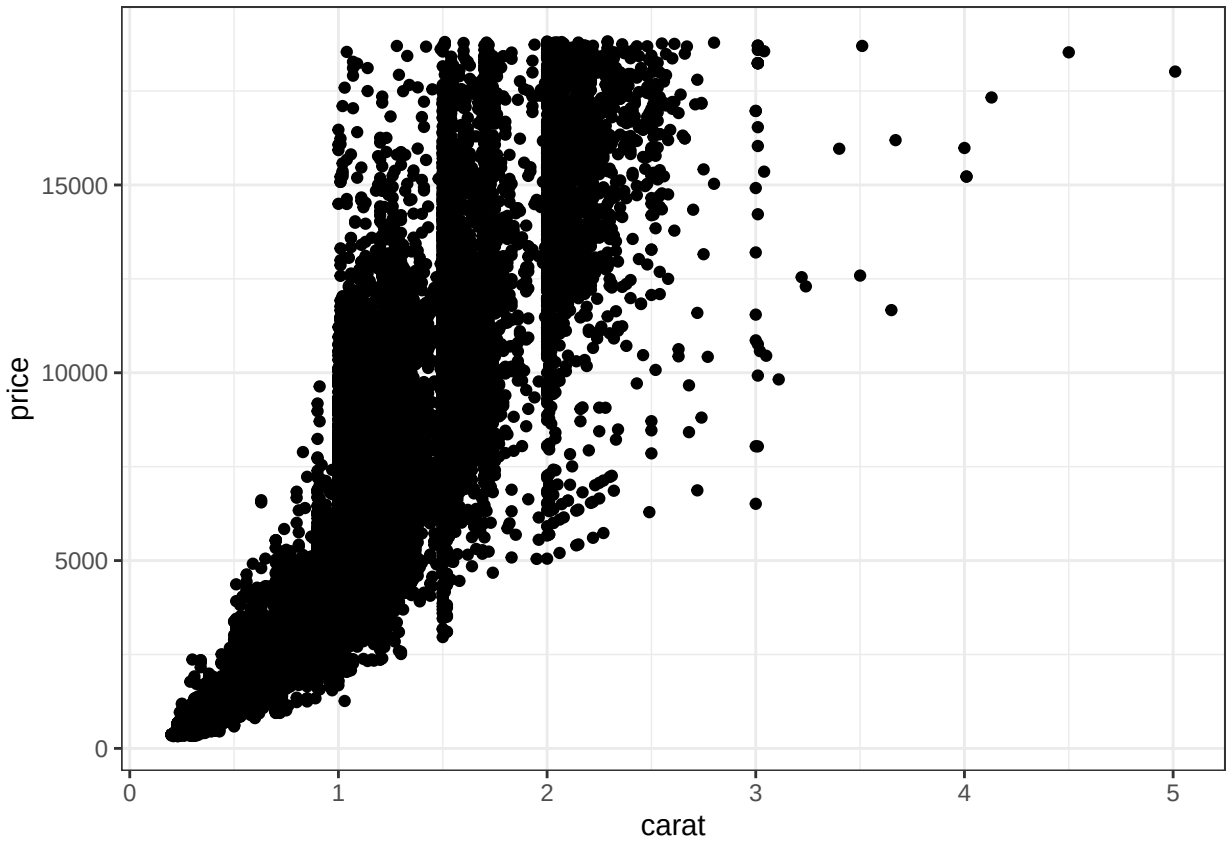
```
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price)) + theme_dark()
```



This darkens the background turning it from a light grey to a dark grey background.

