**Chapter 1**

To process and study our problem, we will use the statistical package R, through which we will be able to collect the required information and present the requested statistical results. For each action we perform in R (typing commands), there will be a relevant reference and some comments explaining why it was used and clarifying its function.

Starting off, we want to pass to R the data stored in the given file (in excel format: Diamond.csv). For this reason, we will use the read.table(.) command as follows:

Where we link the file to R.

> file.choose()

[1] ''C:\\Users\\MY\_PATH\\Downloads\ \Diamond.csv11

Where we link the file to R.

>a <- read.table("C:/Users/your\_username/Documents/Diamond.csv", header = TRUE, sep = ",", na.strings = "\*", colClasses = c(rep("character", 6)))

Where the read.table(.) command reads a file in table format and creates a data frame from it. The header=TRUE parameter is a logical value that indicates whether the file contains the variable names on its first line. The command sep =''," indicates how characters are separated in the file. na.strings="\*" converts the ignored values denoted by "\*" to "Na". The type of object created is a data frame.

We also execute the following commands, which ensure that the specific values are numeric:

> class(a$id) = “numeric”

> class(a$carat) = “numeric”

> class(a$price)= “numeric”

**Chapter 2 – General Sample Characteristics**

We now have 5 variables to study. In detail: l) the weight unit in carats (carat), the color (colour1 with categories “D”, “E”, “F”, “G”, “H”, “I”, “J”), the clarity (clarlty, with categories “SI1”, “VS2”, “VI2”, “VS1”, “VVS1, “VVS2”, “Other”), the certification (Cut, with categories “Fair”, “Good”, “Very Good”, “Premium”, “Ideal”) and the price (price) in dollars A quick yet informative summary can be obtained using the command summary(a, na.rm = TRUE), which displays statistics such as the counts for each category, the number of missing (NA) values, and for numerical variables, the mean, range, minimum, and maximum. The results of its execution are shown below:

> summary(a, na.rm=TRUE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Carat** | **Colour** | **Clarlty** | **Cut** | **Prlce** |
| Min.: 0.2000 | D: 6775 | SI1: 13065 | Fair: 1610 | Min. : 326 |
| 1st Qu.: 0.4000 | E: 9797 | VS2: 12258 | Good: 4906 | 1st Qu. : 950 |
| Median: 0.7000 | F: 9542 | VI2 : 9194 | Very Good: 12082 | Median : 4201 |
| Mean: 0.7979 | G: 11292 | VS1: 8171 | Premium: 13791 | Mean : 3933 |
| 3rd Qu.: 1.0400 | H: 8304 | VVS2: 5066 | Ideal: 21551 | 3rd Qu.: 5324 |
| Max.: 5.0100 | I: 5422 | VVS1: 3655 |  | **Max.** : 18823 |
|  | J: 2808 | (Other): 2531 |  |  |

We note that the quantitative variable 1st Qu. is called the interquartile range and indicates the observation that is greater than or equal to exactly 25% of the observations and, respectively, the 3rd Qu. the observation that is greater than or equal to exactly 75% of the observations.

At the same time, the large size of the sample we have and the relatively small number of NA's data do not make it necessary to replace them with our own estimate.

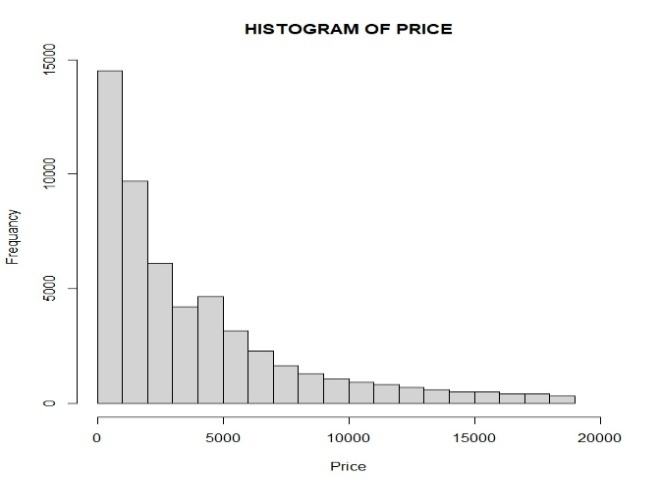
Continuing, it seems useful and even necessary to visualize our variables as it will give us a better sense of them. For this reason, we will use the graphical capabilities offered by R. Starting our graphical study with the numerical variables, we will create two histograms using the hist(x) command. More specifically, we construct the histograms of the variables "Carat" and "Price" as follows:

> hist(a$carat, xlim = c(0,2.5), xlab = "Carat", ylab = "Frequency", main = "HISTOGRAM   OF CARAT", col="darkolivegreen1")

> hist(a$price, xlab = "Price", ylab = "Frequancy", main = "HISTOGRAM OF PRICE" )

And we display them below with the left being the Carat histogram and the right being the Price histogram.

