Model averaging statistical distributions and the inversion principle

David Fox1,2,✉, Rebecca Fisher3,5, and Joe Thorley

June 10, 2025

Abstract

Text of abstract

1 The University of Melbourne, Australia  
2 Environmetrics Australia, Australia  
3 Australian Institute of Marine Science, Australia  
4 Oceans Institute, The University of Western Australia, Australia  
5 iormc

✉ Correspondence: [David Fox <david.fox@environmetrics.net.au>](mailto:david.fox@environmetrics.net.au)

Keywords: model averaging; AICc weights 2; statistical distribution 3

Highlights: This paper descibed how typical arithmetic averaging of statistical distributions fails to satisfy the inversion princple and provides an alternative method for obtaining model averaged values from the extremes of statistical distributions.

# Introduction

Statistical distributions are widely used across all fields of science to provide probabilistic estimates that are critical for scientific inference, and more importantly in a range of decision science frameworks. For example, the species sensitivity distribution (SSD) is used in the Australian (Warne et al., 2018) and Canadian water quality frameworks to determine concentrations that are protection of most species within an ecosystem. In this case regulators are typically interested in the lower tails of the statistical distribution, such as the 1th, 5th and 10th percentile, which would represent concentrations protective of 99, 95 and 90% of the ecosystem. Another example is in maximum streamflow estimations sometimes used in risk management of hydraulic structures, which may be more interested in the extreme upper tails of a statistical distribution, that represent rare events, such a 1 in 100 year, 1 in 200 year, or 1 in 500 year frequency (Bento et al., 2023).

In many situations, the appropriate statistical distribution to use is not clear, and there is no theoretically correct distribution. Uncertainty in the appropriate theoretical distribution has lead to an increasing popularity in the use of model averaging (Burnham and Anderson, 2002). Typically, model averaging is carried out by calculating an arithmetic average of the value of interest, usually weighted according to the relative AICc weights of input distributions Thorley and Schwarz (2018).

The weighted arithmetic mean is conventionally used for averaging model parameters or estimates (Burnham and Anderson, 2002). However, in the case of and values, the estimator fails to satisfy the *inversion principle* (Fox et al., 2024) which requires

This inconsistency has been rectified in ssdtools v2 (Thorley et al., 2025) by estimating the model-averaged (denoted ) directly from the model-averaged cumulative distribution function (*cdf*)

where is the *cdf* for the the *ith* model and is the model weight as before. is then obtained as the solution to

or, equivalently

for the proportion affected .

Here we demonstrate the potential error that may be introduced by the failure of arithmetic model averaging to satisfy the inversion principle, and make a call for the scientific community to lean on the technological advances that ensure model averaging of extreme estimates from statistical distributions is done correctly.

# Methods

## Case study 1 - Species sensitivity distribution (SSD) modelling

We used the example datasets in the ssddata package in R (Fisher & Thorley 2021) to examine the range of differences in estimated model averaged HCx values when calculated using a weighted arithmetic mean (default method for ssdtools versions 0 and 1) relative to the values obtained by estimating the model-averaged HC directly from the model-averaged cumulative distribution function (see equation [Equation 1](#eq-inv-princ) above).

To run the case studies, data were first extracted from ssddata (Fisher and Thorley, 2021) using the function get\_ssddata. These were then fit using ssdtools version 2.0 (Thorley et al., 2025), via the ssd\_fit\_dists function, using the lognormal, loglogistic, log Gumbel, Weibull, gamma and lognormal-lognormal mixture distributions. While there are other distributions available in ssdtools, this set represents the recommended set of stable distributions (Fox et al. 2024).

The estimated model averaged concentrations that protect 99, 95 and 80% of the population (PC99, PC95, PC80) were estimated from the resulting fits using both a weighted average, as well as the model-averaged cumulative distribution function. We then used the estimated hazard concentrations to estimate the percentage of species actually protected for each methods.

## Case study 2 - Maximum streamflow frequency estimation

Data on end of river streamflow rates across a range of catchments on the Great Barrier Reef (GBR) in Queensland were download from the publicly-available QLD discharge data at https://water-monitoring.information.qld.gov.au/. We downloaded the variable “stream discharge (megalitres/day)” for the full dataset. We did not use the provided daily data as these include modeled interpolation data for which there was limited information, and we wanted to ensure the data used represented real, maximum stream flow discharge rates.

We selected a subset of the most relevant gauges for the GBR, which were the lowest gauges in the catchment. All have some impact on coastal communities or roads (e.g., when they flood we have parts of the Bruce Highway cut for days/weeks which impacts transport of goods incl food into NQ and FNQ).

