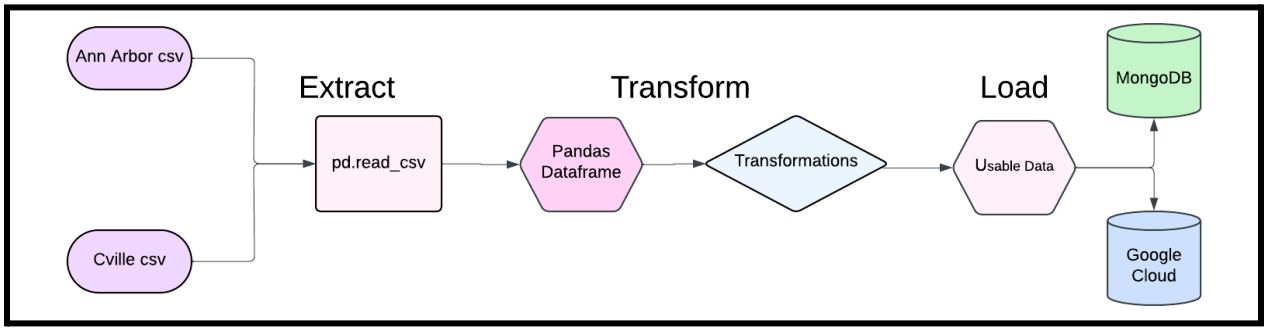
Our analysis of crime data revealed not only patterns in two significant and dynamic college towns, but also valuable insights into the technical and collaborative challenges of data science projects. As has been displayed throughout this semester of data science, the actual coding and data analysis itself is surprisingly a very small amount of the undertaking of a data science project. We began by reading through the assignment multiple times and discussing it amongst our group – a practice that we learned was vital from our first data project this semester. Similar to before, the project was purposefully vaguely defined and could be done with very many strategies. We started by asking ourselves what the final product will look like. In this case, we tried to explain the ETL implementation and cloud storage in simple terms, working backwards from the cloud storage and assessing what needed to exist or be done to get to that point. By going backwards, step-by-step, we eventually had a game plan and a good place to start. Knowing that we needed a better idea of what the data visualization and transformation steps would look like in our plan, we started with data selection.

This section of the project was somewhat straightforward. We had all worked with city-level public datasets in the past and know that they are usually easy to access, relatively simple to understand, and do not require extensive data cleaning. With that agreement, we landed on Charlottesville crime data and began to brainstorm what other dataset would be appropriate to relate to Charlottesville data. Comparing another college town's crime data seemed like a natural fit; we settled on Ann Arbor but ran into our first major challenge. We wanted to make direct comparisons between the two towns but found that the Ann Arbor data was similar but not exactly the same. This was a challenge because we now needed two different plans for the visualization and transformation (even though they were relatively similar) to reach the same endpoint where the datasets would both be easily comparable. This was good practice, though. Data is rarely stored in the exact same form, and it is incredibly rare to find a data project where all of the data is presented in exactly the same way – data involves a degree of creativity and an ability to take what you know and stretch it into a new solution. Despite our urges to restart our search and look for two perfectly correlated datasets, we plunged into our crime-focused datasets.

The next key issue – and likely our largest issue throughout – was the difficulty of group coding. While we each took our own section of the project to work on, many of the later sections relied on the completion of earlier sections to operate correctly. In the past – and with smaller groups – we found success with Google Colab. With upwards of three people working on the code at any given time, though, Google Colab's slow collaborative refresh rate posed an issue for our team. We set up a GitHub repository where we could upload our updated code but found challenges working in GitHub as well. Unfortunately, none of us had a great solution and we ended up sticking with Google Colab and essentially building each block in its own file and putting them together over time. It was not a perfect solution by any means, but the best we could do given our lack of knowledge of better alternatives and lack of general direction when it came to collaborative coding. This is an area where we can certainly improve in the future.

The static nature of the data (csv flat files) meant that extracting information into a usable environment was straightforward. Without the need of web scraping or API's we were able to store the data locally and read it using pandas with one line of code. Once in a dataframe, the data could be transformed however desired. In order to make accurate comparisons between locations we needed to standardize dates and columns of interest. This process required basic planning on what kind of analysis we wanted. Once our end product was identified we were able to use basic transformations (converting to date time, isolating crime specific data, adding usable location information) to create a useful dataset. The final step, loading the transformed data into storage, also required some basic planning. After struggling to effectively store the data locally we reexamined the needs we would have for the transformed data. We identified a need for flexibility in what can be stored and the ability to scale across new different jurisdictions. Knowing this, a cloud storage solution made more sense. So, using pymongo and the google cloud storage library, we were able to connect our python environment to these cloud storage systems and store our transformed data in a flexible and accessible way. We learned that carefully considering the needs of the future best informed strategies for both transformation and storage and saved time over the long term. Future improvements to the ETL pipeline focus on team accessibility and security. Currently, storage access is controlled by a key JSON for both MongoDB and GCS. Without this key, a group member is not able to access the database or bucket on their own. Sharing this JSON file with encryption online potentially risks database

security. Further development of the pipeline would resolve this security issue and could also focus on ingesting a wider range of data and automating transformations.



Basic ETL Pipeline

A key part of this assignment was the obligation to work with a team. While our team got along well, working as a team did not come without its challenges. Even as we discussed our general thoughts of the assignment and instructions, our team dynamic was already starting to form. As we moved through each section, it became obvious that all of us could work on each part and end up with an efficient and effective outcome. Instead, it only made sense to divvy up the work in some way that maximized the combination of both speed and quality. Here, we learned the valuable lesson that communication is necessary from the start. As we were going through each section, we each spoke to our strengths and how they would be able to fit into the assignment. We were each able to focus on one part of the project, but also identify other group members who we may need to rely on if one of their strengths would help a specific section. We have learned that this is a great starting point – we each knew what work we had to do, who we could ask for help, and how each part of the puzzle would come together to create a final product.

What we could have done better, at this point, would have been to establish a de facto group leader. Luckily, our group was all highly motivated and well-organized, so our lack of a clear organizational leader was not a major inhibitor to our success. Still, we tended to lack a clear schedule and consistent meeting time which would occasionally result in “dead periods” where there was very little discussion about our project over the course of a few days and as our focus was put elsewhere. Our individual schedules and varying priorities make this a nearly

unsolvable task, but it could have been remedied by a clearer group leader who, at the very least, could have kept up a schedule and task list. The project worked out in the end and with few speed bumps along the way, but we now know that we could have done even better on the team coordination side of things had we not only assigned work roles but also assigned team roles.

This project offered valuable insights into managing data analysis projects and collaborative workflows. Through addressing challenges in data integration, team coding, and communication, we developed practical strategies and identified areas for improvement. The experience highlighted the importance of adaptability, technical proficiency, and effective teamwork in achieving project goals. These lessons, while specific to this assignment, provide a solid foundation for tackling future data science projects with greater efficiency and precision.