New York University Tandon School of Engineering

Department of Finance and Risk Engineering
FRE7871 – Financial Data Visualization (1.5 Units)
Spring 2019
Professor Jason Yarmish
Tuesdays 6:00pm-8:41pm; 01/29/19 – 03/12/19
[Jan: 29 ; Feb: 5, 12, 19, 26 ; Mar: 5, 12]
RGSH, Room 425

Professor contact information: Yarmish@nyu.edu

Course Pre-requisite: Graduate Standing

Course Objectives/Description:

The purpose of this course is to teach the fundamentals of data visualization analysis, communication, design and creation. We will learn about data graphics critique, integrity and sophistication. We will learn how to properly design and create static and interactive graphics that maximize the information displayed and absorbed by the viewer using, among other things, principles of human perception and cognition. We will do this in both theory and practice using the Theory of Edward Tufte and others and the open source programming language and integrated development environment (IDE), *Processing*.

We will learn to critique, understand and produce data graphics that should:

- Communicate complex ideas with coherence, precision and efficiency
- Avoid the use of detrimental 'non-data ink'
- Not distort what the data has to say, and recognize when someone is doing just that
- Present many numbers in a small space to make large data sets coherent
- Reveal the data at multiple levels of detail

By the end of this course, you should:

- Know and be able to implement the Theory of Data Visualization in your everyday life and in any future visual data exploration or projects in any language
- Know the fundamentals of coding in Java/ Processing and how to expand that knowledge
- Recognize when and how a visualization is lying and how to avoid that in your own work.
- Create static visualizations that convey massive amounts of information in many ways and on many levels.
- Create interactive visualizations that allow for the most productive exploration and discovery.
- Understand fundamental visualization principles that are universal to most visualizations, programs and packages.
- Understand how to use the fundamentals to design new visualization methods.

Coding Environment:

Software developers generally work in an integrated development environment (IDE) with built-in tools for editing, compiling, running and debugging programs. We will be coding in the freely available *Processing*, an IDE and open source programming language. The language builds on Java but uses a simplified syntax and graphics programming model. Click here to go to the download page for processing. Download and unzip the files of the latest release of Processing, appropriate for your system.

To start the program, simply double click the file  processing.exe.

Course Structure:

Lectures, lab, discussion, course readings, question creation, programming assignments, code and visualization critique/peer review, hand-written exams. Our online classroom contains all class content and peer interaction.

Readings:

The required text for this course is:
The Visual Display of Quantitative Information (Second Edition) by Edward Tufte.
ISBN 978-1-930824-13-3.

Recommended but not required is the text:
Processing, A Programming Handbook for Visual Designers and Artists (Second Edition)
ISBN 978-0-262-02828-8

Additionally students will find the following online materials helpful. Required readings will come from these sources as well:

processing.org
processingjs.org
forum.processing.org
edwardtufte.com

Requirements:

Readings, attendance, class participation, an open mind, assignments, exams.

Assessment:

There will be Readings/assignments/projects at the end of each lecture. You are encouraged to collaborate, but any code and write-ups you hand in must be your own. Plagiarism will not be tolerated. Review and critique of your fellow students' code is required after the completion of each assignment. There will be two exams and a project.

Grading Policy:

Exam 1:	25%
Exam 2:	25%
Project :	30%
Assignments/Projects/ Class participation: 20%	

Assignments/ Exams / Project:

At the conclusion of each class there will be assignments that include readings and implementation of the coding and theory discussed in that class. Exams will test your knowledge of the theory and the final project will require a merging and integration of practice and theory.

Course Overview:

Week	Topics
1	**Introduction:** Overview of the course, its structure and our online classroom History and progression of Data Visualization Information Visualization vs Data Visualization Fundamental changes to Data Visualization as a result of mass data collection and automated methods for building visuals Examples of amazing visualizations created with Processing to inspire our journey in the class. **Theory:** Graphical Excellence Graphical Integrity **Implementation of the theory through programming:** Introduction to the Processing language, IDE and Community Canvas/ Shapes and properties/ Usual variable types/ other variable types such as color and it's various forms and properties/ random number generation/ Arrays/ Loops/ Shapes such as ellipses, arcs, quadrangles…/ Our First Visualization
2	**Theory:** Sources of Graphical Integrity and Sophistication **Implementation of the theory through programming:** Setup and draw environments/ Font type/ mouse variables/ loading data/ String functions/ Save Image/ mapping/ Tables/ translate, scale, rotate/ push-pop matrix and style/ scope/ useful functions/ key-mouse inputs Our First Interactive Visualization

3	**Theory:** Data-Ink and Graphical Redesign Chartjunk: Vibrations, Grods and Ducks Data-Ink Maximization and Graphical Design **Implementation of the theory through programming:** Advanced color/ color palettes/ mouse interaction/ key interaction/ distance/ working with images/ filters/ functions/ multiple files/ ide tabs **Exam1:** Graphical practice
4	**Theory:** Multifunctioning Graphical Elements Data Density and Small Multiples Aesthetics and Technique in Data Graphical Design When to be rigid and when to bend the rules **Implementation of the theory through programming:** Hue, saturation, brightness/ building color palettes and color resources/ ternary operator/ switch statements/ interactive zoom, rotate, hover and shifting
5	**A merging of Programming and Theory:** Full visual analysis: discovering patterns difficult to find via statistical techniques **Implementation of the theory through programming:** Libraries with examples/ implementing the building of an interactive slider **Exam2:** Theory of data graphics
6	**A merging of Programming and Theory:** Building interactive maps/ Fibonacci sequence visualization and exploration **Implementation of the theory through programming:** Interaction to movie/ Live data/ cursor/ Debugger(IDE)/ Tweak(IDE)
7	**A merging of Programming and Theory:** **Final Project – Comprehensive Visual Data Analysis** • Exploratory graphics | explanation of progression | story of your data • Final static visualization • Final interactive visualization **Programming:** **(examples of expanding gained knowledge and ability)** Using Processing for Android/ JavaScript and python

Moses Center Statement of Disability:

If you are student with a disability who is requesting accommodations, please contact New York University's Moses Center for Students with Disabilities (CSD) at 212-998-4980 or mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found at www.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 2nd floor.

NYU School of Engineering Policies and Procedures on Academic Misconduct

A. Introduction: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.

B. Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:

1. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
2. Fabrication: including but not limited to, falsifying experimental data and/or citations.
3. Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.
4. Unauthorized collaboration: working together on work that was meant to be done individually.
5. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission has been received from the course instructor(s) or research adviser involved.
6. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.