Homework 2

[Ra-Zakee Muhammad]

Due 9/14/2021

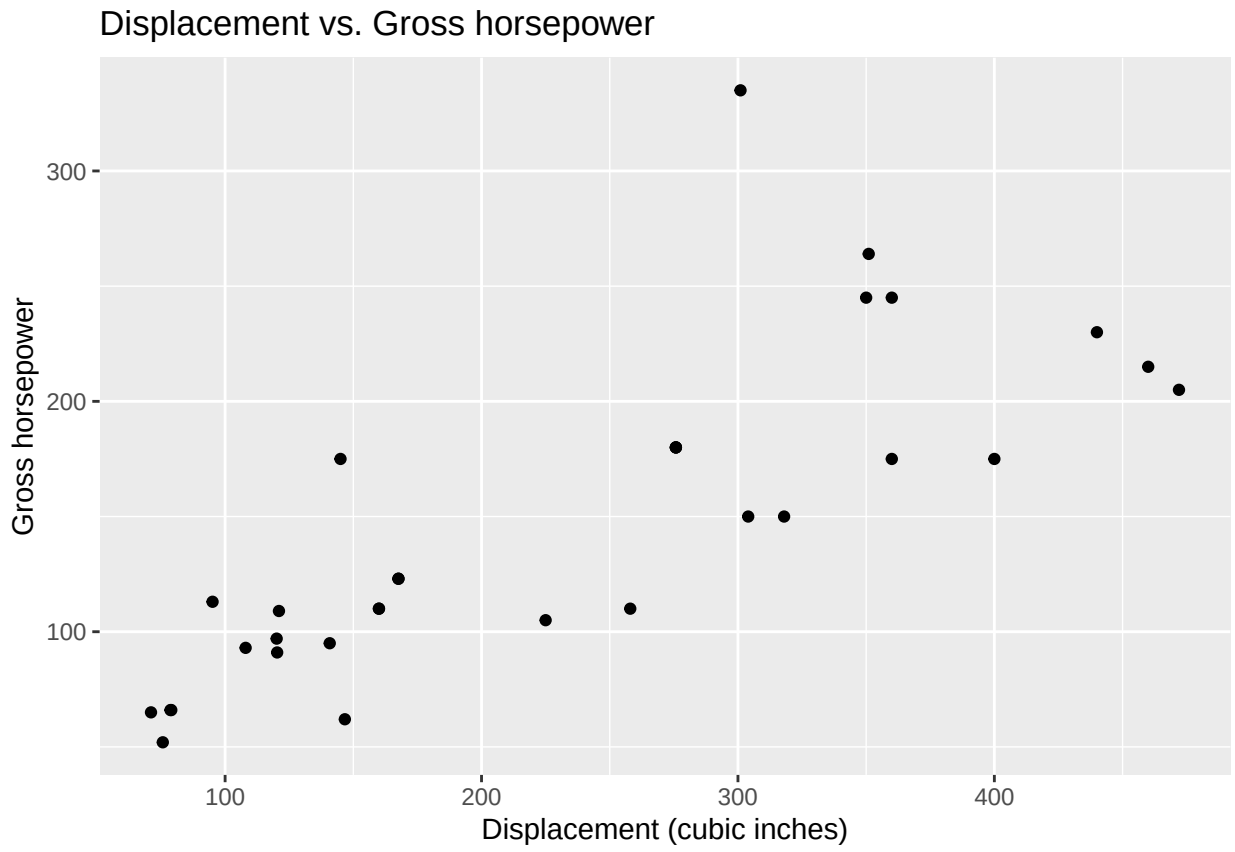
Classmates/other resources consulted: [type answer here]

```
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, mapping = aes(x = disp , y = hp )) +
  geom_point() +
  labs(title = "Displacement vs. Gross horsepower") +
  xlab("Displacement (cubic inches)") +
  ylab("Gross horsepower")
```



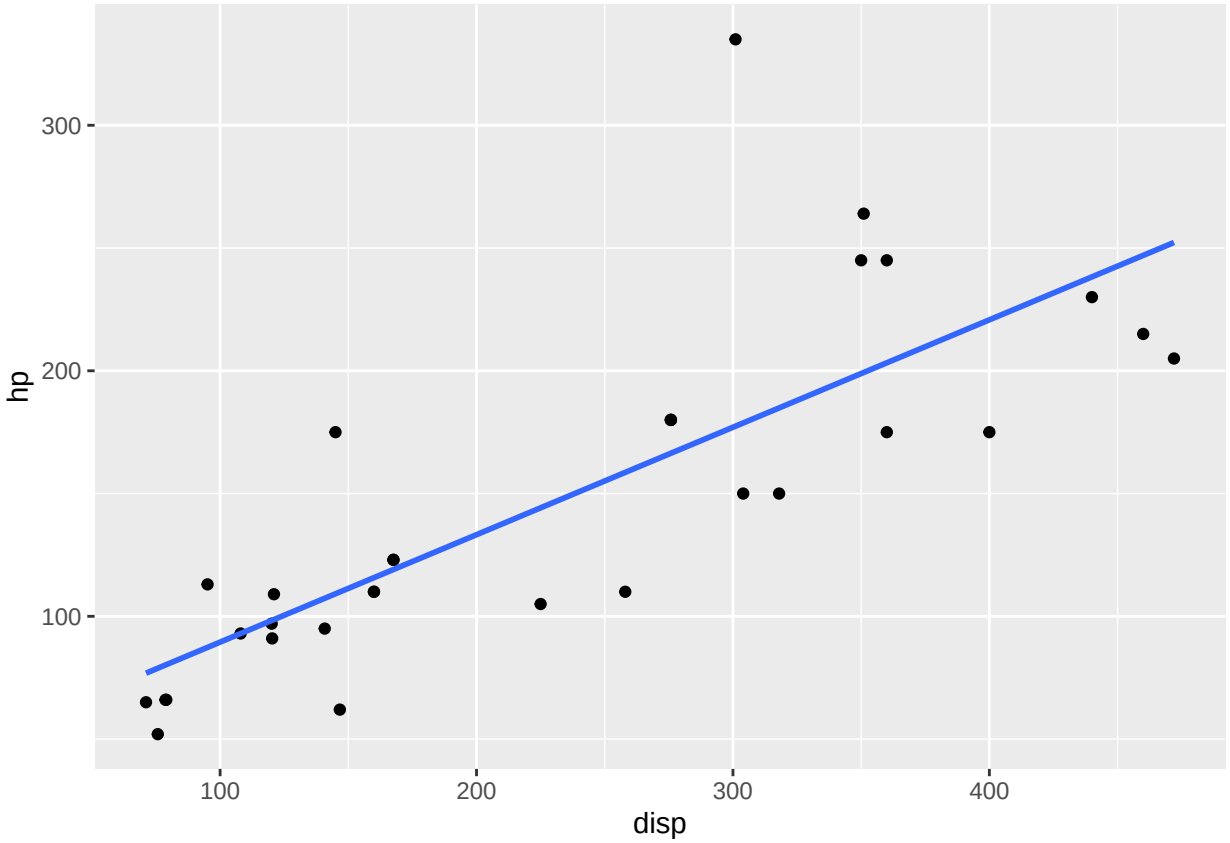
b. **Are there any data points in this data set that fall directly on top of each other? Explain how you know.**

Yes there are because when you run the command "?mtcars" you are told that there are ll observations but in the graph that we just made, there are only 27 points visible.

