Advanced Data Analysis in Python

Koç University

Spring 2020

Syllabus

Instructor

David Carlson

dcarlson@ku.edu.tr

Office: CASE 140

TA

M. Yavuz Yağış

Class Schedule

MON, WED 10.00 — 11.15

Office Hours

MON 11.30 – 12.30; TUES 16.30 – 17.30 (or by appointment)

Online Access

https://ku.blackboard.com (for grades and readings)

https://github.com/carlson9/KocPython2020 (for in-class material)

Introduction

This course, broadly speaking, is designed to familiarize the student with Python 3 and advanced data analysis techniques. We will cover core programming concepts using Python, which apply to programming more generally. These include syntax, data types, functions, loops, recursion, and classes and inheritance. We will then cover data base management, creation, manipulation, and visualization. A brief overview of Bayesian statistics with an emphasis on practical use in the Stan programming language called through Python will be followed by introductions to the most common machine learning methods. This is a demanding course, with the ultimate goal a final project with an original analysis testing one or several hypotheses.

No previous programming experience is assumed. However, a good understanding of linear models is required. Undergraduate prerequisite: MATH 202 (or equivalent); graduate prerequisite: INTL 601 (or equivalent).

Suggested Books

This course contains suggested readings and exercises from two books. Electronic versions of both of these books can be purchased, but can be read for free online. I do not necessarily recommend purchasing them, and I do not recommend purchasing hard versions. As you learn to code, you should also learn how to use online resources that are freely available. Further, you will need to follow along with the readings while coding in Python, making online and soft versions more sensible. In addition to the book readings, assigned articles will be posted on Blackboard, and links for online material are available in the course schedule. The two books, and the associated websites, are:

1) Shaw, Zed A. 2017. *Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code (Zed Shaw's Hard Way Series).* 1st Edition. Addison-Wesley. Available at: https://learnpythonthehardway.org/python3/

2) VanderPlas, Jake. 2016. *Python Data Science Handbook: Essential Tools for Working with Data.* O'Reilly Media. Available at: https://jakevdp.github.io/PythonDataScienceHandbook/

Requirements and Grading

Grades will not be rounded, these represent strict cut-offs. In the rare event of, for example, exactly a 90, the higher grade will be assigned. Pluses and minuses will be applied at the instructor's discretion and will only be used if there are clear separations within a given grade. **Note that the Koç suggested grades are not followed in this course.**

A	90–100
B	80–90
C	70–80
D	60–70
F	<60

1) *Homework: 50%*

Both graded and ungraded homework assignments will be assigned throughout the semester. The ungraded assignments will not be checked, but it is essential to complete them for success in the course and to adequately learn the material. Four graded assignments will be checked. They will be posted on Blackboard at least one week before they are due. It is strongly encouraged that you start the homeworks as early as possible. As we will discuss in the first week, all work must be done on git. Working in collaboration with one another is encouraged, but every keystroke must be your own. That is, you can work together, but you must complete your own assignment for grading and review. Late work will not be accepted. The work must be completed on git before the class meets for the week of the due date.

2) *Final project: 50%*

The final project can be done in groups of two to four students or individually, but the grading will reflect this. More will be expected from groups than from individuals. The final project involves both coding and writing of results. To be more precise, the final project turned in should include a report appropriate for your field or work, and all code, well-organized and commented, for replication. The report should at a minimum explain the hypothesis/hypotheses, discuss in detail the data that is analyzed, discuss in detail the method(s) used and why, and the findings. You will not be in any way down-graded if you find a null result or do not find support for your hypothesis/hypotheses. Exploratory work without any clear hypothesis is generally discouraged, but if it is relevant to your field it is acceptable. Other creative applications are fine, but you should check with the instructor. There are benchmarks to be completed throughout the semester, as reflected in the course schedule. The first benchmark is outlining your hypothesis/hypotheses, expectations, or exploratory goal. This also serves as an opportunity for me to approve or disapprove of a project, give suggestions for improvement, etc. The second benchmark is a detailed data report on the data to be used in the project. The third is a brief description of the modeling choice with a brief justification. The final project must expand on this brief description for readers unfamiliar with the discussed methods. This should include a full model specification, and an intuitive explanation. You are expected to briefly present your work at the end of the semester, but the presentation is not graded.

Course Schedule

Please note this schedule is subject to change.

Week 1: Anaconda set-up, git, shell introduction, basic Python syntax, data types

Readings:

1. Read https://git-scm.com/docs/user-manual.html up until the section "Exploring Git history"
2. Shaw, Appendix
3. Shaw, Exercises 0 – 15.

Ungraded homework:

1. Install Anaconda https://docs.anaconda.com/anaconda/install/
2. Sign up for a free GitHub account https://github.com/
3. Install git https://git-scm.com/downloads
4. Create a public repository called PythonCourse, and add me (carlson9) as a collaborator

Week 2: Functions, loops, recursion, classes

Readings: Shaw, Exercises 16 – 44.

Week 3: Reading from and writing to files, web scraping and APIs

Graded homework 1 due

Readings:

1. VanderPlas, Preface and Chapter 1.

Week 4: Introduction to NumPy

Readings: VanderPlas, Chapter 2.

Week 5: Data manipulation with Pandas

Hypothesis, expectations, or goals due

Readings: VanderPlas, Chapter 3.

Week 6: Visualization with Matplotlib

Graded homework 2 due

Readings: VanderPlas, Chapter 4.

Week 7: Bayesian statistics

Data report due

Readings:

1. https://www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english/
2. Van de Schoot, Rens, David Kaplan, Jaap Denissen, Jens B. Asendorpf, Franz J. Neyer, and Marcel AG van Aken. "A gentle introduction to Bayesian analysis: Applications to developmental research." *Child development* 85, no. 3 (2014): 842-860. Available on Blackboard.

Week 8: Stan: A probabilistic programming language

Readings: Carpenter, Bob, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. "Stan: A probabilistic programming language." *Journal of statistical software* 76, no. 1 (2017). Available on Blackboard.

Ungraded homework: Install pystan and run a basic model (can be from the above article).

Week 9: Machine learning I

Graded homework 3 due

Readings: VanderPlas, Chapter 5 (first half, to page 445).

Week 10: Machine learning II

Brief model justification due

Readings: VanderPlas, Chapter 5 (second half).

Week 11: Gaussian process regression

Readings: Carlson, David. “Modeling Without Conditional Independence: Gaussian Process Regression for Time-Series Cross-Sectional Analyses.” 2018. Available on Blackboard.

Week 12: Neural networks

Graded homework 4 due

Readings:

1. http://neuralnetworksanddeeplearning.com/chap1.html

2. https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn/

Week 13: Final project presentations

Week 14: Final project presentations

Final projects due at end of week