**BAY AREA SALARIES**

***ETL Project*** *by Teresa Ruan, Kendall Jackson & Adam Dunbar*

**San Francisco Salary Data** (source: <https://www.kaggle.com/ronaldtroncoso20/sf-salaries-extended/>)

For this data source, Kaggle user Ronald Troncoso wrote code to determine Gender based on the first name of City employees of San Francisco. While not a 100% accurate methodology, it allows us to analyze (within a reasonable error range) the gender-based pay gap that exists within the City of San Francisco’s workforce. We extracted the data set in CSV form and brought it into a Jupyter Notebook for data cleaning and transformation. Here is a quick summary of the steps taken with Pandas to prepare it for eventual loading into PostgreSQL:

* Import data from Kaggle into Jupyter Notebook
* Use Pandas to clean data, make consistent Null values across all columns
* Make sure data types are correct (conversion was necessary)
* Transform DF Column Names for Loading Into PostgreSQL
* Divide Master/Transformed DF into 4 Separate DF's by Year for Analysis in SQL
* Export Finalized DF to CSV as a Data Back-Up
* Import all 4 separate Tables into PostgreSQL using SQL Alchemy engine

(For all the details, please view the “sf\_salaries\_2011\_2014” jupyter file on GitHub.)

**Kendall’s Data** (source: <https://www.kaggle.com/rainey/gender-pay-gap> )

For this data source, Kaggle user Rainey wrote code to determine the difference between pay for gender for individuals from the United Kingdom. The data set was heavily based on statistical metrics. It allows us to analyze (within a reasonable error range) the different quartile ranges of pay between gender, the company that was responsible and the CEO or director responsible. We extracted the data set in CSV form and brought it into a Jupyter Notebook for data cleaning and transformation. Below is a quick summary of the steps taken with Pandas to prepare it for eventual loading into PostgreSQL:

- Import data from Kaggle

- Checked the data type and fixed any data type that was incorrect

- I split the “Responsible Person” Column into two parts

o The original had the person responsible and their title.

o My data frame had the person responsible in one column and their title in the other

o I achieved this by running a for loop

o Within the for loop I split on the “(“ where the job title started.

o I then replaced the “)” with and empty space “” so that the parenthesis no longer existed.

o I appended the title to one list and the person responsible to another list.

o I then created a new column base on those lists.

- I transformed the DF name for loading into PostgresSQL

- Export Finalized DF to CSV as a Data Back-Up

- (For all the details, please view the “UK\_Gender\_Data” jupyter file on GitHub.)

**Teresa’s Data** (source: <https://www.kaggle.com/rtatman/silicon-valley-diversity-data> )

Rachel Tatman, Kaggle user has published three sets of data regarding diversity and compensation in 23 Silicon Valley tech companies. Higher compensations in the tech industry have been one of the hot topics people discuss in all workforce.

During the ETL process, our team put the effort in the data cleaning process. Our goal is to drop data that do not provide measurable value to users of the database such as analysts. To achieve our goal, we used three approaches during the transform process where is our focus for this project.

1) Extract data from [www.kaggle.com](http://www.kaggle.com) in .csv format

2) Transform

2a) **Inspect:** This is the initial process once CSV file is loaded to Jupyter Notebook. Once function has been used to quickly identify the dataset’s column names, total rows, total columns, null value status, data types and memory usage.

2b) **Filtering:** during this step, we look into any column that has numeric value that either has “na” or “0” value, then we examine if these rows provide any meaningful insights or they can be dropped. After carefully inspect these data, we have drop 0.0 percentage from distributions.csv and 0 count from revel\_EE01.csv.Our next step in filtering process is the inspect each csv by column at a detail level using df[‘column\_name’].unique()

2c) **Cleaning:** the last step in transform process is to clean data at detail level such as naming convention in ethnicity or job category.

3) Load data set into PostgreSQL as tables using sqlAlchemy.