EDA

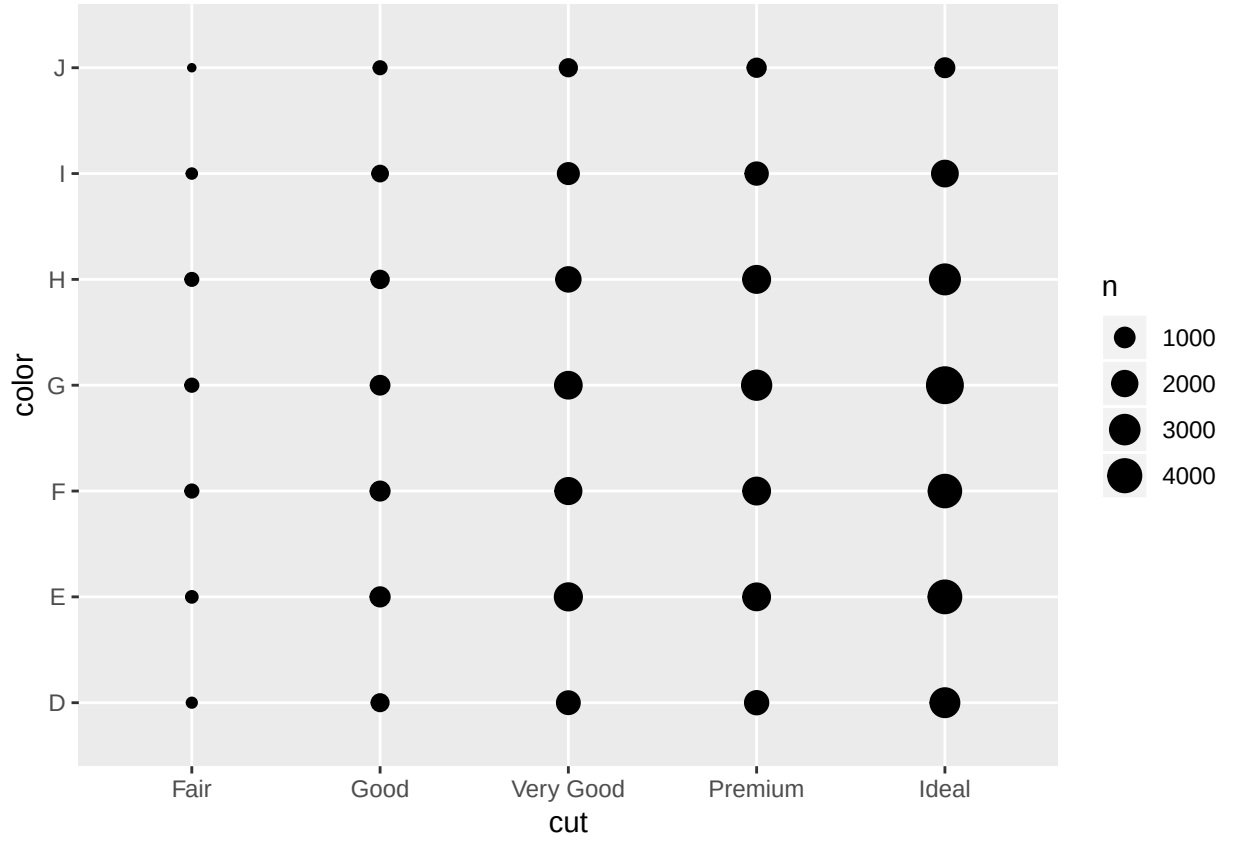
Group 6

5/16/2019

Covariation between two categorical variables (7.5.2) - Martin

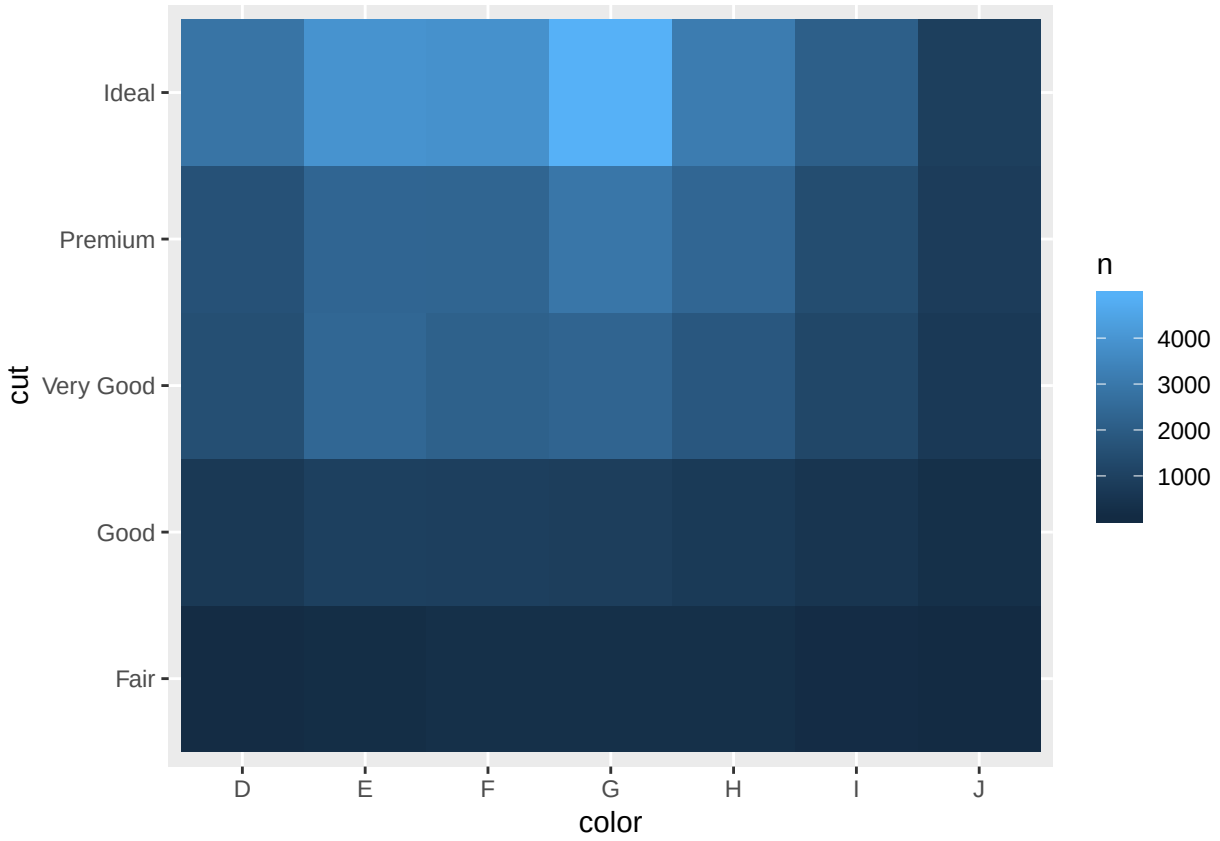
To see how to categorical variables are related we can count the number of observations for each level combination of both variables. We can do this using *geom*$_c$*ount* like in the following example:

```
ggplot(data = diamonds) +
  geom_count(mapping = aes(x = cut, y = color))
```



This graph has the disadvantage that we have to interpret the size of the dots which unless the difference between to neighbors is huge it can be difficult to assess the magnitude. One option, it to use *coun* of ddply and then use the function *geom*$_t$*ile* which make the visualization easier to interpret since use colors to display the covariation between both variables.

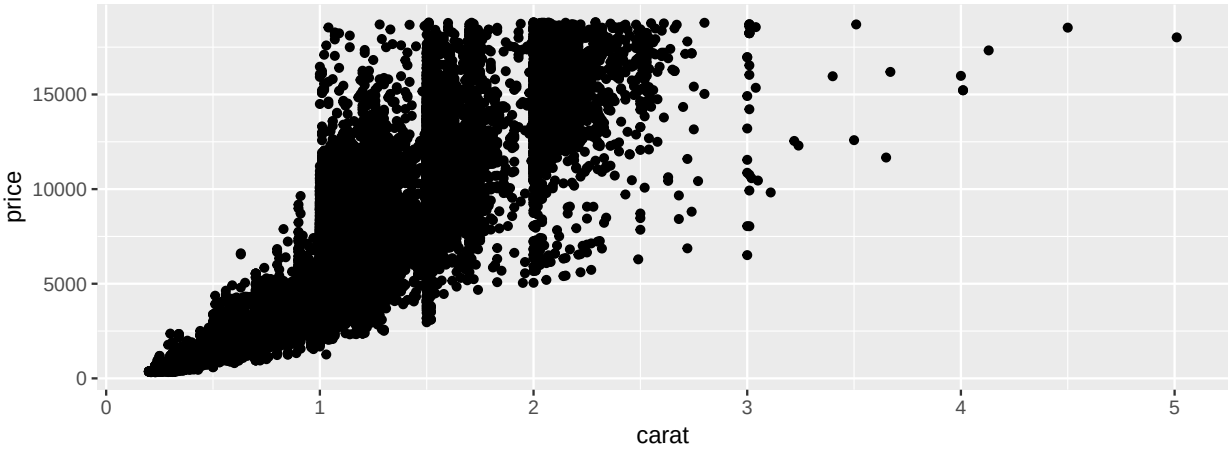
```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
    geom_tile(mapping = aes(fill = n))
```



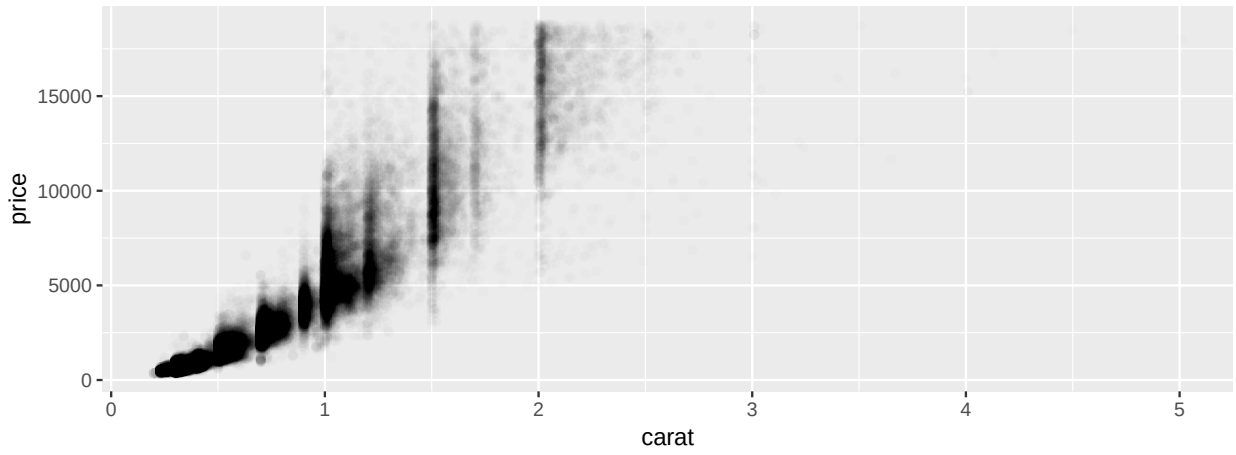
Covariation between two continuous variables (7.5.2) (7.6) - Finn

We've already used scatterplots to examine the coveriance between to continuous variables which tend to get cluttered with a lot of data. In the past we've added transparency using `alpha` to combat this when examining scatterplots woth a lot of data:

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

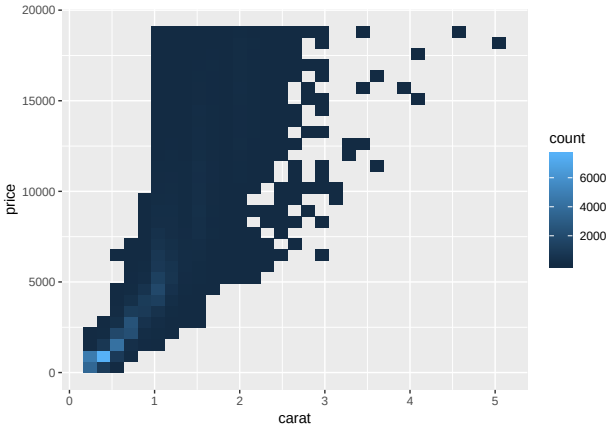


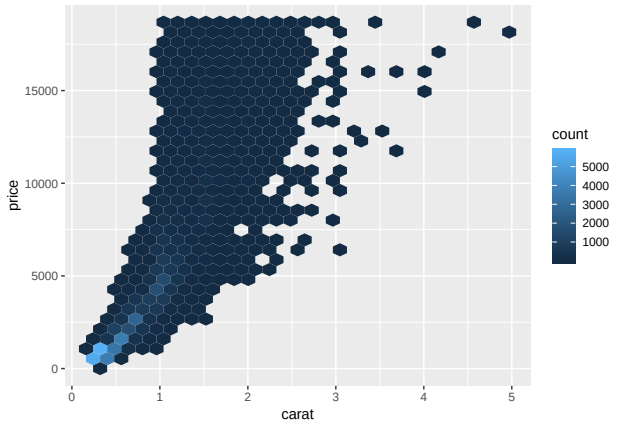
```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price), alpha = 1 / 100)
```



However, this technique of ajusting `alpha` can sometimes be difficult to use. If the data is too large and transparency proved challenging then another possible way of exploring covariance between two continous vriables, which might be useful in this case, is to put them into bins and ajusting the color intead of transparency by using the `geom_bin2d()` and `geom_hex()` functions.

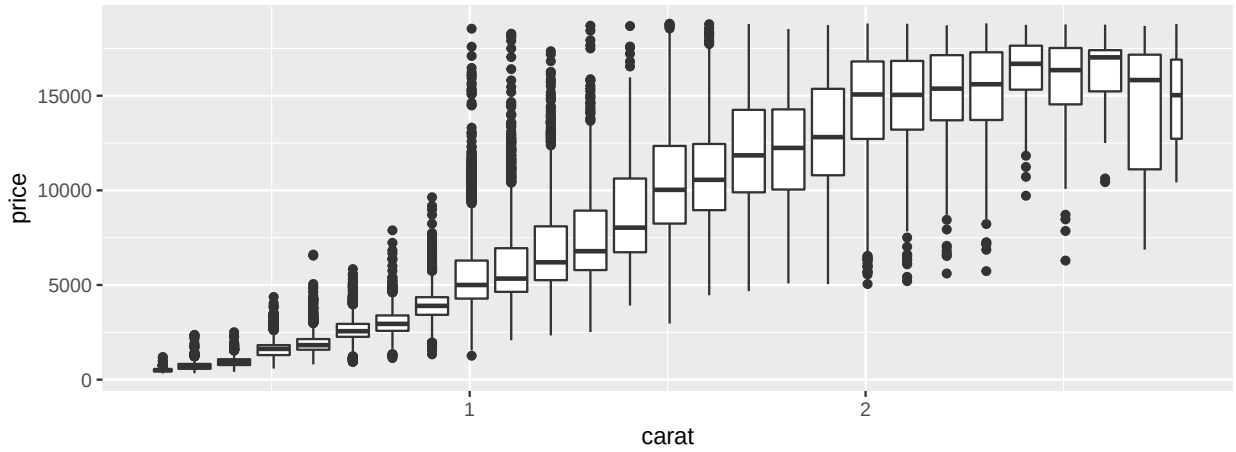
```
smaller <- diamonds %>%
  filter(carat < 3)

ggplot(data = diamonds) +
  geom_bin2d(mapping = aes(x = carat, y = price))

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





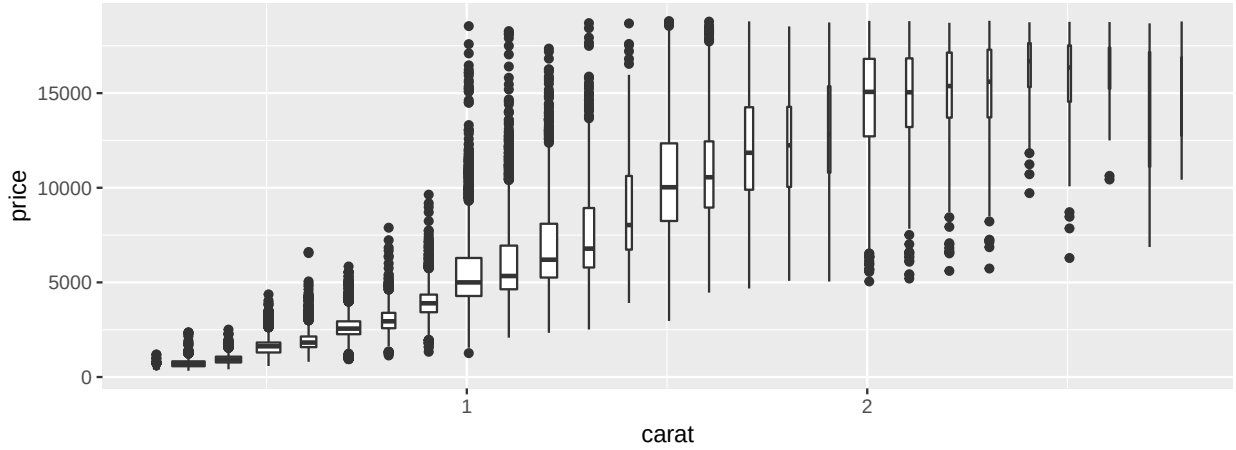
The binning strategy can also be selectivly abblied to just one of the variables then you can plot them as if one were continuous and the other categorical.

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.1)))
```



