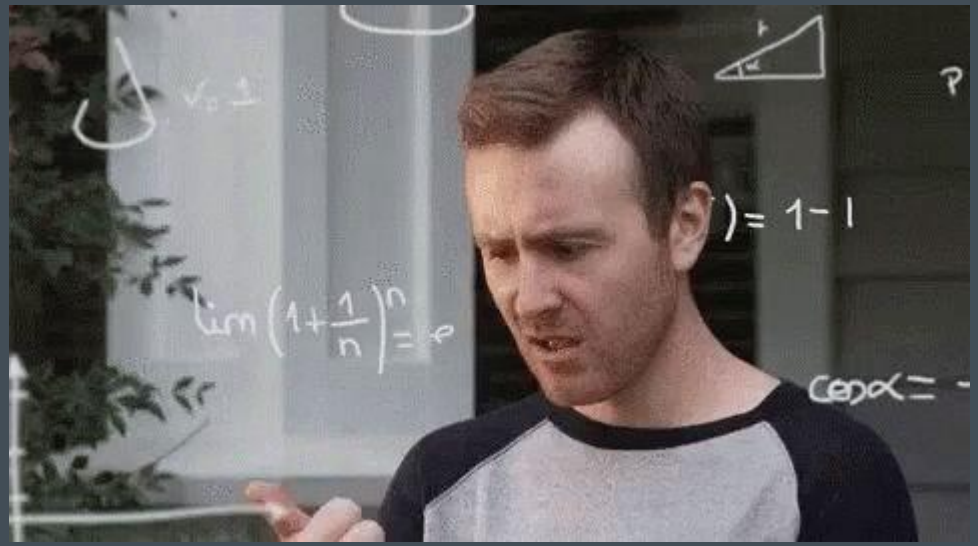
Weather web visualization

Represent by : John Antony
Liam Baker
Nishant Patel

Question!

Are there any strategies to predict weather?

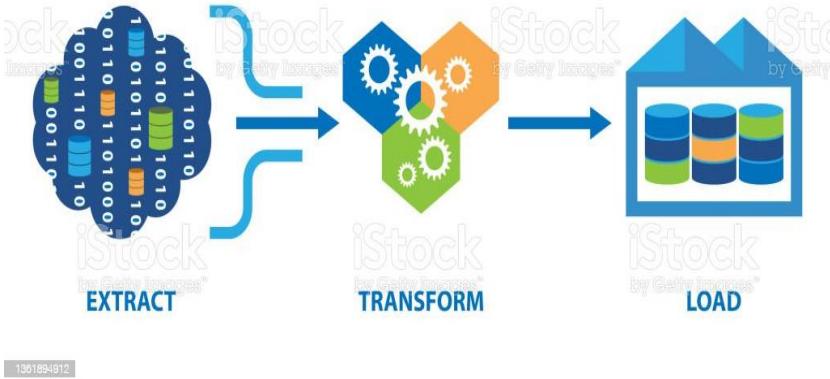


Objectives

- Data cleaning and data wrangling
- Data store with SQL database
- Obtaining the mean temperature variations of weather stations
- Python Flask API route and JavaScript
- Visualisation of final outputs via Leaflet and Plotly
- Which states are most and least accurate with their weather?

Data cleaning and data wrangling:

- Remove duplicates of data
- Remove unnecessary data
- Change column name with new DataFrame
- Store data in SQL with weatherobs database
- Data cleaning is performed in Jupiter notebook.



Coding for getting min and max temperature:

In [3]:
```
#Fetch station data
stationData = pd.read_csv("./Resources/stationData.csv")
stationData.head()
```

Out[3]:

	WMO_NUM	station_number	station_name	LATITUDE	LONGITUDE	STN_HT	AVIATION_ID	REGION	GridPt Lat	GridPt Lon	MSAS elevation	Distance from GridPt	Roughness	Dist
0	94648	23000	ADELAIDE (WEST TERRACE / NGAYIRDAPIRA)	-34.9257	138.5832	29.32	ADWT	SA	NaN	NaN	NaN	NaN	NaN	
1	94489	38076	WINDORAH AP	-25.4117	142.6647	132.16	YWDH	QLD	NaN	NaN	NaN	NaN	NaN	
2	94795	9281	MILLENDON (SWAN VALLEY)	-31.8108	116.0225	16.00	SWVA	WA	NaN	NaN	NaN	NaN	NaN	
3	99218	32194	COWLEY BEACH	-17.6904	146.1126	17.00	CBTA	QLD	NaN	NaN	NaN	NaN	NaN	
4	94794	51164	GIRILAMBONE (OKEH)	-31.0822	146.9294	178.00	NDR2	NSW	NaN	NaN	NaN	NaN	NaN	

