Repo Link: CLICK HERE

Report on Marvel APIs

Stan Leevasz

PART 5 – Report (100 points)

In addition to your API activity results, you will be creating a report for your overall project. The report must include:

1. The goals for your project including what APIs/websites you planned to work with and what data you planned to gather (10 points)
2. The goals that were achieved including what APIs/websites you actually worked with and what data you did gather (10 points)
3. The problems that you faced (10 points)
4. The calculations from the data in the database (i.e. a screen shot) (10 points)
5. The visualization that you created (i.e. screen shot or image file) (10 points)
6. Instructions for running your code (10 points)
7. An updated function diagram with the names of each function, the input, and output and who was responsible for that function (20 points)
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?**

Planned: Project Goal, APIs Used, & Data Gathered

- The project's goal is to investigate the representation of different identities in superhero media.
- APIs Used:
  - Superhero API: https://superheroapi.com/try-now.html
  - Comic Vine API: https://comicvine.gamespot.com/api/documentation
  - Marvel Rivals API: https://marvelrivalsapi.com/
  - MCU Countdown: https://github.com/DiljotSG/MCU-Countdown
- Gathered Data:
  - Height & weight of superheroes, gender of superheroes, birthplace of superheroes, movies & comics appearances of superheroes.
  - Seeing representation in:
    - BMI
    - Gender
    - Birthplace
    - Comics
    - Movies
    - Games

Achieved: Project Goal, APIs Used, & Data Gathered

- The project's goal is to investigate the representation of different identities in superhero media.
- APIs Used:
  - Superhero API: https://superheroapi.com/try-now.html
  - Comic Vine API: https://comicvine.gamespot.com/api/documentation
  - Marvel Rivals API: https://marvelrivalsapi.com/
- Gathered Data:
  - Height & weight of superheroes, gender of superheroes, birthplace of superheroes, & comics appearances of superheroes.
  - Seeing representation in:
    - BMI
    - Gender
    - Birthplace
    - Comics
    - Games

Problems Faced

- Had to transition from using MCU Countdown to Marvel Rivals API
  - MCU Countdown only gave us data for the upcoming movie, not previous Marvel films
- No movie data
  - Comic Vine's documents weren't up to date
    - we couldn't find how many movies certain heroes were featured in
    - Moved to number of comics heroes were in
- 429 Errors
  - Largely occurred when problem solving data issues
- Excluding non-Marvel heroes from lists

Calculations from Data in Database - Blake

```python
def most_played_characters(file, filename):
    """
    Arguements: str file, str filename

    Return: None

    Iterating through the characters used and ordering them
    through most played in a text file for later use
    """
    results = {"Male": 0, "Female": 0, "Unknown": 0}
    cur, conn = marvel_rivals.set_up_database(file)

    cur.execute(
        """
        SELECT mr_id, gender_id FROM superheros
        """
    )

    hero_lst = cur.fetchall()

    for hero in hero_lst:
        count = 0
        id = hero[0]

        if id != None:
            for i in range(1,6):
                #iterating over a single match at a time so I'm not getting the whole row
                cur.execute(
                    f"""
                    SELECT character_by_match.match{i} FROM character_by_match
                    WHERE character_by_match.match{i} = {id}
                    """
                )

                #the length of fetchall should contain number of times character is used
                num = len(cur.fetchall())
                count += num

            cur.execute(
                f"""
                SELECT genders.gender FROM genders
                WHERE genders.gender_id = {hero[1]}
                """
            )

            gender = cur.fetchone()[0]

            results[gender] += count

try:
    open(filename)
except:
    #Only writing if the file doesn't already exist
    with open(filename, "w") as f:
        f.write("gender, number of times played")
        for key in list(results.keys()):
            f.write(f"\n{key},{results[key]}")

conn.close()
pass
```

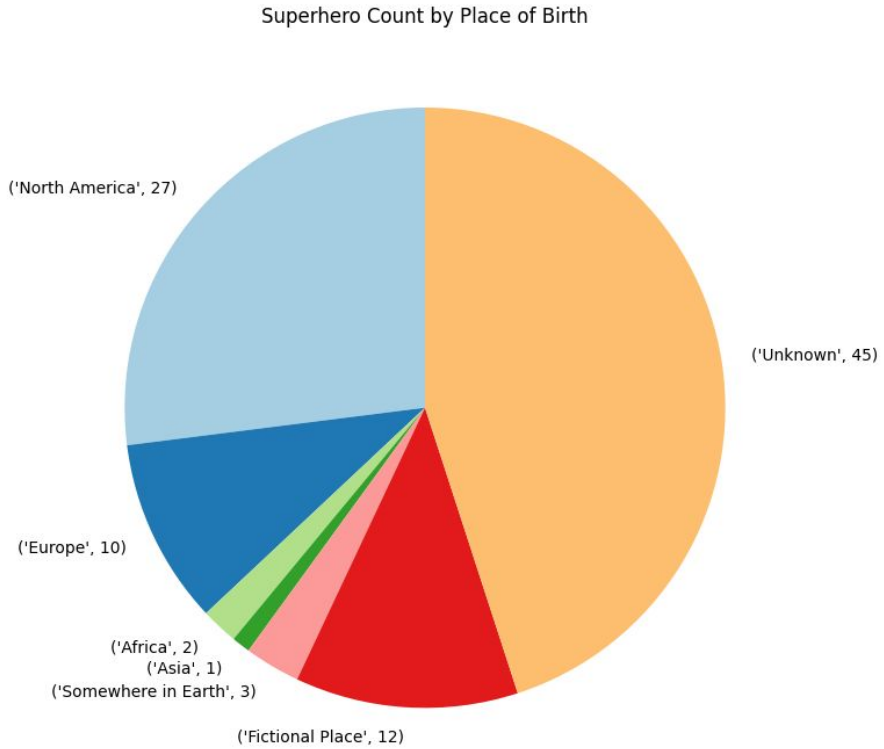
```python
def gender_by_comics(db, filename):
    """
    Aruements: db, filename

    Returns: None

    Going through the superhero.db to grab how many issues different genders feature in
    """
    cur, conn = comicvine.set_up_database(db)
    results = {"Male": 0, "Female": 0, "Unknown": 0}

    cur.execute(
        """
        SELECT numcomics.comics, genders.gender FROM superheros
        JOIN numcomics ON superheros.num_comics = numcomics.id
        JOIN genders ON superheros.gender_id = genders.gender_id
        """
    )

    heroes = cur.fetchall()

    for hero in heroes:
        if hero[0] != "NULL":
            num_comics = int(hero[0].split(" ")[0])
        else:
            num_comics = 0
        results[hero[1]] += num_comics

    try:
        open(filename)
    except:
        with open(filename, "w") as f:
            f.write("gender, number of issues")
            for key in list(results.keys()):
                f.write(f"\n{key},{results[key]}")

    conn.close()
    pass
```

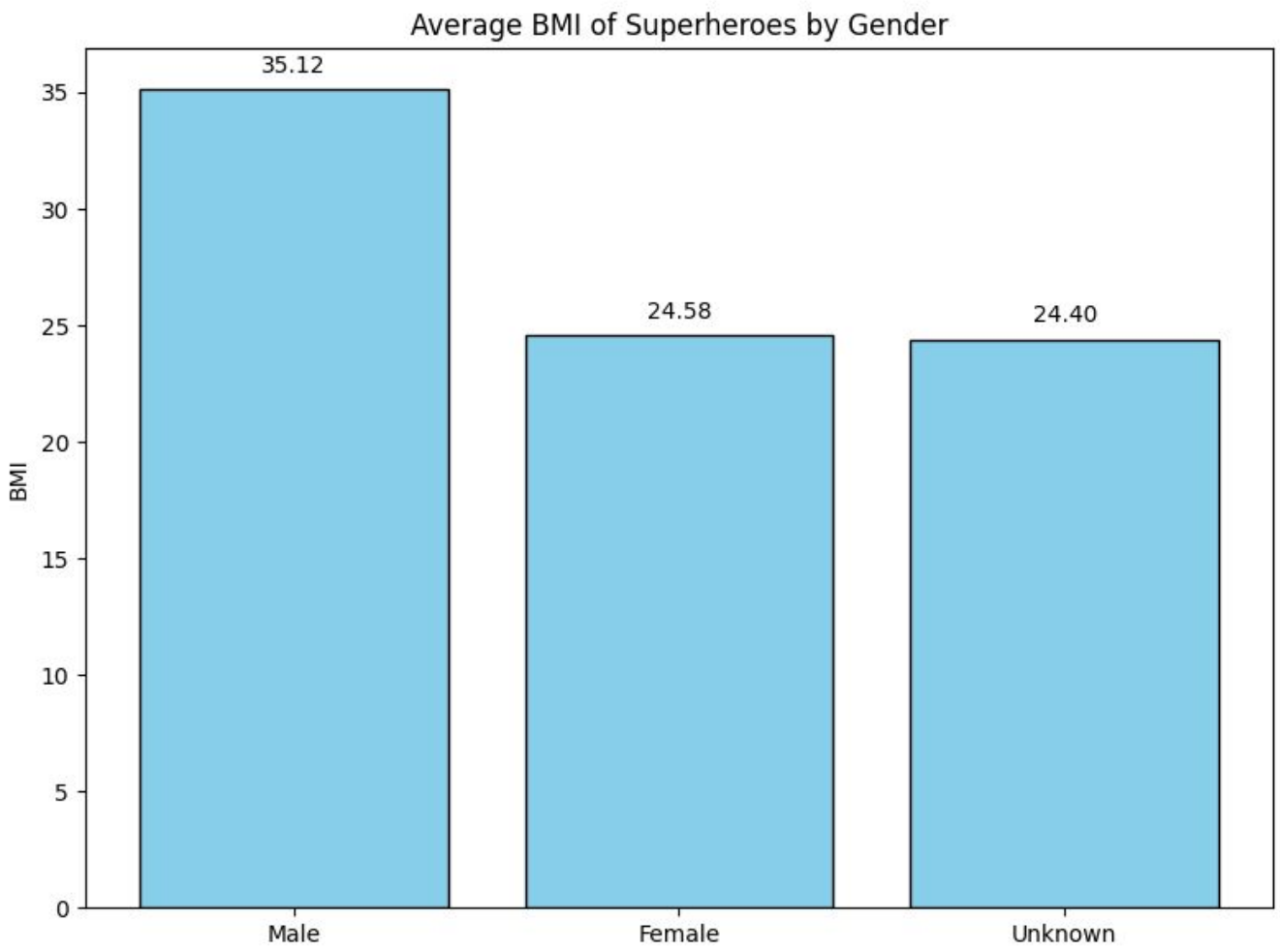
Calculations from Data in Database - Stan

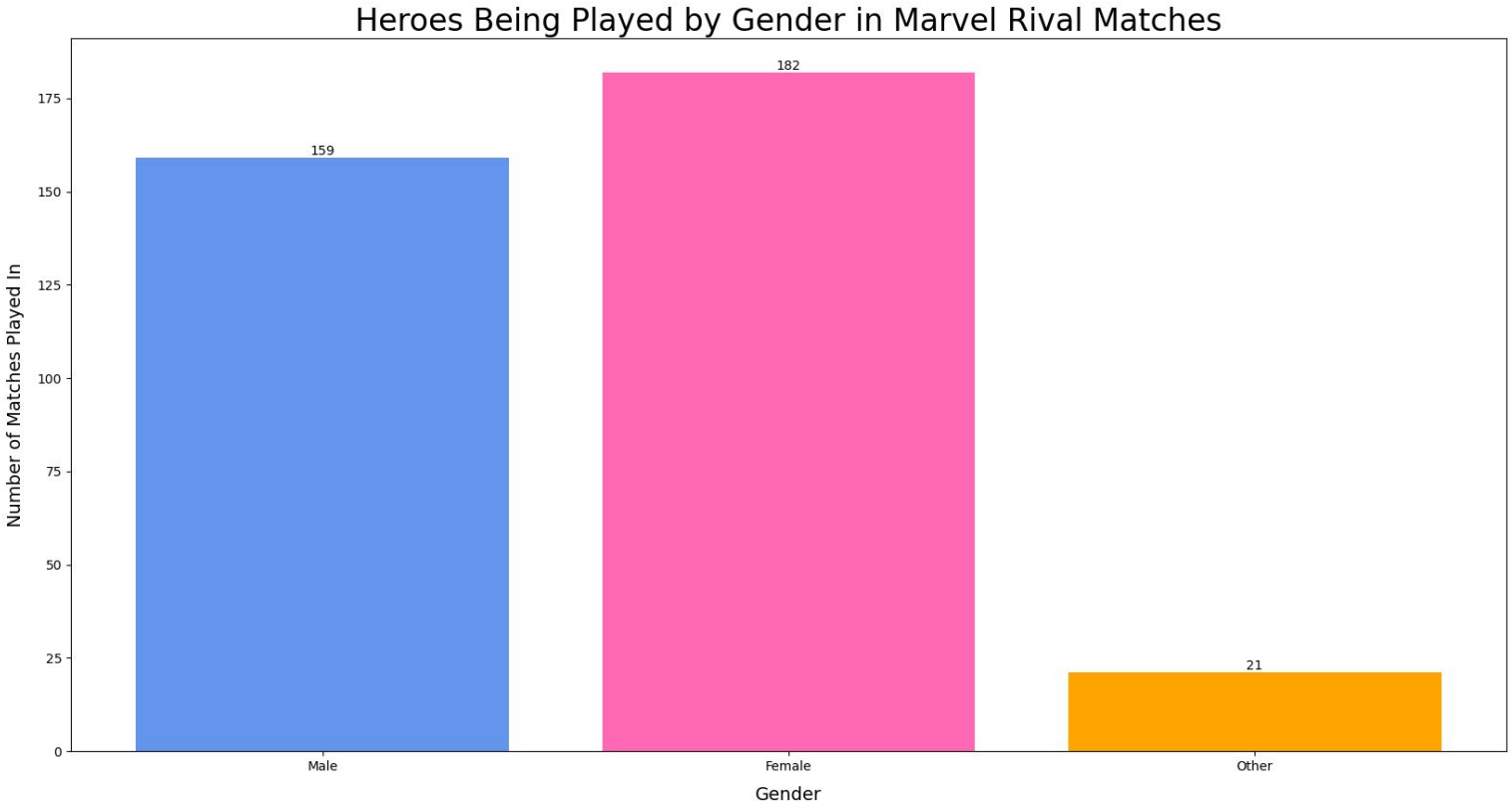
```python
def superhero_api_calculations(db_file, filename, filename2):
    """
    Args: str db_file, str filename
    Out: none
    Calculates the count of superhero for each place or birth and average BMI for supehero by gender
    """
    conn, cur = superhero.setup_db(db_file)
    cur.execute("""
        SELECT places_of_birth.place_of_birth
        FROM superheros
        JOIN places_of_birth
        ON superheros.place_of_birth_id = places_of_birth.place_of_birth_id
    """)
    places_of_birth = cur.fetchall()
    places_count = {
        "North America": 0,
        "Europe": 0,
        "Africa": 0,
        "Asia": 0,
        "Somewhere in Earth": 0,
        "Fictional Place": 0,
        "Unknown": 0
    }
    for place in places_of_birth:
        if place[0] in places_count:
            places_count[place[0]] += 1
    try:
        open(filename)
    except:
        with open(filename, "w") as f:
            for origin, count in places_count.items():
                f.write(f"There are {count} superheros from {origin}\n")
```

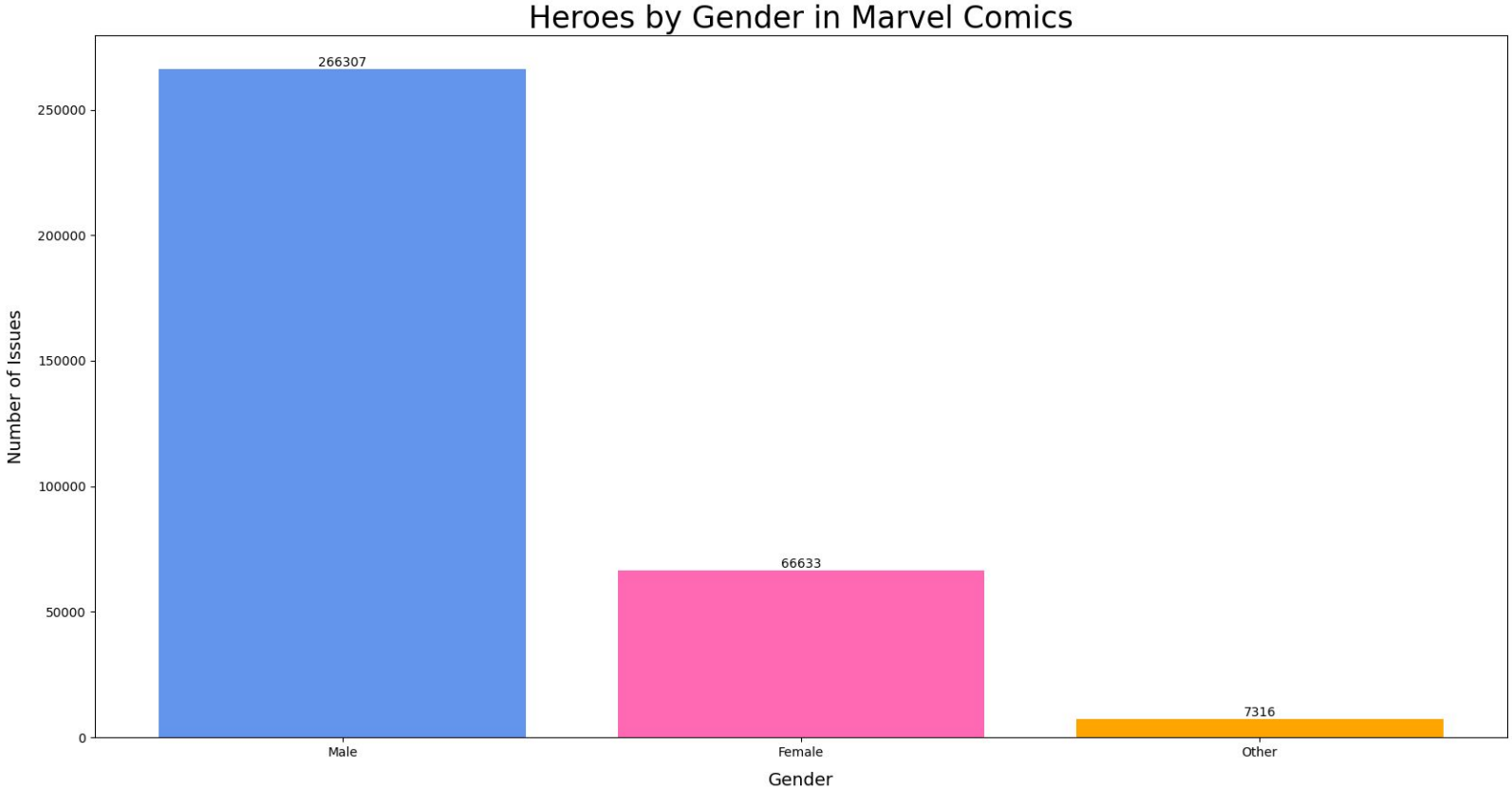
```python
cur.execute("""
    SELECT genders.gender, superheros.height, superheros.weight
    FROM superheros
    JOIN genders
    ON superheros.gender_id = genders.gender_id
""")
gender_height_weight = cur.fetchall()
gender_counts = {"Male": 0, "Female": 0, "Unknown": 0}
gender_bmi = {"Male": 0, "Female": 0, "Unknown": 0}
for g, height, weight in gender_height_weight:
    if g in gender_counts and height and weight:
        try:
            bmi = weight / ((height / 100) ** 2)
            gender_counts[g] += 1
            gender_bmi[g] += bmi
        except Exception:
            continue
try:
    open(filename2)
except:
    with open(filename2, "a") as f:   # append so both results are in same file
        for gender, total_bmi in gender_bmi.items():
            if gender_counts[gender] > 0:
                avg_bmi = total_bmi / gender_counts[gender]
                f.write(f"The average BMI for {gender} superheros is {avg_bmi:.2f}\n")
            else:
                f.write(f"No data for {gender} superheros\n")
```

Visualizations Created




Average BMI of Superheroes by Gender
BMI
35.12
24.58
24.40
0
5
10
15
20
25
30
35
Male
Female
Unknown


Heroes Being Played by Gender in Marvel Rival Matches
Number of Matches Played In
0
25
50
75
100
125
150
175
159
182
21
Male
Female
Other
Gender


Heroes by Gender in Marvel Comics
Number of Issues
0
50000
100000
150000
200000
250000
266307
66633
7316
Male
Female
Other
Gender

Instructions for Running the Code

1. Create an API key from https://superheroapi.com/, https://comicvine.gamespot.com/api/, and https://marvelrivalsapi.com/.

2. Insert each API key into the “replace text with key” on each corresponding line in the api_keys.txt file.

3. Run the superhero.py file 4 times first to create and populate the main database: superhero.db.

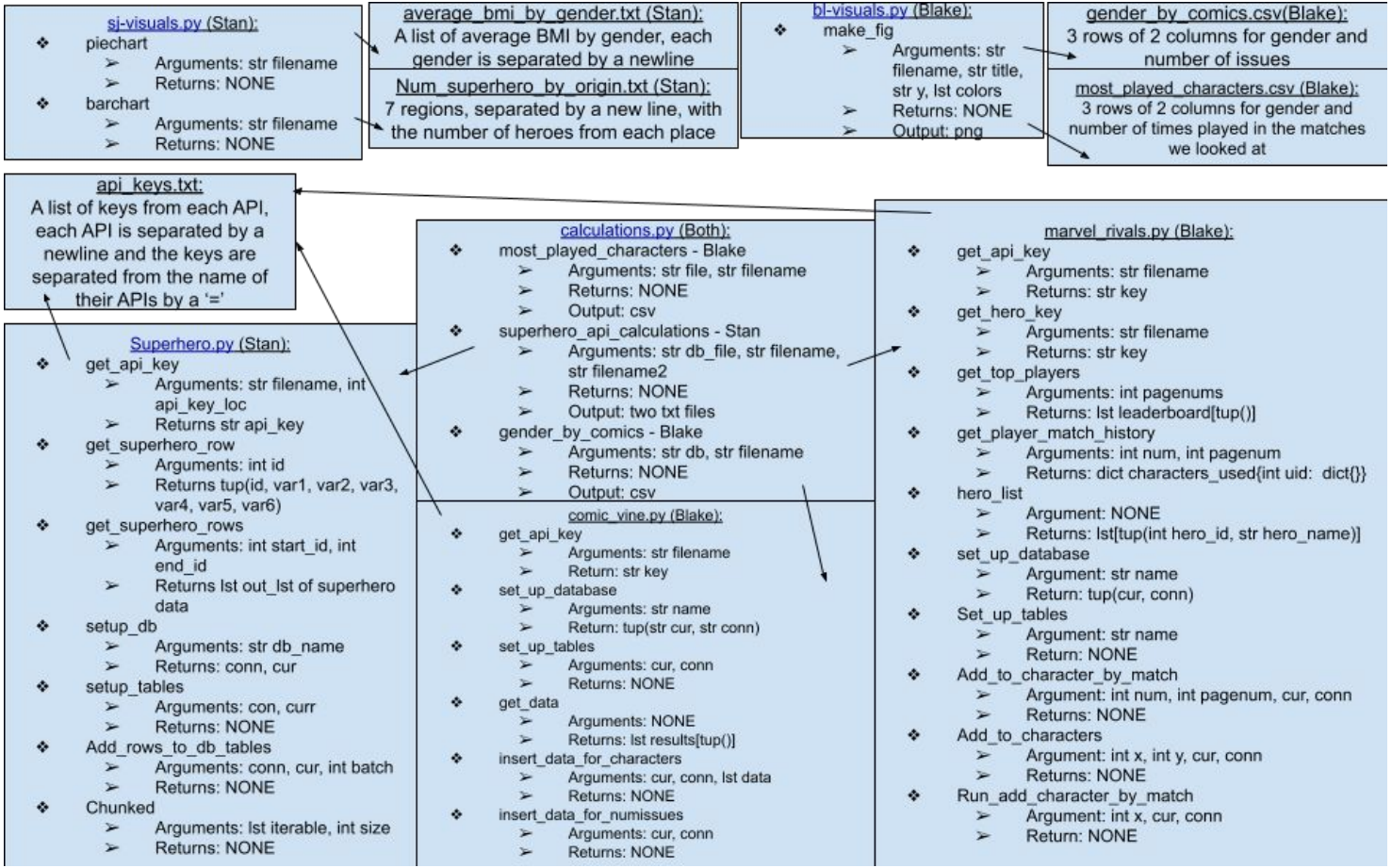
4. Run the marvel_rivals.py 4 times to create and populate its tables to the superhero.db database.

5. Run the comicvine.py file 9 times (or until "Finished searching for data!" prints) to create and populate its tables to the superhero.db database.

6. Run the calculations.py file to perform calculations using data from the superhero.db database to create the data calculations files: gender_by_comics.csv, most_played_characters.csv, num_superhero_by_origin.txt, and average_bmi_by_gender.txt.

5. Run the bl-visuals.py and sj-visuals.py files to create the visualizations from calculated data.

Function Diagram



Resources Used

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
11/18/2025	Difficulty in getting access to Marvel Rivals API key.	https://discord.com/invite/marvelrivalsapi	Yes, the issue was solved.
11/24/2025	Long runtime in creating a database for Superhero API.	https://prismic.io/blog/api-response-times	Yes, the issue was solved.
12/1/2025	References for colors and text for matplotlib for customization of the charts	https://matplotlib.org/stable/api/matplotlib_configuration_api.html	Yes, the issue was solved.