Inference for regression

is the slope real?

`beta-1`

*putting a confidence interval on a **slope***

```
get_regression_table()
specify()
calculate(stat = "slope")
get_ci()
```

Old Faithful: waiting time against eruption duration

The question

A fitted slope is computed from one sample, so it would come out differently on the next one.

One Sample, One Slope

- The slope you fit is a **statistic**, not the truth.
- The population slope is a **parameter** you never see.
- So the slope has a sampling distribution, like any other estimate.
- This chapter puts an interval and a p-value on it.

lm()

Fit the model first and give it a name. Every step after this one reads the fitted model rather than the data.

Example

the fitted model, saved for later

```
faithful_model <- lm(waiting ~ duration, data = old_faithful_2024)
```

response ~ explanatory

where those two variables live

get_regression_table()

Return the **regression table**: each coefficient with its standard error, test statistic, p-value, and confidence interval, all at once.

Example

```
get_regression_table(faithful_model)
```

the verb (function)

the fitted model

```
# A tibble: 2 × 7
  term      estimate std_error statistic p_value lower_ci upper_ci
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept   79.5       7.31       10.9       0   65.0     93.9
2 duration     0.371     0.032      11.4       0    0.307    0.435
```

Reading the row

Each coefficient gets a row, and each row answers a different question.

What The Columns Mean

- **estimate** is the fitted slope or intercept.
- **std_error** is how much it would bounce between samples.
- **p_value** tests whether that coefficient could be zero.
- **lower_ci** and **upper_ci** bracket the plausible values.

conditions

The regression table's standard errors are only trustworthy when the residuals behave. Check them before you quote a p-value.

LINE

Linearity, Independence, Normality of residuals, Equal variance.

Checking LINE

Two plots do most of the work.

Residual Plots

- **Linearity**: the scatterplot should not curve.
- **Independence**: think about how the data were collected.
- **Normality**: a histogram or Q-Q plot of the residuals.
- **Equal variance**: residuals against fitted values, no funnel shape.

get_regression_points()

The residuals you need for those checks, one row per observation.

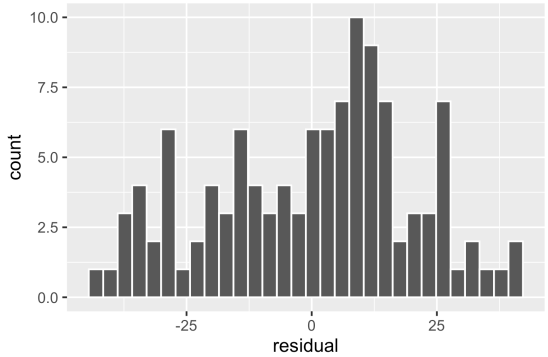
fitted values and residuals

```
faithful_points <- get_regression_points(faithful_model)
ggplot(faithful_points, aes(x = residual)) +
  geom_histogram(color = "white")
```

← **what to plot**

a histogram of the residuals

Example



calculate()

The **infer** workflow again, with the slope as the statistic. Only the **stat** argument differs from chapter 8.

Example

```
bootstrap_distn_slope <- old_faithful_2024 |>
  specify(waiting ~ duration) |>
  generate(reps = 1000, type = "bootstrap") |>
  calculate(stat = "slope")
```

← **response ~ explanatory**

the slope, not a mean this time

observed slope

The same pipeline without **generate()** gives the slope actually fitted to the data. The interval is centered on it.

Example

```
observed_slope <- old_faithful_2024 |>
  specify(waiting ~ duration) |>
  calculate(stat = "slope")
```

no resampling: this is your data

get_ci()

A confidence interval for the population slope, from the bootstrap distribution you just built.

Example

```
bootstrap_distn_slope |>
  get_ci(level = 0.95, type = "se", point_estimate = observed_slope)
```

the verb (function) — **how confident** — **the standard error method** — **center it on the slope you observed**

Putting it together

Two routes to the same conclusion.

Theory & Simulation

- **get_regression_table()** gives the theory-based interval and p-value.
- The **infer** pipeline gives the simulation-based version.
- When the conditions hold, the two agree closely.
- If the interval excludes zero, the slope is distinguishable from zero.