

# SOM - Read and data plotting from the manuscript Paixao et al. 2021 - JASr

Paixão et al.2021 - JARs

2021-04-07 12:04:56

## Brief description of the script

This R markdown document reads, summarizes and plots data for the manuscript *Paixão et al. 2021 Using Mechanical experiments to study Ground Stone Tool use: exploring the formation of percussive and grinding wear traces on Limestone tools. Journal of Archaeological Science: Reports*

The document contains:

1. Manuscript tables
2. Manuscript figures (data analysis)
3. Supplementary material, including extra tables and figures (data analysis)

This R project and respective scripts follow the procedures described by Marwick et al. 2017.

To compile this markdown document do not delete or move files from their original folders. Please note that most of the tables and figures in this file do not match the numbering in the original manuscript.

The authors would like to thank Ivan Calandra and Lisa Schunk for their help and contribution on several chunks of code included here in the script (pieces of code are also adapted from Calandra et al. 2019, Pedernana et al. 2020a, 2020b).

For any questions, comments and inputs, please contact:

João Marreiros, marreiros@rgzm.de, or Eduardo Paixão, paixao@rgzm.de

## Load data into R project

*Imported files are in: './analysis/raw\_data'*

*Figures are saved in: './analysis/plots'*

*Tables are saved in: './analysis/derived\_data'*

## Load libraries

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```

## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.1.0      v dplyr  1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(utils)
library(knitr)
library(janitor)

##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##   chisq.test, fisher.test

library(kableExtra)

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##   group_rows

library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

library(doBy)

##
## Attaching package: 'doBy'

## The following object is masked from 'package:dplyr':
##
##   order_by

library(ggpubr)
library(ggfortify)
library(tools)

```

## Import datasets

```
gisdata <- read_csv("../raw_data/gisdata.csv")
```

```
##
## -- Column specification -----
## cols(
##   sample = col_character(),
##   cycle = col_character(),
##   parameter = col_character(),
##   motion = col_character(),
##   material = col_character(),
##   id = col_double(),
##   elev_min = col_double(),
##   elev_max = col_double(),
##   nparts = col_double(),
##   npoints = col_double(),
##   perimeter = col_double(),
##   area = col_double()
## )
```

```
confocaldata <- read.csv("../raw_data/confocaldata.csv", na.strings = "*****", encoding = "UTF-8")
```

```
data_file <- list.files("../raw_data/", 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)
```

In this study two datasets are used:

- 1) **gisdata.csv**: dataset for the QGIS analysis

```
str(gisdata)
```

```
## spec_tbl_df [355 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ sample   : chr [1:355] "id2-5" "id2-5" "id3-3" "id3-3" ...
## $ cycle    : chr [1:355] "before" "before" "before" "before" ...
## $ parameter: chr [1:355] "tri" "tri" "tri" "tri" ...
## $ motion   : chr [1:355] "Impact" "Impact" "Impact" "Impact" ...
## $ material : chr [1:355] "Flint" "Flint" "Flint" "Flint" ...
## $ id       : num [1:355] 0 1 0 1 2 3 4 0 1 0 ...
## $ elev_min : num [1:355] 0 0.01 0 0.01 0.02 0.03 0.04 0 0.01 0 ...
## $ elev_max : num [1:355] 0.01 0.02 0.01 0.02 0.03 0.04 0.05 0.01 0.02 0.01 ...
## $ nparts   : num [1:355] 1007 1010 131 165 47 ...
## $ npoints  : num [1:355] 67487 59648 14961 16143 3142 ...
## $ perimeter: num [1:355] 485.2 446 160.6 177.5 32.8 ...
## $ area     : num [1:355] 204.49 17.48 83.65 38.12 1.83 ...
## - attr(*, "spec")=
## .. cols(
## ..   sample = col_character(),
## ..   cycle = col_character(),
## ..   parameter = col_character(),
## ..   motion = col_character(),
## ..   material = col_character(),
```

```
## .. id = col_double(),
## .. elev_min = col_double(),
## .. elev_max = col_double(),
## .. nparts = col_double(),
## .. npoints = col_double(),
## .. perimeter = col_double(),
## .. area = col_double()
## .. )
```

2) **confocaldata.csv**: dataset for the Confocal microscopy surface texture analysis analysis

```
str(confocaldata)
```

```
## 'data.frame':   25 obs. of  54 variables:
## $ Name                : chr  "Lime2-5_LSM_50x075_suf1_Topo > Leveled (LS-plane) > Fo
## $ Created.on          : chr  "6/24/2020 12:03:05 PM" "6/24/2020 12:21:59 PM" "6/24/20
## $ sample              : chr  "id2-5" "id2-5" "id2-5" "id3-3" ...
## $ motion              : chr  "impact" "impact" "impact" "impact" ...
## $ workedmaterial      : chr  "flint" "flint" "flint" "flint" ...
## $ Studiable.type      : chr  "Surface" "Surface" "Surface" "Surface" ...
## $ Axis.name...X       : chr  "X" "X" "X" "X" ...
## $ Axis.length...X     : num  255 255 255 255 255 ...
## $ Axis.size...X       : int  3000 3000 3000 1024 1024 3000 3000 1024 3000 3000 ...
## $ Axis.spacing...X    : num  85.2 85.2 85.2 249.6 249.6 ...
## $ Axis.name...Y       : chr  "Y" "Y" "Y" "Y" ...
## $ Axis.length...Y     : num  255 255 255 255 255 ...
## $ Axis.size...Y       : int  3000 3000 3000 1024 1024 3000 3000 1024 3000 3000 ...
## $ Axis.spacing...Y    : num  85.2 85.2 85.2 249.6 249.6 ...
## $ Axis.name...Z       : chr  "Z" "Z" "Z" "Z" ...
## $ Layer.type...Z      : chr  "Topography" "Topography" "Topography" "Topography" ...
## $ Axis.length...Z     : num  8.3 30.6 14.9 48.6 23 ...
## $ Axis.size...Z       : int  128164 333950 130813 193492 260079 181249 180484 428496
## $ Axis.spacing...Z    : num  0.0647 0.0916 0.1139 0.2511 0.0884 ...
## $ NM.points.ratio...Z : int  0 0 0 0 0 0 0 0 0 ...
## $ Sq                  : num  0.836 4.619 2.321 5.348 3.491 ...
## $ Ssk                 : num  0.997 0.107 0.142 -0.491 0.201 ...
## $ Sku                 : num  8.63 4.48 3.11 5.97 3.46 ...
## $ Sp                  : num  4.8 15.02 7.46 23 11.88 ...
## $ Sv                  : num  3.49 15.58 7.45 25.58 11.11 ...
## $ Sz                  : num  8.3 30.6 14.9 48.6 23 ...
## $ Sa                  : num  0.572 3.244 1.819 3.887 2.699 ...
## $ Smr                 : num  0.464 0.497 0.448 0.18 0.354 ...
## $ Smc                 : num  0.775 5.691 2.944 5.799 4.436 ...
## $ Sxp                 : num  1.53 10.61 4.3 11.65 6.74 ...
## $ Sal                 : num  13.5 18.9 20.1 18.7 22.9 ...
## $ Str                 : num  0.371 0.416 0.592 0.468 0.803 ...
## $ Std                 : num  149 65 150 51 124 ...
## $ Sdq                 : num  0.328 1.153 0.688 1.126 0.897 ...
## $ Sdr                 : num  4.36 20.02 16.47 31.23 24.6 ...
## $ Vm                  : num  0.0866 0.3378 0.1331 0.3133 0.2114 ...
## $ Vv                  : num  0.861 6.029 3.078 6.113 4.648 ...
## $ Vmp                 : num  0.0866 0.3378 0.1331 0.3133 0.2114 ...
## $ Vmc                 : num  0.528 3.111 2.097 3.932 3.146 ...
```

## \$ Vvc	: num	0.769 5.335 2.824 5.33 4.303 ...
## \$ Vvv	: num	0.0923 0.694 0.2534 0.7826 0.3444 ...
## \$ Maximum.depth.of.furrows	: num	4.56 20.63 7.65 25.68 13.88 ...
## \$ Mean.depth.of.furrows	: num	0.962 4.63 2.49 5.112 3.932 ...
## \$ Mean.density.of.furrows	: num	4523 3830 4509 2286 2201 ...
## \$ First.direction	: num	1.50e+02 6.36e+01 2.66e-03 9.00e+01 9.00e+01 ...
## \$ Second.direction	: num	180 45 154 45 135 ...
## \$ Third.direction	: num	141.3 56.2 63.5 51.2 123.7 ...
## \$ Isotropy	: num	23.5 26.2 77.8 33.1 73.8 ...
## \$ Lengthscale.anisotropy.Sfrax.epLsar	: num	NA NA NA 0.000493 0.001218 ...
## \$ Lengthscale.anisotropy.NewEplsar	: num	NA NA NA 0.0177 0.0172 ...
## \$ Fractal.complexity.Asfc	: num	8.66 23.18 30.55 37.79 44.11 ...
## \$ Scale.of.max.complexity.Smfc	: num	1.71e+06 1.81e+08 2.71e+06 1.17e+01 1.52e+01 ...
## \$ HAsfc9	: num	0.449 2.496 0.388 0.544 0.254 ...
## \$ HAsfc81	: num	0.659 3.446 0.481 0.701 0.487 ...

# GIS analysis, Terrain analysis for Slope and TRI based on the 3D surface point clouds

## Slope

```
# Compute proportions for perimeter and area grouped by sample and GIS parameter

slope <- filter(gisdata, parameter == "slope")
slopebefore <- filter(slope, cycle == "before")
slopeafter <- filter(slope, cycle == "after")

# before experimental cycles (i.e. natural surfaces)

id2.5before <- filter(slopebefore, sample == "id2-5")
id3.3before <- filter(slopebefore, sample == "id3-3")
id3.8before <- filter(slopebefore, sample == "id3-8")
id3.9before <- filter(slopebefore, sample == "id3-9")
id6.1before <- filter(slopebefore, sample == "id6-1")
id6.3before <- filter(slopebefore, sample == "id6-3")
id6.6before <- filter(slopebefore, sample == "id6-6")
id6.7before <- filter(slopebefore, sample == "id6-7")

id2.5before <- id2.5before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.3before <- id3.3before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.8before <- id3.8before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.9before <- id3.9before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.1before <- id6.1before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)
```

```

id6.3before <- id6.3before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.6before <- id6.6before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.7before <- id6.7before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

# after experimental cycles

id2.5after <- filter(slopeafter, sample == "id2-5")
id3.3after <- filter(slopeafter, sample == "id3-3")
id3.8after <- filter(slopeafter, sample == "id3-8")
id3.9after <- filter(slopeafter, sample == "id3-9")
id6.1after <- filter(slopeafter, sample == "id6-1")
id6.3after <- filter(slopeafter, sample == "id6-3")
id6.6after <- filter(slopeafter, sample == "id6-6")
id6.7after <- filter(slopeafter, sample == "id6-7")

id2.5after <- id2.5after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.3after <- id3.3after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.8after <- id3.8after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.9after <- id3.9after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

```

```

id6.1after <- id6.1after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.3after <- id6.3after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.6after <- id6.6after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.7after <- id6.7after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

newslope <- do.call("rbind", list(id2.5before, id3.3before, id3.8before, id3.9before, id6.1before, id6.3before, id6.6before, id6.7before))

# save outputs
write_csv(newslope, "../derived_data/newslope.csv")

# Plot data

# Number of parts
impactdf <- filter(newslope, motion == "Impact")
grinding <- filter(newslope, motion == "Grinding")

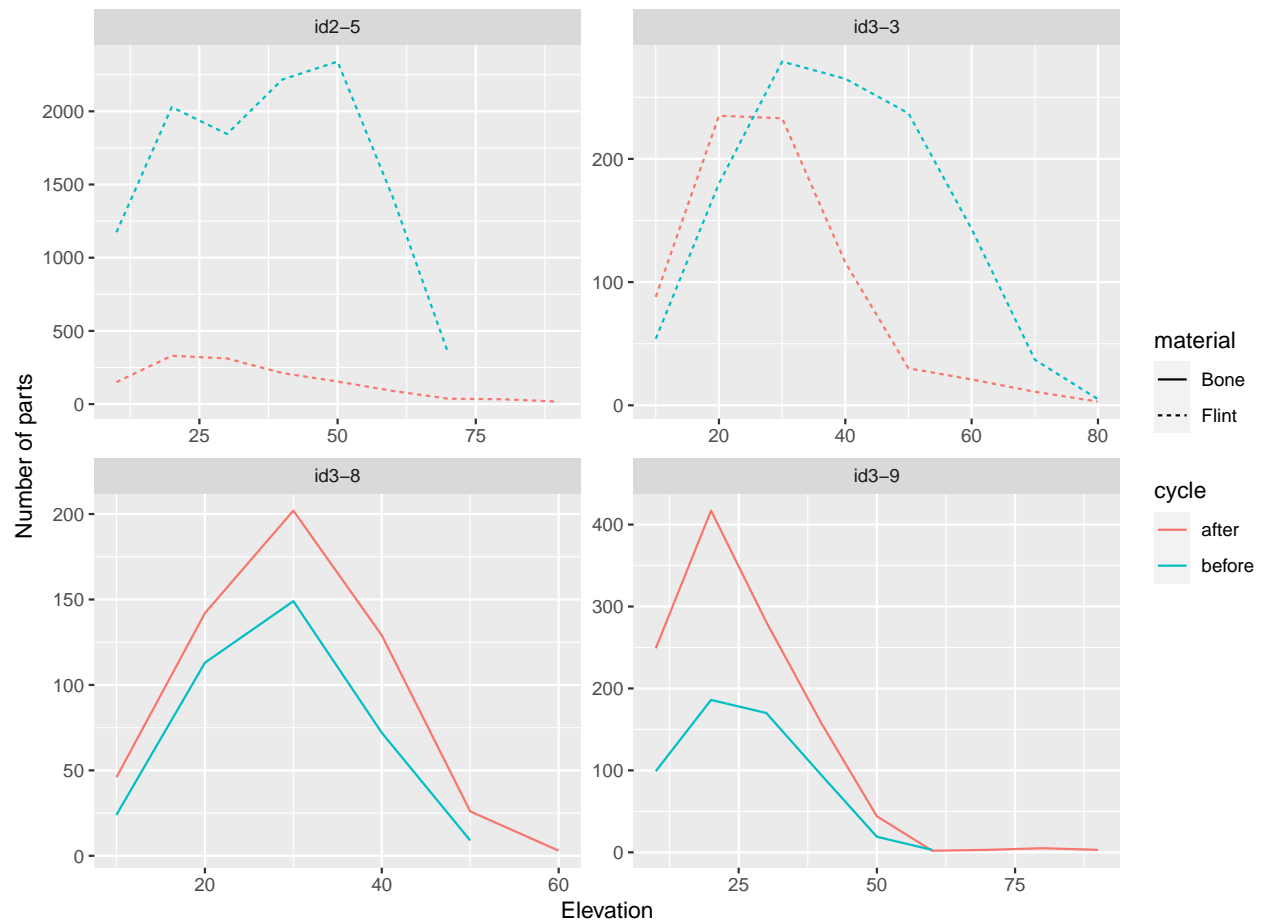
slopepartsexp_impac <- ggplot(impactdf, aes(x = elev_max, y = nparts, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("Slope impact experiment, number of parts") +
  ylab("Number of parts") +
  xlab("Elevation")

slopepartsexp_impac

```



Slope impact experiment, number of parts

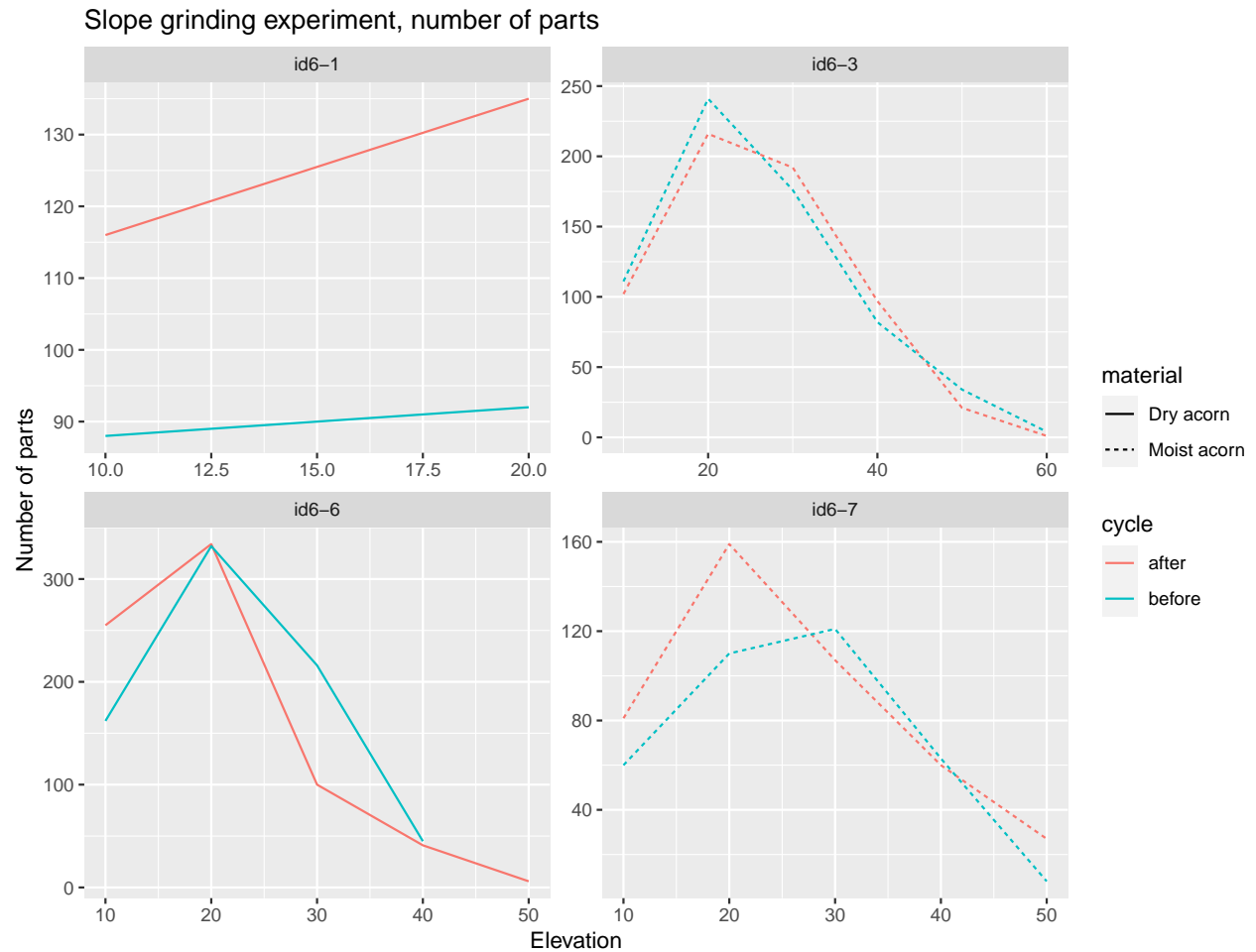


```
ggsave("../plots/slopepartsexp_impac.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
slopepartsexp_grind <- ggplot(grinding, aes(x = elev_max, y = nparts, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("Slope grinding experiment, number of parts") +
  ylab("Number of parts") +
  xlab("Elevation")
```

```
slopepartsexp_grind
```



```
ggsave("../plots/slopepartsexp_grind.png")
```

```
## Saving 8.5 x 6.5 in image
```

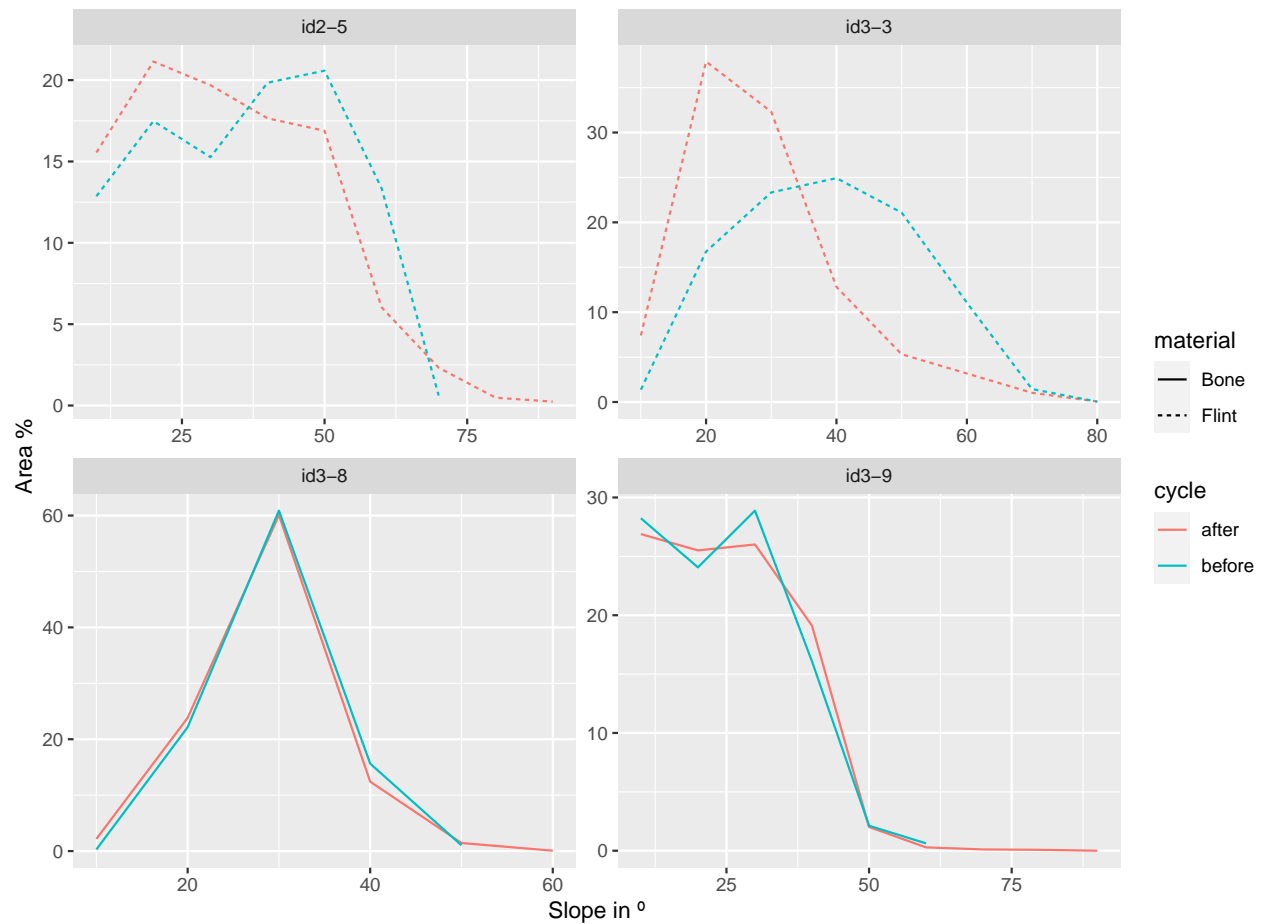
```
# Area %

impactdf <- filter(newslope, motion == "Impact")
grinding <- filter(newslope, motion == "Grinding")

areaimpact <- ggplot(impactdf, aes(x = elev_max, y = areaperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("Slope analysis, impact") +
  ylab("Area %") +
  xlab("Slope in °")

areaimpact
```

## Slope analysis, impact

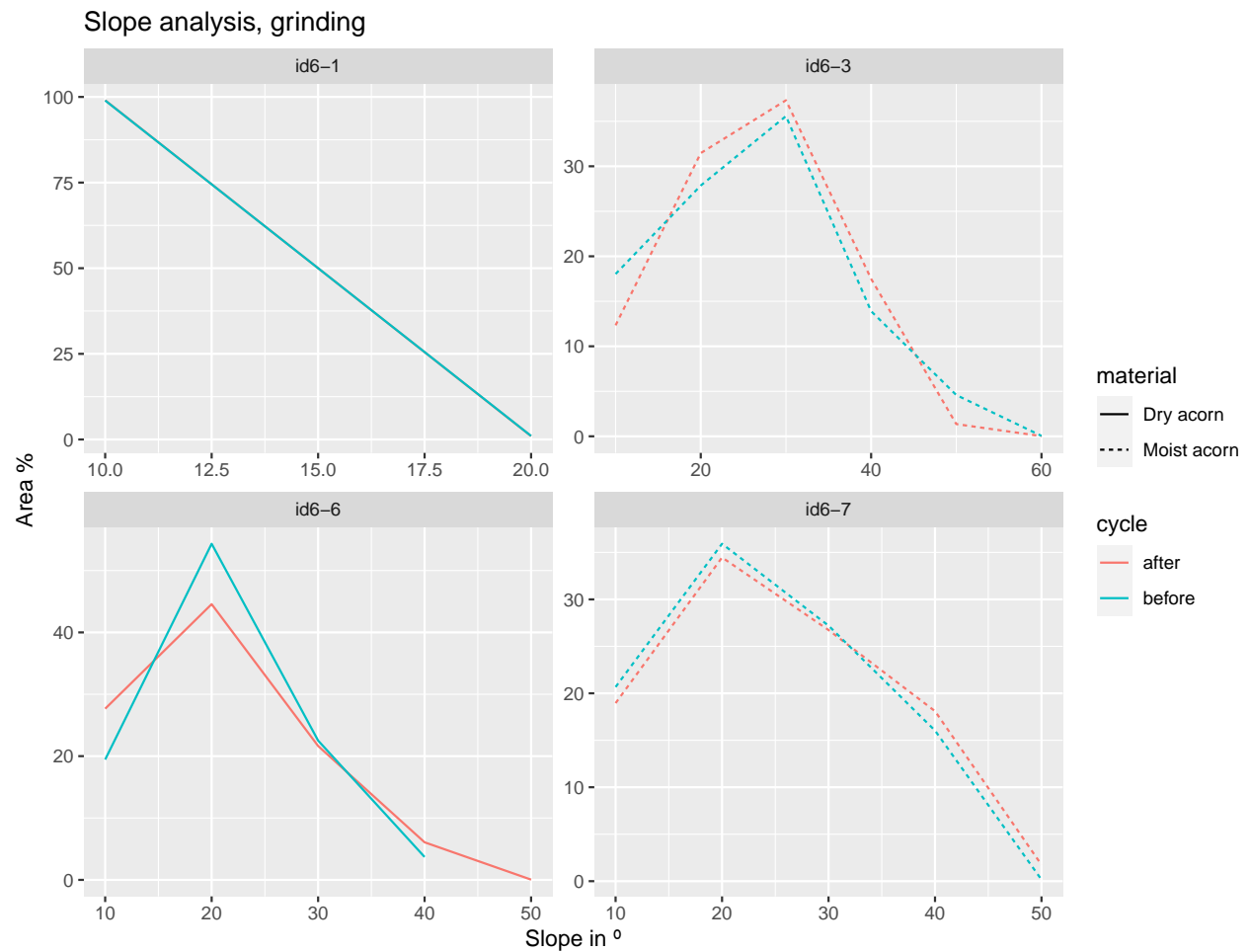


```
ggsave("../plots/slopeareaimpact.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
areagrinding <- ggplot(grinding, aes(x = elev_max, y = areaperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("Slope analysis, grinding") +
  ylab("Area %") +
  xlab("Slope in °")

areagrinding
```



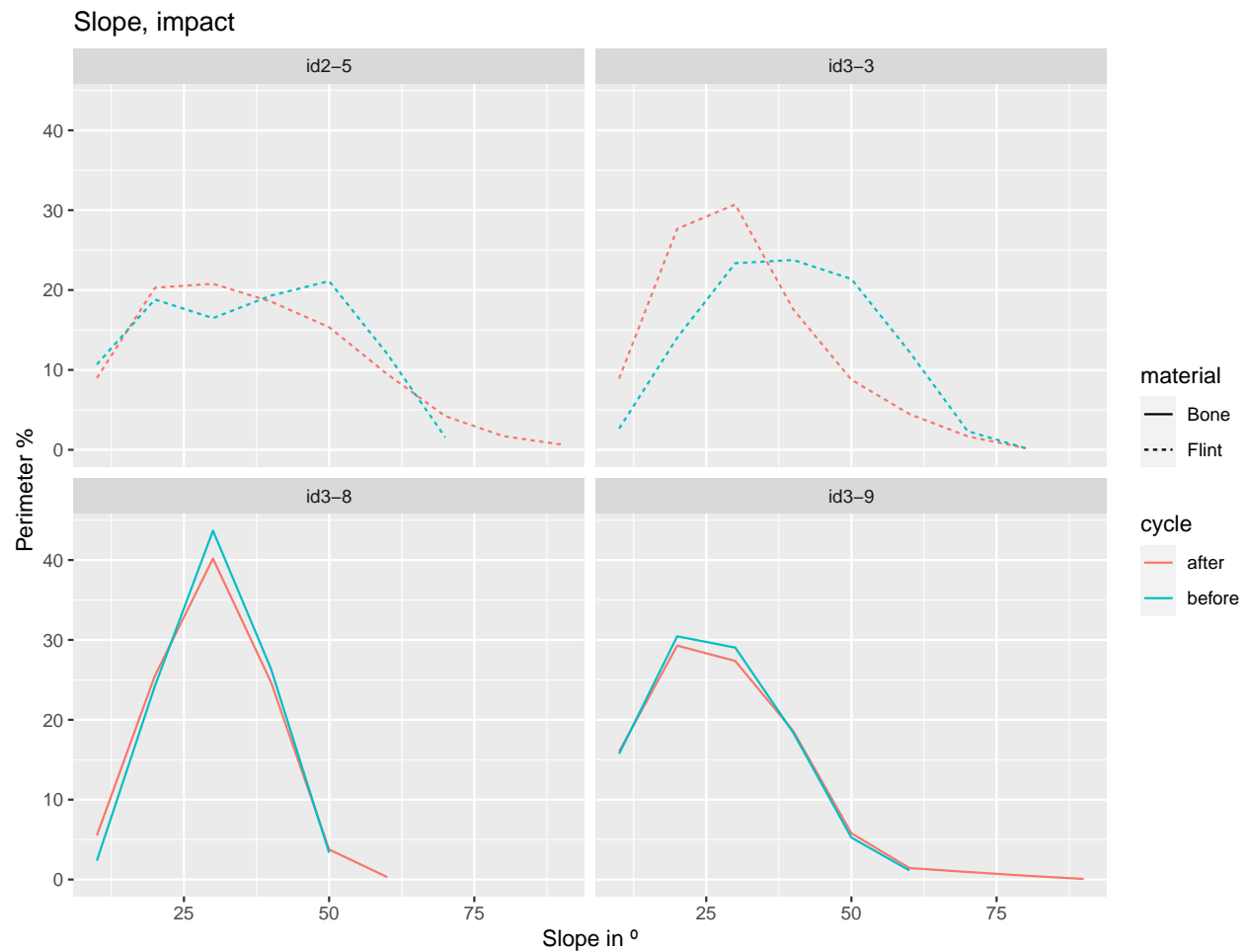
```
ggsave("../plots/slopeareagrinding.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
# Perimeter %

perimimpact <- ggplot(impactdf, aes(x = elev_max, y = perimperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample) +
  ggtitle("Slope, impact") +
  ylab("Perimeter %") +
  xlab("Slope in °")

perimimpact
```

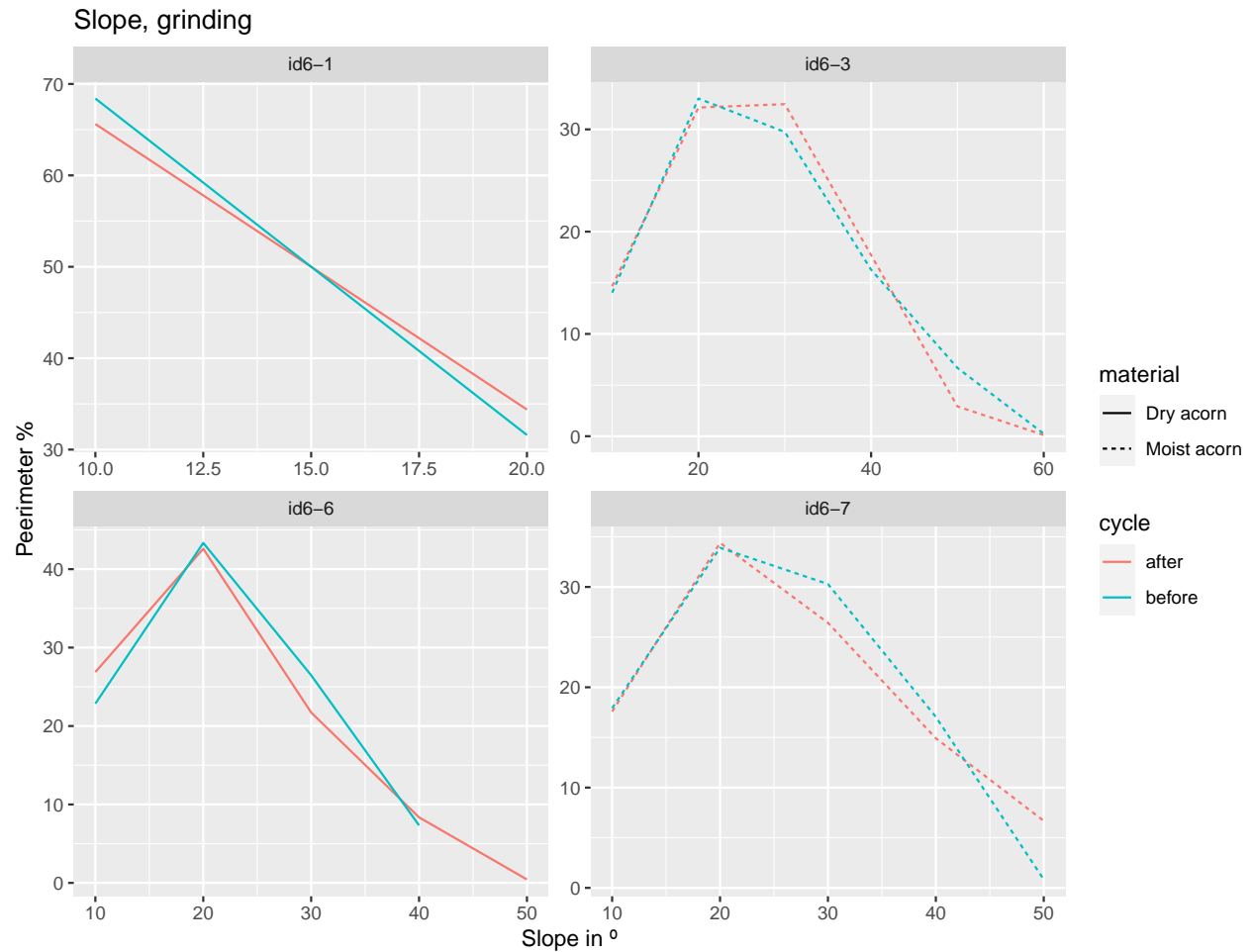


```
ggsave("../plots/slopeperimimpact.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
perimgrinding <- ggplot(grinding, aes(x = elev_max, y = perimperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("Slope, grinding") +
  ylab("Perimeter %") +
  xlab("Slope in °")

perimgrinding
```



```
ggsave("../plots/slopeperimgrinding.png")
```

```
## Saving 8.5 x 6.5 in image
```

## TRI (Terrain roughness index)

```
tri <- filter(gisdata, parameter == "tri")
tribefore <- filter(tri, cycle == "before")
triafter <- filter(tri, cycle == "after")

# before experimental cycles (i.e. natural surfaces)

id2.5before <- filter(tribefore, sample == "id2-5")
id3.3before <- filter(tribefore, sample == "id3-3")
id3.8before <- filter(tribefore, sample == "id3-8")
id3.9before <- filter(tribefore, sample == "id3-9")
id6.1before <- filter(tribefore, sample == "id6-1")
id6.3before <- filter(tribefore, sample == "id6-3")
id6.6before <- filter(tribefore, sample == "id6-6")
```

```

id6.7before <- filter(tribefore, sample == "id6-7")

id2.5before <- id2.5before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.3before <- id3.3before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.8before <- id3.8before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.9before <- id3.9before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.1before <- id6.1before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.3before <- id6.3before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.6before <- id6.6before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.7before <- id6.7before %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

# after experimental cycles

```

```

id2.5after <- filter(triafter, sample == "id2-5")
id3.3after <- filter(triafter, sample == "id3-3")
id3.8after <- filter(triafter, sample == "id3-8")
id3.9after <- filter(triafter, sample == "id3-9")
id6.1after <- filter(triafter, sample == "id6-1")
id6.3after <- filter(triafter, sample == "id6-3")
id6.6after <- filter(triafter, sample == "id6-6")
id6.7after <- filter(triafter, sample == "id6-7")

id2.5after <- id2.5after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.3after <- id3.3after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.8after <- id3.8after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id3.9after <- id3.9after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.1after <- id6.1after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.3after <- id6.3after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.6after <- id6.6after %>%
  group_by(sample) %>%
  mutate(
    areaperc = area / sum(area) * 100,
    perimperc = perimeter / sum(perimeter) * 100)

id6.7after <- id6.7after %>%
  group_by(sample) %>%

```



```

mutate(
  areaperc = area / sum(area) * 100,
  perimperc = perimeter / sum(perimeter) * 100)

newtri <- do.call("rbind", list(id2.5before, id3.3before, id3.8before, id3.9before, id6.1before, id6.3b

# save outputs

write_csv(newtri, "../derived_data/newtri.csv")

# Plot data

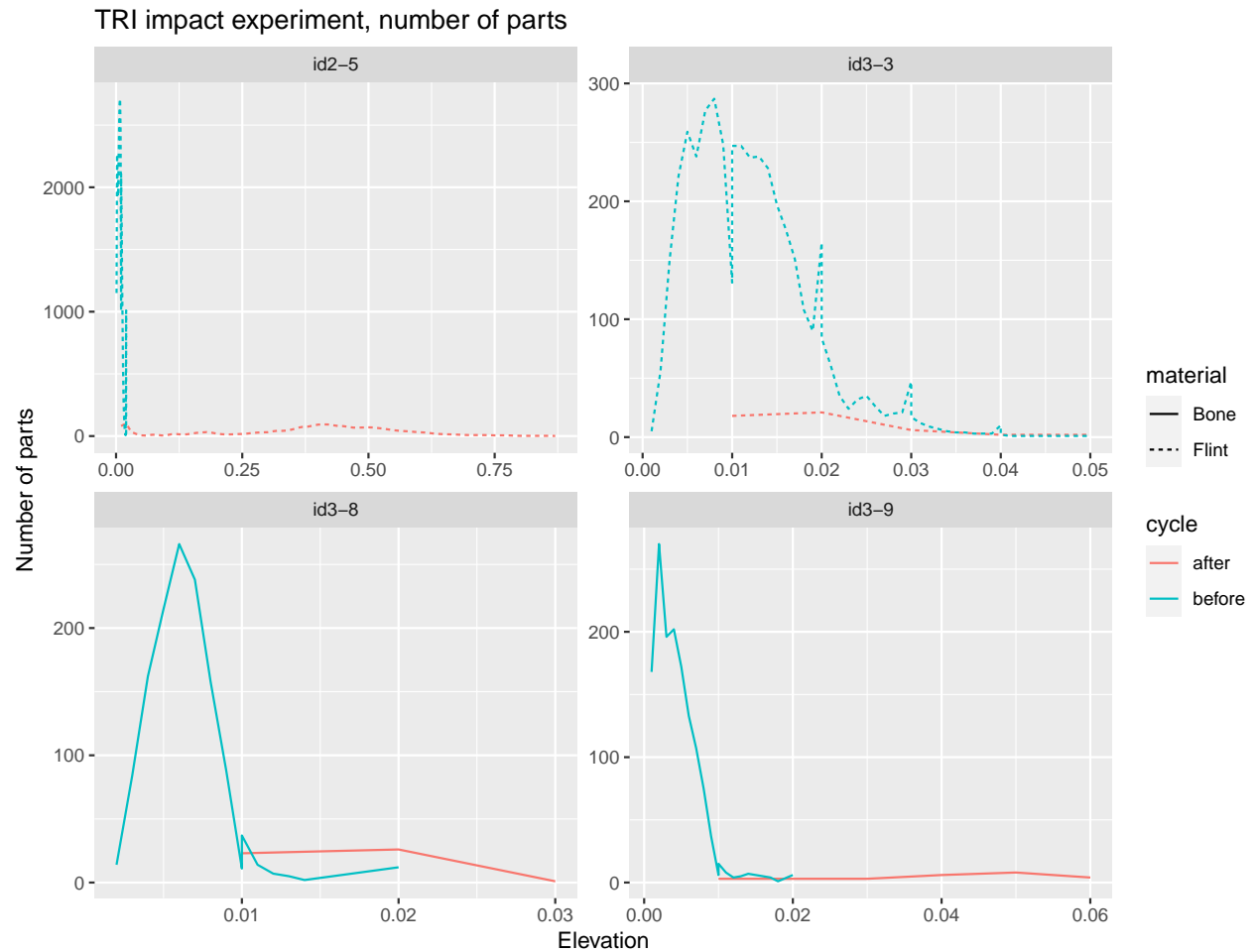
# Number of parts

# Motion
impactdf <- filter(newtri, motion == "Impact")
grinding <- filter(newtri, motion == "Grinding")

imapact_parts <- ggplot(impactdf, aes(x = elev_max, y = nparts, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("TRI impact experiment, number of parts") +
  ylab("Number of parts") +
  xlab("Elevation")

imapact_parts

```

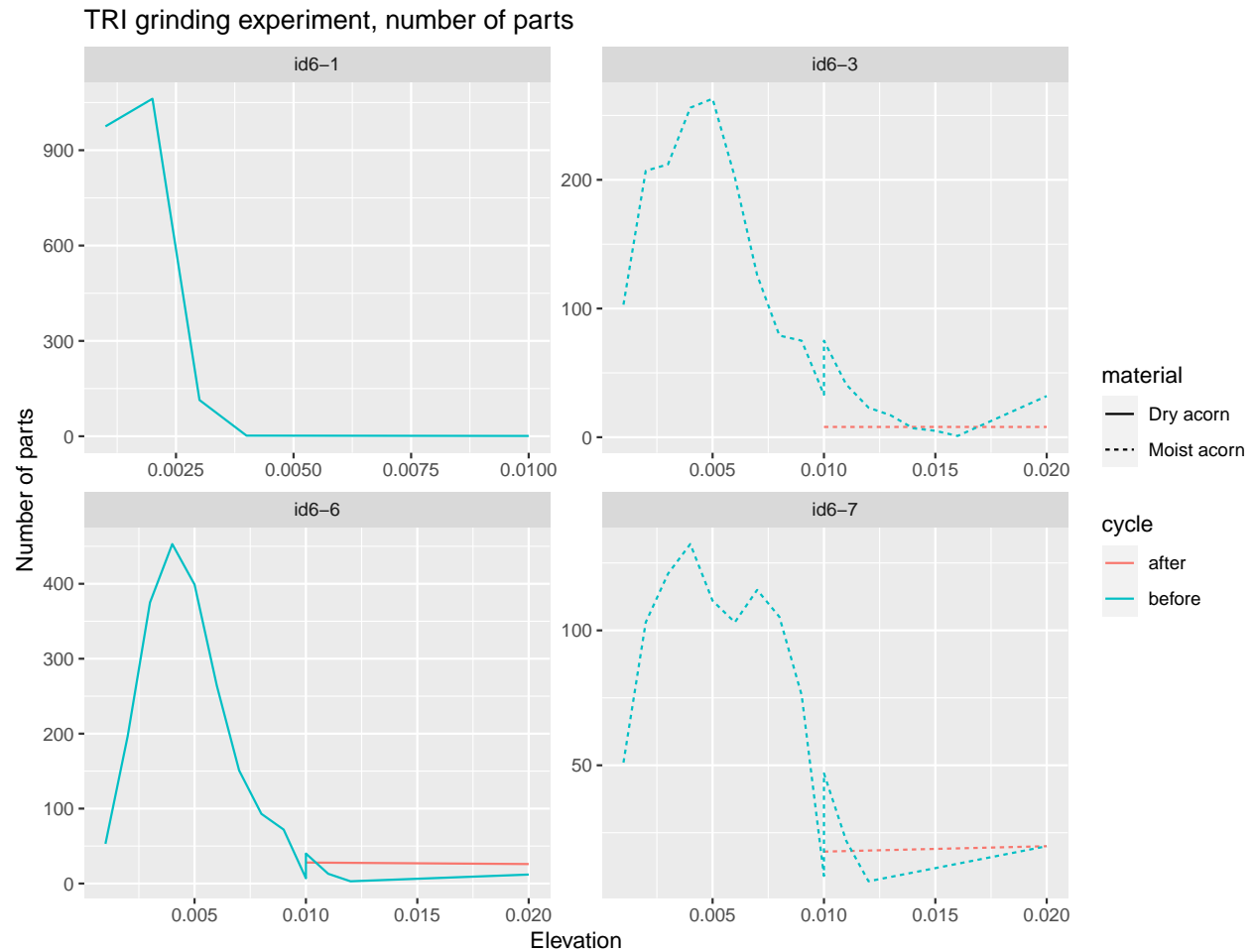


```
ggsave("../plots/tripartsexp_impac.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
grinding_parts <- ggplot(grinding, aes(x = elev_max, y = nparts, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("TRI grinding experiment, number of parts") +
  ylab("Number of parts") +
  xlab("Elevation")
```

```
grinding_parts
```



```
ggsave("../plots/tripartsexp_grind.png")
```

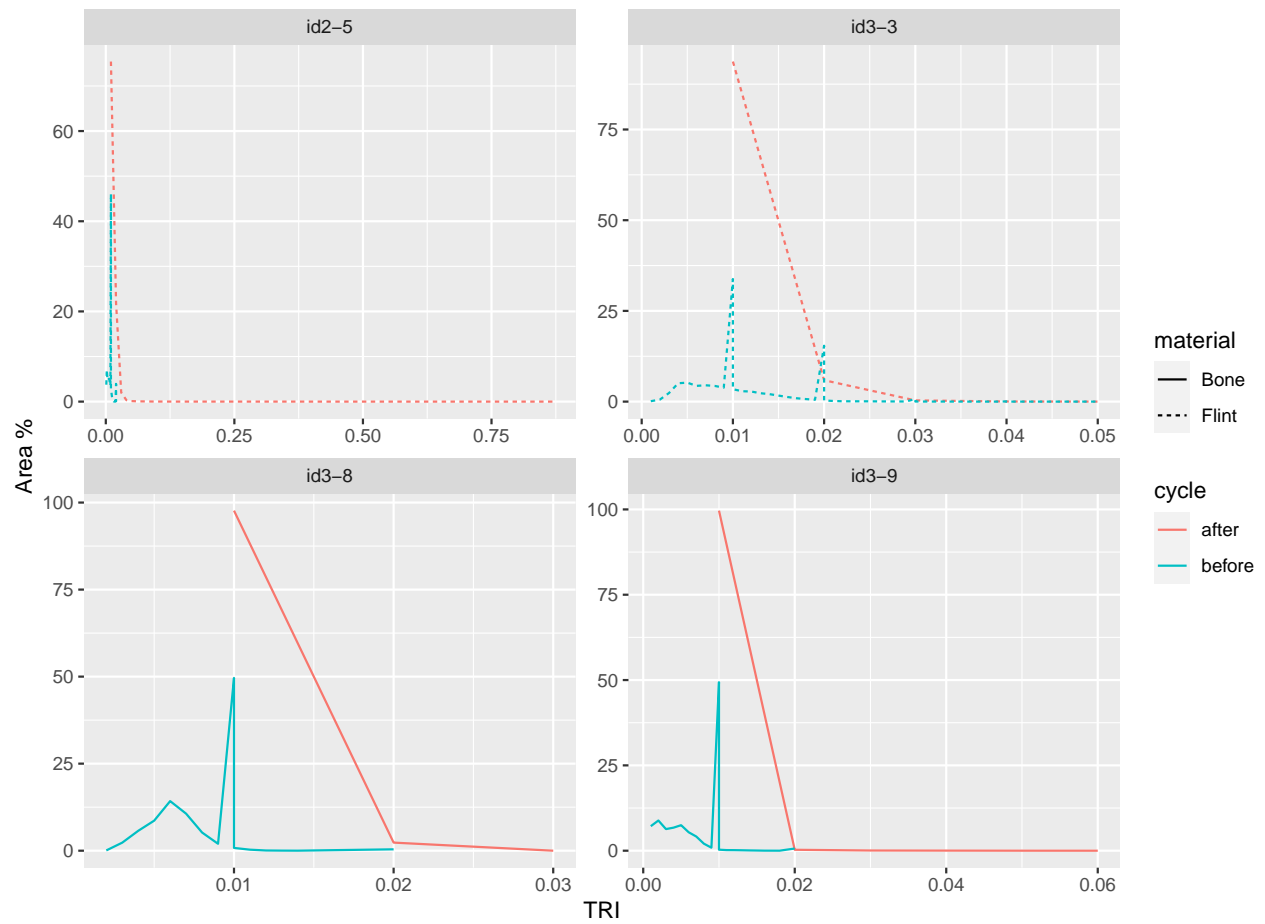
```
## Saving 8.5 x 6.5 in image
```

```
# Area %

areaimpact <- ggplot(impactdf, aes(x = elev_max, y = areaperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("TRI analysis, impact") +
  ylab("Area %") +
  xlab("TRI")

areaimpact
```

### TRI analysis, impact

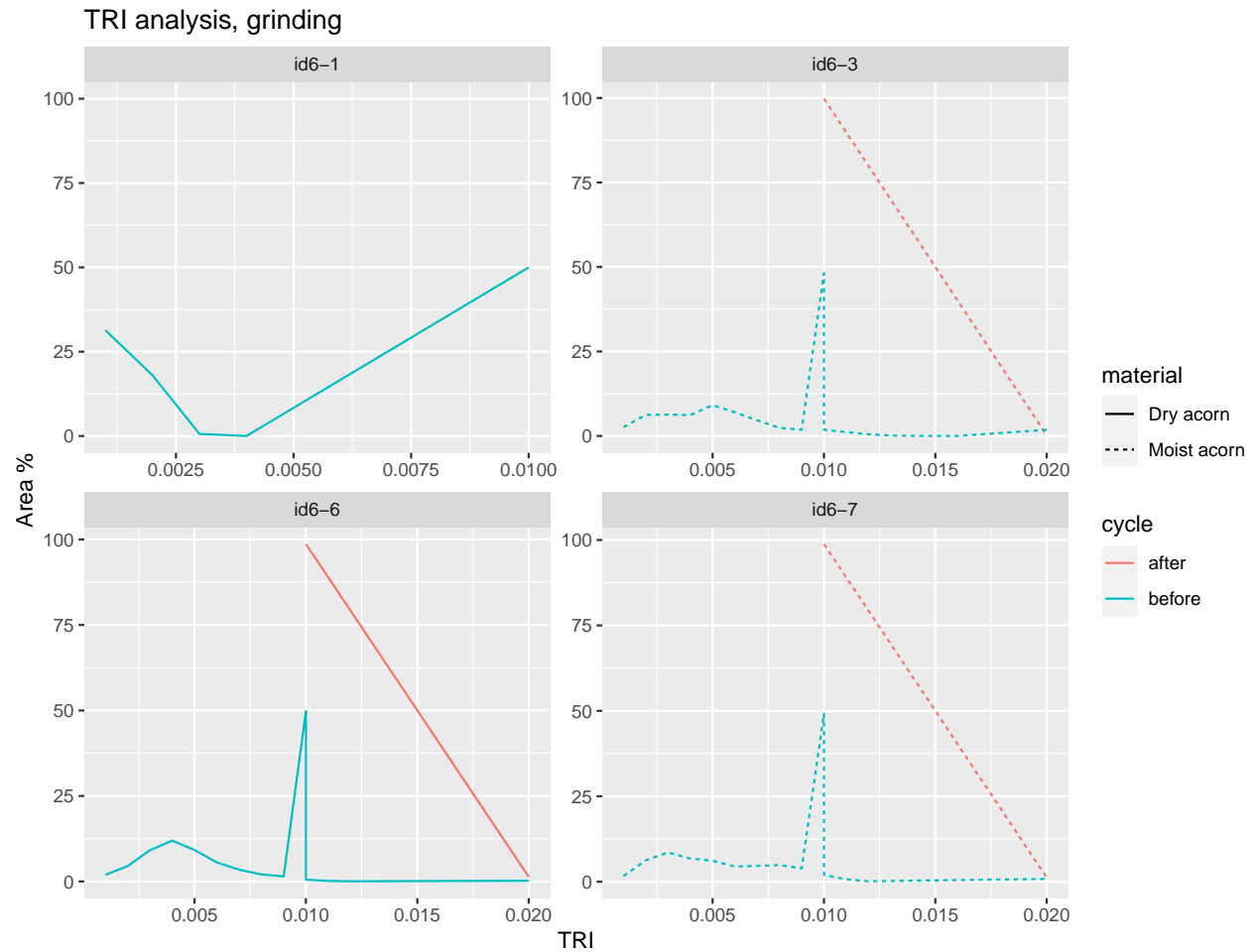


```
ggsave("../plots/triareaimpact.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
areagrinding <- ggplot(grinding, aes(x = elev_max, y = areaperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("TRI analysis, grinding") +
  ylab("Area %") +
  xlab("TRI")
```

```
areagrinding
```



```
ggsave("../plots/triareagrinding.png")
```

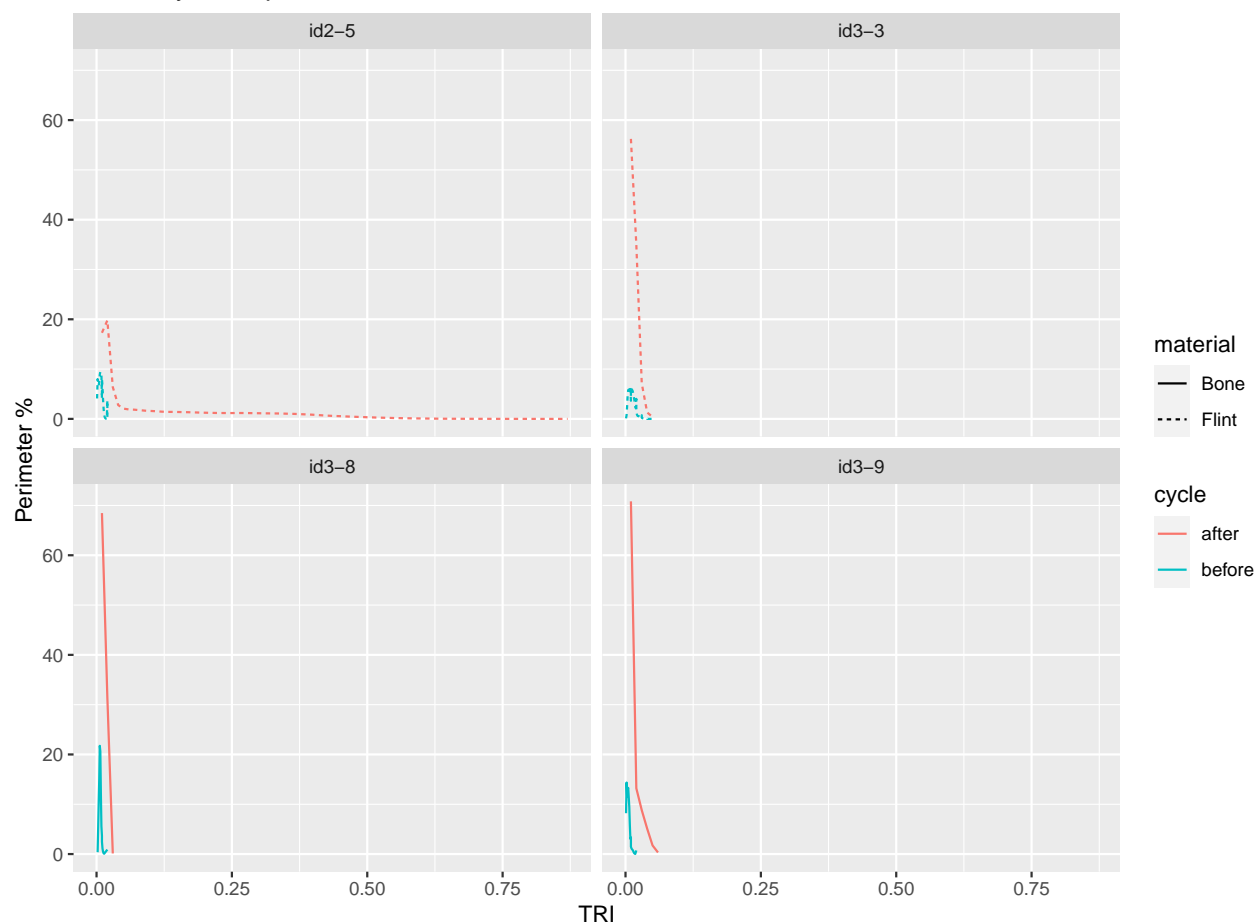
```
## Saving 8.5 x 6.5 in image
```

```
# Perimeter %

perimimpact <- ggplot(impactdf, aes(x = elev_max, y = perimperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample) +
  ggtitle("TRI analysis, impact") +
  ylab("Perimeter %") +
  xlab("TRI")

perimimpact
```

## TRI analysis, impact

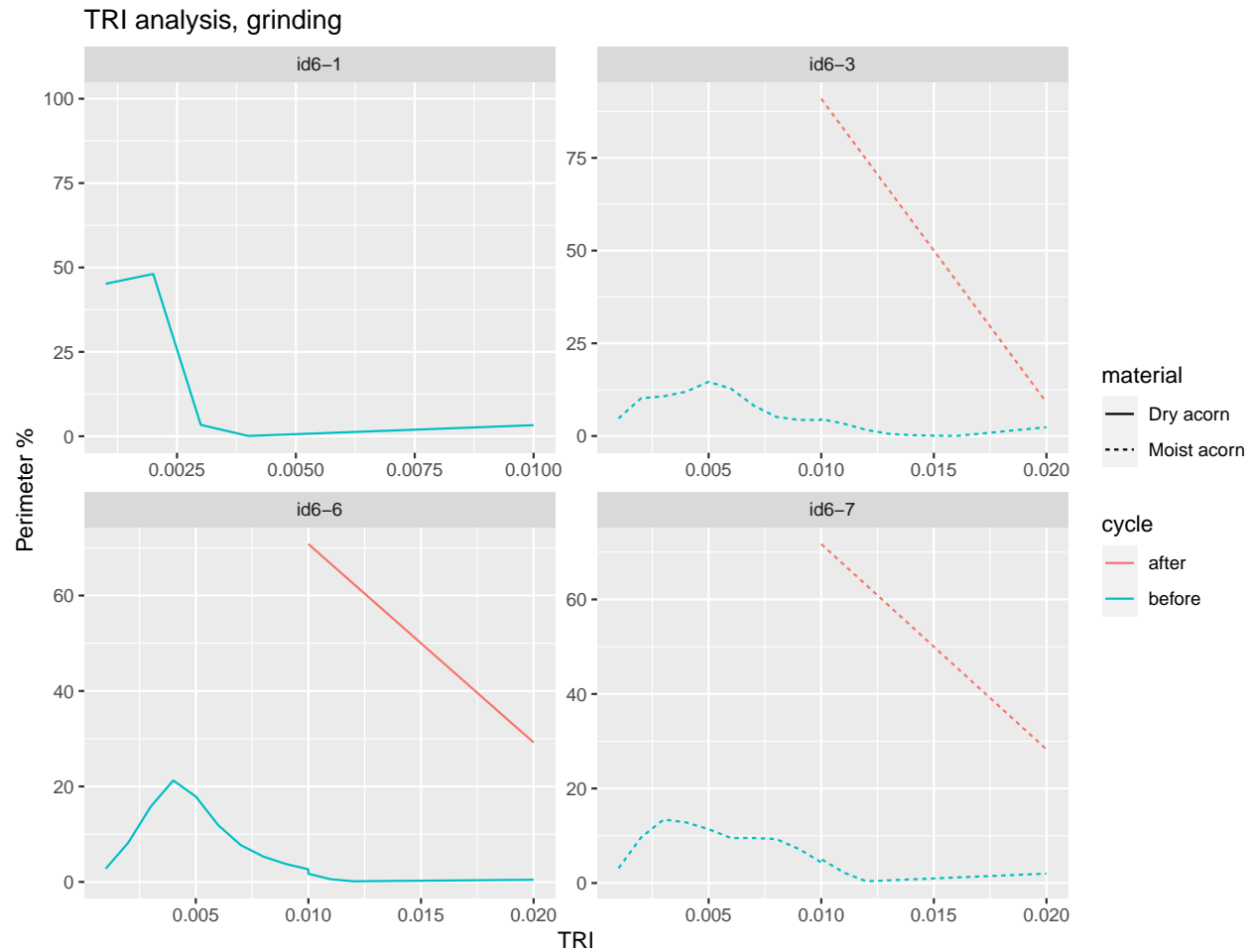


```
ggsave("../plots/triperimimpact.png")
```

```
## Saving 8.5 x 6.5 in image
```

```
perimgrinding <- ggplot(grinding, aes(x = elev_max, y = perimperc, colour = cycle)) +
  geom_line(aes(linetype = material)) +
  facet_wrap(~sample, scale = "free") +
  ggtitle("TRI analysis, grinding") +
  ylab("Perimeter %") +
  xlab("TRI")
```

```
perimgrinding
```



```
ggsave("../plots/triperimgrinding.png")
```

```
## Saving 8.5 x 6.5 in image
```

## Confocal micro surface texture data

### Import and summarize data

```
# compute 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 <- 21:length(confocaldata)

confostatsexp <- summaryBy(~sample + motion + workedmaterial, data=confocaldata[c("sample", "motion", "

## Warning in min(y): no non-missing arguments to min; returning Inf

## Warning in max(y): no non-missing arguments to max; returning -Inf

## Warning in min(y): no non-missing arguments to min; returning Inf

## Warning in max(y): no non-missing arguments to max; returning -Inf

## Warning in min(y): no non-missing arguments to min; returning Inf

## Warning in max(y): no non-missing arguments to max; returning -Inf

## Warning in min(y): no non-missing arguments to min; returning Inf

## Warning in max(y): no non-missing arguments to max; returning -Inf

write_csv(confostatsexp, "../derived_data/confostats.csv")

```

## Plot all paramaters

```

# Loop for plotting all surface texture parameters

for (i in num.var) cat("[",i,"] ", names(confocaldata)[i], "\n", sep = "")

## [21] Sq
## [22] Ssk
## [23] Sku
## [24] Sp
## [25] Sv
## [26] Sz
## [27] Sa
## [28] Smr
## [29] Smc
## [30] Sxp
## [31] Sal
## [32] Str
## [33] Std
## [34] Sdq
## [35] Sdr
## [36] Vm
## [37] Vv

```



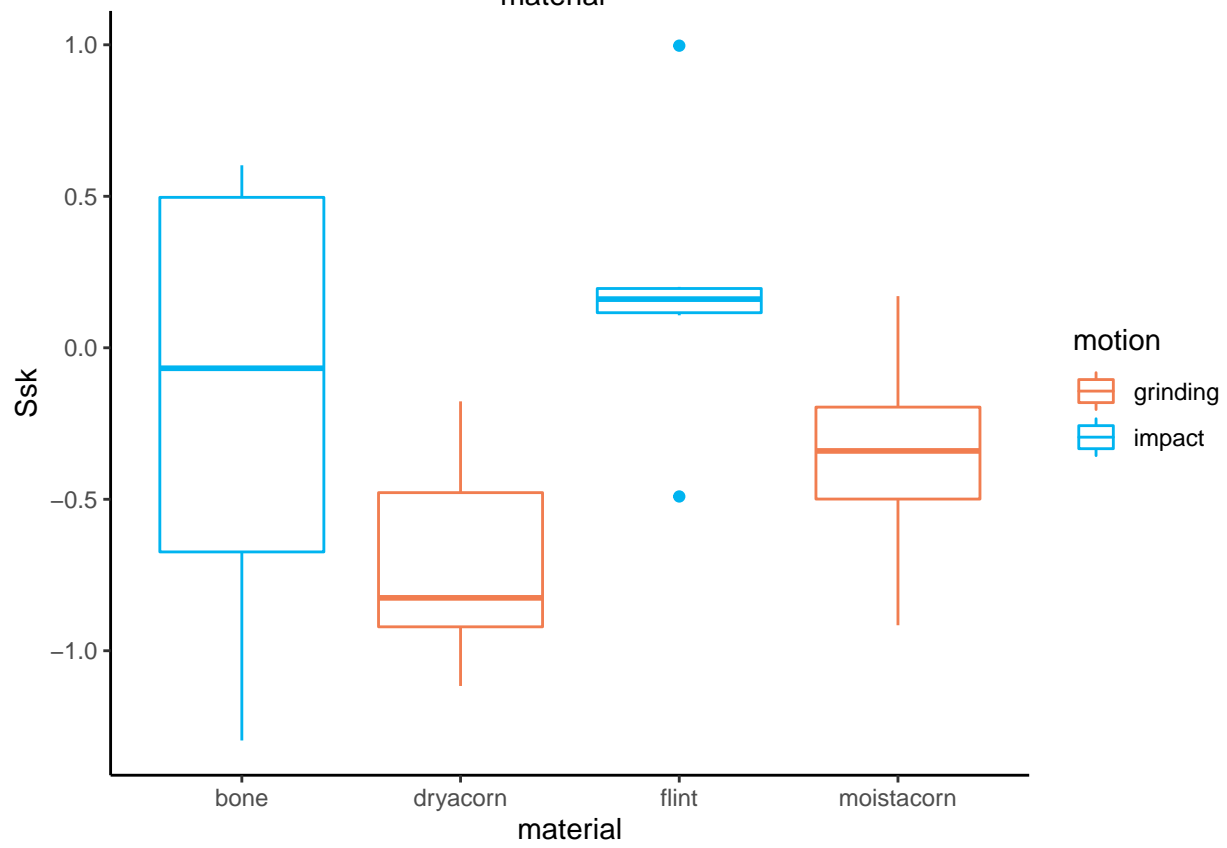
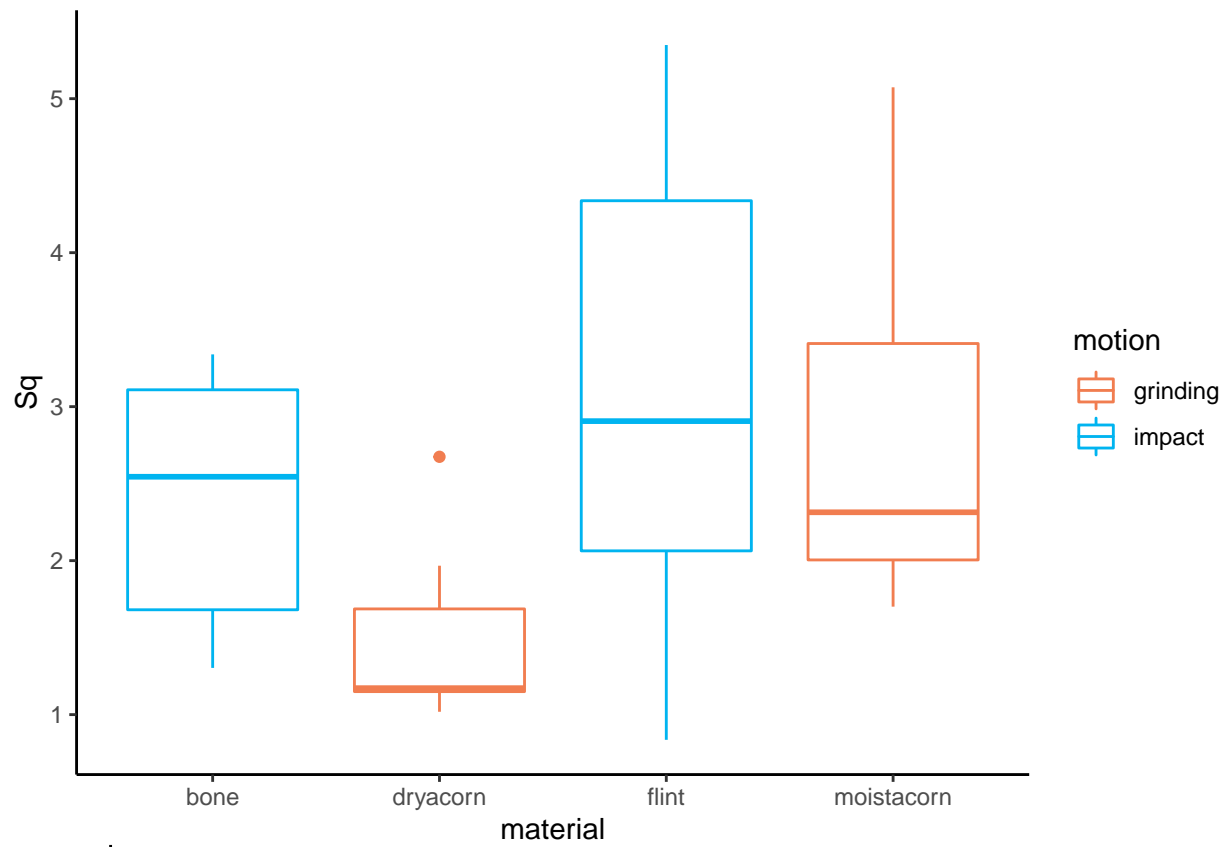
```

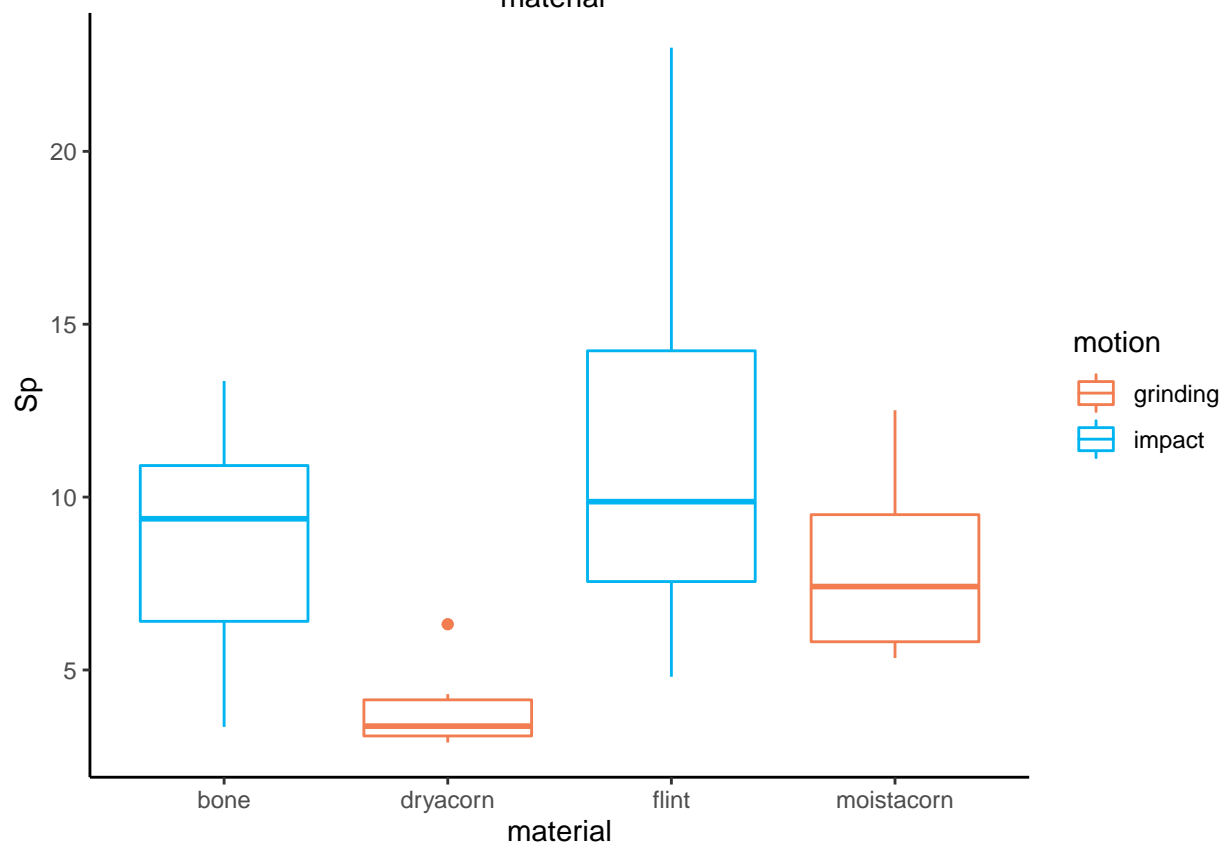
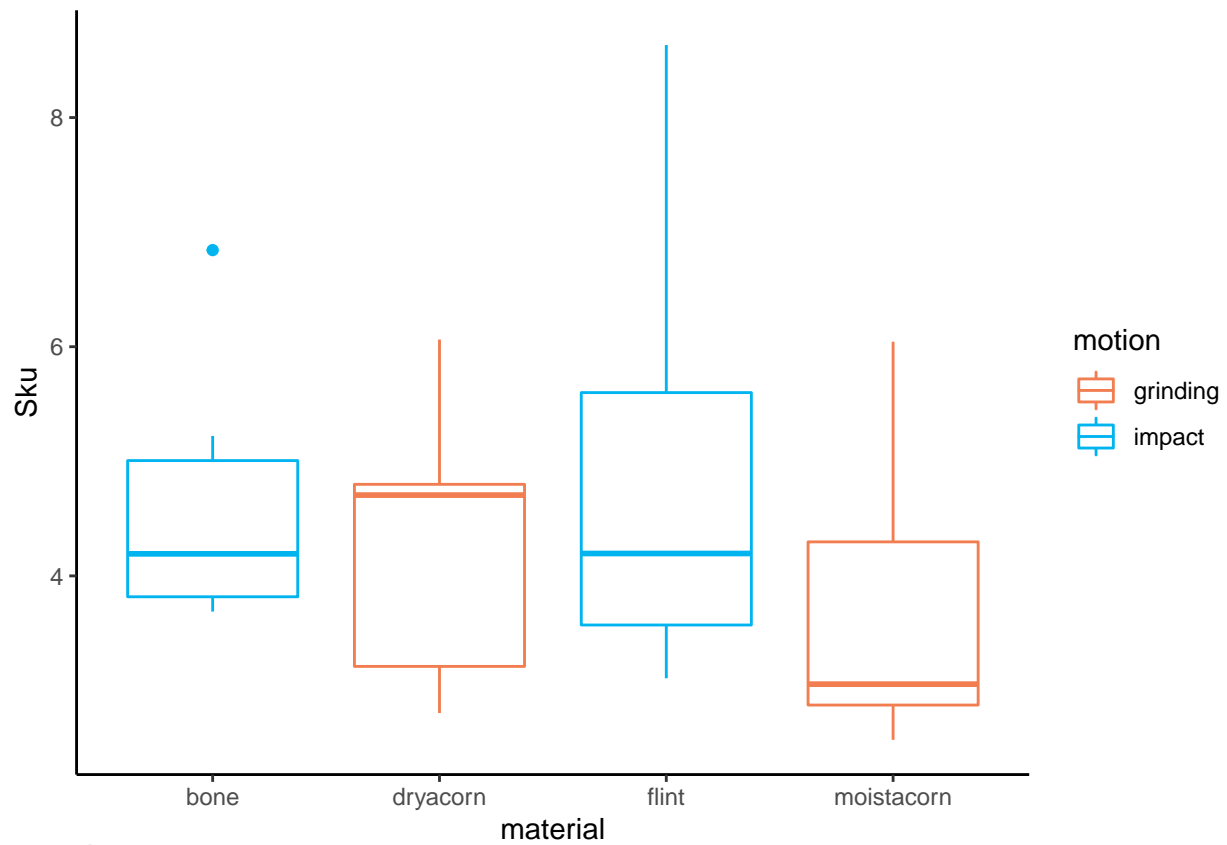
## [38] Vmp
## [39] Vmc
## [40] Vvc
## [41] Vvv
## [42] Maximum.depth.of.furrows
## [43] Mean.depth.of.furrows
## [44] Mean.density.of.furrows
## [45] First.direction
## [46] Second.direction
## [47] Third.direction
## [48] Isotropy
## [49] Lengthscale.anisotropy.Sfrax.epLsar
## [50] Lengthscale.anisotropy.NewEplsar
## [51] Fractal.complexity.Asfc
## [52] Scale.of.max.complexity.Smfc
## [53] HAsfc9
## [54] HAsfc81

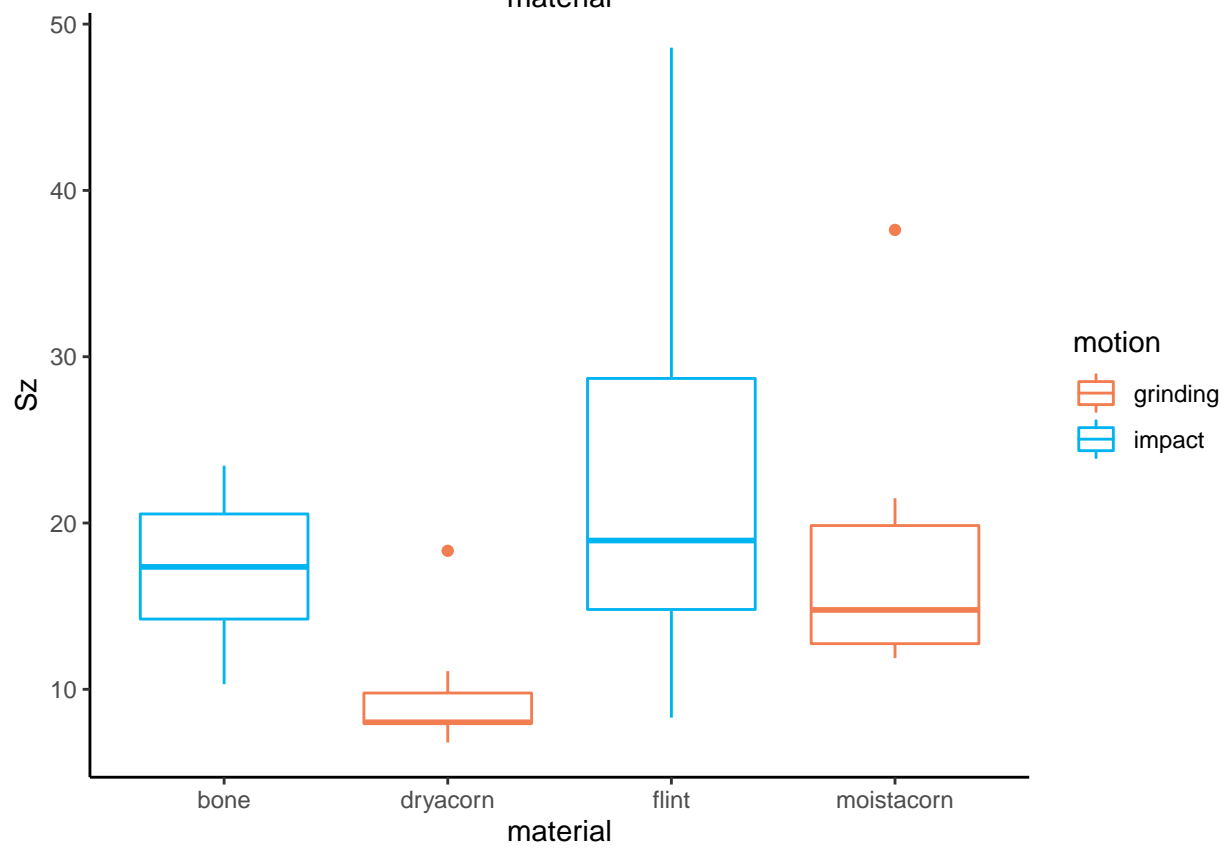
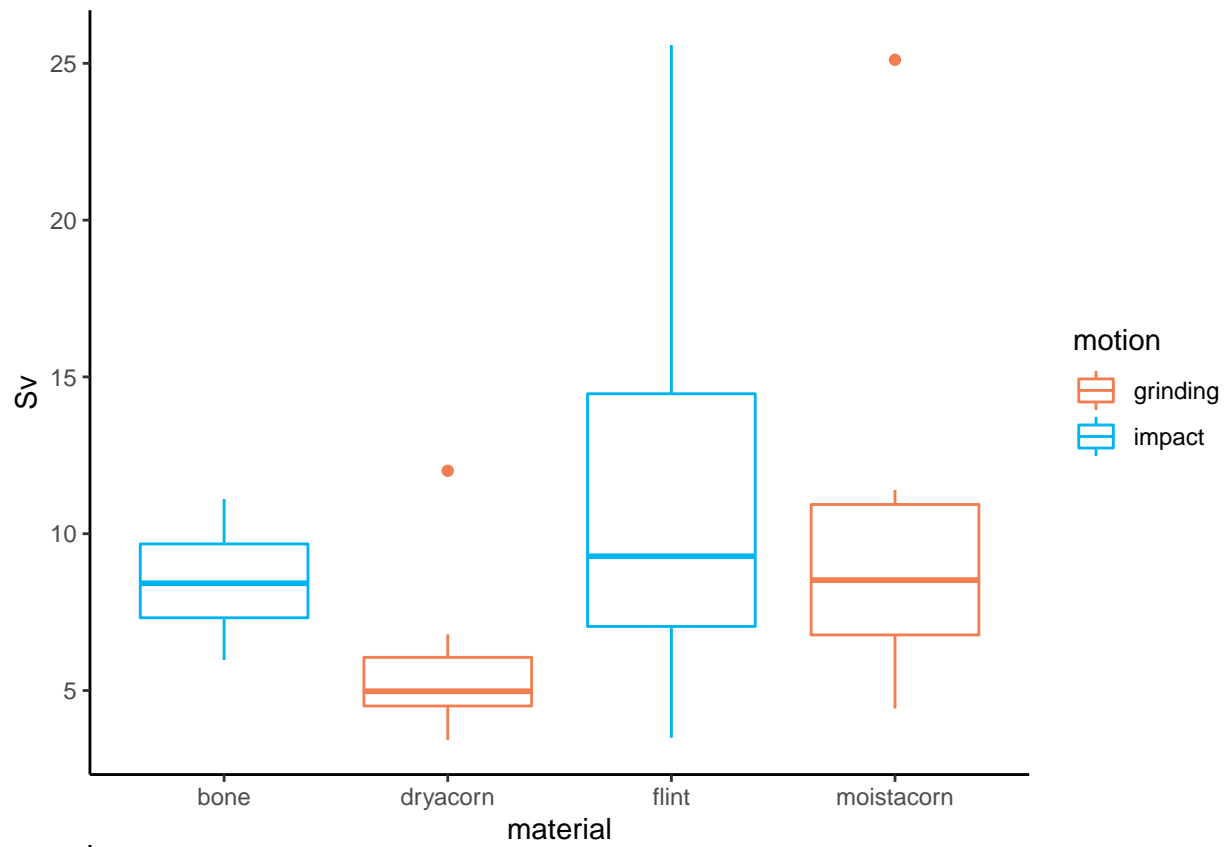
for (i in num.var) {
  p <- ggplot(data = confocaldata, aes_string(x = "workedmaterial", y = names(confocaldata)[i],
                                              colour = "motion")) +
    geom_boxplot() +
    # geom_line(aes(group = motion)) +
    theme_classic() +
    labs(colour = "motion") +
    # facet_wrap(~ sample) +
    labs(x = "material", y = gsub("\\.", " ", names(confocaldata)[i])) +
    scale_colour_hue(h = c(25,225), limits = levels(confocaldata[["motion"]]))
  print(p)

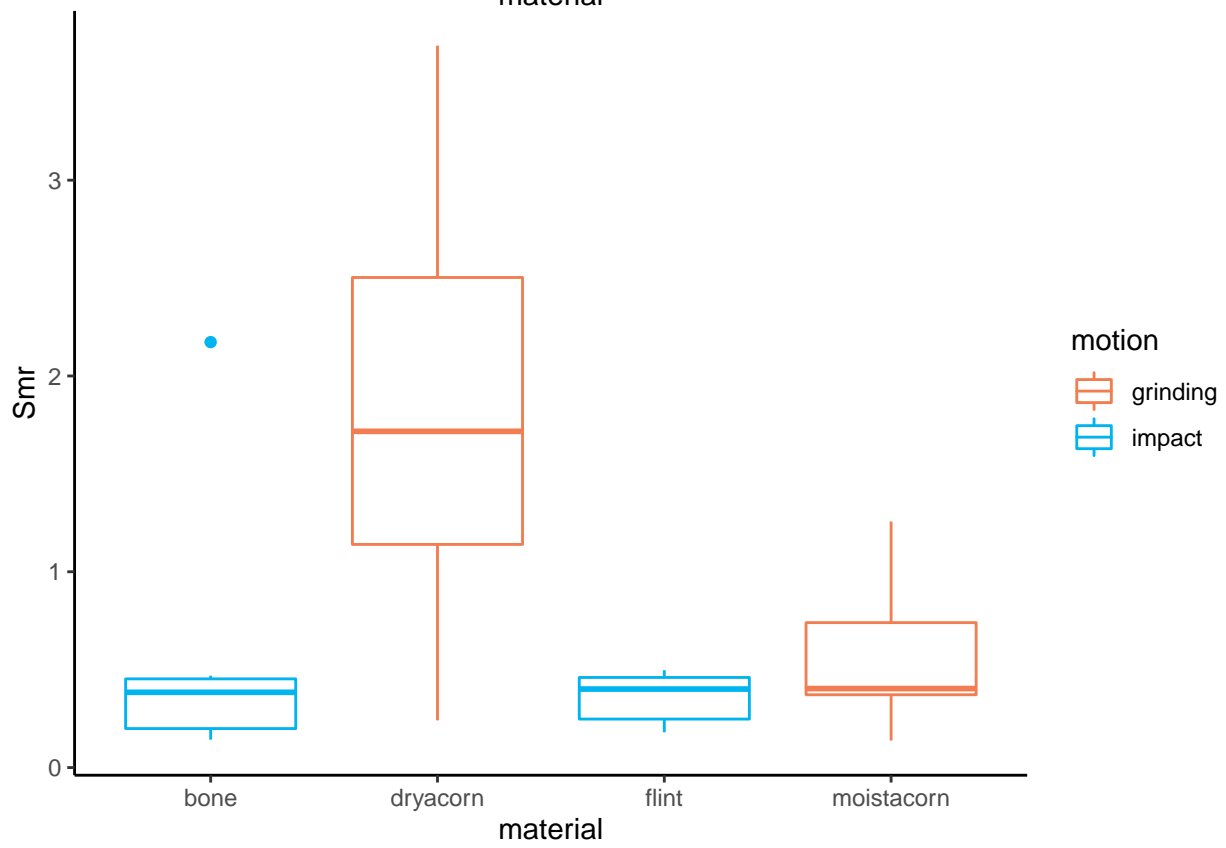
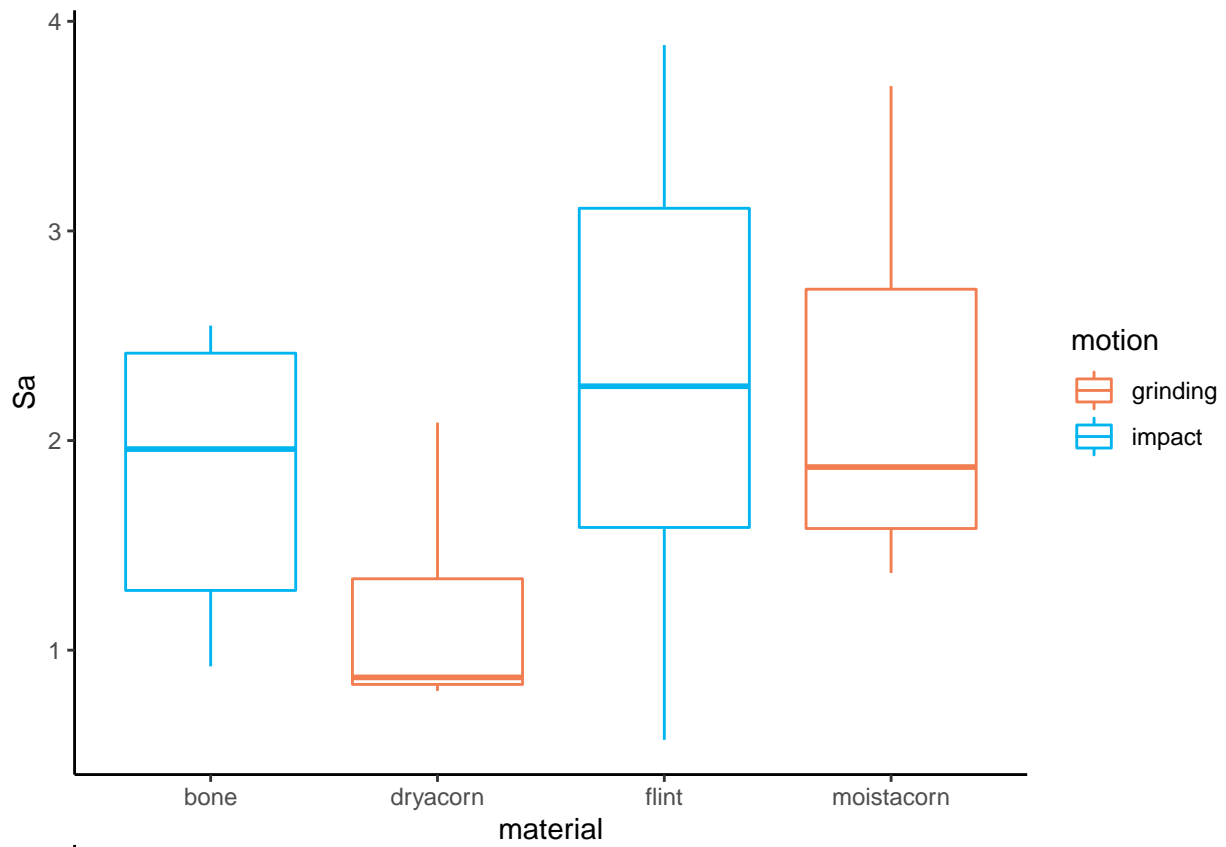
  # saves the plots
  file_out <- paste0(file_path_sans_ext(info_in[["file"]]), "_plot_",
                    names(confocaldata)[i], ".pdf")
  ggsave(filename = file_out, plot = p, path = "../plots", device = "pdf", width = 26,
          height = 21, units = "cm" )
}

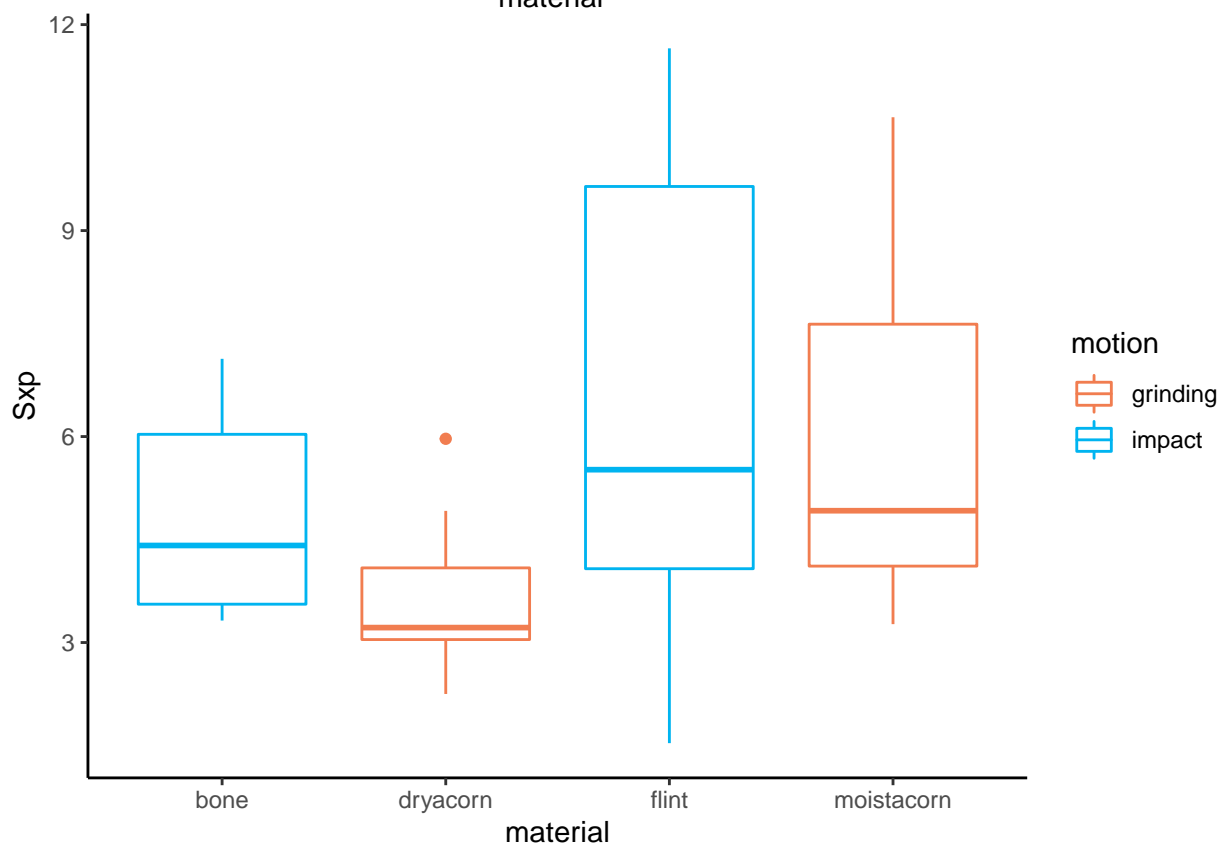
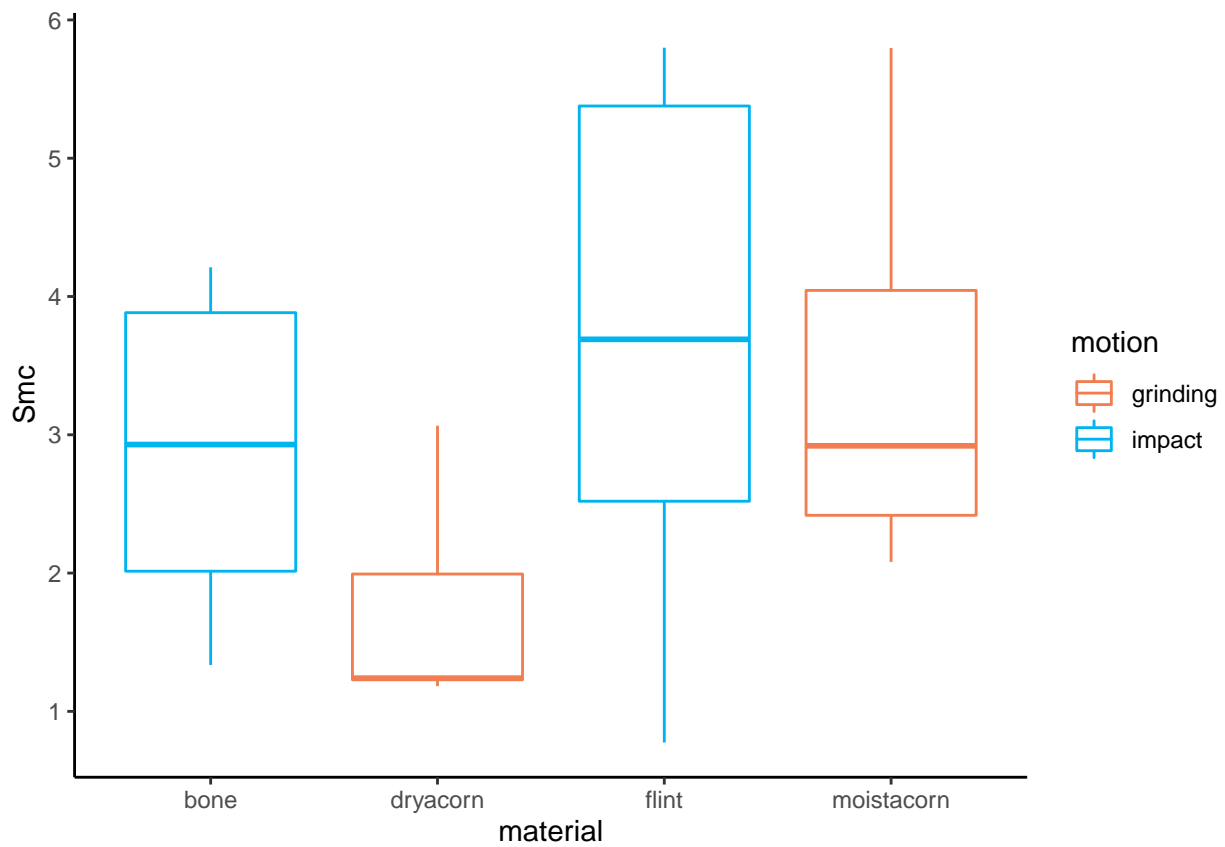
```

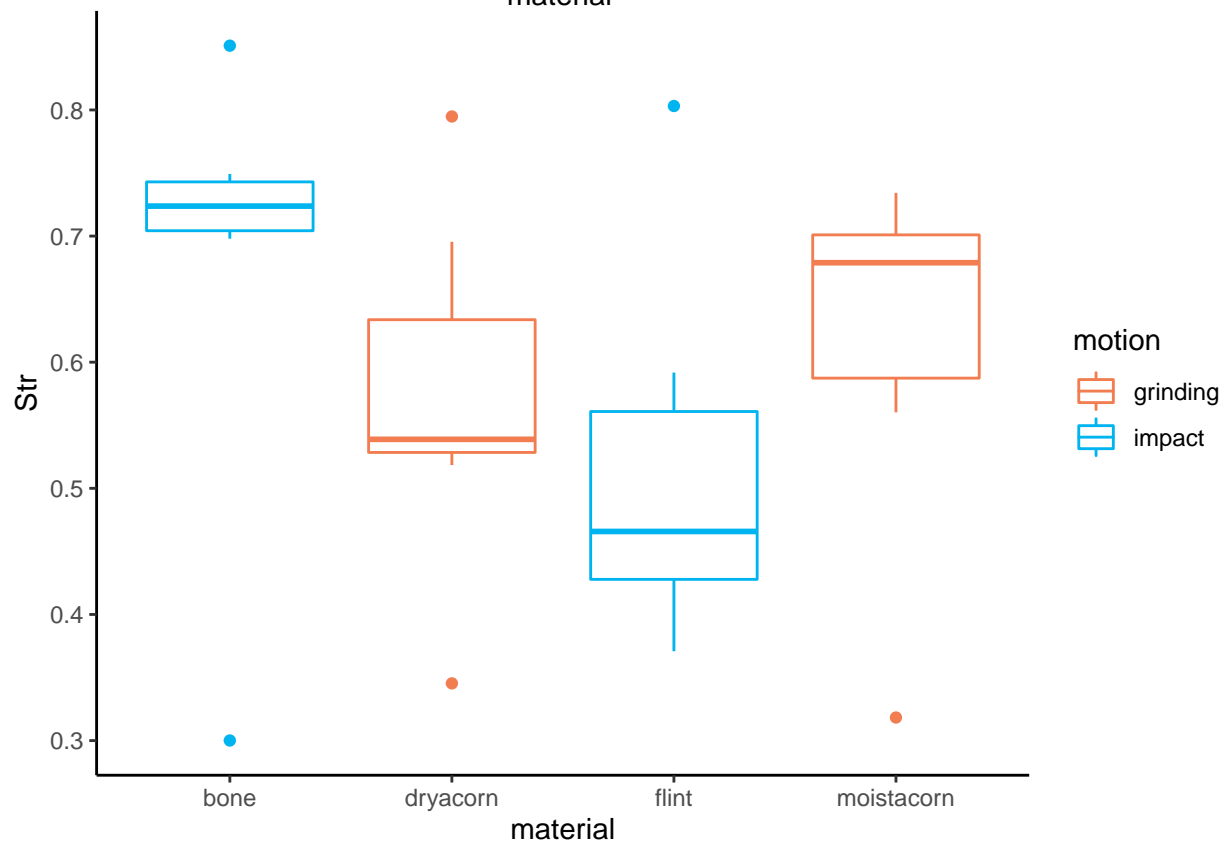
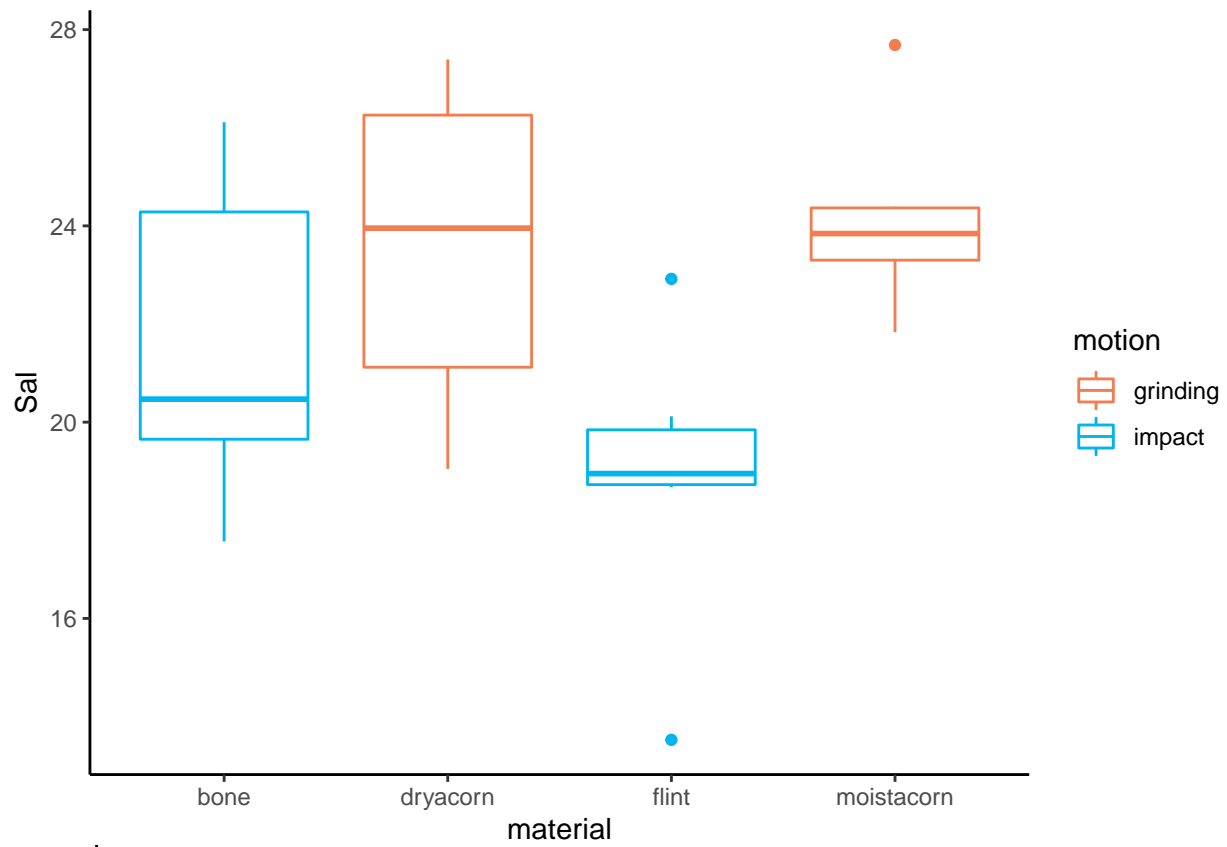


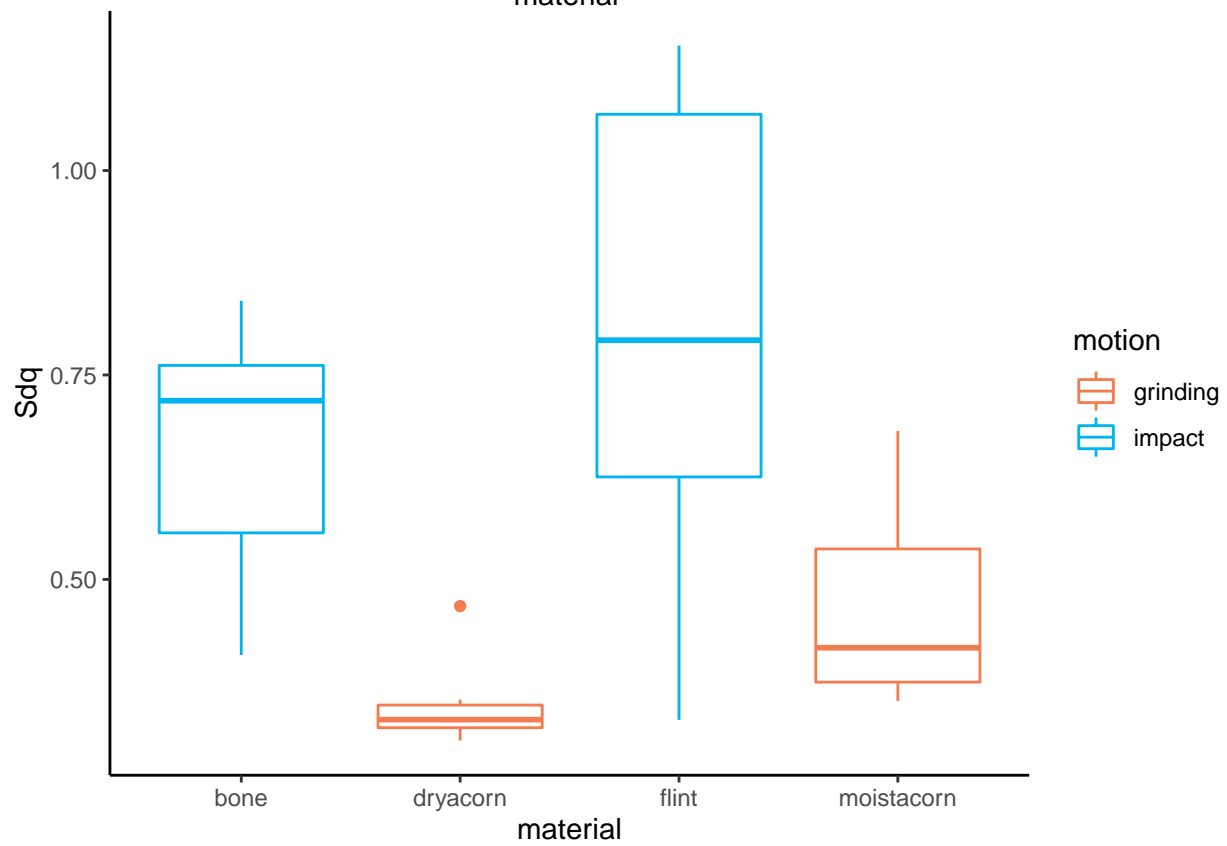
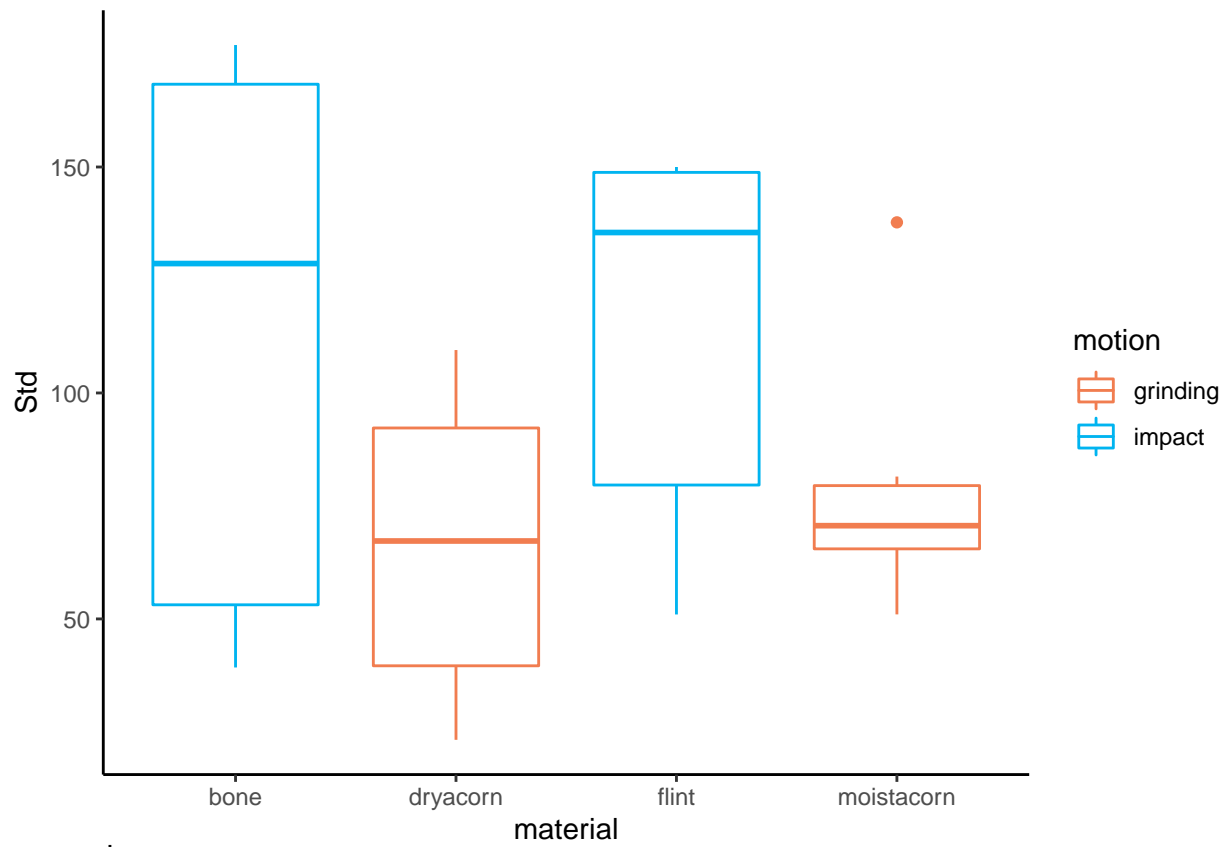




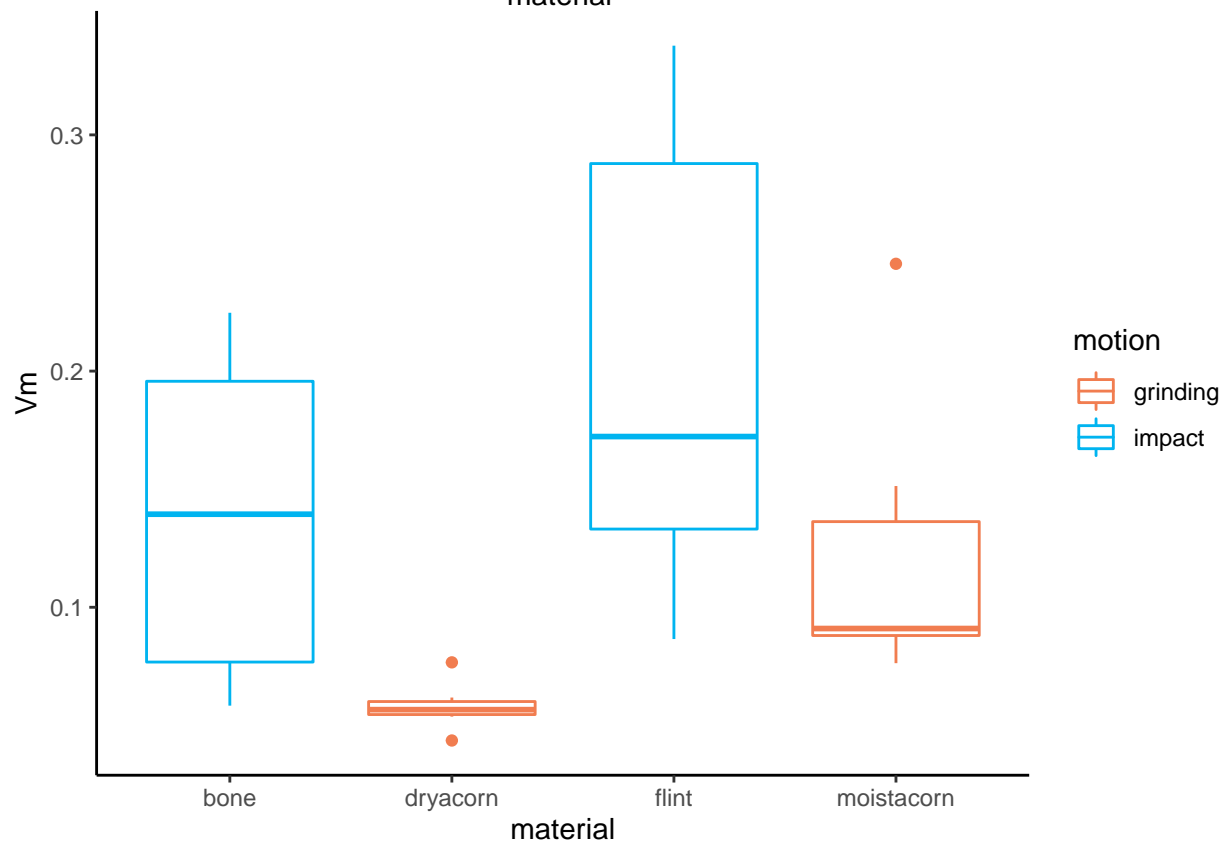
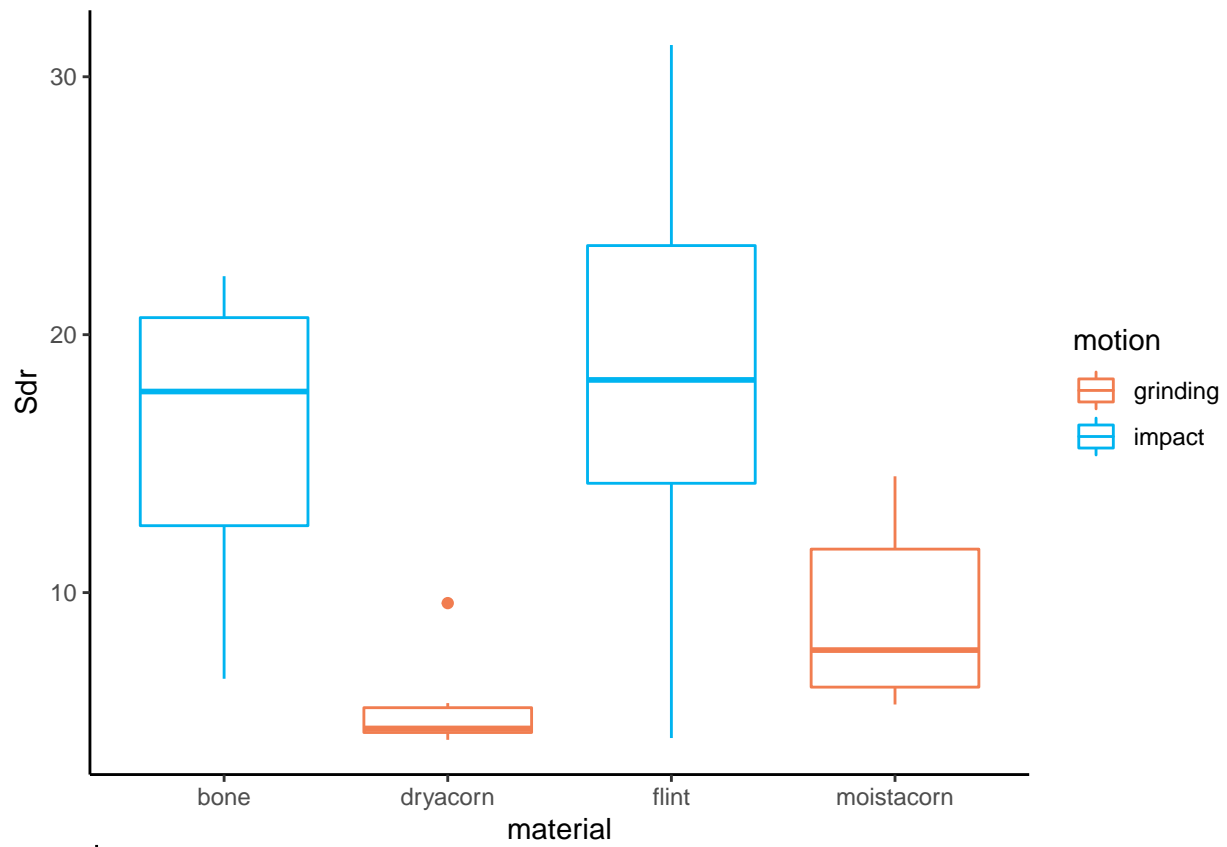


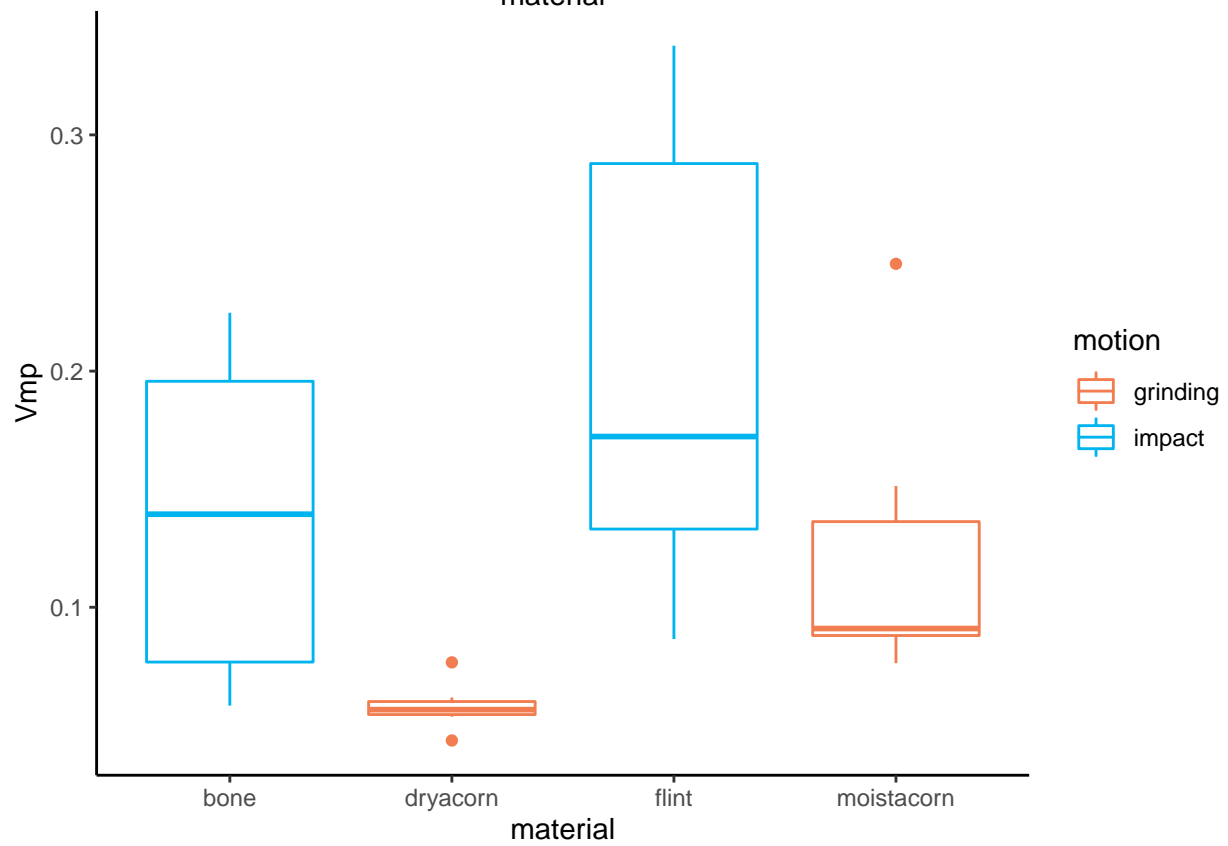
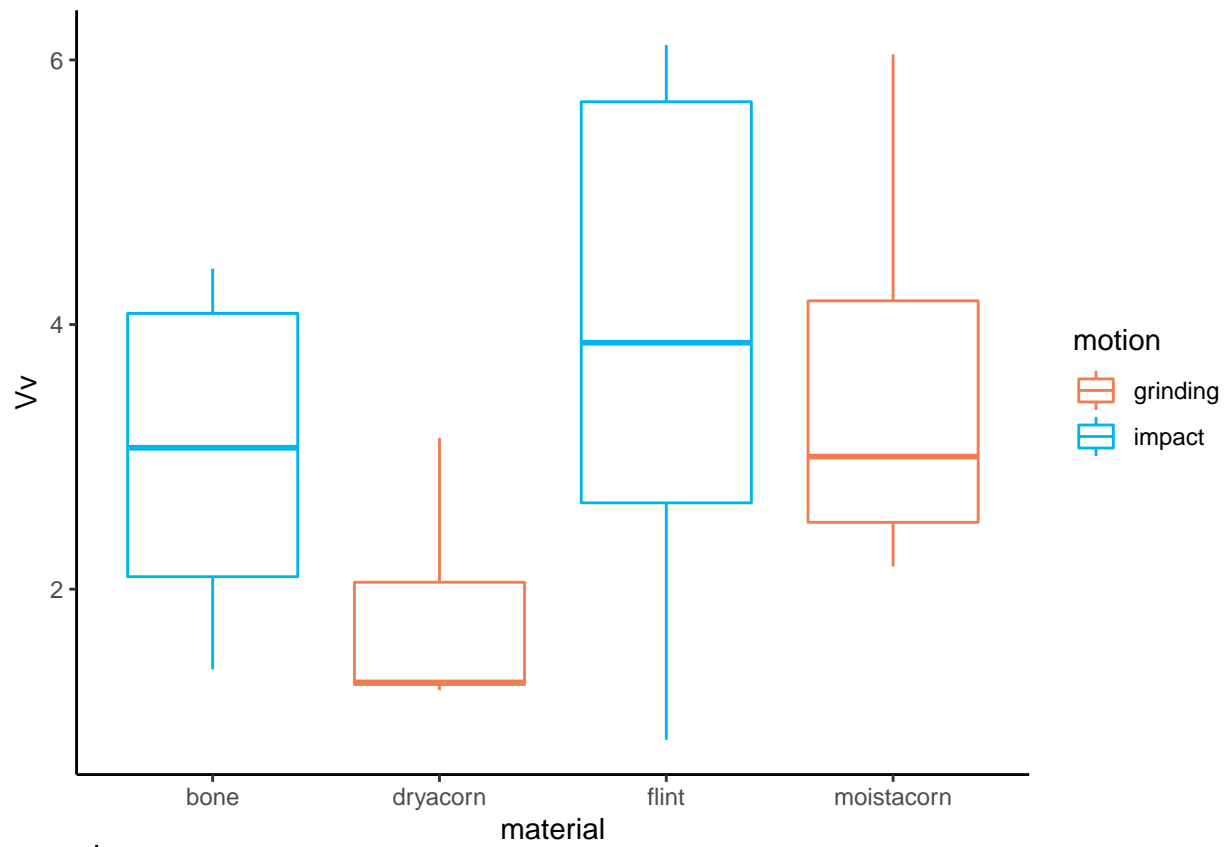


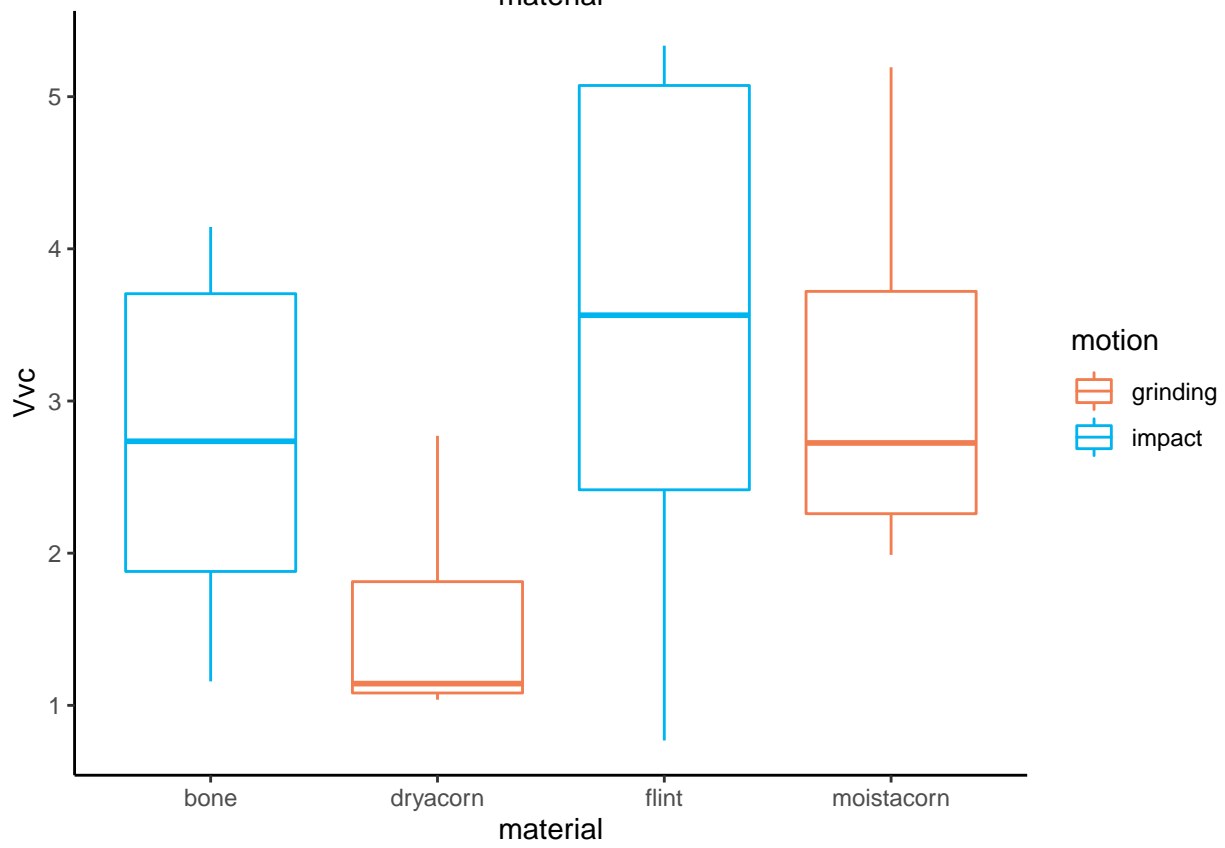
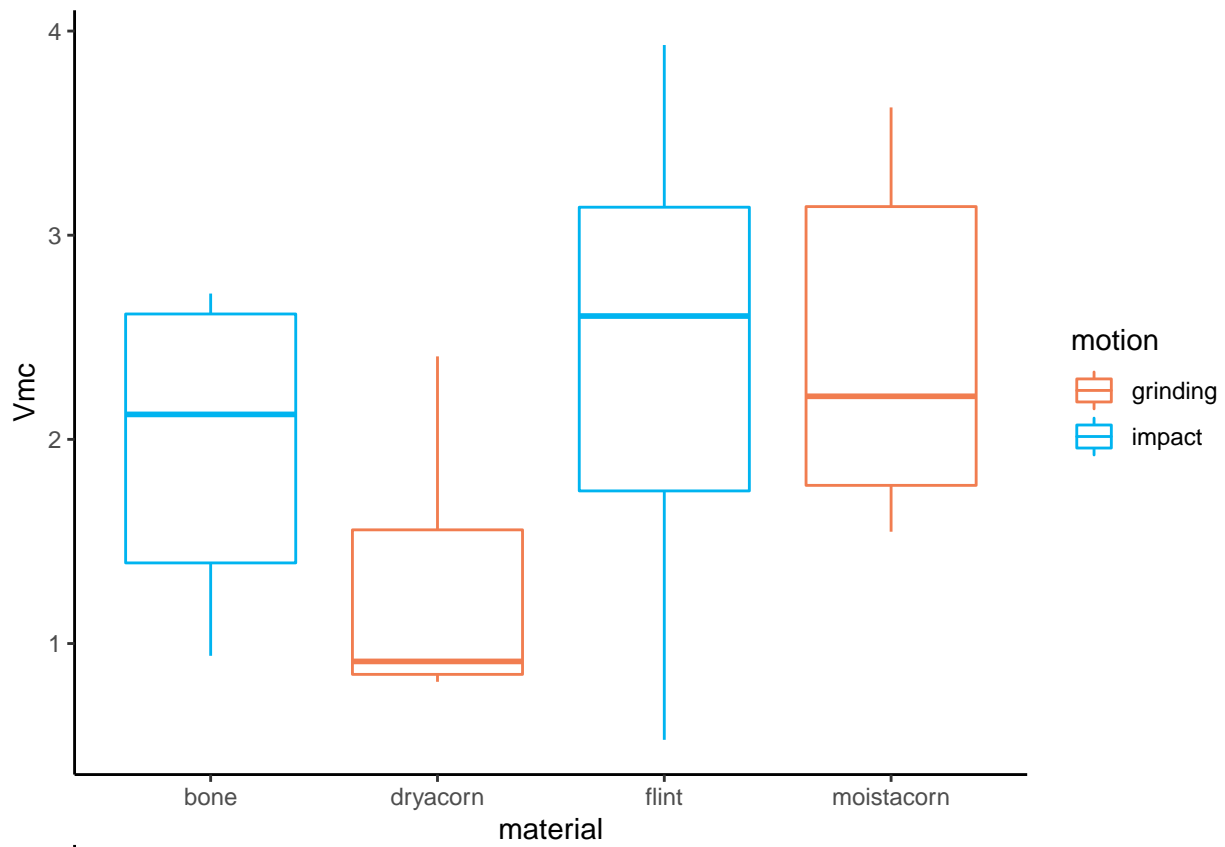


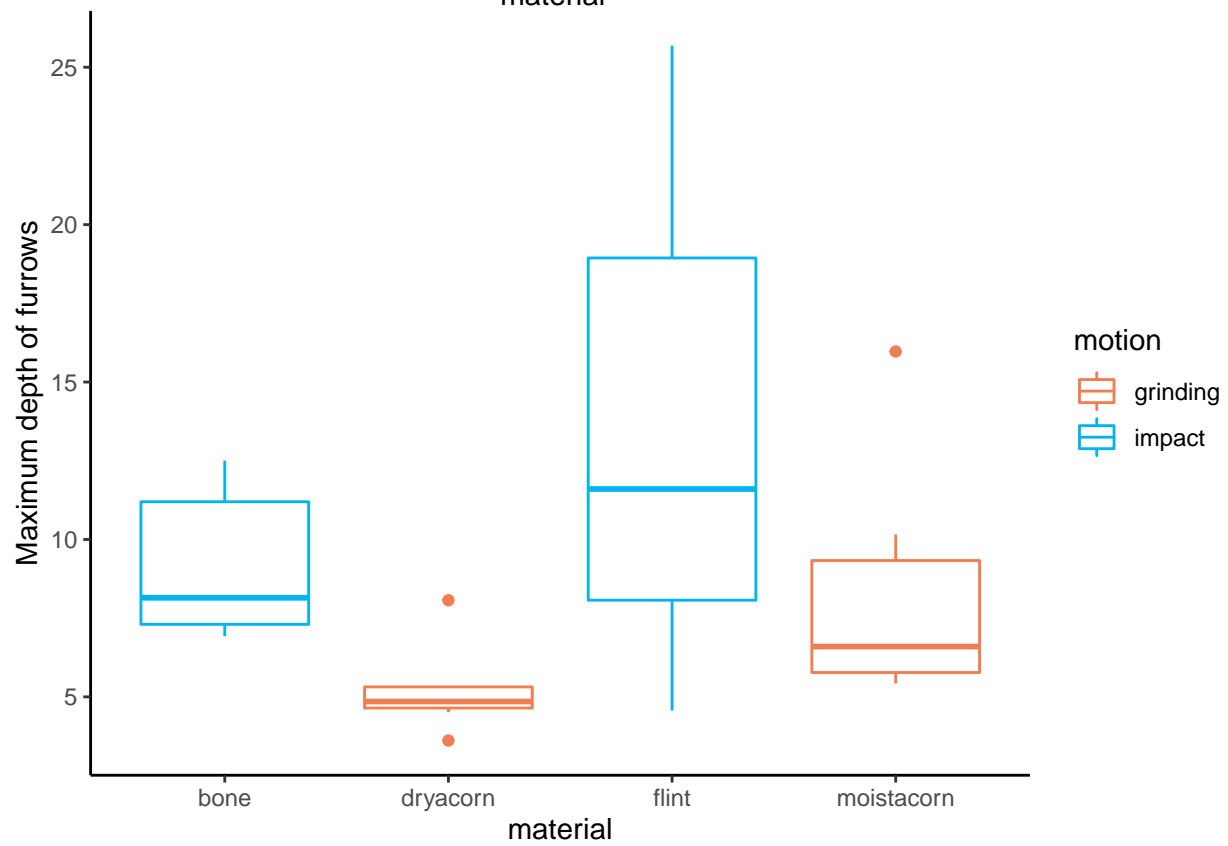
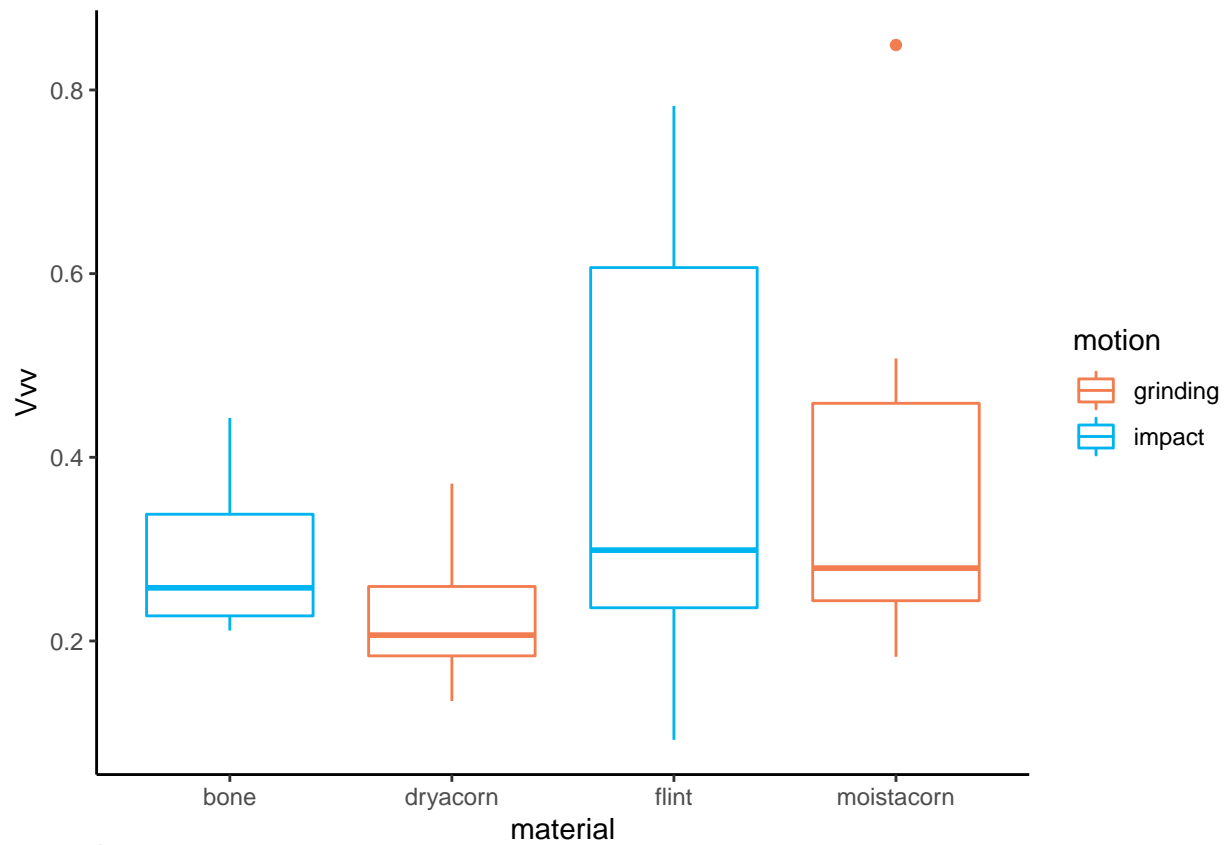


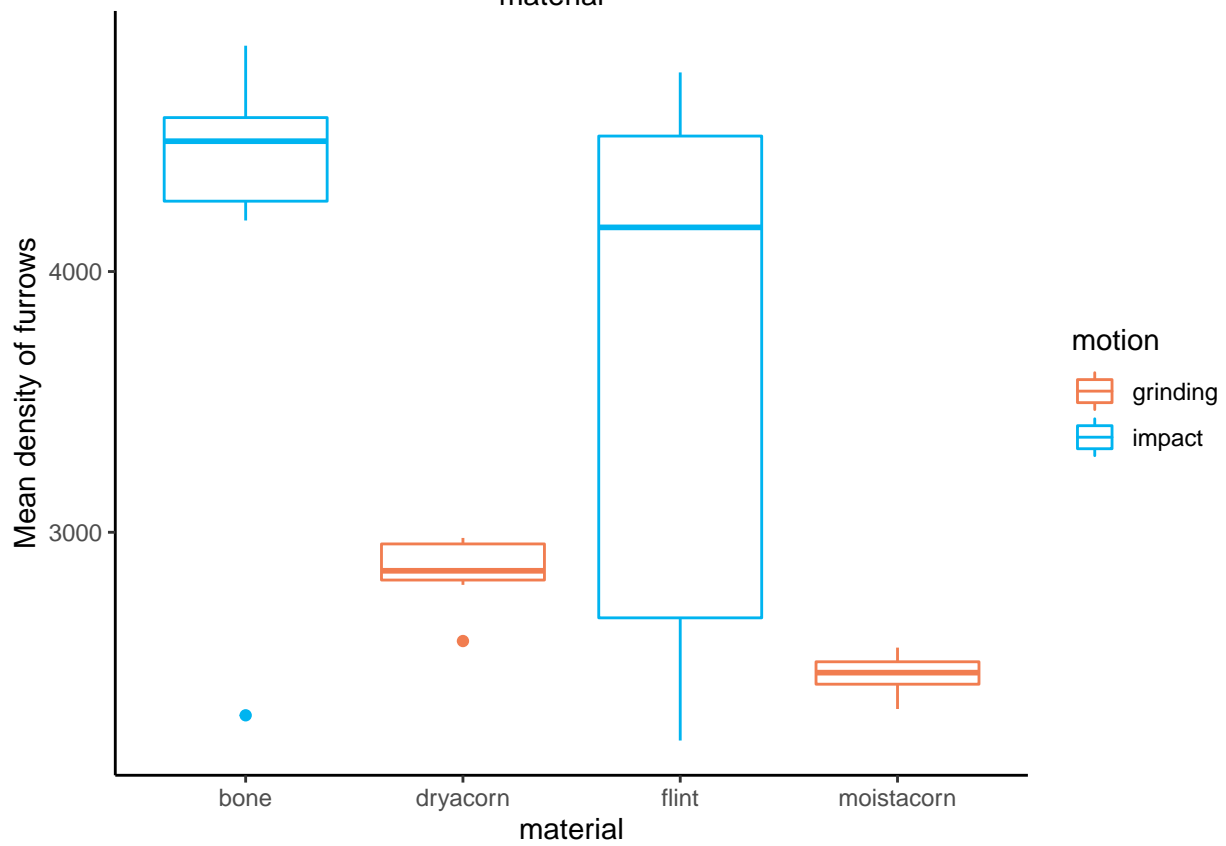
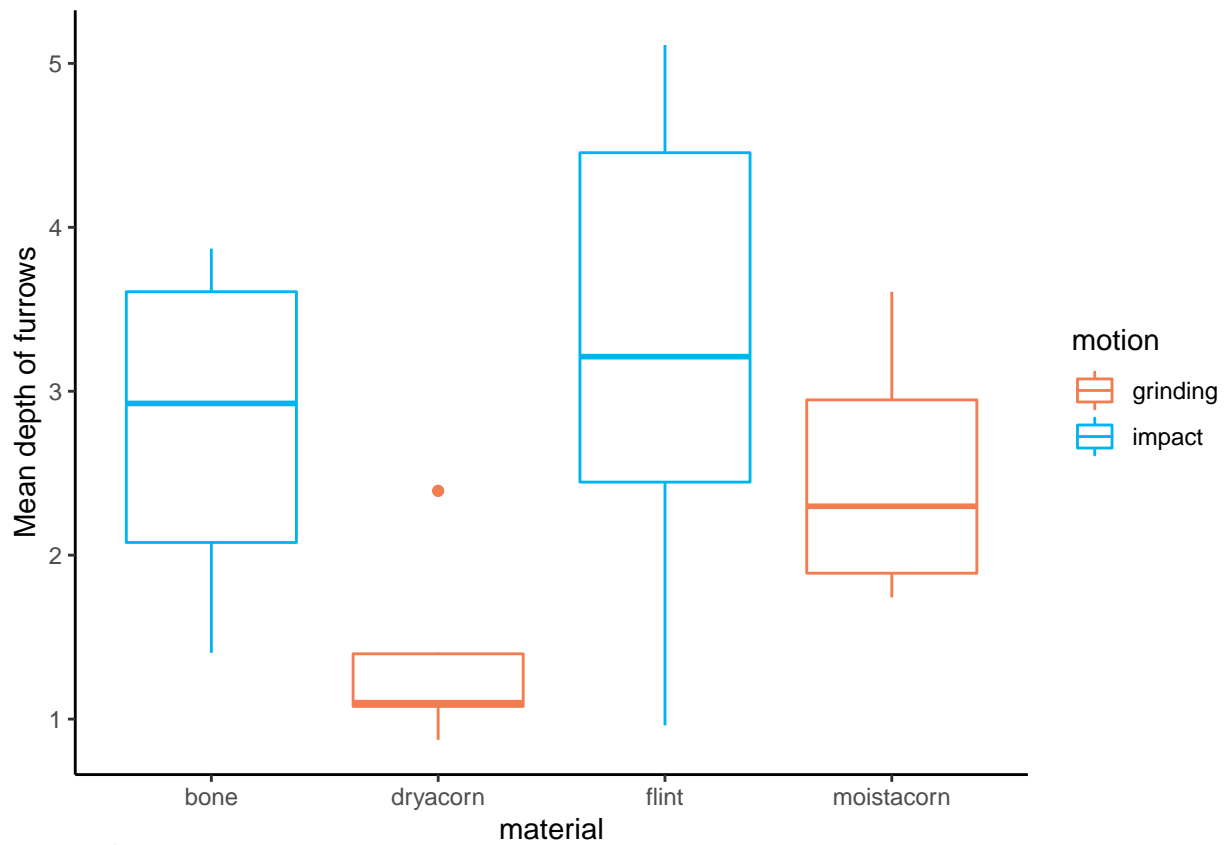


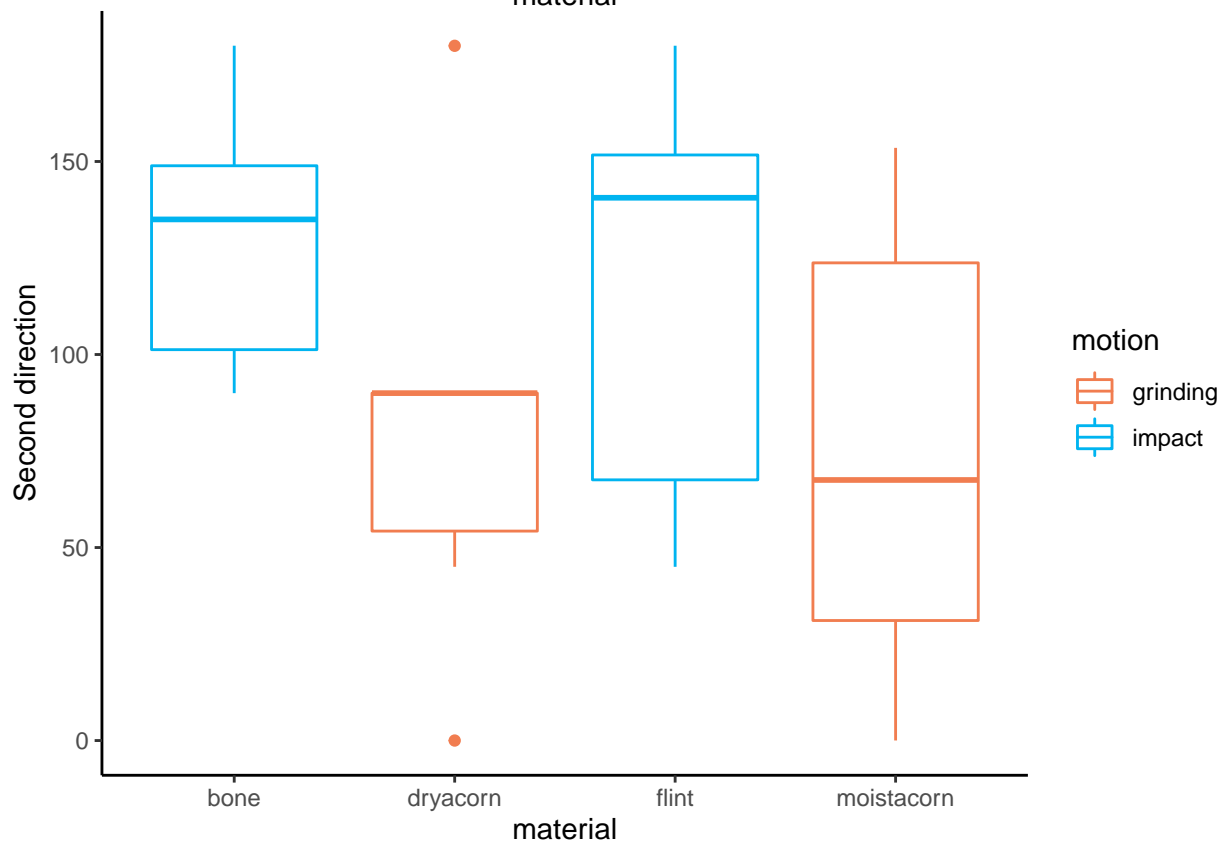
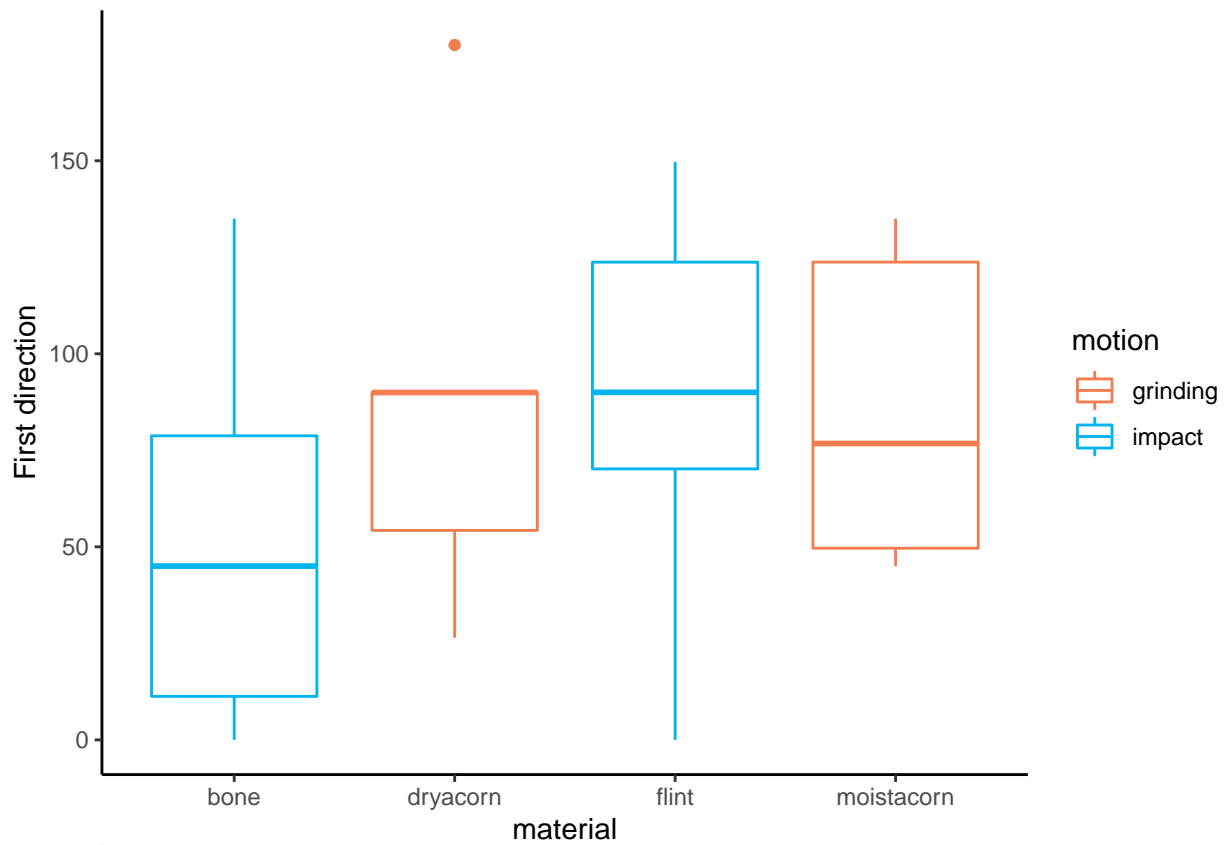


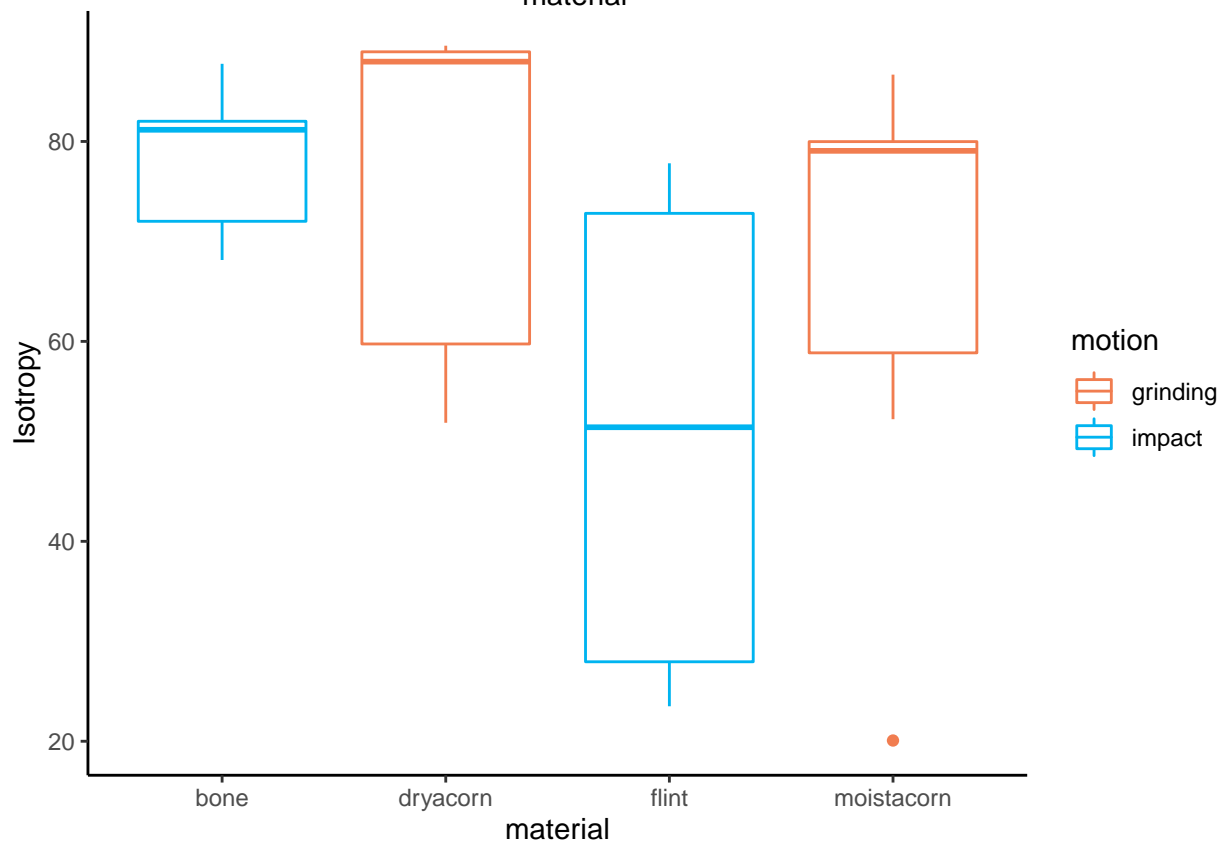
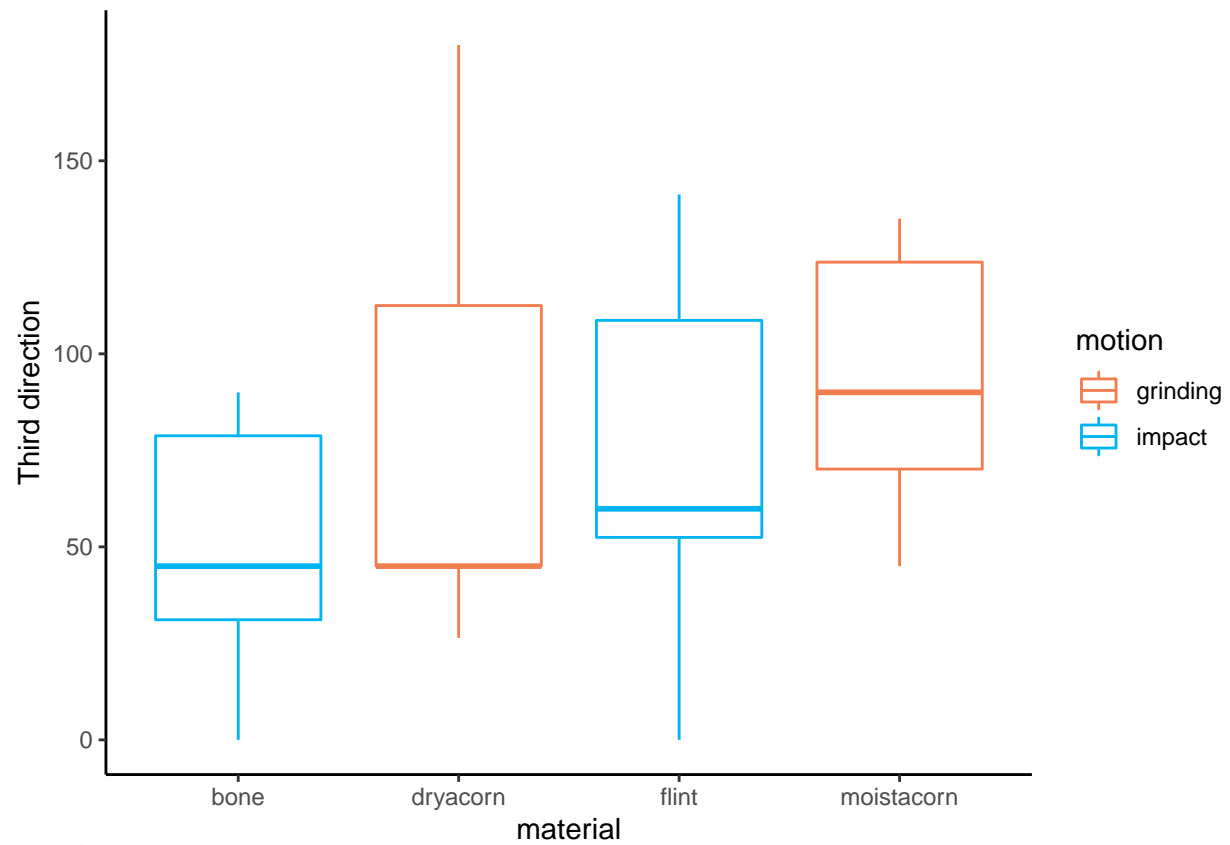






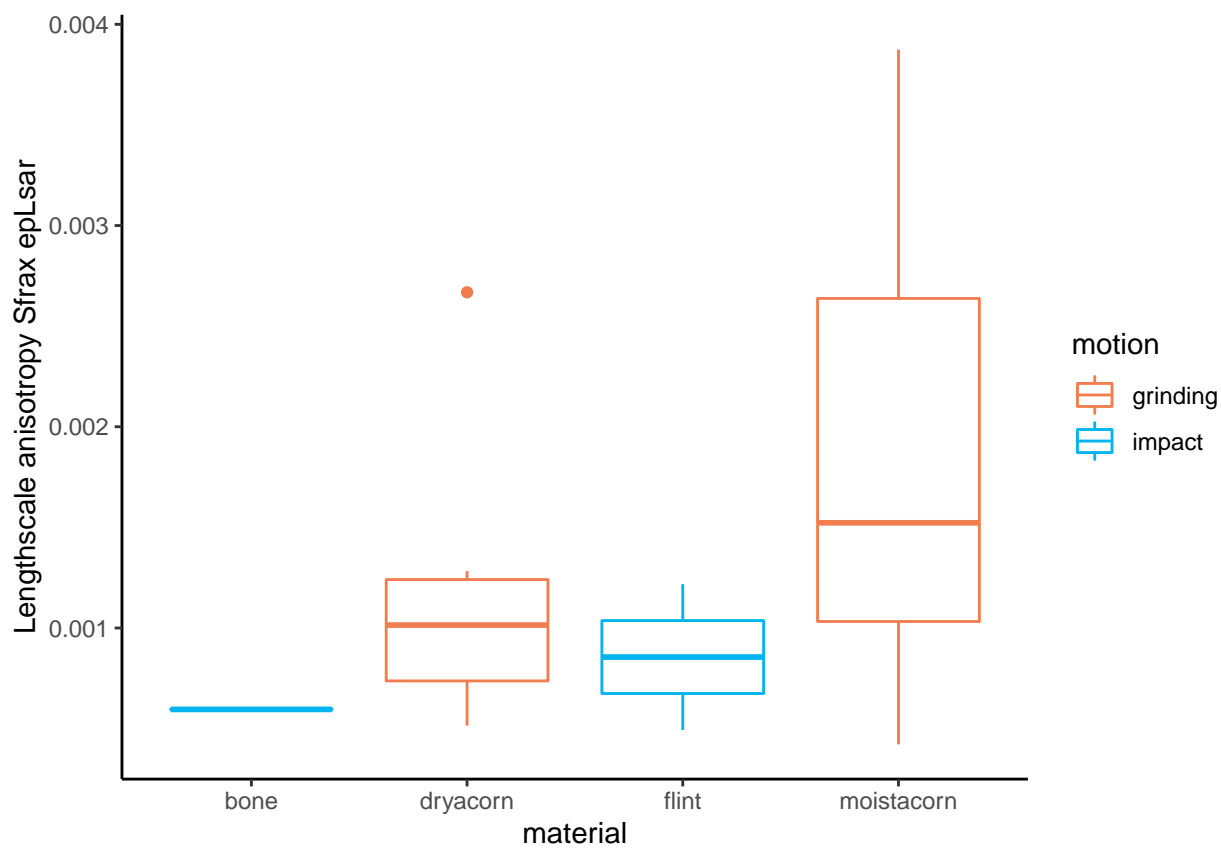






## Warning: Removed 9 rows containing non-finite values (stat\_boxplot).

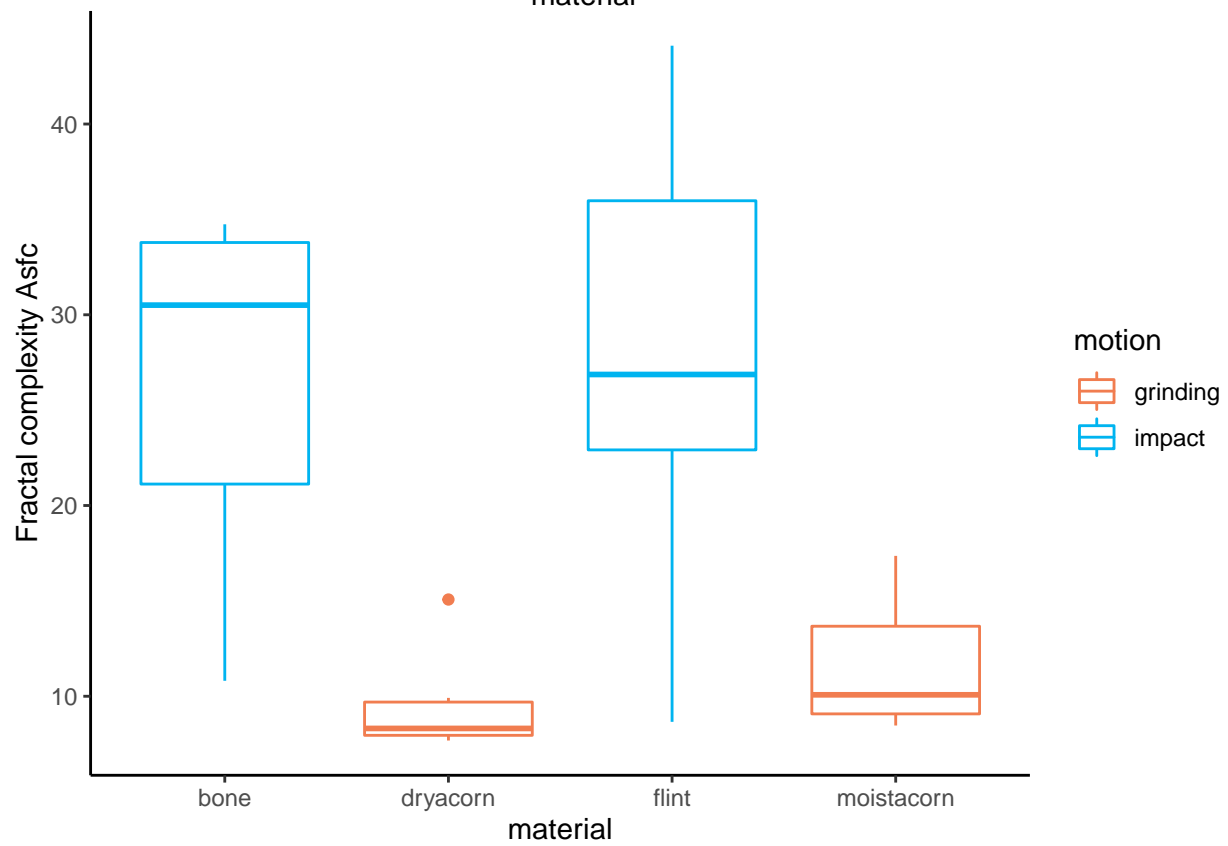
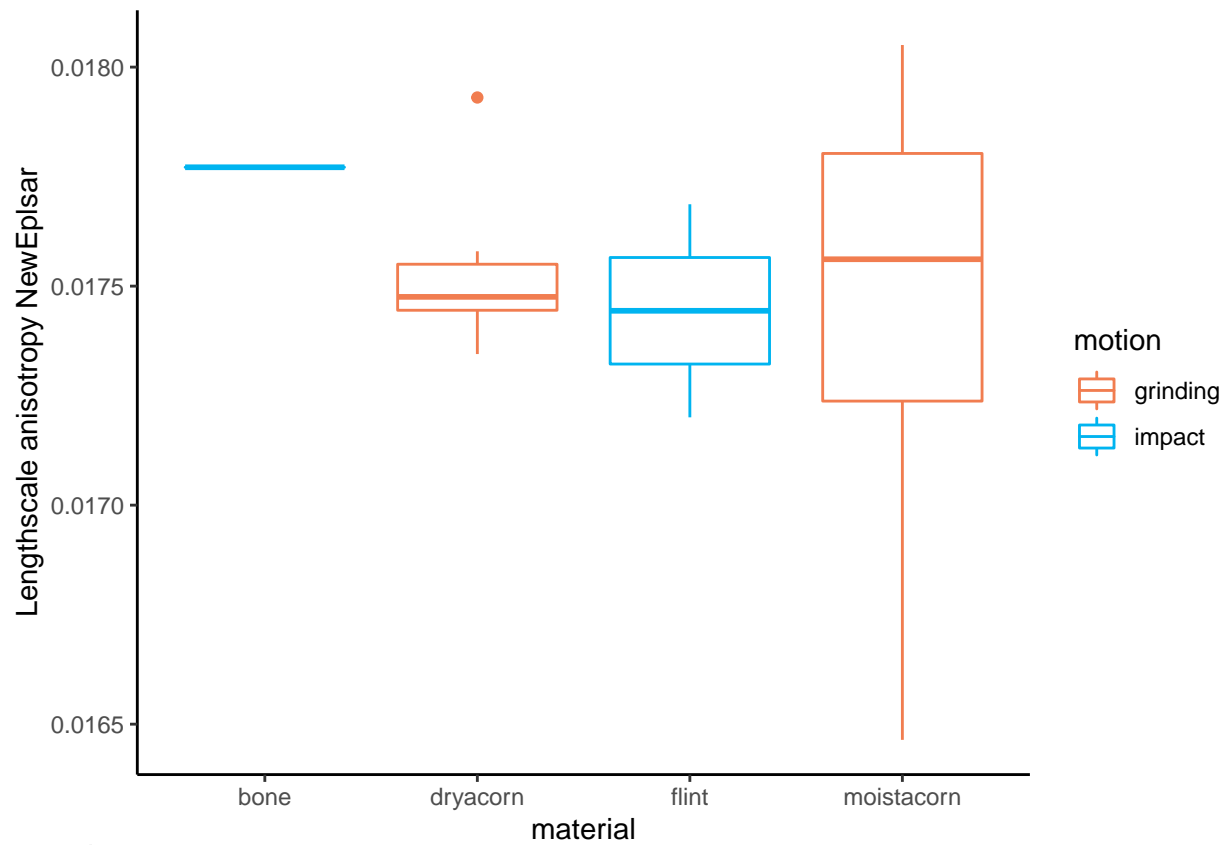
```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```

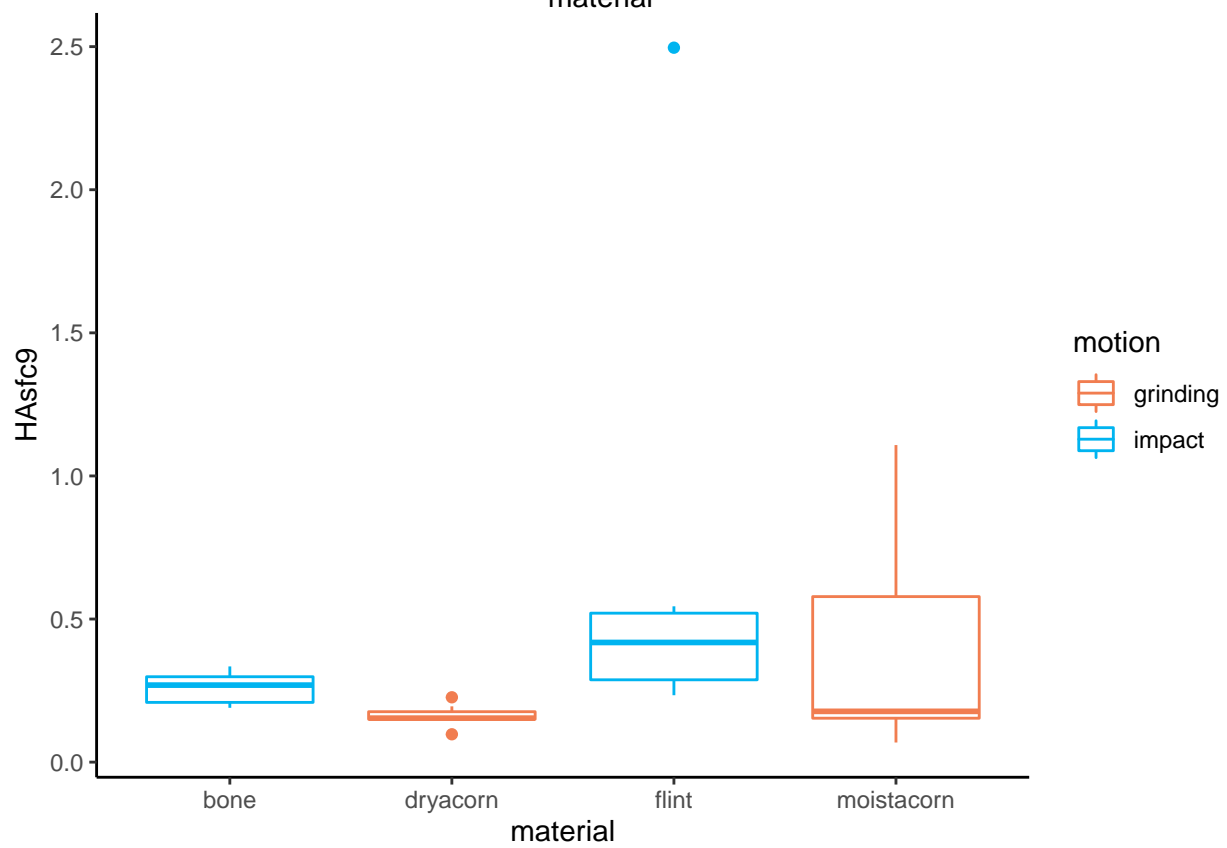
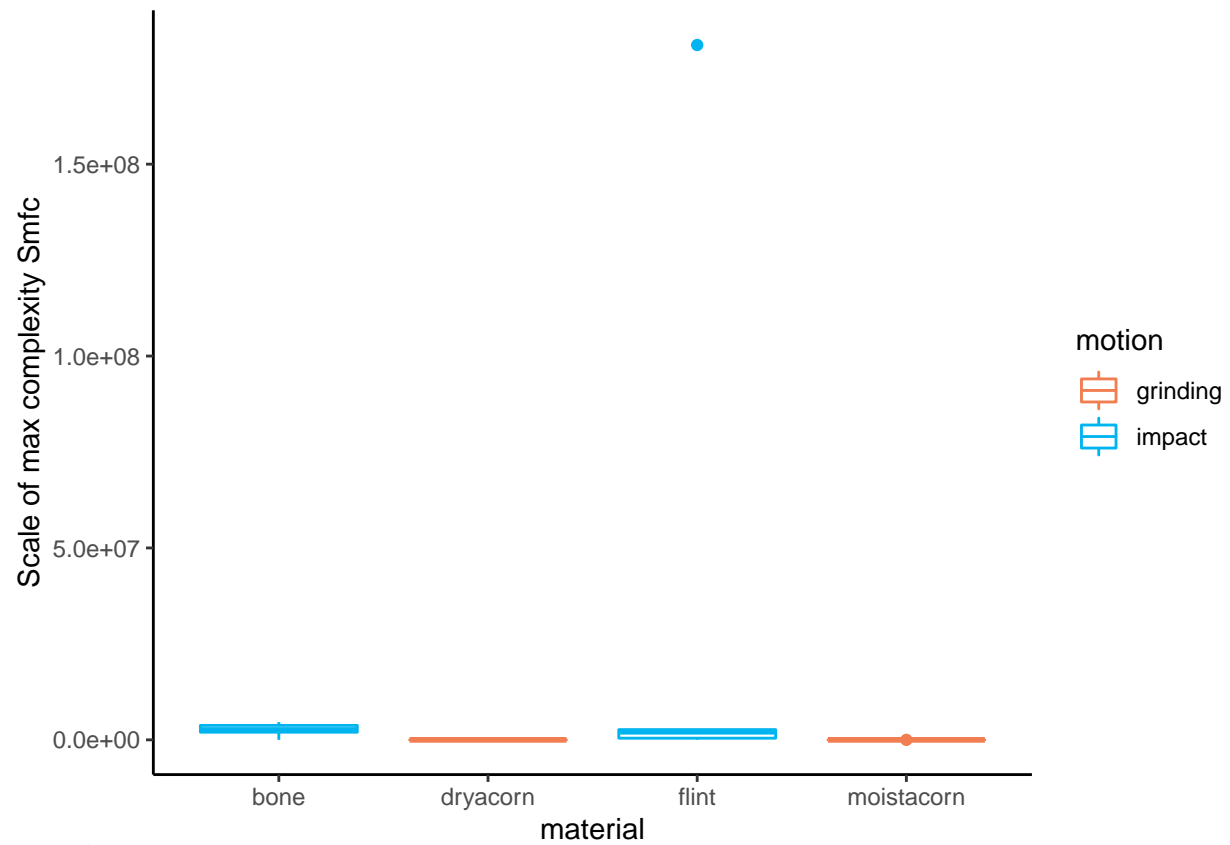


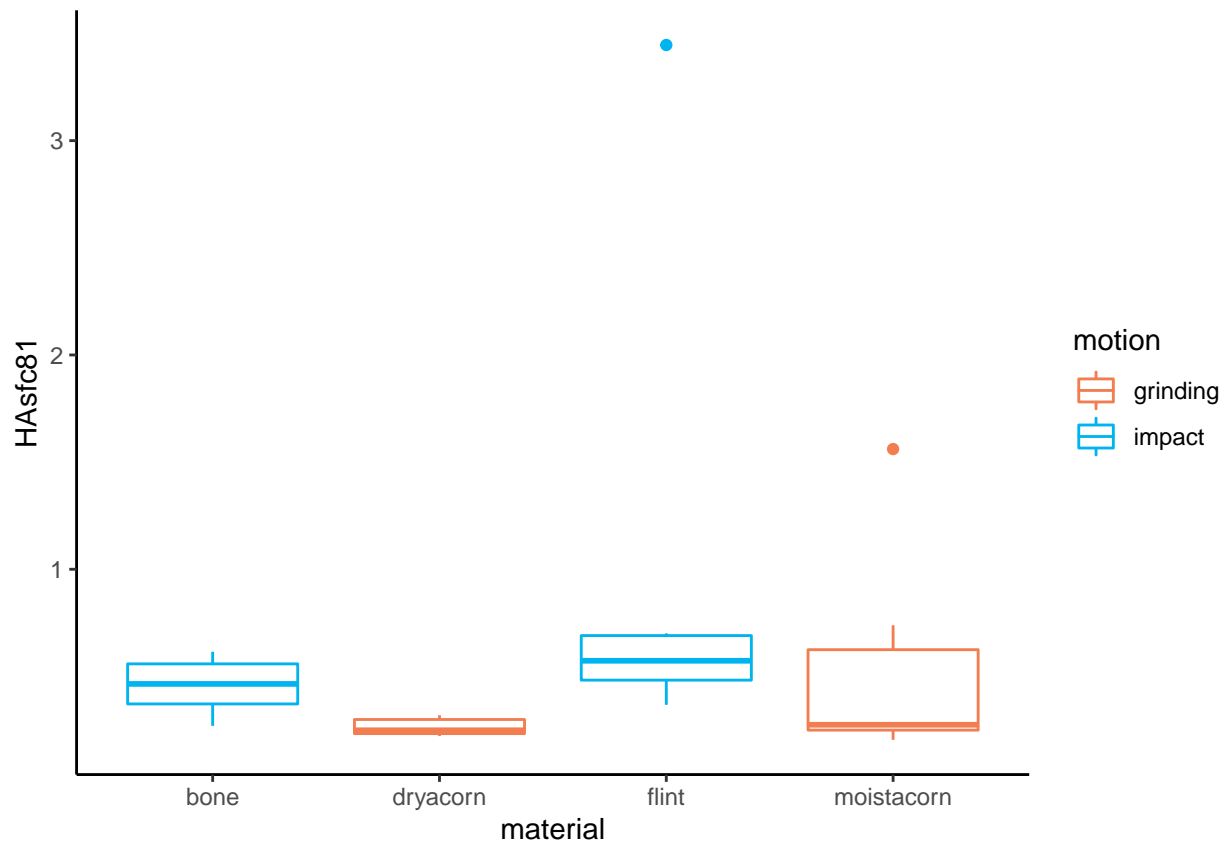
```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```





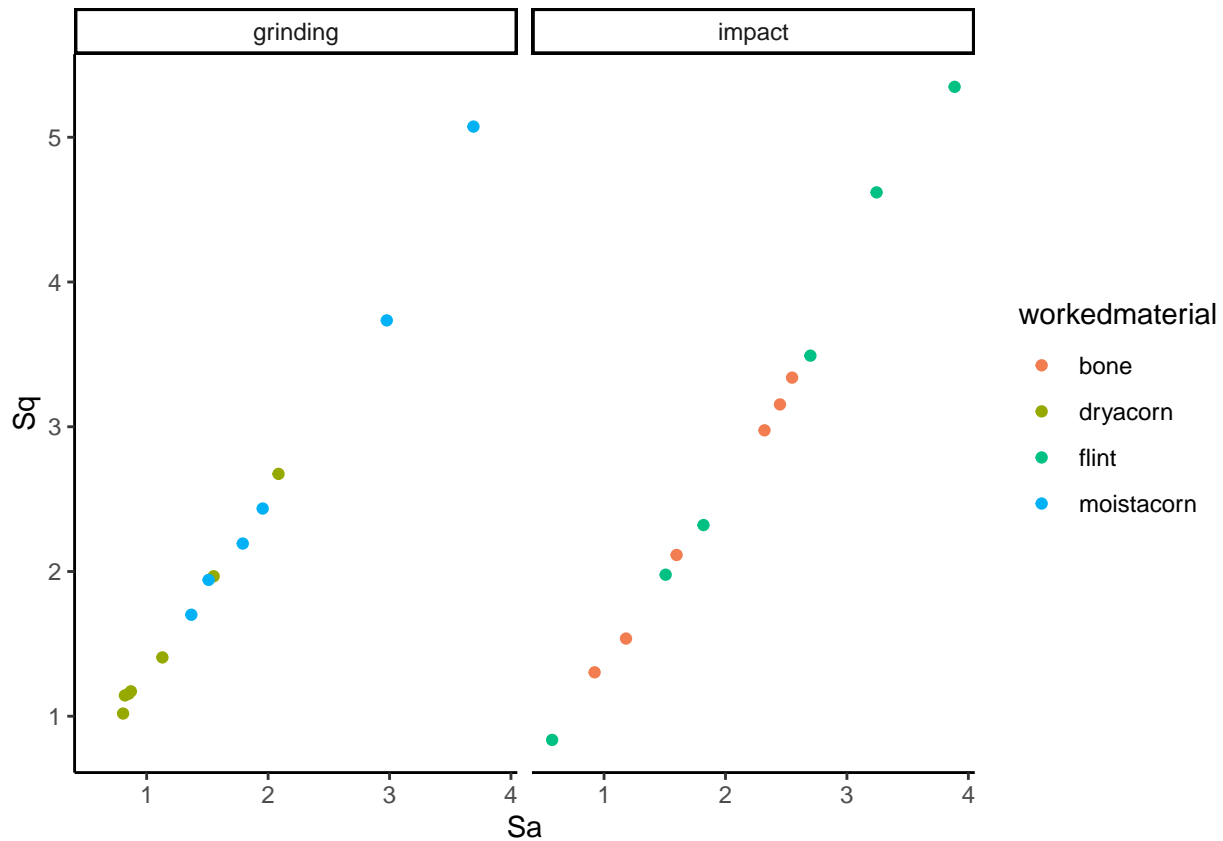




Scatterplots of selected variables combined by “Worked material” and “Motion”

```
# Sa vs. Sq

Sa_Sq <- ggplot(data = confocaldata) +
  geom_point(mapping = aes(x = Sa, y = Sq, colour = workedmaterial)) +
  theme_classic() +
  labs(colour = "workedmaterial") +
  facet_wrap(~ motion) +
  scale_colour_hue(h = c(25, 230))
print(Sa_Sq)
```



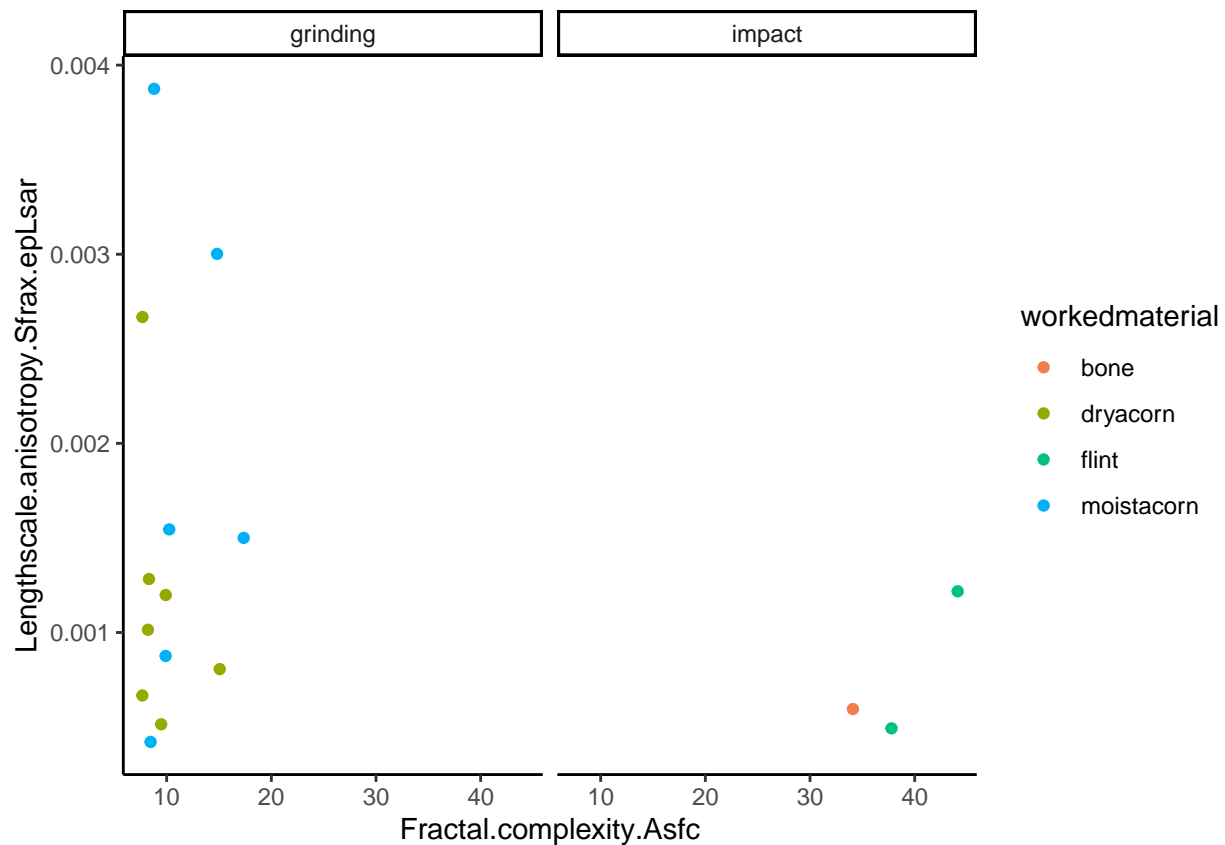
```
file_out <- paste0(file_path_sans_ext(info_in[["file"]]), "_scatterplot_Sa-Sq", ".pdf")
ggsave(filename = file_out, plot = Sa_Sq, path = "../plots", device = "pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
# epLsar vs. Asfc
```

```
ep_As <- ggplot(data = confocaldata) +
  geom_point(mapping = aes(x = Fractal.complexity.Asfc, y = Lengthscale.anisotropy.Sfrax.epLsar,
    theme_classic() +
    labs(colour = "workedmaterial") +
    facet_wrap(~ motion) +
    scale_colour_hue(h = c(25, 230))
print(ep_As)
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



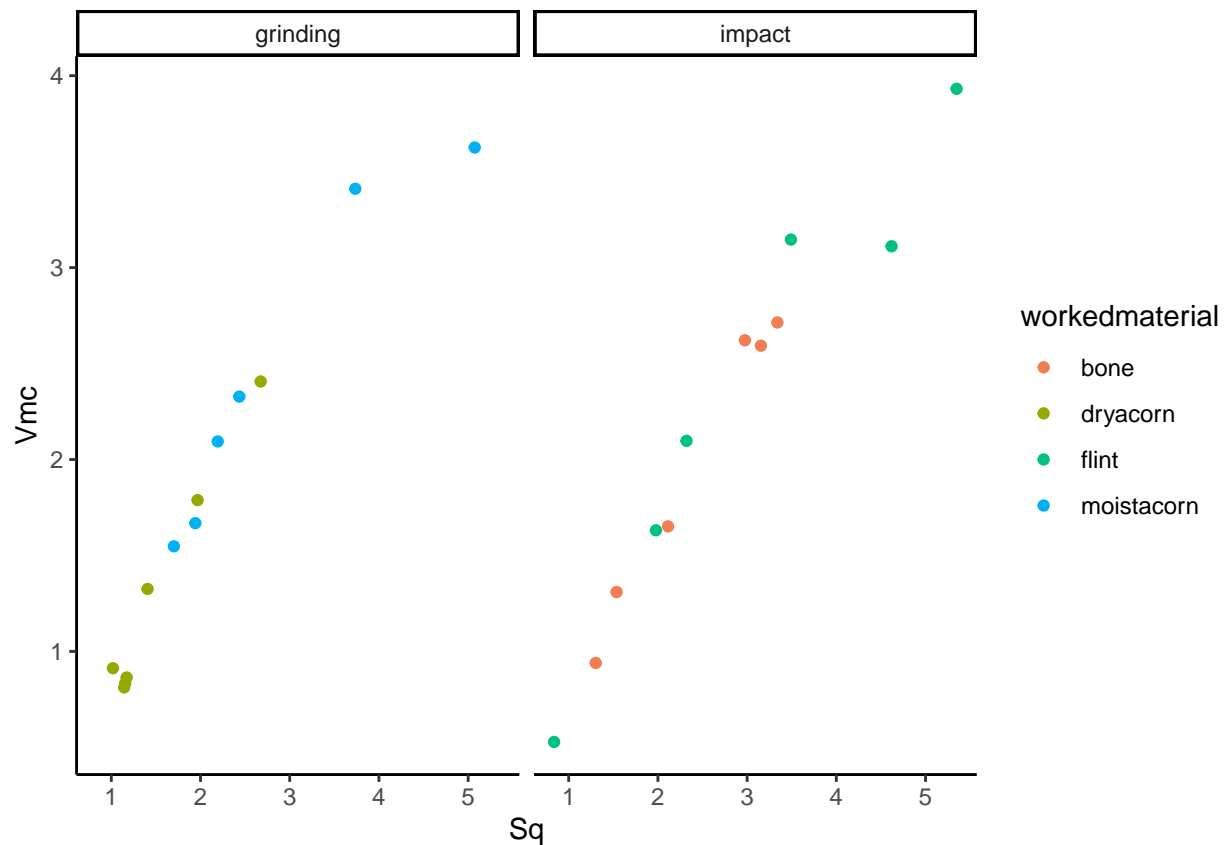
```
file_out <- paste0(file_path_sans_ext(info_in[["file"]]), "_scatterplot_Asfc-epLsar", ".pdf")
ggsave(filename = file_out, plot = ep_As, path = "../plots", device = "pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```

```
# Sq vs. Vmc
```

```
Sq_Vmc <- ggplot(data = confocaldata) +
  geom_point(mapping = aes(x = Sq, y = Vmc, colour = workedmaterial)) +
  theme_classic() +
  labs(colour = "workedmaterial") +
  facet_wrap(~ motion) +
  scale_colour_hue(h = c(25, 230))
print(Sq_Vmc)
```

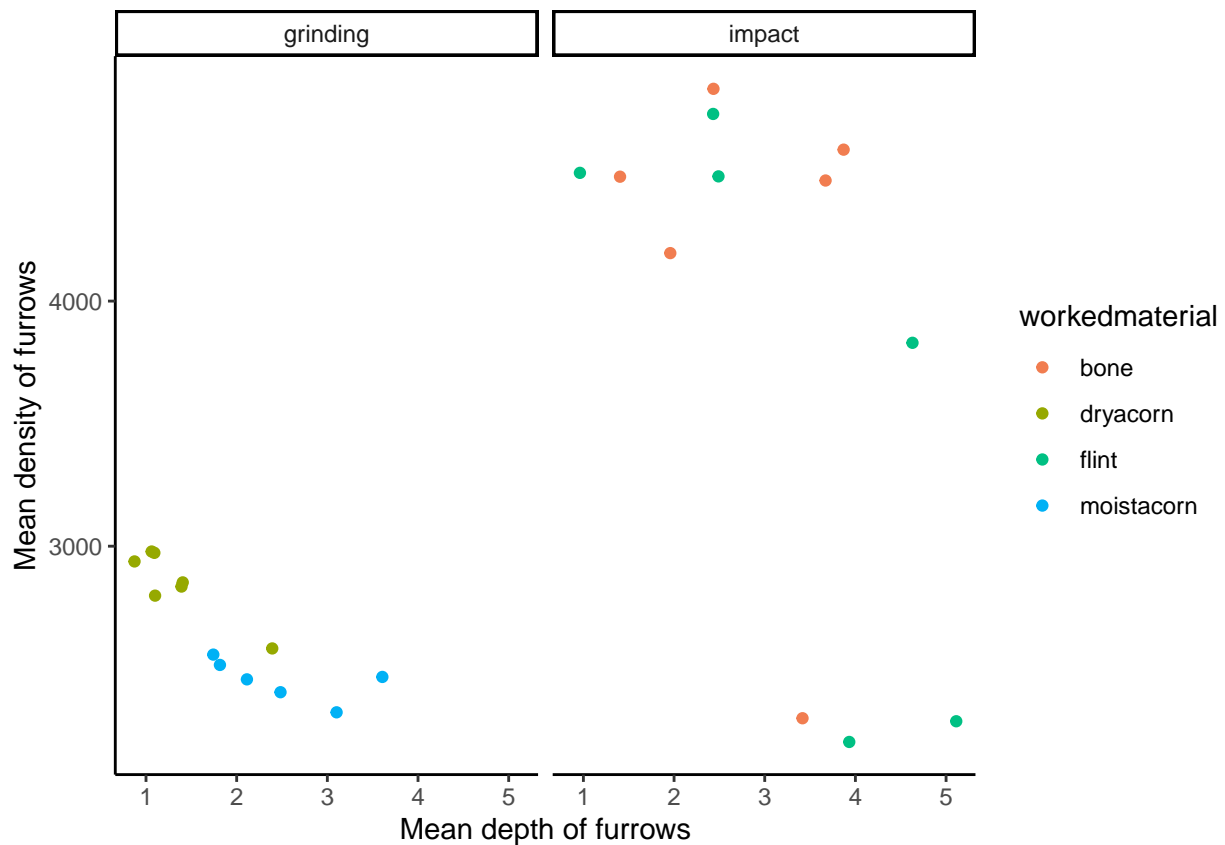


```
file_out <- paste0(file_path_sans_ext(info_in[["file"]]), "_scatterplot_Sq-Vmc", ".pdf")
ggsave(filename = file_out, plot = Sq_Vmc, path = "../plots", device = "pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
# Mean depth of furrows vs. mean density of furrows
```

```
furrows <- ggplot(data = confocaldata) +
  geom_point(mapping = aes(x = Mean.depth.of.furrows, y = Mean.density.of.furrows,
                           colour = workedmaterial)) +
  theme_classic() +
  labs(colour = "workedmaterial", x = "Mean depth of furrows", y = "Mean density of furrows") +
  facet_wrap(~ motion) +
  scale_colour_hue(h = c(25, 230))
print(furrows)
```



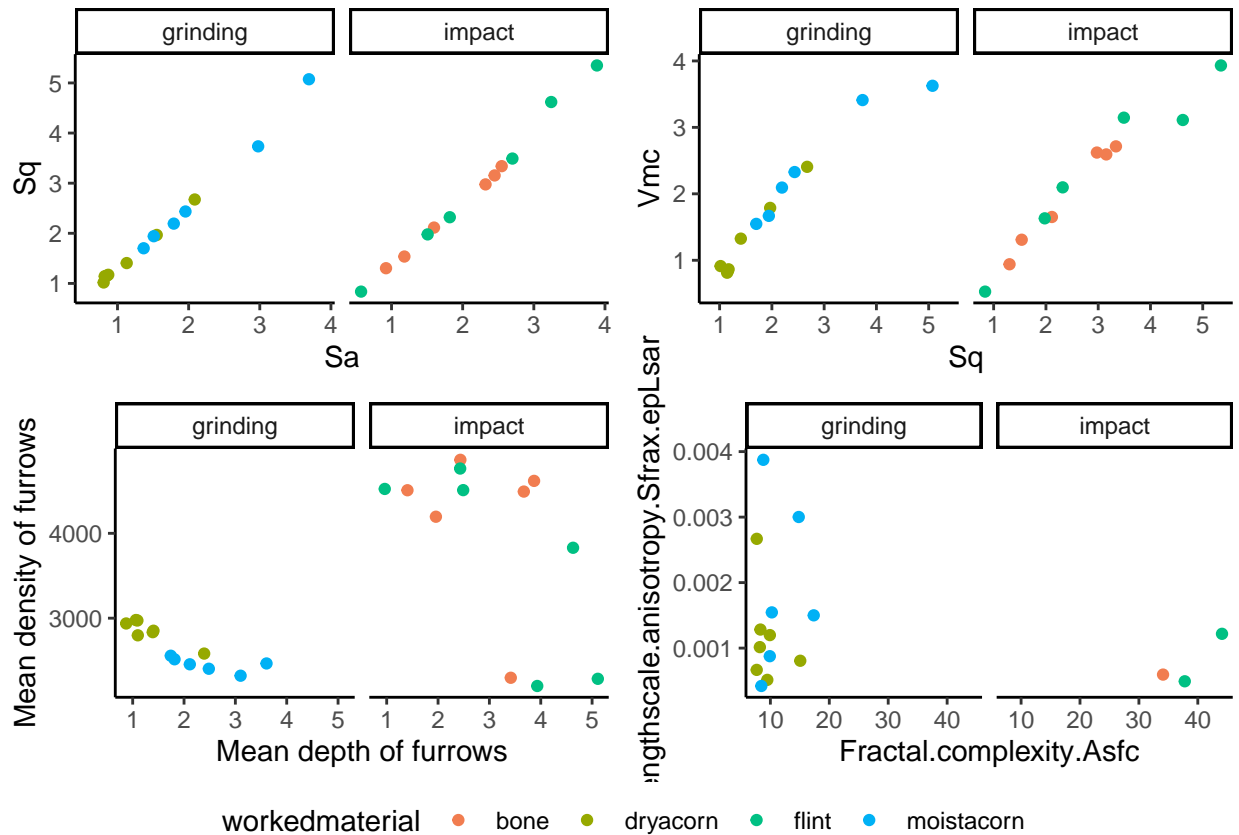
```
file_out <- paste0(file_path_sans_ext(info_in[["file"]]), "_scatterplot_furrows", ".pdf")
ggsave(filename = file_out, plot = furrows, path = "../plots", device = "pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
# combine all in a single image
```

```
ggarrange(Sa_Sq, Sq_Vmc, furrows, ep_As, common.legend = TRUE, legend = "bottom")
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



```
ggsave("../plots/confocalscatterplotsexp.png")
```

```
## Saving 6.5 x 4.5 in image
```

Scatterplot matrix for the ISO 25178 Area scale, Height and volume parameters

```
data(confocaldata, package = "reshape")
```

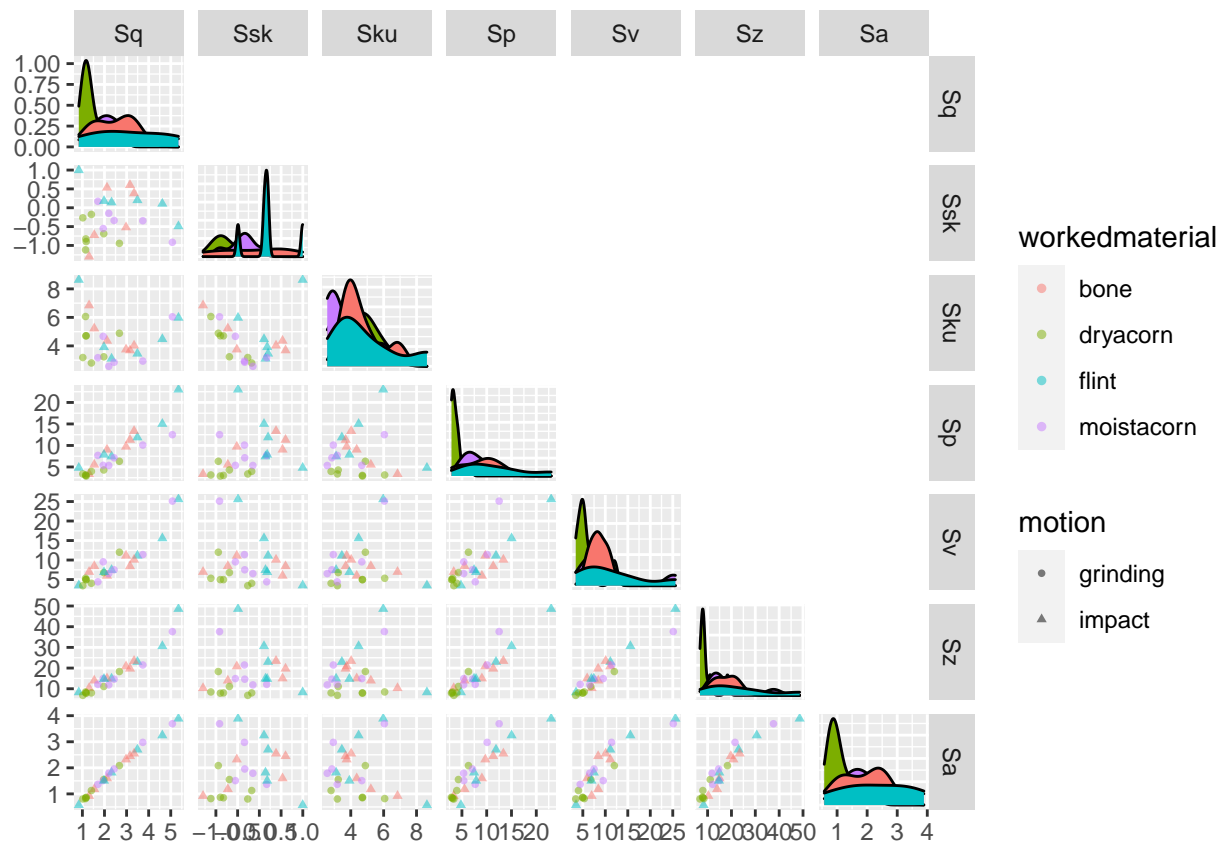
```
## Warning in data(confocaldata, package = "reshape"): data set 'confocaldata' not
## found
```

```
# Height parameters
```

```
ggpairs(data=confocaldata,
  columns = c(21:27),
  cardinality_threshold = 30,
  mapping = ggplot2::aes(color = workedmaterial, shape = motion),
  lower = list(continuous = wrap("points", alpha = 0.5, size = 1)),
  upper = list(continuous = "blank"),
  legend = c(2,1)
) +
```

```
  theme(legend.position = "right") +
  labs(fill = "Micro polish type")
```





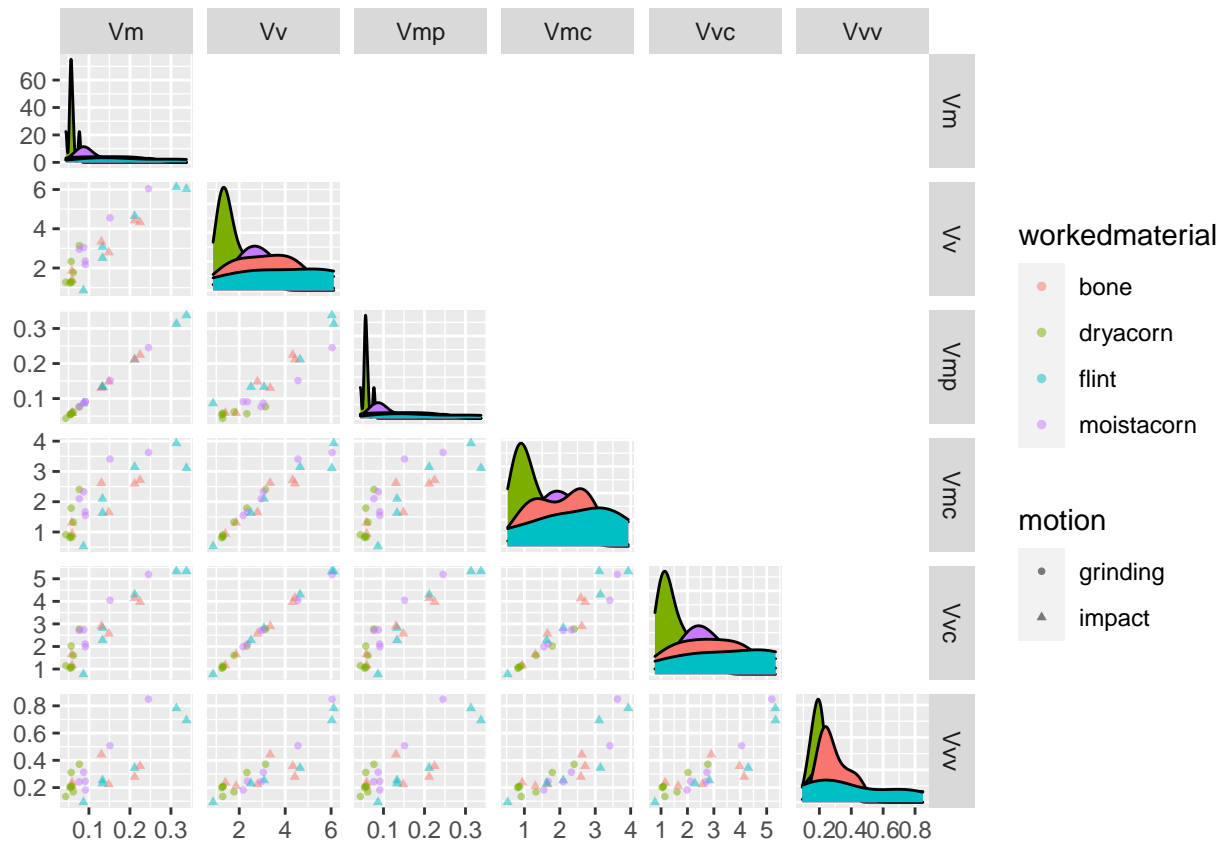
```
ggsave("../plots/confocalarea_matrix.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# Volume parameters
```

```
ggpairs(data=confocaldata,
        columns = c(36:41),
        cardinality_threshold = 30,
        mapping = ggplot2::aes(color = workedmaterial, shape = motion),
        lower = list(continuous = wrap("points", alpha = 0.5, size = 1)),
        upper = list(continuous = "blank"),
        legend = c(2,1)
        ) +

  theme(legend.position = "right") +
  labs(fill = "Micro polish type")
```



```
ggsave("../plots/confocalvolume_matrix.png")
```

```
## Saving 6.5 x 4.5 in image
```

Plot confostats for the ISO 25178 Area-scale, Height and volume parameters

```
# select parameter from dataset

# first Height parameters

heightconfostatsexp <- select(confostatsexp, sample, workedmaterial, Sq.mean, Ssk.mean, Sku.mean, Sp.mean, Sv)

p1 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Sq.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p2 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Ssk.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p3 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Sku.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")
```

```

p4 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Sp.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

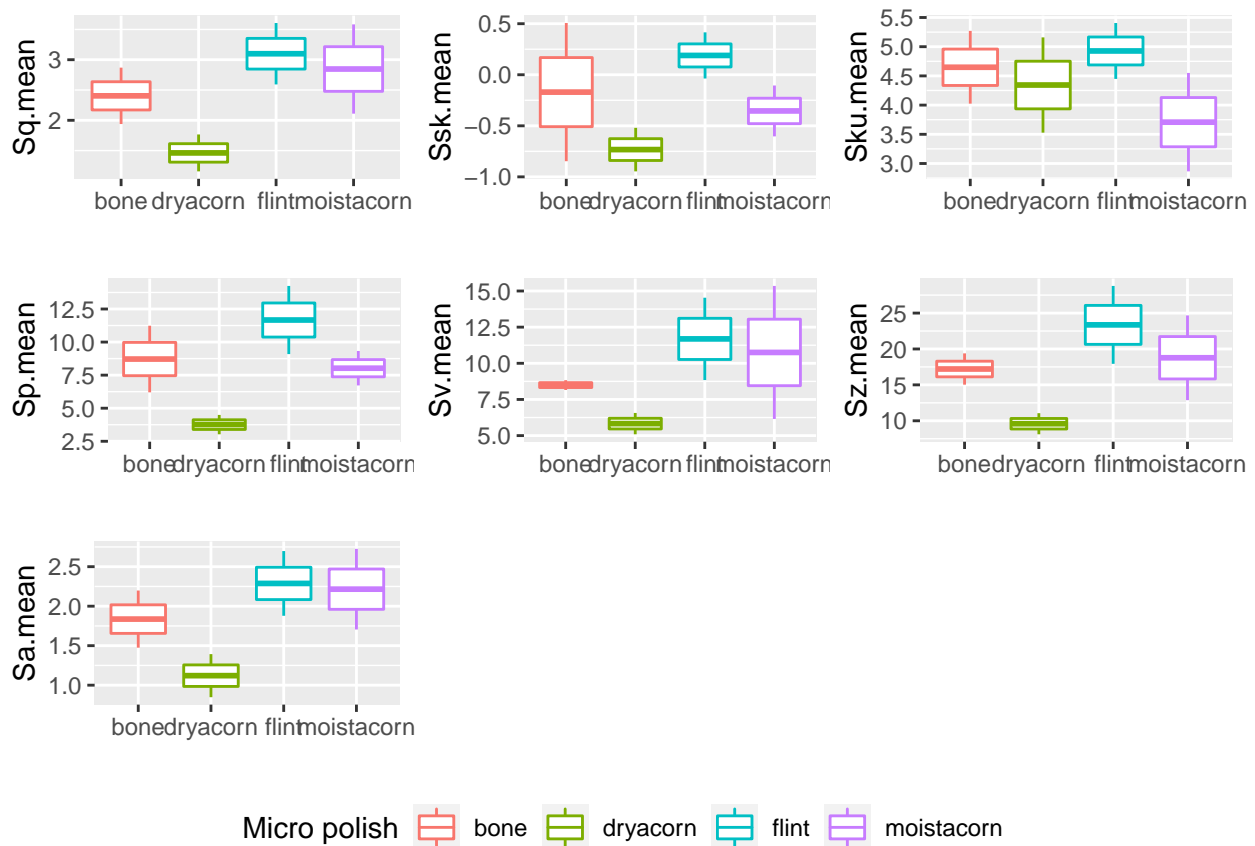
p5 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Sv.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p6 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Sz.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p7 <- ggplot(heightconfostatsexp, aes(x=workedmaterial, y=Sa.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

ggarrange(p1, p2, p3, p4, p5, p6, p7, common.legend = TRUE, font.label = list(size=8), legend="bottom")

```



```
ggsave("../plots/confostatsarea_boxplots.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# Now, compute Volume parameters
```

```
volumeconfostatsexp <- select(confostatsexp, sample, workedmaterial, Vm.mean, Vv.mean, Vmp.mean, Vmc.mean, Vvc.mean)
```

```

p8 <- ggplot(volumeconfostatsexp, aes(x=workedmaterial, y=Vm.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p9 <- ggplot(volumeconfostatsexp, aes(x=workedmaterial, y=Vv.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p10 <- ggplot(volumeconfostatsexp, aes(x=workedmaterial, y=Vmp.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

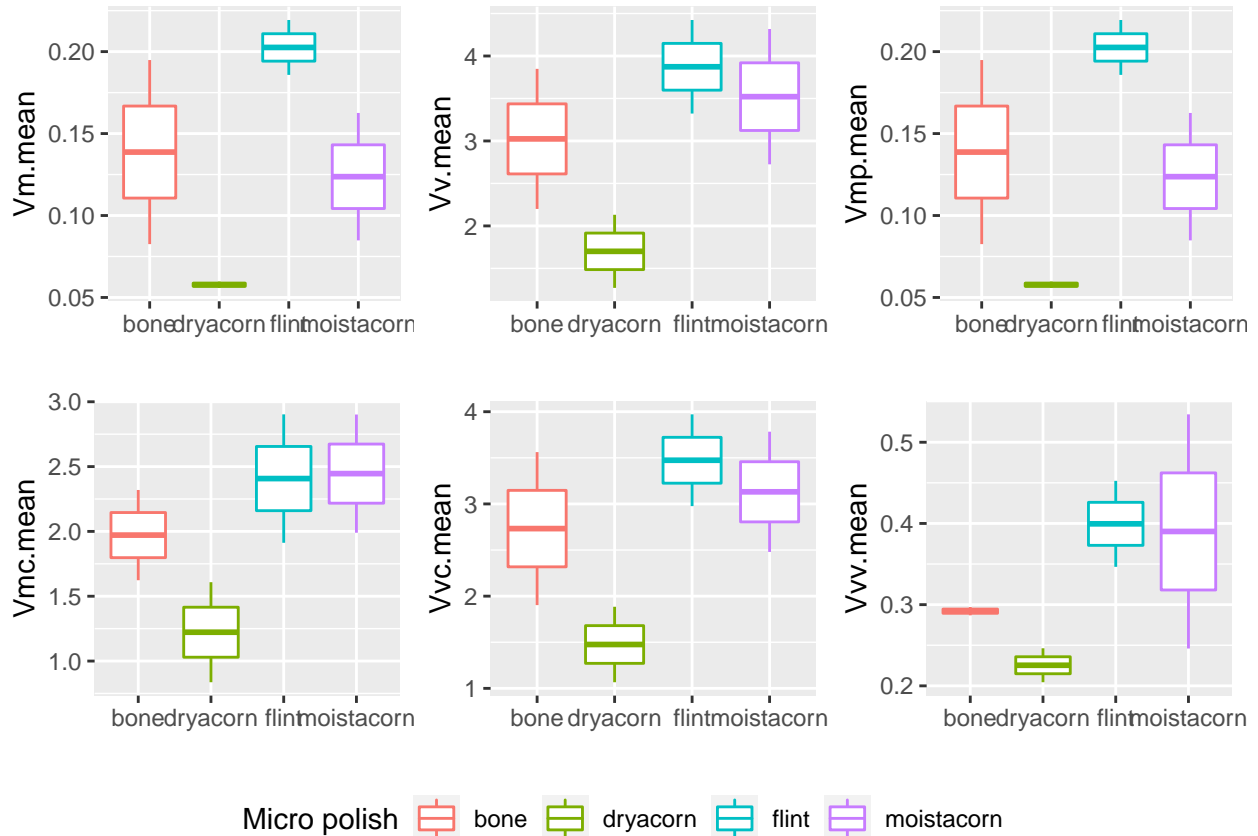
p11 <- ggplot(volumeconfostatsexp, aes(x=workedmaterial, y=Vmc.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p12 <- ggplot(volumeconfostatsexp, aes(x=workedmaterial, y=Vvc.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

p13 <- ggplot(volumeconfostatsexp, aes(x=workedmaterial, y=Vvv.mean, colour=workedmaterial)) +
  geom_boxplot() +
  labs(x="", colour="Micro polish")

ggarrange(p8, p9, p10, p11, p12, p13, common.legend = TRUE, font.label = list(size=8), legend="bottom")

```



```
ggsave("../plots/confostatsvolume_boxplots.png")
```

```
## Saving 6.5 x 4.5 in image
```

## Show plot files information

```
info_out <- list.files(path = "derived_data", pattern = "\\..pdf$", full.names = TRUE) %>%  
  md5sum()
```

## End and Session info

```
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] tools stats graphics grDevices utils datasets methods  
## [8] base  
##  
## other attached packages:  
## [1] ggfortify_0.4.11 ggpubr_0.4.0 doBy_4.6.9 GGally_2.1.1  
## [5] kableExtra_1.3.4 janitor_2.1.0 knitr_1.31 forcats_0.5.1  
## [9] stringr_1.4.0 dplyr_1.0.5 purrr_0.3.4 readr_1.4.0  
## [13] tidyr_1.1.3 tibble_3.1.0 ggplot2_3.3.3 tidyverse_1.3.0  
##  
## loaded via a namespace (and not attached):  
## [1] httr_1.4.2 jsonlite_1.7.2 viridisLite_0.3.0 carData_3.0-4  
## [5] modelr_0.1.8 assertthat_0.2.1 highr_0.8 cellranger_1.1.0  
## [9] yaml_2.2.1 pillar_1.5.1 backports_1.2.1 lattice_0.20-41  
## [13] glue_1.4.2 digest_0.6.27 RColorBrewer_1.1-2 ggsignif_0.6.1  
## [17] rvest_1.0.0 snakecase_0.11.0 colorspace_2.0-0 cowplot_1.1.1  
## [21] htmltools_0.5.1.1 Matrix_1.3-2 plyr_1.8.6 pkgconfig_2.0.3  
## [25] broom_0.7.5 haven_2.3.1 scales_1.1.1 webshot_0.5.2  
## [29] svglite_2.0.0 openxlsx_4.2.3 rio_0.5.26 farver_2.1.0  
## [33] generics_0.1.0 car_3.0-10 ellipsis_0.3.1 withr_2.4.1  
## [37] cli_2.3.1 magrittr_2.0.1 crayon_1.4.1 readxl_1.3.1  
## [41] evaluate_0.14 fs_1.5.0 fansi_0.4.2 MASS_7.3-53.1  
## [45] rstatix_0.7.0 xml2_1.3.2 foreign_0.8-81 data.table_1.14.0
```

## [49]	hms_1.0.0	lifecycle_1.0.0	munsell_0.5.0	reprex_1.0.0
## [53]	zip_2.1.1	compiler_4.0.4	Deriv_4.1.3	systemfonts_1.0.1
## [57]	rlang_0.4.10	grid_4.0.4	rstudioapi_0.13	labeling_0.4.2
## [61]	rmarkdown_2.7	gtable_0.3.0	abind_1.4-5	DBI_1.1.1
## [65]	reshape_0.8.8	curl_4.3	R6_2.5.0	gridExtra_2.3
## [69]	lubridate_1.7.10	utf8_1.2.1	stringi_1.5.3	Rcpp_1.0.6
## [73]	vctrs_0.3.6	dbplyr_2.1.0	tidyselect_1.1.0	xfun_0.22