Last update 9/17/2025

Course Project Overview

Through the remainder of the semester, you will plan and implement an end-to-end project that applies the concepts and techniques that you've learned in class. You will work in teams of 2 or individually (with approval).

Your project must be motivated by a use case or

Milestones

The project has the following milestones. See the corresponding items in Canvas for deadlines.

1. **Team selection:** Select team members or get approval to work individually.
2. **Initial project plan:** ~2 page project plan (~1000-1500 words).
3. **Interim status report**: ~2 page status report (~1000-1500 words).
4. **Final project submission**: GitHub release of your completed project.

Requirements

Your course project must meet the following requirements:

- **Collaboration**: If working in teams, roles and responsibilities of each team member must be clearly defined. Evidence of individual contributions must be evident in the Git history.
- **Git/GitHub**: All projects must use Git/GitHub with a project repository that includes contributions from all team members.
- **Markdown**: Project plans, status reports, and the final project report must be created using Markdown with a well-defined structure.

The following are required for all projects:

- **Data lifecycle** (cf. **Module 1**): Relate your project to one or more of the lifecycle models discussed in class.
- **Ethical data handling** (cf. **Module 2**): Identification of all ethical, legal, or policy constraints and how they were addressed. This includes issues related to consent, privacy/confidentiality, copyright, licenses and terms of use.
- **Data collection and acquisition** (cf. **Module 3**): Collection or acquisition of at least 2 different datasets from distinct trustworthy sources. Selected datasets should either have different access methods (e.g., APIs) or formats/schemas.
- **Storage and organization** (cf. **Modules 4-5**): Select and describe a specific storage and organization strategy. This may include use of tabular, relational, or semi-structured

models via filesystems or databases as well as filesystem structures and naming conventions.
- **Extraction and enrichment** (cf. **Module 6**):
- **Data integration** (cf. **Module 7-8**): Integration of datasets using Python/Pandas and or SQL.
- **Data quality** (cf. **Module 9**): Document data quality assessment results.
- **Data cleaning** (cf. **Module 10**): Describe any data cleaning methods applied (e.g., missing values, outliers, syntactic or semantic cleaning)
- **Workflow automation and provenance** (cf. **Module 11-12**): Provide an automated end-to-end workflow.
- **Reproducibility and transparency** (cf. **Module 13**): Your project must provide sufficient information to allow someone else to reproduce your workflow and analysis.
- **Metadata and data documentation** (cf. **Module 15**): Metadata and data documentation to support discovery, understandability, and reuse.

Team Selection

- Form a team of 2 students or get approval to work individually.
    - If you need help forming a group, reach out via Campuswire (before the deadline)
- Create a Canvas group.
- If you do not submit by the deadline, you will be automatically assigned to a team and deducted 10%
- Teams (re-)formed after the deadline will be penalized 10% per day (up to 5 days)

Note: Working independently is possible but requires instructor approval **before the due date**. Post to Campuswire (Instructors and TAs) indicating why you are unable to work in a team.

Deliverables:

- Form a team of 2 students or get approval to work individually.
- Create a Canvas group.
- Create a new GitHub repository for your course project (one per group).
    - Sign-in/Sign-up to GitHub and get the Student Developer Pack
    - Add all team members as members of the repository.
- Submit the URL to Canvas
- Canvas group

Last update 9/17/2025

Project Plan

See the **Requirements** section above.

Create a project plan that meets the following requirements:

- In your GitHub repository, create a file named **ProjectPlan.md** that contains the following information:
    - **Overview**: Describe the overall goal of your project.
    - **Research Question(s)**: What is/are the question(s) you intend to address?
    - **Team**: Clearly define team member roles and responsibilities
    - **Datasets**: Identify and describe the two datasets that you will use. If you are looking for ideas for datasets to use, please reach out via Campuswire.
    - **Timeline**: Document the plan and timeline for implementing your project including who will complete each task.
        - Your plan must clearly address each of the requirements described above
    - **Constraints:** Describe any known constraints.
    - **Gaps:** Identify any known gaps or areas where you need additional input.
    - *Your plan should anticipate later course topics even if you don't yet know all the details. It is expected that your plan will evolve over time.*
- Submit your plan
    - Add and commit your **ProjectPlan.md** and any other related artifacts (such as images, preliminary code, etc).
    - Push your changes to GitHub
    - Create a **project-plan** tag and release.
    - Submit the URL to your project-plan release via Canvas

Note: After grading, you will be required to update your project plan based on feedback given.

Interim Status Report

Submit a report on the status of your project (~1000-1500 words):

- In your project repository, create a new file named **StatusReport.md** that contains:
    - An update on each of the tasks described on your project plan including references to specific artifacts in your repository (such as datasets, scripts, workflows, workflow diagrams, etc).
    - An updated timeline indicating the status of each task and when they will be completed.
    - A description of any changes to your project plan based on your progress so far
- Commit **all in-progress work** to your GitHub repository

- Submit your status update
    - Add and commit your **StatusReport.md** and any other related artifacts necessary for review
    - Push your changes to GitHub
    - Create a **status-report** tag and release.
    - Submit the URL to your **status-report** release to the assignment item in Canvas.

Final Project Submission

Your final project submission will include a report summarizing your project and data curation actions as well as the actual artifacts (e.g., scripts, data, results, visualizations). Your project grade will be based on both the report and the digital artifacts submitted as described below.

In your project repository, create a file named **README.md** (or use the existing file). This will be your project report. It should have the following structure:

- **Title**: Title of your project
- **Contributors**: Bulleted list of contributors (with optional ORCIDs).
- **Summary**: [500-1000 words] Description of your project, motivation, research question(s), and any findings.
- **Data profile**: [500-1000 words] Description of each dataset used including all ethical/legal constraints.
- **Data quality**: [500-1000 words] Summary of the quality assessment and findings.
- **Findings**: [~500 words] Description of any findings including numeric results and/or visualizations.
- **Future wor**k: [~500-1000 words] Brief discussion of any lessons learned and potential future work.
- **Reproducing**: Sequence of steps required for someone else to reproduce your results.
- **References**: Formatted citations for any papers, datasets, or software used in your project.

Below are examples of artifacts we expect to see as part of your projects:

- **Data collection and acquisition**
    - Script(s) used to programmatically acquire data (e.g., via requests) and check integrity (e.g., SHA-256)
    - Documentation describing steps someone else would use to acquire data, including checksums. This is particularly important if your data cannot be redistributed.
- **Storage and organization**
    - Script(s) used to load data into a relational database.
    - Documentation describing filesystem structure and naming conventions.

- **Extraction and enrichment** (cf. **Module 6**):
- **Data integration**
    - Script(s) used to integrate datasets (e.g., Python or SQL)
    - Conceptual models, integration schema, or query that spans both sources
    - Documentation describing steps used to integrate data.
- **Data quality and cleaning**
    - Script(s) used to profile, assess quality of, and clean data (e.g., Python or SQL)
    - Documentation describing steps used to profile and clean data
    - OpenRefine operation history ("recipe")
- **Data analysis and/or visualization**
    - Script(s) used to analyze data and/or generate associated visualizations
    - Analysis results and/or visualizations
    - Documentation describing steps used to analyze or visualize data
- **Workflow automation and provenance**
    - Snakemake workflow automating your end-to-end analysis workflow from acquisition to result visualization.
    - Run All script that can be used to re-execute your end-to-end analysis workflow
    - Documentation describing the steps required to repeat your workflow
- **Reproducibility and transparency**
- **Reproducible package**
    - Sufficient information to allow someone else to reproduce your analysis including:
        - Documentation describing steps someone else needs to take to reproduce your results
        - Data or documentation describing how to obtain data used
        - All code, workflow scripts, etc., needed to reproduce your results
        - Actual results of your analysis including output files, visualizations, etc.
            - You are required to upload your output data (and optionally, your input data, if not retrieved programmatically) to Box and include, in your report, a shareable link to the folder where the data is stored, as well as information on where the data should be saved in your project folder once it's downloaded from Box.
            - You are responsible for ensuring the shared folder can be accessed by the TAs. If you are unsure about this, you should reach out to the TAs on Campuswire at least 12 hours before the deadline. You may lose points if the TAs cannot access the data you've shared.
            - Make sure you add the path to the data that is already in Box to .gitignore before you push any changes to GitHub.
        - Specification of software dependencies (e.g., requirements.txt) and record of specific packages used (e.g., output of pip freeze). Optionally, a Dockerfile and container image pushed to Dockerhub.

        - Optionally, a Dockerfile and container image pushed to DockerHub or a CodeOcean capsule
    - Licenses for data and software created as part of your project
- **Metadata and data documentation**
    - Data dictionary or codebook as text file, PDF, or self-describing data formats.
    - Descriptive metadata describing your project in conformance with a standard such as DataCite, Schema.org.

Submission:

- Add and commit your **README.md** and all other artifacts to your team GitHub repository.
- Push all changes to GitHub
- Create a **final-project** tag and release
- Submit the URL to your final-project release via Canvas.

Grading

Important: We must see clear contributions from all team members to receive full credit. If this is not apparent from the Git commit history, make clear in your final report through a detailed contribution statement.

To grade your project we will:

- Clone your project repository
- Read your **README.md**
- Attempt to reproduce your complete workflow
- Assess your project for:
    - **[20 pts] Reproducibility**: Did you provide sufficient information for us to independently reproduce your results from data acquisition through analysis?
    - **[20 pts] Transparency**: Are artifacts available for each step in your workflow? This includes data acquisition, integration, profiling, quality assessment, cleaning, analysis/visualization and workflow.
    - **[10 pts] Compliance**: Did you comply with licenses and terms of use?
    - **[10 pts] Citation**: Did you accurately cite data and software used?
    - **[10 pts] FAIR**: Did you publish your project to an archival repository with accurate metadata and obtain a persistent identifier?
    - **[10 pts] Documentation**: Is your project documentation complete? Are all manual steps clearly described?
    - **[20 pts] Quality**: Is your project well organized (e.g., understandable filenames, no unnecessary artifacts)? Is your report complete, clearly structured and well-written?