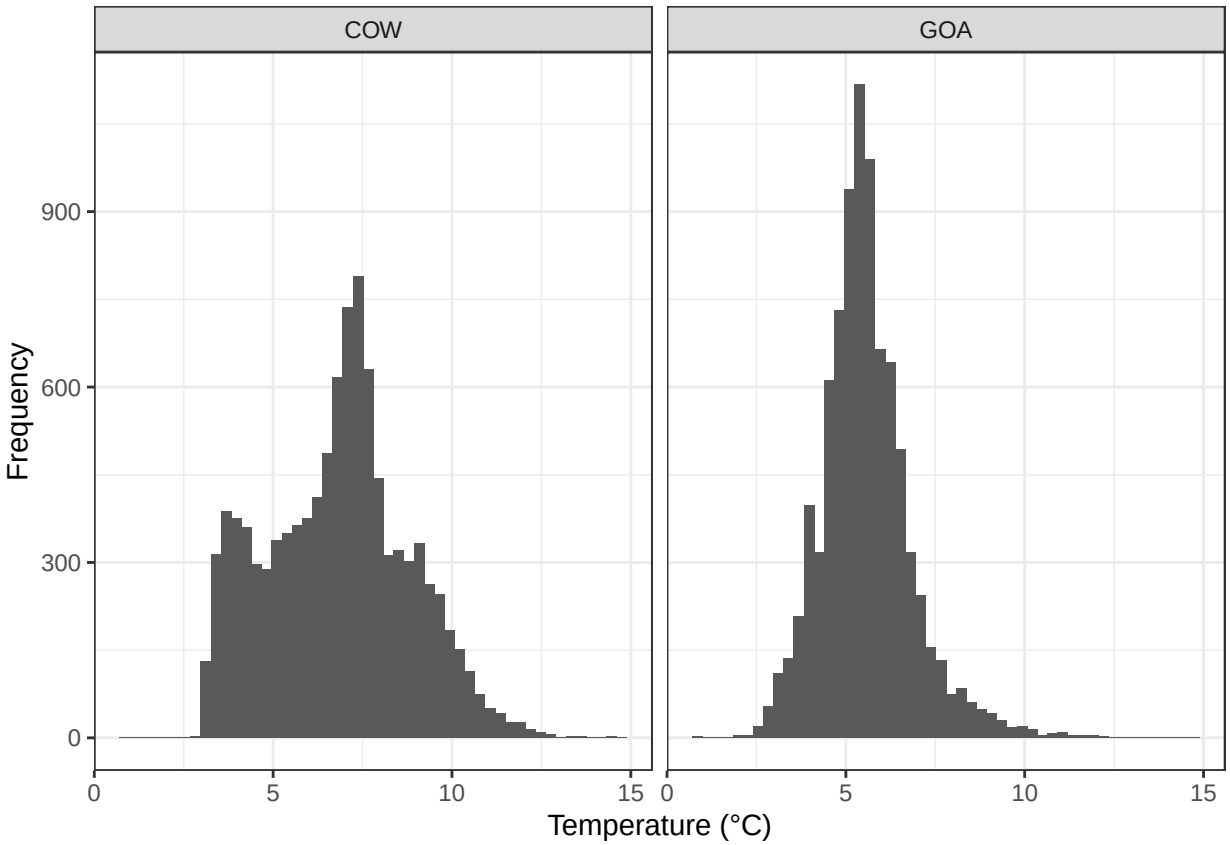
Supplemental Information

EW

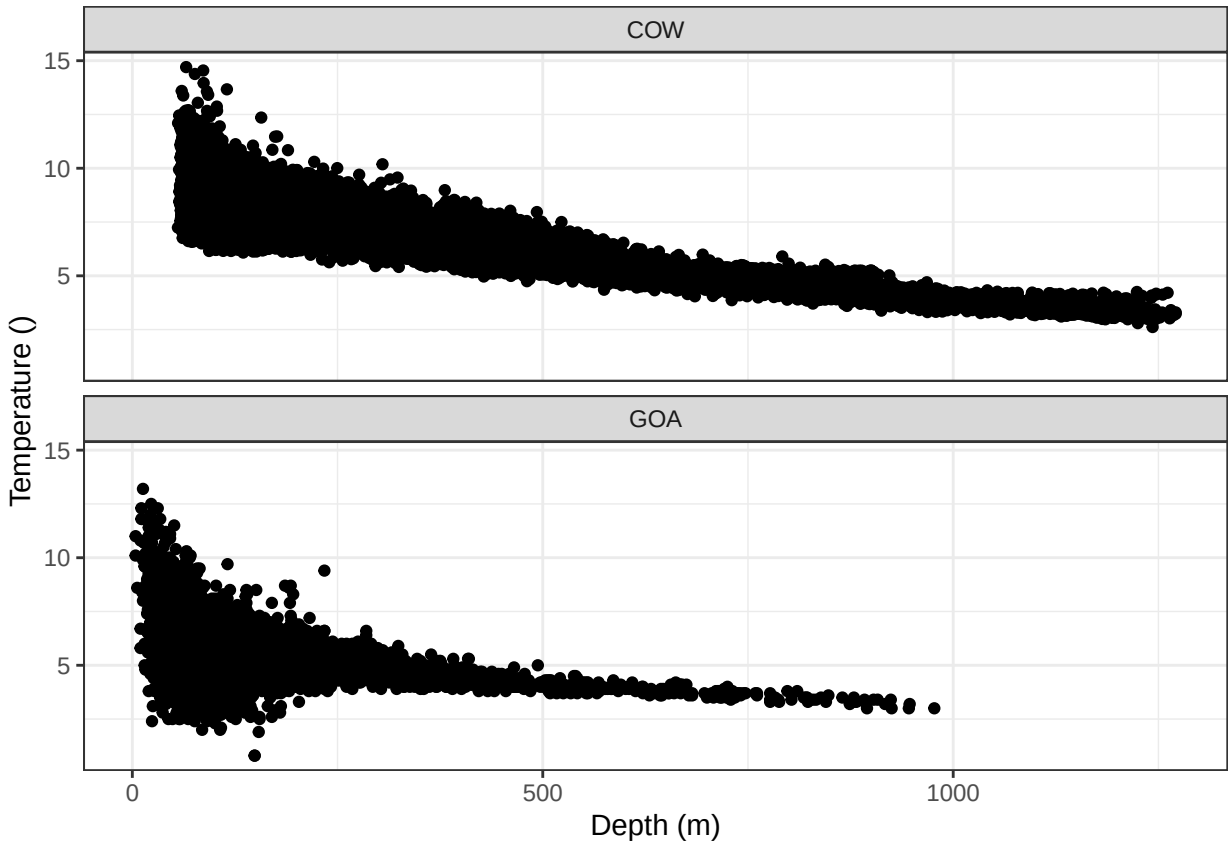
3/1/2021

Empirical observations of temperature

Each survey included in our analysis has unique distributions of temperature, and relationships between temperature and depth. Temperature in the Gulf of Alaska appears much more normally distributed, while temperature along the California-Oregon-Washington coast is more skewed.



There is a strong correlation between temperature and depth, and greater variability in temperature at low depths.



Quadratic verus smooth spline

As a sensitivitiy analysis to look at whether inference about the shape of the temperature relationship is dependent on choice of model, we compared models that treated the temperature effect as quadratic to those that modeled temperature as a low-order smooth (k=3). Temperature is standardized to be on the same scale for all species. Quadratic parameters are not derived for species where the quadratic coefficients is positivie (resulting in concave up parabolas). These results highlight that for nearly all species, the results between the two models as nearly identical.

