DATA PROFILER – ANALYZING EHR DATA

The software package in this toolkit is auto-generated around the dataset (the files in the 'Data' folder) to offer the Investigator a common set of tools to interact with their EHR datasets. The software library is written in Python using the Pandas data analytics library on the Jupyter Notebook platform.

The immediate aim of the project is meant to help Investigators with tools that complete basic and some intermediary data analysis. But, it also aims to educate and encourage Investigators to expand their knowledge of analytics using Python and Pandas.

USEFUL FUNCTIONS

The main library 'ehr_dp_lib.py' is made up of a set of functions that aid in analytical tasks. Depending on the dataset the code blocks are automatically generated:

describe_tables(): Returns a dataframe listing all the files in the 'Data' folder including row and column counts and descriptions

missingness(*dataframe name*): Returns a dataframe of the number of null values per column.

catbar(*dataframe name, column name, graph=(True or False)*): [Generated on *categorical* data type only] Returns a dataframe of counts of all the groups of categories in the specific column in the dataframe. When `graph` argument set to `True` returns a bar graph.

numstats(*dataframe name, column name*): [Generated on *number* data type only] Returns a dataframe of descriptive statistics (ie. mean, max, min, median, quartiles) for the column data.

dateline(*dataframe name, column name*): [Generated on *date* data type only] Returns a line graph of the freuency of specific dates along an x-axis of time.

flow_stats(*flowsheet dataframe*): [Generated only if Flowsheet_Vitals.csv table in Data folder] Returns a dataframe of descriptive statistics for common vitals sign types (ie. Height, Weight, Temperature, Sp02, Pulse, BMI, Respirations).

lab_stats(*lab dataframe, top=(10 or greater)*): [Generated only if Labs.csv table in Data folder] Returns a dataframe of descriptive statistics for top lab procedures in dataset. The `top` argument can be adjusted to capture more lab procedures.

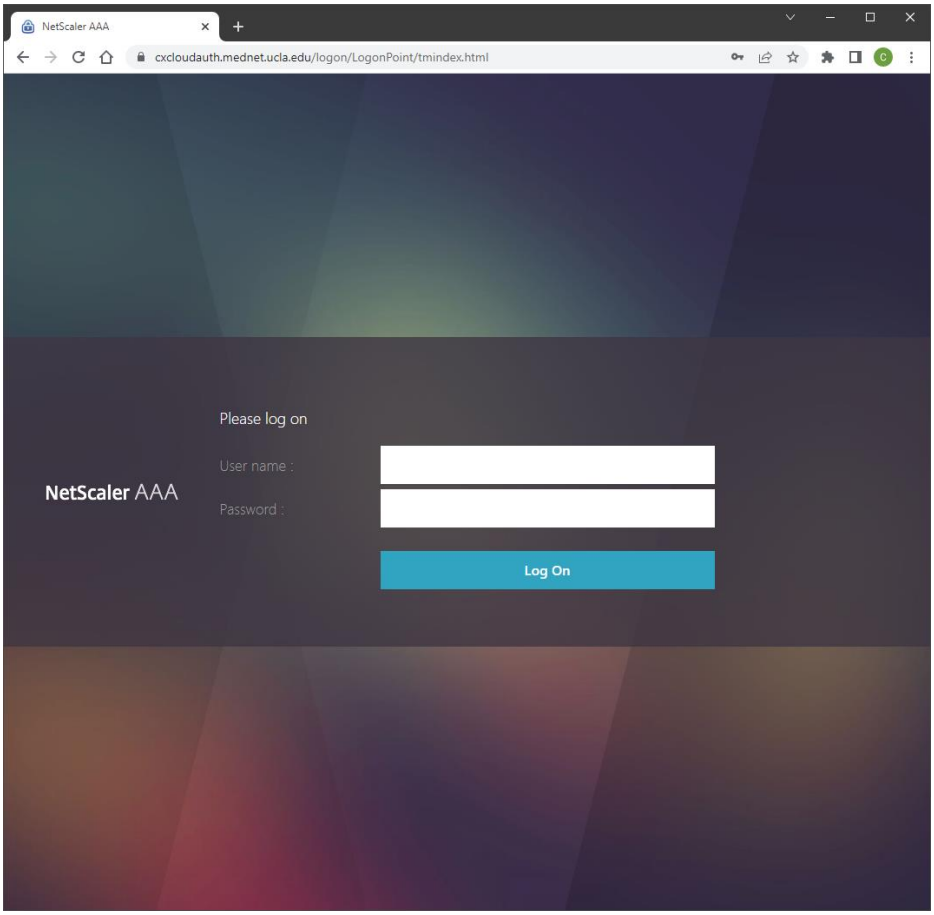
text_search(*dataframe name, column name, text to search, ignore case=(True by default can also be set to False)*): Returns a dataframe based on a free text search of a specific column in an existing dataframe.

