Creating bias-corrected time series of sea temperature

Kelly Ortega-Cisneros, Denisse Fierro-Arcos

2024-10-24

Background

In this notebook, we apply the bias correction workflow described in Ortega-Cisneros et al. (2024). We use the Southern Benguela ecosystem model as an example.

All data used here, with the exception of the regional boundaries, can be downloaded from the FishMIP Input Explorer app. The `parquet` files can be downloaded from the "Model outputs against observations" tab, while the `zarr` file comes from the "GFDL model outputs" tab.

Note that the regional boundaries and `parquet` files are included under the `data` folder of this repository. However, you will need to download the `zarr` files for the **Southern Benguela** region from the app before running this notebook. The data is downloaded as a compressed folder (`zip`) containing four files. Remember to unzip the folder before running this script. This folder was not included in this repository due to its large size.

Loading relevant libraries

Loading relevant files

We use the southern Benguela model as an example, but you can replace it with data for any regional model available via the FishMIP Shiny app. You need to ensure that files are downloaded and stored in the `data` folder of this repository prior to running this script.

```
# Temperature data from GFDL
sb_gfdl <- file.path(
  "../data",
  "gfdl-mom6-cobalt2_obsclim_thetao_15arcmin_southern-benguela_mthly_clim_mean_1981_2010.parquet") |>
  read_parquet()

# Temperature data from WOA
sb_woa <- file.path(
  "../data",
  "regridded_woa_southern-benguela_month_clim_mean_temp_1981-2010.parquet") |>
  read_parquet()

# Area of grid cells
sb_areacello <- file.path(
  "../data",
  "gfdl-mom6-cobalt2_areacello_15arcmin_southern-benguela_fixed.parquet") |>
```

```
  read_parquet() |>
  #Selecting relevant columns
  select(lat:vals) |>
  rename(area = vals)

# Depth of grid cells
sb_depth <- file.path(
  "../data",
  "gfdl-mom6-cobalt2_obsclim_deptho_15arcmin_southern-benguela_fixed.parquet") |>
  read_parquet()

# Regional model boundaries
benguela <- st_read("../data/model_regions_v3_geo3.shp")
```

```
## Reading layer 'model_regions_v3_geo3' from data source
##   '/rd/gem/private/users/ldfierro/Bias_corrected_timeseries/data/model_regions_v3_geo3.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 18 features and 5 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 14.33697 ymin: -37.11961 xmax: 27.83333 ymax: -28.48808
## Geodetic CRS:  WGS 84
```

Finally, we will define the location of the `zarr` folder. Remember, you must first download this folder from the FishMIP Input Explorer app and uncompressed it in the `data` folder included in this repository before you can run the chunk below.

```
sb_temp_zarr_path <- file.path(
  "../data", paste0("gfdl-mom6-cobalt2_obsclim_thetao_15arcmin_southern-",
                    "benguela_monthly_1961_2010.zarr"))
```

Exploring parquet file contents

You can use the `nanoparquet` package to explore the contents of the parquet files as shown below. This is useful to understand the structure of the data and the column types. Alternatively, you can use the `str()` function from base R or `glimpse` from `dplyr` to get a summary of the data.

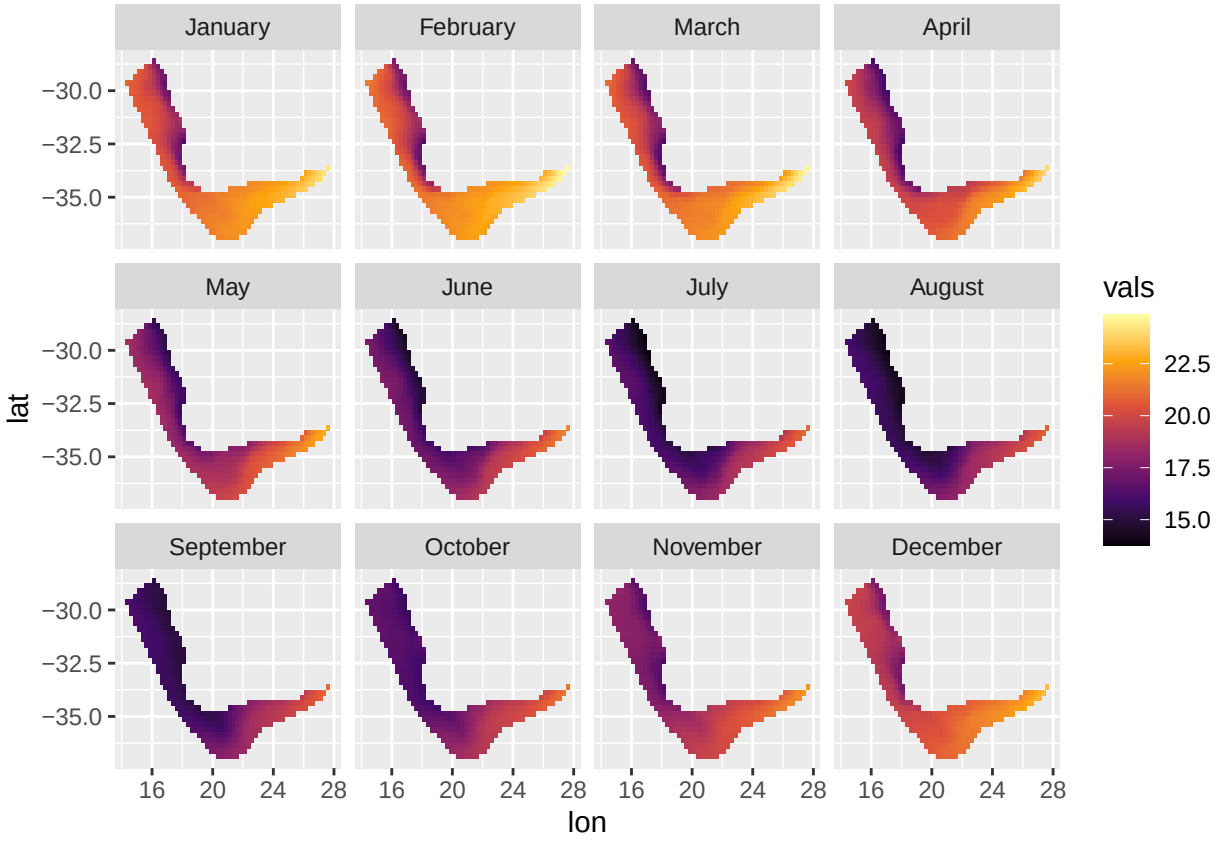
```
parquet_column_types(sb_gfdl)
```

```
## # A data frame: 9 x 6
##   file_name name          type       r_type    repetition_type logical_type
##   <chr>     <chr>         <chr>      <chr>     <chr>           <I<list>>
## 1 <NA>      lat           DOUBLE     double    REQUIRED        <NULL>
## 2 <NA>      lon           DOUBLE     double    REQUIRED        <NULL>
## 3 <NA>      month         BYTE_ARRAY character REQUIRED        <STRING>
## 4 <NA>      depth         DOUBLE     double    REQUIRED        <NULL>
## 5 <NA>      vals          DOUBLE     double    REQUIRED        <NULL>
## 6 <NA>      GFDL_variable BYTE_ARRAY character OPTIONAL        <STRING>
## 7 <NA>      long_name     BYTE_ARRAY character OPTIONAL        <STRING>
## 8 <NA>      standard_name BYTE_ARRAY character OPTIONAL        <STRING>
## 9 <NA>      units         BYTE_ARRAY character OPTIONAL        <STRING>
```

Visualising temperature data

Plotting temperature data for the upper most layer from GFDL monthly climatology file.

```
sb_gfdl |>
  filter(depth == min(depth)) |>
  #Converting months to factor and ordering them prior to plotting
  mutate(month = factor(month, levels = month.name, ordered = T)) |>
  ggplot(aes(lon, lat, fill = vals)) +
  geom_tile()+
  scale_fill_viridis_c(option = "inferno")+
  facet_wrap(~month)
```



The plot shows the monthly climatology (1981-2010) of sea surface temperature for the southern Benguela region from the GFDL model. Warmer colours indicate higher temperatures. The warmest months are January to March.

Processing GFDL monthly climatologies for temperature

Currently, the depth file includes the depth from the surface to the depth bin, so we will calculate the height of the depth bins. We do this because we need to calculate the volume of the grid cell to use as a weight when calculating the weighted mean for temperature.

```
depth_bins <- sb_gfdl |>
  filter(depth <= 500) |>
  distinct(depth) |>
  #Calculating the difference between each depth bin to get the height of the
  #grid cell
  mutate(depth_height = depth-lag(depth, default = 0))

#Checking result
head(depth_bins)
```

```
## # A data frame: 6 x 2
##   depth depth_height
## * <dbl>        <dbl>
## 1   2.5          2.5
## 2  10            7.5
## 3  20           10
## 4  32.5         12.5
## 5  51.2         18.8
## 6  75           23.8
```

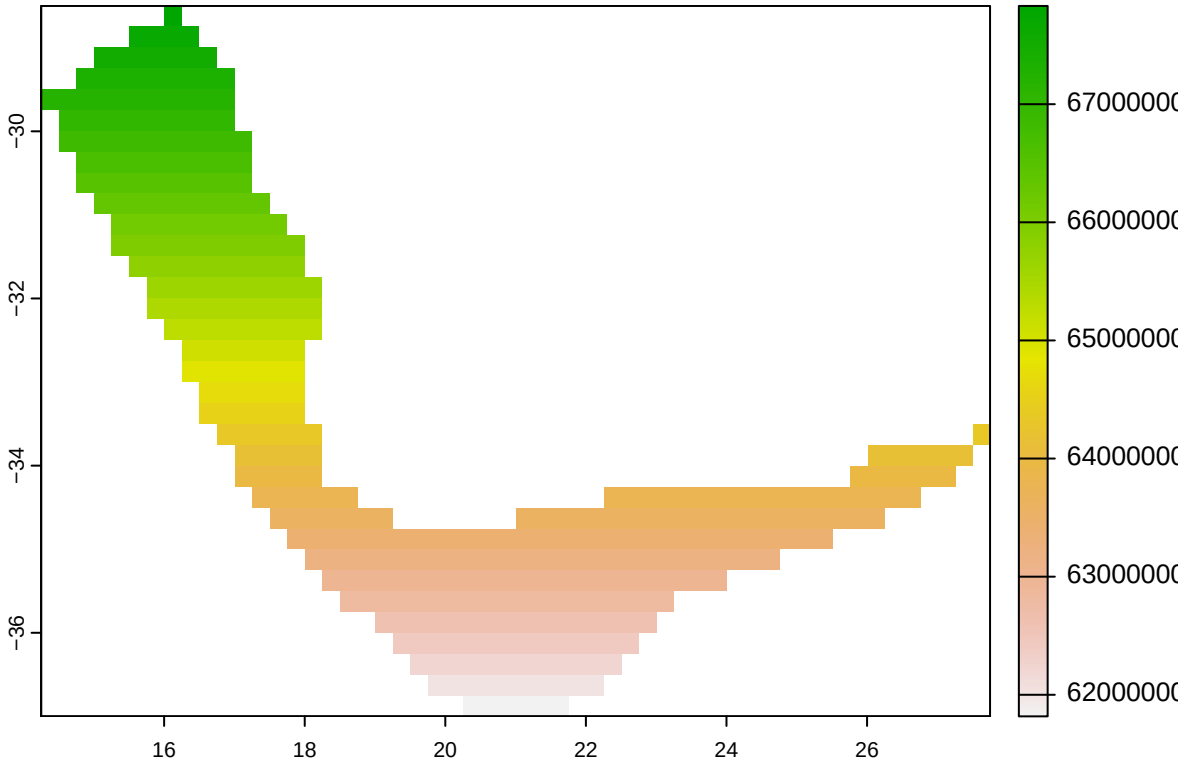
The next step is calculating the climatologies between 1981 and 2010.

```
# Calculating GFDL climatologies from 1981-2010
sb_gfdl_clims <- sb_gfdl |>
  #Keeping relevant columns only
  select(lat:vals) |>
  rename(temperature = vals) |>
  #the maximum depth of the southern Benguela Atlantis model is 500 m
  filter(depth <= 500) |>
  drop_na(temperature) |>
  #Defining depth categories for the southern Benguela Atlantis model
  mutate(layer = case_when(depth < 51 ~ 1, depth > 51 & depth < 101 ~ 2,
                           depth > 100 & depth < 301 ~ 3, depth > 300 ~ 4)) |>
  #Adding height of grid cell
  left_join(depth_bins, by = "depth") |>
  #Adding area of grid cell
  left_join(sb_areacello, by = c("lat", "lon")) |>
  #Calculate volume of grid cell
  mutate(volume = depth_height*area)

#Checking result
head(sb_gfdl_clims)
```

```
## # A data frame: 6 x 9
##     lat   lon month   depth temperature layer depth_height       area     volume
## * <dbl> <dbl> <chr>   <dbl>       <dbl> <dbl>        <dbl>      <dbl>      <dbl>
## 1 -28.6  16.1 January   2.5        17.4     1          2.5 678316774.     1.70e9
## 2 -28.9  15.6 January   2.5        20.0     1          2.5 676694946.     1.69e9
## 3 -28.9  15.9 January   2.5        19.2     1          2.5 676694946.     1.69e9
## 4 -28.9  16.1 January   2.5        18.2     1          2.5 676694946.     1.69e9
## 5 -28.9  16.4 January   2.5        17.6     1          2.5 676694946.     1.69e9
## 6 -29.1  15.1 January   2.5        20.6     1          2.5 675060234.     1.69e9
```

Next, we will turn the above data frame into a grid, which will then be used to create a mask of the different subpolygons of the Southern Benguela region.

