EXPLORATARY DATA ANALYSIS

```r
#installing package
install.packages("ggplot2")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)

library(ggplot2)
install.packages("dplyr")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)

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

install.packages("reshape2")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)

library(reshape2)
#importing dataset
data("diamonds")
#data manipulation
select(diamonds,color)

## # A tibble: 53,940 × 1
##    color
##    <ord>
##  1 E
##  2 E
##  3 E
##  4 I
##  5 J
##  6 J
##  7 I
##  8 H
##  9 E
```

```
## 10 H
## # ℹ 53,930 more rows
```

```r
filter(diamonds,price==max(price))
```

```
## # A tibble: 1 × 10
##   carat cut     color clarity depth table price     x     y     z
##   <dbl> <ord>   <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  2.29 Premium I     VS2      60.8    60 18823   8.5  8.47  5.16
```

```r
diamonds_filtered <- diamonds %>% select(color)
diamonds_filtered
```

```
## # A tibble: 53,940 × 1
##    color
##    <ord>
##  1 E
##  2 E
##  3 E
##  4 I
##  5 J
##  6 J
##  7 I
##  8 H
##  9 E
## 10 H
## # ℹ 53,930 more rows
```

```r
diamonds %>%
filter(color=="D")%>%
select(clarity,price)
```

```
## # A tibble: 6,775 × 2
##    clarity price
##    <ord>   <int>
##  1 VS2       357
##  2 VS1       402
##  3 VS2       403
##  4 VS2       403
##  5 VS1       403
##  6 VS2       404
##  7 SI1       552
##  8 SI1       552
##  9 SI1       552
## 10 VVS1      553
## # ℹ 6,765 more rows
```

```r
diamonds%>%arrange(price)
```

```
## # A tibble: 53,940 × 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
## 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
## # ℹ 53,930 more rows
```

```
diamonds%>%
mutate(price_percentage=price*0.04)
```

```
## # A tibble: 53,940 × 11
##    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
##  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
## # ℹ 53,930 more rows
## # ℹ 1 more variable: price_percentage <dbl>
```

```
diamonds%>%
group_by(clarity)%>%
mutate(price_per_carat=price/carat)
```

```
## # A tibble: 53,940 × 11
## # Groups:   clarity [8]
##    carat cut   color clarity depth table price     x     y     z pri
ce_per_carat
##    <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
      <dbl>
##  1  0.23 Ideal E     SI2      61.5    55   326  3.95  3.98  2.43
      1417.
##  2  0.21 Prem… E     SI1      59.8    61   326  3.89  3.84  2.31
      1552.
##  3  0.23 Good  E     VS1      56.9    65   327  4.05  4.07  2.31
      1422.
##  4  0.29 Prem… I     VS2      62.4    58   334  4.2   4.23  2.63
      1152.
##  5  0.31 Good  J     SI2      63.3    58   335  4.34  4.35  2.75
      1081.
```

```
## 6  0.24 Very… J     VVS2     62.8    57   336  3.94  3.96  2.48
       1400
## 7  0.24 Very… I     VVS1     62.3    57   336  3.95  3.98  2.47
       1400
## 8  0.26 Very… H     SI1      61.9    55   337  4.07  4.11  2.53
       1296.
## 9  0.22 Fair  E     VS2      65.1    61   337  3.87  3.78  2.49
       1532.
## 10 0.23 Very… H     VS1      59.4    61   338  4     4.05  2.39
       1470.
## # ℹ 53,930 more rows
```

```
diamonds %>%
group_by(price) %>%
summarize(n())
```

```
## # A tibble: 11,602 × 2
##    price `n()`
##    <int> <int>
##  1   326     2
##  2   327     1
##  3   334     1
##  4   335     1
##  5   336     2
##  6   337     2
##  7   338     1
##  8   339     1
##  9   340     1
## 10   342     1
## # ℹ 11,592 more rows
```

```
diamonds %>%
  summarize(mean_price=mean(price),
            median_price=median(price),
            min_price=min(price),
            max_price=max(price),
            sd_price=sd(price))
```

```
## # A tibble: 1 × 5
##   mean_price median_price min_price max_price sd_price
##        <dbl>        <dbl>     <int>     <int>    <dbl>
## 1      3933.         2401       326     18823    3989.
```

```
#EDA
#structure of the dataset
str(diamonds)
```

```
## tibble [53,940 × 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 3.8
7 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 ...
```

```r
#Summary statistics for numerical variables
summary(diamonds$depth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   43.00   61.00   61.80   61.75   62.50   79.00
```

```r
summary(diamonds$carat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.2000  0.4000  0.7000  0.7979  1.0400  5.0100
```

```r
summary(diamonds$table)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   43.00   56.00   57.00   57.46   59.00   95.00
```

```r
summary(diamonds$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     326     950    2401    3933    5324   18823
```

```r
summary(diamonds$x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   4.710   5.700   5.731   6.540  10.740
```

```r
summary(diamonds$y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   4.720   5.710   5.735   6.540  58.900
```

```r
summary(diamonds$z)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   2.910   3.530   3.539   4.040  31.800
```

```r
#checking missing values
sum(is.na(diamonds))
```

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

```r
#subsetting data
subset_data=data.frame(diamonds$carat,diamonds$table,diamonds$depth,dia
monds$price)
head(subset_data)
```

```
##   diamonds.carat diamonds.table diamonds.depth diamonds.price
## 1           0.23             55           61.5            326
## 2           0.21             61           59.8            326
## 3           0.23             65           56.9            327
## 4           0.29             58           62.4            334
## 5           0.31             58           63.3            335
## 6           0.24             57           62.8            336
```

```r
#univariet analysis
ggplot(diamonds,aes(x=price))+geom_histogram(fill="skyblue",color="blac
k")+
  labs(title="Distribution of Prices of diamonds",x="Price",y="Frequenc
y")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

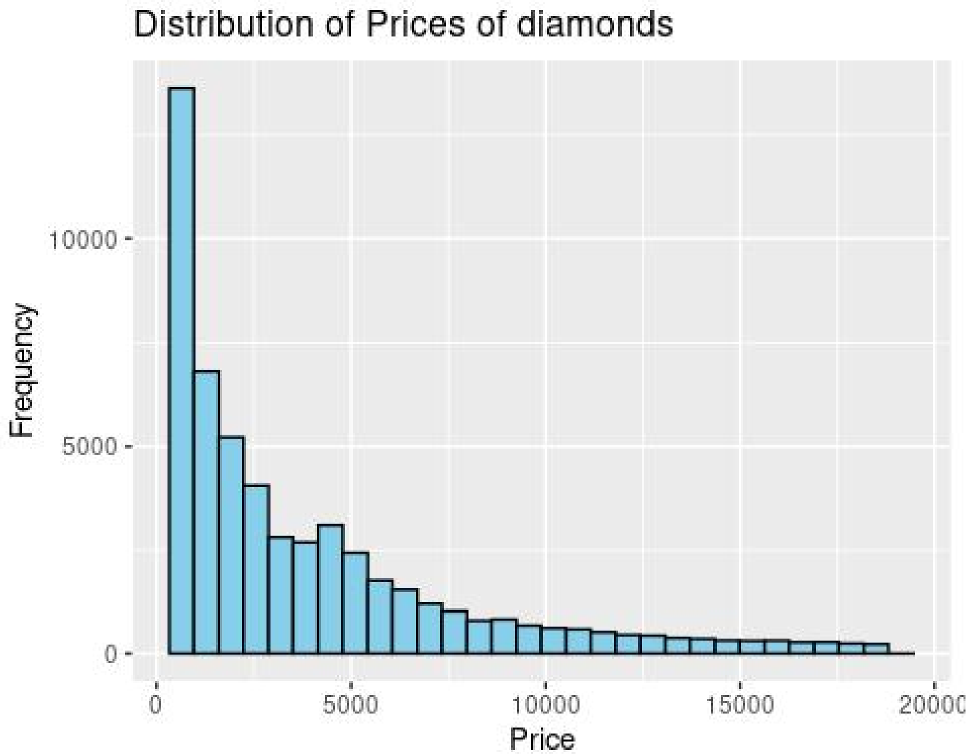

Fig 1.1

```
ggplot(diamonds,aes(x=depth))+geom_histogram(fill="skyblue",color="black")+
  labs(title="Distribution of depth of diamonds",x="Depth",y="Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

