

# smoothSDE Analysis of Atlantic BRS Ziphius DTAG Data: smoothSDEs

Stacy DeRuiter

2023-2024

## Background

### Methods Paragraph

Whale depth and heading data from the DTAGs were decimated to 1/3 Hz (after first applying a no-delay low-pass anti-alias filter).

- *Option: For depth, we computed the first difference of the depth measurements to reduce temporal autocorrelation in the time-series before modeling.*

Dives to greater than 50 meters depth were detected, the classified as deep, presumed foraging dives (dives to over 800m depth) or shallower dives (less than 800m) (Shearer et. al. ref). Each dive was also classified as exposed if there were any controlled MFAS exposure sounds during that dive, and baseline if it occurred before any controlled exposure. We fitted four varying-coefficient stochastic differential equations (SDEs) (Michelot et al. 2022, 2023): for heading and depth-difference data, in shallow and deep dives respectively. Each model describes the heading or depth as a function of proportion of dive-time elapsed, and employs a difference smooth to quantify differences between baseline and exposed dives. The model also includes a random effect to account for whale-to-whale differences, which also allows for visualization of estimated individual responses.

Data processing used the package tagtools (DeRuiter et al. 2023), and models were fitted using the package smoothSDE (Michelot 2021, 2022, 2023).

DeRuiter S, Johnson M, Sweeney D, McNamara-Oh Y, Fynewever S, Tejevbo O, Marques T, Wang Y, Ogedegbe O. (2023). tagtools: Work with Data from High-Resolution Biologging Tags. R package version 0.1.0, <https://CRAN.R-project.org/package=tagtools>.

Michelot, T., Glennie, R., Harris, C., Thomas, L. (2021) “Varying-coefficient stochastic differential equations with applications in ecology” Journal of Agricultural, Biological, and Environmental Statistics. DOI: 10.1007/s13253-021-00450-6.

Michelot, Théo, et al. “Continuous-time modelling of behavioural responses in animal movement.” arXiv preprint arXiv:2212.09574 (2022).

Michelot T (2023). smoothSDE: Varying-coefficient stochastic differential equations. R package version 0.1.

## Preparation

### Read in and prepare data

The data has undergone additional pre-processing in Matlab. See files “prep-SDEs.Rmd” and “smoothSDE\_prep.m” for details.

Data from a given dive will be considered exposed to the CEE if any part of that dive overlapped with the CEE.

The real MFAS dataset is included in the data prep steps and initial graphs, but is not analyzed with the smoothSDE model for now.

```
abrs_zc <- read_csv('data/AtlBRS_Zc_smoothSDE_data.csv',
                      show_col_types = FALSE) |>
  # filter out times before/after first/last dive
  drop_na(diveprop, max_depth) |>
  # filter out post-dive surface intervals
  dplyr::filter(diveprop <= 1) |>
  # append "dive-" to dive ID numbers
  mutate(time = time_hr # we want to keep hr for clarity of units but smoothSDE requires time var
        ) |>
  rename(depth = p_lo,
         pitch = pitch_lo,
         roll = roll_lo,
         heading = head_lo_deg
        ) |>
  # re-create new ID variable so each whale-dive has a unique identifier
  mutate(ID = paste0('dive-', as.numeric(interaction(whaleID, ID, drop = TRUE))))
```

## CEEs

Add in information on CEEs.

Since audits are not available for all tag deployments, this info is not all easily accessible directly from the data files. Note also some of the tag deployment dates are wrong on quicklook doc files, so the dates used here have been validated by checking the Julian-day numbers in the tag ID string and, where needed, the .xml cal files.

Info used is:

```
# cee_meta_from_will <- read_csv('https://raw.githubusercontent.com/atlanticbrs/zcss_data_dev/main/00_d
CEE_meta <- tibble(whaleID = c("zc17_234a",
                               "zc19_218a",
                               "zc20_232a",
                               "zc22_219a"),
                    tagon_time = lubridate::ymd_hms('2017 aug 22 15:10:46',
                                                   '2019 aug 6 15:38:09',
                                                   '2020 aug 19 15:28:50',
                                                   '2022 aug 7 13:56:55'),
                    # based on cee_metadata_flat
                    cee_start_time = lubridate::ymd_hms('2017 aug 22 18:41:00',
                                                       '2019 aug 6 17:37:43',
                                                       '2020 aug 19 17:43:23',
                                                       '2022 aug 7 17:09:00'),
                    cee_end_time = lubridate::ymd_hms('2017 aug 22 18:54:00',
                                                       '2019 aug 6 18:07:43',
                                                       '2020 aug 19 18:13:22',
                                                       '2022 aug 7 18:10:00'),
                    cee_type = c('scaled MFAS',
                               'scaled MFAS',
                               'scaled MFAS',
                               'real MFAS'))
```

```
)
```

Note that the 2017 whale only did one long and deep dive, which ended before the start of the CEE. So if this analysis is looking at deep dives only, then this whale has “no exposed” deep dives.

```
CEE_meta <- CEE_meta |>
  mutate(
    cee_start_hr = round(as.numeric(
      difftime(cee_start_time, tagon_time,
               units = 'hours')
    )),
    digits = 3),
    cee_end_hr = round(as.numeric(
      difftime(cee_end_time, tagon_time,
               units = 'hours')
    )),
    digits = 3)
  )

knitr::kable(CEE_meta)
```

whaleID	tagon_time	cee_start_time	cee_end_time	cee_type	cee_start_hr	cee_end_hr
zc17_234a	2017-08-22 15:10:46	2017-08-22 18:41:00	2017-08-22 18:54:00	scaled MFAS	3.504	3.721
zc19_218a	2019-08-06 15:38:09	2019-08-06 17:37:43	2019-08-06 18:07:43	scaled MFAS	1.993	2.493
zc20_232a	2020-08-19 15:28:50	2020-08-19 17:43:23	2020-08-19 18:13:22	scaled MFAS	2.243	2.742
zc22_219a	2022-08-07 13:56:55	2022-08-07 17:09:00	2022-08-07 18:10:00	real MFAS	3.201	4.218

```
abrs_zc <- left_join(abrs_zc, CEE_meta, by = 'whaleID') |>
  mutate(cee_status = case_when( time_hr < cee_start_hr ~ 'pre',
                                 time_hr >= cee_start_hr & time_hr <= cee_end_hr ~ 'during',
                                 time_hr > cee_end_hr ~ 'post'),
        cee_status = fct_relevel(cee_status, 'pre', 'during', 'post')) |>
  group_by(whaleID, ID) |>
  # this one classes each entire DIVE as during if any part of it was during
  mutate(cee_dive_status_pooled = ifelse(sum(time_hr >= cee_start_hr & time_hr <= cee_end_hr) > 1,
                                         'CEE',
                                         'pre or post'),
        cee_dive_status = ifelse(cee_dive_status_pooled == "CEE",
                               paste0(whaleID, cee_dive_status_pooled),
                               cee_dive_status_pooled)) |>
  ungroup()
```

## Echolocation

Add in information on the times of echolocation clicks.

```
whales <- unique(pull(abrs_zc, whaleID))
echolocation <- list()
```

```

for (w in c(1:length(whales))){
  click_files <- dir(file.path('data/echolocation', whales[w]))
  click_files <- file.path('data/echolocation', whales[w], click_files)
  for (f in c(1:length(click_files))){
    these_clicks <- R.matlab::readMat(click_files[f])
    # note: 1st list-item from readMat() is the data vector/matrix, later items are attributes.
    # the NAME of the data is different from file to file hence use of numeric index
    if (f == 1){
      echolocation[[w]] <- tibble(whaleID = whales[w],
                                    soc = min(these_clicks[[1]], na.rm = TRUE),
                                    eoc = max(these_clicks[[1]], na.rm = TRUE))
    }else{
      echolocation[[w]] <- bind_rows(echolocation[[w]],
                                      tibble(whaleID = whales[w],
                                             soc = min(these_clicks[[1]], na.rm = TRUE),
                                             eoc = max(these_clicks[[1]], na.rm = TRUE)))
    }
  }
}

echolocation <- bind_rows(echolocation) |>
  # get soc and eoc in hours
  mutate(soc_hr = soc / 3600,
         eoc_hr = eoc / 3600)

# next need to: put a binary variable clicking/not into the time series data for plotting
abrs_zc <- mutate(abrs_zc, clicking = 0, foraging_period = 0)
for (d in c(1:nrow(echolocation))){
  abrs_zc <- abrs_zc |>
    mutate(clicking = if_else(whaleID == pull(echolocation, whaleID)[d] &
                               time_hr >= pull(echolocation, soc_hr)[d] &
                               time_hr <= pull(echolocation, eoc_hr)[d],
                               1,
                               clicking),
           foraging_period = if_else(whaleID == pull(echolocation, whaleID)[d] &
                                       time_hr >= pull(echolocation, soc_hr)[d] &
                                       time_hr <= pull(echolocation, eoc_hr)[d],
                                       d,
                                       foraging_period))
}

abrs_zc <- mutate(abrs_zc,
                  soc = if_else(clicking - lag(clicking) == 1, 1, 0),
                  eoc = if_else(clicking - lag(clicking) == -1, 1, 0))

```

## Data Exploration/Verification

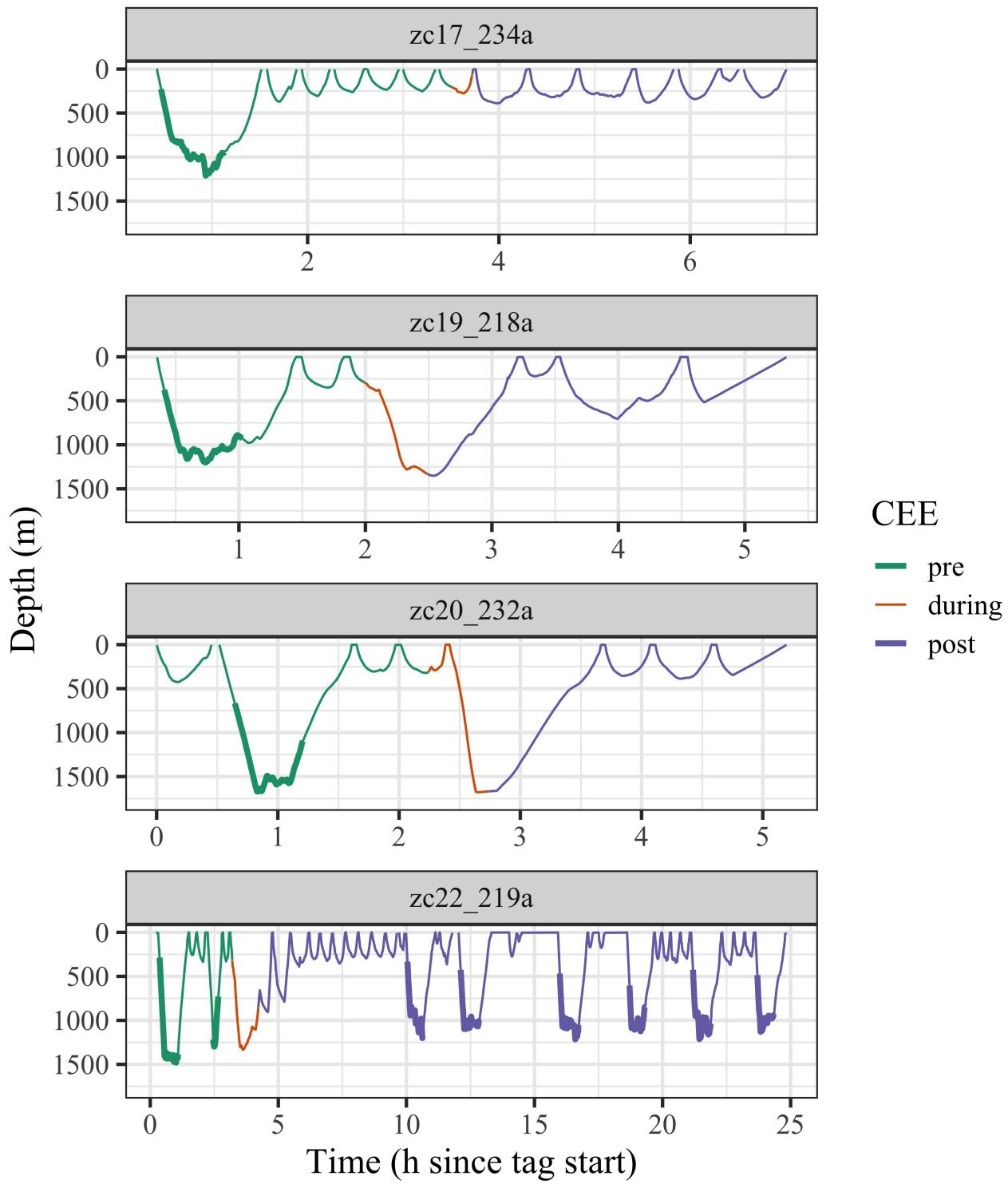
There was some question at some point if the CEE start/end times were correct, so these plots are shown so that the researchers who know the datasets well can verify that the assignment of pre/during/post exposure status is correct.

```

gf_line(depth ~ time_hr,
        color = ~cee_status,
        data = abrs_zc) |>

```

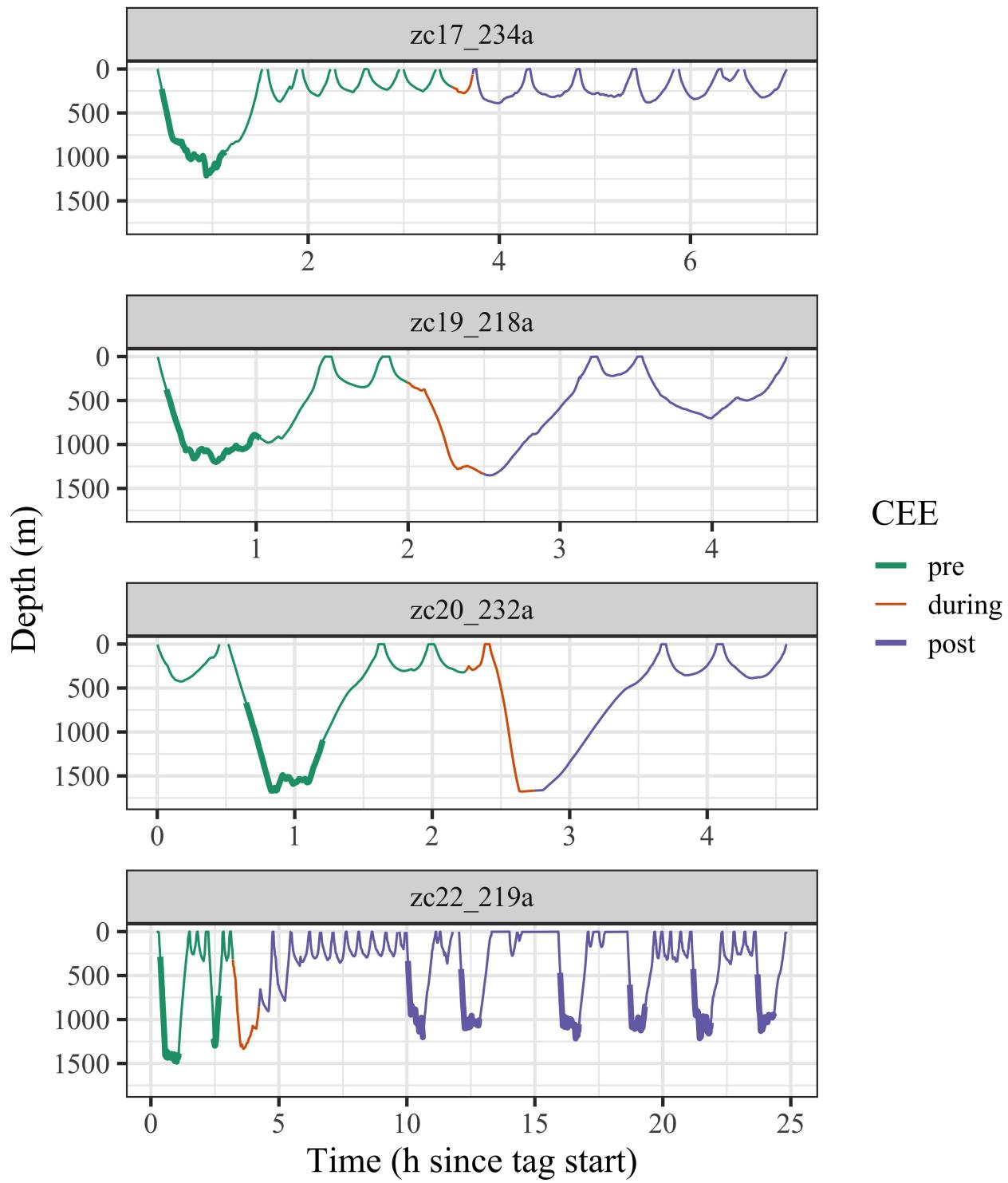
```
gf_path(depth ~ time_hr,
        color = ~cee_status,
        group = ~foraging_period,
        data = abrs_zc |> filter(clicking == 1),
        linewidth = 1.25) |>
gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
gf_theme(scale_color_brewer('CEE', palette = 'Dark2')) |>
gf_lims(y = c(1800, 0)) |>
gf_labs(x = 'Time (h since tag start)', y = 'Depth (m)')
```



At this point we can see that the last shallow dives by `zc20_232a` and `zc19_218a` both have the tag fall off before the end of the dive. So these two shallow dives should be removed from analysis.

```
abrs_zc <- abrs_zc |>
  group_by(whaleID) |>
  filter(whaleID == "zc17_234a" |
    (whaleID == "zc19_218a" & ID != last(ID)) |
```

```
(whaleID == "zc20_232a" & ID != last(ID)) |  
(whaleID == "zc22_219a")  
) |>  
ungroup()  
  
gf_line(depth ~ time_hr,  
        color = ~cee_status,  
        data = abrs_zc) |>  
gf_path(depth ~ time_hr,  
        color = ~cee_status,  
        group = ~foraging_period,  
        data = abrs_zc |> filter(clicking == 1),  
        linewidth = 1.25) |>  
gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>  
gf_theme(scale_color_brewer('CEE', palette = 'Dark2')) |>  
gf_lims(y = c(1800, 0)) |>  
gf_labs(x = 'Time (h since tag start)', y = 'Depth (m)')
```

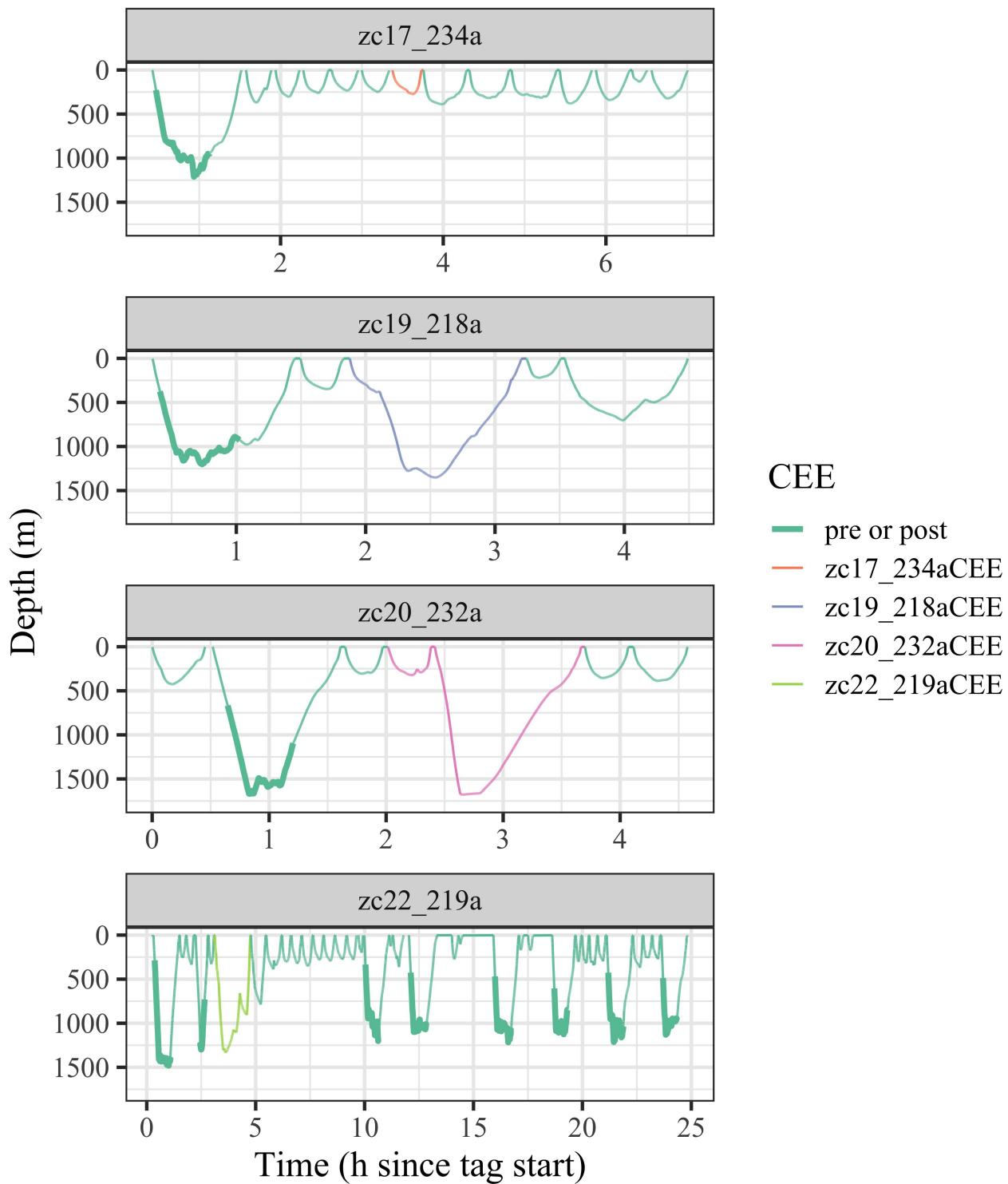


```
gf_line(depth ~ time_hr,
        color = ~cee_dive_status,
        group = ~1,
        data = abrs_zc) |>
gf_path(depth ~ time_hr,
        color = ~cee_dive_status,
        group = ~foraging_period,
```

```

data = abrs_zc |> filter(clicking == 1),
lineWidth = 1.25) |>
gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
gf_lims(y = c(1800, 0)) |>
gf_labs(x = 'Time (h since tag start)', y = 'Depth (m)')

```



## Split Deep and Shallow

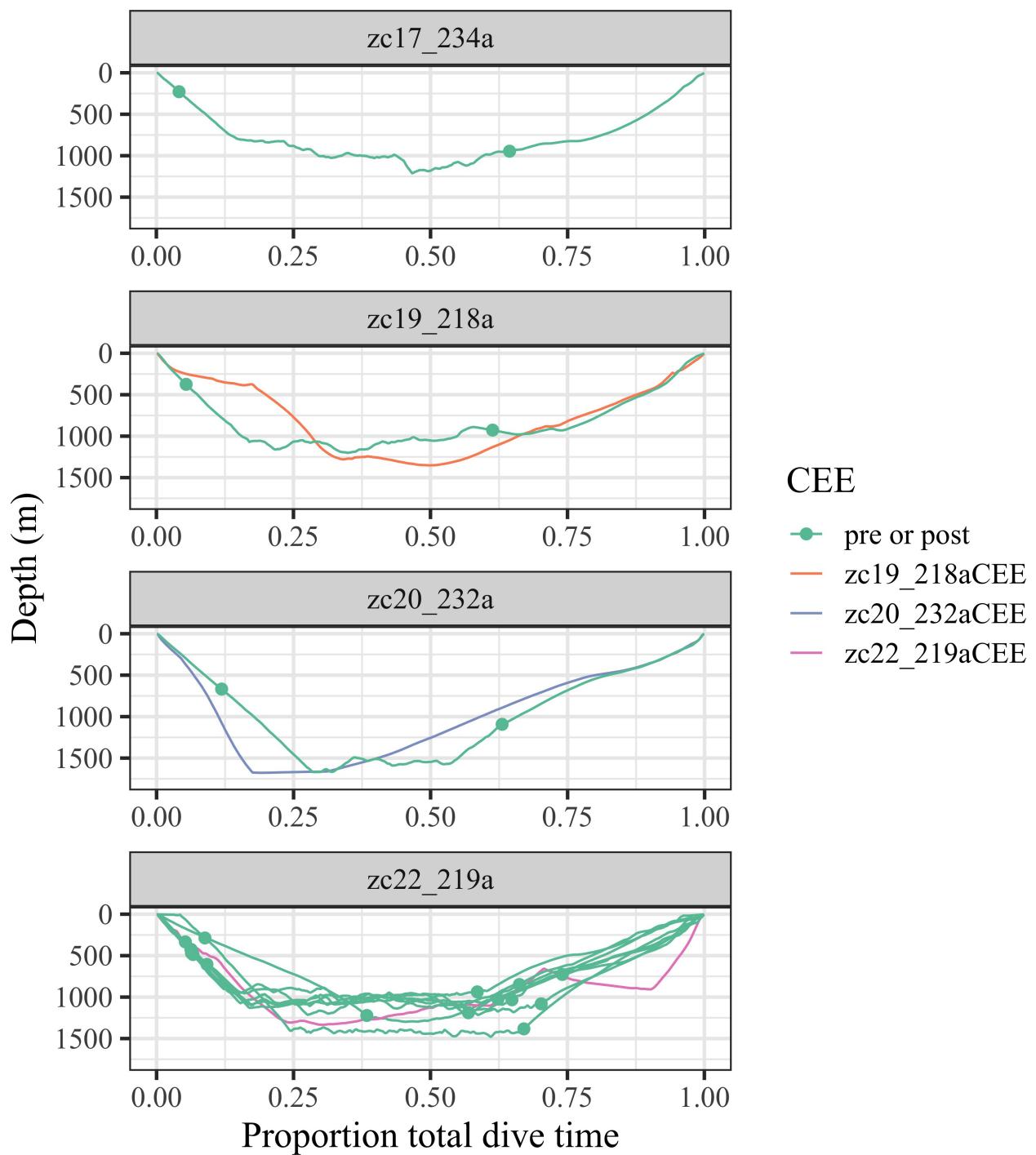
We divide the data into two separate datasets, with deep and shallow dives. For shallow dives, discard all post-exposure dives (they are not to be considered baseline). For deep dives, post-exposure dives are also effectively discarded because there were no post-exposure deep dives in the three whales under study here.

```
abrs_zc_deep <- abrs_zc |>
  filter(max_depth >= 800)

abrs_zc_shallow <- abrs_zc |>
  filter(max_depth < 800 & cee_status != "post")

gf_line(depth ~ diveprop,
        color = ~cee_dive_status,
        group = ~ID,
        data = abrs_zc_deep) |>
gf_point(depth ~ diveprop,
          color = ~cee_dive_status,
          group = ~ID,
          data = abrs_zc_deep |> filter(soc == 1 | eoc == 1),
          size = 2) |>
gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
gf_lims(y = c(1800, 0)) |>
gf_labs(x = 'Proportion total dive time', y = 'Depth (m)',
        title = 'Deep Dives')
```

## Deep Dives



Note that almost all of the non-exposed deep dives are post-exposure dives in the zc\_22 tagout with the real MFAS exposure. The other three tagouts each have one pre-exposure dive.

