First-year Common Core Course

EGR 1330 Computational Thinking with Data Science

(Drafted Nov 12, 2019, Revised Jan 21, 2020)

Course Description: (3 credit hours total with 3 lectures and 3 instruction labs per week so the lab has no credit)

This course provides a hands-on learning of principles of programming and data science by introducing Python programming, its relevant modules and libraries, and computational thinking for solving problems in Data Science. Students will learn data science approaches to importing data, manipulating data and analyzing it as well as modeling and visualizing real-world data sets in various science and engineering disciplines.

Pre-requisite: No technical/programming background required

Textbook:

Ani Adhikari and John DeNero, *Computational and Inferential Thinking, The Foundations of Data Science,* Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0), https://www.inferentialthinking.com/chapters/intro.

Course Content (tentative)

- Computational Thinking for Problem solving: Logical problem solving, Decomposition, Pattern Recognition, Abstraction, Representation, Algorithm Design, Generalization
- Python Programming: Variables, constants, data types, data structures, strings, math operators, Boolean operators, expression, program constructs, functions, loop, I/O files, modules and database
- Data Science Fundamentals:
  - *Experimental setup*: Importing and formatting data sets, displaying data, data pre-processing,
  - *Introductory statistical analysis with Python:* Elementary statistics, randomness, sampling, Probability distribution, hypothesis testing, regression, errors
  - *Basic Data analysis, visualization and machine learning*: Data pre-processing, dimensionality reduction, basic supervised/unsupervised learning, performance evaluation metrics

Learning Outcomes: On completion of the course students should

1. Be able to implement basic python programs using computational thinking concepts
2. Know basic Python programming constructs and libraries relevant to Data Science
3. Be able to write Python scripts to perform fundamental data analytics and basic visualization

Resources/Tools

Platforms for Python Programming

1. Anaconda Platform (https://www.anaconda.com/)
   - Anaconda distribution is an open-source Data Science Distribution Development Platform. It includes Python 3 with over 1,500 data science packages making it easy to manage libraries and dependencies. Available in Linux, Windows, and Mac OS X.
2. jupyter (https://jupyter.org/)
   - JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

- The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Modules for Python Programming

3. math module (https://docs.python.org/3/library/math.html) provides access to the mathematical functions defined by the C standard e.g. factorial, gcd, exponential, logarithm.
4. operator module (https://docs.python.org/3/library/operator.html) exports a set of efficient functions corresponding to the intrinsic operators of Python. For example, operator.add(x, y) is equivalent to the expression x+y.

Python modules for Data Science

5. datascience module (http://data8.org/datascience/) written for use in Berkeley's DS 8 course and contains useful functionality for investigating and graphically displaying data.
6. scipy module (https://www.scipy.org/) a Python-based ecosystem of open-source software for mathematics, science, and engineering. It includes some of the core packages:
   - **Numpy:** base n-dimensional array package
   - **Scipy:** fundamental for scientific computing (e.g linear algorithm, optimization)
   - **matplotlib**: visualizations/2D plotting
   - **IPython**: enhanced interactive console
   - **Sympy**: symbolic mathematics
   - **Pandas:** data structures and analysis
7. Scikit-learn module (https://scikit-learn.org/stable/) a library for machine learning in Python. It is a simple and efficient tool for predictive data analysis. It is built on Built on NumPy, SciPy, and matplotlib modules.

Course schedule (Tentative) 3 hours of lecture + 3 hours of lab per week -- 35 lectures & 35 labs -- out of about 40 hours per long semester. The 5 hours are left to adjust for exams, review and teaching speed.

	Lecture	Reading	Lab
1	Introduction - What is Data Science? Data Science tasks	1, 1.1, 1.2, 1.3 3, 3.1, 3.2, 3.3, 3.4	Concepts of Computational Thinking (CT) Example of CT in programming & Data Science; Set up python
2	Cause and Effect - Establishing Causality - Randomization	2, 2.1, 2.2, 2.3, 2.4, 2.5 4.1, 4.2, 4.3, 5, 5.1, 5.2, 5.3	Introduction to Python - Expression - Data types (e.g. int, float, string) - Sequences (array, list, dictionary) (Math module: mathematical functions Operator module: Standard operators) Numpy module: n-dimensional array object)
3	Building Tables	6.1, 6.2, 6.3, 6.4	- Table - Reading data (from csv, txt files) - Displaying data - Data selection/ filtering - Writing to file (Pandas module: DataFrame Datascience module: Table)
4	Data Manipulation	6.1, 6.2, 6.3, 6.4	- Data cleaning (missing values), Data query (Datascience module, Pandas module)
5	Data Analysis and CT		Analyse two data sets: - Federal Aviation Authority Dataset - NewYork city fire department Dataset (Pandas module)
6	Data Visualization - Numerical, categorical data - Charts	7, 7.1, 7.2, 7.3	- chart displays (Matplotlib module, Datascience module)
7	Census, Histograms	6.3, 6.4, 7.2, 7.3 8, 8.1, 8.2, 8.3, 8.4	- Functions - Grouping - Joins (Pandas module: DataFrame Datascience module: Table)
8	Randomness - Monty Hall Problem - Simulation	9, 9.3 9.1, 9.2	Conditionals and Iteration
9	Randomness - Probabilities	9, 9.5 9.4	- Simulation example - Probability calculation (Numpy module)
10	Sampling	10, 10.1, 10.2, 10.3	Sampling example
11	Empirical Distributions	10, 10.1, 10.2, 10.3	Probability Distributions example
12	Hypotheses Testing	11, 11.1, 11.2	Hypotheses Testing example (Numpy module)
13	Hypotheses Testing	11, 11.1, 11.2	Hypotheses Testing example (Numpy module)
14	Decisions and Uncertainty	11.3, 11.4	Making decision
15	A/B testing	12, 12.1	A/B testing example (data science module)
16	A/B testing	12, 12.1	A/B testing example (data science module)
17	Causality	12.2, 12.3	Causality example (data science module)
18	Confidence Intervals	13, 13.1, 13.2	Calculating/visualizing confidence interval (data science module)

19	Interpreting Confidence	13.3, 13.4	Calculating/visualizing confidence interval (data science module)
20	Center and Spread - Properties of the mean	14, 14.1, 14.2	- Properties of the mean - Variability (data science module)
21	Normal Distribution	14.3, 14.4	- histogram - curve (data science module)
22	Sample Means	14.4, 14.5, 14.6	Sample Means (scipy module)
23	Correlation	15, 15.1	Calculating and visualizing correlations Creating heatmaps (scipy module , data science module)
24	Linear Regression	15.2	Linear regression (scipy module , data science module)
25	Least Squares	15.3, 15.4	Least squares (scipy module , data science module)
26	Residuals	15.5, 15.6	Residuals (scipy module , data science module)
27	Inference for Regression	16, 16.1, 16.2, 16.3	Regression example (scipy module)
28	Inference for Regression	16, 16.1, 16.2, 16.3	Regression example (scipy module)
29	Intro to machine learning - supervised/ unsupervised	17.1, 17.2, 17.3	CT exercise in machine learning (ML) Introduce (scikit-learn module: ML)
30	Training and Testing Classification	17.2, 17.3, 17.4	- Training and testing - KNN (scikit-learn module: machine learning)
31	Classification	17.4, 17.5	Simple Classication Project (scikit-learn module)
32	Clustering	17.4, 17.5	- simple clustering implementation (scikit-learn module)
33	Evaluation metrics - Accuracy - Error	17.5, 17.6	- evaluating classifier (scikit-learn module)
34	Evaluation metrics - Confusion matrix - ROC, Precision/recall		Practice interpreting results (scikit-learn module)
35	Interpretation & Making decision	18, 18.1, 18.2	- decision example (scikit-learn module)