Course Information

D. Jason Koskinen

koskinen@nbi.ku.dk
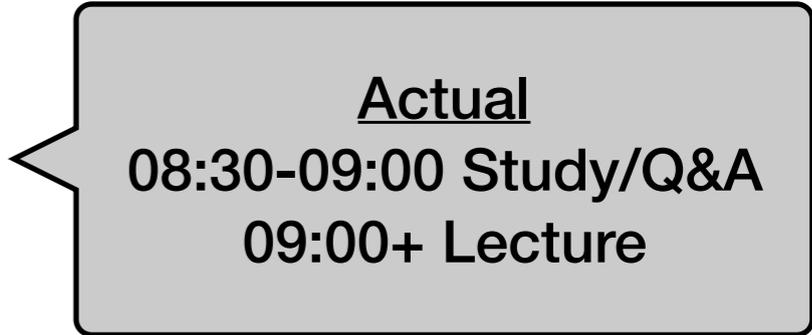
Advanced Methods in Applied Statistics
Feb - Apr 2019

Photo by Howard Jackman

University of Copenhagen   Niels Bohr Institute

Times & Locations

From 08:30-09:00 I will try to be in the appropriate Auditorium
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

<u>Actual</u>
08:30-09:00 Study/Q&A
09:00+ Lecture

- Locations:

  - Tuesday is at øv - bib 4-0-17, Universitetsparken 1-3, DIKU
  - Thursday morning is at Aud 10, Universitetsparken 5, HCØ
  - Thursday afternoon is at Aud 06, Universitetsparken 5, HCØ

- In-class Activities

  - ~20-30% of the time will be lecture
  - <u>**Vast majority**</u> 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.

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 Jean-Loup Tastet
- Ph.D. student in neutrino phenomenology

Computers & Software

- Everyone should have a laptop

- Software specifics are at the preference of the student(s)
  - Suggestions are python, R, MATLAB, or C/C++
    - 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:
    - ROOT (https://root.cern.ch), note that ROOT is *not necessary* and can be painful to install and is therefore not encouraged
    - 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 (ROOT, 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/AppliedStatistics2018.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 <u>encouraged</u> 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

Student Assessment

- Oral Presentation and 2-page summary (10%)

  - Take topic and/or relevant article for presentation to the class

  - Can be done in groups, but no more than 3 people

- 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

  - Can be done in groups (no more than 3 people) with a single 4-6 page written report

- 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

- There will be 2 problem sets to be handed in

- 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

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".

AMAS Grades (2018)
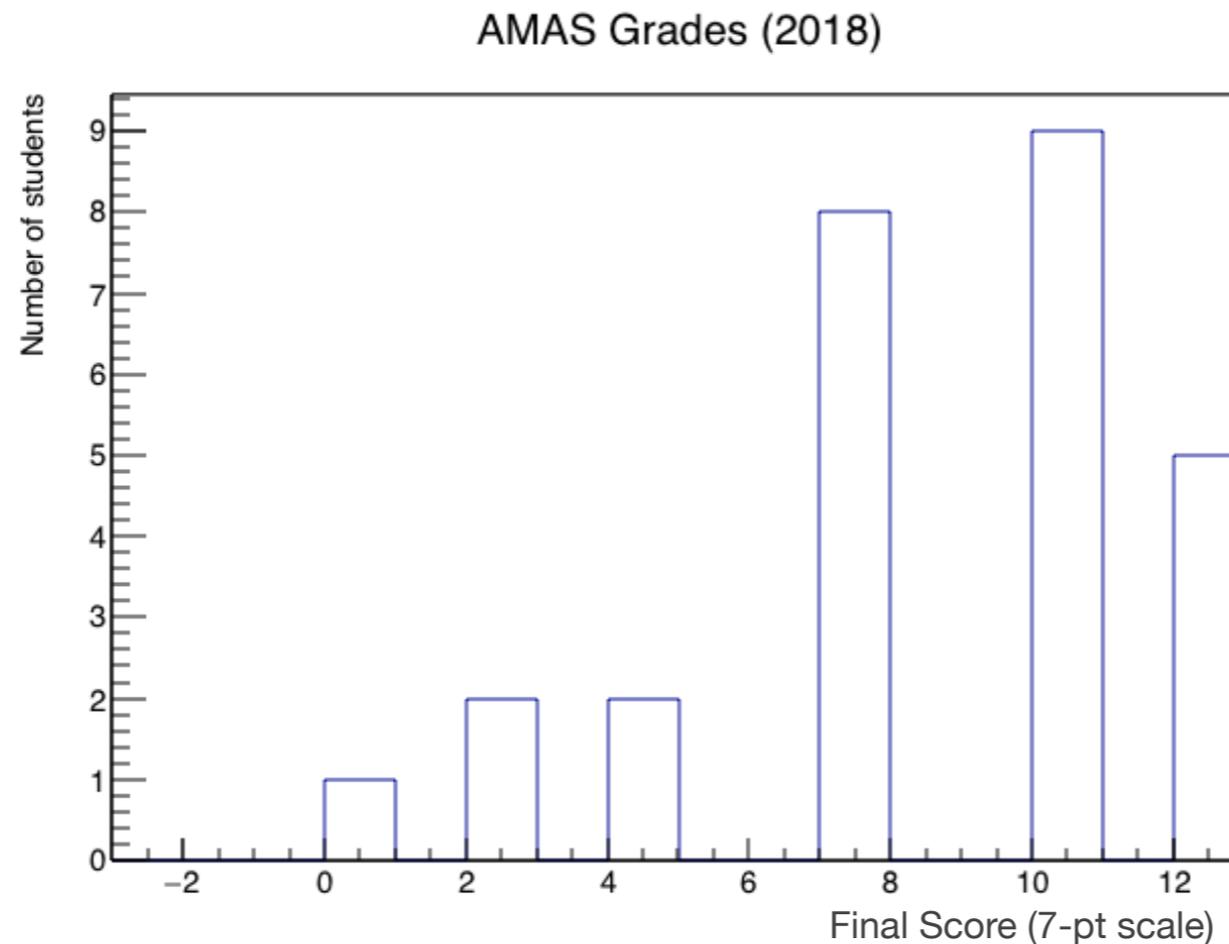
Student Assessment

- 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 I'd like to give the 'benefit of the doubt', but 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 2nd problem set — which is not yet assigned — 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 <u>sole domain</u> 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

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 previously used ROOT, but have switched to 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)

  - I used the ROOT software from CERN

- Splines

  - SciPy has an interpolate function and other spline options

- Bayesian Inference Sampling - MultiNest

  - pymultinest

- Even if you're using python, you don't **need** any of the above mentioned *specific* packages, e.g. iminuit.

My Laptop

- As of Jan. 7, 2019 my laptop was setup as:
  - Mac OS Sierra (10.13.2)
  - Python 2.7.10
    - iPython 5.1.0
    - SciPy 1.0.0
    - NumPy 1.15.1
    - jupyter 4.3.1
  - Homebrew (Mac package manager) 1.7.1
    - homebrew/core
    - homebrew/science
  - pip (python package manager) 18.0