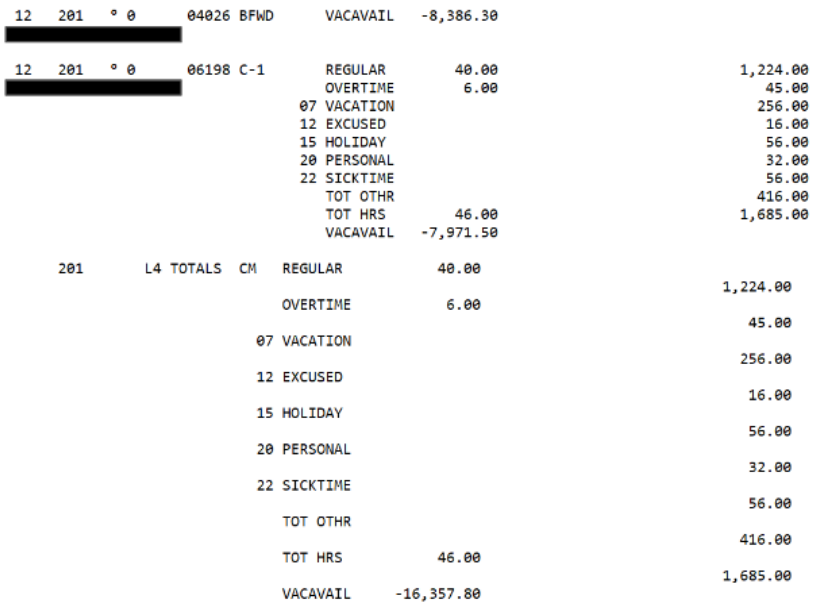
Capstone Project Report

Author: Sarvesh Shah, tuj80000@temple.edu

Problem Description:
SEPTA uses a legacy system to generate periodic reports related to financial data. These reports are exported in text format and do not have a consistent format. For some files, it is possible to use simple excel tools like 'text to columns' but for most files, it requires some additional data wrangling and cleaning. Until now, due to difficulties in data extractions from old files, SEPTA has not been able to make use of these generated reports. Some of these reports included files like employee pay stubs. Once the files could be mined, they could be used to improve business performance by generating key insights on business trends like spending on raw materials, salary information, budget allocations, etc.

Example of an unstructured file:



```
 12   201   ° 0      04026 BFWD      VACAVAIL   -8,386.30
[redacted]

 12   201   ° 0      06198 C-1       REGULAR        40.00                1,224.00
[redacted]                           OVERTIME        6.00                   45.00
                                  07 VACATION                            256.00
                                  12 EXCUSED                              16.00
                                  15 HOLIDAY                              56.00
                                  20 PERSONAL                             32.00
                                  22 SICKTIME                             56.00
                                     TOT OTHR                            416.00
                                     TOT HRS        46.00                1,685.00
                                     VACAVAIL   -7,971.50

     201       L4 TOTALS  CM   REGULAR           40.00
                                                                      1,224.00
                              OVERTIME            6.00
                                                                         45.00
                           07 VACATION
                                                                        256.00
                           12 EXCUSED
                                                                         16.00
                           15 HOLIDAY
                                                                         56.00
                           20 PERSONAL
                                                                         32.00
                           22 SICKTIME
                                                                         56.00
                              TOT OTHR
                                                                        416.00
                              TOT HRS           46.00
                                                                      1,685.00
                              VACAVAIL     -16,357.80
```

Objective:
The objective of this project is to implement a system and establish a workflow to generate reports from these raw text files. The steps are as follows:

1. Write scripts to mine files
2. Establish a data dictionary and establish a database
3. Create text outputs that can be imported into BI tools
4. Generate insights from these reports

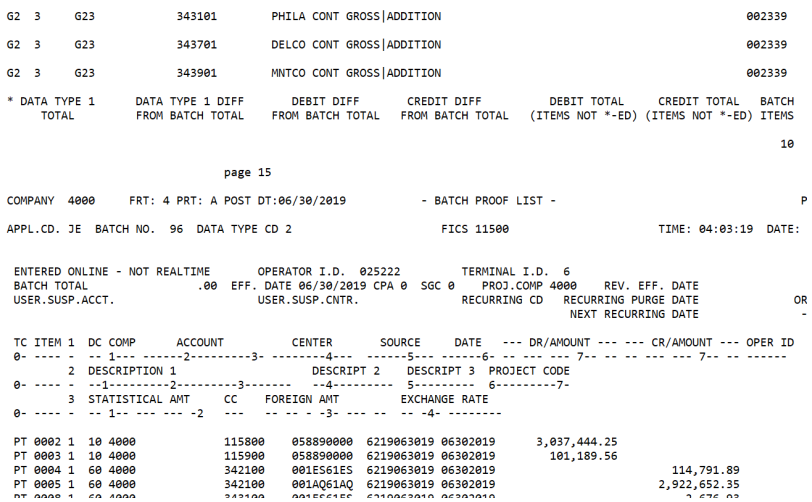
Approach/Methodology:
We plan to start this project by developing the system to mine the data. The language we plan to mine the data is Python, with Pandas and NumPy libraries. We plan to use regex and other Python libraries and functions to develop logic for mining. The next phase of the project also involves Python to use its data frames and high order aggregation functions to create summarized outputs and analysis from the data. Once we have these data outputs, we plan to

develop a SQL workflow to create a database and a data dictionary. Followed by automated dashboards that will process and present insights. The long term goal is to automate this entire workflow using ‘Windows Task Scheduler’ which will auto-update these dashboards biweekly.

Data Description:
The financial data of SEPTA comprises of 35 different textual unstructured files ranging from transactions amongst companies that SEPTA deals with and employee hours and wages information. This data is pumped out on a bi-weekly basis.

Data Issues and Solution Approach:
File Structure:

1. I encountered some serious issues when I was mining certain data files. For instance, there were interesting variations (header changes, structural changes due to different parts of the report) in the files that required me to write additional logic to isolate these cases and then systematically merge these isolated columns in the main files, preserving the order was of utmost importance in order to maintain the integrity of data.
2. Another challenge that we faced was that in certain files, the output file had just one column populated for the category stored in the previous column along with numeric data and I backtracked my code to aggregate and then roll up all those columns in a single row (where the pointer resided).



```
G2  3     G23            343101        PHILA CONT GROSS|ADDITION                                                  002339

G2  3     G23            343701        DELCO CONT GROSS|ADDITION                                                  002339

G2  3     G23            343901        MNTCO CONT GROSS|ADDITION                                                  002339

* DATA TYPE 1      DATA TYPE 1 DIFF       DEBIT DIFF       CREDIT DIFF         DEBIT TOTAL     CREDIT TOTAL   BATCH
     TOTAL         FROM BATCH TOTAL    FROM BATCH TOTAL   FROM BATCH TOTAL   (ITEMS NOT *-ED) (ITEMS NOT *-ED) ITEMS

                                                                                                                10

                                page 15

COMPANY  4000     FRT: 4 PRT: A POST DT:06/30/2019           - BATCH PROOF LIST -                                  P

APPL.CD. JE  BATCH NO.  96  DATA TYPE CD 2                      FICS 11500                      TIME: 04:03:19  DATE:


 ENTERED ONLINE - NOT REALTIME       OPERATOR I.D.  025222         TERMINAL I.D.  6
 BATCH TOTAL                .00  EFF. DATE 06/30/2019 CPA 0  SGC 0    PROJ.COMP 4000    REV. EFF. DATE
 USER.SUSP.ACCT.                 USER.SUSP.CNTR.                RECURRING CD   RECURRING PURGE DATE              OR
                                                                                NEXT RECURRING DATE               -

 TC ITEM 1  DC COMP      ACCOUNT          CENTER       SOURCE     DATE   --- DR/AMOUNT --- --- CR/AMOUNT --- OPER ID
 0- ---- -  -- 1--- ------2---------3- --------4---  ------5--- ------6- -- --- --- 7-- -- -- --- --- 7-- -- ------
        2  DESCRIPTION 1                   DESCRIPT 2    DESCRIPT 3  PROJECT CODE
 0- ---- -  --1---------2---------3-------   --4---------  5---------  6---------7-
        3  STATISTICAL AMT     CC    FOREIGN AMT         EXCHANGE RATE
 0- ---- -  -- 1-- --- --- -2   ---   -- -- - -3- --- --  -- -4- --------

 PT 0002 1  10 4000             115800    058890000  6219063019 06302019      3,037,444.25
 PT 0003 1  10 4000             115900    058890000  6219063019 06302019        101,189.56
 PT 0004 1  60 4000             342100    001ES61ES  6219063019 06302019                        114,791.89
 PT 0005 1  60 4000             342100    001AQ61AQ  6219063019 06302019                      2,922,652.35
```

Within the same file we have a different indentation and thus makes it harder to just split on spaces or a fixed length.

File Format and Name:

1. Since we assume that a user can pass any file as an input to the system, I wrote a file-validation code that ensures that the user selects a text file and the appropriate file from the file list that can be mined by the system. The user can also have files with strange alphanumeric characters in them like ‘Filename - 1’ or ‘Filename (1)’, etc., so I first cleaned the file names and directories and showed the user the core type of file before the mining script is executed.

File Version:

1. As the system pipes out data on a biweekly basis, there was a need for a time tracker that could tell the user the period that this file belongs. In order to achieve this, I wrote a small date extractor function that uses regex to specifically extract the report end date

from the entire text file and add it in the output file name as an extra column in the final excel file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	TC	ITEM	1	DC	COMP	ACCOUNT	CENTER	SOURCE	DATE	R_AMOUN	R_AMOUN	OPER ID	OMMENT	Report Date	
2	PT	0001	1	10	1100	101200	05011000	07040630	########	135473.5	0			2019-07-17 00:00:00	
3	PT	0002	1	60	1100	101200	05011000	07040630	########	0	84949.28			2019-07-17 00:00:00	
4	PT	0004	1	10	1100	830299	05112090	07040630	########	62658.58	0			2019-07-17 00:00:00	
5	PT	0005	1	10	1100	830299	05113090	07040630	########	5870	0			2019-07-17 00:00:00	
6	PT	0006	1	10	1100	830299	05114090	07040630	########	1877.38	0			2019-07-17 00:00:00	
7	PT	0007	1	10	1100	830299	05117090	07040630	########	14543.32	0			2019-07-17 00:00:00	
8	PT	0009	1	60	1100	840599	05112090	07040630	########	0	99925.24			2019-07-17 00:00:00	
9	PT	0010	1	60	1100	840599	05113090	07040630	########	0	9361.22			2019-07-17 00:00:00	
10	PT	0011	1	60	1100	840599	05114090	07040630	########	0	2993.97			2019-07-17 00:00:00	

Report Date added as an extra column later after extraction

Filling missing data:

1. The reports were not designed with a CSV or a spreadsheet format in mind and thus, in a lot of instances, the column will have a single cell with some data followed by empty cells until the value of that cell changes. In cases like this, first I had to identify these missing cells, mark them and use forward fill, backward fill or a combination of these functions to accomplish the task.

```
                1100  62401  5681  0900  88  10  0101              39
07/27/2019  07/27/2019
                2100  62401  5611  0900  88  10                    90
07/27/2019  07/27/2019
                2100  62401  5681  0900  88  10                    17
07/27/2019  07/27/2019
                2200  62401  5611  0900  88  10                    12
07/27/2019  07/27/2019
                2200  62401  5681  0900  88  10                     1
07/27/2019  07/27/2019
                2300  62401  5611  0900  88  10                     1
07/27/2019  07/27/2019

12   201   ° 0      04026 BFWD      VACAVAIL   -8,386.30
[redacted]

12   201   ° 0      06198 C-1       REGULAR        40.00                      1,224.00
[redacted]                          OVERTIME        6.00                         45.00
                                 07 VACATION                                    256.00
                                 12 EXCUSED                                      16.00
                                 15 HOLIDAY                                      56.00
                                 20 PERSONAL                                     32.00
                                 22 SICKTIME                                     56.00
                                    TOT OTHR                                    416.00
                                    TOT HRS         46.00                     1,685.00
                                    VACAVAIL    -7,971.50

     201       L4 TOTALS  CM   REGULAR            40.00
                                                                           1,224.00
                                 OVERTIME           6.00
                                                                              45.00
                              07 VACATION
                                                                             256.00
                              12 EXCUSED
                                                                              16.00
                              15 HOLIDAY
                                                                              56.00
                              20 PERSONAL
                                                                              32.00
                              22 SICKTIME
                                                                              56.00
                                 TOT OTHR
                                                                             416.00
                                 TOT HRS            46.00
                                                                           1,685.00
                                 VACAVAIL      -16,357.80
```

Data Cleaning:

1. As the entire unstructured file is essentially treated as a string, the structured version of the file required a check for data type and reassigning data type. A lot of columns, especially those with currency values, stored them in “$##,###” format. In order for any

numeric aggregation to be possible, I had to clean the '$' and ',' or '.' characters and convert them into numeric formats.

2. The date columns also had the same problem with most of them being a text and not following a standard format. Thus, I had to convert date strings into standard date-time objects for further slicing and dicing.

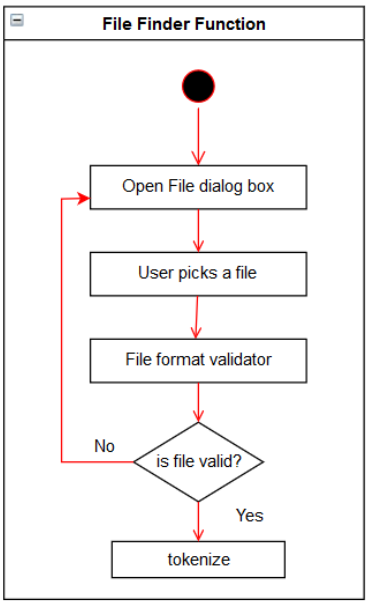
Code Workflow:

The code comprises of 2 modules - file finder, and file miner.

- File finder guides users to use a GUI of a file dialog box to open any file they chose to mine. Finder also generates a token that the miner uses to understand which script is to be executed for mining the given file.
- File miner once invoked, calls the appropriate function. Each function is conveniently named as a token itself to ease the nomenclature and execution process. The function then does all the cleaning, mining parts, aggregations if required, and handles the exporting. An excel writer object is invoked and it uses the file path and the data frame to export the cleaned file as a .xlsx file with the cleaned name and date attached to it.

File Finder Function

File finder function consists of a file browser interface is the only user-facing interface in the system, a user can select files using that window for further processing. This module has all the regex written to clean up the file path and tokenize the file path. This token invokes the appropriate function.



Flowchart of file finder function

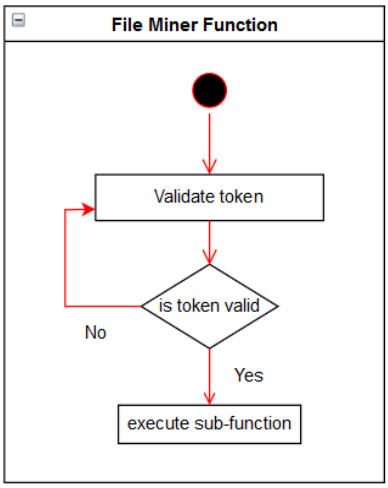
The user can only open a ".txt" file by default and if the user fails to open a ".txt" file, the user is given the appropriate message and sent back to the dialogue box

Once the user selects a text file, the filename is extracted from the file path and tokenized, the tokenization process involves cleaning of the name using regex and then verified against a list of tokens stored in function. If the file name is found, the token is used to run the appropriate function, otherwise redirected back to the file dialogue box.

File Miner Function:

File miner function takes the token and file path as an argument. All the sub-functions are

conveniently named as tokens for ease of use. I used the **'exec()'** function in Python. This function can execute strings as lines of codes. The miner function thus calls the function based on the token.



Flowchart File miner function

Example of a sub-function module:
Example 1:
Hours Register file: This file contains employee payment information for the given period. These text files are indented differently on certain pages (shifted left or right) and there are multiple text blocks for each employee with different information. The parser for this specific file type was designed to identify each of these line types individually, tokenize these strings, pad the white spaces with empty strings and then systematically merge them all in the end. The parser then uses pandas as **ffill()** and **bfill()** function to fill the missing empty text files created before. We then start cleaning columns one at a time. The currency columns are cleaned using regex and then converted to numeric type and date (if applicable) is converted to a DateTime object. The script adds management center information for each employee in a column next to the refined data and also adds a report date (mined separately using a regex function) as a column. We recreate the raw dataset from the information and provide the summarized outputs. These summarized findings are aggregated separately in a different sheet in the same excel file. These aggregations are computed using **Pandas.groupby()** function. Raw File:

```
12   201   ° 0      04026 BFWD      VACAVAIL    -8,386.30
[redacted]

12   201   ° 0      06198 C-1       REGULAR         40.00                    1,224.00
[redacted]                          OVERTIME         6.00                       45.00
                                 07 VACATION                                  256.00
                                 12 EXCUSED                                    16.00
                                 15 HOLIDAY                                    56.00
                                 20 PERSONAL                                   32.00
                                 22 SICKTIME                                   56.00
                                    TOT OTHR                                  416.00
                                    TOT HRS          46.00                  1,685.00
                                    VACAVAIL     -7,971.50

     201       L4 TOTALS  CM    REGULAR          40.00
                                                                           1,224.00
                                 OVERTIME          6.00
                                                                              45.00
                              07 VACATION
                                                                             256.00
                              12 EXCUSED
                                                                              16.00
                              15 HOLIDAY
                                                                              56.00
                              20 PERSONAL
                                                                              32.00
                              22 SICKTIME
                                                                              56.00
                                 TOT OTHR
                                                                             416.00
                                 TOT HRS          46.00
                                                                           1,685.00
                                 VACAVAIL     -16,357.80
```

The resultant file looks like this

B	C	D	E	F	G	H	I	J	K	L	M	N
L3	L4	L5	EMP NUMBER	EMP NAME	TYPE	HC	HOUR DESCRIPTION	Current Hours	YTD HOURS	MGMT CNTR	Period End Date	
12	201	° 0	[redacted]		BFWD	0	VACAVAIL	-8,386.30	0	12201	2019-09-07 00:00:00	
12	201	° 0			BFWD	0			0	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	0	REGULAR	32.00	1104	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	0	OVERTIME		31	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	7	VACATION		256	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	12	EXCUSED		16	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	15	HOLIDAY	8.00	56	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	20	PERSONAL		32	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	22	SICKTIME		56	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	0	TOT OTHR	8.00	416	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	0	TOT HRS	40.00	1551	12201	2019-09-07 00:00:00	
12	201	° 0			C-1	0	VACAVAIL	-7,971.50	0	12201	2019-09-07 00:00:00	
20	111	° 0			C-1	0	REGULAR	35.00	1340	20111	2019-09-07 00:00:00	
20	111	° 0			C-1	0	OVERTIME	26.50	840.5	20111	2019-09-07 00:00:00	
20	111	° 0			C-1	5	NGT SHIF		31.5	20111	2019-09-07 00:00:00	
20	111	° 0			C-1	7	VACATION	5.00	59	20111	2019-09-07 00:00:00	
20	111	° 0			C-1	15	HOLIDAY	8.00	64	20111	2019-09-07 00:00:00	
20	111	° 0			C-1	20	PERSONAL		24	20111	2019-09-07 00:00:00	

Final aggregation: Recreated from raw data

EMP NUMBER	EMP NAME	HOUR DESCRIPTION		
00	[redacted]	EXCUSED		12
		HOLIDAY	8.00	15
		NGT SHIF		5
		OVERTIME		0
		PERSONAL	8.00	20
		REGULAR	51.60	0
		REP/TURN	1.25	4
		SICK PAY		24
		SICKTIME		22
		TOT HRS	68.85	0
		TOT OTHR	17.25	0
		VACATION		7
		VACAVAIL	-828.00	0
	[redacted]	HOLIDAY	8.00	15
		NGT SHIF	56.00	5
		OVERTIME	24.00	0
		PERSONAL		20
		REGULAR	32.00	0
		TOT HRS	88.00	0
		TOT OTHR	32.00	0
		VACATION	-32.00	7
		VACAVAIL	-3,276.00	0
	[redacted]	HOLIDAY	8.00	15
		OVERTIME		0
		PERSONAL		20
		REGULAR	59.50	0

Once, the excel file object is ready. We also add the end date of the report to the file name to make tracking of these reports easy for the user.

Example 2:

Chart of Accounts: Chart of accounts stores a list of active centers along that is associated with an account number. As you can see in the photos below, based on the total number of active centers, there can be a lot of different columns and rows to store that information. The challenge here was to understand how many lines are part of this center alone and how many columns exist for this column.

```
B              5011 ACCT DESC NOT ON GLM       0  0  0  0     ACCT POINTER**
                                                          TOTS-ACTIVE CNTRS-      0 INACTIVE CNTRS-      0  *ERRS-   0 **ERRS-   1

B              5012 ACCT DESC NOT ON GLM       0  0  0  0     ACCT POINTER**
                                                          TOTS-ACTIVE CNTRS-      0 INACTIVE CNTRS-      0  *ERRS-   0 **ERRS-   1

B              5025 ACCT DESC NOT ON GLM       0  0  0  0     ACCT POINTER**
                                                          TOTS-ACTIVE CNTRS-      0 INACTIVE CNTRS-      0  *ERRS-   0 **ERRS-   1

B              5032 ACCT DESC NOT ON GLM       0  0  0  0     ACCT POINTER**        02101**        06201**        16631**
                                                                 16641**        18409**        20111**        20121**
                                                                 33231**        55301**        55304**        55402**
                                                                 55403**        55501**    999999999**
                                                          TOTS-ACTIVE CNTRS-      0 INACTIVE CNTRS-      0  *ERRS-   0 **ERRS-  15

B              5033 ACCT DESC NOT ON GLM       0  0  0  0     ACCT POINTER**        06101**        06201**        06301**
                                                                 11103**        15101**        17101**        17701**
                                                                 18101**        31101**        32201**        32301**
                                                                 32401**        32501**        32601**        32801**
```

Other section of the same report

```
                                                          TOTS-ACTIVE CNTRS-      0 INACTIVE CNTRS-      0  *ERRS-   0 **ERRS-  42

B              5034 ACCT DESC NOT ON GLM       0  0  0  0     ACCT POINTER**        01001**        01002**        01101**
                                                                 01111**        02101**        03101**        04101**
                                                                 06101**        06201**        06301**        06501**
                                                                 06601**        06701**        07101**        07201**
                                                                 11101**        11102**        11103**        13101**
                                                                 13502**        13503**        14301**        14811**
                                                                 14812**        14813**        15101**        16101**
                                                                 16201**        16601**        16621**        16622**
                                                                 16631**        16641**        17101**        17701**
                                                                 18101**        18102**        18201**        18205**
                                                                 18224**        18301**        18401**        18402**
                                                                 18405**        18406**        18409**        18501**
                                                                 18522**        18531**        18601**        18622**
                                                                 18624**        18701**        18723**        19211**
                                                                 19212**        19213**        19216**        20101**
                                                                 20111**        20121**        20131**        20141**
                                                                 25911**        25931**        31101**        31201**
                                                                 32101**        32201**        32301**        32401**
                                                                 32501**        32601**        32701**        32702**
                                                                 32703**        32801**        32901**        33101**
                                                                 33211**        33231**        33311**        33331**
                                                                 34201**        34301**        34401**        34501**
                                                                 38101**        38201**        38301**        38401**
                                                                 38501**        40101**        41101**        41201**
                                                                 41301**        41401**        41501**        41601**
                                                                 41701**        41801**        41901**        42101**
                                                                 42201**        51101**        51102**        51103**
                                                                 52211**        52311**        52312**        52321**
```

In this case, I wrote a logic that would backtrack and merge the succeeding line with the current line. And if there are multiple rows and columns with center information, we use a list to essentially create a list of lists to store all these columns and this list of the list is attached with the main column. Thus collapsing these extra rows. The next part of the code is to melt this nested data structure and add a row for each center.

The output for this file looks like this:

23	B	4042	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
24	B	4043	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
25	B	4045	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
26	B	4050	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
27	B	4051	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
28	B	4052	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
29	B	4060	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
30	B	4061	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
31	B	4062	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
32	B	4063	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
33	B	4071	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
34	B	4072	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
35	B	4073	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
36	B	4074	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
37	B	4075	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
38	B	4076	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
39	B	4077	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
40	B	4078	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
41	B	4079	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903
42	B	4080	ACCT DESC	0	0	0	7	ACCT POINTER**8	TOTS-ACTIVE CNTRS-9	10	011	INACTIVE	...	5901	5902	5903

Limitations

The script is unable to handle minor inconsistencies in some data files because sometimes raw data output is mixed up. At the moment, the script can only clean the data after it is parsed and structured. These discrepancies are unpredictable and random and we cannot exhaustively go over them, thus it is not feasible to write a correction logic for all possible test cases. The script works on the assumption that the given file type will follow the same structure throughout and will not be able to give correct outputs in such cases. The headers for these files have also been hardcoded within the sub-function under the same assumption stated above. The script can only mine .txt extensions at the moment although incorporating a .pdf parser is feasible and easy to implement due to the modularity of code.

Future Scope

The future scope is to have this script run periodically. Instead of it running manually, we will use the ‘Windows task scheduler’ or equivalent to automate the script. SEPTA does not have an API to extract new files and thus the file finder function will be repurposed to automatically find the new files using a date identifier or a pre-decided file path. The excel writer module will be replaced with a SQL connector and data will be piped into DBMS. The BI tools will then use SQL connectors to update the dashboards instead of the excel file as their data source.