Repeated Measures ANOVA

André Meichtry

Contents


```
library(lme4)
library(lmerTest)
library(psych)
library(ggplot2)
```

Repeated measures ANOVA

Model AMT (11.4.1)

Repeated Measures ANOVA with **one within-subject factor**.

$Y_{ij} = \mu + \alpha_j + \pi_i + \epsilon_{ij}, \quad i = 1, ..., n; \quad j = 1, ..., I.$

- π_i are subjects effects, they could be considered **fixed**, but most often, we will treat them as **random effects**, that is
- $\pi_i \sim N(0, \nu^2)$ are **random intercepts** with **between-subject** variance ν^2
- $\epsilon_{ij} \sim N(0, \tau^2)$ with **within-subject** variance τ^2
- within-subject correlation $\rho = Cor(Y_{ij}, Y_{ik}) = \frac{\nu^2}{\nu^2+\tau^2}$ for $j \neq k$.
- $\sigma^2 = \nu^2 + \tau^2$

- This model is called a **Linear Mixed Model (LMM)**. In contrast to linear models, they have **additional random part** to model the **within-subject correlation**. ρ is called the **intra-class correlation**.
- The advantage of treating the π_i as random is that
  - we need less parameters (one between-subject variance ν^2 instead of n parameters π_i)
  - Fixed-effects parameters do not have interpretation as population parameters.

Within-subject factor with 2 levels

The simplest Repeated Measures ANOVA is the **paired t-test** with $I = 2$

Let us simulate some data with an R-function. **You need not to understand the code for simulation**.

```
RepData<-function(n=30,I=2,mu=20,alpha=runif(I-1,mu/2,mu),nu=10,tau=5)
{
    N<-n*I ##observations
    set.seed(4)
    subject<-gl(n,I,N,labels=paste("s",1:n,sep=""))
    U<-rep(rnorm(n,0,nu),each=I)##random intercept
    E <- rnorm(N,0,tau)##random error
    time <- gl(I,1,N,labels=paste("t",1:I,sep=""))
    X<-model.matrix(~time)
    fixed <- c(mu,alpha) ##parameters (mu, alpha)
    response <- X%*%fixed+U+E ##systematic part + random intercept + random error
    data <- data.frame(subject,time,response)
    parameters<-c(n=n,I=I,mu=mu,alpha=alpha,nu=nu,tau=tau,rho=nu^2/(nu^2+tau^2))
    l<-list(data=data,parameters=parameters)
    return(l)
}
```

```
tmp<-RepData() ##with default arguments
d.long2<-tmp$data
parms<-tmp$parameters
```

The data.frame `d.long2` consists of time points 1 and 2. The true values are

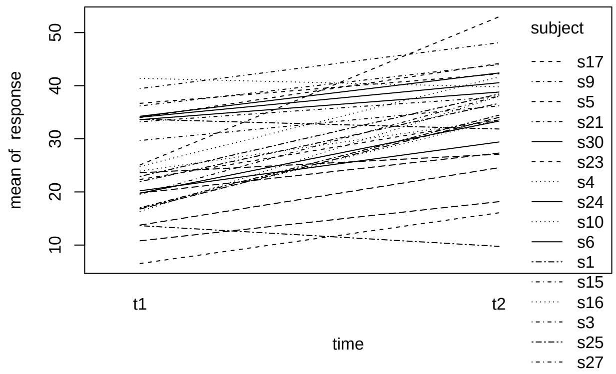
```
parms
```

```
    n     I    mu alpha    nu   tau   rho
 30.0   2.0  20.0  11.0  10.0   5.0   0.8
```

```
aggregate(response~time,data=d.long2,summary)
```

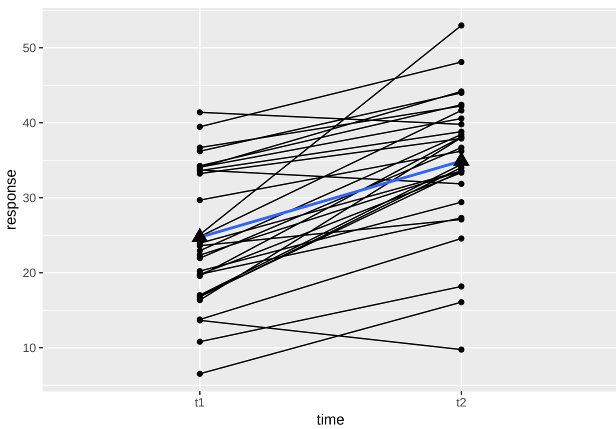
```
  time response.Min. response.1st Qu. response.Median response.Mean response.3rd Qu. response.Max.
1   t1          6.53            17.65           23.27         24.73            33.67         41.39
2   t2          9.75            32.23           36.44         34.89            40.37         52.98
```

```
with(d.long2,interaction.plot(time,subject,response))
```



A popular package for plotting is the **ggplot2** package:

```
p <- ggplot(data = d.long2, aes(x = time, y = response, group = subject))
p <- p+geom_point()+geom_line()+stat_smooth(aes(group = 1),method="lm",se=FALSE)
p <- p + stat_summary(aes(group=1), geom = "point", fun.y = mean,shape = 17, size = 4)
p
```



As paired t-Test

```
t.test(response~time,paired=TRUE,data=d.long2)
```

```
    Paired t-test

data:  response by time
t = -8, df = 29, p-value = 9e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -12.77  -7.54
sample estimates:
mean of the differences
                  -10.2
```

```
cor(d.long2$response[d.long2$time=="t1"],d.long2$response[d.long2$time=="t2"])
```

```
[1] 0.71
```

As one-sample t-Test for the individual changes

```
x<-d.long2$response[d.long2$time=="t1"]
y<-d.long2$response[d.long2$time=="t2"]
t.test(y-x)
```

```
    One Sample t-test

data:  y - x
t = 8, df = 29, p-value = 9e-09
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  7.54 12.77
sample estimates:
mean of x
     10.2
```

Observed correlation

```
cor(x,y)
```

```
[1] 0.71
```

As ANOVA

aov() provides a wrapper to **lm()** for fitting linear models. The main difference from lm is in the way print, summary and so on handle the fit: this is expressed in the traditional language of the analysis of variance rather than that of linear models. If the formula contains a single Error term, this is used to specify error strata, and appropriate models are fitted within each error stratum.

```
modelRep1<-aov(response~time+Error(subject),data=d.long2)
print(summary(modelRep1),digits=4)
```

```
Error: subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 29   4197   144.7

Error: Within
          Df Sum Sq Mean Sq F value  Pr(>F)
time       1 1547.6  1547.6      63 9.4e-09
Residuals 29  712.4    24.6
```

Repeat Sum of Squares...

Let us repeat the concept of **sum of squares** and reproduce the results above.

```
mod0 <- lm(response~1,d.long2)
mods <- lm(response~subject,d.long2)
modt <- lm(response~time,d.long2)
modts <-lm(response~subject+time,d.long2)
```

Model fits

```r
rss.0 <- sum((mod0$residuals)^2)
#(ss.0<-sum((d.long2$response-mod0$fitted)^2)) ##equivalent...
rss.s <- sum((mods$residuals)^2)
rss.t <- sum((modt$residuals)^2)
rss.ts<- sum((modts$residuals)^2)
```

Residual sum of squares

```r
rss.0
```

Explained Sum of Squares

```
[1] 6457
```

```r
rss.0-rss.s
```

```
[1] 4197
```

```r
rss.0-rss.t
```

```
[1] 1548
```

```r
rss.0-rss.ts
```

```
[1] 5744
```

```r
rss.ts
```

```
[1] 712
```

As Linear Mixed Model (LMM)

LMM are an alternative for the analysis of repeated measurements for unbalanced data or data with missing values. We will come back to LMM later. We use the **lmer()** function of the package **lme4** and **lmerTest**. LMM are fitted using **Maximum Likelihood Estimation** (in contrast to **lm()** and **aov()** which are fitted using **Least Squares**).

The syntax for the model is

```r
lmm1<-lmer(response~time+(1|subject), data=d.long2)
summary(lmm1,cor=FALSE)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: response ~ time + (1 | subject)
   Data: d.long2

REML criterion at convergence: 408

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.0380 -0.4876 -0.0289  0.5657  2.1045

Random effects:
 Groups   Name        Variance Std.Dev.
 subject  (Intercept) 60.1     7.75
```

```
 Residual              24.6     4.96
Number of obs: 60, groups:  subject, 30

Fixed effects:
            Estimate Std. Error    df t value Pr(>|t|)
(Intercept)    24.73       1.68 38.57   14.73  < 2e-16
timet2         10.16       1.28 29.00    7.94  9.4e-09
```

Compare to true parameter values:

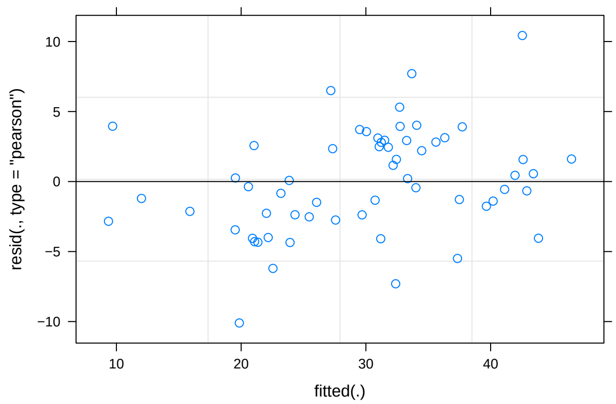
```
parms
```

```
    n     I    mu alpha    nu   tau   rho
 30.0   2.0  20.0  11.0  10.0   5.0   0.8
```

```
anova(lmm1)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
     Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
time   1548    1548     1    29      63 9.4e-09
```

```
plot(lmm1)
```



Arbitrary number of levels

The within-subject factor time now has $I = 4$ levels: Simulate some data:

```
tmp<-RepData(n=50,I=4,mu=20,alpha=c(2,3,4),nu=10,tau=5)
d.long<-tmp$data
parms<-tmp$parameters
```

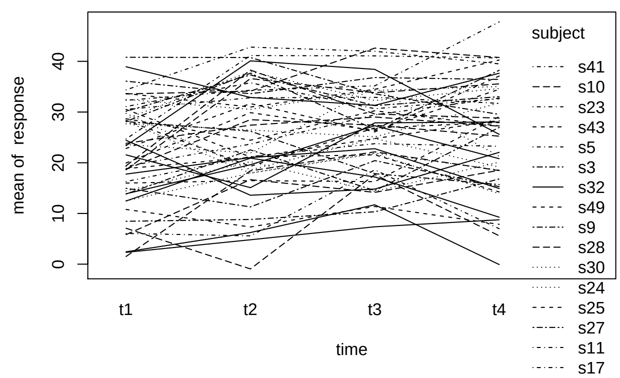
```
parms
```

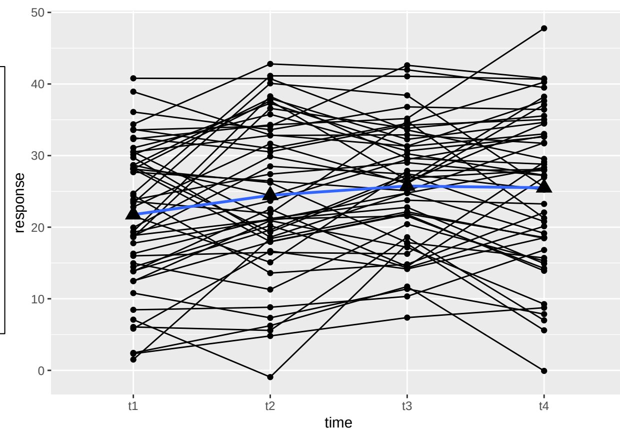
```
     n      I     mu alpha1 alpha2 alpha3     nu    tau    rho
  50.0    4.0   20.0    2.0    3.0    4.0   10.0    5.0    0.8
```

```
aggregate(response~time,data=d.long,summary)
```

```
  time response.Min. response.1st Qu. response.Median response.Mean response.3rd Qu. response.Max.
1   t1         1.514           14.577          23.090        21.755           29.467        40.798
2   t2        -0.935           18.085          23.107        24.454           33.429        42.803
3   t3         7.371           19.054          26.458        25.748           32.032        42.612
4   t4        -0.073           18.473          27.236        25.510           34.115        47.778
```

```r
with(d.long,interaction.plot(time,subject,response))
p <- ggplot(data = d.long, aes(x = time, y = response, group = subject))
p <- p+geom_point()+geom_line()+stat_smooth(aes(group = 1),se=FALSE)
p <- p + stat_summary(aes(group=1), geom = "point", fun.y = mean,shape = 17, size = 4)
p
```





As ANOVA

```r
modelRep2 <-aov(response~time+Error(subject),data=d.long)
summary(modelRep2)
```

```
Error: subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 49  16491     337

Error: Within
           Df Sum Sq Mean Sq F value  Pr(>F)
time        3    502     167    6.71 0.00028
Residuals 147   3669      25
```

As LMM

```r
lmm2 <- lmer(response~time+(1|subject),data=d.long)
summary(lmm2,cor=FALSE)
```

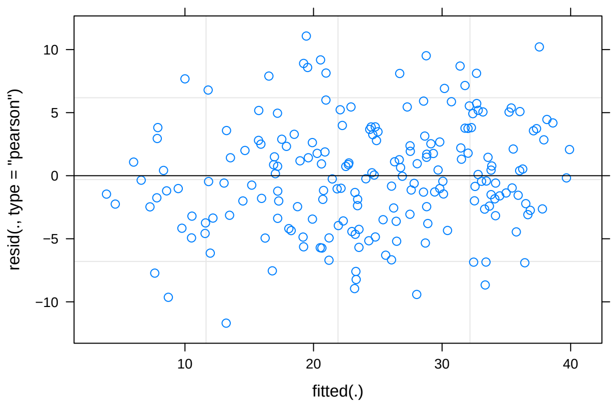
```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: response ~ time + (1 | subject)
   Data: d.long

REML criterion at convergence: 1330

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3401 -0.6134 -0.0499  0.5716  2.2168

Random effects:
 Groups   Name        Variance Std.Dev.
```

```
 subject  (Intercept) 77.9     8.83
 Residual             25.0     5.00
Number of obs: 200, groups:  subject, 50

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)   21.755      1.434  72.041   15.17  < 2e-16
timet2         2.699      0.999 147.000    2.70  0.00771
timet3         3.993      0.999 147.000    4.00  0.00010
timet4         3.755      0.999 147.000    3.76  0.00025
```

```r
anova(lmm2)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
     Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
time    502     167     3   147    6.71 0.00028
```

```r
plot(lmm2)
```



One within-subject, one between-subject factor

A frequent question is the changes of 2 groups from Pre to Post. This corresponds to a model with one **within-subject factor time** and one **between-subject factor group**:

$Y_{ijk} = \mu + \alpha_j \times \beta_k + \pi_i + \epsilon_{ijk}, \quad i = 1, ..., n \quad k = 1, 2 \quad j = 1, 2.$ with

- α_j as time effects
- β_k as group effects
- $\alpha_j : \beta_k$ as interaction effects. (=difference in slopes, effect of one predictor depends on the value on the other predictor.)

Simulation

Let us again simulate some data with an R-function. You need not to understand the code for simulation! But if you are interested, you can play with.

```r
RepData2<-function(n=100,mu=100,alpha=3,beta=5,gamma=0,nu=10,tau=5)
{
    N<-n*2 ##observations
    set.seed(65)
    subject<-gl(n,2,N,labels=paste("s",1:n,sep=""))
```

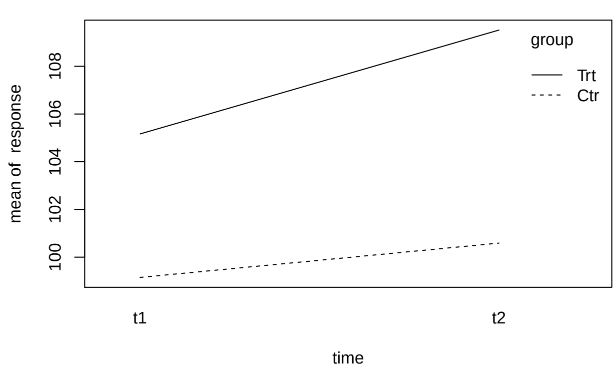
```r
    U<-rep(rnorm(n,0,nu),each=2)##random intercept
    E <- rnorm(N,0,tau)##random error
    time <- gl(2,1,N,labels=paste("t",1:2,sep=""))
    group <- as.factor(c(rep("Ctr",n),rep("Trt",n)))
    X<-model.matrix(~time*group)
    fixed <- c(mu,alpha,beta,gamma) ##parameters (mu, alpha, beta, gamma)
    response <- X%*%fixed+U+E ##systematic part + random intercept + random error
    data <- data.frame(subject,time,group,response)
    parameters<-c(n=n,mu=mu,alpha=alpha,beta=beta,gamma=gamma,nu=nu,tau=tau,rho=nu^2/(nu^2+tau^2))
    l<-list(data=data,parameters=parameters)
    return(l)
}

tmp<-RepData2() ## with default arguments
d.longB<-tmp$data
parms<-tmp$parameters
```

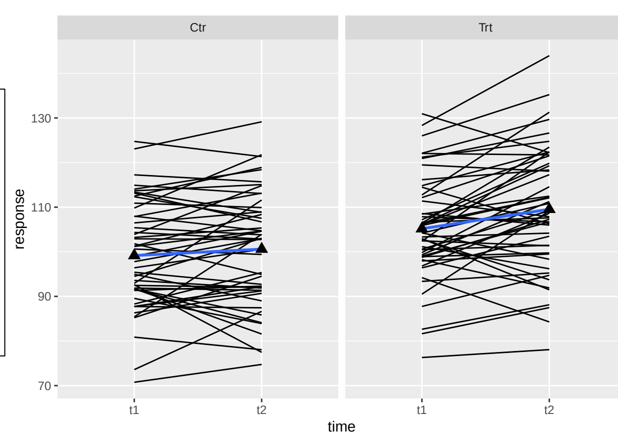
Describe data

```r
headTail(d.longB)
```

```
    subject time group response
1        s1   t1   Ctr    91.66
2        s1   t2   Ctr    85.83
3        s2   t1   Ctr    95.41
4        s2   t2   Ctr    92.69
...    <NA> <NA>  <NA>      ...
197     s99   t1   Trt   100.59
198     s99   t2   Trt   109.22
199    s100   t1   Trt    99.32
200    s100   t2   Trt   111.25
```

```r
with(d.longB,interaction.plot(time,group,response))
## with(d.longB,interaction.plot(time,subject,response))
p <- ggplot(data = d.longB, aes(x = time, y = response, group = subject))
p <- p + geom_line() + facet_grid(. ~ group)
p <- p + stat_smooth(aes(group = 1), method = "lm", se = FALSE) + stat_summary(aes(group = 1), geom = "
p
```





```r
aggregate(response~time+group,data=d.longB,summary)
```

```
  time group response.Min. response.1st Qu. response.Median response.Mean response.3rd Qu. response.Max
1   t1   Ctr          70.8             91.6            98.4          99.2            108.0          124.
2   t2   Ctr          74.7             91.5           102.9         100.6            108.9          129.
3   t1   Trt          76.3             98.9           104.2         105.2            112.0          130.
4   t2   Trt          78.1            100.1           108.9         109.5            119.7          144.
```

As ANOVA

```r
modelRep3 <-aov(response~time*group+Error(subject/time),data=d.longB) ##+Error(subject) is equivalent
print(summary(modelRep3),digits=4)
```

```
Error: subject
          Df Sum Sq Mean Sq F value  Pr(>F)
group      1   2791  2791.3   9.684 0.00244
Residuals 98  28246   288.2

Error: subject:time
           Df Sum Sq Mean Sq F value   Pr(>F)
time        1  419.4   419.4  15.446 0.000158
time:group  1  106.2   106.2   3.912 0.050754
Residuals  98 2661.2    27.2
```

As LMM

```r
lmm3 <- lmer(response~time*group+(1|subject),data=d.longB)
summary(lmm3,cor=FALSE)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: response ~ time * group + (1 | subject)
   Data: d.longB

REML criterion at convergence: 1450

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.8588 -0.4692  0.0127  0.5366  1.6971

Random effects:
 Groups   Name        Variance Std.Dev.
 subject  (Intercept) 130.5    11.43
 Residual              27.2     5.21
Number of obs: 200, groups:  subject, 100

Fixed effects:
                Estimate Std. Error     df t value Pr(>|t|)
(Intercept)        99.15       1.78 116.30   55.83   <2e-16
timet2              1.44       1.04  98.00    1.38    0.171
groupTrt            6.01       2.51 116.30    2.39    0.018
timet2:groupTrt     2.92       1.47  98.00    1.98    0.051
```

```r
anova(lmm3)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
           Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
time          419     419     1    98   15.45 0.00016
group         263     263     1    98    9.68 0.00244
time:group    106     106     1    98    3.91 0.05075
```

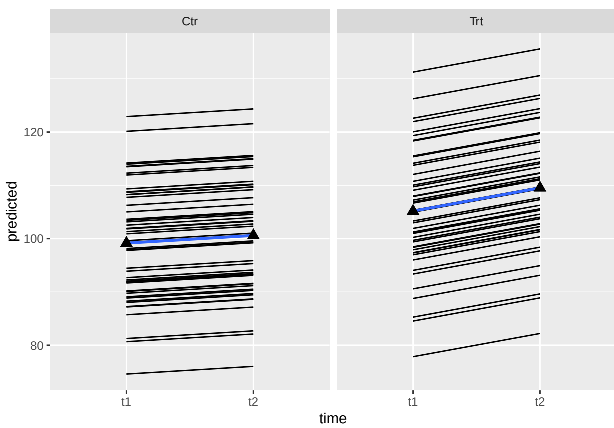
Compare to true values

```r
parms
```

```
    n    mu alpha  beta gamma    nu   tau   rho
100.0 100.0   3.0   5.0   0.0  10.0   5.0   0.8
```

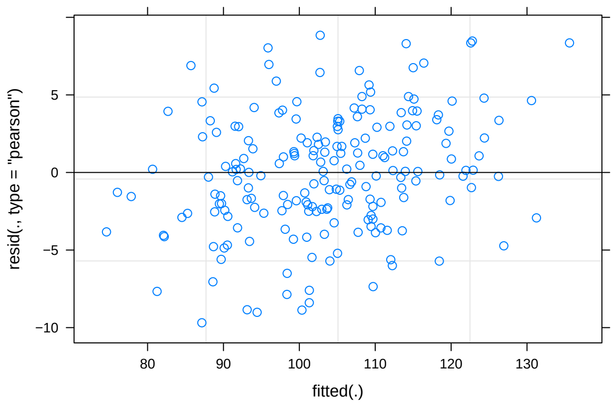
Fitted model

```r
predicted<-predict(lmm3)
p <- ggplot(data = d.longB, aes(x = time, y = predicted, group = subject))
p <- p + geom_line() + facet_grid(. ~ group)
p <- p + stat_smooth(aes(group = 1), method = "lm", se = FALSE) + stat_summary(aes(group = 1), geom = "
p
```



Residual analysis

```r
plot(lmm3)
```


resid(., type = "pearson")
fitted(.)
5
0
−5
−10
80
90
100
110
120
130