Welcome to Outlook Analyzer!

This document assumes that you are interested in learning more about the technical aspects of this Python package. The guide also assumes you have a basic understanding of Python programming.

## Package Dependencies

Outlook Analyzer relies on several packages. Its current version relies on the packages below:

* [**argparse**](https://docs.python.org/3/library/argparse.html)

This package makes it easier to write user-friendly command-line interfaces. Also, automatically generates help and usage message and issues errors when invalid arguments occur.

* [**datetime**](https://docs.python.org/3/library/datetime.html)

This package provides classes in which the program can store and manipulate time-related data.

* [**dateutil**](https://pypi.org/project/python-dateutil/)(“dateutil.relativedelta”)

This package extends the standard datetime package so that the program can represent relative time intervals.

* [**FPDF**](https://pyfpdf.readthedocs.io/en/latest/)

This package is used to generate documentation with python

* [**glob**](https://docs.python.org/3/library/glob.html)

This package is used to extend pathname pattern recognition

* [**matplotlib (“matplotlib.pyplot”)**](https://matplotlib.org/)

This package enables the program to generate analysis plot derived from extracted Outlook information.

* [**os**](https://docs.python.org/3/library/os.html)

This package is a module that provides portable ways of using operating systems dependent functionalities

* [**pandas**](https://pandas.pydata.org/)

This package provides data tables and frames used for file reading, metric organization, and display.

* [**PyPDF2**](https://pypi.org/project/PyPDF2/)

This package is a python PDF library that can split, merge, crop, and transform pages of PDF files.

* [**pywin32 (“win32com.client”)**](https://pypi.org/project/pywin32/)

This package allows the program access to standard Windows APIs, enabling features like email data extraction from the Windows Outlook desktop client.

* [**re**](https://docs.python.org/3/library/re.html)

This package simplifies the process of identifying character patterns during word cloud generation, specifically when removing links from email text bodies.

* [**shutil**](https://docs.python.org/3/library/shutil.html)

This package offers high-level operations on files and collections of files

* [**subprocess**](https://docs.python.org/3/library/subprocess.html)

This package allows spawning new process, connect the processes input/output/error pipes, and helps obtain their return codes

* [**sys**](https://docs.python.org/3/library/sys.html)

This package enables the program to identify and work with Python interpreter-specific variables like function names.

* [**tabulate**](https://pypi.org/project/tabulate/)

This package enables the program to easily print tables as outputs.

* [**time**](https://docs.python.org/3/library/time.html)

This package provides time-related functions and formatting.

* [**tqdm**](https://pypi.org/project/tqdm/)

This package provides the program’s progress bars as displayed when it is extracting email and task information.

* [**traceback**](https://docs.python.org/3/library/traceback.html)

This package provides a standard interface to extract, format, and print stack traces of Python programs.

* [**wordcloud**](https://pypi.org/project/wordcloud/)This package provides the ability to generate wordclouds from email body texts.

## Function Summaries

**append\_to\_error\_list(function\_name, error\_text, optarg = None)**  
Generates a list of errors that have occurred during program execution which are printed at end of run.  
  
**build\_text\_with\_subject\_senderemail\_receivedtime(messages\_list, email\_data\_file)**  
 Reformats item text data to UTF-8   
  
**category\_data\_gen(category\_list,categories\_data\_file)**  
 Generates category data by using the 'category\_list', runs it through the 'unique' function, saves results into a dict, and then prints it on a text file with the variable 'categories\_data\_file'   
  
**convert\_csv\_to\_df\_to\_figure\_to\_pdf(email\_data\_file,title\_str,columns\_list,pdf\_file\_name)**  
 Converts a csv into a data frame and then to a figure and exports it as a single pdf   
  
**convert\_dict\_to\_df\_to\_figure\_to\_pdf(metric\_dict, title\_str, columns\_list, pdf\_file\_name)**  
 Converts a dict into a data frame and then to a figure and exports it as a single pdf   
  
**convert\_time\_range\_to\_date(date\_range)**  
 Convert string value such as 12m (ago) or 10d (ago) into an actual date using relativedelta or timedelta   
  
**create\_pdf\_cover\_page(message\_counter\_int,message\_unread\_counter\_int)**  
 Add images into a PDF file   
  
**def cleanup(inp):**  
 Removes left-to-right embedding characters ("\u202a") and pop directional formatting characters ("\u202c") from strings  
  
**delete\_temp\_files(type\_list)**  
 Goes through directory and removes/cleans the files with specified extensions (i.e .txt, .tmp, .png, etc.)

**extract\_outlook\_information(max\_email\_number\_to\_extract\_input,start\_date,end\_date)**Connects to Outlook client and iterates through items. Collects relavent information from Outlook desktop client.  
  
**generate\_unread\_senders\_viz()**  
 Creates a dataframe from unread sender information and generates a locally saved plot.

**generate\_word\_cloud\_viz()**Creates wordcloud from email body text and saves as local image.  
  
**is\_integer\_num(n)**Checks that input is of integer type  
  
**def main(argv)** Runs through the specified inputs that users will enter to get their analysis data for Outlook

**pdf\_merge(open\_file,output\_file\_name)**  
 Merges all the pdf files in current directory   
 **return\_sender(outlook\_object)** Returns sender email address  **unique (list1)** Identifies unique elements within a list   
 **unread\_senders\_data\_gen(unread\_senders\_raw\_list,unread\_senders\_unique\_dict,sender\_data\_file)** Creates a sorted dictionary from extracted unread senders data

**word\_cloud\_content\_clean()**  
"Creates a .txt file consisting of email body text with hyperlink information removed. Hyperlinks identified with regex indexes matches for '<http', <mail', and '>' within original text body.  
  
**word\_cloud\_extract(messages)**  
Extracts word cloud information from most recent 50 messages.

## Understanding Script Variable Naming Standards

* Camel case is used when required for MS object model variables
* Uppercase is used for global variables (e.g. OUTPUT\_NAME\_STR)
* Lower case is used with an underscore between words for variables used in the script and within functions (variable\_name\_str)
* In general, data type is appended to the end of the variable (dict, str, int, etc) but may not be applied to all variables.
* Indentations are handled with four spaces