The Price of Diamonds

Ranfei(Jessie) Xu

2021-12-11

Abstract

This report is exploring the impact of different properties of diamonds on prices and tries to reduce the impact of the fact that higher-quality diamonds have lower weight in the analysis when doing the model regression. Overall, I found that weight is still the most important factor determining the price of diamonds. The price of natural diamonds is many times different from that produced in the laboratory. The pricing rules of diamonds of different shapes are roughly the same. The most popular round shape is somewhat above average. These conclusions may help consumers buy diamonds more rationally and avoid consumption traps.

Introduction

Buying diamonds represents a huge expenditure of money, the process is very time-consuming and can be frustrating, since there are so many aspects to consider. In this report, I will focus on how diamond level (clarity, color, cut), diamond-type (nature or lab), and diamond weight (carat) affect the prices.

First, I will use EDA to explore the relationship between predictors and outcomes, as well as the relationship between each predictor. And then, since the diamonds are divided into different shapes, I will build a multilevel model, and base on the findings from the EDA process, select the most suitable model for specific analysis and model evaluation and prediction. Finally, I will give some conclusions about diamond pricing.

This report may not only help you avoid consumption traps, make consumption decisions with a theoretical basis, and even provide ideas for understanding the complex logic of general product pricing.

Method

Data for Modeling

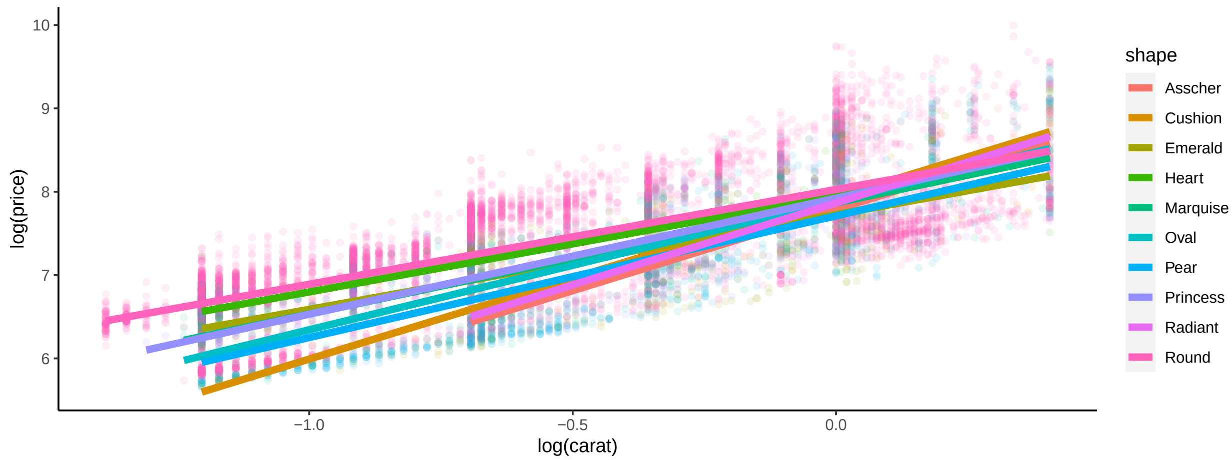
The clean data for regression including 10,062 observations and 10 variables, the specific data processing procedure is provided in the appendix.

Here are necessary explanations of important variables:

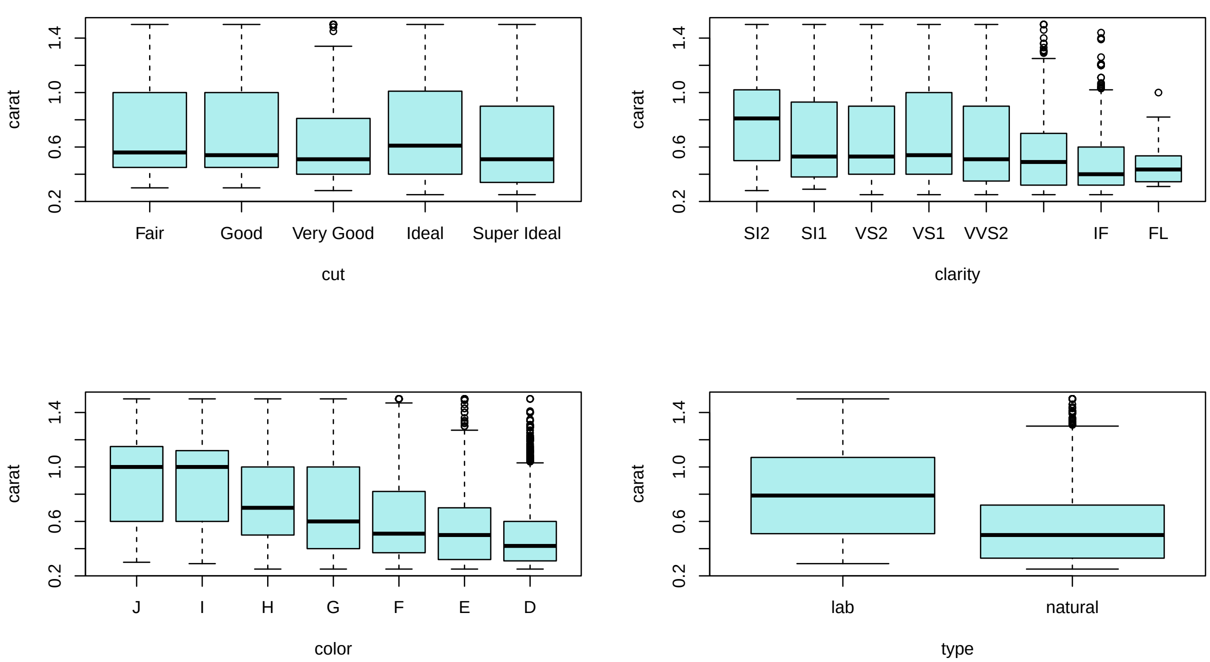
column names	explanation
cut	order of cut: Fair < Good < Very Good < Ideal < Super Ideal
clarity	order of clarity: SI2 < SI1 < VS2 < VS1 < VVS2 < VVS1 < IF < FL
color	order of color: J < I < H < G < F < E < D
shape	different shapes of diamonds
log_price	log(price), log of price in US dollars
log_carat	log(carat), log of weight of the diamond

Exploratory Data Analysis

EDA below helps to see if there is any correlation between some variables and price, which is instructive when I try to make a model regression to solve the problem that how some factors of diamonds affect their price.



