Dumbledor's_army_final

```r
set.seed(291)

data <- read.csv("D:/Stat/R/Stat291/Final/Diamond/diamonds.csv", header = T)
data$X <- NULL #As X is another index column, we don't need it.

df <- sample_n(data, 1000) #as our dataset is TOO long.

str(df)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ carat  : num  0.72 0.3 0.6 0.34 0.51 0.7 0.51 0.71 2.18 1.02 ...
##  $ cut    : chr  "Premium" "Very Good" "Ideal" "Ideal" ...
##  $ color  : chr  "H" "I" "D" "E" ...
##  $ clarity: chr  "SI1" "VVS1" "SI2" "VS2" ...
##  $ depth  : num  62.2 61.2 61.3 62.4 61.7 58.8 62.7 59.8 63.1 60.4 ...
##  $ table  : num  57 60 59 55 56 64 56 58 58 60 ...
##  $ price  : int  2311 552 1338 745 956 2468 1070 3112 16878 4238 ...
##  $ x      : num  5.75 4.27 5.42 4.48 5.17 5.74 5.06 5.78 8.29 6.53 ...
##  $ y      : num  5.72 4.29 5.48 4.53 5.14 5.79 5.08 5.82 8.23 6.49 ...
##  $ z      : num  3.57 2.62 3.34 2.81 3.18 3.39 3.18 3.47 5.21 3.93 ...
```

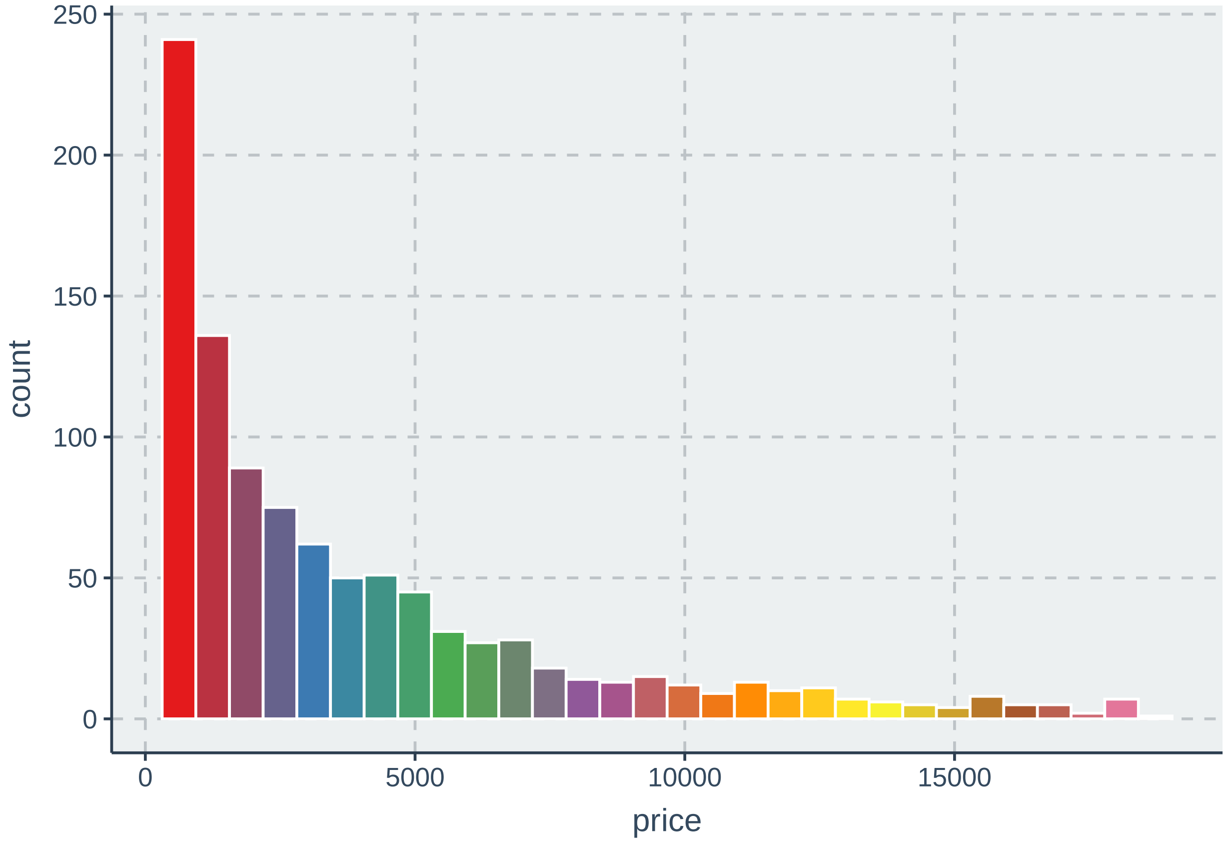
```r
#There are categorical variables in our dataset, we need to change them to
#factor with the relevant levels
```

```r
df$cut <- factor(df$cut, levels = c("Fair","Good","Very Good",
                                    "Premium","Ideal"))
df$color <- factor(df$color, levels = c("J","I","H","G","F","E","D"))
df$clarity <- factor(df$clarity, levels = c("I1", "SI2", "SI1", "VS2",
                                            "VS1", "VVS2", "VVS1", "IF"))

#-- Description
#This dataset is about different diamonds and their respected properties
```

```r
####Part Merve.
##Descriptive Statistics.
#Min-Max of every class.

#There are columns that have wrong types. lets change them.

########################################################################

#Now that they are of correct class, we can calculate the descriptive
#statistics. Lets get to know our dataset.

ggthemr("flat") #ggplot theme that will be applied to all plots

#Firstly, we do descriptive statistic to know relationship between price
#and length(x)in our data set
```

```r
df %>% ggplot() +
  geom_histogram(aes(x=price),
                 fill = colorRampPalette(brewer.pal(8, "Set1"))(30),
                 color = "white")
```

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




```r
min(df$price)#the minimum price needed to buy a dimond
```

```
## [1] 374
```

```r
which.min(df$price) # to find index of minimum value
```

```
## [1] 679
```

```r
df$x[679]
```

```
## [1] 4.36
```

```r
# so ,length of minumum price diamond is 4.36

max(df$price) #price of the most expensive diamond
```

```
## [1] 18470
```

```r
which.max(df$price) # to find index of maximum price value.
```

```
## [1] 413
```

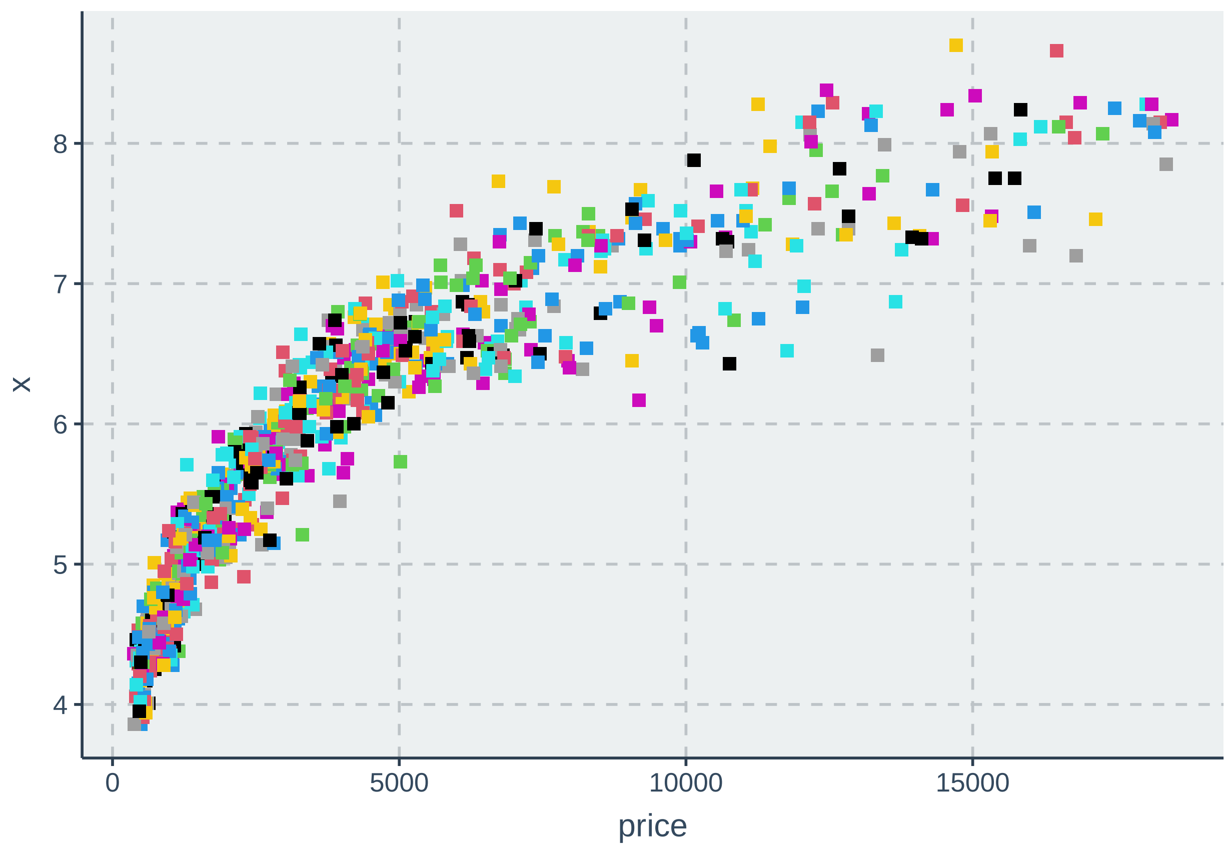
```r
df$x[413]
```

```
## [1] 8.17
```

```r
# so , length of maximimum price diamond is 8.17

# Let's to better see the relationship between price and length.

cor(df$price, df$x) # according to correlation table, 0.890 means that
```

```
## [1] 0.8900019
```

```r
#price and length has a very strong positive relationship.
#Also we can see this relationship using ggplot,

df %>% ggplot(aes(x=price, y=x)) +
  geom_point(size=2, shape= 15, color = df$price) +
  scale_fill_discrete(df$price)
```




```r
# we can observe that ; when x (length) is increasing, price is also increasing.
```

```r
###Rümeysa
#A car buyer is interested in understanding how 3 different brands of car
#makers ("bmw, nissan, volvo") leads to
#highest horsepower. We have multiple samples of different brandings and the
#related cars horsepowers.

#To understand whether there is a statistically significant difference in the
```

