# Python code raingauge\_click\_plots

A new code has been written using Python to enable raingauge quality control. It uses a similar process to the previous Matlab code (plot\_raingauge) but there are some improvements and also some changes which were required due to differences in functionality between Python and Matlab.

The new code still operates in a similar way to previously, in that when erroneous points are highlighted in the window, the corresponding corrections file (*instrument\_id*.corr) in /data/netCDF/corrections is edited to include the time window of that correction. These corrections apply to both the original format cfarr\*.nc netCDF files and the new ncas\*.nc files. The way in which the corrections are applied is slightly different, but the same need to identify false points still exists.

This correction process identifies data points which are suspicious, via the use of the HOLDCAL label in the corrections file. These are usually rain counts which were recorded when there is no evidence of rain being present. In the old cfarr\*.nc netCDF files, these were set to a value of zero over the identified time period. In the new ncas\*.nc files the data value is not changed, but a flag is set to indicate that the value is suspicious if it is above zero.

This code does not handle cases where there is known to be a fault with the gauge, such as a blockage or failure. These cases currently have to be manually entered in the correction file as BADDATA. They are then shown as a fill value in the netCDF file.

## Procedure

Text shown in bold is a command to be typed. Exceed needs to be running on your computer.

1. Start Exceed on your Windows computer if it hasn’t been started automatically.
2. Use PuTTY to start a terminal window connected to Wilma.
3. Log in to Wilma using usual username and password.
4. **ssh python**

This transfers you from Wilma to the “python” computer.

1. Enter your password again
2. Enter the command **conda activate chil\_3\_8**
3. Connect to the directory where the code is located using

**cd /home/chilbolton\_software/python/ncas\_python/raingauge**

1. Enter **./raingauge\_click\_plots.py -s *yyyymmdd***

Substitute the date you want to process for *yyyymmdd* e.g. 20210425

1. After a few seconds a plot screen will appear. If it doesn’t automatically appear as the foreground window you will need to select the Exceed icon in the taskbar to see it.
2. **Click on each plot to either side of any data points that you want to remove**. The pointer stays as a rather clunky pointer rather than neater crosshairs, but position it to either side of the point as best you can.
3. Unlike with the Matlab version, don’t press Return when moving on from one plot to the next. This code recognises which plot you have clicked on by its position in the window.
4. Proceed through the plots along the top row, then the bottom row. If you realise you missed something, don’t go back. Finish this procedure then start again from command , just correcting the missed point.
5. When you have reached the end of the series of plots, click the **Close (X) button** of the window to finish. At this point the corrections get written to the relevant corrections files, provided no errors were detected.
6. The code will spot if there have been a total of an odd number of clicks on plots, or if there are any where the 2nd point of a pair is earlier than the first. If this happens the program will end and no corrections will be written. An error message which begins “Error in clicking plots” will be displayed in the text window where you enter commands.
7. If no errors in the corrections were detected, after a few seconds a window will appear which shows which data were removed by outlining that region in red. Once you have checked whether you are happy with the corrections, click the **Close(X) button** and the program will end. The message “Corrections successfully written” will be shown in the text window where you enter commands.
8. You should have a command prompt at the start of the line, as at the start. If not, there may still be a plot window which needs to be closed. Return to command to process other days.
9. When you have finished work, enter **exit** to leave the Python machine, then **exit** again to leave Wilma. Or you can stay on the Python machine to run other programs.