Advances in Scientific Visualization of Geospatial Data with MATLAB

Lisa Kempler (The MathWorks Inc., Natick (MA), USA) & Steve Schäfer (The MathWorks GmbH, Aachen, Germany)

This document was exported directly from MATLAB. The original file type, a so-called Live Script, is an executable script which combines formatted text and equations with code, the computed results and visualizations. Live Scripts may be enriched with clickable control elements such as drop down menus and Live Editor tasks that serve a more complex purpose such as detrending and smoothing data. In that way, Live Scripts are ideal for both exploratory analyses and the presentation of findings in the form of an interactive narrative. Exporting Live Scripts to various formats (e.g. PDF, Word, HTML, LaTeX) makes sharing easy also with non MATLAB users.

Introduction

Data visualization plays an essential role in conveying complex relationships in real-world physical systems. As geoscience and atmospheric data quantities and data sources have increased, so, too, have the corresponding capabilities in MATLAB and Mapping Toolbox to analyze and visualize them. We will present end-to-end geospatial analysis workflows, from data access to visualization to publication. The talk will include software demonstrations of how to process and visualize large out-of-memory data, including accessing remote and cloud-based file systems; working with multiple data formats and sources, such as Web Map Service (WMS), for visualizing of publicly accessible geospatial information; and applying new 2-D and 3-D high resolution mapping visualizations. MATLAB live notebooks combining code, pictures, graphics, and equations will show the use of interactive notebooks for capturing, teaching, and research sharing.

Plotting Geographic Data

We are using MATLAB to plot data on top of a geographic basemap inside a MATLAB figure. To get an overview of the basic functionalities, we recommend Plot Geographic Data on a Map in MATLAB. Geographic plotting functions in MATLAB to display points, lines, text, density plots, and bubble charts on top of geographic basemaps are explained. Functions for overlaying latitude and longitude data on web map base layers are available in MATLAB. For visualizing data in other projected coordinate systems, converting between systems, performing geodesy calculations, or working with geographic file types, i.e. shapefiles, GPX files and GeoTIFF files, we recommend using the Mapping Toolbox, as preactically illustrated in these examples.

Geospatial Analysis Workflows

We will use earthquake data provided by the U.S. Geological Survey Earthquake Catalog [1], using the documented API this service. This allows us to perform customized searches for earthquake information. We will investigate all worldwide earthquakes that were detected in the last 15 days.

```
t2 = datetime("now") - caldays(1);
t1 = t2 - caldays(15);
daterange = "starttime=" + string(t1) + "&endtime=" + string(t2);
```

We obtain the data for the range of dates selected using the `webread`, which can read content from RESTful web services, and for this case creates a MATLAB table variable.

```
opts = weboptions("Timeout",25);
quakes = webread("https://earthquake.usgs.gov/fdsnws/event/1/query?format=csv&" + ...
    daterange,opts);
```

To preprocess the data for later visualization and analysis, we convert the data type observed event type to a categorical and select the data of interest. The resulting table is then previewed in the Live Editor.

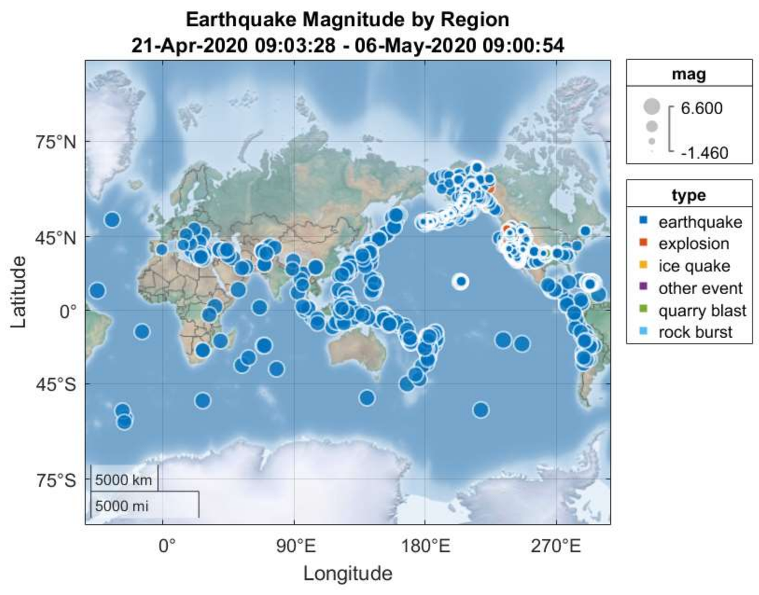
```
quakes.time = datetime(quakes.time,...
    "InputFormat",'yyyy-MM-dd''T''HH:mm:ss.SSS''Z');
quakes = table2timetable(quakes);
quakes.type = categorical(quakes.type);
quakes = quakes(:,["latitude","longitude","depth","mag","place","type"]);
head(quakes)
```

```
ans = 8×6 timetable
```

