

R Diamonds Dataset

Bibek Sapkota

Install and Import ggplot2, factoextra and psych packages

```
packages_to_install <- c("ggplot2", "factoextra", "psych")

for (package_name in packages_to_install) {
  if (!requireNamespace(package_name, quietly = TRUE)) {
    install.packages(package_name)
  }
}

library(ggplot2)
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(psych)

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##       %+%, alpha
```

Loading the data into R

```
data("diamonds")
```

task 1: Displaying first 6 rows data

```
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

task 2: Displaying last 6 rows of data

`tail(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  0.72 Premium D     SI1     62.7    59  2757  5.69  5.73  3.58
## 2  0.72 Ideal   D     SI1     60.8    57  2757  5.75  5.76  3.5 
## 3  0.72 Good   D     SI1     63.1    55  2757  5.69  5.75  3.61
## 4  0.7  Very Good D     SI1     62.8    60  2757  5.66  5.68  3.56
## 5  0.86 Premium H     SI2     61      58  2757  6.15  6.12  3.74
## 6  0.75 Ideal   D     SI2     62.2    55  2757  5.83  5.87  3.64
```

task 3: Displaying more than 6 rows of data

```
head(diamonds, 10)
```

```
## # A tibble: 10 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
## 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
```

task 4: Checking the structure of data

```
str(diamonds)
```

```
## # tibble [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 ...
```

task 5: Checking the dimension of data

```
dim(diamonds)
```

```
## [1] 53940    10
```

task 6: Summarizing the data

```
summary(diamonds)
```

```
##      carat          cut       color     clarity      depth
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065   Min.   :43.00
##  1st Qu.:0.4000  Good    : 4906   E: 9797   VS2    :12258   1st Qu.:61.00
##  Median :0.7000  Very Good:12082  F: 9542   SI2    : 9194   Median :61.80
##  Mean   :0.7979  Premium  :13791   G:11292   VS1    : 8171   Mean   :61.75
##  3rd Qu.:1.0400 Ideal    :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50
##  Max.   :5.0100                    I: 5422   VVS1   : 3655   Max.   :79.00
##                               J: 2808   (Other): 2531
##      table         price        x           y
##  Min.   :43.00   Min.   : 326   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.00  1st Qu.: 950   1st Qu.: 4.710   1st Qu.: 4.720
##  Median :57.00  Median : 2401   Median : 5.700   Median : 5.710
##  Mean   :57.46  Mean   : 3933   Mean   : 5.731   Mean   : 5.735
##  3rd Qu.:59.00  3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :95.00  Max.   :18823   Max.   :10.740   Max.   :58.900
##
##      z
##  Min.   : 0.000
##  1st Qu.: 2.910
##  Median : 3.530
##  Mean   : 3.539
##  3rd Qu.: 4.040
##  Max.   :31.800
##
```

task 7: Describing the data

```
describe(diamonds)
```

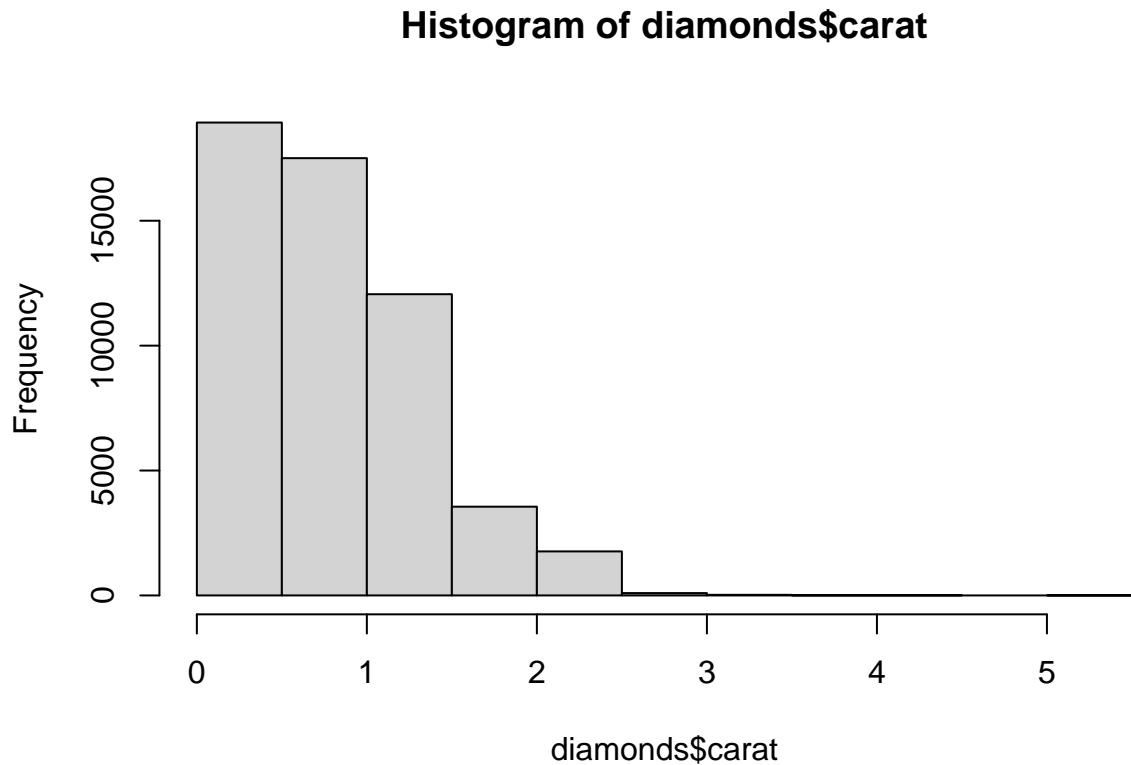
```
##      vars     n   mean     sd median trimmed    mad   min   max
##  ## carat     1 53940  0.80  0.47   0.70   0.73  0.47  0.2  5.01
##  ## cut*     2 53940  3.90  1.12   4.00   4.04  1.48  1.0  5.00
##  ## color*   3 53940  3.59  1.70   4.00   3.55  1.48  1.0  7.00
##  ## clarity* 4 53940  4.05  1.65   4.00   3.91  1.48  1.0  8.00
##  ## depth    5 53940  61.75 1.43  61.80  61.78  1.04 43.0 79.00
##  ## table    6 53940  57.46 2.23  57.00  57.32  1.48 43.0 95.00
##  ## price    7 53940 3932.80 3989.44 2401.00 3158.99 2475.94 326.0 18823.00
##  ## x         8 53940  5.73  1.12   5.70   5.66  1.38  0.0 10.74
##  ## y         9 53940  5.73  1.14   5.71   5.66  1.36  0.0  58.90
##  ## z        10 53940  3.54  0.71   3.53   3.49  0.85  0.0 31.80
##      range   skew kurtosis     se
##  ## carat    4.81  1.12    1.26  0.00
```

```
## cut*      4.00 -0.72   -0.40  0.00
## color*    6.00  0.19   -0.87  0.01
## clarity*  7.00  0.55   -0.39  0.01
## depth     36.00 -0.08    5.74  0.01
## table     52.00  0.80    2.80  0.01
## price    18497.00  1.62    2.18 17.18
## x         10.74  0.38   -0.62  0.00
## y         58.90  2.43   91.20  0.00
## z         31.80  1.52   47.08  0.00
```

Visualization of diamonds dataset

task 1:Building the histogram of carat

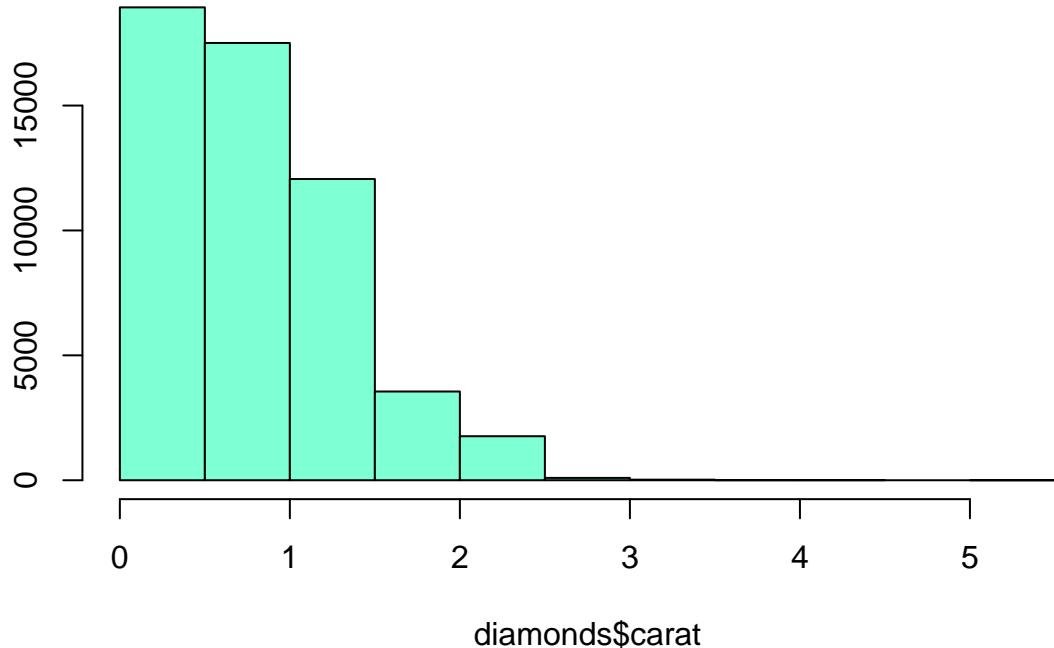
```
hist(diamonds$carat)
```



task 2:Using color to color the bin

```
hist(diamonds$carat, col='aquamarine', ylab = "")
```

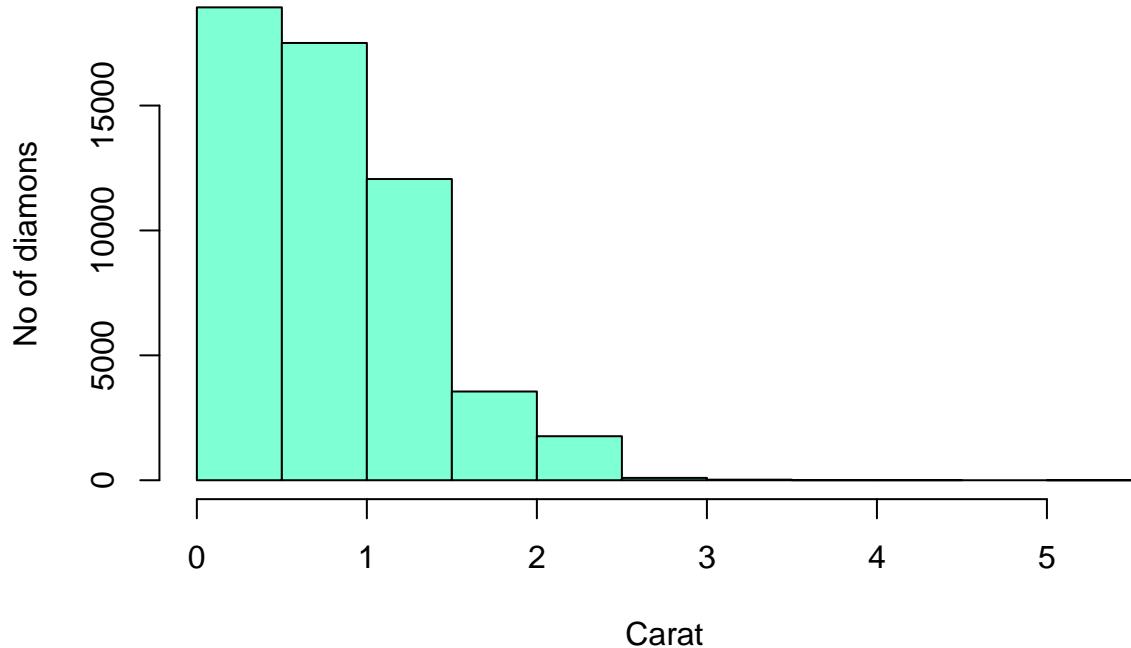
Histogram of diamonds\$carat



task 3: Changing the label of x-axis

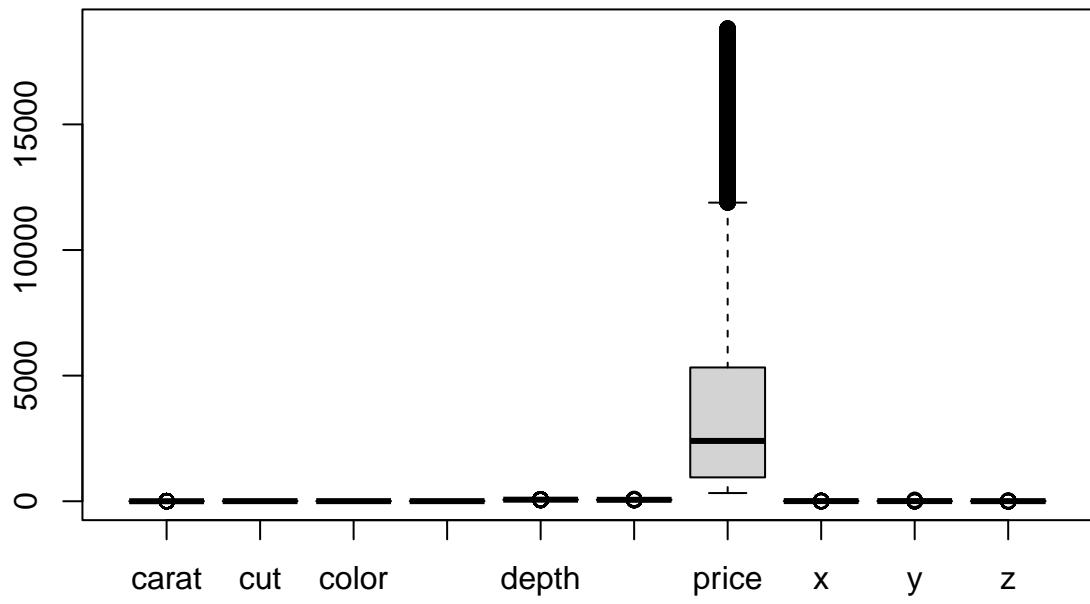
```
hist(diamonds$carat, col='aquamarine', xlab = "Carat", ylab= "No of diamonds")
```

