

Data Analytics Boot Camp - UCI DCE

Unit Overviews

- [2.1 - 2.3: VBA Scripting](#)
- [3.1 - 3.3: Python \(& Git\)](#)
- [4.1 - 4.3: Pandas](#)
- [5.1 - 5.3: Matplotlib](#)
- [6.1 - 6.3: Python APIs](#)
- [7.1 - 7.3: Project 1](#)
- [8.1 - 8.3: Project 1](#)
- [9.1 - 9.3: SQL](#)
- [10.1 - 10.3: Advanced Data Storage and Retrieval](#)
- [11.1 - 11.3: Web](#)
- [12.1 - 12.3: Web Scraping and Document Databases](#)
- [13.1 - 13.3: ETL Project](#)
- [14.1 - 14.3: Introduction to JavaScript](#)
- [15.1 - 15.3: Interactive Visualizations and Dashboards](#)
- [16.1 - 16.3: D3](#)
- [17.1 - 17.3: Leaflet.js & GeoJSON](#)
- [18.1 - 18.3: Projects & R](#)
- [19.1 - 19.3: Project 2](#)
- [20.1 - 20.3: Tableau](#)
- [21.1 - 21.3: Machine Learning](#)
- [22.1 - 22.3: Big Data](#)
- [23.1 - 24.3: Final Projects](#)

2.1 - 2.3 VBA Scripting

Great job on showing your GRIT! Keep up the great work!

Unit Objectives:

- Understand the fundamental building blocks of all programming languages: variables, arrays, conditionals, loops, and functions.
- Create simple VBA macros to trigger pop ups and change cell values.
- Gain practice in writing VBA subroutines that utilize variables and conditionals.
- Begin to develop essential coding skills of syntax recollection, pattern recognition, problem decomposition, and debugging.
- Understand the basic syntax of a VBA for loop.
- Understand how to utilize for-loops in conjunction with conditionals to direct logic flow.
- Understand the value of a nested for-loop and gain basic proficiency in their use.
- Refine fundamental coding skills (syntax recollection, pattern recognition, problem decomposition, and debugging).
- Be comfortable formatting spreadsheets using VBA code.
- Understand how to loop through a table using VBA code and check for changes in values.

Frequently Asked Questions:

What's the relevance of VBA?

You can consider VBA to be an introduction into the basics of programming that will become more relevant later in the course. VBA is also a great tool you can use to automate the data manipulation that we do in Excel. VBA is also prevalent in many older companies and in the financial industry. Many of these companies will value your ability to manipulate their legacy data in Excel with VBA.

Helpful Resources:

- [Excel VBA Programming](#)

Next Up: Python (& Git)!

Python resources:

- [Beginner Python](#)

If anyone wants to start learning Git, see resources below:

- [GitHub Challenges](#)
- [Learn Git](#)

3.1 - 3.3 Python (& Git)

Congratulations on finishing your first 2 weeks in the data bootcamp!

For those that sense they are struggling, remind yourselves that it's OK to feel this way. This is a bootcamp and we'll be working hard for these next few months to learn a lot of new technologies! Embrace the struggle!

At the same time, be aware of negative thoughts you might put on yourselves; don't let [imposter syndrome](#) get the best of you in this class, and make sure to celebrate your small successes each week!

However, we do realize there is a difference between struggling and completely falling behind. If you find you are completely behind at this point, please don't hesitate to reach out to me or your instructional staff. Keep in mind the success principles we outlined when this class started. Ask yourself the following questions:

- Are you reviewing class recordings and activity videos regularly? Repetition is key!
 - Are you using office hours for TA guidance?
 - Are you putting in at least 20 hours per week outside of class of practice?
-

Unit Summary:

This week we will be introducing the Python programming language. The purpose of this unit is to gain a firm understanding on how to use the core Python language to store and manipulate data; read and write to files; and "translate" logical solutions to problems to Python implementations of those solutions.x

This unit will start out with core features of Python, such as variable declarations, and move on to discuss more advanced and idiomatic features of the language. Topics will include reading and writing files, working with the file system, iteration using for loops, iteration using list comprehension, list methods, the notion of dictionaries and methods on them, and additional miscellany where required to facilitate completion of the homework assignment.

Make sure that you have all of your required course tools installed. Check the [pre-work](#) for anything you might have missed.

Unit Objectives:

- Be able to navigate the desktop via the terminal.
- Create Python scripts and run them in terminal.
- Begin to understand programming concepts in Python.
- Feel confident reading data into Python from CSV files.
- Feel confident writing data from Python into CSV files.
- Know how to zip two lists together and when this is helpful.

- Have a firm understanding on how to create and use Python functions.
- Be able to create and use Python dictionaries.
- Be able to read data in from a dictionary.
- Have a firm understanding of coding logic and reasoning.

Frequently Asked Questions:

What's up with this crazy indentation?

With Python, indentation is more than just organization and readability. Python's functionality actually depends on proper indentation!

