Problem Set #5

Assignment Learning Objectives

1. Use meteorological data in an appropriate context
2. Use Python modules to help read and analyze meteorological data
3. Develop programming skills to break down a problem into smaller units

Due 15 October 2020 by 3:00 pm

Due Online

- One script in the PSet5 repository on your GitHub account

Due Via Blackboard

- A PDF version of your code. Make sure the honor code appears in the comment block at the beginning of the script and that it has your full typed out name. This will serve as your assertion that you have upheld the honor code.

Honor Code

You must type the full honor code into each comment block to indicate that you have upheld the code. Authorized aid for this assignment is your notes and any previous programs you have written. You are highly encouraged to work with others in the class to help diagnose bugs and work out programming logic. Any copying of someone else's code IS A VIOLATION OF THE HONOR CODE, whether from a classmate, or the Internet. *Be sure to indicate with whom you have worked in the comment block of your submission.*

Background

It is common to look at a time series of observed data for a single station and it is useful to visualize that time series in a plot called a meteogram. There are any number of different ways to construct this diagram, but they all share the same x-axis for time. The best method is to do a multi-panel plot where some panels have more than one variable plotted. For example, temperature and dew point are often plotted together in a panel and wind speed and direction together in a separate panel. A great example of a meteogram can be found from the Oklahoma Mesonet site at: http://mesonet.org/index.php/weather/meteogram/nrmn/. You do not have to follow this example exactly (nor the one attached to this assignment), but it is a high-quality example. If you search for meteogram (or meteorogram) you'll find more examples.

In addition to plotting data, this assignment will have you compute a few basic statistics for the daily observations. The observations including regular hourly and special reports, so not all data are evenly spaced in time -- this doesn't matter too much for this particular assignment but is important to keep in mind.

Information about the data contained in the file is available at:
https://mesonet.agron.iastate.edu/request/download.phtml?network=IN_ASOS

Problems

Program Name: **meteogram.py**

Input: Have command line input for two different pieces:

```
    Station Three-Letter ID (e.g., ORD)
    Date as YYYYMMDD
```

Output: That specified in problem 2 and the meteogram figur (figure needs to be saved as **<station>_meteogram_<YYYYMMDD>.png**

1. Create a script that will read data from a surface file in the format given w with the example file and generate a meteogram for that days worth of data(which is for ORD for March 12, 2014). *NOTE: A different file will be used to grade this assignment.* The goal is to have a script that could accept any file (with the same format), but it could be for any location and day. The meteogram must include temperature, dewpoint, altimeter (in Hg), wind speed and direction, and precipitation accumulation throughout the day. At runtime the user should be asked for two pieces of input: 1) station three-letter identifier (e.g., ORD) and 2) date in YYYYMMDD format.

   Surface data for a given date starts 10 minutes before 0000 UTC to 2349 UTC. For example, the surface observations for March 12, 2014 is from 2350 UTC 11 March to 2349 UTC 12 March 2014.

2. Compute and print to the screen the following daily statistics for the observation location.

   a. Report the days statistics of maximum, minimum, and average for temperature, dew point temperature, and altimeter pressure.

   b. Report the days maximum, minimum, and average wind speed with the direction for the max and min.

   c. Time of maximum and minimum values for each variable except precipitation.

   d. Total accumulated precipitation in inches.

   e. Time spent at or above 32ºF, time spent below 32ºF.

Notes on print() formatting:

- https://docs.python.org/3/tutorial/inputoutput.html

Notes on Matplotlib:

- Multi-panel Plots
  - https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html
- Formatting x-axis tick labels
  - https://scentellegher.GitHub.io/programming/2017/05/24/pandas-bar-plot-with-formatted-dates.html
- Same x-axis, two different y-axis
  - https://stackoverflow.com/questions/14762181/adding-a-y-axis-label-to-secondary-y-axis-in-matplotlib

- Legend with two different y-axes
    - https://stackoverflow.com/questions/5484922/secondary-axis-with-twinx-how-to-add-to-legend
- Fill between axes
    - https://matplotlib.org/3.3.1/gallery/lines_bars_and_markers/fill_between_demo.html

Example Output

Text Output:

Input Station Three-Letter ID (e.g., ORD): ORD

Input Date: 20140312

Daily Statistics for ORD on 2014-03-12

Statistics for Temperature

Max: 37.94 F at 2014-03-11 23:51:00

Min: 26.06 F at 2014-03-12 16:51:00

Mean: 30.87 F

Statistics for Dew Point Temperature

Max: 30.02 F at 2014-03-12 02:51:00

Min: 6.08 F at 2014-03-12 22:51:00

Mean: 21.09 F

Statistics for Altimeter Pressure

Max: 30.02 in Hg at 2014-03-12 22:51:00

Min: 29.68 in Hg at 2014-03-12 06:51:00

Mean: 29.81 in Hg

Statistics for Wind Speed

Max: 24.0 kt at 2014-03-12 13:51:00

Max Wind from 10.0 degrees

Min: 9.0 kt at 2014-03-12 02:51:00

Min Wind from 360.0 degrees

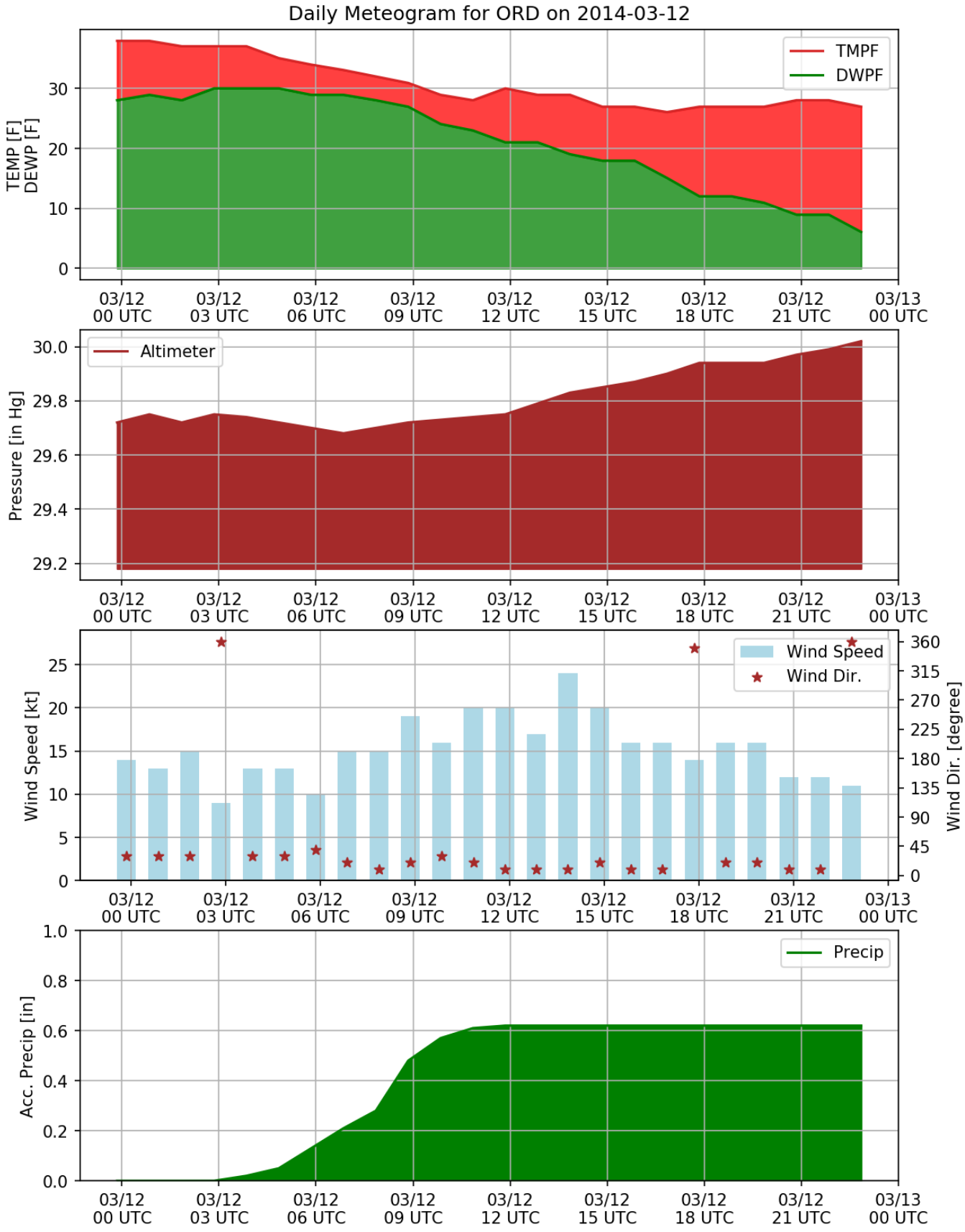
Mean: 15.25 kt

Precipitation Accumulation: 0.6201 in

Hours at or above 32F: 9

Hours below 32F: 15

Meteogram:



Evaluation Criteria

Each of the following criteria will be rated from not present/completed to exemplary, having all of the elements will yield at least a 7/10 for a particular criterion. The assignment is out of 50 points.

- Efficiently coded and correct Problem 1 (15 points)

- Efficiently coded and correct Problem 2 (10 points)
- Overall code is structured and styled well (5 points)
- Informative and Clear Output from running code (10 points)
- Code is well documented (informative comments/descriptions of code blocks; 5 points)
- Code was regularly updated and committed to GitHub repository (5 points)