

Statistical Inference with Repeated Measures (EEG) data

Livio Finos

Contents

Introduction	2
The data	2
Motivation (EDA)	3
Repeated Measures ANOVA	5
Introduction	5
2 conditions, paired observations	5
Linear models with repeated measures	8
Repeated measures	9
Spend your DF in a different way!	10
Sphericity	12
(Further) Limitations of Repeated Measures ANOVA	14
Mixed models	14
Motivation/Introduction	14
The model	15
Plotting tools	21
Validity of the assumptions	26
Contrasts and post-hoc	28
Multivariate ANOVA (MANOVA)	30
Motivation	30
Reshaping the data	31
Manova	34
Mapping results on a scalp	35
(minimal) Bibliography	37
About (generalized) linear Mixed models:	37
Resampling based:	38
More	38

Introduction

```
knitr::opts_chunk$set(echo = TRUE)
```

The data

(Fictitious data)

ERP experiment

- 20 Subjects,
- 6 Channels: O1, O2, PO7, PO8, P7, P8
- Stimuli: pictures. Conditions:
 - 1 (f): fear (face)
 - 2 (h): happiness (face)
 - 3 (d): disgust (face)
 - 4 (n): neutral (face)
 - 5 (o): object
- Measure: Area around the component P170

Setting parameters, importing the data:

```
#  
# # example of files contents:  
# # s01 NC P7 f -7.1121  
# # s01 NC P7 h -7.2582  
# # s01 NC P7 d -7.4540  
# # s01 NC P7 n -5.6729  
# # s01 NC P7 o -2.1812  
# # s01 NC PO7 f -7.4169  
#  
#  
# library(readr)  
# library(dplyr)  
#  
# dati=lapply(datafiles, read_delim,col_names = FALSE ,delim = " ")  
# dati=bind_rows(dati)  
# str(dati)  
# names(dati)=c("Subj","Group","Chan","Condition","Y")  
#  
# # Not used in this analysis  
# dati$Group=NULL  
# dati$Subj=factor(dati$Subj)  
# dati$Chan=factor(dati$Chan)  
# dati$Condition=factor(dati$Condition)  
# str(dati)  
# save(dati,file="datiEEG.Rdata")  
#  
# dati2=subset(dati,(Chan=="O1")&(Condition%in%c("f","n")))  
# dati2$Condition=factor(dati2$Condition)
```

```

# save(dati2,file="dati2EEG.Rdata")
load("./dataset/datiEEG.Rdata")
load("./dataset/dati2EEG.Rdata")

dati$Condition=factor(dati$Condition,levels=c("o","d","f","h","n"))

# VERY IMPORTANT:
contrasts(dati$Chan) <- contr.sum(6)
contrasts(dati$Condition) <- contr.sum(5)
contrasts(dati$Subj) <- contr.sum(nlevels(dati$Subj))

contrasts(dati2$Condition) <- contr.sum(2)
contrasts(dati2$Subj) <- contr.sum(nlevels(dati2$Subj))

```

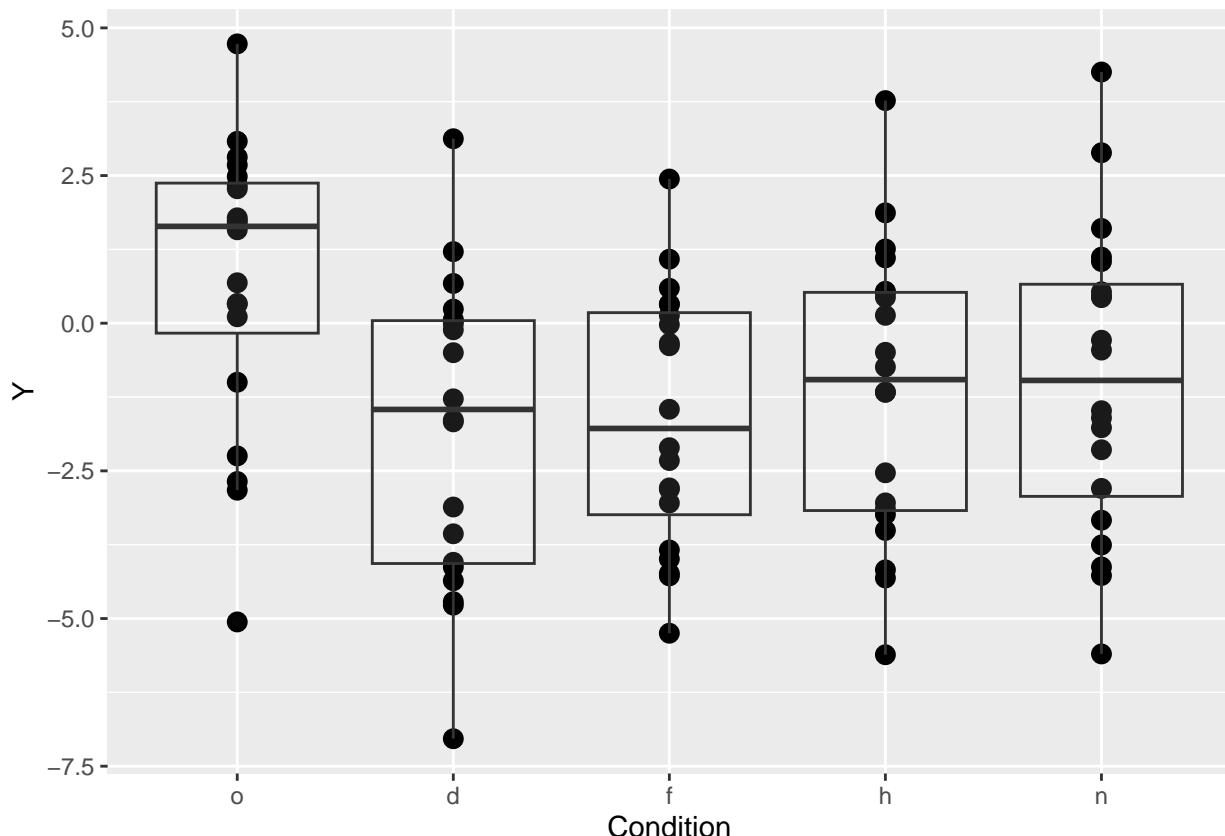
Motivation (EDA)

For Channel 01:

```

library(ggplot2)
p <- ggplot(subset(dati, Chan=="01"), aes(Condition, Y))
p+geom_point(size = 3) +geom_boxplot(alpha=.1)

```

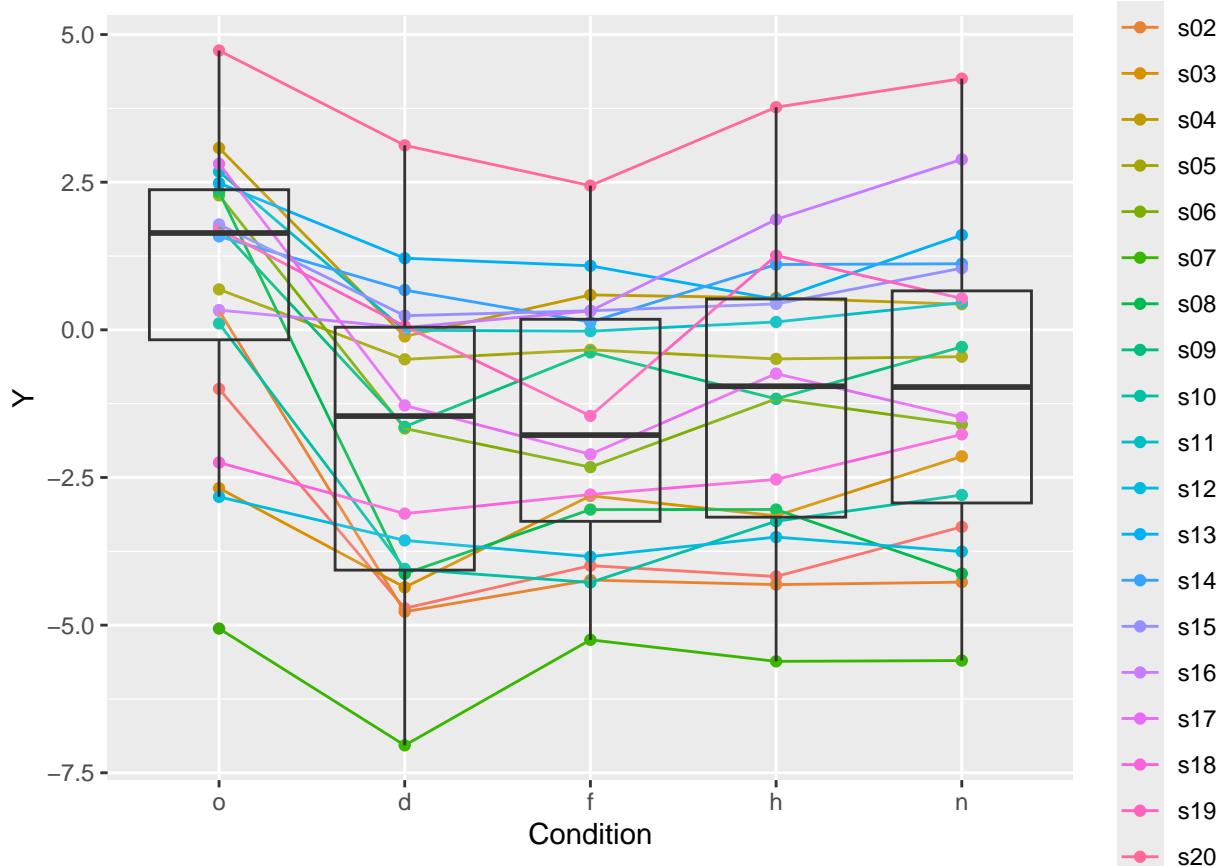


Is there a specificity of the subject?

```

dati01=subset(dati,Chan=="01")
library(ggplot2)
p <- ggplot(dati01,aes(Condition,Y))
p+geom_point(aes(group = Subj, colour = Subj))+
  geom_line(aes(group = Subj, colour = Subj))+
  geom_boxplot(alpha=.1)

```



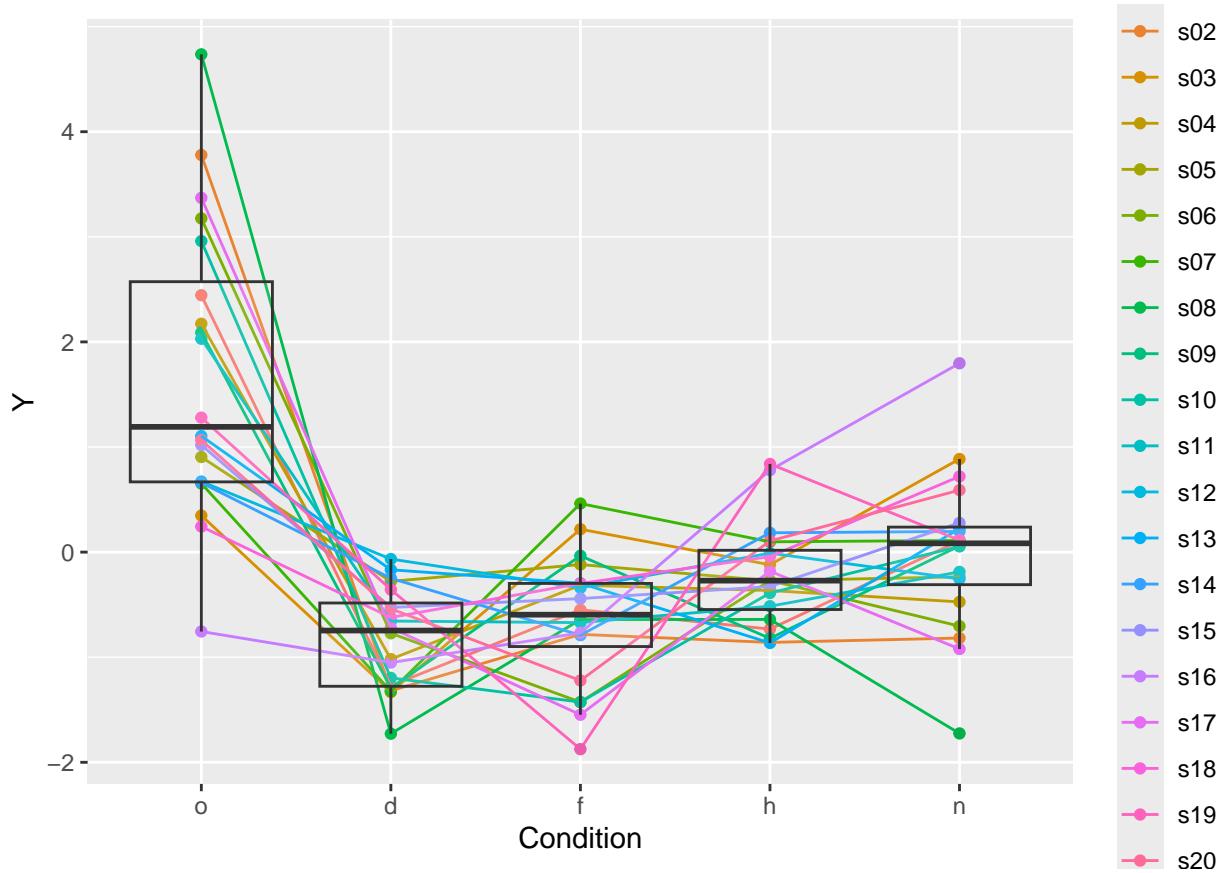
We subtract the subject-specific effect (i.e. subject's mean) to each observation.

```

dati01=subset(dati,Chan=="01")
Y=scale(matrix(dati01$Y,5),scale=FALSE)
dati01$Y=as.vector(Y)

library(ggplot2)
p <- ggplot(dati01,aes(Condition,Y))
p+geom_point(aes(group = Subj, colour = Subj))+
  geom_line(aes(group = Subj, colour = Subj))+
  geom_boxplot(alpha=.1)

```



The dispersion of the data has been largely reduced. This effect is the one taken in account by the models for repeated measures.

Repeated Measures ANOVA

Introduction

wiki reference: https://en.wikipedia.org/wiki/Repeated_measures_design

A nice explanation can be found (in particular see 7.9 and 7.10):

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

and in the Course material of

ST 732, Applied Longitudinal Data Analysis, NC State University by Marie Davidian <https://www.stat.ncsu.edu/people/davidian/courses/st732/notes/chap5.pdf> from <https://www.stat.ncsu.edu/people/davidian/courses/st732/>

2 conditions, paired observations

Let consider the reduced problem: channel `Chan=="01` and `Condition=="n"` or `Condition=="f"`.

How to compare the two conditions? First try:

```
t.test(dati2$Y[dati2$Condition=="n"] ,
       dati2$Y[dati2$Condition=="f"])

##
##  Welch Two Sample t-test
##
## data: dati2$Y[dati2$Condition == "n"] and dati2$Y[dati2$Condition == "f"]
## t = 0.8449, df = 36.861, p-value = 0.4036
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.8868791 2.1552491
## sample estimates:
## mean of x mean of y
## -0.964530 -1.598715
```

Is it ok?

NO! We don't take in account the fact that measures are taken on the same subject!

```
t.test(dati2$Y[dati2$Condition=="n"] ,
       dati2$Y[dati2$Condition=="f"], paired=TRUE)

##
## Paired t-test
##
## data: dati2$Y[dati2$Condition == "n"] and dati2$Y[dati2$Condition == "f"]
## t = 3.287, df = 19, p-value = 0.003877
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
## 0.2303616 1.0380084
## sample estimates:
## mean difference
## 0.634185

## equivalent to
t.test(dati2$Y[dati2$Condition=="n"] -
       dati2$Y[dati2$Condition=="f"])

##
## One Sample t-test
##
## data: dati2$Y[dati2$Condition == "n"] - dati2$Y[dati2$Condition == "f"]
## t = 3.287, df = 19, p-value = 0.003877
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.2303616 1.0380084
## sample estimates:
## mean of x
## 0.634185
```

Can you write it as a linear model?

```

mod2=lm(Y~ Condition+Subj,data=dati2)

summary(mod2)

## 
## Call:
## lm(formula = Y ~ Condition + Subj, data = dati2)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -0.9656 -0.2713  0.0000  0.2713  0.9656 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.28162   0.09647 -13.285 4.56e-11 ***
## Condition1  -0.31709   0.09647  -3.287 0.003877 **  
## Subj1        -2.38228   0.42050  -5.665 1.84e-05 *** 
## Subj2        -2.97278   0.42050  -7.070 9.99e-07 *** 
## Subj3        -1.19488   0.42050  -2.842 0.010433 *   
## Subj4         1.79392   0.42050   4.266 0.000417 *** 
## Subj5         0.88487   0.42050   2.104 0.048892 *   
## Subj6        -0.68168   0.42050  -1.621 0.121470  
## Subj7        -4.14253   0.42050  -9.851 6.69e-09 *** 
## Subj8        -2.30318   0.42050  -5.477 2.77e-05 *** 
## Subj9         0.94687   0.42050   2.252 0.036361 *   
## Subj10       -2.25578   0.42050  -5.365 3.54e-05 *** 
## Subj11        1.50062   0.42050   3.569 0.002049 **  
## Subj12       -2.51503   0.42050  -5.981 9.35e-06 *** 
## Subj13        2.62622   0.42050   6.246 5.35e-06 *** 
## Subj14        1.90672   0.42050   4.534 0.000227 *** 
## Subj15        1.96557   0.42050   4.674 0.000165 *** 
## Subj16        2.88522   0.42050   6.861 1.51e-06 *** 
## Subj17       -0.51293   0.42050  -1.220 0.237463  
## Subj18       -0.99723   0.42050  -2.372 0.028436 *   
## Subj19        0.81917   0.42050   1.948 0.066330 .  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.6101 on 19 degrees of freedom
## Multiple R-squared:  0.9676, Adjusted R-squared:  0.9334 
## F-statistic: 28.35 on 20 and 19 DF,  p-value: 3.682e-10

```

```

anova(mod2)

## Analysis of Variance Table
## 
## Response: Y
##             Df  Sum Sq Mean Sq F value    Pr(>F)    
## Condition  1   4.022  4.0219  10.804  0.003877 **  
## Subj       19  207.022 10.8959  29.270 3.118e-10 *** 
## Residuals  19   7.073  0.3722 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Compare the results. (Different or the same?)

Linear models with repeated measures

Let's consider (and fit) a linear model with Chan*Condition:

```
modlmf=lm(Y~ Chan*Condition,data=dati)
anova(modlmf)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df Sum Sq Mean Sq F value Pr(>F)
## Chan          5  871.9 174.376 25.4499 <2e-16 ***
## Condition      4 1022.9 255.714 37.3209 <2e-16 ***
## Chan:Condition 20   66.6    3.328  0.4857 0.9719
## Residuals     570 3905.5    6.852
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We don't take in account the fact that measures are taken on the same subject!

Can we just add the Subj term?

```
modlmf=lm(Y~ Chan*Condition+Subj,data=dati)
anova(modlmf)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df Sum Sq Mean Sq F value Pr(>F)
## Chan          5  871.88 174.376 68.0714 <2e-16 ***
## Condition      4 1022.86 255.714 99.8233 <2e-16 ***
## Subj          19 2494.02 131.264 51.2418 <2e-16 ***
## Chan:Condition 20   66.56    3.328  1.2992 0.1724
## Residuals     551 1411.48    2.562
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: yes and no.

The estimates are ok, but we need to take care of the residuals SS in the testing step.

All the SS that we need can be found in the saturated linear model. We compute them now and we use them later.

```
modlmf=lm(Y~ Chan*Condition*Subj,data=dati)
anova(modlmf)
```

```
## Warning in anova.lm(modlmf): I test F ANOVA su un adattamento essenzialmente
## perfetto non sono affidabili
```

```

## Analysis of Variance Table
##
## Response: Y
##                               Df  Sum Sq Mean Sq F value Pr(>F)
## Chan                      5   871.88 174.376    NaN    NaN
## Condition                  4  1022.86 255.714    NaN    NaN
## Subj                      19  2494.02 131.264    NaN    NaN
## Chan:Condition              20   66.56   3.328    NaN    NaN
## Chan:Subj                   95 1017.54 10.711    NaN    NaN
## Condition:Subj              76  246.95   3.249    NaN    NaN
## Chan:Condition:Subj         380  146.99   0.387    NaN    NaN
## Residuals                   0    0.00    NaN

```

Repeated measures

```

# The standard way
mod=aov(Y~ Chan*Condition+Subj + Error(Subj/(Chan*Condition)),data=dati)
summary(mod)

```

```

##
## Error: Subj
##           Df Sum Sq Mean Sq
## Subj      19  2494   131.3
##
## Error: Subj:Chan
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Chan       5   871.9  174.38   16.28 1.42e-11 ***
## Residuals 95 1017.5   10.71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Subj:Condition
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Condition  4  1022.9  255.71    78.7 <2e-16 ***
## Residuals 76   246.9    3.25
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Subj:Chan:Condition
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Chan:Condition 20   66.56   3.328   8.604 <2e-16 ***
## Residuals     380  146.99   0.387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

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

```
library(ez)
```

```
## Warning: il pacchetto 'ez' è stato creato con R versione 4.4.2
```

```

mod=ezANOVA(dv=Y, wid=Subj, within=.(Chan,Condition), data=dati, type=3)

## Warning in log(det(U)): Si è prodotto un NaN

print(mod)

## $ANOVA
##          Effect DFn DFn      F      p p<.05      ges
## 2           Chan   5   95 16.280163 1.422895e-11      * 0.18250183
## 3       Condition   4   76 78.697466 2.998429e-26      * 0.20754506
## 4 Chan:Condition  20  380  8.604227 5.232560e-21      * 0.01675807
##
## $`Mauchly's Test for Sphericity`
##          Effect      W      p p<.05
## 2       Chan 0.03433646 3.910057e-07      *
## 3 Condition 0.06754172 4.802965e-07      *
##
## $`Sphericity Corrections`
##          Effect     GGe      p[GG] p[GG]<.05      HFe      p[HF]
## 2       Chan 0.4368229 3.441213e-06      * 0.4957490 9.287363e-07
## 3       Condition 0.4114825 4.226482e-12      * 0.4454085 6.399344e-13
## 4 Chan:Condition 0.1134660 4.452748e-04      * 0.1296611 2.121437e-04
##      p[HF]<.05
## 2      *
## 3      *
## 4      *

```

To see the relation between repeated measures and linear model, again, the Baron material is a good start. Specially see section “7.9.3 The Appropriate Error Terms”

Spend your DF in a different way!

Same number of DF, but spent in a different way

```

dati$Lateral=dati$Chan
levels(dati$Lateral)

## [1] "01"  "02"  "P7"  "P8"  "P07" "P08"

levels(dati$Lateral)[c(1,3,5)]="Left"
levels(dati$Lateral)[-1]="Right"
levels(dati$Lateral)

## [1] "Left" "Right"

contrasts(dati$Lateral) <- contr.sum(2)

dati$ChanL=dati$Chan
# https://en.wikipedia.org/wiki/Regular\_expression
# Digits: \d
(levels(dati$ChanL)=gsub("\\"d,"",levels(dati$ChanL)))

