Exploratory Data Analysis (EDA) of diamonds dataset by Vivek Yadav, PhD

In this report I explore diamonds data set in ggplot2, and build a model to predict diamond price from characteristics of the diamond.

Univariate Analysis

Structure of the data

After loading data from the ggplot data set, I used str command to investigate structure of the data set. From output below, diamonds set has 10 columns of data for 53940 diamonds. The variables are,

1. Price: Price of the diamond in USD
2. Carat: Weight of the diamond in carats ranging from 0.2 to 5.01
3. Cut of the diamond, the levels are (Fair(worst), Good, Very Good, Premium, Ideal(best))
4. Color of the diamond, from J (worst) to D (best)
5. Clarity of the diamond (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
6. x,y,z: length, width and depth of the diamond.
7. depth of the diamond, a measure of aspect ratio. 2*z/(x+y), 43 and 79.
8. width of the diamond, width of the top of the diamond relative to widest point (43-95)

Main Variables

```
## 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 ...
```

As we are interested in predicting the price of the diamond, the main variables investigated here are price, carat, cut and clarity.

Price

Histogram of price for all the diamonds. Surprisingly, there is no diamond whose price lies between 1460 and 1530. There seems to be multiple peaks for prices. Especially, there is a peak around 1000, and another near 4000. These numbers may indicate budgets that individuals are most comforatable spending. There may be other factors that could infludence price of the diamond.

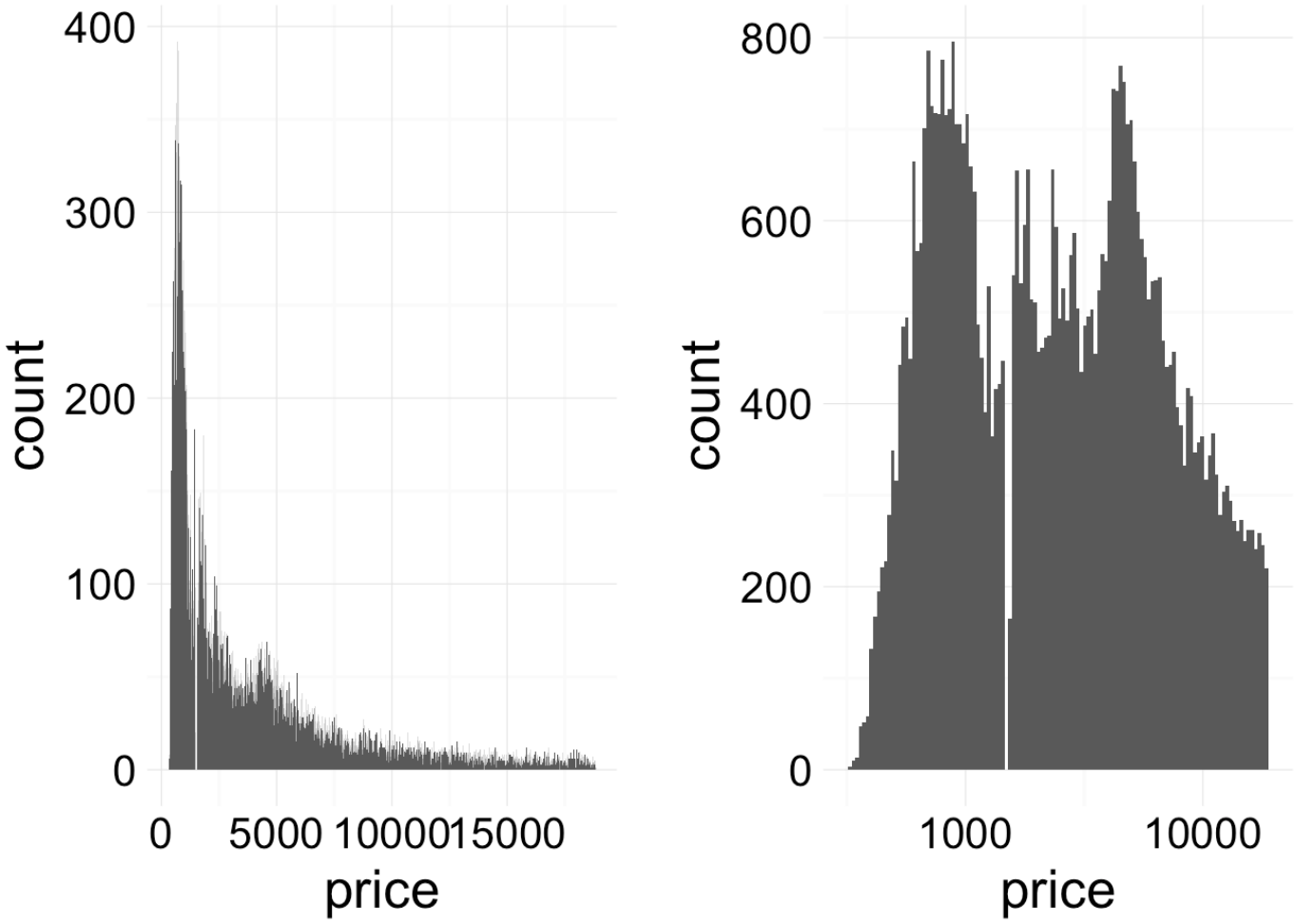


Fig 1. Histogram of Price and Log(price) for the diamonds.

Carat

Histogram of carat for all the diamonds. The distributions seems very unusual. This is not surprising, given that most diamonds people buy are precut into standard sizes. For example, most diamonds are expected to come in some fixed sizes, like .5, 1, 1.5, 2 etc.

Fig 2. Histogram of carat.

A new varaible “carat.cut” is made to accomodate the fact that diamonds are cut in a way to have pre-decided prefered weights. Also, the distributions of cuts is skewed to right because customors may consider a diamonds as a 2 carat diamond only if its above 2 carat, and might reject a 1.98 carat diamond because it weights less than 2. Therefore, the manufacturuer may devise their strategy in such a way that the diamonds cuts are all above some predefined numbers. These values were chosen as, c(0,.29,0.39,.49,.69,.89,.99,1.19,1.49,1.69,1.99,6).

```
# Creating new variable carat.cut
carat_prefered = c(0,.29,0.39,.49,.69,.89,.99,1.19,1.49,1.69,1.99,6)
diamonds$carat.cut = cut(diamonds$carat, breaks = carat_prefered)
lvls_diamonds_caratcut =c("(0,0.29]","(0.29,0.39]","(0.39,0.49]","(0.49,0.69]", "
(0.69,0.89]","(0.89,0.99]","(0.99,1.19]","(1.19,1.49]","(1.49,1.69]","(1.69,1.9
9]","(1.99,6]")
carat.cut = factor(diamonds$carat.cut,levels = levels(diamonds$carat.cut),ordere
d=TRUE)
```

After making the “carat.cut” variable, the histogram is plotted again with color indicating the region of the prefered carat cut. Another variable, DistPrefered is created to quantify deviation from the desired cut. This variable was computed as the minimum difference between the carat value and the largest prefered carat size smaller than the actual value. For example, for a carat of 2.5, the largest value of prefered size below 2.5 is 1.99, and the distance is calculated as 2.5-1.99 = 0.51.

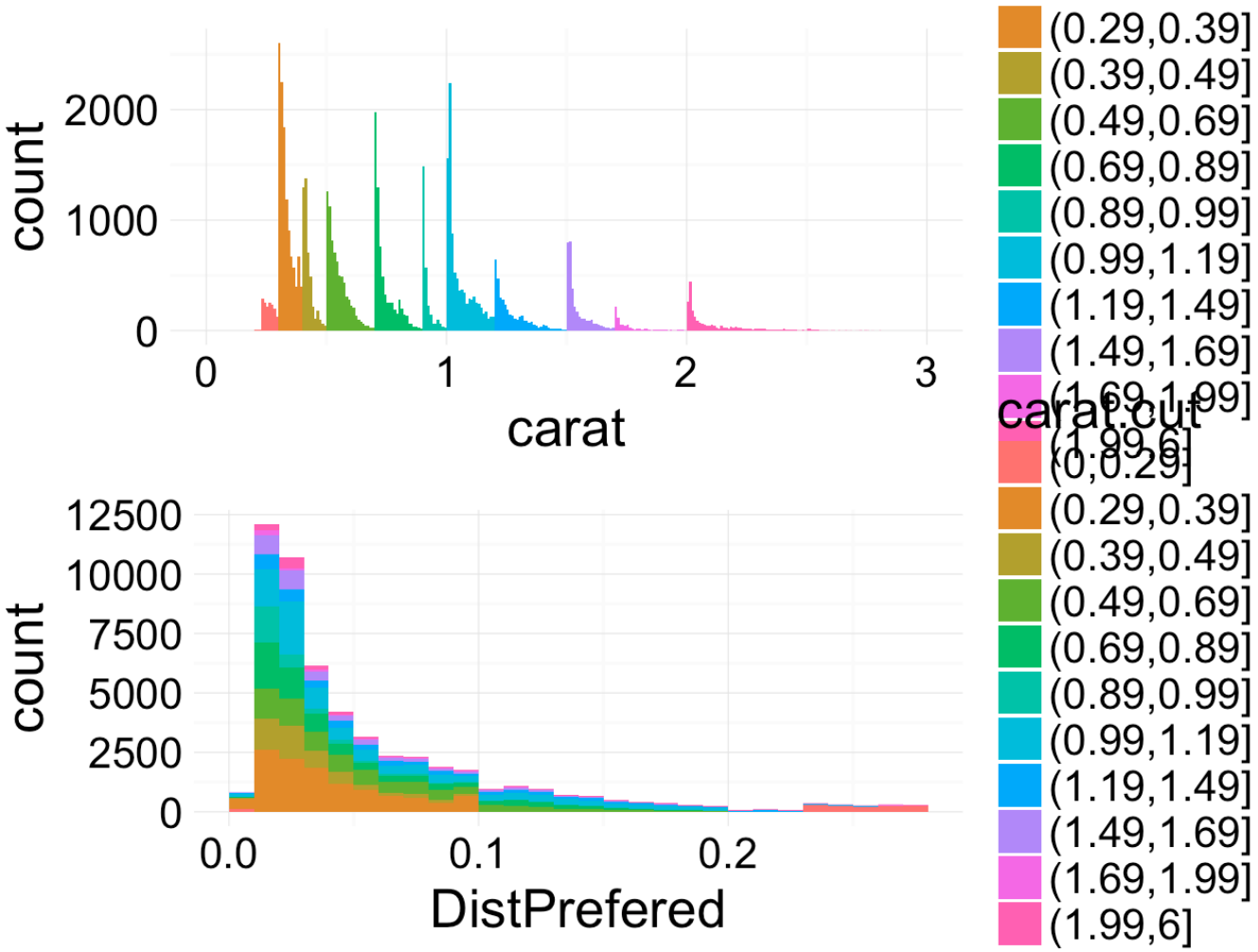


Fig 3. Histogram of Carat and deviation from prefered size.

Depth

Histogram of depth for all the diamonds. The distributions seems well behaved. Depth variable seems to have unimodal symmetric distribution.

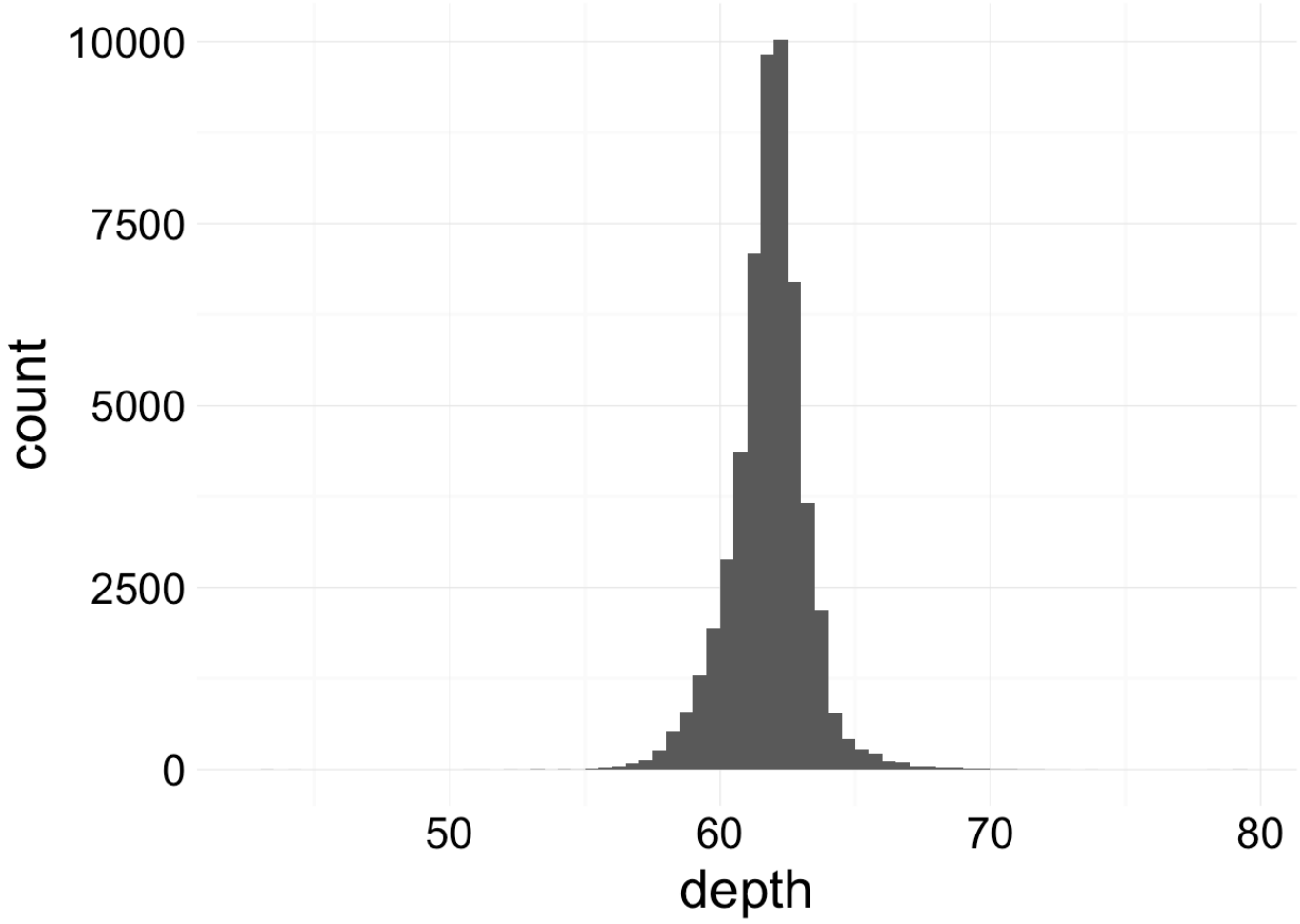


Fig 4. Histogram of Depth

TABLE

Histogram of table for all the diamonds. The distributions seems very odd. Although not all values are integers, 53016 of 53940 are integer values. This variable may be a variable that is used to characterize the shape of the diamond, and may be related to the cut of the diamond. I therefore divided the data into smaller regions based on table size, and used cut function to create factors. I created 2 variables one with finer spacing and other coarse spacing. I created these 2 variables to later check if there are any trends in prices that may be predicted by the table.

```
##
## FALSE  TRUE
##   924 53016
```

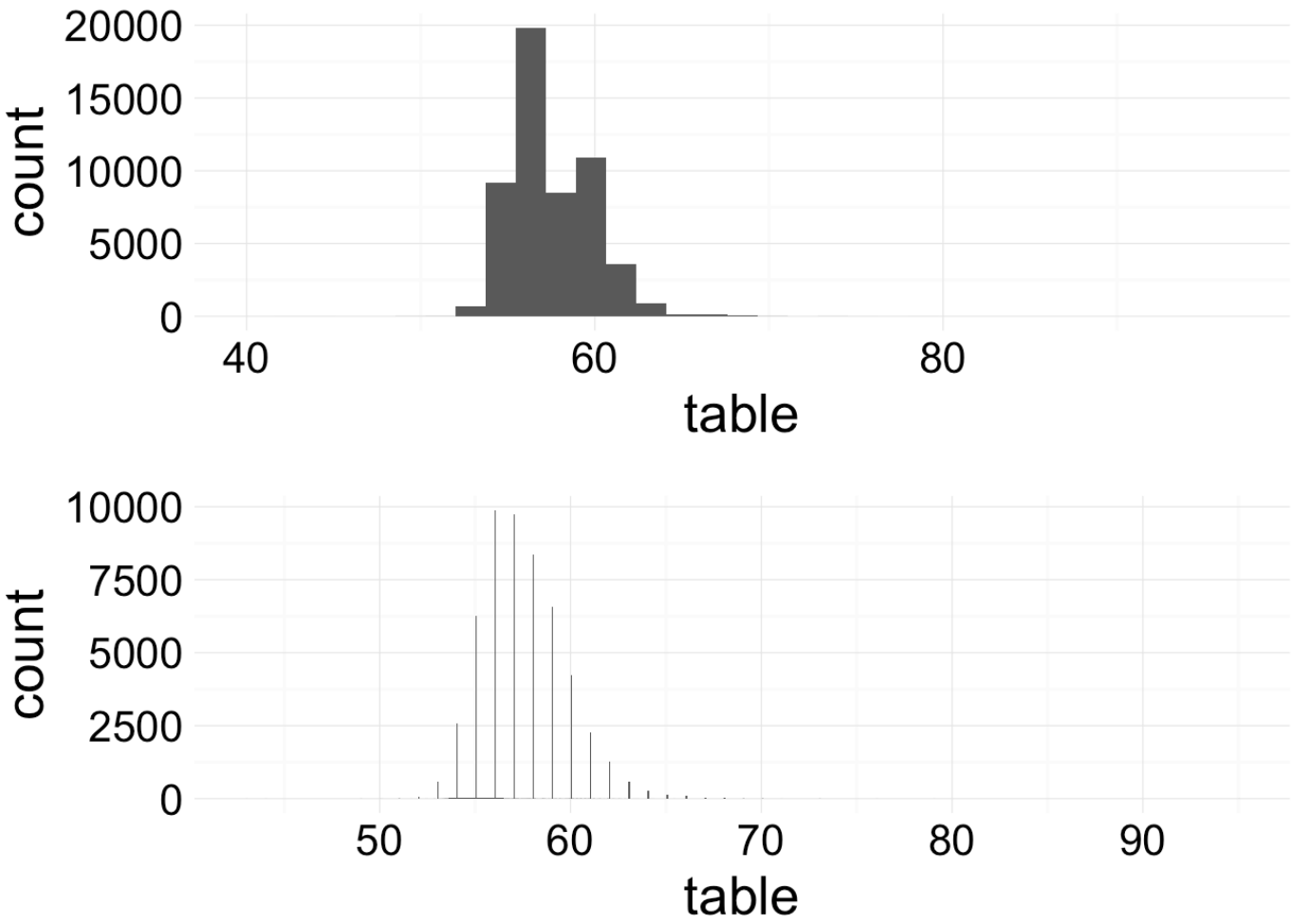


Fig 5. Histogram of Table

```
# Creating 2 variables for table/

table_cut_limits = c(0,54.5,55.5 ,56.5,57.5,58.5,59.5,60.5,61.5,100)
table_cut_limits2 = c(0,55.5 ,57.5,59.5,100)
diamonds$table.cut = cut(diamonds$table, breaks = table_cut_limits)
diamonds$table.cut2 = cut(diamonds$table, breaks = table_cut_limits2)

diamonds$table.cut = factor(diamonds$table.cut,levels = levels(diamonds$table.cu
t),ordered=FALSE)
diamonds$table.cut2 = factor(diamonds$table.cut2,levels = levels(diamonds$table.cu
t2),ordered=FALSE)
```

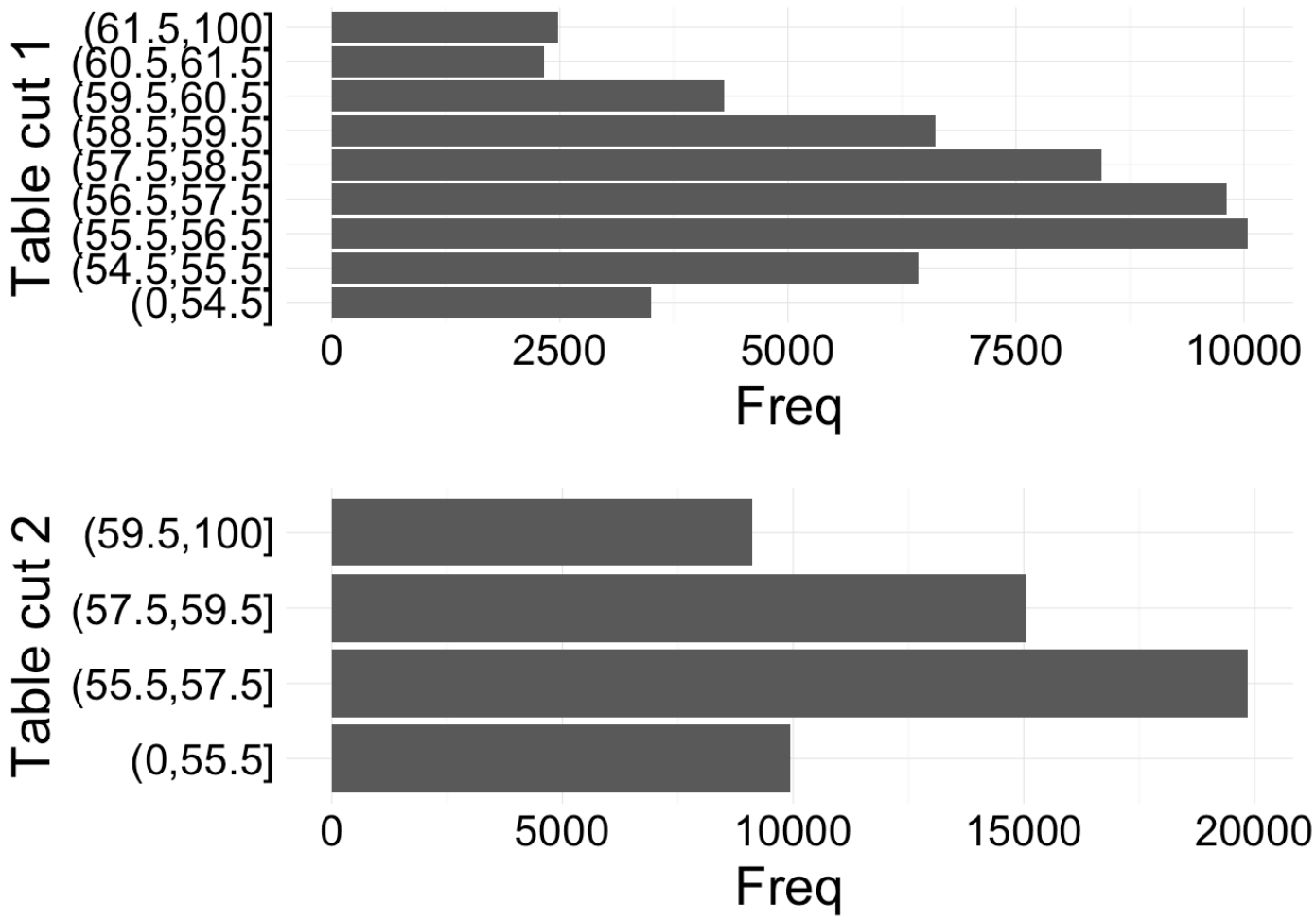


Fig 6. Counts for table.cut and table.cut2

X, Y and Z

Histogram of histograms for x,y and z are below. Based on these histograms, the dimensions seem to be well behaved. There were some values that were 0s. This information is not used for further analysis because when one goes to purchase diamonds, they do not determine the quality (hence the value) of the diamond based on x,y and z dimensions. These may be used to compute volume, however, without knowng the exact shape, x,y and z only define a bounding box around the diamond. Further, any volume information is already available in carats. Because 1 carat is about 2 grams, and for fixed density of the diamond, the volume has a linear relation with the weight in carat.

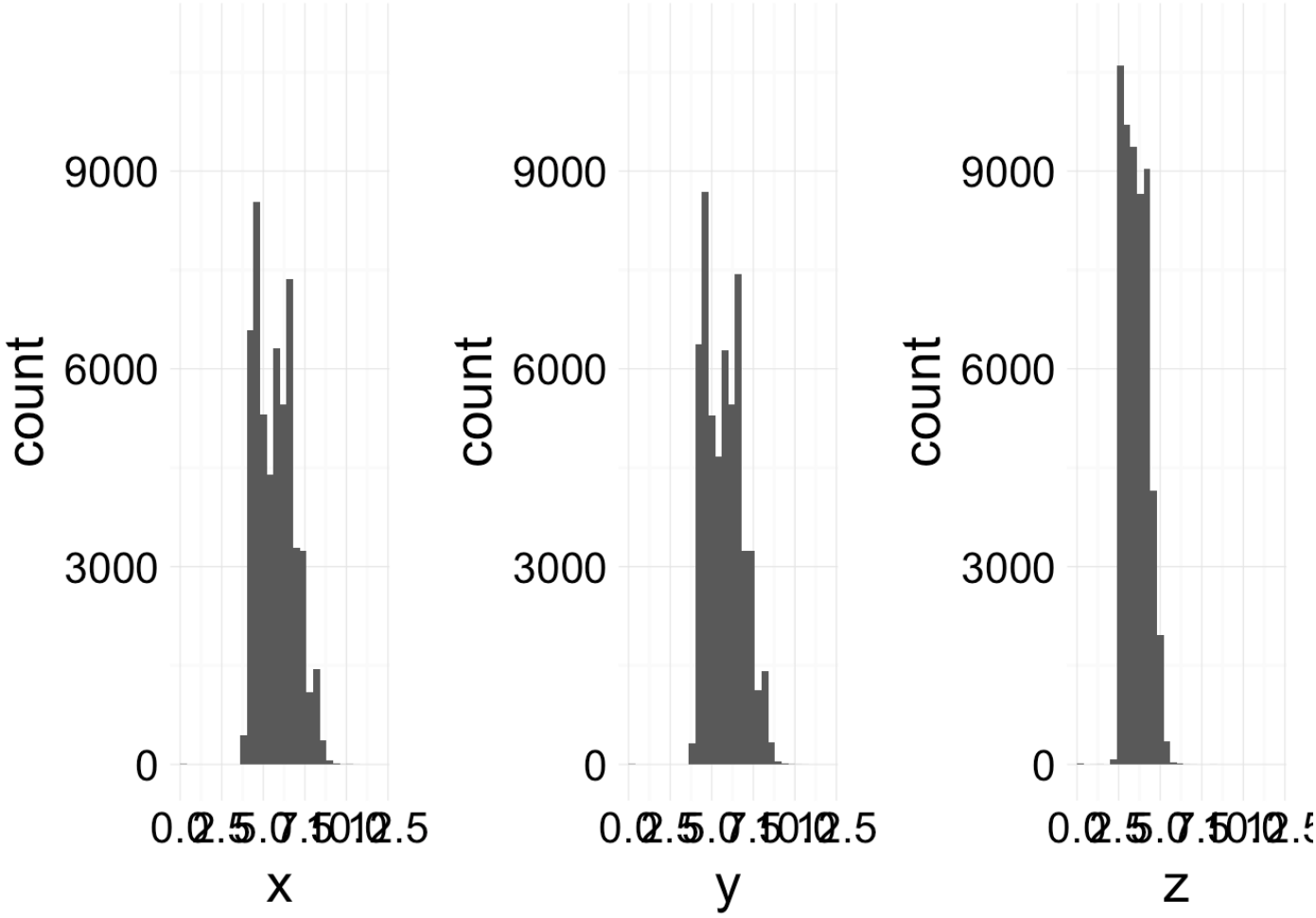


Fig 7. Histogram of X,Y and Z.

Cut, clarity and color.

As these varaiables are factors, a histogram is not generated. However, bargraphs with total counts of diamonds for each color, clarity and cut are plotted. Further, as these variables are ordered factors, and its expected that clearer diamonds will fetch higher prices, the factors are ordered. To make the distinction that these are ordered factors, the counts are plotted as horizontal bars. Most of the diamonds are premium or better cuts. For color and clarity, the distribution is slightly more symmetric than for cut.

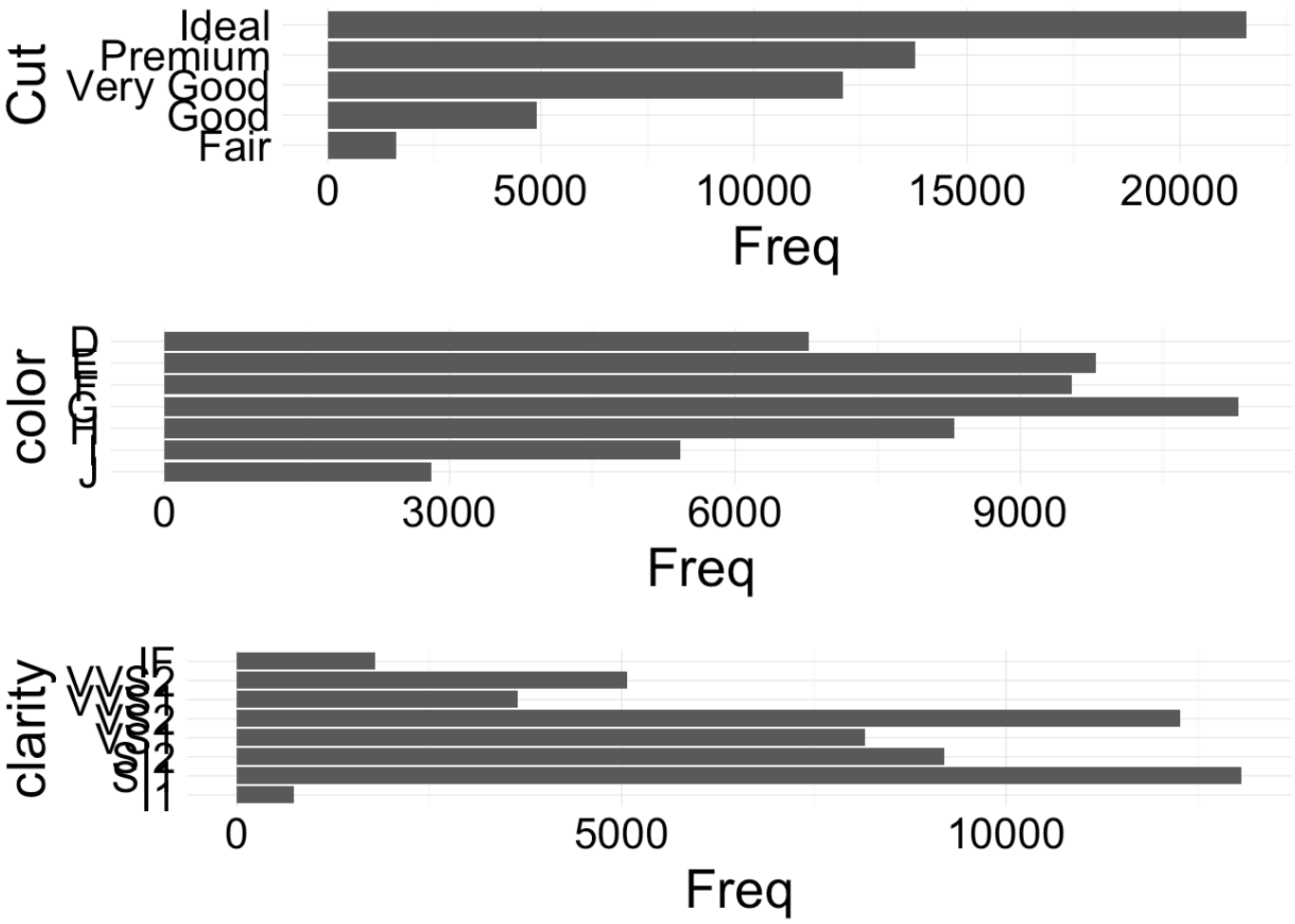


Fig 8. Counts for cut, clarity and color

Univariate Analysis

What is the structure of your dataset?

Diamonds set has 10 columns of data for 53940 diamonds. The variables are,

1. Price: Price of the diamond in USD
2. Carat: Weight of the diamond in carats ranging from 0.2 to 5.01
3. Cut of the diamond, the levels are (Fair(worst), Good, Very Good, Premium, Ideal(best))
4. Color of the diamond, from J (worst) to D (best)
5. Clarity of the diamond (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
6. x,y,z: length, width and depth of the diamond.
7. depth of the diamond, a measure of aspect ratio. 2*z/(x+y), 43 and 79.
8. width of the diamond, width of the top of the diamond relative to widest point (43-95)

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  14 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 "J"<"I"<"H"<"G"<..: 6 6 6 2 1 1 2 3 6 3
...
##  $ clarity     : Ord.factor w/ 8 levels "I1"<"SI1"<"SI2"<..: 3 2 4 5 3 7 6 2 5
4 ...
##  $ 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 ...
##  $ carat.cut   : Factor w/ 11 levels "(0,0.29]","(0.29,0.39]",..: 1 1 1 1 2 1 1
1 1 1 ...
##  $ DistPrefered: num  0.23 0.21 0.23 0 0.02 0.24 0.24 0.26 0.22 0.23 ...
##  $ table.cut   : Factor w/ 9 levels "(0,54.5]","(54.5,55.5]",..: 2 8 9 5 5 4 4
2 8 8 ...
##  $ table.cut2  : Factor w/ 4 levels "(0,55.5]","(55.5,57.5]",..: 1 4 4 3 3 2 2
1 4 4 ...
```

```
##      carat               cut        color        clarity
##  Min.   :0.2000   Fair     : 1610   J: 2808   SI1    :13065
##  1st Qu.:0.4000   Good     : 4906   I: 5422   VS2    :12258
##  Median :0.7000   Very Good:12082   H: 8304   SI2    : 9194
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171
##  3rd Qu.:1.0400   Ideal    :21551   F: 9542   VVS2   : 5066
##  Max.   :5.0100                     E: 9797   VVS1   : 3655
##                                     D: 6775   (Other): 2531
##      depth           table           price             carat.cut
##  Min.   :43.00   Min.   :43.00   Min.   :  326   (0.29,0.39]:11493
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   (0.99,1.19]: 9260
##  Median :61.80   Median :57.00   Median : 2401   (0.49,0.69]: 7507
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   (0.69,0.89]: 6936
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   (0.39,0.49]: 4582
##  Max.   :79.00   Max.   :95.00   Max.   :18823   (1.19,1.49]: 3565
##                                                  (Other)    :10597
##   DistPrefered           table.cut
##  Min.   :0.00000   (55.5,56.5]:10037
##  1st Qu.:0.02000   (56.5,57.5]: 9809
##  Median :0.03000   (57.5,58.5]: 8437
##  Mean   :0.05457   (58.5,59.5]: 6615
##  3rd Qu.:0.07000   (54.5,55.5]: 6431
##  Max.   :3.02000   (59.5,60.5]: 4306
##                    (Other)    : 8305
```

What is/are the main feature(s) of interest in your dataset?

The purpose of this analysis is to predict price. Therefore, most important feature of interest is price. Other features that are of interest are carat, cut, clarity and color.

What other features in the dataset do you think will help support your investigation into your feature(s) of interest?

Carat, color, cut and clarity are most likely going to be the variables that will be used to predict the price of the diamond. Depth and table may be useful in quantifying shape of the diamonds.

Did you create any new variables from existing variables in the dataset?

I created 4 new variables.

- Carat.cut: I saw that carats for the diamonds were clustered around some fixed numbers.This is not surprising because most diamonds are precut, and manufactures may produce diamonds of weight (carats) increasing in predefined increments. Another thing I noticed is that for all the diamonds, the distribution of carats were switched to the right of the prefered numbers. This is understandable because a customer who is looking to buy a 2 carat diamond is more likely to purchase a 2.1 carat diamond instead of 1.9. I therefore created 2 variables. carat.cut and DistPrefered. Carat.cut is a ordered factor that indicated the prefered weight of the diamond.
- DistPrefered quantifies the difference between actual weight in carats and the prefered size of the diamond.
- table.cut and table.cut are finer and coarse factors that quantify the range of diamond's table value.

Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

Diamond's carat values showed very unusual trend, where there were a large number of diamonds for certain predefined values of carats. This is not surprising because most diamonds are precut, and manufactures may produce diamonds of weight (carats) increasing in predefined increments. Another thing I noticed is that for all the diamonds, the distribution of carats were switched to the right of the prefered numbers. This is understandable because a customer who is looking to buy a 2 carat diamond is more likely to purchase a 2.1 carat diamond instead of 1.9. I therefore created 2 variables, carat.cut a factor indicating the prefered size of the diamond, and Distprefered indicating error from the predefined value. This manipulation significantly removed the discontinuties in the histogram of carat.

Another thing I noticed is that the diamond's table values were all integers. Further, it may be possible that this value is a proxy for the shape of the diamond. Therefore, I created 2 variables, table.cut and table.cut2. These 2 factors represent discretization of the table range into a finer and coarser regions.

Further, I ordered cut, clarity, color, and carat.cut. Because for these variables, a better indicator of diamond is expected to have higher price.

I also removed x,y and z from the diamonds data set because, these do not provide a direct measure of diamond's price. These may be used to compute volume, however, without knowng the exact shape, x,y and z only define a bounding box around the diamond. Further, any volume information is already available in carats. Because 1 carat is about 2 grams, and for fixed density of the diamond, the volume has a linear relation with the weight in carat. Therefore, I dont use x,y and z values in further analysis.

Bivariate Plots Section

Panels plot to observe overall patterns.

The strongest predictor of price seems to be carat. In all the 3 variables that are related to carat (carat, carat.cut and Distprefered), the correlation between price and carat is high. Higher carat gets higher price.

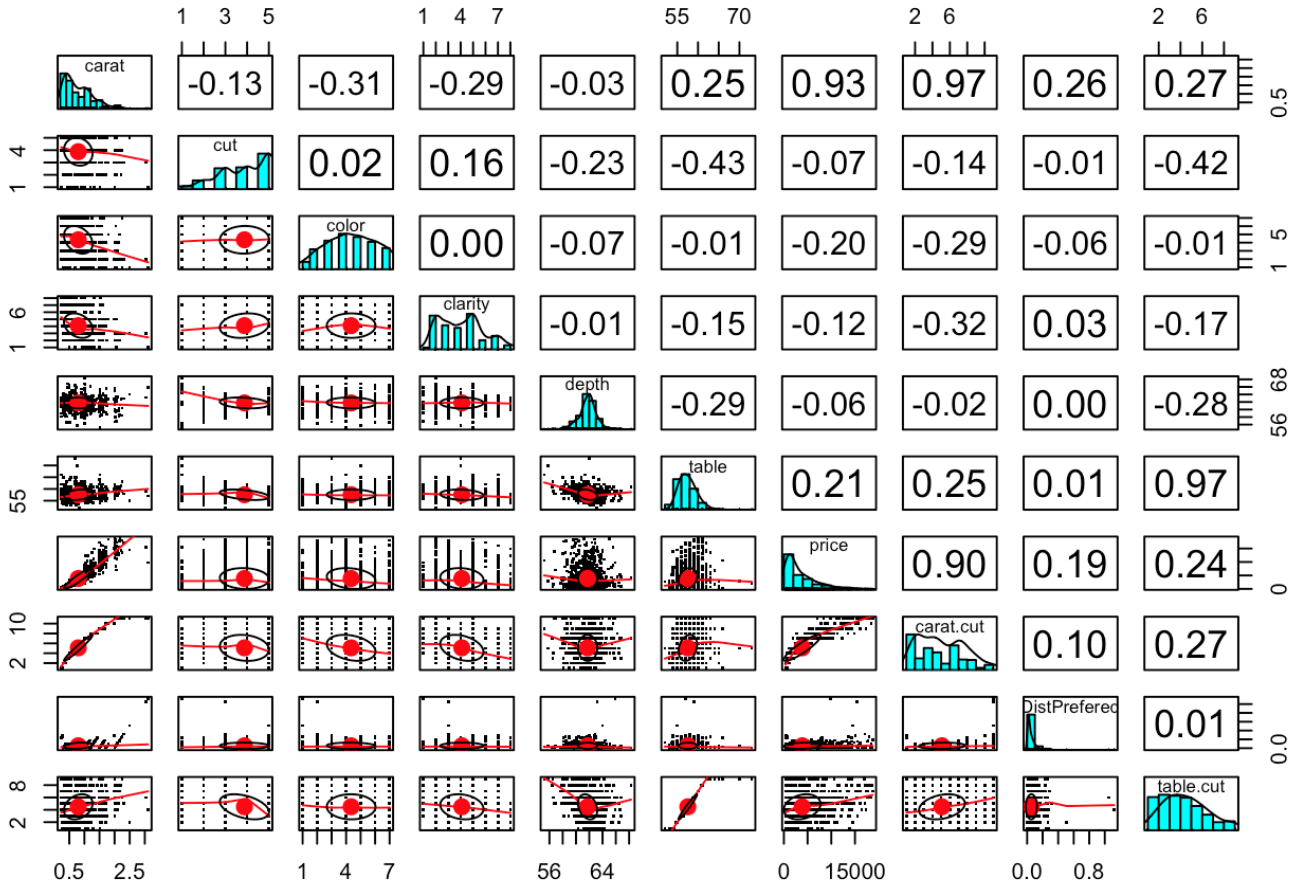


Fig 9. Panel plots for correlation between variables

Price vs Carat variables

As carat increases, the price increases. The relation between price and carat seems to be nonlinear. This may be because it is more difficult to find a larger diamond with minimal to no faults. This pattern is expecially clear after applying log to the price.

