Mini-project: Predicting Distribution of a Species

```
library(ggplot2); theme_set(theme_light())
library(dplyr)
library(readr)
```

Introduction

The goal of the mini-project is to predict the presence of a species (such as a tree/shrub/etc) from presence-only observations. This is useful when trying to mangage populations of species, reducing impacts of invasive species, and preserving endangered species. Predicting habitats in which it is present and grows quickly is important for managing it effectively.

The goal is to predict the distribution from *presence-only* data. This is a machine learning problem but it is different from the standard classification/regression setting. It is an example of a presence-only dataset. An example of what actual presence data looks like can be seen on EDDMapS. The data generally consists of *positive* samples only. Most observations are reported by volunteers. People generally do not report when they do not observe the plant and therefore there are no (or very few) negative samples.

Presence-only data is more common in machine learning than you may expect. Positive samples are often easier to collect than negative ones. Some obvious settings that are similar are predicting the presence of insects, pests, and diseases. A lot of businesses have access to positive samples (their customers) and have little access to negative samples (people who are not their customers). Models that deal with presence-only data are also common in other areas of machine learning, such as natural language modeling and (inverse) reinforcement learning.

In this project, we will use New England's landscape. It is divided into rectangular cells $\mathcal{L}$. Each cell is identified by a unique `cellid`, and for each cell we know its latitude and longitude. The training data is the number of observations n_l reported that have been reported for each cell $l \in \mathcal{L}$. It is important to note that there may be significant population in a cell, but no-one has reported an observation from there. There are 19 biologically-relevant predictors available from WorldClim which can be used to generalize the observation from the current cells.

The goal if this project is to predict observed populations in cells that are not a part of the training set. Biologists instead usually care about predicting the population in a cell, or its suitability for the species, but ground truth data on that is difficult to gather. The project has two following similar sub-problems.

1. A synthetic subproblem. The population in each cell is computed from a simulated model which is based on a real plant species. The synthetic problem makes is possible to have ground truth data and describe exactly the process that generates the observations. The observations in this synthetic model are based only on a subset of the 19 predictors; there are no other considerations.
2. A real subproblem. This is a real dataset derived from EDDMaps for a mystery plant species in New England. There may be no perfect fit for this model. Other features in addition to the ones provided may be useful when fitting the model. Spatial considerations, such as the presence of the species in nearby-cells, may also play a role in this part of the project.

Objective and evaluation

The goal is to predict, for each cell in the landscape not included in the training set, number of observations in the test set. The output should be a *number* for each cell not included in the training set.

Datasets

The following files are available:

1. `landscape.csv`: Landscape data: cellid, latitude, longitude, bio-features
2. `synthetic_training.csv`: Presence-only data for the synthetic problem. It contains cell id, and the number of reports (`freq`)
3. `real_training.csv`: Presence-only data for the real problem. It contains cell id, and the number of reports (`freq`)

The test sets are, of course, secret for now.

Landscape

The landscape dataset contains the list of all cells, their coordinates, and the biologically-relevant variables. The dataset is available as the `landscape.csv` file:

```
landscape <- read_csv("landscape.csv", col_types = cols(cellid = col_integer()))
limits <- function(x){c(min(x), max(x))}
limits_lat <- limits(landscape$lat)
limits_long <- limits(landscape$long)
```

If you would like to see how the landscape data was generated, please see the `landscape.csv` file.

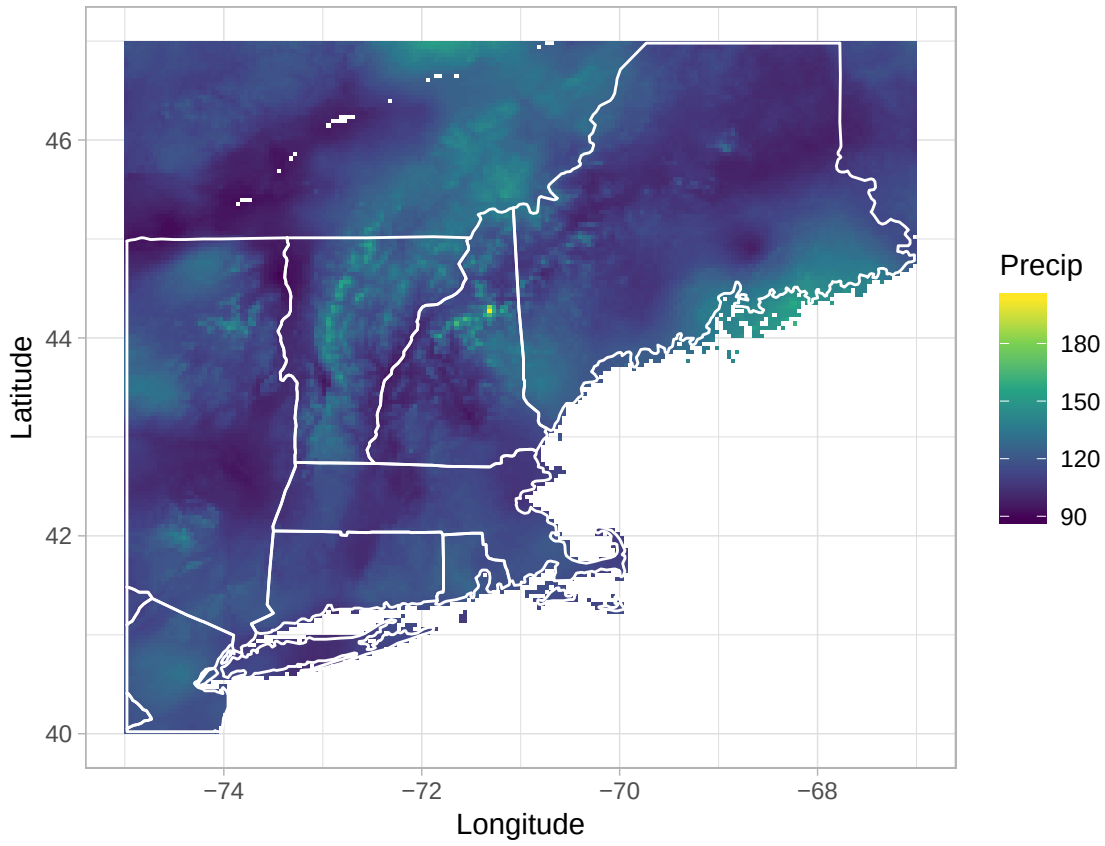
Description of the variables is at: http://worldclim.org/bioclim. Briefly, their meaning is a follows.

Feature	Description
BIO1	Annual Mean Temperature
BIO2	Mean Diurnal Range (Mean of monthly (max temp - min temp))
BIO3	Isothermality (BIO2/BIO7) (* 100)
BIO4	Temperature Seasonality (standard deviation *100)
BIO5	Max Temperature of Warmest Month
BIO6	Min Temperature of Coldest Month
BIO7	Temperature Annual Range (BIO5-BIO6)
BIO8	Mean Temperature of Wettest Quarter
BIO9	Mean Temperature of Driest Quarter
BIO10	Mean Temperature of Warmest Quarter
BIO11	Mean Temperature of Coldest Quarter
BIO12	Annual Precipitation
BIO13	Precipitation of Wettest Month
BIO14	Precipitation of Driest Month
BIO15	Precipitation Seasonality (Coefficient of Variation)
BIO16	Precipitation of Wettest Quarter
BIO17	Precipitation of Driest Quarter
BIO18	Precipitation of Warmest Quarter
BIO19	Precipitation of Coldest Quarter

Lets plot the landscape data to get some sense of what it looks like. This code plots some of the climatic variables and the outline of the states. This is the variable `bio13`:

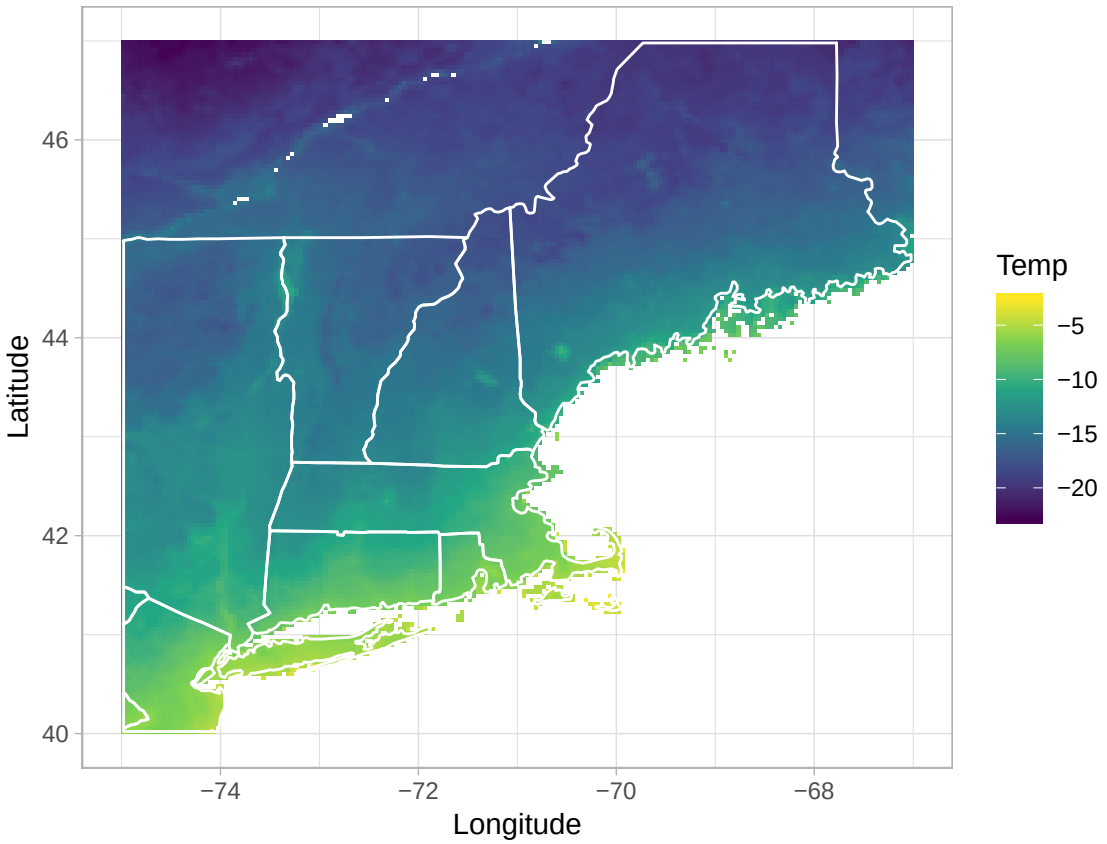
```
states <- map_data("state", regions = ".", xlim = limits_long, ylim = limits_lat,
                   lforce = "e")
ggplot(landscape, aes(x = long, y = lat)) +
  geom_raster(aes(fill = bio13)) +
  geom_polygon(aes(group = group), data = states,
```

```
              fill = NA, color = "white") +
  scale_fill_viridis_c() +
  scale_alpha_continuous(range=c(0.2,0.8)) + coord_fixed() +
  labs(x = "Longitude", y = "Latitude", fill = "Precip")
```



And this is `bio6`:

```
ggplot(landscape, aes(x = long, y = lat)) +
  geom_raster(aes(fill = bio6)) +
  geom_polygon(aes(group = group), data = states,
               fill = NA, color = "white") +
  scale_fill_viridis_c() +
  scale_alpha_continuous(range=c(0.2,0.8)) + coord_fixed() +
  labs(x = "Longitude", y = "Latitude", fill = "Temp")
```

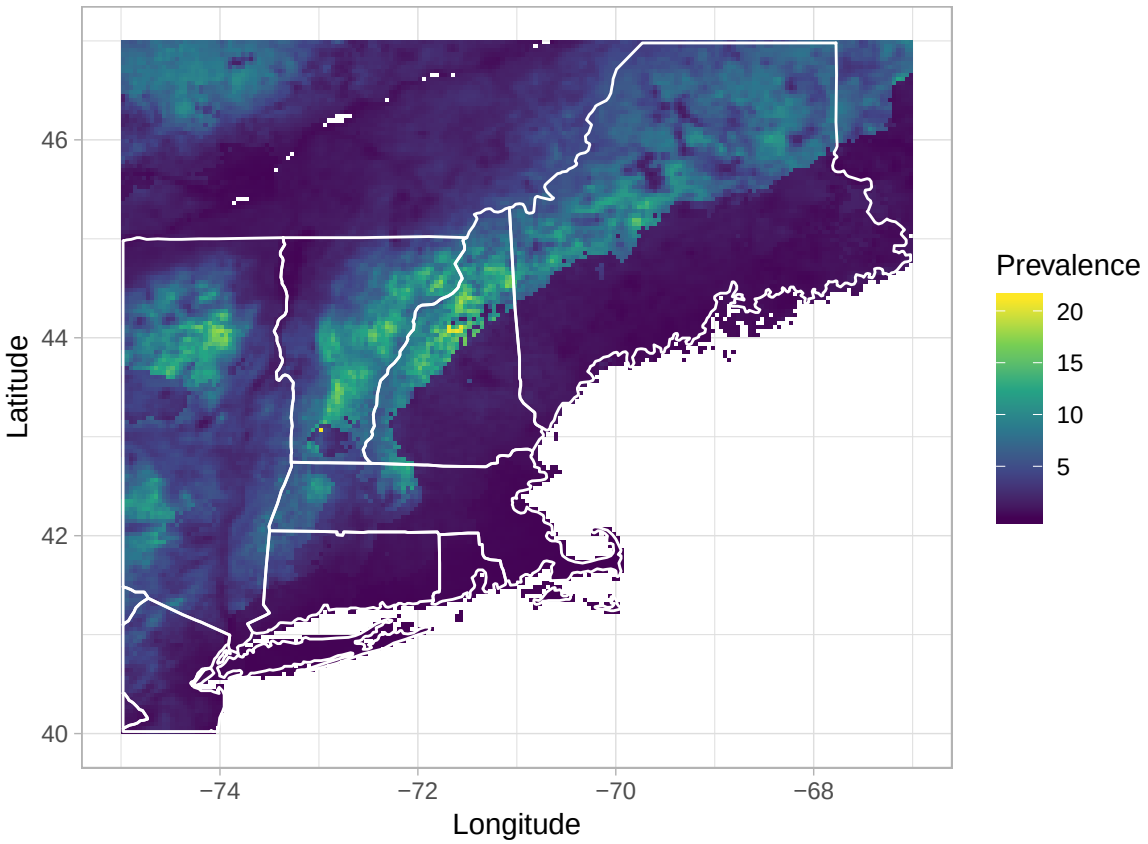


Synthetic (Fake) Presence Data

The code in this section demonstrates how the synthetic data is generated. The underlying presence data generated by this code is *different* from the data that is provided. The purpose of this code is to give you some information that you can use to design your algorithms.

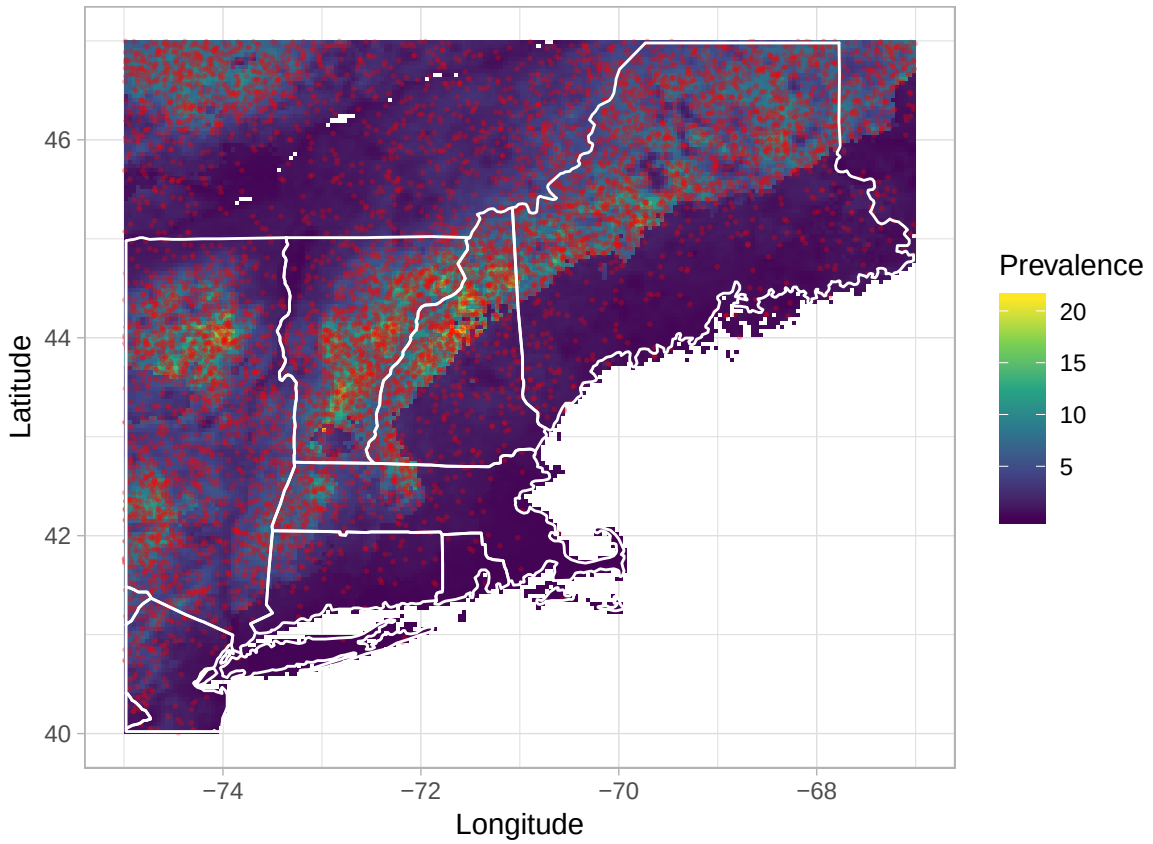
The synthetic presence data uses only features `bio2`, `bio3`, `bio5`, `bio8` and `bio16`. The dependence on these features may be wither linear or non-linear. These features are used to generate true *prevalence* of the species in the landscape for each. Think of this as the number of all specimen for each cell.

```
syn_feature_matrix <- model.matrix(~bio2+bio3+bio5+bio8+bio16, landscape)
beta <- c(-2,1,0.1,-0.5,0.1,0.00001) # made up beta!!
prevalence <- exp(syn_feature_matrix %*% beta)
synthetic_true <- data.frame(cellid = landscape$cellid, prevalence = prevalence)
ggplot(synthetic_true %>% full_join(landscape %>% select(cellid,long,lat), by = "cellid"),
       aes(x = long, y = lat)) +
  geom_raster(aes(fill = prevalence)) +
  geom_polygon(aes(group = group), data = states, fill = NA, color = "white") +
  scale_fill_viridis_c() +
  coord_fixed() + labs(x = "Longitude", y = "Latitude", fill = "Prevalence")
```



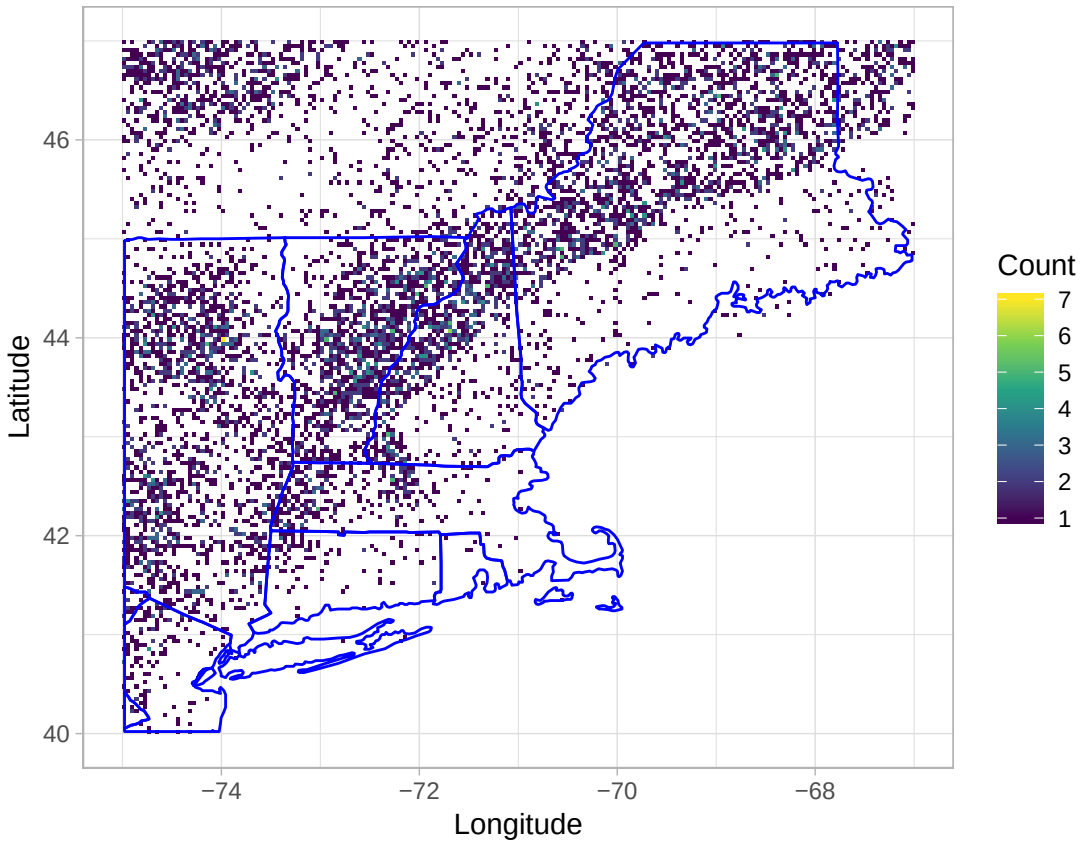
The synthetic observation data is sampled based on the prevalence of the species in each cell. The procedure corresponds to sampling each specimen with some small fixed probability. That mean that cells that have higher prevalence are likely to have more observations. This a common assumption with species models. We can plot the individual observations (since the observations are on top of each other, they are "jittered").

```
set.seed(1000)
synthetic_observations <- data.frame(
  cellid = sample(synthetic_true$cellid, 8000, replace = TRUE,
                  prob = synthetic_true$prevalence))
ggplot(synthetic_true %>% full_join(landscape %>% select(cellid,long,lat), by = "cellid"),
       aes(x = long, y = lat)) +
  geom_raster(aes(fill = prevalence)) +
  geom_jitter(data = synthetic_observations  %>%
                inner_join(landscape %>% select(cellid,long,lat), by = "cellid"),
              size = 0.3, alpha = 0.3, color = "red") +
  geom_polygon(aes(group = group), data = states, fill = NA, color = "white") +
  scale_fill_viridis_c() +
  coord_fixed() + labs(x = "Longitude", y = "Latitude", fill = "Prevalence")
```



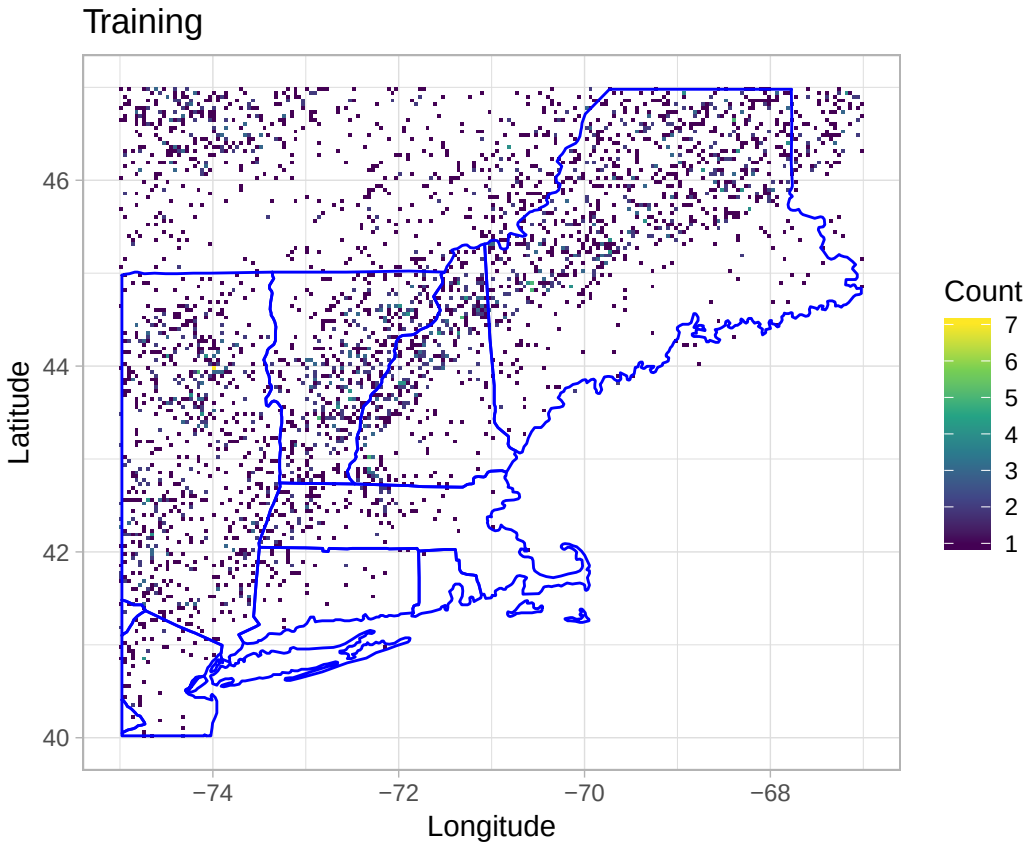
The results are aggregated by the cell to make it more convenient working with them.

```
synthetic_all <- synthetic_observations %>% group_by(cellid) %>% summarize(freq = n())
ggplot(synthetic_all %>% inner_join(landscape %>% select(cellid,lat,long), by="cellid"),
       aes(x = long, y = lat)) +
  geom_raster(aes(fill = freq)) +
  geom_polygon(aes(group = group), data = states, fill = NA, color = "blue") +
  scale_fill_viridis_c() +
  coord_fixed() + labs(x = "Longitude", y = "Latitude", fill = "Count")
```



Finally, about half of the observed cell are included in the training data (provided). The other half is retained for the test set (unavailable).

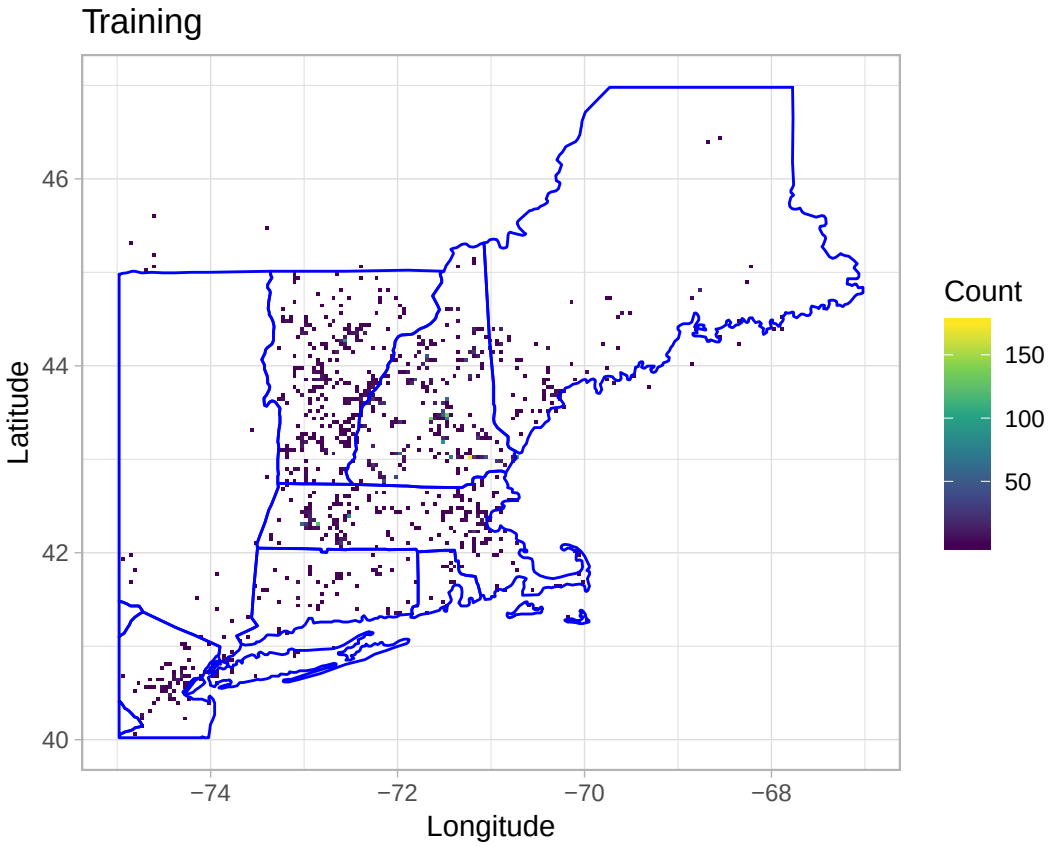
```
synthetic_train <- sample_frac(synthetic_all, 0.5)
synthetic_test <- anti_join(synthetic_all, synthetic_train, by="cellid")
ggplot(synthetic_train %>% inner_join(landscape %>% select(cellid,lat,long), by="cellid"),
       aes(x = long, y = lat)) +
  geom_raster(aes(fill = freq)) +
  geom_polygon(aes(group = group), data = states, fill = NA, color = "blue") +
  scale_fill_viridis_c() +
  coord_fixed() + labs(x = "Longitude", y = "Latitude", fill = "Count", title="Training")
```



Real Reported Presence Data

The real dataset is generated similarly. We take the reported observations, determine which cell they it should be included in, and then aggregate the counts by cell.

```
real_train <- read_csv("real_train.csv", col_types = cols(cellid = col_integer()))
ggplot(real_train %>% inner_join(landscape %>% select(cellid,lat,long), by="cellid"),
       aes(x = long, y = lat)) +
  geom_raster(aes(fill = freq)) +
  geom_polygon(aes(group = group), data = states, fill = NA, color = "blue") +
  scale_fill_viridis_c() +
  coord_fixed() + labs(x = "Longitude", y = "Latitude", fill = "Count", title="Training")
```



Submission and Evaluation

The goal is to predicted the number of observations for each cell in the landscape that matches the test as closely as possible. **The evaluation metric is as follows.** Each correctly-predicted observation **earns 1 point**, while each incorrectly-predicted observation **loses 0.3** points. For examples, lets say that cell 1 in the test set has 5 observations. If you predict 3 then you gain 3 points, and if you predict 7 you lose 0.6 (2*0.3) points. The predicted counts can be real numbers, not just integers.

You should submit the following files:

1. A very *short* document (Rmd, Jupyter, Latex, PDF) describing the method(s) used and results (1-2 pages including plots)
2. `synthetic_pred.csv` CSV file with two columns: `cellid` and `pred_freq` for cells that are NOT in the training set `synthetic_train.csv`
3. `real_pred.csv`: CSV file with two columns: `cellid` and `pred_freq` for cells that are NOT in the training set `real_train.csv`

Observations in Test Sets

The number of observations in the test sets are:

- Synthetic: 5013
- Real: 1981

Example Evaluation

Lets say I figure out the precise coefficients β for the synthetic problem. I can then generate predictions directly from the estimated prevalence data. I also need to remove all cells that are in the training data from

the prediction.

```
predicted <- data.frame(
  cellid = sample(synthetic_true$cellid, 3000, replace = TRUE,
                  prob = synthetic_true$prevalence))
predicted <- predicted %>% group_by(cellid) %>% summarize(pred_freq = n())
predicted <- anti_join(predicted, synthetic_train, by = "cellid")
```

The score can now be computed by joining the results with the test set and computing the difference. The statement below also filters out any extraneous columns, cells that are not in the landscape, and cells from the training set. It assumes that all missing predictions are 0.

```
combined <-
  predicted %>% select(cellid, pred_freq) %>%
    anti_join(synthetic_train, by = "cellid") %>%
    left_join(landscape %>% select(cellid), by="cellid") %>%
    full_join(synthetic_test, by="cellid") %>%
    tidyr::replace_na(list(pred_freq = 0, freq = 0))
sum(pmin(combined$freq, combined$pred_freq)) -
    0.3 * sum(pmax(combined$pred_freq - combined$freq, 0))
```

```
## [1] 96.7
```

Simple Analysis and Ideas

Do not forget about feature transformations.

You can try any technique that you like. Feel free to focus on either the real or the synthetic problem. There are many more considerations that are important when it comes to the real data sets. Some examples:

- The density of the data is subject to density of the human population.
- The assumption that there are more observations when the plant is more prevalent may not be true. If it is growing everywhere nobody bothers to report it.
- There are spatial effects due to how the plant spreads.
- Other features in addition to the ones provided may be useful. You are on your own where to get them.

To get started, think of using some of the methods that we have covered. Can you use them in a problem like this? Perhaps you may want to think about the expressing the likelihood of this dataset and base your method on that.

If you would like to explore some of the literature that has been written on the topic, I recommend these papers as a good starting point:

- Phillips, S. J., Dudik, M., & Schapire, R. E. (2004). A maximum entropy approach to species distribution modeling. In International Conference on Machine Learning (ICML).
- Royle, J. A., Chandler, R. B., Yackulic, C., & Nichols, J. D. (2012). Likelihood analysis of species occurrence probability from presence-only data for modelling species distributions. Methods in Ecology and Evolution, 3, 545–554.
- Hastie, T., & Fithian, W. (2013). Inference from presence-only data; the ongoing controversy. Ecography, 36(8), 864–867.
- Fithian, W., & Hastie, T. (2013). Finite-sample equivalence in statistical models for presence-only data. Annals of Applied Statistics, 7(4), 1917–1939.
- Renner, I. W., & Warton, D. I. (2013). Equivalence of MAXENT and Poisson Point Process Models for Species Distribution Modeling in Ecology. Biometrics, 69(1), 274–281.