ACCESSING ULEAD WITH ANACONDA AND JUPYTER NOTEBOOK

STEP 1:

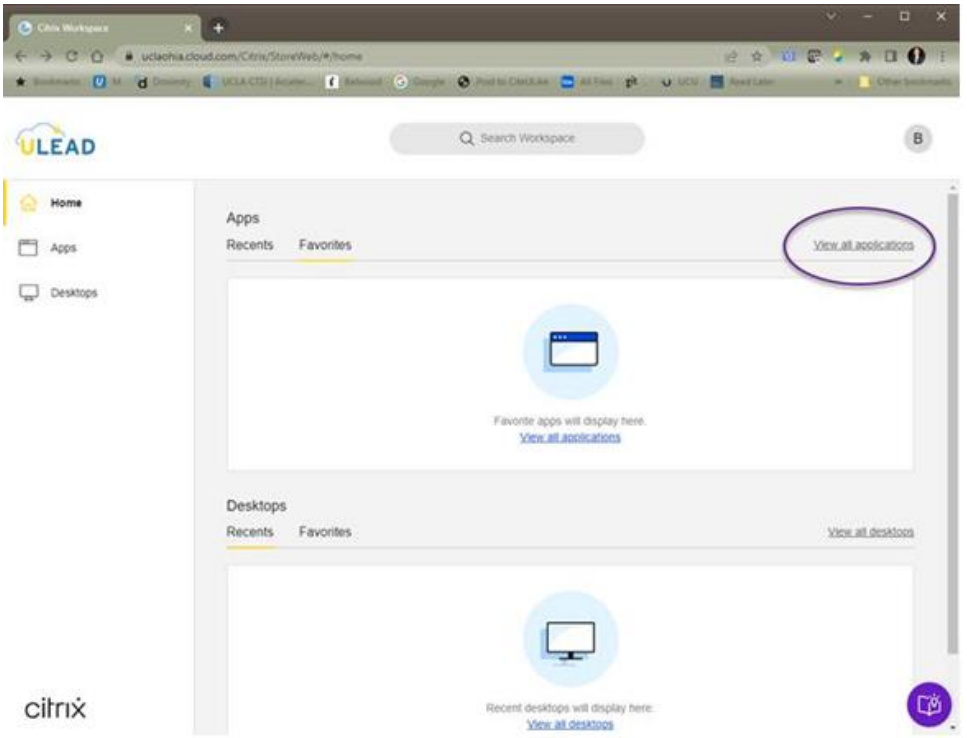
Navigate to: https://uclaohia.cloud.com/Citrix/StoreWeb/

The browser will re-rout and will present a login, enter your AD username / password. Logging in will also prompt the 2FA with Duo.

