Exploring sdmTMB for cod condition

Max Lindmark

9/18/2020

Background

The body condition and growth of Eastern Baltic cod has declined steadily since the regime shift in the early 1990's to a degree that it now can be viewed as collapsed. Several hypotheses have been put forward, including changes in overlap with pelagic prey (e.g. Gårdmark et al, 2015), reduced oxygen levels (e.g. Casini et al, 2016), increased competition for benthic food sources with flounder (Orio et al, 2019) as well as increased intraspecific competition and growth bottlenecks within the population (Svedäng & Hornborg, 2014).

Methods

In this script I will fit spatiotemporal models of allometric weight-length relationships ($w = \alpha l^{\beta}$ -> $log(w) = \alpha + \beta log(l) + \beta_x x$) using sdmTMB. I aim here to first find an appropriate model structure for evaluating how α (log condition factor), has changed over space and time. Next I aim to evaluate the effects of additional covariates at the haul-level ($\beta_x x$) for the predicted weight given a length.

The covariates I currently consider are:

- Density of cod. CPUE available for each haul (added in data)
- Density of flounder. CPUE available for each haul (added in data)
- Oxygen concentration. Possible to link data to nearest haul *(not yet added)*
- Sprat CPUE. *(not yet added)*
- Herring CPUE. *(not yet added)*
- maybe: temperature as a base-covariate through it's effect on growth, metabolism, digestion etc.

(Re pelagics: Semi coarse predictions exist (by ICES rectangle). I can either do a join operation to get the estimated CPUE by rectangle linked to all hauls in that rectangle, or attempt to fit a new model and predict CPUE for the locations where I have haul conditions. Going for the former I think...).

The project currently lives here. Below follows a walkthrough of the first attempts to model this.

```
library(tidyverse)
#> Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame'
#> when loading 'dplyr'
library(tidylog)
library(viridis)
library(sdmTMB)
library(marmap)
library(curl)
```

Now read data

```
#d <- read.csv("data/mdat_cond.csv")
d <- read.csv(
  curl(
    "https://raw.githubusercontent.com/maxlindmark/cod_condition/master/data/mdat_cond.csv"
    ))
```

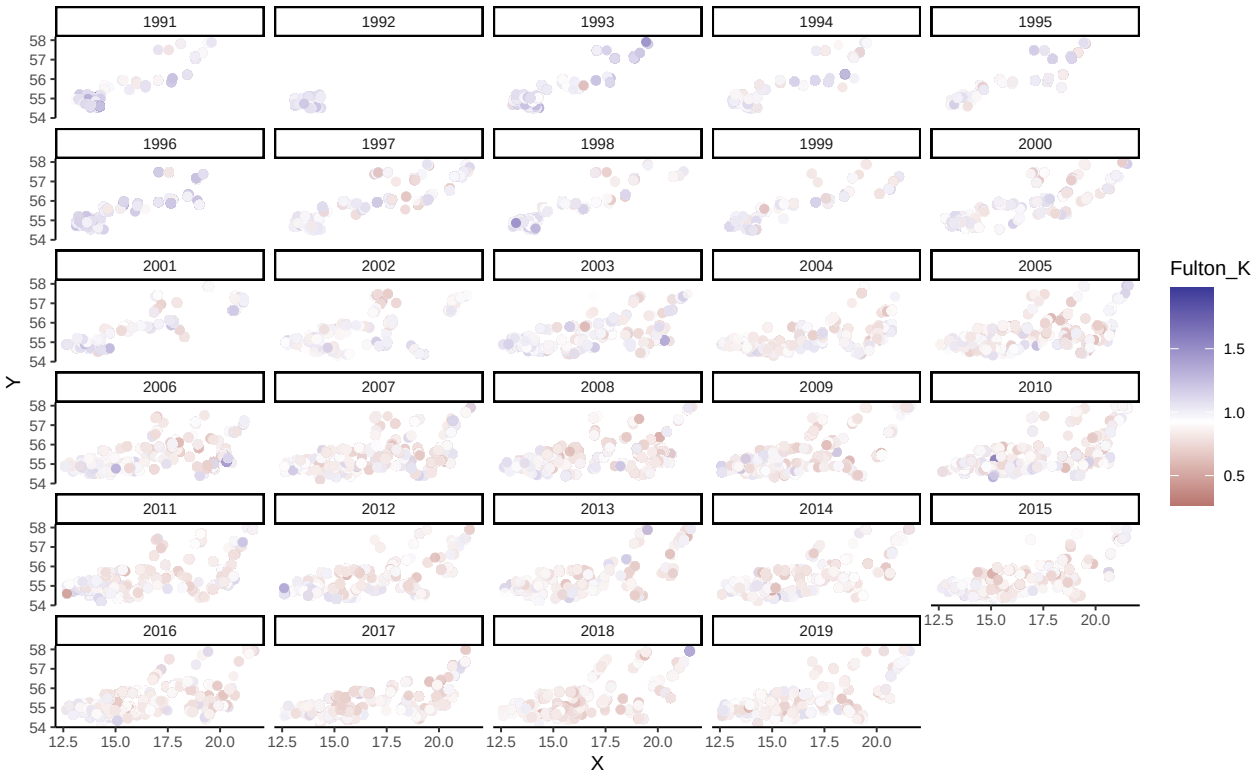
```
# Calculate some variables in data
d <- d %>%
  dplyr::select(-X) %>%
  rename("Y" = "lat", "X" = "lon") %>%
  mutate(ln_weight_g = log(weight_g),
         ln_length_cm = log(length_cm),
         Fulton_K = weight_g/(0.01*length_cm^3)) %>% # Just an approximation for now
  dplyr::select(year, Y, X, sex, length_cm, weight_g, Quarter, CPUE_cod, CPUE_fle,
                ln_length_cm, ln_weight_g, Fulton_K)
#> rename: renamed 2 variables (Y, X)
#> mutate: new variable 'ln_weight_g' (double) with 3,737 unique values and 0% NA
#>         new variable 'ln_length_cm' (double) with 114 unique values and 0% NA
#>         new variable 'Fulton_K' (double) with 14,441 unique values and 0% NA
```

I will next standardize the covariates to have a mean of 0 and variance of 1 to facilitate comparison between different ones, and read in the prediction grid.

```
d <- d %>%
  mutate(CPUE_cod_st = CPUE_cod,
         CPUE_fle_st = CPUE_fle) %>%
  mutate_at(c("CPUE_cod_st", "CPUE_fle_st"), ~(scale(.) %>% as.vector))
#> mutate: new variable 'CPUE_cod_st' (double) with 1,210 unique values and 0% NA
#>         new variable 'CPUE_fle_st' (double) with 816 unique values and 0% NA
#> mutate_at: changed 98,978 values (100%) of 'CPUE_cod_st' (0 new NA)
#>            changed 98,978 values (100%) of 'CPUE_fle_st' (0 new NA)

pred_grid <-
  read.csv(curl(
    "https://raw.githubusercontent.com/maxlindmark/cod_condition/master/data/pred_grid.csv"
    ))

pred_grid <- pred_grid %>%
  dplyr::select(-X.1) %>%
  mutate(ln_length_cm = log(1), # For now we'll predict changes in the intercept (condition factor)
         X = round(X, digits = 4), # I get a strange memory issues when plotting but this solves it
         Y = round(Y, digits = 4)) %>%
  filter(year %in% c(unique(d$year)))
#> mutate: changed 79,019 values (91%) of 'X' (0 new NA)
#>         changed 78,776 values (91%) of 'Y' (0 new NA)
#>         new variable 'ln_length_cm' (double) with one unique value and 0% NA
#> filter: no rows removed
```

We can now plot the Fulton K condition factor to get a glimpse of what we might expect (but finally our estimates are closer to Le Cren's condition index).

```
# Plot "Fulton K" in space and time
d %>%
  ggplot(., aes(X, Y, color = Fulton_K)) +
  geom_point(size = 1.2, alpha = 0.8) +
  facet_wrap(~ year, ncol = 5) +
  scale_color_gradient2(midpoint = mean(d$Fulton_K)) +
  theme_classic(base_size = 8) +
  NULL
```

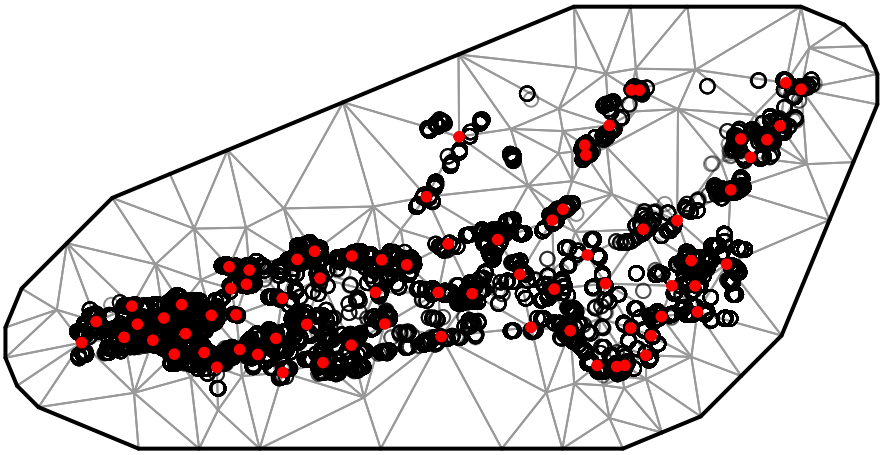


Ok there's a clear temporal development in condition and the spatial coverage of data varies by year (less data initially).

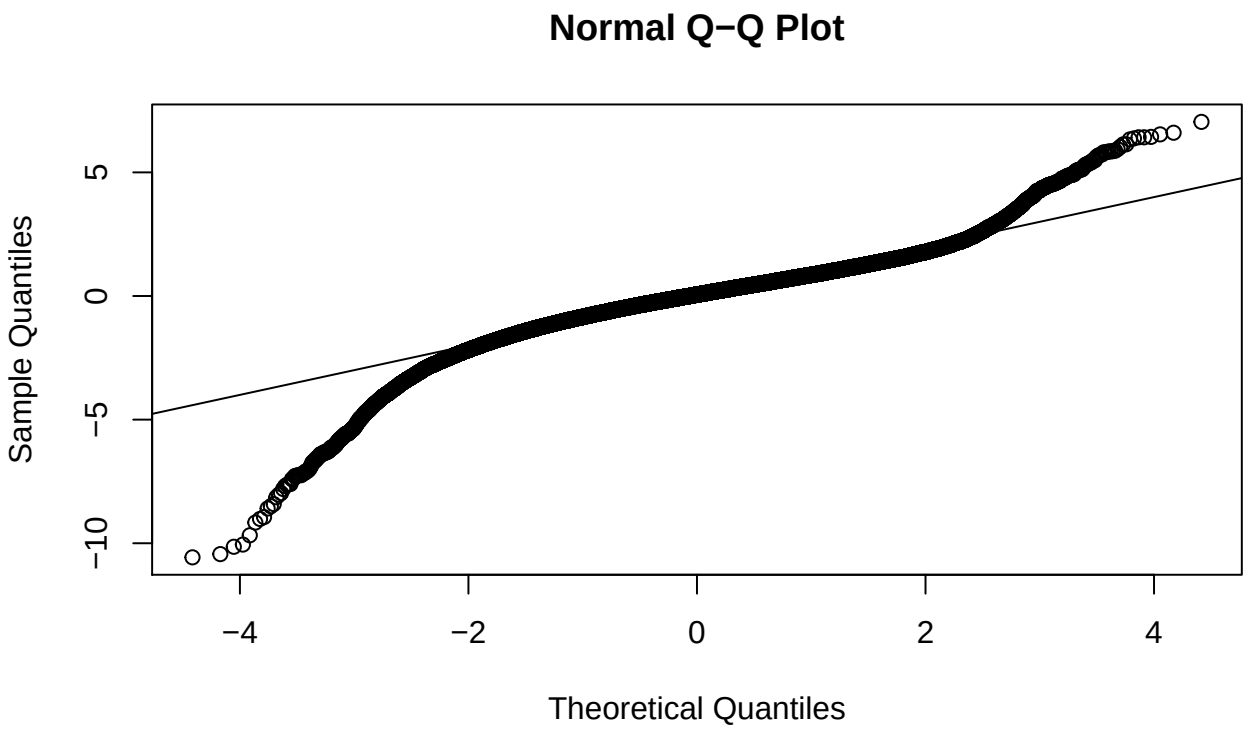
Next we fit spatial/spatiotemporal models to account for this. Given that we do not have samples in all spatial locations for all years, and we believe that there are "hotpots" of body condition, we model a temporal correlation for the spatiotemporal variation (Thorson, 2019), specifically, this is done by estimating the spatiotemporal fields as an AR1 process. Hence, we set ar1_fields = TRUE.

I will start by setting up the SPDE mesh with 75 knots and alter I will fit and compare models with more knots. In the first step I will compare a Gaussian with a student t model to see which distribution seems more appropriate.

```
spde <- make_spde(d$X, d$Y, n_knots = 75)
plot_spde(spde)
```

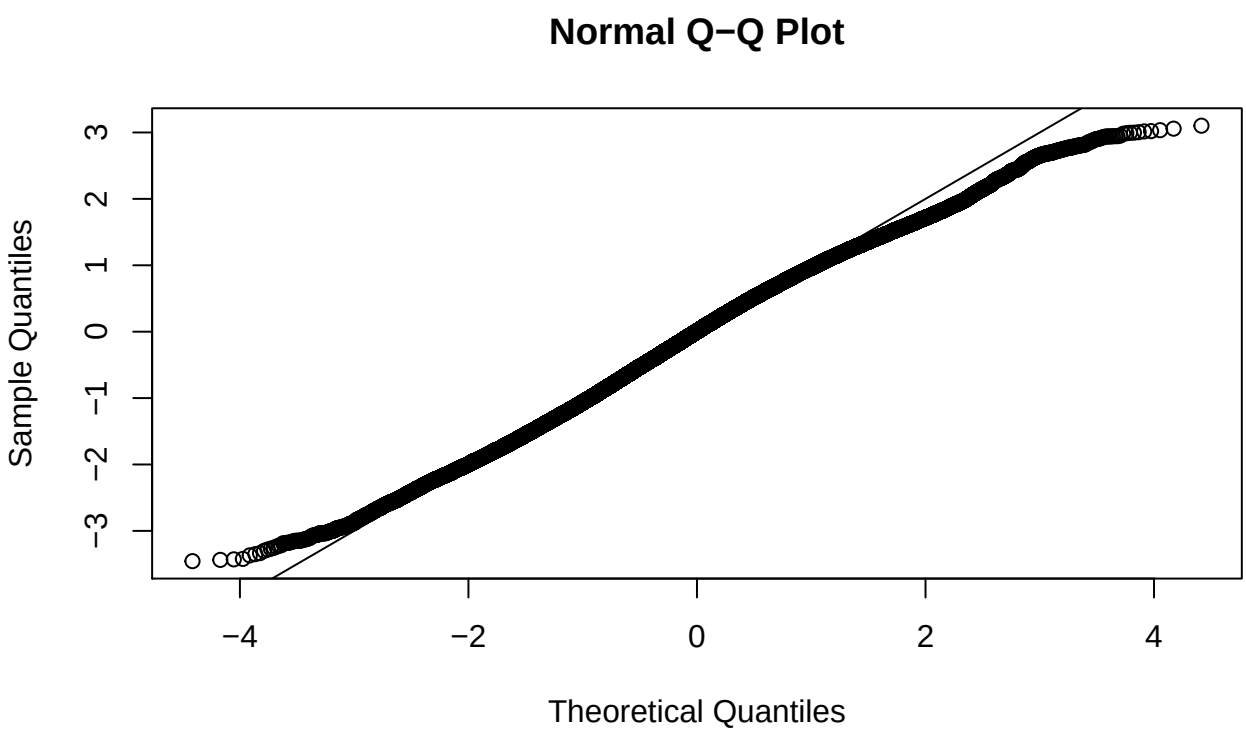


```r
# Compare Gaussian and student t models with a spatiotemporal AR1 process
m0 <- sdmTMB(formula = ln_weight_g ~ ln_length_cm, data = d, time = "year", spde = spde,
             family = gaussian(link = "identity"), ar1_fields = TRUE,
             include_spatial = TRUE,  spatial_trend = FALSE, spatial_only = FALSE)

m1 <- sdmTMB(formula = ln_weight_g ~ ln_length_cm, data = d, time = "year", spde = spde,
             family = student(link = "identity"), ar1_fields = TRUE,
             include_spatial = TRUE, spatial_trend = FALSE, spatial_only = FALSE)

# Inspect fitted models
print(m0)
#> Spatiotemporal model fit by ML ['sdmTMB']
#> Formula: ln_weight_g ~ ln_length_cm
#> SPDE: spde
#> Family: gaussian(link = 'identity')
#>               coef.est coef.se
#> (Intercept)     -4.58    0.02
#> ln_length_cm     2.98    0.00
#>
#> Matern range parameter: 1.04
#> Dispersion parameter: 0.12
#> Spatial SD (sigma_O): 0.08
#> Spatiotemporal SD (sigma_E): 0.08
#> Spatiotemporal AR1 correlation (rho): 0.49
#> ML criterion at convergence: -70218.513
print(m1)
#> Spatiotemporal model fit by ML ['sdmTMB']
#> Formula: ln_weight_g ~ ln_length_cm
#> SPDE: spde
#> Family: student(link = 'identity')
#>               coef.est coef.se
#> (Intercept)     -4.58    0.02
#> ln_length_cm     2.98    0.00
#>
#> Matern range parameter: 1.04
#> Dispersion parameter: 0.08
#> Spatial SD (sigma_O): 0.08
#> Spatiotemporal SD (sigma_E): 0.08
#> Spatiotemporal AR1 correlation (rho): 0.47
#> ML criterion at convergence: -75600.012

# Look at the residuals:
df <- d

df$residuals_m0 <- residuals(m0)
df$residuals_m1 <- residuals(m1)

qqnorm(df$residuals_m0); abline(a = 0, b = 1)
```



Gaussian looks bad.

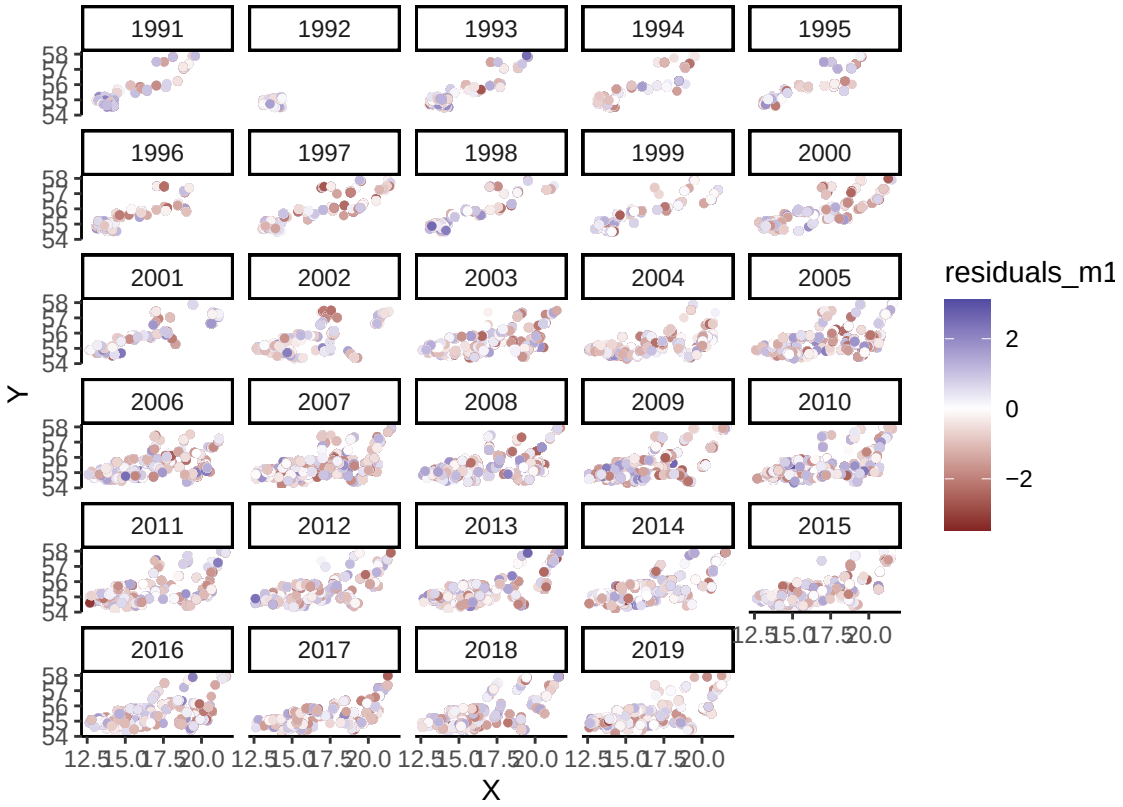
```
qqnorm(df$residuals_m1); abline(a = 0, b = 1)
```



Student looks a lot better but could perhaps be improved further. Check the residuals for the student t model on a map.

```
ggplot(df, aes(X, Y, colour = residuals_m1)) +
  geom_point(size = 1) +
```

```
  facet_wrap(~year, ncol = 5) +
  scale_color_gradient2() +
  theme_classic() +
  coord_fixed()
```



Maybe some clustering remains...

However, moving on we can also look at the AR1 parameter to ensure it is warranted.

```
sd1 <- as.data.frame(summary(TMB::sdreport(m1$tmb_obj)))
sd1$Estimate[row.names(sd1) == "ar1_phi"]
#> [1] 1.019191
sd1$Estimate[row.names(sd1) == "ar1_phi"] +
  c(-2, 2) * sd1$`Std. Error`[row.names(sd1) == "ar1_phi"]
#> [1] 0.7932714 1.2451101
```

Strong support for it, will not run model without AR1 process in the spatiotemporal field for now.

Hence, I will proceed with this model, makes some predictions and compare it to a model with covariates.

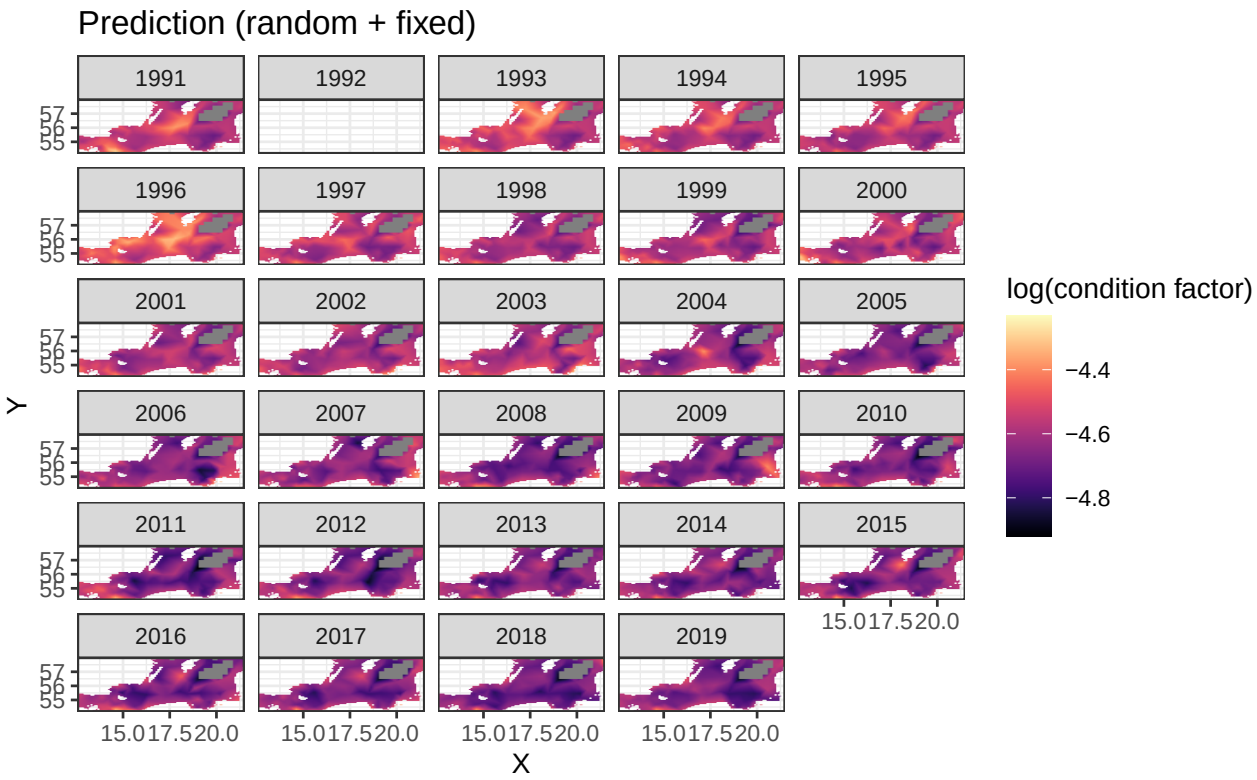
So, now we can predict and plot estimates using all fixed and random effects on pre-made grid. This grid is made by doing an expand grid over survey ranges, then filtering out areas that are actually in the ocean using ICES shapefiles. Lastly some areas are too deep for sampling (-135 m). I've added a depth column so that I can make those predictions NA so it's clear they are different from e.g. land and islands (maybe unnessecary but it makes them grey, i.e. different from land.

```
p <- predict(m1, newdata = pred_grid)

# Replace too-deep predictions with NA
p <- p %>% mutate(est2 = ifelse(depth < -130, NA, est),
                  est_rf2 = ifelse(depth < -130, NA, est_rf),
```

```
                  omega_s2 = ifelse(depth < -130, NA, omega_s))
#> mutate: new variable 'est2' (double) with 80,128 unique values and 8% NA
#>         new variable 'est_rf2' (double) with 80,128 unique values and 8% NA
#>         new variable 'omega_s2' (double) with 3,137 unique values and 8% NA
```

Plot the predicted condition with fixed and random effects:

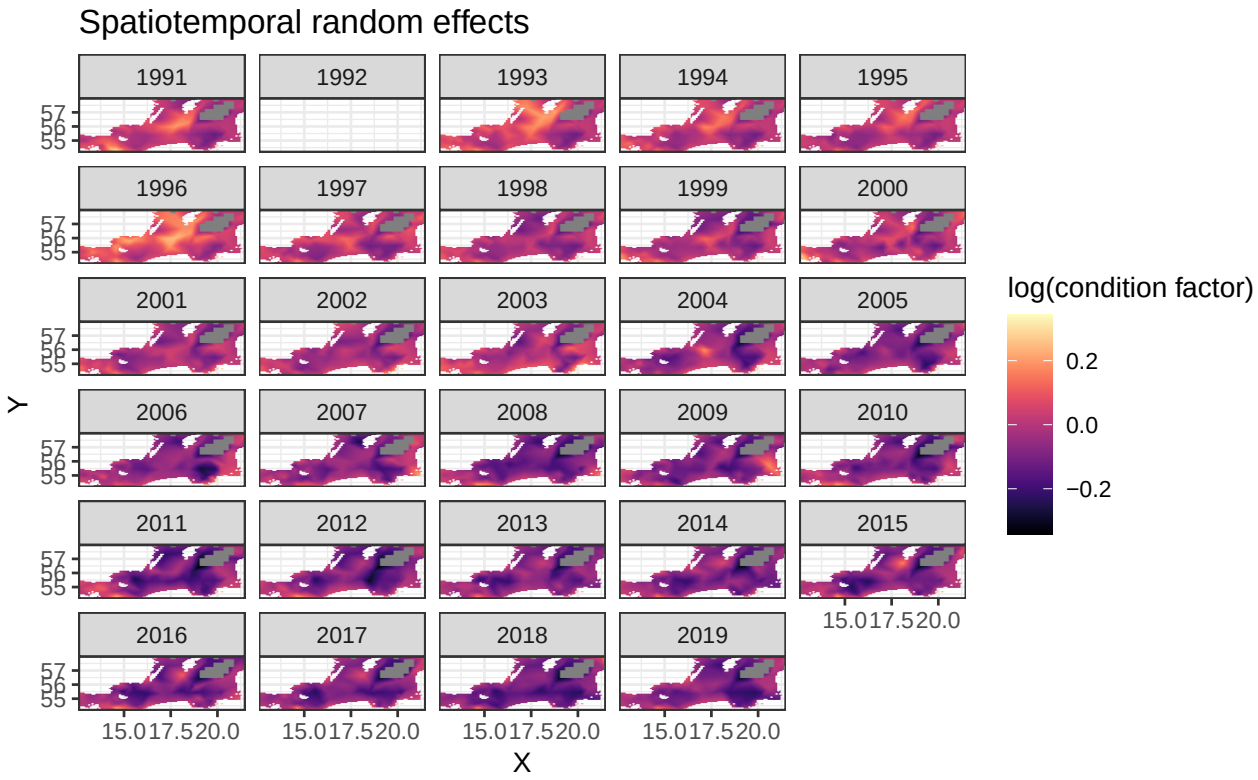
```
ggplot(p, aes(X, Y, fill = est2)) +
  geom_raster() +
  facet_wrap(~year, ncol = 5) +
  scale_fill_viridis(option = "magma",
                     name = "log(condition factor)") +
  theme_bw() +
  coord_cartesian(expand = 0) +
  ggtitle("Prediction (random + fixed)")
#> Warning: Raster pixels are placed at uneven horizontal intervals and will be
#> shifted. Consider using geom_tile() instead.
#> Warning: Raster pixels are placed at uneven vertical intervals and will be
#> shifted. Consider using geom_tile() instead.
```



Plot the spatiotemporal random effects:

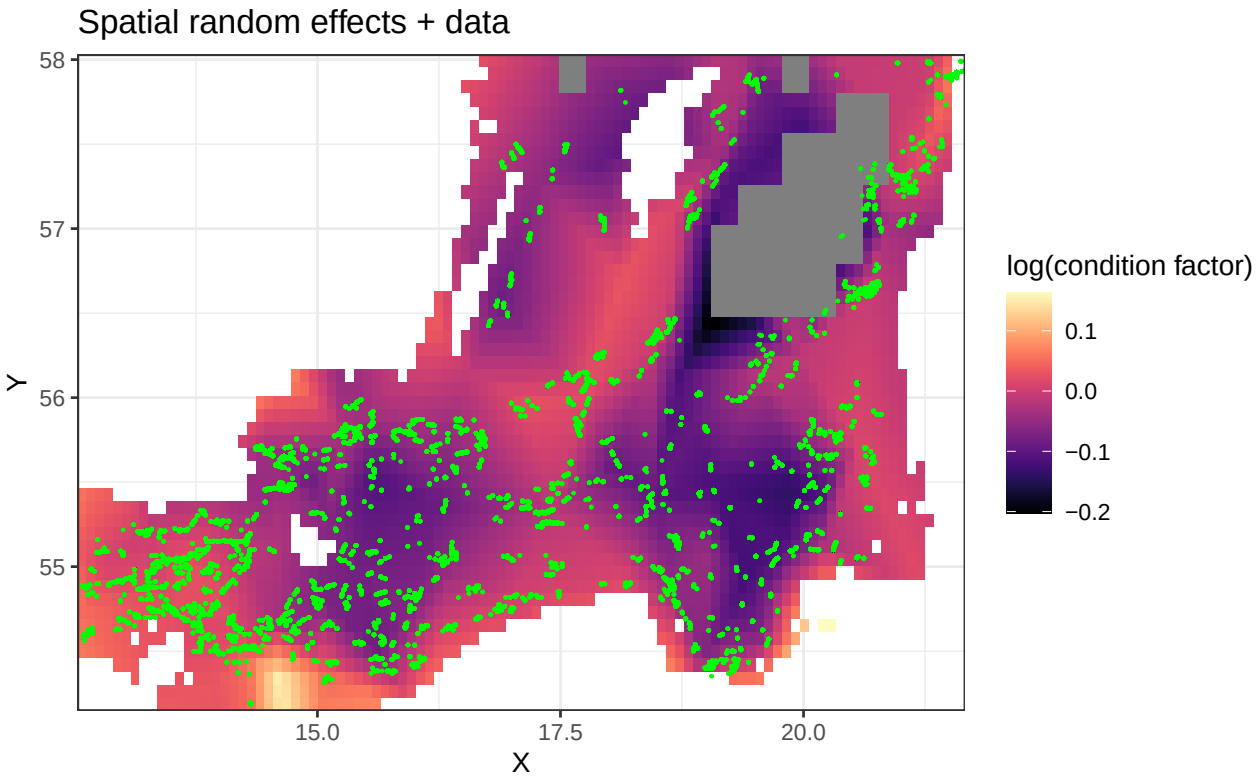
```
ggplot(p, aes(X, Y, fill = est_rf2)) +
  geom_raster() +
  facet_wrap(~year, ncol = 5) +
  scale_fill_viridis(option = "magma",
                     name = "log(condition factor)") +
  theme_bw() +
  coord_cartesian(expand = 0) +
  ggtitle("Spatiotemporal random effects")
```

```
#> Warning: Raster pixels are placed at uneven horizontal intervals and will be
#> shifted. Consider using geom_tile() instead.
#> Warning: Raster pixels are placed at uneven vertical intervals and will be
#> shifted. Consider using geom_tile() instead.
```



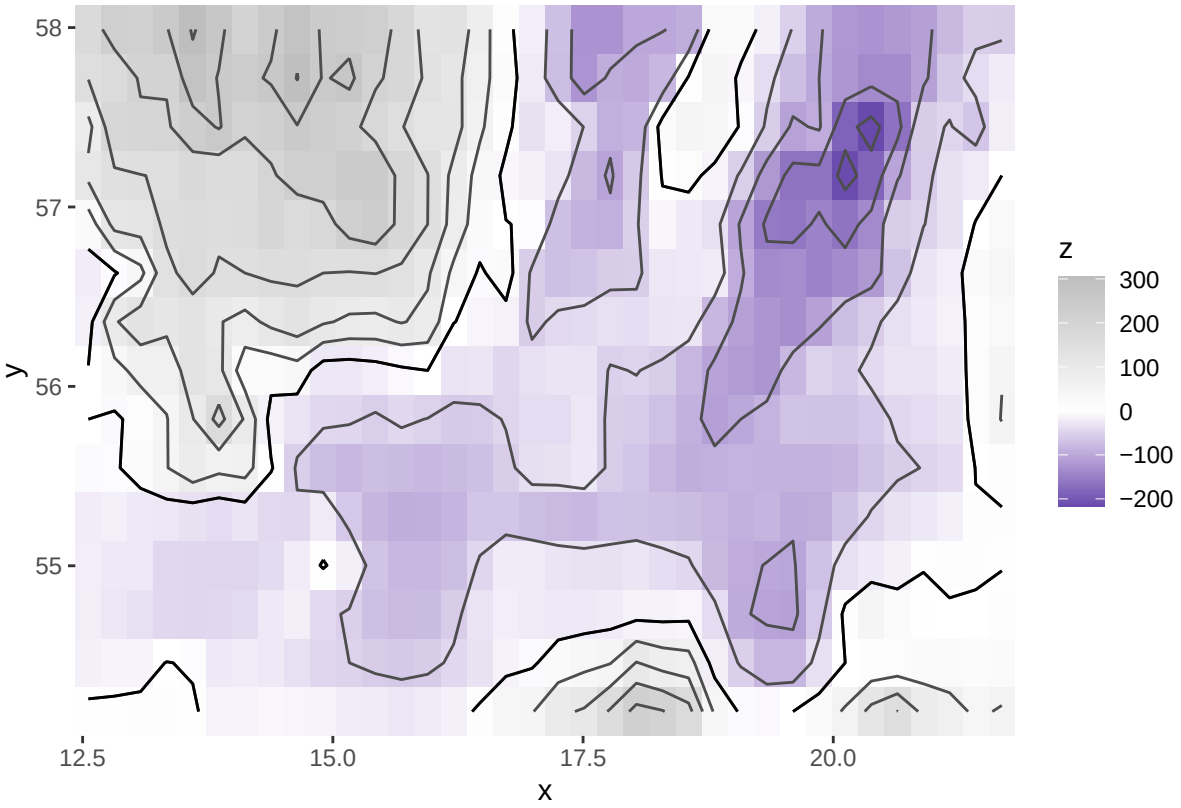
Plot the spatial random effects:

```
ggplot(filter(p, year == 2000), aes(X, Y, fill = omega_s2)) +
  geom_raster() +
  scale_fill_viridis(option = "magma",
                     name = "log(condition factor)") +
  theme_bw() +
  coord_cartesian(expand = 0) +
  geom_point(data = d, aes(X, Y), color = "green", inherit.aes = FALSE, size = 0.2) +
  ggtitle("Spatial random effects + data")
#> filter: removed 83,888 rows (97%), 2,996 rows remaining
#> Warning: Raster pixels are placed at uneven horizontal intervals and will be
#> shifted. Consider using geom_tile() instead.
#> Warning: Raster pixels are placed at uneven vertical intervals and will be
#> shifted. Consider using geom_tile() instead.
```



Predictions with fixed and random effects and the spatiotemporal random effect are quite similar. I guess this is because I don't have any other strong covariates yet (and the length-variable is set to 0 for this prediction). All random effects (and predictions) also seem to largely follow depth. Not really sure how to interpret that. At the deeper areas there is less oxygen and less benthic food so in theory it can make sense).

```
baltic_sea <- getNOAA.bathy(lon1 = min(d$X), lon2 = max(d$X),
                            lat1 = min(d$Y), lat2 = max(d$Y), resolution = 15)
#> Querying NOAA database ...
#> This may take seconds to minutes, depending on grid size
#> Building bathy matrix ...

autoplot(baltic_sea, geom = c("r", "c")) +
  scale_fill_gradient2(low = "darkblue", high = "gray", midpoint = 0)
```



Now we want to refit the same model with the additional fixed effects outlined above.

```
# Fit model with cod cpue as covariate
mcod <- sdmTMB(formula = ln_weight_g ~ ln_length_cm + CPUE_cod_st, data = d, time = "year",
               spde = spde, family = student(link = "identity"),
               ar1_fields = TRUE,
               include_spatial = TRUE,
               spatial_trend = FALSE,
               spatial_only = FALSE)
#> Warning: The model may not have converged. Maximum final gradient:
#> 0.0743696046508351.

# Run extra optimization steps to help convergence:
mcod2 <- run_extra_optimization(mcod, nlminb_loops = 1, newton_steps = 1)
#> Warning: The model may not have converged. Maximum final gradient:
#> 0.0743696046508351.

## ---------- Actually this did not improve convergence this time!

#... And with flounder
mfle <- sdmTMB(formula = ln_weight_g ~ ln_length_cm + CPUE_fle_st, data = d, time = "year",
               spde = spde, family = student(link = "identity"),
               ar1_fields = TRUE,
               include_spatial = TRUE,
               spatial_trend = FALSE,
               spatial_only = FALSE)
#> Warning: The model may not have converged. Maximum final gradient:
#> 0.028401916766768.
```

```r
# Run extra optimization steps to help convergence:
mfle2 <- run_extra_optimization(mfle, nlminb_loops = 1, newton_steps = 1)

# Check the models
print(mcod2)
#> Spatiotemporal model fit by ML ['sdmTMB']
#> Formula: ln_weight_g ~ ln_length_cm + CPUE_cod_st
#> SPDE: spde
#> Family: student(link = 'identity')
#>              coef.est coef.se
#> (Intercept)     -4.57    0.02
#> ln_length_cm     2.98    0.00
#> CPUE_cod_st      0.00    0.00
#>
#> Matern range parameter: 1.05
#> Dispersion parameter: 0.08
#> Spatial SD (sigma_O): 0.08
#> Spatiotemporal SD (sigma_E): 0.08
#> Spatiotemporal AR1 correlation (rho): 0.47
#> ML criterion at convergence: -75604.564
print(mfle2)
#> Spatiotemporal model fit by ML ['sdmTMB']
#> Formula: ln_weight_g ~ ln_length_cm + CPUE_fle_st
#> SPDE: spde
#> Family: student(link = 'identity')
#>              coef.est coef.se
#> (Intercept)     -4.58    0.02
#> ln_length_cm     2.98    0.00
#> CPUE_fle_st      0.00    0.00
#>
#> Matern range parameter: 1.04
#> Dispersion parameter: 0.08
#> Spatial SD (sigma_O): 0.08
#> Spatiotemporal SD (sigma_E): 0.08
#> Spatiotemporal AR1 correlation (rho): 0.47
#> ML criterion at convergence: -75600.560

# Look at the new parameter (cod)
sdmcod <- as.data.frame(summary(TMB::sdreport(mcod2$tmb_obj)))

sdmcod$Estimate[row.names(sdmcod) == "b_j.2"] # The second term, aka cod
#> [1] 0.001785061

sdmcod$Estimate[row.names(sdmcod) == "b_j.2"] +
  c(-2, 2) * sdmcod$`Std. Error`[row.names(sdmcod) == "b_j.2"]
#> [1] 0.0006024204 0.0029677018

#... And the same for flounder
sdmfle <- as.data.frame(summary(TMB::sdreport(mfle2$tmb_obj)))

sdmfle$Estimate[row.names(sdmfle) == "b_j.2"] # The second term, aka flounder
#> [1] 0.0009320629
```

```
sdmfle$Estimate[row.names(sdmfle) == "b_j.2"] +
  c(-2, 2) * sdmfle$`Std. Error`[row.names(sdmfle) == "b_j.2"]
#> [1] -0.0008480145  0.0027121403

# Compare the models with AIC (unsure actually if this is correct for a sdmTMB model...)
aic_m1 <- extractAIC(m1)
aic_mcod <- extractAIC(mcod2)
aic_mfle <- extractAIC(mfle2)

aic_m1
#> [1]       7 -151186
aic_mcod
#> [1]       8.0 -151193.1
aic_mfle
#> [1]       8.0 -151185.1
```

Despite some convergence problems I will proceed just as an example. The model with flounder has a smaller AIC, even though the 95% confidence interval for it's coefficient crosses 0 (unlike the coefficient for cod).

Now let's look more closely at the our estimates. If I understand Thorson (2015) correctly:

"*This implies that γX (the covariate times its coefficient) has a standard deviation of γ such that coefficients can be interpreted via comparison with the standard deviation of spatial, temporal and spatiotemporal variation, as well as that of residual variation.*"

I can now compare the coefficient of flounder with the standard deviation of the spatial and spatiotemporal effects, i.e. σ_E and σ_A in Thorson (2015) (Eqns. 6b-7). These terms are the square roots of the marginal variances of the random fields, i.e. σ_E^2 and σ_A^2.

In sdmTMB I think σ_E and σ_A above correspond to Spatiotemporal SD (sigma_E) and Spatial SD (sigma_O) seen in "print(model)". I think the non-rounded values can be extracted from:

```
mfle2$sd_report$value
#>       sigma_O sigma_O_trend       sigma_E         range
#>    0.08323469    0.10374579    0.07966072    1.04020857
```

These standard deviations can now be compared with the coefficients of the flounder model:

```
sdmfle$Estimate[row.names(sdmfle) == "b_j.2"] # The second term, aka flounder
#> [1] 0.0009320629
```

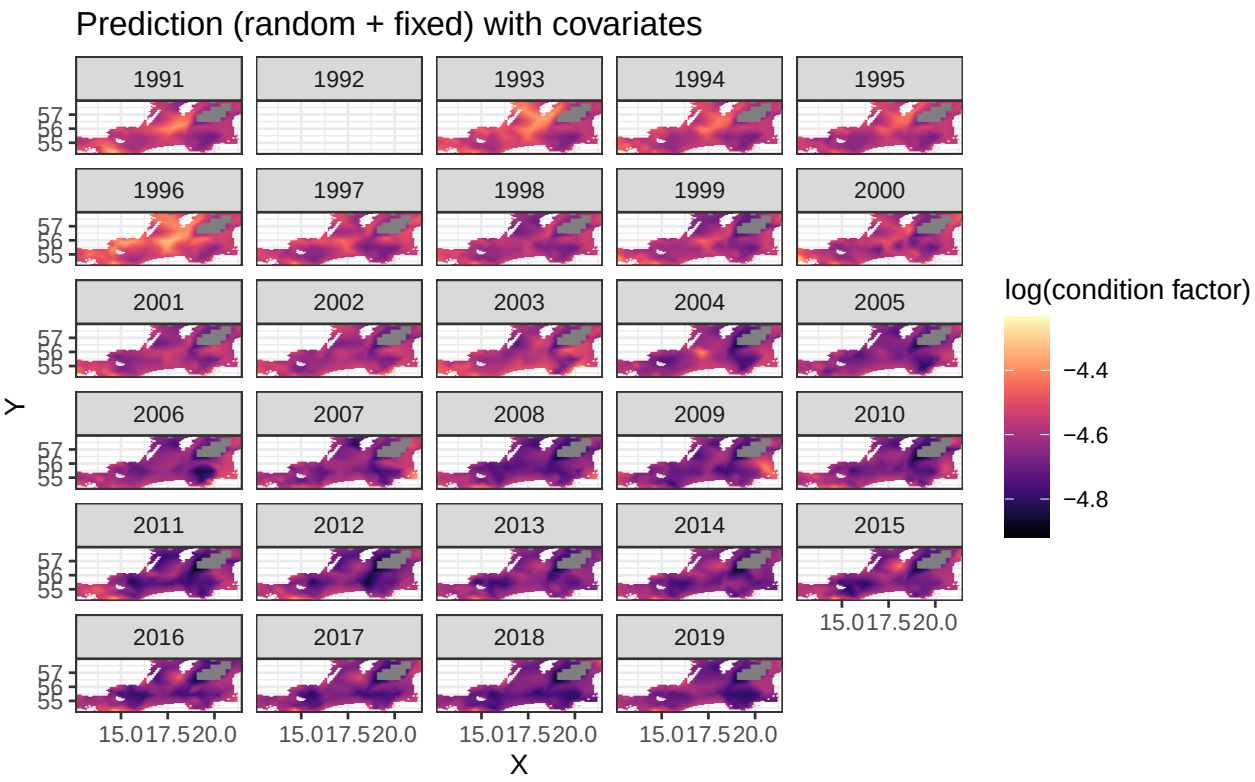
I interpret this as that the flounder coefficient is small relative to other sources of temporally constant variation across space (omega) and factors varying in space from year to year (epsilon). Further, I don't think inclusion of the flounder covariate leads to less variation explained by the spatial and spatiotemporal effects. Compare those standard deviations with the model without covariates:

```
m1$sd_report$value
#>       sigma_O sigma_O_trend       sigma_E         range
#>    0.08397645    0.10368044    0.07974713    1.03955324
```

For the sake of comparison, I can also produce a map to look at the differences there.

```
# Add in a fixed covariate here
pred_grid_fle <- pred_grid
pred_grid_fle$CPUE_fle_st <- 0 # Mean since standardized

pfle <- predict(mfle2, newdata = pred_grid_fle)

# Replace too-deep predictions with NA
```

```
pfle <- pfle %>% mutate(est2 = ifelse(depth < -130, NA, est))
#> mutate: new variable 'est2' (double) with 80,128 unique values and 8% NA

ggplot(pfle, aes(X, Y, fill = est2)) +
  geom_raster() +
  facet_wrap(~year, ncol = 5) +
  scale_fill_viridis(option = "magma",
                     name = "log(condition factor)") +
  theme_bw() +
  coord_cartesian(expand = 0) +
  ggtitle("Prediction (random + fixed) with covariates")
#> Warning: Raster pixels are placed at uneven horizontal intervals and will be
#> shifted. Consider using geom_tile() instead.
#> Warning: Raster pixels are placed at uneven vertical intervals and will be
#> shifted. Consider using geom_tile() instead.
```



To do

- The spatial_trend argument is ignored for now [see e-mail], is my interpretation correct?
- Do I extract the standard deviations for the spatial and spatiotemporal trends correctly, and interpret then correctly in relation to covariates?

References

Anderson, S.C., Keppel, E.A., Edwards, A.M. 2019. A reproducible data synopsis for over 100 species of British Columbia groundfsh. DFO Can. Sci. Advis. Sec. Res. Doc. 2019/041. vii + 321 p.

Casini, M., Käll, F., Hansson, M., Plikshs, M., Baranova, T., Karlsson, O., Lundström, K., Neuenfeldt, S., Gårdmark, A. and Hjelm, J., 2016. Hypoxic areas, density-dependence and food limitation drive the body

condition of a heavily exploited marine fish predator. Royal Society open science, 3(10), p.160416.

Gårdmark, A., Casini, M., Huss, M., van Leeuwen, A., Hjelm, J., Persson, L. and de Roos, A.M., 2015. Regime shifts in exploited marine food webs: detecting mechanisms underlying alternative stable states using size-structured community dynamics theory. Philosophical Transactions of the Royal Society B: Biological Sciences, 370(1659), p.20130262.

Orio, A., Bergström, U., Florin, A.B., Lehmann, A., Šics, I. and Casini, M., 2019. Spatial contraction of demersal fish populations in a large marine ecosystem. Journal of Biogeography, 46(3), pp.633-645.

Svedäng, H. and Hornborg, S., 2014. Selective fishing induces density-dependent growth. Nature communications, 5(1), pp.1-6.

Thorson, J.T., 2015. Spatio-temporal variation in fish condition is not consistently explained by density, temperature, or season for California Current groundfishes. *Marine Ecology Progress Series*, 526, pp.101-112.