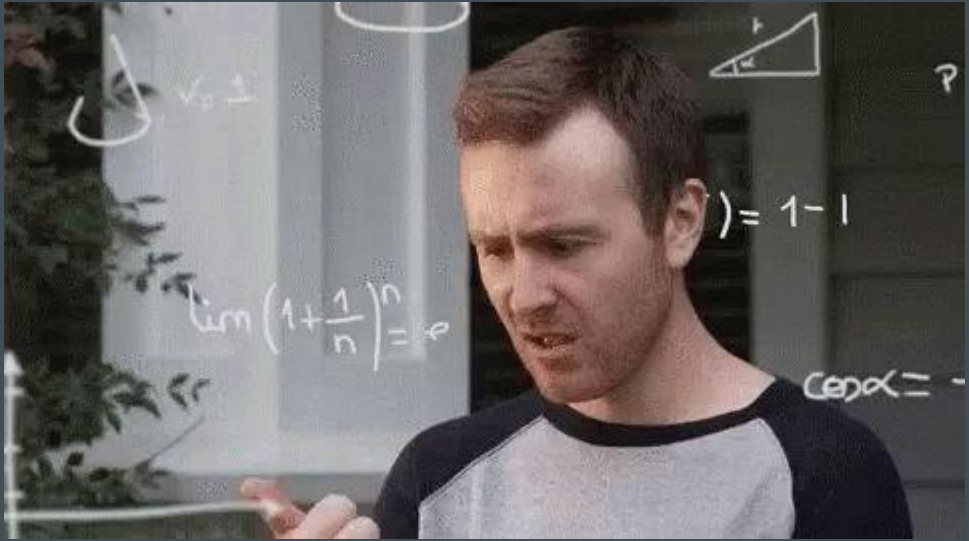
Weather web visualization

Represent by : John Antony
Liam Baker
Nishant Patel

Question!

Is there any strategies to predict weather?



Objectives

- ➢ Data cleaning and data wrangling
- ➢ Data store with sql database
- ➢ Python Flask api route and javascript
- ➢ Leaflet and plotly
- ➢ Coding parts for getting mean variations of weather station
- ➢ Screenshots of final outputs
- ➢ Which states have higher weather accuracy?
- ➢ Which states have lower weather accuracy?

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
- Remove null values
- Data cleaning is performed in Jupiter notebook.

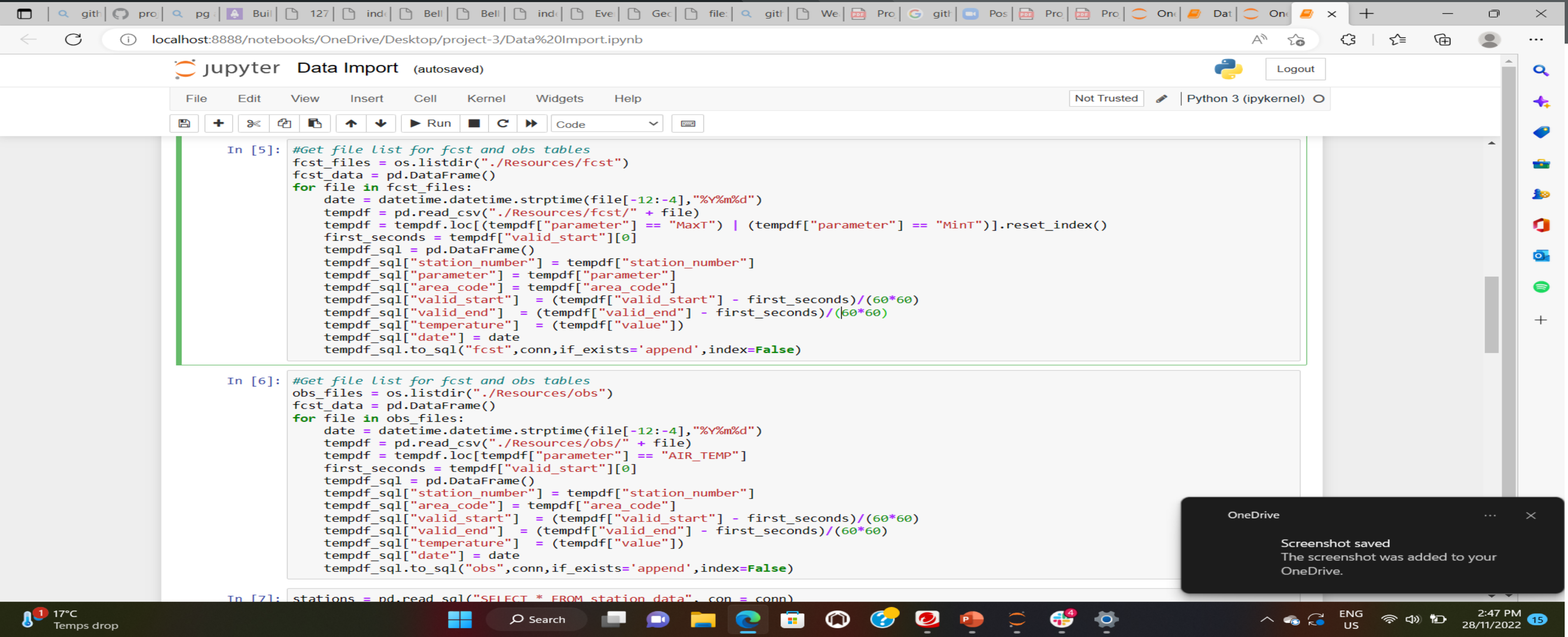
API Routes:

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

Leaflet and Plotly:

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

Coding for getting min and max temperature:



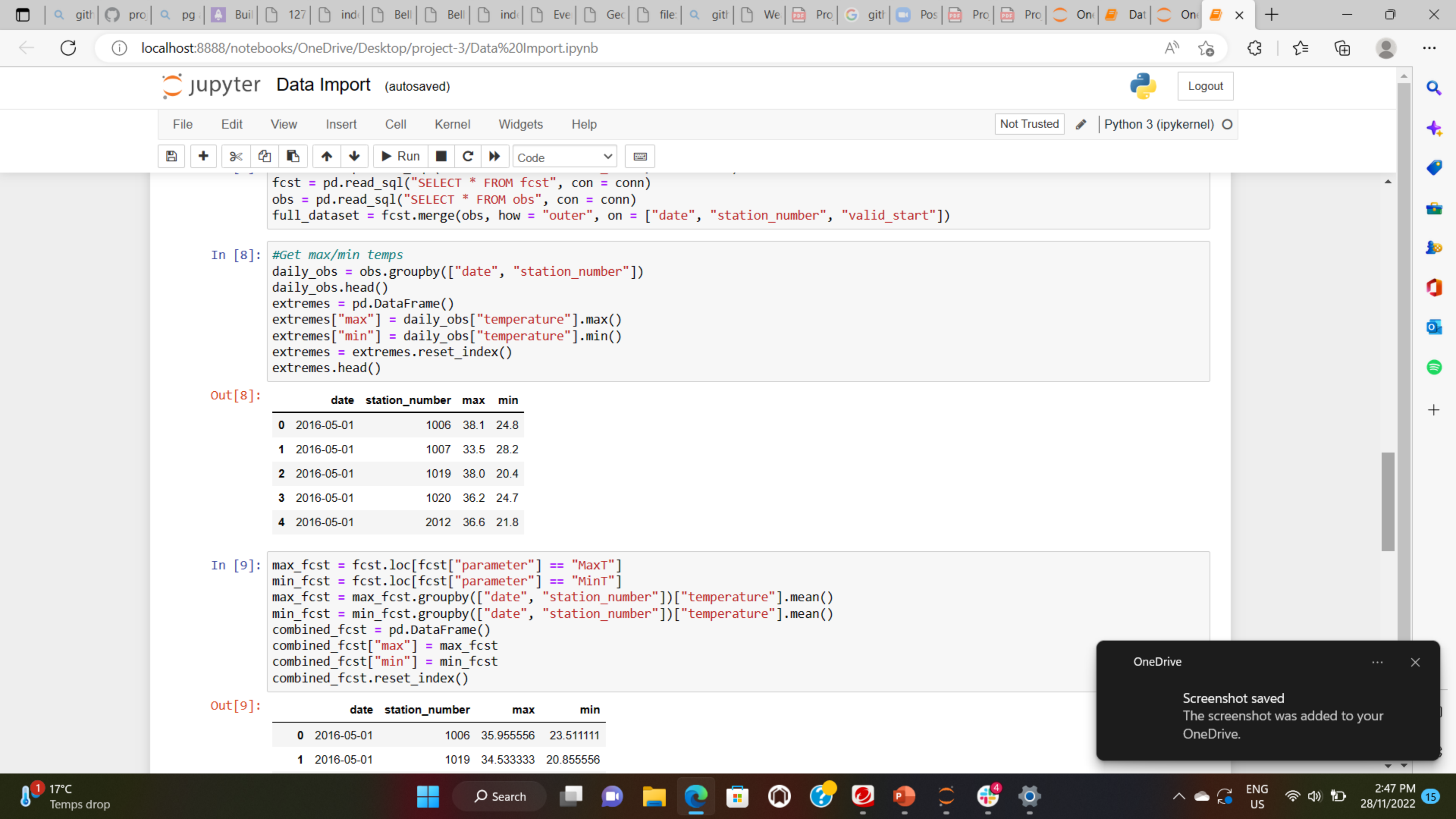
In [5]:
```
#Get file list for fcst and obs tables
fcst_files = os.listdir("./Resources/fcst")
fcst_data = pd.DataFrame()
for file in fcst_files:
    date = datetime.datetime.strptime(file[-12:-4],"%Y%m%d")
    tempdf = pd.read_csv("./Resources/fcst/" + file)
    tempdf = tempdf.loc[(tempdf["parameter"] == "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 [7]:
```
stations = pd.read_sql("SELECT * FROM station_data", con = conn)
```

OneDrive

Screenshot saved
The screenshot was added to your OneDrive.



```
fcst = pd.read_sql("SELECT * FROM fcst", con = conn)
obs = pd.read_sql("SELECT * FROM obs", con = conn)
full_dataset = fcst.merge(obs, how = "outer", on = ["date", "station_number", "valid_start"])
```

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]:

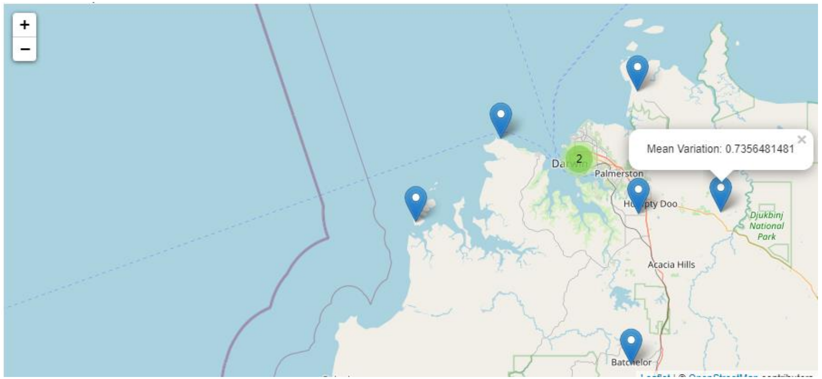
	date	station_number	max	min
0	2016-05-01	1006	35.955556	23.511111
1	2016-05-01	1019	34.533333	20.855556

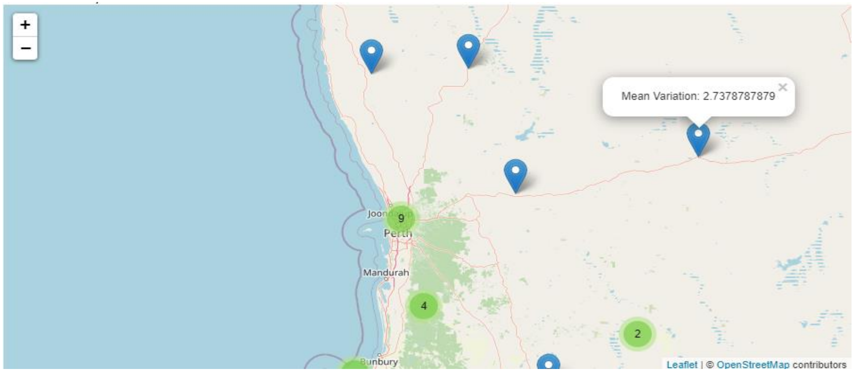
OneDrive

Screenshot saved
The screenshot was added to your OneDrive.