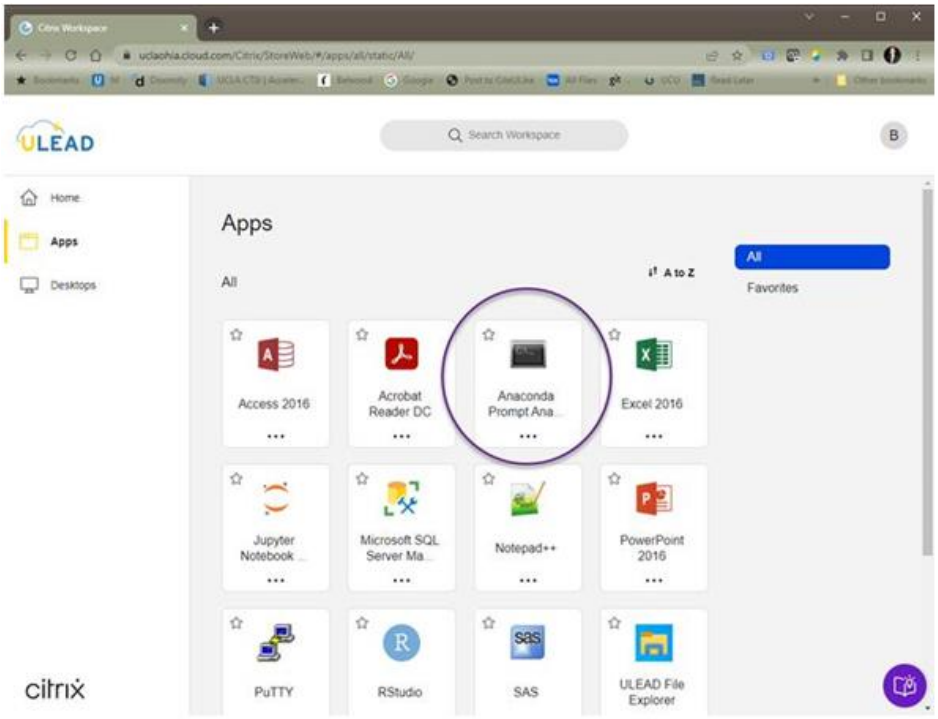


STEP 2:

Once logged in, you will see the ULEAD Apps screen. If not already listed, click on the link: ‘View all applications’



Then click on the icon ‘Anaconda Prompt’



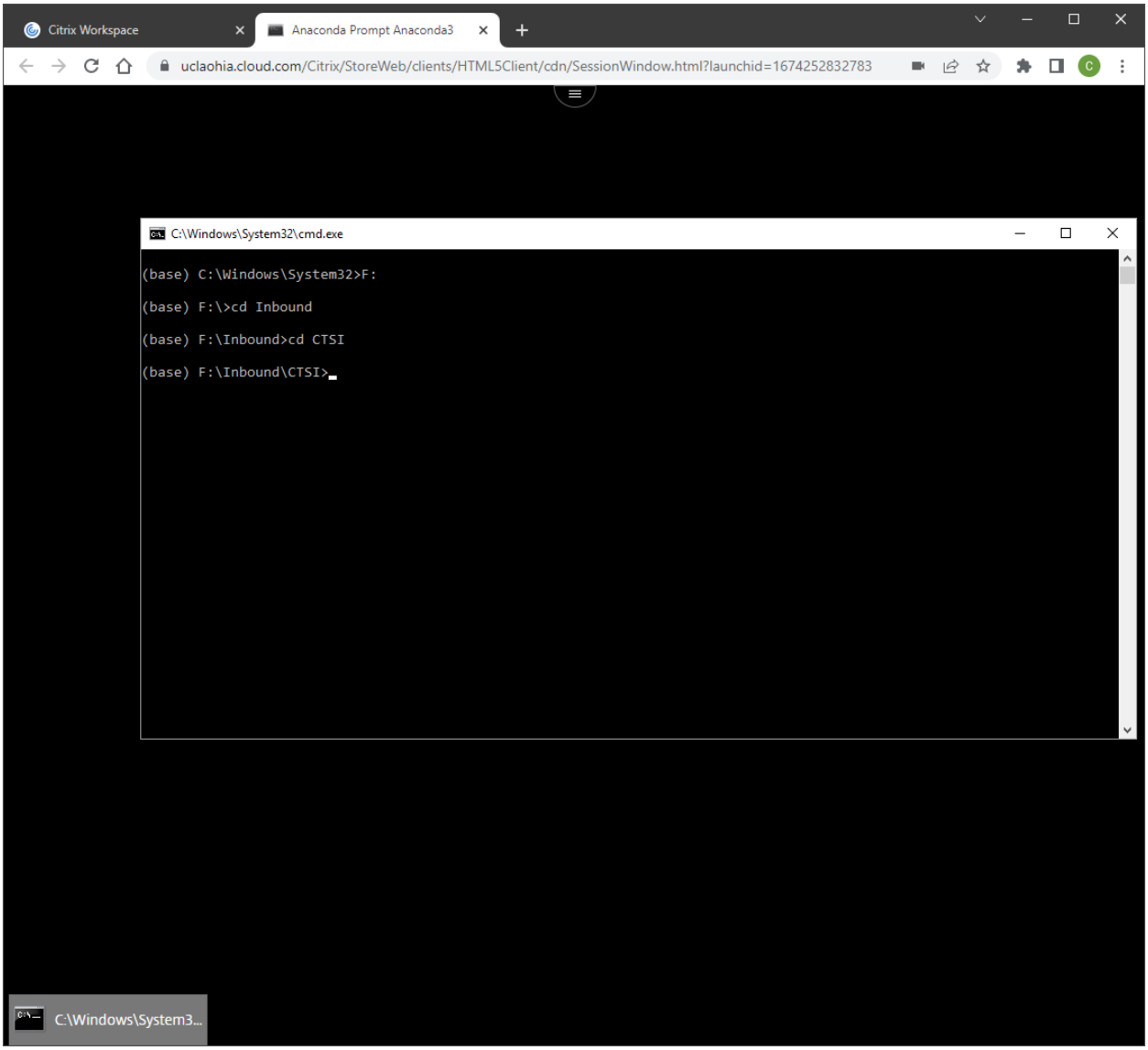
STEP 3:

Once finished loading a prompt will appear in the window in a virtual screen.

In the prompt enter the following commands:

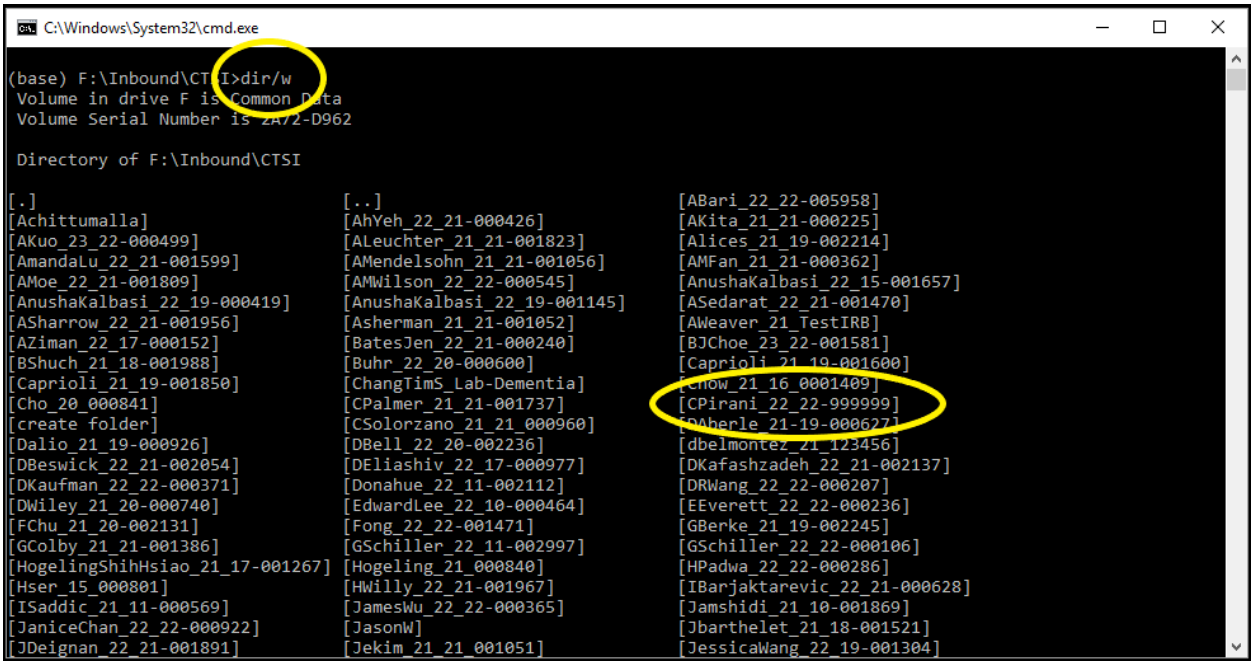
- F: <enter>
- cd Inbound <enter>
- cd CTSI <enter>

