
SOFTWARE METAPAPER

SWCalibrateR: Interactive, Web – Based Calibration of Soil Moisture Sensors

Johannes Brenner¹, Giulio Genova^{2,3}, Giacomo Bertoldi², Georg Niedrist² and Stefano Della Chiesa²

¹ Helmholtz-Centre for Environmental Research – UFZ, Department Computational Hydrosystems, Leipzig, DE

² Eurac research, Institute for Alpine Environment, Bolzano/Bozen, IT

³ Faculty of Science and Technology, Free University of Bolzano/Bozen, IT

Corresponding author: Johannes Brenner (johannes.brenner@ufz.de)

SWCalibrateR is a user-friendly web application. We designed SWCalibrateR to interactively estimate linear regression relationships of any couple of field data series. We specifically developed this toolbox to calibrate soil moisture sensors based on gravimetric soil moisture samples. The application has been implemented using R-shiny (<https://shiny.rstudio.com/>).

As a user you can upload your own dataset and dynamically filter it by categories like soil type, land use, soil depth and others. With SWCalibrateR you can visualise the filtered data scatter and the estimated linear model. You can diagnose your model estimate and thus easily remove outliers influencing your model estimate. SWCalibrateR handles robust estimates of linear models besides ordinary least square estimation. Additional features are an interactive data table view and mapping of the data points.

Keywords: R; Shiny; Soil moisture; Calibration; TDR

Funding statement: This work was supported by the farming consulting centre for fruticulture and viticulture “Südtiroler Beratungsring” and the research grant “MONALISA” of the Provincia Autonoma di Bolzano, Alto Adige, Ripartizione Diritto allo studio, Università e ricerca scientifica.

(1) Overview

Introduction

In environmental research we often need to estimate a linear regression relationship for multi-variable field data series, linked to several attributes. With SWCalibrateR we provide a user-friendly web application to facilitate this task specifically for calibration of soil moisture sensors.

Studies on plant-soil-water interactions for agronomical, as well as for hydrological application, frequently require accurate measurements of soil moisture. Soil moisture varies largely in time and space and is commonly determined by continuous measurement of volumetric soil water content (VWC) by means of time domain reflectometry (TDR). TDR is a non-destructive method to retrieve VWC using a single generalized calibration equation [6], which relates soil dielectric constant and VWC. Even though it is a widely accepted measurement in most mineral soils, to account for soil properties it is generally necessary to conduct further field specific calibration. Field calibration is conducted by gravimetric calibration, which is a precise but destructive method that involves taking a soil sample, weighting, oven drying, and reweighting it. By considering also the soil bulk density, soil moisture can be expressed as VWC. Comparing VWC from TDR and gravimetric measurements finally leads to a soil-specific

calibration equation. SWCalibrateR is an open-source web application, which provides calibration equations for soil moisture sensors. In developing SWCalibrateR we aimed to provide a tool, which interactively allows retrieving a calibration equation, based on user defined requirements.

The application’s current features allow the user to retrieve calibration equations interactively by filtering the data and deciding whether to use ordinary least square (OLS) or the “maximum likelihood type” MM-type estimator. This means, when several soil moisture sensors are installed in heterogeneous soils and locations, SWCalibrateR permits one to find the best calibration equation by grouping the data according to user defined criteria. Furthermore, the user can assess model diagnostic and remove leverage and outlier points if needed. Additional features are an interactive data table view and mapping of the data points.

Implementation and architecture

SWCalibrateR is built using the R-Shiny framework [1] for developing web applications. The structure of a shiny app consists of two components: (1) a user-interface (UI) script that controls the layout and appearance of the app and (2) a server script that contains instructions for the computer to build the app. We implemented

the application in the common R-package structure. The SWCalibrateR package is hosted in GitHub (<https://github.com/JBrenn/SWCalibrateR>) and can be downloaded or cloned for further development from there.

```
> git clone git@github.com:JBrenn/SWCalibrateR.git
# use SSH
> git clone https://github.com/JBrenn/SWCalibrateR.git
# use HTTPS
```

For easy access to the R project open the SWCalibrateR.Rproj file. Within the project, the developer can now discover and adapt the source code to his needs. Pull requests to the GitHub repository are very welcome. Please feel free to report bugs to the GitHub issues: <https://github.com/JBrenn/SWCalibrateR/issues>.

