HW03p part I

[Your Name Goes Here]

April 13, 2018

```
knitr::opts_chunk$set(error = TRUE) #this allows errors to be printed into the PDF
```

1. Load pacakge `ggplot2` below using `pacman`.

```
library(pacman)
pacman::p_load(ggplot2)
```

The dataset `diamonds` is in the namespace now as it was loaded with the `ggplot2` package. Run the following code and write about the dataset below.

```
?diamonds
str(diamonds)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

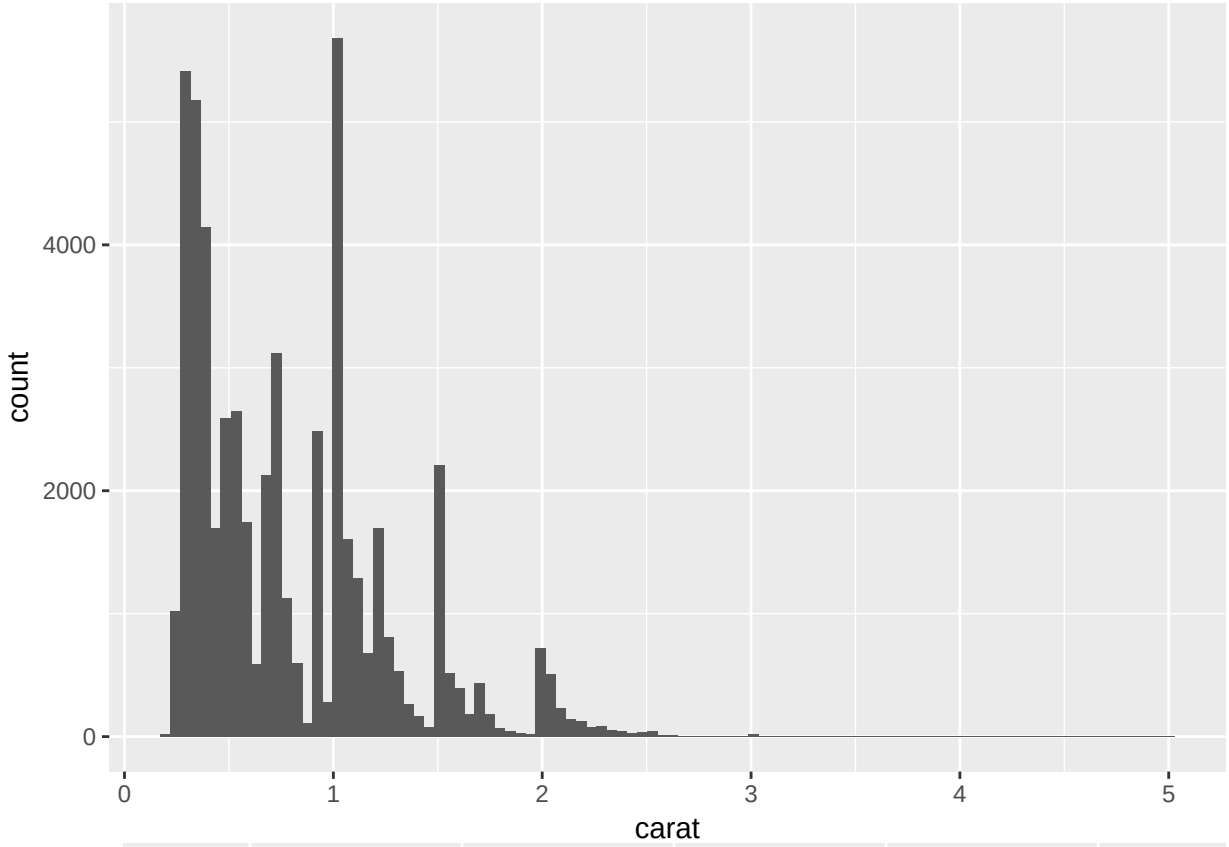
What is n, p, what do the features mean, what is the most likely response metric and why?

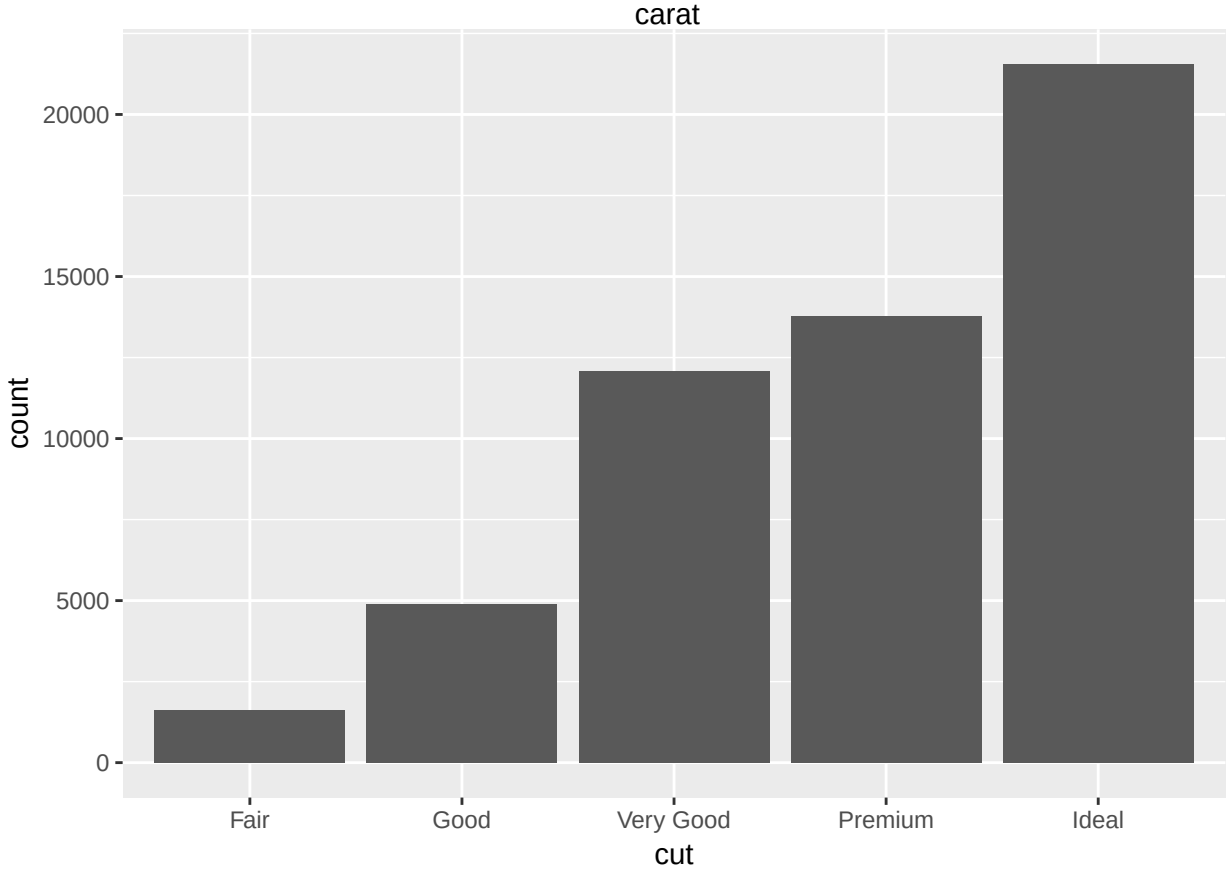
***TO-DO # It's a dataset about a sample of diamonds, their price, size, weight, and quality measures- cut, color, and clarity. n = 53940 observations of 10 variables, though I'd call p = 8 because I'm not including the response variable, and depth is a function of the size variables. This dataset was probably generated in a business setting, where price is the most important variable to predict, thus I expect price is the response metric.

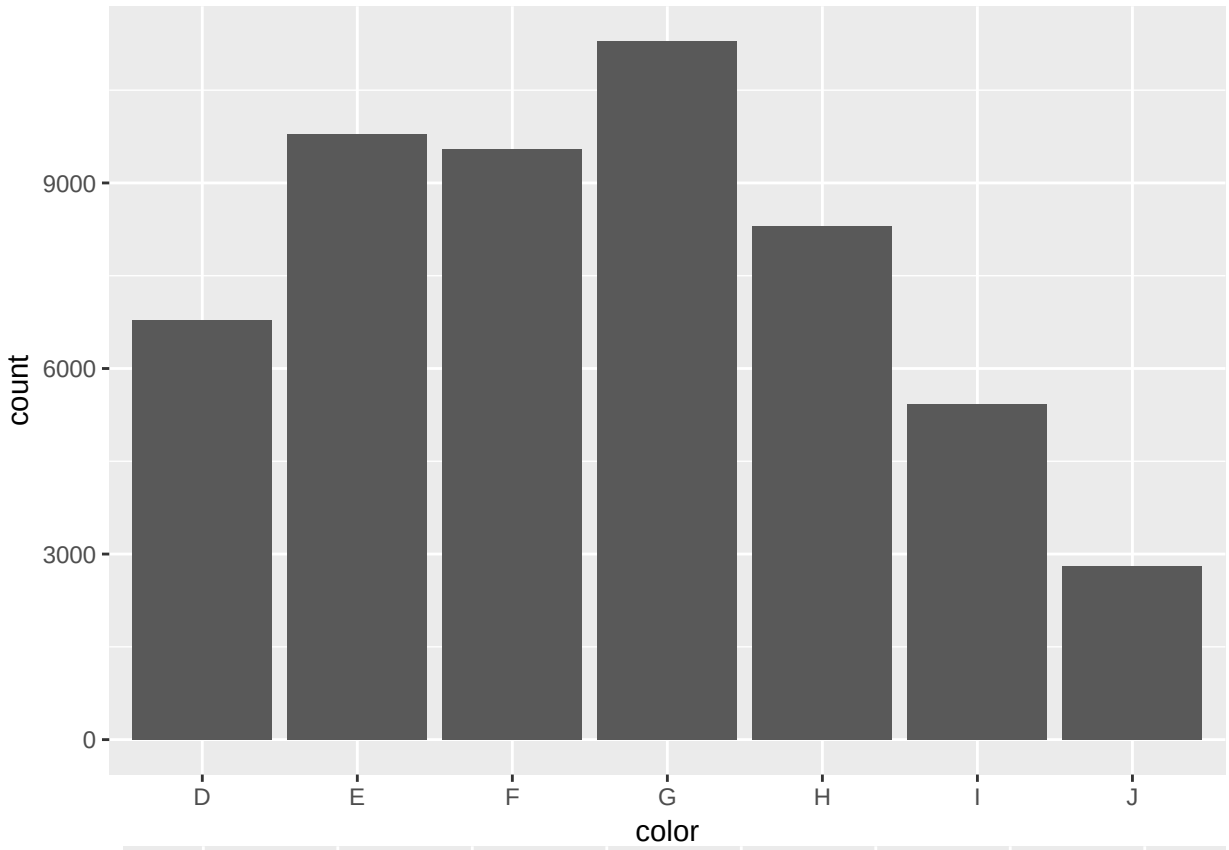
Regardless of what you wrote above, the variable `price` will be the response variable going forward.

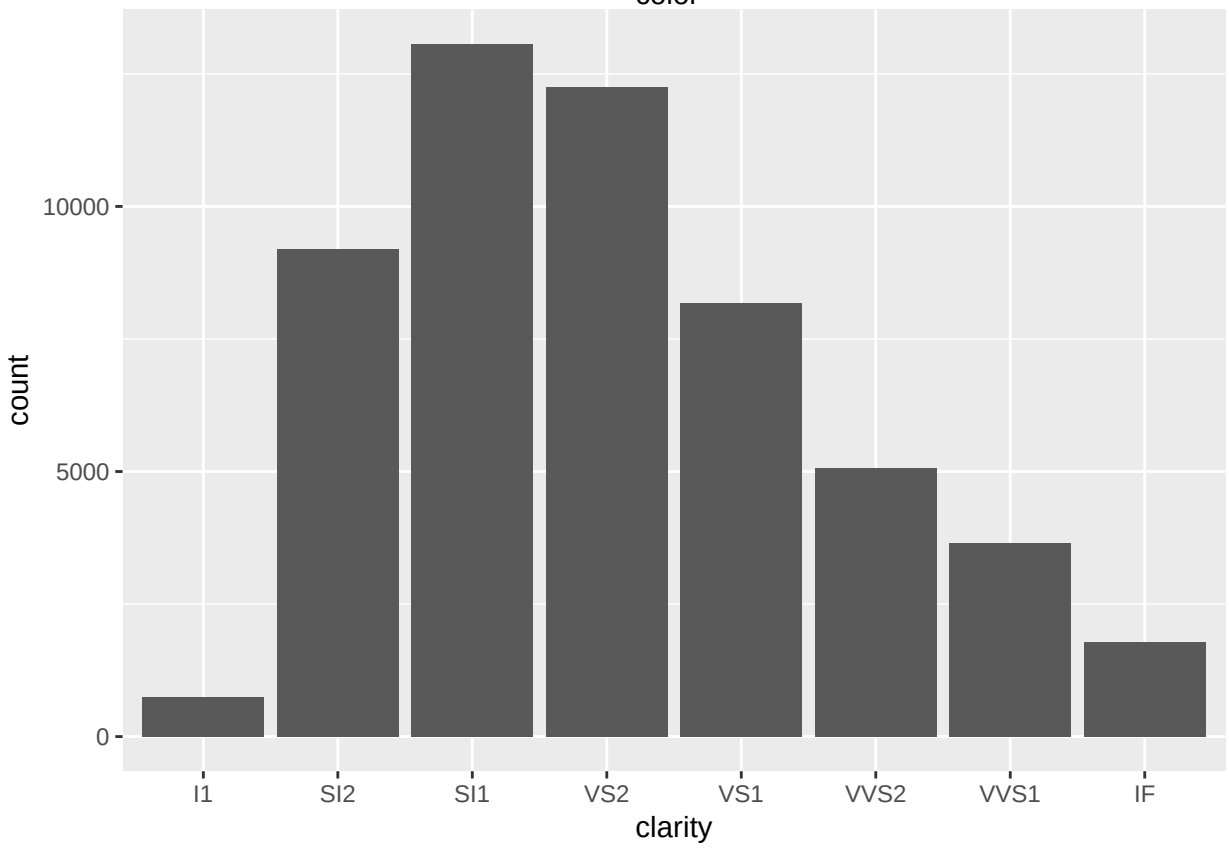
Use `ggplot` to look at the univariate distributions of *all* predictors. Make sure you handle categorical predictors differently from continuous predictors.

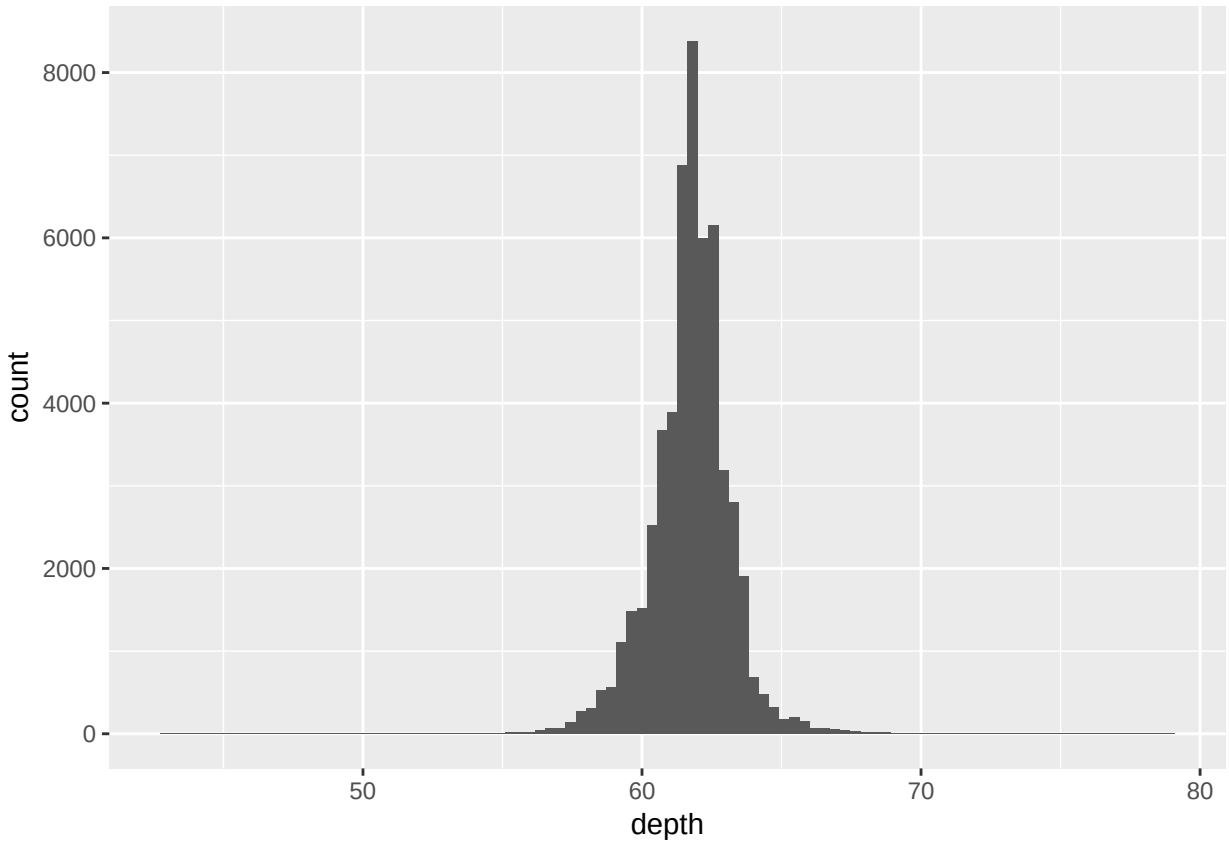
```
#TO-DO

