

Mini-Project 3: Confidence Intervals Simulation Study

Tanner Bessette

2025-03-05

I have followed all rules for collaboration for this project, and I have not used generative AI on this project. Tanner Bessette

Step 1

3 different sample sizes:

small: $n = 5$

medium: $n = 40$

large: $n = 1000$

2 different values for p :

$p = 0.5$ and $p = 0.1$

Steps 2,3,4

Large n ($n = 1000$) and $p = 0.5$

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.1
```

```

v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()   masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become

```

```

n <- 1000
p <- 0.5

generate_samp_prop <- function(n, p) {
  x <- rbinom(1,n,p)

  # number of successes divided by sample size
  phat <- x / n

  # we have to use 1.645 instead of 1.96 bc 90% confidence
  lb <- phat - 1.645 * sqrt(phat * (1 - phat) / n)
  ub <- phat + 1.645 * sqrt(phat * (1 - phat) / n)

  prop_df <- tibble(phat, lb, ub)
  return(prop_df)
}

```

```

# run the function with our assigned n and p values
generate_samp_prop(n = 1000, p = 0.5)

# A tibble: 1 x 3
  phat    lb    ub
<dbl> <dbl> <dbl>
1 0.521 0.495 0.547

```

```

# we want 5000 ci's
n_sim = 5000

prop_ci_df <- map(1:n_sim,
  \(i) generate_samp_prop(n = 1000, p = 0.5)) |>
  bind_rows()

prop_ci_df

```

```
# A tibble: 5,000 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1 0.513 0.487 0.539
2 0.492 0.466 0.518
3 0.496 0.470 0.522
4 0.524 0.498 0.550
5 0.524 0.498 0.550
6 0.511 0.485 0.537
7 0.511 0.485 0.537
8 0.488 0.462 0.514
9 0.5    0.474 0.526
10 0.499 0.473 0.525
# i 4,990 more rows
```

```
prop_ci_df <- prop_ci_df |> mutate(ci_width = ub - lb,
                                   ci_cover_ind = if_else(p > lb & p < ub,
                                                         true = 1,
                                                         false = 0))

# output the average interval widths and the coverage rates
prop_ci_df |> summarise(avg_width = mean(ci_width),
                       coverage_rate = mean(ci_cover_ind))
```

```
# A tibble: 1 x 2
  avg_width coverage_rate
  <dbl>         <dbl>
1 0.0520         0.901
```

For $n = 1000$ and $p = 0.5$, we have an average interval width of 0.0520, and a coverage rate of 89.5%.

Large n ($n = 1000$) and $p = 0.1$

```
n <- 1000
p <- 0.1

# run the function with our assigned n and p values
generate_samp_prop(n = 1000, p = 0.1)
```

```
# A tibble: 1 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1 0.095 0.0797 0.110
```

```
# we want 5000 ci's
n_sim = 5000

prop_ci_df <- map(1:n_sim,
  \(i) generate_samp_prop(n = 1000, p = 0.1)) |>
  bind_rows()

prop_ci_df
```

```
# A tibble: 5,000 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1 0.103 0.0872 0.119
2 0.087 0.0723 0.102
3 0.111 0.0947 0.127
4 0.095 0.0797 0.110
5 0.09  0.0751 0.105
6 0.101 0.0853 0.117
7 0.107 0.0909 0.123
8 0.101 0.0853 0.117
9 0.095 0.0797 0.110
10 0.111 0.0947 0.127
# i 4,990 more rows
```

```
prop_ci_df <- prop_ci_df |> mutate(ci_width = ub - lb,
  ci_cover_ind = if_else(p > lb & p < ub,
    true = 1,
    false = 0))

# output the average interval widths and the coverage rates
prop_ci_df |> summarise(avg_width = mean(ci_width),
  coverage_rate = mean(ci_cover_ind))
```

```
# A tibble: 1 x 2
  avg_width coverage_rate
```

	<dbl>	<dbl>
1	0.0311	0.897

For $n = 1000$ and $p = 0.1$, we have an average interval width of 0.0312, and a coverage rate of exactly 90%.

Medium n ($n = 40$) and $p = 0.5$

```
n <- 40
p <- 0.5

# run the function with our assigned n and p values
generate_samp_prop(n = 40, p = 0.5)
```

```
# A tibble: 1 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1 0.525 0.395 0.655
```

```
# we want 5000 ci's
n_sim = 5000

prop_ci_df <- map(1:n_sim,
  \(i) generate_samp_prop(n = 40, p = 0.5)) |>
  bind_rows()

prop_ci_df
```

```
# A tibble: 5,000 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1 0.475 0.345 0.605
2 0.525 0.395 0.655
3 0.45  0.321 0.579
4 0.525 0.395 0.655
5 0.5    0.370 0.630
6 0.425 0.296 0.554
7 0.475 0.345 0.605
8 0.5    0.370 0.630
```

```

9 0.475 0.345 0.605
10 0.4 0.273 0.527
# i 4,990 more rows

```

```

prop_ci_df <- prop_ci_df |> mutate(ci_width = ub - lb,
                                   ci_cover_ind = if_else(p > lb & p < ub,
                                                           true = 1,
                                                           false = 0))

# output the average interval widths and the coverage rates
prop_ci_df |> summarise(avg_width = mean(ci_width),
                        coverage_rate = mean(ci_cover_ind))

```

```

# A tibble: 1 x 2
  avg_width coverage_rate
    <dbl>         <dbl>
1 0.257         0.914

```

For $n = 40$ and $p = 0.5$, we have an average interval width of 0.257, and a coverage rate of 91.7%.

Medium n ($n = 40$) and $p = 0.1$

```

n <- 40
p <- 0.1

# run the function with our assigned n and p values
generate_samp_prop(n = 40, p = 0.1)

```

```

# A tibble: 1 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1 0.05 -0.00669 0.107

```

```

# we want 5000 ci's
n_sim = 5000

prop_ci_df <- map(1:n_sim,

```

```

      \ (i) generate_samp_prop(n = 40, p = 0.1)) |>
      bind_rows()

prop_ci_df

# A tibble: 5,000 x 3
  phat      lb      ub
  <dbl>    <dbl> <dbl>
1 0.075  0.00649 0.144
2 0.125  0.0390  0.211
3 0.1    0.0220  0.178
4 0.1    0.0220  0.178
5 0.15   0.0571  0.243
6 0.05  -0.00669 0.107
7 0.075  0.00649 0.144
8 0      0        0
9 0.2    0.0960  0.304
10 0.05  -0.00669 0.107
# i 4,990 more rows

prop_ci_df <- prop_ci_df |> mutate(ci_width = ub - lb,
                                   ci_cover_ind = if_else(p > lb & p < ub,
                                                         true = 1,
                                                         false = 0))

# output the average interval widths and the coverage rates
prop_ci_df |> summarise(avg_width = mean(ci_width),
                       coverage_rate = mean(ci_cover_ind))

# A tibble: 1 x 2
  avg_width coverage_rate
  <dbl>         <dbl>
1    0.149         0.901

```

For $n = 40$ and $p = 0.1$, we have an average interval width of 0.150, and a coverage rate of 90.6%.

Small n ($n = 5$) and $p = 0.5$

```
n <- 5
p <- 0.5

# run the function with our assigned n and p values
generate_samp_prop(n = 5, p = 0.5)

# A tibble: 1 x 3
  phat    lb    ub
<dbl> <dbl> <dbl>
1  0.6 0.240 0.960

# we want 5000 ci's
n_sim = 5000

prop_ci_df <- map(1:n_sim,
  \(i) generate_samp_prop(n = 5, p = 0.5)) |>
  bind_rows()

prop_ci_df

# A tibble: 5,000 x 3
  phat    lb    ub
<dbl> <dbl> <dbl>
1  0.6 0.240 0.960
2  0.4 0.0396 0.760
3  0.4 0.0396 0.760
4  0.4 0.0396 0.760
5  0.4 0.0396 0.760
6  0.4 0.0396 0.760
7  0.6 0.240 0.960
8  0.6 0.240 0.960
9  0.4 0.0396 0.760
10 0.4 0.0396 0.760
# i 4,990 more rows

prop_ci_df <- prop_ci_df |> mutate(ci_width = ub - lb,
  ci_cover_ind = if_else(p > lb & p < ub,
```



```

true = 1,
false = 0))

# output the average interval widths and the coverage rates
prop_ci_df |> summarise(avg_width = mean(ci_width),
                        coverage_rate = mean(ci_cover_ind))

# A tibble: 1 x 2
  avg_width coverage_rate
    <dbl>         <dbl>
1    0.631         0.617

```

For $n = 5$ and $p = 0.5$, we have an average interval width of 0.634, and a coverage rate of 62.6%.

Small n ($n = 5$) and $p = 0.1$

```

n <- 5
p <- 0.1

# run the function with our assigned n and p values
generate_samp_prop(n = 5, p = 0.1)

# A tibble: 1 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1     0     0     0

# we want 5000 ci's
n_sim = 5000

prop_ci_df <- map(1:n_sim,
  \ (i) generate_samp_prop(n = 5, p = 0.1)) |>
  bind_rows()

prop_ci_df

```

```
# A tibble: 5,000 x 3
  phat    lb    ub
  <dbl> <dbl> <dbl>
1  0.2 -0.0943 0.494
2  0.2 -0.0943 0.494
3  0    0      0
4  0    0      0
5  0    0      0
6  0.2 -0.0943 0.494
7  0.2 -0.0943 0.494
8  0    0      0
9  0    0      0
10 0    0      0
# i 4,990 more rows
```

```
prop_ci_df <- prop_ci_df |> mutate(ci_width = ub - lb,
                                   ci_cover_ind = if_else(p > lb & p < ub,
                                                         true = 1,
                                                         false = 0))

# output the average interval widths and the coverage rates
prop_ci_df |> summarise(avg_width = mean(ci_width),
                       coverage_rate = mean(ci_cover_ind))
```

```
# A tibble: 1 x 2
  avg_width coverage_rate
  <dbl>         <dbl>
1  0.249         0.398
```

For $n = 5$ and $p = 0.1$, we have an average interval width of 0.253, and a coverage rate of 40.1%.

Table

Table 1: Table of Results

		$n = 5$	$n = 40$	$n = 1000$
$p = 0.5$	Coverage Rate	62.6%	91.7%	89.5%
$p = 0.1$	Coverage Rate	40.1%	90.6%	90%

		$n = 5$	$n = 40$	$n = 1000$
$p = 0.5$	Average Width	0.634	0.253	0.052
$p = 0.1$	Average Width	0.253	0.150	0.0312

Large Sample Assumption Calculations

Check that:

$$n * \hat{p} > 10 \text{ and } n * (1 - \hat{p}) > 10$$

are both satisfied for the large sample assumption to hold.

Setting 1: Large n ($n = 1000$) and $p = 0.5$

$$1000 * 0.5 > 10 \text{ and } 1000 * (1 - 0.5) > 10 \text{ both true}$$

So, the large sample assumption holds.

Setting 2: Large n ($n = 1000$) and $p = 0.1$

$$1000 * 0.1 > 10 \text{ and } 1000 * (1 - 0.1) > 10 \text{ both true}$$

So, the large sample assumption holds.

Setting 3: Medium n ($n = 40$) and $p = 0.5$

$$40 * 0.5 = 20 > 10 \text{ and } 40 * (1 - 0.5) = 20 > 10 \text{ both true}$$

So, the large sample assumption holds.

Setting 4: Medium n ($n = 40$) and $p = 0.1$

$$40 * 0.1 = 4 < 10$$

So, the large sample assumption does not hold.

Setting 5: Small n ($n = 5$) and $p = 0.5$

$$5 * 0.5 = 2.5 < 10$$

So, the large sample assumption does not hold.

Setting 6: Small n ($n = 5$) and $p = 0.1$

$$5 * 0.1 = 0.5 < 10$$

So, the large sample assumption does not hold.

Mini-Project Summary

Generally, the bigger the sample size n , the smaller the average width. The smaller the sample size, the larger the average width, especially with our extremely low sample size ($n = 5$), where we had average widths of 0.634 ($p = 0.5$) and 0.253 ($p = 0.1$)! For all of the different n 's, the simulations yielded a larger average width for $p = 0.5$ than for $p = 0.1$ (about twice as large of a width for $p = 0.5$ than $p = 0.1$).

From our large sample assumption calculations, the settings that have sufficiently large n are $n = 1000$ and $p = 0.5$, $n = 1000$ and $p = 0.1$, and $n = 40$ and $p = 0.5$. For these settings, we are able to interpret: "We are 90% confident that the true population proportion p is contained within our confidence intervals." The settings that do not have sufficiently large n are $n = 40$ and $p = 0.1$, $n = 5$ and $p = 0.5$, and $n = 5$ and $p = 0.1$. In these settings, we do not have a large enough n to trust the confidence interval to be reliable.

With $n = 1000$ and $n = 40$ the coverage rates are extremely close to 90%, which is our confidence interval. For $n = 1000$ and $p = 0.5$, the coverage rate is likely only 89.5% (and not 90%) due to random variation. When the n is too small, i.e. $n = 5$, the coverage rate also goes way down (far below 90% coverage). This is likely in part due to the fact that these two settings weren't even close to satisfying the large sample assumption.

The overall takeaway is that if we get a large enough n , we can have a coverage rate very close, if not exactly equal to, our confidence level, and the average width will be smaller with higher n . With a smaller n , we lose some of the accuracy in coverage rate, and the interval widths grow larger.