STAT394 Group Project Milestone 3: Exploratory Data Analysis (EDA)

Ken MacIver, Tom Tribe, Jundi Yang, Mei Huang

2022-08-21

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)
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 eleven 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)).

1.0.1 Load the dataset into R.

```
diamonds <- read.csv("./diamonds.csv", encoding = "UTF-8")
diamonds[1:5,]
```

```
##   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
```

```
# remove the index column
diamonds$X <- NULL

# get an overview of the structure of the data
str(diamonds)
```

```
## '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

```
# set categorical variables as factors and set levels
diamonds$cut <- factor(diamonds$cut,
                       levels = c("Fair","Good","Very Good","Premium","Ideal"))
diamonds$color <- factor(diamonds$color,
                         levels = c("J","I","H","G","F","E","D"))
diamonds$clarity <- factor(diamonds$clarity,
                           levels = c("I1","SI2","SI1","VS2","VS1","VVS2","VVS1","IF"))

# make data frame of just the numerical variables
diamonds_num <- subset(diamonds, select = c(carat,depth,table,price,x,y,z))

# display first rows to check
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
```

```
# display summary data
summary(diamonds_num)
```

```
##      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
```

```r
# use the summary function to create the summary data for the numerical variables
MySummary <- function(x){
  return(c(
    length(x),
    min(x),
    quantile(x, .25),
    median(x),
    mean(x),
    quantile(x, .75),
    max(x),
    IQR(x),
    sd(x),
    skewness(x),
    kurtosis(x)))
}
```

1.1.0.1 Summary function diamonds

```r
# store the summary data in a variable and modify row names for the output table
summ_diamonds <- apply(diamonds_num, MySummary, MARGIN=2)

rownames(summ_diamonds) <- c("sample size","minimum",
                                    "first quartile","median",
                                    "mean","third quartile",
                                    "maximum","IQR", "standard deviation",
                               "skewness","kurtosis")
```

1.1.0.2 Change array rownames

1.1.1 Summary table diamonds

```r
# produce summary table
knitr::kable(signif(summ_diamonds,2), caption = "Summary statistics for 'diamonds' (2 s.
```

Table 1: Summary statistics for 'diamonds' (2 s.f.)

	carat	depth	table	price	x	y	z
sample size	5.4e+04	5.4e+04	54000.0	54000.0	5.4e+04	54000.0	5.4e+04
minimum	2.0e-01	4.3e+01	43.0	330.0	0.0e+00	0.0	0.0e+00
first quartile	4.0e-01	6.1e+01	56.0	950.0	4.7e+00	4.7	2.9e+00
median	7.0e-01	6.2e+01	57.0	2400.0	5.7e+00	5.7	3.5e+00
mean	8.0e-01	6.2e+01	57.0	3900.0	5.7e+00	5.7	3.5e+00
third quartile	1.0e+00	6.2e+01	59.0	5300.0	6.5e+00	6.5	4.0e+00
maximum	5.0e+00	7.9e+01	95.0	19000.0	1.1e+01	59.0	3.2e+01
IQR	6.4e-01	1.5e+00	3.0	4400.0	1.8e+00	1.8	1.1e+00
standard deviation	4.7e-01	1.4e+00	2.2	4000.0	1.1e+00	1.1	7.1e-01
skewness	1.1e+00	-8.2e-02	0.8	1.6	3.8e-01	2.4	1.5e+00
kurtosis	4.3e+00	8.7e+00	5.8	5.2	2.4e+00	94.0	5.0e+01

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

1.2 Melted version of dataset

```
# create melted version of the dataset to allow easy graphical manipulation
diamonds_melt <- melt(data = diamonds, id.vars = c("cut","color","clarity"),
                      variable.name = "metrics")
```

1.3 Boxplots of 'cut'

```
ggplot(data=diamonds_melt, aes(x=metrics, y=value)) +

  geom_boxplot(aes(col=metrics), notch = TRUE) +
  facet_grid(~ cut) +
  theme(axis.text.x = element_text(size=7, angle=90, hjust=1),
        legend.position = "none")
```

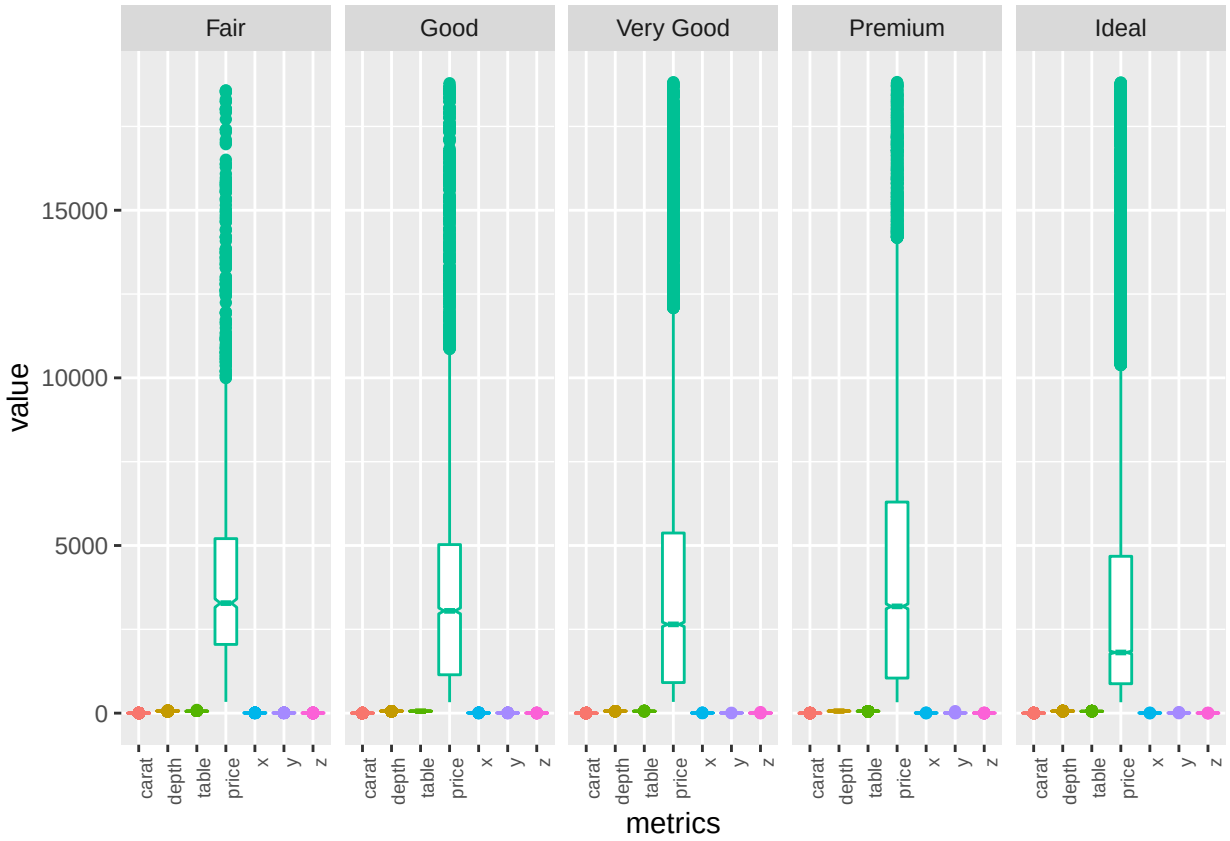


Figure 1: Boxplots of 'cut' vs all the numeric variables

Figure 1 shows that all the variables except 'price' are too compressed to view. Therefore, I will perform a log transform and redo the graph.

1.4 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.

```
ggplot(data=diamonds_melt, aes(x=metrics, y=value)) +

  geom_boxplot(aes(col=metrics), notch = TRUE) +
  scale_y_log10()+
  facet_grid(~ cut) +
  theme(axis.text.x = element_text(size=7, angle=90, hjust=1),
        legend.position = "none")
```

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

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

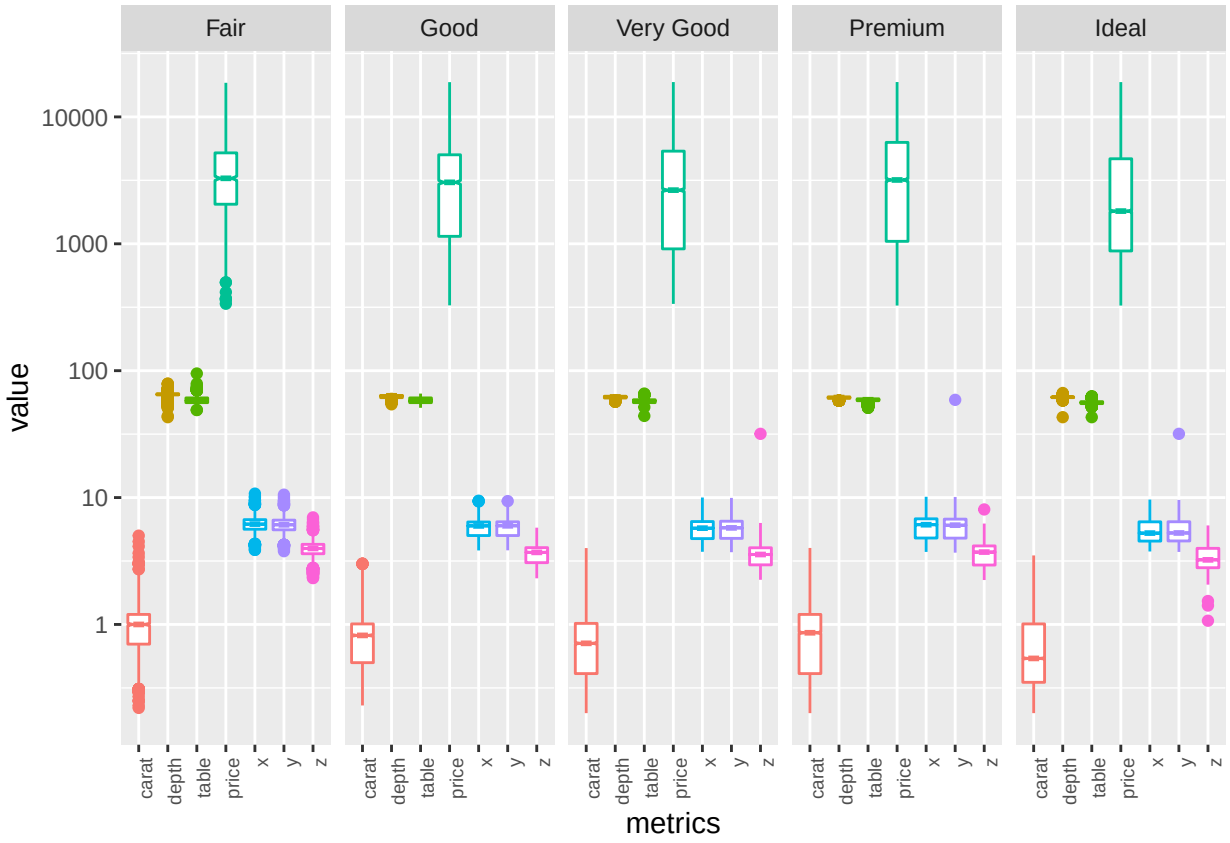


Figure 2: Boxplots of 'cut' vs all the numeric variables (log transformed)

The log transform in Figure 2 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. For a full analysis this would be one place to begin hypothesis testing.

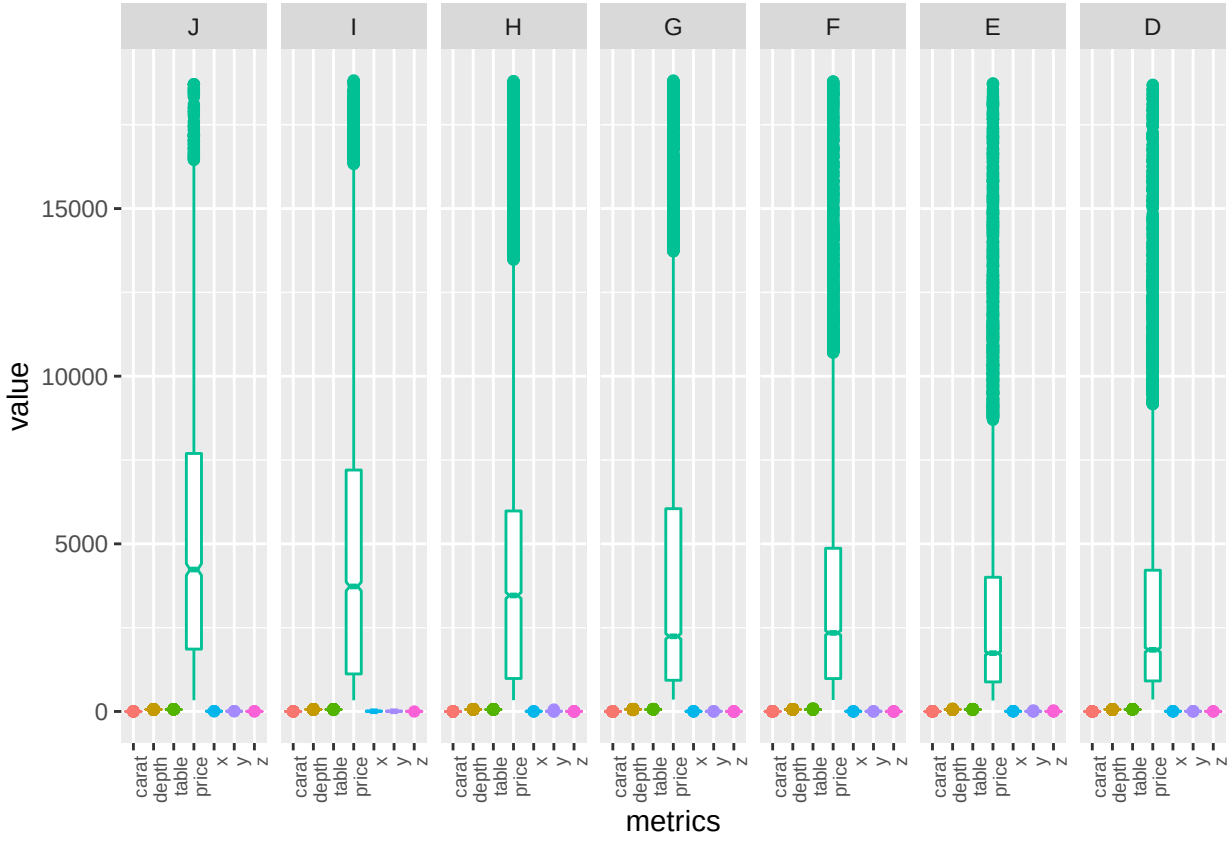
For two of the variables (measurements 'y', purple, and 'z' pink) in the 'Very Good', 'Premium' and 'Ideal' levels of 'cut' there appear to be some very prominent outliers, as evidenced by the pink and purple dots above and below the boxplots. The variable 'y' is a measure of width in millimeters (mm),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

1.5 Boxplots of 'color'

```
ggplot(data=diamonds_melt, aes(x=metrics, y=value)) +

  geom_boxplot(aes(col=metrics), notch = TRUE) +
  facet_grid(~ color) +
```

```
  theme(axis.text.x = element_text(size=7, angle=90, hjust=1),
        legend.position = "none")
```



1.6 Boxplots of 'color' in log scale

As with the 'cut' variable, we redo the boxplots using the log transform on the data.

