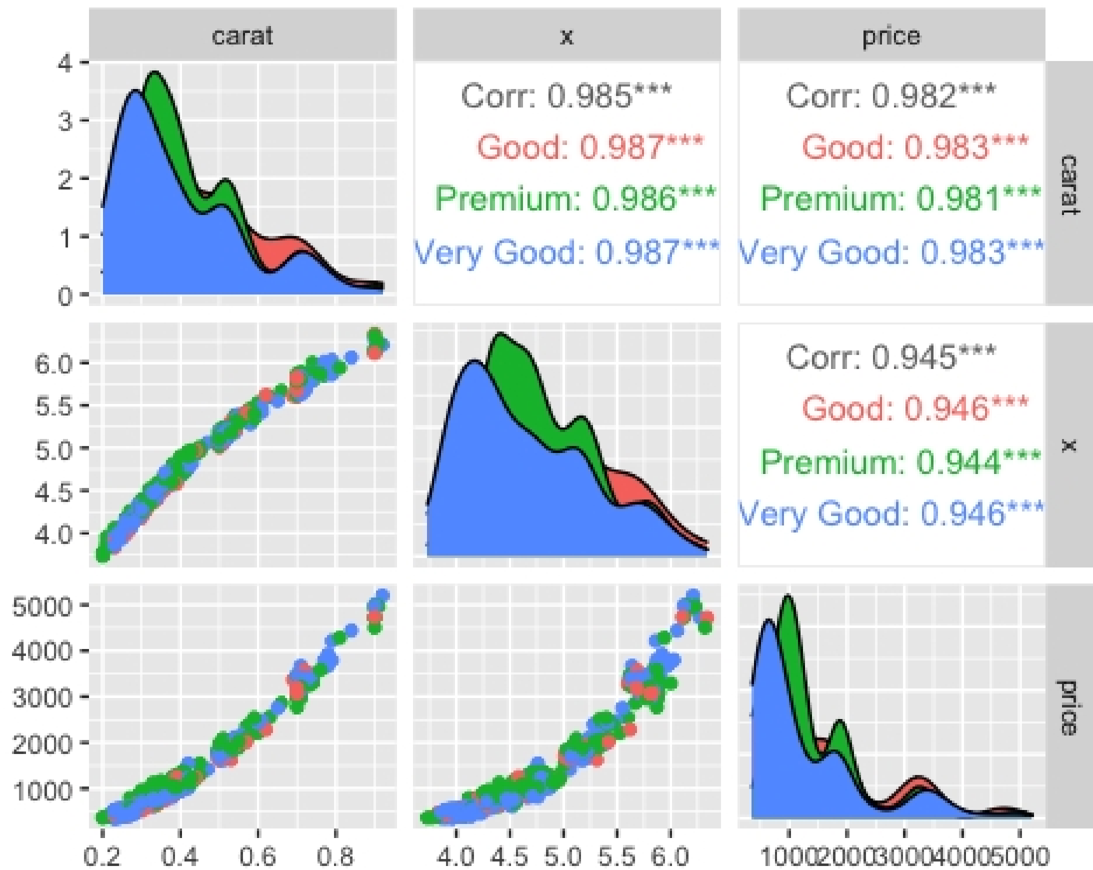
Modeling with both continuous and categorical predictors, and linear transformations of variables

1. Load the `tidyverse` library and the `GGally` library.

```
library(tidyverse)

library(GGally)

diamonds <- read_csv('diamonds.csv')

## Rows: 660 Columns: 4

## ── Column specification
────────────────────────────────────────────────────────

## Delimiter: ","
## chr (1): cut
## dbl (3): price, carat, x

##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

set.seed(31878039)
my_diamonds <- diamonds %>%
  sample_n(640) %>%
  mutate(cut <- factor(cut,
 levels = c("Good", "Very Good", "Premium")))

head(my_diamonds)

## # A tibble: 6 × 5
##   price carat     x cut       `cut <- factor(cut, levels = c("Good", "Very
Good…
##   <dbl> <dbl> <dbl> <chr>     <fct>
## 1  3210  0.7   5.71 Premium   Premium
## 2  1108  0.37  4.57 Very Good Very Good
## 3  1038  0.37  4.66 Premium   Premium
## 4  3183  0.7   5.75 Premium   Premium
## 5   367  0.2   3.73 Premium   Premium
## 6  1117  0.38  4.63 Premium   Premium
```

2. Use `ggpairs` from the `GGally` library to create a matrix of plots and correlations between the continuous variables. Optional: include `aes(colour = cut)` inside the `ggpairs` function to get the points coloured by cut.

```
my_diamonds %>%
  ggpairs(columns = c('carat', 'x', 'price'), aes(colour = cut))
```



3. Use `lm` to create a model called m1 for price using the continuous predictors carat and x and the categorical predictor cut, including both continuous-categorical interactions.

```
m1 <- lm(price ~ carat + x + cut + x:cut + carat:cut, data=my_diamonds)
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: price
##             Df    Sum Sq   Mean Sq    F value    Pr(>F)
## carat        1 565833134 565833134 50013.4182 < 2.2e-16 ***
## x            1  10150479  10150479   897.1906 < 2.2e-16 ***
## cut          2   2665583   1332792   117.8041 < 2.2e-16 ***
## x:cut        2    751106    375553    33.1947 1.962e-14 ***
## carat:cut    2    136264     68132     6.0221  0.002566 **
## Residuals  631   7138898     11314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m2 <- lm(price ~ carat + x + cut + carat:cut + x:cut, data=my_diamonds)
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: price
##              Df    Sum Sq   Mean Sq    F value    Pr(>F)
## carat         1 565833134 565833134 50013.4182 < 2.2e-16 ***
## x             1  10150479  10150479   897.1906 < 2.2e-16 ***
## cut           2   2665583   1332792   117.8041 < 2.2e-16 ***
## carat:cut     2    837339    418669    37.0058 6.358e-16 ***
## x:cut         2     50031     25015     2.2111    0.1104
## Residuals 631   7138898     11314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All of m1's coefficients are statistically significant whereas one of the coefficients of m2 is not, specifically the interaction term x:cut.

4. Create a model called m3 for response price using the continuous predictors carat and x and the categorical predictor cut, including the interaction between carat & cut but without the interaction between x & cut. Your choice on the order of the terms in the formula!

```
m3 <- lm(price ~ carat + x + cut + carat:cut, data=my_diamonds)
anova(m3)

## Analysis of Variance Table
##
## Response: price
##              Df    Sum Sq   Mean Sq   F value    Pr(>F)
## carat         1 565833134 565833134 49822.772 < 2.2e-16 ***
## x             1  10150479  10150479   893.771 < 2.2e-16 ***
## cut           2   2665583   1332792   117.355 < 2.2e-16 ***
## carat:cut     2    837339    418669    36.865 7.172e-16 ***
## Residuals 633   7188929     11357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m3,m1)

## Analysis of Variance Table
##
## Model 1: price ~ carat + x + cut + carat:cut
## Model 2: price ~ carat + x + cut + x:cut + carat:cut
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    633 7188929
## 2    631 7138898  2     50031 2.2111 0.1104

anova(m3,m2)

## Analysis of Variance Table
##
```

```
## Model 1: price ~ carat + x + cut + carat:cut
## Model 2: price ~ carat + x + cut + carat:cut + x:cut
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    633 7188929
## 2    631 7138898  2     50031 2.2111 0.1104
```

The two tests show no difference because they are both an F-test and order does not matter.

anova(m3) is useful for interpreting the interaction between carat:cut, not x:cut. In contrast, the nested model, let's say anova(m3,m2) interprets if the interaction x:cut is significant. It does this by comparing m3 to m2 in terms of regressors. Depending on the p-value, this also assesses if we still would've got the sample data whether there is an interaction or not.

5. Is the interaction between x and cut statistically significant? Give the evidence that you are using and your conclusion (about 40 words in total).

The p-value is large therefore there is no significant evidence that the coefficients for the x:cut interaction are not zero. This indicates that x:cut interaction can be removed. The interaction x:cut is not useful for explaining the price.

6. Create model m4 that removes the interaction between carat and cut from model m3 (ie, uses carat, x and cut as predictors but with no interactions at all). Perform a nested model F-test of the full model (m1 or m2) against m4 using `anova`. Show the code and the `anova` output.

```
m4 <- lm(price ~ carat + x + cut, data=my_diamonds)
anova(m4, m1)

## Analysis of Variance Table
##
## Model 1: price ~ carat + x + cut
## Model 2: price ~ carat + x + cut + x:cut + carat:cut
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    635 8026268
## 2    631 7138898  4    887370 19.608 3.173e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

m4 is compared to m1 to explain the full significance of the coefficients that's been dropped. Comparing m4 to m3 would explain the significance of carat:cut, whereas m4 against m1 would explain the significance of carat:cut and x:cut.

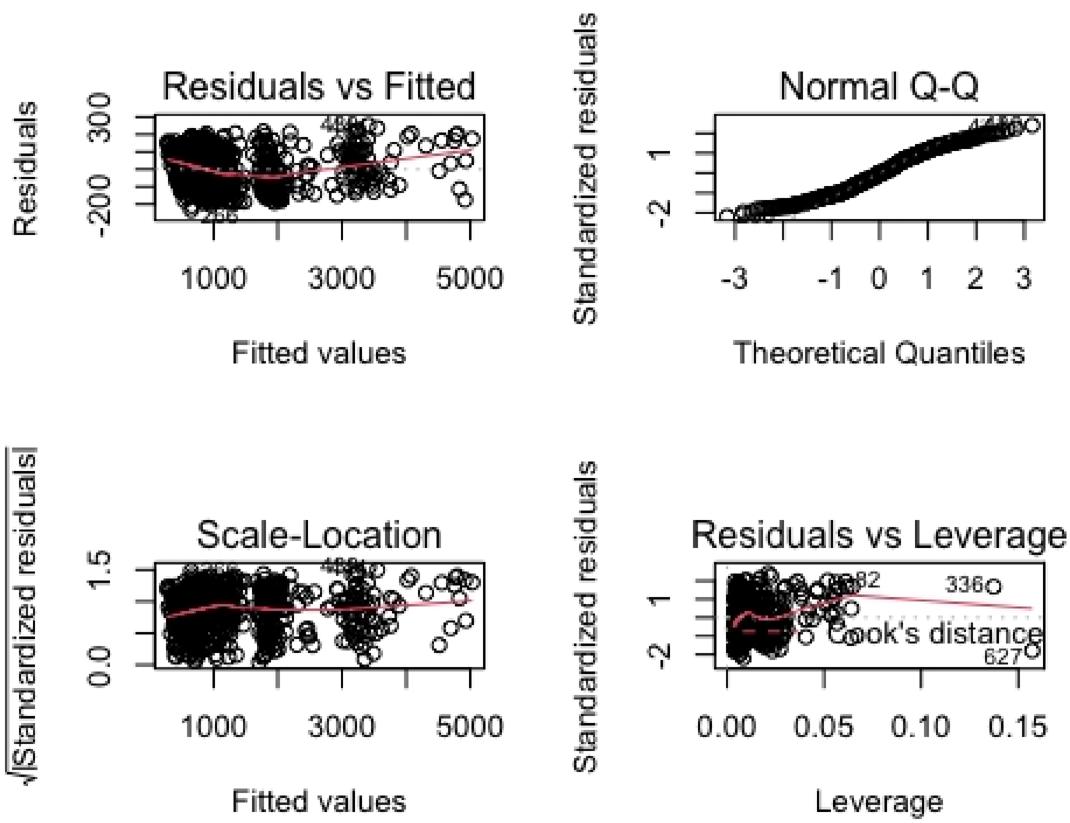
p-value is small so enough evidence that x:cut and carat:cut are not zero. The interaction terms should not be dropped

No. We also need to consider the assumptions of the model. For instance, if the interaction term is dropped and the standardised residuals are not constant then this implies that maybe the interaction term is needed to explain the response. Always check assumptions after dropping any interaction terms from the model.

7. Describe model m3 geometrically in about 30 words. (Eg, an equation *y = mx + c* represents a single straight line; what does model m3 represent?)

Model m3 represents a plane due to the two continuous regressors. In addition, m3 has separate intercepts with three different slopes for x.

```
par(mfrow = c(2,2))
plot(m3)
```



The residuals are not linear. There is a dip and then it went up.

The standardised residuals look reasonably okay as there is no pattern meeting the equal variance assumption.

The q-q plot shows deviations at the extremeties implying that this may not be a normal distribution.

There is no influential points.

8. Show the model summary output for model m3.

```
summary(m3)

##
## Call:
## lm(formula = price ~ carat + x + cut + carat:cut, data = my_diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -232.592  -88.688   -3.537   88.153  252.571
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3662.77     153.48  23.865  < 2e-16 ***
## carat                12104.48     187.61  64.519  < 2e-16 ***
## x                    -1578.14      46.63 -33.842  < 2e-16 ***
## cutPremium             468.37      43.76  10.704  < 2e-16 ***
## cutVery Good           168.79      42.37   3.984 7.58e-05 ***
## carat:cutPremium      -595.91      91.86  -6.487 1.76e-10 ***
## carat:cutVery Good    -151.51      90.25  -1.679   0.0937 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 106.6 on 633 degrees of freedom
## Multiple R-squared:  0.9877, Adjusted R-squared:  0.9876
## F-statistic:  8504 on 6 and 633 DF,  p-value: < 2.2e-16

new_diamonds = my_diamonds %>%
  mutate(price_NZ000 = ((price/1000)*1.45))

m5 <- lm(price_NZ000 ~ carat + x + cut + carat:cut, data = new_diamonds)
summary(m5)

##
## Call:
## lm(formula = price_NZ000 ~ carat + x + cut + carat:cut, data =
new_diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33726 -0.12860 -0.00513  0.12782  0.36623
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.31102    0.22255  23.865  < 2e-16 ***
## carat                17.55150    0.27204  64.519  < 2e-16 ***
## x                    -2.28831    0.06762 -33.842  < 2e-16 ***
## cutPremium            0.67914    0.06345  10.704  < 2e-16 ***
## cutVery Good          0.24474    0.06144   3.984 7.58e-05 ***
## carat:cutPremium     -0.86407    0.13319  -6.487 1.76e-10 ***
```

```
## carat:cutVery Good -0.21969    0.13086  -1.679   0.0937 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1545 on 633 degrees of freedom
## Multiple R-squared:  0.9877, Adjusted R-squared:  0.9876
## F-statistic:  8504 on 6 and 633 DF,  p-value: < 2.2e-16
```

There are changes in the slope and intercepts of m3 to m5. The intercepts shifted to the left due to a decrease and the slopes also shifted downwards due to a decrease. Their standard errors are different too which means that m5's data points are close to the regression line.

Although all of the coefficients have changed, the new predictor variable does not changed the model fit at all. Despite, a lower residual standard error in m5 than m3, both models' adjusted r-squared and multiple r-squared are all the same. This indicates that m5 is just like m3.

9. Create a tibble of new data to use for prediction. The new data should have carat values 0.25, 0.4, 0.6, 0.9, x values all 4.7 and all Premium cut.

```
new_tibble <- tibble(carat=c(0.25, 0.4, 0.6, 0.9),
                      x = 4.7,
                      cut = "Premium")

predict(m5, newdata = new_tibble, interval = 'confidence')

##          fit        lwr        upr
## 1 -0.5930404 -0.6664562 -0.5196247
## 2  1.9100737  1.8920523  1.9280952
## 3  5.2475593  5.1480052  5.3471134
## 4 10.2537877 10.0102634 10.4973119

predict(m5, newdata = new_tibble, interval = 'prediction')

##          fit        lwr        upr
## 1 -0.5930404 -0.9052386 -0.2808423
## 2  1.9100737  1.6060958  2.2140517
## 3  5.2475593  4.9282024  5.5669162
## 4 10.2537877  9.8647094 10.6428659
```

With 95% confidence, the price of a premium cut diamond with 0.25 carat and 4.7 width will fall between $-0.6664562 and $-0.5196247 However this does not makes sense due to the negative numbers. In contrast, premium cut diamond with 0.4 carat and 4.7 width makes more sense as the price of that will fall between $1.8920523 and $1.9280952.

The prediction interval predicts where an individual observation will be in the model without being affected by the observations in the model. Due to lack of precision, prediction intervals are wider than confidence interval. For example, premium cut diamond with 0.4 carat and 4.7 width would cost between $1.6060958 and $2.2140517.