In this snippet, we're using indentation to tell our code where our for loops begin and end. In Python, indenting creates blocks of code that work together. Similarly, indenting backwards tells the program when to end a loop.

```
for x in range(10):  
    print(x)  
  
for x in range(20, 30):  
    print(x)
```

The code you will write in Python will eventually be seen by someone else. Focusing on organization and readability is important because you want colleagues to be able to read your code. If your code is poorly organized it will be difficult to read later on.

Helpful Resources:

- [Python - Beginner](#)
- [Python - Scripting](#)
- [Git - No-nonsense Git Guide \(Videos 1-3\)](#)
- [Git - Visual Guide](#)

Next Up: Pandas

Helpful Resources:

- [Pandas Tutorials](#)

Have a nice week!

4.1 - 4.3 Pandas

Great job finishing out the Python unit! We know it's a lot of information and it feels like we're moving quickly through it, so make sure to invest the time in your education and keep up with your written study schedule of at least 20 hours outside of class per week. This will be especially important in this unit.

There is a lot of material to cover and only 24 weeks to get through it, but know that you have an entire support team to help you out! Whenever you are feeling like you're falling behind, ask your instructional team for help. Ask your fellow students if you're comfortable! We're all in this together and everyone is invested in your success. If you feel comfortable with the material and have the opportunity to help out a fellow classmate, go for it! Teaching someone else will help solidify the concepts you've learned.

Even though we're moving through things quickly, we have the opportunities to practice the concepts for many more weeks. By the end of the program, you'll realize just how much your effort has paid off. Keep up the effort to learn and focus on the tasks at hand!

I know that the Python unit was fast, but please note: **Unit 4 can be one of the most challenging in the course.** We're expanding on your knowledge of Python and adding in a package with lots of new methods and functions that will make working with our data easier in the long run. Hang in there! After this week, units 5-7 will slow down significantly and you'll feel more comfortable leading into the first project.

Unit Objectives:

- Be able to serve Jupyter notebook files from local directories and connect to their development environment.
- Be able to create Pandas DataFrames from scratch.
- Understand how to run functions on Pandas DataFrames.
- Know how to read/write DataFrames from/to CSV files using Pandas.
- Understand how to navigate through DataFrames using Loc and Iloc.
- Understand how to filter and slice Pandas DataFrames.
- Understand how to create and access Pandas GroupBy objects.
- Understand how to sort DataFrames.
- Know how to merge DataFrames together whilst understanding the differences between inner, outer, left, and right merges.
- Be able to slice data using the cut() method and create new values based upon a series of bins.
- Feel more confident with fixing Python/Pandas bugs within Jupyter Notebook.
- Be able to use Google to explore additional Pandas functionality when necessary.

Helpful Resources:

- [Pandas Cheatsheet](#)
- [Pandas Tutorials](#)
- [Pandas Documentation](#)
- [Pandas Guide to Merging](#)
- [Visual Guide to Joins](#)

Frequently Asked Questions:

When should we use Pandas versus just using Python?

Python is a scripting language, while Pandas is a library that works on top of Python and makes data manipulation easier.

Libraries are pre-written chunks of code that make it easier to unlock the capabilities of a programming language. Someone else already did a ton of hard work to create these libraries and make your life easier and your workflow more efficient.

It would be a slog to manually go through all of your data with vanilla Python. Pandas makes this process much quicker.

You can think of it like what it's like using Excel vs using VBA with Excel.

Next Up: Matplotlib

Helpful Resources:

- [Storytelling With Data - Alternatives to Pie Charts](#)
- [Matplotlib Gallery](#)

Have a nice week!

5.1 - 5.3 Matplotlib

We'll be going over Matplotlib this unit, which is a 2D plotting library using the technologies that you've learned in class so far. We'll be able to truly tell stories using our data in visually compelling ways using some of the code covered in class.

This week will be a bit slower in pace than the last couple of weeks, but don't cheat yourself out of your at-home study time! Continue investing your time and energy into your education and practice any material you feel less confident in.

Quick reminder! Audit your attendance and homework submissions. Remember graduation guidelines and ensure you're on track. If you see any discrepancies in Bootcamp Spot (BCS), let me and your instructional team know.

Unit Objectives:

- Understand Matplotlib's pyplot interface.
- Be able to create line; bar; scatter; and pie charts.
- Be familiar with basic plot configuration options, such as `xlim` and `ylim`.
- Feel comfortable creating plots using the `DataFrame.plot()` method.
- Understand the advantages and disadvantages of creating charts using the `DataFrame.plot()` method.
- Be able to work through a complex data set using Pandas and then chart some visualizations based upon the cleaned DataFrame.
- Be able to define mean, median, and mode, and choose which one is most appropriate to describe a given data set.
- Be able to explain the meaning of variance and standard deviation.
- Be able to describe standard error and the difference between a sample and a population.
- Be able to add error bars to plots.
- Be able to fit lines to data.

Frequently Asked Questions:

I'm having trouble picking up on this new syntax...

Luckily, there is documentation for that! Whenever we work with a library, we are using code that is already pre-written. We always need to reference the documentation when using a new library. Think of documentation as a set of detailed instructions. Without looking at the instructions, you will never know how to use a new piece of technology.

We recognize this may be your first time looking at a library that is less semantic, or readable, but this will be a constant for the future. You will see a lot of new technology throughout the course, and each time you will need to dive into the documentation. The documentation quality

and organization will be different for each new technology, so learning how to read general documentation is a skill in itself.

Helpful Resources:

- [NumPy](#)
- [Matplotlib Gallery](#)
- [Pandas Plotting](#)
- [Storytelling with Data - Alternatives to Pie Charts](#)

Next Up: Python APIs

Helpful Resources:

- [JSON Testing](#)
- [OMDb API](#)
- [The New York Times API](#)

Have a nice week!

6.1 - 6.3 Python APIs

This week we will cover using third party APIs with Python. We will programmatically obtain and parse data from sources such as OpenWeatherMap, the US Census, OMDb, and more. We will also plot Data Frames from the API data using Matplotlib. This is one of the ways that we're going to be able to source outside data to use for our projects, and a great opportunity to practice working with external data. Documentation is key! Know that each API we use will have a different set of rules and guidelines and half the fun is learning all the features each one can provide.

Unit Objectives:

- Be able to make GET requests with `requests`.
- Be able to convert JSON into a Python dictionary.
- Read and apply API documentation.
- Sign up for and use an API key.
- Create applications from scratch using nothing but knowledge of Python and an API documentation.
- Load JSON from API responses into a Pandas DataFrame.
- Be able to use `try` and `except` blocks to handle errors.
- Successfully use the Google Maps and Places API to obtain information about geographic areas.
- Understand how to use the Census API wrapper.
- Understand the concept of rate limits and the importance of creating "test cases" prior to running large scripts.
- Have a firmer understanding of how to dissect new API documentation.

Frequently Asked Questions:

How do I differentiate between the client and the server?

The web applications we build in class (and all other web apps) will live on a server, which is basically a gigantic computer or network of computers. When a user goes into their browser and types in an address, the server sends information to the user's computer, or the client.

Basically: Clients request from the server, and the server returns a response.

Where can I find more information on how to use these APIs?

Each API is a pre-written set of code that helps you interact with somebody else's information. Every API will come with its own set of documentation, and much like with a library, you will have to dive into it! There should be a developer section on each site that will explain the service and any potential limitations.

Any time we work with an API, it's a good idea to have the documentation ready and refer to it frequently. This is all part of the process when it comes to picking up and familiarizing yourself with new technologies.

Helpful Resources:

- [JSON Testing](#)
- [OMDb API](#)
- [The New York Times API](#)
- [Open Weather Map API](#)

Next Up: Project #1

Have a nice week!

7.1 - 7.3 Project 1

This week we will be diving into Project 1! This is an exciting time where your class can collaborate on a project together using the technologies that have been covered in class. Check out the linked resources on Git, as Git can be a challenging technology to truly feel comfortable in until you've worked with it for a while. Remember that constant communication with your team is incredibly helpful, and practicing the workflow that's provided in the video instruction a few times on your own can help start you off on a stronger foundation for projects.

Unit Objectives:

- Students will be able to articulate the requirements for Project 1.
- Students will be able to draw and interpret diagrams of Git branching workflows.
- Students will be able to create new branches with Git.
- Students will be able to push local branches to GitHub.
- Students will be able to pull a branch from GitHub.
- Students will be able to merge branches with Git.
- Students will be able to open, review, and merge PRs with GitHub.
- Students will resolve merge conflicts in their working copy.
- Students will push branches to GitHub.
- Students will be able to open a PR against a given branch.

Frequently Asked Questions:

I'm not feeling confident with Git... What are some resources that could help?

Start with our pre-work, which has two modules with video tutorials and activities that can help out with the foundations of working with Git: [Bootcamp Pre-work](#)

Conveniently, GitHub has their own set of guides to help break down how to use the program: [GitHub Guides](#)

Additionally, if you find videos helpful in your learning process, this is roughly an hour of video designed to cover the fundamentals of Git and GitHub: [No Nonsense Github Videos](#)

Finally, we have a handy visual Git reference guide located here: [Visual Git Reference](#)

How will this project be beneficial?

Each project that we tackle in this class serves multiple purposes, and they will all be highly critical in your development as a data professional. First, group projects allow you to start developing a collaborative workflow that is very similar to the workflow you would encounter at many jobs in the field. Good teamwork is an essential real-world skill. Working in a group also allows you to accomplish more and create more impressive projects than you would be able to alone! Take advantage of it!

This project will be your first opportunity to build a data presentation from the ground up. This is something you will do many times as a data professional, so the more you practice and reinforce these skills, the better off you will be in the field!