```

```

## [1] "O"   "O"   "P"   "P"   "PO"  "PO"

contrasts(dati$ChanL) <- contr.sum(3)

# The standard way
# mod=aov(Y~ ChanL*Lateral*Condition+Subj + Error(Subj/(ChanL*Lateral*Condition)),data=dati)
# summary(mod)
#
library(ez)
mod=ezANOVA(dv=Y, wid=Subj, within=.(Condition,Lateral,ChanL),data=dati,type=3)
print(mod)

## $ANOVA
##          Effect DFn DFn      F      p p<.05      ges
## 2        Condition    4    76 78.6974657 2.998429e-26    * 0.2075450604
## 3        Lateral      1    19  0.8340652 3.725436e-01    0.0070380089
## 4        ChanL       2    38 56.1729241 4.478690e-12    * 0.1749978790
## 5 Condition:Lateral    4    76  0.1443064 9.649821e-01    0.0001853800
## 6 Condition:ChanL     8   152 35.1954651 5.609013e-31    * 0.0159782333
## 7 Lateral:ChanL      2    38  2.8073524 7.292266e-02    0.0040220980
## 8 Condition:Lateral:ChanL    8   152  2.6449974 9.620701e-03    * 0.0006202045
##
## $'Mauchly's Test for Sphericity'
##          Effect      W      p p<.05
## 2        Condition 6.754172e-02 4.802965e-07    *
## 4        ChanL 6.299144e-01 1.561471e-02    *
## 5 Condition:Lateral 9.051612e-03 1.045354e-13    *
## 6 Condition:ChanL 1.397323e-05 7.887294e-21    *
## 7 Lateral:ChanL 8.087935e-01 1.480945e-01    *
## 8 Condition:Lateral:ChanL 3.462436e-05 2.626994e-18    *
##
## $'Sphericity Corrections'
##          Effect      GGe      p[GG] p[GG]<.05      HFe
## 2        Condition 0.4114825 4.226482e-12    * 0.4454085
## 4        ChanL 0.7298814 2.181290e-09    * 0.7752249
## 5 Condition:Lateral 0.3235149 7.714124e-01    0.3371918
## 6 Condition:ChanL 0.2488273 2.427683e-09    * 0.2778788
## 7 Lateral:ChanL 0.8394850 8.359370e-02    0.9115913
## 8 Condition:Lateral:ChanL 0.2410174 8.634269e-02    0.2677123
##
##      p[HF] p[HF]<.05
## 2 6.399344e-13    *
## 4 7.702179e-10    *
## 5 7.811776e-01    *
## 6 3.463383e-10    *
## 7 7.863060e-02    *
## 8 7.969075e-02    *

ezPlot(dv=Y, wid=Subj, within=.(ChanL,Lateral,Condition),data=dati,
       x=Condition,split=Lateral,row=ChanL)

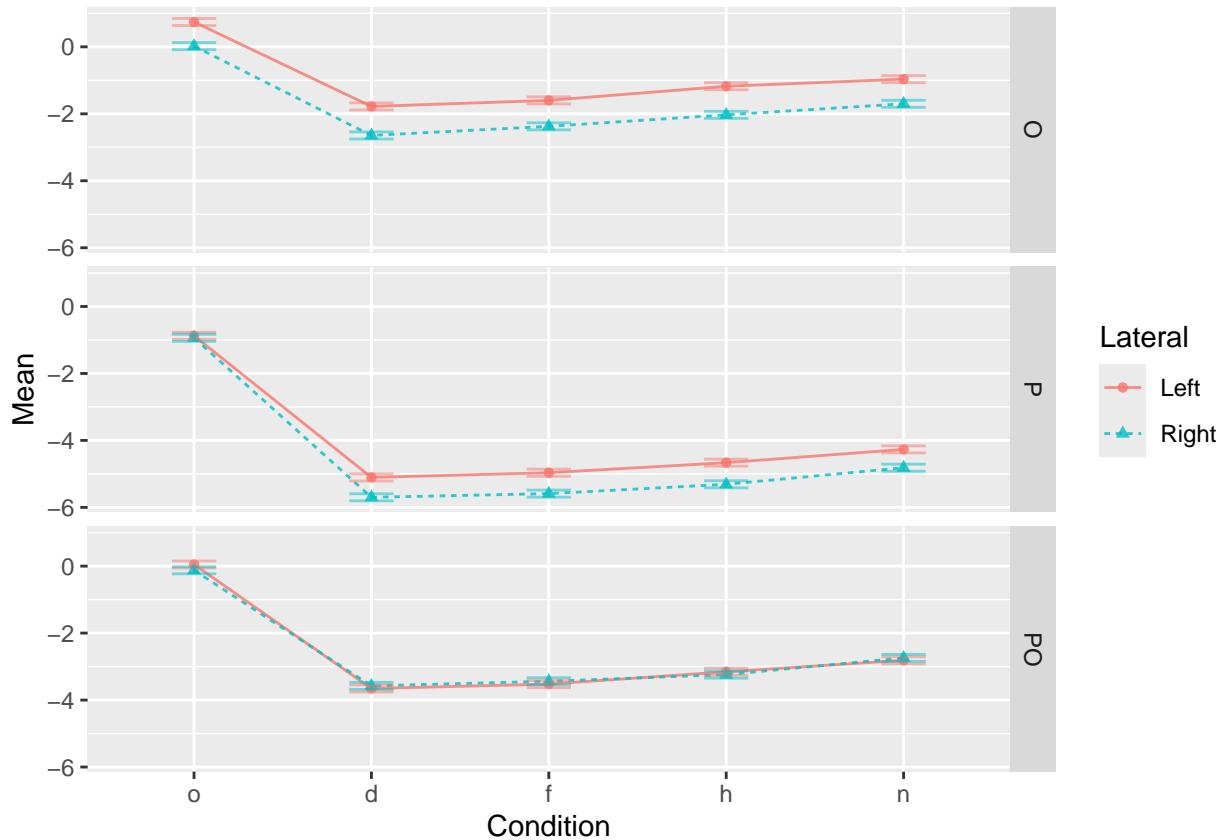
## Warning: The 'facets' argument of 'facet_grid()' is deprecated as of ggplot2 2.2.0.

```

```

## i Please use the 'rows' argument instead.
## i The deprecated feature was likely used in the ez package.
## Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```



Sphericity

Sphericity is an assumption about the structure of the covariance matrix in a repeated measures design. Before we describe it, 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.

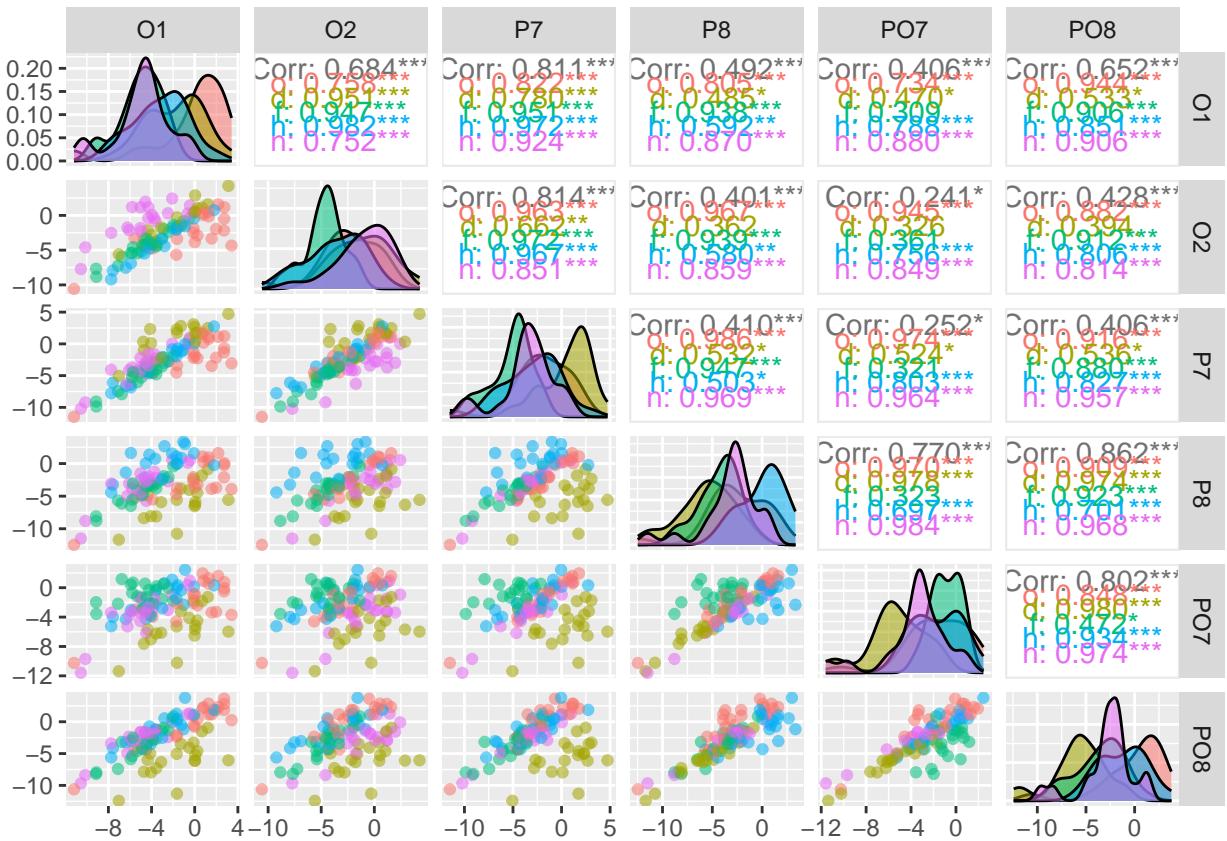
Compound symmetry is met when the correlation between Condition f and Condition h is equal to the correlation between Condition f and Condition o or Condition h and Condition n, etc (same for any other factor within subject, such as Chan). But a more direct way to think about compound symmetry is to say that it requires that all subjects in each group change in the same way over conditions/levels. 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.

Is compound symmetry met in our data?

```
# install.packages("GGally")
library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg    ggplot2

Y=matrix(dat1$Y,byrow = TRUE,nrow = 20*5)
Y=data.frame(Y)
names(Y)=levels(dat1$Chan)
ggpairs(Y,aes(colour = dat1$Condition[1:100], alpha = 0.4))
```



Not really! (correlations do often differ)

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

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 (e.g. subjects/condiction/channel cells)
- It only handle factors, no quantitative variables

Mixed model is a more flexible approach.

Mixed models

Motivation/Introduction

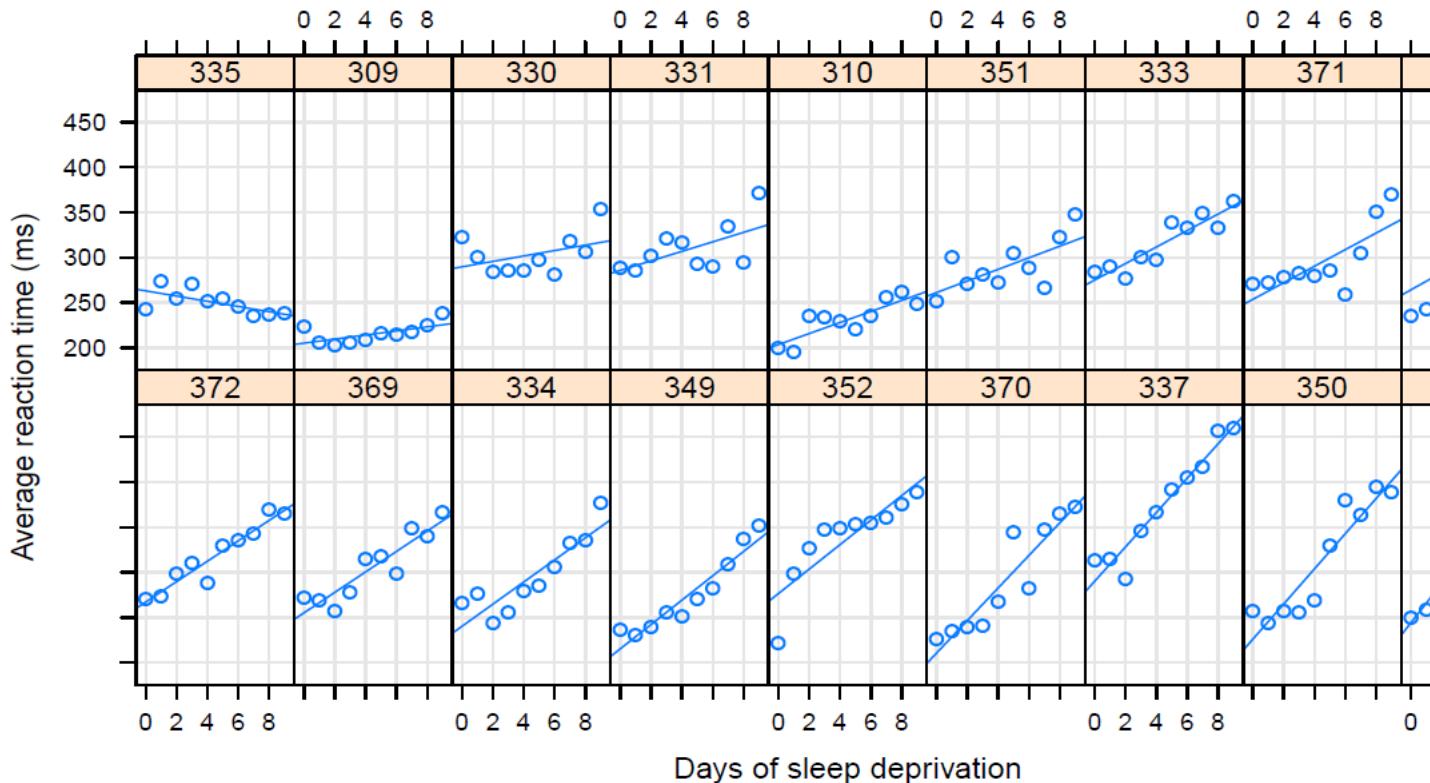
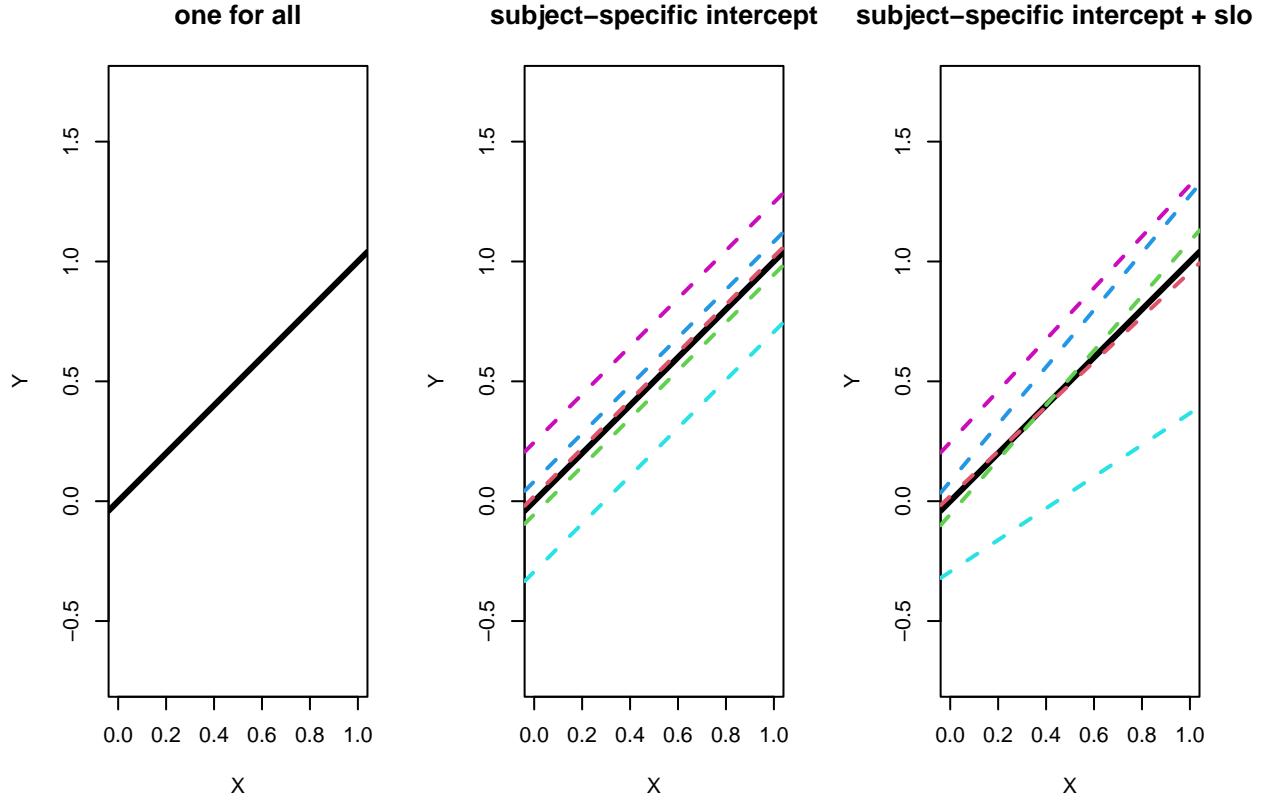


Figure 1: Average reaction time versus days of sleep deprivation by subject. Subjects (from left to right starting on the top row) by increasing slope of subject-specific regressions.



Mixed models allow for more flexible modelization.

I assume you are expert on mixed models, if not https://en.wikipedia.org/wiki/Mixed_model
and much more on: http://webcom.upmf-grenoble.fr/LIP/Perso/DMuller/M2R/R_et_Mixed/documents/Bates-book.pdf
and
<https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf>

The model

Models with random effects can be defined as:

$$Y_{n \times 1} = X_{n \times p} B_{p \times 1} + Z_{n \times q} b_{q \times 1} + \varepsilon_{n \times 1}$$

where

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$$

In the models we will consider, the random effects are modeled as a multivariate normal random variable:

$$b \sim \mathcal{N}(0, \Sigma_{q \times q}),$$

In a *linear mixed model* the conditional distribution $(Y | \mathcal{B} = b)$ is a *spherical* multivariate Gaussian.

In our case $n = \#Subjects \times \#Conditions \times \#Channels = 20 \times 5 \times 6 = 600$. X is the matrix of (dummified) predictors. Z can take many dimensions and values. Examples follow.

Random Intercept (for each Subject)

Z is the matrix of dummy variables of the column `dati$Subj`.

```

library(lmerTest)

## Warning: il pacchetto 'lmerTest' è stato creato con R versione 4.4.2

## Warning: il pacchetto 'Matrix' è stato creato con R versione 4.4.2

# library(lme4)
contrasts(dati$Lateral)=contr.sum
contrasts(dati$ChanL)=contr.sum
contrasts(dati$Condition)=contr.sum

mod=lmer(Y~ Condition*Lateral*ChanL +(1|Subj),data=dati)

car:::Anova(mod)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Y
##              Chisq Df Pr(>Chisq)
## Condition      399.2932  4   < 2.2e-16 ***
## Lateral        10.8062  1    0.001012 **
## ChanL          323.3939  2   < 2.2e-16 ***
## Condition:Lateral 0.2827  4    0.990905
## Condition:ChanL  24.7559  8    0.001710 **
## Lateral:ChanL   6.1568  2    0.046032 *
## Condition:Lateral:ChanL 0.9461  8    0.998566
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Random Channel (for each Subject)

Actually, instead of `Channel`, we use the combination of `ChanL*Lateral`. Same prediction ability (6 channels in Channel and 3X2 combination of ChanL and Lateral), just a different point of view.

`Z` is the matrix of dummy variables of the column `dati$Chan`.

```

contrasts(dati$Chan)<- contr.treatment
mod2=lmer(Y~ 1+Lateral*ChanL*Condition +(0+Chan|Subj),data=dati)
summary(mod2)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [ 
## lmerModLmerTest]
## Formula: Y ~ 1 + Lateral * ChanL * Condition + (0 + Chan | Subj)
## Data: dati
## 
## REML criterion at convergence: 1937.2
## 
## Scaled residuals:
##     Min      1Q  Median      3Q     Max 
## -4.1900 -0.4578 -0.0040  0.5130  3.5464 
## 
## Random effects:

```

```

## Groups   Name    Variance Std.Dev. Corr
## Subj    Chan01  5.1156  2.2618
##          Chan02  8.6139  2.9349  0.73
##          ChanP7  2.6902  1.6402  0.74 0.50
##          ChanP8  6.3837  2.5266  0.55 0.80 0.53
##          ChanP07 5.2647  2.2945  0.87 0.54 0.95 0.56
##          ChanP08 7.9345  2.8168  0.62 0.90 0.49 0.94 0.57
## Residual      0.8518  0.9229
## Number of obs: 600, groups: Subj, 20
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                -2.731516  0.467756 18.998780 -5.840 1.26e-05
## Lateral1                   0.214794  0.235212 19.000279  0.913 0.37258
## ChanL1                     1.381018  0.191777 18.998047  7.201 7.71e-07
## ChanL2                     -1.490922  0.161583 19.023042 -9.227 1.87e-08
## Condition1                 2.544305  0.075355 475.003497 33.764 < 2e-16
## Condition2                 -1.011749  0.075355 475.003497 -13.426 < 2e-16
## Condition3                 -0.849248  0.075355 475.003497 -11.270 < 2e-16
## Condition4                 -0.531397  0.075355 475.003497 -7.052 6.26e-12
## Lateral1:ChanL1            0.180370  0.110031 19.146198  1.639 0.11748
## Lateral1:ChanL2            0.032409  0.105115 19.236351  0.308 0.76115
## Lateral1:Condition1        -0.056016  0.075355 475.003497 -0.743 0.45763
## Lateral1:Condition2        0.015606  0.075355 475.003497  0.207 0.83602
## Lateral1:Condition3        0.004035  0.075355 475.003497  0.054 0.95731
## Lateral1:Condition4        0.049614  0.075355 475.003497  0.658 0.51060
## ChanL1:Condition1           -0.811645 0.106568 475.003497 -7.616 1.42e-13
## ChanL2:Condition1           0.771993 0.106568 475.003497  7.244 1.77e-12
## ChanL1:Condition2           0.149199 0.106568 475.003497  1.400 0.16215
## ChanL2:Condition2           -0.167232 0.106568 475.003497 -1.569 0.11725
## ChanL1:Condition3           0.213801 0.106568 475.003497  2.006 0.04540
## ChanL2:Condition3           -0.204413 0.106568 475.003497 -1.918 0.05569
## ChanL1:Condition4           0.278674 0.106568 475.003497  2.615 0.00921
## ChanL2:Condition4           -0.233477 0.106568 475.003497 -2.191 0.02895
## Lateral1:ChanL1:Condition1  0.020815 0.106568 475.003497  0.195 0.84523
## Lateral1:ChanL2:Condition1  -0.160961 0.106568 475.003497 -1.510 0.13160
## Lateral1:ChanL1:Condition2  0.022442 0.106568 475.003497  0.211 0.83330
## Lateral1:ChanL2:Condition2  0.034156 0.106568 475.003497  0.321 0.74872
## Lateral1:ChanL1:Condition3  -0.011970 0.106568 475.003497 -0.112 0.91062
## Lateral1:ChanL2:Condition3  0.060872 0.106568 475.003497  0.571 0.56813
## Lateral1:ChanL1:Condition4  -0.017273 0.106568 475.003497 -0.162 0.87131
## Lateral1:ChanL2:Condition4  0.026281 0.106568 475.003497  0.247 0.80531
##
## (Intercept) ***
## Lateral1
## ChanL1 ***
## ChanL2 ***
## Condition1 ***
## Condition2 ***
## Condition3 ***
## Condition4 ***
## Lateral1:ChanL1
## Lateral1:ChanL2
## Lateral1:Condition1

```

```

## Lateral1:Condition2
## Lateral1:Condition3
## Lateral1:Condition4
## ChanL1:Condition1      ***
## ChanL2:Condition1      ***
## ChanL1:Condition2
## ChanL2:Condition2
## ChanL1:Condition3      *
## ChanL2:Condition3      .
## ChanL1:Condition4      **
## ChanL2:Condition4      *
## Lateral1:ChanL1:Condition1
## Lateral1:ChanL2:Condition1
## Lateral1:ChanL1:Condition2
## Lateral1:ChanL2:Condition2
## Lateral1:ChanL1:Condition3
## Lateral1:ChanL2:Condition3
## Lateral1:ChanL1:Condition4
## Lateral1:ChanL2:Condition4
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

##
## Correlation matrix not shown by default, as p = 30 > 12.
## Use print(x, correlation=TRUE)  or
##      vcov(x)      if you need it

```

```
car::Anova(mod2)
```

```

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Y
##              Chisq Df Pr(>Chisq)
## Lateral       6.0188  1   0.01415 *
## ChanL        81.2568  2   < 2.2e-16 ***
## Condition    1200.8872  4   < 2.2e-16 ***
## Lateral:ChanL    7.7571  2   0.02068 *
## Lateral:Condition   0.8502  4   0.93160
## ChanL:Condition   74.4541  8   6.344e-13 ***
## Lateral:ChanL:Condition 2.8456  8   0.94367
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
# More flexible, but harder to fit (note that independence among random effects is imposed):
# mod3=lmer(Y~ Condition*ChanL+Lateral+(0+Lateral/Subj)+(0+Condition/Subj)+(0+ChanL/Subj), data=dati)
#
```

NOTE the warning message. Not a good sign, actually. We don't discuss it in this lab. However, one should either, find out a different algorithm or change the model. Remember that the results of a model's fit that doesn't converge, can not be trusted!

Random Emisphere (for each Subject)

A simplified model may be based on Left/Right random effect for each subject.

Z is the matrix of 2 dummy variables from the column `dati$Lateral` (intercept is not included in the random part of the model).

```
mod3=lmer(Y~ Lateral*ChanL*Condition +(1+Lateral|Subj),data=dati)
summary(mod3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Y ~ Lateral * ChanL * Condition + (1 + Lateral | Subj)
## Data: dati
##
## REML criterion at convergence: 2130.3
##
## Scaled residuals:
##      Min    1Q Median    3Q   Max
## -2.96741 -0.57945  0.03594  0.60345  2.56829
##
## Random effects:
## Groups   Name        Variance Std.Dev. Corr
## Subj     (Intercept) 4.327    2.080
##          Lateral1    1.057    1.028   -0.37
## Residual           1.468    1.212
## Number of obs: 600, groups: Subj, 20
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                -2.731516  0.467735 18.999757 -5.840 1.26e-05
## Lateral1                   0.214794  0.235191 19.000133  0.913  0.3725
## ChanL1                      1.381018  0.069948 532.000111 19.743 < 2e-16
## ChanL2                     -1.490922  0.069948 532.000111 -21.315 < 2e-16
## Condition1                 2.544305  0.098922 532.000111 25.720 < 2e-16
## Condition2                 -1.011749  0.098922 532.000111 -10.228 < 2e-16
## Condition3                 -0.849248  0.098922 532.000111 -8.585 < 2e-16
## Condition4                 -0.531397  0.098922 532.000111 -5.372 1.17e-07
## Lateral1:ChanL1            0.180370  0.069948 532.000111  2.579  0.0102
## Lateral1:ChanL2            0.032409  0.069948 532.000111  0.463  0.6433
## Lateral1:Condition1       -0.056016  0.098922 532.000111 -0.566  0.5715
## Lateral1:Condition2       0.015606  0.098922 532.000111  0.158  0.8747
## Lateral1:Condition3       0.004035  0.098922 532.000111  0.041  0.9675
## Lateral1:Condition4       0.049614  0.098922 532.000111  0.502  0.6162
## ChanL1:Condition1         -0.811645  0.139897 532.000111 -5.802 1.13e-08
## ChanL2:Condition1         0.771993  0.139897 532.000111  5.518 5.35e-08
## ChanL1:Condition2         0.149199  0.139897 532.000111  1.066  0.2867
## ChanL2:Condition2         -0.167232  0.139897 532.000111 -1.195  0.2325
## ChanL1:Condition3         0.213801  0.139897 532.000111  1.528  0.1270
## ChanL2:Condition3         -0.204413  0.139897 532.000111 -1.461  0.1446
## ChanL1:Condition4         0.278674  0.139897 532.000111  1.992  0.0469
## ChanL2:Condition4         -0.233477  0.139897 532.000111 -1.669  0.0957
## Lateral1:ChanL1:Condition1 0.020815  0.139897 532.000111  0.149  0.8818
## Lateral1:ChanL2:Condition1 -0.160961  0.139897 532.000111 -1.151  0.2504
## Lateral1:ChanL1:Condition2 0.022442  0.139897 532.000111  0.160  0.8726
## Lateral1:ChanL2:Condition2 0.034156  0.139897 532.000111  0.244  0.8072
## Lateral1:ChanL1:Condition3 -0.011970  0.139897 532.000111 -0.086  0.9318
## Lateral1:ChanL2:Condition3  0.060872  0.139897 532.000111  0.435  0.6637
```

```

## Lateral1:ChanL1:Condition4 -0.017273  0.139897 532.000111 -0.123  0.9018
## Lateral1:ChanL2:Condition4  0.026281  0.139897 532.000111  0.188  0.8511
##
## (Intercept) ***
## Lateral1
## ChanL1 ***
## ChanL2 ***
## Condition1 ***
## Condition2 ***
## Condition3 ***
## Condition4 ***
## Lateral1:ChanL1 *
## Lateral1:ChanL2
## Lateral1:Condition1
## Lateral1:Condition2
## Lateral1:Condition3
## Lateral1:Condition4
## ChanL1:Condition1 ***
## ChanL2:Condition1 ***
## ChanL1:Condition2
## ChanL2:Condition2
## ChanL1:Condition3
## ChanL2:Condition3
## ChanL1:Condition4 *
## ChanL2:Condition4 .
## Lateral1:ChanL1:Condition1
## Lateral1:ChanL2:Condition1
## Lateral1:ChanL1:Condition2
## Lateral1:ChanL2:Condition2
## Lateral1:ChanL1:Condition3
## Lateral1:ChanL2:Condition3
## Lateral1:ChanL1:Condition4
## Lateral1:ChanL2:Condition4
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

##
## Correlation matrix not shown by default, as p = 30 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)      if you need it

```

```
car::Anova(mod3)
```

```

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Y
##              Chisq Df Pr(>Chisq)
## Lateral       0.8341  1   0.361099
## ChanL        564.3879  2   < 2.2e-16 ***
## Condition    696.8476  4   < 2.2e-16 ***
## Lateral:ChanL 10.7449  2   0.004643 **
## Lateral:Condition 0.4933  4   0.974146
## ChanL:Condition 43.2040  8   8.04e-07 ***

```

```

## Lateral:ChanL:Condition    1.6512   8   0.989903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Plotting tools

for the first model:

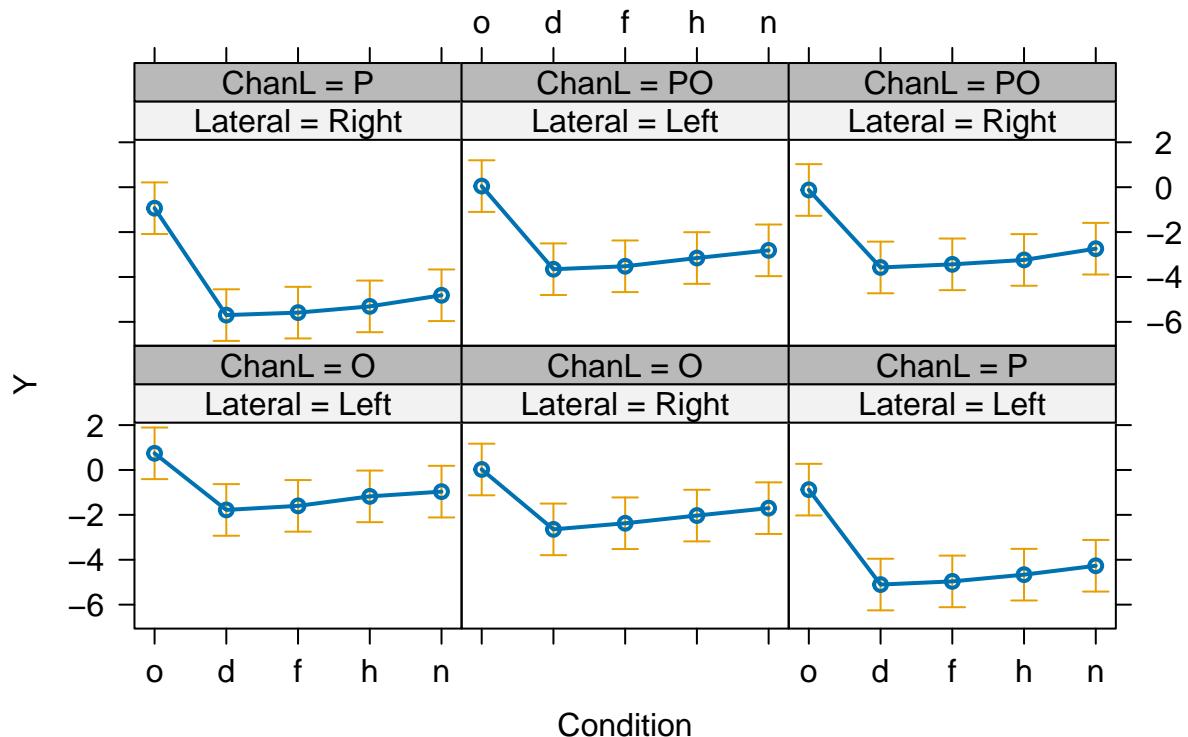
```
library(effects)
```

```
## Warning: il pacchetto 'effects' è stato creato con R versione 4.4.2
```

```
## Warning: il pacchetto 'carData' è stato creato con R versione 4.4.2
```

```
plot(allEffects(mod))
```

Condition*Lateral*ChanL effect plot

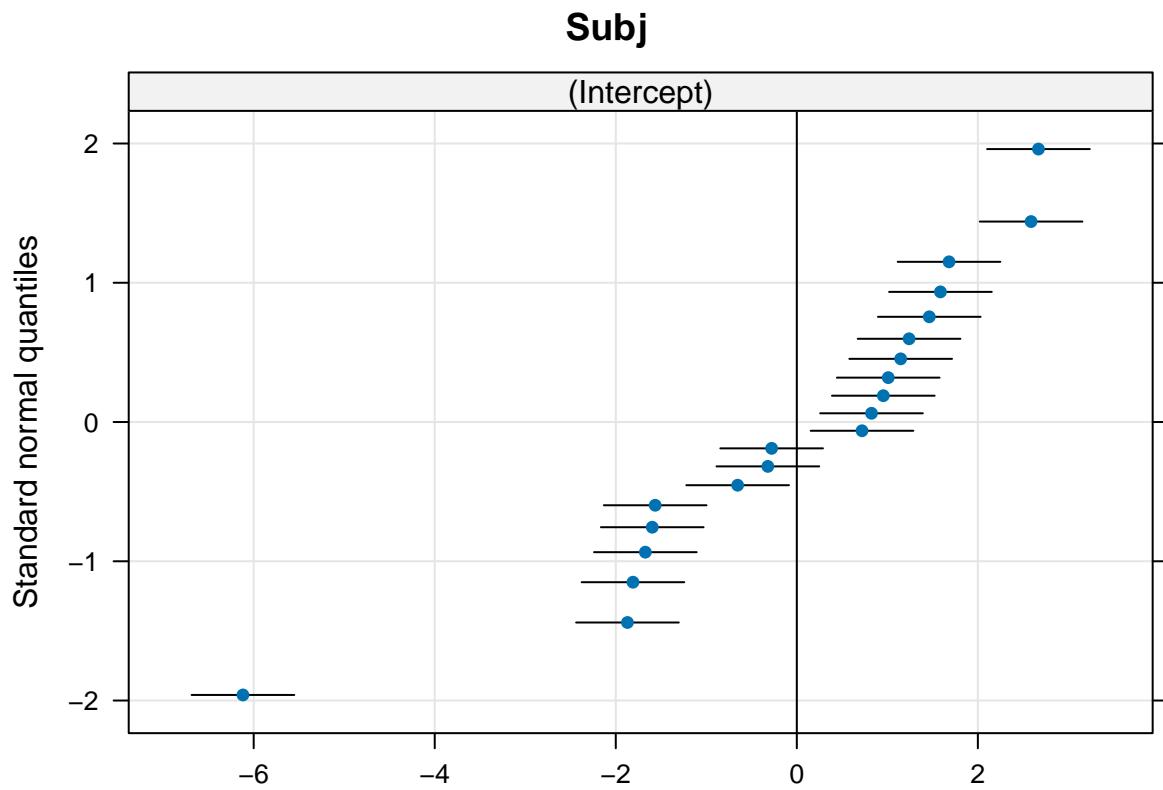


```

#plot random effects:
require(lattice)
qqmath(ranef(mod, condVar=TRUE))

```

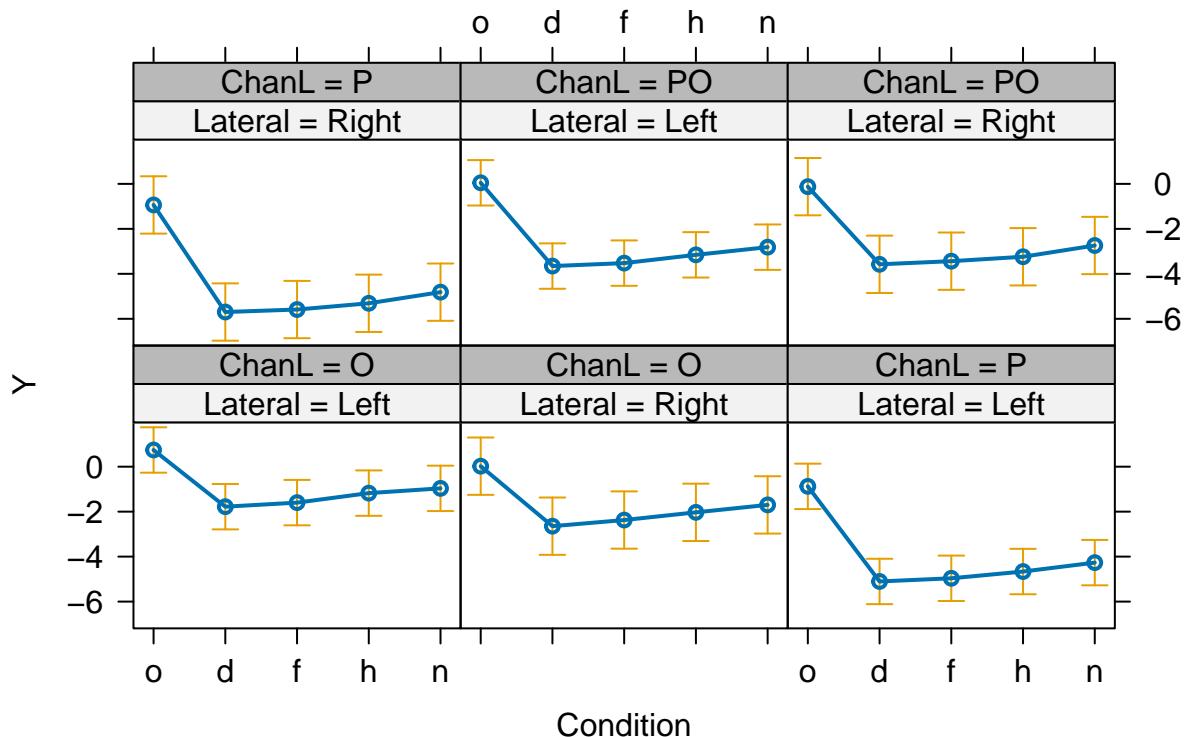
```
## $Subj
```



The second model:

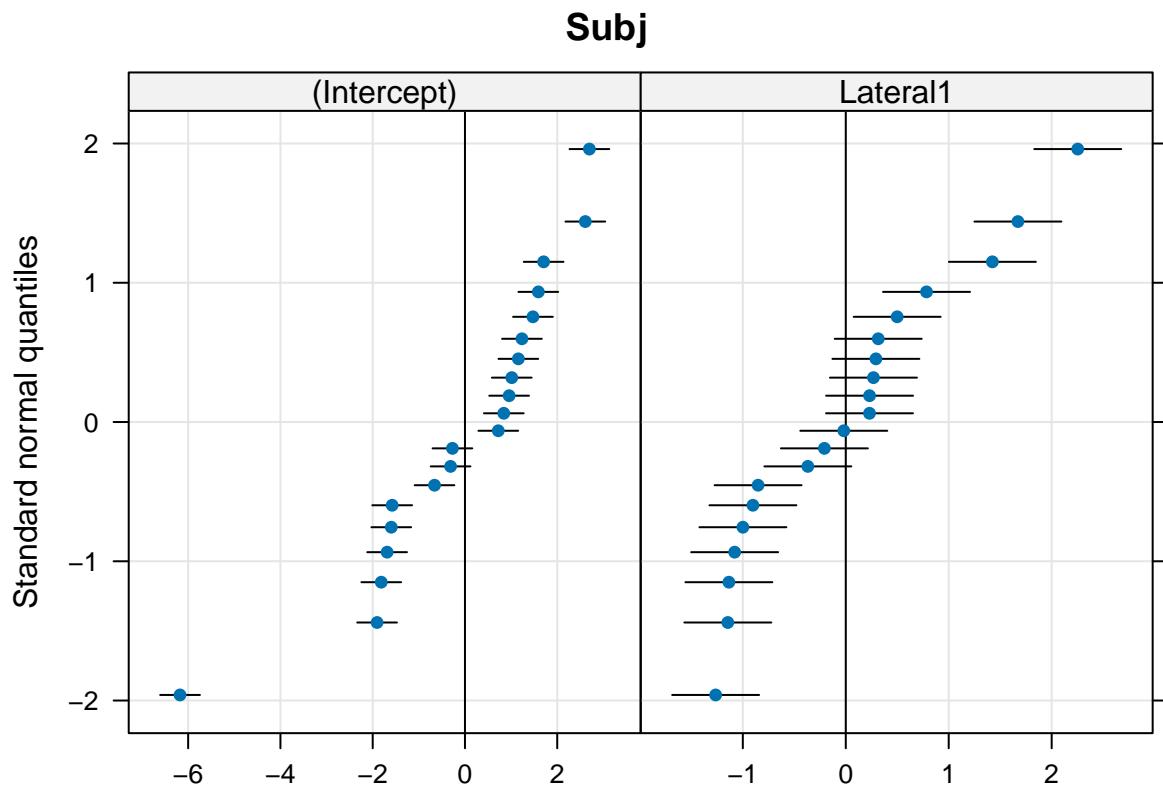
```
library(effects)
plot(allEffects(mod3))
```

Lateral*ChanL*Condition effect plot

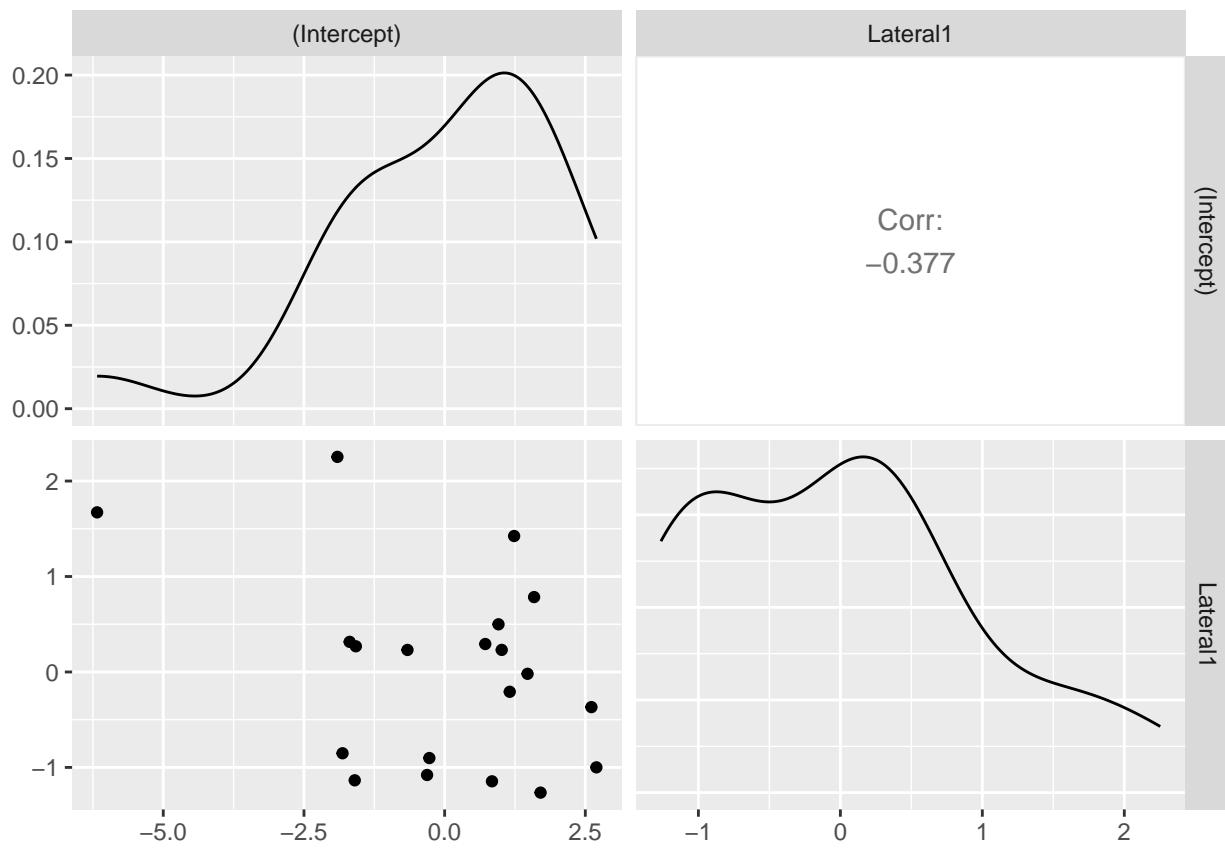


```
#plot random effects:
require(lattice)
qqmath(ranef(mod3, condVar=TRUE))
```

```
## $Subj
```



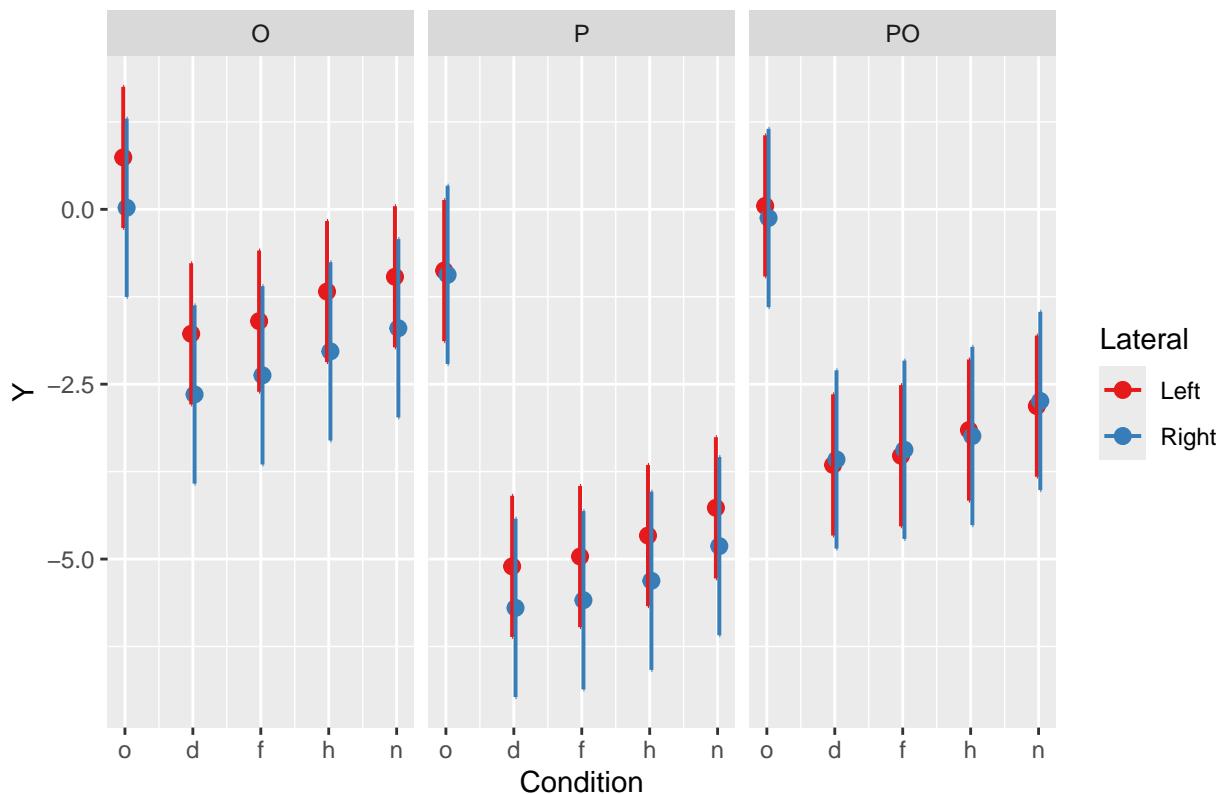
```
# scatter plot
ggpairs(ranef(mod3, condVar=TRUE)$Subj)
```



An alternative plotting tool:

```
library(sjPlot)
library(ggplot2)
plot_model(mod3, type = "pred", terms = c("Condition", "Lateral", "ChanL"))
```

Predicted values of Y



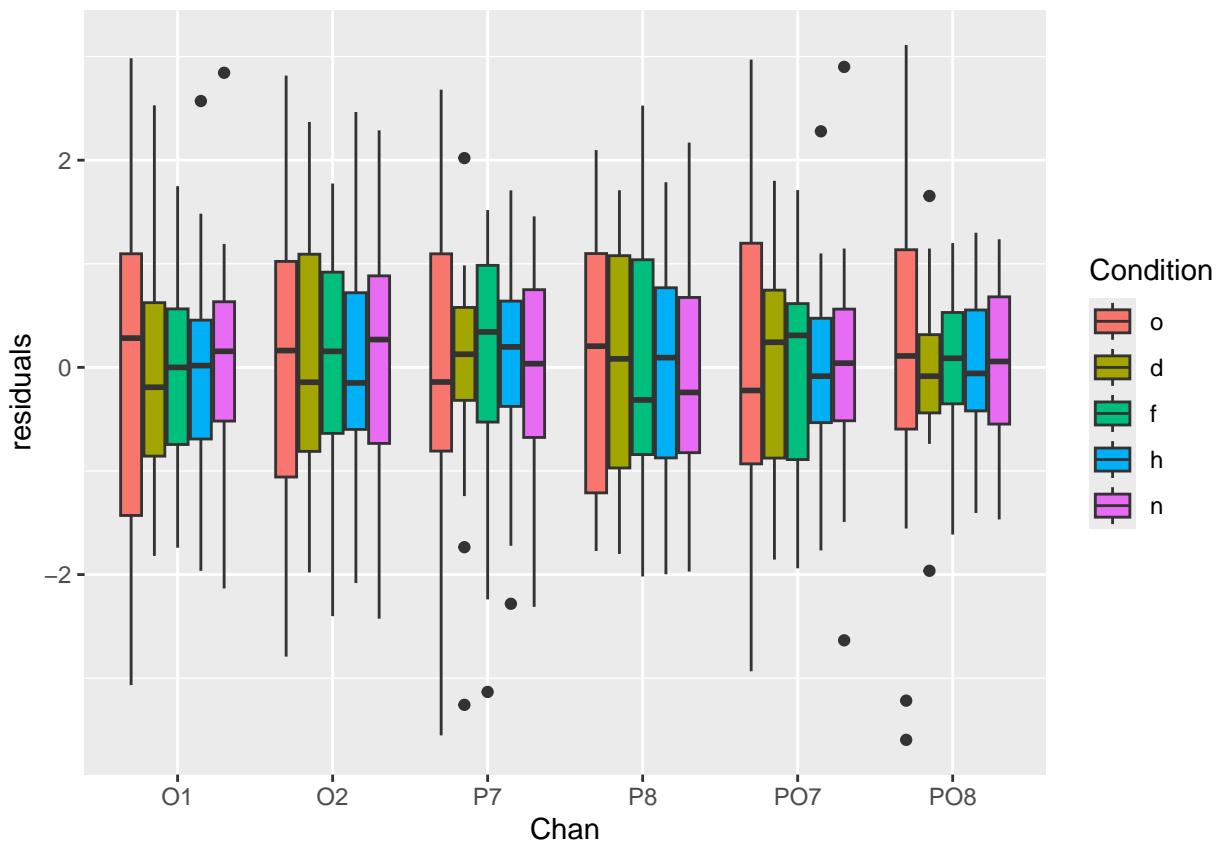
Validity of the assumptions

- Independence of the residuals?
- Normality of the residuals?
- Homoscedasticity of the residuals (i.e. same variance between subject/channel/condition)?
- outliers?
- Leverage? (influential observations)

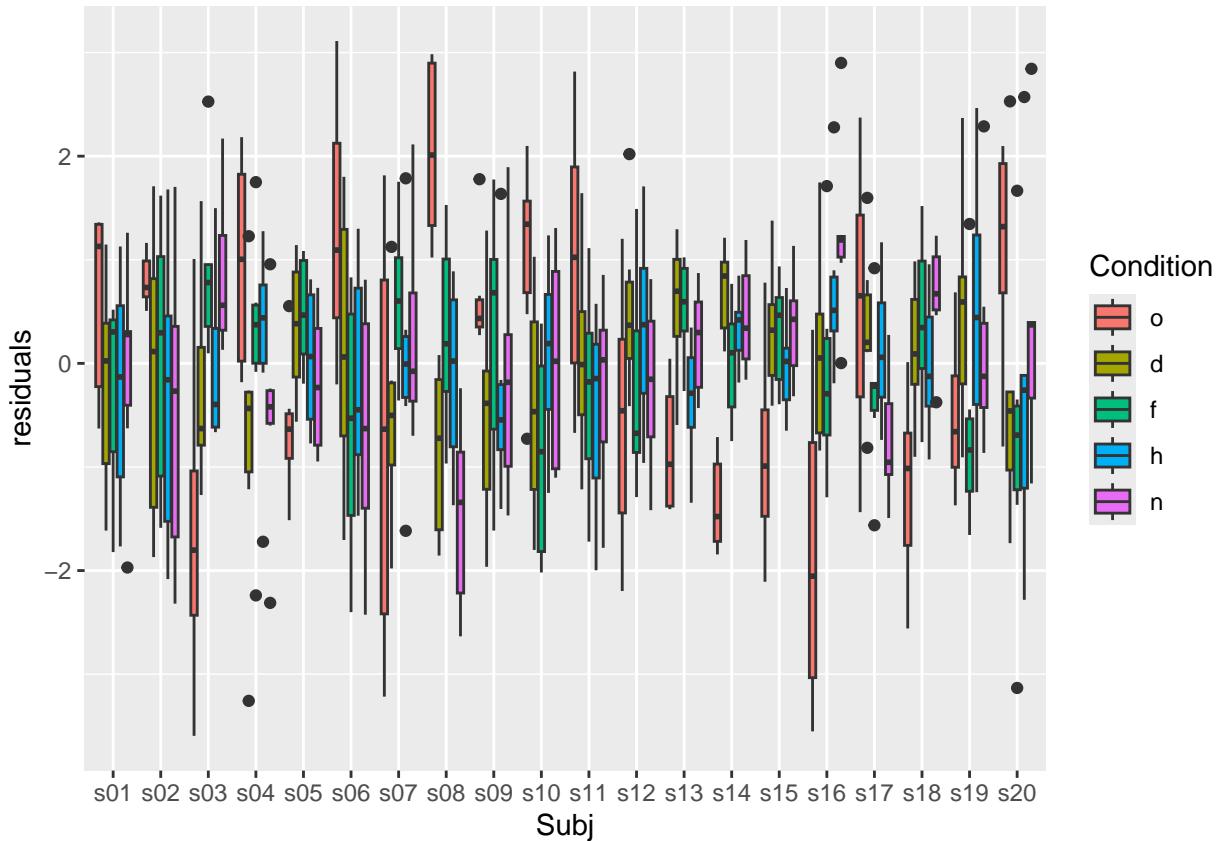
Please, do not test for normality, for homoscedasticity, sphericity etc.

Use Exploratory data Analysis, instead!

```
dati$residuals=residuals(mod3)
p <- ggplot(dati, aes(x=Chan, y=residuals, fill=Condition)) +
  geom_boxplot()
```



```
p <- ggplot(dati, aes(x=Subj, y=residuals, fill=Condition)) + geom_boxplot()
p
```



Contrasts and post-hoc

Post-hoc

```
library(multcomp)

## Warning: il pacchetto 'multcomp' è stato creato con R versione 4.4.2

## Caricamento del pacchetto richiesto: mvtnorm

## Warning: il pacchetto 'mvtnorm' è stato creato con R versione 4.4.2

## Caricamento del pacchetto richiesto: survival

## Warning: il pacchetto 'survival' è stato creato con R versione 4.4.2

## Caricamento del pacchetto richiesto: TH.data

## Warning: il pacchetto 'TH.data' è stato creato con R versione 4.4.2

## Caricamento del pacchetto richiesto: MASS
```

```

## Warning: il pacchetto 'MASS' è stato creato con R versione 4.4.2

##
## Caricamento pacchetto: 'TH.data'

## Il seguente oggetto è mascherato da 'package:MASS':
##
##     geyser

summary(glht(mod3, linfct = mcp(Condition = "Tukey")))

```

```

## Warning in mcp2matrix(model, linfct = linfct): covariate interactions found --
## default contrast might be inappropriate

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = Y ~ Lateral * ChanL * Condition + (1 + Lateral |
##         Subj), data = dati)
##
## Linear Hypotheses:
##             Estimate Std. Error z value Pr(>|z|)
## d - o == 0 -3.5561    0.1564 -22.736 <0.001 ***
## f - o == 0 -3.3936    0.1564 -21.697 <0.001 ***
## h - o == 0 -3.0757    0.1564 -19.664 <0.001 ***
## n - o == 0 -2.6962    0.1564 -17.238 <0.001 ***
## f - d == 0  0.1625    0.1564   1.039  0.8373
## h - d == 0  0.4804    0.1564   3.071  0.0182 *
## n - d == 0  0.8598    0.1564   5.497 <0.001 ***
## h - f == 0  0.3179    0.1564   2.032  0.2505
## n - f == 0  0.6973    0.1564   4.458 <0.001 ***
## n - h == 0  0.3795    0.1564   2.426  0.1083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

```

Custom contrasts

An example:

- neutral vs object in O1 (left)
- disgust vs neutral in O1 (left)
- fear vs neutral in O1 (left)
- happy vs neutral in O1 (left)

```

library(multcomp)
ncoeff=length(coefficients(mod3)[[1]])
contr <- rbind("n - o" = c(0,0,0,0,-1, 0, 0, 0, rep(0,ncoeff-8)),

```

```

    "d - n" = c(0,0,0,0, 0, 1, 0, 0, rep(0,ncoeff-8)),
    "f - n" = c(0,0,0,0, 0, 0, 1, 0, rep(0,ncoeff-8)),
    "h - n" = c(0,0,0,0, 0, 0, 0, 1, rep(0,ncoeff-8)))
compa= glht(mod3, linfct = contr)
summary(compa, test = adjusted("none"))

##  

##   Simultaneous Tests for General Linear Hypotheses  

##  

## Fit: lmer(formula = Y ~ Lateral * ChanL * Condition + (1 + Lateral |  

##       Subj), data = dati)  

##  

## Linear Hypotheses:  

##             Estimate Std. Error z value Pr(>|z|)  

## n - o == 0 -2.54431   0.09892 -25.720 < 2e-16 ***  

## d - n == 0 -1.01175   0.09892 -10.228 < 2e-16 ***  

## f - n == 0 -0.84925   0.09892  -8.585 < 2e-16 ***  

## h - n == 0 -0.53140   0.09892  -5.372 7.79e-08 ***  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

## (Adjusted p values reported -- none method)

# with multiple comparisons
summary(compa)

##  

##   Simultaneous Tests for General Linear Hypotheses  

##  

## Fit: lmer(formula = Y ~ Lateral * ChanL * Condition + (1 + Lateral |  

##       Subj), data = dati)  

##  

## Linear Hypotheses:  

##             Estimate Std. Error z value Pr(>|z|)  

## n - o == 0 -2.54431   0.09892 -25.720 < 1e-07 ***  

## d - n == 0 -1.01175   0.09892 -10.228 < 1e-07 ***  

## f - n == 0 -0.84925   0.09892  -8.585 < 1e-07 ***  

## h - n == 0 -0.53140   0.09892  -5.372 2.62e-07 ***  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

## (Adjusted p values reported -- single-step method)

```

Multivariate ANOVA (MANOVA)

Motivation

Hei, wait a moment... the trials for `object` condition are much more than any other condition, the variance of its estimated component must be (much?) lower, homoschedasticity doesn't hold!!

Let's use a different approach Reshape the data from `long` to `wide` format.

to simplify the example, let's consider the comparison between conditions `neutral` vs `object`.

Reshaping the data

Let's now compute the vectors of contrasts (one vector of each channel, length equal to number of subjects): Happy vs Neutral

```
Y=matrix(dati$Y,byrow = TRUE,nrow = 20)
colnames(Y)=paste(dati$Condition,dati$ChanL,dati$Lateral,sep = "_")[1:30]
```

```
colnames(Y)
```

```
## [1] "f_P_Left"    "h_P_Left"    "d_P_Left"    "n_P_Left"    "o_P_Left"
## [6] "f_PO_Left"   "h_PO_Left"   "d_PO_Left"   "n_PO_Left"   "o_PO_Left"
## [11] "f_O_Left"    "h_O_Left"    "d_O_Left"    "n_O_Left"    "o_O_Left"
## [16] "f_P_Right"   "h_P_Right"   "d_P_Right"   "n_P_Right"   "o_P_Right"
## [21] "f_PO_Right"  "h_PO_Right"  "d_PO_Right"  "n_PO_Right"  "o_PO_Right"
## [26] "f_O_Right"   "h_O_Right"   "d_O_Right"   "n_O_Right"   "o_O_Right"
```

```
contr=matrix(0,30,6)
contr[c(2,4),1]=c(1,-1)
contr[c(2,4)+5,2]=c(1,-1)
contr[c(2,4)+10,3]=c(1,-1)
contr[c(2,4)+15,4]=c(1,-1)
contr[c(2,4)+20,5]=c(1,-1)
contr[c(2,4)+25,6]=c(1,-1)
```

```
dim(contr)
```

```
## [1] 30 6
```

```
head(contr)
```

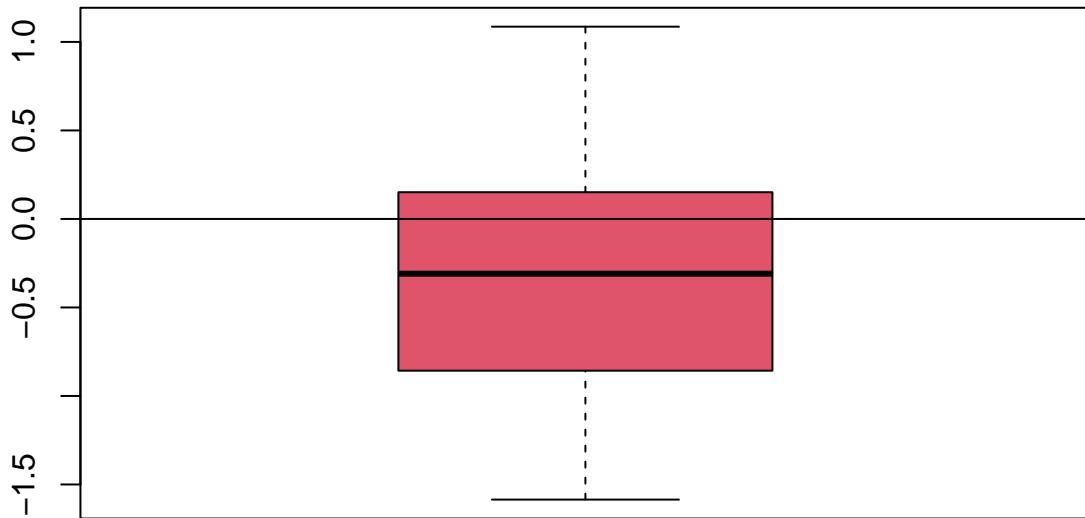
```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]     0     0     0     0     0     0
## [2,]     1     0     0     0     0     0
## [3,]     0     0     0     0     0     0
## [4,]    -1     0     0     0     0     0
## [5,]     0     0     0     0     0     0
## [6,]     0     0     0     0     0     0
```

```
Yhn=Y%*%contr
colnames(Yhn)= levels(dati$Chan)
dim(Yhn)
```

```
## [1] 20 6
```

What we see in O1?

```
boxplot(Yhn[,1],col=2)
abline(0,0)
```



Same test as above, but under a different model

```
t.test(Yhn[, 1])
```

```
##
##  One Sample t-test
##
## data: Yhn[, 1]
## t = -2.4493, df = 19, p-value = 0.02418
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.73552954 -0.05769046
## sample estimates:
## mean of x
## -0.39661
```

We can run the analysis over all channels

```
(uni_t=apply(Yhn, 2, t.test))
```

```
## $01
##
##  One Sample t-test
##
## data: newX[, i]
## t = -2.4493, df = 19, p-value = 0.02418
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.73552954 -0.05769046
## sample estimates:
## mean of x
## -0.39661
##
## 
## $02
##
```

```

## One Sample t-test
##
## data: newX[, i]
## t = -2.3752, df = 19, p-value = 0.02822
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.64355774 -0.04064226
## sample estimates:
## mean of x
## -0.3421
##
##
## $P7
##
## One Sample t-test
##
## data: newX[, i]
## t = -1.4718, df = 19, p-value = 0.1574
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.51150297 0.08913297
## sample estimates:
## mean of x
## -0.211185
##
##
## $P8
##
## One Sample t-test
##
## data: newX[, i]
## t = -3.675, df = 19, p-value = 0.001609
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.7778167 -0.2133333
## sample estimates:
## mean of x
## -0.495575
##
##
## $P07
##
## One Sample t-test
##
## data: newX[, i]
## t = -2.9811, df = 19, p-value = 0.007676
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.8528602 -0.1492698
## sample estimates:
## mean of x
## -0.501065
##
##

```

```

## $P08
##
## One Sample t-test
##
## data: newX[, i]
## t = -2.1369, df = 19, p-value = 0.04583
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.653973307 -0.006776693
## sample estimates:
## mean of x
## -0.330375

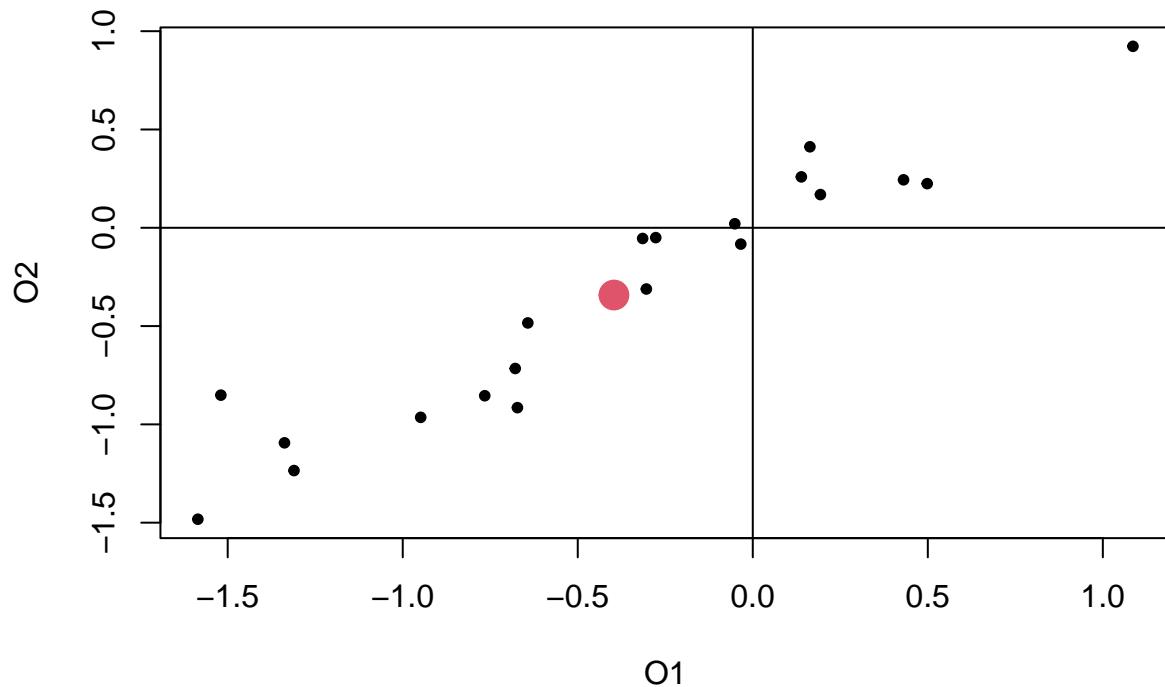
```

Manova

```

plot(Yhn[, 1:2], pch=20)
abline(v=0)
abline(h=0)
points(mean(Yhn[, 1]), mean(Yhn[, 2]), cex=3, col=2, pch=20)

```



Manova test, overall among all channels:

H_0 neutral=object in ANY of the channels. https://en.wikipedia.org/wiki/Multivariate_analysis_of_variance https://en.wikipedia.org/wiki/Hotelling%27s_T-squared_distribution

```

modman <- manova(Yhn ~ 1)
anova(modman)

```

```

## Analysis of Variance Table
##

```

```

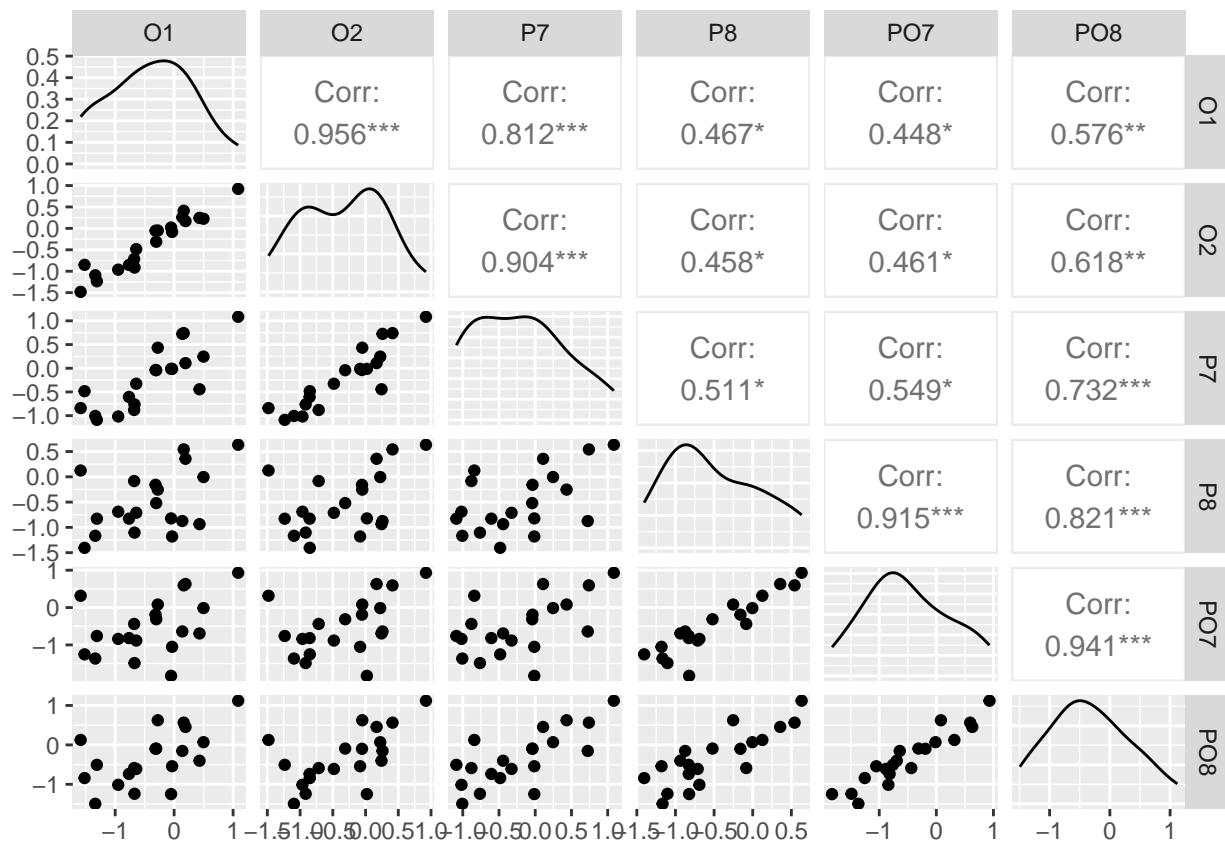
##          Df Pillai approx F num Df den Df Pr(>F)
## (Intercept) 1 0.55314   2.8882      6     14 0.04782 *
## Residuals   19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

equivalent to `anova(modman, manova(Yhn ~ O))`

Assumptions: multivariate normality

`ggpairs(data.frame(Yhn))`



Not so bad, actually.

Mapping results on a scalp

```

# install.packages("eegkit")
library(eegkit)

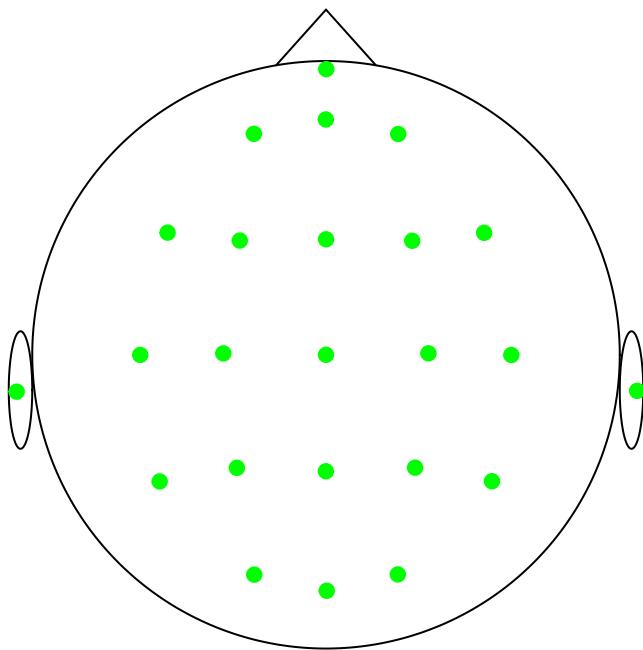
```

```

## Warning: il pacchetto 'eegkit' è stato creato con R versione 4.4.2
## Warning: il pacchetto 'signal' è stato creato con R versione 4.4.2

```

```
# plot 2d cap without labels
eegcap("10-20", plotlabels = FALSE)
```

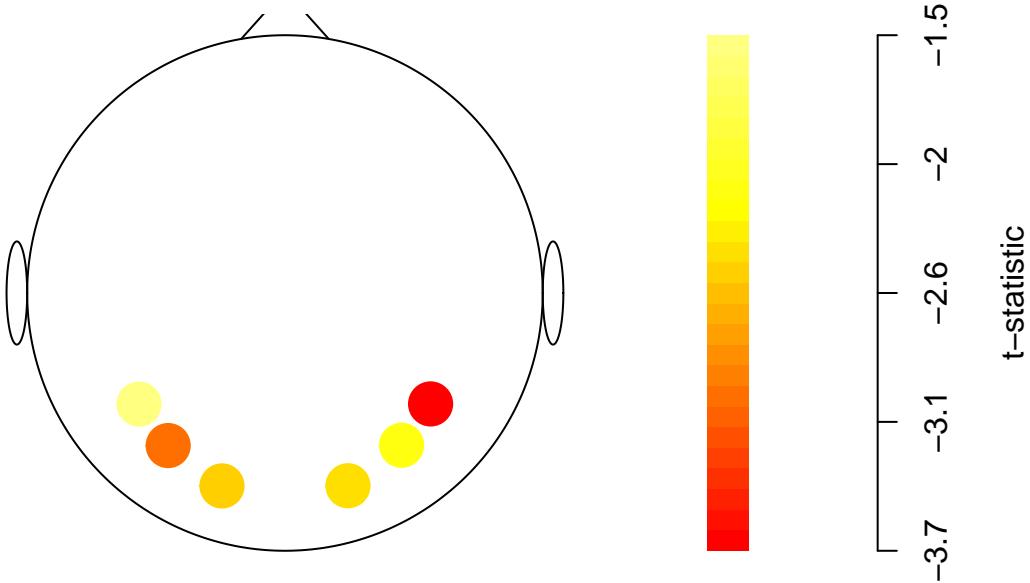


```
# get the t-statistic for each channel:
t_chan=sapply(uni_t,function(chan)chan$statistic)
names(t_chan)=gsub("\\.t","",names(t_chan))

# match to eeg coordinates
data(eegcoord)
cidx <- match(names(t_chan),rownames(eegcoord))

# # plot t-stat in 3d
# open3d()
# eegspace(eegcoord[cidx,1:3],t_chan)

# plot t-stat in 2d
eegspace(eegcoord[cidx,4:5],t_chan,cex.point = 3,colorlab="t-statistic",mycolors=heat.colors(4))
```



I suggest you to play with much nicer plots (based on library `ggplot2`):

- package `eeguana` <https://github.com/bnichenboim/eeguana>
- package `eegUtils` <https://github.com/craddm/eegUtils>

(minimal) Bibliography

About (generalized) linear Mixed models:

`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

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

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>

Resampling based:

The bridge between rep meas. ANOVA and Mixed Models is built and developed by (very nice job (!), by the way):

<https://jaromilfrossard.github.io/gANOVA/>

<https://jaromilfrossard.github.io/gANOVA/articles/spherical-distribution-example.html>

<https://arxiv.org/abs/1903.10766>

More

About Type I error, II, III SS: <https://mcfromnz.wordpress.com/2011/03/02/anova-type-iiiiii-ss-explained/>