Diamond Prices can be impacted by various things, such as inflation, carat, etc. I will be examining a data set with 53,943 diamonds which includes the variables, carat, cut, color, clarity, x, y, z, depth, table, and price. The images below give a visual explanation of each variable.

Diagram, calendar

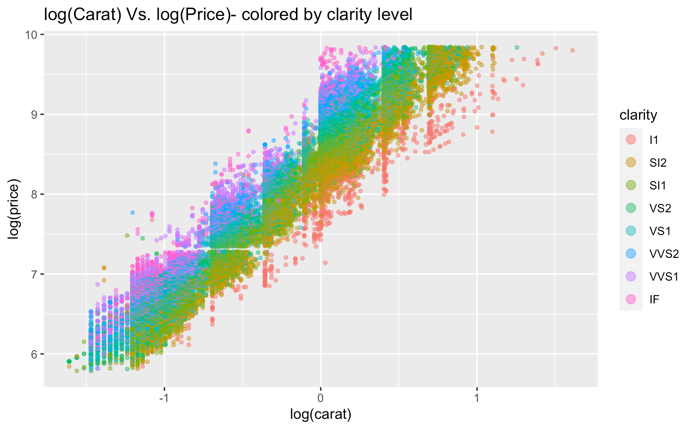
Description automatically generatedShape, rectangle

Description automatically generated

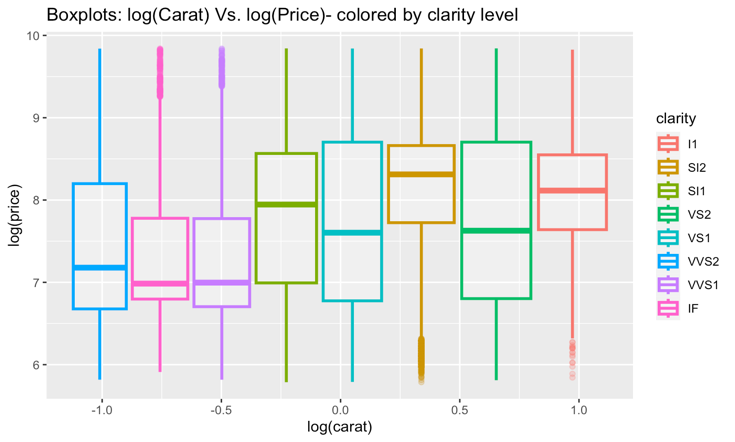
The purpose of this analysis is to answer the following questions:

1. Are all the variables in this data set important in explaining diamond prices? If, not which variables are important?
2. Is there an interaction between diamond clarity and carat weight?
3. Does the fitted model explain diamond prices well?

Chart, scatter chart

Description automatically generatedThe scatterplot of the price against doesn’t appear to follow a linear pattern. This would require a transformation to make the data follow a linear trend. I will transform my data by examining both price and carat on a log scale. After transforming the data, the scatterplot follows a more linear pattern.   


There is not an interaction between diamond clarity and carat weight. If there was an interaction, I would be able to assign different slopes for each clarity level.



The equation for our current model is.

Y: Actual log(diamond price)

: When all other values are at zero the log(price) of a fair cut, J color, I1 clarity diamond is after adjusting for log(carat).

: On average, holding all other variables constant there is a change in log(price) for every 1 unit increase in log(carat).

: On average, holding all other variables constant there is a change in log(price) for every 1 unit increase in depth.

: On average, holding all other variables constant there is a change in log(price) for every 1 unit increase in table.

: On average, holding all other variables constant there is a change in log(price) for every 1 unit increase in x variable.

: On average, holding all other variables constant there is a change in log(price) for every 1 unit increase in y variable.

: On average, holding all other variables constant there is a change in log(price) for every 1 unit increase in z variable.

: Measures the differential log(price) for diamonds in the I color group relative to countries in the J color group after adjusting for log(carat).

: Same as but for color group H.

: Same as but for color group G.

: Same as but for color group F.

: Same as but for color group E.

: Same as but for color group D.

: Measures the differential log(price) for diamonds in the SI2 clarity group relative to diamonds in the I1 clarity group after adjusting for log(carat).

: Same as but for clarity group SI1.

: Same as but for clarity group VS2.

: Same as but for clarity group VS1.

: Same as but for clarity group VVS2.

: Same as but for clarity group VVS1.

: Same as but for clarity group IF.

: Measures the differential log(price) for diamonds in the “good” cut group relative to diamonds in the fair cut group after adjusting for log(carat).

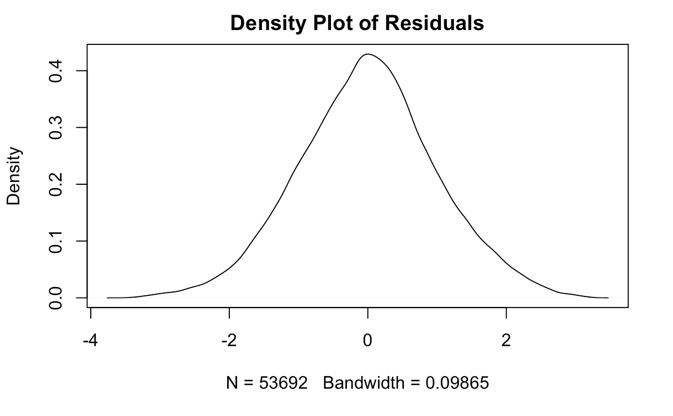
: Same as but for cut group “very good.”

: Same as but for cut group “premium.”

: Same as but for cut group “fair.”

: The error from the observation after transformation ε ~N(0,2)

I used the standard residual plot to locate the outliers in the data. I removed any outlier that was greater than 3 or less than -3. After remove the outliers I checked the regression assumptions using the residual plot, qq-plot, and density plot. Based on the residual plot the equal variances assumption appears to be met. The density plot is approximately normal, and the qq-plot is approximately with a few points that go briefly outside the error bands.

Chart, line chart

Description automatically generatedChart

Description automatically generated

The regression analysis found that the variables y, z, and table are not significant. I removed those form the model. The new fitted model for the data after the non-significant variables and outliers are removed is:

The R^2 after transforming back to the original data is 0.9644476. Almost 97% of variability in the log of diamond prices can be explained by the five variables in our model. This means our model is great at explaining variation in the log of diamond prices.

Now that the analysis is complete, the research questions have been answered below.

1. Are all the variables in this data set important in explaining diamond prices? If, no which variables are important?
   1. Based on the summary of the regression model we see that cut, color, clarity, depth, log(carat) and x have a significant impact on predicting diamond log(prices).
2. Is there an interaction between diamond clarity and carat weight?
   1. Based on the initial boxplots it can be determined that there is not a significant interaction between clarity and log(carat). This was surprising as I believe most people would think that clarity would interact in a significant way with carat weight.
3. Does the fitted model explain variability in diamond prices well?
   1. The fitted model has an r-squared value of about .97. This means that our model explains about 97% of variability in the log of diamond prices.

Code

diamonds=read.csv("/Users/rosie/Downloads/Diamonds Prices2022.csv",header=TRUE);diamonds

diamonds\_new <- diamonds # Duplicate data

diamonds\_new[is.na(diamonds\_new) | diamonds\_new == "NaN"] <- NA

attach(diamonds\_new)

diamonds\_new$cut=factor(diamonds\_new$cut, levels=c("Fair","Good", "Very Good", "Premium", "Ideal"))

diamonds\_new$color=factor(diamonds\_new$color, level=c("J","I","H","G","F","E","D"))

diamonds\_new$clarity=factor(diamonds\_new$clarity, level=c("I1", "SI2","SI1","VS2","VS1", "VVS2", "VVS1", "IF"))

mlr=lm(log(price)~log(carat)+(depth)+(table)+x+y+z+as.factor(color)+as.factor(clarity)+as.factor(cut),data=diamonds\_new)

# Lets look at the original data

library(ggplot2)

ggplot(data =diamonds\_new, aes(x = carat, y = price, color = clarity) ) +

geom\_point(alpha = .5, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ggtitle("Carat Vs. Price- colored by clarity level")

ggplot(data =diamonds\_new, aes(x = carat, y = price, color = clarity) ) +

geom\_boxplot(alpha = .25, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ggtitle("Boxplots: Carat Vs.Price- colored by clarity level")

#the origianl data appears to be curved, I will attempted to fix it by doing log(x) and log(y)

library(ggplot2)

ggplot(data = diamonds\_new, aes(x =log(carat), y = log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ggtitle("log(Carat) Vs. log(Price)- colored by clarity level")

ggplot(data = diamonds\_new, aes(x = log(carat), y = log(price), color = clarity) ) +

geom\_boxplot(alpha = .25, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ggtitle("Boxplots: log(Carat) Vs. log(Price)- colored by clarity level")

diamonds\_new <- diamonds # Duplicate data

diamonds\_new[is.na(diamonds\_new) | diamonds\_new == "NaN"] <- NA

attach(diamonds\_new)

diamonds\_new$cut=factor(diamonds\_new$cut, levels=c("Fair","Good", "Very Good", "Premium", "Ideal"))

diamonds\_new$color=factor(diamonds\_new$color, level=c("J","I","H","G","F","E","D"))

diamonds\_new$clarity=factor(diamonds\_new$clarity, level=c("I1", "SI2","SI1","VS2","VS1", "VVS2", "VVS1", "IF"))

mlr=lm(log(price)~log(carat)+(depth)+(table)+x+y+z+as.factor(color)+as.factor(clarity)+as.factor(cut),data=diamonds\_new)

#base cut is Fair, base color is J, base clarity is I1

summary(mlr)

summary(lm((price)~(carat)+(depth)+(table)+x+y+z+as.factor(color)+as.factor(clarity)+as.factor(cut),data=diamonds\_new)

)

#based on the model summary y,z, and table, are not significant so I will remove them.

#Plots of the different variables

library(ggplot2)

summary(lm(log(price)~log(carat)+(depth)+(table)+x+y+z+as.factor(color)+as.factor(clarity)+as.factor(cut),data=diamonds\_new))

#depth and price

ggplot(data =diamonds\_new, aes(x = (depth), y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ ggtitle("Depth vs. log(Price)- colored by clarity level")

#carat vs price

ggplot(data =diamonds\_new, aes(x = (carat), y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ ggtitle("Carat vs. log(Price)- colored by clarity level")

#cut vs price

ggplot(data =diamonds\_new, aes(x = (cut), y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ ggtitle("Cut vs. log(Price)- colored by clarity level")

#color vs price

ggplot(data =diamonds\_new, aes(x = (color), y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) +

guides(color = guide\_legend(override.aes = list(size = 3) ) )+ ggtitle("Color vs. log(Price)- colored by clarity level")

#table vs price

ggplot(data =diamonds\_new, aes(x =table, y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) + ggtitle("Y vs. log(Price)- colored by clarity level")

#x and price

ggplot(data =diamonds\_new, aes(x =x, y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) + ggtitle("X vs. log(Price)")

#y vs price

ggplot(data =diamonds\_new, aes(x =y, y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) + ggtitle("Y vs. log(Price)- colored by clarity level")

#z vs price

ggplot(data =diamonds\_new, aes(x =z, y =log(price), color = clarity) ) +

geom\_point(alpha = .5, size = 1) + ggtitle("Z vs. log(Price)- colored by clarity level")

#for the outliers do three plots, plot r, plot dfits, plot cooks d, lcarity is still important

# Regression Model

summary(mlr)

#based on the summary output we will remove table y and z as they are not significant

lm(log(price)~log(carat)+(depth)+x+as.factor(color)+as.factor(clarity)+ as.factor(cut),data=diamonds\_new)

model <-lm(log(price)~log(carat)+(depth)+x+as.factor(color)+as.factor(clarity)+ as.factor(cut),data=diamonds\_new)

summary(model)

r <- rstandard(model) # ask for standardized residuals

fit <- model$fitted.values # ask for fitted values

# residuals vs. fitted note: r vs. x is redundant in SLR

plot(fit,r, main = "Residuals vs. Fitted",

xlab = "fitted", ylab = "r")

abline(h=0, col = "blue")

# QQ plot

qqnorm(r)

qqline(r)

library(ggpubr)

ggqqplot(r)+

ggtitle("QQ Plot")

# density plot

plot(density(r), main = "Density Plot of Residuals")

# Index plot of residuals and hat values

model <-lm(log(price)~log(carat)+as.factor(cut)+as.factor(color)+as.factor(clarity)+(depth)+x,data=diamonds\_new)

summary(model)

Influence <- lm.influence(model) # ask for additional diagnostics

hat <- Influence$hat

r <- rstandard(model) # ask for standardized residuals

plot(r, main = "Index Plot of Standardized Residuals", pch = NA)

text(r, labels=rownames(diamonds\_new), cex = 0.5)

abline(h=c(-3,3), col="blue")

r

Dfits <- dffits(model) # ask for DFITS

CooksD <- cooks.distance(model)

# Index plot for DFits

plot(Dfits, pch = NA,

main = "Index Plot of DFits")

text(Dfits, labels=rownames(diamonds\_new), cex = 1.0)

# Index plot for Cooks D

plot(CooksD, pch = NA,

main = "Index Plot of Cook's Distance")

text(CooksD, labels=rownames(diamonds\_new), cex = 1.0)

# Checking for Influential Pts

# fit the MLR Model

lm(log(price)~log(carat)+as.factor(cut)+as.factor(color)+as.factor(clarity)+(depth)+x,data=diamonds\_new)

modsummary <- summary(lm(log(price)~log(carat)+as.factor(cut)+as.factor(color)+as.factor(clarity)+(depth)+x,data=diamonds\_new))

modelMLR <- lm(log(price)~log(carat)+as.factor(cut)+as.factor(color)+as.factor(clarity)+(depth)+x,data=diamonds\_new)

###NO OUTLIER DATA###

diamonds\_noout=diamonds\_new[-c(15, 682, 1642, 3276, 4051, 5326, 5822, 5858, 6140, 6160, 6418, 6736, 8204, 8593, 9601, 9712, 11183, 11407, 11408, 11561, 11964, 12978, 14078, 14239, 14435, 14576, 15952, 16085, 16101, 16405, 16541, 19554, 21887, 21936, 22143, 22525, 22701, 22732, 23410, 24521, 24655, 24884, 25068, 25069, 25315, 25332, 25626, 25925, 25926, 25998, 26004, 26078, 26106, 26199, 26238, 26244, 26312, 26408, 26484, 26492, 26550, 26635, 26661, 26966, 26999, 27050, 27197, 27227, 27350, 27430, 27456, 27458, 27508, 27531, 27636, 27674, 27923, 28711, 29549, 29946, 30948, 31996, 34612, 36652, 36653, 36984, 37449, 38154, 38349, 39502, 40287, 41020, 41021, 41052, 43425, 43426, 43910, 43990, 45866, 46477, 47004, 47950, 49774, 50672, 50673, 50792, 50954, 51174, 51175, 51370, 52378, 52861, 52862, 53599, 660, 2025, 2026, 2412, 5079, 5217, 5245, 5591, 5931, 6349, 6440, 7111, 7150, 7303, 7735, 7744, 7745, 8239, 8790, 9180, 10130, 10188, 10660, 11515, 11605, 11660, 12150, 12563, 12687, 12954, 13224, 13563, 14110, 14928, 15320, 15471, 15906, 15942, 16284, 16297, 16342, 16603, 17197, 17404, 17467, 17898, 17950, 18397, 19059, 19082, 19092, 19180, 19253, 19315, 19340, 19347, 19604, 19867, 19922, 20852, 21139, 21273, 21331, 21725, 21863, 22287, 22400, 22429, 22441, 22494, 22742, 22871, 23219, 23540, 23645, 23775, 23878, 23940, 23961, 24132, 24276, 24298, 24329, 24435, 24448, 24785, 24817, 24946, 25461, 25481, 25526, 25580, 25779, 25819, 25901, 25999, 26000, 26091, 26101, 26130, 26432, 26445, 26497, 26535, 26593, 26658, 26925, 26933, 27006, 27017, 27131, 27355, 27416, 27631, 27650, 27680, 27835, 30165, 30722, 30805, 31809, 32525, 34044, 35874, 38057, 39566, 40806, 41754, 41919, 46345, 48378, 50959, 51293, 52806,49558,49557,3927), ]

model <-lm(log(price)~log(carat)+as.factor(cut)+as.factor(color)+as.factor(clarity)+(depth)+x,data=diamonds\_new)

summary(model)

Influence <- lm.influence(model) # ask for additional diagnostics

hat <- Influence$hat

r <- rstandard(model) # ask for standardized residuals

plot(r, main = "Index Plot of Standardized Residuals", pch = NA)

text(r, labels=rownames(diamonds\_new), cex = 0.5)

abline(h=c(-3,3), col="blue")

r

paste(which(r>=3),collapse=", ")

paste(which(r<=-3),collapse=", ")

model\_noout=lm(log(price)~log(carat)+(depth)+x+as.factor(color)+as.factor(clarity)+as.factor(cut),data=diamonds\_noout)

summary(model\_noout)

Influence <- lm.influence(model\_noout) # ask for additional diagnostics

hat <- Influence$hat

r1 <- rstandard(model\_noout) # ask for standardized residuals

plot(r1, main = "Index Plot of Standardized Residuals", pch = NA)

text(r1, labels=rownames(diamonds\_noout), cex = 0.5)

abline(h=c(-3,3), col="blue")

r1

# residuals vs. fitted note: r vs. x is redundant in SLR

fit=model\_noout$fitted.values

plot(fit,r1, main = "Residuals vs. Fitted",

xlab = "fitted", ylab = "r")

abline(h=0, col = "blue")

# QQ plot

qqnorm(r1)

qqline(r1)

library(ggpubr)

ggqqplot(r1)+

ggtitle("QQ Plot")

# density plot

plot(density(r1), main = "Density Plot of Residuals")

# R squared for transformed model

SSE <- sum((diamonds\_noout$price-exp(model\_noout$fitted.values))^2)

SSY <- sum((diamonds\_noout$price-mean(diamonds\_noout$price))^2)

1-SSE/SSY

# residual standard error calculation

sqrt(SSE/(nrow(diamonds\_noout)-6-1)