

# Satellite Data Exploration

Eleanor (Ella) Crotty

## Contents

<b>Setup</b>	<b>1</b>
<b>Import data</b>	<b>1</b>
Data sources . . . . .	1
<b>Function</b>	<b>3</b>

## Setup

```
# Warnings and startup messages suppressed
library(tidyverse)
library(patchwork)
library(scales)
library(ggrepel)
library(readxl)
library(here)
```

## Import data

### Data sources

#### Temperature

- **Multi-scale Ultra-high Resolution (MUR) SST Analysis fv04.1, Global, 0.01°, 2002-present, Daily**
  - Multi-sensor merged SST dataset made by NASA JPL
  - 1 km resolution
  - Multi-Resolution Variational Analysis (MRVA) method for interpolation
  - [Link](#)
  - [Metadata](#)
  - **This seems like the best one to me since it's new and high-resolution**
- **Sea-Surface Temperature, NOAA ACSPO NOAA-20 VIIRS CoastWatch Co-gridded 4km Daily (degrees C)**
  - `cw_pass_type = day/night`
  - Produced by NOAA/NESDIS/Office of Satellite and Product Operations (OSPO) office from Visible and Infrared Imager/Radiometer Suite (VIIRS) sensor
  - [Link](#)
  - [Metadata](#)
  - **This seems good also, and is probably day and night - between this and the JPL one I can't tell if they both have day and night.**

- *AVHRR Pathfinder Version 5.3 L3-Collated (L3C) SST, Global, 0.0417°, 1981-present, Daytime (1 Day Composite)*
  - These data were provided by GHRSSST and the NOAA National Centers for Environmental Information (NCEI).
  - Sea surface temperature (SST) data produced as part of the AVHRR Pathfinder SST Project. These data were created using Version 5.3 of the Pathfinder algorithm and the file is nearly but not completely compliant with the Global High-Resolution Sea Surface Temperature (GHRSSST) Data Specifications V2.0 (GDS2).
  - [Link](#)
  - [Metadata](#)
  - *Day only*
- *Sea-Surface Temperature, NOAA Geo-polar Blended Analysis Night Only, GHRSSST, Near Real-Time, Global 5km, 2002-Present, Daily (degree C)*
  - Night only
  - Combines multi-satellite retrievals of sea surface temperature into a single analysis of SST. This analysis includes only nighttime data.
  - NOAA NESDIS CoastWatch
  - [Link](#)
  - [Metadata](#)
  - *Probably not going to use it, night only*

## Chlorophyll-a

- *Chlorophyll-a, Aqua MODIS, NPP, L3SMI, Global, 4km, Science Quality, 2003-present (1 Day Composite)*
  - Mean chlorophyll-a,  $\text{mg}/\text{m}^3$
  - This dataset has Level 3, Standard Mapped Image, 4km, chlorophyll-a concentration data from NASA's Aqua Spacecraft. Measurements are gathered by the Moderate Resolution Imaging Spectroradiometer (MODIS) carried aboard the spacecraft. This is Science Quality data. This is the August 2015 version of this dataset. *Is it really?*
  - NOAA NMFS SWFSC ERD
  - [Link](#)
  - [Metadata](#)
- **Chlorophyll, NOAA S-NPP VIIRS, Science Quality, Global 4km, Level 3, 2012-present, Daily**
  - NOAA CoastWatch
  - Level-3 Standard Mapped Image (SMI), Global, 4km, Chlorophyll, Daily composite data from the Visible and Infrared Imager/Radiometer Suite (VIIRS).
  - These are science quality data with a 15-day latency.
  - The data, processed by the NOAA/STAR Ocean Color Team, are produced from MSL12 v1.2 using OC-SDR v04 and were release by CoastWatch as of 2017-08-07.
  - VIIRS is a multi-disciplinary instrument that flies on the Suomi-NPP (National Polar-orbiting Partnership) satellite and is the successor to MODIS. Suomi-NPP is the initial spacecraft in the JPSS (Joint Polar Satellite Systems) series of spacecraft. JPSS is our Nation's next generation polar-orbiting operational environmental satellite system. JPSS is a collaborative program between NOAA and its acquisition agent NASA.
  - [Link](#)
  - [Metadata](#)
  - Probably going to use it, it's the successor to MODIS and I don't need pre-2012 data

