Contents

Statistics 215A, Fall 2016: Applied Statistics 1

Course Overview 2

Administrativia 3

- Prerequisites 3
- Format and assessment 3
- Texts 4
- Code of conduct; attribution of work 4
- Disability accommodations 5

Main topics by week 5

- Weeks 1-2 5
  - Main topics 5
  - Reading 6
  - Assignment 1 (Due 5 September 2016 by 11:59pm) 7
- Weeks 3-4 7
  - Main topics 7
  - Reading 8
  - Assignment 2 (Due 19 September 2016 by 11:59pm) . . . 8
- Weeks 5-6 10
  - Main topics 10
  - Reading 10
  - Assignment 3 (Due 26 September 2016 by 11:59pm) . . . 10

Statistics 215A, Fall 2016: Applied Statistics

Philip B. Stark, Department of Statistics, UC Berkeley

www.stat.berkeley.edu/~stark pbstark AT b.e. @philipbstark

Office: 403 Evans Hall. Office hours: Thursdays 3:30-4:30

GSI: Yuansi Chen, yuansi.chen AT b.e. Office hours: Tuesday 12-2pm, Friday 11-1pm, 446 Evans Hall

This version: 8 September 2016. Latest version: here

Course Overview

Statistics 215A is a course in applied statistics intended primarily for statistics and biostatistics PhD students. Students are expected to have advanced undergraduate training in probability and statistics, to be reasonably fluent in R or Python, to use version control, and to have some familiarity with unit testing.

The course consists primarily of *doing* applied statistics, rather than *taking about how to do* applied statistics.

Students are expected to have the motivation and intellectual maturity to acquire any technical skills and tools they might lack to succeed in the course–without help. In a nutshell, that's what you have to be able to do to be a professional. Below, there are pointers to resources that motivated students can use to remedy deficiencies.

The course focuses on scientific questions and issues that can be addressed using statistics, and in particular on how to think statistically about applied problems. The instruction will emphasize transparent, reproducible, and open research, trying to cultivate good "hygiene" and habits.

This is *not* a course on statistical methodology. The basic approach to applied statistics in this class is:

1. Understand the scientific problem
2. Understand the nature and limitations of the observations that have been made or could be made
3. Think hard about how the (potential) observations relate to the scientific questions of interest
4. Think hard about whether statistics and the data can answer the interesting scientific questions
5. Design data collection methods or use existing data to learn about the world from data. This might involve:

    a. Visualization
    b. Exploratory data analysis
    c. Formulating an appropriate statistical model
    d. Hypothesis testing
    e. Parameter estimation
    f. Uncertainty quantification

6. Document the process as it unfolds
7. Reflect on the meaning of the results
8. Communicate the results
9. Circle back to integrate the new evidence to inform the next set of questions, experiments, etc.

All these steps should be done as transparently and reproducibly as possible, using appropriate tools.

Administrativia

Prerequisites

- At least one semester of upper-division probability
- At least three semesters of upper-division statistics
- Linear algebra and calculus
- Proficiency in R and/or Python, basic knowledge of Git, basic knowledge of unit testing, basic knowledge of Jupyter notebooks, basic knowledge of Markdown
    - If you need help, see:
      http://statistics.berkeley.edu/computing/training/tutorials
      http://statistics.berkeley.edu/computing/r-bootcamp
      https://www.python.org/about/gettingstarted/
      https://youtu.be/XFw1JVXKJss
      https://sites.google.com/site/pythonbootcamp/
      http://jupyter.org/
      http://fperez.org/py4science/git.html
      https://try.github.io/levels/1/challenges/1
      https://git-scm.com/book/en/v2/Getting-Started-Git-Basics
      http://software-carpentry.org/lessons/
      http://doc.pytest.org/en/latest/
      http://docs.python-guide.org/en/latest/writing/tests/
      https://daringfireball.net/projects/markdown/
    - All assignments will require GitHub
    - Most assignments require Jupyter and Markdown
    - Some assignments will require Python; others allow a choice between R and Python

Format and assessment

- two 90-minute lectures and one 2-hour lab per week
- written/computational assignments roughly every two weeks (30% of grade)

- two term projects done in teams of 3-4 students:
  - a refereeing project or reproduction project for an open journal (30% of grade)
  - a data-based project (40% of grade, in lieu of a final exam) At the moment, there are two science projects I'm considering:
    - earthquake early warning systems using cellphone accelerometers http://seismo.berkeley.edu/research/early_warning.html http://www.latimes.com/local/lanow/la-me-ln-app-mobile-phone-detect-earthquakes-20160212-story.html
    - the LIGO detection of gravitational waves https://www.ligo.caltech.edu/page/detection-companion-papers https://github.com/minrk/ligo-binder https://losc.ligo.org/s/events/GW150914/GW150914_tutorial.html

Details of the assignments and term projects will be announced during the term.

