Descriptive statistics and Decision Tree

Suhail Shaikh

3/17/2020

Q.1.a) Summarize the main statistics of all the variables in the data set.

```
summary(attitude)
```

```
##      rating        complaints     privileges       learning  
##  Min.   :40.00   Min.   :37.0   Min.   :30.00   Min.   :34.00  
##  1st Qu.:58.75   1st Qu.:58.5   1st Qu.:45.00   1st Qu.:47.00  
##  Median :65.50   Median :65.0   Median :51.50   Median :56.50  
##  Mean   :64.63   Mean   :66.6   Mean   :53.13   Mean   :56.37  
##  3rd Qu.:71.75   3rd Qu.:77.0   3rd Qu.:62.50   3rd Qu.:66.75  
##  Max.   :85.00   Max.   :90.0   Max.   :83.00   Max.   :75.00  
##      raises          critical        advance        
##  Min.   :43.00   Min.   :49.00   Min.   :25.00  
##  1st Qu.:58.25   1st Qu.:69.25   1st Qu.:35.00  
##  Median :63.50   Median :77.50   Median :41.00  
##  Mean   :64.63   Mean   :74.77   Mean   :42.93  
##  3rd Qu.:71.00   3rd Qu.:80.00   3rd Qu.:47.75  
##  Max.   :88.00   Max.   :92.00   Max.   :72.00
```

1.b) How many observations are in the attitude dataset? What function in R did you use to display this information?

```
nrow(attitude)
```

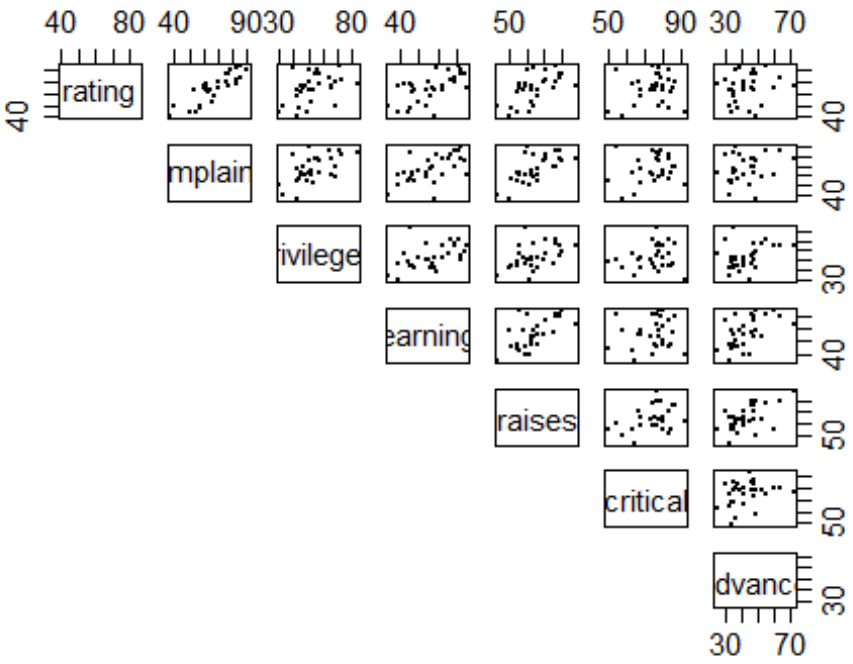
```
## [1] 30
```

```
dim(attitude)
```

```
## [1] 30  7
```

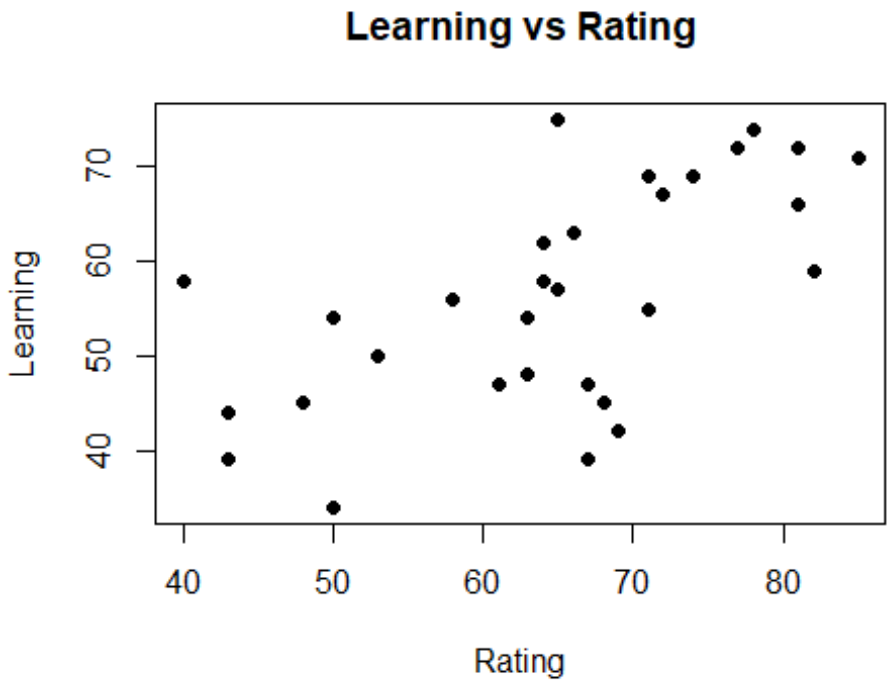
1. 1.c) Produce a scatterplot matrix of the variables in the attitude dataset. What seems to be most correlated with the overall rating?

```
pairs(attitude,lower.panel =NULL,pch=19,cex=0.5,cex.labels=1.4,cex.axis=1.4)
```



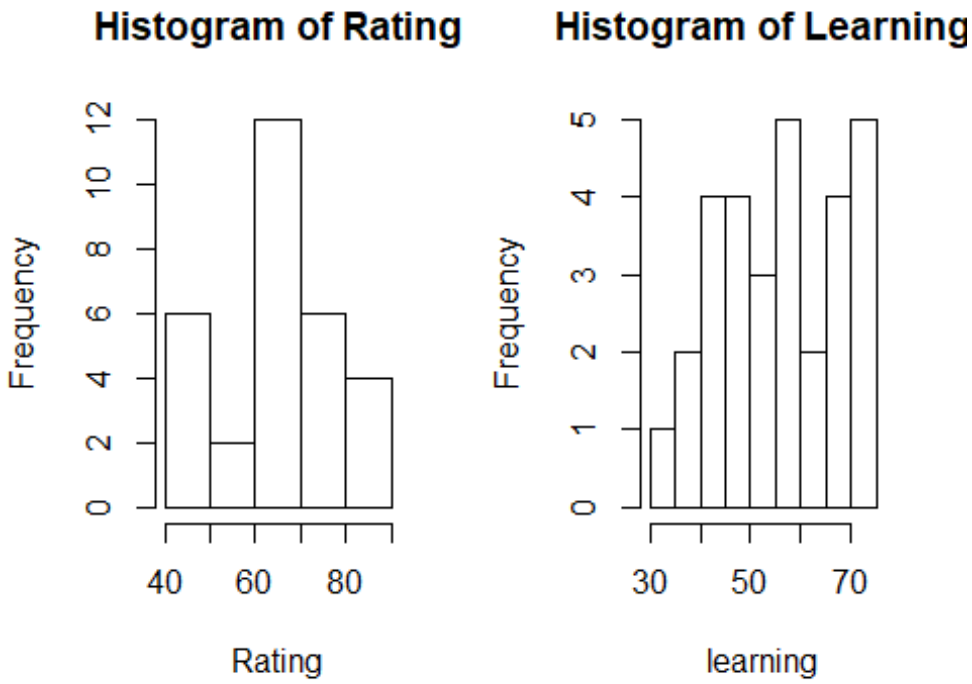
1.d) Produce a scatterplot of rating (on the y-axis) vs. learning (on the x-axis). Add a title to the plot using the title() function.

```
plot(attitude$rating,attitude$learning,pch=19,xlab="Rating",ylab="Learning")
title(main="Learning vs Rating")
```



1.e) Produce 2 side-by-side histograms, one for rating and one for learning. You will need to use par(mfrow=…) to get the two plots together.

```
par(mfrow=c(1,2))

hist(attitude$rating,pch=19,xlab="Rating",main = "Histogram of Rating")
hist(attitude$learning,pch=19,xlab="learning",main = "Histogram of Learning")
```

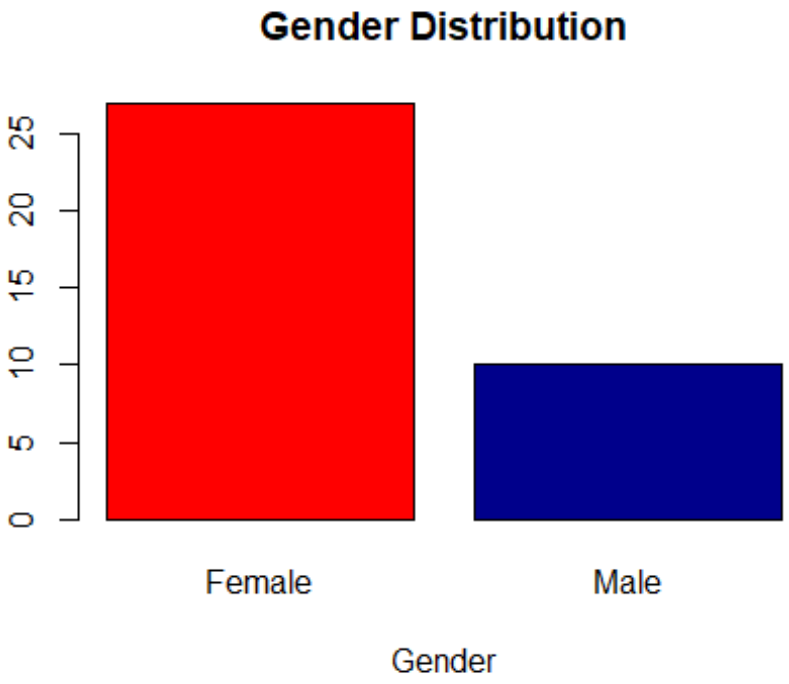


2.a) Make a frequency distribution table for the gender variable to see the frequency distribution

```
library(readxl)
Exercise <-read_excel("exercise.xls")
View(Exercise)

colnames(Exercise) <-
c("weight","height","gender","exerciseperweek","Isregularexercise","friedfood
perweek")
gendertable <- table(Exercise$gender)
```

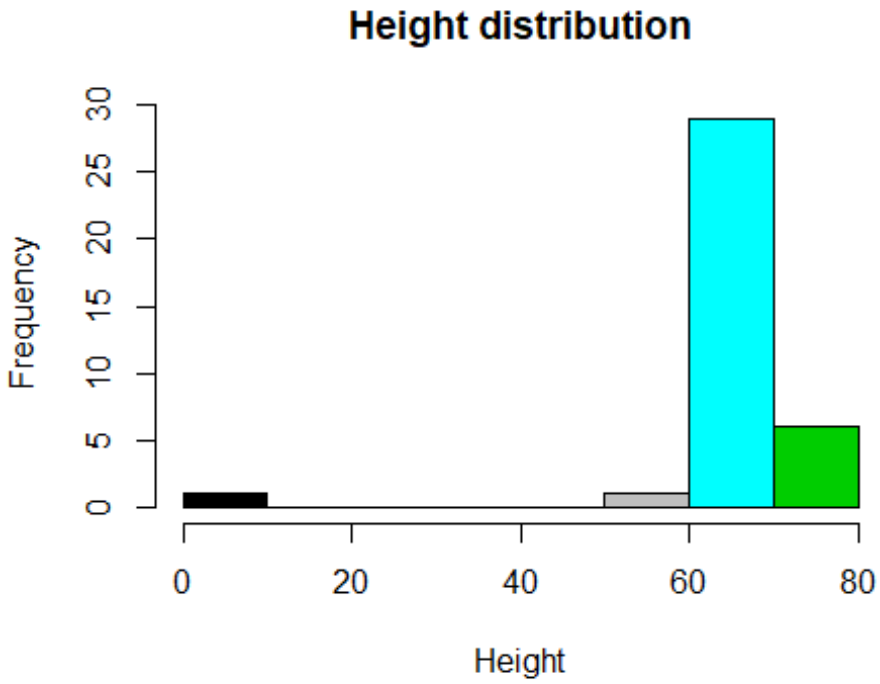
2.b) Make a bar chart for gender variable

```
barplot(gendertable,main="Gender
Distribution",xlab="Gender",col=c("red","darkblue"))     #col is used for
colour
```



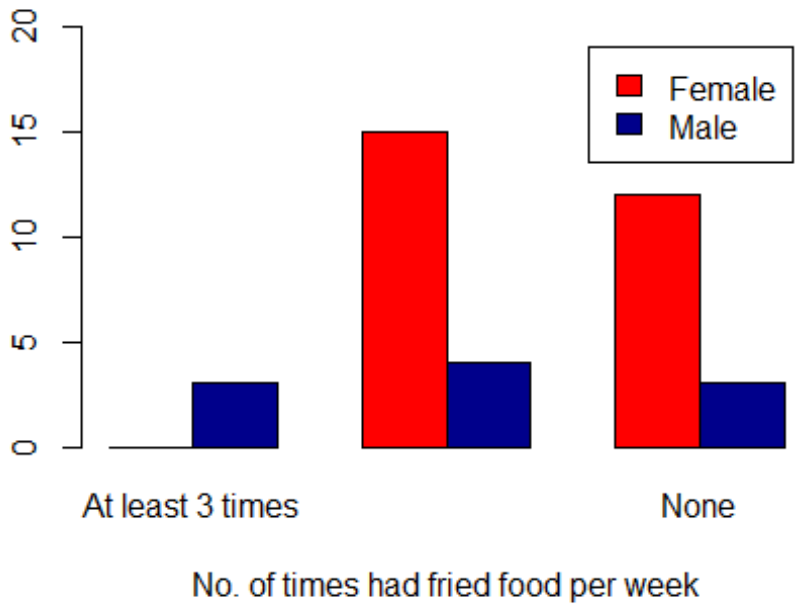
2.c) Make a histogram to display the distribution of the Height variable

```
hist(Exercise$height,main="Height distribution",xlab="Height",col =
Exercise$height)
```



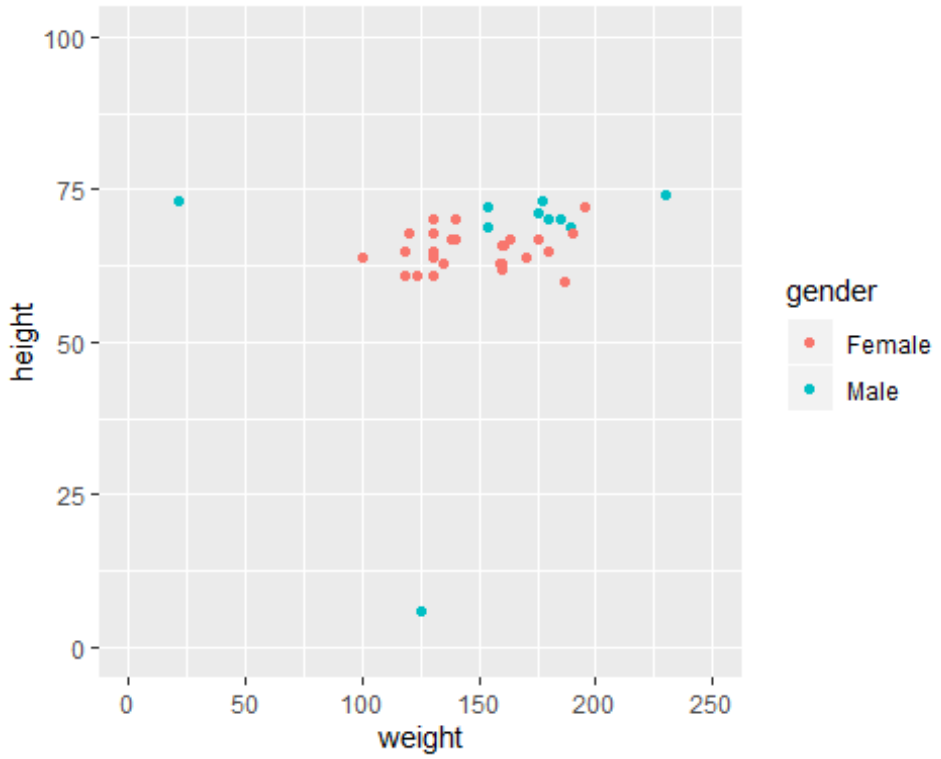
2.d) Make a cluster bar chart (side-by-side bar chart) to examine the correlation between gender and Ate Fried Food variables.

```
t <-table(Exercise$gender,Exercise$friedfoodperweek)
barplot(t,beside = TRUE,legend=TRUE,xlab = "No. of times had fried food per
week",col = c("red","darkblue"),ylim = c(0,20))
```



2.e) Make a scatter plot to examine the correlation between Weight and Height variables, and write a sentence to describe the trend you observed from the scatter plot.

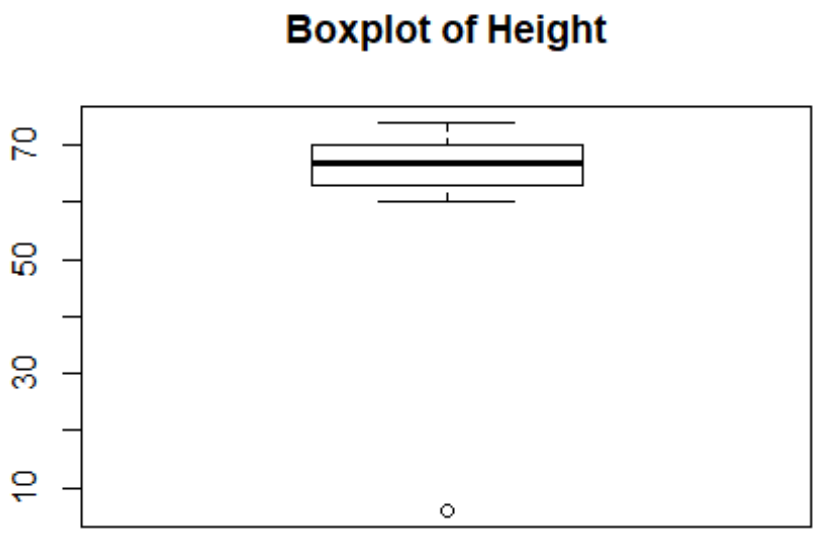
```
library(ggplot2)
s <-ggplot(Exercise,aes(x=weight,y=height))
s+ geom_point(aes(colour=gender)) +ylim(0,100)+xlim(0,250)
```



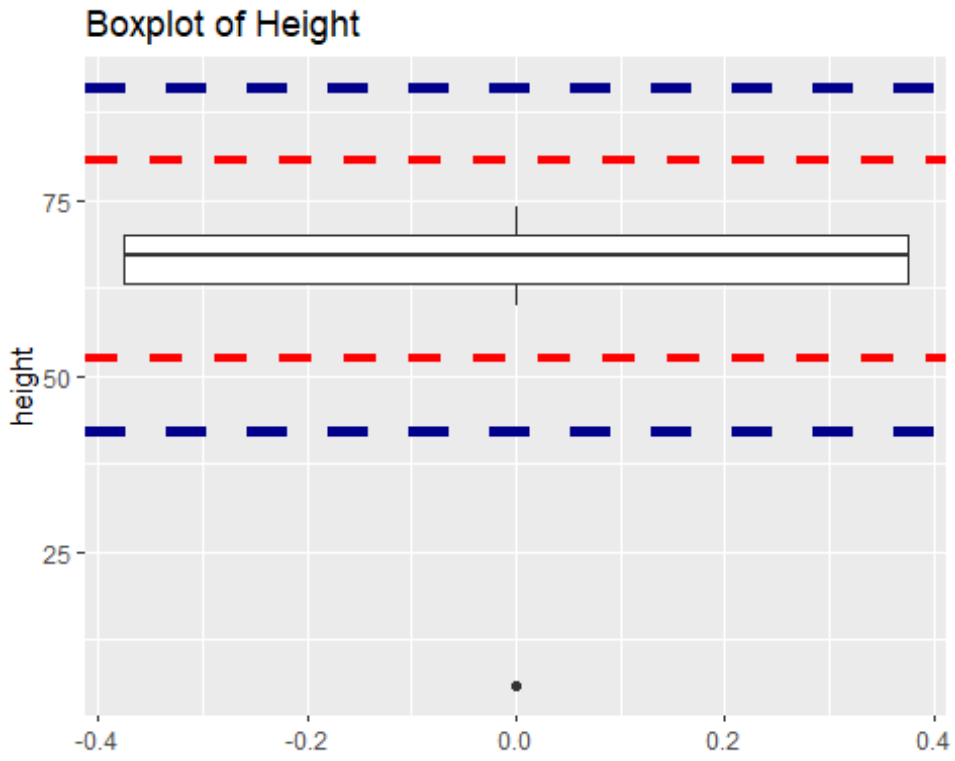

```
#The graph of height Vs Weight is linear with slope zero.
```

2.f) Find the 5-number summary for the Height data and make a boxplot for the Height data with mild and extreme outliers identiï¬ed using inner and outer fences. Draw the boxplot.

```
a <- fivenum(Exercise$height)
#summary(Exercise$height) in summary mean is extra variable which is not
needed

IQR <-a[4]-a[2]
boxplot(Exercise$height,main="Boxplot of Height")
```




```
IF1 <-a[2]-1.5*IQR
IF2 <- a[4]+1.5*IQR

OF1 <-a[2]-3*IQR
OF2 <-a[4]+3*IQR

f <- ggplot(Exercise,aes(y=height))
f+ geom_boxplot()+ggtitle("Boxplot of Height")+
   geom_hline(yintercept =
c(IF1,IF2),linetype="dashed",color="red",size=1.5)+
   geom_hline(yintercept =
c(OF1,OF2),linetype="dashed",color="darkblue",size=2)
```



Q.3.a)All the variables are represented as integer. Write your own function that automatically converts all the integer variables to factors (categorical).

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.2
```

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
```

```
#?birthwt
#View(birthwt)
str(birthwt)
```

```
## 'data.frame':    189 obs. of  10 variables:
##  $ low  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : int  2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: int  0 0 1 1 1 0 0 0 1 1 ...
##  $ ptl  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ht   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ui   : int  1 0 0 1 1 0 0 0 0 0 ...
##  $ ftv  : int  0 3 1 2 0 0 1 1 1 0 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

```
converttofactor<-function(x){
  return(as.factor(x))

}
```

```
str(birthwt)
```

```
## 'data.frame':    189 obs. of  10 variables:
##  $ low  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : int  2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: int  0 0 1 1 1 0 0 0 1 1 ...
##  $ ptl  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ht   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ui   : int  1 0 0 1 1 0 0 0 0 0 ...
##  $ ftv  : int  0 3 1 2 0 0 1 1 1 0 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

```
 #c(1,4:9)
birthwt[,c(1,4,5,6,7,8,9)] <-
lapply(birthwt[,c(1,4,5,6,7,8,9)],converttofactor)
str(birthwt)
```

```
## 'data.frame':    189 obs. of  10 variables:
##  $ low  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
```

```
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : Factor w/ 3 levels "1","2","3": 2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: Factor w/ 2 levels "0","1": 1 1 2 2 2 1 1 1 2 2 ...
##  $ ptl  : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ht   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ui   : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 1 1 1 1 ...
##  $ ftv  : Factor w/ 6 levels "0","1","2","3",..: 1 4 2 3 1 1 2 2 2 1 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

###Q.3.b) Repeat part (a) using mutate() and mapvalues() functions.

```
library(plyr)
birthwt<-mutate(birthwt,
         race = as.factor(mapvalues(race , c("white", "black", "other"),
c("1","2", "3"))),
         smoke =as.factor(mapvalues(smoke , c("no", "yes"), c("0","1"))),
         ptl = as.factor(mapvalues(ptl , c("no", "yes"), c("0","1"))),
         ht = as.factor(mapvalues(ht , c("no", "yes"), c("0","1"))),
         ui = as.factor(mapvalues(ui , c("no", "yes"), c("0","1"))),
         ftv = as.factor(mapvalues(ftv , c("no", "yes"), c("0","1"))),
         low = as.factor(mapvalues(low , c("no", "yes"), c("0","1"))))

## The following `from` values were not present in `x`: white, black, other

## The following `from` values were not present in `x`: no, yes
## The following `from` values were not present in `x`: no, yes
## The following `from` values were not present in `x`: no, yes
## The following `from` values were not present in `x`: no, yes
## The following `from` values were not present in `x`: no, yes
## The following `from` values were not present in `x`: no, yes
```

3.c) Use the tapply() function to see what the average birthweight looks like when broken down by race and smoking status. Does smoking status appear to have an eï¬ect on birth weight? Does the eï¬ect of smoking status appear to be consistent across racial groups? What is the association between race and birth weight?

```
tapply(birthwt$bwt, birthwt$race, mean)

##        1        2        3
## 3102.719 2719.692 2805.284

tapply(birthwt$bwt, birthwt$smoke, mean)

##        0        1
## 3055.696 2771.919

#Theory

#Yes, it looks like smoking status does have an effect on birth weight as
average birth weight for non - smokers is 3055.6, however, for smokers it is
```

```
2771.92*

 # *Yes, the effect of smoking status appears to be consistent across racial groups as the average birth weight for smokers across all the races is comparatively lesser to non-smokers across the racial groups*

  #*We see that Race =1 (white) has the highest average birth weight (3102.72), Race =3 (Other) have a average birth weight of 2805.2 whereas for Race =2 (Black), the average birth weight is the lowest 2719.70*
```

3.d) Use kable() function from knitr to dispaly the table you get in part (c)

```
library(knitr)
#install.packages("kableExtra")
library(kableExtra)

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows

kable(birthwt) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

low
age
lwt
race
smoke
ptl
ht
ui
ftv
bwt
0
19
182
2
0
0
0
1
0
2523
0

33
155
3
0
0
0
0
3
2551
0
20
105
1
1
0
0
0
1
2557
0
21
108
1
1
0
0
1
2
2594
0
18
107
1
1
0
0
1
0
2600
0

21
124
3
0
0
0
0
0
2622
0
22
118
1
0
0
0
0
1
2637
0
17
103
3
0
0
0
0
1
2637
0
29
123
1
1
0
0
0
1
2663
0

26
113
1
1
0
0
0
0
2665
0
19
95
3
0
0
0
0
0
2722
0
19
150
3
0
0
0
0
1
2733
0
22
95
3
0
0
1
0
0
2751
0

30
107
3
0
1
0
1
2
2750
0
18
100
1
1
0
0
0
0
2769
0
18
100
1
1
0
0
0
0
2769
0
15
98
2
0
0
0
0
0
2778
0

25
118
1
1
0
0
0
3
2782
0
20
120
3
0
0
0
1
0
2807
0
28
120
1
1
0
0
0
1
2821
0
32
121
3
0
0
0
0
2
2835
0

31
100
1
0
0
0
1
3
2835
0
36
202
1
0
0
0
0
1
2836
0
28
120
3
0
0
0
0
0
2863
0
25
120
3
0
0
0
1
2
2877
0

28
167
1
0
0
0
0
0
2877
0
17
122
1
1
0
0
0
0
2906
0
29
150
1
0
0
0
0
2
2920
0
26
168
2
1
0
0
0
0
2920
0

17
113
2
0
0
0
0
1
2920
0
17
113
2
0
0
0
0
1
2920
0
24
90
1
1
1
0
0
1
2948
0
35
121
2
1
1
0
0
1
2948
0

25
155
1
0
0
0
0
1
2977
0
25
125
2
0
0
0
0
0
2977
0
29
140
1
1
0
0
0
2
2977
0
19
138
1
1
0
0
0
2
2977
0

27
124
1
1
0
0
0
0
2922
0
31
215
1
1
0
0
0
2
3005
0
33
109
1
1
0
0
0
1
3033
0
21
185
2
1
0
0
0
2
3042
0

19
189
1
0
0
0
0
2
3062
0
23
130
2
0
0
0
0
1
3062
0
21
160
1
0
0
0
0
0
3062
0
18
90
1
1
0
0
1
0
3062
0

18
90
1
1
0
0
1
0
3062
0
32
132
1
0
0
0
0
4
3080
0
19
132
3
0
0
0
0
0
3090
0
24
115
1
0
0
0
0
2
3090
0

22
85
3
1
0
0
0
0
3090
0
22
120
1
0
0
1
0
1
3100
0
23
128
3
0
0
0
0
0
3104
0
22
130
1
1
0
0
0
0
3132
0

30
95
1
1
0
0
0
2
3147
0
19
115
3
0
0
0
0
0
3175
0
16
110
3
0
0
0
0
0
3175
0
21
110
3
1
0
0
1
0
3203
0

30
153
3
0
0
0
0
0
3203
0
20
103
3
0
0
0
0
0
3203
0
17
119
3
0
0
0
0
0
3225
0
17
119
3
0
0
0
0
0
3225
0

23
119
3
0
0
0
0
2
3232
0
24
110
3
0
0
0
0
0
3232
0
28
140
1
0
0
0
0
0
3234
0
26
133
3
1
2
0
0
0
3260
0

20
169
3
0
1
0
1
1
3274
0
24
115
3
0
0
0
0
2
3274
0
28
250
3
1
0
0
0
6
3303
0
20
141
1
0
2
0
1
1
3317
0

22
158
2
0
1
0
0
2
3317
0
22
112
1
1
2
0
0
0
3317
0
31
150
3
1
0
0
0
2
3321
0
23
115
3
1
0
0
0
1
3331
0

16
112
2
0
0
0
0
0
3374
0
16
135
1
1
0
0
0
0
3374
0
18
229
2
0
0
0
0
0
3402
0
25
140
1
0
0
0
0
0
1
3416
0

32
134
1
1
1
0
0
4
3430
0
20
121
2
1
0
0
0
0
3444
0
23
190
1
0
0
0
0
0
3459
0
22
131
1
0
0
0
0
1
3460
0

32
170
1
0
0
0
0
0
3473
0
30
110
3
0
0
0
0
0
3544
0
20
127
3
0
0
0
0
0
3487
0
23
123
3
0
0
0
0
0
3544
0

17
120
3
1
0
0
0
0
3572
0
19
105
3
0
0
0
0
0
3572
0
23
130
1
0
0
0
0
0
3586
0
36
175
1
0
0
0
0
0
3600
0

22
125
1
0
0
0
0
1
3614
0
24
133
1
0
0
0
0
0
0
3614
0
21
134
3
0
0
0
0
2
3629
0
19
235
1
1
0
1
0
0
0
3629
0

25
95
1
1
3
0
1
0
3637
0
16
135
1
1
0
0
0
0
3643
0
29
135
1
0
0
0
0
1
3651
0
29
154
1
0
0
0
0
1
3651
0

19
147
1
1
0
0
0
0
3651
0
19
147
1
1
0
0
0
0
3651
0
30
137
1
0
0
0
0
1
3699
0
24
110
1
0
0
0
0
1
3728
0

19
184
1
1
0
1
0
0
3756
0
24
110
3
0
1
0
0
0
3770
0
23
110
1
0
0
0
0
1
3770
0
20
120
3
0
0
0
0
0
0
3770
0

25
241
2
0
0
1
0
0
3790
0
30
112
1
0
0
0
0
1
3799
0
22
169
1
0
0
0
0
0
3827
0
18
120
1
1
0
0
0
2
3856
0

16
170
2
0
0
0
0
4
3860
0
32
186
1
0
0
0
0
2
3860
0
18
120
3
0
0
0
0
1
3884
0
29
130
1
1
0
0
0
2
3884
0

33
117
1
0
0
0
1
1
3912
0
20
170
1
1
0
0
0
0
3940
0
28
134
3
0
0
0
0
1
3941
0
14
135
1
0
0
0
0
0
0
3941
0

28
130
3
0
0
0
0
0
3969
0
25
120
1
0
0
0
0
2
3983
0
16
95
3
0
0
0
0
1
3997
0
20
158
1
0
0
0
0
1
3997
0

26
160
3
0
0
0
0
0
4054
0
21
115
1
0
0
0
0
1
4054
0
22
129
1
0
0
0
0
0
4111
0
25
130
1
0
0
0
0
2
4153
0

31
120
1
0
0
0
0
2
4167
0
35
170
1
0
1
0
0
1
4174
0
19
120
1
1
0
0
0
0
4238
0
24
116
1
0
0
0
0
1
4593
0

45
123
1
0
0
0
0
1
4990
1
28
120
3
1
1
0
1
0
709
1
29
130
1
0
0
0
1
2
1021
1
34
187
2
1
0
1
0
0
1135
1

25
105
3
0
1
1
0
0
1330
1
25
85
3
0
0
0
1
0
1474
1
27
150
3
0
0
0
0
0
1588
1
23
97
3
0
0
0
1
1
1588
1

24
128
2
0
1
0
0
1
1701
1
24
132
3
0
0
1
0
0
1729
1
21
165
1
1
0
1
0
1
1790
1
32
105
1
1
0
0
0
0
1818
1

19
91
1
1
2
0
1
0
1885
1
25
115
3
0
0
0
0
0
1893
1
16
130
3
0
0
0
0
1
1899
1
25
92
1
1
0
0
0
0
1928
1

20
150
1
1
0
0
0
2
1928
1
21
200
2
0
0
0
1
2
1928
1
24
155
1
1
1
0
0
0
1936
1
21
103
3
0
0
0
0
0
1970
1

20
125
3
0
0
0
1
0
2055
1
25
89
3
0
2
0
0
1
2055
1
19
102
1
0
0
0
0
2
2082
1
19
112
1
1
0
0
1
0
2084
1

26
117
1
1
1
0
0
0
2084
1
24
138
1
0
0
0
0
0
2100
1
17
130
3
1
1
0
1
0
2125
1
20
120
2
1
0
0
0
3
2126
1

22
130
1
1
1
0
1
1
2187
1
27
130
2
0
0
0
1
0
2187
1
20
80
3
1
0
0
1
0
2211
1
17
110
1
1
0
0
0
0
2225
1

25
105
3
0
1
0
0
1
2240
1
20
109
3
0
0
0
0
0
2240
1
18
148
3
0
0
0
0
0
2282
1
18
110
2
1
1
0
0
0
2296
1

20
121
1
1
1
0
1
0
2296
1
21
100
3
0
1
0
0
4
2301
1
26
96
3
0
0
0
0
0
2325
1
31
102
1
1
1
0
0
1
2353
1

15
110
1
0
0
0
0
0
2353
1
23
187
2
1
0
0
0
1
2367
1
20
122
2
1
0
0
0
0
2381
1
24
105
2
1
0
0
0
0
2381
1

15
115
3
0
0
0
1
0
2381
1
23
120
3
0
0
0
0
0
2410
1
30
142
1
1
1
0
0
0
2410
1
22
130
1
1
0
0
0
1
2410
1

17
120
1
1
0
0
0
3
2414
1
23
110
1
1
1
0
0
0
2424
1
17
120
2
0
0
0
0
2
2438
1
26
154
3
0
1
1
0
1
2442
1

20
105
3
0
0
0
0
3
2450
1
26
190
1
1
0
0
0
0
2466
1
14
101
3
1
1
0
0
0
2466
1
28
95
1
1
0
0
0
2
2466
1

14
100
3
0
0
0
0
2
2495
1
23
94
3
1
0
0
0
0
2495
1
17
142
2
0
0
1
0
0
2495
1
21
130
1
1
0
1
0
3
2495

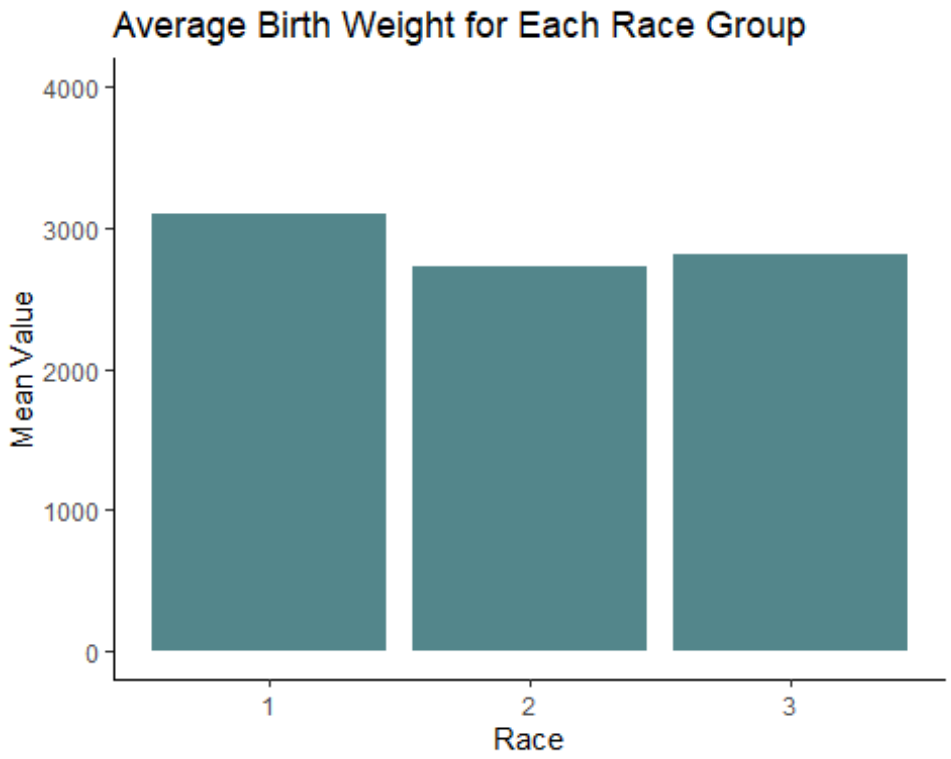
3.e) Use ddply() function to get the average birthweight by motherâ??s race and compare it with tapply() function

```
library(plyr)

table_d<-ddply(birthwt,.(race),summarize,mean_val=mean(bwt))
table_d
```

```
##   race mean_val
## 1    1 3102.719
## 2    2 2719.692
## 3    3 2805.284
```

3.f) Use ggplot2() to plot the average birthweight (computed in part (e)) for each race group in a bar plot

```
library(ggplot2)

ggplot(data=table_d,aes(x=race,y=mean_val),) + geom_bar(stat =
"identity",fill='cadetblue4') + xlab("Race") +
  ylab ("Mean Value") + ggtitle("Average Birth Weight for Each Race Group") +
ylim(c(0,4000)) +
  theme(panel.grid=element_blank(),panel.background = element_blank(),
        axis.line = element_line(colour = "black"))
```



3.g) Use ddply() function to look at the average birthweight and proportion of babies with low birthweight broken down by smoking status

```
str(birthwt)
```

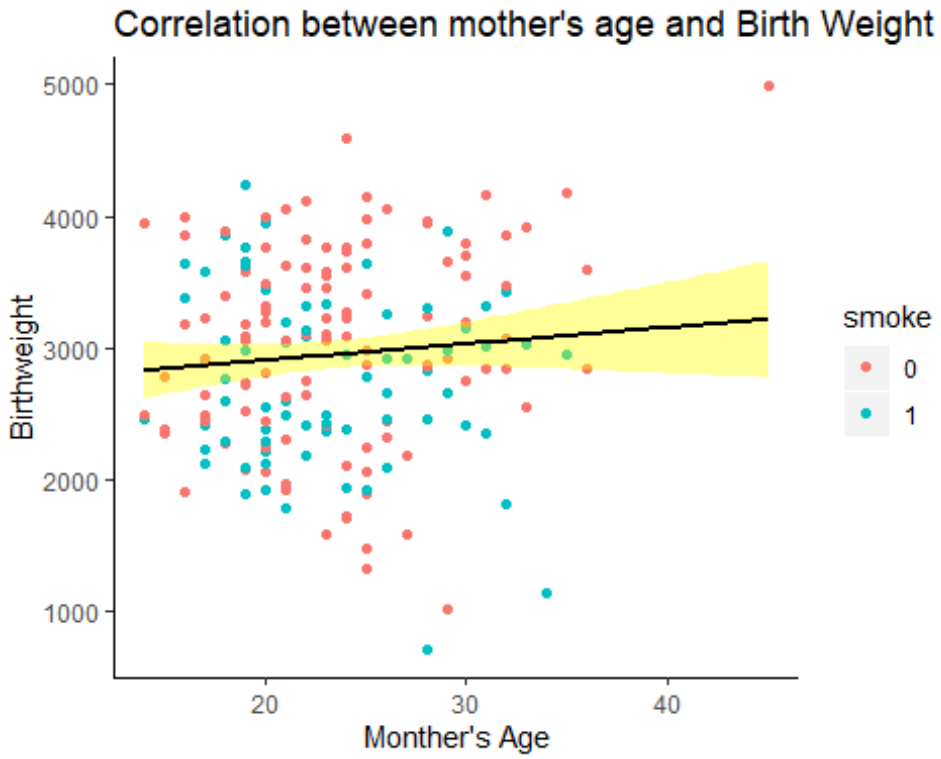
```
## 'data.frame':    189 obs. of  10 variables:
##  $ low  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : Factor w/ 3 levels "1","2","3": 2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: Factor w/ 2 levels "0","1": 1 1 2 2 2 1 1 1 2 2 ...
##  $ ptl  : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ht   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ui   : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 1 1 1 1 ...
##  $ ftv  : Factor w/ 6 levels "0","1","2","3",..: 1 4 2 3 1 1 2 2 2 1 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

```
table_g<-ddply(birthwt, ~ smoke, summarize,
      avg_wt = mean(bwt),
      low_birth_prop = mean(low == 1))

table_g
```

```
##   smoke   avg_wt low_birth_prop
## 1     0 3055.696      0.2521739
## 2     1 2771.919      0.4054054
```

3.i) Is the motherâ??s age correlated with birth weight? Does the correlation vary with smoking status?

```
ggplot(birthwt,aes(age,bwt,smoke)) + geom_point(aes(color=smoke)) +
  xlab("Monther's Age") + ylab("Birthweight") +
  geom_smooth(method='lm',fill='yellow',col='black') +
  ggtitle("Correlation between mother's age and Birth Weight")+
  theme(panel.grid=element_blank(),panel.background = element_blank(),
        axis.line = element_line(colour = "black"))
```




```
#*Yes, the correlation varies with smoking status. We see that most of the
data points above the line of best fit are non-smokers, whereas, the data
points below the line are smokers. This further shows that smokers during
pregnancy have a lower birth rate as compared to the non-smokers*
```

Q.4.a) What type of variable is price? Would you expect its distribution to be symmetric, right-skewed, or left-skewed? Why? Make a histogram of the distribution of diamond prices. Does the shape of the distribution match your expectation? (Use geom histogram()).

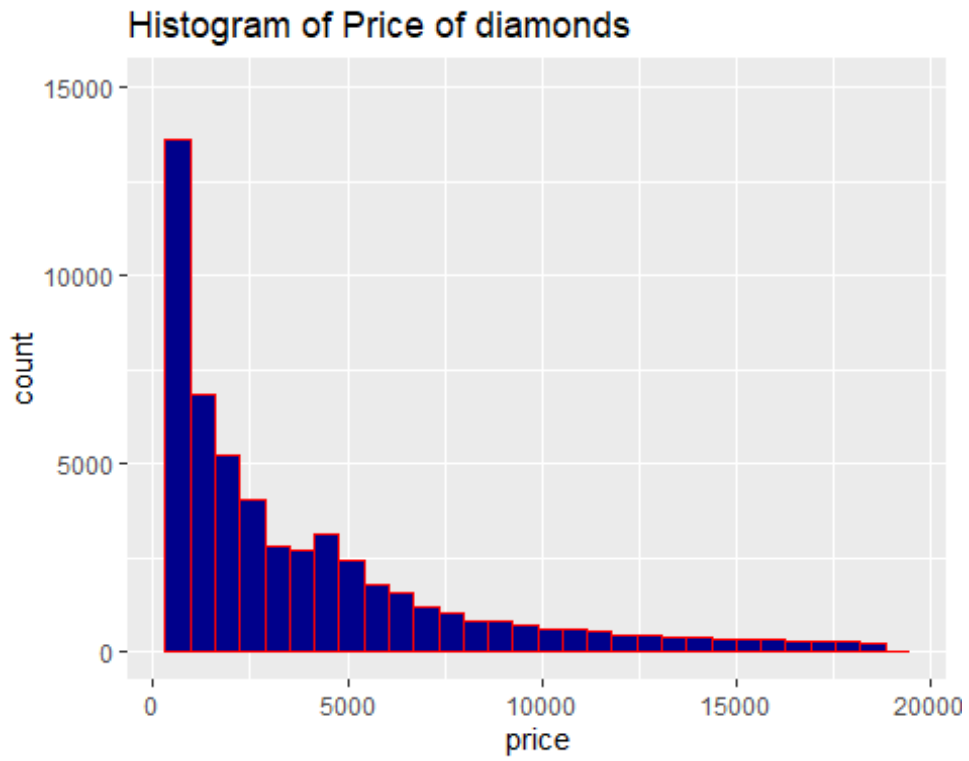
```
library(ggplot2)
#data(diamonds)
View(diamonds)
str(diamonds)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  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  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
```

```
##  $ y       : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z       : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
s<- ggplot(diamonds,aes(x=price))
s+ geom_histogram(fill="darkblue",colour="red")+ggtitle("Histogram of Price
of diamonds")+ylim(0,15000)
```

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




```
#Price is a dependent variable whose value depend on carat and cut.
#The distribution of variable y is right skew  as frequency of price is goes
on decreasing as price increases.
```

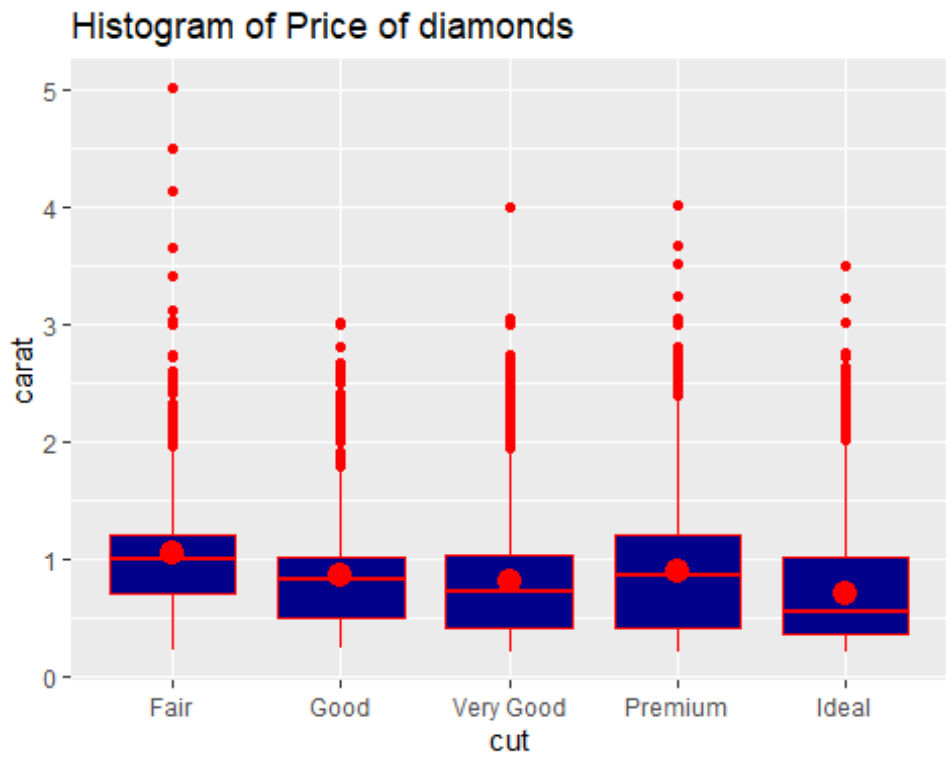
Q.4.b) Visualize a few other numerical variables in the dataset and discuss any interesting features. When describing distributions of numerical variables we might also want to view statistics like mean, median, etc.

```
summary(diamonds)
```

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

```
p<- ggplot(diamonds,aes(x=cut,y=carat))
p+ geom_boxplot(fill="darkblue",colour="red")+ggtitle("Histogram of Price of diamonds")+#points(diamonds$cut, means$carat, col = "red")+
  stat_summary(fun.y=mean, geom="point",colour="red",shape=19,size=4)
```



Q.4.c) What type of variable is color? Which color is most prominently represented in the dataset?

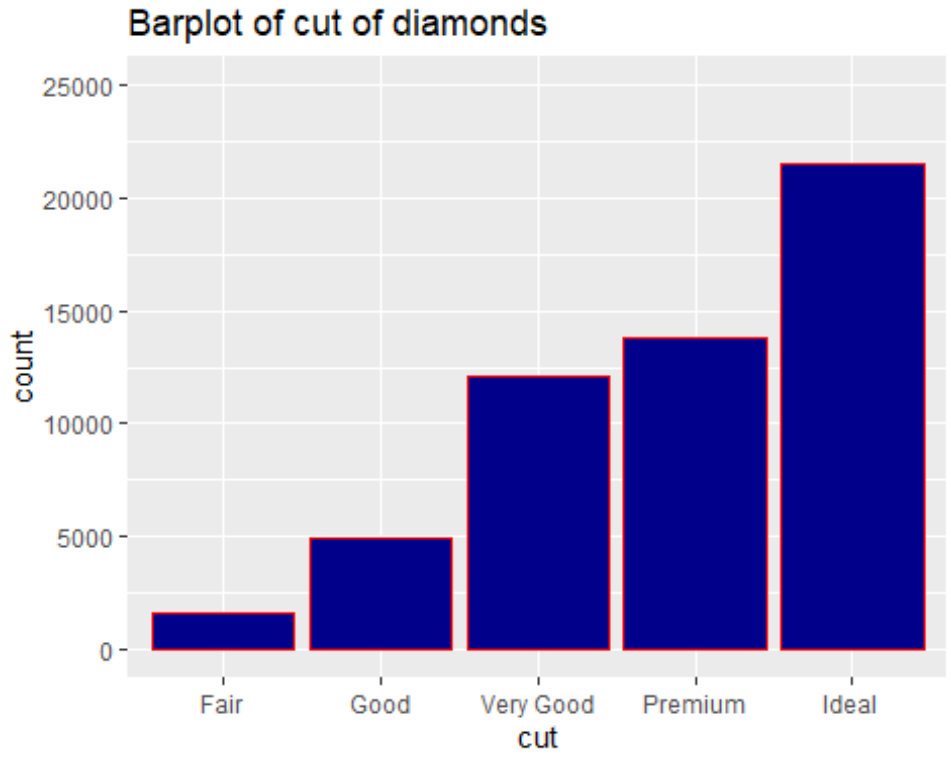
```
str(diamonds$color)
```

```
##  Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
```

```
#colour is a variable of type factor with 7 levels. COlor G is most
prominantly represented in dataset.
```

Q.4.d) Make a bar plot of the distribution of cut, and describe its distribution (Use georm bar())

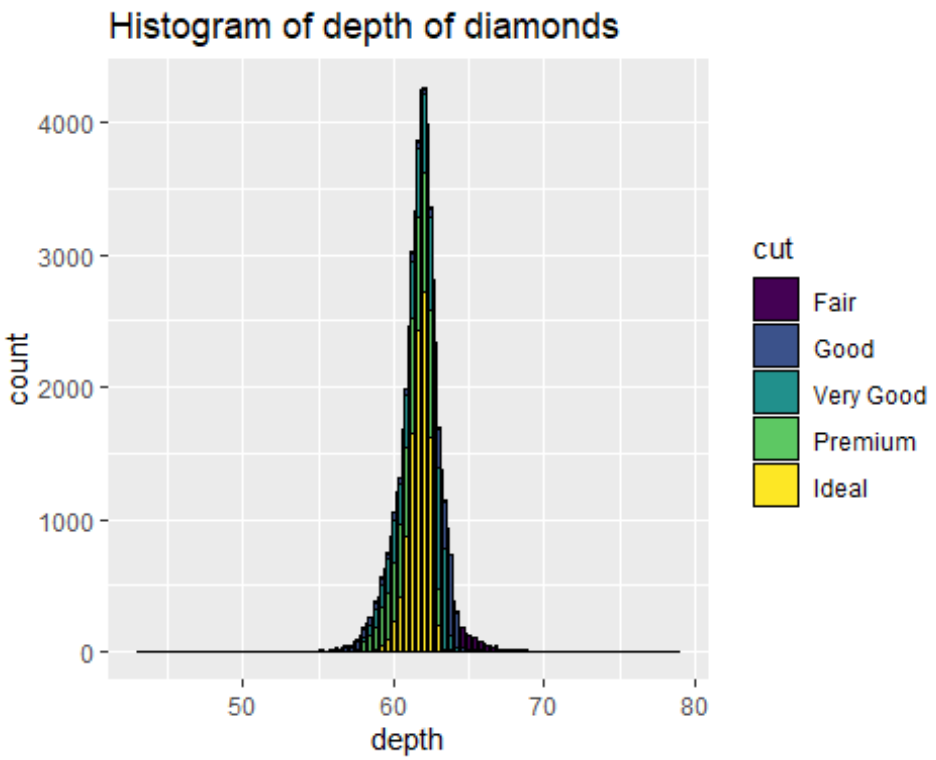
```
b<- ggplot(diamonds,aes(x=cut))
b+ geom_bar(fill="darkblue",colour="red")+ggtitle("Barplot of cut of
diamonds")+ylim(0,25000)
```



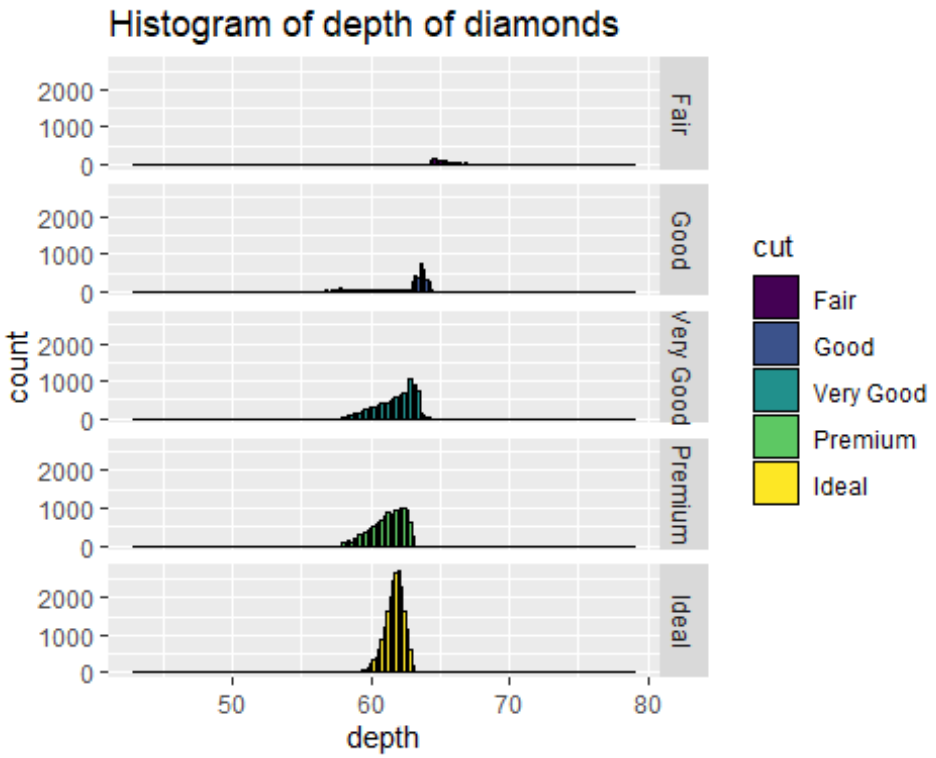

```
#The graph tells that the frequency of diamonds is increasing as the cut is
getting ideal.
```

Q.4.e) Make a histogram of the depths of diamonds, with binwidth of 0.2%, and add another variable (say, cut) to the visualization. You can do this either using an aesthetic or a facet. Typical diamonds of which cut have the highest depth? On average, does depth increase or decrease as cut grade increase or decrease?

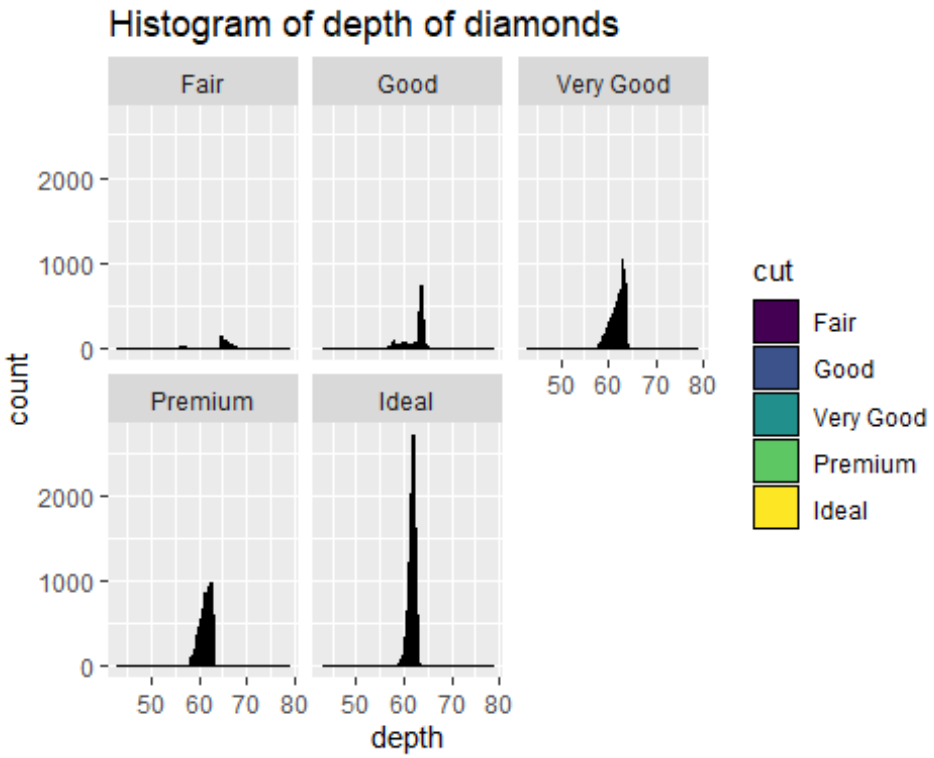
```
b<- ggplot(diamonds,aes(x=depth))
b+
geom_histogram(binwidth=0.2,aes(fill=cut),colour="black")+ggtitle("Histogram
of depth of diamonds")#+ylim(0,15000)
```




```
#By facet_grid

b+
geom_histogram(binwidth=0.2,aes(fill=cut),colour="black")+ggtitle("Histogram
of depth of diamonds")+
  facet_grid(cut~.)
```



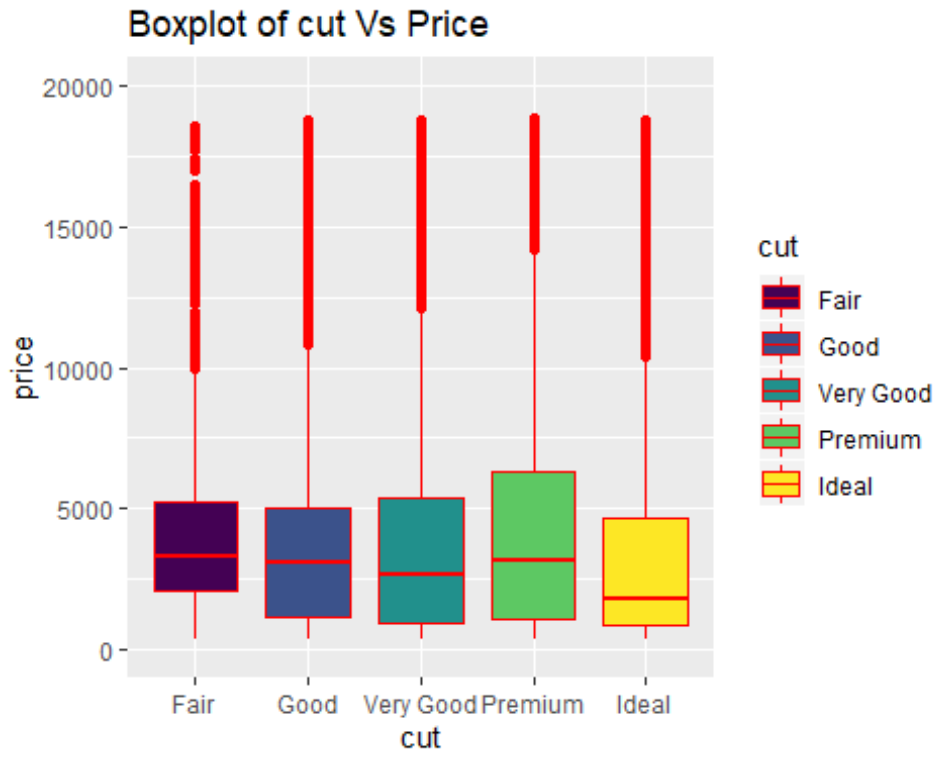

```
#By facet_wrap

b+
geom_histogram(binwidth=0.2,aes(fill=cut),colour="black")+ggtitle("Histogram
of depth of diamonds")+
  facet_wrap(cut~.)
```




```
#As the cut ibecomes better the depth reduces
```

Q.4.f) Compare the distribution of price for the diï¬erent cuts. Does anything seem unusual? Describe.

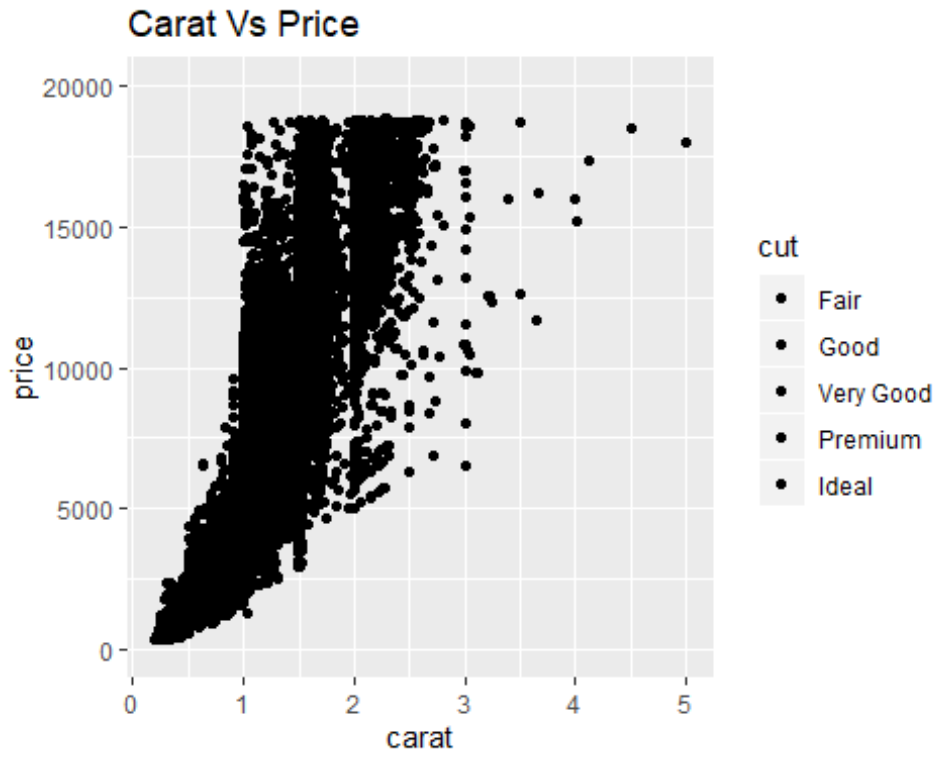
```
f<- ggplot(diamonds,aes(x=cut,y=price))
f+ geom_boxplot(aes(fill=cut),colour="red")+ggtitle("Boxplot of cut Vs
Price")+ylim(0,20000)
```




```
#There are many outliers in this boxplot of Price vs Cut. Range for ideal
diamond cuts is lesser than other diamond cuts.
```

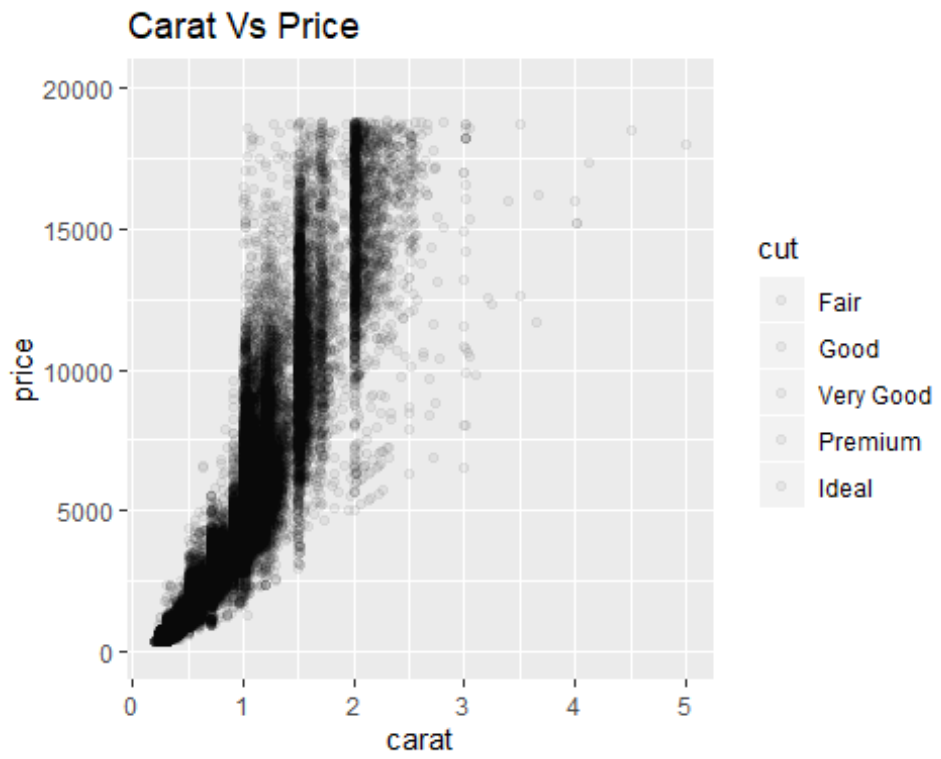
Q.4.g) Draw a scatterplot showing the price (y-axis) as a function of the carat (size).

```
g<- ggplot(diamonds,aes(x=carat,y=price))
g+ geom_point(aes(fill=cut))+geom_jitter()+ggtitle("Carat Vs
Price")+ylim(0,20000)
```



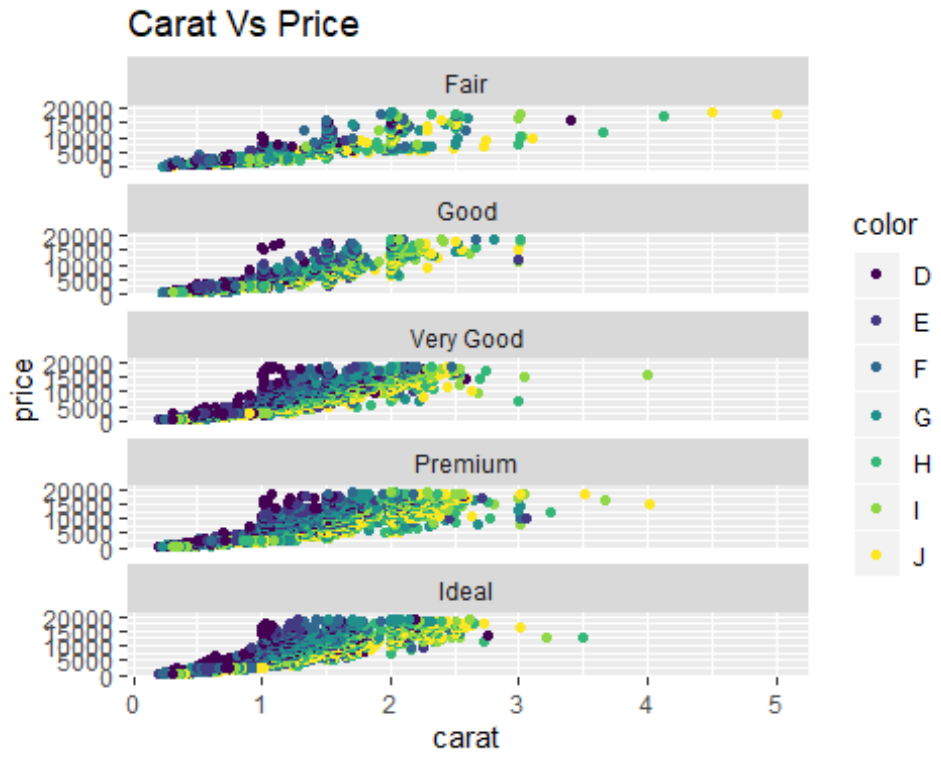
Q.4.h) Shrink the points in your scatter plot in part (g) using the alpha argument in geom point.

```
h<- ggplot(diamonds,aes(x=carat,y=price))
h+ geom_point(alpha=0.05,aes(fill=cut))+ggtitle("Carat Vs
Price")+ylim(0,20000)
```



Q.4.i) Use facet wrap(â¼ factor1 + factor2 + … + factorn) command to create scatter plots showing how diamond price varies with carat size for diï¬erent values of â cutâ (use colour = color in aes()).

```
i <-ggplot(diamonds,aes(x=carat,y=price))
i+ geom_point(aes(colour=color))+ggtitle("Carat Vs Price")+
  facet_wrap(cut~.,ncol=1)+ylim(0,20000)
```



Q.5.a) Load the data and check the attributes of the data. How many variables are in this data set?

```
setwd("E:\\Fall 2019\\IDS 572\\Assg1")
library(readxl)
Diabetes <- read_xlsx("Pima Indian Diabetes.xlsx")

#There are 8 independent variable and one dependent variable `Class variable`
```

Q.5.b) Choose the ï¬rst 80% of the data for training and the remaining 20% data for testing.

```
library(e1071)
library(caret)

## Loading required package: lattice

library(rpart)
library(rpart.plot)

outlier <- function(val){
  iqr <- IQR(val)
  q1 <- as.numeric(quantile(val,.25))
  q3 <- as.numeric(quantile(val,.75))
  upper <- q3+(1.5*iqr)

  lower <- q1-(1.5*iqr)
  ifelse ( val < upper & val > lower, val, NA)
```

```
}
#View(val)

# calling function
diabetes_op<-sapply(Diabetes[,1:8], outlier)

diabetes_cleaned<-data.frame(diabetes_op,Diabetes[,9])
diabetes_F<-na.omit(diabetes_cleaned)
diabetes_F$Class.variable <- as.factor(diabetes_F$Class.variable)

str(diabetes_F)
```

```
## 'data.frame':    639 obs. of  9 variables:
##  $ Number.of.times.pregnant
: num  6 1 8 1 5 3 4 10 5 0 ...
##  $
Plasma.glucose.concentration.a.2.hours.in.an.oral.glucose.tolerance.test: num
148 85 183 89 116 78 110 168 166 118 ...
##  $ Diastolic.blood.pressure
: num  72 66 64 66 74 50 92 74 72 84 ...
##  $ Triceps.skin.fold.thickness
: num  35 29 0 23 0 32 0 0 19 47 ...
##  $ X2.Hour.serum.insulin
: num  0 0 0 94 0 88 0 0 175 230 ...
##  $ Body.mass.index
: num  33.6 26.6 23.3 28.1 25.6 31 37.6 38 25.8 45.8 ...
##  $ Diabetes.pedigree.function
: num  0.627 0.351 0.672 0.167 0.201 0.248 0.191 0.537 0.587 0.551 ...
##  $ Age
: num  50 31 32 21 30 26 30 34 51 31 ...
##  $ Class.variable
: Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 2 2 2 ...
##  - attr(*, "na.action")= 'omit' Named int  5 8 9 10 13 14 16 19 40 44 ...
##   ..- attr(*, "names")= chr  "5" "8" "9" "10" ...
```

```
set.seed(2)
sample=sample(1:nrow(diabetes_F),floor(nrow(diabetes_F)*0.8))
train <-diabetes_F[sample, ]
test <-diabetes_F[-sample,]
```

Q.5.c) Use â□□rpartâ□□ function to create a tree using the training data . What is the accuracy of your model based on training data?

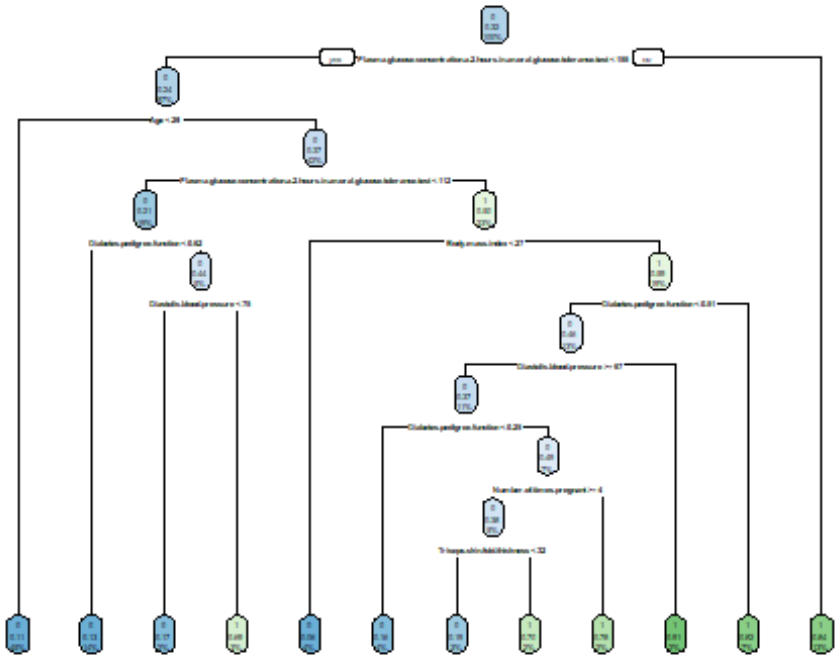
```
#Training the decision tree classifier
tree_model <-rpart(Class.variable~.,data=train,method="class")


#Predictions on training dataset
diabetes.predicted_train<- predict(tree_model,train,type = "class")
```

```r
#Confusion matrix for evaluating the model on training dataset
confusionMatrix(diabetes.predicted_train,train$Class.variable)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 323  44
##          1  27 117
##
##                Accuracy : 0.8611
##                  95% CI : (0.828, 0.8899)
##     No Information Rate : 0.6849
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6686
##
##  Mcnemar's Test P-Value : 0.05758
##
##             Sensitivity : 0.9229
##             Specificity : 0.7267
##          Pos Pred Value : 0.8801
##          Neg Pred Value : 0.8125
##              Prevalence : 0.6849
##          Detection Rate : 0.6321
##    Detection Prevalence : 0.7182
##       Balanced Accuracy : 0.8248
##
##        'Positive' Class : 0
##
```

Q.5.d) Plot your decision tree. How many leaves are in your tree? Are these leaves pure?

```r
rpart.plot(tree_model)
```



```
#There are 15 variables in the tree. All of these leaves are pure.
```

Q.5.e) Provide two strongest If-Then rules from this decision tree. Please explain why these rules are chosen.

```
#*Two strongest decision rules are ->*

#*1. If Plasma Glucose conc < 144 and Age < 29 Then 0*

#*2. If Plasma Glucose conc >= 144 and Glucose conc >= 155 Then 1*
```

Q.5.f) What are the most important variables based on your decision tree models?

```
# Most important variables are at the top nodes of the tree. Here the most
important nodes are ghluccose concentration, age and body mass index.
```

Q.5.g) Apply the decision tree on test data and report your prediction (just the code is suï¬cient for this part). What is the accuracy of your model on the test data?

```
#Predictions on training dataset
diabetes.predicted_test<- predict(tree_model,test,type = "class")

#Confusion matrix for evaluating the model on testing dataset
confusionMatrix(diabetes.predicted_test,test$Class.variable)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 77 21
##          1 12 18
##
##                Accuracy : 0.7422
##                  95% CI : (0.6574, 0.8154)
##     No Information Rate : 0.6953
##     P-Value [Acc > NIR] : 0.1450
##
##                   Kappa : 0.3494
##
##  Mcnemar's Test P-Value : 0.1637
##
##             Sensitivity : 0.8652
##             Specificity : 0.4615
##          Pos Pred Value : 0.7857
##          Neg Pred Value : 0.6000
##              Prevalence : 0.6953
##          Detection Rate : 0.6016
##    Detection Prevalence : 0.7656
##       Balanced Accuracy : 0.6634
##
##        'Positive' Class : 0
##
```

Q.5.h) Use a couple of diï¬erent training samples and check how your decision tree models change. Is your decision tree robust?

```
#Using  couple of  training datasets we saw the accuracy is similar to the
initial accuracy which is 86%. Hence,  the decision tree is robust
```

Q.5.i) Do parts (c), (e), and (f) for a âctreeâ function as well. Are there any signiï¬cant diï¬erences between these decision trees constructed by ctree and rpart?

```
library(party)
```

```
## Loading required package: grid
```

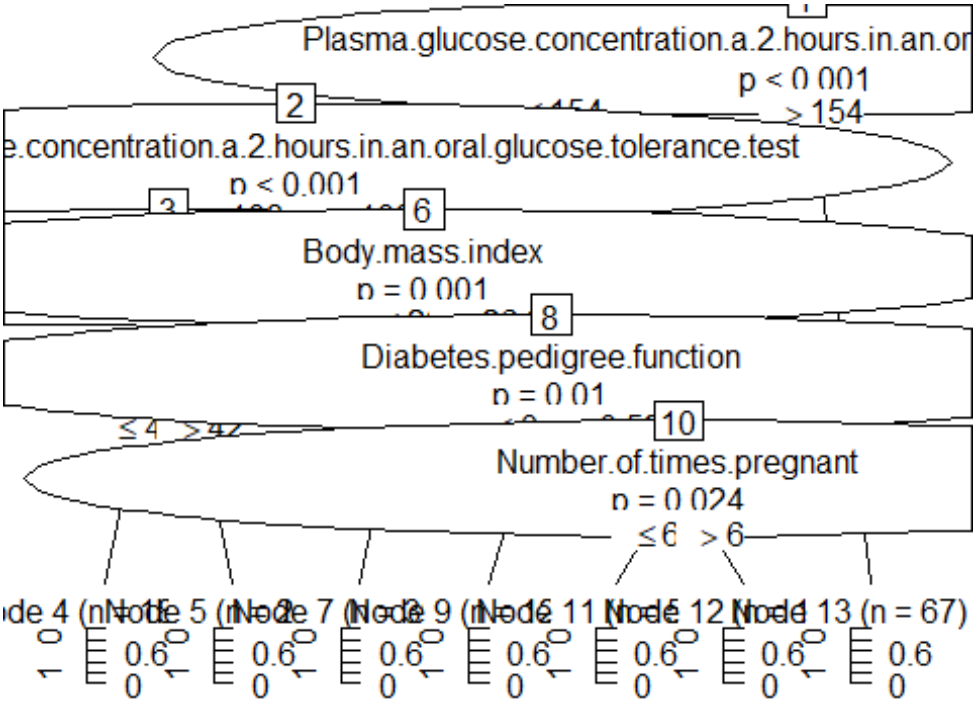
```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'modeltools'
```

```
## The following object is masked from 'package:plyr':
##
##     empty

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

diabetes_ctree<-ctree(Class.variable~.,data = train)
#diabetes_ctree

plot(diabetes_ctree)
```




```
#Predictions on training dataset
diabetes.predicted_train_ctree<- predict(diabetes_ctree,train)

#Confusion matrix for evaluating the model on training dataset
confusionMatrix(diabetes.predicted_train_ctree,train$Class.variable)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 339  91
##          1  11  70
##
##                Accuracy : 0.8004
##                  95% CI : (0.7631, 0.8342)
##     No Information Rate : 0.6849
##     P-Value [Acc > NIR] : 3.340e-09
##
##                   Kappa : 0.4659
##
##  Mcnemar's Test P-Value : 5.192e-15
##
##             Sensitivity : 0.9686
##             Specificity : 0.4348
##          Pos Pred Value : 0.7884
##          Neg Pred Value : 0.8642
##              Prevalence : 0.6849
##          Detection Rate : 0.6634
##    Detection Prevalence : 0.8415
##       Balanced Accuracy : 0.7017
##
##        'Positive' Class : 0
##

# Most important variables are at the top nodes of the tree. Here the most
important nodes are ghluccose concentration, age and body mass index.

#
```