Course Information

D. Jason Koskinen
koskinen@nbi.ku.dk

Advanced Methods in Applied Statistics
Feb - Apr 2021

Photo by Howard Jackman

Times & Locations

From 08:45-09:00 I will try to be available in ZOOM,
but I will only be there for discussion and
student Q&A, i.e. no new material until 09:00

- Hours
  - Course is in Block A
  - Tuesday 08:00 - 12:00
  - Thursday 08:00 - 12:00 and 13:00 - 17:00
- Location:
  - Due to COVID-19 restrictions, all classes will be conducted via ZOOM
  - I will try to record the ZOOM lectures and have them posted to Absalon
- "In-class" Activities
  - ~20-30% of the time will be lecture
  - **Vast majority** of the time will be practical exercises
    - Finding appropriate software package or function
    - Properly instantiating the relevant statistical method
    - Debugging
    - Documentation, plotting, code clean-up, maybe even in-line comments, etc.

Actual
08:45-09:00 Study/Q&A
09:01 Lecture

Teachers

- I go by “Jason”
- My scientific focus is on experimental neutrino oscillation, where I work on the IceCube neutrino observatory situated at the South Pole

- Teaching Assistant is Tetiana “Tania” Kozynets
- Ph.D. student in neutrino physics and astrophysics

Computers & Software

- Everyone should have a computer that they can install software on and use for this course

- Software specifics are at the preference of the student(s)
  - Suggestions are python, R, MATLAB, or C/C++/C18
    - The more common the language the more likely you can use the internet and fellow students (possibly) for help
  - Lectures and examples will be mostly in python using some external packages:
    - SciPy (http://www.scipy.org)
    - NumPy (http://www.numpy.org)

Software, Checklist, & Skills

- Have an installed text editor for writing/editing software
- Have some package for the production of plots and diagrams (matplotlib, R, gnuplot, MATLAB, etc.)
  - See backup slides for some more specific software packages
- I strongly recommend reviewing the undergraduate stats course (http://www.nbi.dk/~petersen/Teaching/AppliedStatistics2020.html)
  - Actually do the exercises, not just scan the material, and see what isn't clear or familiar

Course Material

- **NO** required text or textbooks. I will cover many topics w/ in-class lectures and all the notes will be posted online. But, this may be insufficient in depth or explanation for your personal preference, so students are encouraged to use…
  - The Internet - probably the best source for information and help.
  - "Statistical Data Analysis" by Glen Cowan
  - "Modern Statistical Methods for Astronomy" by Feigelson & Babu
  - Journal articles
    - Any that you might find relevant
    - Some posted by me

Optional Group/Class Communication

- Tania has set up a AMAS 2021 Slack channel which is totally optional
  - See Announcements in Absalon for more info
- It is an option to discuss topics, collaborate w/ group members on projects, exchange coding breakthroughs and software solutions, comment on Jason's ZOOM background, etc.

Student Assessment

- Presentation and 2-page summary (10%)
  - Take topic and/or relevant article for presentation to the class
  - Can be done in groups (see Absalon for full info)
- Graded problem sets (20%)
  - Can be done in groups of any size, but must be submitted individually
- Project (30%)
  - Larger data analysis project, nominally related to your field of research
- Final Exam (40%)

Assessment Scale

- Material is marked on a 10-point scale
  - 9+ is very good
  - 8-9 is pretty good
  - 7-8 is okay
  - 6-7 is acceptable
  - 5-6 subpar
  - 4-5 inadequate
  - <4 reflects serious omissions and/or deficiencies

Problem Sets

- The submission is:
  - A write-up as a PDF document, which includes any plots, diagrams, tables, pictures, and explanations
  - In a separate "file", submit all code used to derive the results
    - Tarball, zipped directory, lots of individual files w/ self-explanatory titles, etc.
    - Include original data files if possible

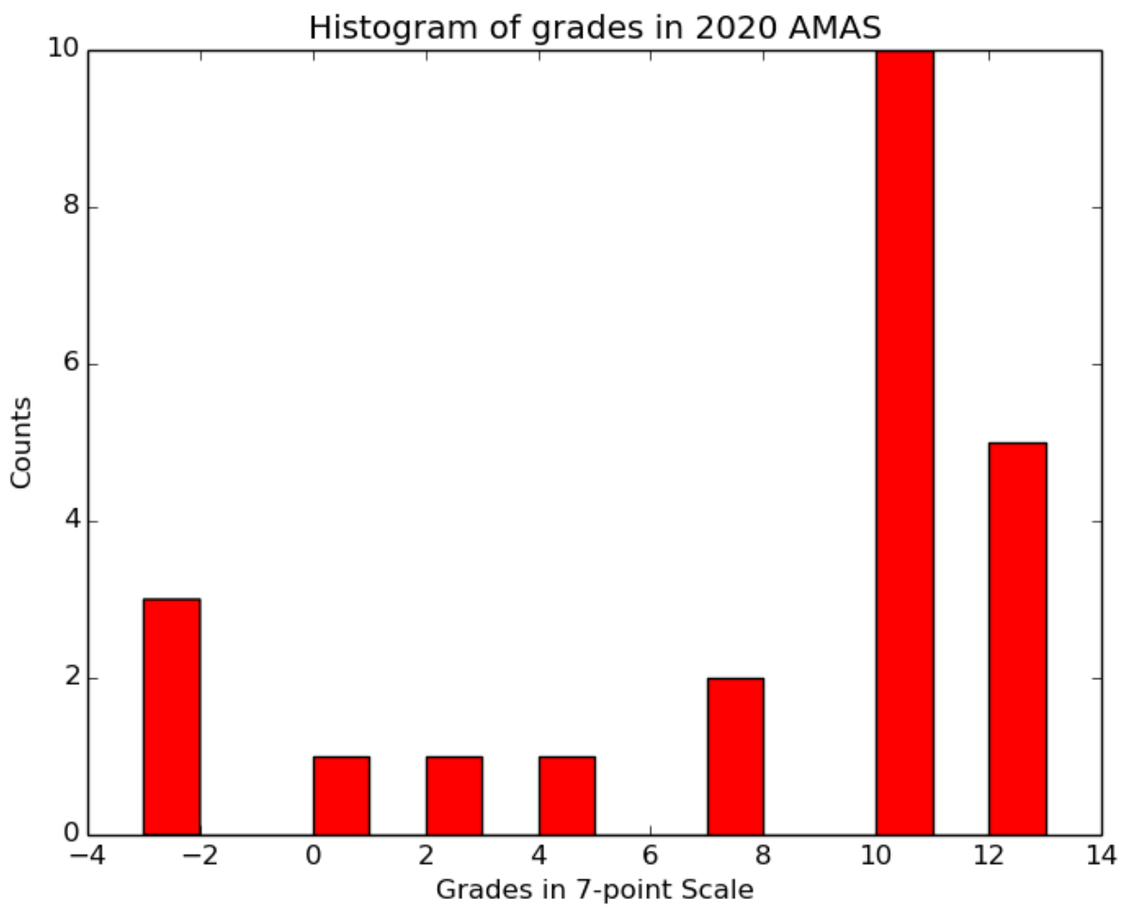
- Late submission
  - -10% for 0-1 hours late
  - -25% for 1-12 hours late
  - -50% for 12-24 hours late
  - -100% for 24+ hours late
  - For extenuating circumstances contact Jason

Final Exam

- 1-day (~28 hour) take home test
- Requires computers, writing/modifying code that has been developed during the course
- **You must work on your own!!!**
  - Along with the answers, the code producing the results must also be submitted
- Doing the in-class exercises and homework is excellent preparation for the exam

Grading Breakdown

- The final course grade is converted to the Danish scale, but all course material uses a 10-point scale.
- The conversion will be ‘roughly’ decile, e.g. a 9.32 final weighted average is very, very likely to get a mark of “12”.
  - In 2017, 9.20 was the cutoff for a “12”, and 8.2 for a “10”.
  - In 2018, 8.75 was the cutoff for a “12”, and 7.7 for a “10”.
  - In 2019, 9.15 was the cutoff for a “12”, and 8.3 for a “10”.
  - In 2020, 9.00 was the cutoff for a “12”, and 8.0 for a “10”.



Student Assessment

- All assessment material will be graded based on the results
  - Code can be sloppy and inefficient and it is unlikely to impact your grade. The exception is where it would be good to give the 'benefit of the doubt', but the grader can't decipher your code.
- I encourage you to share solutions, efficient code, elegant solutions, etc. for everything other than the final exam
  - If you use a portion of someone else's code (which is 100% acceptable) make in-line acknowledgement in the comments of your code
  - Beware, that if your code starts to look like a collection of only other people's code, it's unlikely that the Final Exam will go well

Challenges

- Multiple student backgrounds and multiple topics mean that some students may feel like they would benefit from more challenging material… have no fear ;-)
- I have collected some projects/questions from colleagues
- Potentially pick something on your own and discuss it with me, maybe even put together some lecture material and add it to the course

For the Proficient

- Some people will have excellent software/coding skills and will be able to quickly complete many of the in-class exercises. For those who consistently find themselves in this situation I offer an opportunity.

- The later problem set(s) will be very similar to exercises completed in class. I will offer extra credit for completing the problems using a different device. Playstation, mobile phone, dedicated GPU machine, etc.

Expectations

- As graduate students, there is a rapidly growing importance for self-directed learning
- Software and hardware difficulties and solutions are the sole domain of each student. You can do all the projects on a PlayStation 4 w/ screenshots
- These are nominally advanced topics
  - I am excited to discuss new/other topics to cover in the course
  - We won't always have unassailable experts. So, discussion, and participation by individuals and groups is important

Questions?

Backup

Software Packages

- Some of the methods we will use in the course will require software packages that include:
  - Minimizers: for example BFGS, MIGRAD, SIMPLEX, etc.
  - Markov Chain Monte Carlo
  - Spline routines for interpolation, including basis splines (b-splines)
  - Multi-Variate Machine Learning: boosted decision trees, neural networks, support vector machines, etc. (we will for sure cover boosted decision trees)
- Other more specialized uses I will let you know about in advance of the lecture
  - Wavelets analysis needs deconvolution/decomposition methods and/or libraries
  - MultiNest nested sampling algorithm

More Specifically

- Below I will list the needed packages and some python options
- Plotting
  - I use mostly Matplotlib
- For Python users, I'm a big fan of "Jupyter" notebooks
  - Combination of both text fields, inline figures/plots display, and executable code
  - Great way to keep things organized
- Minimizer Routines
  - I normally use MINUIT2 (via iminuit)
  - SciPy has a minimize function with a bunch of algorithms and is more common nowadays

More Specifically

- Markov Chain Monte Carlo
  - I have used PyMC, but other packages such as MCMC, emcee, or Nestle look like better tools
- Multi-Variate Analysis (MVA)
  - XGBoost, CatBoost
- Splines
  - SciPy has an interpolate function and other spline options
- Bayesian Inference Sampling - MultiNest
  - Nestle
- Even if you're using python, you don't **need** any of the above mentioned *specific* packages, e.g. iminuit.

My Laptop

- As of Feb. 1, 2021 my laptop was setup as:
  - Mac OS Catalina (10.15.7)
  - Python 3.7.3
    - iPython 7.19.0
    - SciPy 1.5.4
    - NumPy 1.19.4
    - jupyter notebook 6.1.5
  - Pip3 (python package manager) 20.2.4