c. **theme_bw()**

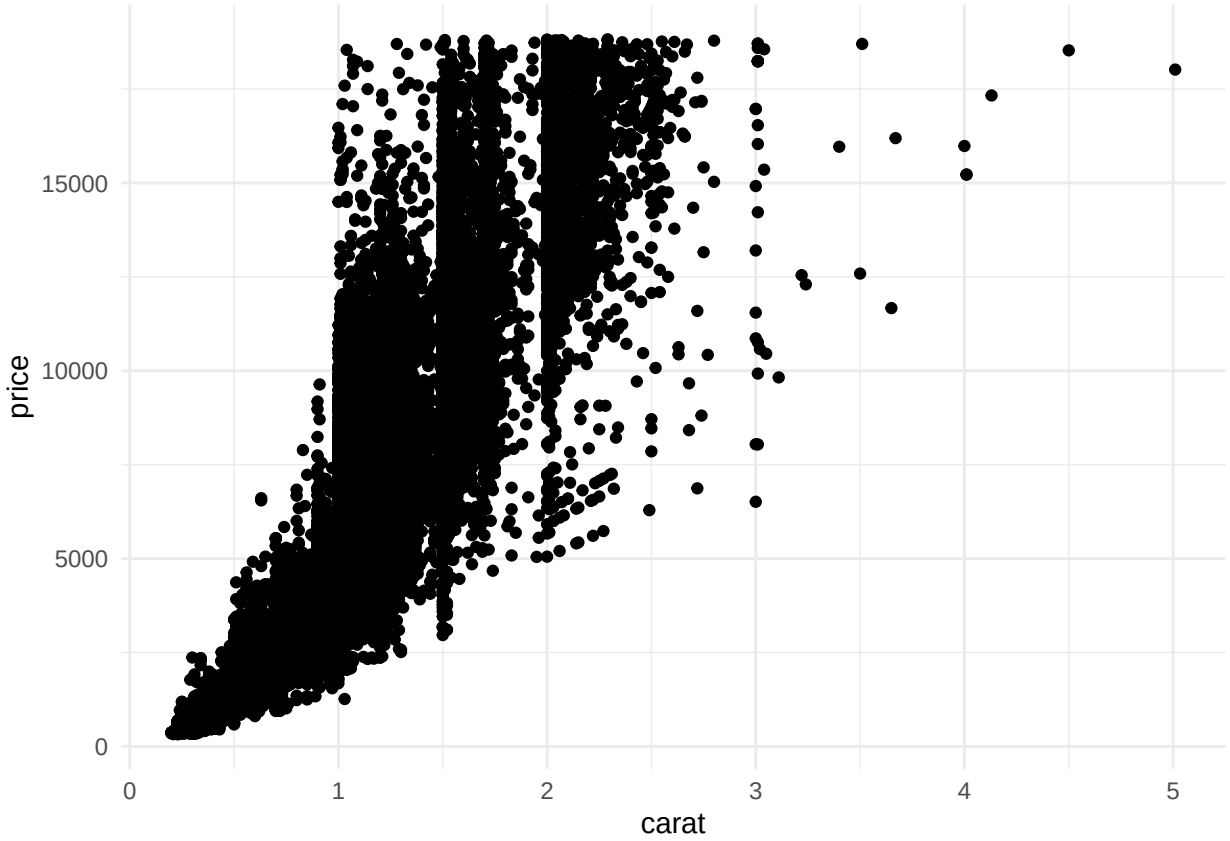
```
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price)) + theme_bw()
```



This makes the plot background black & white instead of grey, to accomplish this it also adds a black boarder so the plot can still be displayed.

d. **Try out several of the other theme options (https://ggplot2.tidyverse.org/reference/ggtheme.html). Pick your favorite, display it here, and describe what it does and what you like about it.**

```
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price)) + theme_minimal()
```



This creates a graph with a without a background, just the grid lines and labels.

