Capstone Project - Group 9 - Climate Change

John Moen
Carmlina Sandoval
Amer Lam
Sanil Veeravu

The Heat Map - Analyzing Climate Change
Executive Summary

Introduction

Climate Change is one of the most critical issues of our time. From change in weather patterns that impact food production, to rising sea levels that increase the risk of catastrophic flooding, spreading wildfires the impacts of climate change are global in scope and unprecedented in scale.

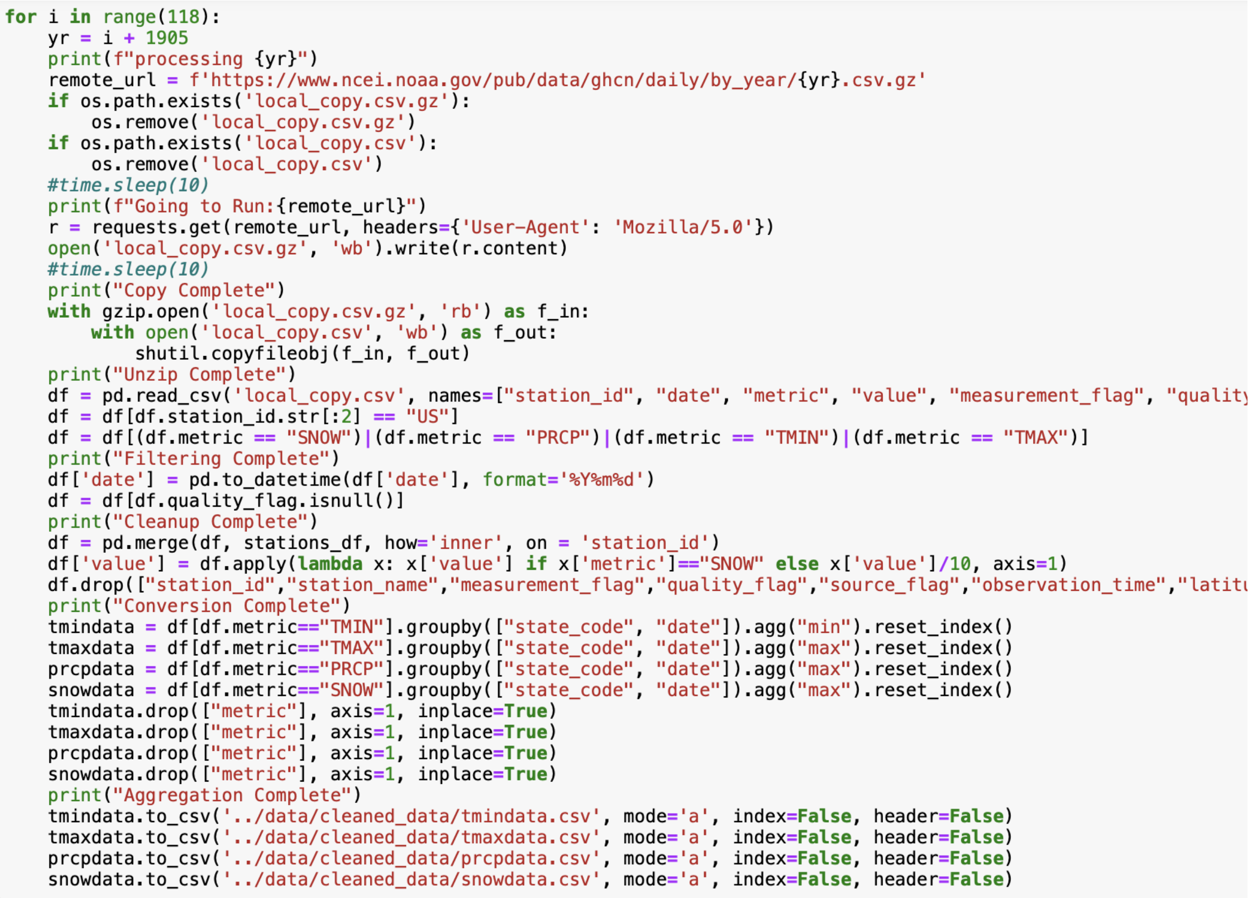
Through this research using historical climate change data from NOAA we did research to predict the climate change expected by US states/countries for future years to show the possible impact to our environment.

Data Collection

National Centers of Environmental Information maintains data from different stations all over the world with different metrics like temperature max/min, snow depth, precipitation and many more. They also provide data with details of stations. All this data was available in this URL - https://www.ncei.noaa.gov/pub/data/ghcn/daily/.

Data was available in different formats and the station data involved data for each month in a row with a column for each day of the month. Initial work was done to convert this data into a row for each day. But with additional research we identified a similar dataset was available in the yearly folder.