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, mapping = aes(x = disp , y = hp )) +
  geom_point() +
  geom_smooth(method = lm , se = FALSE)
```

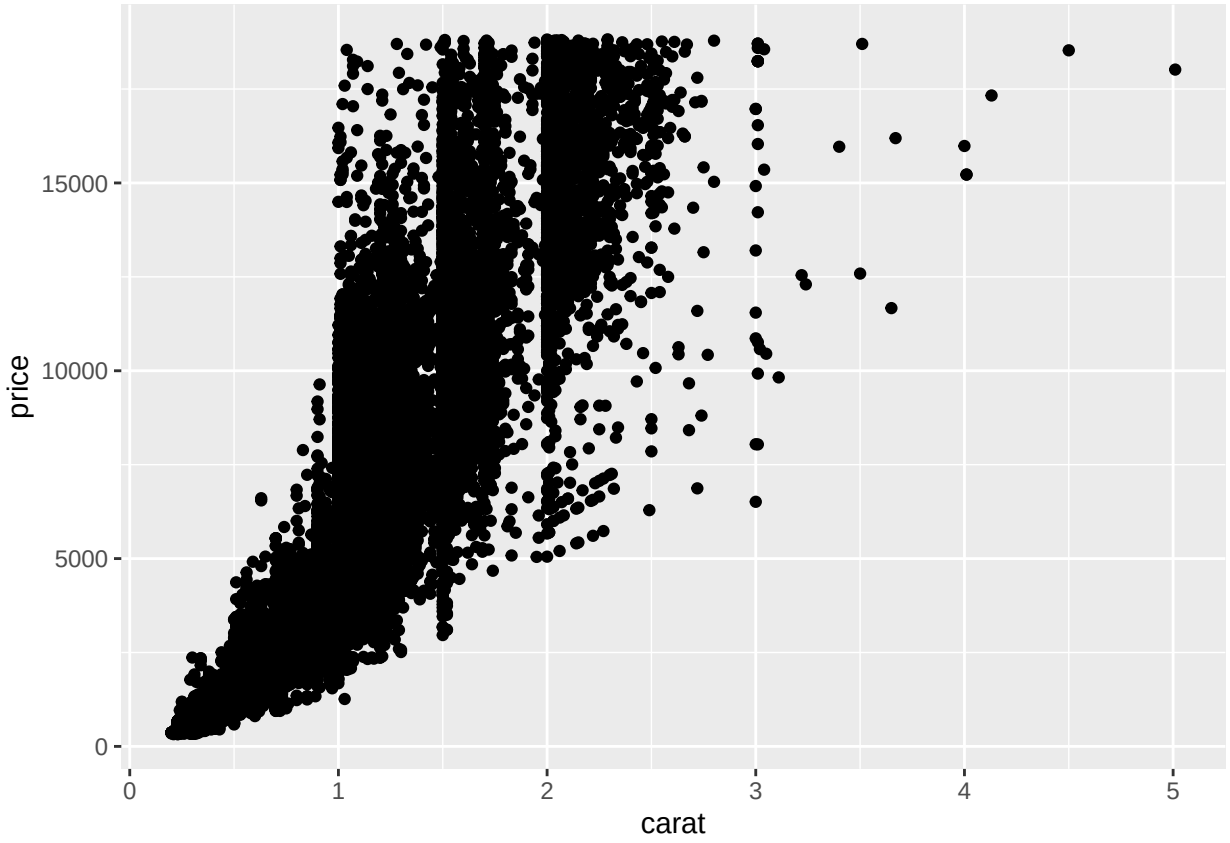
```
## `geom_smooth()` using formula 'y ~ x'
```



Question 2 (12 points)

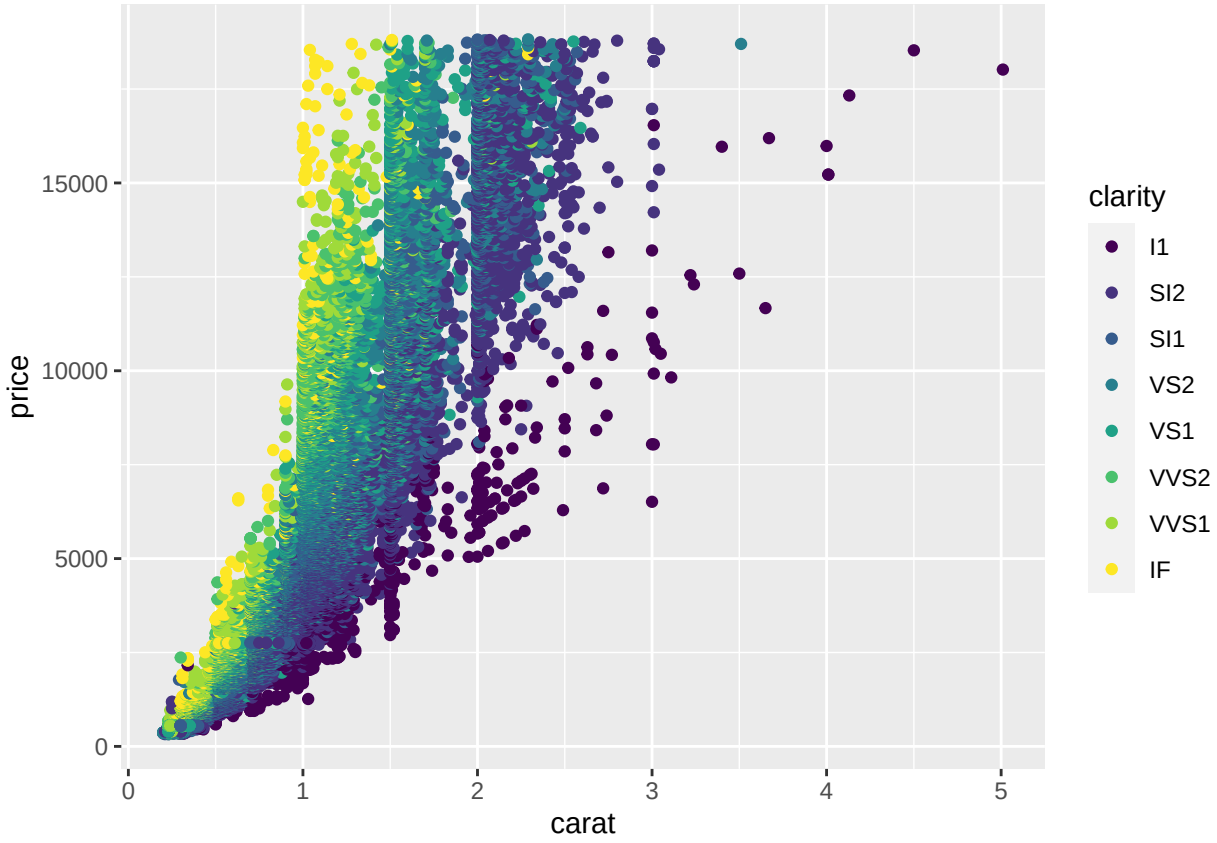
Consider the following scatterplot of price vs. carat of diamonds

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



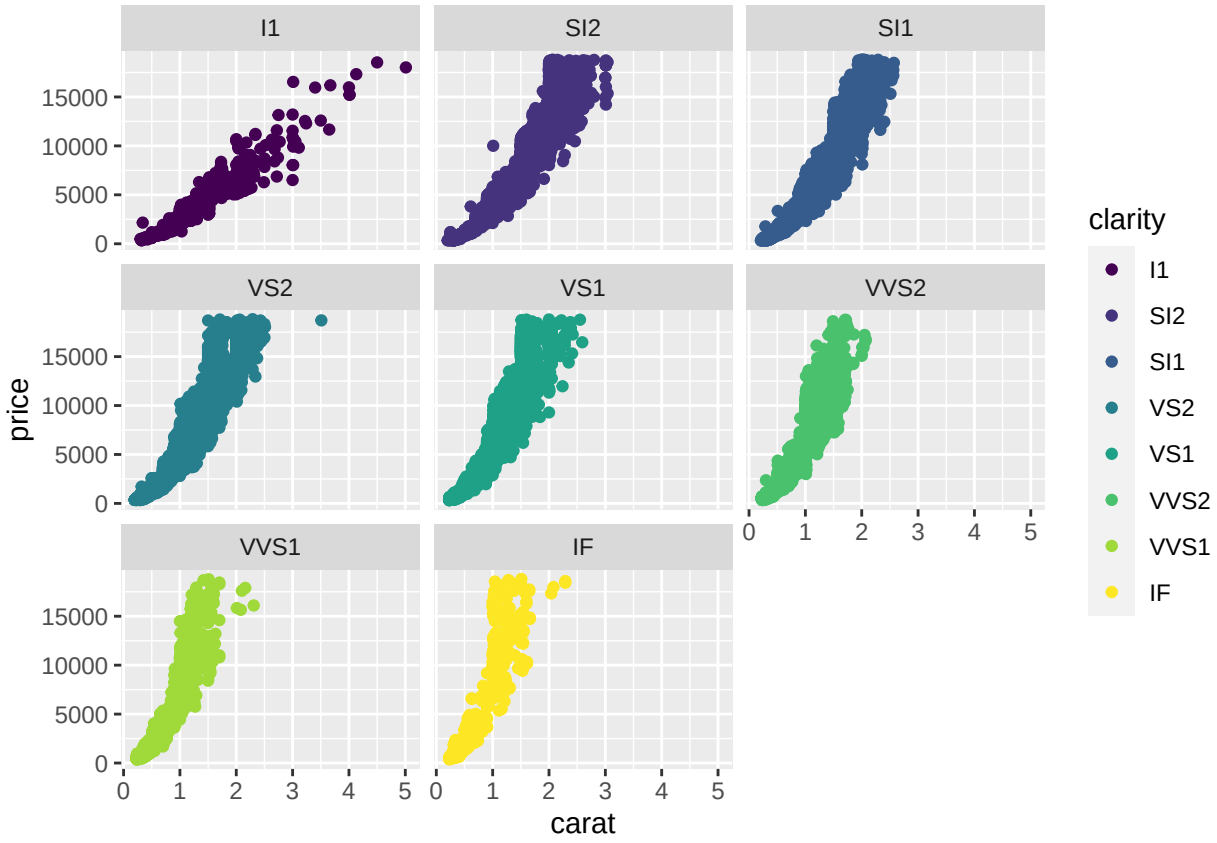
a. **Distinguish your data points based on the clarity of the diamonds, using whatever method you'd like (that is, the data points corresponding to one clarity level should look different than the data points corresponding to another clarity level, etc.)**