## My Filters

- 2021-01-01 to 2023-12-31
- 47.12 to 48.51 N
- -124.18 to -125.68 E

## Function

This function comes from my work last summer, where I used it to clean ERDDAP sea surface temperature data. I don't think this is going to work fantastically, but I am going to use it as a starting point.

```
## Function to clean a dataset from the NOAA ERDDAP data repository
ERDDAP_Import <- function(Data, vars = NULL, dropNAvars = NULL, avname = "ERDDAPavg"){
  # Data - a data frame imported from the ERDDAP website
  # vars - a vector of column names you want to keep (time, lat, and lon will be kept
  ↪ automatically)
  # dropNAvars - a vector of column names you want to have no NA values after processing

  # Check that Data is a dataframe
  if (class(Data) != "data.frame") {
    stop("ERROR: Data must be an object of type data.frame")
  }
  # Check that dropNAvars is only one argument
  if(length(dropNAvars) > 1) {
    stop("ERROR: dropNAvars only accepts one argument at this time")
  }

  Name <- deparse(substitute(Data))
  print(Name)
  print(head(Data[,vars]))
  Data_clean <- filter(Data, Data[,dropNAvars] != "NaN") # I think this only works for
  ↪ one dropNAvar at this time
  Data_clean <- Data_clean[,c("time", "latitude", "longitude", vars)]
  print(Data_clean)

  strcheck <- c(letters, "_", LETTERS)

  newvars <- c()
  # Add units to vars so that later vars calls work
  for (i in 1:length(vars)) {
    var <- vars[i]
    #print(var)
    unit <- (Data[1,var])
    newvar <- paste(var, sep="_", unit)
    newvar <- gsub(" ", "", newvar)
    print(paste("newvar", sep = "=", newvar)) # Remove spaces
    newvars <- c(newvars, newvar)
  }
  print(newvars)

  newdropNAvars <- c()
  # Add units to vars so that later vars calls work
  for (i in 1:length(dropNAvars)) {
    var <- dropNAvars[i]
    #print(var)
    unit <- (Data[1,var])
    newvar <- paste(var, sep="_", unit)
    newvar <- gsub(" ", "", newvar) # Remove spaces
    #print(newvar)
    newdropNAvars <- c(newdropNAvars, newvar)
  }
}
```

```

}
print(newdropNAvars)

if (sum(!(grepl(pattern = paste(strcheck, collapse="|"), x = Data[1,]))) == 0) { # If
  ↪ units row is still present
  # (if all of the values in the first row contain letters and/or underscores)
  colnames(Data_clean) <- gsub(" ", "", paste(colnames(Data_clean), sep = "_",
  ↪ Data_clean[1,])) # Add units + remove spaces
  Data_clean <- Data_clean[-c(1),] # Remove first row (units) now that it's in the
  ↪ column names
  print("=Units removed")
  print(head(Data_clean))
} else {
  print("Units already integrated")
}

timename <- "time.UTC" # Time_units

# Clean up the time column so that it can be converted to POSIXct
Data_clean[,timename] <- gsub("T", " ", Data_clean[,timename])
Data_clean[,timename] <- gsub("Z", "", Data_clean[,timename])
# Convert to POSIXct (datetime format)
Data_clean[,timename] <- as.POSIXct(Data_clean[,timename], tryFormats = c("%Y-%m-%d
  ↪ %H:%M:%OS", "%Y-%m-%d"))

for (i in 1:length(newvars)){
  print(newvars[i])
  Data_clean[,newvars[i]] <- as.numeric(Data_clean[,newvars[i]])
}

print(head(Data_clean))
Data_clean$latitude_degrees_north <- as.numeric(Data_clean$latitude_degrees_north)
Data_clean$longitude_degrees_east <- as.numeric(Data_clean$longitude_degrees_east)

# Clean up date column name
names(Data_clean)[names(Data_clean) == "time.UTC"] <- "Date"

print(head(Data_clean))

return(Data_clean)
#names(Data_clean)[names(Data_clean) == "chlorophyll_mgm-3"] <- "chlorophyll_mgm_3" # I
  ↪ think it would be neat to add newnames = oldnames as a parameter, but not right now
}

```