# Data Analysis Course 2 - Project Requirements

## **Project Deadline: Last Friday of the class, by 11:59am (noon)**

**Please read this document in full.** Make sure to ask any questions before the project is due. If there is any confusion please ask a mentor or a member of the Code Louisville staff.

The purpose of the capstone project is to reinforce what you’ve learned and show off your skills. Your projects will be your go-to tool to show off to potential employers and demonstrate your knowledge. Projects should show that you have a good understanding of Python, it’s methods and implementation.

***This is your project, make something fun and challenging. You are hoping to impress an employer with it!***

## What to expect:

**Weeks 1-3:** Immerse yourself into Python and its methods. Think about the types of programs you have been exposed to. Go over this document with mentors.

**Weeks 4-5:** As you continue to learn and work through your videos, think of ideas that sound fun and interesting. Think about and practice implementation.

**Weeks 6-10:** Project should be worked on in this time frame. You should discuss ideas and implementation with your mentors as you move forward.

**Week 11:** The project **“Turn In Form”** should have been sent out. Please submit your project ahead of the due date. You are able to keep working till the due date/time. **Last Friday of the class; by 11:59am (noon).** Project should be tested, and looked over by a mentor.

**Week 12:** Testing should be done, tidy up the codebase and finalize your project, make sure you have commits to Github. Project should be submitted. **Late projects will not be accepted, failure to meet the project requirements will result in not completing the class.**

### Project Requirements Overview:

Create an application that demonstrates your knowledge of the Python development skills you’ve been learning. This application can be anything of your choosing **as long as it demonstrates the below requirements.**

#### **REQUIREMENTS:**

The below items are the requirements for your personal project for the Python class. Projects are graded on a pass/fail system, thus **every requirement** must be met in order for your project to pass. You should discuss your project/plan with mentors regularly to ensure you’re on the right path to passing.

Program staff and mentors will review projects after they are turned in and will determine if your project meets these requirements:

* The project is uploaded to your GitHub repository and shows at minimum 5 separate commits
  + **Using GitHub’s file uploader does not count as a check-in.** You must upload via Git
  + Gitignore should be used to keep any secrets/passwords used to access APIs / data sources out of the Github repository
* The project includes a README file that explains the following:
  + A one-paragraph or longer description of what your project is about.
  + **Relevant packages that need to be installed to run the project.**
  + Which 3+ features you have included from the below lists to meet the requirements
  + Any special instructions that are required for the reviewer to run your project. (For example: “**run python main.py” from the command line**)
  + Guide to using markdown for README.md files (<https://guides.github.com/features/mastering-markdown/>)
* The project implements a data analysis program that uses pandas, matplotlib, and/or numpy to perform an analysis project of 2 or more pieces of data and implement a rich data visualization in Tableau / Jupyter/Plotly/Matplotlib, or something similar. At a minimum, the program should ingest, analyze, and display data. Any needed data cleaning should be clearly documented and repeatable.
* Choose **at least** 1 item from each category on the **Features** **List** below and implement them in your project
  + *We recommend you pick a 4th item (or more!) to add, just in case something goes wrong with one of your other items - 3 is only the minimum requirement*

**FEATURES**

In Data 2, we want to build off the skills you learned in data 1. If you just pulled in data with a csv, try using an API instead or building your own database. If you just visualized data with a couple matplotlib plots, try using different techniques with those plots and fine tuning them with optional arguments. Or, you can make a Tableau dashboard with your prepared data. We want you to be creative with how you think about working with data, outside of the box thinking for the requirements is encouraged. For example, you could use live streaming data from Twitter that updates your project every minute, you could connect to a Raspberry Pi that reads the temperature in your house/apartment, or you could do an analysis project to answer a research question you’ve always been curious about. Obviously, you don’t need to purchase any extra tools, but any creative way you can think of to get data is great.

We also want this to be a project you’re proud to show off to your employer. You should actually come to some conclusions and have clear descriptions of what your code does and why you’re doing it!

Choose one item from each of the tables below to meet the project requirements. These are somewhat similar in structure and flow to Data 1, but remember, explore the data analysis tools you have further.

1. Loading data. All of the below features should be somewhat familiar to you at this point. However, the challenge with this project is going to be using two separate sources. In feature 2, you’re going to be challenged to combine the datasets using a merge / join, or make a new dataset with different attributes / columns from the separate sets.

| **FEATURE** | **DIFFICULTY** | **RESOURCES** |
| --- | --- | --- |
| Read TWO data files (JSON, CSV, Excel, etc.). | Easy |  |
| Read in TWO text data sources (in any format). For example, email chains or different pages from a book. | Intermediate |  |
| Read TWO data sets in with an API (or use two different APIs that have data you can combine to answer new questions). | Intermediate | [A Cool list of Public APIs](https://public-apis.xyz/) |
| Scrape TWO pieces of data from anywhere on the internet and utilize it in your project. | Intermediate | [RealPython article on webscraping](https://realpython.com/beautiful-soup-web-scraper-python/) |
| Set up a local database and read data in with SQLite or SQLAlchemy | Hard | [SQLite docs](https://docs.python.org/3/library/sqlite3.html) |

1. Clean and operate on the data while combining them. The heart of data science is data wrangling and cleaning, so these features test that. As an example, imagine you have data with countries and their GDPs, then a separate data set with countries and number of geese. Combine the data sets into one then calculate a GDP to geese ratio. Obviously, this is a nonsensical example and geese have no (known) influence on GDP, but this feature is just to test your knowledge of combining and working with multiple sets of data. You may have trouble joining the sets if one has the names of countries in lower case and another has the first letter capitalized. This is where you would need to clean the set in order to merge them.

| **FEATURE** | **DIFFICULTY** | **RESOURCES** |
| --- | --- | --- |
| Clean your data and perform a pandas merge with your two data sets, then calculate some new values based on the new data set. | Intermediate | [Merge and join](https://jakevdp.github.io/PythonDataScienceHandbook/03.07-merge-and-join.html) |
| Clean your data and perform a SQL join with your data sets using either plain sql or the pandasql Python library. | Intermediate | [SQL Join resource](https://www.w3schools.com/sql/sql_join.asp)  [pandasql docs](https://pypi.org/project/pandasql/) |
| If you’re using text data, get some information from your separate documents and summarize them in a DataFrame. This isn’t *literally* a join but accomplishes a similar idea. For example, getting the most frequent word distributions from both documents and then summarizing them in a table. | Intermediate | [Natural Language Processing](https://realpython.com/nltk-nlp-python/) |

1. Visualize / Present your data. In addition to matplotlib and seaborn, you might use Tableau or a pandas pivot table to summarize your data. Imagine you have to present your data to your future manager or in an interview. They should be able to understand what you’re trying to present relatively quickly without having knowledge of how you programmed it.

| **FEATURE** | **DIFFICULTY** | **RESOURCES** |
| --- | --- | --- |
| Make 3 matplotlib or seaborn (or another plotting library) visualizations to display your data. | Easy | [Matplotlib basics](https://matplotlib.org/stable/tutorials/index.html)  [Seaborn basics](https://seaborn.pydata.org/tutorial.html) |
| Make a Tableau dashboard to display your data | Intermediate | [Tableau Basics](https://help.tableau.com/current/guides/get-started-tutorial/en-us/get-started-tutorial-home.htm) |
| Make at least 1 Pandas pivot table and 1 matplotlib/seaborn plot. Pivot tables are a way to summarize your data and present it easily in a way that isn’t just a graph. They can be useful when combined with graphs. | Intermediate | [Pandas pivot tables](https://jakevdp.github.io/PythonDataScienceHandbook/03.09-pivot-tables.html) |
| Make a visualization with Bokeh. You can create interactive online visualizations with this, but it is more involved than the other plotting libraries! Very cool though. | Intermediate / Hard | [Bokeh Gallery](https://docs.bokeh.org/en/latest/docs/gallery.html) |

1. Best practices: these are concepts or ideas that aren’t just writing code, but enhance your project to a higher tier that will impress employers and help other programmers understand your project. These (probably) won’t affect whether or not your project runs, but can help people understand your goals and intent with the project as well as make it more maintainable.

