Detecting Land Cover Change in Rangelands (A3.8)

Authors
Ginger Allington, Natalie Kreitzer

Overview
The purpose of this chapter is to familiarize you with the unique challenges of detecting land cover change in arid rangeland systems, and to introduce you to an approach for classifying such change that provides us with a better understanding of these systems. You will learn how to extract meaningful data about changes in vegetation cover from satellite imagery, and how to create a classification based on trajectories over time.

Learning Outcomes
- Visualizing and explaining the challenges of utilizing established land cover data products in arid rangelands.
- Applying a temporal segmentation algorithm to a time series of information about vegetation productivity.
- Classifying pixels based on similarities in their temporal trajectories.
- Extracting and visualizing data on the new trajectory classes.
- Comparing trajectory classes to information from traditional land cover data.

Helps if you know how to:
- Import images and image collections, filter, and visualize (Part F1).
- Perform pixel-based supervised or unsupervised classification (Chap. F2.1).
- Use expressions to perform calculations on image bands (Chap. F3.1).
- Write a function and `map` it over an `ImageCollection` (Chap. F4.0).
- Interpret the outputs from the LandTrendr algorithm implementation in Earth Engine (Chap. F4.5).
- Write a function and `map` it over a `FeatureCollection` (Chap. F5.1, Chap. F5.2).
- Use the `require` function to load code from existing modules (Chap. F6.1).

Introduction to Theory

Arid and semi-arid rangelands cover approximately 41% of the global land surface (Asner et al. 2004) and provide livelihoods for 38% of the human population (Millennium Ecosystem Assessment 2005) and forage for three-quarters of the world's livestock (Derner et al. 2017). Rangelands are located in regions of the world that are experiencing some of the most rapid changes in climate (Huang et al. 2015, Melillo et al. 2014), which can affect ecosystem productivity and resilience (Briske 2017). Land conversion to agriculture (Lambin and Meyfroidt 2014), urbanization and development (Fan et al. 2016, Sleeter et al. 2013), and afforestation (Cao et al. 2011) are increasing in many rangeland regions globally. Uncertainties about socioeconomic and sociopolitical forces such as land tenure security (Campbell et al. 2005, Li et al. 2007, Liu et al. 2015, Reid et al. 2000), rural out-migrations and urbanization (Lang et al. 2013), and changes in human demography and dietary preferences (Alexandratos and Bruinsma 2012) limit our ability to predict the future sustainability of rangeland systems. In order to understand the causes and consequences of land cover change in rangelands, we must first classify and quantify the change.

There are several readily available datasets commonly used to assess changes in land cover over large regions. The three most prominent global inventories are the annual MODIS 12Q land cover product (500 m) (Friedl et al. 2010), the 300 m GlobCover data from the European Space Agency (ESA) (Arino et al. 2008, Bontemps et al. 2011), and the new 10 m WorldCover data, also from the ESA (Zanaga et al. 2020), the latter two of which are available for single nominal dates. National-level data products typically are available at finer spatial resolutions and tend to use more categories to classify the surface (e.g., NLCD in the US, Homer et al. 2015)-- though they are not available for all countries. While these global and national land cover data products have been useful for documenting a number of important surface phenomena such as forest loss, urbanization, and habitat conversion in a variety of ecosystems (Schneider et al. 2010, Presetele et al. 2016), they generally have poor accuracy in arid portions of the globe (Friedl et al. 2002, Ganguly et al. 2010, García-Mora et al. 2012).

Additionally, traditional methods for change detection are derived from categorical land cover classifications, which can identify change only when a pixel crosses a threshold to a new state (e.g., from Grassland to Barren), and therefore can identify change only after it has occurred, not when it begins. In systems with long time lags in vegetation response (such as has been shown for recovery in arid grasslands), this can create uncertainty about the efficacy of environmental policies and the response of landscapes to large-scale management changes (Fig. A3.8.1).

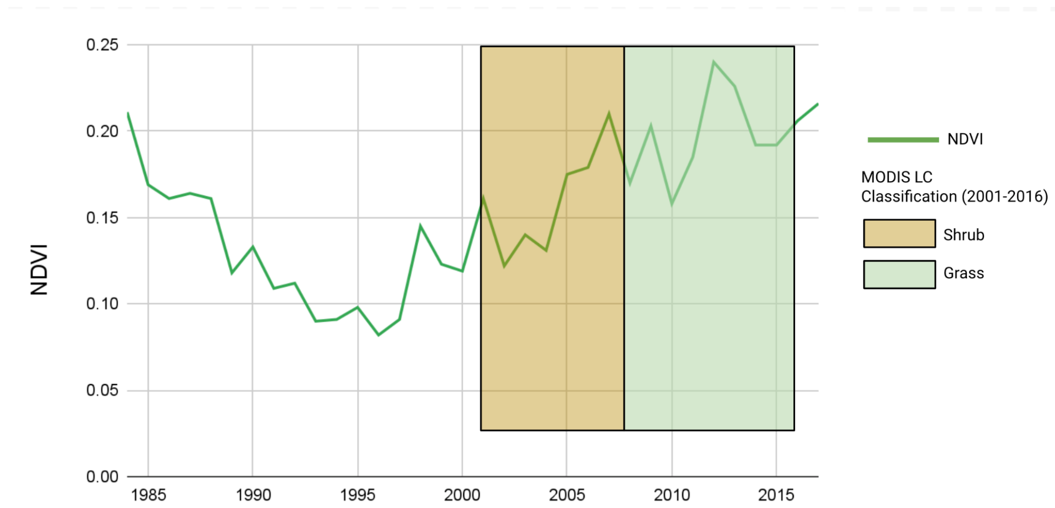

Fig. A3.8.1 Time series of NDVI in Naiman Banner, Inner Mongolia. Line represents the median value calculated over the region. Overlaid colored bars represent the corresponding land cover classes for the years 2001–2016, according to the MODIS-derived MCD12Q1 data.

