RICE UNIVERSITY
School of Engineering

Data Visualization
COMP 665

Professors: Joe Warren, Marmar Orooji
Contact Information: jwarren@rice.edu , marmar.orooji@rice.edu
Office Hours: TBA
TA: TBA

Course Materials:

All required course materials will be available via the Canvas website.

Software Requirements:

In the first week of the class, Tableau software is introduced and students must complete the first practice project in Tableau. Students are required to download and install Tableau free product, which is Tableau Public. The instructions on downloading and installing Tableau Public is provided on the Canvas website.
From the second week of the class up to the end, students will work with Python programming language through the cloud-based programming environment Vocareum. All the required Python packages are installed in Vocareum and students will use this web-based environment to write their Python codes for the weekly projects and submit their solutions. The links to Vocareum for each week's project is provided in the Canvas website.
Although students must use Vocareum to run their Python codes and submit their solutions, Anaconda is also introduced to students as an open-source platform to perform Python locally on their machines. The instructions on downloading and installing Anaconda is provided on the Canvas website.

Course Overview:

Data is being generated by humans and algorithms at an astounding rate. Having the ability to analyze and interpret this data visually is a key technique for coping with this explosion. This course will cover the basic ways that various types of data can be visualized and what properties distinguish useful visualizations from not so useful one. The class will use Python as both the primary tool for processing the data as well as creating visualization of this data. To enhance the students' depth of knowledge, the class will also cover some of the geometric algorithms used to create advanced visualizations.

The **learning outcomes** for this class are:

- Students will have a basic knowledge of various visualization techniques for a range of data types and be able to create high-quality visualizations,
- Students will be proficient working with popular Python packages for processing and visualizing data,
- Students will understand the basic geometric algorithms that underlie creating several types of advanced visualizations.

Course Requirements and Details:

The class will last 14 weeks. Each week contains:

- videos introducing important class topics and walking through examples,
- pages consisting of detailed notes on various packages and visualization topics,
- a practice project, with the provided solution, as a supplement resource to help students with their required project of the week,
- a required assessment:
  - weeks 1 and 2, include quizzes hosted in Canvas,
  - weeks 3-14, include required projects in Python hosted on Vocareum and linked to Canvas as an external tool. These projects will be assessed in Vocaruem using both machine-grading and peer-grading.
- a grading-guide for peer reviews containing the solution and rubrics (will be open after each project is due),
- a feedback survey to collect student's comments on the week's materials,
- a discussion forum to discuss any issue with peers, instructor, and the TA.

Each week includes a required assessment:

- Weeks 1 and 2 required assessments are Canvas quizzes that each count 4% and comprise 8% of your total grade.
- Weeks 3-13 required assessments are Python projects that each count 7% and comprise 77% of your total grade. These projects will be completed in Vocereum and will be both machine-graded and peer-graded.
- Week 14 required assessment is a final Python project that counts 15% of your total grade. This project will be completed in Vocereum and will be both machine-graded and peer-graded.

The class does not have midterm exam. Week 14's project is the final project.

Weekly Activities	Due Date	Points	Grade Weight
Videos	---	---	---

Pages (detailed notes)	---	---	---
Practice Projects (with provided solutions)	---	---	---
Quiz 1	Sunday after week 2's live session, at 10 pm CT	40	4%
Quiz 2	Sunday after week 2's live session, at 10 pm CT	40	4%
Weeks 3-13 Projects	Sunday after each week's live session, at 10 pm CT	70	Each project 7% - together 77% of total grade
Week 14 Project	Sunday after week 14's live session, at 10 pm CT	150	15%
Projects' Peer Review	Wednesday after each week's project's deadline, at 10 pm CT	--- Note: Failure to complete the required number of peer reviews results in 20% deduction of the peer-graded portion of the project's grade.	---
Feedback survey	Monday after each week's live session, at 10 pm CT	---	---

Course Outline:

Week 1 - Data visualization

Topics:

- Elements of visualizations
- Perceptual principles of visualization
- Tableau

Required assessment is Canvas quiz on the basics of visualization (due at end of week 2). Optional practice project is creating a simple visualization in Tableau.

Week 2 - Python environments

Topics:

- Anaconda
- Jupyter notebooks
- Vocareum
- Machine-grading in Vocareum

Required assessment is a Canvas quiz on Python fundamentals. Optional practice project is to implement a simple visualization in Jupyter notebooks that is both machine-graded and peer-graded.

Week 3 - 1D functional data

Topics:

- CSV files and the csv module
- matplotlib package
- Coding style

Project data source is stock market index data from FRED. Read times series data from CSV files using the csv module. Clean/process times series data using the datatime module (machine-graded). Create **line plots** of the results using matplotlib (peer-graded).

Week 4 - 1D parametric data

Topics:

- numpy package
- Scientific visualization
- Animations in matplotlib

Project data source is solutions to orbital equations for Earth and Sun. Code/solve PDEs for Earth/Sun using scipy.optimize (machine-graded). Create **curve** and **animations** of resulting orbits in matplotlib (peer-graded).

Week 5 - 2D heat maps/*images*

Topics:

- Raster graphics in matplotlib
- Structure of matplotlib figures

- Colors and colormaps

Project data source are dynamical systems including Julia sets, Mandelbrot sets, and Newton basins. Implement functions for creating ***images*** of all three systems (machine-graded). Assess correctness of raster images (peer-graded).

Week 6 - 2D text data

Topics:

- nltk package
- Text in matplotlib
- Geometric arrangements of boxes

Project data source is text files of historical documents. Extract word frequency counts using nltk and compute non-overlapping word boxes for a word cloud in Python (machine-graded). Generate **word frequency plots** in nltk and plot **word clouds** in matplotlib (peer-graded).

Week 7 - 2D networks (graphs)

Topics:

- networkx package
- Optimization via scipy.optimize module
- Community detection via python-louvainx package

Project data source is networkx's built-in library of example graphs. Compute layout for small graphs using all-pairs shortest paths (networks) and minimize (scipy.optimize). Also, compute nodes with high betweenness (machine-grade). Visually assess resulting **graph** layouts (peer-graded).

Week 8 - 2D maps (part 1)

Topics:

- pandas package and dataframes
- Vector graphics and SVG files
- XML files and xml.minidom module

Project data sources is county-level map of the USA. Parse .svg file as xml using xml.minidom, extract county boundaries from parsed file and compute county centers, create dataframe and write as .csv file using pandas (machine-graded). Generate html and csv output using pandas, display a map with **county boundaries** and overlying county centers using **scatter** (peer-graded).

Week 9 - 2D maps (part 2)

Topic includes:

- Dataframe manipulation in pandas
- Scripts
- Overlaying images with data

Project data source is EPA county-level cancer-risk data. Load cancer-risk data into dataframe from .csv file, merged cancer risk data with county boundary data from USA map (machine-graded). **Draw county centers in circle** whose size accords to the cancer-risk data(peer-graded).

Week 10 - 2D point clustering (part 1)

Topics include:

- Object-oriented programming and classes
- Algorithmic efficiency
- Testing
- Hierarchical clustering

Project data source is EPA county-level cancer-risk data. Implement fast closest pairs and hierarchical clustering (machine-graded). **Draw hierarchical clusters** for cancer-risk data on USA map(peer-graded).

Week 11 - 2D point clustering (part 2)

Topics include:

- Comparison of visualization quality
- Comparison of code efficiency
- k-means clustering

Project data source is EPA county-level cancer-risk data. Compute k-means clustering (machine-graded). **Draw k-means cluster on the USA map**. Compare the structure of hierarchical vs k-means clusters (peer-graded). The Optional project is to do k-means via scikit-learn.

Week 12 - 2D scalar data meshing (part 1)

Topics include:

- plotly package
- Array computations in numpy
- Image processing in matplotlib

Project data source is NGS digital elevation map of the Grand Canyon. Load tiff file for Grand Canyon DEM, manipulate DEM using numpy, extract features using numpy (machine-graded). Generate **contour plots** and **3D height plots** of Grand Canyon DEM.

Week 13 - 2D scalar data meshing (part 2)

Topics include:

- Web-based visualization
- 1D and 2D arrays in plotly
- Triangular meshs in plotly

Project data source is USGS digital elevation map of the Grand Canyon. Compute unraveled quad mesh for rectangular grid of data. Compute triangulations for quad mesh via quad splitting (machine-graded). Compare various **3D surface** representations in plotly, examine effect of quad splitting methods (peer-graded).

Week 14 - 3D gridded volume visualization

Topics include:

- Class definition in Python
- Binary files via the struct module
- Code refactoring
- Animation with buttons and sliders in plotly

Project data source is UC Davis CT data. Define Volume class for 3D gridded data. Compute synthetic data for Volume objects and implement a contour_volume method using Marching Cubes. Write binary file reader for CT data (machine-graded). Plot **contours** of various synthetic and real volumes in plotly, modify plotly MRI-slicer demo to use CT data and create animations on synthetic and real volumes with buttons and sliders in plotly (peer-graded).

RICE UNIVERSITY
School of Engineering

Honor Code:

All Masters of Computer Science students are sworn to adhere to the tenets set forth in Rice University's Honor Code. All written assignments for this class submitted to the instructor must carry the signed Honor Code pledge: "On my honor, I have neither given nor received any unauthorized aid on this (examination, quiz or paper)." (with signature and/or student ID number). Failure to include this pledge will result in an ungraded and returned assignment.

Please be aware of the Honor Code addendum that is part of the Student Handbook. This will provide more specific language about how the Honor Code should be adhered to as part of the Computer Science program.

Attendance Policy:

Attendance is not required, but strongly encouraged for all live class sessions. The benefit of this program is the time spent during these live sessions interacting with your classmates and faculty, with much of the learning coming from these interactions. You will not be penalized for not attending, but please note that attendance in these live sessions is a major part of not only learning the content, but also the experience as a whole.

Disability-based Accommodations:

If you have a documented disability that may affect academic performance, you should: 1) make sure this documentation is on file with Disability Resource Center (Allen Center, Room 111 / adarice@rice.edu / x5841) to determine the accommodations you need; and 2) meet with the instructors to discuss your accommodation needs.

TITLE IX STATEMENT

Rice University cares about your well being and safety. Rice encourages any student who has experienced an incident of harassment, pregnancy discrimination, gender discrimination, or relationship, sexual, or other forms interpersonal violence to seek support through The SAFE Office. Students should be aware when seeking support on campus that most employees, including myself, as the instructor/TA, are required by Title IX to disclose all incidents of non-consensual interpersonal behaviors to Title IX professionals on campus who can act to support that student and meet their needs. For more information, please visit safe.rice.edu or email titleixsupport@rice.edu.