

## Contents

<b>Syllabus for Statistics 240: Nonparametric and Robust Statistics</b>	<b>1</b>
Philip B. Stark Department of Statistics, UC Berkeley . . . . .	1
This version: January 11, 2023 . . . . .	1
Overview . . . . .	1
Administrativia . . . . .	1
Format and assessment . . . . .	1
Office hours . . . . .	2
Communication . . . . .	2
Grading . . . . .	2
Code of conduct; attribution of work . . . . .	2
Disability accommodations . . . . .	3
Resources . . . . .	3

## Syllabus for Statistics 240: Nonparametric and Robust Statistics

**Philip B. Stark Department of Statistics, UC Berkeley**

**This version: January 11, 2023**

### Overview

This course will focus on nonparametric/exact/conservative methods for inference in a variety of settings, including survey sampling, auditing, and randomized experiments. The emphasis will be on recent theory involving supermartingales.

Technical topics include:

- Nonparametric inference about the treatment effect in randomized, controlled trials
- Nonparametric inference about population means for a variety of sampling designs
- Sequential testing and supermartingales
- Combining hypothesis tests, nonparametric combinations of tests,  $E$ -values
- Nonparametric methods based on ranks
- Density estimation
- Pseudo-random number generation, simulation

Consistent class participation is crucial: we will be discussing subtle substantive, technical, and philosophical issues and reviewing code during class.

### Administrativia

#### Format and assessment

- 3 hours of lecture per week

- approximately 4 written assignments, a term project involving contributing to an open-source python library, and a term project involving data

### **Office hours**

- Wednesdays, 11AM-12PM (Google Meet) or by appointment

### **Communication**

Please use the course Slack channels ([https://join.slack.com/t/slack-iv99638/shared\\_invite/zt-1n1ypzr8l-U7Q8YRIkb6gl7xS9xH8xaw](https://join.slack.com/t/slack-iv99638/shared_invite/zt-1n1ypzr8l-U7Q8YRIkb6gl7xS9xH8xaw)) for questions/comments/discussion about course material and logistics. For personal matters (illness, accommodations, etc.) that should remain private, please send email.

During the work week, I expect to be able to reply to Slack messages and email within 24 hours. On weekends, I might need longer.

### **Grading**

*Submitting assignments:* Submit written assignments by making a pull request to your private repository within the Berkeley GitHub organization for the class, <https://github.berkeley.edu/stat-240-s23>

Use your CalNet credentials to access your private repository. Create a directory for each assignment labeled with the assignment number, e.g., “Assignment1” for the first assignment.

- Text documents should be written in LaTeX or Markdown (Markdown, processed by pandoc, is preferred: using Markdown can really speed up your writing and your source will be much easier to read). A pdf and the source file should be submitted. Microsoft Word is not acceptable.
- Code and analyses should be in python. All code should have accompanying unit tests. Follow PEP8 and PEP257. In some cases, Jupyter notebooks will be the appropriate thing to submit; in others (more extensive analyses), a notebook and a collection of .py files might be more appropriate. For term projects, the “deliverable” will include a repository that includes code, data, analyses, and unit tests. The adequacy and coverage of the unit tests will be assessed.
- Final projects are due on the first day of final exams, 8 May 2023. Final projects might include an in-person or recorded video presentation (to be determined).

### **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 of 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 you 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 <https://dsp.berkeley.edu/>.

### **Resources**

- Communication: class Slack channel [https://join.slack.com/t/slack-iv99638/shared\\_invite/zt-1n1ypzr8l-U7Q8YRIkb6gl7xS9xH8xaw](https://join.slack.com/t/slack-iv99638/shared_invite/zt-1n1ypzr8l-U7Q8YRIkb6gl7xS9xH8xaw)
- Computing resources
  - We will use Jupyter notebooks. We will start with hosted notebooks on <https://datahub.berkeley.edu/>, but you can also install everything on your own device.
  - We will use the campus github server, [github.berkeley.edu](https://github.berkeley.edu).
  - The class notes and most other materials are available at <https://github.com/pbstark/StatNotes>.
  - Assignments should be submitted by pull request to your private repository within the class organization <https://github.berkeley.edu/stat-240-s23>.
- Git and git workflows
  - Introduction to Git.
  - Immersion course
  - git-scm guide
  - Statlab development git workflow

- Continuous integration
  - GitHub actions
- Scientific Python, Jupyter
  - Lecture notes on scientific python
  - Python for scientific computing by Fernando Perez
  - <https://hplgit.github.io/primer.html/doc/pub/half/book.pdf>
  - Elegant SciPy, Stefan van der Walt. The full book and all the notebooks are available.
  - Getting started with Python for research, a gentle introduction to Python in data-intensive research.
  - An introduction to “Data Science”, a collection of Notebooks by BIDS’ Stéfan Van der Walt.
  - Effective Computation in Physics, by Kathryn D. Huff; Anthony Scopatz. Notebooks to accompany the book.
  - A Whirlwind Tour of Python, by Jake VanderPlas.
  - Python for Data Analysis, 2nd Edition, by Wes McKinney, creator of Pandas. Companion Notebooks
  - Effective Pandas, a book by Tom Augspurger, core Pandas developer.
- Docker
  - <https://docs.docker.com/get-started/>
  - <https://docker-curriculum.com/>
- LaTeX
  - <https://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf>
- Markdown
  - <https://daringfireball.net/projects/markdown/syntax>
  - <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
  - <https://www.markdownguide.org/getting-started/>
- Pandoc
  - <https://pandoc.org/getting-started.html>
  - <https://pandoc.org/MANUAL.pdf>
- Miscellaneous computing tutorials
  - Berkeley Statistical Computing Facility tutorials