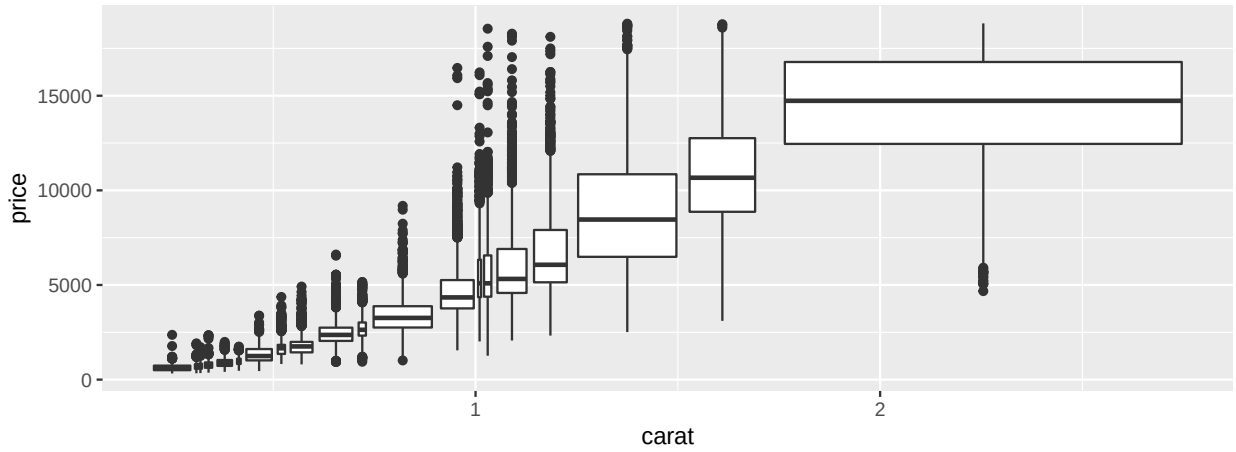
If we want to also have an indicator as to how many ponts each plot is summerizing we can use `varwidth = TRUE`.

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.1)), varwidth = TRUE)
```



This way we can compair the boxes size to see which ones are summerizing smaller or larger sets of datapoints. Since in this case some of the boxes can be hard to see another way of dealing with this problem is to instead divide the variable into equaly sized partions for each bin with the size of the bins instead dictating the range over the axis (sine they now all have roughly the same number of data points) using `cut_number()`.

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_number(carat, 20)))
```


price
15000
10000
5000
0
1
2
carat