In [4]:
```
#Make sure to match schema
stationData_sql = pd.DataFrame()
stationData_sql["station_number"] = stationData["station_number"]
stationData_sql["station_name"] = stationData["station_name"]
stationData_sql["lat"] = stationData["LATITUDE"]
stationData_sql["lon"] = stationData["LONGITUDE"]
stationData_sql["height"] = stationData["STN_HT"]
stationData_sql["region"] = stationData["REGION"]
stationData_sql.to_sql("station_data", conn,if_exists='append',index=False)
```

Out[4]: 575

In [5]:
```
#Get file list for fcst and obs tables
fcst_files = os.listdir("./Resources/fcst")
fcst_data = pd.DataFrame()
for file in fcst_files:
    date = datetime.datetime.strptime(file[-12:-4],"%Y%m%d")
    tempdf = pd.read_csv("./Resources/fcst/" + file)
    tempdf = tempdf.loc[(tempdf["parameter"] == "MaxT") | (tempdf["parameter"] == "MinT")].reset_index()
    first_seconds = tempdf["valid_start"][0]
    tempdf_sql = pd.DataFrame()
    tempdf_sql["station_number"] = tempdf["station_number"]
    tempdf_sql["parameter"] = tempdf["parameter"]
    tempdf_sql["area_code"] = tempdf["area_code"]
    tempdf_sql["valid_start"]  = (tempdf["valid_start"] - first_seconds)/(60*60)
    tempdf_sql["valid_end"]  = (tempdf["valid_end"] - first_seconds)/(60*60)
    tempdf_sql["temperature"]  = (tempdf["value"])
    tempdf_sql["date"] = date
    tempdf_sql.to_sql("fcst",conn,if_exists='append',index=False)
```

In [5]:
```
#Get file list for fcst and obs tables
fcst_files = os.listdir("./Resources/fcst")
fcst_data = pd.DataFrame()
for file in fcst_files:
    date = datetime.datetime.strptime(file[-12:-4],"%Y%m%d")
    tempdf = pd.read_csv("./Resources/fcst/" + file)
    tempdf = tempdf.loc[(tempdf["parameter"] == "MaxT") | (tempdf["parameter"] == "MinT")].reset_index()
    first_seconds = tempdf["valid_start"][0]
    tempdf_sql = pd.DataFrame()
    tempdf_sql["station_number"] = tempdf["station_number"]
    tempdf_sql["parameter"] = tempdf["parameter"]
    tempdf_sql["area_code"] = tempdf["area_code"]
    tempdf_sql["valid_start"]  = (tempdf["valid_start"] - first_seconds)/(60*60)
    tempdf_sql["valid_end"]  = (tempdf["valid_end"] - first_seconds)/(60*60)
    tempdf_sql["temperature"]  = (tempdf["value"])
    tempdf_sql["date"] = date
    tempdf_sql.to_sql("fcst",conn,if_exists='append',index=False)
```

In [6]:
```
#Get file list for fcst and obs tables
obs_files = os.listdir("./Resources/obs")
fcst_data = pd.DataFrame()
for file in obs_files:
    date = datetime.datetime.strptime(file[-12:-4],"%Y%m%d")
    tempdf = pd.read_csv("./Resources/obs/" + file)
    tempdf = tempdf.loc[tempdf["parameter"] == "AIR_TEMP"]
    first_seconds = tempdf["valid_start"][0]
    tempdf_sql = pd.DataFrame()
    tempdf_sql["station_number"] = tempdf["station_number"]
    tempdf_sql["area_code"] = tempdf["area_code"]
    tempdf_sql["valid_start"]  = (tempdf["valid_start"] - first_seconds)/(60*60)
    tempdf_sql["valid_end"]  = (tempdf["valid_end"] - first_seconds)/(60*60)
    tempdf_sql["temperature"]  = (tempdf["value"])
    tempdf_sql["date"] = date
    tempdf_sql.to_sql("obs",conn,if_exists='append',index=False)
```

In [8]:
```
#Get max/min temps
daily_obs = obs.groupby(["date", "station_number"])
daily_obs.head()
extremes = pd.DataFrame()
extremes["max"] = daily_obs["temperature"].max()
extremes["min"] = daily_obs["temperature"].min()
extremes = extremes.reset_index()
extremes.head()
```

Out[8]:

	date	station_number	max	min
0	2016-05-01	1006	38.1	24.8
1	2016-05-01	1007	33.5	28.2
2	2016-05-01	1019	38.0	20.4
3	2016-05-01	1020	36.2	24.7
4	2016-05-01	2012	36.6	21.8

