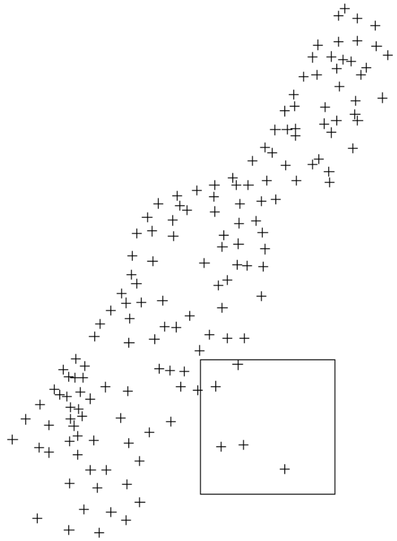
Quiz 8

Create a square polygon data frame with its upper left corner being (180000, 331000), its edge length being 1000, and its attribute being its area. Get the index of the meuse data points that are within the polygon.

```
library(sp)

r1 = cbind(c(180000, 181000, 181000, 180000, 180000), c(331000, 331000, 330000, 330000, 331000))

sr1=Polygons(list(Polygon(r1)),"r1")

sr=SpatialPolygons(list(sr1))

srdf=SpatialPolygonsDataFrame(sr, data.frame(c(1000000), row.names=c("r1")))

plot(srdf)

data(meuse)

coordinates(meuse) = ~x+y

# Show the points and the polygons:

na.omit(over(meuse, srdf))
```



Quiz 9

Generate 50 random training samples and 50 random testing samples from tahoe_highrez.tif (data from lecture 21). Use the training data to build a linear regression model between Lidar tree height (response variable) and tahoe_highrez.tif raster image index tahoe_highrez.1/tahoe_highrez.2 (independent variable). Use the testing data to calculate the correlation coefficient between estimated tree height (from the constructed linear regression model) and observed tree height.

Hint: lecture 21 includes the linear regression analysis between Lidar tree height and NDVI. Instead of NDVI, this quiz will use the ratio index tahoe_highrez.1/tahoe_highrez.2 (first layer/second layer) of tahoe_highrez.tif as the independent variable.

Lecture 22

Raster Analysis 2: Classification

GEOG 489

SPRING 2020

Raster Analysis: Geophysical Variable

Geophysical variables are variables we want to **map** using geospatial data. They are (hopefully) characteristics of the Earth's surface.

Nominal: variables representing discrete categories or classes, e.g. "Land cover class".

Continuous: variables representing ranges of values, e.g. "Percent grass cover".

Raster Models

Geophysical variable maps derived from geospatial data are created through the use of raster models, where:

geophysical variable = f(geospatial data)

If the function ***f*** is known, this function is applied to all pixels (or groups of pixels) in the image, thus producing an estimate of the geophysical variable for every location.

Raster models include continuous variable models and classification models

Continuous Variable Models

Continuous variable models to produce continuous geophysical variables can take many forms, the simplest being a linear relationship.

*percent cover of grass = 1.25 * NIR reflectance*

Thus, if a pixel has a NIR reflectance of 40%, this model estimates the percent cover of grass of that pixel as 1.25 * 40% = 50% covered in grass.

Classification Models

Classification models to produce nominal variables are **rulesets** used to split geospatial responses into different classes. A simple model might be:

Classes: vegetated and non-vegetated

Ruleset:

If NIR reflectance > 50%, then class = vegetated

If NIR reflectance <= 50%, then class = non-vegetated

Thus, if a pixel has a NIR reflectance of 40%, this pixel would be classified as "non-vegetated".

Classification Models

1) Collect training and testing data of classes.

2) Extract pixel values at the location of the training data.

3) Perform transforms on the pixel values (e.g. calculate NDVI and other indices).

4) Try several classification models out with different input predictors.

5) Extract pixel values at the location of the testing data and link with class information.

6) Apply models to test data to predict the variable.

7) Compare the predicted variable vs. the measured variable for the various models and predictor combinations.

8) Pick the best model and...

9) Apply (predict) the model to the raster.

Case Study: Land cover types

Question: what are the land cover types of pixels in the Tahoe high resolution imagery?

Methods:

1. Collect land cover types and spectral features of training sampling locations
2. Perform classification between the field observed land cover types and spectral features derived from raster data at pixel locations where the field data was collected.
3. The classification gives a model (f) of land cover type=f(raster data), so we can apply the model to an entire raster scene, and each pixel will be the estimated land cover type
4. Evaluate the model performance using independent testing samples

Case Study: Land cover types

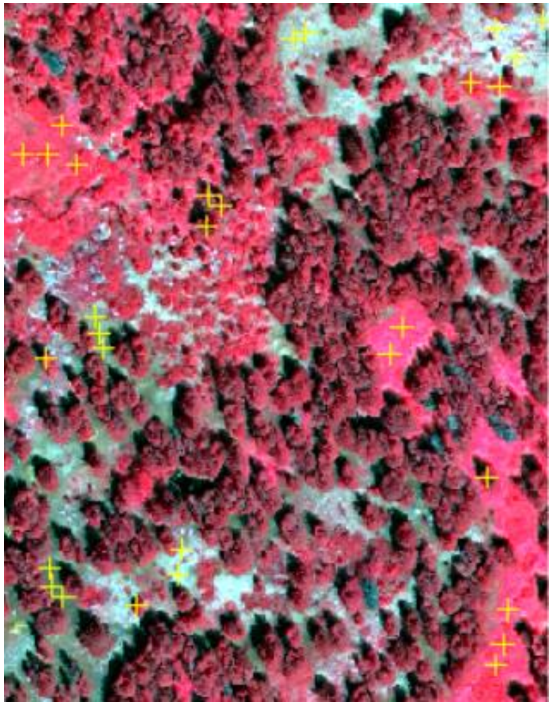
Question: what are the land cover types of pixels in the Tahoe high resolution imagery?

Methods:

1. Collect land cover types and spectral features of training sampling locations

```
# training locations with land cover types
tahoe_highrez_training_points <-
  readOGR(dsn=system.file("external",
package="spatial.tools"),
layer="tahoe_highrez_training_points")
```

```
> tahoe_highrez_training_points
class       : SpatialPointsDataFrame
features    : 30
extent      : -119.9327, -119.9306, 39.28936, 39.29136  (xmin,
coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
variables   : 2
names       : ID,         SPECIES
min values  :  1, Non-vegetation
max values  : 30,            Tree
```



Three land cover types: Non-vegetation, Tree, Shrub

Case Study: Land cover types

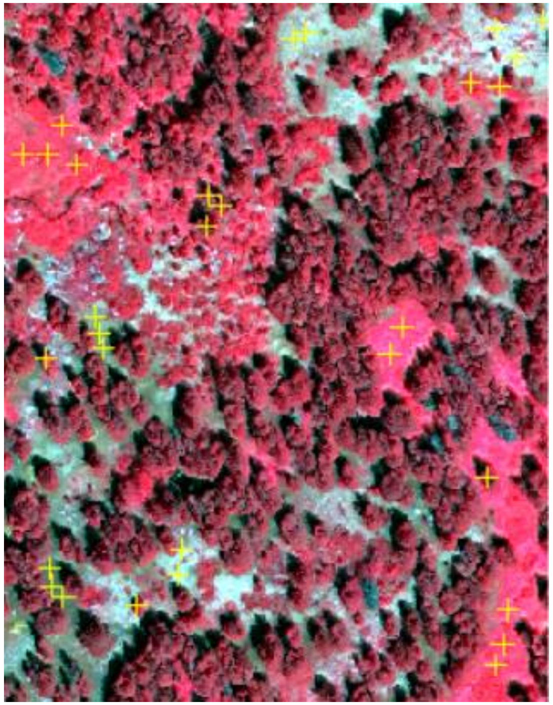
Question: what are the land cover types of pixels in the Tahoe high resolution imagery?

Methods:

1. Collect land cover types and spectral features of training sampling locations

```
# extract the spectral data of training locations
tahoe_highrez_brick <- brick("tahoe_highrez.tif")
tahoe_highrez_training_points_spectral <-
extract(tahoe_highrez_brick,
            tahoe_highrez_training_points,df=TRUE)
```

```
> tahoe_highrez_training_points_spectral
  ID tahoe_highrez.1 tahoe_highrez.2 tahoe_highrez.3
1  1             166              47              74
2  2             165              86             110
3  3             147              28              41
4  4             147              55              68
5  5             119               5               9
```



Case Study: Land cover types

Question: what is the tree height in the Tahoe high resolution imagery?

Methods:

2. Perform classification between the field observed land cover types and spectral features derived from raster data at pixel locations where the field data was collected.

```
> tahoe_highrez_training_points_w_spectra
class       : SpatialPointsDataFrame
features    : 30
extent      : -119.9327, -119.9306, 39.28936, 39.29136  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
variables   : 6
names       : ID,        SPECIES, ID.1, tahoe_highrez.1, tahoe_highrez.2, tahoe_highrez.3
min values  :  1, Non-vegetation,    1,              84,               5,               9
max values  : 30,           Tree,   30,             255,             255,             255
```

Response

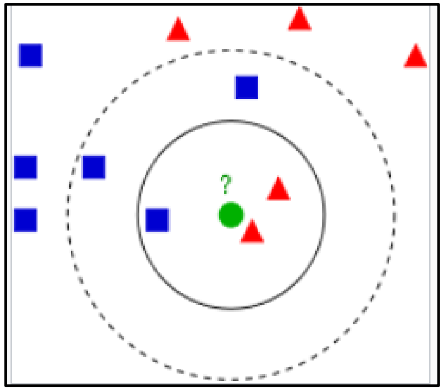
Independent

- K-Nearest Neighbor Classification
- Classification Tree (CART)
- Random Forest

Case Study: Land cover types

k- Nearest Neighbor Classification

This algorithm calculates each class' centroid in multidimensional space, and then determines the class of each unknown value based on its Euclidean distance proximity to the class



The test sample (green dot) should be classified either to the first class of blue squares or to the second class of red triangles.
If $k = 3$ (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle.
If $k = 5$ (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

Case Study: Land cover types

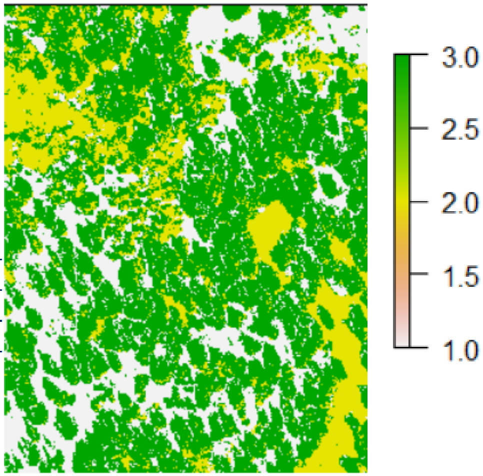
k- Nearest Neighbor Classification

This algorithm calculates each class' centroid in multidimensional space, and then determines the class of each unknown value based on its Euclidean distance proximity to the class

k-Nearest Neighbor Classification in R:

tahoe_knn <- knn(train=tahoe_knn_training, test=tahoe_knn_test, cl=tahoe_knn_training_classes)

```
> tahoe_knn
 [1] Tree           Tree           Shrub          Non-vegetation Tree           T
 [8] Non-vegetation Tree           Shrub          Non-vegetation Tree           T
[15] Non-vegetation Shrub          Shrub          Tree           Tree           T
[22] Tree           Tree           Tree           Tree           Tree           T
[29] Shrub          Tree
Levels: Non-vegetation Shrub Tree
```



Predicted land cover classes

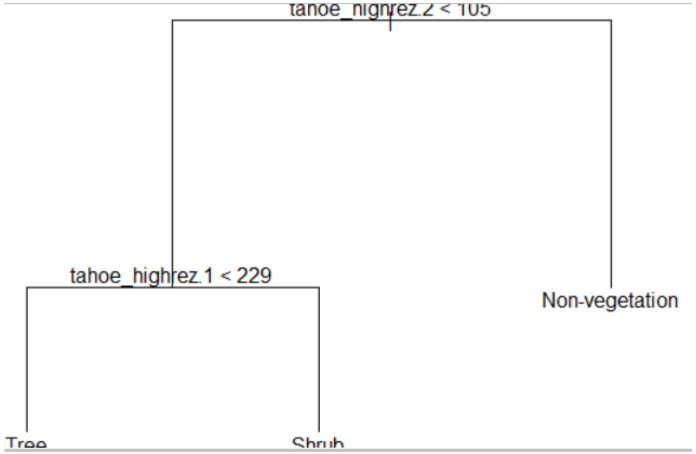
Case Study: Land cover types

Classification trees (CART)

A basic classification tree takes the predictor variables and recursively partition them into binary splits to best predict the response variable (a class, in our case).

A CART output model is a set of binary splits on the predictor variables (e.g. Band 1 > 50? Y/N) that terminate in the predicted category (hence, it looks like a tree).

```
tahoe_tree <- tree(SPECIES ~
    tahoe_highrez.1 + tahoe_highrez.2 + tahoe_highrez.3,
    data=tahoe_highrez_training_points_w_spectra)
```

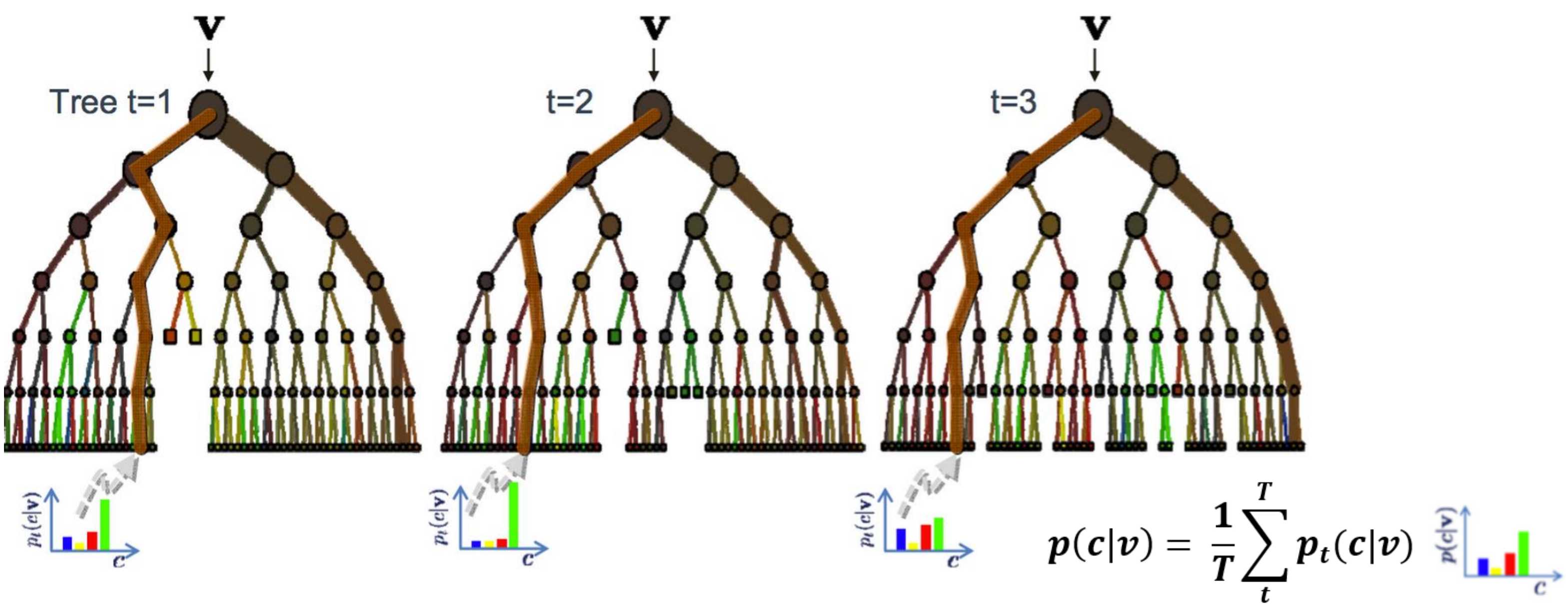


Case Study: Land cover types

Random Forest

RandomForests are an extension of CARTs that improve on many of the shortcomings of a CART, most notably overfitting. It is also considered one of the best classifiers available.

The way it basically works is that it generates a set of trees ("forest") by dropping samples from the training data and running a new tree.



Case Study: Land cover types

Random Forest

RandomForests are an extension of CARTs that improve on many of the shortcomings of a CART, most notably overfitting. It is also considered one of the best classifiers available.

The way it basically works is that it generates a set of trees ("forest") by dropping samples from the training data and running a new tree.

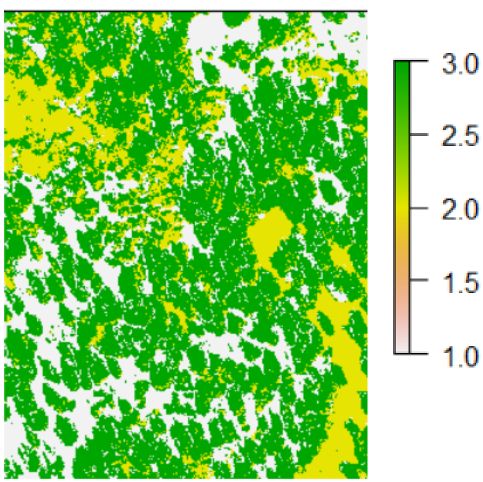
```
# Random Forest
tahoe_randomForest <- randomForest(SPECIES ~
        tahoe_highrez.1 + tahoe_highrez.2 + tahoe_highrez.3,
        data=tahoe_highrez_training_points_w_spectra,
        importance=TRUE)
```

Case Study: Land cover types

Question: what are the land cover types of pixels in the Tahoe high resolution imagery?

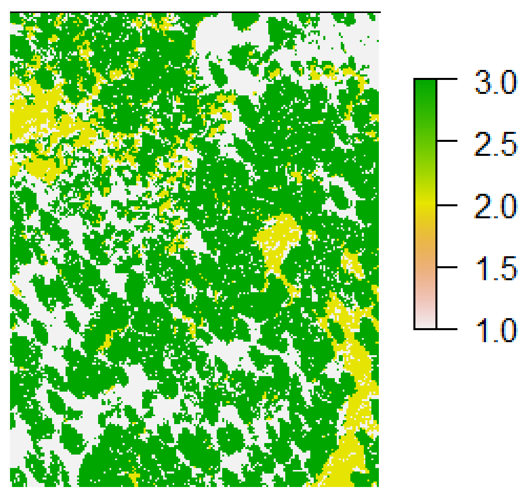
Methods:

3. The classification gives a model (f) of land cover type=f(raster data), so we can apply the model to an entire raster scene, and each pixel will be the estimated land cover type



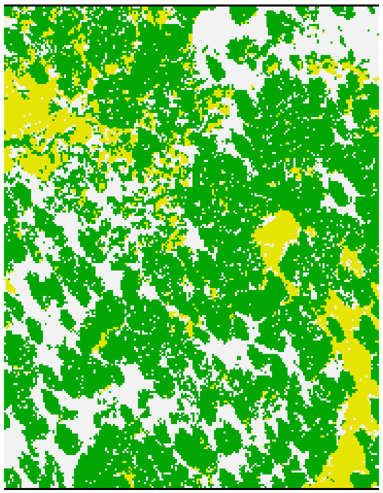
k- Nearest Neighbor

tahoe_knn_raster <- calc(tahoe_highrez_brick, knn_calc_function)



Classification Tree

tahoe_tree_raster <- predict(tahoe_highrez_brick, tahoe_tree,type="class")



Random Forest

tahoe_tree_raster <- predict(tahoe_highrez_brick, tahoe_randomForest)

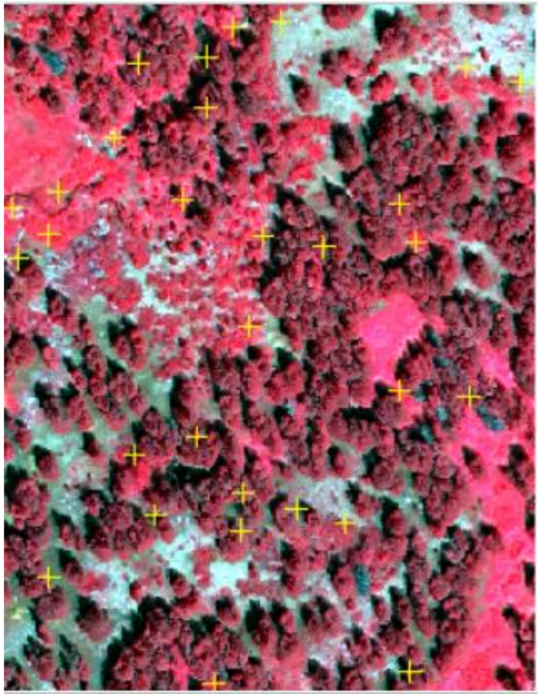
Case Study: Land cover types

Question: what are the land cover types of pixels in the Tahoe high resolution imagery?

Methods:

4. Evaluate the model performance using independent testing samples

```
# Stratified random samples (10 samples for each class):
tahoe_highrez_test_points <-
  sampleStratified(lidar_height_class,
size=10, sp=TRUE)

# Use samples to extract the spectral data
tahoe_highrez_test_points_spectral <- extract(
  tahoe_highrez_brick, tahoe_highrez_test_points,
df=TRUE)
```



Case Study: Land Cover Types - Evaluation

Predicted vs Estimated land cover types (confusion matrix)

```
> tahoe_knn_confusionMatrix
Confusion Matrix and Statistics

                Reference
Prediction       Non-vegetation Shrub Tree
  Non-vegetation              3     0    2
  Shrub                       1     6    0
  Tree                        6     4    8

Overall Statistics

               Accuracy : 0.5667
                 95% CI : (0.3743, 0.7454)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 0.007223

                  Kappa : 0.35
```

k- Nearest Neighbor
(overall accuracy: 56.67%
Kappa: 0.35)

```
> tahoe_tree_confusionMatrix
Confusion Matrix and Statistics

                Reference
Prediction       Non-vegetation Shrub Tree
  Non-vegetation              2     1    2
  Shrub                       1     5    0
  Tree                        7     4    8

Overall Statistics

               Accuracy : 0.5
                 95% CI : (0.313, 0.687)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 0.04348

                  Kappa : 0.25
```

Classification Tree
(overall accuracy: 50%
Kappa: 0.25)

Case Study: Land Cover Types - Evaluation

Predicted vs Estimated land cover types (confusion matrix)

```
> tahoe_randomForest_confusionMatrix
Confusion Matrix and Statistics

                Reference
Prediction        Non-vegetation Shrub Tree
  Non-vegetation               2     1    2
  Shrub                        1     5    0
  Tree                         7     4    8

Overall Statistics

               Accuracy : 0.5
                 95% CI : (0.313, 0.687)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 0.04348

                  Kappa : 0.25
```

Random Forest
(overall accuracy: 50%
Kappa: 0.25)

Case Study: Land cover types

Raster to Vector: To create polygons from the class information

tahoe_randomForest_polys <-
  rasterToPolygons(tahoe_tree_raster, dissolve=TRUE)

Subset out the trees only:

tahoe_randomForest_trees <-
  tahoe_randomForest_polys[
    tahoe_randomForest_polys$layer==3,]

```
> tahoe_randomForest_trees
class       : SpatialPolygonsDataFrame
features    : 1
extent      : -119.9328, -119.9306, 39.28922, 39.29141
coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=
variables   : 1
names       : layer
value       :     3
```