NetCDF_Attempt

James Waterford

2023-09-10

NetCdf Installation

NetCdf is a file format widely used for acessing Earth Science Systems data. While the most common applications it is used with are **Java** and **C++**, Packages exist in both R Studio and Python.

R Install

To install in R, you will need both the `ncd4 package`, and the `NetCDF libraries`.

THIS STEP IS NOT REQUIRED AS `ncd4` INCLUDES ALL LIBRARIES

The NetCDF library can be found on the UCAR website - Download the `netcdf-c-4.9.2.tar.gz` into a location you will remember - From the R-Studio dropdown menu, select: Tools > Install Packages

```
# install.packages("ncdf4")
library(ncdf4)
```

File Connection

Use `nc_open` to read an existing netCDF file. nc_open(filename, write=FALSE)

For the dataset we are working with today, download from USAP. Additionally, the author of the USAP file suggests installing Panoply for the explanation of variables.

```
gen_atmos_file <- ("LENO2000H_10_yr_avg.nc")
gen_atmos <- nc_open(gen_atmos_file)
#print(gen_atmos)

gen_surf_file <- ("LENO2000H_LSX_10_yr_avg.nc")
gen_surf <- nc_open(gen_surf_file)
# print(gen_surf)
## Printing this line creates 6 pages in PDF format ##
```

In our case, we have 63 variables. Trying to determine all 63 named variables will prove to be a challenge. Use Panolopy to gain some insight.

Next Steps

** Create a Data Frame **

Now that we can access the data, we ned to extract the longitude and latitude values of interest from this full data set. ## Current Objectives: ##

- Gather data temperature around the Drake passage
- There are twelve slices for each month
- Extract temperatures over time -Calculate Mean Annual

Define Long & Lat

```
lon <- ncvar_get(gen_surf,"lon")
lon
```

```
##   [1]   1   3   5   7   9  11  13  15  17  19  21  23  25  27  29  31  33  35
##  [19]  37  39  41  43  45  47  49  51  53  55  57  59  61  63  65  67  69  71
##  [37]  73  75  77  79  81  83  85  87  89  91  93  95  97  99 101 103 105 107
##  [55] 109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143
##  [73] 145 147 149 151 153 155 157 159 161 163 165 167 169 171 173 175 177 179
##  [91] 181 183 185 187 189 191 193 195 197 199 201 203 205 207 209 211 213 215
## [109] 217 219 221 223 225 227 229 231 233 235 237 239 241 243 245 247 249 251
## [127] 253 255 257 259 261 263 265 267 269 271 273 275 277 279 281 283 285 287
## [145] 289 291 293 295 297 299 301 303 305 307 309 311 313 315 317 319 321 323
## [163] 325 327 329 331 333 335 337 339 341 343 345 347 349 351 353 355 357 359
```

```
lat <- ncvar_get(gen_surf,"lat")
lat
```

```
##  [1] -89 -87 -85 -83 -81 -79 -77 -75 -73 -71 -69 -67 -65 -63 -61 -59 -57 -55 -53
## [20] -51 -49 -47 -45 -43 -41 -39 -37 -35 -33 -31 -29 -27 -25 -23 -21 -19 -17 -15
## [39] -13 -11  -9  -7  -5  -3  -1   1   3   5   7   9  11  13  15  17  19  21  23
## [58]  25  27  29  31  33  35  37  39  41  43  45  47  49  51  53  55  57  59  61
## [77]  63  65  67  69  71  73  75  77  79  81  83  85  87  89
```

Create Temp Slice

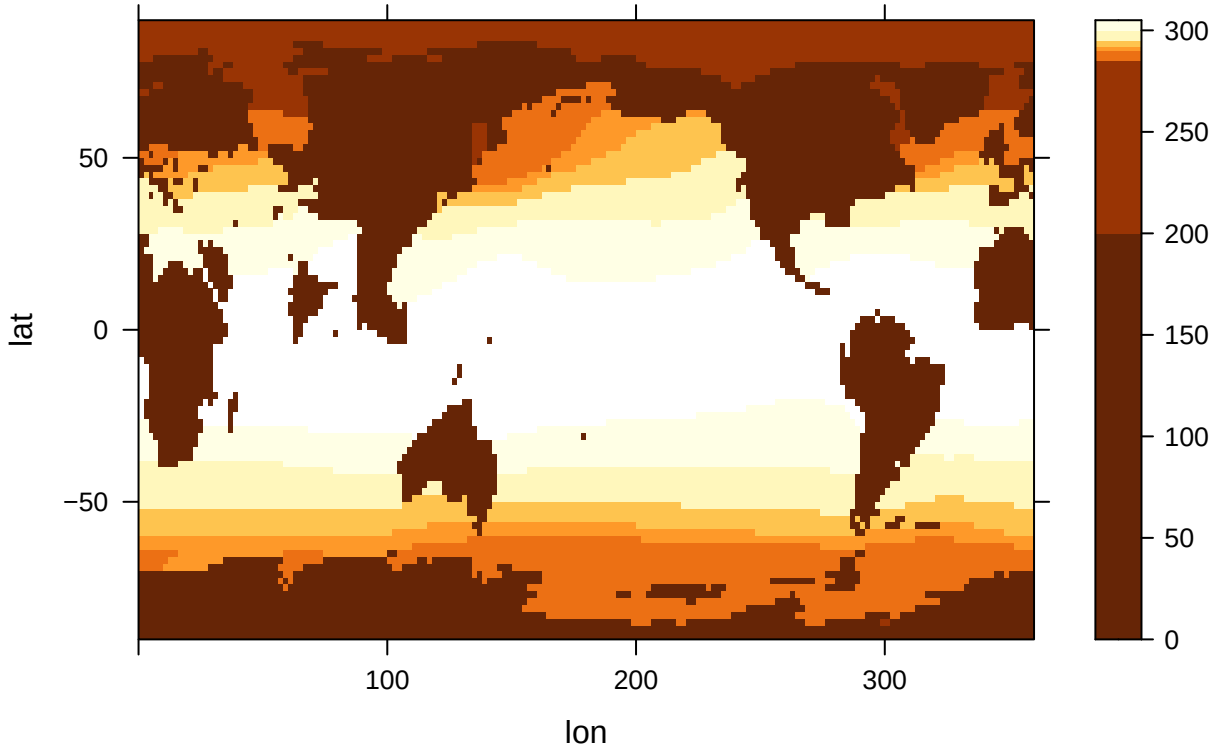
```
 ## Get Ocean Temperature from Surface Data
dname <- "TOCEAN"
tmp_array <- ncvar_get(gen_surf,dname)

# get a single slice or layer (January)
m <- 1
tmp_slice <- tmp_array[,,m]
```

Map and Map Libraries

```
library(lattice)
library(RColorBrewer)
```

This will map all temperature varation across the full dataset.

```
grid <- expand.grid(lon=lon, lat=lat)
cutpts <- c(0,200,285,290,292,295,300,305)
levelplot(tmp_slice ~ lon * lat, data=grid, at=cutpts, cuts=9, pretty=T,
  col.regions=(rev(brewer.pal(9,"YlOrBr"))))
```



Now we need to gather the relevant points into a data frame.

```
# create dataframe -- reshape data


## Create Full data set from map & temperature
lonlat <- as.matrix(expand.grid(lon,lat))
tmp_vec <- as.vector(tmp_slice)

tmp_df01 <- data.frame(cbind(lonlat,tmp_vec))
names(tmp_df01) <- c("lon","lat",paste(dname,as.character(m), sep="_"))


## Subset to Drake Passage
## [lon -> 240.00 : 320.00   | lat -> -89.00 : -15.00]
```

```
tmp_df01 <- tmp_df01[tmp_df01$lat <= -15 & tmp_df01$lon >= 240 & tmp_df01$lon <= 320,]
tmp_df01[tmp_df01 == 0] <- NA
head(na.omit(tmp_df01), 10)
```

```
##     lon lat TOCEAN_1
## 495 269 -85 287.2780
## 496 271 -85 287.2819
## 497 273 -85 287.2865
## 498 275 -85 287.2897
## 510 299 -85 283.2515
## 511 301 -85 283.2520
## 673 265 -83 287.2415
## 674 267 -83 287.2507
## 675 269 -83 287.2654
## 676 271 -83 287.2791
```

Using functions to streamline this process.

```
lonlat <- as.matrix(expand.grid(lon,lat))
tmp_df <- data.frame(cbind(lonlat))
names(tmp_df) <- c("lon","lat")

testr <- function(tmp_df){
  for (i in 1:12) {
    m <- i
    mon <- month.name[i]
    tmp_slice <- tmp_array[,,m]
    tmp_vec <- as.vector(tmp_slice)
    tmp_df[[mon]] <- tmp_vec
  }
  tmp_df ->> DP_tmp
}
testr(tmp_df)
DP_tmp <- DP_tmp[DP_tmp$lat <= -15 & DP_tmp$lon >= 240 & DP_tmp$lon <= 320,]
DP_tmp[DP_tmp == 0] <- NA

head(na.omit(DP_tmp))
```

```
##     lon lat  January February    March    April      May     June     July
## 495 269 -85 287.2780 288.8372 288.4184 287.0331 285.5338 284.1009 282.6804
## 496 271 -85 287.2819 288.8391 288.4197 287.0359 285.5376 284.1049 282.6839
## 497 273 -85 287.2865 288.8410 288.4208 287.0386 285.5417 284.1092 282.6877
## 498 275 -85 287.2897 288.8421 288.4212 287.0401 285.5442 284.1119 282.6902
## 510 299 -85 283.2515 285.1739 285.3073 284.0804 282.5319 280.8941 279.1749
## 511 301 -85 283.2520 285.1741 285.3071 284.0803 282.5318 280.8938 279.1746
##       August September  October November December
## 495 281.2589  280.0689 279.7862 281.2623 284.1310
## 496 281.2631  280.0733 279.7910 281.2665 284.1363
## 497 281.2676  280.0783 279.7967 281.2716 284.1425
## 498 281.2705  280.0818 279.8008 281.2753 284.1469
```

```
## 510 277.3198  275.8362 275.4545 276.8772 279.8409
## 511 277.3195  275.8361 275.4546 276.8777 279.8417
```

```
write.csv(DP_tmp, "DrakePassageTemps.csv", row.names=FALSE)
```

Figures

```
months <- DP_tmp[-c(1,2)]
boxplot(months, border = "steelblue4")
```