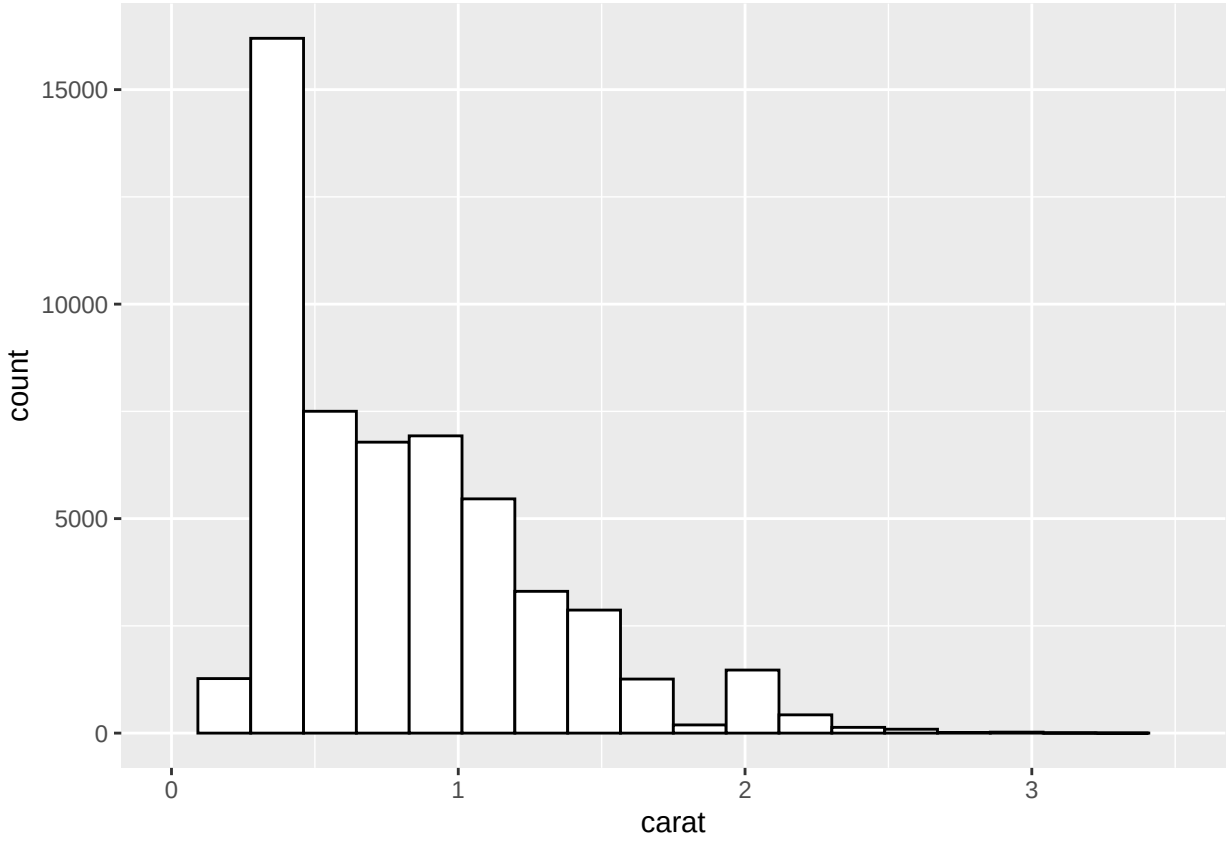
Question 8 (8 points)

a. **(5 points) Make a histogram showing the spread of the variable carat from the data set diamonds. Restrict your histogram to only range from 0 to 3.5 on the x-axis, pick an appropriate bin width, and explain why you picked the binwidth you did.**

```
ggplot(data = diamonds) + geom_histogram(aes(x = carat), color = "black", fill = "white", bins = 20) +
  xlim(c(0, 3.5))
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

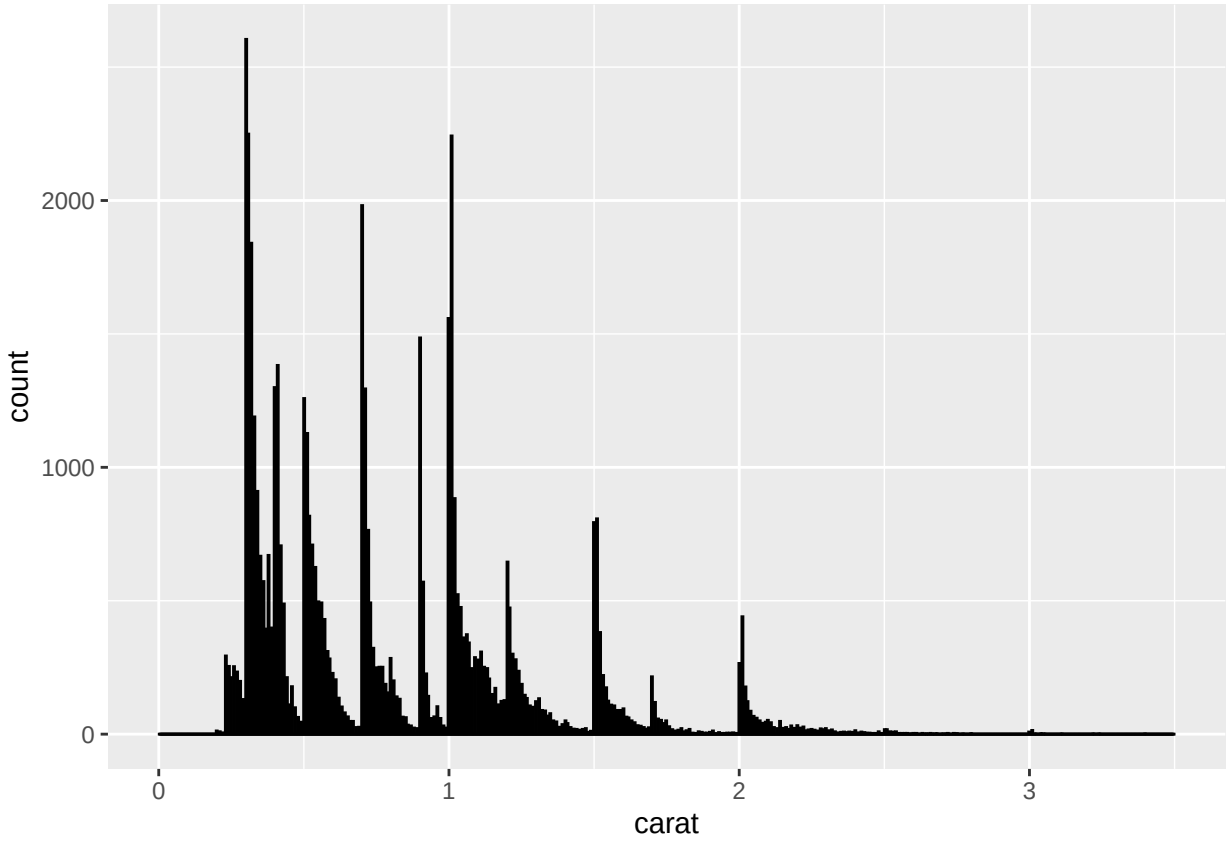


b. **(3 points) Change the binwidth of your histogram in part a to a value that is too big. Explain why what you've produced isn't a good histogram.**

```
ggplot(data = diamonds) + geom_histogram(aes(x = carat), color = "black", fill = "white", bins = 1000) +
  xlim(c(0, 3.5))
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

