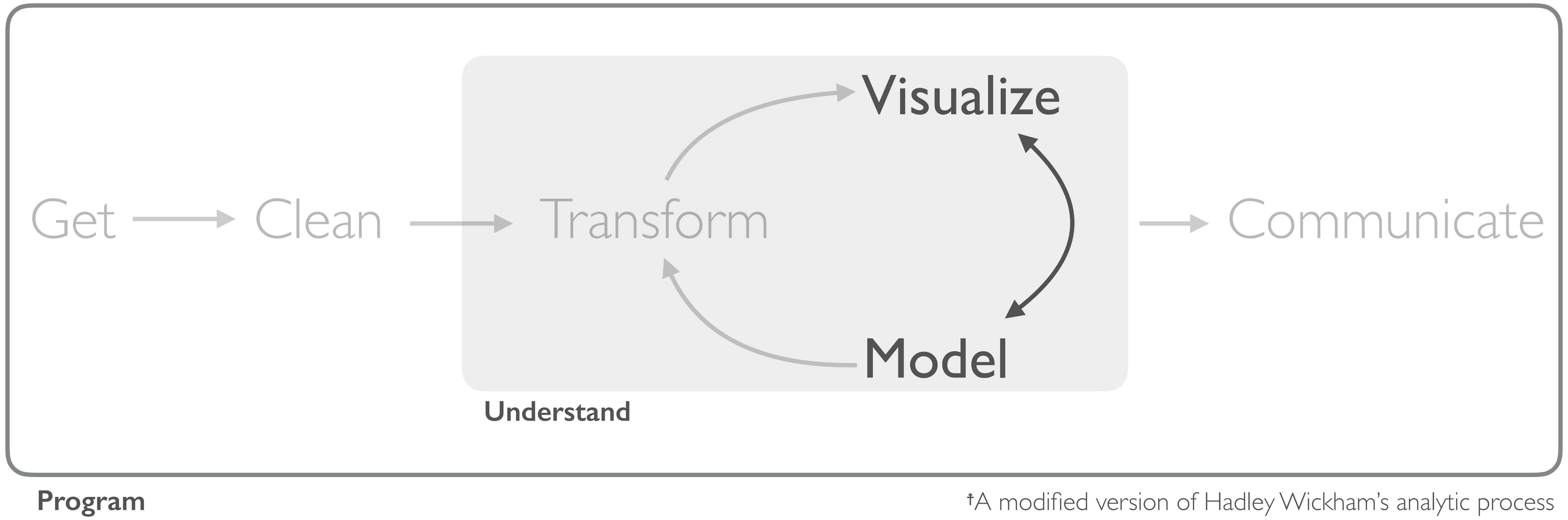
MODEL BUILDING



†A modified version of Hadley Wickham's analytic process

PREREQUISITES

PREREQUISITES

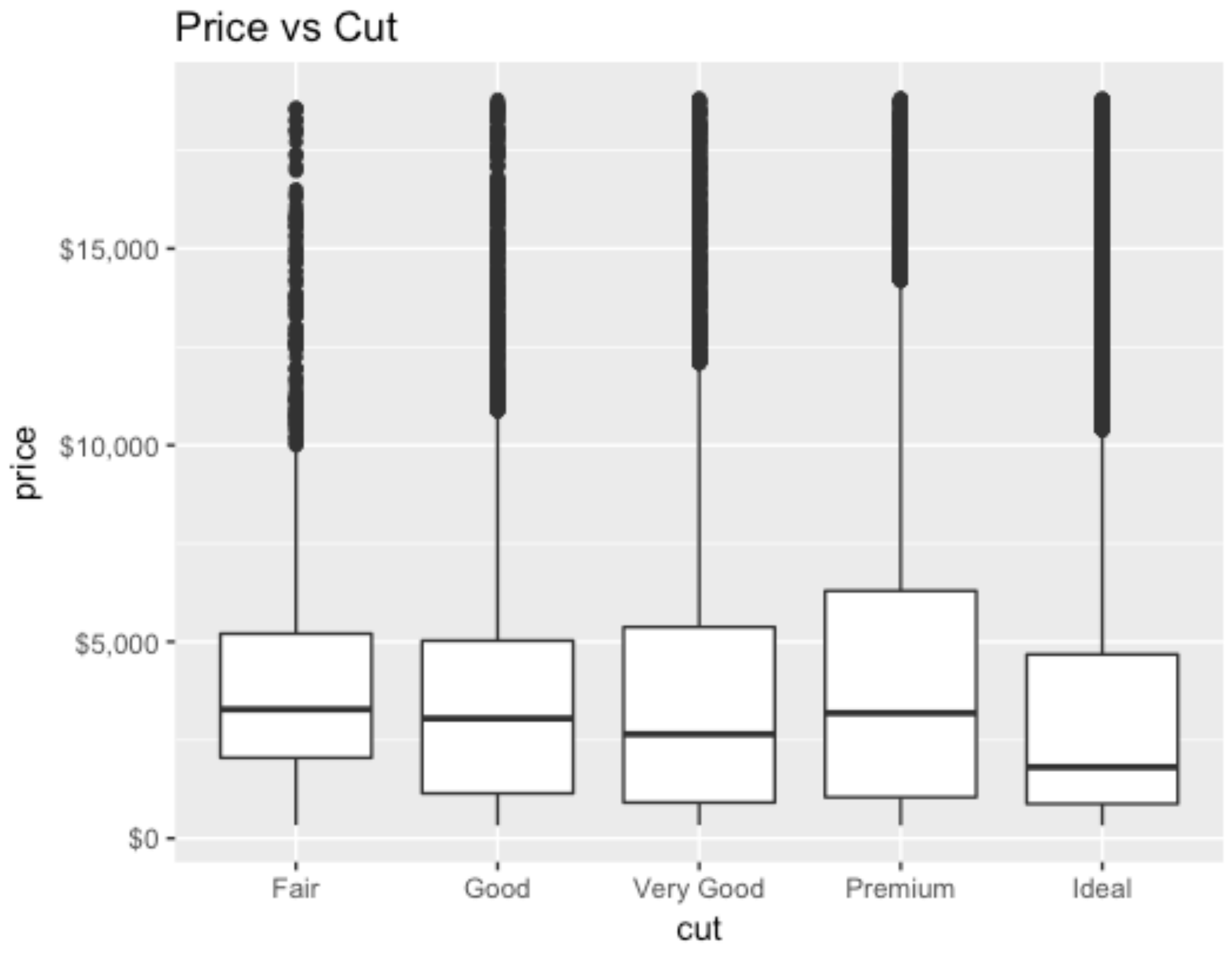
```
library(tidyverse)
library(modelr)
options(na.action = na.warn)
```