```
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, color = clarity)) +
  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?**

a

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

b

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

a

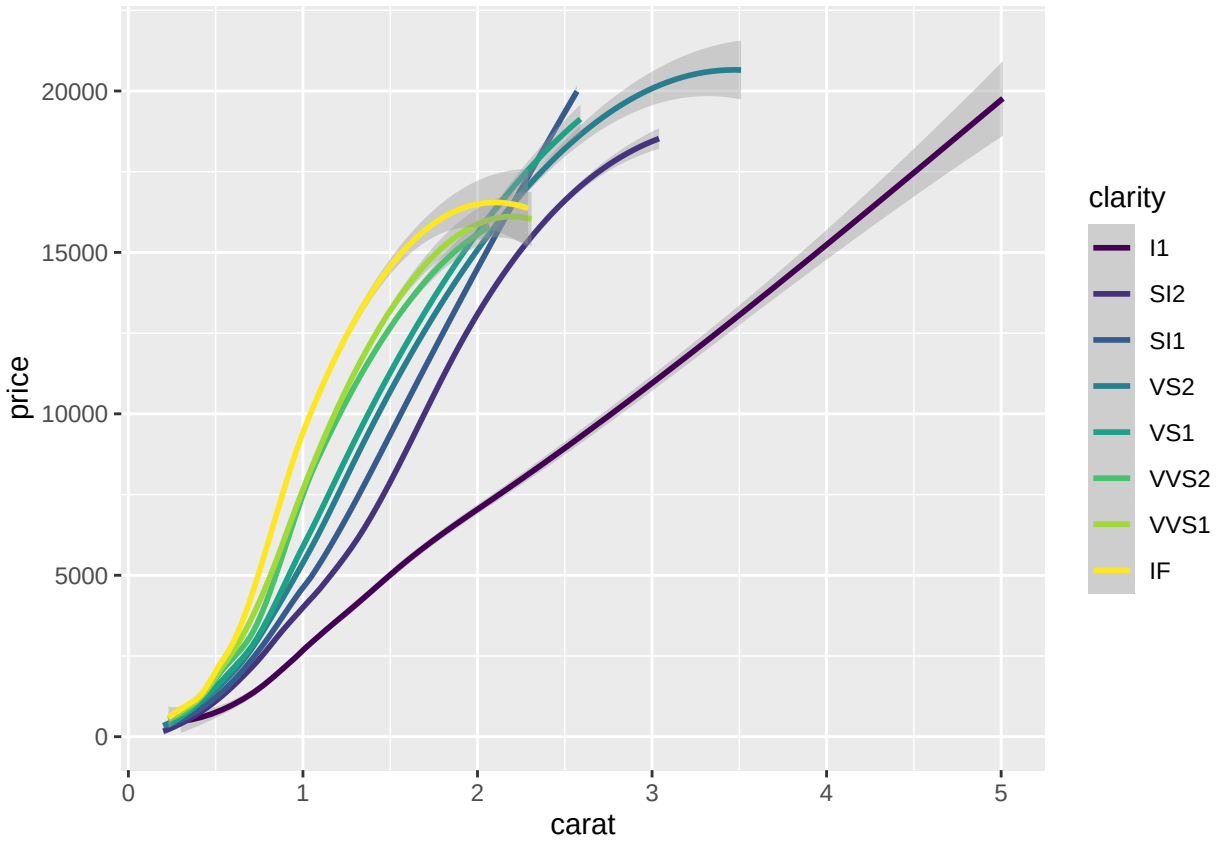
iv. **For what clarity level is the relationship between carats and price strongest?**

