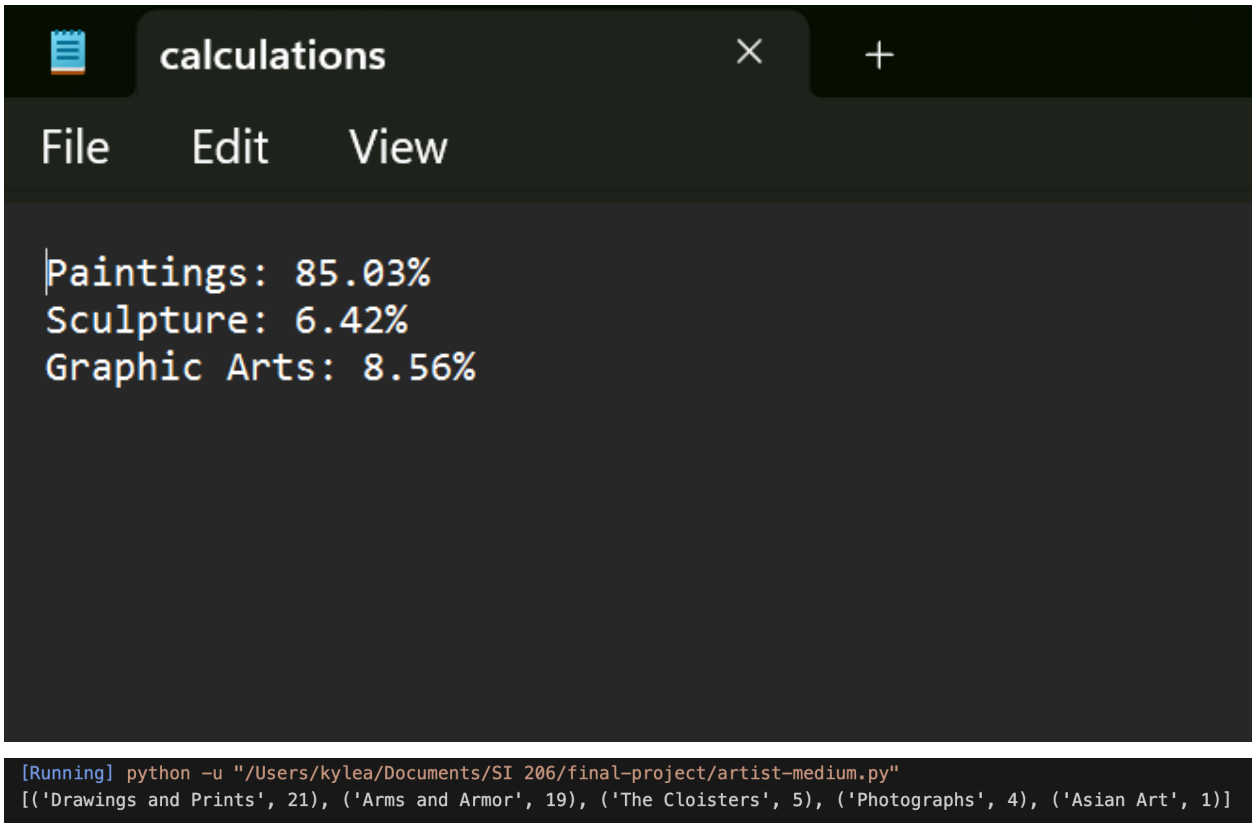
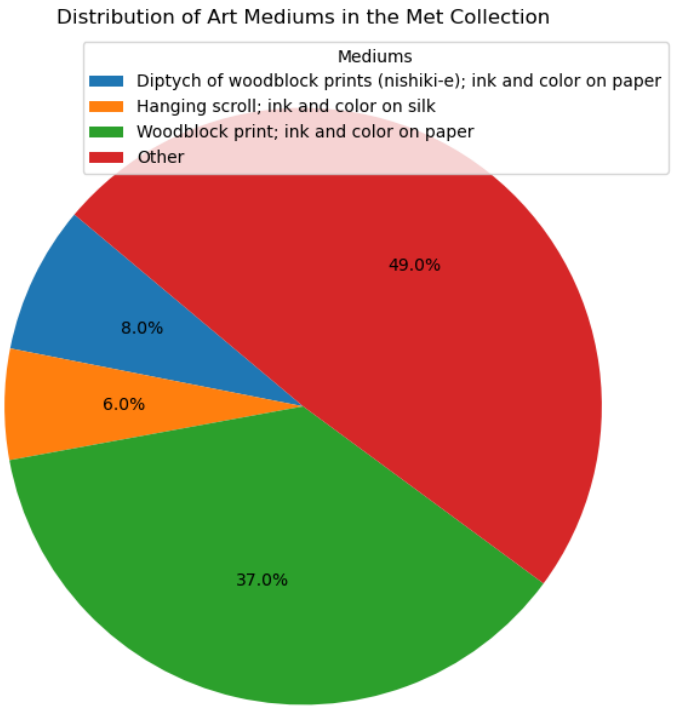
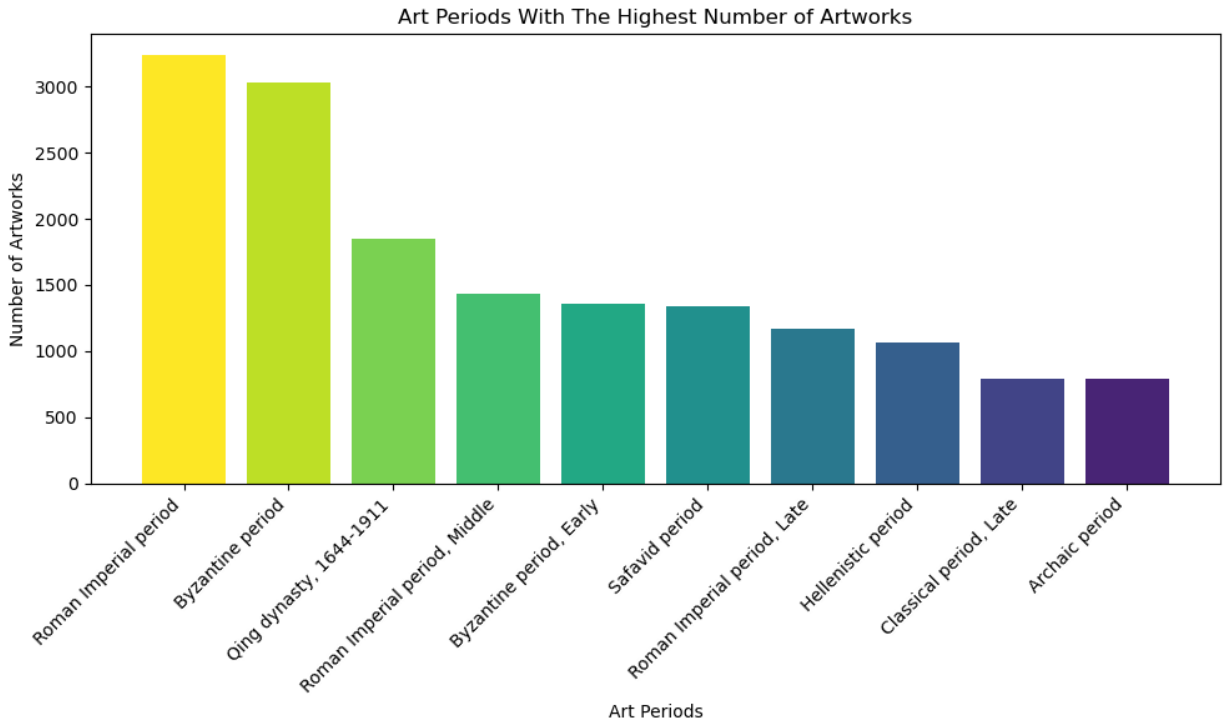
SI 206 Final Project Report

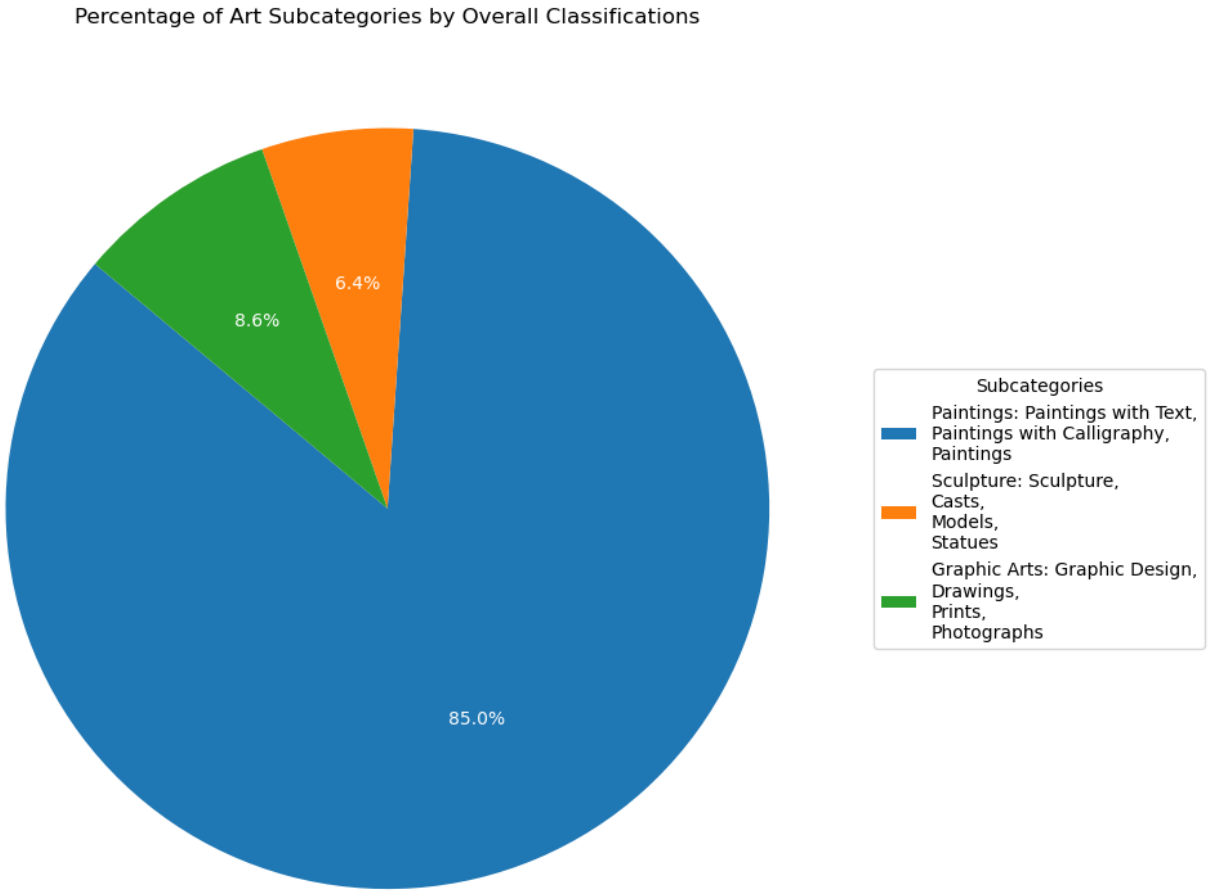
1. The goals for your project including what APIs/websites you planned to work with and what data you planned to gather (10 points)
    a. Our goal for this project was to identify popular art periods from museums. In this case, we worked with the Met Museum of Art and Harvard Art Museums API's.
2. The goals that were achieved including what APIs/websites you actually worked with and what data you did gather (10 points)
    a. The Met Museum of Art and Harvard Art Museums API's were used to gather our data. We specifically looked at popular art periods and the type of art medium most seen at these museums.
3. The problems that you faced (10 points)
    a. The most challenging part of the project was learning the API structure. We used a JSON formattor Chrome plugin to effectively view the data. From there, it was easier to understand what information we needed to add to our code to add the accompanying data to our database.
4. The calculations from the data in the database (i.e. a screenshot) (10 points)

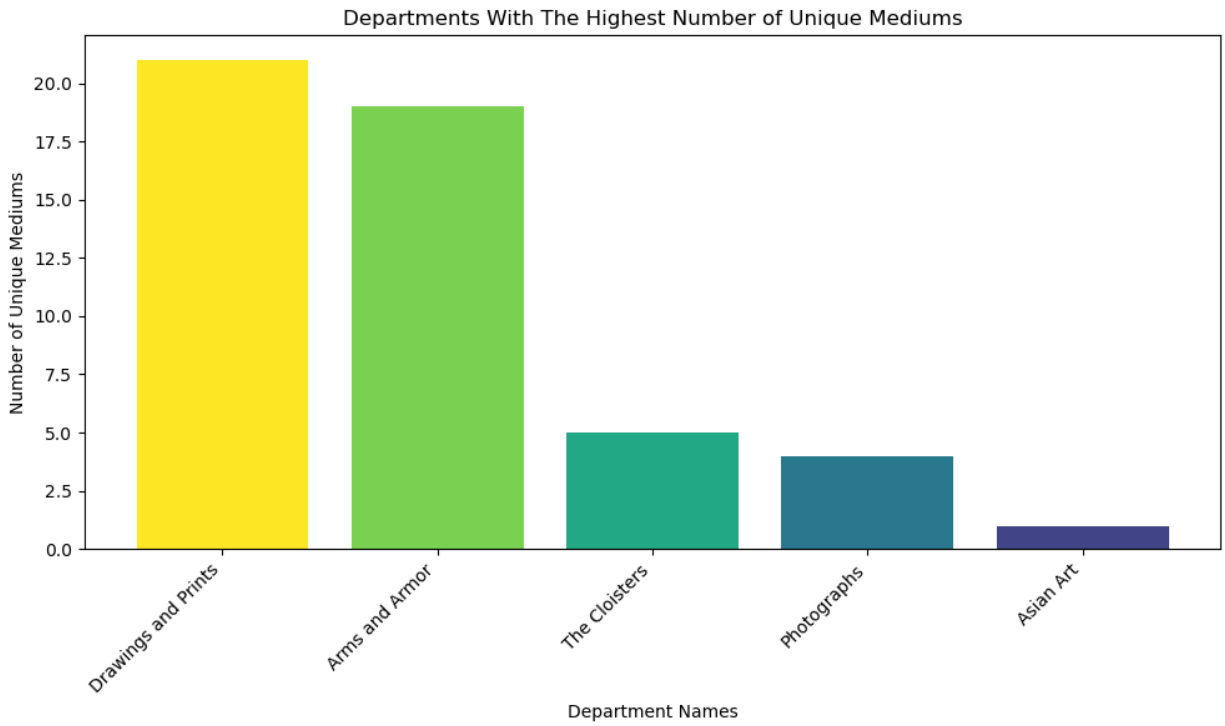


5. The visualization that you created (i.e. screenshot or image file) (10 points)




Art Periods With The Highest Number of Artworks
Number of Artworks
0
500
1000
1500
2000
2500
3000
Roman Imperial period
Byzantine period
Qing dynasty, 1644-1911
Roman Imperial period, Middle
Byzantine period, Early
Safavid period
Roman Imperial period, Late
Hellenistic period
Classical period, Late
Archaic period
Art Periods


Percentage of Art Subcategories by Overall Classifications
6.4%
8.6%
85.0%
Subcategories
Paintings: Paintings with Text, Paintings with Calligraphy, Paintings
Sculpture: Sculpture, Casts, Models, Statues
Graphic Arts: Graphic Design, Drawings, Prints, Photographs



6. Instructions for running your code (10 points)
    a. Download the following files:
        i. final_harvard_gather_data.py
        ii. final_harvard_process_data.py
        iii. met.py
        iv. dept_medium.py
    b. Open each file in Visual Studio Code. Double check that the database to be created is called museums.db.
    c. Run the files met.py, final_harvard_gather_data.py, and dept_medium.py. These files will create the database museums.db with the tables with information from both the Met Museum of Art and Harvard Art Museums API's. The Met visuals should be generated from this output.
    d. Run the final_harvard_process_data.py to generate the Harvard visuals.
7. Documentation for each function that you wrote. This includes describing the input and output for each function (20 points)

final_harvard_process_data.py:

plot_top_classifications function:

Input:

- `cur` (sqlite3.Cursor): SQLite database cursor.
- `classification_data` (list of tuples): List containing tuples of classification data from the database.
- `top_n` (int): Number of top classifications to include in the chart.

Output:

- `plt` (matplotlib.pyplot): Matplotlib pyplot object containing the pie chart.

write_data_to_file function:

Input:

- `data` (dict): Dictionary containing data to be written to the file.
- `output_file` (str): Path to the output text file.

Output:

- None

plot_top_periods function:

Input:

- `cur` (sqlite3.Cursor): SQLite database cursor.
- `period_data` (list of tuples): List containing tuples of period data from the database.
- `top_n` (int): Number of top periods to include in the chart.

Output:

- `plt` (matplotlib.pyplot): Matplotlib pyplot object containing the bar chart.

process_and_visualize_data function:

Input:

- None

Output:

- None

final_harvard_gather_data.py:

get_harvard_info function:

Input:

- `api_key` (str): API key for accessing the Harvard Art Museums API.
- `endpoint` (str): API endpoint for the desired data.
- `params` (dict, optional): Additional parameters for the API request.

Output:

- `harvard_data` (dict): JSON response containing data from the API.

set_up_database function:

Input:

- db (str): Name of the SQLite database.

Output:

- `cur` (sqlite3.Cursor): SQLite database cursor.
- `conn` (sqlite3.Connection): SQLite database connection.

insert_data function:

Input:

- `cur` (sqlite3.Cursor): SQLite database cursor.
- `table` (str): Name of the table to insert data into.
- `data` (list of dicts): List of dictionaries containing data to be inserted.
- `page` (int): Page number for tracking data in the database.

Output:

- None

set_up_table function:

Input:

- `api_key` (str): API key for accessing the Harvard Art Museums API.
- `cur` (sqlite3.Cursor): SQLite database cursor.
- `conn` (sqlite3.Connection): SQLite database connection.
- `table` (str): Name of the table to set up.
- `max_items` (int, optional): Maximum number of items to fetch per API request.
- `additional_rows` (int, optional): Number of additional rows to fetch.

Output:

- None

gather_data function:

Input:

- `api_key` (str): API key for accessing the Harvard Art Museums API.
- `additional_rows` (int, optional): Number of additional rows to fetch.

Output:

- None

Main block (if name == 'main'):

Input:

- None

Output:

- None

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
12/12/23	API Key was not working correctly, said unauthorized access	https://api-toolkit.herokuapp.com/6	Yes
12/12/23	Wanted to make each individual bar a different color in bar graph	https://matplotlib.org/stable/tutorials/pyplot.html#sphx-glr-tutorials-pyplot-py	Yes

Git repo: https://github.com/ngocho07/206-final-project/tree/master