Histogram of diamonds\$carat



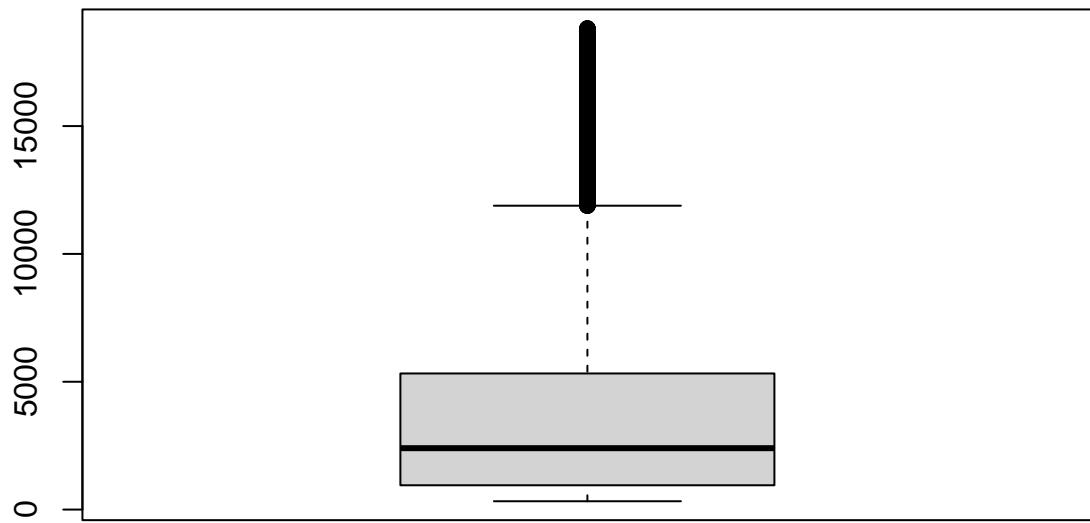
task 4: Creating a box plot of all variables in dataset

```
boxplot(diamonds)
```



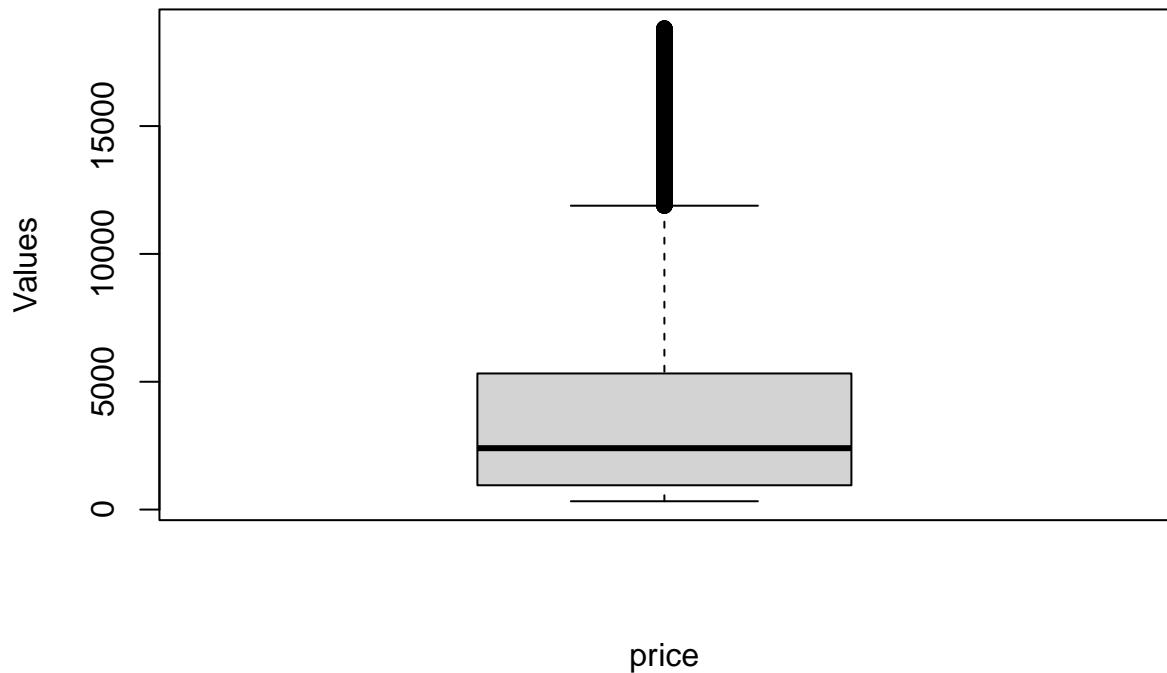
task 5: Creating the box plot of only single variable (say diamonds)

```
boxplot(diamonds$price)
```



task 6:Filling in the label on x-axis and y-axis

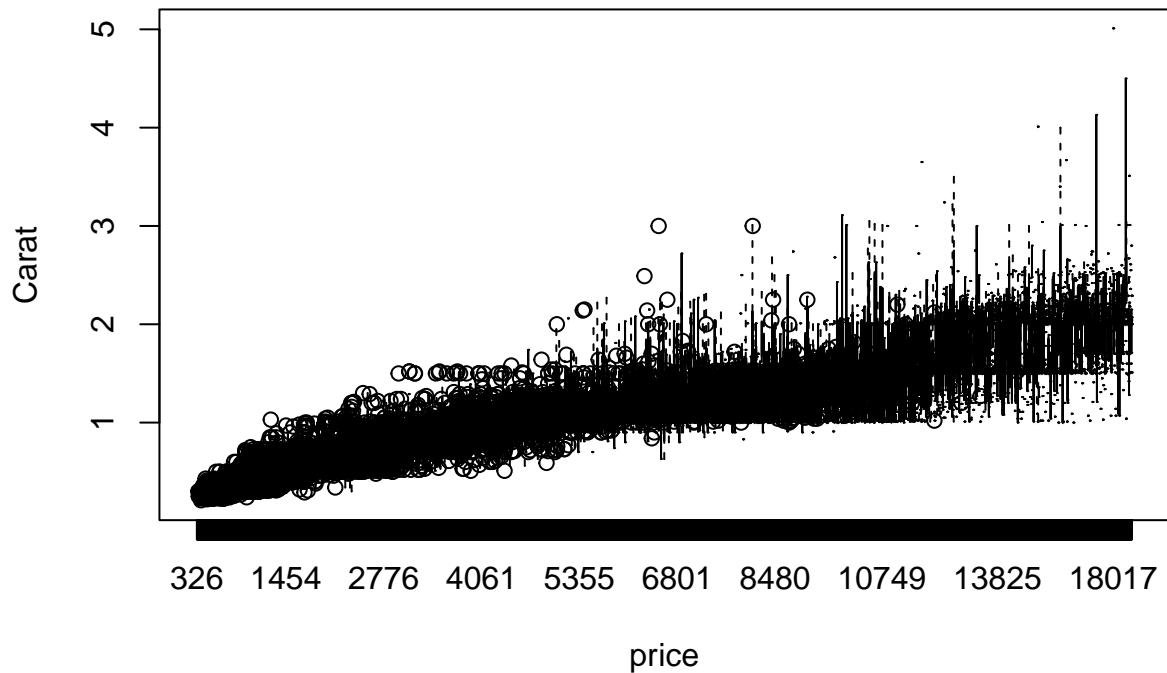
```
boxplot(diamonds$price,  
        xlab="price",  
        ylab="Values")
```



task 7:Displaying Boxplot

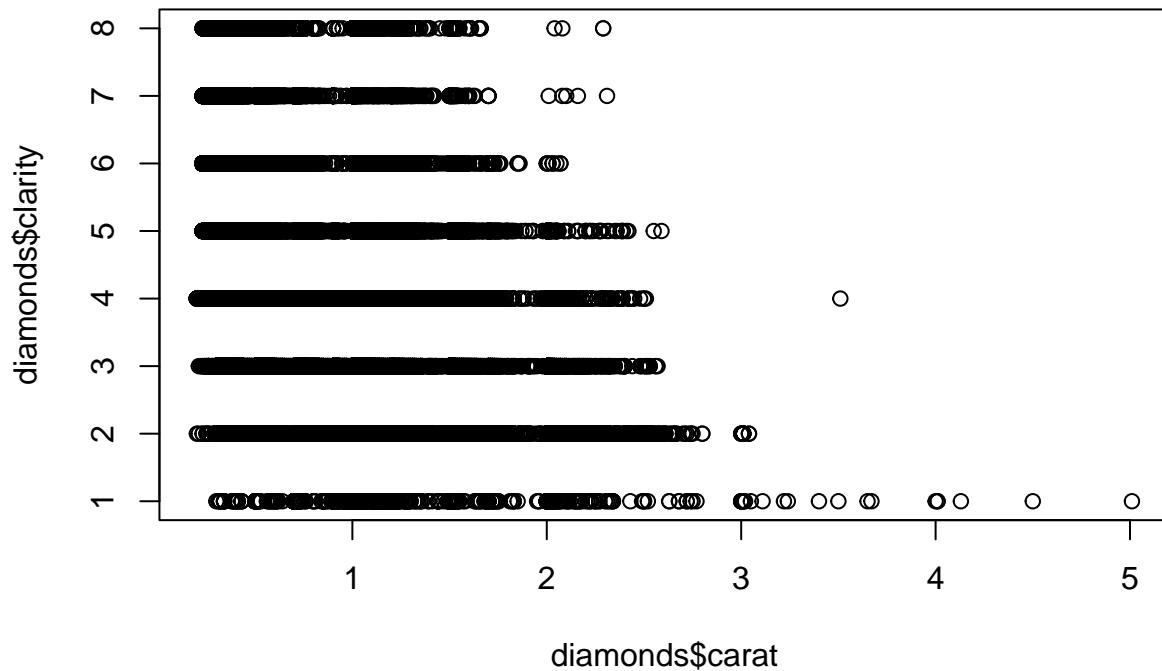
```
boxplot(carat ~ price , data = diamonds,  
xlab = "price",  
ylab = "Carat",  
main = "Boxplot of carat by price")
```

Boxplot of carat by price



task 8:Plotting the Scatter plot

```
plot(diamonds$carat, diamonds$clarity)
```



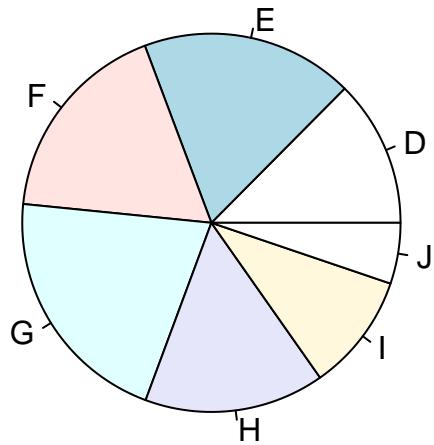
task 9: Calculating the distribution of transmissions

```
trans_dist <- table(diamonds$color)
```

task 10:Creating pie chart

```
pie(trans_dist,
  labels = c("D", "E", "F", "G", "H", "I", "J"),
  main = "colors of diamonds")
```

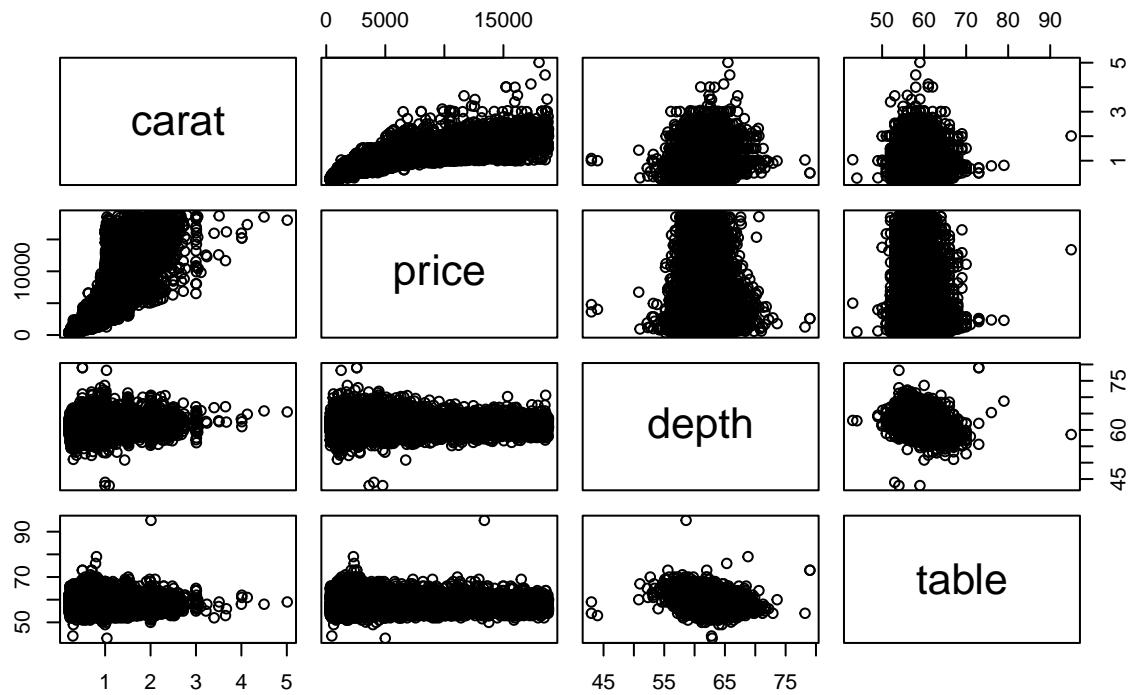
colors of diamonds



task 11:Using Scatter plot matrix for Displaying selected variables

```
pairs(diamonds[, c("carat", "price", "depth", "table")],  
      main = "Scatterplot Matrix")
```

