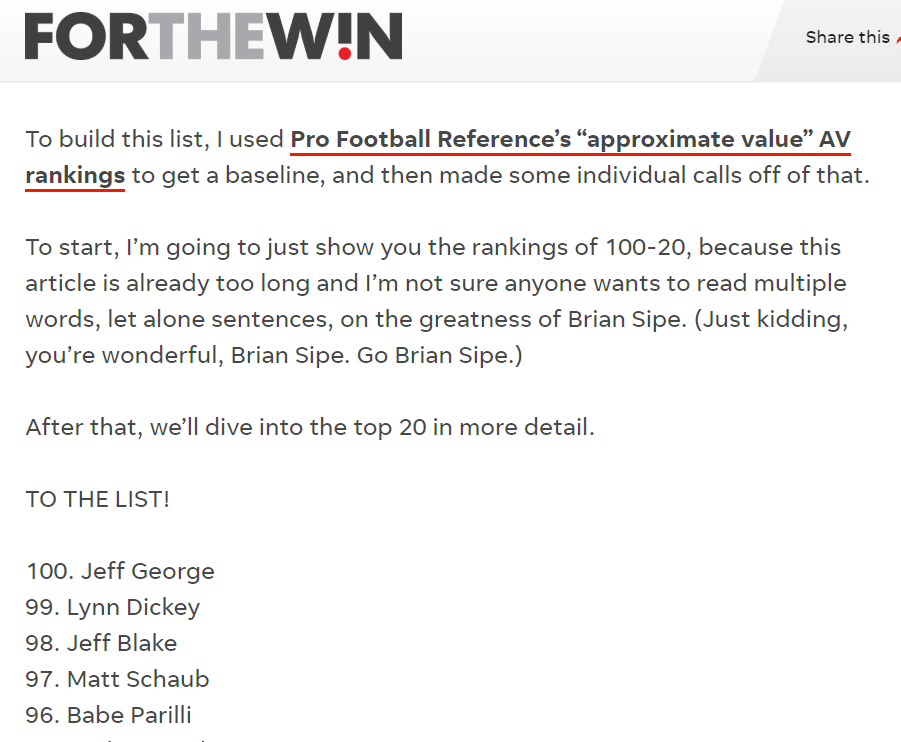
**Data Cleanup & Analysis:**

Data Sources –

[Pro Football Reference](https://www.pro-football-reference.com/leaders/pass_yds_career.htm)

We used csv files exported from the Pro Football Reference website. You can use the search tool below to generate the quarterback stat charts. To get the tables, you have to select a stat to analyze, specify career or season stats, choose all available data to date (2019) then click the green “Go!” button. Once the table is generated you can click the ‘Share & more’ toolbar to export the chart to a csv file that we can manipulate.



[USA Today](https://ftw.usatoday.com/2019/09/nfl-100-best-quarterbacks)

In order to get our list of all-time quarterback rankings we used BeautifulSoup to scrape the HTML code from the USA ‘FOR THE WIN’ sports blog to get a list of names to then compile data for. We needed Their list in order to perform lookup functions in excel (index/match) to then build our data set.

**Type of Transformation:**

For Transformation we mainly used Microsoft Excel and Python via Jupyter Notebook. We generated a list of quarterback names and rankings in Jupyter Notebook from our website scrape then converted the list to a pandas dataframe to then write that dataframe to a csv file. This was our subject csv file to perform our lookup functions to pull data (index/match) from our exported csv’s (12 total).

We then used Excel to clean up the data using “find and replace” functions, “text-to-columns”, “concatenate” and some simple calculations.

**Type of final production database:**

Mongo DB (non-relational)

**The final tables or collections:**

‘use nfl’

db.collection.find()

**ETL:**

Our goal is to analyze the rankings from the article (and possible bias) with relevant data from all-time passing statistics from the subject quarterbacks.

**Extract** - CSV (Pro Football Reference - stats) | HTML (usa today rankings)

**Transform** - Joined python dictionary of all data from both sources

**Load** - Mongo DB to store the data