In [9]:
```
max_fcst = fcst.loc[fcst["parameter"] == "MaxT"]
min_fcst = fcst.loc[fcst["parameter"] == "MinT"]
max_fcst = max_fcst.groupby(["date", "station_number"])["temperature"].mean()
min_fcst = min_fcst.groupby(["date", "station_number"])["temperature"].mean()
combined_fcst = pd.DataFrame()
combined_fcst["max"] = max_fcst
combined_fcst["min"] = min_fcst
combined_fcst.reset_index()
```

Out[9]:

	date	station_number	max	min
0	2016-05-01	1006	35.955556	23.511111
1	2016-05-01	1019	34.533333	20.855556
2	2016-05-01	1020	34.400000	20.300000
3	2016-05-01	2012	32.722222	20.544444
4	2016-05-01	2056	34.500000	21.433333
...	...	...	...	...
5486	2017-04-01	97085	11.760000	4.488889
5487	2017-04-01	98017	20.910000	13.722222

API Routes:

- "/"
- /api/v1.0/stationdata
- /api/v1.0/fcst
- /api/v1.0/obs
- /api/v1.0/var

```python
@app.route("/api/v1.0/stationdata")
def stationdata():
    #Return JSON of station data table
    return jsonify(pd.read_sql("SELECT * FROM station_data", con=conn).to_json())

@app.route("/api/v1.0/fcst")
def fcst():
    return pd.read_sql("SELECT * FROM fcst", con=conn).to_json()
    #Return fcst data tables

@app.route("/api/v1.0/obs")
def obs():
    #Return obs data tables
    return pd.read_sql("SELECT * FROM obs", con=conn).to_json()

@app.route("/api/v1.0/var")
def var():
    #Return obs data tables
    return pd.read_sql("SELECT * FROM variation", con=conn).to_json()



if __name__ == "__main__":
    app.run(debug=True)
```

Leaflet and Plotly:

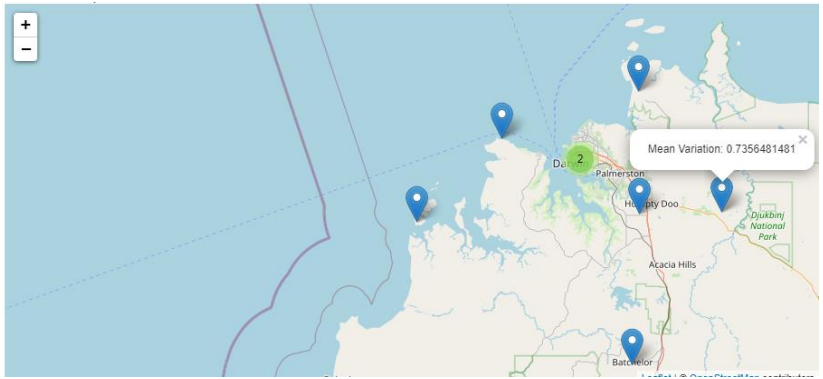
- We used plotly to display charts of weather station data of all states of Australia
- Use drop-down menu
- Used leaflet as JavaScript mapping library
- Create heat maps

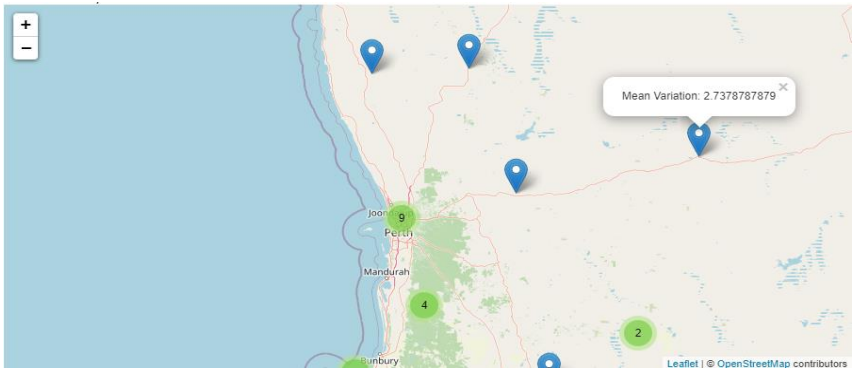
```
let url = "/api/v1.0/var";
let init = true;
let uncertainties = [];
let heatArray = [];
let positionsArray = [];
let markers = L.markerClusterGroup();
let jsonResult;

d3.json(url).then(function(response) {
  jsonResult = response;
  //Initialise Dropdown
  let dropdown = d3.select("#selDataset");
  if(init) {
      let dropdownText = '<option value="All">All</option>';
      let distinctStates = [...new Set(Object.values(response.region))];
      for(let j = 0; j < distinctStates.length; j++){
          dropdownText +=  '<option value="' + distinctStates[j] + '">' + distinctStates[j] + '
      }
      console.log(dropdownText);
      dropdown.html(dropdownText);
  }
```

```
console.log(response);
console.log(response.lat.length);
for(let i = 0; i < Object.keys(response.lat).length; i++) {
  uncertainties.push(response.station_difference[i]);
  heatArray.push([response.lat[i],response.lon[i],response.station_difference[i]]);
  positionsArray.push([response.lat[i],response.lon[i],1]);

  markers.addLayer(L.marker([response.lat[i], response.lon[i]])
    .bindPopup("Mean Variation: " + response.station_difference[i]));

}



console.log(heatArray);
console.log(positionsArray);
maxDiff = Math.max.apply(null,uncertainties);
```