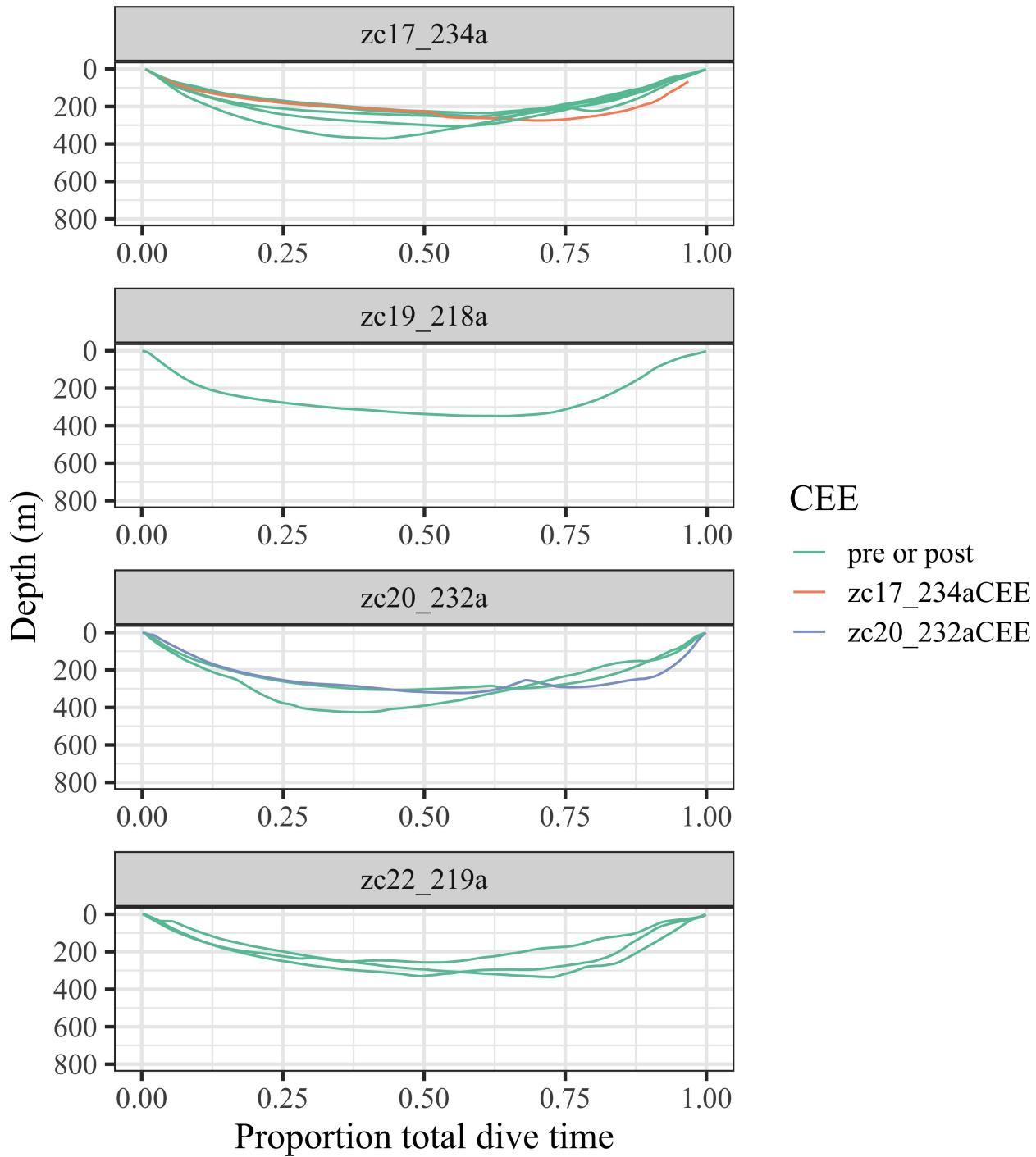
```
gf_line(depth ~ diveprop,
        color = ~cee_dive_status,
        group = ~ID,
        data = abrs_zc_shallow) |>
```

```

gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
  gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
  gf_lims(y = c(800, 0)) |>
  gf_labs(x = 'Proportion total dive time', y = 'Depth (m)',
    title = 'Shallow Dives')

```

## Shallow Dives



## Final data prep

The real MFAS deployment, zc22\_219a, is removed from analysis for now.

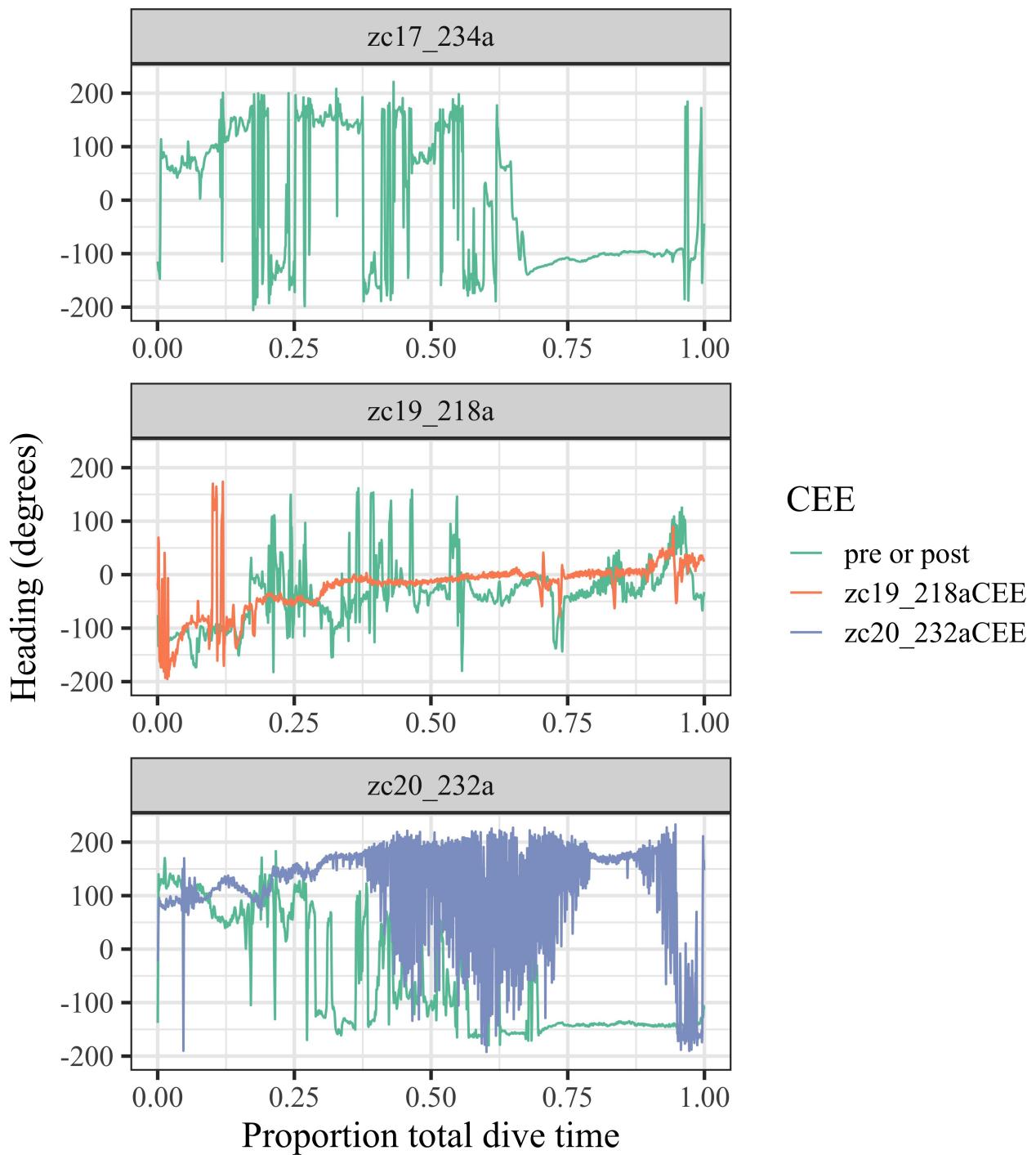
Before fitting a model with the `smoothSDE` package we must also make sure that the data is a `data.frame()` (not a `tibble()`) and that all categorical variables are factors, not character.

```
abrs_zc_deep <- abrs_zc_deep |>
  dplyr::filter(whaleID != 'zc22_219a') |>
  group_by(whaleID, ID) |>
  mutate(diff_depth = c(0, diff(depth)),
         head_rad = heading / 180 * pi,
         diff_head = atan2(sin(head_rad - lag(head_rad, n = 1)),
                           cos(head_rad - lag(head_rad, n = 1)))) |>
  ungroup() |>
  dplyr::select(whaleID, ID, time, diveprop,
                cee_dive_status, depth, diff_depth, heading, diff_head,
                soc, eoc, clicking) |>
  as.data.frame() |>
  mutate(across(where(is_character), as_factor)) |>
  mutate(cee_dive_status = factor(cee_dive_status, ordered = TRUE))

abrs_zc_shallow <- abrs_zc_shallow |>
  dplyr::filter(whaleID != 'zc22_219a') |>
  group_by(whaleID, ID) |>
  mutate(diff_depth = c(0, diff(depth)),
         head_rad = heading / 180 * pi,
         diff_head = atan2(sin(head_rad - lag(head_rad, n = 1)),
                           cos(head_rad - lag(head_rad, n = 1)))) |>
  ungroup() |>
  dplyr::select(whaleID, ID, time, diveprop, cee_dive_status, depth, diff_depth, heading, diff_head) |>
  as.data.frame() |>
  mutate(across(where(is_character), as_factor)) |>
  mutate(cee_dive_status = factor(cee_dive_status, ordered = TRUE))

gf_line(heading ~ diveprop,
        color = ~cee_dive_status,
        group = ~ID,
        data = abrs_zc_deep) |>
  gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
  gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
  gf_labs(x = 'Proportion total dive time', y = 'Heading (degrees)',
           title = 'Deep Dives')
```

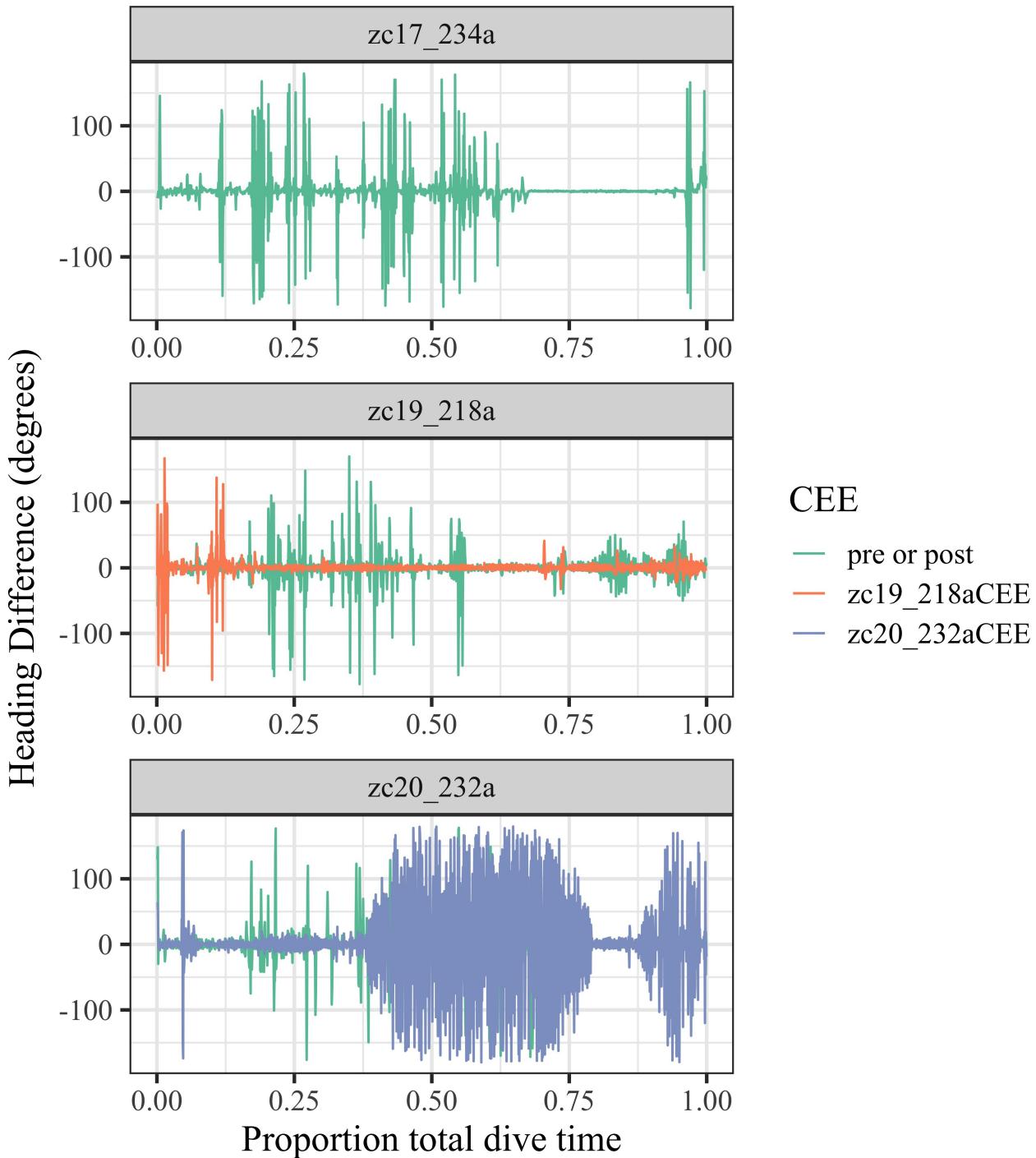
## Deep Dives



```
gf_line(diff_head / pi * 180 ~ diveprop,
        color = ~cee_dive_status,
        group = ~ID,
        data = abrs_zc_deep) |>
  gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
  gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
  gf_labs(x = 'Proportion total dive time', y = 'Heading Difference (degrees)',
```

```
title = 'Deep Dives')
```

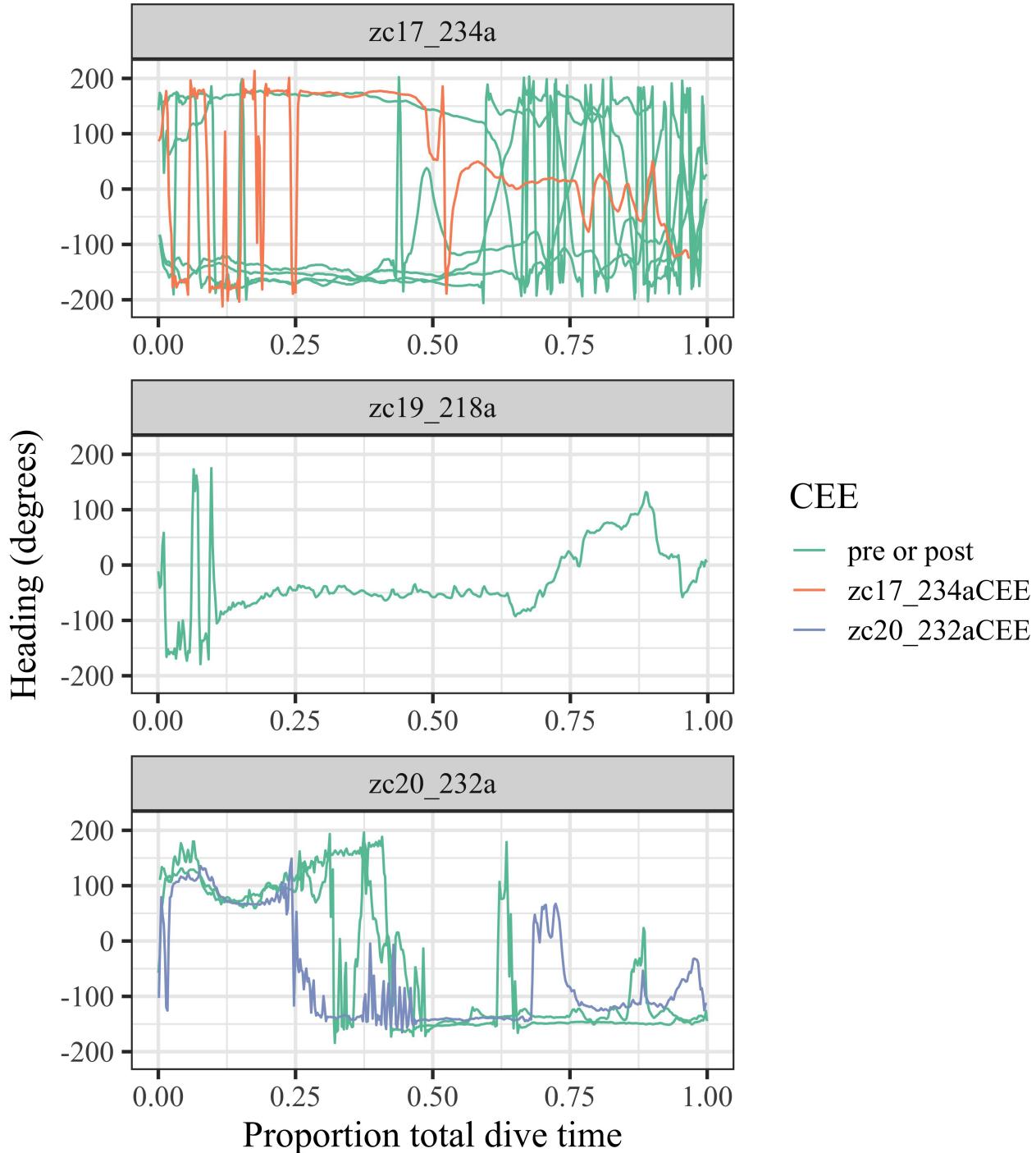
## Deep Dives



```
gf_line(heading ~ diveprop,  
        color = ~cee_dive_status,  
        group = ~ID,  
        data = abrs_zc_shallow) |>  
gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
```

```
gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
  gf_labs(x = 'Proportion total dive time', y = 'Heading (degrees)',
    title = 'Shallow Dives')
```

## Shallow Dives



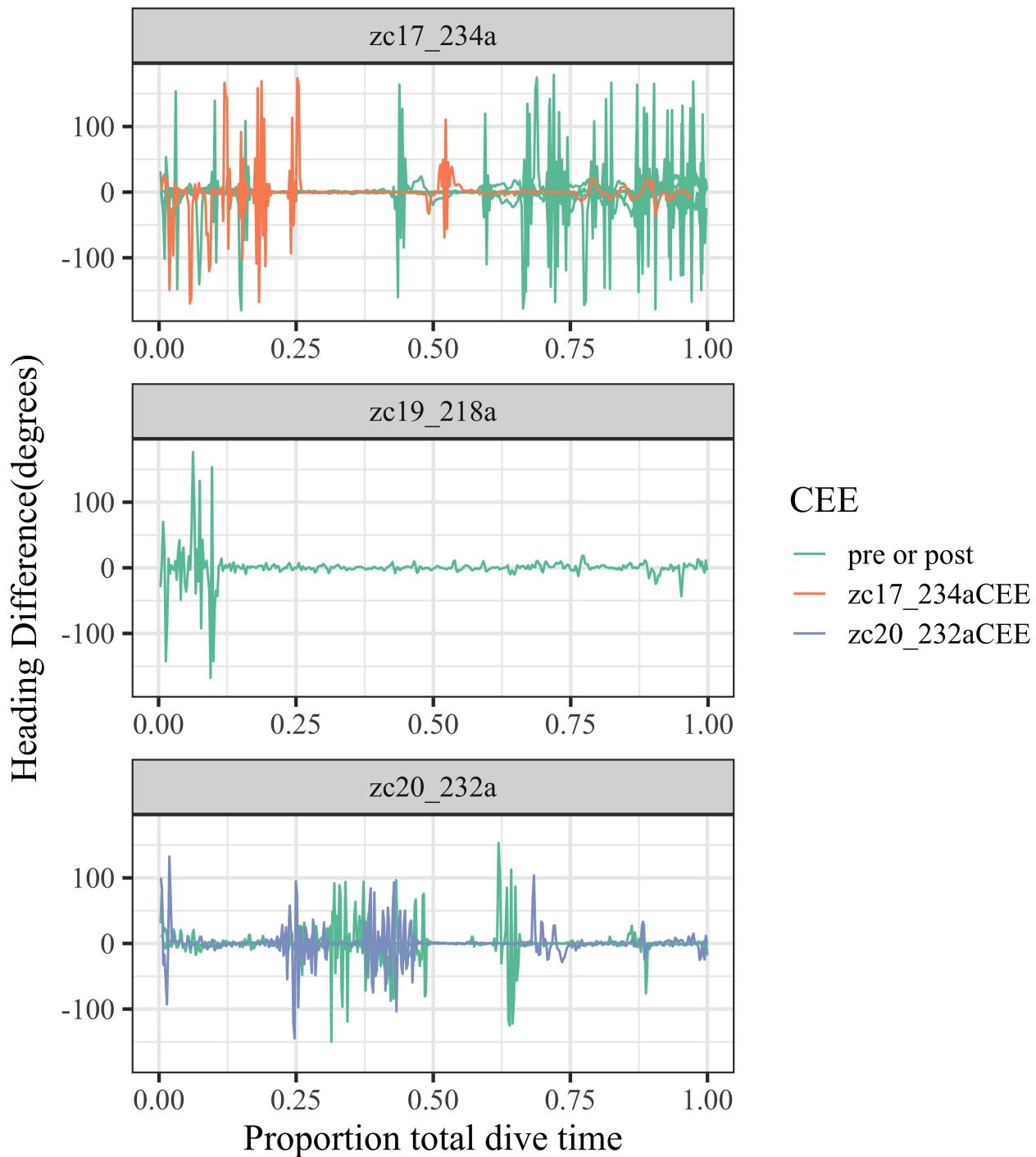
```
gf_line(diff_head / pi * 180 ~ diveprop,
  color = ~cee_dive_status,
  group = ~ID,
```

```

    data = abrs_zc_shallow) |>
gf_facet_wrap(~whaleID, scales = 'free_x', nrow = 4) |>
gf_theme(scale_color_brewer('CEE', palette = 'Set2')) |>
gf_labs(x = 'Proportion total dive time', y = 'Heading Difference(degrees)', title = 'Shallow Dives')

```

## Shallow Dives



CEE

## Models: Deep Dives

### Depth data

To reduce the amount of temporal autocorrelation in the time series data, we model the first difference of the depths (`diff_depth`) instead of the original depth data.

We could include a difference smooth either on the drift (directionality) or on the diffusion (variability) parameter for the depth data.

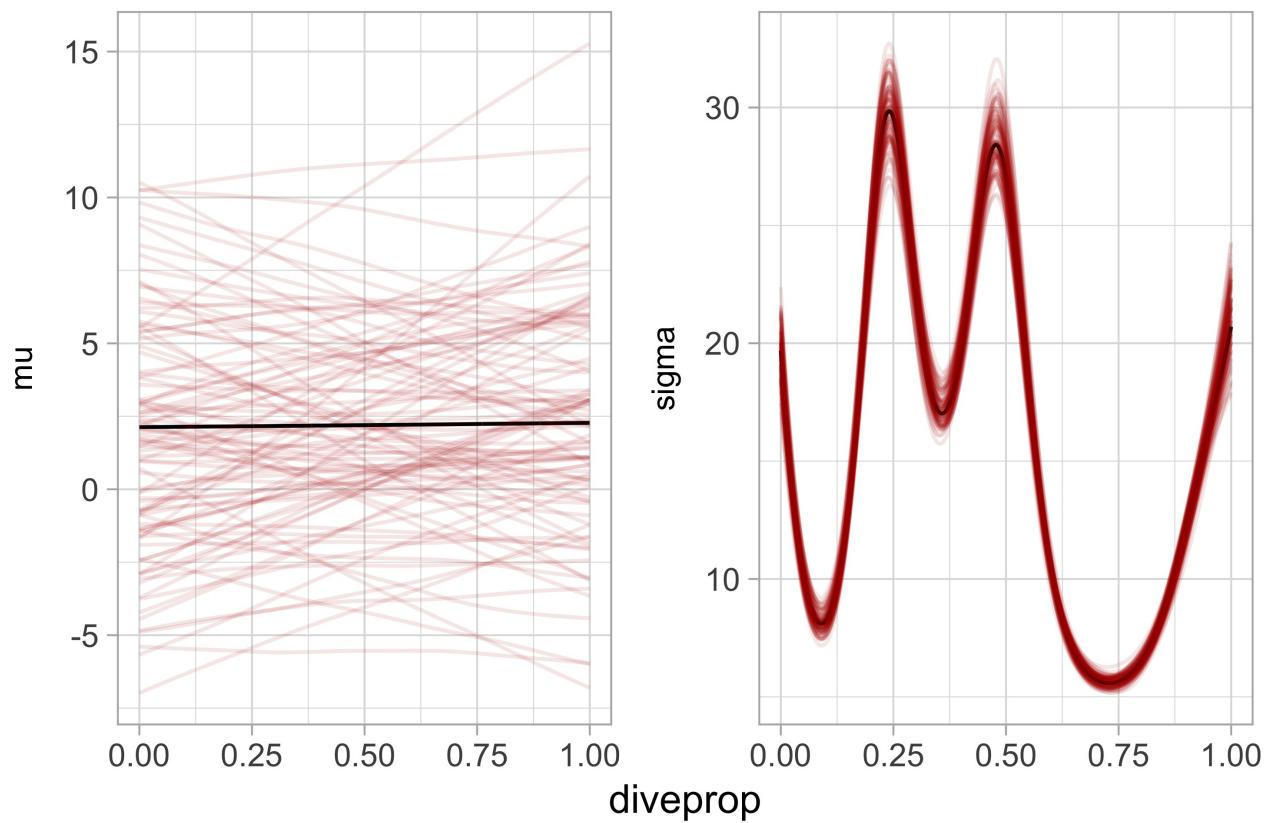
Here, we include the response on the diffusion parameter.

```
r1file <- 'models/deep_ddepth.RDS'
if (file.exists(r1file)){
  deep_ddepth_sde <- readRDS(r1file)
} else{
  f_deep_ddepth <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts"),
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"))
  # Create and fit SDE
  deep_ddepth_sde <- SDE$new(formulas = f_deep_ddepth,
    data = abrs_zc_deep,
    type = "BM",
    response = "diff_depth")
  deep_ddepth_sde$fit(silent = TRUE)
  saveRDS(deep_ddepth_sde, r1file)
}

r1file <- 'models/deep_ddeptht.RDS'
if (file.exists(r1file)){
  deep_ddeptht_sde <- readRDS(r1file)
} else{
  f_deep_ddeptht <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts"),
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"))
  # Create and fit SDE
  deep_ddeptht_sde <- SDE$new(formulas = f_deep_ddeptht,
    data = abrs_zc_deep,
    type = "BM_t",
    other_data = list(df = 3),
    response = "diff_depth")
  deep_ddeptht_sde$fit(silent = TRUE)
  saveRDS(deep_ddeptht_sde, r1file)
}

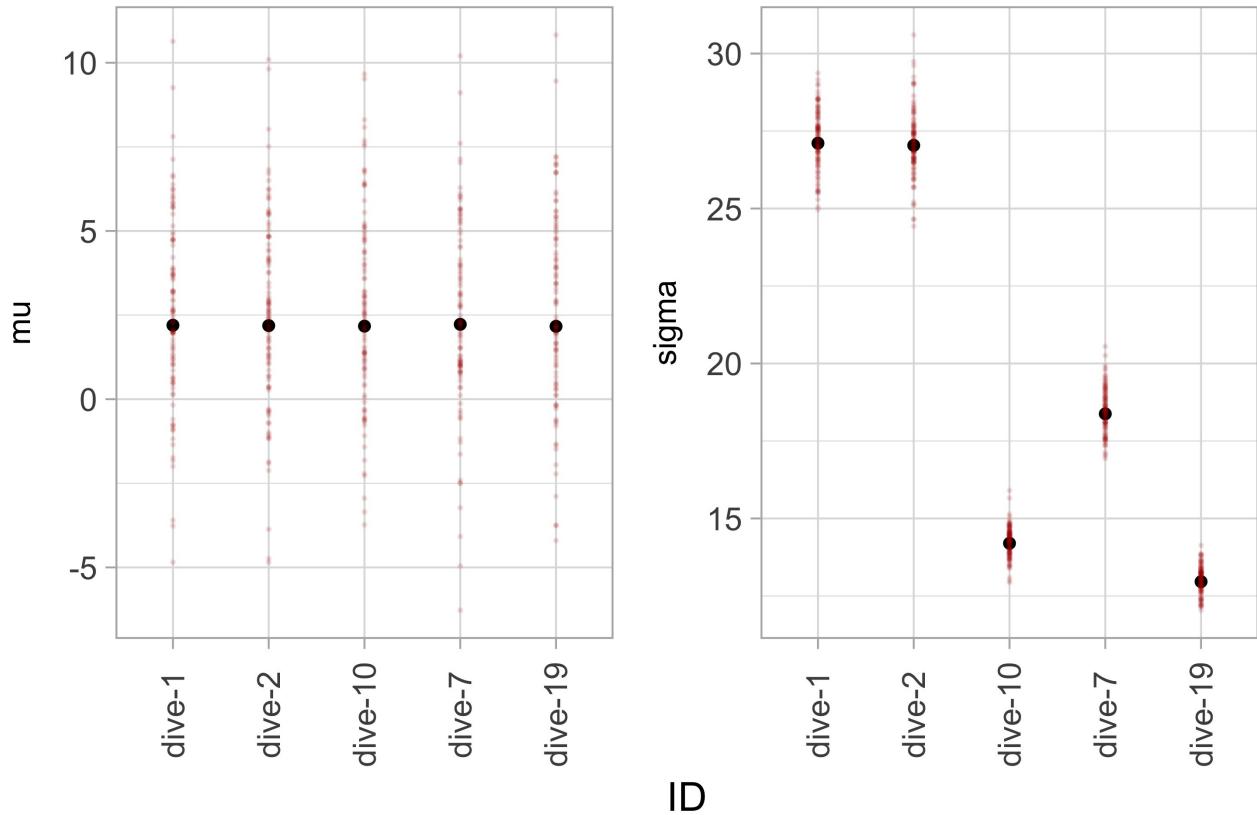
deep_ddepth_sde$plot_par("diveprop")
```

ID = dive-1, cee\_dive\_status = pre or post



```
deep_ddepth_sde$plot_par("ID")
```

`diveprop = 0.5, cee_dive_status = pre or post`



We can plot the difference smooth for  $\sigma$  on the linear predictor scale, together with 95% confidence intervals, as follows.

These plots show the change in the diffusion parameter sigma during CEE compared to baseline. Sigma measures the diffusion or variability of the response variable (either depth difference or heading). So a positive change means additional variation and negative mean less variability.

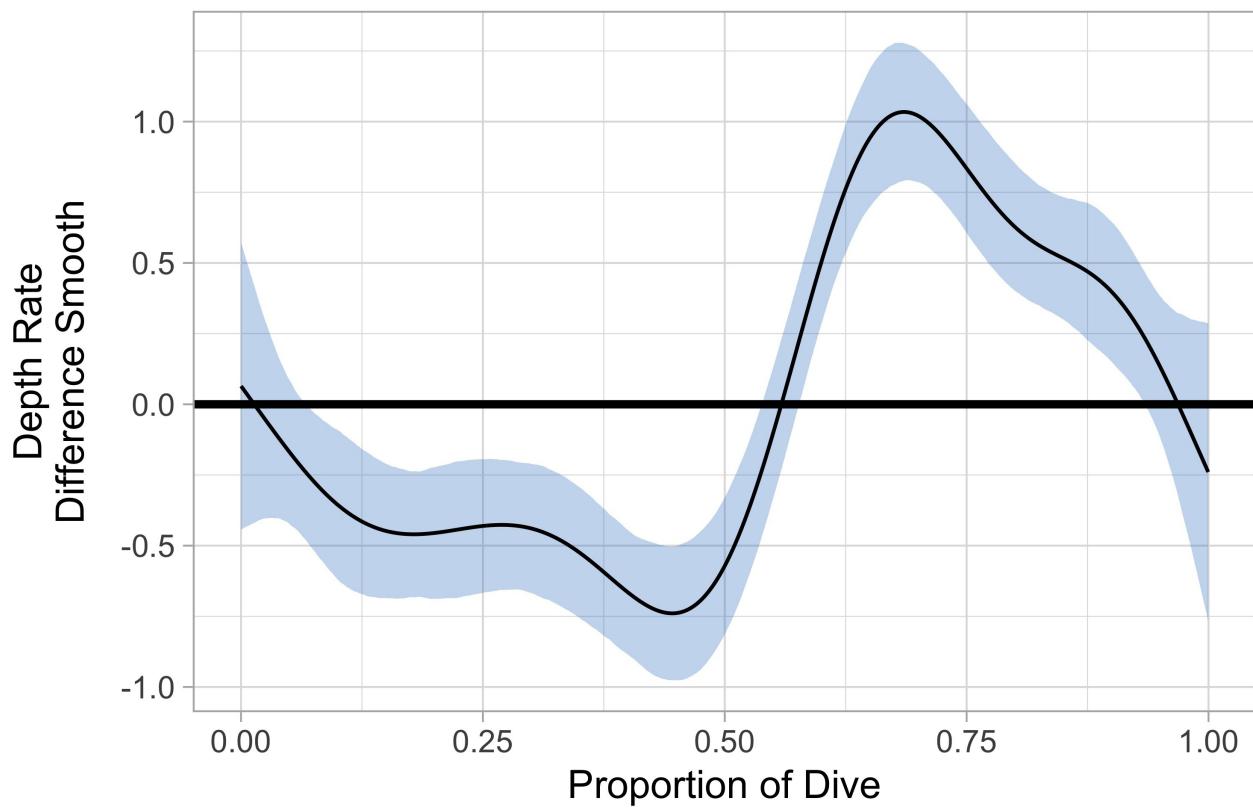
The numeric values for change in sigma are on the scale of the linear predictor, so can be thought of as being in some kind of relative units (note I don't have time/tools to present these on the response scale of meters and degrees instead - I'm not sure enough of the model equation and CI determination code being used to be sure I can just apply an inverse link function to the displayed values).

```
# Plot difference smooth on sigma for exposed dives

deep_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
                         covs = list(cee_dive_status = "zc19_218aCEE"), term = "cee_dive_status",
                         resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc19_218', y = 'Depth Rate\nDifference Smooth', x = 'Proportion of Dive') |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

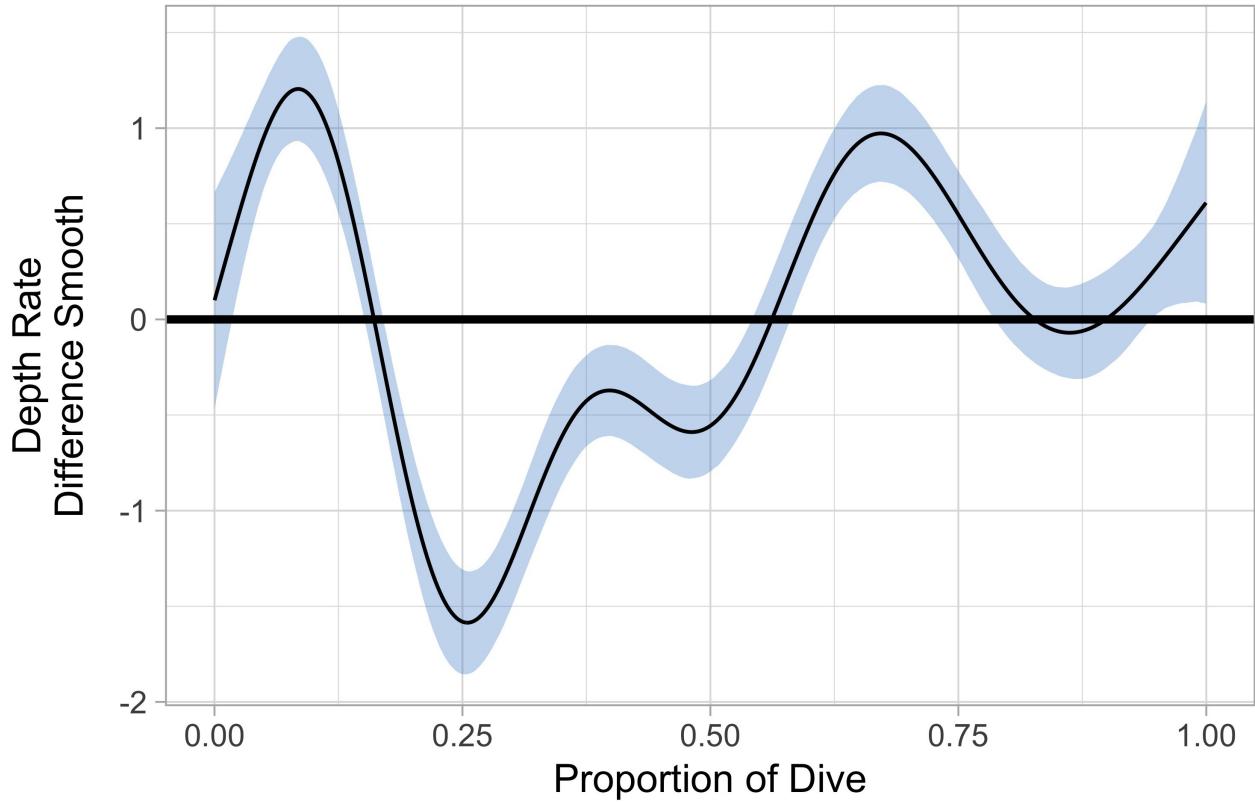
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

## zc19\_218



```
deep_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
                           cova = list(cee_dive_status = "zc20_232aCEE"), term = "cee_dive_status",
                           resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc20_232', y = 'Depth Rate\\nDifference Smooth', x = 'Proportion of Dive') |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)
```

## zc20\_232



### Heading data

Should we be modeling heading? Or difference in heading? Michelot et al just used angles with no worries...but perhaps they got lucky. In our case, either one (heading or diff(heading)) includes many "circular moves" (large changes and/or heading going from for example 350 to 10 degrees).

```
deep_head_sde_file <- 'models/deep_head.RDS'
if (file.exists(deep_head_sde_file)){
  deep_head_sde <- readRDS(deep_head_sde_file)
} else{
  deep_head_f <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"),
  sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"))
  # Create and fit SDE
  deep_head_sde <- SDE$new(formulas = deep_head_f,
    data = abrs_zc_deep,
    type = "BM",
    response = "heading")
  deep_head_sde$fit(silent = TRUE)
  saveRDS(deep_head_sde, deep_head_sde_file)
}

deep_head_tsde_file <- 'models/deep_head_t.RDS'
if (file.exists(deep_head_tsde_file)){
```

```

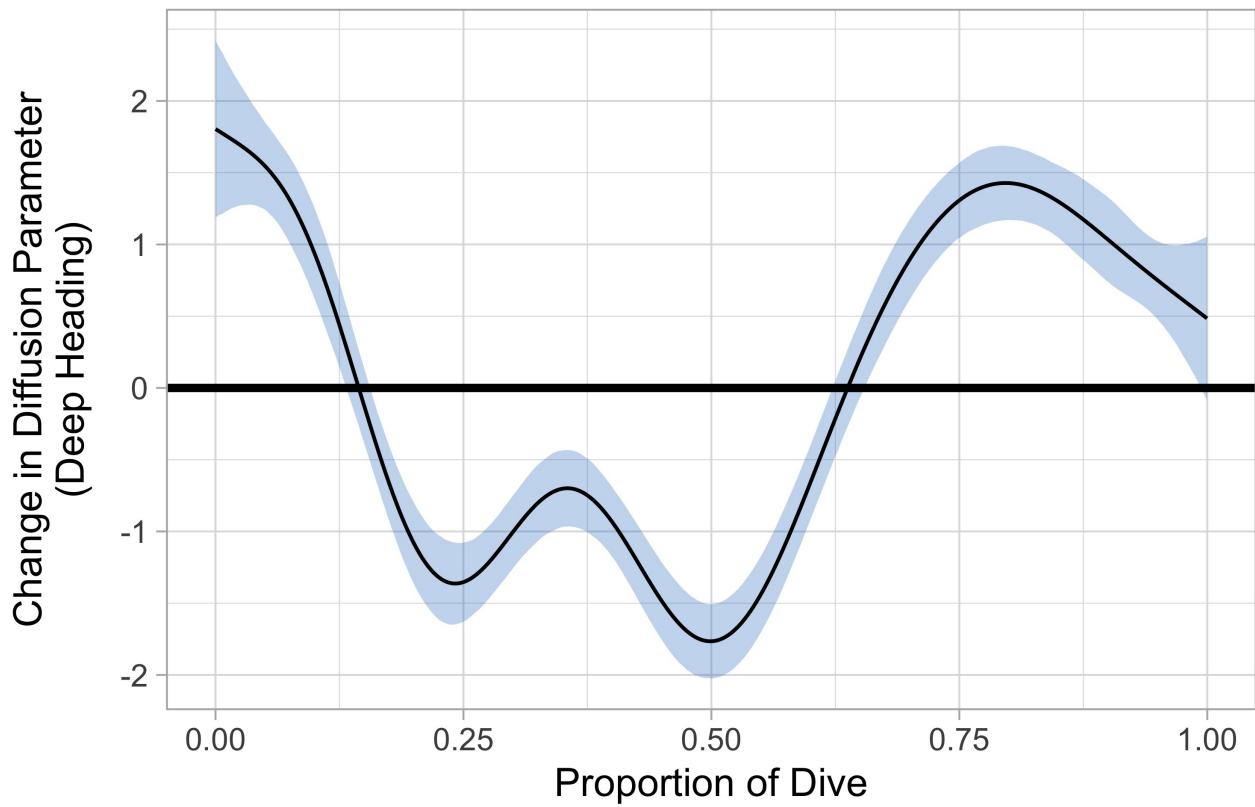
deep_head_tsde <- readRDS(deep_head_tsde_file)
} else{
  deep_head_f <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts")
    ,
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts")
  )
# Create and fit SDE
deep_head_tsde <- SDE$new(formulas = deep_head_f,
  data = abrs_zc_deep,
  type = "BM_t",
  response = "heading",
  other_data = list(df = 3)
)
deep_head_tsde$fit(silent = TRUE)
saveRDS(deep_head_tsde, deep_head_tsde_file)
}

# Plot difference smooth on sigma for exposed dives

deep_head_tsde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
  covs = list(cee_dive_status = "zc19_218aCEE"), term = "cee_dive_status",
  resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc19_218', y = 'Change in Diffusion Parameter\n(Deep Heading)', x = 'Proportion of Dives')
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

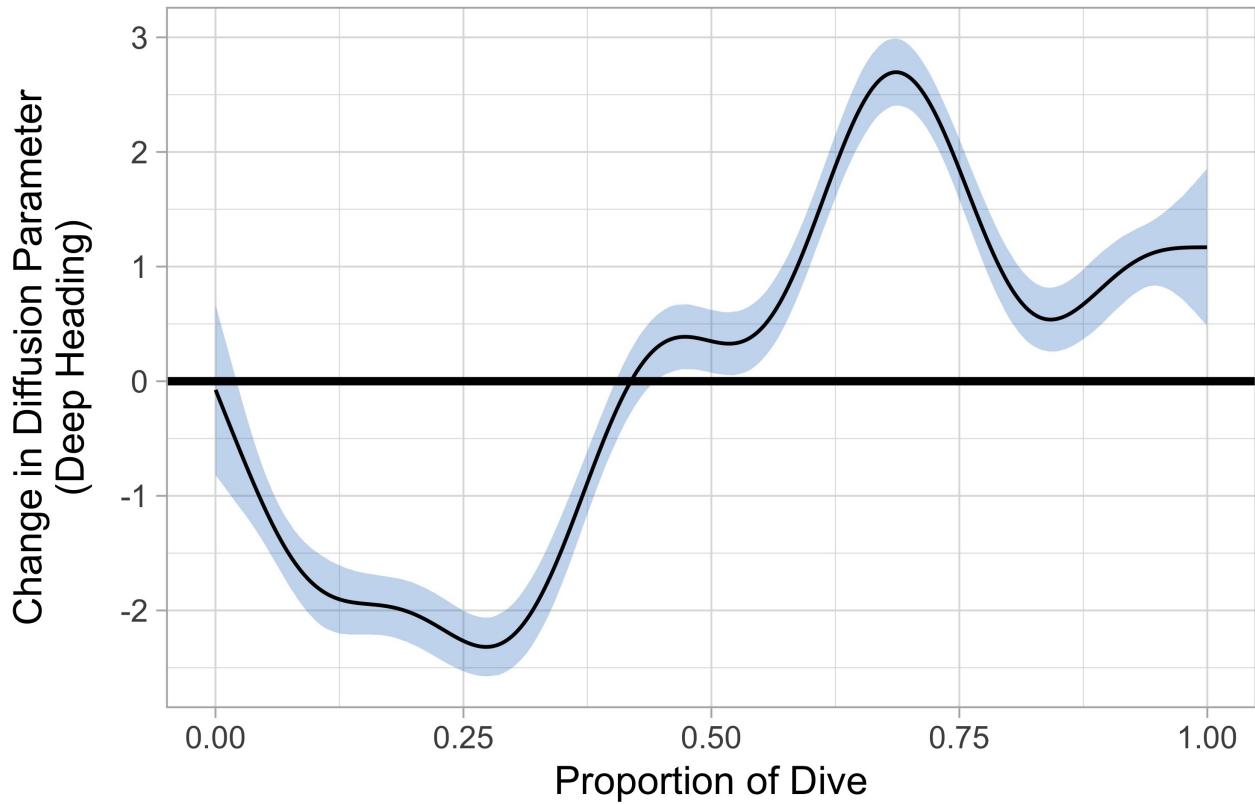
```

## zc19\_218



```
deep_head_tsde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
                        covs = list(cee_dive_status = "zc20_232aCEE"), term = "cee_dive_status",
                        resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc20_232', y = 'Change in Diffusion Parameter\n(Deep Heading)', x = 'Proportion of D')
```

## zc20\_232



Model validation for deep models

```
residz <- sapply(list(deep_ddepth_sde,
                      deep_ddeptht_sde,
                      deep_head_sde,
                      deep_head_tsde
                     ),
                  function(sde) sde$residuals()) |>
  as.data.frame()

names(residz) <- c('Depth difference',
                   'Depth difference (t)',
                   'Heading',
                   'Heading (t)'
                  )

residz <- residz |>
  mutate(ID = pull(abrs_zc_deep, ID))

residz <- pivot_longer(residz, cols = `Depth difference`:`Heading (t)` ,
                       names_to = 'variable',
                       values_to = 'residuals')

gf_dhistogram(~residuals, data = residz |>
```

```

        filter(variable == "Depth difference")) |>
gf_dist(dist = 'norm',
        # df = 3,
        color = 'grey44') |>
# here t would be WORSE
# gf_dist(dist = 't',
#         df = 3,
#         color = 'grey77') |>
gf_lims(x = c(-5, 5)) |>
gf_labs(title = 'Deep Depth Difference')

```

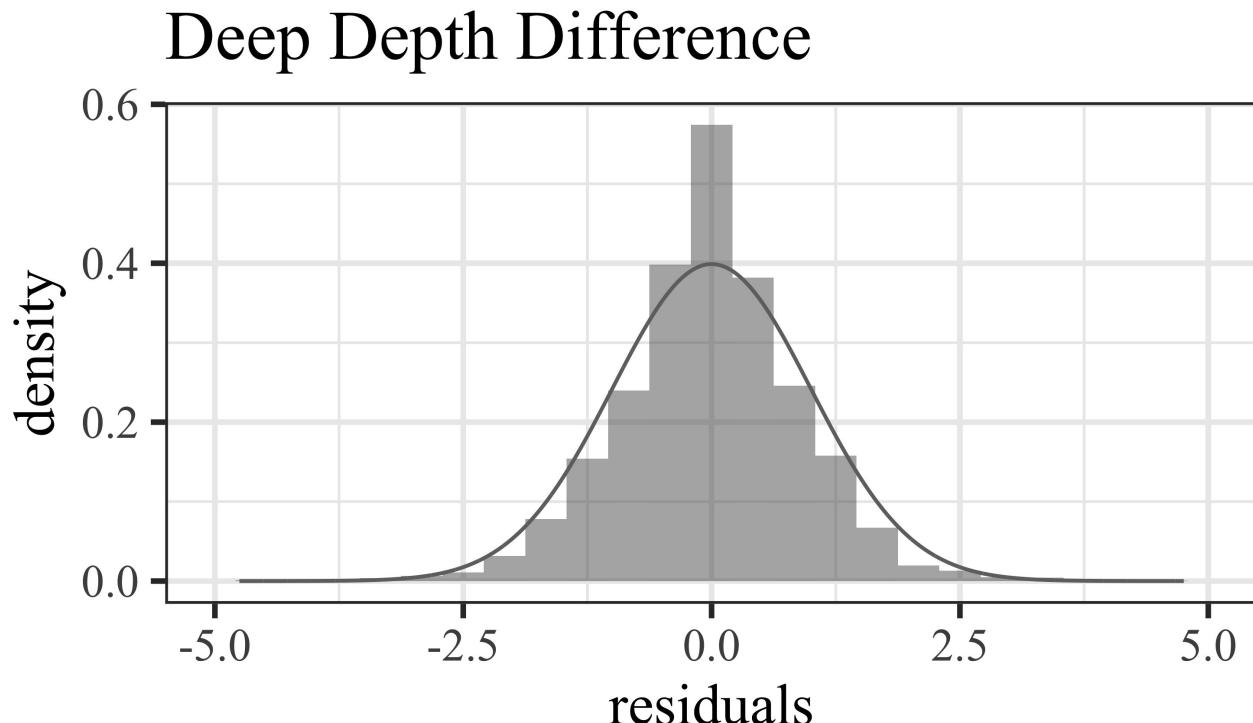
### Residual distribution

```

## Warning: Removed 23 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).

```



```

gf_dhistogram(~residuals, data = residz |>
                filter(variable == "Depth difference (t)") |>
gf_dist(dist = 't',
        df = 3,
        color = 'grey44') |>
gf_lims(x = c(-5, 5)) |>
gf_labs(title = 'Deep Depth Difference')

```