	time	latitude	longitude	depth	mag	place	type
1	06-May-20…	19.2198	-155.4082	30.4800	1.9700	'7km ENE o…	earthquake
2	06-May-20…	33.1848	-115.6122	3.8100	1.4700	'11km SW o…	earthquake
3	06-May-20…	17.9785	-66.9661	5.0000	2.6100	'5km W of F…	earthquake
4	06-May-20…	33.1957	-115.6113	3.0300	1.2800	'10km WS…	earthquake
5	06-May-20…	33.1982	-115.6108	3.3700	1.2700	'10km WS…	earthquake
6	06-May-20…	44.6857	-111.8557	4.0800	2.3200	'58km E of …	earthquake
7	06-May-20…	33.1778	-115.6172	3.8000	1.2200	'11km WN…	earthquake
8	06-May-20…	33.1932	-115.6173	4.2100	0.9200	'10km WS…	earthquake

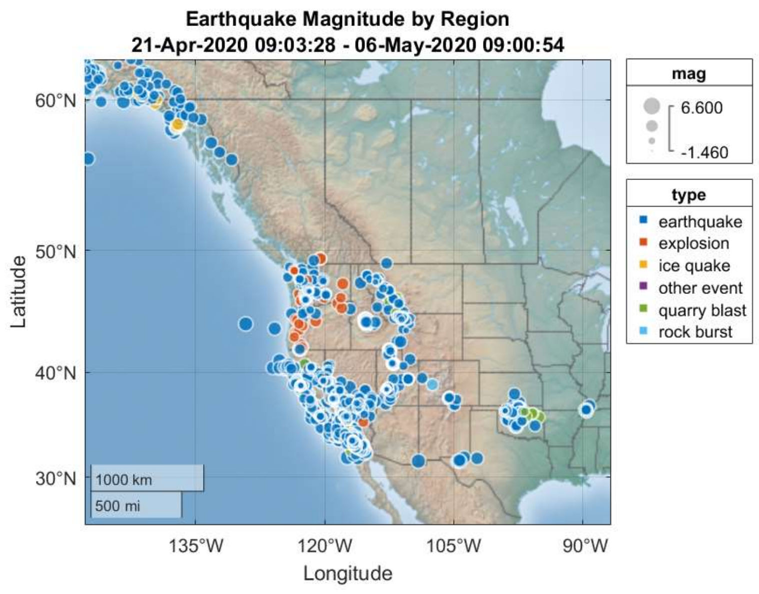
Using a map as a background, the geographic bubble chart plots your data as filled, colored circles, called *bubbles*, at locations on the map specified by longitude and latitude. You can use the size and color of the bubbles to indicate data values at these locations.

```
geobubble(quakes,"latitude","longitude","SizeVariable","mag",...
    "ColorVariable","type","Basemap","colorterrain",...
    "BubbleWidthRange",[2 10]);
title(["Earthquake Magnitude by Region",...
    string(min(quakes.time))+" - "+string(max(quakes.time))])
```



We can zoom in to examine other types of events, by using the interactivity of MATLAB figures in the Live Editor, or by specifying specific latitudes and longitudes as limits for our geographic map

```
geolimits([25.2 62.2],[-147.9 -86.7])
```



We can compute statistical descriptions, i.e. the occurrence of earthquakes within a specific geographic region, and the statistical mean value of an earthquake's magnitude and depth, of the categorical groups by using the groupsummary command. We preprocess the location, to obtain either state or country where an earthquake has been recorded.

```
quakes = preprocessLocations(quakes);
g = groupsummary(quakes,"Loc","mean",["mag","depth"]);
topkrows(g,10,"GroupCount")
```

ans = 10×4 table

	Loc	GroupCount	mean_mag	mean_depth
1	California	2340	0.9417	6.1861
2	Alaska	1863	0.8950	26.6674
3	Puerto Rico	396	2.7413	12.6944
4	Hawaii	254	2.0310	24.8759
5	Utah	228	0.8279	5.5472
6	Idaho	208	2.4145	8.9342
7	Nevada	144	0.7792	5.8535
8	Washington	109	0.8191	7.3942

	Loc	GroupCount	mean_mag	mean_depth
9	Montana	108	0.7177	6.7241
10	Oklahoma	95	1.6423	6.2957

The analysis could be further improved by filtering for specific condition, i.e. considering only earthquake events greater than a given magnitude of interest.

Working with Web Map Service (WMS)

A web map is an interactive, dynamic map display that uses basemaps from Web-based data sources to give your data a visually rich contextual background. With Mapping Toolbox, we can create web map displays from sources such as OpenStreetMap, ESRI ArcGIS Online, and many WMS servers. We can pan across the map, zoom in/out to view higher/lower resolution basemap data, specify the geographic region to view, and more. We will create overlays of markers and plots with related attribute data such as names and colors. Web map displays enable simple map creation using high resolution base maps without having to load the entire dataset into MATLAB.

To get started, we create a web map that highlights an area of interest. We can use the `webmap` command to create a web map which opens in an extra window shown in figure 1. We are displaying OpenTopoMap as the custom basemap.

```
attribution = [ ...
  "map data:  " + char(uint8(169)) + " OpenStreetMap contributors, SRTM", ...
  "map style: " + char(uint8(169)) + " OpenTopoMap (CC-BY-SA)"];
addCustomBasemap('opentopomap','a.tile.opentopomap.org','Attribution',attribution, ...
  'DisplayName','Open Topo Map')
webmap opentopomap
```

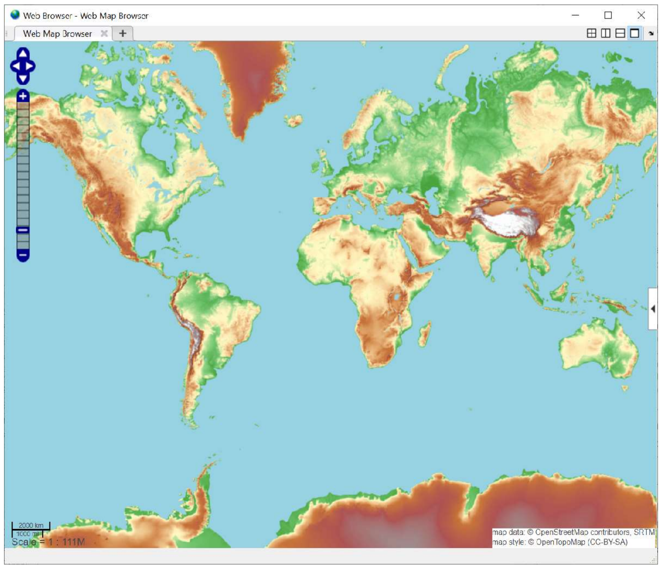

Web Browser - Web Map Browser
Web Map Browser
2000 km
1000 mi
Scale = 1 : 111M
map data: © OpenStreetMap contributors, SRTM
map style: © OpenTopoMap (CC-BY-SA)

Figure 1: Web Map Browser showing the created web map displaing OpenTopoMap as custom basemap.

Set Latitude and Longitude Limits

By zooming and panning around, we can see the latitude and longitude limits on the web map. The command `wmlimits` gets and sets these. We can identify an area of interest, i.e. the venue of the EGU General Assembly, and set the limits programmatically, as can be seen with further modifications in Figure 2.

```
wmlimits([48.24552 48.22515], [16.39547 16.43423])
```

Create a Circular Buffer Around Points

There are multiple ways to buffer a shape or point. For circular shapes, we can use `scircle1`. For polygons, or lines, we can use `bufferm`. As a next step, we add a small circular buffer around the chosen point of interest.

```
[circlat, circlon] = scircle1(48.23528, 16.41444, 0.001);
wmpolygon(circlat, circlon, 'FeatureName', '#shareEGU20','FaceAlpha', 0.3, 'FaceColor', 'cyan');
```

Add Markers Indicating the Location

We then use `wmmarker` to add a marker or icon to the location of interest. We can provide additional information and descriptions when placing the marker. It is possible to provide distinct textual information by placing the marker on top of the circular area.

```
wmmarker(48.23528, 16.41444, 'FeatureName', 'EGU2020', 'Description', 'Sharing Geoscience Online', 'Icon', which('matlabicon.gif'))
```

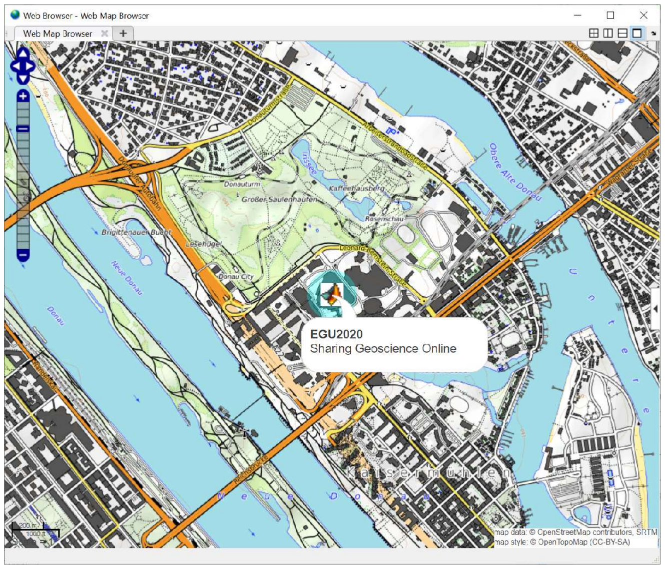


Figure 2: The Web Map Browser shown in Figure 1 with set latitude and longitude limits around the venue of the EGU General Assembly, along with a circular buffer and a marker indicating the location.

2-D and 3-D Visualizations for Northern Hemisphere Sea Ice Concentration

In this example we are visualizing behavior of the sea ice concentration in the northern hemisphere. The data was produced at the EUMETSAT Ocean and Sea Ice Satelitte Application Facility (OSI SAF) as Global Sea Ice Concentration Climate Data Records (1978-2015) [2]. The dataset is available for download from the EUMETSAT OSI SAF webportal, as well as the Copernicus Climate Data Store in Network Common Data Form (NetCDF) file format.

We are interested in the annual changes in sea ice contration. We are using December 1st as our yearly reference day, and utilized the available data for the covered years for our simulations. For this demo we stored the data set in a local folder named data.

MATLAB supports high-level functions for reading and writing NetCDf files, and provides the NetCDF Library Package for low-level functionality. A detailed function overiew is given in the MATLAB documentation. We are using this and other build-in features to preprocess the data, and to build multiple visualisations in 2D and 3D on top of that. The required steps are explained and illustrated below.

Create Variable Containing Data Files

Find files for sea ice concentration in NetCDF4 data format.

```
files = dir('data/*');
t = struct2table(files);
files =  string(t.name(contains(t.name,'conc') & contains(t.name,'.nc4')));
```

Examine Ice Concentration Data

```
file = fullfile(pwd,'data',files(end));
ncdisp(file)
```

```
Source:
           C:\MATLAB Work\EGU\data\ice_conc_nh_ease-125_reproc_201412011200.nc4
Format:
           netcdf4_classic
Global Attributes:
           title                  = 'Reprocessed Sea Ice Concentration from the OSI SAF EUMETSAT'
           product_name           = 'osi_saf_ice_conc_reproc'
           product_status         = 'offline'
           abstract               = 'The reprocessing of sea ice concentration  is  obtained  from
                                    passive microwave satellite data over the polar  regions.  It
                                    is based on atmospherically corretected signal and an optimal
                                    sea ice concentration algorithm. This  product  is  available
                                    for free from  the  EUMETSAT  Ocean  and  Sea  Ice  Satellite
                                    Application Facility (OSI SAF).'
```

```
years = extractAfter(files,'reproc_');
years = extractBefore(years,"12011200.nc4");
```

Read Ice Concentration Data and Data Attribution

```
ice_concentration = ncread(file,'ice_conc');
lat = ncread(file,'lat');
lon = ncread(file,'lon');
latlim = double([min(lat(:)) max(lat(:))]);

attribution = ncreadatt(file,'/','copyright_statement');
attribution = "Data Source: " + string(attribution);
dispYear = years(end);
```

```
titlestr = ["Sea Ice Concentration"; "December " + dispYear; attribution];
```

Create 2D Map of Ice Concentration

Create a map axes with a Lambert Equal Area Azimuthal Projection with the origin at the North Pole. Use a grid color of white.

```
load ice_colormap

figure('Colormap',ice_colormap)
title(titlestr)
ax = axesm('eqaazim','Origin',[90 0 0],'MapLatLimit',latlim, ...
    'Grid','on','GColor',[1 1 1]);
```

Add a color bar using the custom color map and set ticks and label.

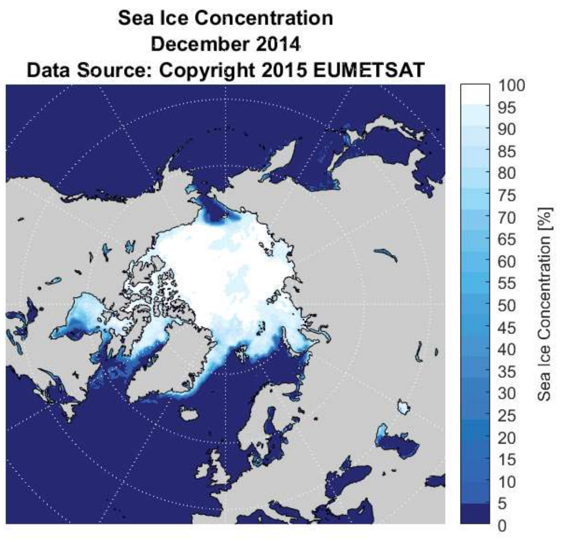
```
levels = 0:5:100;
h = colorbar ;
h.Ticks = levels;
caxis([0 100])
h.Label.String = 'Sea Ice Concentration [%]';
```

Display polygons of land areas and a texture map of ice concentrations.

```
iceTextureMap = geoshow(lat,lon,ice_concentration,'DisplayType','texturemap');
land = shaperead('landareas','UseGeoCoords',true);
geoshow(land,'FaceColor',[204 204 204]/255);
```

Set the axes limits to zoom into the map.

```
axis off
nlim = .82;
set(ax,'XLim',[-nlim,nlim],'YLim',[-nlim,nlim])
```



Animate Sea Ice Concentration Over Time in the Live Editor

Enable for-loop animations in the Live Editor by setting the `matlab.editor.AllowFigureAnimations` setting to `true`:

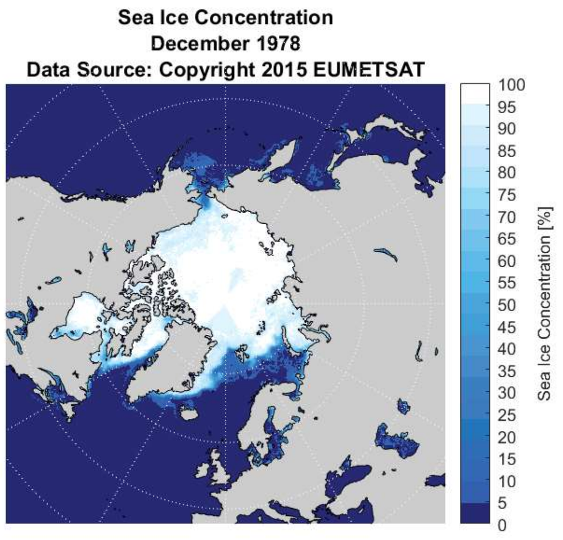
```
settingsObj = settings;
settingsObj.matlab.editor.AllowFigureAnimation.TemporaryValue = 1;
```

The below code shows changes in plotted data over time by animating the yearly changes in ice concentration. The drawnow function displays the changes after each iteration through the loop.

```
for ii = numel(years):-1:1
    yearfile = replace(file,dispYear,years(ii));
    ice_concentration = ncread(yearfile,'ice_conc');
```

Use set to change the displayed data of the graphical object.

```
    set(iceTextureMap,'CData',ice_concentration)
    newTitle = replace(titlestr,dispYear,years(ii));
    title(newTitle)
    drawnow
end
```



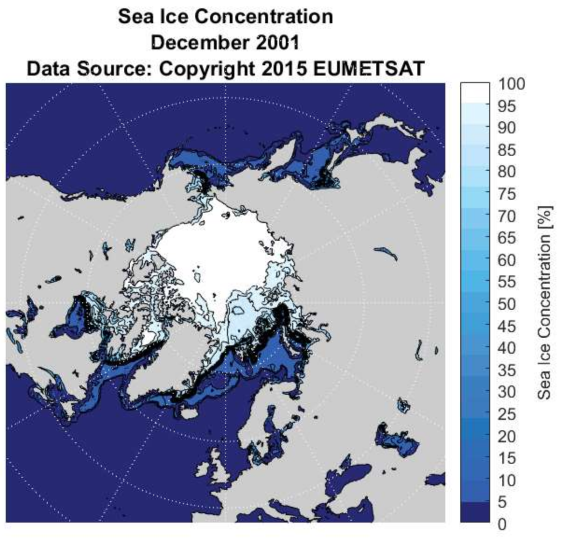
Create Contour Map of Ice Concentration

Select year of interest using Live Controls. The Live Control is configured to rerun the section upon changes.

```
year = "2001";
yearfile = replace(file,dispYear,year);
ice_concentration = ncread(yearfile,'ice_conc');
set(iceTextureMap,'CData',ice_concentration)
newTitle = replace(titlestr,dispYear,year);
title(newTitle)
```

The contour map supports visual distinction between the different levels of ice concentrations. Use `geoloc2grid` to create a regular gridded data set from longitude and latitude information.

```
[Z,refvec] = geoloc2grid(double(lat),double(lon),ice_concentration,.5);
iceContourMap = contourm(Z,refvec,'LevelList',levels,'Parent',ax,'Color','k');
```



Display Sea Ice Concentration Contours on Geographic Globe

Convert the contour matrix created with contourm to a geoshape vector using a local function.

```
iceGeoShape = contourToGeoshape(iceContourMap);
```

Create MATLAB uifigure including colorbar and display the contour lines onto a geographic globe. The result is shown in figure 3.

```
uif = uifigure('Colormap',ice_colormap,'Visible','off', ...
    'AutoResizeChildren','off');
uif.Name = strjoin(newTitle,', ');

ax = axes(uif,'Units','normalized','Position',[.9 0.02 .005 .95]);
```

```
axis(ax,'off')
caxis(ax,[0 100])
h = colorbar(ax);
h.Label.String = 'ice concentration percentage (0:100)';
h.Ticks = levels;

gg = geoglobe(uif,'Terrain','none','Basemap', 'streets-dark', ...
    'NextPlot','add','Position',[0 0 .89 1]);

n = 0;
for level = levels
    index = iceGeoShape.Level == level;
    n = n + 1;
    color = ice_colormap(n,:);
    geoplot3(gg,iceGeoShape(index).Latitude,iceGeoShape(index).Longitude,[],'Color',color)
end
uif.Visible = 'on';
```

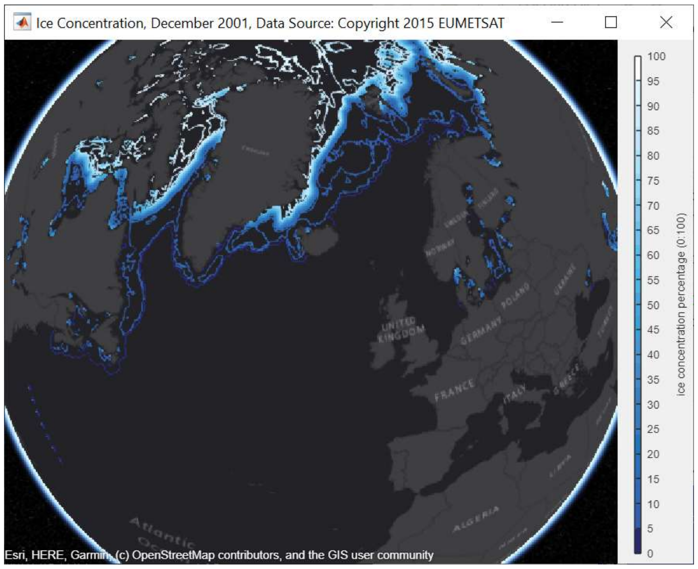

Ice Concentration, December 2001, Data Source: Copyright 2015 EUMETSAT
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
ice concentration percentage (0:100)
UNITED KINGDOM
GERMANY
POLAND
FRANCE
ITALY
NORWAY
ALGERIA
Atlantic
Esri, HERE, Garmin, (c) OpenStreetMap contributors, and the GIS user community

Figure 3: Screen shot of the geographic globe with contour lines showing the percentage of see ice concentration from 0 to 100 percent for December 2001.

References

[1] U.S. Geological Survey, 2020, Earthquake Catalog available online at https://earthquake.usgs.gov/earthquakes/search/. (Accessed May 5, 2020)

[2] EUMETSAT Ocean and Sea Ice Satelitte Application Facility. *Global sea ice concentration climate data records 1978-2015* (v1.2, 2015), [Online]. Norwegian and Danish Meteorological Institutes. doi: http://dx.doi.org/10.15770/EUM_SAF_OSI_0001 and http://dx.doi.org/10.15770/EUM_SAF_OSI_0005.

Local Functions

Display Contours on Geographic Globe

Convert the contour matrix created with contourm to a geoshape vector. Display the contour lines onto a geographic globe and include a colorbar.

```
function s = contourToGeoshape(c)
```

Convert contour matrix to geoshape

```
    lon = c(1,:);
    lat = c(2,:);
    n = length(lat);
    k = 1;
    index = 1;
    S = struct('Lat',[],'Lon',[],'Level',[]);
    while k < n
        level = lon(k);
        numVertices = lat(k);
        S(index).Lat = lat(k+1:k+numVertices);
        S(index).Lon = lon(k+1:k+numVertices);
        S(index).Level = level;
        k = k + numVertices + 1;
        index = index + 1;
    end
```

Remove single vertex.

```
    s = geoshape(S);
    index = false(length(s),1);
    for k = 1:length(s)
        if isscalar(s(k).Latitude)
            index(k) = true;
        end
    end
    s(index) = [];
end
```

Preprocess Location Strings in USGS Earthquake Catalog Data

Extract the state or country from the string variable place.

```
function quakes = preprocessLocations(quakes)
    quakes.Loc = extractAfter(quakes.place,",");
    quakes.Loc = strip(quakes.Loc);
```

If they didn"t have a space, they were left missing and we use the full label in that case.

```
    idx = ismissing(quakes.Loc);
    quakes.Loc(idx) = quakes.place(idx);
    idx = endsWith(quakes.Loc,"region");
    quakes.Loc(idx) = erase(quakes.Loc(idx)," region");
```

Now convert to categorical.

```
    quakes.Loc = categorical(quakes.Loc);
```

Clean up some categories.

```
    quakes.Loc = mergecats(quakes.Loc,["CA","California"],"California");
    quakes.Loc = mergecats(quakes.Loc,["MX","B.C., MX","Mexico"],"Mexico");
end
```