**The Programming Logic that is applied to filter out "bad" data (including outliers) from any Probe dataset.**

***Date of the first draft of this document:*** *2019/02/11*

*ProbeSchedule & MatogenAI*

***Members Involved:*** Jac *Le Roux*, Jacobus *Eksteen*, André *Heydenrych*, Wim *Conradie* & Henri *Branken*

[**Brief overview of data:**](#_6il9pxbysz1p) **1**

[**A (pedantic) question concerning Reference Evapo-transpiration:**](#_30j0zll) **2**

[**"rzm\_source" data column: "software" versus "Electronic Probe"**](#_1fob9te) **2**

[**"total\_irrig" data column.**](#_3znysh7) **2**

[**"profile" data column.**](#_2et92p0) **3**

[**"heat\_units" data column**](#_tyjcwt) **3**

[**"etc", "et0", and "etcp" data columns**](#_3dy6vkm) **4**

[A detailed sub-section on the "etcp" data column](#_1t3h5sf) 5

[**How must we flag events of drought-stress?**](#_2s8eyo1) **6**

[**Preliminary figure of calculated kcp values (Probe P-371, Kromfontein):**](#_17dp8vu) **7**

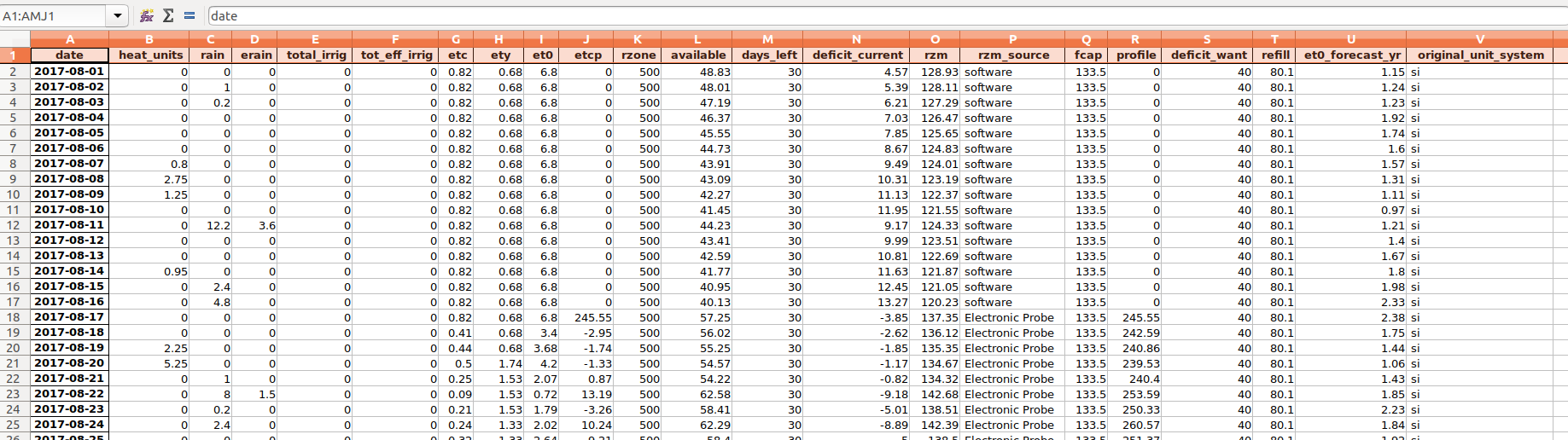
[**How to flag "rain" events?**](#_3rdcrjn) **7**

[**Question on comparing (simulated/theoretical) kc with (emperical/calculated) kcp.**](#_26in1rg) **7**

[**Any other Questions/Comments/Suggestions/Topics for discussion:**](#_lnxbz9) **8**

# Brief overview of data:

Below is a snapshot of an **Excel spreadsheet** containing our data:



Our **column headings** are as follows: ['heat\_units', 'rain', 'erain', 'total\_irrig', 'tot\_eff\_irrig', 'etc', 'ety', 'et0', 'etcp', 'rzone', 'available', 'days\_left', 'deficit\_current', 'rzm', 'rzm\_source', 'fcap', 'profile', 'deficit\_want', 'refill', 'et0\_forecast\_yr', 'original\_unit\_system']

The following data columns are **not of interest** to us for our analysis:

['rzone', 'available', 'days\_left', 'deficit\_current', 'rzm', 'fcap', 'deficit\_want', 'refill', 'et0\_forecast\_yr']

Consequently, we *drop* these columns from our dataset and work with the remaining data columns:

**['heat\_units', 'rain', 'erain', 'total\_irrig', 'total\_eff\_irrig', 'etc', 'ety', 'et0', 'etcp', 'rzm\_source', 'profile', 'original\_units\_system']**

# A (pedantic) question concerning Reference Evapo-transpiration:

What is the scientifically correct subscript for the reference evapotranspiration? Is it the letter o, or is it the number 0?

According to the column headings of the data extracted from the API, it is the number 0. But according to certain literature sources and our discussions, it is the letter o. Hence, there exists a little bit of confusion...

# "rzm\_source" data column: "software" versus "Electronic Probe"

Basically, we flag the data entries for which the column "rzm\_source" contains the entry "software". We do not want to build our model from simulated data, but rather from actual probe readings.

# "total\_irrig" data column.

We need to flag data entries corresponding to irrigation events because it distorts our "Profile" and "etcp" waterbalance readings. This is somewhat complicated by the possibility of a farmer logging an irrigation event on the wrong date.

A suggestion on how to **accurately** flag irrigation events is as follows: Flag a data entry:

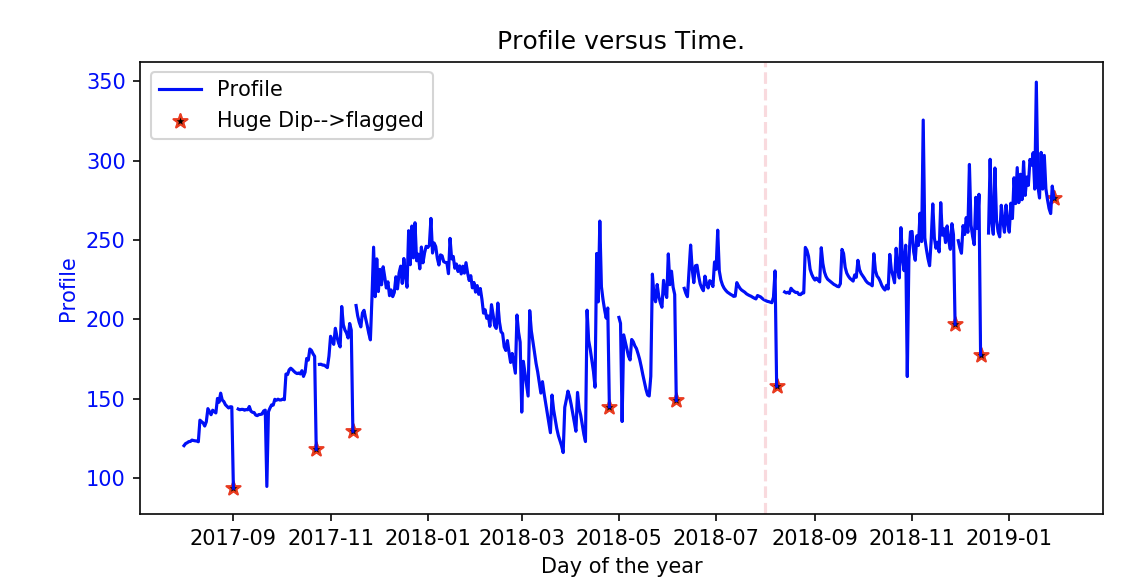
1. If the farmer logged irrigation taking place for that day, **AND**
2. If the corresponding etcp > 0, **AND**
3. If there is no rain for that day.

Jac, are you comfortable with the above suggestion to flag irrigation events?

# "profile" data column.

In the "profile" column, there are certain entries containing 0.0; these entries correspond to missing data. For these missing "profile" entries, we replace the 0.0's with NaN's (Not a Number). We also flag these missing "profile" entries.

