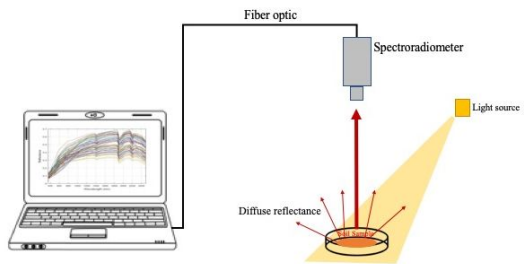


# Vis-NIR Data Exploration

Group 5 - Gabe Alwan, Hendri Winanto, Kevin Hutchins, Andrew Li, Kai Tsuyoshi

# Background

- Soil Spectroscopy
  - Measurement of light absorption (visible, near-infrared) by a soil surface
  - Use the reflectance of the light to determine soil attributes
    - minerals, organic compounds, water content
  - Provides a fast and cheaper alternative to other procedures to determine soil characteristics
- Data is a series of soil samples from the UW-Madison Department of Soil Sciences done in 2019



# Data Overview

- Vis-NIR (Visible-Near Infrared) data
  - views the lower end of the magnetic spectrum in order to see how matter interacts and absorbs the light
  - It is known for its ability to penetrate dense objects, which is why it was used for soil samples
- **Main Variables:**
  - Vis-NIR wavelength (350-2500 nm): length of light wavelength used in treatment
  - SOC%: Soil Organic Carbon percentile value derived from applying the Vis-NIR treatment to each soil sample
  - Depth: Depth in cm. from where the sample was taken from
  - Sample ID: Unique ID for each soil sample (Ex. OB2-1-1)

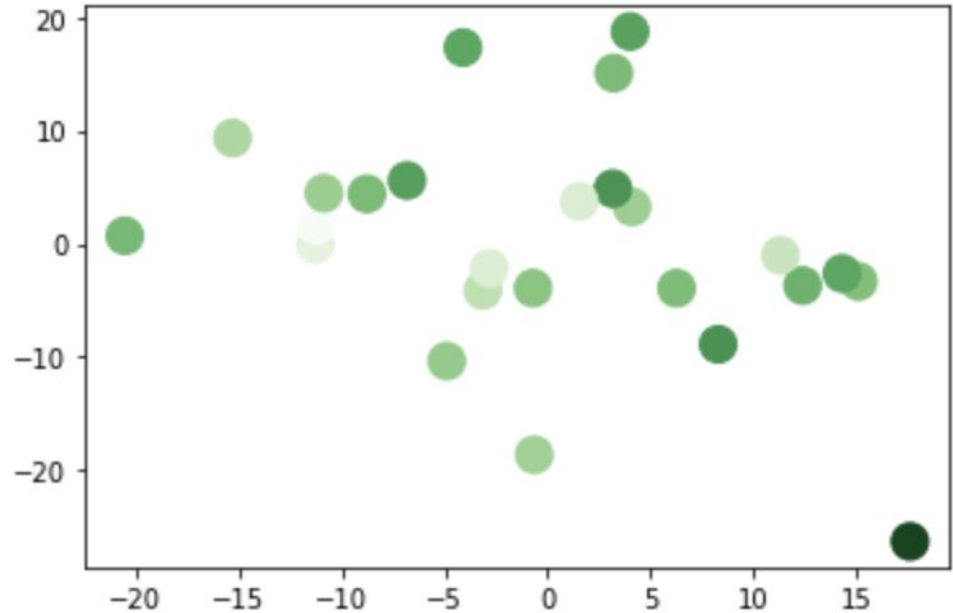


## Data Overview (cont.)

	ID	Depth	pH	Clay (%)	Silt (%)	Sand (%)	SOC (%)	N (%)	350	351	...	2500
0	OB2-1-1	0-10	7.34	18	39	43	1.874400	0.190490	18.438100	18.199500	...	46.418400
1	OB2-1-2	20-0ct	7.44	16	42	42	1.333280	0.163019	18.125900	17.772867	...	40.653467
2	OB2-1-3	20-30	7.40	33	30	36	1.041307	0.108395	17.342200	17.121467	...	35.194333
3	OB2-1-4	30-40	7.68	37	18	45	4.936881	0.135670	19.004733	18.644967	...	36.144633
4	OB2-1-5	40-50	7.84	5	28	66	10.405942	0.063240	24.988100	24.574033	...	48.254733
..	...	...	...	...	...	...	...	...	...	...	...	...
192	OB100-3-4	30-40	6.27	23	22	55	0.384374	0.019708	18.679800	18.453400	...	33.530667
193	OB100-3-5	40-50	6.55	20	17	63	0.243897	0.013222	22.507433	22.223467	...	38.114267
194	OB100-3-6	50-60	6.70	21	24	55	0.220232	0.038898	20.264200	20.063800	...	38.532200
195	OB100-3-7	60-70	6.68	30	53	17	0.212419	0.037179	30.855433	30.692567	...	52.190400
196	OB100-3-8	70-78	7.08	20	25	55	0.150417	0.048146	24.145400	23.865700	...	41.116300

# Assumptions

- Observation location is randomly selected in an area
  - Does not significantly bias any metrics
- Each observation is not independent
- Each sample is conducted such that all are under equal conditions when exposed to the treatment



Location of Observations Based on Long. & Lat.


# Questions

Is it possible to predict the SOC% value of the soil given several different values measured in the soil?

Which algorithm produces higher accuracy and cheaper computation?

---

# Why Soil Organic Carbon (SOC)?

- Carbon is difficult to predict
  - Manual collection and sampling is very labor and time consuming
    - Need many chemical reagents to achieve results
  - SOC is important for the soil health. More carbon in the soil is mainly important for the water-energy balance in the field. By predicting the SOC we can determine good/bad plots of soil which is important for problems in agriculture and climate change.
- 

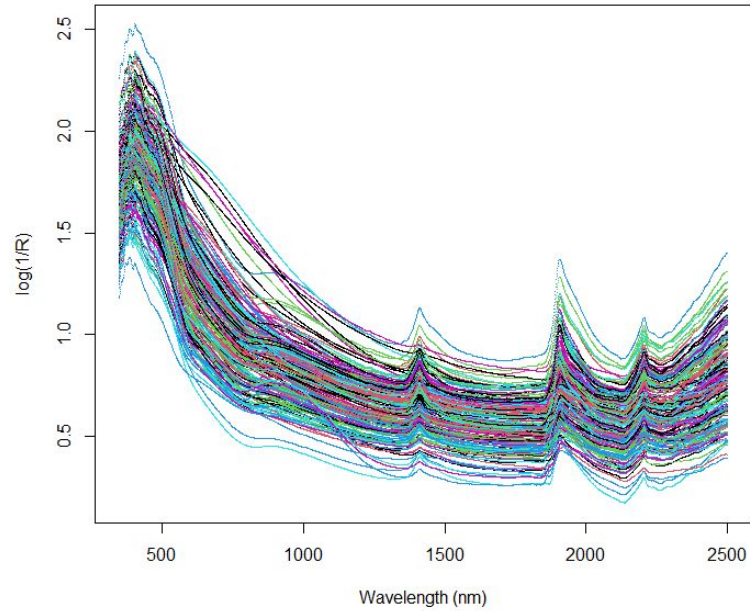
# Methods

- Preprocess the Vis-NIR reflectance that encompasses:
  - Transform the reflectance spectra into absorbance spectra using  $\log(100/R)$  where R is reflectance
  - Smoothing absorbance spectra using Savitzky-Golay digital filter
  - Resample the spectra in range 500 to 2470 with spectra resolution of 10 nm
  - Normalized spectra using standard normal variate
- Split data into 70% training, 30% testing
- Create Model to predict SOC% given the reflectance values as predictors
- Test model to check computation time and cost

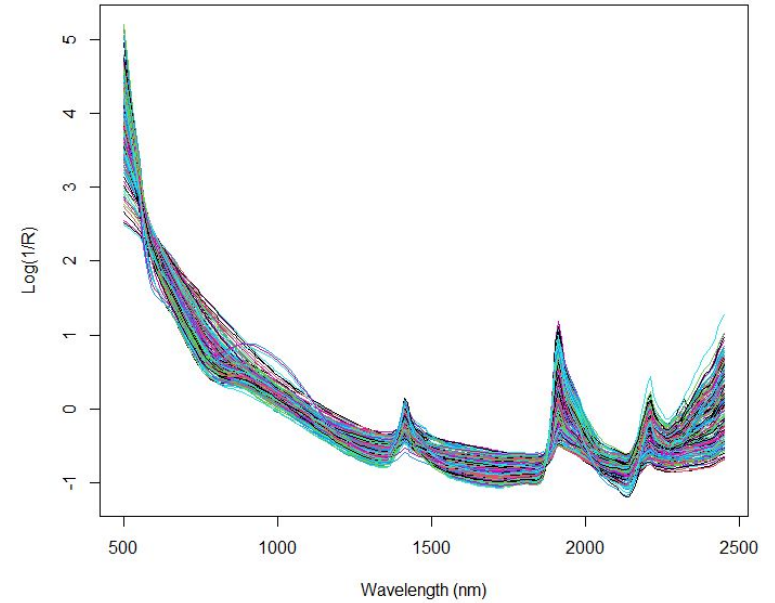




Raw spectra (before preprocessing)



Normalized spectra (after preprocessing)



# Models

- Models to test:
  - Gradient Boosting Regression
  - Random Forest Regression
  - KNN Regression
- Used GridSearch to determine the most accurate model



# Models (cont.)

- Gradient Boosting Regression

- `n_estimators = [10, 50, 100, 150]`
- `learning_rate = [0.01, 0.25, 1, 1.3]`
- `max_depth = [1, 2]`

- Random Forest Regression

- `n_estimators = [10, 50, 100, 200]`
- `max_depth = [1, 2, 3, 4, 5]`

- KNN Regression

- `n_neighbors = [1, 2, 3, 4]`



# Results

- KNN was most accurate at 0.9236
  - `n_neighbor = 1`
  - `MSE = 0.251854`
- KNN was also quickest and cheapest out of the three methods
  - 23.80384063720703 seconds to run `GradientBoostingRegressor(random_state=0)`
  - 65.34340405464172 seconds to run `RandomForestRegressor()`
  - 0.08364105224609375 seconds to run `KNeighborsRegressor()`




# Other Model Results

## Gradient Boosting Regressor:

- Accuracy: `0.041362173077748365`
- Mean Square Error (MSE): `3.1620041103399306`
- The Best Model and Hyperparameter settings:  
`GradientBoostingRegressor(learning_rate=0.01, max_depth=2, n_estimators=10)`

## Random Forest:

- Accuracy: `0.5486261524457836`
  - Mean Square Error (MSE): `1.4888270848320437`
  - The Best Model and Hyperparameter settings:  
`RandomForestRegressor(max_depth=6, n_estimators=10)`
- 



Thank You for Listening!