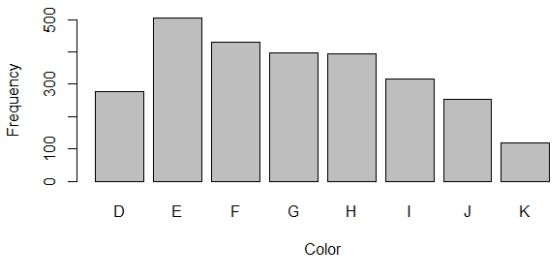
STA 371G: HW-4 Solutions

Hari Prakash Burra

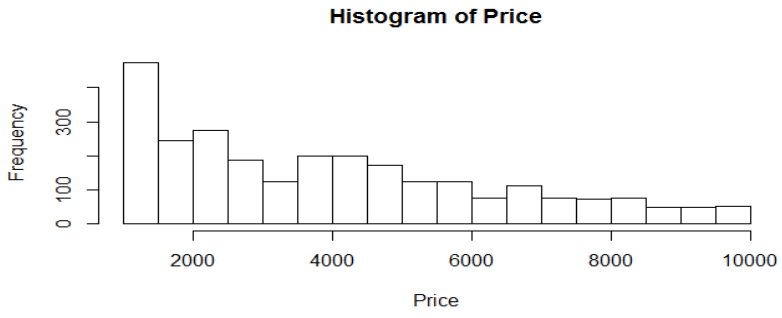
1)



- The plot for Color is more or less evenly distributed between the different color categories, with color 'E' having the highest no. of diamonds.
- For Clarity, there are lesser no. of diamonds in the flawless and nearly flawless compared to other categories.
- For Cut, overwhelming large number of diamonds have an Excellent or Very good cut compared to the miniscule number for Ideal and Good cuts.

```
diamonds <-
read.csv("https://raw.githubusercontent.com/brianlukoff/sta371g/master/data/diamonds.csv")
attach(diamonds)
#Since we are attaching diamonds here, there is no need to specify data=diamonds later in this
code
diamonds$Color <- ordered(diamonds$Color,c('D','E','F','G','H','I','J','K'))
diamonds$Clarity <- ordered(diamonds$Clarity,c('IF','VVS1','VVS2','VS1','VS2','SI1','SI2'))
diamonds$Cut <- ordered(diamonds$Cut,c('Ideal','Excellent','Very Good','Good'))
counts1 <- table(diamonds$Color)
barplot(counts, xlab='Color', ylab='Frequency')
counts2 <- table(diamonds$Clarity)
barplot(counts2, xlab='Clarity', ylab='Frequency')
counts3 <- table(diamonds$Cut)
barplot(counts3, xlab='Cut', ylab='Frequency')
```

2)

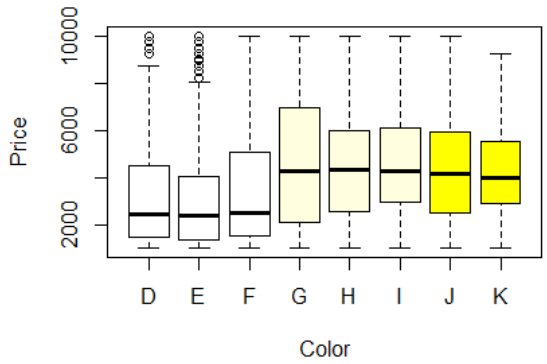


- Price is skewed strongly to the right. We expect less expensive diamonds and more diamonds for lower prices (say less than $2000)

- Carat.Size is also rightly skewed but there is a peak just after 1 Carat. The histograms don't match so other factors like Cut, Color and Clarity may be driving the price for some of the diamonds.

```
hist(Price)
hist(Carat.Size)
```

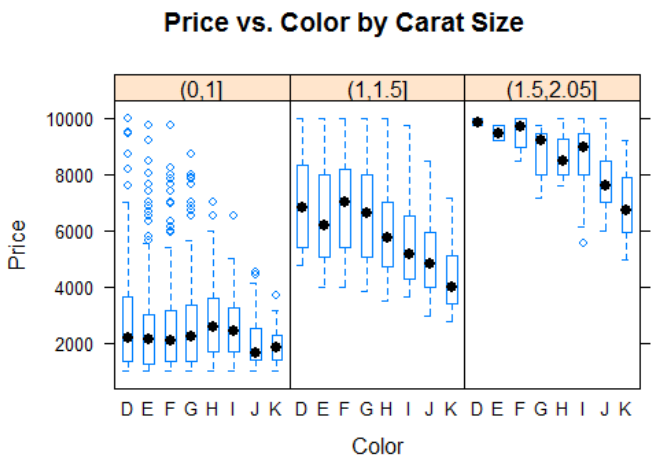
3)



- Colored Diamonds (G to K) seem to be priced higher than colorless ones (D,E,F), which was not expected in first glance as we were told that colourless ones are more expensive.
- This could be because of the other factors like Carat, Cut etc (i.e maybe colored diamonds are generally larger or cut more precisely etc.). There were also quite a few outliers for D and E colors, which could explain some of this as well.
- The 'col' option in the code for boxplot colored the boxplot so that it is easier to understand. For col, we specified a vector with first 3 elements in white, the next three in Light Yellow and the remaining in Yellow.

```
boxplot(Price~Color,col=c(rep("White",3),rep("LightYellow",3),rep("Yellow",3)),xlab="Color",yl
ab="Price")
```

4)

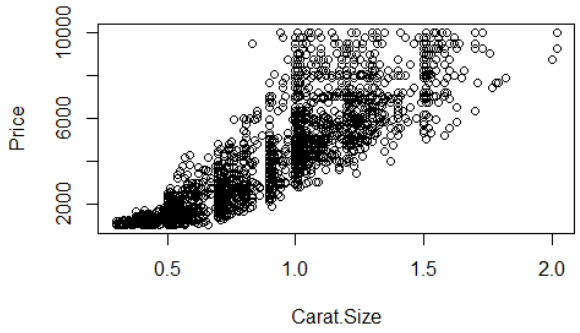


These plots look more sensible since across all the three categories, the prices for the colored diamonds are lower as we would expect. The higher concentration of diamonds in the First box for the lowest Carat Size and the outliers for the colorless categories can also be noticed.

A possible explanation for the plot in Question 3 is that the price was higher for Colored diamonds because most of the colorless diamonds were smaller in size, as it can be seen in the box for (0,1] Carat Size category. As we suspected, other factors like Carat.Size could be driving the price up for colored diamonds.

```
library(lattice)
carat=cut(Carat.Size,breaks=c(0,1,1.5,2.05))
bwplot(Price~Color | carat,xlab="Color",ylab="Price",main="Price vs. Color by Carat Size")
```

5)



The scatterplot shows an increasing trend as expected. We can also notice the higher concentration of smaller diamonds and diamonds just above 1 Carat as we saw in the histogram.

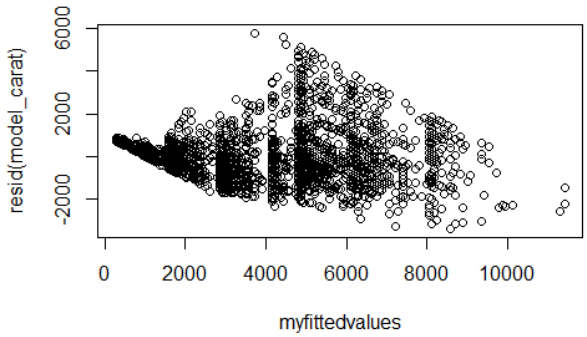
```
plot(Price~Carat.Size)
```

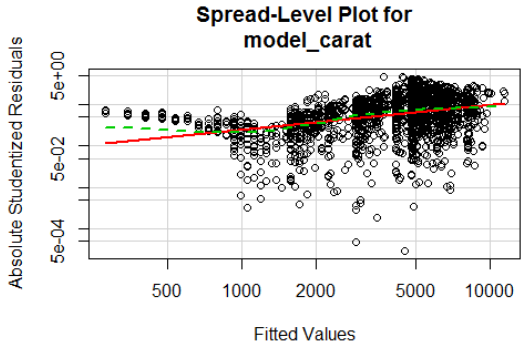
6) Roughly 74.2% of the variation in price is predicted by the Carat Size, as seen from the R-Squared value and the predictor is stastically significant due to the very low p-value.

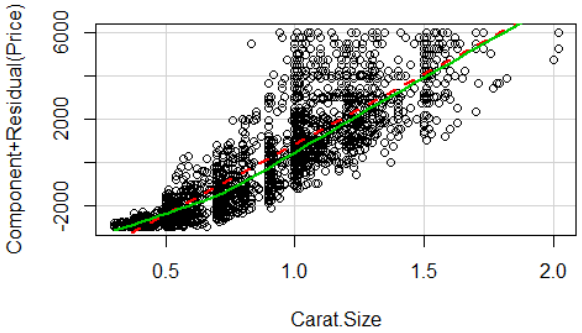
```
model_carat <- lm(Price~Carat.Size)
summary(model_carat)
```

7) The residual plot is concerning because the homoscedasticity assumption doesn't seem to be met at all. We can even observe a small linear trend in the residuals. Hence, the Confidence intervals may not be reliable (due to the violation of the homoscedasticity) and we could get biased predictions at certain carat sizes (due to the violation of the linearity).

```
myfittedvalues<-model_carat$fitted.values
plot(resid(model_carat),myfittedvalues)
#OR using car pacakage functions
library(car)
crPlots(model_carat)
spreadLevelPlot(model_carat)
#OR
plot(model_carat)
```







8) Split the data and run the regressions as shown below:

```
model_all_colors <- lm(Price ~ Carat.Size + Colorless)

diamonds$Colorless <- ifelse(diamonds$Color < "G", c("Yes"), c("No"))
PriceColorless <- diamonds$Price[diamonds$Colorless == "Yes"]
CaratColorless <- diamonds$Carat.Size[diamonds$Colorless == "Yes"]
PriceNotColorless <- diamonds$Price[diamonds$Colorless == "No"]
CaratNotColorless <- diamonds$Carat.Size[diamonds$Colorless == "No"]

model_colorless <- lm(PriceColorless ~ CaratColorless)
model_notcolorless <- lm(PriceNotColorless ~ CaratNotColorless)
```

9) model_all_colors:

Price = -2716 + 7178.2*Carat.Size (For Not Colorless diamonds)

Price = (-2716+980) + 7178.2*Carat.Size = -1736 + 7178.2*Carat.Size (For Colorless diamonds)

model_colorless:

Price = -2189.5 + 7822.5*CaratColorless

model_notcolorless:

Price = -2301.2 + 6765.9*CaratNotColorless

The equations are different but apart from that, the R-square and p-values etc are comparable.

```
summary(model_all_colors)
summary(model_colorless)
summary(model_notcolorless)
```

10)There are 7 dummy variables in this model for each of the colors other than Color 'D'. Color 'D' is taken to be the base level, hence it isn't assigned a Dummy variable.

```
model_carat_color <- lm(Price~Carat.Size+Color)
summary(model_carat_color)]
```

11) The 98% Confidence Interval for the coefficient of Carat.Size is [7459.397, 7788.704]. It is the estimated range of values of the mean change in the price for a unit increase in Carat.Size given that the color is unchanged.

```
confint(model_carat_color, level=0.98)
```

12) There are 15 coefficients in total, 1 for Carat.Size, 7 for the dummy variables of Color, and 7 more interaction terms (one each for the interaction of each color dummy variable with Carat.Size) There won't be interactions between the dummy variables because they are independent of each other.

```
model interact <- lm(Price~ Carat.Size*Color)
summary(model_interact)
```

13) For model_color_carat, H_0: $\beta_1 = \beta_2 = \ldots. = \beta_8 = 0$
For model_interact, H_0: $\beta_1 = \beta_2 = \ldots\ldots\ldots\ldots\ldots = \beta_{15} = 0$

For model_interact, we additionally need all the interaction terms' coefficients also to be zero apart from the main terms since existence of such a relation would mean that the price is still dependent on the variables involved even if their main terms' coefficients are zero.
For model_interact, the prediction lines for Price vs Carat Size can have different slopes for different colors, whereas in model_color_carat, the slope is the same for all colors (only the intercept varies)

14) Yes, the interaction between Carat.Size and Color is statistically significant. This can be concluded by running the anova command and checking the p-value for the Carat.Size:Color term as a whole.

```
anova(model_interact)
```

15) 5570.36 is the predicted value.
The Prediction interval [3601.55, 7539.18] would be more relevant for an individual customer looking to buy a single diamond and this covers for the fluctuating of one diamond better.
The Confidence interval [5444.42 , 5696.31] would be more relevant for the jewellery business as they are interested in the mean value of the 100 diamonds.

```
predict.lm(model_interact, list(Carat.Size=1,Color='F'))
predict.lm(model_interact, list(Carat.Size=1,Color='F'), interval = 'confidence')
predict.lm(model_interact, list(Carat.Size=1,Color='F'), interval = 'prediction')
```

16) Independence can be satisfied if we assume that the price of one diamond will not have much impact on the price of the next diamond. The residual plots don't look that good for the Linearity and Homoscedasticity assumptions. The plot for normality isn't perfect too.

```
myfittedvalues2<-model_interact$fitted.values
plot(myfittedvalues2,resid(model_interact))
library(car)
spreadLevelPlot(model_interact)
qqnorm(model_interact$residuals)
#OR
plot(model_interact)
```