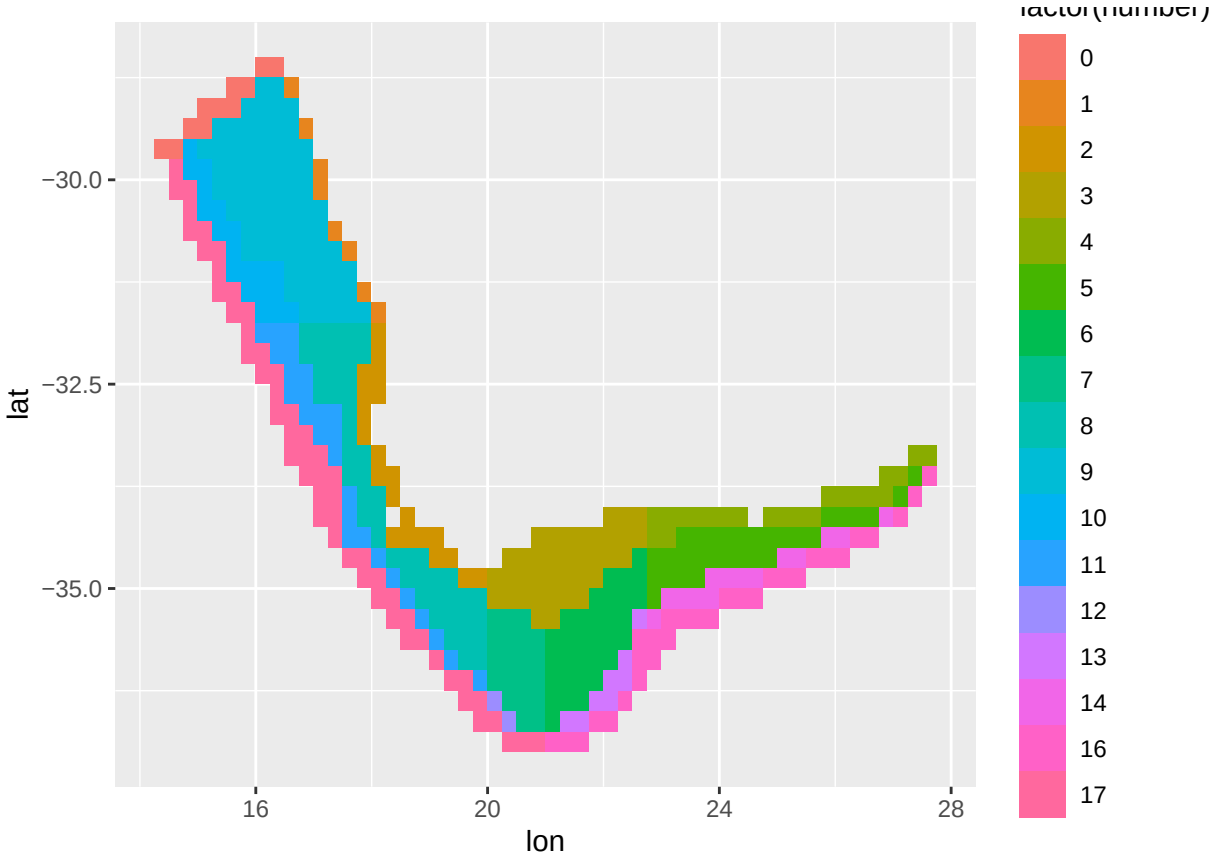
```
#Create raster from data frame
ras <-  sb_gfdl_clims |>
  ungroup() |>
  #A single date and layer is needed as a sample
  filter(layer == min(layer) & month == min(month)) |>
  #Only these three columns are needed to create a raster
  select(lon, lat, area) |>
  #Create raster
  rast(type = "xyz", crs = "epsg:4326")

#Check result
plot(ras)
```



Now we have a grid of our model, which we can use to rasterise the shapefile of Southern Benguela. We are interested in getting information about the subregions contained within Southern Benguela This information is contained in the `number` column of the shapefile.

```
#Create raster mask
cb_mask <- rasterize(benguela, ras, field = "number", fun = max,
                     background = NA) |>
  #Transform to data frame
  as.data.frame(xy = T) |>
  rename(lon = x, lat = y)
```

```
#Check results
cb_mask |>
  ggplot(aes(lon, lat, fill = factor(number)))+
  geom_raster()
```

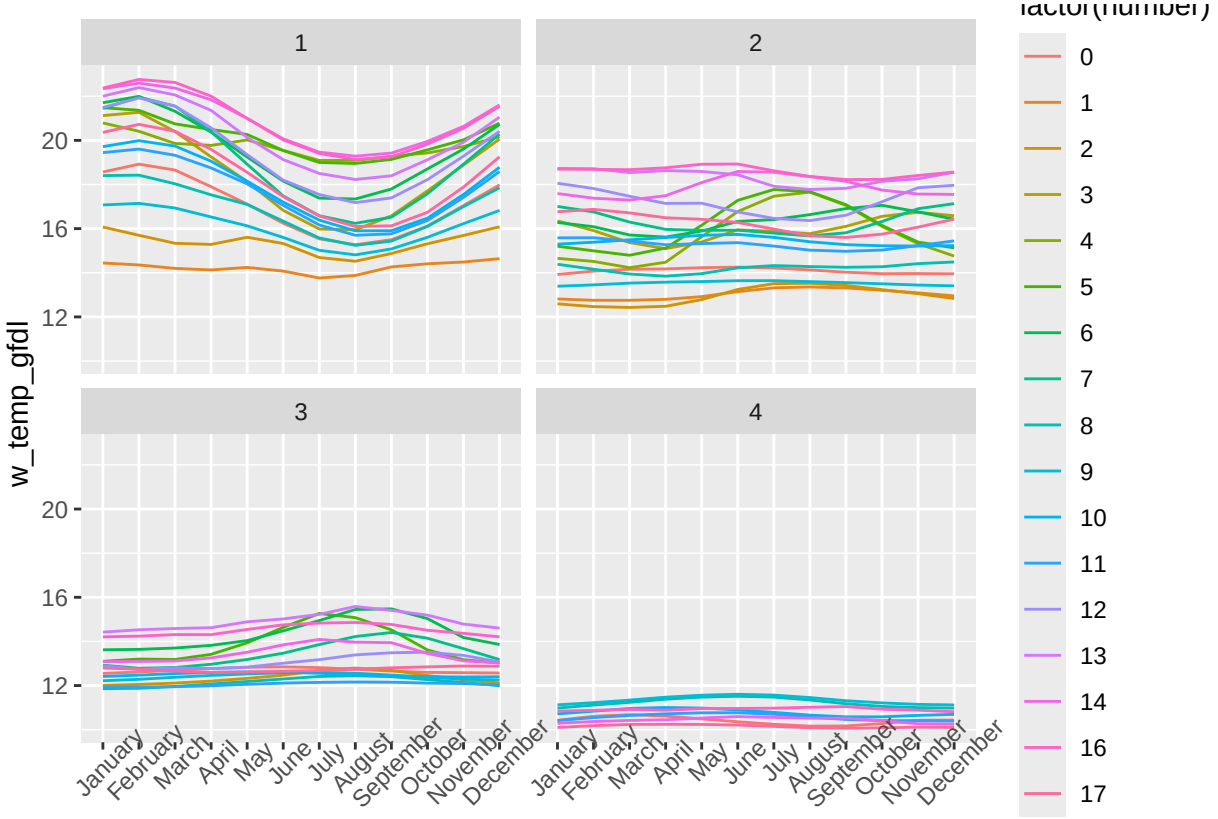


The result is a gridded product that contains the 18 different subregions contained within Southern Benguela (numbered from 0 to 17). We can now add this information to our monthly climatology data.

```
#Add information about subregion
sb_gfdl_clims <- sb_gfdl_clims |>
  left_join(cb_mask, by = c("lat", "lon"))

#Calculate the weighted mean for temperature per month, per depth group and subregion
sb_gfdl_summaries <- sb_gfdl_clims |>
  group_by(layer, month, number)|>
  #Using volume as weights in mean calculation
  summarise(w_temp_gfdl = weighted.mean(temperature, volume, na.rm = TRUE)) |>
  #Transforming month column to factor
  mutate(month = factor(month, levels = month.name)) |>
  arrange(as.numeric(number), month)
```

```
## `summarise()` has grouped output by 'layer', 'month'. You can override using
## the `.groups` argument.
```

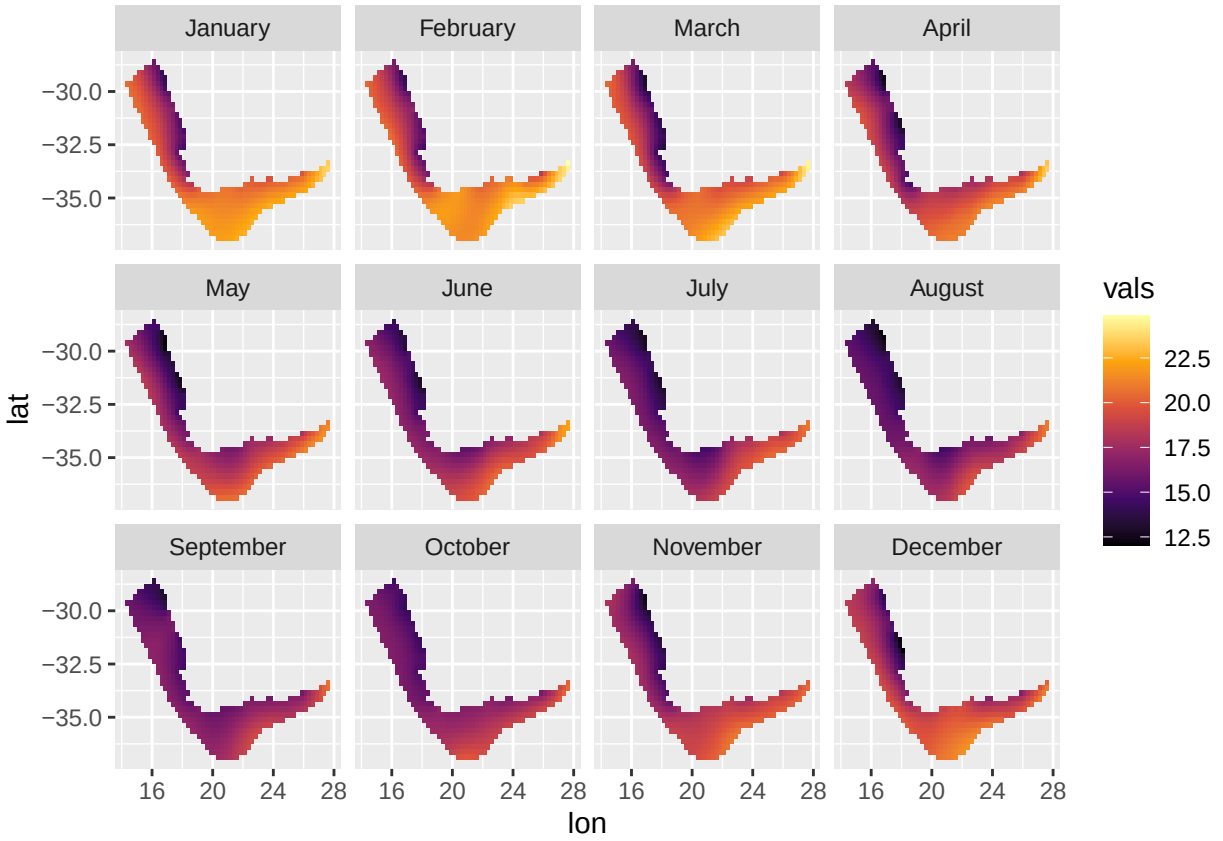
```
#Plotting results
sb_gfdl_summaries |>
  ggplot(aes(month, w_temp_gfdl, color = factor(number)))+
  geom_line(aes(group = number))+
  #Subplot by depth layer
  facet_wrap(~layer)+
  theme(axis.text.x = element_text(angle = 45),
        axis.title.x = element_blank())
```



