The Fundamental Role of Rainfall on Soil Moisture:
A data-driven statistical study on the Aburrá Valley

Author:
Manuel Andrés Furió Urbina

A thesis submitted for the degree of
Bachelor of Science in Civil Engineering

Thesis Advisers:
Derly Estefanny Gómez García, Doctor of Environmental Engineering
Edwin Fabián García Aristizábal, Doctor of Engineering

University of Antioquia
Engineering Faculty
Environmental School
Civil Engineering
Medellín
2025

Quotation	(Furió, 2025)
Reference APA Style 7th edition (2020)	Furió, M. (2025). *The Fundamental Role of Rainfall on Soil Moisture: A data-driven statistical study on the Aburrá Valley* [Undergraduate Thesis]. University of Antioquia, Medellín.

Acknowledgements

This work would not have been possible without the constant stream of ideas and invaluable guidance from my two thesis advisers. Their support was instrumental in helping me create my very first thesis document, and I could not be more grateful.

Doctor Aristizábal was my first point of contact in the development of this thesis. I sought his guidance due to an excellent experience in his Rock Mechanics coursework, a subject I enjoyed far more than I had anticipated. Inspired by that experience, I approached him for assistance in writing about a topic related to the field—a decision that proved highly beneficial. His wisdom became particularly invaluable during the latter stages of the thesis, helping me gain a deeper understanding of the data I was working with. This insight ultimately contributed to one of the paper's most significant results.

Doctor Gómez provided essential support during the early stages of this research. Her goal was to ensure I was well-prepared to face the unique challenges of this study. Many of her insights found their way into the final work, either directly within the document or as frameworks that enhanced my understanding of the data. Her constructive critique of my writing process was particularly impactful, allowing me to refine my skills and elevate the quality of this thesis.

I would also like to extend my heartfelt gratitude to my professors and classmates at the University of Antioquia for their encouragement, insights, and support. Whether through guidance or inspiration, they played a significant role in shaping my academic journey and future aspirations. Thanks everyone!

Table of Contents

Abstract 6

1. Introduction 7

2. Objectives 8

    2.1 General Objective 8

    2.2 Specific Objectives 8

3. Theoretical Framework 9

4. Methodology 10

    4.1 Selection of study area 10

    4.2 Data pre-processing 11

    4.3 Definition of variables 13

    4.4 Individual rainfall event identification 15

5. Results and Discussion 16

    5.1 Daily data findings 16

    5.2 Hourly data findings 17

    5.3 Soil moisture drop-off 18

    5.4 Correlation between rainfall ranges and soil moisture variables 19

Conclusions and Proposed Future Studies 22

References 23

Appendix 26

    A1. Daily data processing (Python code) 26

    A2. Hourly data processing (Python code) 34

    A3. Further statistical analysis for hourly data (Python code) 40

    A4. Rainfall duration and intensity for hourly data (Python code) 41

    A5. Four-dimensional matrix data (current to second offset) 43

List of Figures and Tables

Figure 1 Map of the Aburrá Valley within Colombia 10

Figure 2 Entire dataset completion of station 396 12

Figure 3 First (a) and last (b) steps of filtering algorithm 15

Figure 4 Antecedent rainfall (ARF#) correlation 16

Figure 5 Individual rainfall event analysis 17

Figure 6 Current Rainfall vs Soil Moisture Δ 19

Figure 7 First Rainfall Offset vs Nominal Soil Moisture 20

Figure 8 Second Rainfall Offset vs Soil Moisture Δ 21

Table 1 SIATA Quality Index 11

Table 2 Soil moisture drop-off 18

Abstract

This study took a quantitative approach to understanding how rainfall affects soil moisture, which is a common trigger for landslides. The research focused on a local area (the Aburrá Valley), due to its high susceptibility to landslides. Landslides pose a significant threat to the local population due to the region's steep, landslide-prone terrain and frequent tropical rainfall. This threat impacts the region twofold: it exacerbates poverty and leads to significant loss of life. Many other studies have used empirical, physical, and satellite-based approaches to predict landslides. However, the fluctuating state of soil moisture had not been a focus of previous research. To address this gap, this study placed strong emphasis on soil moisture fluctuations following rainfall events. Our goal was to develop an empirical model that links rainfall and soil moisture measurements. To achieve this, pluviograph data and soil moisture sensor measurements were processed and analyzed to define the variables and parameters for an empirical model capable of predicting soil moisture levels based on rainfall data. Ultimately, we developed empirical equations to fulfill this purpose while leaving room for further research. The results and applications of this model could contribute to future studies on rainfall events that trigger landslides, including the development of early warning systems that, when properly implemented, could help protect populations living in high-risk areas.

Keywords: landslide, rainfall, risk assessment, soil moisture.

1. Introduction

Landslides are a global phenomenon with serious consequences for humanity. They claim thousands of lives annually and cause substantial economic losses, further exacerbating poverty in many areas, particularly in mountainous regions with geological instability, making landslides a significant focus of research across various scientific circles (Aristizábal et al., 2010; Marc et al., 2022; Segoni et al., 2018). In the Colombian Andean region, over the past century, more than 7000 deaths have been attributed to over 10000 landslides (Jaramillo González et al., 2023).

Although rainfall is known to be the most common trigger for landslides in the Colombian Andean region (Gómez et al., 2023), daily rainfall is an unreliable indicator of landslide risk due to the residual effect of past rainfall on soil moisture (Crozier & Eyles, 1980).

Over the years, several methods—empirical, physical, and satellite-based models—have been developed to identify rainfall thresholds that can trigger landslides (Gonzalez et al., 2024; Segoni et al., 2018). Regarding empirical methods, existing research on rainfall thresholds seems to overlook the fluctuating state of soil moisture. This pilot study seeks to address this gap by exploring how soil moisture varies after a rainfall event.

Using rainfall and soil moisture data from a selected area in the Colombian Andean region, this study aimed to define hydrological variables and propose parameters that correlate rainfall with soil moisture, ultimately leading to the development of an empirical model. Additionally, we hope these results will inform future research on rainfall thresholds for landslide prediction. By improving the accuracy of such thresholds, governments, insurance companies, and other stakeholders can better prepare for landslides through emergency evacuations, cost estimates, and other preventive measures (Glade, 2000; Pradhan et al., 2019).

2. Objectives

2.1 General Objective

Study methodologies for determining soil moisture and its relationship with rainfall in the Aburrá Valley as a foundation for future studies on the assessment of landslide hazard and risk triggered by rainfall.

2.2 Specific Objectives

- To select a pilot study area for analysis within the Aburrá Valley region, Antioquia.
- To collect, process, and analyze rainfall and soil moisture datasets.
- To develop an empirical model that establishes the relationship between soil moisture and key rainfall variables.

3. Theoretical Framework

In the past decade, rainfall has been recognized as a leading cause of landslide hazards worldwide (Liu et al., 2023; Segoni et al., 2018), with climate change contributing to an increase in storm frequency (Marc et al., 2022). Recent studies have found that man-made slopes are more susceptible to landslides (Aristizábal et al., 2017; Martinović et al., 2018; Pradhan et al., 2019).

From pluviometric readings, various parameters can be derived, such as cumulative rainfall, antecedent rainfall, and the intensity and duration of storms (Aristizábal et al., 2010). Cumulative and antecedent rainfall have been shown to be significant triggers due to their impact on soil moisture (Garcia-Urquia, 2016; Gonzalez et al., 2024; Oliveira et al., 2022). Unfortunately, there is insufficient theoretical knowledge about how rainfall infiltrates the ground, increasing soil moisture and potentially leading to landslide hazards (Aristizábal et al., 2010). Numerous statistical and empirical methods have been developed to better understand this phenomenon (Crozier & Eyles, 1980; Glade, 2000; Soto et al., 2019). In some areas, it has been shown that daily rainfall readings could indicate a measure on how soil moisture develops (this parameter being very critical since daily rainfall alone isn't enough), using recession coefficients of flood hydrographs, effectively establishing a surrogate measurement of soil moisture (Crozier & Eyles, 1980; Glade, 2000). In soils with slower drainage, rainfall has a prolonged effect on soil moisture over extended periods (Crozier & Eyles, 1980; Iadanza et al., 2016; Wu et al., 2015).

These methods have led to the development of rainfall thresholds, which define conditions under which landslides are likely to occur (Jaramillo González et al., 2023). Rainfall thresholds are now the most used tool for predicting landslides and are often a critical component of early warning systems (EWS). It is essential to understand the various parameters involved to comprehend their influence and effectively prepare for hazardous events (Huggel et al., 2010; Iadanza et al., 2016; Jaramillo González et al., 2023; Martinović et al., 2018; Segoni et al., 2018), as miscalibration of EWS can undermine public trust, complicating emergency evacuations and other preventive measures (Gonzalez et al., 2024; Huggel et al., 2010). These thresholds are strongly influenced by rainfall, soil moisture, storm duration, and intensity (Aristizábal et al., 2010; Liu et al., 2023). Correa et al. (2020) have compiled a list of commonly used thresholds for EWS.

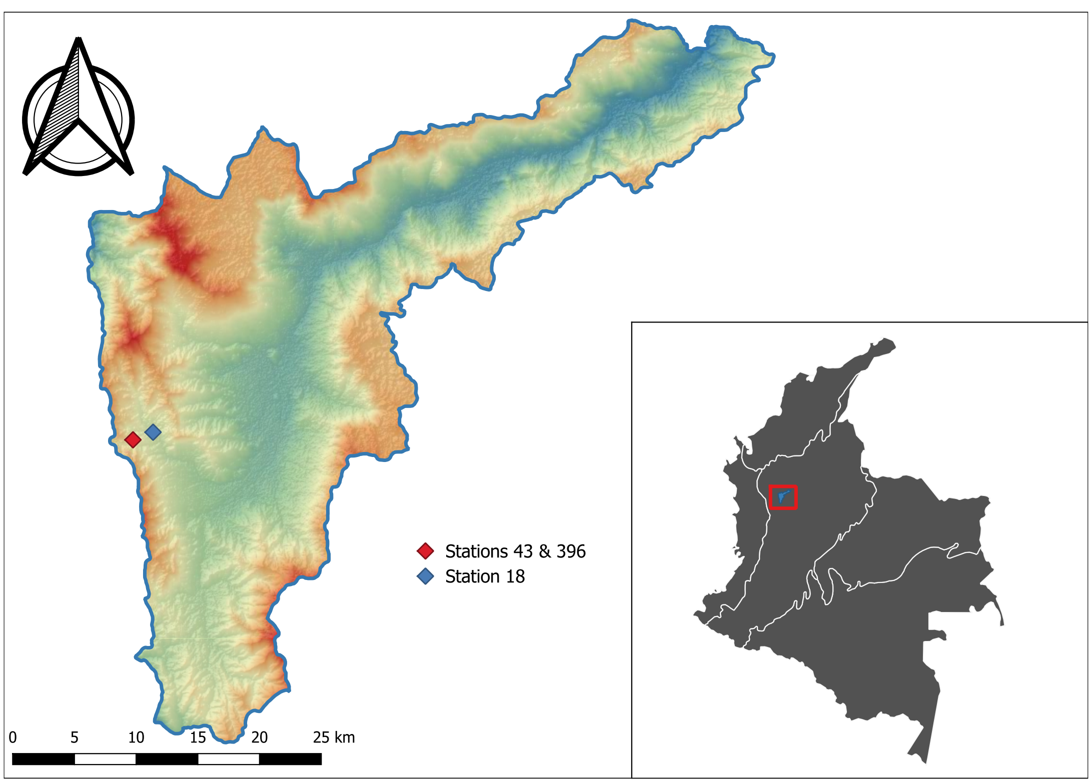
4. Methodology

4.1 Selection of study area

The study area was selected within the Aburrá Valley (**Figure 1**) due to its high susceptibility to landslides. Among the sources consulted, the Sistema de Alerta Temprana del Valle de Aburrá (Early Warning System of the Aburrá Valley, SIATA) provided minute-by-minute soil moisture data from seven different stations using volumetric water content sensors (MAS-1, Decagon Devices) installed across the valley. Volumetric water content is defined as the ratio of the water volume to the total volume of the sampled soil. Depending on the station, soil moisture data were collected from various depths, namely 0.1, 0.5, and 0.9 meters. Each soil moisture sensor is paired with a dual pluviograph, which measures rainfall at a resolution of 0.01 inches (0.254 millimeters) every minute. Of the seven available stations, sensor 396 (paired with pluviograph 43) is the oldest, with data available from late 2019 to the present (**Figure 1**).

Figure 1

Map of the Aburrá Valley within Colombia



The second-oldest sensor began operating in late 2021. Notably, sensor 396 is the only one that collects soil moisture data from all three depths. Considering its extensive dataset, we decided to work exclusively with this sensor. This sensor is located on the grounds of a rural school, Centro Educativo Quebrada Alta, in San Antonio de Prado, Medellín, at an altitude of approximately 2,080 meters. For this research, we were provided data from October 2019 to August 2024.

4.2 Data pre-processing

SIATA provides a quality index for rainfall and soil moisture measurements (**Table 1**). Based on this index, each measurement is classified as either reliable or dubious. Before removing all dubious values, we confirmed that such values were either erratic (-999), nonsensical (e.g., extremely high or low values), or exhibited significant variance in the soil moisture dataset. All indexed data with values of 1 or 2 were retained.

Table 1

SIATA Quality Index

Description	Index
Reliable quality of real-time data	1
Doubtful quality of real-time data in all devices	151
Doubtful quality of real-time data in both pluviographs compared with weather radar	1510
Doubtful quality of real-time data from pluviograph 1	1511
Doubtful quality of real-time data from pluviograph 2	1512
Doubtful quality of data from 0.1m sensor	1521
Doubtful quality of data from 0.5m sensor	1522
Doubtful quality of data from 0.9m sensor	1523
Doubtful quality of data from 0.1m and 0.5m sensor	1526
Doubtful quality of data from 0.1m and 0.9m sensor	1527
Doubtful quality of data from 0.5m and 0.9m sensor	1528
Reliable quality of data not obtained in real time	2
Doubtful quality of data not obtained in real time in all devices	251
Doubtful quality of data not obtained in real time in both pluviographs compared with weather radar*	2510
Doubtful quality of data not obtained in real time from pluviograph 1	2511
Doubtful quality of data not obtained in real time from pluviograph 2	2512
Doubtful quality of data not obtained in real time from 0.1m sensor*	2521
Doubtful quality of data not obtained in real time from 0.5m sensor*	2522
Doubtful quality of data not obtained in real time from 0.9m sensor*	2523
Doubtful quality of data not obtained in real time from 0.1m and 0.5m sensor*	2526
Doubtful quality of data not obtained in real time from 0.1m and 0.9m sensor*	2527
Doubtful quality of data not obtained in real time from 0.5m and 0.9m sensor*	2528

* Index exists without description. Guess based on indexing structure and data behavior

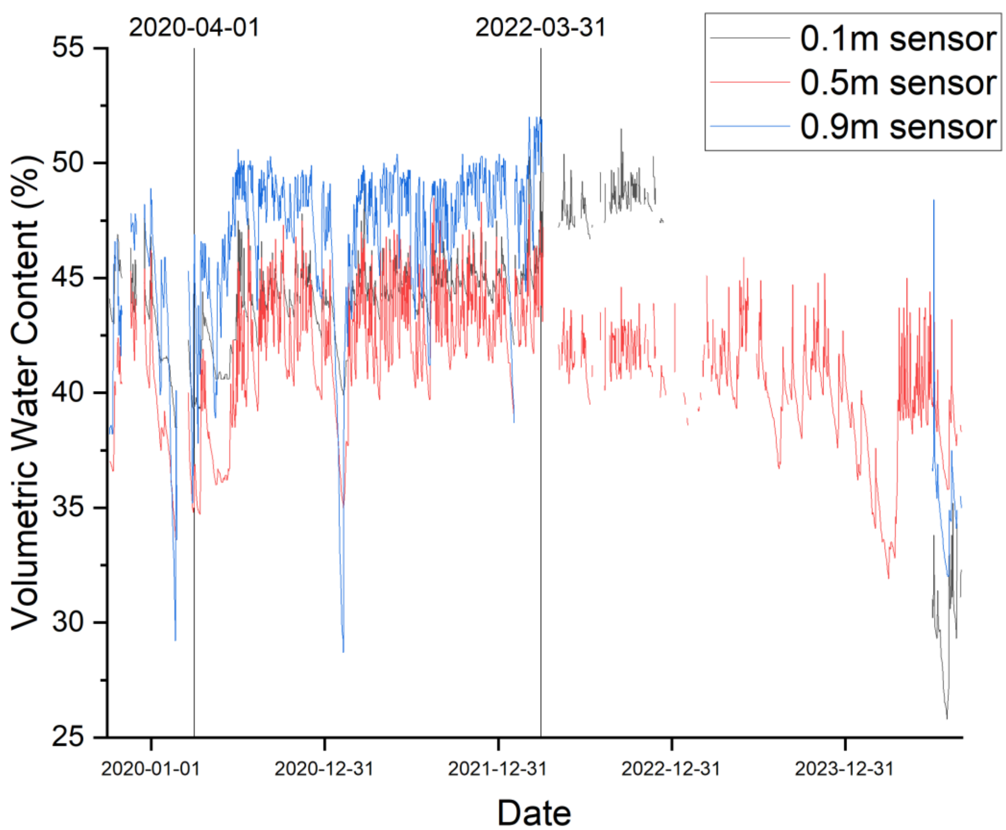
Source. (SIATA, 2024)

Given the availability of minute-by-minute data, we calculated the percentage of data completeness for each day and hour. Using this metric, we excluded all days or hours with less than 75% completion for rainfall data (e.g., days with less than 1080 minutes of data were excluded). For soil moisture data, this threshold was lowered to 40%, owing to its continuous (rather than discrete, as with rainfall) nature.

Examining the dataset from October 2019 to August 2024, we found that roughly outside the period from April 2020 to March 2022, the sensors at 0.1- and 0.9-meter depths had very low data completion percentages (**Figure 2**). Consequently, we limited our analysis to this two-year range to ensure the highest possible concentration of soil moisture data.

Figure 2

Entire dataset completion of station 396



We also repaired portions of the rainfall data from pluviograph 43 using data from pluviograph 18 (**Figure 1**), which is located approximately 1,700 meters away and 130 meters lower in altitude. The similarity of their rainfall data allowed us to repair 93 days of rainfall data during the two-year period. Only one day of data was heavily corrupted; its rainfall value was manually inferred using limited information from pluviographs 18 and 43.

The remaining rainfall and soil moisture data were aggregated to both daily and hourly resolutions for this research. Rainfall values were summed (using the highest of the two minutely measurements), and soil moisture measurements were averaged across each hour or day.

For soil moisture data, additional repairs were performed, but only for the daily dataset. Specifically, we used linear interpolation to estimate six days of missing data. Additionally, we filled an 11-day gap in the dataset by copying a period of no rainfall from January 15, 2021, to January 24, 2022. This reduced our low-completion soil moisture days to only 21.

4.3 Definition of variables

The reviewed literature highlighted that daily or hourly rainfall alone are not sufficient metrics for understanding their influence on soil moisture. This is due not only to the residual effects of past soil moisture (Crozier & Eyles, 1980) but also to the delayed movement of rainwater through the soil, influenced by its permeability (Aristizábal, 2005).

Frequently used for risk assessment is the antecedent rainfall index, which is defined as the cumulative rainfall over a period of days. In this calculation, each day's rainfall prior to the current day is weighted by an exponential decay factor between 0 and 1. This approach simulates the gradual contribution of older rainwater to subterranean flows, which no longer affect superficial soil moisture (Crozier & Eyles, 1980).

Another rainfall variable we considered is offset rainfall, which we defined as the rain that occurred a specified number of hours or days ('x') before the current time. This metric is particularly useful for hourly data analysis, as it helps visualize the time lag between rainfall events and their detection by soil moisture sensors. In other words, it serves as a surrogate for soil permeability.

For volumetric water content measurements, we introduced a Δ-value for each day and hour. The Δ-value at a given time is calculated as the current volumetric water content minus the value from the previous day or hour. The usefulness of the Δ-values lies in the fact that different volumetric water content values can be associated in our dataset with the same amount of rainfall. This is particularly true for certain hours or days without rainfall that were preceded by a rainfall event. The Δ-values are intended to provide a fresh perspective on the possible correlation between rainfall and soil moisture.

Using these variables, we developed a simple correlation model to guide our analysis and identify the rainfall variable that best correlates with soil moisture. This model involves determining the number of days (or hours) in which a given amount of rainfall corresponds to a positive Δ-value, as well as its inverse—the number of days (or hours) where rainfall below this certain threshold correlates with a negative Δ-value. This value ranges from 0 to 730 days but can also be expressed as a percentage. A value of 50% indicates no correlation, as it represents the expected outcome if this binary correlation model were randomized. A second version of this code also includes the option to exclude non-rainy days or hours.

Additionally, we developed a quick calculation to determine how quickly the soil drains each day. While this calculation is feasible using hourly data, the resulting numbers are very small and may be influenced by preceding rainfall events.

The Python code used to compute these variables and perform data preprocessing can be found in **Appendix 1** and **Appendix 2**.

During the mid-to-late stages of this research, a specific result prompted us to analyze the mean, median, and maximum values alongside different hourly rainfall ranges. To facilitate this analysis, we constructed a four-dimensional matrix (**Appendix 3**) with the following axes:

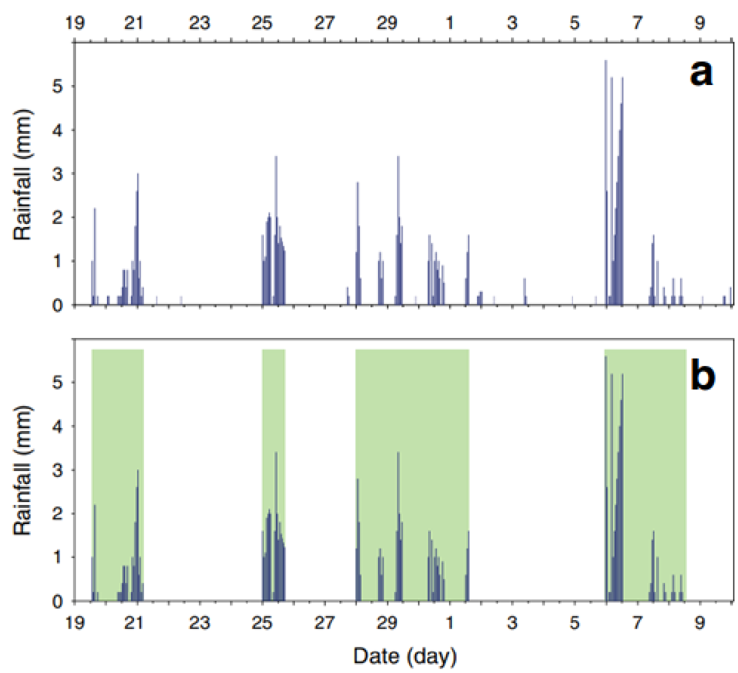
- Mean, median, and maximum values of volumetric water content and Δ-values.
- Soil moisture sensor depth and type (e.g., nominal or Δ).
- Rainfall ranges: 0, 0.254–1, 1–4, 4–7, 7–10, 10–15, 15–20, and 20+ millimeters of rainfall.
- Type of rainfall variable (e.g., current or x-offset).

4.4 Individual rainfall event identification

Motivated by Aristizábal (2005), we utilized hourly data to focus on specific rainfall events, aiming for a better understanding of how soil responds to rainwater. We manually selected twelve rainfall events based on their magnitude and temporal distance from preceding events, prioritizing the strongest ones least influenced by prior rainfall. To facilitate this analysis, we developed a Python adaptation (**Appendix 4**) of the algorithm proposed by Melillo et al. (2015). While less powerful than the original, this adaptation performs several steps to filter out irrelevant rainfall events and identify the onset of significant rainfall events (**Figure 3**). From this refined dataset, we were able to estimate storm duration and intensity.

Figure 3

First (a) and last (b) steps of filtering algorithm



Source. (Melillo et al., 2015)

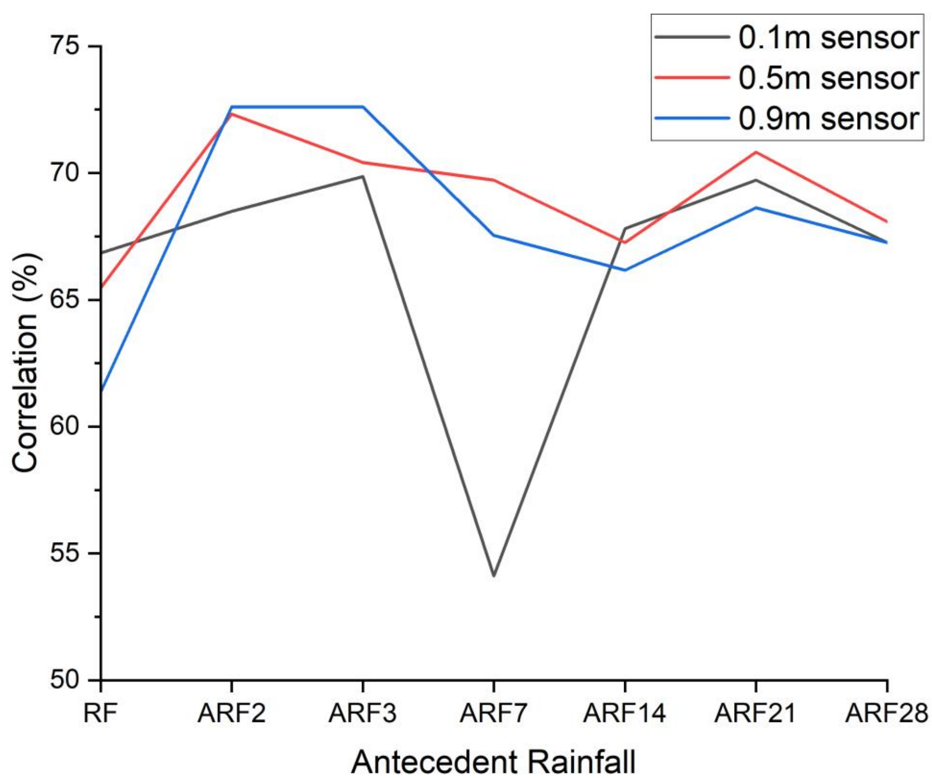
5. Results and Discussion

5.1 Daily data findings

Using our simple correlation model, we thoroughly analyzed the daily data. However, we were unable to identify strong correlations; instead, the data appeared inconsistent and anomalous, resembling noise with only a weak correlation. **Figure 4** shows one of the results. Keep in mind that a 50% value indicates no correlation, as it is the expected outcome from a randomized dataset.

Figure 4

Antecedent rainfall (ARF#) correlation



Note. In ARF#, the hashtag stands for the number of days included in the calculation including the current one.

The lack of results from the daily data, which we initially chose based on the literature we reviewed, shifted our research focus to the analysis of hourly data.

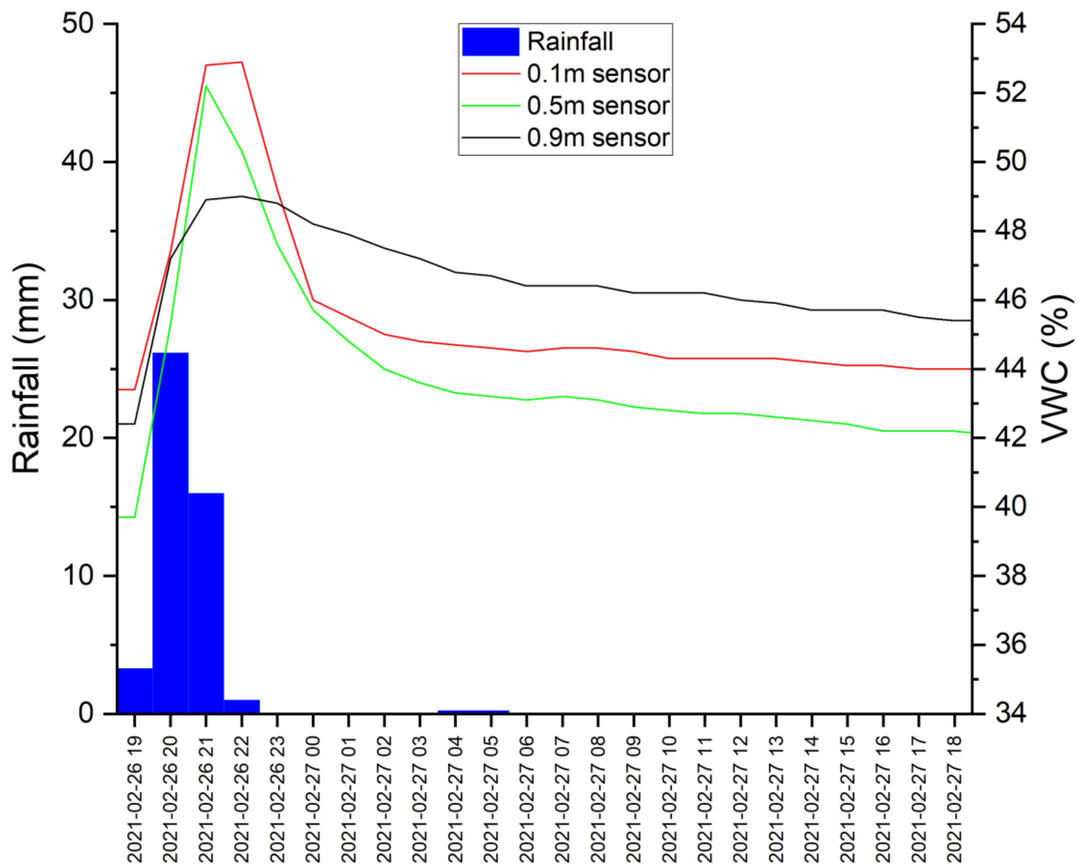
5.2 Hourly data findings

Hourly data showed significant promise when evaluated with our correlation model (using only rainy days), with correlation percentages peaking at 85% for the first offset rainfall hour. However, the correlation drops sharply from the second hour onward, suggesting fast-draining soil. This observation prompted a closer analysis of the events.

Figure 5 presents the graph of one of the twelve individually analyzed rainfall events. These events were randomly selected based on the criteria of being heavy rain events and being temporally distant from prior events. All other events exhibited nearly identical behavior.

Figure 5

Individual rainfall event analysis



One of the first observations is the sharp decline in volumetric water content one to two hours after a rainfall event, as detected by the uppermost sensors. This confirms the initial indication from our basic correlation model that the soil drains relatively quickly, which also helps explain why the daily data appeared so poor.

Similarly, there is a sharp increase in water content during the current hour or the hour immediately following the rainfall, which is logical and expected. The bottom sensor also shows a somewhat sharp rise, but its decline is much more gradual compared to the uppermost sensors.

Another notable statistic, evident when examining all these events simultaneously, is that the volumetric water content never exceeds 53%. This indicates that the soil becomes saturated at this threshold, preventing further rainfall from penetrating and causing it to simply run off.

Lastly, since the maximum rainfall value may not only represent a trend but also be associated with specific ranges, and given our focus on heavy rainfall events, we realized it would be sensible to examine other ranges, as well as their relationship to the mean and median.

5.3 Soil moisture drop-off

As previously observed, the hourly drop in soil moisture can be quite drastic in the hours immediately following a rainfall event. This is likely due to gravitational forces overpowering the capillary forces that retain water in the soil, resulting in rapid drainage. However, this phenomenon diminishes quickly after the initial hours post-rainfall.

Table 2

Soil moisture drop-off

	Daily Dataset		Hourly Dataset	
Depth	Mean	Median	Mean	Median
0.1m	-0.2461%	-0.20%	-0.0646%	0.00%
0.5m	-0.6181%	-0.30%	-0.0908%	0.00%
0.9m	-0.7439%	-0.70%	0.0478%	0.00%

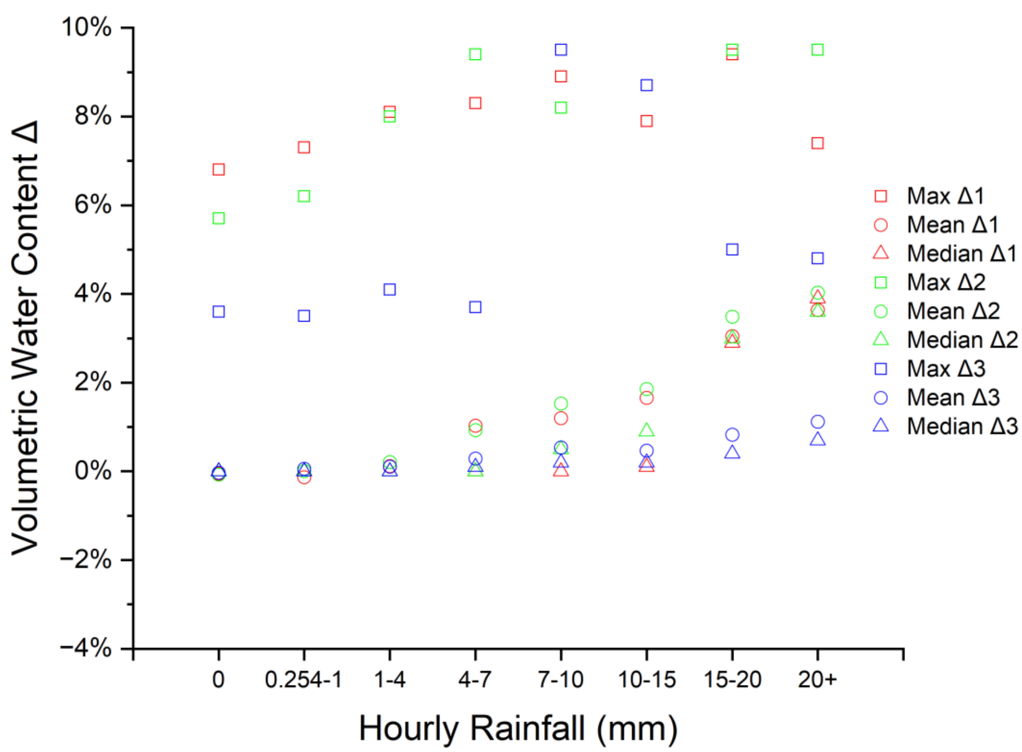
Examining the hourly data in **Table 2** reveals that the soil moisture drop-off is essentially zero based on the median values, while the mean values may be influenced by preceding rainfall events. This is slightly evident in **Figure 5**, where the final soil moisture values appear to flatline. As expected, the daily data appears to be more informative, providing insight into how quickly water drains each day, albeit with some caveats. Notably, the third sensor at 0.9 meters shows unusually high values, and all mean values seem to be influenced by past rainfall events.

5.4 Correlation between rainfall ranges and soil moisture variables

Figure 6 presents highly promising correlations for the mean values of the Δ across all sensors, with Pearson's R values of 98.8%, 99.1%, and 97.4%. However, the third sensor exhibits a flatter slope compared to the other two.

Figure 6

Current Rainfall vs Soil Moisture Δ



Although other equations could be derived to predict soil moisture, we chose to highlight only those with the strongest correlations. Based on these findings, we propose two equations:

$$\Delta \text{ at } 0.1\text{m} = 0.0018 * \text{RF}_0 - 0.0021$$
$$\Delta \text{ at } 0.5\text{m} = 0.0020 * \text{RF}_0 - 0.0019$$

Figure 7 reveals another highly promising correlation, this time for the nominal soil moisture values. The Pearson's R values indicate near-perfect correlations when examining the averages and medians of the uppermost depths. Here, we suggest two additional equations:

$$VWC\ at\ 0.1m = 0.0042 * RF_{-1} + 0.4439$$
$$VWC\ at\ 0.5m = 0.0038 * RF_{-1} + 0.4303$$

Figure 7

First Rainfall Offset vs Nominal Soil Moisture

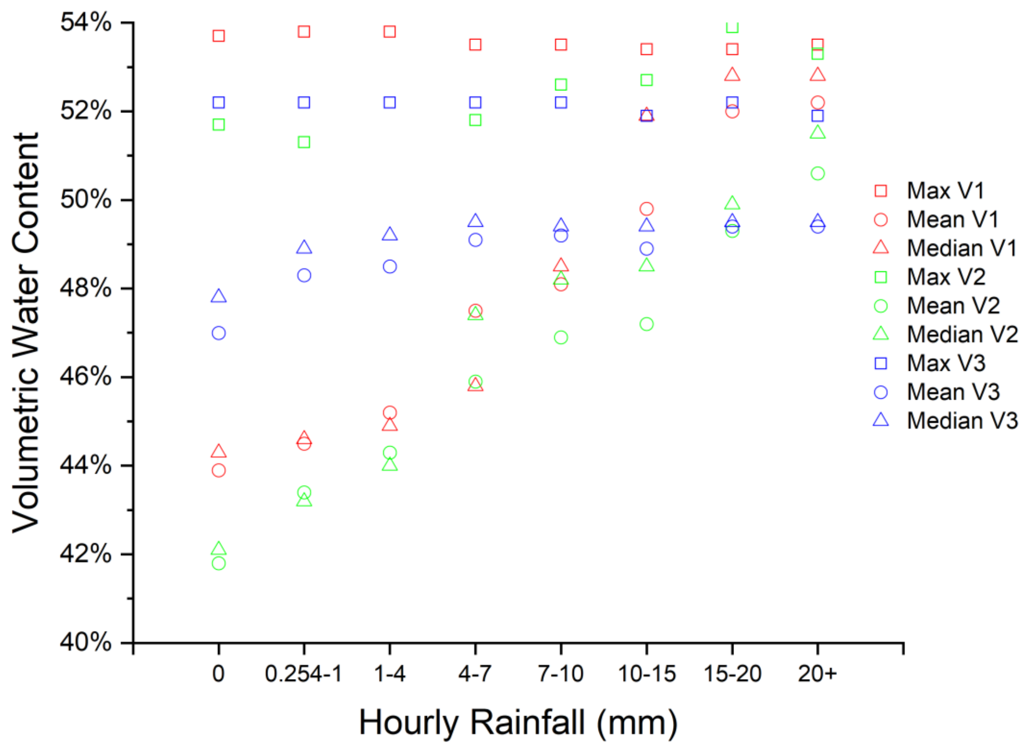
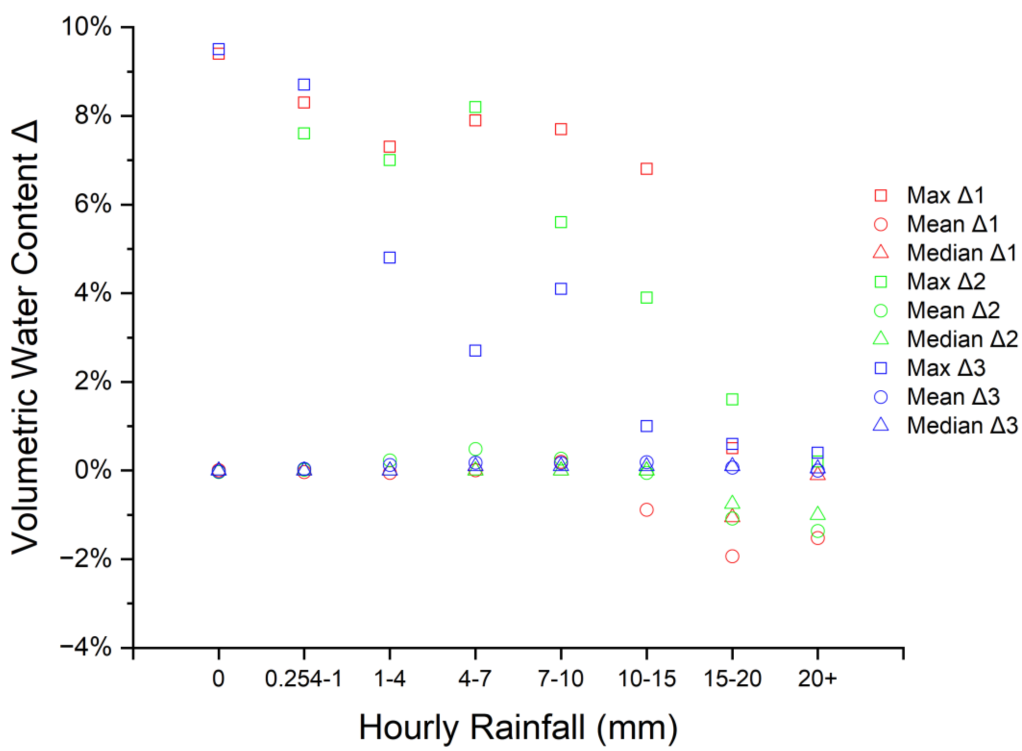


Figure 8 uncovers another interesting result: a very high negative correlation between the second offset and the maximum soil moisture Δ at 0.5 meters, with a Pearson's R of -96.6%. The sole outlier appears in the 4-to-7-millimeter range, likely due to contamination from a prior rainfall event. While the other depths also show strong negative correlations, they seem more affected by previous events. The relatively clean results at the middle depth may be due to luck, as a single contamination could significantly skew the maximum value data, this is evident in **Figure 7**.

Figure 8

Second Rainfall Offset vs Soil Moisture Δ



Lastly, we observe a concentration of average and median values around zero, aligning with the results from **Section 5.3**. This suggests that, within two hours, the drainage process is essentially complete, provided the rainfall for that event is not excessive.

The 4D matrix, containing data from the current rainfall to the second offset, is fully available in both its exploded 2D form and its multiple graph form in **Appendix 5**.

Conclusions and Proposed Future Studies

We have developed an empirical model capable of predicting soil moisture from hourly rainfall data. Through the described methodology and the code available in the **Appendix**, we anticipate being able to describe the rainfall-soil moisture relationship for any soil type. This process involves selecting the most appropriate rainfall offset (or a range of offsets if indicated by the results) as well as the corresponding soil moisture readings. Additionally, this research has opened several avenues for future exploration.

Firstly, as this study focused on a single geographical point in the Aburrá Valley, further research is needed to test the methodology (or a modified version of it) in different climates and soil profiles to evaluate its broader applicability.

The question of whether to use nominal readings or Δ readings remains unresolved. Our research found that both approaches yielded promising results. However, we acknowledge that one method may not always produce accurate correlations between rainfall and soil moisture data, depending on specific conditions such as the current soil moisture status.

In **Section 5.3**, we identified an incidental finding related to soil moisture drop-off. After periods of drought, soil gradually drains, reducing its volumetric water content. However, the rate at which this occurs remains unclear. Future studies employing physically based models in addition to our methodology would be valuable in investigating this phenomenon further.

Another notable phenomenon was the negative correlation between maximum Δ readings and a rainfall offset corresponding to the drainage process (i.e., an offset occurring a few hours after the rainfall event). Further exploration of this finding could prove useful in describing soil moisture dynamics for risk assessment purposes, including ways to remove the contamination of maximum value data.

References

Aristizábal, E. (2005). *Function of Permeable Geosynthetics in Artificial Unsaturated Embankments Subjected to Rainfall Infiltration*. The University of Tokyo.

Aristizábal, E., Martínez, H., & Velez, J. (2010). Una revisión sobre el estudio de movimientos en masa detonados por lluvias. *Revista Academia Colombiana de Ciencias Exactas, Físicas y Naturales ISSN 0370-3908*, *34*, 209–227. https://doi.org/10.18257/raccefyn.34(131).2010.2413

Aristizábal, E., Martínez-Carvajal, H., & García-Aristizábal, E. (2017). Modelling Shallow Landslides Triggered by Rainfall in Tropical and Mountainous Basins. In *Advancing Culture of Living with Landslides* (pp. 207–212). Springer International Publishing. https://doi.org/10.1007/978-3-319-53485-5_23

Correa, O., García, F., Bernal, G., Cardona, O. D., & Rodriguez, C. (2020). Early warning system for rainfall-triggered landslides based on real-time probabilistic hazard assessment. *Natural Hazards*, *100*(1), 345–361. https://doi.org/10.1007/s11069-019-03815-w

Crozier, M. J., & Eyles, R. J. (1980). Assessing the Probability of Rapid Mass Movement. In *Third Australia-New Zealand Conference on Geomechanics: Wellington, May 12-16, 1980*. Institution of Professional Engineers New Zealand. https://search.informit.org/doi/10.3316/informit.649149381088316

Garcia-Urquia, E. (2016). Establishing rainfall frequency contour lines as thresholds for rainfall-induced landslides in Tegucigalpa, Honduras, 1980–2005. *Natural Hazards*, *82*(3), 2107–2132. https://doi.org/10.1007/s11069-016-2297-x

Glade, T. (2000). Applying Probability Determination to Refine Landslide-triggering Rainfall Thresholds Using an Empirical "Antecedent Daily Rainfall Model." *Pure and Applied Geophysics*, *157*(6–8), 1059–1079. https://doi.org/10.1007/s000240050017

Gómez, D., Aristizábal, E., & García, E. (2023). *Antecedent rainfall influence on landslides in the Colombian Andes*. *80*.

Gonzalez, F. C. G., Cavacanti, M. do C. R., Nahas Ribeiro, W., Mendonça, M. B. de, & Haddad, A. N. (2024). A systematic review on rainfall thresholds for landslides occurrence. *Heliyon*, *10*(1), e23247. https://doi.org/10.1016/j.heliyon.2023.e23247

Huggel, C., Khabarov, N., Obersteiner, M., & Ramírez, J. M. (2010). Implementation and integrated numerical modeling of a landslide early warning system: a pilot study in Colombia. *Natural Hazards*, *52*(2), 501–518. https://doi.org/10.1007/s11069-009-9393-0

Iadanza, C., Trigila, A., & Napolitano, F. (2016). Identification and characterization of rainfall events responsible for triggering of debris flows and shallow landslides. *Journal of Hydrology*, *541*, 230–245. https://doi.org/10.1016/j.jhydrol.2016.01.018

Jaramillo González, R., Aristizábal, E., García, E., & Marin, R. (2023). Physically-based Model applied to Rainfall Thresholds for Shallow Landslides: Literature review. *Revista de La Asociacion Geologica Argentina*, *80*, 164–178.

Liu, X., Wang, Y., & Leung, A. K. (2023). Numerical investigation of rainfall intensity and duration control of rainfall-induced landslide at a specific slope using slope case histories and actual rainfall records. *Bulletin of Engineering Geology and the Environment*, *82*(8), 333. https://doi.org/10.1007/s10064-023-03359-1

Marc, O., Oliveira, R. A. J., Gosset, M., Emberson, R., & Malet, J.-P. (2022). Global Assessment of the Capability of Satellite Precipitation Products to Retrieve Landslide-Triggering Extreme Rainfall Events. *Earth Interactions*, *26*(1), 122–138. https://doi.org/10.1175/EI-D-21-0022.1

Martinović, K., Gavin, K., Reale, C., & Mangan, C. (2018). Rainfall thresholds as a landslide indicator for engineered slopes on the Irish Rail network. *Geomorphology*, *306*, 40–50. https://doi.org/10.1016/j.geomorph.2018.01.006

Melillo, M., Brunetti, M. T., Peruccacci, S., Gariano, S. L., & Guzzetti, F. (2015). An algorithm for the objective reconstruction of rainfall events responsible for landslides. *Landslides*, *12*(2), 311–320. https://doi.org/10.1007/s10346-014-0471-3

Oliveira, E. do P., Acevedo, A. M. G., Moreira, V. S., Faro, V. P., & Kormann, A. C. M. (2022). The Key Parameters Involved in a Rainfall-Triggered Landslide. *Water*, *14*(21), 3561. https://doi.org/10.3390/w14213561

Pradhan, A. M. S., Lee, J.-M., & Kim, Y.-T. (2019). Semi-quantitative method to identify the vulnerable areas in terms of building aggregation for probable landslide runout at the regional scale: a case study from Soacha Province, Colombia. *Bulletin of Engineering Geology and the Environment*, *78*(8), 5745–5762. https://doi.org/10.1007/s10064-019-01533-y

Segoni, S., Piciullo, L., & Gariano, S. L. (2018). A review of the recent literature on rainfall thresholds for landslide occurrence. *Landslides*, *15*(8), 1483–1501. https://doi.org/10.1007/s10346-018-0966-4

SIATA. (2024). *Sistema de Alerta Temprana del Valle de Aburrá*. https://siata.gov.co/

Soto, J., Palenzuela, J. A., Galve, J. P., Luque, J. A., Azañón, J. M., Tamay, J., & Irigaray, C. (2019). Estimation of empirical rainfall thresholds for landslide triggering using partial duration series and their relation with climatic cycles. An application in southern Ecuador. *Bulletin of Engineering Geology and the Environment*, *78*(3), 1971–1987. https://doi.org/10.1007/s10064-017-1216-z

Wu, Y., Lan, H., Gao, X., Li, L., & Yang, Z. (2015). A simplified physically based coupled rainfall threshold model for triggering landslides. *Engineering Geology*, *195*, 63–69. https://doi.org/10.1016/j.enggeo.2015.05.022

Appendix

A1. Daily data processing (Python code)

```
# Initialization
import pandas as pd
rf = pd.read_csv('Station 43.csv')
sm = pd.read_csv('Station 396.csv')

# Quality filtering for rainfall
rf.loc[rf['Calidad'] == 2512, ['P2']] = float('nan')
rf.loc[rf['Calidad'] == 2511, ['P1']] = float('nan')
rf.loc[rf['Calidad'] == 2510, ['P1', 'P2']] = float('nan')
rf.loc[rf['Calidad'] == 1512, ['P2']] = float('nan')
rf.loc[rf['Calidad'] == 1511, ['P1']] = float('nan')
rf.loc[rf['Calidad'] == 1510, ['P1', 'P2']] = float('nan')
rf.loc[rf['Calidad'] == 251, ['P1', 'P2']] = float('nan')
rf.loc[rf['Calidad'] == 151, ['P1', 'P2']] = float('nan')
rf.loc[rf['P2'] == -999, ['P2']] = float('nan')
rf.loc[rf['P1'] == -999, ['P1']] = float('nan')
del rf['Calidad']

# Quality filtering for soil moisture
sm.loc[sm['calidad'] == 2528, ['h2', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 2527, ['h1', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 2526, ['h1', 'h2']] = float('nan')
sm.loc[sm['calidad'] == 2523, ['h3']] = float('nan')
sm.loc[sm['calidad'] == 2522, ['h2']] = float('nan')
sm.loc[sm['calidad'] == 2521, ['h1']] = float('nan')
sm.loc[sm['calidad'] == 1528, ['h2', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 1527, ['h1', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 1526, ['h1', 'h2']] = float('nan')
sm.loc[sm['calidad'] == 1523, ['h3']] = float('nan')
sm.loc[sm['calidad'] == 1522, ['h2']] = float('nan')
sm.loc[sm['calidad'] == 1521, ['h1']] = float('nan')
sm.loc[sm['calidad'] == 251, ['h1', 'h2', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 151, ['h1', 'h2', 'h3']] = float('nan')
sm.loc[sm['h3'] == -999, ['h3']] = float('nan')
sm.loc[sm['h2'] == -999, ['h2']] = float('nan')
sm.loc[sm['h1'] == -999, ['h1']] = float('nan')
del sm['calidad']
```

```
# Choosing maximum rainfall, converting to inches, and cutting HH:MM:SS
rf['RF'] = rf[['P1', 'P2']].max(axis=1)
del rf['P1']
del rf['P2']
rf['RF'] = rf['RF'] / 25.4
rf['RF'] = rf['RF'].round(2)
rf = rf.rename(columns={'fecha_hora': 'Date'})
rf['Date'] = rf['Date'].str[:-9]
rf = rf[(rf['Date'] >= '2020-03-01') & (rf['Date'] <= '2022-03-31')]

# Rounding and splitting VWC data, as well as cutting HH:MM:SS
sm['h1'] = sm['h1'].round(1)
sm['h2'] = sm['h2'].round(1)
sm['h3'] = sm['h3'].round(1)
sm = sm.rename(columns={'h1': 'VWC1'})
sm = sm.rename(columns={'h2': 'VWC2'})
sm = sm.rename(columns={'h3': 'VWC3'})
sm = sm.rename(columns={'fecha_hora': 'Date'})
sm['Date'] = sm['Date'].str[:-9]
sm = sm[(sm['Date'] >= '2020-04-01') & (sm['Date'] <= '2022-03-31')]
sm1 = sm.copy()
sm2 = sm.copy()
sm3 = sm.copy()
sm1 = sm1.drop(columns=['VWC2', 'VWC3'])
sm2 = sm2.drop(columns=['VWC1', 'VWC3'])
sm3 = sm3.drop(columns=['VWC1', 'VWC2'])

# Count entries per day and cut days with less than 75% entries (RF)
rf = rf.dropna(subset=['RF'])
count_rf = rf.groupby('Date')['Date'].count()
remove_rf = count_rf[count_rf < 1080].index.tolist()
rf = rf[~rf['Date'].isin(remove_rf)]

# Count entries per day and cut days with less than 40% entries (VWC)
sm1 = sm1.dropna(subset=['VWC1'])
sm2 = sm2.dropna(subset=['VWC2'])
sm3 = sm3.dropna(subset=['VWC3'])
count_sm1 = sm1.groupby('Date')['Date'].count()
count_sm2 = sm2.groupby('Date')['Date'].count()
count_sm3 = sm3.groupby('Date')['Date'].count()
remove_sm1 = count_sm1[count_sm1 < 576].index.tolist()
```

```
remove_sm2 = count_sm2[count_sm2 < 576].index.tolist()
remove_sm3 = count_sm3[count_sm3 < 576].index.tolist()
sm1 = sm1[~sm1['Date'].isin(remove_sm1)]
sm2 = sm2[~sm2['Date'].isin(remove_sm2)]
sm3 = sm3[~sm3['Date'].isin(remove_sm3)]

# Sum rainfall data to get total daily rainfall
rf_grouped = rf.groupby('Date').agg({'RF': 'sum'})
rf_grouped['RF'] = rf_grouped['RF'].round(2)

# Manual fix for rainfall
rf_grouped.loc['2021-09-23', 'RF'] = 1.04

# Average soil moisture data to get average soil moisture
sm1_grouped = sm1.groupby('Date').agg({'VWC1': 'mean'})
sm2_grouped = sm2.groupby('Date').agg({'VWC2': 'mean'})
sm3_grouped = sm3.groupby('Date').agg({'VWC3': 'mean'})
sm1_grouped['VWC1'] = sm1_grouped['VWC1'].round(1)
sm2_grouped['VWC2'] = sm2_grouped['VWC2'].round(1)
sm3_grouped['VWC3'] = sm3_grouped['VWC3'].round(1)

# Data copied from January 2021, which also lacked rain
sm1_grouped.loc['2022-01-24', 'VWC1'] = 43.8
sm1_grouped.loc['2022-01-25', 'VWC1'] = 43.7
sm1_grouped.loc['2022-01-26', 'VWC1'] = 43.5
sm1_grouped.loc['2022-01-27', 'VWC1'] = 43.3
sm1_grouped.loc['2022-01-28', 'VWC1'] = 43.1
sm1_grouped.loc['2022-01-29', 'VWC1'] = 42.9
sm1_grouped.loc['2022-01-30', 'VWC1'] = 42.7
sm1_grouped.loc['2022-01-31', 'VWC1'] = 42.5
sm1_grouped.loc['2022-02-01', 'VWC1'] = 42.2
sm1_grouped.loc['2022-02-02', 'VWC1'] = 42.1
sm2_grouped.loc['2022-01-24', 'VWC2'] = 41.3
sm2_grouped.loc['2022-01-25', 'VWC2'] = 40.9
sm2_grouped.loc['2022-01-26', 'VWC2'] = 40.7
sm2_grouped.loc['2022-01-27', 'VWC2'] = 40.5
sm2_grouped.loc['2022-01-28', 'VWC2'] = 40.2
sm2_grouped.loc['2022-01-29', 'VWC2'] = 39.9
sm2_grouped.loc['2022-01-30', 'VWC2'] = 39.7
sm2_grouped.loc['2022-01-31', 'VWC2'] = 39.5
sm2_grouped.loc['2022-02-01', 'VWC2'] = 39.1
sm2_grouped.loc['2022-02-02', 'VWC2'] = 38.8
```

```
sm3_grouped.loc['2022-01-24', 'VWC3'] = 44.4
sm3_grouped.loc['2022-01-25', 'VWC3'] = 43.7
sm3_grouped.loc['2022-01-26', 'VWC3'] = 43.0
sm3_grouped.loc['2022-01-27', 'VWC3'] = 42.3
sm3_grouped.loc['2022-01-28', 'VWC3'] = 41.5
sm3_grouped.loc['2022-01-29', 'VWC3'] = 40.8
sm3_grouped.loc['2022-01-30', 'VWC3'] = 40.3
sm3_grouped.loc['2022-01-31', 'VWC3'] = 40.0
sm3_grouped.loc['2022-02-01', 'VWC3'] = 39.5
sm3_grouped.loc['2022-02-02', 'VWC3'] = 38.7

# Linearly interpolated
sm1_grouped.loc['2020-09-22', 'VWC1'] = 44.6
sm1_grouped.loc['2020-09-23', 'VWC1'] = 44.4
sm1_grouped.loc['2020-09-24', 'VWC1'] = 44.2
sm2_grouped.loc['2020-09-22', 'VWC2'] = 43.7
sm2_grouped.loc['2020-09-23', 'VWC2'] = 43.0
sm2_grouped.loc['2020-09-24', 'VWC2'] = 42.3
sm3_grouped.loc['2020-09-22', 'VWC3'] = 48.7
sm3_grouped.loc['2020-09-23', 'VWC3'] = 48.3
sm3_grouped.loc['2020-09-24', 'VWC3'] = 47.9
sm1_grouped.loc['2021-02-11', 'VWC1'] = 40.4
sm1_grouped.loc['2021-02-12', 'VWC1'] = 40.7
sm1_grouped.loc['2021-02-13', 'VWC1'] = 41.1
sm2_grouped.loc['2021-02-11', 'VWC2'] = 35.9
sm2_grouped.loc['2021-02-12', 'VWC2'] = 36.5
sm2_grouped.loc['2021-02-13', 'VWC2'] = 37.2
sm3_grouped.loc['2021-02-11', 'VWC3'] = 42.2
sm3_grouped.loc['2021-02-12', 'VWC3'] = 43.0
sm3_grouped.loc['2021-02-13', 'VWC3'] = 43.8

# Create a dataframe for every day
ranges = pd.date_range(start='03/01/2020', end='03/31/2022', freq='D')
dates = pd.DataFrame(ranges, columns=['Date'])
dates['Date'] = dates['Date'].dt.strftime('%Y-%m-%d')

# Merge all files
df_merged = pd.merge(dates, rf_grouped, on='Date', how='left')
df_merged = pd.merge(df_merged, sm1_grouped, on='Date', how='left')
df_merged = pd.merge(df_merged, sm2_grouped, on='Date', how='left')
df_merged = pd.merge(df_merged, sm3_grouped, on='Date', how='left')
df_merged['RF'] = round(df_merged['RF'] * 25.4, 3)
```

```
# Calculating VWC differences
df_merged['Diff. (VWC1)'] = round(df_merged['VWC1'].diff(), 1)
df_merged['Diff. (VWC2)'] = round(df_merged['VWC2'].diff(), 1)
df_merged['Diff. (VWC3)'] = round(df_merged['VWC3'].diff(), 1)
factor = 0.8
df_merged['ARFY'] = round(df_merged['RF'] + df_merged['RF'].shift(1) * factor,
0)
df_merged['ARF3'] = round(df_merged['RF'] + df_merged['RF'].shift(1) * factor
+ df_merged['RF'].shift(2) * factor * factor, 0)
df_merged = df_merged[df_merged['Date'] >= '2020-04-01']

# Calculating antecedent rainfall index and checking correlation
threshold = 16
df_merged['RF/VWC1'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC1)'] >
0)) | ((df_merged['RF'] < threshold) & (df_merged['Diff. (VWC1)'] <= 0)),
'RF/VWC1'] = 1
df_merged['RF/VWC2'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC2)'] >
0)) | ((df_merged['RF'] < threshold) & (df_merged['Diff. (VWC2)'] <= 0)),
'RF/VWC2'] = 1
df_merged['RF/VWC3'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC3)'] >
0)) | ((df_merged['RF'] < threshold) & (df_merged['Diff. (VWC3)'] <= 0)),
'RF/VWC3'] = 1
df_merged['ARFY/VWC1'] = 0
df_merged.loc[((df_merged['ARFY'] >= threshold) & (df_merged['Diff. (VWC1)'] >
0)) | ((df_merged['ARFY'] < threshold) & (df_merged['Diff. (VWC1)'] <= 0)),
'ARFY/VWC1'] = 1
df_merged['ARFY/VWC2'] = 0
df_merged.loc[((df_merged['ARFY'] >= threshold) & (df_merged['Diff. (VWC2)'] >
0)) | ((df_merged['ARFY'] < threshold) & (df_merged['Diff. (VWC2)'] <= 0)),
'ARFY/VWC2'] = 1
df_merged['ARFY/VWC3'] = 0
df_merged.loc[((df_merged['ARFY'] >= threshold) & (df_merged['Diff. (VWC3)'] >
0)) | ((df_merged['ARFY'] < threshold) & (df_merged['Diff. (VWC3)'] <= 0)),
'ARFY/VWC3'] = 1
df_merged['ARF3/VWC1'] = 0
df_merged.loc[((df_merged['ARF3'] >= threshold) & (df_merged['Diff. (VWC1)'] >
0)) | ((df_merged['ARF3'] < threshold) & (df_merged['Diff. (VWC1)'] <= 0)),
'ARF3/VWC1'] = 1
```

```
df_merged['ARF3/VWC2'] = 0
df_merged.loc[((df_merged['ARF3'] >= threshold) & (df_merged['Diff. (VWC2)'] >
0)) | ((df_merged['ARF3'] < threshold) & (df_merged['Diff. (VWC2)'] <= 0)),
'ARF3/VWC2'] = 1
df_merged['ARF3/VWC3'] = 0
df_merged.loc[((df_merged['ARF3'] >= threshold) & (df_merged['Diff. (VWC3)'] >
0)) | ((df_merged['ARF3'] < threshold) & (df_merged['Diff. (VWC3)'] <= 0)),
'ARF3/VWC3'] = 1

corr1 = df_merged['RF/VWC1'].sum()
corr2 = df_merged['RF/VWC2'].sum()
corr3 = df_merged['RF/VWC3'].sum()
corr4 = df_merged['ARFY/VWC1'].sum()
corr5 = df_merged['ARFY/VWC2'].sum()
corr6 = df_merged['ARFY/VWC3'].sum()
corr7 = df_merged['ARF3/VWC1'].sum()
corr8 = df_merged['ARF3/VWC2'].sum()
corr9 = df_merged['ARF3/VWC3'].sum()

print(f"Rainfall matches VWC1 with {round(corr1/730*100, 1)}%, VWC2 with
{round(corr2/730*100, 1)}%, and VWC3 with {round(corr3/730*100, 1)}%")
print(f"Antecedent Rainfall (Yesterday) matches VWC1 with {round(corr4/730*100,
1)}%, VWC2 with {round(corr5/730*100, 1)}%, and VWC3 with {round(corr6/730*100,
1)}%")
print(f"Antecedent Rainfall (Three Days) matches VWC1 with
{round(corr7/730*100, 1)}%, VWC2 with {round(corr8/730*100, 1)}%, and VWC3 with
{round(corr9/730*100, 1)}%")

# Calculating antecedent rainfall index and checking correlation with rainy
days
threshold = 16
df_merged['RF/VWC1'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC1)'] >
0)), 'RF/VWC1'] = 1
df_merged['RF/VWC2'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC2)'] >
0)), 'RF/VWC2'] = 1
df_merged['RF/VWC3'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC3)'] >
0)), 'RF/VWC3'] = 1
df_merged['ARFY/VWC1'] = 0
```

```
df_merged.loc[((df_merged['ARFY'] >= threshold) & (df_merged['Diff. (VWC1)'] >
0)), 'ARFY/VWC1'] = 1
df_merged['ARFY/VWC2'] = 0
df_merged.loc[((df_merged['ARFY'] >= threshold) & (df_merged['Diff. (VWC2)'] >
0)), 'ARFY/VWC2'] = 1
df_merged['ARFY/VWC3'] = 0
df_merged.loc[((df_merged['ARFY'] >= threshold) & (df_merged['Diff. (VWC3)'] >
0)), 'ARFY/VWC3'] = 1
df_merged['ARF3/VWC1'] = 0
df_merged.loc[((df_merged['ARF3'] >= threshold) & (df_merged['Diff. (VWC1)'] >
0)), 'ARF3/VWC1'] = 1
df_merged['ARF3/VWC2'] = 0
df_merged.loc[((df_merged['ARF3'] >= threshold) & (df_merged['Diff. (VWC2)'] >
0)), 'ARF3/VWC2'] = 1
df_merged['ARF3/VWC3'] = 0
df_merged.loc[((df_merged['ARF3'] >= threshold) & (df_merged['Diff. (VWC3)'] >
0)), 'ARF3/VWC3'] = 1

corr1 = df_merged['RF/VWC1'].sum()
corr2 = df_merged['RF/VWC2'].sum()
corr3 = df_merged['RF/VWC3'].sum()
corr4 = df_merged['ARFY/VWC1'].sum()
corr5 = df_merged['ARFY/VWC2'].sum()
corr6 = df_merged['ARFY/VWC3'].sum()
corr7 = df_merged['ARF3/VWC1'].sum()
corr8 = df_merged['ARF3/VWC2'].sum()
corr9 = df_merged['ARF3/VWC3'].sum()

df_merged['Rainy Day'] = 0
df_merged.loc[(df_merged['RF'] >= threshold), 'Rainy Day'] = 1
rainy_days = df_merged['Rainy Day'].sum()
df_merged['Rainy Day ARFY'] = 0
df_merged.loc[(df_merged['ARFY'] >= threshold), 'Rainy Day ARFY'] = 1
rainy_daysy = df_merged['Rainy Day ARFY'].sum()
df_merged['Rainy Day ARF3'] = 0
df_merged.loc[(df_merged['ARF3'] >= threshold), 'Rainy Day ARF3'] = 1
rainy_days3 = df_merged['Rainy Day ARF3'].sum()

print(f"\nRainfall matches VWC1 with {round(corr1/rainy_days*100, 1)}%, VWC2
with {round(corr2/rainy_days*100, 1)}%, and VWC3 with
{round(corr3/rainy_days*100, 1)}%")
```

```
print(f"Antecedent Rainfall (Yesterday) matches VWC1 with {round(corr4/rainy_daysy*100, 1)}%, VWC2 with {round(corr5/rainy_daysy*100, 1)}%, and VWC3 with {round(corr6/rainy_daysy*100, 1)}%")
print(f"Antecedent Rainfall (Three Days) matches VWC1 with {round(corr7/rainy_days3*100, 1)}%, VWC2 with {round(corr8/rainy_days3*100, 1)}%, and VWC3 with {round(corr9/rainy_days3*100, 1)}%")

# Calculating the daily soil moisture drop-off
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC1)'] <= 0)), 'Diff. (VWC1)'].mean()
print(f"The average soil moisture (VWC1) loss on dry days: {round(dryloss, 2)}%")
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC2)'] <= 0)), 'Diff. (VWC2)'].mean()
print(f"The average soil moisture (VWC2) loss on dry days: {round(dryloss, 2)}%")
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC3)'] <= 0)), 'Diff. (VWC3)'].mean()
print(f"The average soil moisture (VWC3) loss on dry days: {round(dryloss, 2)}%")

dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC1)'].mean()
print(f"The average soil moisture (VWC1) loss on dry days: {round(dryloss, 2)}%")
dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC2)'].mean()
print(f"The average soil moisture (VWC2) loss on dry days: {round(dryloss, 2)}%")
dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC3)'].mean()
print(f"The average soil moisture (VWC3) loss on dry days: {round(dryloss, 2)}%")

# Delete noise
del df_merged['Rainy Day']
del df_merged['Rainy Day ARFY']
del df_merged['Rainy Day ARF3']
del df_merged['RF/VWC1']
del df_merged['RF/VWC2']
del df_merged['RF/VWC3']
del df_merged['ARFY/VWC1']
del df_merged['ARFY/VWC2']
del df_merged['ARFY/VWC3']
del df_merged['ARF3/VWC1']
```

```
del df_merged['ARF3/VWC2']
del df_merged['ARF3/VWC3']

# Conversion to CSV
df_merged.to_csv('Daily 43 & 396.csv', encoding='utf-8-sig', index=False)
```

A2. Hourly data processing (Python code)

```
# Initialization
import pandas as pd
rf = pd.read_csv('Station 43.csv')
sm = pd.read_csv('Station 396.csv')

# Quality filtering for rainfall
rf.loc[rf['Calidad'] == 2512, ['P2']] = float('nan')
rf.loc[rf['Calidad'] == 2511, ['P1']] = float('nan')
rf.loc[rf['Calidad'] == 2510, ['P1', 'P2']] = float('nan')
rf.loc[rf['Calidad'] == 1512, ['P2']] = float('nan')
rf.loc[rf['Calidad'] == 1511, ['P1']] = float('nan')
rf.loc[rf['Calidad'] == 1510, ['P1', 'P2']] = float('nan')
rf.loc[rf['Calidad'] == 251, ['P1', 'P2']] = float('nan')
rf.loc[rf['Calidad'] == 151, ['P1', 'P2']] = float('nan')
rf.loc[rf['P2'] == -999, ['P2']] = float('nan')
rf.loc[rf['P1'] == -999, ['P1']] = float('nan')
del rf['Calidad']

# Quality filtering for soil moisture
sm.loc[sm['calidad'] == 2528, ['h2', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 2527, ['h1', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 2526, ['h1', 'h2']] = float('nan')
sm.loc[sm['calidad'] == 2523, ['h3']] = float('nan')
sm.loc[sm['calidad'] == 2522, ['h2']] = float('nan')
sm.loc[sm['calidad'] == 2521, ['h1']] = float('nan')
sm.loc[sm['calidad'] == 1528, ['h2', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 1527, ['h1', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 1526, ['h1', 'h2']] = float('nan')
sm.loc[sm['calidad'] == 1523, ['h3']] = float('nan')
sm.loc[sm['calidad'] == 1522, ['h2']] = float('nan')
sm.loc[sm['calidad'] == 1521, ['h1']] = float('nan')
sm.loc[sm['calidad'] == 251, ['h1', 'h2', 'h3']] = float('nan')
sm.loc[sm['calidad'] == 151, ['h1', 'h2', 'h3']] = float('nan')
```

```
sm.loc[sm['h3'] == -999, ['h3']] = float('nan')
sm.loc[sm['h2'] == -999, ['h2']] = float('nan')
sm.loc[sm['h1'] == -999, ['h1']] = float('nan')
del sm['calidad']

# Choosing maximum rainfall, converting to inches, and cutting MM:SS
rf['RF'] = rf[['P1', 'P2']].max(axis=1)
del rf['P1']
del rf['P2']
rf['RF'] = rf['RF'] / 25.4
rf['RF'] = rf['RF'].round(2)
rf = rf.rename(columns={'fecha_hora': 'Date'})
rf['Date'] = rf['Date'].str[:-6]
rf = rf[(rf['Date'] >= '2020-03-31 00') & (rf['Date'] <= '2022-03-31 23')]

# Rounding and splitting VWC data, as well as cutting MM:SS
sm['h1'] = sm['h1'].round(1)
sm['h2'] = sm['h2'].round(1)
sm['h3'] = sm['h3'].round(1)
sm = sm.rename(columns={'h1': 'VWC1'})
sm = sm.rename(columns={'h2': 'VWC2'})
sm = sm.rename(columns={'h3': 'VWC3'})
sm = sm.rename(columns={'fecha_hora': 'Date'})
sm['Date'] = sm['Date'].str[:-6]
sm = sm[(sm['Date'] >= '2020-03-31 00') & (sm['Date'] <= '2022-03-31 23')]
sm1 = sm.copy()
sm2 = sm.copy()
sm3 = sm.copy()
sm1 = sm1.drop(columns=['VWC2', 'VWC3'])
sm2 = sm2.drop(columns=['VWC1', 'VWC3'])
sm3 = sm3.drop(columns=['VWC1', 'VWC2'])

# Count entries per day and cut hours with less than 75% entries (RF)
rf = rf.dropna(subset=['RF'])
count_rf = rf.groupby('Date')['Date'].count()
remove_rf = count_rf[count_rf < 45].index.tolist()
rf = rf[~rf['Date'].isin(remove_rf)]

# Count entries per day and cut days with less than 40% entries (VWC)
sm1 = sm1.dropna(subset=['VWC1'])
sm2 = sm2.dropna(subset=['VWC2'])
sm3 = sm3.dropna(subset=['VWC3'])
```

```
count_sm1 = sm1.groupby('Date')['Date'].count()
count_sm2 = sm2.groupby('Date')['Date'].count()
count_sm3 = sm3.groupby('Date')['Date'].count()
remove_sm1 = count_sm1[count_sm1 < 24].index.tolist()
remove_sm2 = count_sm2[count_sm2 < 24].index.tolist()
remove_sm3 = count_sm3[count_sm3 < 24].index.tolist()
sm1 = sm1[~sm1['Date'].isin(remove_sm1)]
sm2 = sm2[~sm2['Date'].isin(remove_sm2)]
sm3 = sm3[~sm3['Date'].isin(remove_sm3)]

# Sum rainfall data to get total daily rainfall
rf_grouped = rf.groupby('Date').agg({'RF': 'sum'})
rf_grouped['RF'] = rf_grouped['RF'].round(2)

# Average soil moisture data to get average soil moisture
sm1_grouped = sm1.groupby('Date').agg({'VWC1': 'mean'})
sm2_grouped = sm2.groupby('Date').agg({'VWC2': 'mean'})
sm3_grouped = sm3.groupby('Date').agg({'VWC3': 'mean'})
sm1_grouped['VWC1'] = sm1_grouped['VWC1'].round(1)
sm2_grouped['VWC2'] = sm2_grouped['VWC2'].round(1)
sm3_grouped['VWC3'] = sm3_grouped['VWC3'].round(1)

# Create a dataframe for every hour
ranges = pd.date_range(start='03/31/2020 00', end='03/31/2022 23', freq='h')
dates = pd.DataFrame(ranges, columns=['Date'])
dates['Date'] = dates['Date'].dt.strftime('%Y-%m-%d %H')

# Merge all files
df_merged = pd.merge(dates, rf_grouped, on='Date', how='left')
df_merged = pd.merge(df_merged, sm1_grouped, on='Date', how='left')
df_merged = pd.merge(df_merged, sm2_grouped, on='Date', how='left')
df_merged = pd.merge(df_merged, sm3_grouped, on='Date', how='left')
df_merged['RF'] = round(df_merged['RF'] * 25.4, 3)

# Calculating VWC differences and RF offsets
df_merged['Diff. (VWC1)'] = round(df_merged['VWC1'].diff(), 1)
df_merged['Diff. (VWC2)'] = round(df_merged['VWC2'].diff(), 1)
df_merged['Diff. (VWC3)'] = round(df_merged['VWC3'].diff(), 1)
for x in range(1, 25):
  df_merged[f'RF-{x}'] = df_merged['RF'].shift(x)
df_merged = df_merged[df_merged['Date'] >= '2020-04-01 00']
```

```
# Calculating offset rainfall and checking correlation
df_merged['RF/VWC1'] = 0
df_merged.loc[((df_merged['RF'] >= 1) & (df_merged['Diff. (VWC1)'] > 0)) |
((df_merged['RF'] < 1) & (df_merged['Diff. (VWC1)'] <= 0)), 'RF/VWC1'] = 1
df_merged['RF/VWC2'] = 0
df_merged.loc[((df_merged['RF'] >= 1) & (df_merged['Diff. (VWC2)'] > 0)) |
((df_merged['RF'] < 1) & (df_merged['Diff. (VWC2)'] <= 0)), 'RF/VWC2'] = 1
df_merged['RF/VWC3'] = 0
df_merged.loc[((df_merged['RF'] >= 1) & (df_merged['Diff. (VWC3)'] > 0)) |
((df_merged['RF'] < 1) & (df_merged['Diff. (VWC3)'] <= 0)), 'RF/VWC3'] = 1

corr1 = df_merged['RF/VWC1'].sum()
corr2 = df_merged['RF/VWC2'].sum()
corr3 = df_merged['RF/VWC3'].sum()

for x in range(1, 25):
  df_merged[f'RF-{x}/VWC1'] = 0
  df_merged.loc[((df_merged[f'RF-{x}'] >= 1) & (df_merged['Diff. (VWC1)'] > 0))
| ((df_merged[f'RF-{x}'] < 1) & (df_merged['Diff. (VWC1)'] <= 0)), f'RF-
{x}/VWC1'] = 1
  df_merged[f'RF-{x}/VWC2'] = 0
  df_merged.loc[((df_merged[f'RF-{x}'] >= 1) & (df_merged['Diff. (VWC2)'] > 0))
| ((df_merged[f'RF-{x}'] < 1) & (df_merged['Diff. (VWC2)'] <= 0)), f'RF-
{x}/VWC2'] = 1
  df_merged[f'RF-{x}/VWC3'] = 0
  df_merged.loc[((df_merged[f'RF-{x}'] >= 1) & (df_merged['Diff. (VWC3)'] > 0))
| ((df_merged[f'RF-{x}'] < 1) & (df_merged['Diff. (VWC3)'] <= 0)), f'RF-
{x}/VWC3'] = 1

corr4 = np.zeros(25)
corr5 = np.zeros(25)
corr6 = np.zeros(25)

for x in range(1,25):
  corr4[x] = df_merged[f'RF-{x}/VWC1'].sum()
  corr5[x] = df_merged[f'RF-{x}/VWC2'].sum()
  corr6[x] = df_merged[f'RF-{x}/VWC3'].sum()

print(f"Rainfall matches VWC1 with {round(corr1/17520*100, 1)}%, VWC2 with
{round(corr2/17520*100, 1)}%, and VWC3 with {round(corr3/17520*100, 1)}%")
for x in range (1,25):
```

```
  print(f"Offset Rainfall (RF-{x}) matches VWC1 with {round(corr4[x]/17520*100, 1)}%, VWC2 with {round(corr5[x]/17520*100, 1)}%, and VWC3 with {round(corr6[x]/17520*100, 1)}%")

# Calculating offset rainfall and checking correlation with rainy hours
threshold = 10
df_merged['RF/VWC1'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC1)'] > 0)), 'RF/VWC1'] = 1
df_merged['RF/VWC2'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC2)'] > 0)), 'RF/VWC2'] = 1
df_merged['RF/VWC3'] = 0
df_merged.loc[((df_merged['RF'] >= threshold) & (df_merged['Diff. (VWC3)'] > 0)), 'RF/VWC3'] = 1

corr1 = df_merged['RF/VWC1'].sum()
corr2 = df_merged['RF/VWC2'].sum()
corr3 = df_merged['RF/VWC3'].sum()

for x in range(1, 25):
  df_merged[f'RF-{x}/VWC1'] = 0
  df_merged.loc[((df_merged[f'RF-{x}'] >= threshold) & (df_merged['Diff. (VWC1)'] > 0)), f'RF-{x}/VWC1'] = 1
  df_merged[f'RF-{x}/VWC2'] = 0
  df_merged.loc[((df_merged[f'RF-{x}'] >= threshold) & (df_merged['Diff. (VWC2)'] > 0)), f'RF-{x}/VWC2'] = 1
  df_merged[f'RF-{x}/VWC3'] = 0
  df_merged.loc[((df_merged[f'RF-{x}'] >= threshold) & (df_merged['Diff. (VWC3)'] > 0)), f'RF-{x}/VWC3'] = 1

corr4 = np.zeros(25)
corr5 = np.zeros(25)
corr6 = np.zeros(25)

for x in range(1,25):
  corr4[x] = df_merged[f'RF-{x}/VWC1'].sum()
  corr5[x] = df_merged[f'RF-{x}/VWC2'].sum()
  corr6[x] = df_merged[f'RF-{x}/VWC3'].sum()

df_merged['Rainy Hour'] = 0
df_merged.loc[(df_merged['RF'] >= threshold), 'Rainy Hour'] = 1
```

```
rainy_hours = df_merged['Rainy Hour'].sum()

print(f"Rainfall matches VWC1 with {round(corr1/rainy_hours*100, 1)}%, VWC2 with {round(corr2/rainy_hours*100, 1)}%, and VWC3 with {round(corr3/rainy_hours*100, 1)}%")
for x in range (1,25):
  print(f"Offset Rainfall (RF-{x}) matches VWC1 with {round(corr4[x]/rainy_hours*100, 1)}%, VWC2 with {round(corr5[x]/rainy_hours*100, 1)}%, and VWC3 with {round(corr6[x]/rainy_hours*100, 1)}%")

# Calculating the minutely soil moisture drop-off
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC1)'] <= 0)), 'Diff. (VWC1)'].mean()
print(f"The average soil moisture (VWC1) loss on dry hours: {round(dryloss, 4)}%")
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC2)'] <= 0)), 'Diff. (VWC2)'].mean()
print(f"The average soil moisture (VWC2) loss on dry hours: {round(dryloss, 4)}%")
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC3)'] <= 0)), 'Diff. (VWC3)'].mean()
print(f"The average soil moisture (VWC3) loss on dry hours: {round(dryloss, 4)}%")

dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC1)'].mean()
print(f"The average soil moisture (VWC1) loss on dry hours: {round(dryloss, 4)}%")
dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC2)'].mean()
print(f"The average soil moisture (VWC2) loss on dry hours: {round(dryloss, 4)}%")
dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC3)'].mean()
print(f"The average soil moisture (VWC3) loss on dry hours: {round(dryloss, 4)}%")

dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC1)'] <= 0)), 'Diff. (VWC1)'].median()
print(f"The median soil moisture (VWC1) loss on dry hours: {round(dryloss, 4)}%")
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC2)'] <= 0)), 'Diff. (VWC2)'].median()
```

```
print(f"The median soil moisture (VWC2) loss on dry hours: {round(dryloss,
4)}%")
dryloss = df_merged.loc[((df_merged['RF'] == 0) & (df_merged['Diff. (VWC3)'] <=
0)), 'Diff. (VWC3)'].median()
print(f"The median soil moisture (VWC3) loss on dry hours: {round(dryloss,
4)}%")

dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC1)'].median()
print(f"The median soil moisture (VWC1) loss on dry hours: {round(dryloss,
4)}%")
dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC2)'].median()
print(f"The median soil moisture (VWC2) loss on dry hours: {round(dryloss,
4)}%")
dryloss = df_merged.loc[(df_merged['RF'] == 0), 'Diff. (VWC3)'].median()
print(f"The median soil moisture (VWC3) loss on dry hours: {round(dryloss,
4)}%")

# Delete noise
del df_merged['Rainy Hour']
del df_merged['RF/VWC1']
del df_merged['RF/VWC2']
del df_merged['RF/VWC3']
for x in range(6, 25):
  del df_merged[f'RF-{x}/VWC1']
  del df_merged[f'RF-{x}/VWC2']
  del df_merged[f'RF-{x}/VWC3']

# Conversion to CSV
df_merged.to_csv('Hourly 43 & 396.csv', encoding='utf-8-sig', index=False)
```

A3. Further statistical analysis for hourly data (Python code)

```
# Initialization
import pandas as pd
data = pd.read_csv('Hourly 43 & 396.csv')

# Cutting rainfall offsets above 5 hours
for x in range(6, 25):
  del data[f'RF-{x}']

# Marking rows in range
```

```
t1 = 0 # Choose lower bound
t2 = 0 # Choose upper bound
data['X-0'] = 0
data.loc[((data['RF'] >= t1) & (data['RF'] <= t2)), 'X-0'] = 1
for x in range(1, 6):
  data[f'X-{x}'] = 0
  data.loc[((data[f'RF-{x}'] >= t1) & (data[f'RF-{x}'] <= t2)), f'X-{x}'] = 1

# Printing data for selected range
for x in range (6):
  print(f"{round(data.loc[data[f'X-{x}'] == 1, 'VWC1'].max(), 1)},{round(data.loc[data[f'X-{x}'] == 1, 'VWC1'].mean(), 1)},{round(data.loc[data[f'X-{x}'] == 1, 'VWC1'].median(), 1)}")
  print(f"{round(data.loc[data[f'X-{x}'] == 1, 'VWC2'].max(), 1)},{round(data.loc[data[f'X-{x}'] == 1, 'VWC2'].mean(), 1)},{round(data.loc[data[f'X-{x}'] == 1, 'VWC2'].median(), 1)}")
  print(f"{round(data.loc[data[f'X-{x}'] == 1, 'VWC3'].max(), 1)},{round(data.loc[data[f'X-{x}'] == 1, 'VWC3'].mean(), 1)},{round(data.loc[data[f'X-{x}'] == 1, 'VWC3'].median(), 1)}")
  print(f"{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC1)'].max(), 2)},{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC1)'].mean(), 2)},{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC1)'].median(), 2)}")
  print(f"{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC2)'].max(), 2)},{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC2)'].mean(), 2)},{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC2)'].median(), 2)}")
  print(f"{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC3)'].max(), 2)},{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC3)'].mean(), 2)},{round(data.loc[data[f'X-{x}'] == 1, 'Diff. (VWC3)'].median(), 2)}\n\n")
```

A4. Rainfall duration and intensity for hourly data (Python code)

```
# Initialization
import pandas as pd
data = pd.read_csv('Hourly 43 & 396.csv')
data['RF'] = data['RF'].fillna(0)

# Step 1: Detection and exclusion of isolated rainfall measurements
data['RF-1'] = -data['RF']
for x in range(-3, 4):
  data['RF-1'] = data['RF-1'] + data['RF'].shift(x)
data['RF-1'] = round(data['RF-1'], 3)
```

```
data.loc[((data['RF'] == 0.254) & (data['RF-1'] == 0)), 'RF'] = 0
del data['RF-1']

# Step 2: Detection and exclusion of irrelevant rainfall sub-events
data['RF-2'] = 0
for x in range(-6, 7):
  data['RF-2'] = data['RF-2'] + data['RF'].shift(x)
data['RF-2'] = round(data['RF-2'], 3)
data.loc[data['RF-2'] <= 1.016, 'RF'] = 0
del data['RF-2']

# Step 3: Identification of rainfall events
consecutive_zeros = 0
for i in range(len(data)):
    if data['RF'][i] == 0:
        consecutive_zeros += 1
    else:
        if consecutive_zeros >= 48:
            data.loc[i - consecutive_zeros:i - 1, 'RF'] = -1
        consecutive_zeros = 0

# Step 4: Calculation of event rainfall, duration, and intensity
rfd = pd.DataFrame(index=range(100), columns=['Rainfall', 'Duration'])
current_sum = 0
current_count = 0
start_index = 0
for i in range(len(data)):
    if data['RF'][i] > -1:
        current_sum += data['RF'][i]
        current_count += 1
    elif current_sum > 0 :
        rfd.loc[start_index, 'Rainfall'] = current_sum
        rfd.loc[start_index, 'Duration'] = current_count
        current_sum = 0
        current_count = 0
        start_index += 1
rfd['Intensity'] = rfd['Rainfall'] / rfd['Duration']

# Rounding for presentation
rfd['Rainfall'] = pd.to_numeric(rfd['Rainfall'])
rfd['Intensity'] = pd.to_numeric(rfd['Intensity'])
rfd['Rainfall'] = rfd['Rainfall'].round(0)
```

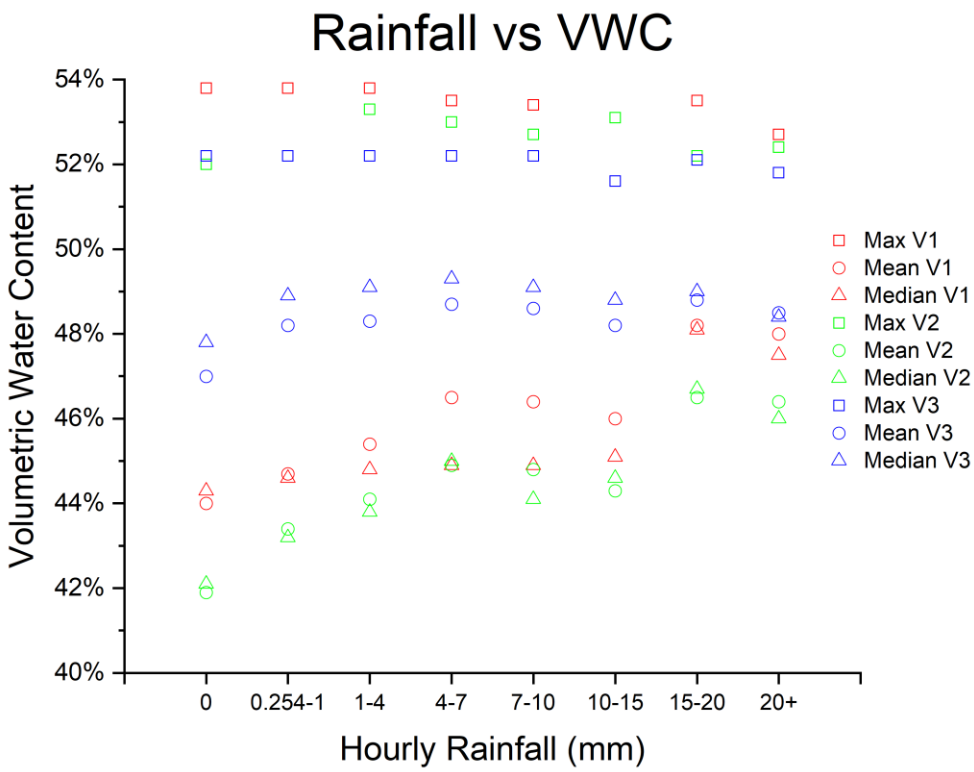
```
rfd['Intensity'] = rfd['Intensity'].round(1)

# Conversion to CSV
data.to_csv('Hourly 43 & 396 (Denoised).csv', encoding='utf-8-sig',
index=False)
rfd.to_csv('Rainfall Duration.csv', encoding='utf-8-sig', index=False)
```

A5. Four-dimensional matrix data (current to second offset)

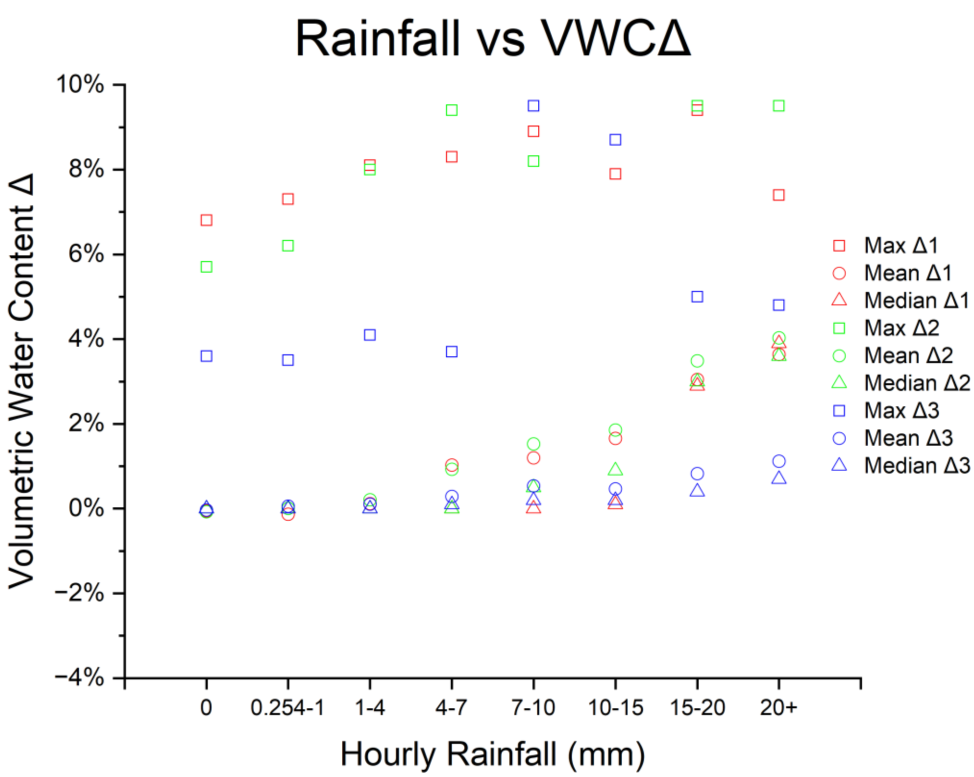
Current Rainfall						
Range Avg.	Mean V1	Mean V2	Mean V3	Mean Δ1	Mean Δ2	Mean Δ3
0	44.0%	41.9%	47.0%	-0.1%	-0.1%	0.0%
0.508	44.7%	43.4%	48.2%	-0.1%	0.0%	0.1%
2.413	45.4%	44.1%	48.3%	0.1%	0.2%	0.1%
5.461	46.5%	44.9%	48.7%	1.0%	0.9%	0.3%
8.509	46.4%	44.8%	48.6%	1.2%	1.5%	0.5%
12.573	46.0%	44.3%	48.2%	1.7%	1.9%	0.5%
17.526	48.2%	46.5%	48.8%	3.1%	3.5%	0.8%
20	48.0%	46.4%	48.5%	3.6%	4.0%	1.1%
Pearson's R	91.8%	87.6%	55.9%	98.8%	99.1%	97.4%
Range Avg.	Median V1	Median V2	Median V3	Median Δ1	Median Δ2	Median Δ3
0	44.3%	42.1%	47.8%	0.0%	0.0%	0.0%
0.508	44.6%	43.2%	48.9%	0.0%	0.0%	0.0%
2.413	44.8%	43.8%	49.1%	0.0%	0.0%	0.0%
5.461	44.9%	45.0%	49.3%	0.0%	0.0%	0.1%
8.509	44.9%	44.1%	49.1%	0.0%	0.5%	0.2%
12.573	45.1%	44.6%	48.8%	0.1%	0.9%	0.2%
17.526	48.1%	46.7%	49.0%	2.9%	3.0%	0.4%
20	47.5%	46.0%	48.4%	3.9%	3.6%	0.7%
Pearson's R	88.7%	89.0%	6.0%	84.7%	92.9%	94.0%
Range Avg.	Max V1	Max V2	Max V3	Max Δ1	Max Δ2	Max Δ3
0	53.8%	52.0%	52.2%	6.8%	5.7%	3.6%
0.508	53.8%	52.2%	52.2%	7.3%	6.2%	3.5%
2.413	53.8%	53.3%	52.2%	8.1%	8.0%	4.1%
5.461	53.5%	53.0%	52.2%	8.3%	9.4%	3.7%
8.509	53.4%	52.7%	52.2%	8.9%	8.2%	9.5%
12.573	53.1%	53.1%	51.6%	7.9%	8.7%	8.7%
17.526	53.5%	52.2%	52.1%	9.4%	9.5%	5.0%
20	52.7%	52.4%	51.8%	7.4%	9.5%	4.8%
Pearson's R	-84.4%	-4.0%	-64.7%	42.5%	79.8%	36.4%

Offset Rainfall -1						
Range Avg.	Mean V1	Mean V2	Mean V3	Mean Δ1	Mean Δ2	Mean Δ3
0	43.9%	41.8%	47.0%	0.0%	-0.1%	0.0%
0.508	44.5%	43.4%	48.3%	-0.3%	-0.1%	0.0%
2.413	45.2%	44.3%	48.5%	-0.2%	0.2%	0.1%
5.461	47.5%	45.9%	49.1%	1.0%	1.0%	0.4%
8.509	48.1%	46.9%	49.2%	1.8%	2.1%	0.6%
12.573	49.8%	47.2%	48.9%	3.9%	2.9%	0.8%
17.526	52.0%	49.3%	49.4%	3.6%	2.5%	0.4%
20	52.2%	50.6%	49.4%	4.2%	2.1%	0.9%
Pearson's R	99.0%	97.4%	76.1%	96.3%	87.5%	86.2%
Range Avg.	Median V1	Median V2	Median V3	Median Δ1	Median Δ2	Median Δ3
0	44.3%	42.1%	47.8%	0.0%	0.0%	0.0%
0.508	44.6%	43.2%	48.9%	0.0%	0.0%	0.0%
2.413	44.9%	44.0%	49.2%	0.0%	0.0%	0.0%
5.461	45.8%	47.4%	49.5%	0.0%	0.1%	0.1%
8.509	48.5%	48.2%	49.4%	0.1%	2.1%	0.3%
12.573	51.9%	48.5%	49.4%	4.5%	2.9%	0.4%
17.526	52.8%	49.9%	49.5%	4.0%	2.4%	0.3%
20	52.8%	51.5%	49.5%	4.6%	2.1%	0.7%
Pearson's R	97.4%	95.3%	64.8%	89.9%	84.7%	89.0%
Range Avg.	Max V1	Max V2	Max V3	Max Δ1	Max Δ2	Max Δ3
0	53.7%	51.7%	52.2%	6.4%	7.5%	2.6%
0.508	53.8%	51.3%	52.2%	6.5%	7.4%	8.7%
2.413	53.8%	52.2%	52.2%	6.8%	9.5%	9.5%
5.461	53.5%	51.8%	52.2%	7.9%	7.4%	3.5%
8.509	53.5%	52.6%	52.2%	8.2%	8.0%	3.9%
12.573	53.4%	52.7%	51.9%	8.9%	9.5%	4.6%
17.526	53.4%	53.9%	52.2%	8.3%	8.7%	1.8%
20	53.5%	53.3%	51.9%	9.4%	8.0%	4.1%
Pearson's R	-79.8%	92.0%	-63.6%	91.8%	30.4%	-48.9%

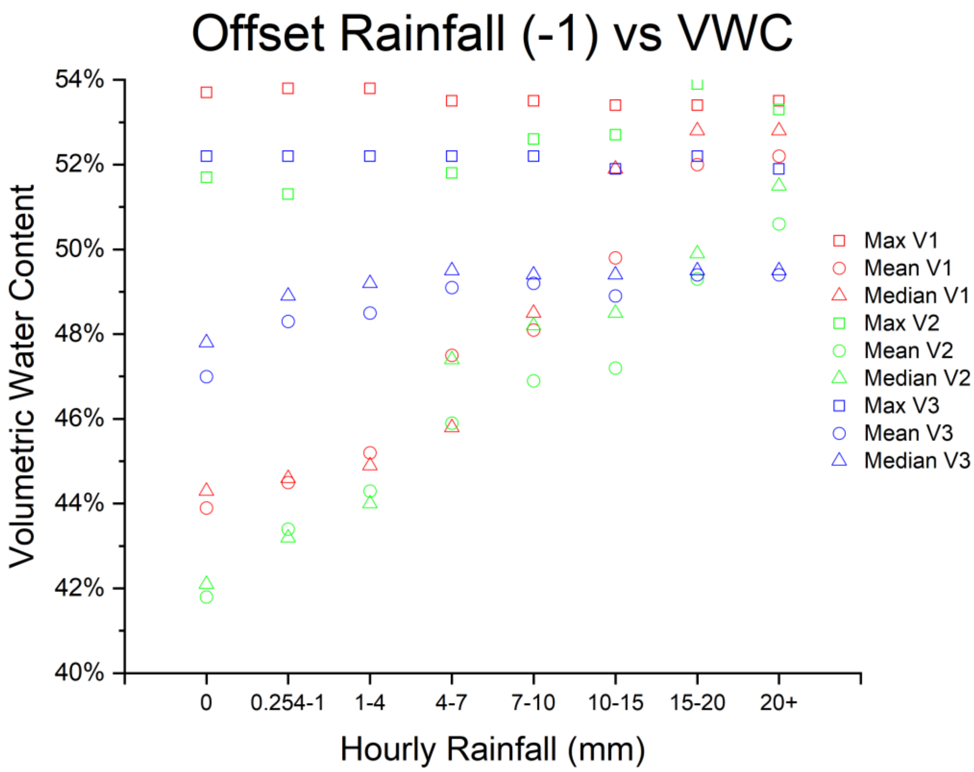
Offset Rainfall -2						
Range Avg.	Mean V1	Mean V2	Mean V3	Mean Δ1	Mean Δ2	Mean Δ3
0	43.9%	41.8%	47.0%	0.0%	0.0%	0.0%
0.508	44.4%	43.4%	48.3%	0.0%	0.0%	0.0%
2.413	45.2%	44.5%	48.6%	-0.1%	0.2%	0.1%
5.461	47.5%	46.4%	49.3%	0.0%	0.5%	0.2%
8.509	48.3%	47.1%	49.3%	0.2%	0.3%	0.2%
12.573	49.0%	47.1%	49.1%	-0.9%	-0.1%	0.2%
17.526	50.0%	48.2%	49.5%	-1.9%	-1.1%	0.1%
20	50.7%	49.3%	49.4%	-1.5%	-1.4%	0.0%
Pearson's R	96.6%	93.0%	73.9%	-88.0%	-79.9%	0.3%
Range Avg.	Median V1	Median V2	Median V3	Median Δ1	Median Δ2	Median Δ3
0	44.3%	42.1%	47.8%	0.0%	0.0%	0.0%
0.508	44.6%	43.2%	48.9%	0.0%	0.0%	0.0%
2.413	44.9%	44.5%	49.3%	0.0%	0.0%	0.0%
5.461	46.6%	47.8%	49.7%	0.0%	0.0%	0.1%
8.509	47.9%	48.2%	49.6%	0.0%	0.0%	0.1%
12.573	50.2%	48.3%	49.5%	0.0%	0.0%	0.1%
17.526	51.1%	48.6%	49.5%	-1.1%	-0.8%	0.1%
20	52.2%	49.3%	49.5%	-0.1%	-1.0%	0.1%
Pearson's R	99.3%	87.2%	58.3%	-54.5%	-83.8%	61.5%
Range Avg.	Max V1	Max V2	Max V3	Max Δ1	Max Δ2	Max Δ3
0	53.7%	53.3%	52.2%	9.4%	9.5%	9.5%
0.508	53.8%	53.9%	52.2%	8.3%	7.6%	8.7%
2.413	53.8%	52.2%	52.2%	7.3%	7.0%	4.8%
5.461	53.5%	52.7%	52.2%	7.9%	8.2%	2.7%
8.509	53.7%	52.5%	52.2%	7.7%	5.6%	4.1%
12.573	53.5%	51.8%	51.9%	6.8%	3.9%	1.0%
17.526	53.5%	52.6%	52.2%	0.5%	1.6%	0.6%
20	53.4%	52.7%	52.0%	0.4%	0.2%	0.4%
Pearson's R	-83.4%	-46.8%	-56.2%	-90.5%	-96.6%	-88.9%

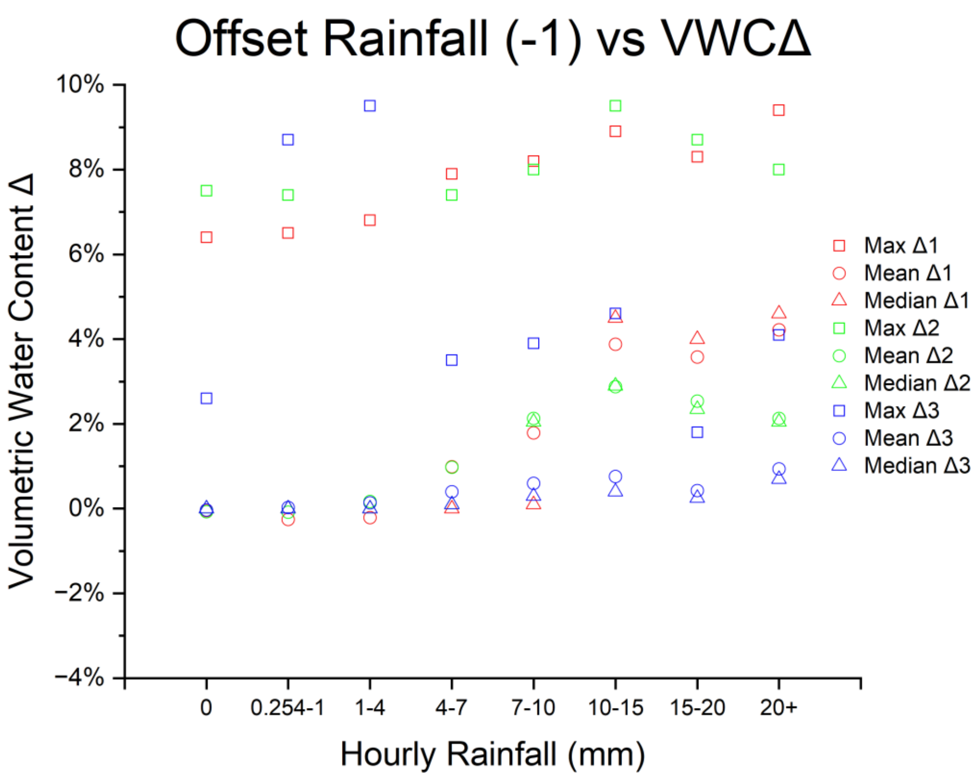


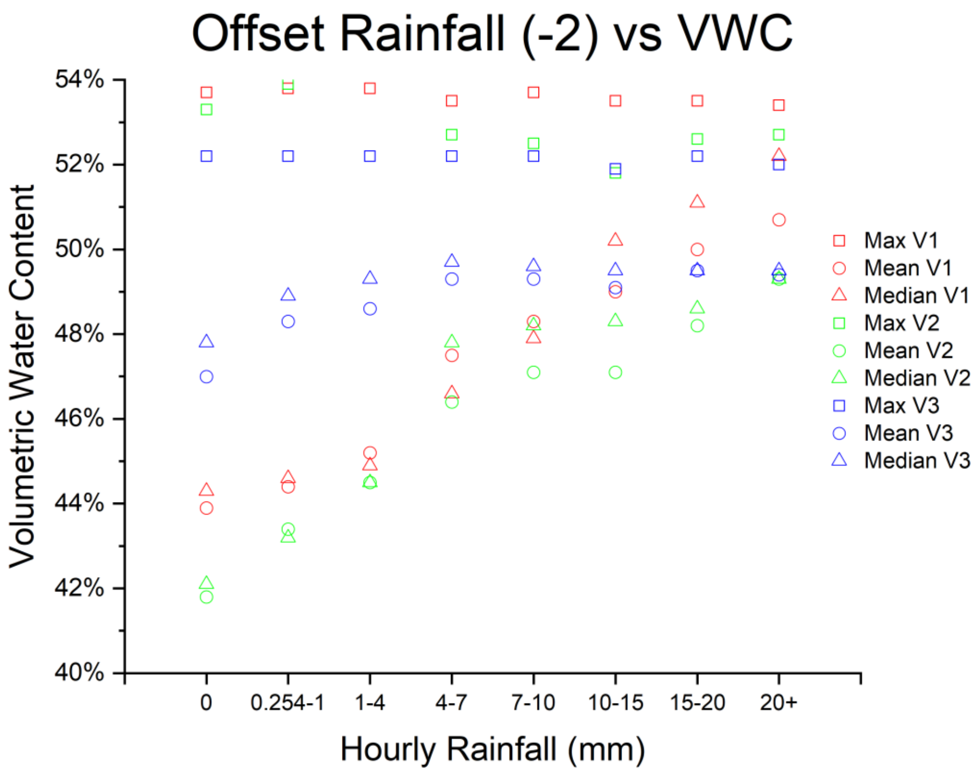
Rainfall vs VWC
Volumetric Water Content
54%
52%
50%
48%
46%
44%
42%
40%
0
0.254-1
1-4
4-7
7-10
10-15
15-20
20+
Hourly Rainfall (mm)
Max V1
Mean V1
Median V1
Max V2
Mean V2
Median V2
Max V3
Mean V3
Median V3

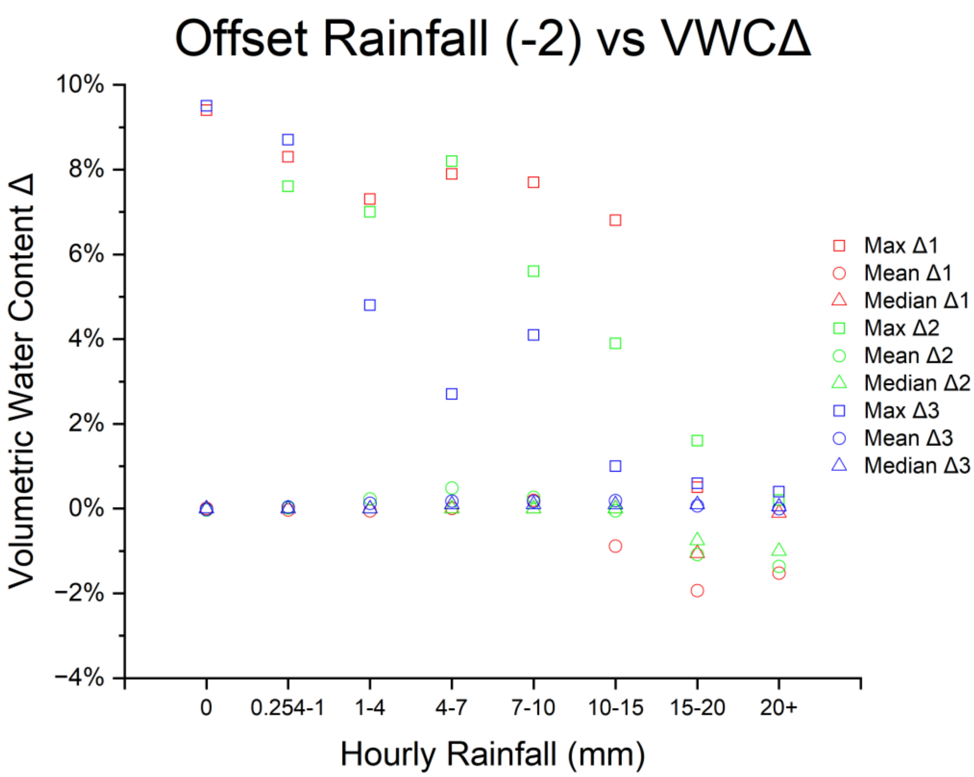


Rainfall vs VWCΔ
Volumetric Water Content Δ
10%
8%
6%
4%
2%
0%
−2%
−4%
0
0.254-1
1-4
4-7
7-10
10-15
15-20
20+
Hourly Rainfall (mm)
Max Δ1
Mean Δ1
Median Δ1
Max Δ2
Mean Δ2
Median Δ2
Max Δ3
Mean Δ3
Median Δ3


Offset Rainfall (-1) vs VWC
Volumetric Water Content
54%
52%
50%
48%
46%
44%
42%
40%
0
0.254-1
1-4
4-7
7-10
10-15
15-20
20+
Hourly Rainfall (mm)
Max V1
Mean V1
Median V1
Max V2
Mean V2
Median V2
Max V3
Mean V3
Median V3


Offset Rainfall (-1) vs VWCΔ
Volumetric Water Content Δ
10%
8%
6%
4%
2%
0%
−2%
−4%
0
0.254-1
1-4
4-7
7-10
10-15
15-20
20+
Hourly Rainfall (mm)
Max Δ1
Mean Δ1
Median Δ1
Max Δ2
Mean Δ2
Median Δ2
Max Δ3
Mean Δ3
Median Δ3


Offset Rainfall (-2) vs VWC
Volumetric Water Content
54%
52%
50%
48%
46%
44%
42%
40%
0
0.254-1
1-4
4-7
7-10
10-15
15-20
20+
Hourly Rainfall (mm)
Max V1
Mean V1
Median V1
Max V2
Mean V2
Median V2
Max V3
Mean V3
Median V3


Offset Rainfall (-2) vs VWCΔ
Volumetric Water Content Δ
10%
8%
6%
4%
2%
0%
−2%
−4%
0
0.254-1
1-4
4-7
7-10
10-15
15-20
20+
Hourly Rainfall (mm)
Max Δ1
Mean Δ1
Median Δ1
Max Δ2
Mean Δ2
Median Δ2
Max Δ3
Mean Δ3
Median Δ3