COMP1204: Data Management
Coursework One: Hurricane Monitoring

James Palmer
35277963

March 7, 2024

1 Introduction

In this coursework, there were 3 aims:

1. Create a bash script to convert convoluted kml data into comma separated values.
2. Take the cleaned data and plot it on a map to monitor a hurricane.
3. Resolve a git conflict by merging two versions of a Python script.

2 Create CSV Script

```
#!/bin/bash
#$1 is input 1, $2 is input 2
#Firstly, clearing the output file
truncate -s 0 $2
#Grabbing all lines of the input file with <tr> tag
#Replacing all tags with nothing
#Removing the 10 blank spaces at the start of the remaining lines
#Removing lines that contain '2020'
#Removing anything after a ; on each line
#Removing lines that contain :
#Removing an extra whitespace every 4 lines, starting on the second line
#Removing every space after a comma
#Combining every two lines,seperating with a whitespace
#Repeating so every 4 lines are combined
grep "<tr>" $1 | sed 's/<[^>]*>//g' | sed -r 's/^.{10}//' | sed -n '/2020/!p' | sed 's/;.*//;/^$/d' | s
#Adds header to the top of the file
echo "Timestamp,Latitude,Longitude,MinSeaLevelPressure,MaxIntensity" | cat - $2 > temp && mv temp $2
```

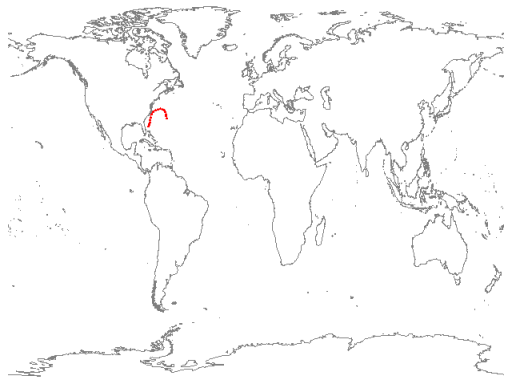
The sh script starts with the truncate command, which reduces the size of a file, with the -s option to specify the resulting size of the file. $2 chooses the file to operate on, which is input 2. Therefore, you are truncating the second input file to 0 bytes - which results in clearing it, to ensure there is no previous content in it.

Following this a series of piped commands to reach the desired text format. I achieved this using an iterative approach , by adding a new command, printing the command, and adding a new sed command. Doing this in a cycle allowed me to eventually achieve the solution

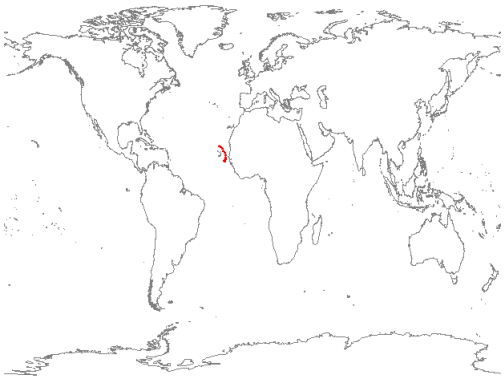
First, the grep command grabs all of the lines with the ¡tr¿ tag, as it precedes all of the key data. The $1 indicates this is done on he first input file. Then a series of sed commands(explained in the comments) remove all unnecessary text, and arrange the data into csv format.

Finally which is added to the file, I add in the row titles.

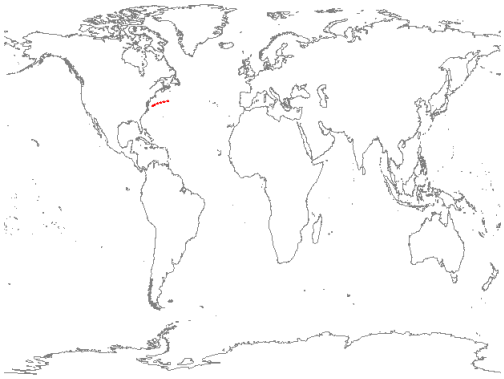
3 Storm Plots



KML:al012020.kml



KML:al102020.kml



KML:al122020.kml

4 Git usage

Commands used:
1.git branch - to view all current branches
2.git merge python-addon - This returned an error, as there is a conflict to resolve

3.vim python-plot-script.py - To open and fix the file
4.git add . - Adds new files
5.git commit
6.git merge python-addon
7.git push

```
import matplotlib.pyplot as plt
import os
import glob
import math
user_key = 16332

def plot_all_csv_pressure():
        storm = pd.read_csv(f)
        storm['Pressure'].plot()
        plt.show()

def plot_all_csv_intensity():
    path = os.getcwd()
    csv_files = glob.glob(os.path.join(path, '*.csv'))

    for f in csv_files:
        storm = pd.read_csv(f)
        storm['Intensity'].plot()
        plt.show()
```