# MSDS Capstone Project Update #2

Well, even more data issues this week. As you already know, I needed to find another option to download all the data since I have a limit on how much download data I can get at my house. I appreciate Dr Perry getting me set up on the server and installing a different version of perl to run OpenGrADS. Will be putting this in an Acknowledgment section of the paper and PowerPoint.

The second issue is that I do not have enough time to download all the data from RDA. They limit to 10 concurrent downloads, so either its 10 files from the same year at once or 10 years at a time. In addition it is 200 GB per year. Unfortunately, I realized this after I attempted to do concurrent downloads using the Multithreading library in Python. The script I tried using is in GitHub under implementation and Scripts Attempted folders.

I decided, due to the download issues, to use <https://nomads.ncdc.noaa.gov/data/narr/>. This only has data through part of 2014, so I am using this as a source for 1979 to 2013. What is nice about this data set, is instead of downloading all 8 files per day as one compressed file, I can download only the 12Z and 00Z data, about 40 GB per year. I’m hoping to run one model using the 12Z data and another using the 00Z data. 12Z is generally 7 AM or 6 AM (Central Time and depending on when clocks change). 00Z is 7 PM or 6 PM Central. The idea is the morning run gives meteorologists an idea of what the day may have in store and the evening run tells us if it is going to happen or not (more detail will be in paper). In addition, weather balloons are released at 12 and 00Z, so there is more accurate upper air data than the other 3 hourly files.

I created an Excel book that based on the date I typed in, create the URL for the 12Z and 00Z files. I could have done this in Python, but since I was working in Excel for work that day, I decided to continue using Excel since I was already in it, and so I wouldn’t try combining the languages like I sometimes do. The Excel book created a text file for each year with all 730 (or 732) URLs for that year. This file was then read into Python and converted to a list to loop through to download all the data.

To get 2014-2019 data I may use <https://www.esrl.noaa.gov/psd/data/gridded/data.narr.html> as they separate the files by variables. I’ll have to open the files and see if it does what I want it to do. If not, I’ll try RDA data, but I might have to compromise and use 1979-2013 data only due to time restrictions.

I am leaning heavily to not worrying about 2014-2019, however. With the data I will have if I don’t, it’ll be 34 years of data with two files per day. In meteorology, climatology uses the past 30 years, so I’ll have 4 more over what we call normal.

I also still need to download and clean the tornado data. The Storm Prediction Center has a data page (<https://www.spc.noaa.gov/wcm/#data>) where they have CSVs of tornadoes, hail and wind reports. There are 4 different types of CSVs for tornadoes. It took some investigating, but I finally found on their page that states “Actual tornado tracks only (not including individual state segments) are provided in the "Actual\_tornadoes.csv"”. My understanding is that this file has each tornado only twice while the others may have the same tornado multiple times if it crossed states. I downloaded that file and “1950-2018\_all\_tornadoes.csv” to check. It turned out to be an easy check as the first recorded tornado crossed MO and IL. In the all\_tornadoes it was listed three times, but in the actual file it was listed only once. The way to see this is that there is an “OM” field. Per documentation, OM is the tornado number. It states that tornado segments that cross states or more than 4 counties will have the same OM number.

Since all I care about is the number of tornadoes per day, the only data field I need to worry about is the date field. On a quick scroll of the 63,646 tornadoes, it seems fine. I will be confirming this in a Pandas DataFrame when I get to this step. This dataset is from 1950-2018, so I will have to extract only 1979-2013.

No matter what, I am starting my code for my machine learning model on Sunday. I can’t spend any more time on the data in order to complete this project. I only have two weeks to run this model before I must turn in the rough draft of the paper, so I need to get going. This data collecting/cleaning is all part of the process that we’ve used in classes, I just need to remember that. I’d rather start the model first day of class, but that just wasn’t possible!