The figure above shows the relationship between carat and price among different shapes. Different colors represent different shapes. It is obvious that with different shapes the slope and intercept are slightly different, which indicates the effect of shape is random. So, it's maybe appropriate to use a random effect for the model.



The figure above shows the relationship between carat and other variables. The obvious difference in the distribution of carat among different levels of `color`, `type`, and `clarity` indicates that it's necessary to add interaction terms in the model. For a more detailed density distribution plots, please refer to the violin diagram in appendix.

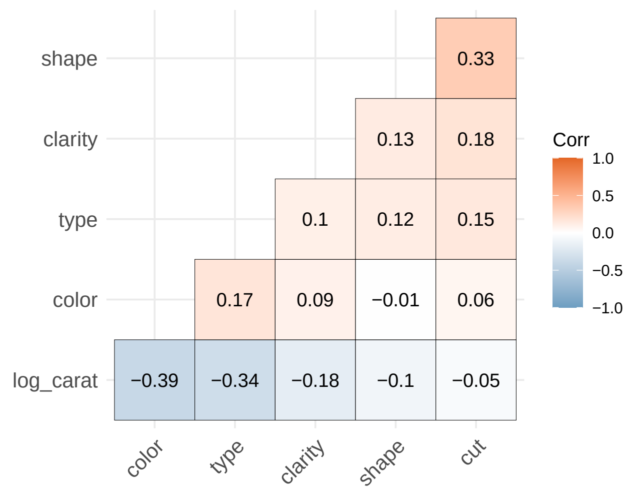


Figure 1: Detecting Correlation

The figure above is the correlation matrix, which displays the correlation coefficients between continuous variables. After we transfer the category variables to numeric type temporarily, we can see that the variables don't have a strong correlation, which ensures the rationality of our model to some extent.

Model Fitting

As the EDA shows above, considering that diamonds have different shapes, it's necessary to use a multilevel model and take shape as a random effect. This is reasonable, because different shapes of diamond have a different expenditure of time and material due to different polishing process, and the popularity of each shape is different, which ultimately affect the relationship between carat and price.

The interaction terms are also necessary to be inculed as the EDA shows above. Thinking realistically, a higher level of clarity and color meaning scarcity, and because of the scarcity of natural resources, it's inevitable that diamonds from natural or having a higher level of clarity or color are smaller in mass but more expensive than lab-produced and normal quality ones. I also tried to add the interaction term of `cut` and `carat` into the model, but the effect is not ideal. That's not surprising, as shown in EDA, there are only slight differences of carat among different cuts. Therefore, the interaction of cut and carat is not included in my final model.

Also, to match the evaluation plots to EDA plots, the model keeps using the log of carat as a predictor and the log of price as the outcome.

After comparing the results and validation of different choices of model, I found the model below is the most fitted one to the sample data. A detailed model selection process is displayed in the appendix.

Below is the final formula of the model:

```
model <- lmer(log_price ~ log_carat + cut + type + color + clarity
              + log_carat*clarity + log_carat*color  + log_carat*type
              + (1 + log_carat | shape), dt)
```

Result

To better interpret the model results, I transformed the category variables `clarity`(8 levels), `color`(7 levels), `cut`(5 levels) from factor type to numerical type. The coefficients don't change significantly after this transformation. The specific coefficient comparison is shown in the appendix.

Random Effect

	Intercept	log_carat
Asscher	-0.042994776	-0.014967941
Cushion	-0.060757032	-0.047508258
Emerald	-0.067008100	0.015462114
Heart	-0.053355903	-0.199597490
Marquise	0.014599556	0.008543657
Oval	0.066080857	0.059309911
Pear	0.003806297	-0.004359567
Princess	-0.032015902	0.090514037
Radiant	0.042811662	0.169333438
Round	0.128833342	-0.076729900

From the result of the random effect, we can see that the influence of different shapes is always not the same.

For `Round shape`, the intercept is 0.13, which is larger than others. I think it may be because 68% of the diamonds in our data are round shapes, which is reflected in its relatively large random influence on the model. At the same time, the random effect of round shape log_carat is -0.08, which also make sense, because most high-level `clarity` and `Color` diamonds, as well as those from `nature`, are round, and as I mentioned above, they are relatively small in mass, so the `round shape` has a negative impact on carats. The figure named "Density about shape" in the appendix displays the detail from a numeric aspect.

Fixed Effect

As the result of fixed effects shows below, all the variables are significant at alpha = 0.05 level.

