tack

Cameron Bale

1/25/2019

Load library.

```
library(tidyverse)

## -- Attaching packages --------------------------------------------------
------------ tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## -- Conflicts -----------------------------------------------------------
------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Coverage is the number of times that the confidence interval contains the true value of interest.

We know that the true probability that the tack points up is 0.4. We want to assess how many times a confidence interval (built from a finite sample of 50) contains the true probability of the tack pointing up.

We will assess this by performing a simulation study.

```
# create 'tack_flipper' which takes as arguements the number of tack flips,
the probability
# that the point is up, and the alpha value. 1 - alpha is how confident we
want to be
tack_flipper <- function (n_flips, prob_point, alpha) {

  # take a sample of 50 flips
  flips <- sample(x = c(1, 0), size = n_flips, prob = c(prob_point, 1 -
prob_point), replace = TRUE)
  # get the proportion of flips that landed with point up
  est <- mean(flips)
  # construct a confidence interval for the probability that the point lands
up
  ci <- est + c(-1, 1) * qnorm(1 - alpha / 2) * sqrt(est * (1 - est) /
n_flips)
  # assess whether the true probability of 0.4 is contained in the confidence
interval
```

```
  covered <- ci[1] < 0.4 && 0.4 < ci[2]

  # report the confidence interval, the probability estimate, and whether the
interval
  # contains the true probability value
  data_frame(
    lower = ci[1],
    probability = est,
    upper = ci[2],
    contains_true = covered
  )

}
```

Test.

```
# test the tack_flipper function with 50 flips, a probability of point up of
0.4, and
# an alpha of .05
tack_flipper(50, 0.4, .05)
```

```
## # A tibble: 1 x 4
##   lower probability upper contains_true
##   <dbl>       <dbl> <dbl> <lgl>
## 1 0.302        0.44 0.578 TRUE
```

Looks good. Lets perform a MC simulation.

```
# create a function 'mc_simulator' which takes arguements for the number of
repititions
# to be performed in the simulation, and the arguements to the function
'tack_flipper'
mc_simulator <- function (n_reps, n_flips, prob_point, alpha) {

  # set the number of repititions
  reps <- 1:n_reps

  # perform tack_flipper for the specified number of repititions
  mc_sim <- reps %>%
    map_df(function (x) tack_flipper(n_flips, prob_point, alpha))

  # obtain the MC estimate coverage of the confidence interval process
  mc_est <- mean(mc_sim$contains_true)

  # obtain a confidence interval for the MC estimate (assessing MC error)
  mc_sim %>%
    summarize(lower = mc_est - qnorm(1 - alpha / 2) * sqrt(mc_est * (1 -
mc_est) / length(reps)),
              ci_coverage = mean(contains_true),
              upper = mc_est + qnorm(1 - alpha / 2) * sqrt(mc_est * (1 -
mc_est) / length(reps)))
}
```

Try out the MC simulation with 1000 repititions, 50 flips per repitition, probability of pointing up of 0.4, and an alpha of .05.

```
mc_simulator(1000, 50, 0.4, .05)

## # A tibble: 1 x 3
##   lower ci_coverage upper
##   <dbl>       <dbl> <dbl>
## 1 0.925        0.94 0.955
```

Reducing the sample size to 32, and try the simulation again.

```
mc_simulator(1000, 32, 0.4, .05)

## # A tibble: 1 x 3
##   lower ci_coverage upper
##   <dbl>       <dbl> <dbl>
## 1 0.938       0.951 0.964
```

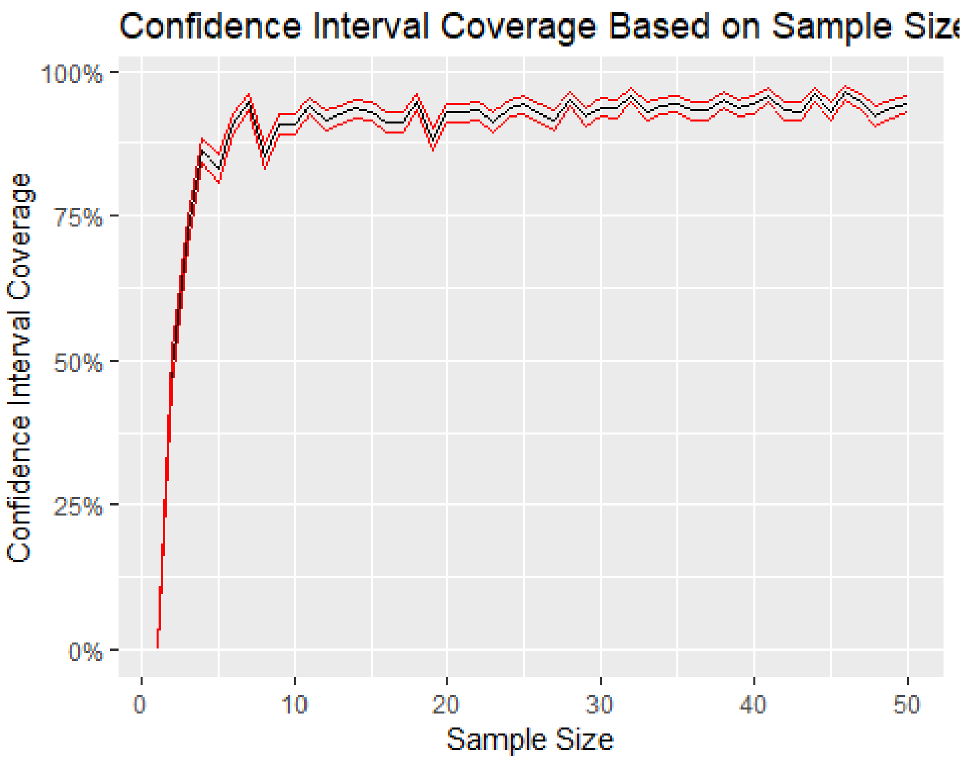
Investigate how sample size affects coverage.

```
# set the different sample sizes
s_sizes <- 1:50

# run the monte carlo simulation for sample sizes
sample_sims <- map_df(s_sizes, function (x) mc_simulator(1000, x, 0.4, .05))

# attach sample sizes to data
sample_sims <- sample_sims %>%
  mutate(sample_size = s_sizes)

# plot the monte carlo estimate of the coverage with upper and lower
confidence bounds in red
sample_sims %>%
  ggplot(aes(x = sample_size, y = ci_coverage)) +
  geom_line() +
  geom_line(aes(y = lower), col = 'red') +
  geom_line(aes(y = upper), col = 'red') +
  labs(x = 'Sample Size',
       y = 'Confidence Interval Coverage',
       title = 'Confidence Interval Coverage Based on Sample Size') +
  scale_y_continuous(labels = scales::percent)
```


Confidence Interval Coverage Based on Sample Size
100%
75%
50%
25%
0%
Confidence Interval Coverage
0
10
20
30
40
50
Sample Size