Import - Edge angle analysis

Lisa Schunk

2021-02-04 16:45:28

# Goal of the script

This script imports and merges all single CSV-files generated with the ‘edge angle method’. The data derives from 3D models of artefacts from the three sites Balver Höhle, Buhlen and Ramioul.  
The script will:

1. Read in the original CSV-files
2. Combine the data from all samples into one
3. Write an XLSX-file and save an R object ready for further analysis in R

dir\_in <- "analysis/raw\_data"  
dir\_out <- "analysis/derived\_data/"

Raw data must be located in “analysis/raw\_data”.  
Formatted data will be saved in “analysis/derived\_data/”. The knit directory for this script is the project directory.

# Load packages

library(tidyverse)

Warning: package 'ggplot2' was built under R version 4.0.3

Warning: package 'readr' was built under R version 4.0.3

Warning: package 'dplyr' was built under R version 4.0.3

Warning: package 'forcats' was built under R version 4.0.3

library(R.utils)  
library(openxlsx)

Warning: package 'openxlsx' was built under R version 4.0.3

library(tools)

# List all files and get names of the files

# List all CSV files in dir\_in  
CSV\_files <- list.files(dir\_in, pattern = "\\.csv$", recursive = TRUE, full.names = TRUE)

# Merge all files and format the data

# Create a list   
data\_final <- vector(mode = "list", length = length(CSV\_files))   
names(data\_final) <- basename(CSV\_files)  
  
# For each sample  
for (s in seq\_along(data\_final)) {  
   
 # Gets sample ID from path names   
 ID <- dirname(dirname(dirname(CSV\_files[s]))) %>%  
 basename()  
  
 # Gets name of the site from path names   
 site <- basename(dirname(dirname(dirname(dirname(dirname(CSV\_files[s])))))) %>%   
 gsub("([A-Za-z0-9\_]\*-)", "", x = .)  
   
 # Gets tool type from path names   
 tool\_type <- dirname(dirname(dirname(dirname(CSV\_files[s])))) %>%  
 basename()   
   
 # Gets section from path names  
 sec <- basename(CSV\_files[s]) %>%  
 gsub("^[A-Za-z0-9\_-]\*\_SEC-", "", x = .) %>%   
 gsub("\_.\*\\.csv$", "", x = .) %>%   
 as.numeric()  
   
 # Gets edge (E1/E2/E3) from path name  
 edge <- basename((dirname(CSV\_files[s])))  
 edge <- unlist(strsplit(edge, "\_"))[3]   
   
   
 # read the data files  
 data\_final[[s]] <- read.csv(CSV\_files[s]) %>%   
 mutate(Site = site, ID = ID, Tool\_type = tool\_type, Section = sec,   
 Edge = edge) %>%   
 select(Site, ID, Tool\_type, Section, Edge, everything()) %>%   
 select(-section) %>%  
 rename(Angle\_number = angel\_number,   
 Distance\_origin = dist.to.origin.on.curve..mm.,   
 Segment = segment.on.section..mm.,  
 Three\_point = angle.1..3.points...degree.,   
 Two\_lines = angle.2..2.constructed.lines...degree.,   
 Best\_fit = angle.3..2.BestFit.lines...degree.)  
}  
  
# rbind all files   
data\_final <- do.call(rbind, data\_final)  
# adds indices as row names   
row.names(data\_final) <- 1:nrow(data\_final)

# Save data

## Format name of output file

file\_out <- "EdgeAngle"

## Write to XLSX

write.xlsx(list(data = data\_final), file = paste0(dir\_out, file\_out, ".xlsx"))

## Save R object

saveObject(data\_final, file = paste0(dir\_out, file\_out, ".Rbin"))

## Show files information

files\_out <- list.files(dir\_out, full.names = TRUE)  
md5\_out <- md5sum(files\_out)  
info\_out <- data.frame(files = basename(names(md5\_out)), checksum = md5\_out,   
 row.names = NULL)

The checksum (MD5 hashes) of the exported files are:

files checksum  
1 EdgeAngle.Rbin c0fbf0d522d77a70c3a828b8ba7880fc  
2 EdgeAngle.xlsx 0596ec4c61d57dfefe809c595c59a2af

# sessionInfo() and RStudio version

sessionInfo()

R version 4.0.2 (2020-06-22)  
Platform: x86\_64-w64-mingw32/x64 (64-bit)  
Running under: Windows 10 x64 (build 19041)  
  
Matrix products: default  
  
locale:  
[1] LC\_COLLATE=German\_Germany.1252 LC\_CTYPE=German\_Germany.1252   
[3] LC\_MONETARY=German\_Germany.1252 LC\_NUMERIC=C   
[5] LC\_TIME=German\_Germany.1252   
  
attached base packages:  
[1] tools stats graphics grDevices utils datasets methods   
[8] base   
  
other attached packages:  
 [1] openxlsx\_4.2.3 R.utils\_2.10.1 R.oo\_1.24.0 R.methodsS3\_1.8.1  
 [5] forcats\_0.5.1 stringr\_1.4.0 dplyr\_1.0.3 purrr\_0.3.4   
 [9] readr\_1.4.0 tidyr\_1.1.2 tibble\_3.0.6 ggplot2\_3.3.3   
[13] tidyverse\_1.3.0   
  
loaded via a namespace (and not attached):  
 [1] tidyselect\_1.1.0 xfun\_0.20 haven\_2.3.1 colorspace\_2.0-0   
 [5] vctrs\_0.3.6 generics\_0.1.0 htmltools\_0.5.1.1 yaml\_2.2.1   
 [9] rlang\_0.4.10 pillar\_1.4.7 glue\_1.4.2 withr\_2.4.1   
[13] DBI\_1.1.1 dbplyr\_2.0.0 modelr\_0.1.8 readxl\_1.3.1   
[17] lifecycle\_0.2.0 munsell\_0.5.0 gtable\_0.3.0 cellranger\_1.1.0   
[21] zip\_2.1.1 rvest\_0.3.6 evaluate\_0.14 knitr\_1.31   
[25] broom\_0.7.4 Rcpp\_1.0.6 scales\_1.1.1 backports\_1.2.1   
[29] jsonlite\_1.7.2 fs\_1.5.0 hms\_1.0.0 digest\_0.6.27   
[33] stringi\_1.5.3 grid\_4.0.2 cli\_2.3.0 magrittr\_2.0.1   
[37] crayon\_1.4.0 pkgconfig\_2.0.3 ellipsis\_0.3.1 xml2\_1.3.2   
[41] reprex\_1.0.0 lubridate\_1.7.9.2 assertthat\_0.2.1 rmarkdown\_2.6   
[45] httr\_1.4.2 rstudioapi\_0.13 R6\_2.5.0 compiler\_4.0.2

RStudio version 1.3.1073.

END OF SCRIPT