```
ggplot(data=diamonds_melt, aes(x=metrics, y=value)) +

  geom_boxplot(aes(col=metrics), notch = TRUE) +
  scale_y_log10()+
  facet_grid(~ cut) +
  theme(axis.text.x = element_text(size=7, angle=90, hjust=1),
        legend.position = "none")
```

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

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

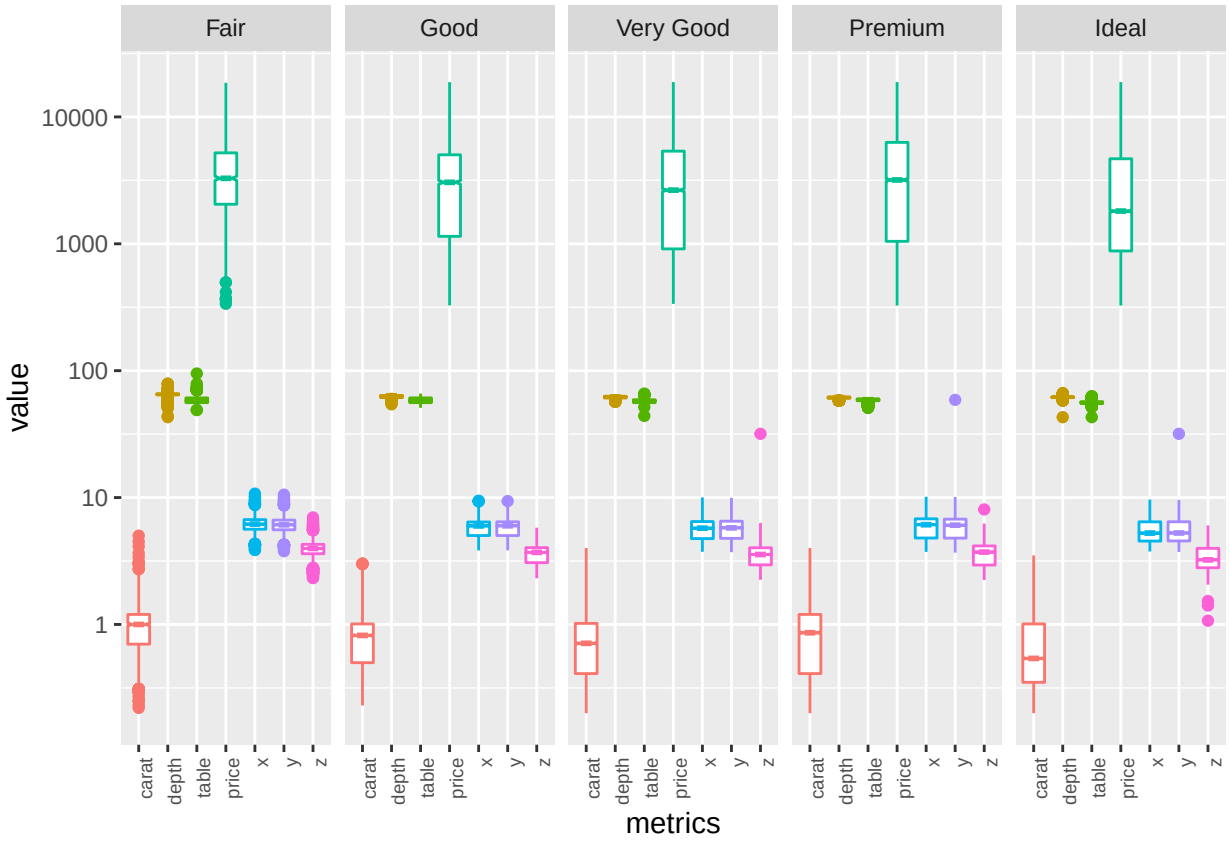


Figure 3: Boxplots of 'color' vs all the numeric variables (log transformed)

Figure 3 shows the log transformed version of the boxplots for the 'color' variable.

1.7 Boxplots of 'clarity'

For the 'clarity' variable I have immediately performed a log transform on the data for the boxplots.

```
ggplot(data=diamonds_melt, aes(x=metrics, y=value)) +

  geom_boxplot(aes(col=metrics), notch = TRUE) +
  scale_y_log10()+
  facet_grid(~ clarity) +
  theme(axis.text.x = element_text(size=7, angle=90, hjust=1),
        legend.position = "none")
```

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

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

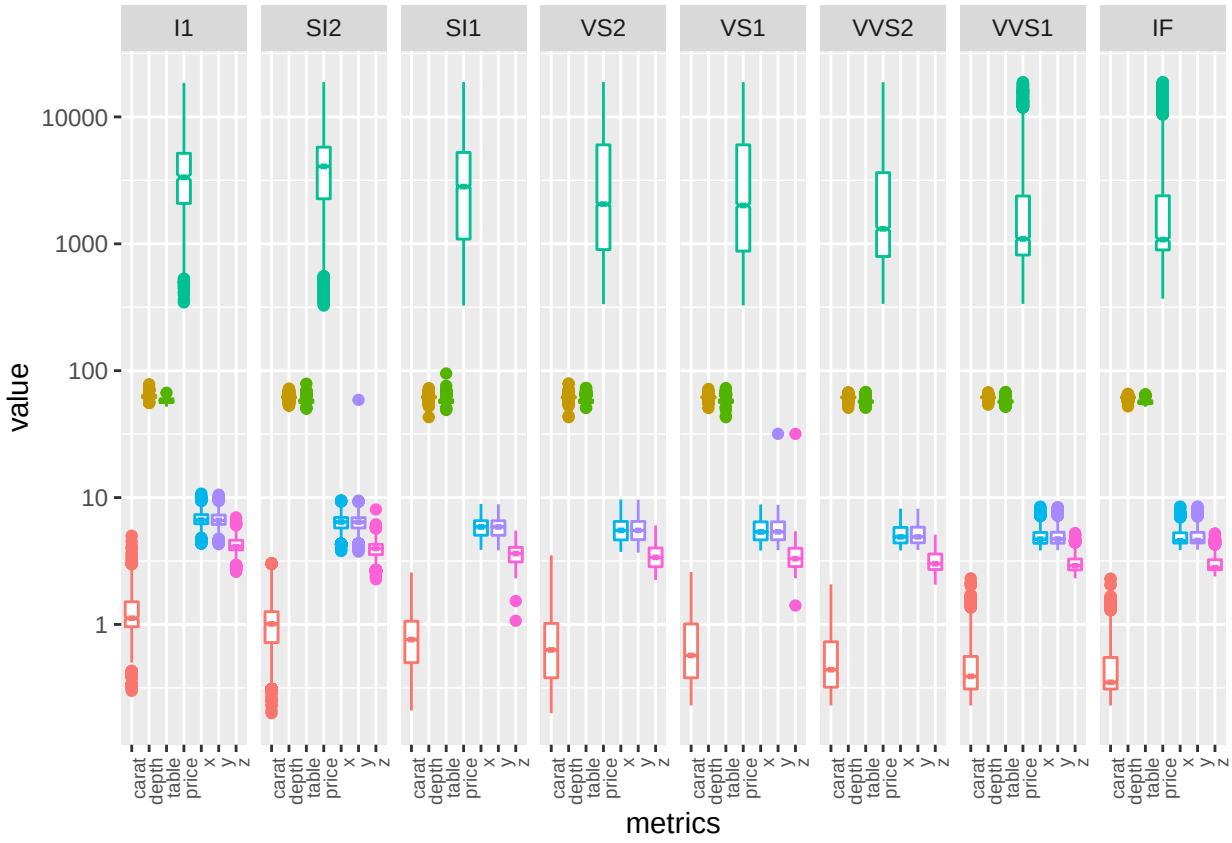


Figure 4: Boxplots of 'clarity' vs all the numeric variables (log transformed)

Figure 4 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.

1.7.1 Means vector