Finally, this project is going to be featured in your resume and portfolio. Having strong examples of your work in your professional materials increases the chances to wow a potential employer and ultimately get hired!

Helpful Resources:

- [Git - No-nonsense Git Guide \(Videos 1-5\)](#)
- [Git - Visual Guide](#)

Next Up: Project #1 Continued

Have a nice week!

8.1 - 8.3 Project 1

This week we will be continuing Project 1! Be sure that you're staying focused on structured tasks to ensure you complete your project by the due date. Keep each other accountable by checking in with each other on a frequent basis, not just in class. Use your Slack workspace to your advantage and keep in communication to build a solid project with your team.

Before your team presents, make sure you know who is speaking to what topic and ensure that you practice storytelling with your data!

Helpful Resources:

- [Git - No-nonsense Git Guide \(Videos 1-5\)](#)
- [Git - Visual Guide](#)

Next Up: SQL

Helpful Resources:

- [PostgreSQL Documentation](#)
- [pgAdmin Documentation](#)
- [PostgreSQL Tutorial](#)

Have a nice week!

9.1 - 9.3 SQL

Congratulations on completing your first Data Project! All your hard work is definitely paying off and you'll continue to grow your presentation skills and abilities to tell stories with your data. We'll be diving into SQL this week!

Unit Objectives:

- Create a localhost connection to a PostgreSQL server and have successfully connect to it.
- Create, use, and populate a SQL database with data.
- Create, populate, and select data from a SQL table.
- Import large CSV datasets into pgAdmin.
- Use pgAdmin to select specific rows/columns of data out from a table.
- Understand the different kinds of joins and how to use them to create new tables in pgAdmin.
- Solidify the foundations of writing basic- to intermediate-level SQL statements.
- Develop an introductory understanding of table design and database management.

Frequently Asked Questions:

What's the relevance of PostgreSQL?

PostgreSQL, or Postgres, is a really cool free, open-source object-relational database management system. Databases are going to be crucial in our work as data professionals! PostgreSQL is also most popular with some of the careers that our course is helping prepare you for.

Remember how we discussed the relationship between client and server earlier? We can think of our database as the third part of that relationship. Essentially, the client sends a request to the server, the server grabs the associated data from the database, and then the retrieved data is returned to the client by the server.

But what does SQL mean?

SQL stands for Structured Query Language. Put simply, SQL is the language that we use to interact with our database. SQL is the most widely used database language, is incredibly effective at manipulating data, and is one of the easier languages to learn! Learning SQL is also great for employability - There are thousands of jobs that look for candidates with SQL experience!

Helpful Resources:

- [PostgreSQL Documentation](#)
- [pgAdmin Documentation](#)

- [PostgreSQL Tutorial](#)

Next Up: Advanced Data Storage and Retrieval
Helpful Resources:

- [Introduction to SQLAlchemy](#)

Have a nice week!

10.1 - 10.3 Advanced Data Storage and Retrieval

Congratulations on completing the SQL unit! This week, we'll be working on using your SQL skills with a new toolkit to store and retrieve datasets.

Unit Objectives:

- Connect to a SQL database using SQLAlchemy.
- Perform basic SQL queries using engine.execute().
- Create Python classes and objects.
- Create, read, update, and delete data from a SQL database using SQLAlchemy's ORM.
- Reflect existing databases.
- Use the SQLAlchemy ORM to create classes that model tables.
- Use the ORM define relationships and foreign key constraints.
- Use joins to query related data.
- Extract query variable path values from GET requests.
- Use variable paths to execute database queries on behalf of the client.
- Return JSONified query results from API endpoints.

Frequently Asked Questions:

What is SQLite and why use it?

SQLite is a dialect for using an SQL database. The syntax is very similar to the PostgreSQL syntax, with the main difference between the two being that SQLite is entirely serverless. SQLite allows for the power of database storage but with a tiny footprint. SQLite is also conveniently part of the standard Python library, so no additional setup is needed to get it up and running.

SQLite is not comparable with PostgreSQL or any other SQL language, rather it focuses on providing local data storage for individual applications.

What does ORM mean?

ORM stands for Object-Relational Mapper. Object-relational mapping allows us to write SQL queries using the object-oriented paradigm of the language with which you prefer to work. In other words, for our purposes in class, it allows us to interact with our database using Python.

How are classes relevant?

As you already know, Python is an "Object-Oriented Programming (OOP) language", which means that it is highly concerned with organization and reusability. Classes are crucial to OOP in that they allow us to group related things and keep them together.

A class is essentially a template that allows us to create objects, which have variables and behaviors associated with them. It helps streamline our code and create efficiencies whenever something needs to be used various times.

Helpful Resources:

- [Introduction to SQLAlchemy](#)
- [Essential SQLAlchemy Book](#)

Next Up: Web

Helpful Resources:

- [CodeCademy HTML & CSS](#)
- [Mozilla HTML Docs](#)

Have a nice week!

