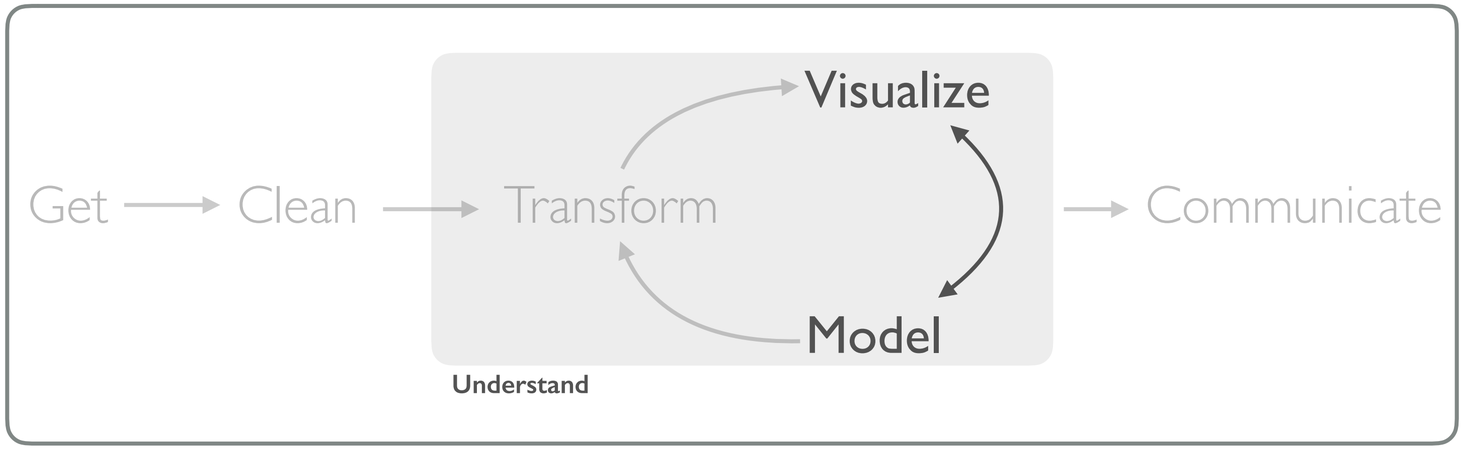
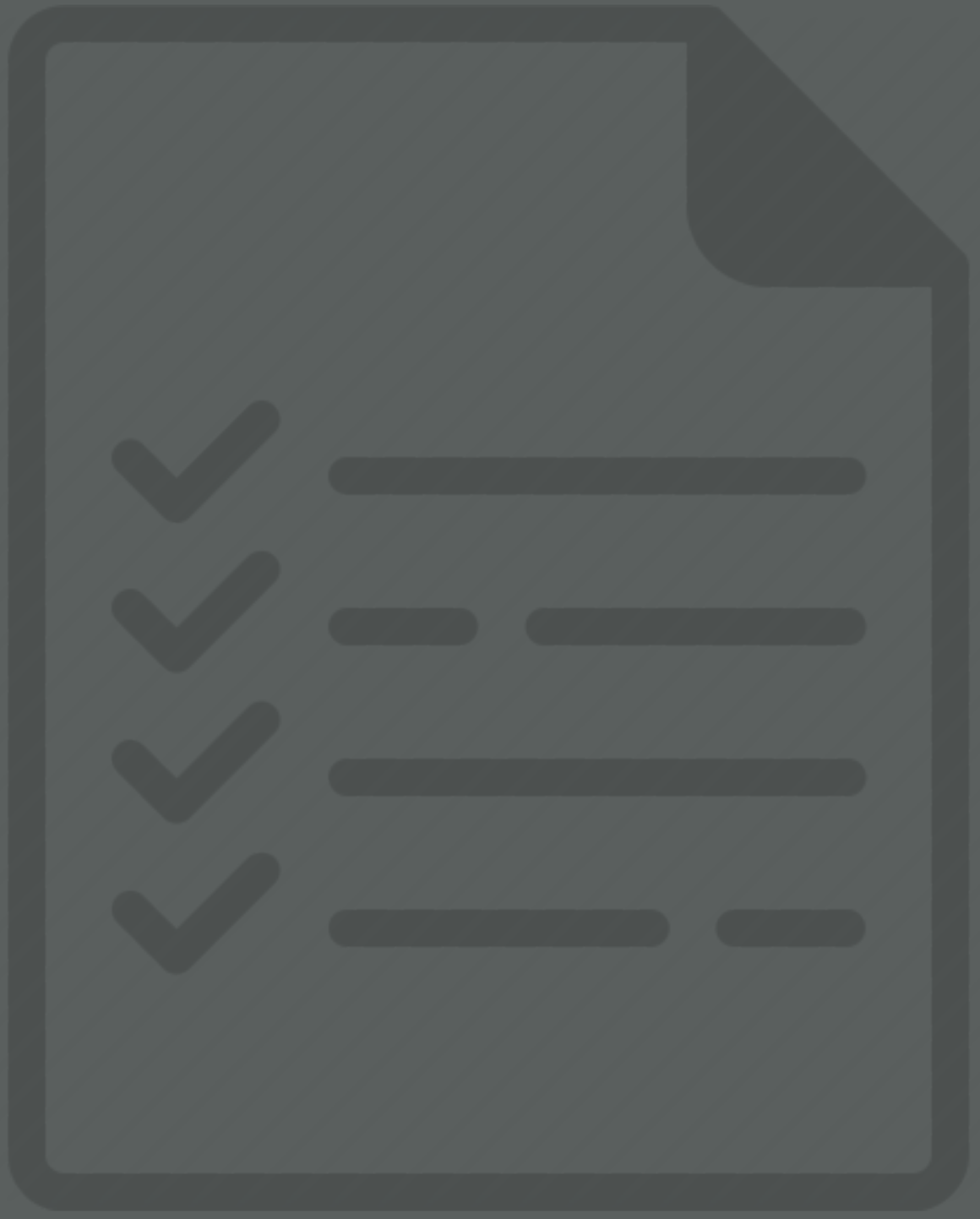
MODEL BUILDING