As an alternative to classification-based change detection, new methods have emerged that identify unique features in time series of remotely sensed data to pinpoint disturbance events such as logging, wildfires, and flooding. These time series-based methods, such as CCDC (see Chap. F4.7) and LandTrendr (see Chap. F4.5), are typically calibrated and executed to detect abrupt changes in spectral reflectance or derived index values associated with those disturbances. This is extremely useful for detecting change events like deforestation or the defoliation resulting from an insect outbreak where the disturbance is punctuated in time and where the difference in spectral index is significant, and regreening occurs on the order of years (Fig. A3.8.2).

For these reasons, contemporary classification approaches are extremely limited in their ability to provide reliable information about landscape change and dynamics in rangeland systems. Current pressures from climate change, land use intensification, and urban expansion create an urgent need for methods to more accurately map and track land cover status in a way that is useful for arid-systems research, land use change detection, and the monitoring and management of rangelands.

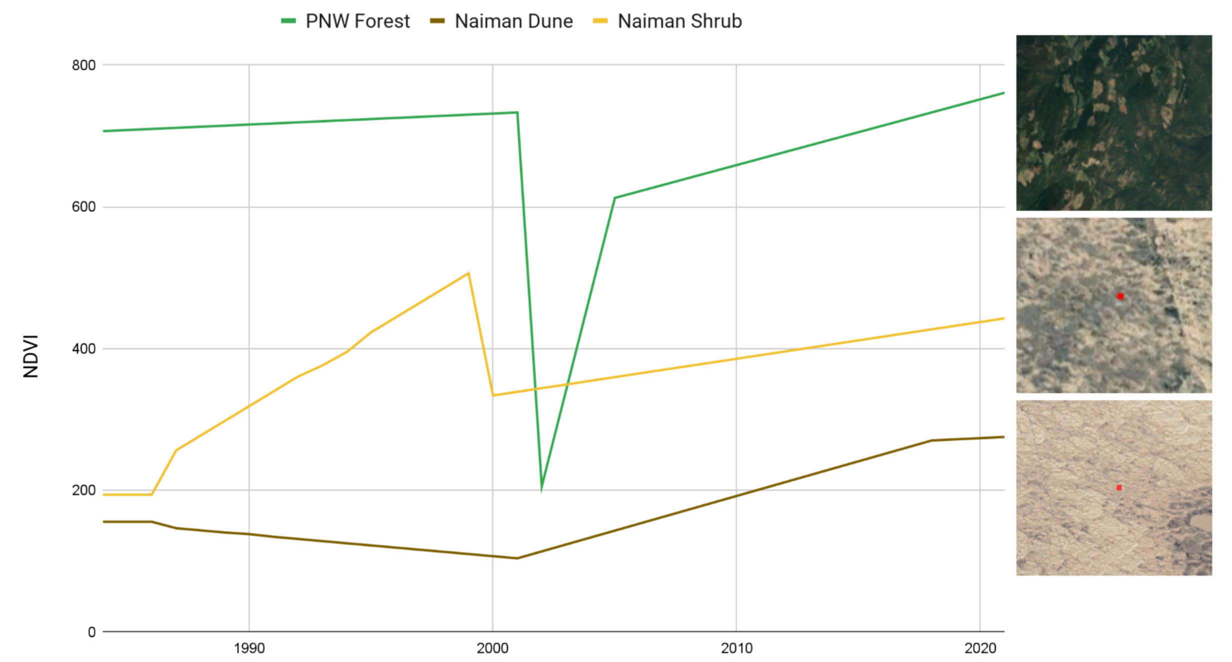

Fig. A3.8.2 Comparison of predicted NDVI for representative pixels from three regions: a forested area in southern British Columbia and two regions in Naiman Banner, Inner Mongolia, one dominated by woody shrubs and perennial forbs, the other by annual grasses and forbs

Forest land cover changes are stark in terms of the change in indices such as NDVI and NBR (Kennedy et al. 2016). A forest pixel might suddenly drop from an NDVI over 0.6 to 0.15, which is easily detectable and a relatively straightforward transition to select for when coding a to-from change detection algorithm. Additionally, the visual change is stark, so a visual interpreter can easily create a training dataset without ancillary information to identify conversion from forest to burned, cleared, or developed.

In contrast, vegetation degradation and recovery in rangelands are typically characterized by more gradual changes over longer periods of time. Unlike with forest change, a trained interpreter using only satellite imagery cannot visually identify a rangeland land cover class transition until years after it has begun, if at all.

However, there is potential to utilize the information generated from temporal segmentation to characterize other aspects of change. Here, we will employ the LandTrendr time series segmentation algorithm (as described in Chap. F4.5) to derive information about how pixels are changing over time, and use that to generate a new land cover classification for a region of northern China.

To truly understand how rangelands are changing in space and time, and to provide adequate monitoring to support sustainable rangeland management, we need tools to help us capture and quantify those changes at landscape scales. Further, we need new ways of measuring changes within land cover types and interpreting those changes in ways that reflect rangeland dynamics and ecological processes. This necessitates rethinking the framework we use to categorize these lands in the first place. Approaches making use of greater temporal information on forest (Healey et al. 2018) and wetland systems (Dronova et al. 2015) show promise for deployment on rangeland systems.

We present a hybrid approach to rangeland classification that categorizes observable dynamic patterns in vegetation cover to classify pixels based on similarities in trajectory history. These new classes reveal much more meaningful information about the current status of a given pixel than a class based on cover type, and also yield information about the potential of a given pixel to respond to stressors in the future.

Practicum

Section 1. Inspecting Information about the Study Area

If you have not already done so, you can add the book's code repository to the Code Editor by entering https://code.earthengine.google.com/?accept_repo=projects/gee-edu/book (or the short URL bit.ly/EEFA-repo) into your browser. The book's scripts will then be available in the script manager panel to view, run, or modify. If you have trouble finding the repo, you can visit bit.ly/EEFA-repo-help for help.

In this section, we will load and explore data sets that illustrate the area, and form the basis for the analysis.

Section 1.1 Inspect the Study Area

Naiman Banner (Fig. A3.8.3) is located in the southeastern portion of the Inner Mongolia Autonomous Region of northern China. Historically, this region was occupied by ethnic Mongolian pastoralists, and the primary vegetation was perennial grasses with some shrub cover. The region has undergone significant intensification of land uses over the past 60 years. Heavy grazing and conversion to crops has removed vegetation, exposing soils to erosion and resulting in extensive desertification. Since the early 1990s, a series of environmental policies and restoration programs, including grazing restriction and afforestation, have been implemented to halt the spread of desertification and promote revegetation of the rangelands. At the same time, cropland has continued to expand, and the use of irrigation has eliminated almost all of the surface water sources and severely lowered the groundwater table.

In this exercise, we will focus on a portion of central Naiman Banner that spans from the extensive Horqin Sandy Lands in the west to the dense agricultural area in the east.

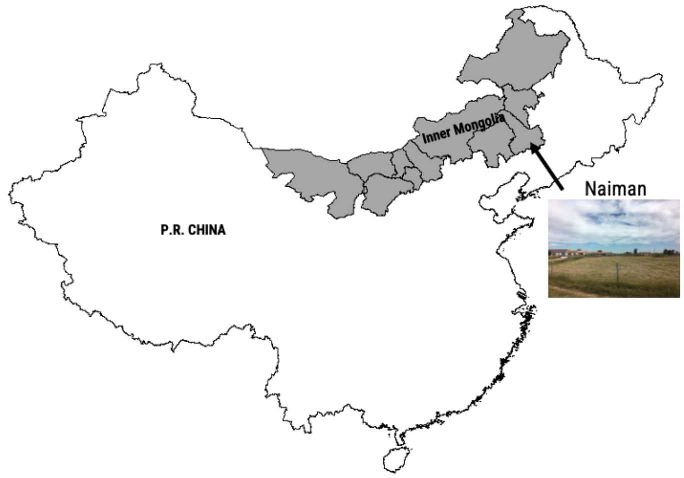

Fig. A3.8.3 Location of the Horqin Sandy Lands within Naiman Banner, Inner Mongolia, People's Republic of China

Our first step is to load a shapefile for our area of interest (AOI) and explore the landscape using the default satellite basemap.

```
// Load the shapefile asset for the AOI as a Feature Collection
var aoi = ee.FeatureCollection(
    'projects/gee-book/assets/A3-8/GEE_Ch_AOI');
Map.centerObject(aoi, 11);
Map.addLayer(aoi, {}, 'Subset of Naiman Banner');
```

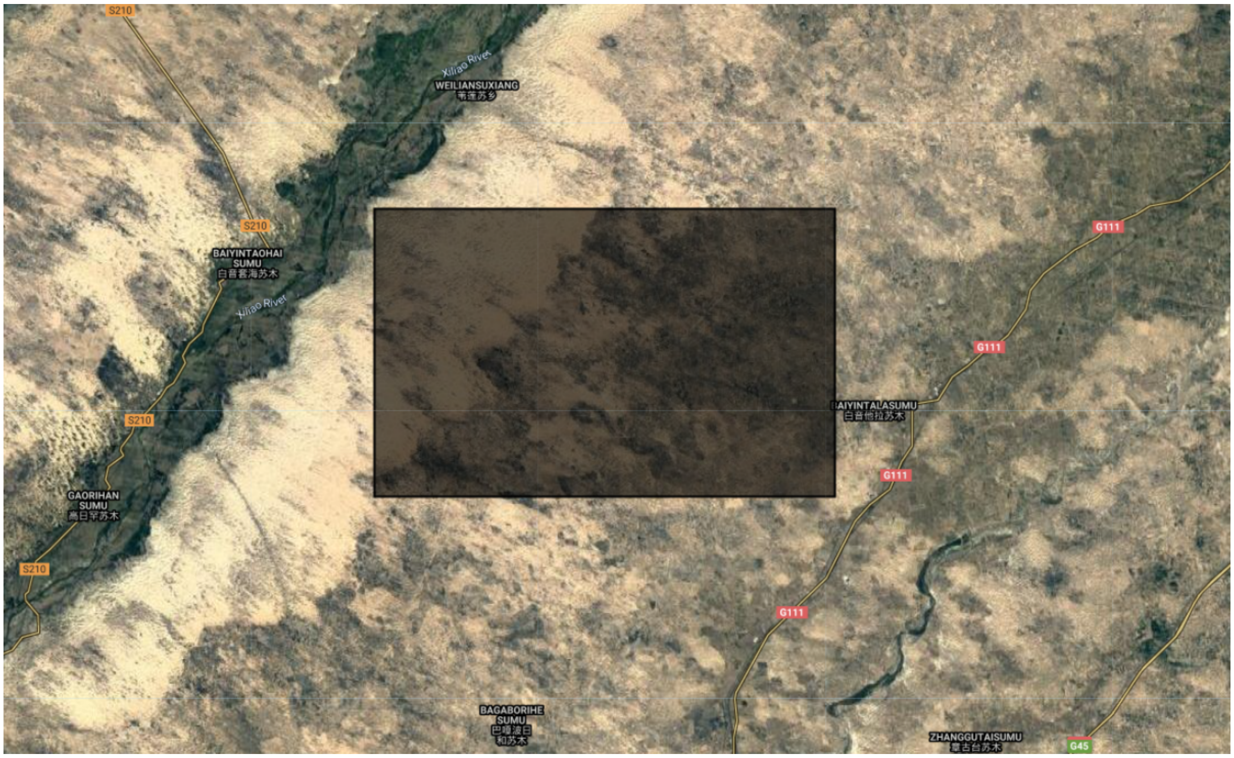

Fig. A3.8.4 Location of the study region, within the Horqin Sandy Lands

Switch the basemap to **Satellite** (Fig. A3.8.4). Turn the layer with the AOI boundary on and off and pan around the study area. Inspect the difference in land uses and cover types between the far western and eastern edges of the AOI. Mark them with pointer markers so you can revisit them later.

Question 1. List four potential land use or land cover classes that you observe in the image.

Section 1.2 Inspect Existing Land Use Data

Next, we will explore two different sources of land cover data to get a sense of how this landscape is typically categorized by these kinds of classified products.

First, we will explore the MODIS MCD12Q1 Land Cover Type dataset. These global layers are available annually from 2001 to the present, at a resolution of 500 m. We will filter and visualize data for 2001, 2009, and 2016 in order to compare how MODIS captures change over this period.

Next, we will define and execute several different functions in this lesson. These functions will help us to execute the same logic across multiple images within different image collections.

```
// Filter the MODIS Collection
var MODIS_LC = ee.ImageCollection('MODIS/006/MCD12Q1').select(
    'LC_Type1');

// Function to clip an image from the collection and set the year
var clipCol = function(img) {
    var date = ee.String(img.get('system:index'));
    date = date.slice(0, 4);
    return img.select('LC_Type1').clip(aoi) // .clip(aoi)
        .set('year', date);
};

// Generate images for diff years you want to compare
var modis01 = MODIS_LC.filterDate('2001-01-01', '2002-01-01').map(
    clipCol);
var modis09 = MODIS_LC.filterDate('2009-01-01', '2010-01-01').map(
    clipCol);
var modis16 = MODIS_LC.filterDate('2016-01-01', '2017-01-01').map(
    clipCol);
// Create an Image for each of the years
var modis01 = modis01.first();
var modis09 = modis09.first();
var modis16 = modis16.first();
```

Now that we have loaded all three datasets, let's take a look at them and see how they compare. The `randomVisualizer` code below assigns random colors to each class in each layer.

```
Map.addLayer(modis01.randomVisualizer(), {}, 'modis 2001', false);
Map.addLayer(modis09.randomVisualizer(), {}, 'modis 2009', false);
Map.addLayer(modis16.randomVisualizer(), {}, 'modis 2016', false);
```

You should end up with something that looks like Fig. A3.8.5. (The colors assigned to classes in your map may differ.)

Fig. A3.8.5 Land cover classes identified in 2001 from MODIS MCD12Q1

Use the slider bar in the **Layer** menu to turn the data on and off to compare classifications across the three years. Use the **Inspector tool** to select a few pixels around the AOI to pull information on the specific classes. Compare the pixel values reported in the **Inspector** to the land cover codes and descriptions listed in Table A3.8.1.

Table A3.8.1 Subset of the 17 IGBP Classes that appear within the study area in the MODIS MCD12Q1 land cover dataset.

Value	Description
7	Open Shrublands: dominated by woody perennials (1-2m height) 10-60% cover.
10	Grasslands: dominated by herbaceous annuals (<2m).
12	Croplands: at least 60% of area is cultivated cropland.
16	Barren: at least 60% of area is non-vegetated barren (sand, rock, soil) areas with less than 10% vegetation.

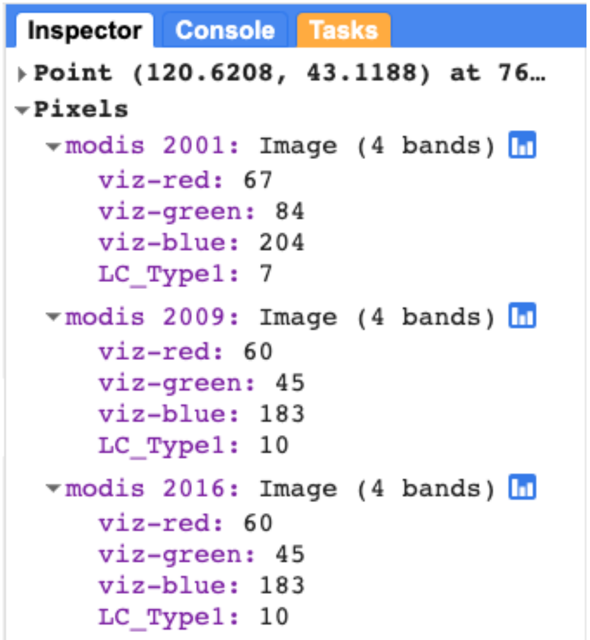

Fig. A3.8.6 Land cover classes assigned to a single pixel in the study area in 2001, 2009, and 2016 from MODIS MCD12Q1

Question 2. What are the four land cover types identified in this region, as classified by the MODIS data? Be sure to check all three time periods.

Next, we will add the WorldCover dataset from the ESA. This dataset was generated for 2020 only, and has a resolution of 10 m.

```
// Add and clip the WorldCover data
var wCov = ee.ImageCollection('ESA/WorldCover/v100').first();
var landcover20 = wCov.clip(aoi);
Map.addLayer(landcover20, {}, 'Landcover 2020');
```

The WorldCover dataset includes palette information in the `'Map'` band, so you should end up with something that looks like Fig. A3.8.7. (The colors assigned to classes in your map may differ.)

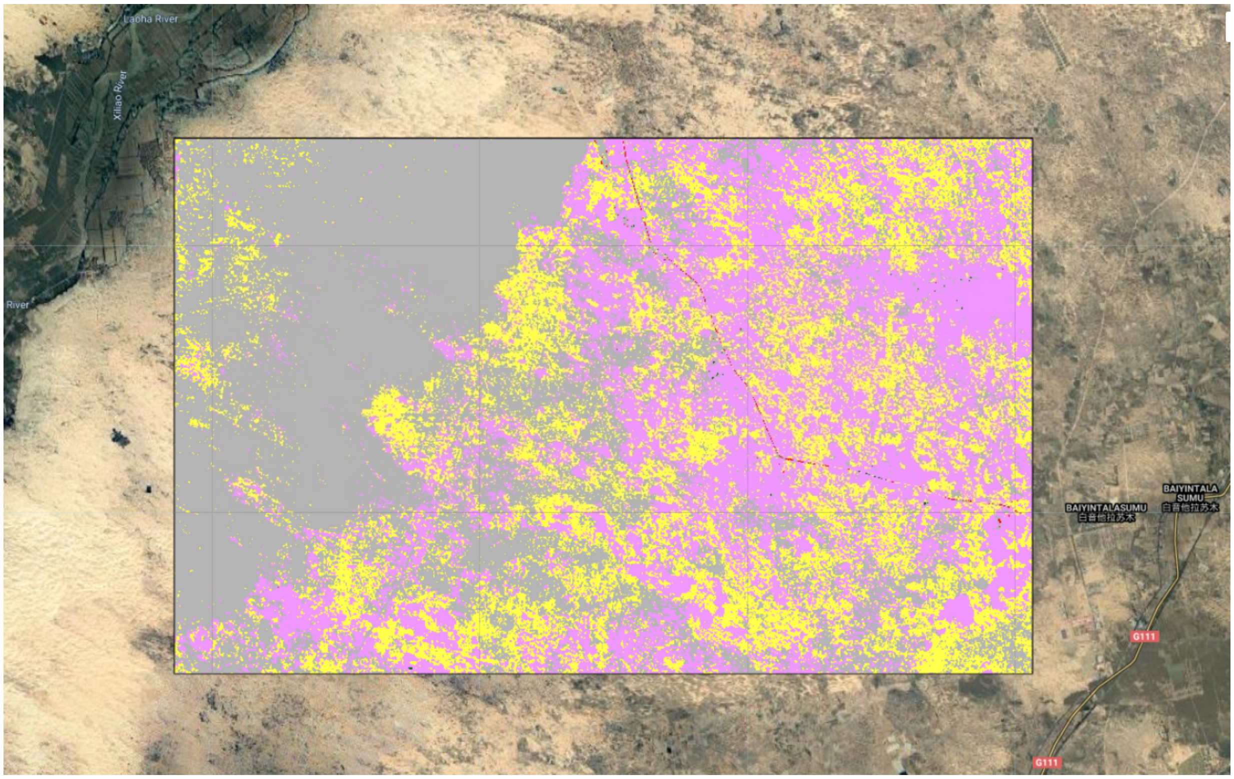

Fig. A3.8.7 Land cover classes identified in 2020 in the ESA WorldCover dataset

Table A3.8.2 Subset of the 11 land cover classes that appear within the study area in the ESA WorldCover dataset

Value	Description
10	Trees
30	Grassland
40	Cropland
50	Built-up
60	Barren / sparse vegetation

Code Checkpoint A38a. The book's repository contains a script that shows what your code should look like at this point.

Question 3. Spend a few minutes turning the different land cover layers on and off and using the Inspector to identify the classes for different pixels. How do the MODIS and WorldCover datasets differ? In what ways are they similar?

Question 4. Return to the points that you flagged in Section 1. Do your estimations of land cover correspond to the classifications from the two different data sources?

Question 5. Qualitatively compare change over time according to the MODIS data. What do you observe from these data? Approximately how much of the AOI has changed between 2001 and 2020, according to these data? Where are the regions of changing classification located? What regions seem to be fairly stable?

Section 2. Compile the Time Series of Vegetation Cover

The Normalized Difference Vegetation Index (NDVI) is an index of vegetative productivity derived from the relationship between the red (approx. 650 nm) spectra and the near-infrared (750–2500 nm) spectra. NDVI is a consistently good proxy for productivity in this region (de Beurs et al. 2015). It is also highly correlated with precipitation (John et al. 2016). This relationship has the potential to introduce short-term responses of increased vegetation productivity that could be falsely identified as land cover change (Fig. A3.8.8 a and b). In order to account for this effect, we need to remove the main effect of precipitation on NDVI for a given year so that we can assess changes in greenness outside of that variability. To do this, we will derive individual regression models for each pixel (as detailed in Chap. F4.6) for the relationship between total annual water-year precipitation and maximum greenness in each year. We will then predict greenness for each pixel in each year based on observed precipitation, and use the residual values of greenness from the predicted model as an input to LandTrendr. We can think of the residuals as the remaining annual primary productivity, after we have removed the effect of precipitation (Fig. A3.8.8 c).

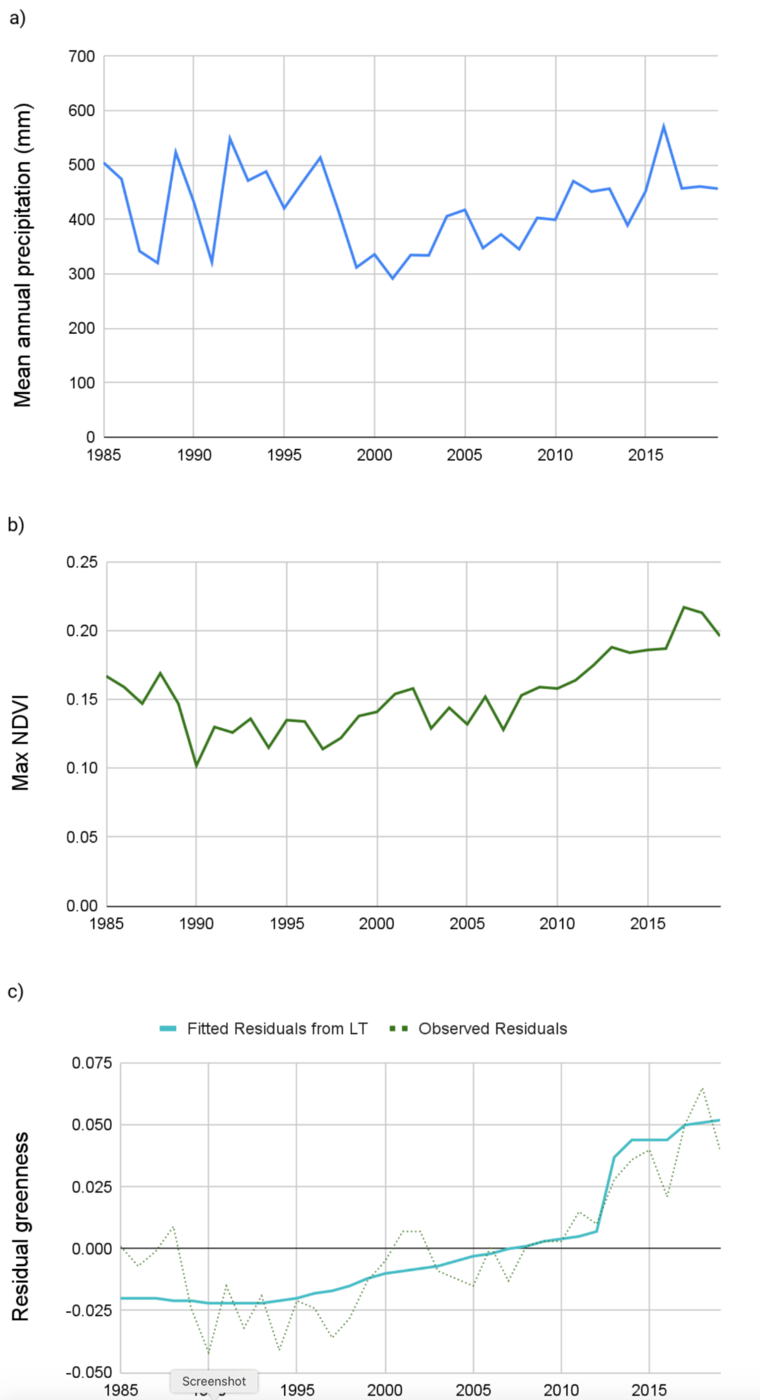

Fig. A3.8.8 Comparison of the variation over time in a) mean annual precipitation and b) maximum NDVI for the AOI; and a time series of residual greenness from a regression of precipitation and NDVI.

Before we go any further, let's add two pre-generated image collection assets for greenness and precipitation (`greennessColl` and `precipColl`) and explore them. Print each of them to the **Console** to inspect the contents (Fig. A3.8.9).

```
var greennessColl = ee.ImageCollection(
    'projects/gee-book/assets/A3-8/GreennessCollection_aoi');
var precipColl = ee.ImageCollection(
    'projects/gee-book/assets/A3-8/PrecipCollection');
print(greennessColl, 'Greenness Image Collection');
print(precipColl, 'Precip Image Collection');
```

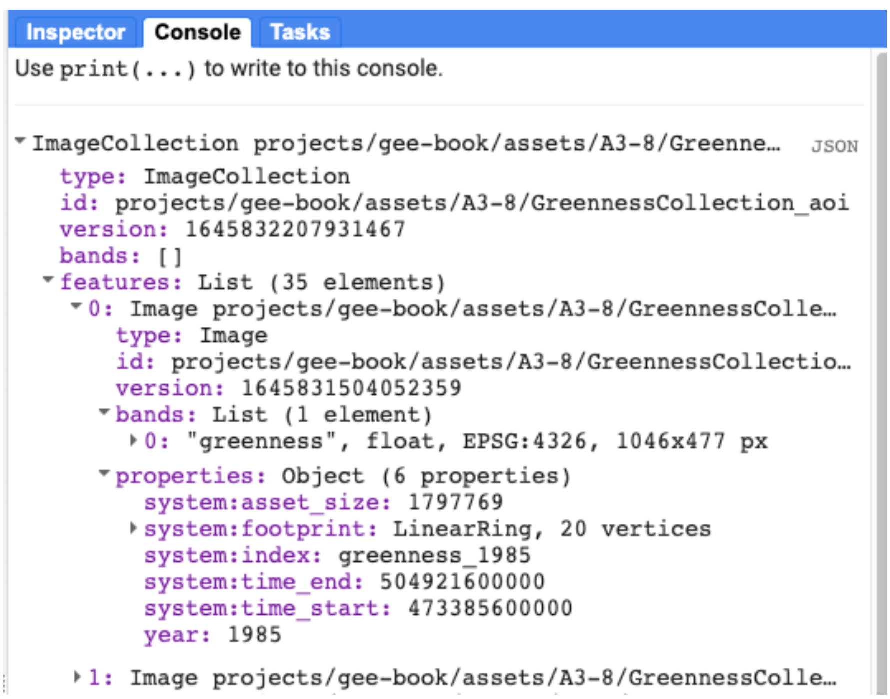

Fig. A3.8.9 Properties of the pre-generated Image Collection of maximum annual greenness values, generated from NDVI

We saw in the plots above (Figs. A3.8.1 and A3.8.2) how NDVI can vary significantly over time in this region. We can also select out a few years to visualize how NDVI ("greenness") varies spatially as well.

```
var greennessParams = {
    bands: ['greenness'],
    max: 0.5,
```

```
    min: 0.06,
    opacity: 1,
    palette: ['e70808', 'ffffff', '1de22c']
};

var greenness1985 = greennessColl.filterDate('1985-01-01',
    '1986-01-01').select('greenness');
var greenness1999 = greennessColl.filterDate('1999-01-01',
    '2000-01-01').select('greenness');

print(greenness1999);
var greenness2019 = greennessColl.filterDate('2019-01-01',
    '2020-01-01').select('greenness');

Map.addLayer(greenness1985, greennessParams, 'Greenness 1985', false);
Map.addLayer(greenness1999, greennessParams, 'Greenness 1999', false);
Map.addLayer(greenness2019, greennessParams, 'Greenness 2019', false);
```

Turn the layers for the different years on and off to compare the range and spatial distribution of NDVI (Fig. A3.8.10).

Question 6. What do you observe about the similarities and differences in the distribution of NDVI across the selected years?

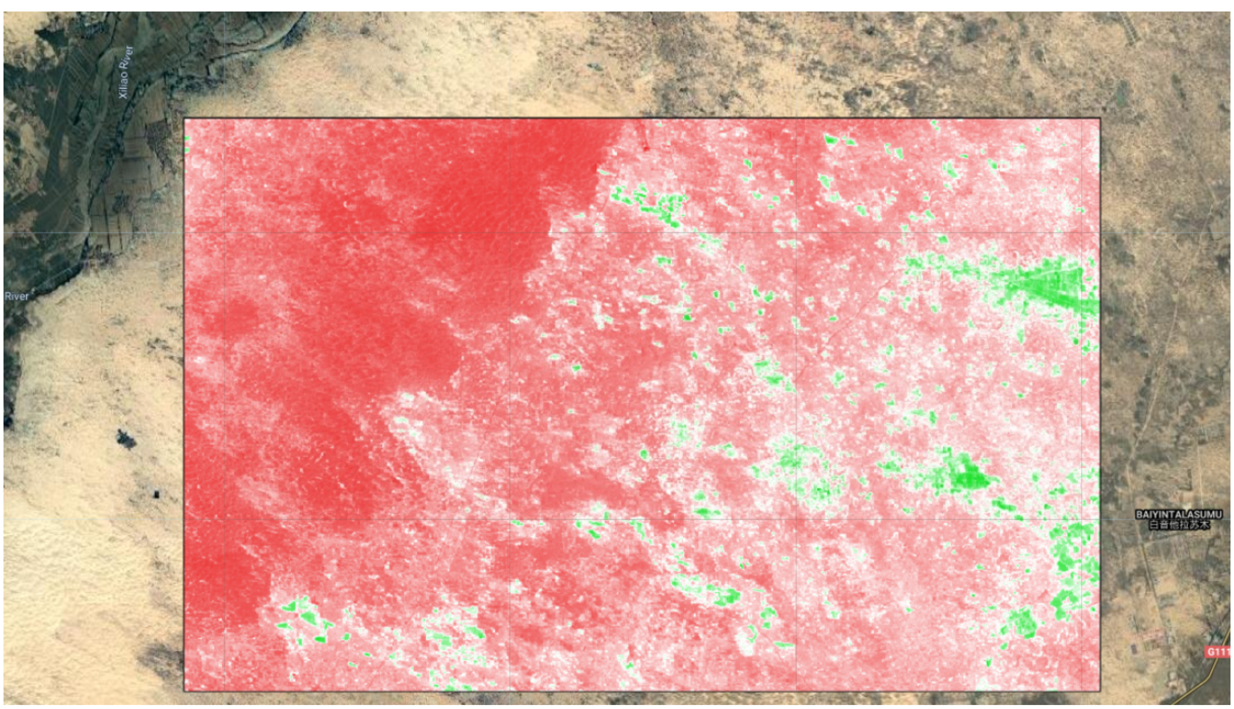

Fig. A3.8.10 Observed greenness (NDVI) in the study region in 2019. The values range from a minimum of 0.07 to a maximum of 0.5.

Combining the greenness and precipitation collections and calculating the model to generate residuals is a relatively long process. To speed things along, we will employ a function that has been defined in a script module called `residFunctions`.

```
// Load a function that will combine the Precipitation and Greenness
collections, run a regression, then predict NDVI and calculate the
residuals.

// Load the module
var residFunctions = require(
    'projects/gee-edu/book:Part A - Applications/A3 - Terrestrial
Applications/A3.8 Detecting Land Cover Change in
Rangelands/modules/calcResid'
);

// Call the function we want that is in that module
// It requires three input parameters:
// the greenness collection, the precipitation collection and the aoi
var residualColl = (residFunctions.createResidColl(greennessColl,
    precipColl, aoi));

// Now inspect what you have generated:
print('Module output of residuals', residualColl);
```

Print the resulting `ImageCollection` and inspect the bands and properties in the **Console** (Fig. A3.8.11).

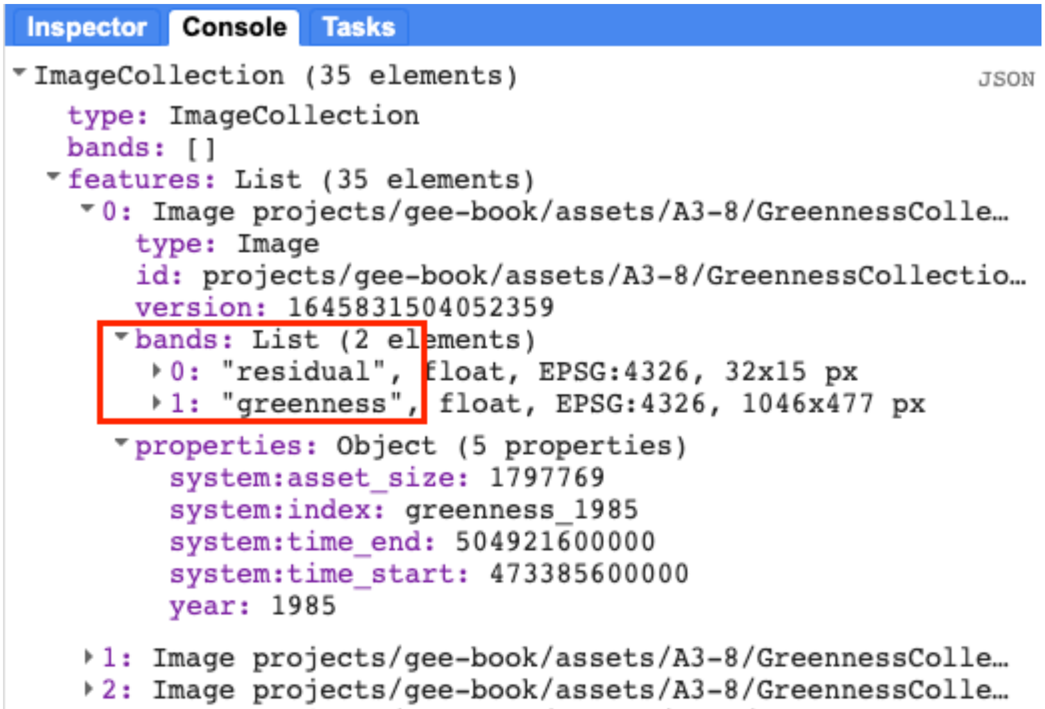

Fig. A3.8.11 Each image in the residualColl image collection contains two bands, residual and greenness. The residual band for will be passed to LandTrendr.

Next, you will filter the `residualColl` collection to map the same years we explored for the observed NDVI (greenness). The code chunk below will pull the image for 1985 (the first year). Use `filterDate` to select the other years you want to view. Add each to the map as new layers.

```
var resids = residualColl.first();
var res1 = resids.select(['residual']);
print(res1.getInfo(), 'residual image');
Map.addLayer(res1, {
    min: -0.2,
    max: 0.2,
    palette: ['red', 'white', 'green']
}, 'residuals 1985', false);
```

Map and compare the residuals versus greenness for a few different years (Fig. A3.8.12).

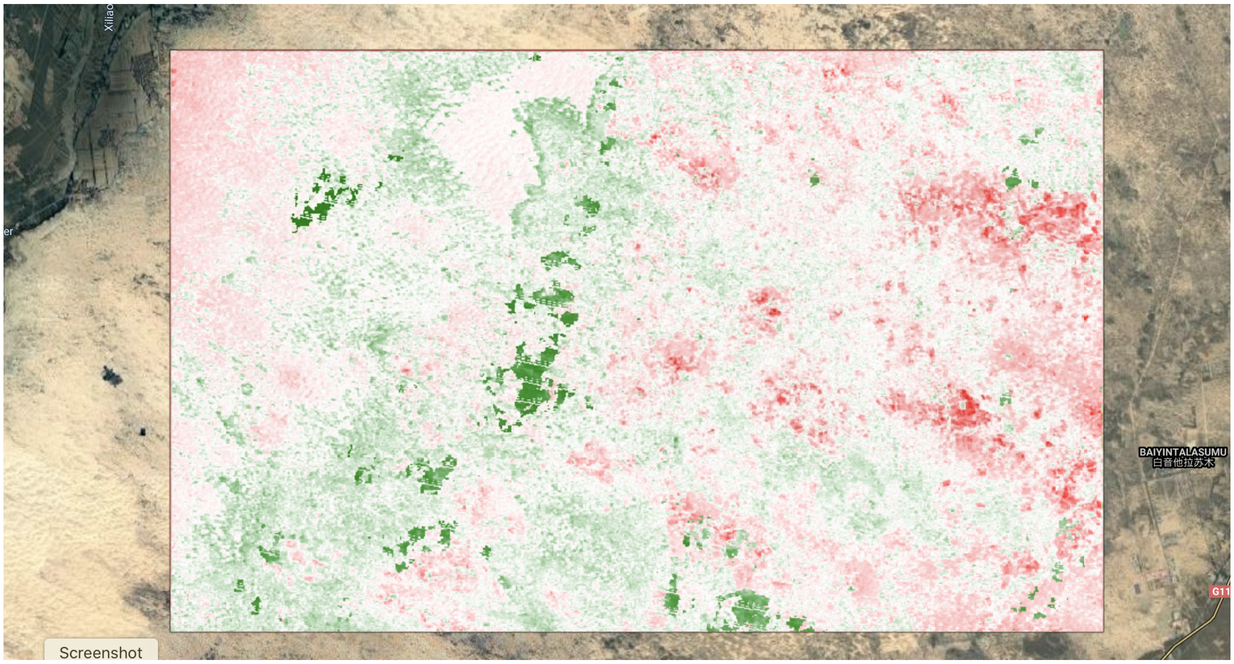

Fig. A3.8.12 Residual greenness in 1985, after removing the effect of precipitation. Models were fit on a per-pixel basis.

Code Checkpoint A38b. The book's repository contains a script that shows what your code should look like at this point.

Question 7. Compare the layers for residual greenness to the observed greenness values from Question 6. Why are we passing residuals to LandTrendr rather than the beginning NDVI values?

Section 3. Time Series Segmentation

Now you are ready to apply the LandTrendr time series segmentation algorithm to your Image Collection of residual greenness for the years 1985-2019. This will generate information for each pixel in the study area about how it has changed over time. In order to execute the code, you need to first define pertinent parameters in a dictionary, which you will provide to the LandTrendr algorithm along with your data. Chap. F4.5 gives an explanation of the LandTrendr algorithm and its parameters. That chapter showed an interface for LandTrendr; below, we utilize JavaScript functions to execute LandTrendr functions directly in the code editor.

```
//---- DEFINE RUN PARAMETERS---//
// LandTrendr run parameters
```

```
var runParams = {
    maxSegments: 6,
    spikeThreshold: 0.9, //
    vertexCountOvershoot: 3,
    preventOneYearRecovery: true,
    recoveryThreshold: 0.25, //
    pvalThreshold: 0.05, //
    bestModelProportion: 0.75,
    minObservationsNeeded: 10 //
};
```

Follow the next steps to combine the dictionary of parameter settings with the image collection and apply LandTrendr with the API functionality `ee.Algorithms.TemporalSegmentation.LandTrendr`. Then print and explore the output.

```
// Append the image collection to the LandTrendr run parameter
dictionary
var srCollection = residualColl;
runParams.timeSeries = srCollection;

// Run LandTrendr
var lt = ee.Algorithms.TemporalSegmentation.LandTrendr(runParams);
// Explore the output from running LT
var ltlt = lt.select('LandTrendr');
print(ltlt);
```

The implementation of the LandTrendr Algorithm in GEE generates a multidimensional array containing subarrays for the observation year, observed residual, fitted residual, and Boolean vertex layer that tells the user if a change in pixel trajectory occured in a given observation year. In order to analyze the outputs year over year, we first must slice out the subarrays of the output multidimensional array, and transform them into Image collections. We will not go into detail about slicing arrays in this lesson. For your purposes here, you can just follow along with the provided code. If you wish to learn more about array indexing and slicing, you can refer back to Chap. F4.6).

```
//---- SLICING OUT DATA -----------------//

// Select the LandTrendr band.
```