BST 222 Final Project - Simulation Study of HPSH's generosity in preparing dollar meals for its students

Bowen Gu, Jojo Kennedy, Pluto Zhang

2022-11-05

Suppose that $Y_1 \ldots, Y_n$ are random variables satisfying

$$Y_i = (\beta - 1)x_i + a\epsilon_i$$

where Y_i is the total profit that HSPH will lose due to providing dollar meals on day i, β is the actual cost of the dollar meal (we use $\beta - 1$ since HSPH will gain 1 dollar by selling one dollar meal), X_i is the number of people who purchased the dollar meal on day i, a is a constant that used to adjust the magnitude of food waste, which does not change through time, and ε_i is profit loss caused by food waste (e.g. some raw materials for the dollar meals are unconsumed and wasted) on day i. Here, we assumed that X_i follows a normal distribution $N(\mu, \sigma^2)$ with μ = 500 and σ = 50. The above number estimation is based on the facts about the students and faculties in HSPH. For ε_i, we assume that it follows the standard half normal distribution since we want the profit loss due to food waste to be non-negative and have a decreasing probability as the profit loss goes. The constant a is set to be 100 to mimic a more realistic profit loss due to food waste.

We used the following three methods to estimate the parameter β:

Estimator 1: $\sum_{i=1}^{n}(XiYi) \,/\, \sum_{i=1}^{n}(Xi^2) - \left(\frac{a\mu\sqrt{\frac{2}{\pi}}}{\sigma^2+\mu^2}\right)$

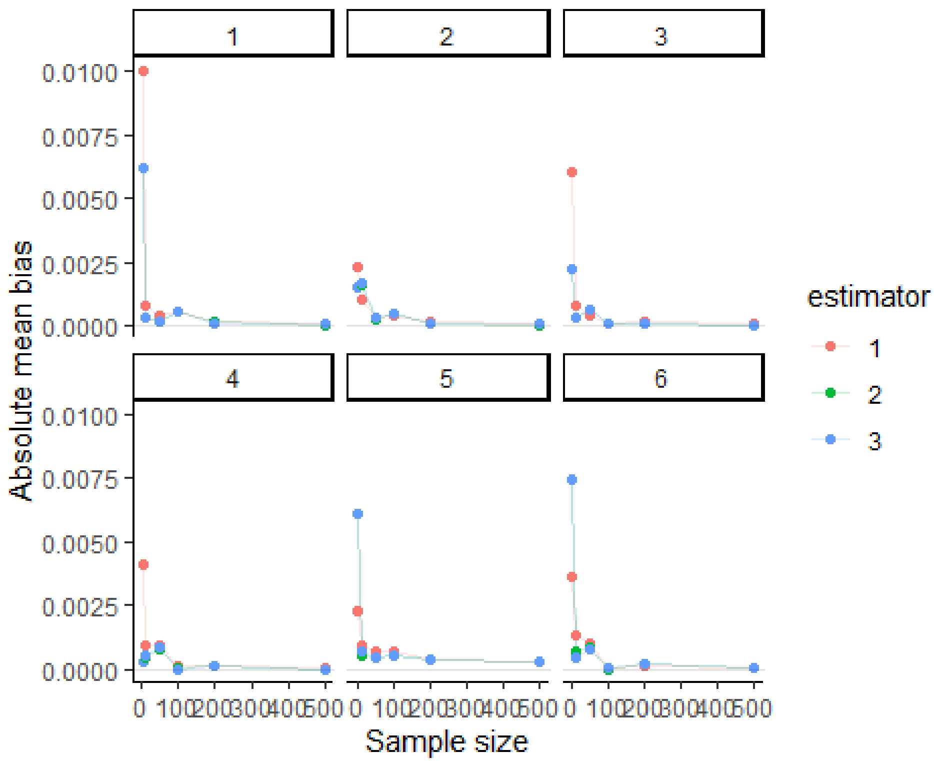
Estimator 2: $\sum_{i=1}^{n}\left(Yi - a\sqrt{\frac{2}{\pi}}\right) / \sum_{i=1}^{n} X\,i$

Estimator 3: $\frac{1}{n}\sum_{i=1}^{n}\left(\left(Yi - a\sqrt{\frac{2}{\pi}}\right) / Xi\right)$