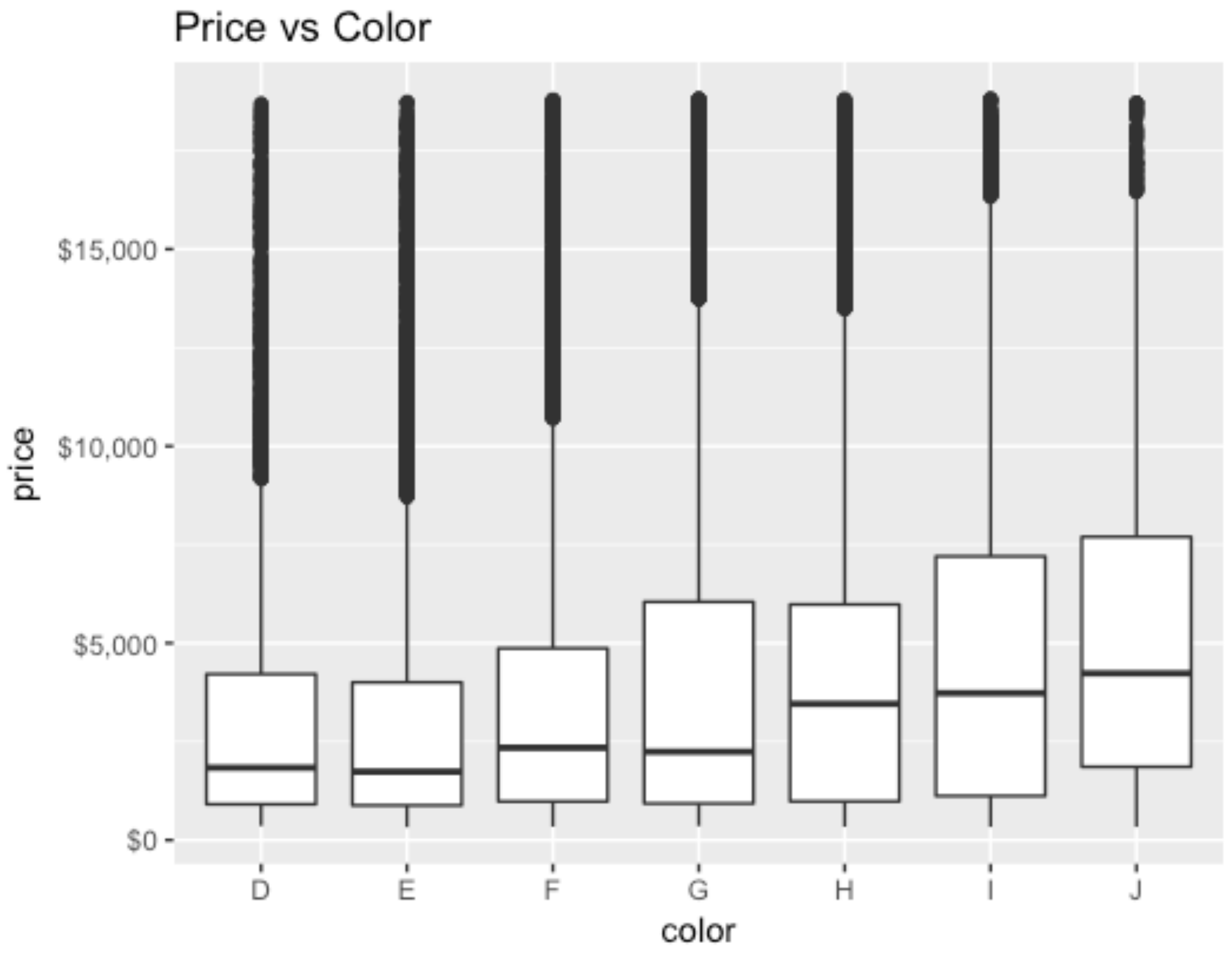
THE SET-UP

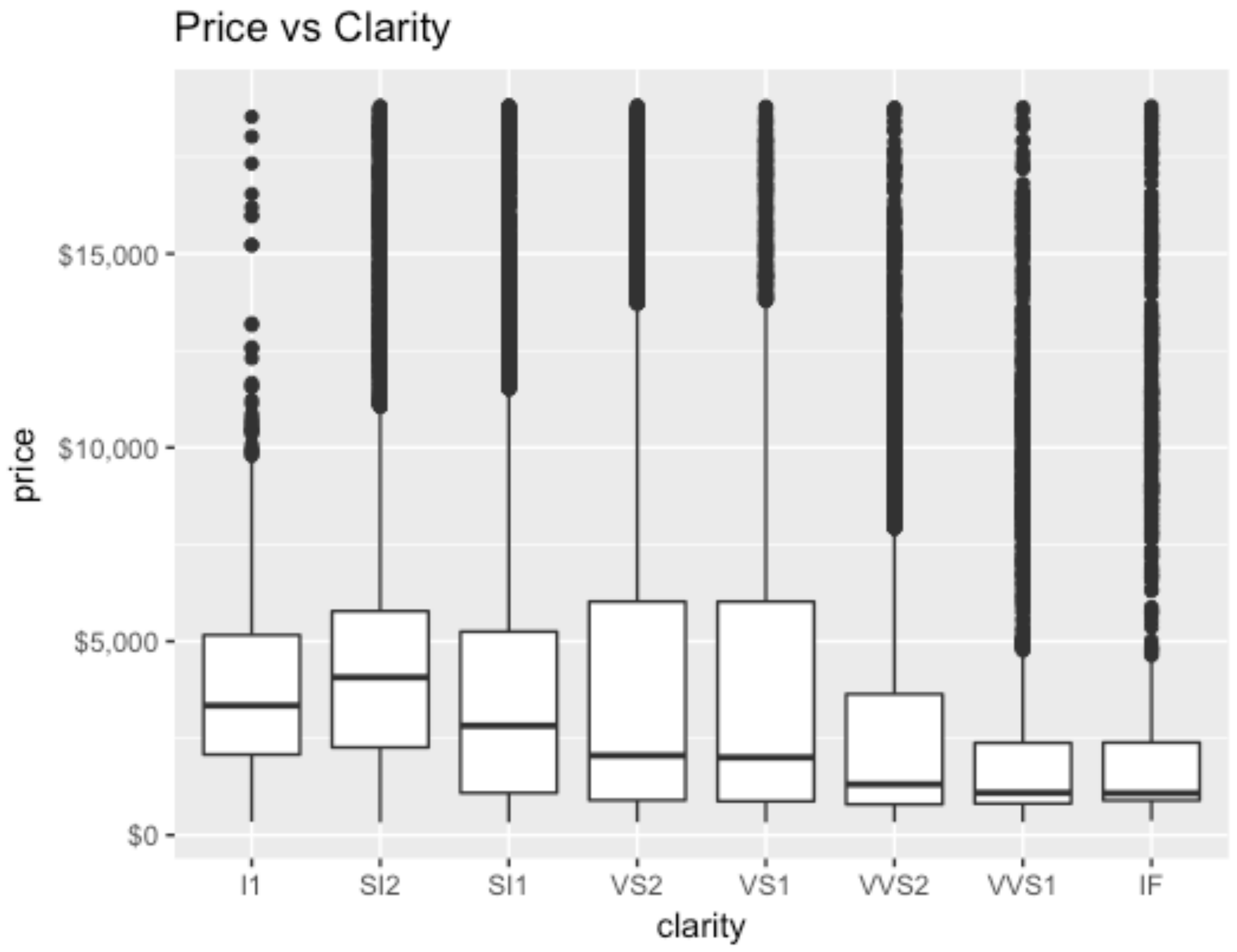

DIAMOND JEWELRY III
PAWN SHOP
718-220-5355
WE BUY GOLD & ELECTRONICS
42W.
46 WEST
BEAUTY
WE PAWN
WE BUY CELL PHONES
OPEN

WHY ARE INFERIOR DIAMONDS MORE EXPENSIVE?

- Another analyst provided your boss with these three charts from your **`diamonds`** data set