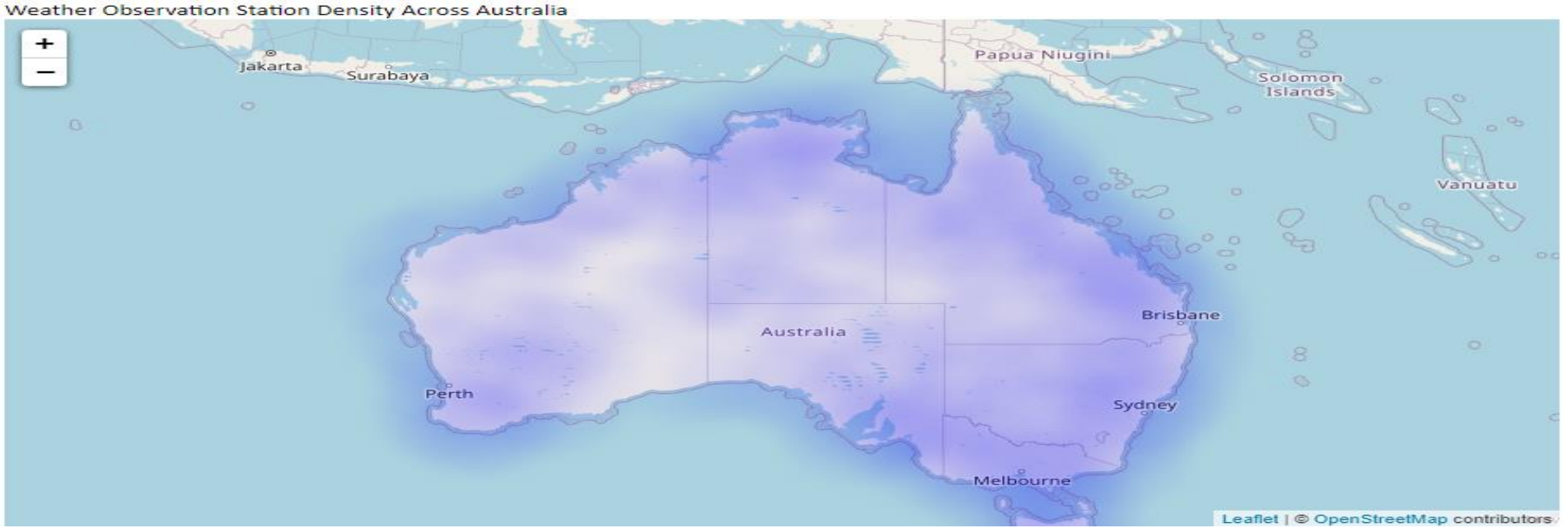
Conditions:

- Mean temperature of weather station > 2.0 => weather accuracy is not good
- Mean temperature of weather station < 2.0 => weather accuracy is good

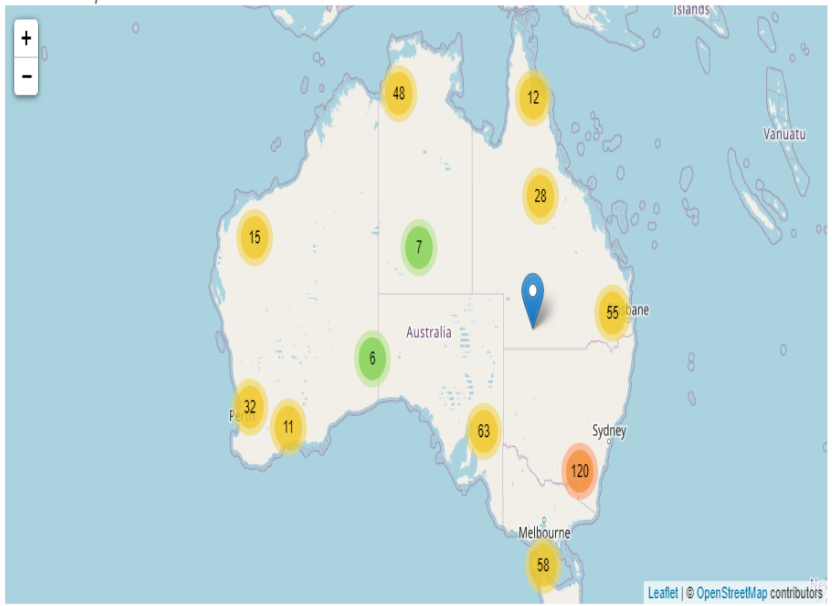




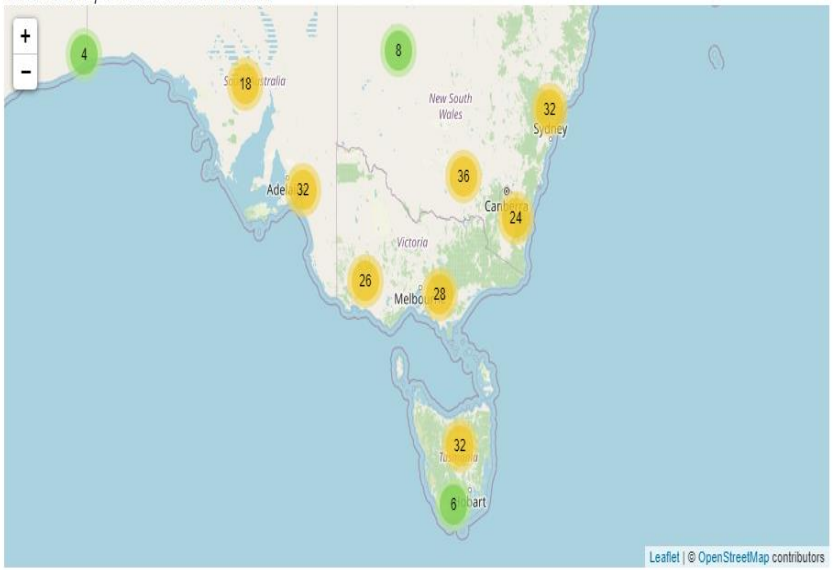
Graphs and maps



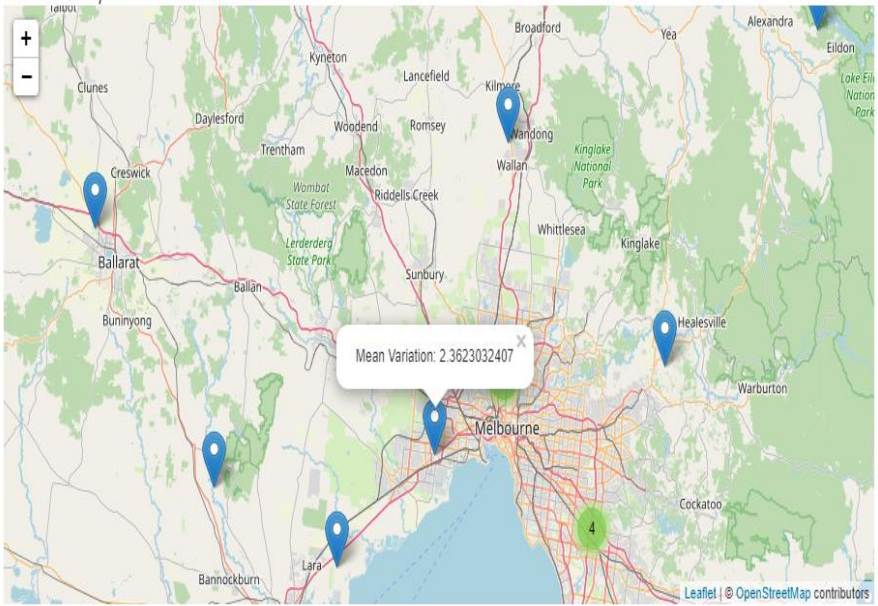
Mean Inaccuracy Of Individual Weather Stations



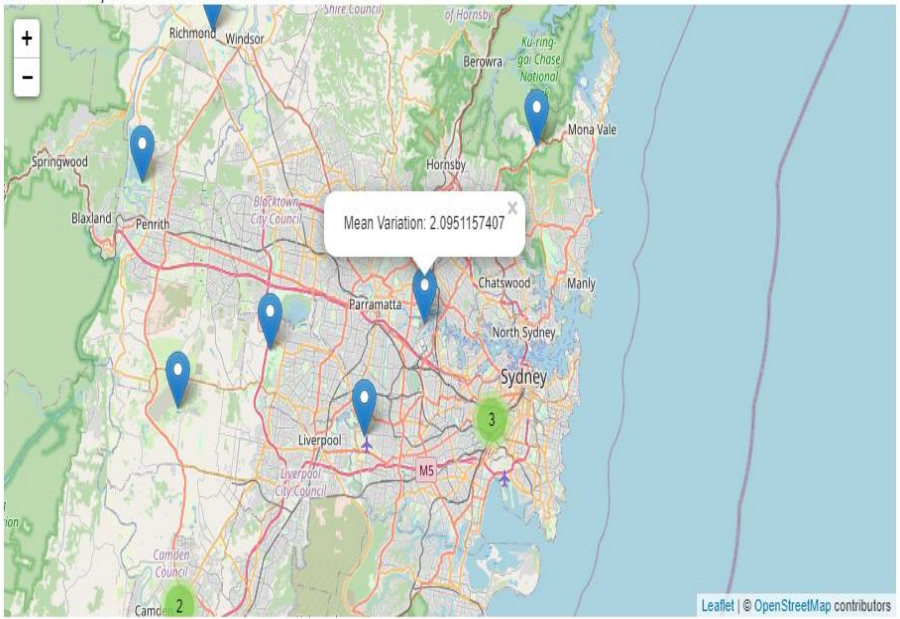
Mean Inaccuracy Of Individual Weather Stations



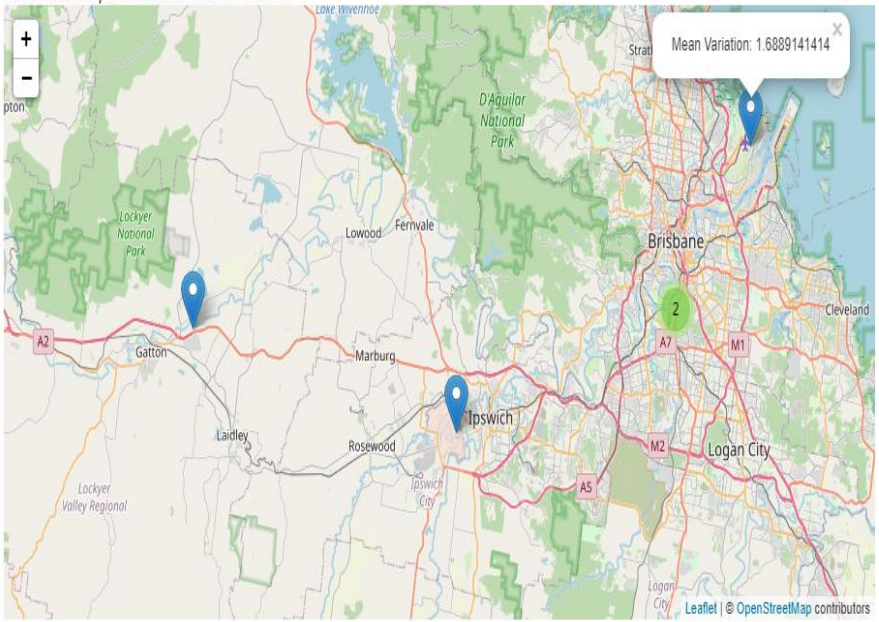
Mean Inaccuracy Of Individual Weather Stations



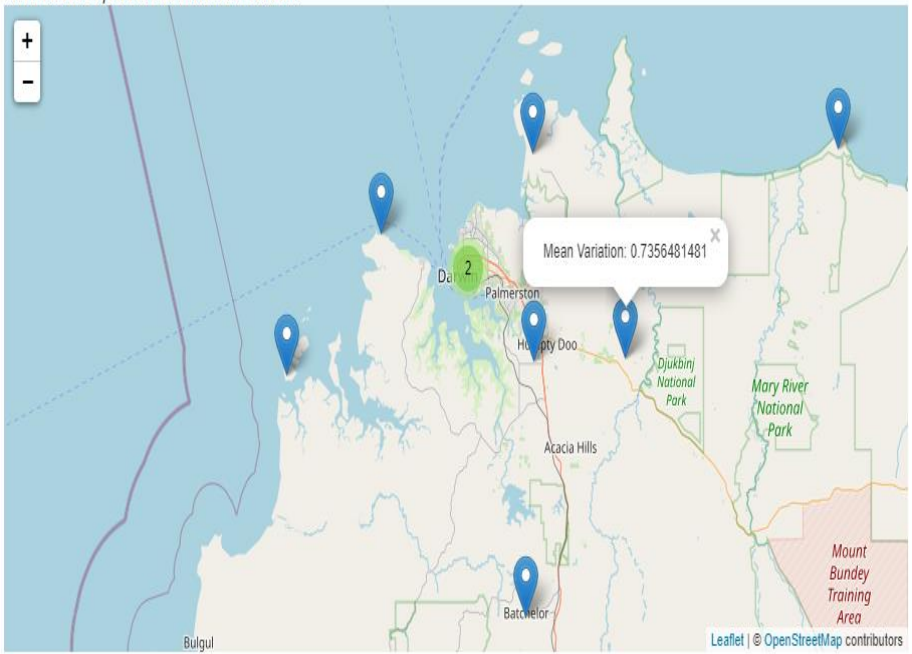
Mean Inaccuracy Of Individual Weather Stations



Mean Inaccuracy Of Individual Weather Stations

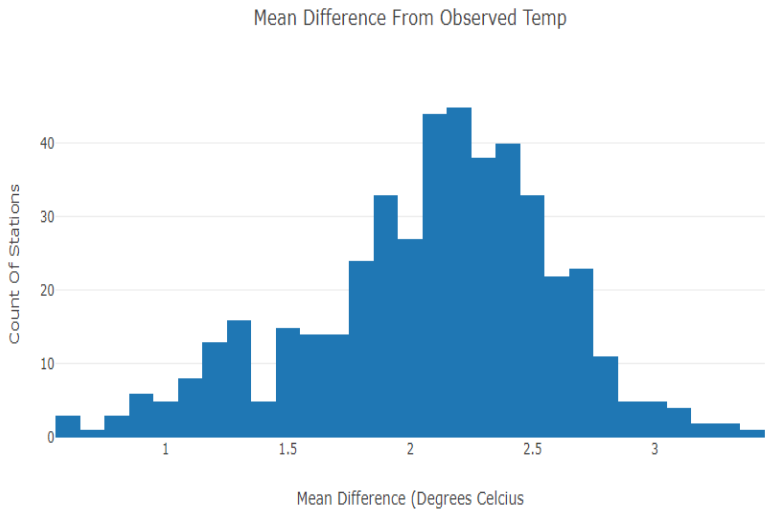


Mean Inaccuracy Of Individual Weather Stations



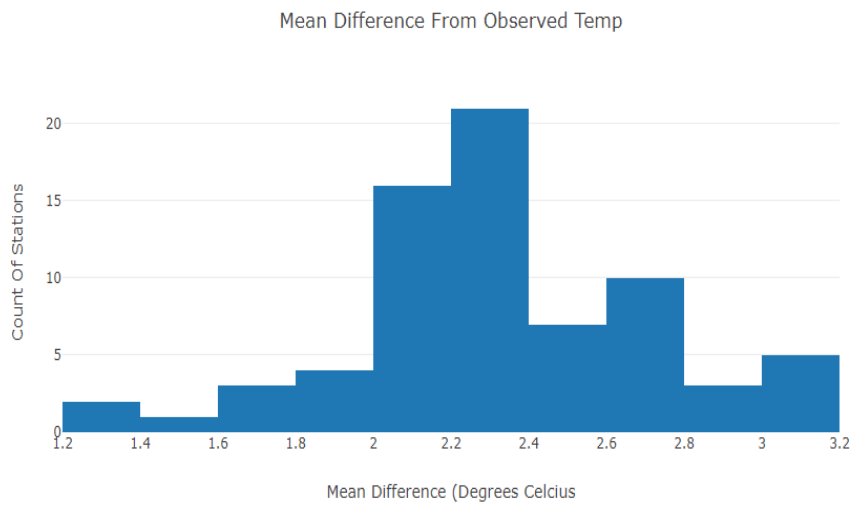
State:

All



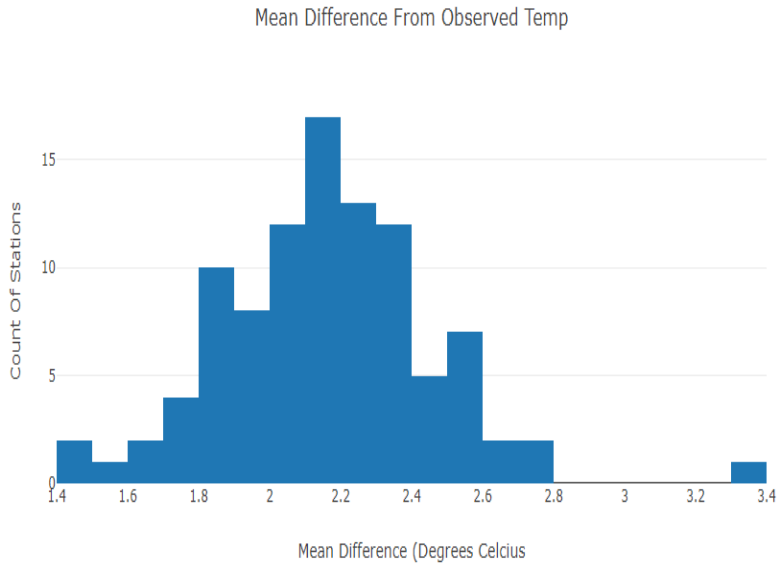
State:

VIC



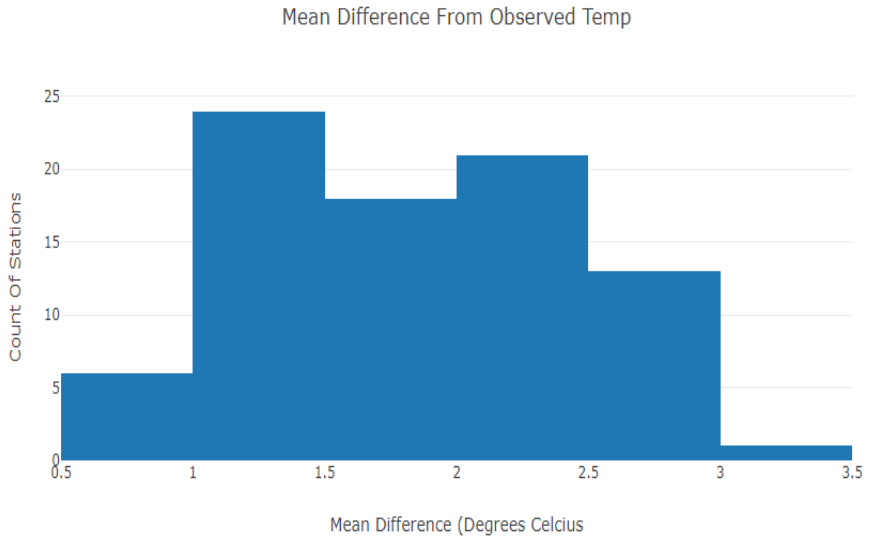
State:

NSW



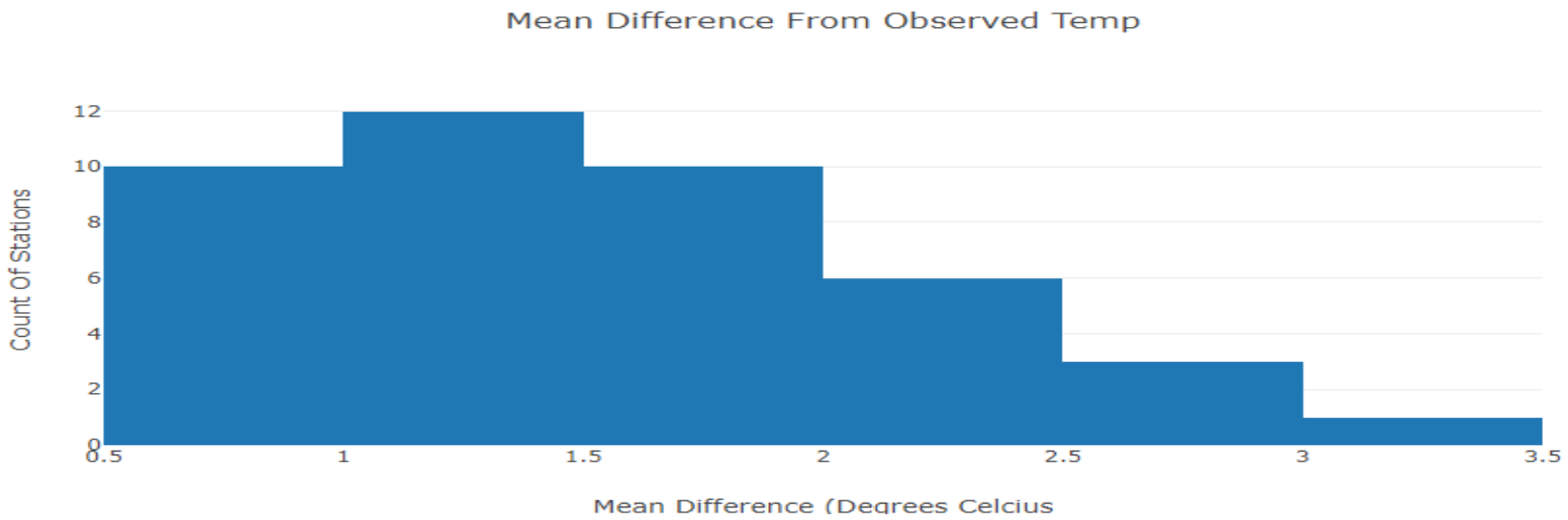
State:

QLD



State:

NT


Mean Difference From Observed Temp
Count Of Stations
12
10
8
6
4
2
0
0.5
1
1.5
2
2.5
3
3.5
Mean Difference (Degrees Celcius

Conclusion

- We made conclusions based on mean temperature of weather stations.
- Based on analysis NSW and VIC have less accuracy of weather station temperature.
- While NT and QLD have higher accuracy of weather station temperature.


thank you!