DataEng S24: Data Validation Activity

High quality data is crucial for any data project. This week you'll gain experience with validating a real data set provided by the Oregon Department of Transportation.

Due: this Friday at 10pm PT
Submit: Make a copy of this document and use it to record your results. Store a PDF copy of the document in your git repository along with any needed code before submitting using the in-class activity submission form.

A. [MUST] Initial Discussion Question

Discuss the following question among your working group members at the beginning of the week and place your own response(s) in this space. Or, if you have no such experience with invalid data then indicate this in the space below.

Have you ever worked with a set of data that included errors? Describe the situation, including how you discovered the errors and what you did about them.

Response:
In a recent project for my LLM subject, we had to train a large language model using data collected from various sources. However, the data we gathered contained numerous errors, including incorrect formats and missing information. To tackle this issue, we initially conducted a thorough analysis of the dataset. Then, we developed Python scripts to clean the data and transform it into the desired format. These scripts helped us identify and rectify errors, ensuring that the dataset was suitable for training our language model. By addressing these issues proactively, we were able to prepare a clean and reliable dataset for our project.

Background

The data set for this week is a listing of all Oregon automobile crashes on the Mt. Hood Hwy (Highway 26) during 2019. This data is provided by the Oregon Department of Transportation and is part of a larger data set that is often utilized for studies of roads, traffic and safety.

Here is the available documentation for this data: description of columns, Oregon Crash Data Coding Manual

Data validation is usually an iterative multi-step process.

B. Create assertions about the data
C. Write code to evaluate your assertions.
D. Run the code, analyze the results
E. Write code to transform the data and resolve any validation errors

B. [MUST] Create Assertions

Access the crash data, review the associated documentation of the data (ignore the data itself for now). Based on the documentation, create English language assertions for various properties of the data. No need to be exhaustive. Develop one or two assertions in each of the following categories during your first iteration through the ABC process.

1. *existence* assertions. Example: “Every crash occurred on a date”
   Every crash has a location specified by latitude and longitude
2. *limit* assertions. Example: “Every crash occurred during year 2019”
   Every crash occurred on Mt. Hood Hwy 26
3. *intra-record* assertions.
   If a crash record has a Participant ID, it also has a Participant Type Code
   For every participant there must be age.
   For every participant there must be gender
4. Create 2+ *inter-record check* assertions. Example: “Every vehicle listed in the crash data was part of a known crash”
   For every carshId there will be at least one vehicle id
   For every crashId, there will be at least 2 types of records
5. Create 2+ *summary* assertions. Example: “There were thousands of crashes but not millions”
   For every crash, the number of participants should be more than or equal to the number of cars
   Crashid is unique across all records
6. Create 2+ *statistical distribution assertions*. Example: “crashes are evenly/uniformly distributed throughout the months of the year.”
   Crash data is evenly distributed throughout the year
   Crashes are more during middle of the day

C. [MUST] Validate the Assertions

1. Study the data in an editor or browser. Study it carefully, this data set is non-intuitive!.

2. Write python code to read in the test data. You are free to write your code any way you like, but we suggest that you use pandas' methods for reading csv files into a pandas Dataframe.
3. Write python code to validate each of the assertions that you created in part A. The pandas package eases the task of creating data validation code.
4. If needed, update your assertions or create new assertions based on your analysis of the data.

D. [MUST] Run Your Code and Analyze the Results

In this space, list any assertion violations that you encountered:

- I encountered 483 data points with the value 4. Since the purpose of this data is unclear, I have removed them to avoid including irrelevant information.

For each assertion violation, describe how to resolve the violation. Options might include:

- revise assumptions/assertions
- discard the violating row(s)
- Ignore
- add missing values
- Interpolate
- use defaults
- abandon the project because the data has too many problems and is unusable

No need to write code to resolve the violations at this point, you will do that in step E.

E. [SHOULD] Resolve the Violations and Transform the Data

For each assertion violation write python code to resolve the violation according to your entry in the "how to resolve" section above.

Output the validated/transformed data to new files. There is no need to keep the same, awkward, single file format for the data. Consider outputting three files containing information about (respectively) crashes, vehicles and participants.

A) I have added the Google notebook with code and CSV files in the github.

F [ASPIRE] Learn and Iterate

The process of validating data usually gives us a better understanding of any data set. What have you learned about the data set that you did not know at the beginning of the current ABC iteration?

Next, iterate through the process again by going back through steps B, C, D and E at least one more time.

A) Through data verification, I identified inconsistencies and inaccuracies in the dataset. This involved creating assertions, writing code to test them, and analyzing the results.I have rewritten many assertions through this process. This not only ensure data quality but also led to a deeper understanding of the data. For example, by examining the "Record Type" column, I discovered that the dataset was actually a merger of three separate collections. This verification process also helped confirm the dataset's completeness and accuracy