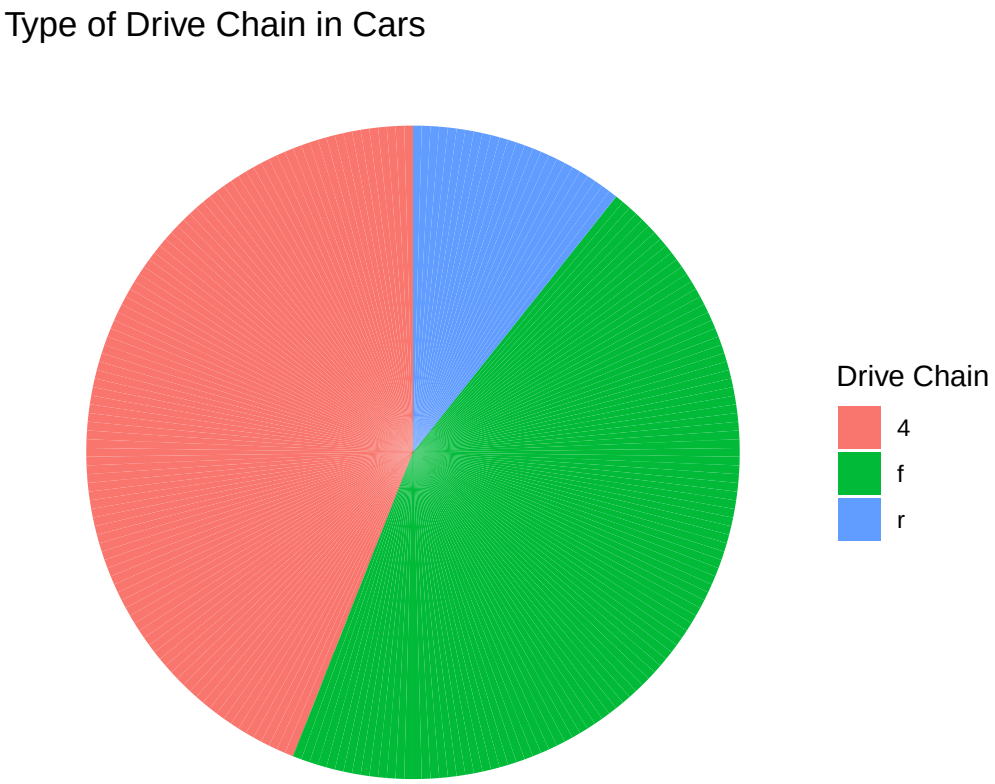


This histogram

Question 9 (8 points)

Make a pie chart showing the type of drive chain among the cars in the mpg data set. Make your chart by doing a polar coordinate transformation to a bar chart with a single bar. Be sure to set the theme of the plot appropriately, add a title, add information about where the data came from, and change the legend title to use complete words rather than a variable name (Hint: the legend title is the label for the fill).

```
ggplot(data = mpg) + geom_bar(mapping = aes(x = "", y = "Drive Chain", fill = drv), stat = "identity") +
  coord_polar("y") +
  labs(title = "Type of Drive Chain in Cars", fill = "Drive Chain") +
  theme_void()
```



Question 10 (10 points)

a. **(1 point) To make a marginal plot using ggMarginal, what library do you need to load?**

library("ggExtra")

b. **(3 points) In class, we made a Marginal plot with type = "histogram". What are the five different types of marginal plots you can make with ggMarginal?**

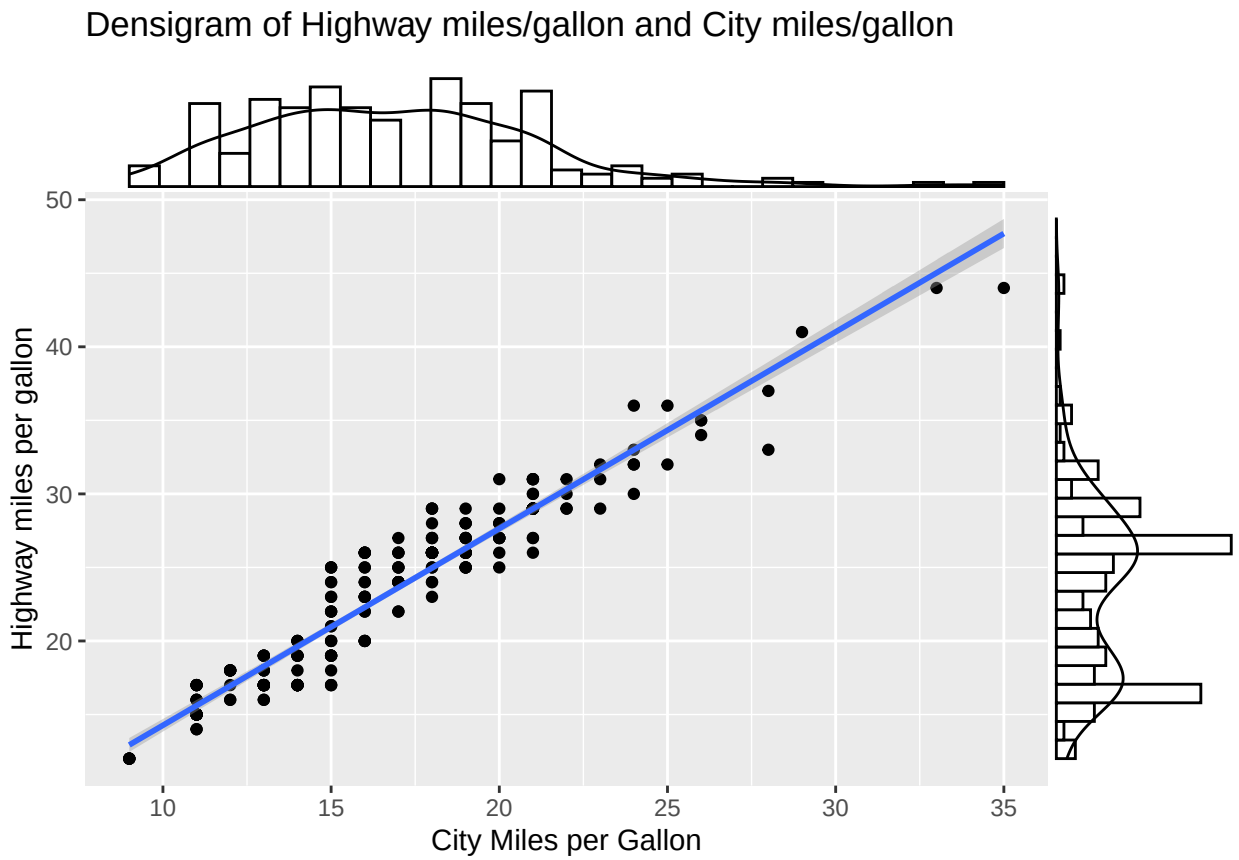
Density, Histogram, Boxplot, Violin, and Denisgram

c. **(6 points) Try out one of these marginal plots. Make your plot look professional and easy to read: Give it a title, label the axes with words not variables names, add source information, put the legend in a convenient spot, etc.**

```
library("ggExtra")
g <- ggplot(data = mpg, aes(x=cty, y = hwy)) +
  geom_point() +
  labs(title = "Densigram of Highway miles/gallon and City miles/gallon", y = "Highway miles per gallon
  geom_smooth(method = "lm")

ggMarginal(g, type = "densigram", fill = "white")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```


Densigram of Highway miles/gallon and City miles/gallon
Highway miles per gallon
50
40
30
20
10
15
20
25
30
35
City Miles per Gallon

Question 11 (12 points)

In this question you'll make a correlogram for the diamonds data set. However, you can only calculate correlations of numeric variables, not categorical variables, so first we'll make a new data set, diamonds_numeric, that only has the numeric variables of the diamonds data set and leaves out the categorical ones:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
diamonds_numeric = select(diamonds, price, carat, x, y, z, depth, table)
head(diamonds_numeric)
```

```
## # A tibble: 6 x 7
##   price carat     x     y     z depth table
##   <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   326  0.23  3.95  3.98  2.43  61.5    55
## 2   326  0.21  3.89  3.84  2.31  59.8    61
## 3   327  0.23  4.05  4.07  2.31  56.9    65
## 4   334  0.29  4.2   4.23  2.63  62.4    58
## 5   335  0.31  4.34  4.35  2.75  63.3    58
## 6   336  0.24  3.94  3.96  2.48  62.8    57
```

a. **Calculate the pairwise correlations of these 7 numeric variables in diamonds_numeric. The result should be a 7x7 table.**