```

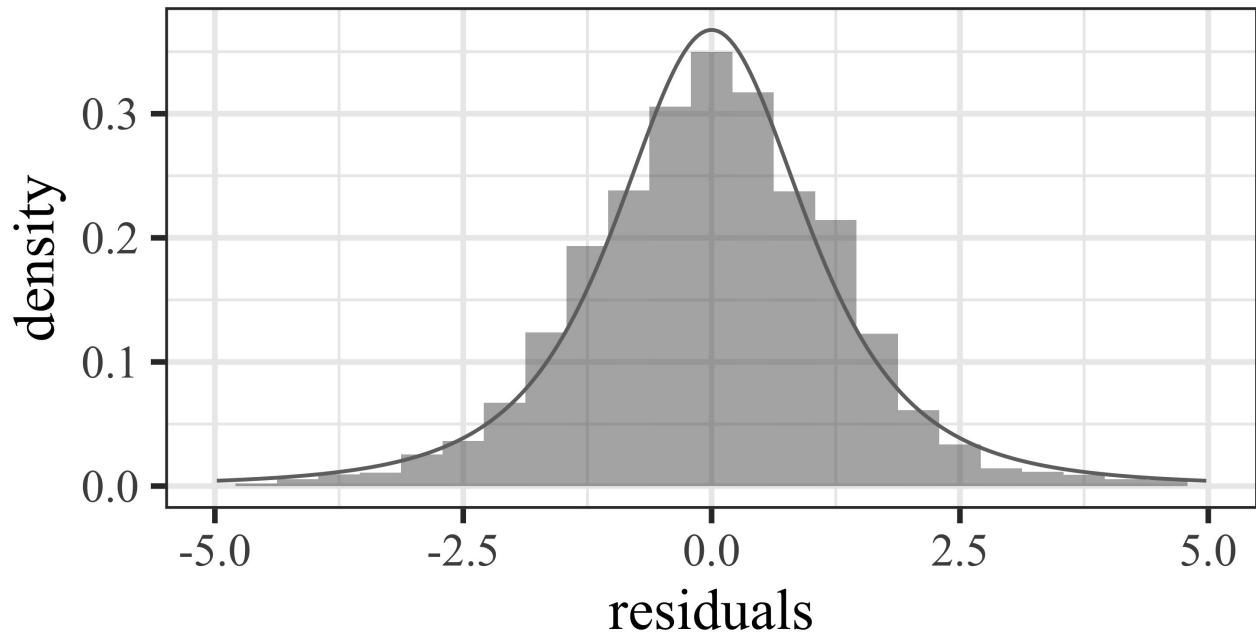
## Warning: Removed 91 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).

```

```
## Warning: Removed 4758 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

# Deep Depth Difference

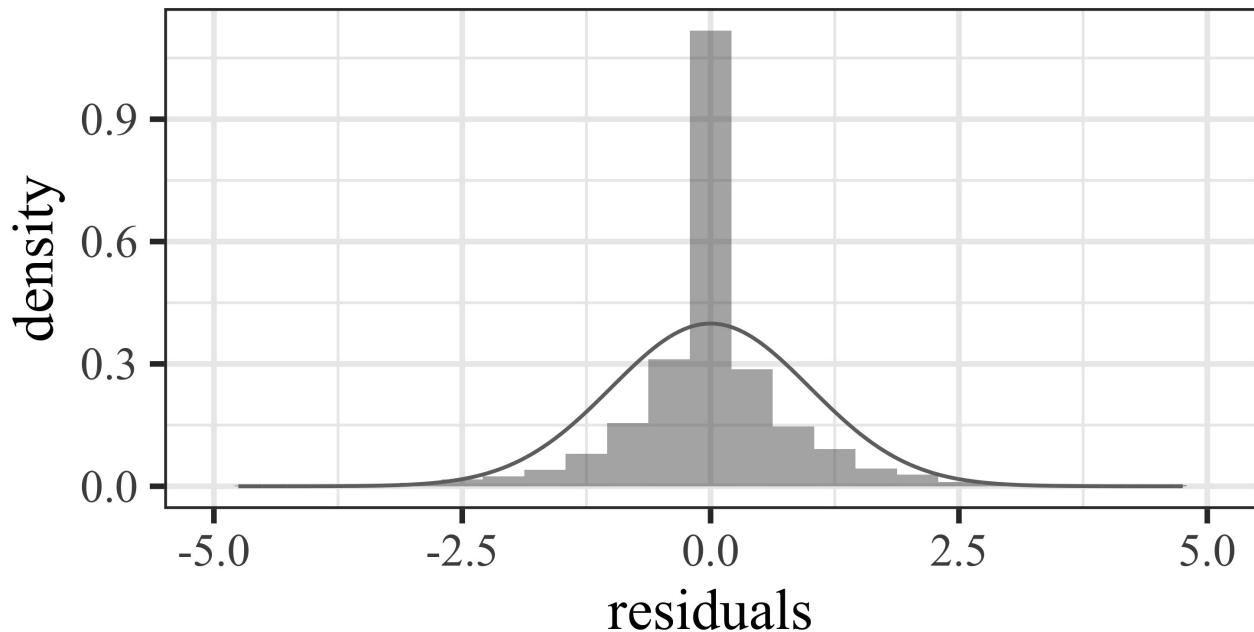


```
gf_dhistogram(~residuals, data = residz |>
  filter(variable == "Heading")) |>
  gf_dist(dist = 'norm',
  color = 'grey44') |>
  gf_lims(x = c(-5, 5)) |>
  gf_labs(title = 'Deep Heading')
```

```
## Warning: Removed 45 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```

# Deep Heading



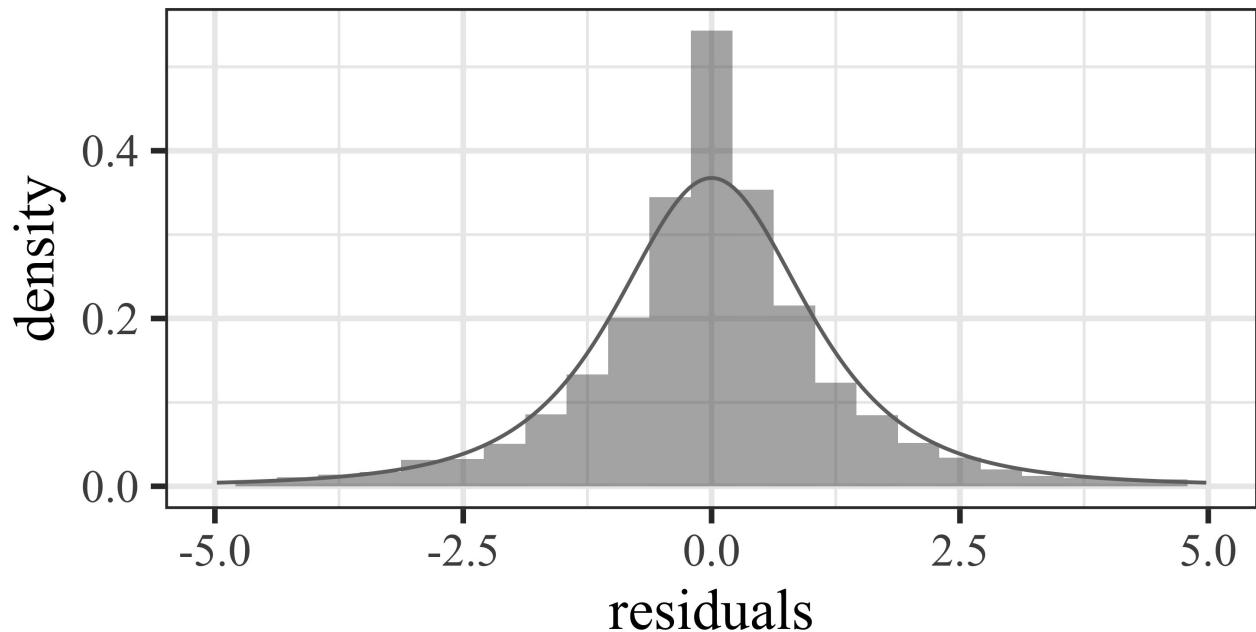
```
gf_dhistogram(~residuals, data = residz |>
  filter(variable == "Heading (t)") |>
  gf_dist(dist = 't',
    df = 3,
    color = 'grey44') |>
  gf_lims(x = c(-5, 5)) |>
  gf_labs(title = 'Deep Heading (t)')

## Warning: Removed 381 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).

## Warning: Removed 4758 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

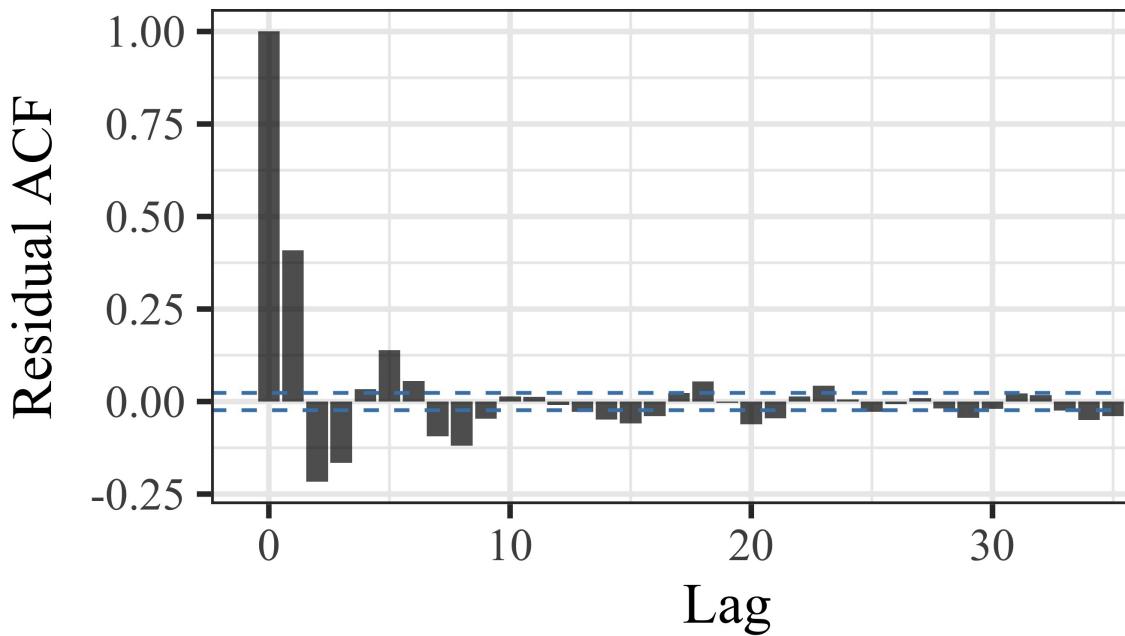
## Deep Heading (t)



Here, the t model for the heading data looks better, and the normal model for the depth difference model looks ok.

```
CalvinBayes::acf_plot(filter(residz,
                             variable == "Depth difference") |>
  pull(residuals) |>
  na.omit()) |>
  gf_labs(title = "Depth Difference", y = 'Residual ACF', x = 'Lag')
```

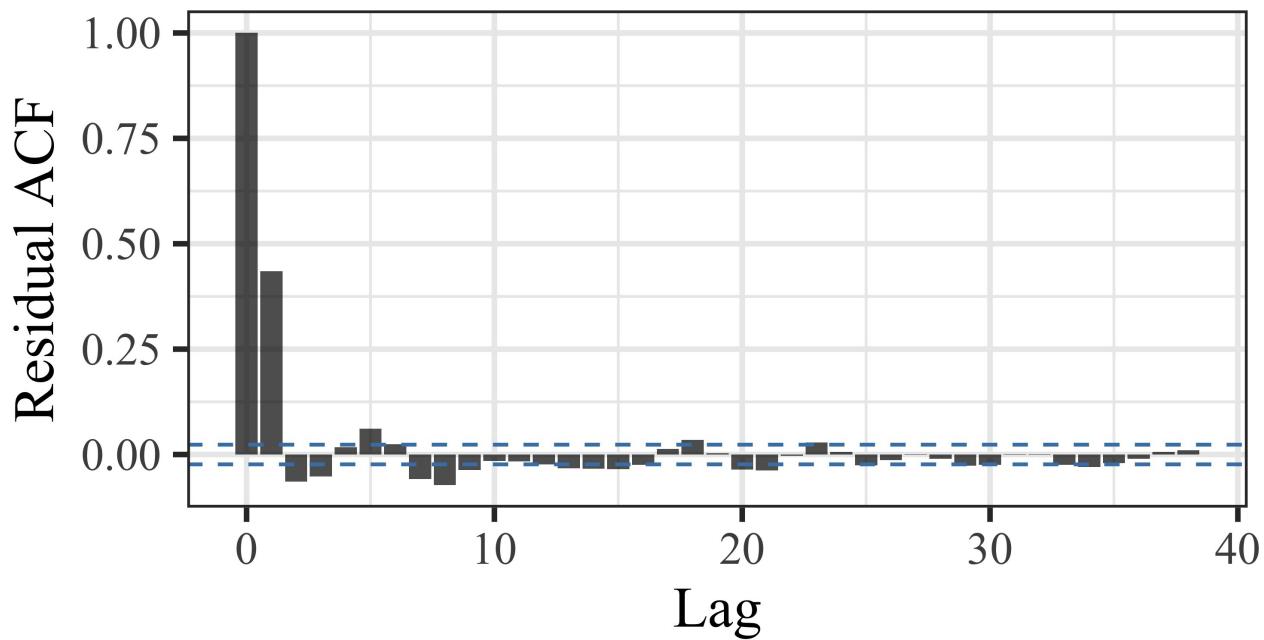
## Depth Difference



Residual independence

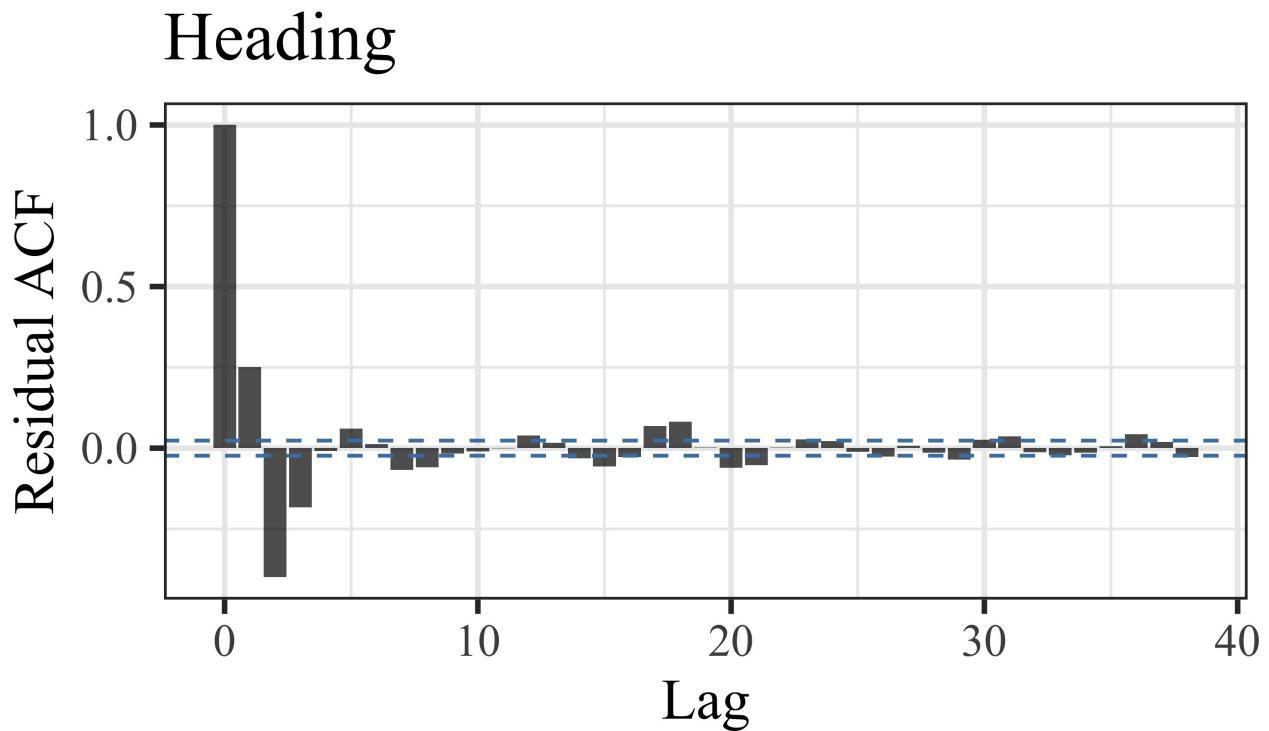
```
CalvinBayes::acf_plot(filter(residz,
                             variable == "Depth difference (t)") |>
  pull(residuals) |>
  na.omit()) |>
  gf_labs(title = "Depth Difference (t)", y = 'Residual ACF', x = 'Lag')
```

## Depth Difference (t)



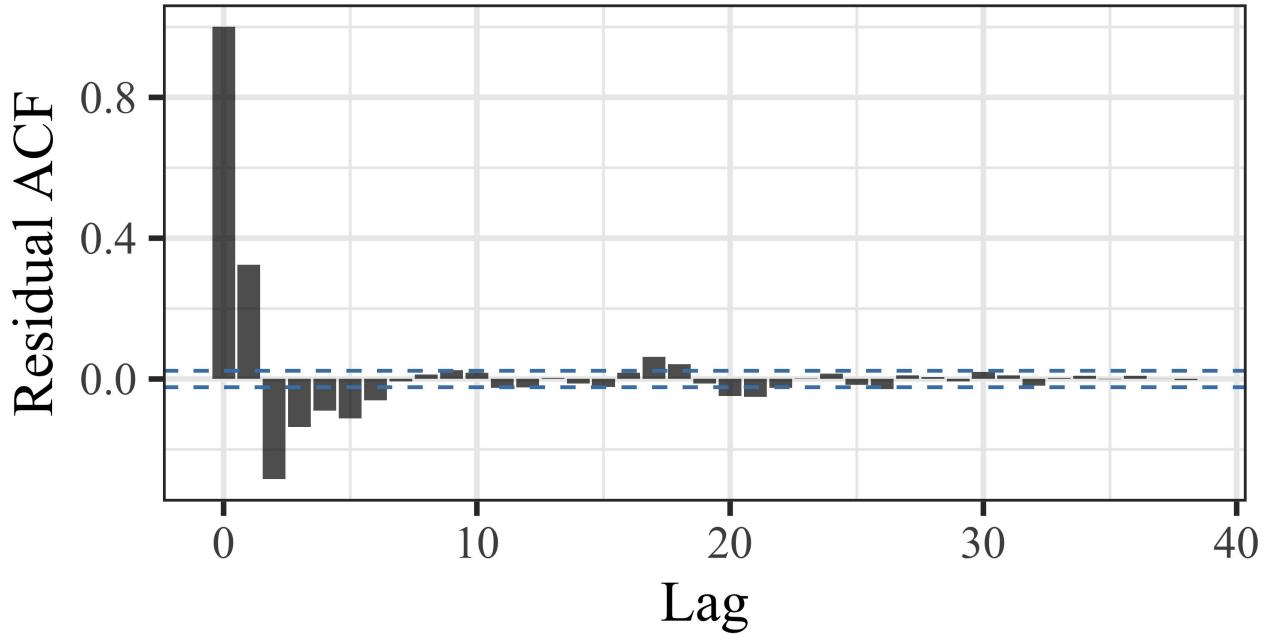
```
CalvinBayes::acf_plot(filter(residz,
                             variable == "Heading") |>
```

```
pull(residuals) |>
na.omit() |>
gf_labs(title = "Heading", y = 'Residual ACF', x = 'Lag')
```



```
CalvinBayes::acf_plot(filter(residz,
variable == "Heading (t)") |>
pull(residuals) |>
na.omit() |>
gf_labs(title = "Heading (t)", y = 'Residual ACF', x = 'Lag')
```

# Heading (t)



Residuals of *all* the models for the deep dives indicate that there are remaining, un-modeled temporal patterns in the data.

## Models: Shallow Dives

### Depth data

To reduce the amount of temporal autocorrelation in the time series data, we model the first difference of the depths (`diff_depth`) instead of the original depth data.

We could include a difference smooth either on the drift (directionality) or on the diffusion (variability) parameter for the depth data.

Here, we include the response on the diffusion parameter.

```
r1file <- 'models/shallow_ddepth.RDS'
if (file.exists(r1file)){
  shallow_ddepth_sde <- readRDS(r1file)
} else{
  f_shallow_ddepth <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts"),
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"))
  # Create and fit SDE
  shallow_ddepth_sde <- SDE$new(formulas = f_shallow_ddepth,
    data = na.omit(abrs_zc_shallow),
    type = "BM",
    response = "diff_depth")
  shallow_ddepth_sde$fit(silent = TRUE)
  saveRDS(shallow_ddepth_sde, r1file)
}
```

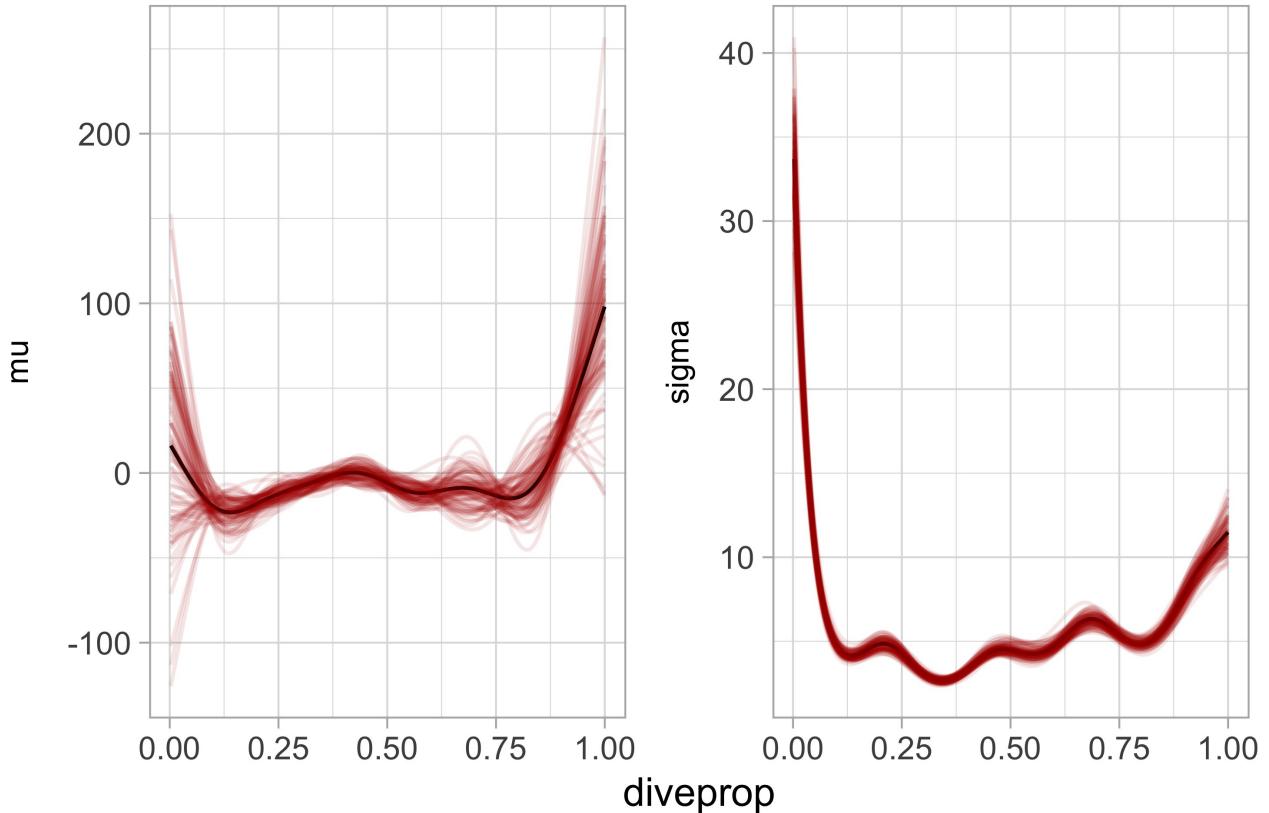
```

r1file <- 'models/shallow_ddeptht.RDS'
if (file.exists(r1file)){
  shallow_ddeptht_sde <- readRDS(r1file)
} else{
  f_shallow_ddeptht <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts"),
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"))
  # Create and fit SDE
  shallow_ddeptht_sde <- SDE$new(formulas = f_shallow_ddeptht,
    data = na.omit(abrs_zc_shallow),
    type = "BM_t",
    other_data = list(df = 3),
    response = "diff_depth")
  shallow_ddeptht_sde$fit(silent = TRUE)
  saveRDS(shallow_ddeptht_sde, r1file)
}

```

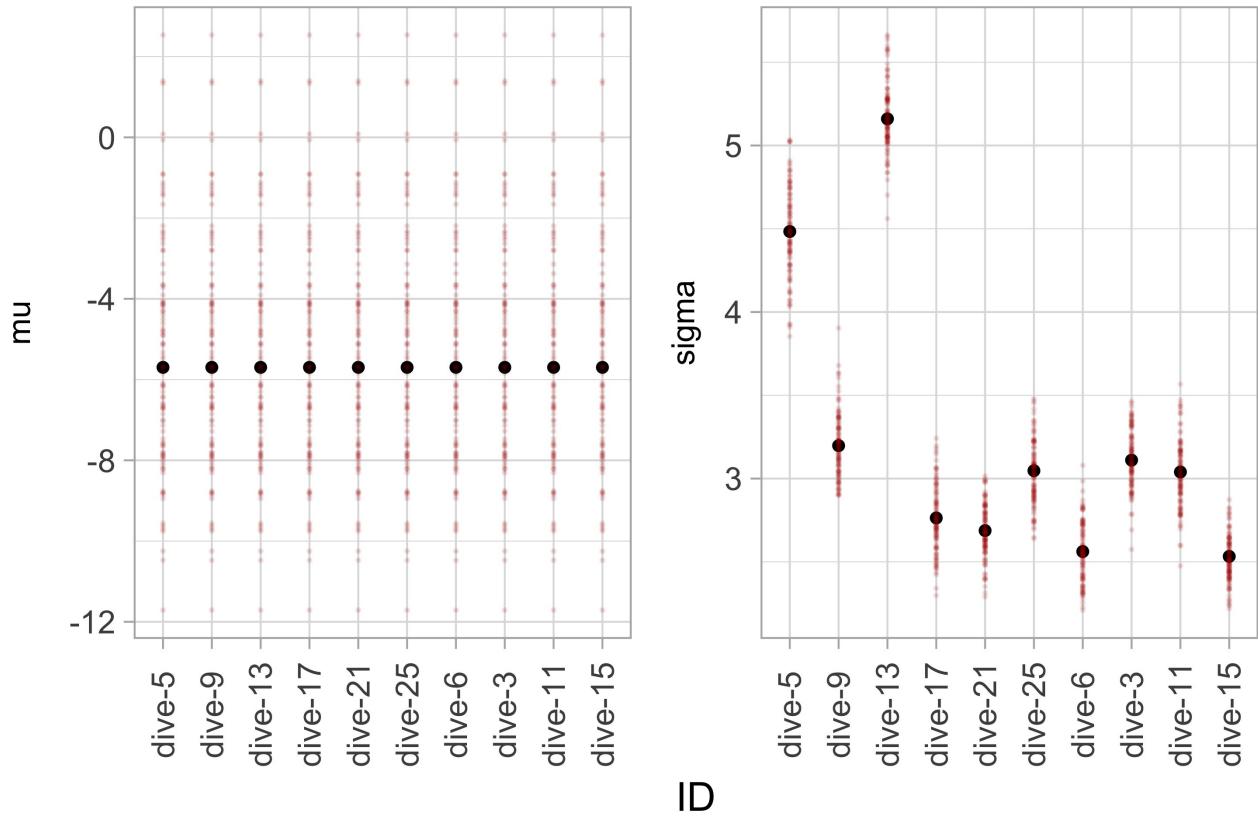
```
shallow_ddepth_sde$plot_par("diveprop")
```

**ID = dive-5, cee\_dive\_status = pre or post**



```
shallow_ddepth_sde$plot_par("ID")
```

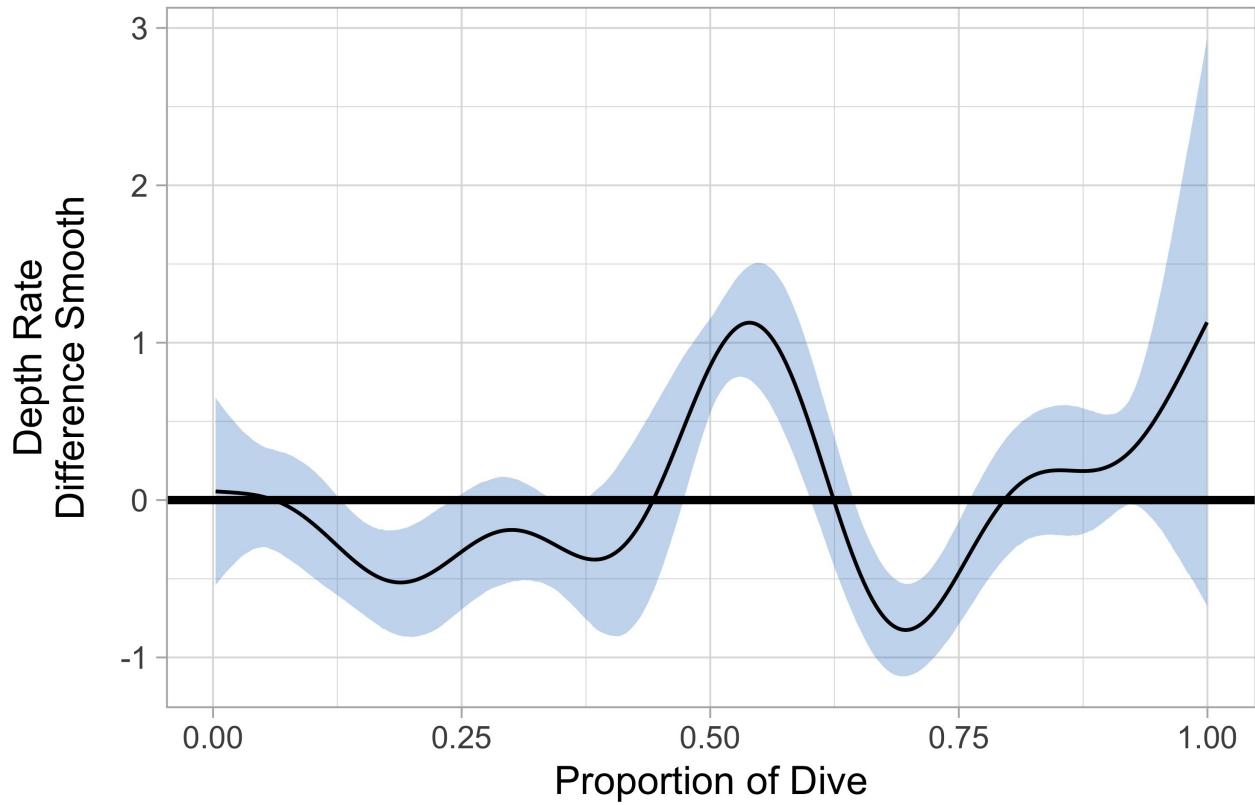
`divideprop = 0.5, cee_dive_status = pre or post`



We can plot the difference smooth for  $\sigma$  on the linear predictor scale, together with 95% confidence intervals, as follows.

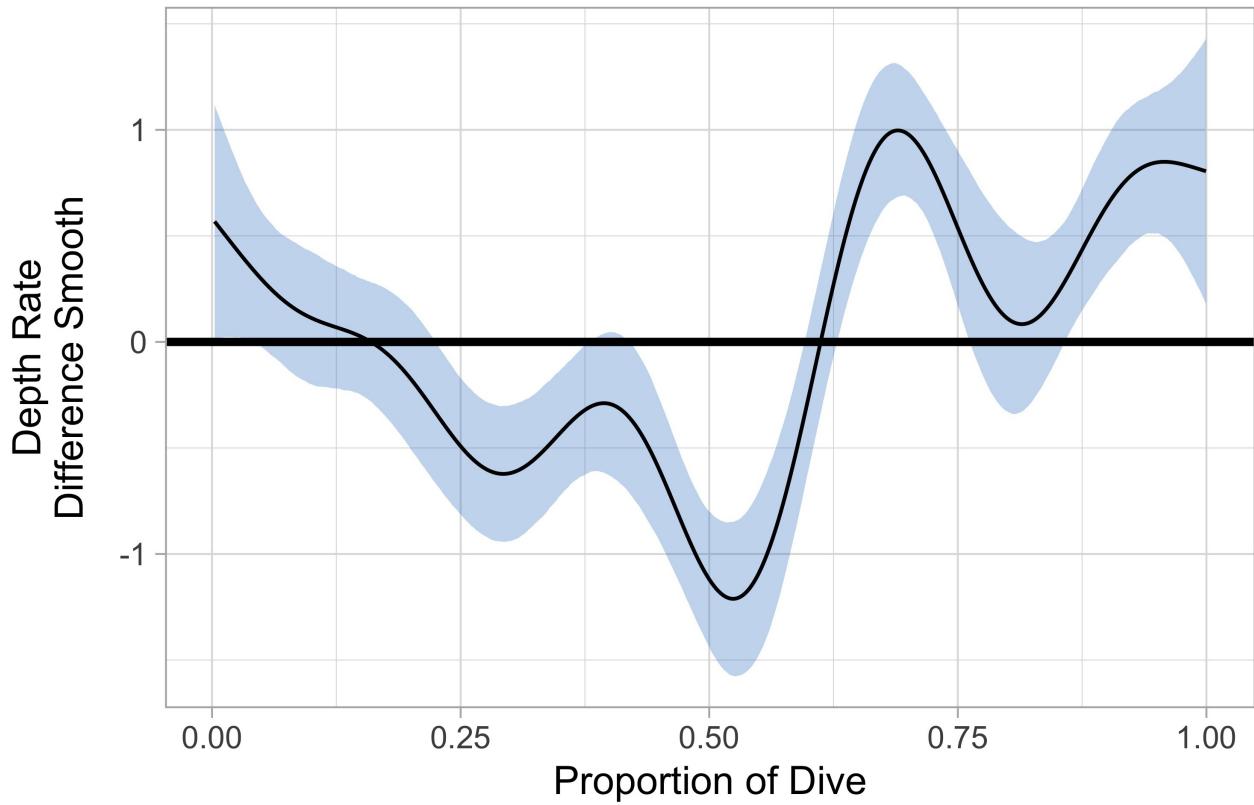
```
# Plot difference smooth on sigma for exposed dives
shallow_ddepth_sde$plot_par("divideprop", par_names = "sigma", n_post = 1e4,
  covs = list(cee_dive_status = "zc17_234aCEE"), term = "cee_dive_status",
  resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc17_234', y = 'Depth Rate\nDifference Smooth', x = 'Proportion of Dive') |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)
```

## zc17\_234



```
shallow_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
  covs = list(cee_dive_status = "zc20_232aCEE"), term = "cee_dive_status",
  resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc20_232', y = 'Depth Rate\\nDifference Smooth', x = 'Proportion of Dive') |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)
```

## zc20\_232



### Heading data

Same caveats discussed earlier - about modeling circular data with a non-circular model - apply for shallow dive models as for the deep dive models.

```
shallow_head_sde_file <- 'models/shallow_head.RDS'
if (file.exists(shallow_head_sde_file)){
  shallow_head_sde <- readRDS(shallow_head_sde_file)
} else{
  shallow_head_f <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"),
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts"))
  # Create and fit SDE
  shallow_head_sde <- SDE$new(formulas = shallow_head_f,
    data = abrs_zc_shallow,
    type = "BM",
    response = "heading")
  shallow_head_sde$fit(silent = TRUE)
  saveRDS(shallow_head_sde, shallow_head_sde_file)
}

shallow_head_tsde_file <- 'models/shallow_head_t.RDS'
if (file.exists(shallow_head_tsde_file)){
  shallow_head_tsde <- readRDS(shallow_head_tsde_file)
```

```

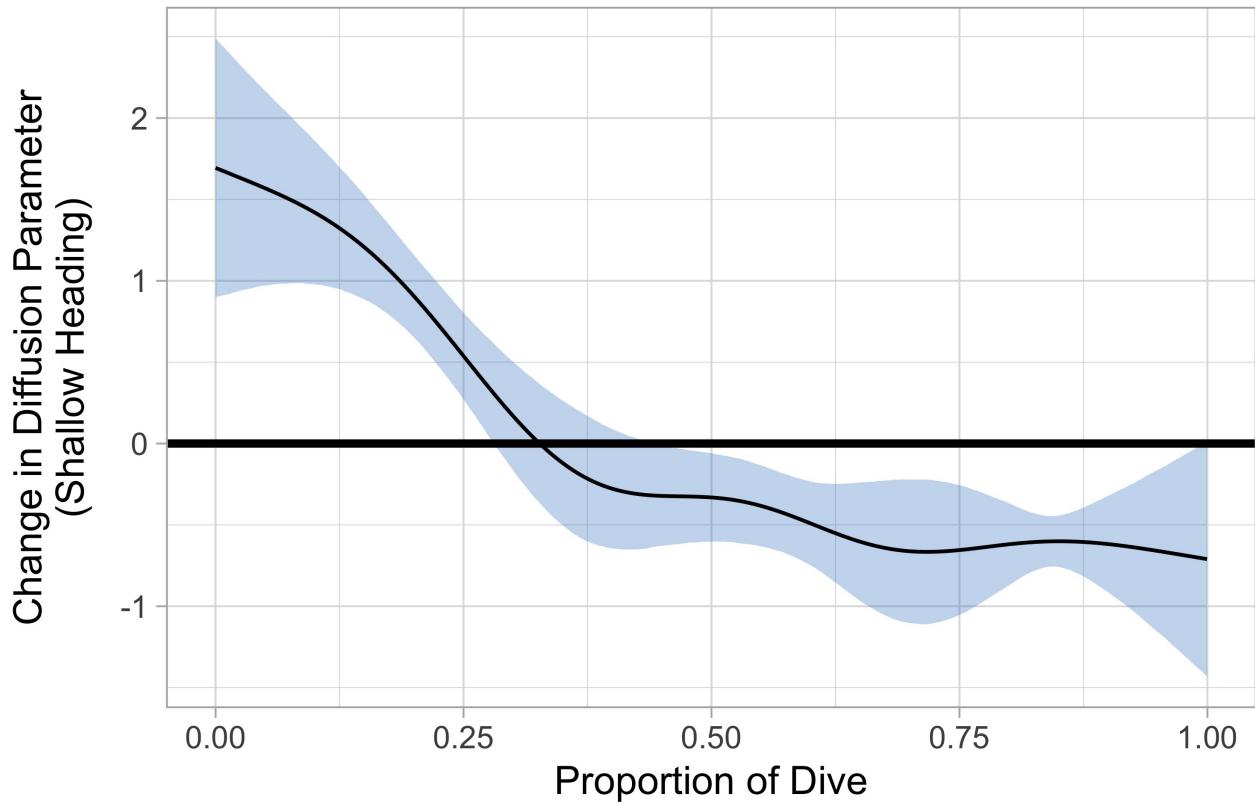
} else{
  shallow_head_f <- list(mu = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts")
    ,
    sigma = ~ s(ID, bs = "re") +
    s(diveprop, k = 10, bs = "ts") +
    s(diveprop, by = cee_dive_status, k = 10, bs = "ts")
  )
# Create and fit SDE
shallow_head_tsde <- SDE$new(formulas = shallow_head_f,
  data = abrs_zc_shallow,
  type = "BM_t",
  response = "heading",
  other_data = list(df = 3)
)
shallow_head_tsde$fit(silent = TRUE)
saveRDS(shallow_head_tsde, shallow_head_tsde_file)
}

# Plot difference smooth on sigma for exposed dives

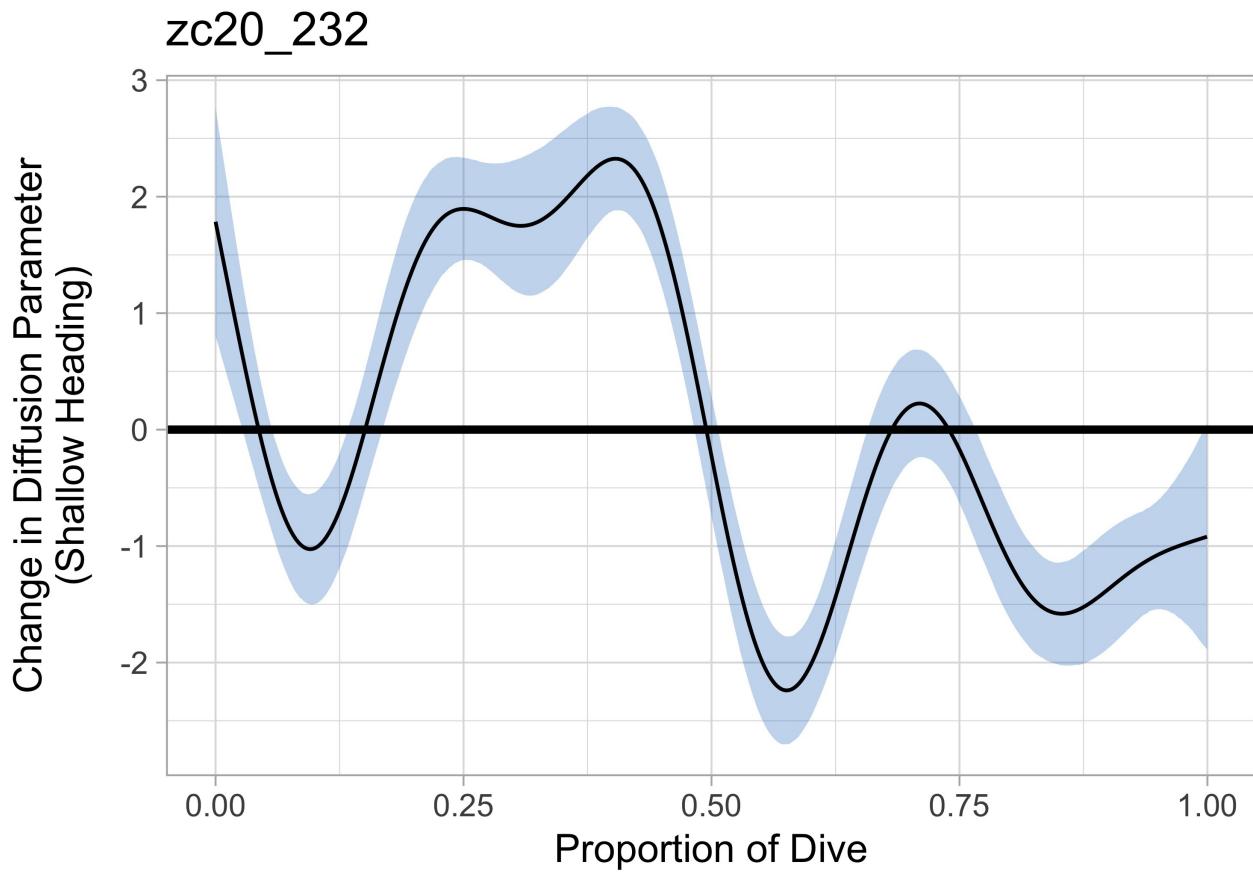
shallow_head_tsde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
  covs = list(cee_dive_status = "zc17_234aCEE"), term = "cee_dive_status",
  resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc17_234', y = 'Change in Diffusion Parameter\n(Shallow Heading)', x = 'Proportion o
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

```

## zc17\_234



```
shallow_head_tsde$plot_par("diveprop", par_names = "sigma",
  covs = list(cee_dive_status = "zc20_232aCEE"), term = "cee_dive_status",
  resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = 'zc20_232', y = 'Change in Diffusion Parameter\n(Shallow Heading)', x = 'Proportion of Dive') |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)
```



#### Model validation for shallow models

```

resids_shal_ddepth <- shallow_ddepth_sde$residuals()
resids_shal_ddeptht <- shallow_ddeptht_sde$residuals()
resids_shal_head <- shallow_head_sde$residuals()
resids_shal_headt <- shallow_head_tsde$residuals()

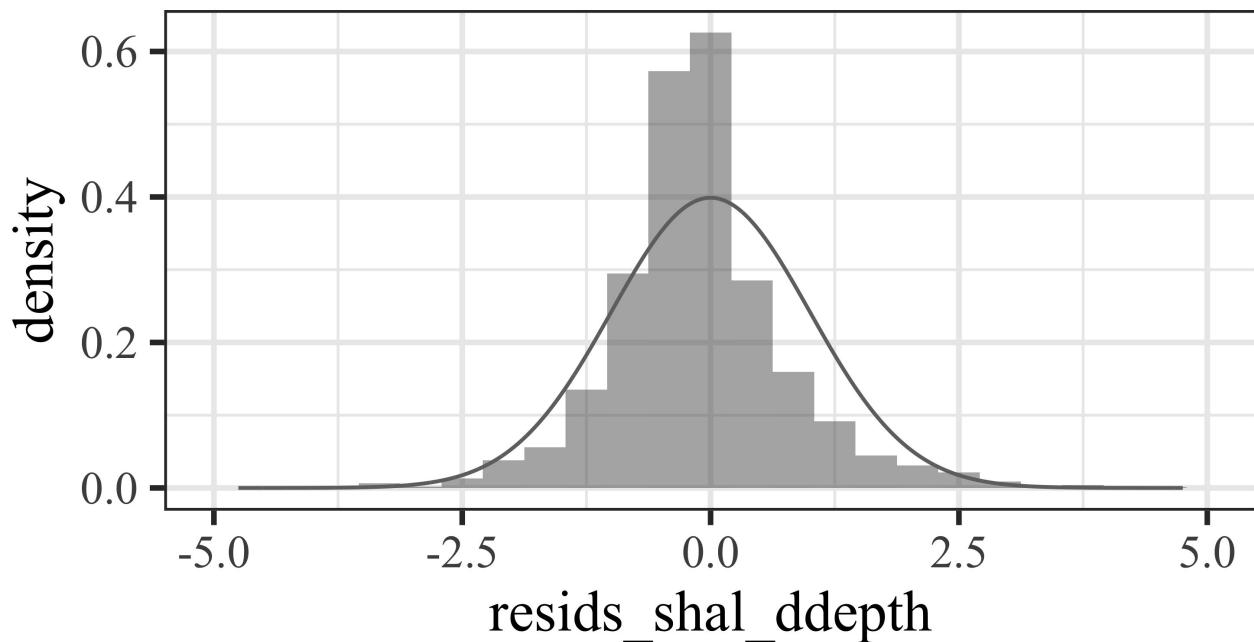
gf_dhistogram(~resids_shal_ddepth) |>
  gf_dist(dist = 'norm',
          color = 'grey44') |>
  gf_lims(x = c(-5, 5)) |>
  gf_labs(title = 'Shallow Depth Difference')

## Warning: Removed 25 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).

```

# Shallow Depth Difference

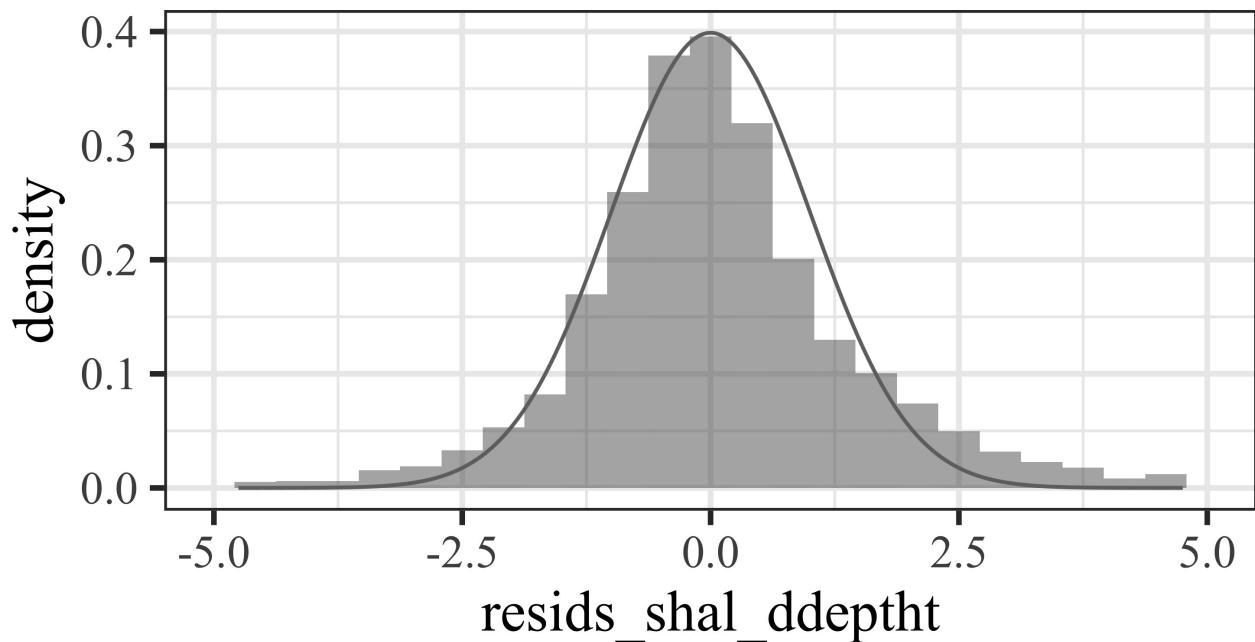


```
gf_dhistogram(~resids_shal_ddepth) |>
  gf_dist(dist = 'norm',
          color = 'grey44') |>
  gf_lims(x = c(-5, 5)) |>
  gf_labs(title = 'Shallow Depth Difference (t)')

## Warning: Removed 92 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```

# Shallow Depth Difference (t)

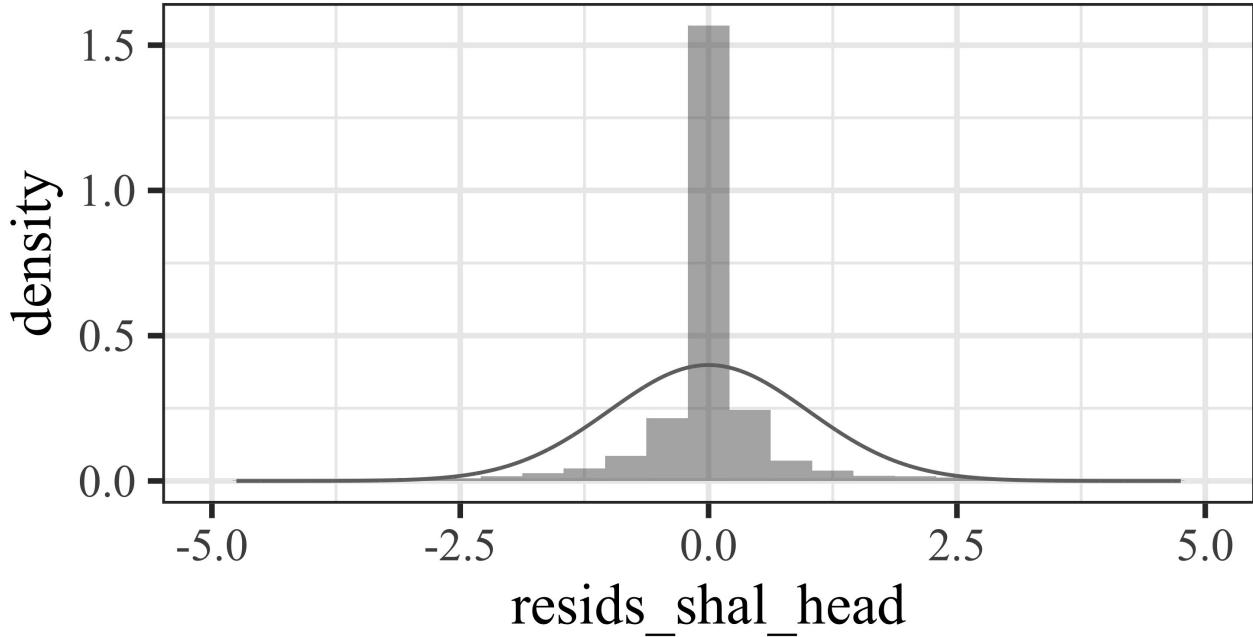


```
gf_dhistogram(~resids_shal_head) |>
  gf_dist(dist = 'norm',
          color = 'grey44') |>
  gf_lims(x = c(-5, 5)) |>
  gf_labs(title = 'Shallow Heading')

## Warning: Removed 52 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```

# Shallow Heading



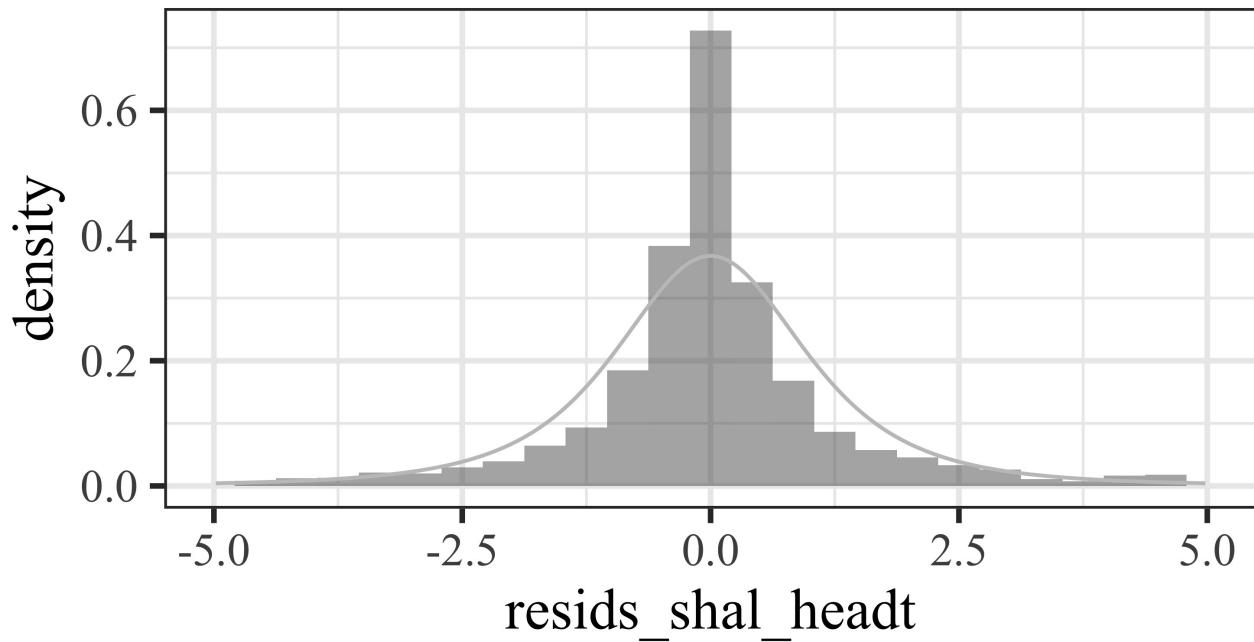
```
gf_dhistogram(~resids_shal_headt) |>
  gf_dist(dist = 't',
          df = 3,
          color = 'grey77') |>
  gf_lims(x = c(-5, 5)) |>
  gf_labs(title = 'Shallow Heading')

## Warning: Removed 353 rows containing non-finite outside the scale range
## (`stat_bin()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).

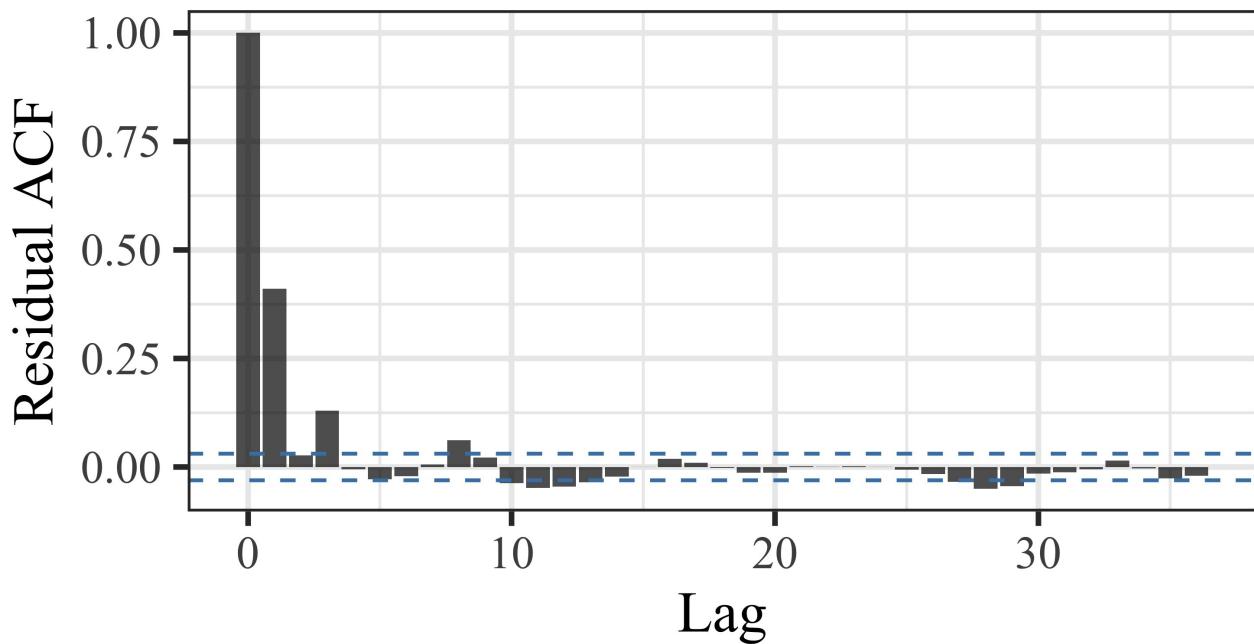
## Warning: Removed 4758 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

## Shallow Heading



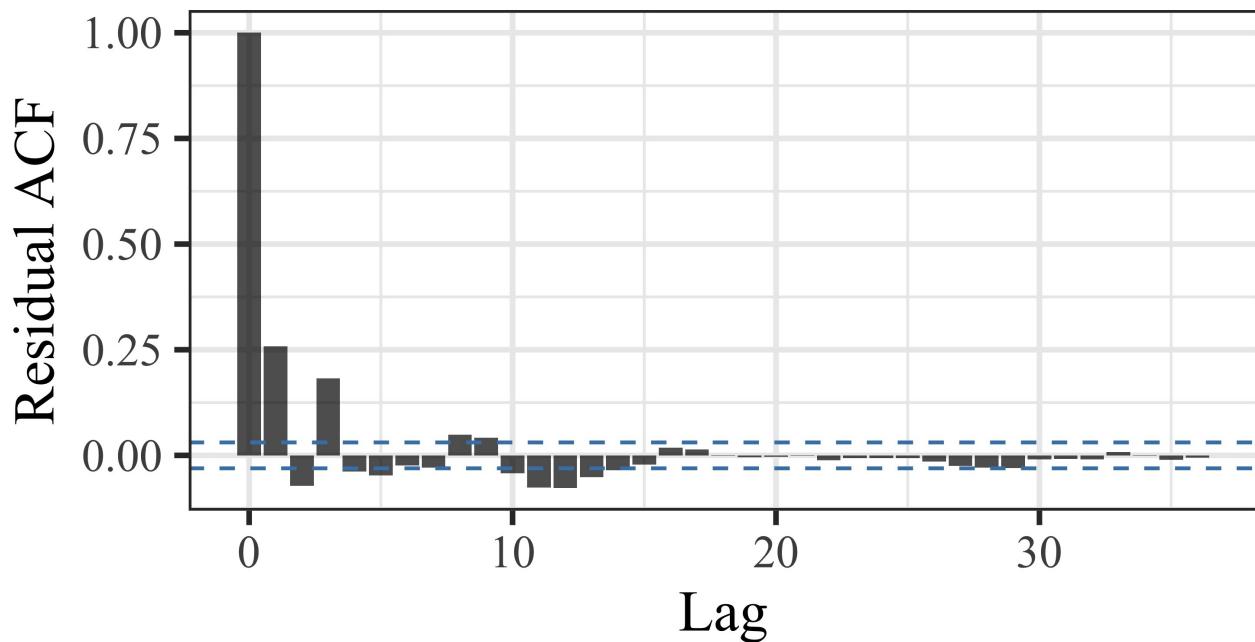
```
CalvinBayes::acf_plot(resids_shal_ddepth |>
                      na.omit()) |>
  gf_labs(title = "Depth Difference", y = 'Residual ACF', x = 'Lag')
```

## Depth Difference



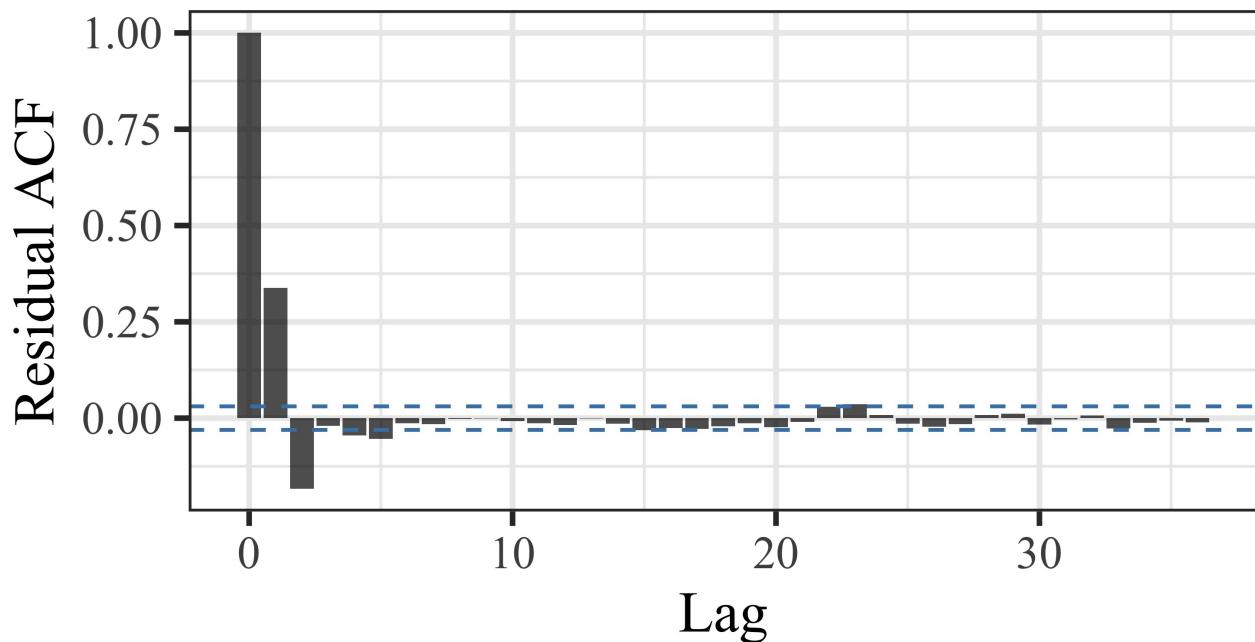
```
CalvinBayes::acf_plot(resids_shal_ddeptht |>
                      na.omit()) |>
  gf_labs(title = "Depth Difference (t)", y = 'Residual ACF', x = 'Lag')
```

## Depth Difference (t)

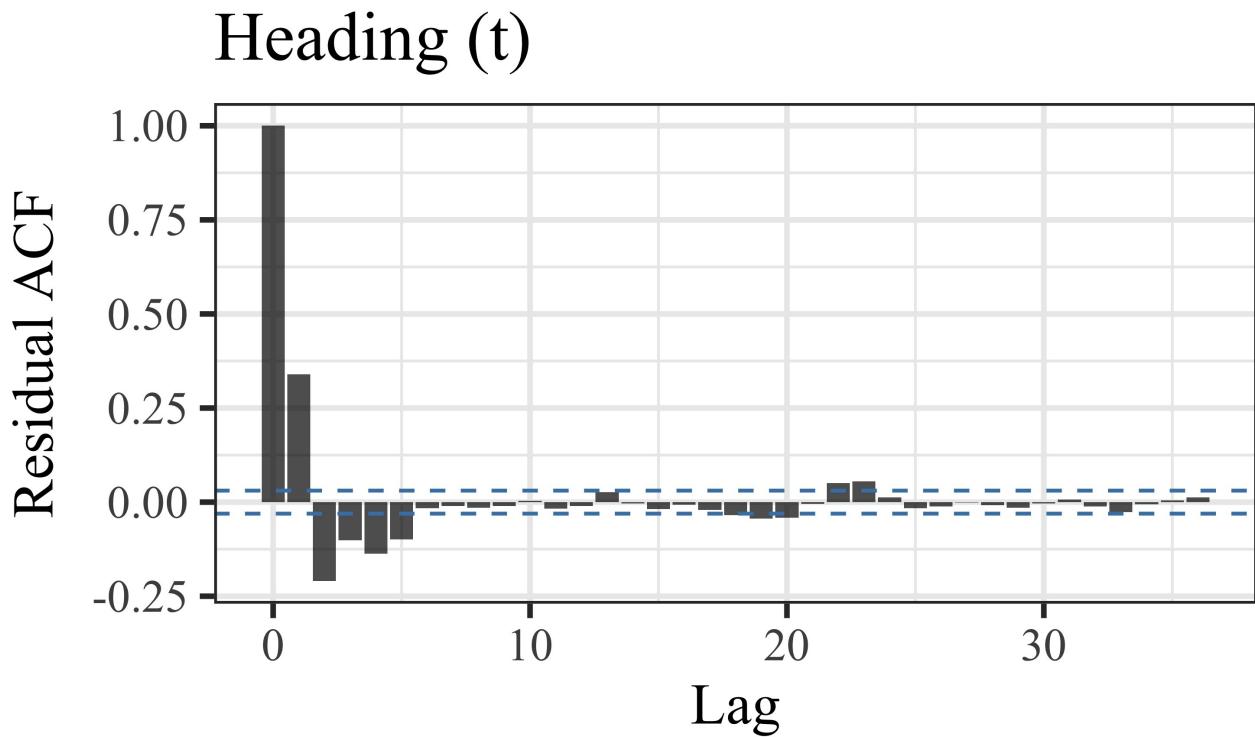


```
CalvinBayes::acf_plot(resids_shal_head |>
  na.omit()) |>
  gf_labs(title = "Heading", y = 'Residual ACF', x = 'Lag')
```

## Heading



```
CalvinBayes::acf_plot(resids_shal_headt |>
  na.omit()) |>
  gf_labs(title = "Heading (t)", y = 'Residual ACF', x = 'Lag')
```



## Figures for paper

```

colrz <- RColorBrewer::brewer.pal(3, 'Set2')
colrz[1] <- colrz[3] <- 'grey50'

abrs_zc25 <- read_csv('data/AtlBRS_Zc_smoothSDE_data_25Hz.csv',
                       show_col_types = FALSE)
abrs_zc25 <- left_join(abrs_zc25, CEE_meta, by = 'whaleID') |>
  mutate(cee_status = case_when( time_hr < cee_start_hr ~ 'pre',
                                 time_hr >= cee_start_hr & time_hr <= cee_end_hr ~ 'during',
                                 time_hr > cee_end_hr ~ 'post'),
         cee_status = fct_relevel(cee_status, 'pre', 'during', 'post'),
         msa1 = msa1 * 9.81) |>
  # 4 hour adj is for local time from UTC
  mutate(datetime = lubridate::seconds(time_hr * 3600) + tagon_time - lubridate::hours(4))

# put a binary variable clicking/not into the time series data for plotting
abrs_zc25 <- mutate(abrs_zc25, clicking = 0, foraging_period = 0)
for (d in c(1:nrow(echolocation))){
  abrs_zc25 <- abrs_zc25 |>
    mutate(clicking = if_else(whaleID == pull(echolocation, whaleID)[d] &
                               time_hr >= pull(echolocation, soc_hr)[d] &
                               time_hr <= pull(echolocation, eoc_hr)[d],
                               1,
                               clicking),
           foraging_period = if_else(whaleID == pull(echolocation, whaleID)[d] &
                                         time_hr >= pull(echolocation, soc_hr)[d] &
                                         time_hr <= pull(echolocation, eoc_hr)[d],
                                         d,
                                         0))
}
  
```

```

        foraging_period))
}

abrs_zc25 <- mutate(abrs_zc25,
                     soc = if_else(clicking - lag(clicking) == 1, 1, 0),
                     eoc = if_else(clicking - lag(clicking) == -1, 1, 0))

if (!file.exists('data/abrs_zc_dtag_25Hz.csv')){
write_csv(abrs_zc25, file = 'data/abrs_zc_dtag_25Hz.csv')
}

if (!file.exists('data/abrs_zc_msa_25Hz.csv')){
write_csv(abrs_zc25 |> select(whaleID, time_hr, msa1, datetime, cee_status),
          file = 'data/abrs_zc_msa_25Hz.csv')
}

abrs_zc <- abrs_zc |>
  # 4 hour adj is for local time from UTC
  mutate(datetime = lubridate::seconds(time_hr * 3600) + tagon_time - lubridate::hours(4))

zc19_218a_dive_RL <- gf_line(depth ~ datetime,
                                color = ~cee_status,
                                data = abrs_zc |> filter(whaleID == 'zc19_218a')) |>
  gf_point(depth ~ datetime,
            color = ~cee_status,
            data = abrs_zc |> filter(whaleID == 'zc19_218a' &
                                      (soc == 1 | eoc == 1)),
            size = 2) |>
  gf_theme(scale_color_manual('CEE', values = colrz,
                               guide = guide_legend(nrow = 1))) |>
  gf_lims(y = c(1800, 0)) |>
  gf_labs(x = '', y = 'Depth\n(m)') |>
  gf_theme(legend.position = 'none') # c(0.75, 0.1))

zc19_218a_head <- gf_point(heading ~ datetime,
                            color = ~cee_status,
                            size = 0.5, alpha = 0.5,
                            data = abrs_zc |> filter(whaleID == 'zc19_218a')) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_labs(x = '', y = 'Heading\n(degrees)') |>
  gf_theme(legend.position = 'none')

this_25 <- abrs_zc25 |> filter(whaleID == 'zc19_218a' &
                                time_hr <= max(pull(abrs_zc |> filter(whaleID == 'zc19_218a'),
                                         time_hr)) &
                                time_hr >= min(pull(abrs_zc |> filter(whaleID == 'zc19_218a'),
                                         time_hr)))

zc19_218a_head_jk <- gf_line(mag_jerk ~ datetime,
                               color = ~cee_status,
                               data = this_25) |>

```

```

gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_labs(x = '', y = 'Turn Angle\n(radians)') |>
  gf_theme(legend.position = 'none')

# show the 25 Hz data and not the 1/3Hz
zc19_218a_msa <- gf_line(msa1 ~ datetime,
                           color = ~cee_status,
                           data = this_25) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_labs(x = '', y = 'MSA\n(m/sec/sec)') |>
  gf_theme(legend.position = 'none')

zc19_218a_depth_diffsmooth <-
  deep_ddepth_sde$plot_par("diveprop",
                            par_names = "sigma",
                            n_post = 1e4,
                            covs = list(cee_dive_status = "zc19_218aCEE"),
                            term = "cee_dive_status",
                            resp = FALSE,
                            show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = '',
          y = 'Depth Rate\nDifference Smooth',
          x = 'Proportion of Dive') |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc19_218a_depth_diffsmooth)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 16
zc19_218a_depth_diffsmooth <- ggplot_gtable(temp)

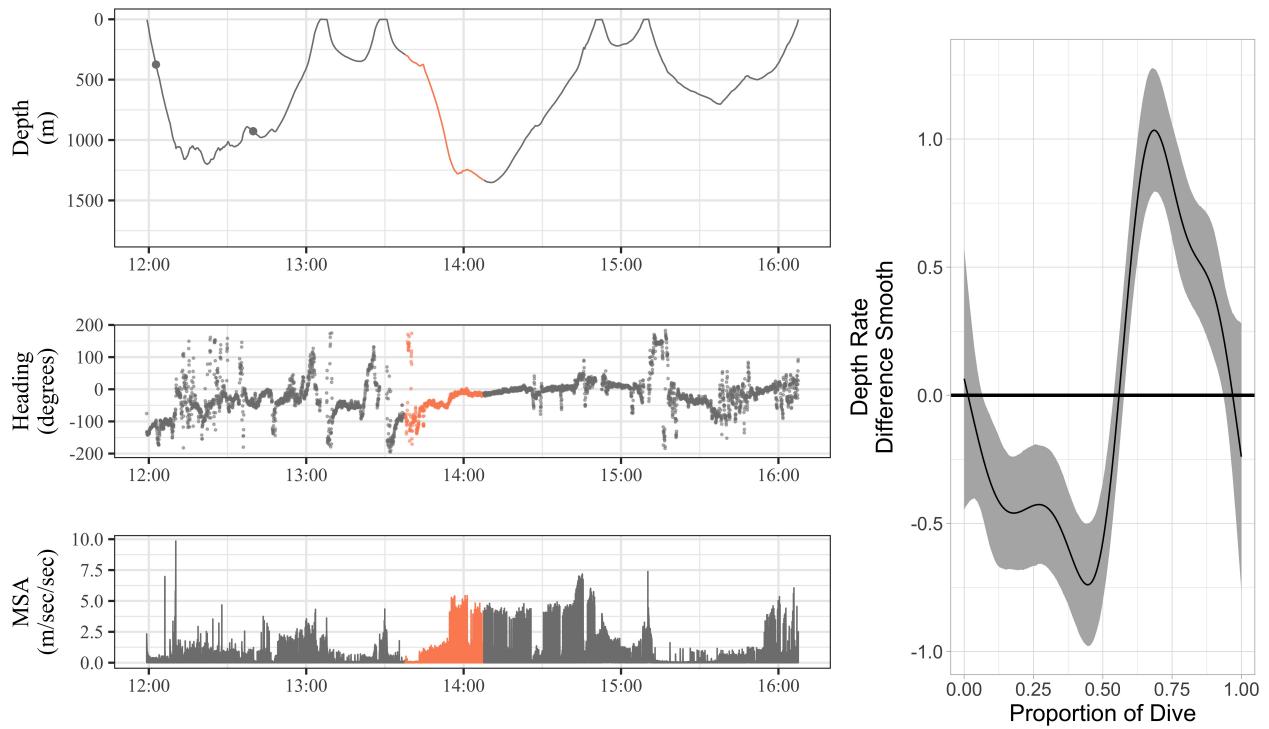
tag_data <- plot_grid(zc19_218a_dive_RL, zc19_218a_head, zc19_218a_msa,
                      rel_heights = c(1.5, 1, 1),
                      ncol = 1,
                      align = 'hv')

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).

# diff_smooths <- plot_grid(p4_depth_diffsmooth)

plot_grid(tag_data, zc19_218a_depth_diffsmooth,
          ncol = 2,
          rel_widths = c(2, 1))

```



```
w <- 'zc17_234a'

zc17_234a_dive_RL <- gf_line(depth ~ datetime,
  color = ~cee_status,
  data = abrs_zc |> filter(whaleID == w)) |>
  gf_point(depth ~ datetime,
  color = ~cee_status,
  data = abrs_zc |> filter(whaleID == w &
    (soc == 1 | eoc == 1)),
  size = 2) |>
  gf_theme(scale_color_manual('CEE', values = colrz,
    guide = guide_legend(nrow = 1))) |>
  gf_lims(y = c(1800, 0)) |>
  gf_labs(x = '', y = 'Depth (m)') |>
  gf_theme(legend.position = 'none') # c(0.75, 0.1))

zc17_234a_head <- gf_point(heading ~ datetime,
  color = ~cee_status,
  size = 0.5, alpha = 0.5,
  data = abrs_zc |> filter(whaleID == w)) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_labs(x = '', y = 'Heading\n(degrees)') |>
  gf_theme(legend.position = 'none')

this_25 <- abrs_zc25 |> filter(whaleID == w &
  time_hr <= max(pull(abrs_zc |> filter(whaleID == w),
    time_hr)) &
  time_hr >= min(pull(abrs_zc |> filter(whaleID == w),
    time_hr)))

zc17_234a_head_jk <- gf_line(mag_jerk ~ datetime,
```

```

        color = ~ cee_status,
        data = this_25 |> filter(whaleID == w)) |>
gf_theme(scale_color_manual('CEE', values = colrz)) |>
gf_labs(x = '', y = 'Turn Angle\n(radians)') |>
gf_theme(legend.position = 'none')

# show the 25 Hz data not the 1/3Hz
zc17_234a_msa <- gf_line(msa1 ~ datetime,
                           color = ~cee_status,
                           data = this_25 |> filter(whaleID == w)) |>
gf_theme(scale_color_manual('CEE', values = colrz)) |>
gf_labs(x = '', y = 'MSA\n(m/sec/sec)') |>
gf_theme(legend.position = 'none')

