Bootstrap Overdispersion Parameter Variability

William Ruth

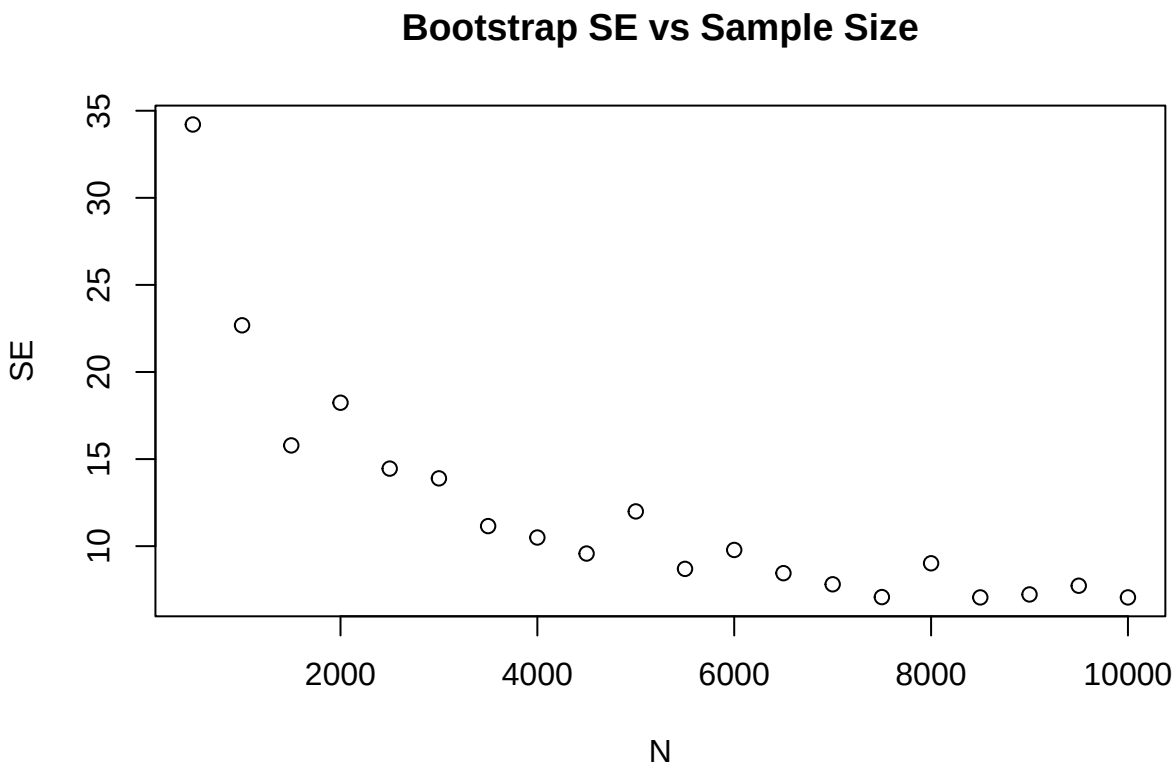
29/11/2021

We use a boostrap analysis with various sample sizes to estimate the standard error of the quasibinomial dispersion parameter.

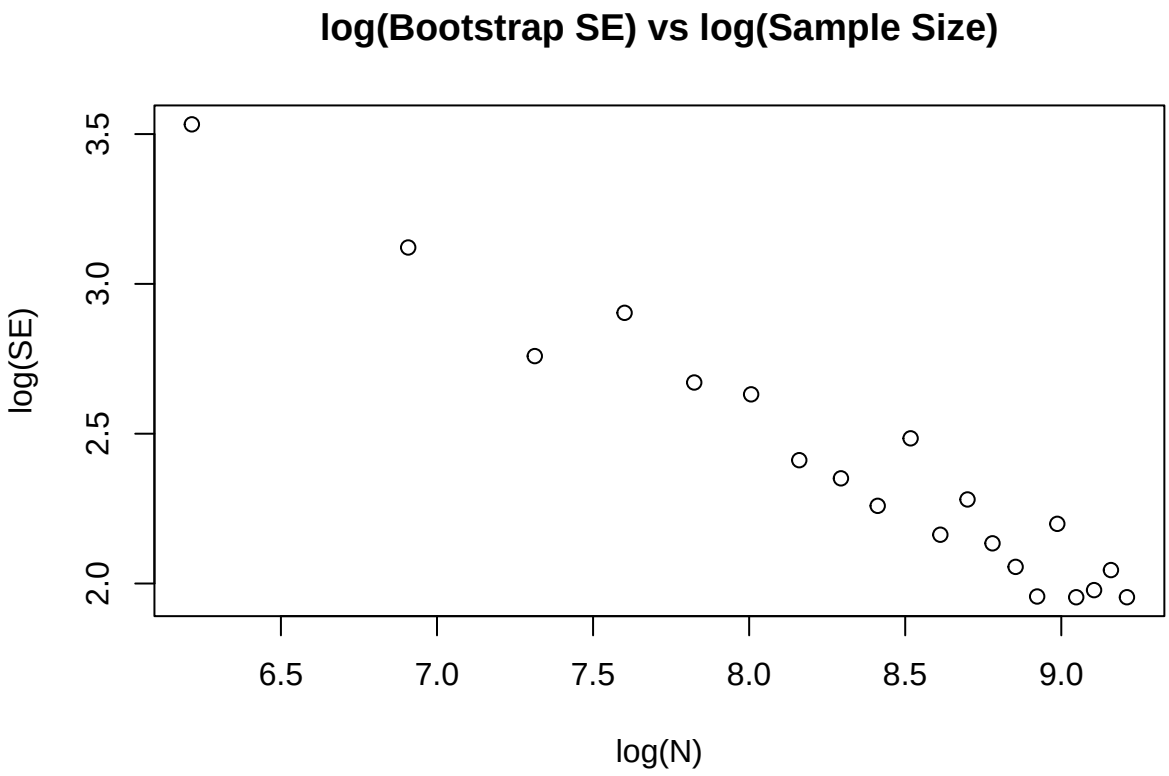
```
set.seed(1)

all_Ms <- seq(500, 10000, by = 500)
B = 100


### Draw boostrap samples of size M ###

all_bootstrap_SEs = c()

for(i in seq_along(all_Ms)){
  # print(paste0(i, " of ", length(all_Ms)))
  M = all_Ms[i]


  bootstrap_indices <- lapply(1:B, function(b) {
    sample(1:num_trials, M)
  })

  bootstrap_samples = lapply(bootstrap_indices, function(inds) {
    data_logit[inds, ]
  })


  bootstrap_estimates = sapply(bootstrap_samples, function(boot_data) {
    fit_glm_boot <- glm(
      form,
      family = quasibinomial(),
      data = boot_data,
      weights = rep(num_students, times = M)
    )
    summary(fit_glm_boot)$dispersion
  })


  all_bootstrap_SEs = c(all_bootstrap_SEs, sd(bootstrap_estimates))
}
```

Next, we make some plots of the relationship between the boostrap sample size, N, and the estimated standard error.

```r
data_boot = data.frame(N = all_Ms,
  SE = all_bootstrap_SEs)

with(data_boot, plot(N, SE, main = "Bootstrap SE vs Sample Size"))
```

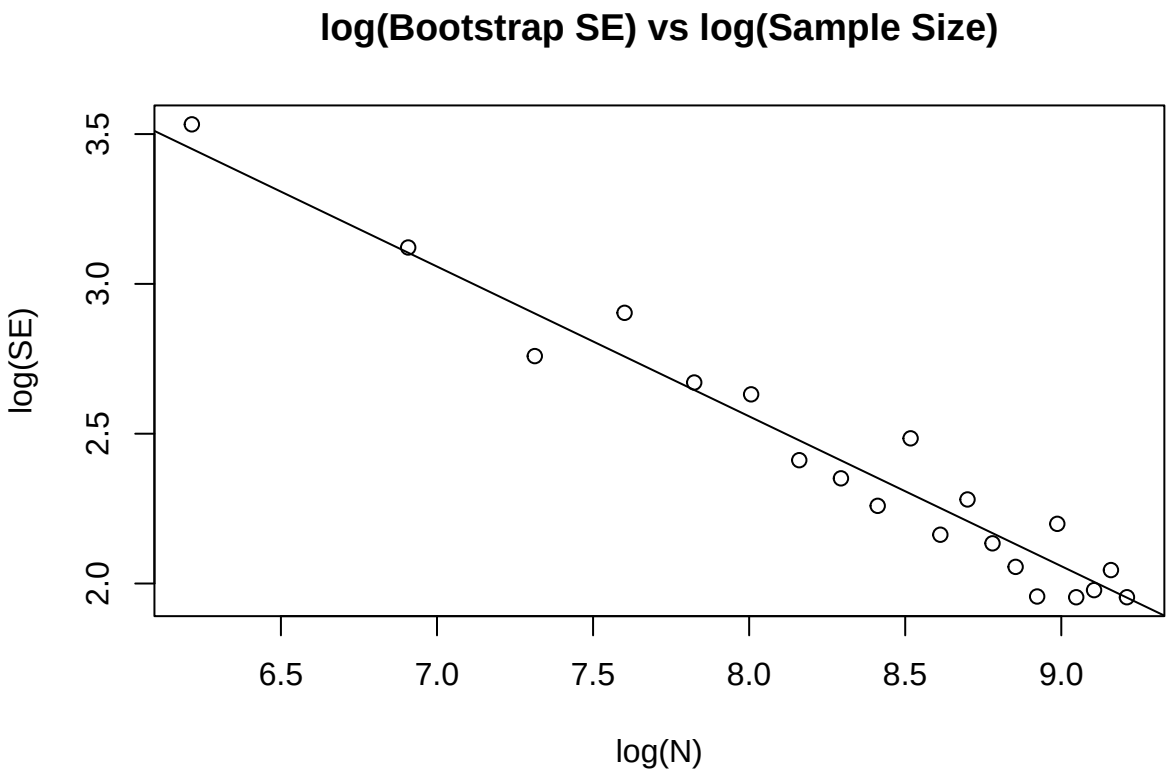



```r
with(data_boot, plot(log(N), log(SE), xlab = "log(N)", ylab = "log(SE)",
  main = "log(Bootstrap SE) vs log(Sample Size)"))
```



