hrv-rev1.R

Niall

Sat Mar 19 14:04:45 2016

```
#Psychophysiological Methods and Analysis: Module 2: Heart Rate Variability
#Version with Time 6 as separate phase
```

```
#Set working directory: replace with your directory path
setwd("C:/Users/Niall/Dropbox/NIALL1/MET/ppmc");

#Read in the hrv dataset prepared by Katherine; Note the use of na.omit function
hrv <- na.omit(read.csv("ISP_HR_partialdata_long.csv"))

#Source Tal Galili's improved version of merge function
source("https://raw.githubusercontent.com/talgalili/R-code-snippets/master/merge.data.frame.r")
```

```
library(car) #For Recode function
library(lme4) #For multilevel modeling
```

```
## Loading required package: Matrix
```

```
library(lmerTest) #To get Satterthwaite df and p values from lmer
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
library(pbkrtest) #To get K-R df and p values from lmer

#Create phase6, a version of phase with time6 as a separate phase
hrv <- within(hrv, {
  phase6 <- Recode(time,
                    ' 1:5 = "baseline"; 6 = "time6"; 7:10 = "practice"; 11:13 = "recovery" ',
                   as.factor.result=TRUE)
}
)

#Make the factor phase6 have levels in the right order,
#i.e., with "baseline" as the reference category
hrv$phase6 <- with(hrv, factor(phase6, levels=c('baseline','time6','practice','recovery')))
```

```
#Sort the hrv dataset by the size of the change in RSA between time 5 and 6

#Create variable that is the difference in RSA between time 5 and 6
# using subset function

# First, subset RSA for time=5 (1 dataline per subj)
time5 <- subset(hrv, time==5, select=c(subj, time, RSA))
names(time5) <- c("subj", "time5", "RSA5")

#Next, subset time=5
time6 <- subset(hrv, time == 6, select=c(subj, time, RSA))
names(time6) <- c("subj", "time6", "RSA6")

#Bind the datasets by column, so there is still just 1 dataline per subj
time56 <- merge.data.frame(time5, time6, by = "subj", keep_order=1)
time56$difRSA <- time56$RSA6 - time56$RSA5

#Merge and order by RSA difference between time 5 and 6
hrv1 <- merge.data.frame(hrv, time56, by = "subj", keep_order=1)
ordhrv1 <- hrv1[order(hrv1$difRSA), ]
ordtime56 <- time56[order(time56$difRSA), ]
```

```
#Raw Data: Panel plots, sorted by RSA change between time 5 at time 6
```

```
pdf(file="hrv-drop-ordered-panel-time-course.pdf", width=14, height=28)
par(mfrow=c(11,7))
for (i in ordtime56$subj) {
  plot(hrv1$time[hrv1$subj==i], hrv1$RSA[hrv1$subj==i],  ylab="HR Variability", xlab="Time",
       type="l", ylim=c(0,10), xlim=c(0, 13))
  abline(v=c(5, 6, 10))
}
dev.off()
```

```
## pdf
##   2
```

```
#MODELS
```

```
#Model with random intercept and random slope
```

```
rsaout2p6<- lmer(RSA ~ Condition*phase6 + (1 + phase6 | subj), data = hrv1)
summary(rsaout2p6, ddf = "Kenward-Roger")
```