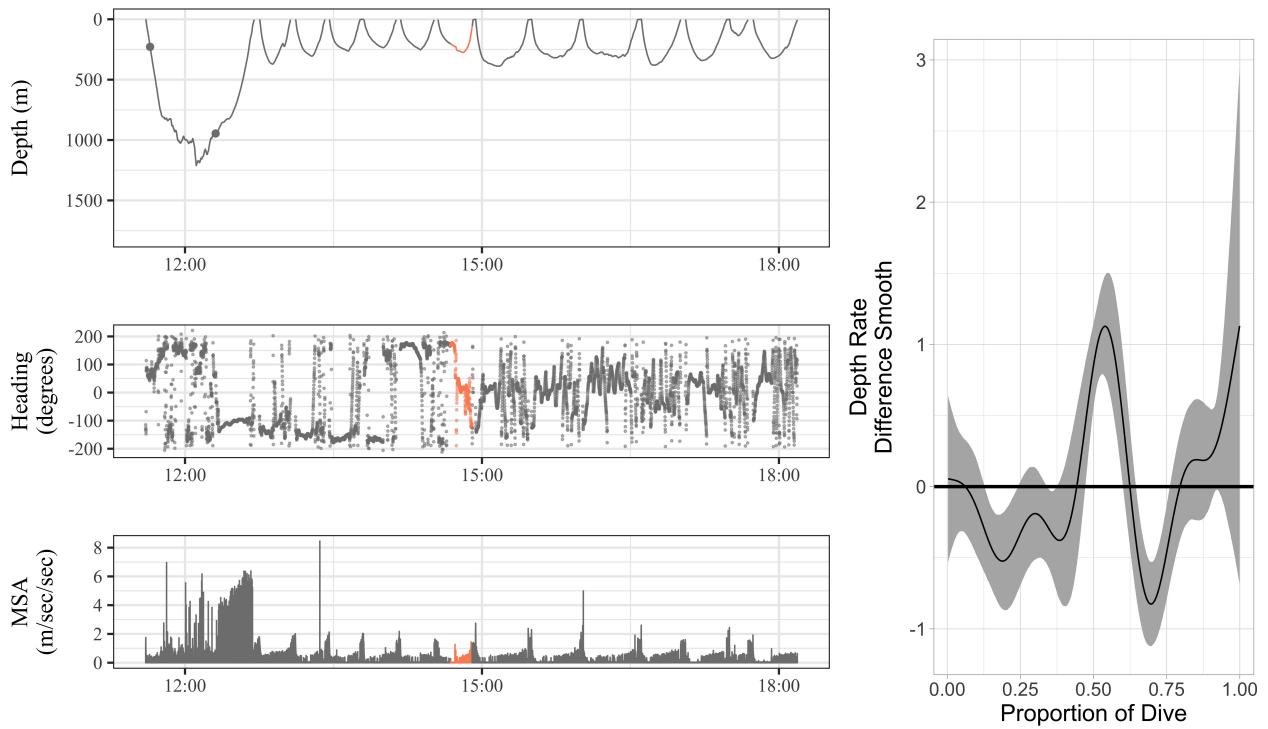
zc17_234a_depth_diffsmooth <-
shallow_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
                            covs = list(cee_dive_status = paste0(w, 'CEE')),
                            term = "cee_dive_status",
                            resp = FALSE, show_CI = "simultaneous") |>
gf_theme(strip.text.y = element_blank()) |>
gf_labs(title = '',
        y = 'Depth Rate\nDifference Smooth',
        x = 'Proportion of Dive') |>
gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc17_234a_depth_diffsmooth)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 16
zc17_234a_depth_diffsmooth <- ggplot_gtable(temp)

tag_data <- plot_grid(zc17_234a_dive_RL,
                      zc17_234a_head,
                      zc17_234a_msa,
                      rel_heights = c(1.5,1,1),
                      ncol = 1,
                      align = 'hv')

plot_grid(tag_data, zc17_234a_depth_diffsmooth,
          ncol = 2,
          rel_widths = c(2,1))

```



```
w <- 'zc20_232a'

zc20_232a_dive_RL <- gf_line(depth ~ datetime,
  color = ~cee_status,
  data = abrs_zc |> filter(whaleID == w)) |>
  gf_point(depth ~ datetime,
  color = ~cee_status,
  data = abrs_zc |> filter(whaleID == w &
    (soc == 1 | eoc == 1)),
  size = 2) |>
  gf_theme(scale_color_manual('CEE', values = colrz,
    guide = guide_legend(nrow = 1))) |>
  gf_lims(y = c(1800, 0)) |>
  gf_labs(x = '', y = 'Depth (m)') |>
  gf_theme(legend.position = 'none') # c(0.75, 0.1))

zc20_232a_head <- gf_point(heading ~ datetime,
  color = ~cee_status,
  data = abrs_zc |> filter(whaleID == w),
  size = 0.5, alpha = 0.5) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_labs(x = '', y = 'Heading\n(degrees)') |>
  gf_theme(legend.position = 'none')

this_25 <- abrs_zc25 |> filter(whaleID == w &
  time_hr <= max(pull(abrs_zc |> filter(whaleID == w),
    time_hr)) &
  time_hr >= min(pull(abrs_zc |> filter(whaleID == w),
    time_hr)))

zc20_232a_head_jk <- gf_line(mag_jerk ~ datetime,
```

```

        color = ~ cee_status,
        data = this_25 |> filter(whaleID == w)) |>
gf_theme(scale_color_manual('CEE', values = colrz)) |>
gf_labs(x = '', y = 'Turn Angle\n(radians)') |>
gf_theme(legend.position = 'none')

# show the 25 Hz data not the 1/3Hz
zc20_232a_msa <- gf_line(msa1 ~ datetime,
                           color = ~cee_status,
                           data = this_25 |> filter(whaleID == w)) |>
gf_theme(scale_color_manual('CEE', values = colrz)) |>
gf_labs(x = '', y = 'MSA\n(m/sec/sec)') |>
gf_theme(legend.position = 'none')

zc20_232a_depth_diffsmooth1 <-
shallow_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
                            covs = list(cee_dive_status = paste0(w, 'CEE')),
                            term = "cee_dive_status",
                            resp = FALSE, show_CI = "simultaneous") |>
gf_theme(strip.text.y = element_blank()) |>
gf_labs(title = '',
        y = 'Depth Rate\nDifference Smooth',
        x = 'Proportion of Dive') |>
gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc20_232a_depth_diffsmooth1)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 16
zc20_232a_depth_diffsmooth1 <- ggplot_gtable(temp)

zc20_232a_depth_diffsmooth2 <-
deep_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
                         covs = list(cee_dive_status = paste0(w, 'CEE')), term = "cee_dive_status",
                         resp = FALSE, show_CI = "simultaneous") |>
gf_theme(strip.text.y = element_blank()) |>
gf_labs(title = '',
        y = 'Depth Rate\nDifference Smooth',
        x = 'Proportion of Dive') |>
gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc20_232a_depth_diffsmooth2)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 16
zc20_232a_depth_diffsmooth2 <- ggplot_gtable(temp)

tag_data <- plot_grid(zc20_232a_dive_RL,
                      zc20_232a_head,
                      zc20_232a_msa,
                      rel_heights = c(1.5, 1, 1),
                      ncol = 1,
                      align = 'hv')

## Warning: Removed 1 row containing missing values or values outside the scale range

```

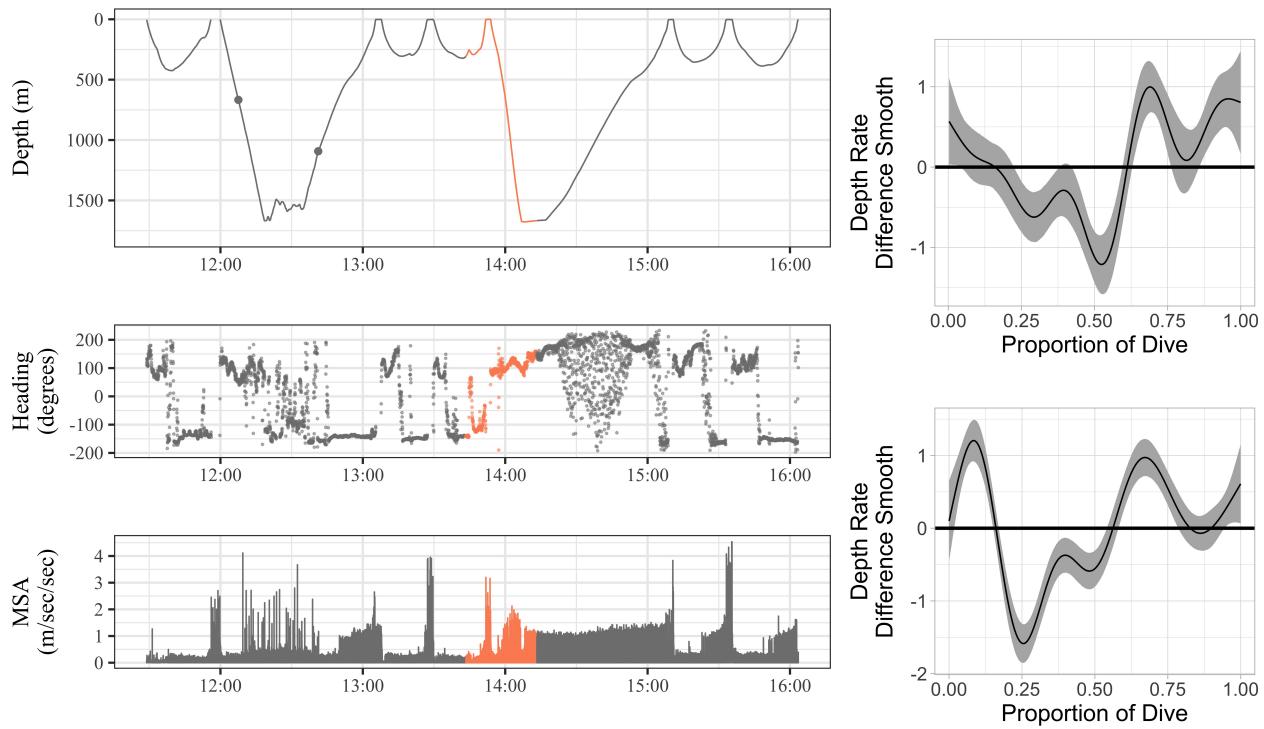
```

## (`geom_line()`).

diff_smooths <- plot_grid(zc20_232a_depth_diffsmooth1,
                           zc20_232a_depth_diffsmooth2,
                           rel_heights = c(1,1),
                           ncol = 1,
                           align = 'hv')

plot_grid(tag_data, diff_smooths,
          ncol = 2,
          rel_widths = c(2,1))

```

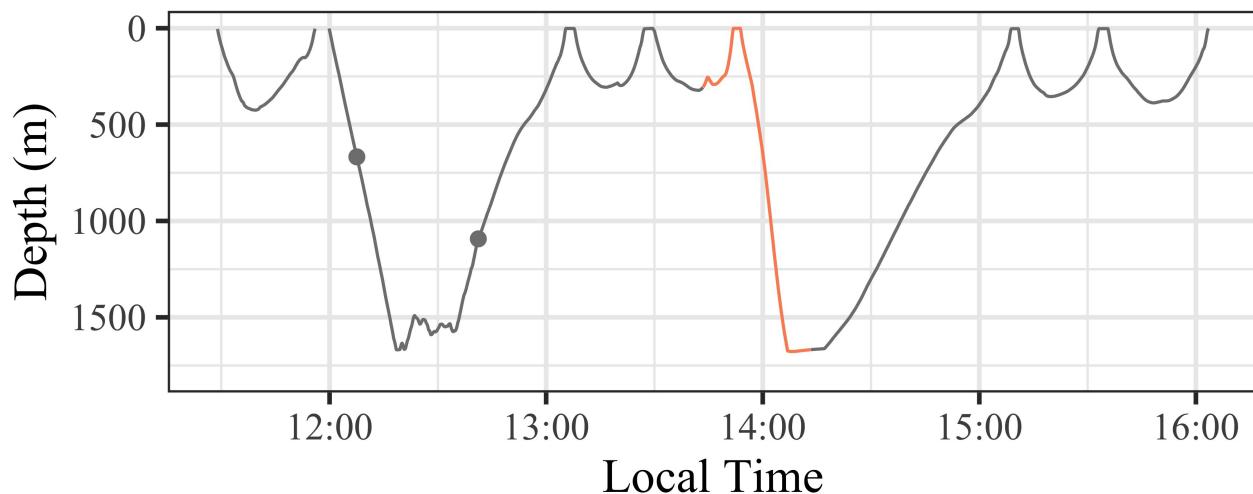


```

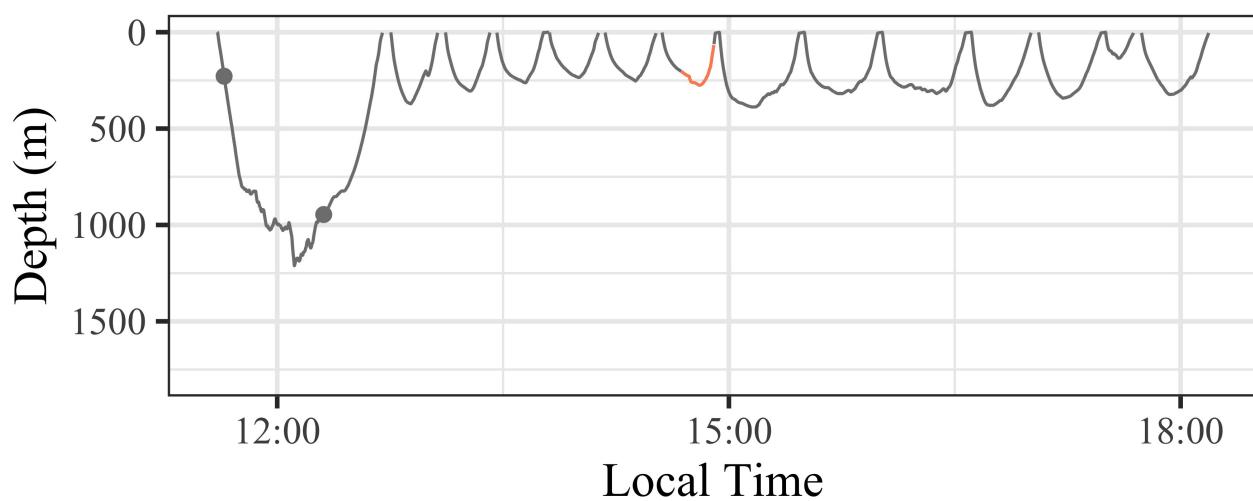
zc20_232a_dive_RL |>
  gf_labs(x = 'Local Time')

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

```

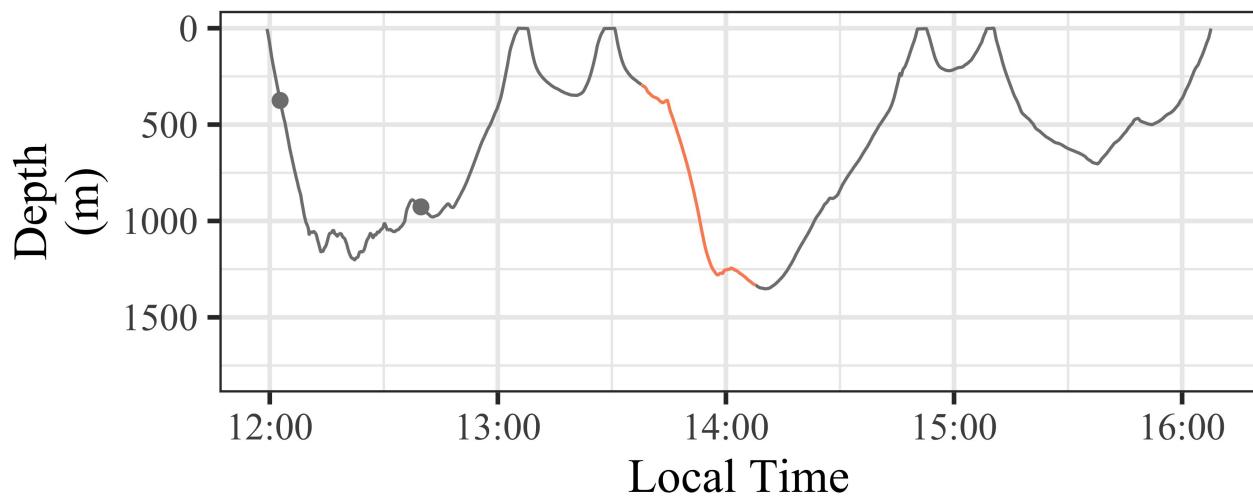


```
zc17_234a_dive_RL |>  
  gf_labs(x = 'Local Time')
```

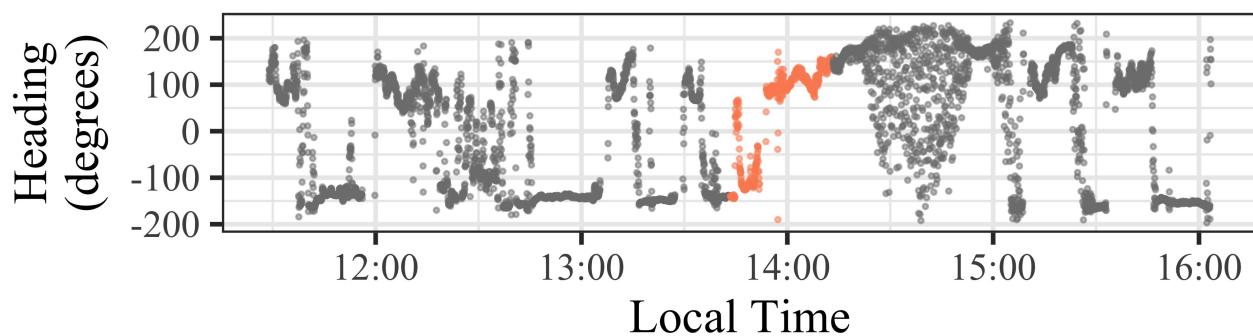


```
zc19_218a_dive_RL |>  
  gf_labs(x = 'Local Time')
```

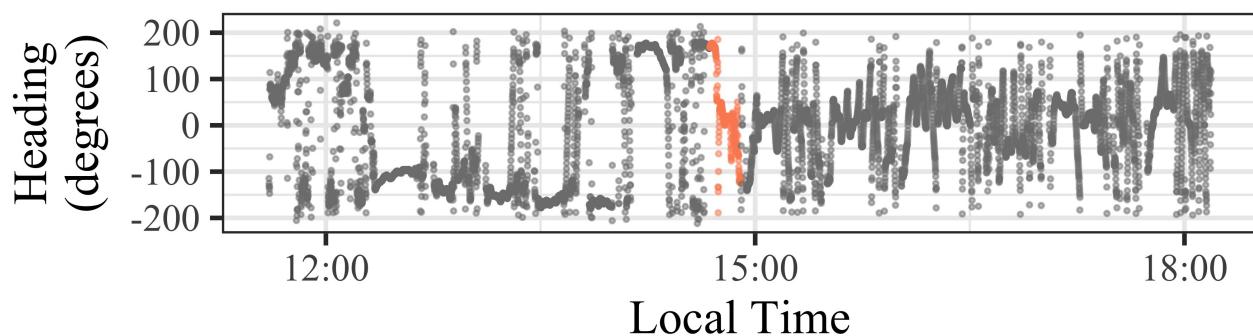
```
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom_line()`).
```



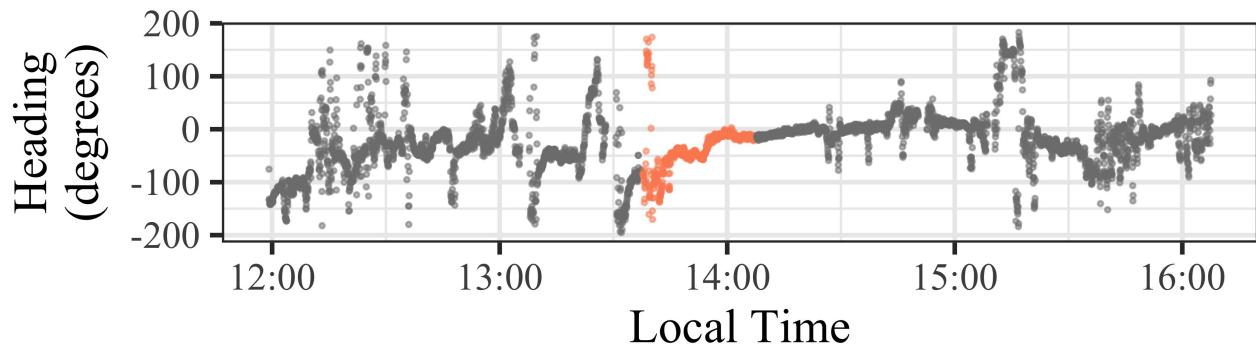
```
zc20_232a_head |>
  gf_labs(x = 'Local Time')
```



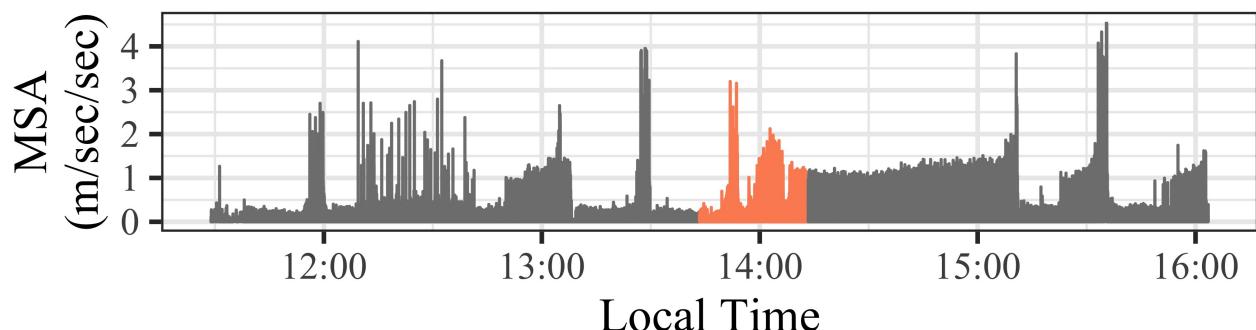
```
zc17_234a_head |>
  gf_labs(x = 'Local Time')
```



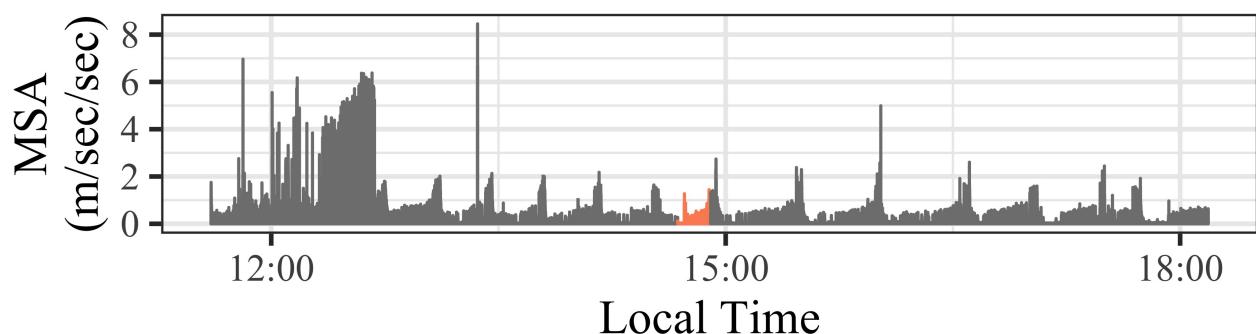
```
zc19_218a_head |>
  gf_labs(x = 'Local Time')
```



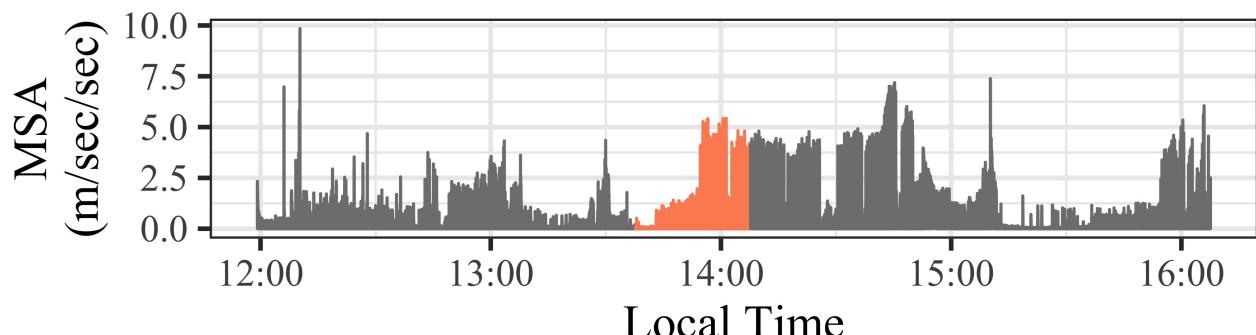
```
zc20_232a_msa |>
  gf_labs(x = 'Local Time')
```



```
zc17_234a_msa |>
  gf_labs(x = 'Local Time')
```



```
zc19_218a_msa |>
  gf_labs(x = 'Local Time')
```



Version with overlaid dive panel

```

this_overlaid_deep <- abrs_zc_deep |>
  filter(whaleID == 'zc20_232a' | cee_dive_status == 'pre or post') |>
  mutate(cee_dive_status = factor(cee_dive_status))

zc20_232a_overlaid_deep <- gf_line(depth ~ diveprop,
  color = ~cee_dive_status,
  group = ~ID,
  data = this_overlaid_deep) |>
  gf_point(depth ~ diveprop,
    color = ~cee_dive_status,
    group = ~ID,
    data = this_overlaid_deep |> filter(soc == 1 | eoc == 1),
    size = 2) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_lims(y = c(1800, 0)) |>
  gf_labs(x = '', # # x = 'Proportion total dive time',
    y = 'Depth (m)',
    title = 'Deep Dives') |>
  gf_theme(legend.position = 'none')

this_overlaid_deep <- abrs_zc_deep |>
  filter(whaleID == 'zc19_218a' | cee_dive_status == 'pre or post') |>
  mutate(cee_dive_status = factor(cee_dive_status))

zc19_218a_overlaid_deep <- gf_line(depth ~ diveprop,
  color = ~cee_dive_status,
  group = ~ID,
  data = this_overlaid_deep) |>
  gf_point(depth ~ diveprop,
    color = ~cee_dive_status,
    group = ~ID,
    data = this_overlaid_deep |> filter(soc == 1 | eoc == 1),
    size = 2) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_lims(y = c(1800, 0)) |>
  gf_labs(x = '', # # x = 'Proportion total dive time',
    y = 'Depth (m)',
    title = 'Deep Dives') |>
  gf_theme(legend.position = 'none')

zc17_234a_overlaid_shallow <- gf_line(depth ~ diveprop,
  color = ~cee_dive_status,
  group = ~ID,
  data = abrs_zc_shallow |> filter(cee_dive_status %in% c('zc17_234aCEE', 'pre or post'))) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_lims(y = c(450, 0)) |>
  gf_labs(x = '', # # x = 'Proportion total dive time',
    y = 'Depth (m)',
    title = 'Shallow Dives') |>
  gf_theme(legend.position = 'none')

zc20_232a_overlaid_shallow <- gf_line(depth ~ diveprop,
  color = ~cee_dive_status,
  group = ~ID,

```

```

    data = abrs_zc_shallow |> filter(cee_dive_status %in% c('zc20_232aCEE', 'pre or post')) |>
  gf_theme(scale_color_manual('CEE', values = colrz)) |>
  gf_lims(y = c(450, 0)) |>
  gf_labs(x = '', # 'Proportion total dive time',
           y = 'Depth (m)',
           title = 'Shallow Dives') |>
  gf_theme(legend.position = 'none')

tag_data <- plot_grid(zc20_232a_dive_RL,
                      zc20_232a_head,
                      zc20_232a_msa,
                      rel_heights = c(1.5,1,1),
                      ncol = 1,
                      align = 'hv')

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

diff_smooths <- plot_grid(zc20_232a_overlaid_deep,
                           zc20_232a_overlaid_shallow,
                           zc20_232a_depth_diffsmooth2,
                           zc20_232a_depth_diffsmooth1,
                           rel_heights = c(1,1),
                           rel_widths = c(1,1),
                           ncol = 2, nrow = 2,
                           align = 'hv')

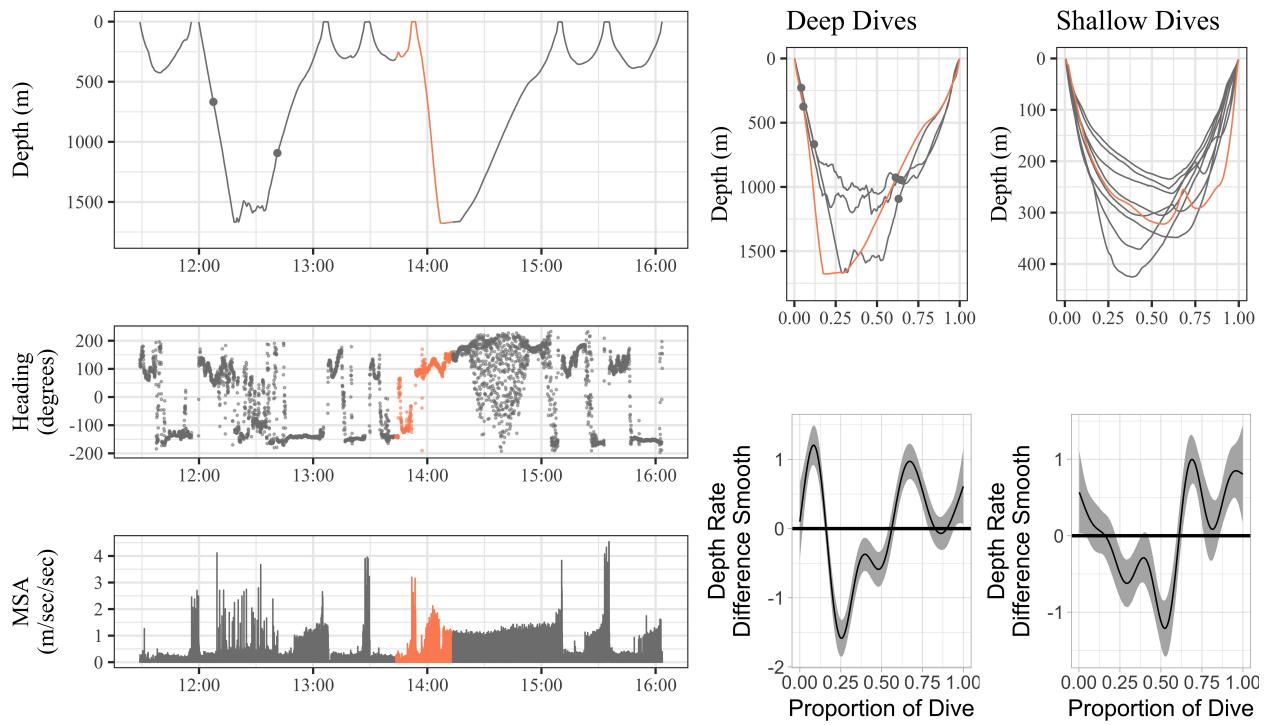
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
## Placing graphs unaligned.

plot_grid(tag_data, diff_smooths,
          ncol = 2,
          rel_widths = c(5,4))

```



```
tag_data <- plot_grid(zc19_218a_dive_RL,
                      zc19_218a_head,
                      zc19_218a_msa,
                      rel_heights = c(1.5,1,1),
                      ncol = 1,
                      align = 'hv')
```

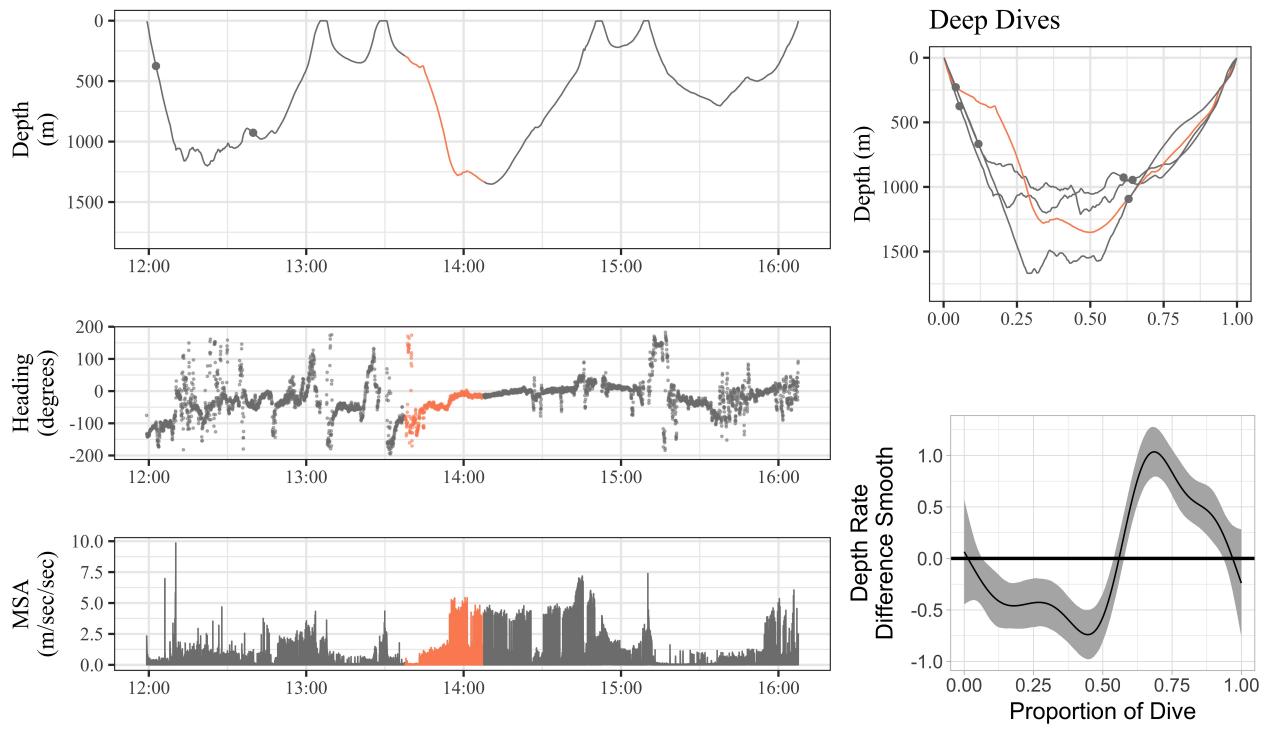
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
diff_smooths <- plot_grid(zc19_218a_overlaid_deep,
                           zc19_218a_depth_diffsmooth,
                           rel_heights = c(1,1),
                           ncol = 1, nrow = 2,
                           align = 'hv')
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
plot_grid(tag_data, diff_smooths,
          ncol = 2,
          rel_widths = c(2,1))
```



```

tag_data <- plot_grid(zc17_234a_dive_RL,
                      zc17_234a_head,
                      zc17_234a_msa,
                      rel_heights = c(1.5,1,1),
                      ncol = 1,
                      align = 'hv')

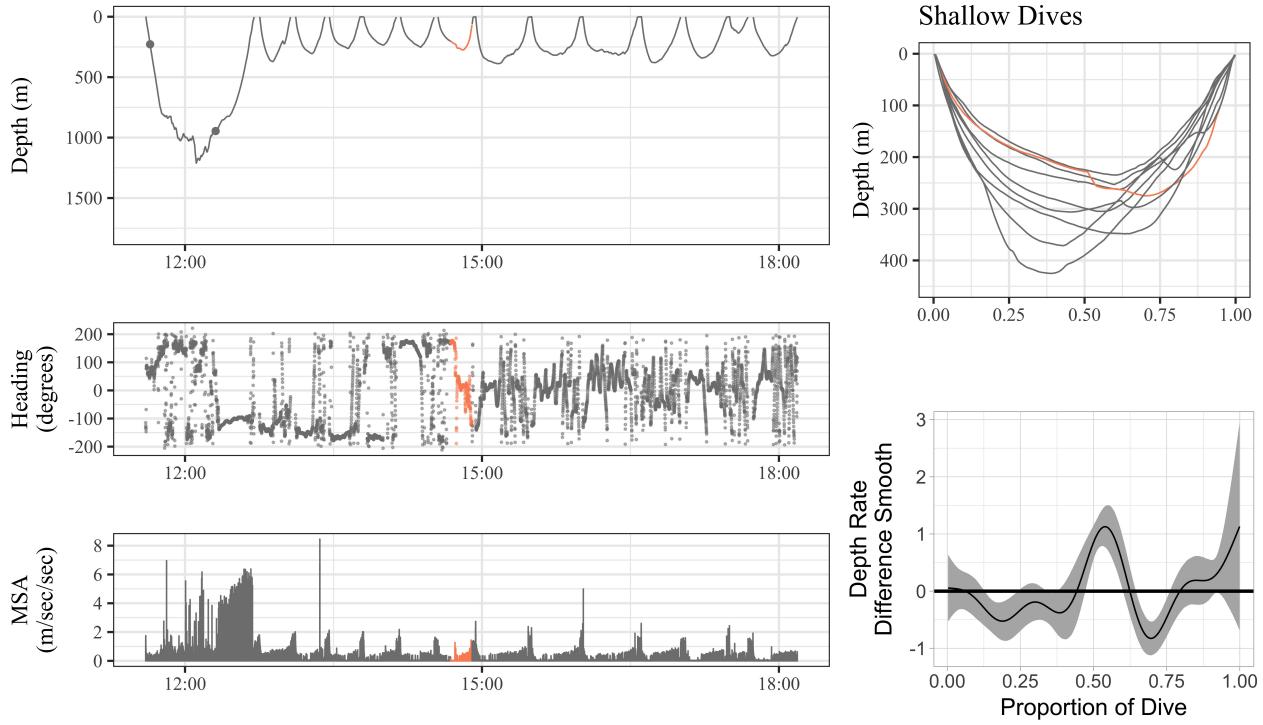
diff_smooths <- plot_grid(zc17_234a_overlaid_shallow,
                           zc17_234a_depth_diffsmooth,
                           rel_heights = c(1,1),
                           ncol = 1, nrow = 2,
                           align = 'hv')

## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
## Placing graphs unaligned.

plot_grid(tag_data, diff_smooths,
          ncol = 2,
          rel_widths = c(2,1))

```



## Graphs Version 2

- Convert time from UTC to local (-4 hours)
- Add actual RLs from file
- all whale data together: one col per whale, several rows each

```
rlfiles <- tibble(whaleID = c('zc17_234a',
                               'zc19_218a',
                               'zc20_232a'),
                  rlfle = c('data/Zc17_234a RLs.csv',
                           'data/Zc19_218a RLs.csv',
                           'data/Zc20_232a_RLs.csv'))

wids <- unique(rlfiles$whaleID)
audit_data <- list()
rl_data <- list()
for (w in c(1:length(wids))){
  this_afile <- paste0('data/audit/', wids[w], 'aud.txt')
  this_rlfle <- rlfiles |> filter(whaleID == wids[w]) |> pull('rlfle')
  these_rls <- read_csv(this_rlfle, show_col_types = FALSE, na = c('', ' ', "NA", "NaN"))
  if (file.exists(this_afile)){
    audit_data[[w]] <- read_delim(this_afile,
                                    delim = '\t',
                                    col_names = c('time', 'dur', 'comment')) |>
      mutate(time = parse_number(time)) |>
      filter(tolower(comment) == 'cee')
  }else{
    audit_data[[w]] <- data.frame(time = NA, dur = NA, comment = NA)
  }
  if (nrow(audit_data[[w]]) == nrow(these_rls)){
    rl_data[[w]] <- bind_cols(these_rls, audit_data[[w]])
  }
}
```

```

}else{
  if(nrow(these_rls) < nrow(audit_data[[w]])){
    n_na_pad <- nrow(audit_data[[w]]) - nrow(these_rls)
    these_rls <- bind_rows(NA * these_rls[1:n_na_pad,],
                           these_rls)

    rl_data[[w]] <- bind_cols(these_rls, audit_data[[w]])
  }else{
    rl_data[[w]] <- these_rls
  }
}
rl_data[[w]] <- mutate(rl_data[[w]], whaleID = wids[w])
}

## Rows: 30 Columns: 3
## -- Column specification -----
## Delimiter: "\t"
## chr (2): time, comment
## dbl (1): dur
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 70 Columns: 3
## -- Column specification -----
## Delimiter: "\t"
## chr (2): time, comment
## dbl (1): dur
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 66 Columns: 3
## -- Column specification -----
## Delimiter: "\t"
## chr (2): time, comment
## dbl (1): dur
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

all_RLs <- bind_rows(rl_data) |>
  select("SPL_rms",
         "SPL_Peak",
         "SNR",
         "whaleID",
         "time")

all_RLs <- left_join(all_RLs, CEE_meta, by = 'whaleID') |>
  mutate(cee_end_local = cee_end_time - lubridate::hours(4))

all_RLs <- all_RLs |>
  group_by(whaleID) |>
  mutate(ping_num = c(n() : 1)) |>
  ungroup()

```

```

all_RLs <- all_RLs |>
  mutate(ping_time_local = if_else(is.na(time),
    # if no ping time from audit, guess using 25s ping interval and CEE nominal
    cee_end_local - lubridate::seconds(25 * (ping_num - 1)),
    # if ping times from audit are there, use them
    tagon_time + lubridate::seconds(time) - lubridate::hours(4)),
  zRL = (160 - SPL_rms) / 60 * 1000 + 100) |>
  filter(SPL_rms > 50)

```

second axis plots

```

zc19_218a_dive_RL <- zc19_218a_dive_RL +
  lims(y = c(1800, 0)) +
  geom_point(data = all_RLs |> filter(whaleID == 'zc19_218a'),
    mapping = aes(x = ping_time_local, y = zRL),
    inherit.aes = FALSE,
    color = colrz[2]) +
  scale_y_reverse(
    "Depth (m)",
    # limits = c(1500,0), # can't set manual limits on double axis (?)
    sec.axis = sec_axis(~ -1 * ((. - 100) * 60 / 1000 - 160),
      name = "RL (dB SPL)"),
  )

```

```

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

```

```

zc17_234a_dive_RL <- zc17_234a_dive_RL +
  lims(y = c(1800, 0)) +
  geom_point(data = all_RLs |> filter(whaleID == 'zc17_234a'),
    mapping = aes(x = ping_time_local, y = zRL),
    inherit.aes = FALSE,
    color = colrz[2]) +
  scale_y_reverse(
    "Depth (m)",
    # limits = c(1500,0), # can't set manual limits on double axis (?)
    sec.axis = sec_axis(~ -1 * ((. - 100) * 60 / 1000 - 160),
      name = "RL (dB SPL)"),
  )

```

```

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

```

```

zc20_232a_dive_RL <- zc20_232a_dive_RL +
  lims(y = c(1800, 0)) +
  geom_point(data = all_RLs |> filter(whaleID == 'zc20_232a'),
    mapping = aes(x = ping_time_local, y = zRL),
    inherit.aes = FALSE,
    color = colrz[2]) +
  scale_y_reverse(
    "Depth (m)",
    # limits = c(1500,0), # can't set manual limits on double axis (?)

```

```

    sec.axis = sec_axis(~ -1 * ((. - 100) * 60 / 1000 - 160),
                        name = "RL (dB SPL)",

)

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

```

## All 3 whales

```

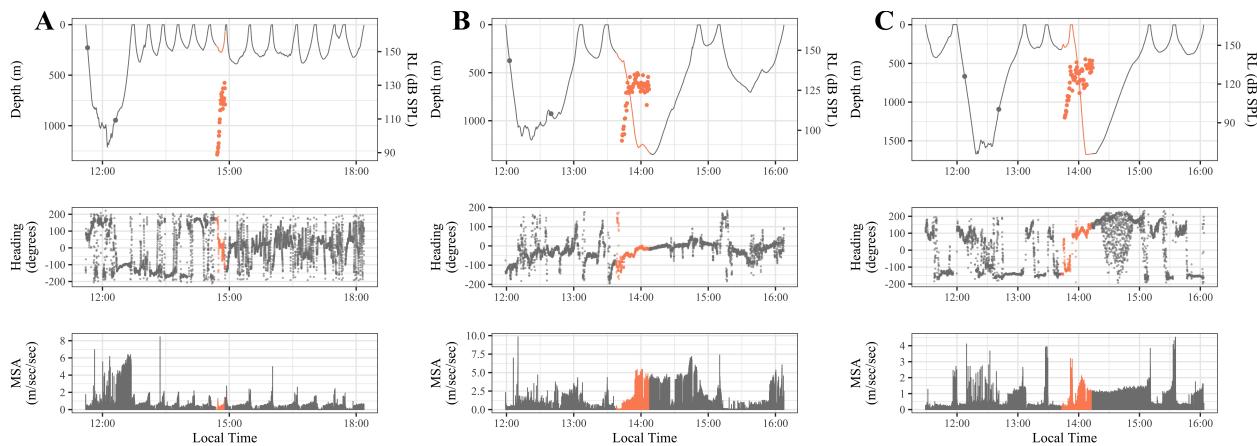
tag_data17 <- plot_grid(zc17_234a_dive_RL,
                         zc17_234a_head,
                         zc17_234a_msa + labs(x = 'Local Time'),
                         rel_heights = c(1.5,1,1),
                         ncol = 1,
                         align = 'hv')

tag_data19 <- plot_grid(zc19_218a_dive_RL,
                         zc19_218a_head,
                         zc19_218a_msa + labs(x = 'Local Time'),
                         rel_heights = c(1.5,1,1),
                         ncol = 1,
                         align = 'hv')

tag_data20 <- plot_grid(zc20_232a_dive_RL,
                         zc20_232a_head,
                         zc20_232a_msa + labs(x = 'Local Time'),
                         rel_heights = c(1.5,1,1),
                         ncol = 1,
                         align = 'hv')

plot_grid(tag_data17, tag_data19, tag_data20,
          labels = c('A', 'B', 'C'),
          label_size = 32, vjust = 1, hjust = -1.5,
          ncol = 3,
          rel_widths = c(1,1),
          align = 'hv')

```



## All smooths

- all overlaid dive traces + smooths together, dives on left, smooths on right

```
theme_set(theme_bw(base_size = 11))

zc17_234a_depth_diffsmoothb <-
  shallow_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
    covs = list(cee_dive_status = 'zc17_234aCEE'),
    term = "cee_dive_status",
    resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = '',
    y = 'Depth Rate\nDifference Smooth',
    x = ''
    # x = 'Proportion of Dive'
  ) |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc17_234a_depth_diffsmoothb)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 11
zc17_234a_depth_diffsmoothb <- ggplot_gtable(temp)

zc19_218a_depth_diffsmoothb <-
  deep_ddeptht_sde$plot_par("diveprop",
    par_names = "sigma",
    n_post = 1e4,
    covs = list(cee_dive_status = "zc19_218aCEE"),
    term = "cee_dive_status",
    resp = FALSE,
    show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = '',
    y = 'Depth Rate\nDifference Smooth',
    x = ''
    # x = 'Proportion of Dive'
  ) |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc19_218a_depth_diffsmoothb)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 11
zc19_218a_depth_diffsmoothb <- ggplot_gtable(temp)

zc20_232a_depth_diffsmooth1b <-
  shallow_ddepth_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
    covs = list(cee_dive_status = 'zc20_232aCEE'),
    term = "cee_dive_status",
    resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = '',
    y = 'Depth Rate\nDifference Smooth',
    x = ''
    # x = 'Proportion of Dive'
```

```

) |>
gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc20_232a_depth_diffsmooth1b)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 11
zc20_232a_depth_diffsmooth1b <- ggplot_gtable(temp)

zc20_232a_depth_diffsmooth2b <-
  deep_ddeptht_sde$plot_par("diveprop", par_names = "sigma", n_post = 1e4,
    covs = list(cee_dive_status = 'zc20_232aCEE'), term = "cee_dive_status",
    resp = FALSE, show_CI = "simultaneous") |>
  gf_theme(strip.text.y = element_blank()) |>
  gf_labs(title = '',
    y = 'Depth Rate\nDifference Smooth',
    x = 'Proportion of Dive'
  ) |>
  gf_hline(yintercept = 0, color = 'black', size = 1.1)

temp <- ggplot_build(zc20_232a_depth_diffsmooth2b)
temp$data[[1]]$fill = 'grey70'
temp$plot$theme$text$size = 11
zc20_232a_depth_diffsmooth2b <- ggplot_gtable(temp)

overlaid_dives <- plot_grid(zc17_234a_overlaid_shallow +
  labs(title = 'zc17_234, Shallow Dives'),
  zc19_218a_overlaid_deep +
  labs(title = 'zc19_218, Deep Dives'),
  zc20_232a_overlaid_shallow +
  labs(title = 'zc20_232, Shallow Dives'),
  zc20_232a_overlaid_deep +
  labs(title = 'zc20_232, Deep Dives',
    x = 'Proportion of Dive'),
  rel_heights = c(1,1,1,1),
  labels = c('A', 'B', 'C', 'D'),
  label_size = 22,
  nrow = 4,
  align = 'hv')

## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_line()`).

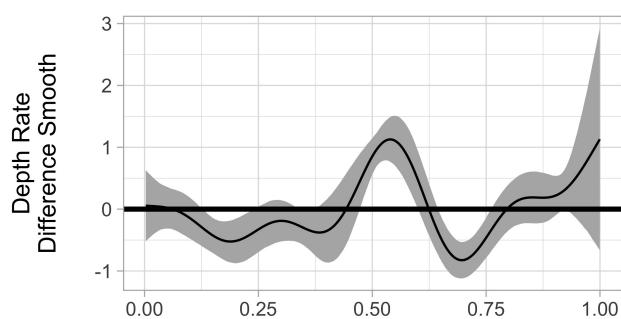
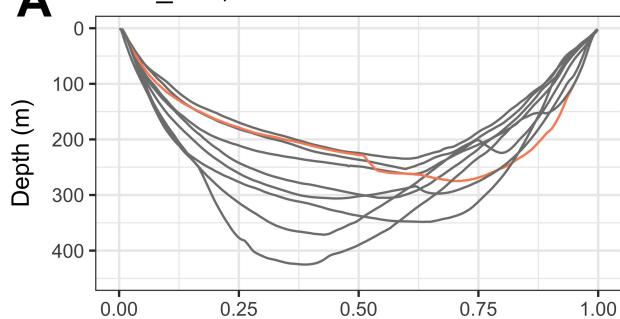
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).

diff_smooths <- plot_grid(zc17_234a_depth_diffsmoothb,
  zc19_218a_depth_diffsmoothb,
  zc20_232a_depth_diffsmooth1b,
  zc20_232a_depth_diffsmooth2b,
  rel_heights = c(1,1,1,1),

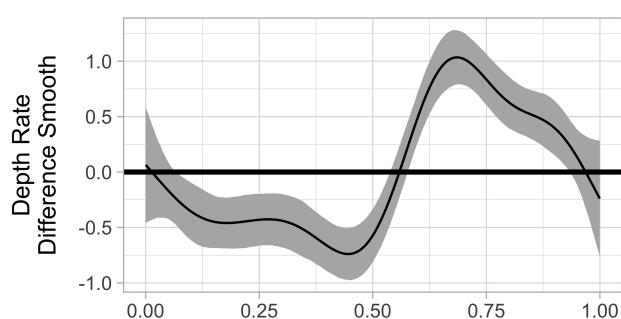
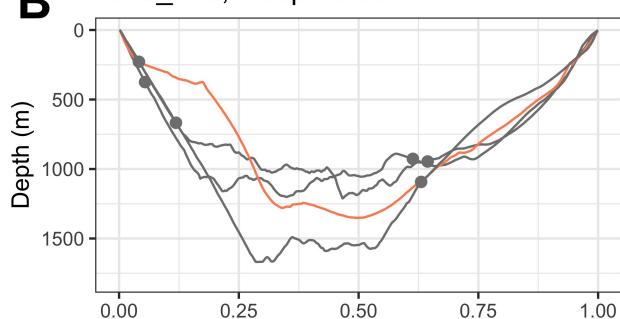
```

```
    ncol = 1,  
    align = 'hv')  
  
plot_grid(overlaid_dives,  
          diff_smooths,  
          rel_widths = c(1,1),  
          ncol = 2,  
          align = 'hv',  
          greedy = FALSE)
```

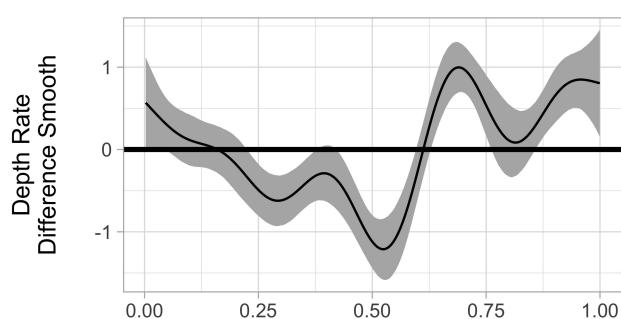
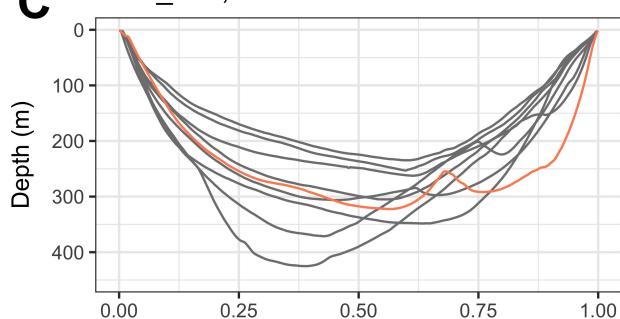
**A** zc17\_234, Shallow Dives



**B** zc19\_218, Deep Dives



**C** zc20\_232, Shallow Dives



**D** zc20\_232, Deep Dives

