

SM_Visualizer.py

```
#####  
# Name of script:      sm_visualizer.py                                #  
# Editor:             Patrick Barrett                                #  
# Last update:        May 1 , 2020                                  #  
# Description :       This script reads a series of csv files containing soil moisture data and      #  
#                     manipulates, performs quality control operations and visualize these data sets  #  
#####
```

Table of Contents

What is sm_visualizer?	1
What version of python and what python library dependencies does sm_visualizer required?.....	1
How should the input files be formatted?	2
How should missing values be recorded within the input .csv files?	2
What if my input data is recorded at hourly, or daily time intervals?.....	2
What is the largest time interval for input data required for sm_visualizer?.....	2
What quality control techniques does sm_visualizer perform on input data?	3
What output files does sm_visualizer produce?.....	3
What steps should I take before executing sm_visualizer?	3
<i>What user information does sm_visualizer require?.....</i>	<i>3</i>
How do I execute sm_visualizer?.....	4

What is sm_visualizer?

sm_visualizer is a python script designed to ingest multiple climatic soil moisture files from different monitoring stations in .csv format, perform simple quality control techniques on each file and export several .csv data files and multiple pane .png images of these .csv files

What version of python and what python library dependencies does sm_visualizer required?

sm_visualizer was built using python version 3.6 and requires the following python libraries:

- numpy
- math
- csv
- os
- pandas
- datetime
- matplotlib
- sys

How should the input files be formatted?

The input file for each station should be a .csv file with 8 columns as shown below.

	A	B	C	D	E	F	G	H	I
1	datetime	year	month	day	hour	minute	soil_moisture	corrected_soil_moisture	
2	18/10/2015 21:28	2015	10	18	21	28	-9998	-9999	
3	18/10/2015 21:29	2015	10	18	21	29	-9998	-9999	
4	18/10/2015 21:30	2015	10	18	21	30	-9998	-9999	
5	18/10/2015 21:31	2015	10	18	21	31	-9998	-9999	
6	18/10/2015 21:32	2015	10	18	21	32	-9998	-9999	
7	18/10/2015 21:33	2015	10	18	21	33	-9998	-9999	
8	18/10/2015 21:34	2015	10	18	21	34	-9998	-9999	
9	18/10/2015 21:35	2015	10	18	21	35	-9998	-9999	
10	18/10/2015 21:36	2015	10	18	21	36	-9998	-9999	
11	18/10/2015 21:37	2015	10	18	21	37	-9998	-9999	
12	18/10/2015 21:38	2015	10	18	21	38	-9998	-9999	
13	18/10/2015 21:39	2015	10	18	21	39	-9998	-9999	
14	18/10/2015 21:40	2015	10	18	21	40	-9998	-9999	
15	18/10/2015 21:41	2015	10	18	21	41	-9998	-9999	
16	18/10/2015 21:42	2015	10	18	21	42	-9998	-9999	
17	18/10/2015 21:43	2015	10	18	21	43	-9998	-9999	
18	18/10/2015 21:44	2015	10	18	21	44	-9998	-9999	
19	18/10/2015 21:45	2015	10	18	21	45	-9998	-9999	

Note: The first column is not actually used by sm_visualizer, so it can be blank, but the column space is required, i.e. sm_visualizer takes the second column to be the “year” column and so on.

How should missing values be recorded within the input .csv files?

sm_visualizer takes “NA”, “Nan”, -9998 and -9999 as missing values for soil moisture, however the time columns (i.e. year, ..., minute) should always contain their appropriate values.

What if my input data is recorded at hourly, or daily time intervals?

In the event that the input data is being recorded at hourly, or daily time intervals, simply represent the remaining columns by 0, indicating either no hours or no minutes or both.

What is the largest time interval for input data required for sm_visualizer?

sm_visualizer calculates weekly (7-day) and bi-weekly (15-days) running averages of soil moisture; hence the recommended largest time interval is weekly.

Note: In the event that a weekly interval is used the QC data files should be the same as the 7-day-mean files.

What quality control techniques does sm_visualizer perform on input data?

sm_visualizer applies a series of quality check techniques on the soil moisture values within each input data file. Let x be an entry within the corrected soil moisture column. The following QC techniques are applied to the soil moisture data.

- Unphysical values – all values that fall outside the range $0 \leq x \leq 100$ are removed as unphysical values
- Outliers – all values that fall out of the range $Q_1 - 1.5 \times IQR \leq x \leq Q_3 + 1.5 \times IQR$ are removed as outliers.
- Proportion of missing data – for the 7-day and 15-day averages, if more than 10% of the data is missing, that week is removed.

What output files does sm_visualizer produce?

sm_visualizer produces the following output files

1. Original input data (.csv format) and multiple pane plots of the original data files (.png format) from each climate station.
2. QC data files (.csv format), resulting from applying simple quality control techniques to original data files, and multiple pane plots of the QC data files (.png format) from each climate station.
3. 7-day running averages (.csv format) and multiple pane plots of 7-day running averages (.png format) using the QC data file for each climate station.
4. 15-day running averages (.csv format) and multiple pane plots of 15-day running averages (.png format) using the QC data file for each climate station.

What steps should I take before executing sm_visualizer?

A successful run of sm_visualizer requires the following set up:

1. Create a folder entitled [SoilMoisture](#) and place the sm_visualizer script within the folder.
2. Place the input .csv soil moisture climate files for the different stations within the [SoilMoisture](#) folder.
3. Open the sm_visualizer script in an appropriate editor and edit the user information.
4. Execute sm_visualizer

What user information does sm_visualizer require?

sm_visualizer requires the user to enter information for 6 separate variables, as shown in the image below

```
##### User Inputs #####
```

```
folder_location = '/Users/Dessyb/Dropbox/PPCR/SoilMoisture/' #name of the folder containing the csv files
stations = ['ATG_Cardi', 'ATG_Christian', 'BAR_Canefield_Nursery_A', 'BAR_Husbands_A', 'DMA_Pondcasse', 'D
csv_locations = ['ATG_Cardi_17061771.csv', 'ATG_Christian_17061854.csv', 'BAR_Canefield_Nursery_(A)_131
columns = 2
rows = 4
resolution = 600
```

- `folder_location` is the path of the `SoilMoisture` folder.
- `stations` is a list of the various climate stations, whose .csv files are already within the `SoilMoisture` folder, that `sm_visualizer` should operate on.
- `csv_locations` is a corresponding array to `stations` that has name of the .csv file for each station.

Note: It is important that the order of the station names in the `stations` array and the order of the .csv files for each station in the exact order.

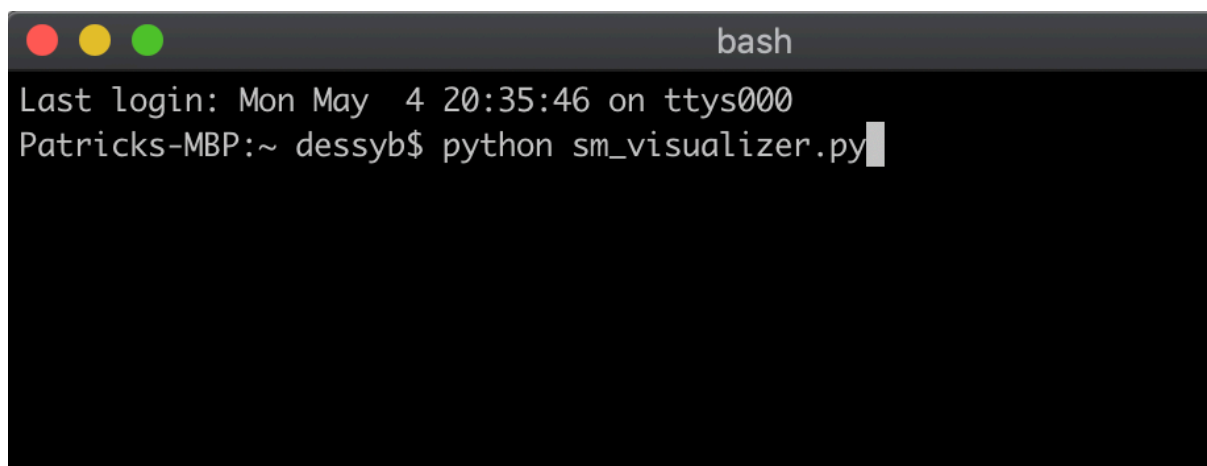
- `columns` and `rows` are the dimensions for the multiple pane .png plots created by `sm_visualizer`.
- `resolution` alters the resolution of the .png files created by `sm_visualizer`.

Note: While increasing the resolution renders clearer images, the script will require more time to execute.

How do I execute `sm_visualizer`?

You can run `sm_visualizer` from any python environment. Every python installation comes with an Integrated Development and Learning Environment (IDLE). Other common environments include Vim, Emacs, Textmate, Eclipse with PyDev plugin and Wing IDE.

Additionally to run `sm_visualizer` from either the windows command line or mac os terminal simply change directory to the **SoilMoisture** folder and run the command “python `sm_visualizer.py`”.

A screenshot of a terminal window with a dark background. The title bar shows three colored circles (red, yellow, green) and the text 'bash'. The terminal content shows 'Last login: Mon May 4 20:35:46 on ttys000' followed by the prompt 'Patricks-MBP:~ dessyb\$' and the command 'python sm_visualizer.py' with a cursor at the end.