Plot - Edge angle analysis - experimental data

Lisa Schunk

2021-02-04 16:36:40

# Goal of the script

This script plots all edge angle values. The three experiments will be treated separately.

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

Raw data must be located in ~/analysis/derived\_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(readxl)  
library(R.utils)  
library(ggplot2)

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

library(tools)  
library(tidyverse)

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(patchwork)

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

library(ggsci)  
library(ggfortify)

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

library(wesanderson)

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

library(doBy)

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

library(ggfortify)

# 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(file = basename(names(md5\_in)), checksum = md5\_in, row.names = NULL)

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

file checksum  
1 EdgeAngle\_experiment.xlsx ed7878fcffd202c8e363d138621a7ce7

# Load data into R object

imp\_data <- read.xlsx(xlsxFile = data\_file, sheet = 1, startRow = 1, colNames = TRUE,  
 rowNames = FALSE, skipEmptyRows = FALSE)   
str(imp\_data)

'data.frame': 29933 obs. of 16 variables:  
 $ section : chr "FLT4-10-1000strokes\_E1\_RE\_SEC-01\_local" "FLT4-10-1000strokes\_E1\_RE\_SEC-01\_local" "FLT4-10-1000strokes\_E1\_RE\_SEC-01\_local" "FLT4-10-1000strokes\_E1\_RE\_SEC-01\_local" ...  
 $ experiment : chr "'aVSn'-experiment" "'aVSn'-experiment" "'aVSn'-experiment" "'aVSn'-experiment" ...  
 $ ID : chr "FLT4-10" "FLT4-10" "FLT4-10" "FLT4-10" ...  
 $ strokes : chr "1000strokes" "1000strokes" "1000strokes" "1000strokes" ...  
 $ edge : chr "E1" "E1" "E1" "E1" ...  
 $ sec : chr "SEC-01" "SEC-01" "SEC-01" "SEC-01" ...  
 $ Raw.material : chr "flint" "flint" "flint" "flint" ...  
 $ Contact.material: chr "bone plate" "bone plate" "bone plate" "bone plate" ...  
 $ Task : chr "cutting" "cutting" "cutting" "cutting" ...  
 $ Edge.angle : chr "60°" "60°" "60°" "60°" ...  
 $ Angle\_number : num 1 2 3 4 5 6 7 8 9 10 ...  
 $ Distance\_origin : num 1 2 3 4 5 6 7 8 9 10 ...  
 $ Segment : num 2 2 2 2 2 2 2 2 2 2 ...  
 $ Three\_point : num 75.2 67 63.5 61.9 60.9 60.3 59.9 59.4 59.2 59 ...  
 $ Two\_lines : num 68.1 58 57.1 56.9 57.2 57.7 56.7 56.7 57.1 57.2 ...  
 $ Best\_fit : num 69.7 58.3 57 57 57.3 57 57.1 57.1 56.6 57.6 ...

The imported file is: “~/analysis/derived\_data/EdgeAngle\_experiment.xlsx”

# Prepare variables

## Define numeric variables

num.var <- 14:length(imp\_data)

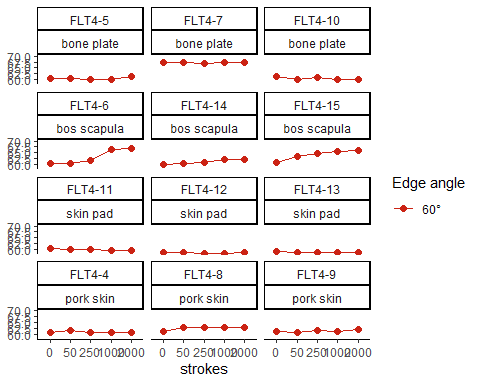
The following variables will be used:

[14] Three\_point  
[15] Two\_lines  
[16] Best\_fit

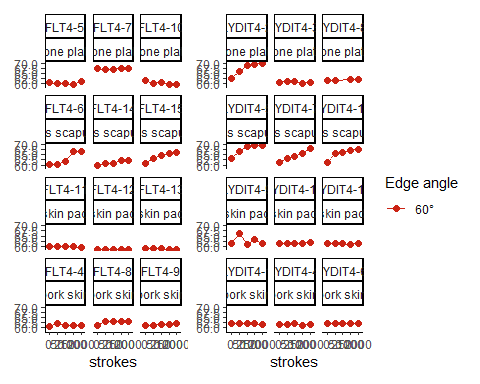
# Facet wrap to plot each experiment individually

## Artfifical VS natural experiment

# selects only the data from the aVSn experiment   
aVSn <- filter(imp\_data, experiment == "'aVSn'-experiment")  
aVSn <- filter(aVSn, Angle\_number == "3" | Angle\_number == "4" | Angle\_number == "5" |  
 Angle\_number == "6")  
aVSn <- filter(aVSn, sec == "SEC-02" | sec == "SEC-03" | sec == "SEC-04" | sec == "SEC-05" |  
 sec == "SEC-06" | sec == "SEC-07" | sec == "SEC-08")  
  
# adds a column that combines sample and location  
aVSn\_data <- unite(aVSn, ID\_cycle, c(ID, strokes), remove = FALSE)  
  
# computes the mean per sample   
aVSn\_mean <- summaryBy(. ~ ID\_cycle+ID+strokes+Contact.material+Raw.material+Edge.angle, data = aVSn\_data, FUN = mean)  
  
# gets new order   
aVSn\_mean$strokes <- factor(aVSn\_mean$strokes, levels=c("before", "50strokes", "250strokes",  
 "1000strokes", "2000strokes"))  
aVSn\_mean$ID <- factor(aVSn\_mean$ID, levels=c("FLT4-5", "FLT4-7", "FLT4-10", "FLT4-6",  
 "FLT4-14", "FLT4-15","FLT4-11", "FLT4-12",  
 "FLT4-13", "FLT4-4", "FLT4-8", "FLT4-9",  
 "LYDIT4-2", "LYDIT4-3", "LYDIT4-8","LYDIT4-5",  
 "LYDIT4-7", "LYDIT4-12", "LYDIT4-9",  
 "LYDIT4-10", "LYDIT4-11", "LYDIT4-1",  
 "LYDIT4-4", "LYDIT4-6"))   
  
# gets the min/max range of the data set   
range\_var <- range(aVSn\_mean[["Three\_point.mean"]])   
  
# plots   
# plots first the lydit samples  
p\_lydite <- ggplot(data = aVSn\_mean[grep("LYDIT", aVSn\_mean[["ID\_cycle"]]), ],   
 aes(x = strokes, y = Three\_point.mean, colour = Edge.angle)) +   
 geom\_point(size = 2) +  
 geom\_line(aes(group = ID)) +  
 facet\_wrap(ID ~ Contact.material, nrow = 4)+  
 theme\_classic()+  
 xlab("strokes") + ylab(NULL) +  
 labs(colour = "Edge angle") +  
 coord\_cartesian(ylim = range\_var) +   
 scale\_colour\_manual(values = "#CB2314") +  
 scale\_x\_discrete(breaks = c("before", "50strokes", "250strokes" , "1000strokes",  
 "2000strokes"), labels = c("0", "50", "250", "1000",  
 "2000"))  
  
  
# plots the flint samples   
p\_flint <- ggplot(data = aVSn\_mean[grep("FLT", aVSn\_mean[["ID\_cycle"]]), ],   
 aes(x = strokes, y = Three\_point.mean, colour = Edge.angle)) +   
 geom\_point(size = 2) +  
 geom\_line(aes(group = ID)) +  
 facet\_wrap(ID ~ Contact.material, nrow = 4)+  
 theme\_classic()+  
 xlab("strokes") + ylab(NULL) +  
 labs(colour = "Edge angle") +  
 coord\_cartesian(ylim = range\_var) +  
 scale\_colour\_manual(values = "#CB2314") +  
 scale\_x\_discrete(breaks = c("before", "50strokes", "250strokes" , "1000strokes",  
 "2000strokes"), labels = c("0", "50", "250", "1000",  
 "2000"))  
  
print(p\_flint)



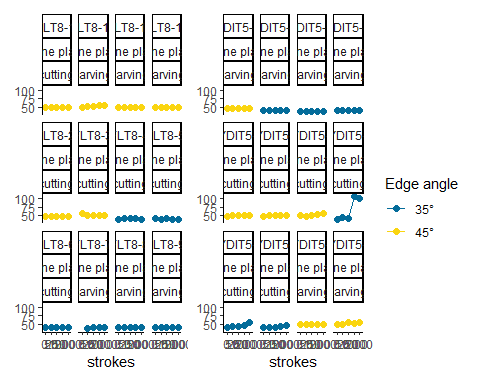
# combines the flint and the lydite plots   
p <- p\_flint + p\_lydite + plot\_layout(guides = 'collect')   
   
print(p)



#save to PDF  
file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_aVSn\_EA\_plot", ".pdf")  
ggsave(filename = file\_out, plot = p, path = dir\_out, device = "pdf",   
 width = 370, height = 280, units = "mm")

## Tool function experiment

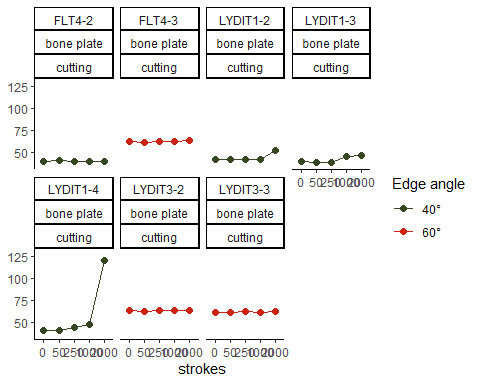
# selects only the data from the TFE experiment and the 'Three-point' method   
#TFE <- imp\_data[7460:20629,1:14]  
TFE <- filter(imp\_data, experiment == "tool\_function-experiment\_cutting" |   
 experiment == "tool\_function-experiment\_carving")  
TFE <- filter(TFE, Angle\_number == "3" | Angle\_number == "4" | Angle\_number == "5" |  
 Angle\_number == "6")  
TFE <- filter(TFE, sec == "SEC-02" | sec == "SEC-03" | sec == "SEC-04" | sec == "SEC-05" |  
 sec == "SEC-06" | sec == "SEC-07" | sec == "SEC-8")  
  
# adds a column that combines sample and location  
TFE\_data <- unite(TFE, ID\_cycle, c(ID, strokes), remove = FALSE)  
TFE\_mean <- summaryBy(. ~ ID\_cycle + ID + strokes + Contact.material +   
 Raw.material + Edge.angle + Task, data = TFE\_data, FUN = mean)  
  
# computes the mean per sample   
TFE\_final <- TFE\_mean[c(0:49, 51:80, 83:122),]  
# removes the test sample LYDIT5-14 (carving)   
  
# gets new order   
TFE\_final$strokes <- factor(TFE\_final$strokes, levels=c("before", "50strokes", "250strokes",  
 "1000strokes", "2000strokes"))  
  
  
# plots   
# plots first the lydite samples  
p\_lydite2 <- ggplot(data = TFE\_final[grep("LYDIT", TFE\_final[["ID\_cycle"]]), ],   
 aes(x = strokes, y = Three\_point.mean, colour = Edge.angle)) +   
 geom\_point(size = 2) +  
 geom\_line(aes(group = ID)) +  
 facet\_wrap(ID ~ Contact.material+Task, nrow = 3)+  
 theme\_classic()+  
 xlab("strokes") + ylab(NULL) +  
 labs(colour = "Edge angle") +  
 ylim(30, 110) +   
 scale\_colour\_manual(values = c("#046C9A", "#FAD510")) +  
 scale\_x\_discrete(breaks = c("before", "50strokes", "250strokes" ,  
 "1000strokes", "2000strokes"), labels =   
 c("0", "50", "250", "1000", "2000"))  
  
  
# plots the flint samples   
p\_flint2 <- ggplot(data = TFE\_final[grep("FLT", TFE\_final[["ID\_cycle"]]), ],   
 aes(x = strokes, y = Three\_point.mean, colour = Edge.angle)) +   
 geom\_point(size = 2) +  
 geom\_line(aes(group = ID)) +  
 facet\_wrap(ID ~ Contact.material+Task, nrow = 3)+  
 theme\_classic()+  
 xlab("strokes") + ylab(NULL) +  
 labs(colour = "Edge angle") +  
 ylim(30, 110) +   
 #coord\_cartesian(ylim = range\_var) +  
 scale\_colour\_manual(values = c("#046C9A", "#FAD510")) +  
 scale\_x\_discrete(breaks = c("before", "50strokes", "250strokes" , "1000strokes",  
 "2000strokes"), labels = c("0", "50", "250", "1000",  
 "2000"))  
  
   
# combines the flint and the lydite plots   
p2 <- p\_flint2 + p\_lydite2 + plot\_layout(guides = 'collect')   
   
print(p2)



#save to PDF  
 file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_TFE\_EA\_plot", ".pdf")  
 ggsave(filename = file\_out, plot = p2, path = dir\_out, device = "pdf", width = 370,   
 height = 280, units = "mm")

## Initial experiment

# selects only the data from the initial experiment   
IE <- filter(imp\_data, experiment == "initial\_experiment")  
# filters the first two and the two last sections out   
IE <- filter(IE, Angle\_number == "3" | Angle\_number == "4" | Angle\_number == "5" |  
 Angle\_number == "6")  
IE <- filter(IE, sec == "SEC-02" | sec == "SEC-03" | sec == "SEC-04" | sec == "SEC-05" |   
 sec == "SEC-06" | sec == "SEC-07" | sec == "SEC-8")  
  
  
# adds a column that combines sample and location  
IE\_data <- unite(IE, ID\_cycle, c(ID, strokes), remove = FALSE)  
  
# computes the mean per sample   
IE\_mean <- summaryBy(. ~ ID\_cycle + ID + strokes + Contact.material +  
 Raw.material + Edge.angle + Task, data = IE\_data, FUN = mean)  
  
# gets new order   
IE\_mean$strokes <- factor(IE\_mean$strokes, levels=c("before", "50strokes", "250strokes",  
 "1000strokes", "2000strokes"))  
  
  
# plots   
# plots the lydite and flint samples  
p3 <- ggplot(data = IE\_mean, aes(x = strokes, y = Three\_point.mean, colour = Edge.angle)) +   
 geom\_point(size = 2) +  
 geom\_line(aes(group = ID)) +  
 facet\_wrap(ID ~ Contact.material+Task, ncol = 4)+  
 theme\_classic()+  
 xlab("strokes") + ylab(NULL) +  
 labs(colour = "Edge angle") +  
 ylim(35, 130) +   
 scale\_colour\_manual(values = c("#354823", "#CB2314")) +  
 scale\_x\_discrete(breaks = c("before", "50strokes", "250strokes" , "1000strokes",  
 "2000strokes"), labels = c("0", "50", "250", "1000",  
 "2000"))  
  
  
print(p3)



#save to PDF  
file\_out <- paste0(file\_path\_sans\_ext(info\_in[["file"]]), "\_IE\_EA\_plot", ".pdf")  
ggsave(filename = file\_out, plot = p3, path = dir\_out, device = "pdf",width = 250,   
 height = 170, units = "mm")

## Show files information

files\_out <- c(paste0(dir\_out, file\_out, ".xlsx"), paste0(dir\_out, file\_out, ".Rbin"))  
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 plotsEdgeAngle\_experiment\_IE\_EA\_plot.pdf.xlsx <NA>  
2 plotsEdgeAngle\_experiment\_IE\_EA\_plot.pdf.Rbin <NA>

# 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 wesanderson\_0.3.6 ggfortify\_0.4.11 ggsci\_2.9   
 [5] patchwork\_1.1.1 forcats\_0.5.1 stringr\_1.4.0 dplyr\_1.0.3   
 [9] purrr\_0.3.4 readr\_1.4.0 tidyr\_1.1.2 tibble\_3.0.6   
[13] tidyverse\_1.3.0 ggplot2\_3.3.3 R.utils\_2.10.1 R.oo\_1.24.0   
[17] R.methodsS3\_1.8.1 readxl\_1.3.1 openxlsx\_4.2.3   
  
loaded via a namespace (and not attached):  
 [1] Rcpp\_1.0.6 lattice\_0.20-41 lubridate\_1.7.9.2 assertthat\_0.2.1   
 [5] digest\_0.6.27 R6\_2.5.0 cellranger\_1.1.0 backports\_1.2.1   
 [9] reprex\_1.0.0 evaluate\_0.14 highr\_0.8 httr\_1.4.2   
[13] pillar\_1.4.7 rlang\_0.4.10 rstudioapi\_0.13 Matrix\_1.2-18   
[17] rmarkdown\_2.6 labeling\_0.4.2 munsell\_0.5.0 broom\_0.7.4   
[21] compiler\_4.0.2 Deriv\_4.1.2 modelr\_0.1.8 xfun\_0.20   
[25] pkgconfig\_2.0.3 htmltools\_0.5.1.1 tidyselect\_1.1.0 gridExtra\_2.3   
[29] crayon\_1.4.0 dbplyr\_2.0.0 withr\_2.4.1 MASS\_7.3-51.6   
[33] grid\_4.0.2 jsonlite\_1.7.2 gtable\_0.3.0 lifecycle\_0.2.0   
[37] DBI\_1.1.1 magrittr\_2.0.1 scales\_1.1.1 zip\_2.1.1   
[41] cli\_2.3.0 stringi\_1.5.3 farver\_2.0.3 fs\_1.5.0   
[45] xml2\_1.3.2 ellipsis\_0.3.1 generics\_0.1.0 vctrs\_0.3.6   
[49] glue\_1.4.2 hms\_1.0.0 yaml\_2.2.1 colorspace\_2.0-0   
[53] rvest\_0.3.6 knitr\_1.31 haven\_2.3.1

RStudio version 1.3.1073.

END OF SCRIPT