for(x_var in colnames(diamonds)[-7]) {
     c = x_var
     if(all(class(diamonds[[c]]) == "numeric")) {
          var_plots = ggplot(diamonds, aes_string(x_var)) +
               geom_histogram(bins = 100)
     } else {
          var_plots = ggplot(diamonds, aes_string(x_var)) +
               geom_bar()
     }
     plot(var_plots)
}
```

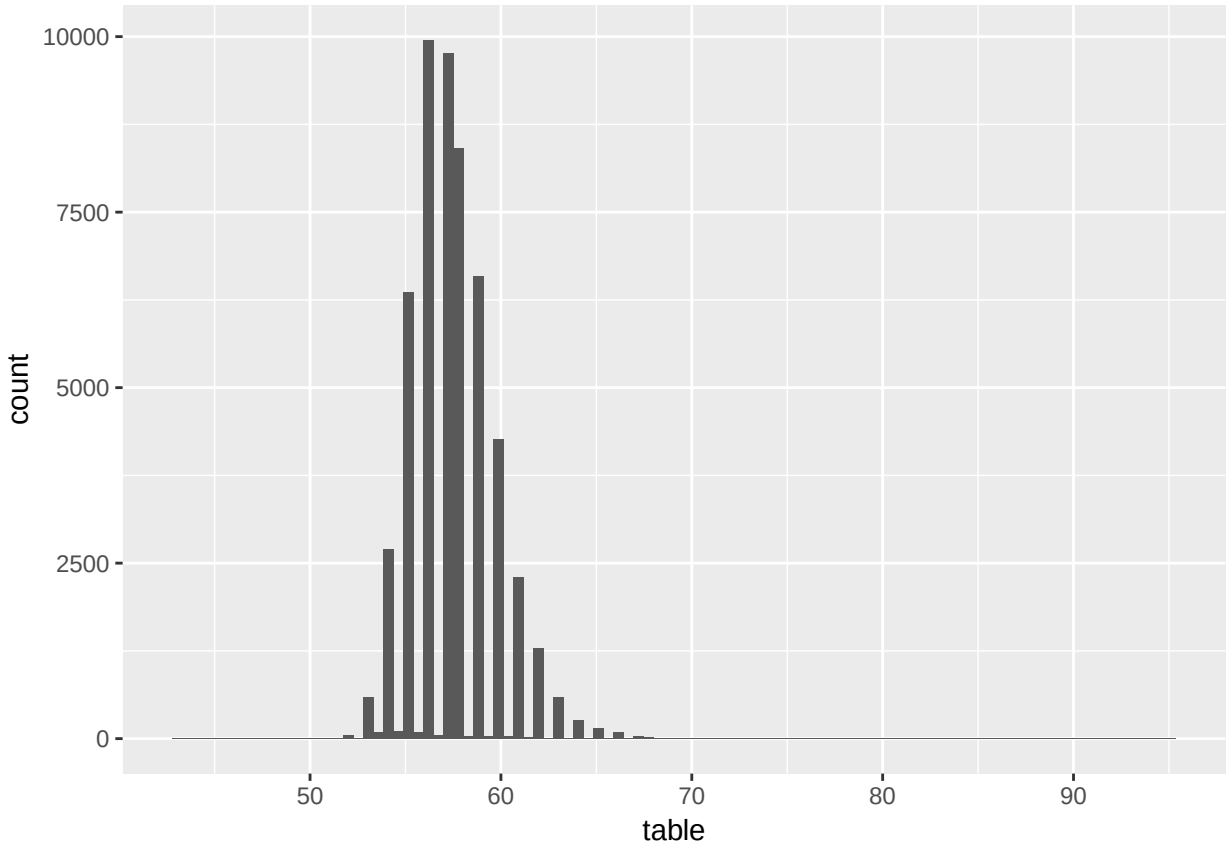

count
4000
2000
0
0
1
2
3
4
5
carat

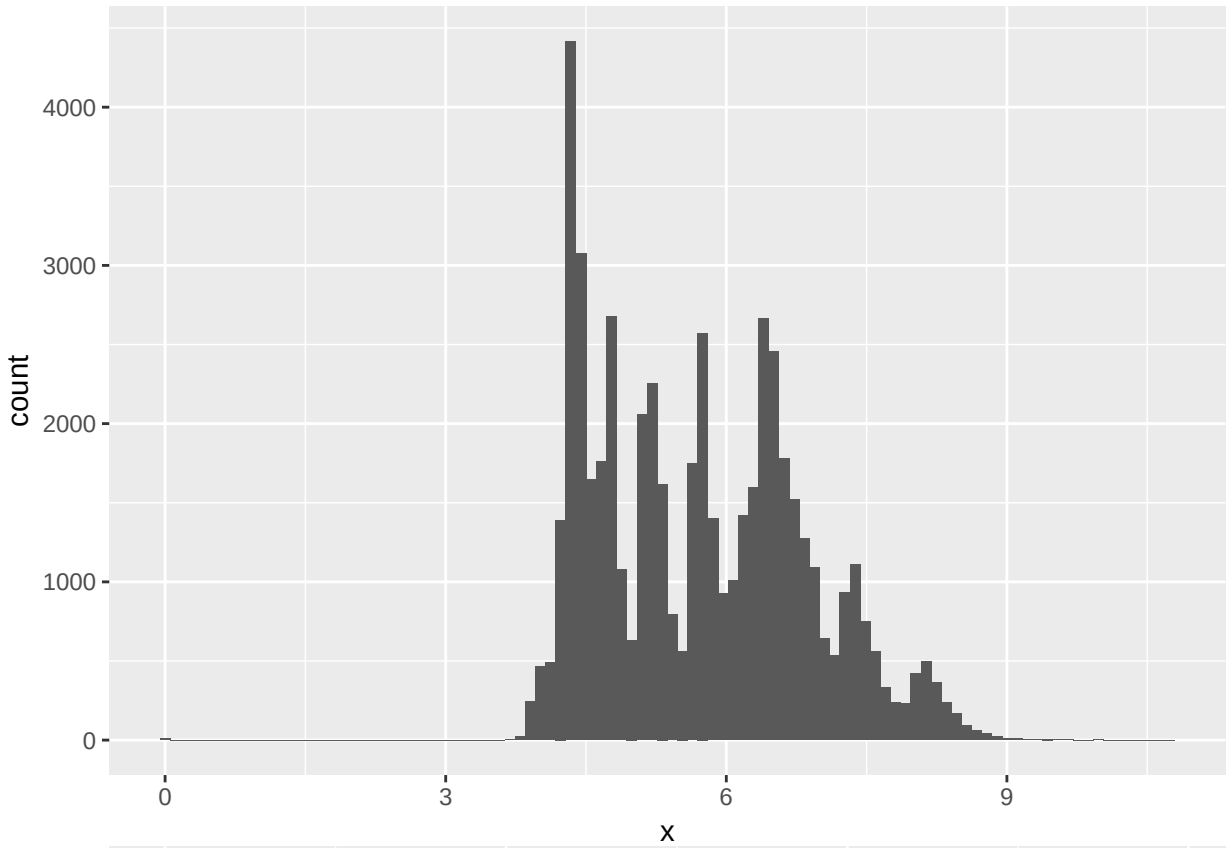

count
20000
15000
10000
5000
0
Fair
Good
Very Good
Premium
Ideal
cut

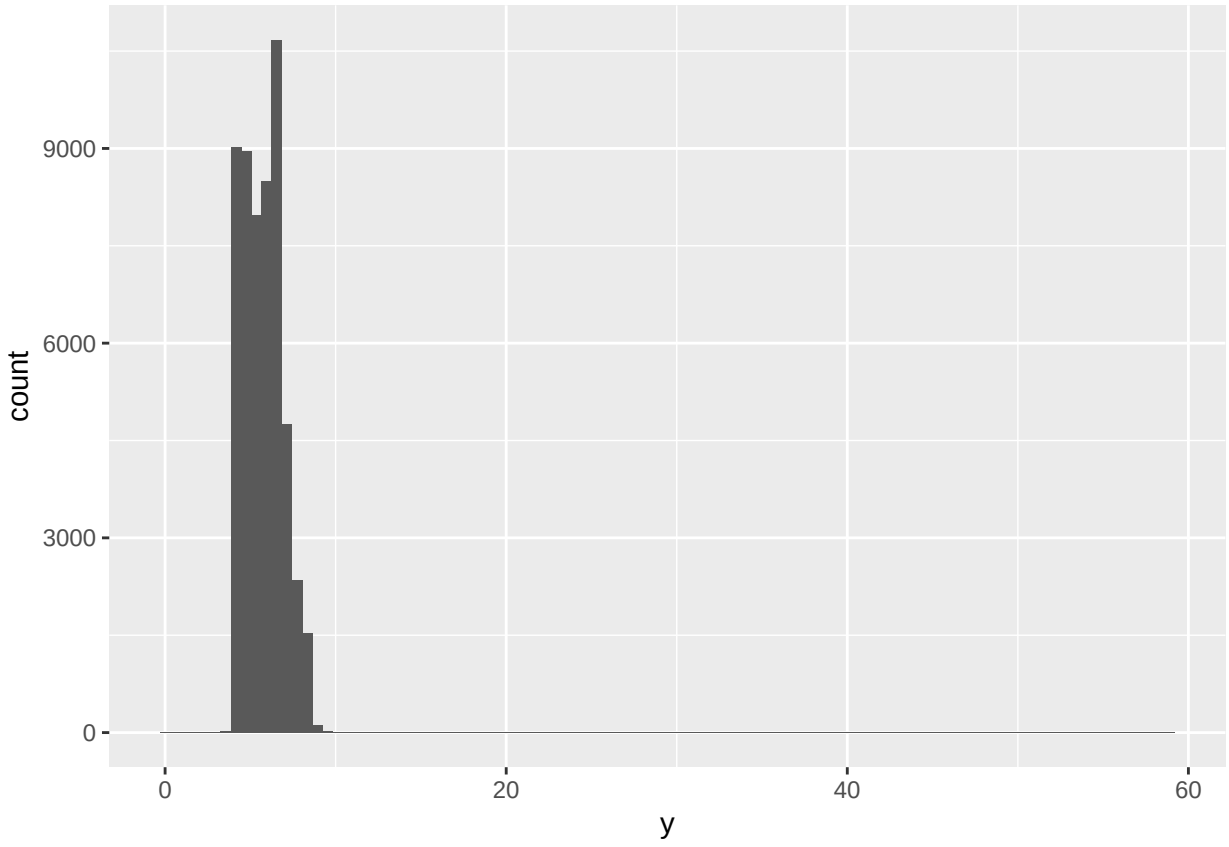

9000
6000
3000
0
count
D
E
F
G
H
I
J
color

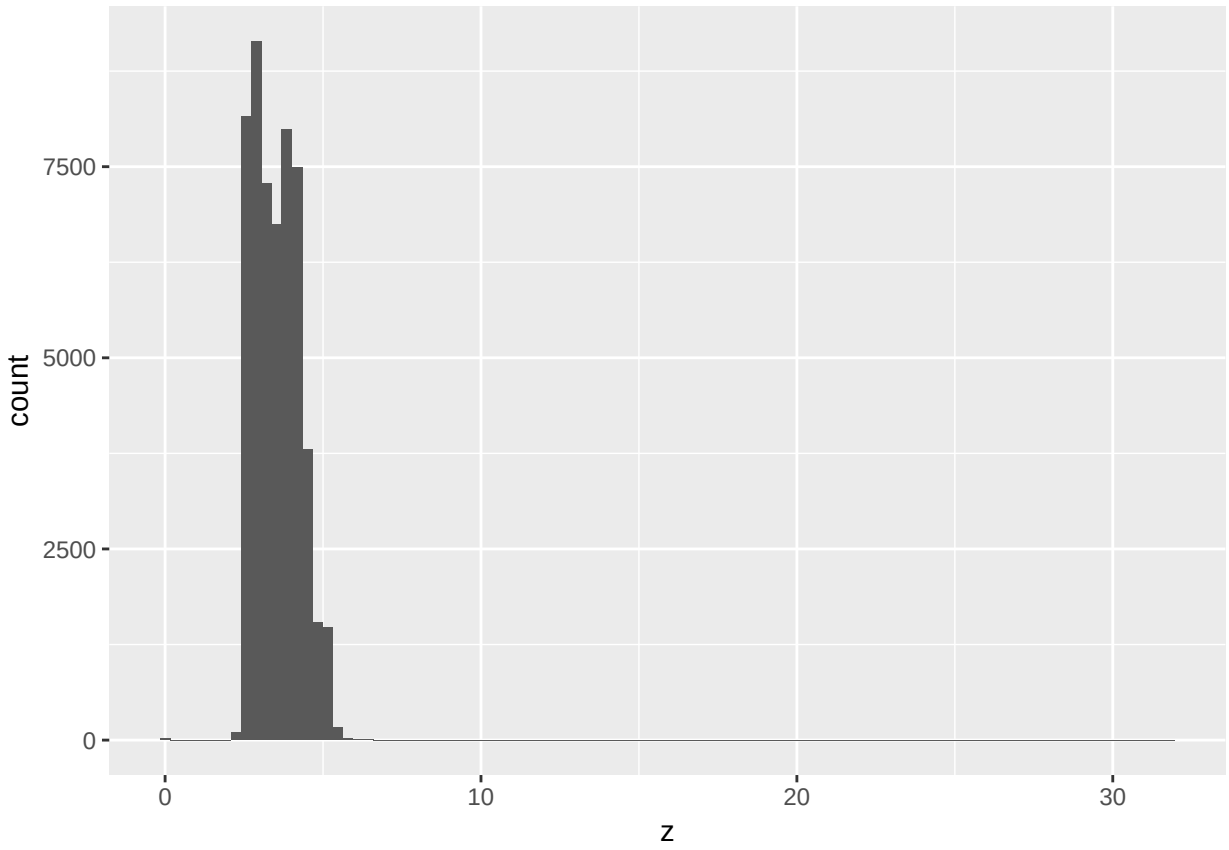

10000
5000
0
count
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
clarity


8000
6000
4000
2000
0
count
50
60
70
80
depth


10000
7500
5000
2500
0
count
50
60
70
80
90
table

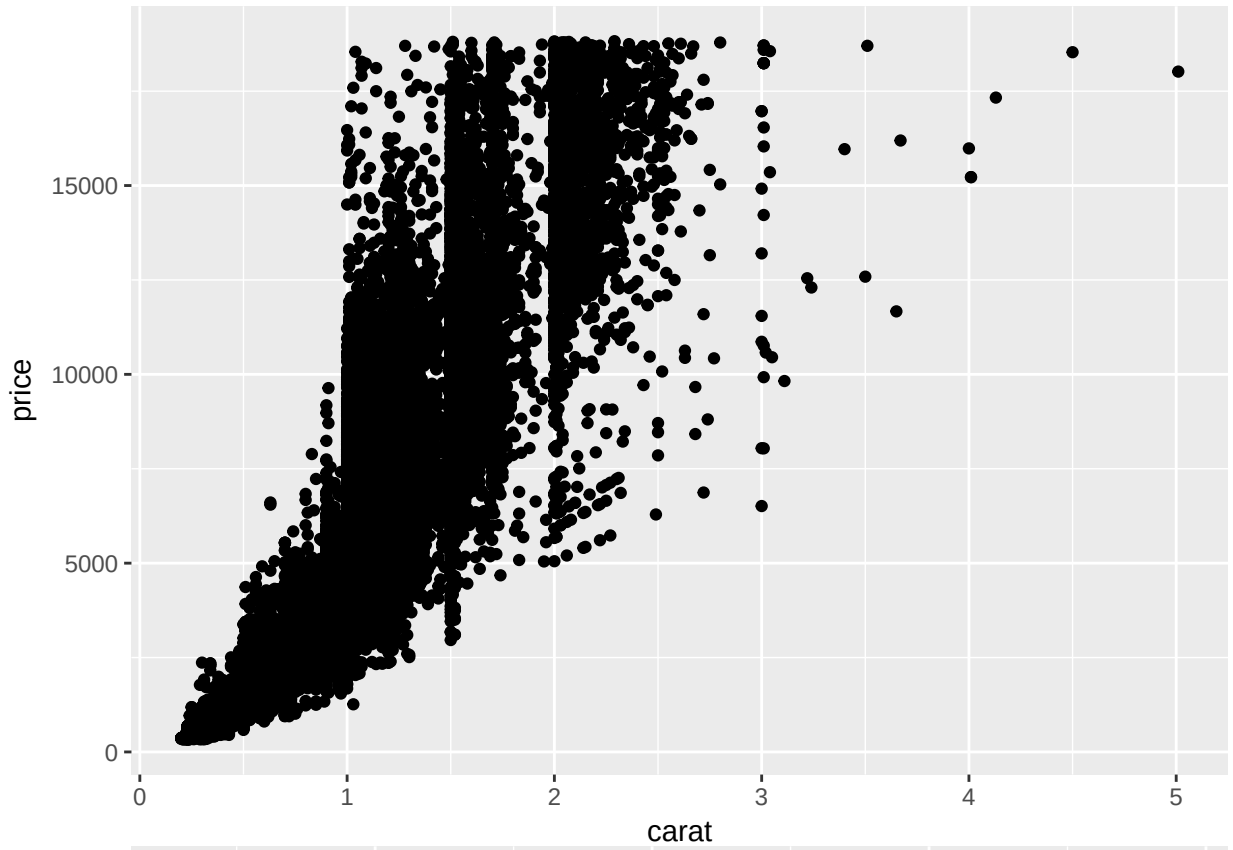

4000
3000
2000
1000
0
count
0
3
6
9
x

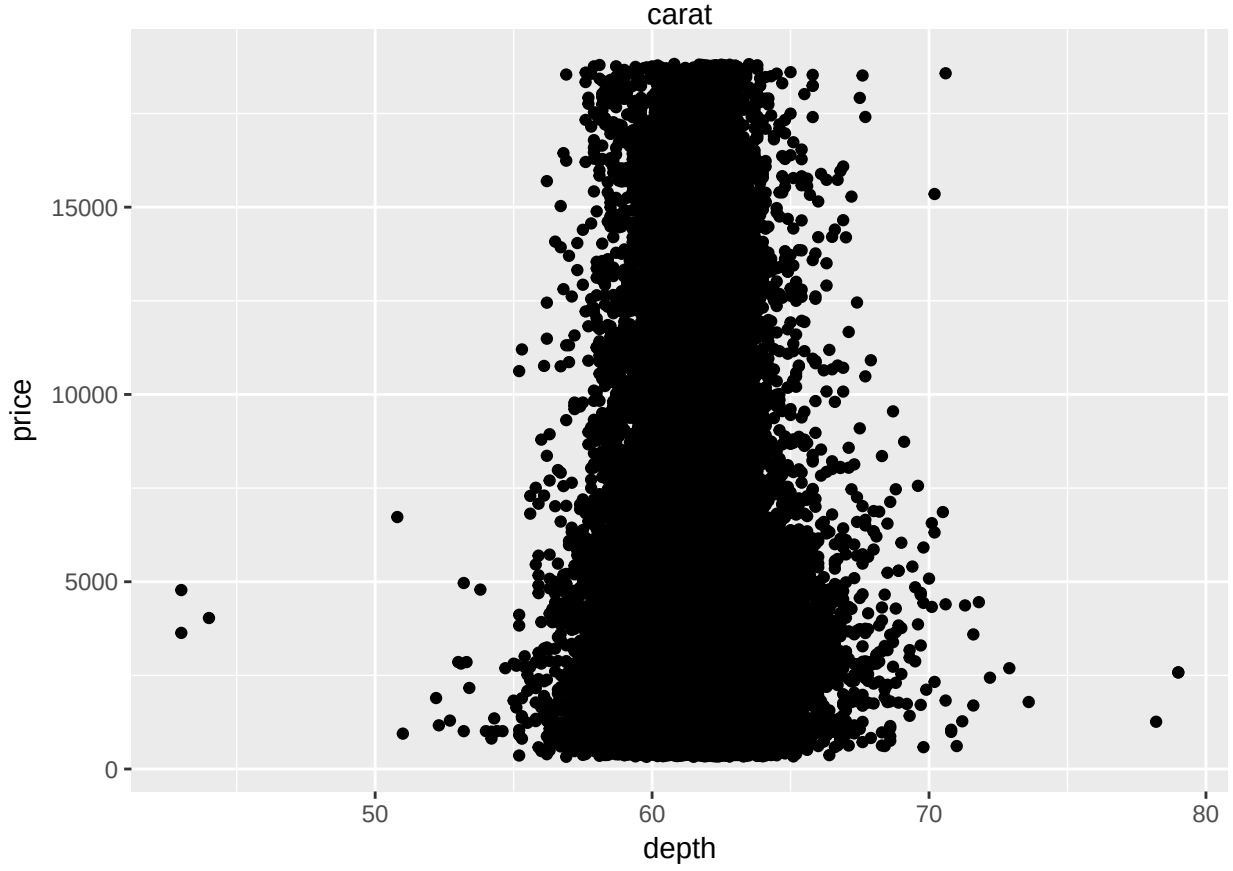

9000
6000
3000
0
count
0
