STAT394 Group Project Milestone 4

Ken MacIver, Tom Tribe, Jundi Yang, Mei Huang

2022-09-09

Contents


```
# load the required packages
require(ggplot2)
require(ggthemes)
library(ggstance)
library(ggcorrplot)
library(ggplot2)
library(mvtnorm)
library(fitdistrplus)
library(GGally)
library(ggExtra)
library(reshape2)
library(xtable)
library(moments)
```

```
library(psych)
library(Hotelling)
library(car)
library(HDtest)
library(ggpubr)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
```

1 The 'diamonds' dataset

NOTE: The size of this dataset means that rendering to PDF takes a long time.

For the STAT394 Group Project, Group 11 have chosen a dataset called 'diamonds', which presents data on 53940 diamonds. It was accessed it from kaggle.com. There are ten variables:

- carat: the diamond's weight (numerical: 0.2 - 5.01)
- cut: a measure of quality (categorical: Fair, Good, Very Good, Premium)

- color: a measure of colour quality (categorical: J, which is poorest quality, to D, which is best)
- clarity: a measure of clearness (categorical: from worst to best = I1, SI2, VS2, VS1, VVS2, IF)
- x: length in mm (0 - 10.74)
- y: width in mm (0 - 58.9)
- z: depth in mm (0 - 31.8)
- depth: total depth percentage = z/mean(x,y) = 2*z/(x+y) (43 - 79)
- table: width of top of diamond relative to widest point
- price: the price of the diamond in US dollars (List adapted from the list at "Diamonds Dataset, Kaggle.com" (2016)).

We are most interested in how these variables relate to and predict diamond price.

1.0.1 Load the dataset into R.

```
##   carat     cut color clarity depth table price    x    y    z
## 1  0.23   Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21 Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23    Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29 Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31    Good     J     SI2  63.3    58   335 4.34 4.35 2.75
```

```
## '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    : chr  "Ideal" "Premium" "Good" "Premium" ...
##  $ color  : chr  "E" "E" "E" "I" ...
##  $ clarity: chr  "SI2" "SI1" "VS1" "VS2" ...
```

```
##  $ 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 ...
```

1.1 Create factor levels and view summary

```
##   carat depth table price    x    y    z
## 1  0.23  61.5    55   326 3.95 3.98 2.43
## 2  0.21  59.8    61   326 3.89 3.84 2.31
## 3  0.23  56.9    65   327 4.05 4.07 2.31
## 4  0.29  62.4    58   334 4.20 4.23 2.63
```

```
##      carat             depth           table           price      
##  Min.   :0.2000   Min.   :43.00   Min.   :43.00   Min.   :  326  
##  1st Qu.:0.4000   1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950  
##  Median :0.7000   Median :61.80   Median :57.00   Median : 2401  
##  Mean   :0.7979   Mean   :61.75   Mean   :57.46   Mean   : 3933  
##  3rd Qu.:1.0400   3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324  
##  Max.   :5.0100   Max.   :79.00   Max.   :95.00   Max.   :18823  
##        x                y               z          
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
##  Median : 5.700   Median : 5.710   Median : 3.530  
##  Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
##  3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
##  Max.   :10.740   Max.   :58.900   Max.   :31.800
```

1.1.0.1 Summary function diamonds

1.1.0.2 Change array rownames

1.1.1 Summary table diamonds

```
# produce summary table
knitr::kable(summ_diamonds,
             digits = 3,
             caption = "Summary statistics for 'diamonds' (3 s.f.)")
```

Table 1: Summary statistics for 'diamonds' (3 s.f.)

	carat	depth	table	price	x	y	z
sample size	53940.000	53940.000	53940.000	53940.000	53940.000	53940.000	53940.000
minimum	0.200	43.000	43.000	326.000	0.000	0.000	0.000
first quartile	0.400	61.000	56.000	950.000	4.710	4.720	2.910
median	0.700	61.800	57.000	2401.000	5.700	5.710	3.530
mean	0.798	61.749	57.457	3932.800	5.731	5.735	3.539
third quartile	1.040	62.500	59.000	5324.250	6.540	6.540	4.040
maximum	5.010	79.000	95.000	18823.000	10.740	58.900	31.800
IQR	0.640	1.500	3.000	4374.250	1.830	1.820	1.130
standard deviation	0.474	1.433	2.234	3989.440	1.122	1.142	0.706
skewness	1.117	-0.082	0.797	1.618	0.379	2.434	1.522
kurtosis	4.256	8.739	5.801	5.177	2.382	94.206	50.082

Table 1 presents the summary statistics for the numerical variables in the diamonds dataset.

1.1.2 Means vector

```
##        [,1]  [,2]  [,3] [,4]  [,5]  [,6]  [,7]
## [1,] 0.7979 61.75 57.46 3933 5.731 5.735 3.539
```

The estimates for the means vector are displayed in both the output above and in vector form below:

$$\hat{\mu} = \begin{pmatrix} 0.7979 \\ 61.75 \\ 57.46 \\ 3933 \\ 5.731 \\ 5.735 \\ 3.539 \end{pmatrix}$$

1.2 Multivariate Tests

Our categorical variables of cut, clarity and color are all ordinal:

- cut: a measure of quality (categorical: Fair, Good, Very Good, Premium)
- color: a measure of colour quality (categorical: J, which is poorest quality, to D, which is best)
- clarity: a measure of clearness (categorical: from worst to best = I1, SI2, VS2, VS1, VVS2, IF)

Using a Hotelling's T-test we will test the equality of mean vectors of the lowest and highest level of each categorical variable. We might expect the diamonds with lower quality cut, color and clarity to differ significantly from diamonds with high quality cut, colour and clarity.

```
(hotelling.test(subset(diamonds, cut == "Fair")[,-(2:4)], subset(diamonds, cut == "Premi
```

```
## Test stat:  6349.2
## Numerator df:  7
## Denominator df:  15393
## P-value:  0
```

The output above comparing the mean vectors of the 'Fair' and 'Premium' levels of the 'cut' variable gives a test statistic of 6349.2 (which is huge) and a p-value of 0. We therefore reject the null hypothesis that the mean vectors of these two samples are equal.

```
(hotelling.test(subset(diamonds, color == "J")[,-(2:4)], subset(diamonds, color == "D")[
```

```
## Test stat:  5602.4
## Numerator df:  7
## Denominator df:  9575
## P-value:  0
```

The output above comparing the mean vectors of the 'J' (poorest quality) and 'D' (best quality) levels of the 'color' variable gives a test statistic of 5602.4 and a p-value of 0. We therefore reject the null hypothesis that the mean vectors of these two levels are equal.

```
(hotelling.test(subset(diamonds, clarity == "I1")[,-(2:4)], subset(diamonds, clarity ==
```

```
## Test stat:  9242.3
## Numerator df:  7
## Denominator df:  2523
## P-value:  0
```

The output above comparing the mean vectors of the 'I1' (poorest quality) and 'IF' (best quality) levels of the 'color' variable gives a test statistic of 9242.3 and a p-value of 0. We therefore reject the null hypothesis that the mean vectors of these two levels are equal.

Summary:

It is no surprise to find that the best and worst categories of cut, color and clarity differ significantly from each other.

1.3 Normality

In our previous EDA we saw that a number of our numeric variables may not follow a normal distribution. We saw this in the Cullen and Frey Plots and also in our summary table by examining the values for kurtosis and skewness. We will create a Normal QQ plot for each numeric variable as well as performing a Kolmogrov-Smirnov goodness of fit test for each numeric variable. The density function of each numeric variable indicate strong deviations

from the normal distribution.

```
a <- ggplot(diamonds, aes(x=carat))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Carat")

b <- ggplot(diamonds, aes(x=depth))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Depth")

c <- ggplot(diamonds, aes(x=table))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Table")

d <- ggplot(diamonds, aes(x=price))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Price")

e <- ggplot(diamonds, aes(x=x))+
  geom_density(color="darkblue", fill="lightblue") + xlab("x")

f <- ggplot(diamonds, aes(x=y))+
  geom_density(color="darkblue", fill="lightblue") + xlab("y")

g <- ggplot(diamonds, aes(x=z))+
  geom_density(color="darkblue", fill="lightblue") + xlab("z")


ggarrange(a, b, c, d, e, f, g + rremove("x.text"),
          labels = c("Carat", "Depth", "Table", "Price", "x", "y", "z"),
          ncol = 2, nrow = 4)
```

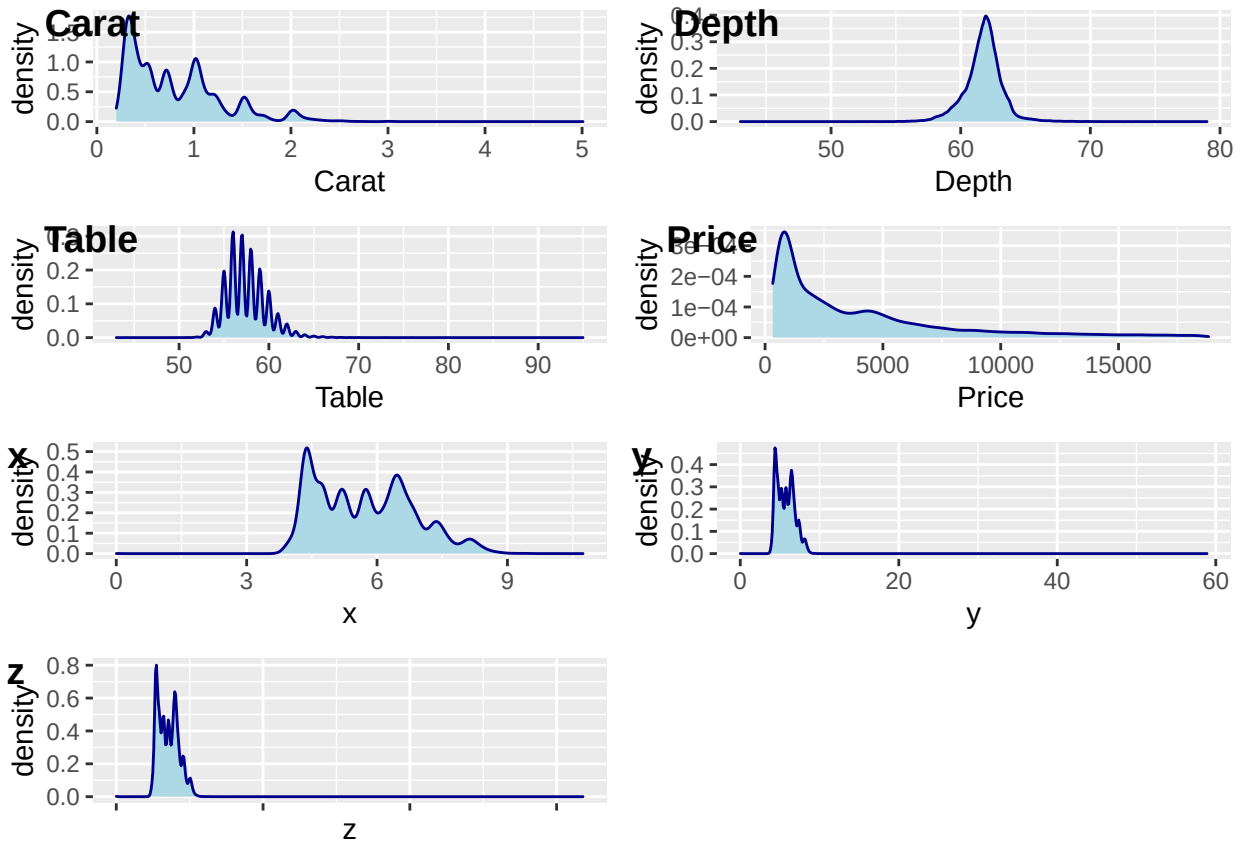

Figure 1: Density of Numeric variables

Figure 1 displays the density plots for the seven numerical variables. Most look like they follow a distribution that is not classically Normal.

