Hypothesis testing

the null distribution

`p-value`

*the same **infer** verbs, with one more step*

specify()
hypothesize()
generate()
calculate()
get_p_value()

Are metal tracks more often popular than deep house tracks?

The logic

Assume nothing is going on, then ask how surprising your data would be.

Assume Nothing, Then Check

- The **null hypothesis** says there is no difference.
- Shuffle the data so any real difference is destroyed.
- Do that a thousand times to build a **null distribution**.
- See how far out your observed statistic falls.

hypothesize()

The one verb hypothesis testing adds. **null = “independence”** says the two variables are unrelated; **null = “point”** tests a single value.

Assume No Difference

Like specify(), it changes no data. It only records the assumption.

specify()

For a two-variable test, give a formula and say which outcome counts as a **success**.

Example

the data frame

```
spotify_metal_deephouse |>
  specify(formula = popular_or_not ~ track_genre, success = "popular")
```

the verb (function)

response ~ explanatory

which level counts as a success

generate()

type = "permute" shuffles the explanatory variable, breaking any link between the two. That is what the null hypothesis claims is true.

Example

```
spotify_metal_deephouse |>
  specify(formula = popular_or_not ~ track_genre, success = "popular") |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute")
```

← **assume no relationship**

the verb (function)

how many shuffles

shuffle, do not resample

calculate()

Reduce each shuffle to one number. For two groups, **order** says which one is subtracted from which.

One Number Per Shuffle

Get the order backwards and the sign of every difference flips.

calculate()

the null distribution, saved for later

```
null_distribution <- spotify_metal_deephouse |>
  specify(formula = popular_or_not ~ track_genre, success = "popular") |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in props", order = c("metal", "deep-house"))
```

the verb (function)

which statistic

metal minus deep house

observed statistic

The same pipeline without **hypothesize()** and **generate()** gives the difference actually seen in the data.

Example

```
obs_diff_prop <- spotify_metal_deephouse |>
  specify(formula = popular_or_not ~ track_genre, success = "popular") |>
  calculate(stat = "diff in props", order = c("metal", "deep-house"))
```

no shuffling: this is your data

get_p_value()

The fraction of the null distribution at least as extreme as what you observed.

```
null_distribution |>
  get_p_value(obs_stat = obs_diff_prop, direction = "right")
```

the verb (function)

what you actually saw

which tail to count

Example

```
# A tibble: 1 × 1
  p_value
    <dbl>
1   0.065
```

shade_p_value()

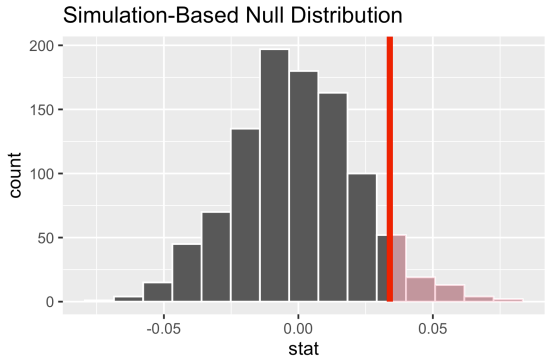
Show the same number as an area under the null distribution.

plot the null distribution

```
visualize(null_distribution) +
  shade_p_value(obs_stat = obs_diff_prop, direction = "right")
```

shade everything at least this extreme

Example



What a p-value is

And the four things it is not.

Reading The p-value

- It is the chance of data this extreme **if the null were true**.
- It is **not** the probability the null is true.
- A small p-value is evidence **against** the null, not proof of the alternative.
- A large p-value is not evidence that nothing is going on.
- Significance is not importance: report the **size** of the effect too.