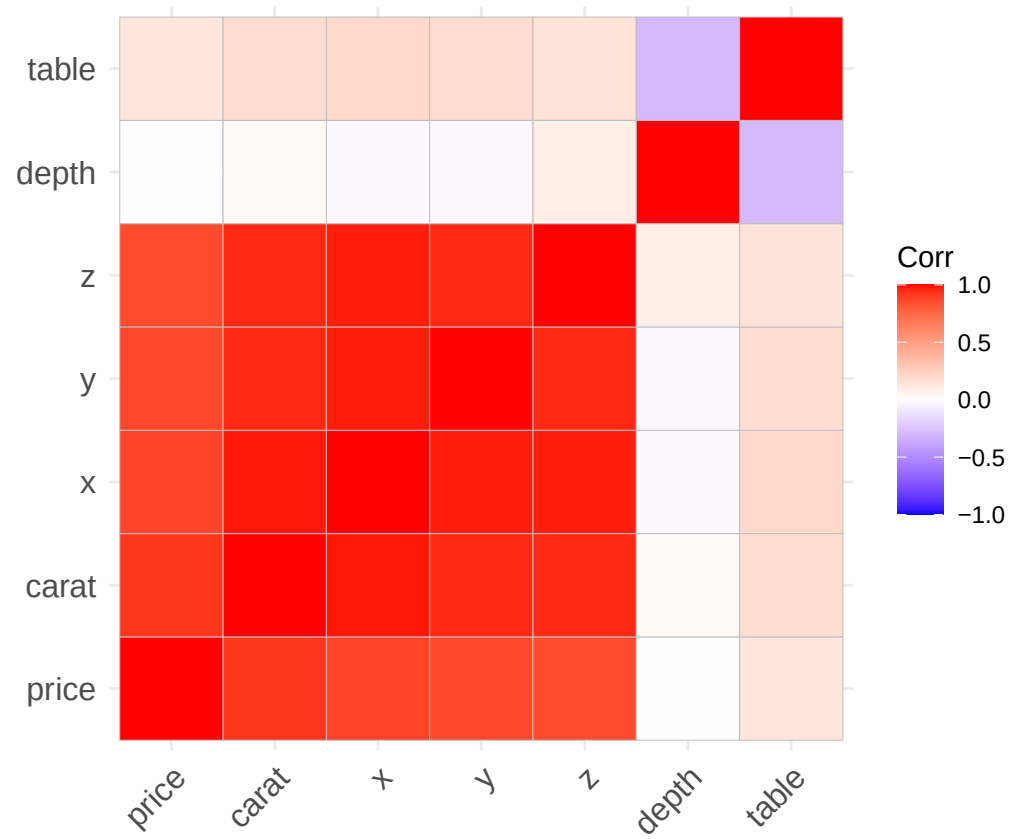
```
diamonds_cor <- cor(diamonds_numeric)
diamonds_cor
```

```
##            price      carat           x           y          z       depth
## price  1.0000000 0.92159130  0.88443516  0.86542090 0.86124944 -0.01064740
## carat  0.9215913 1.00000000  0.97509423  0.95172220 0.95338738  0.02822431
## x      0.8844352 0.97509423  1.00000000  0.97470148 0.97077180 -0.02528925
## y      0.8654209 0.95172220  0.97470148  1.00000000 0.95200572 -0.02934067
## z      0.8612494 0.95338738  0.97077180  0.95200572 1.00000000  0.09492388
## depth -0.0106474 0.02822431 -0.02528925 -0.02934067 0.09492388  1.00000000
## table  0.1271339 0.18161755  0.19534428  0.18376015 0.15092869 -0.29577852
##            table
## price  0.1271339
## carat  0.1816175
## x      0.1953443
```

```
## y      0.1837601
## z      0.1509287
## depth -0.2957785
## table  1.0000000
```

b. **Make a correlogram using this 7x7 table.**

```
library(ggcorrplot)
ggcorrplot(diamonds_cor)
```



c. **Referencing your correlogram, suppose you know that a particular diamond in your data set has a high price. What does that suggest about the other 6 variables for that diamond?**

It will have a positive correlation to price, carat, x, y, and z. So when price increases so does carat, x, y, and, z but depth won't change and table has a small correlation increasing slightly.

d. **Referencing your correlogram, suppose you know that a particular diamond in your data set has a large depth. What does that suggest about the other 6 variables for that diamond?**

It will have no connection to price, carat, x, y, or z but it has a negative correlation to table, so table will be lower.