A graph of a number of green bars

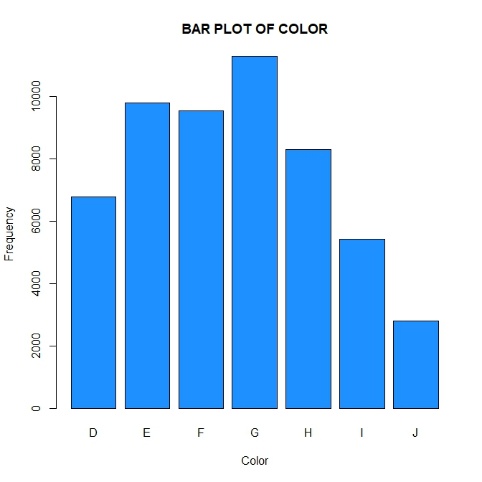
AI-generated content may be incorrect.

We now continue and move on to the categorical variables which are color, clarity and certification. The choice of representation is the bar graph (and not the tomogram) in this specific case as it gives us better information and this is implemented with a barplot(table(A)). In our case we write:

> barplot(table(a$color),xlab = "Color", ylab = "Frequency", main = "BAR PLOT OF COLOR", col="dodgerblue")

> barplot(table(a$clarity),xlab = "Clarity", ylab = "Frequency", main = "BAR PLOT OF CLARITY", col="dodgerblue")

> barplot(table(a$cut),xlab = "Cut", ylab = "Frequency", main = "BAR PLOT OF CUT", col="dodgerblue")

A bar chart of a number of bars

AI-generated content may be incorrect.A bar graph with text

AI-generated content may be incorrect.

**Chapter 3 -**

Once we have examined the general characteristics of the sample, we will proceed to draw more specific conclusions.

We want to check the relationship and dependence between the variable "Price" with both “Color” and Certification. However, with the form of the data we have, we cannot draw any direct conclusion. The graphical representation will help us to satisfactorily achieve the control we desire. Thus, using a thecogram (a suitable choice for comparison), we will try to obtain the desired information. The commands we use to create this are:

> ggplot(diamonds, aes(x = cut, y = price/carat)) + geom\_boxplot(fill = "skyblue") +

  labs(title = "Price per Carat by Cut", y = "Price per Carat")

> ggplot(diamonds, aes(x = clarity, y = price)) + geom\_boxplot(fill = "skyblue") +

  labs(title = "Price per Carat by Clarity", y = "Price per Carat")

and we get:

A graph of a chart

AI-generated content may be incorrect.A graph of blue and black bars

AI-generated content may be incorrect.

One can see that on our dataset there is a great number of **outliers**, which one has to identify and proceed effectively in dealing with them. It is not always straightforward to proceed. An easy way would be to get rid of them by simply ignoring the values that have unexpectedly high values. By its nature, this approach is the simplest and quickest one, but the unjustified way of deleting them could cause statistical issues. It’s essential to understand how outliers occur and whether they might happen again.

Outliers can also occur when we present the price and the carat. Going back to the first table, one can see that the **mean value** of the price is 4201 and the **maximum value** is 18,823. On the other hand, the **mean value** of carats is 0.7979, but the **maximum value** reaches the number 5.01, which is far from the mean value.

We will use a modified version of z-scope to find the outliers, taking as a reference point the price values. By running the following command:

> price <- diamonds$price

> med <- median(price)

> mad\_val <- mad(price)

# Modified z-score formula

> mod\_z <- 0.6745 \* (price - med) / mad\_val

# Threshold: usually > 3.5 is considered an outlier

> outliers <- which(abs(mod\_z) > 3.5)

> diamonds[outliers, ]

> length(outliers)

One gets the price outlier, and the number of them is 1522. By running a similar command for the values of carat, we get only 13 outliers. We decided to get rid of the price outliers by utilizing the following lines:

> diamonds\_clean <- diamonds[abs(mod\_z) <= 3.5,]