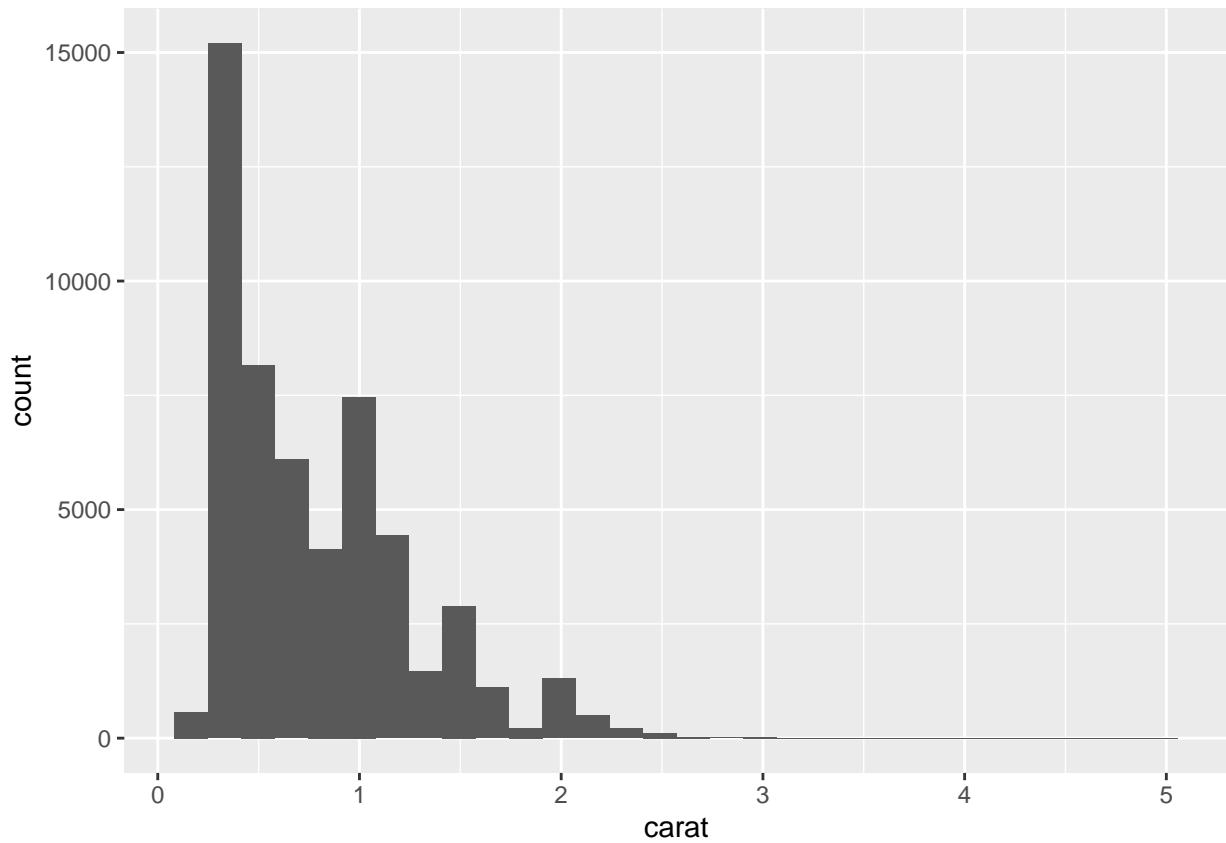
Scatterplot Matrix



task 12: Plotting Histogram of carat

```
ggplot(diamonds, aes(x = carat),) +  
  geom_histogram()
```

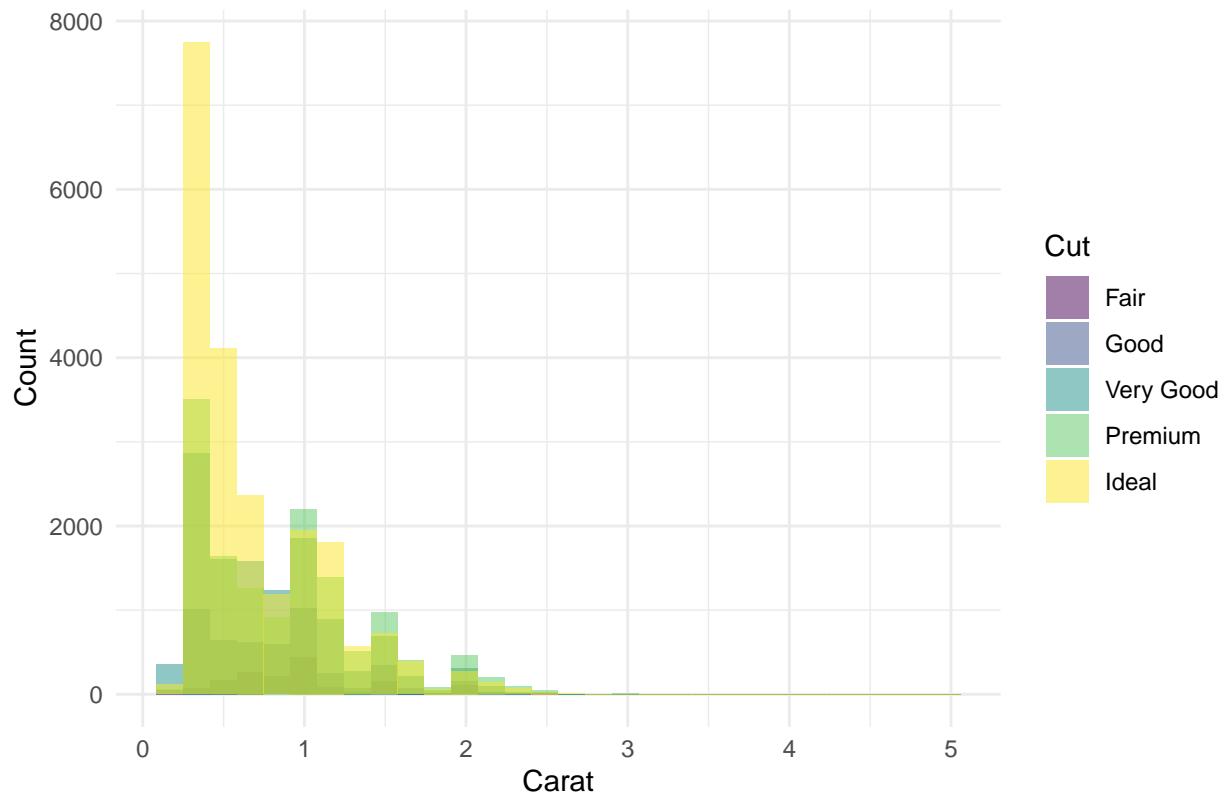
```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



task 13: Using color to color the bin

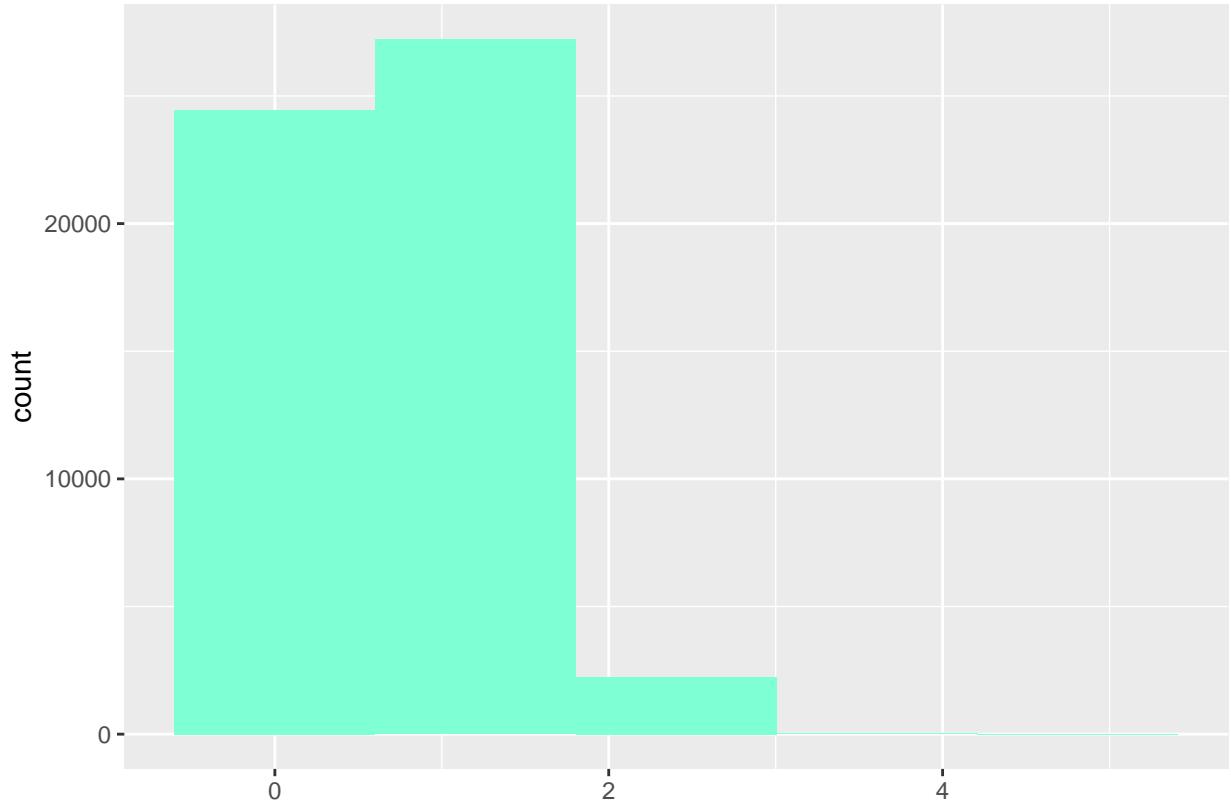
```
ggplot(diamonds, aes(x = carat, fill = cut)) +  
  geom_histogram(bins = 30, position = "identity", alpha = 0.5) +  
  theme_minimal() +  
  labs(title = "Distribution of Diamond Carat Sizes by Cut",  
       x = "Carat",  
       y = "Count",  
       fill = "Cut")
```

Distribution of Diamond Carat Sizes by Cut



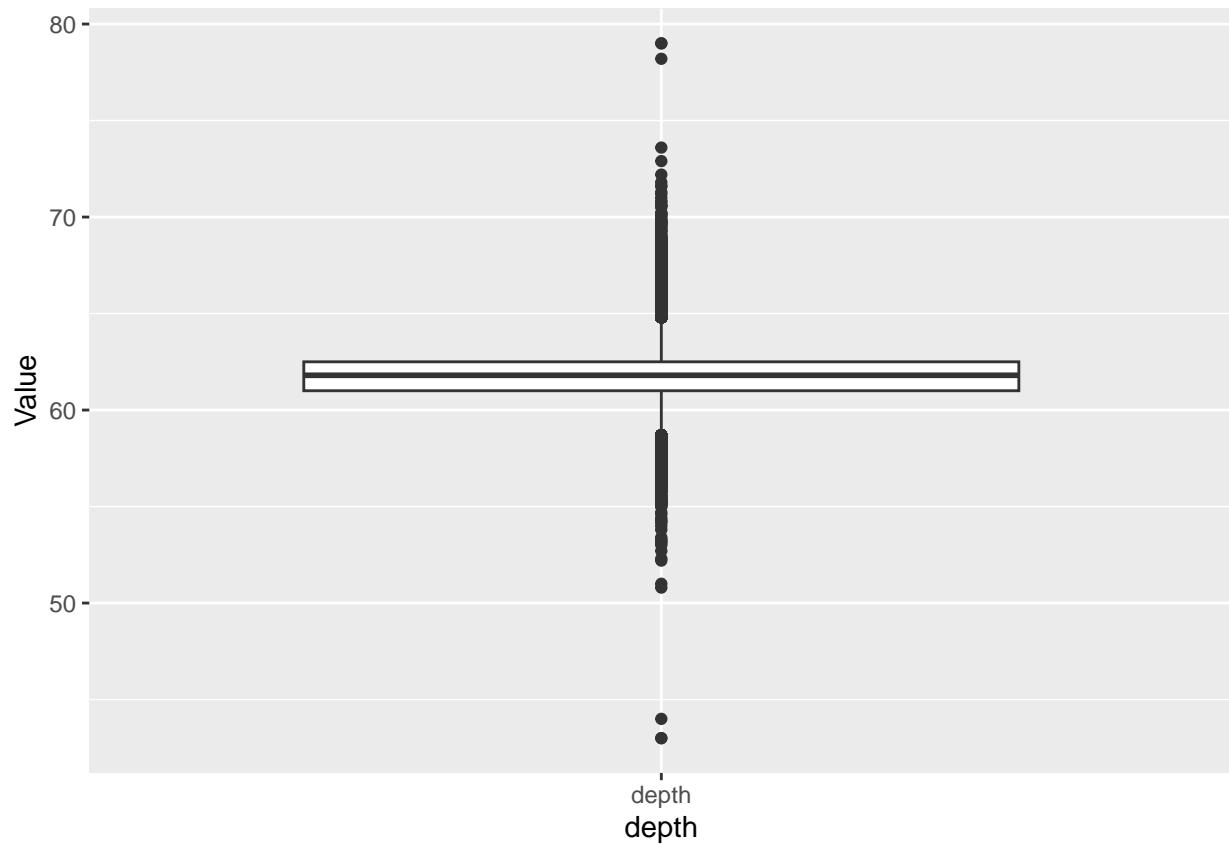
task 14: Changing the label of x-axis

```
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram(fill = 'aquamarine', bins = 5) +  
  xlab("")
```



task 15:Createing the box plot of only single variable

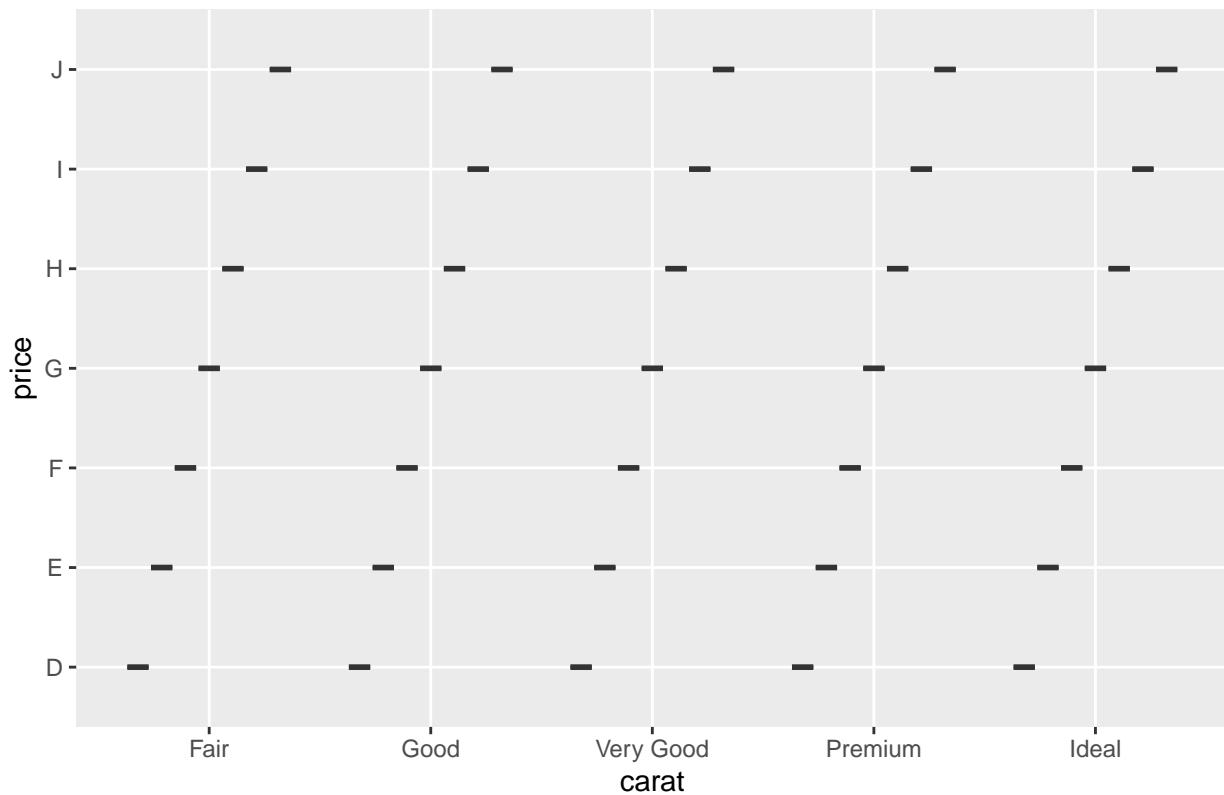
```
ggplot(diamonds, aes(x= factor('depth'),y=depth)) +  
  geom_boxplot() +  
  ylab("Value") +  
  xlab("depth")
```



task 16:Box plotting of carat by price

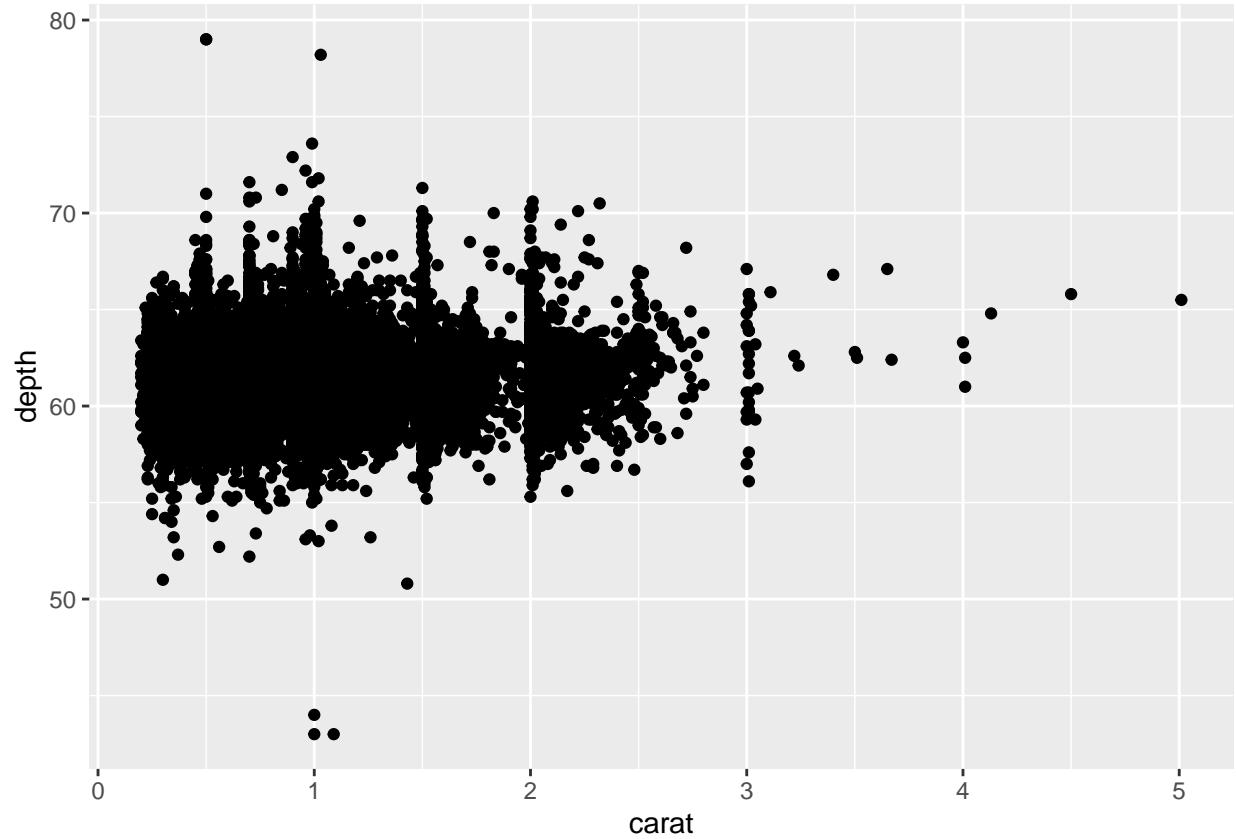
```
ggplot(diamonds, aes(x= factor(cut),y=color))+
  geom_boxplot()+
  xlab("carat") +
  ylab("price") +
  ggtitle("Box plot of carat by price")
```

Box plot of carat by price



task 17: Plotting the Scatter Plot

```
ggplot(diamonds, aes(x = carat, y = depth)) +  
  geom_point()
```



Applying K-means Clustering

task 1:set random seed

```
set.seed(7696)
```

task 2: checking the class and dimension of data

```
class(diamonds)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
dim(diamonds)
```

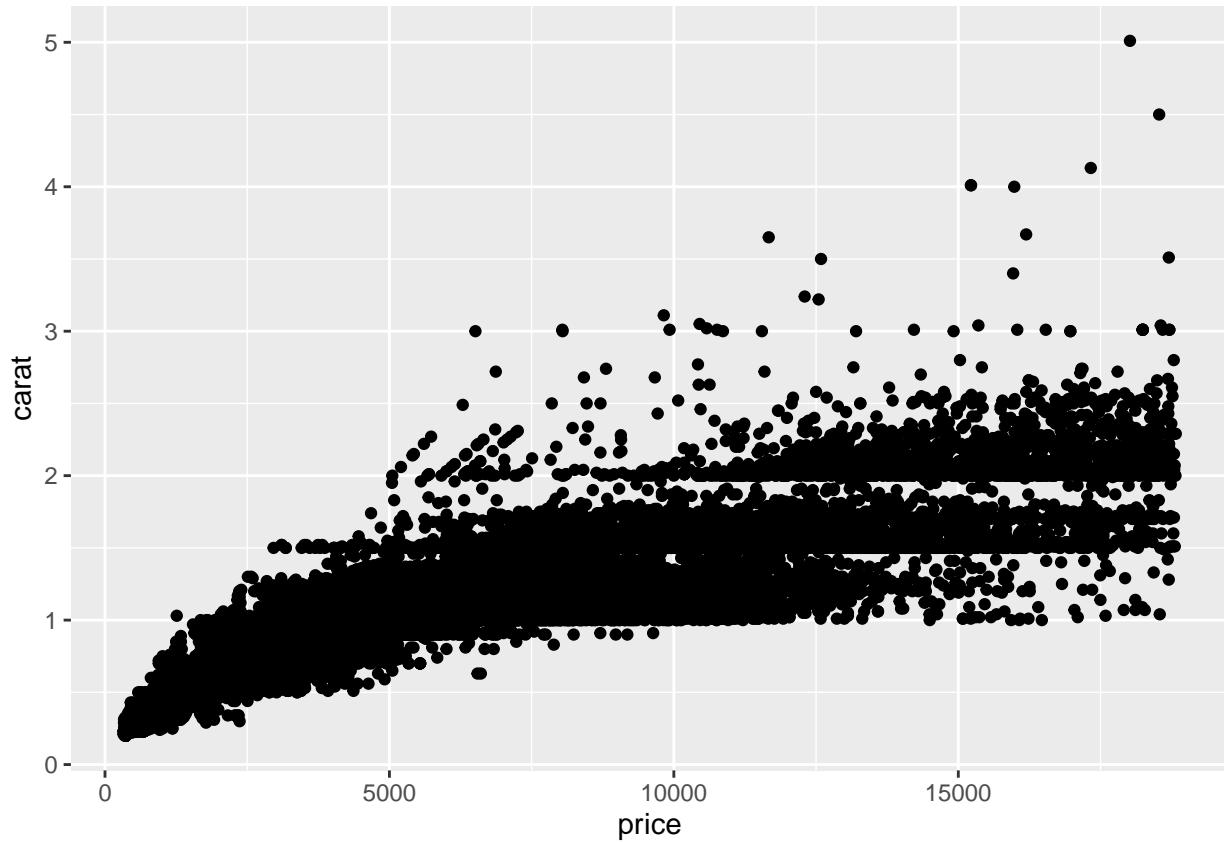
```
## [1] 53940    10
```

task 3:filter data by column name

```
filtered_data1= diamonds[, c("price", "carat")]
```

task 4:Visualizing the data

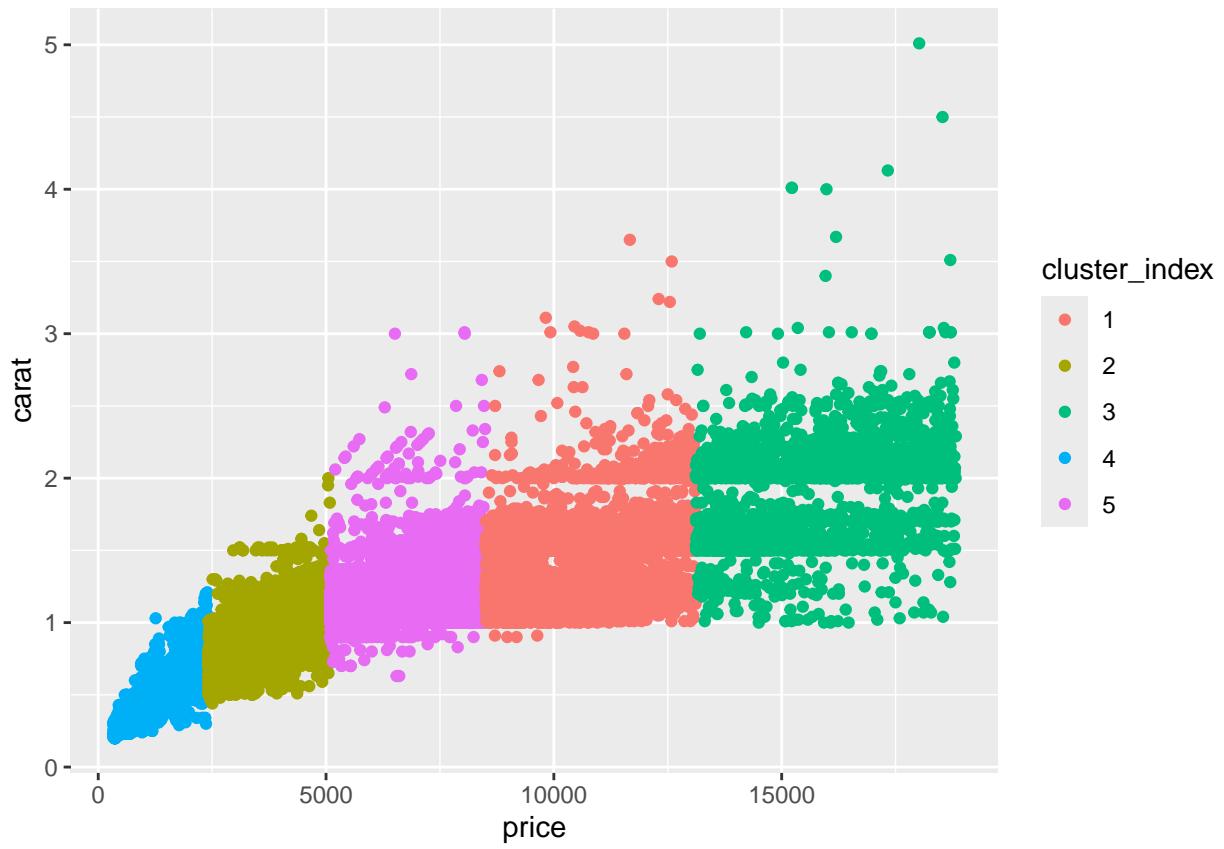
```
ggplot(diamonds, aes(x= price, y= carat))+
  geom_point()
```



task 5:Finding 5 clusters using k-means clustering and viewing the details

task 6: Visualizing the cluster

```
cluster_index = as.factor(clusters$cluster)
ggplot(filtered_data1, aes(x=price, y=carat, color= cluster_index )) +
  geom_point()
```



```
fviz_cluster(clusters, data= filtered_data1, geom= c("point"))
```

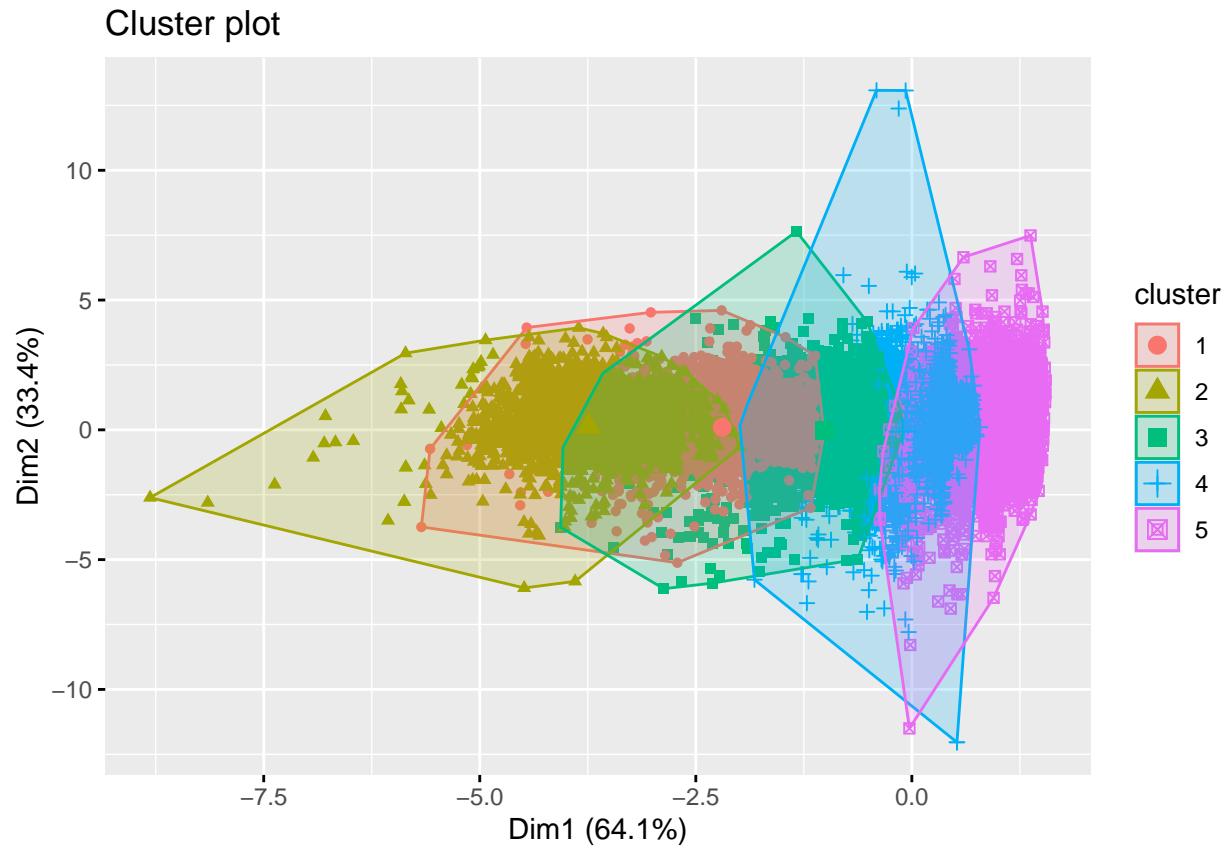


task 7:Filtering the data

```
filtered_data2 = diamonds[, c("depth", "price", "carat")]
```

task 9: Finding 5 clusters using k-means clustering and viewing the details

```
cluster2= kmeans(filtered_data2, 5)  
cluster2
```

task 11: Finding 5 clusters using k-means clustering and viewing the details

```
cluster3= kmeans(filtered_data2, 5 , nstart=25)
cluster3
```


task 12: Visualizing the cluster

```
fviz_cluster(cluster3, data= filtered_data2, geom= c("point"))
```

Cluster plot