	Estimate	Std. Error	df	t value	Pr(>
(Intercept)	6.632e+00	2.309e-02	1.310e+01	287.161	< 2e-16 ***
log_carat	1.478e+00	3.568e-02	9.632e+00	41.439	3.52e-12 ***
cut	4.069e-02	1.808e-03	1.004e+04	22.509	< 2e-16 ***
typenatural	9.963e-01	4.377e-03	1.005e+04	227.607	< 2e-16 ***
color	7.971e-02	1.156e-03	1.004e+04	68.969	< 2e-16 ***
clarity	8.365e-02	1.384e-03	1.004e+04	60.424	< 2e-16 ***
log_carat:clarity	1.926e-02	1.852e-03	1.004e+04	10.397	< 2e-16 ***
log_carat:color	1.575e-02	1.665e-03	1.004e+04	9.459	< 2e-16 ***
log_carat:typenatural	1.440e-01	6.826e-03	1.005e+04	21.097	< 2e-16 ***

An Example of Interpretation

For Round shape, we can conclude this formula:

$$log(price) = 6.75 + 1.40 \cdot log(carat) + 0.04 \cdot cut + 1.00 \cdot typenatural + 0.08 \cdot color + 0.08 \cdot clarity$$
$$+ 0.02 \cdot log\_carat * clarity + 0.02 \cdot log\_carat * color + 0.14 \cdot log\_carat * typenatural$$

When other predictors are held constant,

every 1% increase in the carat is associated with $(1.40 + 0.02 \cdot clarity + 0.02 \cdot color + 0.14 \cdot typenatural)\%$ increase in price on average. Since we converted the ordered variable to a numeric type, the lowest level is recorded as 0. Therefore, for each increase in the level, the corresponding numeric type increases by 1;

each grade of cut increases will multiply the price by about $e^{0.04}$; each grade of clarity increases will multiply the price by about $e^{0.08+0.02 \cdot ln(carat)}$, which means the change of price will be related to the specific carat;

each grade of color increases will multiply the price by about $e^{0.08+0.02 \cdot ln(carat)}$, which also means the change will be related to the specific carat;

and, the average price of a natural diamond is $e^{1+0.14 \cdot ln(carat)}$ times that of a laboratory diamond.

Model Validation

I check the model from numeric and graphic aspects, the good results indicate the model above fits well with the sample data. The detailed evaluation is displayed in the appendix.

Discussion

Conclusion

According to the random effects table, the shape has a very minimal effect, which makes me speculate that it maybe a consumer trap when a seller claims that a diamond of a particular shape (like clover shape) will cost significantly more than a round or oval diamond because of its craftsmanship.

For the fixed table, although the coefficient of `clarity`, `color`, and `cut` is very small, its influence on the price increases by multiple, so its power should not be underestimated when pricing a diamond.

I think the results above balance the brand effect to some extent and can promote rational consumption of consumers.

For further improvement

Deal with confounding variable: There's always been a common discussion about the pricing of diamonds, which is the weight of the diamond seems to be the only factor that determines the price. But in fact, high-quality natural diamonds are often of low weight. I tried to solve the problem caused by the confounding variable `carat` to some extent by adding interaction, but the regression results did not seem to show the effect of quality on weight, so the suspicion that lower-quality diamonds have higher prices was not solved.

Attempt more data and models: Because the data for regression was sifted through the original data set, it would be overgeneralizing to predict a pricing pattern of diamonds with just over 10,000 data. And although my residuals are good, I am somewhat expecting to see some outliers, such as those with large residuals, which means that the price of these diamonds may be too low or too high than expected, which can even provide us with the opportunity to buy the diamonds that have been priced low incorrectly.

Citations/Sources

- Data Source
  - https://www.kaggle.com/miguelcorraljr/brilliant-diamonds/
- Discussion about Diamonds
  - https://www.laurenbjewelry.com/blog/round-diamonds-expensive-shapes/
  - https://www.bluenile.com/education/diamonds/shape
- Mixed Effect Model Interpretation and Model Checking
  - https://www.ssc.wisc.edu/sscc/pubs/MM/MM_TestEffects.html#testing-mixed-models-parameters
  - https://cran.microsoft.com/snapshot/2017-08-01/web/packages/sjPlot/vignettes/sjplmer.html
  - https://vitalflux.com/fixed-vs-random-vs-mixed-effects-models-examples/

Appendix

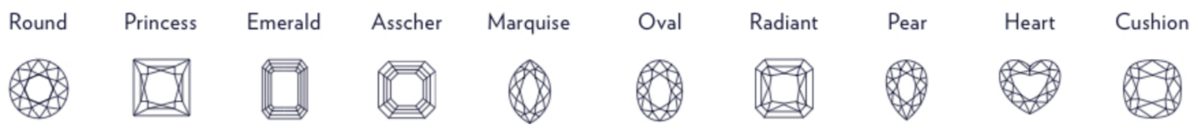


Figure 2: Shape of Diamonds

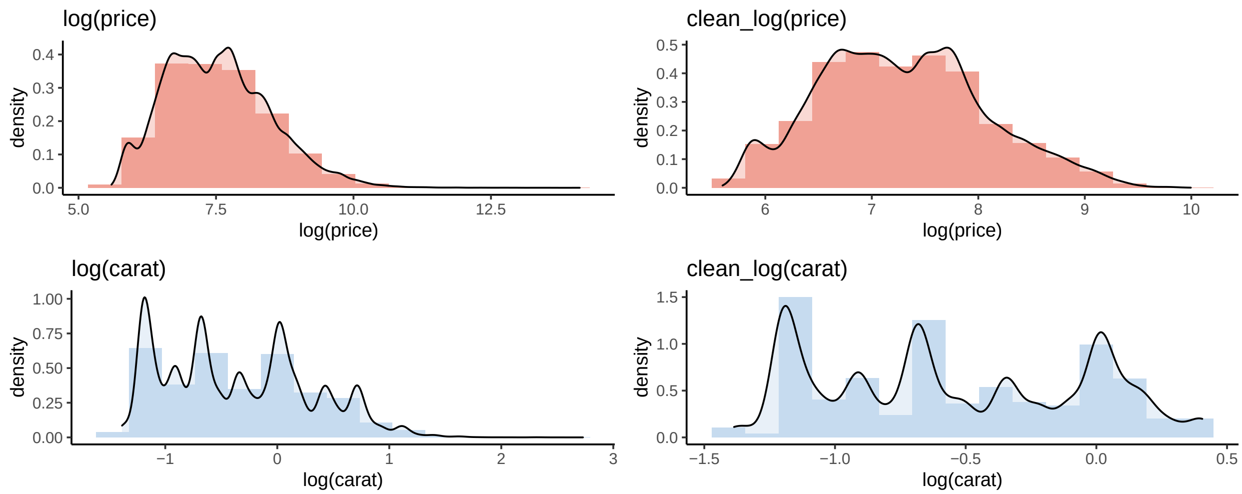
Data Cleaning and Processing

The main data set is published on Kaggle: Brilliant Diamonds-Ultimate Dataset.

Below is the summary of raw data

```
##       shape           price              carat                    cut
##  Round   :76080   Min.   :    270   Min.   : 0.2500   Fair       :  334
##  Oval    :12978   1st Qu.:    900   1st Qu.: 0.4000   Good       : 3398
##  Pear    : 9221   Median :   1770   Median : 0.7000   Very Good  :20914
##  Emerald : 6750   Mean   :   3287   Mean   : 0.8842   Ideal      :39417
##  Princess: 5135   3rd Qu.:   3490   3rd Qu.: 1.1000   Super Ideal:55244
##  Cushion : 4279   Max.   :1348720   Max.   :15.3200
##  (Other) : 4864
##  color        clarity       report           type
##  J: 9012   VS1    :27259   GCAL: 5782   lab    :48994
##  I:14409   VS2    :26440   GIA :68782   natural:70313
##  H:12868   SI1    :21205   HRD : 1153
##  G:17559   VVS2   :17229   IGI :43590
##  F:19833   SI2    :12771
##  E:24730   VVS1   :10534
##  D:20896   (Other): 3869
```

I got 119307 observations from the raw data set. As I see the density and distribution of `price` and `carat`, I find that they have a long tail, which indicates they have a large range. Therefore, I exclude the extreme data that deviates a lot from the median. Because there are only 18669/119307 = 16% diamonds are over 1.5 carat, and only 1338/119307 = 1.12% diamonds are over e^10 = 22026 dollars. So I narrow down the raw data by adding filter carat <= 1.5 & price <= 22026. Besides, I do a log transformation for carat and price variables and mutate them as new columns.

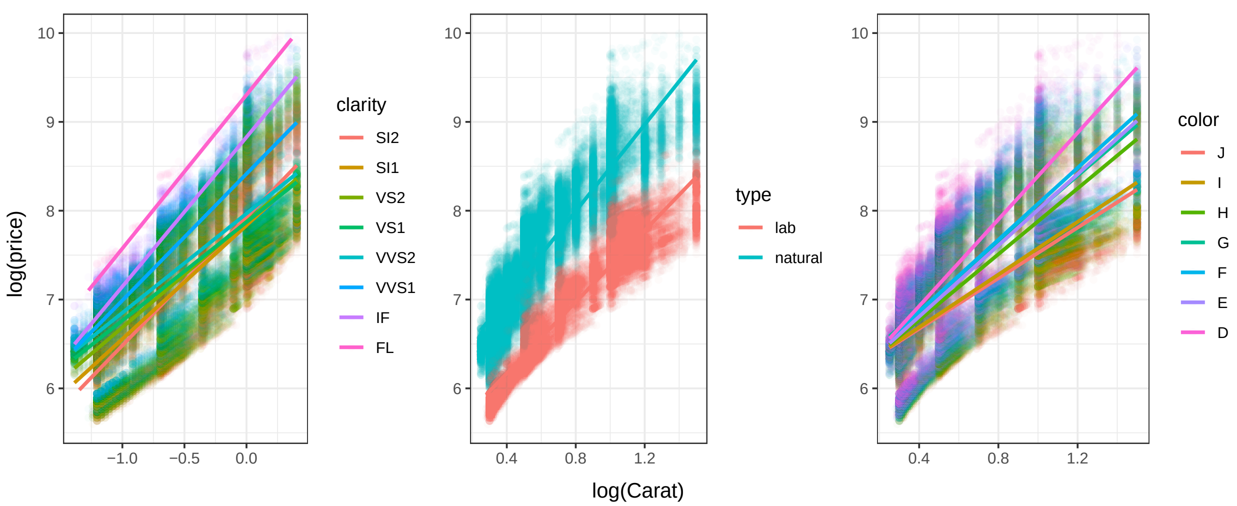


According to the figure above, the data after cleaning is more uniform and neat. Some outliers can be ruled out, such as one natural diamond in the dataset which costs $1,348,720, and weighs 11.41 carat with the highest quality in color, cut, and clarity.

More EDA

To get better figures, I narrow down the size of data for EDA and modeling by selecting randomly. As a result, there are about 10,000 data in the dataset for EDA.

The following plots indicate the different relationships between carats and prices for different grades of diamonds from `color`, `type`, and `clarity` aspects.



The figure above shows the different effects of different levels within a variables indicates it's necessary to add interaction of them to carat.

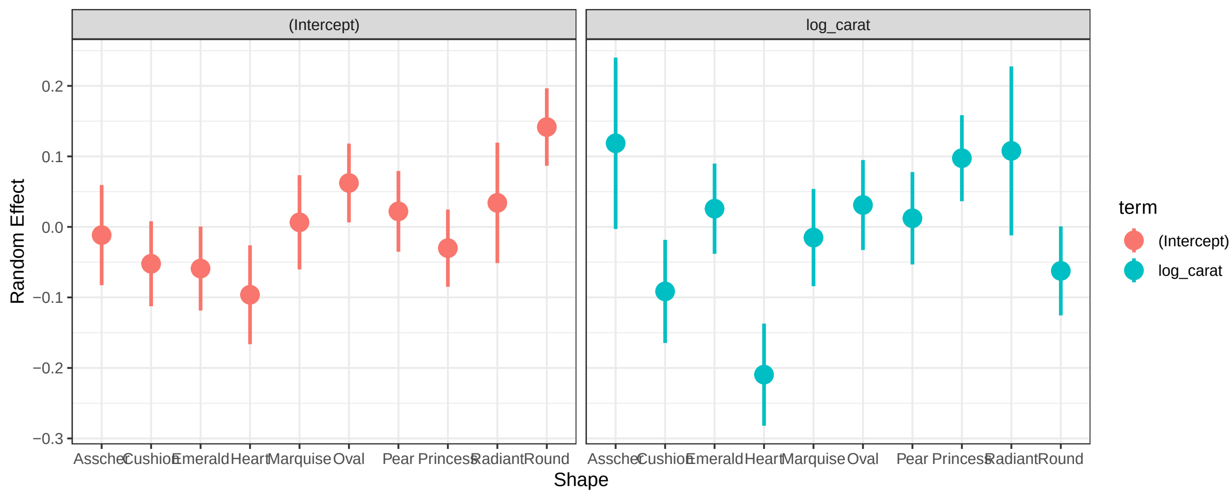
Model Selection

```
## Data: dt
## Models:
## fit2: log_price ~ log_carat + cut + type + (1 + log_carat | color) + (1 + log_carat | clarity) + (1
## fit1: log_price ~ log_carat + cut + type + color + clarity + (1 | shape)
## fit4: log_price ~ log_carat + cut + type + color + clarity + log_carat * clarity + (1 + log_carat | s
```

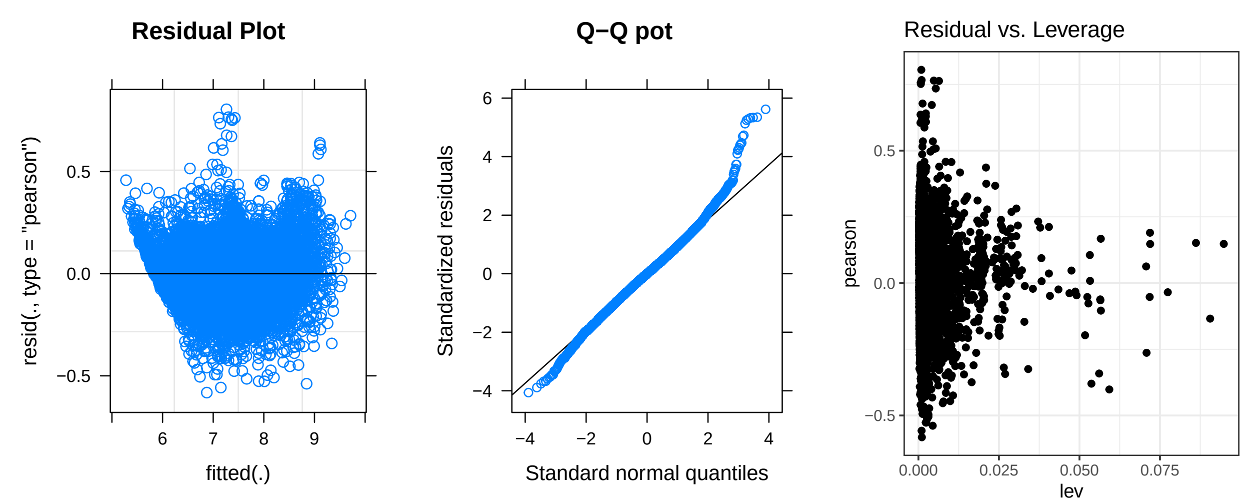
```
## fit3: log_price ~ log_carat + cut + type + color + clarity + log_carat * clarity + color * log_carat
## fit5: log_price ~ log_carat + cut + type + color + clarity + log_carat * clarity + color * log_carat
## fit6: log_price ~ log_carat + cut + type + color + clarity + log_carat * clarity + color * log_carat
##      npar    AIC    BIC logLik deviance   Chisq Df Pr(>Chisq)
## fit2   15 -10709 -10601 5369.5   -10739
## fit1   22 -10741 -10582 5392.4   -10785  45.806  7  9.533e-08 ***
## fit4   31 -11027 -10803 5544.3   -11089 303.780  9  < 2.2e-16 ***
## fit3   35 -10816 -10564 5443.2   -10886   0.000  4          1
## fit5   37 -11038 -10771 5556.0   -11112 225.596  2  < 2.2e-16 ***
## fit6   38 -11462 -11187 5768.7   -11538 425.504  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The smaller AIC always indicate better model.

Visualization of random effect



Eveluation



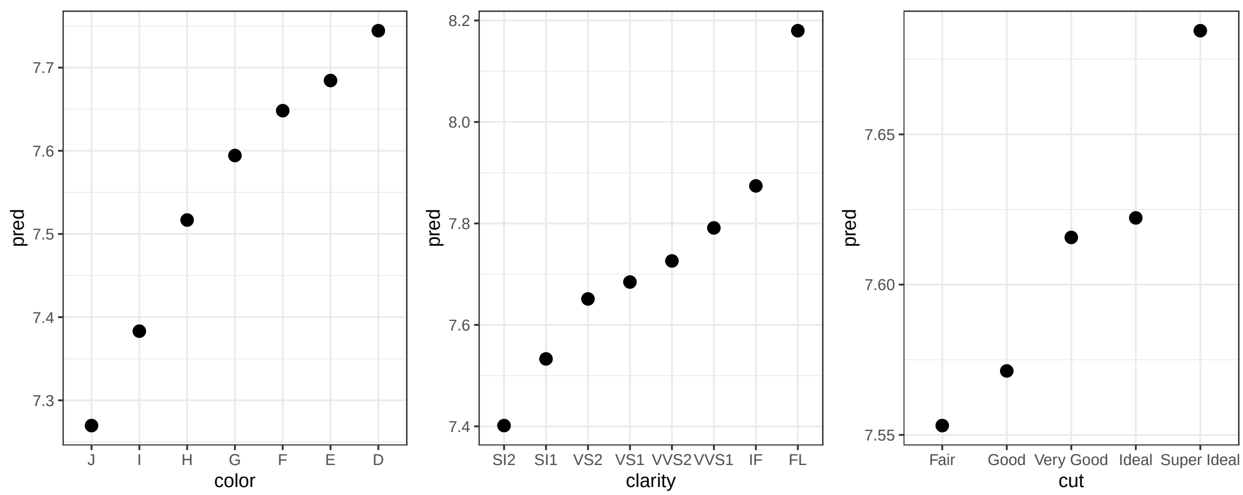
Graphically, from the *Normal Q-Q Plot*, we can tell that all points roughly fall into a straight line which indicates the assumption of normality is satisfied. From the* Residual plot_, we can tell that the residuals are evenly distributed around the center line and there are no particular patterns or clusters which indicates our model has a good fit. From the_ Leverage plot*, we can tell that there is no obvious leverage point.

```
## # Indices of model performance
##
## AIC        |        BIC | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma
## -------------------------------------------------------------------------
## -10360.997 | -10267.182 |      0.967 |      0.953 | 0.298 | 0.143 | 0.143
```

Numerically: The Conditional R^2 and Marginal R^2 is close to 1, including other nice output of checking above that indicate our model fit well.

Prediction

To visualize the model by drawing predicted values, I generate an evenly spaced grid of points from the data.



Full Results

Random effects of model

```
## $shape
##           (Intercept)   log_carat
## Asscher  -0.019355655  0.10714042
## Cushion  -0.053857653 -0.09182083
## Emerald  -0.059678746  0.02580676
## Heart    -0.097554204 -0.20945442
## Marquise  0.004495284 -0.01459524
## Oval      0.061310805  0.03106987
## Pear      0.021063476  0.01095170
## Princess -0.030181152  0.09669937
## Radiant   0.034004417  0.10806862
## Round     0.139753428 -0.06386625
##
## with conditional variances for "shape"
```

Fixed effects of model

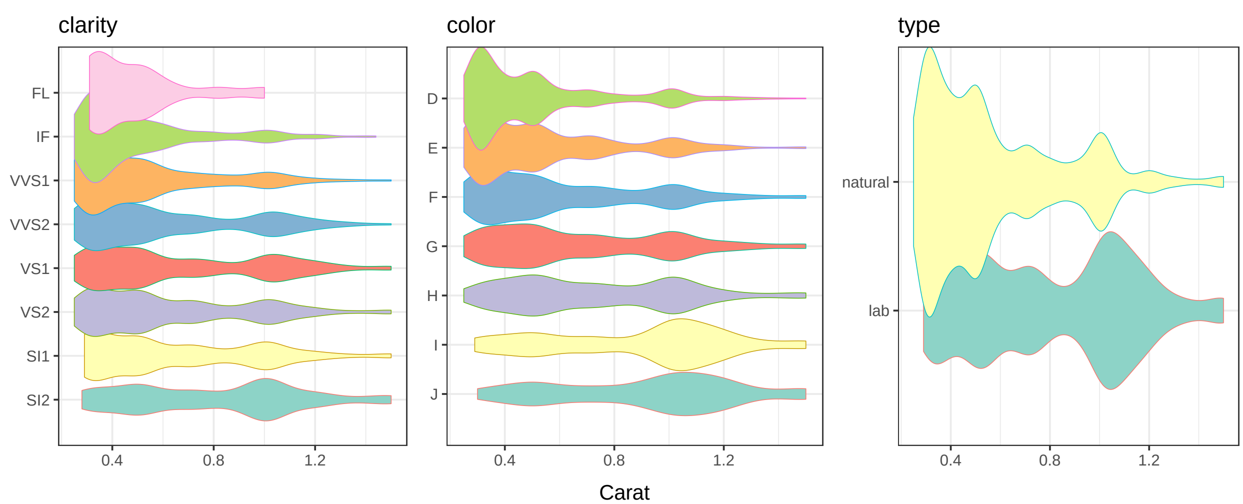
```
##           (Intercept)            log_carat                  cut
##            6.62981720           1.48044742           0.03859983
```

```
##          typenatural                color              clarity
##           1.01131321           0.07756679           0.08446387
##    log_carat:clarity      log_carat:color log_carat:typenatural
##           0.02025438           0.01284346           0.14568779
```

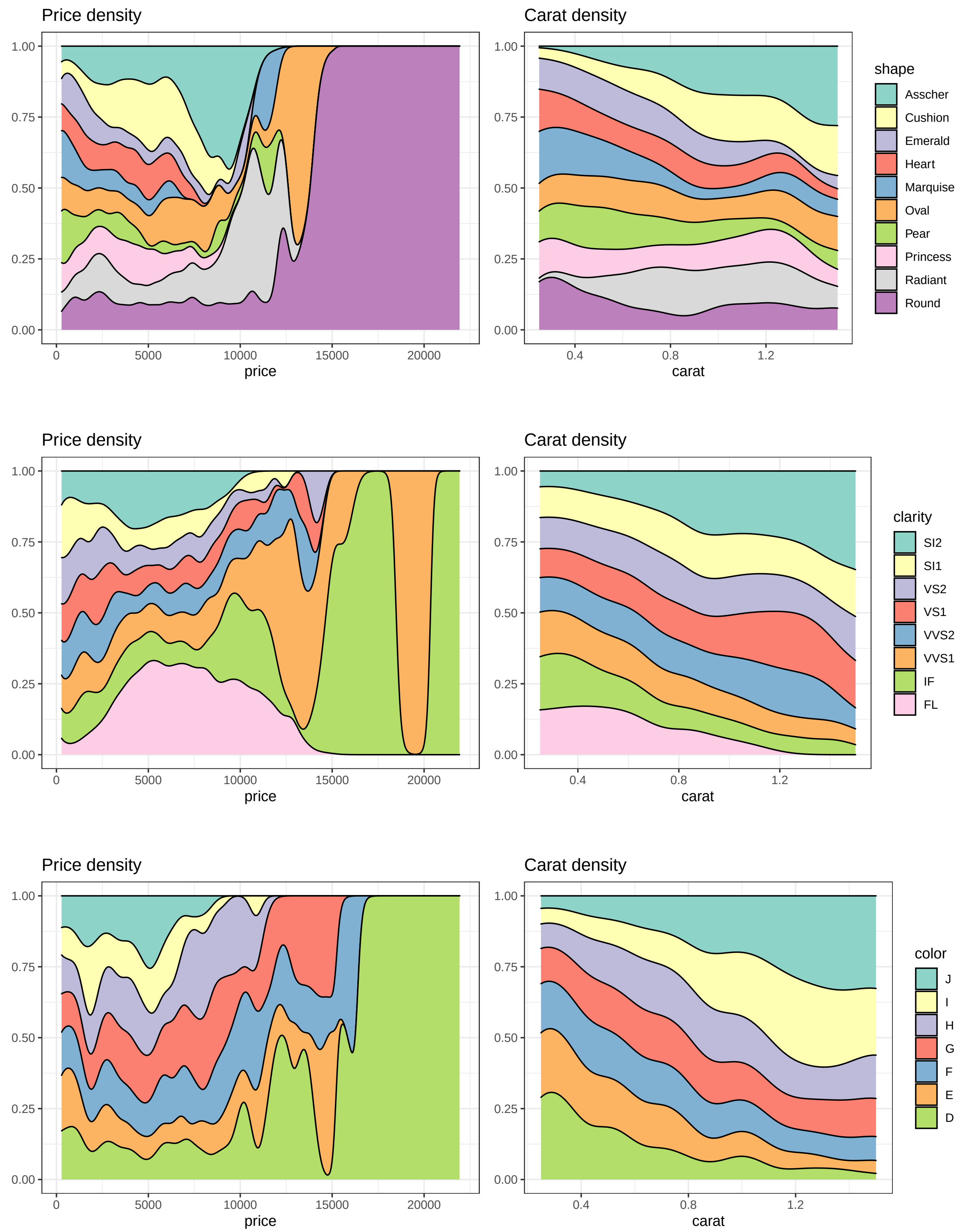
Coefficients of model

```
## $shape
##          (Intercept) log_carat        cut typenatural      color    clarity
## Asscher     6.610462  1.587588 0.03859983    1.011313 0.07756679 0.08446387
## Cushion     6.575960  1.388627 0.03859983    1.011313 0.07756679 0.08446387
## Emerald     6.570138  1.506254 0.03859983    1.011313 0.07756679 0.08446387
## Heart       6.532263  1.270993 0.03859983    1.011313 0.07756679 0.08446387
## Marquise    6.634312  1.465852 0.03859983    1.011313 0.07756679 0.08446387
## Oval        6.691128  1.511517 0.03859983    1.011313 0.07756679 0.08446387
## Pear        6.650881  1.491399 0.03859983    1.011313 0.07756679 0.08446387
## Princess    6.599636  1.577147 0.03859983    1.011313 0.07756679 0.08446387
## Radiant     6.663822  1.588516 0.03859983    1.011313 0.07756679 0.08446387
## Round       6.769571  1.416581 0.03859983    1.011313 0.07756679 0.08446387
##          log_carat:clarity log_carat:color log_carat:typenatural
## Asscher         0.02025438      0.01284346             0.1456878
## Cushion         0.02025438      0.01284346             0.1456878
## Emerald         0.02025438      0.01284346             0.1456878
## Heart           0.02025438      0.01284346             0.1456878
## Marquise        0.02025438      0.01284346             0.1456878
## Oval            0.02025438      0.01284346             0.1456878
## Pear            0.02025438      0.01284346             0.1456878
## Princess        0.02025438      0.01284346             0.1456878
## Radiant         0.02025438      0.01284346             0.1456878
## Round           0.02025438      0.01284346             0.1456878
##
## attr(,"class")
## [1] "coef.mer"
```

Other Plots



Violin plot describes the distribution of the data. The plots above indicate that the data is unevenly distributed between different levels of `color`, `clarity`, and `type`.



The figure above shows that the specific distribution of diamonds and the price of different categorical predictors. Uneven distribution tends to bring confounding problems.

Comparison of the results of model with different attributes predictors.

```
##
## ======================================================================
##                                   numeric(predictor)  factor(predictor)
## ----------------------------------------------------------------------
## (Intercept)                            6.63 ***           6.75 ***
##                                       (0.02)             (0.03)
## log_carat                              1.48 ***           1.57 ***
##                                       (0.04)             (0.04)
## cut                                    0.04 ***
##                                       (0.00)
## typenatural                            1.01 ***           1.01 ***
##                                       (0.00)             (0.00)
## color                                  0.08 ***
##                                       (0.00)
## clarity                                0.08 ***
##                                       (0.00)
## log_carat:clarity                      0.02 ***
##                                       (0.00)
## log_carat:color                        0.01 ***
##                                       (0.00)
## log_carat:typenatural                  0.15 ***           0.14 ***
##                                       (0.01)             (0.01)
## cutGood                                                   0.02
##                                                          (0.03)
## cutVery Good                                              0.06 *
##                                                          (0.02)
## cutIdeal                                                  0.07 **
##                                                          (0.02)
## cutSuper Ideal                                            0.13 ***
##                                                          (0.02)
## colorI                                                    0.12 ***
##                                                          (0.01)
## colorH                                                    0.25 ***
##                                                          (0.01)
## colorG                                                    0.34 ***
##                                                          (0.01)
## colorF                                                    0.39 ***
##                                                          (0.01)
## colorE                                                    0.42 ***
##                                                          (0.01)
## colorD                                                    0.48 ***
##                                                          (0.01)
## claritySI1                                                0.15 ***
##                                                          (0.01)
## clarityVS2                                                0.26 ***
##                                                          (0.01)
## clarityVS1                                                0.31 ***
##                                                          (0.01)
## clarityVVS2                                               0.36 ***
##                                                          (0.01)
## clarityVVS1                                               0.44 ***
```

```
##                                                              (0.01)
## clarityIF                                                     0.52 ***
##                                                              (0.02)
## clarityFL                                                     0.77 ***
##                                                              (0.09)
## log_carat:claritySI1                                          0.03 **
##                                                              (0.01)
## log_carat:clarityVS2                                          0.02 *
##                                                              (0.01)
## log_carat:clarityVS1                                          0.04 ***
##                                                              (0.01)
## log_carat:clarityVVS2                                         0.05 ***
##                                                              (0.01)
## log_carat:clarityVVS1                                         0.08 ***
##                                                              (0.01)
## log_carat:clarityIF                                           0.08 ***
##                                                              (0.02)
## log_carat:clarityFL                                          -0.02
##                                                              (0.10)
## log_carat:colorI                                              0.01
##                                                              (0.02)
## log_carat:colorH                                              0.00
##                                                              (0.02)
## log_carat:colorG                                              0.02
##                                                              (0.02)
## log_carat:colorF                                              0.02
##                                                              (0.02)
## log_carat:colorE                                              0.01
##                                                              (0.02)
## log_carat:colorD                                              0.00
##                                                              (0.02)
## ---------------------------------------------------------------------------
## AIC                                -10361.00          -11197.13
## BIC                                -10267.18          -10922.90
## Log Likelihood                       5193.50            5636.57
## Num. obs.                           10062              10062
## Num. groups: shape                     10                 10
## Var: shape (Intercept)                  0.00               0.01
## Var: shape log_carat                    0.01               0.01
## Cov: shape (Intercept) log_carat        0.00               0.00
## Var: Residual                           0.02               0.02
## ===========================================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Citations

- EDA
  - https://bootstrappers.umassmed.edu/bootstrappers-courses/pastCourses/rCourse_2016-04/Additional_Resources/Rcolorstyle.html#heat.colors
  - https://www.r-graph-gallery.com/135-stacked-density-graph.html (density plot)
  - https://www.google.com/search?q=how+to+interpret+ggcorrplot&oq=how+to+interpret+ggcorrplot&aqs=chrome..69i57j0i13.2887j0j9&sourceid=chrome&ie=UTF-8#kpvalbx=_bVKwYYjIArCcptQPxZO30Ak43 (correlation plot)

– https://www.r-graph-gallery.com/

Supplement

This part includes all the codes uisng in the main part:

```
knitr::opts_chunk$set(echo = FALSE,warning = FALSE, message = FALSE,fig.height=4, fig.width=10,fig.align
pacman::p_load(ggplot2, knitr, arm, data.table, foreign, gridExtra, car, stringr, rstan, rstanarm, zoo,
Sys.setenv(LANGUAGE = "en")

dt0 <- read.csv("diamonds_dataset.csv",header=T,sep=",")
dt0 <- dt0[, c("shape","price","carat","cut","color","clarity","report","type")]
dt0 <- as.data.frame(dt0)
dt0$shape <- as.factor(dt0$shape)
dt0$price <- as.numeric(dt0$price)
dt0$carat <- as.numeric(dt0$carat)
dt0$cut <- factor(dt0$cut, levels =c("Fair","Good","Very Good","Ideal","Super Ideal") )
dt0$color <- factor(dt0$color, levels = c("J","I","H","G","F","E","D"))
dt0$clarity <- factor(dt0$clarity, levels = c("SI2","SI1","VS2","VS1","VVS2","VVS1","IF","FL"))
dt0$report <- as.factor(dt0$report)
dt0$type <- as.factor(dt0$type)

dt1 <- dt0 %>%
  filter(carat <= 1.5) %>%
  filter(price <= 22026)  %>%
  mutate(log_price = log(price), log_carat = log(carat))

smp_size <- floor(0.1 * nrow(dt1))
sam_ind <- sample(seq_len(nrow(dt1)), size = smp_size)
dt <- dt1[sam_ind, ]

ggplot(data = dt)+
  aes(log(carat),log(price))+
  geom_point(aes(color = shape),alpha = 0.1)+
  labs(title="",x="log(carat)",y="log(price)")+
  geom_smooth(aes(color = shape),method = "lm",se=F,size=2)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = elemen

par(mfrow = c(2, 2))
boxplot(data=dt, carat ~ cut, col = "paleturquoise")
boxplot(data=dt, carat ~ clarity, col = "paleturquoise")
boxplot(data=dt, carat ~ color, col = "paleturquoise")
boxplot(data=dt, carat ~ type, col = "paleturquoise")

dt_cor <- dt[, c('log_carat','shape','cut','clarity','color','type')]
dt_cor$shape <- as.numeric(dt_cor$shape)
dt_cor$log_carat <- as.numeric(dt_cor$log_carat)
dt_cor$cut <- as.numeric(dt_cor$cut)
dt_cor$color <- as.numeric(dt_cor$color)
dt_cor$clarity <- as.numeric(dt_cor$clarity)
dt_cor$type <- as.numeric(dt_cor$type)
library(ggcorrplot)
ggcorrplot(cor(dt_cor), hc.order = TRUE, type = "lower",
   outline.col = "black",
```

```
    # ggtheme = ggplot2::theme_gray,
    colors = c("#6D9EC1", "white", "#E46726"),
    lab = TRUE)

library(lmerTest)
dt_num <- dt
dt_num$log_carat <- as.numeric(dt_num$log_carat)
dt_num$cut       <- as.numeric(dt_num$cut)
dt_num$color     <- as.numeric(dt_num$color)
dt_num$clarity   <- as.numeric(dt_num$clarity)

fit <- lmer(log_price ~ log_carat+cut+type+color+clarity+log_carat*clarity+color*log_carat +type*log_ca
summary(fit)

fit0 <- lmer(log_price ~ log_carat+cut+type+color+clarity+log_carat*clarity+color*log_carat +type*log_c
fitlm <- lm(log_price ~ log_carat+cut+type+color+clarity+log_carat*clarity+color*log_carat +type*log_ca
```