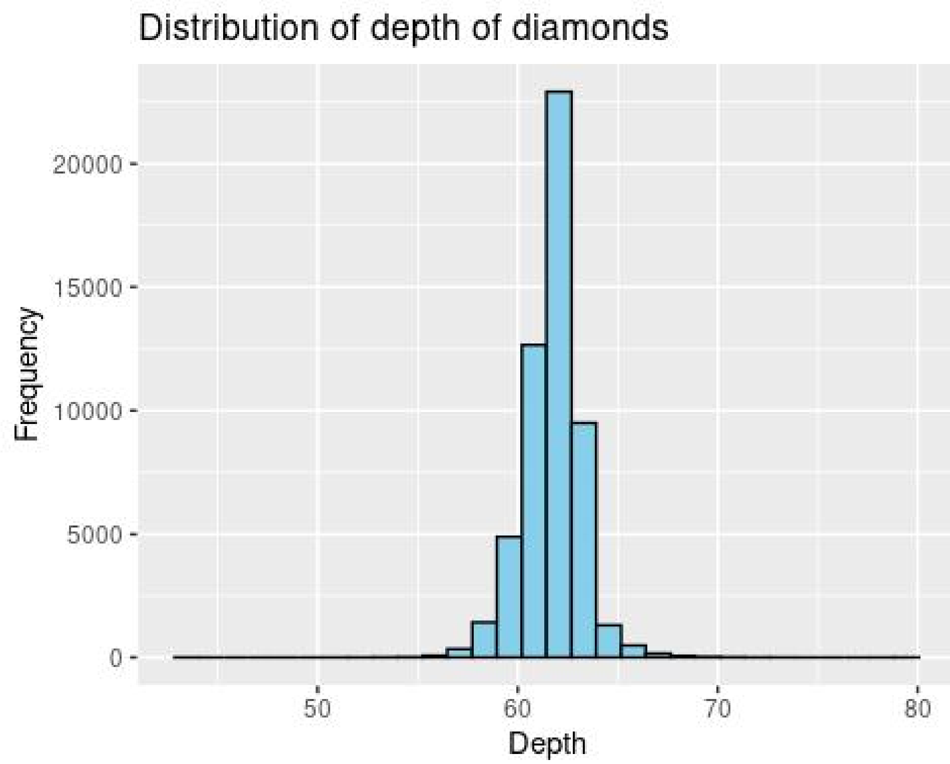

Fig 1.2

```r
ggplot(diamonds,aes(x=carat))+geom_histogram(fill="skyblue",color="blac
k")+
  labs(title="Distribution of carats of diamonds",x="Carat",y="Frequenc
y")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

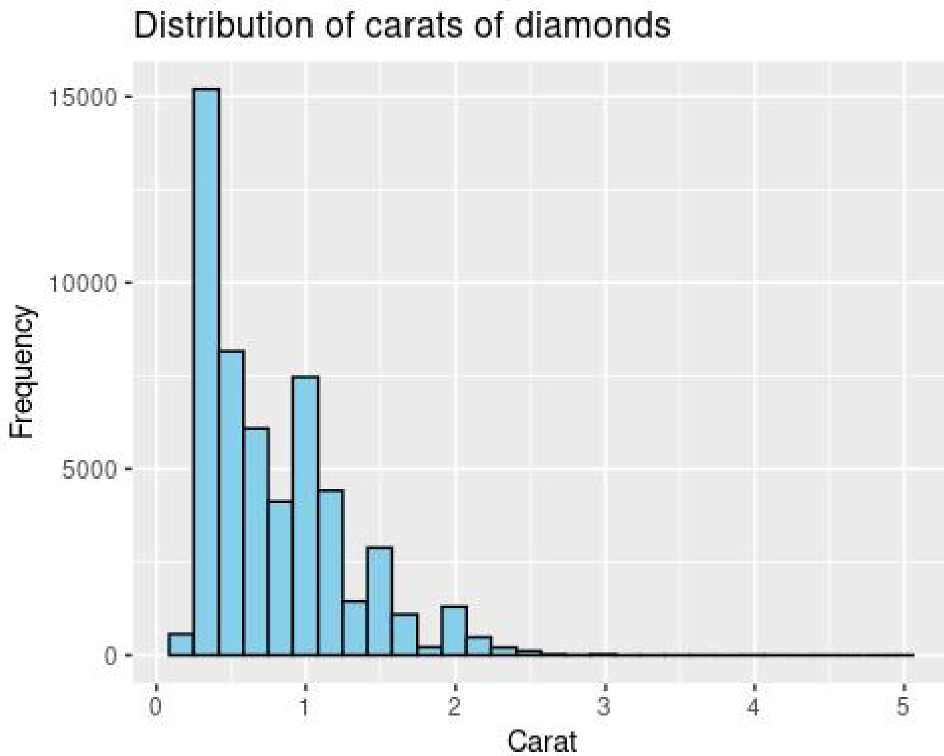

Fig 1.3

```r
ggplot(diamonds,aes(x=table))+geom_histogram(fill="skyblue",color="blac
k")+
  labs(title="Distribution of table of diamonds",x="Table",y="Frequency
")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

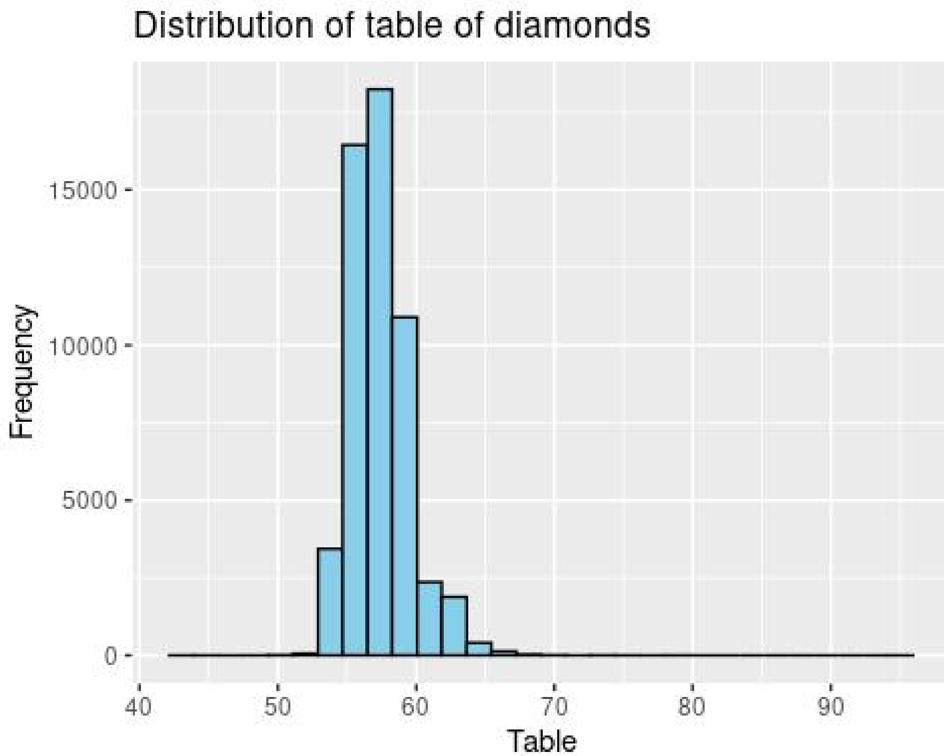

Fig 1.4

```
#bivariet analysis
ggplot(diamonds,aes(factor(color),price,fill=color))+geom_boxplot()+lab
s(title="Relationship of price attribute with color",xlab="Color",ylab=
"Price")
```

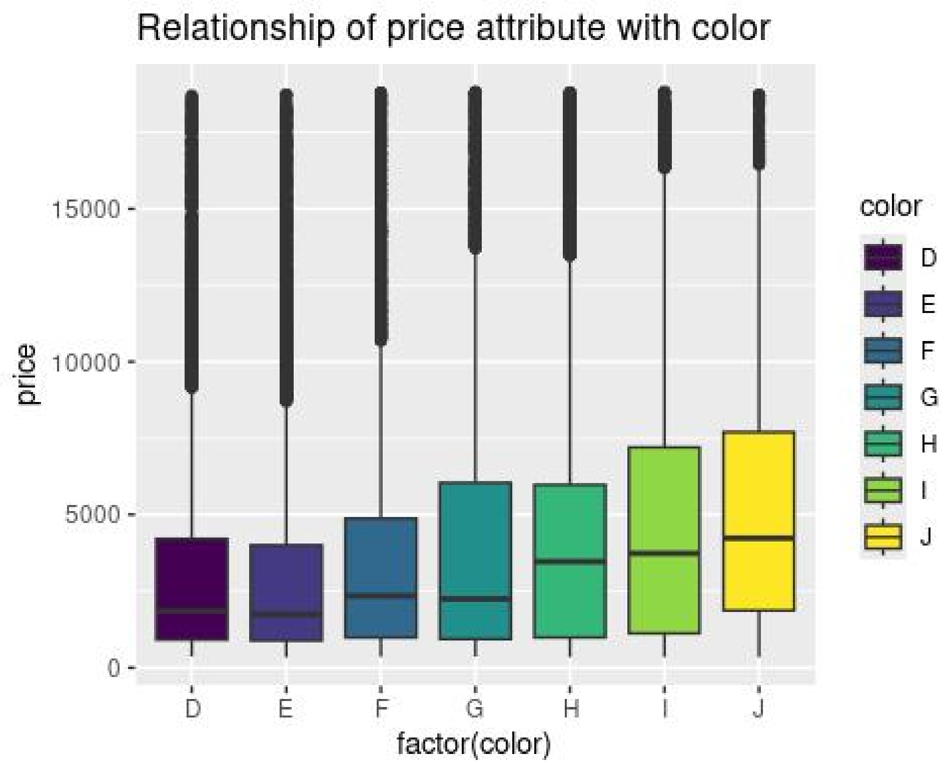

Fig 1.5

```
diamonds %>%
  group_by(clarity, cut) %>%
  ggplot(aes(x = clarity, y = price, group = cut, fill = cut)) +
  geom_boxplot()
```