With the data being so huge, we build a loop to download one year at a time using python requests, perform the necessary aggregations/lookup and store only the required data into the required metric files

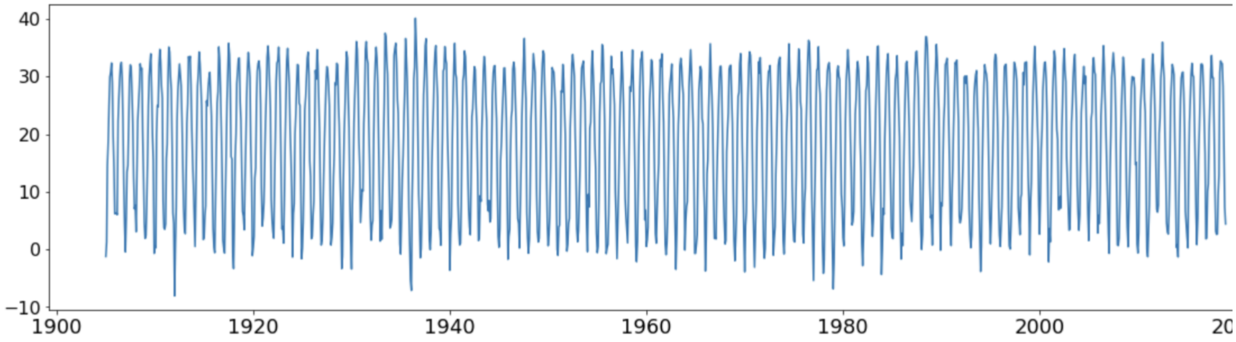


```
for i in range(118):
    yr = i + 1905
    print(f"processing {yr}")
    remote_url = f'https://www.ncei.noaa.gov/pub/data/ghcn/daily/by_year/{yr}.csv.gz'
    if os.path.exists('local_copy.csv.gz'):
        os.remove('local_copy.csv.gz')
    if os.path.exists('local_copy.csv'):
        os.remove('local_copy.csv')
    #time.sleep(10)
    print(f"Going to Run:{remote_url}")
    r = requests.get(remote_url, headers={'User-Agent': 'Mozilla/5.0'})
    open('local_copy.csv.gz', 'wb').write(r.content)
    #time.sleep(10)
    print("Copy Complete")
    with gzip.open('local_copy.csv.gz', 'rb') as f_in:
        with open('local_copy.csv', 'wb') as f_out:
            shutil.copyfileobj(f_in, f_out)
    print("Unzip Complete")
    df = pd.read_csv('local_copy.csv', names=["station_id", "date", "metric", "value", "measurement_flag", "quality
    df = df[df.station_id.str[:2] == "US"]
    df = df[(df.metric == "SNOW")|(df.metric == "PRCP")|(df.metric == "TMIN")|(df.metric == "TMAX")]
    print("Filtering Complete")
    df['date'] = pd.to_datetime(df['date'], format='%Y%m%d')
    df = df[df.quality_flag.isnull()]
    print("Cleanup Complete")
    df = pd.merge(df, stations_df, how='inner', on = 'station_id')
    df['value'] = df.apply(lambda x: x['value'] if x['metric']=="SNOW" else x['value']/10, axis=1)
    df.drop(["station_id","station_name","measurement_flag","quality_flag","source_flag","observation_time","latitu
    print("Conversion Complete")
    tmindata = df[df.metric=="TMIN"].groupby(["state_code", "date"]).agg("min").reset_index()
    tmaxdata = df[df.metric=="TMAX"].groupby(["state_code", "date"]).agg("max").reset_index()
    prcpdata = df[df.metric=="PRCP"].groupby(["state_code", "date"]).agg("max").reset_index()
    snowdata = df[df.metric=="SNOW"].groupby(["state_code", "date"]).agg("max").reset_index()
    tmindata.drop(["metric"], axis=1, inplace=True)
    tmaxdata.drop(["metric"], axis=1, inplace=True)
    prcpdata.drop(["metric"], axis=1, inplace=True)
    snowdata.drop(["metric"], axis=1, inplace=True)
    print("Aggregation Complete")
    tmindata.to_csv('../data/cleaned_data/tmindata.csv', mode='a', index=False, header=False)
    tmaxdata.to_csv('../data/cleaned_data/tmaxdata.csv', mode='a', index=False, header=False)
    prcpdata.to_csv('../data/cleaned_data/prcpdata.csv', mode='a', index=False, header=False)
    snowdata.to_csv('../data/cleaned_data/snowdata.csv', mode='a', index=False, header=False)
```

We were able to condense the data from multiple GBs to 34 MB files.

One key challenge that came during this process was the quality of the data. The initial graph was all around the place with temperature upto 5000 C. After more data exploration identified a column called quality flag which provides certification of the data and once this filter was applied we got a smooth dataset for metrics.

```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```



Database

Visualizations

Web Application

Machine Learning