Geospatial Modeling Crash Course

A bit of a kitchen sink

Goal

Provide a training that exposes individuals to the framework and components of a common geospatial modeling process so they can more quickly engage with the material in their own work.

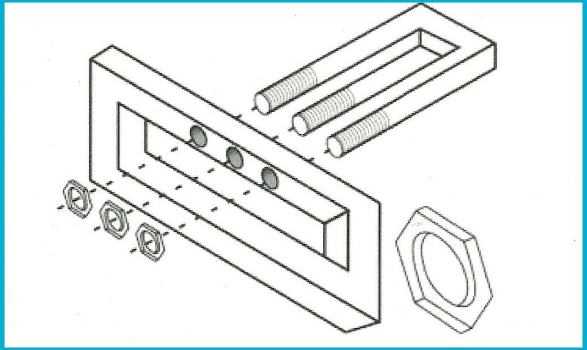
- Drive into the deep end
- Understand potential and follow the process
- Have a resource to look back on

Please Ask Questions

Outline

Part 1: Google Earth Engine (GEE)

- 4 sections
- A bit more straightforward
- You don’t need to know GEE



Part 2: R

- 2 sections
- A bit less straightforward
- You don’t need to know R (hopefully)

Introduction (Part 1)

GEE: Resources

https://drive.google.com/open?id=1W4TZS9eGWq_Sbd-2nmgeq48FbY3t-JxF

Introduction (Part 1)

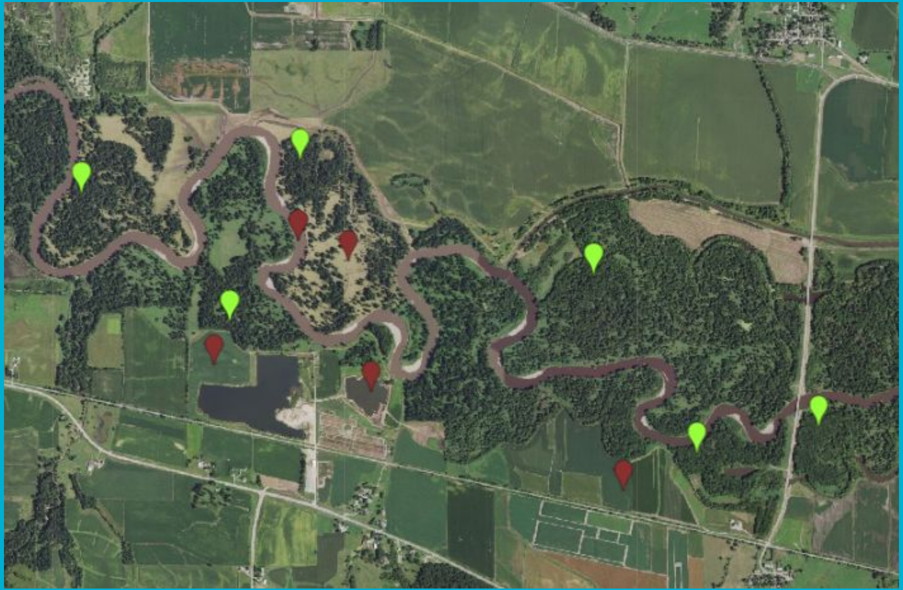
GEE Presentation

https://docs.google.com/presentation/d/191dsyRePGyPruiH-fQ1nhx0jx7oJrJ2wa2pTP4NhK3E/edit#slide=id.g3556a9360b_0_25

Interact with GEE Drop Points (Part 2)

Generate potential training data in GEE

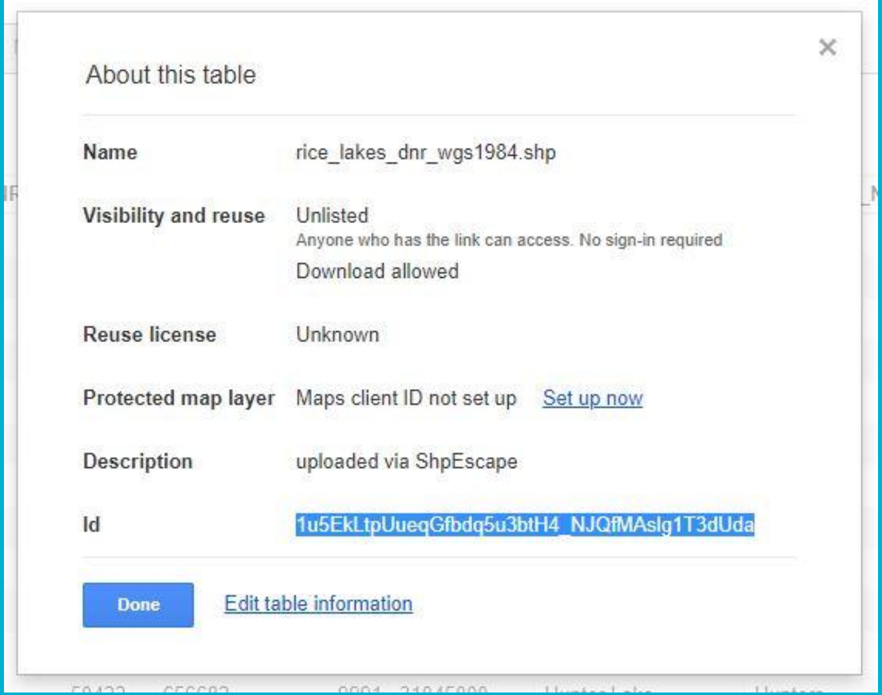
- NAIP imagery
- GEE basic functions



Import features in GEE (Part 3)

Learn how it import existing datasets into GEE

- Interoperability challenges



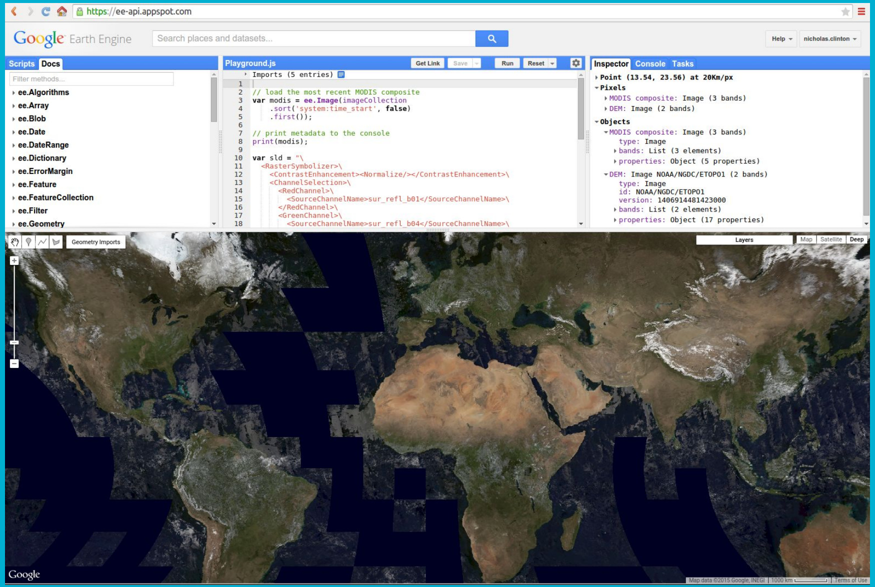
About this table

Name	rice_lakes_dnr_wgs1984.shp
Visibility and reuse	Unlisted Anyone who has the link can access. No sign-in required Download allowed
Reuse license	Unknown
Protected map layer	Maps client ID not set up Set up now
Description	uploaded via ShpEscape
Id	1u5EkLtpUueqGfbdq5u3btH4_NJQfMAslg1T3dUda

Modeling In GEE (Part 4)

Produce a geospatial model in GEE

- Apply advanced functions
- Generate Indices
- Visualize results



```
Imports (5 entries)

// load the most recent MODIS composite
var modis = ee.Image(imageCollection
    .sort('system:time_start', false)
    .first());

// print metadata to the console
print(modis);

var sld = "\
  <RasterSymbolizer>\
    <ContrastEnhancement><Normalize/></ContrastEnhancement>\
    <ChannelSelection>\
      <RedChannel>\
        <SourceChannelName>sur_refl_b01</SourceChannelName>\
      </RedChannel>\
      <GreenChannel>\
        <SourceChannelName>sur_refl_b04</SourceChannelName>\
```

```
Point (13.54, 23.56) at 20Km/px
Pixels
  MODIS composite: Image (3 bands)
  DEM: Image (2 bands)
Objects
  MODIS composite: Image (3 bands)
    type: Image
    bands: List (3 elements)
    properties: Object (5 properties)
  DEM: Image NOAA/NGDC/ETOPO1 (2 bands)
    type: Image
    id: NOAA/NGDC/ETOPO1
    version: 1406914481423000
    bands: List (2 elements)
    properties: Object (17 properties)
```

Break

R

Programming Language that allows for quick access to dataframes

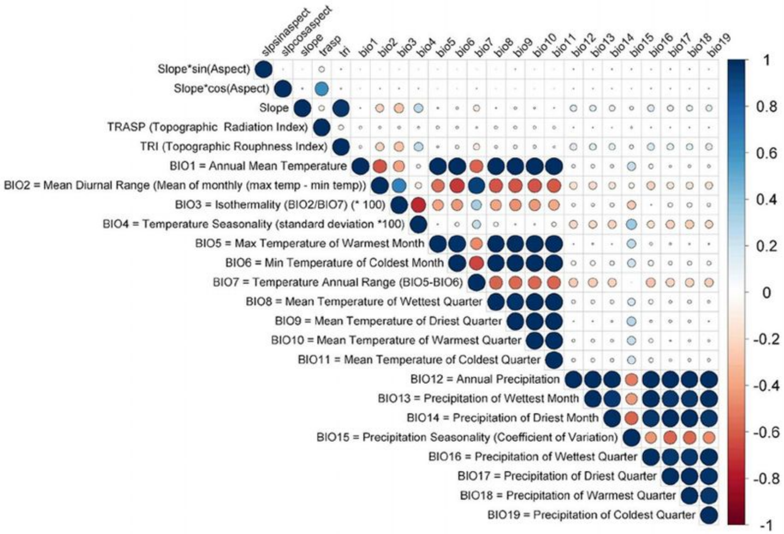
- Developed for statistical purposes so it kinda stand out relative to other computer science languages
- thisIsAThing <- “Why would <- mean =?”
- Call a function on a object   object$function
- Case sensitive
- Paths <- “need\\double\\backslash\\on\\windows”

There is a lot more, we just can’t cover it here

Variable Selection in R (Part 5)

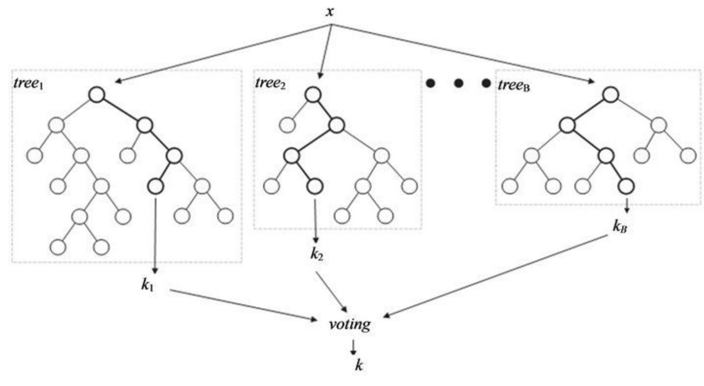
Engage with R packages to inform which predictors are important

- Manipulate dataframes
- Apply algorithms
- View statistical results



Random Forest Modeling in R (Part 6)

Apply a machine learning algorithm to predict



Bad Science

Process not Products

General Notes on Geospatial Modeling

Modeling is full of assumptions, know them

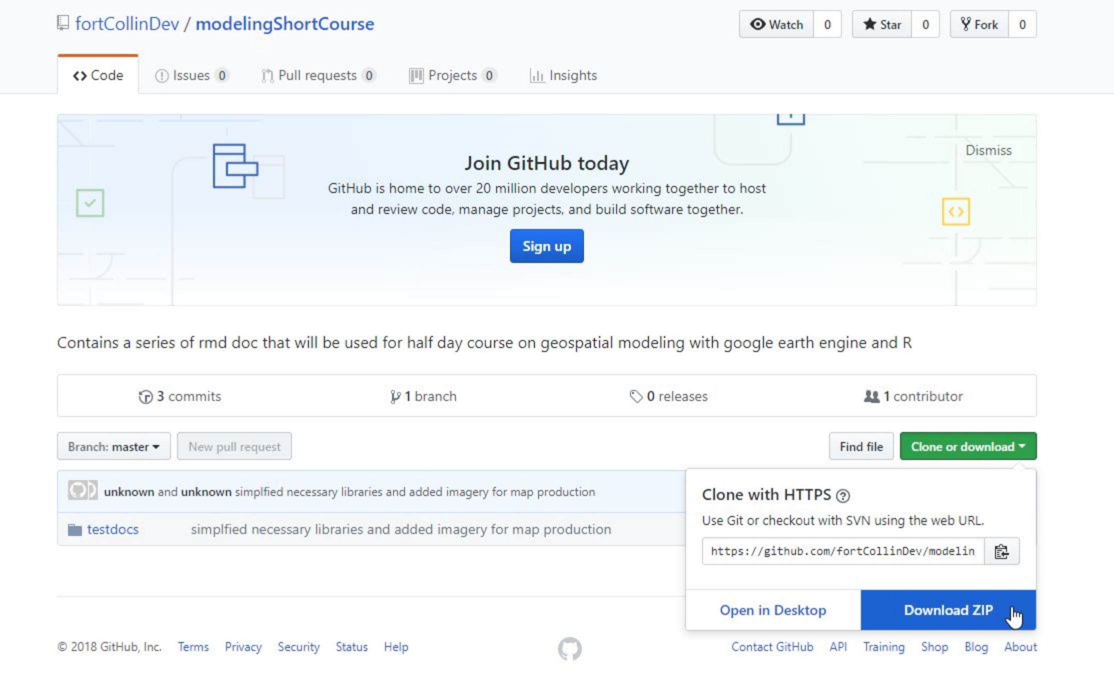
Understanding what it does poorly is equally as important as what it does well

Ideally your model provides more information then was previously available

Understand where the largest uncertainty is and use that to limit the specificity of your output

https://github.com/fortCollinDev

Click on “modelingShortCourse”



fortCollinDev / **modelingShortCourse**

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Contains a series of rmd doc that will be used for half day course on geospatial modeling with google earth engine and R

3 commits | 1 branch | 0 releases | 1 contributor

unknown and **unknown** simplfied necessary libraries and added imagery for map production

testdocs — simplfied necessary libraries and added imagery for map production

Clone with HTTPS

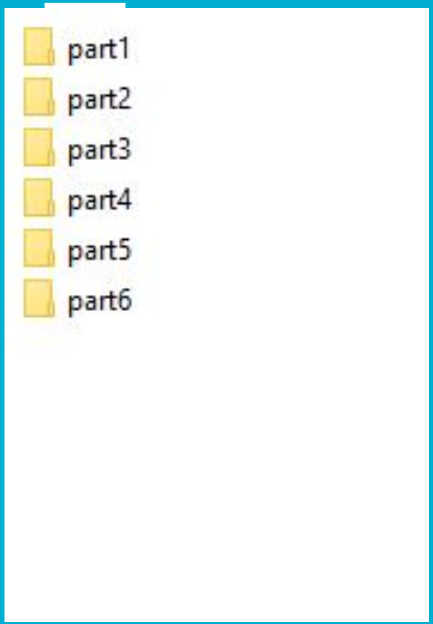
Use Git or checkout with SVN using the web URL.

```
https://github.com/fortCollinDev/modelin
```

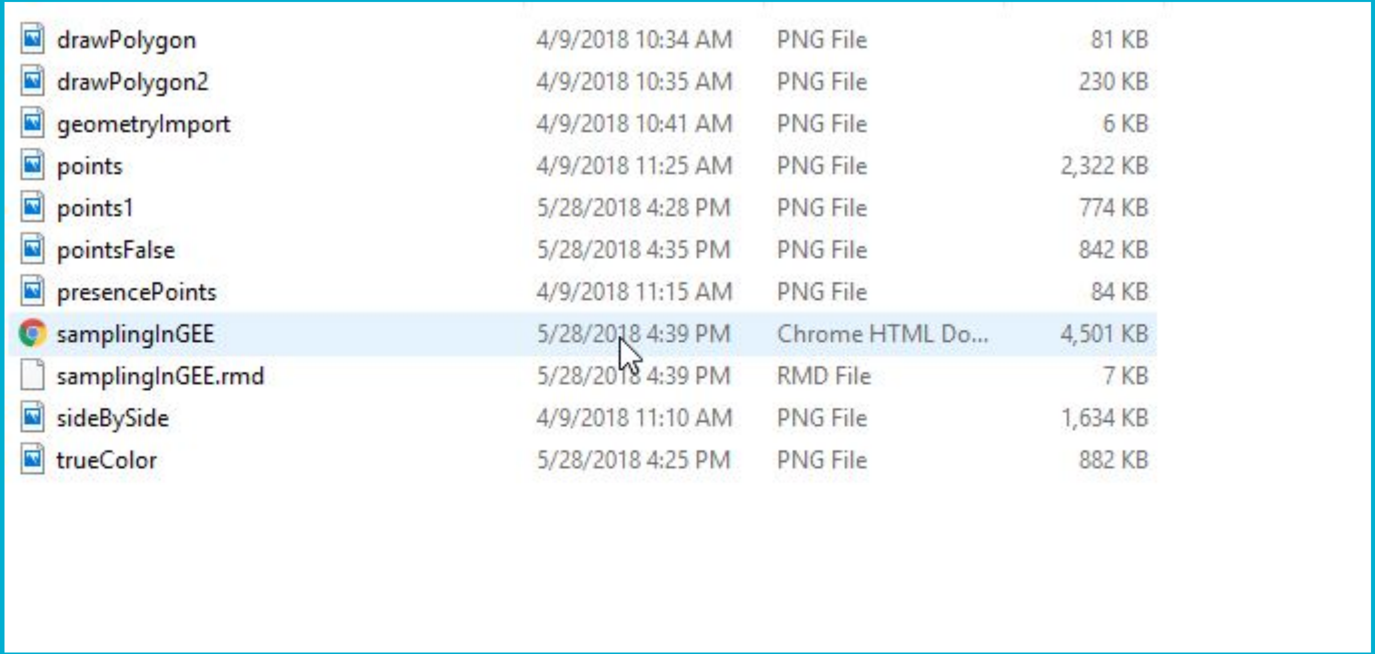
© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

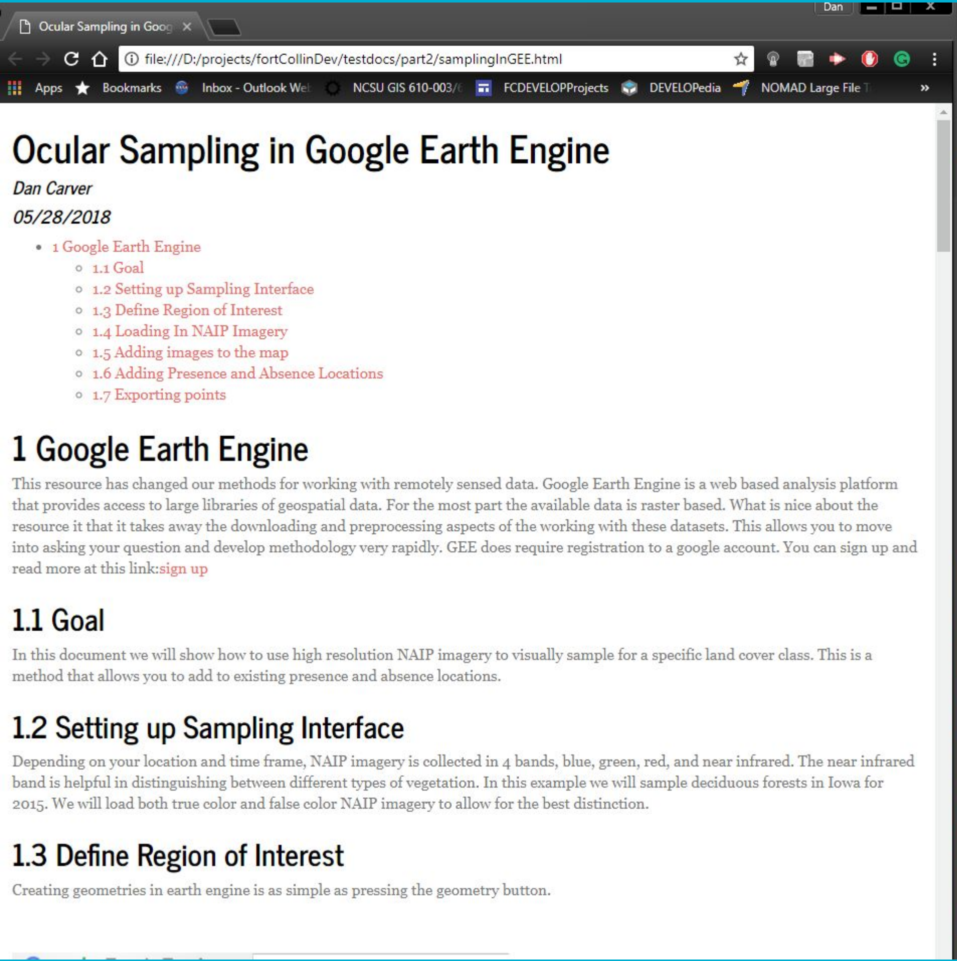
Follow Along by opening the html file



- part1
- part2
- part3
- part4
- part5
- part6



drawPolygon	4/9/2018 10:34 AM	PNG File	81 KB
drawPolygon2	4/9/2018 10:35 AM	PNG File	230 KB
geometryImport	4/9/2018 10:41 AM	PNG File	6 KB
points	4/9/2018 11:25 AM	PNG File	2,322 KB
points1	5/28/2018 4:28 PM	PNG File	774 KB
pointsFalse	5/28/2018 4:35 PM	PNG File	842 KB
presencePoints	4/9/2018 11:15 AM	PNG File	84 KB
samplingInGEE	5/28/2018 4:39 PM	Chrome HTML Do...	4,501 KB
samplingInGEE.rmd	5/28/2018 4:39 PM	RMD File	7 KB
sideBySide	4/9/2018 11:10 AM	PNG File	1,634 KB
trueColor	5/28/2018 4:25 PM	PNG File	882 KB



Ocular Sampling in Google Earth Engine

Dan Carver

05/28/2018

1 Google Earth Engine

This resource has changed our methods for working with remotely sensed data. Google Earth Engine is a web based analysis platform that provides access to large libraries of geospatial data. For the most part the available data is raster based. What is nice about the resource it that it takes away the downloading and preprocessing aspects of the working with these datasets. This allows you to move into asking your question and develop methodology very rapidly. GEE does require registration to a google account. You can sign up and read more at this link:sign up

1.1 Goal

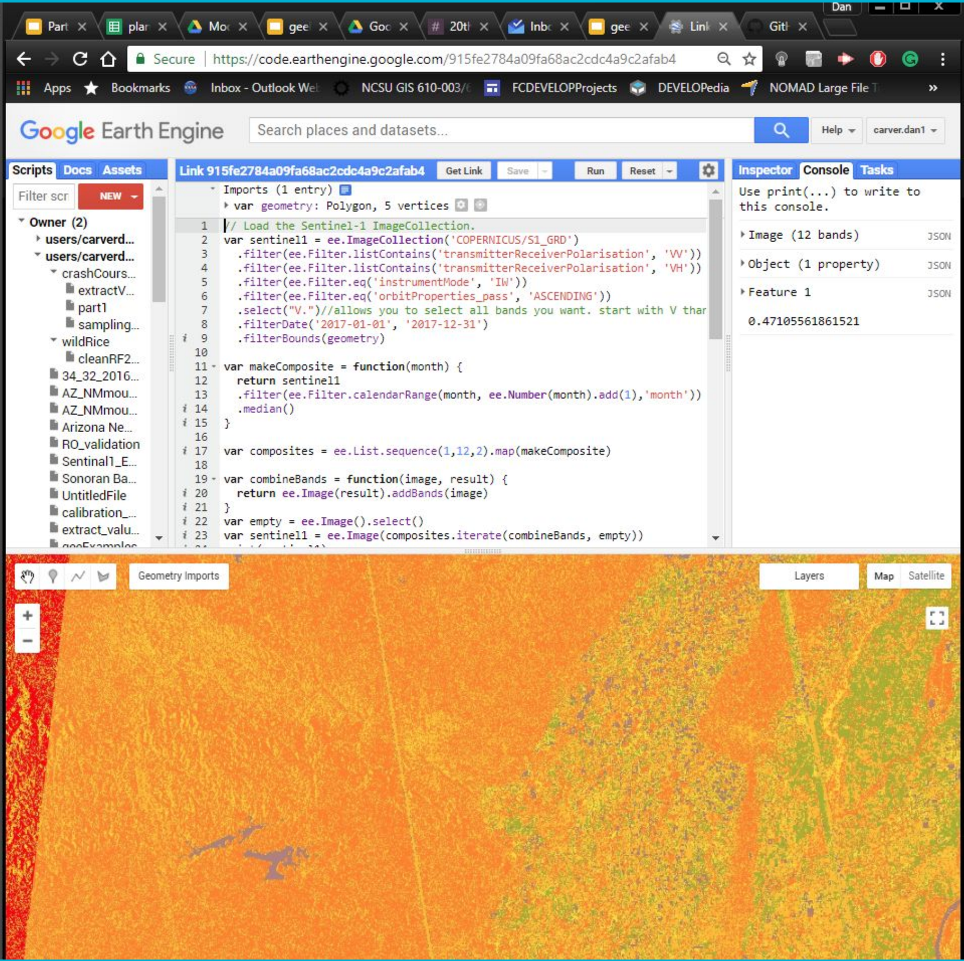
In this document we will show how to use high resolution NAIP imagery to visually sample for a specific land cover class. This is a method that allows you to add to existing presence and absence locations.

1.2 Setting up Sampling Interface

Depending on your location and time frame, NAIP imagery is collected in 4 bands, blue, green, red, and near infrared. The near infrared band is helpful in distinguishing between different types of vegetation. In this example we will sample deciduous forests in Iowa for 2015. We will load both true color and false color NAIP imagery to allow for the best distinction.

1.3 Define Region of Interest

Creating geometries in earth engine is as simple as pressing the geometry button.



```
Imports (1 entry)
  var geometry: Polygon, 5 vertices
```

```
// Load the Sentinel-1 ImageCollection.
var sentinel1 = ee.ImageCollection('COPERNICUS/S1_GRD')
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
  .filter(ee.Filter.eq('instrumentMode', 'IW'))
  .filter(ee.Filter.eq('orbitProperties_pass', 'ASCENDING'))
  .select("V.")//allows you to select all bands you want. start with V thar
  .filterDate('2017-01-01', '2017-12-31')
  .filterBounds(geometry)

var makeComposite = function(month) {
  return sentinel1
  .filter(ee.Filter.calendarRange(month, ee.Number(month).add(1),'month'))
  .median()
}

var composites = ee.List.sequence(1,12,2).map(makeComposite)

var combineBands = function(image, result) {
  return ee.Image(result).addBands(image)
}
var empty = ee.Image().select()
var sentinel1 = ee.Image(composites.iterate(combineBands, empty))
```

```
Use print(...) to write to
this console.

Image (12 bands)        JSON
Object (1 property)     JSON
Feature 1               JSON
0.47105561861521
```

Ask questions