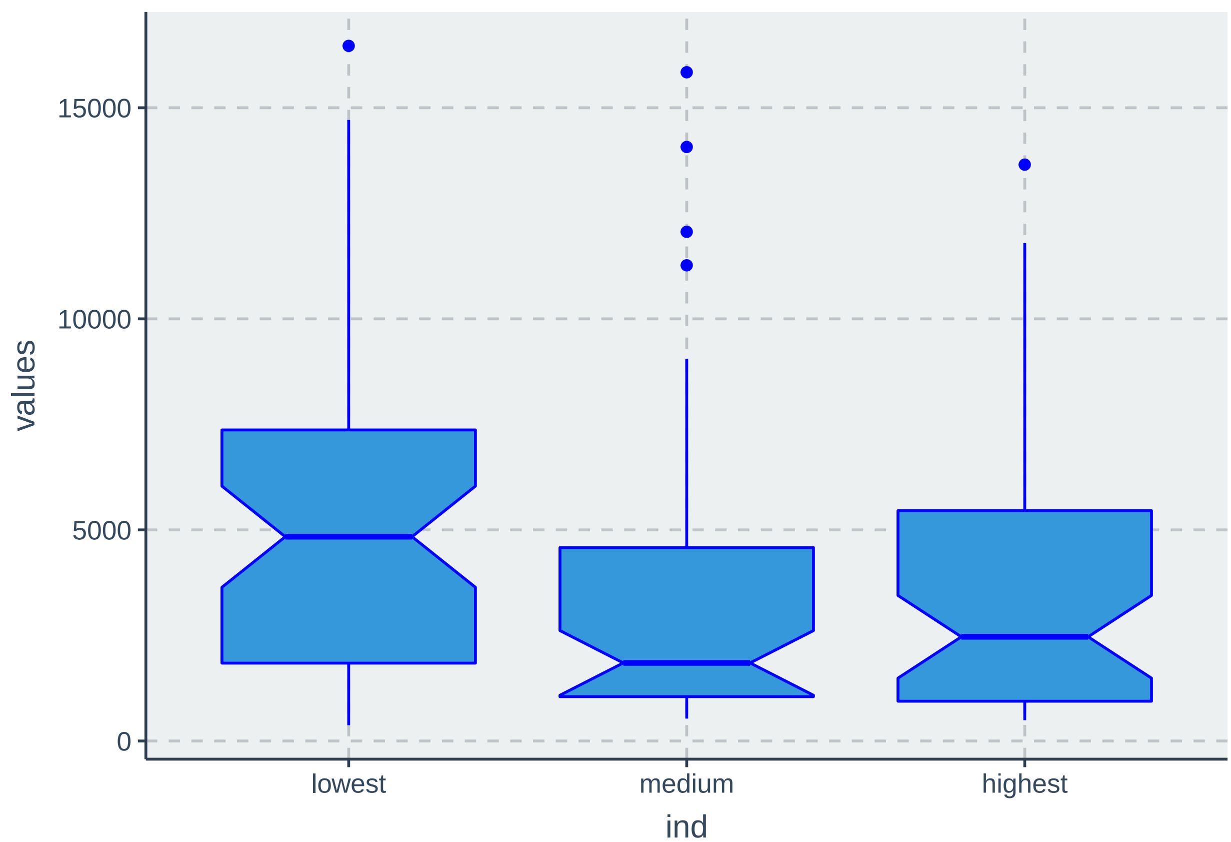
```r
#mean horsepower that
#results from these three brands, researchers can conduct a one-way ANOVA,
#using "make" as the factor and "horsepower" as the response.

#Our null hypothesis => μ1 = μ2

lowest <- df$price[df$color == "J"]
medium <- df$price[df$color == "F"]
highest <- df$price[df$color == "D"]

combined_groups <- data.frame(lowest = lowest[1:53], medium = medium[1:53],
                              highest = highest[1:53])
#We took only 42 of "highest" as two datasets are not equal in lenght
combined_groups
```

```
##    lowest medium highest
## 1     956   1070    1338
## 2    2536   4238    2468
## 3     775   3702     964
## 4     450   5586    5543
## 5    7368    660   11795
## 6    4098   3018   13653
## 7   11463   1323    3406
## 8    5723   4580     796
## 9    4968   5019    1440
## 10   3783   2491    5456
## 11  16466   1849     651
## 12   5455   5655     644
## 13   8203   3918     815
## 14   4927   1614    3965
## 15  12267   3828    3250
## 16   6201   9055    8311
## 17   8712    955    5005
## 18   5242   5292     972
## 19   2426   2870     593
## 20   7695   1594     780
## 21   5715   3519    3153
## 22  12308   1303     939
## 23   4410   3588     574
## 24   2123   1094   10694
## 25   4839   1655     608
## 26   1260    694    6539
## 27    417    633    1066
## 28   6750   4339    3277
## 29   6306   8505    5656
## 30  12554   1050    1002
## 31   1363   1613    3229
## 32  13228  14071    1030
## 33   2147    842    1140
## 34    611    711    7002
## 35    461  11268    3556
## 36    533    540    5342
## 37   6339    608   11138
## 38   5042  12061    4215
```

```
## 39    3881    1571     8192
## 40     374    6597     1569
## 41    6754    1080     5821
## 42    3091    6809    11765
## 43    1844     605     1050
## 44     770     652      709
## 45    3285    6108     1779
## 46    8298   15841     2306
## 47    4315     737      738
## 48    3255    3485      709
## 49    1133     737      942
## 50    7428    1086     7696
## 51   14711    1128      492
## 52     978     530     5088
## 53   10137    3117     3307
```

```r
stacked_groups <- stack(combined_groups)
stacked_groups %>% ggplot(aes(x=ind, y=values, color="blue")) +
  geom_boxplot(notch = T) +
  scale_fill_brewer(palette="Dark2")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```




```r
anova_results <- aov(values ~ ind, data = stacked_groups)
summary(anova_results)
```

```
##               Df    Sum Sq  Mean Sq F value Pr(>F)
## ind           2 9.368e+07 46840035   3.362 0.0372 *
## Residuals   156 2.173e+09 13932387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#The p value which we found is not less than 0.05. Therefore, we can
#say that we wont reject the null hypothesis.
#Now lets see if dropping the outliers changes the result.

stacked_groups_outliers <- boxplot(stacked_groups, plot=FALSE)$out
stacked_groups_oout <- stacked_groups[-which(stacked_groups$values %in% stacked_groups_outliers),]

anova_results_oout <- aov(values ~ ind, data = stacked_groups_oout)
summary(anova_results_oout)
```

```
##               Df    Sum Sq  Mean Sq F value Pr(>F)
## ind           2 6.701e+07 33503716   3.349 0.0378 *
## Residuals   150 1.501e+09 10005086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#It changed the result. From our research it seems that there is no
#particular way of handling outliers in this situation.
#Thus we will leave it at this and use the data with the outliers.
#Thus, we can say that there is no significant difference between
#colors of diamonds and the price.
```

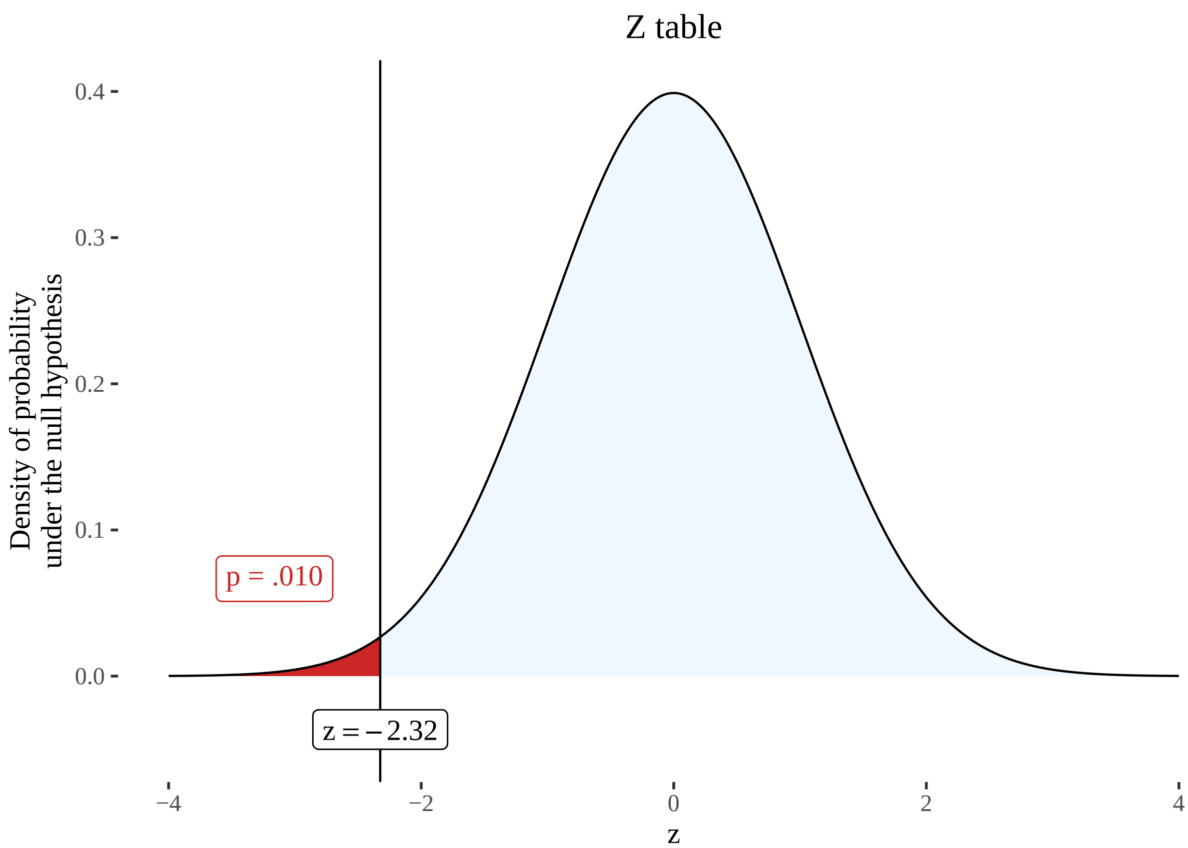
```
###Barine
#We want to see if differences between prices of diamonds that have different cut quality.
#We want to see if two quality points that are next to each other have any differences between their va

#So our Null hypothesis will be μ1 (ideal) > μ2 (premium), thus alternative hypthesis is
#μ1 <= μ2

ideal <- df$price[df$cut == "Ideal"]
premium <- df$price[df$cut == "Premium"]


zval <- (mean(ideal) - mean(premium) - 0)/sqrt(((sd(ideal)^2)/length(ideal))+
                                                ((sd(premium)^2)/length(premium)))
zval
```

```
## [1] -2.324372
```

```
1 - pnorm(zval,0,1)
```

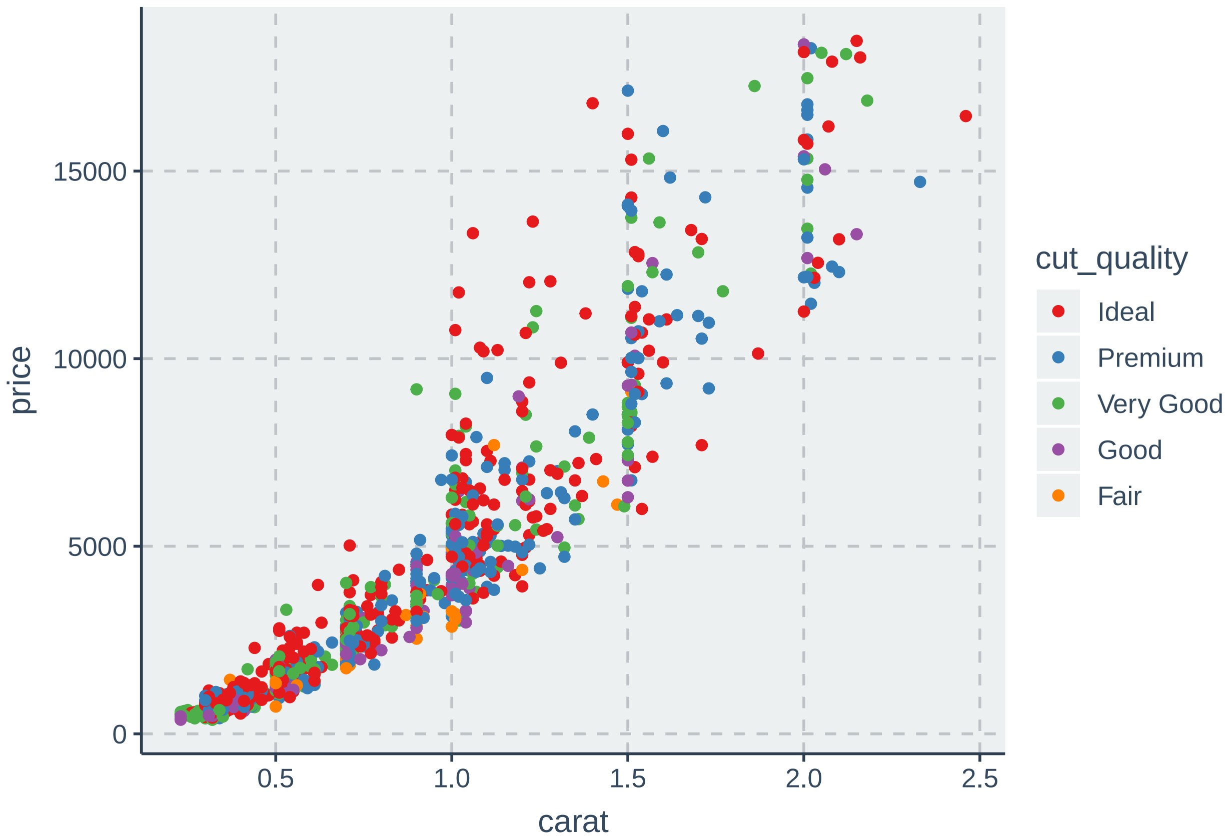
```
## [1] 0.9899472
```

```
plotztest(
  z = zval,
  tails = "one",
  title = "Z table",
  xmax = 4,
)
```

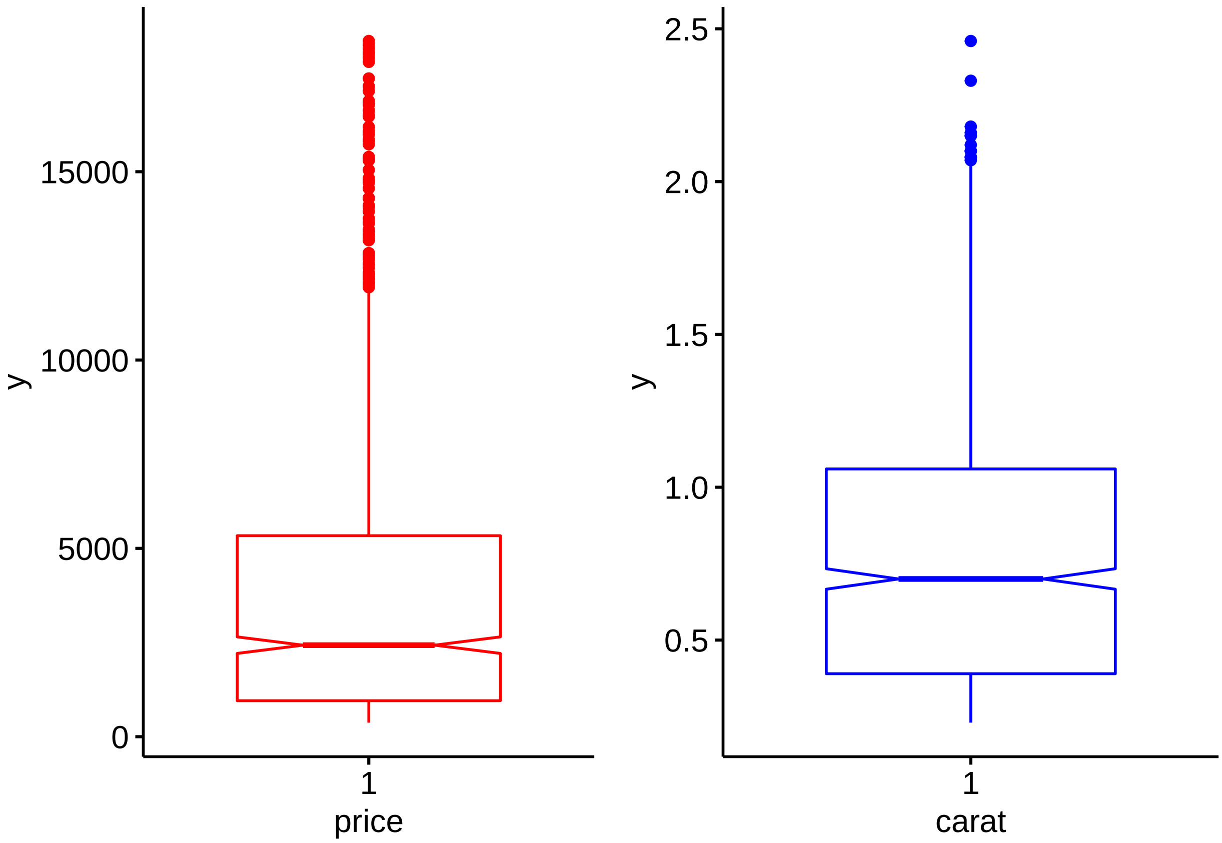



```
#The rejection area does not include p-value, so the null hypothesis could not be rejected.
#The average engine size of cars which have sedan and hatchback bodies are very close to each other.
```

```
###Mert
#We have a diamond, We know its carat property. We want to find its approximate value.

#So first of all, lets see the plot of carat ~ price

cut_quality <- fct_rev(df$cut)
df %>% ggplot(aes(x=carat, y=price, group=cut_quality)) + geom_point(aes(color=cut_quality)) + scale_col
```




```
#From the plot we can see that the relationship does appear to be positive linear.
#As carat increases, the price tends to increase as well in a linear fashion.

#Now we want to check the datas for extreme outliers

plot1 <- ggboxplot(df$price, xlab = "price", color = "red", notch = T)
plot2 <- ggboxplot(df$carat, xlab = "carat", color = "blue", notch = T)
grid.arrange(plot1,plot2,ncol = 2)
```




```
boxplot.stats(data$horsepower)$out
```

```
## NULL
```

```
boxplot.stats(data$engine.size)$out
```

```
## NULL
```

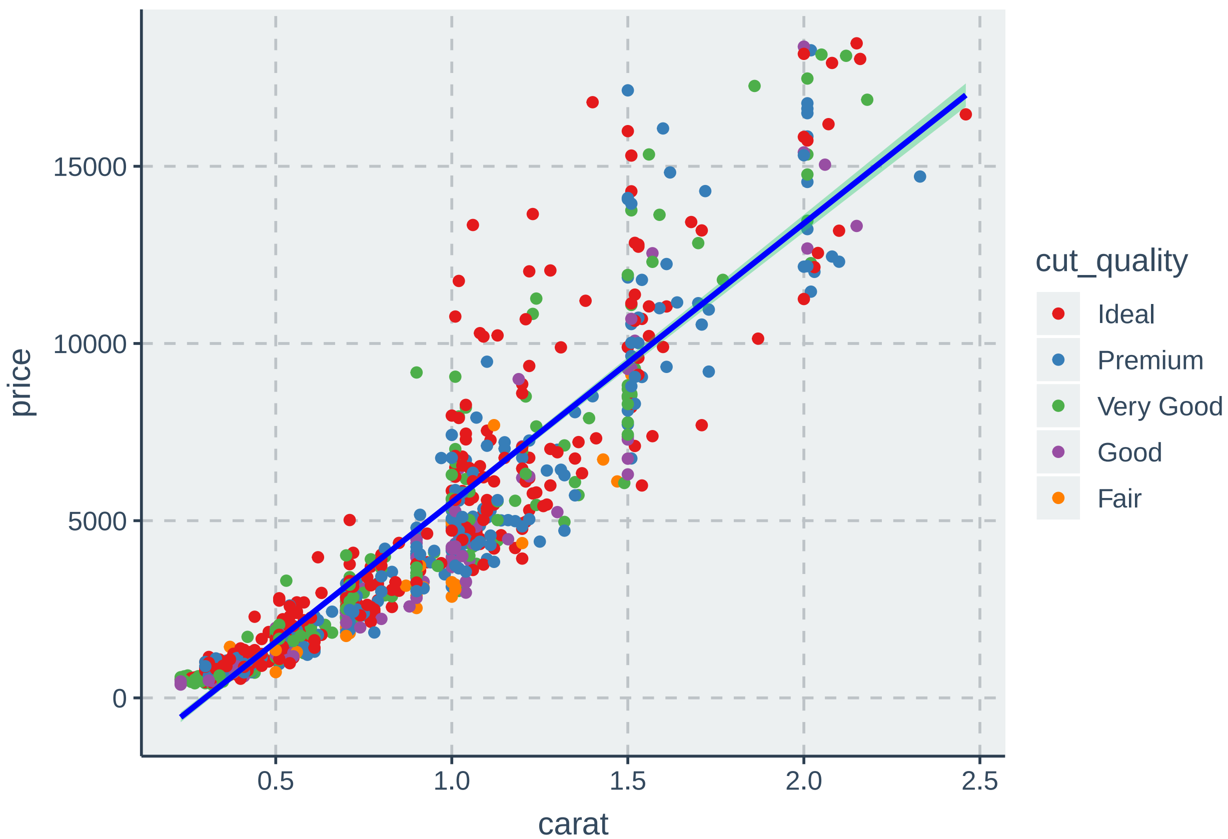
```
#It seems we do have outliers but we wont be dropping them as they follow the same trends.
#Because they follow the same trends they will only add more precision to our linear model instead of b

#Once the relationship between our variables is confirmed to be linear
#and outliers are dealt with, we can proceed to fit a simple
#linear regression model.

r_model <- lm(data = df, formula = price~carat)

#Now that we have our model we can see the specifics about it with summary()

summary(r_model)
```

```
##
## Call:
## lm(formula = price ~ carat, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3772.9  -875.9     1.2   576.9  8143.3
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2357.65      89.47  -26.35   <2e-16 ***
## carat        7873.09      96.66   81.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1437 on 998 degrees of freedom
## Multiple R-squared:  0.8692, Adjusted R-squared:  0.8691
## F-statistic:  6634 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
#This summary tells us that each additional carat is associated with
#an average increase in price of 8029.29 points. And the intercept
#value of -1.2540 tells us the estimated price of..? well its out of
#our analysis anyways. We will take it as 0 :)

# plot the points (actual observations), regression line, and confidence interval!
df %>% ggplot(aes(x=carat, y=price)) +
  geom_point(aes(color=cut_quality, group=cut_quality)) +
  scale_color_brewer(palette = "Set1") + geom_smooth(method = "lm", color = "blue")
```

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




```r
#As our p value for engine.size is <2e-16 which is lower than .05 we can say
#with confidence that horsepower and engine.size have a significant relatence
```

```
#This number tells us the percentage of the variation in the horsepowers can be
#explained by the engine sizes. In this case it seems that %65.5 of the variation of horsepowers
#can be explained with engine sizes
```