Northwest Atlantic



NOT TO BE CITED WITHOUT PRIOR REFERENCE TO THE AUTHOR(S)

Fisheries Organization

Serial No. N7320

NAFO SCR Doc. 22/043

SCIENTIFIC COUNCIL MEETING - JUNE 2022

Approximating uncertainty around indices from stratified-random trawl surveys using the Gamma distribution

Paul M. Regular, Mariano Koen-Alonso, Semra Yalcin, Andrea M.J. Perreault, Laura J. Wheeland

Northwest Atlantic Fisheries Center, Fisheries and Oceans Canada, P.O.Box 5667, St. John's, NL, A1C 5X1, Canada

2024-11-18

Abstract

Many data-limited stock assessments rely on survey indices for the provision of science advice. Estimates of stock size are often derived from design-based approaches, but the quantification of uncertainty around these estimates remains a challenge. Standard practice has been to use quantiles from a Student's t distribution to approximate confidence intervals, but this method sometimes produces negative values for positive quantities like biomass and abundance. As an alternate method, we propose the use of the Gamma distribution to approximate uncertainty around survey indices. This involves the translation of unbiased design-based mean and variance estimators to the shape and scale parameters for the Gamma distribution. Via simulation testing, we show that densities and confidence intervals derived from the Gamma distribution closely match densities derived from bootstrapped samples of simulated survey data. Results also indicate that the Gamma approximation offers a reasonable survey-based approach for quantifying the probability that a stock is above or below a reference level. While preliminary, these results clearly demonstrate that the Gamma distribution offers a more realistic description of the uncertainty around survey indices than the Student's t distribution.

Introduction

A primary objective of fisheries-independent trawl surveys is to obtain indices of stock size and quantify the uncertainty around these indices. Such information plays a critical role in the assessment and management of fish stocks around the world, as they often serve as a leading indicator of trend and status (Kimura and Somerton, 2006; Pennington and Strømme, 1998). Quantifying uncertainty around stock size, trends, and status is a challenging but necessary task, as fisheries management increasingly relies on such information to characterize the risk associated



with alternative management options. Whenever possible, this quantification of uncertainty is done using stock assessment models, which are often calibrated using survey-based indices. However, data limitations, complex stock dynamics, and operational capacity constraints often preclude the implementation of complex stock assessment models. In these cases, the assessment of the stock tends to be based on the survey indices themselves. This may include any additional biological and ecological information that is available. Therefore, when assessments are survey-based, the quantification of uncertainty of the survey indices becomes a primary source for informing risk.

Fisheries surveys follow standardized sampling methods, and the indices derived from them are commonly design-based estimators. One of the most common survey sampling designs is the stratified-random sampling with proportional allocation, where the total sampling area is subdivided into strata, and the number of randomly allocated sampling stations within a stratum is informed by the stratum characteristics (e.g., area) and the minimum number of stations required for calculating stratum-specific statistics of the variable(s) of interest (e.g., abundance, biomass). More specifically, these statistics typically are the stratum mean and variance.

The overall mean and variance of the variable(s) of interest for the survey area are calculated from these stratum-specific indices using the general properties of the mean and variance. In the case of the mean, this translates into the straightforward idea that the overall mean represents a weighted average of the strata means, and where the weights applied are constants derived from the stratification scheme (e.g., **Cochran, 1977**). The standard mean and variance estimators from a stratified-random sampling have the important feature that they do not depend on any assumed distribution; these estimators are valid irrespective of the underlying distribution.

Unlike the mean and variance estimators themselves, the construction of confidence intervals (CIs) for the mean often requires assuming a probability distribution. Standard CIs are typically approximated using quantiles from a Normal distribution when sample size is large, or a Student's t distribution when sample sizes are small (i.e., few degrees of freedom per stratum; (Cochran, 1977)). While these distributions are commonly used, both of them allow for the CIs to include negative values, something that is not realistic when analyzing strictly positive quantities like biomass and abundance (Cadigan, 2011). The issue of negative values from these approximations is therefore an artifact of the distribution assumed in the construction of the CI rather than a problem with the stratified-random mean and variance estimators themselves. This justifies the application of resampling techniques like bootstrap to derive more realistic confidence intervals for stratified-random estimators.

While resampling techniques can be an effective tool to generate CIs, they do not come without their challenges. There are different ways in which resampling techniques can be implemented, and in the case of survey indices, there is little consensus on which implementation performs best. On the practical side, resampling techniques also require additional computing power and programming efforts. While these additional steps should not be a deterrent to implementing them when required, having a simpler alternative that addresses the issues linked to the normal or Student's t approximations can also be useful.

Therefore, the objective of this work is to examine the use of an alternative probability distribution to estimate CIs for fisheries surveys that does not produce negative values, and to compare these results with CIs derived from resampling techniques to evaluate its performance.

Methods

Selection of a probability distribution and rationale for the approach

In the Northwest Atlantic, major bottom trawl surveys follow a stratified-random sampling design with proportional allocation (e.g., **González-Troncoso et al., 2022**; **Rideout et al., 2022**), and the biomass and abundance indices are calculated accordingly (e.g., **Smith and Somerton, 1981**). A cursory examination of biomass and abundance indices from these surveys not only shows that there are no negative values (naturally), but that the distributions are seldom symmetrical; long positive tails are often observed.