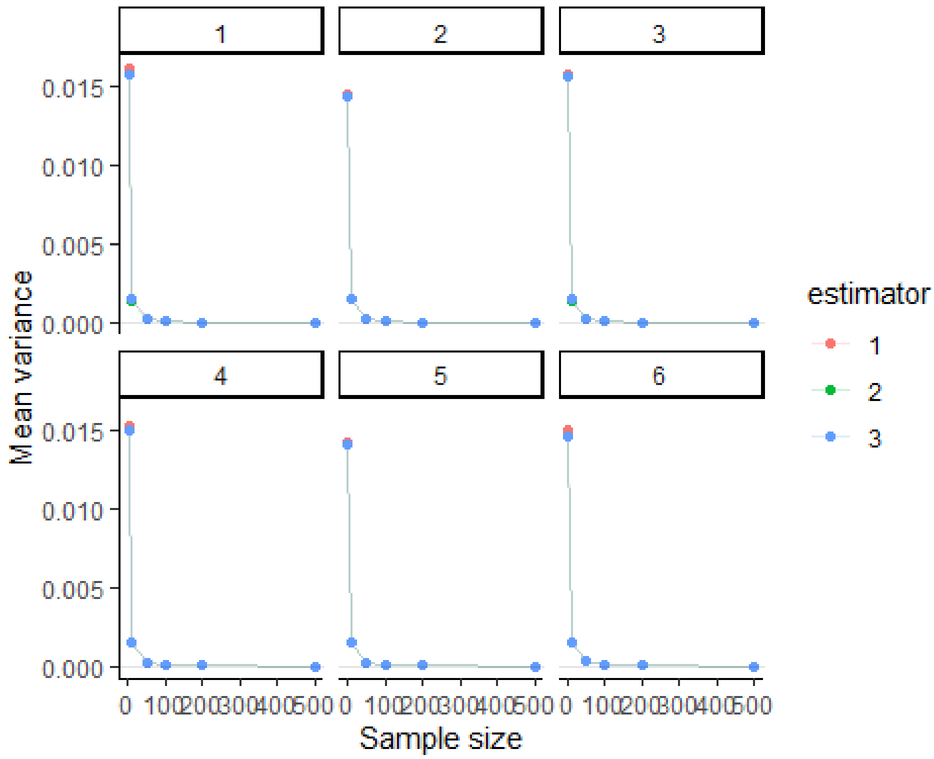
```
# Work through simulation example

# Set global parameters (These may be tweaked later)
set.seed(1)
mu = 500
sigma = 50
a = 100

# In the actual implementation, we played the following tricks. Instead of es
timating beta directly, we choose to estimate (beta - 1) and names it beta-pr
```

```
ime. After we get and estimation of beta-prime, we do (beta-prime - 1) to get
the estimation of beta

# Run through simulation results using lapply and ggplot2s
sim_results <-
  rbindlist(lapply(1:1000, function(i) {
    rbindlist(lapply(c(1, 10, 50, 100, 200, 500), function(n) {
      x_sim <- rnorm(n, mean = mu, sd = sigma)
      rbindlist(lapply(c(0, 1, 2, 3, 4, 5), function(beta) { # c(0, 1, 2, 3,
4, 5) is beta-prime instead of beta
        eps <- rhalfnorm(n) # standard half normal distribution with sigma =
1
        y = beta*x_sim + a * eps

        # Calculate beta hat from different estimators
        est_1_beta = sum(x_sim*y) / sum(x_sim^2) - (a * mu * sqrt(2 / pi) / (
sigma ^ 2 + mu ^ 2))
        est_2_beta = sum(y - a * sqrt(2 / pi)) / sum(x_sim)
        est_3_beta = (1/n)*sum((y - a * sqrt(2 / pi)) / x_sim)

        # Calculate analytic variance
        var_est_1 = var(y) / sum(x_sim^2)
        var_est_2 = (n*var(y)) / (sum(x_sim)^2)
        var_est_3 = (var(y)/n^2) * (sum(1/(x_sim^2)))

        # Return estimates in data table
        data.table(n = n,
                   beta = beta,
                   estimator = factor(c("1", "2", "3")),
                   estimate = c(est_1_beta, est_2_beta, est_3_beta),
                   variances = c(var_est_1, var_est_2, var_est_3))
      }))
    }))
  }))

# Get results from simulation
bias_sim = sim_results[, .(mean_bias = mean(estimate - beta),
                           lower_bias = quantile(estimate - beta, 0.025),
                           upper_bias = quantile(estimate - beta, 0.975)),
                       by = c("n", "beta", "estimator")]


mse_sim <- sim_results[, .(mean_mse = mean((estimate - beta) ^ 2),
                           lower_mse = quantile((estimate - beta) ^ 2, 0.025),
                           upper_mse = quantile((estimate - beta) ^ 2, 0.975))
,
                      by = c("n", "beta", "estimator")]

estimator_var <- sim_results[, .(var_est = var(estimate)), by = c("n", "beta"
```

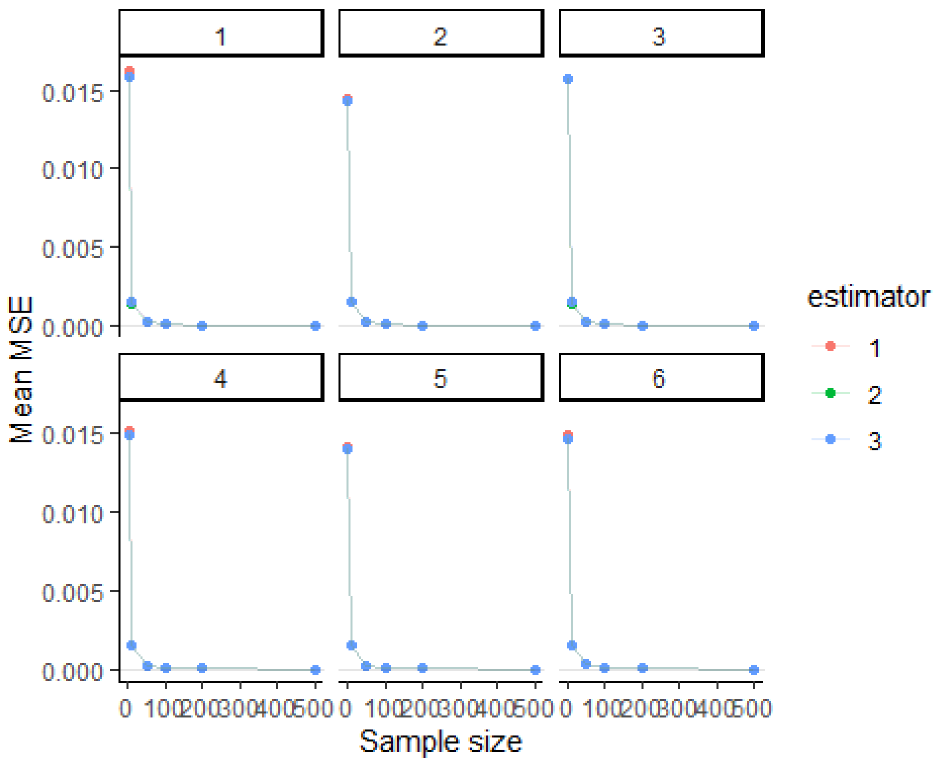
```r
, "estimator")]



# Plot of bias, facet by beta
bias_sim %>%
  ggplot(aes(x = n, y = abs(mean_bias), color = estimator)) +
  #geom_pointrange(aes(ymin = lower_bias, ymax = upper_bias)) +
  geom_line(alpha = 0.2) +
  geom_point() +
  geom_abline(slope = 0, intercept = 0, alpha = 0.1) +
  facet_wrap(~(beta+1)) +
  labs(x = "Sample size", y = "Absolute mean bias") +
  theme_classic()
```




```r
# Plot of variance, facet by beta
estimator_var %>%
  ggplot(aes(x = n, y = var_est, color = estimator)) +
  geom_line(alpha = 0.2) +
  #geom_ribbon(aes(ymin = lower_mse, ymax = upper_mse, fill = estimator, colo
r = NULL), alpha = 0.2) +
  geom_point() +
  geom_abline(slope = 0, intercept = 0, alpha = 0.1) +
  facet_wrap(~(beta+1)) +
  labs(x = "Sample size", y = "Mean variance") +
  theme_classic()
```




```r
# Plot of MSE, facet by beta
mse_sim %>%
  ggplot(aes(x = n, y = mean_mse, color = estimator)) +
  #geom_ribbon(aes(ymin = lower_mse, ymax = upper_mse, fill = estimator, colo
r = NULL), alpha = 0.2) +
  geom_point() +
  geom_line(alpha = 0.2) +
  scale_shape_discrete("Beta value") +
  geom_abline(slope = 0, intercept = 0, alpha = 0.1) +
  facet_wrap(~(beta+1)) +
  labs(x = "Sample size", y = "Mean MSE") +
  theme_classic()
```



1
2
3
4
5
6
0.015
0.010
0.005
0.000
Mean MSE
0
100
200
300
400
500
Sample size
estimator
1
2
3