‘’’ Shane Kramer
Project 1 (MyClimbs) Reflections and Final Write-up
W200
’’’

Idea:

I was very excited to implement MyClimbs for an assignment, since it involves rock climbing training analysis and data visualization, both of which I tend to nerd out about. I also enjoy working on problems with real (and personal) applications and since my daughter and a few friends have been using the same excel template I have been using, the prospect of getting real feedback and iterating/revising was appealing.

Final Functionality:

The final product is capable of analyzing a user’s Bouldering and Summer Top Rope sessions for all years or a specific year, and produces the following products:

- A graph depicting total distance climbed by session date
- A graph depicting VSums (each individual climb being worth its V grade) by session date
- A pyramid graph (frequency distribution for Yosemite grades) for summer roped sessions
- A pyramid graph (frequency distribution for Yosemite grades) for bouldering sessions
- A text based summary summarizing total number of sessions by type, and total vertical distance climbed

```
Analyzing climbs and generating reports ........

--- Number of climbing sessions: 16
  --- Number of bouldering sessions: 9
  --- Number of TR/Sport/Trad sessions: 8
  --- Number of Winter Ropes sessions: 0

--- Total vertical distance climbed this period: 3556
```

Challenges:

I stumbled across a many challenges while working on the project, all of which were honestly unforeseen during the initial design phase.

First, implementing an appropriate OOP framework for my problem space was a bit more difficult than I expected. For example, a climbing session can be one or more types (which is why number of sessions shown above is actually less than number of bouldering sessions + number of TR/Sport/Trad sessions), some reports are only generated for a specific session type, grades vary (wildly) according to type of session, and indoor and outdoor grades are horribly inconsistent, which skews the frequency distribution graphs. Essentially I began work with a simple is-a, has-a design in mind, and ended up having to navigate through a series of many-to-many relationships.

I also ran out of LOCs! I wanted to implement the winter roped session stuff (and now it should be somewhat trivial), but was well over the Lines-of-code threshold. I also had a couple of more ideas for graphs, including a stagnation graph, which depicts how long a user stagnates on a max grade until they bump up to a new grade. I tinkered with this a bit and it would have taken about 100 lines or code I think. I am fairly confident that MyClimbs can be downsized by better utilization of lambdas, comprehensions, and even regular expressions, though I tend to prefer (at my current skill level) more readable code.

Another interesting little challenge I faced was sorting climbing grades, without the use of third party software. I did not think this would be much of an issue, but ended up having to resort to reading through forums and implementing a hand-built (albeit hand-built by someone else) natural sort function.

Lastly I still have some peculiar reservations regarding class attribute scope in Python. My IDE warned me about 100 times about the use of “sessions” in my classes and functions. In java, I always add “m_” to the beginning of class member variable names and “p_” to the beginning of function parameter/argument names, but I am not sure this is “Pythonic”.

What I would like to change/add:

As I mentioned above, I would like to add functionality for Winter Roped Sessions, as well as 2-4 more graphical products. The stagnation graph is the idea I am most excited about currently. I also plan on passing MyClimbs around to friends and revising items based on their feedback.

Lessons Learned:

On the positive side, I learned that file IO is really pretty easy with Python, much more so than a few other languages I have used. This was refreshing. I also found a cool little Pep8 plugin for my IDE, which made handling code style easy. On the negative side, I learned that Python does not have a native natural sort algorithm for alpha-numeric strings. I also learned that I need to practice lambas, regular expressions and list comprehensions more so their use is a bit more natural to me.