Comparing VAST and sdmTMB GOA indices

Contents

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

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 pacific cod data from the GOA AFSC GAP bottom trawl survey. The density units are kg/km^2.

```r
dat_ll <- readRDS(here("species_specific_code", "GOA", "Gadus_macrocephalus",
                       "data", "Data_Geostat_Gadus_macrocephalus.rds"))

dat_ll <-  dplyr::transmute(dat_ll,
                        cpue_kg_km2 = Catch_KG,
                        year = as.integer(Year),
                        vessel = "missing",
                        effort = AreaSwept_km2,
                        lat = Lat,
                        lon = Lon,
                        pass = 0) %>%
               as.data.frame() # ensure not a tibble
```

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 3.

```r
GOAgrid <- read.csv(here("extrapolation_grids", "GOAThorsonGrid_Less700m.csv"))
input_grid <- cbind(Lat=GOAgrid$Latitude,
                    Lon=GOAgrid$Longitude,
                    Area_km2=GOAgrid$Shape_Area/1000000)
```

```r
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 = 3,
  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", "Gadus_macrocephalus", "index_comparison")), show

f <- here("species_specific_code", "GOA", "Gadus_macrocephalus", "index_comparison", "VASTfit.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[, "cpue_kg_km2"],
    a_i = dat_ll[, "effort"],
    input_grid = input_grid,
    working_dir = paste0(here("species_specific_code", "GOA",
                              "Gadus_macrocephalus", "index_comparison"), "/")
  )
  saveRDS(fit, file = f)
} else {
  fit <- readRDS(f)
}
```

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     0.48242841  -Inf  0.48243958   Inf   1.035286e-08
#> 2    ln_H_input     0.36549635  -Inf  0.36549402   Inf   5.622791e-09
#> 3      beta1_ft    -0.55240228  -Inf -0.55247295   Inf  -5.460258e-09
#> 4      beta1_ft    -0.53101452  -Inf -0.53110946   Inf   1.456918e-09
#> 5      beta1_ft    -0.73818420  -Inf -0.73826412   Inf  -1.337597e-09
#> 6      beta1_ft    -1.00062960  -Inf -1.00070730   Inf  -3.676835e-09
#> 7      beta1_ft    -1.51016910  -Inf -1.51023671   Inf  -5.058961e-09
#> 8      beta1_ft    -1.22327797  -Inf -1.22339176   Inf   4.441343e-09
#> 9      beta1_ft    -1.14028519  -Inf -1.14037007   Inf  -1.126162e-09
```