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(method = "loess")
```

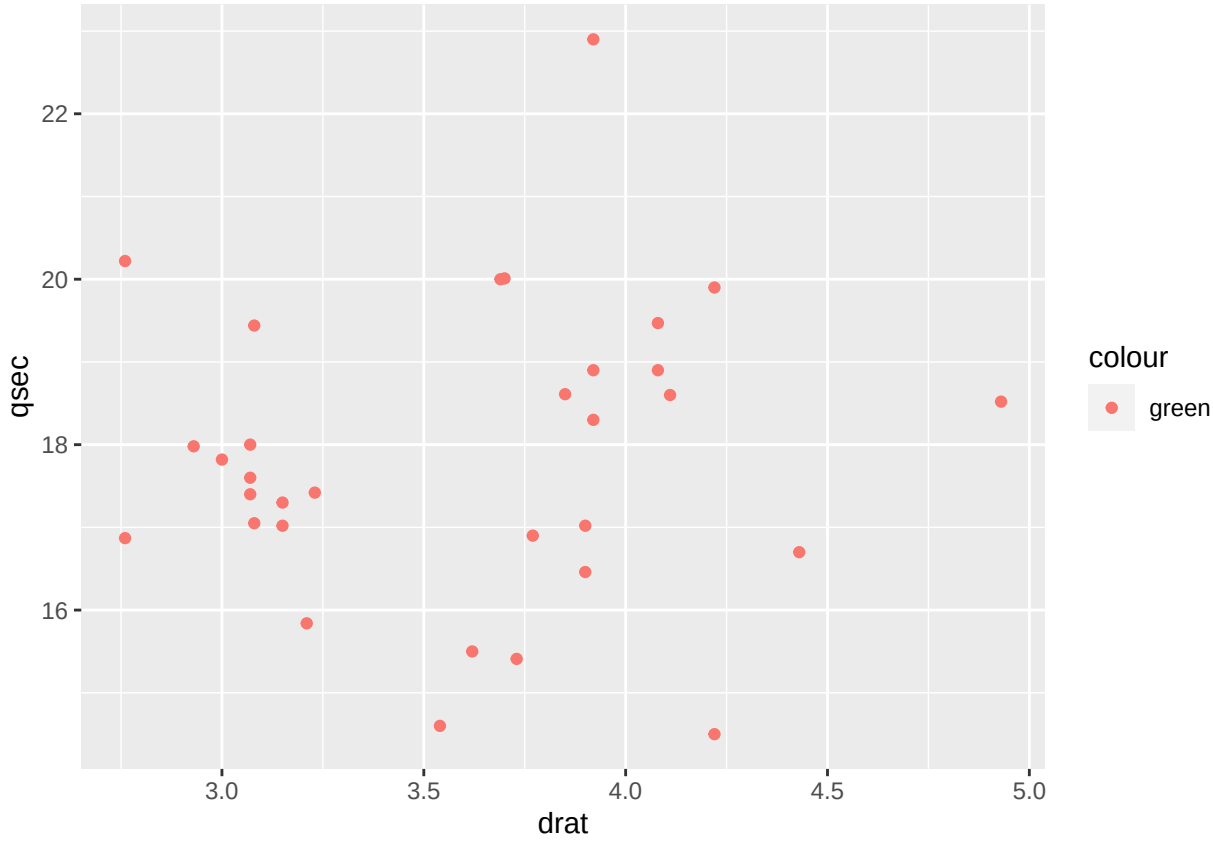
```
## `geom_smooth()` using formula 'y ~ x'
```



Question 3 (4 points)

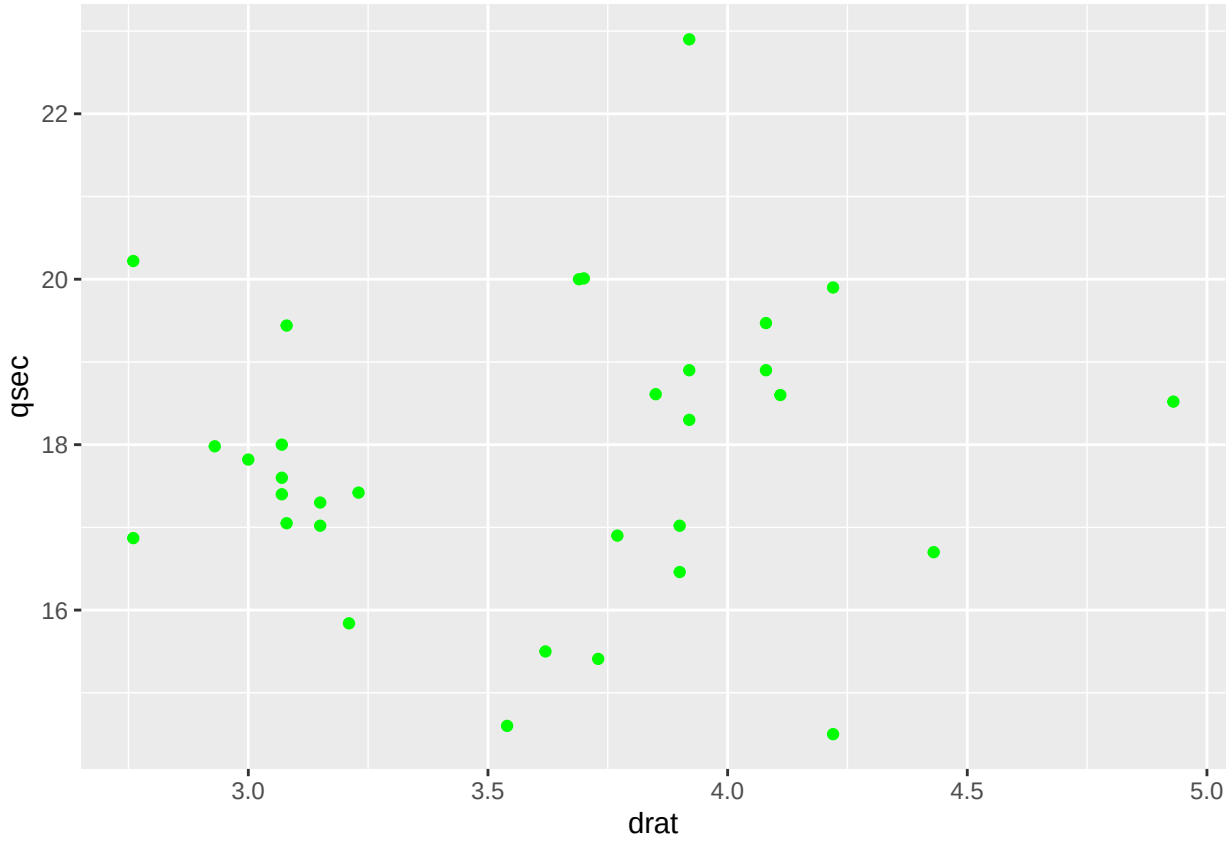
What has gone wrong with this plot, and how would you change it to make all points green?

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



Here is the fixed code:

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



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))

yes because the first parameter of the ggplot function is data and for geom_point it is mapping so you need only put into the argument what you want data and mapping to be equal to resepectivly.

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 it will be the same because ggplot can be used to to specify all plot aesthetics just as geom_point can be used to specify plot aesthetics. All arguments are set to the same things among these two variations.

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 it is the same because none of the points are completely over lapping and that would be the differentiating factor between geom_point and geom_count. Additionally, all aesthetic arguments and parameters are set to the same values therefor, they are the same graph.

Question 5 (10 points)

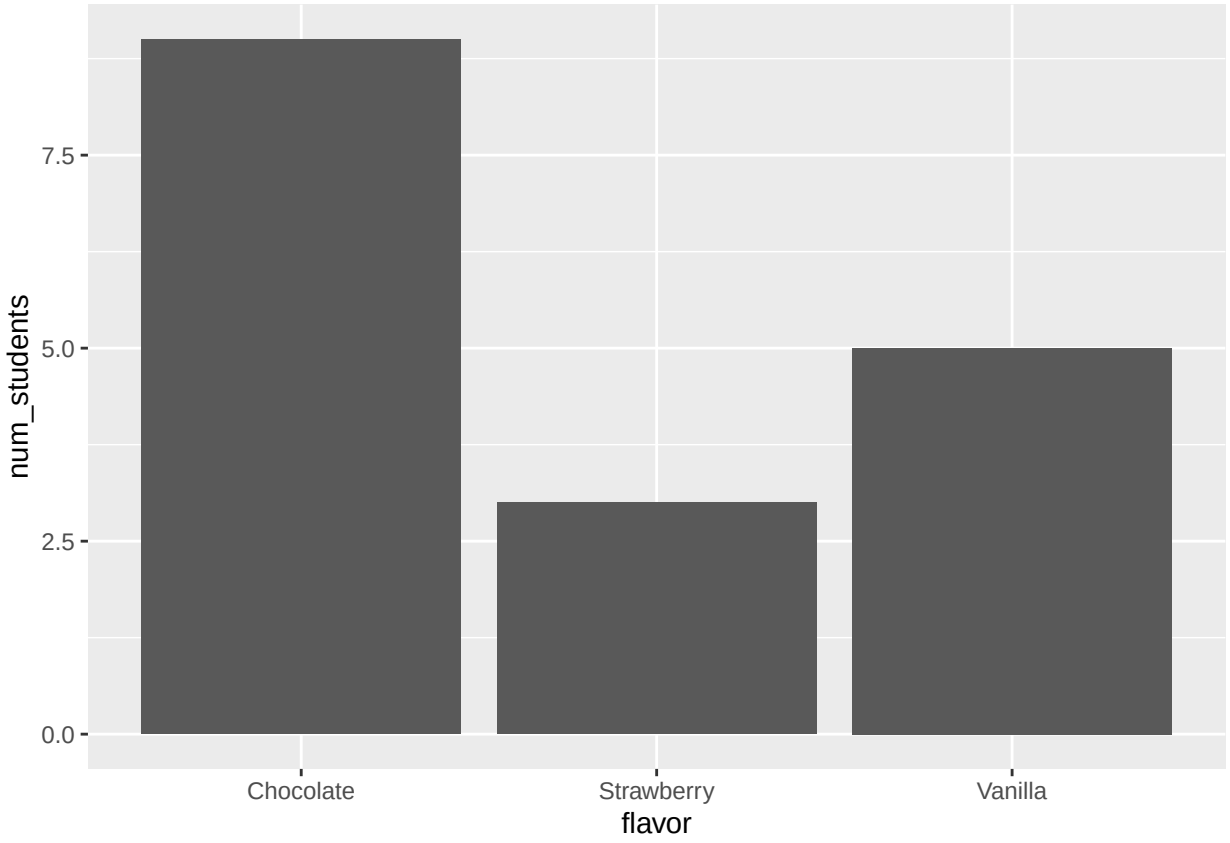
Suppose you have data about students' preferences for ice cream represented in a table, and you want to make a bar chart.

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(mapping = aes(x = flavor, y = num_students), stat = "identity")
```

