Exploratory Data Analysis in R Programing

Thamma Tharasombat

2022-10-30

Today, we are going to analyse 'diamonds' data with ggplot!

Install library

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(patchwork)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.5
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.1     v forcats 0.5.2
## v readr   2.1.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(RColorBrewer)
```

Data Information

- Carat = weight of the diamond (0.2-5.01)
- Cut = quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- Color = diamond color (D, E, F = colorless to G, H, I, J = near colorless)
- clarity = measurement of how clear the diamond is (I1 = worst graded, SI2, SI1, VS2, VS1, VVS2, VVS1, IF = best graded)
- Depth = total depth percentage (43-79)
- Table = width of top of diamond relative to widest point (43-95)
- Price = price in US dollars ($326-$18,823)

- x = length in mm (0-10.74)
- y = width in mm (0-58.9)
- z = depth in mm (0-31.8)

Data Overview

```
glimpse(diamonds)
```

```
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

The Diamonds data set contains information about 53,940 round-cut diamonds with 10 variables

Check null

```
diamonds %>% is.na() %>% sum()
```

```
## [1] 0
```

There is no null data in diamonds.

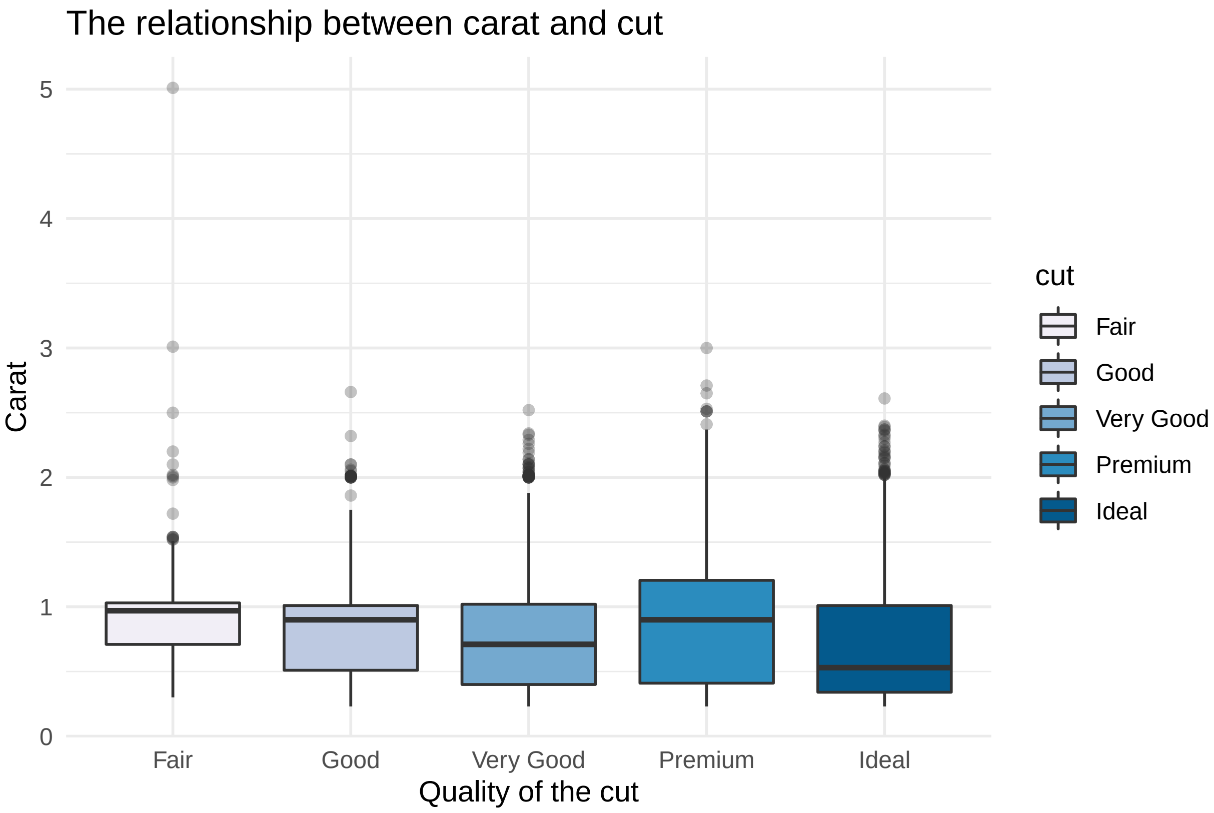
Data cleaing

```
set.seed(1)
df <- diamonds %>% sample_n(size = 5394)
```

The data is huge. We will only take 10% (5,394) of it as a sample.

