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   
  
2022-07-29

# Abstract

# 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](#ref-kimura2006); [Pennington and Strømme, 1998](#ref-pennington1998)). 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 tend to be based on the survey indices themselves, together with any additional biological and ecological information that may be available. Therefore, when assessments are survey-based, the quantification of uncertainty of the survey indices becomes a primary source for informing risk.

Fisheries surveys are generally based on standard sampling techniques and indices are commonly derived using design-based estimators. In the Northwest Atlantic, major bottom trawl surveys follow a stratified-random sampling design with proportional allocation (e.g., [González-Troncoso et al., 2022](#ref-gonzalez2022); [Rideout et al., 2022](#ref-rideout2022)), and the biomass and abundance indices are calculated accordingly (e.g., [Smith and Somerton, 1981](#ref-smith1981)). The derivation of the standard mean and variance estimators from a stratified-random sampling design is constructed using the general properties of the mean and variance, building these estimators from the straightforward concept 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](#ref-cochran1977)). One important feature of this derivation is that it does not depend on any assumed distribution; the estimators of the mean and variance are valid irrespective of the underlying distribution. Approximate confidence intervals, in contrast, depend on an assumed distribution.

Standard confidence intervals are 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](#ref-cochran1977)). However, given both distributions permit negative values, this standard assumption may not be reasonable when analyzing strictly positive quantities like biomass and abundance ([Cadigan, 2011](#ref-cadigan2011)). The issue of negative lower bounds from this approximation is therefore an artifact of the assumed distribution rather than the stratified-random mean and variance estimators. 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 confidence intervals, depending on the case in hand, they could require additional computing power and programming efforts. These additional steps should not be a deterrent to implementing them when required but having a simpler alternative that addresses the issues linked to the normal approximation can also be useful. A cursory examination of biomass and abundance indices from research surveys not only shows that there are no negative values (naturally), but that the distributions are seldom symmetrical; long positive tails are often observed. Given these features, a logical choice for describing these data is the Gamma distribution. 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 simply changing the assumed distribution from Normal or Student’s t to Gamma we can provide a better approximation to the confidence intervals which does not suffer from undesired features (e.g. negative lower values), while better resembling observed distributions (e.g. asymmetrical distribution with longer positive tails).

# Methods

## Calculating Gamma parameters

Provided data from a bottom trawl stratified-random survey, average biomass or abundance () and sampling variance () over the stock area can be estimated using the standard random-stratified design calculations ([Cochran, 1977](#ref-cochran1977); [Smith, 1990](#ref-smith1990); [Smith and Somerton, 1981](#ref-smith1981)). These estimates can then be used to calculate the scale () and shape () parameters of the corresponding Gamma distribution as:

The resulting Gamma distribution is then 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

For some cases there might also be a need to calculate the probability that the current index is above or below an average level from a reference period . Since the reference point 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 we need to combine the variances across the individual indices. This combined variance can be obtained by simply applying the general properties of the variance. Since 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 provides a way for estimating . Furthermore, if we assume that the survey variances across years are independent, we can dismiss the covariances between years and and can be estimated as:

where and are the estimated annual mean and variance for the index in each year within the reference period , and where is the size of the period (i.e. the number of years used to compute the average). As above, and can be converted to and 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 under two use cases: 1) the calculation of confidence intervals from a single survey index, and 2) the estimation of the probability the stock to be above or below some reference point.

We explored these two use cases by simulating a redfish-like population using the R package SimSurvey ([Regular et al., 2020](#ref-regular2020)). 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 3O and 3LN. 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 use case 1, average trawlable abundance () and sampling variance () was calculated by year (1-20) and replicate (1-5) using standard design-based estimators [Canty and Ripley](#ref-canty2021) ([2021](#ref-canty2021)). Densities from these boostrap 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 use case 2, estimates from years 10-15 were treated as the reference period and the average abundance and the associated variance was 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-estimate of abundance across years 10-15 were bootstraped to obtain samples for comparison with density from the Gamma approach. Using distributions derived from the Gamma distribution and bootstrap samples, the probability 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](#app:appendix-a).

