Data Project Rubric:
Data Analysis with Python (Pandas & Matplotlib)

1. Project Overview

- **Objective**: Students should define the project's goal or main question. ○ Example: *Analyze which days of the week have the highest sales.*
- **Data Source**: Describe where the data comes from and how it might answer the question.
    - ○ Example: This data source comes from https://www.kaggle.com/datasets/hhs/health-insurance

2. Data Collection and Loading

● **Load Data**: Use Pandas to load a dataset (CSV, Excel, or database). ● **Initial Check**: Display the first few rows and basic information about the dataset, noting column names, types, and missing values.

- **Selection Options**:
    - ○ Choose a dataset you find interesting (Video game sales, Sports stats, Carbon emissions, etc.).
    - ○ Choose which columns or data to focus on and drop the rest.

3. Data Cleaning and Preparation

- **Handle Missing Values**: Choose how to handle missing values (drop, fill, or leave as is).
    - ○ Explain why you did this drop, fill, etc.
- **Data Type Adjustments**: Convert columns to appropriate types (e.g., dates to datetime).
- **Feature Engineering**: Create at least one new feature from existing data.
    - ○ Example: extracting the month from a date
    - ○ Calculate the percentage
    - ○ Combine first and last name columns into a full name column

4. Exploratory Data Analysis (EDA)

- **Descriptive Statistics**: Provide summary statistics (mean, median, min, max) for numerical columns.
- **Data Visualizations**:
    - ○ **Required**: One visualization (e.g., line plot) if time-series data is relevant. ○ **Optional**: Additional visuals like bar charts for comparisons, histograms for distributions, or scatter plots for relationships.

5. Analysis and Insights

- **Findings**: Summarize any patterns observed and address the main project question.
- **Supporting Data**: Reference specific statistics or plot features to back up findings.

6. Conclusion and Recommendations (10 points)

● **Summarize**: Present the main conclusions drawn from the analysis. ● **Recommendations**: Based on findings, suggest actions or further analysis.

Optional Advanced Section (Bonus)

● **Advanced EDA**: Use pair plots or correlation matrices to explore relationships. ● **Dashboards**: Create a simple dashboard with Matplotlib or extend to Dash/Streamlit.

Features	Description	links
1. Define the question		
	Find a question to answer based on the data you decided to use.	
2. Read in the Data		
Easy: Read a local file	Import the data locally a csv, excel file ect	
Medium: API call	Import data from an API	
Hard: Web Scrape	Scrape data from a website and put it into a DataFrame	
Hard: Database	This would involve building a database and doing a query to pull the data then reading it in.	
3. Data Cleaning		

You must address missing values	This can be done many ways. You can remove it or fill it. You can forward fill, back fill, or fill with Mean, Median, or Mode ect.	
You must convert your columns to proper data types.	Convert your columns to int, Float, String/Object, etc.	
You must engineer one feature	Calculate a new column from your data. example:Group ages or scores into categories (e.g., ‘Child’, ‘Adult’, ‘Senior’) or Extract year, month, day, or weekday from a date column.	

4. Exploratory Data Analysis		
Provide descriptive stats	Provide summary statistics (mean, median, min, max) for numerical columns.	
Data Visualization	Make at least one plot from your data.	
5. Analysis and Insights		
State your findings in a markdown cell	Summarize any patterns observed and address the main project question.	

Support your findings in a markdown cell	Reference specific statistics or plot features to back up findings.	
6. Conclusion and recommendations		
Summarize your findings	Present the main conclusions drawn from the analysis.	
Recommendations	Based on findings, suggest actions or further analysis.	