In [1]:

```
import yaml
import json
import os

import sqlalchemy as sql
import pandas as pd
import plotly.express as px
import pm_query as pq

from Bio import Entrez
pd.options.mode.chained_assignment = None # Stop set copy on slice warning
```

Part 1: Scraper

These are the initial parameters of the scraper module: keys, email, and search. For the purpose of this demo, these parameters have already been assigned. The user should reassign these parameters for individual-use. Descriptions for assigned demo parameters are provided below:

keys: function pq.secret_manager reads a yaml file containing the passwords and API keys necessary for running this module without hardcoding them into the Python script. An empty YAML file has been included for use. Add your own API key to increase polling rate.

email: this has been assigned as the email address of the team member who created the module.

search: this has been assigned as the search term "HIV".

In [8]:

```
keys = pq.secret_manager("apikeys_project.yaml")

email = "rachit.sabharwal@uth.tmc.edu"
search = "HIV"
```

```
Read Successful
```

In the below cell, we gather the data for the final data frame. The get_pmid function queries the eSearch endpoint of the Entrez API to retrieve the corresponding pmids and join them to the input dataframe. Using the pmids retrieved in the get_pmids function, the get_data function queries the eFetch endpoint to retrieve the details for the corresponding citation as a list of dictionaries. The data gathered is then converted from a Python dictionary into a JSON-encoded object and saved as hiv_records.json

For the purpose of time, do not run the cell. We have provided our output JSON file needed to continue the demo past this point.

In [ ]:

```
hiv_pmids = pq.get_pmid(contact=email, key=keys["apikeys"]["ncbikey"]["key"], term=search, mindate="202
0/01/01", maxdate="2020/09/01")

hiv_records = pq.get_data(pmid_list=hiv_pmids, contact=email, key=keys["apikeys"]["ncbikey"]["key"])

with open('hiv_records.json', 'w') as outfile:
    json.dump(hiv_records, outfile)
```

In this section, the retrieved data is cleaned by executing the clean_data and keep_cleaning functions.

The keep_cleaning function performs additional cleaning on the data by: 1) resetting the index of the dataframe, 2) converting the pmid variable to an integer data type, 3) formatting the dates into the %Y-%m-%d' format, and 4) joining the columns for title and abstract on index.

Finally, the information from the dataframe is converted into CSV format.

In [2]:

```
with open('hiv_records.json', 'r') as outfile:
    hiv_records = json.load(outfile)

hiv_clean = pq.clean_data(hiv_records)
hiv_clean = pq.keep_cleaning(hiv_clean)

pq.file_downloader("hiv_csv_clean.csv", hiv_clean)
```

```
Your CSV is already up to date
```

Part 2: Database

Using the csv_bnb function, the CSV file created by the data crawler is read via the pandas read_csv function. This data is then reformatted, converting pmids to int type, dates to datetime type, and unlisting author names to prevent nested lists in the dataframe. The function ultimately returns a cleaned dataframe.

The head function is then used to display the first 5 results from the query.

In [3]:

```
hiv_csv = pq.csv_bnb("hiv_csv_clean.csv")
hiv_csv.head()
```

Out[3]:

	pmid	title	abstract	dates	author(s)
0	32866934	The prevalence and risk factors for systemic h...	Diabetes and hypertension are common chronic d...	2020-08-18	Almobarak Ahmed Omer, Badi Safaa, Siddiq Samar...
1	32866611	Expression, purification and crystallization o...	Cdc-like kinase 1 (CLK1) is a dual-specificity...	2020-08-29	Dekel Noa, Eisenberg-Domovich Yael, Karlas Ale...
2	32866436	COVID-19 pneumonia in an HIV-positive woman on...	COVID-19 pandemic has been a problem worldwide...	2020-08-26	Cipolat Murillo Machado, Sprinz Eduardo
3	32866396	Acute supplementation with beetroot juice impr...	Human immunodeficiency virus (HIV) is associat...	2020-08-31	Nogueira Soares Rogerio, Machado-Santos Ana Pa...
4	32866318	Model Informed Development of VRC01 in Newborn...	VRC01 is a first-in-class, potent, broadly neu...	2020-08-31	Li Jerry, Nikanjam Mina, Cunningham Coleen K, ...

The sqlite_out function is used to take the hiv_csv file and two arguments, a database name (assigned as "HIV_Records") and a table name (assigned as "PubMed_Query"), with SQLite established as the database dialect. This function ultimately involves two checks, first is a database check and second is a table check.

At the first check, the function will check for an existing database with the name "HIV_Records". If a database with the name exists, then the function will proceed to check for the specified table name "PubMed_Query". If the table exists within "HIV_Records", then the function will replace it. If the table does not already exist, then the function will create a new table with the specified name.

If "HIV_Records" does not already exist as a database, however, then the function will create a new database with the specified name and then create "PubMed_Query" as a table within "HIV_Records".

In [4]:

```
pq.sqlite_out(hiv_csv, "HIV_Records", "PubMed_Query")
```

Function sql_author_query is then used for an author query, restricting results to those with a similar author name within a specified database and table by using the pandas read_sql function. This function does not create a new database; it queries the already created database.

As seen in the cell below, the desired name for the query has been set as "Mary", the specified database is "HIV_Records", and the specified table is "PubMed_Query".

The head function is again used to display the first 5 results from this query.

In [5]:

```
sql_df = pq.sql_author_query("Mary", "HIV_Records", "PubMed_Query")
sql_df.head()
```

Out[5]:

	pmid	title	abstract	dates	author(s)
0	32866256	Nursing Considerations for Patients With HIV i...	Infection with HIV is a chronic condition that...	None	Graham Lucy, Makic Mary Beth Flynn
1	32864388	COVID-19 in Hospitalized Adults With HIV.	The spread of SARS-CoV-2 and the COVID-19 pand...	2020-08-01	Stoeckle Kate, Johnston Carrie D, Jannat-Khah ...
2	32860699	Risk factors for COVID-19 death in a populatio...	Risk factors for COVID-19 death in sub-Saharan...	2020-08-29	Boulle Andrew, Davies Mary-Ann, Hussey Hannah,...
3	32859191	Understanding long-term HIV survivorship among...	Persons living with HIV (PLWH) are living long...	2020-08-28	Freeman Robert, Gwadz Marya, Wilton Leo, Colli...
4	32852363	Brief Report: Increased Cotinine Concentration...	There is a strong link between cigarette smoki...	None	Diaz Philip T, Ferketich Amy, Wewers Mary E, B...

Part 3: Visualization

To either display number of publications in each month as a bar graph, visualize the trend of the publications over time as a line graph, or view both simultaneously as the line graph overlays the bar graph, call on the draw_graph function. The created graph is interactive.

The default graph drawn is a line graph, which we have shown below. Users can input the optional string parameter "graph_type" to specify for the desired graph type of "line" (the default), "bar" or "both".

EX: pq.draw_graph(df,"both")

In [6]:

```
pq.draw_graph(hiv_csv)
```

Call on the summary_stats function to display the summary statistics by month. Change the string value to output summary statistics for different months. The current input month value is "january". This string is case-insensitive.

In [7]:

```
summary_stats = pq.summary_stats(hiv_csv, "january")
summary_stats
```

Out[7]:

	January (Publications per Month)
mean	27.580645
std	13.065921
min	3.000000
25%	18.000000
50%	32.000000
75%	37.500000
max	46.000000