Multilevel Modeling of Arbellay et al 2017

Song S. Qian

October 14, 2017

Front Matters

Function for loading/installing packages:

```
> ## load/install packages
> packages<-function(x, repos="http://cran.r-project.org", ...){
+   x<-as.character(match.call()[[2]])
+   if (!require(x,character.only=TRUE)){
+     install.packages(pkgs=x, repos=repos, ...)
+     require(x,character.only=TRUE)
+   }
+ }
```

Set up working directory:

```
base <- getwd()
dataDIR <- paste(base, "data", sep="/")
## put your data set in the Data subdirectory
plotDIR <- paste(base, "Figs", sep="/")
## put created figures in the Figures subdirectory
setwd(base)
```

Load needed packages:

```
packages(lattice)
packages(arm)
packages(reshape2)
packages(tikzDevice)
```

Reading and Processing Data

```
tp282 <- read.csv(paste(dataDIR, "ArbellayAll.csv", sep = "/"))
## removing log-values
tp282 <- tp282[, -grep("(LOG)$", names(tp282))]
## creating events
tp282$Events <- cut(tp282$YEAR, breaks = c(1885, 1890, 1909, 1939, 1946, 1955,
    1967), label = 1:6)
tp282$trt <- 2
tp282$trt[is.element(tp282$YEAR, c(1886, 1887, 1906, 1907, 1935, 1936, 1943,
    1944, 1952, 1953, 1961, 1962))] <- 1
tp_melt <- melt(tp282, id.vars = c("SAMPLE", "YEAR", "Events", "trt"))
```

Multilevel Model

Details of the model are in Qian (2016) [Environmental and Ecological Statistics with R (2nd Ed.), Chapman and Hall/CRC Press], Specifically, Section 10.3 (Multilevel ANOVA) and graphics code used here are similar to the code in Section 10.6 (Multilevel GLM).

```
tp_lmer1 <- lmer(log(value) ~ 1 + (1 | Events) + (1 | trt) + (1 | variable),
    data = tp_melt)
summary(tp_lmer1)  ## This model assumes additive effects for all three factors
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(value) ~ 1 + (1 | Events) + (1 | trt) + (1 | variable)
##    Data: tp_melt
##
## REML criterion at convergence: 5399.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.5695 -0.5725 -0.0281  0.5711  3.9998
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Events   (Intercept) 0.004404 0.06636
##  variable (Intercept) 6.911736 2.62902
##  trt      (Intercept) 0.262636 0.51248
##  Residual             0.377114 0.61410
## Number of obs: 2872, groups:  Events, 6; variable, 4; trt, 2
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   0.3275     1.3639    0.24
```

```
tp_melt$EvTp <- paste(tp_melt$Events, tp_melt$trt)
tp_lmer2 <- lmer(log(value) ~ 1 + (1 | EvTp) + (1 | variable), data = tp_melt)
summary(tp_lmer2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(value) ~ 1 + (1 | EvTp) + (1 | variable)
##    Data: tp_melt
##
## REML criterion at convergence: 5419.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.5404 -0.5847 -0.0388  0.5703  3.9981
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  EvTp     (Intercept) 0.1396   0.3736
##  variable (Intercept) 6.9118   2.6290
##  Residual             0.3755   0.6128
## Number of obs: 2872, groups:  EvTp, 12; variable, 4
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)   0.3274     1.3190   0.248
```

```
tp_melt$MeEvTp <- paste(tp_melt$variable, tp_melt$Events, tp_melt$trt)
tp_lmer3 <- lmer(log(value) ~ 1 + (1 | MeEvTp), data = tp_melt)
summary(tp_lmer3)  ## interactions among all three factors (implicitly)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(value) ~ 1 + (1 | MeEvTp)
##    Data: tp_melt
##
## REML criterion at convergence: 5579.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.5463 -0.5491 -0.0054  0.5030  4.0534
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  MeEvTp   (Intercept) 5.4097   2.3259
##  Residual             0.3656   0.6046
## Number of obs: 2872, groups:  MeEvTp, 48
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   0.3278     0.3359   0.976
```

Processing output

The three models all perform well. The estimated residual variances are all close to 0.36. Using the full model, we summarize the result graphically.

```
est <- as.data.frame(matrix(unlist(strsplit(row.names(coef(tp_lmer3)[[1]]),
    split = " ")), ncol = 3, byrow = T))
names(est) <- c("method", "events", "treatment")
est$treatment <- factor(est$treatment, labels = c("after", "before"))
est$estimate <- coef(tp_lmer3)[[1]][, 1]
est$se <- sqrt(se.fixef(tp_lmer3)^2 + se.ranef(tp_lmer3)[[1]][, 1]^2)
est$se2 <- se.ranef(tp_lmer3)[[1]][, 1]
```

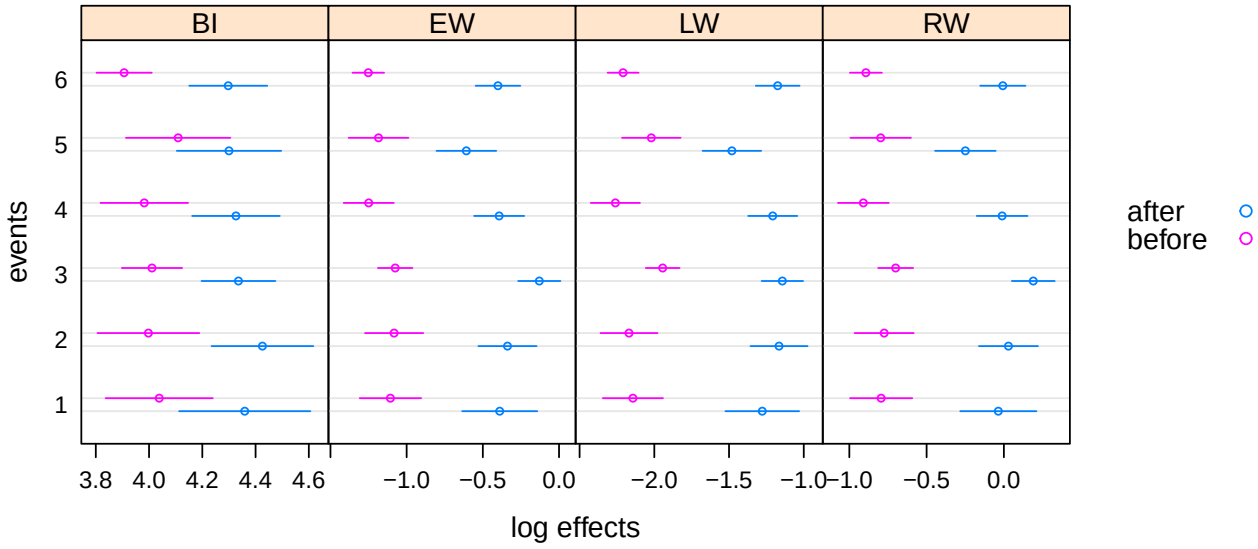
Plotting results

The function `my.panel` is written to add CIs to the estimated effects.

```
my.panel <- function(x, y, subscripts, group.number, col, pch, se, ...){
    myjitter <- c(-0.1,0.1)
    panel.dotplot(x,as.numeric(y)+myjitter[group.number],
                  cex=0.5, pch=pch, col=col)
##    panel.grid()
    panel.segments(est$estimate[subscripts]-est$se2[subscripts]*2,
                   as.numeric(y)+myjitter[group.number],
                   est$estimate[subscripts]+est$se2[subscripts]*2,
                   as.numeric(y)+myjitter[group.number],
                   col=col,
```

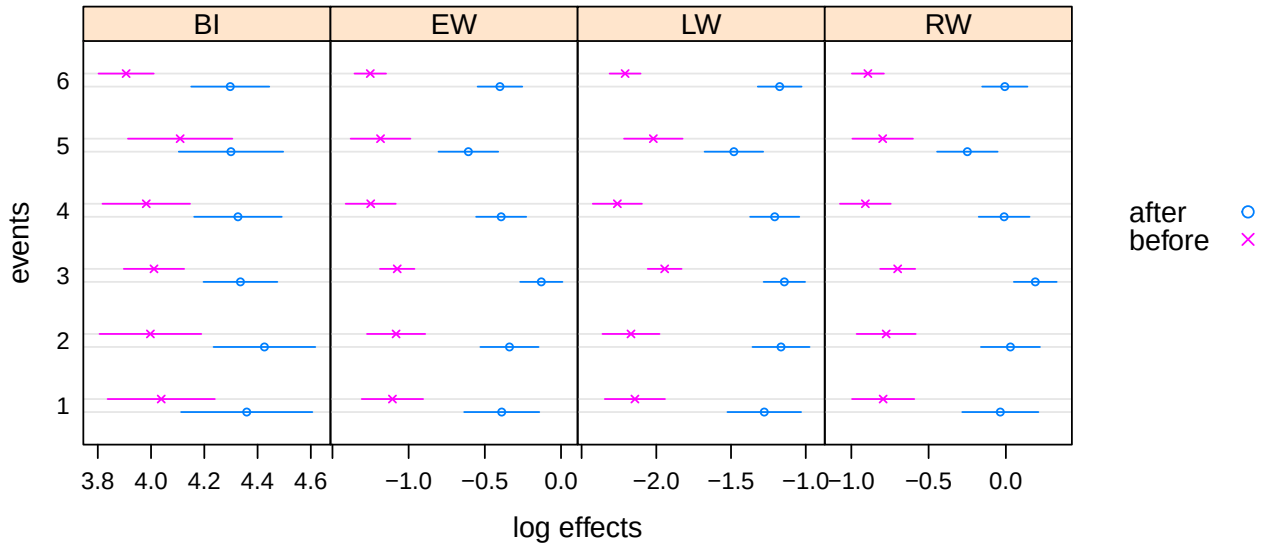
```
                    pch=pch)
}
est_low<-tapply(est$estimate-2*est$se2, est$method, min)
est_up <- tapply(est$estimate+2*est$se2, est$method, max)
est_range <- rbind(est_low, est_up)

##trellis.device(pdf, file=paste(plotDIR, "compareCL.pdf", sep="/"), height=3.5, width=7.5, color = T)
key <- simpleKey(levels(est$treatment), space="right")
dotplot(events~estimate|method, data=est,
        key=key,
        groups=treatment,
        col=key$points$col,
        pch=key$points$pch,
        layout=c(4,1),
        scales=list(x=list(relation="free")),
        panel=my.panel, xlab="log effects", ylab="events",
##        scales=list(x=list(alternating=T)),
        xlim=list(est_range[,1], est_range[,2], est_range[,3], est_range[,4]))
```




```
##dev.off()
##trellis.device(pdf, file=paste(plotDIR, "compareBW.pdf", sep="/"), height=3.5, width=7.5, color=F)

key <- simpleKey(levels(est$treatment), space="right")
key$points$pch=c(1,4)
dotplot(events~estimate|method, data=est,
        key=key,
        groups=treatment,
        col=key$points$col,
        pch=key$points$pch,
        layout=c(4,1),
        scales=list(x=list(relation="free")),
        panel=my.panel, xlab="log effects", ylab="events",
##        scales=list(x=list(alternating=T)),
        xlim=list(est_range[,1], est_range[,2], est_range[,3], est_range[,4]))
```


BI
EW
LW
RW
events
6
5
4
3
2
1
3.8
4.0
4.2
4.4
4.6
−1.0
−0.5
0.0
−2.0
−1.5
−1.0
−1.0
−0.5
0.0
log effects
after
before

```
##dev.off()
```