Homework4

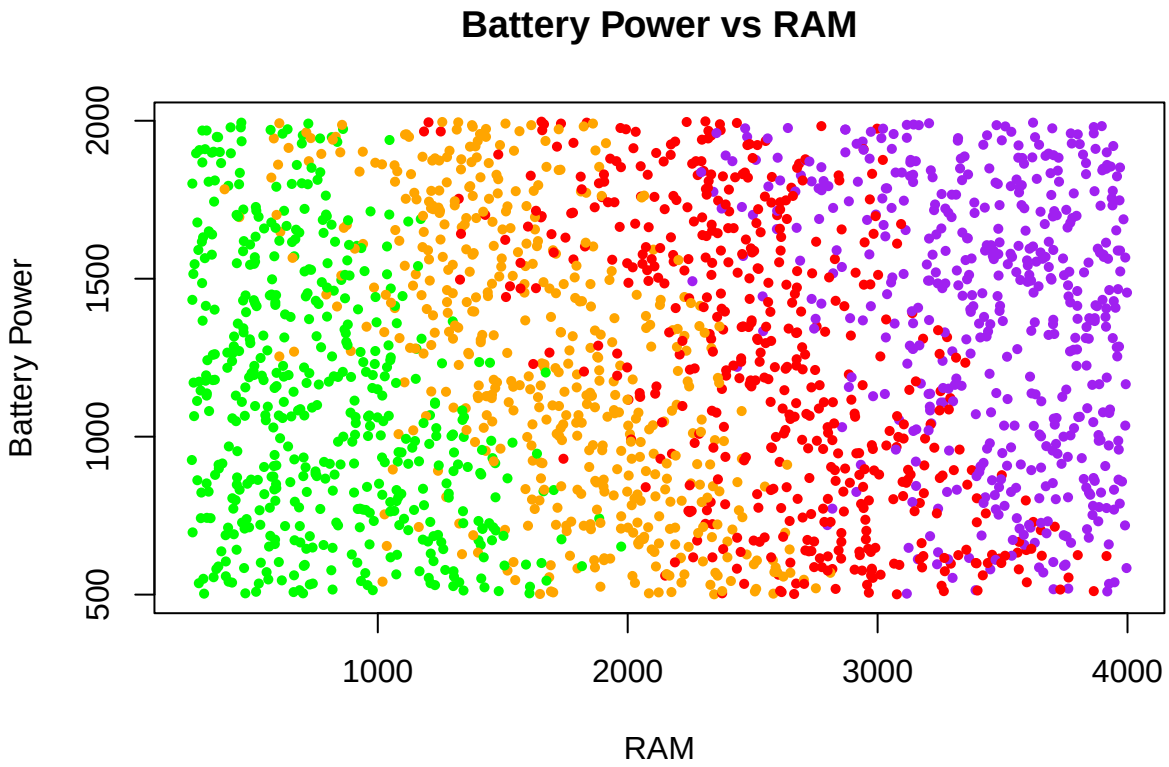
2024-03-20

Question 1

```
setwd("/Users/adi/Desktop/archive")
mobile_data <- read.csv('train.csv',sep=',')
```

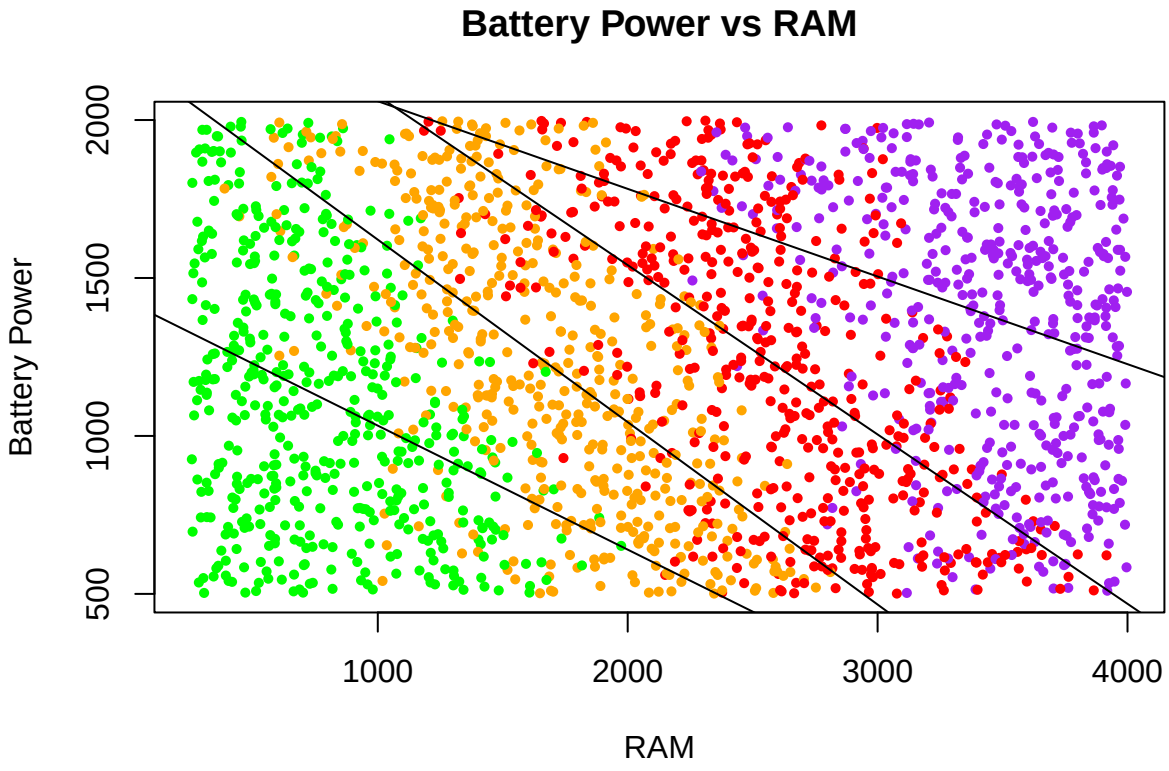
a)

```
mobile_data$price_range <- factor(
                                  mobile_data$price_range, levels = c(
                                    "0","1","2","3"), labels = c(
                                      "low", "medium", "high", "very_high"))

colors <- c("low" = "green", "medium" = "orange", "high" = "red", "very_high" = "purple")
color_code <- colors[mobile_data$price_range]
plot(battery_power ~ ram, data = mobile_data, col = color_code,
     xlab = "RAM", ylab = "Battery Power", pch = 16, cex = 0.7,
     main = "Battery Power vs RAM")
```



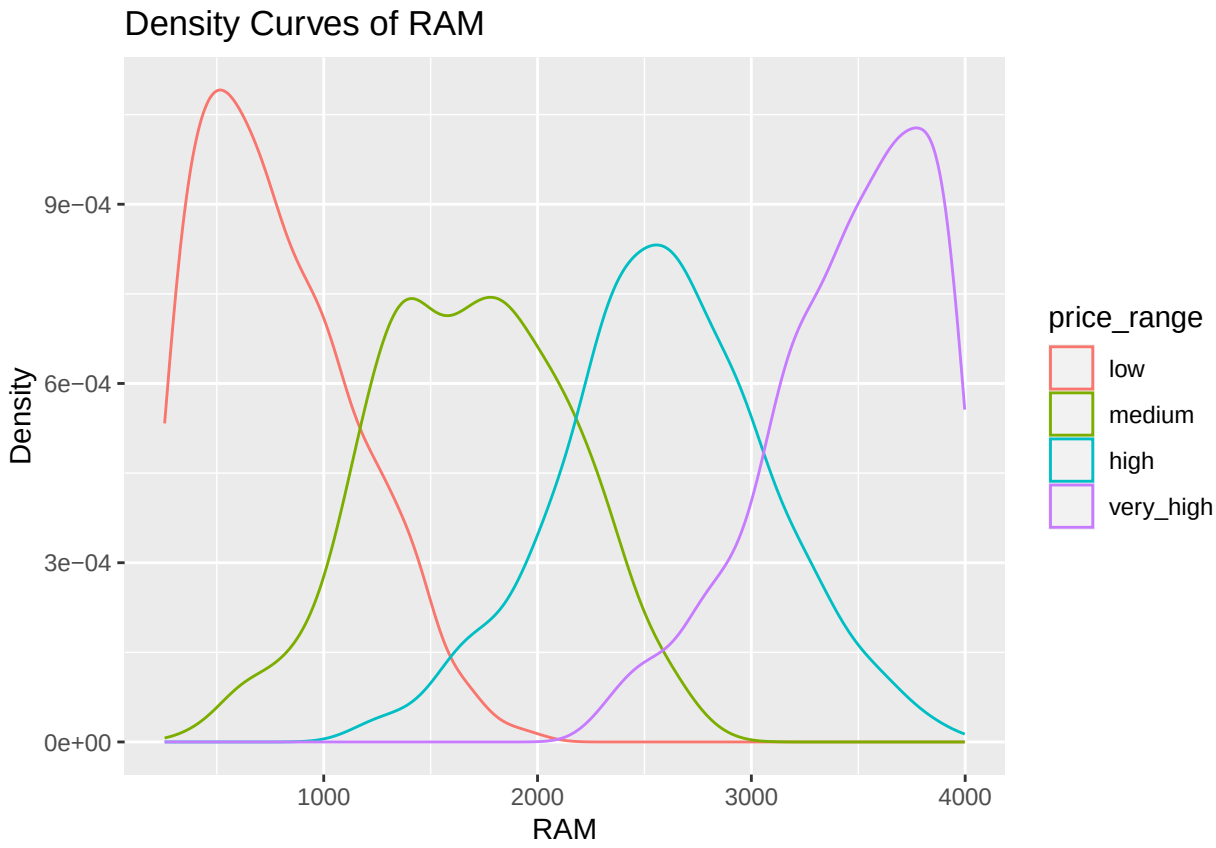
b)

```
priceLow <- mobile_data[which(mobile_data$price_range == "low"), ]
priceMedium <- mobile_data[which(mobile_data$price_range == "medium"), ]
priceHigh <- mobile_data[which(mobile_data$price_range == "high"), ]
priceVeryhigh <- mobile_data[which(mobile_data$price_range == "very_high"), ]

plot(battery_power ~ ram, data = mobile_data, col = color_code,
     pch = 16, cex = 0.7, xlab = "RAM", ylab = "Battery Power",
      main = "Battery Power vs RAM")
m_l <- lm(battery_power ~ ram, data = priceLow)
abline(a = coef(m_l)[1], b = coef(m_l)[2])
m_m <- lm(battery_power ~ ram, data = priceMedium)
abline(a = coef(m_m)[1], b = coef(m_m)[2])
m_h <- lm(battery_power ~ ram, data = priceHigh)
abline(a = coef(m_h)[1], b = coef(m_h)[2])
m_vh <- lm(battery_power ~ ram, data = priceVeryhigh)
abline(a = coef(m_vh)[1], b = coef(m_vh)[2])
```



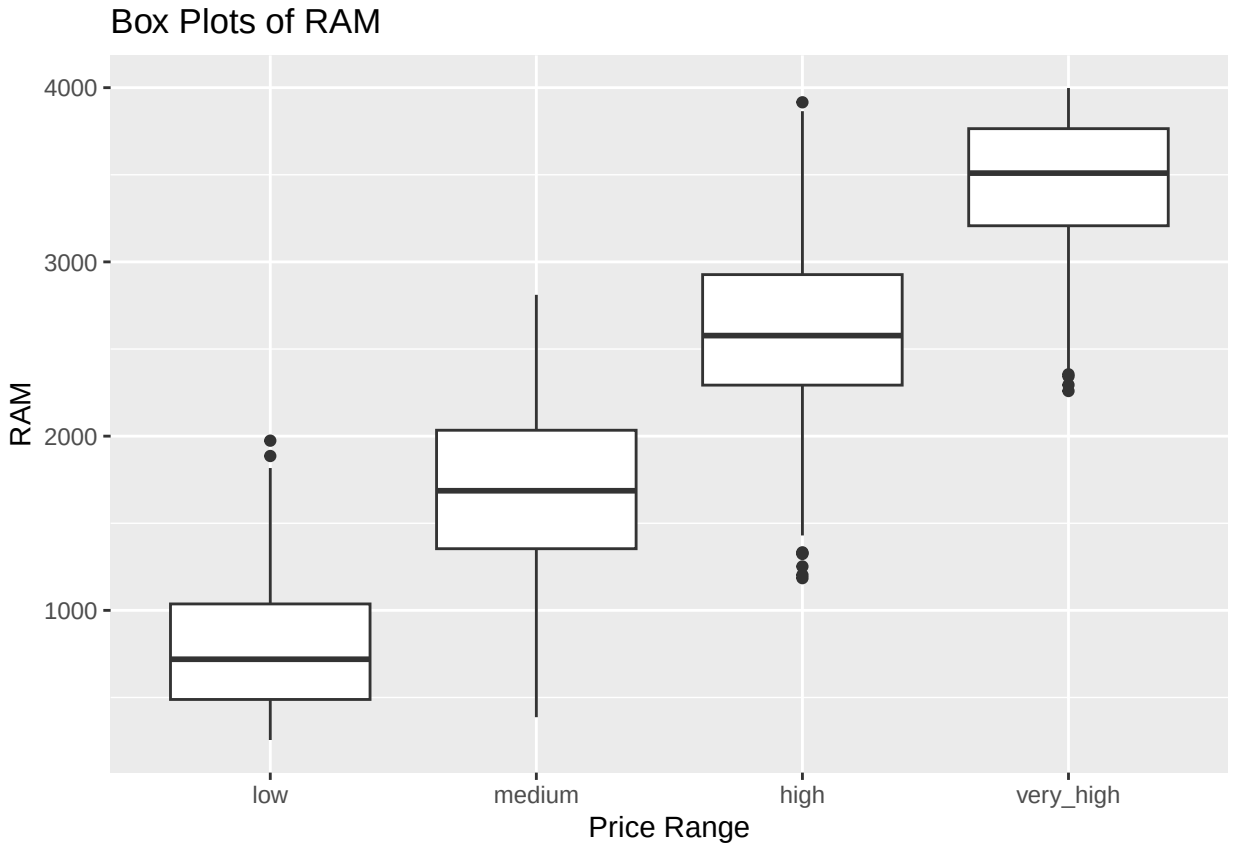
c)

```
library(ggplot2)
ggplot(data = mobile_data) +
geom_density(mapping = aes(
  x = ram, color = price_range)) + labs(
    x = "RAM", y = "Density", title = "Density Curves of RAM")
```



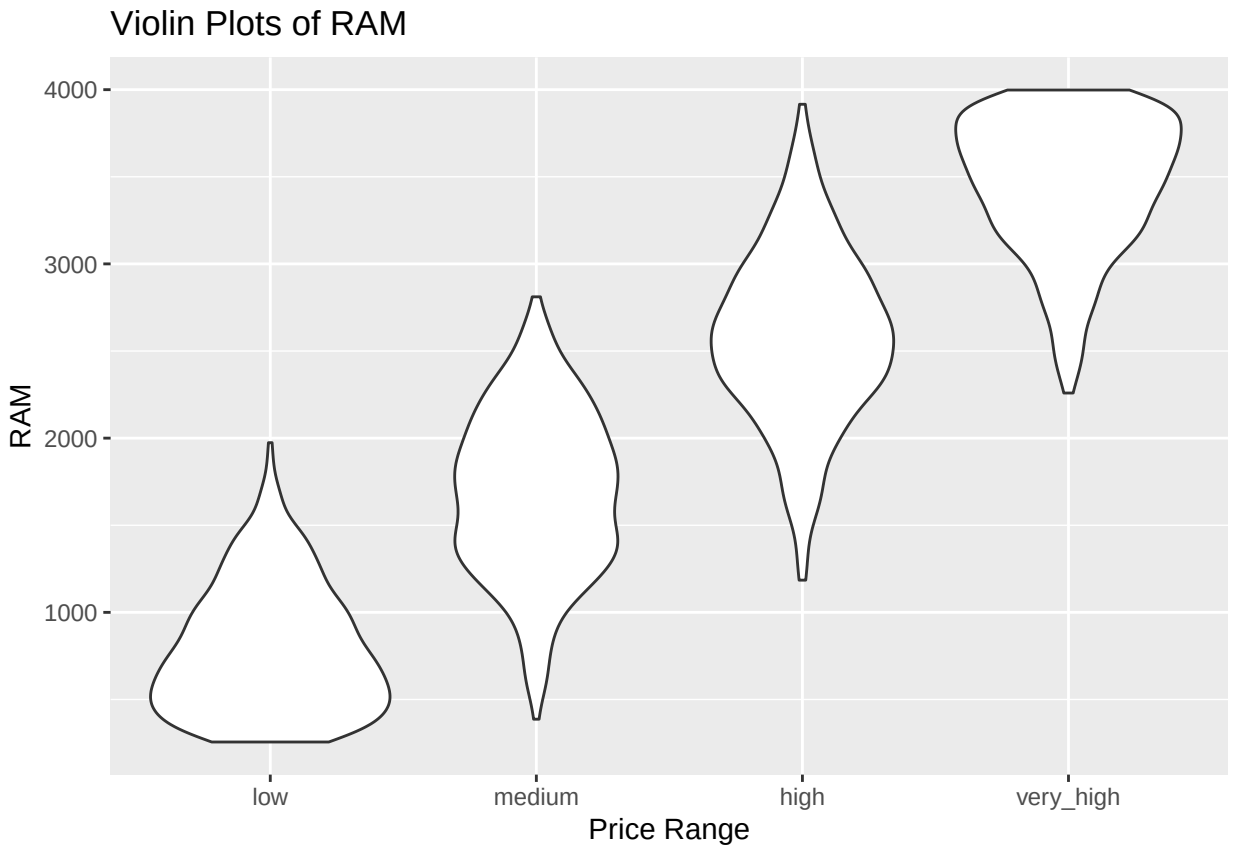
d)

```
library(ggplot2)
ggplot(data = mobile_data) +
geom_boxplot(mapping = aes(
  x = price_range, y = ram)) + labs(
    x = "Price Range", y = "RAM", title = "Box Plots of RAM")
```



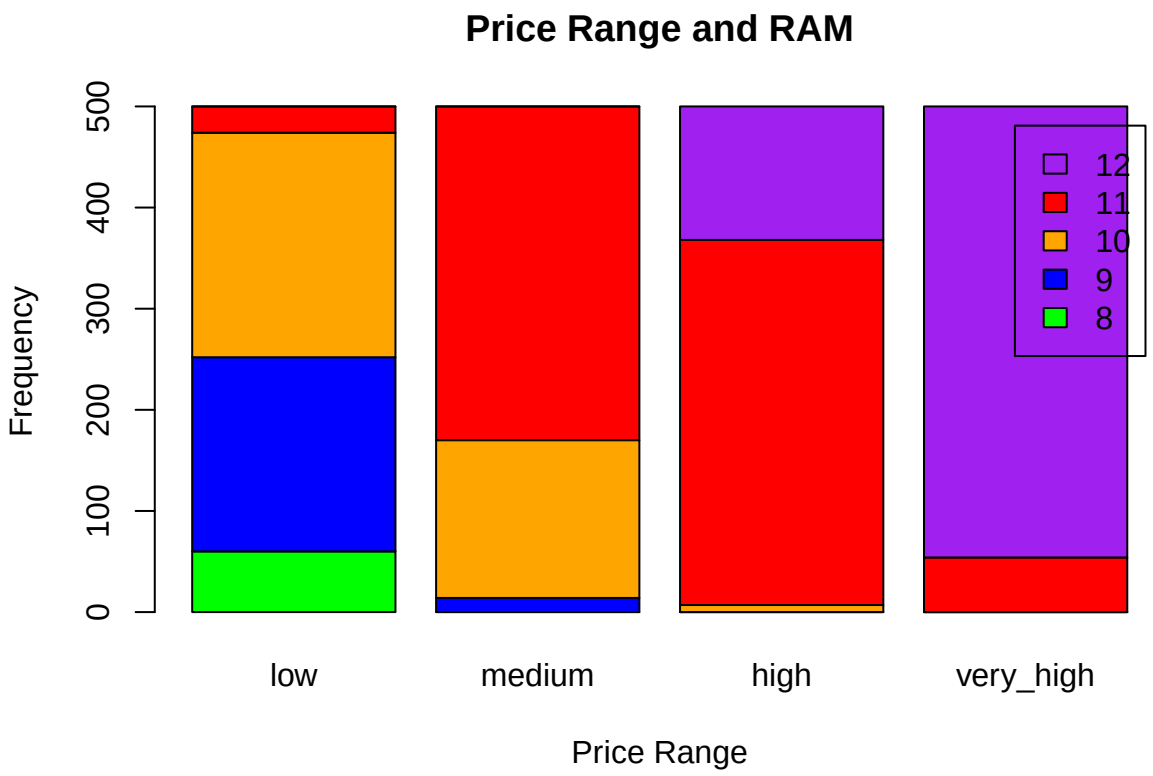
e)

```
library(ggplot2)
ggplot(data = mobile_data) +
geom_violin(mapping = aes(
  x = price_range, y = ram)) + labs(
    x = "Price Range", y = "RAM", title = "Violin Plots of RAM")
```



f)

```
for (i in 1:length(mobile_data$ram)) {
  mobile_data$ram[i] <- round(log2(mobile_data$ram[i]))
}
mobile_data$ram <- factor(mobile_data$ram)

F <- table(mobile_data$ram, mobile_data$price_range)
barplot(F, ylab = "Frequency", xlab = "Price Range", legend.text = TRUE,
        main = "Price Range and RAM", col = c("green", "blue", "orange", "red", "purple"))
```



Question 2

```
library(UsingR)
```

```
## Loading required package: MASS
## Loading required package: HistData
## Loading required package: Hmisc
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
data (UScereal)
cereal <- UScereal
```

a)

```
levels(cereal$mfr) <- c("General Mills", "Kellogs","Nabisco",
                        "Post", "Quaker Oats", "Ralston Purina")
```

b)

```
cereal$shelf <- factor(cereal$shelf)
levels(cereal$shelf) = c("low", "medium", "upper")
```

c)

```
cereal$product <- rownames(cereal)
```

d)

```
#Pearson correlation for protein
proteinPC <- cor(cereal$calories, cereal$protein, method = "pearson")
proteinPC
```

```
## [1] 0.7060105
```

```
#Fat
fatPC <- cor(cereal$calories, cereal$fat, method = "pearson")
fatPC
```

```
## [1] 0.5901757
```

```
#Sodium
sodiumPC <- cor(cereal$calories, cereal$sodium, method = "pearson")
sodiumPC
```

```
## [1] 0.5286552
```

```
#Fibre
fibrePC <- cor(cereal$calories, cereal$fibre, method = "pearson")
fibrePC
```

```
## [1] 0.3882179
```

```
#Carbo
carboPC <- cor(cereal$calories, cereal$carbo, method = "pearson")
carboPC
```

```
## [1] 0.7887227
```

```
#Sugars
sugarsPC <- cor(cereal$calories, cereal$sugars, method = "pearson")
sugarsPC
```

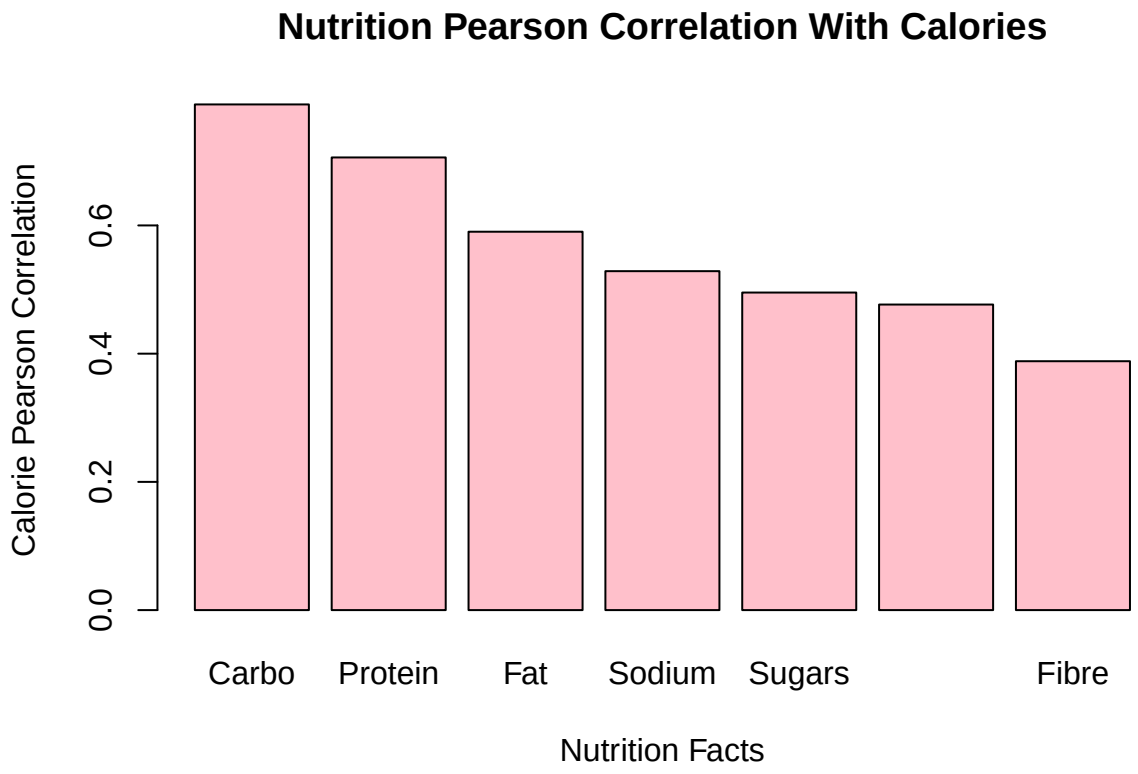
```
## [1] 0.4952942
```

```
#Potassium
potassiumPC <- cor(cereal$calories, cereal$potassium, method = "pearson")
potassiumPC
```

```
## [1] 0.4765955
```

e)

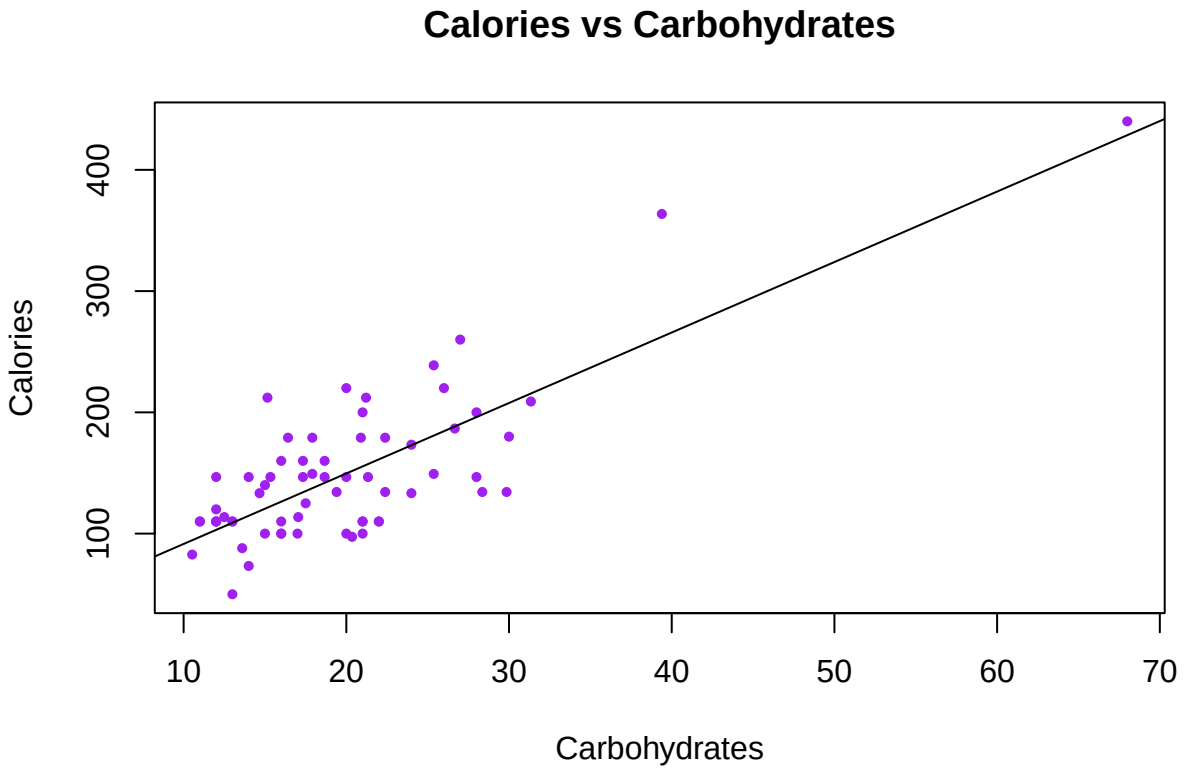
```
nutriPC <- c(proteinPC, fatPC, sodiumPC, fibrePC, carboPC, sugarsPC, potassiumPC)
names(nutriPC) <- c("Protein", "Fat", "Sodium", "Fibre", "Carbo", "Sugars", "Potassium")
nutriPC <- sort(nutriPC, decreasing = TRUE)
barplot(nutriPC, main = "Nutrition Pearson Correlation With Calories", xlab = "Nutrition Facts", ylab =
```



Carbohydrates have the highest values.

f)

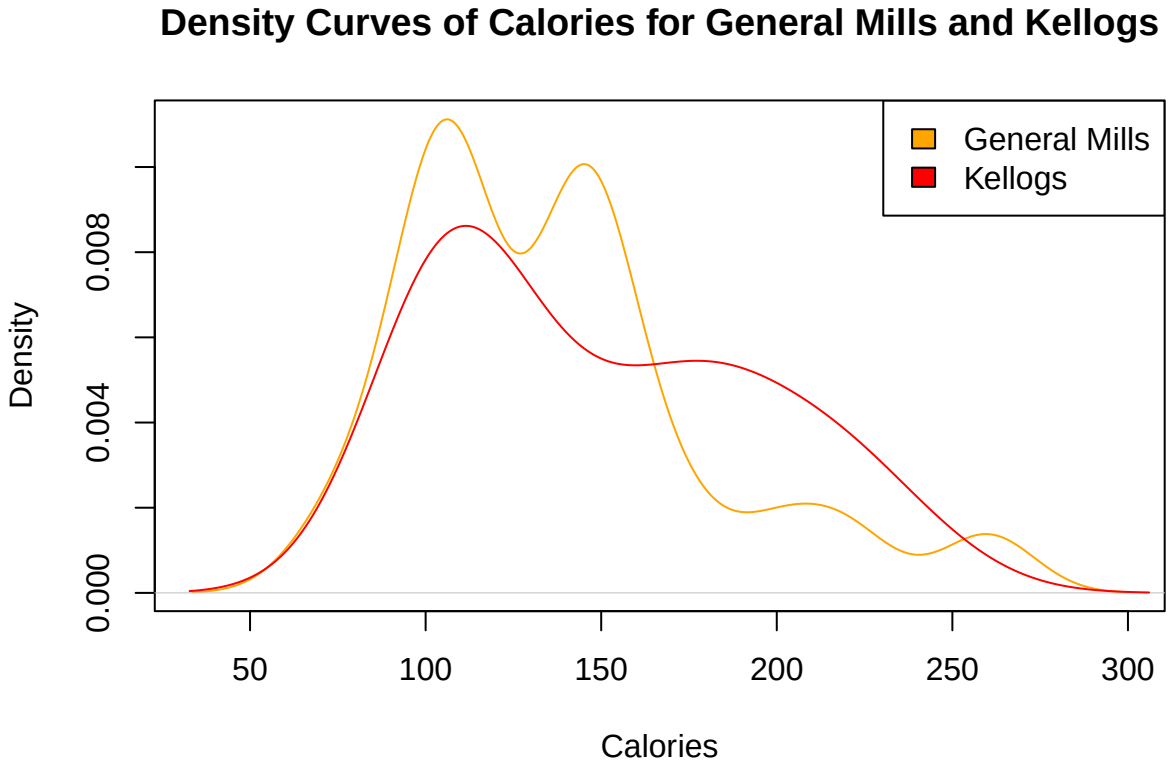
```
plot(calories ~ carbo, data = cereal,
     xlab = "Carbohydrates", col = "purple",
     ylab = "Calories",
     pch = 16, cex = 0.7, main = "Calories vs Carbohydrates")
m <- lm(calories ~ carbo, data = cereal)
abline(a = coef(m)[1], b = coef(m)[2])
```



The slope indicates the rate at which the calories in a cereal increases as the carbohydrate content increases. The intercept indicates the expected value of calories in a cereal with zero carbohydrate content.

g)

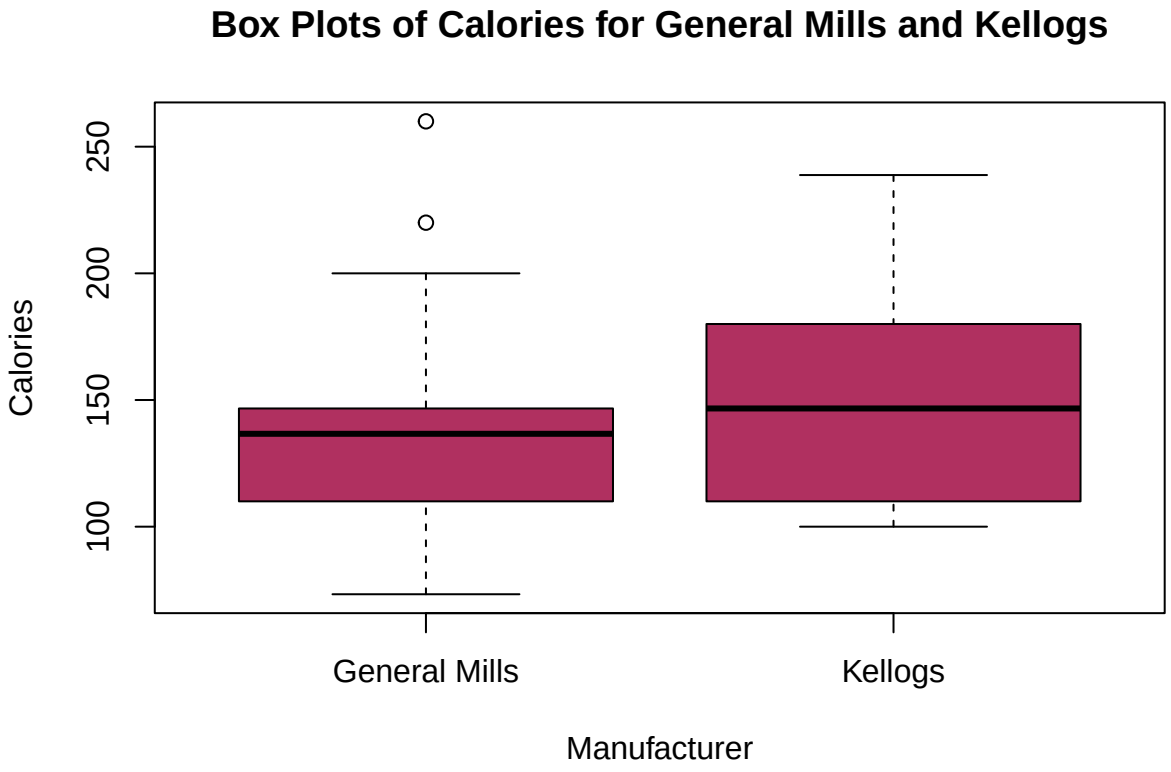
```
GM_cal <- cereal$calories[cereal$mfr == "General Mills"]
Kellogs_cal <- cereal$calories[cereal$mfr == "Kellogs"]

plot(density(GM_cal), col = "orange",
     main = "Density Curves of Calories for General Mills and Kellogs",
     ylab = "Density", xlab = "Calories")
lines(density(Kellogs_cal), col = "red")
legend(x = "topright", legend = c(
       "General Mills", "Kellogs"),
       fill = c("orange", "red"))
```



The density curve of calories for General Mills is bimodal and skewed-right. The density curve of calories for Kellogs is skewed-right.

h)

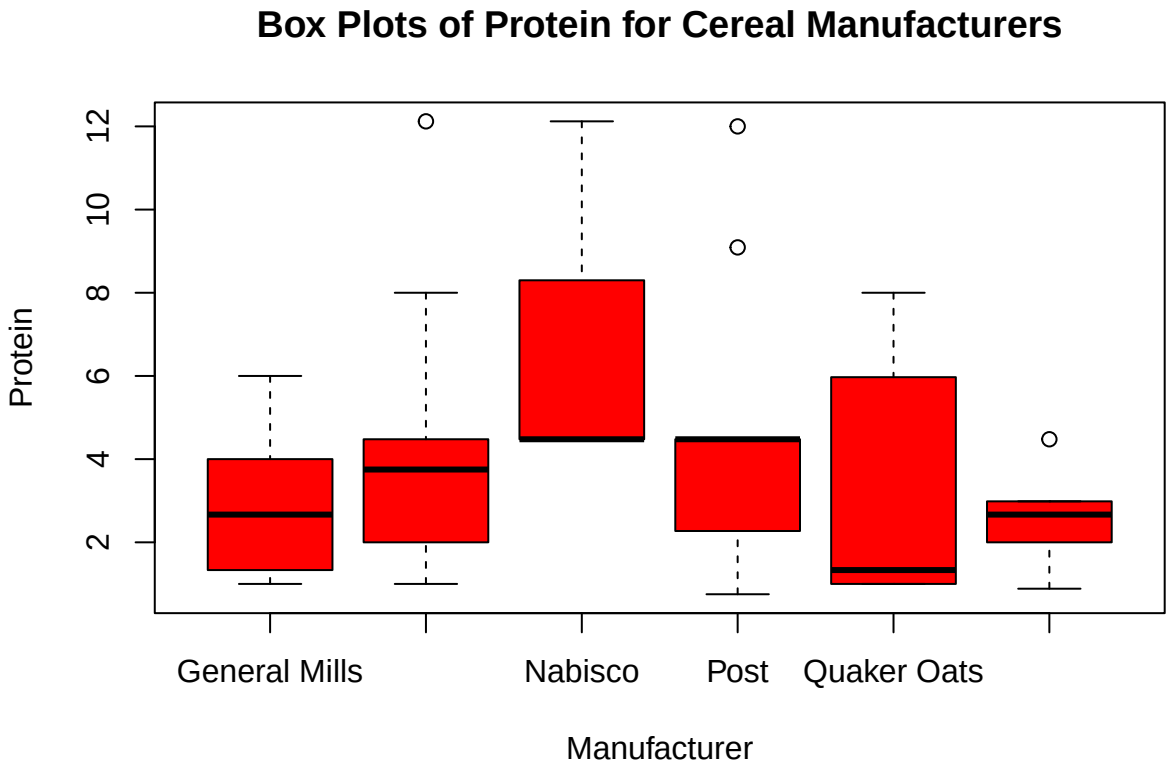
```
mfr <- c()
calories <- c()
for (i in 1:length(cereal$calories)) {
if (cereal$mfr[i] %in% c("Kellogs", "General Mills")) {
  mfr <- c(mfr, cereal$mfr[i])
  calories <- c(calories, cereal$calories[i])
} }
mfr <- factor(mfr, labels = c("General Mills", "Kellogs"))

boxplot(calories ~ mfr, col = "maroon", ylab = "Calories",
        xlab = "Manufacturer",
        main = "Box Plots of Calories for General Mills and Kellogs")
```



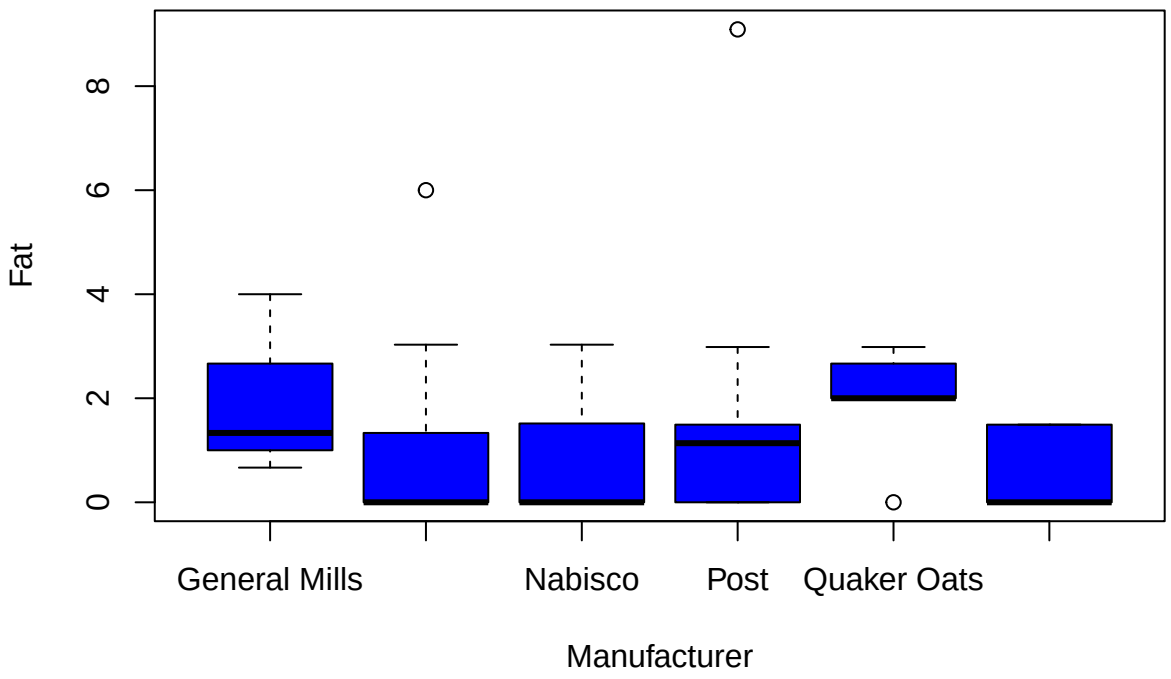
From the boxplot, we can see that the median calorie content of General Mills is almost equal to Kellogs. The calories of General Mills and Kellogs are slightly different, but it is not a significant difference.

i)

```
levels(cereal$mfr) <- c("General Mills", "Kellogs", "Nabisco",
                        "Post", "Quaker Oats", "Ralston")

#Protein
boxplot(protein ~ mfr, data = cereal, xlab = "Manufacturer",
        ylab = "Protein", main = "Box Plots of Protein for Cereal Manufacturers", col = "red")
```

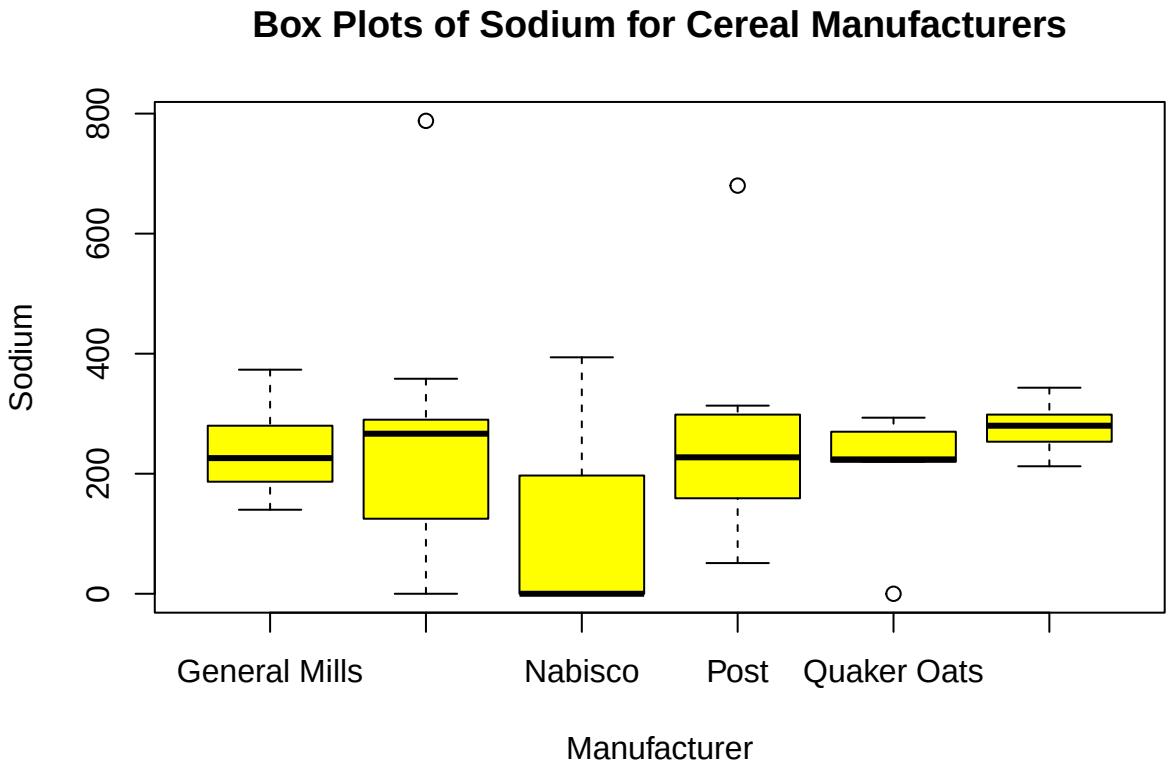



```
#Fat
boxplot(fat ~ mfr, data = cereal, xlab = "Manufacturer",
        ylab = "Fat", main = "Box Plots of Fat for Cereal Manufacturers", col = "blue")
```

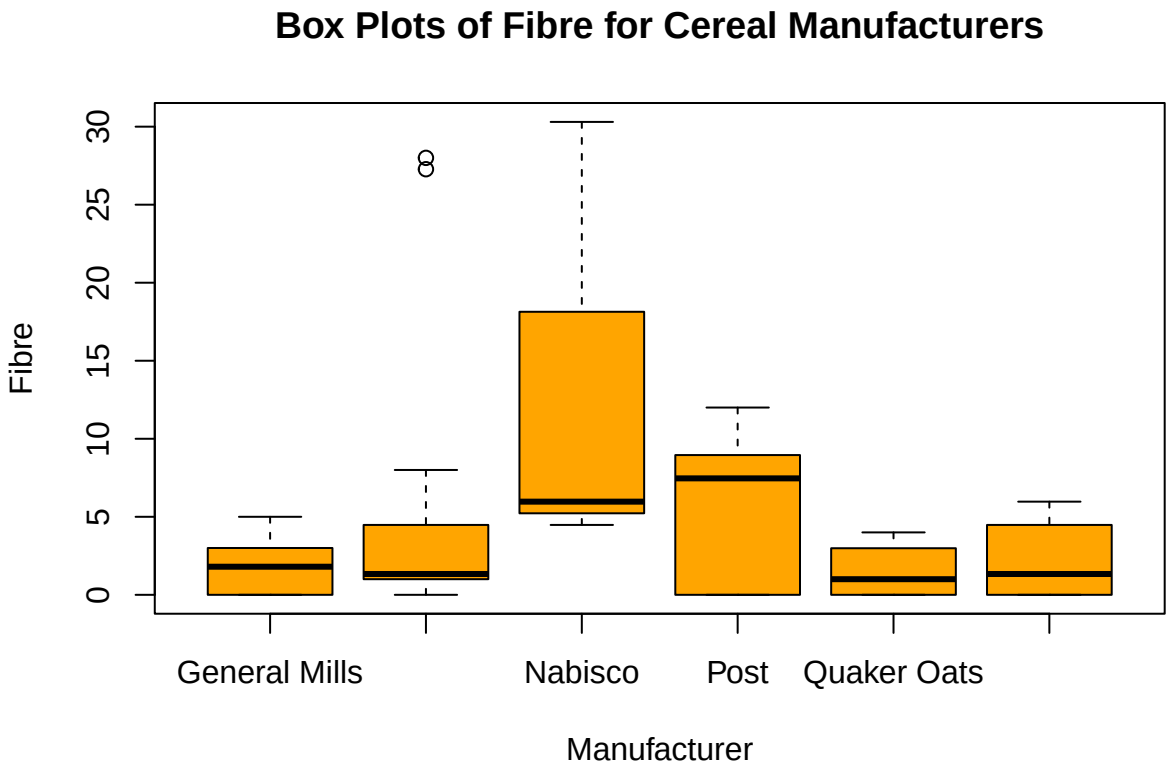



```
#Sodium
boxplot(sodium ~ mfr, data = cereal, xlab = "Manufacturer",
```

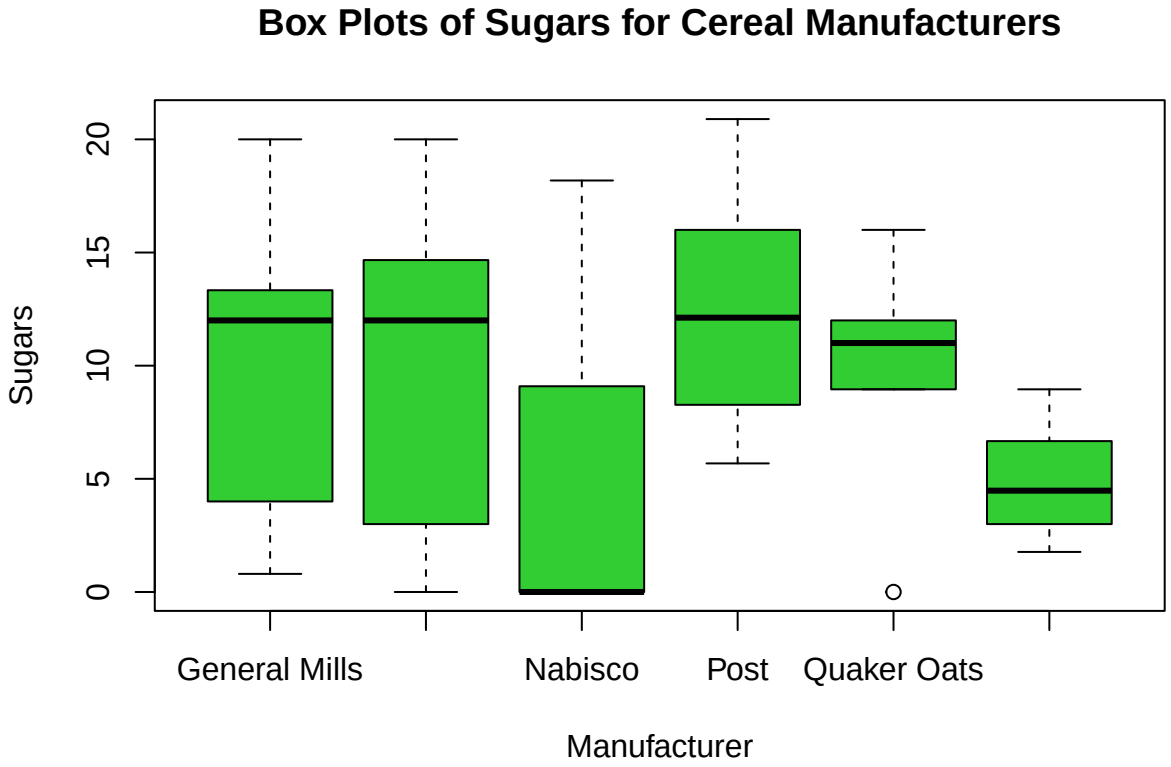
```
        ylab = "Sodium", main = "Box Plots of Sodium for Cereal Manufacturers", col = "yellow")
```



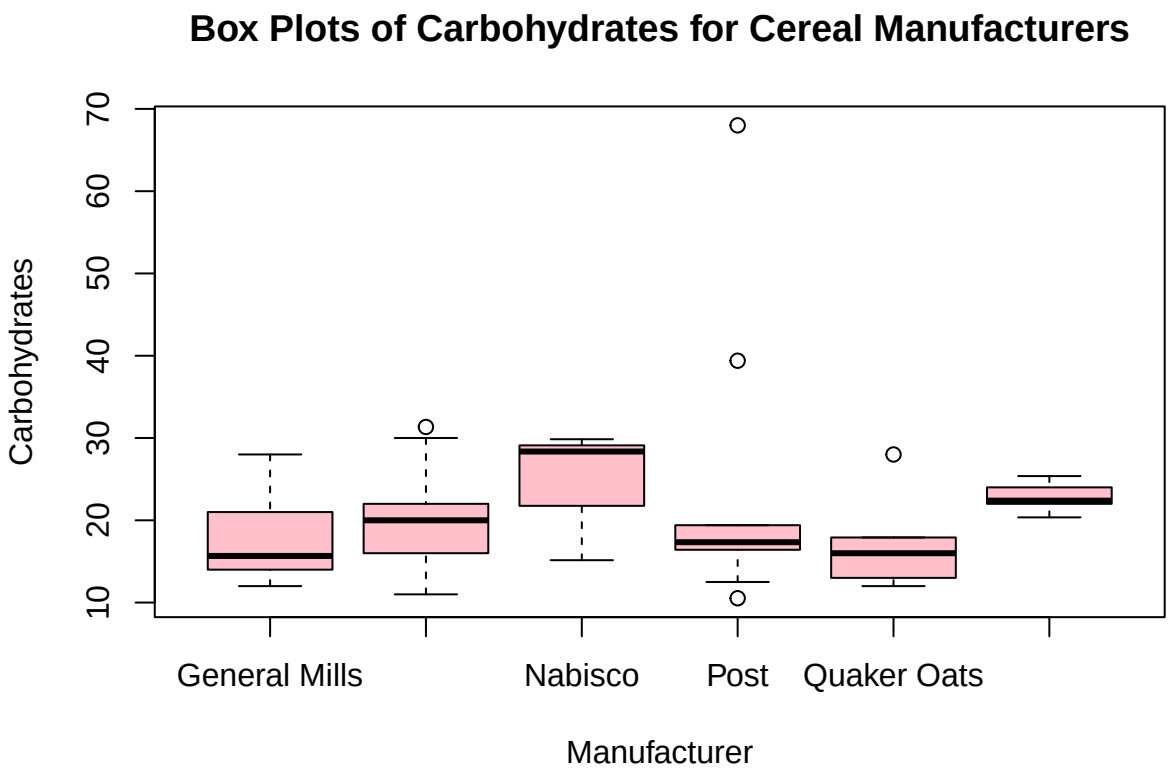

```
#Fibre
boxplot(fibre ~ mfr, data = cereal, xlab = "Manufacturer",
        ylab = "Fibre", main = "Box Plots of Fibre for Cereal Manufacturers", col = "orange")
```



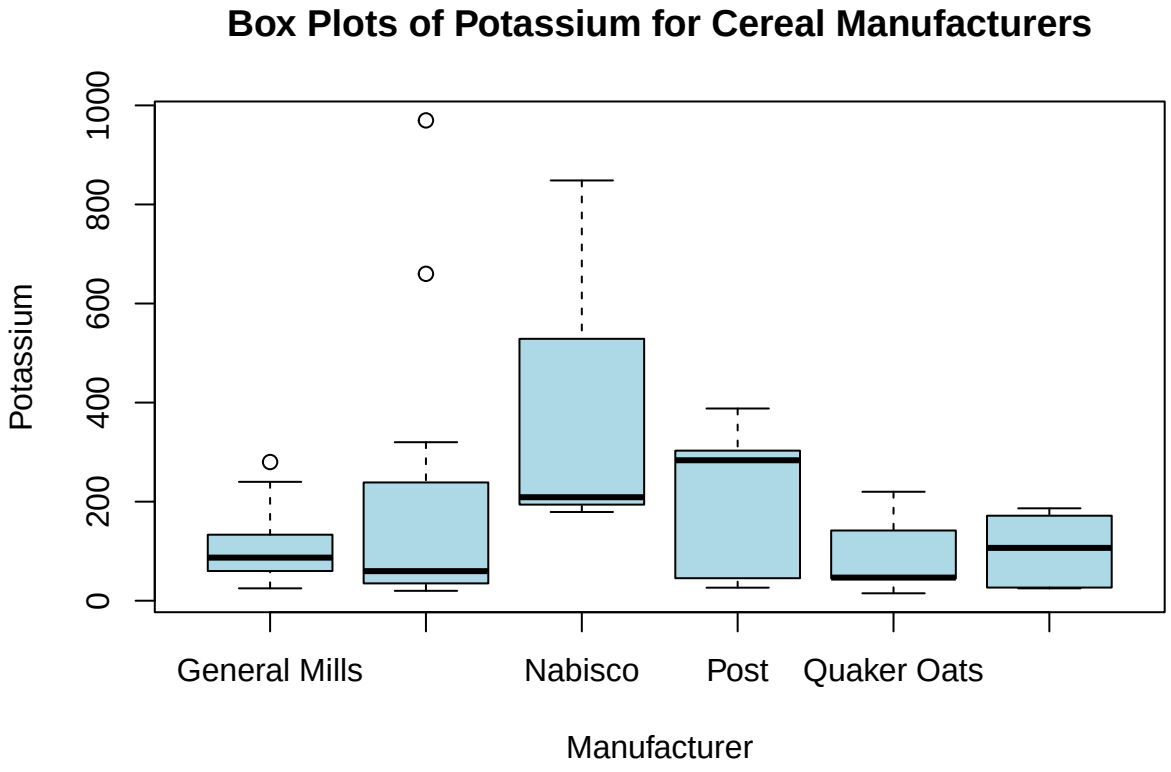
```r
#Sugars
boxplot(sugars ~ mfr, data = cereal, xlab = "Manufacturer",
        ylab = "Sugars", main = "Box Plots of Sugars for Cereal Manufacturers", col = "limegreen")
```




```r
#Carbohydrates
boxplot(carbo ~ mfr, data = cereal, xlab = "Manufacturer",
        ylab = "Carbohydrates", main = "Box Plots of Carbohydrates for Cereal Manufacturers", col = "pi
```

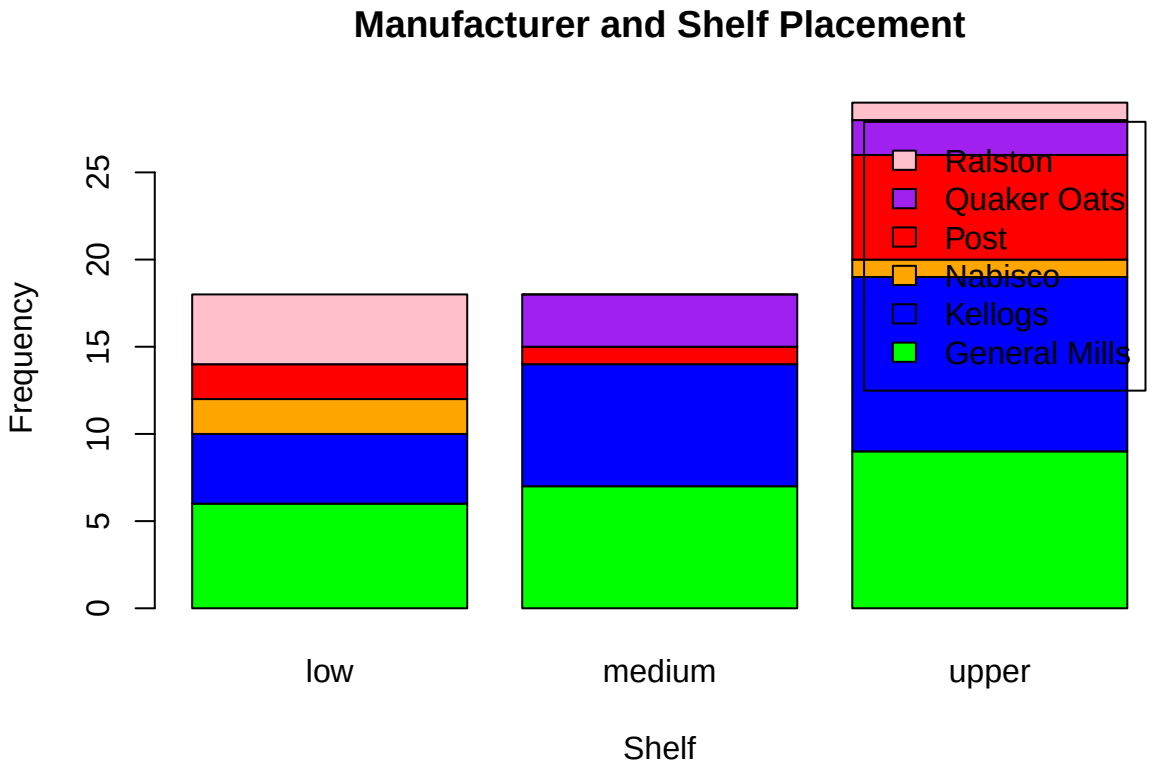



```
#Potassium
boxplot(potassium ~ mfr, data = cereal, xlab = "Manufacturer",
        ylab = "Potassium", main = "Box Plots of Potassium for Cereal Manufacturers", col = "lightblue")
```



j)

```
F <- table(cereal$mfr, cereal$shelf)
barplot(F, ylab = "Frequency", xlab = "Shelf", legend.text = TRUE,
        main = "Manufacturer and Shelf Placement", col = c(
          "green", "blue", "orange", "red", "purple", "pink"))
```

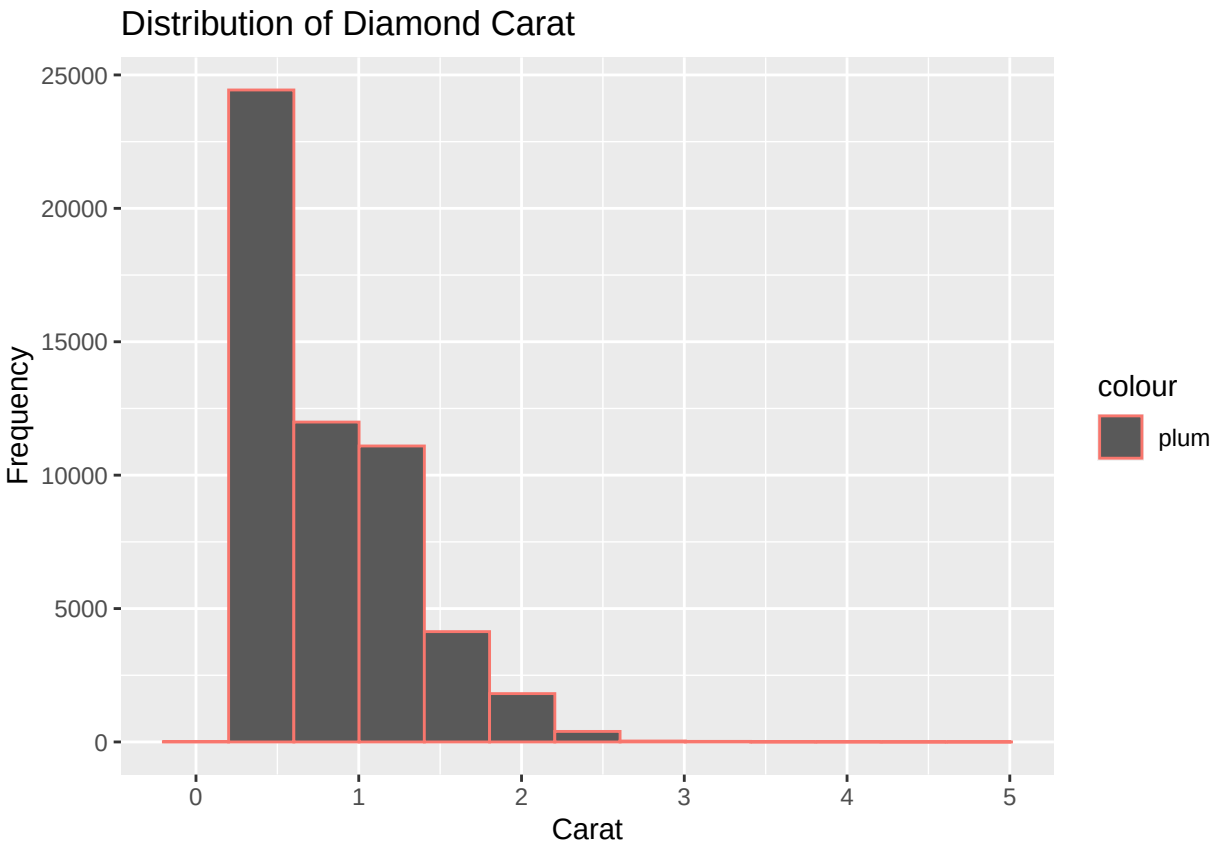


Question 3

```
library(ggplot2)
data (diamonds)
diamond <- diamonds
```

a)

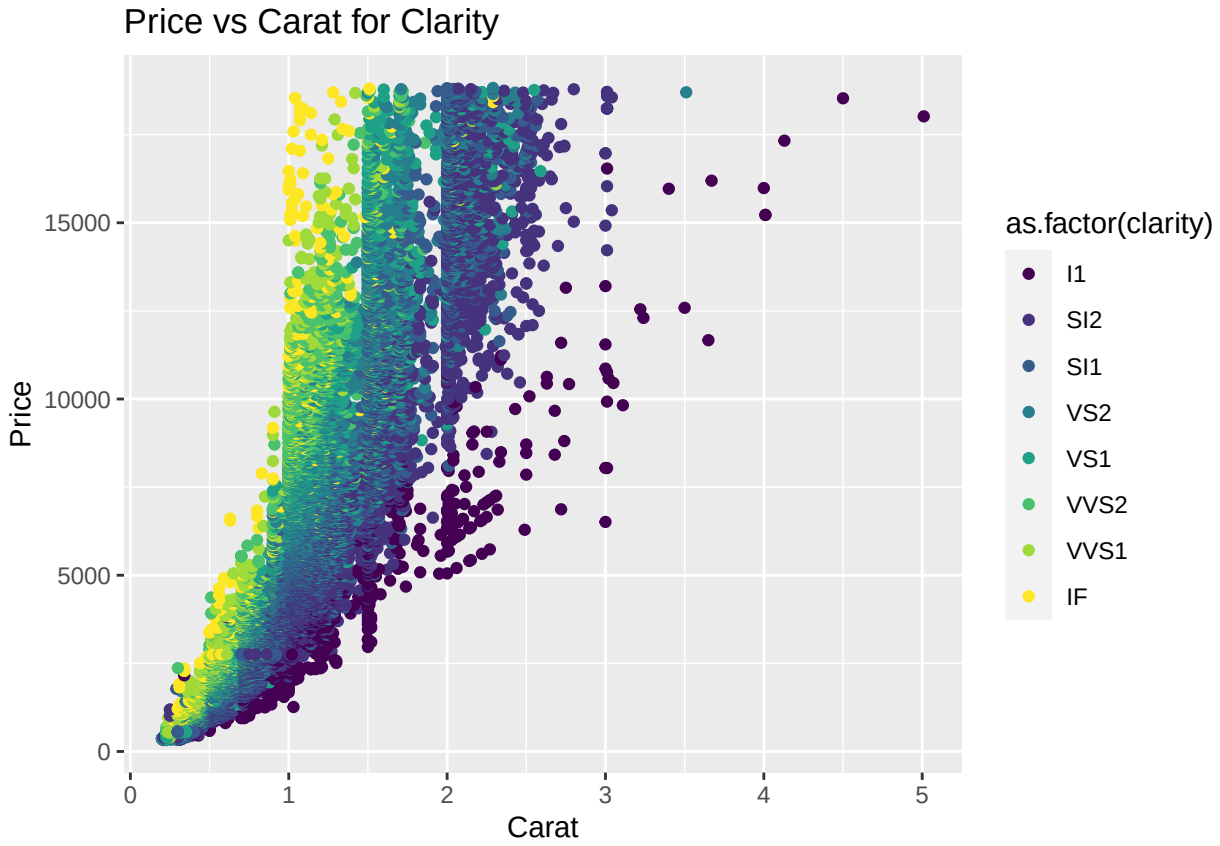
```
ggplot(data = diamond) + geom_histogram(
                         mapping = aes(x = carat, col = "plum"), bins = 13) + labs(
                         x = "Carat", y = "Frequency", title = "Distribution of Diamond Carat")
```



The histogram shows the frequency distribution of diamond carat (weight) for the diamonds in the dataset.

b)

```
ggplot(data = diamond) + geom_point(mapping = aes(x = carat, y = price, color = as.factor(clarity))) +
                                    x = "Carat", y = "Price",title = "Price vs Carat for Clarity")
```

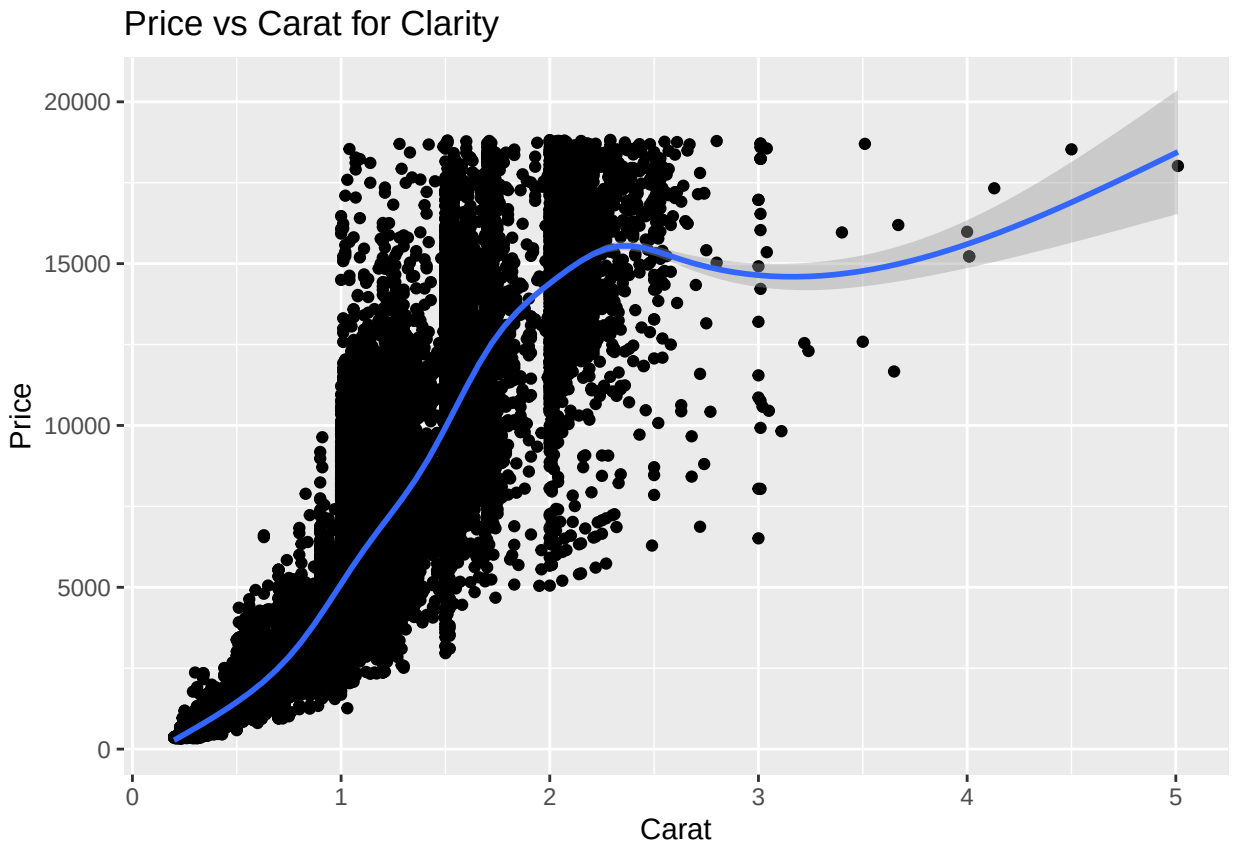


The scatterplot shows the relationship between the weight of the diamond (carat) and the price, based on the the quality of the clarity of each diamond. It can be seen that lighter diamonds are more expensive if they have a have a better clarity rating and heavier diamonds are more expensive if they have a lower clarity rating.

c)

```
ggplot(data = diamond) + geom_point(mapping = aes(x = carat, y = price)) + geom_smooth(
                                    mapping = aes(x = carat, y = price)) +
                                    labs(x = "Carat", y = "Price",title = "Price vs Carat for Clarity")
```

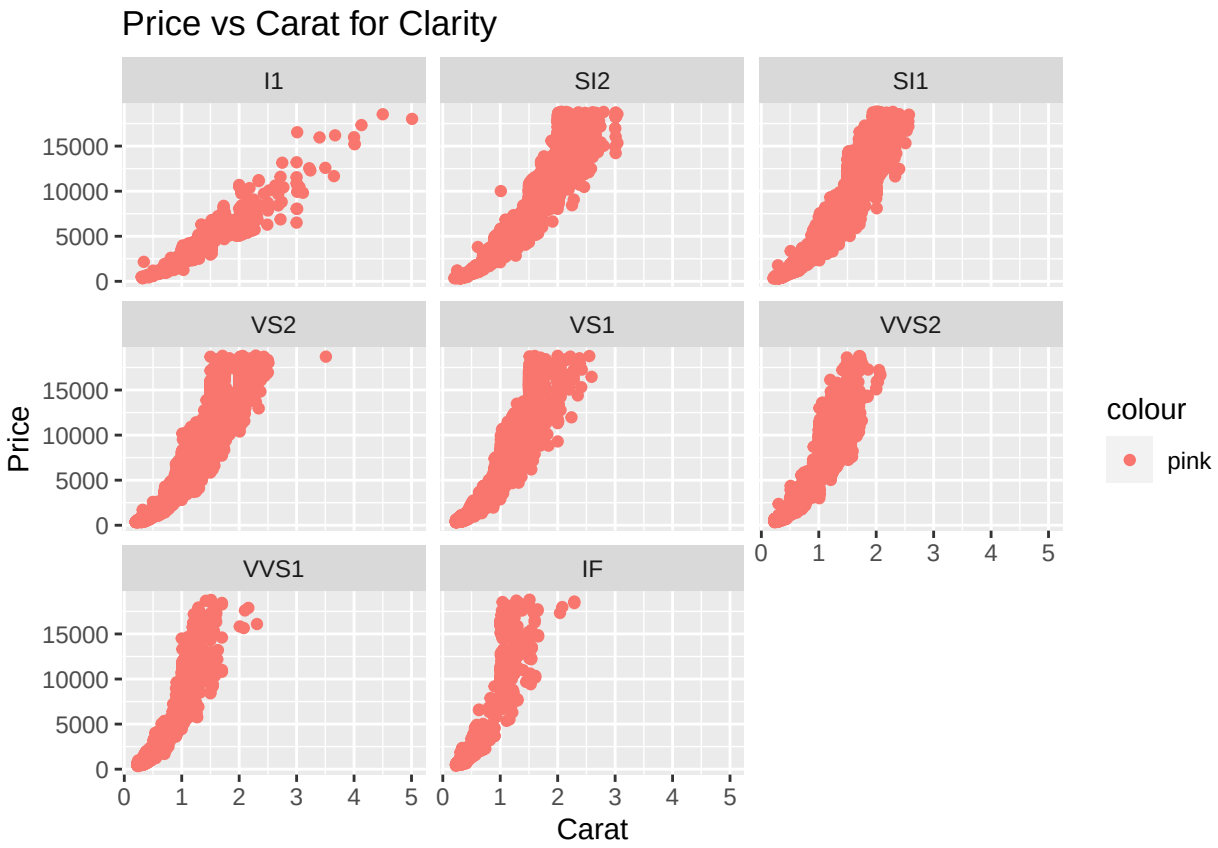
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



The scatterplot consists of a smooth curve that shows the general trend of the relationship between weight and price of the diamonds.

d)

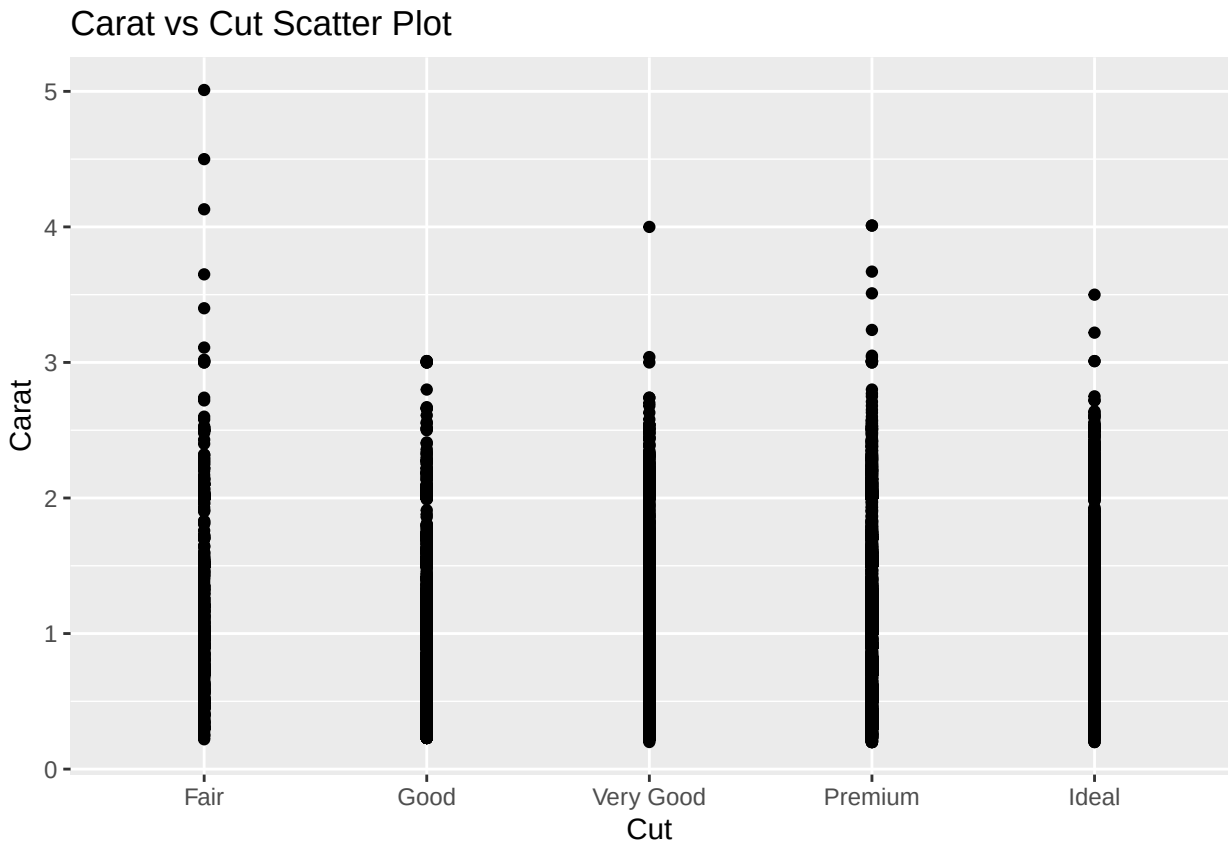
```
 ggplot(data = diamond) + geom_point(mapping = aes(x = carat, y = price, col = "pink")) + facet_wrap(~cl
labs(x = "Carat", y = "Price", title = "Price vs Carat for Clarity")
```



The scatterplot is faceted into eight subplots, each of which shows the relationship between weight and price of the diamonds for a particular clarity level.

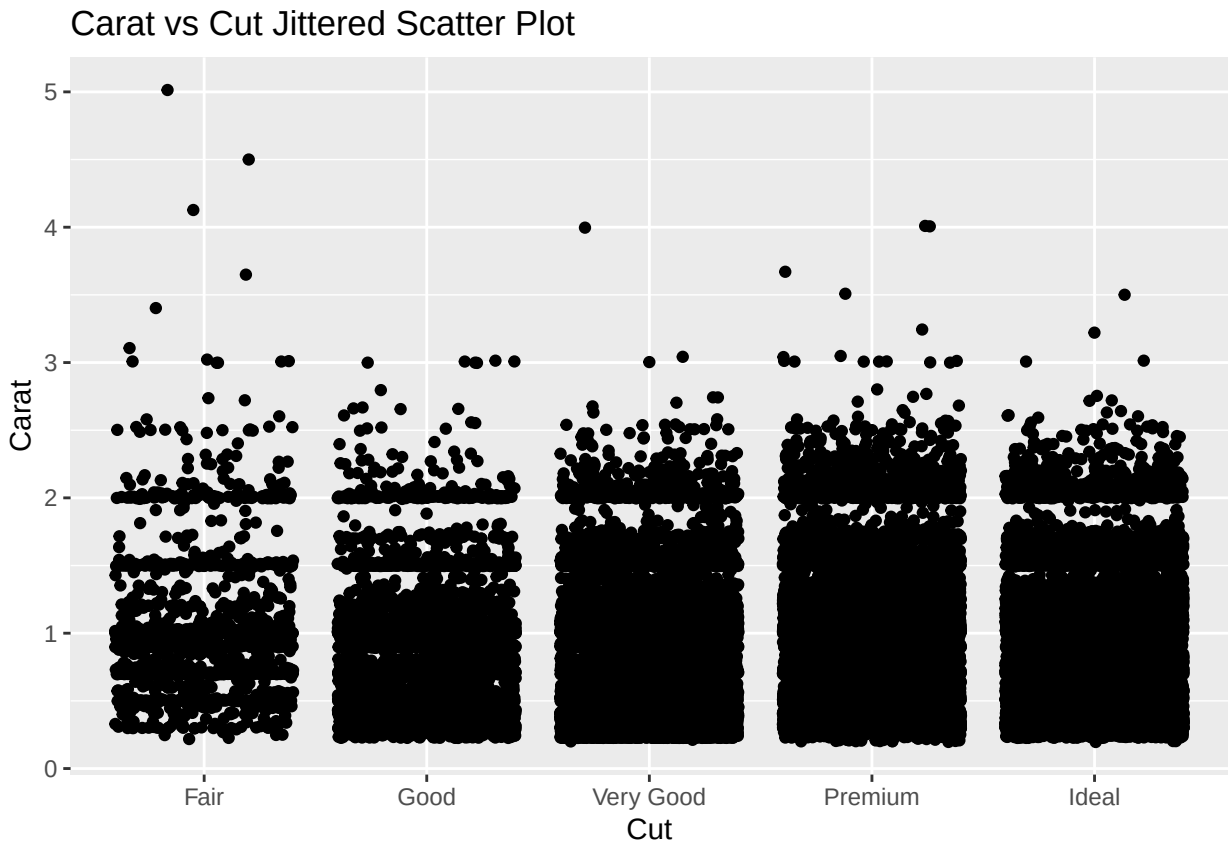
e)

```
ggplot(data = diamond) + geom_point(mapping = aes(x = cut, y = carat)) + labs(x = "Cut", y = "Carat",
                                    title = "Carat vs Cut Scatter Plot")
```



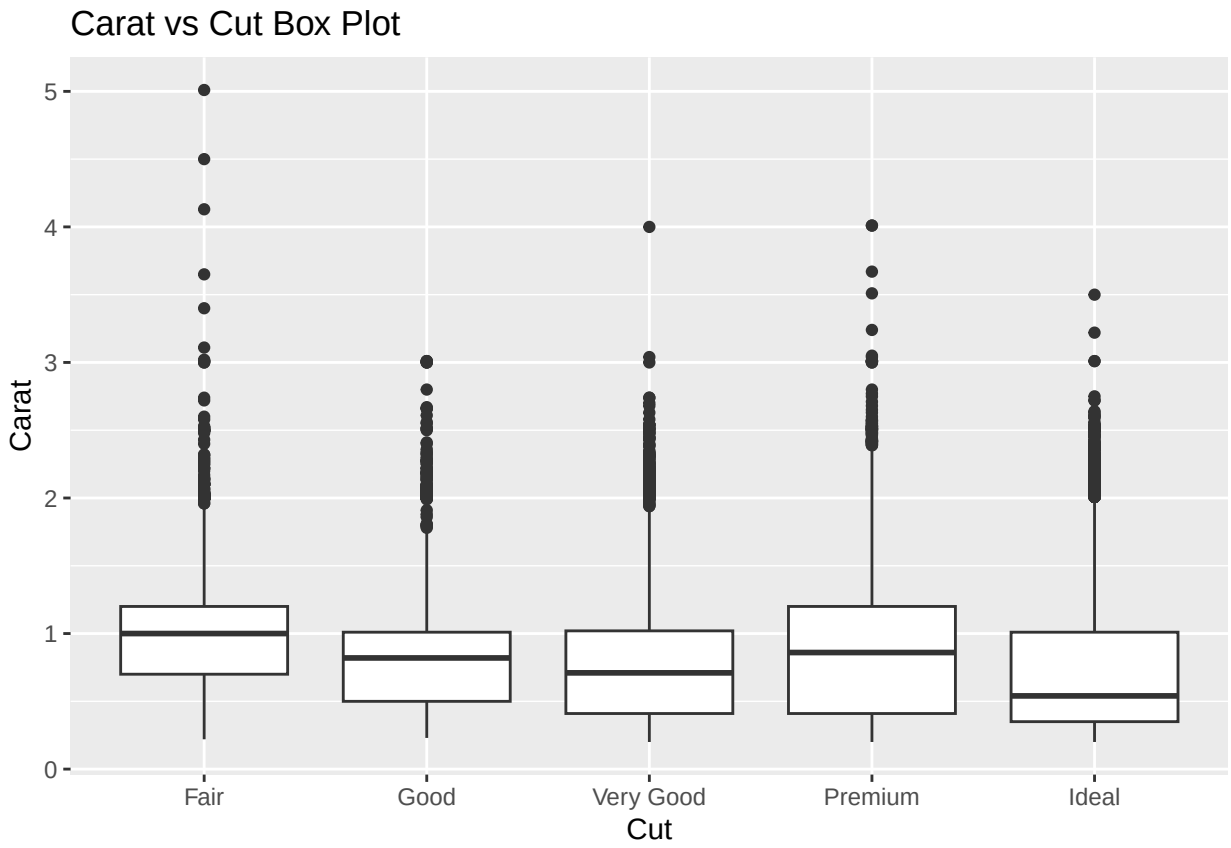
The scatterplot shows the relationship between carat and cut of the diamonds.

```
ggplot(data = diamond) + geom_jitter(mapping = aes(x = cut, y = carat)) + labs(x = "Cut", y = "Carat",
                         title = "Carat vs Cut Jittered Scatter Plot")
```



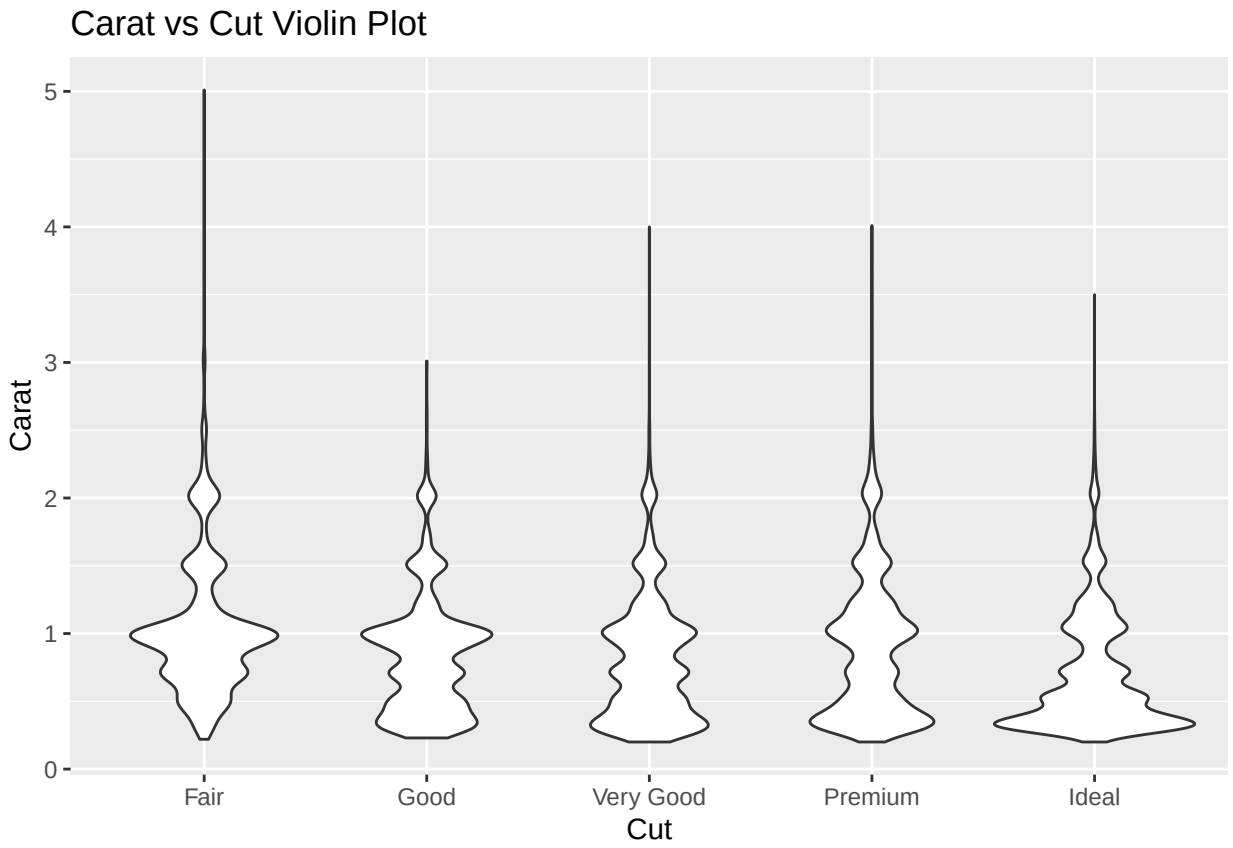
The jittered scatterplot show the relationship between carat and cut but adds some jitter to avoid overlapping of points.

```
ggplot(data = diamond) + geom_boxplot(mapping = aes(x = cut, y = carat)) + labs(x = "Cut", y = "Carat",
                                      title = "Carat vs Cut Box Plot")
```



The boxplot shows the distribution of carat at each level of cut for the diamonds in the dataset.

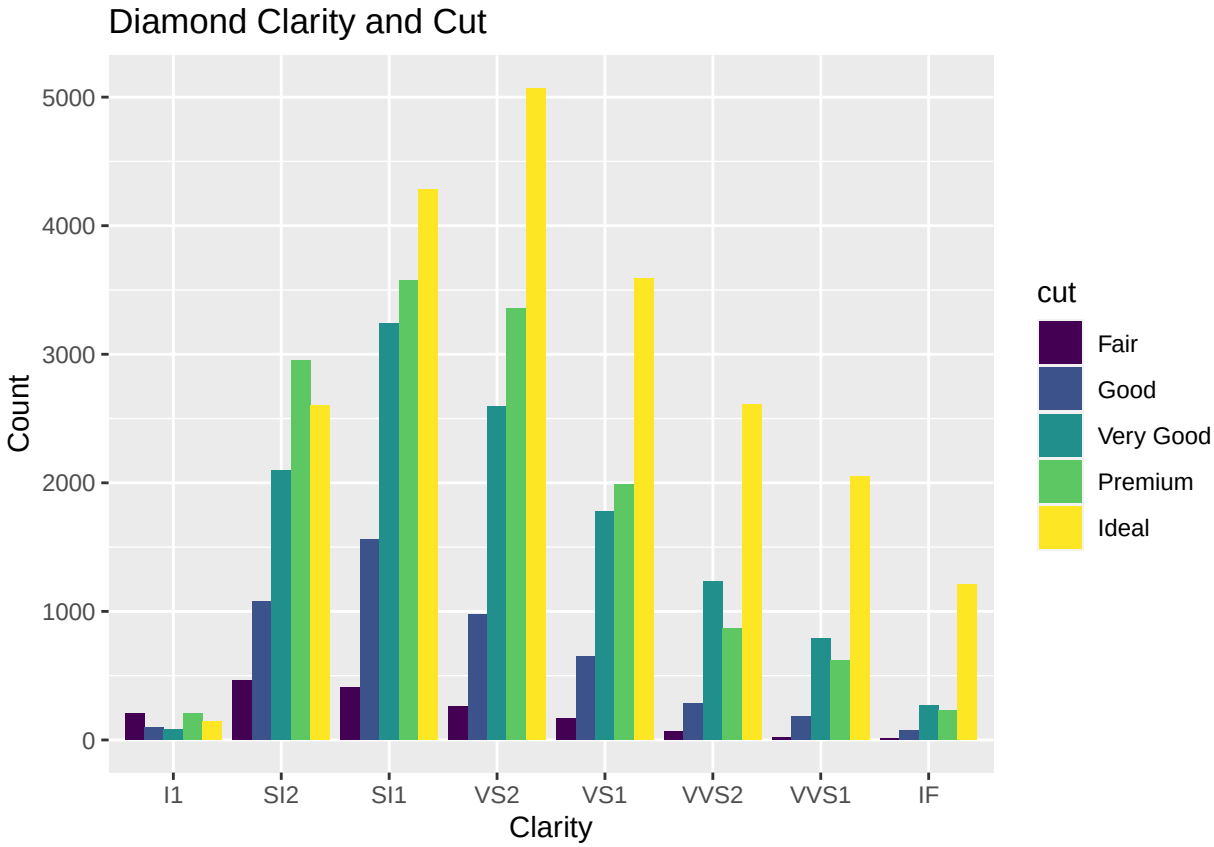
```
ggplot(data = diamond) + geom_violin(mapping = aes(x = cut, y = carat)) + labs(x = "Cut", y = "Carat",
                                     title = "Carat vs Cut Violin Plot")
```



The violin plot also shows the distribution of carat at each level of cut for the diamonds in the dataset. The best plot for visualization in this case is the boxplot as it clearly shows us the distribution of the numeric variable (carat) within different levels of the categorical variable (cut) along with useful information like the median, variability (IQR) and potential outliers.

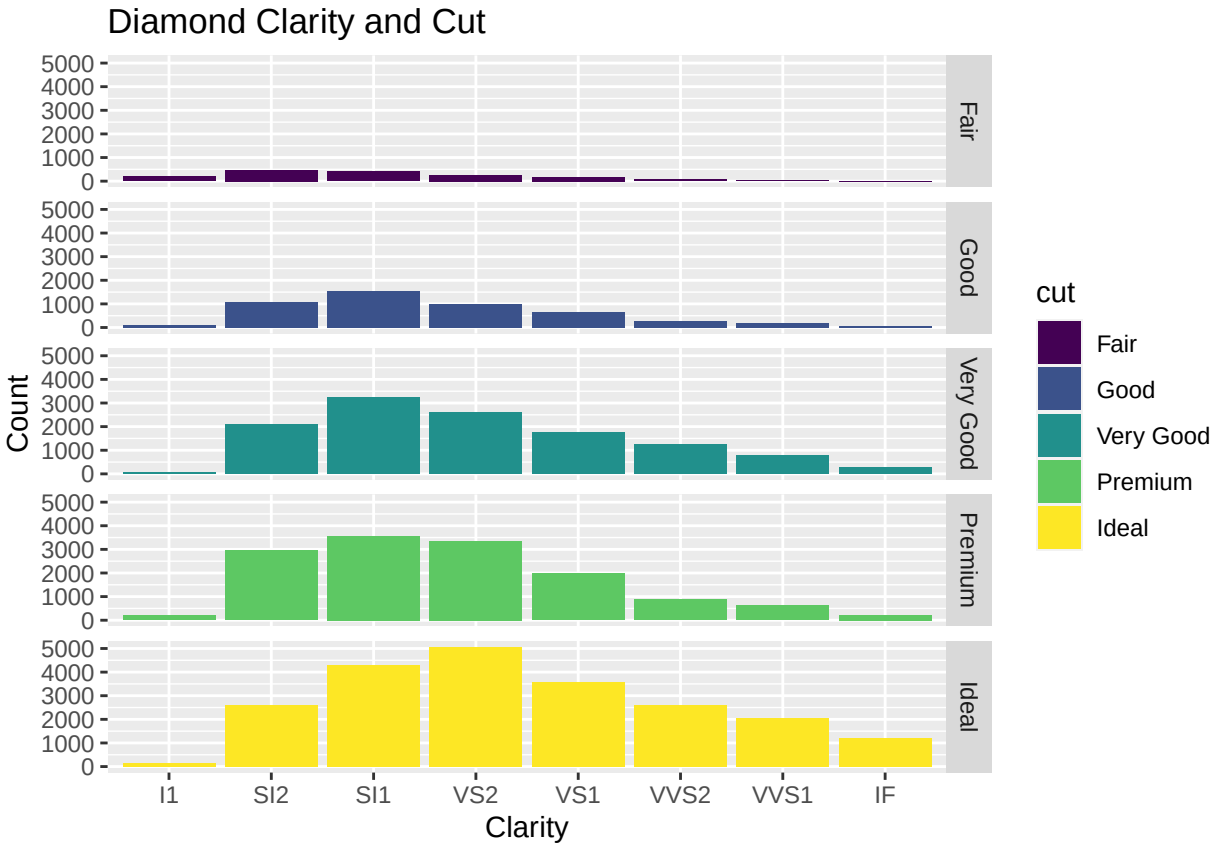
f)

```
ggplot(data = diamond) + geom_bar(mapping = aes(x = clarity, fill = cut), position = "dodge") +
labs(x = "Clarity", y = "Count", title = "Diamond Clarity and Cut")
```



The side-by-side bar plot shows the number of diamonds of each cut quality at a certain clarity level.

```
ggplot(data = diamond) + geom_bar(mapping = aes(x = clarity, fill = cut)) + facet_grid(cut ~.) +
labs(x = "Clarity", y = "Count", title = "Diamond Clarity and Cut")
```

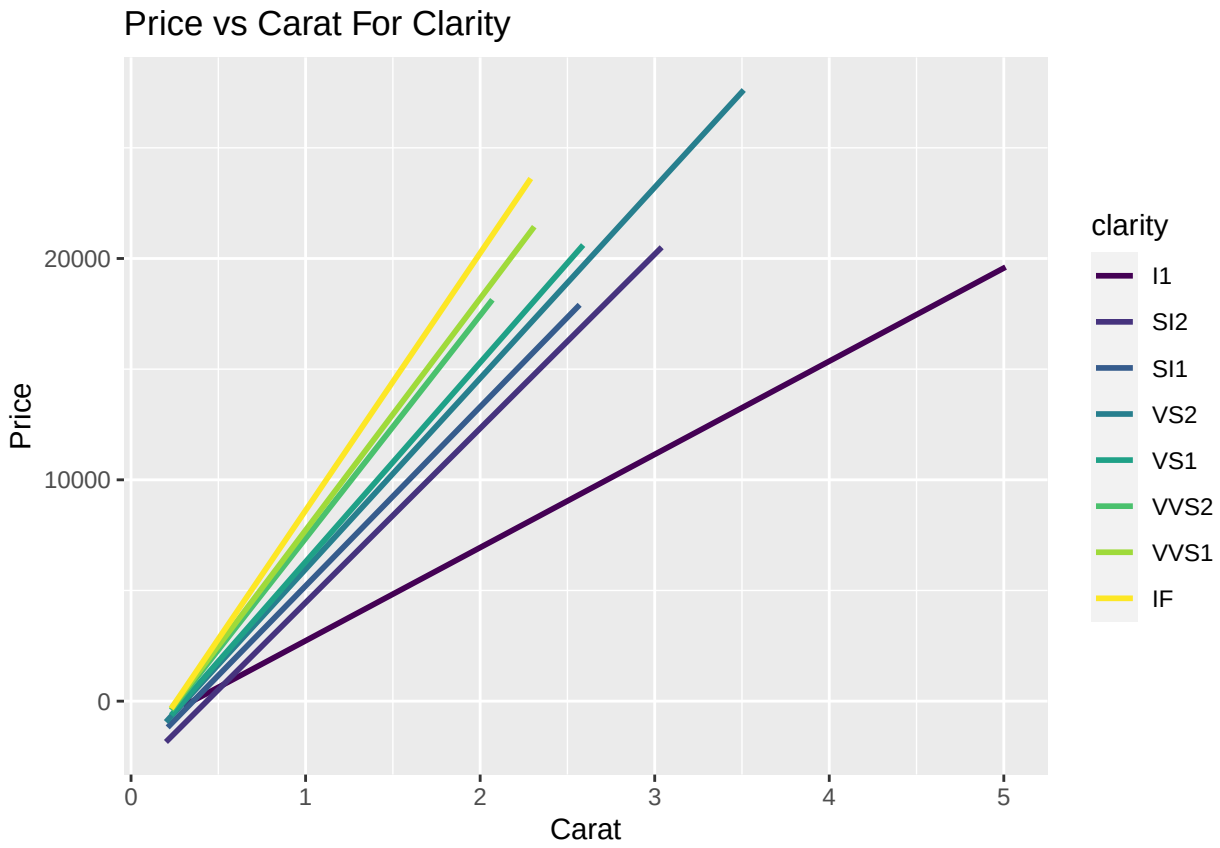


The faceted bar plot shows the number of diamonds of each cut quality at a certain clarity level by dividing into 5 subplots, one for each clarity level. The side-by-side plot allows us to directly compare all the cut qualities of the diamonds at each clarity level since all the bars are displayed in the same plot. The faceted bar plot allows us to observe the number of diamonds of a particular cut quality across all the clarity levels. This type of plot shows in much greater detail the distribution of values across different sub-groups than side-by-side plots.

g)

```
ggplot(data = diamond) + geom_smooth(mapping = aes(x = carat, y = price, color = clarity), se = FALSE, r
labs(x = "Carat", y = "Price", title = "Price vs Carat For Clarity")
```

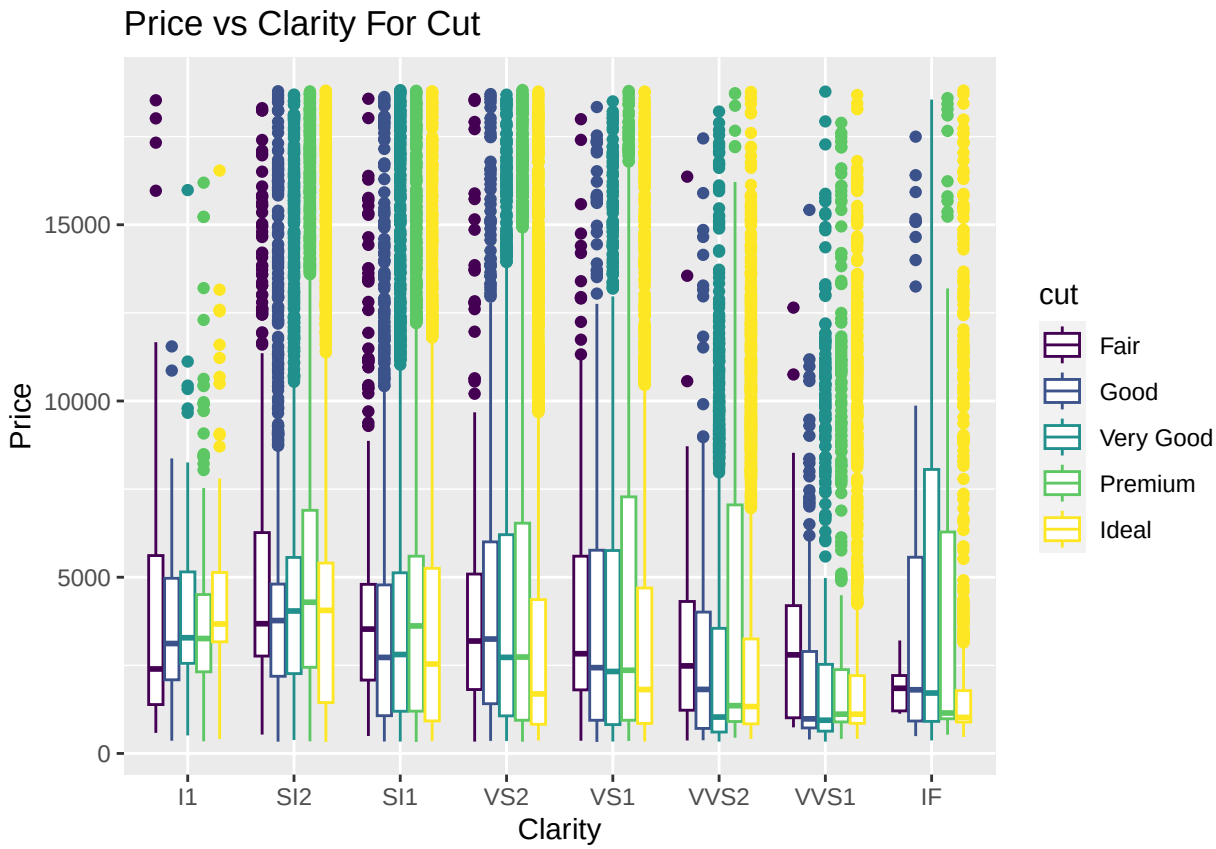
```
## `geom_smooth()` using formula = 'y ~ x'
```



The chart shows the trendlines for the relationship between price and carat for a particular level of clarity.

h)

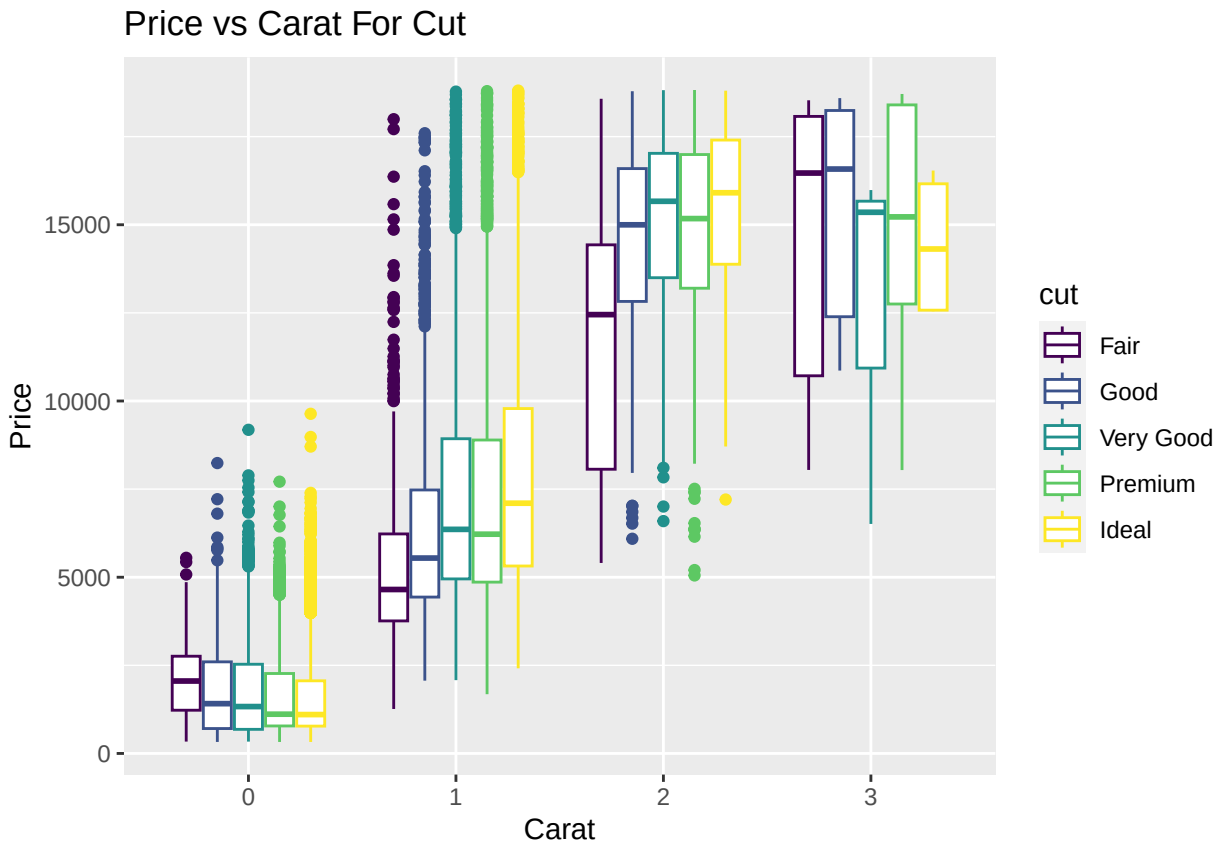
```
ggplot(data = diamond) + geom_boxplot(mapping = aes(x = clarity, y = price, color = cut)) +
labs(x = "Clarity", y = "Price", title = "Price vs Clarity For Cut")
```



The box plot shows the price distribution of diamonds of a particular clarity for each type of cut.

i)

```
for (i in 1:length(diamond$carat)) {
if ((diamond$carat[i] >= 0) && (diamond$carat[i] < 1)) {
diamond$carat[i] = 0
}
else if ((diamond$carat[i] >= 1) && (diamond$carat[i] < 2)) {
  diamond$carat[i] = 1
}
else if ((diamond$carat[i] >= 2) && (diamond$carat[i] < 3)) {
  diamond$carat[i] = 2
}
else {
diamond$carat[i] = 3
} }
diamond$carat <- factor(diamond$carat)
ggplot(data = diamond) + geom_boxplot(mapping = aes(x = carat,
                                                    y = price,
                                                    color = cut)) + labs(
                                                      x = "Carat", y = "Price", title = "Price vs Carat
```

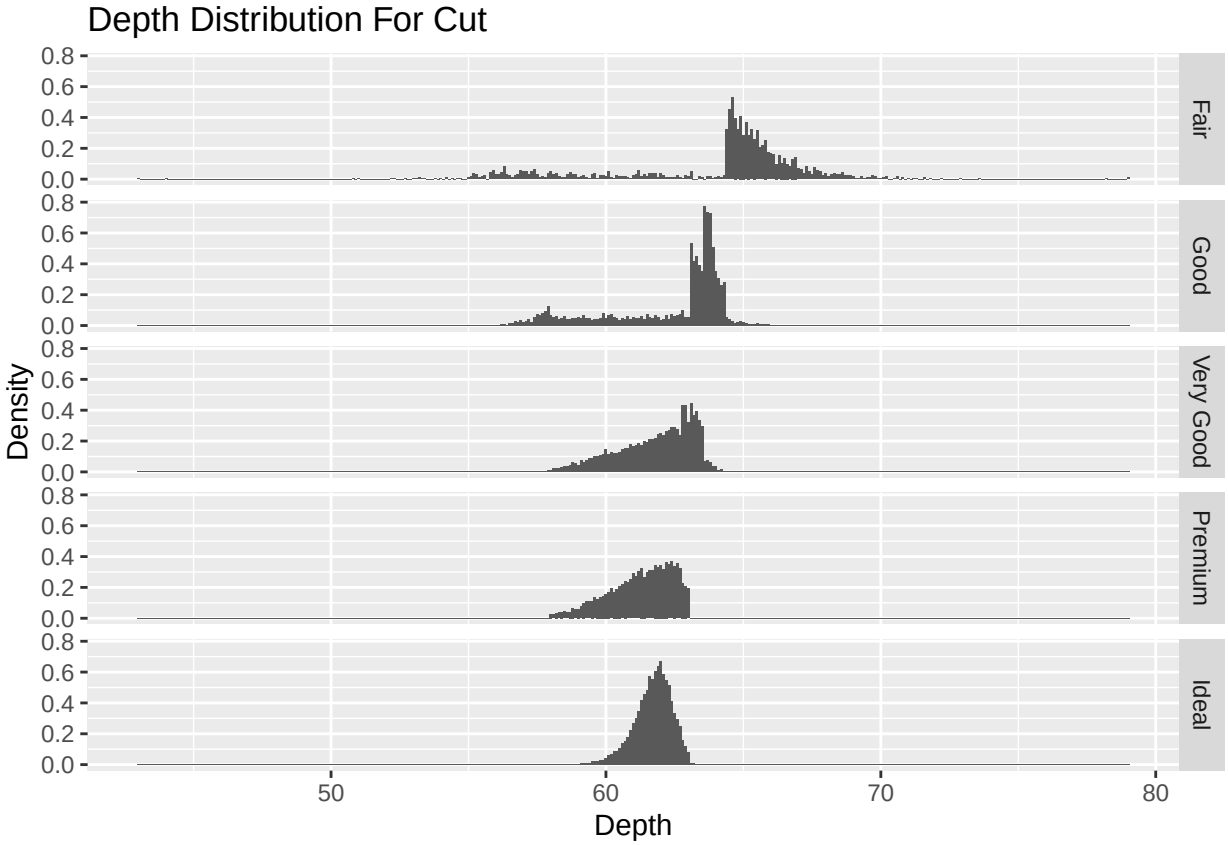


The box plot shows the price distribution of diamonds of a particular carat for each type of cut.

j)

```
ggplot(data = diamond) + geom_histogram(mapping = aes(x = depth, y = ..density..), binwidth = 0.1) + fa
labs(x = "Depth", y = "Density", title = "Depth Distribution For Cut")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The faceted histogram shows the kernel density distribution of diamond depth for each type of cut by dividing into 5 subplots, one for each type of cut.