UW FISH 572 Survey Science

Model based combination of multiple surveys

Lewis Barnett & Eric Ward

Alaska Fisheries Science Center

February 2, 2026

Outline

- There are lots of approaches for combining data from multiple surveys
- These approaches are by no means exhaustive, but represent some of the more common situations
- Also lots of great existing teaching material w/examples:

2023 IMR sdmTMB workshop

2025 DFO sdmTMB workshop

Assumptions

- When surveys are combined, we assume the spatial parameters are shared (range, variance)
- Catchability and trends may differ by regions
- To create a robust index, we want to account for these major sources of variability

Caveats

- Ideally surveys wouldn't differ in selectivity or catchability, but they often do
- Ideally there would be experimental data to leverage when combining surveys with different gears or protocols, but there often isn't
- Best practice would be to combine surveys with more similar methods, but we explore a range of examples

Example 1: spatially adjacent surveys

- 2 bottom trawl surveys: Bering Sea and Aleutian Islands
- Focus on years when both conducted: 2004, 2006, 2010, 2012, 2014, 2016, 2018, 2022, 2024
- We'll use arrowtooth flounder for this case study

General approach for all models

- Mesh uses a cutoff distance of 25 km
- We use a delta - lognormal distribution to separately model occurrence and catch rates
- Spatial and spatiotemporal models included
- Catch weight as a response, with log(effort) as an offset

The data

- Access via 'surveyjoin' R package
- Extracted data included here
- Eastern Bering Sea index will be our focus, its grid also included

Modelling common trends across surveys

- What are the individual arguments for this model doing?

```
mesh <- make_mesh(arrowtooth,
                  xy_cols = c("X","Y"),
                  cutoff = 25)

fit <- sdmTMB(catch_weight ~ 0 + fyear,
        data = arrowtooth,
        offset = log(arrowtooth$effort),
        mesh = mesh,
        family = delta_lognormal(),
        spatial="on",
        spatiotemporal="iid",
        time = "year")
```

Generating the index

- Recall that after fitting, generating the index involves several steps

- The prediction grid needs to be replicated for each year, and it needs to include everything from our model

```
nd <- sdmTMB::replicate_df(ebs_grid, "year",
                   unique(arrowtooth$year))
nd <- sdmTMB::add_utm_columns(nd,
                      ll_names = c("lon","lat"),
                      utm_crs = 32602)
nd$fyear <- as.factor(nd$year)
```

Generating the index

- Next, we can predict to our grid

```
pred <- predict(fit, newdata = nd,
                return_tmb_object = TRUE)
```

- And finally, we can generate the index

```
index <- get_index(pred, bias_correct = TRUE)
```

Generating the index

```
head(index)
#>   year      est      lwr       upr  log_est         se
#> 1 2004 814284.1 651770.6 1017318.9 13.61006 0.11358198 
#> 2 2006 877002.2 684272.7 1124015.1 13.68426 0.12661097 
#> 3 2010 700414.6 554866.8  884141.3 13.45943 0.11885136 
#> 4 2012 440731.2 344434.0  563951.2 12.99619 0.12578428 
#> 5 2014 728811.5 584441.4  908844.2 13.49917 0.11263399 
#> 6 2016 893610.4 736185.3 1084699.3 13.70303 0.09887331 
```

Modelling different catchability by survey

- Add survey offset allowing catchability to differ
- If we only want to change a few things, we can use `update()`

```
fit2 <- update(fit,
        formula = catch_weight ~ 0 + fyear + survey_name)
```

Modelling survey specific trends

- The catchability estimate here is significantly positive
- What does this mean?

```
tidy(fit2)[,c("term","estimate","std.error")]
#> # A tibble: 10 × 3
#>    term                          estimate std.error
#>    <chr>                            <dbl>     <dbl>
#>  1 fyear2004                        -6.40      5.09
#>  2 fyear2006                        -7.83      5.10
#>  3 fyear2010                        -7.68      5.10
#>  4 fyear2012                        -8.45      5.11
#>  5 fyear2014                        -6.88      5.09
#>  6 fyear2016                        -4.24      5.09
#>  7 fyear2018                        -2.35      5.08
#>  8 fyear2022                        -5.91      5.09
#>  9 fyear2024                        -6.53      5.10
#> 10 survey_nameeastern Bering Sea    11.8       4.05
```

Updating the indices

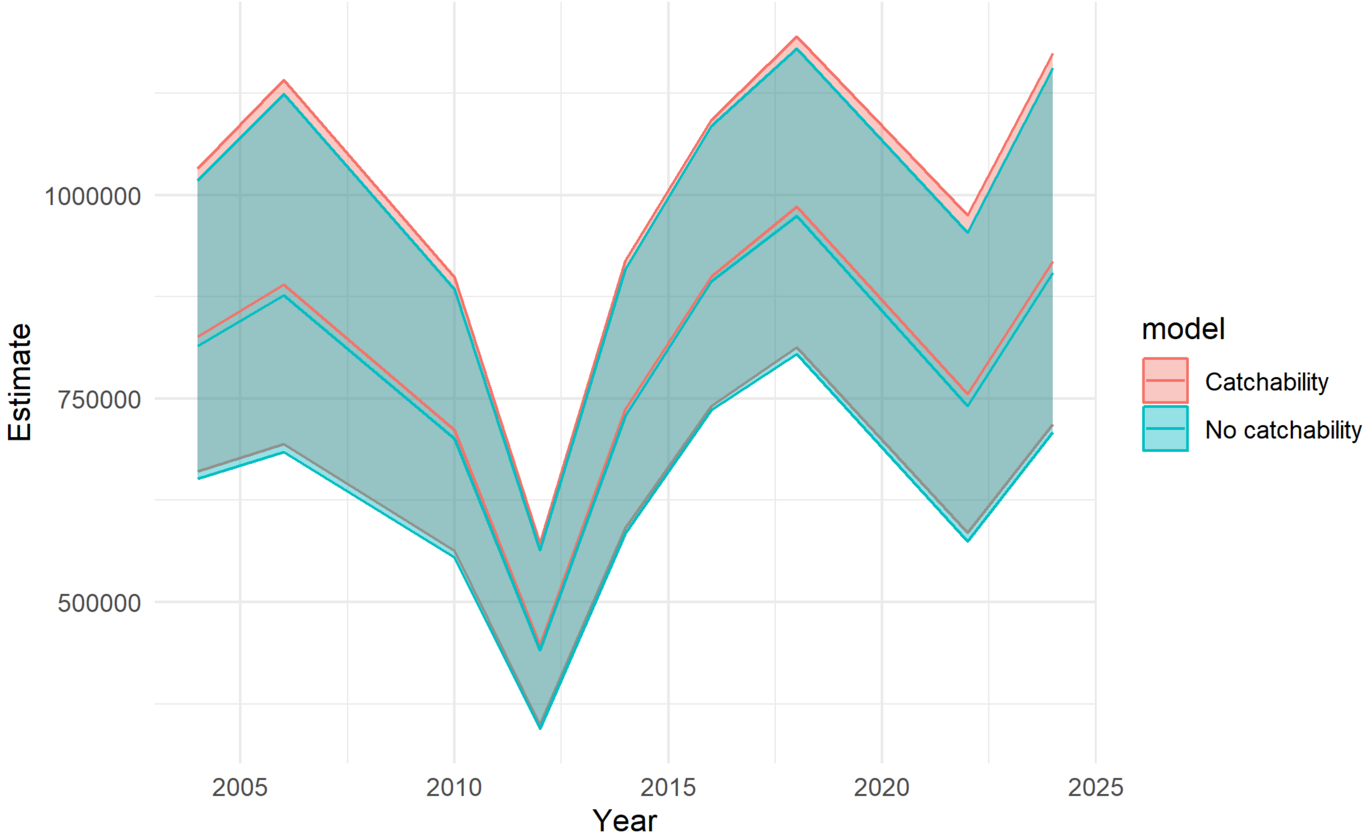
- We can turn the crank and generate the index from this new model (code hidden but is on slides)

- Because catchability just moves the intercept up/down we expect the indices to be perfectly correlated (and they are!)

```
cor(index$est, index2$est)
#> [1] 0.9998313
```

Visualizing the indices

- However the scale is also exactly the same - why??



Let's see what's going on with the spatial parameters

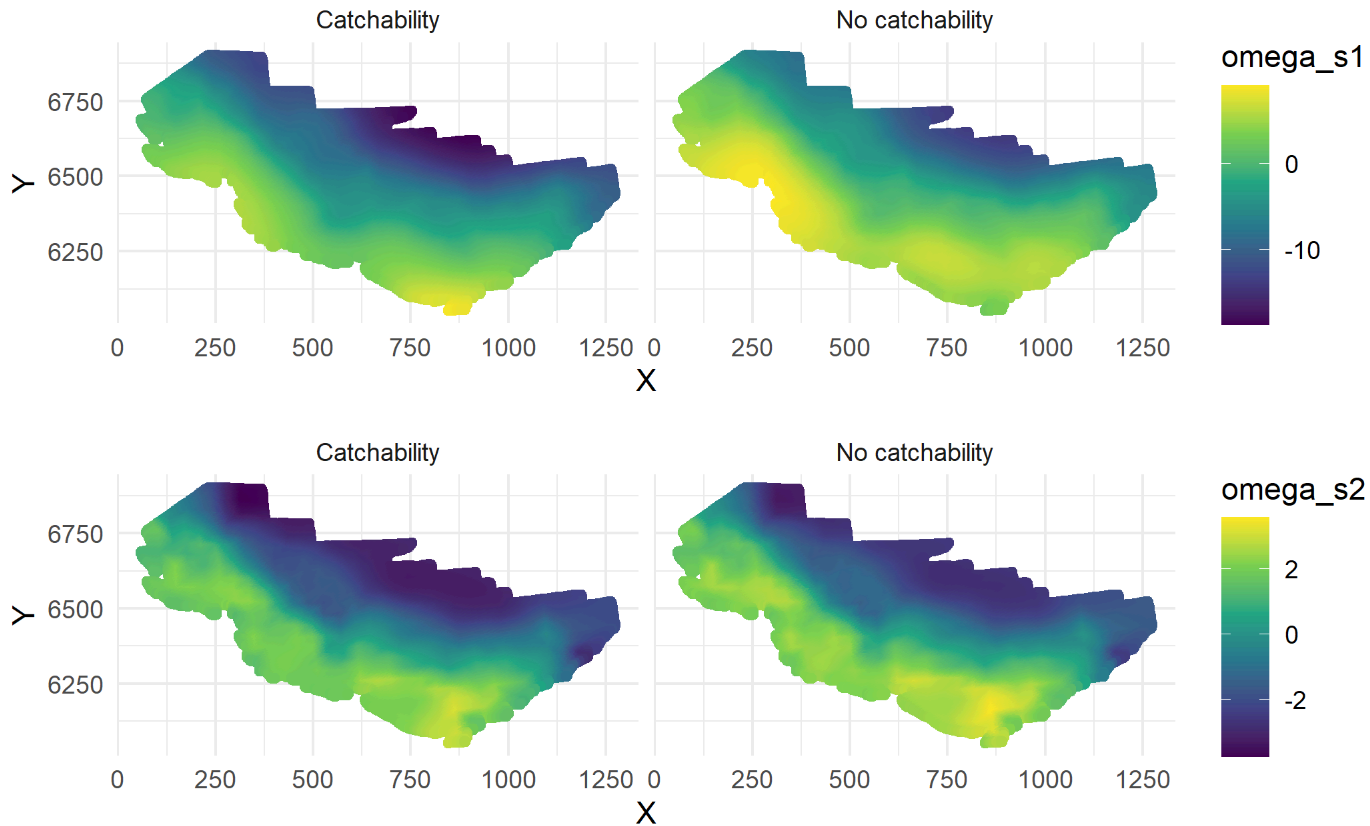
- These look relatively similar

```
#> # A tibble: 3 × 6
#>   model term    estimate std.error conf.low conf.high
#>   <dbl> <chr>      <dbl>     <dbl>    <dbl>     <dbl>
#> 1     1 range     667.      85.6     518.      857.  
#> 2     1 sigma_O     5.69     0.798     4.32      7.49
#> 3     1 sigma_E     1.73     0.185     1.40      2.14
```

```
#> # A tibble: 3 × 6
#>   model term    estimate std.error conf.low conf.high
#>   <dbl> <chr>      <dbl>     <dbl>    <dbl>     <dbl>
#> 1     1 range     729.      94.8     565.      940.  
#> 2     1 sigma_O     6.08     0.857     4.61      8.01
#> 3     1 sigma_E     1.78     0.199     1.43      2.21
```

Let's see what's going on with the spatial parameters

- Spatial fields' flexibility is able to account for catchability when it's not explicitly included



Modelling survey specific trends

- We can also let catchability/trends vary by survey

```
fit3 <- update(fit,
         formula = catch_weight ~ 0 + fyear * survey_name)
```

- On your own: inspect how the spatial parameters and fields differ from our previous approaches

Example 2: surveys with confounding in space-time

- In the previous example, spatially adjacent surveys were collected in the same year
- Enables us to separate out space/time effects
- This is not always the case -- Sean Anderson (DFO) has been working on this with 'checkerboard' survey designs

Illustrating the issue with the Alaska data

- We'll filter these datasets to be non-overlapping

Updating the model fit with the new data

- We'll keep the initial model, but here just swap in the new subset of data

```
mesh <- make_mesh(arrow2,
                  xy_cols = c("X","Y"),
                  cutoff = 25)

fit4 <- sdmTMB(catch_weight ~ 0 + fyear,
        data = arrow2,
        offset = log(arrow2$effort),
        mesh = mesh,
        family = delta_lognormal(),
        spatial="on",
        spatiotemporal="iid",
        time = "year")
```

Something is clearly wrong!

- Characteristic see-saw pattern (unrealistic)

```
head(index4)
#>   year        est         lwr       upr  log_est
#> 1 2004   44964.93    341.3263 5923497.6 10.71364 2.4902
#> 2 2006  916590.88 699763.2778 1200604.3 13.72842 0.1377
#> 3 2010   84558.80   1473.9729 4850964.5 11.34520 2.0661
#> 4 2012  495019.37 378513.1017  647386.3 13.11235 0.1369
#> 5 2014   75323.47    928.0492 6113496.7 11.22955 2.2431
#> 6 2016 1029639.42 825515.2435 1284237.1 13.84472 0.1127
```

Possible solutions

- If years are continuous (2003, 2004, 2005, ...)

    - Include smooth or AR(1) on year effects and spatiotemporal fields
    - `time_varying` formula
    - `time_varying_type` argument

- Can also turn spatiotemporal `off`

Example 3: overlapping surveys

- Case study data: Triennial and annual bottom trawl survey from USA west coast

- These surveys overlap spatially (slightly) and both occurred 2003-2004

```
remotes::install_github("pfmc-assessments/nwfscSurvey")

catch_wcgbts <- nwfscSurvey::pull_catch(survey =
        "NWFSC.Combo",
        common_name = "arrowtooth flounder",
        years = c(2003, 2024))
catch_triennial <- nwfscSurvey::pull_catch(survey =
        "Triennial",
        common_name = "arrowtooth flounder",
        years = c(1900, 2004))
```

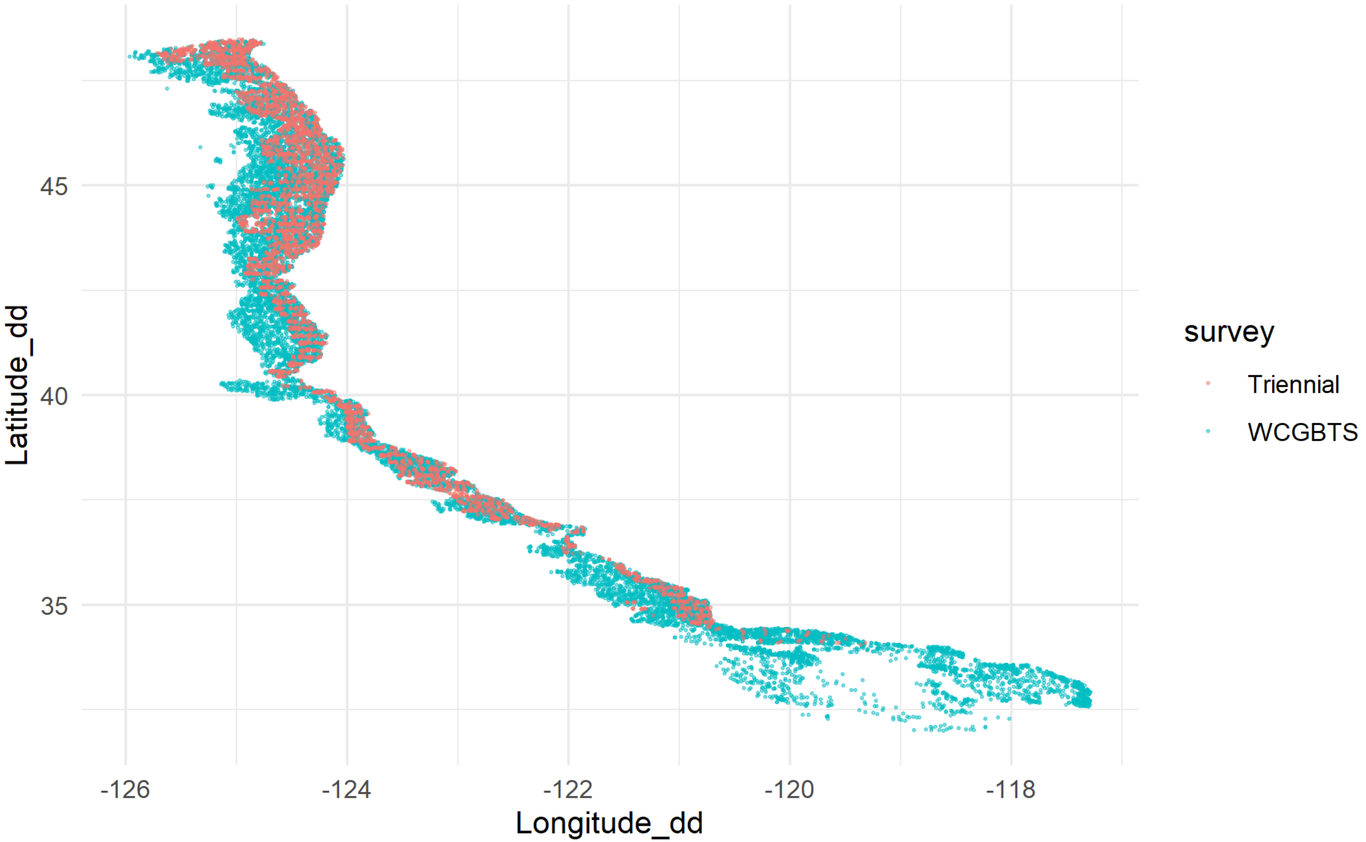
Example 3: overlapping surveys

- Next we'll join the survey data

```
catch_wcgbts$survey <- "WCGBTS"
catch_triennial$survey <- "Triennial"
catch <- rbind(catch_wcgbts, catch_triennial)

# add utm
catch <- sdmTMB::add_utm_columns(catch,
         ll_names = c("Longitude_dd","Latitude_dd"))
```

Example 3: overlapping surveys



Example 3: overlapping surveys

- Like before, we start by constructing the mesh

```
mesh <- sdmTMB::make_mesh(catch,
                          xy_cols = c("X","Y"),
                          cutoff = 25)

mesh$mesh$n
#> [1] 299
```

Example 3: overlapping surveys

- Next we fit the model

```
catch$fyear <- as.factor(catch$Year)
fit5 <- sdmTMB(cpue_kg_km2 ~ 0 + fyear + survey,
          data = catch,
          offset = log(catch$Area_swept_ha),
          mesh = mesh,
          family = delta_lognormal(),
          spatial="on",
          spatiotemporal="iid",
          time = "Year")
```

Example 3: overlapping surveys

- As before we need to prep the prediction grid

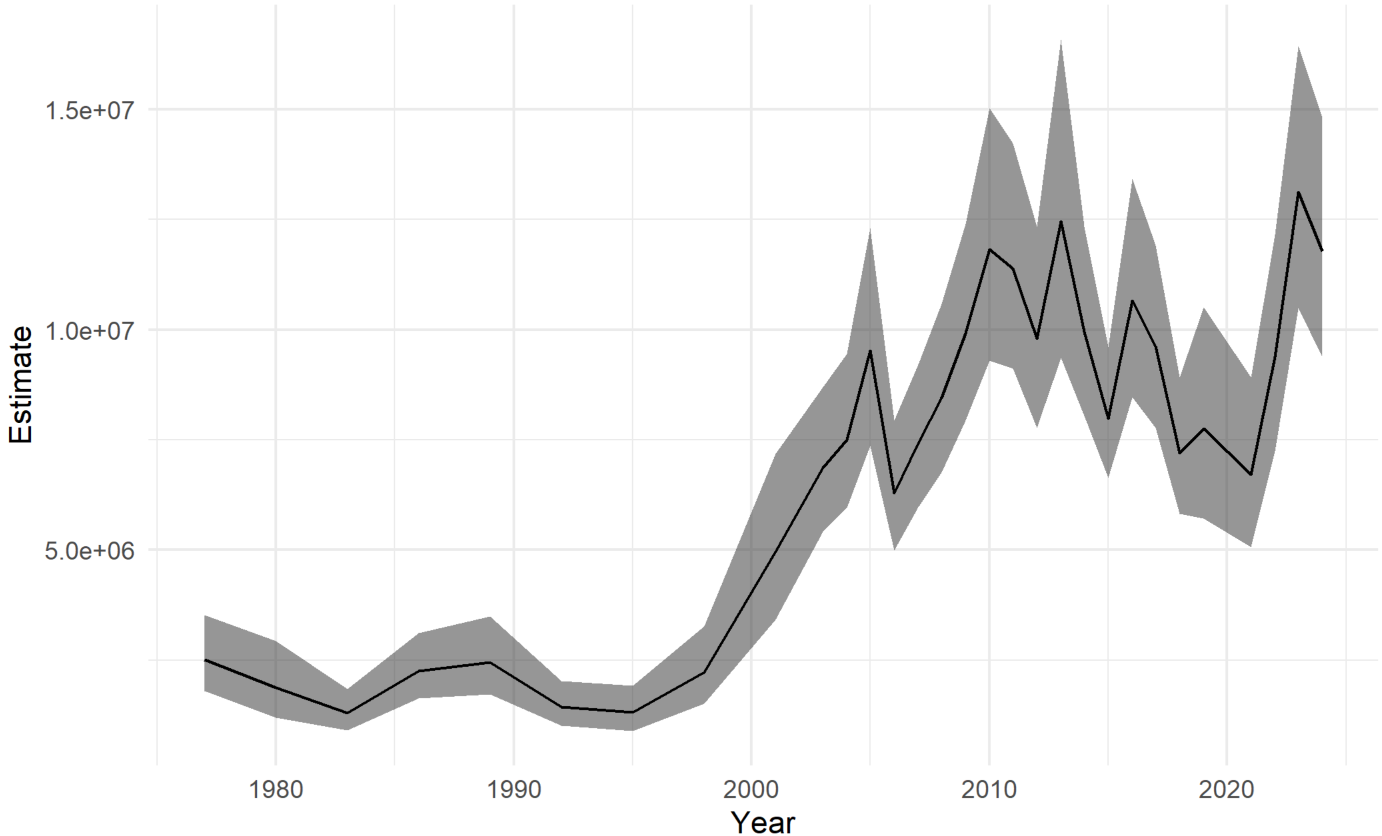
```
grid <- readRDS("surveyjoin_wcgbts_grid.rds")
nd <- sdmTMB::replicate_df(grid, "Year",
                   unique(catch$Year))
nd$fyear <- as.factor(nd$Year)
nd$survey <- "WCGBTS"
```

Example 3: overlapping surveys

- And generate the index

```
pred5 <- predict(fit5, newdata = nd,
                 return_tmb_object = TRUE)
index5 <- get_index(pred5, bias_correct = TRUE)
```

Example 3: overlapping surveys



Summary

- If empirical data does not exist to inform intercalibration, spatiotemporal models can come to the rescue
- Surveys need to occur in the same year to be more tractable
- Integration works better with overlap in space and time