

|  |
| --- |
| ETL Project |
|  |
| November 16  Doug Watola  Tushar Makhija |

|  |
| --- |
| IntroductionWhat is StatsBomb and why we chose their data for our ETL Project?  * [StatsBomb](https://statsbomb.com/) is a soccer data collection company, that also provides consultations to major soccer clubs around the world. * Among other things, soccer clubs find this company helpful in scouting players and manager. * StatsBomb data is extensive, but not free, and also usually available only to businesses * Fortunately for us, they put out some free data on [their GitHub](https://github.com/statsbomb/open-data) :   + This gives a glimpse to their data and,   + Enthusiastic people like us can practice on it * Our vision with this ETL project is to create a meaningful dataset from a player scouting point of view. Therefore the end goal is to create data on each player and measure things like – pass rate (successful passes / attempted passes), conversion rate (goals scored / shots taken) etc. * In interest of time, we only were able to demonstrate pass rate, but the process can be replicated for similar or more complex player stats |
| Step 1: Extract  * Created a clone of this repo: <https://github.com/statsbomb/open-data> * The data is in 3 subfolders named:   + Events   + Lineups   + Matches * Lineups has 637 .JSON files. Each file represents one soccer game. These files have the list of players that made an appearance for each team in the game * Events has 637 .JSON files also, which has a 1-1 relationship with lineups files. These files are the core of the data and have details about everything that happened in the game. Each pass made, shot taken, goal etc. * Formal documentation is on their GitHub: <https://github.com/statsbomb/open-data/tree/master/doc> * We did not touch “Matches” subfolder yet in interest of time * The repo we saved all our work on is this: <https://github.com/tushmakster/stats_bomb_etl> * Out of the 637 .JSON files we only used 30 to not increase processing time. They are saved on our GitHub repo.  Step 2: Transformation  * All the work done is in a notebook here: [insert link] * Libraries we used were pandas, os, json, and sqlalchemy * From lineups, we created a unique list of players who had any appearances * From events, we got passing and shooting data, for each player, for each game * We want the transformation to be more in depth and more robust but in interest of time we were not able to  Step 3: Loading  * The last chunk of the notebook referred above, we export the data into a postgres SQL table * Our SQL code is saved here: [insert link]. Note that I redacted my username/password so be careful if you are trying to recreate everything. * The vision from there is to have a master table, and multiple SQL views for various kinds of summaries. * One schema we had in mind was to have 2 SQL views – one for player-based stats, the other for team stats * Below is a screenshot of one of our final view |