

# COMP1204: Data Management

## Coursework One: Hurricane Monitoring

Damien Ta  
34830294

February 27, 2024

### 1 Introduction

As data scientists for the National Oceanographic and Atmospheric Administration Centre, I was assigned to the tropical cyclone tracking team. As a data scientist I have been tasked with extracting storm data from the tropical cyclone reports and producing maps of where the cyclones have taken place.

### 2 Create CSV Script

Here's my script:

```
create_csv.sh
```

```
#!/bin/bash
csv_input_path=$1
tmp_csv_output=$2

echo "Timestamp,Latitude,Longitude,MinSeaLevelPressure,MaxIntensity" > $tmp_csv_output
dtg="$(grep -R "<dtg>" $csv_input_path | sed 's/.*<dtg>//g' | sed 's/[</dtg>//g')"
```

```
grep -R "<lat>" $csv_input_path | sed 's/.*<lat>//g' | sed 's/[</lat>//g' > lat.csv
grep -R "<lon>" $csv_input_path | sed 's/.*<lon>//g' | sed 's/[</lon>//g' > lon.csv
grep -R "<minSeaLevelPres>" $csv_input_path | sed 's/.*<minSeaLevelPres>//g' | /
sed 's/[</minSeaLevelPres>//g' > minSeaLevelPres.csv
grep -R "<intensity>" $csv_input_path | sed 's/.*<intensity>//g' | /
sed 's/[</intensity>//g' > intensity.csv

lat="$(sed "s/$/ N/" lat.csv)"
lon="$(sed "s/$/ W/" lon.csv)"
minSeaLevelPres="$(sed "s/$/ mb/" minSeaLevelPres.csv)"
maxIntensity="$(sed "s/$/ knots/" intensity.csv)"

paste -d',' <(echo "$dtg") <(echo "$lat") <(echo "$lon") <(echo "$minSeaLevelPres") \
<(echo "$maxIntensity") >> $tmp_csv_output

rm lat.csv
rm lon.csv
rm minSeaLevelPres.csv
rm intensity.csv
#I used / to continue command on next line
```

I will now explain my code in small boxes each in its own independent sections:

### First part of the code

```
1 #!/bin/bash
2 csv_input_path=$1
3 tmp_csv_output=$2
```

```
#!/bin/bash
```

tells the terminal that when you run the script it should use bash to execute it

```
csv_input_path=$1
```

```
tmp_csv_output=$2
```

These two lines of code allow inputs to be entered into the script

### Second part of the code

```
1 echo "Timestamp,Latitude,Longitude,MinSeaLevelPressure,MaxIntensity" > $tmp_csv_output
2 dtg="$(grep -R "<dtg>" $csv_input_path | sed 's/.*<dtg>//g' | sed 's/[</dtg>//g')"
```

```
3 grep -R "<lat>" $csv_input_path | sed 's/.*<lat>//g' | sed 's/[</lat>//g' > lat.csv
4 grep -R "<lon>" $csv_input_path | sed 's/.*<lon>//g' | sed 's/[</lon>//g' > lon.csv
5 grep -R "<minSeaLevelPres>" $csv_input_path | sed 's/.*<minSeaLevelPres>//g' | \
6 sed 's/[</minSeaLevelPres>//g' > minSeaLevelPres.csv
7 grep -R "<intensity>" $csv_input_path | sed 's/.*<intensity>//g' | \
8 sed 's/[</intensity>//g' > intensity.csv
```

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

This command prints the tag headers

```
dtg="$(grep -R "<dtg>" $csv_input_path | sed 's/.*<dtg>//g' | sed 's/[</dtg>//g')"
```

The code creates variable dtg, we grep the kml file inputted at \$1 and search recursively for keyword <dtg>. The sed 's/.\*<dtg>//g' command would replace any character and its occurrences with an empty space including anything before the pattern.

the sed 's/[</dtg>//g' command would replace any characters between the [], so < / d t g > to an empty character and the g stands for global.

```
grep -R "<lat>" $csv_input_path | sed 's/.*<lat>//g' | sed 's/[</lat>//g' > lat.csv
```

grep recursively keyword <lat> from the kml file inputted

The sed 's/.\*<dtg>//g' command would replace any character and its occurrences with an empty space including anything before the pattern.

The sed 's/[</lat>//g' command would replace any characters between the [], so < / l a t >

The outcome is then appended to a file called lat.csv for later use. This same code is then run on the rest of the tag headers Longitude, MinSeaLevelPressure and MaxIntensity.

### Third part of the code

```
1 lat="$(sed "s/$/ N/" lat.csv)"
2 lon="$(sed "s/$/ W/" lon.csv)"
3 minSeaLevelPres="$(sed "s/$/ mb/" minSeaLevelPres.csv)"
4 maxIntensity="$(sed "s/$/ knots/" intensity.csv)"
```

```
lat="$(sed" s// N/" lat.csv)"
```

New variable lat is created which uses sed and adds on each line (\$) " N" from file lat.csv  
This is then run on each line of code for longitude, minSeaLevelPres and maxIntensity for their respective unit of measure.

### Fourth part of the code

```
1 paste -d',' <(echo "$dtg") <(echo "$lat") <(echo "$lon") <(echo "$minSeaLevelPres") /
2 <(echo "$maxIntensity") >> $tmp_csv_output
```

```
paste -d',' <(echo "$dtg") <(echo "$lat") <(echo "$lon") <(echo "$minSeaLevelPres") / <(echo
"$maxIntensity") $tmp_csv_output
```

### 3 Storm Plots

#### 3.1 Storm Plot 1

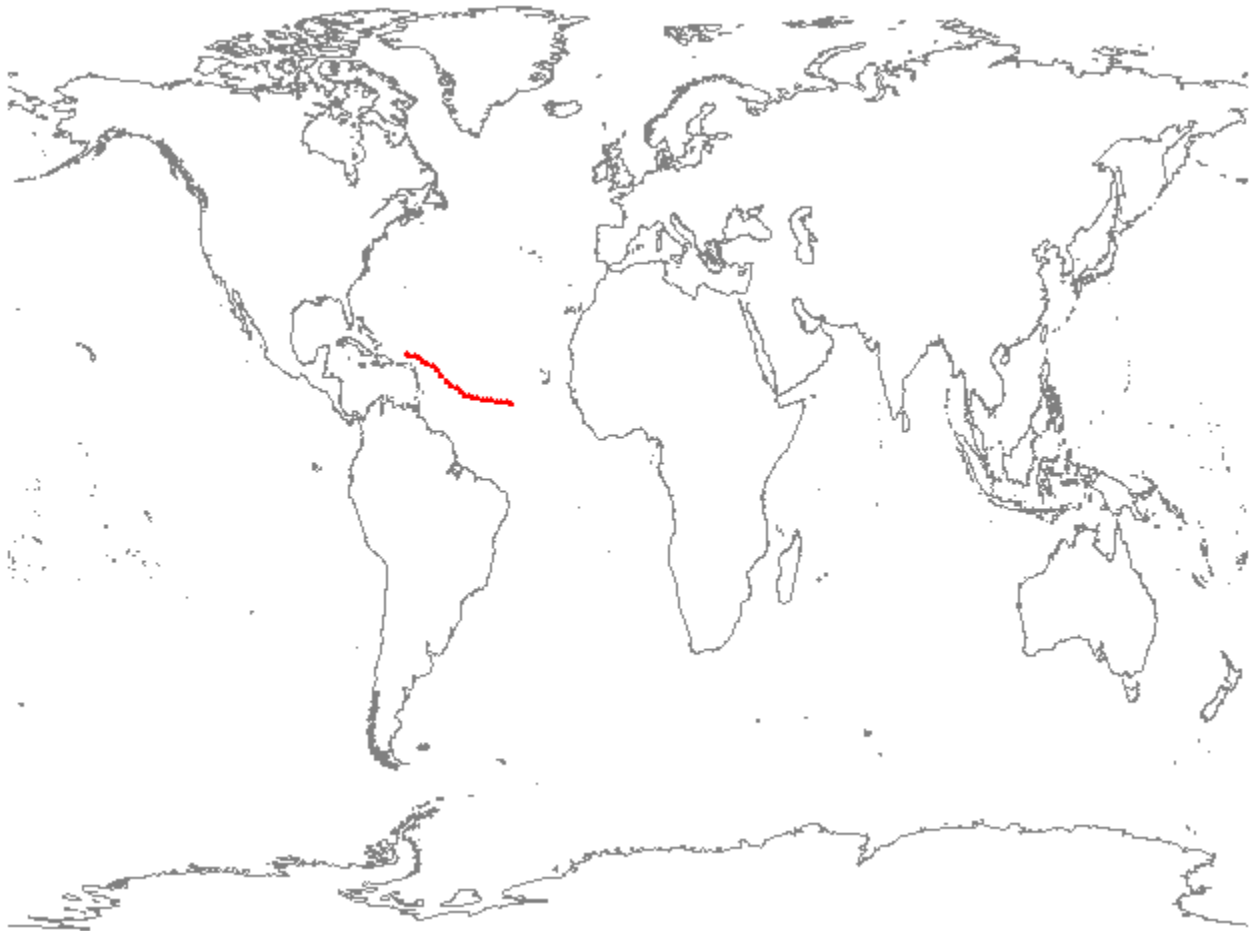


Figure 1: al112020.kml

### 3.2 Storm Plot 2

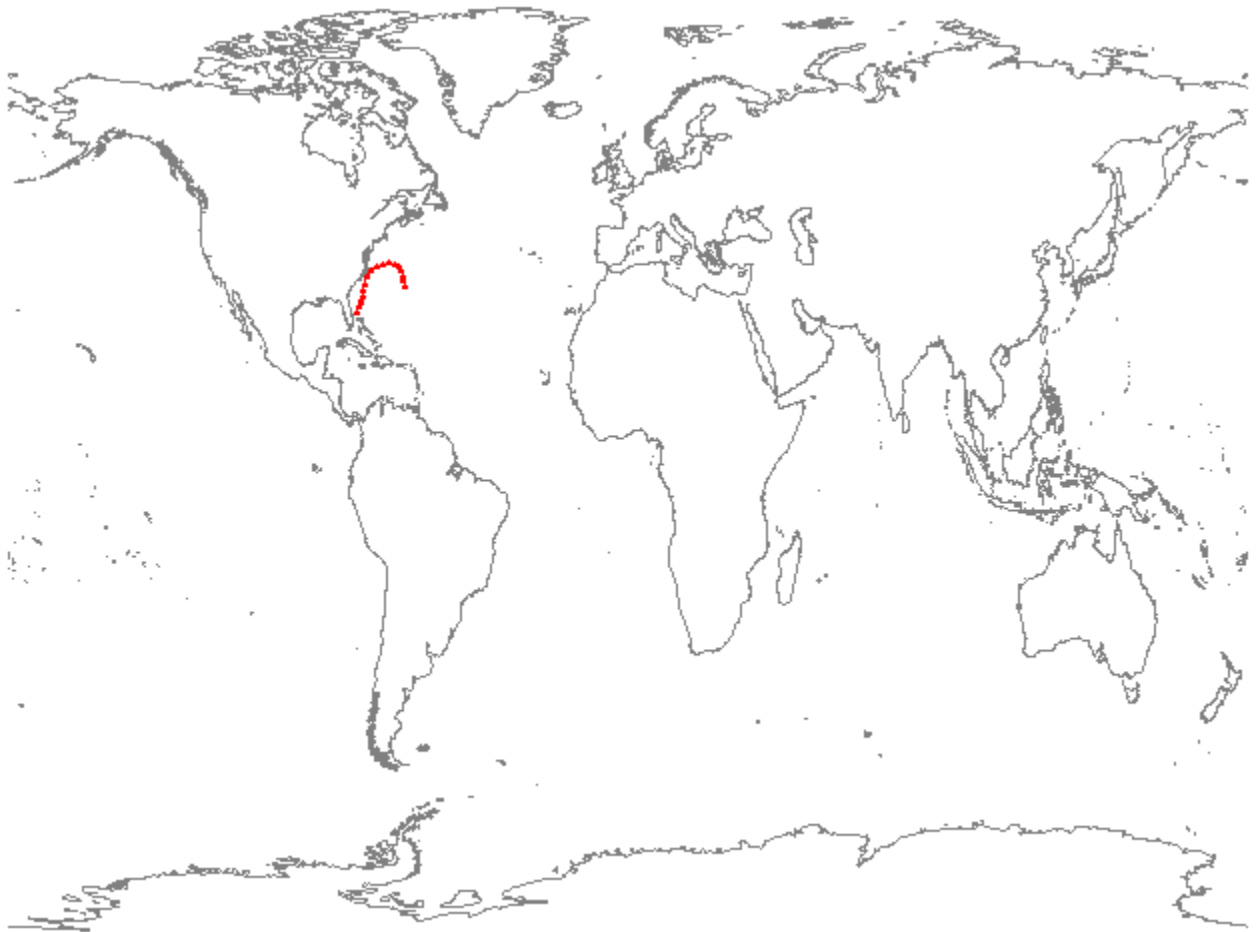


Figure 2: al012020.kml

### 3.3 Storm Plot 3

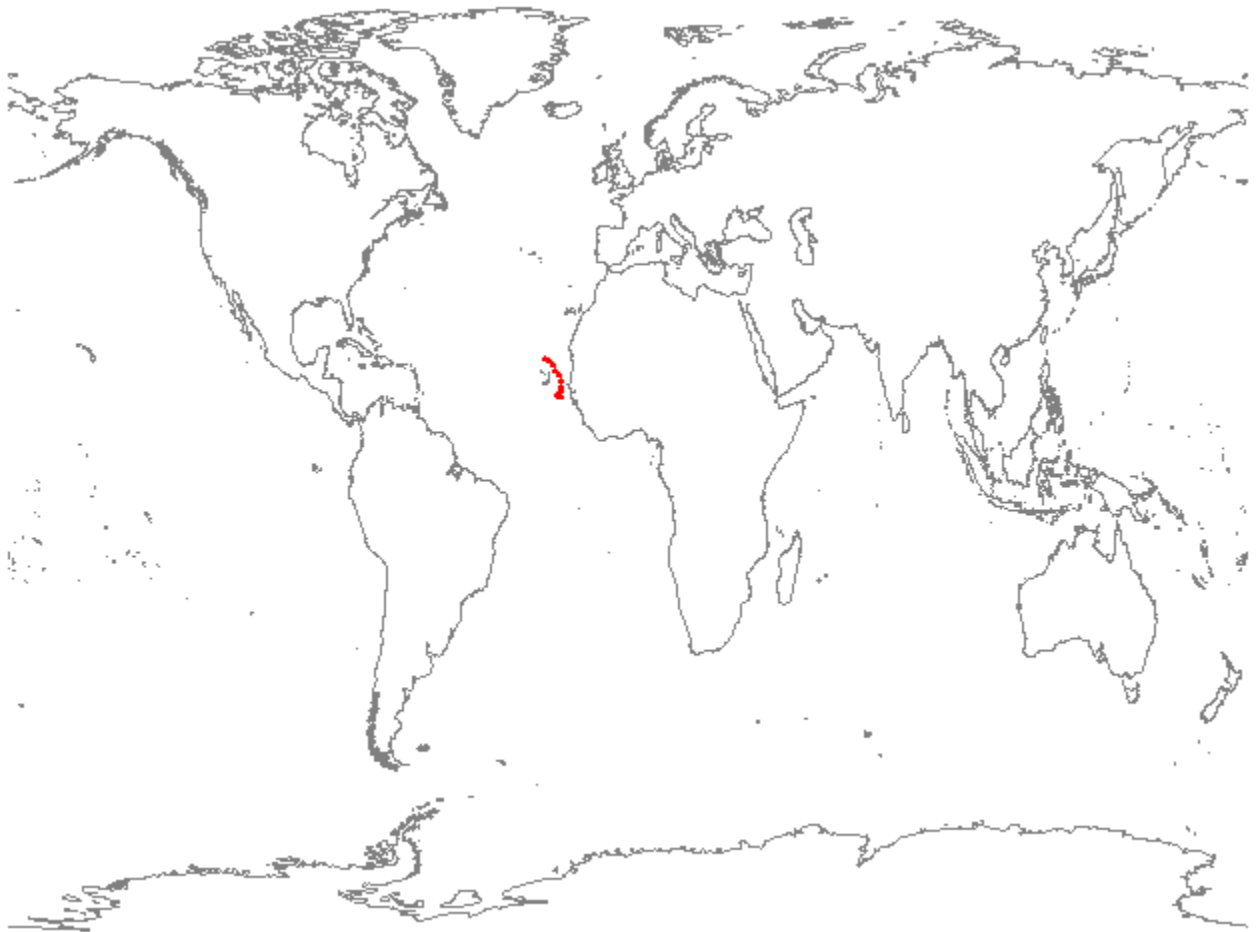


Figure 3: al102020.kml

## 4 Git Usage

### Python Code for Plotting CSV Data

```
import pandas as pd
import matplotlib.pyplot as plt
import os
import glob
import math
user_key = 1773

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()
```