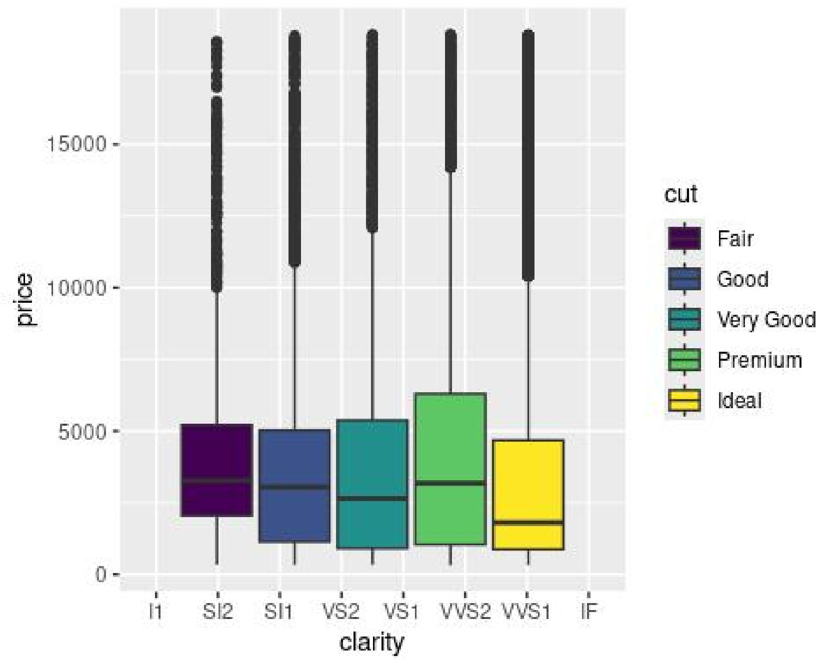

Fig 1.6

```
ggplot(diamonds,aes(x = cut, y = price, fill = cut))+geom_boxplot()+
  labs(title = "Boxplot of Price by Cut Quality",x = "Cut Quality", y =
 "Price")+theme_minimal()
```

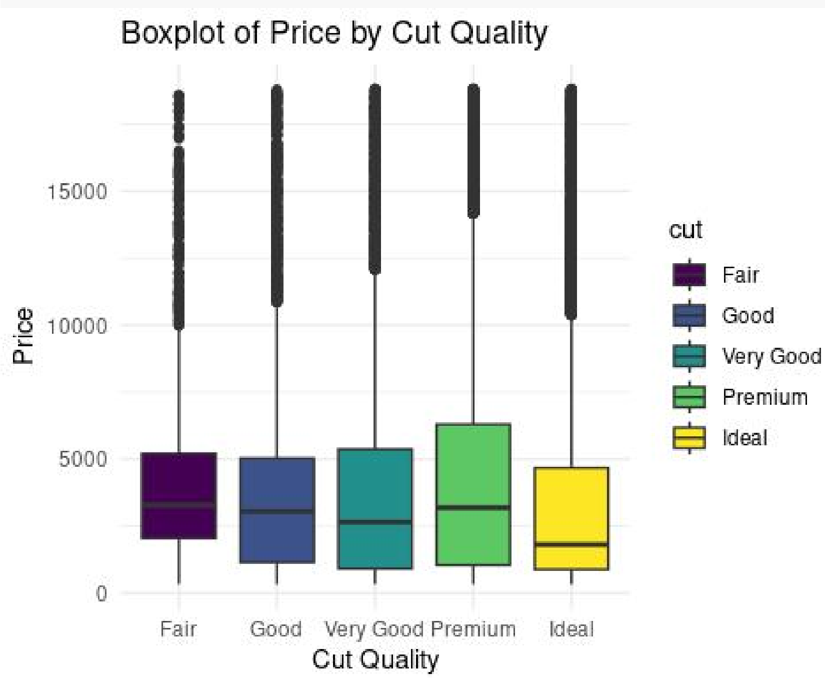

Fig1.7

```
ggplot(diamonds,aes(factor(clarity),price,fill=clarity))+geom_boxplot()
+labs(title="Diamonds price according clarity",xlab="Type of
clarity",ylab="Diamond price in US dollars" )
```

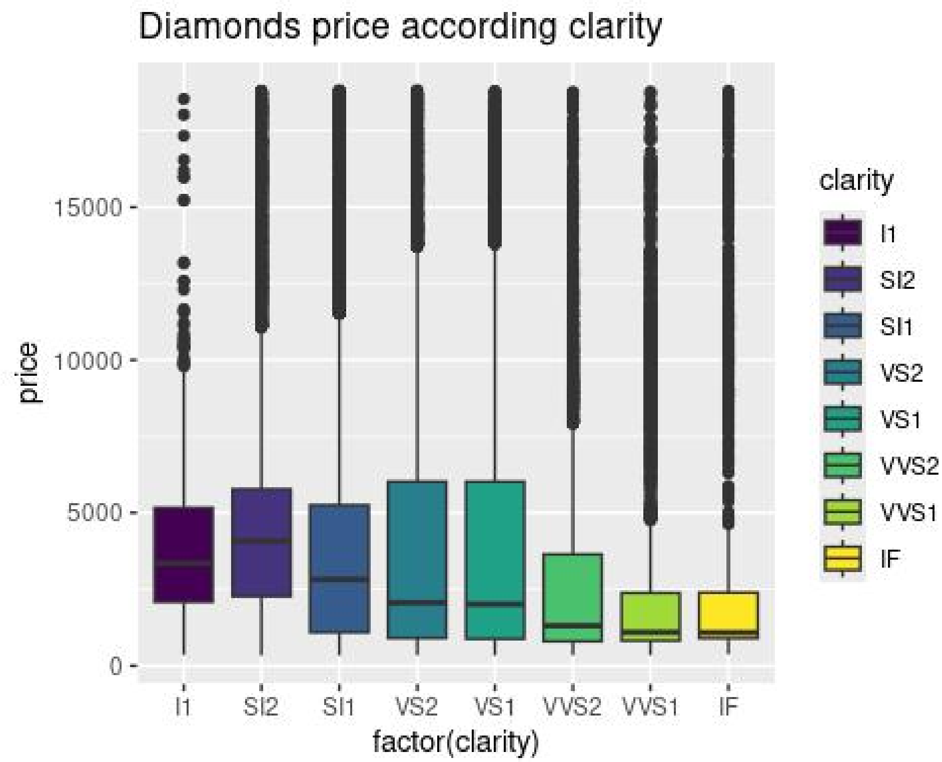

Fig 1.8

```r
ggplot(diamonds, aes(x = clarity, y = price, color = carat)) +geom_boxp
lot() +
  facet_wrap(~ clarity) + labs(title = "Price Distribution by Clarity a
nd Carat Weight",
       x = "Clarity Grade",y = "Price",
       color = "Carat Weight") +theme_minimal()
```

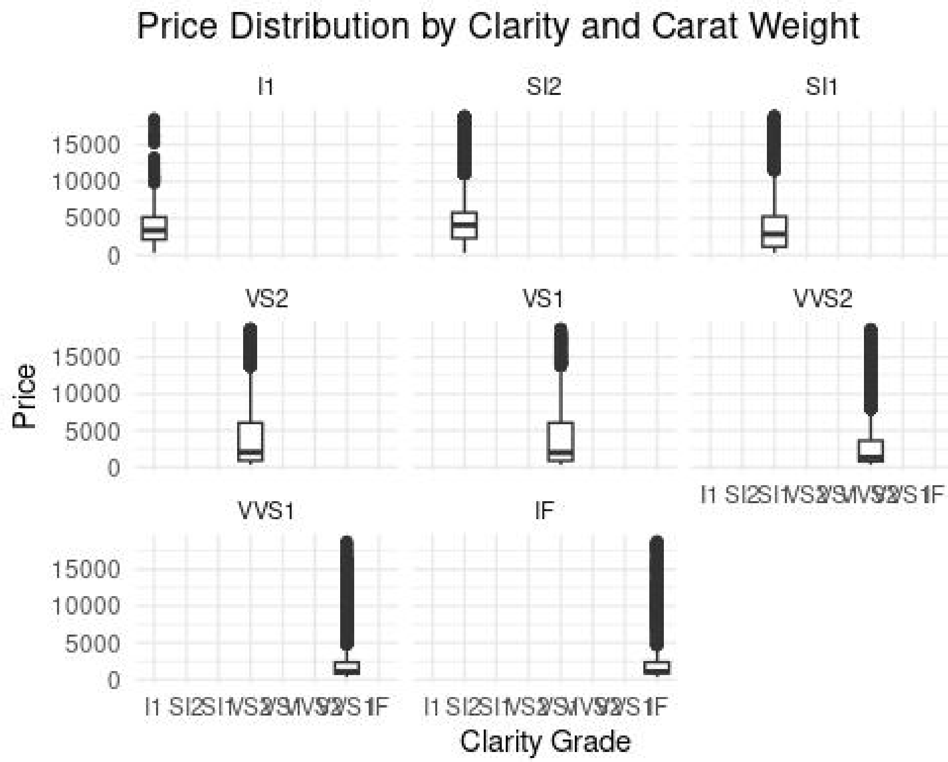

Fig 1.9

```r
#scatter plot
ggplot(diamonds, aes(x = carat, y = price,color="pink")) +
  geom_point() + labs(title = "Scatter plot of Carat vs. Price",x = "Ca
rat", y = "Price")
```

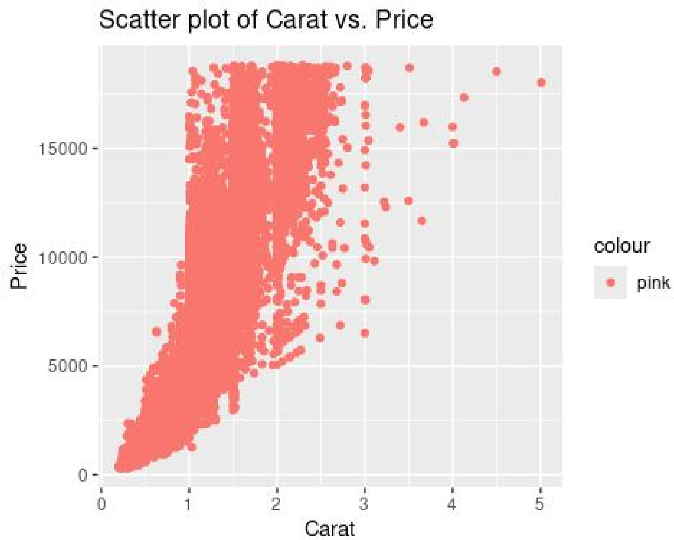

Fig 1.10

```r
ggplot(diamonds, aes(x = carat, y = price)) +geom_point() +
  geom_smooth(method = "lm", se = FALSE) + labs(title = "Price vs. Cara
t Weight",x = "Carat Weight",y = "Price ") +theme_minimal()

## `geom_smooth()` using formula = 'y ~ x'
```

