https://github.com/sabrenachedid/finalproject.git

https://github.com/hananhusein/Final_Project.git

https://github.com/juliamgs/finalproject.git

Project Goals:

The goal of this project was to scrape three APIs - one pertaining to movies/TV shows, one about monthly precipitation rates from 1902-2022, and another about Quran verses. We wanted to determine if there was a correlation between the release dates of the top and bottom 100 rated episodes of The Office and precipitation rates during those time periods. Additionally, we looked to see if there was any correlation between the text of those episodes and passages in the Quran. We scraped the TVmaze API to acquire the release dates and titles for the top and bottom 100 episodes of The Office. This data was compiled into tables for analysis. Next, we scraped precipitation rate data from the World Bank API at intervals aligned with the earliest and latest Office episode release dates. This allowed us to cross-check precipitation rates on the air dates of the show's highest and lowest-rated episodes. We calculated the average precipitation difference between the top and bottom 100 episode air dates.

Visualizations were created plotting the air dates for the top 100 and bottom 100 Office episodes against precipitation rates during those time periods. By comparing these graphs we could analyze any differential correlation with precipitation and episode rating.

Finally, we checked if words and phrases from the episode titles matched text from verses of the Quran. The average word matches were calculated. Ultimately our analyses did not uncover conclusive correlations between precipitation, Office episode ratings, and related Quran passages.

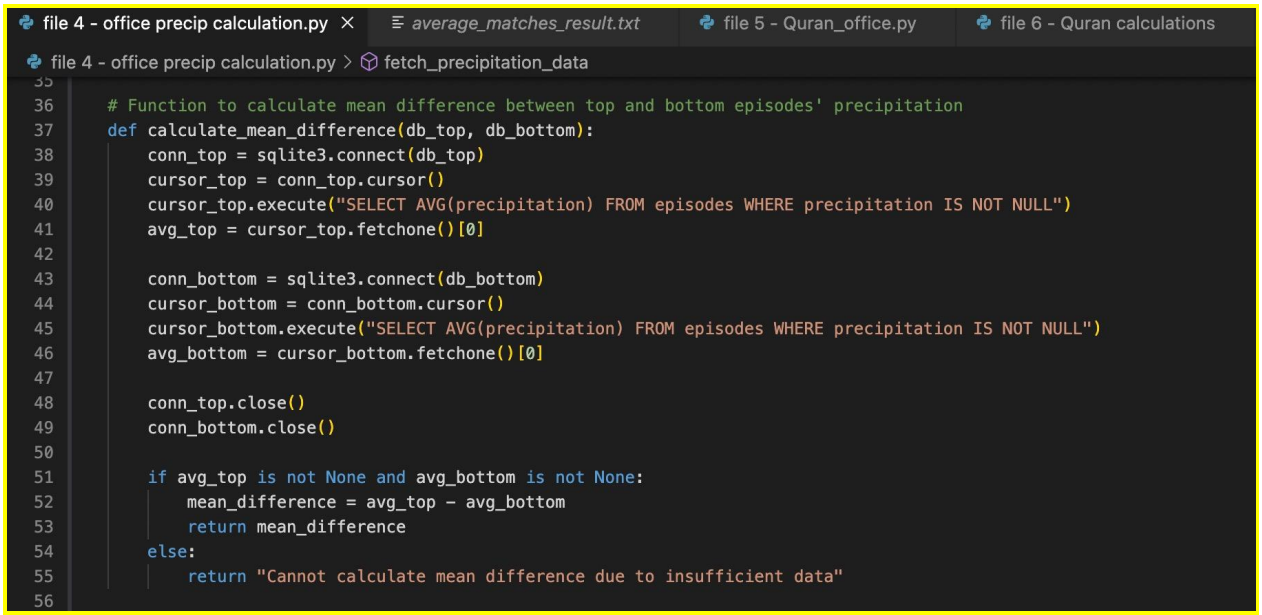
Problems:

Some problems that we encountered as a group mainly revolved around finding working APIs. Due to problems using the Spotify API, we had to completely change our research question to reflect the few APIs we were able to work with and we could not find free historic data APIs. Many of them were difficult to navigate and/or required a purchase. We struggled to implement certain APIs which resulted in us having to do numerous searches for free APIs that we were able to understand and use. We came across some struggles when using SQLite. The database kept duplicating the data we inputted and we had to rework our code to prevent duplication. Another problem we faced was inserting 25 points into the database at a time, as we could not use a for loop. We need to figure out how to change where the code picks up everytime it runs.

Calculations:

Calculations (SQLite Table and Code):

The Office-Precipitation calculation code and output:



file 4 - office precip calculation.py | average_matches_result.txt | file 5 - Quran_office.py | file 6 - Quran calculatic

file 4 - office precip calculation.py > fetch_precipitation_data

```
91
92          # Calculate the mean difference
93          mean_diff = calculate_mean_difference('office_top_precip.db', 'office_bottom_precip.db')
94
95          # Write the result to a text file
96          with open('mean_difference_result.txt', 'w') as file:
97              file.write(f'Mean difference in precipitation: {mean_diff}\n')
98
99          print(f'Mean difference written to mean_difference_result.txt: {mean_diff}')
100
101     if __name__ == "__main__":
102         main()
103
104
105
106
107
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Code

```
[Running] python -u "/Users/julia/Documents/umich/courses/SI206/Final project si206 folder/file 4 - office precip calculation.py"
Mean difference written to mean_difference_result.txt: 2.2506999999999806

[Done] exited with code=0 in 4.948 seconds
```

Quran calculation code and output:

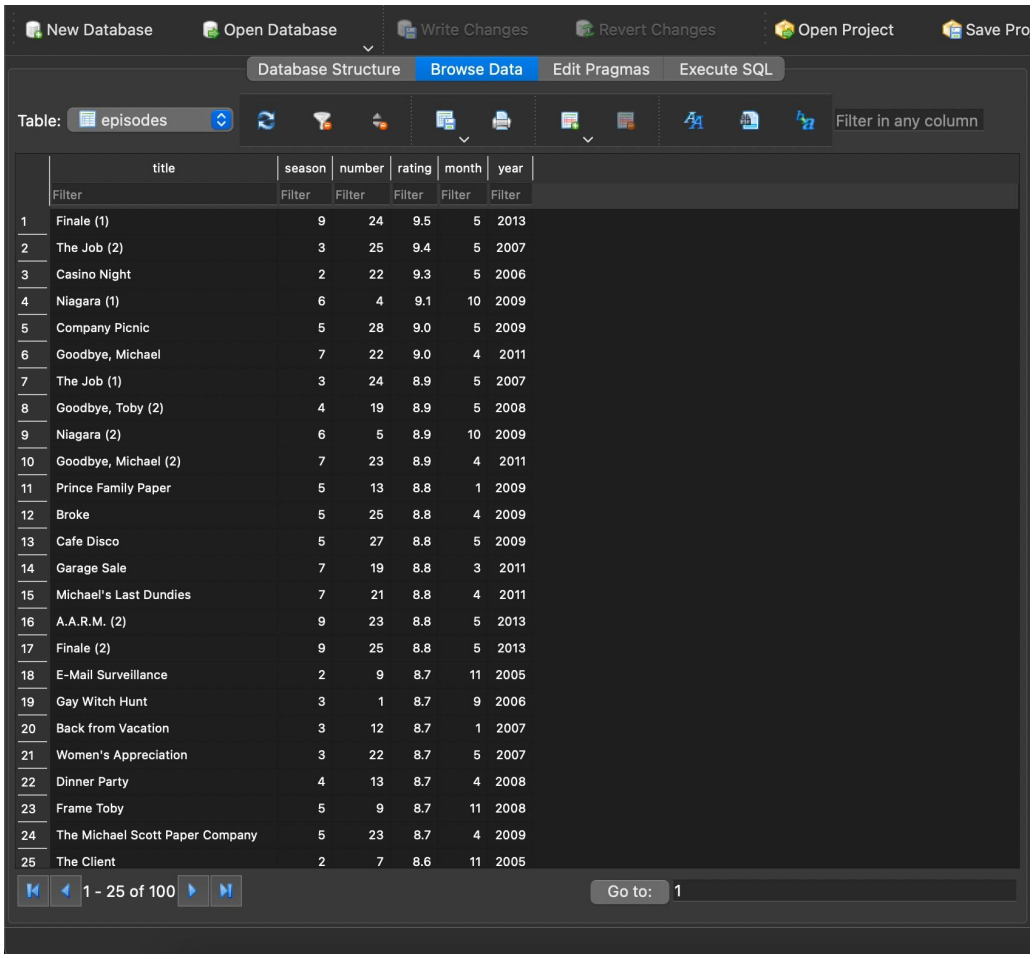
file 4 - office precip calculation.py | average_matches_result.txt | file 5 - Quran_office.py | file 6 - Quran calculations | file 2a - office top 1(

file 6 - Quran calculations > ...

```
40  def calculate_average_matches(episode_titles, quran_verses):
41      episode_words = Counter(word.lower().strip(punctuation) for title in episode_titles for word in title.split())
42      quran_words = Counter(word.lower().strip(punctuation) for verse in quran_verses for word in verse.split())
43      matched_words_count = sum(min(episode_words[word], quran_words[word]) for word in episode_words if word in quran_words)
44      total_words = len(episode_words)
45      average_matches = matched_words_count / total_words if total_words > 0 else 0
46      return average_matches
47
48  def insert_average_matches(conn, cur, average_matches):
49      cur.execute('CREATE TABLE IF NOT EXISTS average_matches (value REAL)')
50      cur.execute('INSERT INTO average_matches (value) VALUES (?)', (average_matches,))
51      conn.commit()
52
53  def main():
54      conn = sqlite3.connect('matching_words.db')
55      cur = setup_db(conn)
56
57      quran_verses = fetch_quran_text()
58      insert_data(conn, cur, 'quran_verses', 'verse', quran_verses)
59
60      all_episode_titles = fetch_all_episodes()
61      insert_data(conn, cur, 'episodes', 'title', all_episode_titles)
62
63      stored_quran_verses = get_all_data(cur, 'quran_verses', 'verse')
64      stored_episode_titles = get_all_data(cur, 'episodes', 'title')
65
66      average_matches = calculate_average_matches(stored_episode_titles, stored_quran_verses)
67      print(f"Average number of matched words in The Office episode titles and the Quran: {average_matches}")
68
69      insert_average_matches(conn, cur, average_matches)
70
71      # Write the result to a text file
72      with open('average_matches_result.txt', 'w') as file:
73          file.write(f'Average number of matched words: {average_matches}\n')
74
75      print(f'Average matches written to average_matches_result.txt: {average_matches}')
76
77      conn.close()
78
```

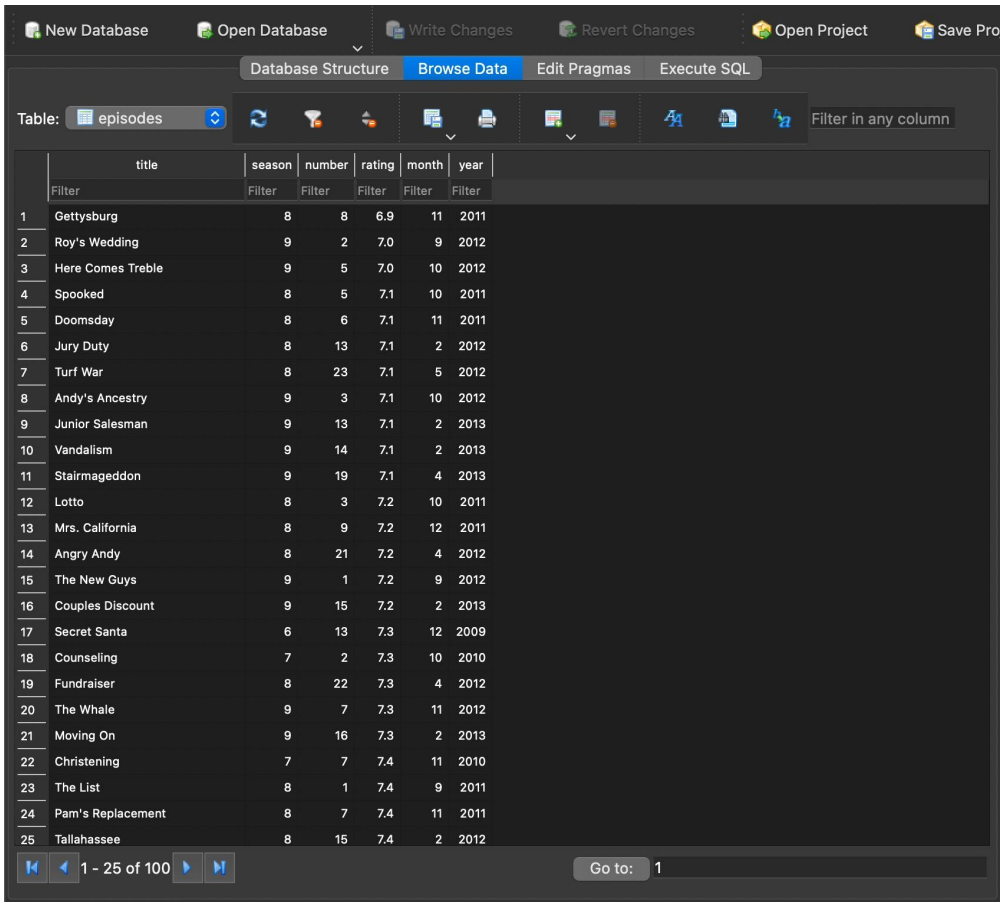
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
[Running] python -u "/Users/julia/Documents/umich/courses/SI206/Final project si206 folder/file 6 - Quran calculations"
Average number of matched words in The Office episode titles and the Quran: 0.3309608540925267
Average matches written to average_matches_result.txt: 0.3309608540925267

[Done] exited with code=0 in 1.51 seconds
```

The Office's top 100 episodes:



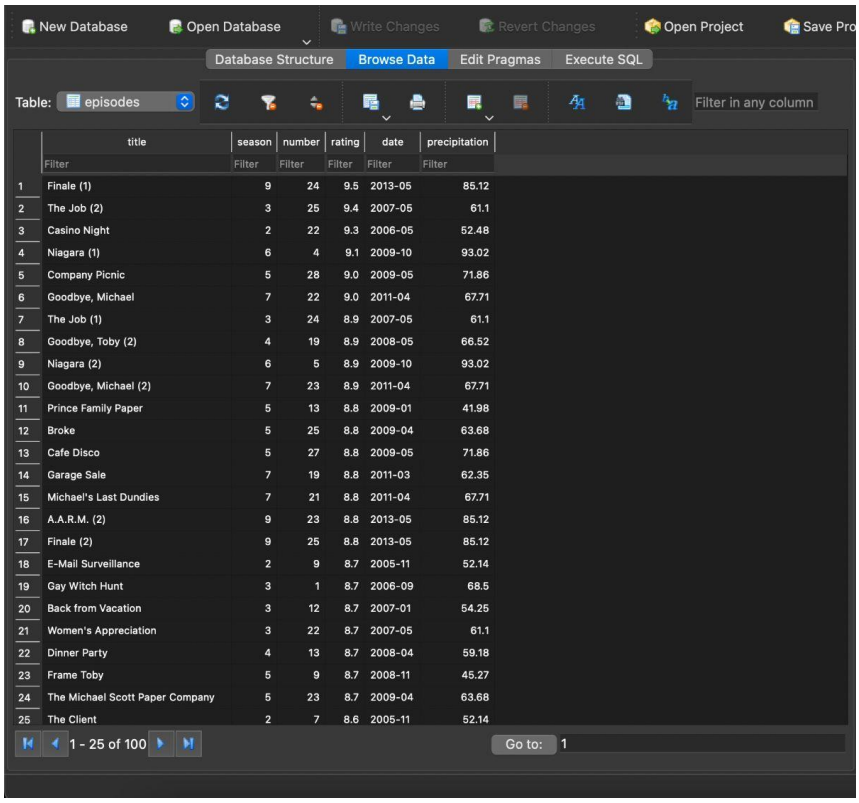
	title	season	number	rating	month	year
	Filter	Filter	Filter	Filter	Filter	Filter
1	Finale (1)	9	24	9.5	5	2013
2	The Job (2)	3	25	9.4	5	2007
3	Casino Night	2	22	9.3	5	2006
4	Niagara (1)	6	4	9.1	10	2009
5	Company Picnic	5	28	9.0	5	2009
6	Goodbye, Michael	7	22	9.0	4	2011
7	The Job (1)	3	24	8.9	5	2007
8	Goodbye, Toby (2)	4	19	8.9	5	2008
9	Niagara (2)	6	5	8.9	10	2009
10	Goodbye, Michael (2)	7	23	8.9	4	2011
11	Prince Family Paper	5	13	8.8	1	2009
12	Broke	5	25	8.8	4	2009
13	Cafe Disco	5	27	8.8	5	2009
14	Garage Sale	7	19	8.8	3	2011
15	Michael's Last Dundies	7	21	8.8	4	2011
16	A.A.R.M. (2)	9	23	8.8	5	2013
17	Finale (2)	9	25	8.8	5	2013
18	E-Mail Surveillance	2	9	8.7	11	2005
19	Gay Witch Hunt	3	1	8.7	9	2006
20	Back from Vacation	3	12	8.7	1	2007
21	Women's Appreciation	3	22	8.7	5	2007
22	Dinner Party	4	13	8.7	4	2008
23	Frame Toby	5	9	8.7	11	2008
24	The Michael Scott Paper Company	5	23	8.7	4	2009
25	The Client	2	7	8.6	11	2005