```
#> 10      beta1_ft    -1.13056902  -Inf -1.13068163   Inf   5.179952e-09
#> 11      beta1_ft    -0.72378420  -Inf -0.72387763   Inf  -1.385416e-10
#> 12      beta1_ft    -0.73144310  -Inf -0.73154977   Inf   3.380933e-09
#> 13      beta1_ft    -0.79529242  -Inf -0.79539104   Inf   1.561261e-09
#> 14      beta1_ft    -0.80073658  -Inf -0.80082836   Inf   1.030664e-09
#> 15      beta1_ft    -1.47051185  -Inf -1.47061109   Inf   1.960168e-09
#> 16      beta1_ft    -1.11217353  -Inf -1.11228315   Inf   4.138217e-09
#> 17      beta1_ft    -1.03036422  -Inf -1.03043810   Inf  -2.591729e-09
#> 18      beta1_ft    -0.83247085  -Inf -0.83254542   Inf  -1.788678e-09
#> 19    L_omega1_z     2.30397015  -Inf  2.30397270   Inf  -1.803375e-08
#> 20 L_epsilon1_z     0.33301289  -Inf  0.33301725   Inf  -7.780432e-08
#> 21     logkappa1    -3.96743218  -Inf -3.96743274   Inf   3.775915e-08
#> 22      beta2_ft     6.10704608  -Inf  6.10700599   Inf  -4.863345e-09
#> 23      beta2_ft     5.96824448  -Inf  5.96824273   Inf  -1.046423e-09
#> 24      beta2_ft     6.21112494  -Inf  6.21111560   Inf  -2.283308e-09
#> 25      beta2_ft     5.91243309  -Inf  5.91239776   Inf  -3.661853e-09
#> 26      beta2_ft     6.06038423  -Inf  6.06036328   Inf  -3.626646e-09
#> 27      beta2_ft     6.06801927  -Inf  6.06801762   Inf  -4.827854e-10
#> 28      beta2_ft     5.99669411  -Inf  5.99667626   Inf  -2.453007e-09
#> 29      beta2_ft     5.95693023  -Inf  5.95694311   Inf   4.446719e-10
#> 30      beta2_ft     6.15315994  -Inf  6.15314929   Inf  -1.268788e-09
#> 31      beta2_ft     6.05487537  -Inf  6.05487916   Inf   7.887522e-10
#> 32      beta2_ft     6.16071804  -Inf  6.16071724   Inf  -6.678675e-10
#> 33      beta2_ft     5.88194032  -Inf  5.88193447   Inf  -1.176375e-09
#> 34      beta2_ft     5.72384921  -Inf  5.72385281   Inf   3.547385e-11
#> 35      beta2_ft     5.72985376  -Inf  5.72986462   Inf   6.811689e-10
#> 36      beta2_ft     5.80003706  -Inf  5.80002497   Inf  -1.572495e-09
#> 37      beta2_ft     5.75483730  -Inf  5.75483677   Inf  -1.187551e-09
#> 38    L_omega2_z     0.83685770  -Inf  0.83685904   Inf  -4.576824e-08
#> 39 L_epsilon2_z     1.27832071  -Inf  1.27833418   Inf  -9.843492e-08
#> 40     logkappa2    -2.02115699  -Inf -2.02113532   Inf   6.799801e-08
#> 41     logSigmaM     0.04476912  -Inf  0.04476892   Inf  -1.310735e-07
```

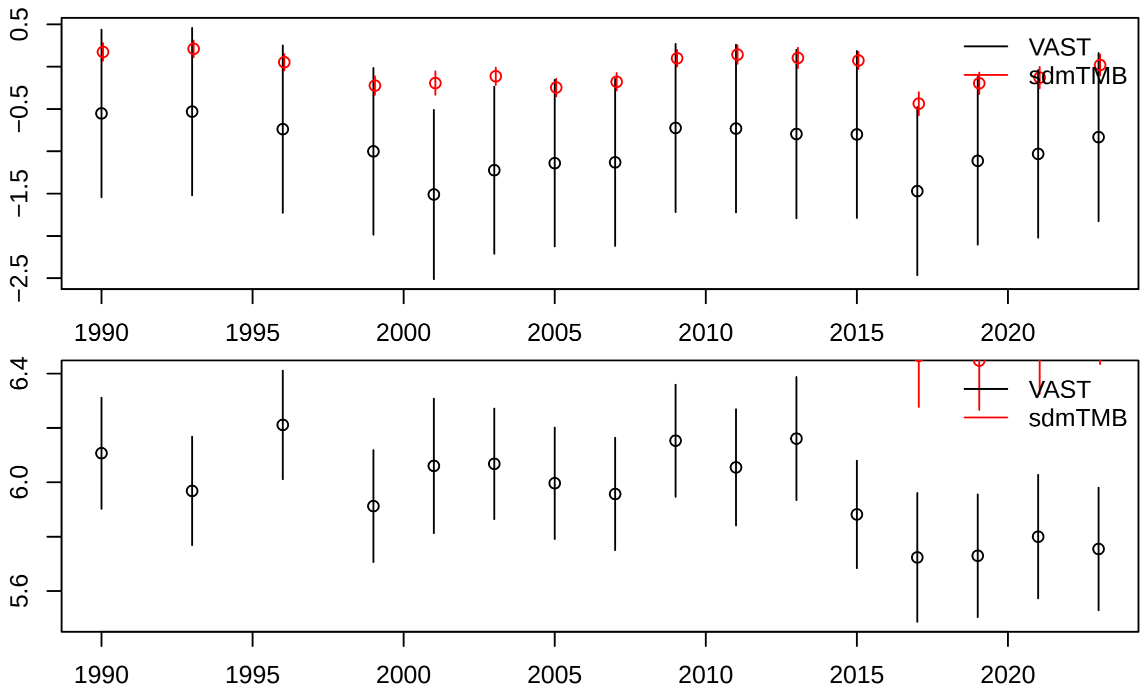
Now we fit the same model in sdmTMB:

```
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=3")))
# scale to km so values don't get too large
dat$X <- dat$coords.x1 / 1000
dat$Y <- dat$coords.x2 / 1000

f1 <- here("species_specific_code", "GOA", "Gadus_macrocephalus",
           "index_comparison", "fit_sdmTMB.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
```

```r
fit_sdmTMB <- sdmTMB(
  cpue_kg_km2 ~ 0 + year_f,
  data = dat,
  mesh = mesh,
  family = delta_gamma(type = "poisson-link"),
  time = "year",
  spatial = "on",
  spatiotemporal = "iid",
  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 = here("species_specific_code", "GOA",
                                "Gadus_macrocephalus", "index_comparison",
                                "fit_sdmTMB.RDS"))
} 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)
```



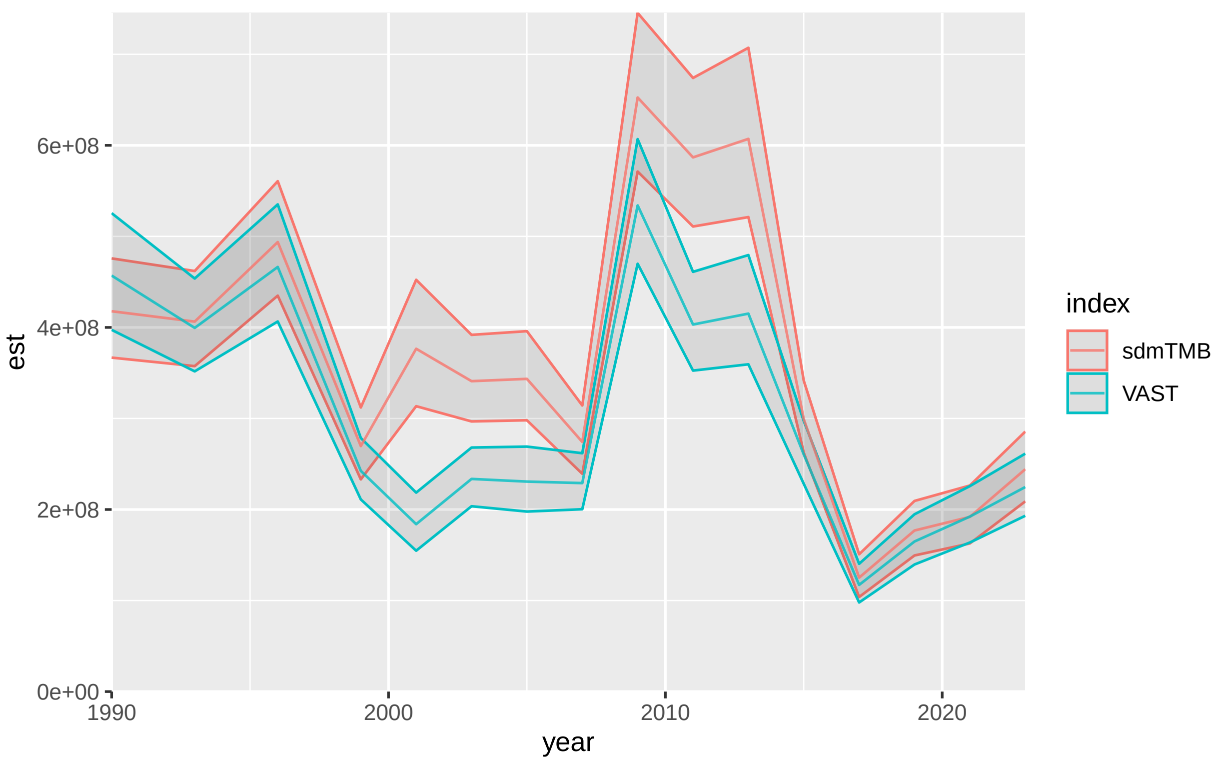
We can compare the index we would get using sdmTMB.

```
# 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=3")))

