# Installation

1. Download and install Anaconda for Windows Python 2.7 https://www.continuum.io/downloads#\_windows
2. Download and install Matlab 2014b or later - https://www.mathworks.com/index.html?s\_tid=gn\_loc\_drop

**After Matlab is installed, install MATLAB Engine for Python:**

**cd "matlabroot\extern\engines\python"**

**python setup.py install**

1. Download and install docxtpl 0.1.9 or later - https://pypi.python.org/pypi/docxtpl/0.1.9

Typically this can be done by running cmd.exe as administrator and entering this command into the prompt: **C:\>pip install docxtpl**

\**If it cannot be done by using pip, it has to be manually installed by using the link above and running the setup.py in that package*

# report\_generator.py

report\_generator.py, by default, expects all of the following to live in the same folder:

1. info.csv
2. template.docx
3. guide.csv (optional)
4. touchstone files (\*.sNp)

These files do not *have* to live in the same folder, but if they do not, then you need to pass parameters in via command line switches to point the program to the correct directories. You can see the optional switches by passing the parameter “help” in the following way: **python report\_generator.py help**

**For basic usage,**

1. **Put the above files in the same directory as report\_generator.py and guide\_UI.py**
2. **Run guide\_UI.py to generate a guide.csv file**
3. **(optional) edit the guide.csv file if needed**
4. **Run report\_generator.py**

# info.csv and template.docx

info.csv has two parts. The first part is a dictionary. For the dictionary, any key term (column 1) found in the template.docx will be replaced by the value (column 2) of the same row. In template.docx the convention for a key is this: {{ key }}, so wherever that tag is found, it will be replaced by the associated value.

The second part of info.csv has two tables. These tables are sandwiched by tags <DUT> table1 </DUT> and <PORT> table2 </PORT>. Everything between these tags is pasted into the template.docx where {{ DUTIdentification }} and the {{ PortConfiguration }} are found.

template.docx should be somewhat self explanatory but anywhere {{ tags }} are found, information will be replaced with data and plots gathered from info.csv and the .sNp files

# guide.csv

The key to this program is the **guide.csv** file. This file contains instructions for how (and which) plots will come together in the final report. **report\_generator.py** interacts with the **guide.csv** by reading from it, or generating one if one does not exist yet.

s1\_1 🡨this is the convention for specifying s-parameters. The underbar helps the program weed out ambiguity from double digit multiport parameters such as s11\_3. Technically you can put in s11 or s12 but it is advised to use the underbar for clarity.

The program assumes guide.csv to live in the same directory as report\_generator.py, but you can specify a particular location using the ‘g’ switch command as follows:

C:\>python report\_generator.py g c:\path\to\directory\where\guide\is

**guide.csv** is generated/edited in one of the following ways:

1. If the guide.csv file **is** found, when report\_generator.pt is run, no matter what commands are passed, guide.csv will be read in, and used to generate the plots. Guide.csv is always given the highest priority. Only if it is absent are other commands used. This brings us to…
2. If **no** guide.csv file is found, and **no** ‘p’ command is specified via terminal then:

A default guide.csv is generated using these four plots:

**"s1\_1m,s1\_1p,s2\_1m,s2\_1p"**

1. If **no** guide.csv is found, and a p switch command is specified such as:

**C:\>python report\_generator.py p s1\_1m,s1\_2m**

Then a guide.csv will be generated of those specified plots

1. There is a guide\_UI.py tool to assist in generating a guide.csv
2. You can manually open up the guide.csv and edit it directly.

Here is an example of a valid guide.csv file:

