**INTRODUCTION**

The Spanish Artists Dictionary (SAD) is a reference source created by scholars at the Frick Art Reference Library. Originally a print publication, the dictionary was formatted as a Filemaker database in the early 1990s and made available through the Frick’s online research portal. The database consists of 5,186 records describing Spanish artists; these records include information about artist name, alternate names, dates (birth, death, and/or activity), field of artistic endeavor, bibliographic references, and Frick Photoarchive holdings.

The data from SAD’s three Filemaker tables has been made available for this project as CSV files.

**VISUALIZATIONS**

Diana’s explanation here!

**NAME MATCHING**

We also wrote a script to address a problem with linking from SAD records to the library catalog. From a SAD record, when a user clicks on one of the three links to a library catalog—the Frick’s library catalog (FRESCO), Arcade (the NYARC catalog), or Worldcat)—the search box that appears in the catalog is populated from a field in the Filemaker database labeled “LC Name.” However, many entries in SAD have incorrect or outdated names in the “LC Name” field, or they do not include a name in the “LC Name” field, in which case the name is populated from the main name field. This produces faulty results (i.e. Goya, Francisco, 1746-1828 ***not*** Goya, Francisco de, 1746-1828) (1181 vs. 2 hits in FRESCO).

To address this problem, the SAD\_fresco.py script compares the main name entry in the SAD database with a list of Spanish names pulled from the library catalog’s subject heading list. A portion of the CSV version of the database (comprised of the 374 female artist names in SAD) was parsed using the CSV DictReader method. Each row of the CSV became a dictionary object. Two methods were then used from FuzzyWuzzy, a string matching library for Python. The .token\_set\_ratio method breaks two strings (in this case an artist names) into words and compares the degree of similarity between the two strings. The .ratio method compares the individual characters in each string. A match was considered those results that achieved a .token\_set\_ratio above 90 and a .ratio above 80. These matches were then inserted back into the Python dictionary rows to populate a new field called FRESCOSH. The updated data was written as a CSV file, which can now be uploaded back to the FileMaker database.