Comparing VAST and sdmTMB GOA indices

Contents

```
#remotes::install_github("pbs-assess/sdmTMB", dependencies = TRUE)
library(VAST)
library(sp)
library(sdmTMB)
library(dplyr)
library(ggplot2)
library(here)

species <- "Gadus_macrocephalus"

phase <- c("hindcast", "production")[1]
```

We will fit geostatistical spatiotemporal models with VAST and sdmTMB for the purposes of index standardization and compare the outputs given the same data. We will use data from the GOA AFSC GAP bottom trawl survey for the species specified above. The density units are kg/km^2.

```
dat_ll <- readRDS(here::here(paste0("data/GOA/", phase, "/dat_allspp.RDS"))) %>%
  filter(species == gsub("_", " ", species)) %>%
  select(year,
         lat = lat_dd,
         lon = lon_dd,
         catch_kg,
         effort = effort_km2,
         cpue_kg_km2) %>%
  mutate(vessel = "missing",
         pass = 0
  )
```

We begin by specifying the VAST model. To specify the mesh used to approximate the spatial process, which is used in the SPDE calculations, we use the k-means method in VAST. Rather than specifying the cutoff distance, meshes in VAST are typically generated by specifying only the number of knots, which we will later pass, along with other model settings to the function make_settings. We will use 750 knots, the same number in the mesh created in the existing production VAST index for this stock and region.

We will include a factor predictor that represents the mean estimate for each time slice. Settings used for index standardization are applied by specifying `purpose = "index2"`.

Unlike in sdmTMB, the fitting and predicting steps are all accomplished with the function `fit_model()` and thus we need to specify the prediction grid (referred to as the "extrapolation grid" in VAST). Here, X and Y are coordinates in UTM zone 5.

```
GOAgrid <- read.csv(here("extrapolation_grids", "GOAThorsonGrid_Less700m.csv"))
input_grid <- cbind(Lat=GOAgrid$Latitude,
```

```r
                    Lon=GOAgrid$Longitude,
                    Area_km2=GOAgrid$Shape_Area/1000000)

settings <- make_settings(
  n_x = 750, # number of vertices in the SPDE mesh
  Region = "user",
  purpose = "index2", # index of abundance with Gamma for positive catches
  fine_scale = TRUE, # use bilinear interpolation from the INLA 'A' matrix
  zone = NA, # detects automatically
  Options = c("Calculate_Range" = TRUE, "Calculate_effective_area" = TRUE,
              "treat_nonencounter_as_zero" = FALSE),
  ObsModel = c(2, 1), # conventional logit-linked delta-Gamma; (2,4) if there are years with 100% encou
  bias.correct = TRUE,
  use_anisotropy = TRUE,
  max_cells = Inf, # use all grid cells from the extrapolation grid, production model used 2000
  knot_method = "grid", # or "samples"
  strata.limits = data.frame(STRATA = as.factor('All_areas')) # customize to sp.
)
```

Next we will fit a GLMM (generalized linear mixed effects model).

```r
# create folder for saved output:
dir.create(paste0(here("species_specific_code", "GOA", species,
                       "index_comparison")), showWarnings = FALSE)

f <- here("species_specific_code", "GOA", species, "index_comparison",
          "VASTfit_catch_effort_offset.RDS")
if (!file.exists(f)) {
  fit <- fit_model(
    settings = settings,
    Lat_i = dat_ll[, "lat"],
    Lon_i = dat_ll[, "lon"],
    t_i = dat_ll[, "year"],
    b_i = dat_ll[, "catch_kg"],
    a_i = dat_ll[, "effort"],
    input_grid = input_grid,
    working_dir = paste0(here("species_specific_code", "GOA",
                              species, "index_comparison"), "/")
  )
  saveRDS(fit, file = f)
} else {
  fit <- readRDS(f)
  fit <- reload_model(fit)
}
#> Maximum absolute gradient of 4.33e-06: No evidence of non-convergence
```

