

Matthew Hileman and Sydney Fowler

CS 3030.001

Final Project Report

18 December 2019

Project Link: <https://github.com/sydneyfowler/CS3030>

## **EXCEL COMMAND LINE TOOL**

### **Introduction**

The purpose of this project was to create a command-line tool of various functions used to work with Excel files. These functions include:

- Importing / exporting Excel files from and to csv
- Data validation and cleanup features for common cleanup functions (phone numbers, email addresses, state codes, zip codes, dates, web addresses, character limits, etc.)
- Compressing files
- Sending a copy of a file to a list of email addresses
- Removing duplicate rows based on a set of criteria
- Data Analysis: Sum, Count, Max, Min, Unique Values, Average, etc.
- Graphing: Taking data from an excel file and plotting to a chart

### **Theory/Existing Other Projects**

There exist many tools and projects that utilize Python to read and manipulate excel files -

OpenPyXL, pandas, NumPy, matplotlib, and various others prove themselves powerful. However,

a user who is unfamiliar with python or coding may have difficulty making use of these tools. A general user driven design, such as some features of [Pyexcel](#), allow users with very little coding experience to export and import excel files with an interface that opens in a browser. Likewise, this project aims to create a navigable command line interface for general users (with little to no programming experience) to still make use of some of these helpful excel utilities. Our project is intended for those looking for a tool to quickly clean and manipulate large excel files.

### Workflow

A total of eight key features, integrated into a main, were developed and divided as follows:

Matthew Hileman	Sydney Fowler
duplicate_removal.py	analysis.py
file_in.py	cleanup.py
file_out.py	compress.py
main.py	graph.py
	share.py

The remaining files are utilities and supporting files, developed and divided as follows:

Matthew Hileman	Sydney Fowler
excel_funcs.py	custom_dictionaries.py
menus.py	custom_regular_expressions.py
menu_info.py	
quit.py	

This is not to say that each member did not refine and edit each of the files above, rather that they were the lead on that given file.

In order to launch the utility, one will have to execute `main.py`. From there, the user will be directed to their desired tool, whereupon completion of said tool will direct them back to main. The flow of the program is based on the file “`menus.py`,” which utilizes a class to navigate the entire program. When the user reaches a key tool or file, a menu object is created where a description and options are provided based on this menu class.

This project is dependent on the following, which has no check as of now: `openpyxl`, `pandas`, `matplotlib`, `numpy`, `importlib`, and `smtplib`. Ideally, the program would check if these imports are found and ask the user to install if they are not already installed. However, the current implementation requires the user to run `$pip3 install -r requirements.txt` to install the required libraries prior to running the tool.

## **Future Work**

Keeping to the goal of our project, creating a more diverse user interface that allows a user to interact with a UI, or perhaps a Python-based app, would prove beneficial. Further standardizing the code and reducing the number of required imports, or having them bunched into the app without the need to install, would also be beneficial. This project is still in its early stage - the key functionality and tools have been created and loosely integrated with each other but could use refinement. For example, `graph.py` can only handle columns of data that are floats or integers, and it can only print a basic plot. In future, this could be expanded to include pie charts,

bar graphs, etc. that could handle strings, enumerations, etc. In addition, the import/export functions can only handle csv. In future, this could be expanded to delimited text files, JSON files, etc. Lastly, the `duplicate_removal.py` feature could be enhanced through the use of regular expressions, checking things like identical dates in different formats, extra spaces, etc.

## **Conclusion**

The purpose of this project was to create a command-line tool of various functions used to work with Excel files. Developing this project gave the team a lot of practice using the following libraries: `openpyxl`, `pandas`, `matplotlib`, `numpy`, `importlib`, and `smtplib`. It also provided an opportunity to integrate several tools learned in class (regular expressions, working with Excel and CSV, `numpy`, `matplotlib`, sending emails, etc.) into one large project, working them together to create something that would be useful in day-to-day life. Overall, working on the project provided a more wholistic view for the power of Python as a tool for future development work.