liv_data

Exploratory analysis of Liv's data

This report contains R code and output intermingled using Quarto, based on the LIterate Programmign paradigm of Knuth.

The input to this code is a simplified version of Liv's .xlsx spreadsheets with extraneous material removed to make them easier to parse.

Load the necessary libraries and calculate the approximate density of each cushion. Note that this is a guess as there is VF and HR in each cushion, but only one weight of the combined cushion.

```
# Load required libraries.
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(TOSTER)
library(ggplot2)
```

Read the recorded measurements and mass and calculate the density. The `density_spec` table is from the Dunlop specification sheets.

```r
# Deal with density data separately as very different structure
# and needs several calculations to get the (approximate) density.

liv_density <- readxl::read_xlsx(here::here("data", "density.xlsx"),
                                  range = readxl::cell_cols("A:K"))
# Specification from Dunlop, in kg/m3
density_spec <- tibble::tribble(
  ~foam, ~min, ~max,
  "VF",       58.0, 62.0,
  "EN40-230", 39.0, 42.0,
  "EN50-250", 49.0, 53.5
)

# Calculate various volumes in mm3
liv_density <- liv_density %>%
  mutate(t_mean = rowMeans(across(starts_with("T"))),
         vf_vol = length * width * vf_thickness,
         sag_vol = (sag_height * sag_width * length), # cutout for seat sag
         hr_vol = length * width * t_mean - vf_vol - sag_vol
        ) %>%
# The mass of VF is unknown, but can be estimated from the density_spec
  mutate(
    vf_min_mass = (vf_vol / 1E9) * filter(density_spec, foam == "VF")$max * 1000,
    vf_max_mass = (vf_vol / 1E9) * filter(density_spec, foam == "VF")$min * 1000,
    hr_max_mass = mass - vf_min_mass, # grams
    hr_min_mass = mass - vf_max_mass,
    hr_min = 1E6 * hr_min_mass / hr_vol,  # Convert back to kg/m3
    hr_max = 1E6 * hr_max_mass / hr_vol
  )
```

Now load the test results, rearrange them for easier autmated processing and calculate some summary staistices.

```r
# Read the testing results
data_file <- here::here("data", "results.xlsx")

# Display sheet names
sheet_names <- readxl::excel_sheets(data_file)
print(sheet_names)
```

```
[1] "lcdod"      "hysteresis"
```

```r
# Read the two sheets constructed from Liv's data & name them
# Convert to long form for easier analysis,
# then combine both data sets by row and
# group by foam type, variable and the load level for each result
liv_data <- sheet_names %>%                  # Take the named sheets
  purrr::set_names() %>%                     # make them a named list
  purrr::map(                                # apply a function to each element
    function(x) {
      readxl::read_excel(data_file, x) %>%   # read the named sheet
      tidyr::pivot_longer(cols = where(is.numeric), # use the results columns
                    names_to = "level",      # column names to 'level'
                    values_to = "value"      # each value to 'value'
                   )
    }
  ) %>%
  bind_rows(.id = "var") %>%      # combine by rows, put original names in 'var'
  group_by(foam, var, level)      # make groups for each independent measurement

# liv_data is now a list with columns named
# var = lcdod or hysteresis
# cushion = cushion ID number
# foam = EN40-230 or EN50-250
# level = load level for measurement
# value = measurement

# Define a function to calculate the span of the given percentile
# percentile defaults to 95% (0.95)
CI = function(sd, percentile = 0.95) {
  interval = (percentile + 1)/2
  qnorm(p = interval, mean = 0, sd = sd)
}

# Show the Summary stats for each variable,including the 95% CI
liv_summary <- liv_data %>%
  summarise(avg = mean(value), sd = sd(value), n = n()) %>%
  mutate(delta = CI(sd, 0.95), lo_95 = avg - delta, hi_95 = avg + delta) %>%
  select(-delta) %>%
  arrange(var, level, foam)
```

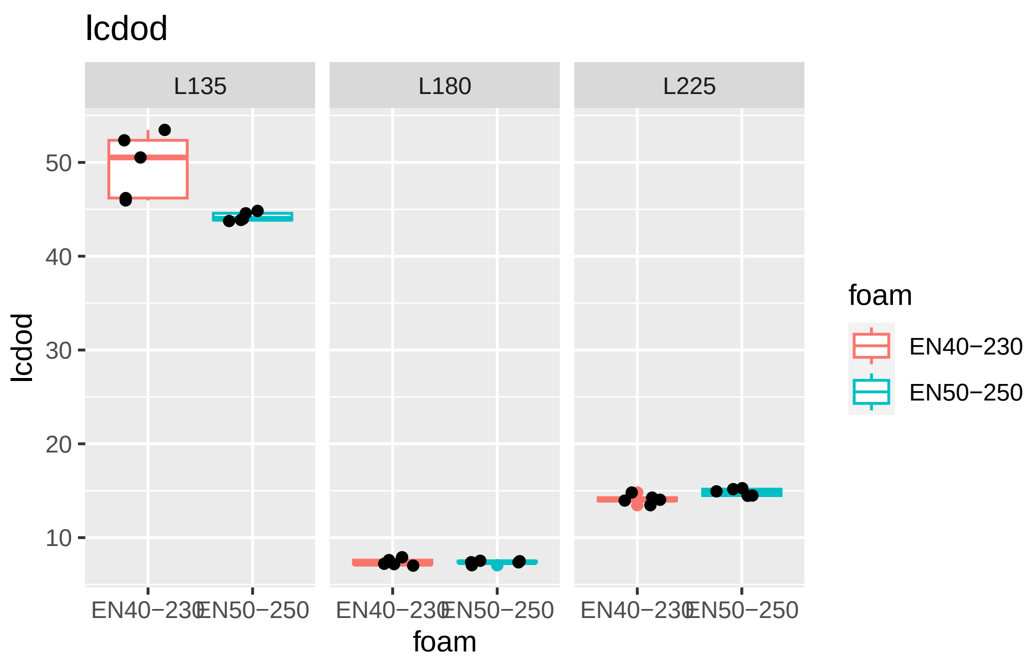
```
`summarise()` has grouped output by 'foam', 'var'. You can override using the
`.groups` argument.
```

```r
print(liv_summary)
```

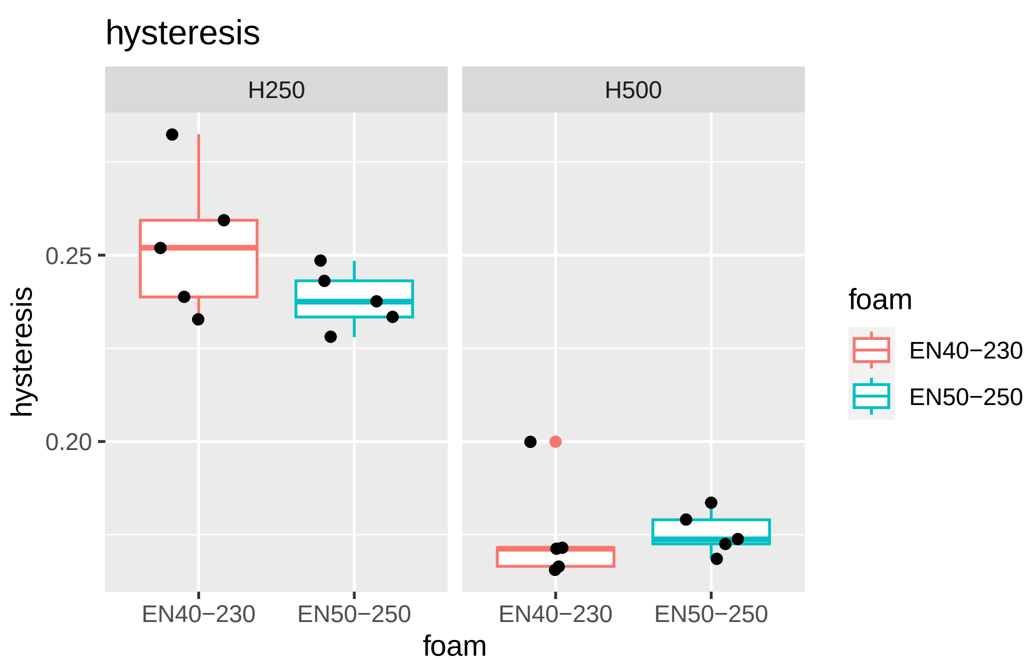
```
# A tibble: 10 x 8
# Groups:   foam, var [4]
   foam     var        level    avg      sd     n  lo_95  hi_95
   <chr>    <chr>      <chr>  <dbl>   <dbl> <int>  <dbl>  <dbl>
 1 EN40-230 hysteresis H250   0.253 0.0195      5  0.215  0.291
 2 EN50-250 hysteresis H250   0.238 0.00802     5  0.222  0.254
 3 EN40-230 hysteresis H500   0.175 0.0142      5  0.147  0.203
 4 EN50-250 hysteresis H500   0.175 0.00587     5  0.164  0.187
 5 EN40-230 lcdod      L135  49.7   3.47        5 42.9   56.5
 6 EN50-250 lcdod      L135  44.2   0.476       5 43.3   45.1
 7 EN40-230 lcdod      L180   7.38  0.362       5  6.67   8.09
 8 EN50-250 lcdod      L180   7.36  0.186       5  7.00   7.73
 9 EN40-230 lcdod      L225  14.1   0.492       5 13.1   15.1
10 EN50-250 lcdod      L225  14.9   0.372       5 14.1   15.6
```

```r
# Make some simple box plots to show the spread of the data
plots <- list()
for (var_name in unique(liv_data$var)) {
  plots[[var_name]] <- liv_data %>%
    filter(var == var_name) %>%
    ggplot() +
    aes(y = value, x = foam, colour = foam) +
    geom_boxplot() +
    geom_jitter(colour = "black", width = 0.25) +
    labs(title = var_name, y = var_name) +
    facet_wrap(~level)
}
print(plots)
```

```
$lcdod
```


lcdod
L135
L180
L225
50
40
30
20
10
lcdod
EN40−230
EN50−250
foam
foam
EN40−230
EN50−250

$hysteresis


hysteresis
H250
H500
0.25
0.20
hysteresis
EN40−230
EN50−250
foam
foam
EN40−230
EN50−250

TOST or equivalence testing

Run a **T**wo **O**ne **S**ided **T**est on each comparison group (i.e. compare foams for each measured variable and level)

```
# Want to compare foams for each variable and level.
# Regroup the data with the required groups them map a function to each group
result_list <- liv_data %>%
  ungroup() %>% # remove the old groups first
  group_by(var, level) %>% # then regroup
  group_map(
    function(data = .x, group = foam, values = value) {
      # t_TOST compares two vectors, create those from
      # the groups defined by the `group` column
      #df <- select(data, !!group, !!values)
      bits <- split(data, ~ {{group}}) # this sprays Warnings
      # then extract the values to compare from the `values` column
      x <- pull(bits[[1]], {{values}});
      y <- pull(bits[[2]], {{values}});
      # Set the size of the equivalence bounds at 10% of the combined mean
      mean_all = mean(c(x, y)) / 10;
      # Then do a TOST using that bound size (half above and half below).
      TOSTER::t_TOST(x, y, eqb = mean_all / 2)
    }
  )
```

```
Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
# Lots of fuss to get the group names back to name the result list
keys <- liv_data %>%
    ungroup() %>% # remove the old groups first
    group_by(var, level) %>% group_keys()
result_names <- paste(keys$var, keys$level)

names(result_list) <- result_names
```

```
# Turn output back on
#| output: asis
print(result_list)
```

```
$`hysteresis H250`
```

```
Welch Two Sample t-test

The equivalence test was non-significant, t(5.33) = 0.84, p = 0.22
The null hypothesis test was non-significant, t(5.33) = -0.307p = 0.77
NHST: don't reject null significance hypothesis that the effect is equal to zero
TOST: don't reject null equivalence hypothesis

TOST Results
                 t    df p.value
t-test     -0.3066 5.325   0.771
TOST Lower  0.8353 5.325   0.220
TOST Upper -1.4485 5.325   0.102

Effect Sizes
                Estimate      SE              C.I. Conf. Level
Raw            -0.003297 0.01075 [-0.0247, 0.0181]         0.9
Hedges's g(av) -0.165019 0.81514 [-1.0466, 0.7315]         0.9
Note: SMD confidence intervals are an approximation. See vignette("SMD_calcs").

$`hysteresis H500`

Welch Two Sample t-test

The equivalence test was non-significant, t(5.8) = -0.68, p = 0.26
The null hypothesis test was non-significant, t(5.8) = 0.622p = 0.56
NHST: don't reject null significance hypothesis that the effect is equal to zero
TOST: don't reject null equivalence hypothesis

TOST Results
                 t    df p.value
t-test      0.6223 5.797   0.557
TOST Lower  1.9248 5.797   0.052
TOST Upper -0.6803 5.797   0.261

Effect Sizes
               Estimate       SE              C.I. Conf. Level
Raw            0.004185 0.006726  [-0.009, 0.0173]         0.9
Hedges's g(av) 0.339947 0.837196 [-0.5868, 1.2387]         0.9
Note: SMD confidence intervals are an approximation. See vignette("SMD_calcs").

$`lcdod L135`
```

```
Welch Two Sample t-test

The equivalence test was non-significant, t(6.72) = 0.0066, p = 0.5
The null hypothesis test was non-significant, t(6.72) = -0.989p = 0.36
NHST: don't reject null significance hypothesis that the effect is equal to zero
TOST: don't reject null equivalence hypothesis

TOST Results
                  t    df p.value
t-test     -0.98934 6.717   0.357
TOST Lower  0.00659 6.717   0.497
TOST Upper -1.98527 6.717   0.045

Effect Sizes
               Estimate     SE              C.I. Conf. Level
Raw             -2.3320 2.3571 [-6.8262, 2.1622]         0.9
Hedges's g(av)  -0.5527 0.8754 [-1.4839, 0.4171]         0.9
Note: SMD confidence intervals are an approximation. See vignette("SMD_calcs").

$`lcdod L180`

Welch Two Sample t-test

The equivalence test was non-significant, t(5.58) = -1.8, p = 0.07
The null hypothesis test was non-significant, t(5.58) = 0.283p = 0.79
NHST: don't reject null significance hypothesis that the effect is equal to zero
TOST: don't reject null equivalence hypothesis

TOST Results
                 t   df p.value
t-test      0.2833 5.58   0.787
TOST Lower  2.3173 5.58   0.031
TOST Upper -1.7507 5.58   0.067

Effect Sizes
               Estimate     SE              C.I. Conf. Level
Raw             0.05133 0.1812 [-0.3056, 0.4083]         0.9
Hedges's g(av)  0.15376 0.8018 [-0.7488, 1.0429]         0.9
Note: SMD confidence intervals are an approximation. See vignette("SMD_calcs").

$`lcdod L225`

Welch Two Sample t-test
```

```
The equivalence test was non-significant, t(8) = -0.65, p = 0.27
The null hypothesis test was non-significant, t(8) = 1.476p = 0.18
NHST: don't reject null significance hypothesis that the effect is equal to zero
TOST: don't reject null equivalence hypothesis

TOST Results
                  t    df p.value
t-test      1.4760 7.999   0.178
TOST Lower  3.5993 7.999   0.003
TOST Upper -0.6473 7.999   0.268

Effect Sizes
               Estimate     SE              C.I. Conf. Level
Raw              0.5033 0.3410 [-0.1308, 1.1375]         0.9
Hedges's g(av)   0.8427 0.9495 [-0.1803, 1.8176]         0.9
Note: SMD confidence intervals are an approximation. See vignette("SMD_calcs").
```