**Documentation for Battery Archive Upload Script**

The upload script utilizes google libraries and apis to upload the battery data into a google bucket. A google bucket is a storage medium where people can download and upload data to for various purposes. The bucket is anonymous and write only meaning you can only upload data to it. This prevents people from modifying the data within the bucket or seeing other data that has not yet been processed through battery archive. The script recognizes valid data formatting’s and will upload both single file batches and folder batches, but not intermixed batches with data in both files and folders.

*Diagram

Description automatically generated*A picture containing icon

Description automatically generated

Data

**Requirements**

1. A way to run a python script [recommend some options and provide links]

* Install latest version of python on your machine: <https://www.python.org/downloads/>
* Guide to installing python: <https://www.infoworld.com/article/3530140/how-to-install-python-the-smart-way.html>

1. Install [docker](https://docs.docker.com/get-docker/)
2. Clone the git repo at Client-tools-BLC using the git clone command and then run:

*pip3 install -r requirements.txt*

*(before this it is recommended to make a virtual env within the repo)*

1. Make sure you have a [google cloud Account](https://cloud.google.com/gcp?utm_source=google&utm_medium=cpc&utm_campaign=na-US-test-all-en-dr-bkws-all-all-trial-e-dr-1009892&utm_content=text-ad-none-any-DEV_c-CRE_491349594127-ADGP_Desk%20%7C%20BKWS%20-%20EXA%20%7C%20Txt%20~%20Storage%20~%20Cloud%20Storage_Cloud-KWID_43700060017921803-kwd-6458750523&utm_term=KW_google%20cloud-ST_google%20cloud&gclid=EAIaIQobChMIqNq6meOR9AIVX21vBB3CggvzEAAYASAAEgL8nvD_BwE&gclsrc=aw.ds) – see instruction below if you do not
2. **Run the following to get dockerized google sdk, replacing the pwd variable with wherever you have cloned the repo to**

***docker run -it -v $(pwd):/client-tools-blc google/cloud-sdk /bin/bash***

Once Google SDK is installed and you have a google cloud account, run the following command in your computer terminal:

*gcloud auth login*

This should take you to a list of google accounts from which you can select arbitrarily. Once you’ve selected an account, create a project.

Graphical user interface, text, application

Description automatically generated Select Account

If you do not have an associated GCP account with that gmail login, you must create an account. Once you create an account google awards you $300 worth of promotional credits.

Graphical user interface, application

Description automatically generated

**When you create an Account**

Then create a random project (select new project and name it whatever). Then rerun the gcloud auth login command using the account you just created

Graphical user interface, text, application, email

Description automatically generated

**Set up data folder for upload**

The data is expected to be structured like this:

All\_Data\_folder

---datatset\_1\_folder (we will be uploading this folder with everything in it)

--------cell\_list.xls(x)

--------data.xls or data folder with excel files

--------another data.xls or data folder with excel files

--------another data.xls or data folder with excel files (however many is fine)

--potentially other dataset that will need to be uploaded separately

Example Upload (Folder Structure)

cell\_list.xlsx

Graphical user interface, application, table, Excel

Description automatically generated

Folder Structure for SNL datasetTable

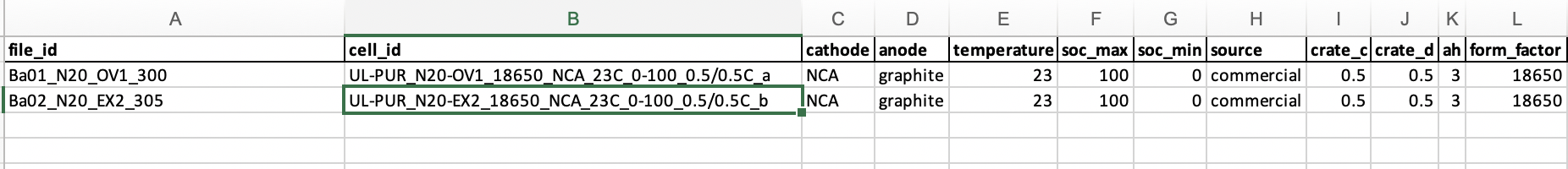
Description automatically generated with low confidence

Text, table

Description automatically generated

Example Upload (single files)

cell\_list.xlsx



Folder Structure for ul\_data dataset

Graphical user interface, text, application

Description automatically generated

Make sure to name the files/folders exactly as they are referred to in the cell list followed by the appropriate file extension(xls, xlsx, csv). As for files in folders there is no naming convention and they will all be processed. Also make sure that no junk files are present that are not data related (the dataset should only contain data files of the appropriate file type and the cell list). If you upload cells as folder as in the first example these folders can also be zipped if you prefer. They will be unpacked either way.

Data File Rules:

The data files must contain the following columns:

["Cycle\_Index","Test\_Time(s)", "Current(A)", "Voltage(V)", "Date\_Time"]

If any of the following columns are missing then the script will be unable to upload the data to the google bucket and an error will appear telling you that you are missing columns. If you are missing date time however that column will be inserted for you automatically.

**Upload the data**

1. Run

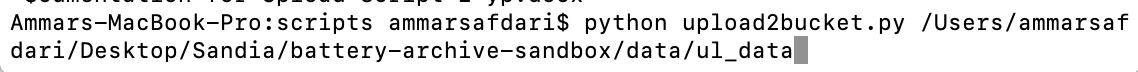
*gcloud init*

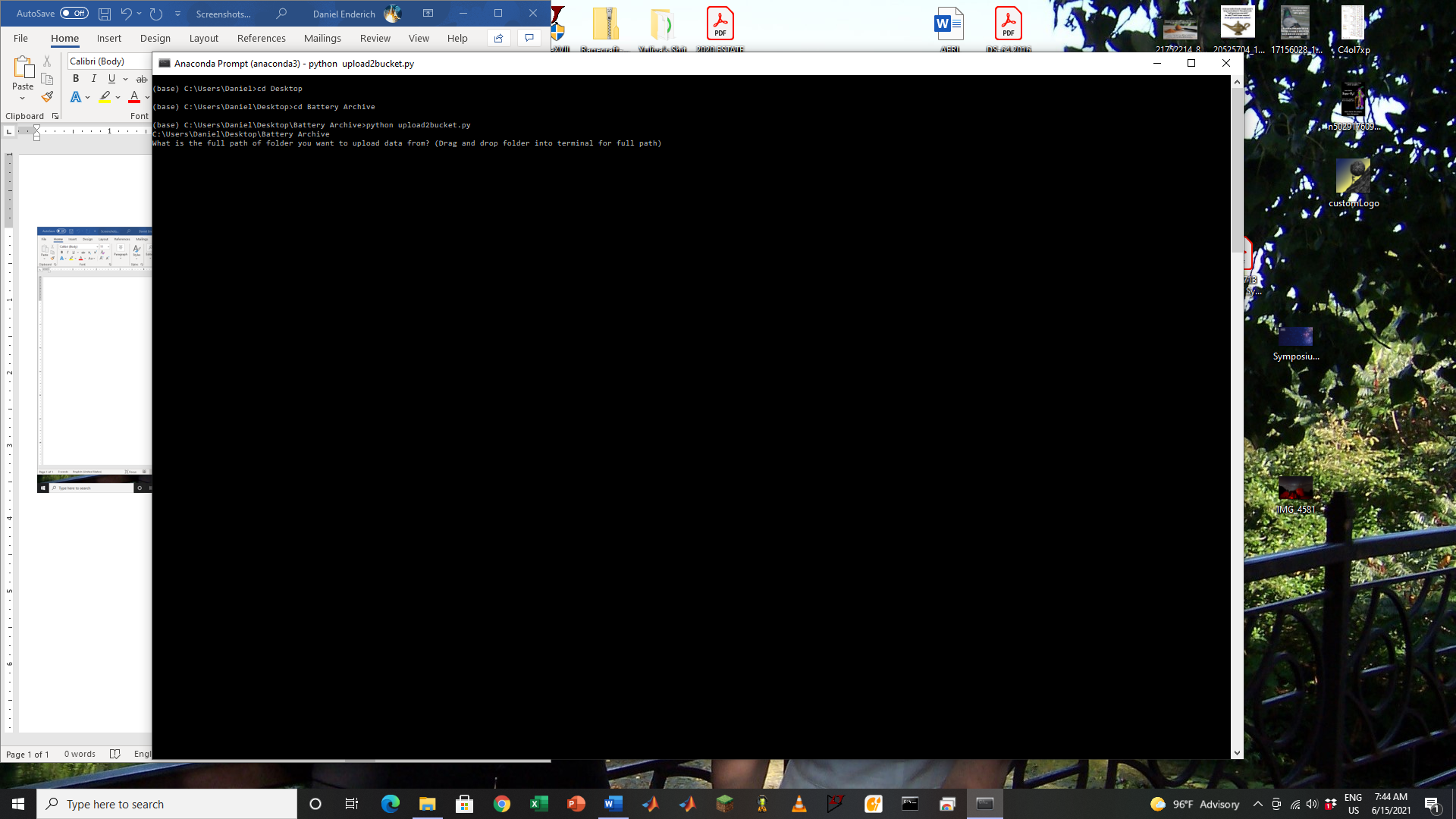
*Select the appropriate account to sign in with and project*

1. Run the following command to initiate data upload. Make sure the terminal window is set to the directory of the script.

*python upload2bucket.py {path to data}*

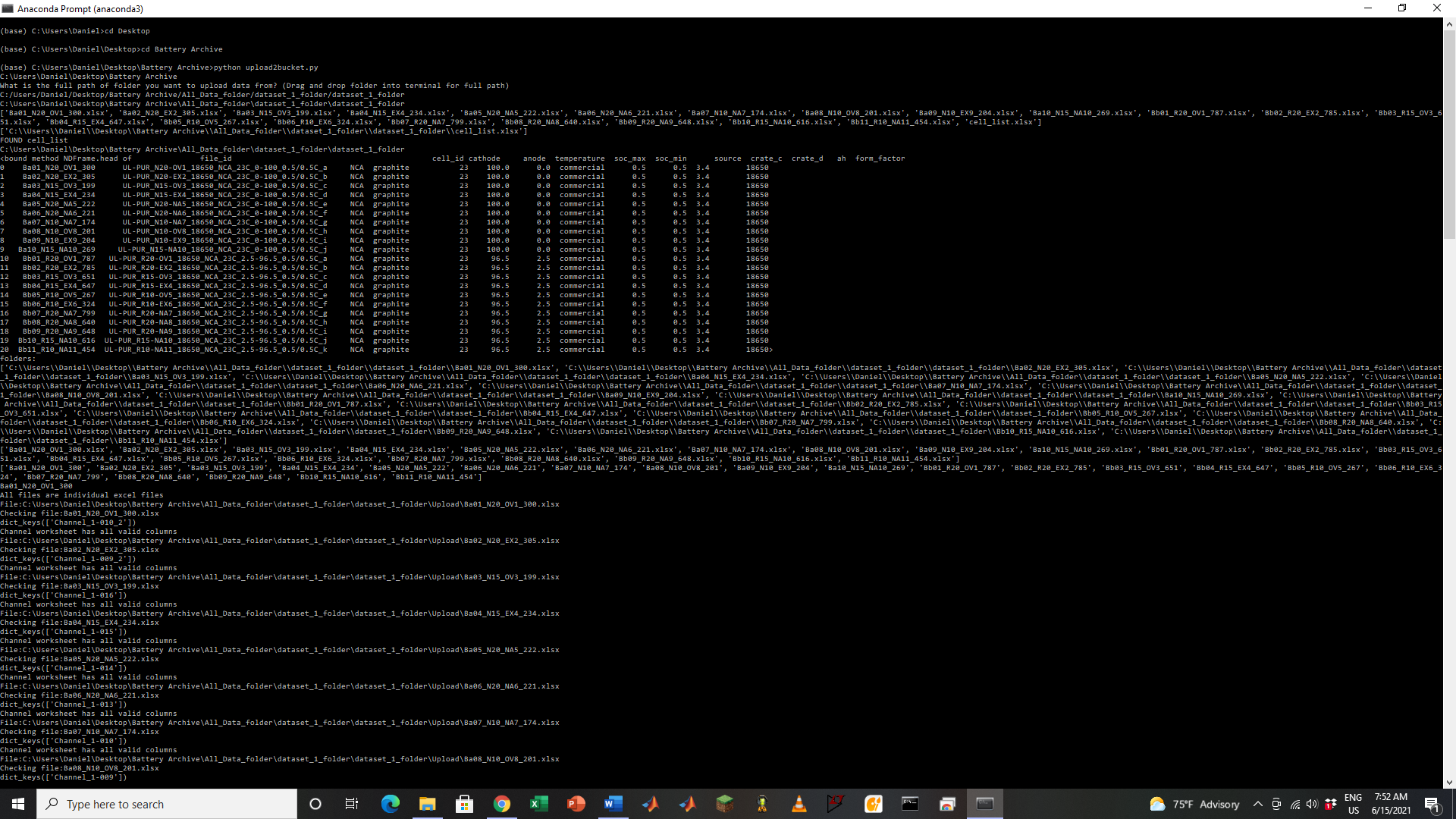
1. The program will ask you to write the full path name of the data folder you would like to upload (in mac you can just drag and drop folder into terminal).





Notes: remove quotes around names if you’re on a Windows machine

The program will immediately begin checking that all of the data files have the correct structure.



**Additional Notes:**

The script will print out error messages if any part of the uploaded file is incorrect. To successfully upload you must fix these messages.