DS4043 - MIDTERM Examination

Name: Ruoxin WANG Grade: _________

Coding Part

3. In this question, we will use the dataset called *painters*. We can simply call the data

```
library(MASS)
data(painters)
```

a. Use help function to find the description of this data.

The description is: The subjective assessment, on a 0 to 20 integer scale, of 54 classical painters. The painters were assessed on four characteristics: composition, drawing, colour and expression. The data is due to the Eighteenth century art critic, de Piles.

b. Select the data such that the *Composition* score is greater or equal to 15 and make it a data frame called *HCpainters*. Print out *HCpainters*.

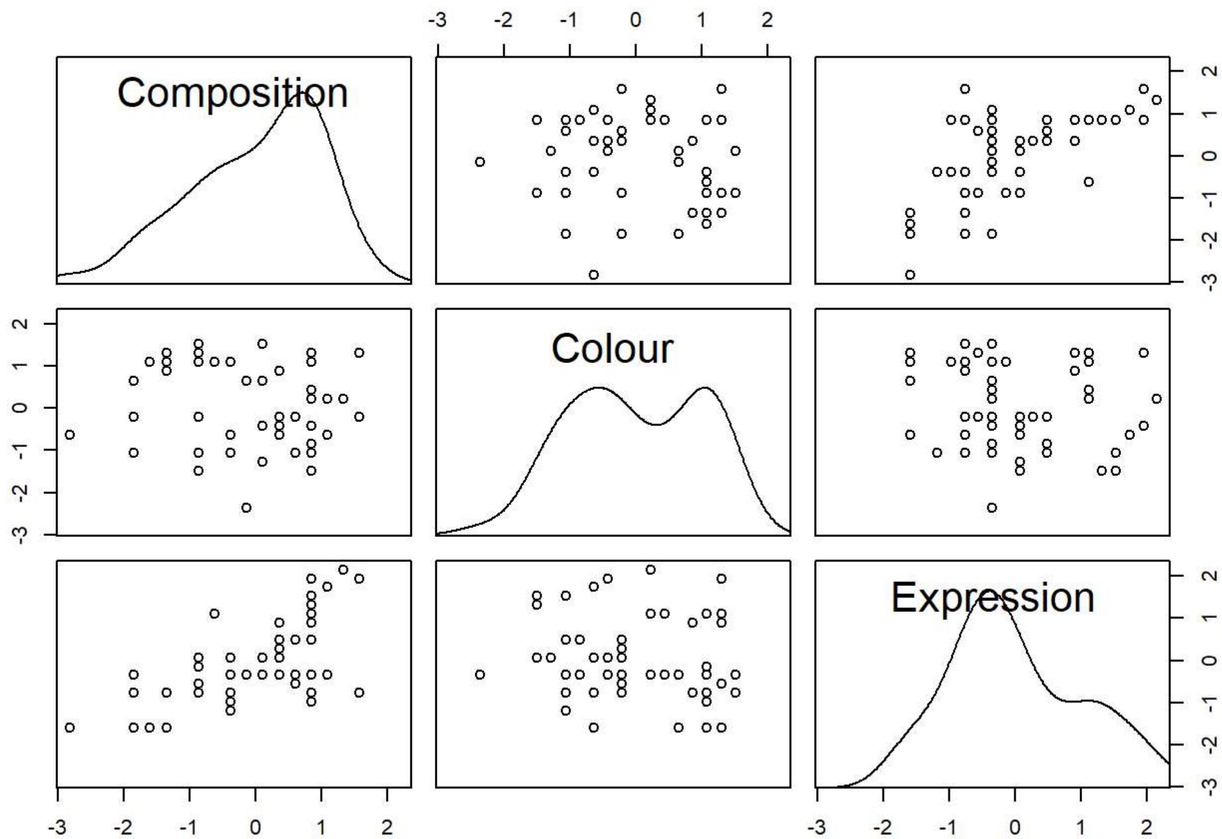
```
# Put your code here
HCpainters <- data.frame(painters[painters$Composition >= 15,])
HCpainters
```

c. Create a new column vector Called *LowColour* indicating if the *Colour* score is less than 11 and append this column to *painters*. Show the first 10 rows. (This question is not related to (b))

```
# Put your code here
LowColour <- painters$Colour < 11
painters <- data.frame(painters, LowColour)
head(painters, 10)
```

d. Obtain the scatterplot matrix of the data *painters* for the *Composition*, *Colour* and *Expression*. Which pair of variables have the strongest positive linear relation.

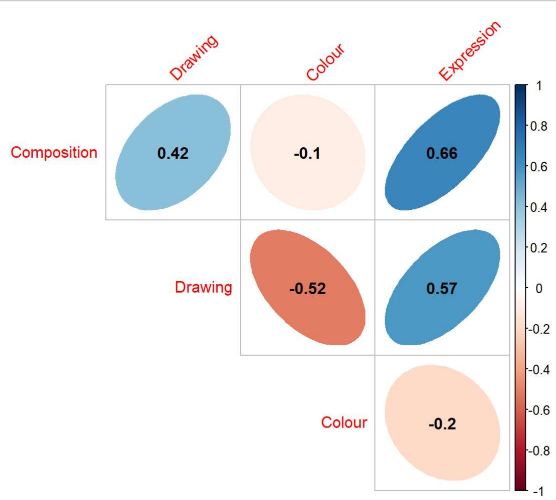
```
# Put your code here
panel.d <- function(x, ...) {
    usr <- par("usr"); on.exit(par(usr));
    par(usr = c(usr[1:2], 0, .5))
    lines(density(x))  # add density curve.
}
x <- scale(painters[, c('Composition', 'Colour', 'Expression')])
r <- range(x)
pairs(x, diag.panel = panel.d, xlim = r, ylim = r)
```



Solution (Which pair): Composition and Expression.

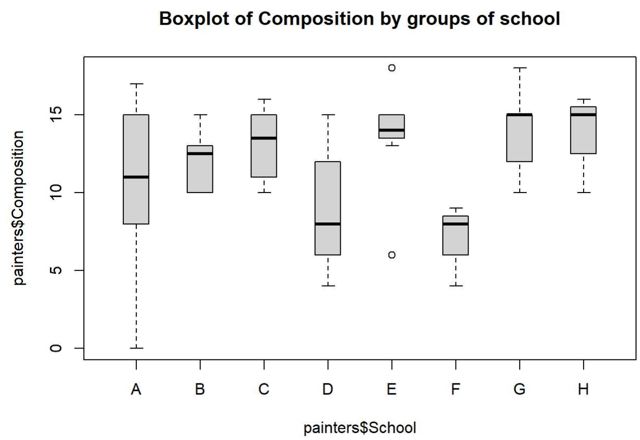
e. Find the correlation matrix of *Composition*, *Drawing*, *Colour* and *Expression*. Then get the correlation plot and make the visualization method as ellipse, the text label as red color, and use “diag=FALSE”.

```
library(corrplot)
# Put your code here
corrMat <- cor(painters[, 1:4]);
corrplot(corrMat, type = "upper", method = "ellipse", addCoef.col = "black", diag=FALSE, t
l.col="red", tl.srt=45);
```



f. Plot the Boxplot of *Composition* by groups of *School*.

```
# Put your code here
boxplot(painters$Composition ~ painters$School, main="Boxplot of Composition by groups of school", xlab="painters$School", ylab = "painters$Composition", boxwex = .4)
```

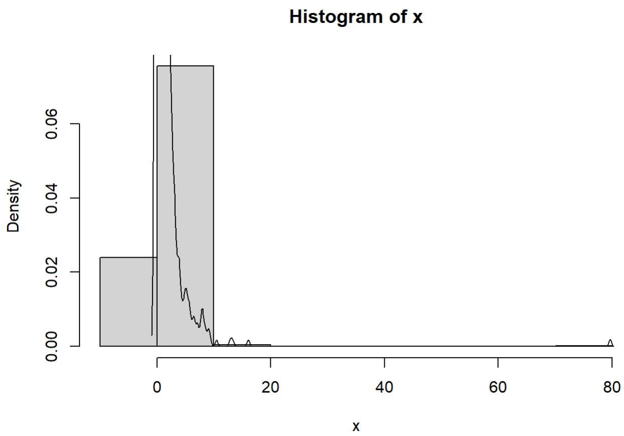


4. Consider the following distribution with density function \[f(y) = \frac{r\beta^r}{(\beta+y)^{r+1}}, \quad y \geq 0. \] Use the inverse transform method to generate 1000 random observations from this distribution with \(r = 3\) and \(\beta = 4\). Print out the first 8 observations of the generated sample. Plot the density histogram of the total sample and plot the sample density curve on the histogram.

```
set.seed(24)
# Put your code here
n <- 1000;
u <- runif(n);
x <- ((192/u)^0.25)-4;
# Print out the first 8 observations of the generated sample.
head(x, 8)
```

```
## [1]  1.06135  1.40545  0.06348  0.38586  0.12581 -0.19963  1.11845 -0.01822
```

```
# Plot the density histogram of the total sample.
hist(x, prob = TRUE)
# Plot the sample density curve on the histogram.
lines(density(x))
```



5. We want to compute a Monte Carlo estimate of $$ \int_{0}^{\pi / 6} x\cos (x) \, dx $$ using the importance sampling method. We choose three importance functions $f_1 = 1$ which is the density of Uniform(0,1), $f_2 =6/\pi$ which is the density of Uniform(0, $\pi/6$) and $f_3 = 2\cos (x), 0 <x<\pi/6$. Show your estimation results and their corresponding standard errors using 10000 replications. **(Hint: the code of the inverse of $sin()$ is $asin()$)**

```r
set.seed(47)
# Put your code here
m <- 10000
theta.hat <- se <- numeric(3)
g <- function(x){
  x*cos(x)*(x>0)*(x<pi/6)
}

# here we use f1.
x <- runif(m)
i <- c(which(x > pi/6), which(x < 0)) #indicator function.
x[i] <- 2 #to catch overflow errors in g(x)
fg <- g(x)
theta.hat[1] <- mean(fg)
se[1] <- sd(fg)

# here we use f2.
x <- runif(m, 0, pi/6)
fg <- g(x) / (6/pi)
theta.hat[2] <- mean(fg)
se[2] <- sd(fg)

# here we use f3.
u <- runif(m) #inverse transform method
x <- acos(u/2)
#i <- c(which(x > pi/6), which(x < 0)) #indicator function.
#x[i] <- 2 #to catch overflow errors in g(x)
fg <- x/2
#fg <- g(x) / (2*cos(x))
theta.hat[3] <- mean(fg)
se[3] <- sd(fg)

rbind(theta.hat,se)
```

```
##             [,1]    [,2]    [,3]
## theta.hat 0.1274 0.12770 0.65728
## se        0.1547 0.06943 0.07493
```