

Appendix of tutorial analyzing cyclic models

Peter Verboon and Roeslan Leontjevas

2018-10-11

Preliminary Steps

There are three functions used in this appendix, which are in the R package `cyclic`. This package can be installed from Github by running the following command in R:

```
devtools::install_github("PeterVerboon/cyclicpkg/cyclic")
```

The package is then activated by running:

```
require(cyclic).
```

The package consists of three functions. The function `fitCyclic()` fits a linear model with `lm()` to obtain cyclic parameters. The function `fitCyclicMLA()` fits a MLA model with `lmer()` from the `lme4` package to obtain cyclic parameters. The function `cycpar()` is an auxiliary function, which computes the cyclic parameters from the linear model parameters.

These functions should be helpful for applying a cyclic model to your data. The functions have methods for printing the results and plotting the fitted cyclic patterns.

As an illustration of a cyclic process we generate perfect cyclic data and construct Figure 1 with these data using the following code. The package `ggplot2` (loaded with “`cyclic`”) is used for plotting.

```
a0 <- 1;
b1 <- 2;
b2 <- 10;
P <- 24;

T <- seq(0, 24, by =0.1);
y <- a0 + b1*cos(2*(pi/P)*(T - b2))

g <- ggplot() + geom_line(aes(x=T, y=y)) ;
g <- g + labs(x = "Time points", y = "Dependent variable ");
g <- g + theme(axis.text = element_text(size = 10, colour="black"));
g <- g + scale_x_continuous(breaks = seq(0,24,2)) ;
g <- g + geom_line(aes(x=b2, y=seq(1,3,0.1)), lty="dashed") ;
g <- g + geom_line(aes(x=seq(0,24,0.1), y=a0)) ;
g <- g + annotate("text", x = 1, y = 1.2, label = "b0");
g <- g + annotate("text", x = 10.8, y = 2, label = "b1");
g <- g + annotate("text", x = 10, y = 0.8, label = "b2");
g
```

The “smoking” data

The “smoking” data are in the object `smokingdat.rda`. This is a dataframe, which contains a selection of variables from a larger data file (for more information about the data, see Bolman et al., 2018). Check the names in the dataframe and then put the data in `dat1`, the first data set used in the analysis.

```
data("smokedat")
names(smokedat)

## [1] "subjnr"      "beepnr"      "daynr"      "positiveAffect"
## [5] "intention"   "stress"

dat1 <- smokedat
nall <- length(unique(dat1$subjnr))
```

Count the number of records per subject and remove subjects with less than 50 records. The resulting data.frame is called dat3.

```
dat1$count <- 1
dat2 <- aggregate(dat1[,c("count")],by=list(dat1$subjnr), FUN=sum, na.rm=T);
dat2$subjnr <- dat2$Group.1
dat1$count <- NULL;
dat2$count <- dat2[,2]

dat3 <- merge(dat1,dat2[,c("subjnr","count")], by.x = "subjnr")
rm(dat2)

dat3 <- subset(dat3, dat3$count >= 50)
nfew <- nall - length(unique(dat3$subjnr))
```

We also compute variability of intention per subject and remove subjects with very small variation in intention. The result is also in dat3.

```
dat2 <- aggregate(dat3$intention,by=list(dat3$subjnr), FUN=sd, na.rm=F)
dat2$subjnr <- dat2$Group.1
dat2$Group.1 <- NULL
dat2 <- merge(dat3, dat2, by.x = "subjnr")
dat3 <- subset(dat2, dat2$V1 > .10)
rm(dat2)
nunique <- length(unique(dat3$subjnr))
nsmall <- (nall - nfew) - nunique
```

The result of these steps are summarized below.

```
## Number of subjects in data set:          49
## Number of subjects with less than 50 records: 8
## Number of subjects with SD smaller than .10: 3
## Number of subjects used in analysis:      38
```

The next step is to construct a dataset that is aggregated over subjects, which is called dat4. This dataframe will be used in subsequent analyses.

```
dat4 <- aggregate(dat3[,c("positiveAffect","stress","intention")],
                  by=list(dat3$beepnr,dat3$daynr), FUN=mean, na.rm=F);
dat4$daynr <- dat4$Group.2
dat4$beepnr <- dat4$Group.1
dat4$Group.1 <- NULL
dat4$Group.2 <- NULL
```

Single-level Analyses

Step 1

The raw data for three subjects and average are plotted. We have to repeat step 1 for the aggregated data in `dat4`, and for subjects 2, 15 and 18. This makes Figure 2 from the tutorial.

```
pdat <- dat4 # averaged over all subjects
pdat <- subset(dat3, dat3$subjnr == 2) # ppn 2, 15, 18 selected for Figure 1

npoints <- dim(pdat)[1]
x <- c(1:npoints)
pdat$day <- as.factor(pdat$daynr)

g <- ggplot(pdat, aes(x=x, y=pdat$intention, colour=pdat$day)) + geom_point() +
  scale_x_discrete(name = "Time points (beeps within days)", labels=pdat$beepnr, limits=c(1:npoints)) +
  theme(axis.text = element_text(size = 6, colour="black"), legend.position="none")
g
```