3

Given these features, the Gamma distribution emerges as a logical choice for describing these data. This distribution is always positive, and depending on the parametrization, it can be fairly symmetrical (i.e., normal-like) or have a long right tail. We propose that by changing the assumed distribution from Normal or Student's t to Gamma, we can provide a better approximation of the confidence intervals. This approach avoids undesired features (e.g., negative lower values) and better resembles observed distributions (e.g., asymmetrical distributions with longer positive tails).

Provided data from a bottom trawl stratified-random survey, average biomass or abundance $(\hat{\mu})$ and sampling variance $(\hat{\sigma}^2)$ over the stock area can be estimated using the standard random-stratified design calculations (Cochran, 1977; Smith, 1990; Smith and Somerton, 1981). These estimates can then be used to calculate the scale $(\hat{\theta})$ and shape $(\hat{\kappa})$ parameters of the corresponding Gamma distribution as:

$$\hat{\theta} = \frac{\hat{\sigma}^2}{\hat{\mu}}$$
, and $\hat{\kappa} = \frac{\hat{\mu}^2}{\hat{\theta}}$ (1)

The resulting Gamma distribution is assumed to describe the data, and its quantiles are used to define the confidence intervals in the same way the Normal or Student's t distributions are used in the standard approach.

Aggregating estimates

Formally, this involves calculating the probability that the current index is above or below an average level $\hat{\mu}_r$ from a reference period *R*. Since the reference point $\hat{\mu}_r$ is an average of the index itself, its value cannot be perfectly known, and the uncertainty around it should also be taken into account in the estimation of the probability. Given that each individual index value included in the average has its own variance, a full consideration of the uncertainty around the average should also consider this variability. Therefore, to fully account for the uncertainty around $\hat{\mu}_r$, we need to combine the variances across the individual index estimates. This combined variance $\hat{\sigma}_r^2$ can be obtained by simply applying the general properties of the variance. Since $\hat{\mu}_r$ is effectively a linear combination of the individual indices (i.e., the sum of random variables multiplied by a constant—the inverse of sample size), the properties of the variance for a linear combination provide a way for estimating $\hat{\sigma}_r^2$. Furthermore, if we assume that the survey variances across years are independent, we can dismiss the covariances between years, and $\hat{\sigma}_r^2$ can be estimated as:

$$\hat{\mu}_r = \frac{\sum_{y=1}^{|R|} \hat{\mu}_y}{|R|}, \text{ and } \hat{\sigma}_r^2 = \frac{\sum_{y=1}^{|R|} \hat{\sigma}_y^2}{|R|^2}$$
 (2)

where $\hat{\mu}_y$ and $\hat{\sigma}_y^2$ are the estimated annual mean and variance for the index in each year *y* within the reference period *R*, and where |R| is the size of the period (i.e., the number of years used to

compute the average). As above, $\hat{\mu}_r$ and $\hat{\sigma}_r^2$ can be converted to $\hat{\theta}$ and $\hat{\kappa}$ parameters to characterize the uncertainty assuming a Gamma distribution (Equation (1)).

Simulation testing

While the proposed approach only involves a change in the assumption made about the distribution of the estimated mean, establishing if it is more realistic than the customary Normal/Student's t distribution assumption requires additional analysis. To address this issue, we use simulations to test the performance of our proposed approach in two relevant scenarios:

Scenario 1. The calculation of the confidence interval for a single survey index.

Scenario 2. The estimation of the probability that the stock is above or below some reference point.

We explored these two scenarios by simulating a redfish-like population using the R package SimSurvey (**Regular et al., 2020**). The simulated population was based on the exponential decay cohort model where parameter settings for mortality, recruitment, and growth were based on assessments of redfish in 30 and 3LN (see **Appendix A** for details). The simulated population was distributed through an area according to the age-year-space covariance with a parabolic relationship with depth. This survey area was 300 x 300 km with 10 km² cell size and had 30 depth-based strata. We simulated stratified random sampling with a 2 m wide trawl hauled for a distance of 1.5 km. The population and survey were simulated over 20 years. The number of sets in a stratum was proportional to its area (approximately 1 set per 1000 km²), and the minimum set per stratum was 2. The survey simulation was replicated five times over the same population.

To assess performance under Scenario 1, average trawlable abundance $(\hat{\mu})$ and sampling variance $(\hat{\sigma}^2)$ were calculated by year (1-20) and replicate (1-5) using standard design-based estimators (Smith, 1990; Smith and Somerton, 1981). These estimates were translated to scale $(\hat{\theta})$ and shape $(\hat{\kappa})$ parameters for the Gamma distribution as described above (Equation (1)). To compare probability densities obtained from the Gamma distribution with densities based on an empirical approach, we applied a non-parametric bootstrap to resample the observations (sets) independently within each stratum with replacement. The resampling and calculation of the mean bootstrap estimator were repeated 5000 times with the R package boot (Canty and Ripley, 2021). Densities from these bootstrap samples were computed for each year and survey replicate for comparison to the Gamma approximation. Finally, standard confidence intervals were computed using the t-distribution for comparison to the bootstrap and Gamma approximations.

To assess performance under Scenario 2, estimates from years 10-15 were treated as the reference period, and the average abundance and the associated variance were constructed by aggregating the annual estimates as described above (Equation (2)). These aggregate estimates were used to calculate the corresponding Gamma distribution parameters (Equation (1)) to characterize the average abundance distribution for the reference period. Likewise, average stratified estimates of abundance across years 10-15 were bootstrapped to obtain samples for comparison with the density from the Gamma approach. Using distributions derived from the Gamma distribution and bootstrap samples, the probability that the population estimate from year 20 is lower than the reference period was computed for comparison.

This simulation can be replicated using code in Appendix A.

Results and Discussion

Despite variability in central tendency caused by population dynamics and sampling noise, the shapes of the Gamma density and bootstrap samples were similar across all years and survey replicates (Figure 1). This result indicates that the Gamma distribution provides a reasonable approximation of the uncertainty around the stratified estimates. A comparison of the confidence intervals derived from the Gamma distribution relative to those obtained from bootstrapping (Figure 2) further supports this conclusion. The 95% CIs from the Gamma distribution appear to be slightly larger than those from the bootstrap method, but they clearly outperform the CIs approximated using the Student's t distribution (Figure 2) and avoid unrealistic negative values.

The Gamma approximation also appears to have comparable performance to a bootstrap approach when assessing the probability that the terminal estimate is below average levels from a reference period (Figure 3). Across all five survey replicates, the mean relative difference between the probabilities computed using the two methods was less than 0.04. While these results are based on only five replicates, and hence are necessarily preliminary, they consistently indicate that the Gamma approximation offers a reasonable option for reliably quantifying the uncertainty around stock status.

Notably, previous simulations have shown that bootstrapped upper bounds are often too low (e.g., Cadigan (2011)). The minor differences observed here, with the Gamma distribution producing slightly higher upper bounds (Figure 2), may suggest a marginally better performance by the Gamma approach.

Further testing is warranted to accurately determine the exact coverage of the 95% confidence intervals estimated using the Gamma distribution approach. Nonetheless, these results indicate that the Gamma distribution clearly outperforms the Student's t distribution, offering a more realistic description of the uncertainty around survey indices. Additionally, it has a generally comparable performance to resampling techniques. This suggests that using the Gamma distribution for approximating CIs is a preferable approach over the standard distributions typically used for this purpose.

References

- Cadigan, N. G. (2011). Confidence intervals for trawlable abundance from stratified-random bottom trawl surveys. *Canadian Journal of Fisheries and Aquatic Sciences*, *68*(5), 781–794.
- Canty, A., and Ripley, B. D. (2021). Boot: Bootstrap r (s-plus) functions.
- Cochran, W. G. (1977). Sampling techniques. John Wiley & Sons.
- González-Troncoso, D., Garrido, I., Rábade, S., Fabeiro, M., Román, E., Tarrío, C., Sánchez, J. M. C., and Alpoim, R. (2022). Results from Bottom Trawl Survey on Flemish Cap of June-July 2021. *NAFO SCR Doc*, *22/004*.
- Kimura, D. K., and Somerton, D. A. (2006). Review of statistical aspects of survey sampling for marine fisheries. *Reviews in Fisheries Science*, *14*(3), 245–283.
- Pennington, M., and Strømme, T. (1998). Surveys as a research tool for managing dynamic stocks. *Fisheries Research*, *37*(1-3), 97–106.



- Regular, P. M., Robertson, G. J., Lewis, K. P., Babyn, J., Healey, B., and Mowbray, F. (2020). SimSurvey: An R package for comparing the design and analysis of surveys by simulating spatiallycorrelated populations [Journal]. *PLOS ONE*, *15*(5), 1–28. https://doi.org/10.1371/journal.pone.0232822
- Rideout, R. M., Rogers, B., Wheeland, L., and Koen-Alonso, M. (2022). Temporal And Spatial Coverage Of Canadian (Newfoundland And Labrador Region) Spring And Autumn Multi-Species RV Bottom Trawl Surveys, With An Emphasis On Surveys Conducted In 2021. *NAFO SCR Doc*, 22/007.
- Smith, S. J. (1990). Use of statistical models for the estimation of abundance from groundfish trawl survey data. *Canadian Journal of Fisheries and Aquatic Sciences*, 47(5), 894–903.
- Smith, S. J., and Somerton, G. D. (1981). STRAP: A User-Oriented Computer Analysis System for Groundfish Research Trawl Survey Data (p. 66). Canadian Technical Report of Fisheries; Aquatic Sciences No. 1030.





Figure 1. The bootstrap and gamma distributions estimated using simulated data from five independent surveys conducted over the same population across 20 years. The blue area shows the density distribution from 1000 bootstrapped samples from each year and survey replicate. The red area shows the gamma probability distribution from each year and survey replicate based on the mean and standard deviation of the designbased index.



Figure 2. Lower and upper 95% confidence intervals derived from the Gamma and Student's t distributions relative to intervals derived from a bootstrap approach. R² values are indicated.



Figure 3. Bootstrap (blue) and gamma (red) distributions estimated from five simulated surveys of a redfish-like population, where terminal estimates (year 20; open area) are compared to a reference period (aggregate estimates from years 10-15; shaded area). Densities for the reference period were obtained by combining the bootstrap samples and by aggregate parameters across the reference period (see Methods section). Probability that the terminal value is below the reference point is indicated.

9

Appendix A

Simulation results can be replicated using the below code.

```
library(SimSurvey)
library(tidyr)
library (future)
library (tictoc)
library(ggplot2)
library(ggridges)
library(ggpubr)
library (patchwork)
library(dplyr)
library(purrr)
library(data.table)
library (NAFOdown)
plan(multisession, workers = floor(availableCores()/2))
n sims <- 5
n boot <- 5000
set.seed(794)
population <- sim_abundance(ages = 1:50,</pre>
                              years = 1:20,
                              R = sim_R (log_mean = log(60000000)),
                                         log_sd = 0.6,
                                         random_walk = F)
                              Z = sim_Z (log_mean = log(0.2)),
                                        log_sd = 0.2,
                                         phi_age = 0.4,
                                         phi_year = 0.4),
                              N0 = sim_N0(N0 = "exp", plot = FALSE),
growth = sim_vonB(Linf = 30, L0 = 0,
                                                 K = 0.1, log_sd = 0.13,
                                                  length_group = 1, digits = 0)) |>
  sim_distribution(grid = make_grid(x_range = c(-150, 150),
                                      y_range = c(-150, 150),
                                       res = c(10, 10),
                                       shelf_depth = 60,
                                       shelf_width = 170,
                                       depth_range = c(0, 1600),
                                       n div = 2,
                                       strat_breaks = seq(0, 1600, by = 65),
                                       strat_splits = 4,
                                       method = "bezier"),
                    ays_covar = sim_ays_covar(sd = 2,
                                                range = 200,
                                                phi_age = 0.5,
                                                phi_year = 0.9),
                    depth_par = sim_parabola(mu = log(190),
                                               sigma = 0.3,
                                                log_space = TRUE))
survey <- sim_survey(population,</pre>
                      n_sims = n_sims,
                      q = sim_logistic(k = 1, x0 = 6.5),
                      trawl_dim = c(1.5, 0.02),
                      resample cells = FALSE,
                      binom_error = TRUE,
                      min_sets = 2,
                      set_den = 1/1000,
                      lengths_cap = 250,
                      ages_cap = 20,
                      age_sampling = "stratified",
                      age_length_group = 1,
                      age_space_group = "division") |>
  run_strat()
## Density from the Gamma distribution ------
```

```
total_strat <- survey$total_strat |>
  mutate(sigma = sampling units * sd,
        scale = sigma ^ 2 / total,
        shape = total / scale)
## Use gamma to generate density by sim and year
rng <- c(0.001, max(total_strat$total) * 2)</pre>
x <- seq(rng[1], rng[2], length.out = 100)</pre>
total_strat_den <- lapply(seq.int(nrow(total_strat)), function(i) {</pre>
  data.frame(sim = total strat$sim[i],
             year = total_strat$year[i],
             total = x,
             den = dgamma(x, shape = total strat$shape[i],
                          scale = total_strat$scale[i]))
}) |> dplyr::bind_rows()
### Density from bootstrapping ------
                                                    setdet <- survey$setdet
split setdet <- split(setdet, paste0(setdet$year, "-", setdet$sim))</pre>
sumYst <- function(data, i = seq_len(nrow(data)), return_mean = FALSE) {</pre>
 x <- data[i, ] >
    ### stratum level
    group_by(year, strat, strat_area) |>
    summarise(meanYh = mean(n), tow_area = mean(tow_area), .groups = "drop_last") >
    mutate(Nh = strat_area/(tow_area)) >
    group_by(year) >
    mutate(N = sum(Nh), Wh = Nh/N, WhmeanYh = Wh * meanYh) >
    ### year level
    summarise(sumYst= mean(N) * sum(WhmeanYh), .groups = "drop_last") >
    pull(sumYst)
 if (return mean) { return(mean(x)) } else { return(x) }
}
boot_one_year <- function(data, reps) {</pre>
 b <- boot::boot(data, statistic = sumYst, strata = data$strat, R = reps)</pre>
 boot <- data.table(b$t) |> dplyr::rename(total = V1) |>
   mutate(samp = seq.int(reps), sim = mean(data$sim), year = mean(data$year))
 return(boot)
1
boot_index <- furrr::future_map_dfr(split_setdet, boot_one_year, reps = n_boot,</pre>
                                     .options = furrr::furrr_options(seed = TRUE))
quantile(boot index$total, prob = c(0.001, 0.999))
den plot <- ggplot() +</pre>
  geom_density_ridges(aes(x = total, y = as.numeric(year), group = factor(year)),
                      color = "grey90", fill = "steelblue", alpha = 0.7,
                      data = boot_index, scale = 1) +
  geom_density_ridges(aes(x = total, y = year, height = den, group = factor(year)),
                      stat = "identity", color = "grey90", fill = "red", alpha = 0.7,
                      data = total_strat_den, scale = -1) +
  coord_flip() + guides(fill = "none") +
  scale_x_continuous(labels = scales::label_number(suffix = "", scale = 1e-8),
                     limits = c(194587641, 5116017391)) +
  ylab("Year") + xlab("Abundance index") +
  facet_grid(rows = "sim") +
  theme_nafo()
## Relative status -----
### Gamma estimates for the reference years
ref est <- total strat >>
 filter(year %in% 10:15) >
  group_by(sim) |>
  summarise(total = mean(total),
            sigma = sqrt(sum(sigma ^ 2) / (n()^2)),
            scale = sigma ^ 2 / total,
            shape = total / scale)
### Bootstrapping for the reference years
ref setdet <- survey$setdet |>
```

```
filter(year %in% 10:15) >
 mutate(year strat = (year * 1000) + strat)
split_ref_setdet <- split(ref_setdet, paste0(ref_setdet$sim))</pre>
ref_boot_fn <- function(data, R) {</pre>
  b <- boot::boot(data, statistic = sumYst, strata = data$year strat, R = n boot, return_mean = TRUE)</pre>
 ref_boot <- data.table(b$t) |> dplyr::rename(total = V1) |>
    mutate(samp = seq.int(R), sim = mean(data$sim))}
ref_boot <- furrr::future_map_dfr(split_ref_setdet, ref_boot_fn, R = n_boot, .options = furrr::furrr_options(seed =</pre>
TRUE))
saveRDS(ref boot, file = "Gamma SCR/data/ref boot.rds")
ref boot <- readRDS("Gamma_SCR/data/ref_boot.rds")</pre>
### Sampling for the gamma distribution
x <- ref boot |>
  group_by(sim) |>
 summarise(seq = seq(min(total), max(total), length.out = 100))
ref den <- NULL
for(i in unique(ref est$sim)) {
  ref_den[[i]] <- x >
    filter(sim == i) >
    summarise(total= seq, den = dgamma(seq, shape = ref_est$shape[i],scale = ref_est$scale[i]))
}
ref_den <- Reduce('rbind', ref_den)</pre>
### Final year results
t est <- total strat |>
 filter(year == 20)
t den <- total strat den |>
 filter(year == 20)
t boot <- boot index |>
 filter(year == 20)
### Calculating the probability for the final year
boot_prob <- bind_rows(t_boot, ref_boot, .id = 'id') %>%
  group_by(sim) %>%
  summarise(boot_prob = mean((total[id == 1] - total[id == 2]) < 0), .groups = 'drop')</pre>
n samp <- 100000
ref samp <- map_df(1:nrow(ref est), function(i) {</pre>
 dat <- rgamma(n_samp, shape = ref_est$shape[i], scale = ref_est$scale[i])</pre>
  data.table(sim=i, sample=dat)
})
t samp <- map_df(1:nrow(t est), function(i) {</pre>
 dat <- rgamma(n_samp, shape = t_est$shape[i], scale = t_est$scale[i])</pre>
 data.table(sim=i, sample=dat)
})
gamma prob <- bind_rows(t samp, ref samp, .id = 'id') %>%
 group_by(sim) %>%
 summarise(gamma_prob = mean((sample[id == 1] - sample[id == 2]) < 0), .groups = 'drop')</pre>
### PLot
text_terminate <- cbind(ref_den |>
                            group_by(sim) |>
                            summarise(max_den = max(ref den$den)* 1.2),
                          total_x = t_est$total)
text_reference <- cbind(ref_den |>
                            group_by(sim) >
                            summarise(max_den = max(ref den$den)* 1.2),
                          total_x = ref_est$total)
prob_text <- cbind(t_est, boot_prob = boot_prob$boot_prob, gamma_prob = gamma_prob$gamma_prob)</pre>
ref plot <- ggplot() +</pre>
  geom_density(aes(x = total), data = ref boot, fill = "steelblue", color = "steelblue", alpha = 0.5) +
 facet grid(~sim)+
```

```
geom_area(aes(x = total, y = -den), data = ref_den, fill = "red", color = "red", alpha = 0.5) +
  geom density (aes (x = total), data = t boot, fill = NA, color = "steelblue", size = .nafo lwd) +
  geom_area(aes(x = total, y = -den), data = t_den, fill = NA, color = "red", size = .nafo_lwd) +
 geom_text(data = prob text, aes(x = total, y = 0, label = round(gamma prob, 2)),
            hjust = 1.2, vjust = 2, color = "red") +
  theme nafo() +
  coord_flip() +
  scale_x_continuous(labels = scales::label_number(suffix = "", scale = 1e-8),
                     limits = c(0, quantile(ref_boot$total, 0.9999))) +
  ylab("") + xlab("Abundance index") +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())
## Comparison CI plots ------
gamma ci <- total strat |>
  group_by(year, sim) |>
  mutate(lower95 = qgamma(0.025, shape = shape, scale = scale),
        upper95 = qgamma(0.975, shape = shape, scale = scale)) >
  distinct(lower95, upper95) >
  rename(lower95_gamma = lower95, upper95_gamma = upper95)
boot ci <- boot index |>
  group_by(year,sim) >
  mutate(lower95 = quantile(total, prob = c(0.025)),
        upper95 = quantile(total, prob = c(0.975))) |>
  distinct(lower95, upper95) |>
  rename(lower95 boot = lower95, upper95 boot = upper95)
all ci <- merge(gamma ci, boot ci)
total_gamma <- merge(gamma_ci, total_strat, by = c("sim", "year"))</pre>
total_boot<- merge(boot_ci, total_strat, by = c("sim", "year"))</pre>
gamma_plot <- data.frame(year = total_gamma$year, sim = total_gamma$sim,</pre>
                          total = total gamma$total, lower95 = total gamma$lower95 gamma,
                          upper95 = total_gamma$upper95_gamma, method = "Gamma")
boot_plot <- data.frame(year = total_boot$year, sim = total_boot$sim,</pre>
                         total = total_boot$total, lower95 = total_boot$lower95_boot,
                         upper95 = total boot$upper95 boot, method = "Bootstrap")
studentt_plot <- data.frame(year = total_gamma$year, sim = total_gamma$sim,</pre>
                              total = total_gamma$total, lower95 = total_gamma$total_lcl,
                             upper95 = total gamma$total ucl, method = "Student")
all plot <- rbind.data.frame(gamma plot, boot plot, studentt plot)
all plot wide <- all plot >>
 pivot_wider(values_from = c(lower95, upper95), names_from = method,
              id_cols = c(year, sim))
lb comp2 <- ggplot(all plot wide, aes(x = lower95 Gamma, y = lower95 Bootstrap, color = factor(sim))) +</pre>
 geom_point(size = .nafo_pts) +
geom_abline(slope = 1, linetype = 2, size = .nafo_lwd) +
  theme_nafo() +
  stat_regline_equation(aes(label = ..rr.label.., color = NULL), size = 3) +
  scale_color_brewer(palette = "Set1", name = "Simulation") +
  scale_x_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) +
scale_y_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) +
  labs(x = "Gamma lower 95% bound", y = "Bootstrapped lower 95% bound") +
  theme(legend.position = "none")
ub_comp2 <- ggplot(all_plot_wide, aes(x = upper95_Gamma, y = upper95_Bootstrap, color = factor(sim))) +</pre>
  geom_point(size = .nafo_pts) +
  geom_abline(slope = 1, linetype = 2, size = .nafo_lwd) +
  theme nafo() +
 stat_regline_equation(aes(label = ..rr.label.., color = NULL), size = 3) +
scale_color_brewer(palette = "Set1", name = "Simulation") +
  scale_x_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) +
  scale_y_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) +
  labs(x = "Gamma upper 95% bound", y = "Bootstrapped upper 95% bound") +
  theme(legend.position = "none")
```

lb_comp3 <- ggplot(all_plot_wide, aes(x = lower95_Student, y = lower95_Bootstrap, color = factor(sim))) +</pre> geom_point(size = .nafo pts) + geom_abline(slope = 1, linetype = 2, size = .nafo_lwd) + theme_nafo() + stat_regline_equation(aes(label = ..rr.label.., color = NULL), size = 3) + scale_color_brewer(palette = "Set1", name = "Simulation") + scale_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) +
scale_y_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) + labs(x = "Student's t lower 95% bound", y = "Bootstrapped lower 95% bound") + theme(legend.position = "right", legend.box.background = element_blank()) ub_comp3 <- ggplot(all_plot_wide, aes(x = upper95_Student, y = upper95_Bootstrap, color = factor(sim))) +</pre> geom_point(size = .nafo_pts) +
geom_abline(slope = 1, linetype = 2, size = .nafo_lwd) + theme_nafo() + stat_regline_equation(aes(label = ..rr.label.., color = NULL), size = 3) +
scale_color_brewer(palette = "Set1", name = "Simulation") + scale_x_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) +
scale_y_continuous(labels = scales::label_number(suffix = "", scale = 1e-8)) + labs(x = "Student's t upper 95% bound", y = "Bootstrapped upper 95% bound") + theme(legend.position = "none")

all comp2 <- (lb comp2 | lb comp3) / (ub comp2 | ub comp3)

Colophon

This version of the document was generated on 2024-11-18 20:26:37.853865 using the R markdown template for SCR documents from **NAFOdown**.

The computational environment that was used to generate this version is as follows:

```
#> - Session info -
#>
   setting value
#>
   version R version 4.3.2 (2023-10-31 ucrt)
#>
   OS
           Windows 10 x64 (build 19045)
#>
   system x86 64, mingw32
#>
            RTerm
   ui
#>
   language (EN)
#>
   collate English United States.utf8
#>
           English United States.utf8
   ctype
#>
             America/St Johns
   tz
#>
   date
             2024-11-18
             3.1.1 @ C:/Program Files/RStudio/resources/app/bin/quarto
#>
   pandoc
/bin/tools/ (via rmarkdown)
#>
#> - Packages -
#>
                      * version
   package
                                   date (UTC) lib source
#>
   abind
                        1.4-5
                                   2016-07-21 [1] CRAN (R 4.3.0)
#>
                        1.2.1
                                   2024-10-04 [1] CRAN (R 4.3.2)
   askpass
#> backports
                        1.4.1
                                   2021-12-13 [1] CRAN (R 4.3.0)
#> bookdown
                        0.35
                                   2023-08-09 [1] CRAN (R 4.3.1)
#>
   broom
                        1.0.5
                                   2023-06-09 [1] CRAN (R 4.3.1)
#>
   cachem
                       1.0.8
                                   2023-05-01 [1] CRAN (R 4.3.1)
#>
                        3.1-2
                                   2023-03-30 [1] CRAN (R 4.3.1)
   car
#>
                        3.0-5
   carData
                                   2022-01-06 [1] CRAN (R 4.3.1)
#>
   cli
                        3.6.1
                                   2023-03-23 [1] CRAN (R 4.3.1)
#>
   colorspace
                        2.1-0
                                   2023-01-23 [1] CRAN (R 4.3.1)
#>
                       1.5.3
                                   2024-06-20 [1] CRAN (R 4.3.2)
   crayon
#>
   crul
                        1.4.0
                                   2023-05-17 [1] CRAN (R 4.3.1)
#>
   curl
                        5.0.1
                                   2023-06-07 [1] CRAN (R 4.3.1)
                       1.14.8
#>
                                   2023-02-17 [1] CRAN (R 4.3.1)
   data.table
#>
   devtools
                        2.4.5
                                   2022-10-11 [1] CRAN (R 4.3.1)
#>
   digest
                        0.6.33
                                   2023-07-07 [1] CRAN (R 4.3.1)
#>
   dplyr
                        1.1.2
                                   2023-04-20 [1] CRAN (R 4.3.1)
#>
   ellipsis
                        0.3.2
                                   2021-04-29 [1] CRAN (R 4.3.1)
#>
   evaluate
                        1.0.1
                                   2024-10-10 [1] CRAN (R 4.3.2)
#> fansi
                        1.0.4
                                   2023-01-22 [1] CRAN (R 4.3.1)
#>
   farver
                        2.1.1
                                   2022-07-06 [1] CRAN (R 4.3.1)
#>
                        1.1.1
                                   2023-02-24 [1] CRAN (R 4.3.1)
   fastmap
                      * 0.9.4
#>
   flextable
                                   2023-10-22 [1] CRAN (R 4.3.2)
#>
   fontBitstreamVera
                        0.1.1
                                   2017-02-01 [1] CRAN (R 4.3.0)
#> fontLiberation
                        0.1.0
                                   2016-10-15 [1] CRAN (R 4.3.0)
#>
                        0.2.1
   fontquiver
                                   2017-02-01 [1] CRAN (R 4.3.1)
#> fs
                        1.6.3
                                   2023-07-20 [1] CRAN (R 4.3.1)
```

Northwest Atlantic Fisheries Orga	inization
-----------------------------------	-----------

#>	gdtools	0.3.3	2023-03-27	[1]	CRAN (R 4.3.1)				
#>	generics	0.1.3	2022-07-05	[1]	CRAN (R 4.3.1)				
#>	gfonts	0.2.0	2023-01-08	[1]	CRAN (R 4.3.1)				
#>	ggplot2 *	3.5.1	2024-04-23	[1]	CRAN (R 4.3.3)				
#>	ggpubr	0.6.0	2023-02-10	[1]	CRAN (R 4.3.3)				
#>	ggridges	0.5.6	2024-01-23	[1]	CRAN (R 4.3.3)				
#>	ggsignif	0.6.4	2022-10-13	[1]	CRAN (R 4.3.3)				
#>	ggthemes	5.0.0	2023-11-21	[1]	CRAN (R 4.3.2)				
#>	glue	1.6.2	2022-02-24	[1]	CRAN (R 4.3.1)				
#>	gtable	0.3.3	2023-03-21	[1]	CRAN (R 4.3.1)				
#>	here *	1.0.1	2020-12-13	[1]	CRAN (R 4.3.1)				
#>	highr	0.11	2024-05-26	[1]	CRAN (R 4.3.2)				
#>	htmltools	0.5.5	2023-03-23	[1]	CRAN (R 4.3.1)				
#>	htmlwidgets	1.6.2	2023-03-17	[1]	CRAN (R 4.3.1)				
#>	httpcode	0.3.0	2020-04-10	[1]	CRAN (R 4.3.1)				
#>	httpuv	1.6.11	2023-05-11	[1]	CRAN (R 4.3.1)				
#>	jsonlite	1.8.7	2023-06-29	[1]	CRAN (R 4.3.1)				
#>	knitr	1.43	2023-05-25	[1]	CRAN (R 4.3.1)				
#>	labeling	0.4.2	2020-10-20	[1]	CRAN (R 4.3.0)				
#>	later	1.3.1	2023-05-02	[1]	CRAN (R 4.3.1)				
#>	lifecycle	1.0.4	2023-11-07	[1]	CRAN (R 4.3.2)				
#>	magrittr	2.0.3	2022-03-30	[1]	CRAN (R 4.3.1)				
#>	memoise	2.0.1	2021-11-26	[1]	CRAN (R 4.3.1)				
#>	mime	0.12	2021-09-28	[1]	CRAN (R 4.3.0)				
#>	miniUI	0.1.1.1	2018-05-18	[1]	CRAN (R 4.3.1)				
#>	munsell	0.5.0	2018-06-12	[1]	CRAN (R 4.3.1)				
#>	NAFOdown *	0.0.1	2024-06-14	[1]	local				
#>	officedown	0.3.1	2023-09-02	[1]	CRAN (R 4.3.2)				
#>	officer	0.6.2	2023-03-28	[1]	CRAN (R 4.3.1)				
#>	openssl	2.1.0	2023-07-15	[1]	CRAN (R 4.3.1)				
#>	patchwork	1.2.0	2024-01-08	[1]	CRAN (R 4.3.3)				
#>	pillar	1.9.0	2023-03-22	[1]	CRAN (R 4.3.1)				
#>	pkgbuild	1.4.4	2024-03-17	[1]	CRAN (R 4.3.2)				
#>	pkgconfig	2.0.3	2019-09-22	[1]	CRAN (R 4.3.1)				
#>	pkgload	1.4.0	2024-06-28	[1]	CRAN (R 4.3.2)				
#>	polynom	1.4-1	2022-04-11	[1]	CRAN (R 4.3.3)				
#>	profvis	0.3.8	2023-05-02	[1]	CRAN (R 4.3.1)				
#>	promises	1.2.0.1	2021-02-11	[1]	CRAN (R 4.3.1)				
#>	purrr	1.0.2	2023-08-10	[1]	CRAN (R 4.3.3)				
#>	R6	2.5.1	2021-08-19	[1]	CRAN (R 4.3.1)				
#>	ragg	1.2.5	2023-01-12	[1]	CRAN (R 4.3.1)				
#>	RColorBrewer	1.1-3	2022-04-03	[1]	CRAN (R 4.3.0)				
#>	Rcpp	1.0.11	2023-07-06	[1]	CRAN (R 4.3.1)				
#>	remotes	2.4.2.1	2023-07-18	[1]	CRAN (R 4.3.2)				
#>	rlang	1.1.1	2023-04-28	[1]	CRAN (R 4.3.1)				
#>	rmarkdown	2.23	2023-07-01	[1]	CRAN (R 4.3.1)				
#>	rprojroot	2.0.4	2023-11-05	[1]	CRAN (R 4.3.2)				
#>	rstatix	0.7.2	2023-02-01	[1]	CRAN (R 4.3.3)				
#>	rstudioapi	0.17.1	2024-10-22	[1]	CRAN (R 4.3.2)				
#>	rvg	0.3.3	2023-05-10	[1]	CRAN (R 4.3.2)				
#>	scales	1.3.0.9000	2024-10-18	[1]	Github (r-lib/scales				
@c0f79d3)									

www.nafo.int

#>	sessioninfo	1.2.2	2021-12-06	[1]	CRAN	(R	4.3.1)		
#>	shiny	1.9.1	2024-08-01	[1]	CRAN	(R	4.3.2)		
#>	showtext	0.9-6	2023-05-03	[1]	CRAN	(R	4.3.1)		
#>	showtextdb	3.0	2020-06-04	[1]	CRAN	(R	4.3.1)		
#>	stringi	1.7.12	2023-01-11	[1]	CRAN	(R	4.3.0)		
#>	stringr	1.5.0	2022-12-02	[1]	CRAN	(R	4.3.1)		
#>	sysfonts	0.8.8	2022-03-13	[1]	CRAN	(R	4.3.1)		
#>	systemfonts	1.0.4	2022-02-11	[1]	CRAN	(R	4.3.1)		
#>	textshaping	0.3.6	2021-10-13	[1]	CRAN	(R	4.3.1)		
#>	tibble	3.2.1	2023-03-20	[1]	CRAN	(R	4.3.1)		
#>	tidyr	1.3.0	2023-01-24	[1]	CRAN	(R	4.3.1)		
#>	tidyselect	1.2.0	2022-10-10	[1]	CRAN	(R	4.3.1)		
#>	urlchecker	1.0.1	2021-11-30	[1]	CRAN	(R	4.3.1)		
#>	usethis	2.2.2	2023-07-06	[1]	CRAN	(R	4.3.1)		
#>	utf8	1.2.3	2023-01-31	[1]	CRAN	(R	4.3.1)		
#>	uuid	1.1-0	2022-04-19	[1]	CRAN	(R	4.3.0)		
#>	vctrs	0.6.3	2023-06-14	[1]	CRAN	(R	4.3.1)		
#>	withr	3.0.1	2024-07-31	[1]	CRAN	(R	4.3.2)		
#>	xfun	0.39	2023-04-20	[1]	CRAN	(R	4.3.1)		
#>	xml2	1.3.5	2023-07-06	[1]	CRAN	(R	4.3.1)		
#>	xtable	1.8-4	2019-04-21	[1]	CRAN	(R	4.3.1)		
#>	yaml	2.3.7	2023-01-23	[1]	CRAN	(R	4.3.0)		
#>	zip	2.3.0	2023-04-17	[1]	CRAN	(R	4.3.1)		
#>									
#>	[1] C:/Users/RegularP/AppData/Local/R/win-library/4.3								
#>	<pre>[2] C:/Program Files/R/R-4.3.2/library</pre>								
#>									
#> -									