†A modified version of Hadley Wickham's analytic process

PREREQUISITES



PREREQUISITES

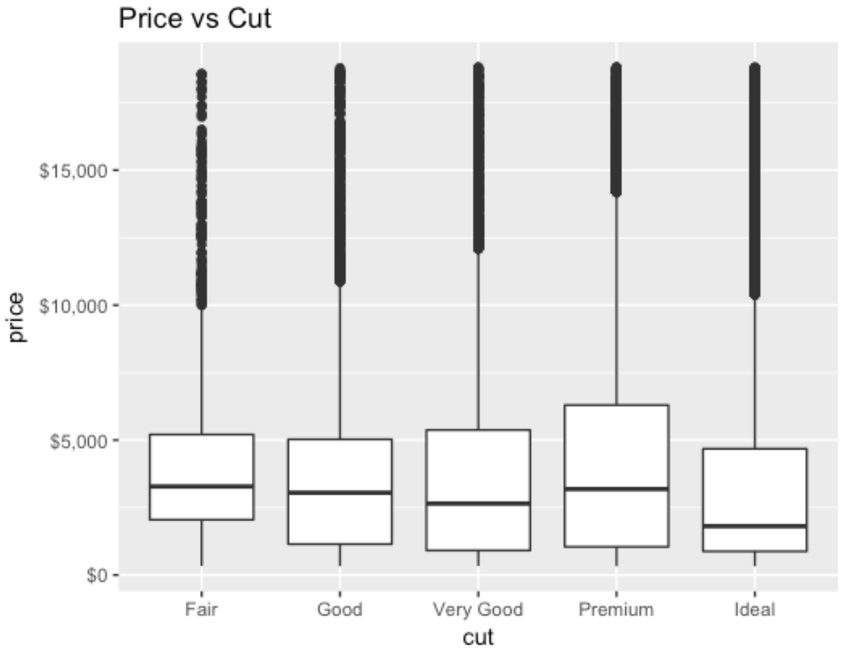
```
library(tidyverse)
library(modelr)
options(na.action = na.warn)
```

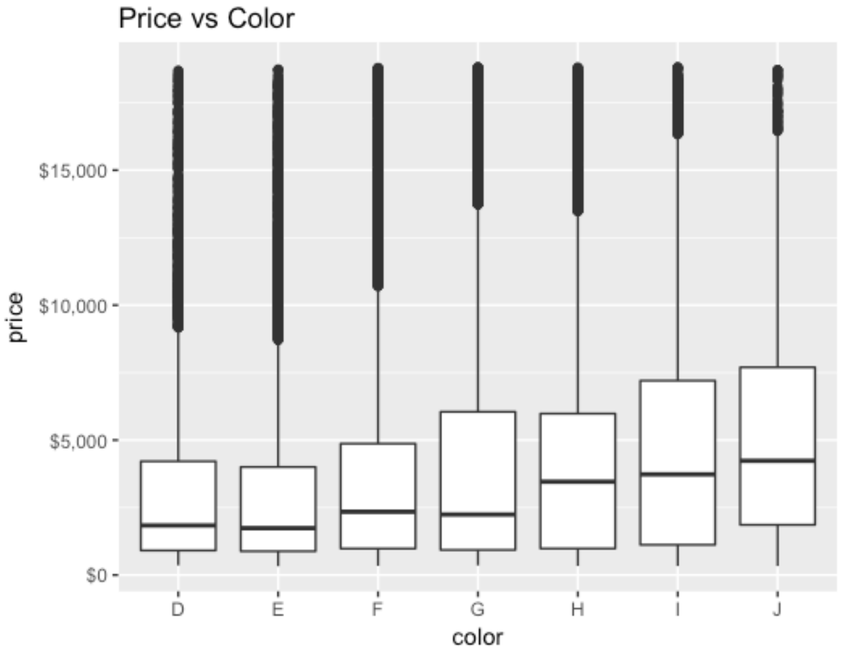
THE SET-UP

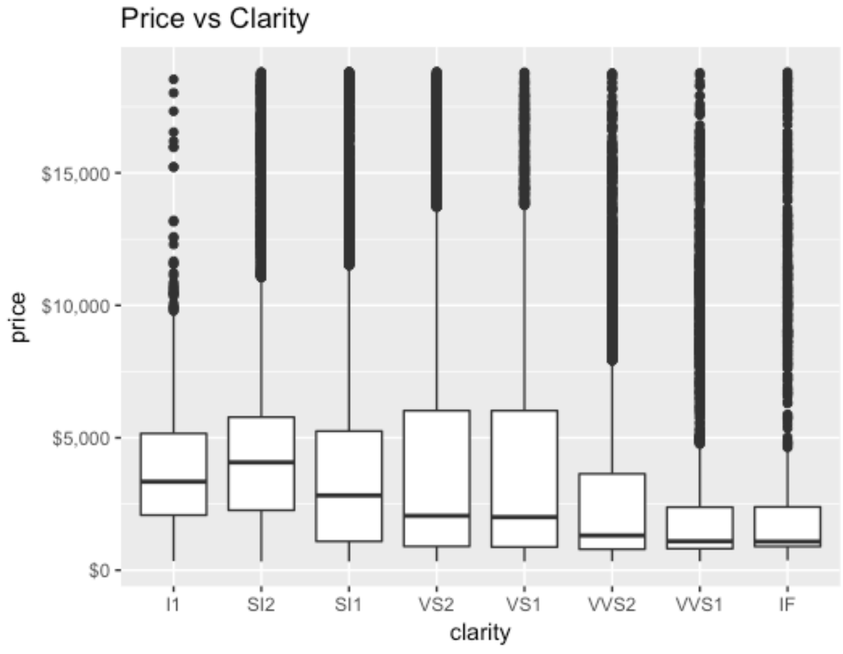

DIAMOND JEWELRY III
PAWN SHOP
718-220-5355
WE BUY GOLD & ELECTRONICS
42W.
WE PAWN
WE BUY CELL PHONES
OPEN

WHY ARE INFERIOR DIAMONDS MORE EXPENSIVE?

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

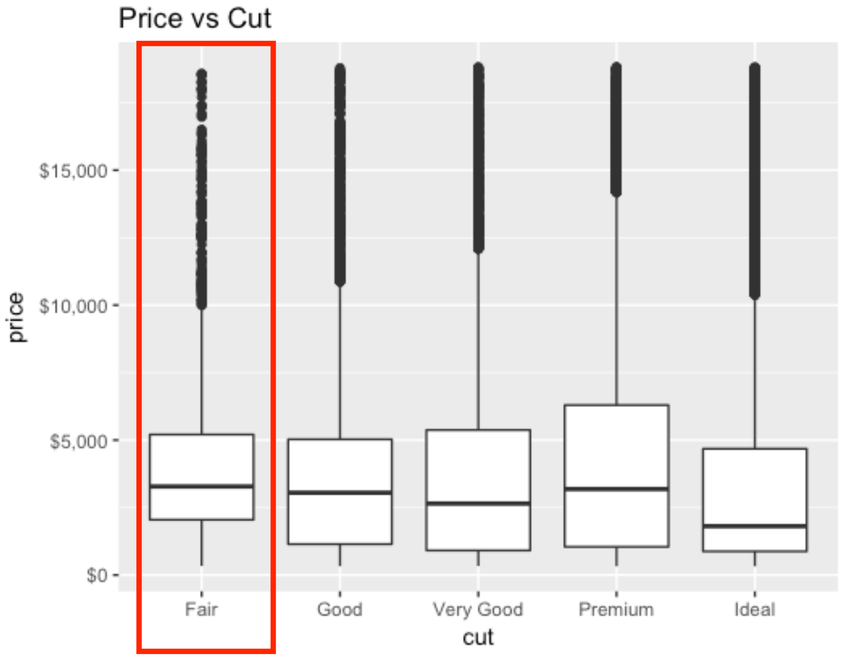


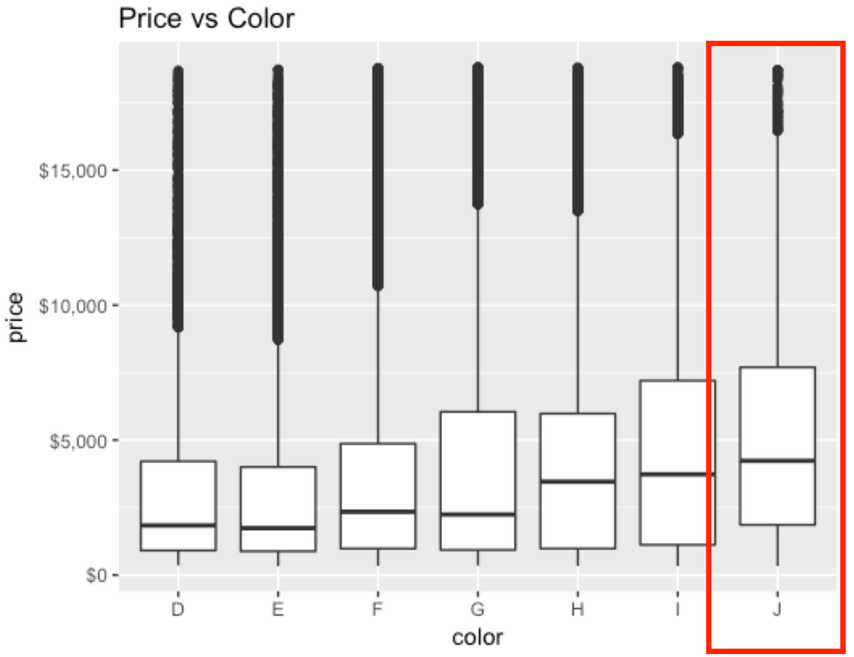


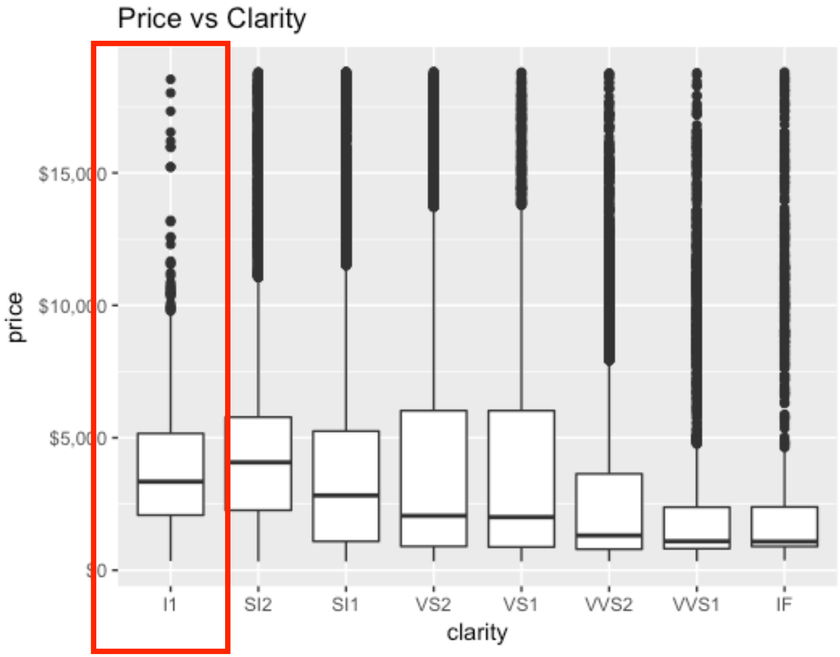


WHY ARE INFERIOR DIAMONDS MORE EXPENSIVE?

- 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







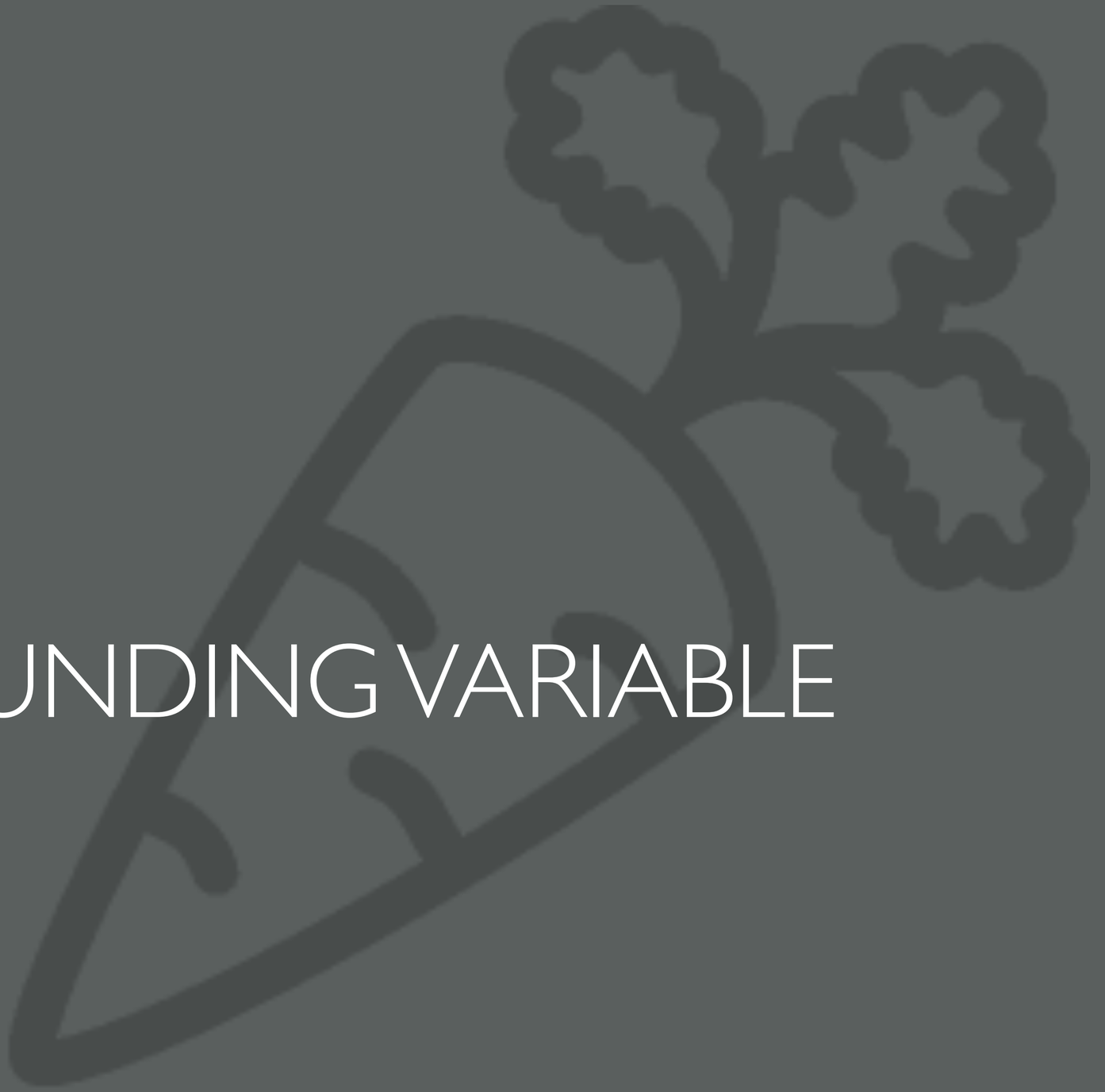
YOUR TURN!

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

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

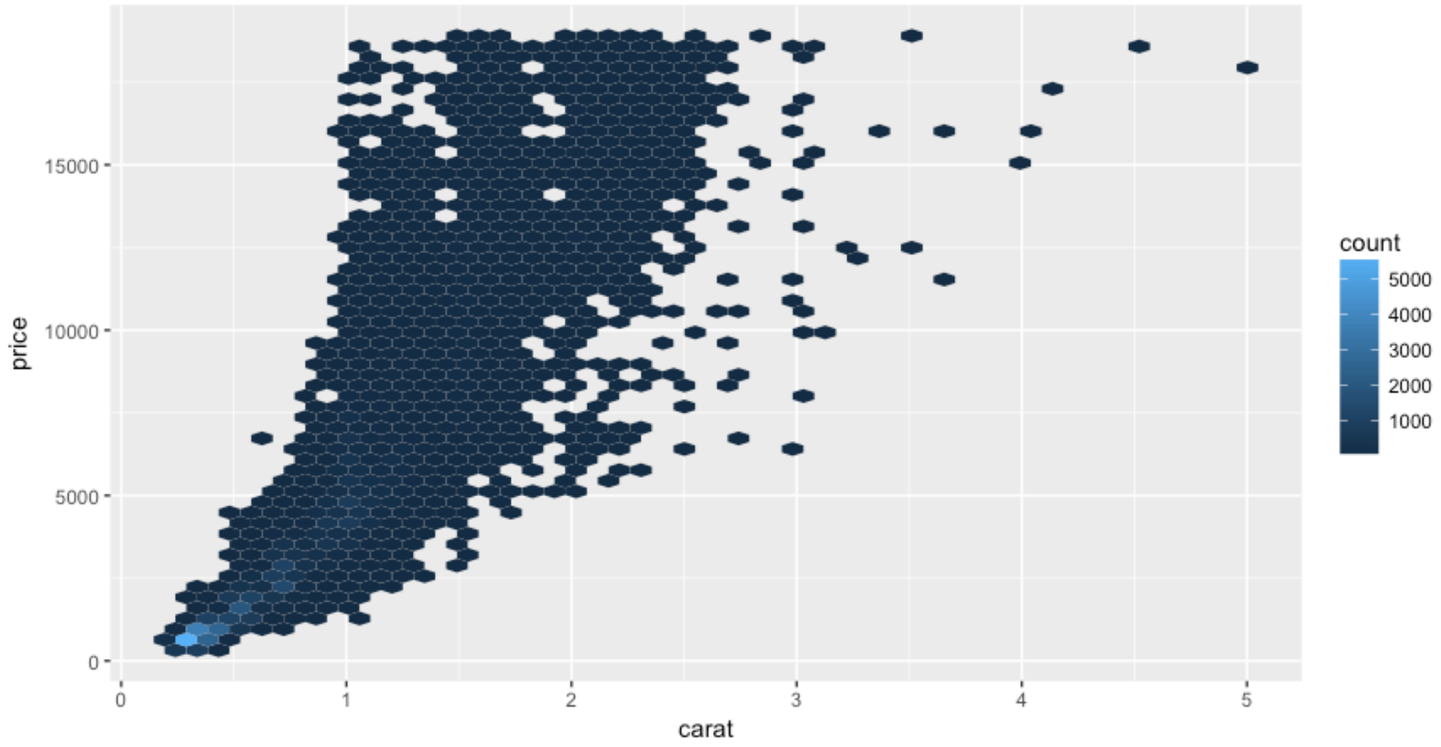
THOUGHTS?

A MAJOR CONFOUNDING VARIABLE



CONFOUNDING VARIABLE

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

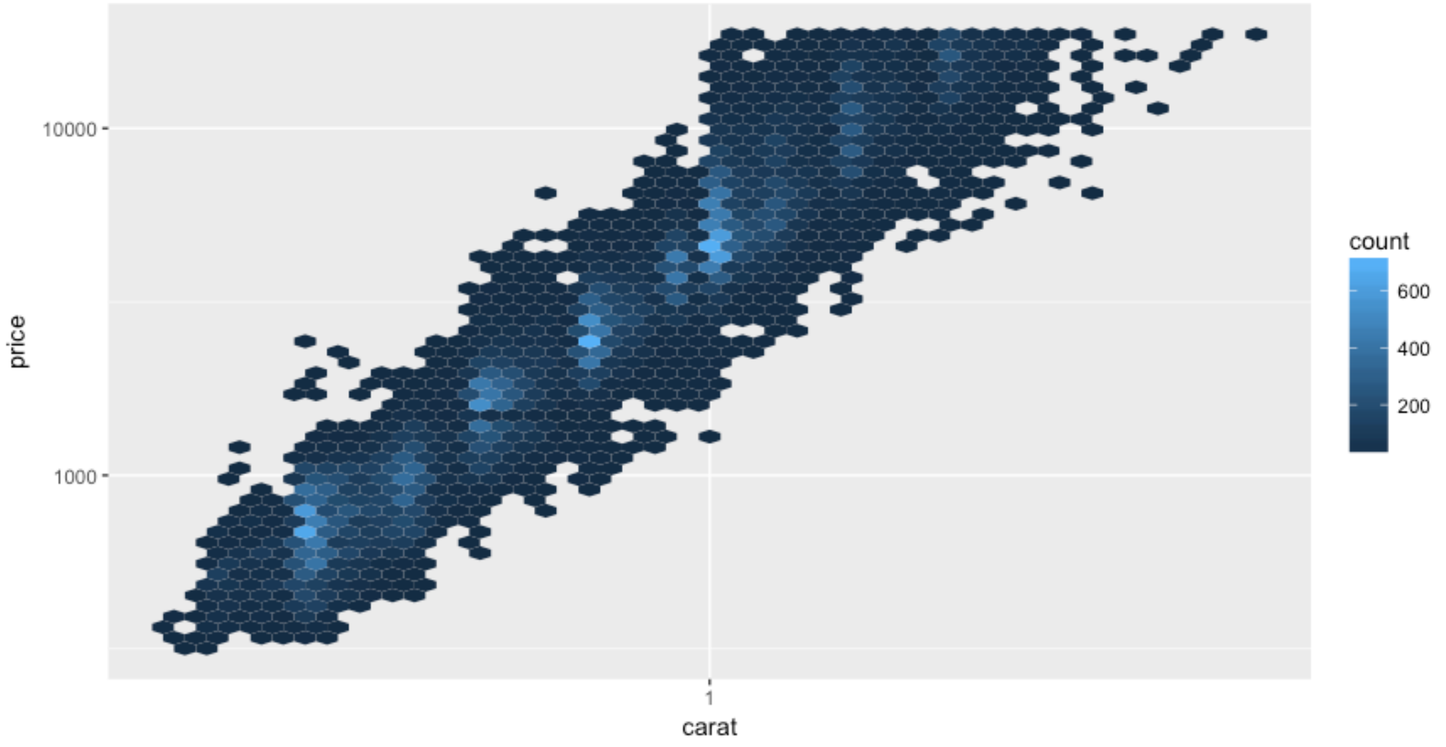


The **carat** variable has a big impact on price but is not captured in the previous 2-dimension plots

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!

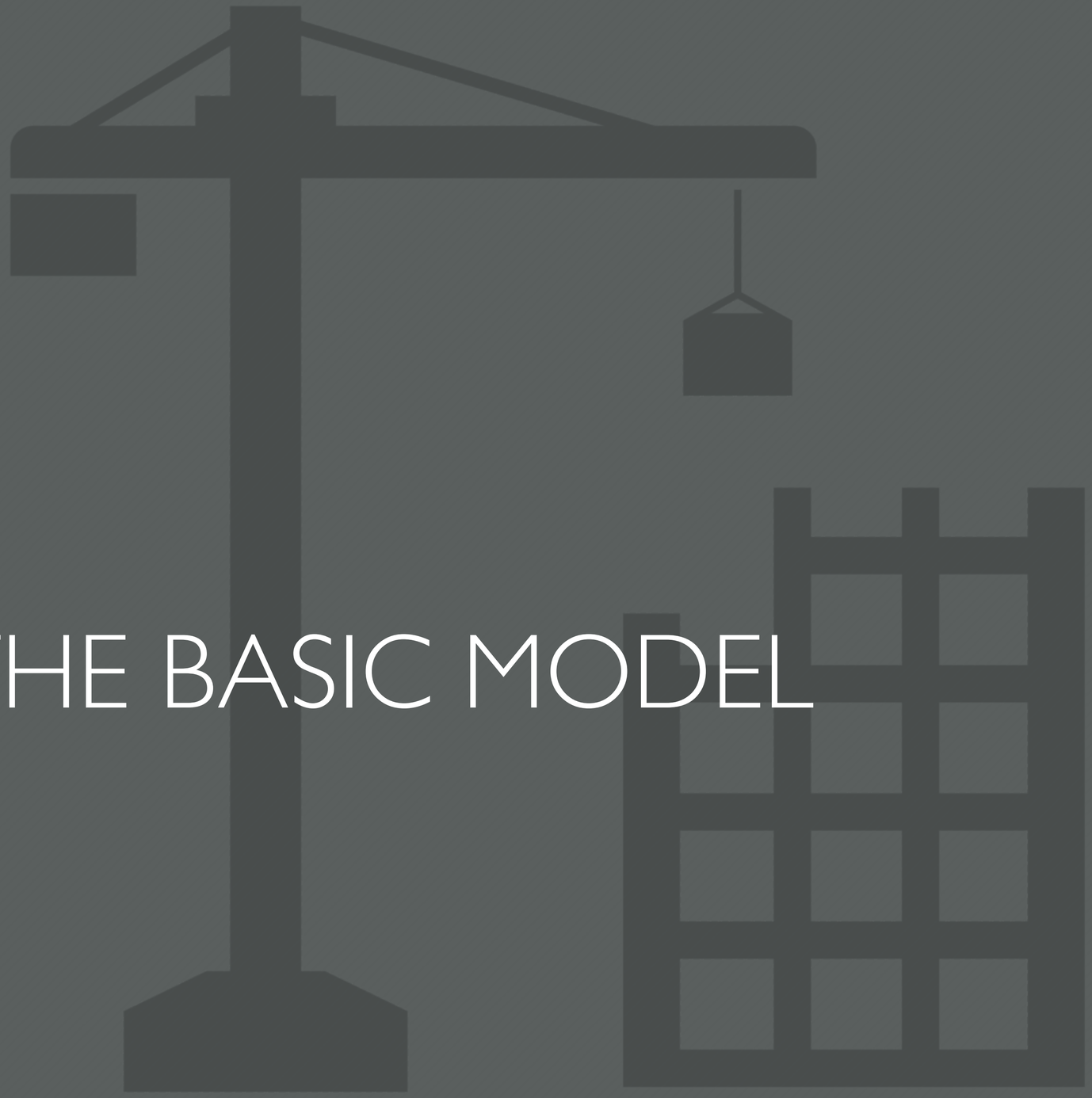
1. *Can you measure the strength of this linear relationship?*

