

## Step2\_Environmental\_data

this is a soil wetness index from 2013 downloaded from the Eco-Commons site

[https://api.data-ingester.app.ecocommons.org.au/api/data/69b045b3-29d1-5068-a5e6-f0783c68c1b8/download/awap\\_run26j\\_2013\\_ann-TempMin.tif](https://api.data-ingester.app.ecocommons.org.au/api/data/69b045b3-29d1-5068-a5e6-f0783c68c1b8/download/awap_run26j_2013_ann-TempMin.tif)

Simply enter the above url, download the file, and load it from the directory path you saved it

#- ideally you would match the time the layer was made to the time # your occurrence data was collected, and you might average over the last ten years, but this is just an example of the kinds of data you can use # the EcoCommons point-and-click environment does this for you when in the SDM workflow #you can select to make all the variables the finest or coarsest resolution

## Environmental data

```
library(raster)
```

```
## Warning: package 'raster' was built under R version 4.1.2
```

```
## Loading required package: sp
```

```
require(rgdal)
```

```
## Loading required package: rgdal
```

```
## Warning: package 'rgdal' was built under R version 4.1.2
```

```
## Please note that rgdal will be retired by the end of 2023,  
## plan transition to sf/stars/terra functions using GDAL and PROJ  
## at your earliest convenience.
```

```
##
```

```
## rgdal: version: 1.5-29, (SVN revision 1165M)
```

```
## Geospatial Data Abstraction Library extensions to R successfully loaded
```

```
## Loaded GDAL runtime: GDAL 3.4.0, released 2021/11/04
```

```
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 8.1.1, September 1st, 2021, [PJ_VERSION: 811]
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.4-6
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
## Overwritten PROJ_LIB was /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/proj
```

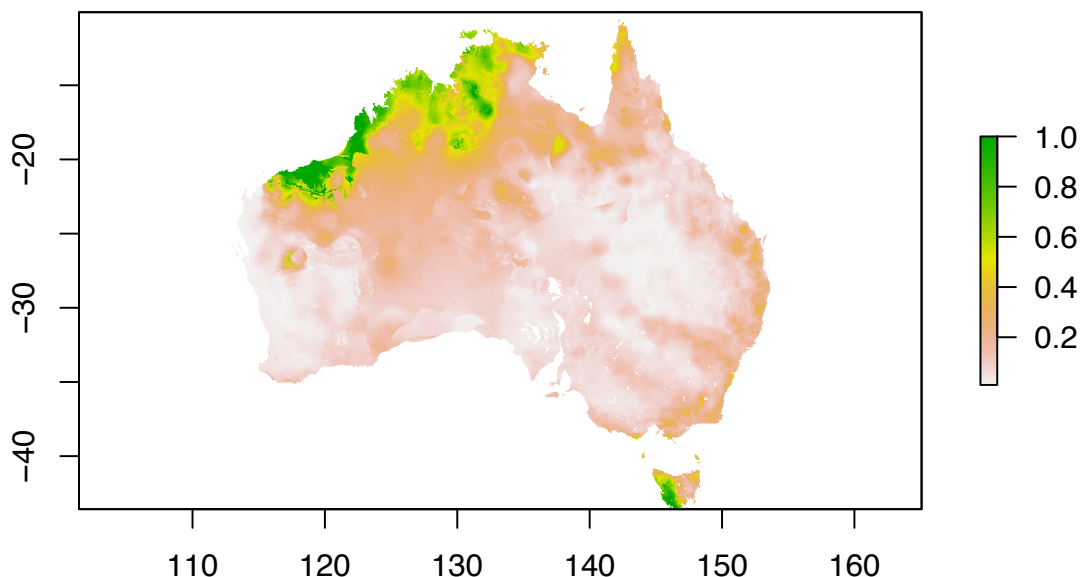
```
direct<- "/Users/s2992269/Documents/Use_cases"
folder <- "/SDM_in_R"
```

```
#this sets your working director for all subsequent chunks of code in your R Markdown script
knitr::opts_knit$set(root.dir = paste0(direct,folder))
```

```
# if you are not working R Markdown, simply use this instead
# setwd(paste0(direct,folder))
```

or use if not using Markdown `setwd(paste0(direct,folder))`

```
AWAP1<-raster("raw_data/awap_WRel1End.tif")
plot(AWAP1)
```



```
AWAP1
```

```
## class      : RasterLayer
## dimensions : 670, 813, 544710 (nrow, ncol, ncell)
## resolution : 0.05, 0.05 (x, y)
## extent     : 112.925, 153.575, -43.575, -10.075 (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : awap_WRel1End.tif
## names      : awap_WRel1End
## values     : 0.01, 1 (min, max)
```

```
base <- raster("data/base_LiPe.asc")
crs(base) <- "+proj=longlat +datum=WGS84 +no_defs"
```

reduce the resolution of this layer on soil wetness, and match the extent of base

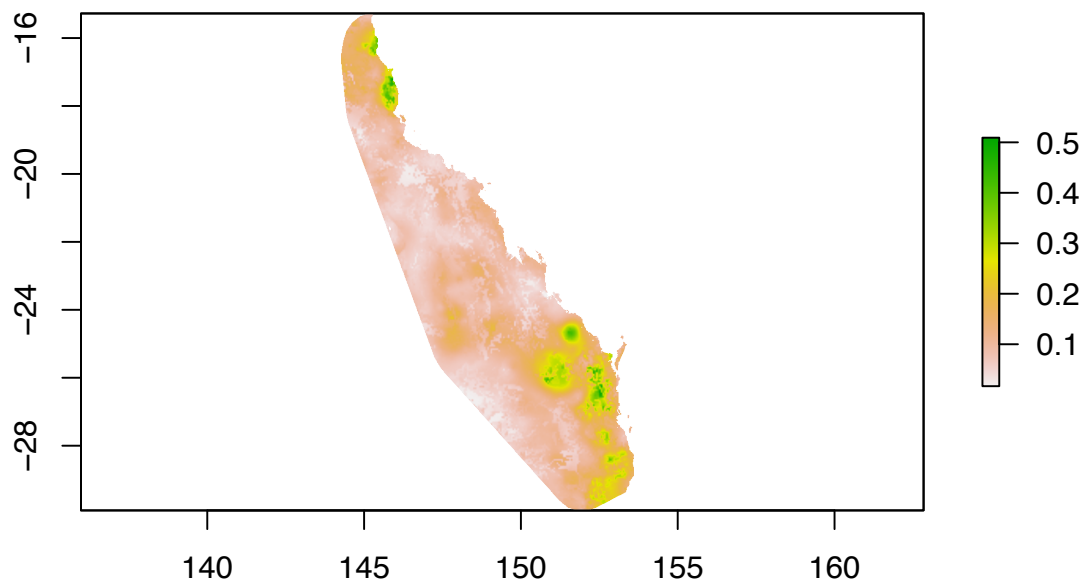
note there are a variety of ways to this. In some cases these decisions on how

to reduce or increase resolution matter to the result. Often the defaults are fine.

know your data, and dig deeper into the methods underlying these functions

```
#try typing # ??raster::projectRaster
```

```
AWAP2 <-projectRaster(AWAP1,base)
AWAP3<- mask(AWAP2,base)
plot(AWAP3)
```



```
AWAP <- AWAP3
```

#now we want to be sure the name of the variable AWAP is the same in memory, and in a folder # full of all predictors so we can predict our fitted model into the entire area # possibly using other point occurrence data (independent data)

use write raster to save files locally

```
writeRaster(AWAP,"predictors/AWAP.asc")
```

```
writeRaster(AWAP,"predictors_future/AWAP.asc")
```

Lets now upload the Vegetation classification layer NVIS from Eco-Commons

[https://api.data-ingester.app.ecocommons.org.au/api/data/ffcf7885-95e0-5873-8178-4a5ecc14123b/download/nvis-2020-90m\\_\\_aus6\\_\\_0e\\_\\_mvg\\_\\_amvg.tif](https://api.data-ingester.app.ecocommons.org.au/api/data/ffcf7885-95e0-5873-8178-4a5ecc14123b/download/nvis-2020-90m__aus6__0e__mvg__amvg.tif)

```
NVIS1<-raster("raw_data/nvis-2020-90m_aus6_0e_mvg_amvg.tif")
```

each grid cell value should correspond to one of these categories

<https://www.awe.gov.au/sites/default/files/env/pages/ba1d4b30-d46f-42f7-bec2-fac391f26072/files/mvsg60-sort-order.pdf>

```
freq(NVIS1)
```

##	value	count
## [1,]	1	2903085
## [2,]	2	4123266
## [3,]	3	22197543
## [4,]	4	1100968
## [5,]	5	89816367
## [6,]	6	40915445
## [7,]	7	3351159
## [8,]	8	2926752
## [9,]	9	9533682
## [10,]	10	5224155
## [11,]	11	36891505
## [12,]	12	15209845
## [13,]	13	30601424
## [14,]	14	20100162
## [15,]	15	1925547
## [16,]	16	83156374
## [17,]	17	10908739
## [18,]	18	3844584
## [19,]	19	49298488
## [20,]	20	141100779

```
## [21,] 21 10693445
## [22,] 22 48712751
## [23,] 23 1073238
## [24,] 24 8510969
## [25,] 25 101612496
## [26,] 26 79395
## [27,] 27 5884151
## [28,] 28 1123643
## [29,] 29 2202202
## [30,] 30 26
## [31,] 31 12788321
## [32,] 32 1174792
## [33,] NA 769651702
```

```
freq(NVIS1,value=44)
```

```
## [1] 0
```

habitat of frog <https://environment.des.qld.gov.au/wildlife/animals/a-z/striped-marsh-frog>

it is hard to see one of these classes being super helpful, class 44 freshwater did not return anything

if we did use it we need to be careful changing the resolution because these are categorical variables

(use nearest neighbor when changing resolution)

Randome useful tidbit

A handy function to clip points to a polygon

```
clipped points <- spatial_point_file[polygon_file, ]
```

get a wetland file

these steps take too long to run here, so we supply the wetland layer, steps are below

download an Australia wetland shapefile from here

<https://ecat.ga.gov.au/geonetwork/srv/eng/catalog.search#/metadata/83135>

select a 0.1 degree grid that covers Australia

read in Wetland polygones

```
wetland <- st_read("SurfaceHydrologyPolygonsNational.gdb")
```

then turn polygons into raster

the argument getCover in the rasterize function calculates the area of each cell covered by wetland

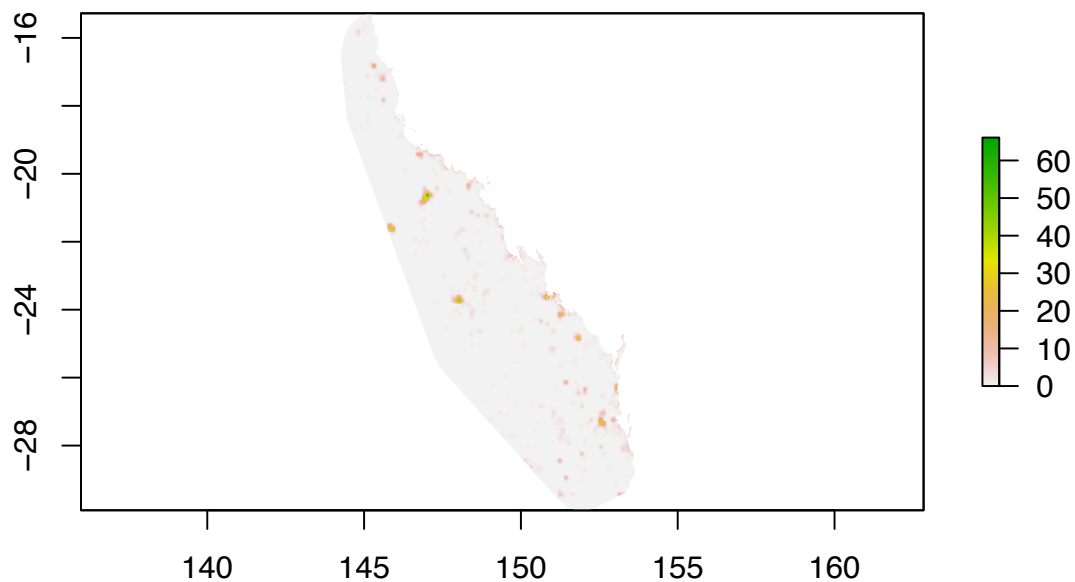
```
wetland <- rasterize(wetland,Point1_degree_grid,getCover = TRUE)
```

for this variable it would have made more sense to source a polygon layer of freshwater only wetland

```
wetland1 <- raster('raw_data/wet_cov.asc')
crs(wetland1) <- "+proj=longlat +datum=WGS84 +no_defs"
wetland2 <- projectRaster(wetland1, base)
```

```
## Warning in projectRaster(wetland1, base): input and output crs are the same
```

```
wetland3 <- mask(wetland2, base)
plot(wetland3)
```



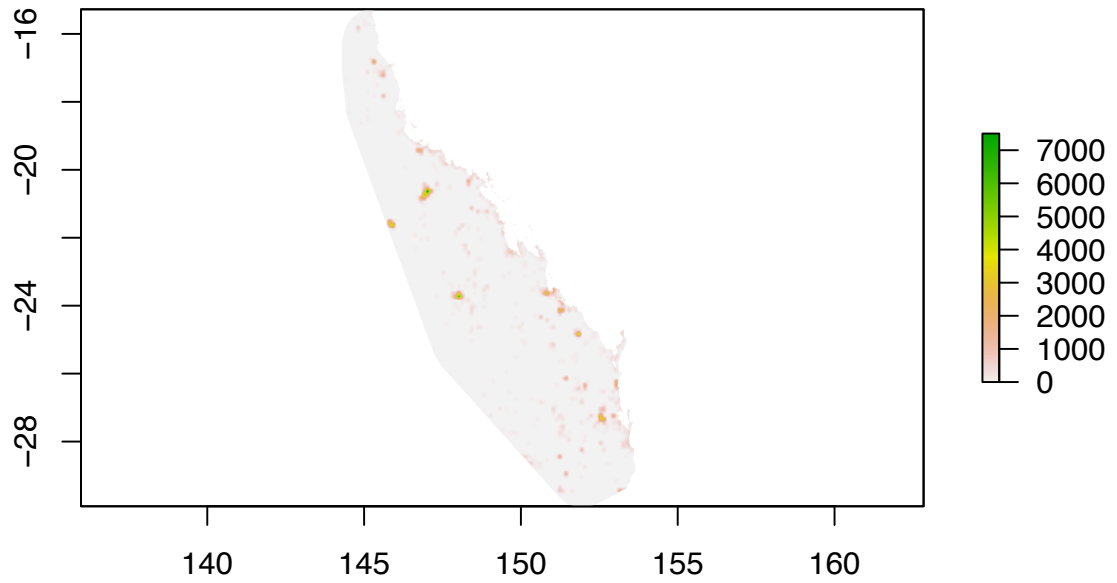
Here we use the focal function (a neighborhood function) to transform each grid cell

value to the sum of the central cell and all cells 5 cells away surrounding that central cell

this is a surrogate of connectivity, isolated wetlands will have lower values



```
wetland_connectivity <- focal(wetland3, w=matrix(1, nrow = 11, ncol = 11), fun=sum, na.rm=TRUE)  
plot(wetland_connectivity)
```



## Step1b

this is a soil wetness index from 2013 downloaded from the Eco-Commons site

[https://api.data-ingester.app.ecocommons.org.au/api/data/69b045b3-29d1-5068-a5e6-f0783c68c1b8/download/awap\\_\\_run26j\\_\\_2013\\_\\_ann-TempMin.tif](https://api.data-ingester.app.ecocommons.org.au/api/data/69b045b3-29d1-5068-a5e6-f0783c68c1b8/download/awap__run26j__2013__ann-TempMin.tif)

Simply enter the above url, download the file, and load it from the directory path you saved it

#- ideally you would match the time the layer was made to the time # your occurrence data was collected, and you might average over the last ten years, but this is just an example of the kinds of data you can use # the EcoCommons point-and-click environment does this for you when in the SDM workflow #you can select to make all the variables the finest or coarsest resolution

## Environmental data

```
library(raster)
```

```
## Warning: package 'raster' was built under R version 4.1.2
```

```
## Loading required package: sp
```

```
library(rgdal)
```

```
## Warning: package 'rgdal' was built under R version 4.1.2
```

```
## Please note that rgdal will be retired by the end of 2023,  
## plan transition to sf/stars/terra functions using GDAL and PROJ  
## at your earliest convenience.  
##
```

```
## rgdal: version: 1.5-29, (SVN revision 1165M)
```

```
## Geospatial Data Abstraction Library extensions to R successfully loaded
```

```
## Loaded GDAL runtime: GDAL 3.4.0, released 2021/11/04
```

```
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/gdal
```

```
## GDAL binary built with GEOS: TRUE
```

```
## Loaded PROJ runtime: Rel. 8.1.1, September 1st, 2021, [PJ_VERSION: 811]
```

```
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.4-6
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
## Overwritten PROJ_LIB was /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/proj
```

```
direct<- "/Users/s2992269/Documents/Use_cases"
folder <- "/SDM_in_R"

#this sets your working director for all subsequent chunks of code in your R Markdown script
knitr::opts_knit$set(root.dir = paste0(direct,folder))

# if you are not working R Markdown, simply use this instead
# setwd(paste0(direct,folder))
```

**Write rasters if needed**

```
writeRaster(wetland_connectivity,"predictors/wetland_connectivity.asc")
```

```
writeRaster(wetland_connectivity,"predictors_future/wetland_connectivity
```

data comes in many formats, and uploading often requires different methods

NetSDF files are increasingly common, below is a script to bring in that file type

```
#require(sf) #require(ncdf4) #require(rasterVis) #require(raster)
```

```
MXtemp <- brick("Terraclim_EY_NSW.nc", varname="tmax")
# this NetCDF file includes many
```

```
#days of maximum temperature data with that subset of data having the varname = "tmax" # MX-
temp_mean <- mean(MXtemp) # if we just want one layer which is the mean of those daily totals
```

**we will now download a vegetation greenness index from EcoCommons**

```
#https://api.data-ingester.app.ecocommons.org.au/api/data/34ab5ea6-650f-503f-9446-88d0ae9effe1/
download/ndlc-2004-250m_trend-evi-mean.tif
```

download the file directly, but here we upload the file to save time

```
EVI1<- raster("https://api.data-ingester.app.ecocommons.org.
au/api/data/34ab5ea6-650f-503f-9446-88d0ae9effe1/download/
ndlc-2004-250m__trend-evi-mean.tif")
```

```
getwd()
```

```
## [1] "/Users/s2992269/Documents/Use_cases/SDM_in_R"
```

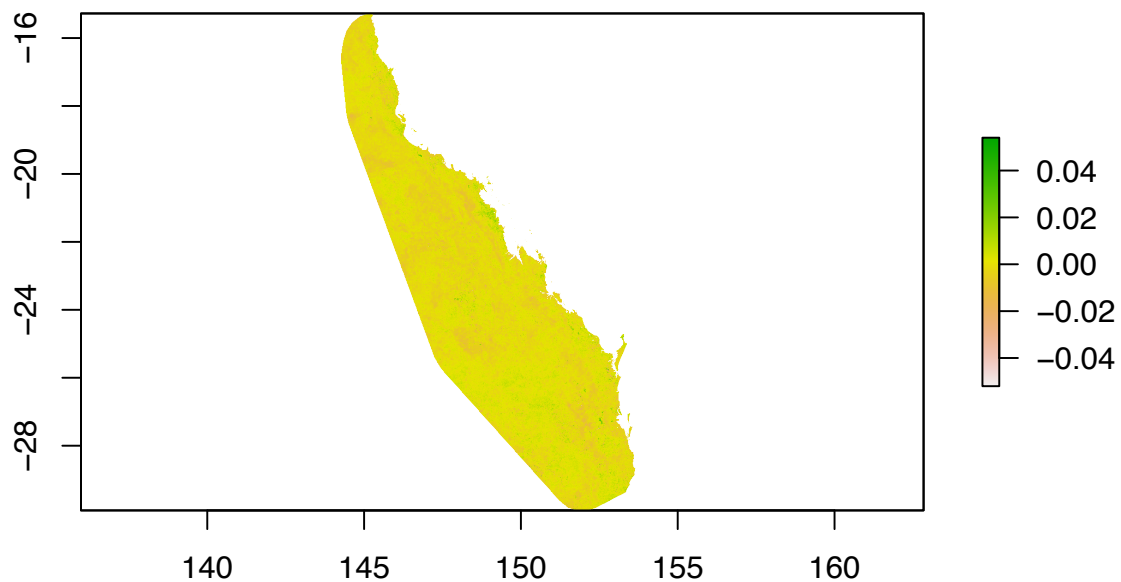
```
EVI1 <- raster::raster("Raw_data/EVI.asc")
raster::crs(EVI1)<-"+proj=longlat +datum=WGS84 +no_defs"
base <- raster::raster("data/base_LiPe.asc")
raster::crs(base)<-"+proj=longlat +datum=WGS84 +no_defs"
AWAP <- raster::raster("Raw_data/AWAP.asc")
raster::crs(AWAP)<-"+proj=longlat +datum=WGS84 +no_defs"
wetland_connectivity <- raster::raster("Raw_data/AWAP.asc")
raster::crs(wetland_connectivity)<-"+proj=longlat +datum=WGS84 +no_defs"
```

process it as previous layers

```
EVI2 <-projectRaster(EVI1,base)
```

```
## Warning in projectRaster(EVI1, base): input and ouput crs are the same
```

```
EVI3<- mask(EVI2,base)
plot(EVI3)
```



```
EVI <- EVI3
```

write rasters

```
writeRaster(EVI,"predictors/EVI.asc")
```

```
writeRaster(EVI,"predictors_future/EVI.asc")
```

Here we just show how to bring in current and future data from EcoCommons

after looking at correlations between bioclim variables this one was dropped

```
#Bioclim01_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/90317596-ddef-5666-91c5-9cbc25c24fbc/download/current_1976-2005_bioclim-01.tif") #Bioclim01_2 <- projectRaster(Bioclim01_1,base)
#Bioclim01_3 <- mask(Bioclim01_2 ,base) #plot(Bioclim01_3) #Bioclim01 <- Bioclim01_3 #writeRaster(Bioclim01,"predictors/Bioclim01.asc")
```

we will not write the bioclim data to the predictions\_future folder because

we have separate future climate predictions we can use. We do not have future

predictions for EVI or Wetlands so we are assuming those things will stay the same

in the future, the names of the future variables need to be the same as the

current time variable names in order for the predict function to work, so just

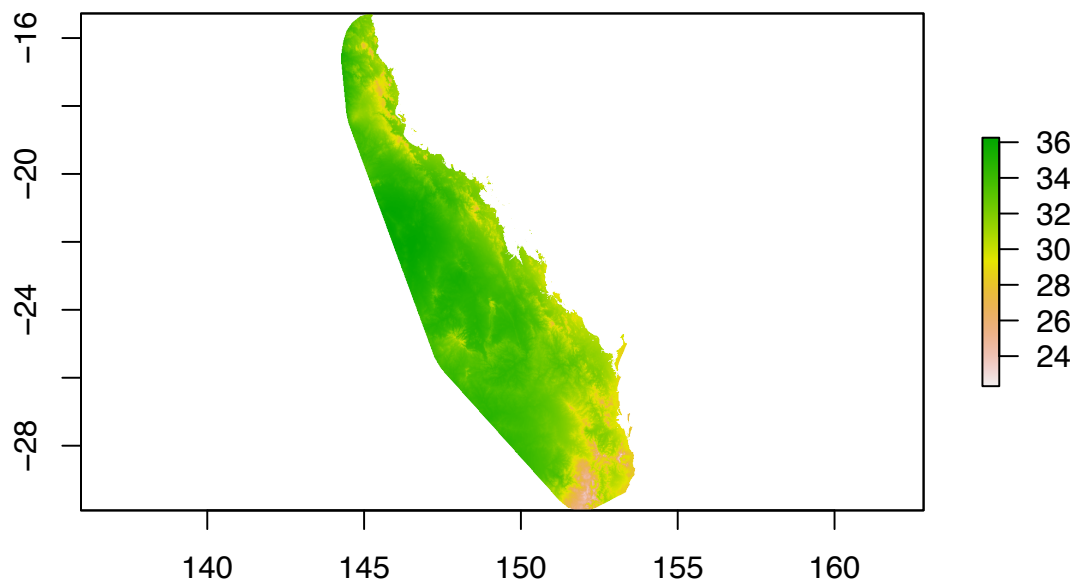
be sure to keep the different variables with the same name in different folders.

```
Bioclim05_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/cc081daa-f524-58c2-939e

## Error in .local(.Object, ...) : Empty reply from server

## Warning in .rasterFromGDAL(x, band = band, objecttype, ...): Could not read RAT
## or Category names

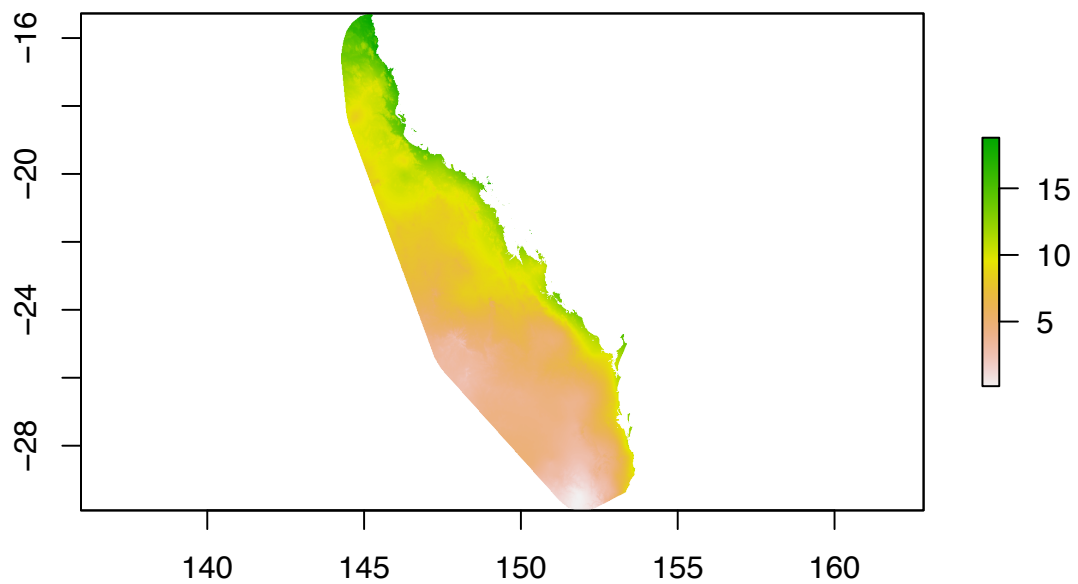
Bioclim05_2 <-projectRaster(Bioclim05_1,base)
Bioclim05_3 <- mask(Bioclim05_2 ,base)
plot(Bioclim05_3)
```



```
Bioclim05 <- Bioclim05_3
```

```
writeRaster(Bioclim05,"predictors/Bioclim05.asc")
```

```
Bioclim06_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/476e4343-99f2-578e-b44a")
Bioclim06_2 <- projectRaster(Bioclim06_1,base)
Bioclim06_3 <- mask(Bioclim06_2,base)
plot(Bioclim06_3)
```

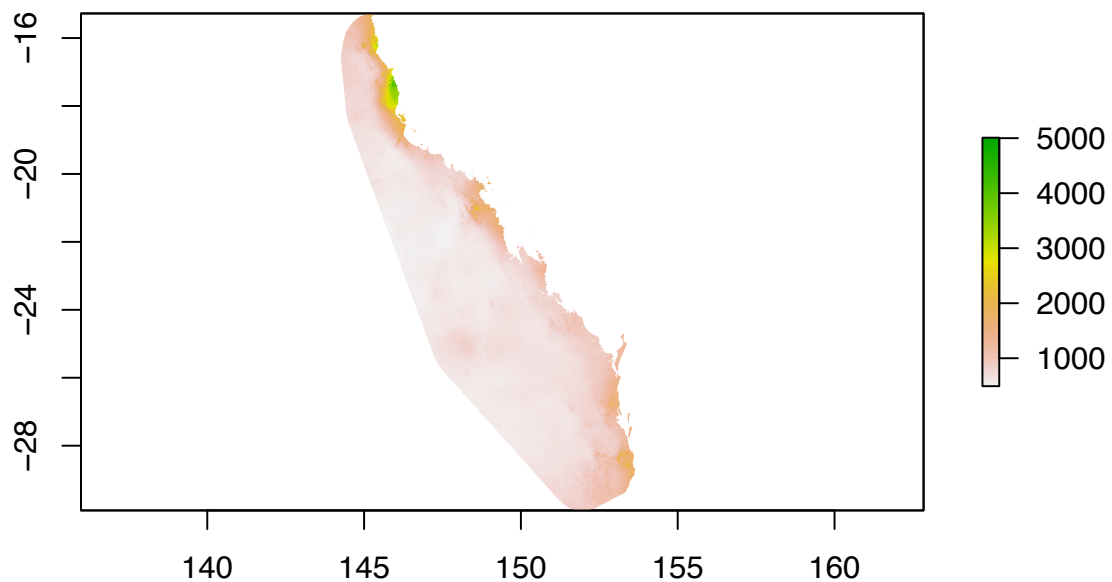


```
Bioclim06 <- Bioclim06_3
```

```
writeRaster(Bioclim06,"predictors/Bioclim06.asc")
```

```
Bioclim12_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/b2d70413-6b74-5366-b5d5  
Bioclim12_2 <-projectRaster(Bioclim12_1,base)  
Bioclim12_3 <- mask(Bioclim12_2,base)  
plot(Bioclim12_3)
```





```
Bioclim12 <- Bioclim12_3
```

```
writeRaster(Bioclim12,"predictors/Bioclim12.asc")
```

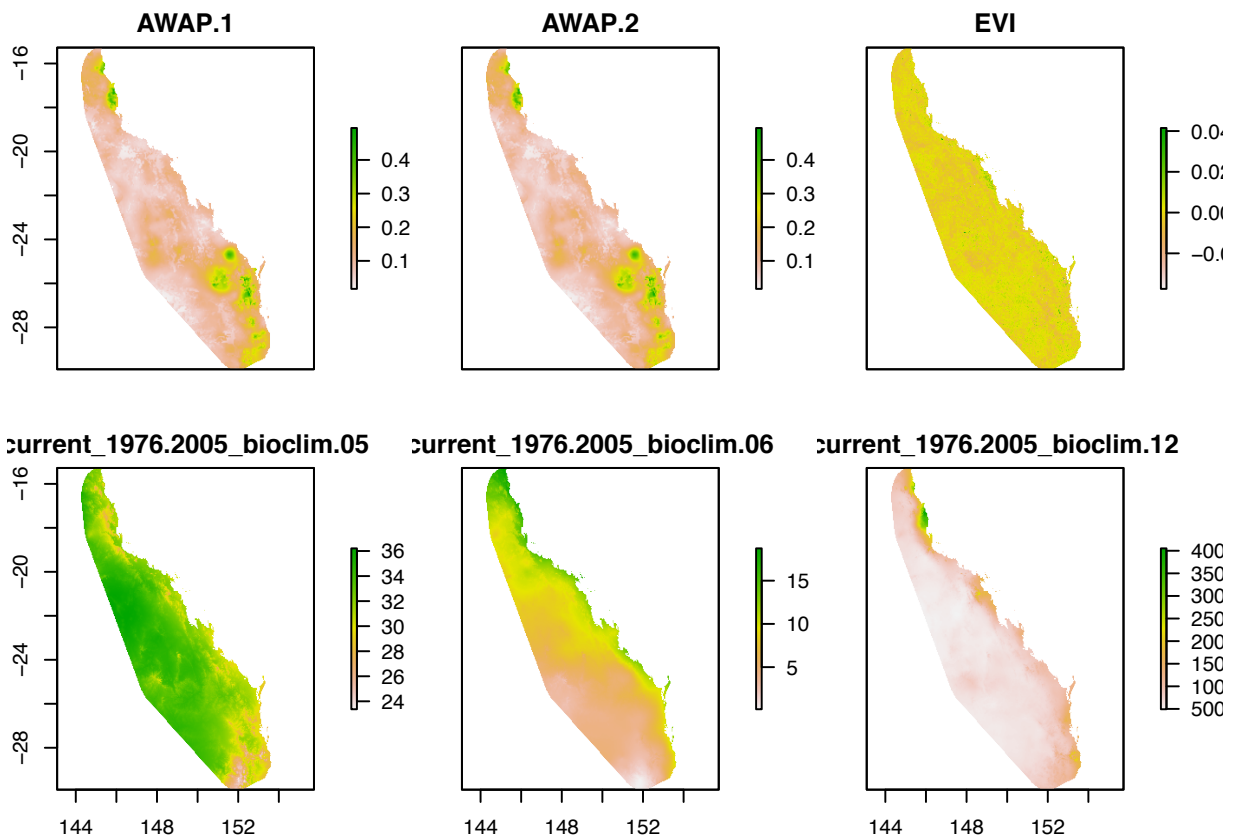
```
rm(Bioclim05_1,Bioclim05_2,Bioclim05_3,Bioclim06_1,Bioclim06_2,Bioclim06_3,Bioclim12_1,Bioclim12_2,Bioclim12_3)
```

after looking at correlations between bioclim variables this one was dropped

```
#Bioclim14_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/e64dfa1d-ece8-50e7-9ed1-d603bfb7a101/download/current_1976-2005_bioclim-14.tif") #Bioclim14_2 <- projectRaster(Bioclim14_1,base)
#Bioclim14_3 <- mask(Bioclim14_2 ,base) #plot(Bioclim14_3) #Bioclim14 <- Bioclim14_3 #writeRaster(Bioclim14,"predictors/Bioclim14.asc")
```

```
#create a raster stack of current data, be careful not to include the future variables below in this stack (they have the same names)
```

```
preds_current <- stack(AWAP,wetland_connectivity,EVI,Bioclim05,Bioclim06,Bioclim12)
plot(preds_current)
```



```
names(preds_current)
```

```
## [1] "AWAP.1" "AWAP.2"
## [3] "EVI" "current_1976.2005_bioclim.05"
## [5] "current_1976.2005_bioclim.06" "current_1976.2005_bioclim.12"
```

```
preds_current2 <- setNames(preds_current,c("AWAP","wetland_connectivity","EVI","Bioclim05","Bioclim06",
names(preds_current2)
```

```
## [1] "AWAP" "wetland_connectivity" "EVI"
## [4] "Bioclim05" "Bioclim06" "Bioclim12"
```

```
rm(preds_current)
```

```
#writeRaster(Bioclim12,"predictors/Bioclim12.asc")
```

## step2c

this is a soil wetness index from 2013 downloaded from the Eco-Commons site

[https://api.data-ingester.app.ecocommons.org.au/api/data/69b045b3-29d1-5068-a5e6-f0783c68c1b8/download/awap\\_\\_run26j\\_\\_2013\\_\\_ann-TempMin.tif](https://api.data-ingester.app.ecocommons.org.au/api/data/69b045b3-29d1-5068-a5e6-f0783c68c1b8/download/awap__run26j__2013__ann-TempMin.tif)

Simply enter the above url, download the file, and load it from the directory path you saved it

*#-* ideally you would match the time the layer was made to the time *#* your occurrence data was collected, and you might average over the last ten years, but this is just an example of the kinds of data you can use *#* the EcoCommons point-and-click environment does this for you when in the SDM workflow *#*you can select to make all the variables the finest or coarsest resolution

## Environmental data

```
library(raster)
```

```
## Warning: package 'raster' was built under R version 4.1.2
```

```
## Loading required package: sp
```

```
library(rgdal)
```

```
## Warning: package 'rgdal' was built under R version 4.1.2
```

```
## Please note that rgdal will be retired by the end of 2023,  
## plan transition to sf/stars/terra functions using GDAL and PROJ  
## at your earliest convenience.  
##
```

```
## rgdal: version: 1.5-29, (SVN revision 1165M)
```

```
## Geospatial Data Abstraction Library extensions to R successfully loaded
```

```
## Loaded GDAL runtime: GDAL 3.4.0, released 2021/11/04
```

```
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/gdal
```

```
## GDAL binary built with GEOS: TRUE
```

```
## Loaded PROJ runtime: Rel. 8.1.1, September 1st, 2021, [PJ_VERSION: 811]
```

```
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.4-6
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
## Overwritten PROJ_LIB was /Library/Frameworks/R.framework/Versions/4.1/Resources/library/rgdal/proj
```

```
direct<- "/Users/s2992269/Documents/Use_cases"
folder <- "/SDM_in_R"

#this sets your working director for all subsequent chunks of code in your R Markdown script
knitr::opts_knit$set(root.dir = paste0(direct,folder))

# if you are not working R Markdown, simply use this instead
# setwd(paste0(direct,folder))
```

Write rasters if needed

```
writeRaster(wetland_connectivity,"predictors/wetland_connectivity.asc")
```

```
writeRaster(wetland_connectivity,"predictors_future/wetland_connectivity
```

data comes in many formats, and uploading often requires different methods

NetSDF files are increasingly common, below is a script to bring in that file type

```
#require(sf) #require(ncdf4) #require(rasterVis) #require(raster)
```

```
MXtemp <- brick("Terraclim_EY_NSW.nc", varname="tmax")
# this NetCDF file includes many
```

```
#days of maximum temperature data with that subset of data having the varname = "tmax" # MX-
temp_mean <- mean(MXtemp) # if we just want one layer which is the mean of those daily totals
```

we will now download a vegetation greenness index from EcoCommons

```
#https://api.data-ingester.app.ecocommons.org.au/api/data/34ab5ea6-650f-503f-9446-88d0ae9effe1/
download/ndlc-2004-250m_trend-evi-mean.tif
```

download the file directly, but here we upload the file to save time

```
EVI1<- raster("https://api.data-ingester.app.ecocommons.org.
au/api/data/34ab5ea6-650f-503f-9446-88d0ae9effe1/download/
ndlc-2004-250m__trend-evi-mean.tif")
```

```
getwd()
```

```
## [1] "/Users/s2992269/Documents/Use_cases/SDM_in_R"
```

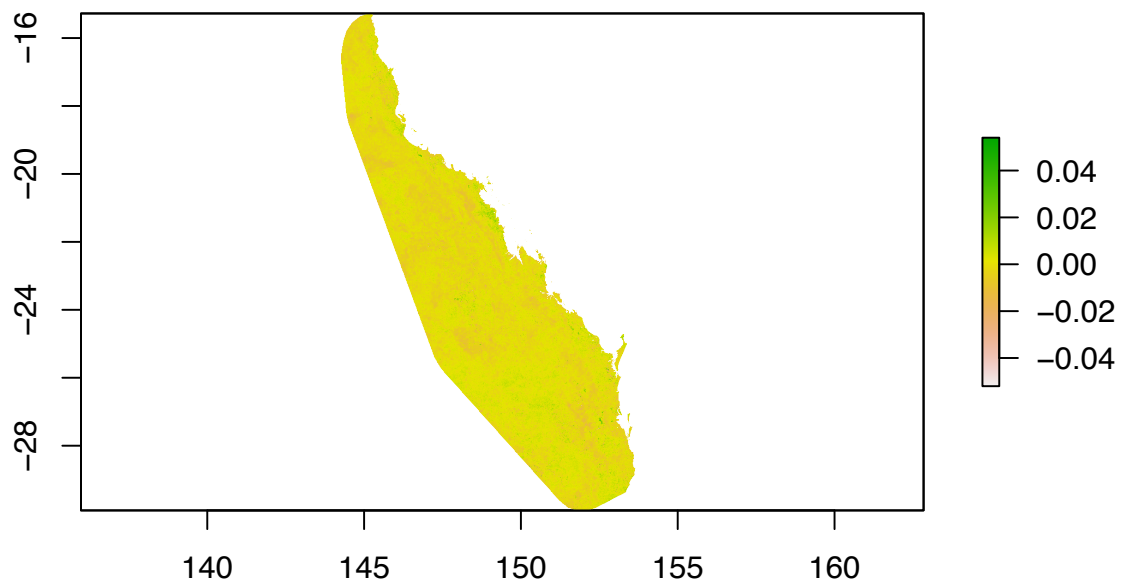
```
EVI1 <- raster::raster("Raw_data/EVI.asc")
raster::crs(EVI1)<-"+proj=longlat +datum=WGS84 +no_defs"
base <- raster::raster("data/base_LiPe.asc")
raster::crs(base)<-"+proj=longlat +datum=WGS84 +no_defs"
AWAP <- raster::raster("Raw_data/AWAP.asc")
raster::crs(AWAP)<-"+proj=longlat +datum=WGS84 +no_defs"
wetland_connectivity <- raster::raster("Raw_data/AWAP.asc")
raster::crs(wetland_connectivity)<-"+proj=longlat +datum=WGS84 +no_defs"
```

process it as previous layers

```
EVI2 <-projectRaster(EVI1,base)
```

```
## Warning in projectRaster(EVI1, base): input and ouput crs are the same
```

```
EVI3<- mask(EVI2,base)
plot(EVI3)
```



```
EVI <- EVI3
```

write rasters

```
writeRaster(EVI,"predictors/EVI.asc")
```

```
writeRaster(EVI,"predictors_future/EVI.asc")
```

**Here we just show how to bring in current and future data from EcoCommons**

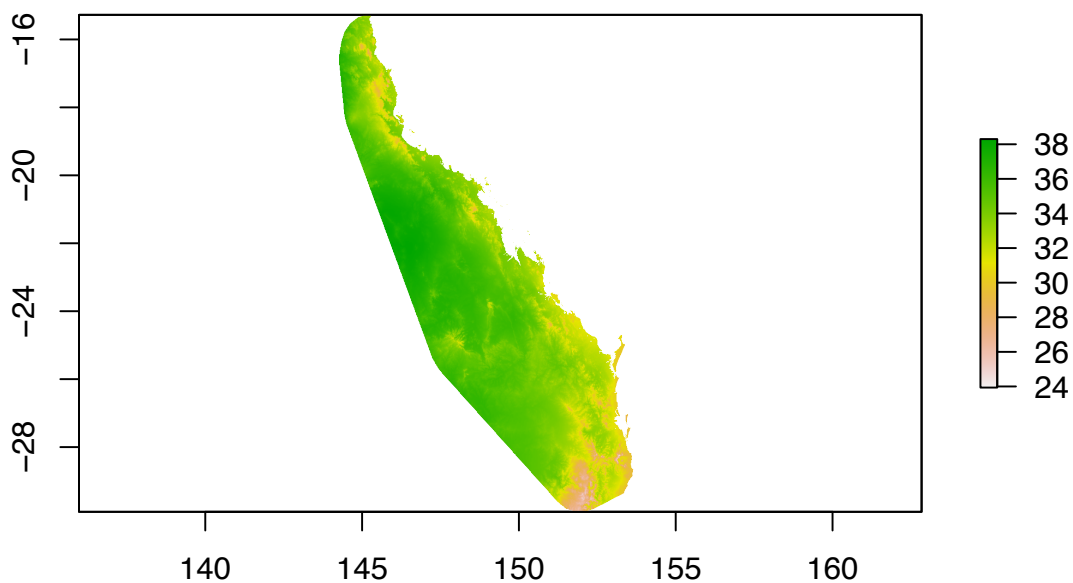
```
#Repeat with future climate data, from EcoCommons # Australia, Climate Projection, SRESA1B based
on INM-CM30, 30 arcsec (~1km) - 2085
```

**after looking at correlations between bioclim variables this one was dropped**

```
#Bioclim01_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/4e5534eb-c743-
52af-a9c3-d137df8ba8c1/download/SRESA1B_inm-cm30_2085_bioclim-01.tif") #Bioclim01_2 <-
```

```
projectRaster(Bioclim01_1,base) #Bioclim01_3 <- mask(Bioclim01_2 ,base) #plot(Bioclim01_3)
#Bioclim01 <- Bioclim01_3 #writeRaster(Bioclim01,"predictors_future/Bioclim01.asc")
```

```
Bioclim05_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/34fe63e3-4152-5406-96f7")
Bioclim05_2 <-projectRaster(Bioclim05_1,base)
Bioclim05_3 <- mask(Bioclim05_2,base)
plot(Bioclim05_3)
```



```
Bioclim05 <- Bioclim05_3
```

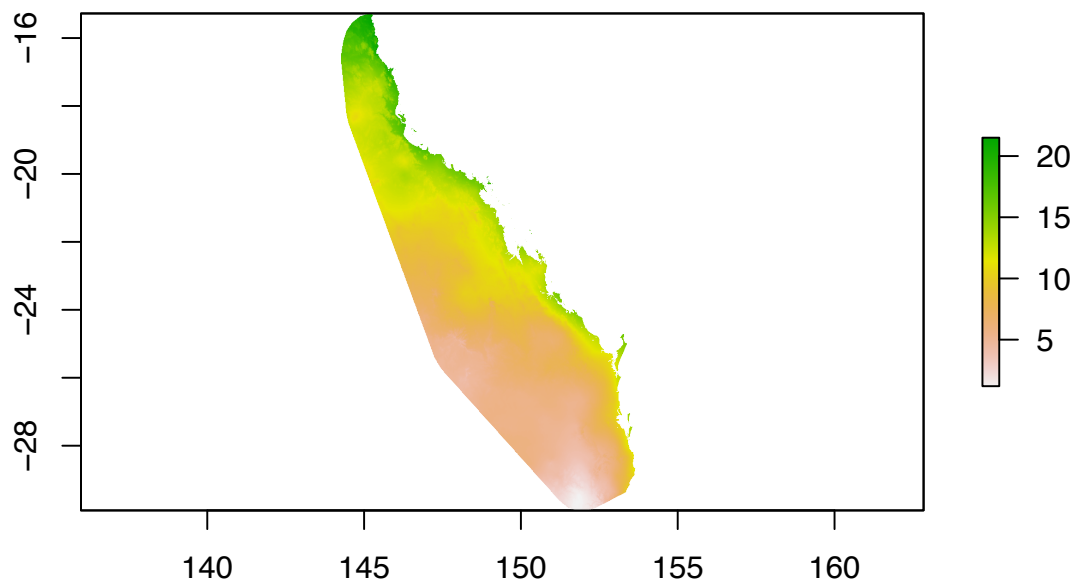
**writeRaster(Bioclim05,"predictors\_future/Bioclim05.asc")**

```
Bioclim06_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/36cf3700-d238-511c-bb8b")
```

```
## Error in .local(.Object, ...) : Empty reply from server
```

```
## Warning in .rasterFromGDAL(x, band = band, objecttype, ...): Could not read RAT
## or Category names
```

```
Bioclim06_2 <-projectRaster(Bioclim06_1,base)
Bioclim06_3 <- mask(Bioclim06_2,base)
plot(Bioclim06_3)
```

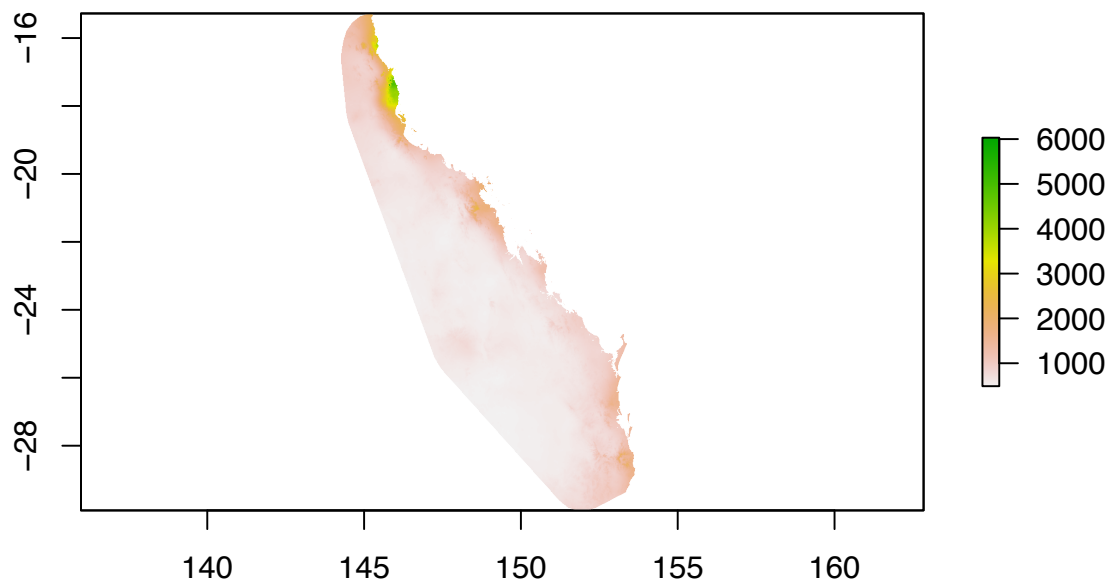


```
Bioclim06 <- Bioclim06_3
```

```
writeRaster(Bioclim06,"predictors_future/Bioclim06.asc")
```

```
Bioclim12_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/449cee85-5f6a-5b25-aca8  
Bioclim12_2 <-projectRaster(Bioclim12_1,base)  
Bioclim12_3 <- mask(Bioclim12_2,base)  
plot(Bioclim12_3)
```





```
Bioclim12 <- Bioclim12_3
```

```
writeRaster(Bioclim12,"predictors_future/Bioclim12.asc")
```

```
rm(Bioclim05_1,Bioclim05_2,Bioclim05_3,Bioclim06_1,Bioclim06_2,Bioclim06_3,Bioclim12_1,Bioclim12_2,Bioclim12_3)
```

**after looking at correlations between bioclim variables this one was dropped**

```
#Bioclim14_1 <- raster("https://api.data-ingester.app.ecocommons.org.au/api/data/178ee1bc-0aa1-58a0-abaf-ff5e19fcf363/download/SRESA1B_inm-cm30_2085_bioclim-14.tif") #Bioclim14_2 <- projectRaster(Bioclim14_1,base) #Bioclim14_3 <- mask(Bioclim14_2 ,base) #plot(Bioclim14_3) #Bioclim14 <- Bioclim14_3 #writeRaster(Bioclim14,"predictors_future/Bioclim14.asc")
```

#again just be sure to run this in order so you are not getting current bioclim data mixed with the future data which has the same name

```
preds_future <- stack(AWAP,wetland_connectivity,EVI,Bioclim05,Bioclim06,Bioclim12)
names(preds_future)
```

```
## [1] "AWAP.1"
```

```
"AWAP.2"
```

```
## [3] "EVI" "SRESA1B_inm.cm30_2085_bioclim.05"
## [5] "SRESA1B_inm.cm30_2085_bioclim.06" "SRESA1B_inm.cm30_2085_bioclim.12"
```

```
preds_future2 <- setNames(preds_future, c("AWAP", "wetland_connectivity", "EVI", "Bioclim05", "Bioclim06", "Bioclim12"))
names(preds_future2)
```

```
## [1] "AWAP" "wetland_connectivity" "EVI"
## [4] "Bioclim05" "Bioclim06" "Bioclim12"
```

```
preds_future2
```

```
## class      : RasterStack
## dimensions : 6228, 4366, 27191448, 6 (nrow, ncol, ncell, nlayers)
## resolution : 0.002349, 0.002349 (x, y)
## extent      : 144.279, 154.5347, -29.90543, -15.27585 (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs
## names       : AWAP, wetland_connectivity, EVI, Bioclim05, Bioclim06, Bioclim12
## min values  : ?, ?, -0.07367065, 23.88057590, 1.20000005, 488.00000005
## max values  : ?, ?, 6.311681e-02, 3.830000e+01, 2.157484e+01, 6.795644e+01
```

```
rm(preds_future)
```

here we will show you how to take a mean of many months of data if you have many months of data in one folder (you may wa

perhaps you are exploring what a suitable niche looks like during a drought

You may then want climate like averages like precipitation, or average temperature during the drought months

one way to do this if you have separate rasters fir each month in one folder

```
#raster_list <- list.files(pattern='tif$', all.files=TRUE) # if there are 30 files of tifs from January, this
#creates a list of those files #all_chl <- stack(raster_list) # this reads each of those files in, and makes them
#into a raster stack #mean_chl <- mean(all_chl) # and this produces one raster where each cell value is the
#mean value of all the rasters that were in the that folder
```

NETCDF files also are a common way for large volumes of data to be

```
#this is an example loop using Terraclimate data # read NetCDF data file obtained from "TerraClimate"
```

import netCDF file - data from TerraClimate python download

```
require(sf)
```

```
require(ncdf4)
```

```
require(rasterVis)
```

```
require(raster)
```

download the file here: <https://drive.google.com/file/d/1d4aCrdwjWRgENFY/view?usp=sharing>

```
setwd("~/Documents/Use_cases/EY_frogs/data") # this is the  
directory where this NETCDF file is stored
```

```
MXtemp <- brick("Terraclim_EY_NSW.nc", varname="tmax")
```

```
Mxtemp_mean <- mean(MXtemp)
```

```
plot(Mxtemp_mean)
```

```
writeRaster(Mxtemp_mean, "MXtemp_TERRA_Sydney_region.asc",  
overwrite=TRUE)
```

```
Rain <- brick("Terraclim_EY_NSW.nc", varname="ppt")
```

```
Rain_mean <- mean(Rain)
```

```
plot(Rain_mean)
```

```
writeRaster(Rain_mean, "Rain_TERRA_Sydney_region.asc",  
overwrite=TRUE)
```

```
MNtemp <- brick("Terraclim_EY_NSW.nc", varname="tmin")
```

```
MNtemp_mean <- mean(MNtemp)
```

```
plot(MNtemp_mean)
```

```
writeRaster(MNtemp_mean, "MNtemp_TERRA_Sydney_region.asc",  
overwrite=TRUE)
```

```
Soil <- brick("Terraclim_EY_NSW.nc", varname="soil")
```