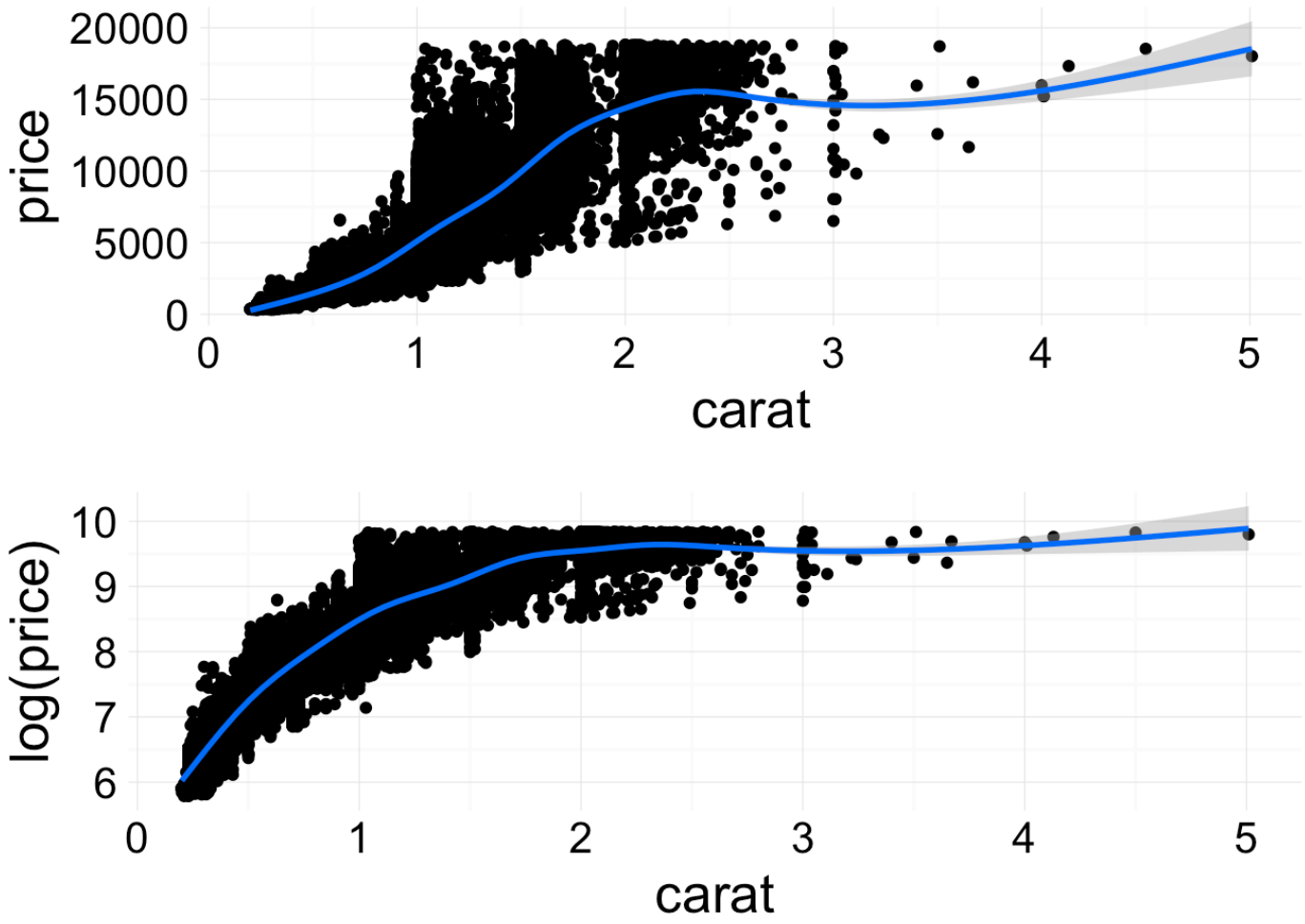


Fig 10. Price vs carat

Two other variables that were created were carat.cut and DistPrefered. Plots below show distribution of price with carat and price variations with value above the prefered value. The second plot is colorcoded with carat.cut because diamonds with higher carat are expected to be pricier.

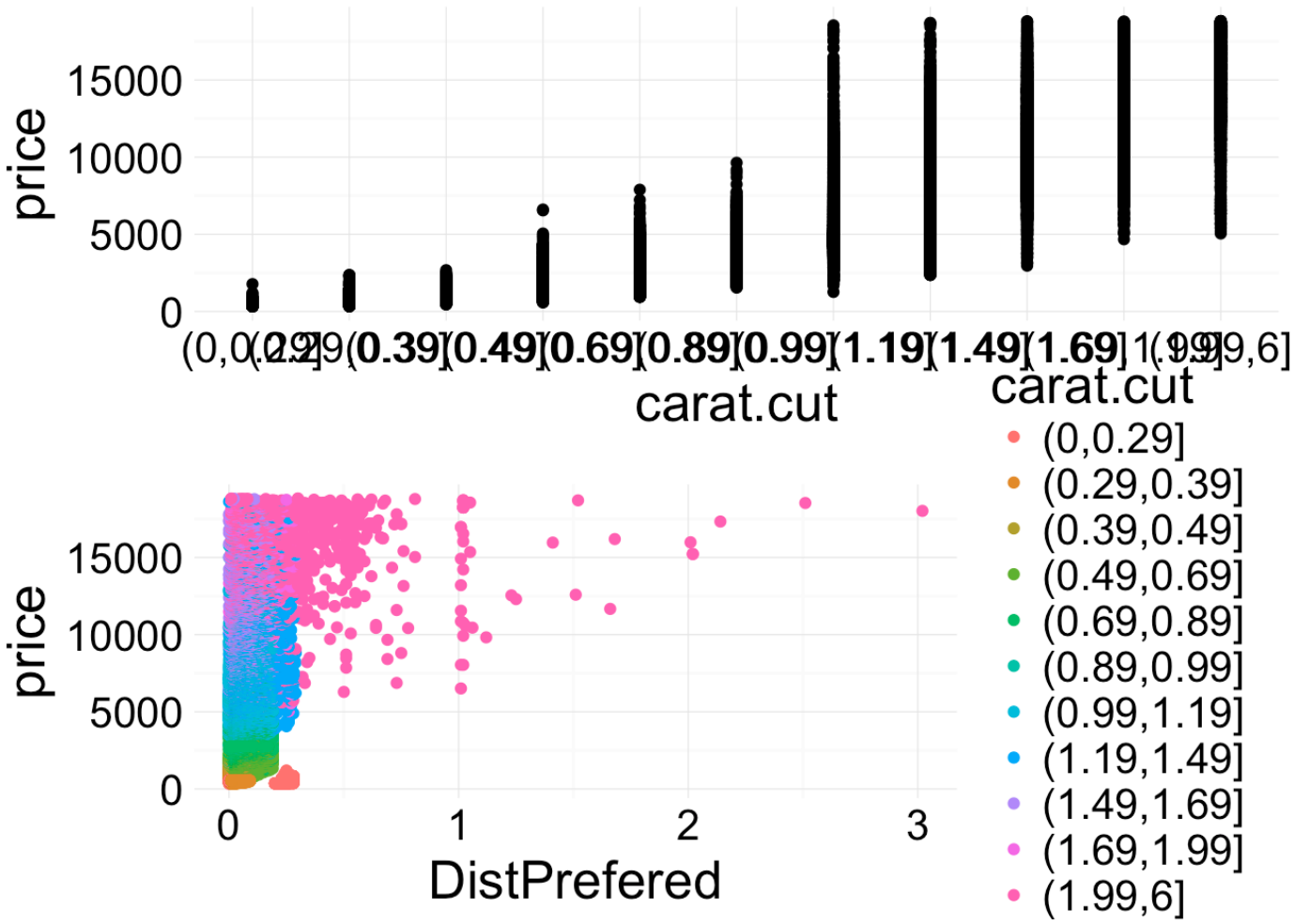


Fig 11. Price vs carat.cut

Price vs Cut

I first checked the price vs cut patterns. It appears that the trends are bimodal. This may be affected by the prefered sizes of diamonds. I next plot log of price vs cut. We jitter the points toget a sense of distribution and plot box plots also. These plots indicate that cut may not be a strong predictor of diamond price. The red line is mean price.

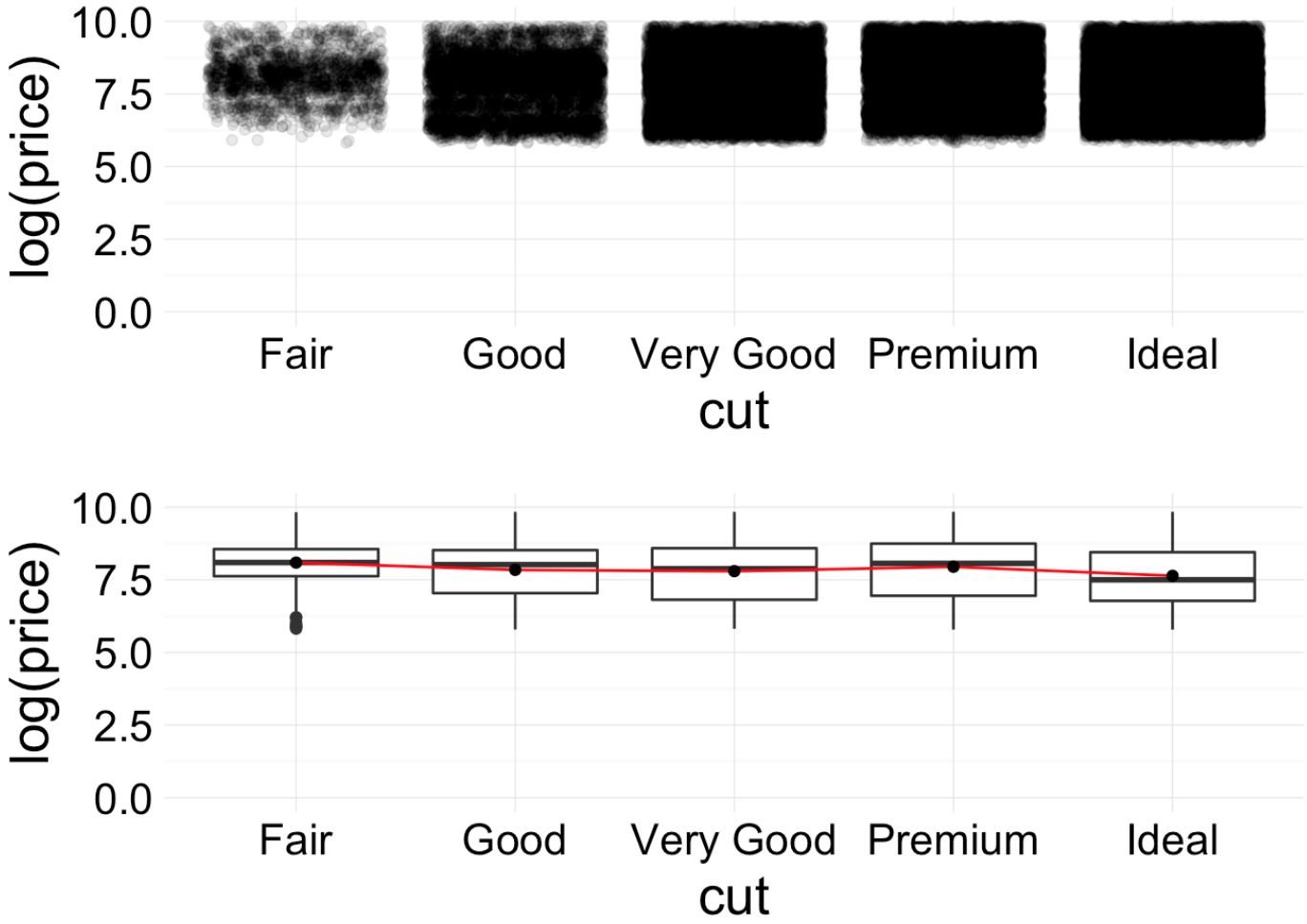


Fig 12. Price vs cut

Diamonds with ideal or premium cut seemed to have lower prices. This may be due to difference in the number of diamonds with better cuts.

Price vs Clarity

We next plot log of price vs Clarity. We jitter the points toget a sense of distribution and plot box plots also. These plots indicate that price varies invesrly with clarity This is surprising because we expect poorer clarity diamonds to be more expensive. However, the skewness in the plots may be due to differences in number of diamonds in each group. We will further investigate this relation in multivariate plots section. The red line is mean price.

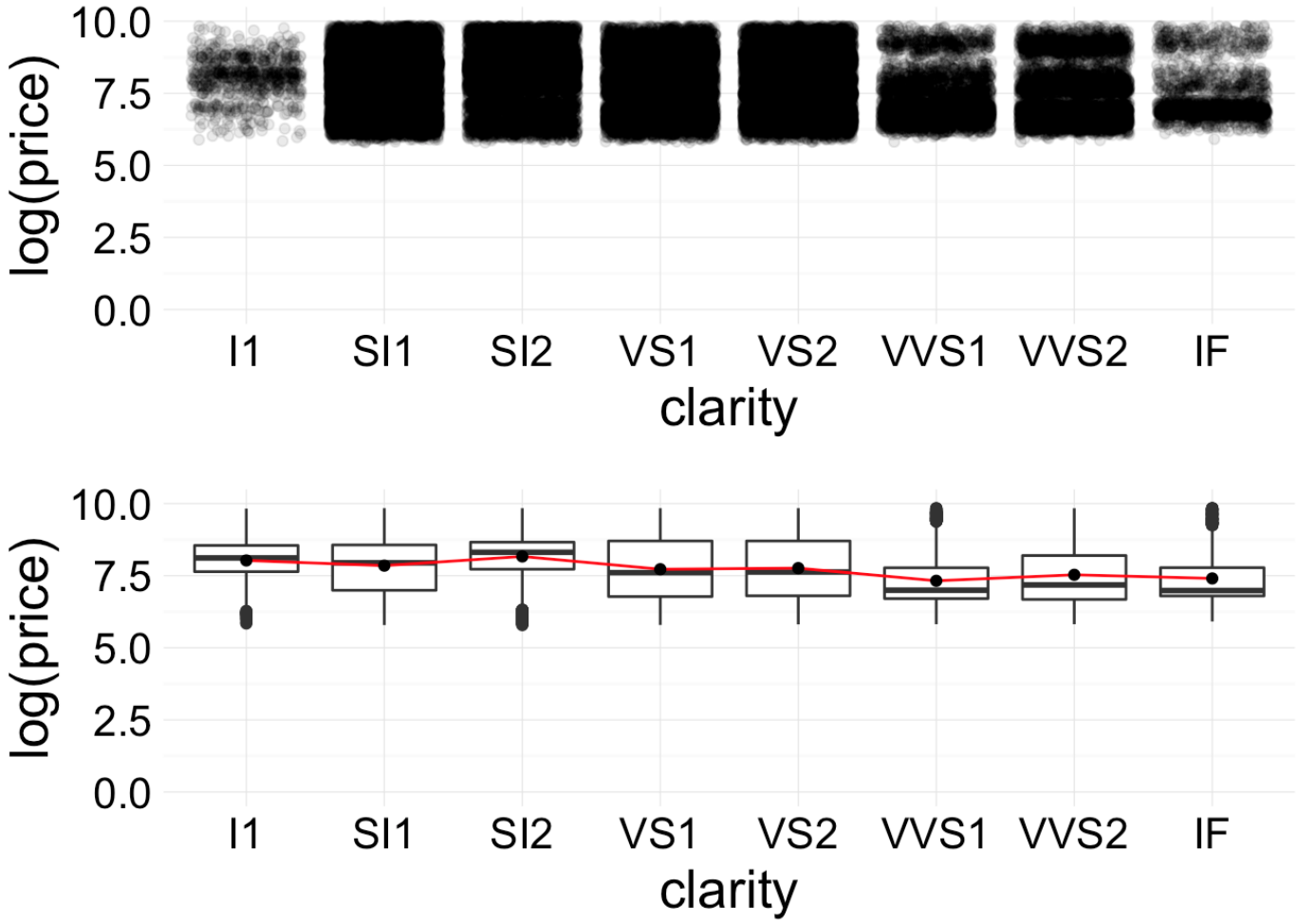


Fig 13. Price vs clarity

Price vs Color

We next plot log of price vs color. We jitter the points toget a sense of distribution and plot box plots also. These plots indicate that price varies proporionately with the quality of color. The red line is mean price.

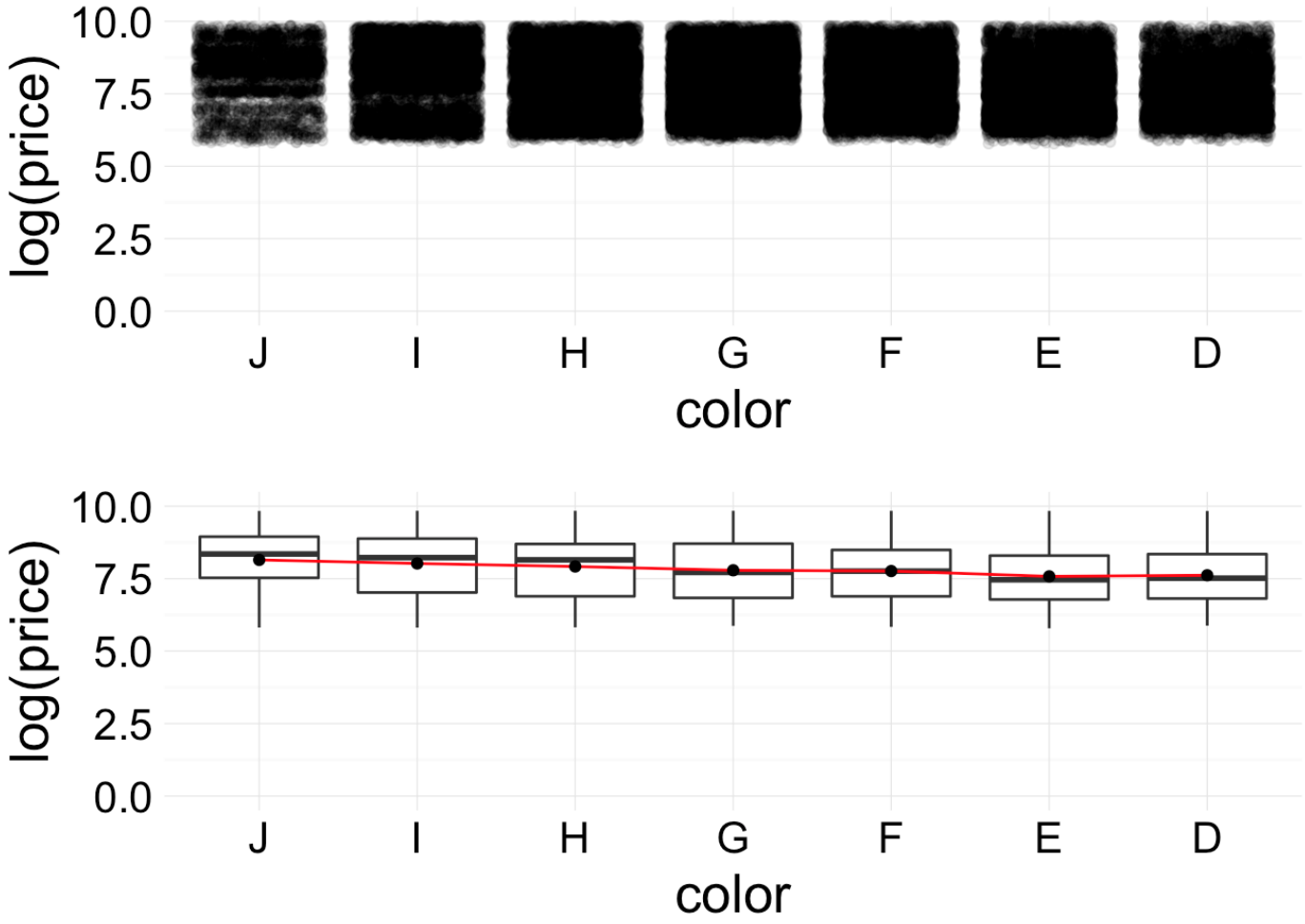


Fig 14. Price vs color

Price vs Depth

We next plot log of price vs depth. We jitter the points toget a sense of distribution and plot box plots also. These plots indicate that price varies proporionately with the quality of color. Depth seems to have no effect on the price, so this variable will mostly not be included in model building.

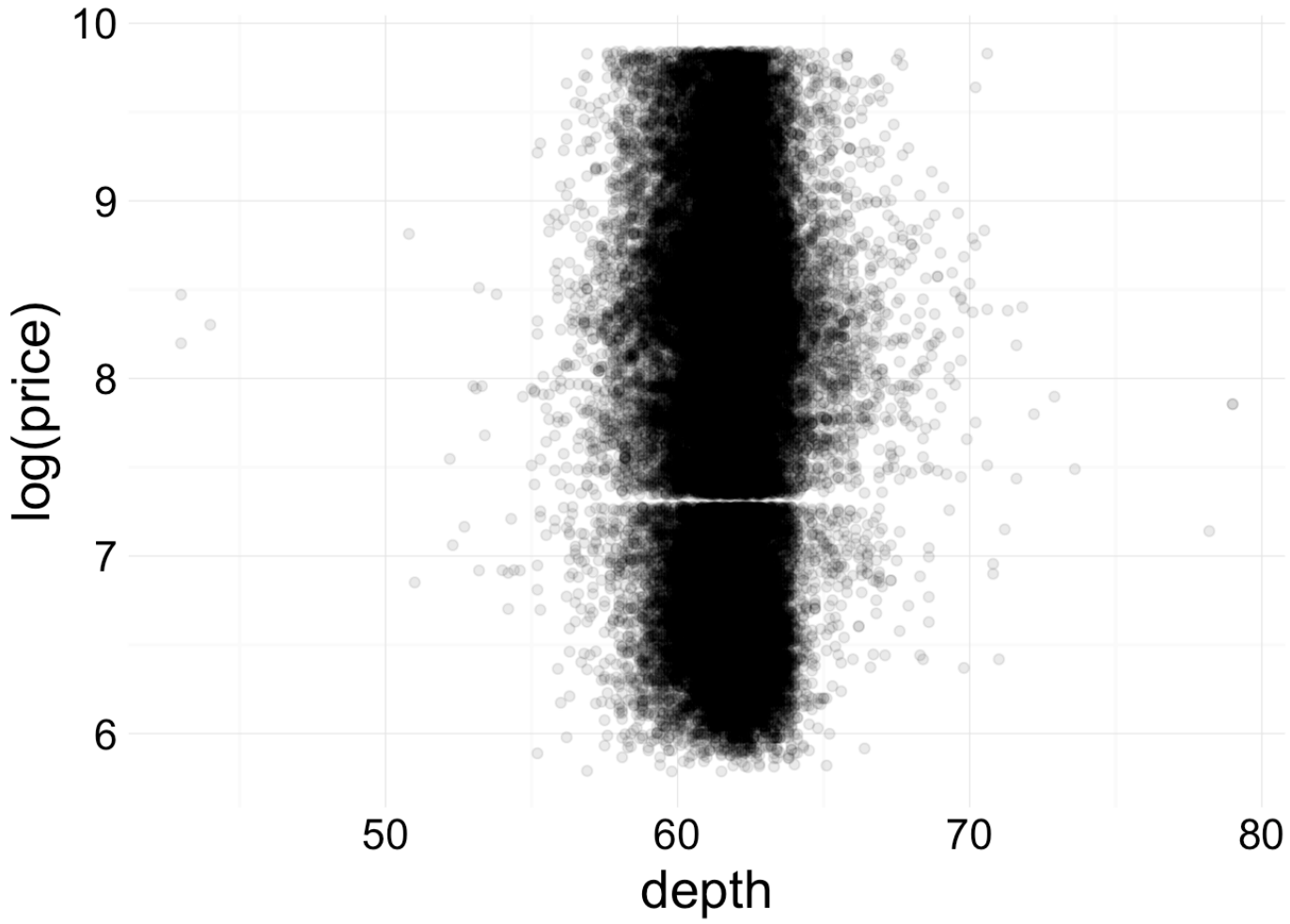


Fig 15. Price vs depth

Price vs Table

We next plot log of price vs table. From plot below the trends are not clear. However, table is a shape factor, and is expected to be different for different diamonds. We therefore made 2 variables table.cut and table.cut2, to quantify shape.

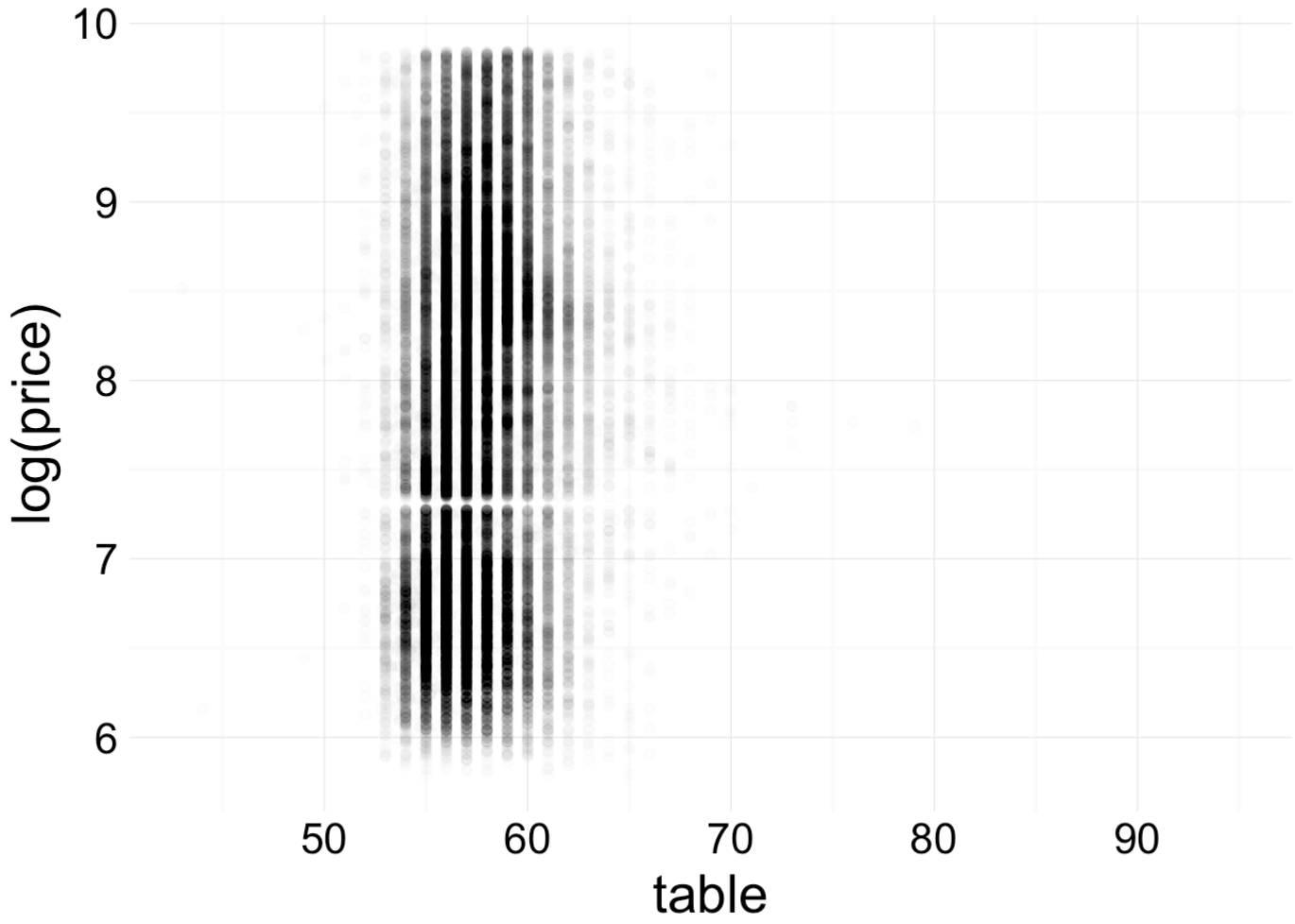


Fig 16. Price vs Table

We therefore plot table.cut and price. We jitter the points toget a sense of distribution and plot box plots also. These plots indicate that price varies proporionately with the quality of color. Depth seems to have no effect on the price, so this variable will mostly not be included in model building.

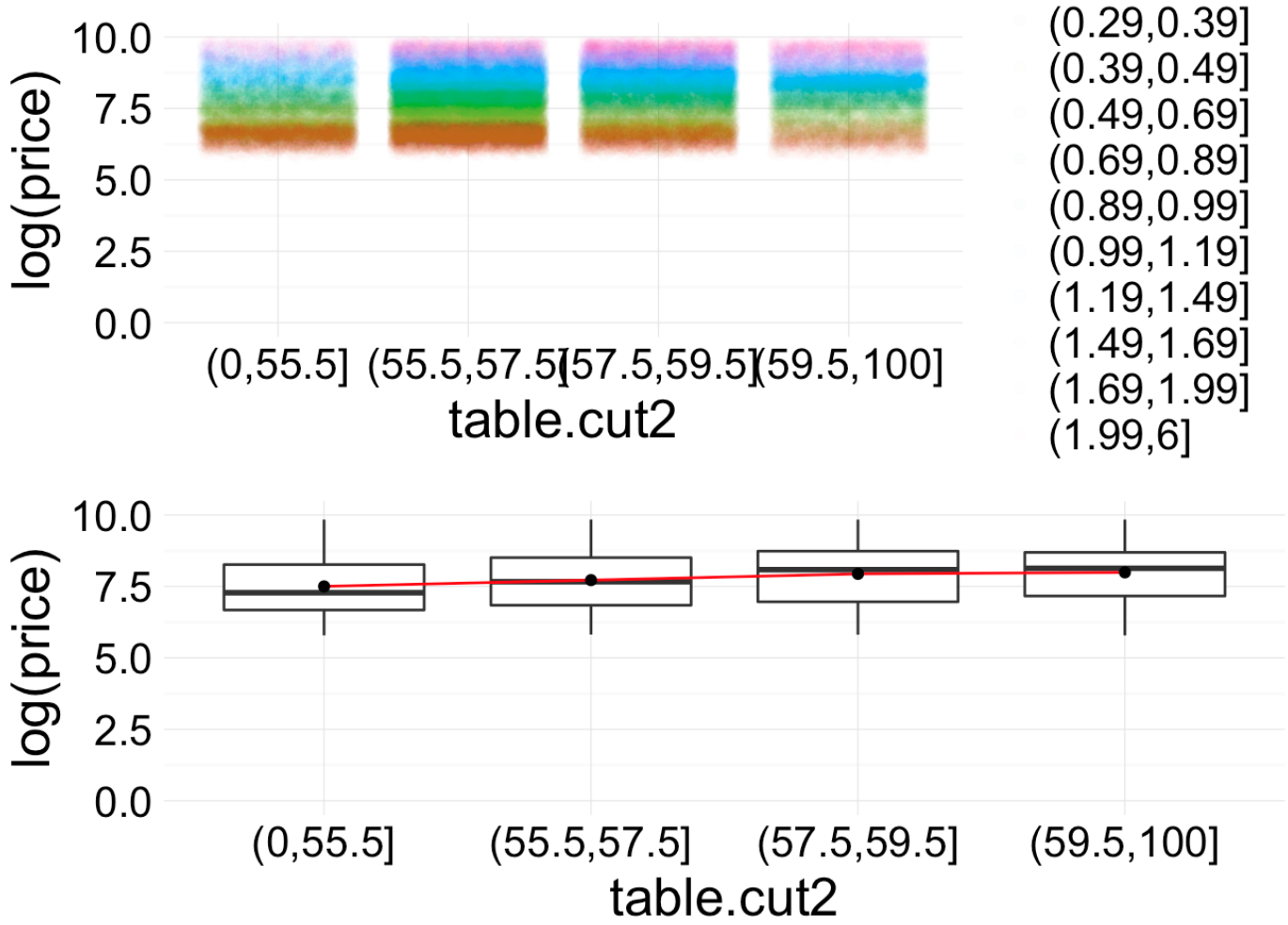


Fig 17. Price vs Table.cut

We next compare a finer and coarse discretization of Table. It appear that the finer gradation may be better suited to predict price of the diamonds.

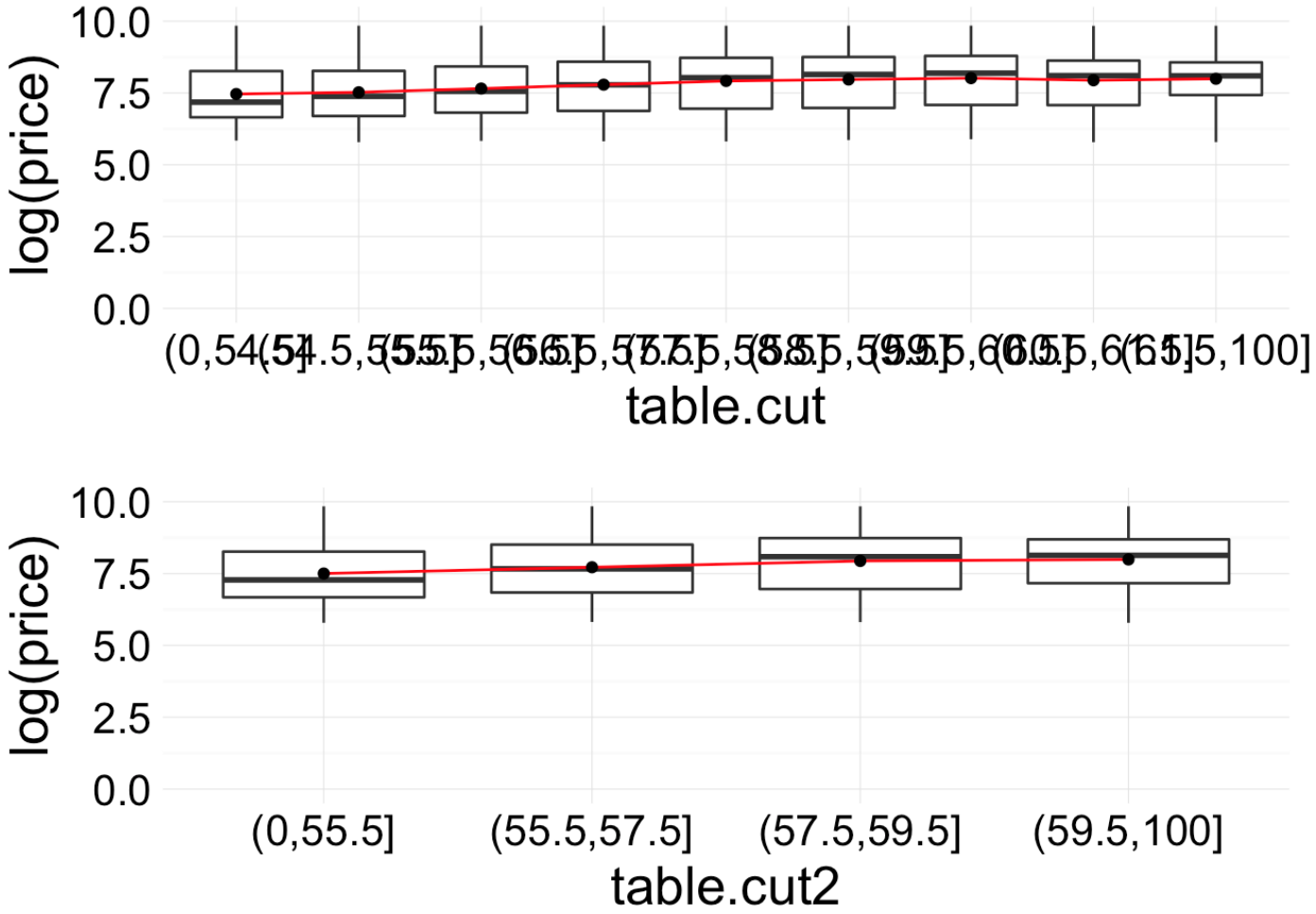


Fig 18. Price vs Table.cut and table.cut2

Bivariate Analysis

Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

- The main feature I found that predicted price the best was carat. All variables related to carat, carat, carat.cut correlated well with price.
- ANother variable that I didnt expect to predict price as well as it did was table. Table is a shape factor, so I discretized it into a finer and coarse bins. After discretization, clear patterns in diamond's price and table emerged. A larger value indicates that the diamond has larger face.
- Cut and clarity seemed to have minimal effect on price. For cut, this may be due to disproportionate number of diamonds that have premium or ideal cuts.
- Price showed increase with color quality (high for D).
- Price had biomodal variation with cut, however, this may be due to differences in the number of diamonds with different carat (or prefered carat) size.

Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

Depth did not change with price. This may be because manufactures make diamonds in some fixed shape, and depth factor may represent shapes for all types of diamonds i.e. its not a determinant of price.

What was the strongest relationship you found?

Strongest realtion was between carat and price. The second strongest relation appeared between price and table. Table may be indicator of the shape of the diamond.

Multivariate Plots Section

In this section, I further explore the relation between price and other variables. I will use analysis from this part to decide variables in the final model building.

Price vs Carat variables

I created 2 additional variables from carat, carat.cut indicating prefered carat number, and DistPrefered indicating deviation from prefered diamond size. Overall, it appears that the greater deviation from prefered size is associated with higher costs. Interesting distribution of DistPrefered is in 0 to .29 range. This is due to the discretization scheme I used. I did not split this into two data sets, because the number of points were very low.

```
## Warning: Removed 318 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 318 rows containing missing values (geom_point).
```

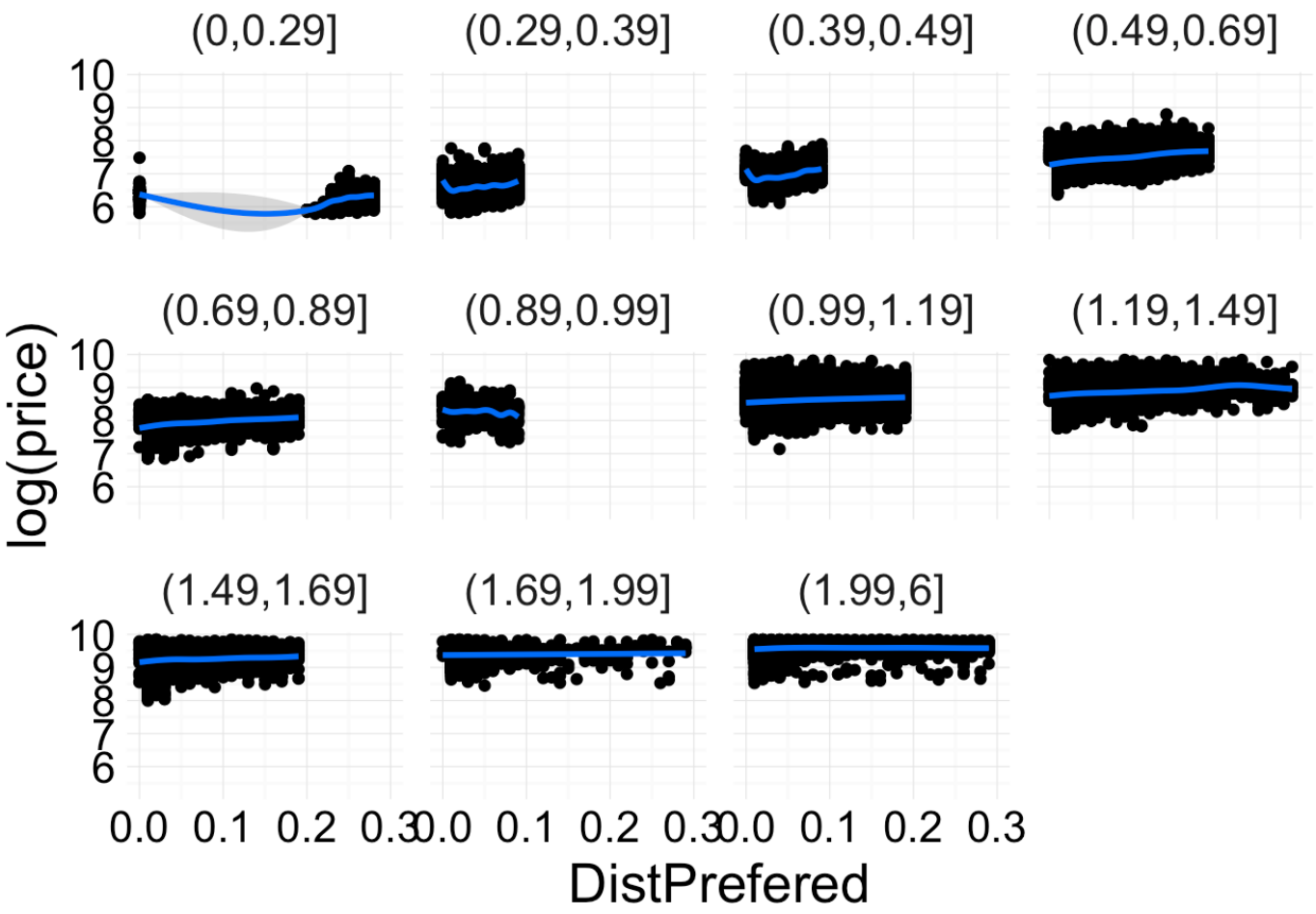


Fig 19. Price vs carat variables

I also plot price vs carat and carat.cut. It is clear that the price varies with carat, and the price for each carat is affected by the prefered size of that diamond.

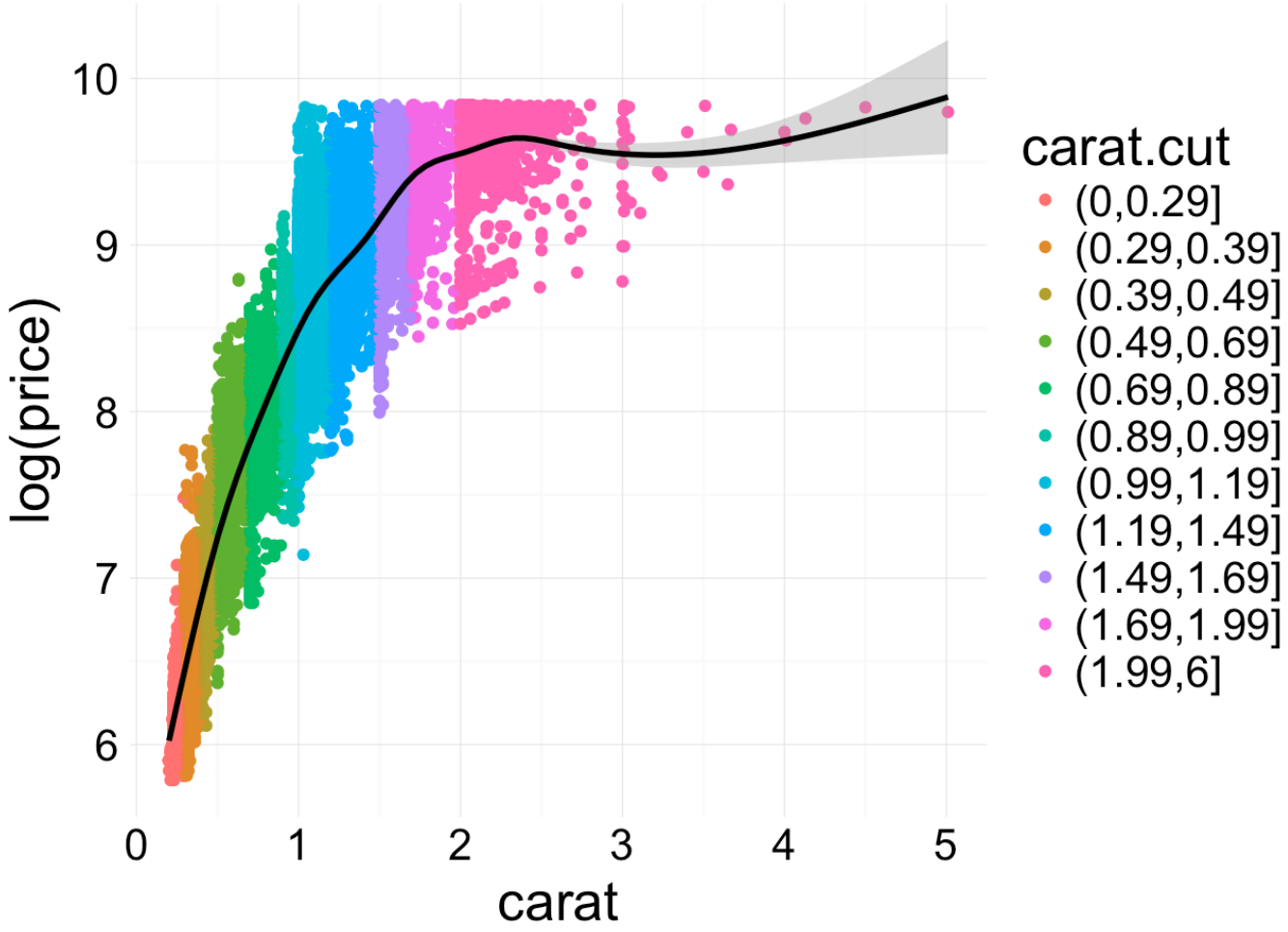


Fig 20. Price vs carat variables

Price for different cuts.

Although overall price vs cut distribution seemed to be bimodal, the distribution of price vs cut in bins refering to carat sizes is unimodal.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

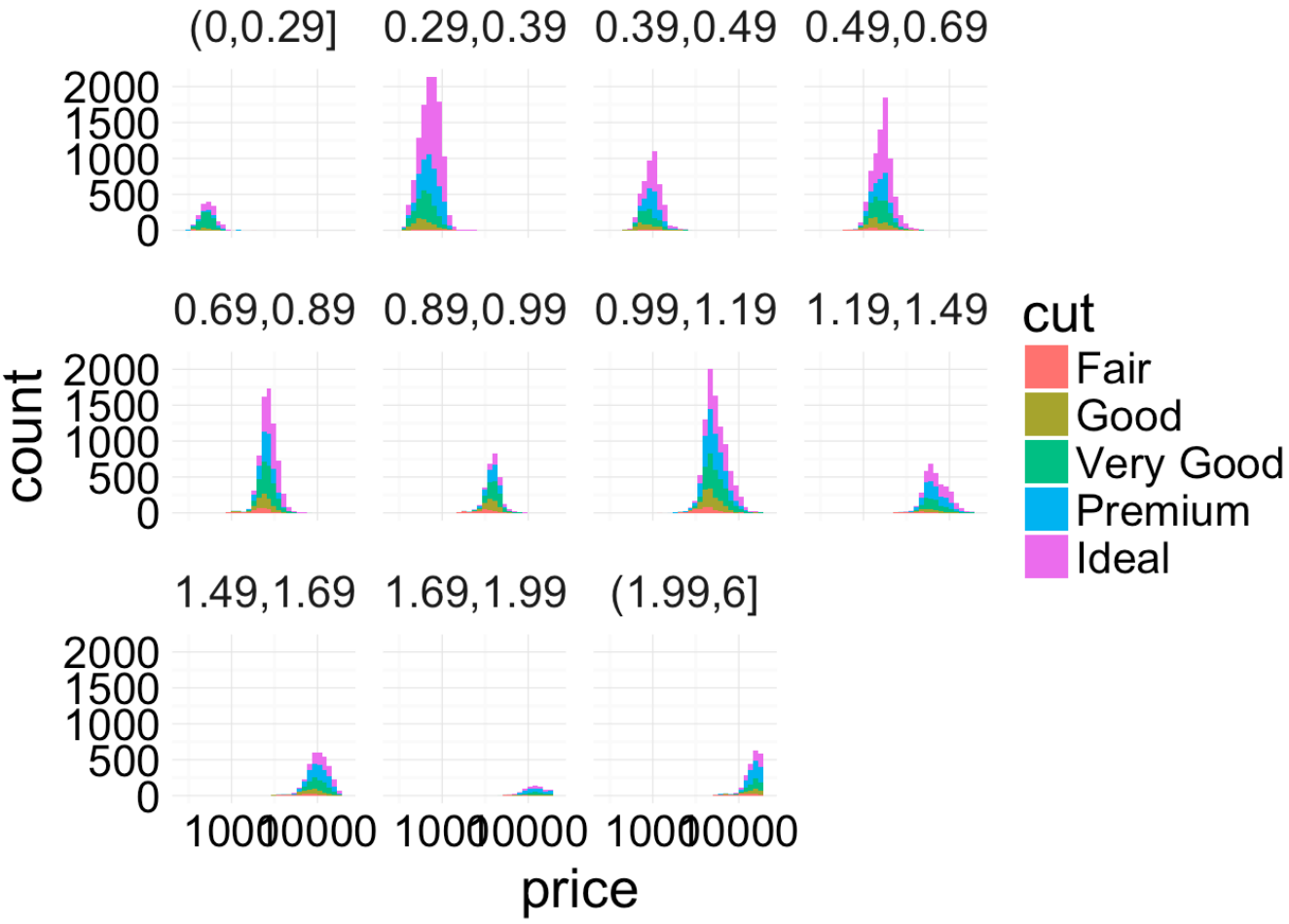


Fig 21. Price vs cut

Overall, price appear to scale with the cut of the diamond well, with ideal cuts being more expensive. These trends however may be less influencial because from the histogram, it appears that most of the diamonds are of Premium or better cut. Therefore, a better variable may be if the diamond is better than premium or worse. These trends show that the cut or cut.quality is a poor predictor of the diamond price.

```
diamonds$cut.quality = "bad"
ind_good = diamonds$cut == "Premium" | diamonds$cut == "Ideal"
diamonds$cut.quality[ind_good] = "good"
```

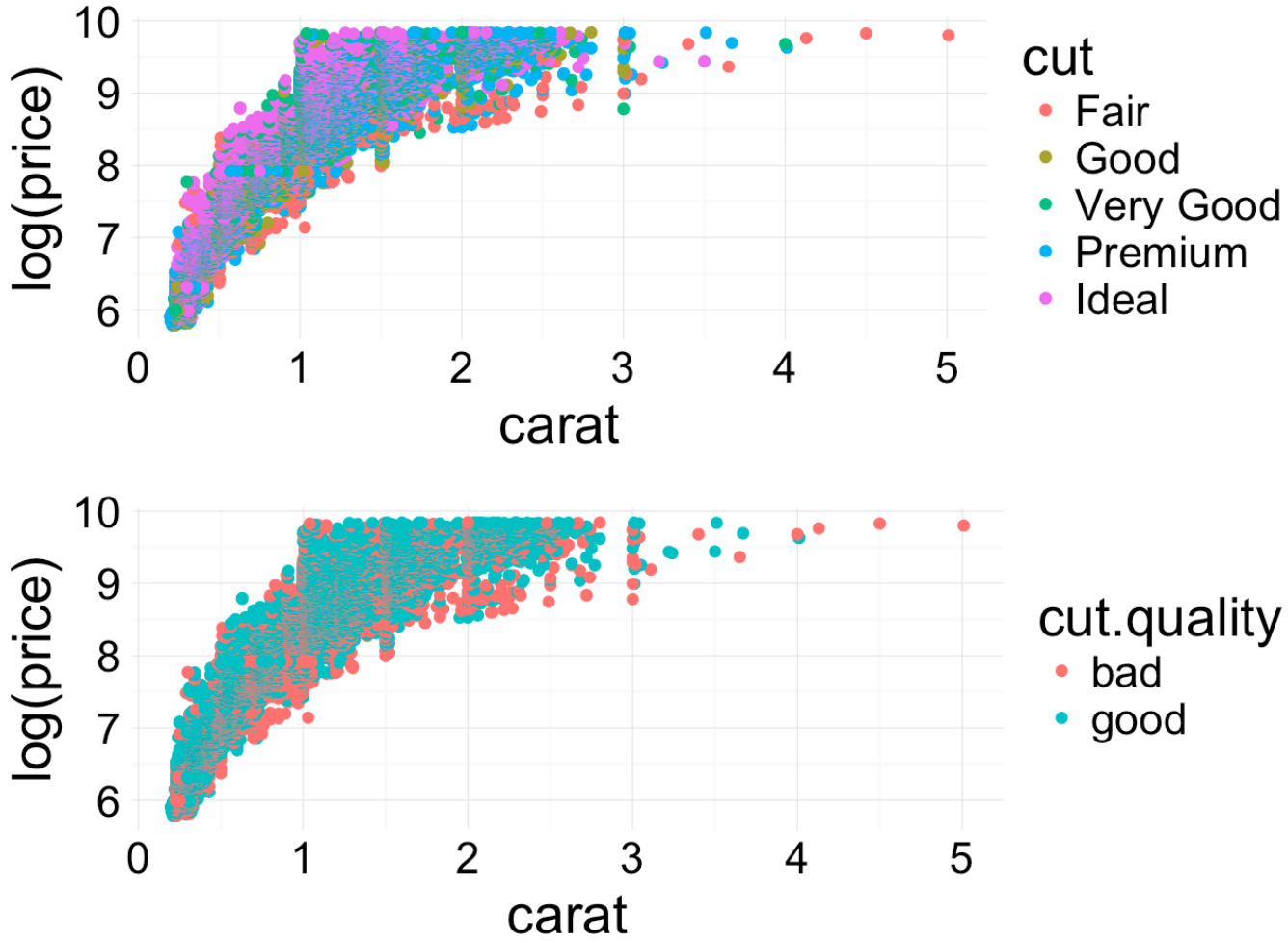


Fig 22. Price vs cut quality

Price for different clarity

Price vs clarity histograms reveal that for clearer diamonds, prices are higher. And clarity seems to be strong predictor of price.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

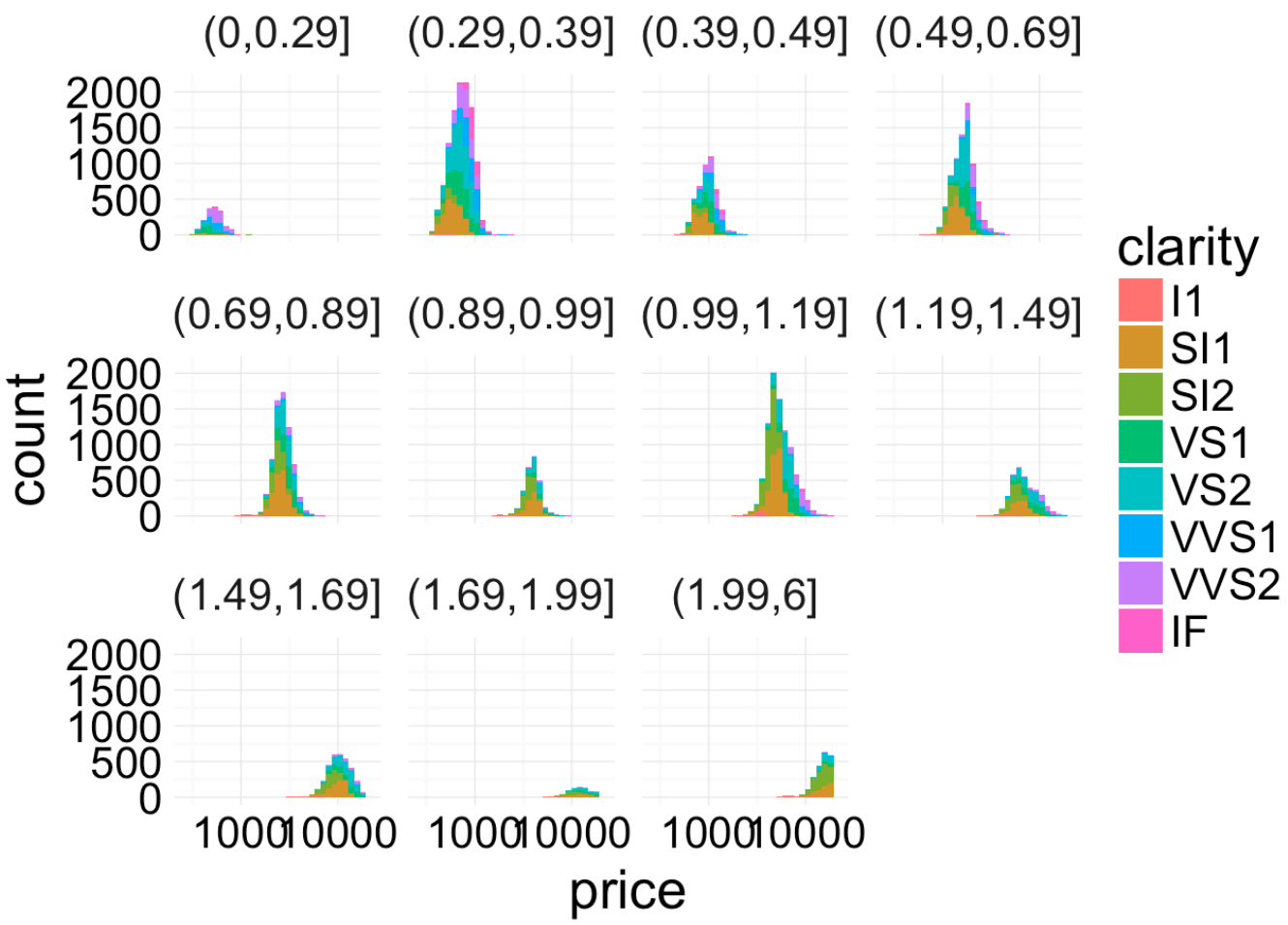


Fig 23. Price vs clarity

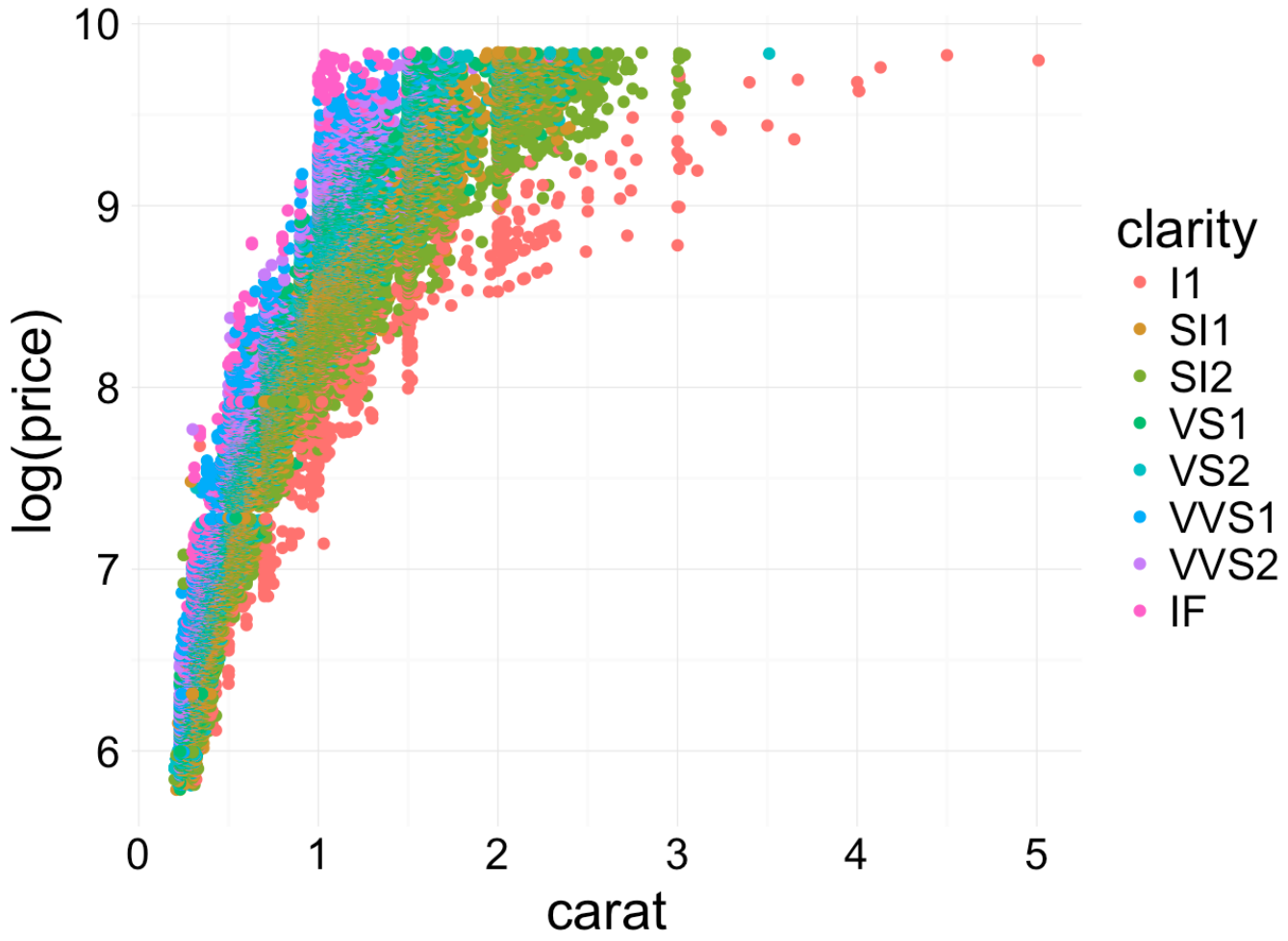


Fig 24. Price vs carat labeled by clarity

Price for different color

Price vs color histograms reveal that for clearer diamonds, prices are higher. And color seems to be strong predictor of price.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

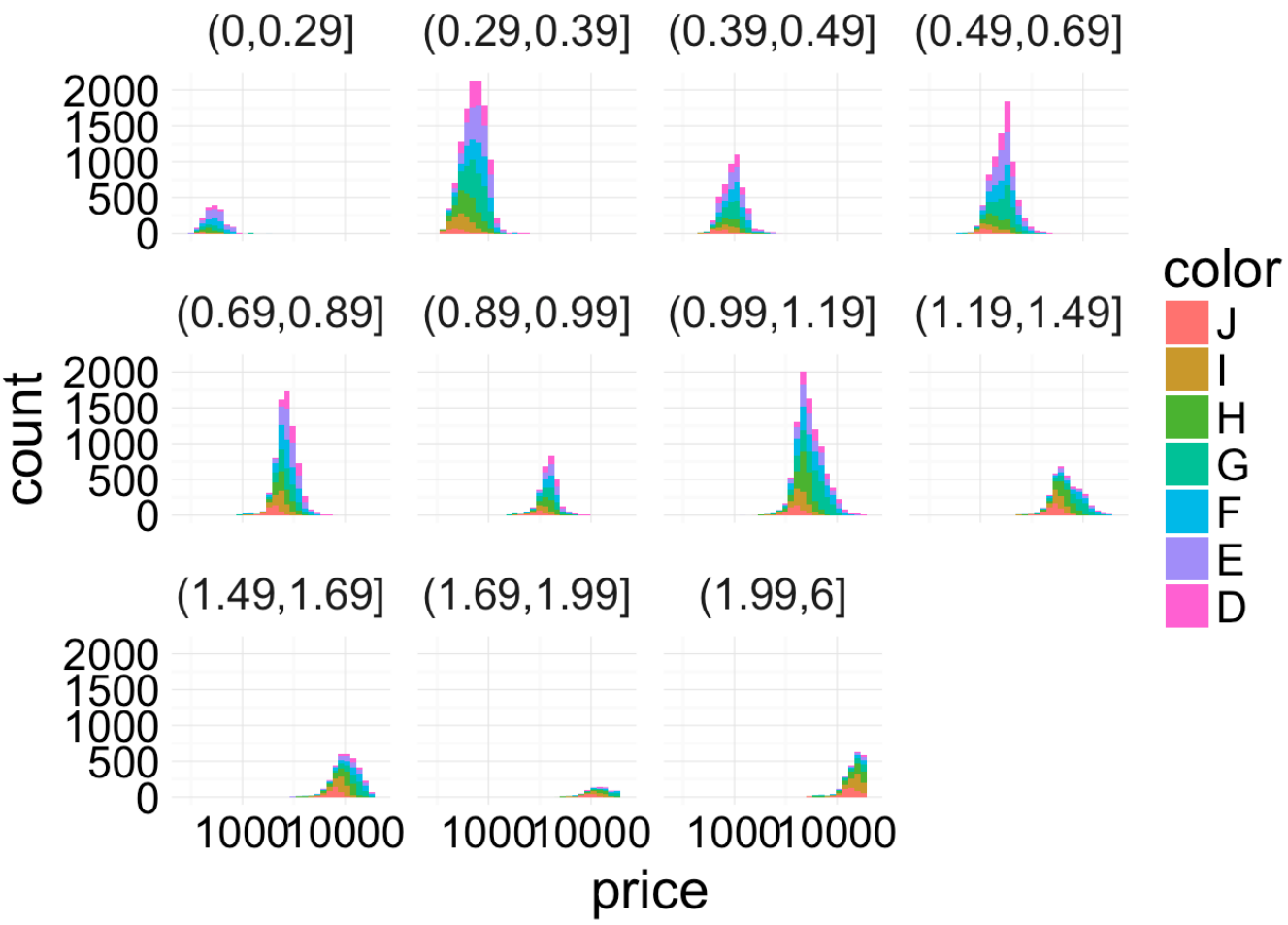


Fig 25. Price vs color

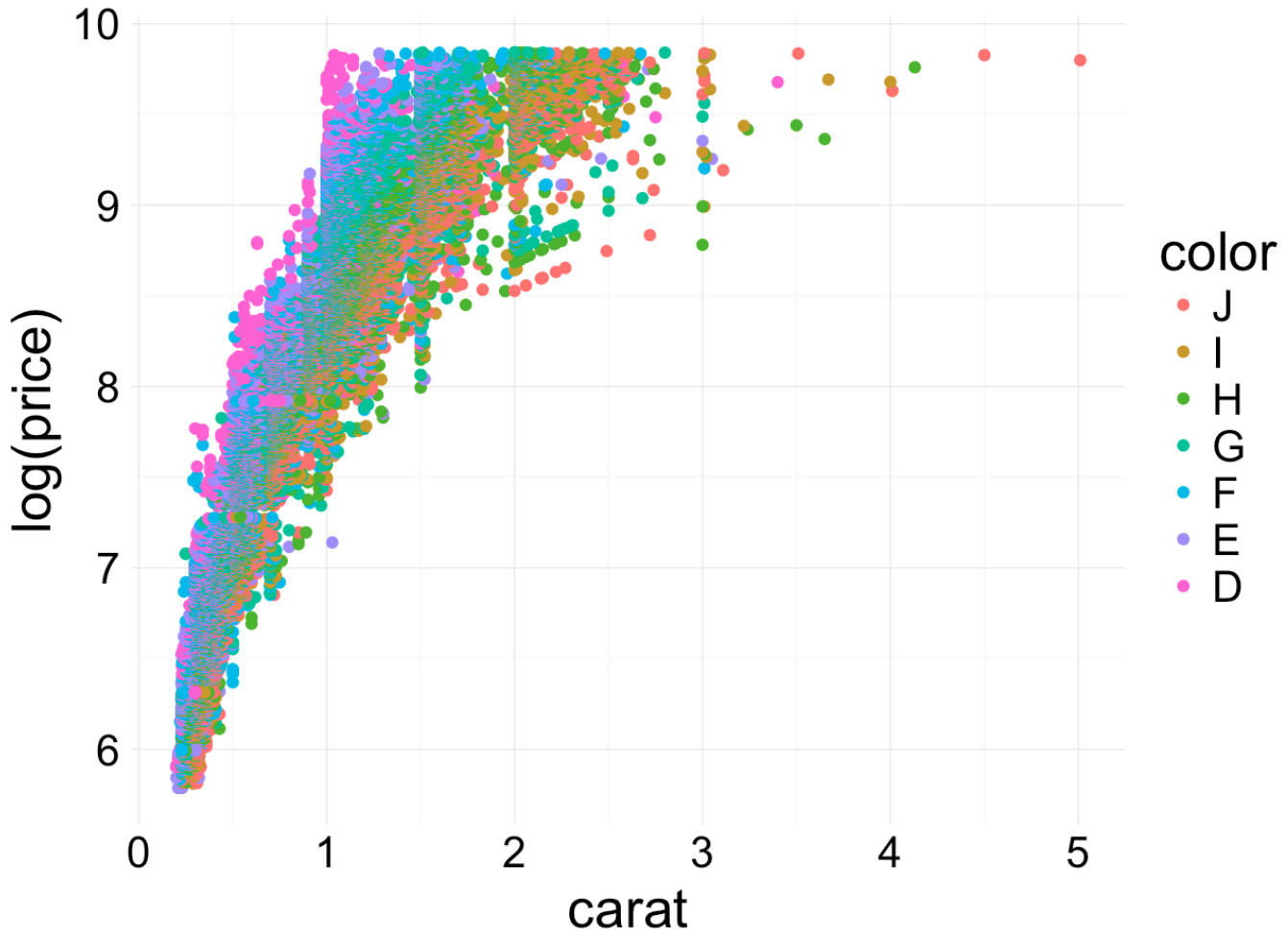


Fig 26. Price vs carat labeled by color

Price for different table

Price vs table.cut histograms reveal no effect of table parameter.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

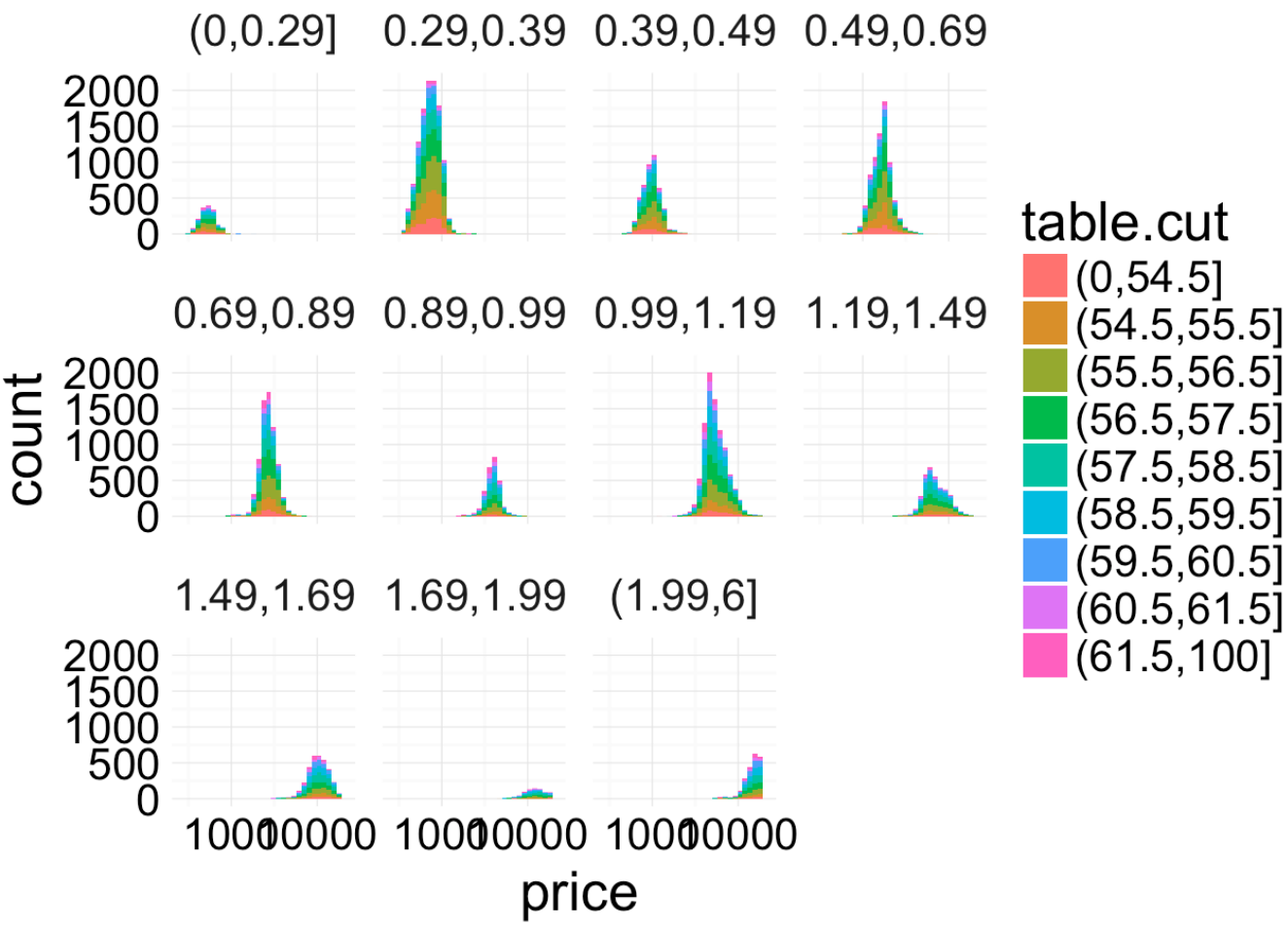


Fig 27. Price vs table

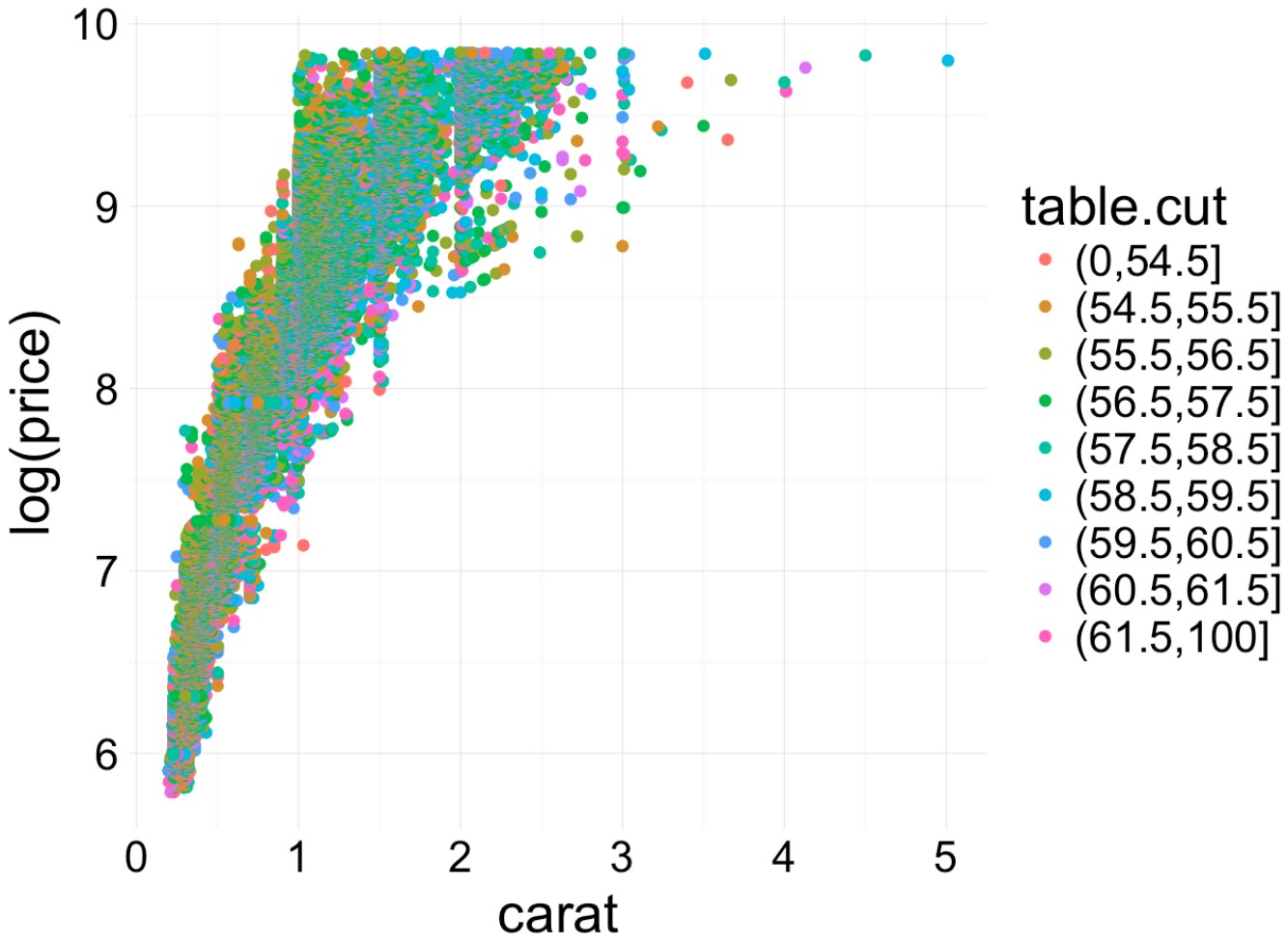


Fig 28. Price vs carat by table

Table is a shape factor, therefore I check if table.cut2 varies with the cut of the diamond. This plot conforms that table.cut varies with the cut of the diamond, with ideal cut diamonds having table.cut value below 57%, and poorer cuts having table.cut values above 57%

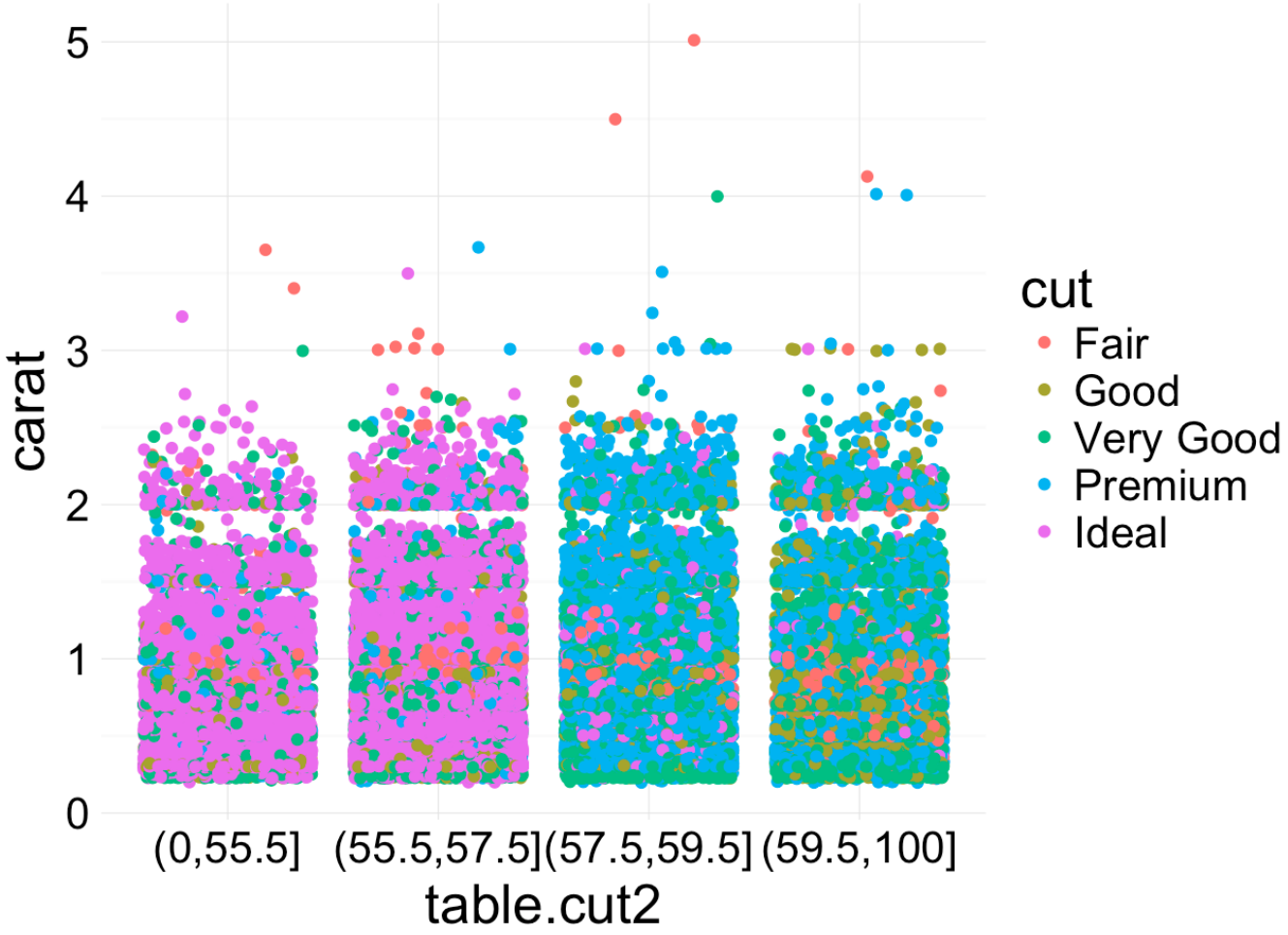


Fig 29. Comparison of finer and coarse table discretization

Price vs carat, color and clarity

Price vs carat plotted for different color reveals that for the best color (D), the variation in Price with clarity for a given carat is higher. These trends are however absent in poorer color (J). Therefore color * clarity will be included as an interaction term in the model.

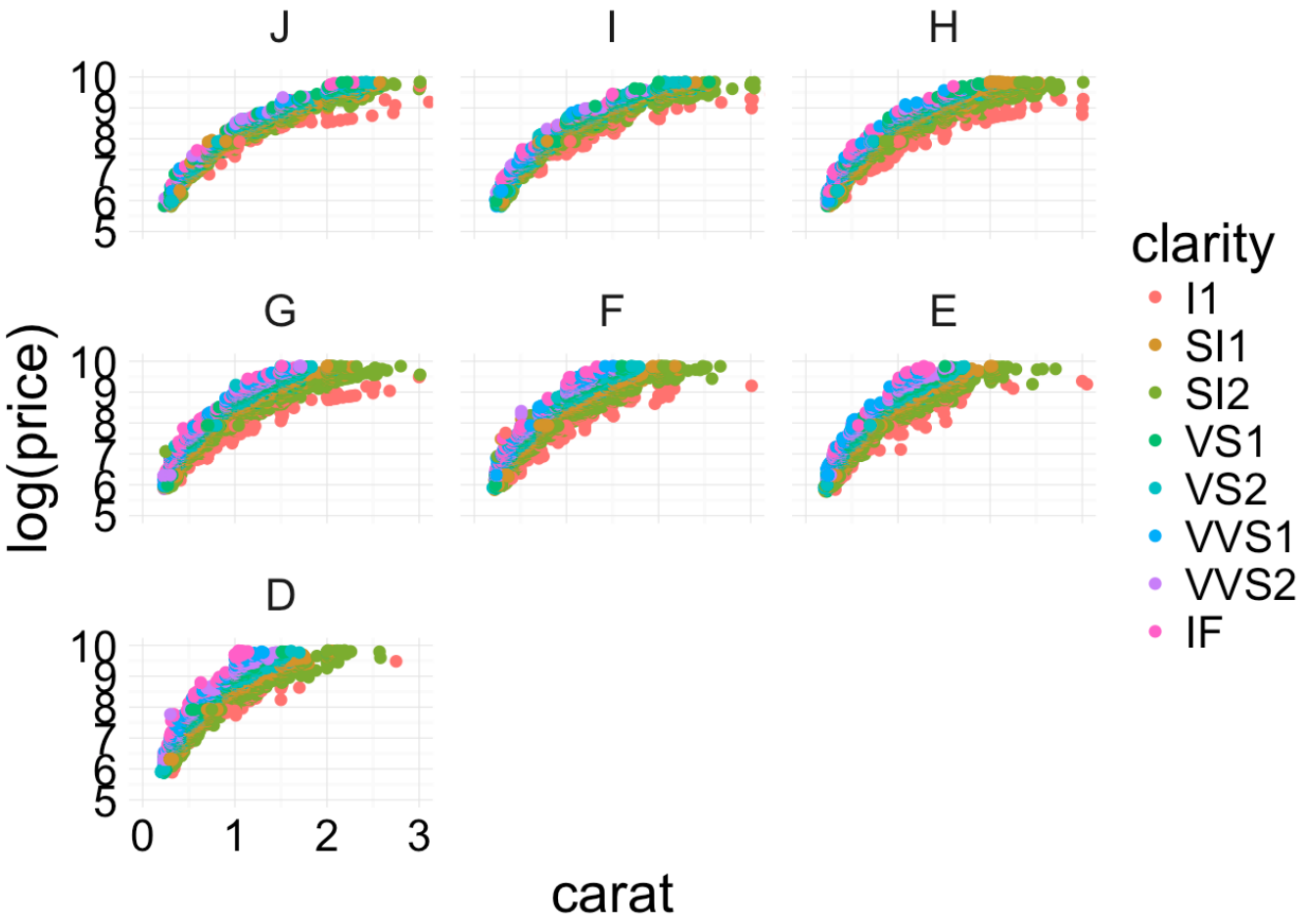


Fig 30. Price vs carat by clarity and color

Price vs color plotted for different prefered carat sizes reveals that for lower carat sizes, the price is determiend by clarity, however for higher carats, the effect of clarity is less. This is understandable because its tough to get larger diamond with no faults. Therefore color * carat.cut will be included as an interaction term in the model.

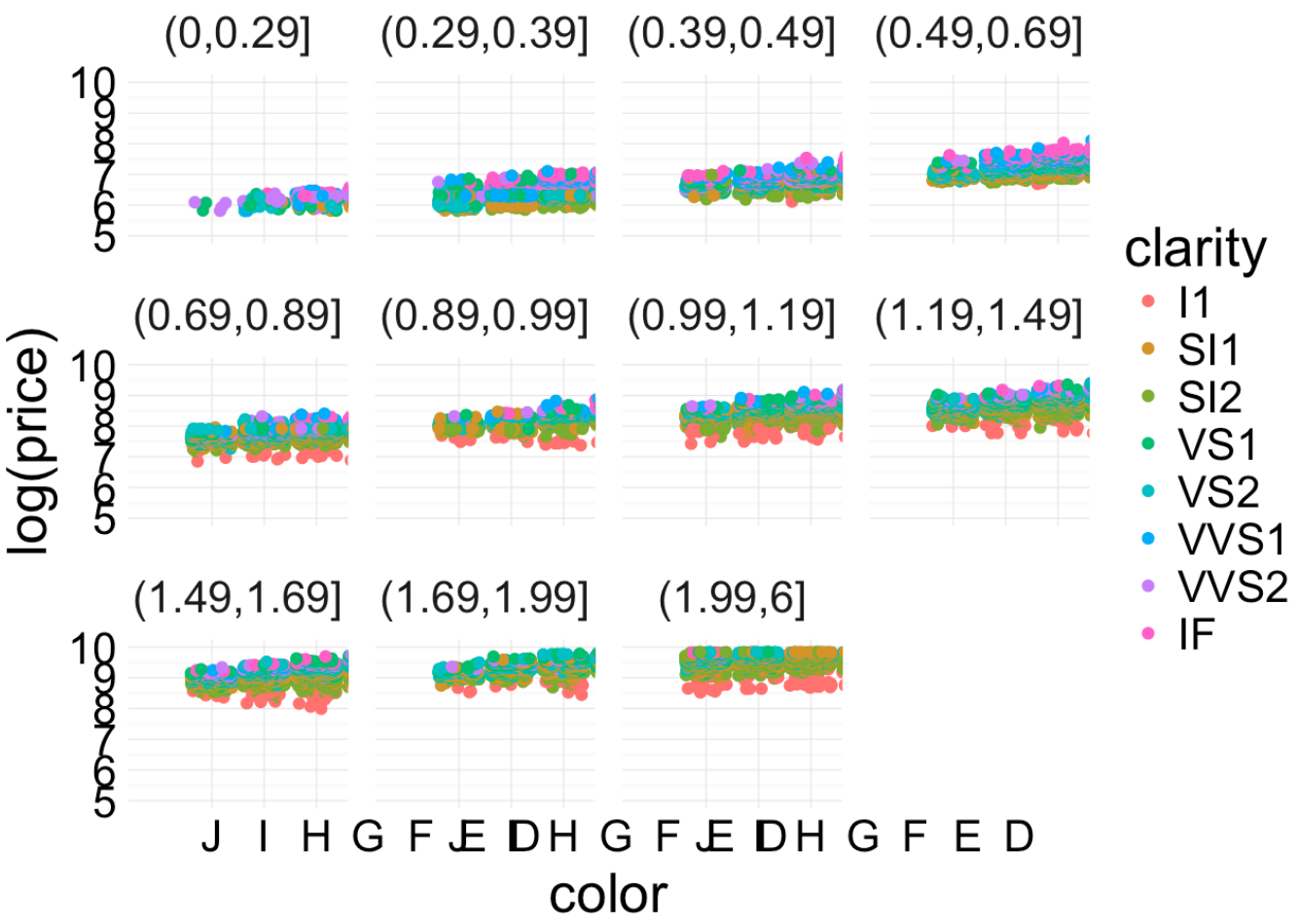


Fig 31. Price vs color by clarity and carat.cut

Price vs clarity plotted for different prefered carat sizes reveals that for higher carat sizes, the price is determiend by color, however for lower carats, the effect of color is less. These effects however do not seem very strong, therefore, carat.cut * clarity will not be included in the model.

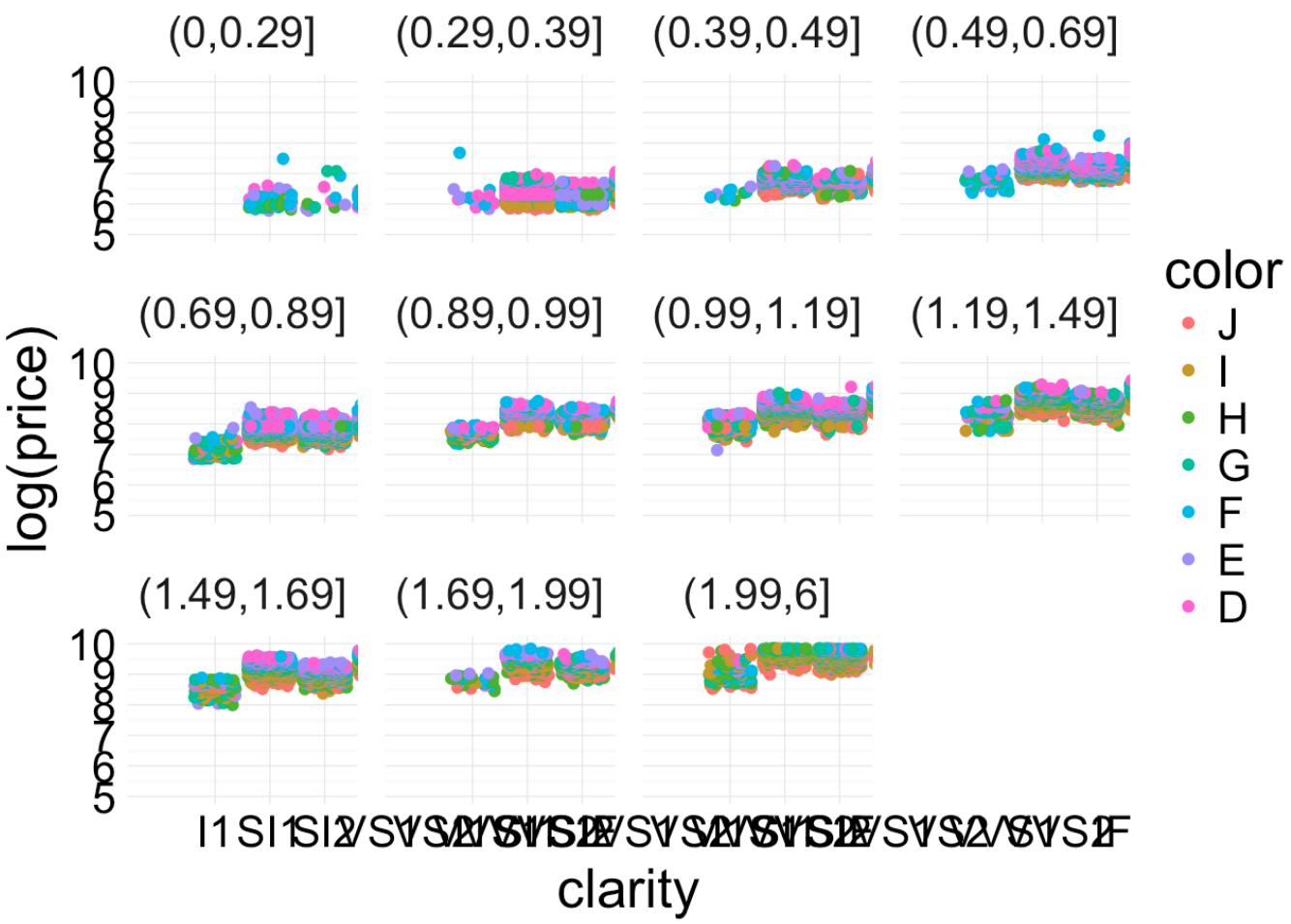


Fig 32. Price vs clarity by color and carat.cut

Modeling

Based on results above, color, carat and clarity are the 3 variables that predict the trends in diamonds prices the best. I will use these variables to make the final model for price. However, there are 3 variables that quanify carat, and so first I perform a variable selection to determine the best set of carat variables to predict price. For this, I remove the extreme data points. I remove the diamonds whose prices are in top 2% of the prices. These may represent specialized markets, so I did not want to include these diamonds. This process removed 1079 of 53940 diamonds. I generated a training set of 85% of the data and a testing set as the remaing 15%.

```
# creating training and testing data
up_limit = quantile(diamonds$price,probs=.98)
ind_rm = diamonds$price>up_limit
diamonds = diamonds[-ind_rm,]

set.seed(123)
train_ind <- sample(seq_len(nrow(diamonds)), size = floor(0.85*nrow(diamonds)))
diamonds_train <- diamonds[train_ind, ]
diamonds_test <-  diamonds[-train_ind, ]
```

Choosing the best carat variable.

The first task was to find the correct variable corresponding to carat value. For this, I fit log(price) to different combination of carat variables. These models were,

- m1: log(price) vs carat,
- m2: log(price) vs carat.cut,
- m3: log(price) vs carat.cut + DistPrefered

Results indicate that carat.cut + DistPrefered model explained greater percentage of price variance than carat alone (0.935 vs 0.847). I chose carat.cut + DistPrefered to model carat information, because this varaible captures 2 important features of the model. First, it considers the diamond's prefered size and second it considered deviations above the prefered size.

```
##
## Calls:
## m1: lm(formula = I(log(price)) ~ carat, data = diamonds)
## m2: lm(formula = I(log(price)) ~ carat.cut, data = diamonds)
## m3: lm(formula = I(log(price)) ~ carat.cut + DistPrefered, data = diamonds)
##
## ============================================================
##                                      m1         m2         m3
## ------------------------------------------------------------
## (Intercept)                        6.215***   6.271***   6.137***
##                                   (0.003)    (0.007)    (0.008)
## carat                              1.970***
##                                   (0.004)
## carat.cut: (0.29,0.39]/(0,0.29]                0.308***   0.423***
##                                               (0.007)    (0.008)
## carat.cut: (0.39,0.49]/(0,0.29]                0.607***   0.725***
##                                               (0.008)    (0.008)
## carat.cut: (0.49,0.69]/(0,0.29]                1.147***   1.250***
##                                               (0.007)    (0.008)
## carat.cut: (0.69,0.89]/(0,0.29]                1.622***   1.730***
##                                               (0.007)    (0.008)
## carat.cut: (0.89,0.99]/(0,0.29]                1.991***   2.112***
##                                               (0.008)    (0.009)
## carat.cut: (0.99,1.19]/(0,0.29]                2.328***   2.430***
##                                               (0.007)    (0.008)
## carat.cut: (1.19,1.49]/(0,0.29]                2.574***   2.667***
##                                               (0.008)    (0.008)
## carat.cut: (1.49,1.69]/(0,0.29]                2.957***   3.065***
##                                               (0.008)    (0.009)
## carat.cut: (1.69,1.99]/(0,0.29]                3.113***   3.213***
##                                               (0.012)    (0.012)
## carat.cut: (1.99,6]/(0,0.29]                   3.313***   3.364***
##                                               (0.009)    (0.009)
## DistPrefered                                              0.578***
##                                                          (0.018)
## ------------------------------------------------------------
## R-squared                             0.847      0.933      0.935
## adj. R-squared                        0.847      0.933      0.935
## sigma                                 0.397      0.262      0.259
## F                                298093.428 75679.438  70166.882
## p                                     0.000      0.000      0.000
## Log-likelihood                   -26726.512 -4223.837  -3727.590
## Deviance                           8507.660  3693.565   3626.223
## AIC                               53459.025  8471.675   7481.181
## BIC                               53485.712  8578.422   7596.824
## N                                 53939      53939      53939
## ============================================================
```

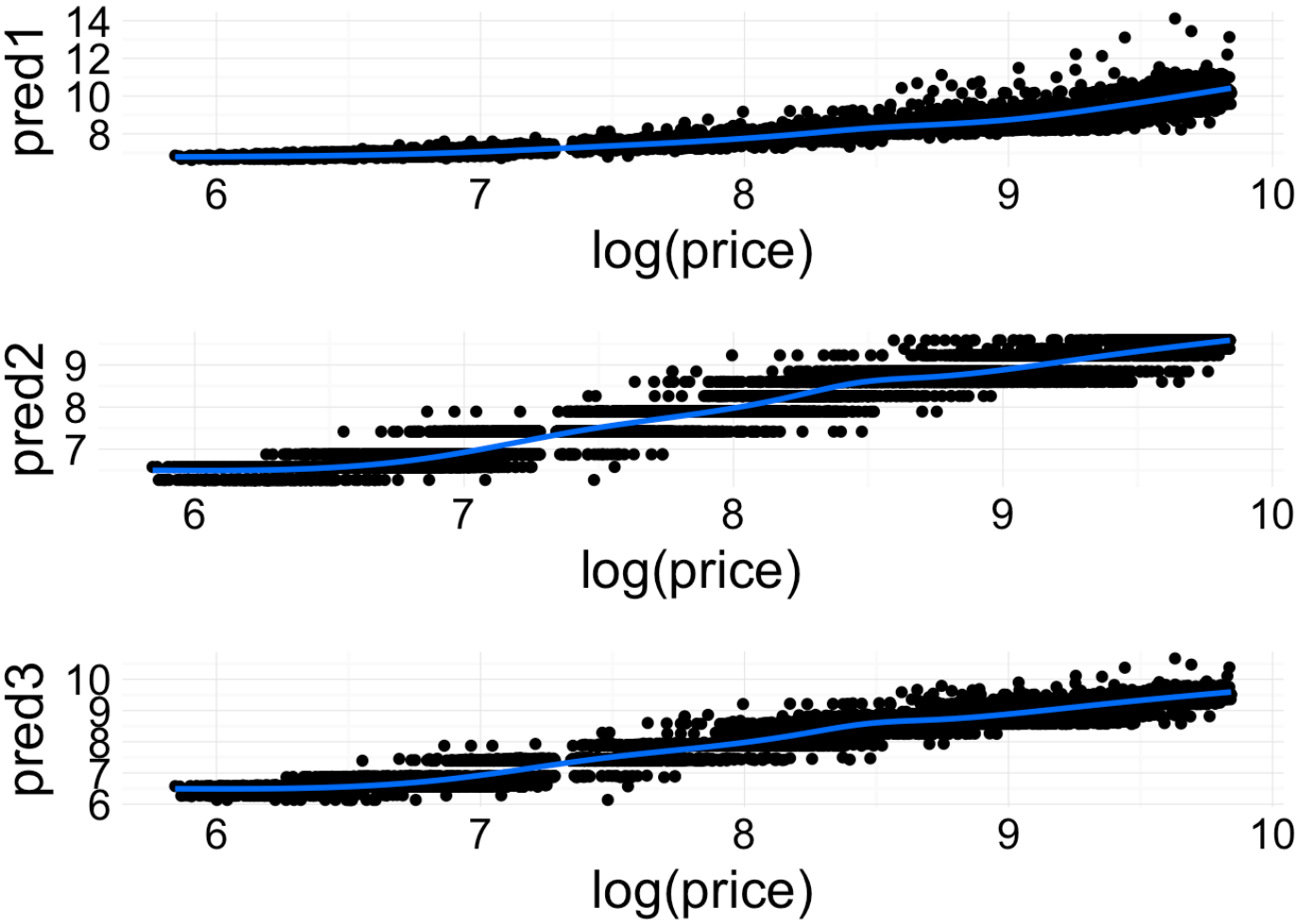


Fig 33. Results of fitted values for different carat variable combinations

Final model to predict diamond price.

I finally combine clarity and color to carat.cut + DistPrefered to build a linear regression model to predict price of the diamond. After combining clarity and color, more than 98% of variance is explained by the model. This is also confirmed by plots of log(price) and predicted values.

```
##
## Calls:
## m1: lm(formula = I(log(price)) ~ carat.cut, data = diamonds_train)
## m2: lm(formula = I(log(price)) ~ carat.cut + DistPrefered, data = diamonds_trai
n)
## m3: lm(formula = I(log(price)) ~ carat.cut + DistPrefered + clarity,
##     data = diamonds_train)
## m4: lm(formula = I(log(price)) ~ carat.cut + DistPrefered + clarity +
##     color, data = diamonds_train)
## m5: lm(formula = I(log(price)) ~ carat.cut + DistPrefered + clarity +
##     color + clarity:color + carat.cut:color, data = diamonds_train)
##
##
================================================================================
=================
##                                                    m1         m2         m3
m4         m5
## --------------------------------------------------------------------------------
--------------------
## (Intercept)                                         6.270***   6.132***   5.890***
5.753***   5.807***
##                                                    (0.007)    (0.008)    (0.006)
(0.005)    (0.009)
## carat.cut: (0.29,0.39]/(0,0.29]                     0.309***   0.427***   0.568***
0.634***   0.589***
##                                                    (0.008)    (0.008)    (0.006)
(0.005)    (0.009)
## carat.cut: (0.39,0.49]/(0,0.29]                     0.607***   0.729***   0.901***
0.961***   0.936***
##                                                    (0.008)    (0.009)    (0.007)
(0.005)    (0.009)
## carat.cut: (0.49,0.69]/(0,0.29]                     1.148***   1.255***   1.454***
1.502***   1.454***
##                                                    (0.008)    (0.009)    (0.006)
(0.005)    (0.009)
## carat.cut: (0.69,0.89]/(0,0.29]                     1.623***   1.734***   1.997***
2.084***   2.042***
##                                                    (0.008)    (0.009)    (0.006)
(0.005)    (0.009)
## carat.cut: (0.89,0.99]/(0,0.29]                     1.990***   2.115***   2.450***
2.554***   2.513***
##                                                    (0.009)    (0.010)    (0.007)
(0.005)    (0.010)
## carat.cut: (0.99,1.19]/(0,0.29]                     2.328***   2.433***   2.727***
2.824***   2.767***
##                                                    (0.008)    (0.008)    (0.006)
(0.005)    (0.009)
## carat.cut: (1.19,1.49]/(0,0.29]                     2.578***   2.673***   2.937***
3.101***   3.045***
##                                                    (0.009)    (0.009)    (0.007)
```

```
(0.005)    (0.009)
## carat.cut: (1.49,1.69]/(0,0.29]              2.960***   3.071***   3.358***
3.511***   3.462***
##                                             (0.009)    (0.009)    (0.007)
(0.005)    (0.009)
## carat.cut: (1.69,1.99]/(0,0.29]              3.113***   3.216***   3.520***
3.705***   3.658***
##                                             (0.013)    (0.013)    (0.009)
(0.007)    (0.011)
## carat.cut: (1.99,6]/(0,0.29]                 3.315***   3.368***   3.760***
3.987***   3.921***
##                                             (0.009)    (0.009)    (0.007)
(0.005)    (0.010)
## DistPrefered                                            0.596***   0.783***
0.909***   0.882***
##                                                        (0.020)    (0.014)
(0.011)    (0.010)
## clarity: .L                                                         0.863***
0.899***   0.878***
##                                                                    (0.005)
(0.004)    (0.005)
## clarity: .Q                                                        -0.238***
-0.236***  -0.219***
##                                                                    (0.005)
(0.004)    (0.004)
## clarity: .C                                                         0.148***
0.149***   0.169***
##                                                                    (0.004)
(0.003)    (0.004)
## clarity: ^4                                                        -0.124***
-0.110***  -0.102***
##                                                                    (0.003)
(0.002)    (0.003)
## clarity: ^5                                                         0.178***
0.193***   0.196***
##                                                                    (0.003)
(0.002)    (0.002)
## clarity: ^6                                                        -0.076***
-0.086***  -0.071***
##                                                                    (0.003)
(0.002)    (0.002)
## clarity: ^7                                                         0.208***
0.240***   0.231***
##                                                                    (0.002)
(0.002)    (0.002)
## color: .L
0.432***   0.200***
##
(0.002)    (0.033)
## color: .Q
-0.090***  -0.057
```

```
##
(0.002)    (0.031)
## color: .C
0.012***   0.012
##
(0.002)    (0.025)
## color: ^4
0.010***  -0.015
##
(0.002)    (0.019)
## color: ^5
0.007***   0.023
##
(0.002)    (0.013)
## color: ^6
0.001      0.012
##
(0.002)    (0.010)
## clarity: .L x color: .L
0.405***
##
(0.015)
## clarity: .Q x color: .L
0.151***
##
(0.014)
## clarity: .C x color: .L
-0.062***
##
(0.013)
## clarity: ^4 x color: .L
0.094***
##
(0.009)
## clarity: ^5 x color: .L
0.059***
##
(0.008)
## clarity: ^6 x color: .L
-0.017*
##
(0.008)
## clarity: ^7 x color: .L
0.128***
##
(0.006)
## clarity: .L x color: .Q
-0.005
##
(0.014)
## clarity: .Q x color: .Q
```

```
0.110***
##
(0.013)
## clarity: .C x color: .Q
0.039***
##
(0.012)
## clarity: ^4 x color: .Q
0.031***
##
(0.009)
## clarity: ^5 x color: .Q
0.022**
##
(0.008)
## clarity: ^6 x color: .Q
0.063***
##
(0.008)
## clarity: ^7 x color: .Q
-0.011
##
(0.006)
## clarity: .L x color: .C
0.063***
##
(0.013)
## clarity: .Q x color: .C
0.087***
##
(0.012)
## clarity: .C x color: .C
0.060***
##
(0.011)
## clarity: ^4 x color: .C
0.036***
##
(0.008)
## clarity: ^5 x color: .C
0.037***
##
(0.007)
## clarity: ^6 x color: .C
0.020**
##
(0.007)
## clarity: ^7 x color: .C
0.002
##
(0.006)
```

```
## clarity: .L x color: ^4
0.098***
##
(0.011)
## clarity: .Q x color: ^4
-0.004
##
(0.011)
## clarity: .C x color: ^4
0.021*
##
(0.009)
## clarity: ^4 x color: ^4
-0.005
##
(0.007)
## clarity: ^5 x color: ^4
0.026***
##
(0.006)
## clarity: ^6 x color: ^4
0.017**
##
(0.006)
## clarity: ^7 x color: ^4
0.024***
##
(0.005)
## clarity: .L x color: ^5
0.028**
##
(0.010)
## clarity: .Q x color: ^5
0.006
##
(0.009)
## clarity: .C x color: ^5
-0.017*
##
(0.008)
## clarity: ^4 x color: ^5
0.008
##
(0.006)
## clarity: ^5 x color: ^5
0.003
##
(0.005)
## clarity: ^6 x color: ^5
-0.008
##
```

```
(0.005)
## clarity: ^7 x color: ^5
0.007
##
(0.004)
## clarity: .L x color: ^6
-0.023**
##
(0.009)
## clarity: .Q x color: ^6
0.031***
##
(0.008)
## clarity: .C x color: ^6
-0.021**
##
(0.007)
## clarity: ^4 x color: ^6
-0.003
##
(0.005)
## clarity: ^5 x color: ^6
0.001
##
(0.005)
## clarity: ^6 x color: ^6
0.016***
##
(0.004)
## clarity: ^7 x color: ^6
0.001
##
(0.004)
## carat.cut: (0.29,0.39]/(0,0.29] x color: .L
0.300***
##
(0.033)
## carat.cut: (0.39,0.49]/(0,0.29] x color: .L
0.177***
##
(0.034)
## carat.cut: (0.49,0.69]/(0,0.29] x color: .L
0.300***
##
(0.034)
## carat.cut: (0.69,0.89]/(0,0.29] x color: .L
0.301***
##
(0.033)
## carat.cut: (0.89,0.99]/(0,0.29] x color: .L
0.295***
```

```
##
(0.034)
## carat.cut: (0.99,1.19]/(0,0.29] x color: .L
0.312***
##
(0.033)
## carat.cut: (1.19,1.49]/(0,0.29] x color: .L
0.356***
##
(0.034)
## carat.cut: (1.49,1.69]/(0,0.29] x color: .L
0.381***
##
(0.034)
## carat.cut: (1.69,1.99]/(0,0.29] x color: .L
0.360***
##
(0.038)
## carat.cut: (1.99,6]/(0,0.29] x color: .L
0.320***
##
(0.035)
## carat.cut: (0.29,0.39]/(0,0.29] x color: .Q
0.031
##
(0.032)
## carat.cut: (0.39,0.49]/(0,0.29] x color: .Q
0.005
##
(0.032)
## carat.cut: (0.49,0.69]/(0,0.29] x color: .Q
0.010
##
(0.032)
## carat.cut: (0.69,0.89]/(0,0.29] x color: .Q
0.017
##
(0.032)
## carat.cut: (0.89,0.99]/(0,0.29] x color: .Q
0.005
##
(0.032)
## carat.cut: (0.99,1.19]/(0,0.29] x color: .Q
-0.014
##
(0.032)
## carat.cut: (1.19,1.49]/(0,0.29] x color: .Q
-0.031
##
(0.032)
## carat.cut: (1.49,1.69]/(0,0.29] x color: .Q
```

```
-0.055
##
(0.032)
## carat.cut: (1.69,1.99]/(0,0.29] x color: .Q
-0.070
##
(0.036)
## carat.cut: (1.99,6]/(0,0.29] x color: .Q
-0.045
##
(0.034)
## carat.cut: (0.29,0.39]/(0,0.29] x color: .C
0.016
##
(0.026)
## carat.cut: (0.39,0.49]/(0,0.29] x color: .C
0.040
##
(0.026)
## carat.cut: (0.49,0.69]/(0,0.29] x color: .C
0.021
##
(0.026)
## carat.cut: (0.69,0.89]/(0,0.29] x color: .C
0.035
##
(0.026)
## carat.cut: (0.89,0.99]/(0,0.29] x color: .C
-0.010
##
(0.026)
## carat.cut: (0.99,1.19]/(0,0.29] x color: .C
0.009
##
(0.026)
## carat.cut: (1.19,1.49]/(0,0.29] x color: .C
0.005
##
(0.026)
## carat.cut: (1.49,1.69]/(0,0.29] x color: .C
-0.002
##
(0.026)
## carat.cut: (1.69,1.99]/(0,0.29] x color: .C
-0.019
##
(0.030)
## carat.cut: (1.99,6]/(0,0.29] x color: .C
-0.017
##
(0.028)
```

```
## carat.cut: (0.29,0.39]/(0,0.29] x color: ^4
0.025
##
(0.019)
## carat.cut: (0.39,0.49]/(0,0.29] x color: ^4
0.017
##
(0.020)
## carat.cut: (0.49,0.69]/(0,0.29] x color: ^4
0.046*
##
(0.020)
## carat.cut: (0.69,0.89]/(0,0.29] x color: ^4
0.018
##
(0.019)
## carat.cut: (0.89,0.99]/(0,0.29] x color: ^4
0.046*
##
(0.020)
## carat.cut: (0.99,1.19]/(0,0.29] x color: ^4
0.049*
##
(0.019)
## carat.cut: (1.19,1.49]/(0,0.29] x color: ^4
0.047*
##
(0.020)
## carat.cut: (1.49,1.69]/(0,0.29] x color: ^4
0.032
##
(0.020)
## carat.cut: (1.69,1.99]/(0,0.29] x color: ^4
-0.006
##
(0.024)
## carat.cut: (1.99,6]/(0,0.29] x color: ^4
-0.000
##
(0.022)
## carat.cut: (0.29,0.39]/(0,0.29] x color: ^5
-0.019
##
(0.014)
## carat.cut: (0.39,0.49]/(0,0.29] x color: ^5
-0.031*
##
(0.015)
## carat.cut: (0.49,0.69]/(0,0.29] x color: ^5
-0.014
##
```

```
(0.014)
## carat.cut: (0.69,0.89]/(0,0.29] x color: ^5
-0.017
##
(0.014)
## carat.cut: (0.89,0.99]/(0,0.29] x color: ^5
-0.019
##
(0.015)
## carat.cut: (0.99,1.19]/(0,0.29] x color: ^5
-0.012
##
(0.014)
## carat.cut: (1.19,1.49]/(0,0.29] x color: ^5
-0.012
##
(0.015)
## carat.cut: (1.49,1.69]/(0,0.29] x color: ^5
0.020
##
(0.015)
## carat.cut: (1.69,1.99]/(0,0.29] x color: ^5
-0.011
##
(0.020)
## carat.cut: (1.99,6]/(0,0.29] x color: ^5
0.015
##
(0.017)
## carat.cut: (0.29,0.39]/(0,0.29] x color: ^6
-0.003
##
(0.011)
## carat.cut: (0.39,0.49]/(0,0.29] x color: ^6
0.014
##
(0.012)
## carat.cut: (0.49,0.69]/(0,0.29] x color: ^6
-0.000
##
(0.011)
## carat.cut: (0.69,0.89]/(0,0.29] x color: ^6
-0.008
##
(0.011)
## carat.cut: (0.89,0.99]/(0,0.29] x color: ^6
0.013
##
(0.012)
## carat.cut: (0.99,1.19]/(0,0.29] x color: ^6
-0.009
```

```
##
(0.011)
## carat.cut: (1.19,1.49]/(0,0.29] x color: ^6
-0.002
##
(0.012)
## carat.cut: (1.49,1.69]/(0,0.29] x color: ^6
-0.015
##
(0.012)
## carat.cut: (1.69,1.99]/(0,0.29] x color: ^6
-0.040*
##
(0.017)
## carat.cut: (1.99,6]/(0,0.29] x color: ^6
-0.009
##
(0.014)
## ------------------------------------------------------------------------------
--------------------
## R-squared                                        0.933      0.934       0.966
0.981      0.983
## adj. R-squared                                   0.933      0.934       0.966
0.981      0.983
## sigma                                            0.262      0.260       0.187
0.139      0.133
## F                                            63989.415  59370.592   72568.453
99366.319  20856.337
## p                                                0.000      0.000       0.000
0.000      0.000
## Log-likelihood                               -3716.796  -3279.825   11850.305
25298.553  27534.042
## Deviance                                      3156.899   3097.293    1600.820
890.360    807.634
## AIC                                           7457.593   6585.650  -23660.610
-50545.105 -54812.083
## BIC                                           7562.390   6699.180  -23485.949
-50318.045 -53694.248
## N                                            45848      45848       45848
45848      45848
##
================================================================================
=================
```

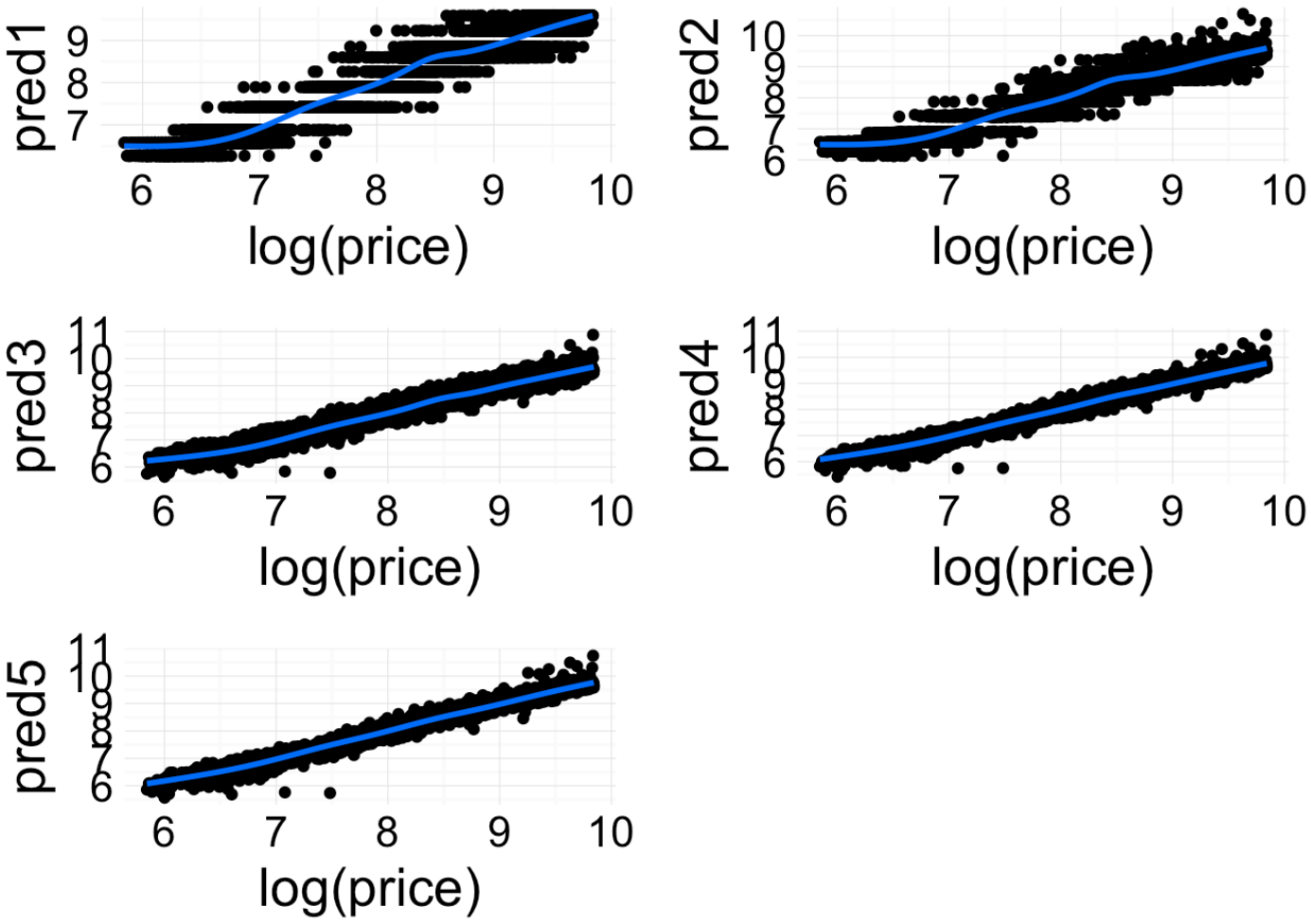


Fig 34. Results of fitted values for adding variables incrementally.

I also compared model performance to predict data between the training set and testing set. it appears that the model generalizes equally well to both the testing set.

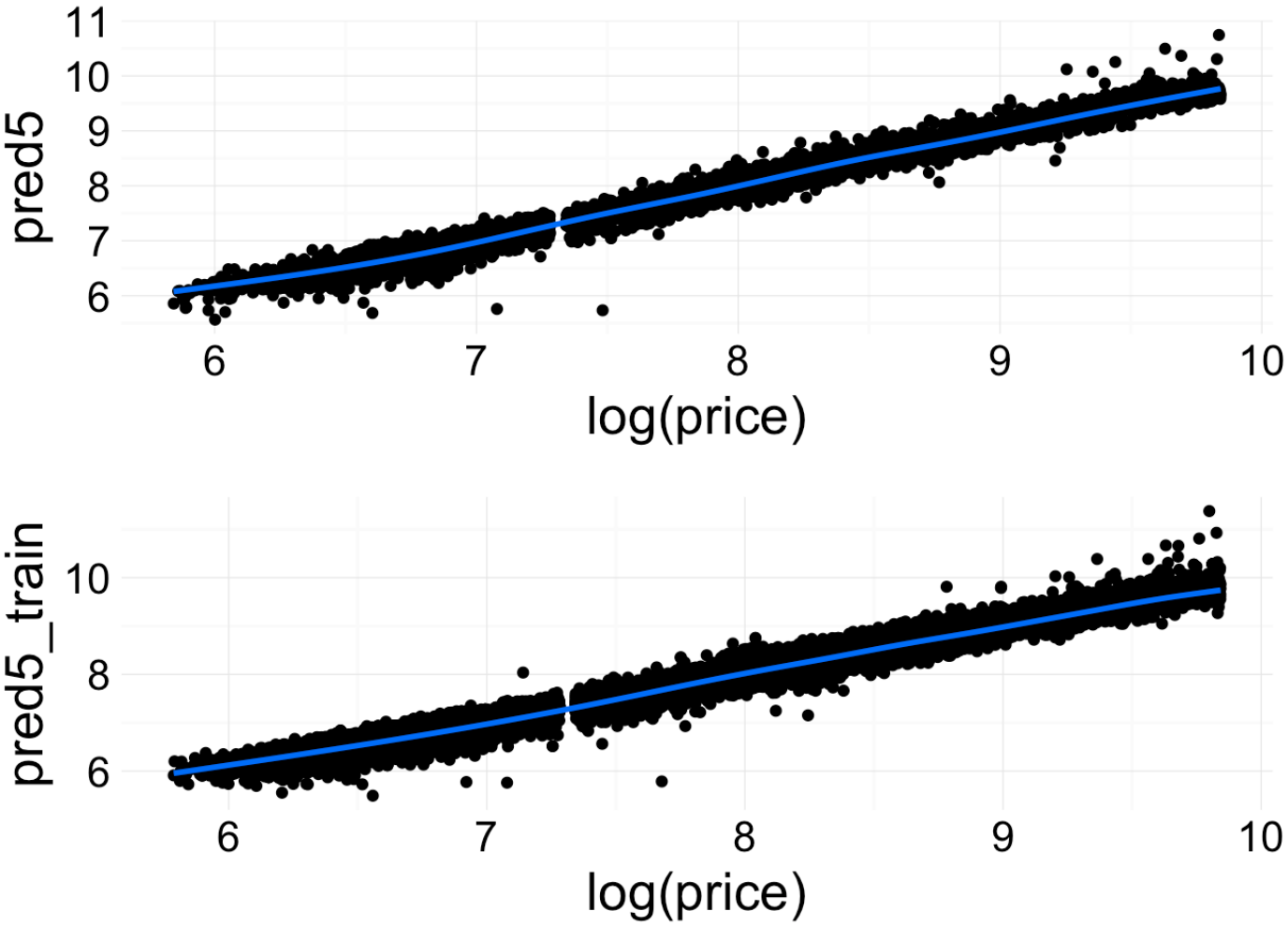


Fig 35. Performance of model on training and tesing set

Multivariate Analysis

Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

It appears that a better predictor of diamond prices is not carat, but the prefered carat size for the diamond. Prefered carat size explained 93.3% of variance as against 84.7% for carat. Carat was represented by 2 variables, 1 indicating prefered carat size, carat.cut and other deviation from prefered size, DistPrefered.

For a given carat, diamonds with lower clarity are almost always cheaper than diamonds with better clarity (worst clarity is I1 and best clarity is IF).

For a given carat, diamonds with poorer color are almost always cheaper than diamonds with better color (worst color is J and best is D).

Therefore, these 4 variables were used to build the final model. In addition, interactions between carat and clarity, and color and clarity were included in the model.

Were there any interesting or surprising interactions between features?

Table is a shape factor, therefore I check if table.cut2 varies with the cut of the diamond. This plot conforms that table.cut varies with the cut of the diamond, with ideal cut diamonds having table.cut value below 57%, and poorer cuts having table.cut values above 57%

OPTIONAL: Did you create any models with your dataset? Discuss the strengths and limitations of your model.

I created a linear model with log(price) as output and carat.cut, DistPrefered, clarity and color as predictors. The results show that 98.4% variance in diamonds' prices can be explained using this model.

Final Plots and Summary

Plot One

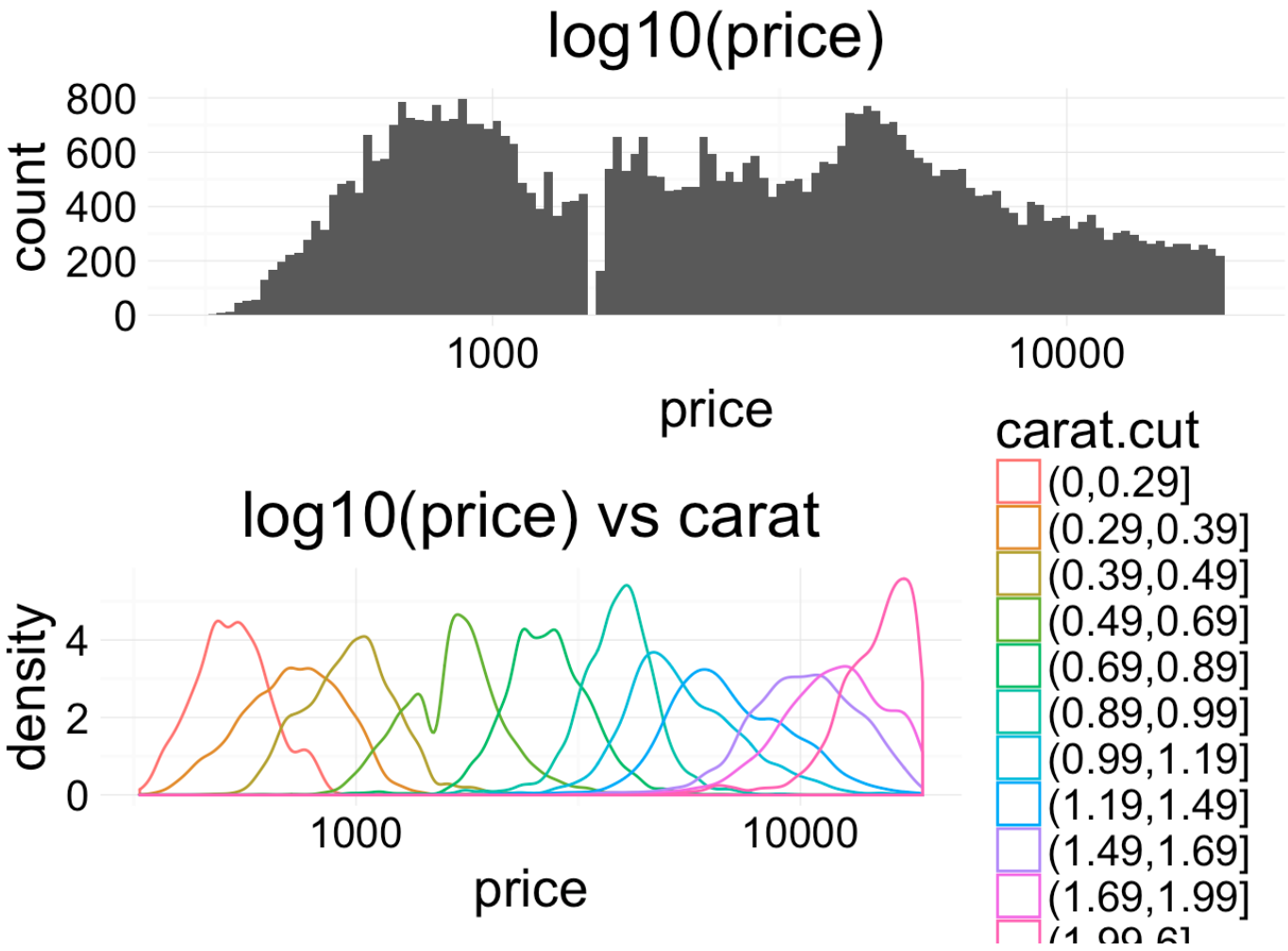


Fig 36. Price and price density for different carat.cut

Description One

The distribution of diamond prices appears to be bimodal. This may be due to buyers purchasing diamonds in different price ranges, and most people being comfortable buying diamonds that have some prefered price (like 1000 or 4000). The peak may also be related to large numbers of diamonds individuals purchase for certain occasions. For example, most prefered size of diamond for engagement rings is .37 carats, and assuming price of 3000 per carat it corresponds to the peak around 1100.

Density plots of price grouped by prefered carats (carat.cut) reveal the clear trend in price increase with carat size.

Plot Two

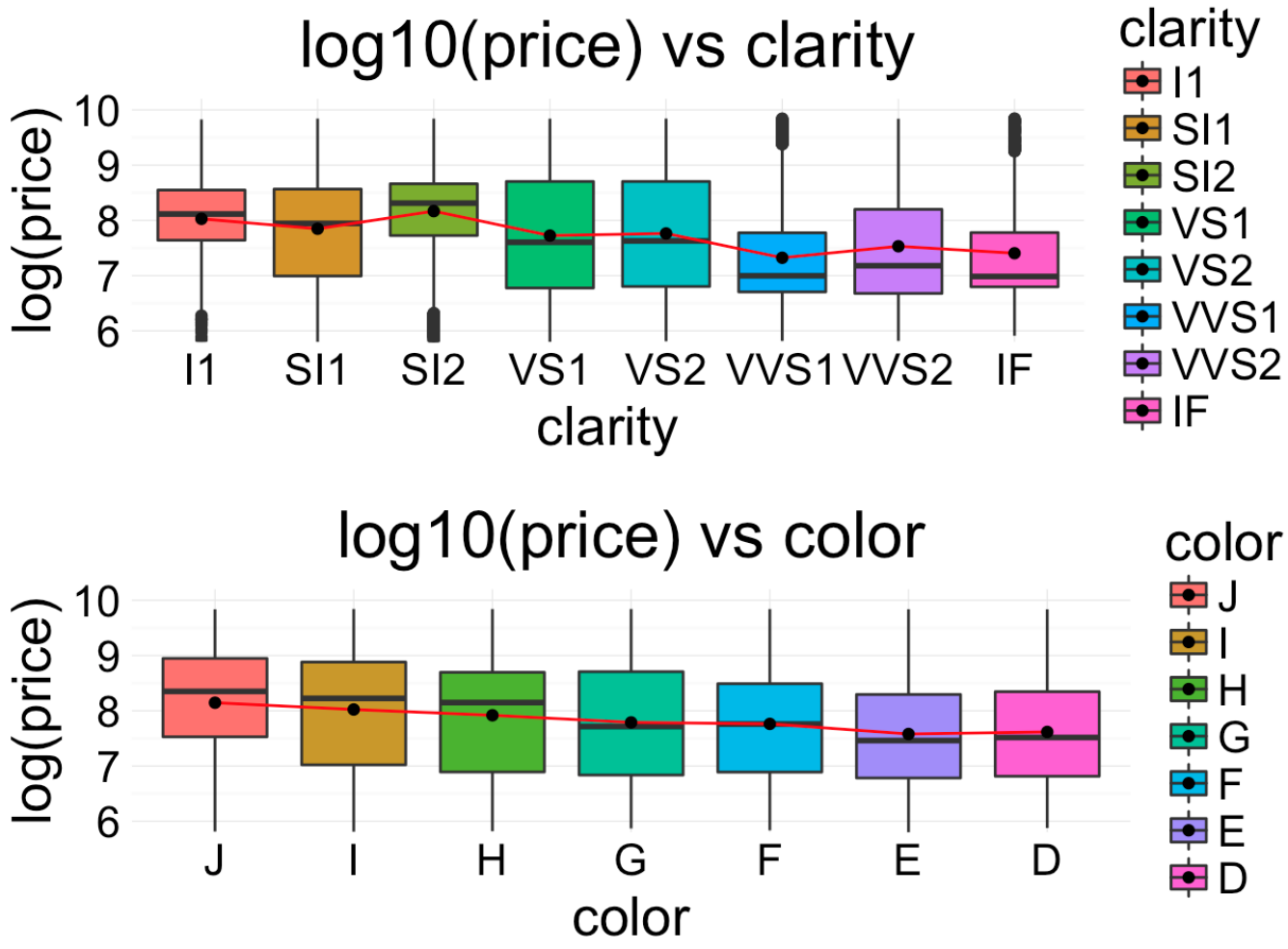


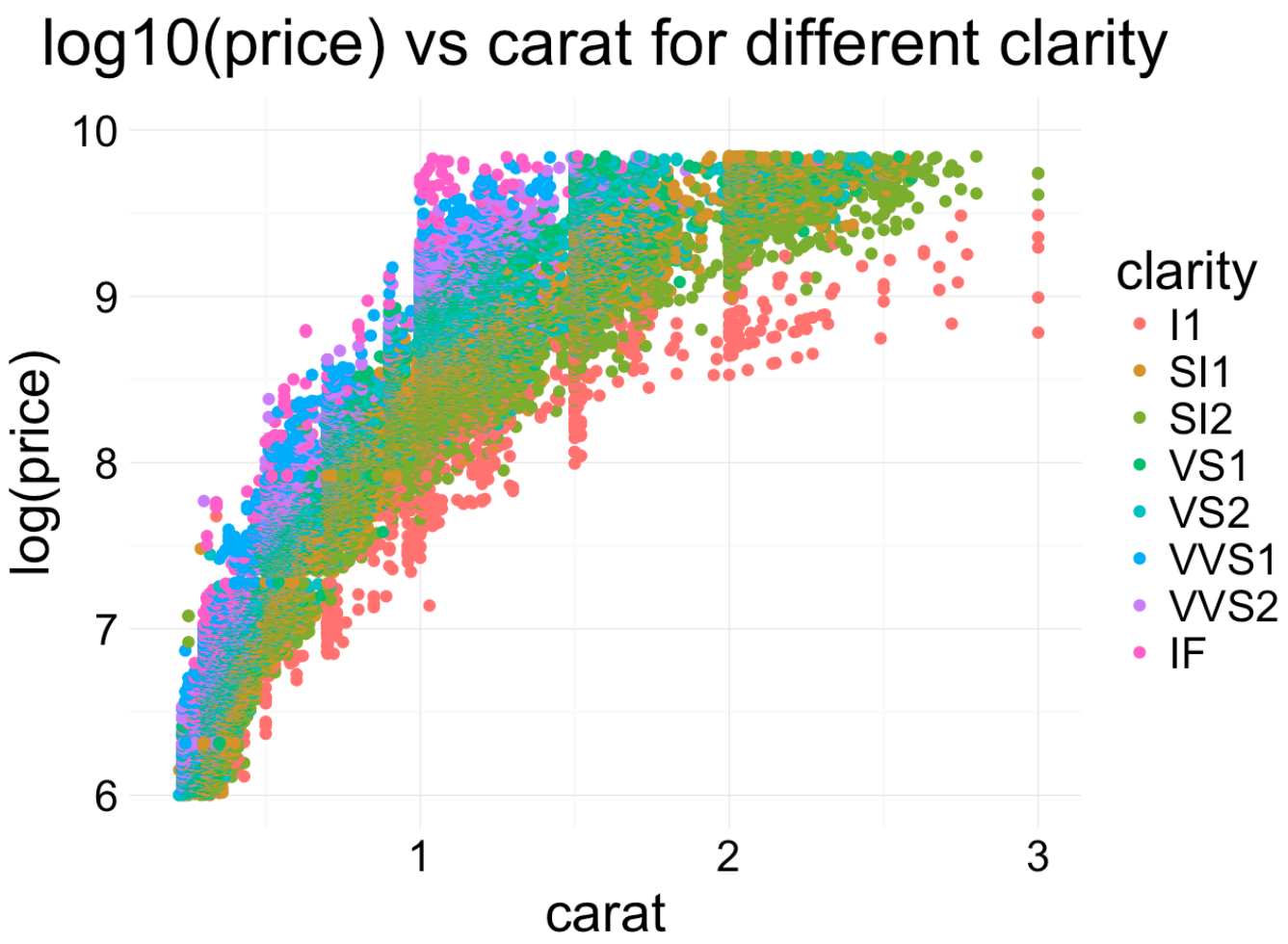
Fig 37. Price vs clarity and color

Description Two

Log of price vs clarity and color clearly indicate that for better clarity (IF), the diamonds are more expensive, and for better color (D) prices are higher. These 2 variables were included in building the final model.

Plot Three

```
## Warning: Removed 319 rows containing missing values (geom_point).
```




```
## Warning: Removed 319 rows containing missing values (geom_point).
```

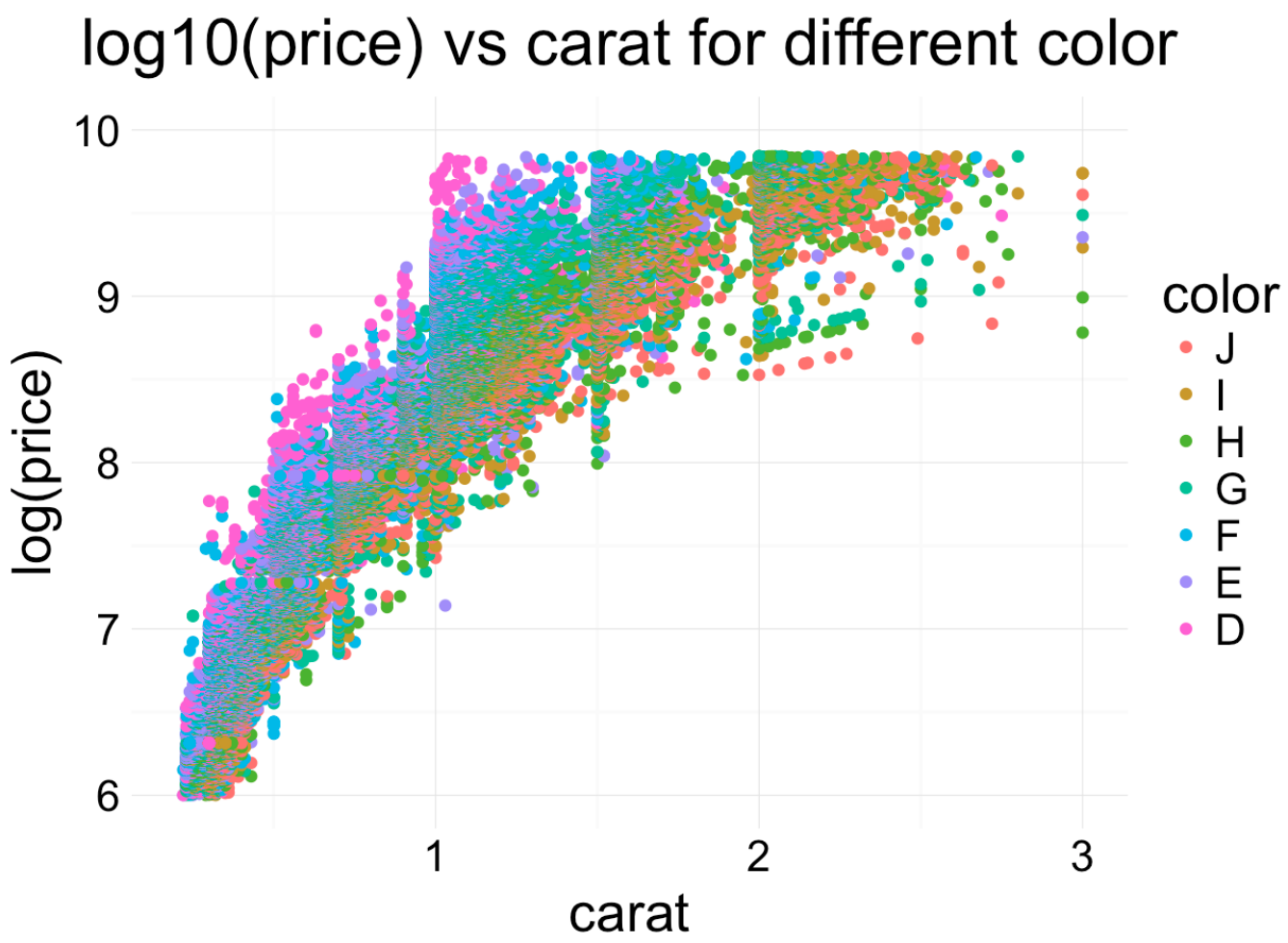


Fig 38. Price vs carat by clarity and color

Description Three

- For a given carat weight, diamonds with higher clarity levels (I1 is worst and IF is best) are almost always cheaper than diamonds with poorer clarity.
- For a given carat weight, diamonds with better color levels (J is worst and D is best) are almost always cheaper than diamonds with poorer color.

Reflection

The diamonds data set contains information on more than 54,000 thousand diamonds from around 2008. I started by exploring trends in individual variables. I found price variation to be bimodal, this may be reflection of the ammount of money people are comforatable spending on a ring. I noted very unique distribution of carat sizes. There were peaks in the distribution indicating preference for diamonds of certain sizes. Based on these values, I created a factor variable out of carat sizes. I descretized the carat sizes such that each bin had one unimodal distribution. Futher, the distribution of carat sizes seemed to have a right sided distribution around these prefered sizes, with the maximum being the bin containing the

prefered size. This indicated a preference for diamond size, because as a customer who wants 2 carat diamond is more likely to buy a 2.1 carat diamond than 1.9 carat. Manufacturers may be aware of these preferences and cut diamonds in certain predetermined sizes.

In a second step, I plotted price vs individual variables. I noted that the price values varied with carat, color and clarity. Surprisingly, diamonds of better cut had lower prices. This may be due to large differences in number of diamonds with premuim or ideal cut vs others. I therefore created 2 varaibles from carat, a variable indicating prefered carat size (carat.cut), and another quanitfying deviation from the prefered size. I also created 3 additional variables, table.cut,table.cut2 and cut.quality. First 2 were discretized values of table, and showed no predictive relation to price. The next variable was quality of cut, and was good for premium and ideal, and bad for others. This variable also did not show any predictive relation to price, and hence was dropped in further analysis. Based on my exploratory analysis, I concluded that carat, clarity and color are the most important variables. I decided to use these variables for my model. However, as I had made 3 varaiables from carat, I performed additional analysis to choose the right set of variables representing carats. I fit a linear model to approximate log(price) by using 3 different varibles representing carat,

- m1: log(price) vs carat, Rsq = 0.847
- m2: log(price) vs carat.cut, Rsq = 0.933
- m3: log(price) vs carat.cut + DistPrefered, Rsq = 0.935

I was surprised to see that carat.cut alone explained more than 93% of variance in the data. This indicated that the diamonds are priced based on predefined sizes they were cut for. I also wanted to include the effect of weight or deviation from desired size in the model, I therefore chose to use carat.cut and DistPrefered as carat variables. I split data into a training set (85%) and a test set and fit the model approximate price with 2 carat variables (carat.cut and DistPrefered), clarity and color. After fitting, the model explained more than 98% of the variance in the data. Although, the model predicted the prices well, one limitation is that the model does not account for time trends. Over years, the prices of diamonds have varied based on socioecomonic conditions that dictate supply/demand of prices. It will be nice to have a better dataset that had information about prices along with year information. It will be interesting to use this model to predict current diamond prices.