**Avaya Call Volume Report Program**

**Script Documentation**

* 1. Program Description:
* The program semi-automatically generates the Avaya call volume report after the raw data from Avaya IPOCC is exported.
* Program created and ran in Python 3.10.4 conda env
  1. Libraries used:
* Openpyxl (Editing excel workbooks)
* Tkinter (To create user GUIs)
* Pywin32 == 225 (Latest versions caused errors)
* Pyinstaller (To create the executable file)

**2.1** Limitations of the program:

* Raw data from Avaya IPOCC has to be exported before the program is able to run.
* The program is unable to directly edit files inside OneDrive.
* The program has to be separated into 2 parts as a library to edit Binary Excel files could not be found.
* After generating the raw report, the user still has to manually edit slicers and copy the data on to the template.
* The daily email for the report is not automatically sent.

**2.2** Possible program improvements:

* Automatically exporting the raw data from Avaya IPOCC.
* Instead of separating the program into 2 parts, combine them and only have 1 executable file. This would require a way to edit the Binary Excel file, which includes a way to edit pivot table slicer values as well as copying specific columns of data from one workbook to another. Openpyxl library does not support editing of Binary Excel files.
* Allow authorisation to Openpyxl to edit files inside OneDrive so the user does not need to manually copy and paste the completed report files into OneDrive.
* Automatically copy and finalised report table and send the email.

**3.1** Files and Functions:

* Generate Avaya Report:
  1. Main.py:
     + Contains all the main function calls and runs the entire program.
  2. askDir.py:
     + Creates a Tkinter GUI to ask the user for a file directory location.
  3. getFiles.py:
     + Calls askDir() function to get file path and checks if the required files needed to run the excel macro are present.
     + If the files are present the program returns them as a list, if not the program creates a Tkinter GUI informing the user that not all files are present or they are invalid.
  4. editMacro.py:
     + Accesses the macro file, changes the file names to the latest version and saves the workbook.
     + It also checks for a permission error in case the user is trying to run the program inside of OneDrive or any unauthorised location.
  5. runMacro.py:
     + Runs the Open\_Avaya\_Report macro.
* Generate Email Report
  1. Main.py:
     + Contains all the main function calls and runs the entire program.
  2. askDir.py:
     + Creates a Tkinter GUI to ask the user for a file directory location.
  3. copyRange.py:
     + Copies a selected range in an excel workbook.
  4. pasteRange.py:
     + Pastes the copied data in a specified location in the workbook.
  5. generateTable.py:
     + Loads the template avaya file and deletes unwanted data, as well as, copies the needed data into the template.
     + It also checks for a Permission Error in case the user runs the program inside of OneDrive or any unauthorised location.