Big Data Analytics

Assignment 1

26/10/2019

Reading the data into R workspace

```
diamond <- read.csv("DiamondData.csv")
```

Task 1.

First printing the summary of all the variables in the dataset

```
summary(diamond)
```

```
     carat               cut        color        clarity     
 Min.   : 0.200   Fair      : 1480   D: 6264   SI1    :12120  
 1st Qu.: 0.400   Good      : 4559   E: 9066   VS2    :11406  
 Median : 0.700   Ideal     :19918   F: 8837   SI2    : 8486  
 Mean   : 0.907   Premium   :12826   G:10493   VS1    : 7563  
 3rd Qu.: 1.050   Very Geod: 2242   H: 7705   VVS2   : 4692  
 Max.   :49.990   Very Good: 8975   I: 5028   VVS1   : 3377  
                                    J: 2607   (Other): 2356  
     depth           table           price             x         
 Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000  
 1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  949   1st Qu.: 4.710  
 Median :61.80   Median :57.00   Median : 2401   Median : 5.700  
 Mean   :61.75   Mean   :57.46   Mean   : 3939   Mean   : 5.732  
 3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5339   3rd Qu.: 6.540  
 Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.230  
 NA's   :471     NA's   :390     NA's   :253     NA's   :221     
       y                z         
 Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 4.720   1st Qu.: 2.910  
 Median : 5.710   Median : 3.530  
 Mean   : 5.734   Mean   : 3.539  
 3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :31.800   Max.   :31.800  
 NA's   :333      NA's   :428     
```

The errors sin the dataset includes:

1. The `carat` variable varies from a minimum value of 0.2 to maximum of 5.01
2. The `Very Good` level under `cut` attribute is also mistyped as `Very Geod`
3. NA's in all the attributes
4. Recalculating the value of `depth`

Correcting the level of `cut` variable

```
diamond$cut[diamond$cut == "Very Geod"] <- "Very Good"
diamond$cut <- as.factor(as.character(diamond$cut))
summary(diamond$cut)
```

```
     Fair      Good     Ideal   Premium Very Good
     1480      4559     19918     12826     11217
```

Removing all the rows containing any NAs's in the dataset

```
diamond <- diamond[complete.cases(diamond), ]
dim(diamond)
```

```
[1] 47940    10
```

Correcting the range for `carat` variable

```
diamond <- diamond[diamond$carat >= 0.2 & diamond$carat <= 5.01,]
summary(diamond$carat)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2000  0.4000  0.7000  0.7982  1.0400  4.5000
```

```
dim(diamond)
```

```
[1] 47792    10
```

Recalculating the values for `depth` variable

```
diamond$depth <- 2*diamond$z/(diamond$x+diamond$y)
diamond <- diamond[complete.cases(diamond), ]
```

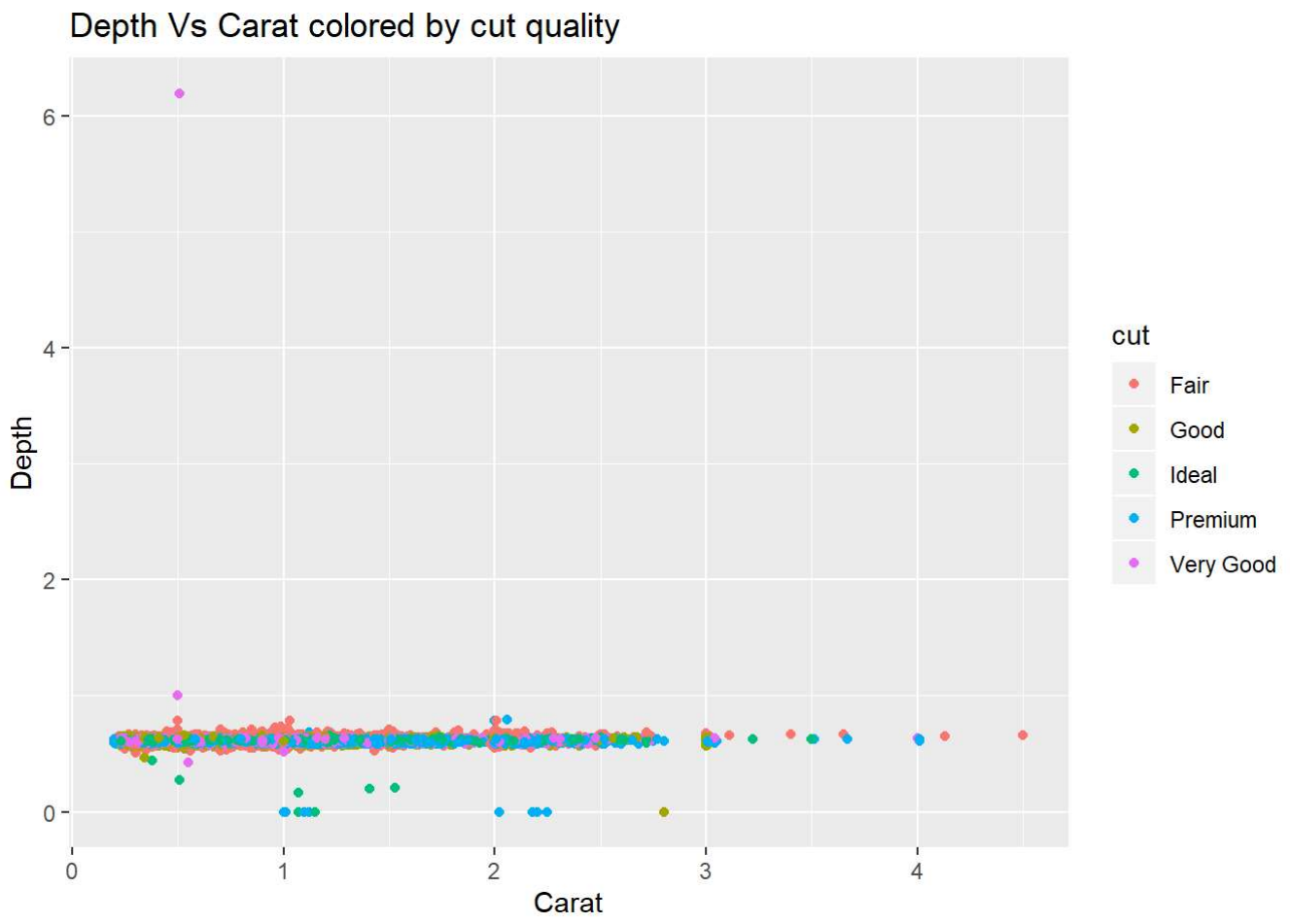
Task 2

```
summary(diamond)
```

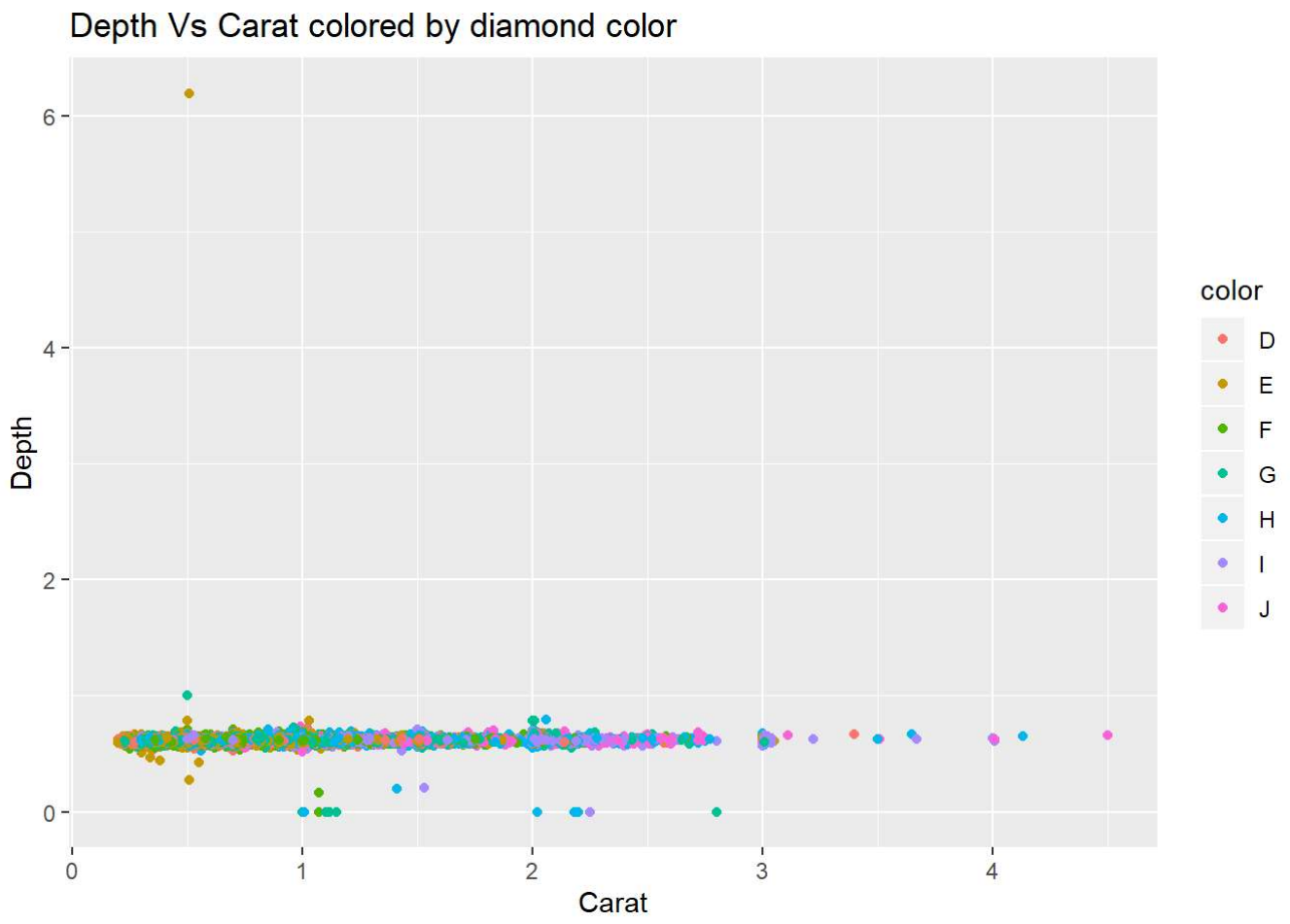
```
     carat               cut        color        clarity     
 Min.   :0.2000   Fair      : 1408   D: 5978   SI1    :11584  
 1st Qu.:0.4000   Good      : 4348   E: 8691   VS2    :10913  
 Median :0.7000   Ideal     :19077   F: 8418   SI2    : 8107  
 Mean   :0.7981   Premium   :12238   G:10028   VS1    : 7219  
 3rd Qu.:1.0400   Very Good:10714   H: 7363   VVS2   : 4484  
 Max.   :4.5000                     I: 4823   VVS1   : 3231  
                                    J: 2484   (Other): 2247  
     depth            table           price             x         
 Min.   :0.0000   Min.   :43.00   Min.   :  326   Min.   : 0.000  
 1st Qu.:0.6104   1st Qu.:56.00   1st Qu.:  948   1st Qu.: 4.710  
 Median :0.6184   Median :57.00   Median : 2401   Median : 5.700  
 Mean   :0.6174   Mean   :57.46   Mean   : 3938   Mean   : 5.732  
 3rd Qu.:0.6252   3rd Qu.:59.00   3rd Qu.: 5342   3rd Qu.: 6.540  
 Max.   :6.1928   Max.   :95.00   Max.   :18823   Max.   :10.230  

       y                z         
 Min.   : 3.680   Min.   : 0.000  
 1st Qu.: 4.720   1st Qu.: 2.910  
 Median : 5.710   Median : 3.530  
 Mean   : 5.735   Mean   : 3.539  
 3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :31.800   Max.   :31.800  
```

