Coding Project Part 1

Making Use of the ‘Diamond Prices’ Dataset

Jake Merry

11.06.24

Contents

Introduction

For this project we will be analyzing the “Diamonds Prices” dataset from Kaggle. Our goal for this project is to analyze the dataset in order to explore relationships between the variables and provide statistical summaries for a sample of the data.

This dataset contains information on almost 54,000 diamonds. Let’s import our dataset:

```
library(skimr)
diamondData <- read_csv("cd1_dataset.csv")
```

In order to get a better initial understanding of the data, we can simply examine the first few rows of the dataset by use of the `head` function.

```
head(diamondData)
```

```
## # A tibble: 6 x 11
##   index carat cut       color clarity depth table price     x     y     z
##   <dbl> <dbl> <chr>     <chr> <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2     2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3     3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4     4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5     5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6     6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

We see that there are 11 fields each corresponding to a different diamond characteristic. Let’s take a closer look at each field individually and what they represent by using the `names` function to return each column name from the dataset.

```
names(diamondData)
```

```
##  [1] "index"   "carat"   "cut"     "color"   "clarity" "depth"   "table"  
##  [8] "price"   "x"       "y"       "z"      
```

- `Index`: This field is a record of the observation number of each diamond. Each entry in this field is a unique positive integer, numbered 1 through 53943.
- `Carat`: This is a measure of the weight of each diamond. Each carat is equal to exactly 200 milligrams or 1/5 of a gram.
- `Cut`: This field gives the cut grade of each diamond. There are five different grades for diamonds in this dataset. From worst grade to best grade, they are given as `fair`, `good`, `very good`, `premium`, and `ideal`.
- `Color`: For this dataset, diamonds are categorized as having one of seven colors: `D`, `E`, `F`, `G`, `H`, `I`, or `J`. Diamond color is organized in alphabetical order, where letters closer to the beginning of the alphabet are more colorless than letters closer to the end of the alphabet. These grades are also more broadly grouped where `D`, `E`, and `F` represent colorless diamonds and `G`, `H`, `I`, and `J` represent nearly colorless diamonds.
- `Clarity`: Diamond clarity represents the amount - or lack - of impurities visible on the surface of and in the interior of each diamond. Clarity is classified under 8 different fields, ranked from least clear to most clear as follows: `I1`, `SI2`, `SI1`, `VS2`, `VS1`, `VVS2`, `VVS1`, and `IF`.
- `Depth`: The depth of a diamond is the measurement of the proportion of the height to the total width of the diamond.
- `Table`: The table value associated with each diamond is the proportion of the width of the flat top to the total width of the diamond.
- `Price`: This field is exactly as it sounds. This is some price, in US dollars, associated with each diamond.
- `x`: This field represents the length of each diamond.
- `y`: This field represents the width of each diamond.
- `z`: This field represents the depth of each diamond.

Data Analysis

For use of this project, we will use the sample composed of the 500 randomly selected rows from the dataset, and we will examine the fields `carat`, `color`, `clarity`, `depth`, `table`, and `price`. We can create a variable called `sample` that retains only these data points for ease of analysis.

```
set.seed(5)