```
means_vec_diam <- matrix(colMeans(diamonds_num),
                         nrow = length(colnames))
signif(means_vec_diam,4)
```

```
##         [,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:

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

1.7.2 The covariance matrix

```
covar_diam <- cov(diamonds_num)
knitr::kable(signif(covar_diam,2), caption = "Covariance matrix for 'diamonds' (2 s.f.)"
```

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.

1.7.3 The correlation matrix

```
correl_diam <- cor(diamonds_num)


knitr::kable(signif(correl_diam,2), caption = "Correlation matrix for 'diamonds' (2 s.f.
```

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.

1.8 Visualisation of the correlation matrix

```
ggcorrplot(cor(diamonds_num),
           method = "circle",
           hc.order = TRUE,
           type = "lower")
```

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

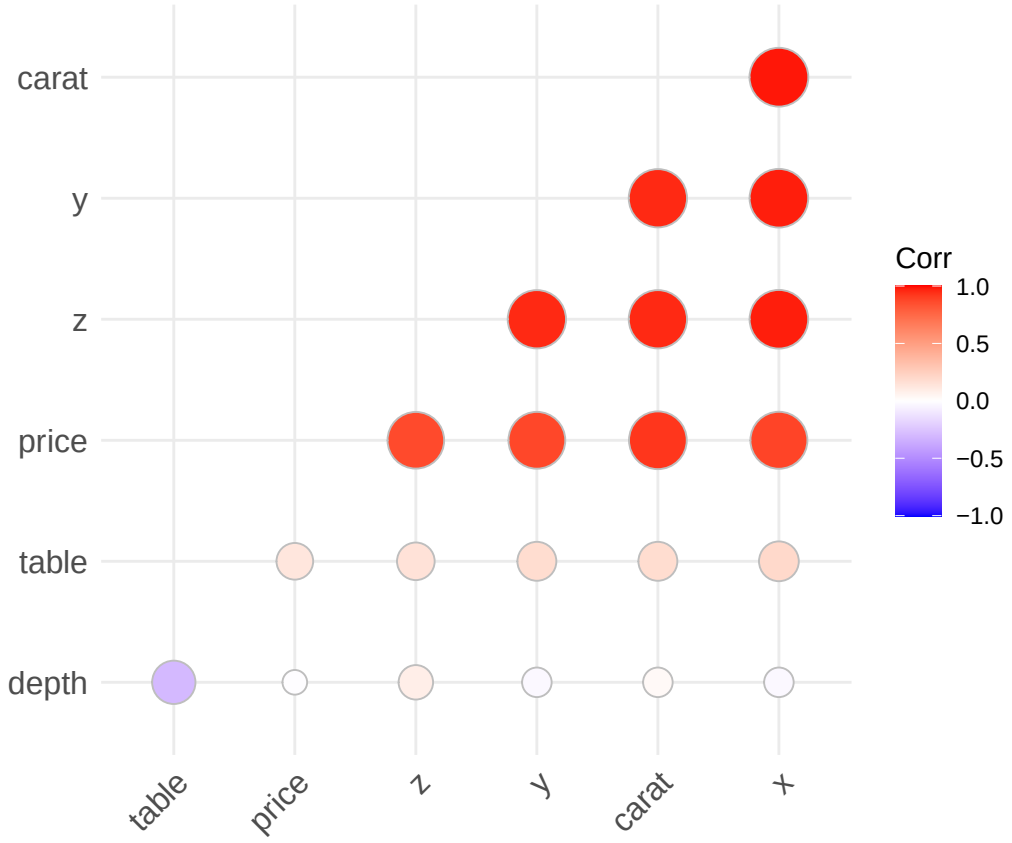


Figure 5: The pairs plot for diamonds

Figure 5 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. 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.

It seems obvious that the diamonds' weight, length, width, depth are all related to its size and size is the main predictor for price.

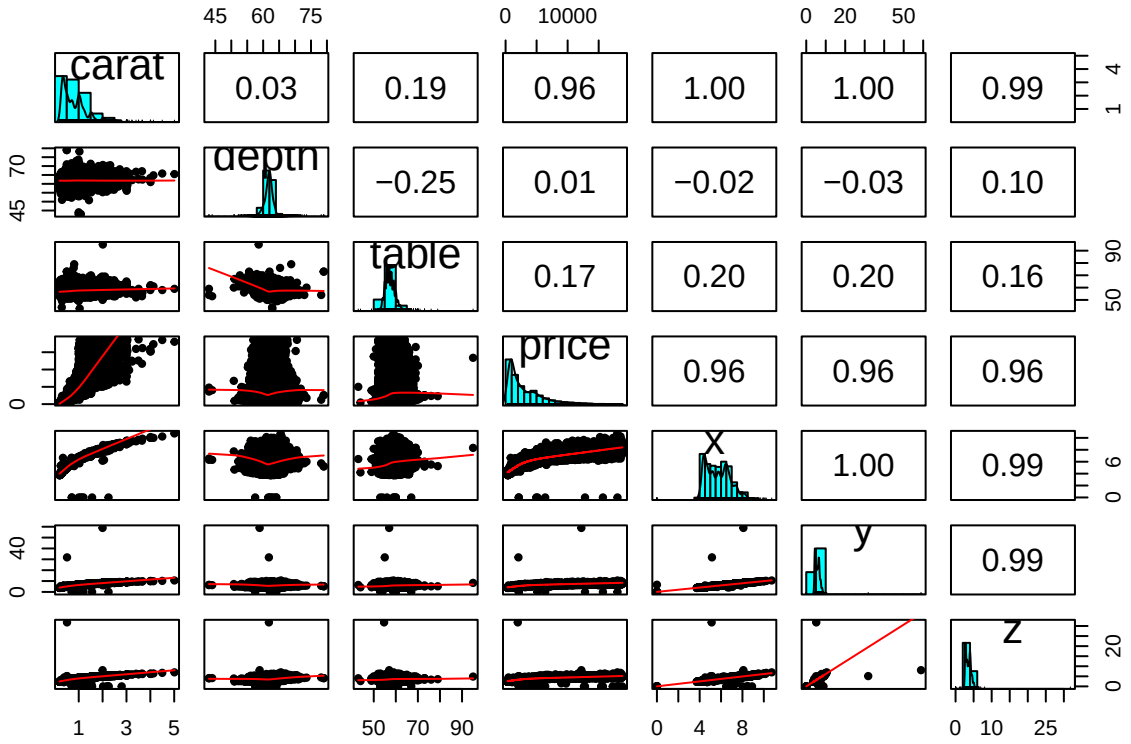


Figure 6: The pairs plot for Diamonds numeric values

Figure 6 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 5).

1.9 Scatterplots

Based on the outcome of the correlation pairs plot (figure 5) we have chosen the pairs 'depth' and 'table', and 'price' and 'carat' to produce scatterplots of, because one pair shows a strong positive correlation while the other shows a strong negative correlation.

```
diam_deptab <- ggplot(diamonds, aes(x=depth, y=table))+
  geom_point()+
  coord_fixed()+

  labs(title = "Scatterplot with \nmarginal boxplots")

ggMarginal(diam_deptab, type = "boxplot", notch=TRUE, size = 15,
           fill="grey")
```

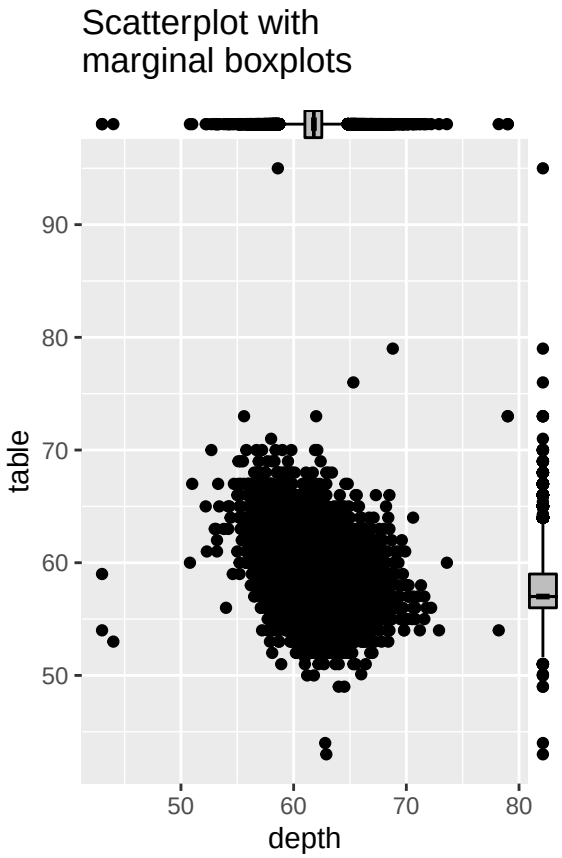


Figure 7: Scatterplot with marginal boxplots of 'depth' vs 'table'

Figure 7 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.

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


  labs(title = "Scatterplot with marginal boxplots of 'carat' vs 'price'")

ggMarginal(diam_deptab, type = "boxplot", notch=TRUE, size = 15,
```

```
          fill="grey")
```

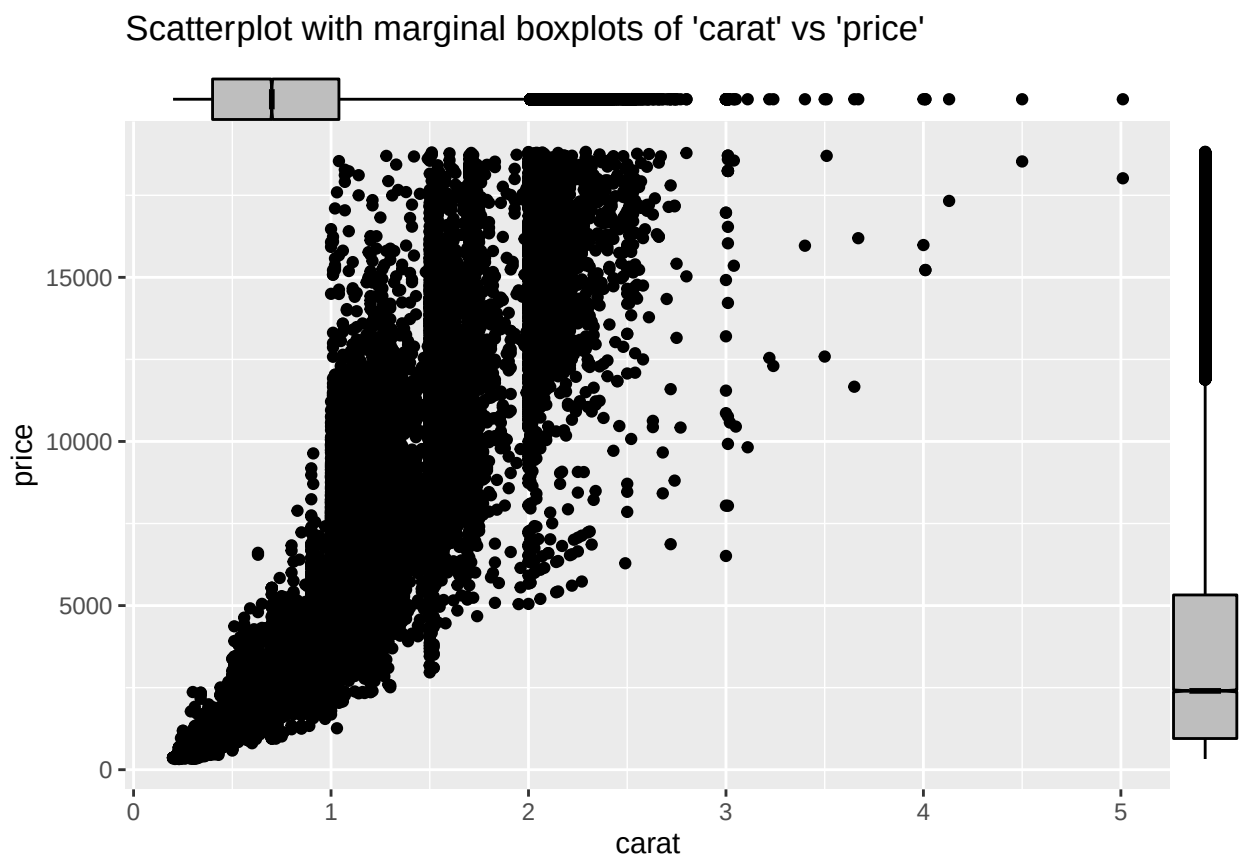


Figure 8: Scatterplot with marginal boxplots of 'carat' vs 'price'

Figure 8 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.

1.10 Cullen and Frey graphs

From the Cullen and Frey plots, we can see the distribution of carat (Figure 9), price (Figure 12), x (Figure 13) appear to follow the beta distribution. While the distribution of depth (Figure 10), table (Figure 11), y (Figure 14), z (Figure 15) are not clear.

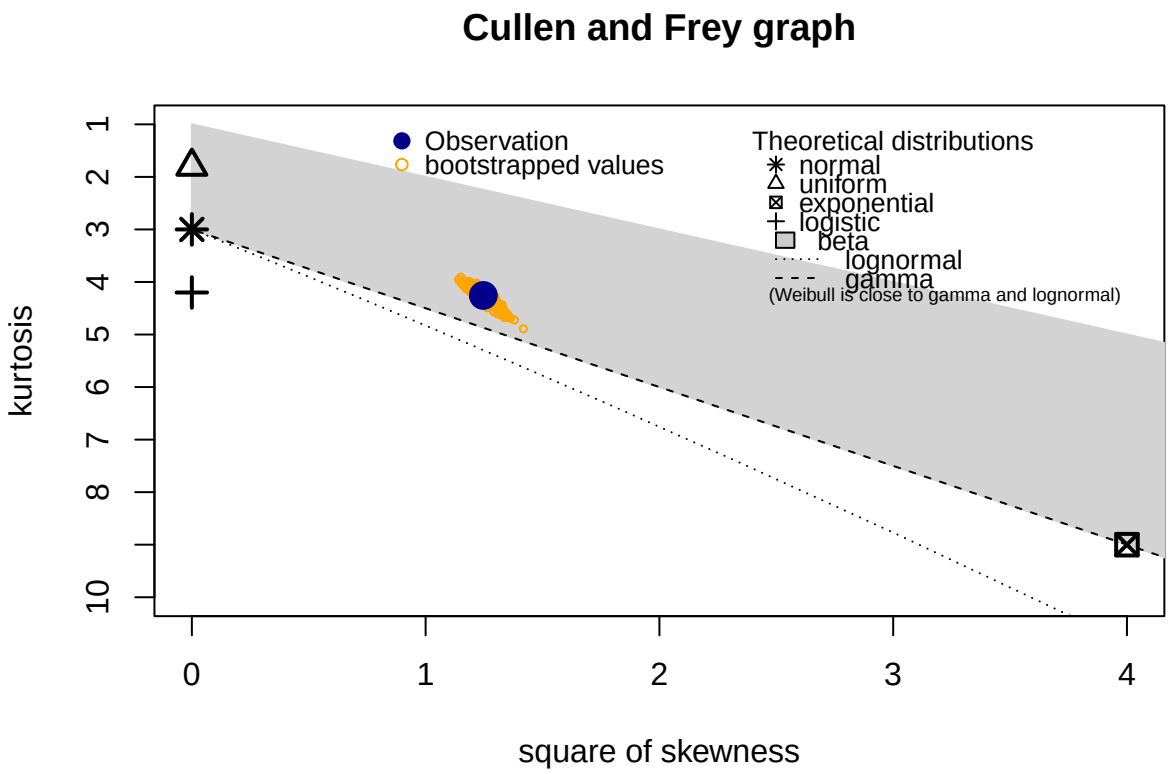


Figure 9: Cullen & Frey plot for carat

```
## summary statistics
## ------
## min:  0.2   max:  5.01
## median:  0.7
## mean:  0.7979397
## estimated sd:  0.4740112
## estimated skewness:  1.116646
## estimated kurtosis:  4.256635
```

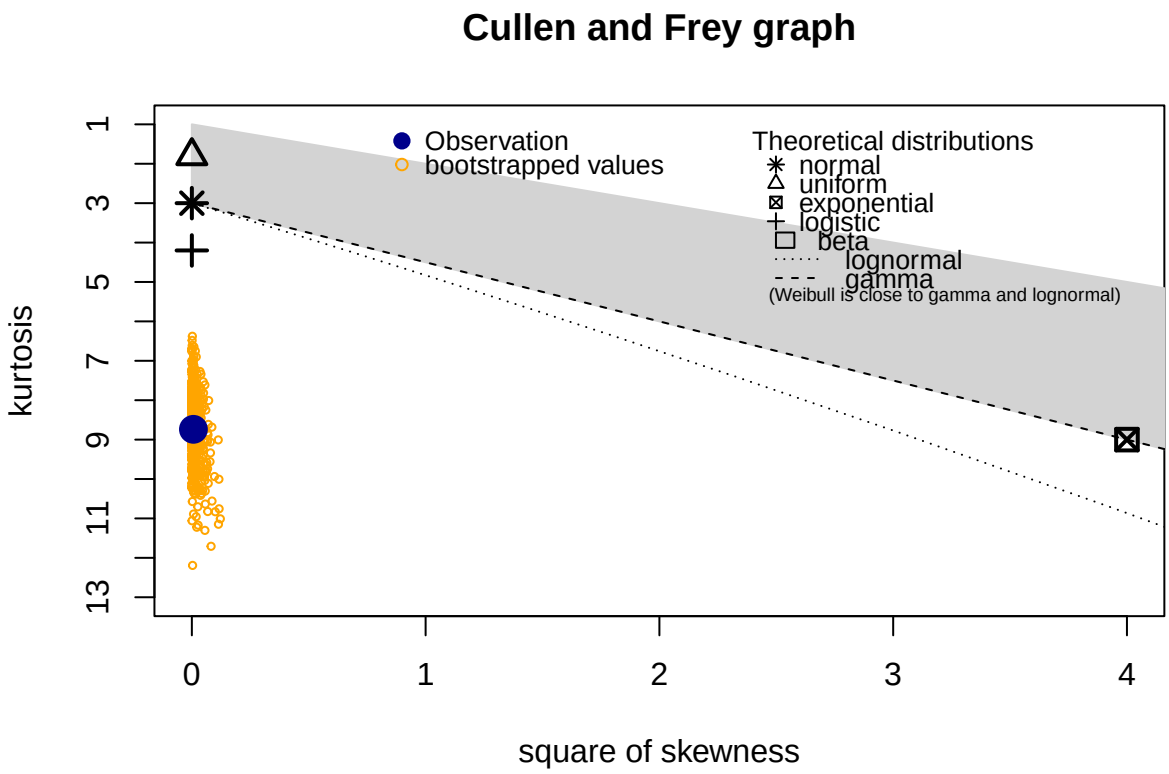


Figure 10: Cullen & Frey plot for depth

```
## summary statistics
## ------
## min:  43   max:  79
## median:  61.8
## mean:  61.7494
## estimated sd:  1.432621
## estimated skewness:  -0.08229403
## estimated kurtosis:  8.739415
```

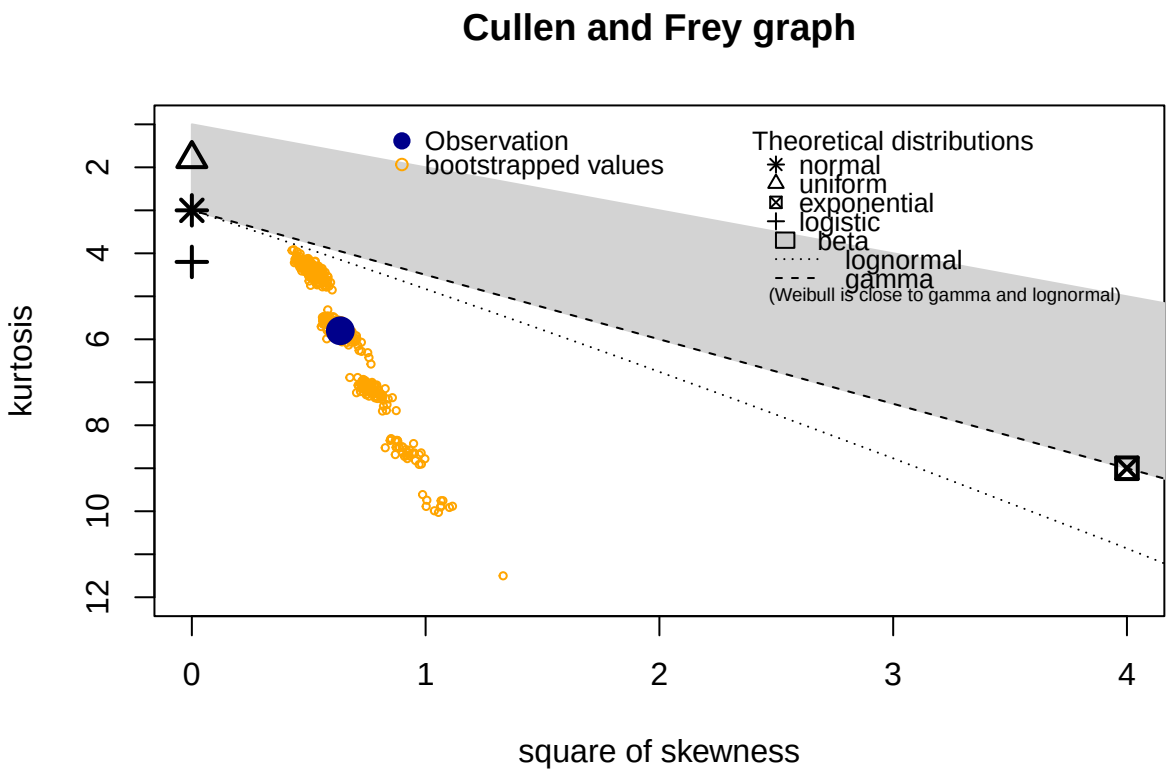


Figure 11: Cullen & Frey plot for table

```
## summary statistics
## ------
## min:  43   max:  95
## median:  57
## mean:  57.45718
## estimated sd:  2.234491
## estimated skewness:  0.7968958
## estimated kurtosis:  5.801857
```

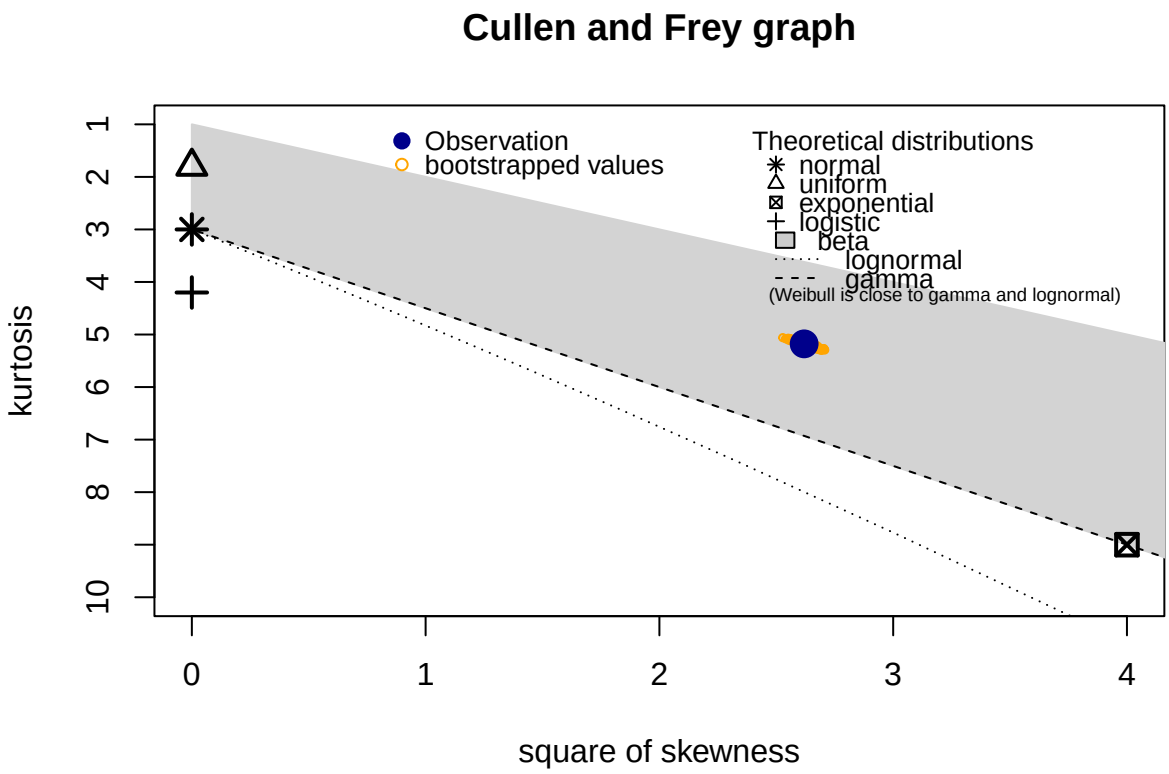


Figure 12: Cullen & Frey plot for price

```
## summary statistics
## ------
## min:  326   max:  18823
## median:  2401
## mean:  3932.8
## estimated sd:  3989.44
## estimated skewness:  1.618395
## estimated kurtosis:  5.177696
```

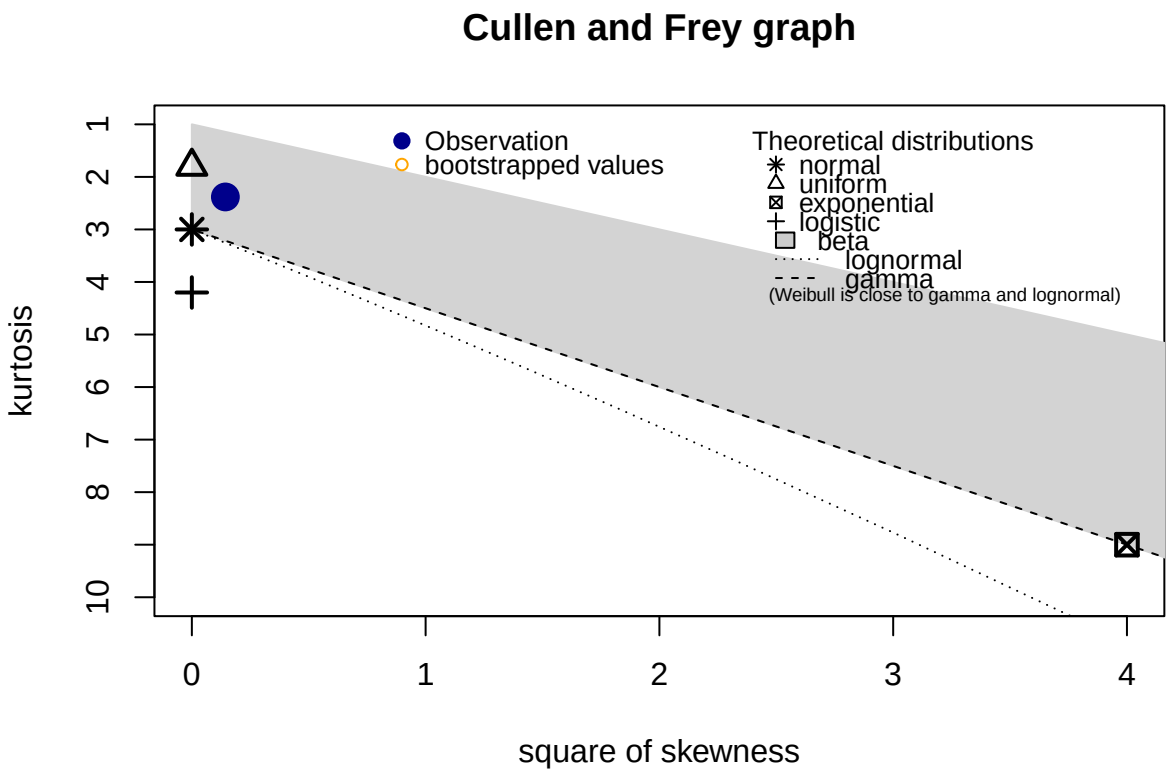


Figure 13: Cullen & Frey plot for x

```
## summary statistics
## ------
## min:  0   max:  10.74
## median:  5.7
## mean:  5.731157
## estimated sd:  1.121761
## estimated skewness:  0.3786763
## estimated kurtosis:  2.381839
```

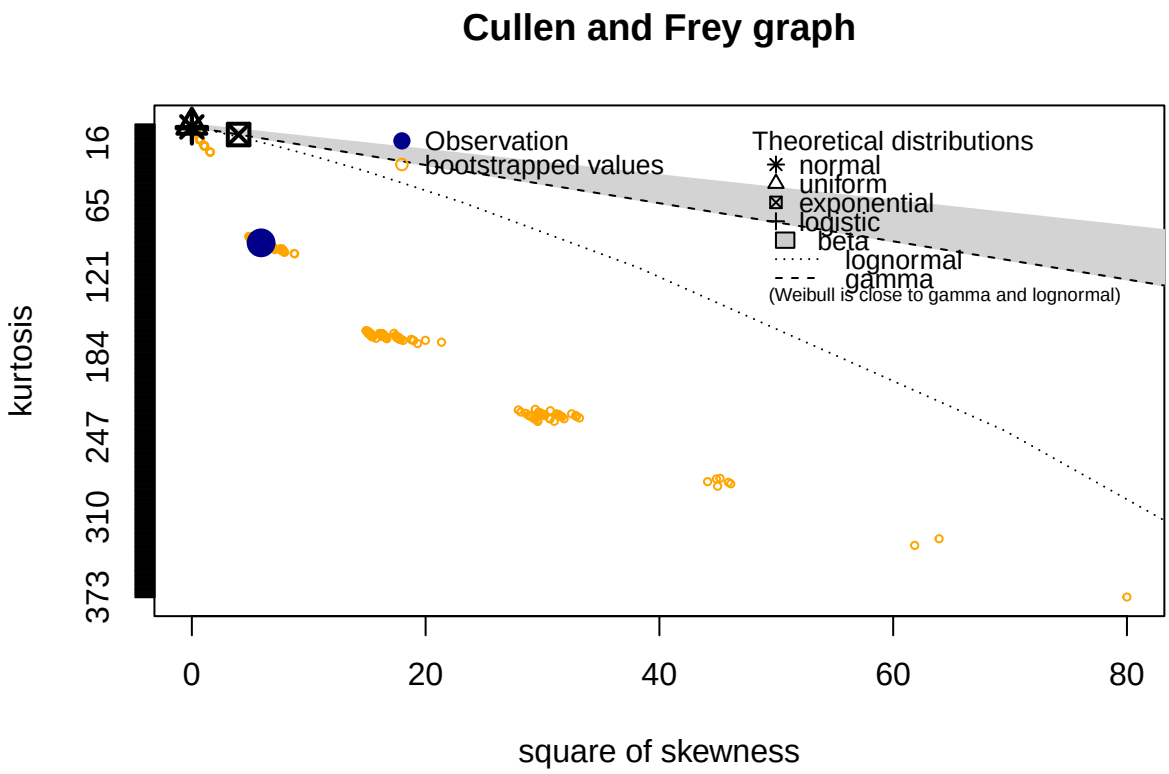


Figure 14: Cullen & Frey plot for y

```
## summary statistics
## ------
## min:  0   max:  58.9
## median:  5.71
## mean:  5.734526
## estimated sd:  1.142135
## estimated skewness:  2.434167
## estimated kurtosis:  94.21456
```

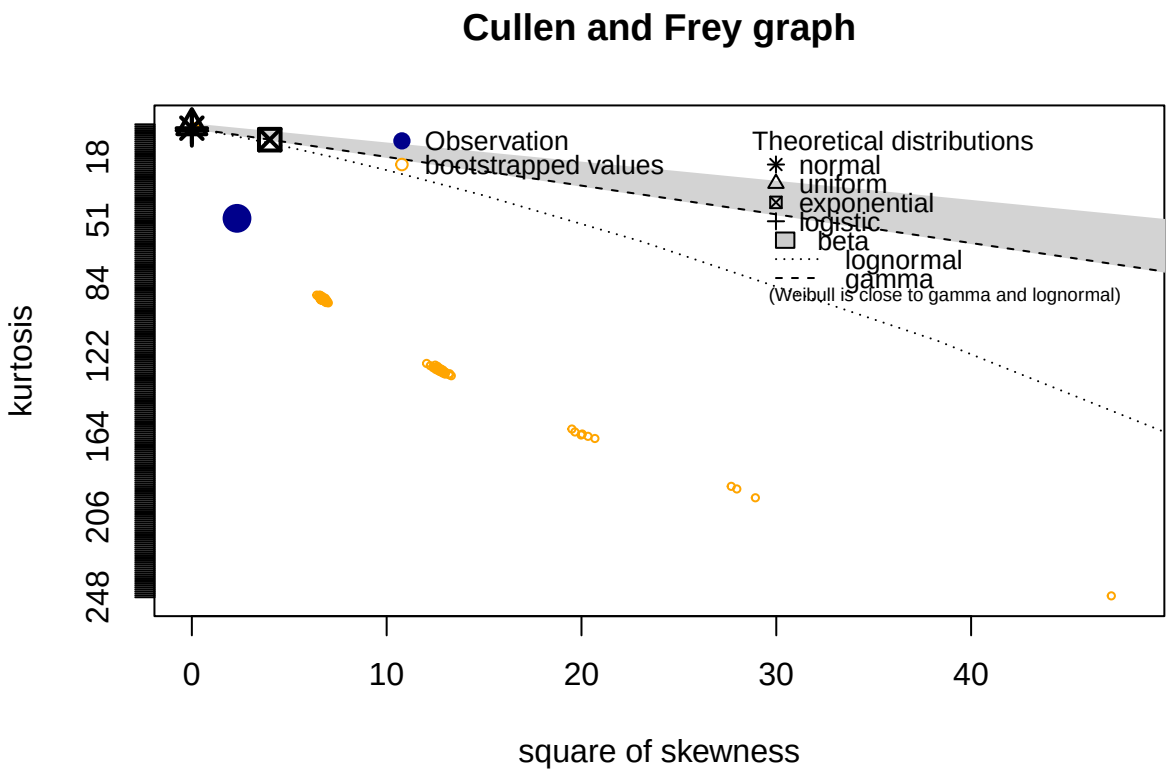


Figure 15: Cullen & Frey plot for z

```
## summary statistics
## ------
## min:  0   max:  31.8
## median:  3.53
## mean:  3.538734
## estimated sd:  0.7056988
## estimated skewness:  1.522423
## estimated kurtosis:  50.08662
```

References

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