Title Goes Here

Agnes Schneider[2,*]

12 September, 2021

Abstract

Text of abstract

Contents

[1] Department of Physical Geography
[2] Philipps University Marburg

[*] Correspondence: Agnes Schneider <euboia@gmail.com>

Keywords: keyword 1; keyword 2; keyword 3

Highlights: These are the highlights.

#I. Introduction

The use of quantitative methods in archaeology reach back to the end of the 19th century, but it was only after the middle of the 20th century that they became computer-based. The first applications of statistical-mathematical methods using computers revolutionized the handling of spatial and quantitative archaeological data (e.g. Goldmann 1979) and also the view on archaeology itself (e.g. Clark 1968 and almost a decade later Hodder and Orton 1976, Ihm and Zimmermann 1978 to begin with). This quantitative analytical view preceded and prepared archaeologists for the advent of the public availability of soon high resolution digital datasets which revolutionized and broadened Aerial Archaeology, drawing upon the expertise of Satellite Remote Sensing and branching into a new discipline: Archaeological Remote Sensing. These new data sources, including geophysical platforms, led to the need to handle - in archaeological terms - big spatial data, thus to the specialized use of GIS platforms and the development and application of new methods like predictive modelling (van Leusen – Kamermans 2005; Kamermans et al. 2009). The sophistication of airborne platforms, sensors and imaging technologies, like LiDAR, hyper-spectral imagery and drone derived multispectral and hyper-spectral imagery (Agapiou – Lysandrou 2015; Luo et al. 2019) facilitated the diversification of the toolset of Archaeological Remote Sensing. New technical developments continuously push specialists to look for, borrow and adapt new methods to analyse in archaeological sense big data, which first lead to automating tasks such as georeferencing or applying the same function on multiple rasters (Raun and Patterson 2018, 70). Soon this lead to more complex automation – that is automating whole workflows and also analysis.

Thus in contrast to other disciplines, Archaeological Remote Sensing and Archaeological Science itself was quite late in adopting and applying automated methods and thus, Automated Archaeological Remote Sensing is still in its infancy (Opitz and Herrmann 2018, 30). Although accordingly, the expression 'automated analysis' is still controversial, it is nonetheless used throughout this Master's thesis. It has to be emphasized that 'automation' does not stand for completely automated workflows. It was chosen as a short phrase to address workflows with at least partly – or mostly – automated elements, because no one advocates 'automatic archaeology' (Cowley 2012, 6). Automated analysis, often also called semi-automated analysis, 'simply' means that a part, often a major part of a specific workflow, is automated, meaning computer aided. Critical voices about predictive modelling in GIS   which can be seen as the precursor and actual starting point of automated analysis   were very fittingly summarized by Wheatley 2004: predictive modelling is dehumanising, anti-historical and substitutes human actions with mathematical equations (Wheatley 2004). This distrust is reflected in the criticism towards automated analysis methods in the Archaeological and Archaeological Remote Sensing community, claiming, that archaeological projects are always locally specialized (Parcak 2009, 120) and that there is no generalized automation method that can be used location independently, which can locate atypical archaeological objects without producing a quite large miss-rate (Casana 2014, Casana 2020, S93). Taking these points into account: why only use one type of automated analysis method (Davis 2019, 5)? Large-scale landscapes feature lots of different landscape forms and archaeological features which need to be addressed in different ways. It is evident that it is not possible to detect everything with one analysis method or algorithm, due to the variability of the archaeological record. Other considerations include reflections on Automated Archaeological Remote Sensing detecting round and square shapes over and over and when it will move to something more useful that actually resembles archaeological objects (Rog Palmer in personal discussion and several AARGnews Editorials and contributions, e.g. AARGnews 62, 61)?

It is also somehow self-evident, that although pre-trained and known object classes are going to be detected, new and unexpected and also not detected objects always have to be expected. Thus expecting to detect unexpected or unknown object classes with supervised learning/automation is misleading and completely unrealistic (see a similar discussion about the use of predictive modelling in Wheatley 2004, 3.2.3). It is a fatal move to expect automation to be the magic trick, when it is not: it should be more seen as an extension of the archaeologist's toolbox. Because automated analysis is not to replace manual data evaluation which, based on the personal experience of the operator and them being human, can be prone to missing knowledge or error, or even archaeologists completely, but to be complementary. Automated analysis should be followed by field observation and identification for assessment, which on the other hand also can be biased for the same reason as manual data evaluation (Cowley 2012, Bennett et al 2014, 899). All in all automated analysis is to ease and facilitate the amount of work archaeologists are facing with the increasing amount of data and different data types to analyse and evaluate. As Davis (2020a, 3) emphasizes: automated analysis is precise and manual analysis is accurate. Thus both methods combined can help archaeologists to penetrate a hidden level of any dataset and landscape, which is invisible or hardly visible even for the trained eye.

Taking it further, reproducible analysis (Marwick 2017, Rokem et al. 2017, Marwick et al. 2018, Calero Valdez 2020) can help to facilitate automated analyses and can serve as control for human individuality: because archaeologists do see different things in the same data set and unconsciously see what they are familiar with and what they know (Cowley 2012, 7), very similarly to an algorithm trained to find a certain class of object. Human detection cannot be reproduced but automated analysis can: via documented, reproducible workflows. In order to achieve this goal open-source scripting languages like R, Python or open-source platforms like Google Engine should be used, to be able to repeat, reproduce and even replicate workflows to facilitate their use by other scholars. When writing code, the semantic syntax has to be clear and consistent. This also makes sure that the ontology and the semantic syntax will be the same when the code is used by a different operator on a different dataset – in contrary to manual analysis, where independent, subjective manual operators would define features or objects differently – thus arriving to the same or a least similar bias and errors, of course depending on the dataset. This argument is investigated more thoroughly in chapter III and the expressions reproducible and replicable are discussed in more detail.

Being entirely different disciplines, there is a semantic gap between the ontologies of Computer Vision, Image Analysis and Remote Sensing on one hand and Archaeology on the other hand – especially because the archaeological record itself is fragmented, multifaceted and poses ontological problems in itself for the interoperability of projects or databases, conducted by different operators with possible different metalanguage. Although the aim of this thesis is not the creation of an ontological and semantic framework and/or a metalanguage for Automated Archaeological Remote Sensing, these points have to be discussed because they affect the way how automation can be harmonized with Archaeological Remote Sensing.
This fundamental difference is reflected in the fact, that a distinction has to be made between 'object' and 'feature' in sensu stricto automation, based on termini technici from Computer Vision, on which ground 'object' is referred to real-world entities in remote sensing images and 'feature' to elements of an image and of an object (Traviglia et al. 2016, 12; Lambers et al 2019, 2), in contrary to the every-day use of the expression 'archaeological feature' at an excavation site or in reports. Reproducible code and workflows enable to conduct the same 'procedure' in a controlled environment and thus possible semantic problems which can result from how different operators see archaeological objects and features can be detected, distinguished and solved. Thus the accurate expert knowledge of manual analysis can be integrated in the precision and consistency of computational semantics (Davis 2020a, Fig 1.).

Here is a citation (Marwick 2017)

1 Background

2 Methods

3 Results

Figure 1 shows how we can have a caption and cross-reference for a plot

Here is an example of inline code 3.14 in the middle of a sentence.

4 Discussion

5 Conclusion

6 Acknowledgements

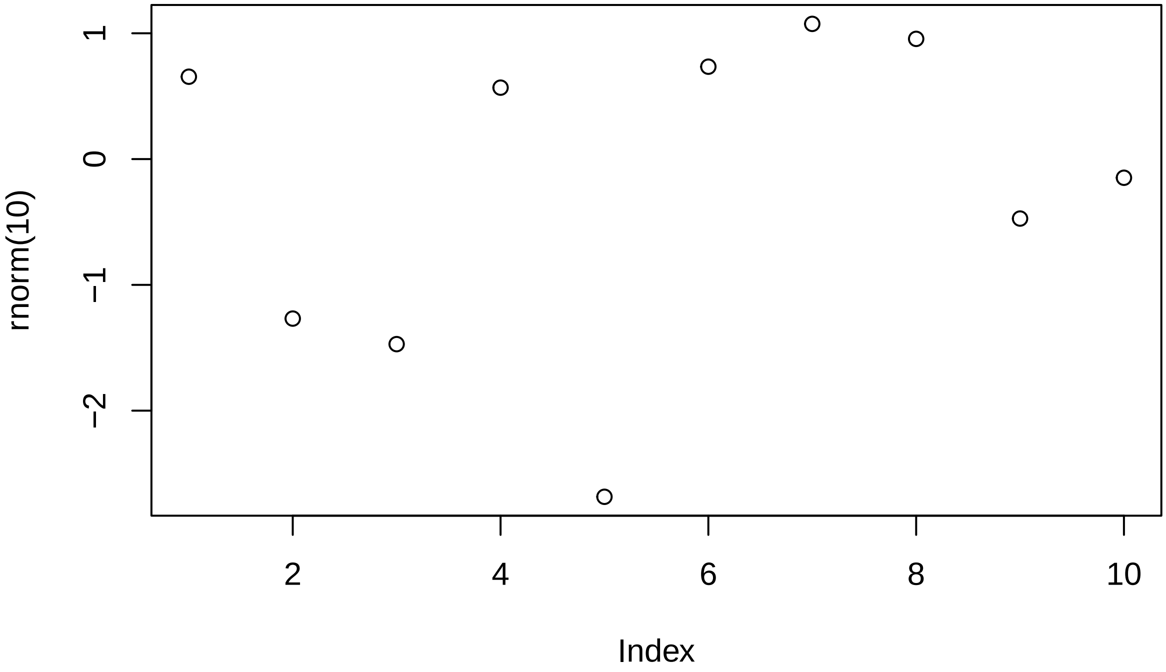

Figure 1: A plot of random numbers

7 References

Marwick, B. 2017. Computational reproducibility in archaeological research: Basic principles and a case study of their implementation. *Journal of Archaeological Method and Theory* 24: 424–50. https://doi.org/10.1007/s10816-015-9272-9. https://doi.org/10.1007/s10816-015-9272-9.

7.0.1 Colophon

This report was generated on 2021-09-12 18:41:42 using the following computational environment and dependencies:

```
#> - Session info ---------------------------------------------------------------
#>  setting  value
#>  version  R version 4.1.1 (2021-08-10)
#>  os       Windows 10 x64
#>  system   x86_64, mingw32
#>  ui       RTerm
#>  language (EN)
#>  collate  German_Germany.1252
#>  ctype    German_Germany.1252
#>  tz       Europe/Berlin
#>  date     2021-09-12
#>
#> - Packages -------------------------------------------------------------------
#>  package     * version date       lib source
#>  bookdown      0.24.1  2021-09-12 [1] Github (rstudio/bookdown@7ce6dc6)
#>  cachem        1.0.6   2021-08-19 [1] CRAN (R 4.1.1)
#>  callr         3.7.0   2021-04-20 [1] CRAN (R 4.1.0)
#>  cli           3.0.1   2021-07-17 [1] CRAN (R 4.1.0)
#>  crayon        1.4.1   2021-02-08 [1] CRAN (R 4.1.0)
#>  desc          1.3.0   2021-03-05 [1] CRAN (R 4.1.0)
#>  devtools      2.4.2   2021-06-07 [1] CRAN (R 4.1.0)
#>  digest        0.6.27  2020-10-24 [1] CRAN (R 4.1.0)
#>  ellipsis      0.3.2   2021-04-29 [1] CRAN (R 4.1.0)
#>  evaluate      0.14    2019-05-28 [1] CRAN (R 4.1.0)
#>  fastmap       1.1.0   2021-01-25 [1] CRAN (R 4.1.0)
#>  fs            1.5.0   2020-07-31 [1] CRAN (R 4.1.0)
#>  glue          1.4.2   2020-08-27 [1] CRAN (R 4.1.0)
#>  highr         0.9     2021-04-16 [1] CRAN (R 4.1.0)
#>  htmltools     0.5.2   2021-08-25 [1] CRAN (R 4.1.1)
#>  knitr         1.34    2021-09-09 [1] CRAN (R 4.1.1)
#>  lifecycle     1.0.0   2021-02-15 [1] CRAN (R 4.1.0)
#>  magrittr      2.0.1   2020-11-17 [1] CRAN (R 4.1.0)
#>  memoise       2.0.0   2021-01-26 [1] CRAN (R 4.1.0)
#>  pkgbuild      1.2.0   2020-12-15 [1] CRAN (R 4.1.0)
#>  pkgload       1.2.2   2021-09-11 [1] CRAN (R 4.1.1)
#>  prettyunits   1.1.1   2020-01-24 [1] CRAN (R 4.1.0)
#>  processx      3.5.2   2021-04-30 [1] CRAN (R 4.1.0)
#>  ps            1.6.0   2021-02-28 [1] CRAN (R 4.1.0)
#>  purrr         0.3.4   2020-04-17 [1] CRAN (R 4.1.0)
#>  R6            2.5.1   2021-08-19 [1] CRAN (R 4.1.1)
#>  remotes       2.4.0   2021-06-02 [1] CRAN (R 4.1.0)
#>  rlang         0.4.11  2021-04-30 [1] CRAN (R 4.1.0)
#>  rmarkdown     2.10    2021-08-06 [1] CRAN (R 4.1.0)
#>  rprojroot     2.0.2   2020-11-15 [1] CRAN (R 4.1.0)
#>  rstudioapi    0.13    2020-11-12 [1] CRAN (R 4.1.0)
#>  sessioninfo   1.1.1   2018-11-05 [1] CRAN (R 4.1.0)
#>  stringi       1.7.4   2021-08-25 [1] CRAN (R 4.1.1)
#>  stringr       1.4.0   2019-02-10 [1] CRAN (R 4.1.0)
#>  testthat      3.0.4   2021-07-01 [1] CRAN (R 4.1.0)
#>  usethis       2.0.1   2021-02-10 [1] CRAN (R 4.1.0)
#>  withr         2.4.2   2021-04-18 [1] CRAN (R 4.1.0)
#>  xfun          0.25    2021-08-06 [1] CRAN (R 4.1.0)
#>  yaml          2.2.1   2020-02-01 [1] CRAN (R 4.1.0)
#>
#> [1] C:/Users/kelto/Documents/R/win-library/4.1
#> [2] C:/Program Files/R/R-4.1.1/library
```

The current Git commit details are:

```
#> Local:    main C:/Users/kelto/Documents/iSEGMound
#> Remote:   main @ origin (https://github.com/keltoskytoi/iSEGMound)
#> Head:     [64ce64e] 2021-09-12: push files
```