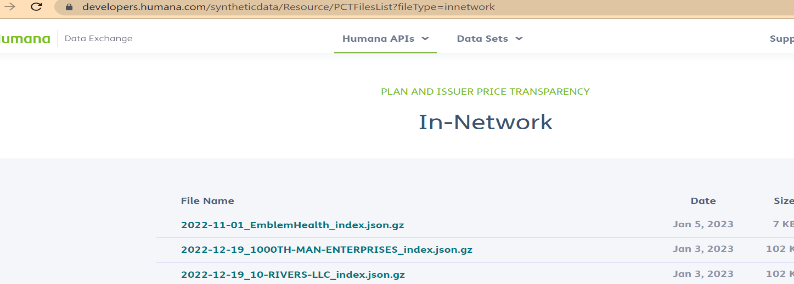
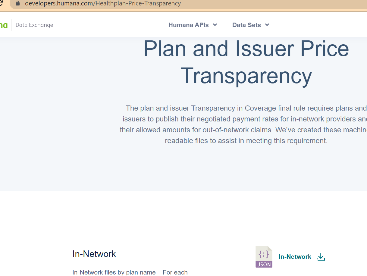
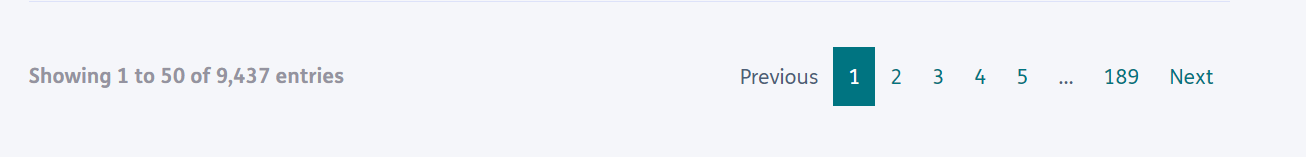
**Humana Files for Transparency in Coverage**

Humana places its Price Transparency in Coverage files at <https://developers.humana.com/Healthplan-Price-Transparency>

Clicking on the first option takes us to the in-network files.





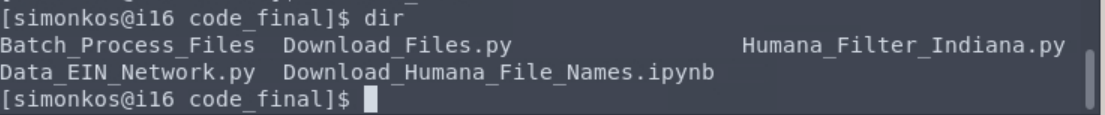
(Note that in Jan 2023, which is this screen shot, the number of files is far smaller than in 2022, when typically there was about 460,000 entries and about 9,000 pages).

There are two file types there.

1. Index JSON files- this file lists all the CSV GZ files which comprise that particular Plan ID/EIN (example 2022-09-29\_ZWIL-INC\_index. json). There are sometimes multiple plans within that entity.
2. CSV GZ files– consisting of all the data fields the CMS template for the files requires, apart from the EINs/Plan ID (example 2022-09-22\_1\_in-network-rates\_000000000000.csv.gz)

The way we interface with this website is to use the steps below to learn about the data and deduce some of the patterns:

The code referred to below is saved in ““/N/project/TIC/Humana/code\_final/” where the contents are below. The “Download\_Humana\_File\_names.ipynb” file has the code to run



1. Text

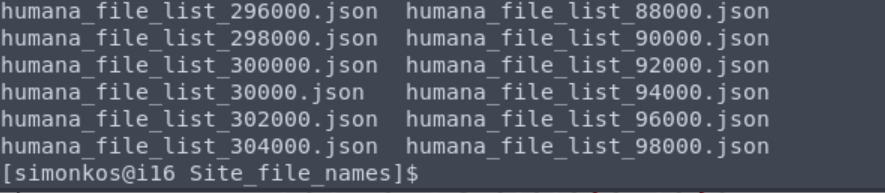
   Description automatically generated with medium confidenceWhat this line of code will do is download just the file names (index JSON files and CSV GZ files) for each month.. Use the code file mentioned below to download the file names.:

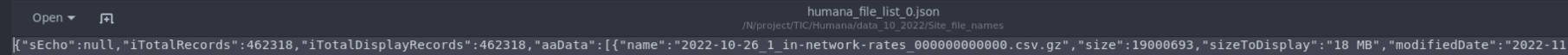
**NOTE:** Check the URL <https://developers.humana.com/syntheticdata/Resource/GetData?fileType=innetwork> for parameter “iTotalRecords” and put that in place of 15249.

It will then save the file names in chunks in the location below ( image as shown)

/N/project/TIC/Humana/data\_10\_22/Site\_file\_names/

(that directory will look like this. The number after the “\_list\_..” is the item number from the Humana website. Notice these increment by 2,000, so this list below has about 230 separate files, with 2,000 entries each, which account for about the 460,000 separate entries we saw above.



