// NASA\_NEXDCO30\_FWW

// Calculate Frequency of Warm Winters (FWW)

// from NASA/NEX-DCP30 downscaled climate data

// A "warm winter" is a winter during which the mean monthly temperature

// exceeds 0 degree C during any one of the core winter months (Dec., Jan., Feb.)

//

// Created 8 February 2017, David E. Rupp, [david.rupp@oregonstate.edu](mailto:david.rupp@oregonstate.edu)

//

// Updated and modified by Gene Mar, 8 June 2017

// Modified categories and color pallete and added legends: David Rupp 19 Sept 2017

// References: https://cds.nccs.nasa.gov/nex/

// https://www.skepticalscience.com/rcp.php?t=1

//

// Updated by Gene Mar 29 Sept 2017 - Added histogram info to each category in legend

//

// Updated and modified by Anne Nolin, 28 Oct 2020

// Commented out the references to ski resorts (GEE no longer uses fusion tables)

var collectionPath = 'NASA/NEX-DCP30';

// var skiResorts = ee.FeatureCollection('ft:1q\_x12r4gjHhPpnGB2krnlsKIEks6Yr7TzpCzY875');

var scenario1 = 'historical';

var scenario2 = 'rcp85';

var scenario1Updated = 0;

var scenario2Updated = 0;

var scenario1YearsUpdated = 0;

var scenario2YearsUpdated = 0;

var GCMUpdated = 0;

var GCM = 'ACCESS1-0';

var nYears1;

var nYears2;

var theMask;

var huc = 'None';

var WaterShedsFC;

var FWW1;

var FWW2;

var deltaFWW;

var initLat = 45.34161;

var initLon = -121.68887;

var nexdcp30 = {};

// name of the legend

var names = ['0-24.9','25-49.9','50-74.9','75-99.9','100'];

// Creates and styles 1 row of the legend.

var makeRow = function(color, name) {

// Create the label that is actually the colored box.

var colorBox = ui.Label({

style: {

backgroundColor: '#' + color,

// Use padding to give the box height and width.

padding: '8px',

margin: '0 0 4px 0'

}

});

// Create the label filled with the description text.

var description = ui.Label({

value: name,

style: {margin: '0 0 4px 6px'}

});

// return the panel

return ui.Panel({

widgets: [colorBox, description],

layout: ui.Panel.Layout.Flow('horizontal')

});

};

nexdcp30.fwwApply = function() {

Map.clear();

var FWW1 = nexdcp30.fww(scenario1, 1);

FWW1 = FWW1.updateMask(ee.Image(theMask));

//Map.addLayer(FWW1, {min: 0, max: nYears1, palette: ['FFFAFA', 'FF0000']}, scenario1);

Map.addLayer(FWW1, {min: 0, max: 100, palette: ['FFFAFA', 'FF0000']}, scenario1 + ' (%)');

//Map.addLayer(FWW1, {min: 0, max: 100, palette: ['f7fbff', '08306b']}, scenario1 + ' (%)');

Map.setCenter(initLon, initLat);

var FWW2 = nexdcp30.fww(scenario2, 0);

FWW2 = FWW2.updateMask(ee.Image(theMask));

//Map.addLayer(FWW2, {min: 0, max: nYears2, palette: ['FFFFFF', 'FF0000']}, scenario2);

Map.addLayer(FWW2, {min: 0, max: 100, palette: ['FFFAFA', 'FF0000']}, scenario2 + ' (%)');

//Map.addLayer(FWW2, {min: 0, max: 100, palette: ['fff5eb', '7f2704']}, scenario2 + ' (%)');

//Map.setCenter(-114.35, 48.48, 4);

// Subtract FWW2 from FWW1

var deltaFWW = FWW2.subtract(FWW1);

//var deltaFWW\_p = deltaFWW.divide(Math.max(nYears1, nYears2)).multiply(100);

deltaFWW = deltaFWW.updateMask(ee.Image(theMask));

//deltaFWW\_p = deltaFWW\_p.updateMask(ee.Image(theMask));

var layerText = scenario2 + ' - ' + scenario1 + ' (%)';

Map.addLayer(deltaFWW, {min: 0, max: 100, palette: ['FFFF00', 'FF0000']}, layerText);

//Map.addLayer(deltaFWW, {min: 0, max: Math.max(nYears1, nYears2), ski: ['FFFFFF', 'FF0000']}, theText3);

//Map.addLayer(deltaFWW\_p, {min: 0, max: 100, palette: ['FFFFFF', 'FF0000']}, 'Change (%)');

//Map.setCenter(-100, 38, 4);

if (huc != 'None') {

alert('Please select approximate location of watershed on map, but first\n\n \xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0Press OK to continue!');

WaterShedsFC = ee.FeatureCollection('USGS/WBD/2017/HUC' + huc);

Map.onClick(function(coords) {

print(coords);

var point = ee.Geometry.Point([coords.lon, coords.lat]);

var watershed = WaterShedsFC.filter(ee.Filter.bounds(point));

//print(watershed)

var shed = ee.Feature(watershed.first());

//print(shed);

var shedname = shed.get('name');

//print(shedname);

var watershedname = ee.String(shedname).getInfo();

print(watershedname);

//print(point);

print(coords.lon);

print(coords.lat);

Map.addLayer(watershed, {color: '000000'}, watershedname, 1, 0.5);

Map.setCenter(coords.lon, coords.lat, 6);

var roi = watershed.first().geometry();

var ws\_deltaFWW = deltaFWW.reduceRegion(ee.Reducer.fixedHistogram(0, 125, 5), roi, 1000);

var ws\_FWW1 = FWW1.reduceRegion(ee.Reducer.fixedHistogram(0, 125, 5), roi, 1000);

var ws\_FWW2 = FWW2.reduceRegion(ee.Reducer.fixedHistogram(0, 125, 5), roi, 1000);

var ws\_theMask = theMask.unmask(1).reduceRegion(ee.Reducer.fixedHistogram(0, 2, 2), roi, 1000);

//print(ws\_deltaFWW);

print(ws\_FWW1);

//print(ws\_FWW2);

//print(ws\_theMask);

// var ARS = blank.where(tas.gt(273.15),1);

var bucket21 =[];

var bucket2 = [];

var bucket1 = [];

var total = ee.List(ws\_theMask.get('constant')).getInfo()[1][1];

for (var i=0; i < 5; i++) {

bucket21[i] = ee.List(ws\_deltaFWW.get('constant')).getInfo()[i][1];

bucket2[i] = ee.List(ws\_FWW2.get('constant')).getInfo()[i][1];

bucket1[i] = ee.List(ws\_FWW1.get('constant')).getInfo()[i][1];

}

var maxBucket21 = 0;

var maxBucket2 = 0;

var maxBucket1 = 0;

for (i=0; i < 5; i++) {

bucket21[i] = (100\*bucket21[i]/total).toFixed(1);

if (bucket21[i] > maxBucket21) maxBucket21 = bucket21[i];

bucket2[i] = (100\*bucket2[i]/total).toFixed(1);

if (bucket2[i] > maxBucket2) maxBucket2 = bucket2[i];

bucket1[i] = (100\*bucket1[i]/total).toFixed(1);

if (bucket1[i] > maxBucket1) maxBucket1 = bucket1[i];

}

//print(maxBucket);

for (i=0; i < 5; i++) {

if (maxBucket21 == 100) {

bucket21[i] = ('00' + bucket21[i]).slice(-5);

}

else {

bucket21[i] = ('00' + bucket21[i]).slice(-4);

}

if (maxBucket2 == 100) {

bucket2[i] = ('00' + bucket2[i]).slice(-5);

}

else {

bucket2[i] = ('00' + bucket2[i]).slice(-4);

}

if (maxBucket1 == 100) {

bucket1[i] = ('00' + bucket1[i]).slice(-5);

}

else {

bucket1[i] = ('00' + bucket1[i]).slice(-4);

}

}

//Map.remove(legend);

var legend = ui.Panel({

style: {

position: 'bottom-right',

padding: '8px 15px'

}

});

var layerText0 = scenario2 + ' - ' + scenario1;

legendTitle = ui.Label({

value: watershedname + ' : ' + layerText0,

style: {

fontWeight: 'bold',

fontSize: '16px',

margin: '0 0 4px 0',

padding: '0'

}

});

legend.add(legendTitle);

//print(bucket);

//names = [bucket21[0] + '% \xa0\xa00',

// bucket21[1] + '% \xa0\xa025',

// bucket21[2] + '% \xa0\xa050',

// bucket21[3] + '% \xa0\xa075',

// bucket21[4] + '% \xa0\xa0100'];

names = ['\xa0\xa00-24.9%\xa0\xa0\xa0\xa0:\xa0\xa0' + bucket21[0] + '% of area',

'\xa0\xa025-49.9%\xa0\xa0:\xa0\xa0' + bucket21[1] + '% of area',

'\xa0\xa050-74.9%\xa0\xa0:\xa0\xa0' + bucket21[2] + '% of area',

'\xa0\xa075-99.9%\xa0\xa0:\xa0\xa0' + bucket21[3] + '% of area',

'\xa0\xa0100%\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0:\xa0\xa0' + bucket21[4] + '% of area'];

//print(names);

var palette =['FFF000','FFB400','FF7800','FF3C00','FF0000'];

for (i = 0; i < 5; i++) {

legend.add(makeRow(palette[i], names[i]));

}

Map.add(legend);

legend = ui.Panel({

style: {

position: 'bottom-left',

padding: '8px 15px'

}

});

legendTitle = ui.Label({

value: watershedname + ' : ' + scenario2,

style: {

fontWeight: 'bold',

fontSize: '16px',

margin: '0 0 4px 0',

padding: '0'

}

});

legend.add(legendTitle);

//print(bucket);

//names = [bucket2[0] + '% \xa0\xa00',

// bucket2[1] + '% \xa0\xa025',

// bucket2[2] + '% \xa0\xa050',

// bucket2[3] + '% \xa0\xa075',

// bucket2[4] + '% \xa0\xa0100'];

names = ['\xa0\xa00-24.9%\xa0\xa0\xa0\xa0:\xa0\xa0' + bucket2[0] + '% of area',

'\xa0\xa025-49.9%\xa0\xa0:\xa0\xa0' + bucket2[1] + '% of area',

'\xa0\xa050-74.9%\xa0\xa0:\xa0\xa0' + bucket2[2] + '% of area',

'\xa0\xa075-99.9%\xa0\xa0:\xa0\xa0' + bucket2[3] + '% of area',

'\xa0\xa0100%\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0:\xa0\xa0' + bucket2[4] + '% of area'];

//print(names);

palette =['FFFAFA','FFBBBB','FF7D7D','FF3E3E','FF0000'];

for (i = 0; i < 5; i++) {

legend.add(makeRow(palette[i], names[i]));

}

legendTitle = ui.Label({

value: watershedname + ' : ' + scenario1,

style: {

fontWeight: 'bold',

fontSize: '16px',

margin: '0 0 4px 0',

padding: '0'

}

});

legend.add(legendTitle);

//print(bucket);

//names = [bucket1[0] + '% \xa0\xa00',

// bucket1[1] + '% \xa0\xa025',

// bucket1[2] + '% \xa0\xa050',

// bucket1[3] + '% \xa0\xa075',

// bucket1[4] + '% \xa0\xa0100'];

names = ['\xa0\xa00-24.9%\xa0\xa0\xa0\xa0:\xa0\xa0' + bucket1[0] + '% of area',

'\xa0\xa025-49.9%\xa0\xa0:\xa0\xa0' + bucket1[1] + '% of area',

'\xa0\xa050-74.9%\xa0\xa0:\xa0\xa0' + bucket1[2] + '% of area',

'\xa0\xa075-99.9%\xa0\xa0:\xa0\xa0' + bucket1[3] + '% of area',

'\xa0\xa0100%\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0:\xa0\xa0' + bucket1[4] + '% of area'];

//print(names);

for (i = 0; i < 5; i++) {

legend.add(makeRow(palette[i], names[i]));

}

Map.add(legend);

//alert('Hit \"Apply\" to select another watershed, but first\n\n \xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0Press OK to continue!');

});

}

else {

// LEGEND for Frequency of Warm Winters //

// set position of panel

var legend = ui.Panel({

style: {

position: 'bottom-left',

padding: '8px 15px'

}

});

// Create legend title

var legendTitle = ui.Label({

value: 'Frequency (%)',

style: {

fontWeight: 'bold',

fontSize: '16px',

margin: '0 0 4px 0',

padding: '0'

}

});

// Add the title to the panel

legend.add(legendTitle);

