ACLIM2 CMIP6 ROMSNPZ Indices quick start guide

K. Holsman

Contents

Download the ACLIM2 repo & data

Clone the ACLIM2 repo

To run this tutorial first clone the ACLIM2 repository to your local drive:

Option 1: Use R

This set of commands, run within R, downloads the ACLIM2 repository and unpacks it, with the ACLIM2 directory structrue being located in the specified `download_path`. This also performs the folder renaming mentioned in Option 2.

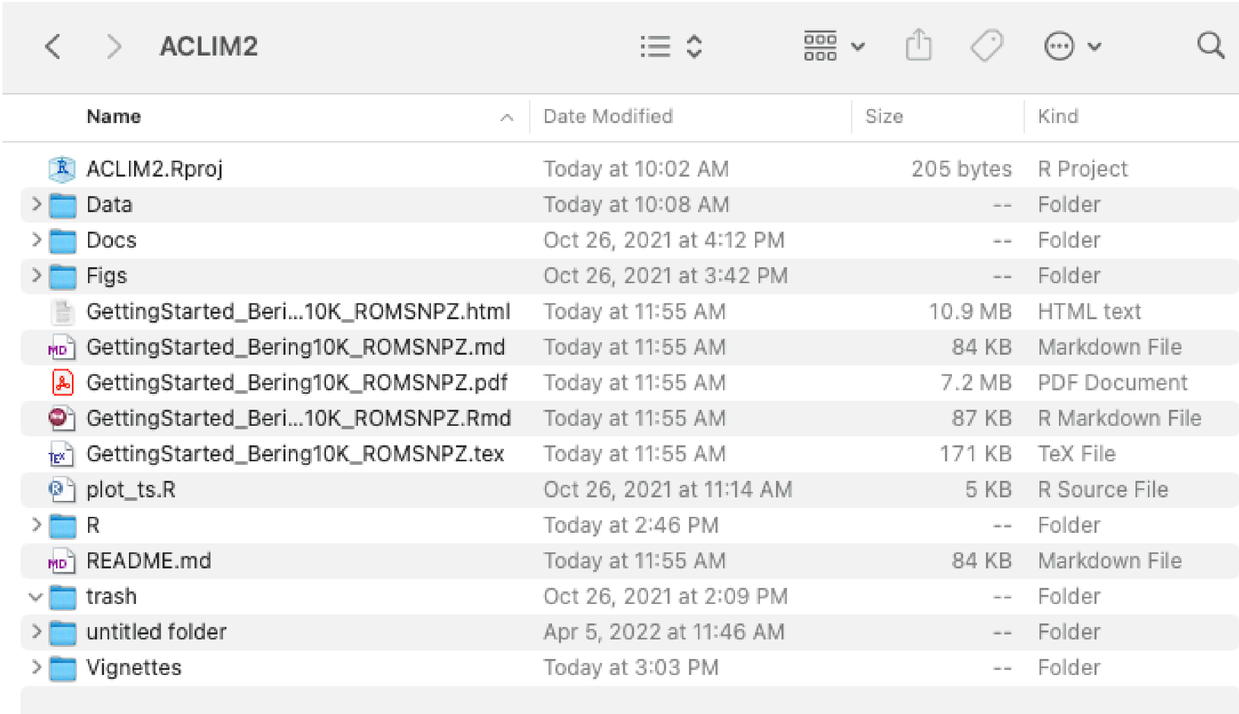
```
    # Specify the download directory
    main_nm       <- "ACLIM2"

    # Note: Edit download_path for preference
    download_path <-  path.expand("~")
    dest_fldr     <- file.path(download_path,main_nm)

    url           <- "https://github.com/kholsman/ACLIM2/archive/main.zip"
    dest_file     <- file.path(download_path,paste0(main_nm,".zip"))
    download.file(url=url, destfile=dest_file)

    # unzip the .zip file (manually unzip if this doesn't work)
    setwd(download_path)
    unzip (dest_file, exdir = download_path,overwrite = T)

    #rename the unzipped folder from ACLIM2-main to ACLIM2
    file.rename(paste0(main_nm,"-main"), main_nm)
    setwd(main_nm)


# Caption: Timeseries of season Aug East Bering Sea bottom temp or 400m temp (which ever is shallower) 
```

Option 2: Download the zipped repo

Download the full zip archive directly from the **ACLIM2 Repo** using this link: **https://github.com/kholsman/ACLIM2** and unzip its contents while preserving directory structure.

Important! If downloading from zip, please **rename the root folder** from `ACLIM2-main` (in the zipfile) to `ACLIM2` (name used in cloned copies) after unzipping, for consistency in the following examples.

Your final folder structure should look like this:



Option 3: Use git commandline

If you have git installed and can work with it, this is the preferred method as it preserves all directory structure and can aid in future updating. Use this from a **terminal command line, not in R**, to clone the full ACLIM2 directory and sub-directories:

```
git clone https://github.com/kholsman/ACLIM2.git
```

Get the data

Step 1)[can skip if not using .nc files directly] * Go to the google drive and download the zipped file with the R data `2022_03_07_Rdata.zip`:

- 00_ACLIM_shared > 02_Data > Newest_Data(use this) > unzip_and_putin_Data_in_folder.zip
- Unzip the folder and move the contents of the zipped folder to your local folder `ACLIM2/Data/in`.

Step 2) * Go to the google drive and download the zipped file with the R ACLIM2 indices `ACLIM2_indices.zip`:

- 00_ACLIM_shared > 02_Data > Newest_Data(use this) > unzip_and_putin_Data_out_folder.zip
- Unzip the folder and move the contents of the zipped folder to your local folder `ACLIM2/Data/out`.

Set up the Workspace

Open R() and used 'setwd()' to navigate to the root ACLIM2 folder (.e.g, ~/mydocuments/ACLIM2)

```r
# set the workspace to your local ACLIM2 folder
# e.g., "/Users/kholsman/Documents/GitHub/ACLIM2"
# setwd( path.expand("~/Documents/GitHub/ACLIM2") )

# --------------------------------------
# SETUP WORKSPACE
tmstp  <- format(Sys.time(), "%Y_%m_%d")
main   <- getwd()  #"~/GitHub_new/ACLIM2"

# loads packages, data, setup, etc.
suppressWarnings(source("R/make.R"))
```

```
## ------------------------------
## ALIM2/R/setup.R settings
## ------------------------------
## data_path            : D:/romsnpz/roms_for_public
## Rdata_path           : D:/romsnpz/2022_10_17_Rdata/roms_for_public
## redownload_level3_mox: FALSE
## update.figs          : FALSE
## load_gis             : FALSE
## update.outputs       : TRUE
## update.figs          : FALSE
## dpiIN                : 150
## update.figs          : FALSE
## ------------------------------
## ------------------------------
##
## The following datasets are public, please cite as Hermann et al. 2019 (v.H16) and Kearney et al. 2020
## B10K-H16_CMIP5_CESM_BIO_rcp85
## B10K-H16_CMIP5_CESM_rcp45
## B10K-H16_CMIP5_CESM_rcp85
## B10K-H16_CMIP5_GFDL_BIO_rcp85
## B10K-H16_CMIP5_GFDL_rcp45
## B10K-H16_CMIP5_GFDL_rcp85
## B10K-H16_CMIP5_MIROC_rcp45
## B10K-H16_CMIP5_MIROC_rcp85
## B10K-H16_CORECFS
## B10K-K20_CORECFS
##
## The following datasets are still under embargo, please do not share outside of ACLIM:
## B10K-K20P19_CMIP6_cesm_historical
## B10K-K20P19_CMIP6_cesm_ssp126
## B10K-K20P19_CMIP6_cesm_ssp585
## B10K-K20P19_CMIP6_gfdl_historical
## B10K-K20P19_CMIP6_gfdl_ssp126
## B10K-K20P19_CMIP6_gfdl_ssp585
## B10K-K20P19_CMIP6_miroc_historical
## B10K-K20P19_CMIP6_miroc_ssp126
## B10K-K20P19_CMIP6_miroc_ssp585
```

Read this before you start

Overview

The **ACLIM2 github repository** contains R code and Rdata files for working with netcdf-format data generated from the **downscaled ROMSNPZ modeling** of the ROMSNPZ Bering Sea Ocean Modeling team; Drs. Hermann, Cheng, Kearney, Pilcher,Ortiz, and Aydin. The code and R resources described in this tutorial are maintained by Kirstin Holsman as part of NOAA's **ACLIM project** for the Bering Sea. *See Hollowed et al. 2020 for more information about the ACLIM project.*

This document provides an overview of accessing, plotting, and creating bias corrected indices for ACLIM2 based on CMIP6 (embargoed for ACLIM2 users until 2023) and CMIP5 (publicly available) simulations. This guide assumes analyses will take place in R() and that users have access to the data folder within the ACLIM2 shared drive. For more information also see the full tutorial ("GettingStarted_Bering10K_ROMSNPZ" available at the bottom of **this repo page**.

Important! A few key things to know before getting started are detailed below. Please review this information before getting started.

ROMSNPZ versions

Important! ACLIM1 CMIP5 and ACLIM2 CMIP5 and CMIP6 datasets use different base models.

There are two versions of the ROMSNPZ model:

1. ACLIM1 an older 10-depth layer model used for CMIP5 ("H-16")
2. ACLIM2 a new 30-depth layer model used for CMIP6 ("K20" or "K20P19")

The models are not directly comparable, therefore the projections should be bias corrected and recentered to baselines of hindcasts of each model (forced by "observed" climate conditions). i.e. CMIP5 and CMIP6 have corresponding hindcasts:

1. Hindcast for CMIP5 "H19" –> H16_CORECFS
2. Hindcast for CMIP5 "K20P19" –> H16_CORECFS
3. Hindcast for CMIP6 "K20P19" –> K20_CORECFS

In addition for CMIP6 "historical" runs are available for bias correcting. We will use those below.

For a list of the available simulations for ACLIM enter the following in R():

```r
# list of the climate scenarios
data.frame(sim_list)
```

```
##                               sim_list
## 1                     B10K-K20_CORECFS
## 2        B10K-H16_CMIP5_CESM_BIO_rcp85
## 3            B10K-H16_CMIP5_CESM_rcp45
```

```
## 4           B10K-H16_CMIP5_CESM_rcp85
## 5       B10K-H16_CMIP5_GFDL_BIO_rcp85
## 6           B10K-H16_CMIP5_GFDL_rcp45
## 7           B10K-H16_CMIP5_GFDL_rcp85
## 8          B10K-H16_CMIP5_MIROC_rcp45
## 9          B10K-H16_CMIP5_MIROC_rcp85
## 10                   B10K-H16_CORECFS
## 11       B10K-K20P19_CMIP5_CESM_rcp45
## 12       B10K-K20P19_CMIP5_CESM_rcp85
## 13       B10K-K20P19_CMIP5_GFDL_rcp45
## 14       B10K-K20P19_CMIP5_GFDL_rcp85
## 15      B10K-K20P19_CMIP5_MIROC_rcp45
## 16      B10K-K20P19_CMIP5_MIROC_rcp85
## 17  B10K-K20P19_CMIP6_cesm_historical
## 18      B10K-K20P19_CMIP6_cesm_ssp126
## 19      B10K-K20P19_CMIP6_cesm_ssp585
## 20  B10K-K20P19_CMIP6_gfdl_historical
## 21      B10K-K20P19_CMIP6_gfdl_ssp126
## 22      B10K-K20P19_CMIP6_gfdl_ssp585
## 23 B10K-K20P19_CMIP6_miroc_historical
## 24     B10K-K20P19_CMIP6_miroc_ssp126
## 25     B10K-K20P19_CMIP6_miroc_ssp585
```

