UNIVERSITY OF VIRGINIA SUMMER 2025

DS 6001: Practice and Application of Data Science

Live Sessions: Wednesdays 7:15-8:15pm, Online

Learn How to Surf the Data Pipeline

Data is almost never ready to be analyzed without a great deal of work to prepare the data first. Data scientists spend at least 80% of their time getting, cleaning, and managing data. The goal of this course is to make this huge part of data analysis easier, faster, less frustrating, and more enjoyable.

This course begins with the single most important skill for a data scientist: **how to find the help you need** to solve the inevitable problems, errors, and anomalies that will occur as you code. After that, the course is divided into three parts. First, **how do we acquire data?** We will discuss external files with flat, tabular structure, JSONs, APIs, web-scraping, and remote SQL and NoSQL databases. Second, **how do we clean data?** We will cover SQL and pandas, including merging and reshaping dataframes.

Instructor
Jonathan Kropko

Microsoft Teams
DS 6001: Practice and Applications of DS

Office Hours
Over Teams: Send me a message anytime

Over Zoom: See Canvas for regular times

Email
jkropko@virginia.edu

Download or update the following free software as soon as possible:

1

MINICONDA

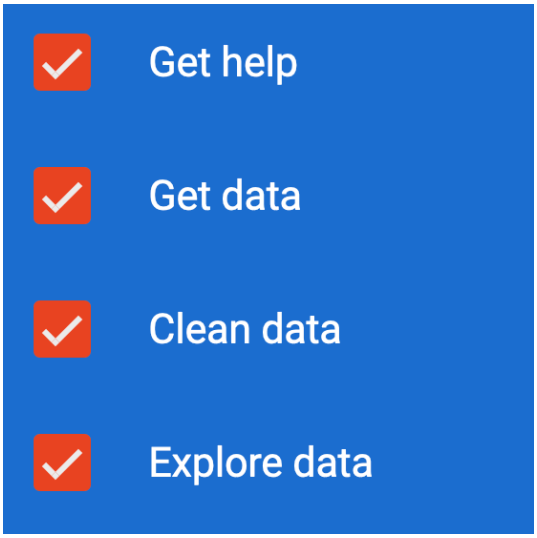
Virtual Python environments: https://www.anaconda.com/docs/getting-started/miniconda/install#quickstart-install-instructions

2

VS CODE

A user interface for Python and other programming languages: https://code.visualstudio.com/

Third, **how do we perform simple analyses to understand our data?** We will work with summary and descriptive statistics tables, static visualizations using matplotlib and seaborn, and interactive visualizations using plotly.



Course Objectives

By the end of the semester you will be able to

1. Recognize how to get help on code in a way that is accurate and efficient while using code forums and large language model-based chatbots effectively
2. Implement methods for acquiring electronic data in many formats — CSVs and flat files, JSONs, from APIs, and using web scraping — and loading it into Python
3. Understand the purpose, typology, and language of relational and NoSQL databases, including how to implement SQLite, PostgreSQL, MySQL, and MongoDB in Python, and how to query databases with SQL and the MongoDB query language
4. Employ methods for wrangling, joining, and aggregating data using pandas
5. Understand relationships in the data using summary statistics, hypothesis tests, and measurement models, as well as visualization using matplotlib, seaborn, plotly, and dash

PYTHON LIBRARIES

Python is a general programming language that can be used for many purposes. But for data management and modeling we will need to download packages that contain additional functions. For example, one additional package we will use is *pandas*, which is a powerful engine for working with data in tabular format. To download this package open a console in JupyterLab or Visual Studio Code, or open a command terminal, and type:

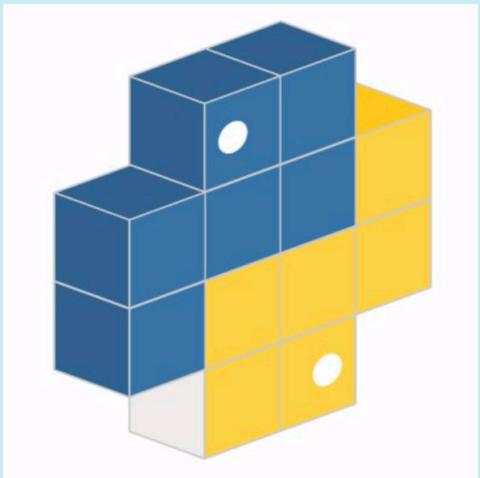
pip install pandas

Another command that works is

conda install pandas

To use the functions in the *pandas* library, in your code you will need to type

import pandas as pd



Readings

There are no textbooks required for purchase. I am currently writing a textbook that discusses all of the course material that I will make available for you to use as a reference when working on the lab assignments. I would be very grateful for any feedback and comments on the book as I continue to develop it.

Some of the readings for this course will come from the O'Reilly online data science library, which is free for UVA students. To access this library, you must go to this website: https://www.oreilly.com/library/view/temporary-access/, click on "Select your institution" and "Not listed? Click here.", and enter your UVA email address. Some of the texts we will use are listed below along with links that should work once you log on.

- *Python Data Science Handbook* by Jake Vanderplas: https://www.oreilly.com/library/view/python-data-science/9781491912126/
- *Python API Development Fundamentals* by Jack Huang, Ray Chung, Jack Chan: https://www.oreilly.com/library/view/python-api-development/9781838983994/
- *Getting Started with Beautiful Soup* by Vineeth G. Nair: https://www.oreilly.com/library/view/getting-started-with/9781783289554/
- *Mastering pandas - Second Edition* by Ashish Kumar: https://www.oreilly.com/library/view/mastering-pandas-/9781789343236/
- *Hands-On Data Analysis with Pandas* by Stefanie Molin: https://www.oreilly.com/library/view/hands-on-data-analysis/9781789615326
- *MySQL Cookbook, 3rd Edition* by Paul DuBois: https://www.oreilly.com/library/view/mysql-cookbook-3rd/9781449374112/

PYTHON: THE WILD WEST OF DATA-BASED COMPUTING

No one person or company creates Python. It's a collective effort of thousands of researchers in many fields around the world. When a researcher develops a new technique to conduct statistics or work with data, he or she writes Python code to perform this task and distributes it through the Python Package Index (PyPi) or another repository. That's one of the best things about Python.

But the drawback is that there are often many, many ways to do the same thing in Python. The approaches we will discuss are not the only way to perform the tasks we need to accomplish. Whenever possible, I will try to present the approaches in class that are easiest to teach and to understand, that use fewer lines of code, and run more quickly. You might find another way to do these things, and that's great. But if you stick to the ways we talk about in class, I can more easily follow your work and give you full credit.

- *Python for Data Analysis, 2nd Edition* by Wes McKinney: https://learning.oreilly.com/library/view/python-for-data/9781491957653/

- *Practical Statistics for Data Scientists, 2nd Edition* by Peter Bruce, Andrew Bruce, Peter Gedeck: https://learning.oreilly.com/library/view/practical-statistics-for/9781492072935/

- *Fundamentals of Data Visualization* by Claus O. Wilke: https://serialmentor.com/dataviz/

There will be additional readings posted on Collab in the form of academic articles and data science blog posts, all of which will be accessible online free of charge.

Lectures: Live Coding Demonstrations

This course is about the real-world challenges that will arise in any data project outside and beyond analytic models. The best way to discuss and demonstrate these issues and the paths to solving them is to work on applied problems together during our live sessions. During these sessions, I will present a real-world use case for the material we've been practicing. I will provide you in advance with the links and background on the problem we will try to solve. Then during class I will work through the problem, making mistakes as I go and consulting help documentation and Stack Overflow as needed since that is the honest workflow of professional data science. I will ask the class questions as I go so that we work on solving the problem together, and I welcome questions and comments on the work at any time as I proceed.

You will be responsible for writing a script or a notebook that reproduces the work that I do during this session, and you will be required to convert your notebook to a text-based Python script, then copy your code into the Canvas assignment. These assignments will only be graded for submission, not for accuracy. That said, I will instruct the TAs to spot-check these submissions to make sure students make a good faith effort to write the complete code. I will record these sessions and share the videos for students' reference.

With one hour per live session, we won't have enough time to cover all of the topics you will be responsible for in the lab assignments. But we will focus on relevant parts of the material that are trickier. We will also devote some time during the beginning of class for questions about the labs, quizzes, and other parts of the course.

Course Website and Equipment

All grades, labs, and readings will be posted on the UVa Canvas site for the course, accessible at https://canvas.its.virginia.edu. If you are officially enrolled in the course, the course website should already be accessible to you. If you are not officially enrolled, please speak to me so I can arrange for you to have access to the course material. You will need a computer capable of running Python, and a stable internet connection for accessing course material, downloading data, and installing packages for extra functionality in Python.

Assessment

There are 1800 possible points in this course. Your grade will be determined by:

- Your performance on twelve **lab assignments** (100 points each, 1200 points total, 66.7% of the grade)
- Your performance on twelve **reading quizzes** (25 points each, 300 points total, 16.7% of the grade)
- Your submissions for ten **live coding sessions** (30 points each, 300 points total, 16.7% of the grade)

There is no midterm or final exam in this course. The final grades will be determined from the final percents according to the table on the right.

Percent range	The letter grade will be no lower than
> 97%	A+
93% - 97%	A
90% - 93%	A-
87% - 90%	B+
83% - 87%	B
80% - 83%	B-
77% - 80%	C+
73% - 77%	C
70% - 73%	C-
67% - 70%	D+
63% - 67%	D
60% - 63%	D-
< 60%	F

Lab Assignments

There will be one lab assignment for each of the 12 modules. These assignments will help you practice the techniques in Python for working with data. Each lab contains several questions that ask you to write and run code, interpret results, and describe in plain language what the does and why. You will write a lab report using a Jupyter Notebook, which is a document that allows you to easily combine text, images, Python code, and the results of Python code all in the same document. You will be required to format these lab reports in a clean and professional style, using the Markdown language to format the document.

We will be using a grading platform called Gradescope to grade the labs. Gradescope allows us to keep your assignments well-organized so that we can return grades to you much more quickly. Gradescope makes things harder for you when you submit your homework: the biggest drawback is that Gradescope cannot accept .ipynb notebook files. You will have to follow some steps carefully to convert your notebook to a PDF then mark the areas on your PDF that belong with each question. But the payoff is more and higher quality feedback on your work, returned much more quickly than with the default Canvas method. Instructions for submitting your lab are listed in this Google document: tinyurl.com/DS6001gradescope.

Gradescope also provides a system for requesting regrades that allow you to point out the exact question that you would like us to take another look at. We are happy to consider regrade requests, but please use the Gradescope interface.

Reading Quizzes

Every module includes 2 to 4 articles, blog posts, or chapters from an O'Reilly textbook that provide the necessary background to better understand the course material. You will be responsible for completing a 10 question multiple choice quiz that requires you to reflect on these readings. The

quizzes and links to the readings will be available on Collab. You are free to use the textbook I am writing as a reference if you want, but the quizzes will focus on the external background readings for each module. You may have all readings and course material open while you take the quiz.

Late Work Policy

We understand that sometimes life gets very busy and that you won't be able to meet every deadline. To help you manage your priorities and not fall behind with your learning, you will be given three automatic extensions for lab assignments, three automatic extensions for reading quizzes, and three automatic extensions for live coding submissions. This policy allows you to submit your homework up to 7 days beyond the due date during the term without penalty or question. If any labs, quizzes, or live coding assignments are submitted later than the due date for the fourth time or more then that assignment will automatically be given a zero unless prior permission is obtained or there are significant extenuating circumstances.

Accessing Grades and Feedback on Canvas

We will release grades for quizzes immediately upon submission, and grades for lab and live coding assignments within one week of the due date. Numeric grades will be posted in the Canvas gradebook, and specific feedback will be posted on Gradescope, which is the same platform students will use to submit their lab assignments.

Microsoft Teams

Teams is a web-based service that provides message boards and file sharing to organizations. I've started a Team for our course, and **I intend for this Team to be our primary mode of communication**. Part of the logic of Teams is to replace long email chains that clog up our inboxes. Teams also should make it easier for students to communicate with each other and with me.

One problem many people have with Teams, especially if they are not used to this platform, is that they tend to forget to check the Teams website and they miss messages. To help with this issue, I strongly recommend downloading the Teams desktop application here: https://www.microsoft.com/en-us/microsoft-teams/download-app. I strongly recommend **keeping the desktop app open at all times** so that you can be notified when you are mentioned or when you receive a message.

Teams is used for private messages and for public class discussion. To send me a message, find "Chat" on the left and click on the new chat button at the top. Then type Jonathan Kropko into the "To:" field. I will have the Teams app open most of the time, and I will do my best to respond promptly to messages over Teams. I will respond to emails as well, but **Teams is the best way to contact me**.

On Teams, we've created the "DS 6001: Practice and Applications of Data Science" team for general discussion about the class, troubleshooting the material and solving problems with the labs (without sharing code — see "Collaboration and Cheating" below), miscellaneous discussion related to data and Python, and questions regarding issues with getting software and packages to run properly. It is also a good place for sharing examples of interesting/inspiring/frightening examples of data being applied in the world and ethical and

technical issues regarding these examples. Please note that we will be using Teams for chat, only. All course material will be posted to UVA Canvas, and not to Teams.

Large Language Models (LLMs), Chatbots, and ChatGPT

We now live in a world where AI such as ChatGPT, GitHub Copilot, and Windsurf can take on coding tasks, and can often generate code that both accomplishes the goal and is straightforward and current. That presents us with both an opportunity and a real challenge.

The opportunity is a chance to learn about very modern tools that are being employed in industry by professional data scientists to improve and speed up our work. I will take time during class to discuss options for using generative AI, we will use Windsurf together in a way that is integrated with an independent development environment like VS Code, and we will use it as part of our live coding in class.

The challenge is that gen AI like Windsurf can do so many basic coding tasks that we never learn how to do these tasks ourselves. Overuse of gen AI can prevent you from acquiring basic data skills, and these skills are necessary to use gen AI from the level of mastery needed to become truly efficient and effective as a data scientist that uses gen AI as a tool. Not only that, but a rule barring gen AI for these tasks would be unenforceable, and it can create a classroom culture that is more confrontational than supportive if the professor tries to crack down on AI while students find new ways to circumvent any prohibitions.

To make the class the best it can be for all of us, let's make an agreement: I will show you some of the best gen AI tools for data science available at the present moment and I will model their use every day in class as part of a gen AI workflow, and you will do your best to learn the methods, approaches, and code to accomplish the tasks you need without using gen AI to replace the need to acquire this knowledge. If we both hold up our end of the agreement, then you will become a confident user of data who uses gen AI to speed up a workflow you have already mastered.

With that said, you are free to use gen AI to help you on lab assignments. For the specific rules and guidelines regarding use of gen AI with respect to academic dishonesty and cheating, see the "Collaboration and Cheating" section below.

Collaboration and Cheating

Although every student is responsible for their own work, you may chat and Zoom with one another to work together on labs and you may consult with one another on your live coding submissions. In that context, the line between collaboration and cheating can get a bit fuzzy. In general, the difference between collaboration and cheating comes down to intent. Cheating is trying to

circumvent the learning process. Collaboration is trying to help yourself and your classmates learn the material more deeply. Use your own sense of right and wrong, but to help clarify the difference here are examples of cheating:

- Sharing/showing code for the purpose of circumventing the learning process (for example, letting someone copy code because they are running up against the deadline)
- Plagiarizing text on a lab assignment.
- Posting your answers for a lab or live coding assignment on GitHub or another public website.
- Asking people on Stack Overflow or another website to help you with the lab assignment itself (although asking about the general skills and techniques needed to complete the lab, or about specific errors is fine). In general, before asking anyone for help, you should make a real effort to complete the work on your own.
- Submitting a lab that is an exact copy of another student's lab assignment, such as copying a file and changing only the name of the submitter.

The penalty for violations of the policies stated above is a 0 on the assignment in question.

Here are examples of collaboration that are acceptable and encouraged:

- Paired or small group programming on the lab and live coding assignments: in a paired programming session, one person generally "drives" by writing the code, and the other person "navigates" by talking through the code as the driver types. Usually, the partners switch roles frequently. This kind of paired programming is acceptable and encouraged, and can be a very effective way for two people to work together. It tends to fail when there are more than three people involved as the potential for the most experienced individuals to dominate the session increases. Two participants is ideal.
- It is OK for two people to submit labs that partly share the same code if they have been working together in a paired programming session. But it is better if each person types out all of the code for themselves to avoid submitting code with exactly the same formatting. Any text in the document must be original, and not copied from a partner.
- Using code that was suggested on Stack Overflow or another website is OK to use sparingly and as needed to solve problems and correct errors.
- Consulting with one another on your data pipeline projects is highly encouraged. You may suggest code to help each other get unstuck and accomplish tasks, but you should not write the code for another student's project or take complete code from another student.

Policy on Use of Gen AI for Live Coding Assignments

For the live coding assignments, you will be typing the same code we type in class. You are free to use gen AI for this code, especially if we use gen AI in class for parts of the code. However, if we notice that your code departs significantly from the code we wrote together in class, especially in ways that fundamentally alter the output of the code, we will penalize the grade for the check-in. The purpose of this policy is to preserve what is effective about live coding - working on a project collaboratively.

Policy on Use of Gen AI for Lab Assignments

Labs will require a mix of code and written responses as I ask you to perform tasks and reflect on those tasks. Gen AI can both generate code and plain language. For the labs, the policy on use of gen AI is

- No generative AI is allowed for questions that require a written response in plain English. I want your own words.
- For code, gen AI is allowed. However:
  - If gen AI creates code that accomplishes the task and follows the approaches we use in class or another approach that matches current best practice, we will award full credit.
  - If gen AI creates code that does not answer the question correctly, we will penalize the grade.
  - If gen AI creates code that works for accomplishing the task, but uses code that is very deprecated or overly complex, we will penalize the grade.
  - If gen AI creates code that works, but is very different from the approaches discussed in class, we will ask you to revise your submission by adding text or comments that explain every line of code in your own (non AI) words for the given question before we grade it.

The purpose of the policy listed above is to give you the freedom to use gen AI as a tool but to also encourage you to meaningfully engage with the underlying Python code, and to work towards a mastery-level skillset that includes both gen AI and Python.

Policy on Use of Gen AI for Reading Quizzes

Generative AI can be used to analyze readings, if you choose. However, using gen AI in this way to summarize the readings may result in losing key details that are needed to answer the quiz questions. So we strongly recommend that you do the readings, even if you also use gen AI to help keep track of main points. We ask that you do not use gen AI to answer quiz questions themselves: it

defeats the purpose of the quizzes to foster a deep reading and encourage thinking about the readings.

Schedule, Readings, Important Dates

Module 1: Getting Yourself Unstuck

- Readings:
  - Textbook: Vanderplas, chapter 1 https://www.oreilly.com/library/view/python-data-science/9781491912126/
  - Blog: https://www.atommorgan.com/blog/stackoverflows-toxicity-problem
  - Blog: https://medium.com/@Aprilw/suffering-on-stack-overflow-c46414a34a52
  - "Surfing the Data Pipeline with Python", chapter 1
- Live session (Zoom link on Canvas):
  - Wednesday, May 21, 7:15-8:15pm
- Reading quiz and lab assignment due dates (no live coding assignment this week):
  - Tuesday, May 27, 11:59pm

Module 2: Working with Electronic Data Files

- Readings:
  - Textbook: Kumar, "I/Os of Different Data Formats with pandas" through "URL and S3" https://www.oreilly.com/library/view/mastering-pandas-/9781789343236/
  - Textbook: Molin, "Working with Pandas DataFrames" (and all subsections through"Further Reading") https://www.oreilly.com/library/view/hands-on-data-analysis/9781789615326
  - "Surfing the Data Pipeline with Python", chapter 2
- Live session (Zoom link on Canvas):
  - Wednesday, May 28, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, June 3, 11:59pm

Module 3: Working with JSON Data

- Readings:
  - Blog: https://medium.com/analytics-vidhya/python-dictionary-and-json-a-comprehensive-guide-ceed58a3e2ed
  - Blog: https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/
  - Official documentation: https://www.json.org/json-en.html
  - "Surfing the Data Pipeline with Python", chapter 3
- Live session (Zoom link on Canvas):
  - Wednesday, June 4, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, June 10, 11:59pm

Module 4: Working with APIs in Python

- Readings:
  - Textbook: Huang, Chung, and Chan, chapter 1 https://www.oreilly.com/library/view/python-api-development/9781838983994/
  - Article: http://computationalculture.net/objects-of-intense-feeling-the-case-of-the-twitter-api/
  - "Surfing the Data Pipeline with Python", chapter 4
- Live session (Zoom link on Canvas):
  - Wednesday, June 11, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, June 17, 11:59pm