Submitting assignments: Assignments **must** be submitted through GitHub (links to the assignments on GitHub will be circulated using bcourses). Weekly assignments and the data-based term project should be submitted as Jupyter notebooks, with R or Python as the programming language. The refereeing / reproduction assignment can combine a Jupyter notebook and separate PDF, HTML, RTF, or MarkDown document, but *not* Microsoft Word. All code submitted as part of the term project should be version controlled and should include unit tests.

Texts

- Freedman, D.A., 2009. *Statistical Models: Theory and Practice*, Cambridge University Press, ISBN 978-0521743853

- Freedman, D.A., 2009. *Statistical Models and Causal Inference: A Dialogue with the Social Sciences*, Cambridge University Press, ISBN 978-0521123907

Code of conduct; attribution of work

The high academic standard at the University of California, Berkeley, is reflected in each degree awarded. Every student is expected to maintain this high standard by ensuring that all academic work reflects unique ideas or properly attributes the ideas to the original sources.

These are some basic expectations of students with regards to academic integrity: Any work submitted should be your own individual thoughts, and should not have been submitted for credit in another course unless you have prior written permission to re-use it in this course from this instructor.

All assignments must use “proper attribution,” meaning that you have identified the original source and extent or words or ideas that you reproduce or use in your assignment. This includes drafts and homework assignments! If you are unclear about expectations, ask your instructor.

Do not collaborate or work with other students on assignments or projects unless the instructor gives you permission or instruction to do so.

Disability accommodations

If you need an accommodation for a disability, if you have information your wish to share with the instructor about a medical emergency, or if you need special arrangements if the building needs to be evacuated, please inform the instructor as soon as possible.

If you are not currently listed with DSP (the Disabled Students’ Program) and believe you might benefit from their support, please apply online at dsp.berkeley.edu

Main topics by week

Weeks 1-2

Main topics

1. Course overview; student and instructor hopes and expectations
2. Two sample papers:

    a. DeCellesa, K.A., and M.I. Norton, 2016. Physical and situational inequality on airplanes predicts air rage, *PNAS*, www.pnas.org/cgi/doi/10.1073/pnas.1521727113

    b. Urban, M.C., 2015. Accelerating Extinction Risk from Climate Change, *Science*, *348*, 571-573 (and supporting materials) http://www.sciencemagazinedigital.org/sciencemagazine/01_may_2015?folio=571#pg113

3. Brief discussion of term projects: scientific questions in earthquake early warning and LIGO
4. Probability and ontological commitments in applied problems

    a. Theories of probability

    b. Probability by fiat–randomization

    c. Probability from “physics”

    d. Probability as compact description

    e. Probability and prediction

    f. Probability as metaphor

    g. Probability models I. Calibration vs. testing

    h. Causal inference: response schedules

        i. Statistical models as empirical commitments

5. Numerics and computing
    a. Floating point arithmetic
    b. Condition number
        I. Solving linear systems of equations

Reading

1. DeCellesa, K.A., and M.I. Norton, 2016. Physical and situational inequality on airplanes predicts air rage, *PNAS*,
www.pnas.org/cgi/doi/10.1073/pnas.1521727113

2. Fryer, R.G., 2016. An Empirical Analysis of Racial Differences in Police Use of Force, Working Paper 22399, National Bureau Of Economic Research,
http://www.nber.org/papers/w22399
Press: http://www.nytimes.com/2016/07/12/upshot/surprising-new-evidence-shows-bias-in-police-use-of-force-but-not-in-shootings.html

3. Urban, M.C., 2015. Accelerating Extinction Risk from Climate Change, *Science*, *348*, 571-573 (and supporting materials)
http://www.sciencemagazinedigital.org/sciencemagazine/01_may_2015?folio=571#pg113

4. Pitt, J. and H.Z. Hill, 2016. Statistical analysis of numerical preclinical radiobiological data, *ScienceOpen Research*,
DOI: 10.14293/S2199-1006.1.SOR-STAT.AFHTWC.v1

5. Freedman, D.A., 2009. *Statistical Models: Theory and Practice* (SMTP), Chapters 1-4, 6, 7.

6. Freedman, D.A., 2009. *Statistical Models and Causal Inference: A Dialogue with the Social Sciences* (SMCI), Chapters 1, 2, 8.

7. Introduction to unit testing using py.test: https://youtu.be/P-AhpukDIik
https://youtu.be/FxSsnHeWQBY (Ned Batchelder: Getting Started Testing - PyCon 2014)

8. Koren, J.R., 2016. Feds use Rand formula to spot discrimination. The GOP calls it junk science. Los Angeles Times, 28 August 2016.
http://www.latimes.com/business/la-fi-rand-elliott-20160824-snap-story.html
http://www.rand.org/content/dam/rand/pubs/research_reports/RR1100/RR1162/RAND_RR1162.pdf
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2653886/

9. Allen, J.G., P. MacNaughton, U. Satish, S. Santanam, J. Vallarino, and J.D. Spengler, 2016. Associations of Cognitive Function Scores with Carbon

Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments, *Environ Health Perspectives*; DOI:10.1289/ehp.1510037 http://ehp.niehs.nih.gov/15-10037/

10. Goldberg, D., 1991. What every computer scientist should know about floating-point arithmetic, *ACM Computing Surveys (CSUR)*, *23*, 5–48. DOI: 10.1145/103162.103163.
(Reprint: https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html)

Assignment 1 (Due 5 September 2016 by 11:59pm)

1. If you do not have a GitHub account, get one. Request an educational account, so you can have private repositories at no cost.
2. Fill out this form to register your GitHub user name for the class.
3. Hand in the following using classroom.github.com, via the link sent for the first assignment:
   a. Freedman, SMTP, problems 4.B.7, 4.B.8, 4.B.11, 4.5.3, 4.5.5, 4.5.6, 4.5.10, 4.5.11.
   b. Give an example where floating point addition is not associative.
   c. Explain the Kahan Summation Formula in your own words, and explain why it is important.
   d. Explain why the condition number of a linear system matters.
   e. Suppose that A is a positive definite square matrix. Prove algebraically that solving the linear system $Ax = y$ by calculating $A^{-1}y$ squares the condition number.
   f. Under what circumstances does it make sense to invert a matrix numerically rather than solve a linear system by factorization, Gaussian elimination, back-substitution, or a similar method?

Weeks 3-4

Main topics

1. Hypothesis tests and p-values
   a. What's a p-value?
   b. ASA statement on p-values, and comments
   c. Cargo-cult confidence
   d. The two-sample problem
   e. Two-sample t-test, parametric and permutation versions
   f. Logistic regression

   g. Multiplicity
      i. Per-comparison error rate (PCER)
      ii. Familywise error rate (FWER)
      iii. False discovery rate (FDR)

2. Reproducibility

   a. Terminology: reproducibility, replicability, repeatability, ...
   b. Why?
   c. Reproducibility as hygiene, lab technique, *mise en place*
   d. Failure stories: http://eusprig.org/
   e. Workflows and tools: revision control, issue trackers, unit tests, coverage

3. Bayesian and frequentist approaches to inference

Reading

1. ASA statement on *p*-values, and supplementary material/comments, 2016. *The American Statistician*, *70*, http://amstat.tandfonline.com/doi/full/10.1080/00031305.2016.1154108

2. Stark, P.B., 2016. *Nullius in Verba*, http://www.stat.berkeley.edu/~stark/Preprints/nulliusInVerba16.pdf

3. Stark, P.B., 2016. Pay no attention to the model behind the curtain. http://www.stat.berkeley.edu/~stark/Preprints/eucCurtain15.pdf

4. Stark, P.B., 2015. Constraints versus priors. *SIAM/ASA Journal on Uncertainty Quantification*, *3* (1), 586–598. doi:10.1137/130920721, Reprint: http://epubs.siam.org/doi/10.1137/130920721, Preprint: http://www.stat.berkeley.edu/~stark/Preprints/constraintsPriors15.pdf

5. Stark, P.B., 1997. *SticiGui* Chapter on Hypothesis Testing: http://www.stat.berkeley.edu/~stark/SticiGui/Text/testing.htm

6. Feynman, R.P., 1974. Cargo Cult Science, *Engineering and Science*, *37*, 10–13. http://calteches.library.caltech.edu/51/2/CargoCult.htm

Assignment 2 (Due 19 September 2016 by 11:59pm)

1. True or false, and explain: A *p*-value

   a. is the chance the null hypothesis is true
   b. is the probability of observing the data by chance alone
   c. is computed by assuming that the null hypothesis is true

d. is the conditional probability that the null hypothesis is true, given the data
e. is the chance that the experiment will give the same results in a new trial
f. depends on the alternative hypothesis

2. True or false, and explain: If the p-value is small,

   a. the null hypothesis is false
   b. the null hypothesis is probably false
   c. the alternative hypothesis is true
   d. the alternative hypothesis is probably true
   e. there is an effect
   f. there is not an effect
   g. the effect is large enough to matter
   h. the effect is probably large enough to matter

3. Explain the difference between p-values, significance levels, confidence levels, and power.

4. We propose to test whether a coin is fair using the number of times it lands "heads" in 100 independent tosses. The alternative hypothesis is that the chance the coin lands heads is greater than 50%.

   a. To test at significance level $\alpha = 0.05$, when should we reject the null hypothesis?
   b. Using the previous answer as the threshold for the test, what is the power of the test against the alternative hypothesis that the chance the coin lands heads is 52%?

5. Implement from scratch in Python a two-sample permutation test using the mean as the test statistic. The test should have options for the alternative: left, right, two-sided. It should estimate the p-value by simulation; the number of replications should be an argument. Include unit tests of your code. One unit test should be based on an analytical calculation using the hypergeometric distribution.

6. What is the null hypothesis of the test? What alternatives would you expect it to have the most power against?

7. What is the null hypothesis of the standard t-test? What alternatives is it designed to have the most power against?

8. Give an example of a population of size $N = 10$ for which, when the population is split randomly into two pieces of size $n = m = 5$, the nominal p-value of the parametric t-test will be systematically too small. Use 10^4 iterations in your simulation. Explain heuristically why the level of the parametric test is wrong.

9. Freedman, SMTP, problems 7.B.2, 7.B.3, 7.C.5, 7.D.7, 7.E.2, 7.E.3, 7.E.10, 7.5.2, 7.5.3, 7.5.4, 7.5.5

Weeks 5-6

Main topics

1. Random sampling
    a. Random number generation
    b. Sampling algorithms
2. TBA

Reading

1. https://en.wikipedia.org/wiki/Pseudorandom_number_generator

2. L'Ecuyer, P. and R. Simard, 2007. TestU01: A C library for empirical testing of random number generators, *ACM Transactions on Mathematical Software (TOMS)*, *33*, DOI: 10.1145/1268776.1268777 http://dl.acm.org/citation.cfm?doid=1268776.1268777

3. O'Neill, M.E., 2015. PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation, http://www.pcg-random.org/pdf/toms-oneill-pcg-family-v1.02.pdf

4. SHA-2. https://en.wikipedia.org/wiki/SHA-2

5. random.py source code: https://hg.python.org/cpython/file/2.7/Lib/random.py

6. Eisen, M., 2016. Coupling Pre-Prints and Post-Publication Peer Review for Fast, Cheap, Fair, and Effective Science Publishing. http://www.michaeleisen.org/blog/?p=1820

7. Appendix D of the Commission on the Future of the UC Berkeley Library report (pp29–33), http://evcp.berkeley.edu/sites/default/files/Commission%20on%20Future%20of%20UC%20Berkeley%20Li

Assignment 3 (Due 26 September 2016 by 11:59pm)

1. Hand in the following using classroom.github.com, via the link sent for the second assignment:

a. Implement an object-oriented PRNG based on the SHA-256 hash of a seed concatenated with a sequence number, in Python, by subclassing the class Random and overwriting the methods random(), seed(), getstate(), setstate(), jumpahead(), and getrandbits(). Your implementation should inherit shuffle(), choice(), sample(), randbelow(), etc., from the parent Random class.

b. Implement tests of your class methods using nose (https://nose.readthedocs.io/en/latest/) or pytest (http://doc.pytest.org/en/latest/). Implement tests of:

  - setting the seed
  - setting the state
  - getting the state
  - jumping ahead
  - at least two tests for uniformity, with "reasonable" power to detect a bug, including a test using the Kolmogorov-Smirnov statistic for random() and a binomial test for the single-bit frequency for getrandbits().
  - tests that your class inherited the expected methods from the parent class Run your tests on the default Python PRNG (based on the Mersenne Twister) as well as your PRNG.

c. Use the Jupyter "magic" %timeit command to compare the amount of time it takes to generate 10^7 PRNs using your SHA-256 method and using the default Python PRNG.