Qualitative use-wear analysis

Lisa Schunk

2021-02-04 16:01:17

# Goal of the script

This script formats the output of the resulting CSV-file from digitalising the location of use-wear traces in QGIS. The script will:

1. Read in the original CSV-file
2. Format the data
3. Calculate the percentages
4. Plot the data as pie charts

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

Raw data must be located in ~/analysis/raw\_data.  
Formatted data will be saved in ~/analysis/plots/.

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(tools)  
library(R.utils)  
library(chron)  
library(ggplot2)

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

library(wesanderson)

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

library(dplyr)

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

library(doBy)

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

# Get names, path and information of the file

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

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

file checksum  
1 all.csv 963cf31444f489d15dd903c0bc9fe7b6

# Read in original CSV-file

imp\_data <- read.csv(data\_file, header = TRUE, stringsAsFactors = FALSE, na.strings = "\*\*\*\*\*")  
str(imp\_data)

'data.frame': 321 obs. of 6 variables:  
 $ tool.type : chr "Keilmesser" "Keilmesser" "Keilmesser" "Keilmesser" ...  
 $ id : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ site : chr "Buhlen" "Buhlen" "Buhlen" "Buhlen" ...  
 $ specimen : chr "BU-090" "BU-093" "BU-097" "BU-004" ...  
 $ area : chr "A1" "B1" "B1" "A2" ...  
 $ use.wear.type: chr "A" "A" "A" "A" ...

# Percentages

## Percentages of use-wear types for all tool types together

# splits the data into single areas   
sp\_all <- split(imp\_data, imp\_data[["area"]])  
pct\_all <- vector(mode = "list", length = length(sp\_all))  
  
  
for (i in seq\_along(sp\_all)) {  
 pct\_all[[i]] <- sp\_all[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_all)[i])  
}  
  
PCT\_all <- do.call(rbind, pct\_all) %>%   
 select(area, use.wear.type, perc)

## Percentages of use-wear types for all Keilmesser

# defines which part of the data belongs to Keilmesser   
KM <- imp\_data[imp\_data[["tool.type"]] == "Keilmesser", ]  
  
# splits the data into single areas   
sp\_KM <- split(KM, KM[["area"]])  
pct\_KM <- vector(mode = "list", length = length(sp\_KM))  
  
  
for (i in seq\_along(sp\_KM)) {  
 pct\_KM[[i]] <- sp\_KM[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_KM)[i])  
}  
  
PCT\_KM <- do.call(rbind, pct\_KM) %>%   
 select(area, use.wear.type, perc)

## Percentages of use-wear types for all Keilmesser from Buhlen

# defines which part of the data belongs to Keilmesser   
KM <- imp\_data[imp\_data[["tool.type"]] == "Keilmesser", ]  
# selects only the data from Buhlen  
KM\_Buhlen <- KM [ , ] %>% arrange(site)   
KM\_Buhlen <- KM\_Buhlen[105:177, ]  
  
# splits the data into single areas   
sp\_KM\_Buhlen <- split(KM\_Buhlen, KM\_Buhlen[["area"]])  
pct\_KM\_Buhlen <- vector(mode = "list", length = length(sp\_KM\_Buhlen))  
  
  
for (i in seq\_along(sp\_KM\_Buhlen)) {  
 pct\_KM\_Buhlen[[i]] <- sp\_KM\_Buhlen[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_KM\_Buhlen)[i])  
}  
  
PCT\_KM\_Buhlen <- do.call(rbind, pct\_KM\_Buhlen) %>%   
 select(area, use.wear.type, perc)

## Percentages of use-wear types for all Keilmesser from Balve

# defines which part of the data belongs to Keilmesser   
KM <- imp\_data[imp\_data[["tool.type"]] == "Keilmesser", ]  
# selects only the data from Balve  
KM\_Balve <- KM [ , ] %>% arrange(site)   
KM\_Balve <- KM\_Balve[1:104, ]  
  
# splits the data into single areas   
sp\_KM\_Balve <- split(KM\_Balve, KM\_Balve[["area"]])  
pct\_KM\_Balve <- vector(mode = "list", length = length(sp\_KM\_Balve))  
  
  
for (i in seq\_along(sp\_KM\_Balve)) {  
 pct\_KM\_Balve[[i]] <- sp\_KM\_Balve[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_KM\_Balve)[i])  
}  
  
PCT\_KM\_Balve <- do.call(rbind, pct\_KM\_Balve) %>%   
 select(area, use.wear.type, perc)

## Percentages of use-wear types for all Keilmesser from Ramioul

# defines which part of the data belongs to Keilmesser   
KM <- imp\_data[imp\_data[["tool.type"]] == "Keilmesser", ]  
# selects only the data from Ramioul  
KM\_Ramioul <- KM [ , ] %>% arrange(site)   
KM\_Ramioul <- KM\_Ramioul[178:195, ]  
  
# splits the data into single areas   
sp\_KM\_Ramioul <- split(KM\_Ramioul, KM\_Ramioul[["area"]])  
pct\_KM\_Ramioul <- vector(mode = "list", length = length(sp\_KM\_Ramioul))  
  
  
for (i in seq\_along(sp\_KM\_Ramioul)) {  
 pct\_KM\_Ramioul[[i]] <- sp\_KM\_Ramioul[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_KM\_Ramioul)[i])  
}  
  
PCT\_KM\_Ramioul <- do.call(rbind, pct\_KM\_Ramioul) %>%   
 select(area, use.wear.type, perc)

## Percentages of use-wear types for all Pradnik scraper

# defines which part of the data belongs to Pradnick scraper   
PS <- imp\_data[imp\_data[["tool.type"]] == "Pradnick scraper", ]  
  
# splits the data into single areas   
sp\_PS <- split(PS, PS[["area"]])  
pct\_PS <- vector(mode = "list", length = length(sp\_PS))  
  
  
for (i in seq\_along(sp\_PS)) {  
 pct\_PS[[i]] <- sp\_PS[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_PS)[i])  
}  
  
PCT\_PS <- do.call(rbind, pct\_PS) %>%   
 select(area, use.wear.type, perc)

Error in UseMethod("select"): no applicable method for 'select' applied to an object of class "NULL"

## Percentages of use-wear types for all Pradnik spalls

# defines which part of the data belongs to Pradnick spall   
LSS <- imp\_data[imp\_data[["tool.type"]] == "Pradnick spall", ]  
  
# splits the data into single areas   
sp\_LSS <- split(LSS, LSS[["area"]])  
pct\_LSS <- vector(mode = "list", length = length(sp\_LSS))  
  
  
for (i in seq\_along(sp\_LSS)) {  
 pct\_LSS[[i]] <- sp\_LSS[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_LSS)[i])  
}  
  
PCT\_LSS <- do.call(rbind, pct\_LSS) %>%   
 select(area, use.wear.type, perc)

Error in UseMethod("select"): no applicable method for 'select' applied to an object of class "NULL"

## Percentages of use-wear types for all scraper

# defines which part of the data belongs to Pradnick spall   
S <- imp\_data[imp\_data[["tool.type"]] == "scraper", ]  
  
# splits the data into single areas   
sp\_S <- split(S, S[["area"]])  
pct\_S <- vector(mode = "list", length = length(sp\_S))  
  
  
for (i in seq\_along(sp\_S)) {  
 pct\_S[[i]] <- sp\_S[[i]] %>%   
 group\_by(use.wear.type) %>%  
 summarize(perc = round(n() / nrow(.) \* 100, digits = 2)) %>%   
 mutate(area = names(sp\_S)[i])  
}  
  
PCT\_S <- do.call(rbind, pct\_S) %>%   
 select(area, use.wear.type, perc)

# Pie charts

## Colour definitions

#05100c black   
#999999 gray   
#52854c green   
#c3d7a4 light green   
#487bb6 blue   
#a6cee3 light blue   
#9a0f0f red  
#d16103 orange  
#fdbf6f apricot  
#ffdb6d yellow  
#985633 brown   
#134680 dark blue  
  
custom.col <- data.frame(type = unique(imp\_data$use.wear.type),   
 col = c("#999999", "#52854c", "#c3d7a4", "#487bb6", "#a6cee3",  
 "#9a0f0f",   
 "#d16103", "#ffdb6d", "#985633", "#134680", "#05100c"))

## Plots per tool type

### Keilmesser

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_KM)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_KM[[i]][["use.wear.type"]]), "col"]  
 KM\_pie <- ggplot(data = sp\_KM[[i]], aes(x = area, fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +  
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
   
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_KM\_pie\_",   
 names(sp\_KM)[i], ".pdf")  
 ggsave(filename = file\_out, plot = KM\_pie, path = dir\_out, device = "pdf")  
   
}

### Keilmesser from Buhlen

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_KM\_Buhlen)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_KM\_Buhlen[[i]][["use.wear.type"]]), "col"]  
 KM\_Buhlen\_pie <- ggplot(data = sp\_KM\_Buhlen[[i]], aes(x = area, fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +  
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
   
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_KM\_Buhlen\_pie\_",   
 names(sp\_KM\_Buhlen)[i], ".pdf")  
 ggsave(filename = file\_out, plot = KM\_Buhlen\_pie, path = dir\_out, device = "pdf")  
   
}

### Keilmesser from Balve

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_KM\_Balve)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_KM\_Balve[[i]][["use.wear.type"]]), "col"]  
 KM\_Balve\_pie <- ggplot(data = sp\_KM\_Balve[[i]], aes(x = area, fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +  
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
   
