**Process to Extract Specific Billing Code from Raw CSV GZ Files**

After the Humana TIC files are saved, we create analytics data sets out of it. We extract billing codes of data at a time, in the process described below. This makes sense because most of the research will be done for specific billing codes at a time, and because the raw data files are so large that we need to “chunk them” down to smaller files to analyse anyway. Once given a set of billing codes needed for the research, we follow the steps below to create the data extract.

In order to extract the billing codes, we have to use 3 code files that are as follows:

1. run\_code.py
2. option\_util.py
3. Humana\_Billing\_Code\_CSV\_GZ.py/ Humana\_Billing\_Code\_JSON\_GZ.py

Each python file serves a particular purpose that are as mentioned below:

1. run\_code.py File – It helps to create the batch script, a file used to execute a program at High Performance Computer specifying the application to launch and the resources required to run it e.g., memory, type of partition, wall time, etc. and divide the items into small chunks of CSV GZ files. The chunks are created in the JSON file format and saved separately in the folder. In the case of Bash files, all the parameters are defined which are required by Slurm as defined by the HPC team e.g., wall time, cpus, gigs, partition, etc. Jobs are submitted on the Slurm using the same bash file.

2. option\_util.py – This file consists of functions that are used to get the arguments from the input parameters and interpret it. As defined in the step 7, parameters like --runcode”, --fun\_arg”, “--chunks”, etc. are used and defined in this code file. The file path e.g., main, code, bash, input json directories are defined in this which are later used to execute the code.

3. Humana\_Billing\_Code.py – The python file consists of the code to extract individual billing code which are executed in the Slurm in batch mode using a small number of CSV GZ/JSON GZ files that depends on the chunk size. The small chunks are saved as input in the JSON file to process the billing codes.

**NOTE:** Keep all the .py code files in a single folder and mention it as specified in step 3.

The input CSV file consisting of complete file paths of CSV GZ/JSON GZ that are used for the process must be divided into small parts or chunks.

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

1. Create the CSV/JSON file that will be divided into small parts or chunks. The CSV file will consist of the complete file paths of the CSV GZ/JSON GZ from each network as shown below:

A picture containing text, newspaper

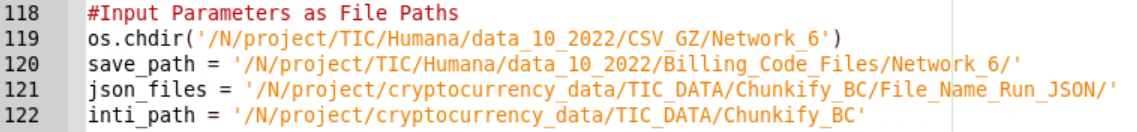
Description automatically generated

The code used to generate the CSV file is as follows:

A picture containing text

Description automatically generated

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

   Description automatically generatedUpdate the file paths of main, 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 for CSV/JSON files 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. Update the required billing code in the Humana\_Billing\_Code.py file, particularly in the variable “bl\_code” as shown below:
5. Save all the files and open “Terminal” with file path that consists of all the codes.
6. 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:

1. Once all the programs/jobs are executed, all the small chunks must be consolidated using the below mentioned code:

Graphical user interface, text, application

Description automatically generated

The code used to extract the details of the billing code is as follows:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