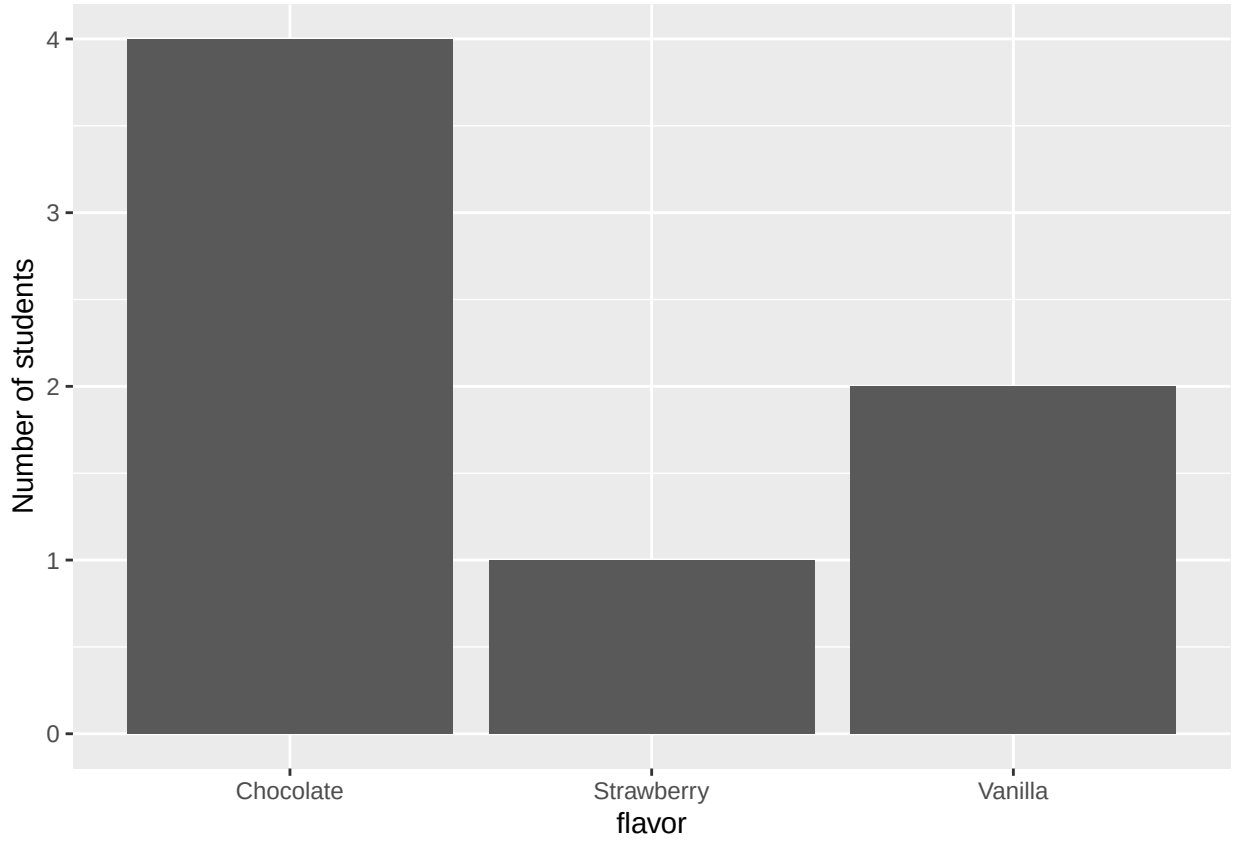


b. **Now, suppose your table has a row for each student. The first column, named 'student', has each student's name, and the second column, named 'flavor', has that student's favorite ice cream flavor. How would you make a bar chart that shows how many students prefer each type of ice cream?**

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

```
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(data = ice_cream_b) +
  geom_bar(mapping = aes(x = flavor)) +
  ylab("Number of students")
```



Question 6 (4 points)

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

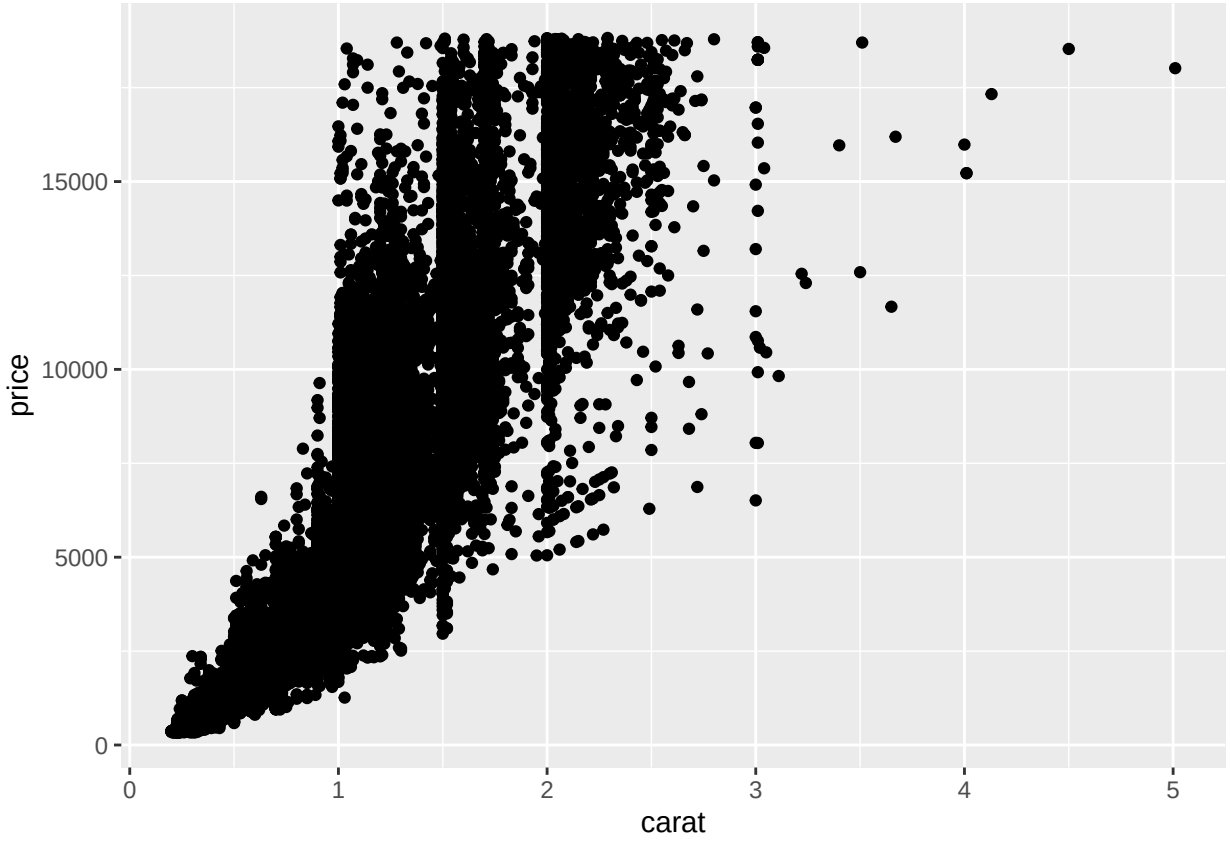
When consulting the help window in RStudio, this is what we are told about geom_col:

"There are two types of bar charts: geom_bar() and geom_col(). geom_bar() makes the height of the bar proportional to the number of cases in each group (or if the weight aesthetic is supplied, the sum of the weights). If you want the heights of the bars to represent values in the data, use geom_col() instead. geom_bar() uses stat_count() by default: it counts the number of cases at each x position. geom_col() uses stat_identity(): it leaves the data as is."

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))
```



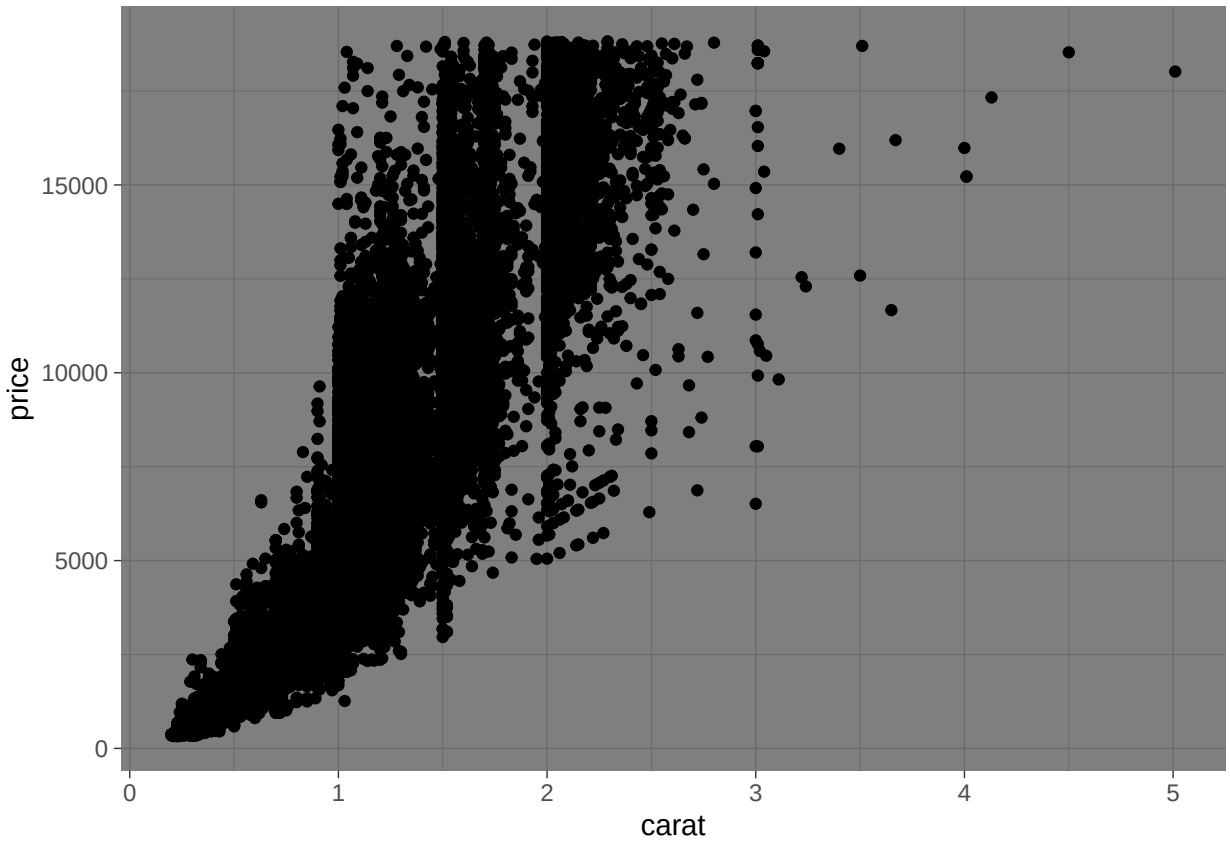
a. **theme_void()**

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

This theme removes the grid and coloring so that all that is left is the data points on a white background.

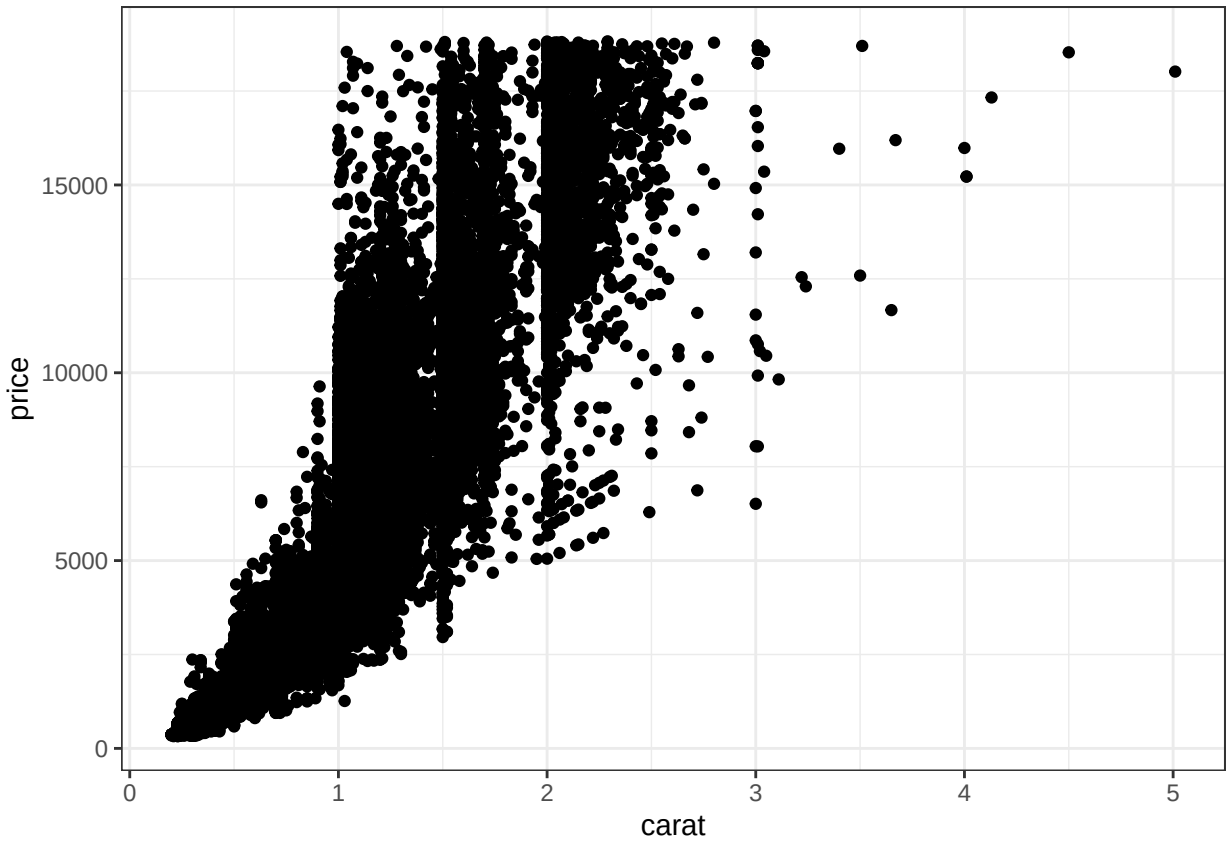
b. **theme_dark()**

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



This makes the color of the background grey (dark). > c. **theme_bw()**

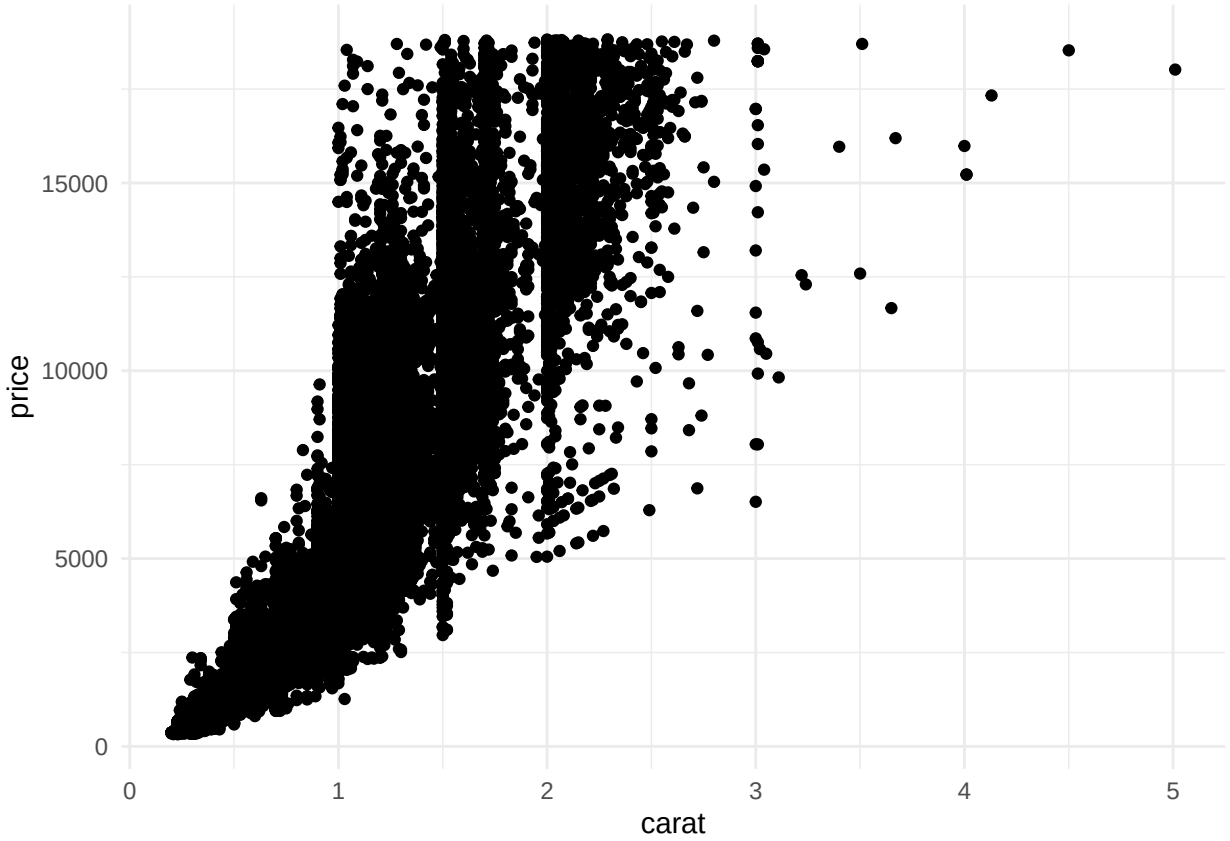
```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price)) +
  theme_bw()
```



This makes the color scheme of the plot black and white.

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 theme, theme_minimal gets rid 0f tick marks on the axis and it gets rid of outer borders so all that is left is labels, axis scales, and data points.

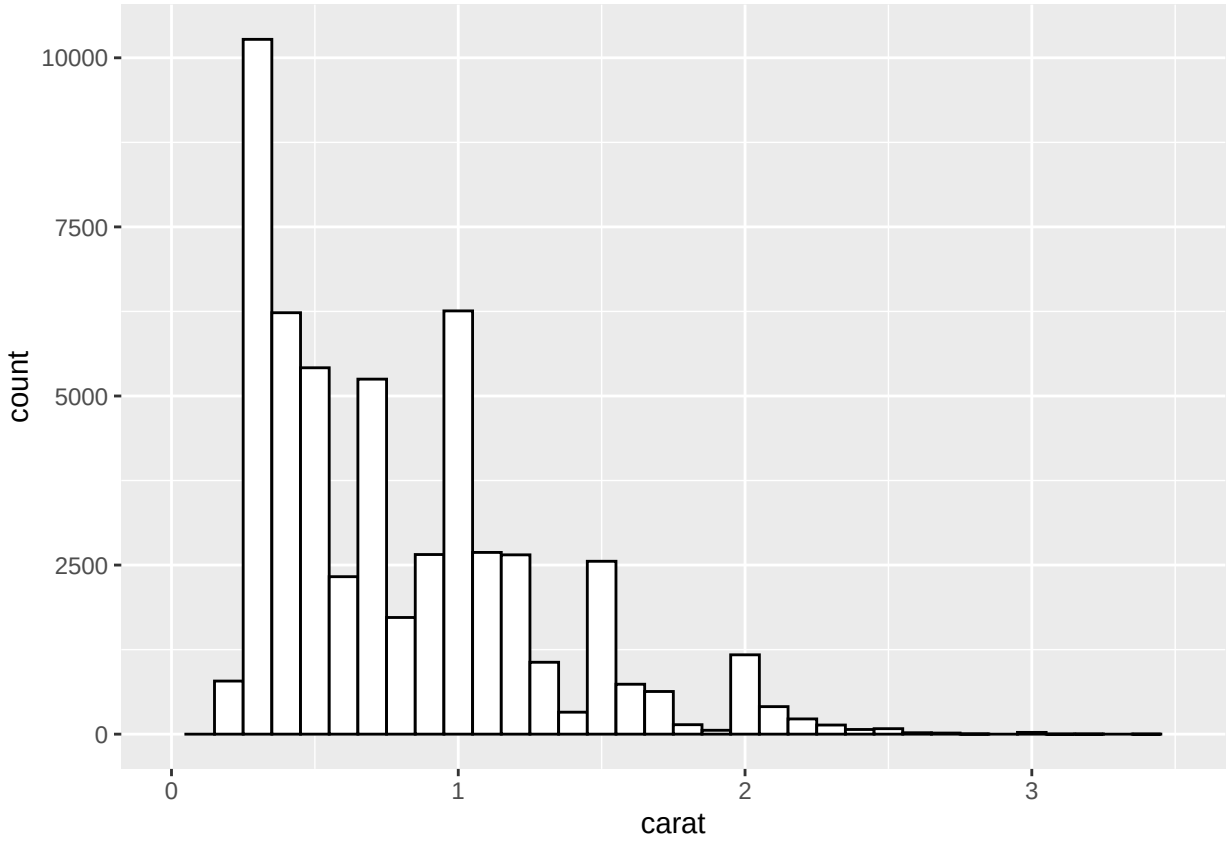
Question 8 (8 points)

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(diamonds, mapping = aes(x = carat)) +
  geom_histogram(binwidth = .1 , color = "black", fill = "white") +
  xlim(0,3.5)
```

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

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



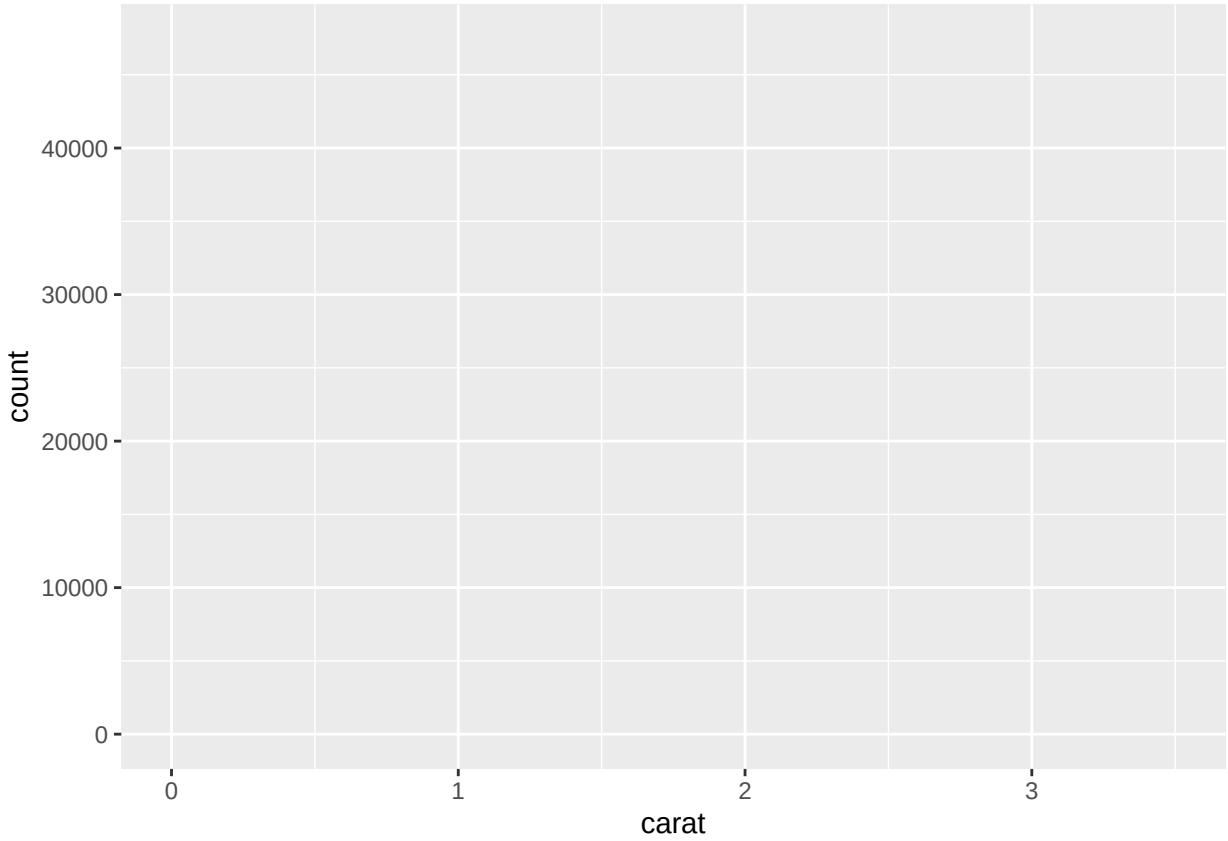
I picked the bin width .1, because when you look at the table of carats and number of observations, you see that carats start at .2 and go up by .01 and so grouping by .1 combines these 10 .01s together so that their width is visible on the histogram but the value ranges are still distinguishable.

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(diamonds, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 2.75 , color = "black", fill = "white") +
  xlim(0,3.5)
```

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

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



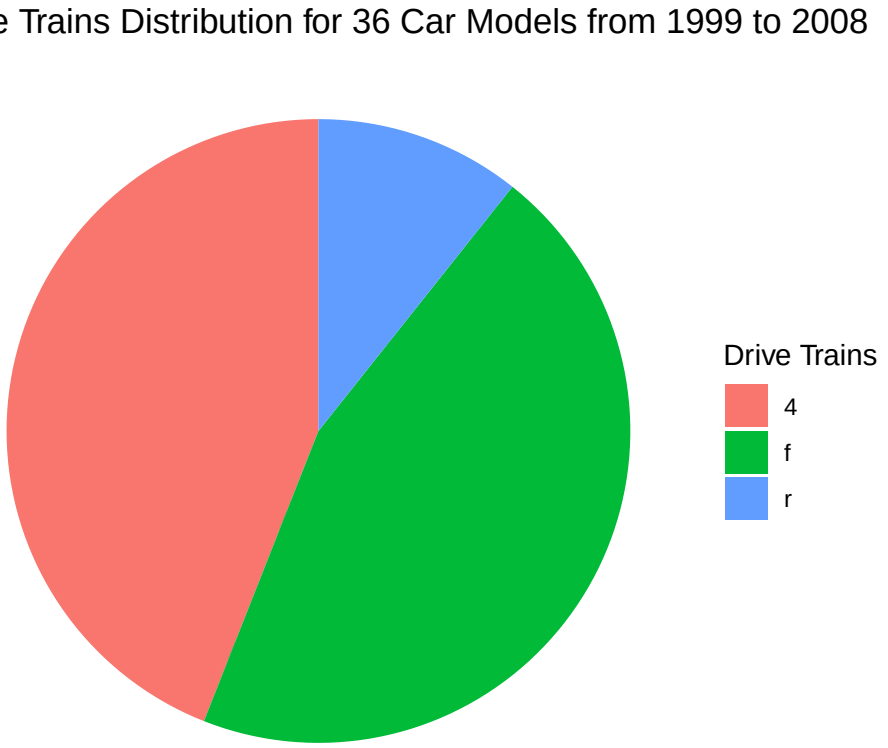
As the bin width increases the left most bin is pushed further and further to the right until the bins to the right of that one are supposed to contain points that are outside of our x axis limits of 0 to 3.5 and there for are not displayed in the histogram at all. If the bin width gets too large (for examples binwidth = 3) then the bin will be centered at 3 with 1.5 on each side but the right side will be out of the 3.5 boundary so the entire bin will not be visible on the histogram. Histograms of bin width too large do not display large portions of the data frame that are still present within the x-boundary making the graph unhelpful.

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).

```
?mpg
```

```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = "", fill = drv) ) +
  coord_polar("y") +
  theme_void()+ labs(title = "Drive Trains Distribution for 36 Car Models from 1999 to 2008", fill = "D
```



ɔn: This dataset contains a subset of the fuel economy data made available by the EPA .

Question 10 (10 points)

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

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, densigram

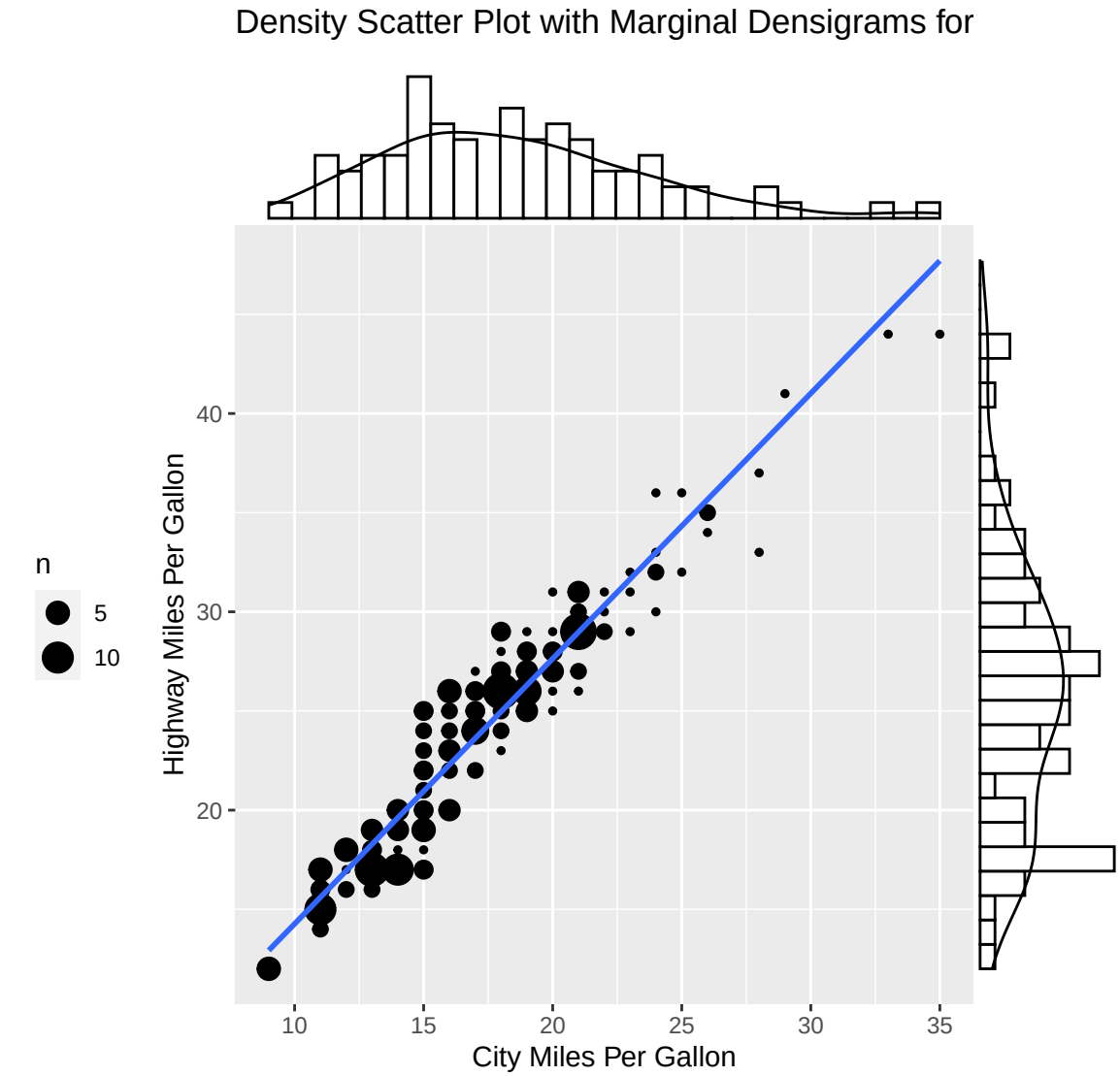
c. **(6 points) Try out one of these marginal plots. Make your plot look professional and easy to read: Give it a title, label the axes with words not variables names, add source information, put the legend in a convenient spot, etc.**

```
library(ggExtra)


g <- ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_count() +
  geom_smooth(method = "lm", se = FALSE) +
```

```
  labs(title = "Density Scatter Plot with Marginal Densigrams for City Miles Per Gallon vs Highway Miles
  theme(legend.position = "left") +
  xlab("City Miles Per Gallon") +
  ylab("Highway Miles Per Gallon")
ggMarginal(g, type = "densigram", fill = "white")
```

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



set of the fuel economy data made available by the EPA for new car models from 1999 to 2008.

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.**

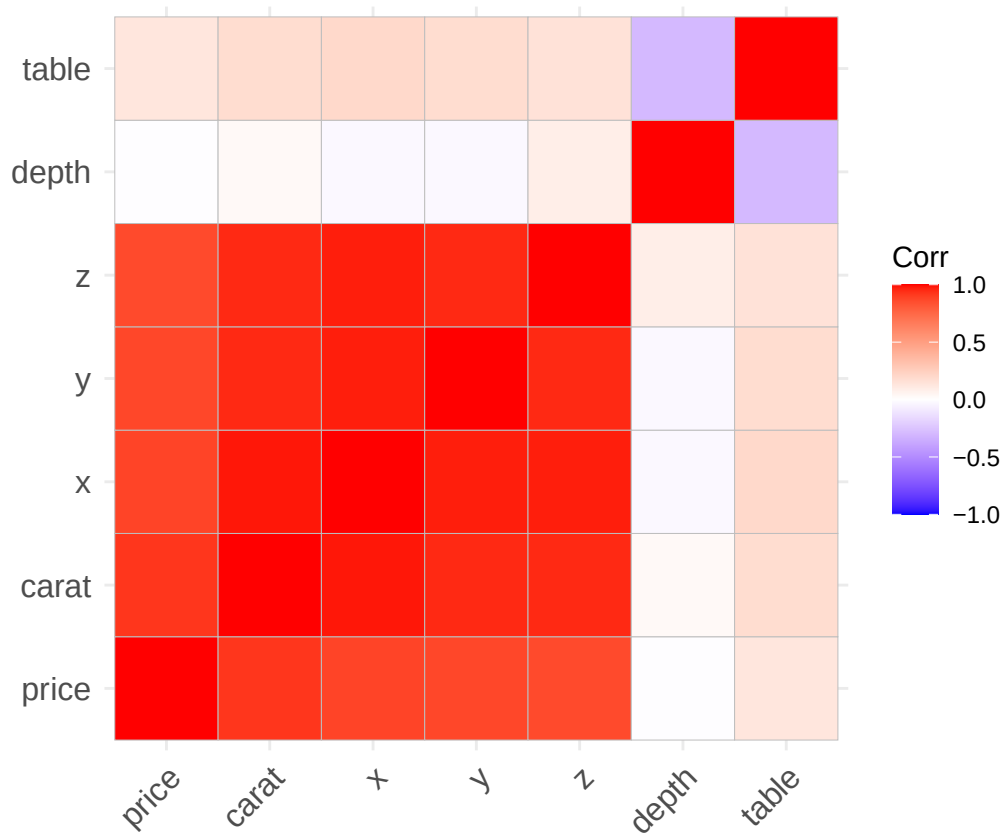
```
hhh <- cor(diamonds_numeric)
hhh
```

```
##            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(hhh)
```



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?**

If a diamond has a high price then it has a high carat, a high x, a high y, and a high z, but a slightly lower than average depth and a slightly higher than average table.

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?**

If a diamond has a high depth then it has an average price, carat, x, y, z, but low table.