ROMSNPZ variables

For a list of the available variables from the ROMSNPZ:

```
# Metadata for variables
(srvy_var_def[-(1:5),])
```

```
##                    name                            units
## 6                   Ben                      mg C m^-2
## 7                DetBen                      mg C m^-2
## 8                  Hsbl                          meter
## 9               IceNH4                     mmol N m^-3
## 10              IceNO3                     mmol N m^-3
## 11              IcePhL                       mg C m^-3
## 12                 aice                               
## 13                 hice                          meter
## 14               shflux                   watt meter-2
## 15               ssflux                 meter second-1
## 16       Cop_integrated                (mg C m^-3)*m
## 17        Cop_surface5m                    mg C m^-3
## 18      EupO_integrated                (mg C m^-3)*m
## 19       EupO_surface5m                    mg C m^-3
## 20      EupS_integrated                (mg C m^-3)*m
## 21       EupS_surface5m                    mg C m^-3
## 22        Iron_bottom5m              micromol Fe m-3
## 23      Iron_integrated          (micromol Fe m-3)*m
## 24       Iron_surface5m              micromol Fe m-3
## 25       Jel_integrated                (mg C m^-3)*m
## 26        Jel_surface5m                    mg C m^-3
## 27       MZL_integrated                (mg C m^-3)*m
```

```
## 28        MZL_surface5m                            mg C m^-3
## 29      NCaO_integrated                      (mg C m^-3)*m
## 30       NCaO_surface5m                          mg C m^-3
## 31      NCaS_integrated                      (mg C m^-3)*m
## 32       NCaS_surface5m                          mg C m^-3
## 33         NH4_bottom5m                        mmol N m^-3
## 34       NH4_integrated                  (mmol N m^-3)*m
## 35        NH4_surface5m                        mmol N m^-3
## 36         NO3_bottom5m                        mmol N m^-3
## 37       NO3_integrated                  (mmol N m^-3)*m
## 38        NO3_surface5m                        mmol N m^-3
## 39       PhL_integrated                    (mg C m^-3)*m
## 40        PhL_surface5m                          mg C m^-3
## 41       PhS_integrated                    (mg C m^-3)*m
## 42        PhS_surface5m                          mg C m^-3
## 43  prod_Cop_integrated                    mg C m^-2 d^-1
## 44 prod_EupO_integrated                    mg C m^-2 d^-1
## 45 prod_EupS_integrated                    mg C m^-2 d^-1
## 46  prod_Eup_integrated (milligram carbon meter-3 d-1)*m
## 47  prod_Jel_integrated                    mg C m^-2 d^-1
## 48  prod_MZL_integrated                    mg C m^-2 d^-1
## 49 prod_NCaO_integrated                    mg C m^-2 d^-1
## 50 prod_NCaS_integrated                    mg C m^-2 d^-1
## 51  prod_NCa_integrated (milligram carbon meter-3 d-1)*m
## 52  prod_PhL_integrated                    mg C m^-2 d^-1
## 53  prod_PhS_integrated                    mg C m^-2 d^-1
## 54       salt_surface5m
## 55        temp_bottom5m                            Celsius
## 56      temp_integrated                        (Celsius)*m
## 57       temp_surface5m                            Celsius
## 58       uEast_bottom5m                    meter second-1
## 59      uEast_surface5m                    meter second-1
## 60      vNorth_bottom5m                    meter second-1
## 61     vNorth_surface5m                    meter second-1
##                                                        longname
## 6                                 Benthic infauna concentration
## 7                                Benthic detritus concentration
## 8                      depth of oceanic surface boundary layer
## 9                                   Ice ammonium concentration
## 10                                   Ice nitrate concentration
## 11                                     Ice algae concentration
## 12                              fraction of cell covered by ice
## 13                                average ice thickness in cell
## 14                                        surface net heat flux
## 15                           surface net salt flux, (E-P)*SALT
## 16          Small copepod concentration, integrated over depth
## 17                Small copepod concentration, surface 5m mean
## 18    Offshore euphausiid concentration, integrated over depth
## 19          Offshore euphausiid concentration, surface 5m mean
## 20    On-shelf euphausiid concentration, integrated over depth
## 21          On-shelf euphausiid concentration, surface 5m mean
## 22                            iron concentration, bottom 5m mean
## 23                   iron concentration, integrated over depth
## 24                           iron concentration, surface 5m mean
```

```
## 25               Jellyfish concentration, integrated over depth
## 26                     Jellyfish concentration, surface 5m mean
## 27       Microzooplankton concentration, integrated over depth
## 28             Microzooplankton concentration, surface 5m mean
## 29 Offshore large copepod concentration, integrated over depth
## 30       Offshore large copepod concentration, surface 5m mean
## 31 On-shelf large copepod concentration, integrated over depth
## 32       On-shelf large copepod concentration, surface 5m mean
## 33                       Ammonium concentration, bottom 5m mean
## 34                Ammonium concentration, integrated over depth
## 35                      Ammonium concentration, surface 5m mean
## 36                        Nitrate concentration, bottom 5m mean
## 37                 Nitrate concentration, integrated over depth
## 38                       Nitrate concentration, surface 5m mean
## 39    Large phytoplankton concentration, integrated over depth
## 40          Large phytoplankton concentration, surface 5m mean
## 41    Small phytoplankton concentration, integrated over depth
## 42          Small phytoplankton concentration, surface 5m mean
## 43                  Cop net production rate, summed over depth
## 44                 EupO net production rate, summed over depth
## 45                 EupS net production rate, summed over depth
## 46    secondary production Euphausiids, integrated over depth
## 47                  Jel net production rate, summed over depth
## 48                  MZL net production rate, summed over depth
## 49                 NCaO net production rate, summed over depth
## 50                 NCaS net production rate, summed over depth
## 51     secondary production Neocalanus, integrated over depth
## 52                  PhL net production rate, summed over depth
## 53                  PhS net production rate, summed over depth
## 54                                     salinity, surface 5m mean
## 55                     potential temperature, bottom 5m mean
## 56              potential temperature, integrated over depth
## 57                    potential temperature, surface 5m mean
## 58          u-momentum component, geo-rotated, bottom 5m mean
## 59         u-momentum component, geo-rotated, surface 5m mean
## 60          v-momentum component, geo-rotated, bottom 5m mean
## 61         v-momentum component, geo-rotated, surface 5m mean
```

Data outputs

Important! There are 2 types of post-processed data available for use in ACLIM.

The ROMSNPZ team has developed a process to provide standardized post-processed outputs from the large (and non-intuitive) ROMSNPZ grid. These have been characterized as:

1. Level 1 (original ROMSNPZ U,V, grid, not rotated or corrected)

2. Level 2 (lat long bi-weekly high res versions, shouldn't be needed and are difficult to work with)

3. **Level 3 indices (depth corrected and area weighted means for each model variable; i.e., what we will mostly use)**

   a. "ACLIMsurveyrep_": groundifsh survey replicated (replicated in space and time)
   b. "ACLIMregion_": weekly strata based averages

To get more information about each of these level 3 datasets enter this in R:

```
# Metadata for Weekly ("ACLIMregion_...") indices
head(all_info1)
```

```
##                            name                   Type B10KVersion  CMIP  GCM
## 1 B10K-H16_CMIP5_CESM_BIO_rcp85 Weekly regional indices         H16 CMIP5 CESM
## 2     B10K-H16_CMIP5_CESM_rcp45 Weekly regional indices         H16 CMIP5 CESM
## 3     B10K-H16_CMIP5_CESM_rcp85 Weekly regional indices         H16 CMIP5 CESM
## 4 B10K-H16_CMIP5_GFDL_BIO_rcp85 Weekly regional indices         H16 CMIP5 GFDL
## 5     B10K-H16_CMIP5_GFDL_rcp45 Weekly regional indices         H16 CMIP5 GFDL
## 6     B10K-H16_CMIP5_GFDL_rcp85 Weekly regional indices         H16 CMIP5 GFDL
##     BIO Carbon_scenario               Start                 End nvars
## 1  TRUE           rcp85 2006-01-22 12:00:00 2099-12-27 12:00:00    59
## 2 FALSE           rcp45 2006-01-22 12:00:00 2081-02-16 12:00:00    59
## 3 FALSE           rcp85 2006-01-22 12:00:00 2099-12-27 12:00:00    59
## 4  TRUE           rcp85 2006-01-22 12:00:00 2099-12-27 12:00:00    59
## 5 FALSE           rcp45 2006-01-22 12:00:00 2099-12-27 12:00:00    59
## 6 FALSE           rcp85 2006-01-22 12:00:00 2099-12-27 12:00:00    59
```

```
# Metadata for Weekly ("ACLIMsurveyrep_...") indices
head(all_info2)
```

```
##                            name              Type B10KVersion  CMIP  GCM   BIO
## 1 B10K-H16_CMIP5_CESM_BIO_rcp85 Survey replicated         H16 CMIP5 CESM  TRUE
## 2     B10K-H16_CMIP5_CESM_rcp45 Survey replicated         H16 CMIP5 CESM FALSE
## 3     B10K-H16_CMIP5_CESM_rcp85 Survey replicated         H16 CMIP5 CESM FALSE
## 4 B10K-H16_CMIP5_GFDL_BIO_rcp85 Survey replicated         H16 CMIP5 GFDL  TRUE
## 5     B10K-H16_CMIP5_GFDL_rcp45 Survey replicated         H16 CMIP5 GFDL FALSE
## 6     B10K-H16_CMIP5_GFDL_rcp85 Survey replicated         H16 CMIP5 GFDL FALSE
##   Carbon_scenario Start  End nvars
## 1           rcp85  1970 2100    60
## 2           rcp45  1970 2100    60
## 3           rcp85  1970 2100    60
## 4           rcp85  1970 2100    60
## 5           rcp45  1970 2100    60
## 6           rcp85  1970 2100    60
```

Indices & bias correction

ACLIM2 Indices

The next step creates ACLIM2 indices (i.e., Level4) based on the Level3 output for each hindcast, historical run, and CMIP6 projection. The script below then bias corrects each index using the historical run and recenters the projection on the corresponding hindcast (such that projections are Δ from historical mean values for the reference period `deltayrs <- 1970:2000`).

NESB & SEBS averaged indices The average water column values for each variable from the ROMSNPZ model strata x weekly Level2 outputs ('ACLIMregion') was calculated and used to calculate the strata-area weighted mean value for the NEBS and SEBS weekly, monthly, seasonally, and annually. Similarly, for survey replicated ('ACLIMsurveyrep') Level2 outputs the average water column value for each variable at each station was calculated used to calculate the strata-area weighted mean value for the NEBS and SEBS annually. These indices were calculate for hindcast, historical, and projection scenarios, and used to bias correct the projections. More information on the methods for each can be found in the tabs below and the code immediately following this section will re-generate the bias corrected indices. All of the bias corrected outputs can be found in the “Data/out/CMIP6” folder.

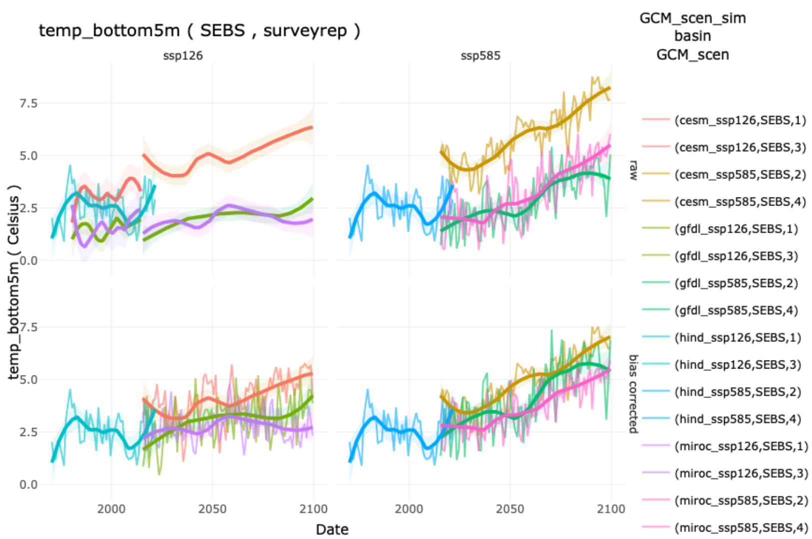


Figure 1: **Raw (top row) and bias corrected (bottom row)bottom temperature indices based on survey replicated Level3 outputs for the SEBS**

Important! Note that for projections the ‘mn_val’ represents raw mean values, while ‘val_biascorrected’ is the bias corrected mn_val (should be used instead of the raw values). In all cases, for variables that are log-normally distributed (cannot be < 0), the ln(mn_val) were used to bias correct and were then back transformed to non-log space after correction:

For normally distributed variables (Y):

$$Y_{t,k}^{fut'} = \bar{Y}_{k,\bar{T}}^{hind} + \left(\frac{\sigma_{k,\bar{T}}^{hind}}{\sigma_{k,\bar{T}}^{hist}} * (Y_{t,k}^{fut} - \bar{Y}_{k,\bar{T}}^{hist}) \right)$$

where $\bar{Y}_{y,k}^{fut'}$ is the bias corrected varable k value for time-step t (e.g., year, month, or season), $\bar{Y}_{k,\bar{T}}^{hind}$ is the mean value of the variable k during the reference period $\bar{T} = [1980, 2013]$ from the hindcast model, $\sigma_{k,\bar{T}}^{hind}$ is the standard deviation of the hindcast during the reference period $\bar{T}$, $\sigma_{k,\bar{T}}^{hist}$ is the standard deviation of the historical run during tje reference period, $Y_{t,k}^{fut}$ is the value of the variable from the projection at time-step t and $\bar{Y}_{k,\bar{T}}^{hist}$ is the average value from the historical run during reference period $\bar{T}$.

For log-normally distributed variables(Y):

$$Y_{y,k}^{fut'} = e^{\ln \bar{Y}_{k,\bar{T}}^{hind} + \left(\frac{\hat{\sigma}_{k,\bar{T}}^{hind}}{\hat{\sigma}_{k,\bar{T}}^{hist}} * (\ln Y_{t,k}^{fut} - \ln \bar{Y}_{k,\bar{T}}^{hist}) \right)}$$

, where $\hat{\sigma}_{k,\bar{T}}^{hist}$ and $\hat{\sigma}_{k,\bar{T}}^{hind}$ are the standard deviation of the $\ln \bar{Y}_{k,t}^{hist}$ and $\ln \bar{Y}_{k,t}^{hind}$ during the reference period $\hat{T}$ (respectively).

Weekly indices

Uses the strata x weekly data (‘ACLIMregion’) to generate strata-specific averages in order to generate the strata area-weighted averages for each week w each year y.

$$\bar{Y}_{w,y,k} = \frac{\sum_{l}^{n_s}(\frac{1}{n_i}\sum_{t}^{n_t}Y_{k,w,y,s,t}) * A_s}{\sum_{s}^{n_s}A_s}$$

, where $Y_{k,w,y,s,t}$ is the value of the variable k in strata s at time t in year y, A_s is the area of strata s, n_i is the number of stations in strata s, and n_s is the number of strata s in each basin (NEBS or SEBS).

$\bar{Y}_{w,y,k}$ was calculated for the hindcast, historical run, and projection time-series. For projections $\bar{Y}_{w,y,k}$ was bias corrected using the corresponding historical and hindcast values such that:

$$\bar{Y}^{fut'}_{w,y,k} = \bar{Y}^{hind}_{w,k} + \left(\frac{\sigma^{hind}_{w,k}}{\sigma^{hist}_{w,k}} * (\bar{Y}^{fut}_{w,y,k} - \bar{Y}^{hist}_{w,k})\right)$$

, where $\bar{Y}^{hist}_{w,k}$ and $\bar{Y}^{hind}_{w,k}$ are the average historical weekly values across years in the period (1980 to 2012 ; adjustable in `R/setup.R`).

Monthly indices

Uses the strata x weekly data ('ACLIMregion') to generate strata-specific averages in order to generate the strata area-weighted averages for each month m each year y.

$$\bar{Y}_{m,y,k} = \frac{1}{n_w}\sum_{w}^{n_w}\bar{Y}_{w,y,k}$$

, where $\bar{Y}_{w,y,k}$ are the weekly average indices for variable k in year y from the previous step ,n_w is the number of weeks in each month m.

