COMP1204: Data Management
Coursework One: Hurricane Monitoring

Patrick Jfremov-Kustov
33461104

March 17, 2023

1 Introduction

I've been asked to gather storm information and produce storm-tracking maps. In order to accurately track the storm, this task requires extracting and analysing the storm data. This is done using a bash script created to illustrate storm plots, as well as any git conflicts that occurred in the process.

2 Create CSV Script

The script takes two arguments, the input.kml file and the output.csv file. If these two arguments aren't provided, it will not allow you to run the script. In case there's an existing file with the same name, it will overwrite that existing file. For every line in the file, it will search whether the line contains the tags with the necessary information inside. The tags in the IF sequence are aligned with the order of tags in the input file. If it does contain any of the tags, it will print the line and pipe it into sed where it will remove the tags and store that in a separate variable. Afterwards, it will append to the file in the format specified. Once it reaches the end of the file, it will close the .kml file and finish the script.

```
#!/bin/bash

# Takes the input KML file as the first argument
inputFile="$1"

# Create a new CSV file as the second argument
outputFile="$2"

# Will only continue if both arguments are provided, showing proper usage.
if [ $# -ne 2 ]; then
        echo "Incorrect usage of the script. Usage: $0 inputFile outputFile"
        exit 1
fi

# Will rewrite an existing output file with the same name
rm -f "$outputFile"

echo "Converting $inputFile -> $outputFile..."

echo "Timestamp,Latitude,Longitude,MinSeaLevelPressure,MaxIntensity" >> "$outputFile"
```

```
# Opens the .kml file
exec 3< "$inputFile"

# Read each line of the file, not treating \ as escape characters
# Finds the information inside each tag, and removes the tags
# Then for each line, it will append to the file until the end of file is reached
# IF statements sequential to the layout of the .kml file (truncation / formatting errors otherwise)
while read -r line <&3; do
        if [[ $line == *"<lat>"* ]]; then
                lat=$(echo $line | sed -n ’s/.*<lat>\(.*\)<\/lat>.*/\1/p’)
        elif [[ $line == *"<lon>"* ]]; then
                lon=$(echo $line | sed -n ’s/.*<lon>\(.*\)<\/lon>.*/\1/p’)
        elif [[ $line == *"<intensity>"* ]]; then
                intensity=$(echo $line | sed -n ’s/.*<intensity>\(.*\)<\/intensity>.*/\1/p’)
        elif [[ $line == *"<minSeaLevelPres>"* ]]; then
                minSeaLevelPres=$(echo $line | sed -n ’s/.*<minSeaLevelPres>\(.*\)<\/minSeaLevelPres>.*/
        elif [[ $line == *"<dtg>"* ]]; then
                dtg=$(echo $line | sed -n ’s/.*<dtg>\(.*\)<\/dtg>.*/\1/p’)
        echo "$dtg,$lat N,$lon W,$minSeaLevelPres mb,$intensity knots" >> "$outputFile"
        fi
done

# Closes the .kml file
exec 3<&-

echo ’Done!’
```

3 Storm Plots

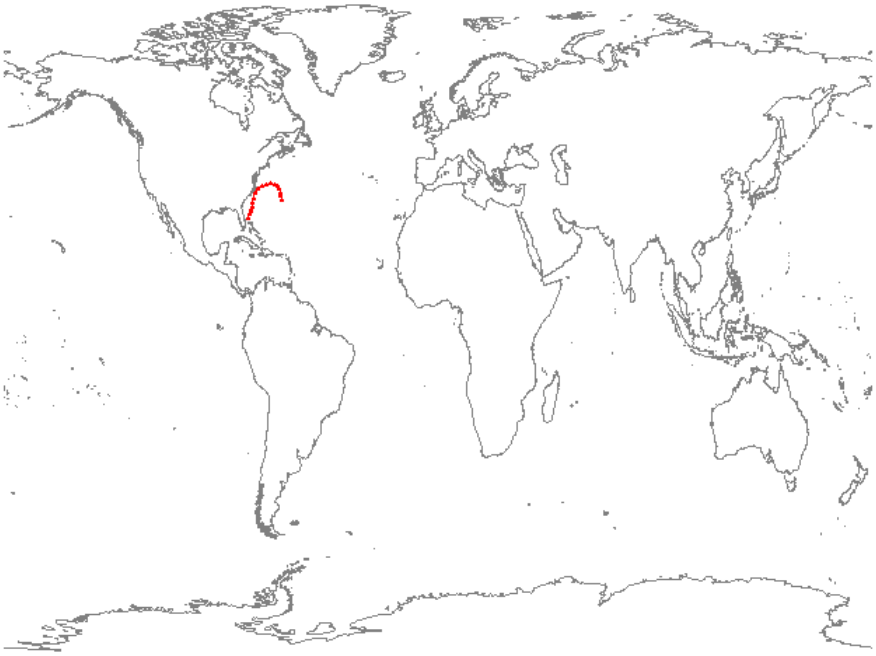

Figure 1: Storm Plot from al012020.kml

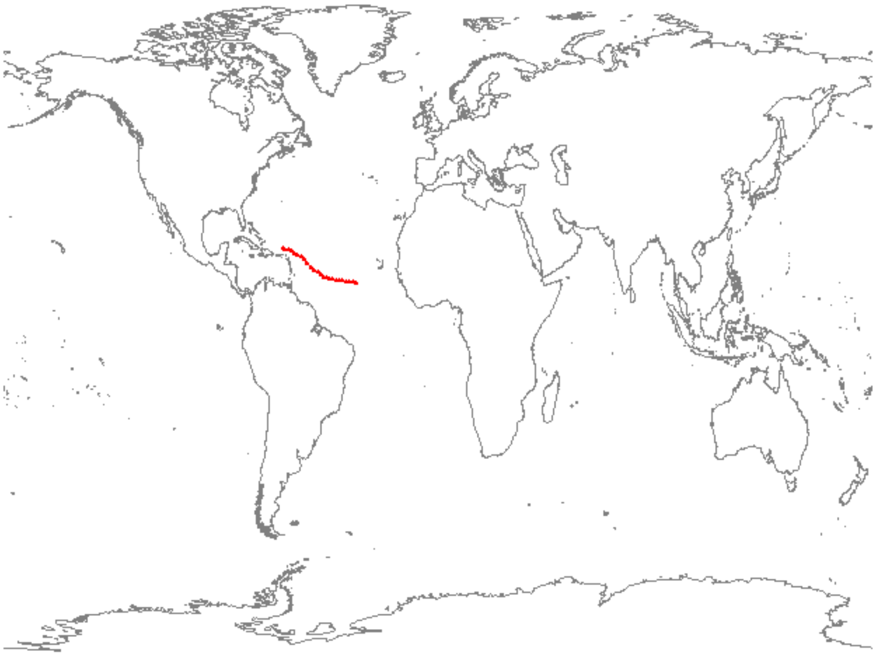

Figure 2: Storm Plot from al112020.kml

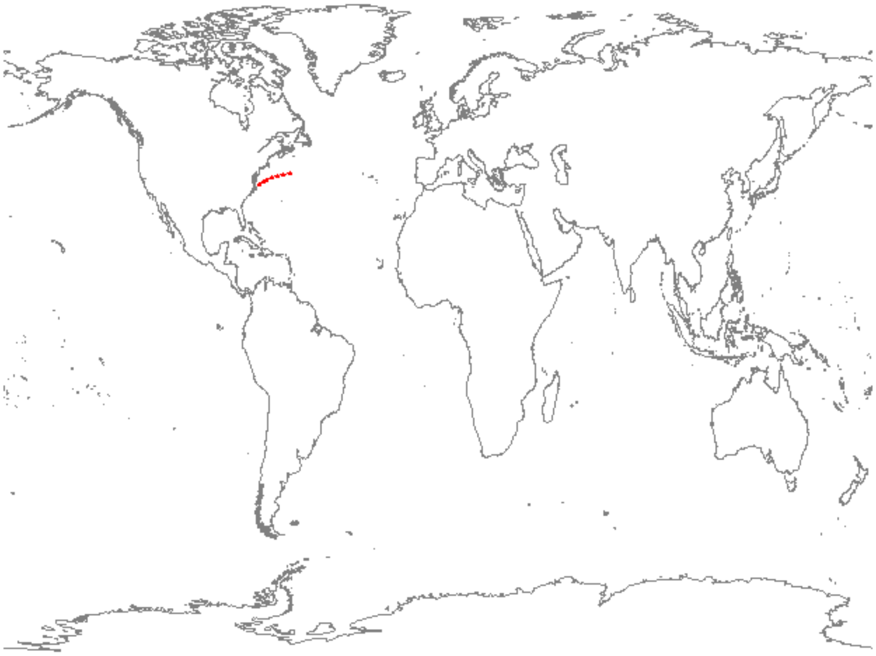

Figure 3: Storm Plot from al122020.kml

4 Git usage

A conflict was generated between the main branch and python-addon branch for the "python-plot-script.py" file. Upon inspection, a for loop was duplicated where one was mispelt. One of the modules "math" was missing, and there appears to be two user keys. These changes was amended and committed with "git commit -am "Resolved conflicts in python-plot-script.py" and merged using "git merge python-addon". Lastly, it was pushed to the remote repository using "git push".

```
import pandas as pd
import matplotlib.pyplot as plt
import os
import glob
import math
user_key = 32078
# user_key = 4315

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

    for f in csv_files:
        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()
```