

Mapping Species Sightings via Webscraping

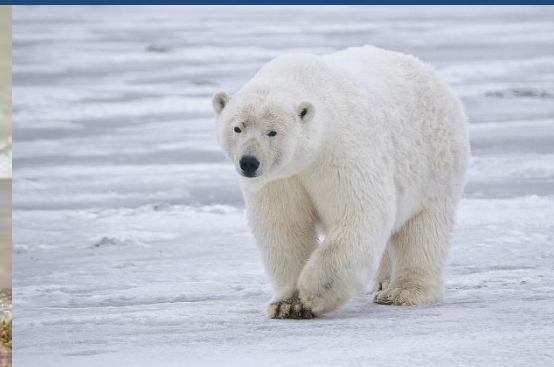
Alex Rufrano & Kate Blackwell

Stony Brook University,
Stony Brook, NY, USA



Objective

Utilize data sourced from publicly available photos opportunistically collected by the public to assist with updating known distribution of species



Expertise and Lessons Learned

- **Alex Rufrano (10%):**

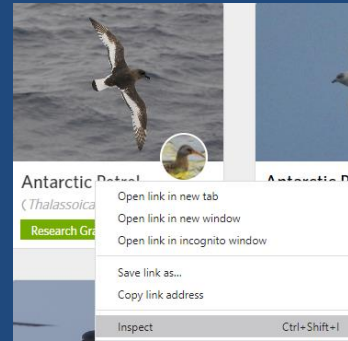
- Increased familiarity with HTML and Pandas
- How to handle “None” datatype produced by Geocoder
- Overcome Type() errors and HTML errors in Geocoder
- Create Google Maps API key and extensions of GMAPS
- Learned how to use Selenium, BeautifulSoup, Requests, and GeoPy modules

- **Kate Blackwell (0%):**

- Increased comfort with Pandas module
- Basics of webscraping and GMAPS
- Using Shapely, shapefiles, and Area modules to work with geometric data
- Fully implementing own Python script in Jupyter notebook

Part 1: Techniques & Tools

- **Step 1:** Find desired piece of data and inspect element by right-clicking on element.
 - Determines location of data in HTML code in web browser. EX: Find species' name and location
- **Step 2:** Use Selenium to import HTML code
 - Scroll through entire page(s) to collect all data
 - Determine which tag and class the data is located
- **Step 3:** Use BeautifulSoup to extract HTML data
 - find_all function for tags and classes
- **Step 4:** Create dataframe using Pandas to store and clean data
- **Step 5:** Use Geocoder via GeoPy to convert species locations to global coordinates
- **Step 6:** Use Google Maps (GMAPS) for Python to plot coordinates on heat map



```
<a class="display-name comname" href="/observations/64248418" ng-show="taxon.id" target="_self"> == $0
  <span class="rank ng-binding ng-hide" ng-show="(!secondaryName() || user.prefs.scientific_name_first) && showRank()">...</span>
  "Antarctic Petrel
  "
```



```
soup.find_all("a", class_ = "display-name comname")
[<a class="display-name comname" href="/observations/64248418" ng-show="taxon.id" target="_self"><span class="rank ng-binding ng-hide" ng-show="(!secondaryName() || user.prefs.scientific_name_first) && showRank()">Species<span> </span></span>
>Antarctic Petrel
</a>]
```



```
#Create the DataFrame and clean the data.
locname = pd.DataFrame({"Location" : loc_clean,
locname
```



	Location	Species
0	Drake Passage	Species Antarctic Petrel\n
1	Drake Passage	Species Antarctic Petrel\n
2	-60.386337,-67.075401	Species Antarctic Petrel\n

	Species	Address	Coordinates	Coord_Lat	Coord_long
0	Antarctic Petrel	Drake Passage	(-58.555462, -62.48308020569927)	-58.555462	-62.483080
1	Antarctic Petrel	Drake Passage	(-58.555462, -62.48308020569927)	-58.555462	-62.483080
2	Antarctic Petrel	None	(-60.386337, -67.075401)	-60.386337	-67.075401



Part 1: Results



Figure 1. Heatmap of the white-tailed deer occurrence from low (green) to high (red).

Part 2: Techniques & Tools + Results

Species	# of iNaturalist sightings inside IUCN Red List Area	# of iNaturalist sightings outside IUCN Red List Area	Total area of IUCN Red List polygon (sq km)	Total area of iNaturalist sightings polygon (sq km)
American bison	5	648	196,913	22,138,114
Antarctic petrel	2	21	4,809,157	5,984,679
Humpback whale	42	83	46,274,955	9,962,567
Polar bear	177	131	19,620,988	7,672,211
White-tailed deer	398	390	18,843,732	33,577,871

Part 2: Techniques & Tools + Results



Figure 2. A) Polygon of the ICUN Red List white-tailed deer distribution. B) Polygon of the iNaturalist white-tailed deer distribution.

References

¹iNaturalist. Available from <https://www.inaturalist.org>. Accessed [05/01/2021].

²IUCN 2021. The IUCN Red List of Threatened Species. Version 2021-1. <https://www.iucnredlist.org>. Accessed [05/01/2021].

³G. van Rossum, Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995

⁴Kluyver, T., Ragan-Kelley, B., Fernando Perez, Granger, B., Bussonnier, M., Frederic, J., ... Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90).

⁵The Selenium Browser Automation Project. (n.d.). Retrieved May 2, 2021, from <https://www.selenium.dev/documentation/en/>

⁶Time - Time access and conversions¶. (n.d.). Retrieved May 2, 2021, from <https://docs.python.org/3/library/time.html>

⁷Richardson, L. (2007). Beautiful soup documentation. *April*.

⁸McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

⁹Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>

¹⁰Gillies, S., & others. (2007). *Shapely: manipulation and analysis of geometric objects*. Retrieved from <https://github.com/Toblerity/Shapely>

¹³Lawhead, J. (n.d.). Retrieved May 2, 2021, from <https://pythonhosted.org/Python Shapefile Library/>

¹²Area. (n.d.). Retrieved May 2, 2021, from <https://pypi.org/project/area/>

¹³Tancell, C., Sutherland, W. J., & Phillips, R. A. (2016). Marine spatial planning for the conservation of albatrosses and large petrels breeding at South Georgia. *Biological Conservation*, 198, 165-176.