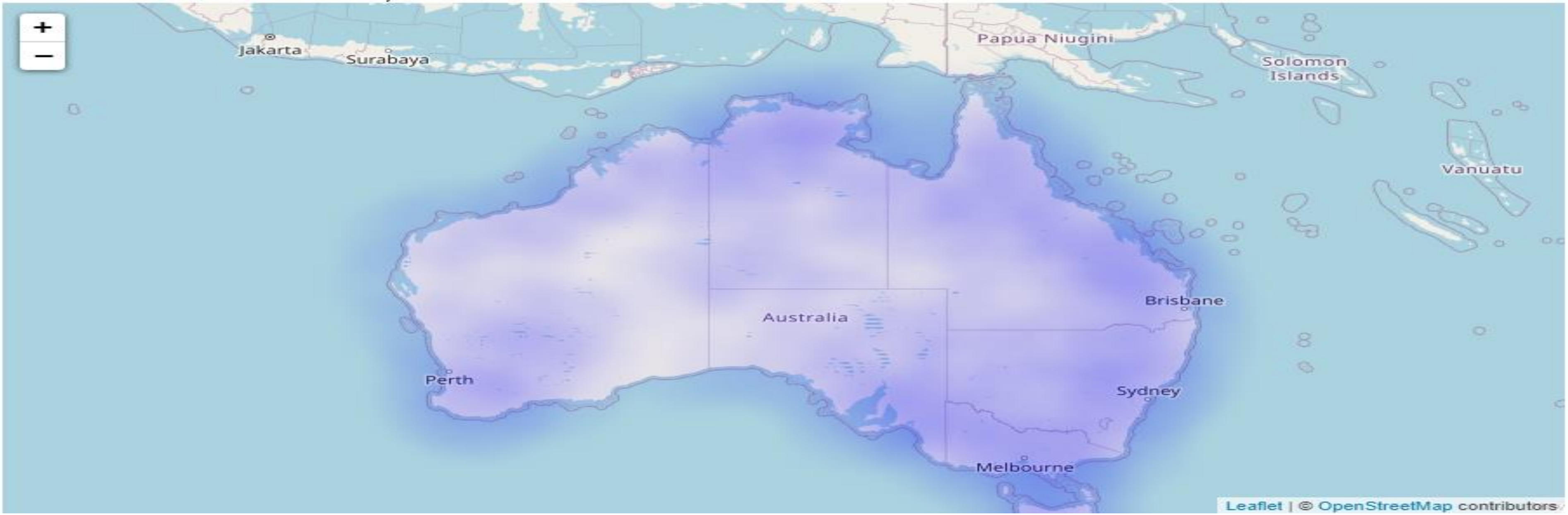
Conditions:

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



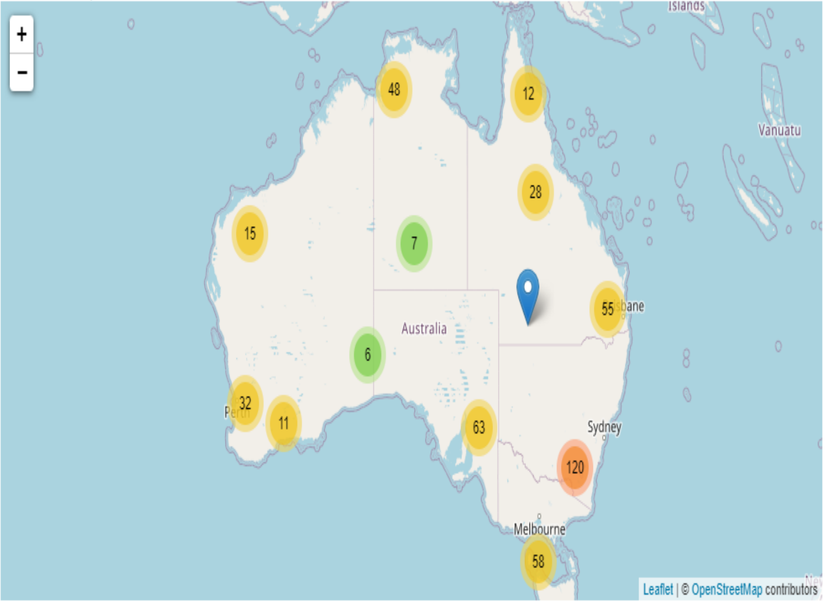

+
−
Mean Variation: 2.7378787879
Joon
9
Perth
Mandurah
4
2
Bunbury
Leaflet | © OpenStreetMap contributors

Graphs and maps

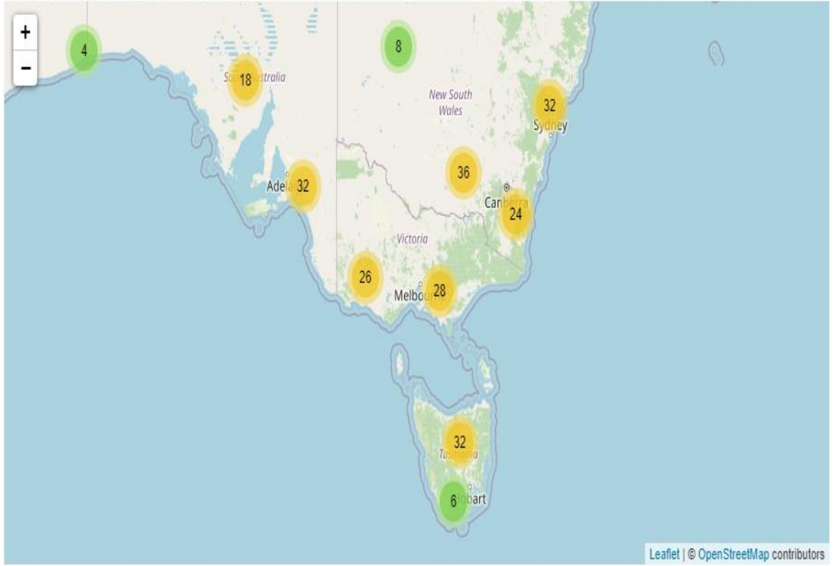