The Office's bottom 100 episodes:



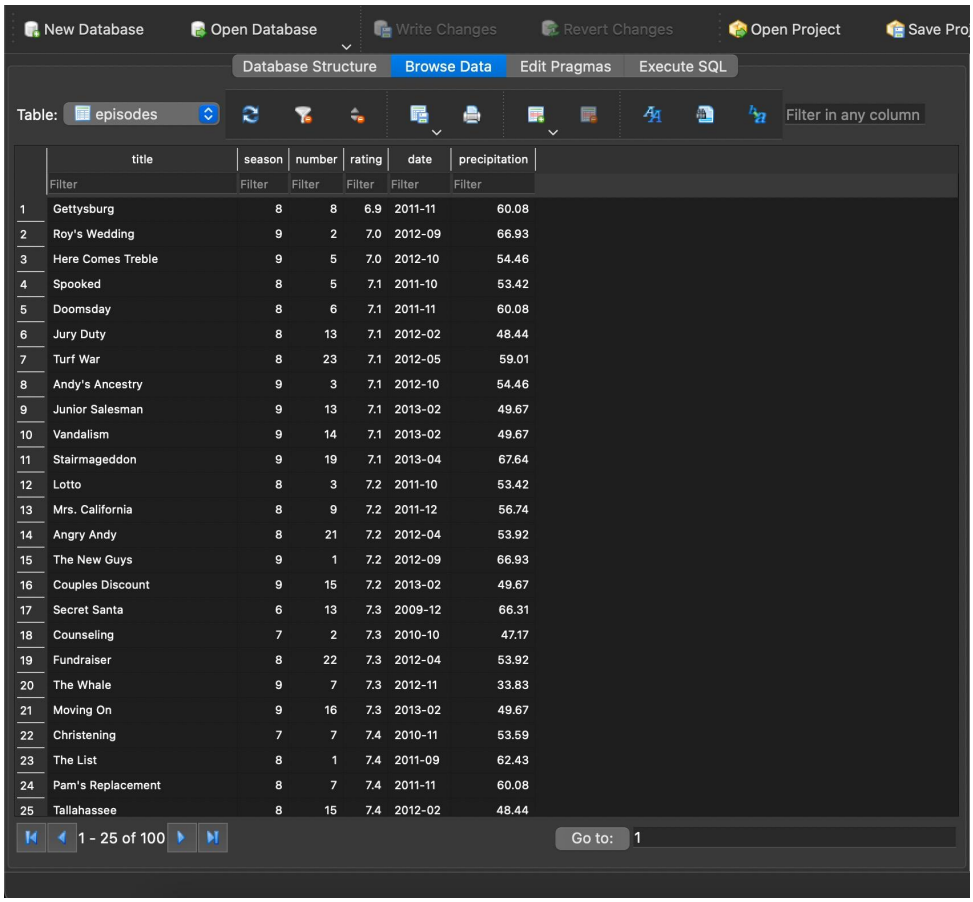
	title	season	number	rating	month	year
	Filter	Filter	Filter	Filter	Filter	Filter
1	Gettysburg	8	8	6.9	11	2011
2	Roy's Wedding	9	2	7.0	9	2012
3	Here Comes Treble	9	5	7.0	10	2012
4	Spooked	8	5	7.1	10	2011
5	Doomsday	8	6	7.1	11	2011
6	Jury Duty	8	13	7.1	2	2012
7	Turf War	8	23	7.1	5	2012
8	Andy's Ancestry	9	3	7.1	10	2012
9	Junior Salesman	9	13	7.1	2	2013
10	Vandalism	9	14	7.1	2	2013
11	Stairmageddon	9	19	7.1	4	2013
12	Lotto	8	3	7.2	10	2011
13	Mrs. California	8	9	7.2	12	2011
14	Angry Andy	8	21	7.2	4	2012
15	The New Guys	9	1	7.2	9	2012
16	Couples Discount	9	15	7.2	2	2013
17	Secret Santa	6	13	7.3	12	2009
18	Counseling	7	2	7.3	10	2010
19	Fundraiser	8	22	7.3	4	2012
20	The Whale	9	7	7.3	11	2012
21	Moving On	9	16	7.3	2	2013
22	Christening	7	7	7.4	11	2010
23	The List	8	1	7.4	9	2011
24	Pam's Replacement	8	7	7.4	11	2011
25	Tallahassee	8	15	7.4	2	2012

The Office's top 100 episodes and precipitation:



	title	season	number	rating	date	precipitation
	Filter	Filter	Filter	Filter	Filter	Filter
1	Finale (1)	9	24	9.5	2013-05	85.12
2	The Job (2)	3	25	9.4	2007-05	61.1
3	Casino Night	2	22	9.3	2006-05	52.48
4	Niagara (1)	6	4	9.1	2009-10	93.02
5	Company Picnic	5	28	9.0	2009-05	71.86
6	Goodbye, Michael	7	22	9.0	2011-04	67.71
7	The Job (1)	3	24	8.9	2007-05	61.1
8	Goodbye, Toby (2)	4	19	8.9	2008-05	66.52
9	Niagara (2)	6	5	8.9	2009-10	93.02
10	Goodbye, Michael (2)	7	23	8.9	2011-04	67.71
11	Prince Family Paper	5	13	8.8	2009-01	41.98
12	Broke	5	25	8.8	2009-04	63.68
13	Cafe Disco	5	27	8.8	2009-05	71.86
14	Garage Sale	7	19	8.8	2011-03	62.35
15	Michael's Last Dundies	7	21	8.8	2011-04	67.71
16	A.A.R.M. (2)	9	23	8.8	2013-05	85.12
17	Finale (2)	9	25	8.8	2013-05	85.12
18	E-Mail Surveillance	2	9	8.7	2005-11	52.14
19	Gay Witch Hunt	3	1	8.7	2006-09	68.5
20	Back from Vacation	3	12	8.7	2007-01	54.25
21	Women's Appreciation	3	22	8.7	2007-05	61.1
22	Dinner Party	4	13	8.7	2008-04	59.18
23	Frame Toby	5	9	8.7	2008-11	45.27
24	The Michael Scott Paper Company	5	23	8.7	2009-04	63.68
25	The Client	2	7	8.6	2005-11	52.14

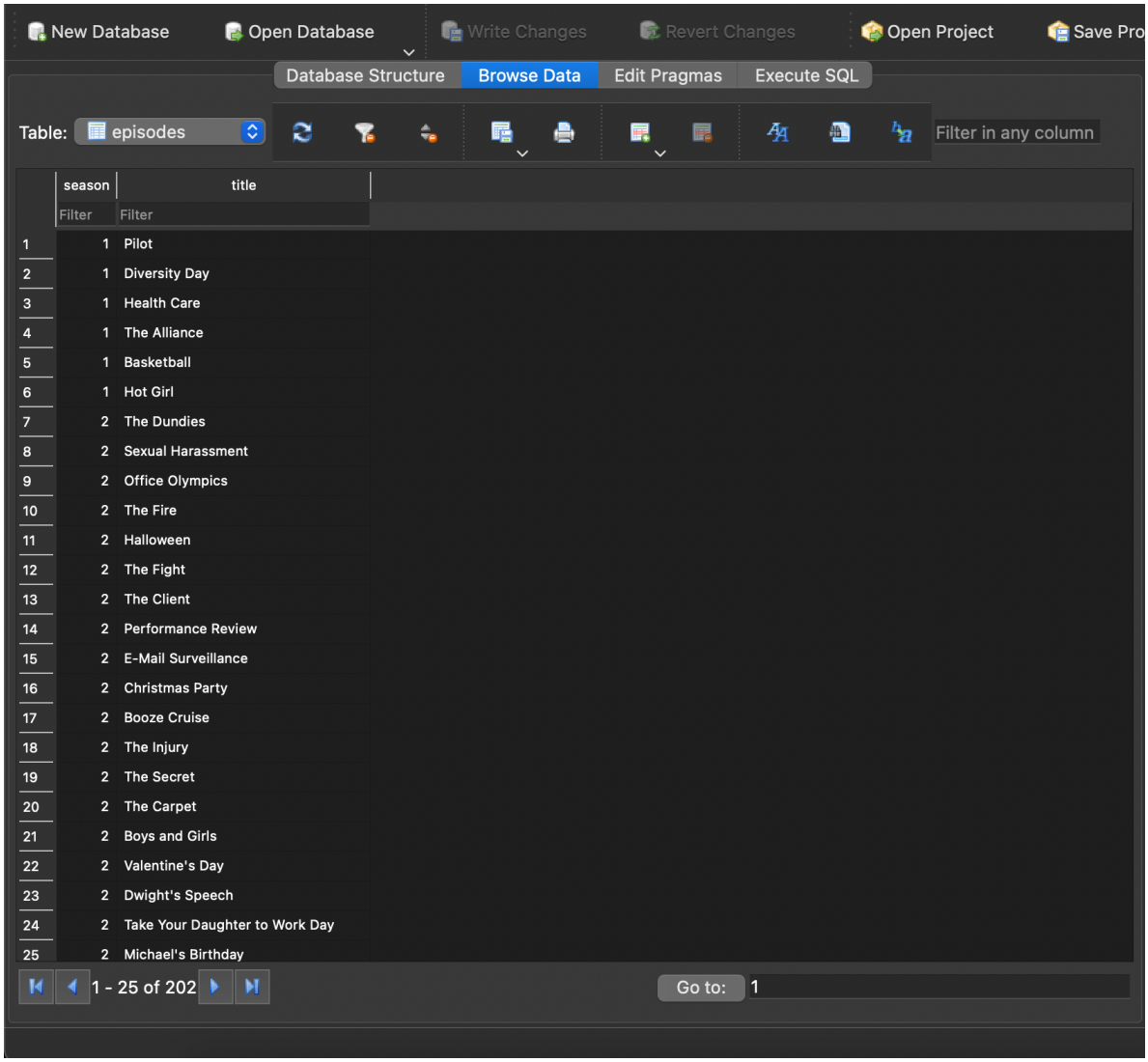
The Office’s bottom 100 episodes and precipitation:



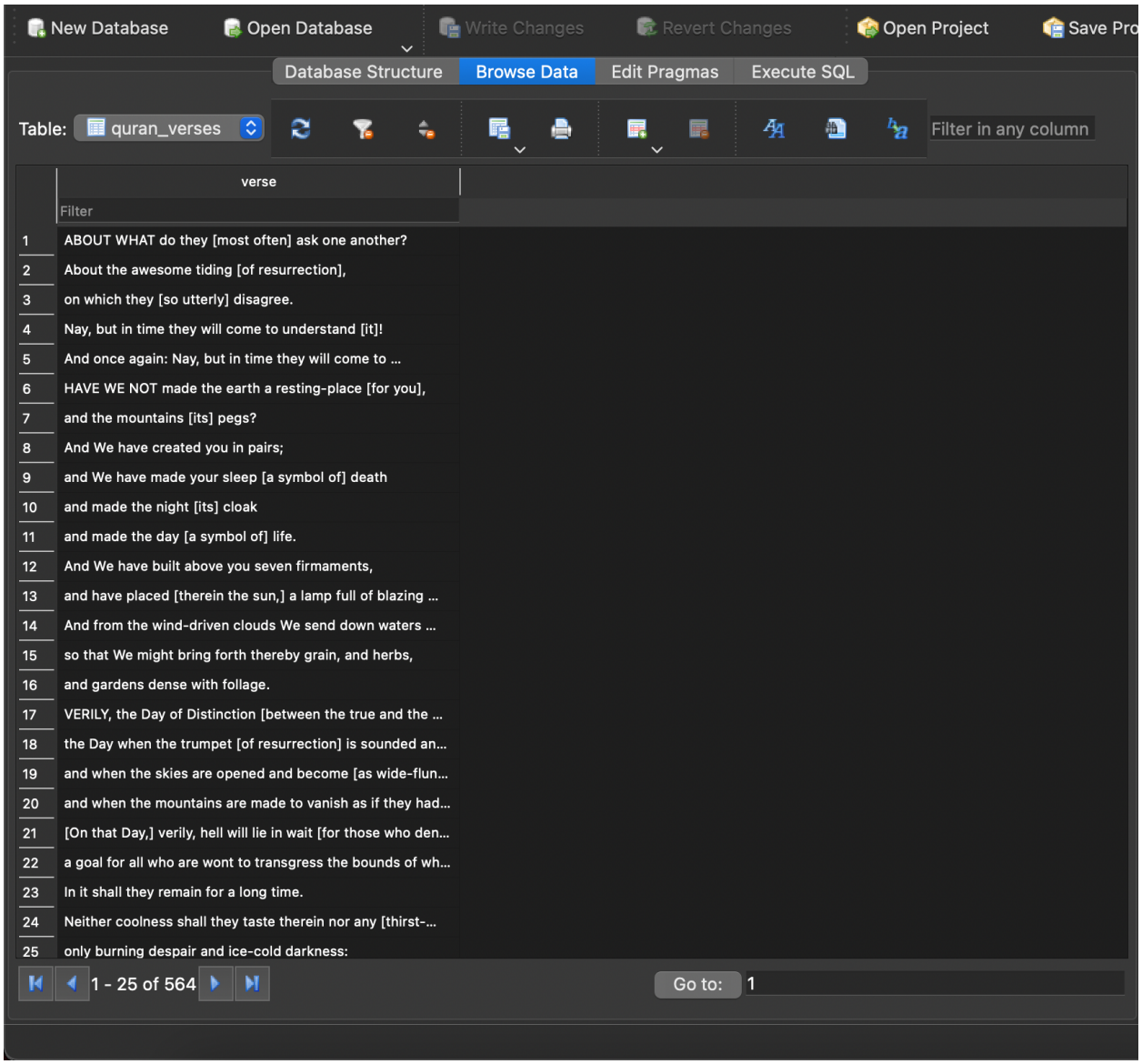
	title	season	number	rating	date	precipitation
	Filter	Filter	Filter	Filter	Filter	Filter
1	Gettysburg	8	8	6.9	2011-11	60.08
2	Roy's Wedding	9	2	7.0	2012-09	66.93
3	Here Comes Treble	9	5	7.0	2012-10	54.46
4	Spooked	8	5	7.1	2011-10	53.42
5	Doomsday	8	6	7.1	2011-11	60.08
6	Jury Duty	8	13	7.1	2012-02	48.44
7	Turf War	8	23	7.1	2012-05	59.01
8	Andy's Ancestry	9	3	7.1	2012-10	54.46
9	Junior Salesman	9	13	7.1	2013-02	49.67
10	Vandalism	9	14	7.1	2013-02	49.67
11	Stairmageddon	9	19	7.1	2013-04	67.64
12	Lotto	8	3	7.2	2011-10	53.42
13	Mrs. California	8	9	7.2	2011-12	56.74
14	Angry Andy	8	21	7.2	2012-04	53.92
15	The New Guys	9	1	7.2	2012-09	66.93
16	Couples Discount	9	15	7.2	2013-02	49.67
17	Secret Santa	6	13	7.3	2009-12	66.31
18	Counseling	7	2	7.3	2010-10	47.17
19	Fundraiser	8	22	7.3	2012-04	53.92
20	The Whale	9	7	7.3	2012-11	33.83
21	Moving On	9	16	7.3	2013-02	49.67
22	Christening	7	7	7.4	2010-11	53.59
23	The List	8	1	7.4	2011-09	62.43
24	Pam's Replacement	8	7	7.4	2011-11	60.08
25	Tallahassee	8	15	7.4	2012-02	48.44

1 - 25 of 100 Go to: 1

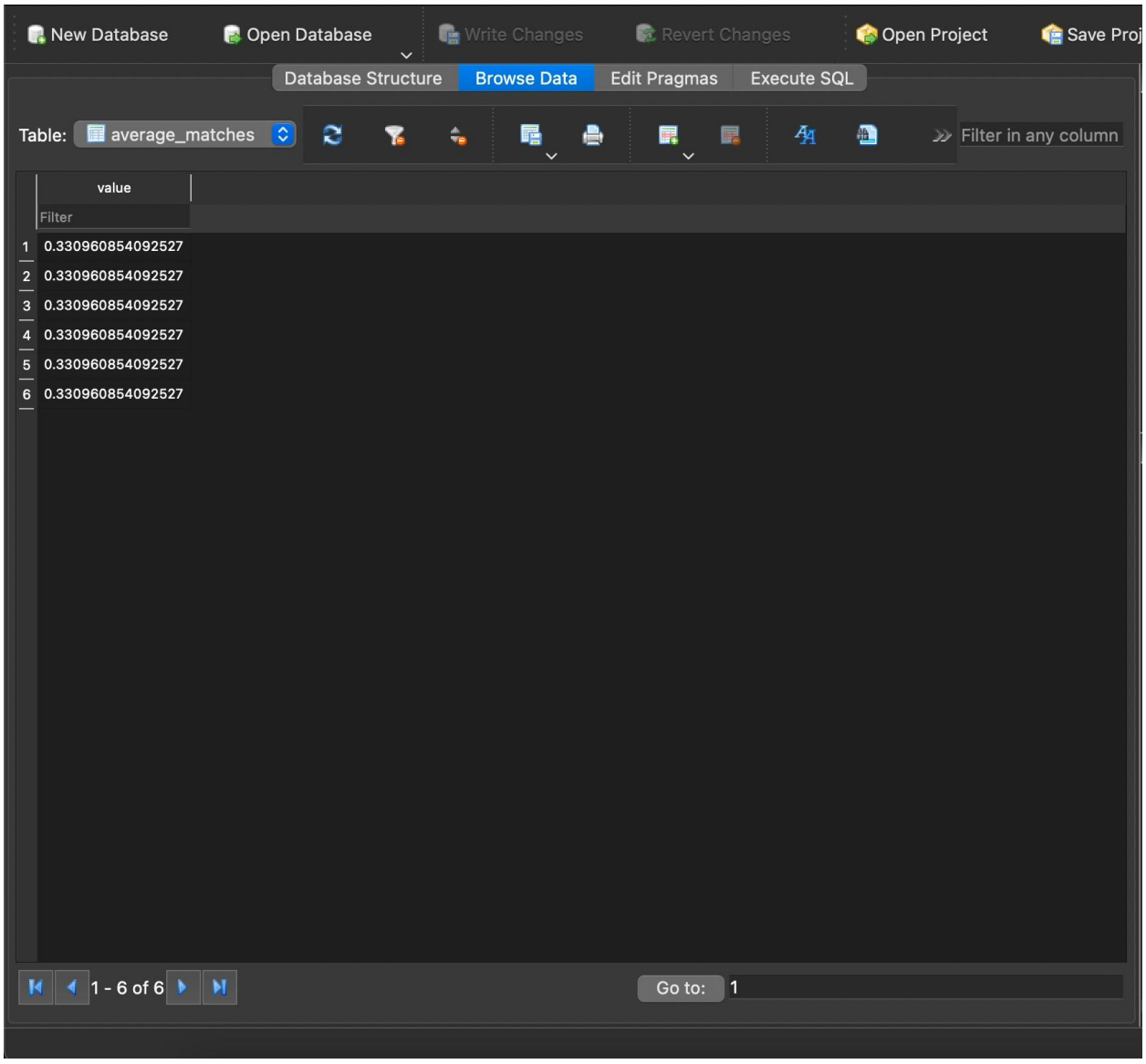
The Office-Quran comparison (episodes):


New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Pro
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: episodes
Filter in any column
season
title
Filter
Filter
1 1 Pilot
2 1 Diversity Day
3 1 Health Care
4 1 The Alliance
5 1 Basketball
6 1 Hot Girl
7 2 The Dundies
8 2 Sexual Harassment
9 2 Office Olympics
10 2 The Fire
11 2 Halloween
12 2 The Fight
13 2 The Client
14 2 Performance Review
15 2 E-Mail Surveillance
16 2 Christmas Party
17 2 Booze Cruise
18 2 The Injury
19 2 The Secret
20 2 The Carpet
21 2 Boys and Girls
22 2 Valentine's Day
23 2 Dwight's Speech
24 2 Take Your Daughter to Work Day
25 2 Michael's Birthday
1 - 25 of 202
Go to:
1

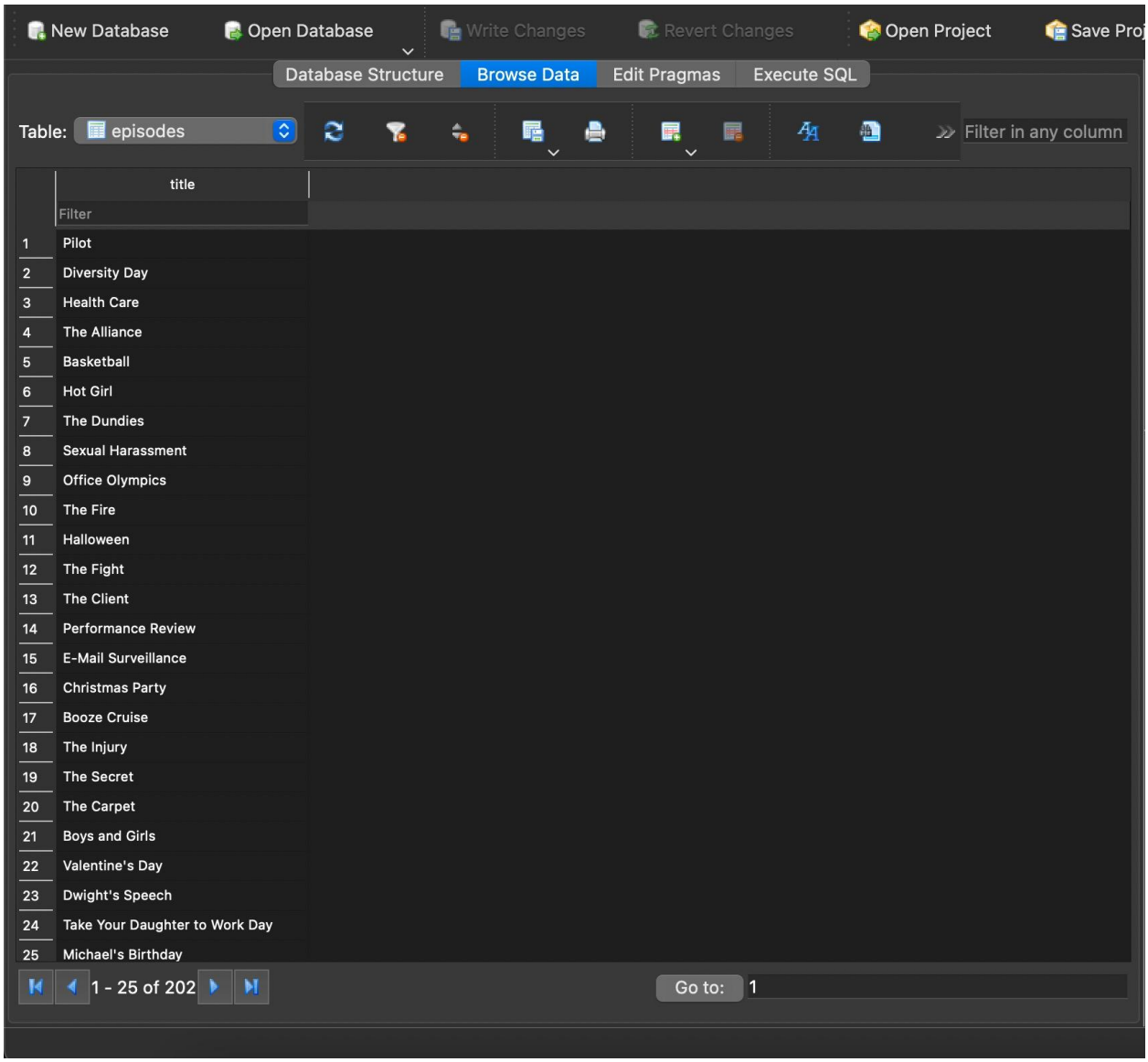
The Office-Quran Comparison (verses):



