Capstone Report

Alexander Devora, Sandra Ovuegbe, Anastasia Jones, Chandershekhar Shori

1. **Program Design:**

   We constructed a Python program in the collaborative program called Jupyter, which utilized data from PUBMED. We first constructed a crawler module using the following packages: biopython, pandas, nbformat, chart-studio, cufflinks, colorama, matplotlib. We also used the following modules in the scraper package: Biopython Entrez, Csv module, Json module, and http clients. The purpose of the Scraper module was to retrieve the paper title, author list, publication time, and abstract from PUBMED for a given keyword within a pre-specified time window, such that the retrieved data should be saved in the CSV format.

   Next, we have the database module which makes use of the CSV file from the crawler module to import the CSV file to SQLite to build a database automatically. The database module then implements SQL code to query the publications by author's name. The database module utilizes the following modules: CSV module and sqlite3 module.

   Lastly, we used the visualization module to read the CSV file, show the number of publications in each month, generate and visualize the summary statistics for the publication numbers per month, and visualize the trend of the publication numbers over time. This module made use of the following packages: chart-studio and cufflinks. It also uses the following Python modules: pandas, datetime, numpy, statistics, plotly.graph_objs, plotly.offline, colorama, matplotlib.pyplot, and matplotlib.dates.

Table 1: Tool Workflow

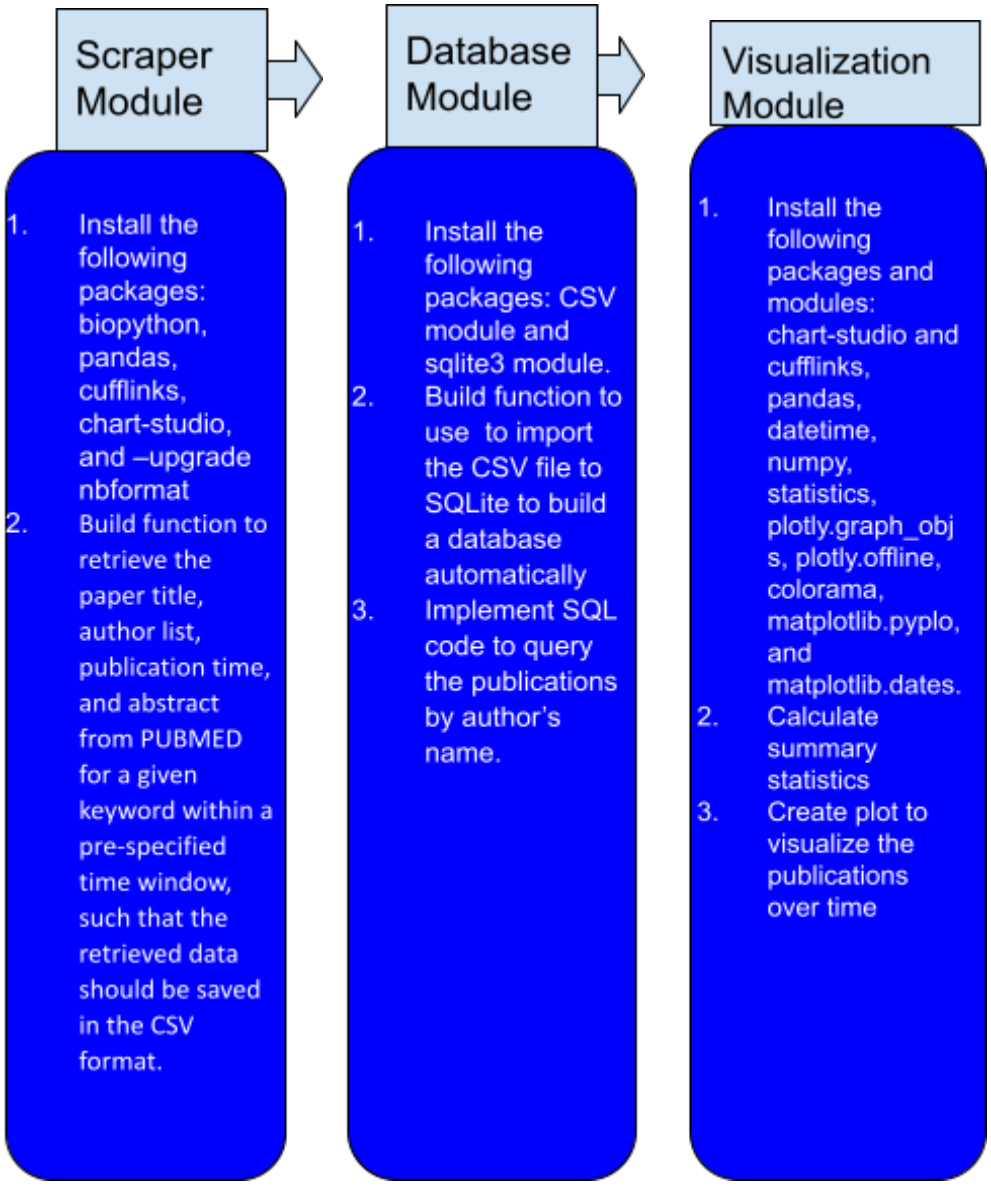
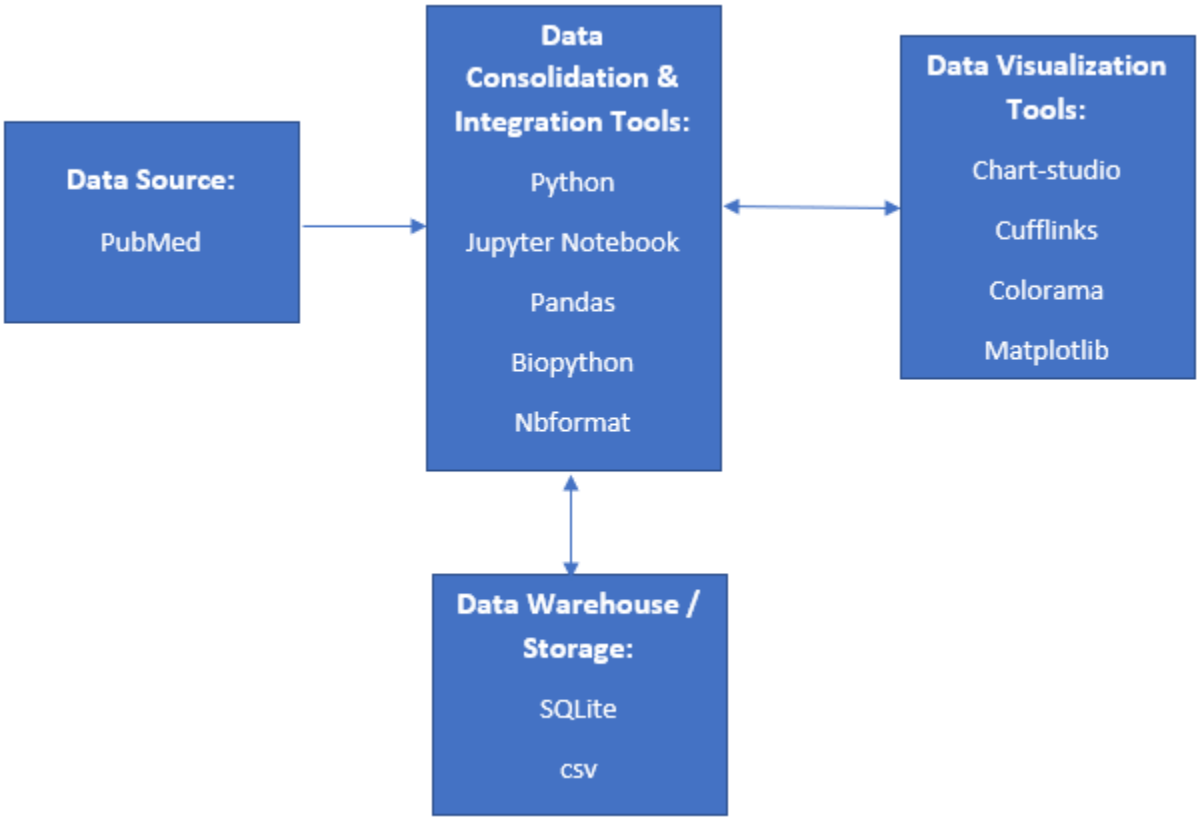


Table 2: Tool Architecture



2. **<u>Implementation details</u>:**

Scraper Module

Our scraper module uses the Python modules csv, Bio, json, and http.client. It first installs all Python packages that will be used in our program, of which biopython and nbformat are required for running the scraper module. This module collects the title, authors, abstract, and date of electronic publication for papers on PubMed with the keyword "HIV" electronically published between 1/1/2020 and 8/30/2020. Once scraped, the data is written to a .csv file.

- Cell 1.1 pip installs all packages to be used in developing the proposed program i.e biopython, pandas, nbformat, chart-studio, cufflinks, colorama, and matplotlib.
- Cell 1.2 installs entrez and medline modules from the biopython package in 1.1. Entrez will enable access to pubmed articles while Medline allows access to reference pages of the articles. Additionally, cell 1.2 installs a csv module for manipulating csv files and installs a json module to read data from web servers.
- Cell 1.3 assigns the start date, end date and title filters for the articles to be extracted
- Cell 1.4 runs a try, except, else loop to destabilize a certificate error code that may happen when running cell 1.5. It is commented out, but if necessary the first # symbol from each line can be removed and this cell run to address the aforementioned error.
- Cell 1.5 reports the users email to the National Center for Biotehnology Information which is the official body that grants access to the library of research articles. Cell 1.5 also uses entrez.search

to specify the database of interest as pubmed, the keyword, and the number of articles to return. Thereafter, entrez.read reads the list of extracted articles.

- Cell 1.6 sums the number of rows in the list
- Cell 1.7 opens a new csv file and writes the list into it. Next, the 'writerow()' function is used to write column headers into the first row
- Cell 1.8 returns the references of the articles extracted in 1.5. 'Entrez.fetch()' function is used to retrieve the full references list from medline. 'Medline.parse()' is used to read paper title, author, electronic publication date and abstract from references. A 'for loop' is used to specify that only title,author, pubdate, and abstract should be extracted from the reference list of those articles. Next, a 'csv.writerow()' function writes the output into the csv file created in cell 1.7

Database Module

Our database module uses the Python modules sqlite3 and csv. It creates a database in sqlite3 filled with a table of the PubMed data extracted from our scraper module. Finally, it prompts the user for an author's name and returns those papers scraped from PubMed that include the input author.

- Cell 2.2 first imports the Python modules csv and sqlite3. It then uses csv.reader to read the output csv from the scraper module line by line. The values being read are the paper's title, the author list, the date of electronic publication, and the abstract. Finally, it uses sqlite3 to create a "pubmed.db" database in sqlite.
- Cell 2.2 prompts the user to input an author's name. It then executes a query of publications in pubmed.db by author for the user's input. The command prints the author's name and returns the number of associated articles.

Visualization Module

Our visualization module uses the following Python modules: pandas, datetime, numpy, statistics, plotly.graph_objs, plotly.offline, colorama, matplotlib.pyplot, and matplotlib.dates. It also requires that the Python packages pandas, chart-studio, colorama, matplotlib, and cufflinks are installed (all pip installations occur in the first cell of our Scraper Module). Some cells in our visualization module have subsections separated by commented dashed lines, which are addressed by sub-bullet points.

- Cell 3.1 imports pandas, then reads in the .csv file containing the output from the PubMed scrape performed in Q1 and saves it as a pandas dataframe called *df*. *df* is returned.
- Cell 3.2 imports datetime, date, and timedelta from the module datetime. It creates the value *NoDate*, which represents the number of studies which do not have an electronic publication date listed. It then returns the count of each column in *df* before and after dropping the studies that are missing an electronic publication date. Using the datetime module, this cell converts the *publication_time* column of *df* to date format. It then creates 3 new columns, *publication_month*, *month*, and *publication_year*, which are different time formats and components of *publication_time* that will each be used later.
- Cell 3.3 filters the studies in *df* for *publication_year* and *month* values which are in the specified date range. It returns a count of each column before and after this filtration takes place.
- Cell 3.4 shows the number of publications in each month.
    - First, *df* is sorted by *publication_time*
    - A list of unique publication months is created from *publication_month*, titled *monthlist*.

    - A while-loop creates a new list, *countlist*, which consists of the number of papers in *df* corresponding to each month of publication in *monthlist*. *monthlist* and *countlist* are then bound together into a new pandas dataframe called *monthcountdf*.
    - Two new variables, *sdate* and *edate,* are created. These correspond to the first and last months in *monthlist*, respectively.
    - A copy of *monthcountdf*, *monthcountdf2*, is created. Any missing values in *monthcountdf2* are filled with zeroes. The contents of the *Count* column in *monthcountdf2* are converted to integers.
    - A similar pandas dataframe, *monthcountdf3* is created. This dataframe has the dates in range (*sdate*, *edate*) as its indices and a single column for *Count*.
- Cell 3.5 generates and visualizes the summary statistics for publication counts per month.
    - First, the Python modules numpy, statistics, plotly.graph_objs, plotly.offline, and colorama are imported.
    - Summary statistics are generated from monthcountdf2. Mean, standard deviation, and median are calculated using the statistics module. A list of quartile values are calculated using the numpy module. Q1 and Q3 values are then extracted from the quartile list. Minimum and maximum values are derived and combined into a list representing the range using base Python commands.
    - Each of the aforementioned summary statistics are then printed in fabulous bright colors using the colorama module. Different colors group related summary statistics.
    - The modules plotly.offline and plotly.graph_objs are then used to represent the summary statistics in the form of a boxplot, which specifies each statistic when scrolled over. The boxplot is customized to display mean and standard deviation as a dotted line and diamond, respectively. Note that if the first quartile and minimum or the third quartile and maximum are the same value, that value will only be labeled in the boxplot as its respective quartile value. You can verify whether this quartile value equals the minimum or maximum by checking the printed statistics directly above the boxplot.
- Cell 3.6 visualizes the trend of electronic publications over time.
    - The Python modules matplotlib.pyplot and matplotlib.dates are imported.
    - The figure size and layout are specified using the matplotlib.pyplot module. A plot is created with monthcountdf3 index (recall that this is month of electronic publication) as the x-axis and count of papers on the y-axis.
    - Next, the plot is customized using the matplotlib.pyplot module. Four labels are created with various combinations of specified fontsize, fontweight, and color: suptitle, title, x-axis label, and y-axis label. The title is further customized with the startdate and enddate specified in Q1, while the x-axis label has the word “Electronic” underlined. Colors are specified for the x- and y-axis ticks and for the plot face.
    - Finally, the dates on the x-axis are formatted such that they do not overlap. Using the matplotlib.dates modules, their representation on the plot is set to the more intuitive Mon-Year format. The plot is returned.

3. **<u>Results</u>**:

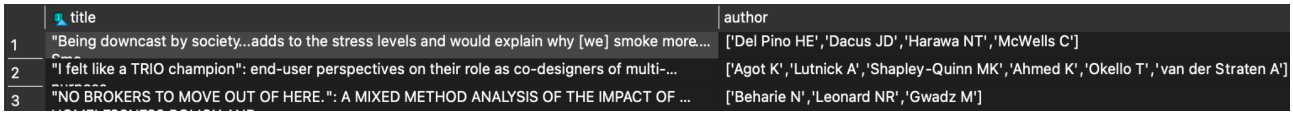
Part 1:

The first few rows of pubmed.results.csv looks like:

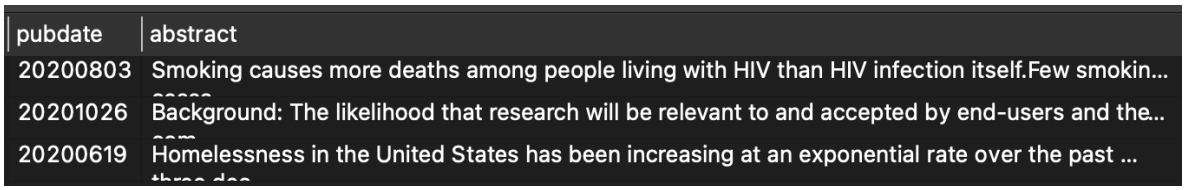
	title	author	publication_time	abstract
0	A multi-parameter diagnostic clinical decision...	['van Hoving DJ', 'Meintjes G', 'Maartens G', ...	20220512	Background: Early diagnosis is essential to re...
1	The combination of low level laser therapy and...	['Lugongolo MY', 'Manoto SL', 'Ombinda-Lemboum...	20200604	BACKGROUND: Human immunodeficiency virus (HIV)...
2	Barriers to HIV Care by Viral Suppression Stat...	NaN	20200817	NaN
3	Evaluating Translation of HIV-Related Legal Pr...	['Adia AC', 'Lee CJ', 'Restar AJ', 'Obiakor BC...	20200325	Legal protections for people living with HIV (...
4	Host Directed Therapies for Tuberculous Mening...	['Davis AG', 'Donovan J', 'Bremer M', 'Van Too...	20210701	A dysregulated host immune response significan...
...	...	...	...	...
9989	HIV residual risk in Canada under a three-mont...	["O'Brien SF", 'Gregoire Y', 'Pillonel J', 'St...	20191127	BACKGROUND AND OBJECTIVES: In Canada, the defe...

Part 2:

- The following images display a few rows from the SQLite database that was generated.



	title	author
1	"Being downcast by society...adds to the stress levels and would explain why [we] smoke more....	['Del Pino HE','Dacus JD','Harawa NT','McWells C']
2	"I felt like a TRIO champion": end-user perspectives on their role as co-designers of multi-...	['Agot K','Lutnick A','Shapley-Quinn MK','Ahmed K','Okello T','van der Straten A']
3	"NO BROKERS TO MOVE OUT OF HERE.": A MIXED METHOD ANALYSIS OF THE IMPACT OF ...	['Beharie N','Leonard NR','Gwadz M']



pubdate	abstract
20200803	Smoking causes more deaths among people living with HIV than HIV infection itself.Few smokin...
20201026	Background: The likelihood that research will be relevant to and accepted by end-users and the...
20200619	Homelessness in the United States has been increasing at an exponential rate over the past ...

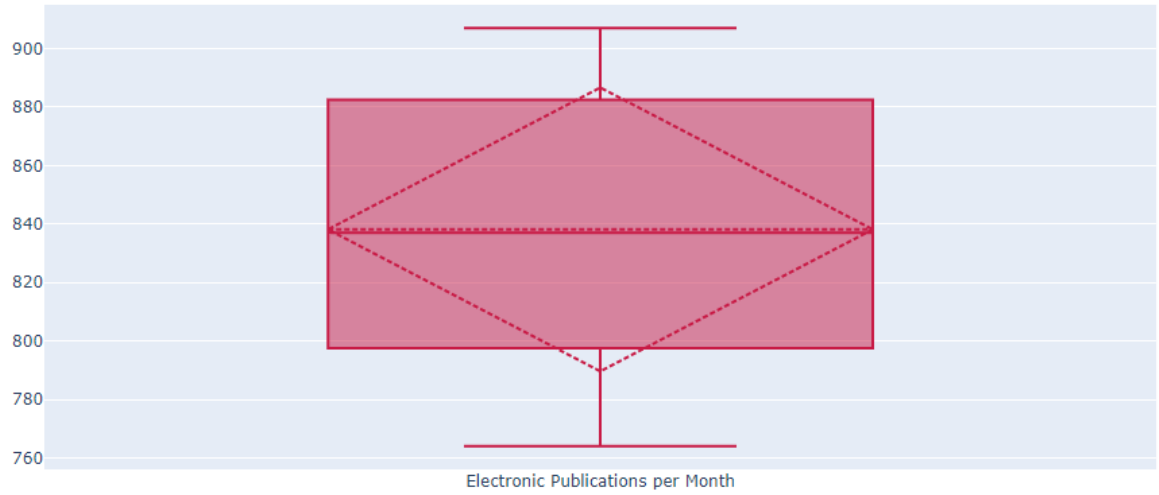
- We will now use the SQL code to query an author's publications by their name. In this demo, we input the author "Chan PA" and their number of publications is returned along with the article titles, author lists, publication time, and abstracts.

```
What author would you like to search for?: Chan PA
Chan PA has 10 articles
0 ('Awareness and use of HIV pre-exposure prophylaxis among people who engage in sex work presenting to a sexually
1 ('HIV Pre-Exposure Prophylaxis Awareness and Use Among Men Who Have Sex with Men Only and Men Who Have Sex with
2 ('eTest: a limited-interaction, longitudinal randomized controlled trial of a mobile health platform that enable
3 ('Public Health Approaches Toward Eliminating Hepatitis C Virus in Rhode Island.', "['Murphy M', 'Howe K', 'Maral
4 ('Evaluating statewide HIV preexposure prophylaxis implementation using All-Payer Claims Data.', "['Raifman J',
5 ('Potential Impact of Targeted HIV Pre-Exposure Prophylaxis Uptake Among Male Sex Workers.', "['Goedel WC', 'Mim
6 ('Undetectable Equals Untransmittable: A Game Changer for HIV Prevention.', "['Patel RR', 'Curoe KA', 'Chan PA']
7 ('Interest and Knowledge of HIV Pre-Exposure Prophylaxis in a Unified Jail and Prison Setting.', "['Brinkley-Rub
8 ('A cross-sectional evaluation of HIV testing practices among women in the rural Dominican Republic.', "['Montgo
9 ('Leveraging Medicaid to Enhance Preexposure Prophylaxis Implementation Efforts and Ending the HIV Epidemic.', "
```

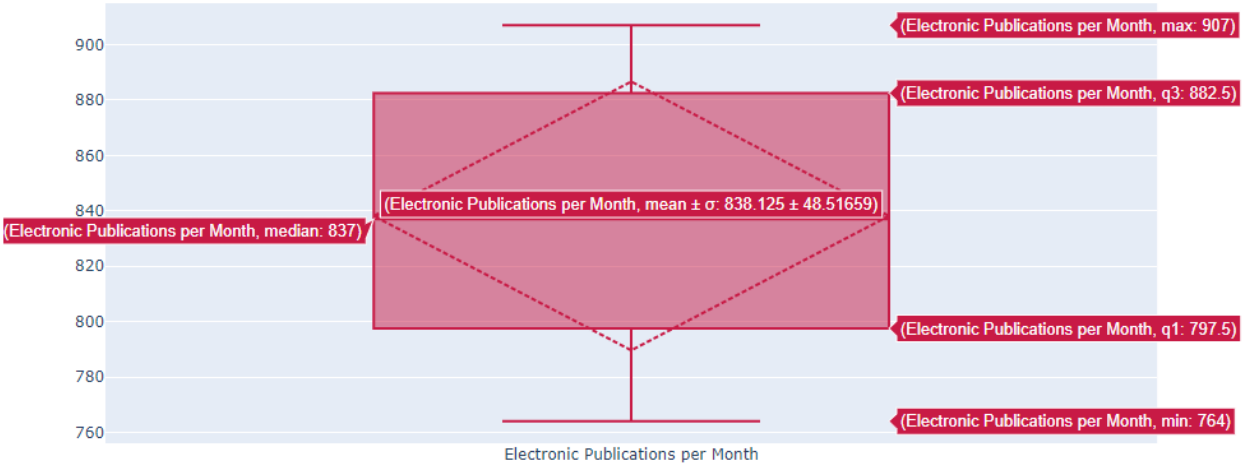
Part 3:

- Running cell 3.5 generates and visualizes summary statistics for the publication numbers per month.

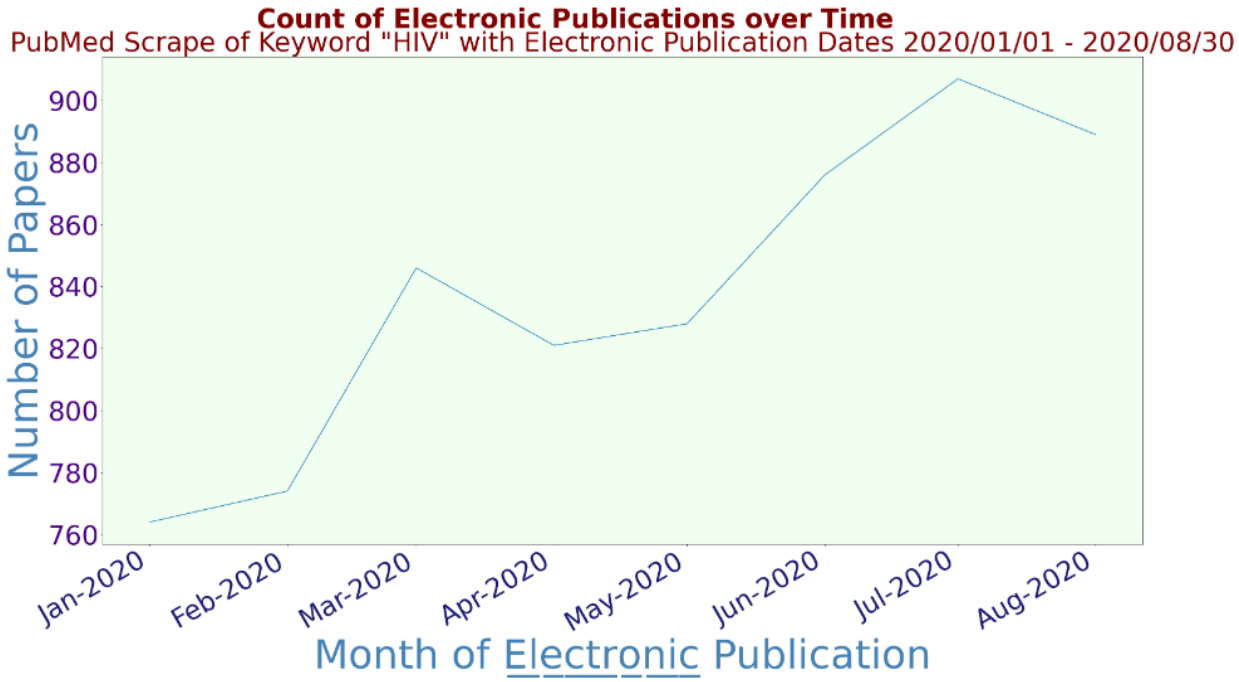
```
Summary Statistics for Count of Electronic Publications per Month
Mean: 838.125
Standard Deviation: 51.86641495225981
Median: 837.0
Quartiles: [809.25 837.   879.25 907.  ]
1st Quartile: 809.25
3rd Quartile: 879.25
Minimum: 764
Maximum: 907
Range: [764, 907]
```



- View when scrolling over boxplot:



- Running Cell 3.6 visualizes the trend of the publication numbers over time by months.



4. **<u>User Manual/Guide</u>:**
    - Open Jupyter Notebook and upload Capstone_Project.ipynb. Open Capstone_Project.ipynb in Jupyter Notebook.
    - If the most recent versions of any of the Python packages listed in Section 3 are not already installed on your device, remove the hashtags in the first cell preceding the pip install commands for the Python packages you are missing. Run the first cell.
    - Run cell 1.2 to import special features from the packages installed the first cell, as well as other inbuilt python packages. Do this step to be able to proceed with web scraping.
    - Run cell 1.3 This step defines the keywords of interests.
    - Run cell 1.5 This cell searches the entrez library for articles with the keywords and puts them in a list. Remember to input an email address to get access from NCBI.
    - Remove the first # symbol from each line of cell 1.4 and then run it if faced with a certificate verification error in 1.5. If you had to do this, rerun cell 1.5.
    - Run cell 1.6 to get the total sum of articles from the list in 1.5.
    - Run cell 1.7 to open a new csv file and write the column headers.
    - Run cell 1.8 This step fetches the reference list of the articles in 1.5 and puts them in the csv file you ceated. Run the 'for loop' to pick what you want drawn from the references. Import the http client package at this step. The 'http client' package makes it posssible to run queries and extract information from webpages like pubmed.
    - Run cells 2.1 and 2.2. When prompted for an author's name, write an author's name and press enter. Format should be last name, followed by first name initial and middle name initial. For example "Chan PA". If the author does not have a middle name, just put the last name and first name initial. For example "Pike R".

- In Q3, some cells have subsections separated by commented dashed lines. This user manual addresses each subsection individually.
- Run cell 3.1. This cell reads in the .csv file containing the output from the PubMed scrape performed in Q1 and saves it as a dataframe called df. df is returned.
- Run cell 3.2. This cell shows the number of electronic publications in each month.
- Run cell 3.3. This generates and visualizes the summary statistics for publication counts per month.
- Run cell 3.4. This visualizes the trend of electronic publications over time.

References

Anaconda Software Distribution. (2020). *Anaconda Documentation*. Anaconda Inc. Retrieved from https://docs.anaconda.com/

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., … others. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423.

Hipp, R. D. (2020). *SQLite*. Retrieved from https://www.sqlite.org/index.html

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95.

Kluyver, T., Ragan-Kelley, B., Fernando P'erez, Granger, B., Bussonnier, M., Frederic, J., … Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90).

McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.