Processing WOA monthly climatologies for temperature

We will apply now follow the same process with the WOA data.

```
#Plotting temperature data for the upper most layer from WOA climatology
sb_woa |>
  filter(depth == min(depth)) |>
  #Converting months to factor and ordering them prior to plotting
  mutate(month = factor(month, levels = month.name, ordered = T)) |>
  ggplot(aes(lon, lat, fill = vals)) +
  geom_tile()+
  scale_fill_viridis_c(option = "inferno")+
  facet_wrap(~month)
```



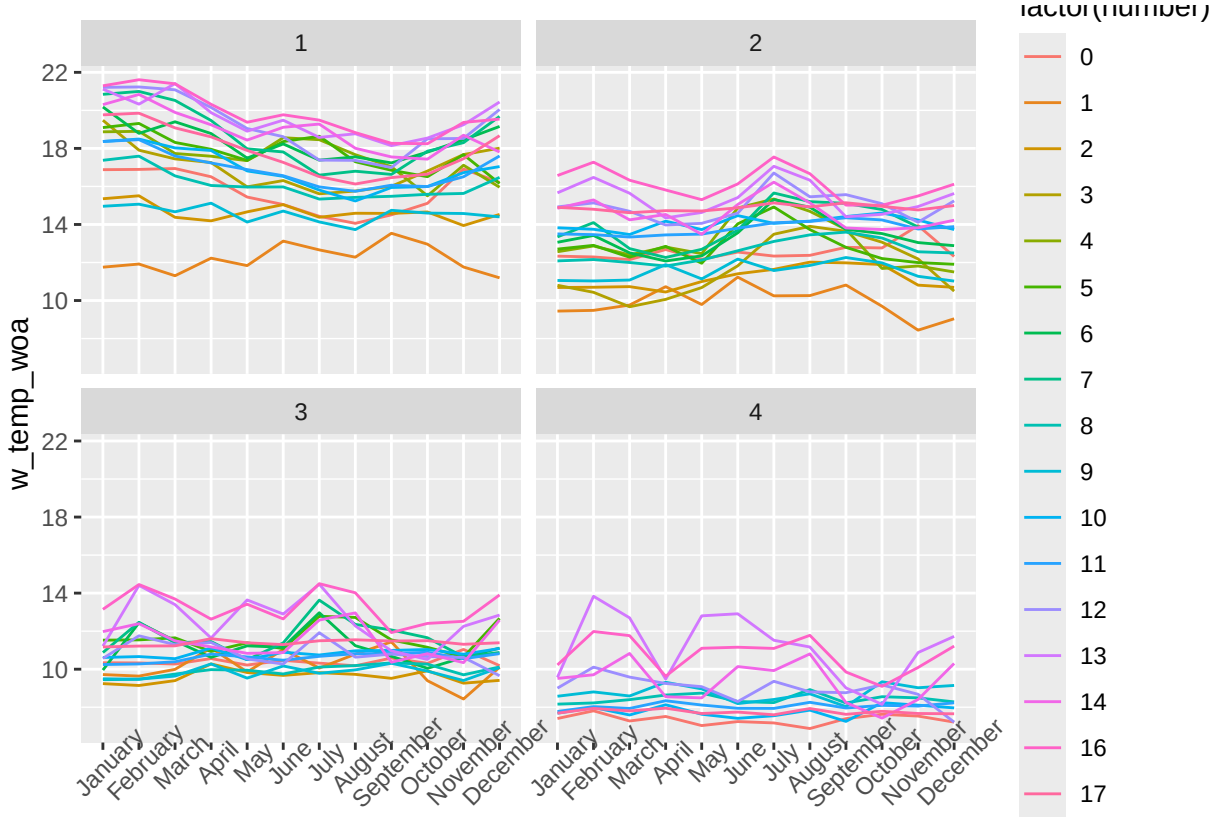
Broad scale patterns are similar to the GFDL monthly climatologies. In both cases, the warmest months are between January and April.

```
# Calculating WOA climatologies from 1981-2010 for the 0-50 m depth layer or realm of this model
sb_woa_summaries <- sb_woa |>
  #Keeping relevant columns only
  select(lat:vals) |>
  rename(temperature = vals) |>
  #the maximum depth of the southern Benguela Atlantis model is 500 m
  filter(depth <= 500) |>
  #Defining depth categories for the southern Benguela Atlantis model
  mutate(layer = case_when(depth < 51 ~ 1, depth > 51 & depth < 101 ~ 2,
                           depth > 100 & depth < 301 ~ 3, depth > 300 ~ 4)) |>
  #Adding height of grid cell
  left_join(depth_bins, by = "depth") |>
  #Adding area of grid cell
  left_join(sb_areacello, by = c("lat", "lon")) |>
  #Adding information about Southern Benguela subregions
  left_join(cb_mask, by = c("lat", "lon")) |>
  #Calculate volume of grid cell
  mutate(volume = depth_height*area) |>
  #Excludes grid cells that are not present in GFDL
  drop_na(volume, temperature) |>
  group_by(layer, month, number) |>
  #Using area as weights in mean calculation
  summarise(w_temp_woa = weighted.mean(temperature, volume, na.rm = TRUE)) |>
```

```
  #Transforming month column to factor
  mutate(month = factor(month, levels = month.name)) |>
  arrange(number, month)
```

```
## `summarise()` has grouped output by 'layer', 'month'. You can override using
## the `.groups` argument.
```

```
#Plotting results
sb_woa_summaries |>
  ggplot(aes(month, w_temp_woa, color = factor(number)))+
  geom_line(aes(group = number))+
  #Subplot by depth layer
  facet_wrap(~layer)+
  theme(axis.text.x = element_text(angle = 45),
        axis.title.x = element_blank())
```



The WOA data reaches a higher temperature towards the beginning of the year for the first two depth layers than GFDL. The opposite is true for the two deepest layers.