We can look at parameter estimates. First we see estimates from the binomial component and second we see estimates from the positive Gamma component.

```r
fit$parameter_estimates$diagnostics
#>            Param starting_value     Lower         MLE     Upper final_gradient
#> 1     ln_H_input    0.008403635 -5.000000  0.00845355  5.000000  -2.496435e-07
#> 2     ln_H_input   -1.144901347 -5.000000 -1.14525926  5.000000   4.710798e-07
```

```
#> 3       beta1_ft    2.824030116      -Inf  2.82409087       Inf   2.906213e-08
#> 4       beta1_ft    2.830266341      -Inf  2.83032397       Inf   2.434782e-07
#> 5       beta1_ft    3.263730173      -Inf  3.26378496       Inf   1.307747e-07
#> 6       beta1_ft    3.200290054      -Inf  3.20034663       Inf   1.456757e-07
#> 7       beta1_ft    3.137780459      -Inf  3.13784111       Inf   1.398500e-08
#> 8       beta1_ft    3.258787444      -Inf  3.25884771       Inf   8.762631e-08
#> 9       beta1_ft    3.256309076      -Inf  3.25637647       Inf   1.999462e-07
#> 10      beta1_ft    3.173039798      -Inf  3.17310585       Inf   2.723304e-08
#> 11      beta1_ft    3.256522662      -Inf  3.25658986       Inf  -2.923244e-08
#> 12      beta1_ft    3.286790328      -Inf  3.28686174       Inf  -4.201972e-09
#> 13      beta1_ft    3.279359722      -Inf  3.27942148       Inf   1.549883e-08
#> 14      beta1_ft    3.222146094      -Inf  3.22219027       Inf   2.765643e-08
#> 15      beta1_ft    3.279064105      -Inf  3.27912577       Inf   3.400076e-09
#> 16      beta1_ft    3.381958166      -Inf  3.38199211       Inf  -1.187919e-08
#> 17      beta1_ft    3.279958581      -Inf  3.28001474       Inf   1.169991e-08
#> 18      beta1_ft    3.325294856      -Inf  3.32534911       Inf   1.407698e-08
#> 19    L_omega1_z    0.559733453      -Inf  0.55978237       Inf  -1.281221e-06
#> 20  L_epsilon1_z    0.130793016      -Inf  0.13080226       Inf  -4.325529e-06
#> 21     logkappa1   -4.120992161 -6.775053 -4.12086070 -1.659693   3.470394e-07
#> 22      beta2_ft    4.314464782      -Inf  4.31458229       Inf   3.456163e-08
#> 23      beta2_ft    4.464550441      -Inf  4.46461953       Inf   1.668817e-08
#> 24      beta2_ft    4.114780400      -Inf  4.11483174       Inf  -2.409126e-08
#> 25      beta2_ft    3.696763737      -Inf  3.69681485       Inf  -1.208234e-08
#> 26      beta2_ft    4.070033058      -Inf  4.07013164       Inf   1.323062e-08
#> 27      beta2_ft    4.087917562      -Inf  4.08797712       Inf   2.066042e-08
#> 28      beta2_ft    4.196421637      -Inf  4.19654856       Inf   3.724431e-08
#> 29      beta2_ft    4.183359892      -Inf  4.18338178       Inf  -1.098944e-08
#> 30      beta2_ft    4.028655360      -Inf  4.02872438       Inf   7.045987e-09
#> 31      beta2_ft    4.139750628      -Inf  4.13978249       Inf   3.533330e-09
#> 32      beta2_ft    4.357596131      -Inf  4.35775711       Inf   5.461687e-08
#> 33      beta2_ft    4.282350309      -Inf  4.28246688       Inf   3.278202e-08
#> 34      beta2_ft    3.903810457      -Inf  3.90384336       Inf  -2.672066e-08
#> 35      beta2_ft    3.749249350      -Inf  3.74927758       Inf  -1.352287e-08
#> 36      beta2_ft    4.070699578      -Inf  4.07079692       Inf   2.579307e-08
#> 37      beta2_ft    4.105601535      -Inf  4.10568822       Inf   4.192493e-09
#> 38    L_omega2_z    1.278188078      -Inf  1.27816917       Inf  -4.795491e-07
#> 39  L_epsilon2_z    1.257356694      -Inf  1.25742551       Inf  -9.399974e-07
#> 40     logkappa2   -3.724984208 -6.775053 -3.72491664 -1.659693   8.202309e-07
#> 41     logSigmaM    0.541091406      -Inf  0.54108869 10.000000  -3.763412e-06
```