# 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=3")

# 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", "Gadus_macrocephalus",
           "index_comparison", "predictions.RDS")
if (!file.exists(f2)) {
```

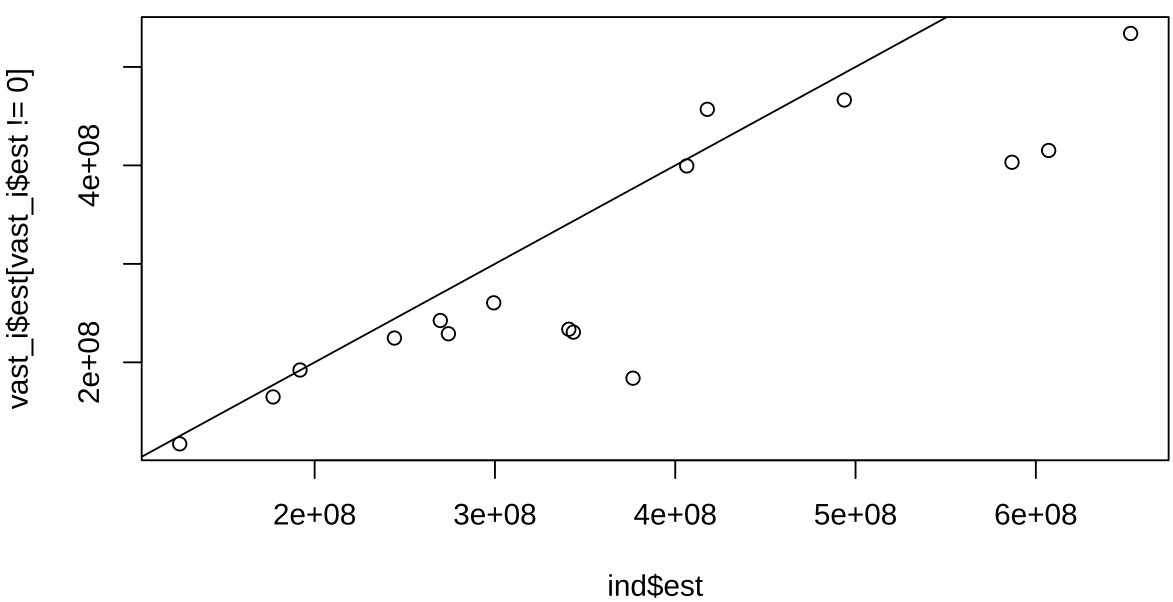
```r
p <- predict(fit_sdmTMB, newdata = pred_grid, return_tmb_object = TRUE)
saveRDS(p, file = here("species_specific_code", "GOA", "Gadus_macrocephalus", "index_comparison", "pred
} else {
p <- readRDS(f2)
}

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

Now, we can compare the indices.

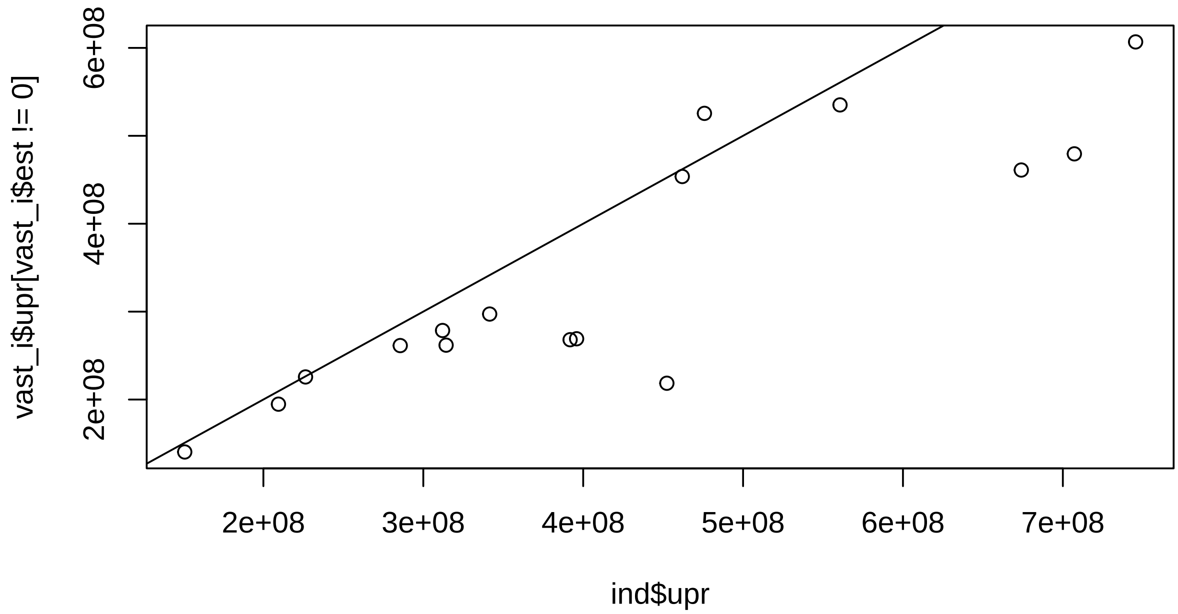
```r
vast_i <- read.csv(here("species_specific_code", "GOA", "Gadus_macrocephalus", "index_comparison", "Ind
  mutate(index = "VAST", year = as.numeric(Time), est = Estimate,
    se = Std..Error.for.ln.Estimate.) %>%
  select(index, year, est, se) %>%
  mutate(lwr = exp(log(est) + qnorm(0.025) * se)) %>%
  mutate(upr = exp(log(est) + qnorm(0.975) * se))
sdm_i <- ind %>% mutate(index = "sdmTMB")
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)) +
  coord_cartesian(expand = FALSE)
```



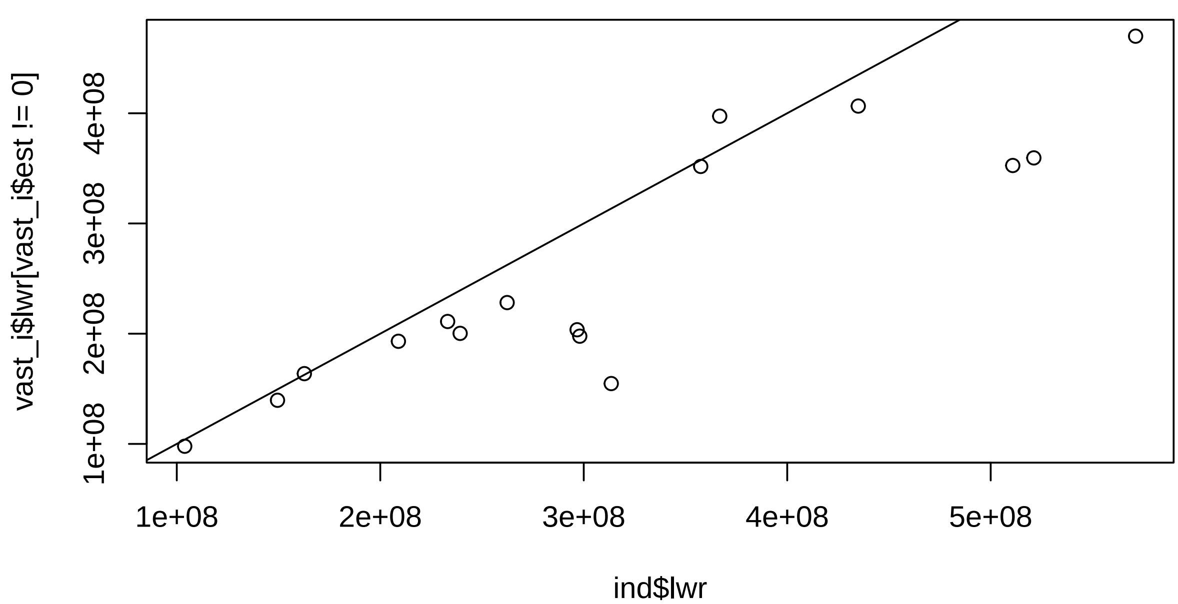

```
plot(ind$est, vast_i$est[vast_i$est != 0]);abline(0, 1)
```




```
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] -0.085734329  0.017272839  0.058654506  0.112355963  1.047740498
#>  [6]  0.459709390  0.489071685  0.197380848  0.222014407  0.455317083
#> [11]  0.462295665  0.149277170  0.068018618  0.073242851 -0.001760198
#> [16]  0.087218410
(ind$upr - vast_i$upr[vast_i$est != 0]) / vast_i$upr[vast_i$est != 0]
#>  [1] -0.094540207  0.018210021  0.047595785  0.120388830  1.069702237
#>  [6]  0.461805388  0.471098394  0.200241187  0.228529651  0.461869089
#> [11]  0.474932887  0.148961352  0.074550092  0.075410053  0.002873784
#> [16]  0.092665826
(ind$lwr - vast_i$lwr[vast_i$est != 0]) / vast_i$lwr[vast_i$est != 0]
#>  [1] -0.076842811  0.016336520  0.069829966  0.104380689  1.026011797
#>  [6]  0.457616396  0.507264567  0.194527326  0.215533716  0.448794443
#> [11]  0.449766718  0.149593075  0.061526844  0.071080016 -0.006372767
#> [16]  0.081798152
```

This document was built using:

```
R.Version()$version.string
#> [1] "R version 4.3.0 (2023-04-21 ucrt)"
packageVersion("VAST")
#> [1] '3.10.0'
packageVersion("FishStatsUtils")
#> [1] '2.12.0'
```