

I started this project to analyze housing prices from a Kaggle dataset and find patterns that could help in understanding what factors influence housing costs. The project began with identifying the structure of the dataset from Kaggle. After downloading the dataset, I studied its fields, such as price, bedrooms, distance from the metro, net square meters, etc. This understanding guided the design of the `HouseRecord` struct in the `data_loader` module. Each field in this struct corresponded to a column in the dataset. I created the `load_data` function to read the CSV file and put it into `HouseRecord` structs. Once I successfully loaded the data, I moved on to the analysis. The first function, `calculate_statistics`, computes basic metrics like average price and the average number of bedrooms and bathrooms. I found this step important for understanding the overall distribution of the data. Then, I added a function to find price trends, particularly focusing on the price per square foot, which is a common metric in real estate analysis. Finally, the `correlation_analysis` function evaluates how strongly the price correlates with living area size. This part of the analysis aims to reveal the linear relationships in the data. To make the results more interpretable, I included a visualization component. Using the `plotter` crate, I implemented a `generate_histogram` function to create visual representations of the data, such as price distributions. This visualization was instrumental in identifying potential outliers or skewed distributions in the dataset. Testing was an essential part of the development process. I wrote tests to validate the core analysis functions in the `tests` module. These tests ensured the correctness of calculations and protected against future regressions. For example, the tests confirmed that the average calculations and correlation computations produce expected outputs for small, controlled datasets. I documented each step of the process and maintained a clean Git workflow. My commits were frequent and descriptive, capturing every significant addition or change to the codebase. This not only helped me track progress but also ensured that I could roll back changes if needed. One of the biggest challenges I faced was aligning the Rust code with the Kaggle dataset's structure and ensuring compatibility with real-world data irregularities. Rust's strict type system was both a blessing and a hurdle, as it forced me to handle every edge case explicitly.