$\bar{Y}_{m,y,k}$ was calculated for the hindcast, historical run, and projection time-series. For projections $\bar{Y}_{m,y,k}$ was bias corrected using the corresponding historical and hindcast values such that:

$$\bar{Y}^{fut'}_{m,y,k} = \bar{Y}^{hind}_{m,k} + \left(\frac{\sigma^{hind}_{m,k}}{\sigma^{hist}_{m,k}} * (\bar{Y}^{fut}_{m,y,k} - \bar{Y}^{hist}_{m,k})\right)$$

, where $\bar{Y}^{hist}_{m,k}$ and $\bar{Y}^{hind}_{m,k}$ are the average historical monthly values across years in the period (1980 to 2012 ; adjustable in `R/setup.R`).

Seasonal indices

Uses the strata x weekly data ('ACLIMregion') to generate strata-specific averages in order to generate the strata area-weighted averages for each season l each year y.

$$\bar{Y}_{l,y,k} = \frac{1}{n_w}\sum_{w}^{n_w}\bar{Y}_{w,y,k}$$

, where $\bar{Y}_{w,y,k}$ are the weekly average indices for variable k in year y from the previous step ,n_w is the number of weeks in each season l.

$\bar{Y}_{l,y,k}$ was calculated for the hindcast, historical run, and projection time-series. For projections $\bar{Y}_{l,y,k}$ was bias corrected using the corresponding historical and hindcast values such that:

$$\bar{Y}^{fut'}_{l,y,k} = \bar{Y}^{hind}_{l,k} + \left(\frac{\sigma^{hind}_{l,k}}{\sigma^{hist}_{l,k}} * (\bar{Y}^{fut}_{l,y,k} - \bar{Y}^{hist}_{l,k})\right)$$

, where $\bar{Y}^{hist}_{l,k}$ and $\bar{Y}^{hind}_{l,k}$ are the average historical seasonal values across years in the reference period (1980 to 2012 ; adjustable in `R/setup.R`).

Annual indices

Uses the strata x weekly data ('ACLIMregion') to generate strata-specific averages in order to generate the strata area-weighted averages for each season l each year y.

$$\bar{Y}_{y,k} = \frac{1}{n_w} \sum_{w}^{n_w} \bar{Y}_{w,y,k}$$

, where $\bar{Y}_{w,y,k}$ are the weekly average indices for variable k in year y from the previous step ,n_w is the number of weeks in each year y.

$\bar{Y}_{y,k}$ was calculated for the hindcast, historical run, and projection time-series. For projections $\bar{Y}_{y,k}$ was bias corrected using the corresponding historical and hindcast values such that:

$$\bar{Y}_{y,k}^{fut'} = \bar{Y}_{k}^{hind} + \left(\frac{\sigma_k^{hind}}{\sigma_k^{hist}} * (\bar{Y}_{y,k}^{fut} - \bar{Y}_{k}^{hist}) \right)$$

, where $\bar{Y}_{k}^{hind}$ and $\bar{Y}_{k}^{hist}$ are the average historical values across years in the reference period (1980 to 2012 ; adjustable in `R/setup.R`).

Annual survey rep. indices

Uses the station specific survey replicated (in time and space) data ('ACLIMsurveyrep') to generate strata-specific averages in order to generate the strata area-weighted averages for each year y.

$$\bar{Y}_{y,k} = \frac{\sum_{l}^{n_s} (\frac{1}{n_i} \sum_{i}^{n_i} Y_{k,y,s,i}) * A_s}{\sum_{s}^{n_s} A_s}$$

, where $Y_{k,y,s,i}$ is the value of the variable k at station i in strata s in year y, A_s is the area of strata s, n_i is the number of stations in strata s, and n_s is the number of strata s in each basin (NEBS or SEBS).

$\bar{Y}_{y,k}$ was calculated for the hindcast, historical run, and projection time-series. For projections $\bar{Y}_{y,k}$ was bias corrected using the corresponding historical and hindcast values such that:

$$\bar{Y}_{y,k}^{fut'} = \bar{Y}_{k}^{hind} + \left(\frac{\sigma_k^{hind}}{\sigma_k^{hist}} * (\bar{Y}_{y,k}^{fut} - \bar{Y}_{k}^{hist}) \right)$$

, where $\bar{Y}_{k}^{hind}$ and $\bar{Y}_{k}^{hist}$ are the average historical values across years in the reference period (1980 to 2012 ; adjustable in `R/setup.R`).

Appendix A includes the code used to generate the ACLIM2 indices and bias correct them. That code can be run to re-make the indices if you like but takes approx 30 mins a CMIP to run.

Plot & concat Indices

The following code will open an interactive shiny() app for exploring the indices. You can also view this online at (kkh2022.shinyapps.io/ACLIM2_indices)[https://kkh2022.shinyapps.io/ACLIM2_indices/].

```
tmpwd<-getwd()
setwd("R/shiny_aclim/ACLIM2_indices")
shiny::runApp("app.R")
setwd(tmpwd)
```

```r
# alternatively you can extract the data you want using the get_var()function

df <- get_var(typeIN = "annual",plotvar = "temp_bottom5m",plothist = F)

df$plot
head(df$dat)
```

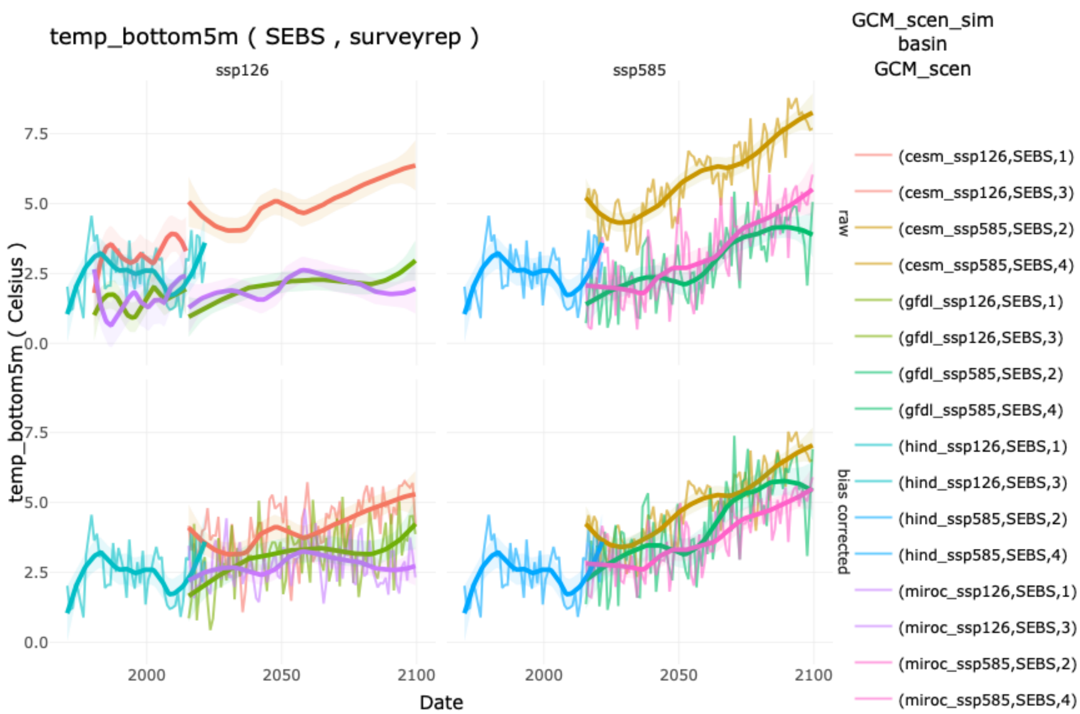
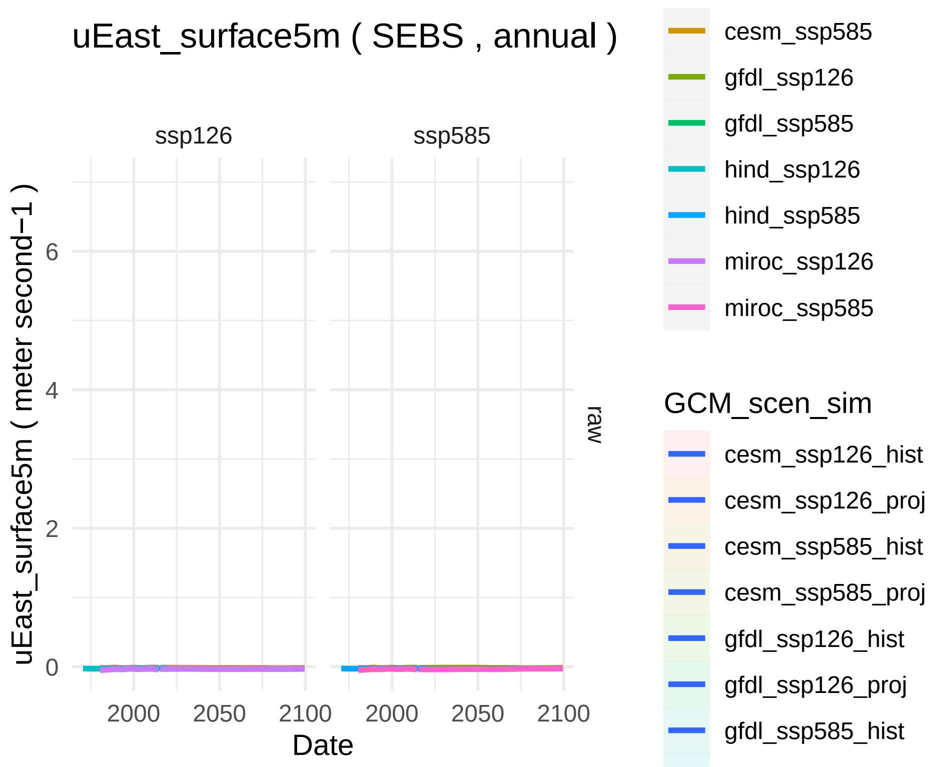


Figure 2: "Raw (top row) and bias corrected (bottom row)bottom temperature indices based on survey replicated Level3 outputs for the SEBS"

NRS indices (André)

```r
    suppressMessages(source("R/make.R"))
    # preview possible variables

    #load(file = "Data/out/weekly_vars_C.Rdata")
    load(file = "Data/out/weekly_vars.Rdata")
    #load(file = "Data/out/srvy_vars_C.Rdata")
    load(file = "Data/out/srvy_vars.Rdata")

    load(paste0("Data/out/K20P19_CMIP6/allEBS_means/ACLIM_annual_hind_mn.Rdata"))
    varall  <- unique(ACLIM_annual_hind$var)
    varall

    scens   <- c("ssp126", "ssp585")
    GCMs    <- c("miroc", "gfdl", "cesm" )
```

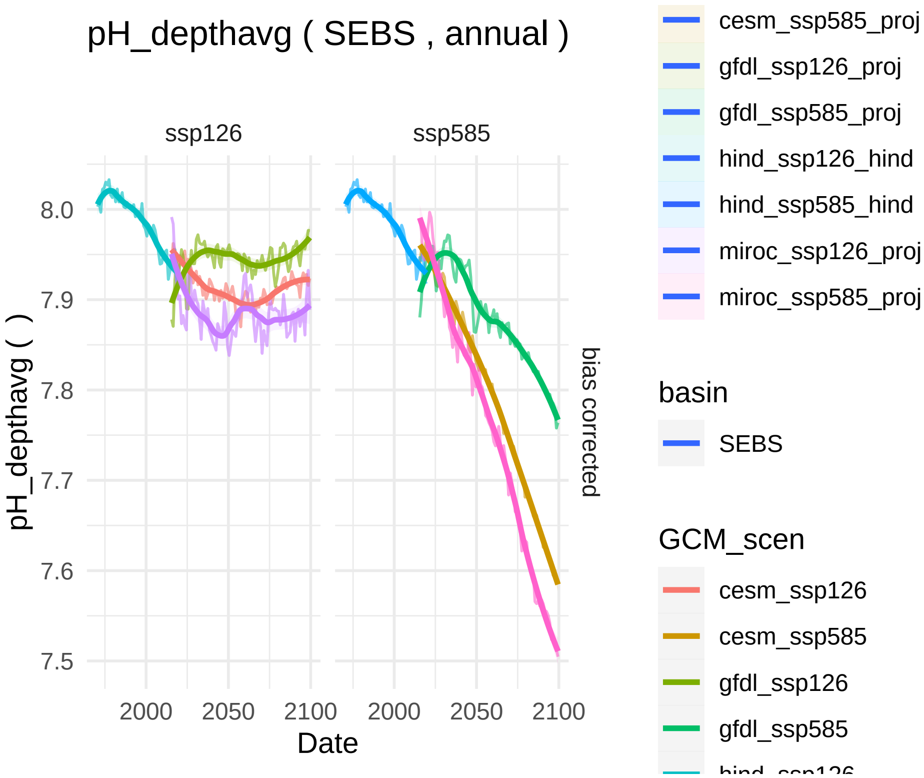
```r
varlist <- c("temp_bottom5m", "fracbelow2", "uEast_surface5m", "vNorth_surface5m")

# get the variable you want:
df <- get_var( typeIN     = "annual",
               plotvar    = varlist[3],
               bcIN       = "raw",
               CMIPIN     = "K20P19_CMIP6",
               plothist   = T,  # ignore the hist runs
               removeyr1  = T)  # "Remove first year of projection ( burn in)")

df$plot+coord_cartesian(ylim = c(0, 7))
```




```r
head(df$dat)
```

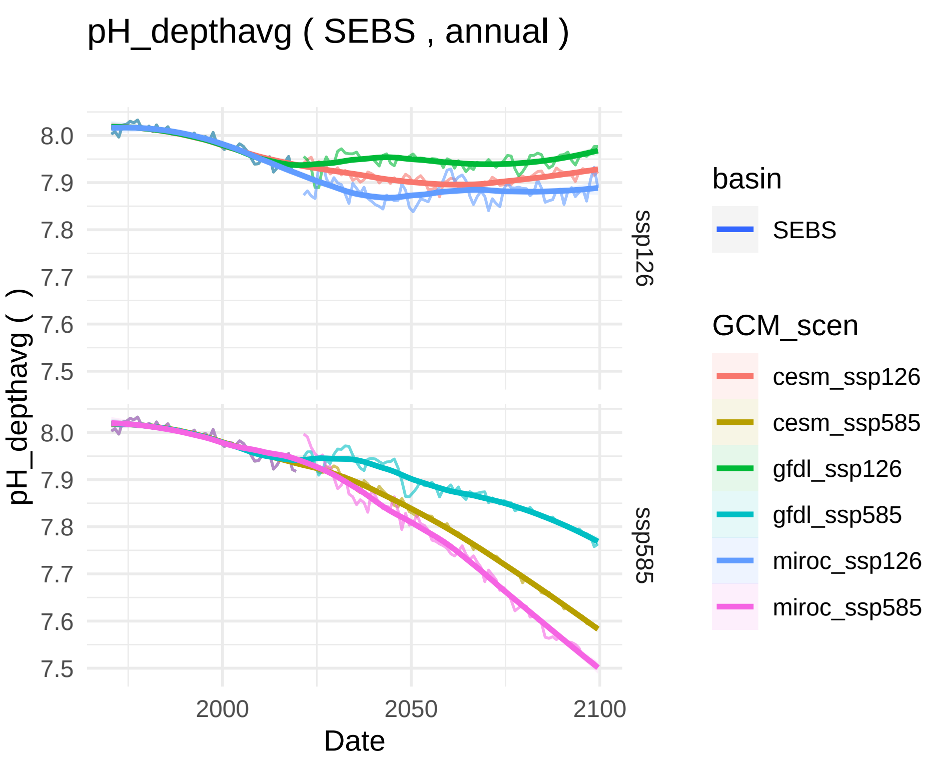
Continuous timeseries of hind + fut

```r
# get the variable you want:
df <- get_var( typeIN     = "annual",
               plotvar    = "pH_depthavg",
               bcIN       = "bias corrected",
               CMIPIN     = "K20P19_CMIP6",
               plothist   = F,  # ignore the hist runs
               removeyr1  = T)  # "Remove first year of projection ( burn in)")

df$plot
```

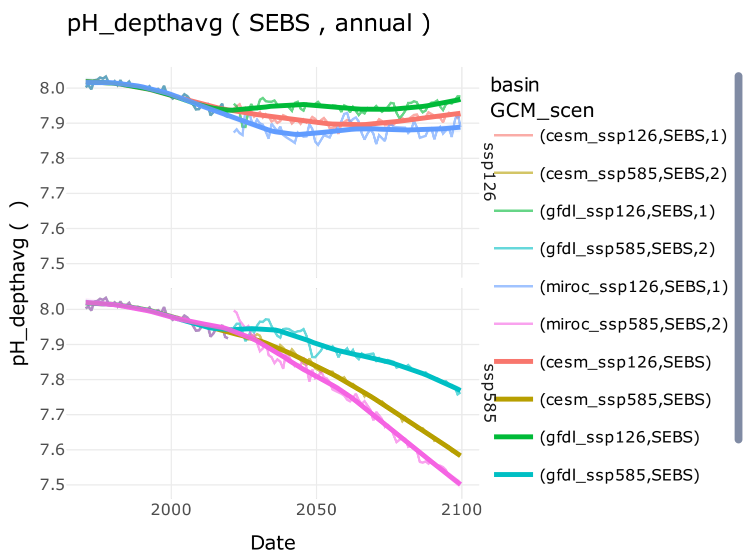



```
    head(df$dat)

# concat the hind and fut runs by removing years from projection
maxDin <- max(as.vector(df$dat%>%
                          dplyr::filter(sim_type=="hind")%>%ungroup()%>%
                          dplyr::select(mnDate))[[1]])

newdat <- stitchTS(dat = df$dat,
                   maxD  = maxDin)

# newdat has the full set of data
# select miroc_ssp126
head(newdat%>%dplyr::filter(GCM_scen==paste0(GCMs[1],"_",scens[1])))
tail(newdat%>%dplyr::filter(GCM_scen==paste0(GCMs[1],"_",scens[1])))

pp  <-  ggplot(newdat) +
        geom_line(aes(x=mnDate,y=mn_val,color= GCM_scen, linetype = basin),
                    alpha = 0.6,show.legend = FALSE) +
        geom_smooth(aes(x = mnDate,
                        y = mn_val,
                        color     = GCM_scen,
                        fill      = GCM_scen,
                        linetype = basin),
                    alpha       = 0.1,
                    method      = "loess",
                    formula     = 'y ~ x',
                    span        = .5,
                    show.legend = T) +
          theme_minimal() +
```

```
          labs(x = "Date",
               y = paste(newdat$var[1],"(",newdat$units[1],")"),
                 subtitle = "",
                 legend   = "",
                 title    = paste(newdat$var[1],"(",newdat$basin[1],",",newdat$type[1],")"))+
        scale_color_discrete() +
        facet_grid(scen~.)
pp
```



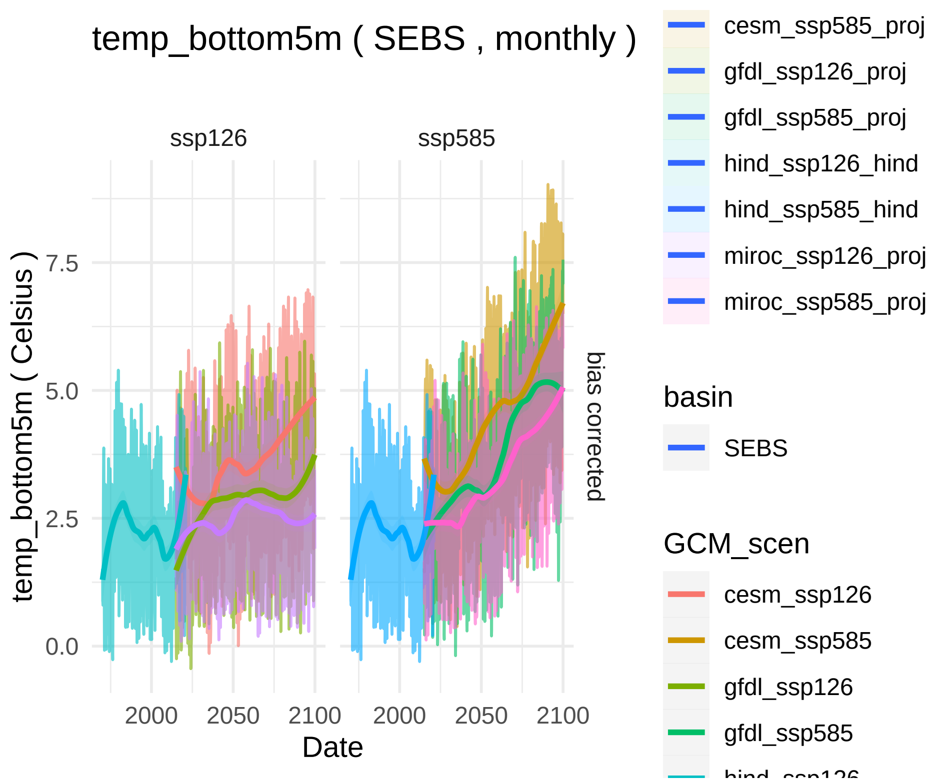

```
# plot it interactively
plotly::ggplotly(pp)
```



monthly indices (Andy)

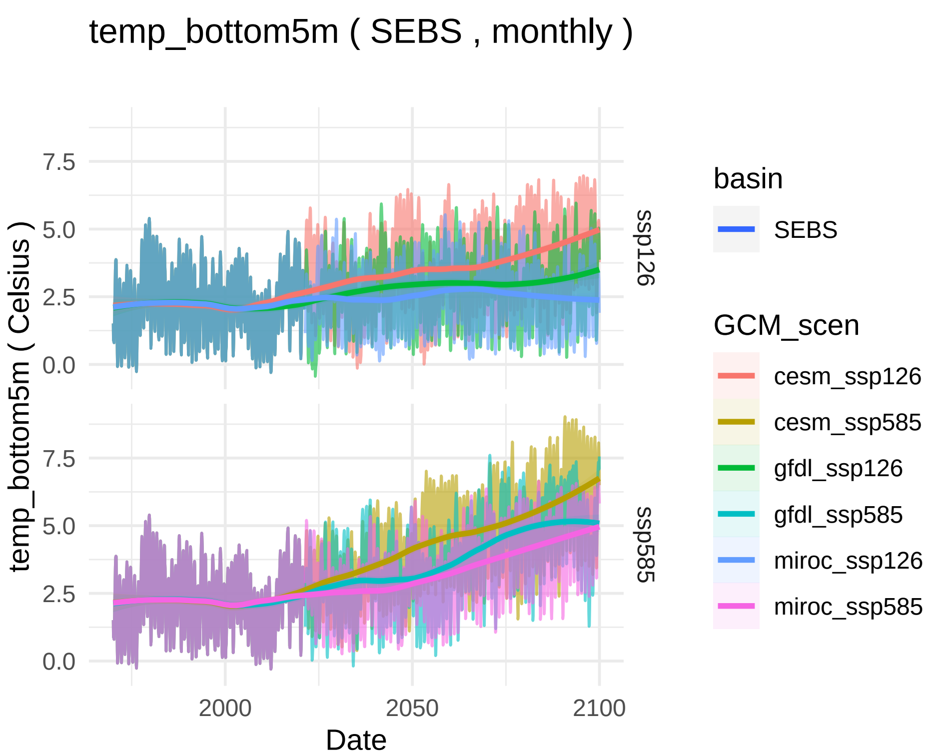
```
suppressMessages(source("R/make.R"))

# preview possible variables
load(paste0("Data/out/K20P19_CMIP6/allEBS_means/ACLIM_monthly_hind_mn.Rdata"))
varall  <- unique(ACLIM_monthly_hind$var)
varall

scens   <- c("ssp126","ssp585")
GCMs    <- c("miroc","gfdl",  "cesm" )
varlist <- c("temp_bottom5m","fracbelow2","uEast_surface5m")

# get the variable you want:
df <- get_var( typeIN    = "annual",
               CMIPIN    = "K20P19_CMIP6",
               plotvar   = "uEast_surface5m",
               bcIN      = "bias corrected",
               plothist  = F,  # ignore the hist runs
               removeyr1 = T)  #"Remove first year of projection ( burn in)")

df <- get_var( typeIN    = "monthly",
```

```
            CMIPIN     = "K20P19_CMIP6",
            monthIN    = 2,
            plotvar    = "temp_bottom5m",
            bcIN       = "bias corrected",
            plothist   = F,  # ignore the hist runs
            removeyr1 = T)  #"Remove first year of projection ( burn in)")


head(df$dat)
df$plot
```

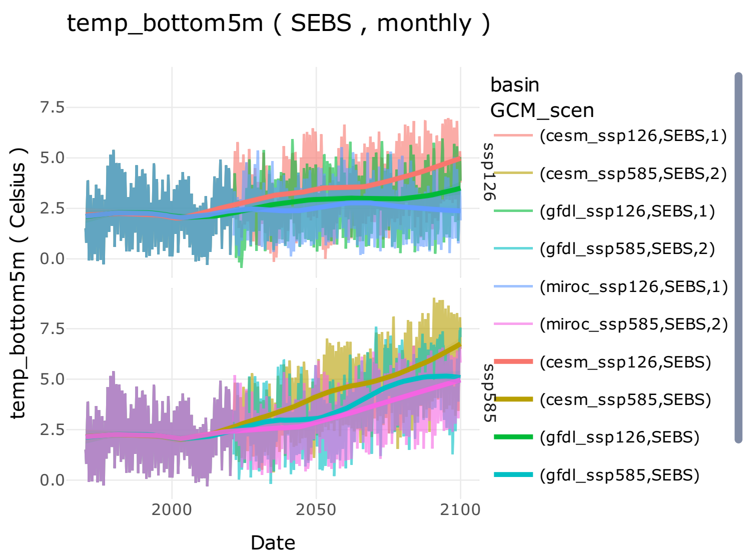



```
# concat the hind and fut runs by removing years from projection
maxDin <- max(as.vector(df$dat%>%dplyr::filter(sim_type=="hind")%>%
                          ungroup()%>%dplyr::select(mnDate))[[1]])

newdat <- stitchTS(dat = df$dat,
                 maxD  = maxDin)

# newdat has the full set of data
# select miroc_ssp126
head(newdat%>%dplyr::filter(GCM_scen==paste0(GCMs[1],"_",scens[1])))
tail(newdat%>%dplyr::filter(GCM_scen==paste0(GCMs[1],"_",scens[1])))


pp  <- ggplot(newdat)+
          geom_line(aes(x=mnDate,y=mn_val,color= GCM_scen, linetype = basin),
                    alpha = 0.6,show.legend = FALSE)+
          geom_smooth(aes(x=mnDate,y=mn_val,color= GCM_scen,
```

```
                                fill=GCM_scen,linetype = basin),alpha=0.1,
                            method="loess",formula='y ~ x',span = .5,show.legend=T)+
            theme_minimal() +
            labs(x="Date",
                   y=paste(newdat$var[1],"(",newdat$units[1],")"),
                   subtitle = "",
                   legend = "",
                   title = paste(newdat$var[1],"(",newdat$basin[1],",",newdat$type[1],")"))+
          scale_color_discrete()+
          facet_grid(scen~.)
pp
```

