Step	Input	➔	Process	➔	Output	Description
1	Wikipedia: Teams by Metropolitan Area	➔	Save to CSV	➔	cities.csv	**Create Master List of Cities.** 1) Navigate to Wikipedia page, 2) copy & paste results into CSV. Fields: • WikipediaMetro • City • State
2	US Census: ➔American FactFinder: ➔➔Community Facts	➔	Save to CSV	➔	City_Demographic.csv	**Create Demographic Data Set.** 1) Navigate to FactFinder UI, 2) search for each city, 3) copy & paste results into CSV
3	• cities.csv • City_Demographic.csv	➔	Run Jupyter Notebook "demo_df.ipynb"	➔	demo_df.csv	**Cleanse Demographic Data Set.** Run Jupyter Notebook to 1) replace "None"s with nulls, 2) save results as a Postgres-ready CSV
4	Bureau of Economic Analysis: ➔Regional Economic Accounts: ➔➔Annual GDP by Metropolitan Statistical Area	➔	Save to CSV	➔	MAGDP9_2001_2017_ALL_AREAS.csv	**Create Economic Data Set.** 1) Navigate to BEA download page, 2) choose "Annual GDP by Metropolitan Statistical Area", 3) extract Real GDP file from downloaded zip
5	St. Louis Fed: ➔Case-Shiller House Price Indexes: ➔➔A-L Cities	➔	Save to CSV	➔	fredgraph1.csv	**Create House Price Data Set (A-L Cities).** 1) Navigate to Fred graph page, 2) click "Download" to create CSV.
6	St. Louis Fed: ➔Case-Shiller House Price Indexes: ➔➔M-Z Cities	➔	Save to CSV	➔	fredgraph2.csv	**Create House Price Data Set (M-Z Cities).** 1) Navigate to Fred graph page, 2) click "Download" to create CSV.
7	cities.csv	➔	Enrich Master List of Cities with 2017 Metropolitan Area names	➔	cities.csv	**Modify Master List of Cities.** Manually add new field to cities.csv to accommodate mapping to 2017 economic data. Field: • 2017Metro
8	• cities.csv • fredgraph1.csv • fredgraph2.csv • MAGDP9_2001_2017_ALL_AREAS.csv	➔	Run Jupyter Notebook "econdata.ipynb"	➔	• econdata.csv • housedata.csv	**Cleanse Economic Data Set (1).** Run Jupyter Notebook to 1) merge Case-Shiller data sets, 2) drop unneeded Case-Shiller columns, 3) drop unneeded Real GDP cities, 4) drop low-level Real GDP industry classifications, 5) drop pre-2007 Real GDP data, 6) save results as intermediate CSVs.
9	econdata.csv	➔	Run Jupyter Notebook "econ_df.ipynb"	➔	econ_df.csv	**Cleanse Economic Data Set (2).** Run Jupyter Notebook to 1) replace "None"s with nulls, 2) cast 2017 column as a float, 3) save results as a Postgres-ready CSV.
10	ESPN MLB Attendance	➔	Save to CSV	➔	mlb_attendance.csv	**Create Attendance Data Set (MLB).** 1) Navigate to ESPN site, 2) search for each city, 3) copy & paste results into CSV
11	ESPN NBA Attendance	➔	Save to CSV	➔	nba_attendance.csv	**Create Attendance Data Set (NBA).** 1) Navigate to ESPN site, 2) search for each city, 3) copy & paste results into CSV
12	ESPN NFL Attendance	➔	Save to CSV	➔	nfl_attendance.csv	**Create Attendance Data Set (NFL).** 1) Navigate to ESPN site, 2) search for each city, 3) copy & paste results into CSV

Step	Input	➔	Process	➔	Output	Description
13	ESPN NHL Attendance	➔	Save to CSV	➔	nhl_attendance.csv	**Create Attendance Data Set (NHL).** 1) Navigate to ESPN site, 2) search for each city, 3) copy & paste results into CSV
14	cities.csv	➔	Enrich Master List of Cities with sports teams names	➔	sports_city.csv	**Create Sports_City Data Set.** 1) modify team neams in CSV to avoid data redundancy.
15	• mlb_attendance.csv • nba_attendance.csv • nfl_attendance.csv • nhl_attendance.csv • sports_city.csv	➔	Run Jupyter Notebook "attendance_transformation.ipynb"	➔	attendance.csv	**Create Attendance Data Set.** 1) import dependencies and read all attendance CSV files, 2) create LEAGUE column to identify associated sports, 3) concatenate all imported attendance data CSV files into single dataframe, 4) create dictionary from sports_city.csv to hold team and city association, 4) create function to loop through dictionary and identify team names and associated city, 5) add CITY column to dataframe and apply function to add city to column based on team name, 6) drop unwanted columns from dataframe, 7) rename and re-order columns
16	• attendance.csv • cities.csv • demo_df.csv • econ_df.csv • housedata.csv	➔	Run Postgres script "schema.sql"	➔	Postgres Tables • attendance • cities • demo • econ • housing • sport_city	**Create Postgres Tables.** 1) create each individual table and add column names with specific data types, 2) import CSVs into tables, 3) alter tables by dropping unwated columns, 4) combine all tables by creating an inner join through mutual columns