Import - Lithic analysis Buhlen

Lisa Schunk

2021-02-04

# Goal of the script

This script reads the xlsx file (database techno-typological analysis) generated with E4 and formats the data for a statistical analysis.  
The script will:

1. Reads in the original xlsx file
2. Changes and sort the data in order to do stats
3. Saves the data as a new xlsx file and R object

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

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

# Load packages

library(openxlsx)

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

library(readxl)  
library(R.utils)  
library(tools)  
library(data.table)

Warning: package 'data.table' was built under R version 4.0.3

library(chron)  
library(dplyr)

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

# Get name, path and information of the file

data\_file <- list.files(dir\_in, pattern = "\\.xlsx$", full.names = TRUE)  
md5\_in <- md5sum(data\_file)  
info\_in <- data.frame(files = basename(names(md5\_in)), checksum = md5\_in,   
 row.names = NULL)

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

files checksum  
1 Buhlen.xlsx 033dbad42d8f4bf646179adc12b13919

# Read in original xlsx-file

imp\_data <- read.xlsx(xlsxFile = data\_file, sheet = 1, startRow = 1, colNames = TRUE,  
 rowNames = FALSE, skipEmptyRows = FALSE)   
#select the columns to keep   
keep\_col <- c(1:2, 9:19, 21:35, 37:38, 40:43)  
data\_final <- imp\_data[,keep\_col]  
  
data\_final$length <- as.numeric(data\_final$length)  
data\_final$width <- as.numeric(data\_final$width)  
data\_final$thickness <- as.numeric(data\_final$thickness)  
data\_final$weight <- as.numeric(data\_final$weight)  
data\_final$perimeter.basis.back <- as.numeric(data\_final$perimeter.basis.back)  
data\_final$perimeter.arch <- as.numeric(data\_final$perimeter.arch)

Error in `$<-.data.frame`(`\*tmp\*`, perimeter.arch, value = numeric(0)): replacement has 0 rows, data has 199

data\_final$perimeter.active.edge <- as.numeric(data\_final$perimeter.active.edge)  
data\_final$perimeter.total <- as.numeric(data\_final$perimeter.total)  
data\_final$thickness.back <- as.numeric(data\_final$thickness.back)  
  
str(data\_final)

'data.frame': 199 obs. of 34 variables:  
 $ site : chr "Buhlen" "Buhlen" "Buhlen" "Buhlen" ...  
 $ ID : chr "BU-002" "BU-003" "BU-004" "BU-005" ...  
 $ raw.material : chr "silicified\_schist" "silicified\_schist" "silicified\_schist" "silicified\_schist" ...  
 $ technological.class : chr "Keilmesser" "Keilmesser" "Keilmesser" "Keilmesser" ...  
 $ artefact.state : chr "complete" "complete" "complete" "complete" ...  
 $ blank : chr "core" "core" "core" "core" ...  
 $ morpho.type : chr "Bockstein" "Buhlen" "Pradnik" "Balve" ...  
 $ cortex : chr "YES" "YES" "YES" "YES" ...  
 $ cortex.percentage : chr "25-50" "<25" "<25" "50-75" ...  
 $ cortex.location : chr "back" "base" "back" "ventral\_and\_dorsal" ...  
 $ morphology.back : chr "cortex/unworked" "cortex/partly\_retouched" "cortex/unworked" "cortex/unworked" ...  
 $ retouch.active.edge : chr "YES" "YES" "YES" "YES" ...  
 $ retouch.type.edge : chr "bifacial" "bifacial" "bifacial" "bifacial" ...  
 $ tip.morphology : chr "rounded" "rounded" "rounded" "pointed" ...  
 $ application.Pradnik.method : chr "NO" "YES" "YES" "NO" ...  
 $ frequency.application.Pradnik.method: chr NA "one" "one" NA ...  
 $ type.lateral.sharpening.spall : chr NA NA NA NA ...  
 $ tool.lateralisation : chr "dex." "dex." "dex." "dex." ...  
 $ length : num 48 58 56.3 69 53.7 ...  
 $ width : num 35 31 38.8 46 36.2 ...  
 $ thickness : num 14 18 16 14 17 18 19 31 13 12 ...  
 $ weight : num 0.0259 0.0337 0.0391 0.0561 0.0367 0.0258 0.0387 0.0733 0.0224 0.0204 ...  
 $ perimeter.basis.back : num 8.4 8.2 7.5 8.4 8.2 7.6 7.7 9.4 8.1 8.8 ...  
 $ perimeter.distal.posterior.part : num 0 1.6 2.1 2.2 1 1.5 3.9 2.5 2.4 1 ...  
 $ perimeter.active.edge : num 5.6 4.7 6.5 8.2 4.4 4.2 5.7 7 5.1 4.7 ...  
 $ perimeter.total : num 0 11 0 0 0 0 0 0 0 0 ...  
 $ thickness.back : num 14 11 14 12 17.3 ...  
 $ taphonomic.visual.inspection : chr "Sharp\_edges\_and\_preserved\_surface" "Sharp\_edges\_and\_preserved\_surface" "Sharp\_edges\_and\_preserved\_surface" "Sharp\_edges\_and\_preserved\_surface" ...  
 $ tool.edges.preservation : chr "edges\_preserved" "edges\_preserved" "edges\_preserved" "edges\_preserved" ...  
 $ macroscopically.visible.use-wear : chr "NO" "NO" "NO" "NO" ...  
 $ use-wear.analysis : chr "NO" "YES" "YES" "NO" ...  
 $ 3D-scan : chr "YES" "YES" "YES" "YES" ...  
 $ schistosity : chr "N/A" "NO" "NO" "YES" ...  
 $ orientation.schistosity : chr NA NA NA "parallel\_to\_the\_active\_edge" ...

# Data analsysis - sorting

## Dimension

# keeps only columns relevant for dimensions and sorts them based on   
# their technological class  
keep\_col <- c(1:2, 4:5,19:21)  
dimensions <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_dimensions <- dimensions[2:131, ] %>% arrange(artefact.state)  
KM.point\_dimensions <- KM\_dimensions[113:127, ]  
KM.only\_dimensions <- KM\_dimensions[-(113:127), ]  
KM.complete\_dimensions <- KM\_dimensions[1:111, ]  
PS\_dimensions <- dimensions[174:197, ] %>% arrange(artefact.state)  
LSS\_dimensions <- dimensions[132:173, ] %>% arrange(artefact.state)  
S\_dimensions <- dimensions[198:199, ] %>% arrange(artefact.state)

## Perimeter

# keeps only columns relevant for perimeter measurements and sorts them   
# based on their technological class  
keep\_col <- c(1:2, 4:5,7,23:26)  
perimeter <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_perimeter <- perimeter[2:131, ] %>% arrange(artefact.state)  
PS\_perimeter <- perimeter[174:197, ] %>% arrange(artefact.state)

## Weight

# keeps only columns relevant for weight measurements and sorts them based   
# on their technological class  
keep\_col <- c(1:2, 4:5, 22)  
weight <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_weight <- weight[2:131, ] %>% arrange(artefact.state)  
PS\_weight <- weight[174:197, ] %>% arrange(artefact.state)  
LSS\_weight <- weight[132:173, ] %>% arrange(artefact.state)  
S\_weight <- weight[198:199, ] %>% arrange(artefact.state)

## Raw material

# keeps only columns relevant for raw material classification and sorts them   
# based on their technological class  
keep\_col <- c(1:2, 4:5, 3)  
raw\_material <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_raw\_material <- raw\_material[2:131, ] %>% arrange(artefact.state)  
PS\_raw\_material <- raw\_material[174:197, ] %>% arrange(artefact.state)  
LSS\_raw\_material <- raw\_material[132:173, ] %>% arrange(artefact.state)  
S\_raw\_material <- raw\_material[198:199, ] %>% arrange(artefact.state)

## Cortex + blanks

# keeps only columns relevant for cortex and blank classification and sorts them   
# based on their technological class  
keep\_col <- c(1:2, 4:5, 6, 8:10)  
cortex\_blanks <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_cortex\_blanks <- cortex\_blanks[2:131, ] %>% arrange(artefact.state)  
PS\_cortex\_blanks <- cortex\_blanks[174:197, ] %>% arrange(artefact.state)

## Back

# keeps only columns relevant for back modifications and sorts them based   
# on their technological class  
keep\_col <- c(1:2, 4:5, 11, 27)  
back <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_back <- back[2:131, ] %>% arrange(artefact.state)  
PS\_back <- back[174:197, ] %>% arrange(artefact.state)

## Edge retouch

# keeps only columns relevant for edge retouch classification and sorts them   
# based on their technological class  
keep\_col <- c(1:2, 4:5, 12:13, 29)  
edge\_retouch <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_edge\_retouch <- edge\_retouch[2:131, ] %>% arrange(artefact.state)  
PS\_edge\_retouch <- edge\_retouch[174:197, ] %>% arrange(artefact.state)  
LSS\_edge\_retouch <- edge\_retouch[132:173, ] %>% arrange(artefact.state)  
S\_edge\_retouch <- edge\_retouch[198:199, ] %>% arrange(artefact.state)

## Morpho type

# keeps only columns relevant for morpho type classification and sorts them   
# based on their technological class  
keep\_col <- c(1:2, 4:5, 7, 19:21)  
morpho.type <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_morpho.type <- morpho.type[2:131, ] %>% arrange(artefact.state)  
PS\_morpho.type <- morpho.type[174:197, ] %>% arrange(artefact.state)

## Application ‘Pradnik method’

# keeps only columns relevant for 'morpho type 'Pradnik method' classification   
# and sorts them based on their technological class  
keep\_col <- c(1:2, 4:5, 15:16)  
Pradnik.method <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_Pradnik.method <- Pradnik.method[2:131, ] %>% arrange(artefact.state)  
PS\_Pradnik.method <- Pradnik.method[174:197, ] %>% arrange(artefact.state)

## Lateralisation

# keeps only columns relevant for lateralisation and sorts them based on their  
# technological class  
keep\_col <- c(1:2, 4:5, 18)  
lateralisation <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
KM\_lateralisation <- lateralisation[2:131, ] %>% arrange(artefact.state)  
PS\_lateralisation <- lateralisation[174:197, ] %>% arrange(artefact.state)

## Type lateral sharpening spall

# keeps only columns relevant for lateral sharpening spall classification and sorts   
# them based on their technological class  
keep\_col <- c(1:2, 4:5, 17:18)  
lss\_type <- data\_final[, keep\_col] %>% arrange(technological.class)  
  
LSS\_type <- lss\_type[132:173, ] %>% arrange(artefact.state)

# Save data

## Format name of output file

file\_out <- "Buhlen\_lithic\_analysis"

The files will be saved as “~/Buhlen\_lithic\_analysis.[ext]”.

## Write to XLSX

write.xlsx(list(data = data\_final, dimensions = dimensions, KM\_dimensions = KM\_dimensions,  
 KM.point\_dimensions =  
 KM.point\_dimensions, KM.only\_dimensions = KM.only\_dimensions,  
 KM.complete\_dimensions = KM.complete\_dimensions, PS\_dimensions =  
 PS\_dimensions, LSS\_dimensions = LSS\_dimensions, S\_dimensions =  
 S\_dimensions, KM\_perimeter = KM\_perimeter, PS\_perimeter = PS\_perimeter,  
 KM\_weight = KM\_weight, PS\_weight = PS\_weight,LSS\_weight = LSS\_weight,  
 S\_weight = S\_weight, KM\_raw\_material = KM\_raw\_material,PS\_raw\_material =  
 PS\_raw\_material, LSS\_raw\_material = LSS\_raw\_material, S\_raw\_material =  
 S\_raw\_material, KM\_cortex\_blanks = KM\_cortex\_blanks, PS\_cortex\_blanks =  
 PS\_cortex\_blanks, KM\_back = KM\_back, PS\_back = PS\_back, KM\_edge\_retouch  
 = KM\_edge\_retouch, PS\_edge\_retouch = PS\_edge\_retouch, LSS\_edge\_retouch =  
 LSS\_edge\_retouch, S\_edge\_retouch = S\_edge\_retouch, KM\_morpho.type =  
 KM\_morpho.type, PS\_morpho.type = PS\_morpho.type, KM\_Pradnik.method =  
 KM\_Pradnik.method, PS\_Pradnik.method = PS\_Pradnik.method,  
 KM\_lateralisation = KM\_lateralisation, PS\_lateralisation =  
 PS\_lateralisation, LSS\_type = LSS\_type),  
 file = paste0(dir\_out, file\_out, ".xlsx"))

## Save R object

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

RStudio version 1.3.1056.

END OF SCRIPT