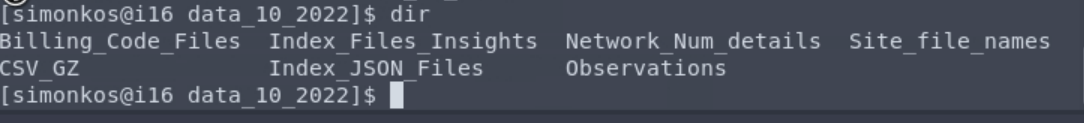
If we open one of the .json files above, it will look like this screen shot below (will have 2,000 such lines in each file)

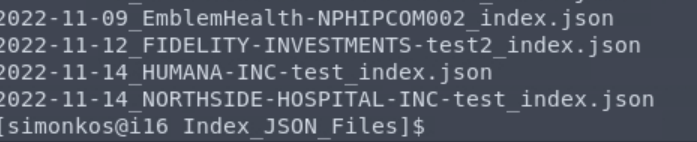
1. Once the name of the files are saved in the local folder, we can list all the ones that are index json files, then we will download just those files, using the code below (from Download\_Humana\_File\_names.ipynb” saved in the location as “/N/project/TIC/Humana/code\_final/” )

Graphical user interface, text, application, email

Description automatically generated

We download the data directly into a folder called “/N/project/TIC/Humana/data\_10\_2022/ Index\_JSON\_Files/” (This now consists of all the index JSON files in Humana)



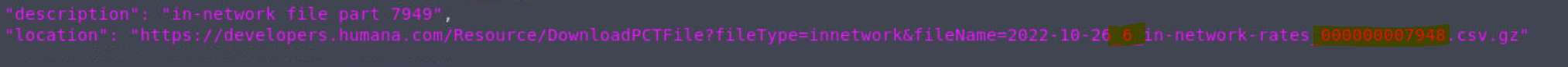


1. Next, we download the CSV GZ files (there are MANY of them!) using the batch process, based on what we learn from the INDEX JSON files.

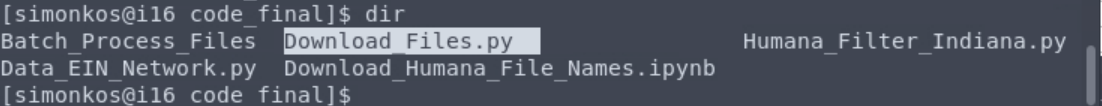
See “Network Information Extraction.docx” for explanation of how we use the JSON index files to figure out what are the most often used networks.

We observe that an employer/entity uses all the files from a given network number. i.e. all the sub-network files are used from any of the network number.

e.g., Big Brother Big Sisters uses all the sub-network files from 2,1 and 6.



In the above image, highlighted image number “1” is the network number and highlighted image number “2” is the sub-network file (that is, network 1 has been split into many small files that are numbered consecutively.

The parameter below should be updated in line number 138 with the file name as “Download\_Files.py” saved in the path “/N/project/TIC/Humana/code\_final”, based on the network number to be downloaded. (And this is used just for downloading xsv gz files). That is, we know that say, network 1 has 17,000 subparts. We don’t have code to report out that 17,000 number, we have to manually discover it based on looking at a JSON index file and seeing that its always 17,000 subparts for network 1, for example. That’s ok, as we know from the “Network Information Extraction.docx file that there are a limited number of networks, and the JSON index files that correspond to each. 

The sub-network file numbers are not sequential in nature. Therefore to obtain all the sub-network numbers we use the code below saved in the Download\_Humana\_File\_names.ipynb.

Text

Description automatically generated

Update the below highlighted URL in the Download\_Files.py to the latest path from one of the index JSON file.



**NOTE:** The other files used for batch processing are saved in “/N/project/TIC/Humana/code\_final/Batch\_Process\_Files”.

In the py file, you will set the parameter to end at 17,000, for example. We divide it into 400 or so jobs running simultaneously.

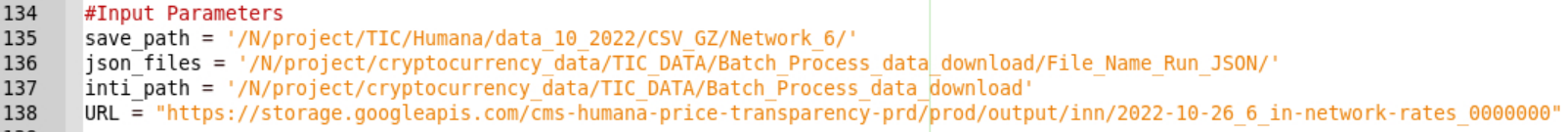
The steps to set up the batch processing are as follows:

1. Create the CSV file that will be divided into small parts or chunks and consists of all the subnetwork numbers which are part of the main network based on our observations in the index JSON file. The CSV file will consist of the file numbers from each network as shown below:

A picture containing text, electronics

Description automatically generated

1. Write the complete file path of the CSV file which was generated in the previous step in the run\_code.py file having variable name as “ofilenames”.
2. Text

   Description automatically generatedKeep all the codes for the batch process in one directory and write the path in “codedir”. Update the file paths of main directory and create sub-directories as bash, code, etc. in the “get\_dirs” function mentioned in option\_util.py as shown in the image:
3. All the Humana\_Billing\_Code.py file consists of the file paths as mentioned:
   1. save\_path – To save the final output files. e.g., downloaded CSV\_GZ files, CSV file consisting separate billing codes, etc.
   2. json\_files – Saved small chunks of JSON files after running the run\_code.py file. Mention the same file path as in “get\_dirs” function in option\_util.py file.
   3. inti\_path – Mainly used to save the log/file tracking for the completed/processed data.
   4. URL – As we download one network file at a time, mention the common path as part of the URL variable.
4. Save all the files and open “Terminal” with file path that consists of all the codes.
5.  Write the below mentioned code and execute it in the terminal.

**NOTE:** “—chunks” parameter specifies the number of chunks to be created for the process.

1. Graphical user interface, text

   Description automatically generatedThe programs/jobs which are in progress or queued can be reviewed timely using the Terminal with the code “squeue –user=” and username of the user as shown below: