Stat 222: Mouse Lifestyles

March 28, 2016

1 Data Description

For this project, your primary data source will be mouse behavioral data from the Tecott Lab at UCSF.[1] The lab has recently developed a method for continuous high-resolution behavioral data collection and analysis, which enables them to observe and study the structure of spontaneous patterns of behavior ("Lifestyles") in the mouse [4, 5, 2, 1]. They have found that using this method: 1) reveals a set of fundamental principles of behavioral organization that have not been previously reported, 2) permits classification by genotype with unprecedented accuracy, and 3) enables fine dissection of behavioral patterns.

A central goal for this project is to give you hands-on experience working on a real-world applied statistics[2] project. In the real-world, problems are not pre-packaged as a numbered list of short questions. So understanding the problem is often the first step in an applied statistics project. As you start to come to an understanding of the problem, questions will emerge. And as the questions become clear, you will need to determine how you will attempt to answer them. This process will be slow and at times vexing. While the Tecott Lab, Johnny, and I will provide guidance on the project, you will ultimately be responsible for determining what you do.

2 Your Assignment

We will work with the Tecott Lab to understand the problem and then to refine the project scope and focus. In order to manage all of your contributions, I will require that (1) all contributions undergo a rigorous review process, (2) all code is tested and that those tests are included as part of our automated test suite, and (3) all code and text components follow the best practices adopted by many open source scientific Python packages [3]. All project code will be distributed under the simplified BSD license and test data will be made publicly available. Project documentation will be published using GitHub Pages.

[1] http://www.neuroscience.ucsf.edu/neurograd/faculty/tecott.html

[2] See Philip Stark's "Thoughts on applied Statistics" here: http://www.stat.berkeley.edu/~stark/other.htm

2.1 What's different about this assignment?

Class assignments commonly have well-defined problems to solve. When those assignments involve data, the typical situation is that the data are clean and trivial to work with. This assignment is intended to closely mimic an applied statistics project in the real world. A large part of the work will be understanding the nature of problem, and making that understanding clear in discussion, documentation, and code. It will be your responsibility to convince your classmates, instructors, and project sponsors that you have done so. The nature of the work is open-ended and uncertain. If you haven't done this kind of work before, it may make you uncomfortable. But if you persist, this will be excellent preparation for your future working life.

Given the unusual nature of the assignment, it will be instructive to distinguish the following stages of the project:

1. **Understand the fundamental problem.**

   This is the starting point for the project and is the unifying thread that will run throughout all the elements of the project. The problem provides the context for why you are doing what you are doing. It leads to the research questions and hypotheses as well as guides the conclusions you make from the results of your analysis.

2. **Data acquisition, pre-processing, and cleaning.**

   As applied statisticians, we would ideally help in the experimental design part of the project. However, in this project (as in many applied statistics projects), the experiment has already been conducted and the data collected. So your next step, will be to understand the data that has been collected. You will also need to verify that the data you received is correct (as far as you can tell) and that your understanding of what it represents is correct.

3. **Algorithm development.**

   Before you can start writing code, you will need to decide what you want to do. While you will want to consider computational efficiency and numerical stability, your first and primary objective will be to determine whether your approach to the problem is sensible given the problem you wish to address and the data you have available.

4. **Code development.**

   While you will start coding fairly early in the project, the majority of the coding will occur in the second half of the project. Perhaps unlike your previous experience coding, the goal is not to produce a stream of consciousness script. Rather you will carefully engineer a well-designed, rigorously tested, systematically documented codebase. This will involve design discussions, use case considerations, and a systematic code review process. It is likely that code will be heavily refactored and improved as part of the project.

5. **Data analysis.**

   Your training so far is likely to have been focused on the aspect of the work. Yet, this will be the last step and least involved aspect of this project. Once you've understood the problem and available data, determined and implemented your algorithms, the final data analysis step should be more or less straightforward.

While I've sequentially enumerated the above stages of the project, it is unlikely that the project will proceed in an entirely linear fashion. Rather you will likely proceed in accord with something more like a spiral plan. In the spiral plan, you would proceed in the order 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5.

Timeline and logistics

Here is the tentative schedule:

Monday	Wednesday
(3/7) Project introduction	(3/9) Git I
(3/14) Git II	(3/16) Git III
Spring break	*Spring break*
(3/28) Start final project	(3/30) TBD
(4/4) Project check in	(4/6) TBD
(4/11) Project check in	(4/13) TBD
(4/18) Presentation	(4/20) TBD
(4/25) Project check in	(4/27) TBD
(5/2) Project check in	(5/4) TBD
RRR week	*RRR week*
Final week	*Final week*

For the remainder of the class we will be focused on the final project. Every Monday will be devoted to merging pull requests and planning the week's work. Depending on how things progress, Wednesdays will be used for discussing assigned readings or student presentations relating to the project.

On Monday, March 28th, I will go over the general project plan in class as well as orient you to the project repository. I will ask you to form 11 teams of 3 members each by Wednesday, March 30th. When forming teams, you will need to make sure that there aren't any major weaknesses on the team. Each team will need a mix of strengths including strong written and verbal communication skills, core mathematical and statistical competency, as well as a programming maturity.

On Wednesday, March 30th, teams will be assigned one of several tasks: 1) project overview, 2) subproject descriptions, 3) project infrastructure, and 4) data loaders and tests. As a class, we will discuss the various tasks and come to some agreement about what

they entail. The project overview and subproject descriptions will focus on understanding the fundamental problem and will involve writing a roughly one page description of the overall project as well as one page descriptions of each subproject. As you are initially trying to understand the fundamental problem(s) addressed in this project, you may want to focus as much on determining what parts of the problem you don't understand as on the explaining the parts you do. When trying to think about what you don't understand, you should try to distill your lack of understanding in to simple, direct questions. During this process you may find that you come to better understand the problem. If not, you will at least have specific questions to direct (in order) to your classmates, teaching staff, and problem sponsors.

For each one page description, one or more teams will prepare a pull request by the beginning of class on Monday, April 4th. I've created a Git repository for you with a directory with the following structure inside of the `doc` directory:

```
problem/
|-- overview.md
|-- behavior.md
|-- path.md
|-- dynamics.md
|-- ultradian.md
|-- classification.md
`-- distribution.md
```

The pull requests will be reviewed by the project sponsors as well as the Johnny and me. I will also ask each team to review another team's project description.

I will discuss the details of the other tasks in more detail in class. While I've created the tasks for each team for the first week, I will rely on you to help define the tasks moving forward. However, I expect that each team will prepare one or more substantial pull requests each week, which will be ready for review before class on the following Monday.

I expect everyone to follow all aspects of the class project. To help meet my expectation, I will require that for the first few weeks teams will take turns leading and reviewing the various aspects of the project.

On Monday, April 18th, you will present the initial work on the project in class. Our project sponsors will attend the presentation, so I expect to see something substantial, professional, and interesting. At a minimum, I expect that you will have made substantial progress on understanding the problem, the data, and the algorithms. You should also have made progress on the implementation. In particular, I expect that you will have massively refactored the code given to us by the project sponsors. This refactoring should include making the code more modular as well as documenting and testing everything thoroughly.

Part of the goal of the presentation on April 18th will be to propose what each team will focus on for the last half of the project. While I will require every team to work on all aspects of the project before the 18th, it will be up to you to determine how you will

divide up the work remaining in the project. If there are several teams that want to work on one subproject, I may have to assign final projects. If this happens, I will tend to assign teams to their preferred projects based on how much they've contributed during the first part of the project.

3 Python package

By the end of the semester, you will have produced a Python package including extensive and high quality online and printable documentation (you should view the project documentation as your final report). The entire class will be responsible for the Python package and there will be one grade for the final project. While there will be one grade for the final project, I reserve the right to fail anyone who doesn't fully engage in the work.

I will be responsible for creating the initial project infrastructure on GitHub. If there are code or infrastructure issues that the class can't agree on, I will be responsible for making the final decision. However, before I intervene, you will need to carefully think through the issue and prepare arguments for and against any decision you wish me to make.

While you will be responsible for the majority of design decisions, I will require that the Python package have:

- an automated test suite with a reasonably high test coverage,
- a comprehensive code review process for all contributed code (using GitHub's pull request mechanism and continuous integration using TravisCI and Coveralls), and
- extensive, high quality documentation using Sphinx.

I will maintain the official project repository[3] and will be the primary gatekeeper (i.e., I will be the one primarily responsible for merging all pull requests). However, I will require that pull requests undergo a high level of review and scrutiny before I will consider merging them. As a class, we will develop a code review process, which will include (among other things) program correctness, test coverage, code readability, and style consistency.

4 Participation

While the entire class will receive one grade for the final project, I will use the participation portion (10%) of your class grade to assess your contribution to the project. I will provide additional details about how this will work, but here are a few things I will expect from everyone:

[3] https://github.com/berkeley-stat222/mouse-lifestyles

1. Attend class. Several of you have had difficulties attending class. This is unacceptable. Moving forward I expect everyone to attend every class. I also expect that you will arrive prior to the start of class. Given the attendance problems in the first half of the class, I will start taking attendance. I will provide additional details in class.

2. Participate in class. Simply occupying space in the classroom is unacceptable. You must pay attention to others. In particular, this means that working on your computer when you should be listening to others or discussing things with your team is prohibited. Participation is also more involved than merely facing others as they speak. You will need to speak in front of the class. You will need to volunteer for tasks. You will need to help refine ideas and contribute to the class' understanding.

3. Contribute to all project aspects. At a minimum every student will be expected to contribute to understanding the problem, developing code, writing tests, finding and fixing bugs, writing and revising documentation, and thinking through the general approach.

This is an open-ended, real-life applied statistics project. If you wait until the last minute each week and try to do as little as possible, you will not succeed. As with anything worthwhile, you will have to work constantly and thoughtfully. You will need to be able to discard work that doesn't lead anywhere and be open to trying new approaches throughout the project.

A core principle throughout the project is that you will be required to convince me that what you did met my expectations. In regard to the participation portion of your grade, this means that you will ultimately be responsible for convincing me that you've participated fully and productively. While details will follow, you will need to submit a self-evaluation at the end of the semester as well as possibly meet with me during finals week to discuss your contributions.

Here are few things to keep in mind while working on your self evaluation:

1. Take responsibility for failures and shortcomings.

2. Do not over embellish.

3. Outline constraints you faced as well as reasons performance was hampered.

4. Include your weakness, but view them as opportunities for improvement.

5. Provide feedback on your experience during the project, course, and program.

6. Stay objective.

7. Demonstrate areas of growth.

8. Highlight skills acquired.

9. Include a discussion of problem-solving abilities you used during the project.

I recommend that you keep notes of your contributions each week to help you prepare your self-evaluation at the end of the semester.

References

[1] David J Anderson and Pietro Perona. Toward a science of computational ethology. *Neuron*, 84(1):18–31, 2014. http://www.sciencedirect.com/science/article/pii/S0896627314007934.

[2] Evan H Goulding, A Katrin Schenk, Punita Juneja, Adrienne W MacKay, Jennifer M Wade, and Laurence H Tecott. A robust automated system elucidates mouse home cage behavioral structure. *Proceedings of the National Academy of Sciences*, 105(52):20575–20582, 2008. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2634928/.

[3] K Jarrod Millman and Fernando Pérez. Developing open-source scientific practice. *Implementing reproducible research. CRC Press, Boca Raton, FL*, pages 149–183, 2014. http://www.jarrodmillman.com/publications/millman2014developing.pdf.

[4] Laurence H Tecott. The genes and brains of mice and men. *American Journal of Psychiatry*, 2003. http://dx.doi.org/10.1176/appi.ajp.160.4.646.

[5] Laurence H Tecott and Eric J Nestler. Neurobehavioral assessment in the information age. *Nature neuroscience*, 7(5):462–466, 2004. http://www.nature.com/neuro/journal/v7/n5/full/nn1225.html.