The interval, and the sentence around it

Why the textbook interval gives 0.88% for a proportion that cannot go below zero, the logit interval that fixes it, degrees of freedom you have 72 of, and why two overlapping intervals do not mean no difference.

Pierre Robentz Cassion

Lesson 7 of 8

Survey Analysis, Sampling and Weighting — What you may publish

What this lesson covers

- What the interval is for
- The textbook interval, and where it fails
- The logit interval
- Degrees of freedom you actually have
- Overlapping intervals are not a test
- Rounding, and false precision
- The sentence
- Reading an interval against a threshold
- What comes next

What the interval is for

- A survey estimate is a guess about a population, made from a sample.

The textbook interval, and where it fails — Example

```
p +/- z * se
```

The textbook interval, and where it fails — In Python

```
p, se = 0.0217, 0.0066
print(f"Wald: {p - 1.96 * se:.2%} to {p + 1.96 * se:.2%}")
```

The textbook interval, and where it fails — In R

```
p <- 0.0217; se <- 0.0066
c(p - 1.96 * se, p + 1.96 * se)
```

The logit interval — In Python

```
import math

logit = math.log(p / (1 - p))
se_logit = se / (p * (1 - p))
low, high = (logit - 1.96 * se_logit, logit + 1.96 * se_logit)

print(f"logit: {1 / (1 + math.exp(-low)):.2%} to {1 / (1 + math.exp(-high)):.2%}")
```

The logit interval — In R

```
library(survey)
svyciprop(~I(child_muac_mm < 115), child_design, method = "logit")
```

The logit interval

- 1.19% to 3.92% — Asymmetric — further above the estimate than below — which is the correct shape for a small. . .

Degrees of freedom you actually have — Example

```
df = number of PSUs - number of strata
```

Degrees of freedom you actually have — In Python

```
clusters = survey["ea_id"].nunique()
strata = survey["stratum"].nunique()
print(f"{clusters} clusters, {strata} strata, df = {clusters - strata}")
```

Degrees of freedom you actually have — In R

```
c(clusters = n_distinct(survey$ea_id), strata = n_distinct(survey$stratum))
degf(design)
```

Degrees of freedom you actually have

- Use the t multiplier, and report the degrees of freedom — It costs nothing and it tells a reader how many clusters you...

Overlapping intervals are not a test — In Python

```
for stratum in ["urban", "rural-accessible", "rural-remote"]:
    print(stratum)   # displaced households
```

Overlapping intervals are not a test

Stratum	Displaced	95% interval
Urban	15.6%	12.0 – 19.2%
Rural accessible	8.0%	4.7 – 11.3%
Rural remote	15.7%	11.0 – 20.4%

Overlapping intervals are not a test — In R

```
svyttest(I(displacement_status == "displaced") ~ I(stratum == "urban"),
         subset(design, stratum != "rural-remote"))
```

Overlapping intervals are not a test — In Python

```
# The difference of two estimates, with its own standard error
diff = p_urban - p_rural
se_diff = math.sqrt(se_urban**2 + se_rural**2)
print(f"difference {diff:.1%}, 95% CI {diff - 1.99 * se_diff:.1%} to "
      f"{diff + 1.99 * se_diff:.1%}")
```

Rounding, and false precision — In Python

```
print(f"{p:.1%} (95% CI {low:.1%} to {high:.1%})")
```

Rounding, and false precision — In R

```
sprintf("%.1f%% (95%% CI %.1f-%.1f)", 100 * p, 100 * low, 100 * high)
```

Rounding, and false precision

- **One decimal place** for a percentage from a survey of this size. Two implies a precision of 0.01 points, which is a...
- **Round the interval outwards**, never inwards. 25.47–32.71 becomes 25.4–32.8.
- **Never round the point estimate to a threshold.** An estimate of 14.96% reported as "15%" beside a 15% emergency...

The sentence

Household food insecurity was estimated at **29.1%** (95% CI 25.5–32.7), from 996 households in 75 clusters, weighted to the sampling frame and adjusted for non-response; design effect 1.60, effective sample 623.

The sentence

Food insecurity affects 32.8% of households.

Reading an interval against a threshold — In Python

```
THRESHOLD = 0.15

if high < THRESHOLD:
    verdict = "below the threshold"
elif low > THRESHOLD:
    verdict = "above the threshold"
else:
    verdict = "cannot be distinguished from the threshold"
print(verdict)
```

Reading an interval against a threshold — In R

```r
dplyr::case_when(
  high < 0.15 ~ "below the threshold",
  low  > 0.15 ~ "above the threshold",
  TRUE        ~ "cannot be distinguished from the threshold"
)
```

What comes next

- The interval tells you what your whole sample supports.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/survey-analysis-sampling/survey-07-confidence-intervals