ETL Report on Vehicle-related fatal accidents in texas

# Extraction

We learnt about driving related accidents through Kaggle which linked to FARS (Fatality Analysis Reporting System), <https://www-fars.nhtsa.dot.gov>. All our data was obtained from [https://www-fars.nhtsa.dot.gov](https://www-fars.nhtsa.dot.gov/) website. Our Data set contains vehicle associated fatal incidents with in the state of Texas, in the year 2016. This dataset contains Pre-Crash and Crash relevant information with: State Number, Case number (Case Number represents each fatal accident) and Vehicle Number being common to both Pre-Crash and Crash fields.

Data within the variables in the Pre-Crash and Crash fields are number coded. Each number code further represents details associated with fatalities. Each representation table has codes in one column and its interpretations as another column. Excel files of field tables and their interpretation tables were first downloaded, cleaned for irrelevant labels, and finally converted to CSV files.

## Pre-Crash Information (Shilpa’s Data set)

### Variables in Pre-Crash field included: Crash Type, Driver Distracted By, Driver Maneuvered to Avoid, Driver’s Vision Obscured By.

## Crash Information (Lisa’s Data set)

### Variables in Crash field included: Atmospheric Conditions, City Name, County Name, Crash Related Factors, Drowsy, First Harmful Event, Holiday Related, Injury Severity, Large Truck Related, Light Conditions, Police Reported Alcohol Involvement, Police Reported Drug Involvement, Speeding, Work Zone.

# Transform (data Cleaning)

All the data cleaning was performed in Jupyter Notebook using Pandas and Excel. Pre-Crash and Crash field tables were imported into Pandas DataFrame.

## Pre-Crash Information (Shilpa’s Data set)

### All Null values were removed using dropna().

### Values of Case Number and Vehicle Number was converted to string, and leading zeros were added using pd.zfill() method.

### Unique VehicleID was then created by combining zfill(ed) Case Number and Vehicle Number.

### Driver Distraction column contained two values for 7 cases. This meant the driver was distracted by more than one event. Therefore, this column was duplicated and both columns were cleaned to have only one value in each cell.

### To change the values from numeric to its interpreted meaning, specific Code Definition tables were imported and pd.merge ()was used to replace the numbers to its interpretation.

### Columns were renamed as State Number, Case Number, Crash Type, First and Second Distraction, Driver Action and Obscured Vision

## Crash Information (Lisa’s Data set)

### Null values were removed. Using dropna()

### VehicleID and unique PersonID were created using the pd.zfill() method to combine case number, vehicle number, and person number

### Because we are focused on information specific to a vehicle, instances where a person is not associated with a vehicle (i.e. a pedestrian struck by a car) are not included.

### The initial pull of crash data from the FARS website provide 101 different fields to describe a crash event. Only 23 variables are kept for this study.

### The crash information provides a row of data for all people involved in the crash regardless of whether the person was fatally injured by the accident. We only keep rows of person information for individuals that died in the crash.

### Number code tables were imported the interpret the meaning of the numbers listed in the crash information table. The values of the interpretation fields were cleaned to remove leading umber code values.

### The crash data table was merged to number code tables to replace number codes with their interpretation.

### The field that represents the number of people and the number of vehicles involved in a crash are translated to integer format.

### The date string value is translated to a datetime field.

### Any remaining field names that include parentheses are renamed.

### Some vehicle information is not available in the distraction data. Those rows are removed from the crash information data. VehicleIDs removed are listed: 09780002, 14820002, 15560002, 15560002, 17090002, 27400002, 27930002

# Load

Transformed data in Jupyter NoteBook was later loaded in a Postgres SQL database. Postgres SQL was chosen because the data is organized in a table-based relational format. Two tables named crashinfo and crashdistraction were created in SQL. The primary key for the crashdistraction table is VehicleID. This field is used as the foreign key in crashinfo. The primary key for crashinfo is PersonID.