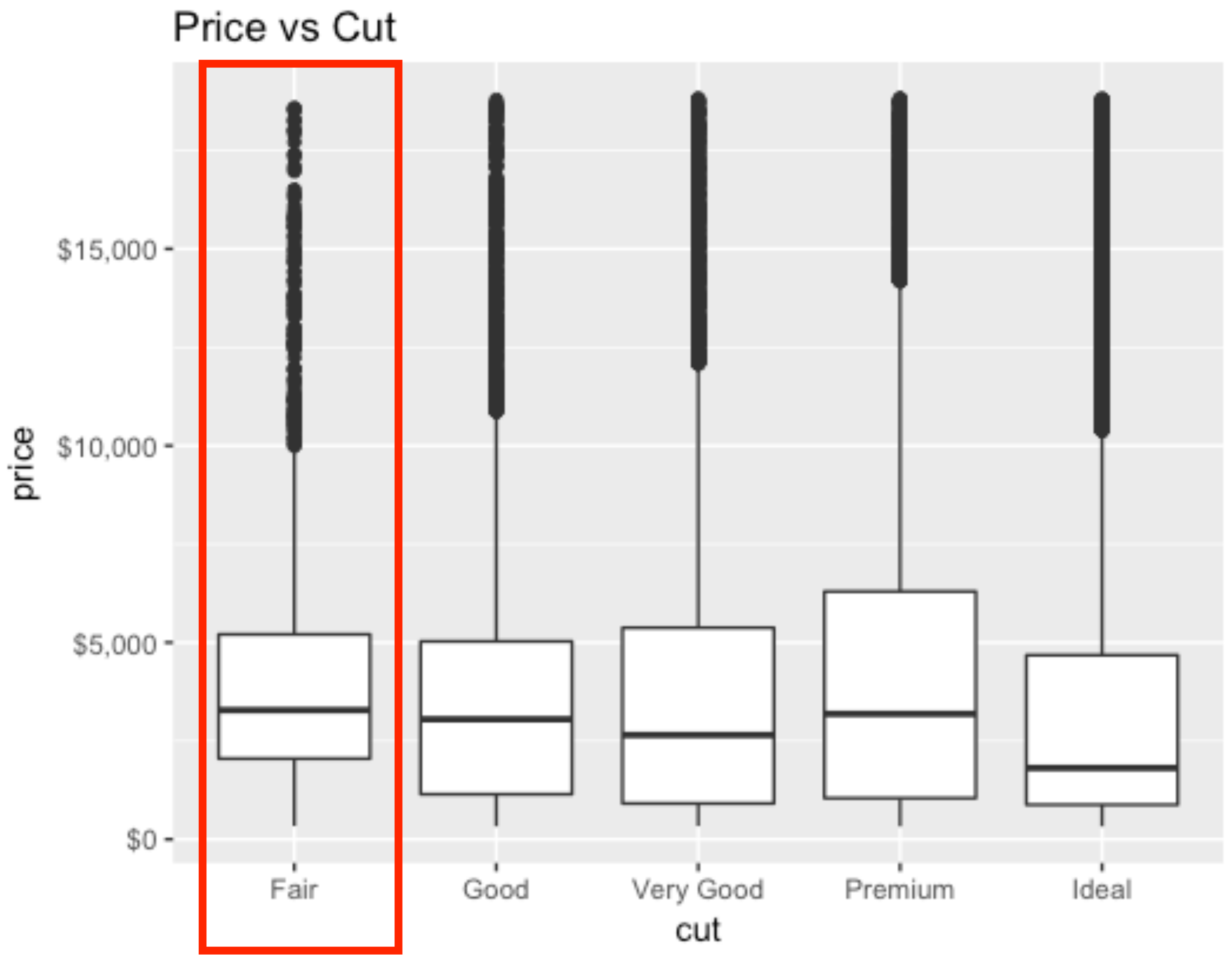


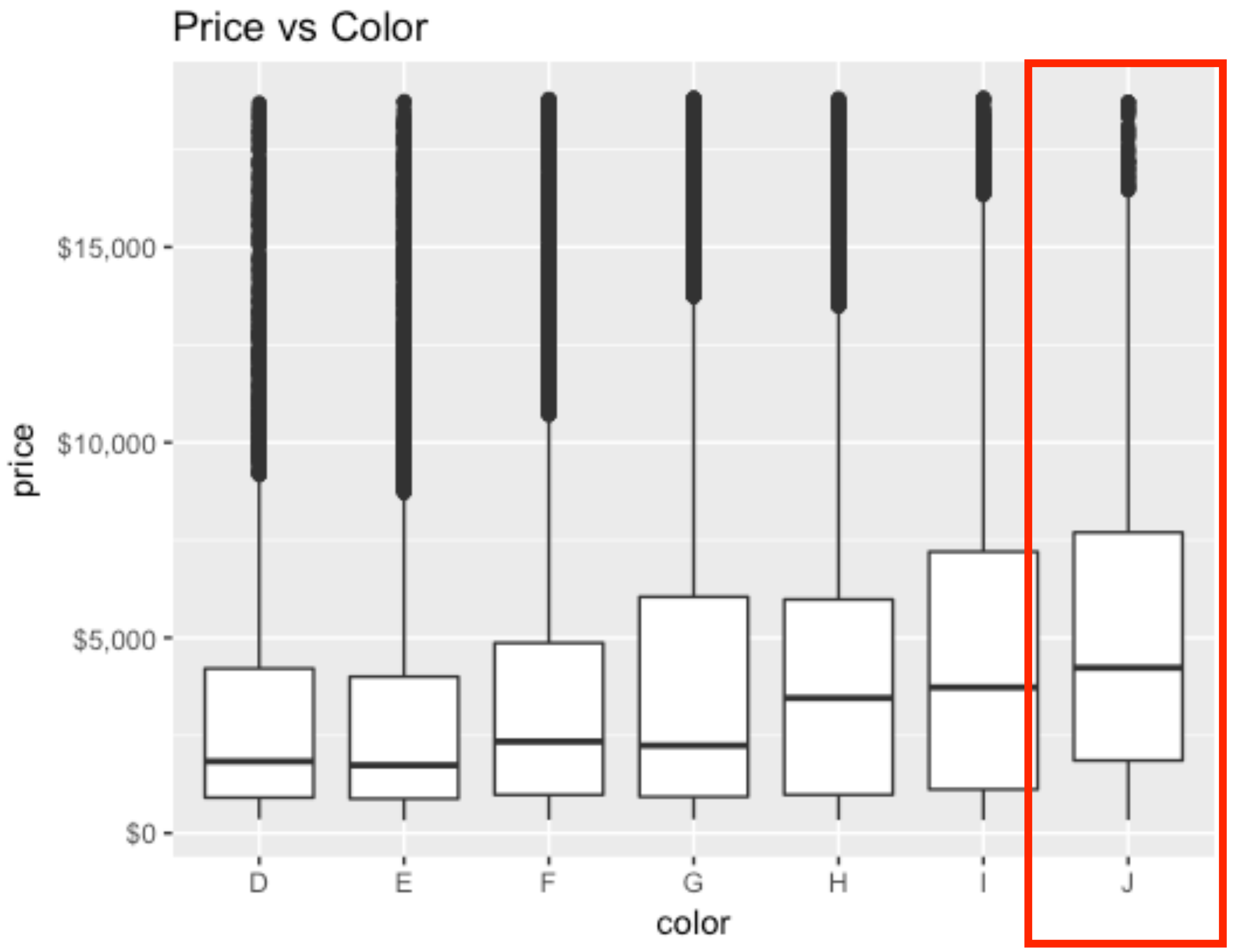


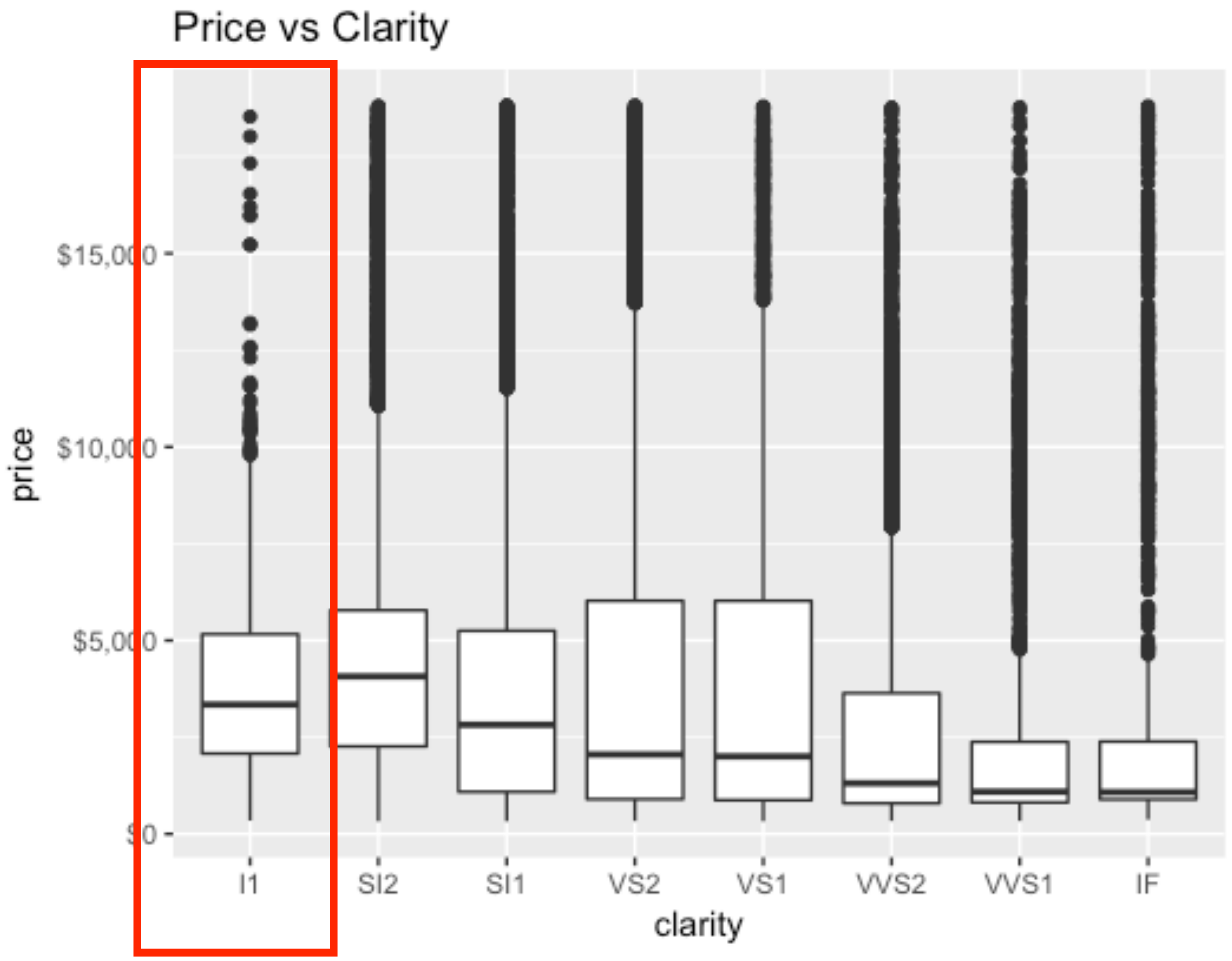


WHY ARE INFERIOR DIAMONDS MORE EXPENSIVE?

- Another analyst provided your boss with these three charts from your `diamonds` data set
- This led to your boss wondering why inferior diamonds are more expensive







YOUR TURN!

Spend a few minutes discussing the logic behind this with your neighbor

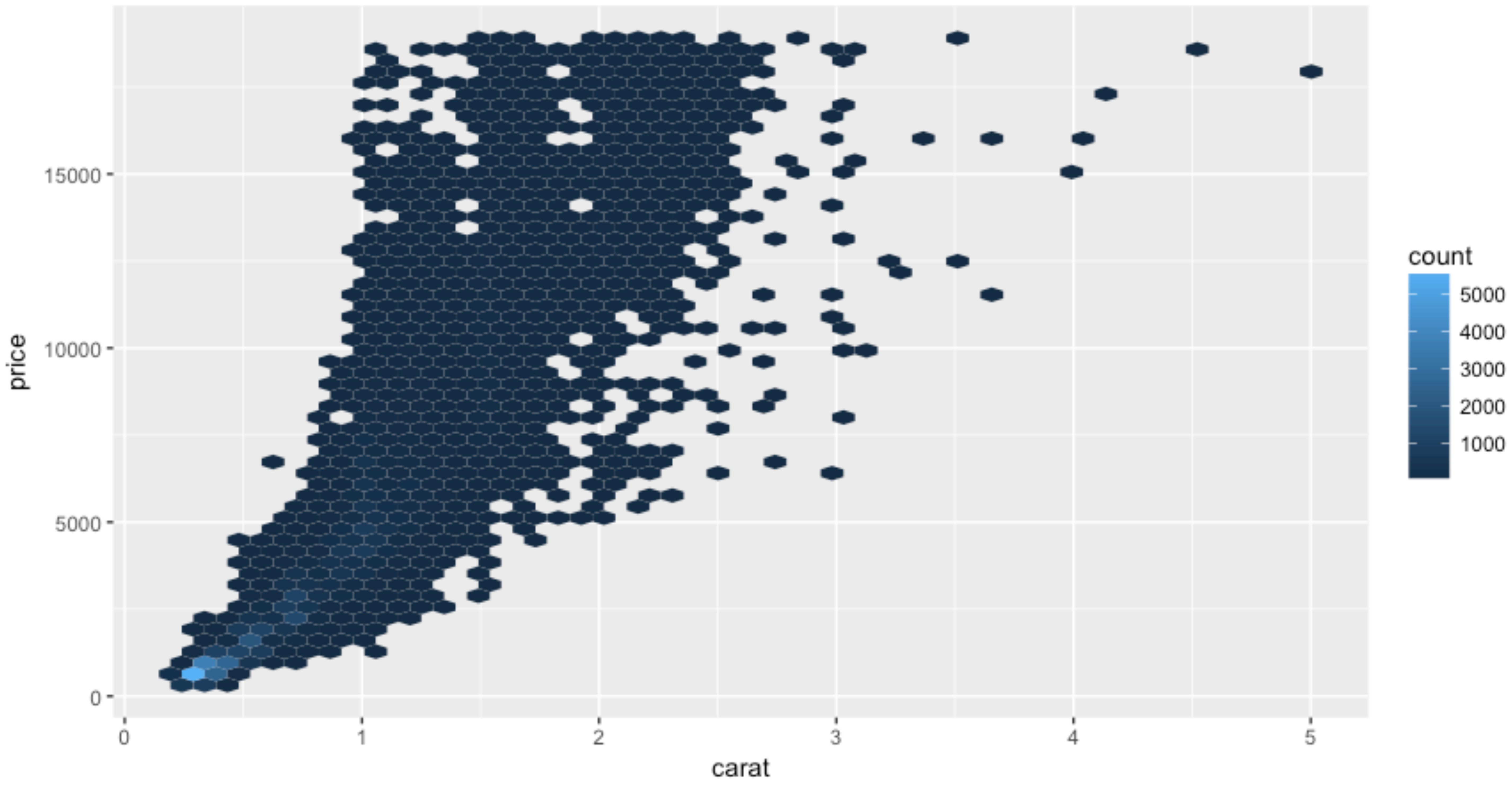
Feel free to explore the `diamonds` *data set*

THOUGHTS?