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_KM\_Balve\_pie\_",   
 names(sp\_KM\_Balve)[i], ".pdf")  
 ggsave(filename = file\_out, plot = KM\_Balve\_pie, path = dir\_out, device = "pdf")  
   
}

### Keilmesser from Ramioul

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_KM\_Ramioul)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_KM\_Ramioul[[i]][["use.wear.type"]]), "col"]  
 KM\_Ramioul\_pie <- ggplot(data = sp\_KM\_Ramioul[[i]], aes(x = area,   
 fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +  
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
   
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_KM\_Ramioul\_pie\_",   
 names(sp\_KM\_Ramioul)[i], ".pdf")  
 ggsave(filename = file\_out, plot = KM\_Ramioul\_pie, path = dir\_out, device = "pdf")  
   
}

### Pradnick scraper

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_PS)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_PS[[i]][["use.wear.type"]]), "col"]  
 PS\_pie <- ggplot(data = sp\_PS[[i]], aes(x = area, fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +   
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
  
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_PS\_pie\_",   
 names(sp\_PS)[i], ".pdf")  
 ggsave(filename = file\_out, plot = PS\_pie, path = dir\_out, device = "pdf")  
   
}

### Pradnick spall

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_LSS)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_LSS[[i]][["use.wear.type"]]), "col"]  
 LSS\_pie <- ggplot(data = sp\_LSS[[i]], aes(x = area, fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +   
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
   
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_LSS\_pie\_",   
 names(sp\_LSS)[i], ".pdf")  
 ggsave(filename = file\_out, plot = LSS\_pie, path = dir\_out, device = "pdf")  
   
}

### Scraper

# plots first as a bar plot and then converts into a pie chart   
for (i in seq\_along(sp\_S)){  
   
 col\_i <- custom.col[custom.col$type %in% unique(sp\_S[[i]][["use.wear.type"]]), "col"]  
 S\_pie <- ggplot(data = sp\_S[[i]], aes(x = area, fill = use.wear.type )) +  
 geom\_bar(stat = "count", width = 0.5) +   
 coord\_polar("y", start = 0) +  
 theme\_void() +   
 scale\_fill\_manual(values = col\_i) +  
 labs(fill = gsub("\\.", " ", "use-wear type"))   
   
  
 # saves the plots   
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_S\_pie\_",   
 names(sp\_S)[i], ".pdf")  
 ggsave(filename = file\_out, plot = S\_pie, path = dir\_out, device = "pdf")  
   
}

# Save data

## Format name of output file

file\_out <- "Use-wear\_qualitative\_stats"

The file will be saved as “~/analysis/summary\_stats/.[ext]”.

## Write to XLSX

write.xlsx(list(all = PCT\_all, KM = PCT\_KM, PCT\_PS = PS, PCT\_LSS = LSS, PCT\_S = S),   
 file = paste0(dir\_out, file\_out, ".xlsx"))

# 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] doBy\_4.6.8 dplyr\_1.0.3 wesanderson\_0.3.6 ggplot2\_3.3.3   
[5] chron\_2.3-56 R.utils\_2.10.1 R.oo\_1.24.0 R.methodsS3\_1.8.1  
[9] openxlsx\_4.2.3   
  
loaded via a namespace (and not attached):  
 [1] zip\_2.1.1 Rcpp\_1.0.6 compiler\_4.0.2 pillar\_1.4.7   
 [5] digest\_0.6.27 lattice\_0.20-41 evaluate\_0.14 lifecycle\_0.2.0   
 [9] tibble\_3.0.6 gtable\_0.3.0 pkgconfig\_2.0.3 rlang\_0.4.10   
[13] Matrix\_1.2-18 DBI\_1.1.1 yaml\_2.2.1 xfun\_0.20   
[17] withr\_2.4.1 stringr\_1.4.0 knitr\_1.31 generics\_0.1.0   
[21] vctrs\_0.3.6 tidyselect\_1.1.0 grid\_4.0.2 glue\_1.4.2   
[25] R6\_2.5.0 rmarkdown\_2.6 farver\_2.0.3 tidyr\_1.1.2   
[29] purrr\_0.3.4 magrittr\_2.0.1 backports\_1.2.1 MASS\_7.3-51.6   
[33] scales\_1.1.1 htmltools\_0.5.1.1 ellipsis\_0.3.1 assertthat\_0.2.1   
[37] colorspace\_2.0-0 Deriv\_4.1.2 labeling\_0.4.2 stringi\_1.5.3   
[41] munsell\_0.5.0 broom\_0.7.4 crayon\_1.4.0

RStudio version 1.3.1073.

END OF SCRIPT