Setup instructions

```
conda create -n test python=3.8
conda activate test
pip install -r requirements.txt
brew install wget
```

Command

```
python pipeline.py --help

usage: python pipeline.py [-h] [--config CONFIG] [--output_path OUTPUT_PATH]

Collects clinical trial event data. Assigns stock tickers and fsym_ids for a company and

optional arguments:
  -h, --help            show this help message and exit
  --config CONFIG       configuration file for the data pipeline. configuration file is u
                        pipeline frequency, aws s3 credentials
  --output_path OUTPUT_PATH
                        where to save output events csv file on disk

Thank you!
```

Sample output

`lead_company_name` : name of lead company / manufacturer
`lead_company_ticker` : stock ticker of lead company
`lead_company_fsym_id` : fsym id of lead company
`partner_company_names` : name of partner companies / manufacturers
`partner_company_tickers` : stock tickers of partner companies / manufacturers
`partner_company_fsym_ids` : fsym ids of partner companies
`event_type` : type of event
`event_title` : title of event

	Unnamed: 0	expected_date_range_begin	drug_brand_name	lead_company_name
0	0	2013-09-04 00:00:00	Belbuca	BioDelivery Sciences International\xa0Collegiu Pharmaceutical (COLL)
1	1	2013-09-10 00:00:00	Cadazolid	Actelion\xa0Johnson & Johnson (JNJ)
2	2	2013-10-25 00:00:00	Zinbryta	Abbott Labs (ABT)\xa0AbbVie (ABBV
3	3	2014-02-21 00:00:00	Xeljanz	Pfizer (PFE)\xa0Pfizer Lt (India) (500680.IN)
4	4	2014-03-04 00:00:00	Tavalisse	Rigel Pharmaceuticals (RIGL)

Part 1

To run the data pipeline use the following command,

```
python pipeline.py --config config/pipeline_config_local.yaml --output_path events.csv
```

Configuration File

Example pipeline configuration and description of each field is given below.

```
# define properties for sa_events
sa_events:
  # path for sa_events file (supports url or file pointing to a csv, tsv, xlsx)
  source: 'https://raw.githubusercontent.com/harsha070/data-pipeline/048d989b592c374d4b88
bmt_events:
  # path for bmt_events file (supports url or file pointing to a csv, tsv, xlsx)
  source: 'https://raw.githubusercontent.com/harsha070/data-pipeline/048d989b592c374d4b88
# path for fsym_to_ticker mapping file (supports url or file pointing to a csv, tsv, xlsx
fsym: 'https://raw.githubusercontent.com/harsha070/data-pipeline/048d989b592c374d4b88bb87
# flag to upload output events to s3. if set to True, pass aws config (s3_access_key and
upload_to_s3: False
# define your AWS credentials
aws:
  s3_access_key: admin
  s3_secret_key: password
# cron job frequency (in minutes). Set to run the data pipeline runs every 1 minute. If y
job_interval: None # minutes
```

Part 1 Approach

Processing sa_events file

1. Load sa_events file (`sa_events.xlsx`), and map column names to be consistent with the `bmt_events` file. Following mapping is used.

```
COLUMN_NAME_MAP = {
    'Date': 'expected_date_range_begin',
    'Drug Name': 'drug_brand_name',
    'Manufacturer': 'lead_company_name',
    'Partners': 'partner_company_names',
    'Event Type': 'event_type',
    'Details': 'event_title',
}
```

2. Add a column `lead_company_ticker` with the lead company stock ticker. To identify the stock ticker, I used the `Manafacturer` / `lead_company_name` column. As most stock tickers are marked within brackets, using a regex pattern, I identified the possible stock tickers from the `Manafacturer` / `lead_company_name` column. For each matched pattern, I checked if the pattern is present in the `fsym_to_ticker.csv` mapping to confirm if it is indeed a stock ticker. If found, value is assigned. If not, `None` .

3. Add a column `partner_company_tickers` with the partner company stock tickers. To identify the partner company stock tickers, I used the `Partners` / `partner_company_names` column. I split the

partner company `str` with `||`, as follow the same process as step `2` on each partner company name to get stock ticker. After getting stock tickers, I joined them back with `||`

Processing bmt_events file

1. Load bmt_events file (`bmt_events.tsv`)
2. `lead_company_ticker` column has the lead company's stock ticker
3. `partner_company_tickers` column has the partner company's stock tickers

Merging, saving and uploading

1. Add a column `lead_company_fsym_id` for the `fsym_id` for each company using the mapping file `fsym_to_ticker.tsv` .
2. Add a column `partner_company_fsym_ids` for the `fsym_id` for all partner companies using the mapping file `fsym_to_ticker.tsv` .

Part 2 Approach

To run the data pipeline use the following command,

```
python pipeline.py --config config/pipeline_config_api.yaml --output_path events.csv
```

Configuration file for Part II

```
# define properties for sa_events
sa_events:
  # path for sa_events file (supports url or file pointing to a csv, tsv, xlsx)
  source: 'https://raw.githubusercontent.com/harsha070/data-pipeline/048d989b592c374d4b88
bmt_events:
  # path for bmt_events file (supports url or file pointing to a csv, tsv, xlsx)
  source: 'https://raw.githubusercontent.com/harsha070/data-pipeline/048d989b592c374d4b88
# path for fsym_to_ticker mapping file (supports url or file pointing to a csv, tsv, xlsx
fsym: 'https://raw.githubusercontent.com/harsha070/data-pipeline/048d989b592c374d4b88bb87
# flag to upload output events to s3. if set to True, pass aws config (s3_access_key and
upload_to_s3: False
# define your AWS credentials
aws:
  s3_access_key: admin
  s3_secret_key: password
```

```
# cron job frequency (in minutes). Set to run the data pipeline runs every 1 minute.
job_interval: 1 # minutes
```

Part 2 Approach

Define a CRON job schedule for the data pipeline

1. Using `schedule` library, I run the data pipeline at a specific frequency
2. As the files are not static, I download the file and compare with existing file to get added and removed rows.
3. Compiled the added rows and deleted rows, and the final events csv file

Upload events csv file AWS S3

1. Using `boto3` library, I can upload the final events csv file to `S3` . Uploading to `S3` uses the `s3_access_key` and `s3_access_secret` specified in the configuration yaml file.