Course: **Analytics Programming (DAV 5400)**
Credits: 3 Credits / Graduate
Pre/Coreqs: Spreadsheet Modeling and Analytics
Instructor: James Topor

COURSE OVERVIEW

Code-based solutions can be richer, more accurate, and more flexible than those that rely on off-the-shelf software and analytic packages. This course teaches the programming skills that data analysts need to prepare structured and unstructured data for downstream analysis. Students will learn to use high-level programming languages to create rich data analysis workflows.

COURSE LEARNING OUTCOMES

By the end of this course, students will be able:

- Write programs to clean, filter, aggregate, restructure, and combine data.
- Load data from both structured and unstructured sources.
- Perform basic exploratory data analysis.
- Apply principles of scientific method and reproducible reporting to the development and presentation of code-based solutions.

REQUIRED MATERIALS

- Wes McKinney, *Python for Data Analysis*, 2nd edition, O'Reilly. 2017.
- Mark Pilgrim, *Dive Into Python 3*, freely available web-based content: http://diveintopython3.problemsolving.io/table-of-contents.html

Web-based readings on related topics will also be assigned.

Relevant Software, Hardware, or Other Tools:
We will make use of Python via the freely available ***Anaconda*** environment, including Jupyter notebooks and the Spyder IDE. Details for obtaining and installing the appropriate software will be provided in the course materials. All of the software will work on (or from) both PCs and Macs.

ASSIGNMENTS & GRADING

Approach to Assignments. All projects and assignments are to be written in IPython (Jupyter) notebooks and submitted via GitHub, which is a collaborative data/code repository and version control system. This will give students a publicly available portfolio of increasingly complex projects to help showcase their skills to employers.

Evaluation Criteria. All course projects will be evaluated like work assignments from a demanding employer. The primary evaluation basis is adherence to the deliverables stated in each assignment's functional requirements. To achieve a top grade, students must also adhere to best practices for software engineering principles, including reproducibility; following appropriate coding guidelines; and DRY. Furthermore, assignments must be clearly and concisely written using proper English language grammar and should present relevant supporting text in a logical flow. Presentations should include an appropriate level of detail for their intended audience.

Assignments	Grading
Discussions / Weekly Response Assignments (14 x 10pts) The fourteen weekly discussions will focus primarily on use cases related to analytical programming. Students will prepare short responses to weekly discussion questions, which will be used to prompt group discussion.	14%
Assignments (7 x 30pts) On most weeks when projects are not due, there will be short-form ("mini-project") assignments to help reinforce the current learning material. These assignments may include completing tasks using course analytical tools. Some assignments may require working in small groups.	21%
Projects (4 x 100pts) Students will work individually and in teams on four analytical programming projects. At the end of the course, each student will have a portfolio of increasingly complex projects ready to show an employer.	40%
Final Project (200pts) and Presentation (50pts) Working individually or as part of a small team, students will create a requirements document that outlines a *useful* analytical application that will be applied to available source data. They will then implement the application described in their requirements document. Students will present their final projects to their peers for feedback.	*25%*

- All projects and assignments, unless otherwise noted, are due end of day on Sundays.
- Each week's materials (with the exception of the week of Aug 26) will be made available via Canvas on the previous Friday at 6:00 a.m. ET.
- **Course Completion Requirements**: As a prerequisite to passing this course, you must complete all five projects (including the final), and make the final presentation during the final class session. Failure to either submit any one of the five projects or present your final project will preclude you from achieving a passing grade in this course. Please note that completion of the five projects is not the sole determinant of whether you will receive a passing grade: however, failure to submit any one of the five will prevent you from achieving a passing grade.
- **Discussions / Weekly Response Assignments**: While this material is important, please note that this work only makes up 14% of your grade. Please do the readings, and participate in the discussions and any discussion-related group assignments If you have limited time for the course, please remember to invest the majority of your efforts in completing the projects and assignments. The assignments merit close attention because they will help you to be successful on the projects.
- **Reproducibility Requirement, Testing Requirement, But Not Perfection!** Students are responsible for providing all code and data so that I can test your work. If you turn in code that does not run, you will not receive credit, unless you also include an explanatory note at the time of submission. At the same time, you don't need to turn in perfect code. Generous partial credit will be given for deliverables that are timely, tested, and reproducible. Cutting corners—as long as such measures are documented at the time of submission—is also acceptable.
- **Policy on Sharing and "Stealing" Code.** In this course, you may collaborate and you may take base code from whatever sources you wish. But you must document what you started with, and what you added, so you are graded on your own contributed work!

- **Late work policy**. Please note: **Assignments, discussion responses, and projects cannot be accepted after their due dates for any reason**. Any assignment, discussion or project that is not submitted before its associated deadline will automatically be assigned a grade of ZERO. You will enhance your chances for success in this class if you start early, and turn in your work on time (even if it's not perfect!).

- Students that complete all work in a satisfactory and timely manner will earn a maximum grade of A-. To earn a grade of A in *Analytics Programming*, you'll need to demonstrate work above and beyond what is expected.

GRADING SCALE:

Quality of Performance	Letter Grade	Range %	GPA/ Quality Pts.
Excellent - work is of exceptional quality	A	97 – 100+	4
	A-	92 - 96.9	3.7
Good - work is above average	B+	87 - 91.9	3.3
Satisfactory	B	83 - 86.9	3
Below Average	B-	80 - 82.9	2.7
Poor	C+	77 - 79.9	2.3
	C	70 - 76.9	2
Failure	F	< 70	0

How This Course Works:

Classes are held every week on ***Mondays from 5.40 p.m. to 8:00 p.m. ET***, with the exception of Katz School official holidays. You are strongly encouraged to attend these weekly classes since each will include opportunities for hands-on learning via in-class assignments and case studies as well as a presentation / demonstration of many of the concepts you will need to use for any assignment or project due that week. You are also required to bring your laptop to class as this will serve to facilitate the hands-on learning segments. Class dates can be found in the Course Schedule shown on the following page.

Office Hours can be scheduled by appointment. If you need extra help and are willing to invest the time and effort to be successful, I'll make the time to help you. But…you should not be asking for extra help on a project or assignment the day before or the day it is due, since this indicates that you're not investing the time and effort to be successful.

You are encouraged to ask questions on Canvas where other students will be able to benefit from your inquiries. For the most part, you can expect me to respond to questions asked either via email or via Canvas within one business day.

Instructor Contact Information:

James Topor
james.topor@yu.edu

COURSE SCHEDULE

Students should expect to spend a minimum of 9 hours each week outside of the classroom sessions on the materials, assignments, discussions, and projects required for this course.

WEEK	TOPIC	SCHEDULE OF MAJOR ASSIGNMENTS
Week 1 Aug 26 – Sep 1 Class: M Aug 26	Building Your Toolset: Anaconda and Github;	Environment Setup: Anaconda/Jupyter/Spyder + Github
Week 2 Sep 2 – Sep 8 ***** Class: T Sep 3***	Python Basics: A Refresher (syntax, data types, control flow)	Assignment 2
Week 3 Sep 9 - Sep 15 Class: M Sep 9	Python Data Structures + Functions (includes list comprehensions)	Assignment 3
Week 4 Sep 16 – Sep 22 Class: M Sep 16	Intro to Numpy: Fast Vectorized Operations	Assignment 4
Week 5 Sep 23 – Sep 29 Class: M Sep 23	Pandas Series and Dataframe Objects	**Project 1 Due**
Week 6 Sep 30 – Oct 6 ***** Class: R Oct 3***	Pandas: Data Loading + Storage; Exploratory Data Analysis	Assignment 5
Week 7 Oct 7 – Oct 13 Class: M Oct 7	Data Visualization in Python *** Final Project Requirements Distributed ***	
Week of *Oct 14 – Oct 20*	***** No Class Session This Week due to University Holiday Closures *****	**Project 2 Due**
Week of *Oct 21 – Oct 27*	***** No Class Session This Week due to University Holiday Closures *****	**1st Draft of Final Project Proposal Due**
Week 8 Oct 28 – Nov 3 Class: M Oct 28	Data Cleaning & Preparation (includes regex)	Assignment 6
Week 9 Nov 4 – Nov 10 Class: M Nov 4	Text Mining	**Project 3 Due**
Week 10 Nov 11 – Nov 17 Class: M Nov 11	Combining and Reshaping Data in Pandas	Assignment 7
Week 11 Nov 18 – Nov 24 Class: M Nov 18	Data Aggregation + Grouping	**Final Project Proposal Due Wed. Nov 20**
Week 12 Nov 25 – Dec 1 Class: M Nov 25	Working with Web Data	Assignment 8
Week 13 Dec 2 – Dec 8 Class: M Dec 2	Modeling + Machine Learning with Python Libraries	**Project 4 Due**
Week 14 Dec 9 – Dec 15 Class: M Dec 9	Data Ethics	
Week 15 Dec 16 - 18 Class: M Dec 16	Final Project Presentations + Writeups Due **** Final Project Presentations Monday Dec 16 ****	**** Final Project Writeups Due Wed Dec 18 ****

UNIVERSITY POLICIES & RESOURCES

ACCESSIBILITY AND ACCOMMODATIONS
The Office of Disability Services collaborates with students, faculty and staff to provide reasonable accommodations and services to students with disabilities. Students with disabilities who are enrolled in this course and who will be requesting documented disability-related accommodations should make an appointment with the Office of Disability Services, (646) 592-4132, rkohn1@yu.edu, during the first week of class. Once you have been approved for accommodations, please submit your accommodation letter to ensure the successful implementation of those accommodations. For more information, please visit: http://yu.edu/Student-Life/Resources-and-Services/Disability-Services/

ACADEMIC INTEGRITY
The submission by a student of any examination, course assignment, or degree requirement is assumed to guarantee that the thoughts and expressions therein not expressly credited to another are literally the student's own. Evidence to the contrary will result in appropriate penalties. For more information, visit http://yu.edu/registrar/grad-catalog/

STUDENT SUPPORT SERVICES
If you need any additional help, please visit Student Support Services: http://yu.edu/academics/services/