

---

# Water Balance and Drought (A2.5)

---

## Authors

Ate Poortinga, Quyen Nguyen, Nyein Soe Thwal, Andréa Puzzi Nicolau

---

## Overview

In this chapter, you will learn simple water balance calculations using remote-sensing-derived products related to precipitation and evapotranspiration. You will work at the river basin scale and perform time-series analysis, while comparing the data series with remote sensing vegetation and drought indices using the Earth Engine platform. You will also overlay the various indices with a land cover map to estimate potential drought impacts throughout the region.

## Learning Outcomes

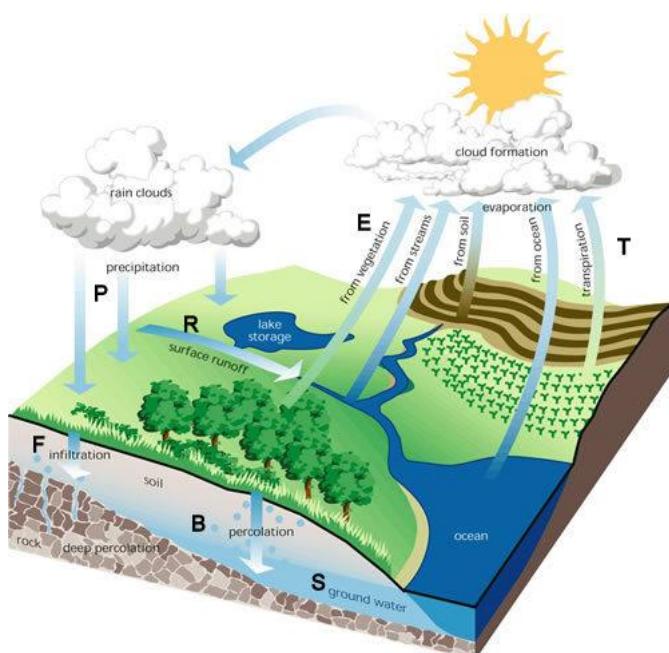
- Understanding the basics of remote-sensing-derived precipitation and evapotranspiration products.
- Calculating monthly aggregate statistics.
- Performing time-series analysis.
- Calculating vegetation and drought indices.

## Helps if you know how to:

- Import images and image collections, filter, and visualize (Part F1).
- Create a graph using `ui.Chart` (Chap. F1.3).
- Perform basic image analysis: select bands, compute indices, create masks (Part F2).
- Write a function and map it over an ImageCollection (Chap. F4.0).
- Summarize an ImageCollection with reducers (Chap. F4.0, Chap. F4.1).
- Aggregate data to build a time series (Chap. F4.2).
- Work with CHIRPS rainfall data (Chap. F4.2)
- Mask cloud, cloud shadow, snow/ice, and other undesired pixels (Chap. F4.3).

## Introduction to Theory

Water is vital for sustaining human life, ensuring food security, generating power, and supporting industrial processes in river basins. Both terrestrial and aquatic ecosystems are dependent on water to provide valuable ecosystem services, not only for the current generation but also for generations in the future. Managing the complex flow paths of water to and from these different water-use sectors requires a quantitative understanding of hydrological processes. Quantitative insights are necessary to help manage water consumption more efficiently by means of retention, withdrawals, and land use change. Managers need background data to help them optimize water allocation to sectors without further depleting the natural capital in the basin.

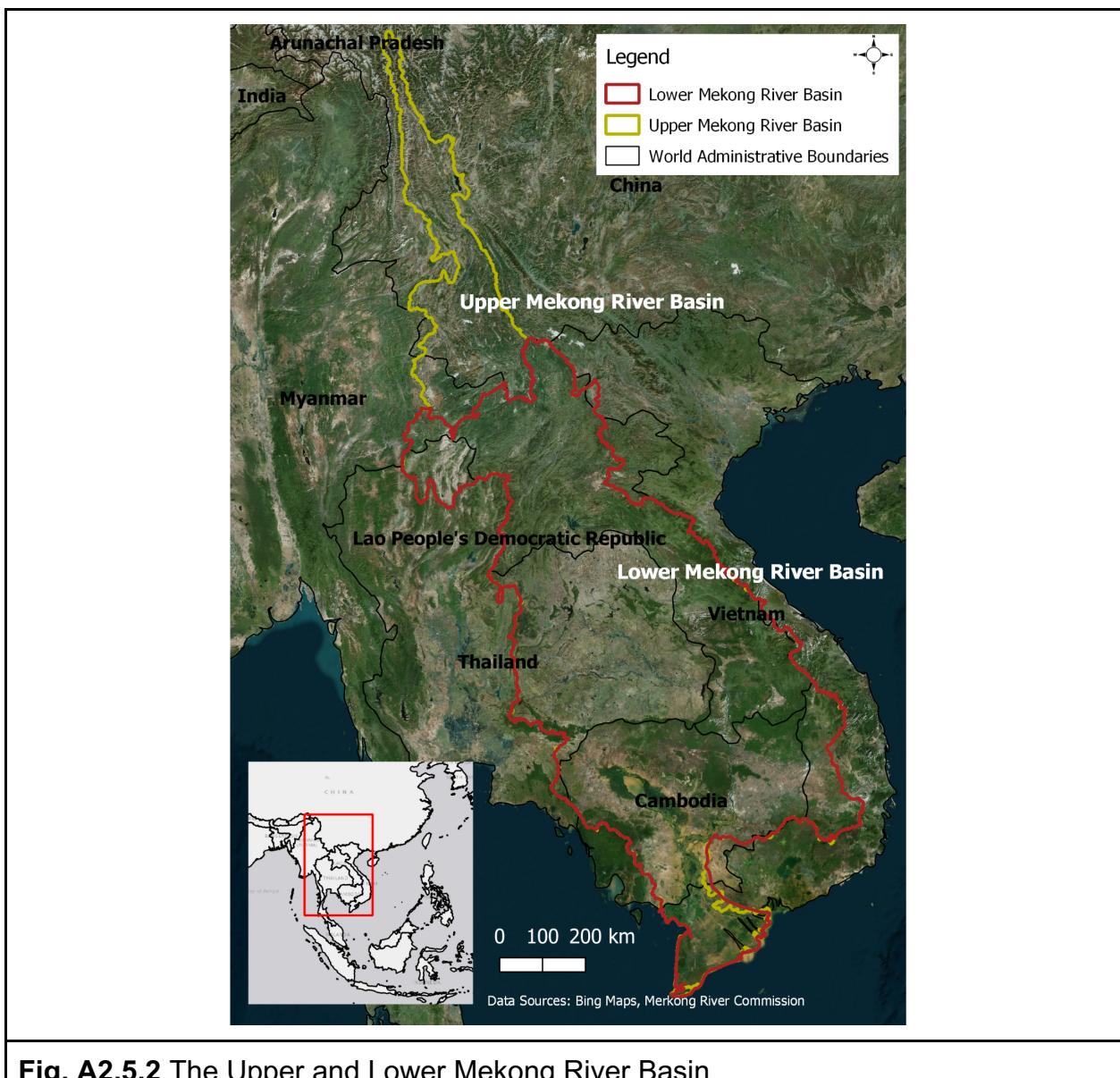


**Fig. A2.5.1** The key components of the hydrological cycle

The water balance (Fig. A2.5.1) is the key concept in understanding the availability of water resources in a hydrological system. The water balance includes both input and extractions of water. In its simplest form, the water balance can be defined as Equation A2.5.1. Inputs to the hydrological system are defined by precipitation (P; rainfall and snow). Extractions for the system are from runoff (Q) and evapotranspiration (ET), with evapotranspiration denoting the sum of evaporation from the land surface plus transpiration from plants. Water balance changes in groundwater and soil storage are indicated by  $\Delta S$ .

$$P = Q + ET + \Delta S \quad (\text{A2.5.1})$$

A hydrological system, also referred to as river basin or drainage basin, is any land area where precipitation collects and drains off into a common outlet. The hydrological processes between upstream and downstream are interconnected: For example, extractions of water resources upstream will impact the amount of available downstream water resources. Similarly, upstream activities such as logging and swidden agriculture might impact the quality of downstream water resources. Fig. A2.5.2 shows the boundaries of the Lower Mekong River Basin, which covers parts of Laos, Thailand, Cambodia, Myanmar, and Vietnam. Water is a shared resource among the countries, although each country has its own legal framework to maintain and protect water supplies. A quantitative understanding of this shared resource is imperative to formulating effective management strategies.



In the following exercises, we will calculate the main components of the water balance and link them with vegetation growth, drought information, and land cover information in the Lower Mekong Basin.

## Practicum

### **Section 1. Calculating Monthly Precipitation**

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](https://code.earthengine.google.com/?accept_repo=projects/gee-edu/book) (or the short URL [bit.ly/EEFA-repo](http://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](http://bit.ly/EEFA-repo-help) for help.

Precipitation has been measured for many centuries (Strangeways 2010). The traditional method is point measurement, which was standardized in the previous century to make measurements comparable in space and time. Fig. A2.5.3 shows a conventional weather station used to measure various weather-related parameters, including the amount of rainfall. Although statistical methods exist to calculate area averaged rainfall from weather stations, the limited number of data points remains a constraint, especially in developing countries and sparsely populated regions where the density of weather stations is low. Satellites can fill this information gap, as they observe the planet at a regular interval with calibrated sensors.



**Fig. A2.5.3** Conventional weather station that measures various parameters, including precipitation

The Tropical Rainfall Measuring Mission (TRMM), a joint mission of the Japan Aerospace Exploration Agency (JAXA) and NASA, was a notable effort to monitor and study tropical rainfall (Kummerow et al. 1998). The satellite, which operated for 17 years, contained various instruments to measure clouds and cloud structures in order to advance understanding of the global energy and water cycles. The Global Precipitation Measurement (GPM; Fig. A2.5.4) mission is the successor, with the primary aim of making frequent (every 2–3 hours) observations of Earth's precipitation (Hou et al. 2014). A wide variety of data products are available for both TRMM and GPM. Precipitation estimates are derived from the Precipitation Radar (PR), TRMM Microwave Imager (TMI), Visible Infrared Scanner (VIRS), Clouds and Earth's Radiant Energy System (CERES), and Lightning Imaging Sensor (LSI). Frequently used products of TRMM have a spatial resolution of 0.25 degrees (~25 km), whereas GPM has a higher resolution of 0.1 degrees (~10 km). Data is available on a three-hour time interval. The data can be obtained through NASA but also can be accessed from the Earth Engine data repository.

The Climate Hazards Group InfraRed Precipitation with Station (CHIRPS) data is a quasi-global rainfall dataset (Funk et al. 2015) covering more than 35 years. CHIRPS provides precipitation information at a 0.05 degrees (~5 km) spatial resolution. The dataset estimates precipitation by combining data from observing meteorological stations with satellite data. CHIRPS data is available at intervals from daily to annual and can be very valuable in hydrology studies, as it provides a long and consistent time series with precipitation estimates at a relatively high spatial resolution.



**Fig. A2.5.4** The satellite for the Global Precipitation Measurement (GPM) mission

Below is the code and an exercise to calculate monthly precipitation using these satellite data.

We begin by importing our area of interest, the Lower Mekong River Basin (Fig. A2.5.5).

```
// Import the Lower Mekong boundary.  
var mekongBasin = ee.FeatureCollection(  
    'projects/gee-book/assets/A2-5/lowerMekongBasin');  
  
// Center the map.  
Map.centerObject(mekongBasin, 5);  
  
// Add the Lower Mekong Basin boundary to the map.  
Map.addLayer(mekongBasin, {}, 'Lower Mekong basin');
```



**Fig. A2.5.5** The Lower Mekong Basin

In the next step, we set the start and end dates for our analysis. We create a list for both years and months, which we will later use to iterate over.

```

// Set start and end years.
var startYear = 2010;
var endYear = 2020;

// Create two date objects for start and end years.
var startDate = ee.Date.fromYMD(startYear, 1, 1);
var endDate = ee.Date.fromYMD(endYear + 1, 1, 1);

// Make a list with years.
var years = ee.List.sequence(startYear, endYear);

// Make a list with months.
var months = ee.List.sequence(1, 12);

```

We import the CHIRPS [ImageCollection](#) and select the imagery for the relevant dates, as presented in Chap. F4.2. Note that we used the pentad time series; each image in this collection contains the accumulated rainfall for five days. The daily product is also available in Earth Engine. The pentad dataset was used rather than the daily data product to reduce the number of computations needed to aggregate the data.

```

// Import the CHIRPS dataset.
var CHIRPS = ee.ImageCollection('UCSB-CHG/CHIRPS/PENTAD');

// Filter for the relevant time period.
CHIRPS = CHIRPS.filterDate(startDate, endDate);

```

The year and month lists are used in the function below to calculate the monthly rainfall. We use a server-side nested loop where we first [map](#) over the years (2010, 2011, ... 2020) and then [map](#) over the months (1, 2, ... 12). This returns an image with the total rainfall for each month. We set the year, month, and timestamp (`'system:time_start'`) for each image and flatten the image to turn the object into a single [ImageCollection](#).

```

// We apply a nested loop where we first map over
// the relevant years and then map over the relevant
// months. The function returns an image with the total (sum)
// rainfall for each month. A flatten is applied to convert a
// feature collection of features into a single feature collection.
var monthlyPrecip = ee.ImageCollection.fromImages(
  years.map(function(y) {

```

```

    return months.map(function(m) {
      var w = CHIRPS.filter(ee.Filter
          .calendarRange(y, y, 'year'))
          .filter(ee.Filter.calendarRange(m, m,
              'month'))
          .sum();
      return w.set('year', y)
          .set('month', m)
          .set('system:time_start', ee.Date
              .fromYMD(y, m, 1));
    });
  }).flatten()
);

```

Add a layer with the monthly mean precipitation to the map and calculate a chart with monthly mean precipitation (Fig. A2.5.6).

```

// Add the layer with monthly mean. Note that we clip for the Mekong
river basin.
var precipVis = {
  min: 0,
  max: 250,
  palette: 'white, blue, darkblue, red, purple'
};

Map.addLayer(monthlyPrecip.mean().clip(mekongBasin),
  precipVis,
  '2015 precipitation');

// Set the title and axis labels for the chart.
var title = {
  title: 'Monthly precipitation',
  hAxis: {
    title: 'Time'
  },
  vAxis: {
    title: 'Precipitation (mm)'
  },
};

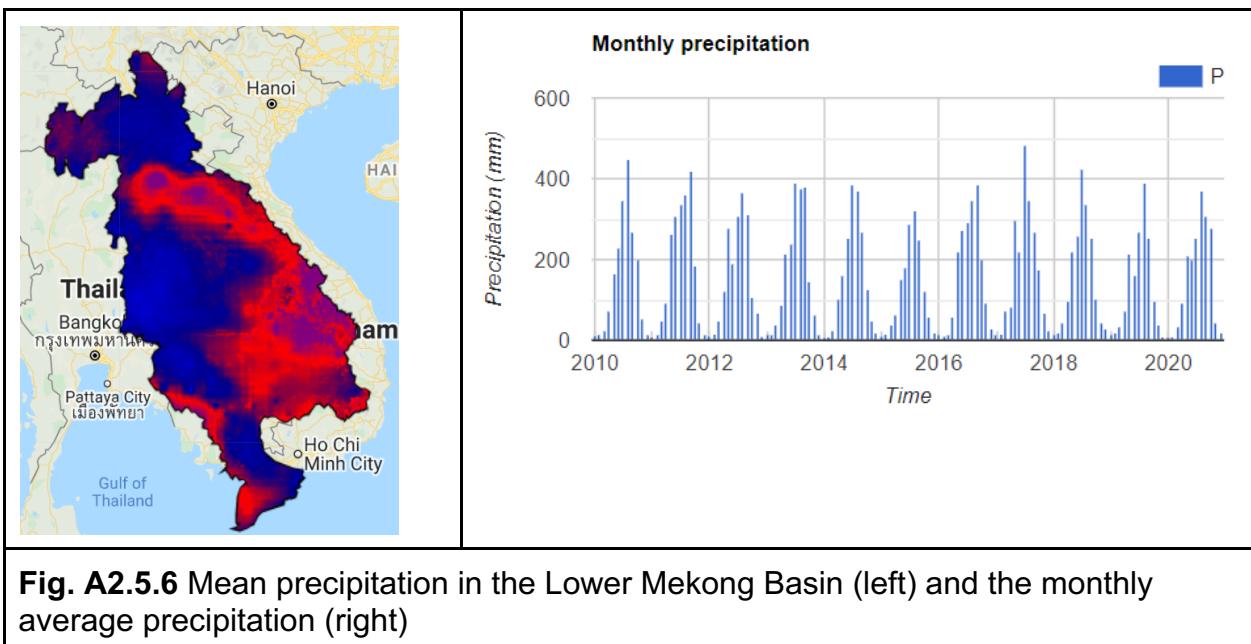
```

```

// Plot the chart using the Mekong boundary.
var chartMonthly = ui.Chart.image.seriesByRegion({
  imageCollection: monthlyPrecip,
  regions: mekongBasin.geometry(),
  reducer: ee.Reducer.mean(),
  band: 'precipitation',
  scale: 5000,
  xProperty: 'system:time_start'
}).setSeriesNames(['P'])
.setOptions(title)
.setChartType('ColumnChart');

// Print the chart.
print(chartMonthly);

```



**Fig. A2.5.6** Mean precipitation in the Lower Mekong Basin (left) and the monthly average precipitation (right)

**Code Checkpoint A25a.** The book's repository contains a script that shows what your code should look like at this point.

### **Section 2. Calculating Monthly Evapotranspiration**

Measuring evapotranspiration at large scales is important for assessing climate and anthropogenic effects on natural and agricultural ecosystems (Kustas and Norman 1996). Methods exist to measure ET at a field scale, but those methods cannot be

extrapolated to larger areas. Traditional ways of estimating ET have been to use reference ET, derived from various weather-related parameters, with a crop coefficient. However, there are large uncertainties due to, for example, spatial and temporal heterogeneity and data gaps. Satellite information can be very useful as it provides spatially and temporally dense information for ET estimation.

Different methods exist to map ET from remote sensing data, including simple empirical models that relate spectral reflectance with ET, vegetation index models, energy budget, and deterministic models (Courault et al. 2005). Fundamentally, however, ET is governed by the energy budget and driving variables such as surface temperature. The total amount of available net radiant energy is divided into a soil heat flux and the atmospheric convective fluxes, which are the sensible heat flux ( $H$ ) and latent energy exchanges (LE). This essentially means that the temperature decreases when energy is used for ET. Indeed, there are different ways to further quantify the different energy fluxes in more detail, and there is a wide body of scientific literature that describes those methods.

There are different readily available ET products derived from the Moderate Resolution Imaging Spectroradiometer (MODIS), including Atmosphere-Land Exchange Inverse (ALEXI; Anderson et al. 1997, Mecikalski et al. 1999), the operational Simplified Surface Energy Balance (SSEB; Senay et al. 2013), CSIRO MODIS Rescaled Evapotranspiration (CMRSET; Guerschman et al. 2009), and MOD16. The MOD16 algorithm is based on the logic of the Penman-Monteith equation, which uses daily meteorological reanalysis data and eight-day remotely sensed vegetation property dynamics from MODIS as inputs. The MOD16 product is available in the Earth Engine assets, and we will use this product for ET estimation in the next exercise.

First, we start a new script and repeat the first two steps of the previous section by copying and pasting the following code:

```
// Import the Lower Mekong boundary.  
var mekongBasin = ee.FeatureCollection(  
    'projects/gee-book/assets/A2-5/lowerMekongBasin');  
  
// Center the map.  
Map.centerObject(mekongBasin, 5);  
  
// Add the Lower Mekong Basin boundary to the map.  
Map.addLayer(mekongBasin, {}, 'Lower Mekong basin');
```

```

// Set start and end years.
var startYear = 2010;
var endYear = 2020;

// Create two date objects for start and end years.
var startDate = ee.Date.fromYMD(startYear, 1, 1);
var endDate = ee.Date.fromYMD(endYear + 1, 1, 1);

// Make a list with years.
var years = ee.List.sequence(startYear, endYear);

// Make a list with months.
var months = ee.List.sequence(1, 12);

```

We import the MOD16 dataset and select the ET band, which represents total evapotranspiration.

```

// Import the MOD16 dataset.
var mod16 = ee.ImageCollection('MODIS/006/MOD16A2').select('ET');

// Filter for the relevant time period.
mod16 = mod16.filterDate(startDate, endDate);

```

We use the same function to calculate monthly values as in the previous section. Note that we multiply by 0.1 as a scaling factor. The scaling factor can be found in the description of the dataset in Earth Engine. Scaling factors are applied to reduce the required storage capacity by changing the data type.

```

// We apply a nested loop where we first map over
// the relevant years and then map over the relevant
// months. The function returns an image with the total (sum)
// evapotranspiration for each month. A flatten is applied to convert
// a
// collection of collections into a single collection.
// We multiply by 0.1 because of the ET scaling factor.
var monthlyEvap = ee.ImageCollection.fromImages(
  years.map(function(y) {
    return months.map(function(m) {
      var w = mod16.filter(ee.Filter

```

```

        .calendarRange(y, y, 'year'))
        .filter(ee.Filter.calendarRange(m, m,
            'month'))
        .sum()
        .multiply(0.1);
    return w.set('year', y)
        .set('month', m)
        .set('system:time_start', ee.Date
            .fromYMD(y, m, 1));
    });
}).flatten()
);

```

We use the code below to visualize the results. Note that we changed the color of the bar chart and applied the reducer on a 500 m spatial resolution.

```

// Add the layer with monthly mean. Note that we clip for the Mekong
river basin.
var evapVis = {
    min: 0,
    max: 140,
    palette: 'red, orange, yellow, blue, darkblue'
};

Map.addLayer(monthlyEvap.mean().clip(mekongBasin),
    evapVis,
    'Mean monthly ET');

// Set the title and axis labels for the chart.
var title = {
    title: 'Monthly evapotranspiration',
    hAxis: {
        title: 'Time'
    },
    vAxis: {
        title: 'Evapotranspiration (mm)'
    },
    colors: ['red']
};

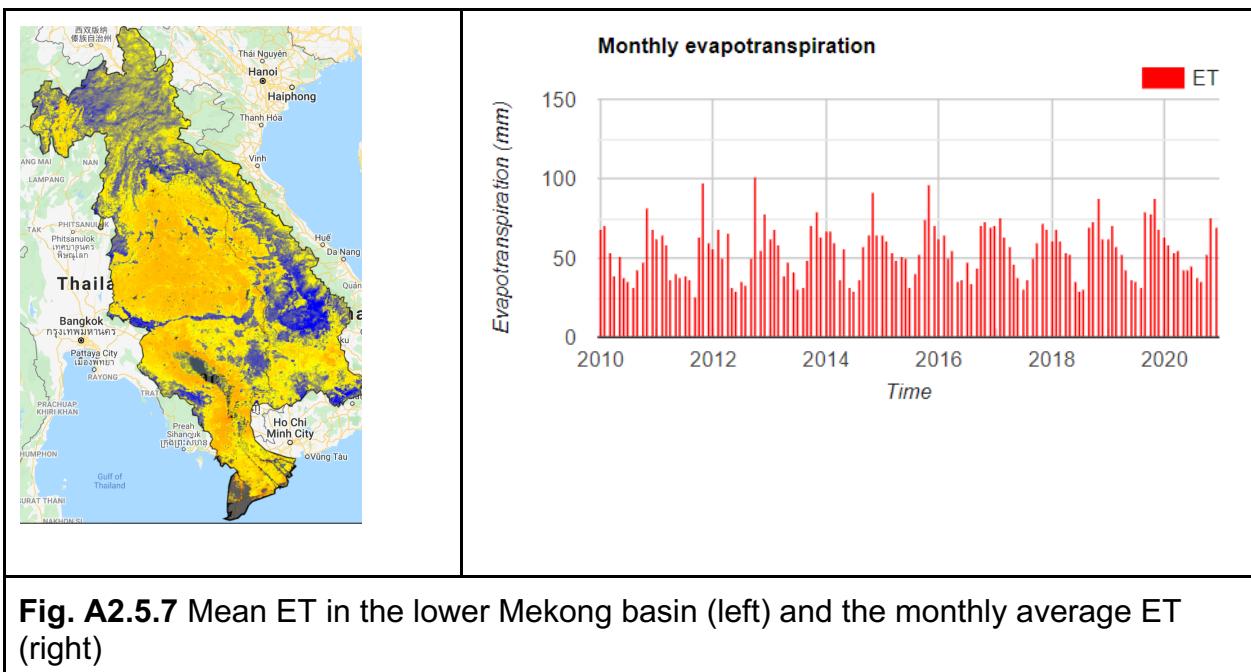
```

```

// Plot the chart using the Mekong boundary.
var chartMonthly = ui.Chart.image.seriesByRegion({
  imageCollection: monthlyEvap,
  regions: mekongBasin.geometry(),
  reducer: ee.Reducer.mean(),
  band: 'ET',
  scale: 500,
  xProperty: 'system:time_start'
}).setSeriesNames(['ET'])
.setOptions(title)
.setChartType('ColumnChart');

// Print the chart.
print(chartMonthly);

```



**Fig. A2.5.7** Mean ET in the lower Mekong basin (left) and the monthly average ET (right)

**Code Checkpoint A25b.** The book's repository contains a script that shows what your code should look like at this point.

### Section 3. Monthly Water Balance

We learned that the water balance is calculated using precipitation, evapotranspiration, runoff, and storage changes (Eq. A2.5.1). In the previous two sections, we calculated

the monthly precipitation (P) on a 5 km spatial resolution and the monthly evapotranspiration (ET) on a 500 m spatial resolution. In Equation A2.5.2 we rearrange Equation 2.7.1 so that we can calculate the portion of Q and  $\Delta S$  on a pixel level and aggregate that information to a basin level.

$$P - E = Q + \Delta S \quad (\text{A2.5.2})$$

In this section we will use the previous data to calculate the monthly water balance. First, we set the dates and import the relevant `ImageCollection`, as shown in the previous sections. Copy and paste the code below in a new script.

```
// Import the Lower Mekong boundary.  
var mekongBasin = ee.FeatureCollection(  
    'projects/gee-book/assets/A2-5/lowerMekongBasin');  
  
// Center the map.  
Map.centerObject(mekongBasin, 5);  
  
// Add the Lower Mekong Basin boundary to the map.  
Map.addLayer(mekongBasin, {}, 'Lower Mekong basin');  
  
// Set start and end years.  
var startYear = 2010;  
var endYear = 2020;  
  
// Create two date objects for start and end years.  
var startDate = ee.Date.fromYMD(startYear, 1, 1);  
var endDate = ee.Date.fromYMD(endYear + 1, 1, 1);  
  
// Make a list with years.  
var years = ee.List.sequence(startYear, endYear);  
  
// Make a list with months.  
var months = ee.List.sequence(1, 12);  
  
// Import the CHIRPS dataset.  
var CHIRPS = ee.ImageCollection('UCSB-CHG/CHIRPS/PENTAD');  
  
// Filter for relevant time period.  
CHIRPS = CHIRPS.filterDate(startDate, endDate);
```

```

// Import the MOD16 dataset.
var mod16 = ee.ImageCollection('MODIS/006/MOD16A2').select('ET');

// Filter for relevant time period.
mod16 = mod16.filterDate(startDate, endDate);

```

Now we use the function that we used earlier to calculate monthly ET and P to calculate the water balance.

```

// We apply a nested loop where we first map over
// the relevant years and then map over the relevant
// months. The function returns an image with P - ET
// for each month. A flatten is applied to convert an
// collection of collections into a single collection.
var waterBalance = ee.ImageCollection.fromImages(
  years.map(function(y) {
    return months.map(function(m) {

      var P = CHIRPS.filter(ee.Filter
        .calendarRange(y, y, 'year'))
        .filter(ee.Filter.calendarRange(m, m,
          'month'))
        .sum();

      var ET = mod16.filter(ee.Filter
        .calendarRange(y, y, 'year'))
        .filter(ee.Filter.calendarRange(m, m,
          'month'))
        .sum()
        .multiply(0.1);

      var wb = P.subtract(ET).rename('wb');

      return wb.set('year', y)
        .set('month', m)
        .set('system:time_start', ee.Date
          .fromYMD(y, m, 1));
    });
  });

```

```
    }).flatten()
);
```

Next we add the monthly mean water balance to the map and calculate the monthly water balance (Fig. A2.5.8). Note that negative numbers in the map indicate regions with an overall surplus of ET, whereas negative monthly water balances indicate a surplus ET for the whole region.

```
// Add layer with monthly mean. note that we clip for the Mekong river
basin.
var balanceVis = {
  min: -50,
  max: 200,
  palette: 'red, orange, yellow, blue, darkblue, purple'
};

Map.addLayer(waterBalance.mean().clip(mekongBasin),
  balanceVis,
  'Mean monthly water balance');

// Set the title and axis labels for the chart.
var title = {
  title: 'Monthly water balance',
  hAxis: {
    title: 'Time'
  },
  vAxis: {
    title: 'Evapotranspiration (mm)'
  },
  colors: ['green']
};

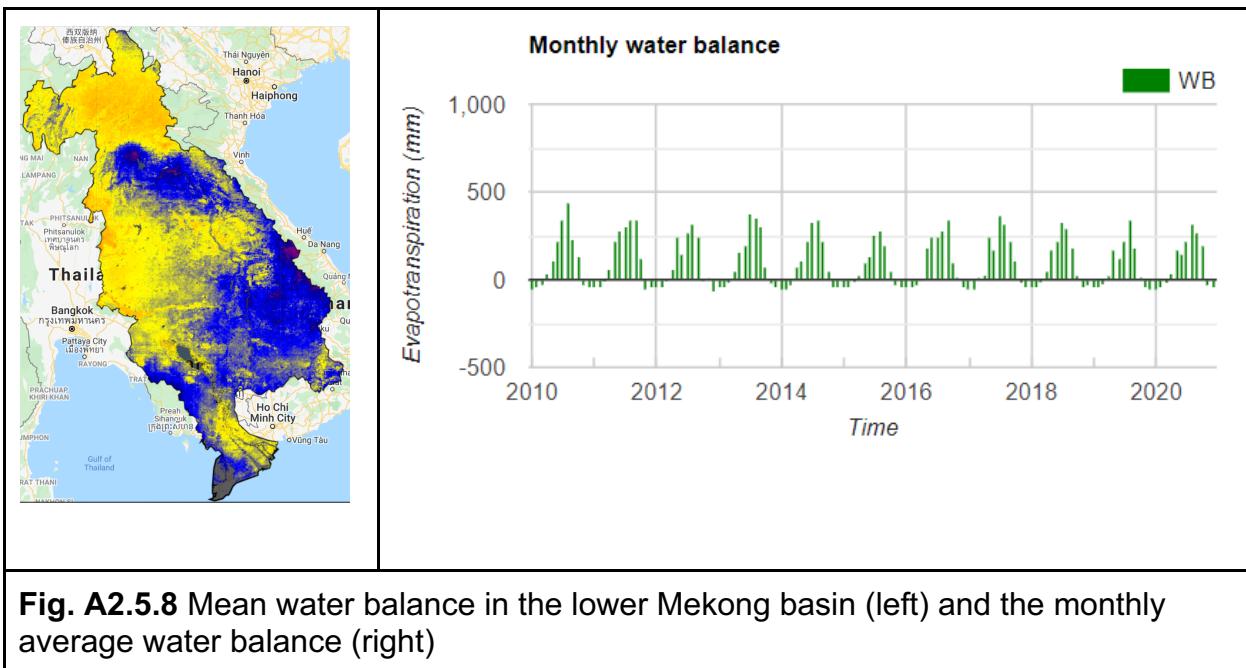
// Plot the chart using the Mekong boundary.
var chartMonthly = ui.Chart.image.seriesByRegion({
  imageCollection: waterBalance,
  regions: mekongBasin.geometry(),
  reducer: ee.Reducer.mean(),
  band: 'wb',
  scale: 500,
  xProperty: 'system:time_start'
```

```

}).setSeriesNames(['WB'])
.setOptions(title)
.setChartType('ColumnChart');

// Print the chart.
print(chartMonthly);

```



**Fig. A2.5.8** Mean water balance in the lower Mekong basin (left) and the monthly average water balance (right)

**Code Checkpoint A25c.** The book's repository contains a script that shows what your code should look like at this point.

#### Section 4. Vegetation and Drought Indices

Calculating vegetation indices is a common practice when working with remote sensing data. The Normalized Difference Vegetation Index (NDVI; Rouse et al. 1973) and Enhanced Vegetation Index (EVI; Chap. F3.1) (Huete et al. 1994) are among the most commonly used. Vegetation indices rely on the absorption and reflection spectra of chlorophyll, often including the red band, where absorption is high, and near infrared, where reflection is high. Vegetation indices are often used to measure crop health and density, but they can also be used to measure, for example, biophysical health over longer time periods (Poortinga et al. 2018). Vegetation indices can be calculated from the spectral reflectance, but readily available products can be used as well. The latter have been processed and contain, for example, corrections for outliers and artifacts.

Besides vegetation indices, there are many other indices to describe specific natural phenomena or detect specific land surface features. These indices often rely on simple band ratios or more sophisticated formulas containing multiple bands. Drought indices are another category of important indicators as they enable us to depict spatiotemporal drought patterns in great detail. This can be particularly useful for remote areas where people rely on local agriculture for their livelihoods but production data is scarce. In the next exercise we will be using the Moisture Stress Index (MSI; Vogelmann 1985), as this index has been shown to be highly related to soil moisture. Eq. A2.5.3 shows that MSI is calculated from the shortwave infrared (SWIR) and near infrared (NIR) bands.

$$\text{MSI} = \text{SWIR} / \text{NIR} \quad (\text{A2.5.3})$$

In this section, we use the EVI product for the vegetation index from MODIS. This data is collected daily and can be used for calculation of the monthly vegetation index. You may note that we import the readily available EVI product and also import the spectral reflectance dataset.

We first import all the relevant datasets and define the period of interest, as in the other exercises, starting a new script.

```
// Import the Lower Mekong boundary.
var mekongBasin = ee.FeatureCollection(
  'projects/gee-book/assets/A2-5/lowerMekongBasin');

// Center the map.
Map.centerObject(mekongBasin, 5);

// Add the Lower Mekong Basin boundary to the map.
Map.addLayer(mekongBasin, {}, 'Lower Mekong basin');

// Set start and end years.
var startYear = 2010;
var endYear = 2020;

// Create two date objects for start and end years.
var startDate = ee.Date.fromYMD(startYear, 1, 1);
var endDate = ee.Date.fromYMD(endYear + 1, 1, 1);

// Make a list with years.
var years = ee.List.sequence(startYear, endYear);
```

```

// Make a list with months.
var months = ee.List.sequence(1, 12);

// Import the CHIRPS dataset.
var CHIRPS = ee.ImageCollection('UCSB-CHG/CHIRPS/PENTAD');

// Filter for relevant time period.
CHIRPS = CHIRPS.filterDate(startDate, endDate);

// Import the MOD16 dataset.
var mod16 = ee.ImageCollection('MODIS/006/MOD16A2').select('ET');

// Filter for relevant time period.
mod16 = mod16.filterDate(startDate, endDate);

// Import and filter the MOD13 dataset.
var mod13 = ee.ImageCollection('MODIS/006/MOD13A1');
mod13 = mod13.filterDate(startDate, endDate);

// Select the EVI.
var EVI = mod13.select('EVI');

// Import and filter the MODIS Terra surface reflectance dataset.
var mod09 = ee.ImageCollection('MODIS/006/MOD09A1');
mod09 = mod09.filterDate(startDate, endDate);

```

Two steps are needed to calculate the MSI. First, we need to remove the clouds and cloud shadows. This can be done by using the StateQA quality band that comes with the MOD09 product. After all the artifacts have been removed, we can calculate the MSI in the next function.

```

// We use a function to remove clouds and cloud shadows.
// We map over the mod09 image collection and select the StateQA band.
// We mask pixels and return the image with clouds and cloud shadows
// masked.
mod09 = mod09.map(function(image) {
  var quality = image.select('StateQA');
  var mask = image.and(quality.bitwiseAnd(1).eq(
    0)) // No clouds.

```

```

        .and(quality.bitwiseAnd(2).eq(0)); // No cloud shadow.

    return image.updateMask(mask);
});

// We use a function to calculate the Moisture Stress Index.
// We map over the mod09 image collection and select the NIR and SWIR
bands
// We set the timestamp and return the MSI.
var MSI = mod09.map(function(image) {
    var nirband = image.select('sur_refl_b02');
    var swirband = image.select('sur_refl_b06');

    var msi = swirband.divide(nirband).rename('MSI')
        .set('system:time_start', image.get(
            'system:time_start'));
    return msi;
});

```

We use the same nested loop as the previous sections, but now we calculate all layers and return an image where the layers are included as bands. Note that EVI and MOD16 are multiplied by the scale factor. For P, ET, and WB we calculate the sum; for MSI and EVI we calculate the mean.

```

// We apply a nested loop where we first map over
// the relevant years and then map over the relevant
// months. The function returns an image with bands for
// water balance (wb), rainfall (P), evapotranspiration (ET),
// EVI and MSI for each month. A flatten is applied to
// convert an collection of collections
// into a single collection.
var ic = ee.ImageCollection.fromImages(
    years.map(function(y) {
        return months.map(function(m) {
            // Calculate rainfall.
            var P = CHIRPS.filter(ee.Filter
                .calendarRange(y, y, 'year'))
                .filter(ee.Filter.calendarRange(m, m,
                    'month'))
                .sum();

```

```

// Calculate evapotranspiration.
var ET = mod16.filter(ee.Filter
    .calendarRange(y, y, 'year'))
    .filter(ee.Filter.calendarRange(m, m,
        'month'))
    .sum()
    .multiply(0.1);

// Calculate EVI.
var evi = EVI.filter(ee.Filter
    .calendarRange(y, y, 'year'))
    .filter(ee.Filter.calendarRange(m, m,
        'month'))
    .mean()
    .multiply(0.0001);

// Calculate MSI.
var msi = MSI.filter(ee.Filter
    .calendarRange(y, y, 'year'))
    .filter(ee.Filter.calendarRange(m, m,
        'month'))
    .mean();

// Calculate monthly water balance.
var wb = P.subtract(ET).rename('wb');

// Return an image with all images as bands.
return ee.Image.cat([wb, P, ET, evi, msi])
    .set('year', y)
    .set('month', m)
    .set('system:time_start', ee.Date
        .fromYMD(y, m, 1));

});
}).flatten()
);

```

We display the monthly mean EVI and MSI and show the time series (Fig. A2.5.9).

```

// Add the mean monthly EVI and MSI to the map.

var eviVis = {
  min: 0,
  max: 0.7,
  palette: 'red, orange, yellow, green, darkgreen'
};

Map.addLayer(ic.select('EVI').mean().clip(mekongBasin),
  eviVis,
  'EVI');

var msiVis = {
  min: 0.25,
  max: 1,
  palette: 'darkblue, blue, yellow, orange, red'
};

Map.addLayer(ic.select('MSI').mean().clip(mekongBasin),
  msiVis,
  'MSI');

// Define the water balance chart and print it to the console.

var chartWB =
  ui.Chart.image.series({
    imageCollection: ic.select(['wb', 'precipitation', 'ET']),
    region: mekongBasin,
    reducer: ee.Reducer.mean(),
    scale: 5000,
    xProperty: 'system:time_start'
  })
  .setSeriesNames(['wb', 'P', 'ET'])
  .setOptions({
    title: 'water balance',
    hAxis: {
      title: 'Date',
      titleTextStyle: {
        italic: false,
        bold: true
      }
    },
  },

```

```

    vAxis: {
      title: 'Water (mm)',
      titleTextStyle: {
        italic: false,
        bold: true
      }
    },
    lineWidth: 1,
    colors: ['green', 'blue', 'red'],
    curveType: 'function'
  });

// Print the water balance chart.
print(chartWB);

// Define the indices chart and print it to the console.
var chartIndices =
  ui.Chart.image.series({
    imageCollection: ic.select(['EVI', 'MSI']),
    region: mekongBasin,
    reducer: ee.Reducer.mean(),
    scale: 5000,
    xProperty: 'system:time_start'
  })
  .setSeriesNames(['EVI', 'MSI'])
  .setOptions({
    title: 'Monthly indices',
    hAxis: {
      title: 'Date',
      titleTextStyle: {
        italic: false,
        bold: true
      }
    },
    vAxis: {
      title: 'Index',
      titleTextStyle: {
        italic: false,
        bold: true
      }
    }
  });

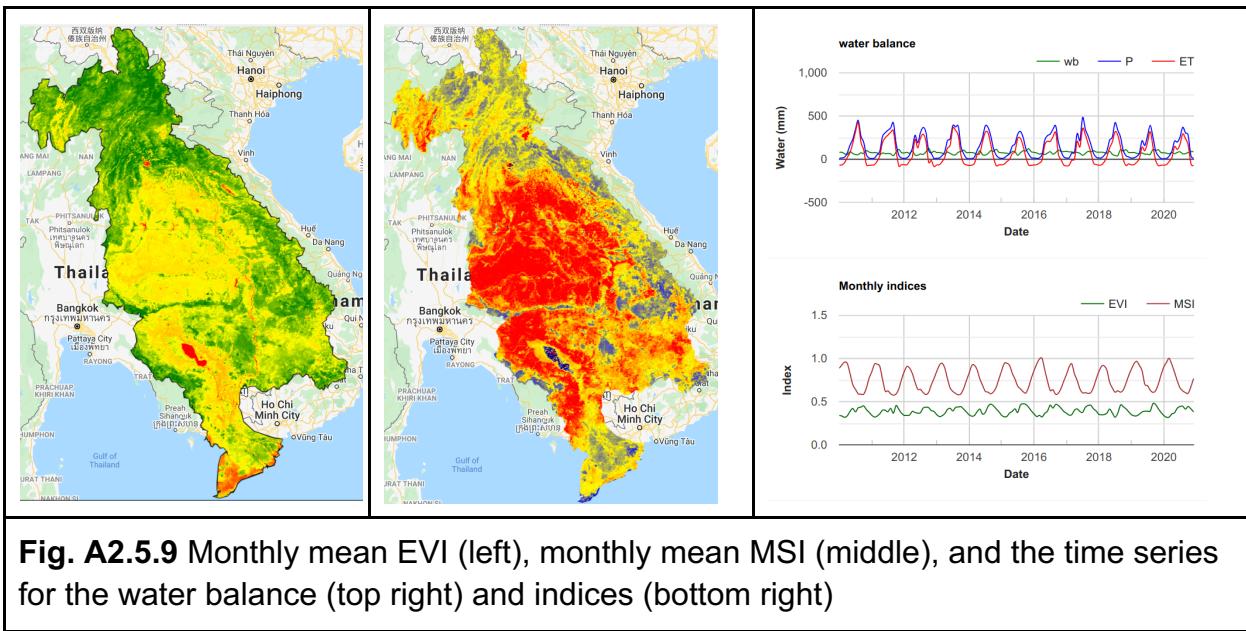
```

```

        },
        lineWidth: 1,
        colors: ['darkgreen', 'brown'],
        curveType: 'function'
    });

// Print the indices chart.
print(chartIndices);

```



**Code Checkpoint A25d.** The book's repository contains a script that shows what your code should look like at this point.

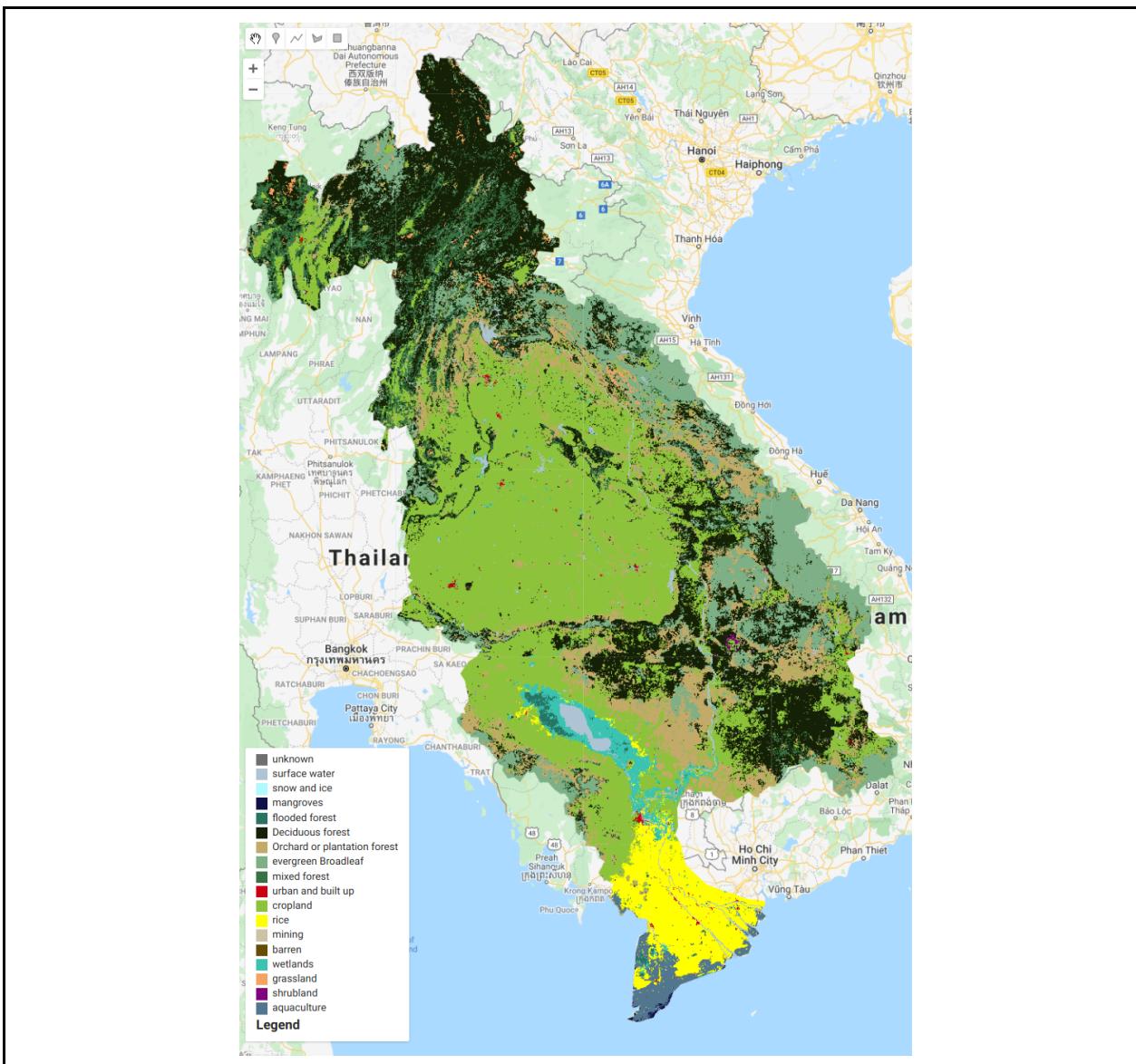
### Section 5. Partitioning Water Resources and Mapping Drought Impacts

Historical and near-real-time remote-sensing-derived data can be very useful to assess the impact on the ground. For example, overlaying information layers on water resources with land cover information enables us to partition water resources by land cover and investigate consumptive use of, for example, different agricultural commodities. Overlaying land cover with vegetation and drought indicators helps to investigate land cover categories that are most impacted by water shortage. It also helps to assess the impacts on crop production and food security (Poortinga et al. 2019), as well as potential environmental impacts, including impacts on biodiversity. However, in many cases maps are produced and updated only infrequently and are often not

accompanied by appropriate accuracy assessment information and documentation (Saah et al. 2019). Therefore, SERVIR-Mekong developed a time series of yearly land cover maps covering the greater Mekong region, including the Lower Mekong Basin (Saah et al. 2020, Potapov et al. 2019, Poortinga et al. 2020). Fig. A2.5.10 shows the land cover map of 2018.

### Section 5.1. Annual Classifications

In the following subsections, we illustrate annual land cover maps and a variety of interpretations of them. The annual maps can be accessed using script **A25s1 - Annual** in the book repository.

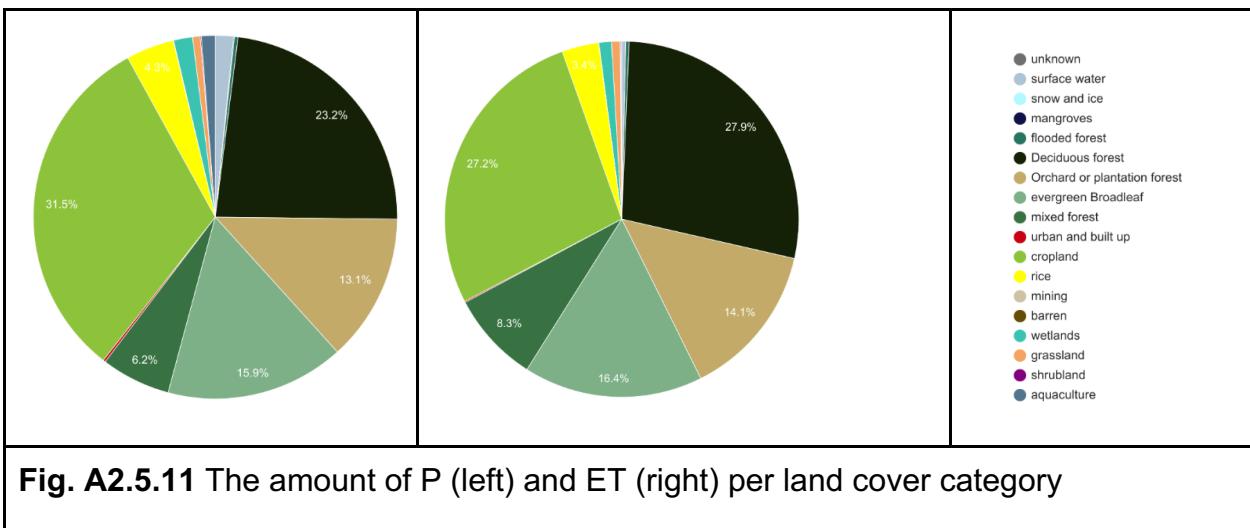


**Fig. A2.5.10** Land cover map of the Lower Mekong Basin

Note: if you get an error related to “too many concurrent aggregations” in this last script link or the following ones, try re-running the script.

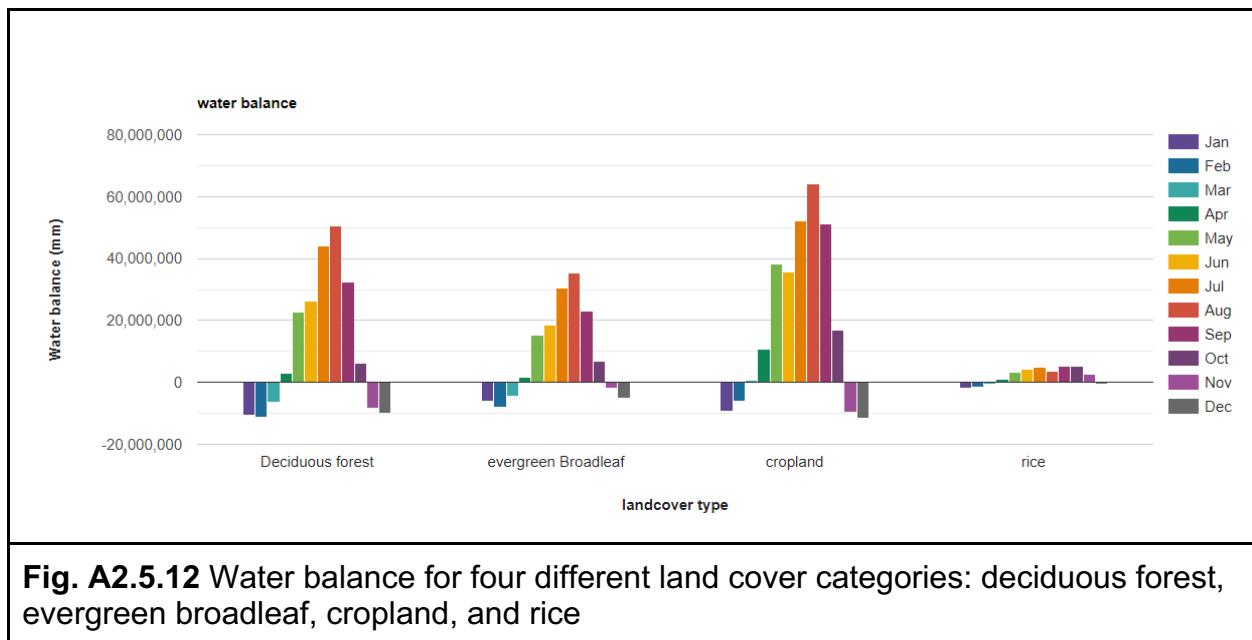
### **Section 5.2. Precipitation and Evapotranspiration**

An example of partitioning water inputs and consumption is shown in Fig. A2.5.11. Here we calculate the mean P and ET for each land cover category using the land cover map. We can see that the largest portions of water are being used by cropland and agriculture; plantations also consume a large portion of the water. The pie charts can be created and viewed using script **A25s2 - PET** in the book repository.



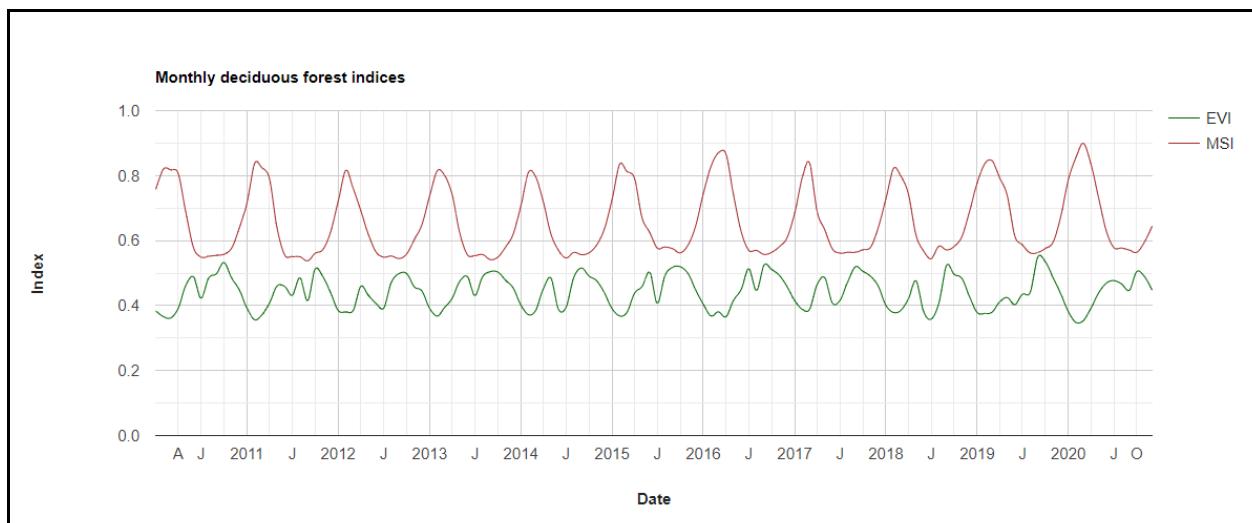
### **Section 5.3. Monthly Water Balance**

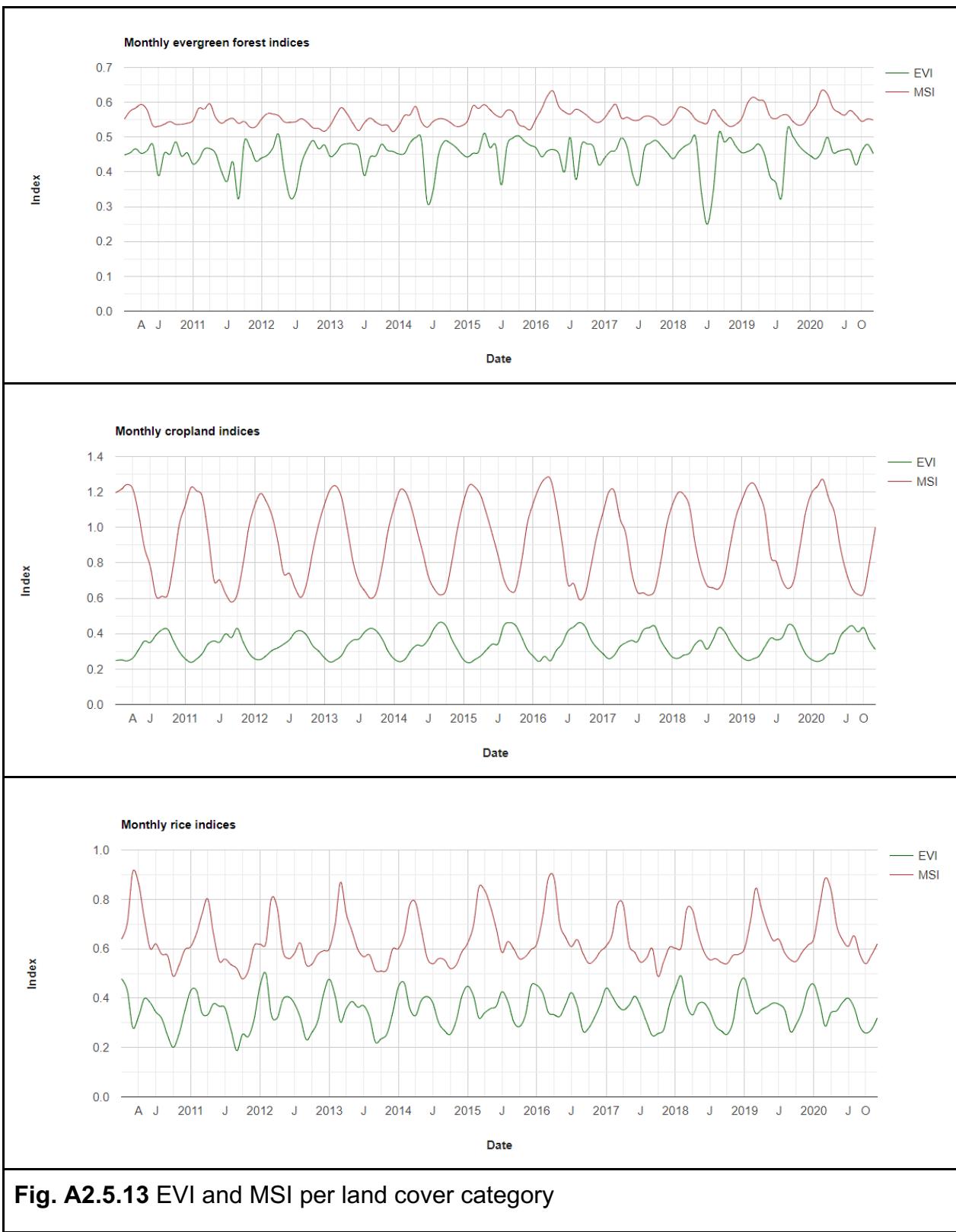
We can apply a similar overlaying method of partitioning for the water balance. Fig. A2.5.12 shows the monthly water balance for deciduous forest, evergreen broadleaf, cropland, and rice. It can be seen that a large portion of the water stored in the wet season is used by forest and cropland in the dry season. The area of paddy rice is smaller, so the total water consumption in the dry season of those categories is smaller. The monthly water balance charts can be found in script **A25s3 - Monthly** in the book repository.



#### Section 5.4. Per-class Water Balance Across Seasons

Partitioning can also be done per land cover category. In Fig. A2.5.13, we calculated the EVI and MSI for four land cover categories. There are very distinct patterns: we see little variation in the signal from the EVI, but large variations for the deciduous forest. For cropland, we found a signal that closely corresponds to the yearly dry and wet seasons, whereas we see multiple cropping seasons per year. The long time series enables us to investigate deviations from the mean, which in turn provides valuable information on, for example, potential drought impacts. The per-class water balance charts can be found in script **A25s4 - Per Class Balance** in the book repository.





## Synthesis

With what you learned in this chapter, you can analyze large-scale hydrological processes in a river basin. The approach can be applied for any river basin in the world using your own data or the open access data in the exercises.

**Assignment 1.** Test the approach in another part of the world using your own data or open-access data, or use your own training data for a more refined classification model.

**Assignment 2.** For further analysis, we encourage you to (1) replace MODIS with data from the Visible Infrared Imaging Radiometer Suite (VIIRS); (2) replace the CHIRPS data with the Integrated Multi-satellite Retrievals for GPM (IMERG) data and change the time intervals; and (3) use a different land cover map for partitioning the water resources.

## Conclusion

A safe and sustainable supply of water is essential for drinking, washing, cleaning, cooking, and growing food. However, water is often a scarce resource that needs to be managed in a sustainable and equitable way. Satellite data products in Earth Engine can help us to map the quantity of water in space and time. It enables us to partition water according to its consumptive use and evaluate how this affects a wide variety of important functions within a water basin.

## Feedback

To review this chapter and make suggestions or note any problems, please go now to [bit.ly/EEFA-review](https://bit.ly/EEFA-review). You can find summary statistics from past reviews at [bit.ly/EEFA-reviews-stats](https://bit.ly/EEFA-reviews-stats).

## References

Anderson MC, Norman JM, Diak GR, et al (1997) A two-source time-integrated model for estimating surface fluxes using thermal infrared remote sensing. *Remote Sens Environ* 60:195–216. [https://doi.org/10.1016/S0034-4257\(96\)00215-5](https://doi.org/10.1016/S0034-4257(96)00215-5)

Courault D, Seguin B, Olioso A (2005) Review on estimation of evapotranspiration from remote sensing data: From empirical to numerical modeling approaches. *Irrig Drain Syst* 19:223–249. <https://doi.org/10.1007/s10795-005-5186-0>

Funk C, Peterson P, Landsfeld M, et al (2015) The climate hazards infrared precipitation with stations – a new environmental record for monitoring extremes. *Sci Data* 2:1–21. <https://doi.org/10.1038/sdata.2015.66>

Guerschman JP, Van Dijk AIJM, Mattersdorf G, et al (2009) Scaling of potential evapotranspiration with MODIS data reproduces flux observations and catchment water balance observations across Australia. *J Hydrol* 369:107–119. <https://doi.org/10.1016/j.jhydrol.2009.02.013>

Hou AY, Kakar RK, Neeck S, et al (2014) The global precipitation measurement mission. *Bull Am Meteorol Soc* 95:701–722. <https://doi.org/10.1175/BAMS-D-13-00164.1>

Huete A, Justice C, Liu H (1994) Development of vegetation and soil indices for MODIS-EOS. *Remote Sens Environ* 49:224–234. [https://doi.org/10.1016/0034-4257\(94\)90018-3](https://doi.org/10.1016/0034-4257(94)90018-3)

Kummerow C, Barnes W, Kozu T, et al (1998) The tropical rainfall measuring mission (TRMM) sensor package. *J Atmos Ocean Technol* 15:809–817. [https://doi.org/10.1175/1520-0426\(1998\)015<0809:TTRMMT>2.0.CO;2](https://doi.org/10.1175/1520-0426(1998)015<0809:TTRMMT>2.0.CO;2)

Kustas WP, Norman JM (1996) Use of remote sensing for evapotranspiration monitoring over land surfaces. *Hydrol Sci J* 41:495–516. <https://doi.org/10.1080/02626669609491522>

Mecikalski JR, Diak GR, Anderson MC, Norman JM (1999) Estimating fluxes on continental scales using remotely sensed data in an atmospheric-land exchange model. *J Appl Meteorol* 38:1352–1369. [https://doi.org/10.1175/1520-0450\(1999\)038<1352:EFOCSU>2.0.CO;2](https://doi.org/10.1175/1520-0450(1999)038<1352:EFOCSU>2.0.CO;2)

Poortinga A, Aekakkararungroj A, Kityuttachai K, et al (2020) Predictive analytics for identifying land cover change hotspots in the Mekong region. *Remote Sens* 12:1472. <https://doi.org/10.3390/RS12091472>

Poortinga A, Clinton N, Saah D, et al (2018) An operational before-after-control-impact (BACI) designed platform for vegetation monitoring at planetary scale. *Remote Sens* 10:760. <https://doi.org/10.3390/rs10050760>

Poortinga A, Nguyen Q, Tenneson K, et al (2019) Linking Earth observations for assessing the food security situation in Vietnam: A landscape approach. *Front Environ Sci* 7:186. <https://doi.org/10.3389/fenvs.2019.00186>

Potapov P, Tyukavina A, Turubanova S, et al (2019) Annual continuous fields of woody vegetation structure in the Lower Mekong region from 2000-2017 Landsat time-series. *Remote Sens Environ* 232:111278. <https://doi.org/10.1016/j.rse.2019.111278>

Rouse Jr JW, Haas RH, Schell JA, Deering DW (1973) Paper a 20. In: Third Earth Resources Technology Satellite-1 Symposium: Section AB. Technical presentations. pp 309

Saah D, Tenneson K, Matin M, et al (2019) Land cover mapping in data scarce environments: Challenges and opportunities. *Front Environ Sci* 7:150. <https://doi.org/10.3389/fenvs.2019.00150>

Saah D, Tenneson K, Poortinga A, et al (2020) Primitives as building blocks for constructing land cover maps. *Int J Appl Earth Obs Geoinf* 85:101979. <https://doi.org/10.1016/j.jag.2019.101979>

Senay GB, Bohms S, Singh RK, et al (2013) Operational evapotranspiration mapping using remote sensing and weather datasets: A new parameterization for the SSEB approach. *J Am Water Resour Assoc* 49:577–591. <https://doi.org/10.1111/jawr.12057>

Strangeways I (2010) A history of rain gauges. *Weather* 65:133–138. <https://doi.org/10.1002/wea.548>

Vogelmann JE, Rock BN (1985) Spectral characterization of suspected acid deposition damage in red spruce (*Picea Rubens*) stands from Vermont. In: Proc. of the Airborne Imaging Spectrometer Data Anal. Workshop.