# Results

## Case study 1

We found that there is substantial error when converting between the HCx and the HPx, when HCx estimates are obtained using a geometric averaging method, but there is no error when HCx estimates are obtained using model-averaged cumulative distribution function ([Figure 1](#fig-pc-plot)). Substantial error for the averaging method occurs across most of the case studies examined for the PC99 (1% effected), and always resulted in a greater number of species being effected when compared to the intended target percentage ([Figure 1](#fig-pc-plot)). From some datasets (e.g. ccme\_glyphosate) the percentage of species actually effected using the averaging method was up to 3 times the target value ([Figure 1](#fig-pc-plot)). In general the error in the estimated number of species effected declines as the target percentage increases ([Figure 1](#fig-pc-plot)). For some datasets the number of species effected is actually less than the target percentage at higher percentage values (e.g. aims\_molybdenum\_marine, 5-10%).

hc\_dat |>  
 ggplot(aes(x=target\_p, y=ratio, color = multi\_est)) +  
 geom\_line() +  
 scale\_y\_log10() +  
 xlab("target % effected") +  
 ylab("ratio (estimated % effected / target % effected)") +  
 facet\_wrap(~dataset, ncol = 4)

|  |
| --- |
| Figure 1: The ratio of estimated percentage of species effected against the target percentage of species effected, plotted against the target percentage for the arithmetic (multi\_est = FALSE) and the multi (multi\_est = TRUE) averaging methods. |

## Case study 2

ggarrange(plotlist = fits\_dat$autoplot\_f,   
 common.legend = TRUE, ncol = 2, nrow = 4,  
 labels = fits\_dat$station\_id)

|  |
| --- |
| Figure 2: Fitted cumulative distributions for six statistical distributions, across seven example river flow datasets. |

We found there could be very large deviations from the intended original target frequency when the back calculated estimated proportions were converted into the corresponding estimated frequencies ([Figure 3](#fig-frequency-plot)). In most cases, the estimated frequencies were higher than the target frequency, meaning that events were predicted to be more common than they actually were ([Figure 3](#fig-frequency-plot)). For one river, the estimated frequency was as high as 5x the original target, particularly when considering extreme rare events (for example, a one in 500 year event).

freq\_ratio\_plot <- hc\_dat |>  
 ggplot(aes(x=target\_frequency, y=ratio, color = multi\_est)) +  
 geom\_line() +  
 scale\_y\_log10() +  
 xlab("target frequency") +  
 ylab("ratio (estimated frequency / target frequency)") +  
 facet\_wrap(~station\_id)  
freq\_ratio\_plot

|  |
| --- |
| Figure 3: The ratio of estimated frequency against the target frequency, plotted against the target frequency for the arithmetic (multi\_est = FALSE) and the multi (multi\_est = TRUE) averaging methods. |

Here is an example of inline code 3.14 in the middle of a sentence.

# Discussion

# Conclusion

# Acknowledgements

# References

Bento, A.M., Gomes, Pêgo, Viseu, Couto, L. and, 2023. Improved assessment of maximum streamflow for risk management of hydraulic infrastructures. A case study. International Journal of River Basin Management 21, 489–499. <https://doi.org/10.1080/15715124.2021.2016783>

Burnham, K., Anderson, D., 2002. Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer New York, New York, NY. <https://doi.org/10.1007/b97636>

Fisher, R., Thorley, J., 2021. ssddata: Species Sensitivity Distribution Data. <https://doi.org/10.32614/cran.package.ssddata>

Fox, D., Fisher, R., Thorley, J., 2024. Final report of the joint investigation into SSD modelling and ssdtools implementation for the derivation of toxicant guidelines values in Australia and New Zealand. Report prepared for the department of climate change, energy, the environment and water. Environmetrics Australia; Australian Institute of Marine Science. <https://doi.org/10.25845/xtvt-yc51>

Thorley, J., Fisher, R., Fox, D., Schwarz, C., 2025. Ssdtools v2: An R package to fit Species Sensitivity Distributions. Journal of Open Source Software 10, 7492. <https://doi.org/10.21105/joss.07492>

Thorley, J., Schwarz, C., 2018. Ssdtools: An R package to fit Species Sensitivity Distributions. Journal of Open Source Software 3, 1082. <https://doi.org/10.21105/joss.01082>

Warne, M., Batley, G.E., Dam, R.V., Chapman, J., Fox, D.R., Hickey, C.W., Stauber, J.L., 2018. Revised method for deriving Australian and New Zealand water quality guideline values for toxicants. <https://doi.org/10.13140/RG.2.2.36577.35686>

### Colophon

This report was generated on 2025-06-10 11:54:44.779099 using the following computational environment and dependencies:

# which R packages and versions?  
if ("devtools" %in% installed.packages()) devtools::session\_info()

─ Session info ───────────────────────────────────────────────────────────────  
 setting value  
 version R version 4.4.1 (2024-06-14)  
 os Debian GNU/Linux 13 (trixie)  
 system x86\_64, linux-gnu  
 ui X11  
 language (EN)  
 collate C  
 ctype en\_US.UTF-8  
 tz Australia/Perth  
 date 2025-06-10  
 pandoc 3.1.11.1 @ /usr/bin/ (via rmarkdown)  
 quarto 1.4.555 @ /usr/lib/rstudio-server/bin/quarto/bin/quarto  
  
─ Packages ───────────────────────────────────────────────────────────────────  
 package \* version date (UTC) lib source  
 abind 1.4-8 2024-09-12 [3] CRAN (R 4.4.1)  
 backports 1.5.0 2024-05-23 [3] CRAN (R 4.4.1)  
 broom 1.0.7 2024-09-26 [3] CRAN (R 4.4.1)  
 cachem 1.1.0 2024-05-16 [3] CRAN (R 4.4.1)  
 car 3.1-3 2024-09-27 [3] CRAN (R 4.4.1)  
 carData 3.0-5 2022-01-06 [3] CRAN (R 4.1.2)  
 chk 0.10.0 2025-01-24 [1] CRAN (R 4.4.1)  
 cli 3.6.5 2025-04-23 [1] CRAN (R 4.4.1)  
 codetools 0.2-20 2024-03-31 [4] CRAN (R 4.3.3)  
 commonmark 1.9.5 2025-03-17 [1] CRAN (R 4.4.1)  
 cowplot 1.1.3 2024-01-22 [3] CRAN (R 4.3.2)  
 devtools 2.4.5 2022-10-11 [3] CRAN (R 4.2.1)  
 dichromat 2.0-0.1 2022-05-02 [3] CRAN (R 4.2.0)  
 digest 0.6.37 2024-08-19 [3] CRAN (R 4.4.1)  
 dplyr \* 1.1.4 2023-11-17 [3] CRAN (R 4.4.2)  
 ellipsis 0.3.2 2021-04-29 [3] CRAN (R 4.1.1)  
 evaluate 1.0.3 2025-01-10 [3] CRAN (R 4.4.2)  
 farver 2.1.2 2024-05-13 [3] CRAN (R 4.4.1)  
 fastmap 1.2.0 2024-05-15 [3] CRAN (R 4.4.1)  
 Formula 1.2-5 2023-02-24 [3] CRAN (R 4.2.2)  
 fs 1.6.6 2025-04-12 [1] CRAN (R 4.4.1)  
 furrr 0.3.1 2022-08-15 [3] CRAN (R 4.2.1)  
 future 1.58.0 2025-06-05 [1] CRAN (R 4.4.1)  
 generics 0.1.4 2025-05-09 [1] CRAN (R 4.4.1)  
 ggplot2 \* 3.5.2 2025-04-09 [1] CRAN (R 4.4.1)  
 ggpubr \* 0.6.0 2023-02-10 [1] CRAN (R 4.4.1)  
 ggsignif 0.6.4 2022-10-13 [3] CRAN (R 4.2.1)  
 ggtext 0.1.2 2022-09-16 [1] CRAN (R 4.4.1)  
 globals 0.18.0 2025-05-08 [1] CRAN (R 4.4.1)  
 glue 1.8.0 2024-09-30 [1] CRAN (R 4.4.1)  
 goftest 1.2-3 2021-10-07 [1] CRAN (R 4.4.1)  
 gridExtra 2.3 2017-09-09 [3] CRAN (R 4.0.0)  
 gridtext 0.1.5 2022-09-16 [1] CRAN (R 4.4.1)  
 gtable 0.3.6 2024-10-25 [1] CRAN (R 4.4.1)  
 htmltools 0.5.8.1 2024-04-04 [3] CRAN (R 4.4.0)  
 htmlwidgets 1.6.4 2023-12-06 [3] CRAN (R 4.3.2)  
 httpuv 1.6.16 2025-04-16 [1] CRAN (R 4.4.1)  
 jsonlite 2.0.0 2025-03-27 [1] CRAN (R 4.4.1)  
 knitr 1.50 2025-03-16 [3] CRAN (R 4.4.3)  
 labeling 0.4.3 2023-08-29 [3] CRAN (R 4.3.1)  
 later 1.4.2 2025-04-08 [1] CRAN (R 4.4.1)  
 lattice 0.22-7 2025-04-02 [4] CRAN (R 4.4.3)  
 lifecycle 1.0.4 2023-11-07 [3] CRAN (R 4.3.2)  
 listenv 0.9.1 2024-01-29 [3] CRAN (R 4.3.2)  
 litedown 0.7 2025-04-08 [1] CRAN (R 4.4.1)  
 magrittr 2.0.3 2022-03-30 [3] CRAN (R 4.2.0)  
 markdown 2.0 2025-03-23 [1] CRAN (R 4.4.1)  
 Matrix 1.7-3 2025-03-11 [4] CRAN (R 4.4.3)  
 memoise 2.0.1 2021-11-26 [3] CRAN (R 4.1.2)  
 mime 0.13 2025-03-17 [1] CRAN (R 4.4.1)  
 miniUI 0.1.2 2025-04-17 [1] CRAN (R 4.4.1)  
 parallelly 1.45.0 2025-06-02 [1] CRAN (R 4.4.1)  
 pillar 1.10.2 2025-04-05 [1] CRAN (R 4.4.1)  
 pkgbuild 1.4.7 2025-03-24 [1] CRAN (R 4.4.1)  
 pkgconfig 2.0.3 2019-09-22 [3] CRAN (R 4.0.0)  
 pkgload 1.4.0 2024-06-28 [3] CRAN (R 4.4.1)  
 plyr 1.8.9 2023-10-02 [3] CRAN (R 4.3.1)  
 profvis 0.4.0 2024-09-20 [3] CRAN (R 4.4.1)  
 promises 1.3.2 2024-11-28 [3] CRAN (R 4.4.2)  
 purrr \* 1.0.4 2025-02-05 [1] CRAN (R 4.4.1)  
 R6 2.6.1 2025-02-15 [1] CRAN (R 4.4.1)  
 rbibutils 2.3 2024-10-04 [3] CRAN (R 4.4.1)  
 RColorBrewer 1.1-3 2022-04-03 [3] CRAN (R 4.2.0)  
 Rcpp 1.0.14 2025-01-12 [1] CRAN (R 4.4.1)  
 Rdpack 2.6.4 2025-04-09 [1] CRAN (R 4.4.1)  
 remotes 2.5.0 2024-03-17 [3] CRAN (R 4.4.1)  
 rlang 1.1.6 2025-04-11 [1] CRAN (R 4.4.1)  
 rmarkdown 2.29 2024-11-04 [3] CRAN (R 4.4.2)  
 rstatix 0.7.2 2023-02-01 [3] CRAN (R 4.2.2)  
 rstudioapi 0.17.1 2024-10-22 [3] CRAN (R 4.4.1)  
 scales 1.4.0 2025-04-24 [1] CRAN (R 4.4.1)  
 sessioninfo 1.2.3 2025-02-05 [3] CRAN (R 4.4.2)  
 shiny 1.10.0 2024-12-14 [3] CRAN (R 4.4.2)  
 ssddata \* 1.0.0 2021-11-05 [1] CRAN (R 4.4.1)  
 ssdinversionrr \* 0.0.0.9000 2025-05-26 [1] local  
 ssdtools \* 2.3.0.9004 2025-06-05 [1] Github (poissonconsulting/ssdtools@1e1ae89)  
 stringi 1.8.7 2025-03-27 [1] CRAN (R 4.4.1)  
 stringr 1.5.1 2023-11-14 [3] CRAN (R 4.3.2)  
 tibble 3.2.1 2023-03-20 [3] CRAN (R 4.4.2)  
 tidyr \* 1.3.1 2024-01-24 [3] CRAN (R 4.3.2)  
 tidyselect 1.2.1 2024-03-11 [3] CRAN (R 4.4.0)  
 TMB 1.9.17 2025-03-10 [1] CRAN (R 4.4.1)  
 universals 0.0.5 2022-09-22 [1] CRAN (R 4.4.1)  
 urlchecker 1.0.1 2021-11-30 [3] CRAN (R 4.2.1)  
 usethis 3.1.0 2024-11-26 [3] CRAN (R 4.4.2)  
 vctrs 0.6.5 2023-12-01 [3] CRAN (R 4.3.2)  
 withr 3.0.2 2024-10-28 [3] CRAN (R 4.4.1)  
 xfun 0.52 2025-04-02 [1] CRAN (R 4.4.1)  
 xml2 1.3.8 2025-03-14 [1] CRAN (R 4.4.1)  
 xtable 1.8-4 2019-04-21 [3] CRAN (R 4.0.0)  
 yaml 2.3.10 2024-07-26 [3] CRAN (R 4.4.1)  
  
 [1] /home/rfisher/R/x86\_64-pc-linux-gnu-library/4.4  
 [2] /usr/local/lib/R/site-library  
 [3] /usr/lib/R/site-library  
 [4] /usr/lib/R/library  
 \* ── Packages attached to the search path.  
  
──────────────────────────────────────────────────────────────────────────────

The current Git commit details are:

# what commit is this file at?   
if ("git2r" %in% installed.packages() & git2r::in\_repository(path = ".")) git2r::repository(here::here())

Local: main /mnt/c/Rworking/ssdinversionrr  
Remote: main @ origin (git@github.com:poissonconsulting/ssdinversionrr.git)  
Head: [6a15568] 2025-05-29: Merge pull request #2 from beckyfisher/main