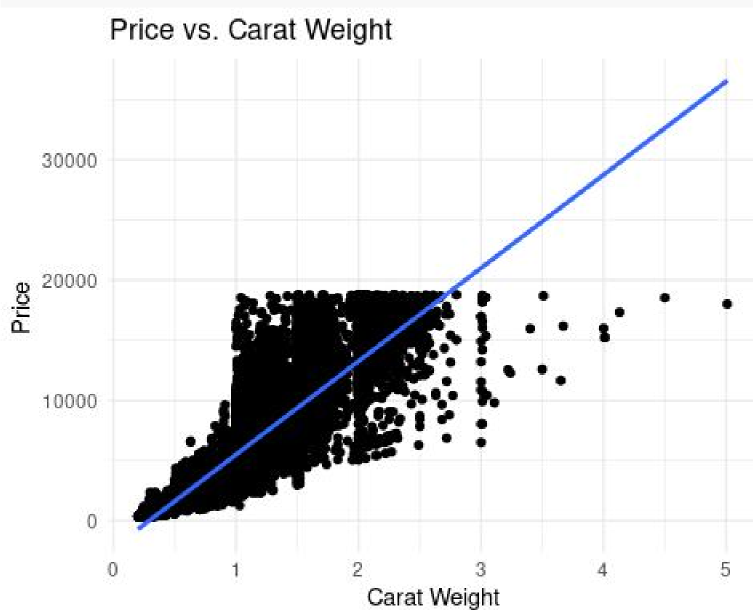

Fig 1.11

```
ggplot(diamonds, aes(x = table, y = price,color="pink")) +
  geom_point(alpha = 0.5) + labs(title = "Scatter plot of Width of top
of diamond vs. Price",
       x = "Table", y = "Price") +theme_minimal()
```

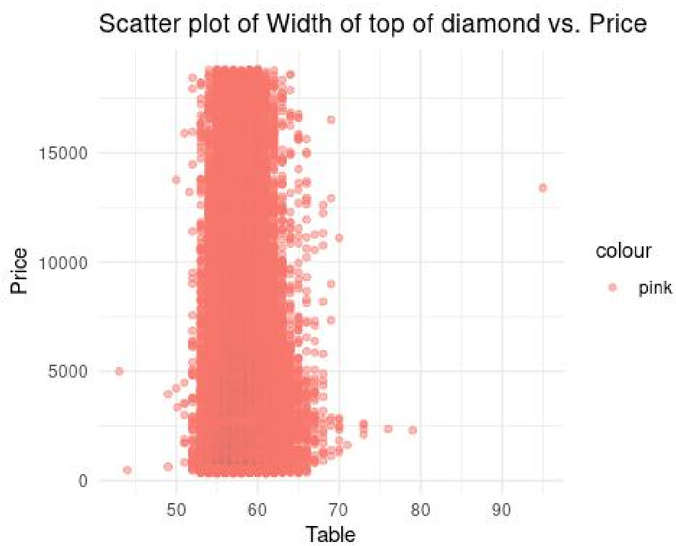

Fig 1.12

```
ggplot(diamonds, aes(x = depth, y = price,color="pink")) +
  geom_point(alpha = 0.5) + labs(title = "Scatter plot of Depth vs. Pri
ce",x = "Depth", y = "Price") +theme_minimal()
```

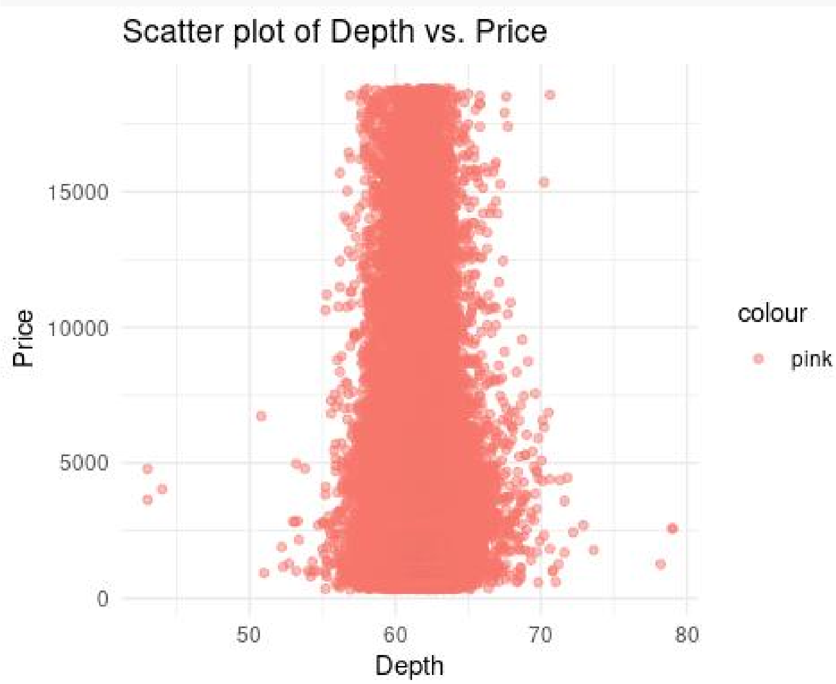

Fig 1.13

```
ggplot(diamonds,aes(x = x, y = z,color="pink")) +
  geom_point(alpha = 0.5)+labs(title = "Scatter plot of Length vs. 
Depth in mm",
      x = "length in mm", y = "depth in mm") +theme_minimal()
```

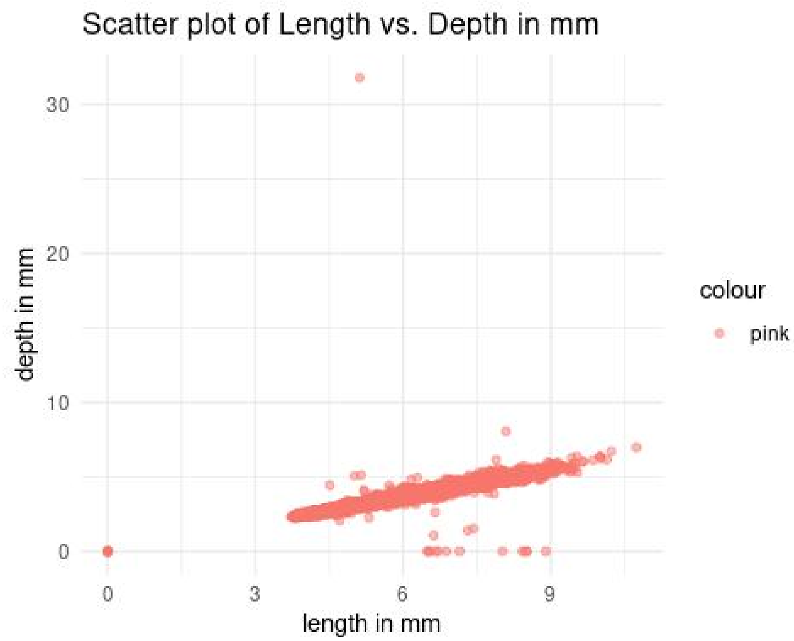

Fig 1.14

```
ggplot(diamonds, aes(x = color, y = price, color = clarity)) +geom_poin
t() +  labs(title = "Price vs. Color by Clarity",x = "Color Grade",y = 
"Price",color = "Clarity") +theme_minimal()
```

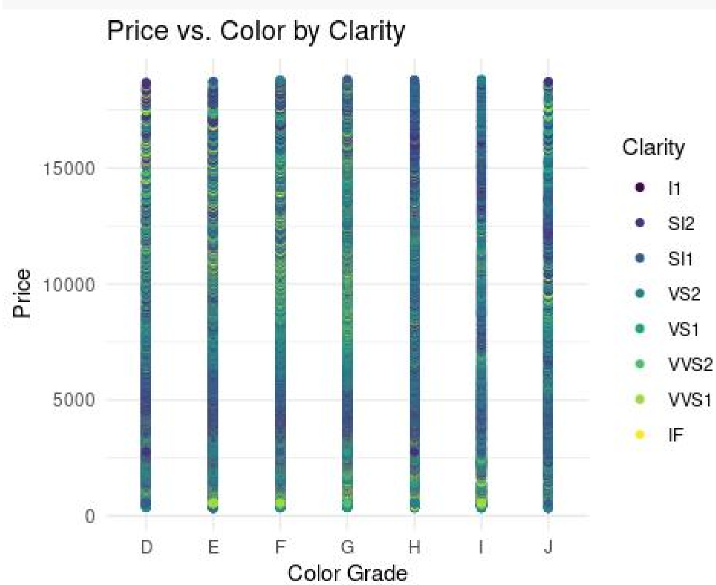

Fig 1.15

```
#multivariet analysis
numerical_attributes=diamonds[c("carat","depth","table","price","x","y",
"z")]
correlation_matrix=cor(numerical_attributes)
data1= melt(correlation_matrix)
ggplot(data1, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  labs(title = "Correlation Heatmap",x="numerical
attributes",y="numerical attributes")
```

Fig 1.16