Tell your story with data

*the whole pipeline on one **case study***

Seattle house prices

`log10()`

read_csv()
mutate()
ggplot()
lm()
get_regression_table()

Price, size, and condition for 21,000 houses

The pipeline

Nothing in this chapter is new. It is every earlier chapter, in order, on one dataset.

Import, Wrangle, Visualize, Model

- **Import** the data and look at what you actually have.
- **Wrangle** it into the shape the question needs.
- **Visualize** before you model, always.
- **Model** last, then say what it means in plain words.

Look first

Three things worth catching before you fit anything.

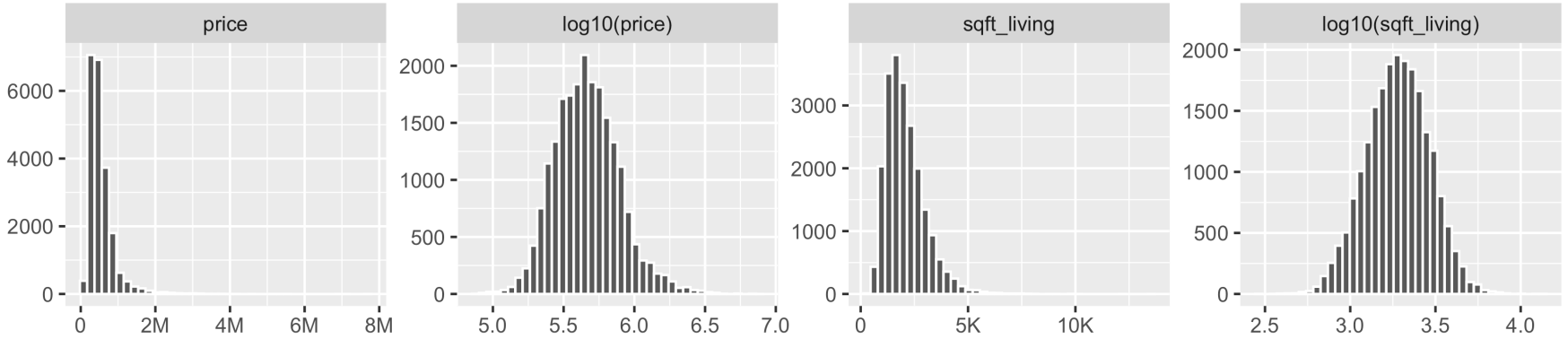
Data Quality

- **Skew**: a long right tail makes a mean misleading.
- **Missing values**: know how many, and why they are missing.
- **Outliers**: decide whether they are errors or real cases.
- A histogram of every numerical variable takes a minute and saves an hour.

log10()

A **log transform** pulls in a long right tail, turning a lopsided cloud into one a straight line can describe.

Tame The Skew



Price and size are both heavily right-skewed. Their logarithms are not.

mutate()

Create the transformed variables once, then model with them.

Example

```
house_prices <- house_prices |>
  mutate(log10_price = log10(price), log10_size = log10(sqft_living))
```


the verb (function)

what to call the new column

what to put in it

ggplot()

Plot the relationship, split by the categorical variable, before deciding what model to fit.

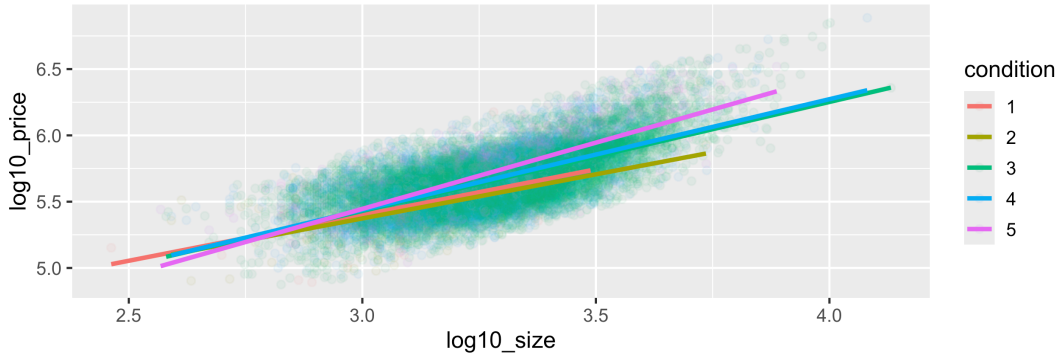
one color per condition

```
ggplot(house_prices, aes(x = log10_size, y = log10_price, color = condition)) +
  geom_point(alpha = 0.05) +
  geom_smooth(method = "lm", se = FALSE)
```

← **make 21,000 points faint**

one straight line per condition

Example



interaction

If the clouds for different conditions trend at different angles, one shared slope will not do.

One Slope Per Group

An **interaction** gives each condition its own intercept and its own slope.

lm()

Use a **star**, not a plus: `*` fits an interaction, so each condition gets its own slope.

Example

the verb (function)

interaction, not parallel slopes

```
price_interaction <- lm(log10_price ~ log10_size * condition,
                        data = house_prices)
```

the data frame

get_regression_table()

Read the coefficients, remembering that they are on the **log10** scale, not in dollars.

Example

```
get_regression_table(price_interaction)
```

the verb (function)

the fitted model

```
# A tibble: 10 × 7
   term                    estimate std_error statistic p_value lower_ci upper_ci
   <chr>                      <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
 1 intercept                   3.33     0.451      7.38   0         2.45     4.22
 2 log10_size                  0.69     0.148      4.65   0         0.399    0.98
 3 condition-2                 0.047    0.498      0.094  0.925    -0.93     1.02
 4 condition-3                -0.367    0.452     -0.812  0.417    -1.25     0.519
 5 condition-4                -0.398    0.453     -0.879  0.38     -1.29     0.49
 6 condition-5                -0.883    0.457     -1.93   0.053    -1.78     0.013
 7 log10_size:condition-2     -0.024    0.163     -0.148  0.882    -0.344    0.295
 8 log10_size:condition-3      0.133    0.148      0.893  0.372    -0.158    0.424
 9 log10_size:condition-4      0.146    0.149      0.979  0.328    -0.146    0.437
10 log10_size:condition-5      0.31     0.15       2.07   0.039     0.016    0.604
```

Telling it

The last step is the one that gets graded in the real world.

Say What It Means

- Translate coefficients back out of log scale before quoting them.
- Name the units. A slope without units is not a finding.
- Say what the model **cannot** tell you, including anything causal.
- One clear plot beats a table of numbers for a non-technical reader.