

Data Analysis Course 1 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:

- **Project Plan for Final Project.**

- A project plan defines all work in a project and identifies who will do it. ○
Project Plan is to be submitted starting Week 6, no later than the Sunday of Week 8 11:59PM.
- Project Plan should list the items from the "Features List" you plan to include.
- Project Plan could include:
 - Low to High-Fidelity Wireframe with Features List
 - Project Brief explaining the project and the features you plan to include.
- Please ask a mentor to make sure your Project Plan meets final project requirements.
- Project Plan does not need to be a finished mockup/design/image.

- 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. Do not skip this.
 - **Relevant packages that need to be installed to run the project.** This can be a requirements.txt, a config.py file, or just commands formatted neatly in markdown on your readme that say “pip install pandas” for example. But you have to specify which packages need to be installed.
 - Which 5+ features you have included from the below lists to meet the requirements
 - Any special instructions 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/>)
 - Describe the data used in the analysis in the README file.

A common layout for a data analysis project is as highlighted below. Pick **one of the features from each table below** and incorporate them into your analysis project and you will be well on your way to a data analyst role. Please note, though, *this is absolutely not the only way data analysis is performed*. If you want to do something more exotic like building a GUI to read in user inputs, perform statistical analysis, and tweet out the results, be our guest. Just talk to your mentor beforehand about what you want to accomplish, and there’s a good chance you would meet the requirements and then some in any case. But if you want a basic analysis project, just meet one of each of the requirements below.

In class, you will follow along with the example project [HERE](#)

1. Read data in. This can be anything you can imagine, from the weight of rocks you found in your yard and hand recorded into excel, to pulling from an API that connects to the international space station.

FEATURE	DIFFICULTY RESOURCES
Make a list, dictionary, tuple, or other standard python data structure to read in data for your program. You can do something as easy as a dictionary that contains 5 key value pairs with information about	Easy Cheat sheet for lists, tuples, dicts Data Structure Documentation

Read in data from a local csv, excel file, json, or any other file type. There are many ways to do this, but using Pandas read_ functions is pretty easy.	Intermediate pandas read_csv function
Scrape one piece of data from anywhere on the internet and utilize it in your project.	Intermediate RealPython article on webscraping
Use an API to pull in data. The most common library for this is “requests”, while pulling the data itself is usually pretty easy, sometimes going through the specific API documentation can be kind of complicated.	Intermediate / Harder requests library documentation
Make your own database and query it with SQLAlchemy. Only for the most daring students who have some prior knowledge of SQL and are really interested in the data engineering aspects of Python.	Hardest SQLAlchemy

2. Manipulate and clean your data. This is a very broad category, but you’ll have an idea of how to “clean” and manipulate the data once you see a few videos, so we’re not being too particular here about what that means for you. For example, if you had telephone numbers in a DataFrame, some might be written as “(502) 234-2434” and some might be “502-234-2434”. Still, some might be “5022342434”. Obviously this presents challenges if you’re trying to compare them, so you might need RegEx to pull out the relevant info. That’s only an example though, and your mentors can explain further.

FEATURE	DIFFICULTY RESOURCES
Use built-in pandas or numpy functions to do things like remove 0’s and null values where they don’t belong in your dataset.	Easy / Intermediate Handling Missing Data

Use custom functions or lambdas to perform specific operations to clean or	Easy / Intermediate Making Your Own Functions
--	---

manipulate your data, return those values, then use them in other parts of your project.	
--	--

3. Analyze your data! This is usually the more fun part and probably has more approaches than I can write here.

FEATURE	DIFFICULTY RESOURCES
Use at least 5 different built-in Python functions to find out <i>something</i> about your data. If you had a list for example, finding the length of that list with <code>len(<list>)</code> does tell us a little bit about the data.	Easiest Cheat sheet for lists, tuples, dicts Data Structure Documentation

<p>Do 5 basic calculations with Pandas, like finding the <code>sum()</code>, <code>median()</code>, <code>mean()</code>, or <code>mode()</code> of a column. You could divide two columns by each other. You could multiply a column by a random integer. You could use string operations and find the most common letter in a given entry.</p> <p>Write custom functions to operate on your data. You may discover that you want to find out something particular about data that just doesn't have a built-in Pandas function that accomplishes your goal. Maybe you want your function to read in a DataFrame, search the columns for any mention of "Cars", then return the lowest-priced car in the column along with the mileage. This category is very open to interpretation, so any function operating on your data will work.</p>	<p>Easy Operating on Data w/ Pandas</p> <p>Working with strings</p> <p>Intermediate Making Your Own Functions</p> <p>Note that making your own functions applies to both cleaning and analyzing your data. It's a best practice though to encapsulate a single idea in a function, so you probably shouldn't try to clean and analyze data with the same function.</p>
---	--

4. Visualize your data. The standard choice here is just making a couple visualizations then interpreting them to say something about your data. It can literally be as simple as writing `plt.plot(x,y)` and dropping that in the middle of your Jupyter Notebook then saying a few things about it. This is an extremely useful skill to have. Options 2 and 3 below are outside the scope of the class, but still worth mentioning because some students have had particular interests in these areas and have resulted in really interesting projects. If you don't want to do the extra work, though, that's completely okay.

<p>Make 2 basic plots with matplotlib, seaborn, or any other kind of visualization library that you think looks interesting.</p>	<p>Easy / Intermediate Seaborn example gallery</p> <p>Matplotlib examples</p>
--	---

Make a website and visualize your data with a framework like django or flask. This is definitely not <i>necessary</i> but some students in the past have done this and made really amazing projects, if you're looking for a challenge.	Hard Django Flask
Use a GUI library like tkinter to make an interactive visualization. Again, a few students find this interesting and it makes for truly unique projects. This is something incredible to show off to employers, but you may not have time to do it in the class which is completely okay.	Hard tkinter RealPython article

5. Interpret your data and graphical output. If your project is in a Jupyter Notebook, this should be between the important cells. If you're in a .py file, include your interpretation of your project in the README. This is often overlooked, but we want to know *why* you're programming certain things. Make sure to put this in markdown in between the cells. No one, even the best data scientists, will be able to just look at your raw code and understand your motivation behind why you're doing certain things. Your explanations don't have to be complicated, a quick 2 or 3 sentences on topics is sufficient.

FEATURE	DIFFICULTY RESOURCES
Write markdown cells in Jupyter explaining your thought process and code. If you make a few plots with matplotlib, explain what the reader is seeing and why you chose to plot things that way.	Easy Markdown in Jupyter Notebook Example: Notebook with Markdown Cells
If using some format other than a notebook, make sure your README	Easy (though an art form) Writing a good README

explains your project.	
------------------------	--

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.