

# GlobalZBio\_03\_SurfaceMap

Jasmine Fowler-Morrow

2022-10-17

```
source("utils.R") # load the harmonic function
library(visreg)
library(raster)

## Loading required package: sp

library(ggplot2)
library(terra)

## terra 1.7.71

library(sf)

## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:terra':
##       intersect, union

## The following objects are masked from 'package:raster':
##       intersect, select, union

## The following objects are masked from 'package:stats':
##       filter, lag

## The following objects are masked from 'package:base':
##       intersect, setdiff, setequal, union
```

```

library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## vforcats    1.0.0     vstringr    1.5.1
## vlubridate  1.9.3     vtibble      3.2.1
## vpurrr      1.0.2     vtidy       1.3.1
## vreadr      2.1.5

## -- Conflicts ----- tidyverse_conflicts() --
## x tidyr::extract() masks terra::extract(), raster::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks raster::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(splines)
library(patchwork)

##
## Attaching package: 'patchwork'
##
## The following object is masked from 'package:terra':
##
##     area
##
## The following object is masked from 'package:raster':
##
##     area

# IMPORT BIOMASS GLMM RDS FILE
mdl <- readRDS(file.path("Output", "Newglm12.rds"))

## IMPORT BATHYMETRY DATA AND ORIENT LATITUDES TO BE SOUTH TO NORTH
bathy_data <- readRDS(file.path("Data", "Bathy_raster_oneDeg.rds"))
bathy_matrix <- t(as.matrix(bathy_data$Bathy))
bathy_matrix <- bathy_matrix[, 180:1]

## Calculate areas of grid cells
glob_area <- t(as.matrix(raster::area(raster())))

```

Create the dataframe/array to produce predictions

```

##### Set up array to make predictions and save outputs #####
save_array <- array(NA, dim = c(12, 13, 64800))
# matches month x variable x bathy #
dimnames(save_array)[[1]] <- c("Jan", "Feb", "Mar", "Apr",
                               "May", "Jun", "Jul", "Aug",
                               "Sep", "Oct", "Nov", "Dec")
dimnames(save_array)[[2]] <- c("Longitude", "Latitude",
                               "BiomassMethod",
                               "Mesh", "Depth", "Gear",

```

```

        "DatasetID", "HarmTOD",
        "Bathy", "HarmDOY", "Thetao",
        "Chl", "GLM_Mesozoo")

## Set mesh, start depth and time of day
save_array[,"Mesh",] <- 25
save_array[,"Depth",] <- 0.5
save_array[,"HarmTOD",] <- 0

## Final output matrix, with non-important factors removed
save_array2 <- save_array[,-c(3,4,5,6,7,8,10),]
#keeps lon, lat, bathy, predictions

#Create a Longitude x Latitude matrix
lonlat <- as.matrix(expand.grid("lons" = -179.5:179.5,
                                 "lats" = -89.5:89.5))

## Harmonic day of year for each month
days_of_year_harmonic <- seq(15,365,30)/365*2*pi

## Depth is set to surface, this can be changed to a vector of depths...
depth <- 0.5

```

Now predict surface biomass distribution

```

## LOOP OVER EACH MONTH FOR BIOMASS PREDICTION
for (k in 1:12){
  print(paste0("Now working on month ",k))

  ## Import current month (k) sst and chl climatology
  curr_sst <-
    t(as.matrix(readRDS(list.files(
      path = './Data/',
      pattern = glob2rx(paste("SST*",
                                dimnames(save_array)[[1]][k], "*",
                                sep = "")), full.names = TRUE))))[, 180:1]
  curr_chl <-
    t(as.matrix(readRDS(list.files(
      path = './Data/',
      pattern = glob2rx(paste("Chl*",
                                dimnames(save_array)[[1]][k],
                                "*", sep = "")), full.names = TRUE))))[, 180:1]

  ## Fill in this month's slice of save_array
  save_array[k, "Longitude", ] <- lonlat[, 1]
  save_array[k, "Latitude", ] <- lonlat[, 2]
  save_array[k, "Bathy", ] <- as.vector(bathy_matrix)

  save_array[k, "HarmDOY", c(1:32400)] <-
    days_of_year_harmonic[k] # Southern hemisphere months

  save_array[k, "HarmDOY", c(32401:64800)] <-
    days_of_year_harmonic[k] # Northern hemisphere months

```

```

# input monthly SST and Chl
save_array[k, "Thetao", ] <- as.vector(curr_sst)
save_array[k, "Chl", ] <- as.vector(curr_chl)

#Align SST and chlo maps with bathy (where bathy is land, mask sst and chlo)
save_array[k, "Thetao",
           which(is.na(save_array[k, "Bathy", ] == TRUE))] <- NA
save_array[k, "Chl",
           which(is.na(save_array[k, "Bathy", ] == TRUE))] <- NA

# Convert to dataframe and add random effects factors and 0.5m depth for glmm prediction
kk <- as.data.frame(t(save_array[k, , ])) #dataframe of month k only
kk$BiomassMethod <- as.factor("Carbon")
kk$Gear <- as.factor("116")
kk$DatasetID <- as.factor("100")
kk$Depth2 <- depth

##### SURFACE LAYER PREDICTIONS #####
# Get surface layer estimate
kk$GLM_Mesozoo <- (exp(predict(mdl,
                                type = "link",
                                newdata = kk,
                                re.form = NA)))

## Dump output into this month's time slice of save array2
save_array2[k, "Longitude", ] <- as.vector(kk$Longitude)
save_array2[k, "Latitude", ] <- as.vector(kk$Latitude)
save_array2[k, "Bathy", ] <- as.vector(kk$Bathy)
save_array2[k, "Thetao", ] <- as.vector(kk$Thetao)
save_array2[k, "Chl", ] <- as.vector(kk$Chl)
save_array2[k, "GLM_Mesozoo", ] <- as.vector(kk$GLM_Mesozoo)
}

## [1] "Now working on month 1"
## [1] "Now working on month 2"
## [1] "Now working on month 3"
## [1] "Now working on month 4"
## [1] "Now working on month 5"
## [1] "Now working on month 6"
## [1] "Now working on month 7"
## [1] "Now working on month 8"
## [1] "Now working on month 9"
## [1] "Now working on month 10"
## [1] "Now working on month 11"
## [1] "Now working on month 12"

## now change the way monthly predictions are stored for average maps
biomass_monthly <- data.frame(Jan = NA, Feb = NA, Mar = NA,
                               Apr = NA, May = NA, Jun = NA,
                               Jul = NA, Aug = NA, Sep = NA,
                               Oct = NA, Nov = NA, Dec = NA,
                               Longitude = lonlat[,1],

```

```

        Latitude = lonlat[,2])

for (j in 1:12){
  biomass_monthly[,j] <- save_array2[j, "GLM_Mesozoo", ]
}

## code to take an average over a few months and produce/save figure
monthly_mean_plot <- function(months){

  names <- colnames(biomass_monthly[,months])
  plot_name <- paste(names[1],names[3], sep = "-") #plot title

  #take mean over multiple months
  biomass_monthly$mean <- rowMeans(biomass_monthly[,months], na.rm = T)

  ## plot with projection
  kk_sf <- biomass_monthly %>%
    sf::st_as_sf(coords=c("Longitude", "Latitude"))

  lon_lat <- "+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"

  sf::st_crs(kk_sf) <- lon_lat

  landmass <- rnaturalearth::ne_countries(scale = "large") %>%
    sf::st_as_sf(crs = lon_lat)

  # Mollweide equal-area projection
  moll <- "+proj=moll +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m no_defs"

  # Transform data using projection
  kk_transformed <- kk_sf %>%
    sf::st_transform(moll)

  # Transform the landmass to the same projection
  landmass <- landmass %>%
    sf::st_transform(crs = moll)

  p <- ggplot() +
    geom_sf(data = kk_transformed, aes(color = mean), size = 0.01) +
    geom_sf(data = landmass, fill = "grey20", color = NA, size = 0.01) +
    scale_color_viridis_c(trans = "log10",
                          na.value = "grey60",
                          name = expression(paste(
                            "Zooplankton biomass mg m"^-3)),
                          limits = c(5,150),oob = scales::squish) +
    theme_classic() +
    ggtitle(label = plot_name) +
    coord_sf(label_graticule = "SW")
  return(p)
}

##### Seasonal plots #####
#Dec- Feb

```

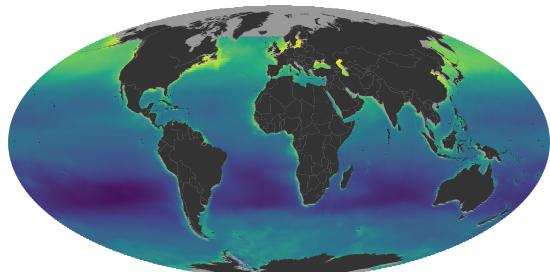
```

p1 <- monthly_mean_plot(c(12,1,2))
#Mar - May
p2 <- monthly_mean_plot(c(3:5))
#Jun - Aug
p3 <- monthly_mean_plot(c(6:8))
#Sep - Nov
p4 <- monthly_mean_plot(c(9:11))

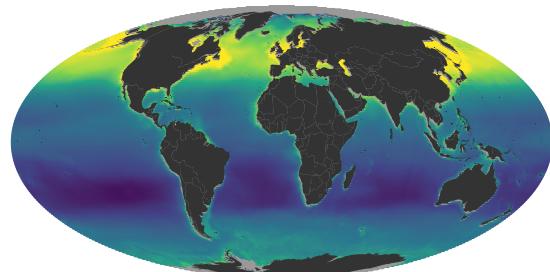
#combining them with incorrect colour scales is not a good visualisation
(p1 | p2) / (p3 | p4) + plot_annotation(tag_levels = "A", tag_suffix = ". ") +
  plot_layout(guides = "collect") & theme(legend.position = "bottom",
                                         plot.margin = margin(0.25,0.25,0.25,0.25),
                                         plot.title = element_text(size = 9),
                                         plot.tag = element_text(size = 9,
                                         hjust = -3, vjust = -4))

```

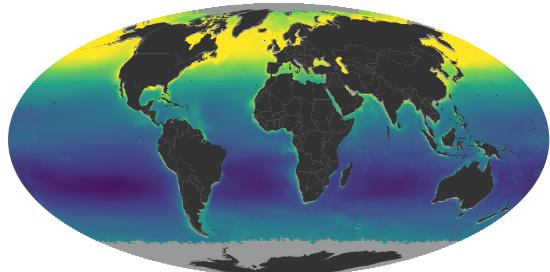
A. Dec–Feb



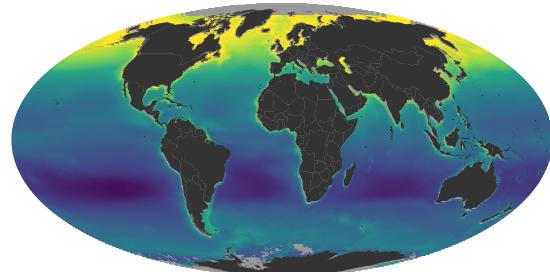
B. Mar–May



C. Jun–Aug



D. Sep–Nov



Zooplankton biomass  $\text{mg m}^{-3}$

10	30	100
----	----	-----

```

# save output
ggsave(paste0("./Figures/","monthly_maps",".jpeg"),
       width = 8, height = 6, dpi = 400)

```

```
## return only summer winter plot for powerpoint presentation
```

```

#combining them with incorrect colour scales is not a good visualisation
p1 / p3 + plot_annotation(tag_levels = "A", tag_suffix = ". ") +
  plot_layout(guides = "collect") & theme(legend.position = "right",
                                         plot.margin = margin(0.25,0.25,0.25,0.25),
                                         plot.title = element_text(size = 9),
                                         plot.tag = element_text(size = 9,
                                         hjust = -3, vjust = -4))

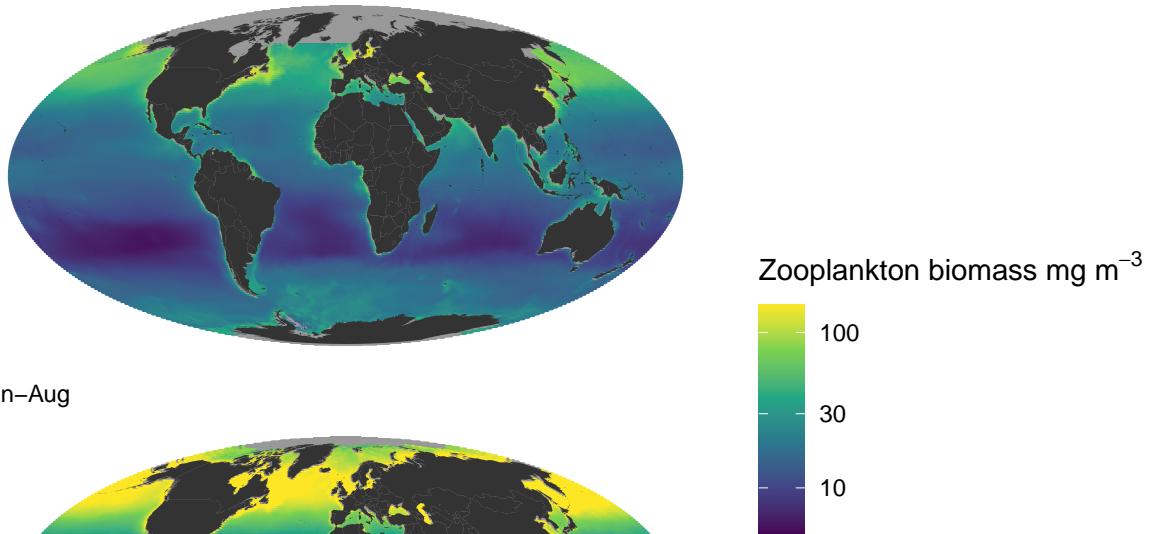
```

```

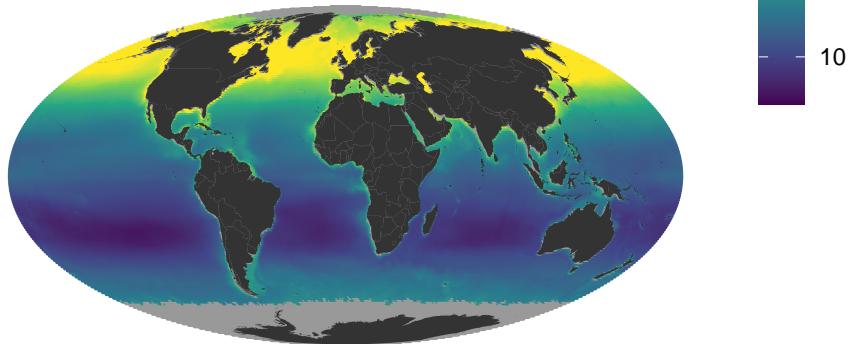
plot.margin = margin(0.25,0.25,0.25,0.25),
plot.title = element_text(size = 9),
plot.tag = element_text(size = 9,
hjust = -3, vjust = -4))

```

A. Dec–Feb



B. Jun–Aug



```

ggsave(paste0("./Figures/","NEW_monthly_map_pair",".jpeg"),
       width = 8, height = 6, dpi = 400)

```

```

## ADDITIONAL: Plot single month projection
#convert to sf
kk_sf <- kk %>%
  sf::st_as_sf(coords=c("Longitude", "Latitude"))

lonlat <- "+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"

sf::st_crs(kk_sf) <- lonlat

landmass <- rnaturalearth::ne_countries(scale = "large") %>%
  sf::st_as_sf(crs = lonlat)

# Mollweide equal-area projection
moll <- "+proj=moll +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m no_defs"

# Transform data using projection
kk_transformed <- kk_sf %>%

```

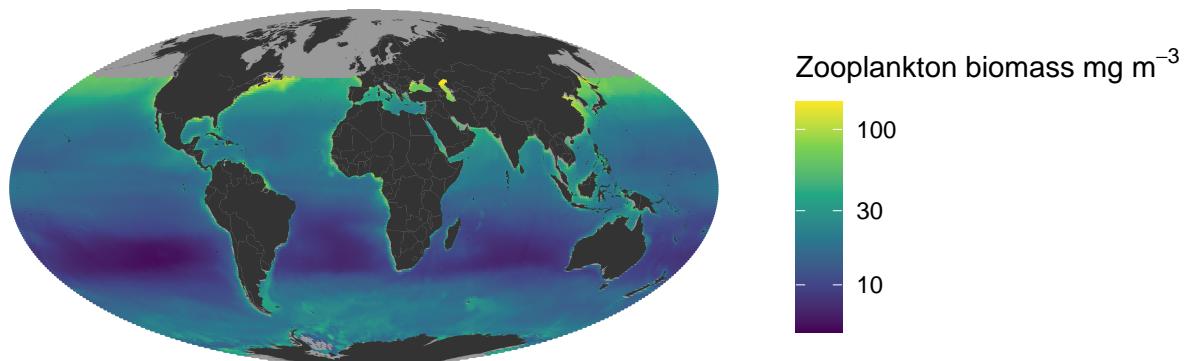
```

sf::st_transform(moll)

# Transform the landmass to the same projection
landmass <- landmass %>%
  sf::st_transform(crs = moll)

ggplot() +
  geom_sf(data = kk_transformed, aes(color = GLM_Mesozoo), size = 0.01) +
  geom_sf(data = landmass, fill = "grey20", color = NA, size = 0.01) +
  scale_color_viridis_c(trans = "log10",
    na.value = "grey60",
    name = expression(paste(
      "Zooplankton biomass mg m"^-3)),
    limits = c(5,150), oob = scales::squish) +
  theme_classic()

```



```

ggsave(paste0("./Figures/","year_map",".jpeg"),
       width = 8, height = 5, dpi = 400)

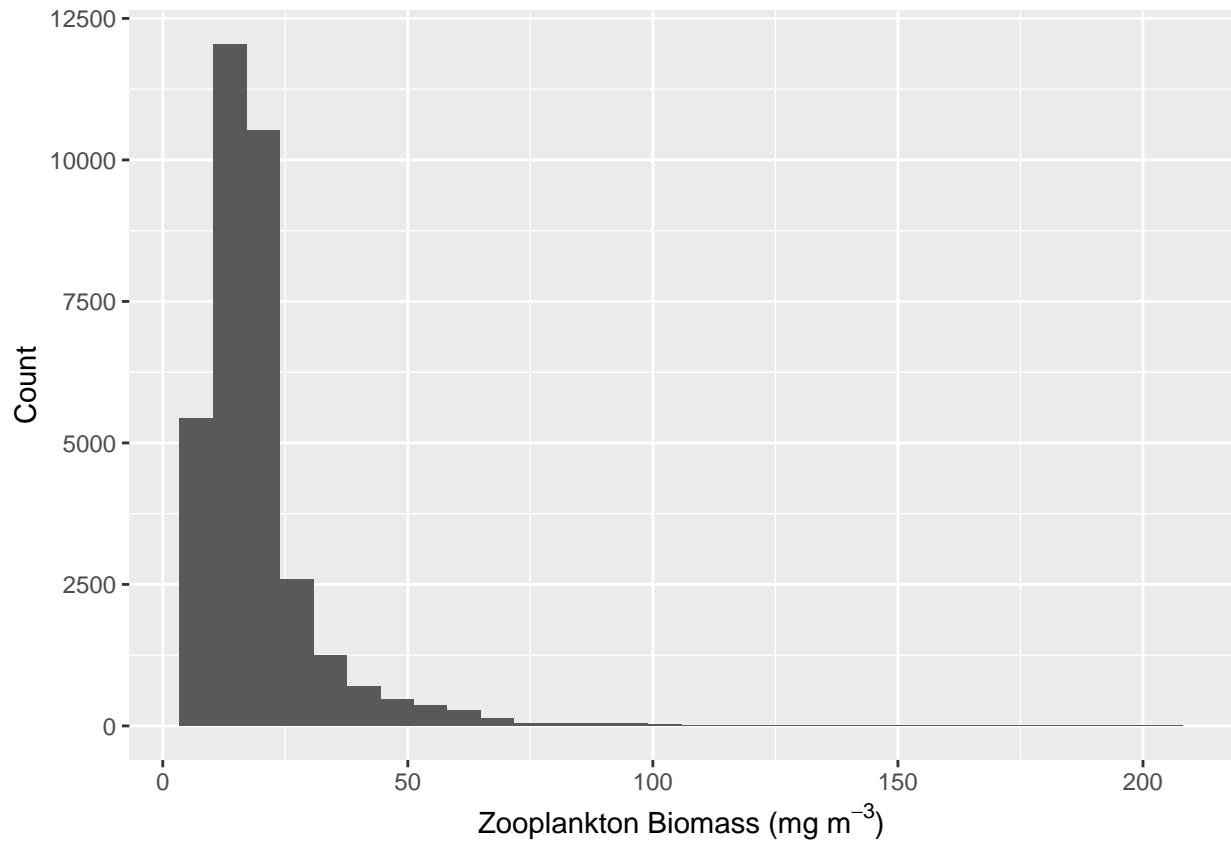
## histogram of prediction values ##
ggplot() + geom_histogram(data = kk, aes(GLM_Mesozoo)) +
  labs(x = expression("Zooplankton Biomass (mg m"^-3*")"),
       y = "Count")

```

```

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 30766 rows containing non-finite values ('stat_bin()').

```

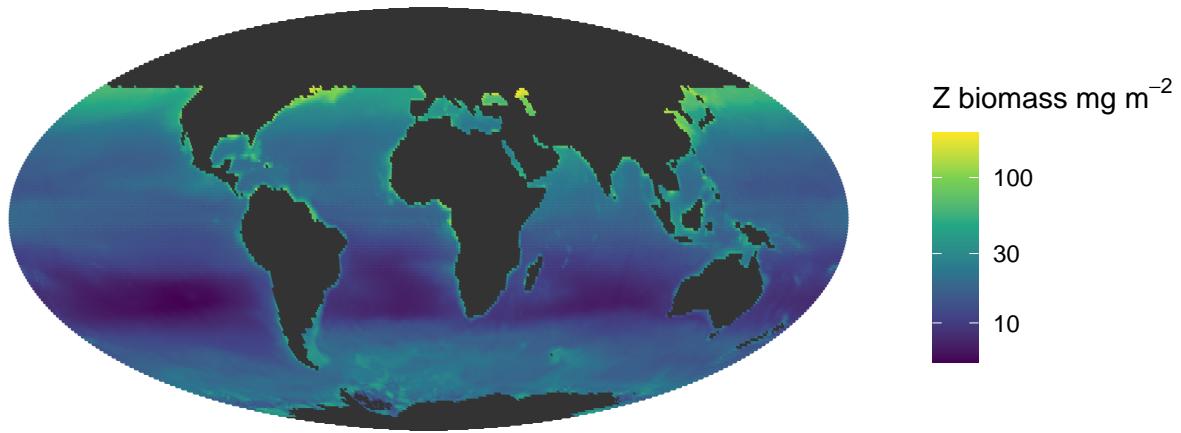


```

#ADDITIONAL: plot code
kk_transformed2 <-
  sf_transform_xy(kk_sf, target_crs = moll, source_crs = lonlat)

ggplot() +
  geom_sf(data = landmass, fill = "grey20", color = NA, size = 0.01) +
  geom_sf(data = kk_transformed2, aes(color = GLM_Mesozoo), size = 0.01) +
  scale_color_viridis_c(trans = "log10",
                        na.value = "grey20",
                        name = expression(paste("Z biomass mg m"^-2))) +
  theme_minimal()

```



## TMAP TRIAL

```
## Map yearly average biomass using tmaps

#take mean over 12 months
biomass_monthly$mean <- rowMeans(biomass_monthly[,1:12], na.rm = T)

## plot with projection
kk_sf <- biomass_monthly %>%
  sf::st_as_sf(coords=c("Longitude", "Latitude"))

## use CRS
lon_lat <- "+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"

sf::st_crs(kk_sf) <- lon_lat

landmass <- rnatural-earth::ne_countries(scale = "large") %>%
  sf::st_as_sf(crs = lon_lat)

# Mollweide equal-area projection
moll <- "+proj=moll +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m no_defs"

#moll is 54009

# Transform data using projection
kk_transformed <- kk_sf %>%
```

```

sf::st_transform(crs = moll)

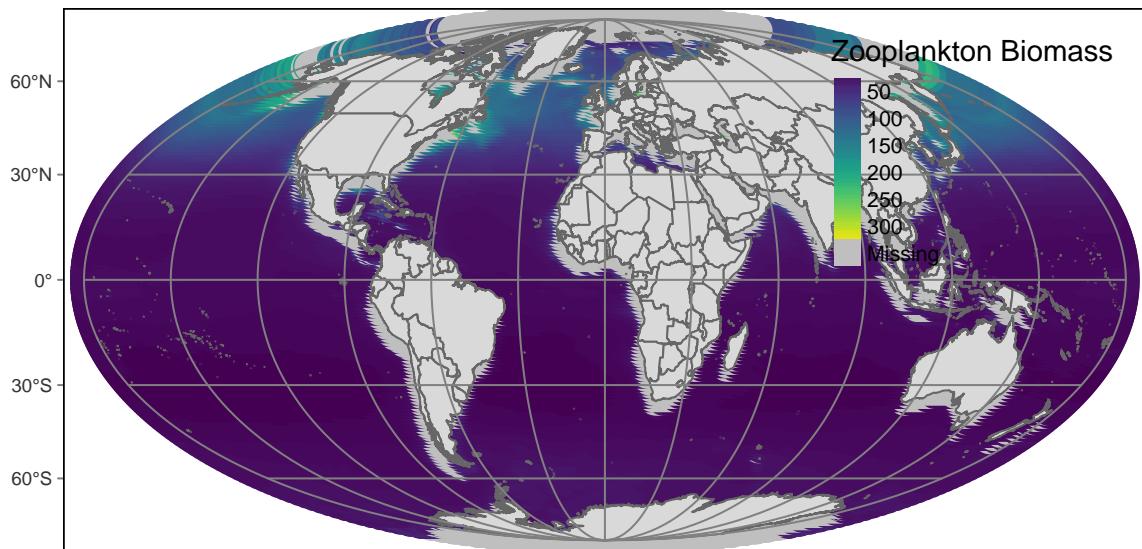
# Transform the landmass to the same projection
landmass <- landmass %>%
  sf::st_transform(crs = moll)

library(tmap)

## Breaking News: tmap 3.x is retiring. Please test v4, e.g. with
## remotes::install_github('r-tmap/tmap')

tm_shape(kk_transformed) +
  tm_dots(col = "mean", title = "Zooplankton Biomass",
          style = "cont", palette = "viridis",
          size = 1) +
  tm_shape(landmass) +
  tm_polygons() + tm_graticules()

```



Next step: do uncertainty map for annual mean surface biomass