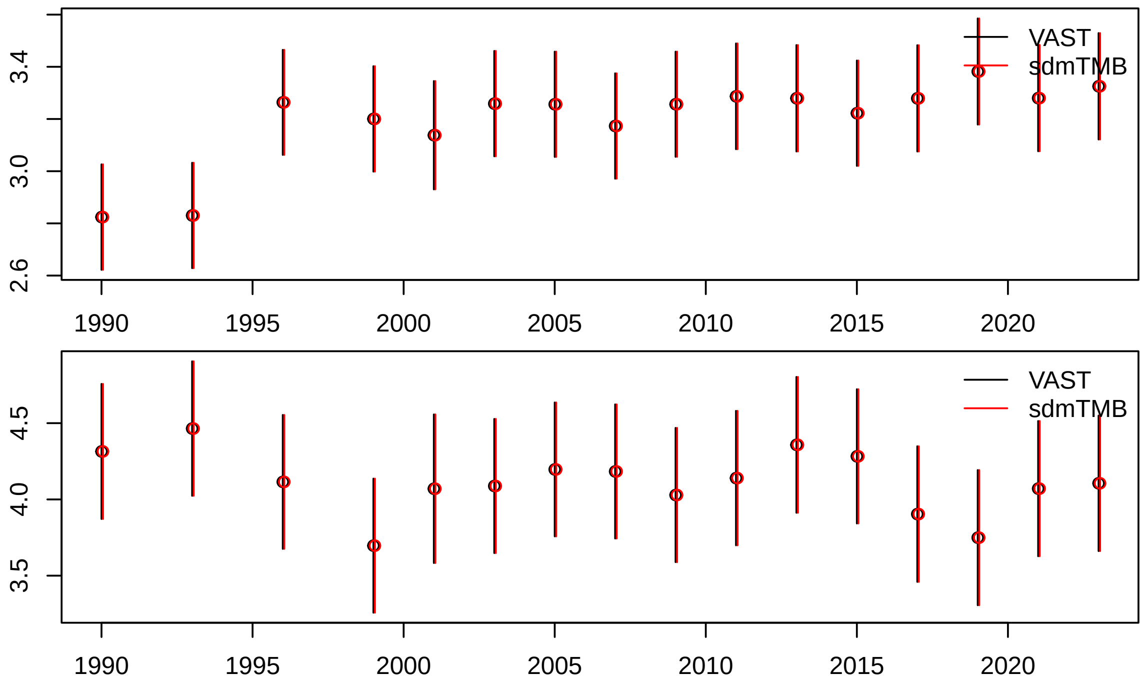
Now we fit the same model in sdmTMB:

```
dat <- dat_ll %>%
  rename(X = lon, Y = lat)

dat$year_f <- as.factor(dat$year)

coordinates(dat) <- ~ X + Y
proj4string(dat) <- CRS("+proj=longlat +datum=WGS84")
dat <- as.data.frame(spTransform(dat, CRS("+proj=utm +zone=5")))
# scale to km so values don't get too large
dat$X <- dat$coords.x1 / 1000
dat$Y <- dat$coords.x2 / 1000
```

```r
f1 <- here("species_specific_code", "GOA", species,
           "index_comparison", "fit_sdmTMB_catch_effort_offset.RDS")
if (!file.exists(f1)) {
# make mesh and fit model
mesh <-  make_mesh(dat, xy_cols = c("X", "Y"), mesh = fit$spatial_list$MeshList$anisotropic_mesh) #pass
#mesh <-  make_mesh(dat, xy_cols = c("X", "Y"), n_knots = 50, type = "kmeans") #coarser mesh for experi

fit_sdmTMB <- sdmTMB(
  catch_kg ~ 0 + year_f,
  data = dat,
  mesh = mesh,
  family = delta_gamma(type = "poisson-link"),
  time = "year",
  spatial = "on",
  spatiotemporal = "iid",
  offset = log(dat$effort),
  silent = FALSE,
  anisotropy = TRUE,
  do_fit = TRUE
  #, do_index = TRUE (to compute index at same time, requires passing args)
)
fit_sdmTMB
saveRDS(fit_sdmTMB, file = f1)
} else {
fit_sdmTMB <- readRDS(f1)
}

# diagnose estimation issues due to model structure
#TMBhelper::check_estimability(fit_sdmTMB$tmb_obj)
```

We wrote some custom code to extract comparable parameters (not shown above). Here are the annual mean estimates in link space with 95% confidence intervals for the two components to the delta model:

```r
par(mfrow = c(2, 1), cex = 0.8, mar = c(1.5, 1, 1, 1), oma = c(2, 3, 1, 1))
plot_betas(fit, fit_sdmTMB, "beta1_ft", sdmTMB_pars = 1)
plot_betas(fit, fit_sdmTMB, "beta2_ft", sdmTMB_pars = 2)
```



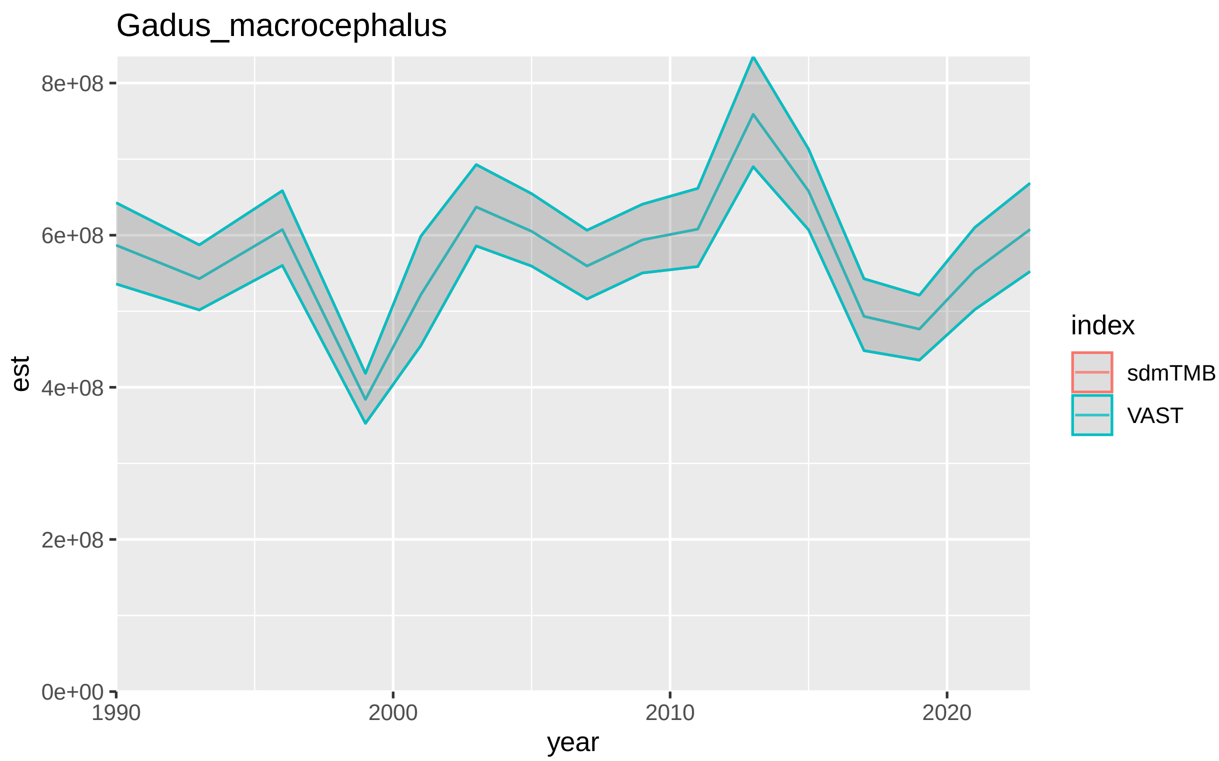
We can compare the index we would get using sdmTMB.

```r
# prep prediction grid and transform to UTM projection
grid_ll <- as.data.frame(input_grid)
names(grid_ll) <- tolower(names(grid_ll))
coordinates(grid_ll) <- ~ lon + lat
proj4string(grid_ll) <- CRS("+proj=longlat +datum=WGS84")
grid <- as.data.frame(spTransform(grid_ll, CRS("+proj=utm +zone=5")))

# rename and scale to km so values don't get too large
grid$X <- grid$coords.x1 / 1000
grid$Y <- grid$coords.x2 / 1000

# or with sf:
# grid_ll <- sf::st_as_sf(
#   x = grid_ll,
#   coords = c("lon", "lat"),
#   crs = "+proj=longlat +datum=WGS84"
# )
# grid <- sf::st_transform(grid_ll, crs = "+proj=utm +zone=5")

# replicate extrapolation grid for each year in data
pred_grid <- replicate_df(grid, "year_f", unique(dat$year_f))
pred_grid$year <- as.integer(as.character(factor(pred_grid$year_f)))

# make predictions and get index
f2 <- here("species_specific_code", "GOA", species,
           "index_comparison", "predictions_catch_effort_offset.RDS")
if (!file.exists(f2)) {
```

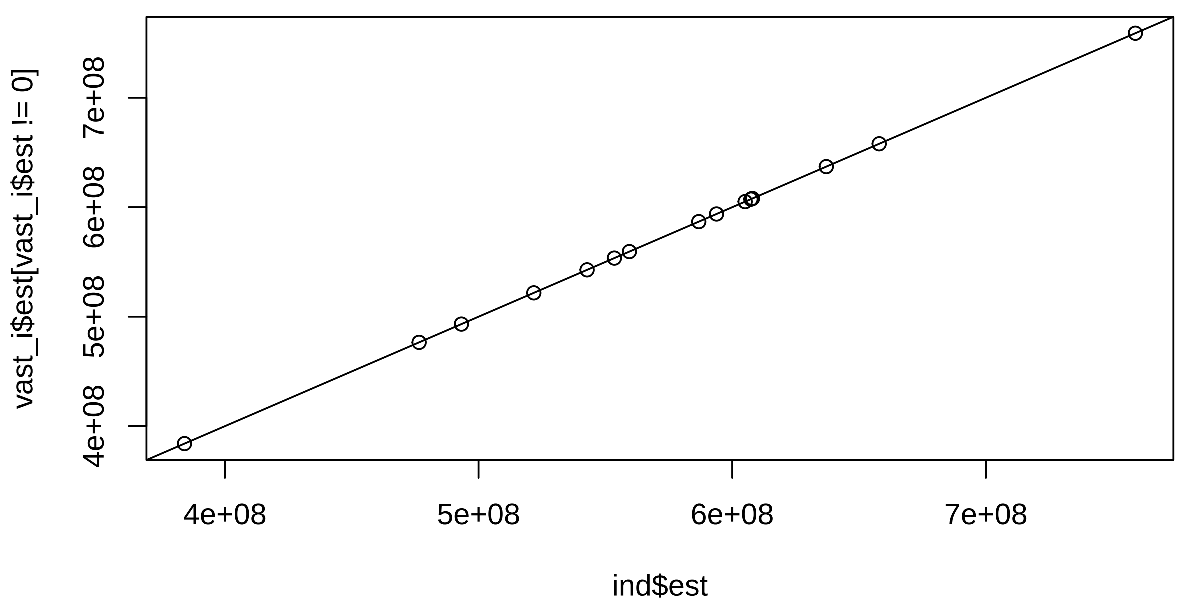
```
p <- predict(fit_sdmTMB, newdata = pred_grid, return_tmb_object = TRUE)
saveRDS(p, file = f2)
} else {
p <- readRDS(f2)
}

f3 <- here("species_specific_code", "GOA", species,
           "index_comparison", "index_catch_effort_offset.RDS")
if (!file.exists(f3)) {
ind <- get_index(p, bias_correct = TRUE, area = p$data$area_km2)
saveRDS(ind, file = f3)
} else {
ind <- readRDS(f3)
}
```

Now, we can compare the indices.

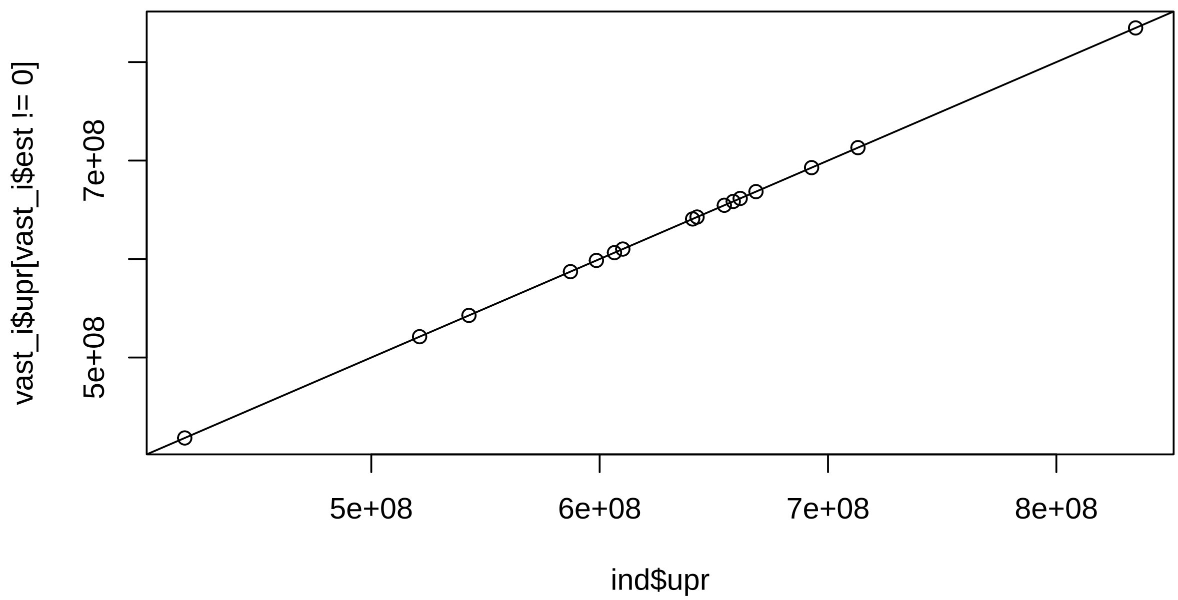
```
sdm_i <- ind %>% mutate(index = "sdmTMB")
vast_i <- read.csv(here("species_specific_code", "GOA", species,
                        "index_comparison", "Index_catch_effort_offset.csv")) %>%
  mutate(index = "VAST", year = as.numeric(Time), est = Estimate,
    se = Std..Error.for.ln.Estimate.) %>%
  select(index, year, est, se) %>%
  filter(year %in% unique(sdm_i$year)) %>%
  mutate(lwr = exp(log(est) + qnorm(0.025) * se)) %>%
  mutate(upr = exp(log(est) + qnorm(0.975) * se))
both_i <- bind_rows(sdm_i, vast_i) %>% filter(est > 0)

ggplot(both_i, aes(x = year, y = est, ymin = lwr, ymax = upr, colour = index)) +
  geom_ribbon(alpha = 0.1) +
  geom_line(alpha = 0.8) +
  ylim(0, max(both_i$upr)) +
  ggtitle(species) +
  coord_cartesian(expand = FALSE)
```



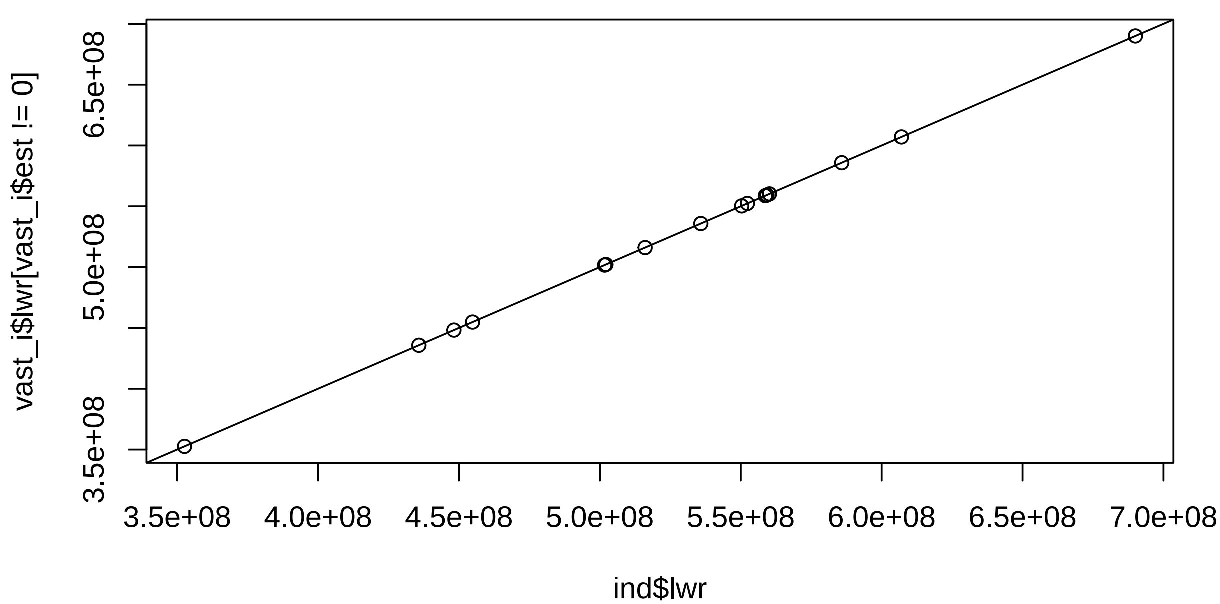

```r
plot(ind$est, vast_i$est[vast_i$est != 0]);abline(0, 1)
```




```r
plot(ind$upr, vast_i$upr[vast_i$est != 0]);abline(0, 1)
```




```
plot(ind$lwr, vast_i$lwr[vast_i$est != 0]);abline(0, 1)
```




```
(ind$est - vast_i$est[vast_i$est != 0]) / vast_i$est[vast_i$est != 0]
```

```
#>  [1] -8.842009e-10 -2.136041e-10 -4.394049e-10 -4.469164e-10 -2.821520e-09
#>  [6] -3.850075e-10 -1.216496e-09 -6.352668e-10 -2.457378e-11 -1.295155e-09
#> [11] -1.095235e-09 -1.136628e-09 -8.728911e-10 -7.739413e-10  1.455672e-09
#> [16] -2.138708e-10
(ind$upr - vast_i$upr[vast_i$est != 0]) / vast_i$upr[vast_i$est != 0]
#>  [1] -1.435716e-09 -1.853876e-10 -6.011724e-10 -2.281624e-09 -5.114089e-09
#>  [6] -8.682832e-10 -1.384656e-09 -6.331043e-10  6.727417e-11 -1.275886e-09
#> [11] -1.335717e-09 -1.198359e-09 -9.114630e-10 -7.634072e-10  3.918196e-09
#> [16] -4.455280e-10
(ind$lwr - vast_i$lwr[vast_i$est != 0]) / vast_i$lwr[vast_i$est != 0]
#>  [1] -3.326832e-10 -2.418260e-10 -2.776410e-10  1.387789e-09 -5.289493e-10
#>  [6]  9.826795e-11 -1.048342e-09 -6.374314e-10 -1.164226e-10 -1.314426e-09
#> [11] -8.547510e-10 -1.074895e-09 -8.343192e-10 -7.844747e-10 -1.006853e-09
#> [16]  1.778844e-11
```

This document was built using:

```
R.Version()$version.string
#> [1] "R version 4.3.0 (2023-04-21 ucrt)"
packageVersion("VAST")
#> [1] '3.11.2'
packageVersion("FishStatsUtils")
#> [1] '2.13.1'
```