

# Sleep and Reaction

Livio Finos

## Contents

<b>Import data</b>	<b>1</b>
<b>First analysis (wrong)</b>	<b>2</b>
<b>Second analysis: Repeated measures ANOVA</b>	<b>4</b>
Sphericity . . . . .	5
<b>Third Analysis: Mixed model</b>	<b>6</b>
<b>Fourth analysis: Random Coefficient Analysis (= Summary Statistics = Second-level/Group-level Analysis)</b>	<b>9</b>
Summary statistics, one for each subject . . . . .	9
Parametric Analysis . . . . .	10
Working with permutations (Resampling-based) . . . . .	10
<b>Detour</b>	<b>16</b>
<b>(minimal) Bibliography</b>	<b>18</b>

## Import data

```
#install.packages("lme4")
data("sleepstudy", package="lme4")
summary(sleepstudy)
```

```
##      Reaction      Days      Subject
##  Min.   :194.3   Min.   :0.0   308    : 10
##  1st Qu.:255.4   1st Qu.:2.0   309    : 10
##  Median :288.7   Median :4.5   310    : 10
##  Mean   :298.5   Mean   :4.5   330    : 10
##  3rd Qu.:336.8   3rd Qu.:7.0   331    : 10
##  Max.   :466.4   Max.   :9.0   332    : 10
##                               (Other):120
```

```
library(lmerTest)
```

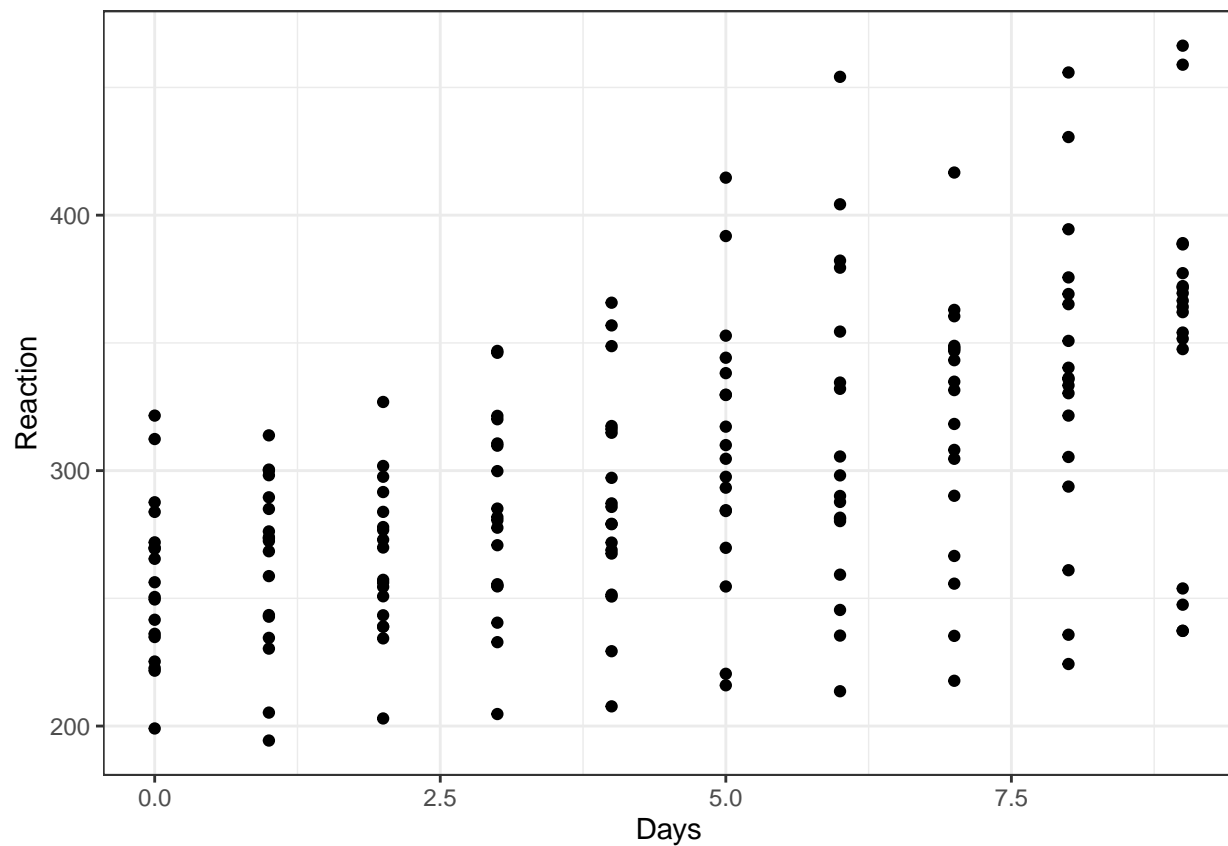
```
## Caricamento del pacchetto richiesto: lme4
## Caricamento del pacchetto richiesto: Matrix
##
## Caricamento pacchetto: 'lmerTest'
## Il seguente oggetto è mascherato da 'package:lme4':
```

```
##
## lmer
## Il seguente oggetto è mascherato da 'package:stats':
##
## step
```

## First analysis (wrong)

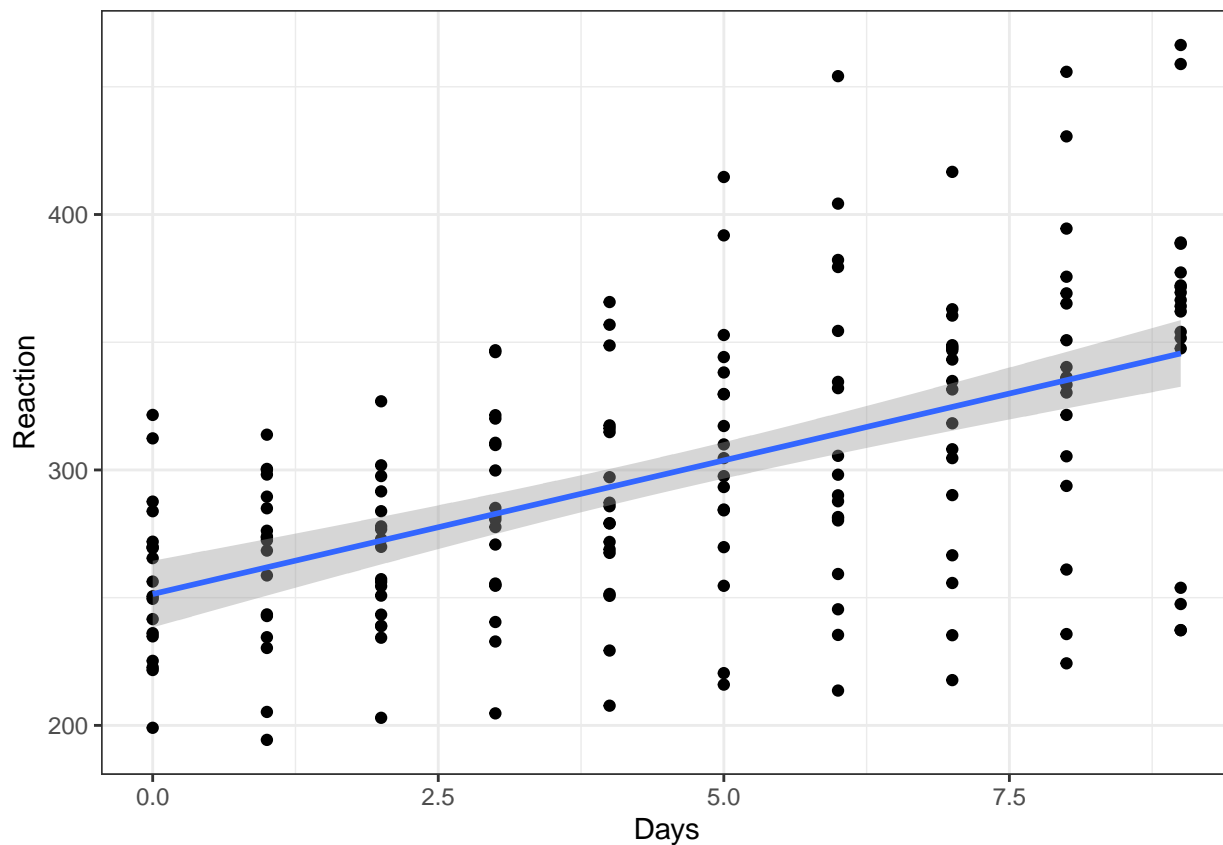
```
library(ggplot2)
p=ggplot(sleepstudy, aes(x = Days, y = Reaction)) + geom_point() + theme_bw()
```

p



```
p+geom_smooth(method=lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
mod <- lm(Reaction ~ Subject + Days, sleepstudy)
summary(mod)
```

```
##
## Call:
## lm(formula = Reaction ~ Subject + Days, data = sleepstudy)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-100.540	-16.389	-0.341	15.215	131.159

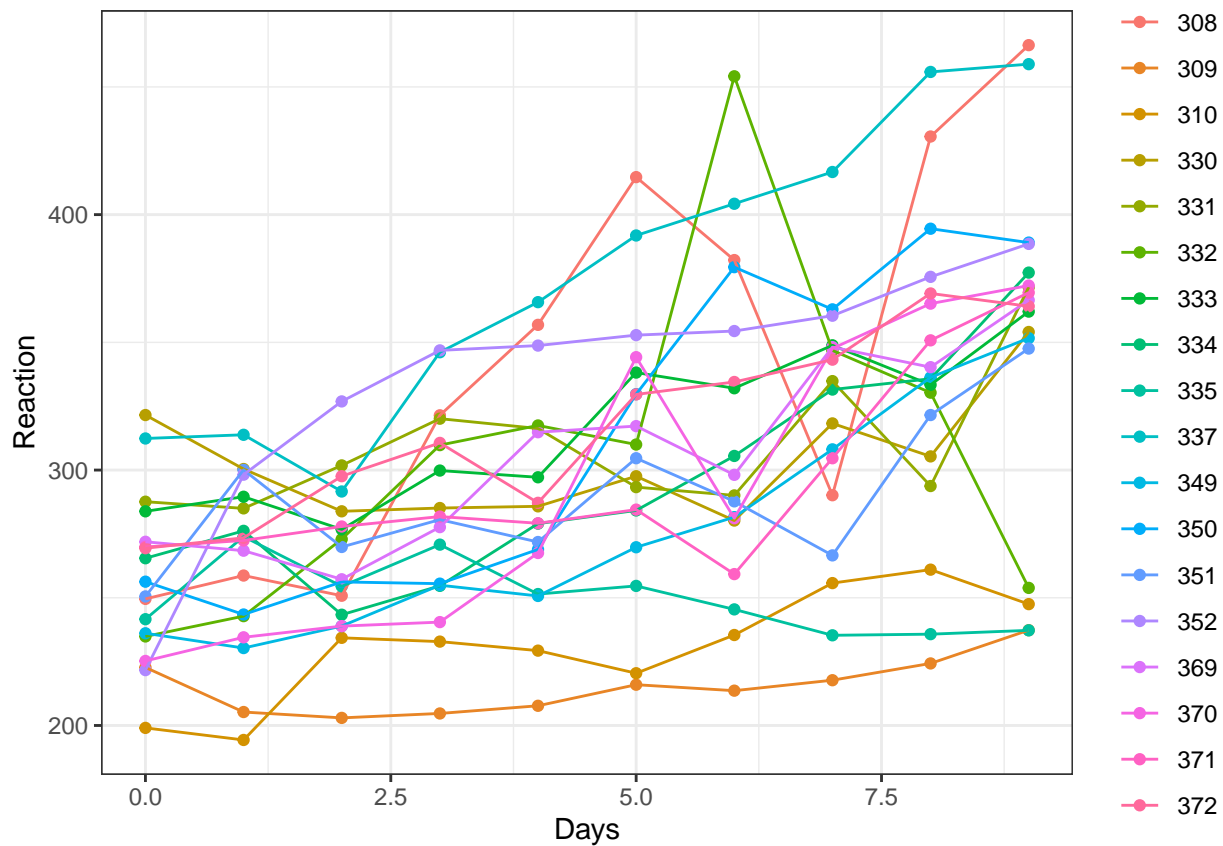
```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	295.0310	10.4471	28.240	< 2e-16 ***
Subject309	-126.9008	13.8597	-9.156	2.35e-16 ***
Subject310	-111.1326	13.8597	-8.018	2.07e-13 ***
Subject330	-38.9124	13.8597	-2.808	0.005609 **
Subject331	-32.6978	13.8597	-2.359	0.019514 *
Subject332	-34.8318	13.8597	-2.513	0.012949 *
Subject333	-25.9755	13.8597	-1.874	0.062718 .
Subject334	-46.8318	13.8597	-3.379	0.000913 ***
Subject335	-92.0638	13.8597	-6.643	4.51e-10 ***
Subject337	33.5872	13.8597	2.423	0.016486 *
Subject349	-66.2994	13.8597	-4.784	3.87e-06 ***
Subject350	-28.5311	13.8597	-2.059	0.041147 *
Subject351	-52.0361	13.8597	-3.754	0.000242 ***
Subject352	-4.7123	13.8597	-0.340	0.734300

```
## Subject369    -36.0992    13.8597    -2.605 0.010059 *
## Subject370    -50.4321    13.8597    -3.639 0.000369 ***
## Subject371    -47.1498    13.8597    -3.402 0.000844 ***
## Subject372    -24.2477    13.8597    -1.750 0.082108 .
## Days          10.4673      0.8042    13.015 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.99 on 161 degrees of freedom
## Multiple R-squared:  0.7277, Adjusted R-squared:  0.6973
## F-statistic: 23.91 on 18 and 161 DF,  p-value: < 2.2e-16
```

## Second analysis: Repeated measures ANOVA

```
p=ggplot(sleepstudy, aes(x = Days, y = Reaction, colour = Subject, group = Subject)) + geom_point() +
p+ theme_bw()
```



```
# The standard way
mod=aov(Reaction ~ Days + Error(Subject/(Days)),data=sleepstudy)
summary(mod)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 17 250618   14742
##
## Error: Subject:Days
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Days          1 162703   162703    45.85 3.26e-06 ***
## Residuals    17   60322     3548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Residuals  144   94312     654.9
```

A better output and slightly more complete analysis (Sphericity Corrections):

```
library(ez)
```

```
## Warning: il pacchetto 'ez' è stato creato con R versione 4.5.2
```

```
mod=ezANOVA(dv=Reaction, wid=Subject, within=.(Days),data=sleepstudy,type=3)
```

```
## Warning: "Days" will be treated as numeric.
```

```
## Warning: There is at least one numeric within variable, therefore aov() will be
## used for computation and no assumption checks will be obtained.
```

```
print(mod)
```

```
## $ANOVA
##   Effect DFn DFd      F      p p<.05      ges
## 1    Days   1  17 45.85301 3.263788e-06 * 0.7295277
```

## Sphericity

Sphericity is an assumption about the structure of the covariance matrix in a repeated measures design. It is not an issue in this example, but it is relevant when more than two conditions are measured within subject and compared. Assume, for example that three conditions are compared: **Condition A**, **Condition B** and **Condition C**.

Before we describe Sphericity, let's consider a simpler (but more strict) condition.

### Compound symmetry

Compound symmetry holds true when the variances within conditions are equal (this is the same as the homogeneity of variance assumption in between-group designs) but also when the covariances between pairs of conditions are roughly equal. As such, we assume that the variation within experimental conditions is fairly similar and that no two conditions are any more dependent than any other two.

Provided the observed covariances are roughly equal in our samples (and the variances are OK too) we can be pretty confident that compound symmetry is not violated.

Compound symmetry is met when the correlation between **Condition A** and **Condition B** is equal to the correlation between **Condition A** and **Condition C** and between **Condition B** and **Condition C** (and all pairwise comparisons, in general). But a more direct way to think about compound symmetry is to say that it requires that all subjects in each group change in the same way over trials. In other words the slopes of the lines regressing the dependent variable on time are the same for all subjects. Put that way it is easy to see that compound symmetry can really be an unrealistic assumption.

## Sphericity

Although compound symmetry has been shown to be a sufficient condition for conducting ANOVA on repeated measures data, it is not a necessary condition. Sphericity is a less restrictive form of compound symmetry. Sphericity refers to the equality of variances of the differences between treatment levels. If you were to take

each pair of treatment levels, and calculate the differences between each pair of scores it is necessary that these differences have equal variances.

We can check sphericity assumption using the covariance matrix, but it turns out to be fairly laborious. Remember that variance of differences can be computed as:

$$Var(x - y) = S_{x-y}^2 = S_x^2 + S_y^2 - 2S_{xy}$$

Further reading: [https://en.wikipedia.org/wiki/Mauchly%27s\\_sphericity\\_test](https://en.wikipedia.org/wiki/Mauchly%27s_sphericity_test)

This is often an unrealistic assumption in EEG data (spatial location of channel relates to correlation between measures)

### (Further) Limitations of Repeated Measures ANOVA

- (Design and) Data must be balanced
- Repeated Measures Anova doesn't allow for missing data
- It only handle factors, no quantitative variables

Mixed model is a more flexible approach.

## Third Analysis: Mixed model

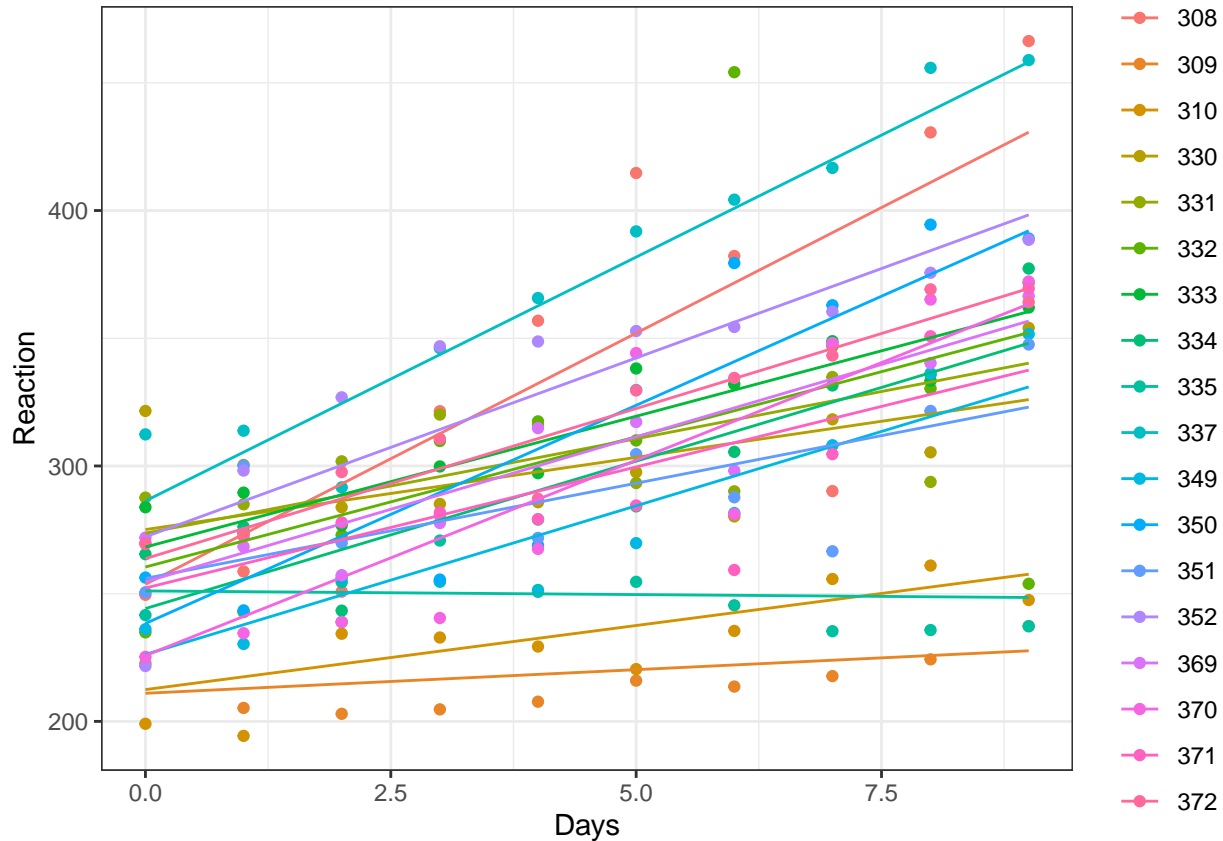
### Random intercept and slope

```
library(lmerTest)
mod <- lmer(Reaction ~ 1+Days + (1 + Days | Subject), data=sleepstudy)
summary(mod)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Reaction ~ 1 + Days + (1 + Days | Subject)
## Data: sleepstudy
##
## REML criterion at convergence: 1743.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.9536 -0.4634  0.0231  0.4634  5.1793
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## Subject (Intercept) 612.10 24.741
## Days 35.07 5.922 0.07
## Residual 654.94 25.592
## Number of obs: 180, groups: Subject, 18
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 251.405 6.825 17.000 36.838 < 2e-16 ***
## Days 10.467 1.546 17.000 6.771 3.26e-06 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr)
```

```
## Days -0.138
```

```
sleepstudy$predicted=predict(mod)
library(ggplot2)
p=ggplot(sleepstudy, aes(x = Days, y = Reaction, colour = Subject, group = Subject)) + geom_point() + g
p+ theme_bw()
```



What if we set only the Intercept as random component?

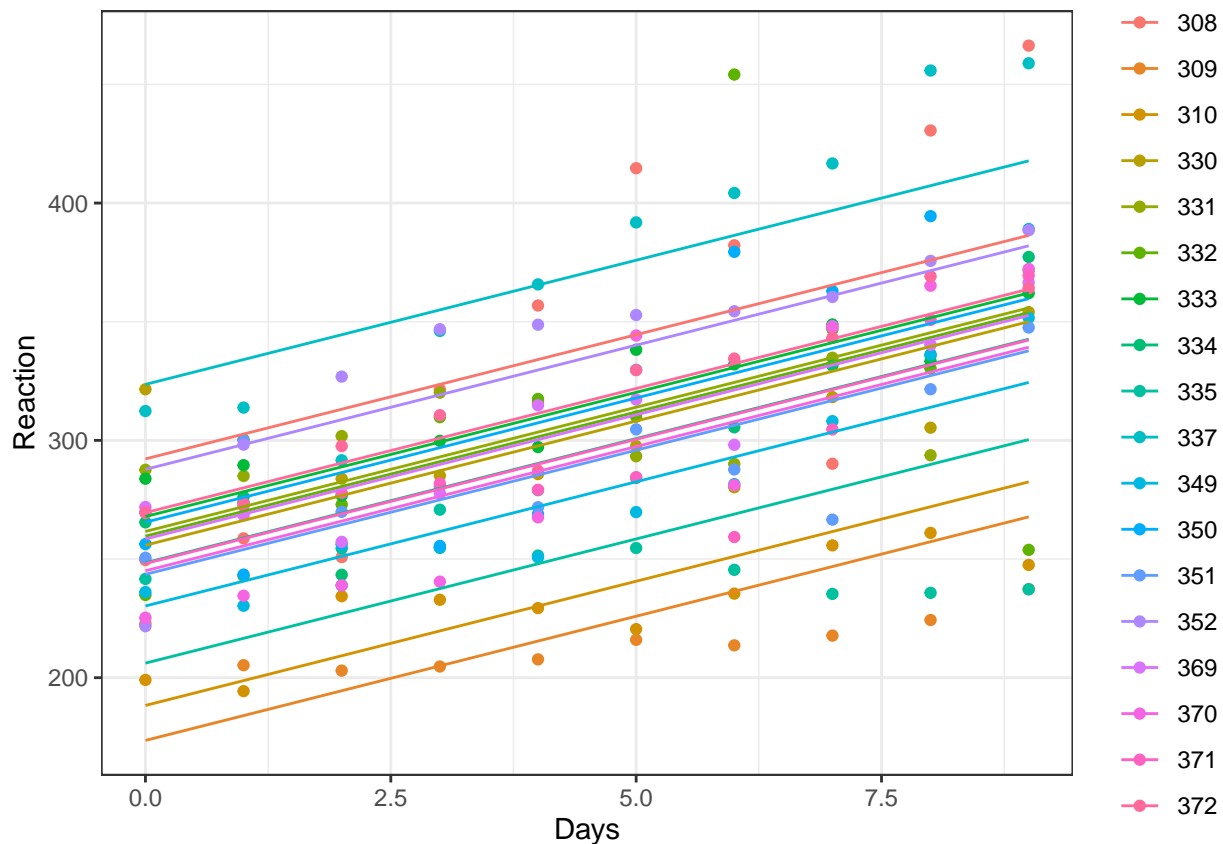
### Random intercept only

```
library(lmerTest)
mod <- lmer(Reaction ~ 1 + Days + (1 | Subject), data=sleepstudy)
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Reaction ~ 1 + Days + (1 | Subject)
## Data: sleepstudy
##
## REML criterion at convergence: 1786.5
##
## Scaled residuals:
##    Min     1Q  Median     3Q    Max
## -3.2257 -0.5529  0.0109  0.5188  4.2506
##
## Random effects:
## Groups   Name            Variance Std.Dev.
## Subject (Intercept) 1378.2    37.12
```

```
## Residual          960.5   30.99
## Number of obs: 180, groups: Subject, 18
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 251.4051    9.7467   22.8102   25.79  <2e-16 ***
## Days        10.4673     0.8042  161.0000   13.02  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.371
```

```
sleepstudy$predicted=predict(mod)
library(ggplot2)
p=ggplot(sleepstudy, aes(x = Days, y = Reaction, colour = Subject, group = Subject)) + geom_point() + geom_line()
p+ theme_bw()
```



Is this model adequate? Does it adequately fit your data?

REMARKS:

- Mixed models require careful attention in defining random effects; for example, if we do not also include a random slope coefficient, we implicitly make the same assumptions about the sphericity of the repeated measures. There is still a lively debate today about the most appropriate way to define the random component structure in mixed models, see for example Barr et al. (2013) and Bates et al. (2015) [Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3), 255-278.



Bates, D., Kliegl, R., Vasishth, S., & Baayen, H. (2015). Parsimonious mixed models. arXiv preprint arXiv:1506.04967].

- Regarding the lines estimated with mixed models: they are similar but not identical to those we obtain if we estimate a line for each subject; in mixed models, we impose that the coefficients come from the same (normally distributed) population, and this implies that the estimates are (a bit) biased.

## Fourth analysis: Random Coefficient Analysis (= Summary Statistics = Second-level/Group-level Analysis)

Summary statistics, one for each subject

```
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.1      v stringr    1.5.2
## v lubridate  1.9.4      v tibble     3.3.0
## v purrr      1.1.0      v tidyr      1.3.1
## -- Conflicts ----- tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x tidyr::pack()    masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

datillev=sleepstudy %>%
  group_by(Subject) %>%
  summarize(Slope=(coefficients(lm(Reaction ~ Days))[2]))

head(datillev)

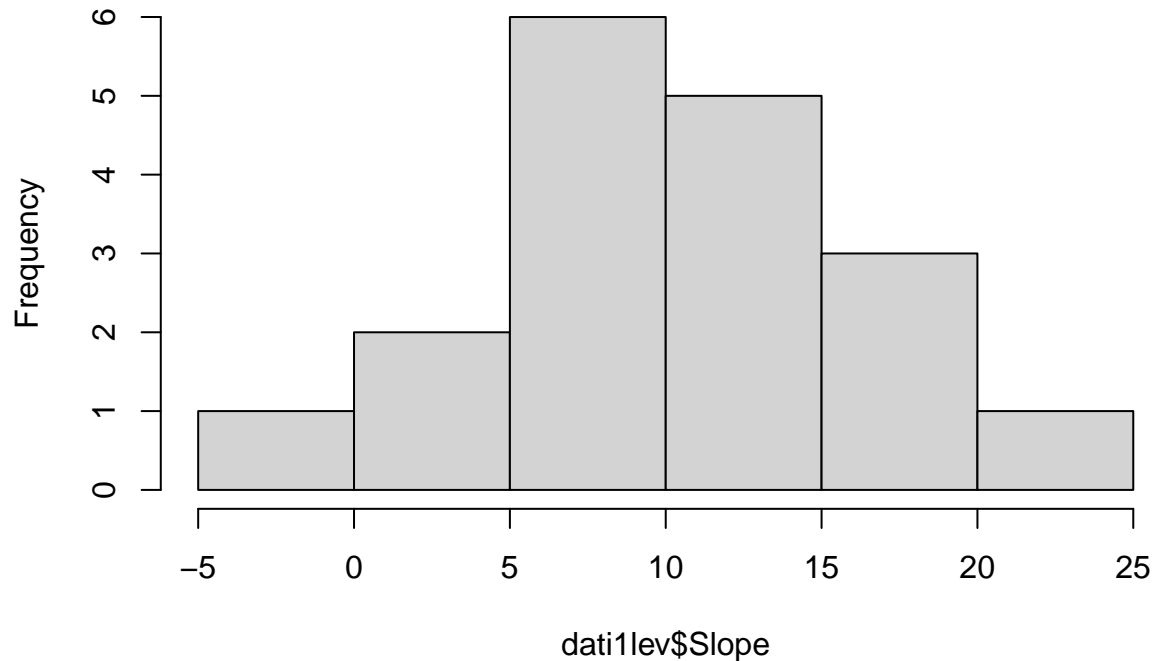
## # A tibble: 6 x 2
##   Subject Slope
##   <fct>   <dbl>
## 1 308     21.8
## 2 309     2.26
## 3 310     6.11
## 4 330     3.01
## 5 331     5.27
## 6 332     9.57

mean(datillev$Slope)

## [1] 10.46729

hist(datillev$Slope)
```

## Histogram of dati1lev\$Slope



### Parametric Analysis

If we are interested in testing if the sleep deprivation is associated with larger reaction times, we can simply test if the average of the slopes is equal to 0. This can be done with a simple one-sample t-test.

```
t.test(dati1lev$Slope)
```

```
##
## One Sample t-test
##
## data:  dati1lev$Slope
## t = 6.7715, df = 17, p-value = 3.264e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  7.205956 13.728615
## sample estimates:
## mean of x
## 10.46729
```

What are the assumptions that are done in this approach?

1. Independence of observations (= subjects in this method): OK
2. Homoscedasticity (= same variance for each observation/subject). It implies balanced design and same variability of the measures among all subjects. OK? Always OK?
3. Normality, but with an adequate number of observations, we can invoke the Central Limit Theorem.

### Working with permutations (Resampling-based)

The lack of assumption 2 can't be (easily) solved via parametric approach. Let's consider a nonparametric permutation approach.

The simple code one can use is:

```
library(jointest)
```

```
##  
## Caricamento pacchetto: 'jointest'  
## Il seguente oggetto è mascherato da 'package:dplyr':  
##  
##      combine  
## Il seguente oggetto è mascherato da 'package:stats':  
##  
##      p.adjust
```

```
mod=flip2sss(Reaction ~ 1 + Days,  
             cluster =sleepstudy$Subject,data=sleepstudy)  
summary(mod)
```

```
##           model coefficient estimate   score      se      z   pcor      p  
## 1 .Intercept. (Intercept)   251.41 4525.3 1073.28 4.216 0.9938 2e-04  
## 2           Days           Days    10.47  188.4   51.99 3.624 0.8541 2e-04
```

and in the following we show how this result is reached.

```
library(ggplot2)
```

```
perm=sleepstudy
```

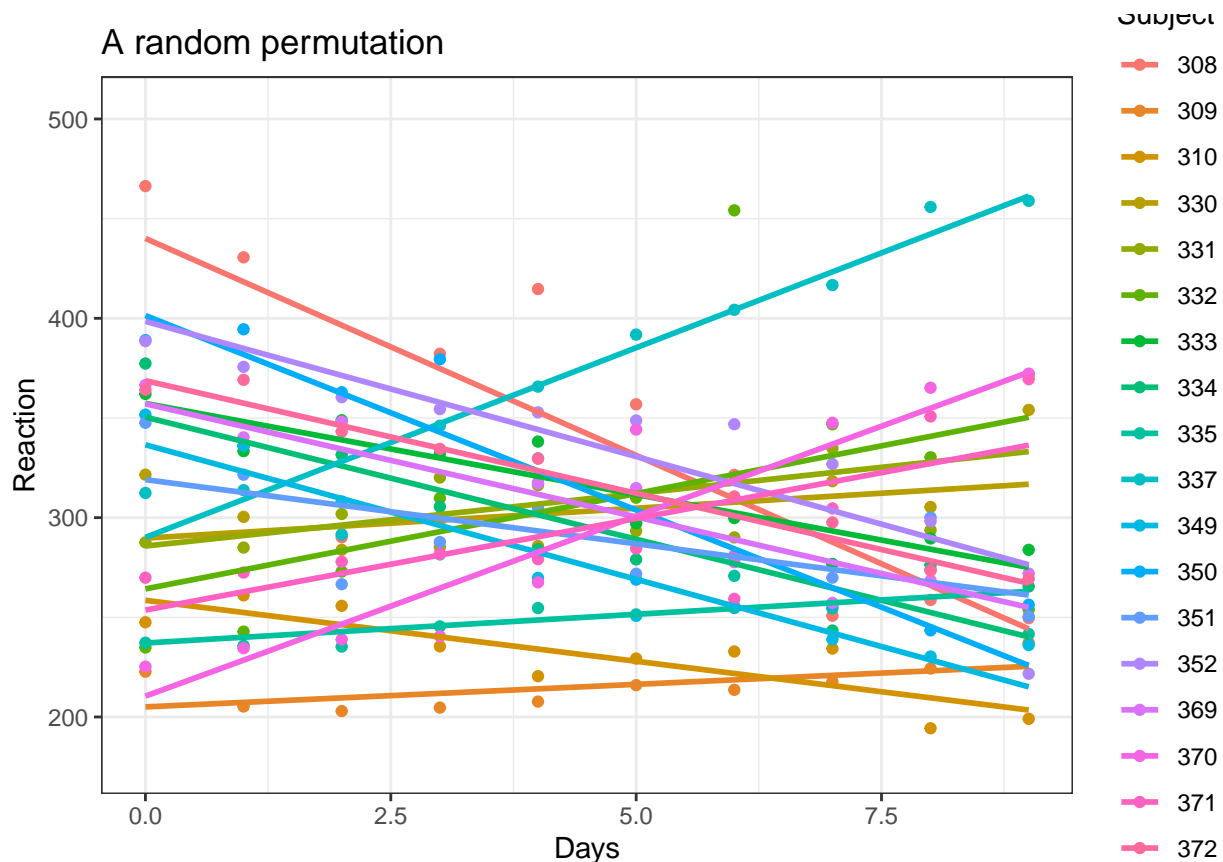
```
rnd=sapply((0:17)*10,function(strt) {  
  if(rbinom(1,1,.5)==1)  
    strt+(1:10) else  
    strt+(10:1)  
})
```

```
rnd=as.vector(rnd)
```

```
perm$Reaction=sleepstudy$Reaction[rnd]
```

```
p=ggplot(perm, aes(x = Days, y = Reaction, colour = Subject, group = Subject)) + geom_point() +geom_smooth()  
p + ggtitle("A random permutation") + theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
datillevperm=perm %>%
  group_by(Subject) %>%
  reframe(estim=(coefficients(lm(Reaction ~ Days))))
datillevperm$coeff=(rep(c("Interc", "Slope"), nlevels(sleepstudy$Subject)))
datillevperm=spread(datillevperm, key = coeff, value = estim)
```

```
head(datillevperm)
```

```
## # A tibble: 6 x 3
##   Subject Interc Slope
##   <fct>    <dbl> <dbl>
## 1 308      440. -21.8
## 2 309      205.   2.26
## 3 310      259. -6.11
## 4 330      290.   3.01
## 5 331      286.   5.27
## 6 332      264.   9.57
```

```
mean(datillevperm$Slope)
```

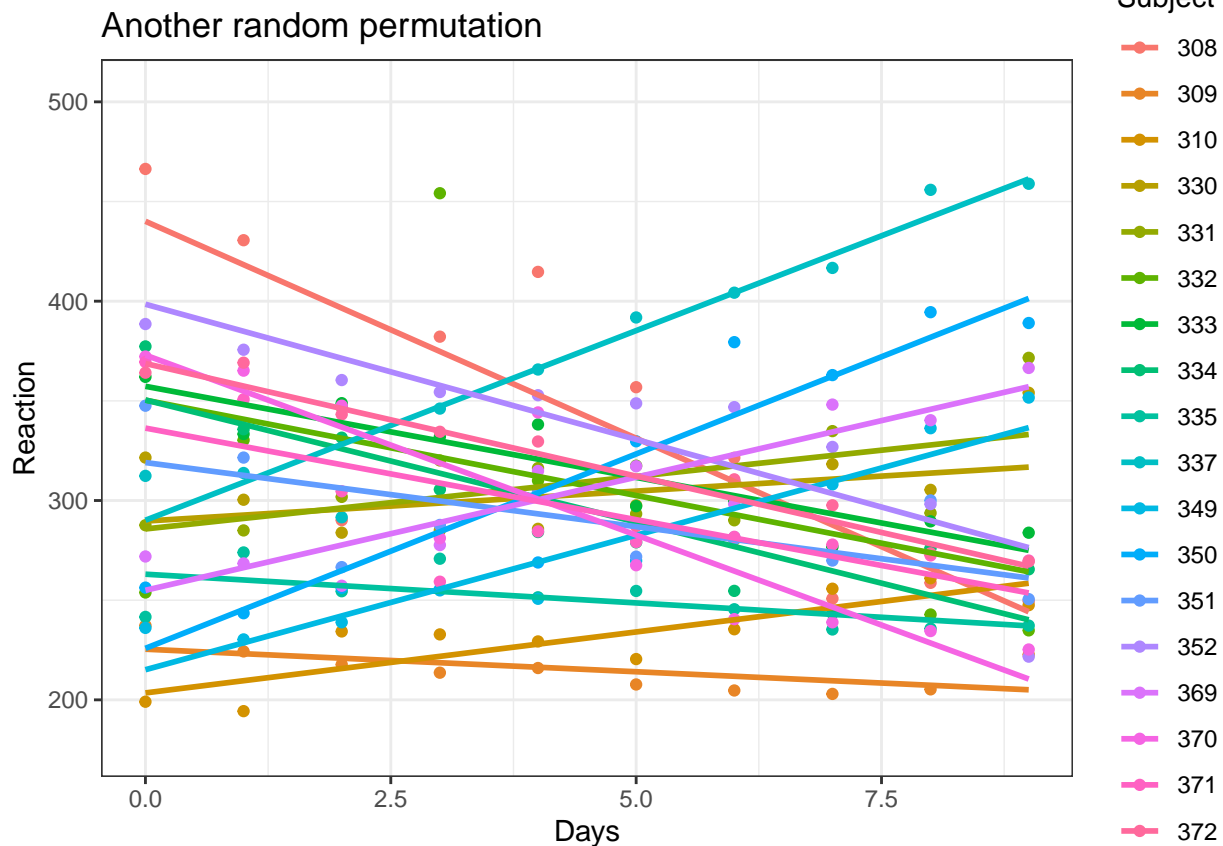
```
## [1] -3.092484
```

```
rnd=sapply((0:17)*10,function(strt) {
  if(rbinom(1,1,.5)==1)
    strt+(1:10) else
    strt+(10:1)
})
rnd=as.vector(rnd)
```

```
perm$Reaction=sleepstudy$Reaction[rnd]
```

```
p=ggplot(perm, aes(x = Days, y = Reaction, colour = Subject, group = Subject)) + geom_point() + geom_smooth()
p + ggtitle("Another random permutation") + theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
datillevperm=perm %>%
  group_by(Subject) %>%
  summarize(estim=(coefficients(lm(Reaction ~ Days))))
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'Subject'. You can override using the
## `.groups` argument.
```

```
datillevperm$coeff=(rep(c("Intercept", "Slope"), nlevels(sleepstudy$Subject)))
datillevperm=spread(datillevperm, key = coeff, value = estim)
```

```
head(datillevperm)
```

```
## # A tibble: 6 x 3
```

```
## # Groups:   Subject [6]
##   Subject Interc Slope
##   <fct>    <dbl> <dbl>
## 1 308      440. -21.8
## 2 309      225. -2.26
## 3 310      203.  6.11
## 4 330      290.  3.01
## 5 331      286.  5.27
## 6 332      350. -9.57
```

```
mean(datillevperm$Slope)
```

```
## [1] -2.147287
```

```
Slopes_flips=replicate(100,{
  rnd=sapply((0:17)*10,function(strt) {
    if(rbinom(1,1,.5)==1)
      strt+(1:10) else
      strt+(10:1)
  })
  rnd=as.vector(rnd)
  perm$Reaction=sleepstudy$Reaction[rnd]
  datillevperm=perm %>%
  group_by(Subject) %>%
  summarize(Slope=(coefficients(lm(Reaction ~ Days))[2]))
  mean(datillevperm$Slope)
})
```

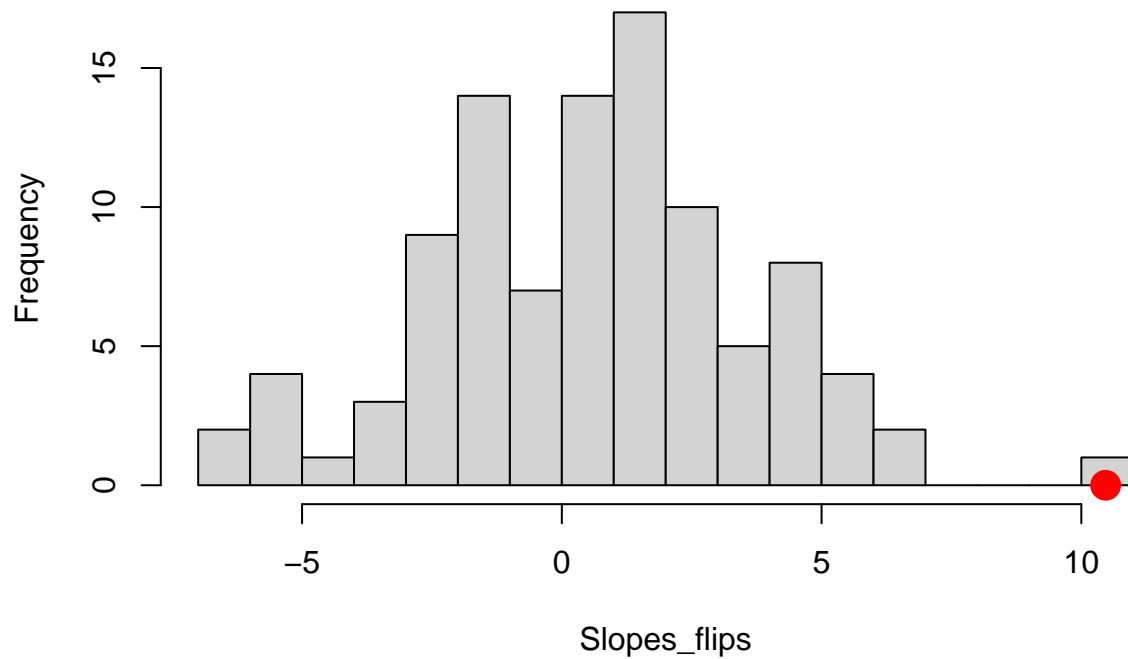
```
# ON OBSERVED DATA:
datillev=sleepstudy %>%
group_by(Subject) %>%
summarize(Slope=(coefficients(lm(Reaction ~ Days))[2]))
(Observed=mean(datillev$Slope))
```

```
## [1] 10.46729
```

```
# ALWAYS ADD THE OBSERVED TEST STATISTIC!!! (it is one of the possible permutations)
Slopes_flips=c(Observed,Slopes_flips)
```

```
hist(Slopes_flips,20)
points(Observed,0,cex=3,col="red",pch=20)
```

## Histogram of Slopes\_flips



Computing the p-value:

```
# One sided alternative
mean(Slopes_flips >= Observed)
```

```
## [1] 0.00990099
```

```
# Two sided alternative
mean(abs(Slopes_flips) >= abs(Observed))
```

```
## [1] 0.00990099
```

```
# devtools::install_github("livioivil/flipscores")
library(flipscores)
# effect of Days:
(res = flipscores(Slope ~ 1, data = dati1lev))
```

```
## Flip Score Test:
##       score_type = standardized , n_flips =
## Call: flipscores(formula = Slope ~ 1, data = dati1lev)
##
## Coefficients:
## (Intercept)
##      10.46729
```

```
summary(res)
```

```
##
## Call:
## flipscores(formula = Slope ~ 1, data = dati1lev)
##
## Coefficients:
##           Estimate      Score Std. Error z value Part. Cor Pr(>|z|)
```

```
## (Intercept) 10.467 188.411 51.994 3.624 0.854 4e-04 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 43.01034)
##
## Null deviance: 731.18 on 17 degrees of freedom
## Residual deviance: 731.18 on 17 degrees of freedom
## AIC: 121.76
##
## Number of Fisher Scoring iterations: 2
```

## Detour

Let's work on Time 0 and 1 only, one can perform paired t-test

```
sleepstudy2=sleepstudy[sleepstudy$Days<=1,]

t.test(sleepstudy2$Reaction[sleepstudy2$Days==1]-sleepstudy2$Reaction[sleepstudy2$Days==0])

##
## One Sample t-test
##
## data: sleepstudy2$Reaction[sleepstudy2$Days == 1] - sleepstudy2$Reaction[sleepstudy2$Days == 0]
## t = 1.3972, df = 17, p-value = 0.1803
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -4.000254 19.688154
## sample estimates:
## mean of x
## 7.84395
```

or linear model:

```
sleepstudy2$Subject=factor(sleepstudy2$Subject)
mod <- lm(Reaction ~ Days+ Subject, sleepstudy2)
summary(mod)

##
## Call:
## lm(formula = Reaction ~ Days + Subject, data = sleepstudy2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.336  -5.336   0.000   5.336  34.336
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  250.210    12.235   20.450 2.08e-13 ***
## Days          7.844     5.614    1.397  0.18031
## Subject309   -40.132    16.842   -2.383  0.02911 *
## Subject310   -57.439    16.842   -3.411  0.00333 **
## Subject330    56.839    16.842    3.375  0.00360 **
## Subject331    32.172    16.842    1.910  0.07312 .
## Subject332   -15.296    16.842   -0.908  0.37645
```



```
## Subject333      32.566      16.842      1.934      0.06998 .
## Subject334      16.705      16.842      0.992      0.33517
## Subject335       3.645      16.842      0.216      0.83121
## Subject337      58.954      16.842      3.500      0.00274 **
## Subject349     -20.922      16.842     -1.242      0.23099
## Subject350      -4.257      16.842     -0.253      0.80349
## Subject351      21.160      16.842      1.256      0.22596
## Subject352       5.803      16.842      0.345      0.73464
## Subject369      16.048      16.842      0.953      0.35401
## Subject370     -24.239      16.842     -1.439      0.16825
## Subject371      17.029      16.842      1.011      0.32612
## Subject372      17.311      16.842      1.028      0.31843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.84 on 17 degrees of freedom
## Multiple R-squared:  0.87, Adjusted R-squared:  0.7324
## F-statistic: 6.323 on 18 and 17 DF, p-value: 0.0001954
```

or Repeated Measures:

```
mod=aov(Reaction ~ Days + Error(Subject/(Days)),data=sleepstudy)
summary(mod)
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 17 250618   14742
##
## Error: Subject:Days
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Days       1 162703  162703   45.85 3.26e-06 ***
## Residuals 17  60322    3548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 144  94312    654.9
```

or Mixed model (note that we can't set a random slope when we have only two observations; the random coefficients would be not separable from the random error):

```
library(lmerTest)
mod <- lmer(Reaction ~ 1+Days + (1 | Subject), data=sleepstudy2)
summary(mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Reaction ~ 1 + Days + (1 | Subject)
## Data: sleepstudy2
##
## REML criterion at convergence: 326.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.04455 -0.26699 -0.01685  0.30707  2.03303
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   Subject (Intercept) 791.3    28.13
##   Residual              283.6    16.84
## Number of obs: 36, groups: Subject, 18
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   256.652      7.728   22.051   33.211   <2e-16 ***
## Days           7.844      5.614   17.000    1.397    0.18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.363
```

They all provide the same result!

## (minimal) Bibliography

lme4 package <https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf>

Jonathan Baron (2011) Notes on the use of R for psychology experiments and questionnaires [https://www.sas.upenn.edu/~baron/from\\_cattell/rpsych/rpsych.html](https://www.sas.upenn.edu/~baron/from_cattell/rpsych/rpsych.html)

Pinheiro, J., & Bates, D. (2006) Mixed-effects models in S and S-PLUS. <https://www.springer.com/gp/book/9780387954580>

Bates, D. (2022) lme4: Mixed-effects modeling with R. <https://stat.ethz.ch/~maechler/MEMO-pages/lmmwR.pdf>

Fox, J. (2015) Applied regression analysis and generalized linear models. <https://us.sagepub.com/en-us/nam/applied-regression-analysis-and-generalized-linear-models/book246125>

Fox, J., & Weisberg, S. (2018) An R companion to applied regression. <https://us.sagepub.com/en-us/nam/an-r-companion-to-applied-regression/book246125>

Bliese, P. (2022) Multilevel modeling in R (2.7). [https://cran.r-project.org/doc/contrib/Bliese\\_Multilevel.pdf](https://cran.r-project.org/doc/contrib/Bliese_Multilevel.pdf)

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008) Mixed-effects modeling with crossed random effects for subjects and items. <https://doi.org/10.1016/j.jml.2007.12.005>

Bürkner, P. C. (2017) brms: An R package for Bayesian multilevel models using Stan. <https://doi.org/10.18637/jss.v080.i01>

McElreath, R. (2020) Statistical rethinking: A Bayesian course with examples in R and Stan. <https://xcelab.net/rm/statistical-rethinking/>

GLMM FAQ by Ben Bolker and others <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>

A complete course on mixed models (in italian) <https://osf.io/b7tkp/>