// Create legend subtitle

var legendTitle2 = ui.Label({

value: '(' + scenario1 + ' 100% is',

style: {

fontSize: '14px',

margin: '0 0 4px 0',

padding: '0'

}

});

// Add the subtitle to the panel

legend.add(legendTitle2);

// Create legend subtitle

var legendTitle3 = ui.Label({

value: 'left transparent)',

style: {

fontSize: '14px',

margin: '0 0 4px 0',

padding: '0'

}

});

// Add the subtitle to the panel

legend.add(legendTitle3);

// Palette with the colors

var palette =['FFFAFA','FFBBBB','FF7D7D','FF3E3E','FF0000'];

// Add color and and names

for (var i = 0; i < 5; i++) {

legend.add(makeRow(palette[i], names[i]));

}

// add legend to map

Map.add(legend);

// LEGEND for change in FWW //

// set position of panel

legend = ui.Panel({

style: {

position: 'bottom-right',

padding: '8px 15px'

}

});

// Create legend title

//var theText = scenario2.concat(' - ');

legendTitle = ui.Label({

value: scenario2 + ' - ',

style: {

fontWeight: 'bold',

fontSize: '16px',

margin: '0 0 4px 0',

padding: '0'

}

});

// Add the title to the panel

legend.add(legendTitle);

// Create legend title

//var theText = scenario1.concat(' (%)');

legendTitle2 = ui.Label({

value: scenario1 + ' (%)',

style: {

fontWeight: 'bold',

fontSize: '16px',

margin: '0 0 4px 0',

padding: '0'

}

});

// Add the title to the panel

legend.add(legendTitle2);

// Palette with the colors

palette =['FFF000','FFB400','FF7800','FF3C00','FF0000'];

// Add color and and names

for (i = 0; i < 5; i++) {

legend.add(makeRow(palette[i], names[i]));

}

// add legend to map

Map.add(legend);

}

// Map.addLayer(skiResorts, {'color':'FF00FF'}, 'Ski Resorts');

// Map.addLayer(skiResorts, {'color':'000000'}, 'Ski Resorts');

};

nexdcp30.filters = {

scenario1YearRange : ui.Textbox('Scenario1YearRange','1970 : 1999', function(){scenario1YearsUpdated = 1; return;}),

scenario2YearRange : ui.Textbox('Scenario2YearRange','2070 : 2099', function(){scenario2YearsUpdated = 1; return;}),

applyButton: ui.Button('Apply', nexdcp30.fwwApply)

};

nexdcp30.filters.panel = ui.Panel({

widgets:[

ui.Label({

value: 'Frequency of Warm Winters Calculator',

style: {fontWeight: 'bold'}

}),

ui.Panel([

ui.Label({

value:'GCM :',

style: {fontWeight: 'bold'}}),

ui.Select({

items: [

{label: 'ACCESS1-0', value: 'ACCESS1-0'},

{label: 'BNU-ESM', value: 'BNU-ESM'},

{label: 'CCSM4', value: 'CCSM4'},

{label: 'CESM1-BGC', value: 'CESM1-BGC'},

{label: 'CESM1-CAM5', value: 'CESM1-CAM5'},

{label: 'CMCC-CM', value: 'CMCC-CM'},

{label: 'CNRM-CM5', value: 'CNRM-CM5'},

{label: 'CSIRO-Mk3-6-0', value: 'CSIRO-Mk3-6-0'},

{label: 'CanESM2', value: 'CanESM2'},

{label: 'FGOALS-g2', value: 'FGOALS-g2'},

{label: 'FIO-ESM', value: 'FIO-ESM'},

{label: 'GFDL-CM3', value: 'GFDL-CM3'},

{label: 'GFDL-ESM2G', value: 'GFDL-ESM2G'},

{label: 'GFDL-ESM2M', value: 'GFDL-ESM2M'},

{label: 'GISS-E2-H-CC', value: 'GISS-E2-H-CC'},

{label: 'GISS-E2-R-CC', value: 'GISS-E2-R-CC'},

{label: 'GISS-E2-R', value: 'GISS-E2-R'},

{label: 'HadGEM2-AO', value: 'HadGEM2-AO'},

{label: 'HadGEM2-CC', value: 'HadGEM2-CC'},

{label: 'HadGEM2-ES', value: 'HadGEM2-ES'},

{label: 'IPSL-CM5A-LR', value: 'IPSL-CM5A-LR'},

{label: 'IPSL-CM5A-MR', value: 'IPSL-CM5A-MR'},

{label: 'IPSL-CM5B-LR', value: 'IPSL-CM5B-LR'},

{label: 'MIROC-ESM-CHEM', value: 'MIROC-ESM-CHEM'},

{label: 'MIROC-ESM', value: 'MIROC-ESM'},

{label: 'MIROC5', value: 'MIROC5'},

{label: 'MPI-ESM-LR', value: 'MPI-ESM-LR'},

{label: 'MPI-ESM-MR', value: 'MPI-ESM-MR'},

{label: 'MRI-CGCM3', value: 'MRI-CGCM3'},

{label: 'NorESM1-M', value: 'NorESM1-M'},

{label: 'bcc-csm1-1-m', value: 'bcc-csm1-1-m'},

{label: 'bcc-csm1-1', value: 'bcc-csm1-1'},

{label: 'inmcm4', value: 'inmcm4'}

],

placeholder: 'ACCESS1-0',

value: 'ACCESS1-0',

onChange: function(value) {

GCM = value;

GCMUpdated = 1;

}

})

], ui.Panel.Layout.flow('horizontal')),

ui.Label({

value:'Scenario1 :',

style: {fontWeight: 'bold'}}),

ui.Select({

items: [

{label: 'Historical (1950 - 2005)', value: 'historical'},

{label: 'RCP 2.6 (2006 - 2099)', value: 'rcp26'},

{label: 'RCP 4.5 (2006 - 2099)', value: 'rcp45'},

{label: 'RCP 6.0 (2006 - 2099)', value: 'rcp60'},

{label: 'RCP 8.5 (2006 - 2099)', value: 'rcp85'}

],

placeholder: 'Historical (1950 - 2005)',

value: 'historical',

onChange: function(value) {

scenario1 = value;

scenario1Updated = 1;

}

}),

ui.Label({

value:'Scenario 1 Winter Year Range : ',

style: {fontWeight: 'bold'}}), nexdcp30.filters.scenario1YearRange,

ui.Label({

value:'Scenario2 :',

style: {fontWeight: 'bold'}}),

ui.Select({

items: [

{label: 'Historical (1950 - 2005)', value: 'historical'},

{label: 'RCP 2.6 (2006 - 2099)', value: 'rcp26'},

{label: 'RCP 4.5 (2006 - 2099)', value: 'rcp45'},

{label: 'RCP 6.0 (2006 - 2099)', value: 'rcp60'},

{label: 'RCP 8.5 (2006 - 2099)', value: 'rcp85'}

],

placeholder: 'RCP 8.5 (2006 - 2099)',

value: 'rcp85',

onChange: function(value) {

scenario2 = value;

scenario2Updated = 1;

}

}),

ui.Label({

value:'Scenario 2 Winter Year Range : ',

style: {fontWeight: 'bold'}}), nexdcp30.filters.scenario2YearRange,

ui.Panel([

ui.Label({

value:'HUC Watershed Level : ',

style: {fontWeight: 'bold'}

}),

ui.Select({

items: [

{label: 'None', value: 'None'},

{label: '02', value: '02'},

{label: '04', value: '04'},

{label: '06', value: '06'},

{label: '08', value: '08'},

{label: '10', value: '10'},

{label: '12', value: '12'}

],

placeholder: 'None',

onChange: function(value) {

huc = value;

}

})

], ui.Panel.Layout.flow('horizontal')),

nexdcp30.filters.applyButton

]

});

nexdcp30.fww = function(scenario, first) {

//print('Frequency of Warm Winters Calculator');

// Scenario1 years

var years;

if (first == 1) {

years = nexdcp30.filters.scenario1YearRange.getValue().replace(/\s+/g,'').split(':');

} else {

years = nexdcp30.filters.scenario2YearRange.getValue().replace(/\s+/g,'').split(':');

}

var startYear = years[0];

var endYear = years[1];

var nYears = endYear - startYear + 1;

if (first == 1) nYears1 = nYears;

else nYears2 = nYears;

//print(startYear, endYear);

print('Number of years: ', nYears);

// PROCESS FIRST TIME PERIOD

// Filter collection by scenario and GCM

//print(scenario, GCM);

var tasminImageCollection = ee.ImageCollection(collectionPath)

.filter(ee.Filter.eq('scenario', scenario))

.filter(ee.Filter.eq('model', GCM))

.select("tasmin");

var tasmaxImageCollection = ee.ImageCollection(collectionPath)

.filter(ee.Filter.eq('scenario', scenario))

.filter(ee.Filter.eq('model', GCM))

.select("tasmax");

//print(tasminCollection, tasmaxCollection);

var FWW = ee.Image(0);

var sumMask = ee.Image(0);

var blank;

var tas;

var list\_length = 3;

// Loop through years

for (var j = 0; j < nYears; j++) {

var Year0 = parseInt(startYear) + j - 1;

var Year1 = parseInt(Year0) + 1;

var startDate = Year0 + '-12-01';

var endDate = Year1 + '-02-28';

//print('Year:', Year1);

//print(startDate, endDate);

var tasminCollection = ee.ImageCollection(tasminImageCollection).filterDate(startDate, endDate).toList(110);

var tasmaxCollection = ee.ImageCollection(tasmaxImageCollection).filterDate(startDate, endDate).toList(110);

//print(tasminCollection, tasmaxCollection);

//var list\_length = tasminCollection.length().getInfo();

//list\_length = 3;

//print('Collection\_List Length : ' + list\_length);

var sum01 = ee.Image(0);

// Loop through winter months to determine if warm winter occurrs

for (var i = 0; i < list\_length; i++) {

// Get tasmin and tasmax images from collection.

var tasmin = ee.Image(tasminCollection.get(i)).unmask();

var tasmax = ee.Image(tasmaxCollection.get(i)).unmask();

// Calculate average of tasmin and tasmax

tas = (tasmin.add(tasmax)).divide(2);

blank = ee.Image(0);

// Where tas > 273.15 K

var output01 = blank.where(tas.gt(273.15),1);

// Output contains 0s and 1s.

output01 = output01.updateMask(tas);

// Add result to previous month's result

sum01 = sum01.add(output01);

// Find cells where winter tasmin is above threshold to use as mask

// for where snow accumulation is small/rare

blank = ee.Image(0);

var outputMask = blank.where(tasmin.gt(273.15),1);

// Output contains 0s and 1s.

outputMask = outputMask.updateMask(tas);

// Add result to previous month's result

sumMask = sumMask.add(outputMask);

// Find cells where winter tasmin is above threshold to use as mask

// for where snow accumulation is small/rare

} // end of i month loop

//Map.addLayer(sum01, {min: 0, max: 3});

//Map.setCenter(-100, 38, 4);

// Set value to one 1 of value currently > 0

blank = ee.Image(0);

var winter01 = blank.where(sum01.gt(0),1);

winter01 = winter01.updateMask(tas);

//Map.addLayer(winter01, {min: 0, max: 1});

//Map.setCenter(-100, 38, 4);

// Accumulate warm winters

FWW = FWW.add(winter01);

} // End of j year loop

FWW = FWW.divide(nYears/100.0); // Convert from number of winters to percent

blank = ee.Image(0);

sumMask = sumMask.divide(nYears\*3);

if (first == 1) {

//theMask = blank.where(sumMask.lte(0.25),1);

theMask = blank.where(sumMask.lte(0.1),1);

theMask = theMask.updateMask(theMask);

}

//print('Mask: ', theMask);

//Map.addLayer(theMask, {min: 0, max: 1});

//Map.setCenter(-100, 38, 4);

//Map.setCenter(-115, 43, 6);

//var FWW = FWW.divide(nYears);

FWW = FWW.updateMask(tas);

return FWW;

};

/\*\* Creates the application interface. \*/

nexdcp30.boot = function() {

///////////////////////// DATE RANGE //////////////////////////////////

// First, create variables for the date range.

// There are many ways to do this, and I have not figured out the best way yet.

// To change the date range of analysis, simply change the date below.

// Set filter variables.

var main = ui.Panel({

widgets: [

nexdcp30.filters.panel

]

});

ui.root.insert(0, main);

alert('Please select GCM, Scenarios, Year Ranges, and HUC Level, but first\n\n \xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0\xa0Press OK to continue!');

};

nexdcp30.boot();

// Set the default map's cursor to a "crosshair".

Map.style().set('cursor', 'crosshair');