CSE – 3020

Data Visualization

Lab DA – 3

Name	**:**	**Anish Desai**
Reg. No.	**:**	**20BCE0461**
Slot	**:**	**L39 + L40**
Guided by	**:**	**Prof. Jyotismita Chaki**

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Question – 1 :

Analyze data in five different ways using LDA. Properly interpret every visualization.

Code :

```
#Q1
# Linear Discriminant Analysis LDA
# Load Required Libraries
library(MASS)
library(ggplot2)
library(tidyverse)
# Attach diamonds dataset to make it easy to work
attach(diamonds)
# View the dataset
View(diamonds)
#View structure of dataset
str(diamonds)
#Create a copy of the dataset
diamonds_data <- diamonds
# Scale the values of numeric columns which are
# to be used as predictor variables
diamonds_data[c(5,8,9,10)] <- scale(diamonds_data
                                     [c(5,8,9,10)])

# Find mean of each predictor variable
apply(diamonds_data[c(5,8,9,10)], 2, mean)

#Find standard deviation of each predictor variable
apply(diamonds_data[c(5,8,9,10)], 2, sd)
```

```
# Use 70% of dataset as training dataset and
# remaining 30% as testing set
sample <- sample(c(TRUE, FALSE),
                 nrow(diamonds_data),
                 replace = TRUE,
                 prob = c(0.7, 0.3))

train <- diamonds_data[sample, ]
test  <- diamonds_data[!sample, ]

# Q1.1
# Training the model
# Fit LDA model using training dataset
model <- lda(cut ~., data = train)
# View model output
model
predicted <- predict(model, test)
# View predicted class for first six observations
                                        in test set

head(predicted $class)
# View posterior probabilities for first six
                     observations in test set

head(predicted $ posterior)
```

```
# View linear discriminant for first six
                    observations in test set
head (predicted $ x)

# predicted $ class is factor data type which
# makes it incompatible, hence convert to ord. factor
predicted $ class <- as. ordered (predicted $ class)

# Find accuracy of model
mean( predicted $ class == test $ cut)
# Define and Gather data to plot
lda_plot <- cbind (train, predict(model) $ x)

# Create plot
ggplot (lda_plot, aes(LD1, LD2)) +
     geom_point (aes (color = cut))

# Q1.2
model <- lda (color ~. , data = train)
model
predicted <- predict (model, test)
head(predicted $ class)
head(predicted $ posterior)
head (predicted $ x)
```

```
predicted $ class <- as.ordered(predicted $ class)
mean(predicted$class == test $ color)
lda_plot <- cbind(train, predict(model) $x)
ggplot(lda_plot, aes(LD1, LD2)) +
      geom_point(aes(color = color))

# Q1.3
model <- lda(clarity~., data=train)
model
predicted <- predict(model, test)
head(predicted $ class)
head(predicted $ posterior)
head(predicted $ x)
predicted $ class <- as.ordered(predicted $ class)
mean(predicted $ class == test $ clarity)
lda_plot <- cbind(train, predict(model) $x)
ggplot(lda_plot, aes(LD1, LD2)) +
      geom_point(aes(color = clarity))
```

```r
#Q1.4
model <- lda (carat ~., data = train)
model
predicted <- predict (model, test)
head (predicted $ class)
head (predicted $ posterior)
head (predicted $ x)
predicted $class <- as.ordered (predicted $ class)
mean ( predicted $ class == test $ carat)
lda_plot <- cbind (train, predict(model) $ x)
ggplot (lda_plot, aes (LD1, LD2)) +
        geom_point (aes (color = carat))
```

```r
#Q1.5
model <- lda(price ~., data=train)
model
predicted <- predict(model, test)
head (predicted $ class)
head (predicted $ posterior)
head (predicted $ x)
predicted $ class <- as.ordered (predicted $ class)
mean( predicted $ class == test $ price)
lda_plot <- cbind ( train, predict (model) $ x)
ggplot (lda_plot , aes (LD1, LD2)) +
        geom_point (aes (color = price))
```

Output :

```
> 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 ...
```

(i) Structure of dataset

```
> #Find mean of each predictor variable
> apply(diamonds_data[c(5,8,9,10)], 2, mean)
        depth             x             y             z 
 9.722488e-16  2.451758e-16 -6.542419e-17 -2.634540e-16 
> #Find standard deviation of each predictor variable
> apply(diamonds_data[c(5,8,9,10)], 2, sd)
depth     x     y     z 
    1     1     1     1 
```

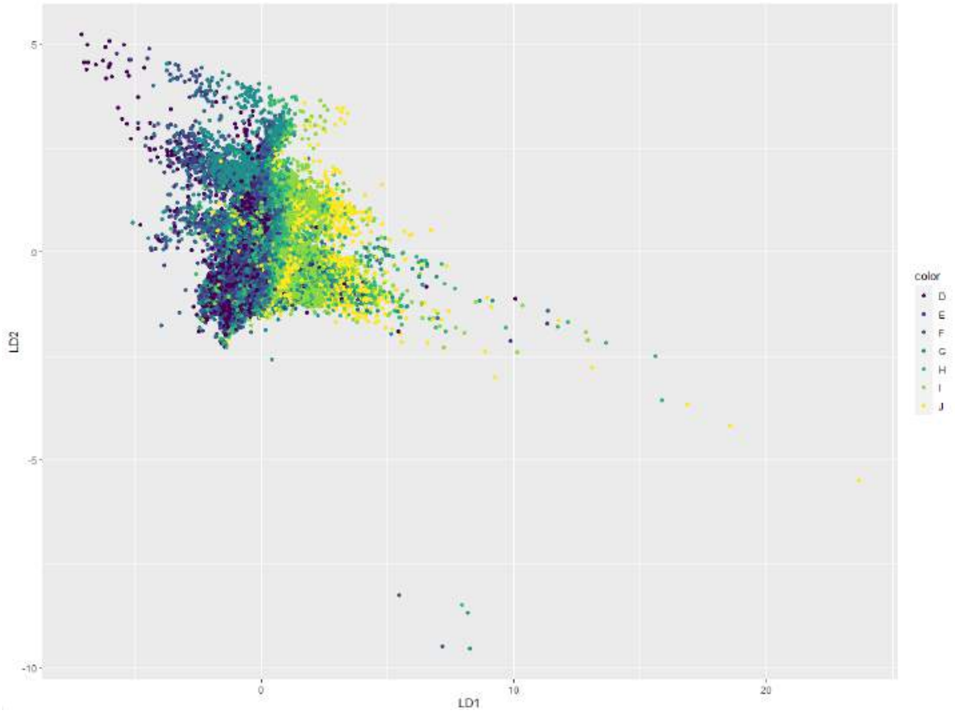
(ii) Obtaining Mean and SD of predictor variables : depth, x, y, z.

```
Prior probabilities of groups:
      Fair       Good  Very Good    Premium      Ideal 
0.03002715 0.09138583 0.22445829 0.25419262 0.39993611 


Proportion of trace:
   LD1    LD2    LD3    LD4 
0.7760 0.1768 0.0424 0.0048 

> #View predicted class for first six observations in test set
> head(predicted$class)
[1] Ideal     Very Good Fair      Ideal     Premium   Ideal    
Levels: Fair Good Very Good Premium Ideal
> #View posterior probabilities for first six observations in test set
> head(predicted$posterior)
          Fair        Good  Very Good    Premium       Ideal
1 9.681844e-06 0.010314095 0.11786767 0.06789229 0.803916269
2 2.596451e-03 0.212958795 0.34617020 0.29004530 0.148229253
3 4.801409e-01 0.357825601 0.07937756 0.08067949 0.001976476
4 3.705643e-06 0.005938395 0.07156221 0.03911594 0.883379746
5 5.361278e-04 0.124547522 0.23906339 0.61556868 0.020284286
6 3.637112e-06 0.004741506 0.06081779 0.03248178 0.901955290
> #View linear discriminant for first six observations in test set
> head(predicted$x)
         LD1         LD2        LD3       LD4
1  1.5135049 -0.23159876 -0.4918400 0.5555368
2 -0.7720671  0.05263329 -0.8810029 1.5640718
3 -3.2798698  1.12683603 -0.6414398 1.1491202
4  1.8981845 -0.10672041 -0.3718881 1.3258016
5 -1.7435120 -1.90504945 -0.9146641 0.1102139
6  1.9968377  0.01774148 -0.1971440 1.2032037
```

```
> #Find accuracy of model
> mean(predicted$class==test$cut)
[1] 0.6270307
```

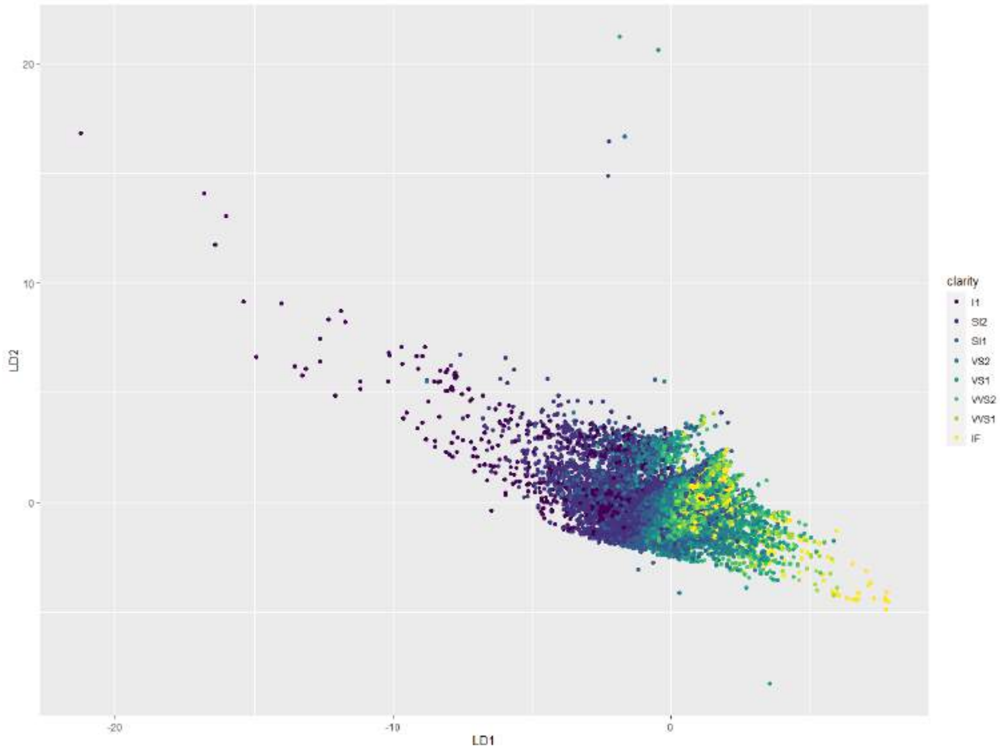
1. **Accuracy and LDA Graph for decision variable 'CUT'**

```
Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6
0.8909 0.0840 0.0129 0.0054 0.0039 0.0030

> mean(predicted$class==test$color)
[1] 0.302736
```



2. **Accuracy and LDA Graph for decision variable 'COLOR'**

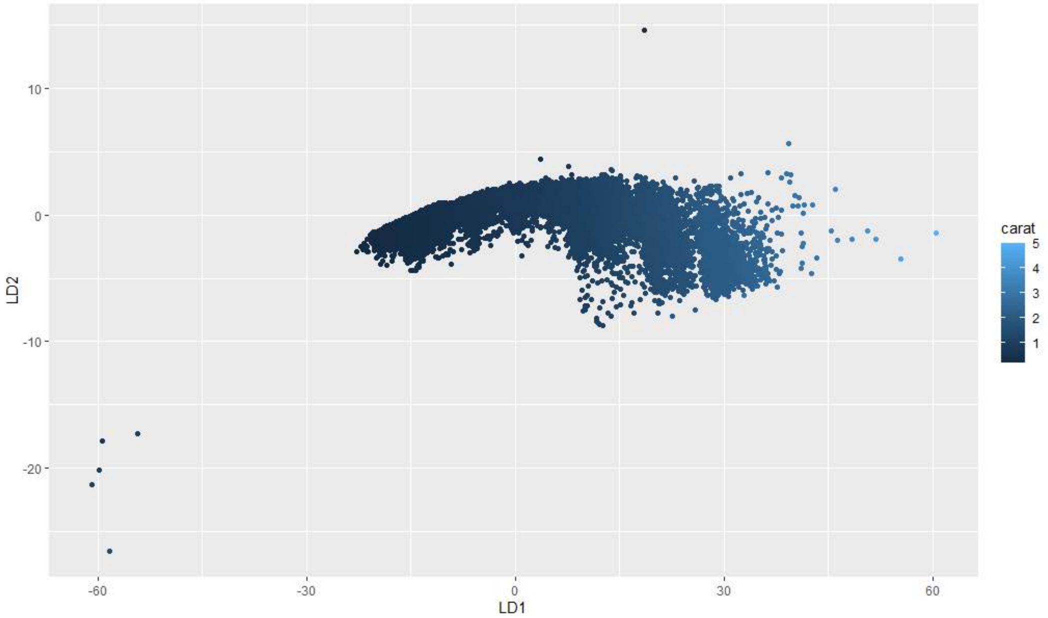
```
Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6    LD7
0.9054 0.0636 0.0129 0.0092 0.0053 0.0023 0.0014

> mean(predicted$class==test$clarity)
[1] 0.3597777
```



3. **Accuracy and LDA Graph for decision variable 'CLARITY'**

```
Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6    LD7    LD8    LD9   LD10   LD11   LD12   LD13   LD14
0.9891 0.0075 0.0007 0.0007 0.0004 0.0003 0.0002 0.0002 0.0002 0.0001 0.0001 0.0001 0.0001 0.0001
  LD15   LD16   LD17   LD18   LD19   LD20   LD21   LD22   LD23
0.0001 0.0001 0.0001 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

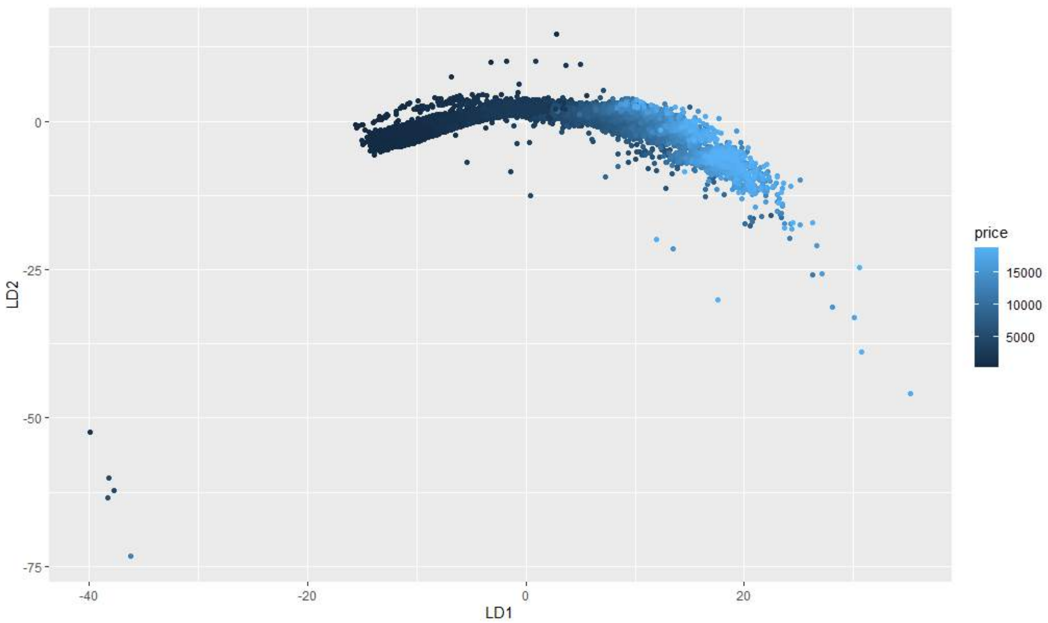
```
> mean(predicted$class==test$carat)
[1] 0.3252718
```



4. Accuracy and LDA Graph for decision variable 'CARAT'

```
Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6    LD7    LD8    LD9   LD10   LD11   LD12   LD13   LD14
0.8118 0.0637 0.0147 0.0102 0.0082 0.0071 0.0068 0.0065 0.0059 0.0058 0.0056 0.0055 0.0052 0.0052
  LD15   LD16   LD17   LD18   LD19   LD20   LD21   LD22   LD23
0.0051 0.0048 0.0047 0.0044 0.0041 0.0040 0.0038 0.0035 0.0033
```

```
> mean(predicted$class==test$price)
[1] 0.03933065
```



5. Accuracy and LDA Graph for decision variable 'PRICE'

Interpretation :

Interpretation #Q1

For the given question, the predictor variables taken are 'depth', 'x', 'y' and 'z'.

The five decision variables considered :

1. cut
2. Color
3. clarity
4. carat
5. price

For each of the decision variable, we obtain LDAs, train the models for classification and compute accuracies of the models, about which we are more concerned and is of utmost importance.

1. For CUT,

The accuracy of the model is 62.70%.

**

Another important parameter is Proportion of trace. This displays the percentage separation achieved by each LDA function.

**

For CUT, LD1 : 77.60 %

LD2 : 17.68 %

These two LDs achieve maximum separation and we can clearly discriminate the data

2. For COLOR,

Accuracy of the model is 30.27 %.

LD1 : 89.09 %.

LD2 : 8.40 %.

LD1 is almost enough for us to discriminate the data

3. For CLARITY,

Accuracy is 35.97 % whereas

LD1 : 90.54% and LD2 : 6.36 %.

4. For CARAT,

Accuracy is 32.52 %.

LD1 : 98.91 %. and LD2 : 0.75 %.

LD3 onwards even more negligible

We can infer that for the given decision variable, using the predictor variables, LD1 gives a very clear separation of data.

5. For PRICE,

Accuracy of the model is 3.93 %
which is very poor model

LD1 : 81.18 %. and LD2 : 6.37 %.

We can infer that :

1. Using the predictor variables depth, x (length), y (width) and z (depth) in mm, the variable 'cut' can be classified with the best accuracy of 62.70%. For the rest, accuracies are very less and thus a poor model.

2. The LDI for 'carat' having the highest proportion can be the best linear discriminant function to differentiate the dataset

---

Question – 2 :

Analyze data in five different ways using correlation analysis. Properly interpret every visualization.

Code :

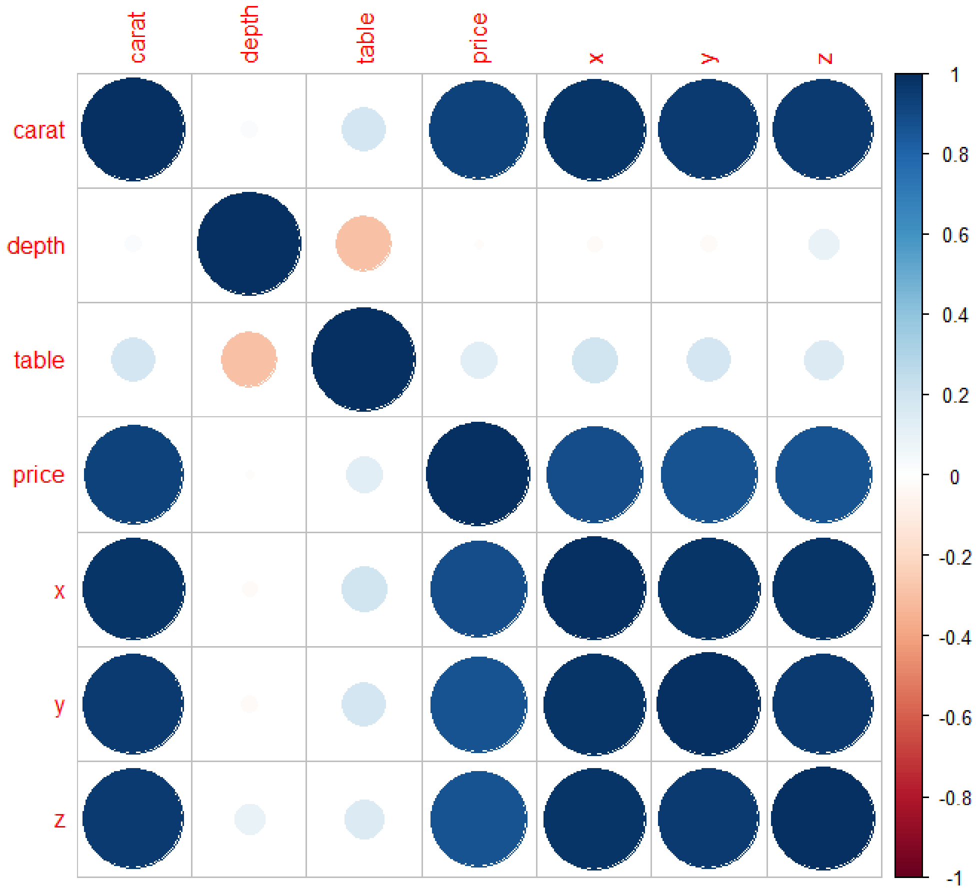
```
#Q2
# Correlation Analysis
# Load Required Libraries
library(ggplot2)
library(tidyverse)
library ("ggpubr")
diamonds_data_2 <- diamonds[c(5,1,6,7,8,9,10)]
view(diamonds_data_2)
```

CORRELATION MATRIX

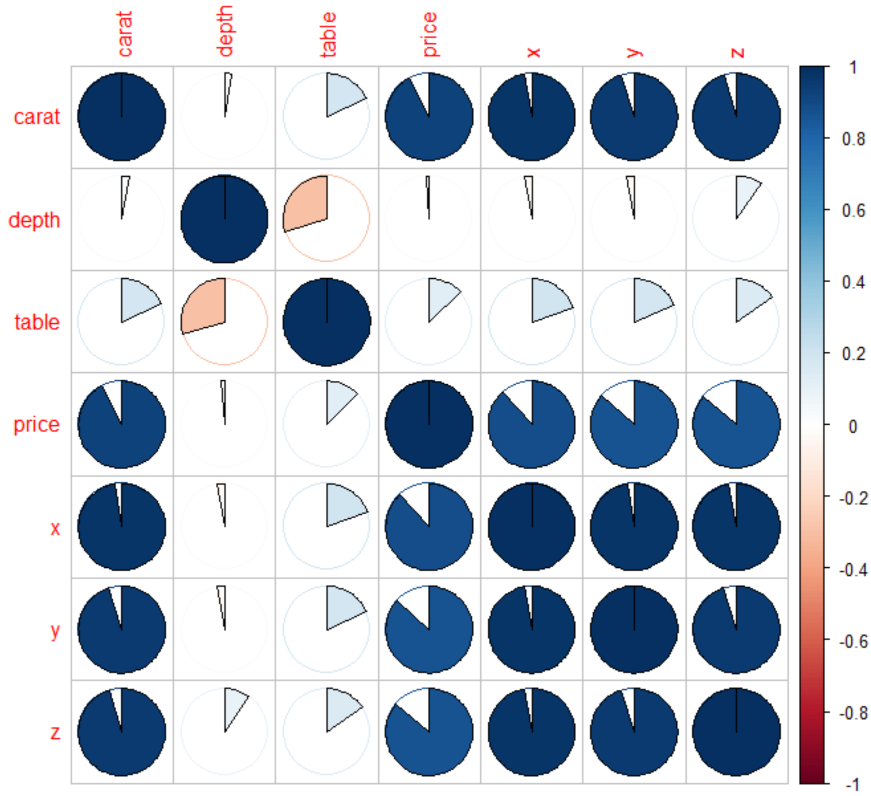
```
# correlation coefficients between possible pairs
                                 of variables

D <- cor(diamonds_data_2)
round (D, 2)

# Correlogram : Visualizing correlation matrix
library (corrplot)

#Q2.1
corrplot (D, method = "circle")

#Q2.2
corrplot (D, method = "pie")

#Q2.3
corrplot (D, method = "color")

#Q2.4
corrplot (D, method = "number")

#Q2.5
# Display chart of correlation matrix
library ("PerformanceAnalytics")
diamonds_data_3 <- diamonds[, c(1,5,6,7,8,9,10)]
chart.correlation(diamonds_data_3, histogram= TRUE, pch = 19)
```

Output :

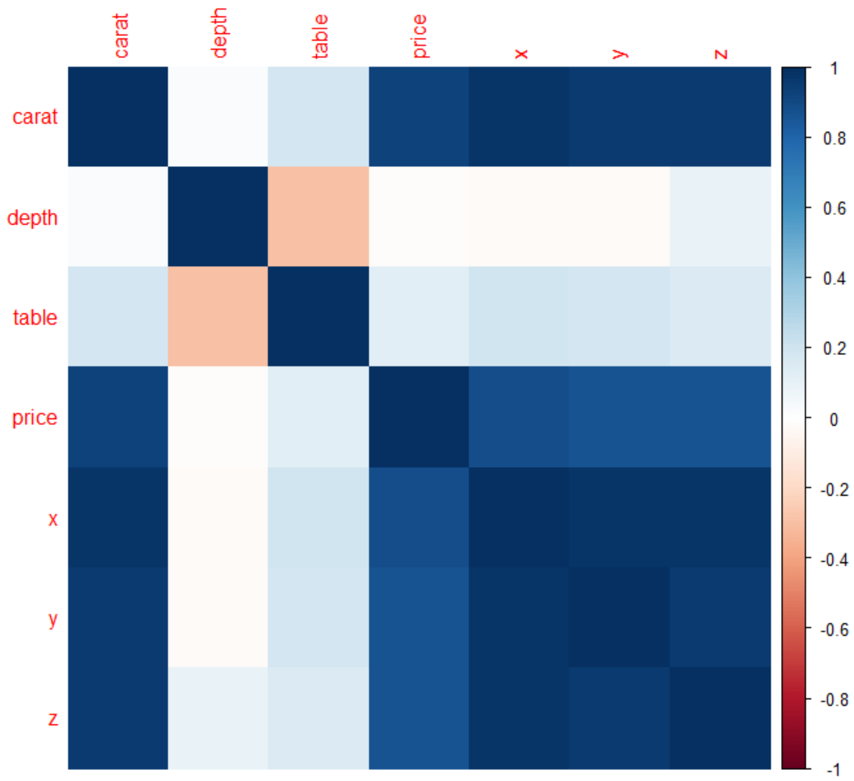
```
> #CORRELATION MATRIX
> #correlations coefficients between the possible pairs of variables
> D<-cor(diamonds_data_2)
> round(D,2)
      carat depth table price     x     y    z
carat  1.00  0.03  0.18  0.92  0.98  0.95 0.95
depth  0.03  1.00 -0.30 -0.01 -0.03 -0.03 0.09
table  0.18 -0.30  1.00  0.13  0.20  0.18 0.15
price  0.92 -0.01  0.13  1.00  0.88  0.87 0.86
x      0.98 -0.03  0.20  0.88  1.00  0.97 0.97
y      0.95 -0.03  0.18  0.87  0.97  1.00 0.95
z      0.95  0.09  0.15  0.86  0.97  0.95 1.00
```



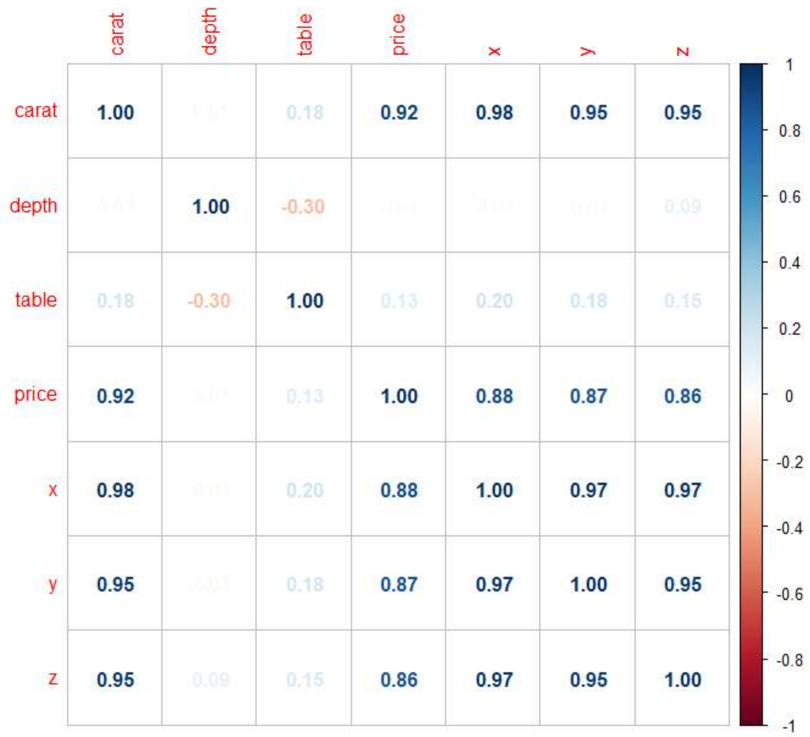
1. **Circle Plot**



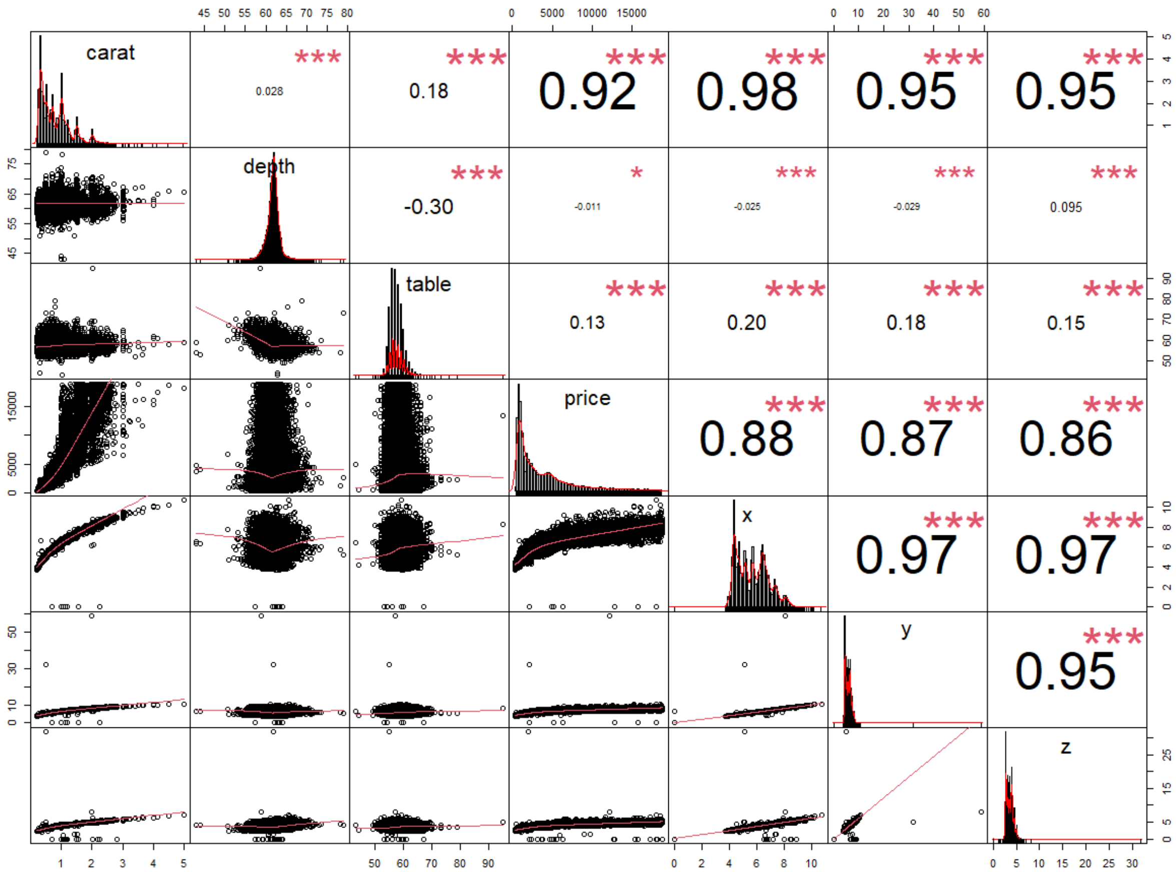
2. Pie Plot



3. Color Plot



4. Number Plot



5. Correlation Chart

Interpretation :

Interpretation #Q2

Using correlation analysis, we have obtained the magnitude and direction of correlations between various variables of the dataset.

Inference of Five different analysis :

1. Using Circle Plot,
we can notice that if the colour is nearer to the red end, it implies negative correlation and if the colour is nearer to the blue end, it implies positive correlation between the variables.
Larger the circle, larger is the magnitude of the correlation coefficients.
The diagonal shows correlation between same variable (=1), hence it must be the largest and darkest shade of blue
Other large blue circles implies the correlation coefficients are large positive numbers.
The two slight reddish small circles imply correlation coefficient is small and negative

Few blocks with no circles visible implies the correlation coefficient circle is white in colour and thus is nearly zero(≈ 0).

2. Using Pie Plot,

Nearer to red end $\rightarrow$ Negative corr.

Nearer to blue end $\rightarrow$ Positive corr.

Greater the angle of pie, greater is the magnitude

$0^\circ \rightarrow \pm 0$ $\quad$ $90^\circ \rightarrow \pm 0.25$

$180^\circ \rightarrow \pm 0.5$ $\quad$ $270^\circ \rightarrow \pm 0.75$

$360^\circ \rightarrow \pm 1$

The blocks which had not been visible in the circle plot are now visible in pie plot, which makes it a better Viz tool (Can use both the colour as well as the angle traced by pie to interpret)

3. Using color plot,
The intensity of the color decides the correlation magnitude and the sign depends upon the blue or red shade.

4. Using number plot,
it gives the correlation coefficient figures directly and is the most easiest to interpret and obtain values.

5. Using correlation chart,
The values in the blocks represent the correlation b/w variables; The symbols '*' represent various significance level

*** ⇒ p = 0   * ⇒ p = 0.01

** ⇒ p = 0.001   • ⇒ p = 0.05

' ' ⇒ p = 0.1

We can conclude that the corr coeff b/w 'carat' (of diamond) and its length 'x' is the highest (~0.98) and hence is closely related than the rest. 'x' 'y' and 'x' 'z' are also very closely related (~0.97). 'Depth' and 'price' have the least corr. coeff of about −0.011 and hence are not related at all.

--------------------------------------Thank you--