A MAJOR CONFOUNDING VARIABLE

CONFOUNDING VARIABLE

```
ggplot(diamonds, aes(carat, price)) +
  geom_hex(bins = 50)
```

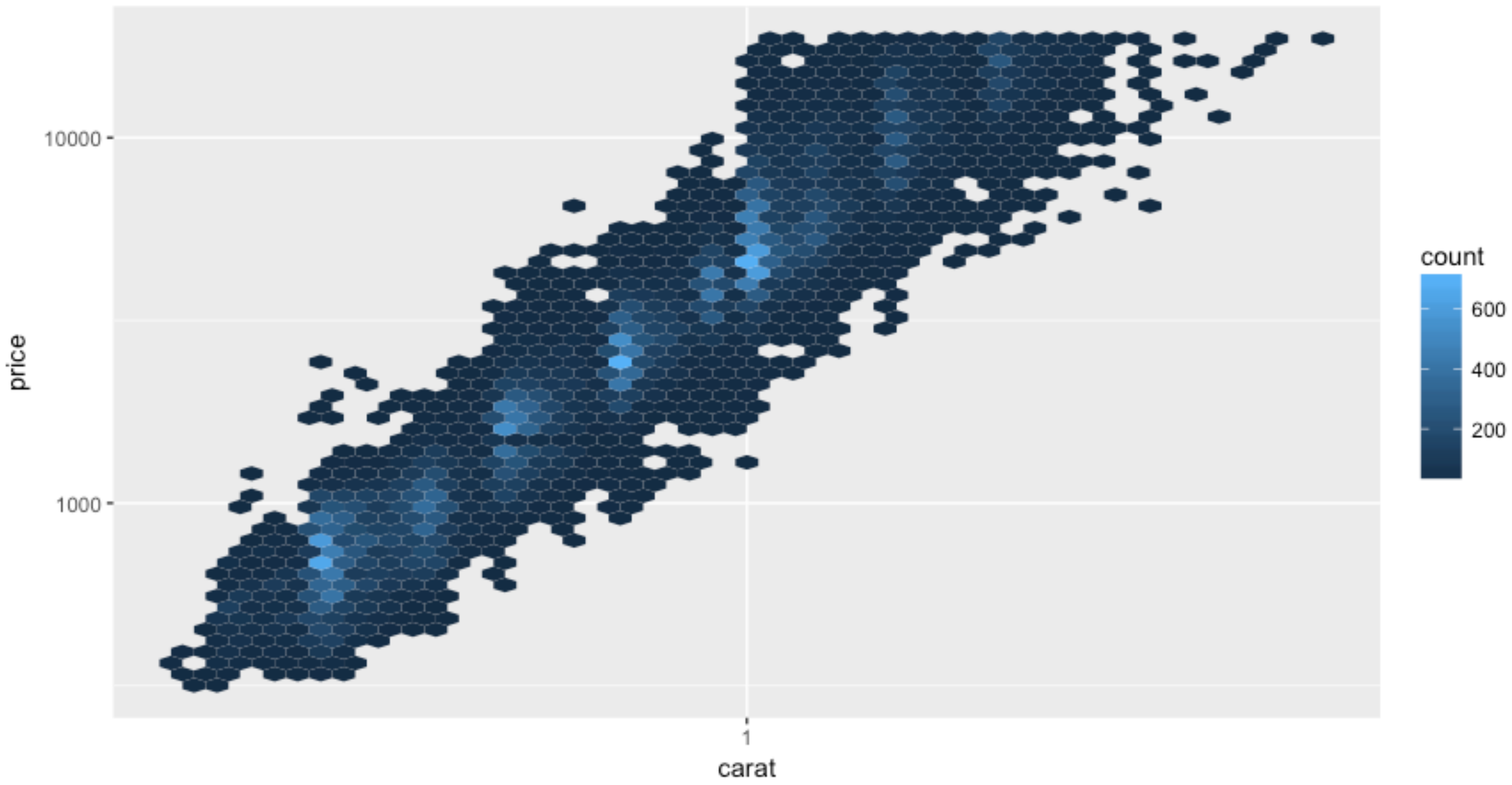


The `carat` variable has a big impact on price but is not captured in the previous 2-dimension plots

The relationship is non-linear. How could you transform the variables to assess a linear relationship?

CONFOUNDING VARIABLE

```
ggplot(diamonds, aes(carat, price)) +
  geom_hex(bins = 50) +
  scale_x_log10() +
  scale_y_log10()
```



The `carat` variable has a big impact on price but is not captured in the previous 2-dimension plots

The relationship is non-linear. How could you transform the variables to assess a linear relationship?

YOUR TURN - PART 1!

1. *Can you measure the strength of this linear relationship?*
2. *Does the strength of the linear relationship differ depending on the different levels of cut, color, and clarity?*

YOUR TURN - PART 1!

1. *Can you measure the strength of this linear relationship?*
2. *Does the strength of the linear relationship differ depending on the different levels of cut, color, and clarity?*

SOLUTION

```
# what is the strength of this linear relationship?
cor.test(log10(diamonds$carat), log10(diamonds$price))

Pearson's product-moment correlation

data:  log10(diamonds$carat) and log10(diamonds$price)
t = 866.59, df = 53938, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9653436 0.9664747
sample estimates:
      cor
0.9659137
```

YOUR TURN - PART 1!

1. *Can you measure the strength of this linear relationship?*
2. *Does the strength of the linear relationship differ depending on the different levels of cut, color, and clarity?*

SOLUTION

```
# Does it differ depending on the different levels of cut, color, and clarity
diamonds %>%
  group_by(cut) %>%
  summarise(corr = cor(log10(carat), log10(price)),
            p_value = cor.test(log10(carat), log10(price))$p.value)
# A tibble: 5 × 3
        cut      corr p_value
      <ord>     <dbl>   <dbl>
1      Fair 0.9085131       0
2      Good 0.9687510       0
3 Very Good 0.9716746       0
4   Premium 0.9697578       0
5     Ideal 0.9661884       0
```

YOUR TURN - PART 2!

1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

YOUR TURN - PART 2!

1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

SOLUTION

```
# step 1: fit model
mod_carat <- lm(log10(price) ~ log10(carat), data = diamonds)
```

YOUR TURN - PART 2!

1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

SOLUTION

```
# step 2: assess model numerically
summary(mod_carat)

Call:
lm(formula = log10(price) ~ log10(carat), data = diamonds)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.65506 -0.07362 -0.00257  0.07225  0.58106 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.6692067  0.0005927  6190.9   <2e-16 ***
log10(carat) 1.6758167  0.0019338   866.6   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

YOUR TURN - PART 2!

1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

SOLUTION

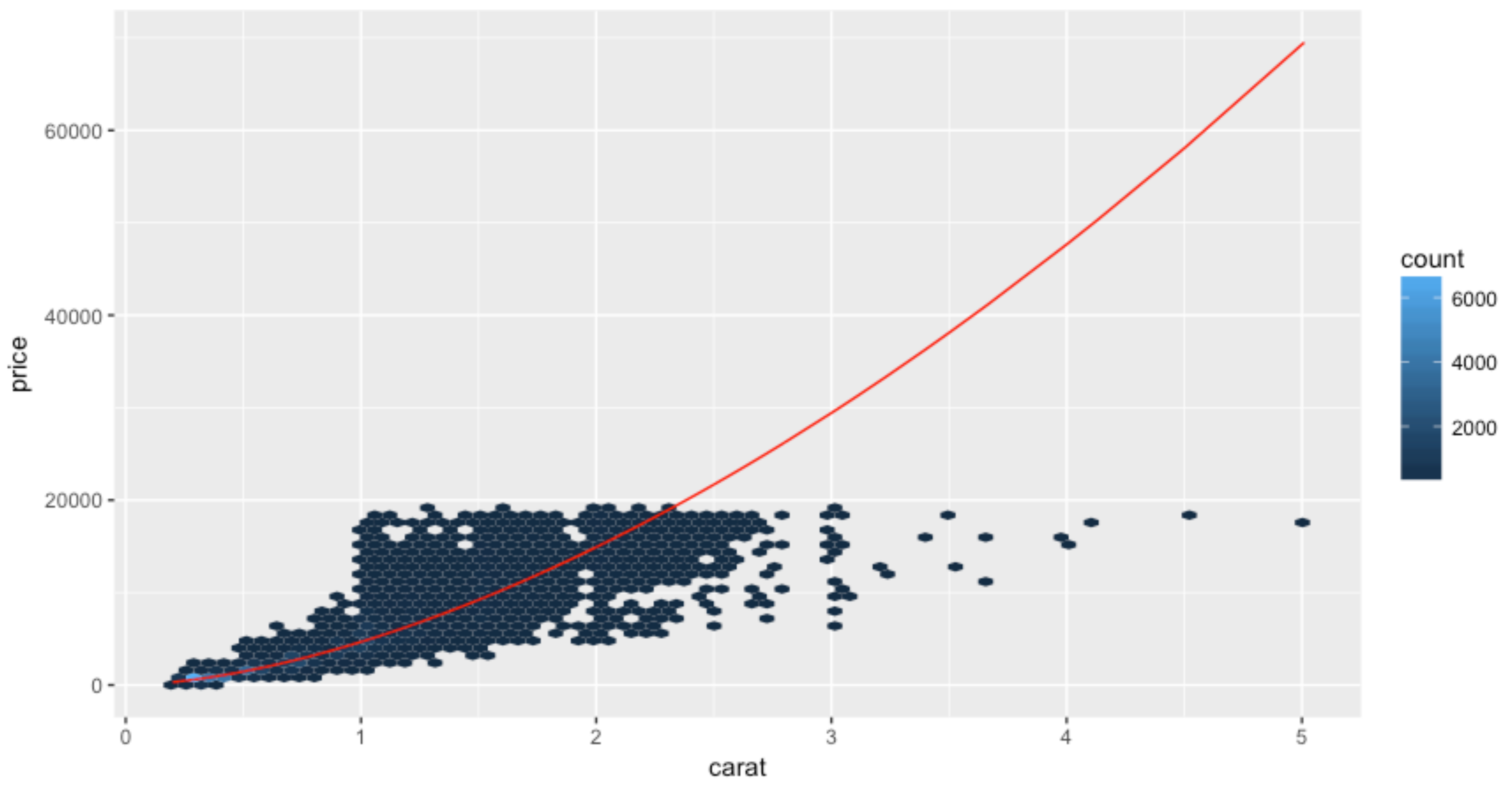
```
# step 3: get prediction and residual data
diamonds2 <- diamonds %>%
  add_predictions(mod_carat) %>%
  add_residuals(mod_carat) %>%
  mutate(trans_pred = 10 ^ pred)

diamonds2
# A tibble: 53,940 × 13
   carat       cut color clarity depth table price     x     y     z     pred       resid trans_pred
   <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>    <dbl>       <dbl>      <dbl>
1   0.23     Ideal     E     SI2  61.5    55   326  3.95  3.98  2.43 2.599580 -0.08636196   397.7220
2   0.21   Premium     E     SI1  59.8    61   326  3.89  3.84  2.31 2.533370 -0.02015289   341.4841
3   0.23      Good     E     VS1  56.9    65   327  4.05  4.07  2.31 2.599580 -0.08503181   397.7220
4   0.29   Premium     I     VS2  62.4    58   334  4.20  4.23  2.63 2.768284 -0.24453784   586.5220
5   0.31      Good     J     SI2  63.3    58   335  4.34  4.35  2.75 2.816822 -0.29177734   655.8766
6   0.24 Very Good     J    VVS2  62.8    57   336  3.94  3.96  2.48 2.630554 -0.10421509   427.1244
7   0.24 Very Good     I    VVS1  62.3    57   336  3.95  3.98  2.47 2.630554 -0.10421509   427.1244
8   0.26 Very Good     H     SI1  61.9    55   337  4.07  4.11  2.53 2.688809 -0.16117938   488.4378
```

YOUR TURN - PART 2!

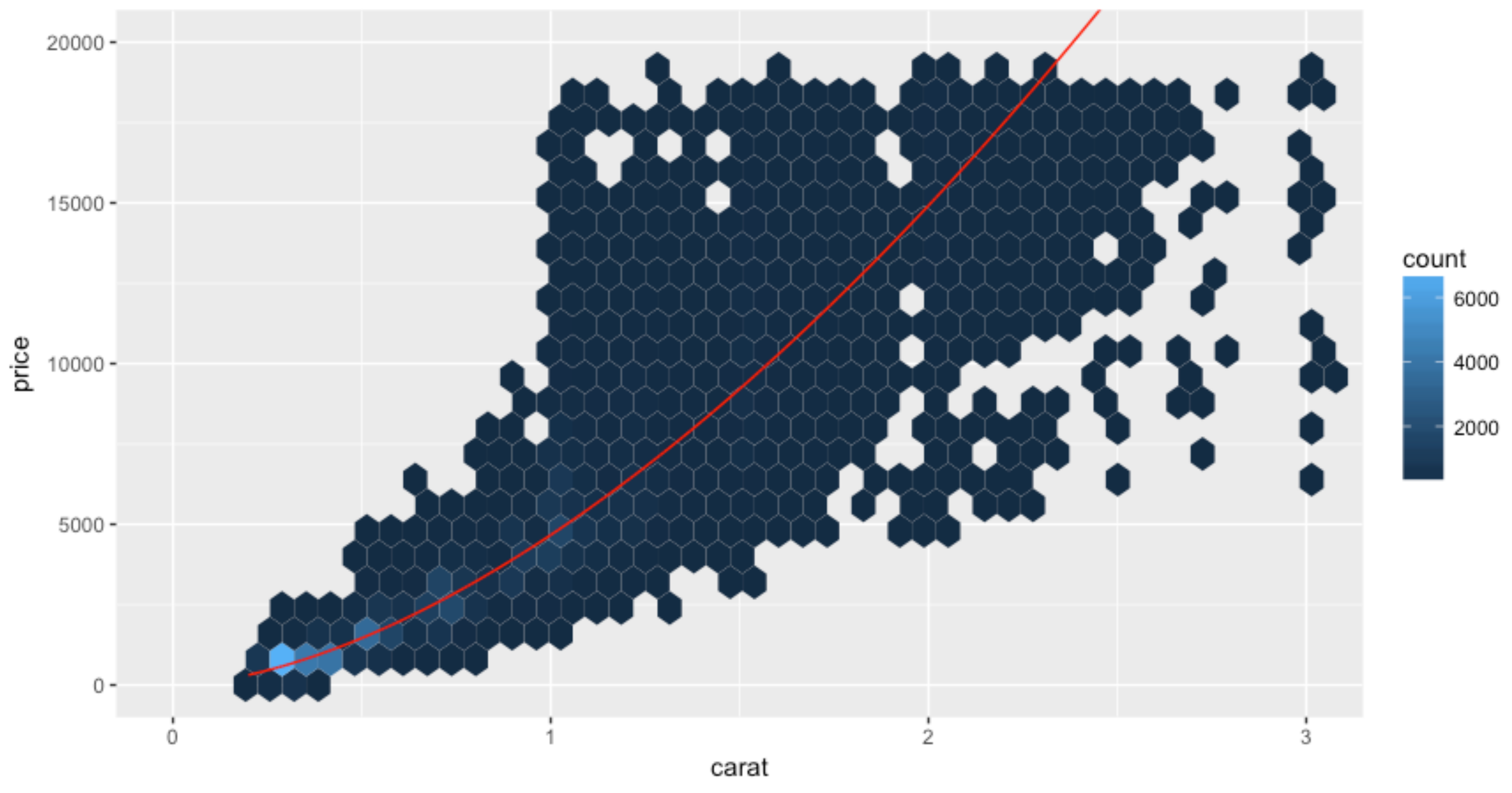
1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

SOLUTION

```r
# step 4: assess model predictions visually
ggplot(diamonds2, aes(carat, price)) +
  geom_hex(bins = 75) +
  geom_line(aes(y = trans_pred), color = "red")
```




```r
# step 4: assess model predictions visually
ggplot(diamonds2, aes(carat, price)) +
  geom_hex(bins = 75) +
  geom_line(aes(y = trans_pred), color = "red") +
  coord_cartesian(xlim = c(0, 3), ylim = c(0, 20000))
```

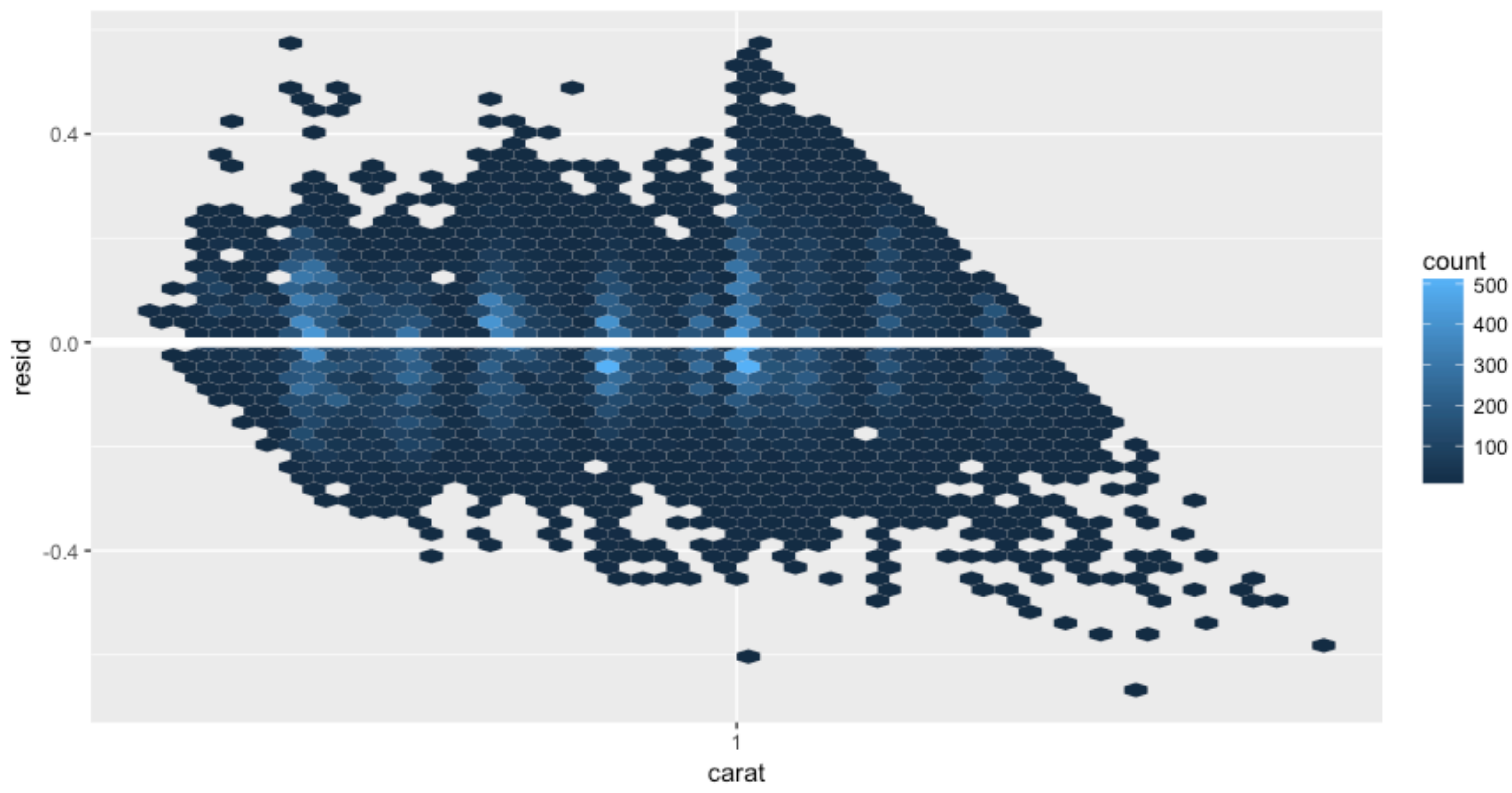


YOUR TURN - PART 2!

1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

SOLUTION

```
# step 5: assess model residuals visually
ggplot(diamonds2, aes(carat, resid)) +
  geom_hex(bins = 50) +
  geom_ref_line(h = 0) +
  scale_x_log10()
```

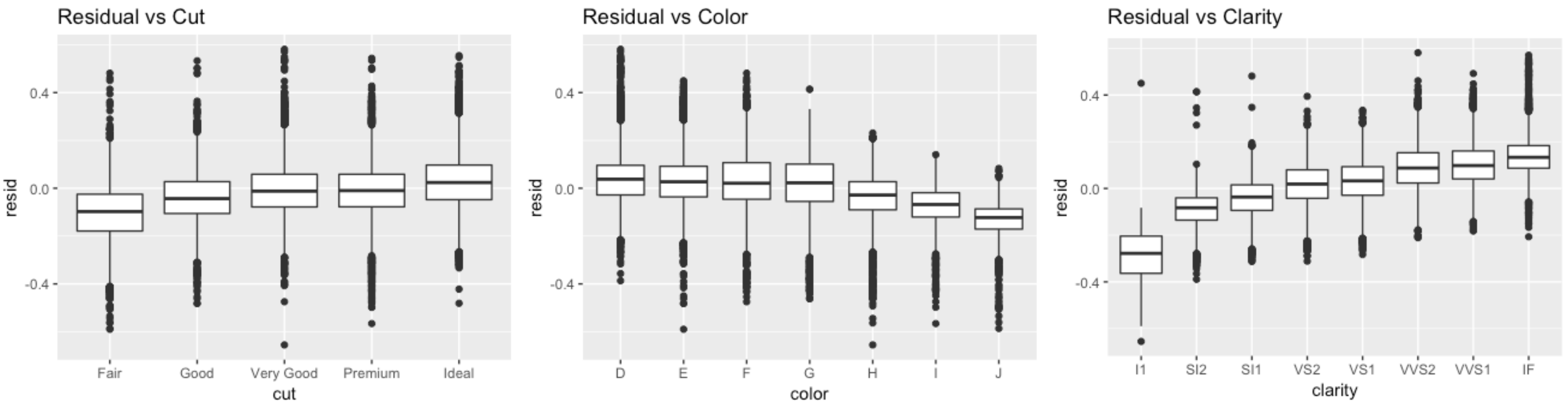


YOUR TURN - PART 2!

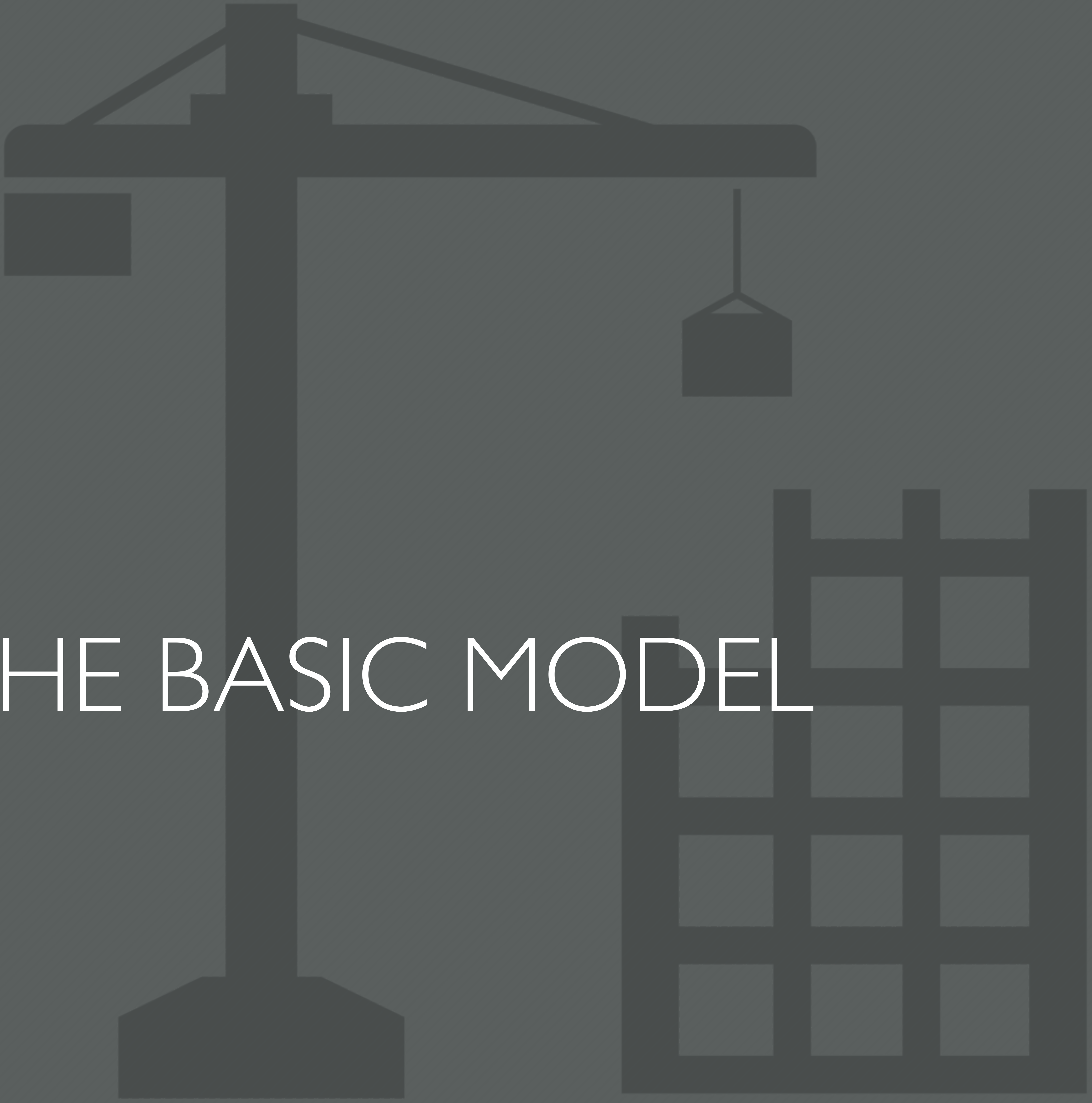
1. *Fit a linear model between the price and carat variables*
2. *Assess model numerically*
3. *Get prediction and residual data and add it to the diamonds data set*
4. *Visually assess model predictions*
5. *Visually assess model residuals*
6. *Visually assess relationship between residuals and cut, color, clarity. What does this tell you?*

SOLUTION

```r
# step 6: reassess relationship between residuals and other characteristics
p1 <- ggplot(diamonds2, aes(cut, resid)) + geom_boxplot() + ggtitle("Residual vs Cut")
p2 <- ggplot(diamonds2, aes(color, resid)) + geom_boxplot() + ggtitle("Residual vs Color")
p3 <- ggplot(diamonds2, aes(clarity, resid)) + geom_boxplot() + ggtitle("Residual vs Clarity")
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```



BUILDING ONTO THE BASIC MODEL



A MORE COMPLEX MODEL

Results from our `price ~ carat` residual assessment suggest that cut, color, and clarity may have an influence in price

Create a model that extends our previous model by incorporating cut, color, and clarity (without interaction)

A MORE COMPLEX MODEL

```
diamonds3 <- diamonds %>%
  select(price, carat, color, cut, clarity)

mod_diamond <- lm(log10(price) ~ log10(carat) +
color + cut + clarity, data = diamonds3)
```

Results from our `price ~ carat` residual assessment suggest that cut, color, and clarity may have an influence in price

How does this model appear to fit numerically?

A MORE COMPLEX MODEL

```
summary(mod_diamond)

Call:
lm(formula = log10(price) ~ log10(carat) + color + cut +
clarity,
    data = diamonds3)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43910 -0.03751 -0.00010  0.03622  0.84591

Coefficients:
               Estimate Std. Error  t value Pr(>|t|)
(Intercept)   3.6728414  0.0005071 7242.225  < 2e-16 ***
log10(carat)  1.8837175  0.0011288 1668.750  < 2e-16 ***
color.L      -0.1909054  0.0008804 -216.828  < 2e-16 ***
color.Q      -0.0415287  0.0008090  -51.335  < 2e-16 ***
```

Results from our **`price ~ carat`** residual assessment suggest that cut, color, and clarity may have an influence in price

How does this model appear to fit numerically?

VISUALLY ASSESSING A COMPLEX MODEL

Assessing predictions in a more complex model like this is hard to do visually…

VISUALLY ASSESSING A COMPLEX MODEL

```
diamonds3 %>%
  data_grid(cut, .model = mod_diamond)
# A tibble: 5 × 4
        cut carat color clarity
      <ord> <dbl> <chr>   <chr>
1      Fair   0.7     G     SI1
2      Good   0.7     G     SI1
3 Very Good   0.7     G     SI1
4   Premium   0.7     G     SI1
5     Ideal   0.7     G     SI1
```

…but using **`data_grid`** with **`.model`** helps

- This creates a table with each unique value of `cut` and…
- adds the most typical value for the other variables in the model

VISUALLY ASSESSING A COMPLEX MODEL

```
diamonds3 %>%
  data_grid(cut, .model = mod_diamond) %>%
  add_predictions(mod_diamond)
# A tibble: 5 × 5
        cut carat color clarity     pred
      <ord> <dbl> <chr>   <chr>    <dbl>
1      Fair   0.7     G     SI1 3.308263
2      Good   0.7     G     SI1 3.343028
3 Very Good   0.7     G     SI1 3.359169
4   Premium   0.7     G     SI1 3.368780
5     Ideal   0.7     G     SI1 3.378279
```

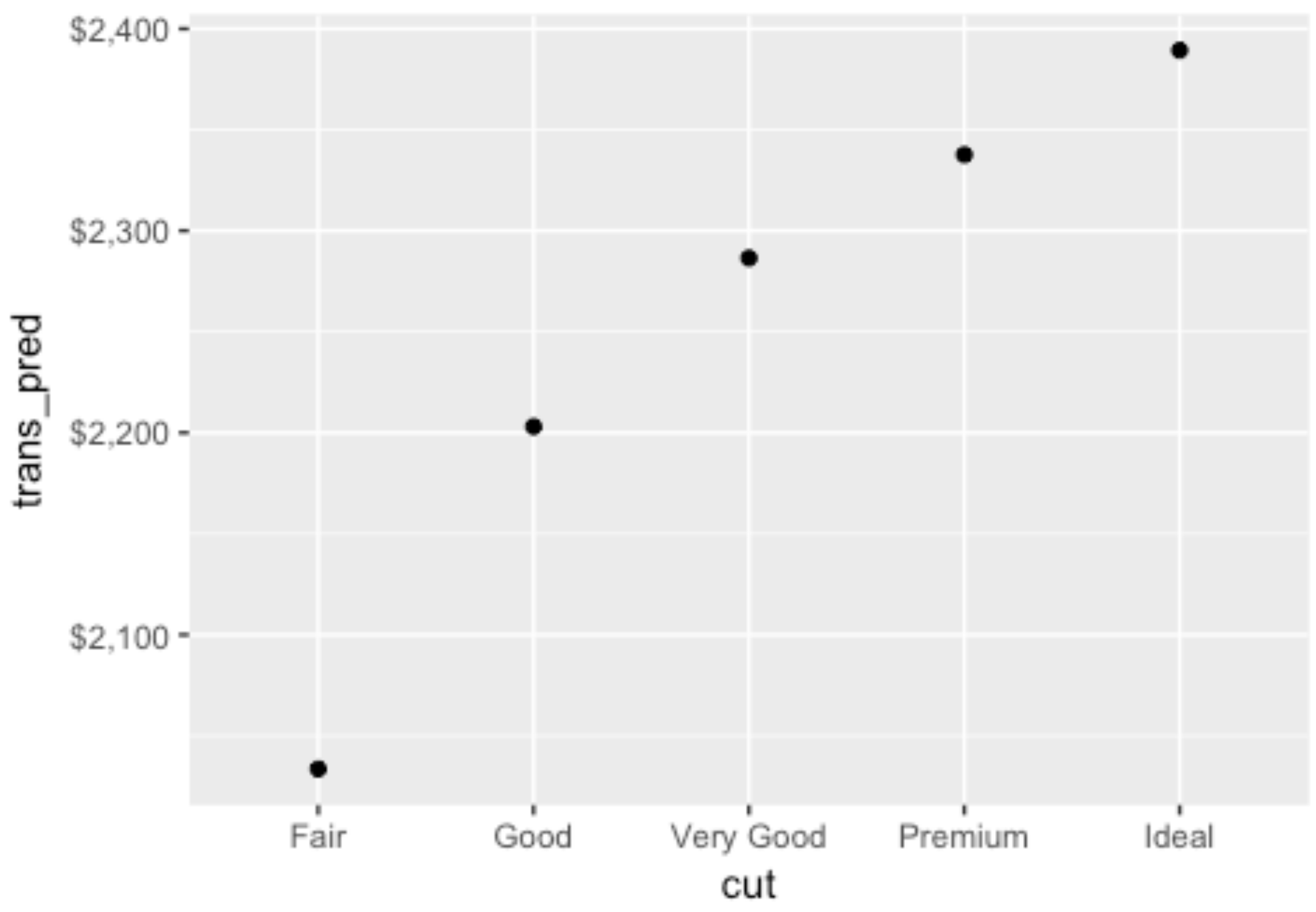
we can then add the most likely predicted values for each level of cut holding all else constant

VISUALLY ASSESSING A COMPLEX MODEL

```
diamonds3 %>%
  data_grid(cut, .model = mod_diamond) %>%
  add_predictions(mod_diamond) %>%
  mutate(trans_pred = 10 ^ pred) %>%
  ggplot(aes(cut, trans_pred)) +
  geom_point() +
  scale_y_continuous(labels = scales::dollar)
```

we can then transform our predicted values back to dollars…

and plot the most likely price for each level of cut



VISUALLY ASSESSING A COMPLEX MODEL

```
diamonds3 %>%
  data_grid(cut, .model = mod_diamond) %>%
  add_predictions(mod_diamond) %>%
  mutate(trans_pred = 10 ^ pred) %>%
  ggplot(aes(cut, trans_pred)) +
  geom_point() +
  scale_y_continuous(labels = scales::dollar)
```

changing `cut` to `color` or `clarity` will allow you to see similar plots for those variables.

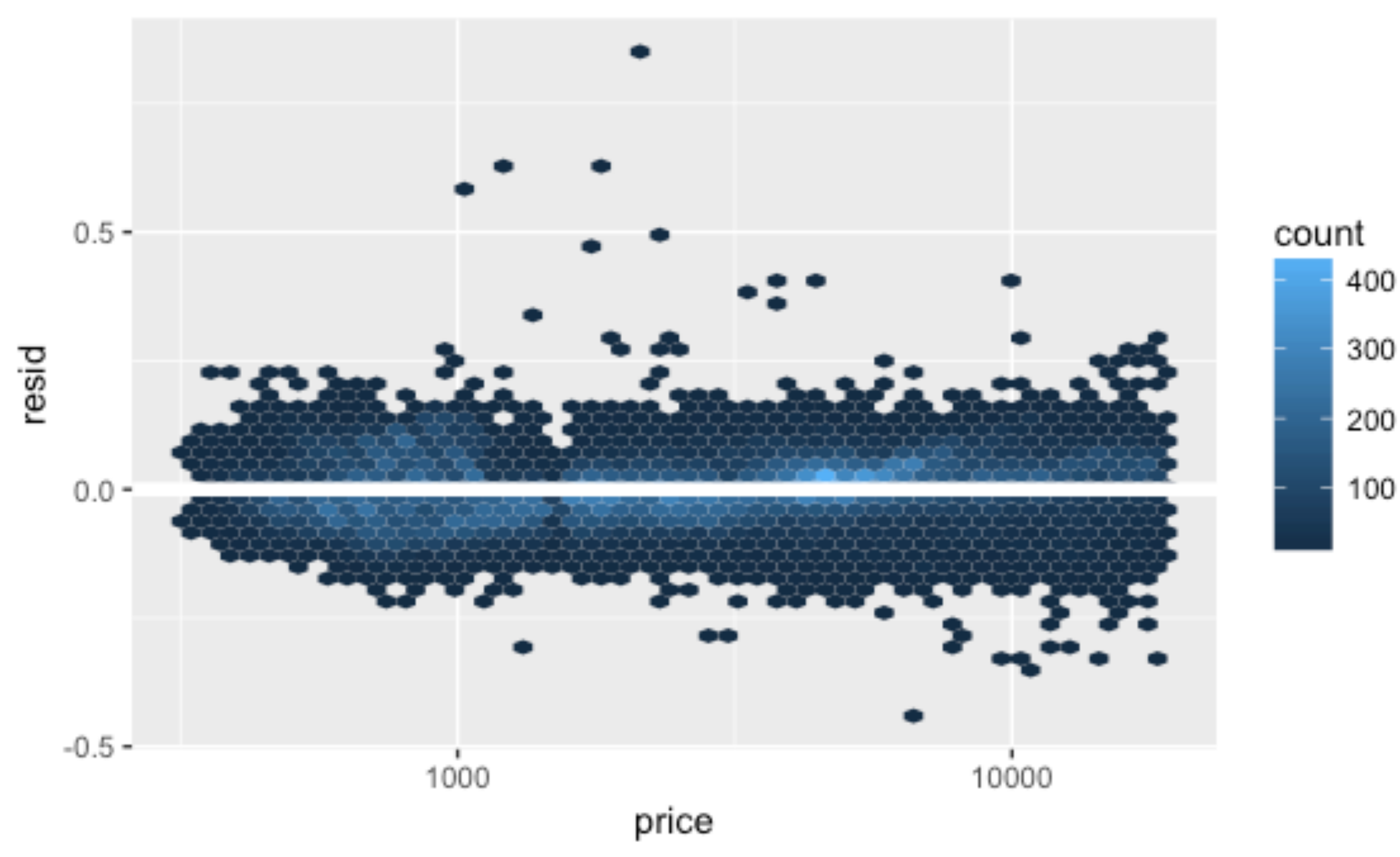
Opportunity to create a function!

YOUR TURN!

Lastly, how do the residuals look for this `mod_diamond` *model?*

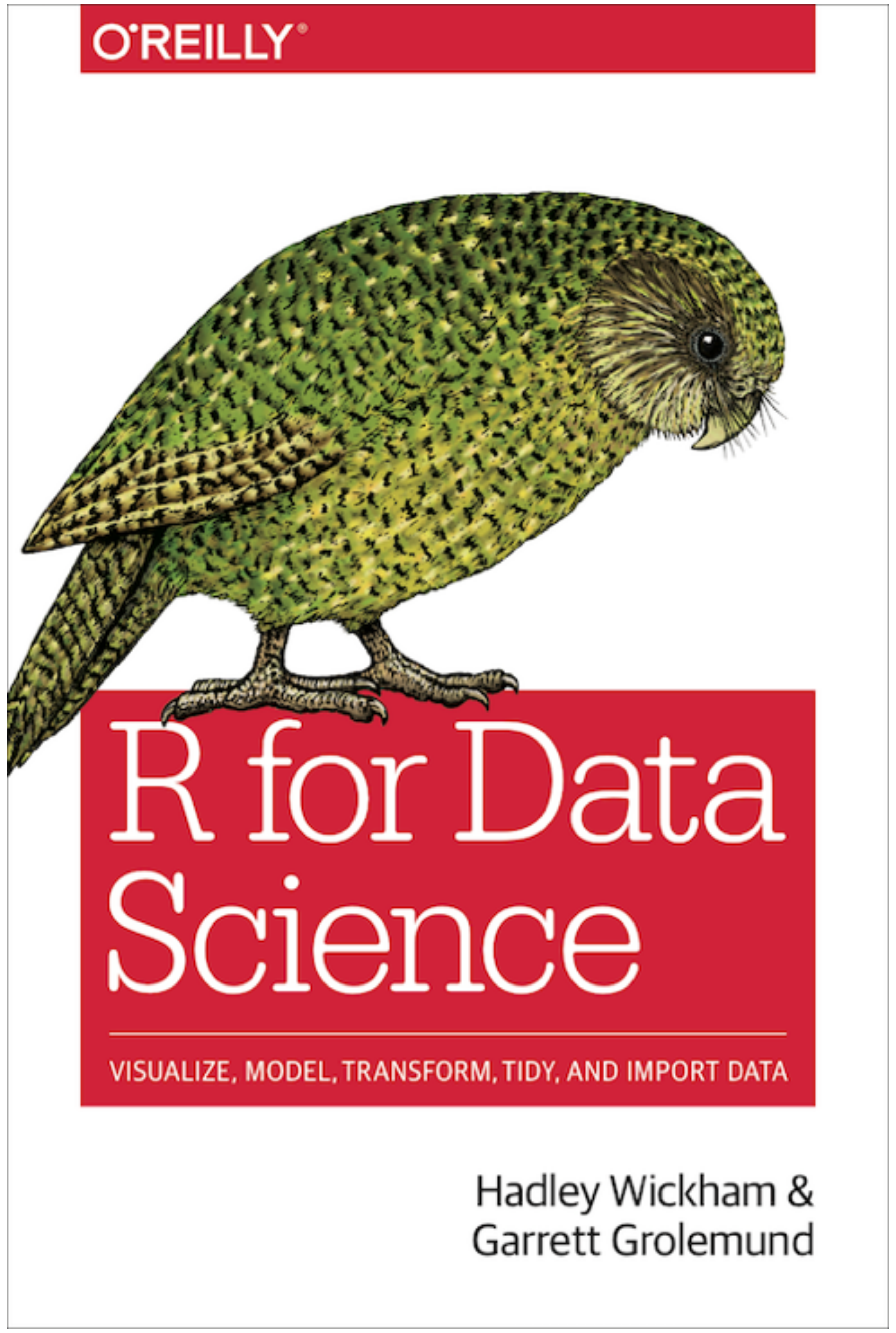
SOLUTION

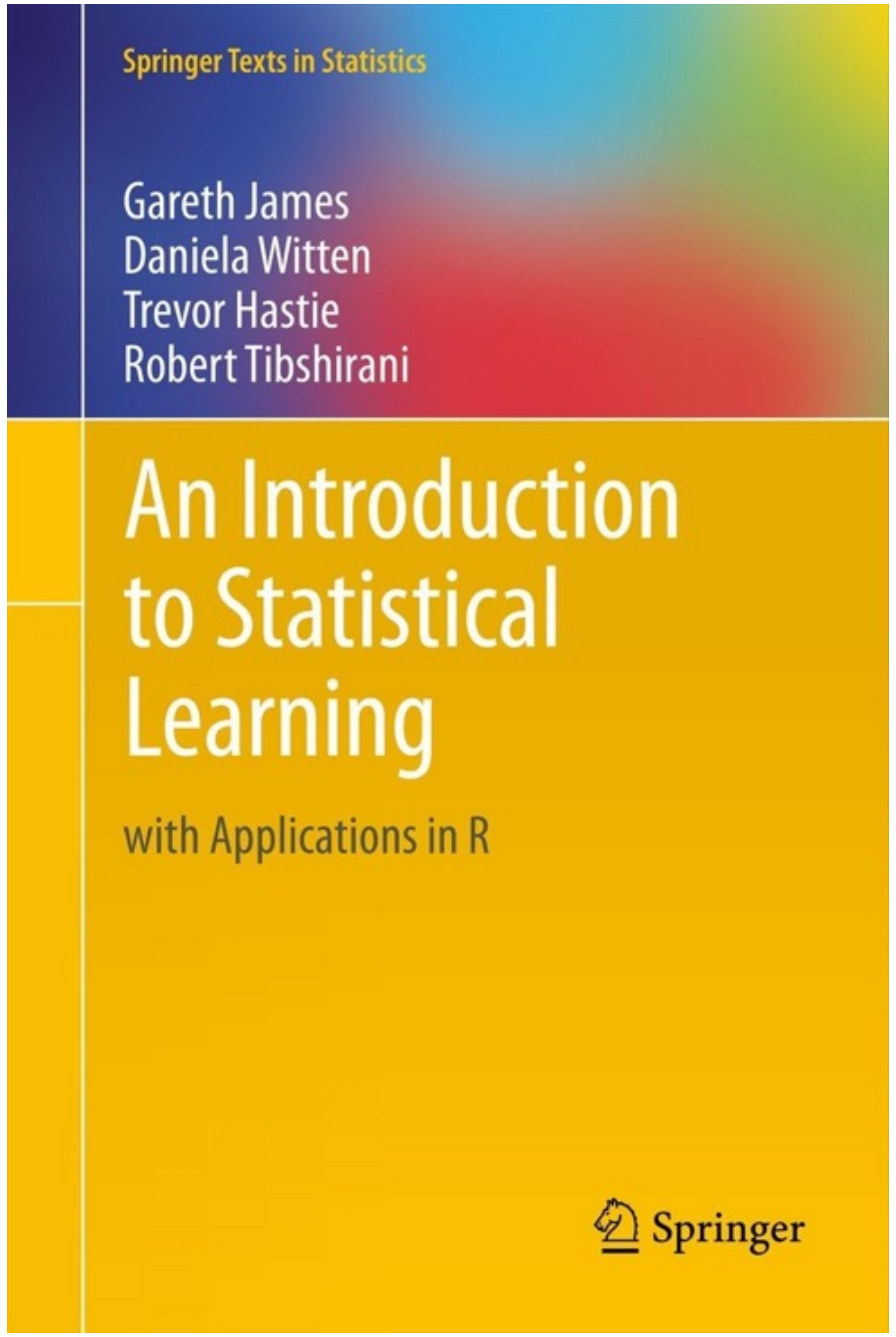
```r
# assess residuals
diamonds3 %>%
  add_residuals(mod_diamond) %>%
  ggplot(aes(price, resid)) +
  geom_hex(bins = 50) +
  geom_ref_line(h = 0) +
  scale_x_log10()
```

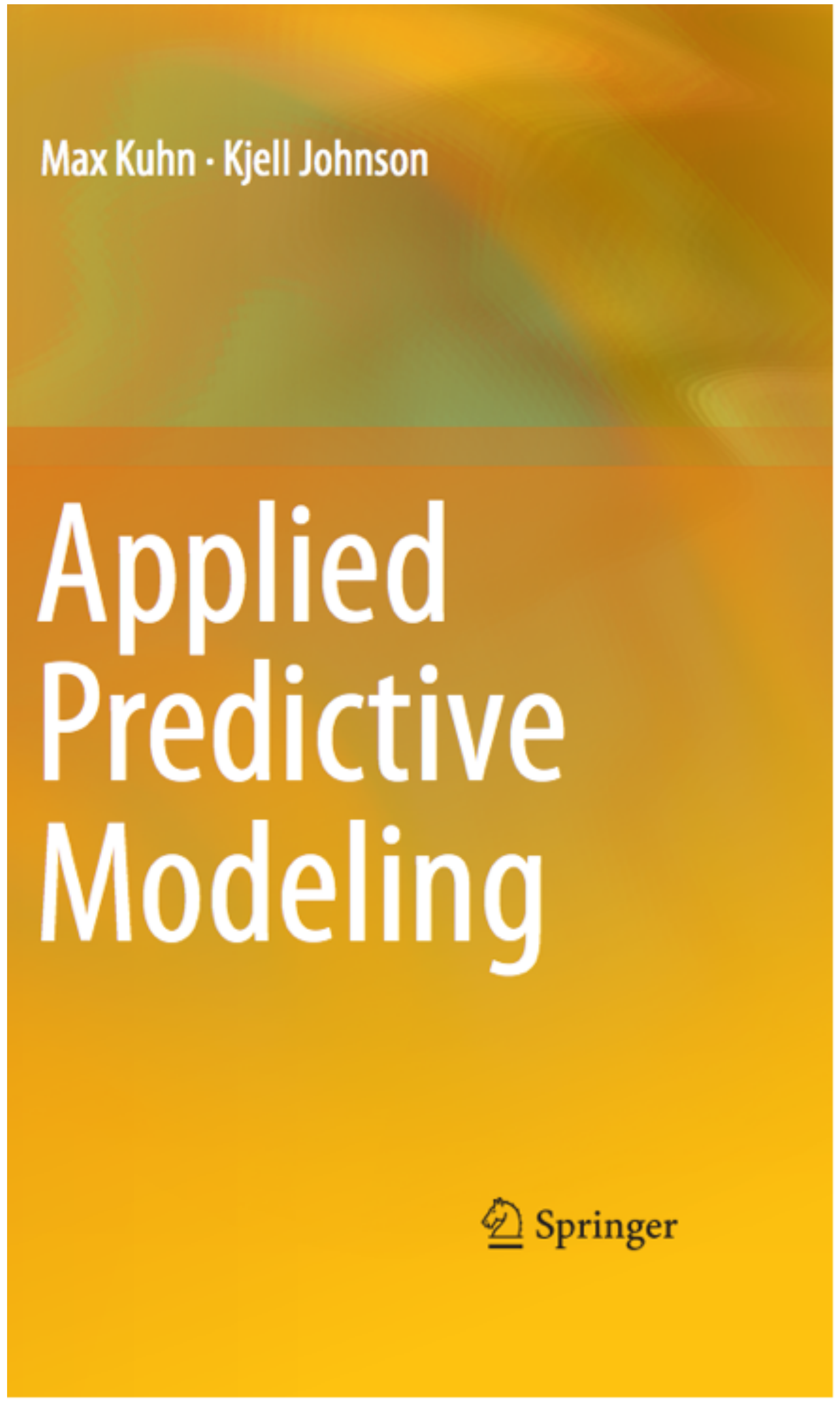


SO LITTLE TIME!

LEARN MORE


O'REILLY
R for Data Science
VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA
Hadley Wickham &
Garrett Grolemund


Springer Texts in Statistics
Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani
An Introduction to Statistical Learning
with Applications in R
Springer


Max Kuhn · Kjell Johnson
Applied Predictive Modeling
Springer

WHAT TO REMEMBER

FUNCTIONS TO REMEMBER

Operator/Function	Description
`cor, cor.test`	Compute correlation
`pairs, geom_ref_line`	Plot pairwise x-y scatterplots, add reference line to ggplot (great for assessing residual)
`lm(y ~ x, data = df)`	Linear model specification
`summary, residuals, fitted.values, coef`	Summarize and extract components out of the `lm()` object
`add_predictions, add_residuals, gather_predictions, gather_residuals`	Shortcut functions to add predicted values and residuals from an `lm()` object to a new or existing data frame
`model_matrix`	assess model specification