ETL: NCAA College Football Statistics

Kimberley Reeves, Kevin Clark, Poonam Goel, Rebekah Rowland

Original Project Proposal:

- **Data sources:** We choose to scrape ESPN.com for both full team data and individual player statistics.
- **What Transformations:** Loading website HTML into Jupyter notebooks to be converted into pandas dataframes. Then, converting the data into a JSON and uploading into MongoDB for use as an API later down the road
- **Where is data going to loaded:** Jupyter notebooks, then to MongoDB

Extract:

- We decided to use BeautifulSoup to scrape the URLs that were set up in HTML to get the passing, receiving, and rushing stats for all 130 NCAA teams.
    - **Full Team URL:** https://www.espn.com/college-football/teams
    - **Individual Team URLS: 130 Teams** Example:https://www.espn.com/college-football/team/stats/_/id/2229/florida-international-panthers

Transform:

- The HTML string itself was set up pretty clearly and fairly easy to parse. Using a four loop, we were able to import the individual team URLs into a Pandas Dataframe. From there, we were able to clean up the multiple statistical tables into three clean dataframes: passing, rushing, and receiving. We then imported those data frames into a dictionary, converting the dataframes into JSON strings that were ready to be uploaded.

```
def stats_table_merge(stats_tables):
    passing=pd.merge(stats_tables[0], stats_tables[1],  left_index=True, right_index=True)
    rushing=pd.merge(stats_tables[2], stats_tables[3],  left_index=True, right_index=True)
    receiving=pd.merge(stats_tables[4], stats_tables[5],  left_index=True, right_index=True)
    team_data={ "passing":passing.to_json(),
        "rushing":rushing.to_json(),
        "receiving":receiving.to_json()
         }
    return(team_data)
```

Load:

We chose to load our information into MongoDB because we wanted to avoid creating schemas and we had uniform tables which made it easy to upload without having to worry about primary keys.

```
client = MongoClient('localhost', 27017)
db = client['passing_db']
collection_passing = db['passing']

with open('C:\\Users\\Poona\\Desktop\\Passing_DataFrame.json') as f:
    file_data = json.load(f)

# use collection_passing.insert(file_data) if pymongo version < 3.0
collection_passing.insert_many(file_data)
client.close()
```

Final MongoDB:

The final collection has the passing, rushing, and receiving statistics saved as JSON strings for 130 NCAA teams.