CMSC6950 Project: Growing Degree Days

This project is intended to give you experience working with a computational workflow for a scientific problem.

Introduction

From wikipedia.org: *Growing degree day(s) (GDD), also called growing degree units (GDUs), are a heuristic tool in phenology. GDD are a measure of heat accumulation used by horticulturists, gardeners, and farmers to predict plant and animal development rates such as the date that a flower will bloom, or a crop will reach maturity.*

A quick search on Google with the terms 'growing degree day' gives several sites describing this concept. In particular, note that a GDD is defined with reference to both a base temperature and often an upper threshold temperature.

Project specfication

This project can be thought of being made up of a set of tasks to complete. There are minimum core tasks that are completed by all groups in the class. Secondary task offer some choice and flexibility. A Final task is more open-ended.

Minimum Core Tasks

All projects are expected to demonstrate the following components:

1. Download daily historical temperature data for several citiesfrom
    - http://climate.weather.gc.ca (http://climate.weather.gc.ca) This process should be automated. Additional information on bulk downloads is available here:
    - ftp://client_climate@ftp.tor.ec.gc.ca/Pub/Get_More_Data_Plus_de_donnees/
    - You could either use wget as suggested or use the requests python library.
2. Create a plot showing an annual cycle of min/max daily temperatures. Do this for at least three selected Canadian cities.
3. Write a command line program that takes arguments:
    - $ gdd *temperatures.csv tbase tupper*
    - this program should calculate the GDD. Internally your program should handle the command line arguments (looking up the python package argparse could be useful) and implement the actual calculation as one or more functions.
    - The output from this program needs to be persisently stored (written to a file). Your choice on how to implement this storage. Later steps in your work flow must use the results of these calculations.
4. Create plots showing accumulated GDD vs time for selected cities. Examples of such plots can be found on the following webpage.

- http://www.greatnorthwestwine.com/2016/09/05/2016-vintage-cools-down-slightly-in-pacific-northwest/ (http://www.greatnorthwestwine.com/2016/09/05/2016-vintage-cools-down-slightly-in-pacific-northwest/)

5. Use version control (git) and collaboration tools (GitHub) throughout this project. Important: make sure you have properly set up your git configuration with your MUN email address. We will use this information to assess your individual contributations to this team project.

   - Team members should all collaborate on a single github repository. The use of branches is permitted but do not use github forks and pull requests.
6. Create a LaTeX report summarizing the results of your project.
7. Create a web based presenation for your results.

   - The remark-js library is nice for doing HTML based presenations
   - Host your presenation on a gh-pages branch of GitHub.
8. Implement your entire workflow as a Makefile. Ensure that your entire project is reproducible.
9. Create a test suite (using the Python package nose) to demonstrate your GDD calculation works as intended (see chapter 18).
10. Your project should include adequate documentation both with your source code and an overall project Readme.md file to explain how to use/build your project. See chapter 19 for some ideas on documentation. While the use of docstrings in your code is encouraged, you do *not* need to set up an automated documenation framework like Sphinx.

Secondary tasks

Do at least four of the following tasks:

1. Create a plot showing GDD, like the example below for selected Canadian cities.

   - https://mesonet.agron.iastate.edu/onsite/features/2015/05/150507.png (https://mesonet.agron.iastate.edu/onsite/features/2015/05/150507.png)
2. Create a map showing effective growing degrees over both all of Canada and only for the island of Newfoundland.

   - http://www.agr.gc.ca/resources/prod/img/env/climat/egdd_prairie_base_eng.jpg (http://www.agr.gc.ca/resources/prod/img/env/climat/egdd_prairie_base_eng.jpg)
3. Explore how GDD calculation depends on the choice of T_base. Show your results for either selected cities or create maps.
4. Create standalone bokeh plots embedded in your HTML presentation so that users can interactively select with the data (hover points)
5. Create a bokeh server plot so that you can look at the accumulated GDD for any city in Canada.
6. Compare GDD year-over-year. Do some analysis (perhaps linear regression) to determine if GDDs have been changing in a statistically significant way.
7. Time/profile your work flow and estimate the amount of time spent in each step. Explore parallelization (either using python multiprocessing or make -j n) to make your work flow execute more quickly.

Final task

1. Free choice. Explore some aspect of the GDD dataset using analysis / visualization. For example, could you make a map across Canada showing the expected date for spring flowers to emerge?

Evaluation

Assessment will based on both the actual results and demonstration of good scientific computing techniques as discussed within the course and the textbook.

- Minimum tasks 55%
- Optional tasks 20%
- Free choice task 5%
- Individual/team participation 10% (based on git contributions)
- Presentation 10%