Upon closer inspection of the graph of Profile versus Time, we get the following:

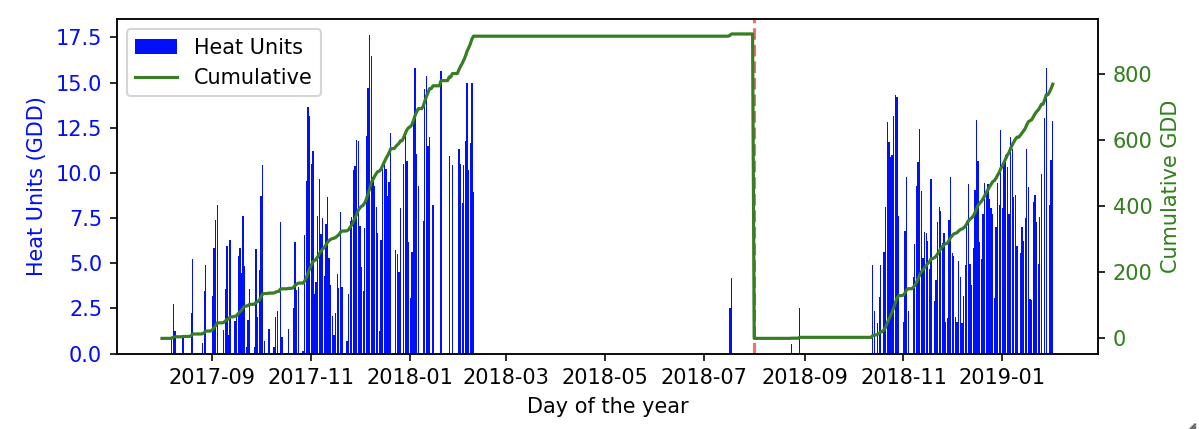


Notice that for the dates adjacent to missing data gaps, there is always a strange dip in the "profile" value. These large dips are indicated by the red stars in the above plot. The data entries associated with these huge dips are also flagged. What is the reason for these particular readings dipping so low? These large dips imply that the apple tree had a massive water uptake via absorption through its roots, but physiologically this is not possible.

What's still bothersome is the datapoints indicated by the green arrows. We find it a little suspicious that that "profile" readings dip so low when compared to the neighbouring/adjacent data points. Should we refine our logic to also flag suspicious "profile" readings such as these?

# "heat\_units" data column

There are many dates for which the value of "heat\_units" is jammed and repeats for a long time interval. This is due to faulty weatherstation data. Entries for which "heat\_units" values repeat are flagged, and the actual "heat\_units" values are replaced with 0.



As can be seen in the above plot, there are many gaps due to faulty weather station data. Therefore the green curve representing **cumulative** growing-degree-days is also affected and not accurate.

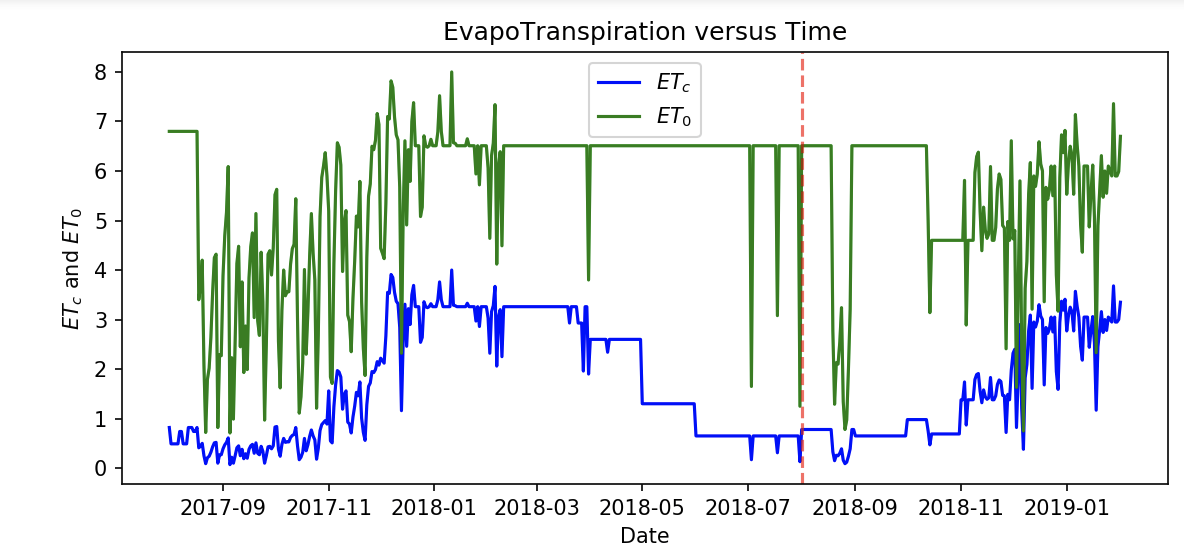
Jac, we would like to know at what value of cumulative GDD (after beginning August) does *transpiration* come into play for (golden delicious) apple trees?

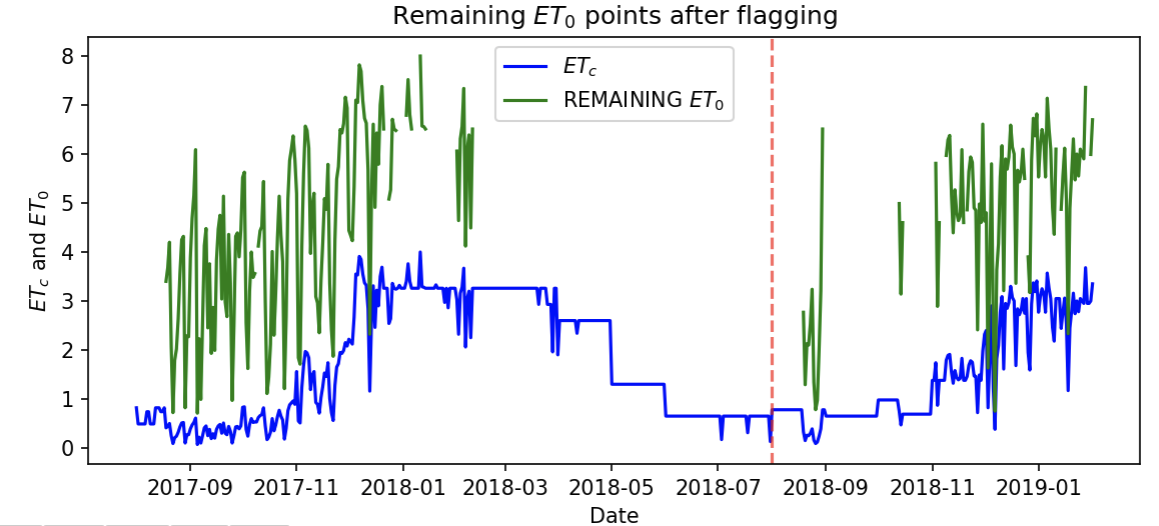
# "etc", "et0", and "etcp" data columns

There are time intervals for which "et0" (and also "heat\_units") are stuck: they repeat identical values for a long duration of time. Obviously, it is not possible to get such repetitive values when considering the fact that weather data is very random. The data entries associated with these repeating values are flagged.

It is rather unfortunate that a relatively large volume of data is lost when flagging these "stuck" values. A loss of up to 54% of the data is seen for the *Koue Bokkeveld* probes.

Below are two figures comparing a dataset *before flagging bad "et0" values*, and *after flagging bad "et0" values*:





#### A detailed sub-section on the "etcp" data column

"etcp" is defined as the difference between consecutive "profile" readings:

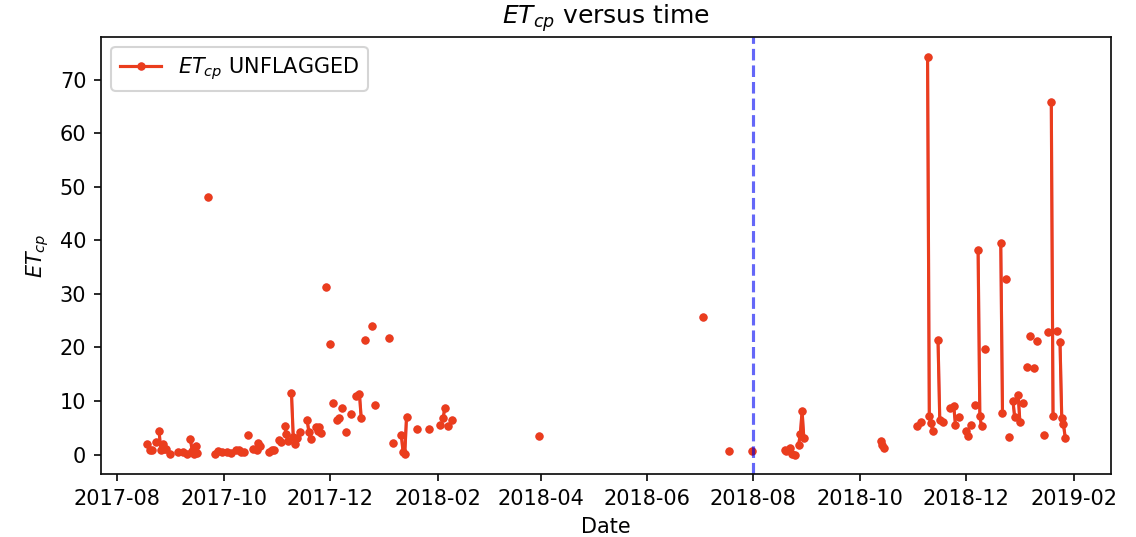
etcp(t) = profile(t) - profile(t-1)

We are mostly interested in the etcp entries for which etcp < 0. These negative entries reflect incidents in which water was lost from the soil due to: (1) Water Drainage, (2) Luxurious water uptake, (3) Normal water uptake, ((4) and Drought-stress maybe?)

We are not interested in the dates in which etcp was distorted by irrigation and/or rain. For such dates we expect that etcp >= 0. Therefore, data entries corresponding to etcp >= 0 are flagged. Furthermore, all etcp >= 0 entries are set to NaN values (Not a Number).

At this stage, we are now only left with the etcp < 0 entries (i.e. entries associated with water drainage, luxurious/normal water uptake, and drought-stress). For simplicity, we multiply these remaining etcp values with -1 so that henceforth we only work with positive values of etcp (which is a little bit more convenient for programming purposes).

As can be seen in the figure below, there are some outliers still present in the remaining etcp dataset.



These large outliers are most likely associated with phases of water-drainage and luxurious water uptake. We, on the other hand, are only interested in phases corresponding to **normal water uptake**.

From an educated guess, let us accept a maximum eto of 12.0 mm. Let us make another educated guess and allow for a maximum kcp value of 0.8 (of course, these educated guesses vary from cultivar to cultivar). This implies that the maximum allowed value for etcp is as follows:

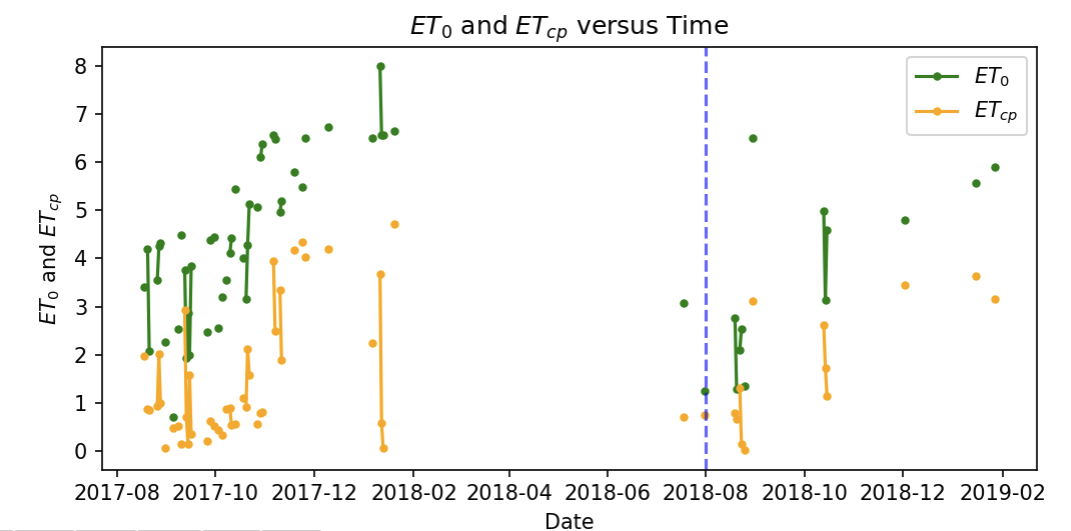
max(etcp) = 0.8 x 12 = 9.6, which is approximately **10 mm** (for Golden Delicious Apples)

Consequently, we flag all etcp > 10 mm data entries.

Jac, are you comfortable with the calculation of max(etcp) and consequent flagging for entries satisfying etcp > max(etcp)?

But remember, we only tolerate etcp values for which kcp <= 0.8. Therefore, to ensure that **all** luxurious water-uptake phases are completely flagged, we perform another flagging operation that flags data entries for which etcp > 0.8 x et0 (for extra insurance).

In the figure below, we see an example of the remaining etcp values together with their corresponding eto values.



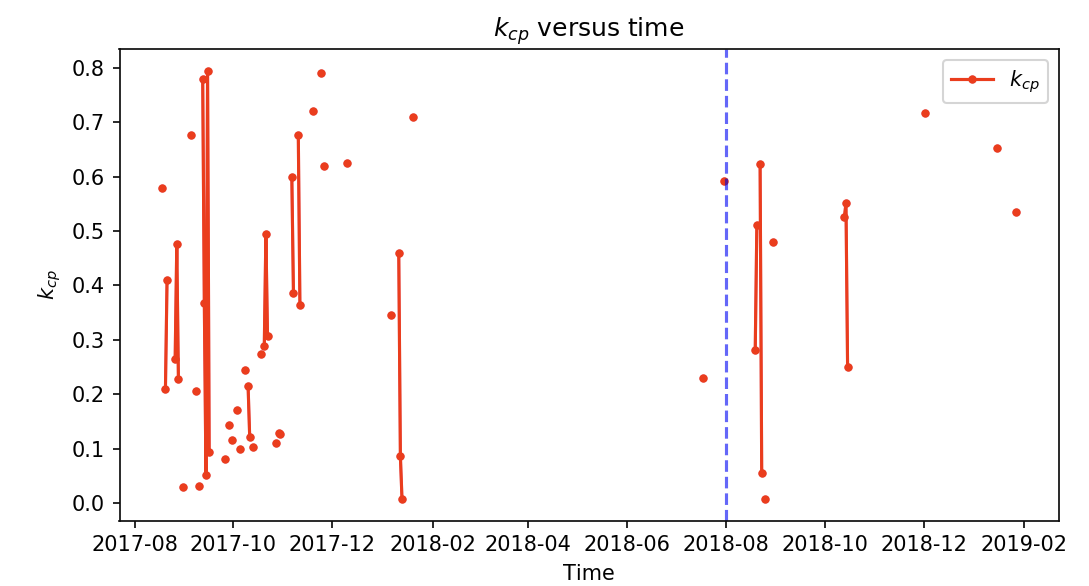
# How must we flag events of drought-stress?

One line of reasoning is that during phases of drough-stress we expect very small changes in the waterbalance "Profile" readings. According to this logic, we expect for drought-stress that:

profile(t) – profile(t – 1) = etcp(t) <= *e*

where *e* is a relatively small value, such as, for example, *e* = 0.1. Jac, what do you think is a realistic value of *e* that corresponds to drought-stress? Or, would you suggest another method (different from the above reasoning) that is more effective for flagging periods of drought-stress?

# Preliminary figure of calculated kcp values (Probe P-371, *Kromfontein*):



Of course, we still need to further refine our logic on flagging "bad" data. The above figure is an indication of our current progress as of this date (12 February 2019). Jac, do you maybe have any comments on this preliminary figure?

# How to flag "rain" events?

If memory serves us correctly, Jac suggested that we should flag events in which "rain" > 2 mm. Jac, do you find this to be correct, or is it a crude approximation?

Another question is as follows: Wouldn't it be more efficient to only concentrate on the "etcp" data column. If rain, on any particular day, does indeed have a significant effect, then we would expect that the soil moisture content would increase. In short, we would expect that etcp > 0. However, as discussed previously, we already implement flagging for etcp > 0 entries. Therefore, doesn't the etcp > 0 flagging implicitly also take care of flagging rain incidents that distort our waterbalance readings? In short, isn't etcp > 0 flagging sufficient for accounting for (significant) rain events?

**If** it is indeed true that flagging etcp > 0 entries is sufficient, then we can simply ignore/drop the "rain" and "erain" columns.

(Co-incidentally, can't the very same logic also be used to address the problem of how to accurately flag irrigation events?)

# Question on comparing (simulated/theoretical) kc with (emperical/calculated) kcp.

Can André maybe update the API such that we also have a column of the simulated/theoretical kc values with which we can compare our calculated/emperical kcp values?

If we remember correctly, the simulated kc values are only calculated on a monthly basis? Jac, is this correct?

# Any other Questions/Comments/Suggestions/Topics for discussion:

(**A suggested rule of thumb:** Let us first ensure that the "basic" flagging is thoroughly implemented, and then **gradually** build our way up to more sophisticated methods of flagging and refining our model.)

<Add remaining questions/comments for discussion here>

...

...

...

...

...

...