20
40
60
y

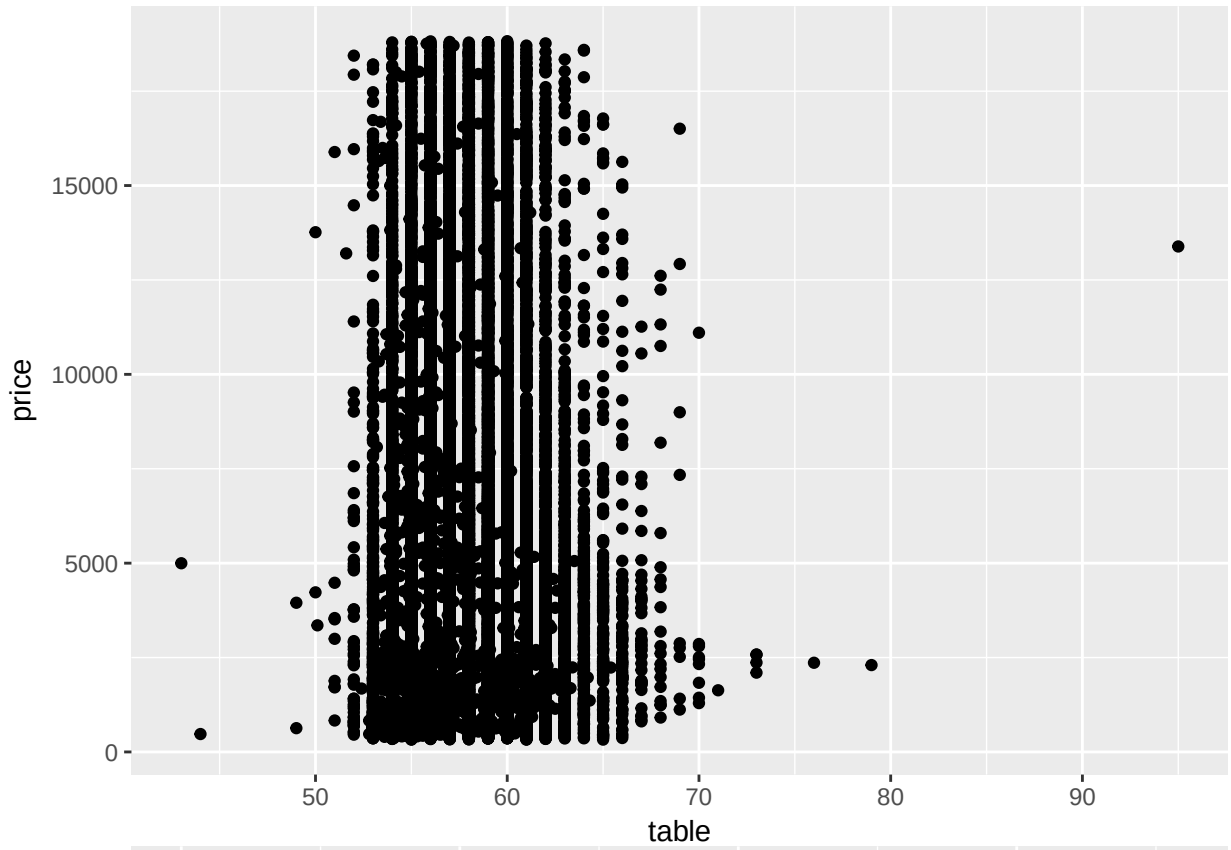


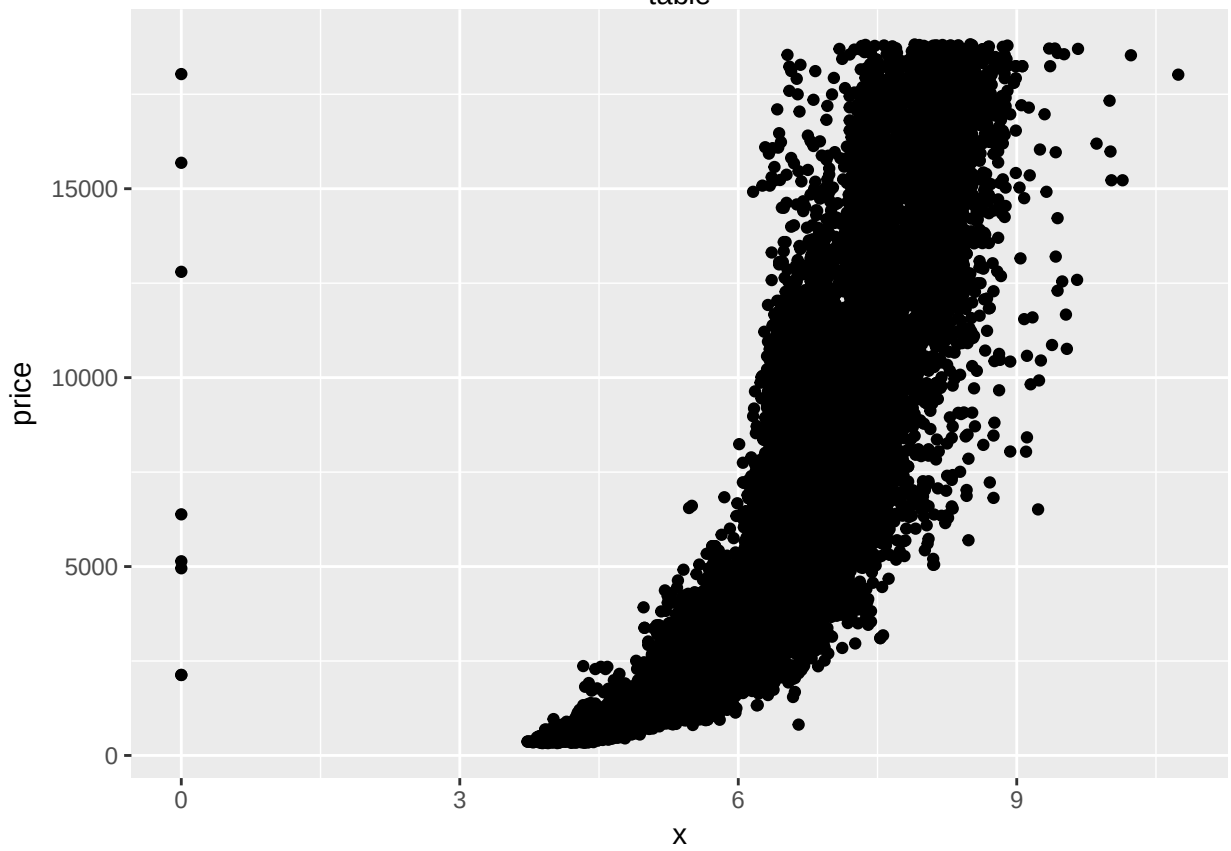
Use `ggplot` to look at the bivariate distributions of the response versus *all* predictors. Make sure you handle categorical predictors differently from continuous predictors. This time employ a for loop when an logic that handles the predictor type.

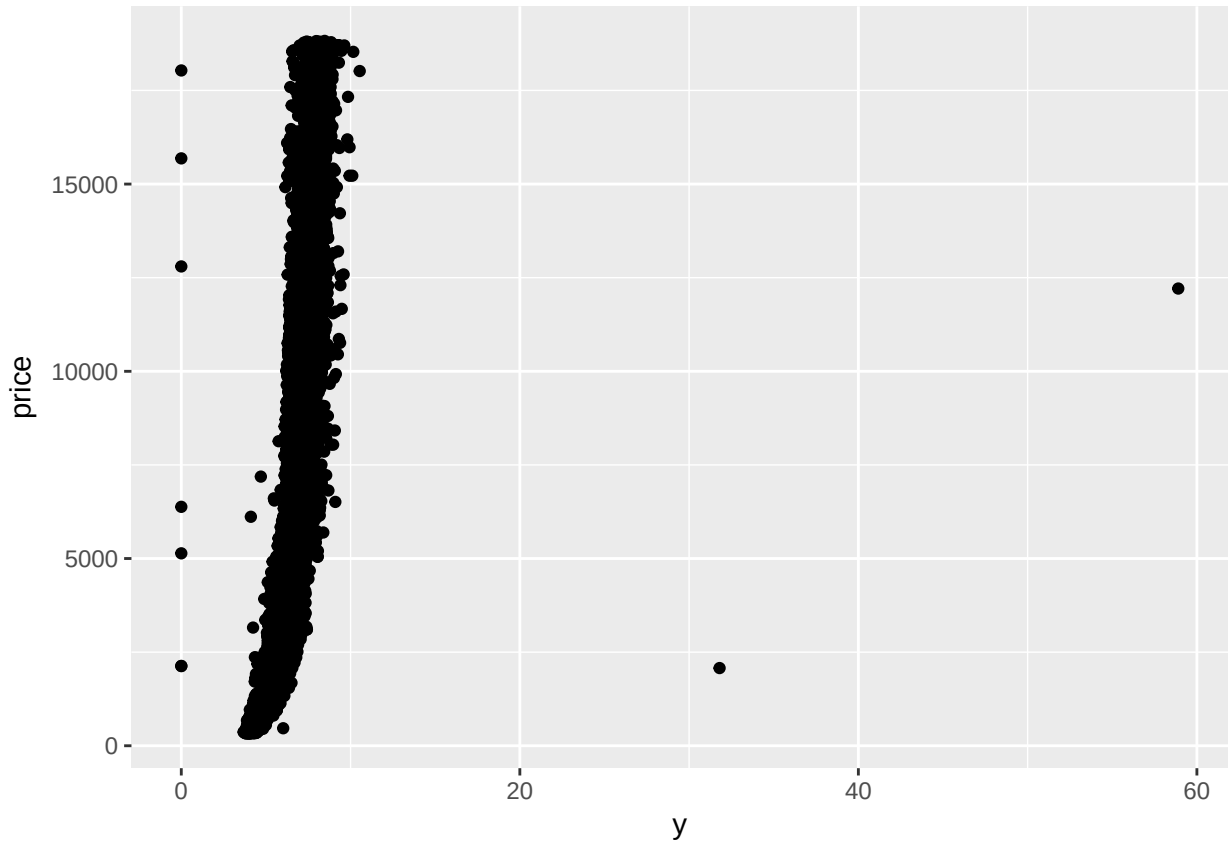
```
#TO-DO
for(x_var in colnames(diamonds)[-c(2:4,7)]) {
     var_plots = ggplot(diamonds, aes_string(x = x_var, y = "price")) +
          geom_point()
     plot(var_plots)
}
```

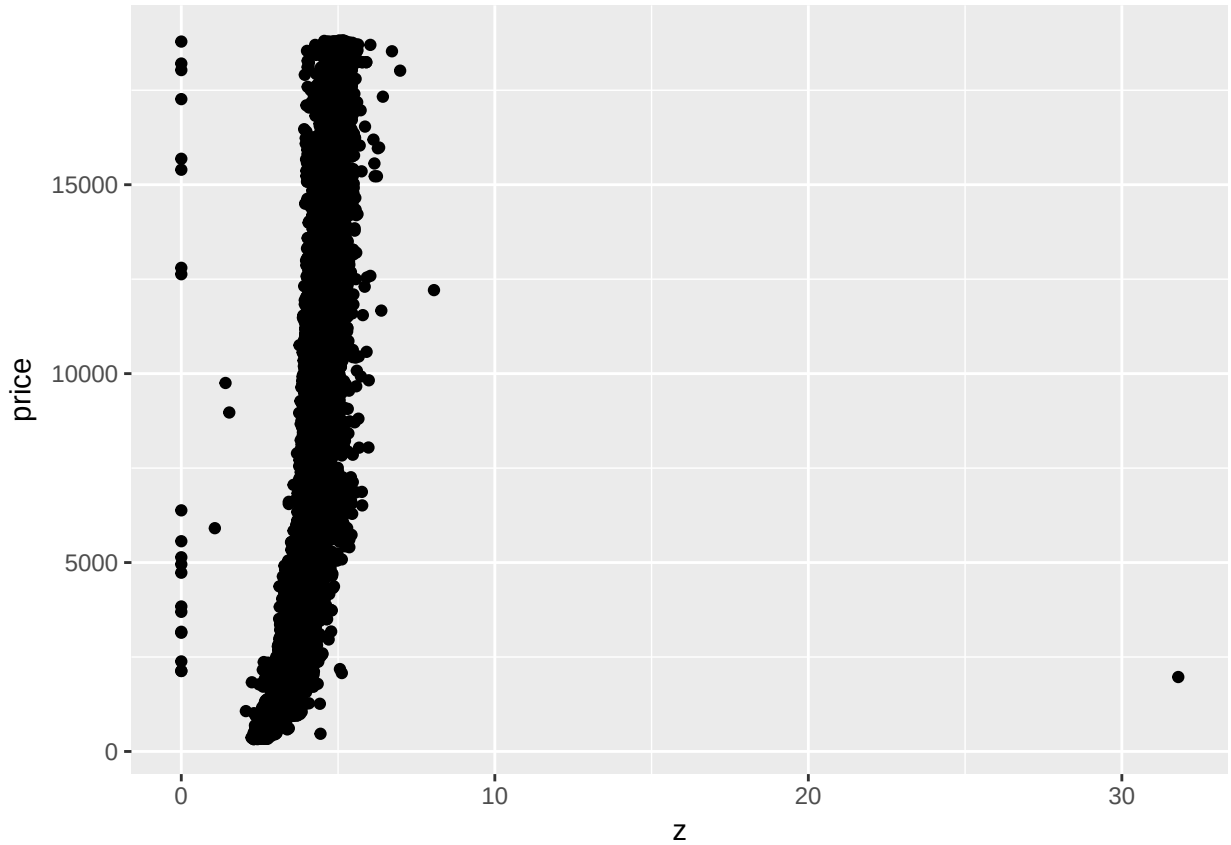

price
15000
10000
5000
0
0
1
2
3
4
5
carat


price
15000
10000
5000
0
50
60
70
80
depth

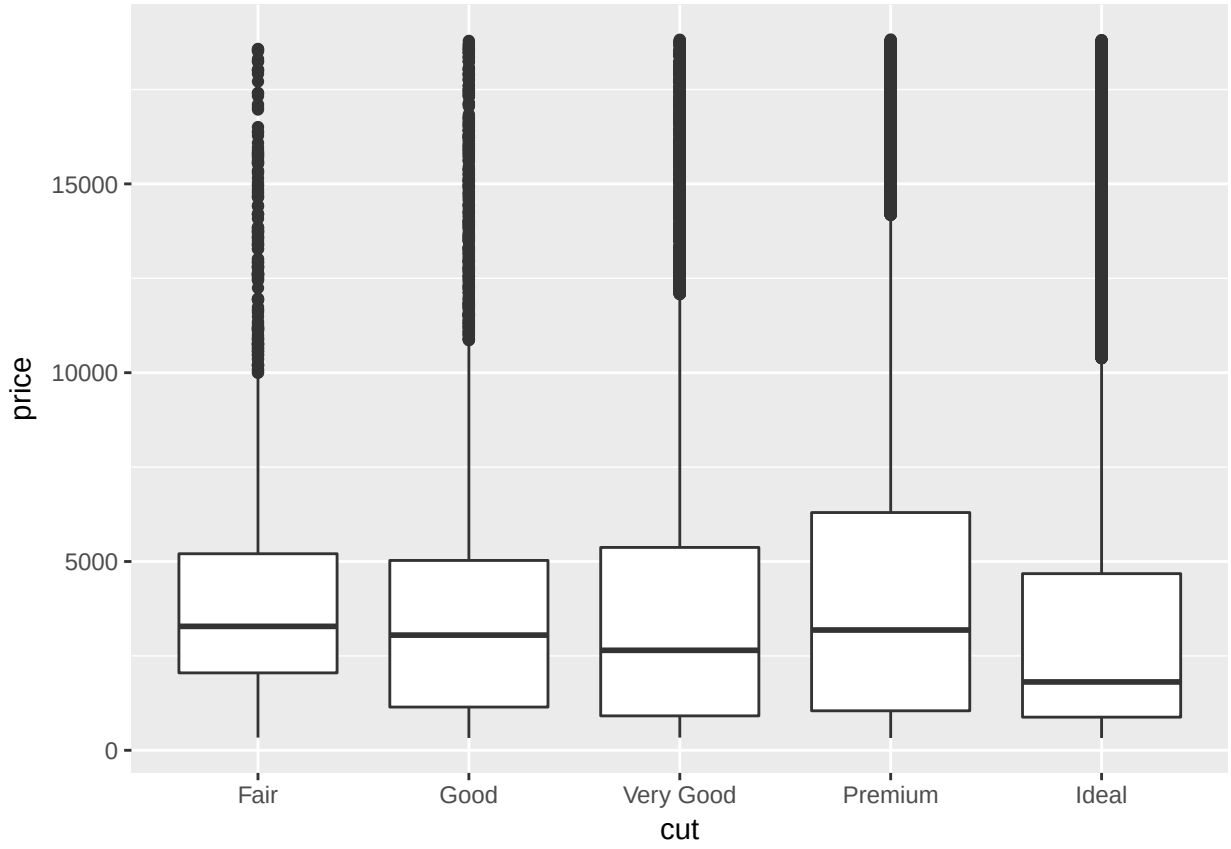

price
15000
10000
5000
0
50
60
70
80
90
table

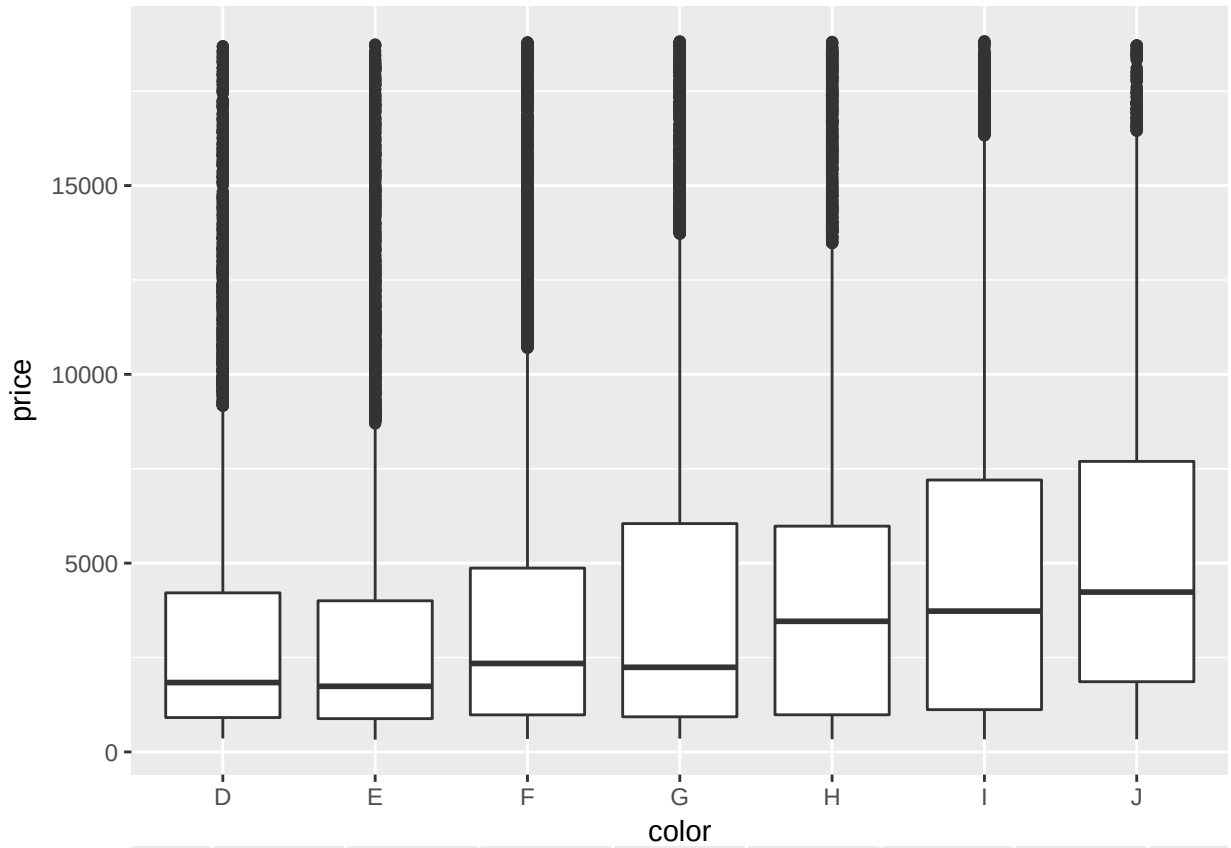

price
15000
10000
5000
0
0
3
6
9
x

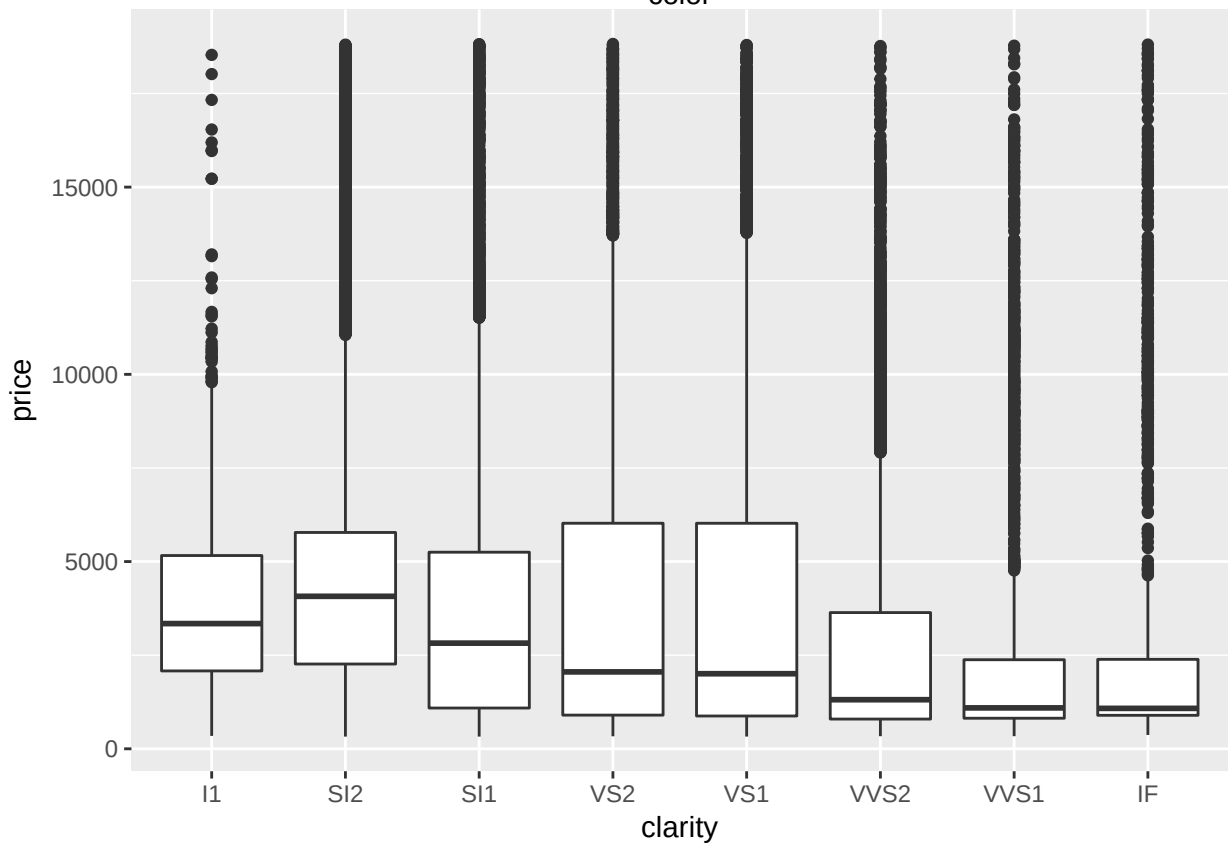

price
15000
10000
5000
0
0
20
40
60
y


price
15000
10000
5000
0
0
10
20
30
z

