EDA_on_diamonds_dataset

Ayobami Alaran

2022-06-26

Exploratory Data Analysis on Diamond dataset

Importing the libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
```

Assessing the dataset

head functions shows the top 6 rows of the set

```
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 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  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
str(diamonds)
```

```
## tibble[,10] [53,940 x 10] (S3: tbl_df/tbl/data.frame)
##  $ carat  : num [1:53940] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
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.~
```

Lets check the column names

```
colnames(diamonds)
```

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

Cleaning the data

Renaming the color column name to colour

```
rename(diamonds, colour=color)
```

```
## # A tibble: 53,940 x 10
##    carat cut       colour clarity depth table price     x     y     z
##    <dbl> <ord>     <ord>  <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  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  4.2   4.23  2.63
##  5  0.31 Good      J      SI2      63.3    58   335  4.34  4.35  2.75
##  6  0.24 Very Good J      VVS2     62.8    57   336  3.94  3.96  2.48
##  7  0.24 Very Good I      VVS1     62.3    57   336  3.95  3.98  2.47
##  8  0.26 Very Good H      SI1      61.9    55   337  4.07  4.11  2.53
##  9  0.22 Fair      E      VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H      VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

Exploratory data analysis

What is the average carat of the diamond

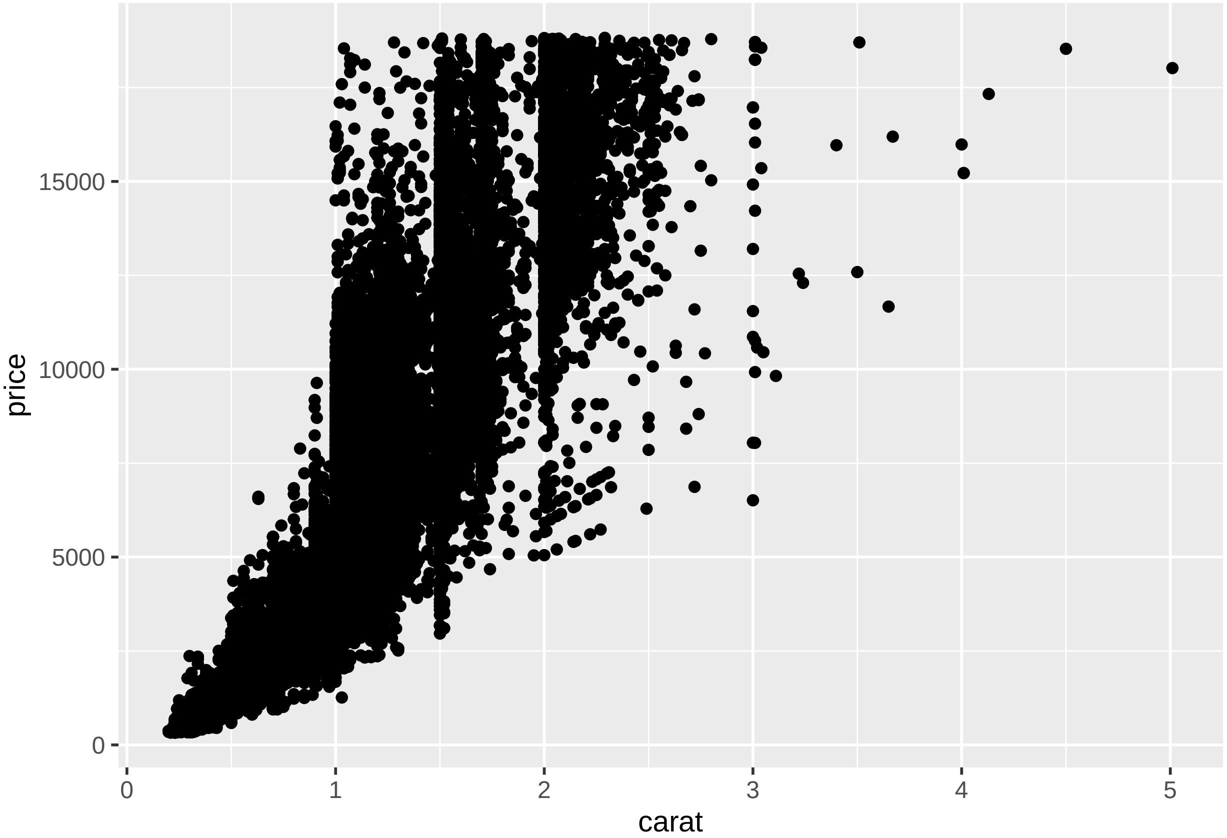
```
summarise(diamonds, avg_carat=mean(carat))
```

```
## # A tibble: 1 x 1
##   avg_carat
##       <dbl>
## 1     0.798
```

Data Visualization

showing the relationship between the diamond carat and price

```
ggplot(data = diamonds, ) +
  geom_point(mapping=aes(x = carat, y = price))
```



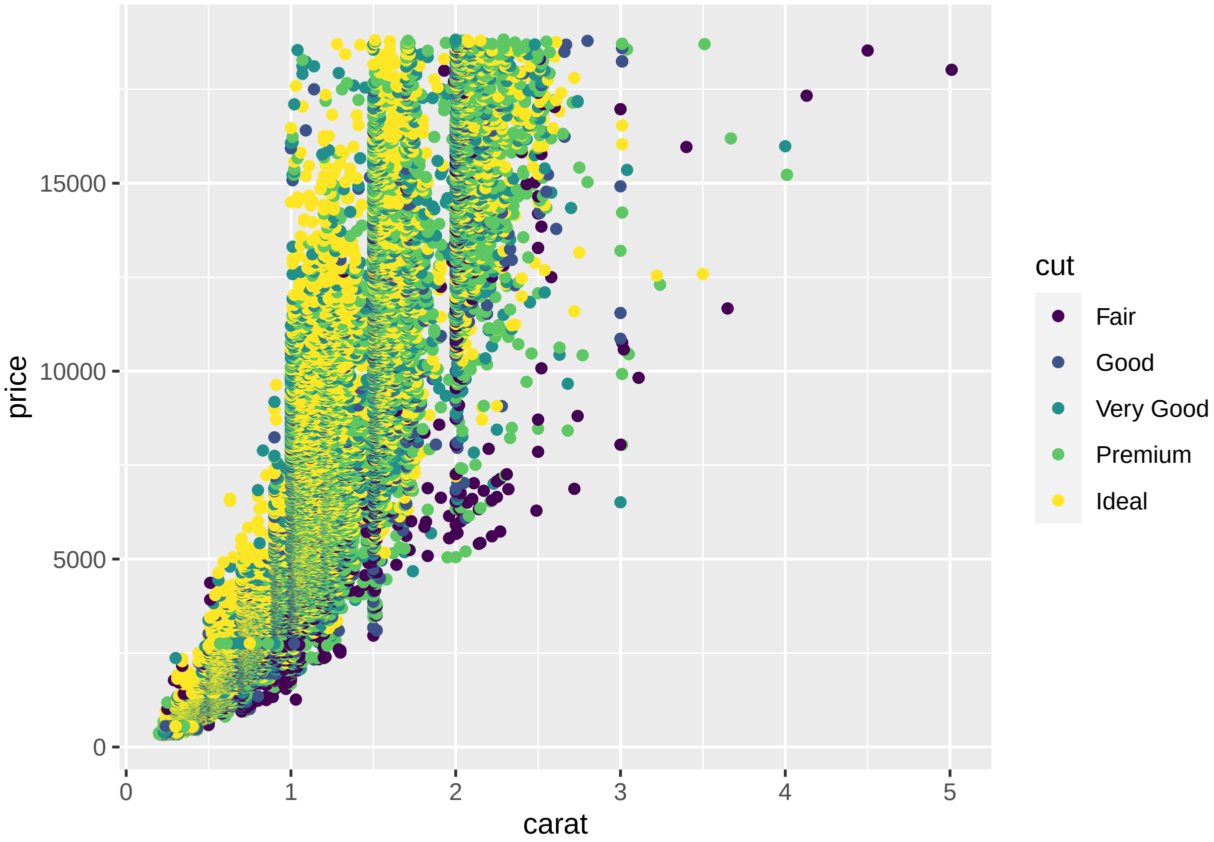
```
cor(select(diamonds, carat, price))
```

```
##           carat     price
## carat 1.0000000 0.9215913
## price 0.9215913 1.0000000
```

Interpretation

from the data viz result, we can see that there is an upward movement in the trend that is there is a relationship between the the two variables, the bigger the carat the higher the price. for further investigation, the correlation shows that there is 0.92 correlation between the carat and price.

```
ggplot(data = diamonds, aes(x = carat, y = price, color = cut)) +
  geom_point()
```



price
15000
10000
5000
0
0
1
2
3
4
5
carat
cut
● Fair
● Good
● Very Good
● Premium
● Ideal