CoastWatch Tutorials on GitHub

NOAA CoastWatch Satellite Course

Viewing and Analyzing Ocean/Coastal Events and Water Quality Using Satellites

Last Update: March 29, 2025

Software Tutorials on GitHub

https://github.com/coastwatch-training/CoastWatch-Tutorials

Python-setup	updated py setup readme	last week
Tutorial1-basics	made edits from V's review	yesterday
Tutorial2-timeseries-compare-sens...	final edits	18 hours ago
calculate-seaice-extent	updated calculate-seaice-extent-R	yesterday
convert-180+180-to-0-360-longitu...	fixed lat typo problem	last week
create-virtual-buoy-with-satellite-d...	Virtual-buoy-Python	3 hours ago
define-marine-habitat	Vs edits on python modules	53 minutes ago
extract-satellite-data-within-bound...	incorporated V's edit in extract-within-boundary-R	6 hours ago
map-data-with-different-projections	resolved the conflicts in 3 files	last week
matchup-satellite-buoy-data	#77 corrected warning	4 days ago
matchup-satellite-data-to-track-lo...	Vs edits on python modules	53 minutes ago
transform-to-another-map-projecti...	Vs edits on python modules	53 minutes ago

- Each tutorial module is designed to illustrate the process of accessing and manipulating satellite data from the CoastWatch ERDDAP data servers.
- Code is usually available for both R and python
- R folders contain both .md (for internet viewing) and .rmd (for downloading) files

List of Tutorials on GitHub

ERDDAP-basics An introduction to what ERDDAP is and an overview of the different CoastWatch ERDDAP servers. Learn how to visualize and download data from ERDDAP, and how to interpret an ERDDAP url.

Tutorial1-basics Learn to access satellite data from CoastWatch ERDDAP data server and to work with NetCDF files. Visualize sea surface temperature on a map and plot time series data.

Tutorial2-timeseries-compare-sensors Learn common ways to download data from ERDDAP servers to access time-series chlorophyll data from four different satellite datasets and summarize and visualize the data for comparison.

calculate-seaice-extent View sea ice concentration (SIC) data on a map with the polar stereographic projection. Calculate and compare sea ice area/ extent from multi-year SIC datasets.

convert-180+180-to-0-360-longitude Work with datasets with -180° to +180° longitude values in a region that crosses the antimeridian. Convert the coordinates from (-180, +180) to (0, 360) and visualize data on a map.

create-virtual-buoy-with-satellite-data Create a “virtual” buoy using satellite data to fill the gaps in in-situ data collected by a physical buoy. Extract data from a location close to an existing buoy. Clean dataset by removing outliers, and aggregate (resample) to achieve a reduced temporal resolution. Plot time series data.

extract-satellite-data-within-boundary Extract sea surface temperature satellite data for an non-rectangular geographical region from an ERDDAP server using a shapefile, make maps, and plot a timeseries of the seasonal cycle of SST within the boundary.

map-data-with-different-projections Download and examine a polar stereographic projected dataset, plot the data on a projected map. Add animal track data with geographical coordinates onto the projected map.

matchup-satellite-buoy-data Temporally and geospatially subset satellite data to match with buoy data (tabular), run statistical analysis and produce a map of the satellite data with overlaying buoy data.

matchup-satellite-data-to-track-locations Extract satellite data along a set of points defined by longitude, latitude, and time coordinates like that produced by an animal telemetry tag, a ship track, or a glider track.

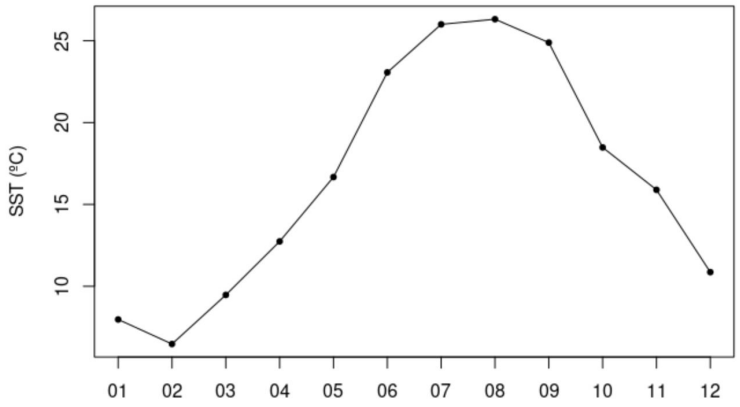
using EPSG code.

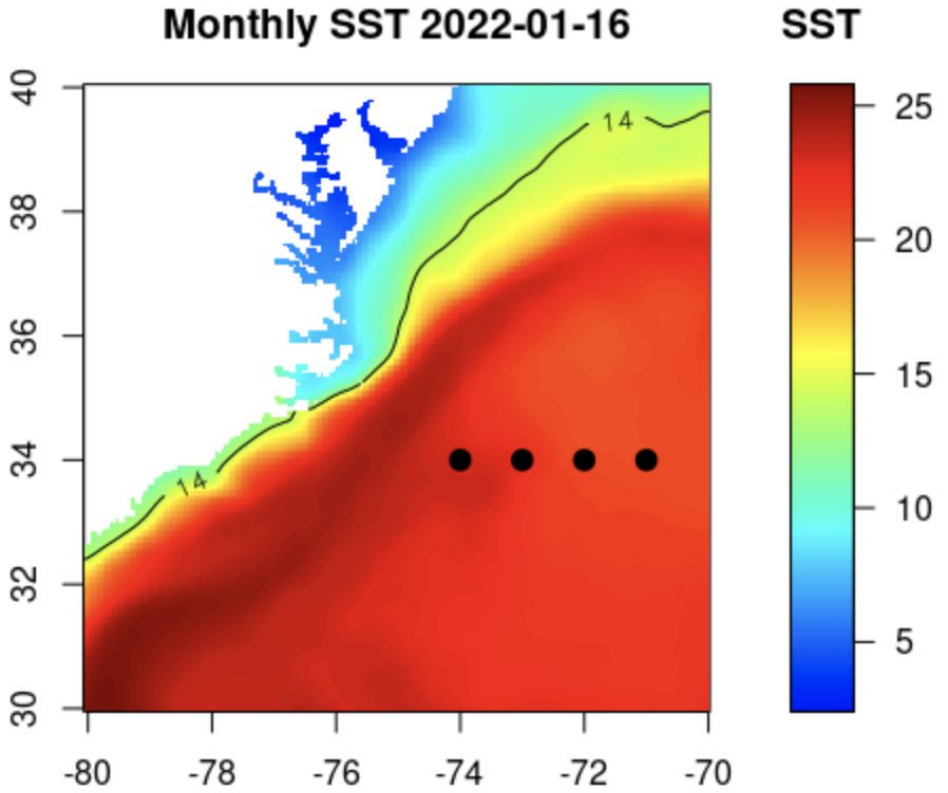
Tutorial1-basics

Learn to access satellite data from CoastWatch ERDDAP data server and to work with NetCDF files. Visualize sea surface temperature on a map and plot time series data.

Tutorial demonstrates:

- Locating a satellite product in ERDDAP
- Manually changing the constraints
- Copying the URL defining the data request
- Downloading the resulting NetCDF file
- Opening and examining the NetCDF file
- Making basic maps and time series plots



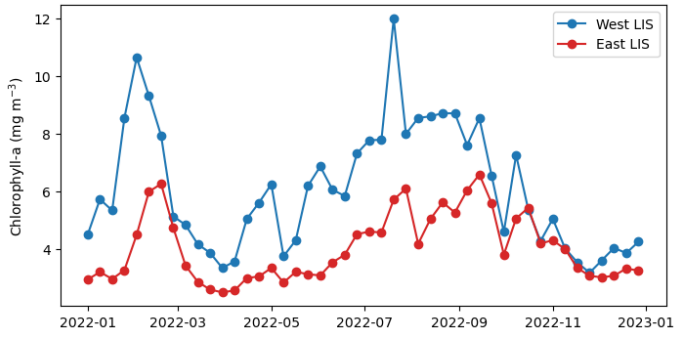


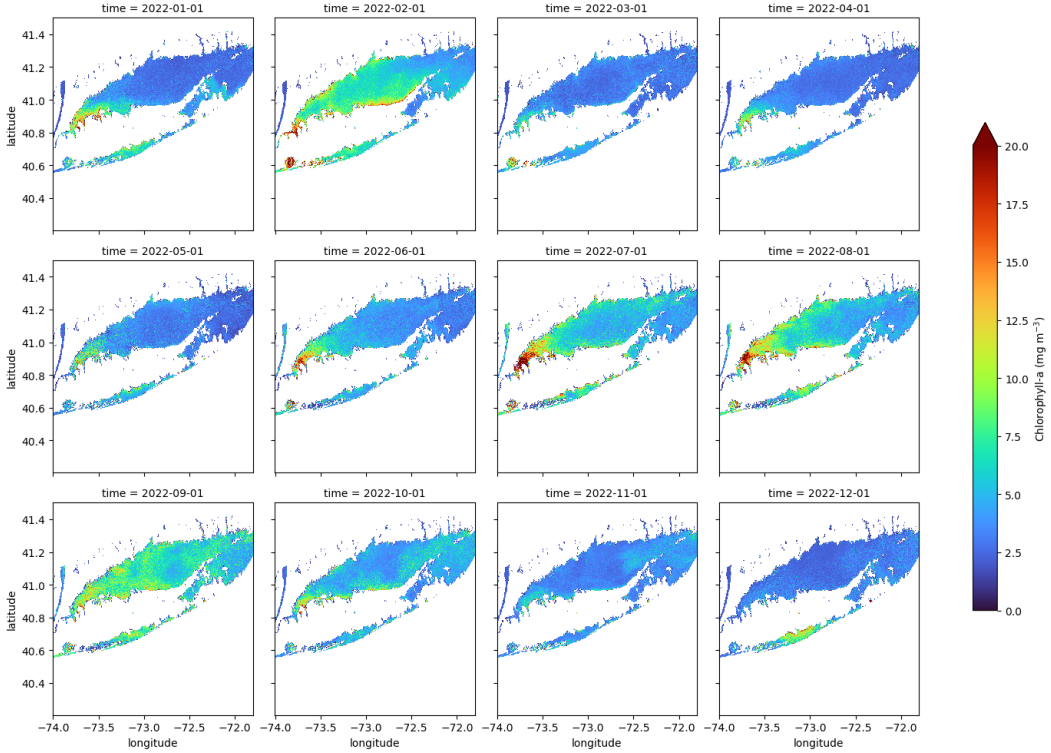
Long Island Sound Chlorophyll-a Dynamics Tutorial

Learn to access Long Island Sound OLCI data from CoastWatch ERDDAP data server. Visualize monthly chlorophyll-a in a multi-panel figure and compare chlorophyll-a time series across regions

Tutorial demonstrates:

- Locating a satellite product in ERDDAP
- Manually changing the constraints
- Copying the URL defining the data request
- Downloading the resulting NetCDF file
- Opening and examining the NetCDF file
- Resampling data to weekly and monthly means
- Making multi-panel maps
- Spatial subsetting and time series plots
-



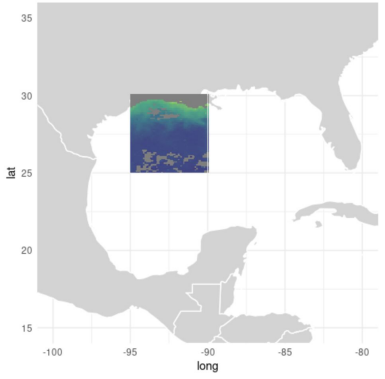


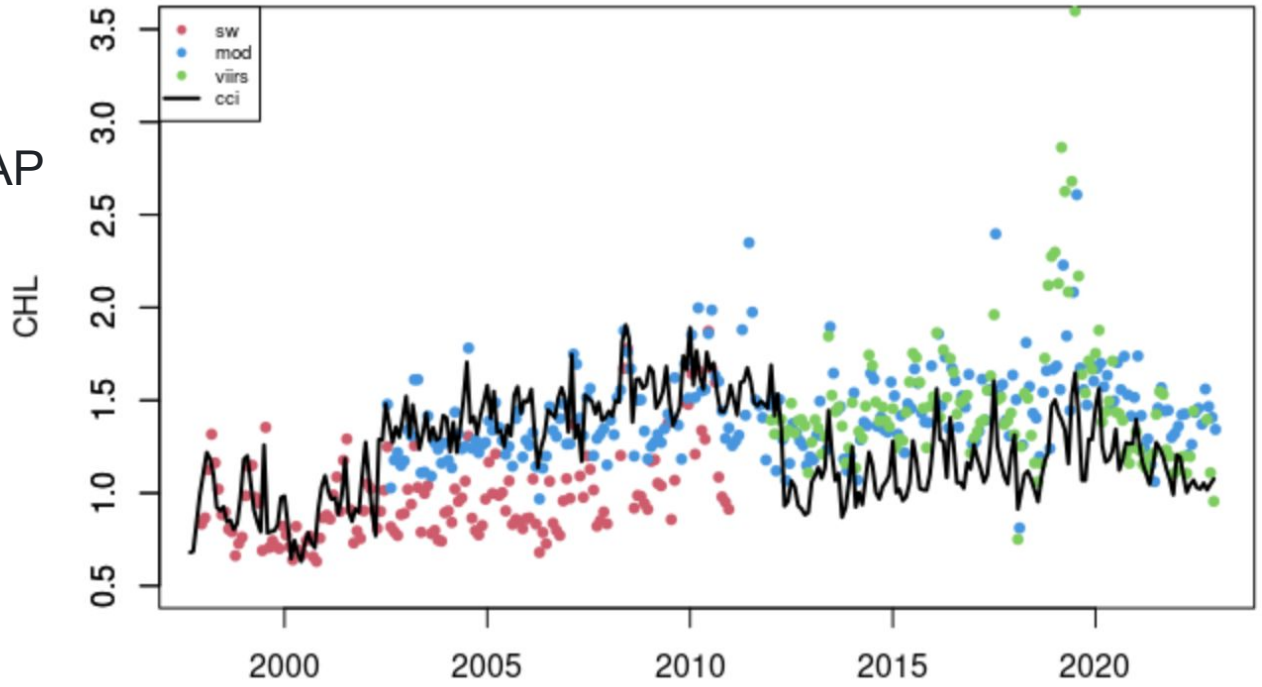
Tutorial2-timeseries-compare-sensors

Learn common ways to download data from ERDDAP servers to access time-series chlorophyll data from four different satellite datasets and summarize and visualize the data for comparison.

Tutorial demonstrates:

- Using **rerddap** to extract data from a rectangular area of the ocean over time
- Retrieve information about a dataset from ERDDAP
- Comparing results from different sensors
- Averaging data spatially
- Producing timeseries plots
- Drawing maps with satellite data



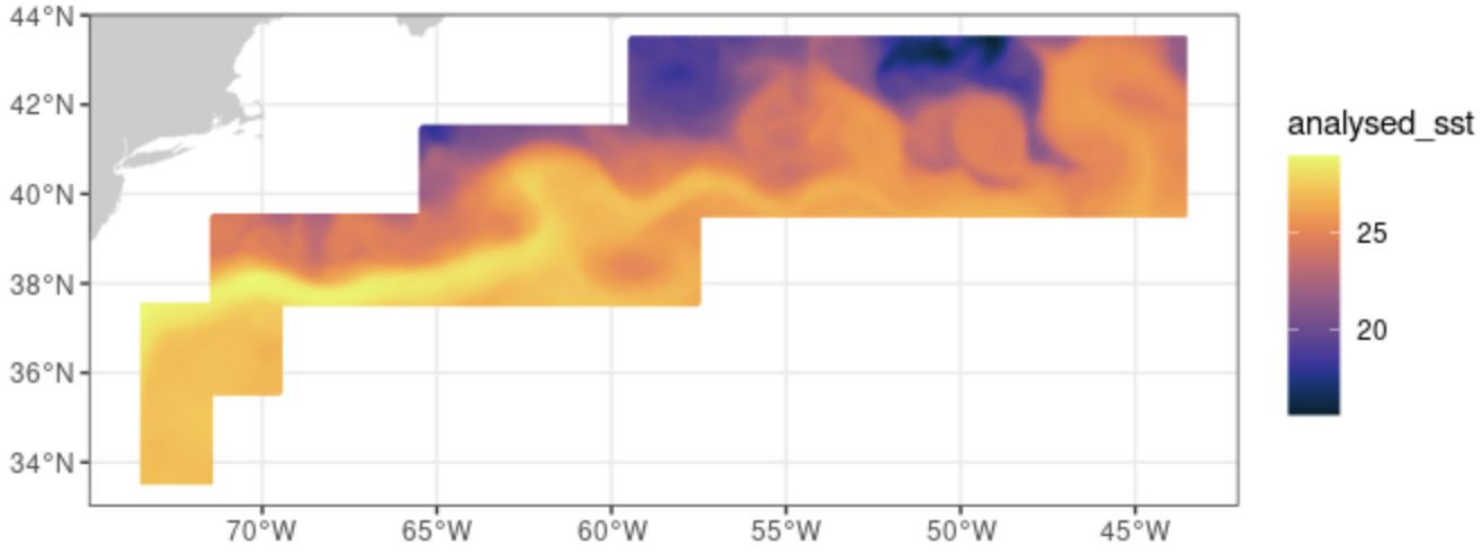


extract-satellite-data-within-boundary

Extract satellite data for an non-rectangular geographical region from ERDDAP using a shapefile, make maps, and plot a timeseries of the seasonal cycle of data within the boundary.

Tutorial demonstrates:

- Using **rerddapXtracto** package to extract data from a polygon
- Downloading data from ERDDAP
- Visualizing data on a map

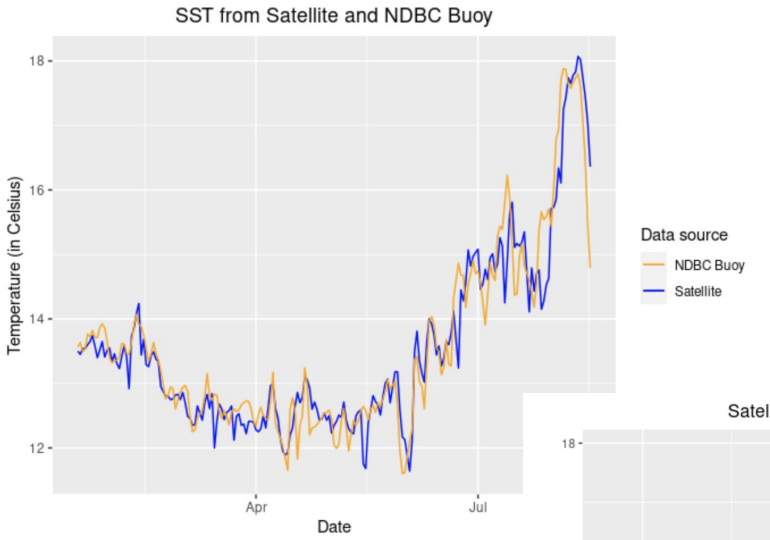


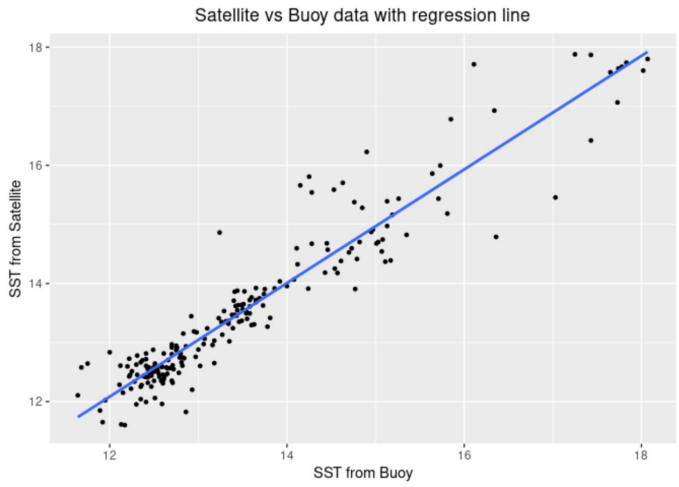
create-virtual-buoy-with-satellite-data

Create a “virtual” buoy using satellite data to fill the gaps in in-situ data collected by a physical buoy. Extract data from a location close to an existing buoy. Clean dataset by removing outliers, and aggregate (resample) to achieve a reduced temporal resolutio¯

Tutorial demonstrates:

- Downloading the satellite and buoy data from ERDDAP
- Visualizing the datasets
- Reshaping the satellite data into a buoy data format
- Resampling buoy data (aggregation) to match satellite data temporal resolution
- Validating the satellite data with the actual buoy data
- Performing a linear regression of satellite vs. buoy data
- Creating a scatter plot of satellite vs. buoy data with the regression line



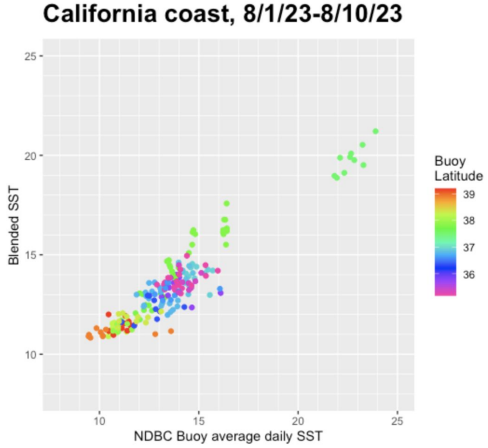


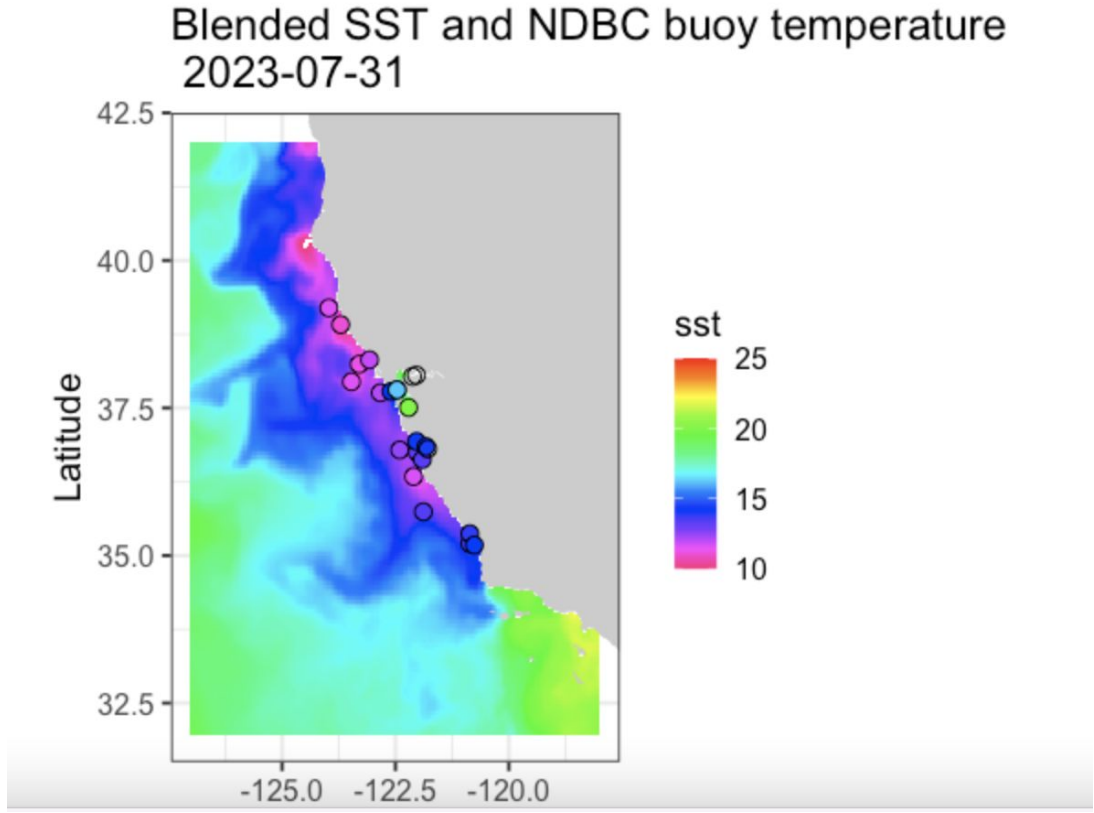
matchup-satellite-buoy-data

Temporally and geospatially subset satellite data to match with buoy data (tabular), run statistical analysis and produce a map of the satellite data with overlaying buoy data.

Tutorial demonstrates:

- Downloading tabular data (buoy data) from ERDDA
- Retrieving information about a dataset from ERDDAP
- Matching satellite data with the buoy data
- Running statistical analysis to compare buoy and satellite data
- Producing satellite maps and overlaying buoy data



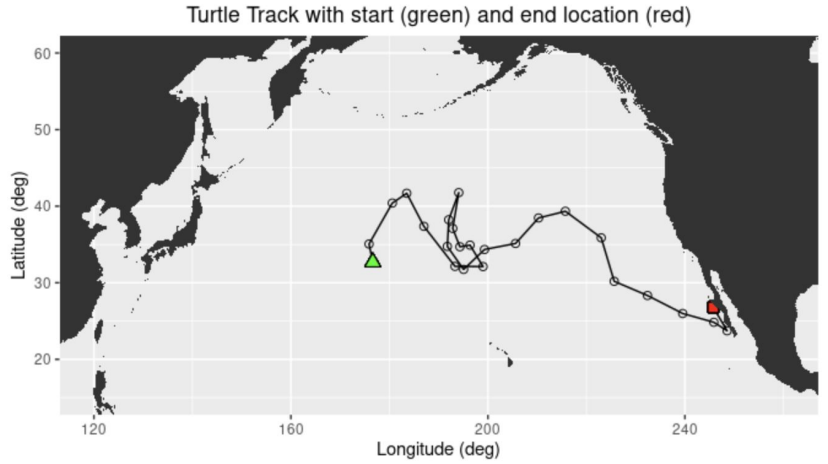


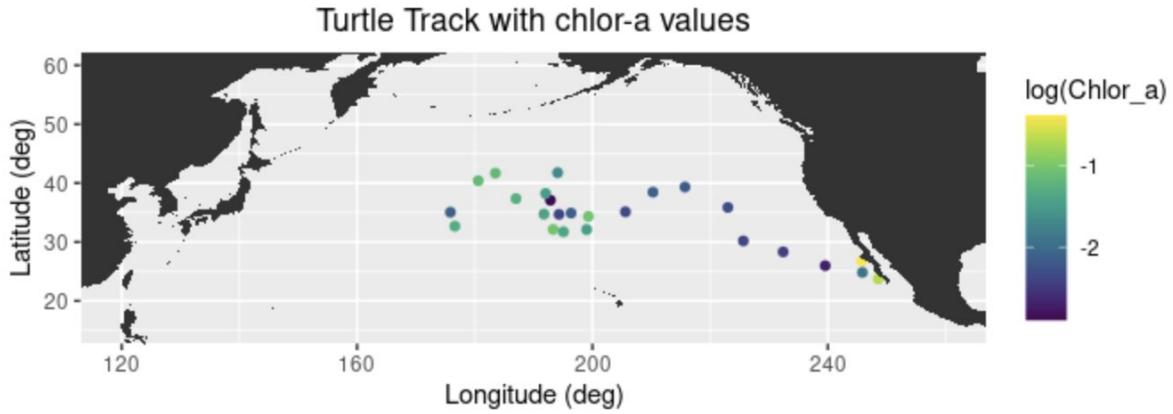
matchup-satellite-data-to-track-locations

Extract satellite data along a set of points defined by longitude, latitude, and time coordinates like that produced by an animal telemetry tag, a ship track, or a glider track.

Tutorial demonstrates:

- Importing track data in csv file to data frame
- Using **rerddapXtracto** package to extract satellite data associated with xyt points
- Plotting the latitude/longitude points onto a map
- Extracting satellite data from an ERDDAP data server along a track
- Plotting the satellite data onto a map



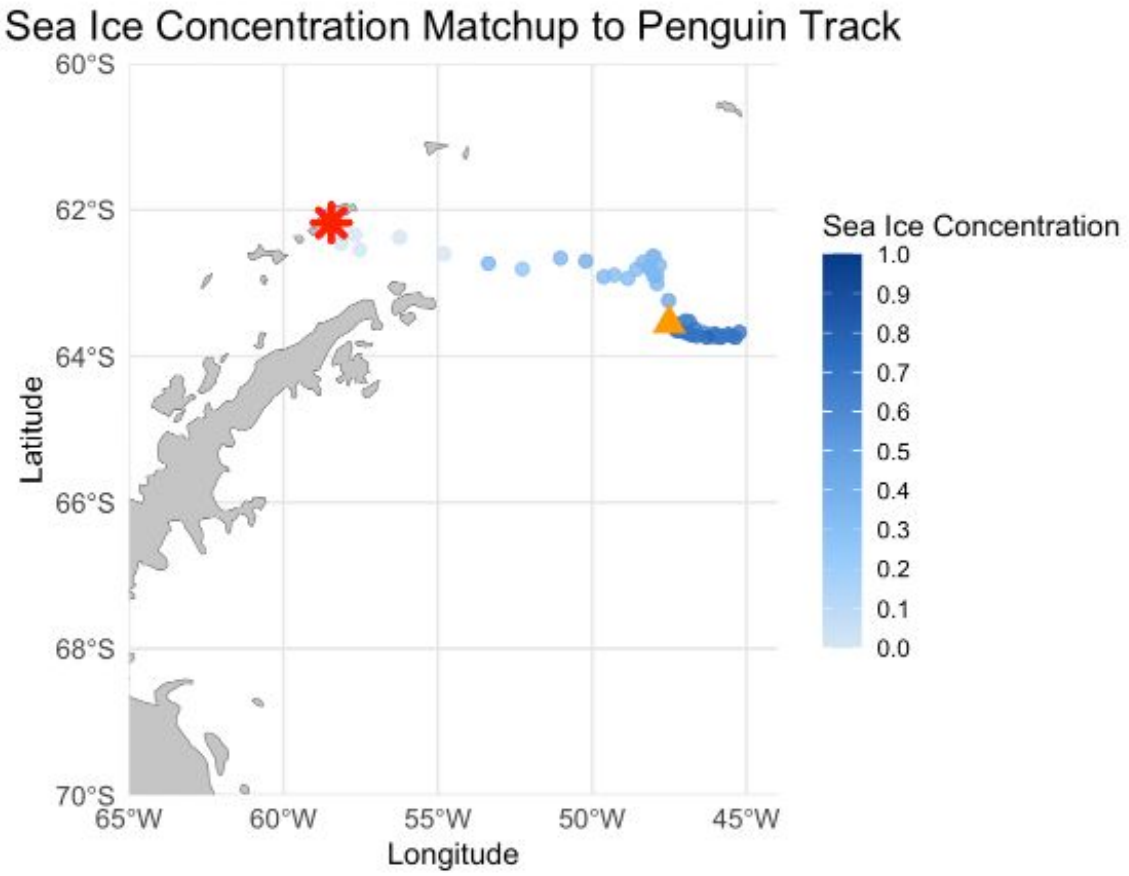


matchup-polar-data-to-track-data

Match up telemetry data (a set of moving x, y, t points) to projected data by converting the projection of the trackpoints

Tutorial demonstrates:

- Accessing satellite data from PolarWatch ERDDAP
- Changing the projection of a dataset
- Using rerddapXtraco witha projected dataset



rerddapXtracto package

- R package written by Roy Mendelssohn (SWFSC/ERD)
- Uses the rerddap and plotdap packages
- erddap, plotdap and rerddapXtracto are all available on cran
- rerddapXtracto contains several functions:
  rxtracto: extracts a variable along xyt points (i.e. a tagged animal)
  rxtractogon: extracts a variable within a user-supplied polygon
  rxtracto_3D: extracts a 3-dimensional (latitude, longitude and time) cube of a variable
  plotTrack: plots the results from rxtracto (including creating animations)
  plotBox: plots the output from rxtracto_3D
- Will work on any dataset on any ERDDAP (option to change the default ERDDAP)
- https://github.com/coastwatch-training/CoastWatch-Tutorials/blob/main/R-help/troubleshooting-rerddapXtraco.md

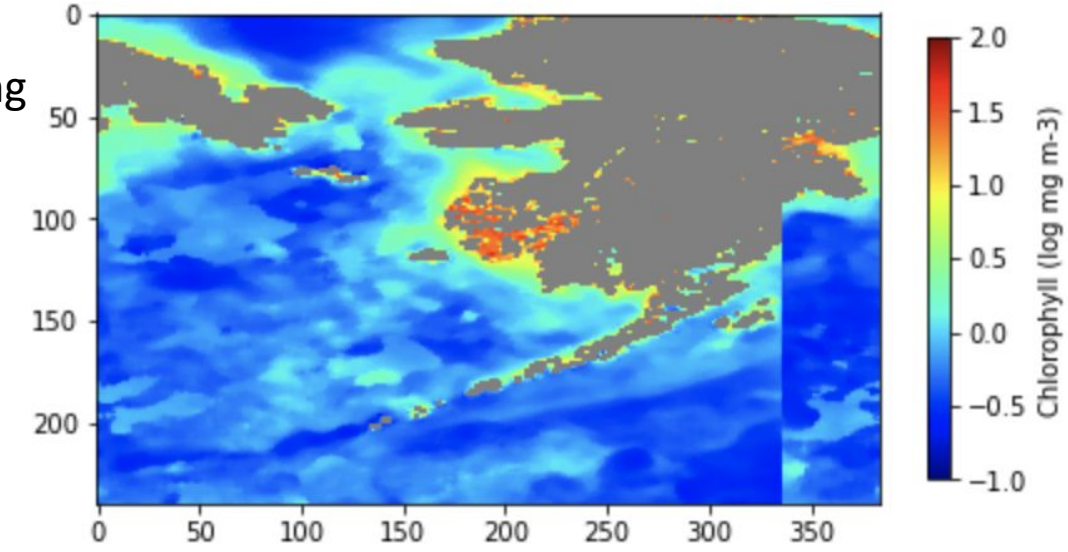
Convert-180+180-to-0-360-longitude

Work with datasets with -180° to +180° longitude values in a region that crosses the antimeridian. Convert the coordinates from (-180, +180) to (0, 360) and visualize data on a map

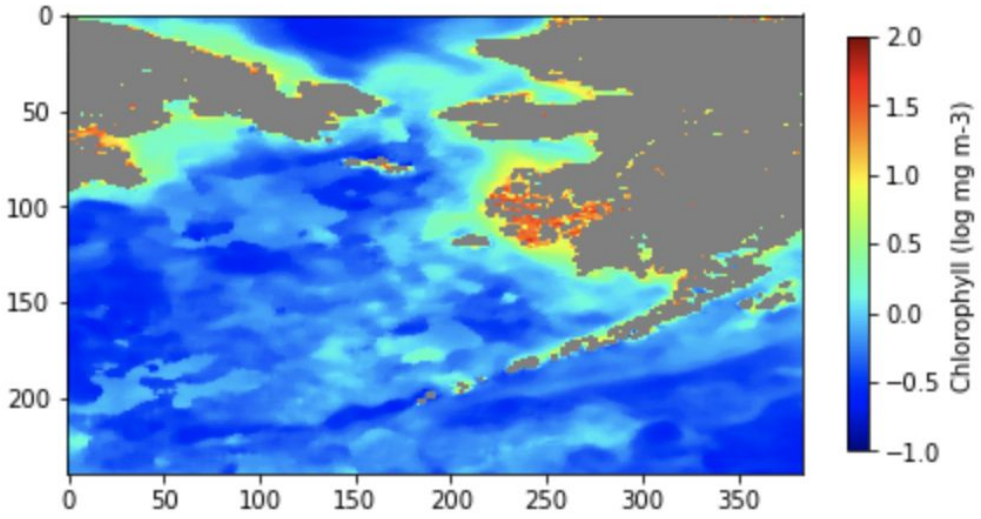
Tutorial demonstrates:

- Downloading data that crosses the antimeridian from a dataset with -180 to +180 longitude values
- Converting the data to a 0-360 longitude values
- Reordering the longitude axis so that the longitude values are in ascending order

Correcting this Map ->



To this Map ->

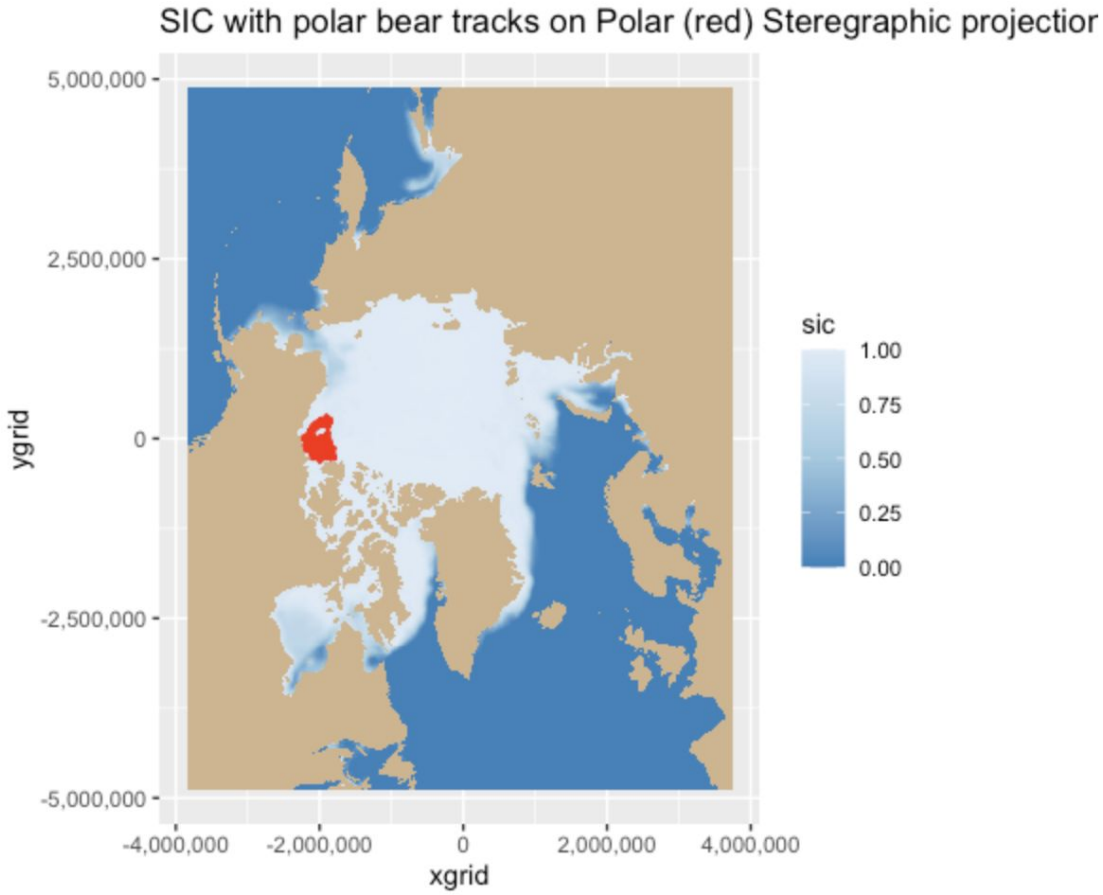


map-data-with-different-projections

Download and examine a polar stereographic projected dataset, plot the data on a projected map. Add animal track data with geographical coordinates onto the projected map.

Tutorial demonstrates:

- Accessing satellite data from ERDDAP
- Making a projected map
- Adding projected data
- Adding geographical data

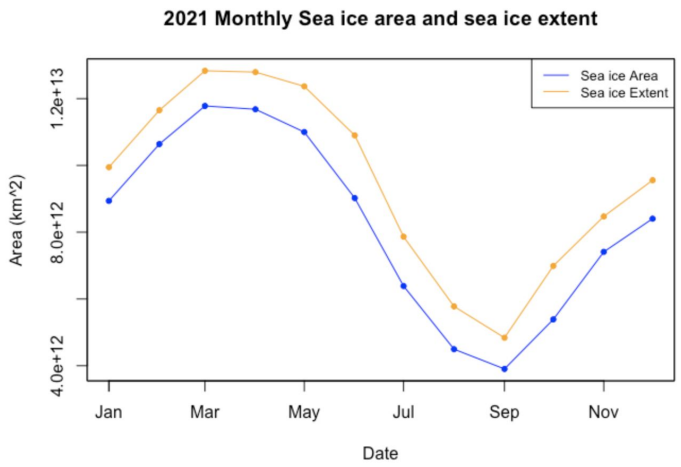


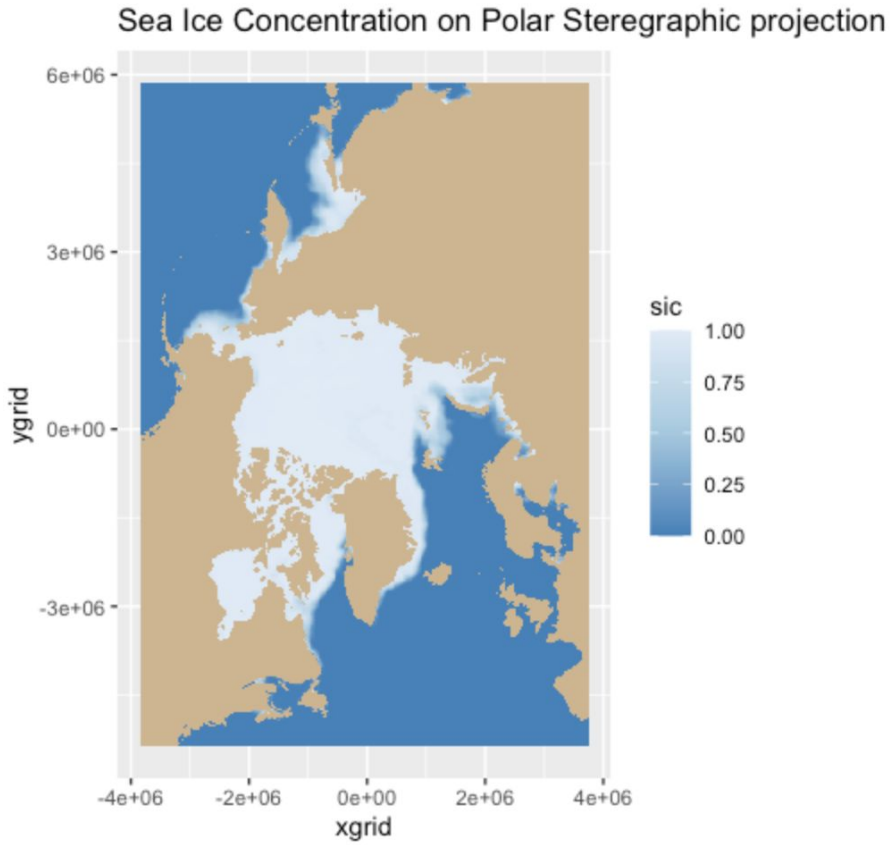
calculate-seaice-extent

View sea ice concentration (SIC) data on a map with the polar stereographic projection.
Calculate and compare sea ice area/extent from multi-year SIC datasets.

Tutorial demonstrates:

- Downloading and saving a netcdf file from the PolarWatch ERDDAP
- Accessing satellite data and metadata in polar stereographic projection
- Downloading and adding grid cell area data to a map
- Computing sea ice area and extent using sea ice concentration data
- Plotting a time series of sea ice area and extent





Transform-to-another-map-projection

Access satellite data with polar stereographic coordinates and transform it into a different coordinate system using EPSG code.

Tutorial demonstrates:

- Downloading a netcdf file from PolarWatch ERDDAP
- Accessing satellite data and metadata in polar stereographic projection
- Converting netcdf data into a dataframe
- Transforming coordinates using EPSG codes
- Mapping data using the transformed coordinates