Project Design

Topic Selection

Given that this project focuses on webscraping and linear regressions, I brainstormed topics that had both a continuous output variable to estimate, and publicly available data to scrape (a few ideas had to be ruled out because their data was too neatly organized into a csv). When exploring the idea of estimating hiking trail density, I quickly realized there wasn't enough data to properly explore this topic. However, I discovered there was much data to analyze the adjacent topic of estimating traffic at national parks. The information was displayed online but only through the National Park Services' webpage, which made it an ideal topic choice for learning webscraping, and it had a clear output variable I could reasonably expect to estimate.

Web Scraping

My webscraping began with producing an MVP of scraping one park's data. The webpage made heavy use of div tags that were not hashed, so they would change each time the browser was opened. After much time exploring BeautifulSoup and Selenium, I eventually found a different way to select the information I desired – by searching for a specific pattern that consistently surrounded the target text.

After finishing scraping all the data for the first park I had about 500 observations of my output variable, and I decided that I needed more rows of data for my project. I wanted good practice with webscraping, so I worked on making my code iterable for all parks, and after some time managed to write code that would in one session scrape a park's data, select and open the webpage for the next park from a dropdown menu, scrape its data, and repeat until it had finished with every national park. The code is quite iterable, and can easily scrape for all National Historical Sites, National Monuments, and so forth.

Data Cleaning and Manipulation

Fortunately, the data I scraped came with certain information I thought would be quite useful: the year and month each observation was collected, the park's name that corresponded to each observation, and the yearly total of visitors for that park. Whereas originally each row contained 12 observations, I organized the data into a pandas dataframe such that each row had a unique observation, and found a way to preserve other identifying information about each observation (month, year, and so forth).

Aside from these variables, I created a few other columns of inputs – only a few of which made it into my final product. I created a column with the number of national holidays during the month of each observation, a column that lagged the number of people who had visited that same park during the entire prior year, a column that tracked the change in yearly total visits for that park from the prior two years, and others.

Modeling

My minimum MVP involved regressing against these inputs to estimate the number of visitors. However, with 51 national parks scraped along with 12 months, I had to be careful about creating too many input variables for a robust model. In order to avoid having to use the park name in my regression, I used the lagged number of people who visited that park the year

prior. This served as a weighting variable that would indicate the general size for a given observation's park. I used the change in total visits per year as a growth variable, hoping to capture trends of more or less visitation for each park. Finally, I used the month dummies, as I theorized that people were more likely to visit the park during the spring and summer months. Ultimately, my model produced a very robust regression that could explain 95% of the variance in the data (taken from the adjusted R^2).

Aside time spent regressing against a few different input variables I discarded, much of my time modeling was spent trying to best capture the monthly trends. Graphically, plotting the visitors and the months displayed a parabolic relationship, so I attempted a polynomial fit. I created a column that numerically indexed the months, and regressed against it as a second-degree polynomial. However, I found that the R^2 decreased significantly, and opted not to try higher degree polynomials as they did not seem to match the underlying relationship.

Cross-Validation

I chose to have a holdout sample of about 30% of my data, so as to test the robustness of my regression and ensure I wasn't overfitting. Because my data is a time series, my training set wasn't random; rather, it included all of the observations up until the year 2006 (again, about 70% of my data). The score of the predictions for the test set was .95, which validated its ability to predict the number of visitors.

Tools

Selenium
Statsmodels
SciKit-Learn
Matplotlib

Algorithm

OLS Regression

What I'd do differently next time

I spent a lot of time trying to figure out how to best plot monthly trends – in part because I wanted to get a robust regression, but mostly because I wanted to practice transformations (especially polynomials). I don't consider it time wasted as I learned how to make a singular input variable into a polynomial, but next time I'd allocate more time towards gathering other data I wanted to test as features. These would mostly be weather variables: monthly highs + lows and days with precipitation for each park.