# About

This project was completed by Jason, Ashley A, and Erica.

The goal of this project is to put together a small database containing data on videogames and their sales performance stats. This project takes a look at two data sources but separated into four tables that was later combined together into one file. Details to the dataset below.

# Datasets

|  |  |
| --- | --- |
| **Data Source** | **URL** |
| Kaggle (sales\_2019\_vg.csv) | https://www.kaggle.com/ashaheedq/video-games-sales-2019?select=vgsales-12-4-2019.csv4 |
| Pricecharting.com | <https://www.pricecharting.com/> |
| Kaggle (Amazon Best Sellers) | https://www.kaggle.com/edwardtoledolpez/amazon-mexico-top-50-best-sellers |

# Extract

The original data source from Kaggle was a massive collection of data in csv file containing video game data dating back to 1980s. It was a clean data with some fields missing due to the age of the video game published.

For Pricecharting.com data, we resorted to basic copy and pasting to save time. We then used Excel and saved the data as csv format.

For the final CSV this was extracted from Kaggle.com. It was the Amazon Best Sellers, it included a CSV with only videogame sales in Brazil.

# Transform

For Kaggle data source, we resorted to saving two separate csv files from it: **ps4\_sales.csv** and **xboxone\_sales.csv**. This allows us to compare and combine the titles that are shared between the two platforms while eliminating those that are exclusive to one or the other.

For Pricecharting.com source, we saved the two separate csv files as **ps4\_prices.csv** and **xboxone\_prices.csv** and conducted preliminary clean up using Excel by eliminating columns containing Loose Prices and NIB Prices.

In Python, we loaded **ps4\_sales.csv** and **xboxone\_sales.csv** into separate data frames using PANDAS and proceeded to merge them into one data frame called **ps4\_sales\_df**. Same process goes for **ps4\_prices.csv** and **xboxone\_prices.csv** and we named it as **ps4\_prices\_df**. For ps4\_prices\_df, we had to find the average of the two data frames to get a better gauge of the video game prices. Lastly, we merged **ps4\_sales\_df** with **ps4\_prices­\_df** into **cleaned\_df**, which we dropped the rows containing no or zero under the New Average Price column. **Cleaned\_df** was saved as a csv file called **final\_data.csv**. using the to\_csv functionality from PANDAS.

Transformation for the **brazil\_sales\_ps** was mostly done in excel. I first only kept rows where the Playstation was the author/company. I also eliminated all rows that did not have a value for the price\_std\_or\_min. This was used in replace of drop null values. There were a few other rows dropped that did not have enough complete information to include- “stars”, and “reviews”. After the initial clean up I put this into pandas and into a dataframe, **brazil\_sales**. Then converted to csv in pandas as well naming it **final\_data1**.

# Load

With the newly minted csv file, we then decided to use pyMongo module and MongoDB as its final resting place. We called the database **videoGames** and called the collection **videoGamesSales** to illustrate the main purpose of the database.

Using pyMongo module, we wrote a script to connect to a localhost to create a local database. In the same script, we created the database and collection above and, with quick research, figured out how to turn our data in csv into json and feed that information into the database with pyMongo.

The reason we chose MongoDB and pyMongo is due to their relative freshness in our finds and fresh interface with MongoDB Compass. Another reason is the availability of resources and documentations for them.

For the loading of the second database in MongoDB, I followed all the same steps as previous database.

I followed the script in pandas to set up a connection to localhost. I created a database names **BrazilVideoGames** and collection. After the collection is set-up I used the script of converting csv to json so that information can be fed into the database.