11.1 - 11.3 Web

We're switching things up this unit! In this unit we will cover the anatomy of a website. Students will build websites using HTML and add styles to them with CSS. Students will also learn to use the Bootstrap CSS framework and deploy their websites to Github Pages. Learning these technologies will make it easier for you to display your data for others, and the skills will translate to other technologies we will be covering later in the course.

Unit Objectives:

- Gain a high-level understanding of HTML, CSS, and JavaScript and what their roles are when creating websites.
- Understand the basic parts of an HTML web page and how to create one from scratch.
- Learn to cover and utilize some of the most common HTML tags and selectors.
- Understand how to deploy HTML web pages to the internet using GitHub Pages.
- Understand the basics of CSS styling.
- Position HTML elements on a webpage using CSS.
- Be able to discuss media queries, the technology that is used to create the responsive Bootstrap grid.
- Understand the Bootstrap Grid and discover how to utilize it to position the elements on the page.
- Discover how to quickly and easily build web pages using pre-built Bootstrap components.

Frequently Asked Questions:

What's the relevance of using web technologies like HTML and CSS?

Learning these web technologies teaches us an entirely new way to present our data: in a beautiful web application! Having an impressive GUI with which to present your findings can truly take your work to the next level, and looks great to employers and stakeholders!

Helpful Resources:

- [CodeCademy HTML & CSS](#)
- [Mozilla HTML Docs](#)
- [GitHub Pages](#)
- [Bootstrap 4 Tutorial](#)

Next Up: Web Scraping and Document Databases

Helpful Resources:

- [Mongo in 30 Minutes](#)

Have a nice week!

12.1 - 12.3 Web Scraping and Document Databases

This week we'll be learning how to use the data we find on the web in our applications using web scraping.

Unit Objectives:

- Create and connect to local MongoDB databases.
- Create, read, update, and delete MongoDB documents using the Mongo Shell.
- Create simple Python applications that connect to and modify MongoDB databases using the PyMongo library.
- Use BeautifulSoup to scrape their own data from the web.
- Save the results of web scraping into MongoDB.
- Use BeautifulSoup to scrape data.
- Use PyMongo to save data to a Mongo database.

Frequently Asked Questions:

Why are we using Mongo now? I thought SQL was the best!

While SQL is an extremely powerful language for defining and manipulating data, it can actually be restrictive. SQL requires you to use predefined schemas to determine your data structure, and requires that all of your data follow the same structure.

A NoSQL database has a dynamic schema for unstructured data and can store data in a variety of ways. This gives it much more flexibility in practice as not all data will come structured in the same way.

Helpful Resources:

- [Mongo in 30 Minutes](#)
- [Web Scraping with BeautifulSoup](#)

Next Up: ETL Case Study Project

Have a nice week!

13.1 - 13.3 ETL Project

This week should be exciting! It's another opportunity to use the skills you've gained over the last few months to create a project as part of a small team. It's a change to synthesize what you've learned over time with new material while using real-world data sets.

Remember to maintain scope while you work on this project. You have only one week to deliver your final product, so hold yourself and your teammates accountable to the project timelines.

Unit Objectives:

- Deliver a finalized group project.
- Understand the benefits and challenges of the ETL process.

Frequently Asked Questions:

Why are we doing this project?

ETL, or Extract, Transform, Load, is a critical aspect of working with data. With ETL, we are essentially pulling in data from various sources, cleaning and restructuring it, and then writing it into a database for storage. This workflow is something you will see time and time again as a data professional. It will be an opportunity to see everything built from the ground up in class.

Next Up: Intro to JavaScript

Helpful Resources:

- [You Don't Know JS \(book series\)](#)
- [JavaScript Tutorial](#)

Have a nice week!

14.1 - 14.3 Introduction to JavaScript

This is the beginning of a few weeks of JavaScript related content! It can be a very exciting time, but learning something this new and complex can also be frustrating. Make sure that you're investing time in yourself and your education that you deserve. Focus on meeting those 20 hours outside of class that you agreed to when you first enrolled, utilize office hours to your advantage, and ask questions! The JavaScript foundation we will be building this week will help in your future exploration of new tools to create visualizations of data that your storytelling needs.

Unit Objectives:

- Understand JS fundamentals: arrays, conditionals, loops, functions, objects.
- Understand functional programming with map, forEach.
- Work with common data structures.
- Be introduced to data driven documents (d3.js).
- Understand how to select elements using d3.select.
- Use d3 for basic DOM manipulation.
- Understand how to use callbacks.
- Understand the structure of html tables.
- Populate a table using static data structures.
- Understand events.
- Use d3 to attach events to DOM elements.
- Dynamically manipulate the DOM through events.
- Filter data with JavaScript.

Frequently Asked Questions:

Oh no! More new syntax that I don't understand!?

Don't be afraid! Learning new syntax can certainly feel like learning a whole new language. And that's what we're doing! We don't expect you to be JavaScript masters! All of this information is iterative and your skills will only improve with time and practice. If it becomes overwhelming take advantage of office hours to spend time with the instructional staff to help get you over the hill. Please don't be afraid to reach out for help! We are all here for your success.

Why do we need to learn JavaScript? I thought this was a Python course...

Learning how to build functional web applications with HTML, CSS, and JavaScript is going to open up a world of possibilities for presenting our data! Many of the libraries that we will be using for impressive visualizations are JavaScript libraries, making it an essential skill for some of our more interactive visualizations.

Learning JavaScript also gives you another real world skill to add on to your resume that employers love!

Helpful Resources:

- [Interactive JavaScript Sheet](#)
- [Scrimba Intro to JavaScript](#)
- [You Don't Know JS \(book series\)](#)
- [JavaScript Tutorial](#)

Next Up: Interactive-Visualizations and Dashboards

Helpful Resources:

- [Plotly.js Getting Started Guide](#)

Have a nice week!

15.1 - 15.3 Interactive Visualizations and Dashboards

Great job with your first week of JavaScript content! I know it can be overwhelming to be in a fast-paced bootcamp environment, but trust that the technologies that you are learning will come together to make you a stronger storyteller with data. Ensure that you're dedicating time and energy to your at-home practice and reviewing the class recordings, and never be afraid to ask questions during class.

Unit Objectives:

- Use Plotly to create the fundamental charts: Box, scatter, bar, pie, and line plots.
- Use the Plotly `layout` object to customize the appearance of their charts.
- Annotate charts with labels; text; and hover info.
- Create and manipulate advanced Plotly charts.
- Create bubble charts to visualize three-dimensional data.

Frequently Asked Questions:

Why would I use Plotly instead of Matplotlib or another library?

While Plotly.js is not necessarily "better" than Matplotlib or another plotting library, it is useful for a variety of reasons! Plotly is also incredibly popular in the data community.

Plotly.js uses both D3.js and WebGL to render graphics, works with JSON schema, and supports a large variety of map charts, including basic, statistical, financial, and scientific!

Helpful Resources:

- [Plotly.js Getting Started Guide](#)

Next Up: D3

Helpful Resources:

- [D3 Gallery](#)
- [D3 Tutorial](#)

Have a nice week!

16.1 - 16.3 D3

This week we'll be diving more deeply into D3 and expanding our knowledge in this library to start building bar charts, line charts, scatterplots, and more. D3 is a very deep well, so we'll only be scratching the surface of what this library can truly do. Take some time to look through the gallery of D3 projects linked below to see what kinds of things can be built using this library and some JavaScript.

Unit Objectives:

- Gain a high-level understanding of SVG elements and how to append/modify them using D3.
- Understand how to bind data to SVG elements using D3 so as to create basic bar charts from scratch.
- Create a bar chart with axes using D3 so as to visualize data.
- Create different types of charts and graphs using D3.
- Cover scales in greater depth.
- Plot multiple columns from a dataset, either simultaneously or in alternation.
- Gain a better understanding of reusable code.

Frequently Asked Questions:

This seems like a really difficult library, I'm worried that it will never make sense...

D3 is certainly an intimidating library, but it is also one of the most powerful Data Visualization libraries available to us! It's a deep well of knowledge, and while we're only tapping into a fraction of it for this unit, you have the tools to expand your understanding after we complete this unit.

Just like with any other skill, your knowledge of D3 will only strengthen with time and practice. Take advantage of the resources available to you, ask the instructional staff and your fellow students for help if you need it, and don't be afraid to get your hands dirty and make mistakes! Encountering and solving errors is all part of the process, and will only make you a better programmer.

Helpful Resources:

- [D3 Gallery](#)
- [D3 Tutorial](#)
- [Scrimba D3.js Tutorial](#)
- [D3 Official Website](#)

Next Up: GeoJSON & Leaflet

Helpful Resources:

- [Leaflet Documentation](#)
- [MapBox API](#)

17.1 - 17.3 Leaflet.js & GeoJSON

I hope you had fun expanding your knowledge in D3! It's an expansive library with a lot of room to grow your skill set in, which can be very exciting for future projects. This week, we'll be going through Leaflet.js and working with visualizing data with maps. Your toolkit for showing off data in a meaningful way is growing!

Unit Objectives:

- Understand the benefits that visualizing data with maps can provide.
- Learn the basics of creating maps and plotting data with the Leaflet.js library.
- Gain an understanding of the GeoJSON format.
- Understand the concept of layers and layer controls and how we can use them to add interactivity to our maps.
- Gain a firm grasp of mapping with GeoJSON.
- Learn about and practice using Leaflet plugins and third-party libraries.
- Learn how different maps can effectively visualize different datasets.
- Gain a Leaflet mastery by completing an in-class project.
- Learn the basics of creating maps with CARTO, including writing custom CSS and SQL queries to style and filter data, while also incorporating multiple data sets within the same map.
- Understand how different types of maps are better for visualizing different datasets.

Frequently Asked Questions:

Why is it useful to map out data geographically?

Your datasets should be visualized and explained in different ways depending on the background context of the data and what your stakeholders are most interested in learning. Oftentimes, the geography in the data is highly relevant. For example, a marketing stakeholder would be interested to know the popularity of their product in specific states to understand how to best move forward with their campaigns.

Helpful Resources:

- [Leaflet Documentation](#)

Next Up: Introduction to R

Helpful Resources:

- [R for Data Science \(book\)](#)

Have a nice week!

18.1 - 18.3 Projects & R

This week we will be focusing on R and ending with the beginning of your next project! You'll notice that R and Python have a lot of overlap. These similarities will likely make it easier for you to grasp this new syntax, which is something you'll find as you learn new languages in the future. Check out RStudio and the linked tutorial to familiarize yourself with the directory structure outside of class, and feel free to rely on the graphical interface as well.

Unit Objectives:

- Learn the basics of R syntax.
- Learn the fundamental R data types.
- Gain familiarity with RStudio.
- Learn how to create tibbles.
- manipulate data in tibbles.
- Compare and contrast the features of Python and R.
- Load data into tibbles.
- Use the pipe operator to sequentialize operations.
- Create tibbles.
- Manipulate data in tibbles.

Frequently Asked Questions:

What is the relevance of R?

R is a programming language used for statistical computing, commonly used in data analysis and by data miners. It's also used for its strength in graphics.

While you don't need to walk away with a mastery of this language, it's helpful to gain exposure to multiple languages and technologies throughout this course to gain some familiarity with their core concepts.

Helpful Resources:

- [R for Data Science \(book\)](#)
- [RStudio](#)
- [RStudio tutorial](#)

Next Up: Project 2

Have a nice week!

19 - Project 2

Congratulations on starting your second larger-scale project! By this time you have some experience working in groups and have even more skills that you've been building over the last few months. Maintain your project scope, hold yourself and your teammates accountable, and knock this one out of the park! I can't wait to see what everyone comes up with.

Have a nice week!

20.1 - 20.3 Tableau

Congratulations on your impressive work with Project 2! The hard work and dedication that you've been putting in over these months has a lot of benefits. We're nearing the end of the program soon - make sure you're keeping up with your at-home practice and review time, and taking full advantage of things like office hours and your peer network.

Unit Objectives:

- Use Tableau to rapidly manipulate tables of data and create visualizations using a drag-and-drop style interface.
- Connect various data formats such as CSV and Excel Workbooks to Tableau.
- Perform exploratory data analysis using Tableau.
- Create groups and sets.
- Create maps and use built-in U.S. Census data.
- Create custom calculations.
- Understand what LOD calculations entail.

Frequently Asked Questions:

How is this different than Excel or the other ways I've learned to make a chart?

While Excel and VBA allow you to be very flexible in how you manipulate your data, Tableau allows you to more easily discover visualizations to which you want to apply your data, even ones that aren't readily available in Excel, like world maps.

Tableau also allows for high flexibility in changing data sources for creating charts, changing your mind on what kind of chart you want to use, and allows you to add data and change scenarios on the fly without much overhead. The drag-and-drop abilities of Tableau make it incredibly simple to integrate more data (and it can handle a lot of data, more so than Excel).

Different tools for different uses makes it easier for you to adapt and adjust, giving your stakeholders what they want.

Why are we using the public version and not student?

Typically, you'll find that a lot of platforms that offer student discounts or trials do not extend this to students of continuing education programs or students without university email addresses. Please ask your SSM about your university policies on this.

What is the difference between public and pro?

Tableau Public is a free version of the software that provides all the same tools as Pro. However, because it's free, it requires all data that is input to be freely accessible to everyone, and has a limit of 1 million rows. This version should not be used if you're dealing with any sensitive or privileged datasets. This version also does not allow saving to your machine and you must save to their server.

Helpful Resources:

- [Tableau Learning Site](#)
- [Tableau Certification](#)

Next Up: Machine Learning

Helpful Resources:

- [Machine Learning with Python Cookbook](#)
- [Deep Learning with Python](#)

Have a nice week!

21.1 - 21.3 Machine Learning

Welcome to Machine Learning! This is a very exciting topic to cover. Like with some other technologies and concepts that we've covered in the past few months, Machine Learning is a very deep well of information. This unit may feel like a firehose of information because there is so much to learn. Deep understanding of Machine Learning will come with time and more focused study outside of class, building on what is covered in class. The Machine Learning content that we will cover will be complex, but you will walk away with a better understanding with which to continue your learning.

Unit Objectives:

- Calculate and apply regression analysis to datasets.
- Understand the difference between linear and non-linear data.
- Understand how to quantify and validate linear models.
- Understand how to apply scaling and normalization as part of the data preprocessing step in machine learning.
- Understand how to calculate and apply the fundamental classification algorithms: logistic regression, SVM, KNN, decision trees, and random forests.
- Understand how to quantify and validate classification models including calculating a classification report.
- Understand how to apply `GridSearchCV` to hyper tune model parameters.
- Understand unsupervised learning and how to apply the k-means algorithm.
- Articulate specific problems on which neural nets perform well.
- Use sklearn's to build and train a deep neural network.
- Use Keras to build and a train a deep neural network.

Frequently Asked Questions:

What is the relevance of machine learning?

Machine Learning is one kind of data analysis that automates model building leading us into the creation of systems that can identify patterns, make decisions, and learn without much human intervention. It's an exciting realization of the growth of artificial intelligence as a computer learns from previous computations and produces reliable and repeatable results from their decisions.

While Machine Learning has been around for a while, there's been a spike in its popularity in more recent years. You can hear about it used more by marketing professionals through Twitter, in the recommendation system that Netflix, Hulu, and Amazon use, and even within self-driving cars. The purpose of Machine Learning here is to analyze more data in less time, delivering accurate results every time.

This branch of data analytics is going to continue growing in various areas including the financial sector to identify investment opportunities, government work to detect fraud, and the further advancement of wearable healthcare devices.

How much Machine Learning knowledge will I walk away with?

Machine learning is a very deep well of knowledge. We could have an entire course dedicated to Machine Learning and still walk away wanting to know more. This course is meant to provide you with an introduction into Machine Learning and give you the tools and familiarity to continue this education on your own. Take advantage of in-class time, office hours, and your network of peers to continue learning more about Machine Learning.

Helpful Resources:

- [Machine Learning with Python Cookbook](#)
- [Deep Learning with Python](#)
- [Scikit-Learn](#)
- [MNIST and Neural Networks](#)

Next Up: Big Data

Helpful Resources:

- [NLP for Big Data: What Everyone Should Know](#)
- [What is Natural Language Processing?](#)
- [TD-IDF Explained](#)

Have a nice week!

22.1 - 22.3 Big Data

This week we'll be going over Big Data! This is another exciting topic, as there are an astounding number of companies that rely on the information found within Big Data and the predictions they can make about their customer base and market.

As we quickly approach the end of class I wanted to remind you that you will be receiving an invitation to our Trilogy Alumni Slack group approximately a week after the class is over. This is a great place to stay connected with your classmates and to connect with alumni across the globe. Accompanying that invitation you will receive a notification that your classes slack channel will close 30-45 days after graduation, so please save any content not saved in your class repository to your local files.

Unit Objectives:

- Identify the pieces of the Hadoop ecosystem.
- Identify the differences and similarities between Hadoop and Spark.
- Write MapReduce jobs locally with MRjob.
- Manipulate data using PySpark dataframes.
- Explain why NLP is necessary in a big data toolkit.
- Apply transformations resulting from NLP data processing to PySpark dataframes.
- Explain and utilize PySpark text processing methods like tokenization, stop words, n-grams, term and document frequency.
- Utilize a NLP data processing pipeline to create a spam filter.

Frequently Asked Questions:

What is the relevance of Big Data?

Our methods of data collection have grown quite a bit in the past few years. We now have a massive amount of data available, but adequately parsing, processing, and analyzing this data is difficult to do using our traditional applications. 2.3 trillion gigabytes of new data is created each day in both structured and unstructured formats. Knowing how to process and analyze this data is a very in-demand skill.

Using Big Data, businesses can gain understanding in where, when, and why their customers buy their product. They can provide targeted advertisement in a more efficient way. They can predict market trends and anticipate future production needs. They can have an edge on the competition or their own future improvement. They can identify new sources of revenue. Big Data can be a game changer to a company.

How much Big Data knowledge will I walk away with?

Similar to Machine Learning, Big Data is a very deep well of knowledge. We could have an entire course dedicated to understanding how to handle Big Data effectively and still walk away wanting to know more. This course is meant to provide you with an introduction into Big Data and give you the tools and familiarity to continue this education on your own. Take advantage of

in-class time, office hours, and your network of peers to continue learning more about Big Data and its real world use cases.

Helpful Resources:

- [Recommended Book: Python Natural Language Processing Techniques](#)
- [NLP for Big Data: What Everyone Should Know](#)
- [What is Natural Language Processing?](#)
- [TD-IDF Explained](#)

Next Up:

Helpful Resources:

- [NLP for Big Data: What Everyone Should Know](#)
- [What is Natural Language Processing?](#)
- [TD-IDF Explained](#)

Have a nice week!

23 - 24 Final Projects

Congratulations on reaching the final project of the Data Bootcamp! The work that you've all put into the program so far has been impressive. I am incredibly excited to see what your teams produce for your last day, showing what you've been able to learn and what skills you've been able to refine over the course of the last 6 months.

As always with projects, ensure that you are collaborating effectively. Communicate daily with your team, ensure that you are maintaining an achievable scope for your Minimum Viable Product and only then expanding on it, and focus on your target audience. This is an opportunity for you all to be creative and put the new skills to the test by choosing a project that is important to you.

Have a nice week!