Weather Observation Station Density Across Australia

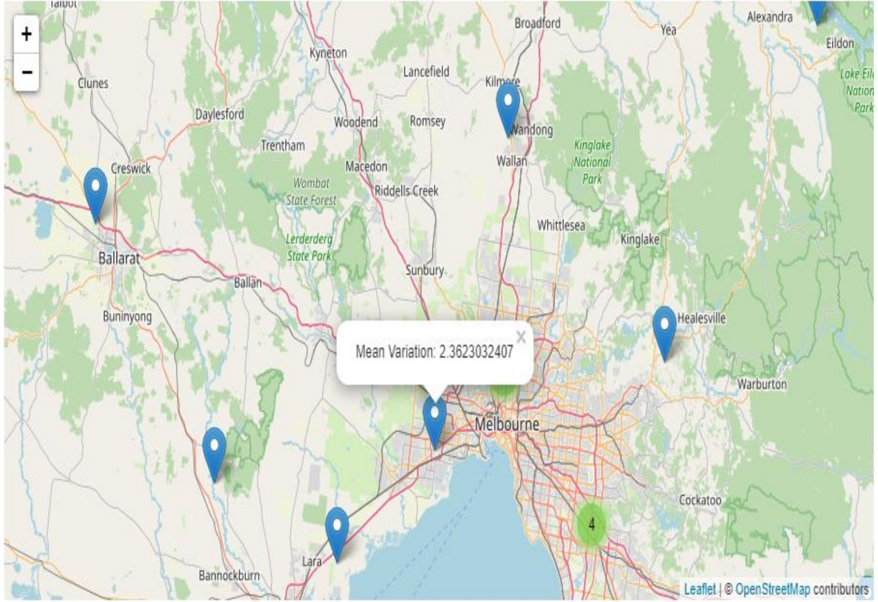
Mean Inaccuracy Of Individual Weather Stations



Mean Inaccuracy Of Individual Weather Stations

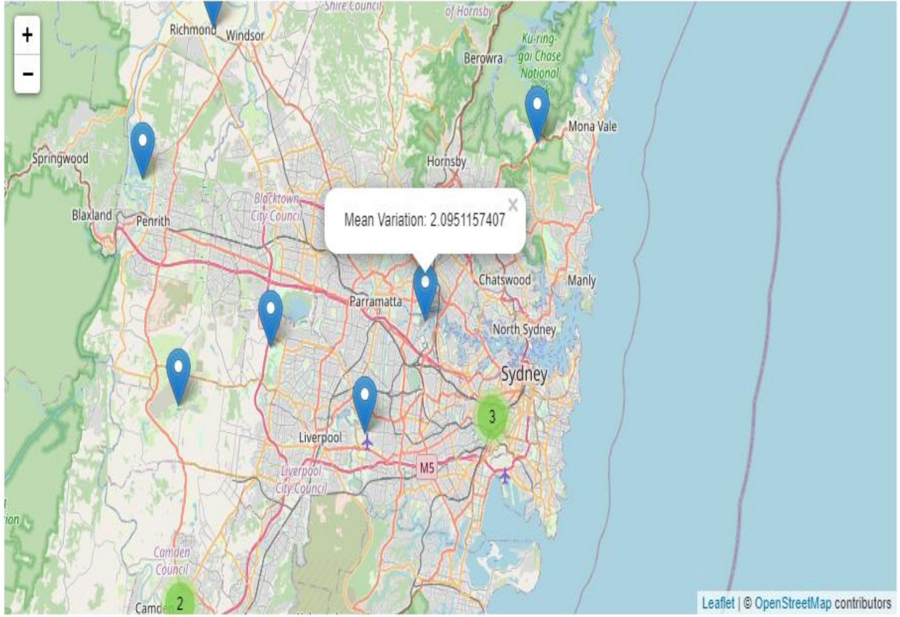


Mean Inaccuracy Of Individual Weather Stations

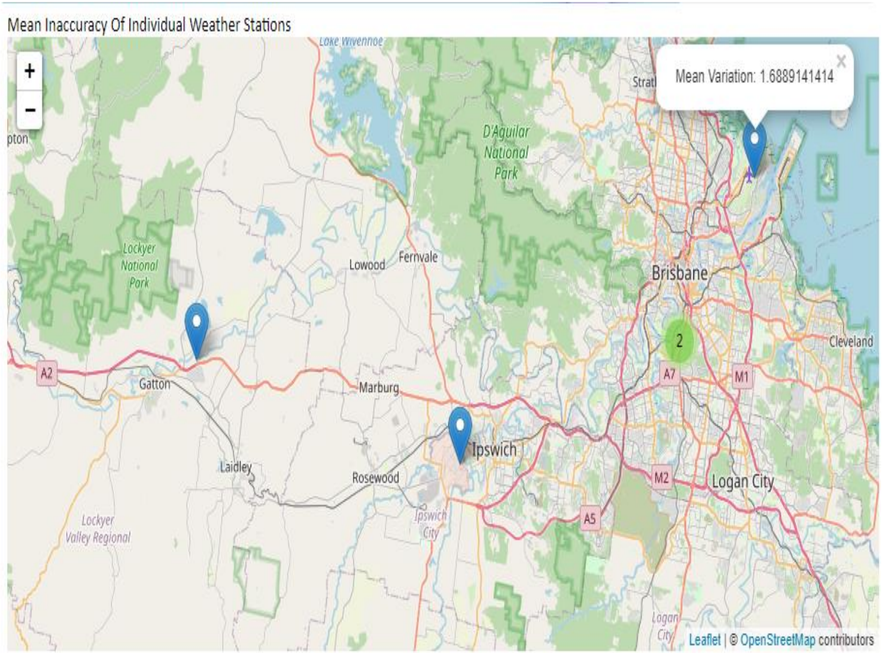


Mean Variation: 2.3623032407

Mean Inaccuracy Of Individual Weather Stations

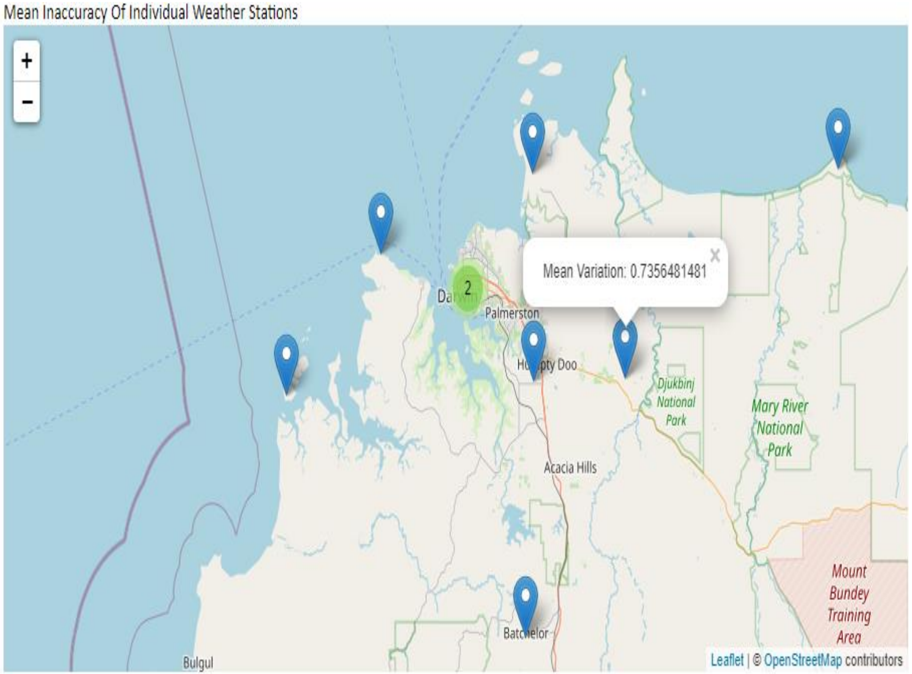


Mean Variation: 2.0951157407



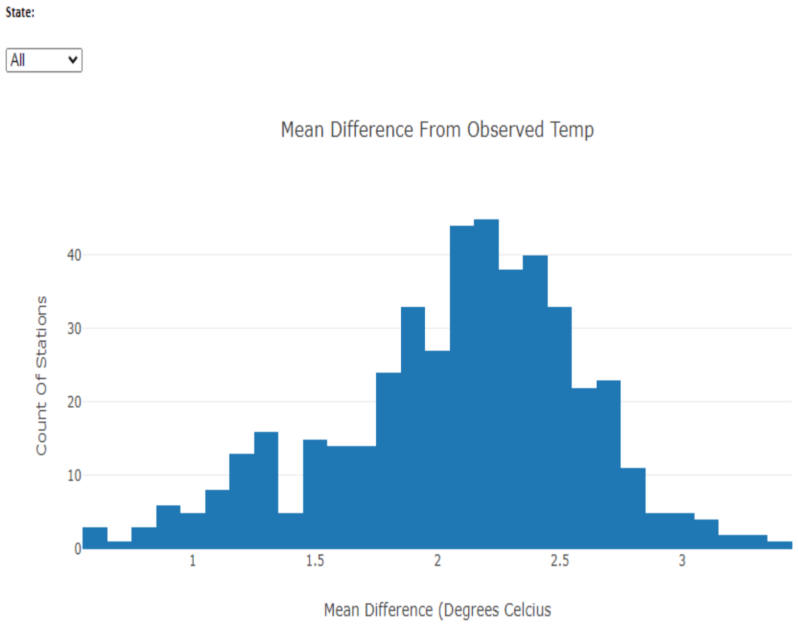
Mean Inaccuracy Of Individual Weather Stations

Mean Variation: 1.6889141414



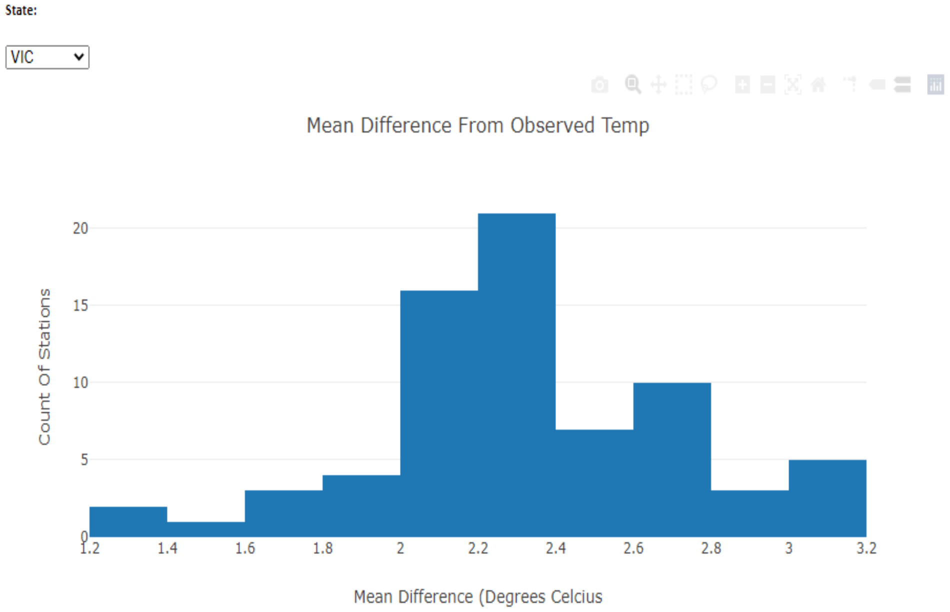
Mean Inaccuracy Of Individual Weather Stations

Mean Variation: 0.7356481481



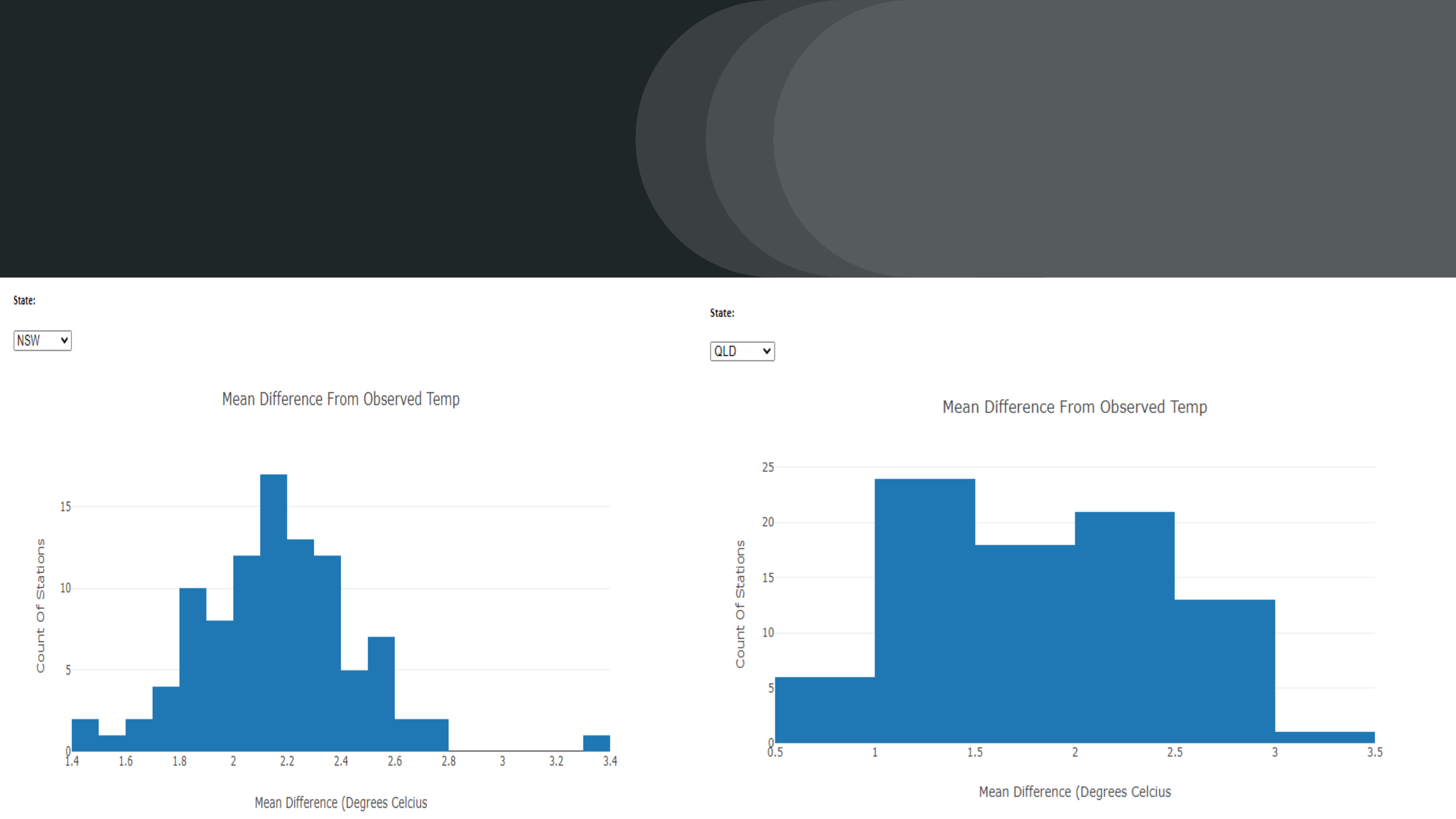
State: All

Mean Difference From Observed Temp
Count Of Stations
0
10
20
30
40
1
1.5
2
2.5
3
Mean Difference (Degrees Celcius



State: VIC

Mean Difference From Observed Temp
Count Of Stations
0
5
10
15
20
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
Mean Difference (Degrees Celcius



State:

NSW

Mean Difference From Observed Temp

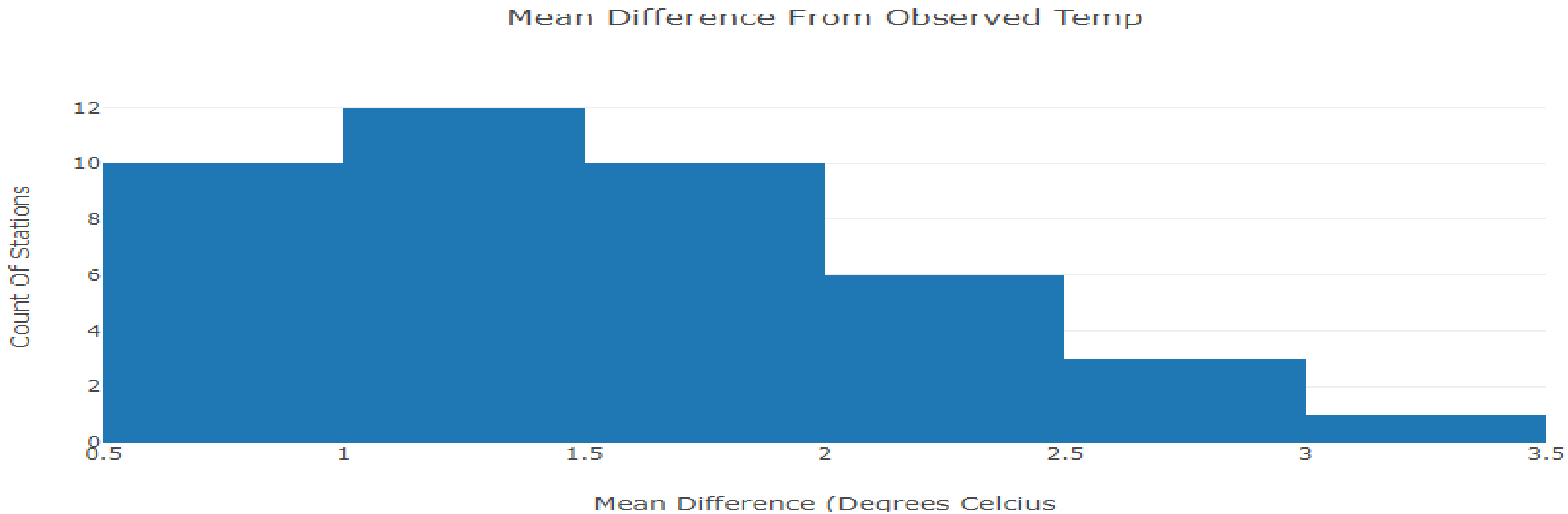
State:

QLD

Mean Difference From Observed Temp

State:

NT


Mean Difference From Observed Temp
Count Of Stations
0
2
4
6
8
10
12
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!