| **FEATURE** | **DIFFICULTY** | **RESOURCES** |
| --- | --- | --- |
| Utilize a virtual environment and include instructions in your README on how the user should set one up | Intermediate | [conda virtual environments](https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html) (note: you can do this with a few different modules, so either venv or virtualenv are fine as well) |
| Write 3 unit tests and include instructions on how the user can run them. This will mostly only apply if you’re building custom functions and classes. | Intermediate | [Pytest docs](https://docs.pytest.org/en/7.1.x/) |
| Build a custom data dictionary and include it either in your README or as a separate document. This will only apply if your data set does not already *have* a data dictionary or if you’re building a custom data set. For an example, see the resources to the right. | Easy | [Data Dictionary example](https://data.louisvilleky.gov/dataset/animal-service-intake-and-outcome) (look under the 6th row on this Louisville Metro Data data set) |
| Any other “best practices” your mentor can think of: this is open to interpretation, but if your mentor has a particular idea for a best practice about your specific project, that will meet the requirement for this table. | n/a | Discuss w/ mentor |

### Interpretation of your data. This requires no code but is **VERY** important. You’re going to be practicing your written skills here, which is an essential job skill. You can just say “clear communication skills” on a resume, you have to prove it to them. We want few grammatical errors, good markdown formatting where relevant, and explanations of *why* you’re coding what you’re coding. When you read through the class work-through project [HERE](https://colab.research.google.com/drive/1JOOZCxAw07BD9kCTIQfBVuId_gGZstvh#scrollTo=120a76c9) you can see what I’m thinking because of how I have things written in the markdown cells above and below the code. Obviously, you will have grammatical and formatting errors, that’s human. But if someone that’s not super familiar with Python or your project picks it up, they should be able to make sense of it relatively quickly. This last box is a “polishing” step to make sure everything is tidy and ready to present to an employer.

Also, notice that your README uses the file extension .md. This is just markdown, and it’s the same “markup language” used in Jupyter Notebook markdown cells.

| **REQUIREMENT** | **DIFFICULTY** | **RESOURCES** |
| --- | --- | --- |
| Annotate your code with markdown cells in Jupyter Notebook, write clear code comments, and have a well-written README.md. Tidy up your notebook, and make sure you don’t have any empty cells or incomplete cells that don’t do anything. Make sure it’s all functional before your final github commit. | Intermediate | [A Guide to Good Comments](https://realpython.com/python-comments-guide/)  [A Guide to Markdown](https://medium.com/analytics-vidhya/the-ultimate-markdown-guide-for-jupyter-notebook-d5e5abf728fd) |
| Annotate your .py files with well-written comments and a clear README.md (only applicable if you’re not using a jupyter notebook). | Intermediate | [A Guide to Good Comments](https://realpython.com/python-comments-guide/)  [A Guide to Markdown](https://medium.com/analytics-vidhya/the-ultimate-markdown-guide-for-jupyter-notebook-d5e5abf728fd) |

### Clarifications and Commonly Asked Questions

* **How projects are reviewed**
  + The project reviewer will use git to clone your project to their local machine and follow the directions listed in your readme file.
  + They will look for the features you documented in your readme, verify they work properly and display an understanding of the code.
  + If all requirements are met, your project passes.
* We are looking to ensure you understand what you wrote, so a poorly implemented project may still not pass if we believe you do not understand the code.
* **Naming your project**
  + Choose a name for your GitHub repo that is relevant to the subject of your project. Do not name your repo "CodeLouisvilleProject" or similar. Choose a name based on what your project is about. For example “Brian’s House of Pancakes”, “Recipe conversion calculator”, etc.
* **GitHub**
  + Yes, GitHub is a requirement. It’s not just how we find and view your project, it’s a critical skill you need to understand. Not having your project checked in on GitHub will result in not completing the class.
  + 5 commits is a minimum to show you’ve made multiple updates. Hopefully you have dozens of commits!

#### **TESTING YOUR PROJECT**

You should test your project on another computer by having someone else obtain and run your project. Several projects in the past have failed to meet requirements because it was written in a way that only worked on that person's computer and not the reviewer's. Your project reviewer will not be responsible for tracking down why your project does not work and this may result in a failed project.

#### **MENTORS**

You should talk about your project early and often with your mentors. Explain your idea and the features you are thinking of implementing. They will help you understand if it will meet the requirements or if you've possibly decided to tackle too large of a project for the 12-week time period. Towards the end of the session, you should again show your project to your mentors and get confirmation that it meets the requirements of the project. You are perfectly fine to ask them to confirm so there should be no surprises about whether your project will meet the requirements when it comes time to submit.