```
for(x_var in colnames(diamonds)[2:4]) {
     var_plots = ggplot(diamonds, aes_string(x = x_var, y = "price")) +
          geom_boxplot()
     plot(var_plots)
}
```





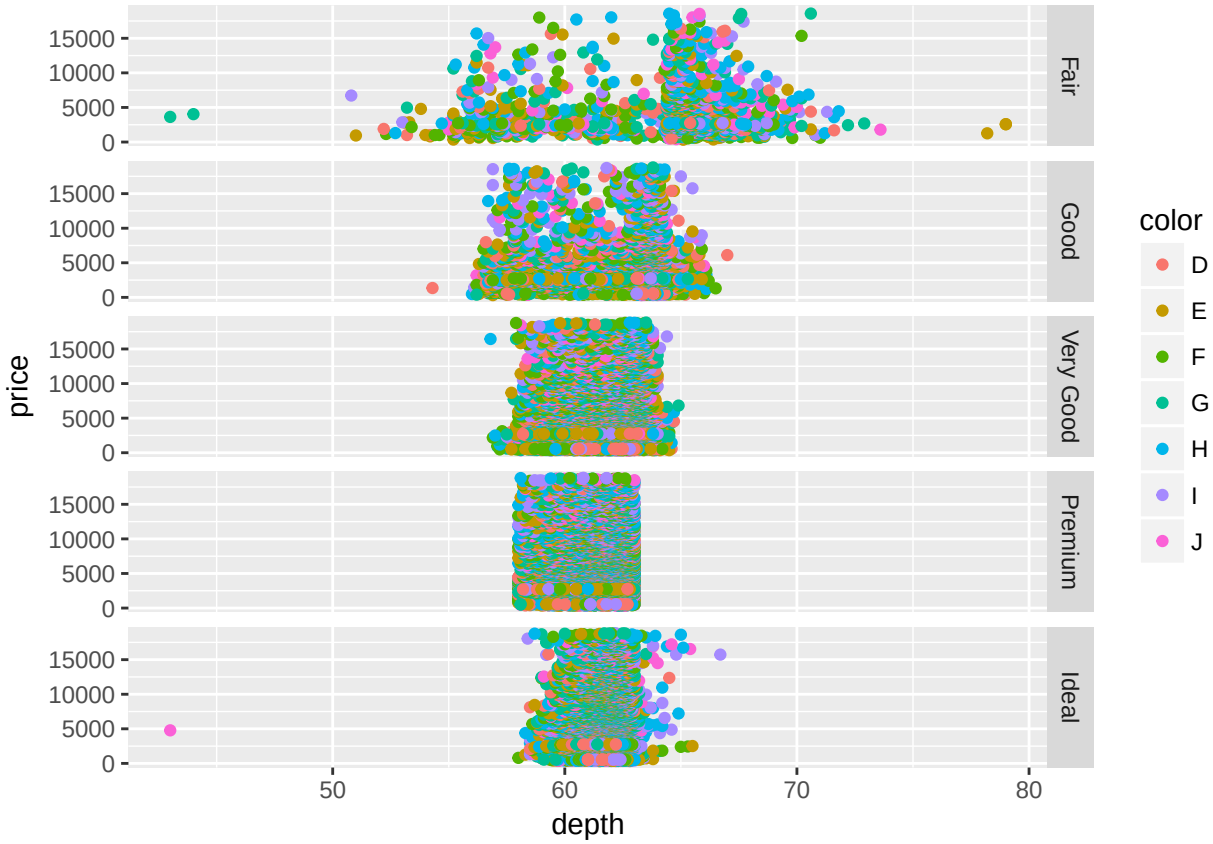


Does depth appear to be mostly independent of price?

**TO-DO # It appears independent in mean, but not in variance. Depth varies more when price is lower.

Look at depth vs price by predictors cut (using faceting) and color (via different colors).

```
#TO-DO
ggplot(diamonds, aes(x = depth, y = price)) +
     geom_point(aes(col = color)) +
     facet_grid(cut ~ .)
```



Does diamond color appear to be independent of diamond depth?

**TO-DO # Yes, the color seems to be evenly mixed width-wise in all graphs.

Does diamond cut appear to be independent of diamond depth?

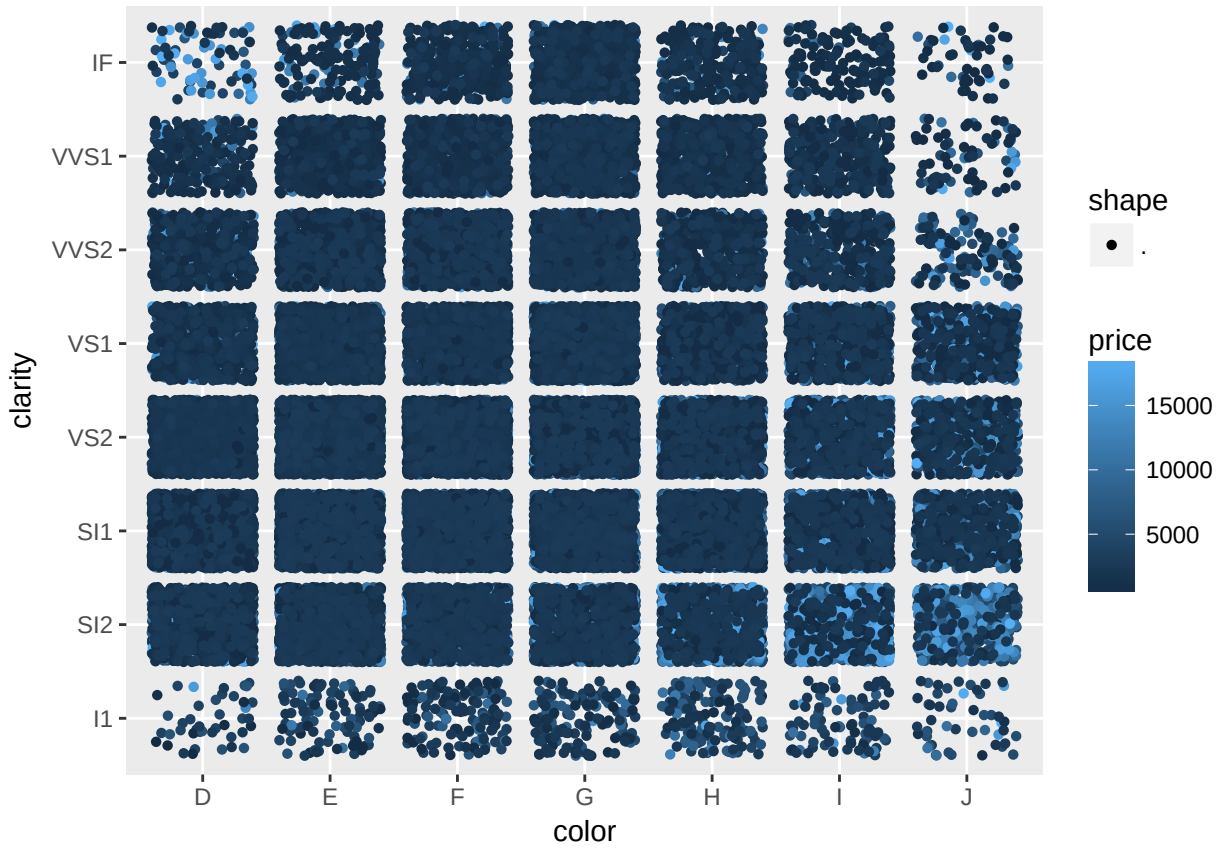
**TO-DO # No, diamond depth seems to vary more for lower-quality cuts.

Do these plots allow you to assess well if diamond cut is independent of diamond price? Yes / no

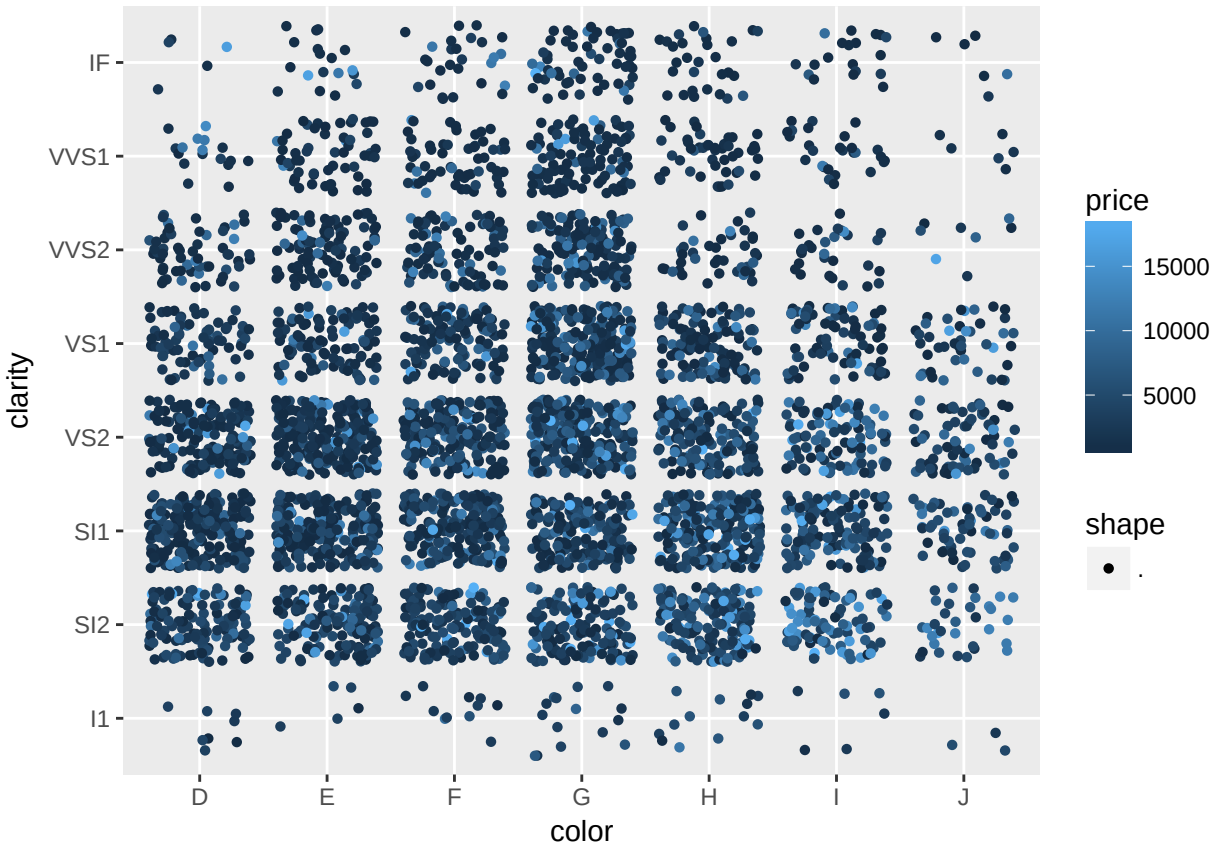
**TO-DO # Not really, the plot is too dense to determine if the density is greater higher on the y-axis (price) for the various facets (cut).

We never discussed in class bivariate plotting if both variables were categorical. Use the geometry "jitter" to visualize color vs clarity. visualize price using different colors. Use a small sized dot.

```
#TO-DO
ggplot(diamonds, aes(color, y = clarity)) +
     geom_jitter(aes(col = price, shape = "."))
```




```
sample_5000 = sample(1:53940, 5000)
ggplot(diamonds[sample_5000,], aes(color, y = clarity)) +
     geom_jitter(aes(col = price, shape = "."))
```



Does diamond clarity appear to be mostly independent of diamond color?

**TO-DO # Yes, but it's hard to pick up trends from this graph because it's hard to compare categories with different numbers of occurences. It would be easier to see on overlapping bar graphs subset by color or clarity, or by faceting by color or clarity.

2. Use `lm` to run a least squares linear regression using depth to explain price.

```
#TO-DO
dvp = lm(price ~ depth, diamonds)
summary(dvp)
```

```
##
## Call:
## lm(formula = price ~ depth, data = diamonds)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3766  -2986  -1521   1396  14937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5763.67     740.56   7.783 7.21e-15 ***
## depth         -29.65      11.99  -2.473   0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3989 on 53938 degrees of freedom
## Multiple R-squared:  0.0001134,  Adjusted R-squared:  9.483e-05
```

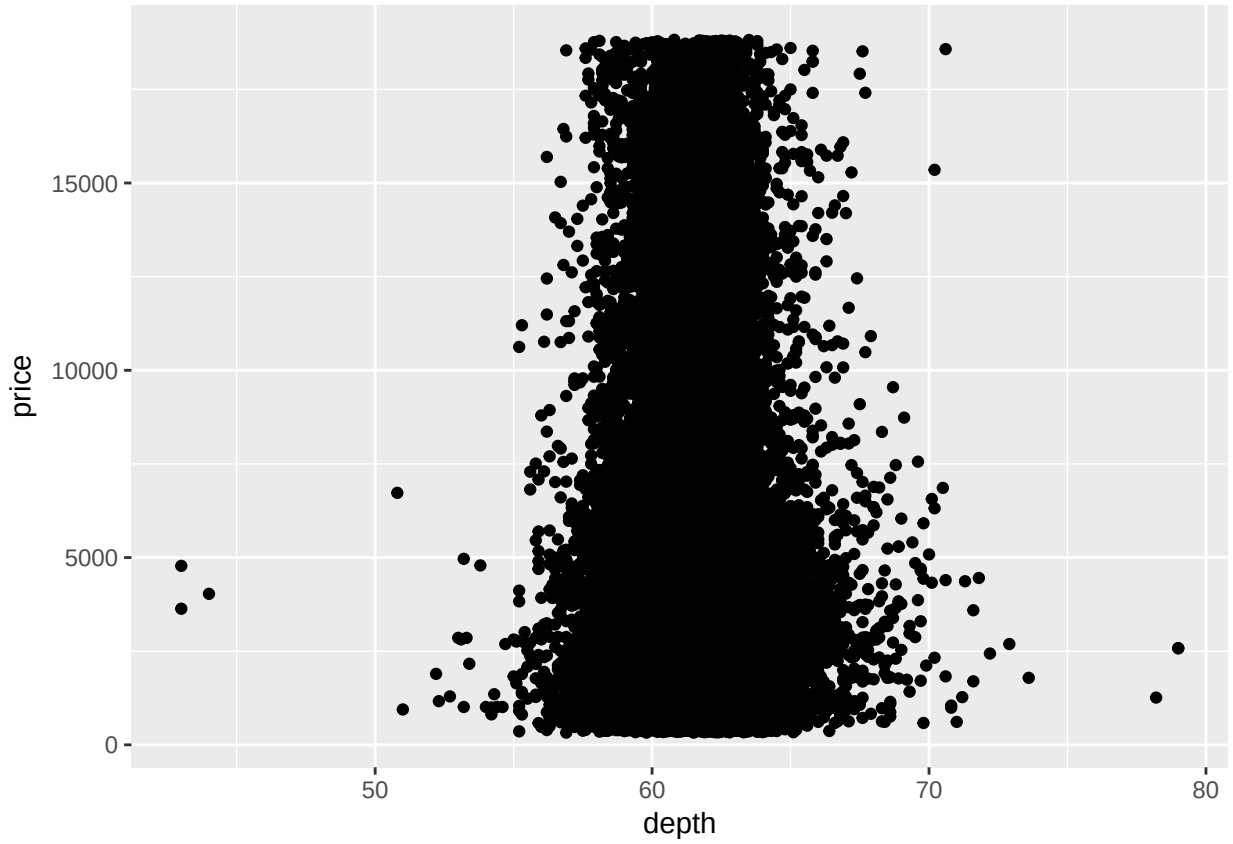
```
## F-statistic: 6.115 on 1 and 53938 DF,  p-value: 0.0134
```

What is b, R^2 and the RMSE? What was the standard error of price originally?

```
#TO-DO
list(
     "b = " = as.numeric(dvp$coefficients[2]),
     "R^2 = " = summary(dvp)$r.squared,
     "RMSE = " = summary(dvp)$sigma,
     "SD[price] = " = sd(diamonds$price)
)
```

```
## $`b = `
## [1] -29.64997
##
## $`R^2 = `
## [1] 0.0001133672
##
## $`RMSE = `
## [1] 3989.251
##
## $`SD[price] = `
## [1] 3989.44
```

```
ggplot(diamonds, aes(x = depth, y = price)) +
          geom_point()
```



Are these metrics expected given the appropriate or relevant visualization(s) above?

**TO-DO # These seem appropriate. The coefficient of ~-30 is approximately = 0 since the regression line

will only vary by $1200 across the entire range of depths, which is very small compared to the total range of $20,000 in price. The R^2 is small since depth explains very little of the variance in price. And the RMSE value is almost exactly the same as the null model standard deviation because we have explained none of the variance in price.

Use `lm` to run a least squares linear regression using carat to explain price.

```
#TO-DO
caratvprice = lm(price ~ carat, diamonds)
summary(caratvprice)
```

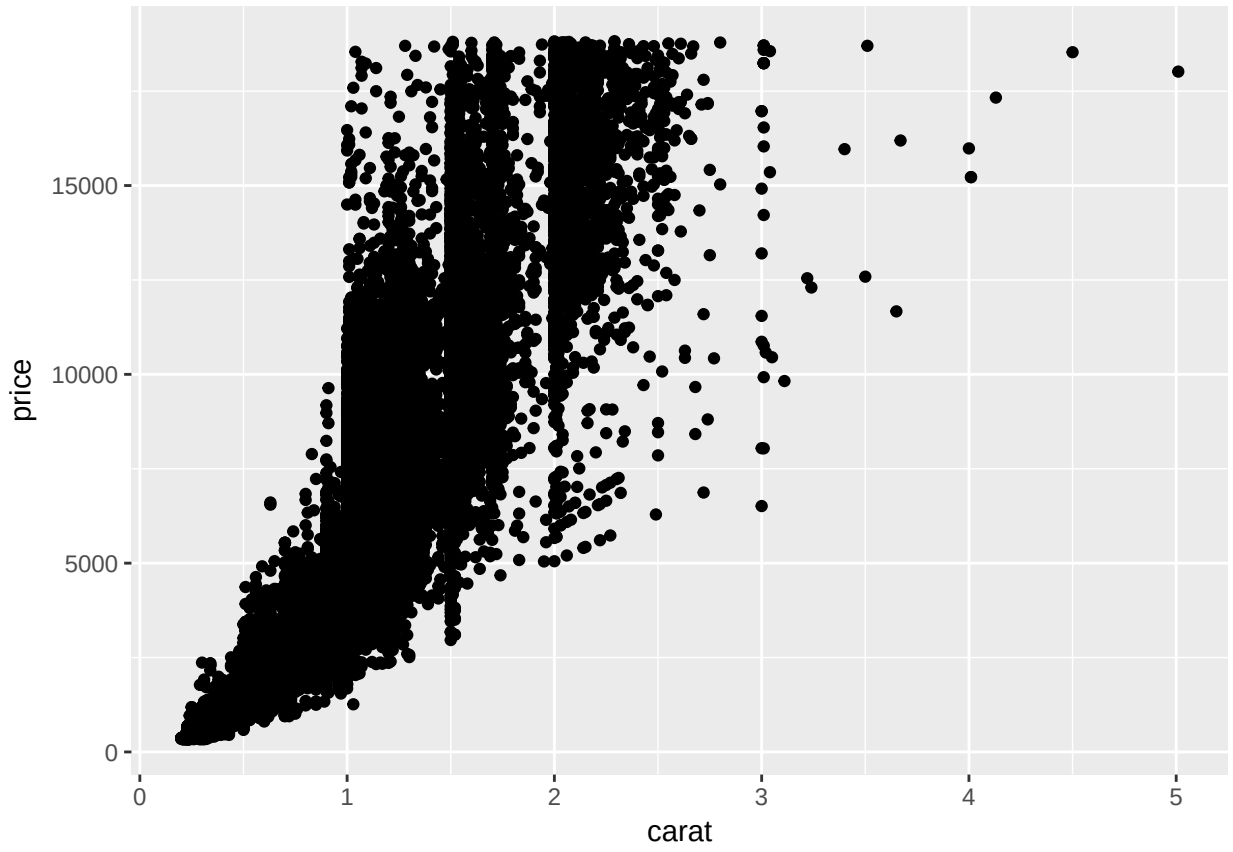
```
##
## Call:
## lm(formula = price ~ carat, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18585.3   -804.8    -18.9    537.4  12731.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2256.36      13.06  -172.8   <2e-16 ***
## carat        7756.43      14.07   551.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1549 on 53938 degrees of freedom
## Multiple R-squared:  0.8493, Adjusted R-squared:  0.8493
## F-statistic: 3.041e+05 on 1 and 53938 DF,  p-value: < 2.2e-16
```

What is b, R^2 and the RMSE? What was the standard error of price originally?

```
#TO-DO
list(
     "b = " = as.numeric(caratvprice$coefficients[2]),
     "R^2 = " = summary(caratvprice)$r.squared,
     "RMSE = " = summary(caratvprice)$sigma,
     "SD[price] = " = sd(diamonds$price)
)
```

```
## $`b = `
## [1] 7756.426
##
## $`R^2 = `
## [1] 0.8493305
##
## $`RMSE = `
## [1] 1548.562
##
## $`SD[price] = `
## [1] 3989.44
```

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



Are these metrics expected given the appropriate or relevant visualization(s) above?

**TO-DO # These metrics also seem appropriate. The coefficient of ~8000 makes sense given that the weight of the diamond and its price seem to be highly correlated, both on the graph, and intuitively. The R^2 is small since depth explains very little of the variance in price. And the RMSE value is about 39% of the pre-regression standard error because we have explained a significant amount of the variance in price using the carat variable.

3. Use `lm` to run a least squares anova model using color to explain price.

```
#TO-DO
colormod = lm(price ~ as.character(color), diamonds)
names(colormod$coefficients) = c("(Intercept)", "color E", "color F", "color G", "color H", "color I",
colormod
```

```
##
## Call:
## lm(formula = price ~ as.character(color), data = diamonds)
##
## Coefficients:
## (Intercept)      color E      color F      color G      color H
##      3170.0        -93.2        554.9        829.2       1316.7
##     color I      color J
##      1921.9       2153.9
```

```
summary(colormod)
```

```
##
## Call:
## lm(formula = price ~ as.character(color), data = diamonds)
```

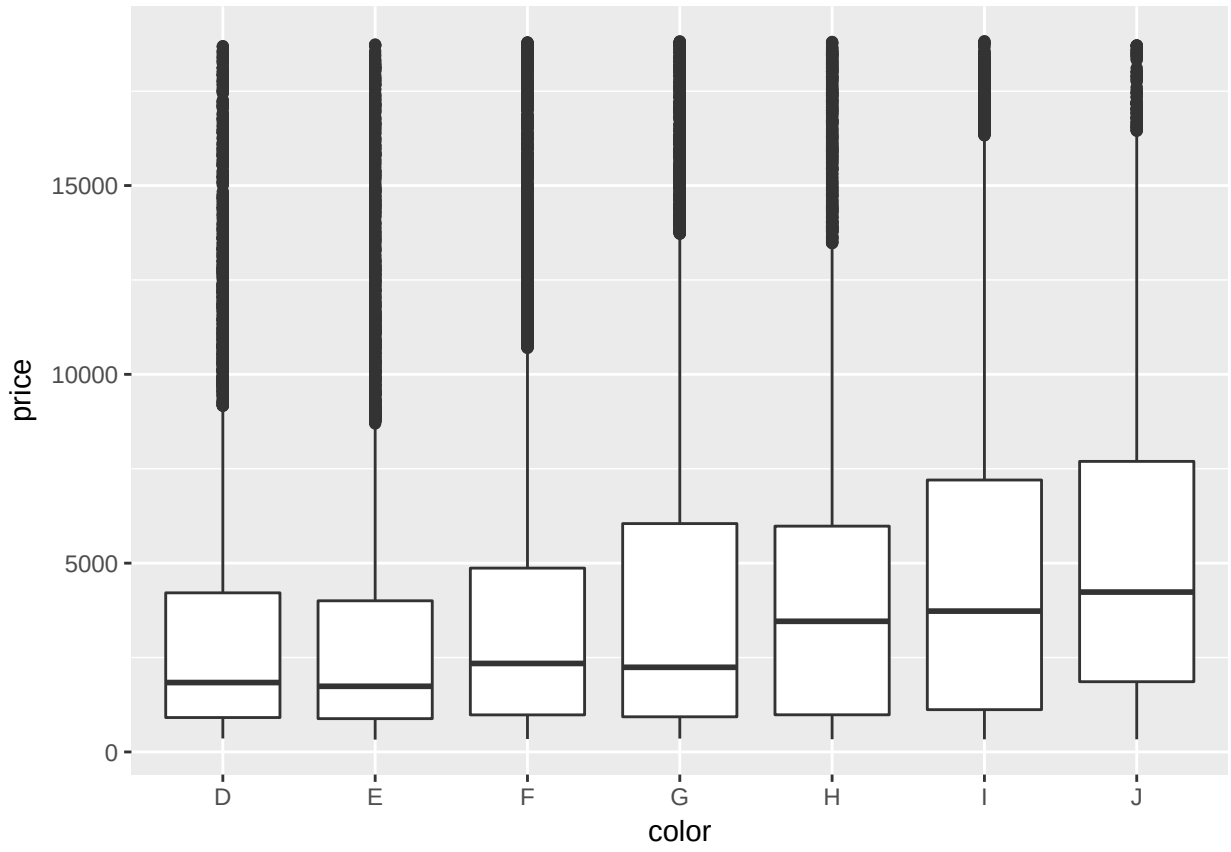
```
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4989  -2619  -1376   1374  15654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3169.95      47.71  66.446   <2e-16 ***
## color E       -93.20      62.05  -1.502    0.133
## color F       554.93      62.39   8.895   <2e-16 ***
## color G       829.18      60.34  13.741   <2e-16 ***
## color H      1316.72      64.29  20.482   <2e-16 ***
## color I      1921.92      71.55  26.860   <2e-16 ***
## color J      2153.86      88.13  24.439   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3927 on 53933 degrees of freedom
## Multiple R-squared:  0.03128,    Adjusted R-squared:  0.03117
## F-statistic: 290.2 on 6 and 53933 DF,  p-value: < 2.2e-16
```

What is b, R^2 and the RMSE? What was the standard error of price originally?

```
#TO-DO
list(
     "b = " = coef(colormod),
     "R^2 = " = summary(colormod)$r.squared,
     "RMSE = " = summary(colormod)$sigma,
     "SD[price] = " = sd(diamonds$price)
)
```

```
## $`b = `
## (Intercept)     color E     color F     color G     color H     color I
##  3169.95410   -93.20162   554.93230   829.18158  1316.71510  1921.92086
##     color J
##  2153.86392
##
## $`R^2 = `
## [1] 0.03127542
##
## $`RMSE = `
## [1] 3926.777
##
## $`SD[price] = `
## [1] 3989.44
```

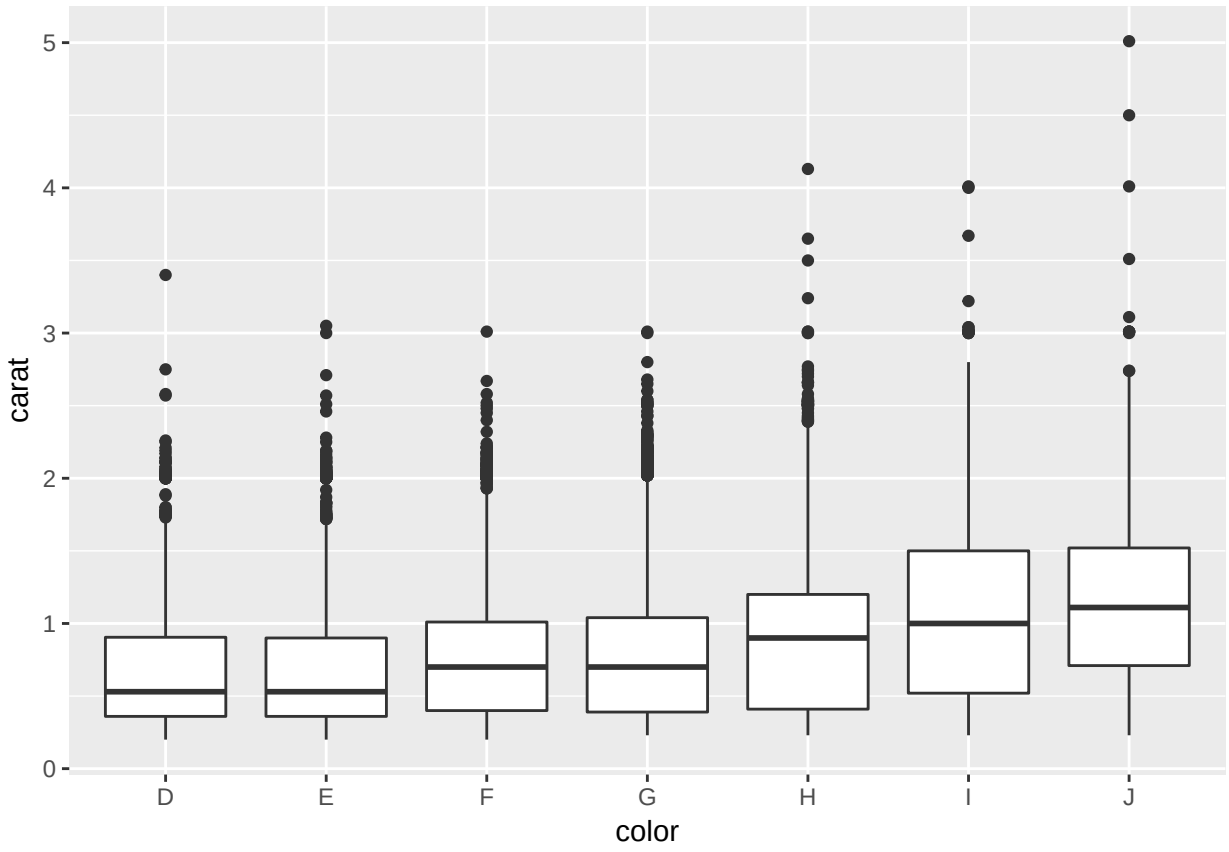
```
ggplot(diamonds, aes(x = color, y = price)) +
     geom_boxplot()
```



Are these metrics expected given the appropriate or relevant visualization(s) above?

**TO-DO # Yes, given the visualization above, the metrics are appropriate. The intercept (the mean for D) is a little bit above the median displayed on the box plot for D. The coefficient for E shows that E's mean is slightly below D's, and the rest of the coefficients show the other colors each have a progressively higher mean. # However, this doesn't make sense based on intuition about the price~color relationship. Since D is the best quality color, shouldn't it be sold at the highest prices? The only explanation I've come up with is that perhaps it's so difficult to find diamonds with a high color-quality that the few they can find are small or less valuable in some other metric. And I think I've found some evidence below. There are many more large poor-color diamonds than large good-color diamonds. Also, when you account for other variables such as carat in the regression, the coefficients for color, cut, and clarity actually switch from worse-quality diamonds giving higher prices to better-quality diamonds giving higher prices.

```
ggplot(diamonds, aes(x = color, y = carat)) +
    geom_boxplot()
```




```
lm(price ~ carat + as.character(cut) + as.character(color) + as.character(clarity) + depth + table + x +
```

```
##
## Call:
## lm(formula = price ~ carat + as.character(cut) + as.character(color) +
##     as.character(clarity) + depth + table + x + y + z, data = diamonds)
##
## Coefficients:
##                 (Intercept)                        carat
##                    2184.477                    11256.978
##      as.character(cut)Good      as.character(cut)Ideal
##                     579.751                     832.912
##   as.character(cut)Premium  as.character(cut)Very Good
##                     762.144                     726.783
##       as.character(color)E        as.character(color)F
##                    -209.118                    -272.854
##       as.character(color)G        as.character(color)H
##                    -482.039                    -980.267
##       as.character(color)I        as.character(color)J
##                   -1466.244                   -2369.398
##    as.character(clarity)IF    as.character(clarity)SI1
##                    5345.102                    3665.472
##   as.character(clarity)SI2    as.character(clarity)VS1
##                    2702.586                    4578.398
##   as.character(clarity)VS2   as.character(clarity)VVS1
##                    4267.224                    5007.759
##  as.character(clarity)VVS2                       depth
```

```
##                    4950.814                      -63.806
##                       table                            x
##                     -26.474                    -1008.261
##                           y                            z
##                       9.609                      -50.119
```

Our model only included one feature - why are there more than two estimates in b?

**TO-DO # There are more than two estimates because that's how linear models deal with categorical variables. The first estimate in b gives the mean for color level "D"; this is the intecept. The second gives the estimate for the amount the mean for "E" differs from that of "D". The third gives the amount the estimate for "F" differs from "D", and on until the last one which gives the amount "J" differs from "D".

Verify that the least squares linear model fit gives the sample averages of each price given color combination. Make sure to factor in the intercept here.

```
#TO-DO
q = sort(unique(diamonds$color))
color_means = numeric(0)
for(i in q) {
     color_means[i] = mean(diamonds$price[diamonds$color == i])
}

all.equal(as.numeric(color_means[1]), as.numeric(colormod$coefficients[1]))
```

```
## [1] TRUE
```

```
j = 2
for(i in q[-1]) {
     print(all.equal(as.numeric(color_means[i]), as.numeric(colormod$coefficients[j] + colormod$coeffic
     j = j+1
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

Fit a new model without the intercept and verify the sample averages of each colors' prices *directly* from the entries of vector b.

```
#TO-DO
nointecept = lm(price ~ 0 + color, diamonds)
nointecept
```

```
##
## Call:
## lm(formula = price ~ 0 + color, data = diamonds)
##
## Coefficients:
## colorD  colorE  colorF  colorG  colorH  colorI  colorJ
##   3170    3077    3725    3999    4487    5092    5324
```

What would extrapolation look like in this model? We never covered this in class explicitly.

**TO-DO # I think extrapolation in this model would be to assert that if there were a class "C", it would have an average price lower than class "D", and likewise if there were a class "K", it would have a higher

average price than class "J".

4. Use `lm` to run a least squares linear regression using all available features to explain diamond price.

```
#TO-DO
fullmod = lm(price ~ carat + as.character(cut) + as.character(color) + as.character(clarity) + depth + 
names(fullmod$coefficients) = c("(Intercept)", "carat", "cut Good", "cut Ideal", "cut Premium", "cut Ve
fullmod
```

```
##
## Call:
## lm(formula = price ~ carat + as.character(cut) + as.character(color) +
##     as.character(clarity) + depth + table + x + y + z, data = diamonds)
##
## Coefficients:
##   (Intercept)          carat       cut Good      cut Ideal    cut Premium
##      2184.477      11256.978        579.751        832.912        762.144
## cut Very Good        color E        color F        color G        color H
##       726.783       -209.118       -272.854       -482.039       -980.267
##       color I        color J     clarity IF    clarity SI1    clarity SI2
##     -1466.244      -2369.398       5345.102       3665.472       2702.586
##   clarity VS1    clarity VS2   clarity VVS1   clarity VVS2          depth
##      4578.398       4267.224       5007.759       4950.814        -63.806
##         table              x              y              z
##       -26.474      -1008.261          9.609        -50.119
```

What is b, R^2 and the RMSE? Also - provide an approximate 95% interval for predictions using the empirical rule.

```
#TO-DO
list(
     "b = " = coef(fullmod),
     "R^2 = " = summary(fullmod)$r.squared,
     "RMSE = " = summary(fullmod)$sigma,
     "CI, 95% = " = noquote(paste("±", 2 * summary(fullmod)$sigma))
)
```

```
## $`b = `
##   (Intercept)          carat       cut Good      cut Ideal    cut Premium
##   2184.477350  11256.978307    579.751446    832.911845    762.143950
## cut Very Good        color E        color F        color G        color H
##    726.782591   -209.118085   -272.853832   -482.038904   -980.266675
##       color I        color J     clarity IF    clarity SI1    clarity SI2
##  -1466.244474  -2369.398063   5345.102246   3665.472080   2702.586294
##   clarity VS1    clarity VS2   clarity VVS1   clarity VVS2          depth
##   4578.397915   4267.223565   5007.759045   4950.814072    -63.806100
##         table              x              y              z
##    -26.474085  -1008.261098      9.608886    -50.118891
##
## $`R^2 = `
## [1] 0.9197915
##
## $`RMSE = `
## [1] 1130.094
##
## $`CI, 95% = `
## [1] ± 2260.18885197136
```

Interpret all entries in the vector b.

**TO-DO # The intercept is the expected value for a 0 carat diamond with a Fair cut, of color D, of clarity I1, depth 0, table 0, and x, y, and z lengths of 0. The carat coefficient measures how much the price goes up on average for each 1 carat increase in diamond weight. The coefficients for the other numerical characteristics, depth, table, and x, y, and z lengths, go up on average by the amount of their coefficient for each 1 unit increase. cut Good says how much higher the average price of a diamond with a Good cut is than the average for a diamond with a Fair cut. All the other ordinal characteristics work the same way.

Are these metrics expected given the appropriate or relevant visualization(s) above? Can you tell from the visualizations?

**TO-DO # I don't feel like I can tell from the visualizations above how good a fit the linear model is. I think I'd need to see a 10-dimensional plot of our features against price, and see if the points were close to the hyperplane. Unfortunately, as a human and 3-4 dimensional being, I am limited in this capacity :(

Comment on why R^2 is high. Think theoretically about diamonds and what you know about them.

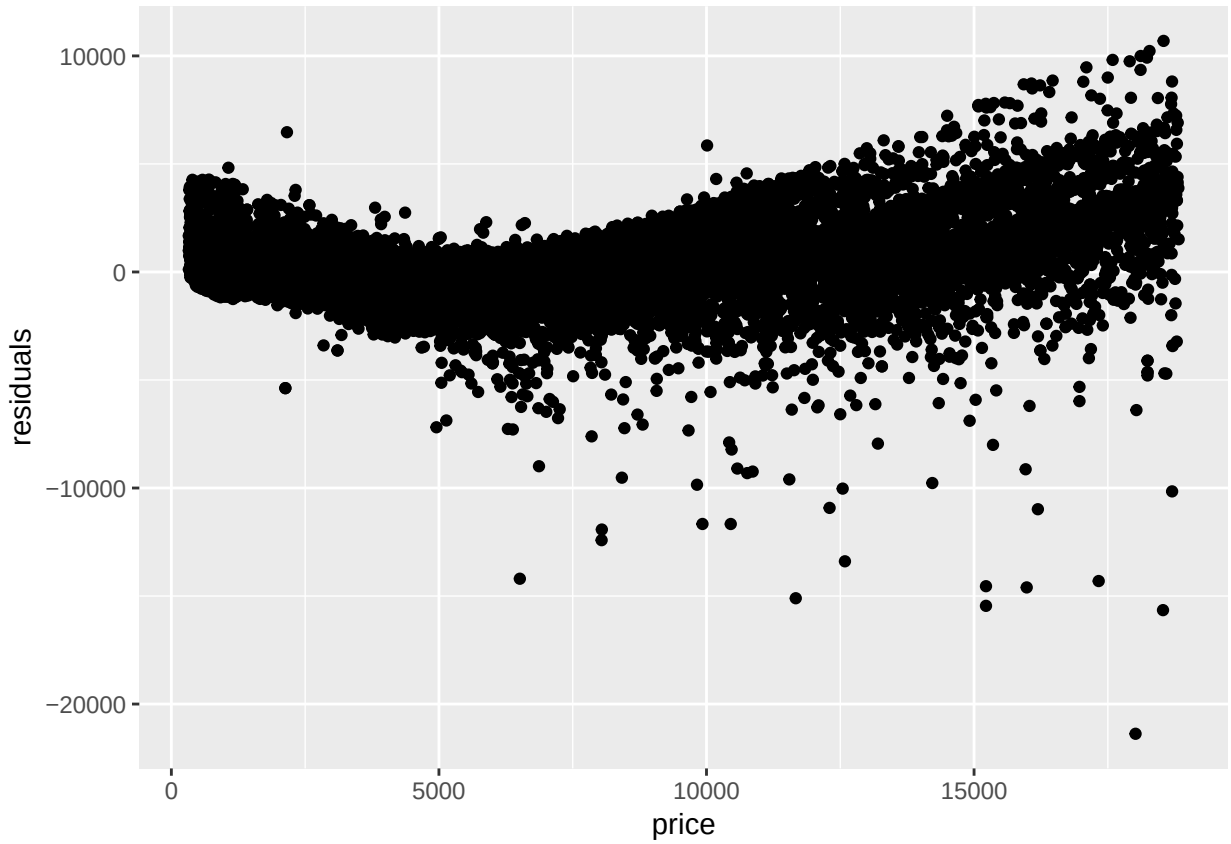
**TO-DO # Since diamonds are priced by people based on the characteristics described in our feature set, they should explain most of the variance in price.

Do you think you overfit? Comment on why or why not but do not do any numerical testing or coding.

**TO-DO # I don't think we've overfit because we only have 10 degrees of freedom explaining a dataset of ~50,000 observations.

Create a visualization that shows the "original residuals" (i.e. the prices minus the average price) and the model residuals.

```
#TO-DO
residuals = diamonds$price - predict(fullmod, diamonds)
ggplot(diamonds, aes(x = price, y = residuals)) +
     geom_point()
```


10000
0
−10000
−20000
residuals
0
5000
10000
15000
price