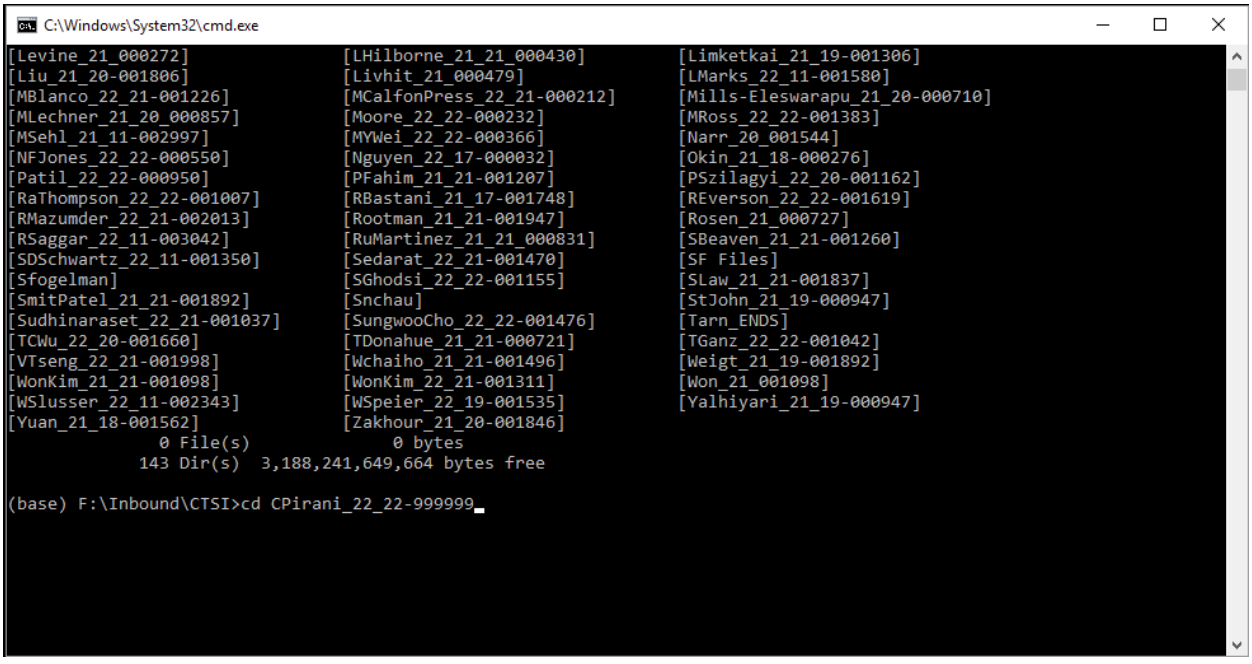
After this the prompt should look like this:



STEP 4:

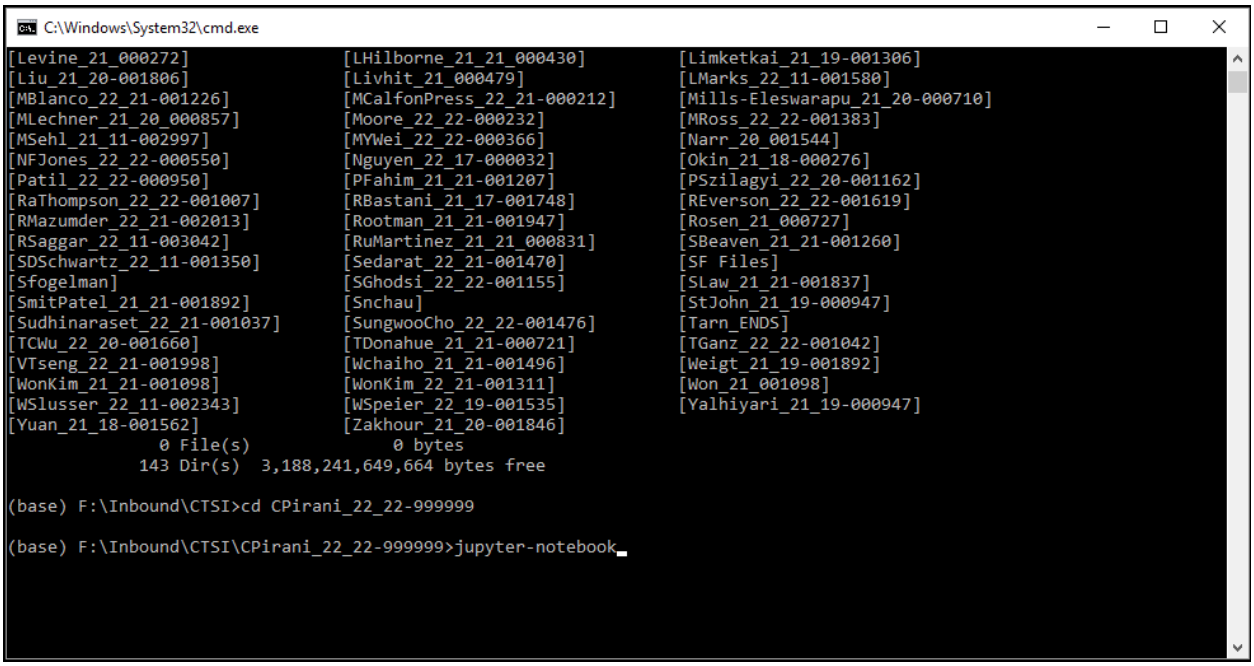
Now enter the command “dir/w” to list all of the folders in the directory, find your folder and enter: “cd <folder name> <enter>”





STEP 5:

Now enter the command “jupyter-notebook” and press <enter>. After a short time it will open a Chrome browser with the Jupyter Notebook explorer in your own ULEAD folder.



At this point you can click on the file: ‘Data_Profiler.ipynb’ to launch the notebook: