R Notebook: Kuppu Statistical learning using multilevel regression discontinuity analysis

07/08/2017

```r
# required packages

packages <- c("optimx", "lme4", "ggplot2", "rmarkdown", "doBy",
    "RColorBrewer")
if (length(setdiff(packages, rownames(installed.packages()))) >
    0) {
    install.packages(setdiff(packages, rownames(installed.packages())),
        repos = "https://www.stats.bris.ac.uk/R/")
}

library(optimx)
library(lme4)
library(ggplot2)
library(doBy)
library(RColorBrewer)
```

```r
library(doBy)
library(knitr)
namelist = c("002", "003", "004", "005", "006", "007")  #,'008','009' )
nsubs = length(namelist)

main.data <- data.frame(ID = factor(), SetInd = integer(), Type = integer(),
    TargetRT = integer())

for (i in 1:nsubs) {

    myname = namelist[i]
    mycsv = paste0("C:/Users/pthompson/Dropbox/SL and WM paper/Revision/Pilot_Data_Revised task/Version
        myname, ".csv")
    mydata = read.csv(mycsv)  # read csv file
    # get rid of RTs of inaacurate responses and any RT greater
    # than 3000 ms:replace with NA.
    Rwdata = mydata
    rawdata = Rwdata[, c(1, 10, 12, 26)]
    # in the original analysis the following section will be
    # removed

    ####### this bit ensures inaccurate becomes, so not eliminated in
    ####### next section, and at least RT can be extracted.
    rawdata$TargetRT[rawdata$Type == 19 & rawdata$TargetRT >
        3000] <- 2999
    ###############
    rawdata$TargetRT[rawdata$TargetACC == 0] <- NA
    rawdata$TargetRT[rawdata$TargetRT < 100] <- NA
    rawdata$TargetRT[rawdata$TargetRT > 3000] <- NA

    RWdata <- rawdata
    # rename the types so median can be taken for each block
    RWdata$Type[RWdata$Type == 1] <- "Adj_D"
    RWdata$Type[RWdata$Type == 2] <- "Adj_D"
    RWdata$Type[RWdata$Type == 3] <- "Adj_D"
    RWdata$Type[RWdata$Type == 4] <- "Adj_P"
    RWdata$Type[RWdata$Type == 5] <- "Adj_P"
    RWdata$Type[RWdata$Type == 6] <- "Adj_P"
    RWdata$Type[RWdata$Type == 7] <- "Adj_P"
    RWdata$Type[RWdata$Type == 8] <- "Adj_P"
    RWdata$Type[RWdata$Type == 9] <- "Adj_P"
    RWdata$Type[RWdata$Type == 10] <- "Non_D"
    RWdata$Type[RWdata$Type == 11] <- "Non_D"
    RWdata$Type[RWdata$Type == 12] <- "Non_D"
    RWdata$Type[RWdata$Type == 13] <- "Non_P"
    RWdata$Type[RWdata$Type == 14] <- "Non_P"
    RWdata$Type[RWdata$Type == 15] <- "Non_P"
    RWdata$Type[RWdata$Type == 16] <- "Non_P"
    RWdata$Type[RWdata$Type == 17] <- "Non_P"
    RWdata$Type[RWdata$Type == 18] <- "Non_P"
    RWdata$Type[RWdata$Type == 19] <- "rand"

    RWdata$ID <- substring(RWdata$ID, 7, 7)
    RWdata$Type <- as.factor(RWdata$Type)
    RWdata$Type <- factor(RWdata$Type, levels = c("rand", "Adj_D",
        "Adj_P", "Non_D", "Non_P"))

    detaildata <- summaryBy(TargetRT ~ SetInd + Type, data = RWdata,
        FUN = c(median), na.rm = TRUE)

    detaildata$ID <- rep(RWdata$ID[4], length(detaildata[, 1]))

    names(detaildata) <- c("SetInd", "Type", "TargetRT", "ID")

    main.data <- rbind(main.data, detaildata)

}



main.data$ID <- as.factor(main.data$ID)
```

model summary and fitted values for each individuals random slopes and intercepts.

```r
main.data2 <- main.data

main.data2$broke1 <- ifelse(main.data2$SetInd %in% c(1:24), 1,
    0)
main.data2$broke2 <- ifelse(main.data2$SetInd %in% c(33:40),
    1, 0)

bp1 = 25  #cutpoint 1
bp2 = 33  #cutpoint 2
#
b1 <- function(x, bp1) ifelse(x < bp1, bp1 - x, 0)
```

```
b2 <- function(x, bp1, bp2) ifelse(x >= bp1 & x < bp2, x - bp1,
    0)
b3 <- function(x, bp2) ifelse(x < bp2, 0, x - bp2)

mod1d <- lmer(TargetRT ~ Type + b1(SetInd, bp1) + b2(SetInd,
    bp1, bp2) + b3(SetInd, bp2) + Type * b1(SetInd, bp1) + Type *
    b2(SetInd, bp1, bp2) + Type * b3(SetInd, bp2) + (broke1 +
    broke2 + b1(SetInd, bp1) + b2(SetInd, bp1, bp2) + b3(SetInd,
    bp2) | ID), data = main.data2, REML = FALSE, control = lmerControl(optimizer = "optimx",
    calc.derivs = FALSE, optCtrl = list(method = "nlminb", starttests = FALSE,
        kkt = FALSE)))

summary(mod1d)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## TargetRT ~ Type + b1(SetInd, bp1) + b2(SetInd, bp1, bp2) + b3(SetInd,
##     bp2) + Type * b1(SetInd, bp1) + Type * b2(SetInd, bp1, bp2) +
##     Type * b3(SetInd, bp2) + (broke1 + broke2 + b1(SetInd, bp1) +
##     b2(SetInd, bp1, bp2) + b3(SetInd, bp2) | ID)
##    Data: main.data2
## Control:
## lmerControl(optimizer = "optimx", calc.derivs = FALSE, optCtrl = list(method = "nlminb",
##     starttests = FALSE, kkt = FALSE))
##
##      AIC      BIC   logLik deviance df.resid
##  15719.4  15933.2  -7817.7  15635.4     1158
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0592 -0.5253 -0.0701  0.4086 10.9012
##
## Random effects:
##  Groups   Name                 Variance Std.Dev. Corr
##  ID       (Intercept)           8835.57  93.998
##           broke1               16835.53 129.752  -0.93
##           broke2                3636.42  60.303  -0.53  0.52
##           b1(SetInd, bp1)         33.68   5.803   0.86 -0.85 -0.06
##           b2(SetInd, bp1, bp2)   102.69  10.133  -0.84  0.98  0.46 -0.81
##           b3(SetInd, bp2)        249.82  15.806  -0.10  0.21 -0.73 -0.59
##  Residual                      25768.34 160.525
##
##
##
##
##
##
##   0.27
##
## Number of obs: 1200, groups:  ID, 6
##
## Fixed effects:
##                                   Estimate Std. Error t value
## (Intercept)                       771.2504    24.7863  31.116
## TypeAdj_D                         -106.8153    28.6936  -3.723
## TypeAdj_P                         -180.5468    28.6936  -6.292
## TypeNon_D                          -73.6682    28.6936  -2.567
## TypeNon_P                          -93.0794    28.6936  -3.244
## b1(SetInd, bp1)                      8.6978     1.7649   4.928
## b2(SetInd, bp1, bp2)                -0.5272     7.0476  -0.075
## b3(SetInd, bp2)                     46.8470     7.3926   6.337
## TypeAdj_D:b1(SetInd, bp1)           -6.3753     2.1998  -2.898
## TypeAdj_P:b1(SetInd, bp1)           -4.1683     2.1998  -1.895
## TypeNon_D:b1(SetInd, bp1)           -7.2076     2.1998  -3.276
## TypeNon_P:b1(SetInd, bp1)           -6.7768     2.1998  -3.081
## TypeAdj_D:b2(SetInd, bp1, bp2)      18.8845     9.7101   1.945
## TypeAdj_P:b2(SetInd, bp1, bp2)      32.8135     9.7101   3.379
## TypeNon_D:b2(SetInd, bp1, bp2)      13.1813     9.7101   1.357
## TypeNon_P:b2(SetInd, bp1, bp2)      19.0629     9.7101   1.963
## TypeAdj_D:b3(SetInd, bp2)          -63.3834     9.7101  -6.528
## TypeAdj_P:b3(SetInd, bp2)          -52.6811     9.7101  -5.425
## TypeNon_D:b3(SetInd, bp2)          -56.2378     9.7101  -5.792
## TypeNon_P:b3(SetInd, bp2)          -59.0847     9.7101  -6.085
```

```
##
## Correlation matrix not shown by default, as p = 20 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

```
## convergence code: 1
```

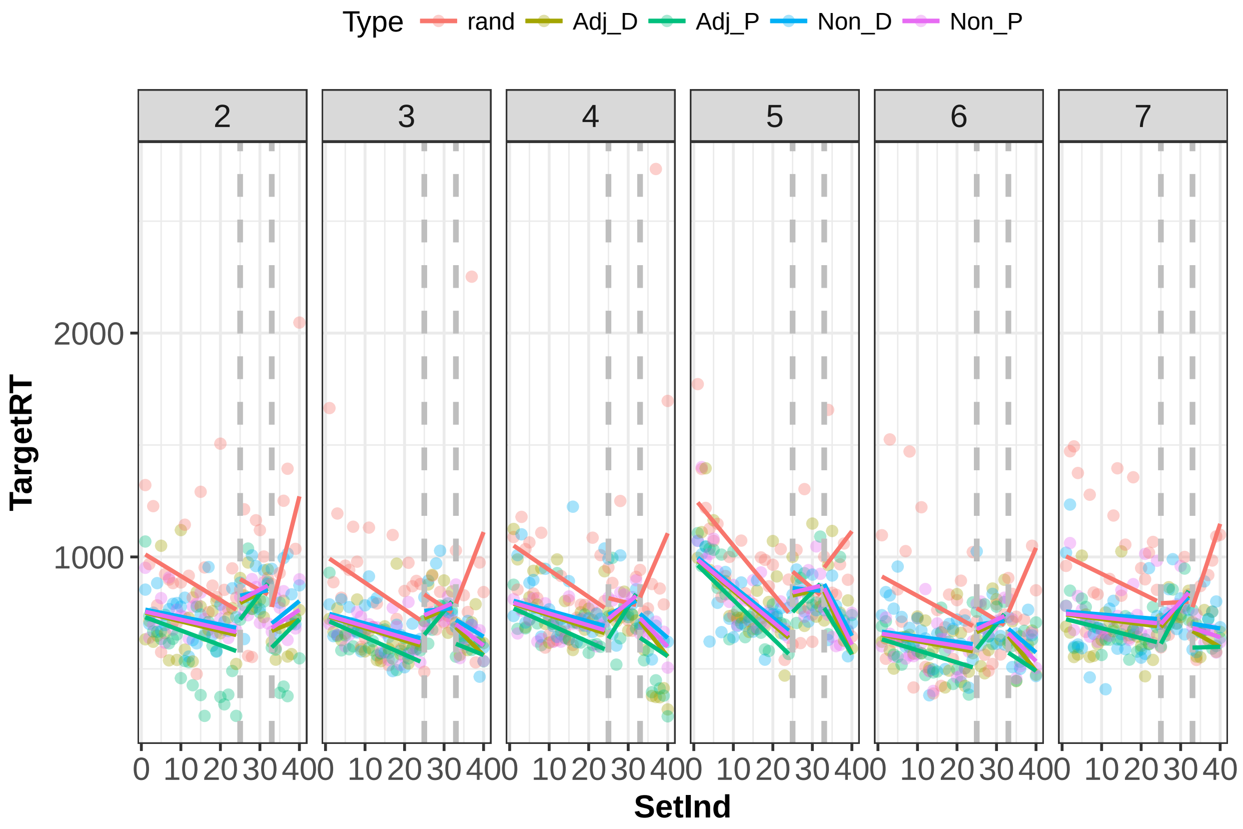
```
ranef(mod1d)
```

```
## $ID
##   (Intercept)      broke1      broke2 b1(SetInd, bp1) b2(SetInd, bp1, bp2)
## 2 129.5186307 -147.054236 -124.474730      2.03266923            -9.251541
## 3  61.5638613 -127.350615  -40.411750      3.26020526           -11.059302
## 4  44.9695753  -55.269444    3.353854      3.37772877            -3.871134
## 5 163.2746669 -204.226605   18.163949     12.67486140           -14.372802
## 6  -0.5825178  -88.666540  -18.006576      0.91326746           -10.169496
## 7  21.8924812    2.073479  -17.668876     -0.05780105             1.523762
##   b3(SetInd, bp2)
## 2       23.764165
## 3       -1.313047
## 4       -5.970302
## 5      -23.687325
## 6       -5.726130
## 7        6.367202
```

Plot the random slopes for individual participants by condition,

```
newdat1d <- expand.grid(Type = unique(main.data$Type), SetInd = 1:24,
    ID = unique(main.data$ID))
newdat2d <- expand.grid(Type = unique(main.data$Type), SetInd = 25:32,
    ID = unique(main.data$ID))
newdat3d <- expand.grid(Type = unique(main.data$Type), SetInd = 33:40,
    ID = unique(main.data$ID))

newdat1d$broke1 <- ifelse(newdat1d$SetInd %in% c(1:24), 1, 0)
newdat1d$broke2 <- ifelse(newdat1d$SetInd %in% c(33:40), 1, 0)
```

```
newdat2d$broke1 <- ifelse(newdat2d$SetInd %in% c(1:24), 1, 0)
newdat2d$broke2 <- ifelse(newdat2d$SetInd %in% c(33:40), 1, 0)

newdat3d$broke1 <- ifelse(newdat3d$SetInd %in% c(1:24), 1, 0)
newdat3d$broke2 <- ifelse(newdat3d$SetInd %in% c(33:40), 1, 0)



ggplot(main.data, aes(x = SetInd, y = TargetRT, color = Type)) +
    geom_point(alpha = 0.35) + geom_vline(aes(xintercept = 25),
    color = "grey", size = 1, linetype = "dashed") + geom_vline(aes(xintercept = 33),
    color = "grey", size = 1, linetype = "dashed") + geom_line(data = newdat1d,
    aes(y = predict(mod1d, newdata = newdat1d)), size = 0.75) +
    geom_line(data = newdat2d, aes(y = predict(mod1d, newdata = newdat2d)),
        size = 0.75) + geom_line(data = newdat3d, aes(y = predict(mod1d,
    newdata = newdat3d)), size = 0.75) + theme_bw() + facet_grid(~ID) +
    scale_fill_brewer(palette = "Set1") + theme(legend.position = "top",
    strip.text = element_text(size = 12), axis.text = element_text(size = 12),
    axis.title = element_text(size = 12, face = "bold"))
```


Type
rand
Adj_D
Adj_P
Non_D
Non_P
TargetRT
SetInd