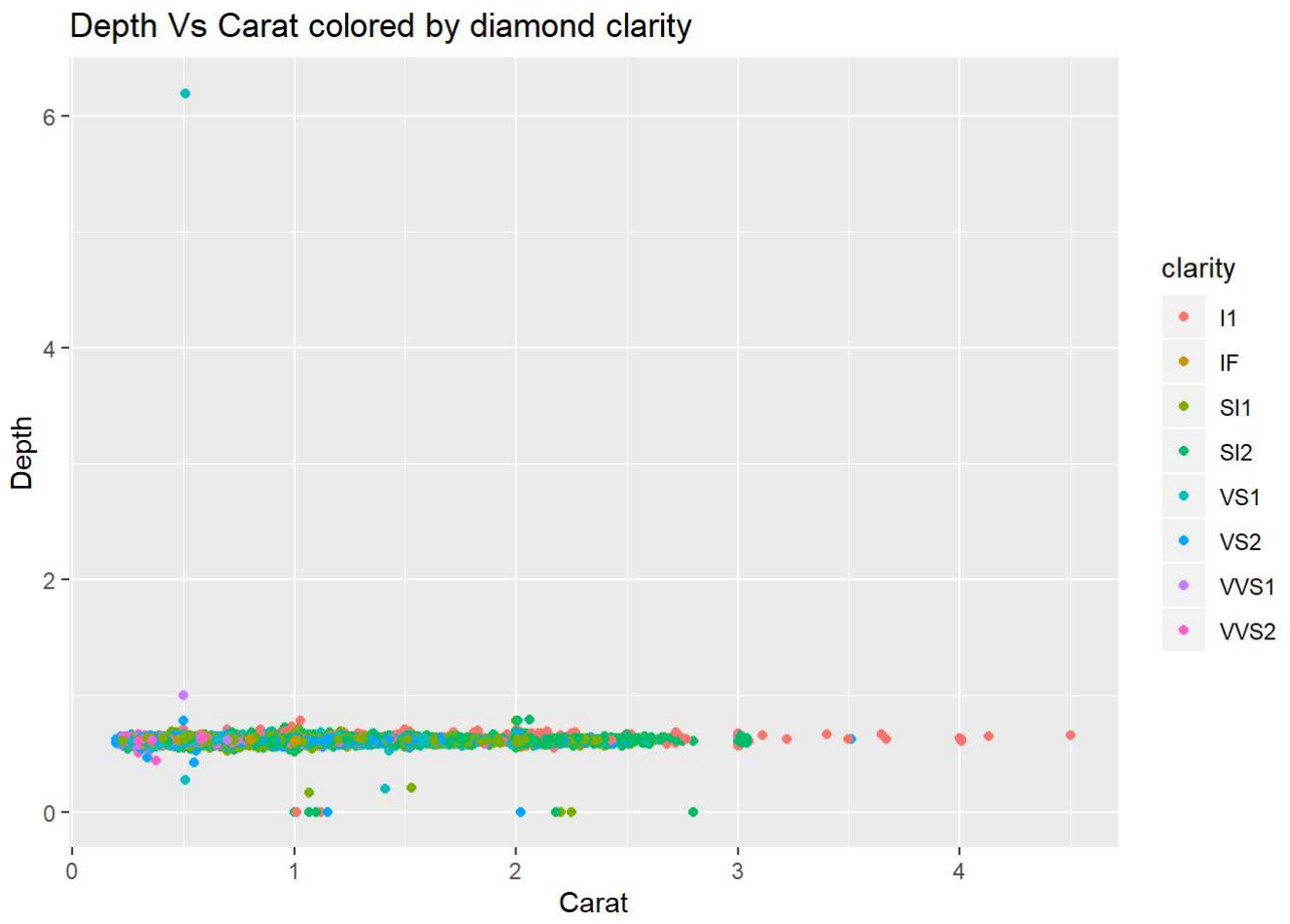
```
ggplot(diamond, aes(x=carat,y=depth)) +
  geom_point(aes(col=cut)) +
  labs(
    x = "Carat",
    y = "Depth",
    title = "Depth Vs Carat colored by cut quality"
  )
```



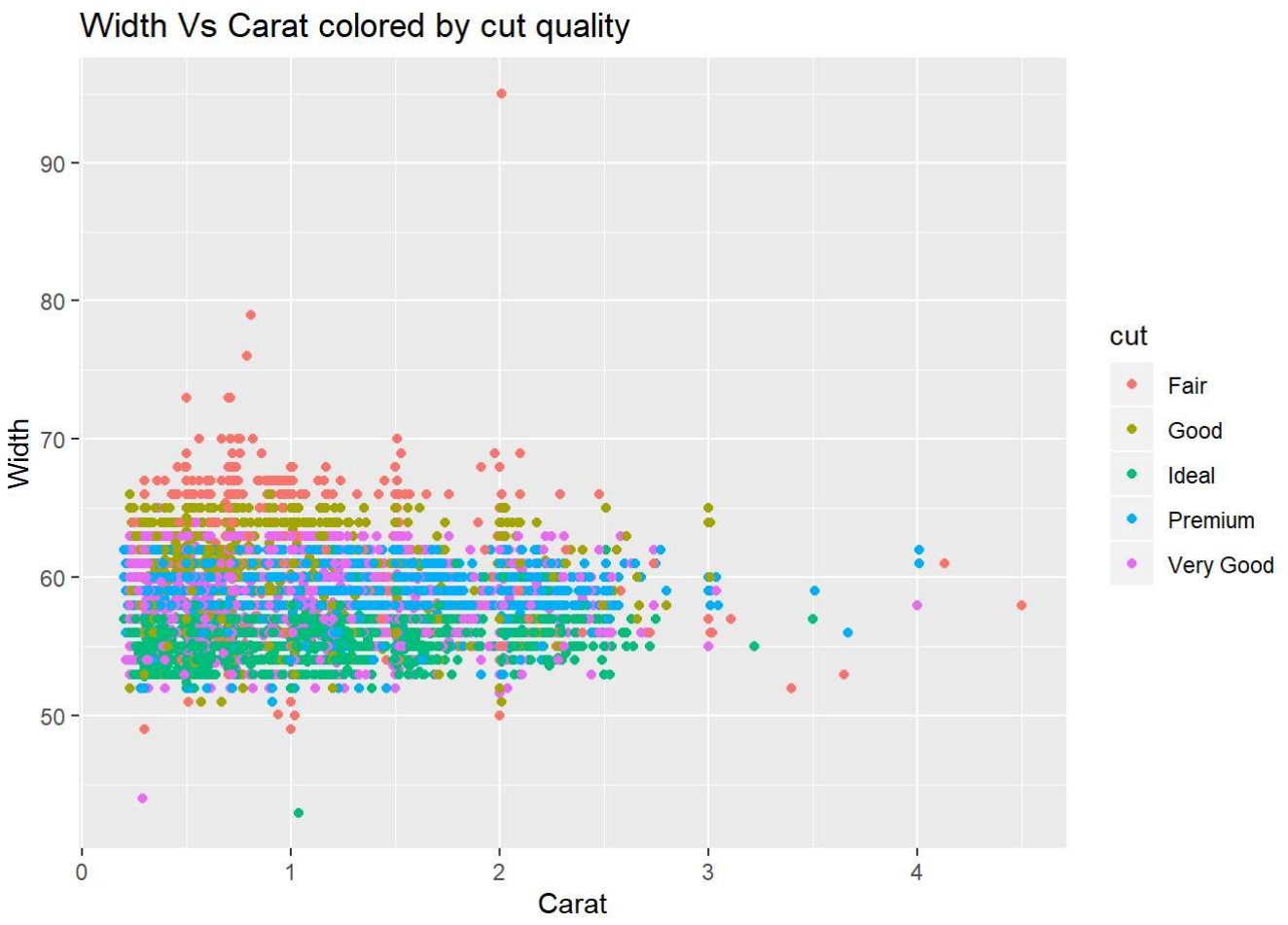

```
ggplot(diamond, aes(x=carat,y=depth)) +
  geom_point(aes(col=color)) +
  labs(
    x = "Carat",
    y = "Depth",
    title = "Depth Vs Carat colored by diamond color"
  )
```



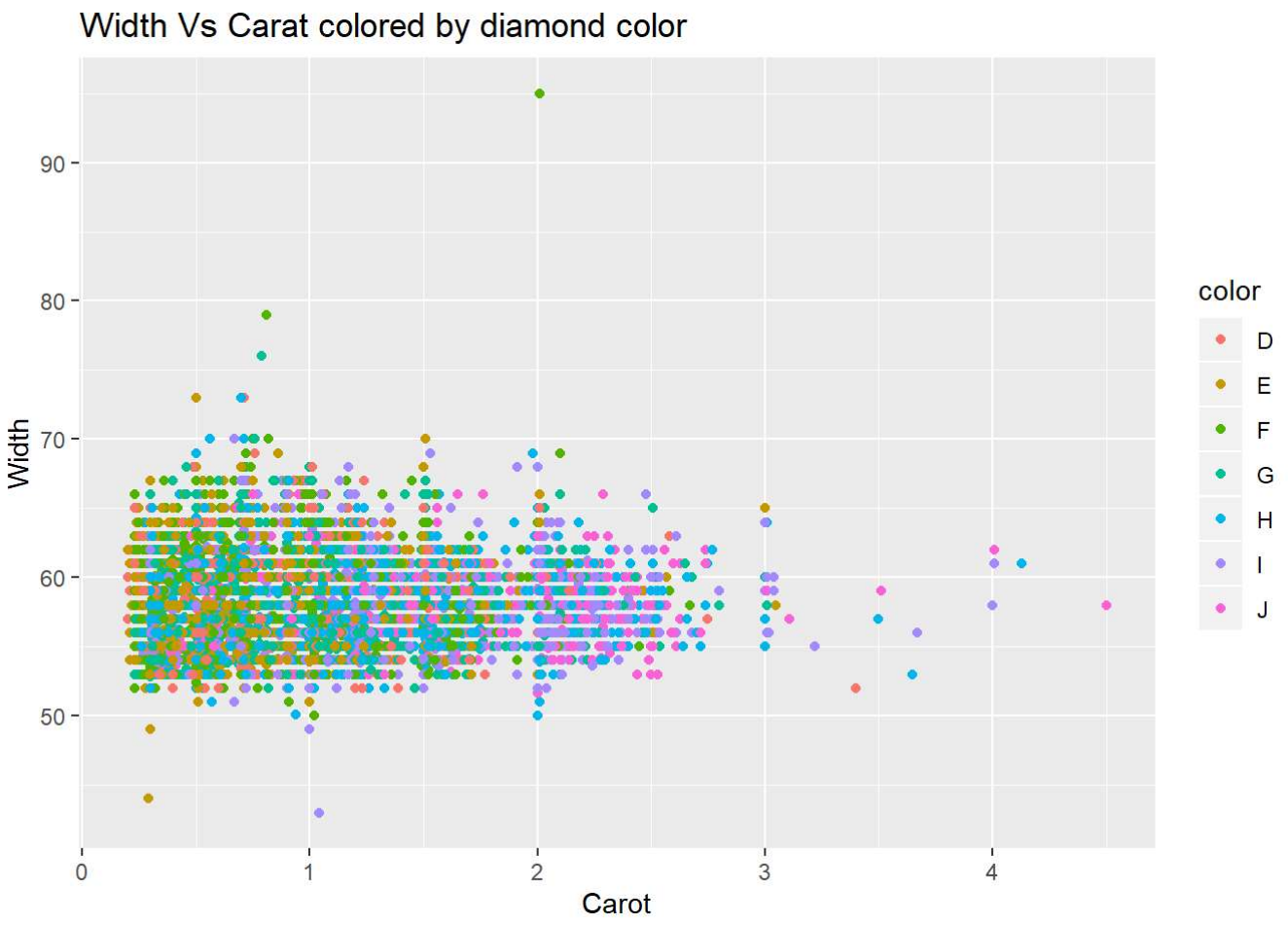

```
ggplot(diamond, aes(x=carat,y=depth)) +
  geom_point(aes(col=clarity)) +
  labs(
    x = "Carat",
    y = "Depth",
    title = "Depth Vs Carat colored by diamond clarity"
  )
```



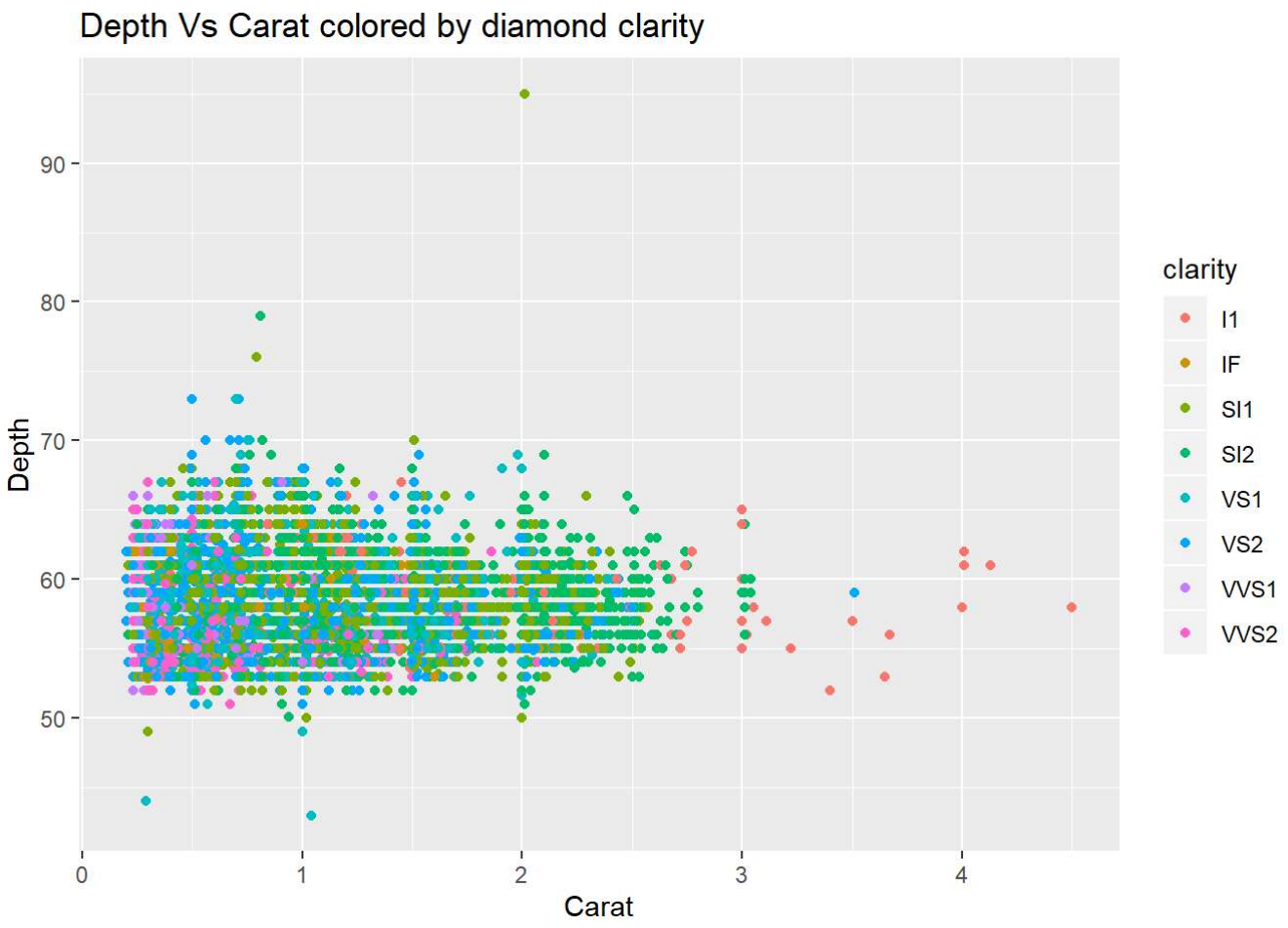

```
ggplot(diamond, aes(x=carat,y=table)) +
  geom_point(aes(col=cut)) +
  labs(
    x = "Carat",
    y = "Width",
    title = "Width Vs Carat colored by cut quality"
  )
```



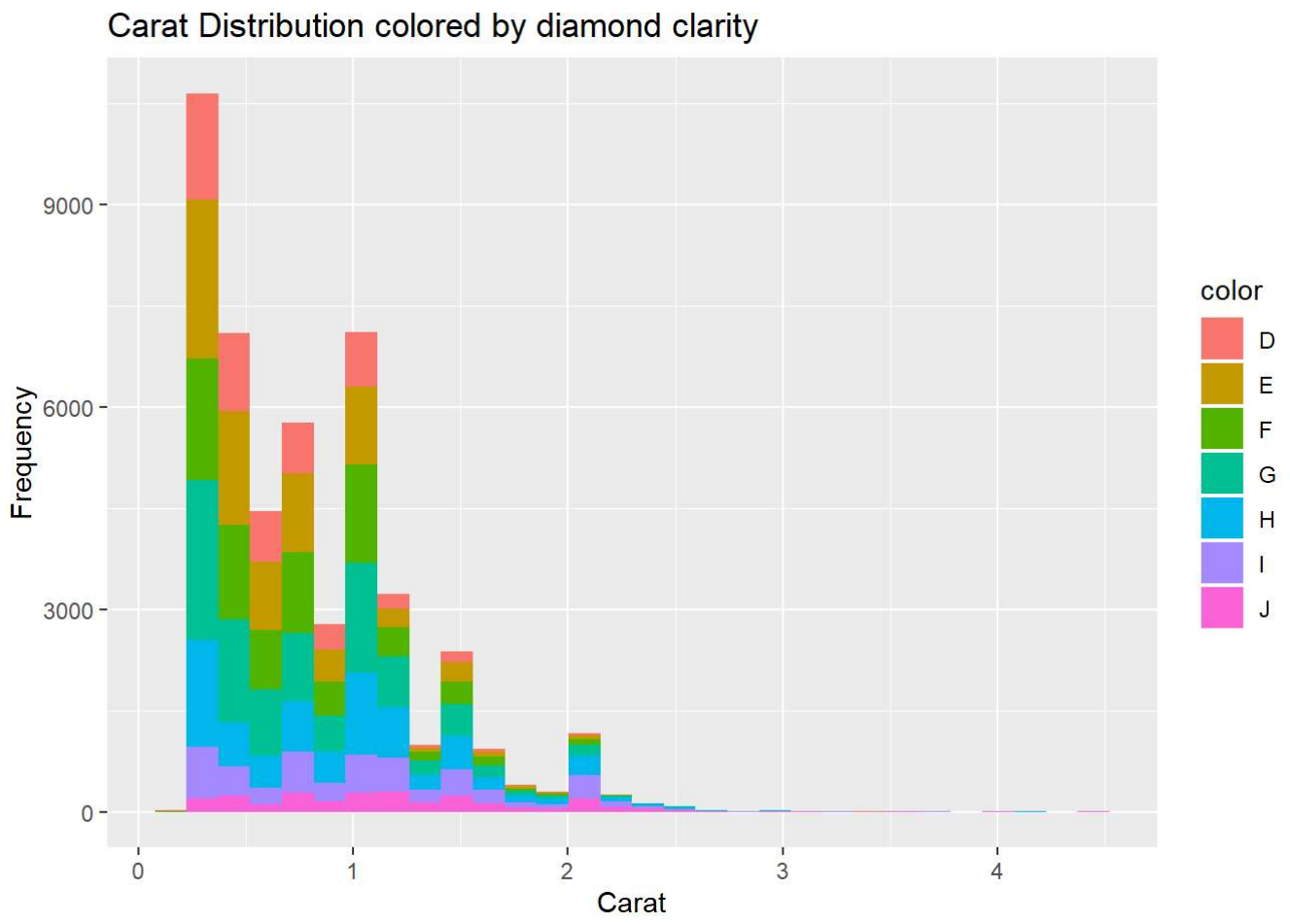

```
ggplot(diamond, aes(x=carat,y=table)) +
  geom_point(aes(col=color)) +
  labs(
    x = "Carot",
    y = "Width",
    title = "Width Vs Carat colored by diamond color"
  )
```



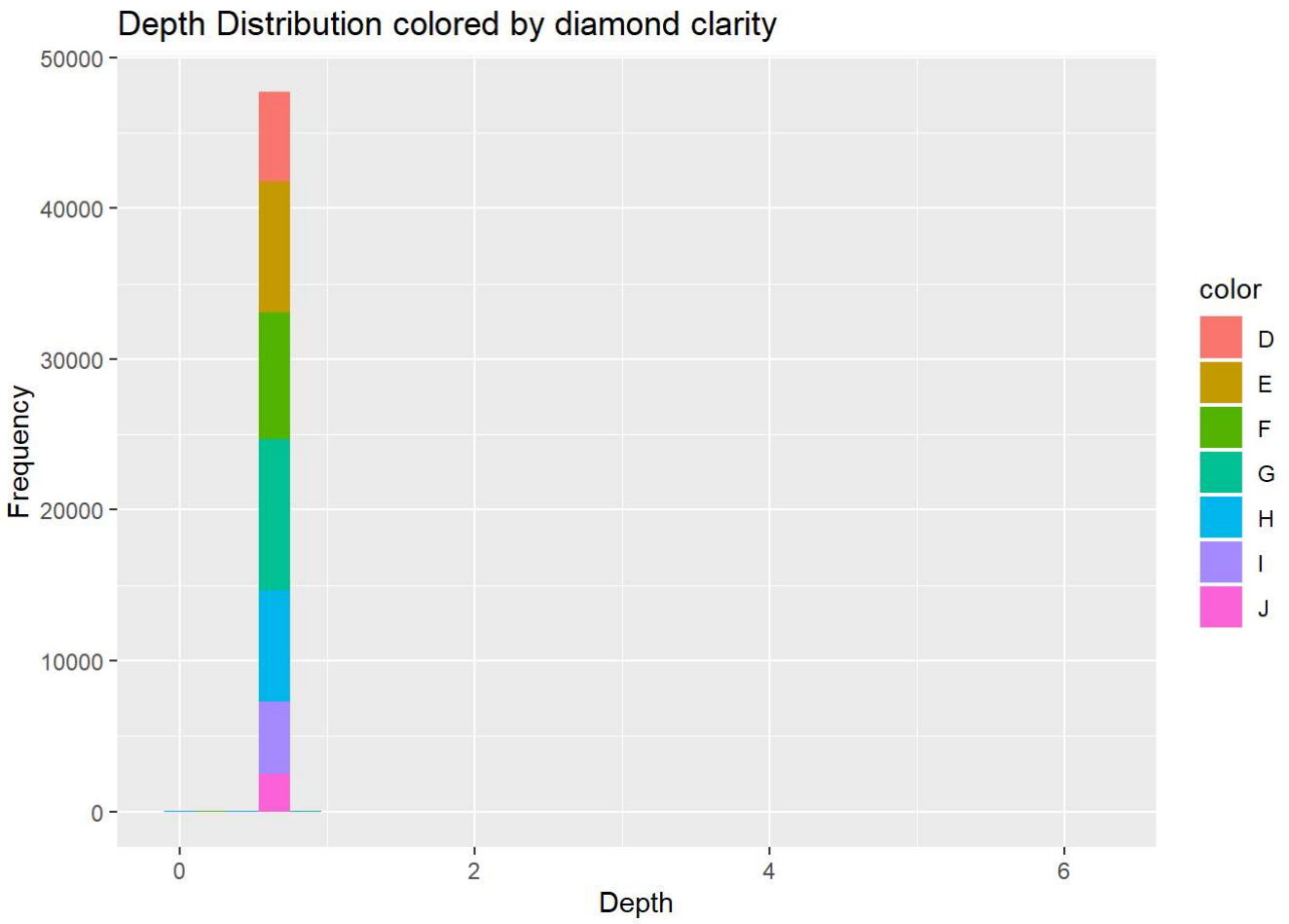

```
ggplot(diamond, aes(x=carat,y=table)) +
  geom_point(aes(col=clarity)) +
  labs(
    x = "Carat",
    y = "Depth",
    title = "Depth Vs Carat colored by diamond clarity"
  )
```



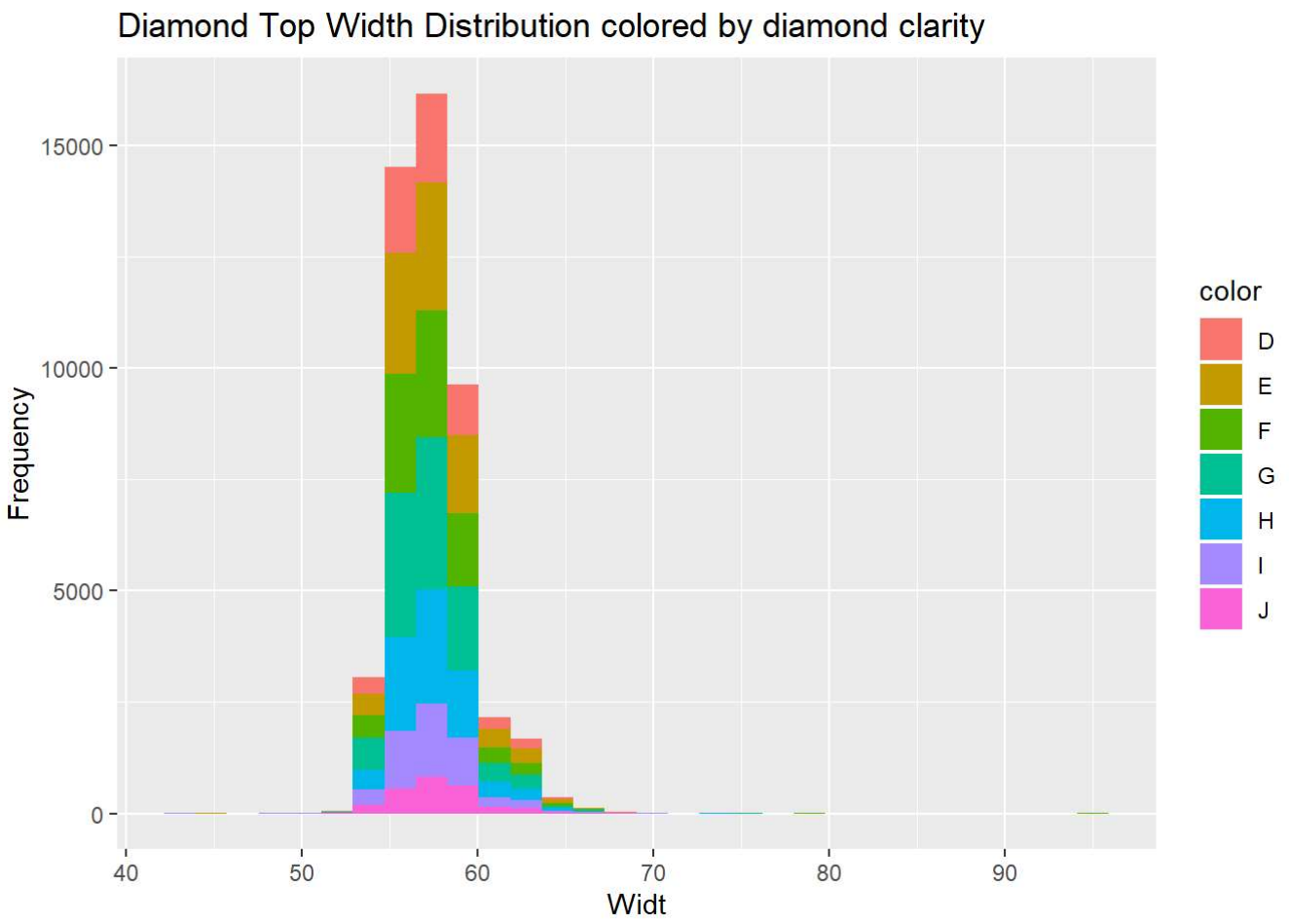

```
ggplot(diamond, aes(x=carat, fill=color)) +
  geom_histogram() +
  labs(
    x = "Carat",
    y = "Frequency",
    title = "Carat Distribution colored by diamond clarity"
  ) +
  scale_color_grey()
```



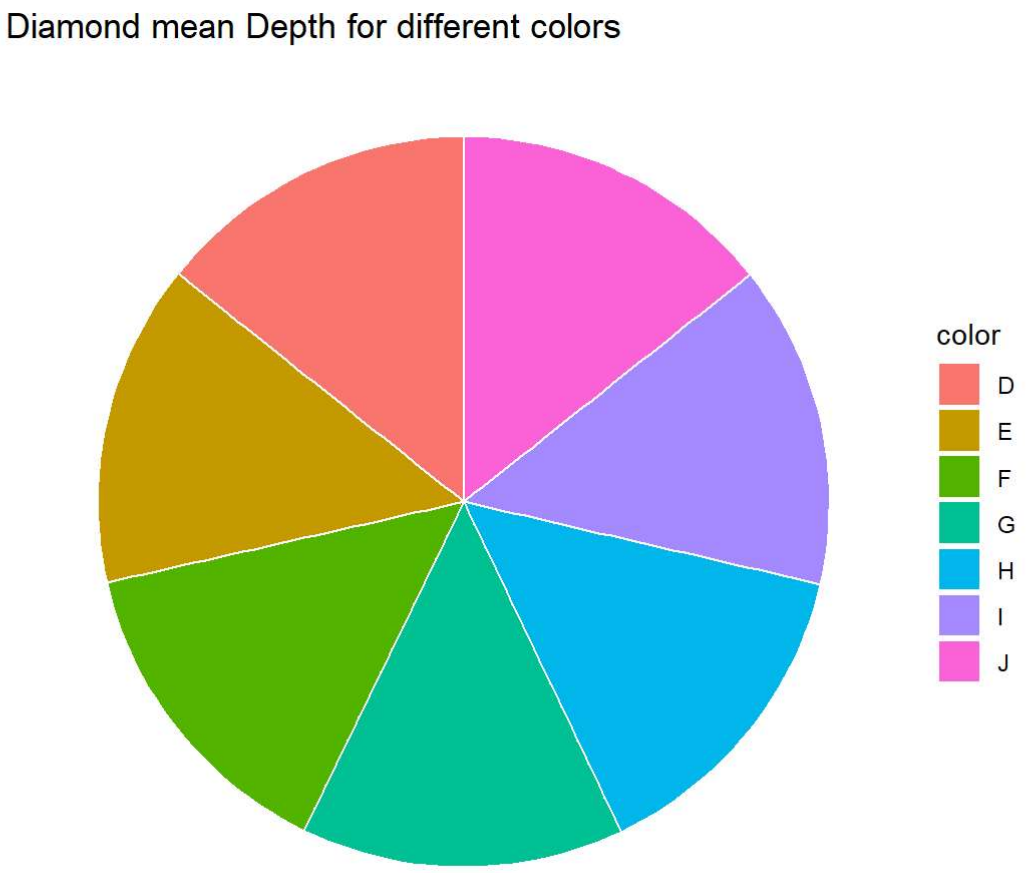

```
ggplot(diamond, aes(x=depth, fill=color)) +
  geom_histogram() +
  labs(
    x = "Depth",
    y = "Frequency",
    title = "Depth Distribution colored by diamond clarity"
  ) +
  scale_color_grey()
```



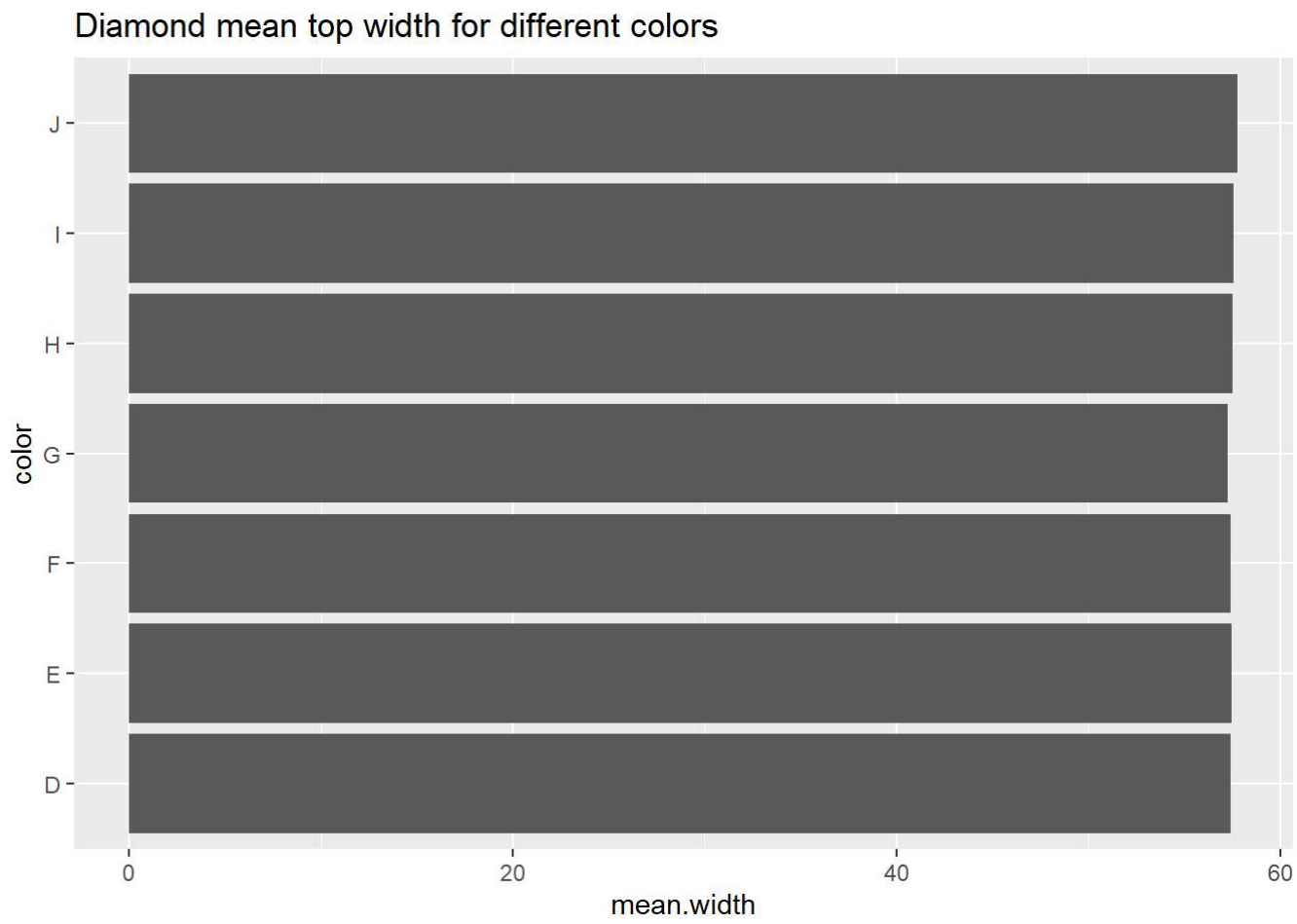

```
ggplot(diamond, aes(x=table, fill=color)) +
  geom_histogram() +
  labs(
    x = "Widt",
    y = "Frequency",
    title = "Diamond Top Width Distribution colored by diamond clarity"
  ) +
  scale_color_grey()
```




```
diamond %>%
  group_by(color) %>%
  summarise(
    mean.depth = mean(depth)
  ) %>%
  ggplot(aes(x="", y=mean.depth, fill=color)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0)+
  theme_void() +
  labs(
    title = "Diamond mean Depth for different colors"
  )
```




```
diamond %>%
  group_by(color) %>%
  summarise(
    mean.width = mean(table)
  ) %>%
  ggplot(aes(y=mean.width, x=color)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Diamond mean top width for different colors"
  ) +
  coord_flip()
```



Task 3

Part (a)

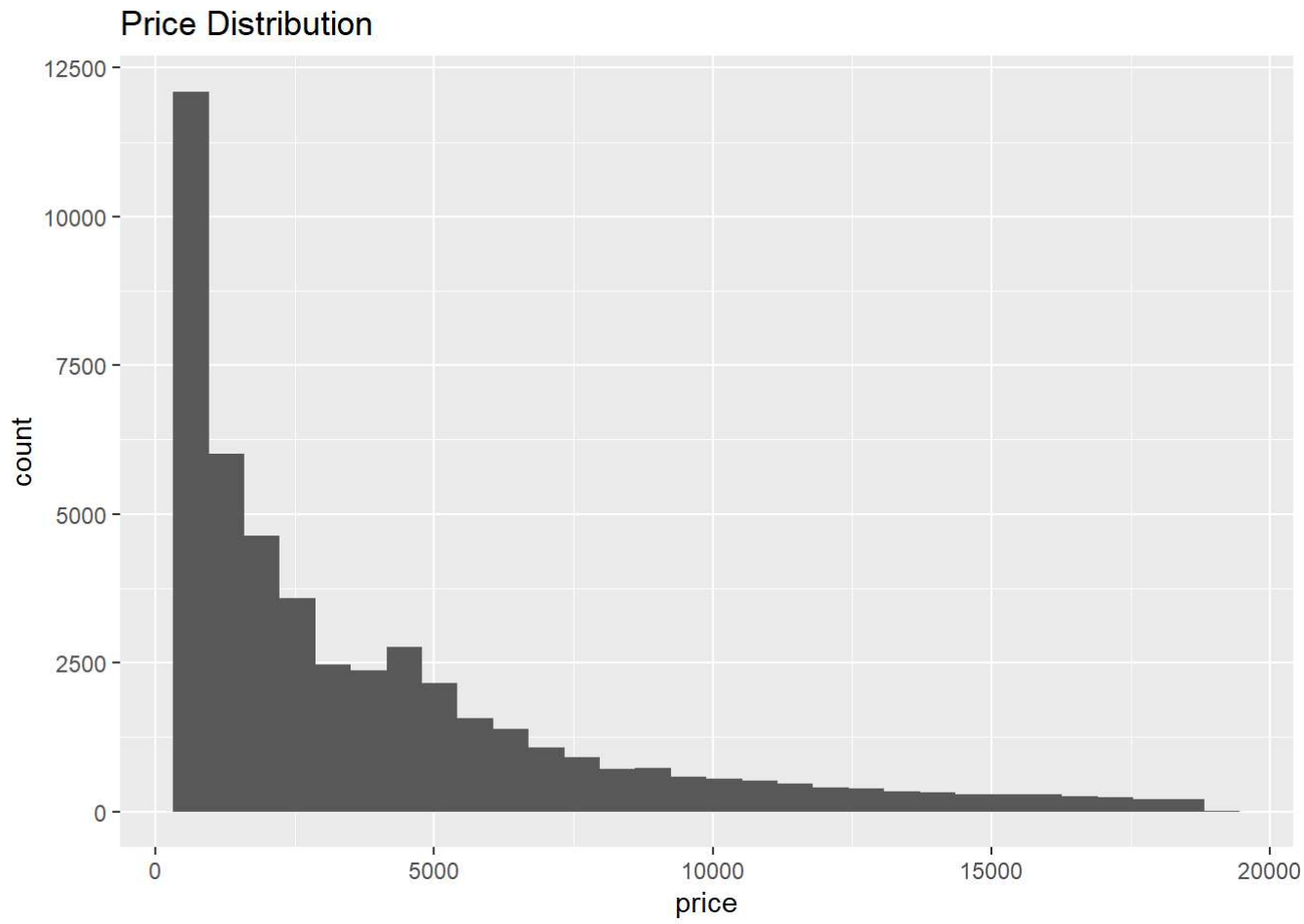
The following results shows the summary statistics for the `price` variable

```
summary(diamond$price)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    326     948    2401    3938    5342   18823 
```

Now plotting the histogram for `price` variable

```
diamond %>%
  ggplot(aes(price)) +
  geom_histogram() +
  labs(
    title = "Price Distribution"
  )
```



From the above histrogram, we can observe that `price` variable is positively skewed.

Part (b)

```
price.group <- cut(diamond$price, 3, include.lowest=TRUE, labels=c("Low", "Med", "High"))
table(price.group)
```

```
price.group
  Low   Med  High 
38587  6528  2670 
```

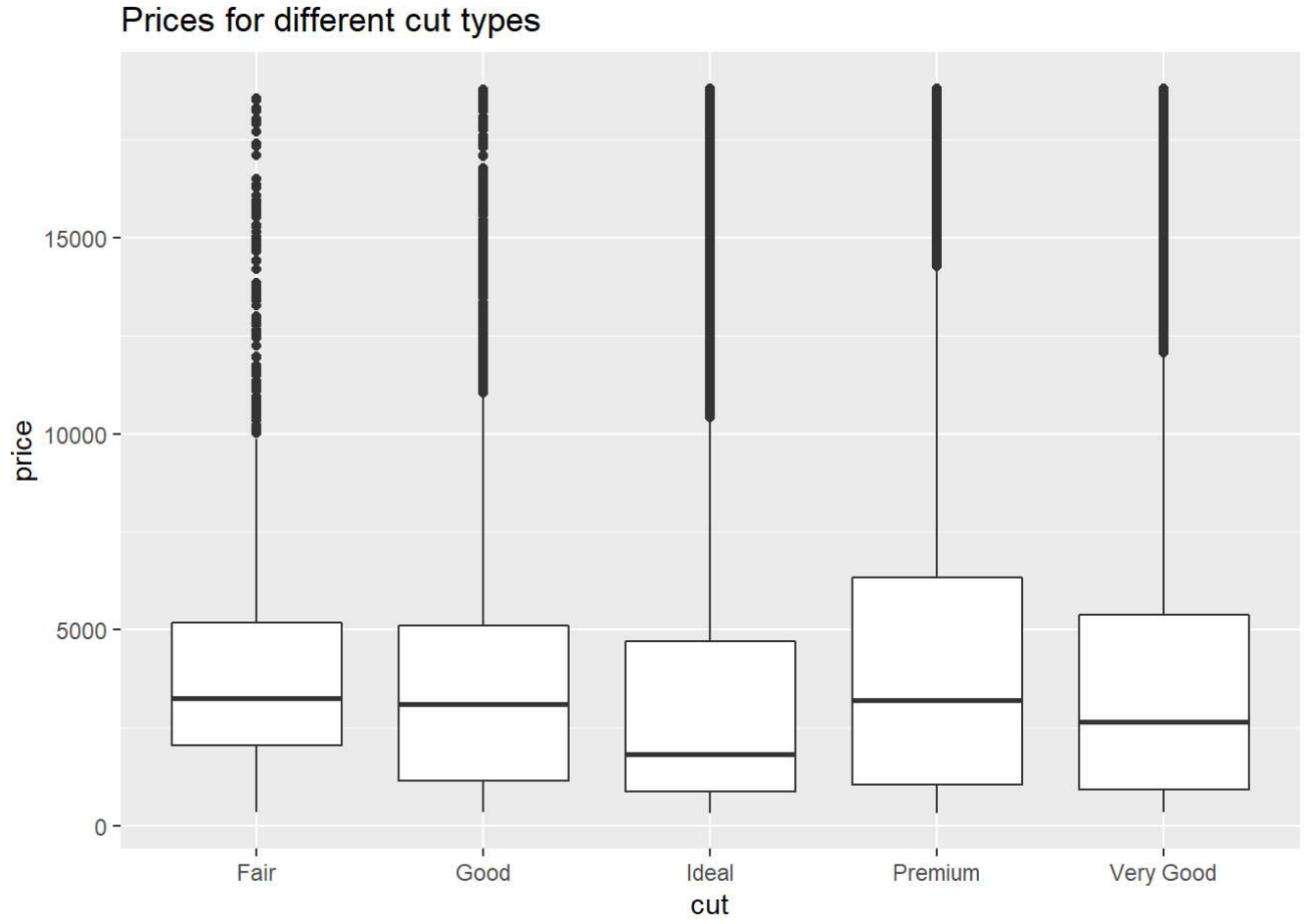
The following table shows the mea value of different numerical attributes in the dataset in different price groups

```
diamond$price.group <- price.group
t(diamond %>%
    group_by(price.group) %>%
    summarise_all(mean))
```

```
            [,1]        [,2]        [,3]
price.group "Low"       "Med"       "High"
carat       "0.6277124" "1.3599433" "1.8867978"
cut         NA          NA          NA
color       NA          NA          NA
clarity     NA          NA          NA
depth       "0.6176455" "0.6169428" "0.6157091"
table       "57.38333"  "57.71964"  "57.99157"
price       " 2290.802" " 8975.875" "15417.328"
x           "5.356730"  "7.068978"  "7.893491"
y           "5.361162"  "7.064868"  "7.883670"
z           "3.309592"  "4.359131"  "4.855292"
```

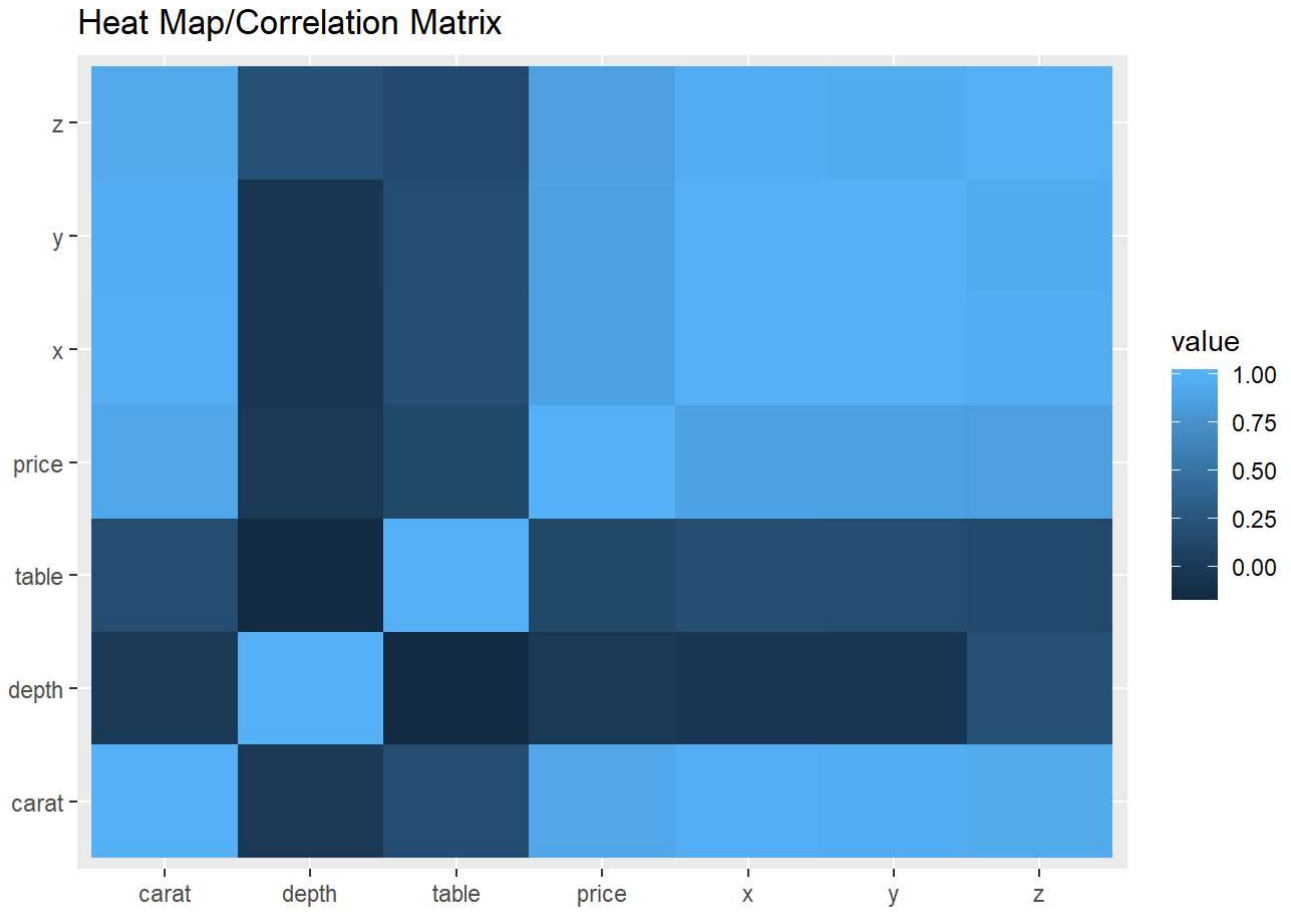
Part (c)

```
diamond %>%
  ggplot(aes(x=cut,y=price)) +
  geom_boxplot() +
  labs(
    title = "Prices for different cut types"
  )
```



Part (d)

```
nums <- unlist(lapply(diamond, is.numeric))
diamond.nums <- diamond[,nums]
cormat <- round(cor(diamond.nums),2)
melted_cormat <- melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  labs(
    x = "",
    y = "",
    title = "Heat Map/Correlation Matrix"
  )
```



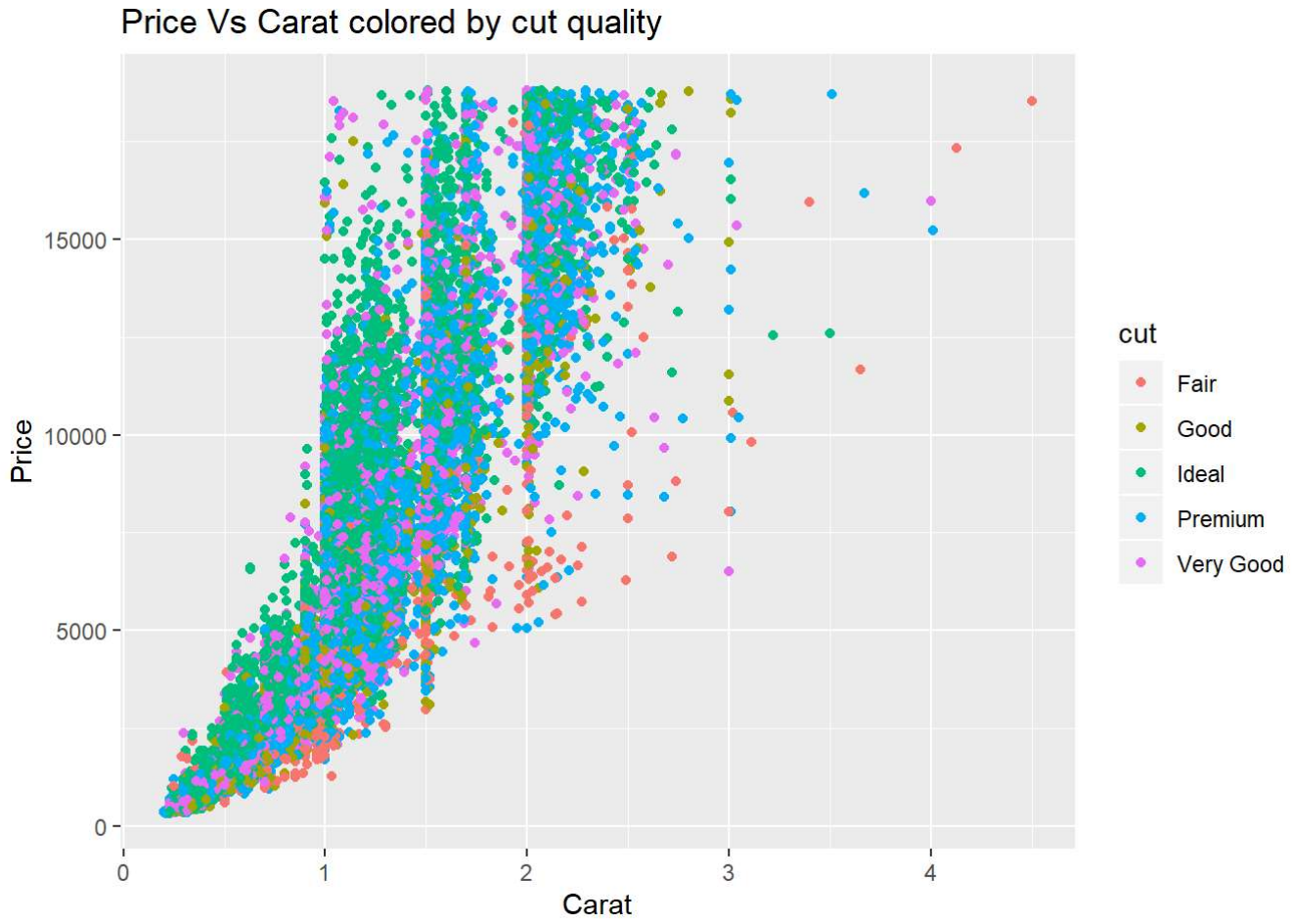
From the above plot, we can see that `carat` `x` and `y` are 3 most correlated variables with price.

Task 4

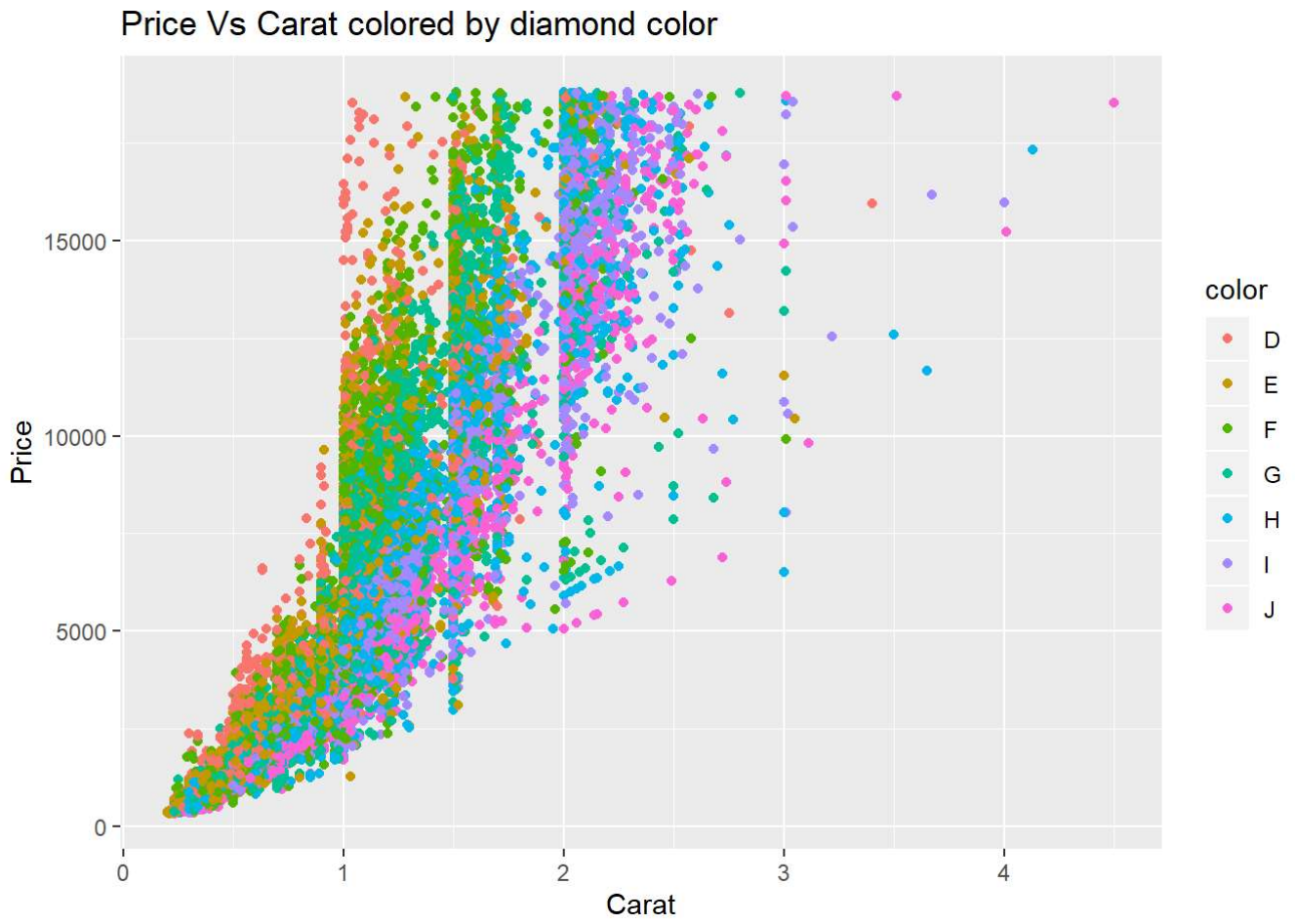
```
cut <- diamond$cut
clarity <- diamond$clarity
table(cut, clarity)
```

```
          clarity
cut           I1   IF  SI1  SI2  VS1  VS2 VVS1 VVS2
  Fair       187    8  353  400  145  241   15   59
  Good        88   68 1396  954  577  855  162  248
  Ideal      127 1068 3779 2282 3197 4518 1799 2307
  Premium    179  207 3175 2598 1752 3007  555  765
  Very Good   73  242 2881 1873 1548 2292  700 1105
```

```
ggplot(diamond, aes(x=carat,y=price)) +
  geom_point(aes(col=cut)) +
  labs(
    x = "Carat",
    y = "Price",
    title = "Price Vs Carat colored by cut quality"
  )
```



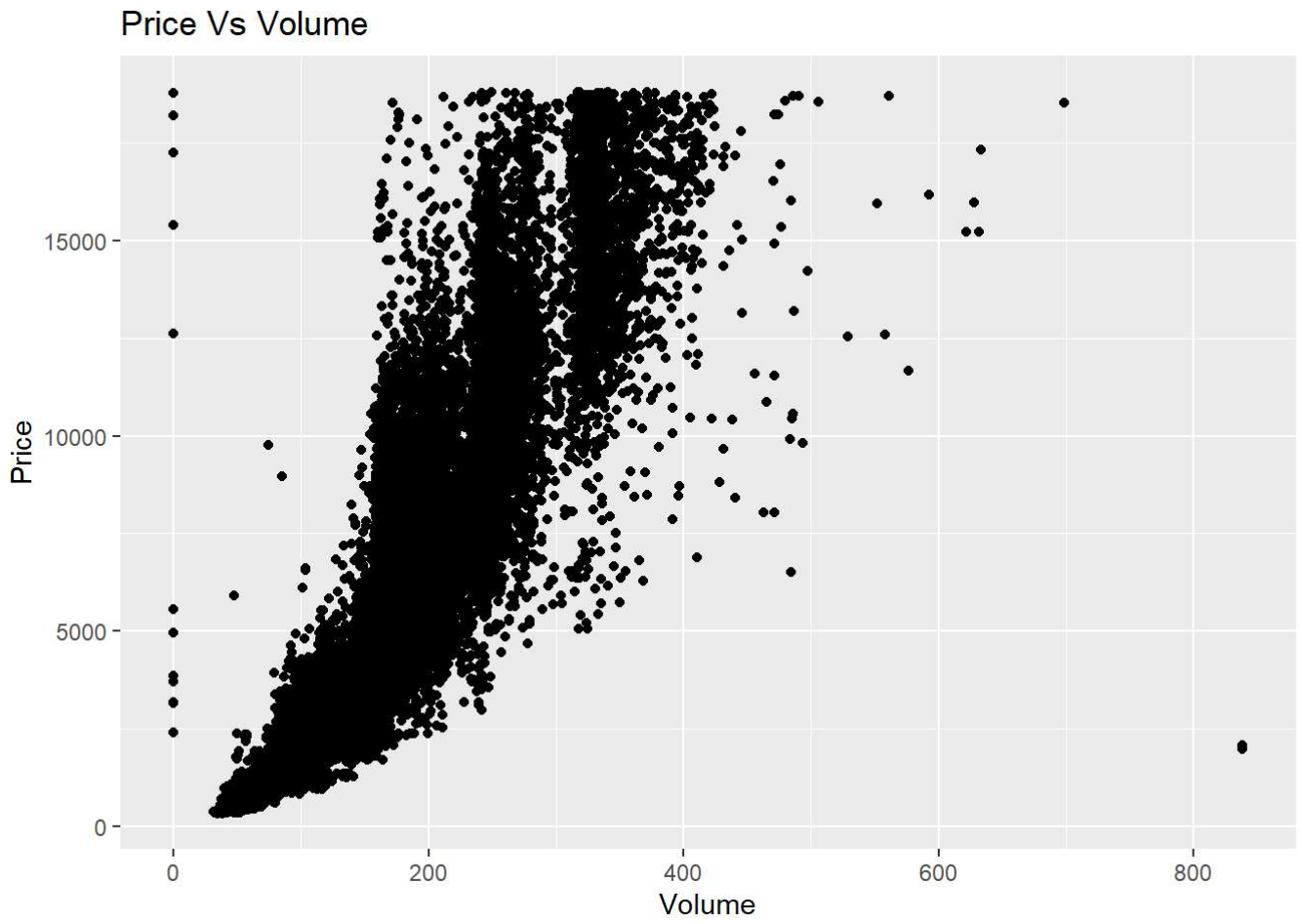

```
ggplot(diamond, aes(x=carat,y=price)) +
  geom_point(aes(col=color)) +
  labs(
    x = "Carat",
    y = "Price",
    title = "Price Vs Carat colored by diamond color"
  )
```



Task 5

Part (a)

```
diamond$volume <- diamond$x*diamond$y*diamond$z
ggplot(diamond, aes(x=volume,y=price)) +
  geom_point() +
  labs(
    x = "Volume",
    y = "Price",
    title = "Price Vs Volume"
  )
```



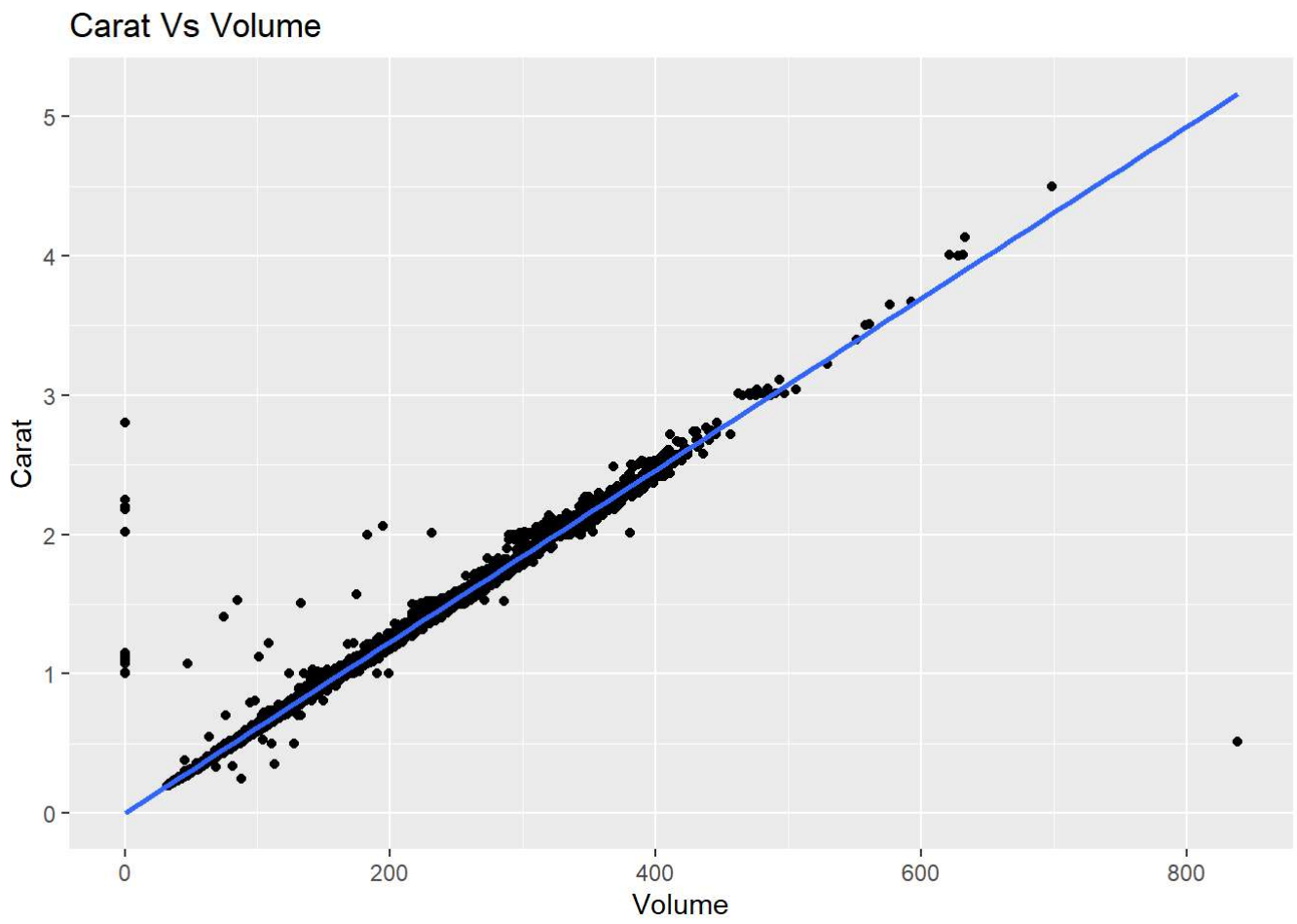
Part (b)

```
cor(diamond$carat, diamond$volume)
```

```
[1] 0.9952694
```

From the above correlatio cefficient, we can tell that `volume` and `carat` are highly correlated.

```
ggplot(diamond, aes(x=volume,y=carat)) +
  geom_point() +
  labs(
    x = "Volume",
    y = "Carat",
    title = "Carat Vs Volume"
  ) +
  geom_smooth(method='lm')
```



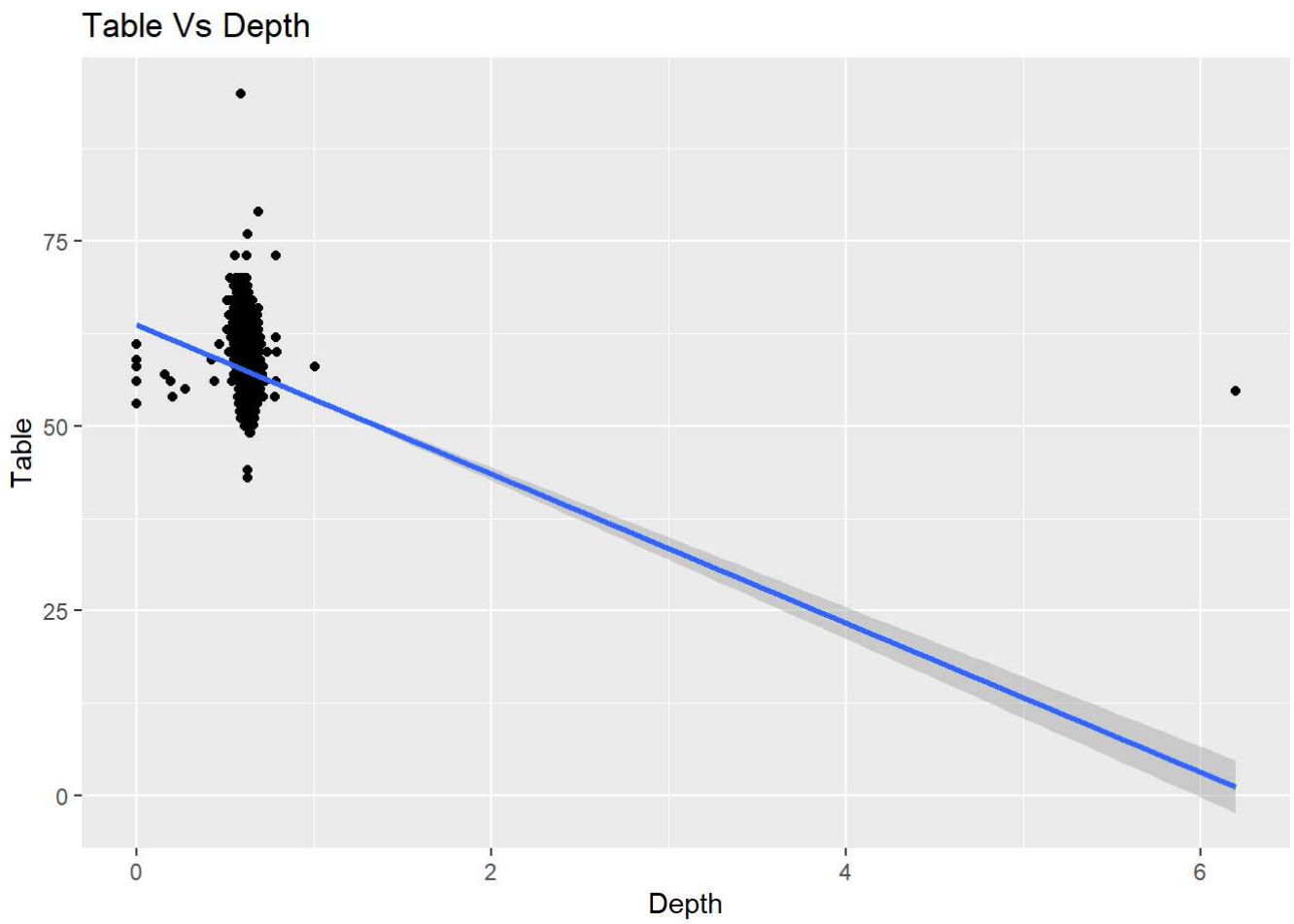
Part (c)

```
cor(diamond$table, diamond$depth)
```

```
[1] -0.1401277
```

From the above correlatio cefficient, we can tell that `table` and `depth` are not strongly correlated and the correlation is negative.

```
ggplot(diamond, aes(x=depth,y=table)) +
  geom_point() +
  labs(
    x = "Depth",
    y = "Table",
    title = "Table Vs Depth"
  ) +
  geom_smooth(method='lm')
```



Part (d)

```
cormat
```

```
      carat depth table price     x     y    z
carat  1.00  0.00  0.18  0.92  0.98  0.97 0.95
depth  0.00  1.00 -0.14 -0.01 -0.02 -0.03 0.22
table  0.18 -0.14  1.00  0.13  0.20  0.19 0.15
price  0.92 -0.01  0.13  1.00  0.89  0.88 0.86
x      0.98 -0.02  0.20  0.89  1.00  0.99 0.97
y      0.97 -0.03  0.19  0.88  0.99  1.00 0.96
z      0.95  0.22  0.15  0.86  0.97  0.96 1.00
```