```
## Note: method with signature 'sparseMatrix#ANY' chosen for function 'kronecker',
##  target signature 'dgCMatrix#ngCMatrix'.
##  "ANY#sparseMatrix" would also be valid
```

```
## Linear mixed model fit by REML t-tests use Kenward-Roger approximations
##   to degrees of freedom [lmerMod]
## Formula: RSA ~ Condition * phase6 + (1 + phase6 | subj)
##    Data: hrv1
```

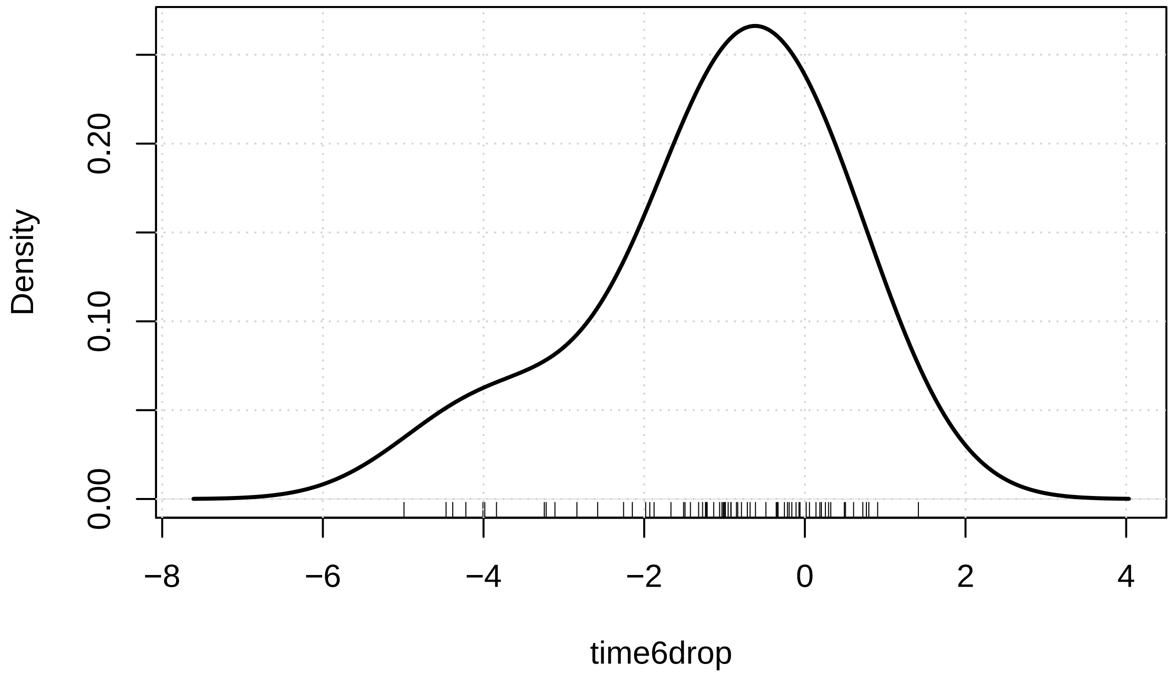
```
##
## REML criterion at convergence: 2111.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9427 -0.5177  0.0180  0.5133  4.1510
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  subj     (Intercept)    1.2483   1.1173
##           phase6time6    2.5149   1.5859   -0.45
##           phase6practice 0.8259   0.9088   -0.53  0.73
##           phase6recovery 0.4805   0.6932   -0.48  0.48  0.68
##  Residual                0.3354   0.5791
## Number of obs: 894, groups:  subj, 69
##
## Fixed effects:
##                                  Estimate Std. Error      df t value
## (Intercept)                       6.4317     0.1939 81.7300  33.177
## ConditionVisible                 -0.5276     0.2762 81.7300  -1.910
## phase6time6                      -1.4441     0.2887 74.5900  -5.002
## phase6practice                   -0.5823     0.1671 78.5400  -3.485
## phase6recovery                   -0.5126     0.1385 81.9200  -3.701
## ConditionVisible:phase6time6      0.6436     0.4113 74.5900   1.565
## ConditionVisible:phase6practice   0.3913     0.2380 78.5400   1.644
## ConditionVisible:phase6recovery   0.2099     0.1964 81.5900   1.069
##                                  Pr(>|t|)
## (Intercept)                       < 2e-16 ***
## ConditionVisible                 0.059599 .
## phase6time6                      3.66e-06 ***
## phase6practice                   0.000807 ***
## phase6recovery                   0.000388 ***
## ConditionVisible:phase6time6     0.121847
## ConditionVisible:phase6practice 0.104115
## ConditionVisible:phase6recovery 0.288305
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                 (Intr) CndtnV phs6t6 phs6pr phs6rc CnV:66 CndtnVsbl:phs6p
## ConditnVsbl     -0.702
## phase6time6     -0.441  0.309
## phase6prctc     -0.534  0.375  0.664
## phase6rcvry     -0.466  0.327  0.426  0.609
## CndtnVsb:66      0.309 -0.441 -0.702 -0.466 -0.299
## CndtnVsbl:phs6p  0.375 -0.534 -0.466 -0.702 -0.427  0.664
## CndtnVsbl:phs6r  0.328 -0.468 -0.301 -0.429 -0.705  0.428  0.611
```

```
anova(rsaout2p6, type = 3, ddf = "Kenward-Roger")
```

```
## Analysis of Variance Table of type III  with  Kenward-Roger
## approximation for degrees of freedom
##                   Sum Sq Mean Sq NumDF  DenDF F.value    Pr(>F)
## Condition         0.2933  0.2933     1 74.579  0.8745    0.3527
## phase6           11.9857  3.9952     3 74.206 11.9124 1.881e-06 ***
## Condition:phase6  1.0167  0.3389     3 74.206  1.0104    0.3930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Pull random effects of time6 drop; add to fixed effect of drop (averaged over Inv, Vis)
#Look at its distribution

time6drop<- - 1.114 + ranef(rsaout2p6)$subj[, 2]
densityPlot( ~ time6drop, bw="SJ", adjust=1.8, kernel="gaussian")
```



```
#Panel Plots HRV actual and HRV predicted vs. Time for subjects sorted by time 6 drop

pdf(file="hrv-panel-HRV-pred-vs-time-for-time6-drop.pdf", width=14, height=28)
par(mfrow=c(11,7))
for (i in ordtime56$subj) {
  plot(hrv1$time[hrv1$subj==i], hrv1$RSA[hrv1$subj==i],  ylab="HR Variability", xlab="Time",
       type="o", pch=19, col="blue", ylim=c(.8,10), xlim=c(0, 13),
       main=paste("id", i, "(", round(ordtime56$difRSA[ordtime56$subj==i], digits=1),")"))
  points(hrv1$time[hrv1$subj==i], predict(rsaout2p6)[hrv1$subj==i], pch="-", cex=3, col="red")
  abline(v=c(5, 6, 10))
}
dev.off()
```

```
## pdf
##   2
```

```
#Misspecified Model: Fixed intercept and slope
```

```
rsaout0p6<- lm(RSA ~ Condition*phase6, data = hrv1)
summary(rsaout0p6)
```

```
##
## Call:
## lm(formula = RSA ~ Condition * phase6, data = hrv1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1627 -0.7394 -0.0001  0.8519  3.4660
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       6.43172    0.09242  69.594  < 2e-16 ***
## ConditionVisible                 -0.52758    0.13166  -4.007 6.66e-05 ***
## phase6time6                      -1.44407    0.22637  -6.379 2.87e-10 ***
## phase6practice                   -0.58226    0.13863  -4.200 2.94e-05 ***
## phase6recovery                   -0.49344    0.15230  -3.240  0.00124 **
## ConditionVisible:phase6time6      0.64362    0.32249   1.996  0.04626 *
## ConditionVisible:phase6practice   0.39132    0.19748   1.982  0.04784 *
## ConditionVisible:phase6recovery   0.19077    0.21596   0.883  0.37729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.223 on 886 degrees of freedom
## Multiple R-squared:  0.07917,    Adjusted R-squared:  0.0719
## F-statistic: 10.88 on 7 and 886 DF,  p-value: 3.2e-13
```

```
Anova(rsaout0p6, type="3")
```

```
## Anova Table (Type III tests)
##
## Response: RSA
##                  Sum Sq  Df   F value    Pr(>F)
## (Intercept)      7239.2   1 4843.3893 < 2.2e-16 ***
## Condition          24.0   1   16.0582 6.659e-05 ***
## phase6             71.7   3   15.9877 3.955e-10 ***
## Condition:phase6    9.3   3    2.0805    0.1013
## Residuals        1324.3 886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Misspec (rasout0) Panel Plots HRV actual and HRV predicted vs. Time for subjects sorted by time 6 drop

pdf(file="rsaout0-hrv-panel-HRV-pred-vs-time-for-time6-drop.pdf", width=14, height=28)
par(mfrow=c(11,7))
for (i in ordtime56$subj) {
  plot(hrv1$time[hrv1$subj==i], hrv1$RSA[hrv1$subj==i],  ylab="HR Variability", xlab="Time",
       type="o", pch=19, col="blue", ylim=c(.8,10), xlim=c(0, 13),
       main=paste("id", i, "(", round(ordtime56$difRSA[ordtime56$subj==i], digits=1),")"))
```

```r
  points(hrv1$time[hrv1$subj==i], predict(rsaout0p6)[hrv1$subj==i], pch="-", cex=3, col="red")
  abline(v=c(5, 6, 10))
}
dev.off()
```

```
## pdf
##   2
```

```r
#Misspecified Model: Random intercept and fixed slope
```

```r
rsaout1p6<- lmer(RSA ~ Condition*phase6 + (1 | subj), data = hrv1)
summary(rsaout1p6, ddf = "Kenward-Roger")
```

```
## Linear mixed model fit by REML t-tests use Kenward-Roger approximations
##   to degrees of freedom [lmerMod]
## Formula: RSA ~ Condition * phase6 + (1 | subj)
##    Data: hrv1
##
## REML criterion at convergence: 2381.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4888 -0.5674  0.0521  0.6139  3.7652
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj     (Intercept) 0.8458   0.9197
##  Residual             0.6651   0.8155
## Number of obs: 894, groups:  subj, 69
##
## Fixed effects:
##                                  Estimate Std. Error        df t value
## (Intercept)                       6.43172    0.16723  79.70000  38.461
## ConditionVisible                 -0.52758    0.23823  79.70000  -2.215
## phase6time6                      -1.44407    0.15101 819.00000  -9.563
## phase6practice                   -0.58226    0.09247 819.00000  -6.297
## phase6recovery                   -0.51374    0.10185 819.60000  -5.044
## ConditionVisible:phase6time6      0.64362    0.21512 819.00000   2.992
## ConditionVisible:phase6practice   0.39132    0.13173 819.00000   2.971
## ConditionVisible:phase6recovery   0.21107    0.14424 819.30000   1.463
##                                  Pr(>|t|)
## (Intercept)                       < 2e-16 ***
## ConditionVisible                  0.02965 *
## phase6time6                       < 2e-16 ***
## phase6practice                   4.96e-10 ***
## phase6recovery                   5.61e-07 ***
## ConditionVisible:phase6time6      0.00286 **
## ConditionVisible:phase6practice   0.00306 **
## ConditionVisible:phase6recovery   0.14377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##                  (Intr) CndtnV phs6t6 phs6pr phs6rc CnV:66 CndtnVsbl:phs6p
## ConditnVsbl     -0.702
## phase6time6     -0.150  0.106
## phase6prctc     -0.246  0.173  0.272
## phase6rcvry     -0.223  0.157  0.247  0.404
## CndtnVsb:66      0.106 -0.150 -0.702 -0.191 -0.173
## CndtnVsbl:phs6p  0.173 -0.246 -0.191 -0.702 -0.283  0.272
## CndtnVsbl:phs6r  0.158 -0.224 -0.174 -0.285 -0.706  0.249  0.406
```

```
anova(rsaout1p6, type="3", ddf = "Kenward-Roger")
```

```
## Analysis of Variance Table of type III  with  Kenward-Roger
## approximation for degrees of freedom
##                   Sum Sq Mean Sq NumDF  DenDF F.value    Pr(>F)
## Condition          0.582  0.5820     1  70.48   0.875  0.352743
## phase6            82.134 27.3779     3 819.13  41.165 < 2.2e-16 ***
## Condition:phase6   9.297  3.0989     3 819.13   4.659  0.003089 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Misspec (rsaout1): Panel Plots HRV actual and HRV predicted vs. Time for subjects sorted by time 6 drop

pdf(file="rsaout1-hrv-panel-HRV-pred-vs-time-for-time6-drop.pdf", width=14, height=28)
par(mfrow=c(11,7))
for (i in ordtime56$subj) {
  plot(hrv1$time[hrv1$subj==i], hrv1$RSA[hrv1$subj==i],  ylab="HR Variability", xlab="Time",
       type="o", pch=19, col="blue", ylim=c(.8,10), xlim=c(0, 13),
       main=paste("id", i, "(", round(ordtime56$difRSA[ordtime56$subj==i], digits=1),")"))
       points(hrv1$time[hrv1$subj==i], predict(rsaout1p6)[hrv1$subj==i], pch="-", cex=3, col="red")
       abline(v=c(5, 6, 10))
}
dev.off()
```

```
## pdf
##   2
```

```
#Prepare for Spaghetti Plot
```

```
#Create phase means for plotting: First, Condition = "Invisible"
minv<-matrix(c(1, 0, 0, 0, 0, 0, 0, 0,
               1, 0, 1, 0, 0, 0, 0, 0,
               1, 0, 0, 1, 0, 0, 0, 0,
               1, 0, 0, 0, 1, 0, 0, 0), nrow=4, byrow=TRUE)
library(lme4)
phasemeansinv<-minv%*%fixef(rsaout2p6)
#Set up  (13, 1) vector of phase means
vinv<-c(rep(phasemeansinv[1,1], 5), phasemeansinv[2, 1], rep(phasemeansinv[3,1], 4), rep(phasemeansinv[4
#Set up time
time<-seq(1:13)
#Bind in a matrix
predinv<-cbind(time, vinv)
```

```
#Create phase means for plotting: Second, Condition = "Visible"
mvis<-matrix(c(1, 1, 0, 0, 0, 0, 0, 0,
               1, 1, 1, 0, 0, 1, 0, 0,
               1, 1, 0, 1, 0, 0, 1, 0,
               1, 1, 0, 0, 1, 0, 0, 1), nrow=4, byrow=TRUE)

phasemeansvis<-mvis%*%fixef(rsaout2p6)
#Set up  (13, 1) vector of phase means
vvis<-c(rep(phasemeansvis[1,1], 5), phasemeansvis[2, 1], rep(phasemeansvis[3,1], 4), rep(phasemeansvis[
#Set up time
time<-seq(1:13)
#Bind in a matrix
predvis<-cbind(time, vvis)


#Phase6 spaghetti plot
pdf(file="phase6-lmer-hrv-spaghetti-plot.pdf", width=14, height=10)
par(mfcol=c(1,2))
plot(hrv1$time[hrv1$Condition=="Invisible"], hrv1$RSA[hrv1$Condition=="Invisible"],
     ylab="HR Variability", xlab="Time", type="n", pch=4, xlim=c(0,13), ylim=c(0,10), main="Invisible C
for (i in unique(hrv1$subj[hrv1$Condition=="Invisible"]))
{
  lines(hrv1$time[hrv1$Condition=="Invisible" & hrv1$subj==i],
        predict(rsaout2p6)[hrv1$Condition=="Invisible" & hrv1$subj==i], lwd=2)
}
#Add fixed effects: phasemeansinv
lines(predinv[, 1], predinv[, 2], lwd=5, col="red")
abline(v=c(5, 6, 10))

plot(hrv1$time[hrv1$Condition=="Visible"], hrv1$RSA[hrv1$Condition=="Visible"], ylab="HR Variability",
     xlab="Time", type="n", pch=4, xlim=c(0,13), ylim=c(0,10), main="Visible Condition")
for (i in unique(hrv1$subj[hrv1$Condition=="Visible"]))
{
  lines(hrv1$time[hrv1$Condition=="Visible" & hrv1$subj==i],
        predict(rsaout2p6)[hrv1$Condition=="Visible" & hrv1$subj==i], lwd=2)
}
#Add fixed effects: phasemeansvis
lines(predvis[, 1], predvis[, 2], lwd=5, col="red")
abline(v=c(5, 6, 10))
dev.off()
```

```
## pdf
##   2
```