As a developer be aware that the application, besides R (>=3.5.0), depends on the following R packages: Cairo (>=1.5–9), dplyr (>=0.7.6), ggplot2 (>=3.0.0), leaflet (>= 2.0.2), leaflet.extras (>=1.0.0), robustbase (>=0.93–3), shiny (>=1.1.0), tidyr (>=0.8.1). These can be installed manually using the devtools library:

```
devtools::install_version("Cairo", version = "1.5-9")
devtools::install_version("dplyr", version = "0.7.6")
...
```

By installing the application in RStudio via R-devtools R libraries, the software depends on, should be installed on the fly.

```
devtools::install_github("JBrenn/SWCalibrateR")
```

Once installed the user can load the package and start the SWCalibrateR application:

```
library(SWCalibrateR)
runShinyapp()
```

Within the package, we make available an example dataset consisting of VWC sensor and VWC sample observation data paired with metadata. This dataset does load in general when the user starts the application. For further information about the example dataset have a look at its documentation or load the dataset:

```
?SWCalibrateR::data
data(data)
```

The user can upload their own dataset and analyse it. Find the tab for data upload at the left corner of the sidebar (see **Figure 1**). The sidebar does include all categories the user can filter the dataset for. Besides filtering the dataset, the user can decide for additional features of the app: use OLS or robust estimation of the model fit, and visualisation options.

The application comes with five main panels:

- **Model fit** visualises the subset point scatter and the estimated model with confidence interval. The user can fine-tune the model fit by toggle points.
- **Diagnostics** visualises model fit diagnostic, to help the user find leverage and outlier points.
- **Data Table** shows the data subset in a dynamic data table.

- **Map** maps the location of the selected samples using a leaflet application.
- **Description** documents the functionalities of the application.

Working online example

You can find a working online example of the application here: <https://jgbr.shinyapps.io/shiny/>. The same version runs with the following code:

```
shiny::runGitHub("JBrenn/SWCalibrateR",
  subdir="inst/shiny/", launch.browser = TRUE)
```

Below we describe in more detail the functionality of the application going through a working example.

Data subset

The input dataset is visualised in the panel *Model Fit*. The raw example dataset can be found here: <https://github.com/JBrenn/SWCalibrateR/blob/master/data/data.csv>. The dataset requires the following column categories (column names in **bold**): (1) name of the research **project ID**, (2) name of the observation **station ID**, geographic coordinates at station consisting of (3) **Latitude** and (4) **Longitude**, (5) **Altitude** of the station in meter a.s.l., (6) **Date** of observation, (7) **Landuse** type at the station, (8) **Soil depth** of soil moisture measurement in cm, (9) **Soil type** at the station, (10) **Sensor type**: generic soil moisture sensor type and (11) **Sensor ID**: specific soil moisture sensor name measuring SWC in-situ, (12) **Sensor VWC**: average VWC measured by soil moisture sensor (± 1 h time of observation), (13) **Sample VWC**: average soil core VWC (by default three soil core replicates are sampled per station, date and soil depth). VWC units are in %vol/%vol, ranging from 0 to 1. The user can upload their own dataset using the "Choose CSV file" option at the sidebar bottom. After data are uploaded, a handful of the above categories can be used to subset the dataset (see sidebar **Figure 1**). Multiple selection for all categorical variables is implemented.

Model fit

The data subset scatter and model fit, accompanied by the 95% confidence interval, is visualised in panel *Model Fit*. Estimates are computed either with

- `stats::lm`: linear regression model, based on OLS [2, 7], see **Figure 1**.
- `robustbase::lmrob`: MM-type estimator for linear regression model [3, 8], see **Figure 2**.

In the sidebar the user can control the regression method, visualise the row ID of the subset data table (row.ID in panel *Data Table*) and zoom in to the main data cloud. The user can toggle interactive leverage outliers by clicking one or mark multiple observations and apply the tab *Toggle points*. Reset to the initial dataset with the tab *Reset*. A decision tool indicating leverage and outlier points are the model diagnostic plots (panel *Diagnostics*). We do explain the panel *Diagnostics* in more detail later.

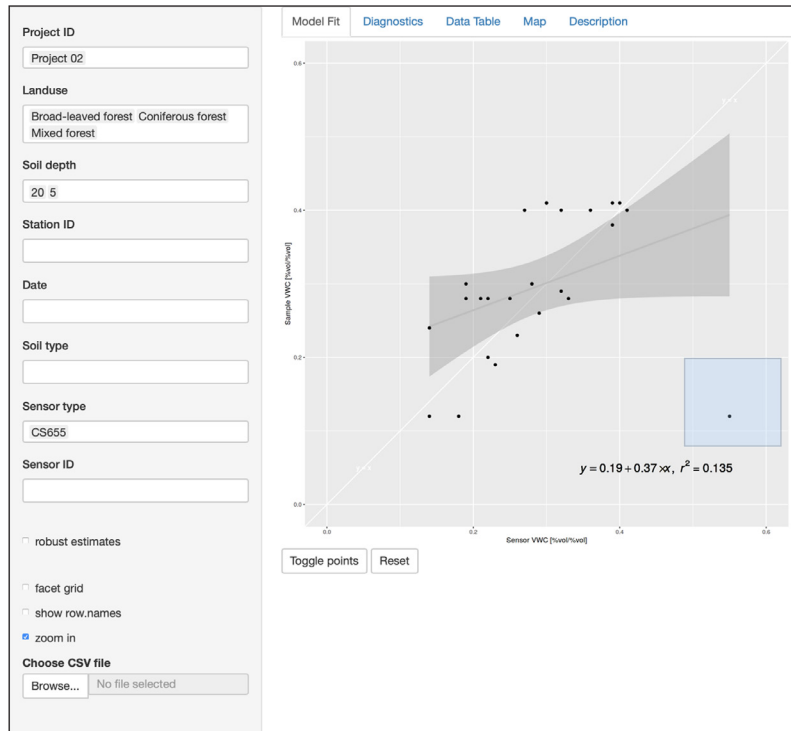


Figure 1: OLS model estimation for an example data subset. The subset arises from a choice in categories at the left sidebar (Project 02, forest ecosystems, upper soil layer, and sensor type CS655). Zoom in to the main data cloud is activated. Further options are a robust MM-type estimator, and visualization of row IDs. The user can browse and upload their own dataset. The application is organized in five panels. The first one shows the current OLS model fit (scatter plot accompanied by 95% confidence interval for the model estimate). A potential outlier point is brushed.

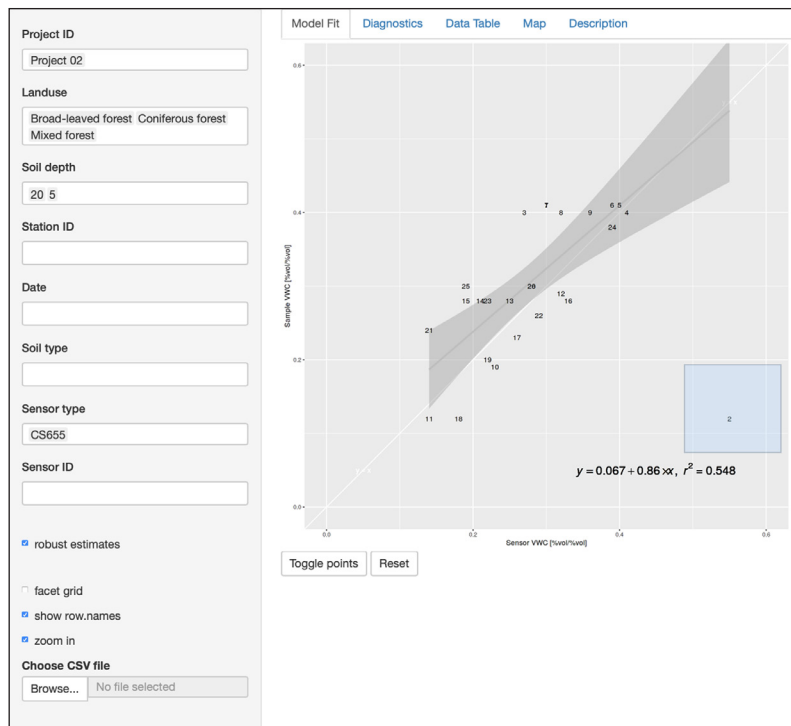


Figure 2: MM-type model estimation for an example data subset. Same data subset as in Figure 1 is visualised. Switching to the robust MM-type estimator enhances the coefficient of determination as the outlier point #2 (brushed point) does loose leverage on model estimation. The model equation and r^2 are comparable to the OLS estimate without point #2, which does refer to $y = 0.062 + 0.87x, r^2 = 59$. Applying visualisation of the row IDs facilitates identifying leverage and outlier points with the model diagnostic plots (panel Diagnostics).

OLS linear regression model

stats::lm applies OLS to find the relationship between an independent variable (Sample VWC) and a dependent variable (Sensor VWC). OLS has favourable properties if underlying assumptions are true. If assumptions are not met, OLS is known to be not robust. OLS assumes constant variance of model residuals (heteroscedasticity). Moreover, OLS is highly sensitive to outliers. If the outliers result from non-normal measurement error or some other violation of standard OLS assumptions the regression results lose validity. Thus, we advise the user to use MM-type robust regression techniques (tab *robust estimates*) if assumptions for OLS are violated.

MM-type estimator for linear regression

MM estimation (as implemented in *robustbase::robM*) is an alternative to OLS. It combines benefits of S-estimator and M-estimator. The M-estimator, as a “maximum likelihood type” estimator, is robust to outliers in the response variable, but turns out not to be resistant to outliers in the explanatory variables (leverage points). In fact, when there are outliers in the explanatory variables, the method has no advantage over OLS. The S-estimator is highly resistant to leverage points and is robust to outliers in the response. However, this method was also found to be inefficient. The MM-type estimator attempts to retain the robustness and resistance of S-estimation, whilst gaining the efficiency of M-estimation. For more details, please refer to e.g. [5].

Model diagnostic

The panel *Diagnostics* (Figures 3 and 4) contains model diagnostic plots created with the *stats::plot.lm* and

robustbase::plot.lmrob function respectively. The next paragraph describes how to read the diagnostic plots. An example illustrates diagnosing model estimates with OLS and MM-type estimator. Both methods share the first three diagnostic plot, only in the last they do differ (*stats::plot.lm*: plot 4a, *robustbase::plot.lmrob*: plot 4b).

Plot 1. Residuals vs. Fitted Values

Do the model residuals have non-linear patterns? Besides non-linearity of the residuals, this plot inherits a first hint towards unequal error variances, and outliers. For our example in the drier range of the fitted soil moisture values (points #11 and #18) and for one point (#2) in wet conditions we see higher error variances and thus non-linearity of the model residuals (Figure 3). This plot already advertises the use of a robust model estimator.

Plot 2. Normal Q-Q vs. Residuals

Are the model residuals normally distributed? Normal distribution of the residual is an underlying assumption for OLS. Strong deviations from the $y=x$ line, as we can see in Figure 3, does support the suspicion that this assumption is violated.

Plot 3. Scale-Location

Are the model residuals spread equally along the ranges of predictors? Alike in plot 1 we check the assumption of equal variance in the residuals (homoscedasticity) this time applying standardised residuals. Summing up we can identify three outliers in Figure 3: points #2, #11, #18. Probably not a single of these points is influential to the model estimation.

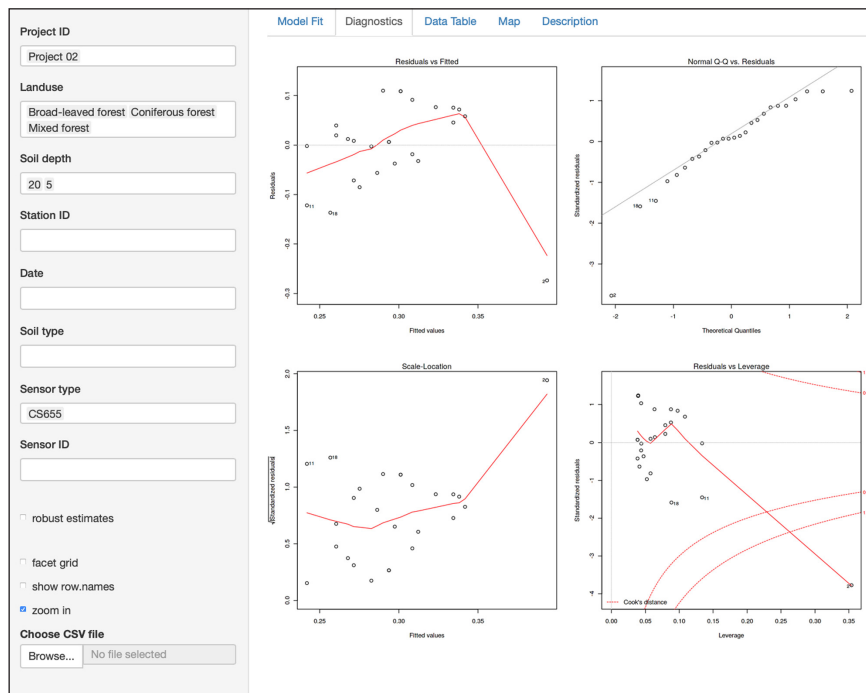


Figure 3: Model diagnostic plots for OLS estimator. The model diagnostic plots are applied to answer the following questions: (1) Do the model residuals have non-linear patterns? (2) Are the model residuals normally distributed? (3) Are the model residuals spread equally along the ranges of predictors? (4) Which are the influential outliers? Facing these questions helps testing underlying assumptions for OLS and in the end is beneficial for detection of outlier and leverage points (e.g. point #2).

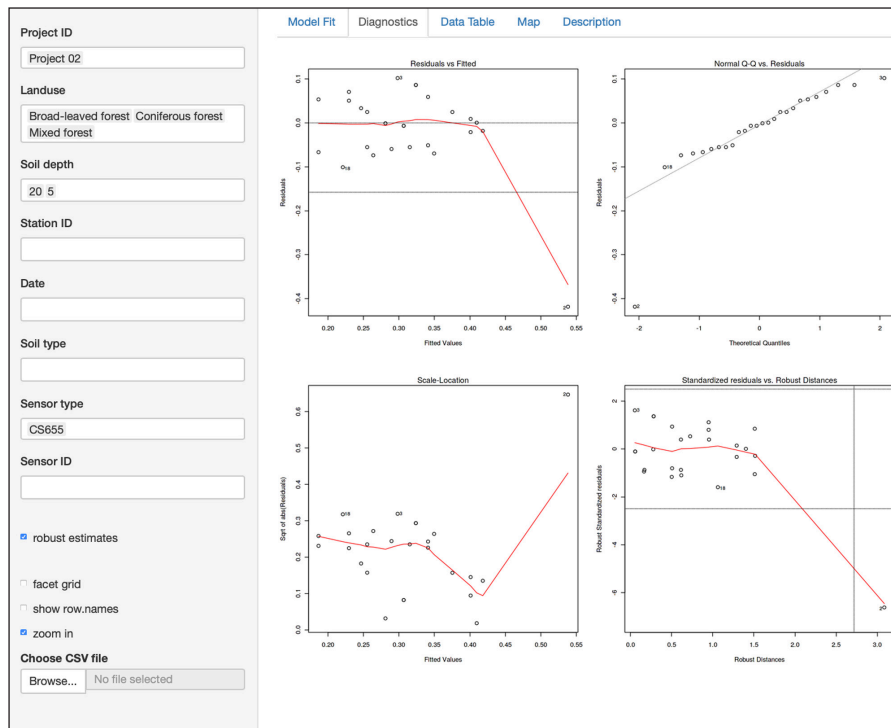


Figure 4: Model diagnostic plots for MM-type estimator (further explanation see caption Figure 3).

Plot 4a. *plot.lm*: Residuals vs. Leverage

Which are the influential outliers? Unlike the other plots, here patterns are not relevant. We simply look for cases outside the curved lines, the Cook's distance. These points should be excluded from model estimation with care, as they do influence the regression result. Point #2 can be clearly identified as outlier and leverage point.

Plot 4b. *plot.lmrob*: Standardised Residuals (SR) vs. Robust Distances (MD)

Alike in plot 4a we test for influential points. The plot in **Figure 5** is divided in four regions marked by darkened lines: (1) Regular, (2) Outlier, (3) Leverage, (4) Outlier and Leverage. As for plot 4a we can identify point #2 as outlier point, influencing the model estimate. For interpretation help see **Figure 5**. For detail on the methods we refer to [4].

Quality control

SWCalibrateR has been tested in with several web browsers, including Google Chrome, Safari, Firefox, and IE10+. The application is available at <https://jgbr.shinyapps.io/shiny/>. Full documentation using Roxygen2 is available at <https://github.com/JBrenn/SWCalibrateR>, where also issues or requests may be filed.

(2) Availability

Operating system

SWCalibrateR is a platform-independent software package, the web application is compatible with modern web browsers (IE 10+, Google Chrome, Firefox, Safari, etc.).

Programming language

R

Additional system requirements

None

Dependencies

SWCalibrateR depends on R ($\geq 3.5.0$) and several R packages: Cairo ($\geq 1.5-9$), dplyr ($\geq 0.7.6$), ggplot2 ($\geq 3.0.0$), leaflet ($\geq 2.0.2$), leaflet.extras ($\geq 1.0.0$), robustbase ($\geq 0.93-3$), shiny ($\geq 1.1.0$), tidyR ($\geq 0.8.1$).

List of contributors

Johannes Brenner
 Giulio Genova
 Georg Niedrist
 Giacomo Bertoldi
 Stefano Della Chiesa

Software location

Archive

Name: Zenodo
Persistent identifier: <https://doi.org/10.5281/zenodo.1745327>
Licence: MIT
Publisher: Johannes Brenner
Version published: v1.1.0
Date published: 2019-03-19

Code repository

Name: GitHub
Identifier: <https://github.com/JBrenn/SWCalibrateR>
Licence: MIT
Language: English

(3) Reuse potential

SWCalibrateR is a reusable toolbox, with which researchers in the fields of agriculture and hydrology can easily calibrate soil moisture sensors depending on their

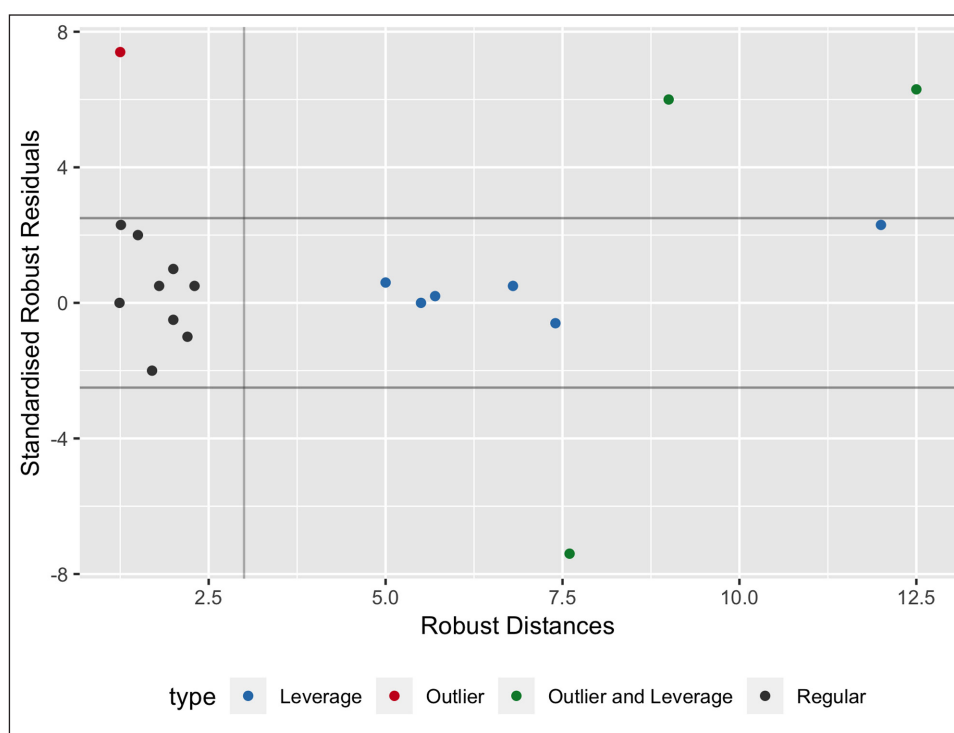


Figure 5: Interpretation of standardised residuals vs. robust distance. The plot divides the value range in four regions marked by lines and coloured points: (1) Regular, (2) Outlier, (3) Leverage, (4) Outlier and Leverage. Points in the fourth category are of major importance as they are outlier points, influencing the model fit if removed from estimation.

requirements. Previously researchers might have adopted a simple spreadsheet to retrieve calibration equation, now SWCalibrateR does fill this gap. Being implemented as a user-friendly package within the R environment, including a web interface a large variety of users will be able to reuse the software. Sensor calibration is a very common task in environmental science, thus, SWCalibrateR could be further developed with a generalised design to calibrate any sort of sensor or measurements. Moreover, further development could consider import time series of uncalibrated measurements and provide in output calibrated time series which in turn can be directly used for any sort of application which require high quality data.

Competing Interests

The authors have no competing interests to declare.

References

1. **Chang, W, Cheng, J, Allaire, J J, Xie, Y and McPherson, J** 2015 Shiny: web application framework for R. R package version 0.11, 1(4): 106.
2. **Chambers, J M** 1992 Linear models. Chapter 4 of Statistical Models in S, Chambers, J M and Hastie, T J (eds.), Wadsworth & Brooks/Cole.
3. **Koller, M and Stahel, W A** 2011 Sharpening Wald-type inference in robust regression for small samples. *Computational Statistics and Data Analysis*, 55(8): 2504–2515. DOI: <https://doi.org/10.1016/j.csda.2011.02.014>
4. **Rousseeuw, P J and van Zomeren, B C** 1990 Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411): 633–639. DOI: <https://doi.org/10.1080/01621459.1990.10474920>
5. **Susanti, Y, Pratiwi, H, Sulistijowati, H S and Liana, T** 2014 M estimation, S estimation, AND MM estimation in robust regression. *International Journal of Pure and Applied Mathematics*, 91(3): 349–360. DOI: <https://doi.org/10.12732/ijpam.v91i3.7>
6. **Topp, G C, Davis, J L and Annan, A P** 1980 Electromagnetic Determination of Soil Water Content: Measurements in Coaxial Transmission Lines. *Water Resour. Res.*, 16: 574–582. DOI: <https://doi.org/10.1029/WR016i003p00574>
7. **Wilkinson, G N and Rogers, C E** 1973 Symbolic Description of Factorial Models for Analysis of Variance. *Applied Statistics*, 22(3): 392. DOI: <https://doi.org/10.2307/2346786>
8. **Yohai, V J** 1987 High Breakdown-Point and High Efficiency Robust Estimates for Regression. *The Annals of Statistics*, 15(2): 642–656. DOI: <https://doi.org/10.1214/aos/1176350366>

How to cite this article: Brenner, J, Genova, G, Bertoldi, G, Niedrist, G and Chiesa, S D 2019 SWCalibrateR: Interactive, Web – Based Calibration of Soil Moisture Sensors. *Journal of Open Research Software*, 7: 20. DOI: <https://doi.org/10.5334/jors.254>

Submitted: 04 December 2018 **Accepted:** 23 April 2019 **Published:** 20 June 2019

Copyright: © 2019 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS 