Linear_model

2024-05-27

Using diamonds as example

```
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
str(diamonds)
```

```
## tibble [53,940 x 10] (S3: tbl_df/tbl/data.frame)
##  $ carat  : num [1:53940] 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 [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
anyNA(diamonds)
```

```
## [1] FALSE
```

```
diamonds_no_na <- diamonds[complete.cases(diamonds), ]
anyDuplicated(diamonds)
```

```
## [1] 1006
```

```
diamonds_unique <- diamonds[!duplicated(diamonds), ]
```

Select data for lm

We want know the linear relationship in carat and price and is it influenced by cut.
So cut = ideal and fair selected

```r
data_ideal <- subset(diamonds, cut == "Ideal")[ , c("carat", "price")]
data_fair <- subset(diamonds, cut == "Fair")[ , c("carat", "price")]
head(data_fair)
```

```
## # A tibble: 6 x 2
##   carat price
##   <dbl> <int>
## 1  0.22   337
## 2  0.86  2757
## 3  0.96  2759
## 4  0.7   2762
## 5  0.7   2762
## 6  0.91  2763
```

Firstly caculating the correlation of carat and price in both cut quality

```r
correlation_ideal <- cor.test(data_ideal$carat,data_ideal$price, use = "complete.obs")
correlation_fair <- cor.test(data_fair$carat,data_fair$price, use = "complete.obs")
print(correlation_ideal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data_ideal$carat and data_ideal$price
## t = 374.94, df = 21549, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9293791 0.9329287
## sample estimates:
##      cor
## 0.931176
```

```r
print(correlation_fair)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data_fair$carat and data_fair$price
## t = 67.369, df = 1608, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8459582 0.8715640
## sample estimates:
##       cor
## 0.8592985
```

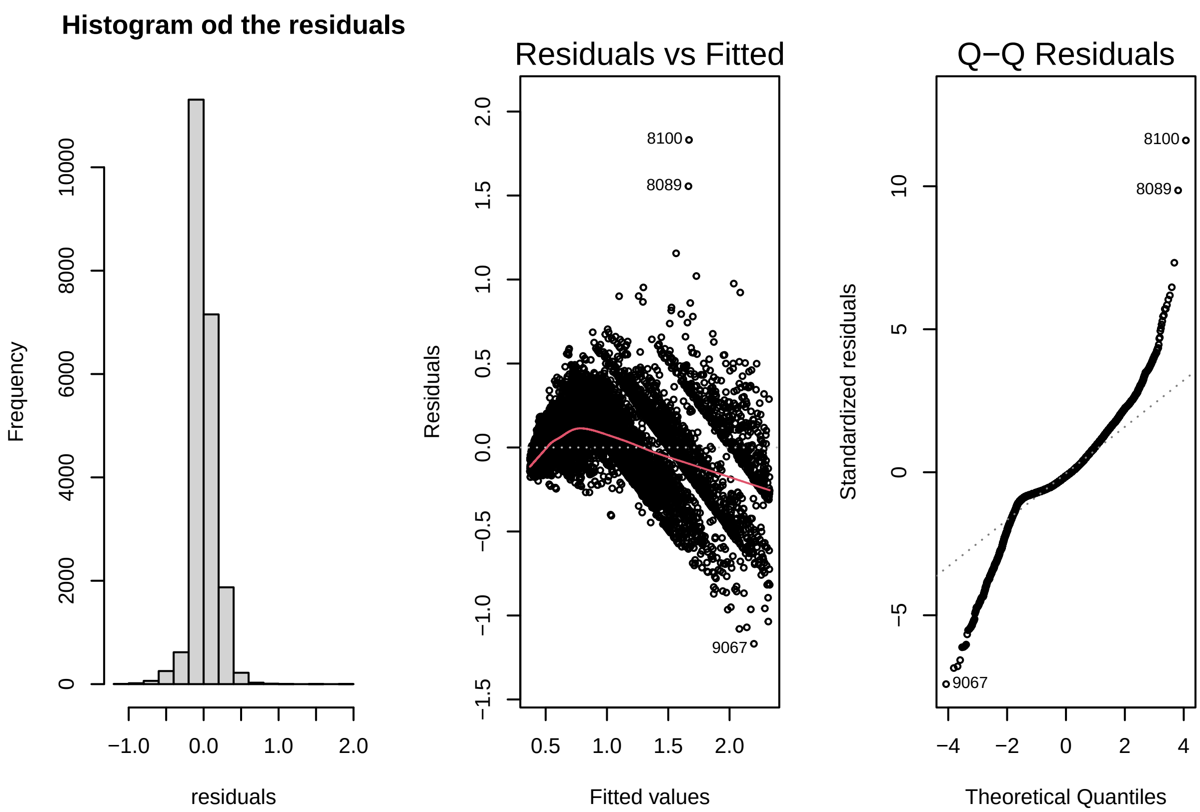
Both value greater than 0.85 and it can said that in two different cuts, there is an linear relationship exists.

Model building and Evaluation

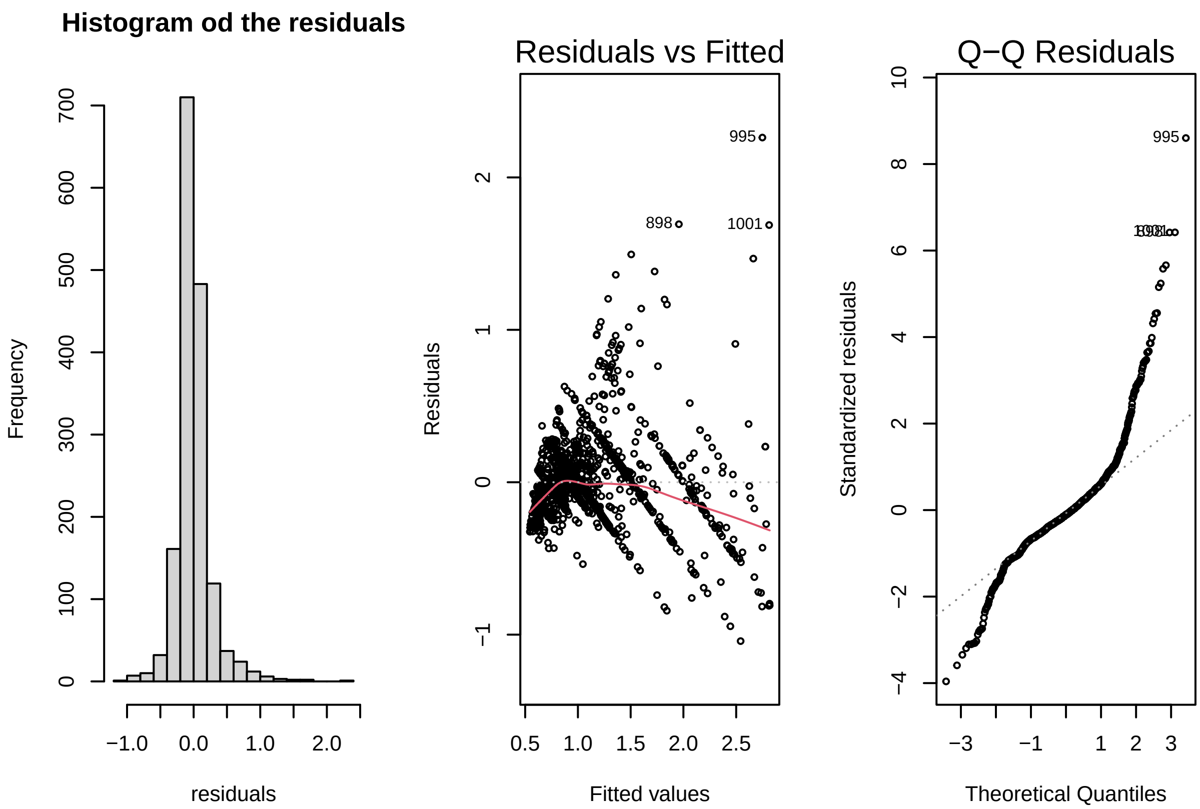
```
model_ideal <- lm(data_ideal$carat ~ data_ideal$price)
model_fair <- lm(data_fair$carat ~ data_fair$price)
```

Evaluate

```
#For ideal
par(mfrow = c(1,3))
hist(residuals((model_ideal),breaks = 5, col = "grey"),
     main = "Histogram od the residuals", xlab = "residuals", cex = 0.6)
plot(model_ideal, which= c(1,2), cex = 0.6)
```




```
#For fair
par(mfrow = c(1,3))
hist(residuals((model_fair),breaks = 5, col = "grey"),
     main = "Histogram od the residuals", xlab = "residuals", cex = 0.6)
plot(model_fair, which= c(1,2), cex = 0.6)
```



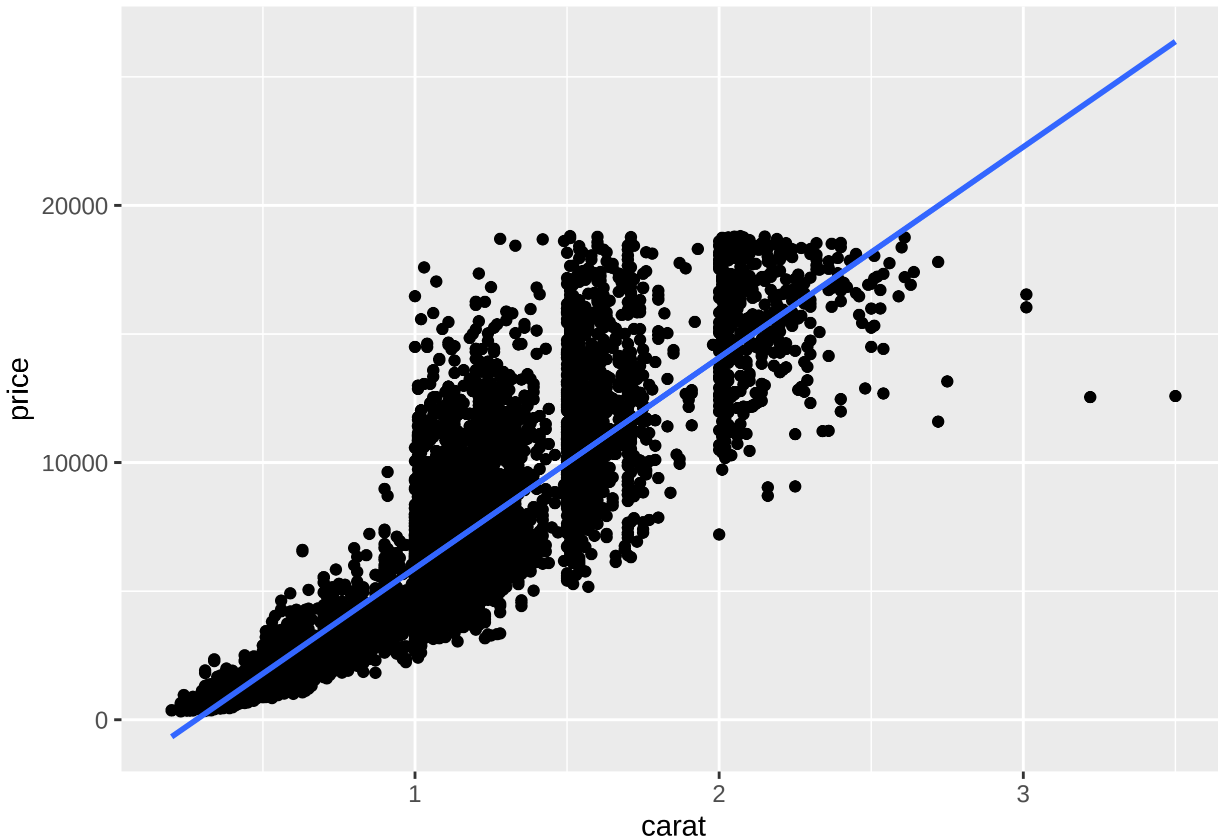
Assume it is good enough

Visulizaton

```
library(ggplot2)
p <- ggplot(data_ideal, aes(x = carat, y = price))
p <- p + geom_point()
p <- p + geom_smooth(method = "lm", se = FALSE)
print(p)
```

Only one group

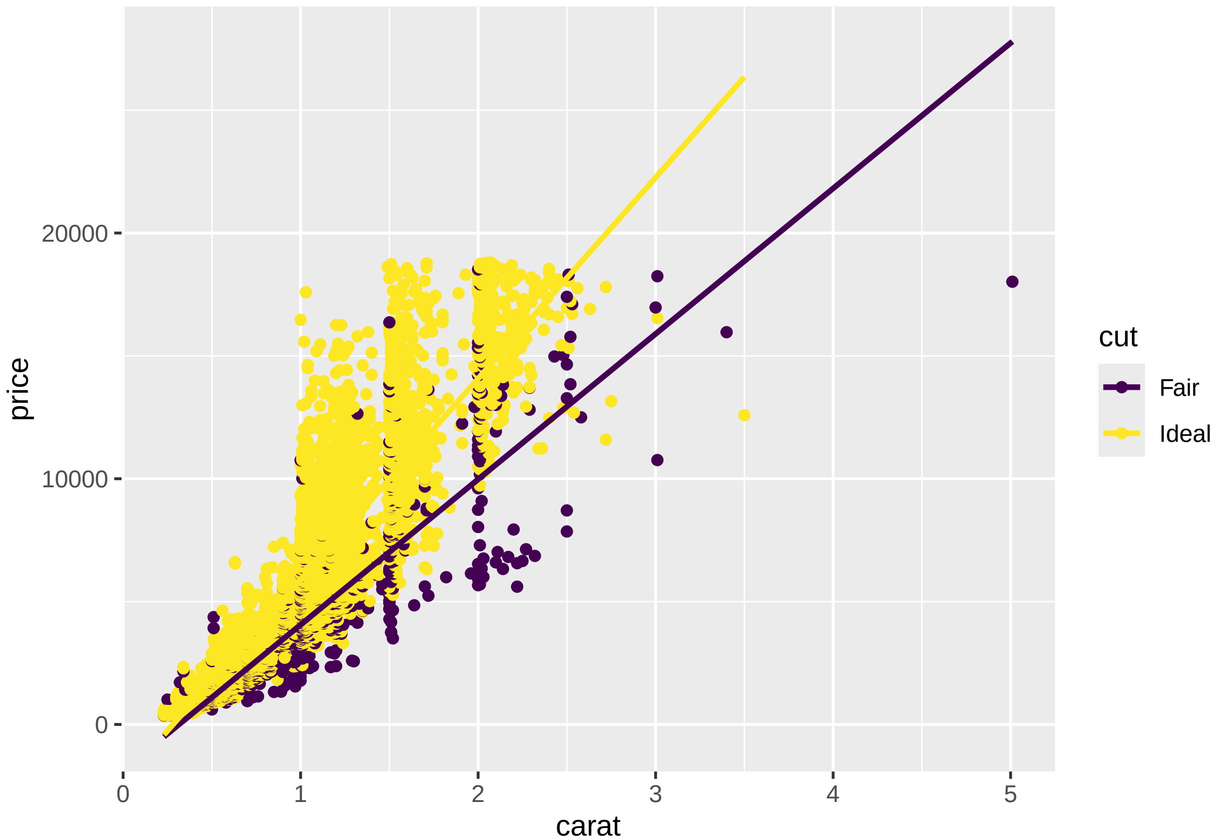
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
data_selected <- subset(diamonds, cut == c("Ideal","Fair"))[ , c("carat", "price","cut")]
p <- ggplot(data_selected, aes(x = carat, y = price, color = cut))
p <- p + geom_point()
p <- p + geom_smooth(method = "lm", se = FALSE)
print(p)
```

Two group need in one data frame

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



Calculate Z score

Next, z score is calculated to determine whether the slop of two fitted lines are significantly differently.
H0: There is no difference between the two correlation coefficients.
H1: There is significant difference between the two correlation coefficients.

```
# Extract regression coefficients and standard errors
beta1 <- summary(model_ideal)$coefficients["data_ideal$price", "Estimate"]
se_beta1 <- summary(model_ideal)$coefficients["data_ideal$price", "Std. Error"]
beta2 <- summary(model_fair)$coefficients["data_fair$price", "Estimate"]
se_beta2 <- summary(model_fair)$coefficients["data_fair$price", "Std. Error"]
# Calculate z-score
z <- (beta1 - beta2) / sqrt(se_beta1^2 + se_beta2^2)
print(z) # badly
```

```
## [1] -10.04229
```

Here, **z is greater than 1.96 (significance level 0.05), indicating a significant difference between the two correlation coefficients, so we reject H0.** This suggests that there is a gender difference in the impact of drug addiction, with males being more affected and having a higher growth rate.