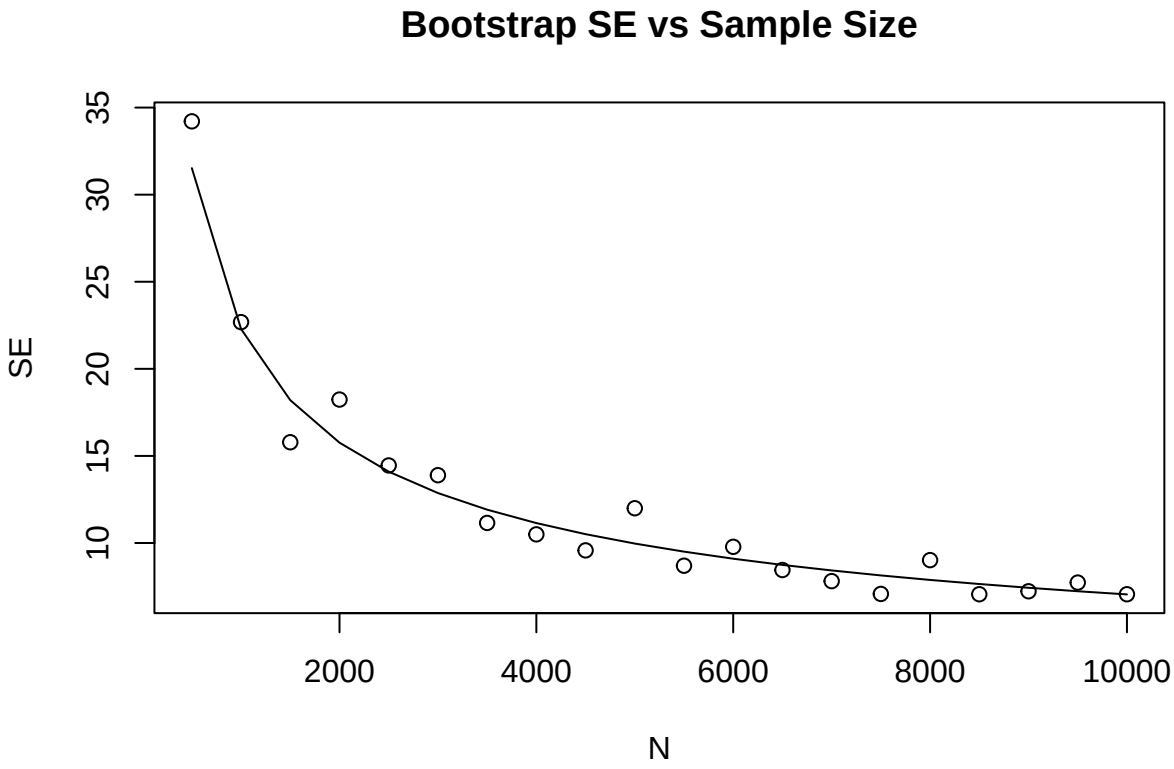
It looks like the relationship on the log-scale is approximately linear. Most estimators have standard error decaying like $1/\sqrt{n}$. Let's try fitting a linear regression model to the log-scale data with slope constrained to $-1/2$.

```
data_boot %<>% mutate(Y = log(SE) + 0.5*log(N))

fit = lm(Y ~ 1, data = data_boot)
intercept = fit$coefficients[1]

with(data_boot, plot(log(N), log(SE), xlab = "log(N)", ylab = "log(SE)",
  main = "log(Bootstrap SE) vs log(Sample Size)"))
abline(a = intercept, b = -0.5)
```



This looks like a pretty good fit. Returning to the original scale, our fit looks similarly accurate.

```
SD_Y = exp(intercept)
with(data_boot, plot(N, SE, main = "Bootstrap SE vs Sample Size"))
with(data_boot, lines(N, SD_Y / sqrt(N)))
```



Finally, extrapolating out to our actual sample size of 262440, we get the following predicted standard error.

```
SE_phi = SD_Y / sqrt(num_trials)
SE_phi
```

```
## (Intercept)
##    1.375871
```