Graph 1. The relationship between carat and cut.

```
ggplot(df,aes(cut,carat, fill = cut))+
    geom_boxplot(outlier.alpha = 0.3)+
        theme_minimal()+
    labs(title = "The relationship between carat and cut",
         x = "Quality of the cut",
         y = "Carat",
         caption = "Source: Diamonds from ggplots2 package")+
    scale_fill_brewer(type = "seq", palette = "PuBu")
```



Source: Diamonds from ggplots2 package

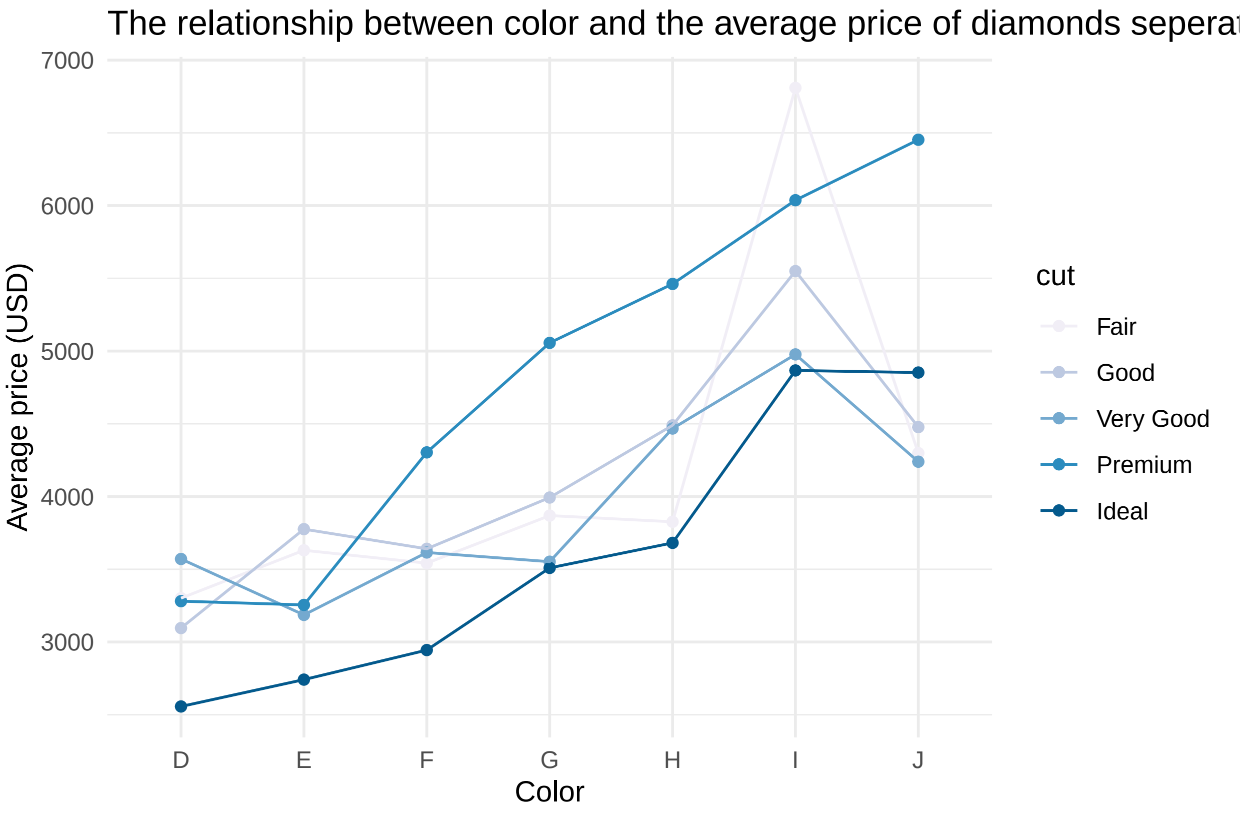
This box plot shows the relationship between carats and cut.

The fair-graded diamonds have the largest interquartile range and they have a median of around 1, which is the highest. On the other hand, the ideal graded diamonds have the lowest median, which is a little bit higher than 0.5.

Graph 2. The relationship between color and the average price of diamonds seperated, by cut.

```
df %>% group_by(color, cut) %>%
    summarise(avg_p = mean(price)) %>%
    ggplot(aes(color,avg_p, group = cut, color = cut))+
    geom_point()+
    geom_line()+
    theme_minimal()+
     labs(title = "The relationship between color and the average price of diamonds seperated, by cut."
         x = "Color",
         y = "Average price (USD)",
         caption = "Source: Diamonds from ggplots2 package")+
    scale_color_brewer(type = "seq", palette = "PuBu")
```

```
## `summarise()` has grouped output by 'color'. You can override using the
## `.groups` argument.
```



Source: Diamonds from ggplots2 package

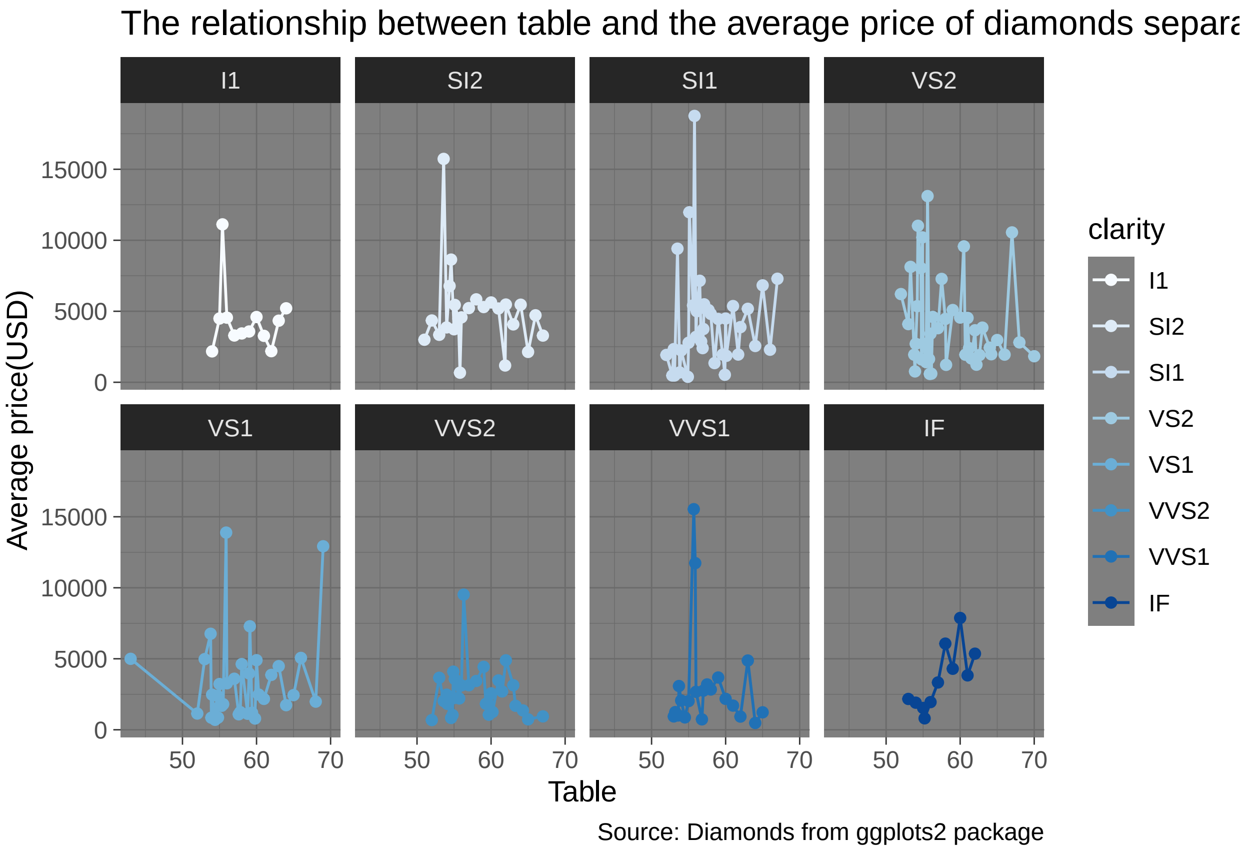
This line graph shows the relationship between color and the average price of diamonds, separated by cut.

It is shocking that the ideal graded diamonds have the lowest average price of all diamond colors except J. Plus, the fact that fair-graded diamonds, which are the lowest graded diamonds, have the highest price where the color of diamonds is I.

Graph 3. The relationship between table and the average price of diamonds separated, by clarity.

```
df %>% group_by(table, clarity) %>%
    summarise(avg_p = mean(price)) %>%
    ggplot(aes(table, avg_p, group = clarity, color = clarity)) +
    geom_point()+
    geom_line()+
    theme_dark()+
    facet_wrap(~ clarity, ncol = 4)+
    labs(title = "The relationship between table and the average price of diamonds separated, by clarit
         x = "Table",
         y = "Average price(USD)",
         caption = "Source: Diamonds from ggplots2 package")+
    scale_color_brewer(type = "seq", palette = "Blues")
```

```
## `summarise()` has grouped output by 'table'. You can override using the
## `.groups` argument.
```


The relationship between table and the average price of diamonds separa

Source: Diamonds from ggplots2 package

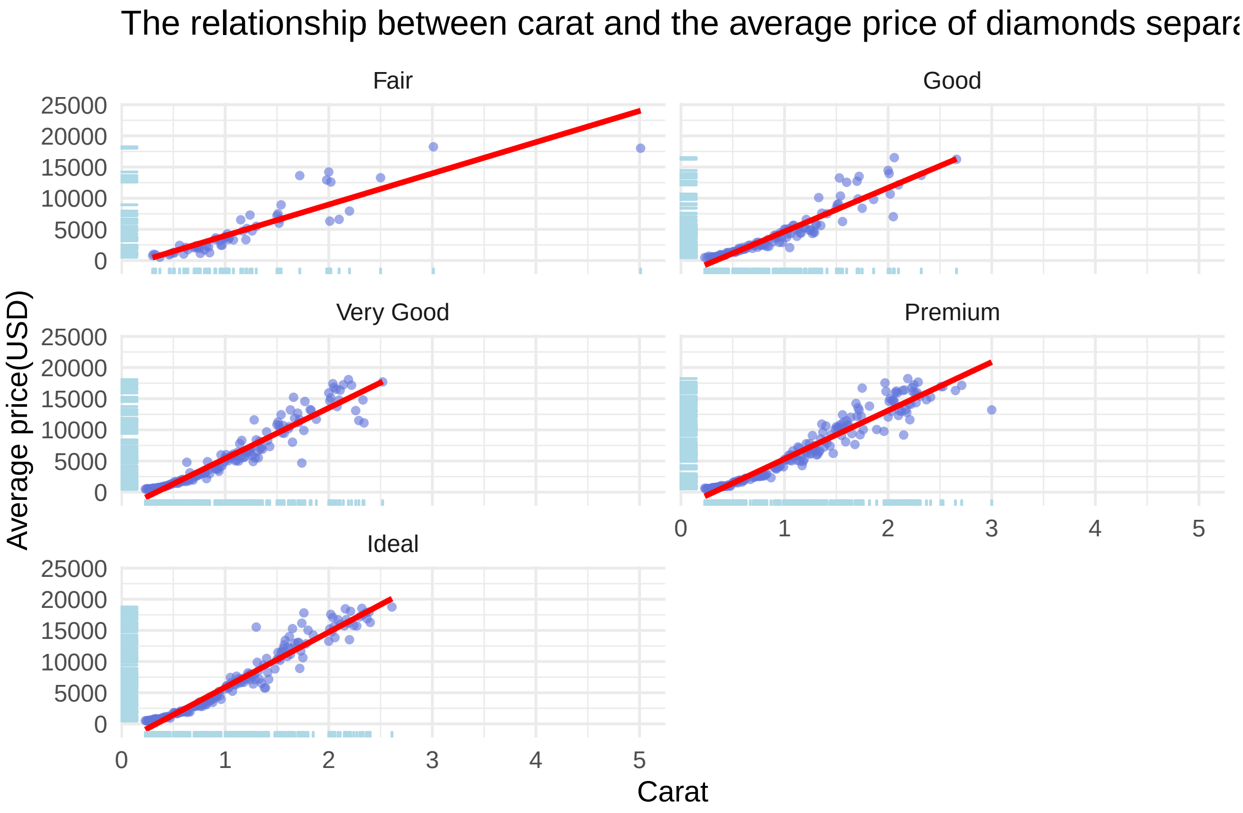
This line graphs show the relationship between table and the average price of diamonds separated, by clarity.

If we look at the trend, we can say that table size does not have a relationship with price because the wider the table does not mean the higher the price. The graph shows that diamonds with a clarity grade of I1 are the only type of diamond that starts with the highest price, which is estimated to be around 2,500 USD.

Graph 4. The relationship between carat and the average price of diamonds separated, by cut.

```
df %>% group_by(carat,cut) %>%
    summarise(avg_p = mean(price)) %>%
    ggplot(aes(carat, avg_p, group = cut)) +
    geom_point(size = 1, col = "#6074DA", alpha = 0.6) +
    facet_wrap(~ cut, ncol =2) +
    geom_smooth(col = "red", method = "lm",
                se = FALSE)+
    geom_rug(col = "lightblue")+
    theme_minimal() +
    labs(title = "The relationship between carat and the average price of diamonds separated, by cut",
         x = "Carat",
         y = "Average price(USD)",
         caption = "Source: Diamonds from ggplots2 package")
```

```
## `summarise()` has grouped output by 'carat'. You can override using the
## `.groups` argument.
## `geom_smooth()` using formula 'y ~ x'
```



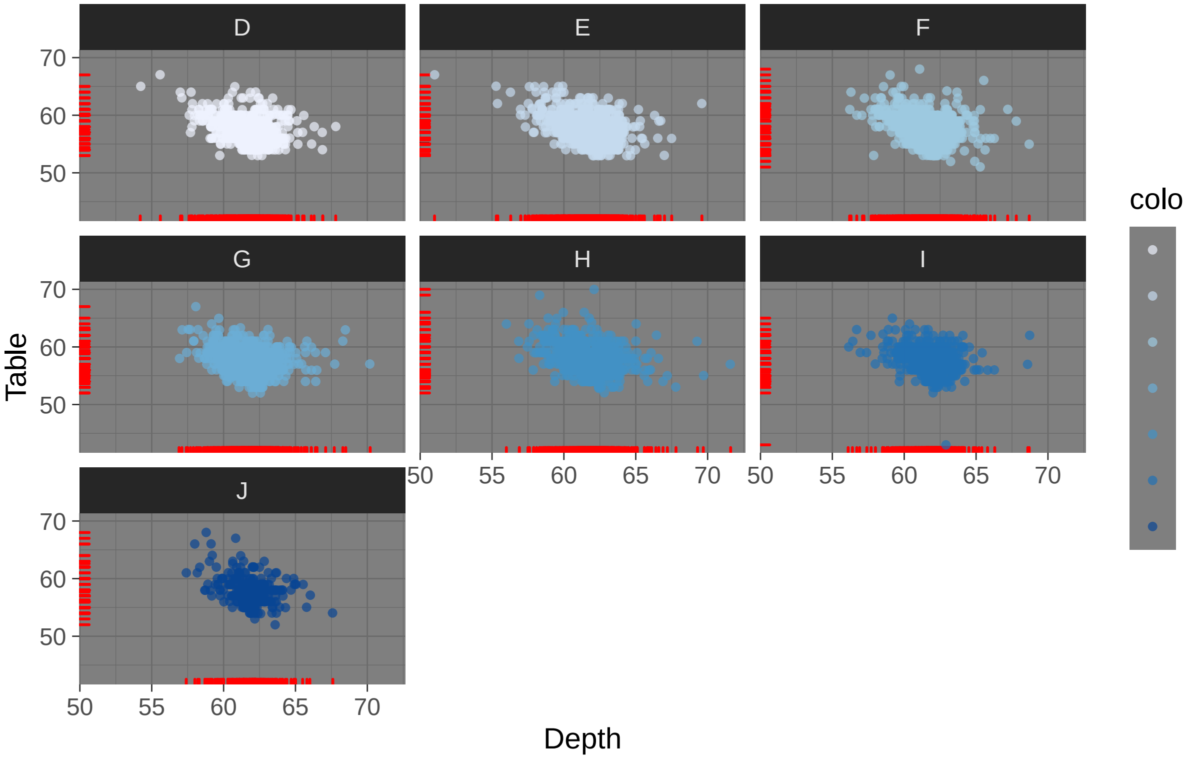
Source: Diamonds from ggplots2 package

This bar graphs show the relationship between carat and the average price of diamonds separated, by cut. If we look at the trend, we can say that the higher the price, the higher the number of carats.

Graph 5. The relationship between depth and table separated, by color.

```
ggplot(df,aes(depth,table, color = color)) +
    geom_rug(col = "red") +
    geom_jitter(size = 1, alpha = 0.7) +
    facet_wrap(~ color, ncol =3)+
    scale_color_brewer(type = "seq", palette = "Blues")+
    theme_dark() +
    labs(title = "The relationship between depth and table separated, by color.",
         x = "Depth",
         y = "Table",
         caption = "Source: Diamonds from ggplots2 package")
```

The relationship between depth and table separated, by color.



Source: Diamonds from ggplots2 package

These dot bar graphs show the relationship between depth and table separated by color.

If we look at the graphs, we can tell that all the diamonds have a table between 50 and 65 and a depth of 60 to 65 with a few out liners.