1. **LAND SURFACE TEMPERATURE.**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load ERA5-Land surface temperature data and filter by date and region

var collection = ee.ImageCollection('ECMWF/ERA5\_LAND/MONTHLY')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Function to calculate yearly mean temperature

var yearlyTemperature = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collection.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.mean().set('year', year).clip(roi);

})

);

// Compute yearly temperature thresholds for the region

var yearlyThresholds = yearlyTemperature.map(function(image) {

var meanTemperature = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), Threshold: meanTemperature.get('temperature\_2m')});

});

// Export yearly thresholds data to a CSV file

Export.table.toDrive({

collection: yearlyThresholds,

description: 'Yearly\_Thresholds\_Mt\_Kenya\_Temperature',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly thresholds

print(yearlyThresholds);

// Function to calculate monthly mean temperature

var monthlyTemperature = ee.ImageCollection.fromImages(

ee.List.sequence(1, 12).map(function(month) {

var filtered = collection.filter(ee.Filter.calendarRange(month, month, 'month'));

return filtered.mean().set('month', month).clip(roi);

})

);

// Extract monthly temperature values for each month

var monthlyTemperatureData = monthlyTemperature.map(function(image) {

var month = ee.String('Month').cat(ee.String(ee.Number(image.get('month')).int()));

var meanTemperature = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Month: month, MeanTemperature: meanTemperature.get('temperature\_2m')});

});

// Export monthly temperature data to a CSV file

Export.table.toDrive({

collection: monthlyTemperatureData,

description: 'Monthly\_Temperature\_Mt\_Kenya\_Forest',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print monthly temperature data

print(monthlyTemperatureData);

// Chart temperature against month

var monthlyChart = ui.Chart.feature.byFeature(monthlyTemperatureData, 'Month', 'MeanTemperature')

.setOptions({

title: 'Monthly Mean Temperature for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Month'},

vAxis: {title: 'Mean Temperature (°C)'},

lineWidth: 1,

pointSize: 3

});

// Print monthly chart

print(monthlyChart);

// Chart temperature against year

var yearlyChart = ui.Chart.feature.byFeature(yearlyThresholds, 'Year', 'Threshold')

.setOptions({

title: 'Yearly Temperature Thresholds for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Temperature Threshold (°C)'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Visualize yearly temperature

var yearlyTemperatureVis = {

min: -40,

max: 40,

palette: ['blue', 'green', 'yellow', 'red']

};

var yearlyTemperatureList = yearlyTemperature.toList(yearlyTemperature.size());

for (var i = 0; i < yearlyTemperatureList.size().getInfo(); i++) {

var image = ee.Image(yearlyTemperatureList.get(i));

var year = image.get('year').getInfo();

map.addLayer(image.select('temperature\_2m'), yearlyTemperatureVis, 'Yearly Temperature ' + year);

}

// Visualize monthly temperature

var monthlyTemperatureVis = {

min: -40,

max: 40,

palette: ['blue', 'green', 'yellow', 'red']

};

var monthlyTemperatureList = monthlyTemperature.toList(monthlyTemperature.size());

for (var j = 0; j < monthlyTemperatureList.size().getInfo(); j++) {

var image = ee.Image(monthlyTemperatureList.get(j));

var month = image.get('month').getInfo();

map.addLayer(image.select('temperature\_2m'), monthlyTemperatureVis, 'Monthly Temperature ' + month);

}

// Export yearly temperature images to Google Drive

for (var k = 0; k < yearlyTemperatureList.size().getInfo(); k++) {

var image = ee.Image(yearlyTemperatureList.get(k));

var year = image.get('year').getInfo();

Export.image.toDrive({

image: image.select('temperature\_2m'),

description: 'Yearly\_Temperature\_' + year,

folder: 'Temperature\_Images',

fileNamePrefix: 'Yearly\_Temperature\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Export monthly temperature images to Google Drive

for (var l = 0; l < monthlyTemperatureList.size().getInfo(); l++) {

var image = ee.Image(monthlyTemperatureList.get(l));

var month = image.get('month').getInfo();

Export.image.toDrive({

image: image.select('temperature\_2m'),

description: 'Monthly\_Temperature\_' + month,

folder: 'Temperature\_Images',

fileNamePrefix: 'Monthly\_Temperature\_' + month,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Display the map

ui.root.widgets().reset([map]);

// Print the map

print(map);

1. **PRECIPITATION**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load CHIRPS daily precipitation data and filter by date and region

var collection = ee.ImageCollection('UCSB-CHG/CHIRPS/DAILY')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Function to calculate yearly mean precipitation

var yearlyPrecipitation = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collection.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.mean().set('year', year).clip(roi);

})

);

// Compute yearly precipitation thresholds for the region

var yearlyThresholds = yearlyPrecipitation.map(function(image) {

var meanPrecipitation = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), Threshold: meanPrecipitation.get('precipitation')});

});

// Export yearly thresholds data to a CSV file

Export.table.toDrive({

collection: yearlyThresholds,

description: 'Yearly\_Thresholds\_Mt\_Kenya\_Precipitation',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly thresholds

print(yearlyThresholds);

// Function to calculate monthly mean precipitation

var monthlyPrecipitation = ee.ImageCollection.fromImages(

ee.List.sequence(1, 12).map(function(month) {

var filtered = collection.filter(ee.Filter.calendarRange(month, month, 'month'));

return filtered.mean().set('month', month).clip(roi);

})

);

// Extract monthly precipitation values for each month

var monthlyPrecipitationData = monthlyPrecipitation.map(function(image) {

var month = ee.String('Month').cat(ee.String(ee.Number(image.get('month')).int()));

var meanPrecipitation = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Month: month, MeanPrecipitation: meanPrecipitation.get('precipitation')});

});

// Export monthly precipitation data to a CSV file

Export.table.toDrive({

collection: monthlyPrecipitationData,

description: 'Monthly\_Precipitation\_Mt\_Kenya\_Forest',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print monthly precipitation data

print(monthlyPrecipitationData);

// Chart precipitation against month

var monthlyChart = ui.Chart.feature.byFeature(monthlyPrecipitationData, 'Month', 'MeanPrecipitation')

.setOptions({

title: 'Monthly Mean Precipitation for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Month'},

vAxis: {title: 'Mean Precipitation'},

lineWidth: 1,

pointSize: 3

});

// Print monthly chart

print(monthlyChart);

// Chart precipitation against year

var yearlyChart = ui.Chart.feature.byFeature(yearlyThresholds, 'Year', 'Threshold')

.setOptions({

title: 'Yearly Precipitation Thresholds for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Precipitation Threshold'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Visualize yearly precipitation

var yearlyPrecipitationVis = {

min: 0,

max: 3000, // Adjust the max value based on expected precipitation range

palette: ['blue', 'green', 'yellow', 'red']

};

var yearlyPrecipitationList = yearlyPrecipitation.toList(yearlyPrecipitation.size());

for (var i = 0; i < yearlyPrecipitationList.size().getInfo(); i++) {

var image = ee.Image(yearlyPrecipitationList.get(i));

var year = image.get('year').getInfo();

map.addLayer(image, yearlyPrecipitationVis, 'Yearly Precipitation ' + year);

}

// Visualize monthly precipitation

var monthlyPrecipitationVis = {

min: 0,

max: 500, // Adjust the max value based on expected monthly precipitation range

palette: ['blue', 'green', 'yellow', 'red']

};

// Add monthly precipitation composite to the map

var monthlyPrecipitationComposite = monthlyPrecipitation.median();

map.addLayer(monthlyPrecipitationComposite, monthlyPrecipitationVis, 'Monthly Precipitation Composite');

// Export yearly precipitation images to Google Drive

for (var k = 0; k < yearlyPrecipitationList.size().getInfo(); k++) {

var image = ee.Image(yearlyPrecipitationList.get(k));

var year = image.get('year').getInfo();

Export.image.toDrive({

image: image,

description: 'Yearly\_Precipitation\_' + year,

folder: 'Precipitation\_Images',

fileNamePrefix: 'Yearly\_Precipitation\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Export monthly precipitation images to Google Drive

for (var l = 0; l < monthlyPrecipitationList.size().getInfo(); l++) {

var image = ee.Image(monthlyPrecipitationList.get(l));

var month = image.get('month').getInfo();

Export.image.toDrive({

image: image,

description: 'Monthly\_Precipitation\_' + month,

folder: 'Precipitation\_Images',

fileNamePrefix: 'Monthly\_Precipitation\_' + month,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Display the map

ui.root.widgets().reset([map]);

// Print the map

//print(map);

1. **SOIL MOISTURE**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load SMAP soil moisture data and filter by date and region

var collection = ee.ImageCollection('NASA\_USDA/HSL/soil\_moisture')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Function to calculate yearly mean soil moisture content

var yearlySoilMoisture = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collection.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.mean().set('year', year).clip(roi);

})

);

// Compute yearly soil moisture thresholds for the region

var yearlyThresholds = yearlySoilMoisture.map(function(image) {

var meanSoilMoisture = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), Threshold: meanSoilMoisture.get('ssm')});

});

// Export yearly thresholds data to a CSV file

Export.table.toDrive({

collection: yearlyThresholds,

description: 'Yearly\_Thresholds\_Mt\_Kenya\_Soil\_Moisture',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly thresholds

print(yearlyThresholds);

// Function to calculate monthly mean soil moisture content

var monthlySoilMoisture = ee.ImageCollection.fromImages(

ee.List.sequence(1, 12).map(function(month) {

var filtered = collection.filter(ee.Filter.calendarRange(month, month, 'month'));

return filtered.mean().set('month', month).clip(roi);

})

);

// Extract monthly soil moisture values for each month

var monthlySoilMoistureData = monthlySoilMoisture.map(function(image) {

var month = ee.String('Month').cat(ee.String(ee.Number(image.get('month')).int()));

var meanSoilMoisture = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Month: month, MeanSoilMoisture: meanSoilMoisture.get('ssm')});

});

// Export monthly soil moisture data to a CSV file

Export.table.toDrive({

collection: monthlySoilMoistureData,

description: 'Monthly\_Soil\_Moisture\_Mt\_Kenya\_Forest',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print monthly soil moisture data

print(monthlySoilMoistureData);

// Chart soil moisture content against month

var monthlyChart = ui.Chart.feature.byFeature(monthlySoilMoistureData, 'Month', 'MeanSoilMoisture')

.setOptions({

title: 'Monthly Mean Soil Moisture Content for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Month'},

vAxis: {title: 'Mean Soil Moisture Content'},

lineWidth: 1,

pointSize: 3

});

// Print monthly chart

print(monthlyChart);

// Chart soil moisture content against year

var yearlyChart = ui.Chart.feature.byFeature(yearlyThresholds, 'Year', 'Threshold')

.setOptions({

title: 'Yearly Soil Moisture Content Thresholds for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Soil Moisture Content Threshold'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Visualize yearly soil moisture content

var yearlySoilMoistureVis = {

min: 0,

max: 1, // Adjust the max value based on expected soil moisture content range (0-1)

palette: ['blue', 'green', 'yellow', 'red']

};

var yearlySoilMoistureList = yearlySoilMoisture.toList(yearlySoilMoisture.size());

for (var i = 0; i < yearlySoilMoistureList.size().getInfo(); i++) {

var image = ee.Image(yearlySoilMoistureList.get(i));

var year = image.get('year').getInfo();

map.addLayer(image.select('ssm'), yearlySoilMoistureVis, 'Yearly Soil Moisture Content ' + year);

}

// Visualize monthly soil moisture content

var monthlySoilMoistureVis = {

min: 0,

max: 1, // Adjust the max value based on expected soil moisture content range (0-1)

palette: ['blue', 'green', 'yellow', 'red']

};

var monthlySoilMoistureList = monthlySoilMoisture.toList(monthlySoilMoisture.size());

for (var j = 0; j < monthlySoilMoistureList.size().getInfo(); j++) {

var image = ee.Image(monthlySoilMoistureList.get(j));

var month = image.get('month').getInfo();

map.addLayer(image.select('ssm'), monthlySoilMoistureVis, 'Monthly Soil Moisture Content ' + month);

}

// Export yearly soil moisture content images to Google Drive

for (var k = 0; k < yearlySoilMoistureList.size().getInfo(); k++) {

var image = ee.Image(yearlySoilMoistureList.get(k));

var year = image.get('year').getInfo();

Export.image.toDrive({

image: image.select('ssm'),

description: 'Yearly\_Soil\_Moisture\_' + year,

folder: 'Soil\_Moisture\_Images',

fileNamePrefix: 'Yearly\_Soil\_Moisture\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Export monthly soil moisture content images to Google Drive

for (var l = 0; l < monthlySoilMoistureList.size().getInfo(); l++) {

var image = ee.Image(monthlySoilMoistureList.get(l));

var month = image.get('month').getInfo();

Export.image.toDrive({

image: image.select('ssm'),

description: 'Monthly\_Soil\_Moisture\_' + month,

folder: 'Soil\_Moisture\_Images',

fileNamePrefix: ' Monthly\_Soil\_Moisture\_' + month,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Display the map

ui.root.widgets().reset([map]);

// Print the map

//print(map);

1. **WIND DIRECTION**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load SMAP soil moisture data and filter by date and region

var collection = ee.ImageCollection('NASA\_USDA/HSL/soil\_moisture')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Function to calculate yearly mean soil moisture content

var yearlySoilMoisture = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collection.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.mean().set('year', year).clip(roi);

})

);

// Compute yearly soil moisture thresholds for the region

var yearlyThresholds = yearlySoilMoisture.map(function(image) {

var meanSoilMoisture = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), Threshold: meanSoilMoisture.get('ssm')});

});

// Export yearly thresholds data to a CSV file

Export.table.toDrive({

collection: yearlyThresholds,

description: 'Yearly\_Thresholds\_Mt\_Kenya\_Soil\_Moisture',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly thresholds

print(yearlyThresholds);

// Function to calculate monthly mean soil moisture content

var monthlySoilMoisture = ee.ImageCollection.fromImages(

ee.List.sequence(1, 12).map(function(month) {

var filtered = collection.filter(ee.Filter.calendarRange(month, month, 'month'));

return filtered.mean().set('month', month).clip(roi);

})

);

// Extract monthly soil moisture values for each month

var monthlySoilMoistureData = monthlySoilMoisture.map(function(image) {

var month = ee.String('Month').cat(ee.String(ee.Number(image.get('month')).int()));

var meanSoilMoisture = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Month: month, MeanSoilMoisture: meanSoilMoisture.get('ssm')});

});

// Export monthly soil moisture data to a CSV file

Export.table.toDrive({

collection: monthlySoilMoistureData,

description: 'Monthly\_Soil\_Moisture\_Mt\_Kenya\_Forest',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print monthly soil moisture data

print(monthlySoilMoistureData);

// Chart soil moisture content against month

var monthlyChart = ui.Chart.feature.byFeature(monthlySoilMoistureData, 'Month', 'MeanSoilMoisture')

.setOptions({

title: 'Monthly Mean Soil Moisture Content for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Month'},

vAxis: {title: 'Mean Soil Moisture Content'},

lineWidth: 1,

pointSize: 3

});

// Print monthly chart

print(monthlyChart);

// Chart soil moisture content against year

var yearlyChart = ui.Chart.feature.byFeature(yearlyThresholds, 'Year', 'Threshold')

.setOptions({

title: 'Yearly Soil Moisture Content Thresholds for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Soil Moisture Content Threshold'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Visualize yearly soil moisture content

var yearlySoilMoistureVis = {

min: 0,

max: 1, // Adjust the max value based on expected soil moisture content range (0-1)

palette: ['blue', 'green', 'yellow', 'red']

};

var yearlySoilMoistureList = yearlySoilMoisture.toList(yearlySoilMoisture.size());

for (var i = 0; i < yearlySoilMoistureList.size().getInfo(); i++) {

var image = ee.Image(yearlySoilMoistureList.get(i));

var year = image.get('year').getInfo();

map.addLayer(image.select('ssm'), yearlySoilMoistureVis, 'Yearly Soil Moisture Content ' + year);

}

// Visualize monthly soil moisture content

var monthlySoilMoistureVis = {

min: 0,

max: 1, // Adjust the max value based on expected soil moisture content range (0-1)

palette: ['blue', 'green', 'yellow', 'red']

};

var monthlySoilMoistureList = monthlySoilMoisture.toList(monthlySoilMoisture.size());

for (var j = 0; j < monthlySoilMoistureList.size().getInfo(); j++) {

var image = ee.Image(monthlySoilMoistureList.get(j));

var month = image.get('month').getInfo();

map.addLayer(image.select('ssm'), monthlySoilMoistureVis, 'Monthly Soil Moisture Content ' + month);

}

// Export yearly soil moisture content images to Google Drive

for (var k = 0; k < yearlySoilMoistureList.size().getInfo(); k++) {

var image = ee.Image(yearlySoilMoistureList.get(k));

var year = image.get('year').getInfo();

Export.image.toDrive({

image: image.select('ssm'),

description: 'Yearly\_Soil\_Moisture\_' + year,

folder: 'Soil\_Moisture\_Images',

fileNamePrefix: 'Yearly\_Soil\_Moisture\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Export monthly soil moisture content images to Google Drive

for (var l = 0; l < monthlySoilMoistureList.size().getInfo(); l++) {

var image = ee.Image(monthlySoilMoistureList.get(l));

var month = image.get('month').getInfo();

Export.image.toDrive({

image: image.select('ssm'),

description: 'Monthly\_Soil\_Moisture\_' + month,

folder: 'Soil\_Moisture\_Images',

fileNamePrefix: ' Monthly\_Soil\_Moisture\_' + month,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Display the map

ui.root.widgets().reset([map]);

// Print the map

//print(map);

1. **WIND SPEED**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load ERA5-Land wind speed data and filter by date and region

var collection = ee.ImageCollection('ECMWF/ERA5\_LAND/MONTHLY')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Function to calculate yearly mean wind speed

var yearlyWindSpeed = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collection.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.mean().set('year', year).clip(roi);

})

);

// Compute yearly wind speed thresholds for the region

var yearlyThresholds = yearlyWindSpeed.map(function(image) {

var meanWindSpeed = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), Threshold: meanWindSpeed.get('u\_component\_of\_wind\_10m')});

});

// Export yearly thresholds data to a CSV file

Export.table.toDrive({

collection: yearlyThresholds,

description: 'Yearly\_Thresholds\_Mt\_Kenya\_Wind\_Speed',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly thresholds

print(yearlyThresholds);

// Function to calculate monthly mean wind speed

var monthlyWindSpeed = ee.ImageCollection.fromImages(

ee.List.sequence(1, 12).map(function(month) {

var filtered = collection.filter(ee.Filter.calendarRange(month, month, 'month'));

return filtered.mean().set('month', month).clip(roi);

})

);

// Extract monthly wind speed values for each month

var monthlyWindSpeedData = monthlyWindSpeed.map(function(image) {

var month = ee.String('Month').cat(ee.String(ee.Number(image.get('month')).int()));

var meanWindSpeed = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Month: month, MeanWindSpeed: meanWindSpeed.get('u\_component\_of\_wind\_10m')});

});

// Export monthly wind speed data to a CSV file

Export.table.toDrive({

collection: monthlyWindSpeedData,

description: 'Monthly\_Wind\_Speed\_Mt\_Kenya\_Forest',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print monthly wind speed data

print(monthlyWindSpeedData);

// Chart wind speed against month

var monthlyChart = ui.Chart.feature.byFeature(monthlyWindSpeedData, 'Month', 'MeanWindSpeed')

.setOptions({

title: 'Monthly Mean Wind Speed for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Month'},

vAxis: {title: 'Mean Wind Speed (m/s)'},

lineWidth: 1,

pointSize: 3

});

// Print monthly chart

print(monthlyChart);

// Chart wind speed against year

var yearlyChart = ui.Chart.feature.byFeature(yearlyThresholds, 'Year', 'Threshold')

.setOptions({

title: 'Yearly Wind Speed Thresholds for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Wind Speed Threshold (m/s)'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Visualize yearly wind speed

var yearlyWindSpeedVis = {

min: 0,

max: 20, // Adjust the max value based on expected wind speed range

palette: ['blue', 'green', 'yellow', 'red']

};

var yearlyWindSpeedList = yearlyWindSpeed.toList(yearlyWindSpeed.size());

for (var i = 0; i < yearlyWindSpeedList.size().getInfo(); i++) {

var image = ee.Image(yearlyWindSpeedList.get(i));

var year = image.get('year').getInfo();

map.addLayer(image.select('u\_component\_of\_wind\_10m'), yearlyWindSpeedVis, 'Yearly Wind Speed ' + year);

}

// Visualize monthly wind speed

var monthlyWindSpeedVis = {

min: 0,

max: 20, // Adjust the max value based on expected wind speed range

palette: ['blue', 'green', 'yellow', 'red']

};

var monthlyWindSpeedList = monthlyWindSpeed.toList(monthlyWindSpeed.size());

for (var j = 0; j < monthlyWindSpeedList.size().getInfo(); j++) {

var image = ee.Image(monthlyWindSpeedList.get(j));

var month = image.get('month').getInfo();

map.addLayer(image.select('u\_component\_of\_wind\_10m'), monthlyWindSpeedVis, 'Monthly Wind Speed ' + month);

}

// Export yearly wind speed images to Google Drive

for (var k = 0; k < yearlyWindSpeedList.size().getInfo(); k++) {

var image = ee.Image(yearlyWindSpeedList.get(k));

var year = image.get('year').getInfo();

Export.image.toDrive({

image: image.select('u\_component\_of\_wind\_10m'),

description: 'Yearly\_Wind\_Speed\_' + year,

folder: 'Wind\_Speed\_Images',

fileNamePrefix: 'Yearly\_Wind\_Speed\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Export monthly wind speed images to Google Drive

for (var l = 0; l < monthlyWindSpeedList.size().getInfo(); l++) {

var image = ee.Image(monthlyWindSpeedList.get(l));

var month = image.get('month').getInfo();

Export.image.toDrive({

image: image.select('u\_component\_of\_wind\_10m'),

description: 'Monthly\_Wind\_Speed\_' + month,

folder: 'Wind\_Speed\_Images',

fileNamePrefix: 'Monthly\_Wind\_Speed\_' + month,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

}

// Display the map

ui.root.widgets().reset([map]);

// Print the map

print(map);

1. **SPECIFIC HUMIDITY**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load the ERA5 dataset and filter by date and region

var collection = ee.ImageCollection('ECMWF/ERA5\_LAND/MONTHLY')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Define the band to use (available bands in your dataset)

var bandName = 'dewpoint\_temperature\_2m'; // Replace with the desired band

// Function to calculate yearly mean for the chosen parameter

var yearlySpecificHumidity = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collection.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.select(bandName).mean().set('year', year).clip(roi);

})

);

// Visualize yearly specific humidity

var yearlySpecificHumidityVis = {

min: 250, // Adjust these values based on the data range

max: 300,

palette: ['blue', 'green', 'yellow', 'red']

};

// Add yearly specific humidity images to the map

var yearlySpecificHumidityList = yearlySpecificHumidity.toList(yearlySpecificHumidity.size());

for (var i = 0; i < yearlySpecificHumidityList.size().getInfo(); i++) {

var image = ee.Image(yearlySpecificHumidityList.get(i));

var year = image.get('year').getInfo();

Map.addLayer(image, yearlySpecificHumidityVis, 'Yearly Specific Humidity ' + year);

}

// Compute yearly mean specific humidity values for the region

var yearlyThresholds = yearlySpecificHumidity.map(function(image) {

var meanValue = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), Threshold: meanValue.get(bandName)});

});

// Export yearly thresholds data to a CSV file

Export.table.toDrive({

collection: yearlyThresholds,

description: 'Yearly\_Thresholds\_Mt\_Kenya\_Specific\_Humidity',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly thresholds

print(yearlyThresholds);

// Chart specific humidity against year

var yearlyChart = ui.Chart.feature.byFeature(yearlyThresholds, 'Year', 'Threshold')

.setOptions({

title: 'Yearly Specific Humidity Thresholds for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Specific Humidity Threshold'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Add yearly specific humidity composite to the map

var yearlySpecificHumidityComposite = yearlySpecificHumidity.median();

map.addLayer(yearlySpecificHumidityComposite, yearlySpecificHumidityVis, 'Yearly Specific Humidity Composite');

// Export yearly specific humidity images to Google Drive

yearlySpecificHumidity.toList(yearlySpecificHumidity.size()).getInfo().forEach(function(imageInfo) {

var image = ee.Image(imageInfo.id);

var year = imageInfo.properties.year;

Export.image.toDrive({

image: image,

description: 'Yearly\_Specific\_Humidity\_' + year,

folder: 'Specific\_Humidity\_Images',

fileNamePrefix: 'Yearly\_Specific\_Humidity\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

});

// Display the map

ui.root.widgets().reset([map]);

// Print the map

print(map);

1. **RELATIVE HUMIDITY**

// Load the forest shapefile for Mount Kenya

var roi = ee.FeatureCollection("users/kelipaul1/True\_forest");

// Define the time range

var startDate = '2000-01-01';

var endDate = '2022-12-31';

// Load the ERA5 dataset and filter by date and region

var collection = ee.ImageCollection('ECMWF/ERA5\_LAND/MONTHLY')

.filterBounds(roi)

.filterDate(startDate, endDate);

// Function to calculate relative humidity from temperature and dew point temperature

function computeRelativeHumidity(image) {

var T = image.select('temperature\_2m').subtract(273.15); // Convert from Kelvin to Celsius

var Td = image.select('dewpoint\_temperature\_2m').subtract(273.15); // Convert from Kelvin to Celsius

var rh = Td.expression(

'100 \* (exp((17.625 \* Td) / (Td + 243.04)) / exp((17.625 \* T) / (T + 243.04)))', {

'T': T,

'Td': Td

}).rename('relative\_humidity');

return image.addBands(rh);

}

// Compute relative humidity for each image in the collection

var collectionWithRH = collection.map(computeRelativeHumidity);

// Function to calculate yearly mean relative humidity

var yearlyRelativeHumidity = ee.ImageCollection.fromImages(

ee.List.sequence(2000, 2022).map(function(year) {

var filtered = collectionWithRH.filter(ee.Filter.calendarRange(year, year, 'year'));

return filtered.select('relative\_humidity').mean().set('year', year).clip(roi);

})

);

// Visualize yearly relative humidity

var yearlyRelativeHumidityVis = {

min: 0, // Adjust these values based on the data range

max: 100,

palette: ['blue', 'green', 'yellow', 'red']

};

// Add yearly relative humidity images to the map

var yearlyRelativeHumidityList = yearlyRelativeHumidity.toList(yearlyRelativeHumidity.size());

for (var i = 0; i < yearlyRelativeHumidityList.size().getInfo(); i++) {

var image = ee.Image(yearlyRelativeHumidityList.get(i));

var year = image.get('year').getInfo();

Map.addLayer(image, yearlyRelativeHumidityVis, 'Yearly Relative Humidity ' + year);

}

// Compute yearly mean relative humidity values for the region

var yearlyRHMeans = yearlyRelativeHumidity.map(function(image) {

var meanValue = image.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: roi,

scale: 250

});

return ee.Feature(null, {Year: image.get('year'), RH\_Mean: meanValue.get('relative\_humidity')});

});

// Export yearly mean relative humidity data to a CSV file

Export.table.toDrive({

collection: yearlyRHMeans,

description: 'Yearly\_RH\_Means\_Mt\_Kenya',

folder: 'Statistics',

fileFormat: 'CSV'

});

// Print yearly mean relative humidity

print(yearlyRHMeans);

// Chart relative humidity against year

var yearlyRHChart = ui.Chart.feature.byFeature(yearlyRHMeans, 'Year', 'RH\_Mean')

.setOptions({

title: 'Yearly Relative Humidity Means for Mt. Kenya Forest (2000-2022)',

hAxis: {title: 'Year'},

vAxis: {title: 'Relative Humidity (%)'},

lineWidth: 1,

pointSize: 3

});

// Print yearly chart

print(yearlyRHChart);

// Create a map and add layers

var map = ui.Map();

map.centerObject(roi, 8);

// Add yearly relative humidity composite to the map

var yearlyRHComposite = yearlyRelativeHumidity.median();

map.addLayer(yearlyRHComposite, yearlyRelativeHumidityVis, 'Yearly Relative Humidity Composite');

// Export yearly relative humidity images to Google Drive

yearlyRelativeHumidity.toList(yearlyRelativeHumidity.size()).getInfo().forEach(function(imageInfo) {

var image = ee.Image(imageInfo.id);

var year = imageInfo.properties.year;

Export.image.toDrive({

image: image,

description: 'Yearly\_RH\_' + year,

folder: 'RH\_Images',

fileNamePrefix: 'Yearly\_RH\_' + year,

region: roi.geometry(),

scale: 250,

maxPixels: 1e13

});

});

// Display the map

ui.root.widgets().reset([map]);

// Print the map

print(map);