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

YOUR TURN - PART 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?*

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
```

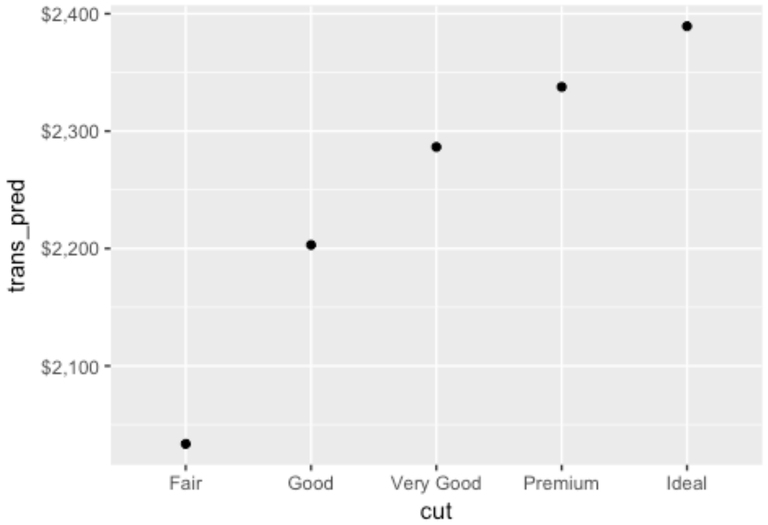
we can then add the most likely predicted values for each level of cut holding all else constant

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.

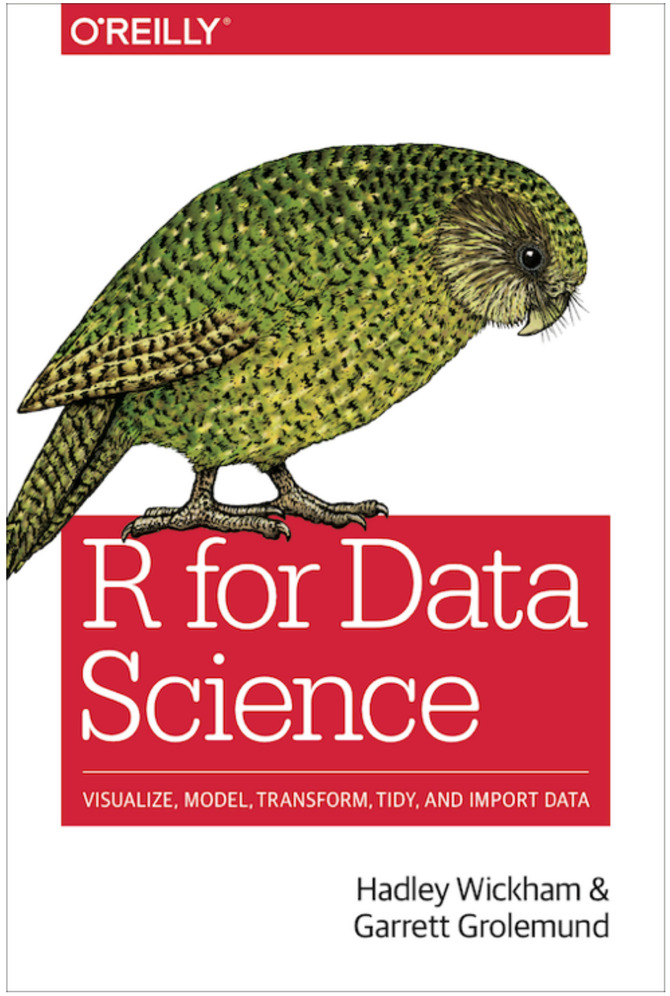
Opportunity to create a function!

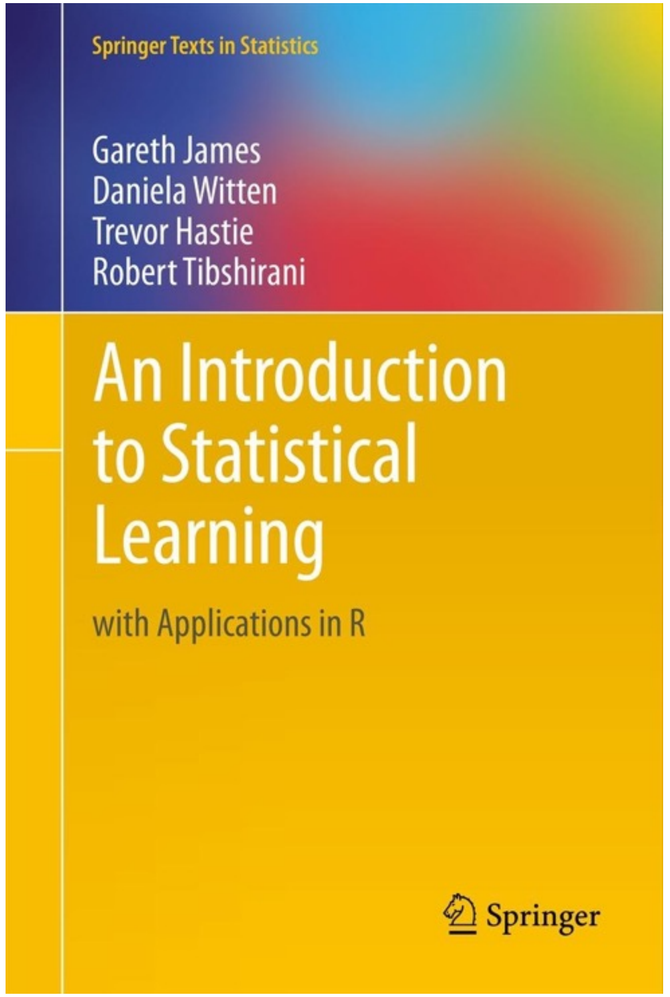
YOUR TURN!

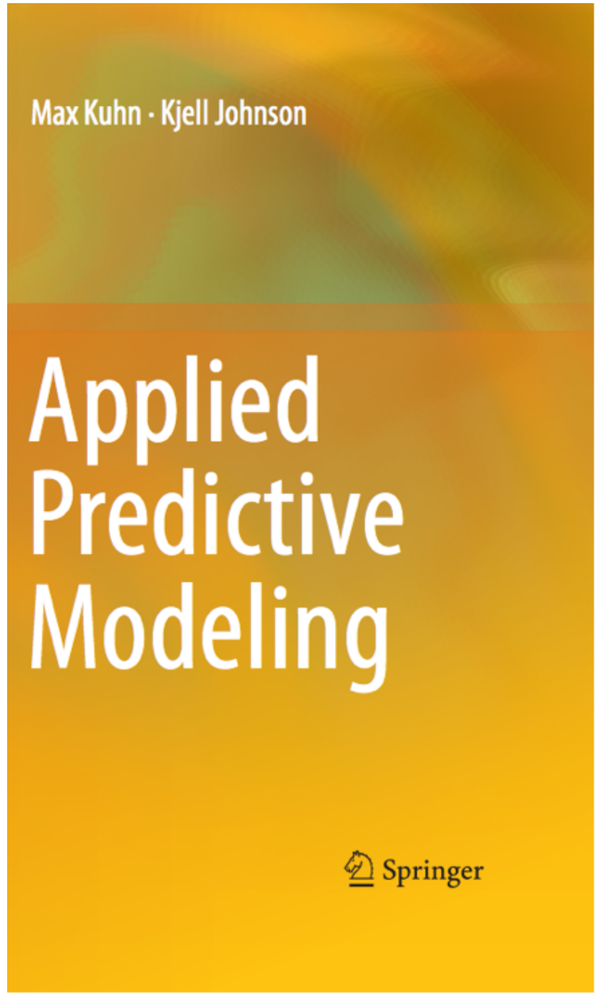
Lastly, how do the residuals look for this **`mod_diamond`** *model?*

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