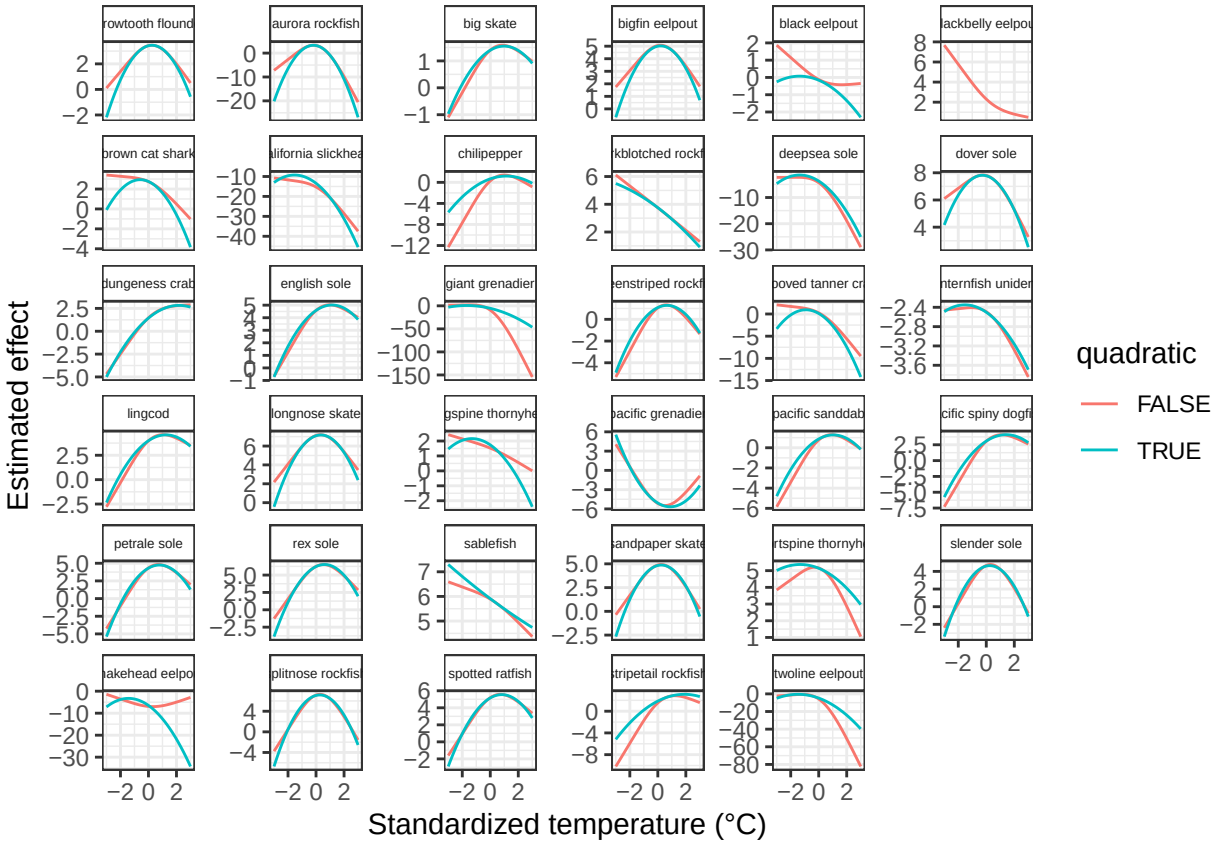


Figure 1: Comparison between quadratic and spline models of temperature for groundfish (COW)

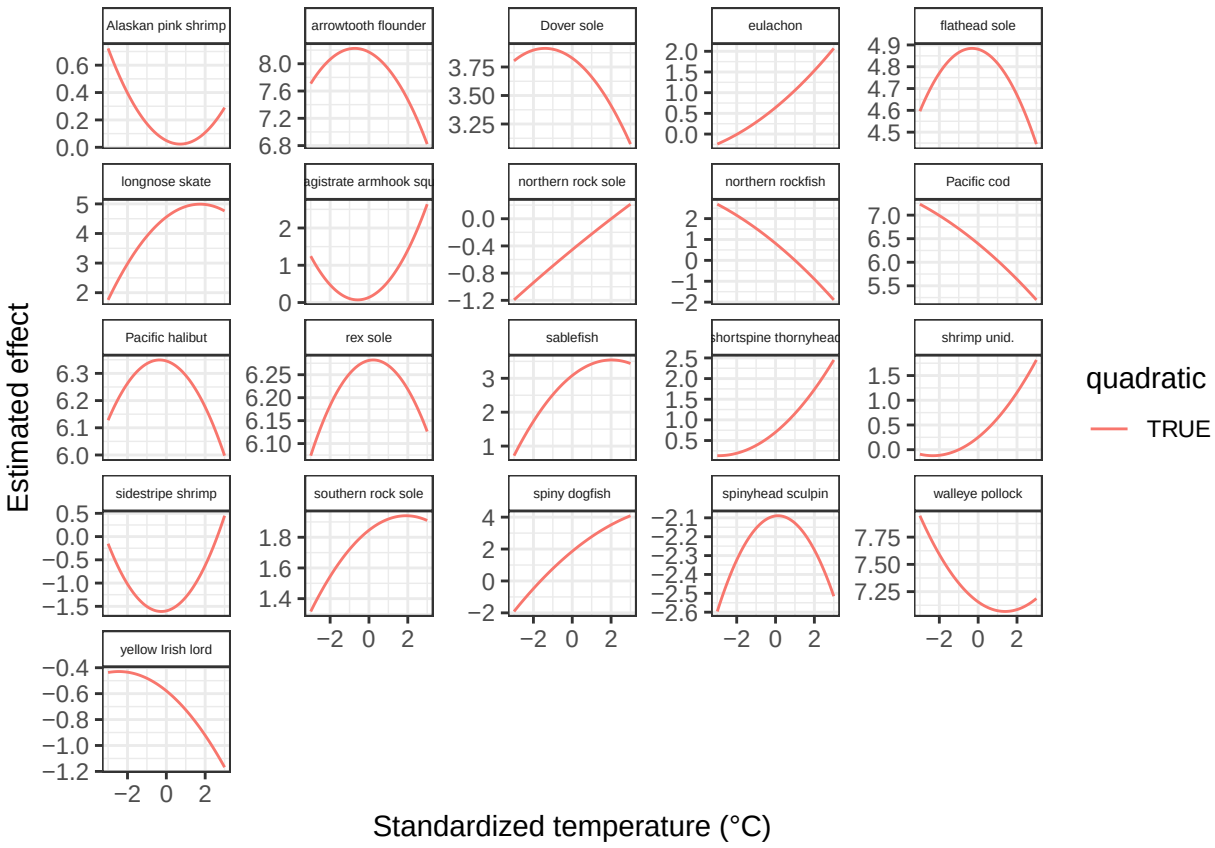
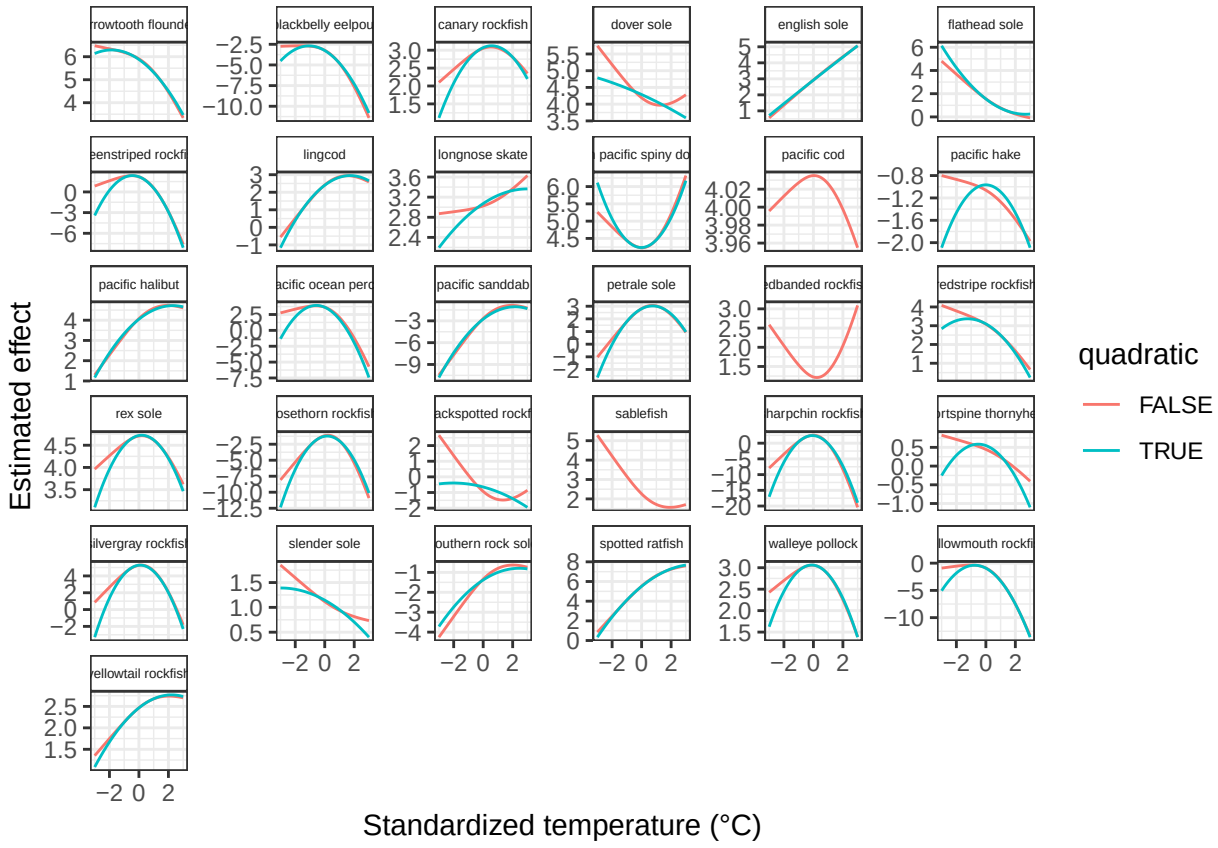


Figure 2: Comparison between quadratic and spline models of temperature for groundfish (GOA)



Simulations

To aid in the interpretation of the quadratic ranges, it is worth simulating relationships based on temperature, depth, or both predictors. We'll do this based on 6 years of the trawl survey data, and use the observed depth and temperature values as predictors, along with the spatial field.

```
dat = readRDS(here("survey_data/joined_nwfsc_data.rds"))
dat = dplyr::filter(dat, species=="greenstriped rockfish",
                    year %in% seq(2013,2018),
                    !is.na(temp),!is.na(depth))

# UTM transformation
dat_ll = dat
coordinates(dat_ll) <- c("longitude_dd", "latitude_dd")
proj4string(dat_ll) <- CRS("+proj=longlat +datum=WGS84")
# convert to utm with spTransform
dat_utm = spTransform(dat_ll,
  CRS("+proj=utm +zone=10 +datum=WGS84 +units=km"))
# convert back from sp object to data frame
dat = as.data.frame(dat_utm)
dat = dplyr::rename(dat, longitude = longitude_dd,
  latitude = latitude_dd)
```

Example 1: density not determined by depth or temperature

```
y = -3 + rnorm(nrow(dat),0,0.1)

dat$cpue_kg_km2 = y

# scale temperature before passing into sdmTMB
scaled_d = dat
scaled_d$temp = as.numeric(scale(dat$temp))
scaled_d$depth = as.numeric(scale(log(dat$depth)))

spde = make_mesh(scaled_d, xy_cols = c("longitude","latitude"),
                 cutoff=25)
    # use PC prior for matern model
    priors = sdmTMBpriors(
      matern_s = pc_matern(
        range_gt = 5, range_prob = 0.05,
        sigma_lt = 25, sigma_prob = 0.05
      )
    )
# note intercept is included at the end
fit1 = sdmTMB(cpue_kg_km2 ~ -1 + temp + I(temp^2) + depth + I(depth^2) + as.factor(year),
       spde = spde,
       data = scaled_d,
       time="year",
       priors=priors,
      control = sdmTMBcontrol(quadratic_roots = TRUE),
       spatial_only = FALSE)

sd_report <- summary(fit1$sd_report)

params <- as.data.frame(sd_report[grep("quadratic", row.names(sd_report)), ])
#knitr::kable(params, digits=2)
knitr::kable(tidy(fit1), digits=3)
```

term	estimate	std.error
temp	-0.008	0.005
I(temp^2)	0.005	0.002
depth	-0.007	0.005
I(depth^2)	-0.004	0.003
as.factor(year)2013	-3.005	0.005
as.factor(year)2014	-3.003	0.004
as.factor(year)2015	-3.005	0.005
as.factor(year)2016	-3.004	0.004
as.factor(year)2017	-3.006	0.004
as.factor(year)2018	-2.998	0.004

This model has a hard time converging, but the parameter estimates for the quadratic effect of temperature indicate that estimates are close to 0. The quadratic min/max values (vertices) are also centered on 0, which is an indicator of no relationship.

Example 2: density only determined by temperature

Here we simulate CPUE using a quadratic form. After standardizing both temperature and depth, we include both as quadratic predictors in the model. We also include spatiotemporal and spatial variation.

```
y = -3 + 2*dat$temp - 0.12*dat$temp*dat$temp + rnorm(nrow(dat),0,0.1)

dat$cpue_kg_km2 = y

# scale temperature before passing into sdmTMB
scaled_d = dat
scaled_d$temp = as.numeric(scale(dat$temp))
scaled_d$depth = as.numeric(scale(log(dat$depth)))

spde = make_mesh(scaled_d, xy_cols = c("longitude","latitude"),
                 cutoff=25)
# note intercept is included at the end
fit2 = sdmTMB(cpue_kg_km2 ~ -1 + temp + I(temp^2) + depth + I(depth^2) + as.factor(year),
       spde = spde,
       data = scaled_d,
       priors=priors,
      control = sdmTMBcontrol(quadratic_roots = TRUE),
       spatial_only = FALSE,
       time="year")

sd_report <- summary(fit2$sd_report)

params <- as.data.frame(sd_report[grep("quadratic", row.names(sd_report)), ])
#knitr::kable(params, digits=2)
knitr::kable(tidy(fit2), digits=3)
```

term	estimate	std.error
temp	0.703	0.005
I(temp^2)	-0.492	0.002
depth	-0.003	0.005
I(depth^2)	-0.005	0.003
as.factor(year)2013	5.074	0.005
as.factor(year)2014	5.086	0.005
as.factor(year)2015	5.082	0.005
as.factor(year)2016	5.087	0.004
as.factor(year)2017	5.086	0.005
as.factor(year)2018	5.087	0.005

In the simulated data, we assumed temperature had a positive effect up to a point,

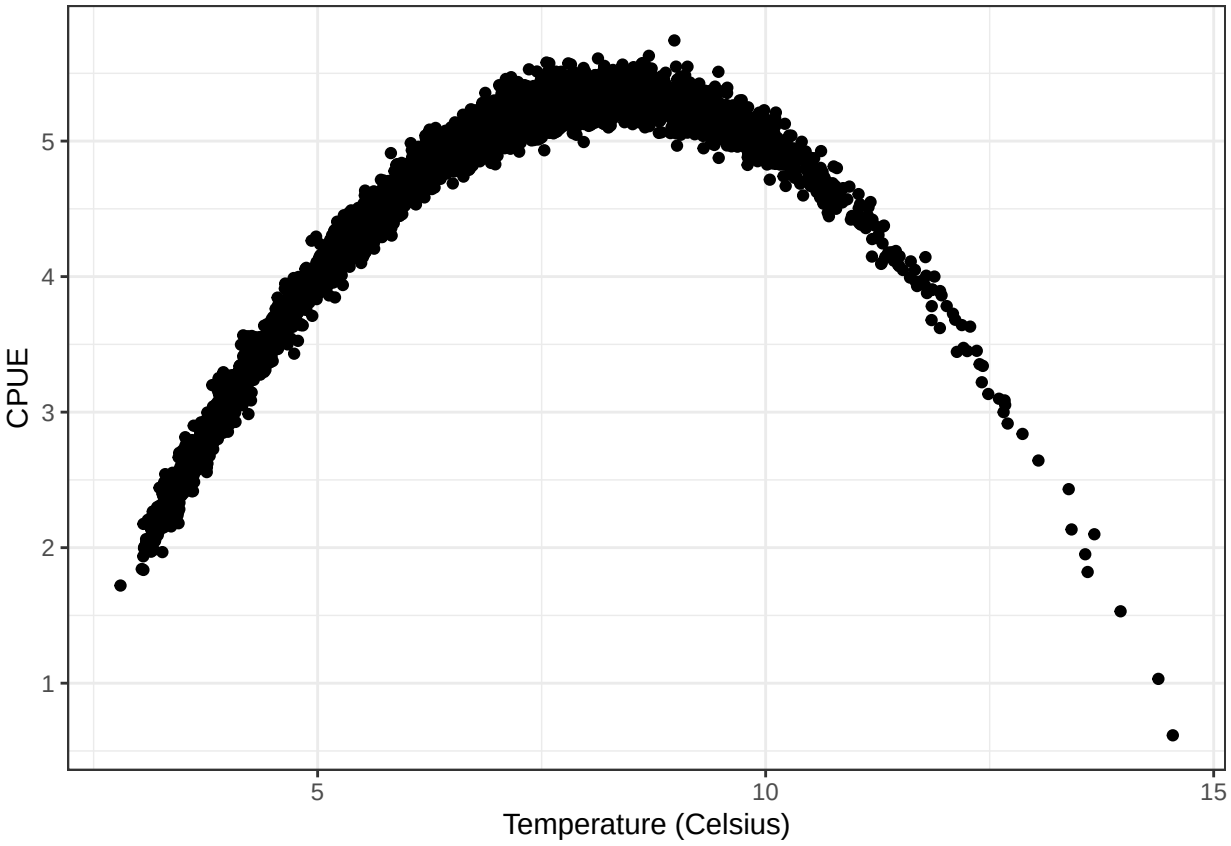
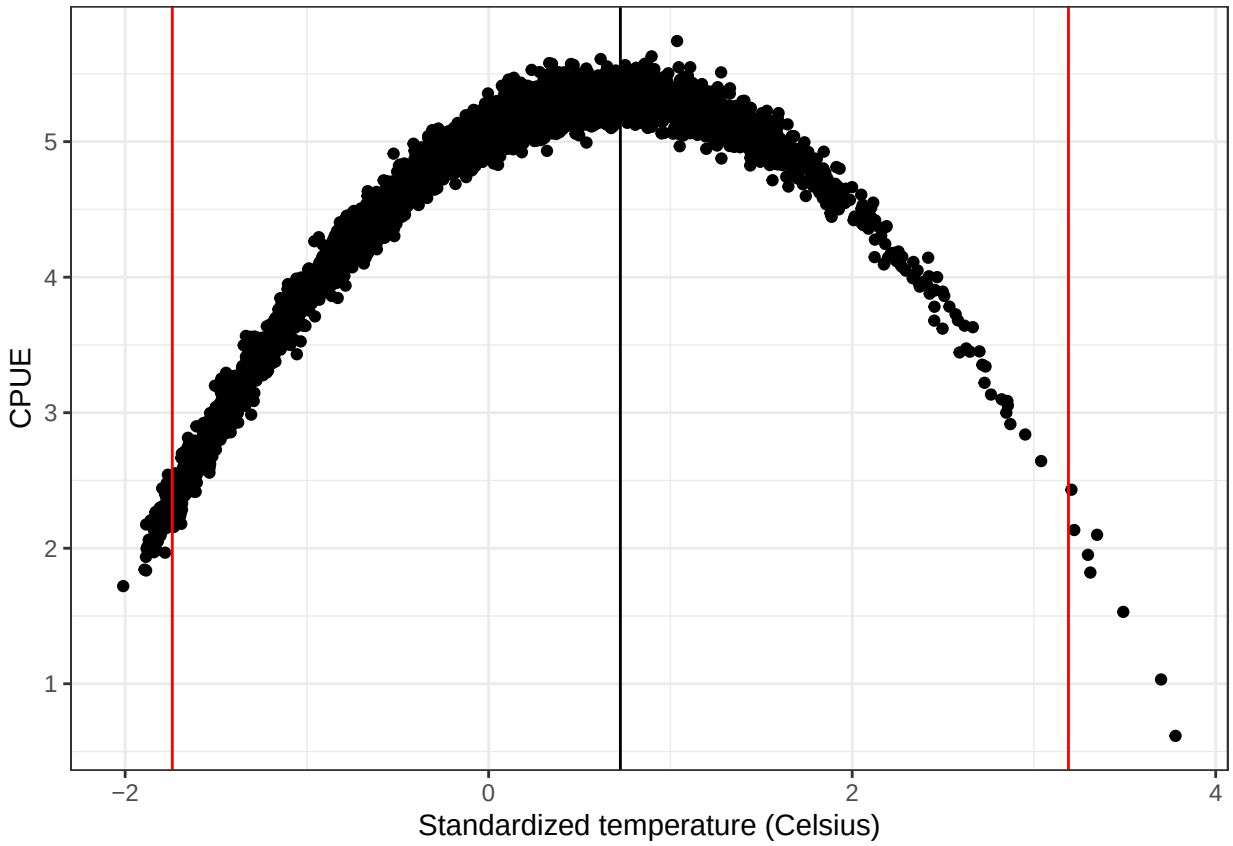


Figure 3: Simulated relationship between CPUE and temperature (raw)



Example 3: density determined by temperature and depth

Here we add to the previous example by including quadratic effects of both depth and temperature. The temperature effect is coded the same as above, and the depth effect is modeled as an increasing function (abundance increases with depth)

```
y = -3 + 2*dat$temp - 0.12*dat$temp*dat$temp + rnorm(nrow(dat),0,0.1) +
  -0.0000002*dat$depth + 0.000002*dat$depth*dat$depth

dat$cpue_kg_km2 = y

# scale temperature before passing into sdmTMB
scaled_d = dat
scaled_d$temp = as.numeric(scale(dat$temp))
scaled_d$depth = as.numeric(scale(log(dat$depth)))

spde = make_mesh(scaled_d, xy_cols = c("longitude","latitude"),
                 cutoff=25)
# note intercept is included at the end
fit3 = sdmTMB(cpue_kg_km2 ~ -1 + temp + I(temp^2) + depth + I(depth^2) + as.factor(year),
       spde = spde,
       data = scaled_d,
       time="year",
       priors=priors,
      control = sdmTMBcontrol(quadratic_roots = TRUE),
       spatial_only = FALSE)
```

```
sd_report <- summary(fit3$sd_report)

params <- as.data.frame(sd_report[grep("quadratic", row.names(sd_report)), ])
#knitr::kable(params, digits=2)
knitr::kable(tidy(fit3), digits=3)
```

term	estimate	std.error
temp	0.625	0.017
I(temp^2)	-0.440	0.006
depth	0.504	0.016
I(depth^2)	0.371	0.010
as.factor(year)2013	5.224	0.574
as.factor(year)2014	5.240	0.574
as.factor(year)2015	5.237	0.574
as.factor(year)2016	5.235	0.574
as.factor(year)2017	5.241	0.574
as.factor(year)2018	5.246	0.574

Example 4: confounded density determined by temperature and depth

To try to make this as difficult as possible, we can simulate data with nearly identical quadratic relationships for temperature and depth,

```
scaled_d = dat
scaled_d$temp = as.numeric(scale(dat$temp))
scaled_d$depth = as.numeric(scale(log(dat$depth)))

y = 5 + 0.65*scaled_d$temp - 0.45*scaled_d$temp*scaled_d$temp + rnorm(nrow(dat),0,0.1) +
  0.5*scaled_d$depth -0.4*scaled_d$depth*scaled_d$depth

dat$cpue_kg_km2 = y

# scale temperature before passing into sdmTMB
scaled_d = dat
scaled_d$temp = as.numeric(scale(dat$temp))
scaled_d$depth = as.numeric(scale(log(dat$depth)))

spde = make_mesh(scaled_d, xy_cols = c("longitude","latitude"),
                 cutoff=25)
# note intercept is included at the end
fit4 = sdmTMB(cpue_kg_km2 ~ -1 + temp + I(temp^2) + depth + I(depth^2) + as.factor(year),
       spde = spde,
       data = scaled_d,
       priors=priors,
      control = sdmTMBcontrol(quadratic_roots = TRUE),
       spatial_only = FALSE,
       time="year")

sd_report <- summary(fit4$sd_report)

params <- as.data.frame(sd_report[grep("quadratic", row.names(sd_report)), ])
#knitr::kable(params, digits=2)
knitr::kable(tidy(fit4), digits=3)
```

term	estimate	std.error
temp	0.649	0.005
I(temp^2)	-0.450	0.002
depth	0.500	0.004
I(depth^2)	-0.398	0.003
as.factor(year)2013	4.999	0.005
as.factor(year)2014	4.996	0.004
as.factor(year)2015	4.998	0.004
as.factor(year)2016	4.999	0.004
as.factor(year)2017	4.995	0.004
as.factor(year)2018	4.998	0.004