UNIVERSITY OF VIRGINIA SUMMER 2022

DS 6001: Practice and Application of Data Science

Tuesdays and Thursdays 1:00 - 3:45pm, Ridley G004 & Dell 1

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

Teams Channel
#ds6001

Office Hours
Over Teams: Send me a message anytime

Over Zoom: Thursdays, 7-9pm. Link on Collab

Email
jkropko@virginia.edu

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

1

PYTHON 3

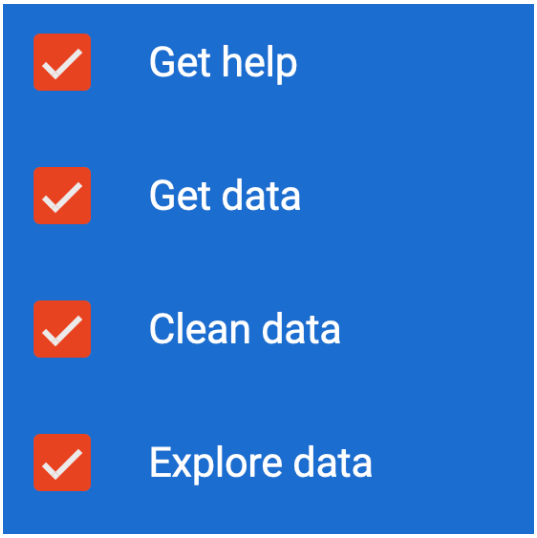
Available from the Python Software Foundation: https://www.python.org/downloads/

2

ANACONDA NAVIGATOR

User interfaces for Python: https://www.anaconda.com/products/individual/

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 demonstrating how to be a good citizen in online forums
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, and plotly

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 Spyder (both available through Anaconda Navigator), or open a command terminal, and type:

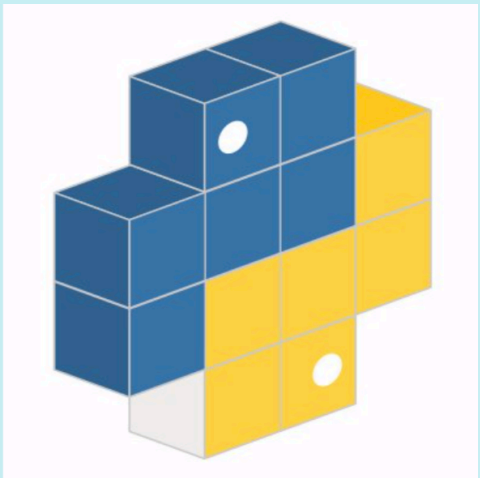
pip install pandas

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.

Class Meetings: Live Coding Demonstrations

We will devote the first hour of each class to a live coding demonstration that engages with real-world challenges that will arise in any data project outside and beyond analytic models. 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 submit your script or notebook on Collab. 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. We will meet in the official classroom for the live coding part of class, then we will move to the Dell 1 common space to give you time to work on the lab assignment.

Class Meetings: Paired Programming

This course is divided into 12 modules, and each module has a lab assignment that you are responsible for completing. The second part of each class will be devoted to giving you time to work on these assignments. We will use the common space in Dell 1, as this space works better for collaborative work. I will ask you to work in pairs and to help each other with the assignment. Paired programming is a technique that is common in agile software development workflows. In a paired programming session, the two individuals take on the roles of "driver" (the person writing code) and "observer" (the person commenting on the code and staying focused on the goals), and switch roles

frequently. You are free to choose any partner you want, but I will ask you not to work with the same person for more than one module. You are free to share code with your partner so long as it for the purposes of collaboration and not for avoiding doing the work, but each person is responsible for their own assignment, and any text-based answers must be originally written by each person (no copying text from one another). Please see the section on Collaboration and Cheating below for more specific guidelines on acceptable collaboration.

Course Website and Equipment

All grades, labs, and readings will be posted on the UVa Collab site for the course, accessible at https://collab.itc.virginia.edu/portal. 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 360 possible points in this course. Your grade will be determined by:

- Your performance on twelve **lab assignments** (20 points each, 240 points total, 66.7% of the grade)
- Your performance on twelve **reading quizzes** (5 points each, 60 points total, 16.7% of the grade)
- Your submissions for ten **live coding sessions** (6 points each, 60 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
> 93%	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 code 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 Collab 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.

Microsoft Teams

Microsoft Teams is a web-based service that provides message boards to organizations. We've started a Teams channel for our course, and **I intend for this channel 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 a Teams page 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 can be used for private messages and for public class discussion. To send me a message, find "Chat" on the left, click on the new message button, and type Jonathan Kropko in the recipient bar. 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, "channels" are different message boards for different topics, visible to all class participants. We've created the **#ds6001** channel 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.

Collaboration and Cheating

I wrote the following section of the syllabus in class with a previous cohort of DS 6001 students. We all agreed that we wanted to avoid a situation in which students' work had to be policed and treated with skepticism. We had a frank discussion about where the line between collaboration and cheating exists, and we came up with the following guidelines together:

Cheating tends to happen at higher rates in introductory programming-based courses because students get frustrated when their code won't run, because of the feeling that there is only one correct way to write the code, and because of how easy it is to copy and paste a few lines of someone else's code. Cheating also happens at higher rates during high pressure situations, such as a course like this one that is taught on a rigid timeframe.

Although every student is responsible for their own lab reports, you may chat and Zoom with one another to work together on labs. 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**:

1. Directly copying someone else's text word for word

2. 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)

3. Asking for help without doing anything to try to solve the problem first; asking someone to do the work for you

4. Making your homework freely available on GitHub or another website, or posting answers indiscriminately on Teams, Slack, or another message board visible to other students

5. Sharing answers to reading quizzes

Here are things that are okay:

1. Two people with the same code is okay as long as one person didn't copy-and-paste it from another person

2. Sharing/showing individual lines of code for the purpose of teaching/explaining or helping someone understand the material

3. Debugging together (this is only possible if both people have already written their own code, otherwise there's nothing to debug)

4. Sharing strategies, external texts, blogs, and other resources for completing problems on the lab assignments

There are many ways in which I am limited in my ability to enforce these rules, but I ask you to promise on your honor to not to share code or quiz answers. Cheating means that you do not give yourself the opportunity to master the skills to start working with data in Python. Why rob yourself? If you are stressed out about the intensity of the course, please message me and we can work together to get you back on track.

Schedule, Readings, Important Dates

Module 0: Getting Started with Python and JupyterLab (optional: at your convenience prior to class)

- Ungraded introductory lab: https://colab.research.google.com/drive/1oMEcZVC0P-VUGwLf-72XvHAPOoXEUkPh?usp=sharing

Module 1: Getting Yourself Unstuck (one class session)

- 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
- Class session: Tuesday, June 21
  - Course Introduction: 1:00-2:00pm, Ridley G004
  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space
- Reading quiz and lab assignment due dates (no live coding assignment for this module):
  - Sunday, June 26, 11:59pm

Module 2: Working with Electronic Data Files (one class session)

- 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
- Class session: Thursday, June 23
  - Live Coding Session: 1:00-2:00pm, Ridley G004

  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space

- Reading quiz, lab assignment, and live coding assignment due dates:

  - Tuesday, June 28, 11:59pm

Module 3: Working with JSON Data (one class session)

- 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

- Class session: Tuesday, June 28

  - Live Coding Session: 1:00-2:00pm, Ridley G004
  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space

- Reading quiz, lab assignment, and live coding assignment due dates:

  - Sunday, July 3, 11:59pm

Special Class: Guest Speakers Tell Us How They Built Their Data Pipelines (one class session)

- Class Session: Thursday, June 30, 1:00-3:45pm, Ridley G004
- No reading quiz, lab assignment, or live coding assignment for this class session

Module 4: Working with APIs in Python (one class session)

- 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

- Class session: Tuesday, July 5
  - Live Coding Session: 1:00-2:00pm, Ridley G004
  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Sunday, July 10, 11:59pm

Module 5: Web-scraping with BeautifulSoup (one class session)

- 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
- Class session: Thursday, July 7
  - Live Coding Session: 1:00-2:00pm, Ridley G004
  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, July 12, 11:59pm

Module 6: Databases in Python (two class sessions)

- 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

- Class sessions: Tuesday, July 12 and Thursday, July 14
  - Live Coding Sessions: 1:00-2:00pm, Ridley G004
  - Paired Programming Sessions: 2:15-3:45pm, Dell 1 Common Space
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Sunday, July 17, 11:59pm

Module 7: Database Queries (two class sessions)

- 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
- Class sessions: Tuesday, July 19 and Thursday, July 21
  - Live Coding Sessions: 1:00-2:00pm, Ridley G004
  - Paired Programming Sessions: 2:15-3:45pm, Dell 1 Common Space
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Sunday, July 24, 11:59pm

Module 8: Data Cleaning with Pandas (one class session)

- 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
- Class session: Tuesday, July 26

- Live Coding Session: 1:00-2:00pm, Ridley G004
- Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space

- Reading quiz, lab assignment, and live coding assignment due dates:
  - Sunday, July 31, 11:59pm

Module 9: Merging and Reshaping Dataframes in Pandas (one class session)

- 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
- Class session: Thursday, July 28
  - Live Coding Session: 1:00-2:00pm, Ridley G004
  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space
- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, August 2, 11:59pm

Module 10: Exploratory Data Analysis (one class session)

- 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
- Class session: Tuesday, August 2

- Live Coding Session: 1:00-2:00pm, Ridley G004
- Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space

- Reading quiz, lab assignment, and live coding assignment due dates:
  - Sunday, August 7, 11:59pm

Module 11: Static Visualizations (one class session)

- 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

- Class session: Thursday, August 4
  - Live Coding Session: 1:00-2:00pm, Ridley G004
  - Paired Programming Session: 2:15-3:45pm, Dell 1 Common Space

- Reading quiz, lab assignment, and live coding assignment due dates:
  - Tuesday, August 9, 11:59pm

Module 12: Interactive Visualizations and Dashboards (two class sessions)

- 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

- Class sessions: Tuesday, August 9 and Thursday, August 11

    - Live Coding Sessions: 1:00-2:00pm, Ridley G004
    - Paired Programming Sessions: 2:15-3:45pm, Dell 1 Common Space
- Lab assignment due dates (no reading quiz or live coding assignment this week):
    - Sunday, August 14, 11:59pm