Seminar 2

Ingrid Canelles

Question 1

```
library(tidyverse)
```

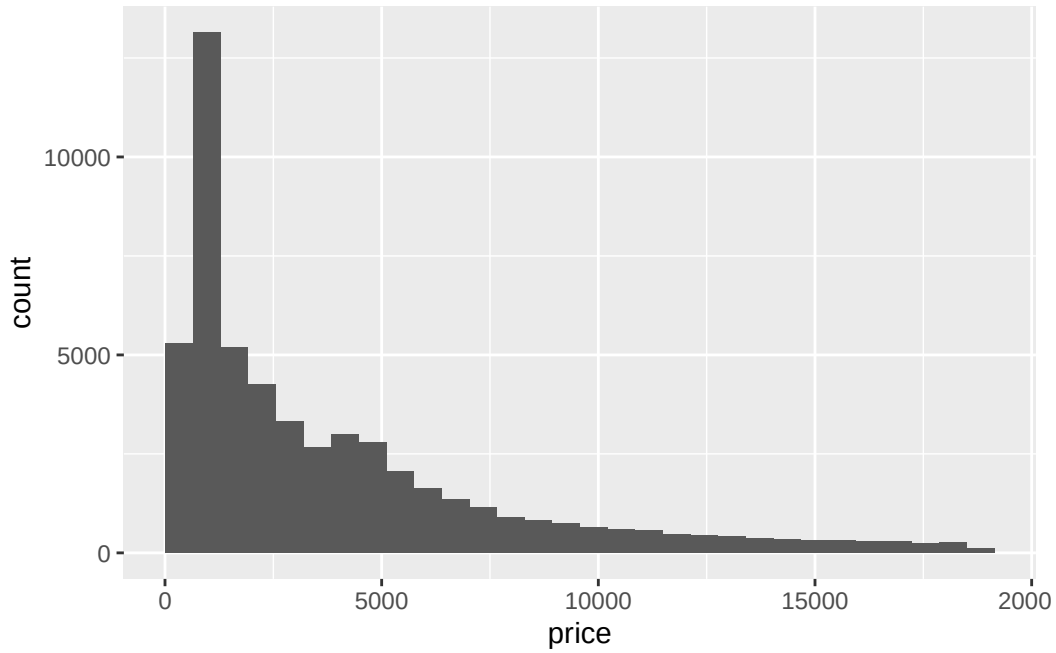
```
data(diamonds)
```

```
glimpse(diamonds)
```

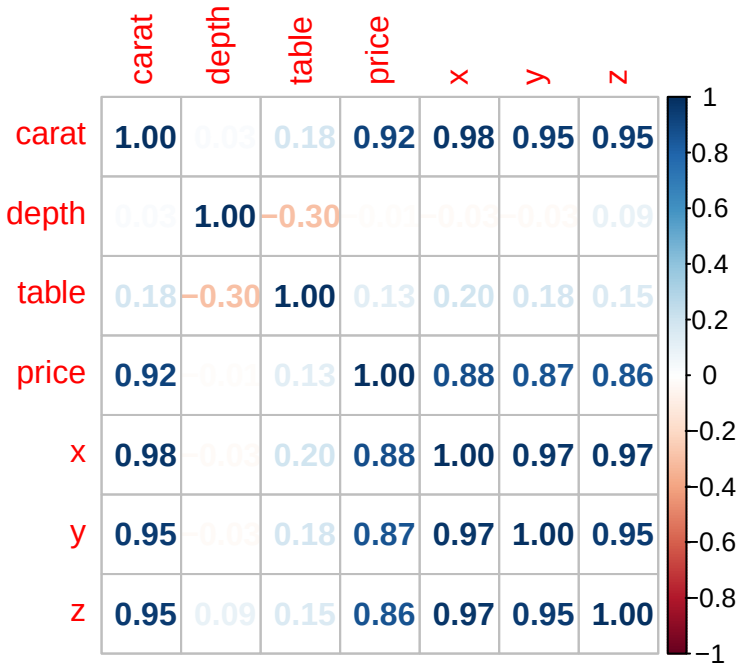
```
Rows: 53,940
Columns: 10
$ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
$ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
$ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
$ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
$ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
$ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
$ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
$ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
$ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
$ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

```
diamonds %>%
  ggplot (aes(price))+
  geom_histogram()
```

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




```
# I want to select only the columns that are numeric
diamonds %>%
  select(where(is.numeric)) %>%
  cor() %>%
  corrplot::corrplot(method="number")
```

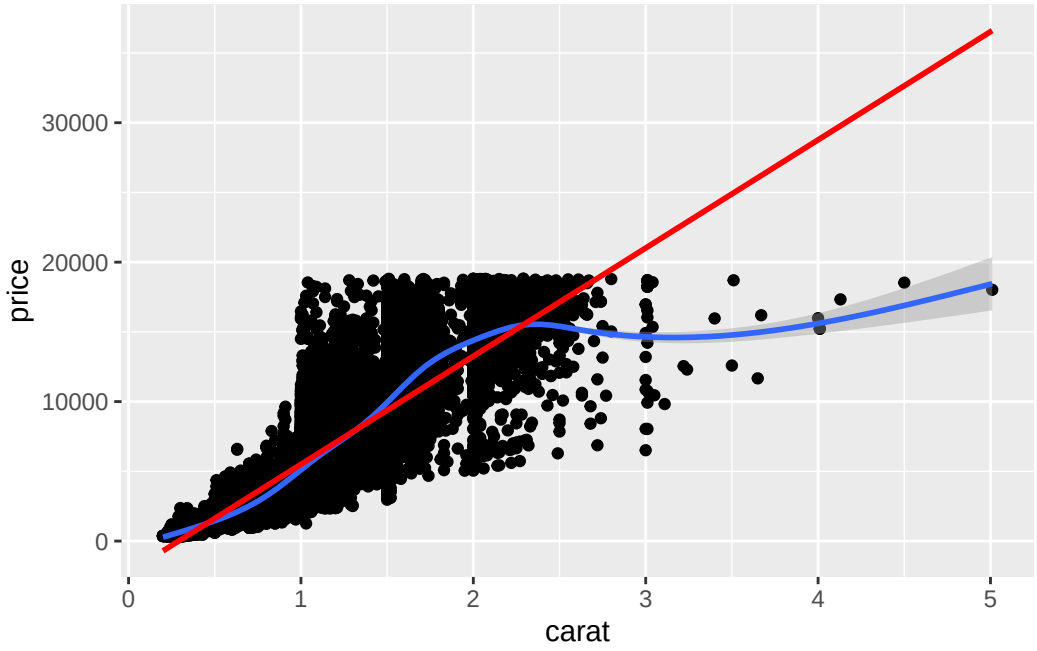


- variable with itself perfectly correlated
- carat is highly correlated with x,y,z –> including both could be redundant
- For forecasting is not a problem to keep adding variables. For econometrics adding variables can adds multicolinearity

Question 2

```
#scatterplot between price and carat, add a linearfit, and nonlinearfit
diamonds %>%
  ggplot(aes(carat,price))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm",se=FALSE, color="red")
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
`geom_smooth()` using formula = 'y ~ x'
```



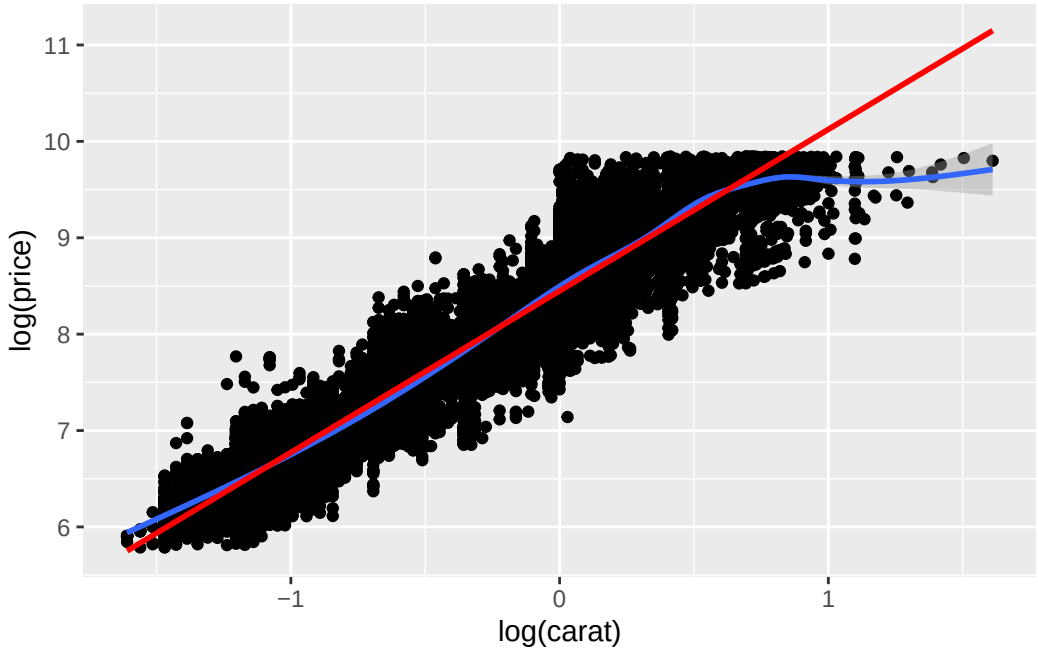
When carat (weight) is big, model differs a lot of being a linear model.

- we can fit a quadratic model (when we have non linear relationship) by adding in the linear model a quadratic variable
- when i increase corat by 1%, price ingrease by

Question 3

```
# transforming the model in logs
diamonds %>%
  ggplot(aes(log(carat),log(price)))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm",se=FALSE, color="red")
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
`geom_smooth()` using formula = 'y ~ x'
```




```
model1 = lm(log(price) ~ log(carat), data = diamonds)
model1$coefficients
```

```
(Intercept)  log(carat)
   8.448661    1.675817
```

```
model1[["coefficients"]]
```

```
(Intercept)  log(carat)
   8.448661    1.675817
```

```
coefficients(model1)
```

```
(Intercept)  log(carat)
   8.448661    1.675817
```

```
summary(model1)
```

```
Call:
```

```
lm(formula = log(price) ~ log(carat), data = diamonds)

Residuals:
     Min       1Q   Median       3Q      Max
-1.50833 -0.16951 -0.00591  0.16637  1.33793

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.448661   0.001365  6190.9   <2e-16 ***
log(carat)  1.675817   0.001934   866.6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2627 on 53938 degrees of freedom
Multiple R-squared:  0.933, Adjusted R-squared:  0.933
F-statistic: 7.51e+05 on 1 and 53938 DF,  p-value: < 2.2e-16
```

```
model2 = lm(log(price) ~ log(carat) + x +y + z, data = diamonds)
model2$coefficients
```

```
(Intercept)  log(carat)           x           y           z
 8.04035573  1.55629804  0.06925257  0.02420659 -0.04934297
```

```
model2[["coefficients"]]
```

```
(Intercept)  log(carat)           x           y           z
 8.04035573  1.55629804  0.06925257  0.02420659 -0.04934297
```

```
coefficients(model2)
```

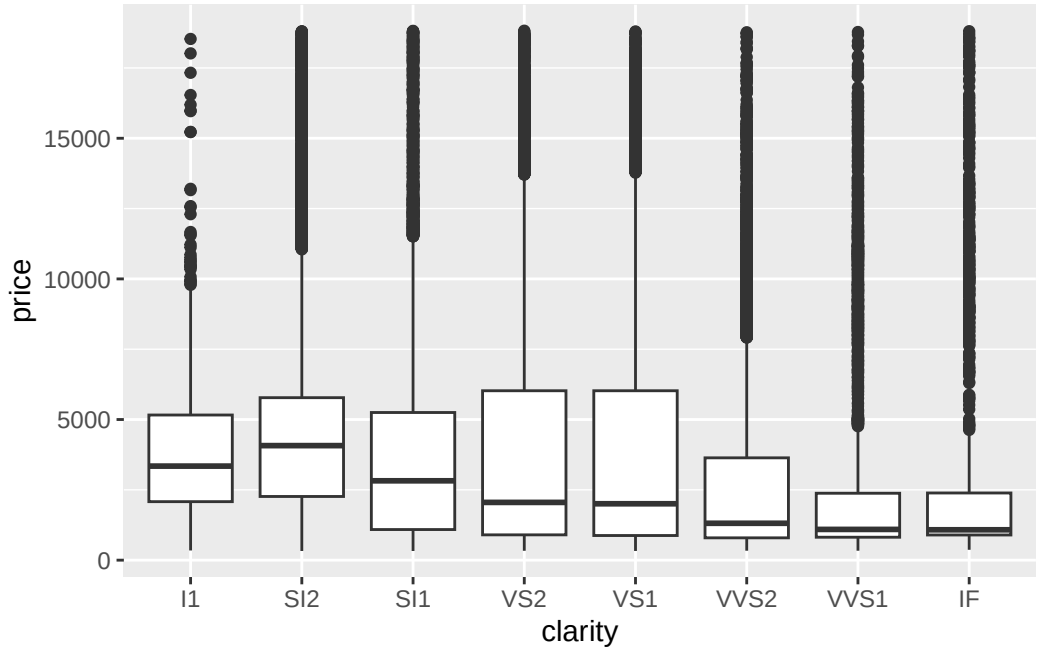
```
(Intercept)  log(carat)           x           y           z
 8.04035573  1.55629804  0.06925257  0.02420659 -0.04934297
```

```
summary(model2)
```

```

Call:
lm(formula = log(price) ~ log(carat) + x + y + z, data = diamonds)
```

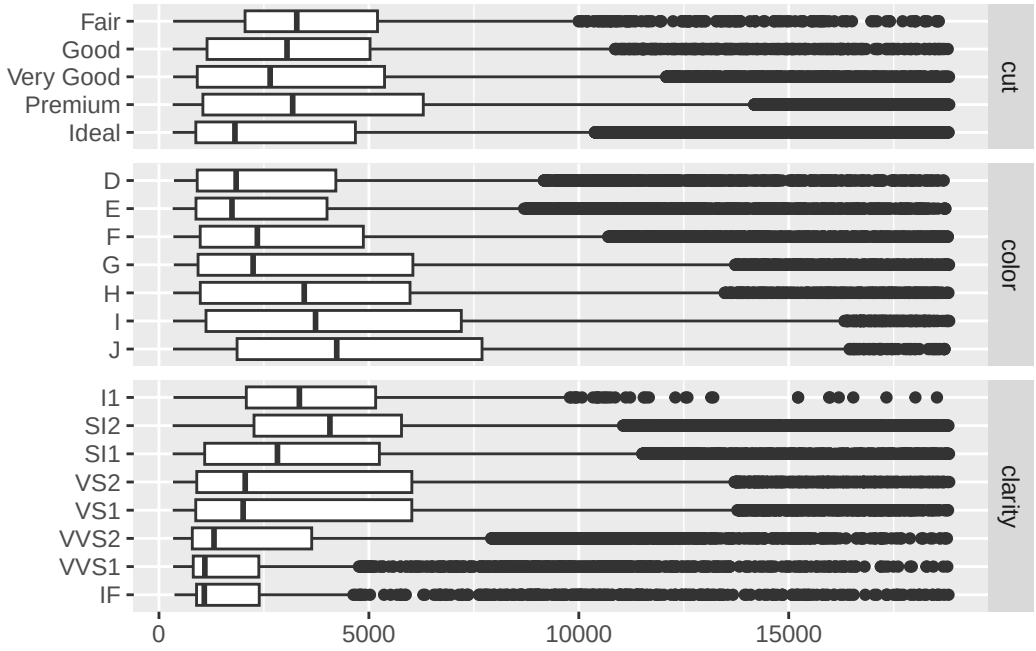
```
Residuals:
     Min       1Q   Median       3Q      Max
-1.54349 -0.16947 -0.00544  0.16689  1.68323

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.040356   0.048030 167.403  < 2e-16 ***
log(carat)   1.556298   0.014195 109.638  < 2e-16 ***
x            0.069253   0.008528   8.121 4.73e-16 ***
y            0.024207   0.004450   5.440 5.36e-08 ***
z           -0.049343   0.006891  -7.161 8.12e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2622 on 53935 degrees of freedom
Multiple R-squared:  0.9332,    Adjusted R-squared:  0.9332
F-statistic: 1.884e+05 on 4 and 53935 DF,  p-value: < 2.2e-16
```

```
diamonds %>%
  ggplot(aes(clarity,price))+
  geom_boxplot()
```



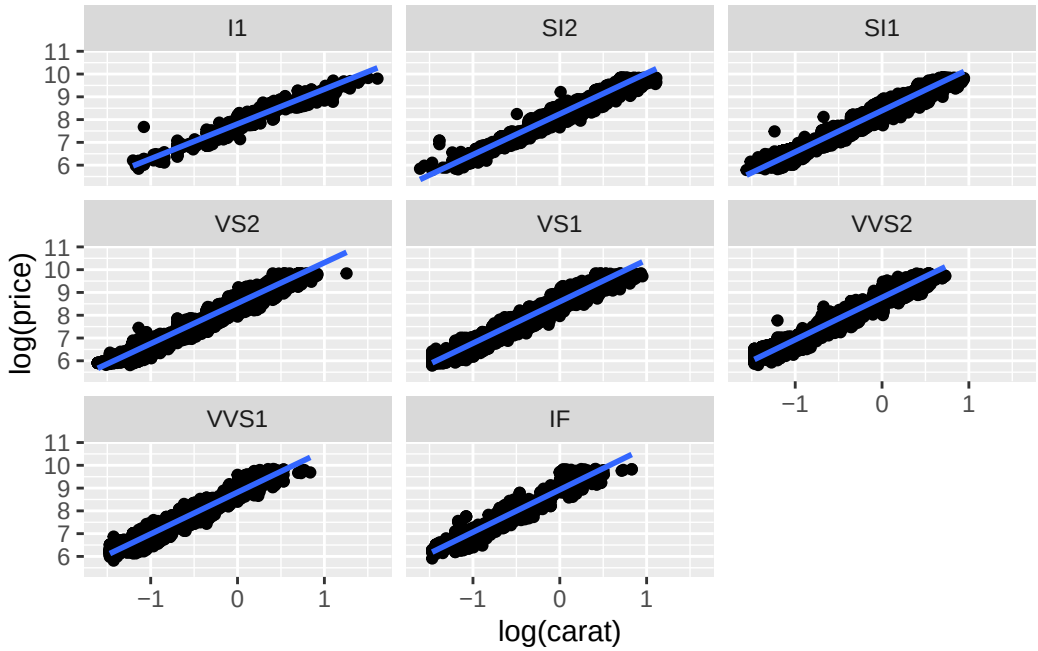
- clarity is an order factor

```
diamonds %>%
  select(c(price,cut,color,clarity)) %>%
  GGally::ggbivariate(outcome="price")
```



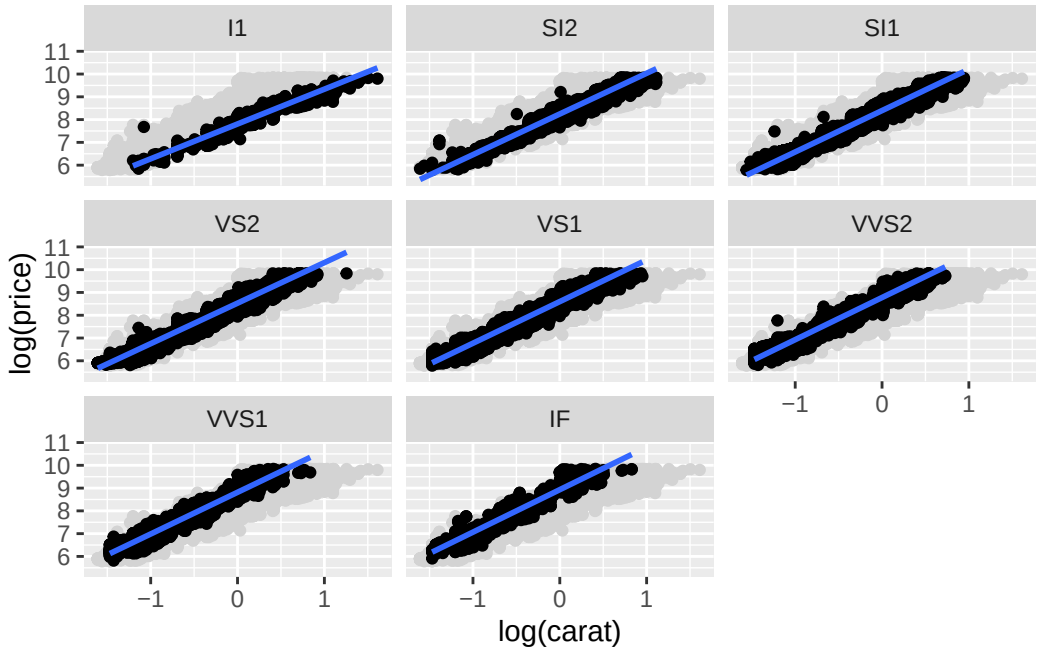

```
diamonds %>%
  ggplot(aes(log(carat),log(price)))+
  geom_point()+
  geom_smooth(method="lm", se=FALSE)+
  facet_wrap(vars(clarity))
```

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




```
diamonds %>%
  ggplot(aes(log(carat),log(price)))+
  geom_point(data=mutate(diamonds,clarity=NULL), color="lightgrey") +
  geom_point()+
  geom_smooth(method="lm", se=FALSE)+
  facet_wrap(vars(clarity))
```

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




```
#generate all formulas
vars = c("color", "cut", "clarity")
interactions = map_chr(vars, \(x) str_c("carat:", x))
vars = c(vars, interactions)

#combinations
combinations = map(0:6, \(x) combn(vars, x, simplify = FALSE)) %>%
  unlist(recursive = FALSE)
formulas = map_chr(combinations, \(x) paste(c("price ~ carat", x), collapse = ' + ' ))
formulas
```

```
 [1] "price ~ carat"
 [2] "price ~ carat + color"
 [3] "price ~ carat + cut"
 [4] "price ~ carat + clarity"
 [5] "price ~ carat + carat:color"
 [6] "price ~ carat + carat:cut"
 [7] "price ~ carat + carat:clarity"
 [8] "price ~ carat + color + cut"
 [9] "price ~ carat + color + clarity"
[10] "price ~ carat + color + carat:color"
[11] "price ~ carat + color + carat:cut"
[12] "price ~ carat + color + carat:clarity"
```

```
[13] "price ~ carat + cut + clarity"
[14] "price ~ carat + cut + carat:color"
[15] "price ~ carat + cut + carat:cut"
[16] "price ~ carat + cut + carat:clarity"
[17] "price ~ carat + clarity + carat:color"
[18] "price ~ carat + clarity + carat:cut"
[19] "price ~ carat + clarity + carat:clarity"
[20] "price ~ carat + carat:color + carat:cut"
[21] "price ~ carat + carat:color + carat:clarity"
[22] "price ~ carat + carat:cut + carat:clarity"
[23] "price ~ carat + color + cut + clarity"
[24] "price ~ carat + color + cut + carat:color"
[25] "price ~ carat + color + cut + carat:cut"
[26] "price ~ carat + color + cut + carat:clarity"
[27] "price ~ carat + color + clarity + carat:color"
[28] "price ~ carat + color + clarity + carat:cut"
[29] "price ~ carat + color + clarity + carat:clarity"
[30] "price ~ carat + color + carat:color + carat:cut"
[31] "price ~ carat + color + carat:color + carat:clarity"
[32] "price ~ carat + color + carat:cut + carat:clarity"
[33] "price ~ carat + cut + clarity + carat:color"
[34] "price ~ carat + cut + clarity + carat:cut"
[35] "price ~ carat + cut + clarity + carat:clarity"
[36] "price ~ carat + cut + carat:color + carat:cut"
[37] "price ~ carat + cut + carat:color + carat:clarity"
[38] "price ~ carat + cut + carat:cut + carat:clarity"
[39] "price ~ carat + clarity + carat:color + carat:cut"
[40] "price ~ carat + clarity + carat:color + carat:clarity"
[41] "price ~ carat + clarity + carat:cut + carat:clarity"
[42] "price ~ carat + carat:color + carat:cut + carat:clarity"
[43] "price ~ carat + color + cut + clarity + carat:color"
[44] "price ~ carat + color + cut + clarity + carat:cut"
[45] "price ~ carat + color + cut + clarity + carat:clarity"
[46] "price ~ carat + color + cut + carat:color + carat:cut"
[47] "price ~ carat + color + cut + carat:color + carat:clarity"
[48] "price ~ carat + color + cut + carat:cut + carat:clarity"
[49] "price ~ carat + color + clarity + carat:color + carat:cut"
[50] "price ~ carat + color + clarity + carat:color + carat:clarity"
[51] "price ~ carat + color + clarity + carat:cut + carat:clarity"
[52] "price ~ carat + color + carat:color + carat:cut + carat:clarity"
[53] "price ~ carat + cut + clarity + carat:color + carat:cut"
[54] "price ~ carat + cut + clarity + carat:color + carat:clarity"
[55] "price ~ carat + cut + clarity + carat:cut + carat:clarity"
```

```
[56] "price ~ carat + cut + carat:color + carat:cut + carat:clarity"
[57] "price ~ carat + clarity + carat:color + carat:cut + carat:clarity"
[58] "price ~ carat + color + cut + clarity + carat:color + carat:cut"
[59] "price ~ carat + color + cut + clarity + carat:color + carat:clarity"
[60] "price ~ carat + color + cut + clarity + carat:cut + carat:clarity"
[61] "price ~ carat + color + cut + carat:color + carat:cut + carat:clarity"
[62] "price ~ carat + color + clarity + carat:color + carat:cut + carat:clarity"
[63] "price ~ carat + cut + clarity + carat:color + carat:cut + carat:clarity"
[64] "price ~ carat + color + cut + clarity + carat:color + carat:cut + carat:clarity"
```

For each we estimate a model.

```
models <- map(formulas, ~ lm(as.formula(.x), data = diamonds))
```

Adjust the model using r squared.

```
model_fit <- tibble(
  formula = formulas,
  adj_r2 = map_dbl(models, ~ summary(.x)$adj.r.squared))
```

Select the best model. The higher the r squared.

```
adj_r2 <- sapply(models, function(m) summary(m)$adj.r.squared)
adj_r2
```

```
 [1] 0.8493277 0.8639443 0.8564615 0.8948397 0.8667060 0.8589129 0.9074577
 [8] 0.8711236 0.9139389 0.8671456 0.8736305 0.9257206 0.8969067 0.8743178
[15] 0.8590409 0.9095368 0.9180574 0.8987755 0.9086351 0.8763832 0.9279484
[22] 0.9097983 0.9159125 0.8746546 0.8737691 0.9277724 0.9188132 0.9178397
[29] 0.9272917 0.8767774 0.9286880 0.9280337 0.9202740 0.8992556 0.9106958
[36] 0.8765192 0.9303260 0.9098475 0.9219507 0.9296780 0.9109199 0.9303729
[43] 0.9209487 0.9184091 0.9292677 0.8769177 0.9309615 0.9280881 0.9226380
[50] 0.9304615 0.9295507 0.9310510 0.9222597 0.9319118 0.9110040 0.9305281
[57] 0.9320714 0.9229881 0.9325929 0.9296409 0.9311778 0.9327848 0.9322048
[64] 0.9329026
```

```
best_index <- which.max(adj_r2)
best_model <- models[[best_index]]

best_formula <- formulas[[best_index]]
best_adj_r2 <- adj_r2[[best_index]]
```

```
best_index
```

```
[1] 64
```

```
best_model
```

```

Call:
lm(formula = as.formula(.x), data = diamonds)

Coefficients:
    (Intercept)            carat          color.L          color.Q
      -3197.730         8980.499         -607.063          116.477
        color.C          color^4          color^5          color^6
        -72.772           63.356          213.850            9.141
          cut.L            cut.Q            cut.C            cut^4
        367.389         -286.218          203.386         -127.540
      clarity.L        clarity.Q        clarity.C        clarity^4
        -29.305          573.593         -989.707          656.623
      clarity^5        clarity^6        clarity^7    carat:color.L
       -510.237          207.319          -62.421        -1361.475
  carat:color.Q    carat:color.C    carat:color^4    carat:color^5
       -646.960           32.168          -34.588         -311.501
  carat:color^6      carat:cut.L      carat:cut.Q      carat:cut.C
        -25.463          374.107           56.280          -69.648
    carat:cut^4  carat:clarity.L  carat:clarity.Q  carat:clarity.C
        186.820         4831.965        -1179.967         1490.792
carat:clarity^4  carat:clarity^5  carat:clarity^6  carat:clarity^7
       -725.051          598.171         -108.650          255.707
```

```
best_formula
```

```
[1] "price ~ carat + color + cut + clarity + carat:color + carat:cut + carat:clarity"
```

```
best_adj_r2
```

```
[1] 0.9329026
```

All possible linear models including `carat`, `color`, `cut`, `clarity`, and their interactions with `carat` were fitted and compared using the adjusted R^2.
The best-performing model included all main effects and all interactions with `carat`, achieving an adjusted R^2 of 0.933.
This indicates that the model explains a large proportion of the variability in diamond prices.
The inclusion of interaction terms suggests that the effect of carat on price depends on color, cut, and clarity.
Predicted prices closely match the observed values, indicating a strong overall model fit.