**ETL**

We looked at several data sources to pull Black Friday themed data. Using Kaggle and Data.World as well as two websites: Reviews.org and Blackfridaydeathcount.com. We found several csv files and web tables we wanted to pull to eventually use in further analysis. The data includes the following;

* Incidents - Details of 27 different Black Friday incidents (death or injury) in the U.S.
* Places - same data as incidents but focuses on the location where the incident occurred.
* BlackFriday - Dataset of 550 000 observations about the black Friday in a retail store, it contains different kinds of variables either numerical or categorical. It contains missing values.
* State Violent Crimes – Details the number of violent crimes by state during Black Friday sales.
* Incident by Year – Highlights news articles that describe incidents related to Black Friday sales since 2010.

**Extract**

The first three tables, Incidents, Places, and Blackfriday were downloaded from csv files while the State Violent Crimes and Incidents by Year were web tables that were screen scraped. All data sources were pulled into a Jupyter notebook and Pandas was used to create dataframes in order to more easily manipulate the data.

It should be noted that both the ‘State Violent Crimes’ and ‘Incident by Year’ tables were loaded into MySQL and MongoDB. This was done in the event one option was chosen over the other in future analysis if fields or columns were changed. In which case, MongoDB would be preferred since the code will not require updating.

As such, the extraction of data, to eventually load in to MySQL, used the pd.read\_html method. This was fairly straightforward and easy to extract. The extraction of data to eventually load in to MongoDB was done in 2 ways. The first was done by identifying the html tags within the site and running a loop to extract the data in to Mongo format. This was done for the ‘State Violent Crimes’ table. The second leveraged the pd.read\_html method. Then converted to format condusive to loading in to MongoDB.

**Transform**

The csv files were in good shape to begin with. We decided to pair down the columns for manageability. Column names needed to be changed to adapt to MySQL.

The data extracted from websites via screen scraping required additional steps to clean and prep for loading in to a database. For MySQL, The ‘State Violent Crimes’ table required one column to be removed since it was irrelevant for future analysis. The ‘Incident by Year’ table took more effort. The html table contained blank rows that had to be removed. Additionally, values of zero were represented by a ‘-‘ which had to be converted to a ‘0’. Lastly, one of the article titles contained a ‘:’ which resulted in an error when the data was loaded. Therefore, that row was removed from the set. Lastly, the index was reset.

**Load**

We created the databases and tables in MySQL to prepare to load the data from Jupyter into MySQL. Then back in Jupyter, we created the engine to load the data into each table that was created. Then we checked to make sure the data was loaded properly and without error.

The upload to MongoDB was done in to ways. The ‘State Violent Crimes’ table was screen scraped and immediately loaded in to a MongoDB format. The ‘Incident by Year’ was converted to a dictionary first and then loaded in to MongoDB.