diamondSample <- diamondData[
  sample(1:nrow(diamondData), 500),
  c("carat", "color", "clarity", "depth", "table", "price")
]
diamondSample
```

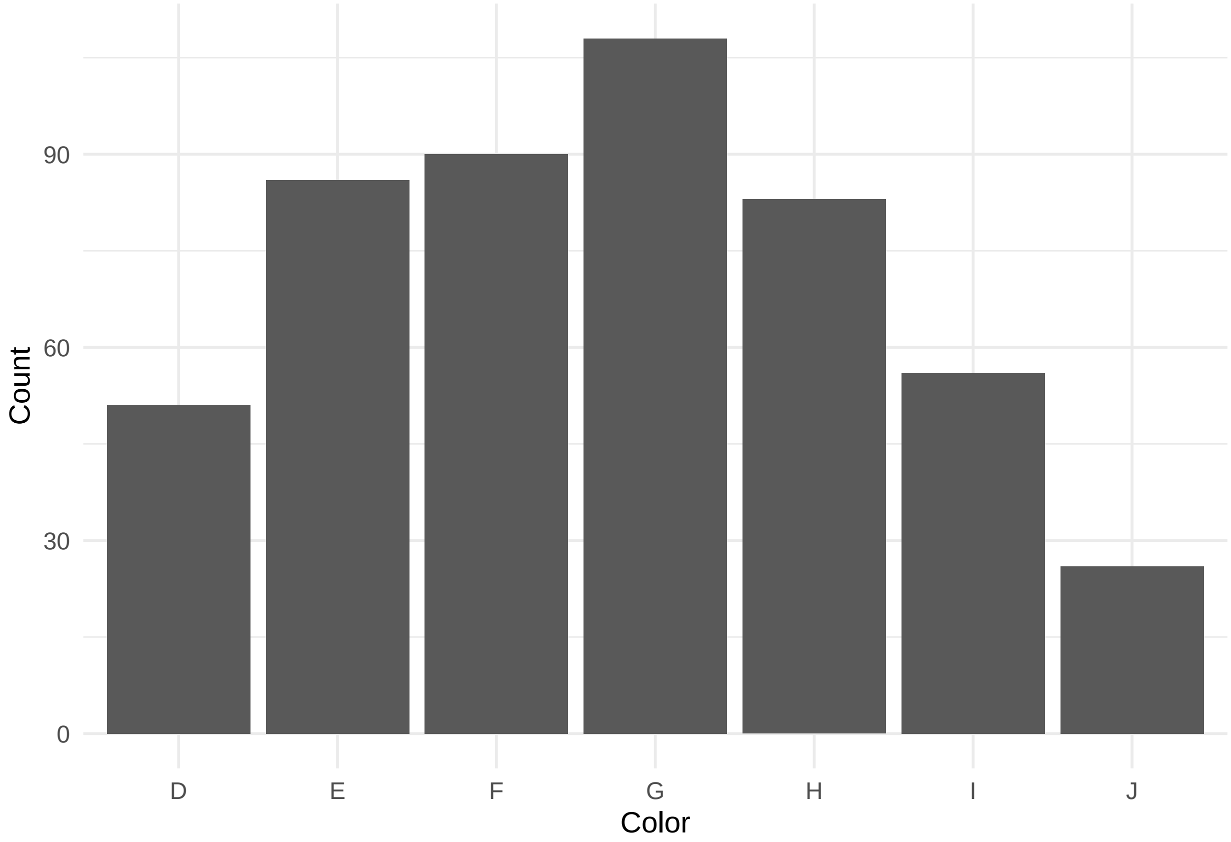
```
## # A tibble: 500 x 6
##    carat color clarity depth table price
##    <dbl> <chr> <chr>   <dbl> <dbl> <dbl>
##  1  0.53 F     VS1      61.7    56  1630
##  2  1.46 H     SI2      61.4    59  7604
##  3  0.56 H     VVS2     59.8    57  1723
##  4  0.7  G     VS2      61.8    54  2593
##  5  1.13 I     SI2      59.9    57  4195
##  6  0.32 F     VVS2     62      55   786
##  7  0.43 F     SI1      63.6    53   948
##  8  1.29 J     VS1      61.7    59  5463
##  9  0.71 I     VS2      64      59  1840
## 10  1.09 H     VS1      60.2    61  5951
## # i 490 more rows
```

In order to give statistical summaries of the fields of interest, we must note that the `color` and `clarity` fields are purely qualitative and do not yield detailed numerical summaries, so we can instead examine a frequency table and barplot for each variable in order to conduct analyses.

```
colorTable <- table(diamondSample$color)
colorPTable <- prop.table(colorTable)
data.frame(
  Color = names(colorTable),
  Count = as.vector(colorTable),
  Percentage = round(as.vector(colorPTable), 2)
)
```

```
##   Color Count Percentage
## 1     D    51       0.10
## 2     E    86       0.17
## 3     F    90       0.18
## 4     G   108       0.22
## 5     H    83       0.17
## 6     I    56       0.11
## 7     J    26       0.05
```

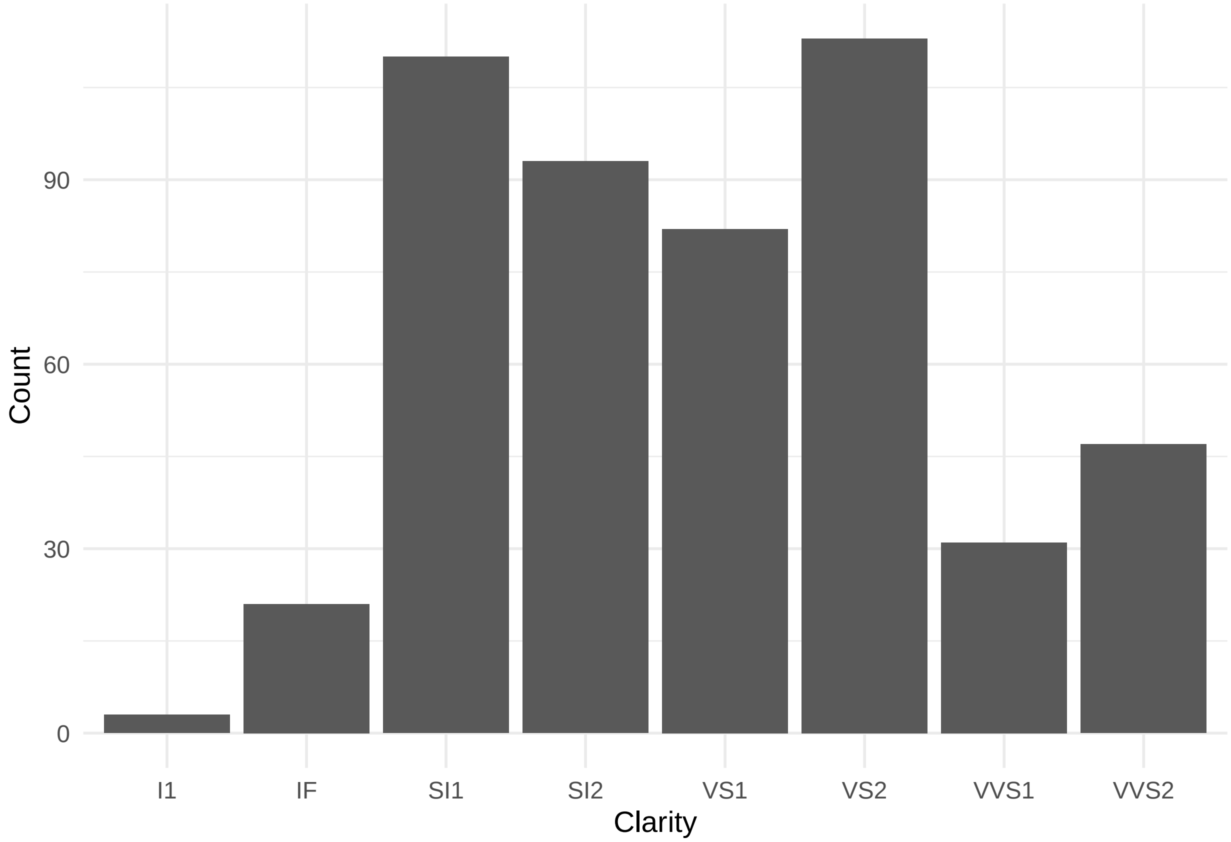
```
ggplot(diamondSample, aes(x=color)) +
  geom_bar() +
  xlab("Color") +
  ylab("Count") +
  theme_minimal()
```



```
clarityTable <- table(diamondSample$clarity)
clarityPTable <- prop.table(table(diamondSample$clarity))
data.frame(
  Clarity = names(clarityTable),
  Count = as.vector(clarityTable),
  Percentage = round(as.vector(clarityPTable), 2)
)
```

```
##   Clarity Count Percentage
## 1      I1     3       0.01
## 2      IF    21       0.04
## 3     SI1   110       0.22
## 4     SI2    93       0.19
## 5     VS1    82       0.16
## 6     VS2   113       0.23
## 7    VVS1    31       0.06
## 8    VVS2    47       0.09
```

```
ggplot(diamondSample, aes(x = clarity)) +
  geom_bar() +
  xlab("Clarity") +
  ylab("Count") +
  theme_minimal()
```



From these diagrams, we can conclude that neither of these fields seem to be dominated by one value in our sample.

Now, for the quantitative analyses, we can import the `skimr` library and make use of the `skim` function that returns a much more visually pleasing and in depth summary of the data. Running this function with the sample data as the parameter, we get the following table.

```r
skim(diamondSample)
```

Table 1: Data summary

Name	diamondSample
Number of rows	500
Number of columns	6
Column type frequency:	
character	2
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
color	0	1	1	1	0	7	0
clarity	0	1	2	4	0	8	0

Variable type: numeric

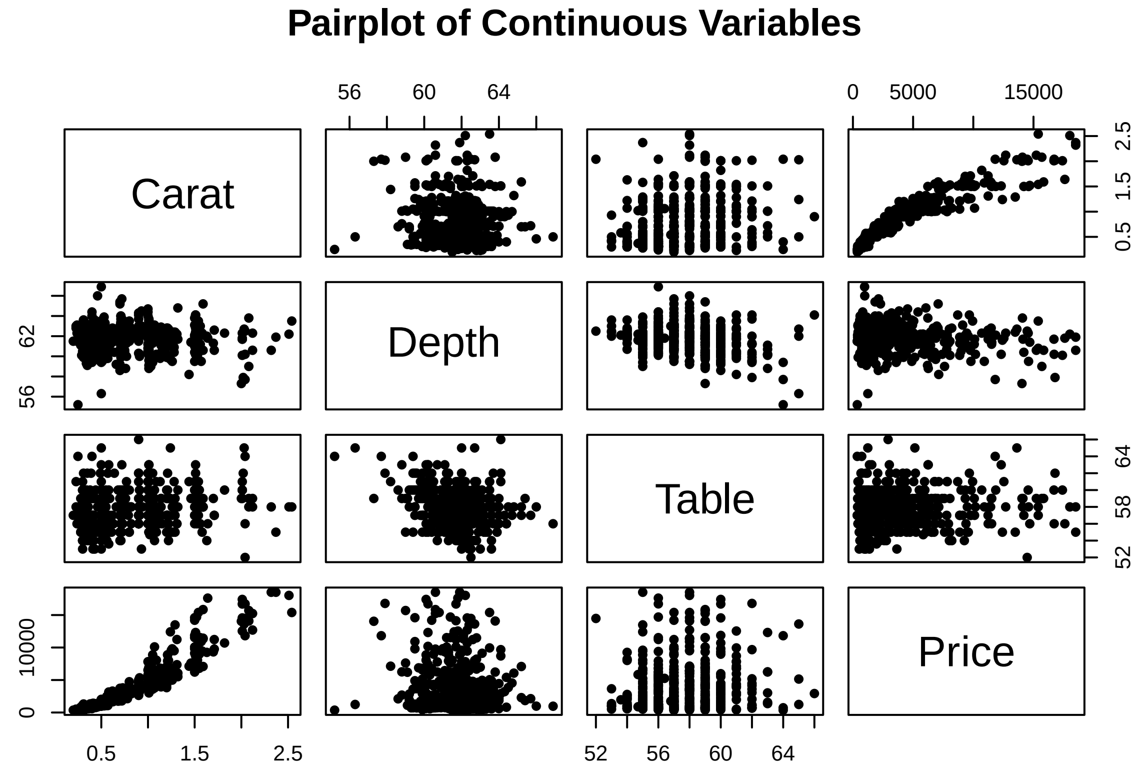
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
carat	0	1	0.78	0.47	0.2	0.39	0.7	1.03	2.54	
depth	0	1	61.74	1.30	55.2	61.10	61.8	62.40	66.90	
table	0	1	57.52	2.23	52.0	56.00	57.0	59.00	66.00	
price	0	1	3774.25	3858.12	357.0	942.75	2273.0	5189.25	18508.00	

From this table we can see, not only the five number summary of each field, but also additional information such as the standard deviation (`sd`), the number of incomplete data points (`n_missing`), and a simple histogram of the distribution (`hist`). Notice that here the five number summary, which was given as the minimum, 1st quartile, median, 3rd quartile, and maximum by the `summary` function, are here given as the percentile values, `p0`, `p25`, `p50`, `p75`, and `p100` respectively.

Note that the `skim` function also returns a summary of the qualitative fields. While these summaries are not as detailed as the ones returned for the quantitative fields, we can still easily see the number of incomplete data points as well as the number of unique entries for each field.

In addition to these tables, we can also create what is called a pairplot of all the continuous variables in the sample. This will return a table of scatterplots between all the continuous variables with each other continuous variable.

```r
pairs(
  diamondSample[c("carat", "depth", "table", "price")],
  pch = 16,
  label = c("Carat", "Depth", "Table", "Price"),
  main = "Pairplot of Continuous Variables"
)
```



These plots allow us to examine different relationships between each variable. For instance, we can see a clear correlation between the carat measure of each diamond with its price. This idea can be further satisfied by find the actual correlation value as follows:

```r
cor(diamondSample$carat, diamondSample$price)
```

```
## [1] 0.9421419
```

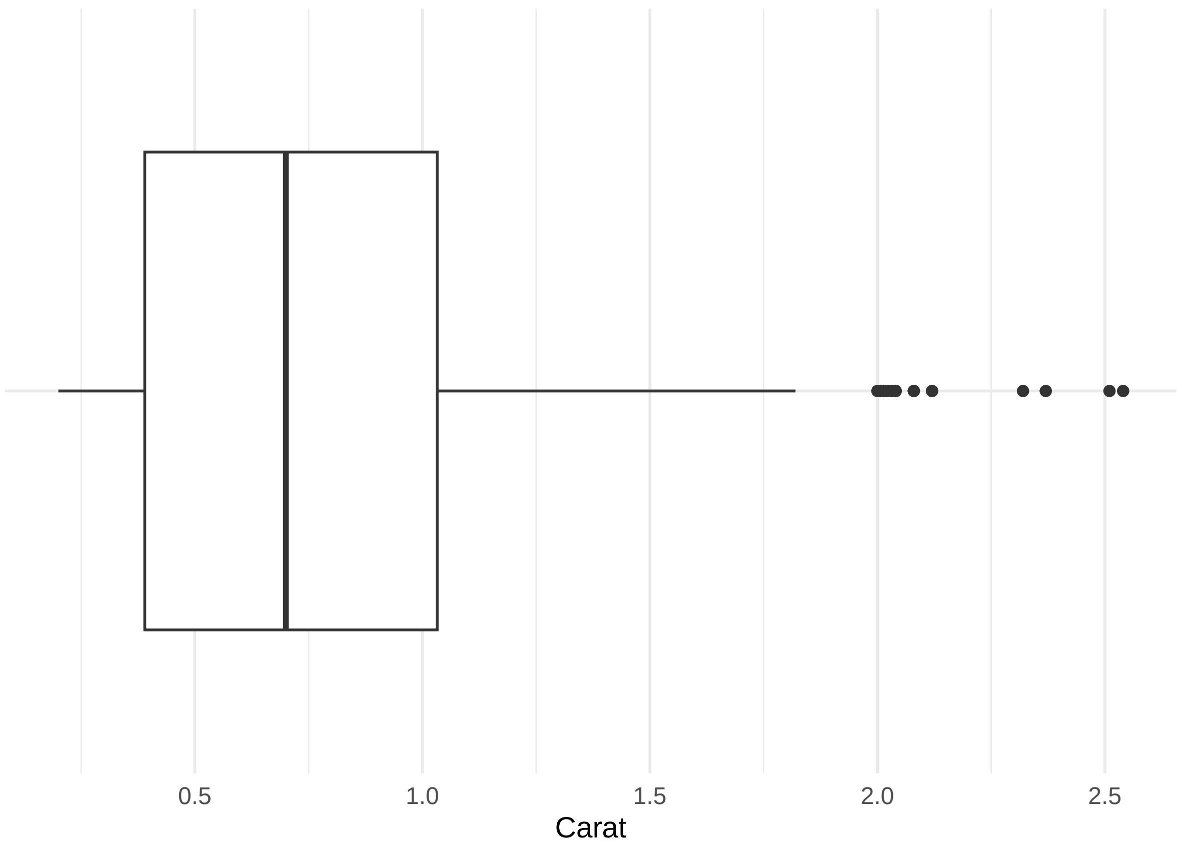
Since the correlation value is close to one, we can say that these fields are closely related.

Interesting Points

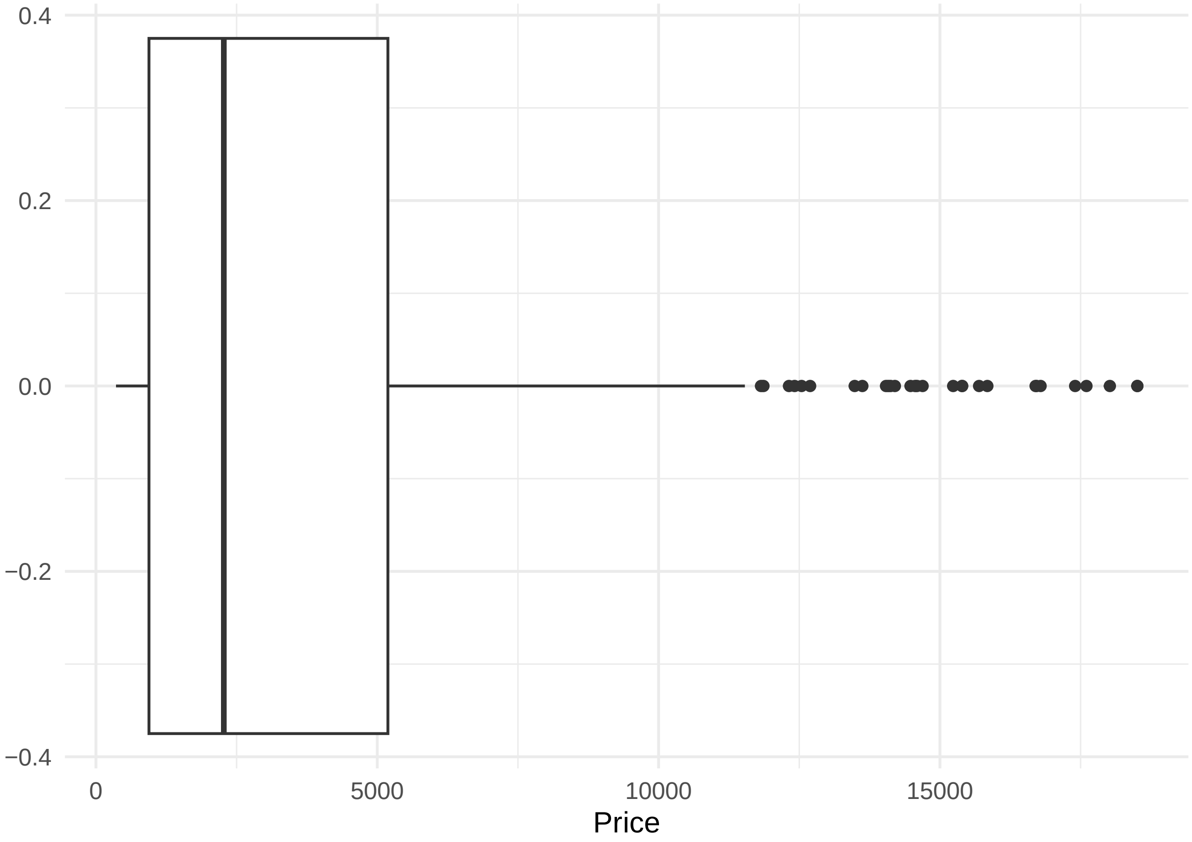
(1) Briefly, observe that while the `table` field is technically continuous, as the entries can take any numerical value, most of them seem to be positive integer values. We will, however, continue to treat them as continuous.

(2) The next note about the taken sample is that the `carat` and `price` fields appear to be weighted heavily to the left. In order to get a better understanding of the distributions of these variables, we can create individual boxplots of the fields to run further analysis.

```r
ggplot(diamondSample, aes(x = carat, y = "")) +
  geom_boxplot() +
  xlab("Carat") +
  ylab("") +
  theme_minimal()
```



```
ggplot(diamondSample, aes(x = price)) +
  geom_boxplot() +
  xlab("Price") +
  ylab("") +
  theme_minimal()
```



We can see here that both variables contain several outlier points. Although there are few of these points compared to the total number of points, they could have a disproportionate effect on the data. Specifically, there may be a small number of high caret diamonds with high prices within the chosen sample.

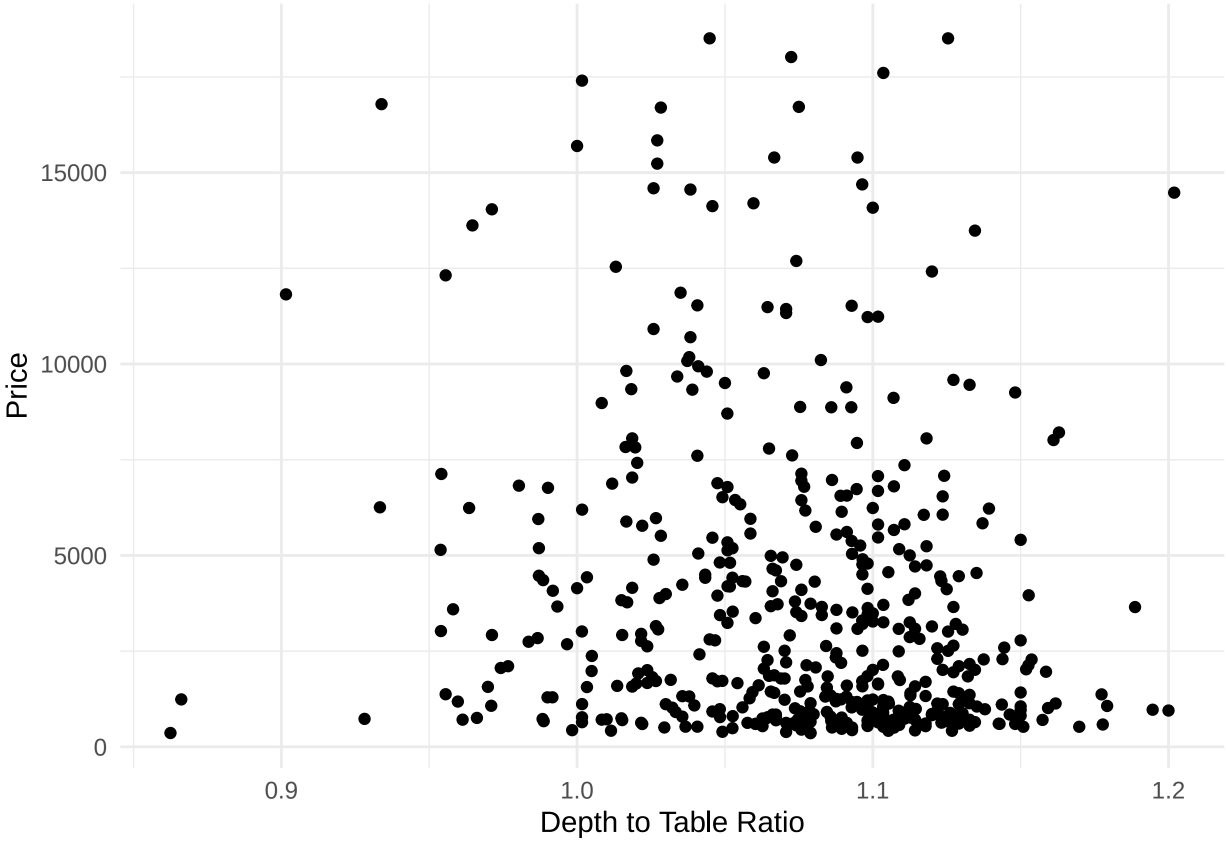
(3) Recall that the pairplot we made above revealed a correlation between the carat and price fields. Aside from this, the depth and table fields do not seem to be correlated with the price, as would have seemed natural. However, these fields do seem to share similar percentile values. We could potentially try to find a correlation between the ratio of depth to table and the price of each diamond to see if a certain size proportion is preferred to others, rather than analyzing then separately.

Let's briefly explore this idea.

```
cor(diamondSample$depth/diamondSample$table, diamondSample$price)
```

```
## [1] -0.1841065
```

```
ggplot(diamondSample, aes(x = depth/table, y = price)) +
  geom_point() +
  xlab("Depth to Table Ratio") +
  ylab("Price") +
  theme_minimal()
```



After a simple analysis, there does not appear to be a strong correlation between depth and table ratio with the price either, as the correlation value was close to zero and the plot does not seem to follow any sort of clear path. However, after further analysis, there could be other underlying relationships between these variables. For example, if there is such a low correlation between the price of the diamonds and these variables, it may be good to perform further analyses between price and other variables in the future.

(4) One final thing to note is that while this dataset does include a large number of data points with a vast range of characteristics, there are still more properties that these diamonds could have. For instance, we only have data on the top two color groups and, as well, we lack data from some of the lower tiered clarity classifications. While this does not make any of the data we are using less useful, it does mean that other analyses on other datasets could draw conclusions that may not be possible with our current data.