

## DATA 200 Spring 2020 Homework 5

Due March 13, 2020 at 11:59pm

**Note:** Please only use python standard libraries covered in class lectures (i.e. json, csv, urllib etc). Other libraries such as pandas, numpy are *\*not\** allowed in your work.

1. This problem involves pulling data from public data APIs, deriving results from the data and outputting the computed results as a json file. You will utilize the APIs from MetaWeather (<https://www.metaweather.com/api>). For details on how to use their APIs to get weather data, please refer to the site's documentation and the jupyter notebook that was covered in class. Write Python code to perform the following:

- i. Pull daily weather data for San Francisco for the month of February (2/1/2020 – 2/29/2020). Filter out all the forecast weather data (i.e. only include data which has the “*applicable\_date*” on the same day as the “*created*” field). For example, the following shows the partial data (only *created*, *applicable\_date* and *the\_temp* are shown) for 2/27/2020. Only the rows with “*created*” on 2020-02-27 should be kept (shown in red) and all the others should be filtered out.

```
created, applicable_date, the_temp
2020-02-28T06:16:17.158155Z, 2020-02-27: 21.48
2020-02-28T03:16:18.485130Z, 2020-02-27: 21.37
2020-02-28T00:16:19.035257Z, 2020-02-27: 22.425
2020-02-27T21:16:16.650683Z, 2020-02-27: 22.509999999999998
2020-02-27T18:16:16.726534Z, 2020-02-27: 17.96
2020-02-27T15:16:17.638181Z, 2020-02-27: 17.925
2020-02-27T12:16:18.432348Z, 2020-02-27: 18.44
2020-02-27T09:16:16.569841Z, 2020-02-27: 17.11
2020-02-27T06:16:20.383380Z, 2020-02-27: 18.97
2020-02-27T03:16:22.092276Z, 2020-02-27: 18.78
2020-02-27T00:16:19.599788Z, 2020-02-27: 18.405
2020-02-26T21:16:19.986096Z, 2020-02-27: 18.560000000000002
2020-02-26T18:16:20.506721Z, 2020-02-27: 18.759999999999998
2020-02-26T15:16:19.896683Z, 2020-02-27: 18.68
2020-02-26T12:16:20.614150Z, 2020-02-27: 18.655
2020-02-26T09:16:20.722575Z, 2020-02-27: 18.07
2020-02-26T06:16:22.915857Z, 2020-02-27: 17.900000000000002
2020-02-26T03:16:23.117203Z, 2020-02-27: 17.7
2020-02-26T00:16:23.447687Z, 2020-02-27: 18.56
.
.
.
```

You may notice that the first 3 rows have “*created*” date as 2/28 with the “*applicable\_date*” of 2/27. The reason is that the datetime string for “*created*” is in UTC timezone which is 7 hours ahead of Pacific Daylight Time (PDT) and so those

3 rows are still technically data on 2/27 PDT. For simplicity, you don't have to take this timezone offset into account and only include the rows that fall on the same day as the applicable date.

- ii. Compute the maximum, minimum and average temperature for each day in February from the temperatures extracted in previous step. For example, for 2/27, you will compute the maximum, minimum and average temperature using the 8 temperature values (in the red bold rows above) for that day.
- iii. Write out the maximum, minimum and average temperatures by day to a json file (**temp\_stats.json**) and it should look like this (again the temperature values shown below are not the true values):

```
{
  "2020-02-01": [16.59, 12.51, 14.001875],
  "2020-02-02": [12.2, 10.85, 11.401250000000001],
  .
  .
  .
  "2020-02-29": [14.561 12.785 13.47]
}
```

(20 points)