Step 2

Now we analyze the cyclic model and plot the result, again for the average and for subjects 2, 15, and 18. This is Figure 3 from the tutorial.

```
model_a <- fitCyclic(pdat, yvar = "intention", xvar = "beepnr", grp = "daynr",
  ymin = -2.5, ymax = 1.5, step=.30)
print(model_a)
plot(model_a, type = c("raw", "means"))
```

Step 3

Fit an extra term in the model with day as covariate in aggregated data. Show the parameters and the fit of the model.

```
model_b <- fitCyclic(dat4, yvar = "intention", xvar = "beepnr", grp = "daynr",
  cov = "daynr", ymin=-0.5, ymax=0.5)
print(model_b)
```

Step 4

Apply the cyclic model with daily period for the variables stress and positive affect for subject 15 to obtain a plot of the raw data with the fitted cycles and a plot of the aggregated data. Together with intention (model_a) this makes Figure 4.

```
pdat <- subset(dat3, dat3$subjnr == 15)

model_c <- fitCyclic(pdat, yvar = "stress", xvar = "beepnr", grp = "daynr",
  ymin = -1.0, ymax = 0.5, step= 0.25)
plot(model_c, type = c("raw", "means"))
model_c <- fitCyclic(pdat, yvar = "positiveAffect", xvar = "beepnr", grp = "daynr",
```

```

                                ymin = -1.0, ymax = 0.5, step= 0.25)
plot(model_c, type = c("raw", "means"))

```

Step 5

Apply, only for subject 15, the cyclic model with a weekly period for intention, stress and positive affect, instead of a daily period. Below the code is given which produces the six separate plots of Figure 5.

```

pdat <- subset(dat3, dat3$subjnr == 15)

model_d <- fitCyclic(pdat, yvar = "intention", xvar = "daynr", grp = "daynr",
                    ymin = -2.0, ymax = 1.0, step= 0.25)

plot(model_d, type = c("raw", "means"))
model_d <- fitCyclic(pdat, yvar = "stress", xvar = "daynr", grp = "daynr",
                    ymin = -2.0, ymax = 1.0, step= 0.25)

plot(model_d, type = c("raw", "means"))
model_d <- fitCyclic(pdat, yvar = "positiveAffect", xvar = "daynr", grp = "daynr",
                    ymin = -2.0, ymax = 1.0, step= 0.25)
plot(model_d, type = c("raw", "means"))

```

Multilevel Analyses

Step 6

Analyze the data with the cyclic model using MLA and model comparison. Each consecutive model is more complex. The intraclass correlation is computed with the null model (model1). This is the first model that is fitted on the data.

```

model1 <- fitCyclicMLA(dat=dat3, yvar="intention", id = "subjnr")
print(model1)

```

The next two models fit a daily cycle to the data. Model2 with only the intercept as random term and model3 includes the cyclic terms as random effects.

```

model2 <- fitCyclicMLA(dat=dat3, yvar="intention", xvar1="beepnr", xvar2="daynr", id = "subjnr",
                      ncycle = 1, random = "intercept" )
print(model2)

model3 <- fitCyclicMLA(dat=dat3, yvar="intention", xvar1="beepnr", xvar2="daynr", id = "subjnr",
                      ncycle = 1, random = "first")
print(model3)
plot(model3)

```

Model 4 and 5 are fitting two cyclic process: patterns within days and patterns within weeks. In model 5 every effect is assumed to be random, whereas in model 4 only the within days cyclic terms are assumed random. Plotting model 5 gives a plot as is shown in Figure 6.

```

model4 <- fitCyclicMLA(dat=dat3, yvar="intention", xvar1="beepnr", xvar2="daynr", id = "subjnr",
                      ncycle = 2, random = "first")
print(model4)

```

```

model5 <- fitCyclicMLA(dat=dat3, yvar="intention", xvar1="beepnr", xvar2="daynr", id = "subjnr",
                      ncycle = 2, random = "all", ymin = -0.5, ymax = 0.5, step=0.10 )
print(model5)
plot(model5)

```

Model 6 also contains covariate. Finally all models are compared with each other, this is shown in table 1.

```

model6 <- fitCyclicMLA(dat=dat3, yvar="intention", xvar1="beepnr", xvar2="daynr", id = "subjnr",
                      cov = "stress", ncycle = 2, random = "all" )
print(model6)

anova(model6$fit, model5$fit, model4$fit, model3$fit, model2$fit, model1$fit)

```