MSRS LME with Random Effects

Harshi Gupta

12/14/2019

Setup

In this section, please change the "SC_category" variable to the correct concentration. R script is run 5 times to get all the concentrations.

```
################################################## Set Working Directory
setwd("C:/Users/HGupta/Documents/Hopkins/MSRS/2020/Data/Results_Copy")

### clear the workspace
rm(list = ls())

################################################## Load libraries
library(lme4)
library(lattice)
library(reshape2)
library(dplyr)
library(openxlsx)
library(readxl)
library(stats)
library(data.table)
library(psych)
library(ggplot2)
library(knitr)
library(knitr)

opts_chunk$set(dev = "png")
# Ensure PDF isn't cut off
opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)

# File naming information
SC_category = "clinical"

# Import the names of the judges for the given SC_category (i.e. concentration)
# above.
raw_judges <- read.table(file = paste(SC_category, "_judges.txt", sep = ""), header = FALSE,
    sep = "\t", quote = "", stringsAsFactors = FALSE)
# Check to see if this looks right.
raw_judges
```

```
##                    V1
## 1        Aarti Mathur
```

```
## 2    Alejandro Garcia
## 3         Anne Murphy
## 4       Caitlin Hicks
## 5        Corey Tapper
## 6     Cozumel Pruette
## 7    Debraj Mukherjee
## 8   Douglas Gladstone
## 9     Elisabeth Marsh
## 10     Elizabeth Wise
## 11  Gerald Brandacher
## 12    Gislin Dagnelie
## 13      James Ferriss
## 14      Kristin Bibee
## 15 meghan berkenstock
## 16     Paul Rosenberg
## 17    paul sponseller
## 18      Rafael Llinas
## 19   Rajani Sebastian
## 20 Raul Chavez-Valdez
## 21       Sanjay Desai
## 22           Som Saha
## 23     Stefano Schena
## 24       Thomas Smith
## 25         Tim Witham
## 26          Tina Tran
```

Import the raw data, which includes each judges' scores for each abstract.

```
# ############ Data ## Import the raw data from the Excel
# file. Check Github repository for proper format for input
# file.
all_data <- read_excel("all_raw.xlsx")
summary(all_data)
```

```
##  Please enter your name in the following format.
##  Length:608
##  Class :character
##  Mode  :character
##
##
##
##  Please enter the 3 digit abstract code.
##  Min.   :101.0
##  1st Qu.:297.0
##  Median :588.0
##  Mean   :561.5
##  3rd Qu.:796.0
##  Max.   :995.0
##  Clear Goals and Rationale: Was the student able to adequately articulate the background of their to
##  Min.   :1.000
##  1st Qu.:4.000
##  Median :5.000
##  Mean   :5.076
```

```
##  3rd Qu.:6.000
##  Max.   :7.000
##  Appropriate Methods: Did the student propose and carry out a scholarly approach that would appropria
##  Min.   :1.000
##  1st Qu.:4.000
##  Median :5.000
##  Mean   :4.819
##  3rd Qu.:6.000
##  Max.   :7.000
##  Effective Presentation: Was the student able to present the information in an orderly way?  What was
##  Min.   :1.00
##  1st Qu.:4.00
##  Median :5.00
##  Mean   :4.77
##  3rd Qu.:6.00
##  Max.   :7.00
##  Conclusions: Was there critical reflection on findings, limitations, and/or the direction of furthe
##  Min.   :1.000
##  1st Qu.:4.000
##  Median :5.000
##  Mean   :4.558
##  3rd Qu.:6.000
##  Max.   :7.000
```

```r
# Reorder columns. Add the total score for judges and remove
# extraneous columns.
all_data$Score <- rowSums(all_data[3:6])
all_data <- all_data[-c(3:6)]

# Rename columns in dataframe to Judge, ID, Score
colnames(all_data) <- c("Judge", "ID", "Score")

# Extract values for judges in category assigned above
raw_data <- all_data[all_data$Judge %in% raw_judges[, 1], ]

# Order dataframe by name of Judge
raw_data <- raw_data[order(raw_data$Judge), ]


adjusted_data <- raw_data  #rename raw data

# Convert ID column to a character class.
adjusted_data$ID <- as.character(adjusted_data$ID)

str(adjusted_data)
```

```
## tibble [260 x 3] (S3: tbl_df/tbl/data.frame)
##  $ Judge: chr [1:260] "Aarti Mathur" "Aarti Mathur" "Aarti Mathur" "Aarti Mathur" ...
##  $ ID   : chr [1:260] "844" "716" "659" "613" ...
##  $ Score: num [1:260] 19 23 19 22 21 23 27 21 9 16 ...
```

Linear Mixed Model

The following code will do following things:
1) Run LME with random effects.
Dependent variable: Score
Fixed effect: ID (Abstract)
Random effect: Judges
2) Add column for LMER-Adjusted Score and Z-score (based on ID and Judge)
3) Average the Adjusted Scores for each ID and generate a 2-column list with ID, LMER-Adjusted, and Z-Scores

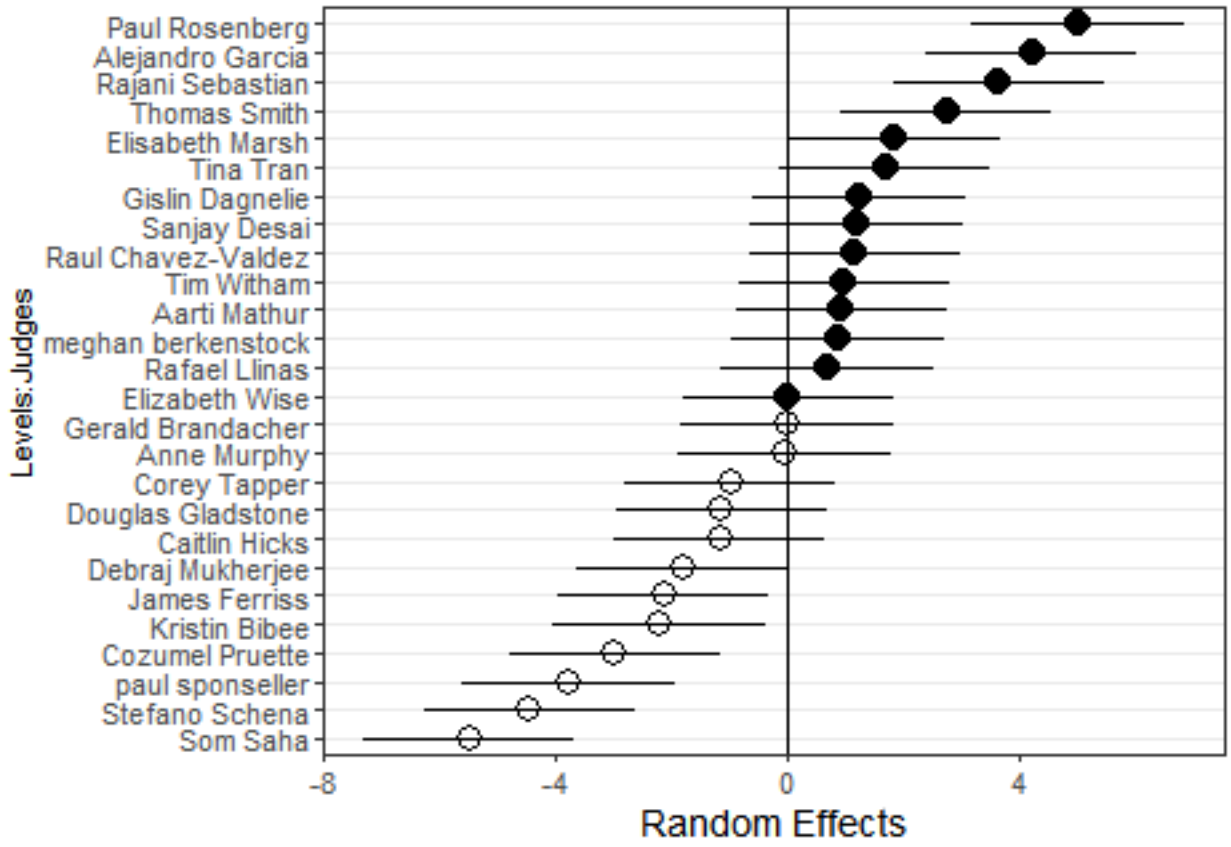
```
# We have a data-frame adjusted_data with variables Judge,
# ID, and Score ID has values of 3-digit ID numbers Judge has
# values by names Score has values from 4 (all 1's) to 28
# (all 7's)

##### Fit a linear-mixed-model with random effects for the judges

# LMER model generated
lmer_model <- lmer(Score ~ factor(ID) + (1 | Judge), data = adjusted_data)
# lme_model2 <-lme(behaviour ~ task*sex, random = ~
# 1|ID/task, method='ML', data=dat)

summary(lmer_model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Score ~ factor(ID) + (1 | Judge)
##    Data: adjusted_data
##
## REML criterion at convergence: 1165.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.54294 -0.50090 -0.05115  0.54049  2.09592
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Judge    (Intercept) 7.664    2.768
##  Residual             9.304    3.050
## Number of obs: 260, groups:  Judge, 26
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)   19.50317    1.52786  12.765
## factor(ID)120 -1.64153    2.16376  -0.759
## factor(ID)134 -1.02440    2.13482  -0.480
## factor(ID)136 -0.22127    2.04763  -0.108
## factor(ID)139 -2.56797    2.13874  -1.201
## factor(ID)162  3.67746    2.01604   1.824
## factor(ID)168 -4.91121    2.16286  -2.271
## factor(ID)178  3.25422    2.01662   1.614
## factor(ID)182 -1.02349    2.13748  -0.479
## factor(ID)184 -3.78813    2.08479  -1.817
## factor(ID)188 -2.31147    2.16285  -1.069
```

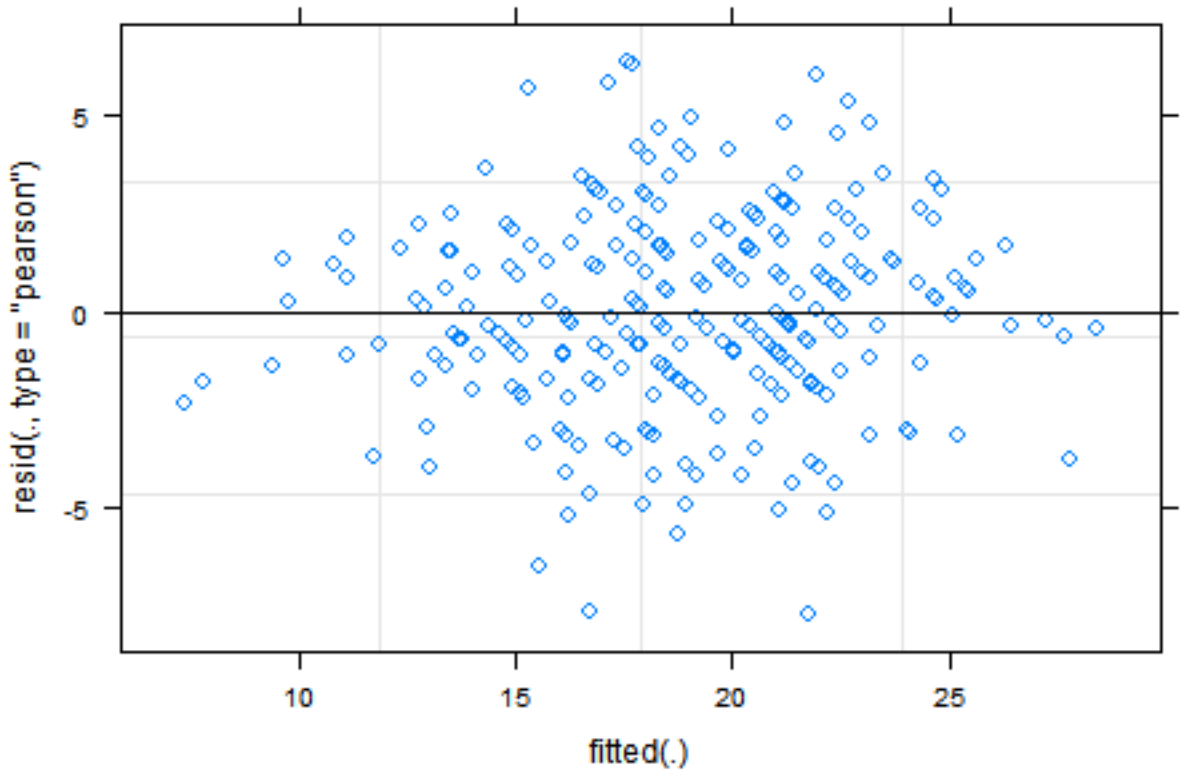
```
## factor(ID)194  0.20682    2.10915   0.098
## factor(ID)197  0.42295    2.01680   0.210
## factor(ID)210 -6.23563    2.11289  -2.951
## factor(ID)236  2.02183    1.99247   1.015
## factor(ID)261 -4.61774    2.02254  -2.283
## factor(ID)287  0.91462    2.02033   0.453
## factor(ID)313 -0.27759    2.14008  -0.130
## factor(ID)347 -1.45831    2.01321  -0.724
## factor(ID)409  1.87068    1.99414   0.938
## factor(ID)415  1.15214    2.16973   0.531
## factor(ID)428 -2.22867    1.99302  -1.118
## factor(ID)438 -1.56444    2.01811  -0.775
## factor(ID)443 -4.64279    2.04076  -2.275
## factor(ID)447 -0.53921    2.03826  -0.265
## factor(ID)482 -2.17994    2.13309  -1.022
## factor(ID)520 -2.97046    1.99634  -1.488
## factor(ID)546 -1.62100    2.15921  -0.751
## factor(ID)565 -0.28506    2.16897  -0.131
## factor(ID)572 -0.07541    2.04462  -0.037
## factor(ID)611 -2.14692    2.05098  -1.047
## factor(ID)613  0.71298    2.01691   0.354
## factor(ID)620 -0.48716    2.01835  -0.241
## factor(ID)645 -5.27336    2.01864  -2.612
## factor(ID)651  1.49000    2.01413   0.740
## factor(ID)655  0.33384    2.02581   0.165
## factor(ID)659  0.71867    2.01700   0.356
## factor(ID)668  2.98127    2.16844   1.375
## factor(ID)697 -7.72227    1.99159  -3.877
## factor(ID)707  2.01551    2.01812   0.999
## factor(ID)713 -5.40154    2.16503  -2.495
## factor(ID)716 -3.29441    2.02002  -1.631
## factor(ID)726  0.97598    1.99345   0.490
## factor(ID)728 -2.15824    2.16983  -0.995
## factor(ID)776  1.80458    2.04402   0.883
## factor(ID)784  0.58952    2.17415   0.271
## factor(ID)818 -1.93677    1.99728  -0.970
## factor(ID)828  3.51942    2.03727   1.728
## factor(ID)841  2.87514    2.13447   1.347
## factor(ID)844  0.47357    2.17205   0.218
## factor(ID)860 -0.63210    2.13666  -0.296
## factor(ID)911 -3.35984    2.16033  -1.555
## factor(ID)934  7.12302    2.16726   3.287
## factor(ID)937  1.64949    2.16728   0.761
## factor(ID)940 -2.31429    1.99200  -1.162
## factor(ID)972  3.92859    2.04784   1.918
## factor(ID)995  0.95208    2.16418   0.440
```

```
# Add adjusted LMER scores corresponding to each actual score
# as a column, removing the random effect.
adjusted_data$Score_LMER <- predict(lmer_model, re.form = NA,
    data = adjusted_data)
```

```r
###### Make caterpillar plot of conditional SD of random variables
###### (Judge)
randoms <- ranef(lmer_model, postVar = TRUE)
qq <- attr(ranef(lmer_model, postVar = TRUE)[[1]], "postVar")
rand.interc <- randoms$Judge
df <- data.frame(Intercepts = randoms$Judge[, 1], sd.interc = 2 *
    sqrt(qq[, , 1:length(qq)]), lev.names = rownames(rand.interc))
df$lev.names <- factor(df$lev.names, levels = df$lev.names[order(df$Intercepts)])
p <- ggplot(df, aes(lev.names, Intercepts, shape = lev.names))

# Added horizontal line at y=0, error bars to points and
# points with size two
p <- p + geom_hline(yintercept = 0) + geom_errorbar(aes(ymin = Intercepts -
    sd.interc, ymax = Intercepts + sd.interc), width = 0, color = "black") +
    geom_point(aes(size = 2))

# Removed legends and with scale_shape_manual point shapes
# set to 1 and 16
p <- p + guides(size = FALSE, shape = FALSE) + scale_shape_manual(values = c(rep(x = 1,
    nrow(df[df$Intercepts < 0, ])), rep(x = 16, nrow(df[df$Intercepts >
    0, ]))))

# Changed appearance of plot (black and white theme) and x
# and y axis labels
p <- p + theme_bw() + xlab("Levels:Judges") + ylab("Random Effects")

# Final adjustments of plot
p <- p + theme(axis.text.x = element_text(size = rel(1.2)), axis.title.x = element_text(size = rel(1.3)
    axis.text.y = element_text(size = rel(1.2)), panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank())

# To put levels on y axis you need to use coord_flip()
p <- p + coord_flip()
print(p)
```




```
###### Make residual plot
plot(lmer_model)
```




```
##### Add column of z-scores for each student based on judge
##### average and SD
adjusted_data <- setDT(adjusted_data)[, `:=`(Z.score, scale(Score)),
    Judge]

### Create a file that has the adjusted scores and raw scores.
write.xlsx(adjusted_data, file = paste(SC_category, "_", "raw_adjusted.xlsx",
    sep = ""), append = FALSE, sep = "\t", dec = ".", row.names = FALSE,
    col.names = TRUE, quote = FALSE)

### Make a ranked table grouped by IDs and corresponding
### average adjusted and raw scores
RANKED_DATA <- adjusted_data %>% group_by(ID) %>% summarise(Avg_Score_LMER = mean(Score_LMER),
    Avg_Z_score = mean(Z.score), Avg_raw_score = mean(Score)) %>%
    ungroup() %>% arrange(desc(Avg_Score_LMER)) %>% as.data.frame()

write.xlsx(RANKED_DATA, file = paste(SC_category, "_", "means_adjusted.xlsx",
    sep = ""), append = FALSE, sep = "\t", dec = ".", row.names = FALSE,
    col.names = TRUE, quote = FALSE)
```