**TECHNICAL REPORT: PROJECT 2**

**Team Members: Tae Lee, Damian Sloane, Yanuo Zhou, John Pospisil**

**Keywords: Python, Pandas, SQL, PostgreSQL, SQLAlchemy, Kaggle**

**Project Goal:** Combine publicly available dataset containing information about aircraft accidents with a dataset containing counts of flight data for the month of February 2011. The focus was on domestic flights only. Over time these data could be used to model the probability, severity and mortality of aircraft accidents relative to a variety of factors including make and model of the aircraft, weather conditions, visual flight rules and location.

**Data Extraction:**  Data were downloaded as CSV files from Kaggle. In Python (see Jupyter notebooks in <https://github.com/NU-DataCHI-4-Boeing737-ETL-Project> datasets were converted in Pandas dataframes. Using the library SQLAchemy, the create\_engine function was imported which allowed for loading dataframes into PostgreSQL database.

Data located at: <https://www.kaggle.com/khsamaha/aviation-accident-database-synopses>

<https://www.kaggle.com/marcslaughter/airport>

**Data Cleaning and Transformation:** Variables were renamed to allow joining of files based on airport code and to facilitate importing into PostgreSQL. One dataset contained a character that had to be eliminated for import. Additionally, that dataset also had fields in the rows that contained commas. When imported, this created an inadvertent splitting of some singular fields into multiple fields.

While there are other methods to handle unwanted commas, the solution to handling such problematic variables was to drop them. Several variables were originally dropped from the original accident dataset as they were either sparsely populated or contained values the team could not decipher. Eliminating the problematic was simple and effective.

The cleaning and loading of data were an iterative process. Over several cycles, minor errors were found in the loaded database. Subsequently, data cleaning eliminated errors until data loaded well.

**Loading:** Cleaned datasets were loaded into PostgreSQL tables and a “left join” was used to combine the datasets based on airport code. This formed a simple relational database.

**Future Steps:** Were this project to persist beyond 9 Oct 2019, a process to eliminate unwanted commas from the data could be used to add more variables to the database. Stock queries could be written that would allow a user to make simple queries while only changing some fields within the query. Conceivably, a web application could be developed to allow any user to query the database.