'Carat' has a series of modes (four or five) which suggest a categorical variable that is also influencing the shape. This is also true of 'table' and 'x', and also possibly 'y' and 'z', although the horizontal compression of the data makes it harder to distinguish individual modes. 'Depth' is the one that most resembles a classic Bell curve. The left hand side of 'price' resembles a Normal density, but then has an extremely long right tail, which is apparently characteristic of some types of monetary data. Note also that the lowest value of 'price' is not zero; this makes sense, as no diamond, no matter how small, would be sold for zero dollars. Attempting a log transform of numeric variables to increase normality did not yield much success.

```
a <- ggplot(diamonds, aes(x=log(carat)))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Carat")

b <- ggplot(diamonds, aes(x=log(depth)))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Depth")

c <- ggplot(diamonds, aes(x=log(table)))+
```

```
  geom_density(color="darkblue", fill="lightblue") + xlab("Table")

d <- ggplot(diamonds, aes(x=log(price)))+
  geom_density(color="darkblue", fill="lightblue") + xlab("Price")

e <- ggplot(diamonds, aes(x=log(x)))+
  geom_density(color="darkblue", fill="lightblue") + xlab("x")

f <- ggplot(diamonds, aes(x=log(y)))+
  geom_density(color="darkblue", fill="lightblue") + xlab("y")

g <- ggplot(diamonds, aes(x=log(z)))+
  geom_density(color="darkblue", fill="lightblue") + xlab("z")


ggarrange(a, b, c, d, e, f, g + rremove("x.text"),
          labels = c("Carat", "Depth", "Table", "Price", "x", "y", "z"),
          ncol = 2, nrow = 4)
```

```
## Warning: Removed 8 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 7 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 20 rows containing non-finite values (stat_density).
```

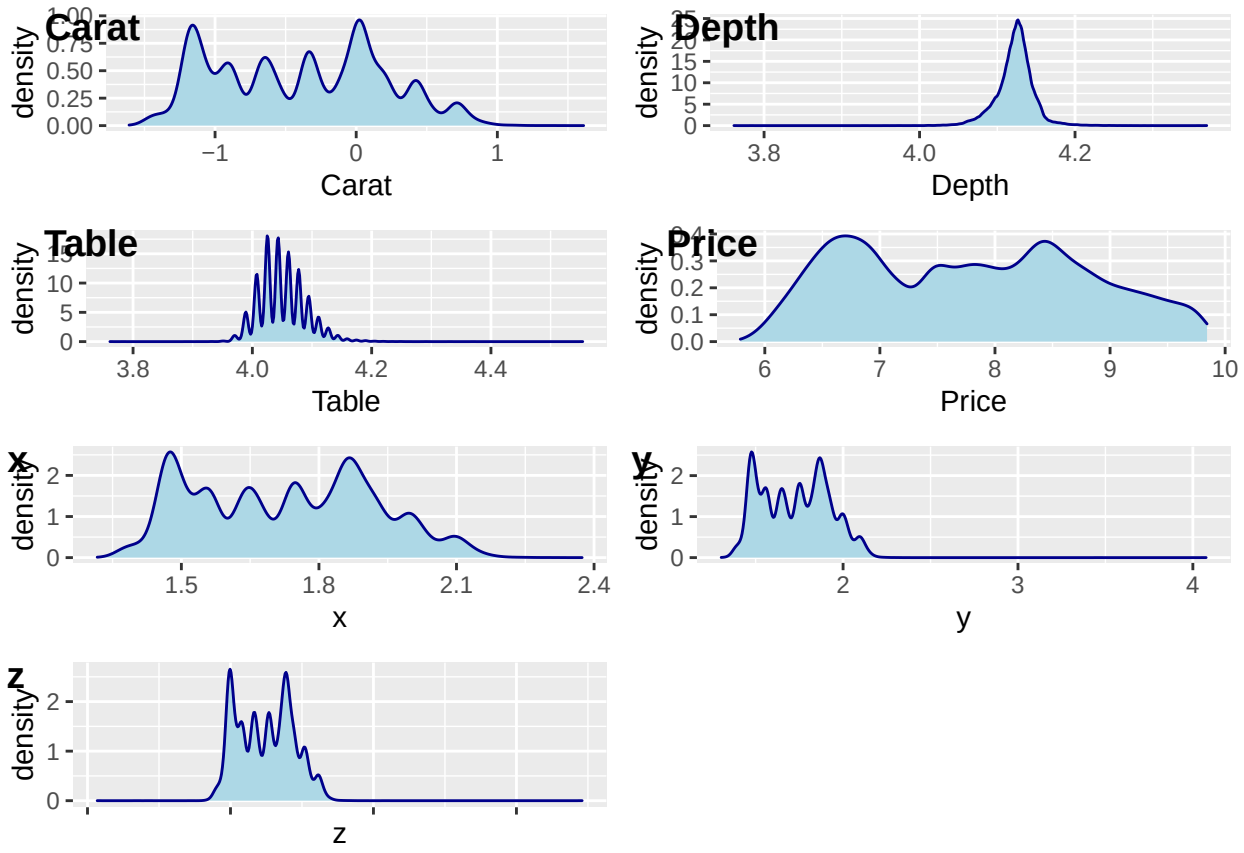

Figure 2: Density of Numeric variables

Figure 2 shows the density plots of the log transformed data. While the data has clearly been stretched, the shapes of the densities remains similar to the original data.

1.4 Q-Q plots

The Normal Q-Q plots below show that most of the variables have deviations from Normality.

1.4.1 Carat

```
qqnorm(diamonds$carat, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$carat, col = "blue", lwd =2)
```

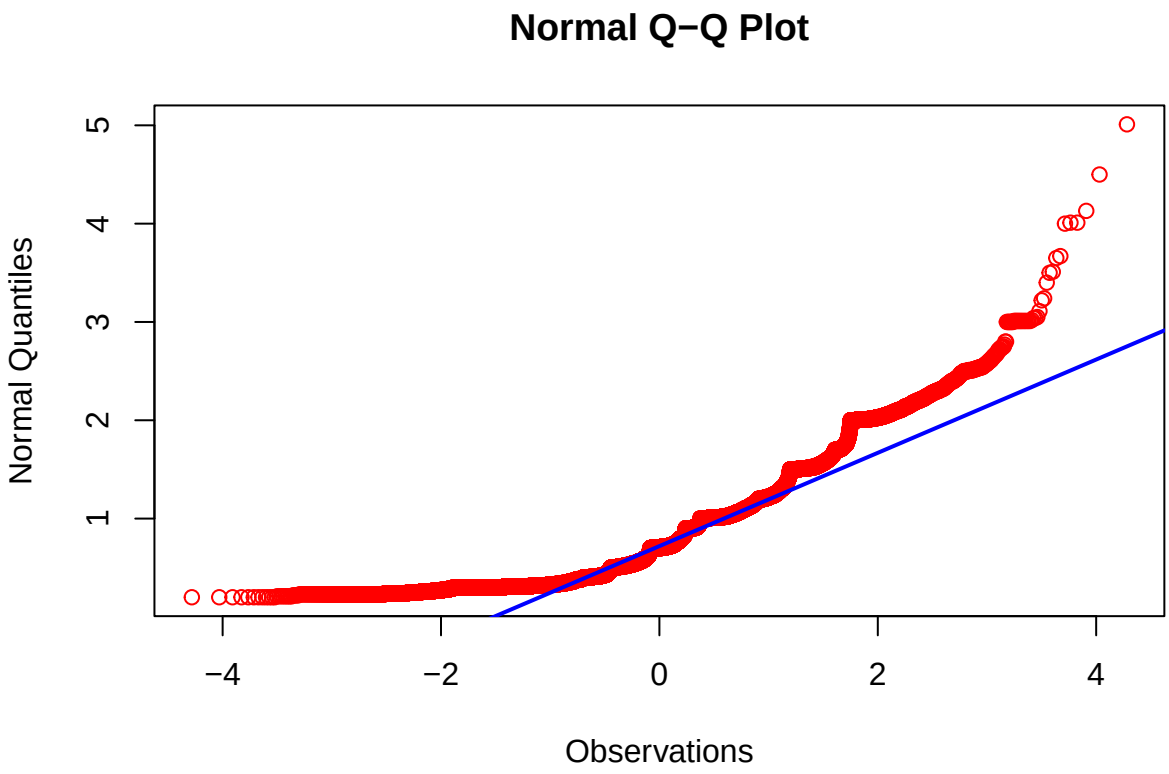

Figure 3: Carat

```
ks.test(diamonds$carat, "pnorm", mean=mean(diamonds$carat), sd=sd(diamonds$carat))
```

```
## Warning in ks.test.default(diamonds$carat, "pnorm", mean =
## mean(diamonds$carat), : ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$carat
## D = 0.12274, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The QQ plot shows strong deviations from the normal distribution particularly at the tails and this is confirmed in our hypothesis test of normality.

```
qqnorm(log(diamonds$carat), xlab = "Observations", ylab = "Normal Quantiles", col = "red
qqline(log(diamonds$carat), col = "blue", lwd =2)
```

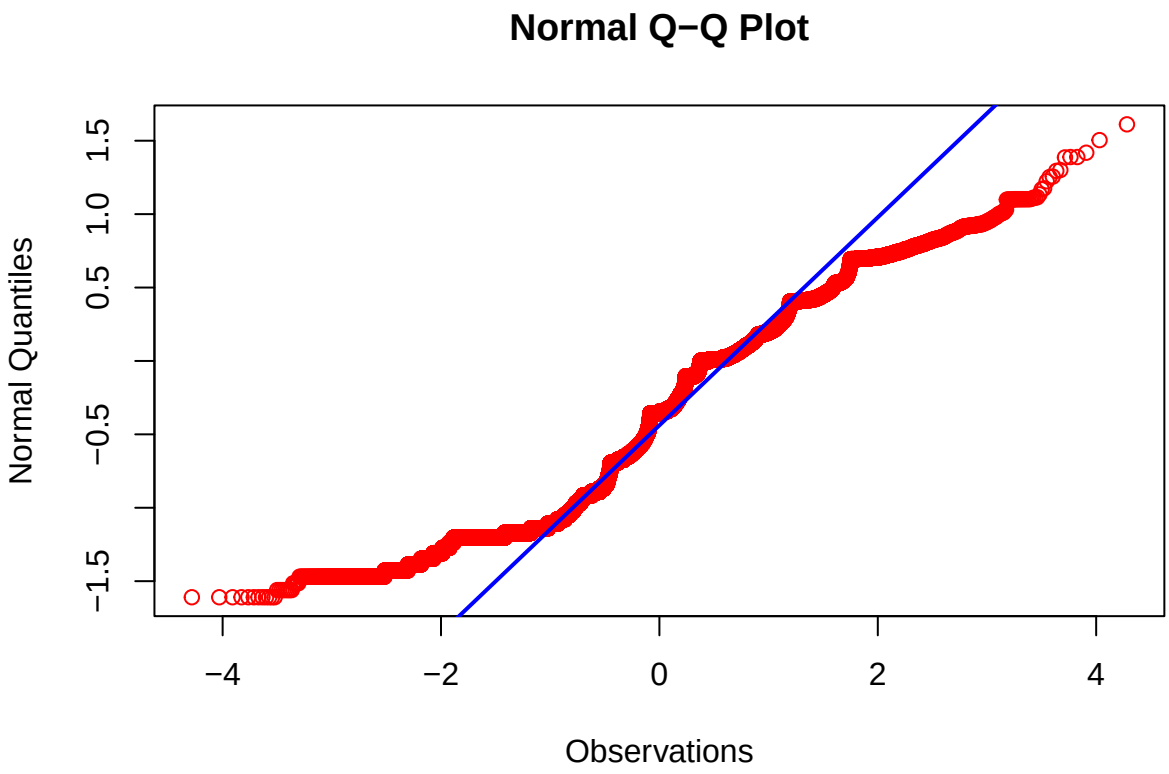

Figure 4: Carat log transformed

Even with the log transformation, the carat values do not fit the predicted line particularly well (figure 4), suggesting that the data is not Normally distributed.

1.4.2 Depth

```
qqnorm(diamonds$depth, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$depth, col = "blue", lwd =2)
```

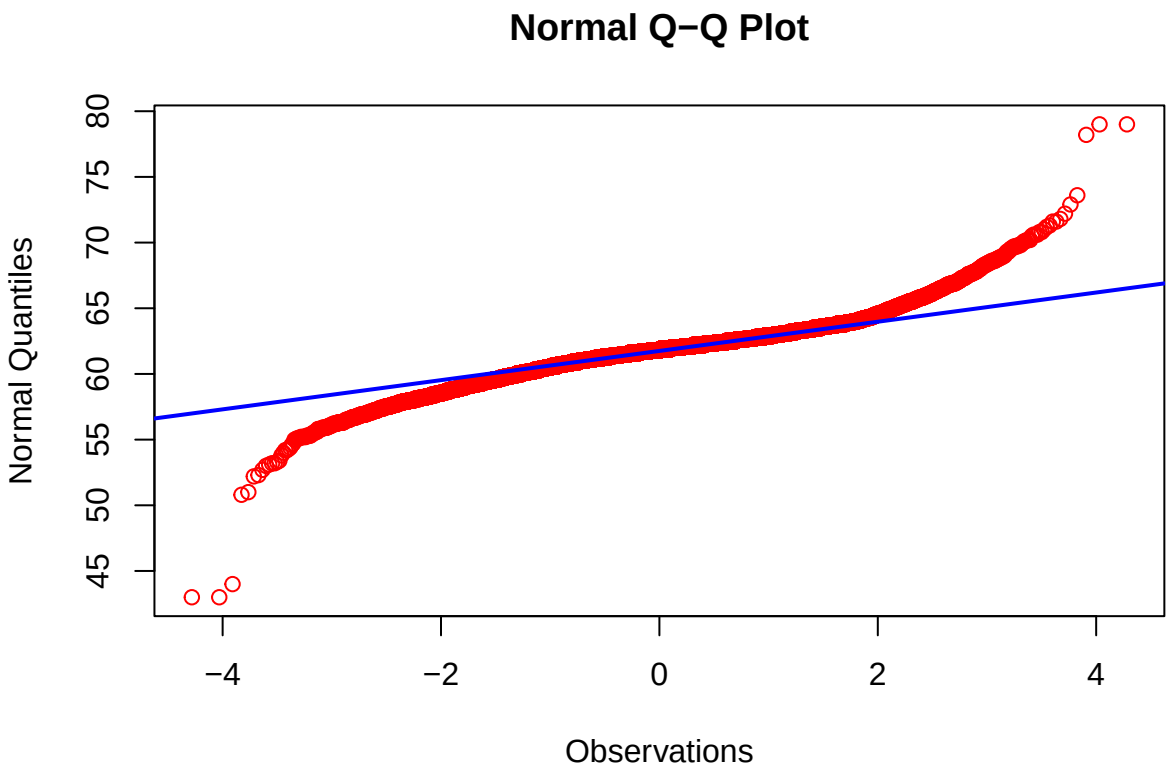

Figure 5: Depth

```
ks.test(diamonds$depth, "pnorm", mean=mean(diamonds$depth), sd=sd(diamonds$depth))
```

```
## Warning in ks.test.default(diamonds$depth, "pnorm", mean =
## mean(diamonds$depth), : ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$depth
## D = 0.075871, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Again the QQplot for depth (figure 4) shows substantial deviations from the Normal distribution at the tails. The Kolmogorov-Smirnov test returned a p-value of $< 2.2e\text{-}16$ (machine precision zero). Therefore, we reject the null hypothesis that the data is normally distributed.

1.4.3 Table

```
qqnorm(diamonds$table, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$table, col = "blue", lwd =2)
```

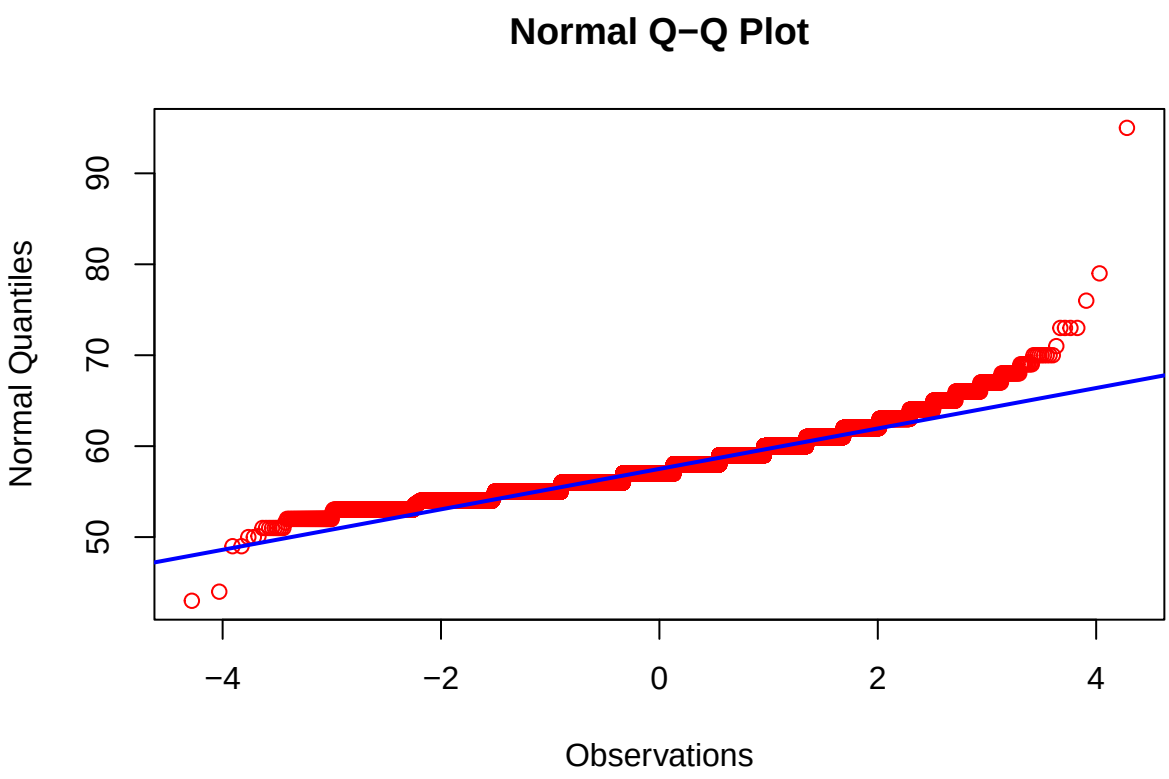

Figure 6: Table

```
ks.test(diamonds$table, "pnorm", mean=mean(diamonds$table), sd=sd(diamonds$table))
```

```
## Warning in ks.test.default(diamonds$table, "pnorm", mean =
## mean(diamonds$table), : ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$table
## D = 0.13225, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Figure 6 of the 'table' variable also suggests deviations from Normality with the upper tail in particular sweeping upward away from the predicted line. The Kolmogorov-Smirnov test returned a p-value of $<$ 2.2e-16, confirming that the data does not follow a Normal

distribution.

1.4.4 Price

```
qqnorm(diamonds$price, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$price, col = "blue", lwd =2)
```

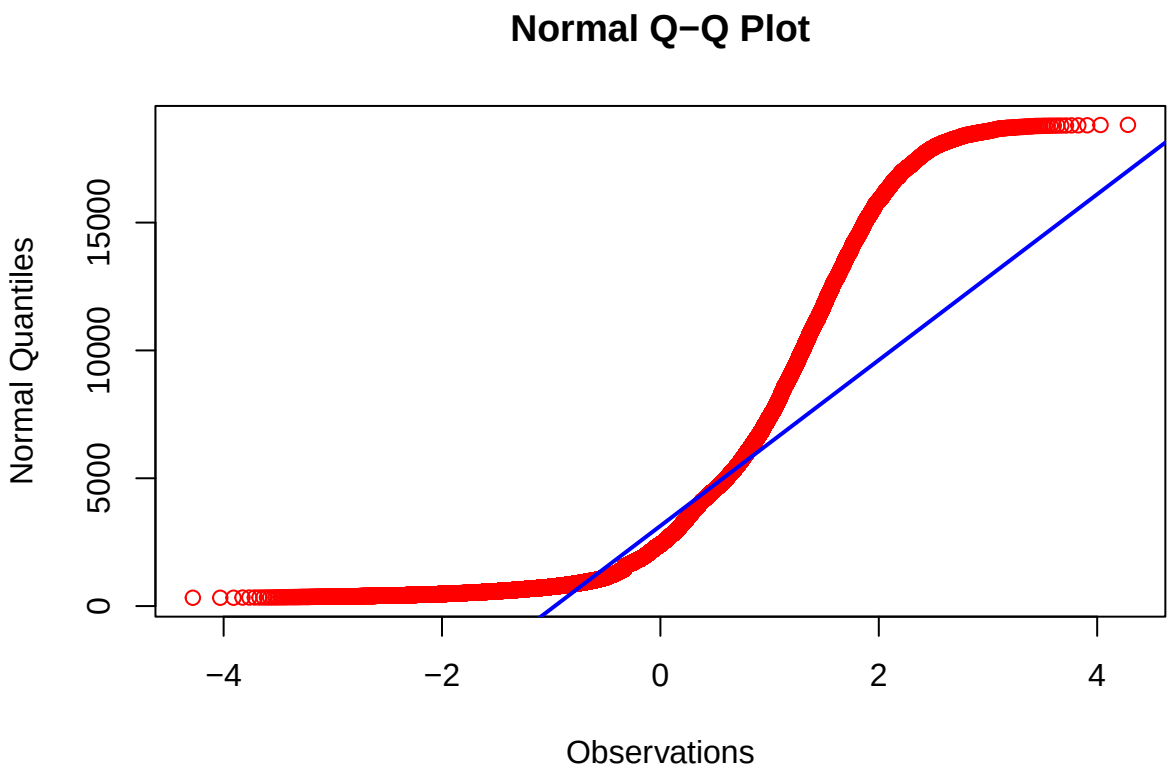

Figure 7: Price

```
ks.test(diamonds$price, "pnorm", mean=mean(diamonds$price), sd=sd(diamonds$price))
```

```
## Warning in ks.test.default(diamonds$price, "pnorm", mean =
## mean(diamonds$price), : ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$price
## D = 0.18467, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Figure 7 shows an unusually shaped red curve, which strongly suggests a distribution other than the Normal. Again, the Kolmogorov-Smirnov test returned a p-value of $< 2.2e\text{-}16$.

1.4.5 x (length)

```
qqnorm(diamonds$x, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$x, col = "blue", lwd =2)
```

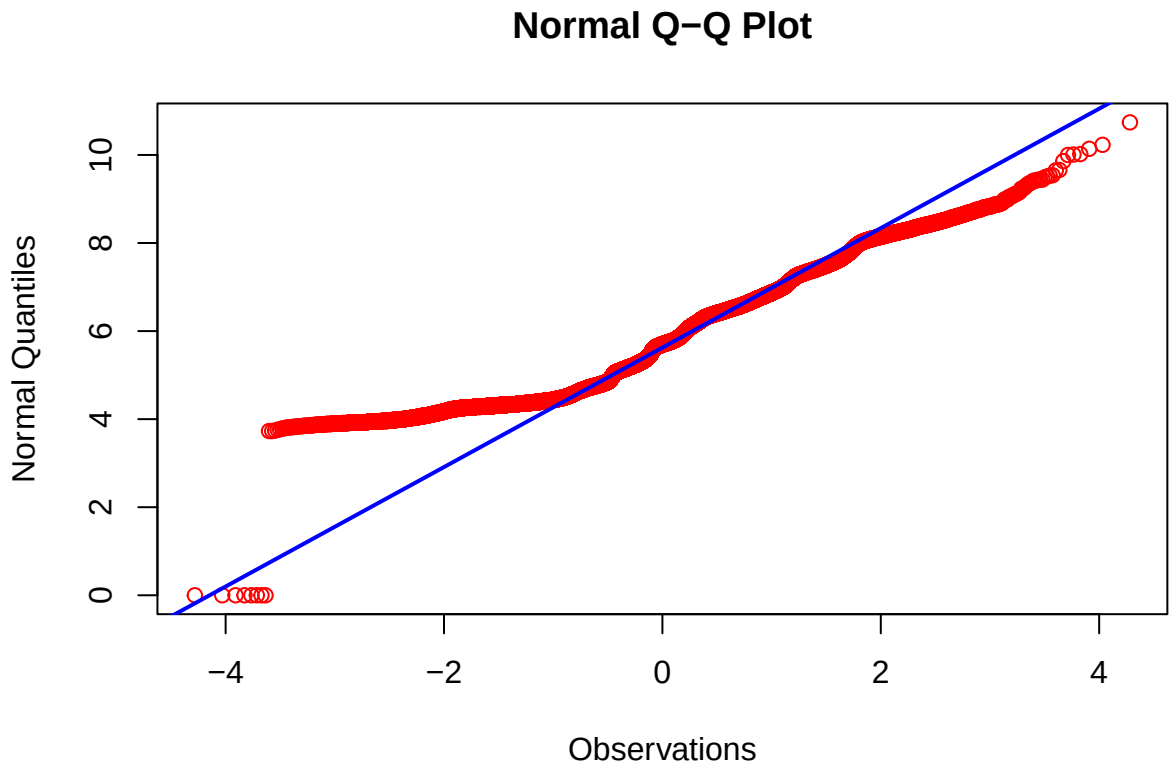

Figure 8: X

```
ks.test(diamonds$x, "pnorm", mean=mean(diamonds$x), sd=sd(diamonds$x))
```

```
## Warning in ks.test.default(diamonds$x, "pnorm", mean = mean(diamonds$x), : ties
## should not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$x
## D = 0.093545, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

1.4.6 y (width)

```
qqnorm(diamonds$y, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$y, col = "blue", lwd =2)
```

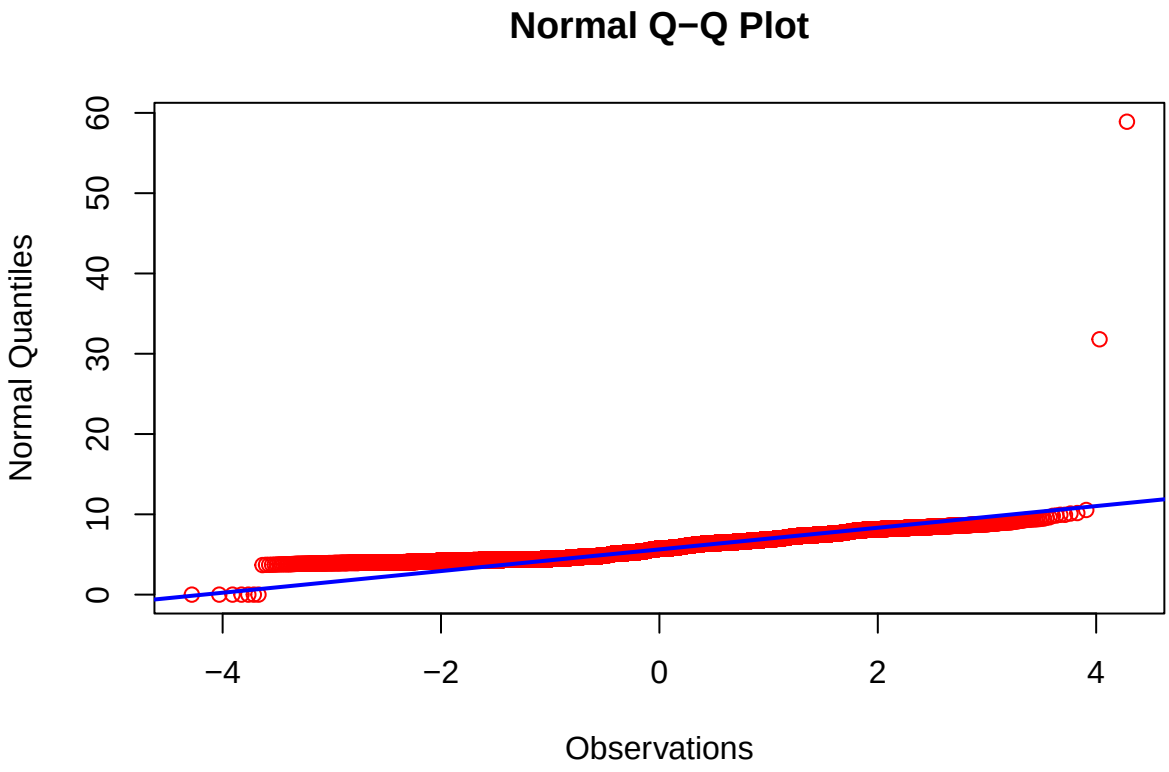

Figure 9: Y

```
ks.test(diamonds$y, "pnorm", mean=mean(diamonds$y), sd=sd(diamonds$y))
```

```
## Warning in ks.test.default(diamonds$y, "pnorm", mean = mean(diamonds$y), : ties
## should not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$y
## D = 0.088528, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Figure 9 shows that the observed values for the 'y' variable actually fit the predicted line quite well, with the exception of two extreme values at the upper end (right hand side).

1.4.7 z (depth)

```
qqnorm(diamonds$depth, xlab = "Observations", ylab = "Normal Quantiles", col = "red")
qqline(diamonds$depth, col = "blue", lwd =2)
```

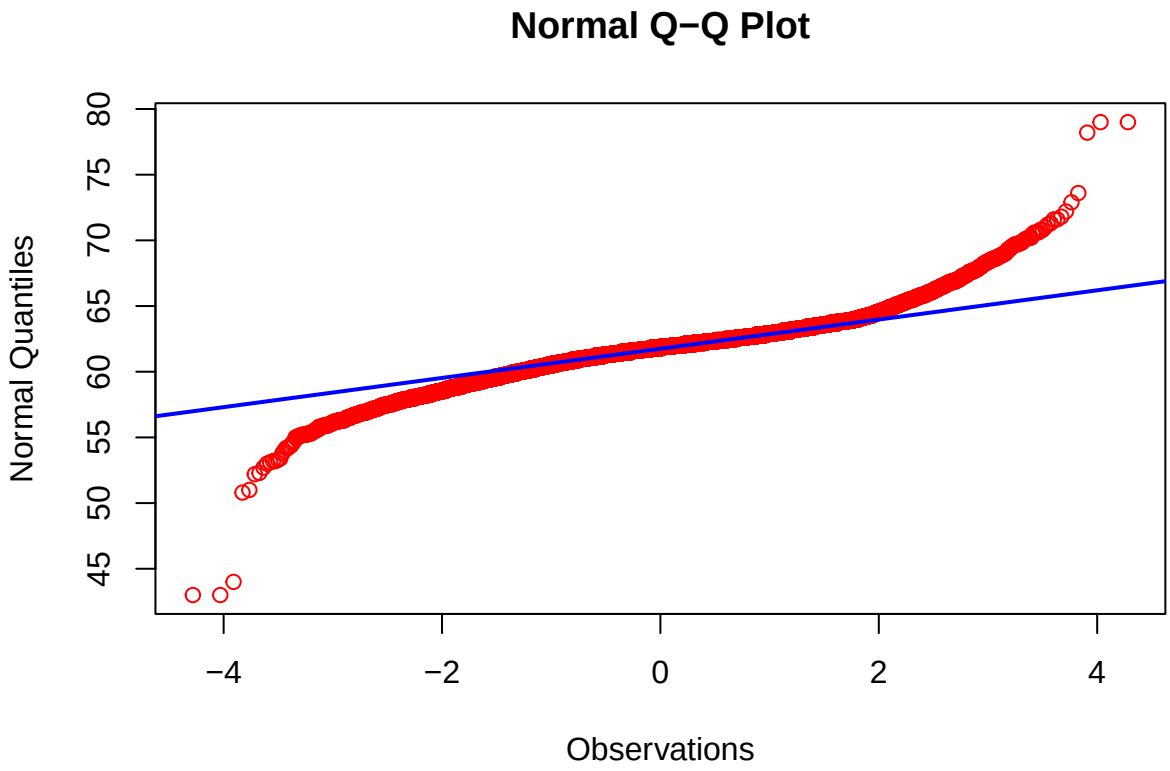

Figure 10: Z

```
ks.test(diamonds$depth, "pnorm", mean=mean(diamonds$depth), sd=sd(diamonds$depth))
```

```
## Warning in ks.test.default(diamonds$depth, "pnorm", mean =
## mean(diamonds$depth), : ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  diamonds$depth
## D = 0.075871, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Figures 8 (variable 'X') and 10 (variable 'Y') both show deviations from Normality.

All of our numerical variables, except perhaps y (width), show obvious deviation from the

normal distribution in the QQ plots. The KS goodness of fit test finds evidence that these varaibles do not follow a normal distribution.

1.5 Melted version of dataset

1.6 Boxplots and table of 'cut'

Table: cut count

Cut	Count
Fair	1610
Good	4960
Ideal	21551
Premium	13791
Very Good	12082

The table above gives a breakdown of the how many diamonds are in each level of the 'cut' variable. We can see that most are in the 'Ideal', with a substancial number also in the 'Premium' and 'Very Good'.

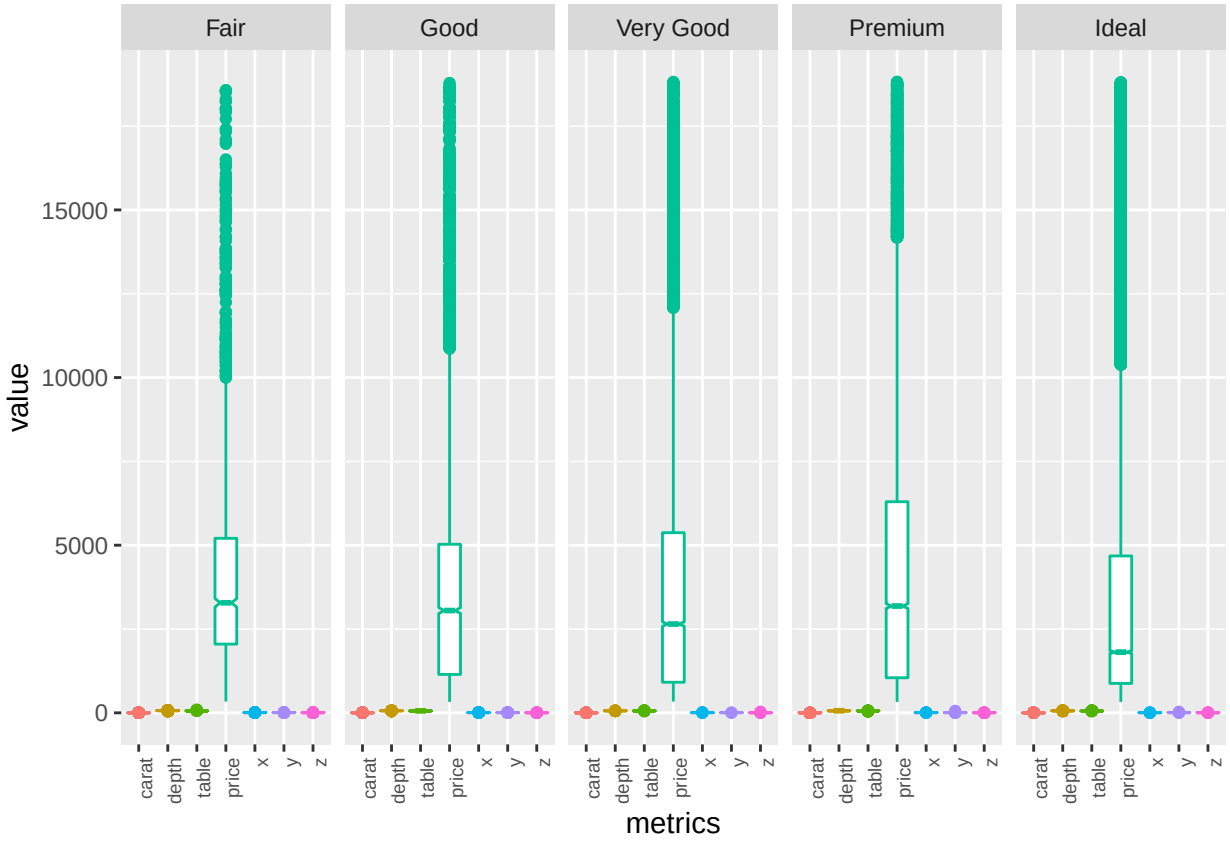

Figure 11: Boxplots of 'cut' vs all the numeric variables

Figure 11 shows that all the variables except 'price' are too compressed to view. Therefore, a log transform was performed and the graph redone.

1.7 Boxplots of 'cut' in log scale

The summary of the dataset shows that there are no negative or zero values, so we can proceed with a log transform.

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 35 rows containing non-finite values (stat_boxplot).
```

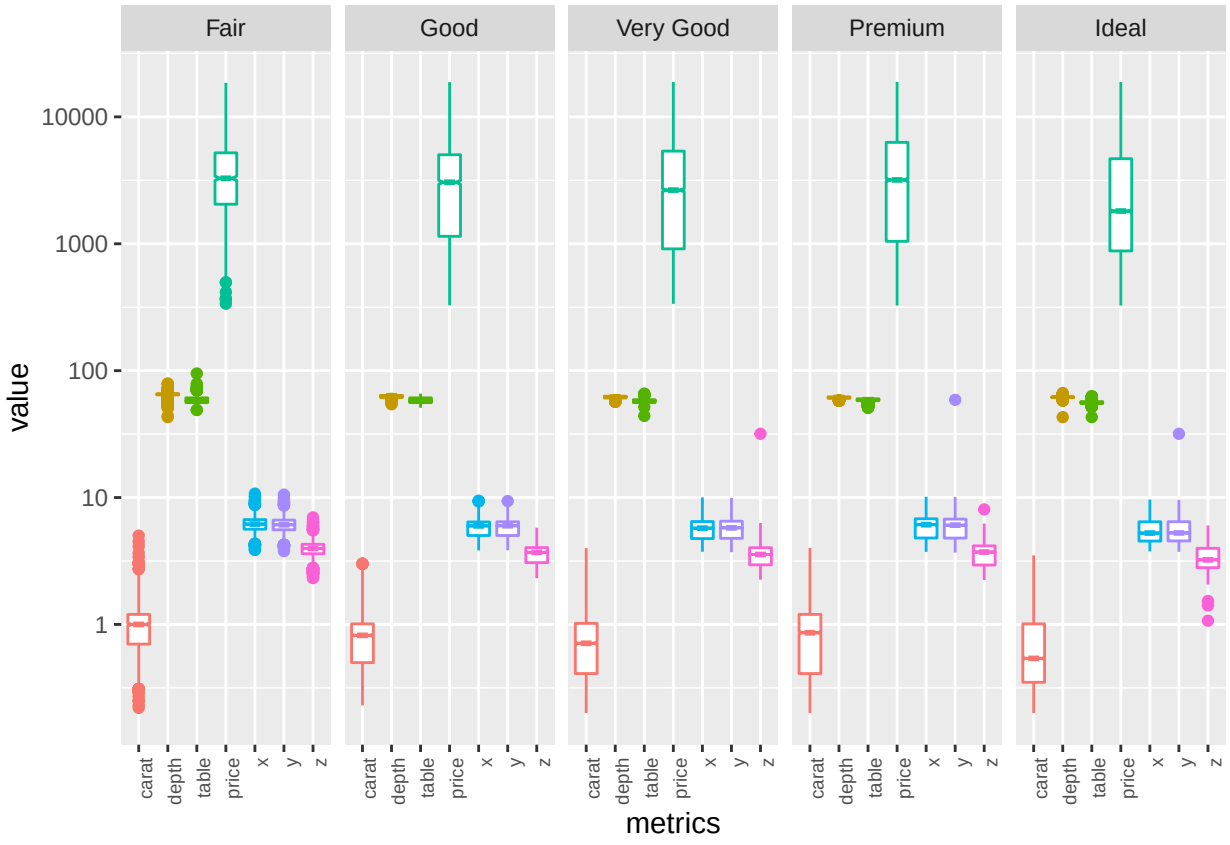

Figure 12: Boxplots of 'cut' vs all the numeric variables (log transformed)

The log transform in Figure 12 gives a much better idea of the data. 'Price' (green boxes) is consistent across the different levels of 'cut'. Indeed, on this graph the medians and variances of all the variables look similar across the different levels of 'cut'. However, genuine differences might be difficult to perceive due to the scale of the graph and because the sample size is so large, meaning that a seemingly small difference on the graph could still be significant. Most of the confidence interval notches on the boxplots are too compressed to be of help. Below we conduct an ANOVA to determine if we have evidence of differences in price for different levels of the categorical variable cut.

For two of the variables (measurements 'y', purple, and 'z' pink) in the 'Very Good', 'Premium' and 'Ideal' levels of 'cut' there appear to be some very prominent outliers, as evidenced by the pink and purple dots above and below the boxplots. The variable 'y' is a measure of width in millimeters (mm), while 'z' is a measure of depth in mm.

1.8 Differences in Price for different levels of Cut

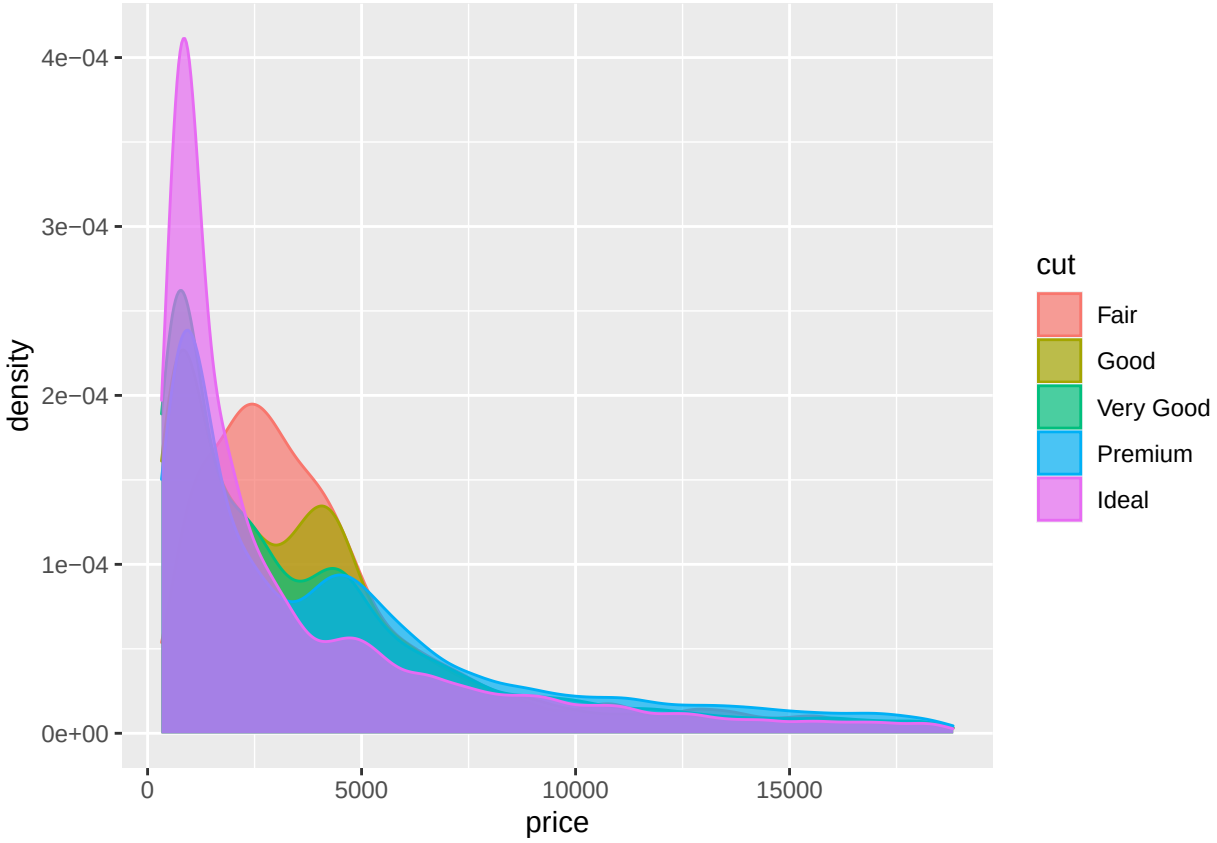

Figure **??** shows the density plots for the different levels of the 'price' variable. The legend on the right shows the level names and their order ('fair' = poorest, 'ideal' = best). Of note is the fact that the two best leves (premium and ideal) have significant peaks near the lower end of the price range compared with the other three. This is somewhat surprising, as intuitively one would imagine price to increase as the quality of cut increases. Perhaps it is easier to complete a premium or ideal cut on a smaller diamond, which would then be sold at a cheaper price than a rougher cut on a larger diamond?

We will now test whether there are significant differences in mean price for different levels of "cut".

```
cutanova <- aov(price ~ cut, data = diamonds)
summary(cutanova)
```

```
##                Df    Sum Sq   Mean Sq F value Pr(>F)
## cut             4 1.104e+10 2.760e+09   175.7 <2e-16 ***
```

```
## Residuals   53935 8.474e+11 1.571e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA results above return a p-value of $< 2.2e\text{-}16$, meaning that there is strong evidence to suggest that mean price differs across levels of "cut".

We will now perform a Tukey Test to determine which pairwise differences are significant

```
TukeyHSD(cutanova, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = price ~ cut, data = diamonds)
##
## $cut
##                          diff         lwr        upr     p adj
## Good-Fair          -429.89331  -740.44880  -119.3378 0.0014980
## Very Good-Fair     -376.99787  -663.86215   -90.1336 0.0031094
## Premium-Fair        225.49994   -59.26664   510.2665 0.1950425
## Ideal-Fair         -901.21579 -1180.57139  -621.8602 0.0000000
## Very Good-Good       52.89544  -130.15186   235.9427 0.9341158
## Premium-Good        655.39325   475.65120   835.1353 0.0000000
## Ideal-Good         -471.32248  -642.36268  -300.2823 0.0000000
## Premium-Very Good   602.49781   467.76249   737.2331 0.0000000
## Ideal-Very Good    -524.21792  -647.10467  -401.3312 0.0000000
## Ideal-Premium     -1126.71573 -1244.62267 -1008.8088 0.0000000
```

At the 5% significance level, the only pairs between which we do not see a significant difference in mean price are "very Good" and "Good" as well as "Premium" and "Fair" (output above).

From the graph of the distributions of price for different levels of cut we can see that not all of them have a shape consistent with being normally distributed. A one Way ANOVA is reasonably robust to departures from normality, particularly as we have a very large sample. We will also perform Levene's test to test the assumption of equal variances and then, if significant, a non-parametric Kruskal Wallis test to determine if there are significant differences in median price for different levels of cut. Again, a one way ANOVA is reasonably robust to departures from equal variance if the sample sizes are the same.

```
leveneTest(price ~ cut, data= diamonds)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     4   123.6 < 2.2e-16 ***
##       53935
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
kruskal.test(price ~ cut, data = diamonds)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  price by cut
## Kruskal-Wallis chi-squared = 978.62, df = 4, p-value < 2.2e-16
```

Both tests return significant results (output above) indicating that: a) the assumption of equal variance is violated and; b) that we have evidence of a significant difference in median prices for different levels of cut.

1.9 Boxplots of 'color' in log scale

1.9.1 Table of 'color' count

Table: color count

Color	Count
D	6775
E	9797
F	9542
G	11292
H	8304
I	5422
J	2808

The table above shows the number of diamonds in each level of the 'color' variable.

As with the 'cut' variable, we redo the boxplots using the log transform on the data. Figure 13 shows the log transformed version of the boxplots for the 'color' variable.

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 35 rows containing non-finite values (stat_boxplot).
```

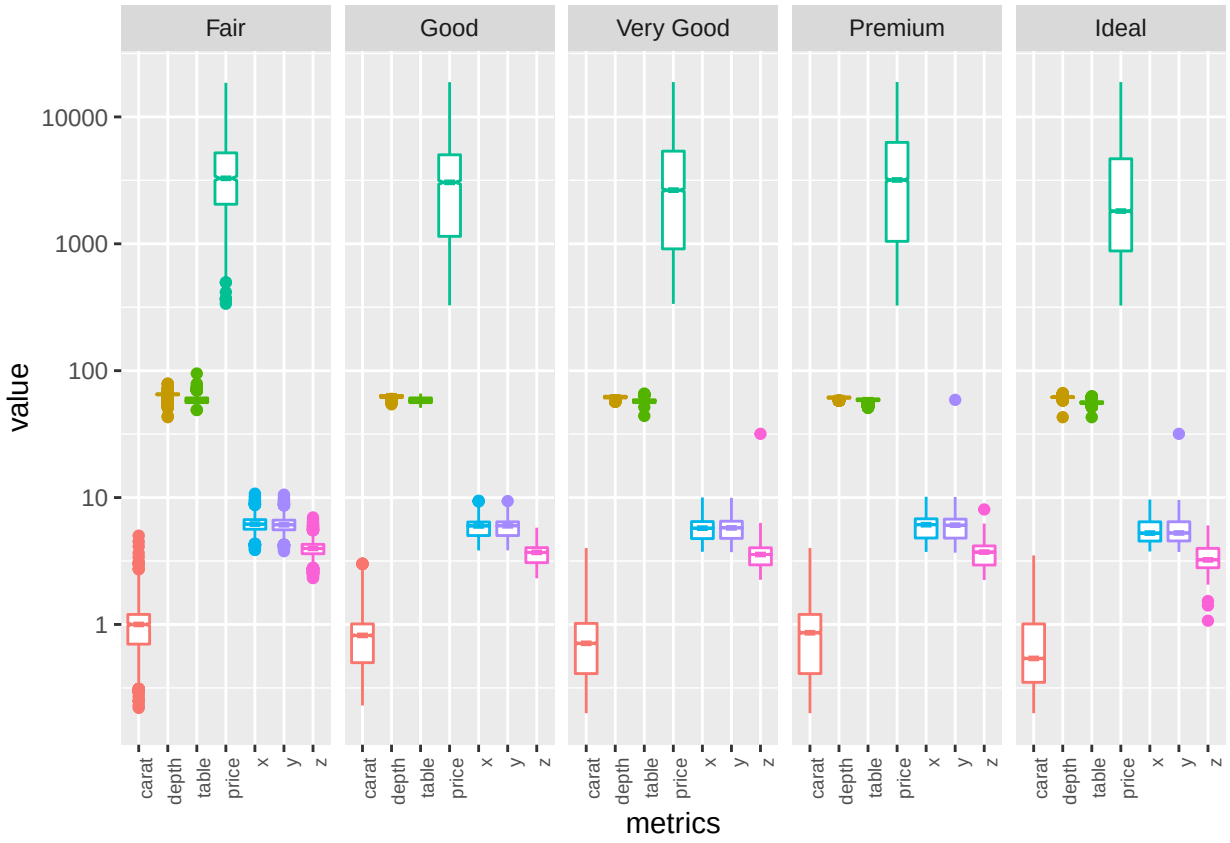

Figure 13: Boxplots of 'color' vs all the numeric variables (log transformed)

1.10 Differences in Price for different colors

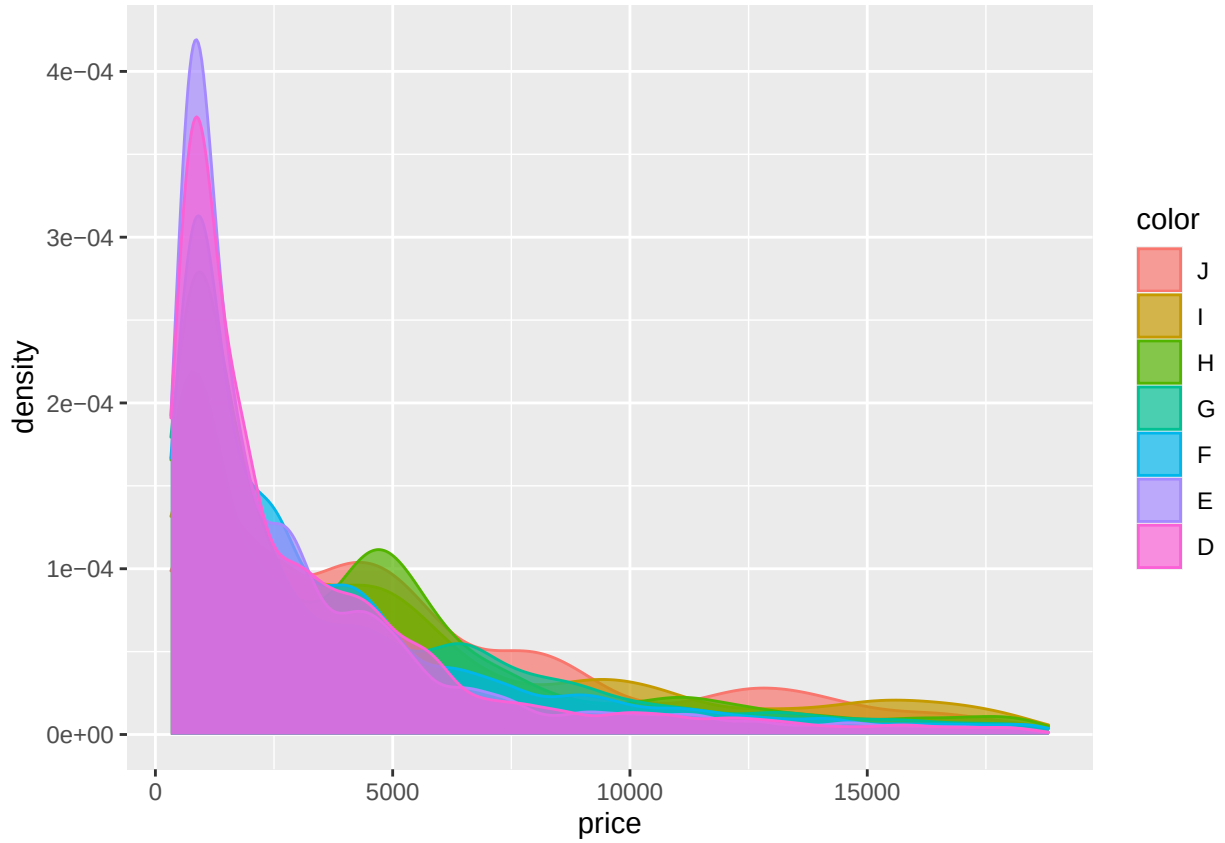

Figure 14: Densities of 'price' for the different levels of 'color'

Figure 14 shows the density plots of 'price' for the different levels of the 'color' variable. There is a prominent peak on the left for the better quality levels of color (D in pink, E in purple, and F in blue), but otherwise the densities appear to be fairly similar.

We will now test whether there are significant differences in mean price for different diamond colours. While we see potential evidence that the ANOVA assumptions of normality and equal variance may be violated, ANOVA is reasonably robust to these violations if the sample size is big enough.

```
coloranova <- aov(price ~ color, data = diamonds)
summary(cutanova)
```

```
##                Df    Sum Sq   Mean Sq F value Pr(>F)
## cut             4 1.104e+10 2.760e+09   175.7 <2e-16 ***
## Residuals   53935 8.474e+11 1.571e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have strong evidence to suggest that mean price differs across levels of "color". We will

now perform a Tukey Test to determine which pairwise comparisons are significant

```
TukeyHSD(coloranova, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = price ~ color, data = diamonds)
##
## $color
##            diff         lwr         upr     p adj
## I-J  -231.94307  -501.11666    37.23053 0.1449244
## H-J  -837.14882 -1089.88345  -584.41420 0.0000000
## G-J -1324.68235 -1568.82093 -1080.54376 0.0000000
## F-J -1598.93162 -1847.48867 -1350.37458 0.0000000
## E-J -2247.06554 -2494.88601 -1999.24508 0.0000000
## D-J -2153.86392 -2413.70535 -1894.02250 0.0000000
## H-I  -605.20576  -807.34909  -403.06243 0.0000000
## G-I -1092.73928 -1284.02646  -901.45210 0.0000000
## F-I -1366.98856 -1563.88381 -1170.09331 0.0000000
## E-I -2015.12248 -2211.08707 -1819.15789 0.0000000
## D-I -1921.92086 -2132.88224 -1710.95948 0.0000000
## G-H  -487.53352  -654.89884  -320.16821 0.0000000
## F-H  -761.78280  -935.53004  -588.03556 0.0000000
## E-H -1409.91672 -1582.60860 -1237.22484 0.0000000
## D-H -1316.71510 -1506.25419 -1127.17600 0.0000000
## F-G  -274.24927  -435.23673  -113.26182 0.0000106
## E-G  -922.38320 -1082.23107  -762.53532 0.0000000
## D-G  -829.18158 -1007.09708  -651.26607 0.0000000
## E-F  -648.13392  -814.65208  -481.61576 0.0000000
## D-F  -554.93230  -738.86403  -371.00057 0.0000000
## D-E    93.20162   -89.73351   276.13675 0.7437450
```

The output above shows significant differences in mean price for nearly all pairwise comparisons of diamond colors.

We will also perform a Levene's test to test the assumption of equal variances and then, if significant, a non-parametric Kruskal Wallis test to determine if there are significant differences in median price for different levels of cut.

```
leveneTest(price ~ color, data= diamonds)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     6  219.12 < 2.2e-16 ***
##       53933
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
kruskal.test(price ~ color, data = diamonds)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  price by color
## Kruskal-Wallis chi-squared = 1335.6, df = 6, p-value < 2.2e-16
```

The output above shows a p-value of $<$ 2.2e-16, meaning that there is strong evidence of a significant difference in the median price for different colour diamonds.

1.11 Boxplots and table of count of 'clarity'

Table: clarity count

Clarity	Count
I1	741
IF	1790
SI1	13065
SI2	9194
VS1	8171
VS2	12258
VVS1	3655
VVS2	5066

The table above shows the counts for the different levels of the 'clarity' variable.

For the 'clarity' variable a log transform on the data has been performed for the boxplots.

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 35 rows containing non-finite values (stat_boxplot).
```

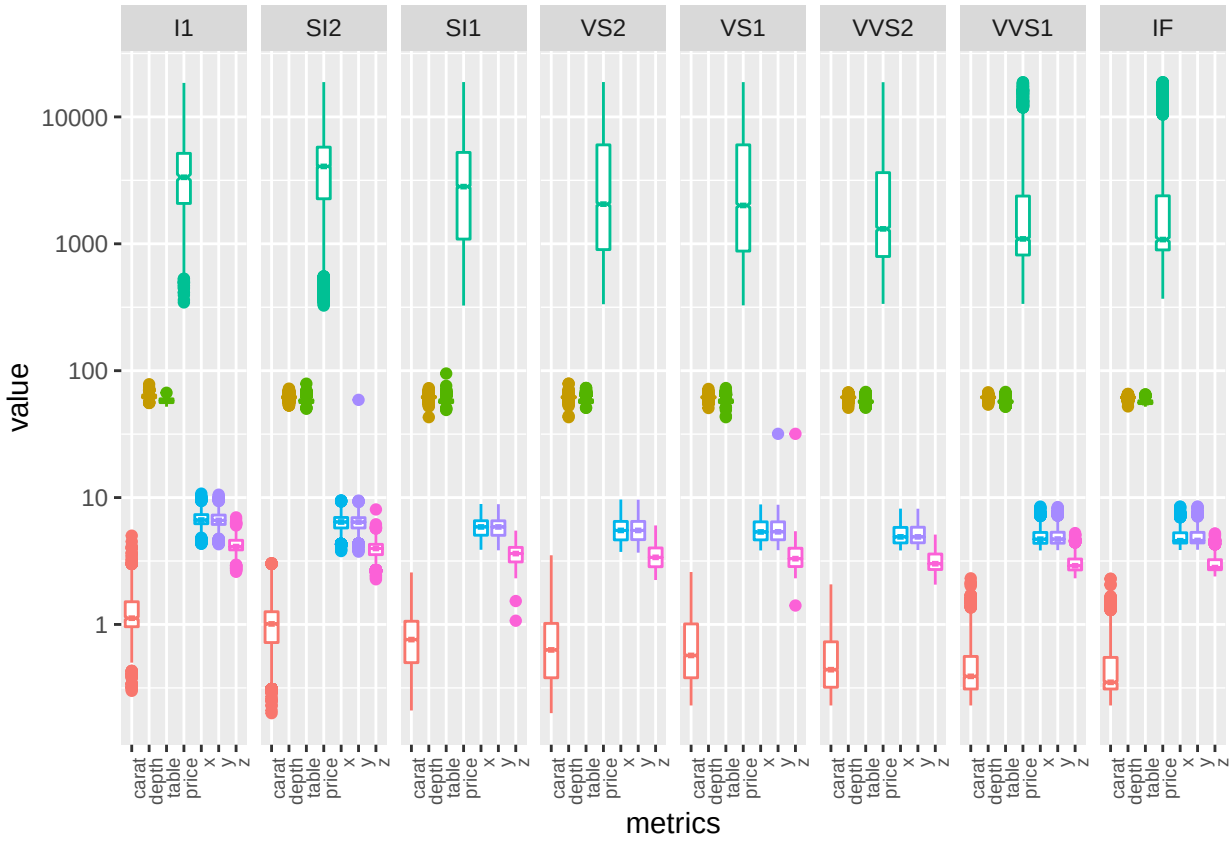

Figure 15: Boxplots of 'clarity' vs all the numeric variables (log transformed)

Figure 15 shows the boxplots for the log transformed data across the different 'clarity' metrics. Much like the previous two graphs, medians and ranges look relatively constant across the variables.

1.12 Differences in price for different levels of clarity

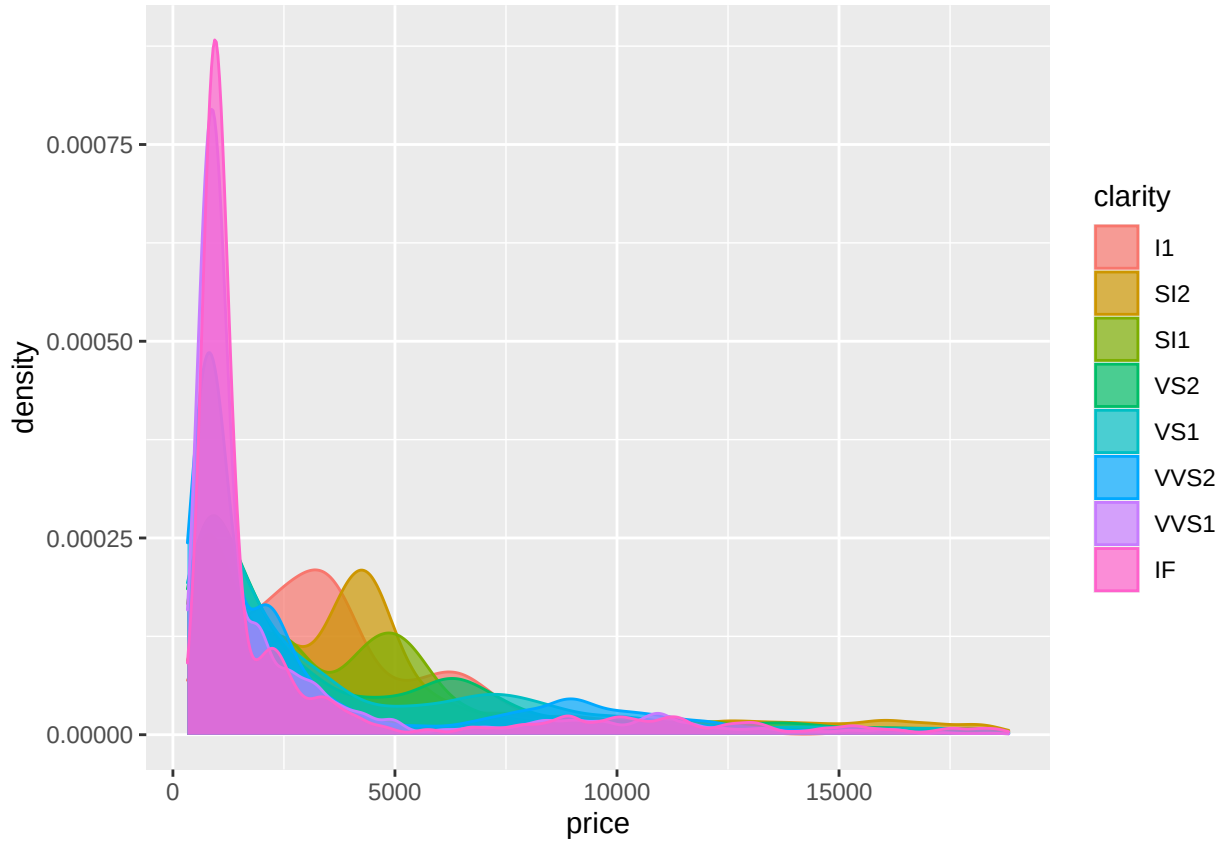

Figure **??** shows the densities of the different levels of 'clarity' for price. A prominent peak is visible near the left for the better quality levels (IF, best, pink; WS1 second best, purple; WS2, third best, blue).

We will now test whether there are significant differences in mean price for different diamond colours. While we see potential evidence that the ANOVA assumptions of normality and equal variance may be violated, ANOVA is reasonably robust to these violations if the sample size is big enough.

```
clarityanova <- aov(price ~ clarity, data = diamonds)
summary(clarityanova)
```

```
##                Df    Sum Sq   Mean Sq F value Pr(>F)
## clarity         7 2.331e+10 3.330e+09     215 <2e-16 ***
## Residuals   53932 8.352e+11 1.549e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output above returns a p-value of $<$ 2.2e-16, meaning there is strong evidence to suggest that mean price differs across levels of "clarity". We will now perform a Tukey Test to determine which pairwise comparisons are significant

```
TukeyHSD(clarityanova, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = price ~ clarity, data = diamonds)
##
## $clarity
##                     diff          lwr         upr     p adj
## SI2-I1     1138.8599147   683.395891  1594.32394 0.0000000
## SI1-I1       71.8324571  -378.570901   522.23582 0.9997320
## VS2-I1        0.8207037  -450.377702   452.01911 1.0000000
## VS1-I1      -84.7132999  -542.298929   372.87233 0.9992819
## VVS2-I1    -640.4316203 -1109.531923  -171.33132 0.0009165
## VVS1-I1   -1401.0540535 -1881.569711  -920.53840 0.0000000
## IF-I1     -1059.3295848 -1580.334655  -538.32451 0.0000000
## SI1-SI2   -1067.0274575 -1229.386830  -904.66808 0.0000000
## VS2-SI2   -1138.0392109 -1302.591274  -973.48715 0.0000000
## VS1-SI2   -1223.5732146 -1404.907129 -1042.23930 0.0000000
## VVS2-SI2  -1779.2915349 -1987.983831 -1570.59924 0.0000000
## VVS1-SI2  -2539.9139681 -2773.136347 -2306.69159 0.0000000
## IF-SI2    -2198.1894995 -2506.318797 -1890.06020 0.0000000
## VS2-SI1     -71.0117534  -220.988718    78.96521 0.8410824
## VS1-SI1    -156.5457571  -324.764949    11.67343 0.0899007
## VVS2-SI1   -712.2640774  -909.667681  -514.86047 0.0000000
## VVS1-SI1  -1472.8865106 -1696.064436 -1249.70859 0.0000000
## IF-SI1    -1131.1620420 -1431.760399  -830.56369 0.0000000
## VS1-VS2     -85.5340037  -255.870471    84.80246 0.7958312
## VVS2-VS2   -641.2523240  -840.463263  -442.04138 0.0000000
## VVS1-VS2  -1401.8747572 -1626.652874 -1177.09664 0.0000000
## IF-VS2    -1060.1502885 -1361.938605  -758.36197 0.0000000
## VVS2-VS1   -555.7183203  -769.001243  -342.43540 0.0000000
## VVS1-VS1  -1316.3407535 -1553.679770 -1079.00174 0.0000000
## IF-VS1     -974.6162849 -1285.873083  -663.35949 0.0000000
## VVS1-VVS2  -760.6224332 -1019.466585  -501.77828 0.0000000
## IF-VVS2    -418.8979645  -746.848084   -90.94785 0.0027364
## IF-VVS1     341.7244687    -2.356168   685.80510 0.0531204
```

The output of the Tukey test is above. Most pairwise combinations show a significant difference. There are only six that do not show a difference and they are: SI1-I1; VS2-I1; VS1-I1; VS2-SI1; VS1-VS2; and IF-VVS1.

We will also perform a Levene's test to test the assumption of equal variances and then, if significant, a non-parametric Kruskal Wallis test to determine if there are significant differences in median price for different levels of cut.

```
leveneTest(price ~ clarity, data= diamonds)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     7  77.809 < 2.2e-16 ***
##       53932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
kruskal.test(price ~ clarity, data = diamonds)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  price by clarity
## Kruskal-Wallis chi-squared = 2718.2, df = 7, p-value < 2.2e-16
```

The output of the two tests above show that there is not equal variances between the levels, and that there is a significant difference between different levels of clarity.

1.12.1 The covariance matrix

Table 2: Covariance matrix for 'diamonds' (2 s.f.)

	carat	depth	table	price	x	y	z
carat	2.2e-01	0.019	0.19	1700	0.520	0.520	3.2e-01
depth	1.9e-02	2.100	-0.95	-61	-0.041	-0.048	9.6e-02
table	1.9e-01	-0.950	5.00	1100	0.490	0.470	2.4e-01
price	1.7e+03	-61.000	1100.00	16000000	4000.000	3900.000	2.4e+03
x	5.2e-01	-0.041	0.49	4000	1.300	1.200	7.7e-01
y	5.2e-01	-0.048	0.47	3900	1.200	1.300	7.7e-01
z	3.2e-01	0.096	0.24	2400	0.770	0.770	5.0e-01

The covariance matrix can be seen in table 2.

1.12.2 The correlation matrix

Table 3: Correlation matrix for 'diamonds' (2 s.f.)

	carat	depth	table	price	x	y	z
carat	1.000	0.028	0.18	0.920	0.980	0.950	0.950
depth	0.028	1.000	-0.30	-0.011	-0.025	-0.029	0.095
table	0.180	-0.300	1.00	0.130	0.200	0.180	0.150
price	0.920	-0.011	0.13	1.000	0.880	0.870	0.860
x	0.980	-0.025	0.20	0.880	1.000	0.970	0.970
y	0.950	-0.029	0.18	0.870	0.970	1.000	0.950
z	0.950	0.095	0.15	0.860	0.970	0.950	1.000

The correlation matrix can be seen in table 3.

1.13 Visualisation of the correlation matrix

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

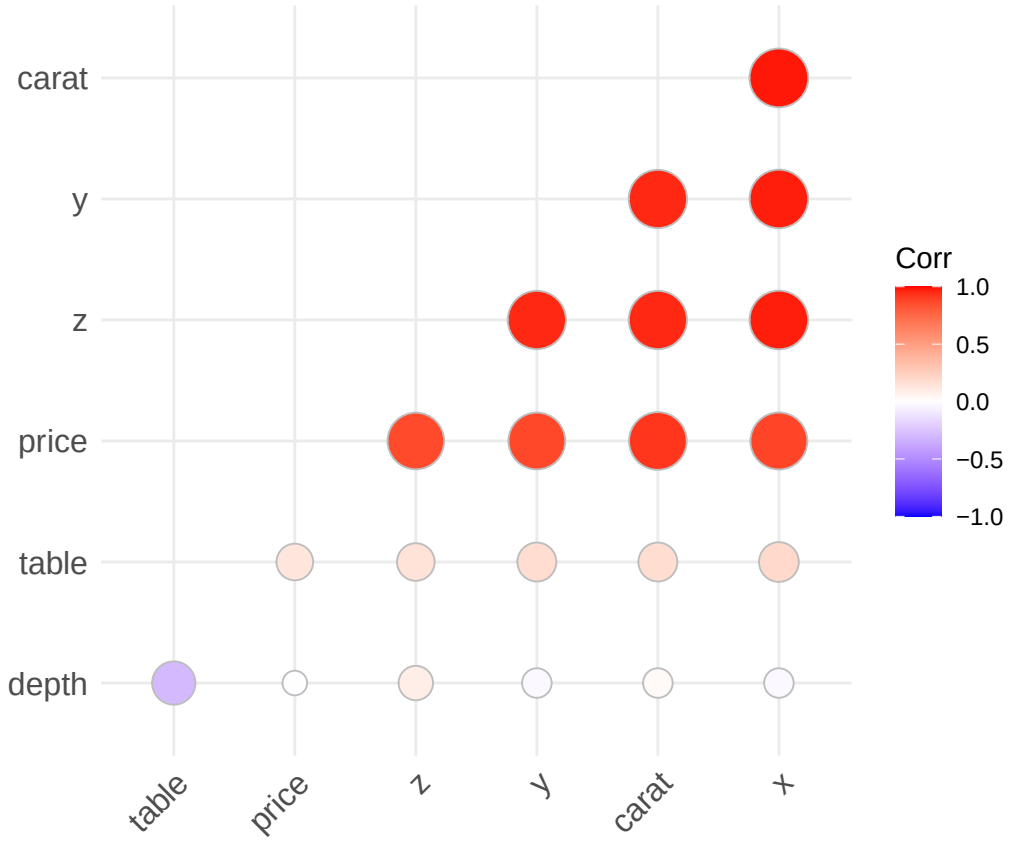

Figure 16: The pairs plot for diamonds

Figure 16 shows the correlation plot for the numerical variables in the diamonds data. 'Carat', 'x', 'y', 'z' and 'price' all show very strong correlations with each other, as evidenced by the large red dots. As "x", "y" and "z" are all measures of size we should expect this and there may be some redundancy in these predictors. The 'table' variable is relatively uncorrelated with any of the others. 'Depth' and 'table' are negatively correlated (large purple dot), while depth is not correlated with any other variable. The strongest predictor of price is carat with length, width, depth also strongly correlated with price. Table is only very weakly correlated with price while depth is negatively correlated with price.

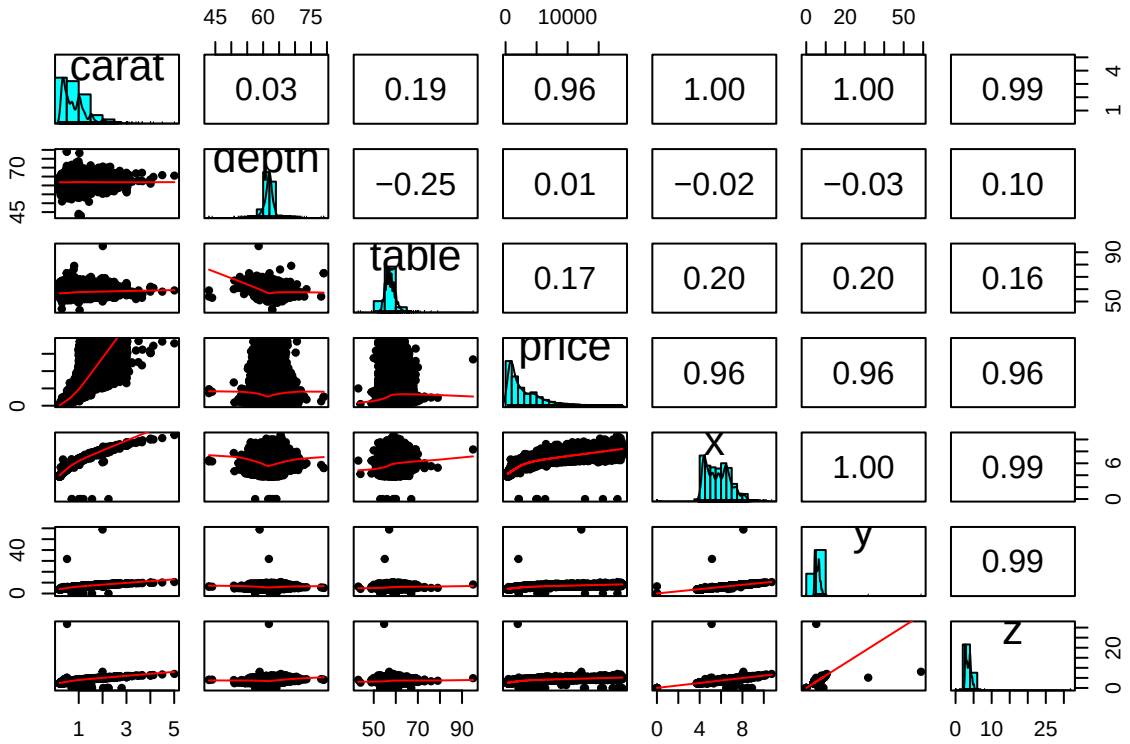

Figure 17: The pairs plot for Diamonds numeric values

Figure 17 shows the pairs plot for the numeric variables. While the scatterplots are small, it is clear that a number of pairs show little to no correlation, supporting the results from the correlation plot (figure 16).

1.14 Scatterplots

Based on the outcome of the correlation pairs plot (figure 16) we have chosen the pairs 'depth' and 'table', and 'price' and 'carat' to produce scatter plots of, because one pair shows a strong positive correlation while the other shows a strong negative correlation.

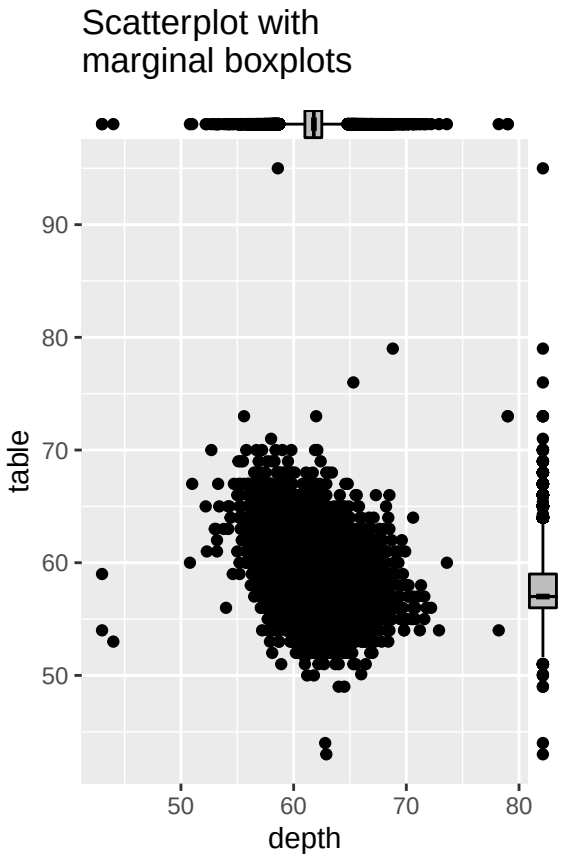

Figure 18: Scatterplot with marginal boxplots of 'depth' vs 'table'

Figure 18 shows the scatterplot with marginal boxplots of the 'depth' and 'table' variables. Because of the large number of observations, some of the visualisation is compressed to the point where it is difficult to read, for example the outliers on the marginal boxplot along the top. In the above example, the effects ratio is fixed to allow easier visualisation of the negative correlation, but this has resulted in a horizontal compression.

We can see from the marginal boxplots (grey boxes along the top and right) that most of the datapoints are clustered tightly around the medians of both variables, causing an area in the middle of the scatterplot that is so dense at to be black. There are a few outliers for each variable, but not many considering that there are over 53,000 observations. The strong negative correlation that we saw in the pairs plot is reasonably visible as evidenced by the dark directional band going from top left toward the bottom right.

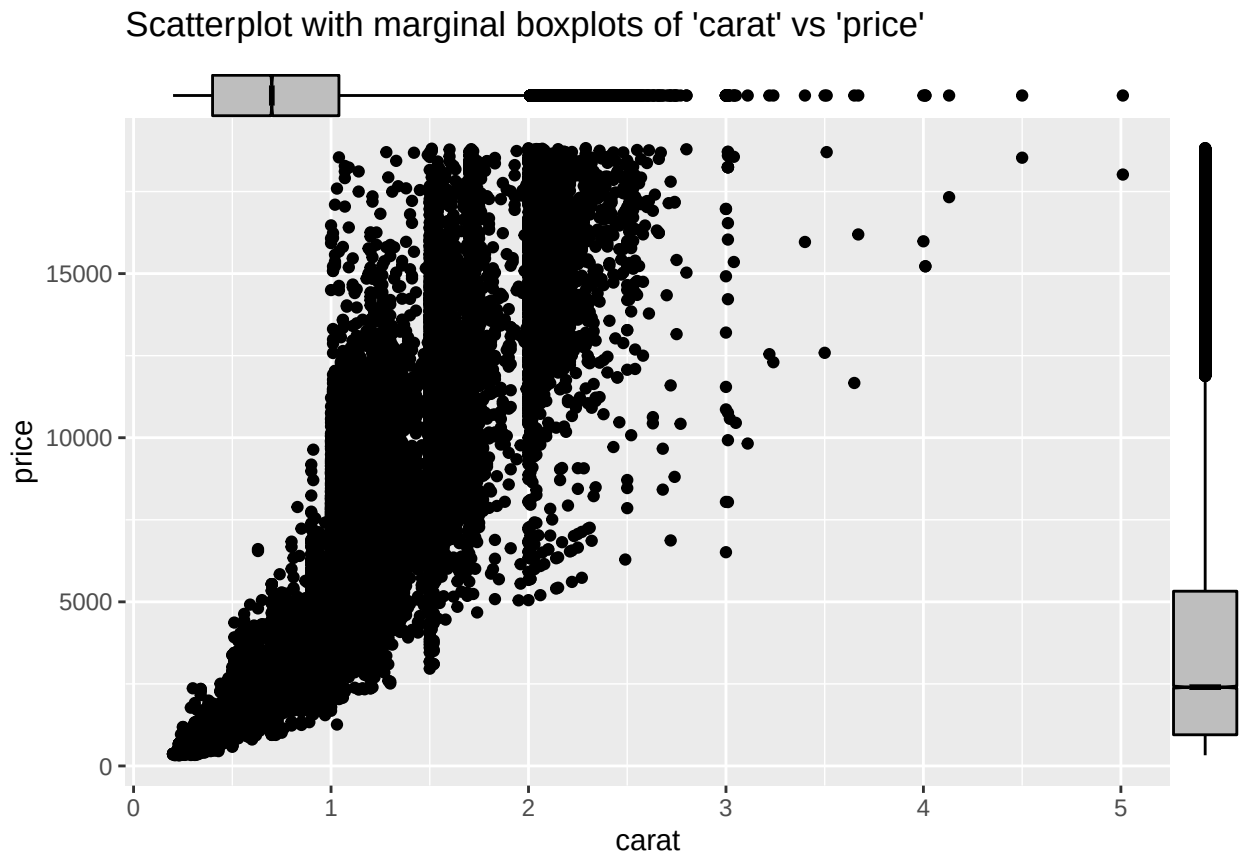

Figure 19: Scatterplot with marginal boxplots of 'carat' vs 'price'

Figure 19 shows the scatterplot with marginal boxplots of the 'carat' and 'price' variables. The expected positive correlation is clearly visible as a dark band running from the bottom left steeply towards the top right. There are vertical bands of visible at the $1, 1.5$ *and* 2 values of carat. This is a curious finding and one that is an obvious point of investigation for a more comprehensive analysis. Perhaps jewelers are in the habit of rounding down to the nearest whole or half number, despite carat being a continuous variable? Another curious aspect is why the lower parts of those ranges (from 1.5 to 1.6, for example) are so densly packed with observations, while the upper parts (1.8 to 2) appear virtually empty. It seems very unlikely that by chance there were few stones of this weight, so presumably another factor is at play.

```
## Warning: Use of `diamonds$x` is discouraged. Use `x` instead.
## Warning: Use of `diamonds$z` is discouraged. Use `z` instead.
## Warning: Use of `diamonds$x` is discouraged. Use `x` instead.
## Warning: Use of `diamonds$z` is discouraged. Use `z` instead.
## Warning: Use of `diamonds$x` is discouraged. Use `x` instead.
## Warning: Use of `diamonds$z` is discouraged. Use `z` instead.
```

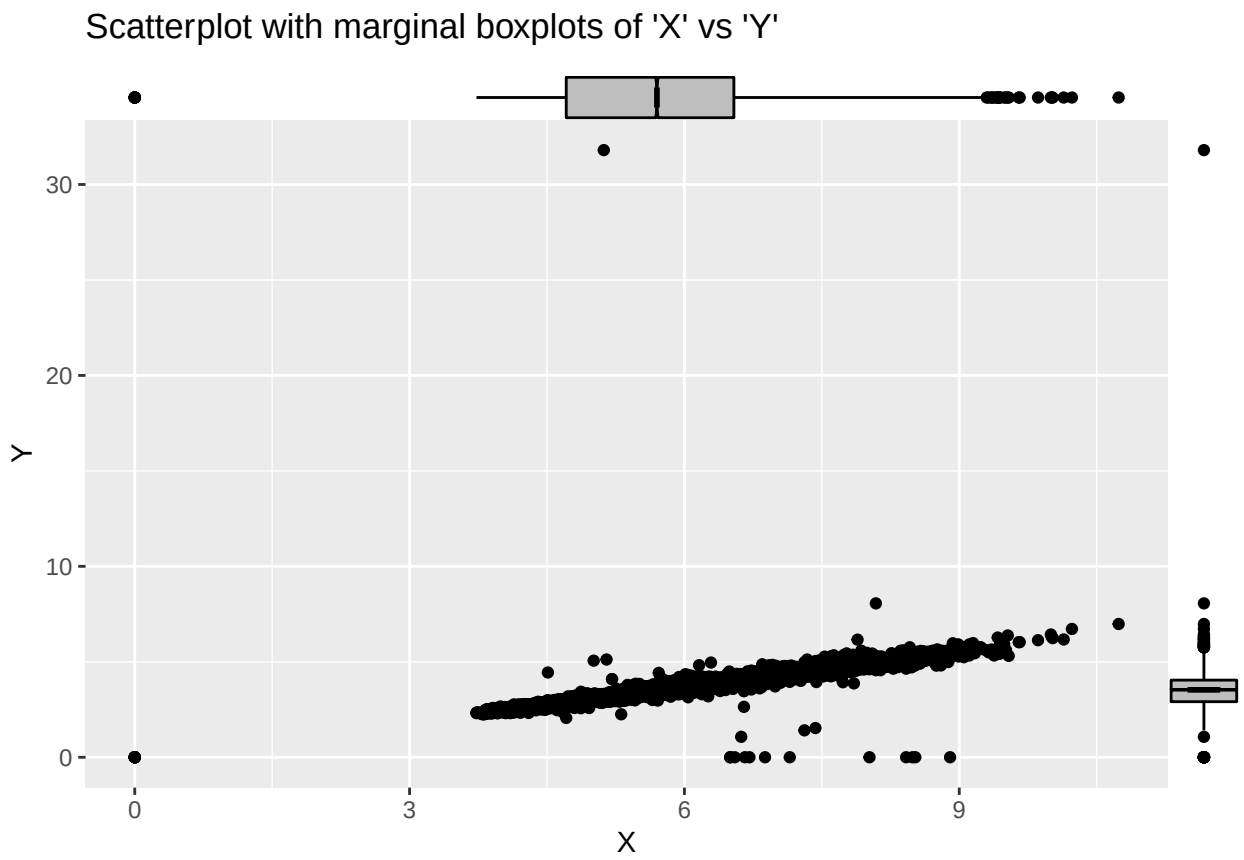

Figure **??** illustrates the strong positve correlation between the measurement dimensions of length and depth (x and z). The Correlation plot shows strong positive correlations between x (length in mm), y (width in mm) and z (depth in mm) indicating that there may be some redundancy between these variables. Of note are two extreme outliers: one in the middle at the top and the other at (0,0). The latter suggests a data entry error or a failure to measure that particular variable, as it makes no sense to include a diamond with zero width and zero depth. The outlier at the middle top could also be an error, as it does not make sense that one diamond would have a width (Y variable) three to four times the next widest, but a depth (X variable) that is quite small. The output of the code below shows that both the 'X' and 'Y' variables have eight and seven zero values respectively, while the 'Z' variable has twenty.

The code below also shows that there are two extreme upper values for 'Y', both of which are so far above normal that it is likely they are errors of some sort. There is also a suspiciously high value for 'z'.

```
# count the number of zero values for x, y and z
sum(diamonds$x==0)
```

```
## [1] 8
```

```
sum(diamonds$y==0)
```

```
## [1] 7
```

```
sum(diamonds$z==0)
```

```
## [1] 20
```

```
# display the ten largest values for x, y and z

sort(diamonds$x, decreasing = T)[1:5]
```

```
## [1] 10.74 10.23 10.14 10.02 10.01
```

```
sort(diamonds$y, decreasing = T)[1:5]
```

```
## [1] 58.90 31.80 10.54 10.16 10.10
```

```
sort(diamonds$z, decreasing = T)[1:5]
```

```
## [1] 31.80  8.06  6.98  6.72  6.43
```

```
diamonds_num[1:4,]
```

```
##   carat depth table price    x    y    z
## 1  0.23  61.5    55   326 3.95 3.98 2.43
## 2  0.21  59.8    61   326 3.89 3.84 2.31
## 3  0.23  56.9    65   327 4.05 4.07 2.31
## 4  0.29  62.4    58   334 4.20 4.23 2.63
```

1.15 Mahalanobis Distance

We will examine and display surprising points in our dataset using the Mahalanobis Distance.

```
diamonds_num$price <- as.numeric(diamonds_num$price)
mu.hat <- colMeans(diamonds_num)
sigma.hat <- cov(diamonds_num)
dM <- mahalanobis(diamonds_num, center = mu.hat, cov = sigma.hat)
upper.quantiles <- qchisq(c(.9,.95,.99), df = 7)
density.at.quantiles <- dchisq(x = upper.quantiles, df = 7)
cut.points <- data.frame(upper.quantiles, density.at.quantiles)
```

```
diamonds_num$dM <- dM
diamonds_num$surprise <- cut(diamonds_num$dM, breaks = c(0, upper.quantiles, Inf), label
table(diamonds_num$surprise)
```

```
##
##    Typical   Somewhat Surprising       very
##      49611       1001       1489       1839
```

We see from the output above that while the vast majority of diamonds are typical there are a reasonable amount of "Surprising" and "Very Surprising" points in this dataset. In terms of the 'very surprising' distances, we would expect 1% of the data to be this distant.

However, 3.4% of values lie this distant (code below). This suggests that these values do not belong to the same distribution as the rest of the data.

```
1839/length(diamonds$carat)
```

```
## [1] 0.03409344
```

1.16 Linear regression model and model equation

We have created a linear regression model with model equation:

$y = \alpha + \beta carat + \gamma cut + \tau color + \omega clarity + \epsilon$

```
lm.milestone4 <- lm(y~carat+cut+color+clarity, data = diamonds, x = T)
summary(lm.milestone4)
```

```
##
## Call:
## lm(formula = y ~ carat + cut + color + clarity, data = diamonds,
##     x = T)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.067 -0.156  0.052  0.169 50.405
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.628356   0.016682 217.501  < 2e-16 ***
## carat        2.290701   0.003610 634.558  < 2e-16 ***
## cutGood      0.110681   0.010090  10.970  < 2e-16 ***
## cutVery Good 0.134532   0.009383  14.338  < 2e-16 ***
## cutPremium   0.109488   0.009279  11.800  < 2e-16 ***
## cutIdeal     0.130812   0.009196  14.224  < 2e-16 ***
## colorI       0.018082   0.008087   2.236 0.025356 *
## colorH       0.045873   0.007638   6.006 1.92e-09 ***
## colorG       0.072076   0.007452   9.672  < 2e-16 ***
## colorF       0.085127   0.007601  11.199  < 2e-16 ***
## colorE       0.058897   0.007644   7.705 1.33e-14 ***
## colorD       0.054016   0.008016   6.738 1.62e-11 ***
## claritySI2   0.129426   0.013436   9.633  < 2e-16 ***
## claritySI1   0.137995   0.013379  10.315  < 2e-16 ***
## clarityVS2   0.102886   0.013452   7.649 2.07e-14 ***
## clarityVS1   0.109915   0.013660   8.047 8.69e-16 ***
## clarityVVS2  0.051763   0.014066   3.680 0.000234 ***
## clarityVVS1  0.007183   0.014462   0.497 0.619446
## clarityIF    0.014973   0.015640   0.957 0.338399
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.347 on 53921 degrees of freedom
## Multiple R-squared:  0.9077, Adjusted R-squared:  0.9077
## F-statistic: 2.946e+04 on 18 and 53921 DF,  p-value: < 2.2e-16
```

```
#pchisq(lm.milestone4$, df=lm.milestone4$, lower.tail=FALSE)
```

The output above from the linear regression model shows that most of the variables appear to be useful in predicting price (based on their p-values) except for clarityVVS1 and ClarityIF.

"Diamonds Dataset, Kaggle.com." 2016. https://www.kaggle.com/datasets/shivam2503/diamonds.