Module 5: Web-scraping with BeautifulSoup

- Readings:
  - Official documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

  - Textbook: Nair, chapter 8 https://learning.oreilly.com/library/view/getting-started-with/9781783289554/
  - Blog: https://towardsdatascience.com/ethics-in-web-scraping-b96b18136f01
  - "Surfing the Data Pipeline with Python", chapter 5
- Live session (Zoom link on Canvas):
  - Wednesday, June 18, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, June 24, 11:59pm

Module 6: Databases in Python

- Readings:
  - Article: https://clutejournals.com/index.php/IJMIS/article/view/7587/7653
  - Article: https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf
  - Blog: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems
  - "Surfing the Data Pipeline with Python", chapter 6
- Live sessions (Zoom link on Canvas):
  - Wednesday, June 25, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, July 1, 11:59pm

Module 7: Database Queries

- Readings:
  - Textbook: DuBois, Chapters 3, 14 https://www.oreilly.com/library/view/mysql-cookbook-3rd/9781449374112/
  - Article: https://ieeexplore.ieee.org/document/6359709
  - "Surfing the Data Pipeline with Python", chapter 7

- Live session (Zoom link on Canvas):
  - Wednesday, July 2, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, July 8, 11:59pm

Module 8: Data Cleaning with Pandas

- Readings:
  - Article: https://www.jstatsoft.org/article/view/v059i10/
  - Article: http://people.cs.uchicago.edu/~aelmore/class/topics17/wrangling-wild.pdf
  - Textbook: McKinney, chapters 7, 10, 12 https://learning.oreilly.com/library/view/python-for-data/9781491957653/
  - "Surfing the Data Pipeline with Python", chapter 8
- Live session (Zoom link on Canvas):
  - Wednesday, July 9, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, July 15, 11:59pm

Module 9: Merging and Reshaping Dataframes in Pandas

- Readings:
  - Textbook: McKinney, chapter 8 https://learning.oreilly.com/library/view/python-for-data/9781491957653/
  - Article: https://drops.dagstuhl.de/opus/volltexte/2020/11960/pdf/OASIcs-PLATEAU-2019-6.pdf
  - "Surfing the Data Pipeline with Python", chapter 9
- Live session (Zoom link on Canvas):
  - Wednesday, July 16, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:

- Tuesday, July 22, 11:59pm

Module 10: Exploratory Data Analysis

- Readings:
  - Textbook, chapters 1 and 3: https://learning.oreilly.com/library/view/practical-statistics-for/9781492072935/
  - Article: https://www.tandfonline.com/doi/pdf/10.1080/00031305.2016.1154108?needAccess=true
  - John W. Tukey "Exploratory Data Analysis: Past, Present, and Future", pages 1-7: https://apps.dtic.mil/sti/pdfs/ADA266775.pdf
  - “Surfing the Data Pipeline with Python”, chapter 10
- Live session (Zoom link on Canvas):
  - Wednesday, July 23, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, July 29, 11:59pm

Module 11: Static Visualizations

- Readings:
  - Textbook: Molin "Visualizing Data with Pandas and Matplotlib”, "Plotting with Seaborn and Customization Techniques" https://www.oreilly.com/library/view/hands-on-data-analysis/9781789615326
  - Textbook: Wilke, chapters 2, 17, 29 https://serialmentor.com/dataviz/
  - “Surfing the Data Pipeline with Python”, chapter 11
- Live session (Zoom link on Canvas):
  - Wednesday, July 30, 7:15-8:15pm
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, August 5, 11:59pm

Module 12: Interactive Visualizations and Dashboards

- Readings:
  - Browsing the Plotly Gallery to see what is possible and how to code different graphs: https://plotly.com/python/plotly-fundamentals/
  - Working through the Dash tutorial: https://dash.plotly.com/installation
  - Some thoughts on how to make an effective UX design: https://www.toptal.com/designers/data-visualization/dashboard-design-best-practices
  - "Surfing the Data Pipeline with Python", chapter 12
- Live sessions (Zoom link on Canvas):
  - Wednesday, August 6, 7:15-8:15pm
- Lab assignment due dates (no reading quiz or live coding assignment this week):
  - Tuesday, August 12, 11:59pm
  - IMPORTANT: No late assignments can be accepted for module 12 because of UVA's summer semester grade deadline.