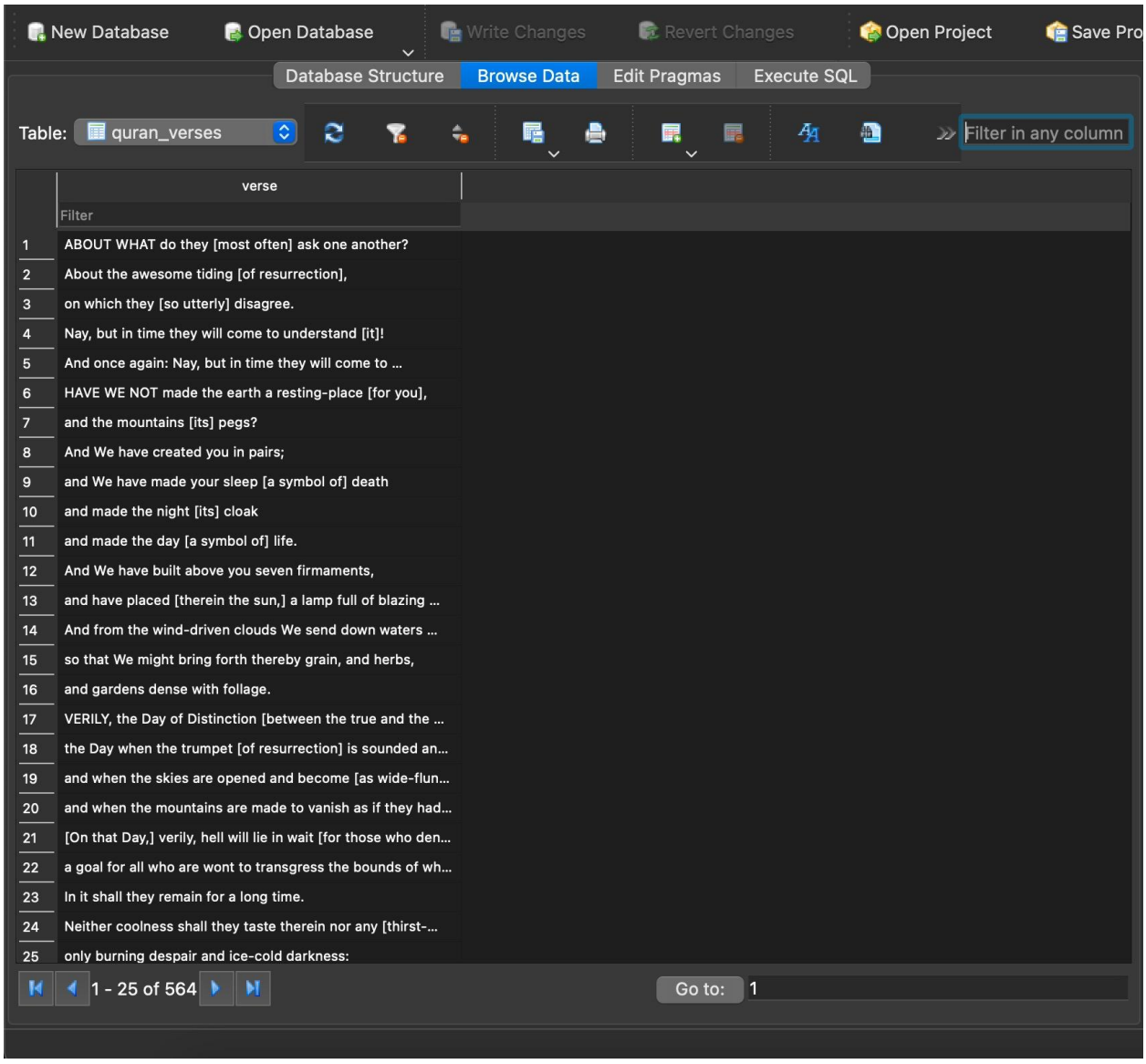
Quran calculation (averages):



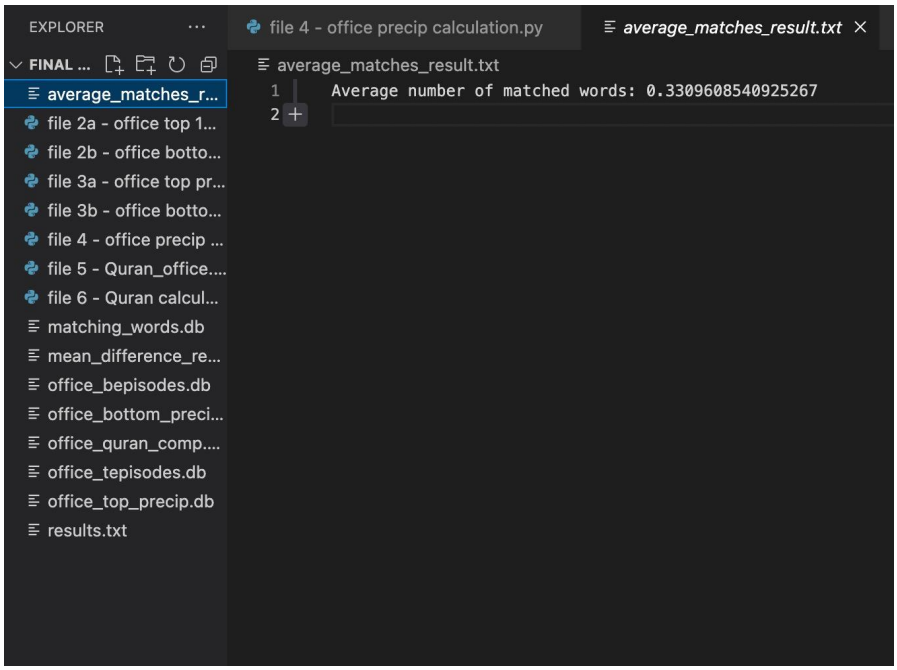
Quran calculation (episodes):



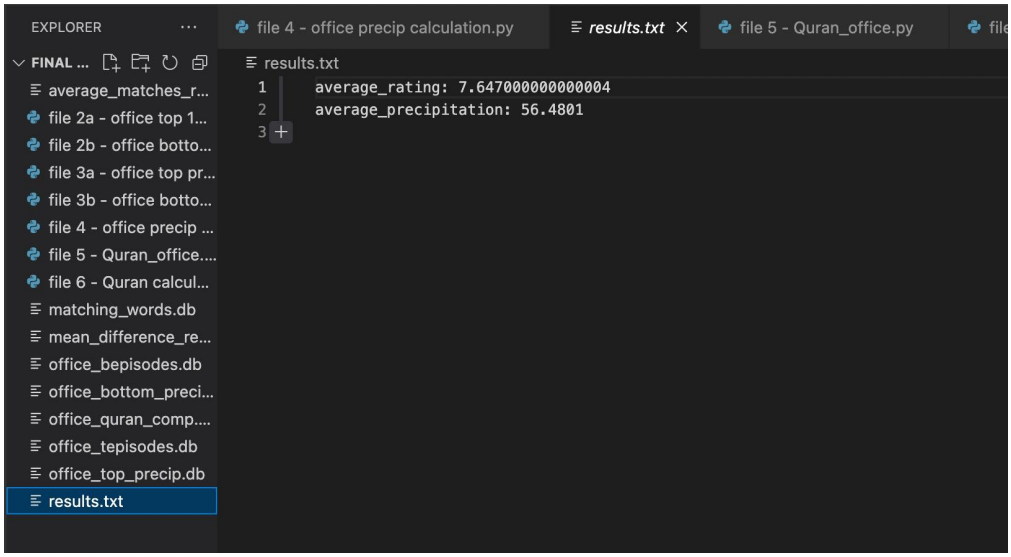
Quran calculation (verses):

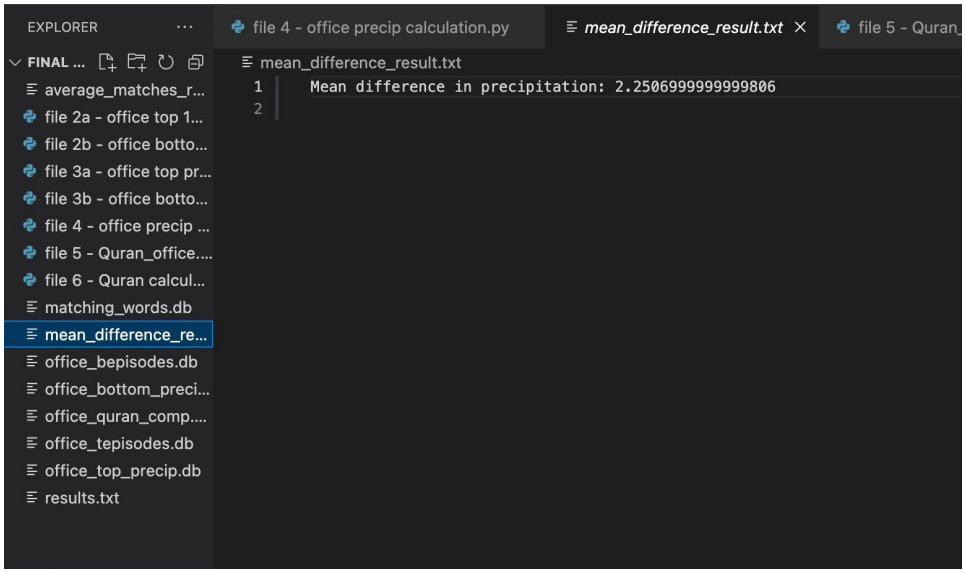


Quran calculation code and output:

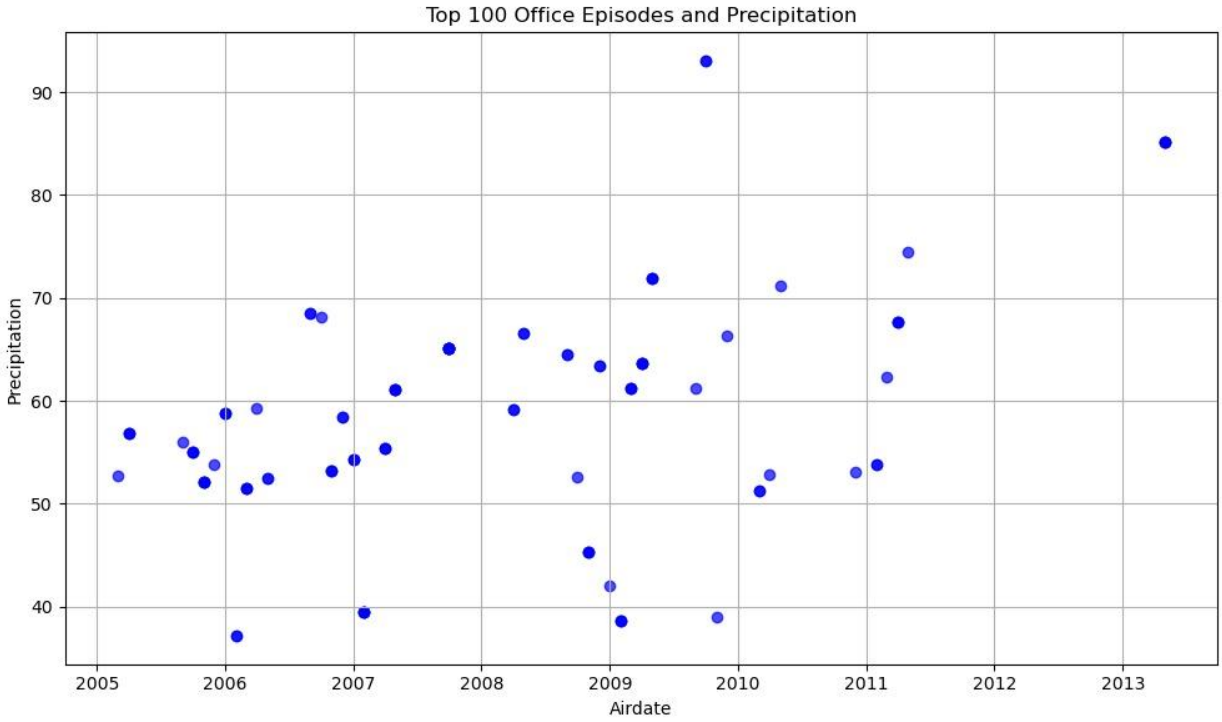


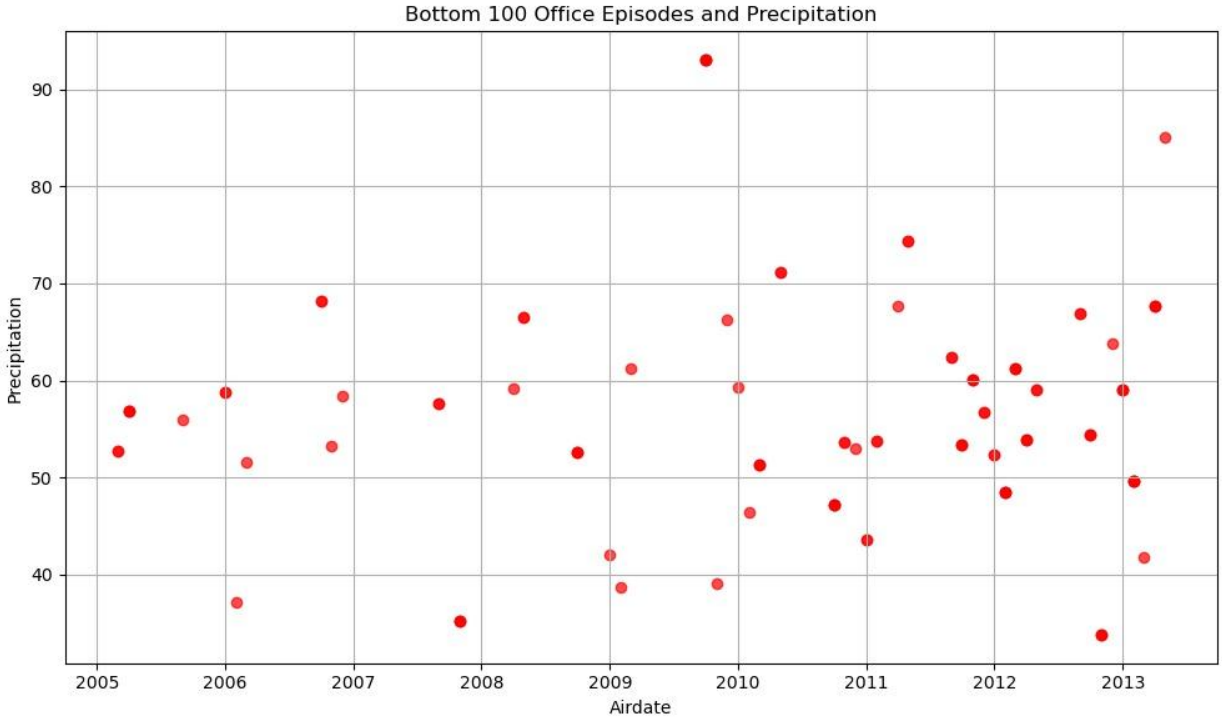
Precipitation Calculations output:

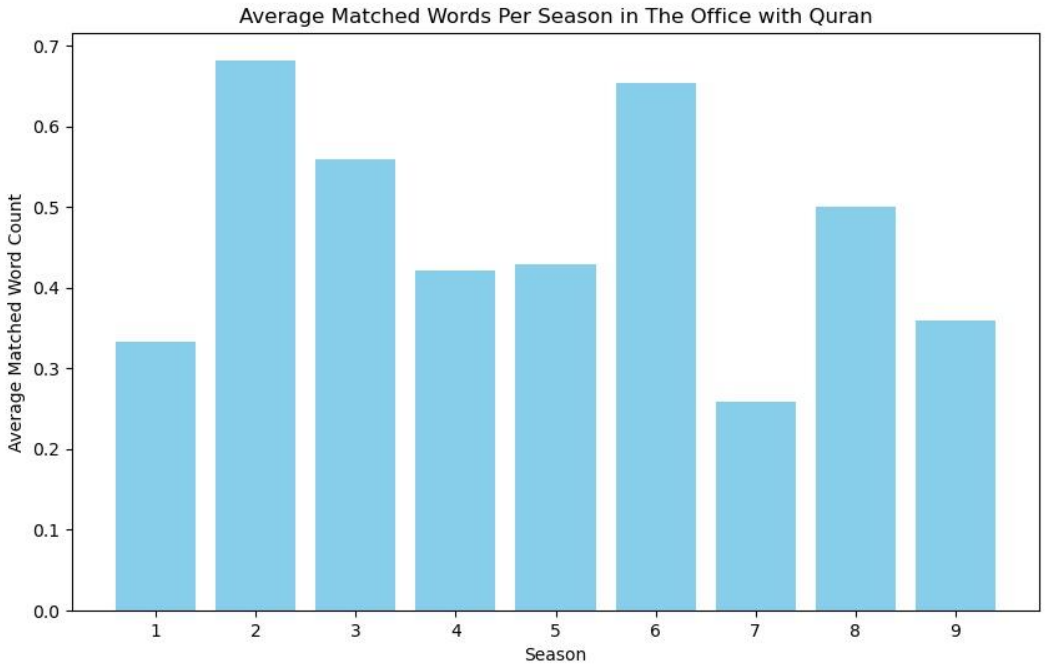




Visualizations:







Instructions:

There is no clear way to run our code, but for it to make sense cohesively, you may run the code in this manner:

1. Begin by running File 2a and File 2b to see the data on the top and bottom 100 episodes of *The Office*.
2. Run File 3a to see the correlation between the top 100 episode dates and precipitation on the month and year that those were released in a graph.
3. Run File 3b to see the correlation between the bottom 100 episode dates and precipitation on the month and year that those were released in a graph.
4. Run File 4 to see our calculations for the average difference in precipitation between the top and bottom episode release dates.
5. Run File 5 to see the correlation between The Office episode titles and words in the Quran in a graph.

6. Run File 6 to find the average matches of words between The Office episode titles and words in the Quran.

Documentation:

Date	Issue Description	Location of Resource	Result
12/02/23	Finding API	https://www.tvmaze.com/api	TV show API with ratings and date.
12/02/23	Trouble with navigating the TVmaze API.	https://youtu.be/jiJJ2V8K1ik?si=FdyS4TfDJVvh6uyG	Helped our group understand and use the TVmaze API.
12/02/23	Trouble with getting our code to work.	https://chat.openai.com/	Helped us edit our code and give us instructions on what to do better.
12/02/23	Trouble with finding free and working APIs.	https://github.com/public-apis/public-apis	Helped us find many APIs to use.
12/03/23	Trouble figuring out if the APIs worked and had the data we needed.	https://www.postman.com/	Helped us comb through APIs to make sure it has the data and information we need.
12/06/23	Trouble with creating visuals using matplotlib	Discussion 13	Helped with understanding the code to create visualizations.
12/11/23	Finding a weather API	https://climateknowledgeportal.worldbank.org/	Helped us find a usable weather API.
12/11/12	Finding a third API to work with	https://alquran.cloud/api#collapseThree	Gave us Quran verses to work with.

Code Documentation:

File 2a - office top 100.py

- Function: insert_episode_data:
    - Creates SQLite table. It sorts top 100 episodes by rating, extracts the month and year from the episode's "airstamp," and inserts the data into the table.
    - Inputs
        - conn: SQLite database connection object
        - cur: SQLite cursor object
        - episode_data: Tuple with episode data containing title, season, number, rating, month, and year.
    - Outputs:
        - None

File 2b - office bottom 100

- Function: insert_episode_data:
    - Creates SQLite table. It sorts the bottom 100 episodes by rating, extracts the month and year from the episode's "airstamp," and inserts the data into the table.
    - Inputs
        - conn: SQLite database connection object
        - cur: SQLite cursor object
        - episode_data: Tuple with episode data containing title, season, number, rating, month, and year.
    - Outputs:
        - None

File 3a - office top precipitation.py

- create_db_table(conn, cur)
    - Creates SQLite table if does not exist and the table contains columns for title, season, number, rating, date, and precipitation, avoiding duplicates through unique title, season, and number
    - Inputs:
        - conn: SQLite connection object
        - cur: SQLite cursor object
    - Outputs:

        - None
- insert_episode_data(conn, cur, episode_data)
    - Inserts episode data into the SQLite table created before, also ensuring uniqueness.
    - Inputs:
        - conn: SQLite connection object
        - cur: SQLite cursor object
        - episode_data: Tuple containing episode information - (title, season, number, rating, date, precipitation)
    - Outputs:
        - None
- fetch_precipitation_data()
    - Obtains USA precipitation data from the World Bank API in dictionary format
    - Inputs:
        - None
    - Outputs:
        - Dictionary containing USA precipitation data
- main()
    - Retrieve episode data from TVMaze API for “The Office” and precipitation data from the World Bank API
    - Stores episode and precipitation data into SQLite database 25 rows at a time until 100 rows are stored
    - Generates scatterplot showcasing the relationship between episode air dates and precipitation levels for the top 100 episodes of “The Office”
    - Inputs:
        - Accesses the TVMaze API and World Bank API
        - Uses an SQLite database named 'office_top_precip.db'
    - Outputs:
        - Scatter plot showing the relationship between top air dates and precipitation values

File 3b - office bottom and precipitation

- create_db_table(conn, cur)
    - Creates SQLite table if does not exist and the table contains columns for title, season, number, rating, date, and precipitation, avoiding duplicates through unique title, season, and number
    - Inputs:
        - conn: SQLite connection object
        - cur: SQLite cursor object
    - Outputs:
        - None
- insert_episode_data(conn, cur, episode_data)
    - Inserts episode data into the SQLite table created before, also ensuring uniqueness.
    - Inputs:
        - conn: SQLite connection object
        - cur: SQLite cursor object
        - episode_data: Tuple containing episode information - (title, season, number, rating, date, precipitation)
    - Outputs:
        - None
- fetch_precipitation_data()
    - Obtains USA precipitation data from the World Bank API in dictionary format
    - Inputs:
        - None
    - Outputs:
        - Dictionary containing USA precipitation data
- main()
    - Retrieve episode data from TVMaze API for “The Office” and precipitation data from the World Bank API
    - Stores episode and precipitation data into SQLite database 25 rows at a time until 100 rows are stored

    - Generates scatterplot showcasing the relationship between episode air dates and precipitation levels for the bottom 100 episodes of “The Office”
    - Inputs:
        - Accesses the TVMaze API and World Bank API
        - Uses an SQLite database named 'office_bottom_precip.db'
    - Outputs:
        - Scatter plot showing the relationship between bottom air dates and precipitation values

File 5 - Quran_office.py

- setup_db(conn)
    - Creates tables if they do not already exist in the database
    - Inputs:
        - conn: SQLite data
    - Output:
        - cur: Cursor object for executing SQL commands
- fetch_quran_text()
    - Retrieves English translations of Quran verses from an API
    - Output:
        - list of tuples containing season number and episode title
- fetch_all_episodes()
    - Retrieves all episodes of “The Office” from the TVMaze API
    - Output:
        - List of English translated verses from the Quran
- insert_data(conn, cur, table, data)
    - Inserts data into the specified table with the connected database
    - Inputs:
        - cur: cursor object for executing SQL commands
        - conn: SQLite database connection
        - table: name of the table to insert data into
        - data: data to be inserted into the table
- get_all_data(cur, table)

    - Retrieves all data from specified table
    - Inputs:
        - cur: cursor object for executing SQL commands
        - Table: name of the table to obtain data from
    - Output:
        - List of tuples containing data from the specified table
- calculate_match_counts_by_season(episodes, quran_verses)
    - Computes the average matched word count per season between episodes and Quran verses
    - Inputs:
        - episode: List of tuples containing episode title and season number
        - quran_verses: list of English translations of Quran verses
    - Outputs:
        - Dictionary with season number as keys and the average matched word count as values
- main()
    - main function orchestrating the data retrieval, storage, analysis, and visualization process for Quran verses and “The Office” episodes. It establishes connections, fetches data, calculates matches, and creates a bar graph showing the average matched words per season

File 4 - office precip calculations.py

- setup_db(conn)
    - Creates tables if they do not exist in the database
    - Input:
        - conn: SQLite database connection
    - Output:
        - cur: cursor object for executing SQL commands
- fetch_quran_text()
    - Retrieves English translations of Quran verses from an API
    - Inputs:

        - None
    - Outputs:
        - List of English-translated Quran verses
- insert_data(conn, cur, table, data)
    - Inserts data into the specified SQLite table within the connected database
    - Inputs:
        - conn: SQLite database connection
        - cur: cursor object for executing SQL commands
        - table: name of the table to put data into
        - data: information to be put into table
    - Outputs:
        - None
- get_all_data(cur, table)
    - Retrieves all data from a specified table
    - Inputs:
        - cur: cursor object for executing SQL commands
        - table: name of the table to get data from
    - Output:
        - List of tuples containing data from the specified table
- calculate_match_counts_by_season(episodes, quran_verses)
    - Computes the average matched word count per season for episodes of "The Office" and Quran verses
    - Inputs:
        - episodes: list of tuples containing season number and episode title
        - quran verses: list of English translated Quran verses
    - Outputs:
        - None
- main()
    - Orchestrating the data retrieval, storage, analysis, and visualization creation for "The Office" episodes and Quran verses. Establishes connections, retrieves data,

calculates matches, and plots a bar graph showing average matched words per season

File 6 - Quran calculations

- setup_db(conn)
    - Creates SQLite tables ‘episodes’ and ‘quran_verses’ if they don’t exist, or use the ones that exist
    - Inputs:
        - conn: SQLite database connection
    - Output:
        - Cursor object for executing SQL commands
- fetch_quran_text()
    - Fetches text data of verses from Quran API
    - Inputs:
        - None
    - Outputs:
        - List of Quran text verses
- fetch_all_episodes()
    - Retrieves the titles of episodes from “The Office
    - Inputs:
        - None
    - Outputs:
        - List of episode titles
- insert_data(conn, cur, table, column, data)
    - Inserts data into the specified table and column of the SQLite database
    - Inputs:
        - conn: SQLite database connection
        - cur: Cursor object to execute SQL commands
        - table: Name of the table
        - column: Name of the column where data will be put into
        - data: List of data to be put into the table

    - Outputs:
        - None
- get_all_data(cur, table, column)
    - Retrieves data from a specific column in the specified table
    - Inputs:
        - cur: Cursor object to execute SQL commands
        - table: Name of the table
        - column: Name of the column
    - Outputs:
        - List of data retrieved from the given column
- calculate_average_matches(episode_titles, quran_verses)
    - Calculates the average number of matched words between episode titles and Quran verses
    - Inputs:
        - episode_titles: List of episode titles
        - quran_verses: List of verses from the Quran
    - Outputs:
        - Average number of matched words
- insert_average_matches(conn, cur, average_matches)
    - Inserts calculated average matches in to the corresponding table in the database (“average_matches”)
    - Inputs:
        - conn: SQLite database connection
        - cur: Cursor object to execute SQL commands
        - average_matches: Average number of matched words
    - Outputs:
        - None
- main()
    - Executes the script; establishes the connection with SQLite database, fetches Quran verses and episode titles, calculates the average number of match words,

inserts the result into the database, writes it to “average_matches_result.txt,” and closes the database connection