Exploring monthly temperature data from GFDL

This information is contained within the `zarr` folder, which you need to download and uncompress to the `data` folder of this repository **before** you can continue running this script.

```
zarr_overview(sb_temp_zarr_path)
```

```
## Type: Group of Arrays
## Path: /rd/gem/private/users/ldfierro/Bias_corrected_timeseries/data/gfdl-mom6-cobalt2_obsclim_thetao
## Arrays:
## ---
##   Path: depth_bin_m
##   Shape: 35
##   Chunk Shape: 35
##   No. of Chunks: 1 (1)
##   Data Type: float64
##   Endianness: little
##   Compressor: blosc
## ---
##   Path: lat
##   Shape: 36
##   Chunk Shape: 36
##   No. of Chunks: 1 (1)
##   Data Type: float64
##   Endianness: little
##   Compressor: blosc
## ---
##   Path: lon
##   Shape: 55
##   Chunk Shape: 55
##   No. of Chunks: 1 (1)
##   Data Type: float64
##   Endianness: little
##   Compressor: blosc
## ---
##   Path: thetao
##   Shape: 600 x 35 x 36 x 55
##   Chunk Shape: 600 x 5 x 36 x 55
##   No. of Chunks: 7 (1 x 7 x 1 x 1)
##   Data Type: float32
##   Endianness: little
##   Compressor: blosc
## ---
##   Path: time
##   Shape: 600
##   Chunk Shape: 600
##   No. of Chunks: 1 (1)
##   Data Type: int64
##   Endianness: little
##   Compressor: blosc
```

This is telling us that the `zarr` folder contains the following variables: `depth_bin_m` (depth in meters), `lat`, `lon` (coordinates), `thetao` (temperature of the water column), and `time`.

We can now load the data as shown below.

Loading data

To load the data, we will need to add the variable name to the end of the `zarr` path we defined at the beginning of this notebook.

```
temp_data <- read_zarr_array(file.path(sb_temp_zarr_path, "thetao"))
depth <- read_zarr_array(file.path(sb_temp_zarr_path, "depth_bin_m"))
lat <- read_zarr_array(file.path(sb_temp_zarr_path, "lat"))
lon <- read_zarr_array(file.path(sb_temp_zarr_path, "lon"))
time <- read_zarr_array(file.path(sb_temp_zarr_path, "time"))
```

Checking dimensions of temperature data

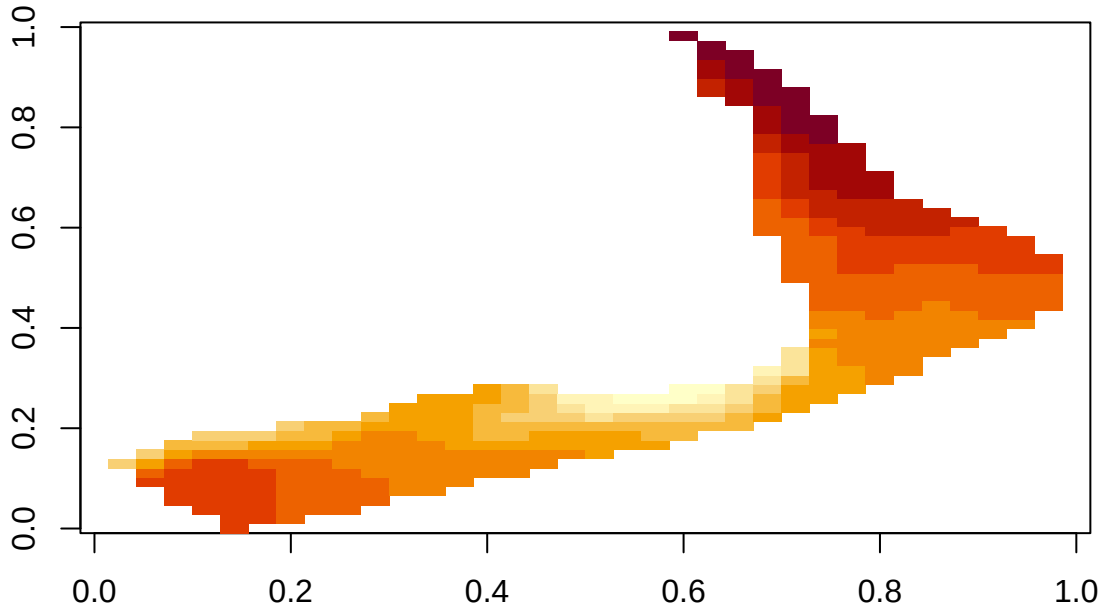
```
dim(temp_data)
```

```
## [1] 600  35  36  55
```

The temperature data has 600 timesteps (`time`), 35 depth levels (`depth_bin_m`), 36 grid cells along latitude (`lat`) and 55 grid cells along longitude (`lon`).

We can now plot the first timestep and depth bin.

```
image(temp_data[1,1,,])
```

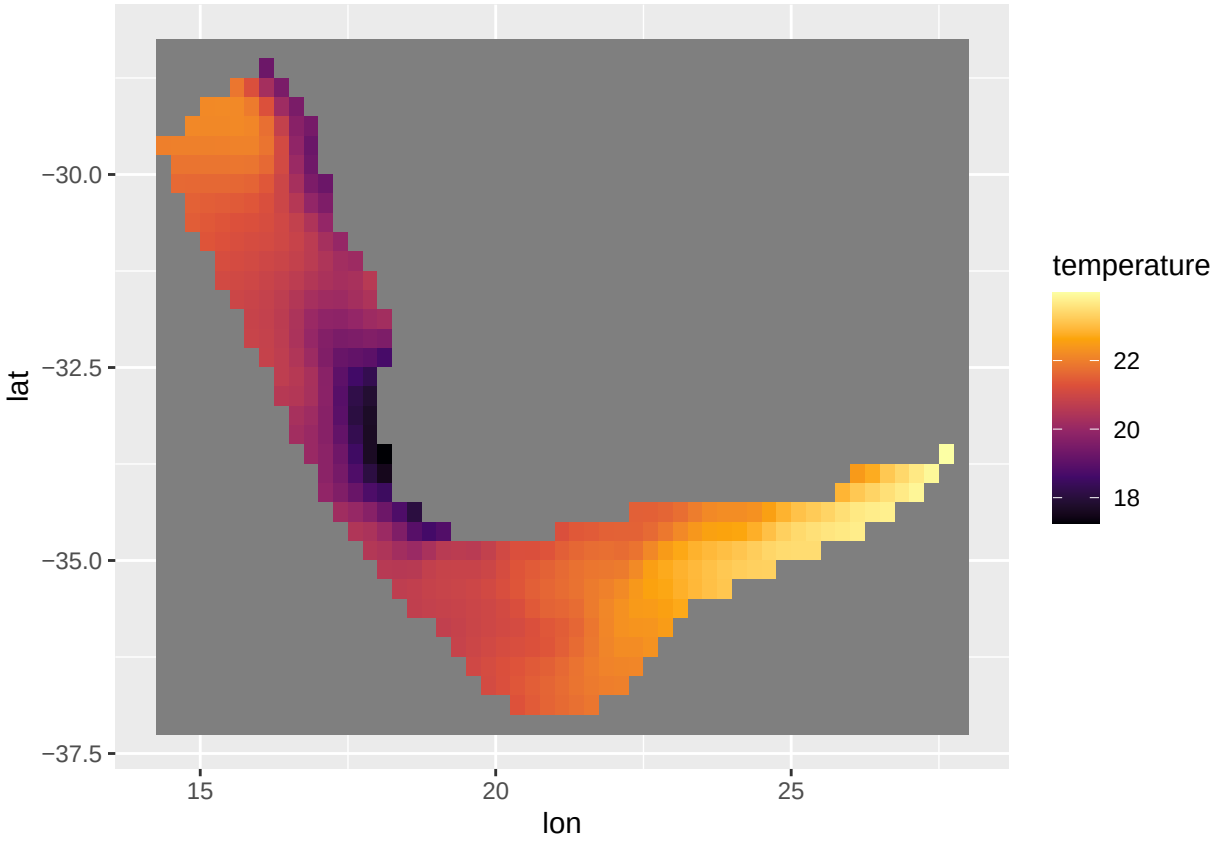


This map does not look right. The shape is not correct and the axes do not contain information about coordinates. We need to process this data a little more.

Creating a data frame from all components

It is important to check dimensions of dataset. Note that dimensions need to match the order of dimensions in the temperature data. That is 600 timesteps (`time`), 35 depth levels (`depth_bin_m`), 36 grid cells along latitude (`lat`) and 55 grid cells along longitude (`lon`).

```
#We will create a grid with time, depth and coordinates
temp_sb <- cbind(expand.grid(time, depth, lat, lon),
                 #finally add temperature
                 val = as.vector(temp_data))

#Rename the columns to reflect their contents
names(temp_sb) <- c("time", "depth", "lat", "lon", "temperature")

#Plotting result for first time step and depth
temp_sb |>
  filter(time == min(time) & depth == min(depth)) |>
  ggplot(aes(lon, lat, fill = temperature))+
  geom_tile()+
  scale_fill_viridis_c(option = "inferno")
```



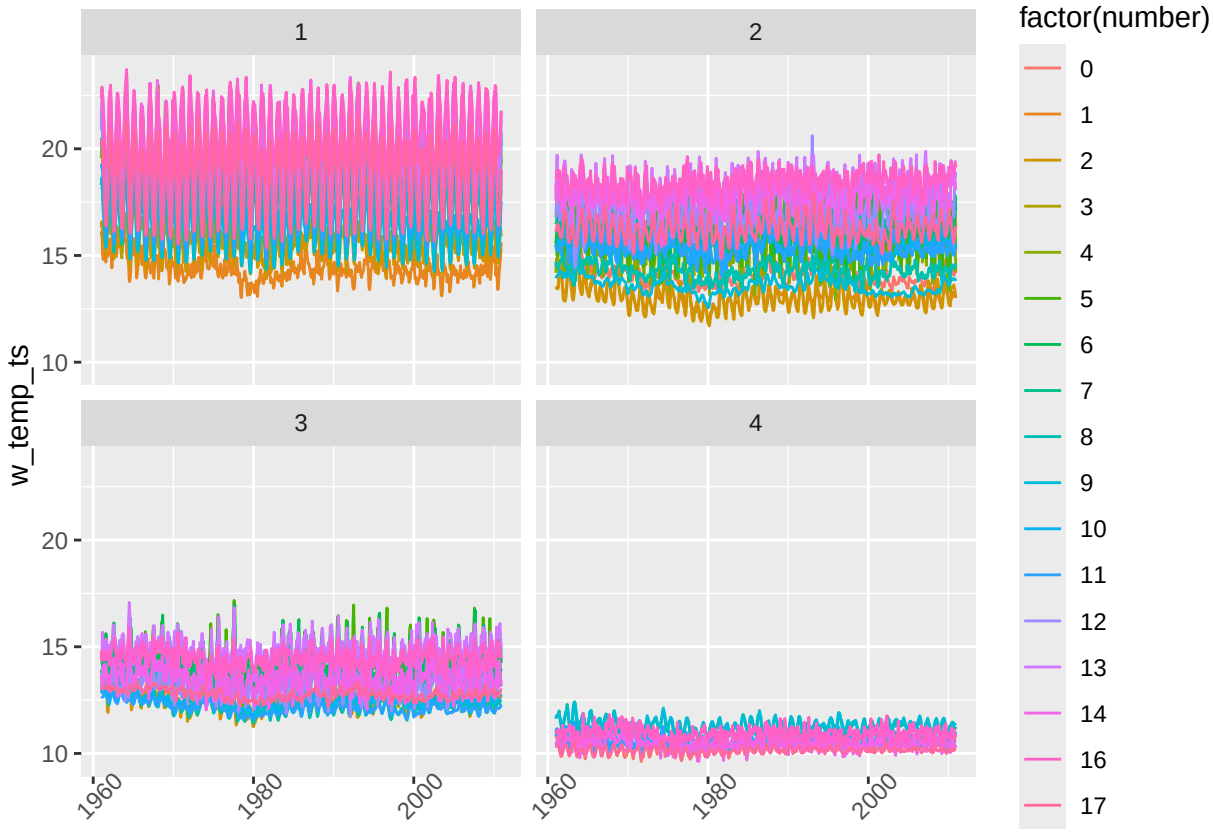
This looks exactly as we need it to be. We can now calculate the weighted temperature for every timestep in our dataset.

Calculate monthly weighted temperature means

```
#Process data from GFDL obsclim
sb_gfdl_box <- temp_sb |>
  #the maximum depth of the southern Benguela Atlantis model is 500 m
  filter(depth <= 500) |>
  drop_na(temperature) |>
  #Transforming date column to date format
  mutate(date = as.Date("1961-01-01") %m+% months(round(time/30.417, 0))) |>
  #Defining depth categories for the southern Benguela Atlantis model
  mutate(layer = case_when(depth < 51 ~ 1, depth > 51 & depth < 101 ~ 2,
                           depth > 100 & depth < 301 ~ 3, depth > 300 ~ 4)) |>
  left_join(depth_bins, by = "depth") |>
  #Adding area of grid cell
  left_join(sb_areacello, by = c("lat", "lon")) |>
  #Adding information about Southern Benguela subregions
  left_join(cb_mask, by = c("lat", "lon")) |>
  #Calculate volume of grid cell
  mutate(volume = depth_height*area) |>
  #Excludes grid cells that are not present in GFDL
  drop_na(volume, temperature) |>
  #Calculated weighted means by depth group, timestep and subregion
  group_by(layer, date, number) |>
  #Using area as weights in mean calculation
  summarise(w_temp_ts = weighted.mean(temperature, volume, na.rm = TRUE)) |>
  arrange(number, date)
```

```
## `summarise()` has grouped output by 'layer', 'date'. You can override using the
## `.groups` argument.
```

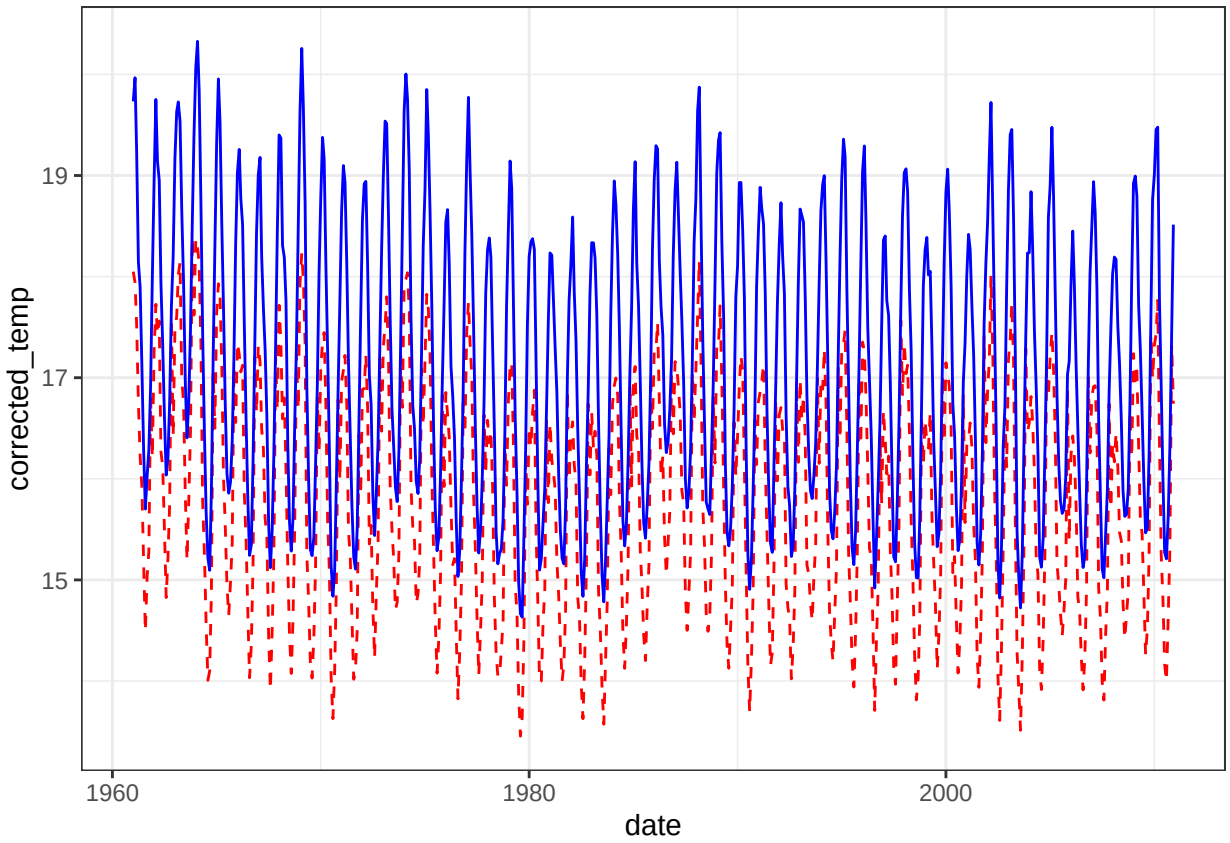
```
#plotting weighted temperature from January 1961 to December 2010
sb_gfdl_box |>
  # filter(number == 0 & layer == 1)|>
  ggplot(aes(date, w_temp_ts, color = factor(number))) +
  geom_line(aes(group = number)) +
  #Subplot by depth layer
  facet_wrap(~layer)+
  theme(axis.text.x = element_text(angle = 45),
        axis.title.x = element_blank())
```



Calculate bias corrected temperature time series

We will join all datasets together to make bias correction easier.

```
#Join monthly means with monthly climatologies from GFDL and WOA
sb_temp_corr_box <- sb_gfdl_box |>
  mutate(month = factor(month.name[month(date)])) |>
  left_join(sb_gfdl_summaries, by = c("layer", "month", "number")) |>
  left_join(sb_woa_summaries, by = c("layer", "month", "number")) |>
  #Apply correction
  mutate(corrected_temp = (w_temp_ts-w_temp_gfdl)+w_temp_woa)

#Plotting results for first depth layer and first subregion
sb_temp_corr_box |>
  ungroup() |>
  filter(layer == min(layer) & number == min(number)) |>
  ggplot(aes(date))+
  geom_line(aes(y = corrected_temp), linetype = "dashed", color = "red") +
  geom_line(aes(y = w_temp_ts), color = "blue")+
  theme_bw()
```



It appears that there may be differences larger than $1°C$. We will calculate the difference and check the results.

```
#calculate differences between datasets
sb_temp_corr_box |>
  mutate(diff = corrected_temp-w_temp_ts) |>
  arrange(desc(abs(diff))) |>
  #Arrange by absolute differences (from largest to smallest)
  select(!c(month, w_temp_woa, w_temp_gfdl)) |>
  head(12)
```

```
## # A tibble: 12 x 6
## # Groups:   layer, date [12]
##    layer date       number w_temp_ts corrected_temp  diff
##    <dbl> <date>      <int>     <dbl>          <dbl> <dbl>
##  1     2 1961-12-01      3      16.0           9.93 -6.09
##  2     2 1962-12-01      3      16.6          10.5  -6.09
##  3     2 1963-12-01      3      16.9          10.9  -6.09
##  4     2 1964-12-01      3      17.0          10.9  -6.09
##  5     2 1965-12-01      3      16.1          10.0  -6.09
##  6     2 1966-12-01      3      16.5          10.4  -6.09
##  7     2 1967-12-01      3      16.0           9.88 -6.09
##  8     2 1968-12-01      3      16.6          10.5  -6.09
##  9     2 1969-12-01      3      16.6          10.5  -6.09
## 10     2 1970-12-01      3      16.0           9.90 -6.09
## 11     2 1971-12-01      3      15.7           9.65 -6.09
```

```
## 12     2 1972-12-01      3      16.1           9.97 -6.09
```

Since differences are much larger than $1°C$ for some subregions (over $6°C$ in subregion 3), we recommend that the bias corrected temperature (`corrected_temp`) is used to force this ecosystem model.