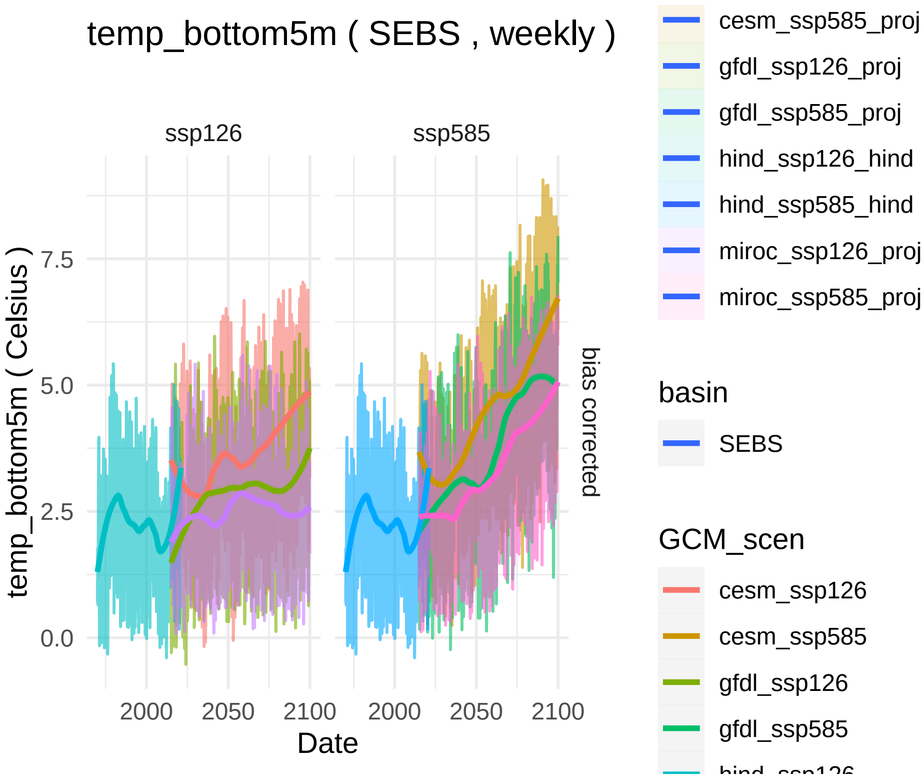



```
# plot it interactively
plotly::ggplotly(pp)
```

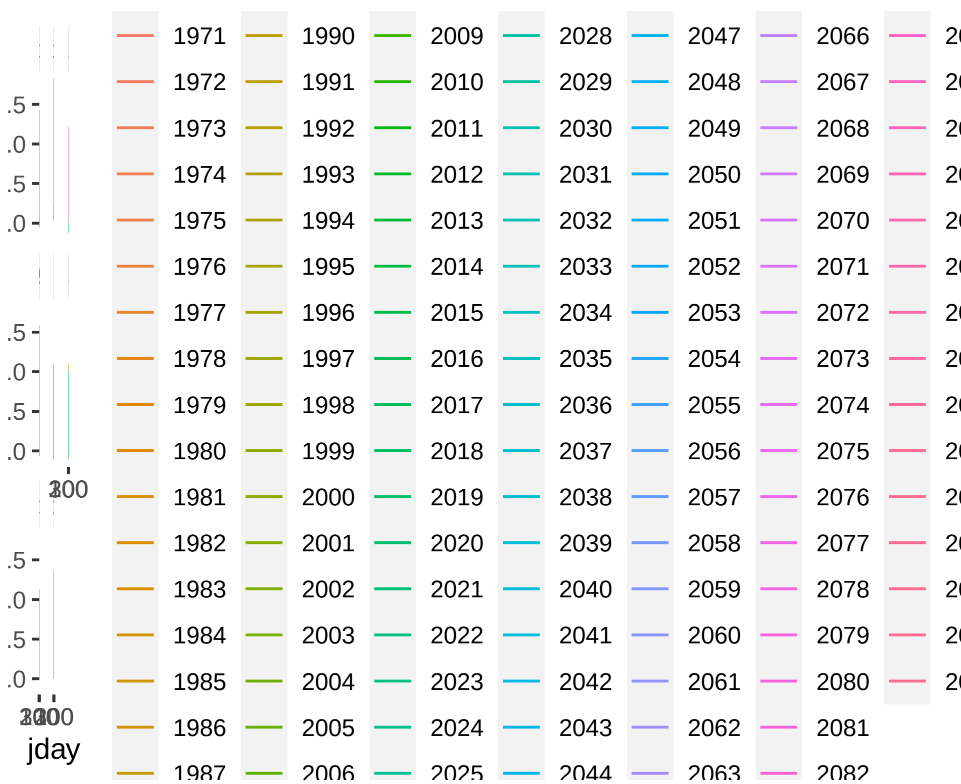


weekly indices (Jon)

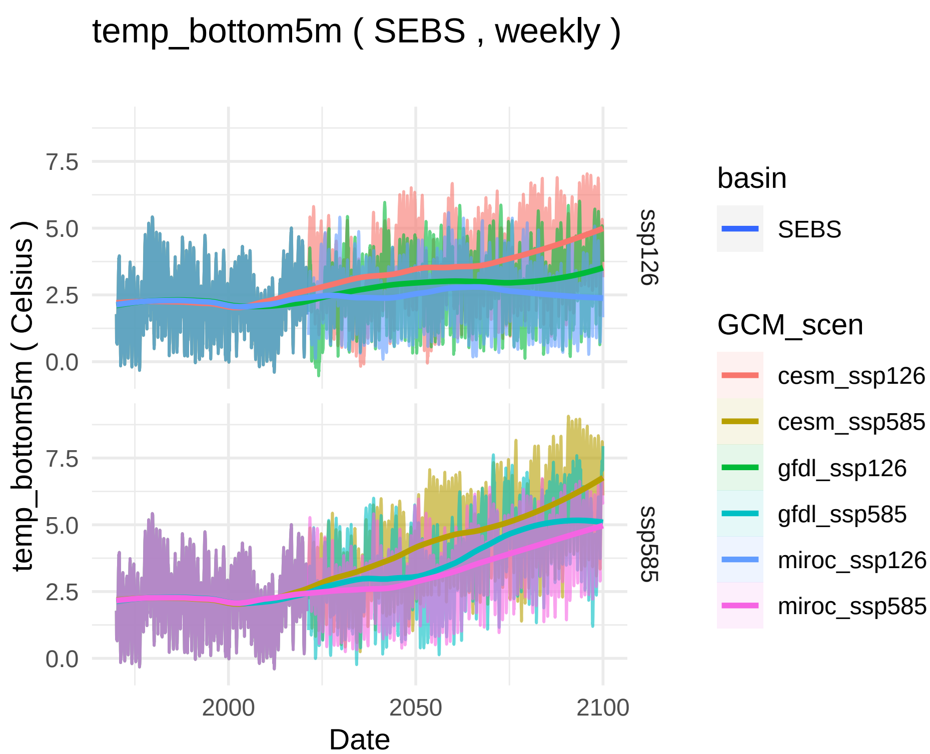
```
suppressMessages(source("R/make.R"))

# preview possible variables
load(paste0("Data/out/K20P19_CMIP6/allEBS_means/ACLIM_weekly_hind_mn.Rdata"))
varall  <- unique(ACLIM_weekly_hind$var)
varall


scens   <- c("ssp126","ssp585")
GCMs    <- c("miroc","gfdl",  "cesm" )
varlist <- c("temp_bottom5m","fracbelow2","uEast_surface5m")

# get the variable you want:
df <- get_var( typeIN     = "weekly",
               plotvar    = "temp_bottom5m",
               bcIN       = "bias corrected",
               plothist   = F,  # ignore the hist runs
               removeyr1  = T)  #"Remove first year of projection ( burn in)")

df$plot
```



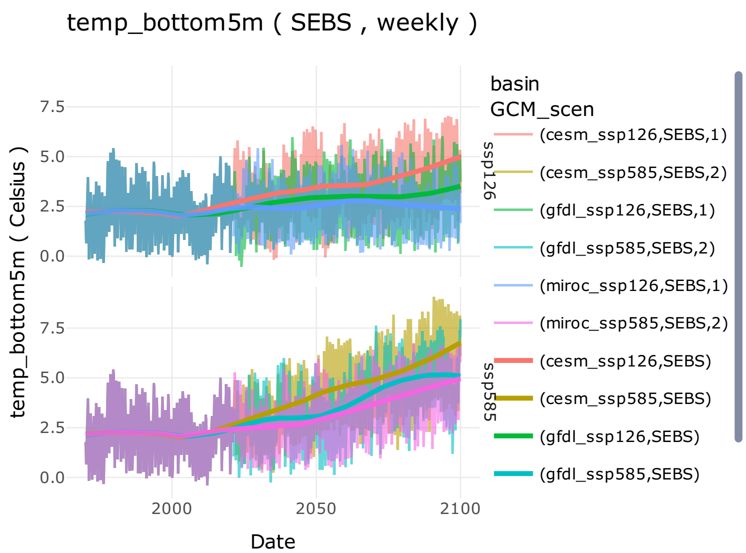

```
head(df$dat)

ggplot(df$dat%>%filter(basin=="SEBS"))+ geom_line(aes(x=jday, y= mn_val, color=factor(year)))+facet_wr
```



```r
# concat the hind and fut runs by removing years from projection
maxDin <- max(as.vector(df$dat%>%dplyr::filter(sim_type=="hind")%>%dplyr::select(mnDate))[[1]])

newdat <- stitchTS(dat = df$dat,
                  maxD  = maxDin)

# newdat has the full set of data
# select miroc_ssp126
head(newdat%>%dplyr::filter(GCM_scen==paste0(GCMs[1],"_",scens[1])))


pp  <- ggplot(newdat)+
          geom_line(aes(x=mnDate,y=mn_val,color= GCM_scen, linetype = basin),
                    alpha = 0.6,show.legend = FALSE)+
          geom_smooth(aes(x=mnDate,y=mn_val,color= GCM_scen,
                          fill=GCM_scen,linetype = basin),alpha=0.1,
                      method="loess",formula='y ~ x',span = .5,show.legend=T)+
          theme_minimal() +
          labs(x="Date",
                 y=paste(newdat$var[1],"(",newdat$units[1],")"),
                 subtitle = "",
                 legend = "",
                 title = paste(newdat$var[1],"(",newdat$basin[1],",",newdat$type[1],")"))+
        scale_color_discrete()+
        facet_grid(scen~.)
# plot it
pp
```



```r
# plot it interactively
plotly::ggplotly(pp)
```



Output to .dat file (ADMB/ TMB users)

For CEATTLE I create a .dat file that is read into the ADMB script. That .dat file includes the bias corrected values (e.g., bottom temperature in deg C) used for the bioenergetics and temperature-dependent growth functions as well as Z-score (scaled) values used as covariates on the recruitment function. The section below will step through that .dat file creation for a subset of variables as well as demo chunks of ADMB code for reading that into a ADMB based model.

Use R to make .dat file using ACLIM suite

```r
# 1 -- create .dat filename & path
# 2 -- rescale (Z-score) data and get variables
# 3 -- write data to hind .dat file
# 3 -- write data to fut  .dat file
```

```r
# 1 -- create .dat filename & path
# -------------------------------------
suppressMessages(source("R/make.R"))


# switches
  thisYr <- format(Sys.time(), "%Y")
  today  <- format(Sys.time(), "%b %d, %Y")
  lastyr_hind <- as.numeric(thisYr)  #2021
  hind_yrs    <- 1979:lastyr_hind   # define the years of your estimation model
  fut_yrs     <- (lastyr_hind+1):2100   # define the years of your projections
  log_adj     <- 1e-4
  plotbasin   <- "SEBS"

# Define the name for the .dat file
  file.name   <- "ACLIM2_CMIP6_short"
  fn          <- paste(file.name,"_bcs.dat",sep="")
  archive_old <- T  # Archive the older version of the .dat file?
  normlist    <- read.csv(file=file.path(Rdata_path,"../normlist.csv"))

  outpath     <- "Data/out/ADMB_datfiles"
  if(!dir.exists(outpath)) dir.create(outpath)

# define hind and fut data files
  fndat_hind  <- file.path(outpath,paste("KKHhind_",fn,sep=""))
  fndat_fut   <- file.path(outpath,paste("KKHfut_",fn,sep=""))
  fndat_hind2 <- file.path(outpath,paste("hind_",fn,sep=""))
  fndat_fut2  <- file.path(outpath,paste("fut_",fn,sep=""))

  # create and archive .dat files
    outfile    <- fndat_fut
  if(file.exists(outfile)&archive_old){
      # archive older version
          archivefl <- paste0(substr(outfile,start=1,stop=nchar(outfile)-4),
                          format(Sys.time(), "%Y%m%d_%H%M%S"),".dat")
          file.rename(outfile, archivefl)
          #file.remove(outfile)
  }

  file.create(outfile)
  outfile  <- fndat_hind
  if(file.exists(outfile)&archive_old){
      # archive older version
          archivefl <- paste0(substr(outfile,start=1,stop=nchar(outfile)-4),
                          format(Sys.time(), "%Y%m%d_%H%M%S"),".dat")
          file.rename(outfile, archivefl)
          #file.remove(outfile)
  }

  file.create(outfile)

  # 2 -- rescale (Z-score) data and get variables
  # CMIPS <- c("K20P19_CMIP6","K20P19_CMIP5")
```

```r
  # CMIPS <- c("K20P19_CMIP6_C")

  CMIPS <- c("K20P19_CMIP6","K20P19_CMIP5")

  # preview possible variables
  load(paste0("Data/out/",CMIPS[1],"/allEBS_means/ACLIM_annual_hind_mn.Rdata"))
  varall  <- unique(ACLIM_annual_hind$var)
  varall

  # get each variable, convert to TS and rbind

  for(c in 1:length(CMIPS)){

    # first for annual mean values:
      varlist <- c("largeZoop_integrated","fracbelow2",
                   "temp_bottom5m","temp_surface5m","pH_depthavg")
      varlist <- c("largeZoop_integrated","fracbelow2",
                   "temp_bottom5m","temp_surface5m","pH_depthavg")

    # norm_sub <- normlist%>%filter(var%in%varlist)
      typeIN <- "annual"
      load(paste0("Data/out/",CMIPS[c],"/allEBS_means/ACLIM_",typeIN,"_hind_mn.Rdata"))
      load(paste0("Data/out/",CMIPS[c],"/allEBS_means/ACLIM_",typeIN,"_fut_mn.Rdata"))
      eval(parse(text = paste0("dhind <- ACLIM_",typeIN,"_hind")))
      eval(parse(text = paste0("dfut  <- ACLIM_",typeIN,"_fut")))

    # Z-score the data
     tmphind    <- suppressWarnings(dhind%>%
       dplyr::filter(var%in%varlist,basin==plotbasin,year%in%hind_yrs)%>%
     # dplyr::left_join(normlist,by=c("var"="var"))%>%
       dplyr::select(var,basin,year,mnDate,mn_val,
                     mnVal_hind,sdVal_hind, sim,gcmcmip,
                     CMIP,GCM,scen,sim_type ,units,long_name)%>%
       dplyr::mutate(bc = "bias corrected",
                     GCM_scen = paste0(GCM,"_",scen),
                     mn_val_scaled =  (mn_val-mnVal_hind )/sdVal_hind ))

     tmpfut    <-  suppressWarnings(dfut%>%
       dplyr::filter(var%in%varlist,basin==plotbasin,year%in%fut_yrs)%>%
       dplyr::select(var,basin,year,mnDate,mn_val, val_biascorrected,
                     mnVal_hind,sdVal_hind, sim,gcmcmip,
                     CMIP,GCM,scen,sim_type ,units,long_name)%>%
       dplyr::mutate(bc = "bias corrected",
                     GCM_scen = paste0(GCM,"_",scen),
                     mn_val_scaled =  (val_biascorrected-mnVal_hind )/sdVal_hind))

     if(1==2){

        hind2 <- rbind(
          tmphind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
          tmphind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "b) bias corrected"),
          tmphind%>%select(year,var, mn_val_scaled, scen, GCM)%>%
            rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))
```

```
    fut2 <- rbind(
      tmpfut%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
      tmpfut%>%select(year,var, val_biascorrected, scen, GCM)%>%
        rename(mn_val = val_biascorrected)%>%mutate(type = "b) bias corrected"),
      tmpfut%>%select(year,var, mn_val_scaled, scen, GCM)%>%
        rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))

    pcompare <- ggplot(hind2)+
     geom_line(aes(x=year,y=mn_val ,color=scen),size=.8)+
     geom_line(data =fut2,aes(x=year,y=mn_val,color=scen, linetype = GCM ),size=.8)+
     facet_wrap(var~type,scales="free_y",nrow =length(varlist))+theme_minimal()
    pcompare

    sclr <-1.2
    png("Figs/compare_scaled2raw_2022.png",
        width =6*sclr, height =9*sclr, units = "in",res = 150)
    print( pcompare)  # hypoxic (O2<70mmol m-3) or suboxic (O2<5mmol m-3),
    dev.off()


 }

# now for seasonal mean values:

typeIN    <- "seasonal"
seasonsIN <- unique(seasons$season)

load(paste0("Data/out/",CMIPS[c],"/allEBS_means/ACLIM_",typeIN,"_hind_mn.Rdata"))
load(paste0("Data/out/",CMIPS[c],"/allEBS_means/ACLIM_",typeIN,"_fut_mn.Rdata"))
eval(parse(text = paste0("dhind <- ACLIM_",typeIN,"_hind")))
eval(parse(text = paste0("dfut  <- ACLIM_",typeIN,"_fut")))


 # z-score the data
 tmphind2    <- dhind%>%
   dplyr::filter(var%in%varlist,basin==plotbasin,
                 year%in%hind_yrs,
                 season%in%seasonsIN)%>%
   dplyr::mutate(var = paste0(var,"_",season))%>%
    dplyr::select(var,basin,year,mnDate,mn_val,
                 mnVal_hind,sdVal_hind, sim,gcmcmip,
                 CMIP,GCM,scen,sim_type ,units,long_name)%>%
   dplyr::mutate(bc = "bias corrected",
                 GCM_scen = paste0(GCM,"_",scen),
                 mn_val_scaled = (mn_val-mnVal_hind )/sdVal_hind)

 tmpfut2    <- dfut%>%
   dplyr::filter(var%in%varlist,basin==plotbasin,
                 year%in%fut_yrs,season%in%seasonsIN)%>%
   dplyr::mutate(var = paste0(var,"_",season))%>%
   dplyr::select(var,basin,year,mnDate,mn_val, val_biascorrected,
                 mnVal_hind,sdVal_hind, sim,gcmcmip,
                 CMIP,GCM,scen,sim_type ,units,long_name)%>%
```

```
      dplyr::mutate(bc = "bias corrected",
                    GCM_scen = paste0(GCM,"_",scen),
                    mn_val_scaled = (val_biascorrected-mnVal_hind )/
                      (sdVal_hind))

    if(c ==1){
       hind  <- rbind(tmphind,tmphind2)
       fut   <- rbind(tmpfut,tmpfut2)
    }else{
       hind  <- rbind(hind,tmphind,tmphind2)
       fut   <- rbind(fut,tmpfut,tmpfut2)
    }

  }

  if(1==2){

       hind2 <- rbind(
         hind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
         hind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "b) bias corrected"),
         hind%>%select(year,var, mn_val_scaled, scen, GCM)%>%
           rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))

       fut2 <- rbind(
         fut%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
         fut%>%select(year,var, val_biascorrected, scen, GCM)%>%
           rename(mn_val = val_biascorrected)%>%mutate(type = "b) bias corrected"),
         fut%>%select(year,var, mn_val_scaled, scen, GCM)%>%
           rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))

       pcompare2 <- ggplot(hind2[grep("largeZoop",x=hind2$var),])+
         geom_line(aes(x=year,y=mn_val ,color=scen),size=.8)+
         geom_line(data =fut2[grep("largeZoop",x=fut2$var),],
                   aes(x=year,y=mn_val,color=scen, linetype = GCM ),size=.8)+
         facet_wrap(var~type,scales="free_y",ncol =3)+theme_minimal()
       pcompare2

       sclr <-1.4
       png("Figs/compare_scaled2raw_2022_all.png",
           width =6*sclr, height =9*sclr, units = "in",res = 150)
       print( pcompare2)  # hypoxic (O2<70mmol m-3) or suboxic (O2<5mmol m-3),
       dev.off()


  }



    # plot the data
     p <- ggplot(hind)+
       geom_line(aes(x=mnDate,y=mn_val,color=GCM_scen))+
       geom_line(data=fut,aes(x=mnDate,y=mn_val,color=GCM_scen))+
       facet_wrap(.~var,scales="free_y")+
```

```
      theme_minimal()
  p

  # plot the data
   p_scaled <- ggplot(hind)+
      geom_line(aes(x=mnDate,y=mn_val_scaled,color=GCM_scen))+
      geom_line(data=fut,aes(x=mnDate,y=mn_val_scaled,color=GCM_scen))+
      facet_wrap(.~var,scales="free_y")+
      theme_minimal()
  p_scaled

  # now identify which covars are highly correlated

  #convert wide matrix
  d_wide   <- reshape2::dcast(hind%>%dplyr::filter(year!=thisYr)%>%
                              select(year, var, mn_val_scaled),
                           year~var,mean,
                           value.var = "mn_val_scaled")


  # calculate correlations and display in column format
  #
  num_col  <- ncol(d_wide[,-1])
  out_indx <- which(upper.tri(diag(num_col)))
  cor_cols <- d_wide %>%
              do(melt(cor(.[,-1],
                         method="spearman",
                         use="pairwise.complete.obs"),
                      value.name="cor")[out_indx,])

  corr     <- cor(na.omit(d_wide[,-1]))

  long_dat <- reshape2::melt(corr,variable.name = "variable") %>%
  as.data.frame()

  # plot co variation between variables
  corplot <- long_dat %>%arrange(value)%>%
                ggplot(aes(x=Var1, y=Var2, fill=value)) +
                geom_raster() +
                scale_fill_viridis_c()+
                theme_minimal()+
                theme(axis.text.x = element_text(angle = 90))

  # # remove those where cov is high (temp by season and cold pool by season)
  # subset <- long_dat$>$filter(abs(value)<0.6)

# 3 -- write data to hind .dat file
# ------------------------------------


  # CEATTLE uses a spp overlap index - you can skip this

            overlapdat <- data.frame(
```

```
              atf_OL=c(0.9391937,0.8167094,0.808367,0.5926875,0.7804481,0.5559549,
                       0.4006931,0.5881404,0.7856776,0.511565,0.6352048,0.5583476,
                       0.5792738,0.5417657,0.8212887,0.6287613,0.4536608,0.6587292,
                       0.4884194,0.8289379,0.4399257,0.5950167,0.6388434,0.6111834,
                       0.8742649,0.7868746,0.8024257,0.6227457,0.4956742,0.4347917,
                       0.4791108,0.4369006,0.5613625,0.4353015),
              south_OL=c(0.9980249,0.9390368,0.9959974,0.6130846,0.951234,0.5851891,
                         0.4934879,0.641471,0.9809618,0.5596813,0.7196964,0.6754698,
                         0.5774808,0.6041351,0.9406521,0.7949525,0.5306435,0.7977694,
                         0.5345031,0.9879945,0.5079171,0.7148121,0.8997132,0.7340859,
                         0.9962068,0.9627235,0.998043,0.8111,0.6087638,0.513057,0.5492621,
                         0.4971361,0.665453,0.5969653)
              )


        includeOverlap <- F
            overlap        <- matrix(1,3,length(sort(unique(hind$year))))
            overlap_fut    <- array(1,c(3,length(unique(fut$GCM_scen))+1,length(sort(unique(fut$ye
      if(includeOverlap){
        overlap[3,] <- overlapIN
        overlap[3,][overlap[3,]>1]<-1 #covs$BT2to6/covs$BT0to6
      }

      # replace NA values with the mean

  # Kir's .dat file
  makeDat_hind(datIN   = hind,
               outfile = fndat_hind,
               value2use = "mn_val_scaled",
               NAVal     = "mean",
               nsppIN    = 3,
               overlapIN = overlap,
               nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
               Scaled_covlist    = unique(hind$var))

  makeDat_fut( datIN             = fut,
               hinddatIN         = hind,
               outfile           = fndat_fut,
               value2use         = "mn_val_scaled",
               NAVal             = "mean",
               nsppIN            = 3,
               last_nyrs_avg     = 10,
               overlapIN         = overlap_fut,  #(nspp,nsim+1,nyrs_fut)
               overlap_hind      = overlap,
               nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
               Scaled_covlist    = unique(hind$var))

  ### Here's a generic version that doesn't include nspp and overla[]
      # generic .dat file

  makeDat_hind(datIN             = hind,
               outfile           = fndat_hind2,
               nsppIN            = NULL,
```

```
                    overlapIN         = NULL,
                    nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
                    Scaled_covlist    = unique(hind$var))

        # generic .dat file
               makeDat_fut( datIN       = fut,
                    hinddatIN         = hind,
                    outfile           = fndat_fut2,
                    nsppIN            = NULL,
                    last_nyrs_avg     = 10,
                    overlapIN         = NULL,  #(nspp,nsim+1,nyrs_fut)
                    overlap_hind      = NULL,
                    nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
                    Scaled_covlist    = unique(hind$var))
```

Use R to make .dat file using operational hindcast

```
# 1 -- create .dat filename & path
# 2 -- rescale (Z-score) data and get variables
# 3 -- write data to hind .dat file
# 3 -- write data to fut  .dat file

# 1 -- create .dat filename & path
# -------------------------------------
suppressMessages(source("R/make.R"))


# switches
  thisYr <- format(Sys.time(), "%Y")
  today  <- format(Sys.time(), "%b %d, %Y")
  lastyr_hind <- as.numeric(thisYr)  #2021
  hind_yrs    <- 1979:lastyr_hind   # define the years of your estimation model
  fut_yrs     <- (lastyr_hind+1):2100   # define the years of your projections
  log_adj     <- 1e-4
  plotbasin   <- "SEBS"

# Define the name for the .dat file
  file.name   <- "ACLIM2_CMIP5n6_operational_short"
  fn          <- paste(file.name,"_bcs.dat",sep="")
  archive_old <- T  # Archive the older version of the .dat file?
 # normlist     <- read.csv(file=file.path(Rdata_path,"../normlist.csv"))

  outpath     <- "Data/out/ADMB_datfiles"
  if(!dir.exists(outpath)) dir.create(outpath)

# define hind and fut data files
  fndat_hind  <- file.path(outpath,paste("KKHhind_operat_",fn,sep=""))
  fndat_fut   <- file.path(outpath,paste("KKHfut_operat_",fn,sep=""))
  fndat_hind2 <- file.path(outpath,paste("hind_operat_",fn,sep=""))
  fndat_fut2  <- file.path(outpath,paste("fut_operat_",fn,sep=""))
```

```r
# create and archive .dat files
  outfile    <- fndat_fut
if(file.exists(outfile)&archive_old){
    # archive older version
        archivefl <- paste0(substr(outfile,start=1,stop=nchar(outfile)-4),
                        format(Sys.time(), "%Y%m%d_%H%M%S"),".dat")
        file.rename(outfile, archivefl)
        #file.remove(outfile)
}

file.create(outfile)
outfile  <- fndat_hind
if(file.exists(outfile)&archive_old){
    # archive older version
        archivefl <- paste0(substr(outfile,start=1,stop=nchar(outfile)-4),
                        format(Sys.time(), "%Y%m%d_%H%M%S"),".dat")
        file.rename(outfile, archivefl)
        #file.remove(outfile)
}

file.create(outfile)

# 2 -- rescale (Z-score) data and get variables
# CMIPS <- c("K20P19_CMIP6","K20P19_CMIP5")
# CMIPS <- c("K20P19_CMIP6_C")

CMIPS <- c("CMIP5_H16_operational","CMIP6_K20P19_Indices_operational")
CMIPS <- c("H16_CMIP5","CMIP6_K20P19_Indices_operational")
CMIPS <- c("CMIP5_H16_operational","CMIP6_K20P19_Indices_operational")
hinduse<- "CMIP6_K20P19_Indices_operational"
# preview possible variables
load(paste0("Data/out/",CMIPS[1],"/allEBS_means/ACLIM_annual_hind_mn.Rdata"))
varall  <- unique(ACLIM_annual_hind$var)
varall

# get each variable, convert to TS and rbind

for(c in 1:length(CMIPS)){

  # first for annual mean values:
    varlist <- c("largeZoop_integrated","fracbelow2",
                 "temp_bottom5m","temp_surface5m","pH_depthavg")
    varlist <- c("largeZoop_integrated","fracbelow2",
                 "temp_bottom5m","temp_surface5m","pH_depthavg")

  # norm_sub <- normlist%>%filter(var%in%varlist)
    typeIN <- "annual"
    load(paste0("Data/out/",hinduse,"/allEBS_means/ACLIM_",typeIN,"_hind_mn.Rdata"))
    load(paste0("Data/out/",CMIPS[c],"/allEBS_means/ACLIM_",typeIN,"_fut_mn.Rdata"))
    eval(parse(text = paste0("dhind <- ACLIM_",typeIN,"_hind")))
    eval(parse(text = paste0("dfut  <- ACLIM_",typeIN,"_fut")))

  # Z-score and recenter the data
```

```r
tmphind    <- suppressWarnings(dhind%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,year%in%hind_yrs)%>%
  dplyr::select(var,basin,year,mnDate,mn_val,
                mnVal_hind,sdVal_hind, sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled =  (mn_val-mnVal_hind )/sdVal_hind))

 tmphind    <- suppressWarnings(dhind%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,year%in%hind_yrs)%>%
    dplyr::mutate(  mnVal_hindUSE=mnVal_hind,sdVal_hindUSE=sdVal_hind)%>%
  dplyr::select(var,basin,year,mnDate,mn_val,
                 mnVal_hindUSE,sdVal_hindUSE,
                mnVal_hind,sdVal_hind,mnVal_hist,sdVal_hist,sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled =  (mn_val-mnVal_hind )/sdVal_hind,
                mn_val_BCRC = mn_val,
                mn_val_scaledBCRC = mn_val_scaled))

tmp1 <- dhind%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,year%in%hind_yrs)%>%
  dplyr::group_by(var,basin)%>%
  dplyr::summarize(sdVal_hindUSE = mean(sdVal_hind, na.rm=T),
                   mnVal_hindUSE = mean(mnVal_hind, na.rm=T))%>%ungroup()


tmpfut    <-  suppressWarnings(dfut%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,year%in%fut_yrs)%>%
  dplyr::select(var,basin,year,mnDate,mn_val, val_biascorrected,
                mnVal_hind,sdVal_hind,sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled =
                  (val_biascorrected-mnVal_hind )/sdVal_hind))
 tmpfut    <-  suppressWarnings(dfut%>%left_join(tmp1)%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,year%in%fut_yrs)%>%
  dplyr::select(var,basin,year,mnDate,mn_val, val_biascorrected,
                 mnVal_hindUSE,sdVal_hindUSE,
                mnVal_hind,sdVal_hind,mnVal_hist,sdVal_hist,sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled =
                  (val_biascorrected-mnVal_hind )/sdVal_hind))%>%
   dplyr::mutate(mn_val_BCRC = mnVal_hindUSE + ((sdVal_hindUSE/sdVal_hist)*
                                          (val_biascorrected-mnVal_hist)),
              mn_val_scaledBCRC = (mn_val_BCRC-mnVal_hindUSE )/sdVal_hindUSE)
```

```
#mnVal_hind + ((sdVal_hind/sdVal_hist)*(bcIT-mnVal_hist)),

      if(1==2){

         hind2 <- rbind(
           tmphind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
           tmphind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "b) bias corrected"),
           tmphind%>%select(year,var, mn_val_scaled, scen, GCM)%>%
             rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))
         fut2 <- rbind(
           tmpfut%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
           tmpfut%>%select(year,var, mn_val_BCRC, scen, GCM)%>%
             rename(mn_val = mn_val_BCRC)%>%mutate(type = "b) bias corrected"),
           tmpfut%>%select(year,var, mn_val_scaledBCRC, scen, GCM)%>%
             rename(mn_val = mn_val_scaledBCRC)%>%mutate(type = "c) bias corrected &  scaled"))

         pcompare <- ggplot(hind2)+
          geom_line(aes(x=year,y=mn_val ,color=scen),size=.8)+
          geom_line(data =fut2,aes(x=year,y=mn_val,color=scen, linetype = GCM ),size=.8)+
          facet_wrap(var~type,scales="free_y",nrow =length(varlist))+theme_minimal()
         pcompare

         sclr <-1.2
         png("Figs/compare_scaled2raw_2022.png",
             width =6*sclr, height =9*sclr, units = "in",res = 150)
         print( pcompare)  # hypoxic (O2<70mmol m-3) or suboxic (O2<5mmol m-3),
         dev.off()


      }

     # now for seasonal mean values:

     typeIN    <- "seasonal"
     seasonsIN <- unique(seasons$season)
     rm("dhind")
     rm("dfut")
     load(paste0("Data/out/",hinduse,"/allEBS_means/ACLIM_",typeIN,"_hind_mn.Rdata"))
     load(paste0("Data/out/",CMIPS[c],"/allEBS_means/ACLIM_",typeIN,"_fut_mn.Rdata"))
     eval(parse(text = paste0("dhind <- ACLIM_",typeIN,"_hind")))
     eval(parse(text = paste0("dfut  <- ACLIM_",typeIN,"_fut")))
      rm("ACLIM_annual_hind")
     rm("ACLIM_annual_fut")
      rm("ACLIM_seasonal_hind")
     rm("ACLIM_seasonal_fut")

      # z-score the data
      tmphind2    <- dhind%>%
        dplyr::filter(var%in%varlist,basin==plotbasin,
                      year%in%hind_yrs,
                      season%in%seasonsIN)%>%
        dplyr::mutate(var = paste0(var,"_",season))%>%
         dplyr::select(var,basin,year,mnDate,mn_val,
```

```
                mnVal_hind,sdVal_hind, sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled = (mn_val-mnVal_hind )/sdVal_hind)

  # z-score the data
tmphind2    <- dhind%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,
                year%in%hind_yrs,
                season%in%seasonsIN)%>%
  dplyr::mutate(var = paste0(var,"_",season),
                mnVal_hindUSE=mnVal_hind,sdVal_hindUSE=sdVal_hind)%>%
   dplyr::select(var,basin,year,mnDate,mn_val,
                 mnVal_hindUSE,sdVal_hindUSE,
                mnVal_hind,sdVal_hind, mnVal_hist,sdVal_hist, sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled = (mn_val-mnVal_hind )/sdVal_hind,
                mn_val_BCRC = mn_val,
                mn_val_scaledBCRC = mn_val_scaled)

tmp1 <- dhind%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,
                year%in%hind_yrs,
                season%in%seasonsIN)%>%
  dplyr::group_by(var,basin,season)%>%
  dplyr::summarize(sdVal_hindUSE = mean(sdVal_hind, na.rm=T),
                   mnVal_hindUSE = mean(mnVal_hind, na.rm=T))%>%ungroup()

tmpfut2    <- dfut%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,
                year%in%fut_yrs,season%in%seasonsIN)%>%
  dplyr::mutate(var = paste0(var,"_",season))%>%
  dplyr::select(var,basin,year,mnDate,mn_val, val_biascorrected,
                mnVal_hind,sdVal_hind, sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
                mn_val_scaled = (val_biascorrected-mnVal_hind )/
                  (sdVal_hind))

  tmpfut2    <- dfut%>%left_join(tmp1)%>%
  dplyr::filter(var%in%varlist,basin==plotbasin,
                year%in%fut_yrs,season%in%seasonsIN)%>%
  dplyr::mutate(var = paste0(var,"_",season))%>%
  dplyr::select(var,basin,year,mnDate,mn_val, val_biascorrected,
                 mnVal_hindUSE,sdVal_hindUSE,
                mnVal_hind,sdVal_hind, mnVal_hist,sdVal_hist, sim,gcmcmip,
                CMIP,GCM,scen,sim_type ,units,long_name)%>%
  dplyr::mutate(bc = "bias corrected",
                GCM_scen = paste0(GCM,"_",scen),
```

```
                    mn_val_scaled = (val_biascorrected-mnVal_hind )/
                      (sdVal_hind))%>%
        dplyr::mutate(mn_val_BCRC = mnVal_hindUSE + ((sdVal_hindUSE/sdVal_hist)*
                                                     (val_biascorrected-mnVal_hist)),
                  mn_val_scaledBCRC = (mn_val_BCRC-mnVal_hindUSE )/sdVal_hindUSE)

     if(c==1){
        hind  <- rbind(tmphind,tmphind2)
        fut   <- rbind(tmpfut,tmpfut2)
     }else{
        #hind  <- rbind(hind,tmphind,tmphind2)
        fut   <- rbind(fut,tmpfut,tmpfut2)
     }

  }

  if(1==3){

        hind2 <- rbind(
          hind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
          hind%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "b) bias corrected"),
          hind%>%select(year,var, mn_val_scaled, scen, GCM)%>%
            rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))

        fut2 <- rbind(
          fut%>%select(year,var, mn_val, scen, GCM)%>%mutate(type = "a) raw"),
          fut%>%select(year,var, val_biascorrected, scen, GCM)%>%
            rename(mn_val = val_biascorrected)%>%mutate(type = "b) bias corrected"),
          fut%>%select(year,var, mn_val_scaled, scen, GCM)%>%
            rename(mn_val = mn_val_scaled)%>%mutate(type = "c) bias corrected &  scaled"))

        pcompare2 <- ggplot(hind2[grep("largeZoop",x=hind2$var),])+
          geom_line(aes(x=year,y=mn_val ,color=scen),size=.8)+
          geom_line(data =fut2[grep("largeZoop",x=fut2$var),],
                    aes(x=year,y=mn_val,color=scen, linetype = GCM ),size=.8)+
          facet_wrap(var~type,scales="free_y",ncol =3)+theme_minimal()
        pcompare2

        sclr <-1.4
        png("Figs/compare_scaled2raw_2022_all.png",
            width =6*sclr, height =9*sclr, units = "in",res = 150)
        print( pcompare2)  # hypoxic (O2<70mmol m-3) or suboxic (O2<5mmol m-3),
        dev.off()


  }


  if(!dir.exists("Data/out/ADMB_datfiles/Figs")) dir.create("Data/out/ADMB_datfiles/Figs")

    # plot the data
  linew <-.6
     p <- ggplot(hind)+
```

```
    geom_line(aes(x=mnDate,y=mn_val,color=GCM_scen),size=linew)+
     geom_line(aes(x=mnDate,y=mnVal_hind,color=GCM_scen),size=1,linetype="dashed")+
    geom_line(data=fut,aes(x=mnDate,y=val_biascorrected,color=GCM_scen),size=linew)+
     geom_line(data=fut,aes(x=mnDate,y=mnVal_hist,color=GCM_scen),size=1,linetype="dashed")+
    facet_wrap(.~var,scales="free_y")+
    theme_minimal()
p

# plot the data
 p_scaled <- ggplot(hind)+
    geom_line(aes(x=mnDate,y=mn_val,color=GCM_scen),size=linew)+
     geom_line(aes(x=mnDate,y=mnVal_hind,color=GCM_scen),size=1,linetype="dashed")+
    geom_line(data=fut,aes(x=mnDate,y=mn_val_BCRC,color=GCM_scen),size=linew)+
     geom_line(data=fut,aes(x=mnDate,y=mnVal_hist,color=GCM_scen),size=1,linetype="dashed")+
    facet_wrap(.~var,scales="free_y")+
    theme_minimal()
p_scaled

h <- 8; w <-12; sclr <- 1.2
png(filename = "Data/out/ADMB_datfiles/Figs/scaled_covars.png",units="in",res = 250, height=h*scl
# now identify which covars are highly correlated
print(p_scaled)
dev.off()

png(filename = "Data/out/ADMB_datfiles/Figs/nonscaled_covars.png",units="in",res = 250, height=h*
# now identify which covars are highly correlated
print(p)
dev.off()


#convert wide matrix
d_wide   <- reshape2::dcast(hind%>%dplyr::filter(year!=thisYr)%>%
                              select(year, var, mn_val_scaled),
                           year~var,mean,
                           value.var = "mn_val_scaled")


# calculate correlations and display in column format
#
num_col  <- ncol(d_wide[,-1])
out_indx <- which(upper.tri(diag(num_col)))
cor_cols <- d_wide %>%
            do(melt(cor(.[,-1],
                       method="spearman",
                       use="pairwise.complete.obs"),
                   value.name="cor")[out_indx,])

corr     <- cor(na.omit(d_wide[,-1]))

long_dat <- reshape2::melt(corr,variable.name = "variable") %>%
as.data.frame()

# plot co variation between variables
```

```r
  corplot <- long_dat %>%arrange(value)%>%
              ggplot(aes(x=Var1, y=Var2, fill=value)) +
              geom_raster() +
              scale_fill_viridis_c()+
              theme_minimal()+
              theme(axis.text.x = element_text(angle = 90))
    h <- 5; w <-6
  png(filename = "Data/out/ADMB_datfiles/Figs/corplot.png",units="in",res = 250, height=h,width=w)
  # now identify which covars are highly correlated
  print(corplot)
  dev.off()

  # # remove those where cov is high (temp by season and cold pool by season)
  # subset <- long_dat$>$filter(abs(value)<0.6)

# 3 -- write data to hind .dat file
# ------------------------------------


  # CEATTLE uses a spp overlap index - you can skip this

          overlapdat <- data.frame(
              atf_OL=c(0.9391937,0.8167094,0.808367,0.5926875,0.7804481,0.5559549,
                       0.4006931,0.5881404,0.7856776,0.511565,0.6352048,0.5583476,
                       0.5792738,0.5417657,0.8212887,0.6287613,0.4536608,0.6587292,
                       0.4884194,0.8289379,0.4399257,0.5950167,0.6388434,0.6111834,
                       0.8742649,0.7868746,0.8024257,0.6227457,0.4956742,0.4347917,
                       0.4791108,0.4369006,0.5613625,0.4353015),
              south_OL=c(0.9980249,0.9390368,0.9959974,0.6130846,0.951234,0.5851891,
                         0.4934879,0.641471,0.9809618,0.5596813,0.7196964,0.6754698,
                         0.5774808,0.6041351,0.9406521,0.7949525,0.5306435,0.7977694,
                         0.5345031,0.9879945,0.5079171,0.7148121,0.8997132,0.7340859,
                         0.9962068,0.9627235,0.998043,0.8111,0.6087638,0.513057,0.5492621,
                         0.4971361,0.665453,0.5969653)
              )


        includeOverlap <- F
            overlap       <- matrix(1,3,length(sort(unique(hind$year))))
            overlap_fut   <- array(1,c(3,length(unique(fut$GCM_scen))+1,length(sort(unique(fut$ye
      if(includeOverlap){
        overlap[3,] <- overlapIN
        overlap[3,][overlap[3,]>1]<-1 #covs$BT2to6/covs$BT0to6
      }

      # replace NA values with the mean

    # Kir's .dat file
    makeDat_hind(datIN    = hind,
                 outfile = fndat_hind,
                 value2use = "mn_val_scaled",
                 NAVal     = "mean",
                 nsppIN    = 3,
```

```
                   overlapIN = overlap,
                   nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
                   Scaled_covlist    = unique(hind$var))

     makeDat_fut( datIN              = fut,
                   hinddatIN          = hind,
                   outfile            = fndat_fut,
                   value2use          = "mn_val_scaled",
                   NAVal              = "mean",
                   nsppIN             = 3,
                   last_nyrs_avg      = 10,
                   overlapIN          = overlap_fut,  #(nspp,nsim+1,nyrs_fut)
                   overlap_hind       = overlap,
                   nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
                   Scaled_covlist    = unique(hind$var))

     ### Here's a generic version that doesn't include nspp and overla[]
         # generic .dat file

     makeDat_hind(datIN              = hind,
                   outfile            = fndat_hind2,
                   nsppIN             = NULL,
                   overlapIN          = NULL,
                   nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
                   Scaled_covlist    = unique(hind$var))

     # generic .dat file
             makeDat_fut( datIN       = fut,
                   hinddatIN          = hind,
                   outfile            = fndat_fut2,
                   nsppIN             = NULL,
                   last_nyrs_avg      = 10,
                   overlapIN          = NULL,  #(nspp,nsim+1,nyrs_fut)
                   overlap_hind       = NULL,
                   nonScaled_covlist = c("temp_bottom5m","temp_surface5m"  ),
                   Scaled_covlist    = unique(hind$var))
     save(fut, file=paste0("Data/out/ADMB_datfiles/Figs/fut_",format(Sys.time(), "%Y%m%d_%H%M"),".Rda
     save(hind, file=paste0("Data/out/ADMB_datfiles/Figs/hind_",format(Sys.time(), "%Y%m%d_%H%M"),".R
     save(fut, file="Data/out/ADMB_datfiles/Figs/fut.Rdata", overwrite=T)
     save(hind, file="Data/out/ADMB_datfiles/Figs/hind.Rdata", overwrite=T)
```

APPENDIX A: Create & bias correct ACLIM2 indices

The following code shows how the ACLIM2 indices and bias correction was done. You do not need to re-run this (it is included so you can if you want to). To explore the indices skep to the next section.

```
# --------------------------------------
# SETUP WORKSPACE
# rm(list=ls()); setwd("D:/GitHub_cloud/ACLIM2")
# loads packages, data, setup, etc.
tmstp       <- "2022_10_17"
suppressMessages(source("R/make.R"))
```

```r
tmstp       <- "2022_10_17"
Rdata_path  <- paste0("../../romsnpz/",tmstp,"_Rdata")
main        <- getwd()  #"~/GitHub_new/ACLIM2"
tmstamp1    <- format(Sys.time(), "%Y%m%d")
# tmstamp1  <- "20220428"


update_hind  <- TRUE   # set to true to update hind and hindS; needed annually
update_proj  <- TRUE   # set to true to update fut; not needed
update_hist  <- TRUE   # set to true to update fut; not needed

# the reference years for bias correcting in R/setup.R
ref_years

# the year to z-score scale / delta in R/setup.R
deltayrs
# the year to z-score scale / delta in R/setup.R
deltayrs
data_path

#load(file.path(Rdata_path,"../weekly_vars_C.Rdata"))
load(file.path(Rdata_path,"weekly_vars.Rdata"))
#load(file.path(Rdata_path,"../srvy_vars_C.Rdata"))
load(file.path(Rdata_path,"srvy_vars.Rdata"))

load(file.path(Rdata_path,"l3srvy_varlist.Rdata"))
load(file.path(Rdata_path,"l3wk_varlist.Rdata"))
load(file.path(Rdata_path,"l3srvy_varlist_H16.Rdata"))
load(file.path(Rdata_path,"l3wk_varlist_H16.Rdata"))

load(file.path(Rdata_path,"l2_vars.Rdata"))


vl1   <- l3srvy_varlist #srvy_vars[!srvy_vars%in%rm_var_list]
vl2   <- l3wk_varlist# weekly_vars[!weekly_vars%in%rm_wk_list]

# add in largeZoop (gets generated in make_indices_region_new.R)
vl <- c(unique(c(vl1,vl2)),"largeZoop_integrated")

# Identify which variables would be normally
# distributed (i.e., can have negative values)
 normvl <- c( vl[grep("pH",vl)],
              vl[grep("temp",vl)],
              vl[grep("vNorth",vl)],
              vl[grep("uEast",vl)])

normlist <- data.frame(var = vl, lognorm = TRUE)
normlist$lognorm[normlist$var%in%normvl] <- FALSE

save(normlist,file      = file.path(Rdata_path,"normlist.Rdata"))
write.csv(normlist,file = file.path(Rdata_path,"normlist.csv"))
save(weekly_vars,file   = "Data/out/weekly_vars.Rdata")
save(srvy_vars,file     = "Data/out/srvy_vars.Rdata")
```

```
write.csv(normlist,file = file.path("Data/out/","normlist.csv"))


# generate indices and bias corrected projections
# This takes approx 30 mins each

gcmcmipL <- c("B10K-K20P19_CMIP6_miroc",
              "B10K-K20P19_CMIP6_gfdl",
              "B10K-K20P19_CMIP6_cesm")

 CMIP6_K20P19_Indices <- suppressMessages(
                   makeACLIM2_Indices(
                   BC_target = "mn_val",
                   hind_sim   =  "B10K-K20P19_CORECFS",
                   histLIST   = paste0(gcmcmipL,"_historical"),
                   gcmcmipLIST = gcmcmipL,
                   scenIN     = c("ssp126","ssp585"),
                   Rdata_pathIN = file.path(Rdata_path,"roms_for_public"),
                   regnm     = "ACLIMregion",
                   srvynm     = "ACLIMsurveyrep",
                   normlist_IN = normlist,
                   usehist     = TRUE,
                   sim_listIN = sim_list[-grep("historical",sim_list)]))

 if("CMIP6_K20P19_Indices"%in%ls()){
  saved <- FALSE
  saved <- save_indices(flIN = CMIP6_K20P19_Indices,
              subfl = "allEBS_means",
              post_txt = "_mn",
              CMIP_fdlr ="K20P19_CMIP6")
  fl <- "Data/out/CMIP6_K20P19_Indices_list.Rdata"

  if(file.exists(fl)) file.remove(fl)
  save(CMIP6_K20P19_Indices, file = fl)
   if(saved){
    rm(CMIP6_K20P19_Indices)}else{
      stop("Indices not saved!")
    }
  gc()
 }


if(10==1){
  # Depreciated - now using K20P19 hindcast instead
 CMIP6_K20_Indices <- suppressMessages(
                   makeACLIM2_Indices(
                   BC_target = "mn_val",
                   hind_sim   =  "B10K-K20_CORECFS",
                   scenIN     = c("ssp126","ssp585"),
                   histLIST   = paste0(gcmcmipL,"_historical"),
                   gcmcmipLIST = gcmcmipL,
                   Rdata_pathIN = file.path(Rdata_path,"roms_for_public"),
                   normlist_IN = normlist,
```

```
                        sim_listIN = sim_list[-grep("historical",sim_list)]))

  if("CMIP6_K20_Indices"%in%ls()){
   saved <- FALSE
   saved <- save_indices(flIN = CMIP6_K20_Indices,
                   subfl = "allEBS_means",
                   post_txt = "_mn",
                   CMIP_fdlr ="K20P19_CMIP6")
   fl <- "Data/out/CMIP6_K20_Indices_list.Rdata"

   if(file.exists(fl)) file.remove(fl)
   save(CMIP6_K20_Indices, file = fl)
   if(saved){
     rm(CMIP6_K20_Indices)}else{
       stop("Indices not saved!")
     }
   gc()
  }
 }

# CMIP5 K20P19
gcmcmipL2 <- c("B10K-K20P19_CMIP5_MIROC",
               "B10K-K20P19_CMIP5_GFDL",
               "B10K-K20P19_CMIP5_CESM")

CMIP5_K20P19_Indices <- suppressMessages(
                    makeACLIM2_Indices(
                    BC_target = "mn_val",
                    hind_sim  =  "B10K-K20P19_CORECFS",
                    #histLIST  = rep("B10K-K20P19_CORECFS",3),
                    histLIST  = gcmcmipL2,
                    usehist   = FALSE,
                    gcmcmipLIST = gcmcmipL2,
                    normlist_IN = normlist,
                    scenIN    = c("rcp45","rcp85"),
                    Rdata_pathIN = file.path(Rdata_path,"roms_for_public"),
                    regnm     = "ACLIMregion",
                    srvynm    = "ACLIMsurveyrep",
                    sim_listIN = sim_list[-grep("historical",sim_list)]))

if("CMIP5_K20P19_Indices"%in%ls()){
  saved <- FALSE
  saved <- save_indices(flIN = CMIP5_K20P19_Indices,
                 subfl = "allEBS_means",
                 post_txt = "_mn",
                 CMIP_fdlr ="K20P19_CMIP5")

    fl <- "Data/out/CMIP5_K20P19_Indices_list.Rdata"
    if(file.exists(fl)) file.remove(fl)
    save(CMIP5_K20P19_Indices, file = fl)

    if(saved){
    rm(CMIP5_K20P19_Indices)}else{
```

```r
      stop("Indices not saved!")
    }
    gc()
}

# CMIP5 H16
gcmcmipL2 <- c("B10K-H16_CMIP5_MIROC",
               "B10K-H16_CMIP5_GFDL",
               "B10K-H16_CMIP5_CESM")
CMIP5_H16_Indices <- suppressMessages(
                    makeACLIM2_Indices(
                    BC_target = "mn_val",
                    scenIN    = c("rcp45","rcp85"),
                    hind_sim  =  "B10K-H16_CORECFS",
                    histLIST = gcmcmipL2,
                      usehist   = FALSE,
                    #histLIST  = rep("B10K-H16_CORECFS",3),
                    gcmcmipLIST = gcmcmipL2,
                    Rdata_pathIN = file.path(Rdata_path,"roms_for_public"),
                    sim_listIN = sim_list[-grep("historical",sim_list)]))
if("CMIP5_H16_Indices"%in%ls()){
  saved <- FALSE
  saved <- save_indices(flIN = CMIP5_H16_Indices,
               subfl = "allEBS_means",
               post_txt = "_mn",
               CMIP_fdlr ="H16_CMIP5")

  fl <- "Data/out/CMIP5_H16_Indices_list.Rdata"
  if(file.exists(fl)) file.remove(fl)
  save(CMIP5_H16_Indices, file = fl)
  if(saved){
    rm(CMIP5_H16_Indices)}else{
      stop("Indices not saved!")
    }
  gc()
}
if(1==10){
  save(CMIP6_Indices, file = "Data/out/CMIP6_Indices_List.Rdata")
  save(CMIP5_K20_Indices, file = "Data/out/CMIP5_K20_Indices_List.Rdata")
  save(CMIP5_H16_Indices, file = "Data/out/CMIP5_H16_Indices_List.Rdata")
}
```

misc

$$B0^k_{input} = \bar{B0}^k_{(2004:2014)} \left(\frac{B0^a_{2015}}{\bar{B0}^a_{(2004:2014)}} \right)$$

Where B0kinput is the unfished biomass used for setting inputs of (e.g., B0ktarget = 0.4B0kinput) and is determined by re-scaling the spawning stock biomass from the status quo assessment in 2015 (B0a2015) to the average model spawning stock biomass for your model between 2004-2014 (i.e., B0k) using the average unfished biomass from the stock assessment model during the same period (B0a).

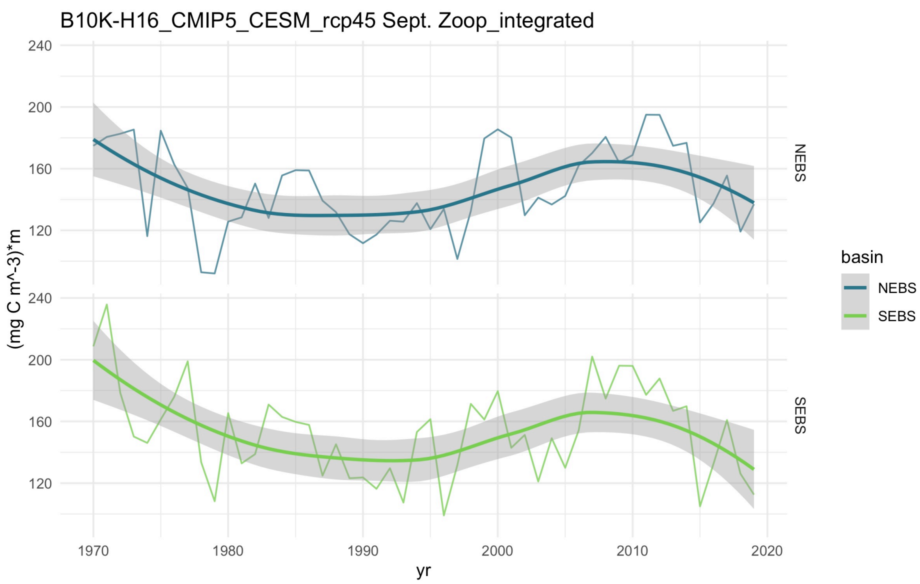


Figure 3: September large zooplankton integrated concentration