Equotip hardness measurements

Joao Marreiros and David Nora

2021-07-14

# Goal of the script

This script reads the csv file (measurements have been generated with the Equotip Leeb C rebound) and formats the data for a statistical analysis.  
The script will:

1. Read in the original csv file and organise the data
2. Plot the data
3. Write an XLSX-file and save an R object ready for further analysis in R

Imported files are in: ‘../analysis/raw\_data’

Figures are saved in: ‘../analysis/plots’

Tables are saved in: ‘../analysis/derived\_data’

# Load packages

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.2 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(AICcmodavg)  
library(ggplot2)  
library(utils)  
library(qwraps2)  
library(kableExtra)

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

library(doBy)

##   
## Attaching package: 'doBy'

## The following object is masked from 'package:dplyr':  
##   
## order\_by

# Read in original xlsx-file

# List all CSV files in dir\_in  
  
imp\_data <- read\_csv("../raw\_data/data.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## ID = col\_character(),  
## rawmaterial = col\_character(),  
## Date = col\_character(),  
## Time = col\_time(format = ""),  
## M1 = col\_double(),  
## M2 = col\_double(),  
## M3 = col\_double(),  
## M4 = col\_double(),  
## M5 = col\_double(),  
## M6 = col\_double(),  
## M7 = col\_double(),  
## M8 = col\_double(),  
## M9 = col\_double(),  
## M10 = col\_double()  
## )

# Organize data

# organizing data  
longdata <- imp\_data %>%  
 gather("M1", "M2","M3", "M4", "M5","M6","M7", "M8", "M9", "M10", key = Measurment, value = HLC)  
  
longdata

## # A tibble: 230 x 6  
## ID rawmaterial Date Time Measurment HLC  
## <chr> <chr> <chr> <time> <chr> <dbl>  
## 1 QTZ3-6 quartzite 09/06/2021 14:27:34 M1 876  
## 2 QTZ3-5 quartzite 09/06/2021 14:22:10 M1 919  
## 3 QTZ3-4 quartzite 09/06/2021 14:18:29 M1 913  
## 4 QTZ3-3 quartzite 09/06/2021 14:15:35 M1 822  
## 5 QTZ3-2 quartzite 09/06/2021 14:12:22 M1 863  
## 6 QTZ3-1 quartzite 09/06/2021 14:05:18 M1 910  
## 7 OBS4-6 obsidian 09/06/2021 13:59:22 M1 927  
## 8 OBS4-5 obsidian 09/06/2021 13:51:50 M1 969  
## 9 OBS4-4 obsidian 09/06/2021 13:48:41 M1 966  
## 10 OBS4-3 obsidian 09/06/2021 13:43:07 M1 957  
## # … with 220 more rows

write\_csv(longdata, "../derived\_data/longdata.csv")

# Data analsysis - descriptive stats

# descriptive statistics   
  
nminmaxmeanmedsd <- function(x){  
 y <- x[!is.na(x)]  
 n\_test <- length(y)  
 min\_test <- min(y)  
 max\_test <- max(y)  
 mean\_test <- mean(y)  
 med\_test <- median(y)  
 sd\_test <- sd(y)  
 out <- c(n\_test, min\_test, max\_test, mean\_test, med\_test, sd\_test)  
 names(out) <- c("n", "min", "max", "mean", "median", "sd")  
 return(out)  
}  
  
num.var <- 6:length(longdata)  
stats <- summaryBy(.~rawmaterial, data=longdata[c("rawmaterial", names(longdata)[num.var])], FUN=nminmaxmeanmedsd)  
  
stats

## # A tibble: 4 x 7  
## rawmaterial HLC.n HLC.min HLC.max HLC.mean HLC.median HLC.sd  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 dacite 60 865 979 958. 963 18.5  
## 2 flint 50 900 972 954. 960. 17.5  
## 3 obsidian 60 893 970 952. 958. 16.0  
## 4 quartzite 60 735 939 886. 894 37.7

write\_csv(stats, "../derived\_data/stats.csv")

# ANOVA analysis

# anova  
#longdata$rawmaterial <- as.factor(longdata$rawmaterial)  
  
anova <- aov(HLC ~ rawmaterial, data = longdata)  
anovafactor <- summary(aov(HLC ~ factor(rawmaterial), data = longdata))  
one <- oneway.test(HLC ~ rawmaterial, data = longdata)  
tuk <- TukeyHSD(aov(HLC ~ factor(rawmaterial), data = longdata))  
  
anova

## Call:  
## aov(formula = HLC ~ rawmaterial, data = longdata)  
##   
## Terms:  
## rawmaterial Residuals  
## Sum of Squares 213345.5 134127.6  
## Deg. of Freedom 3 226  
##   
## Residual standard error: 24.36155  
## Estimated effects may be unbalanced

anovafactor

## Df Sum Sq Mean Sq F value Pr(>F)   
## factor(rawmaterial) 3 213345 71115 119.8 <2e-16 \*\*\*  
## Residuals 226 134128 593   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

one

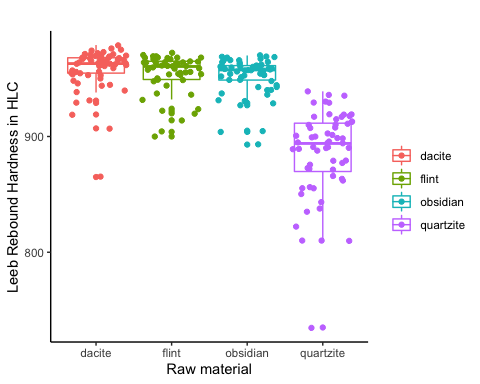
##   
## One-way analysis of means (not assuming equal variances)  
##   
## data: HLC and rawmaterial  
## F = 62.731, num df = 3.00, denom df = 122.33, p-value < 2.2e-16

tuk

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = HLC ~ factor(rawmaterial), data = longdata)  
##   
## $`factor(rawmaterial)`  
## diff lwr upr p adj  
## flint-dacite -4.556667 -16.63054 7.517202 0.7628076  
## obsidian-dacite -5.800000 -17.31198 5.711983 0.5613191  
## quartzite-dacite -72.566667 -84.07865 -61.054684 0.0000000  
## obsidian-flint -1.243333 -13.31720 10.830536 0.9933580  
## quartzite-flint -68.010000 -80.08387 -55.936131 0.0000000  
## quartzite-obsidian -66.766667 -78.27865 -55.254684 0.0000000

# Data analsysis - plot

# boxplot  
data\_plot <- ggplot (longdata, aes(rawmaterial, HLC, color = rawmaterial)) +   
 theme\_classic() +   
 theme(legend.title = element\_blank()) +  
 geom\_boxplot() +  
 geom\_jitter() + labs(x="Raw material", y="Leeb Rebound Hardness in HLC", title="")   
  
print(data\_plot)



# sessionInfo() and RStudio version

sessionInfo()

## R version 4.0.4 (2021-02-15)  
## Platform: x86\_64-apple-darwin17.0 (64-bit)  
## Running under: macOS Catalina 10.15.7  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] doBy\_4.6.10 kableExtra\_1.3.4 qwraps2\_0.5.2 AICcmodavg\_2.3-1  
## [5] forcats\_0.5.1 stringr\_1.4.0 dplyr\_1.0.7 purrr\_0.3.4   
## [9] readr\_1.4.0 tidyr\_1.1.3 tibble\_3.1.2 ggplot2\_3.3.5   
## [13] tidyverse\_1.3.1   
##   
## loaded via a namespace (and not attached):  
## [1] httr\_1.4.2 VGAM\_1.1-5 jsonlite\_1.7.2   
## [4] viridisLite\_0.4.0 splines\_4.0.4 modelr\_0.1.8   
## [7] microbenchmark\_1.4-7 assertthat\_0.2.1 highr\_0.9   
## [10] sp\_1.4-5 stats4\_4.0.4 cellranger\_1.1.0   
## [13] yaml\_2.2.1 pillar\_1.6.1 backports\_1.2.1   
## [16] lattice\_0.20-44 glue\_1.4.2 digest\_0.6.27   
## [19] rvest\_1.0.0 colorspace\_2.0-2 htmltools\_0.5.1.1   
## [22] Matrix\_1.3-4 plyr\_1.8.6 pkgconfig\_2.0.3   
## [25] broom\_0.7.8 raster\_3.4-13 curry\_0.1.1   
## [28] haven\_2.4.1 xtable\_1.8-4 scales\_1.1.1   
## [31] webshot\_0.5.2 svglite\_2.0.0 farver\_2.1.0   
## [34] generics\_0.1.0 ellipsis\_0.3.2 withr\_2.4.2   
## [37] cli\_2.5.0 survival\_3.2-11 magrittr\_2.0.1   
## [40] crayon\_1.4.1 readxl\_1.3.1 evaluate\_0.14   
## [43] fs\_1.5.0 fansi\_0.5.0 nlme\_3.1-152   
## [46] MASS\_7.3-54 xml2\_1.3.2 tools\_4.0.4   
## [49] hms\_1.1.0 lifecycle\_1.0.0 munsell\_0.5.0   
## [52] reprex\_2.0.0 Deriv\_4.1.3 compiler\_4.0.4   
## [55] systemfonts\_1.0.2 rlang\_0.4.11 grid\_4.0.4   
## [58] rstudioapi\_0.13 labeling\_0.4.2 rmarkdown\_2.9   
## [61] gtable\_0.3.0 codetools\_0.2-18 DBI\_1.1.1   
## [64] R6\_2.5.0 lubridate\_1.7.10 knitr\_1.33   
## [67] utf8\_1.2.1 stringi\_1.6.2 parallel\_4.0.4   
## [70] unmarked\_1.1.1 Rcpp\_1.0.6 vctrs\_0.3.8   
## [73] dbplyr\_2.1.1 tidyselect\_1.1.1 xfun\_0.24