Data Analysis Capstone Project Guidelines

Project Deadline: Submission by 11:59 AM (Noon) EST on the last Friday of Module 4.

Your projects are your number one way to show off your skills and land a job as a developer. This document will provide an outline of the minimum requirements to pass Code:You. We encourage you to go well beyond what is detailed here, show your progress to your mentors regularly, and strive for something you want to use to impress employers.

Please read this document carefully and be sure to ask any questions you may have early on.

Expectations by Module:

Module 3: Develop your Project Plan during this phase. Collaborate with mentors to refine your ideas and strategies.

Module 4 (Capstone): Ensure the submission of the "Turn In Form" for the project. Submissions can be made until the due date/time. The project should be rigorously tested and reviewed by a mentor.

REQUIREMENTS:

The subsequent items outline the requirements for your personal Python project. Projects are graded on a pass/fail basis, necessitating the fulfillment of every requirement. Regular discussions with mentors will ensure you stay on track. The program staff and mentors will evaluate projects post-submission to ascertain adherence to these requirements:

GitHub Repository and Commits:

The project must be uploaded to your GitHub repository with a minimum of 5 distinct commits.
Utilize Git for uploading, as GitHub's file uploader is not considered a check-in.
Utilize Gitignore to safeguard secrets/passwords related to APIs/data sources.

README File:

Include a README file explaining your project.
Describe your project in a paragraph or more.
Identify 5+ features from the provided list that you've integrated.
Add any special instructions for the reviewer to run your project.

Visual Appeal:

Design your project to be visually appealing; follow industry trends.
Aim to create projects that employers find attractive and engaging.
Explore other websites for inspiration. Emulate styles and functionalities you find compelling.
Select a color palette and font stack to enhance design consistency.

Data Analysis Implementation:

Develop a data analysis program utilizing pandas, matplotlib, and/or numpy. Execute an analysis project on 2 or more data pieces and produce comprehensive data visualizations via Tableau, Jupyter, Plotly, Matplotlib, or similar tools. Ensure data ingestion, analysis, and display are well-documented and reproducible. Data cleaning steps should be clearly outlined.

Review and Polishing:

Include a section in your project that explains your data interpretation. This highlights your written communication skills, emphasizing clarity and coherence. Ensure that someone unfamiliar with your project can comprehend its content and purpose.

Feature Selection:

Choose at least one item from each category in the Features List provided.
Consider including more than 5 features in case of issues with other selections.

Data Analysis Capstone Features List

Choose **ONE** item from each table below to meet the project requirements.

1. **Loading data.**

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	Intermediate	A Cool List of Public APIs

questions).		
Scrape TWO pieces of data from anywhere on the internet and utilize it in your project.	Intermediate	RealPython article on webscraping
Set up a local database and read data in with SQLite or SQLAlchemy	Hard	SQLite docs

2. **Clean and operate on the data while combining them.**
   Choose **ONE** item from each table below to meet the project requirements.

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
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 pandasql docs
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

3. **Visualize / Present your data.**
   Choose **ONE** item from each table below to meet the project requirements.

FEATURE	DIFFICULTY	RESOURCES
Make 3 matplotlib or seaborn (or another plotting library) visualizations to display your data.	Easy	Matplotlib basics Seaborn basics

Make a Tableau dashboard to display your data	Intermediate	Tableau Basics
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
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

4. **Best practices: Enhance your project to a higher tier that will impress employers and help other programmers understand your project.**
   Choose **TWO** items from each table below to meet the project requirements.

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 (note: you can do this with a few different modules, so either venv or virtualenv are fine as well)
Utilize functions in your code to make it clear and modular.	Easy	https://www.geeksforgeeks.org/python-docstrings/
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
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	Easy	Data Dictionary example (look under the 6th row on this Louisville Metro Data data set)

a custom data set. For an example, see the resources to the right.		
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

5. **Interpretation of your data.**
   Choose **ONE** item from each table below to meet the project requirements.

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 A Guide to Markdown
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 A Guide to Markdown

Review Process:

Projects are evaluated through the following steps:

- The project reviewer clones your project using Git.
- They follow your readme instructions to setup/run your project.
- They will test your app for the requirements in this document and also look through the code
- Reviewers will try only the most minimal troubleshooting steps if your project does not run “out of the box”

Testing Your Project:

You should test your project on more than one computer. A very common mistake is to use an absolute path in your code. For example, if you are reading in an Excel file and your code reads it from “C:/Brian/datafile.xlsx” that will only work on your computer and the reviewer will not be able to run your project.

Mentor Engagement:

Regularly consult with mentors about your project. Share your ideas early and ensure you're on track to a successful project.

Demo Day:

After the completion of the pathways, Code:You will host a Demo Day for graduates of the program to show off their projects to the community. Employers, mentors, and other members of the tech community will be invited to see the projects and meet you - the developer of that project!

Presentation slots for the event will be invite-only, as we will not have enough time for every Code:You student to participate (sorry, there are hundreds of projects!). To incentivize you to do as well as you can on your project, invites to present at Demo Day will be based on the most impressive projects to employers. What determines that selection is ultimately subjective - how do you compare the visual presentation of one project to the technical skill level of another? But we will strive to be as fair as we can during the process.