# Results and Discussion

The Gamma probability density distribution showed high variability among survey simulations, as did the bootstrap samples (Figure 1. Nevertheless, the shape of both the Gamma density and the bootstrap samples were similar across all years and survey replicates, indicating that the Gamma distribution provides a reasonable approximation of the uncertainty around the stratified estimates. The similarity holds when survey indices are aggregated (Figure 2). Though further quantitative analysis is required to assess the performance of these methods for calculating the confidence intervals, these results indicate that confidence intervals from the Gamma approach would be similar to those obtained using bootstrap samples. At the very least, confidence intervals from the Gamma distribution represents an improvement over the sometimes negative intervals derived from the t-distribution.

The BMSY proxy and associated limit reference point (30% BMSY proxy) proposed for redfish was accepted as an interim reference point as was the Gamma-based method for quantifying uncertainty. Since this proxy is based on survey indices, it is not considered perfectly known. Estimates were therefore aggregated to account for this uncertainty. Determining status relative to the LRP considering uncertainty in both the proxy-BMSY and the current Biomass level provides was considered the most fulsome formulation of uncertainty in stock status and the most precautionary approach to advice.

It is important to note, however, that survey indices can show unrealistic fluctuations year over year. This is apparent in both the simulations and across the refish indices. Large inter-annual fluctuations may be a consequence of sampling noise or distribution shifts. A single year may therefore be insufficient to indicate a true change in stock status.

# 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 spatially-correlated 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.

# Figures

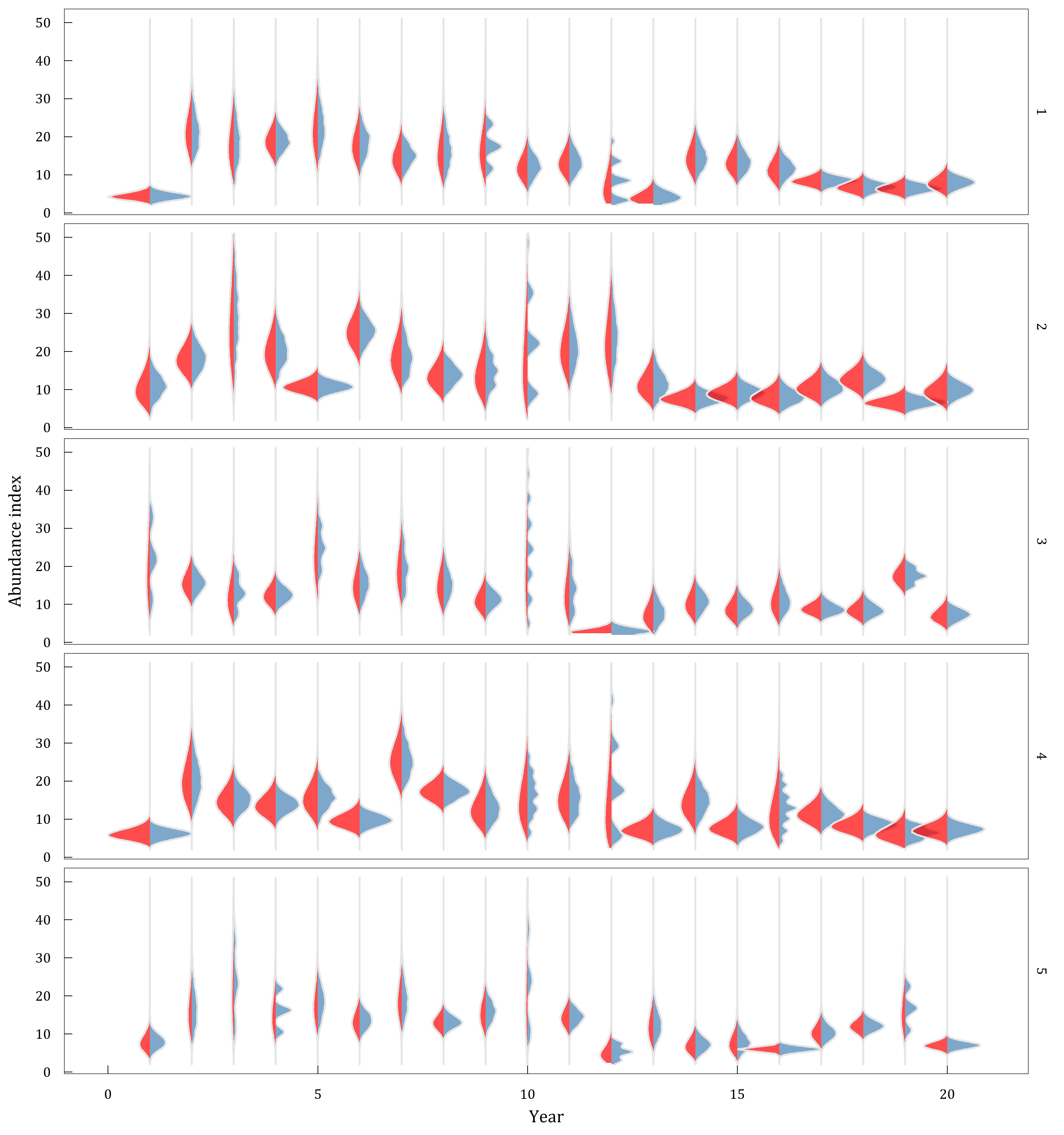


Fig 1: The bootstrap and gamma distributions estimated using simulated data from five independent surveys conducted over the same population across 20 years. The red area shows the density distribution from 1000 bootstrapped samples from each year and survey replicate. The blue area shows the gamma probability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

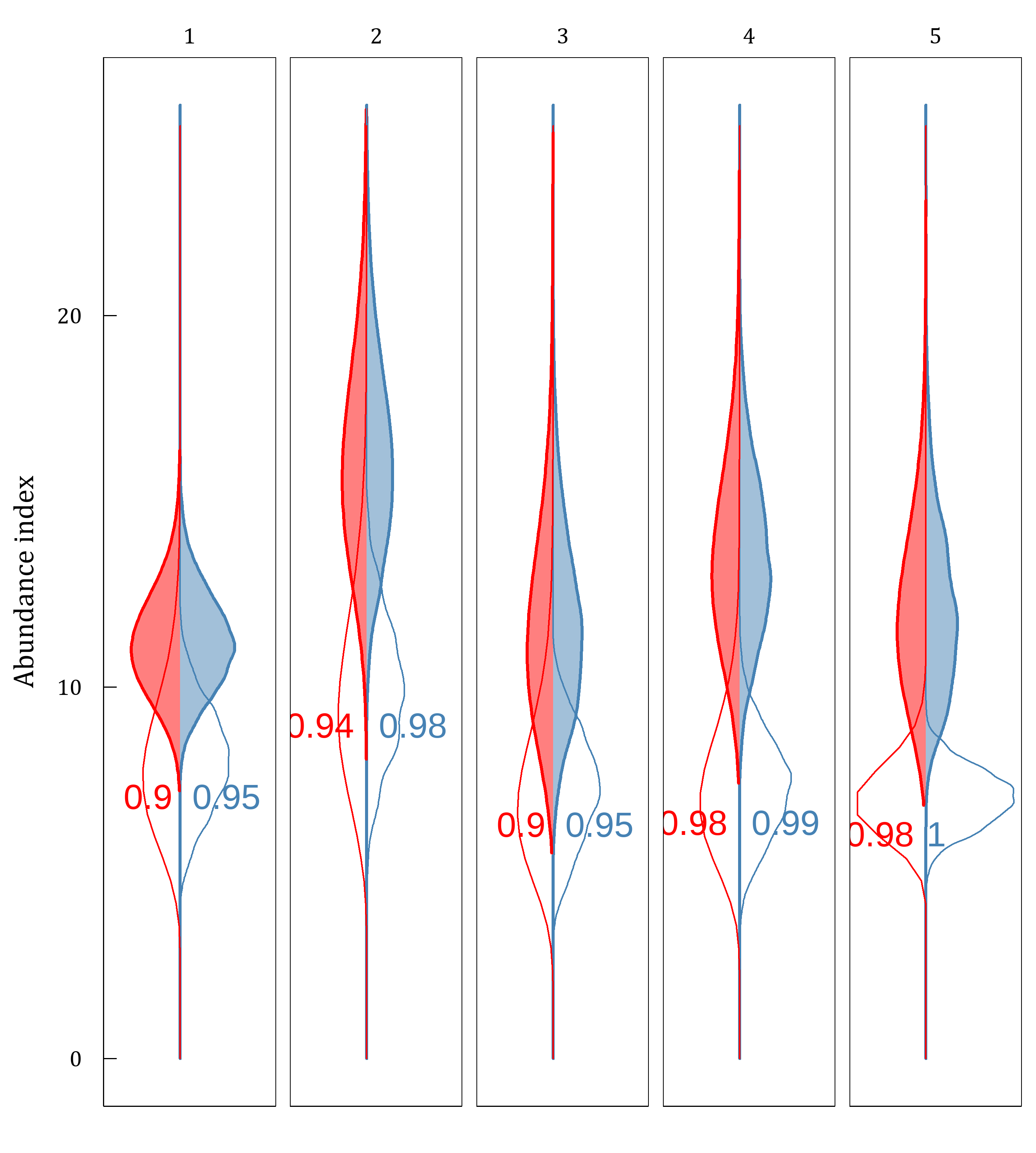


Fig 2: Bootstrap (blue) and gamma (red) distributions estimated from simulation 1 of a redfish-like population, where terminal estimates (year 20; open area) are compared to a reference period (aggregate estimates from years 2-9; shaded area). Densities for the reference period were obtained by combining 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.

# 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,  
 years = 1:20,  
 R = sim\_R(log\_mean = log(600000000),  
 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,  
 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$total) \* 2)  
x <- seq(rng[1], rng[2], length.out = 100)  
total\_strat\_den <- lapply(seq.int(nrow(total\_strat)), function(i) {  
 data.frame(sim = total\_strat$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))  
  
sumYst <- function(data, i = seq\_len(nrow(data)), return\_mean = FALSE) {  
 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) {  
 b <- boot::boot(data, statistic = sumYst, strata = data$strat, R = reps)  
 boot <- data.table(b$t) |> dplyr::rename(total = V1) |>  
 mutate(samp = seq.int(reps), sim = mean(data$sim), year = mean(data$year))  
 return(boot)  
}  
  
boot\_index <- furrr::future\_map\_dfr(split\_setdet, boot\_one\_year, reps = n\_boot,  
 .options = furrr::furrr\_options(seed = TRUE))  
  
quantile(boot\_index$total, prob = c(0.001, 0.999))  
  
den\_plot <- ggplot() +  
 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))  
  
ref\_boot\_fn <- function(data, R) {  
 b <- boot::boot(data, statistic = sumYst, strata = data$year\_strat, R = n\_boot, return\_mean = TRUE)  
 ref\_boot <- data.table(b$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 = TRUE))  
  
saveRDS(ref\_boot, file = "Gamma\_SCR/data/ref\_boot.rds")  
  
ref\_boot <- readRDS("Gamma\_SCR/data/ref\_boot.rds")  
  
### 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)  
  
### 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')  
  
n\_samp <- 100000  
  
ref\_samp <- map\_df(1:nrow(ref\_est),function(i){  
 dat <- rgamma(n\_samp, shape = ref\_est$shape[i], scale = ref\_est$scale[i])  
 data.table(sim=i, sample=dat)  
})  
  
t\_samp <- map\_df(1:nrow(t\_est),function(i){  
 dat <- rgamma(n\_samp, shape = t\_est$shape[i], scale = t\_est$scale[i])  
 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')  
  
### 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)  
  
ref\_plot <- ggplot() +  
 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(boot\_prob, 2)),  
 hjust = -0.2, vjust = 2, color = "steelblue") +  
 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"))  
total\_boot<- merge(boot\_ci, total\_strat, by = c("sim", "year"))  
  
gamma\_plot <- data.frame(year = total\_gamma$year, sim = total\_gamma$sim,  
 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,  
 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,  
 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))) +  
 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))) +  
 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))) +  
 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 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))) +  
 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 2022-07-29 16:47:31 using the R markdown template for SCR documents from [NAFOdown](https://github.com/nafc-assess/NAFOdown).

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

#> - Session info ---------------------------------------------------------------  
#> setting value  
#> version R version 4.1.2 (2021-11-01)  
#> os Windows 10 x64 (build 19042)  
#> system x86\_64, mingw32  
#> ui RTerm  
#> language (EN)  
#> collate English\_United States.1252  
#> ctype English\_United States.1252  
#> tz America/St\_Johns  
#> date 2022-07-29  
#> pandoc 2.14.0.3 @ C:/Program Files/RStudio/bin/pandoc/ (via rmarkdown)  
#>   
#> - Packages -------------------------------------------------------------------  
#> package \* version date (UTC) lib source  
#> assertthat 0.2.1 2019-03-21 [1] CRAN (R 4.1.2)  
#> base64enc 0.1-3 2015-07-28 [1] CRAN (R 4.1.0)  
#> bookdown 0.24 2021-09-02 [1] CRAN (R 4.1.1)  
#> cachem 1.0.6 2021-08-19 [1] CRAN (R 4.1.1)  
#> callr 3.7.0 2021-04-20 [1] CRAN (R 4.1.1)  
#> cli 3.1.0 2021-10-27 [1] CRAN (R 4.1.2)  
#> colorspace 2.0-2 2021-06-24 [1] CRAN (R 4.1.1)  
#> crayon 1.4.1 2021-02-08 [1] CRAN (R 4.1.1)  
#> data.table 1.14.0 2021-02-21 [1] CRAN (R 4.1.1)  
#> DBI 1.1.1 2021-01-15 [1] CRAN (R 4.1.1)  
#> desc 1.3.0 2021-03-05 [1] CRAN (R 4.1.1)  
#> devtools 2.4.3 2021-11-30 [1] CRAN (R 4.1.2)  
#> digest 0.6.29 2021-12-01 [1] CRAN (R 4.1.2)  
#> dplyr 1.0.7 2021-06-18 [1] CRAN (R 4.1.1)  
#> ellipsis 0.3.2 2021-04-29 [1] CRAN (R 4.1.1)  
#> evaluate 0.14 2019-05-28 [1] CRAN (R 4.1.1)  
#> fansi 0.5.0 2021-05-25 [1] CRAN (R 4.1.1)  
#> farver 2.1.0 2021-02-28 [1] CRAN (R 4.1.1)  
#> fastmap 1.1.0 2021-01-25 [1] CRAN (R 4.1.1)  
#> flextable \* 0.6.9 2021-10-07 [1] CRAN (R 4.1.0)  
#> fs 1.5.2 2021-12-08 [1] CRAN (R 4.1.2)  
#> gdtools 0.2.3 2021-01-06 [1] CRAN (R 4.1.1)  
#> generics 0.1.1 2021-10-25 [1] CRAN (R 4.1.2)  
#> ggplot2 \* 3.3.5 2021-06-25 [1] CRAN (R 4.1.1)  
#> ggridges 0.5.3 2021-01-08 [1] CRAN (R 4.1.3)  
#> ggthemes 4.2.4 2021-01-20 [1] CRAN (R 4.1.3)  
#> glue 1.4.2 2020-08-27 [1] CRAN (R 4.1.1)  
#> gtable 0.3.0 2019-03-25 [1] CRAN (R 4.1.1)  
#> here \* 1.0.1 2020-12-13 [1] CRAN (R 4.1.1)  
#> highr 0.9 2021-04-16 [1] CRAN (R 4.1.1)  
#> htmltools 0.5.2 2021-08-25 [1] CRAN (R 4.1.1)  
#> knitr 1.34 2021-09-09 [1] CRAN (R 4.1.1)  
#> labeling 0.4.2 2020-10-20 [1] CRAN (R 4.1.0)  
#> lifecycle 1.0.1 2021-09-24 [1] CRAN (R 4.1.3)  
#> magrittr 2.0.1 2020-11-17 [1] CRAN (R 4.1.1)  
#> memoise 2.0.1 2021-11-26 [1] CRAN (R 4.1.2)  
#> munsell 0.5.0 2018-06-12 [1] CRAN (R 4.1.1)  
#> NAFOdown \* 0.0.1.9000 2022-07-27 [1] local  
#> officer 0.4.0 2021-09-06 [1] CRAN (R 4.1.1)  
#> pillar 1.6.2 2021-07-29 [1] CRAN (R 4.1.1)  
#> pkgbuild 1.2.0 2020-12-15 [1] CRAN (R 4.1.1)  
#> pkgconfig 2.0.3 2019-09-22 [1] CRAN (R 4.1.1)  
#> pkgload 1.2.2 2021-09-11 [1] CRAN (R 4.1.0)  
#> plyr 1.8.6 2020-03-03 [1] CRAN (R 4.1.1)  
#> prettyunits 1.1.1 2020-01-24 [1] CRAN (R 4.1.1)  
#> processx 3.5.2 2021-04-30 [1] CRAN (R 4.1.1)  
#> ps 1.6.0 2021-02-28 [1] CRAN (R 4.1.1)  
#> purrr 0.3.4 2020-04-17 [1] CRAN (R 4.1.1)  
#> R6 2.5.1 2021-08-19 [1] CRAN (R 4.1.1)  
#> Rcpp 1.0.7 2021-07-07 [1] CRAN (R 4.1.1)  
#> remotes 2.4.0 2021-06-02 [1] CRAN (R 4.1.1)  
#> rlang 1.0.2 2022-03-04 [1] CRAN (R 4.1.3)  
#> rmarkdown 2.11 2021-09-14 [1] CRAN (R 4.1.1)  
#> rprojroot 2.0.2 2020-11-15 [1] CRAN (R 4.1.1)  
#> rstudioapi 0.13 2020-11-12 [1] CRAN (R 4.1.1)  
#> scales 1.1.1 2020-05-11 [1] CRAN (R 4.1.1)  
#> sessioninfo 1.2.2 2021-12-06 [1] CRAN (R 4.1.2)  
#> showtext 0.9-4 2021-08-14 [1] CRAN (R 4.1.1)  
#> showtextdb 3.0 2020-06-04 [1] CRAN (R 4.1.1)  
#> stringi 1.7.4 2021-08-25 [1] CRAN (R 4.1.1)  
#> stringr 1.4.0 2019-02-10 [1] CRAN (R 4.1.1)  
#> sysfonts 0.8.5 2021-08-09 [1] CRAN (R 4.1.1)  
#> systemfonts 1.0.3 2021-10-13 [1] CRAN (R 4.1.2)  
#> testthat 3.1.1 2021-12-03 [1] CRAN (R 4.1.2)  
#> tibble 3.1.4 2021-08-25 [1] CRAN (R 4.1.1)  
#> tidyselect 1.1.1 2021-04-30 [1] CRAN (R 4.1.1)  
#> usethis 2.0.1 2021-02-10 [1] CRAN (R 4.1.1)  
#> utf8 1.2.2 2021-07-24 [1] CRAN (R 4.1.1)  
#> uuid 0.1-4 2020-02-26 [1] CRAN (R 4.1.1)  
#> vctrs 0.3.8 2021-04-29 [1] CRAN (R 4.1.1)  
#> withr 2.4.3 2021-11-30 [1] CRAN (R 4.1.2)  
#> xfun 0.26 2021-09-14 [1] CRAN (R 4.1.0)  
#> xml2 1.3.2 2020-04-23 [1] CRAN (R 4.1.1)  
#> yaml 2.2.1 2020-02-01 [1] CRAN (R 4.1.0)  
#> zip 2.2.0 2021-05-31 [1] CRAN (R 4.1.1)  
#>   
#> [1] C:/Users/RegularP/Documents/R/win-library/4.1  
#> [2] C:/Program Files/R/R-4.1.2/library  
#>   
#> ------------------------------------------------------------------------------