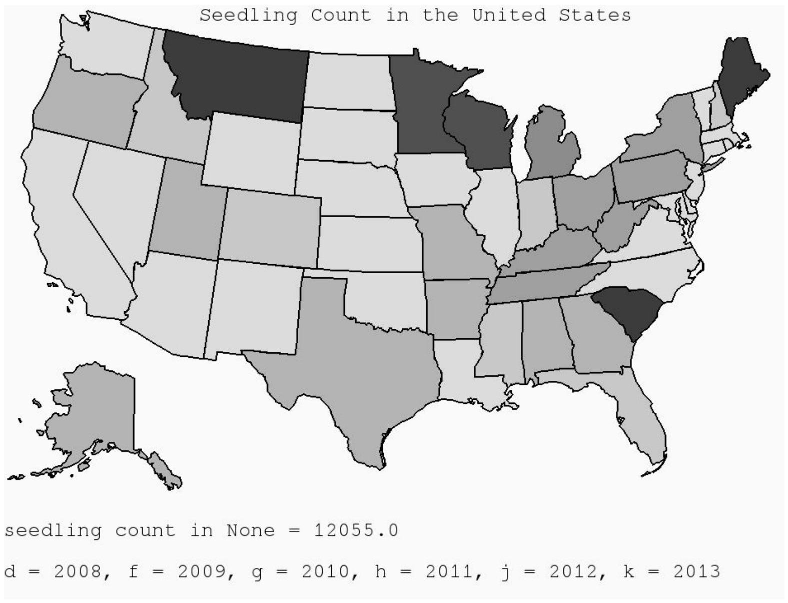
Project Overview:

We did a data visualization project using seedling count data for the United States. When the user mouses over a state, the number of seedlings in that state appears at the bottom of the screen. When the user presses down certain keys on the keyboard, the map switches to show data from different years.

Results

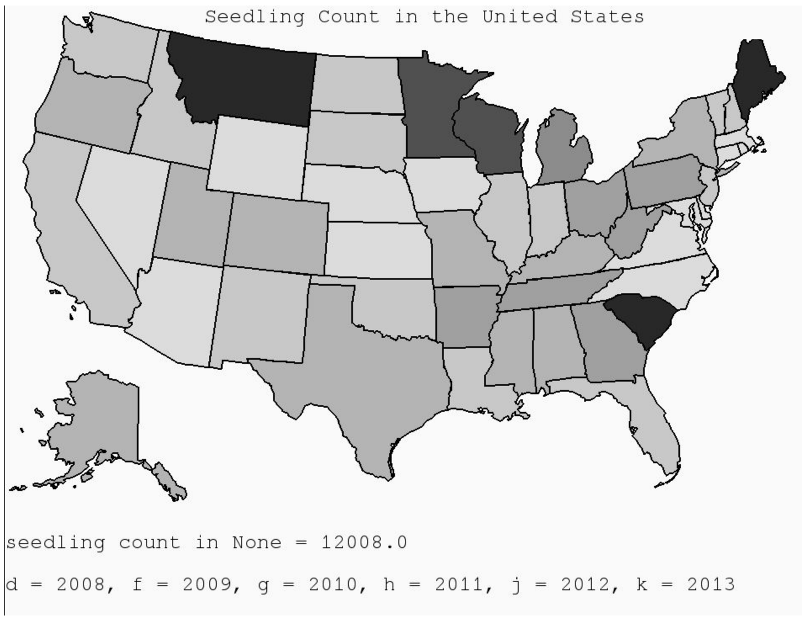
We were able to get reasonable results from this project in the three days since we started using pygame. The map shows our seedling data in the years 2008 (Fig. 1), 2009 (Fig. 2), 2010 (Fig. 3), 2011 (Fig. 4), 2012 (Fig. 5), and 2013 (Fig. 6). The text at the bottom of the map changes depending on which state the user is mousing over.

Fig. 1



```
seedling count in None = 12055.0
d = 2008, f = 2009, g = 2010, h = 2011, j = 2012, k = 2013
```

Fig. 2



```
seedling count in None = 12008.0
d = 2008, f = 2009, g = 2010, h = 2011, j = 2012, k = 2013
```

Fig. 3

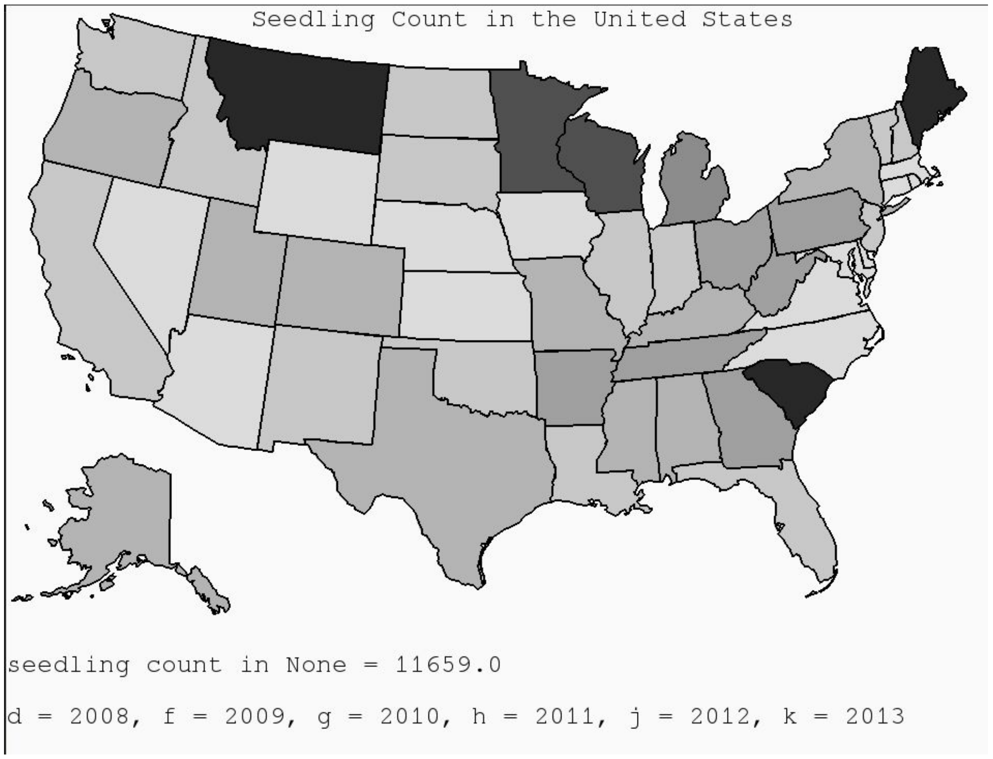

Seedling Count in the United States
seedling count in None = 11659.0
d = 2008, f = 2009, g = 2010, h = 2011, j = 2012, k = 2013

Fig. 4

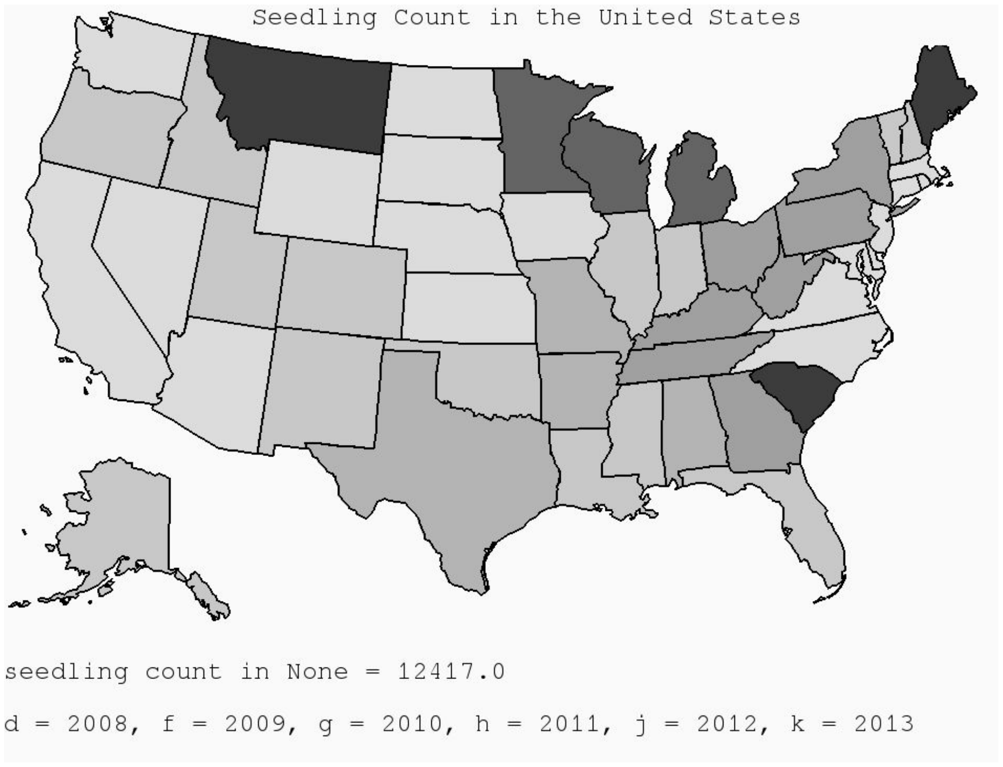

Seedling Count in the United States
seedling count in None = 12417.0
d = 2008, f = 2009, g = 2010, h = 2011, j = 2012, k = 2013

Fig. 5

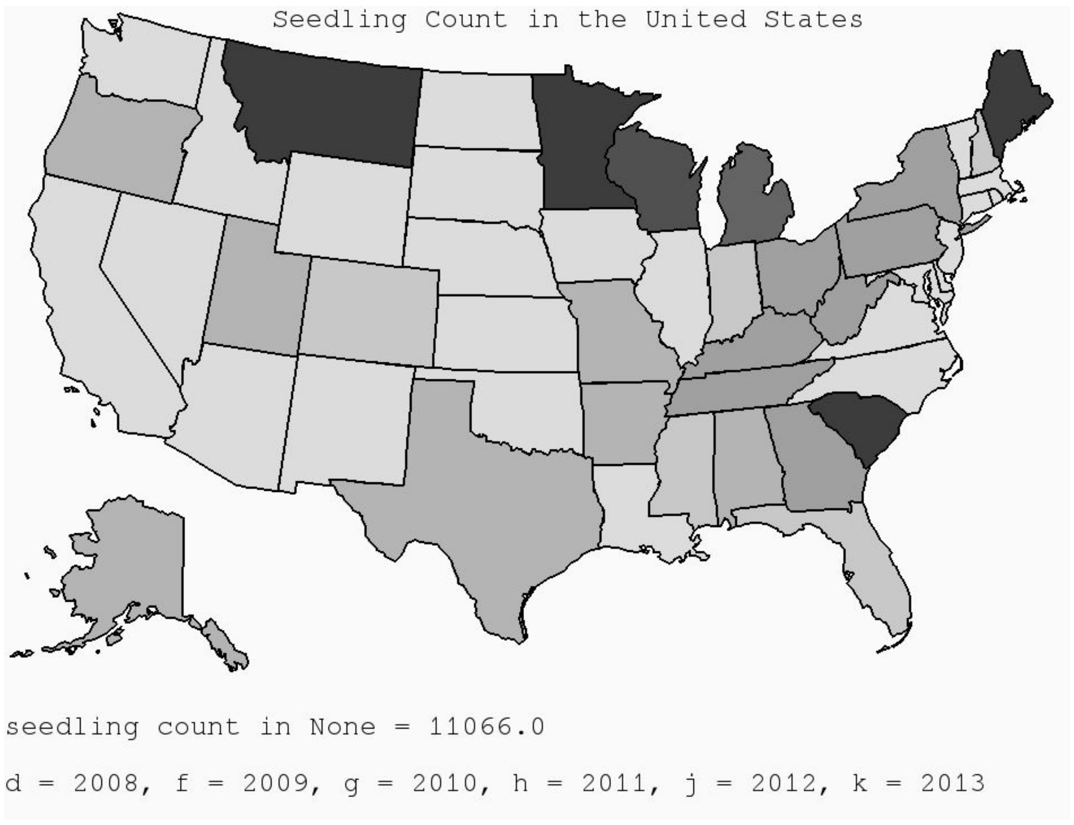
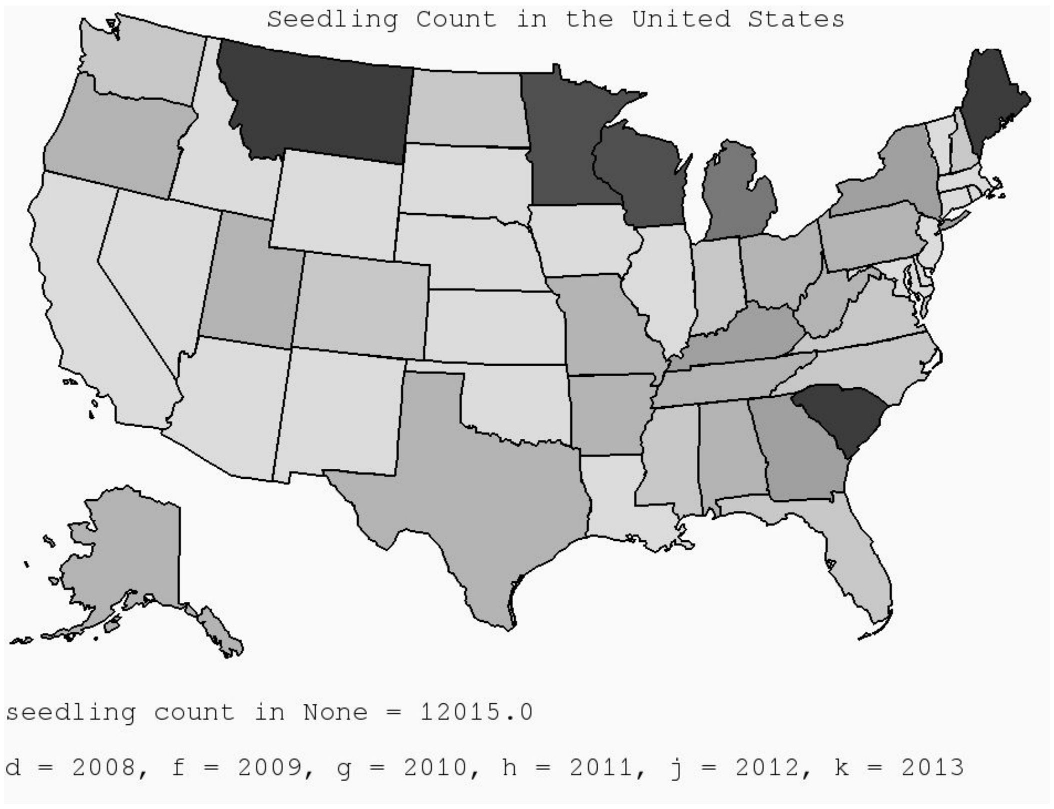

Seedling Count in the United States
seedling count in None = 11066.0
d = 2008, f = 2009, g = 2010, h = 2011, j = 2012, k = 2013

Fig. 6


Seedling Count in the United States
seedling count in None = 12015.0
d = 2008, f = 2009, g = 2010, h = 2011, j = 2012, k = 2013

Implementation

One of our main functions served to draw the map of the states. Inside the function, the list of colors is established, the csv file containing yearly data is updated, and the states are drawn and filled in depending on the current filename. This function uses the helper functions remap_interval and the panda file converter function.

We took the remap_interval function from the computational art project and used it as a helper function in order to create a series of values equal to the number of elements in the colors list. In this context, it served to assign a color to a seedling count value by putting the seedling count value within the interval bounded by the color list. We had another helper function to convert the panda reads of csv files to usable lists from something called a ‘series’.

A second main function serves to ouput which state the mouse is in. This function updates the file name and then uses the polygon helper function that checks if a point is inside a polygon to check which state the mouse is in.

The only code not inside a function initializes the pygame window, tells it which csv file to initially use, makes explanatory text appear, and outlines which keys are associated with each year. There was then a while loop that ran the function, kept track of the state and updated the year data shown depending on which key was pressed.

We would have liked to work in classes for this project, but due to issues at the beginning of the project, we had to put this map together in less than four days. Due to this time constraint we decided to work primarily with functions.

Reflection:

We spent an awful lot of time (about 10 hours cumulatively) figuring out which module to use by trying out a few different ones (bokeh, plotly, matplotlib), writing code with them, and installing literally everything ever (it might have just felt that way because of how often installations would fail and then take half an hour to troubleshoot). This is the main aspect of the project that went poorly, and it was really frustrating to not be able to truly start our project until we were directed to use pygame on Monday. We got an awful lot of conflicting advice from different sources about what to use and how to make our project happen, so until Paul said, “Just use pygame.” in class on Monday. Before that, we came up with semi-working code in plotly and matplotlib, and then we ended up scrapping it to use pygame. Once we started writing with pygame, things were much more productive and we enjoyed working on the project a lot more. We tended to work on separate aspects of the project at the same time, for example one person handling the data while the other person tried to solve a problem in the code, one person debugging and the other trying to move forward, or both people working on separate aspects of the code. This was a really great strategy for our needs (getting the project done in three days), but it wasn’t the best way for both of us to learn a lot. The project was definitely appropriately scoped to be done in a week, but not appropriately scoped to be done in three days.

We both agreed that if we had to do this again, we would just make a game. We definitely would have been able to learn more of what we were supposed to learn in less time with a game. There was so much more support for that path, so it would have been worth it even though we both find data visualization a lot more interesting.

We really wish we had known:

1. The basic guidelines of which specific tools to use to do our project
2. How to do this with classes. For some reason we didn’t quite realize what exactly the model-view-controller framework was or how it applied to the project until we made Patrick really sad at his ninja hours on Tuesday night. That might be mostly our fault, but we still think it wouldn’t hurt to offer

more clarity/support in that area. For example, the pygame map example code on piazza, while very helpful, did not use classes.