

# Statistical Analysis of EEG-experiment data

*Livio Finos*

*09 ottobre 2018*

## Contents

<b>Introduction</b>	<b>1</b>
The data . . . . .	2
Motivation (EDA) . . . . .	2
<b>Repeated Measures ANOVA</b>	<b>5</b>
Introduction . . . . .	5
2 conditions, matched observations . . . . .	5
Linear models with repeated measures . . . . .	7
Repeated measures . . . . .	8
Spend your DF in a different way! . . . . .	9
Sphericity . . . . .	11
(Further) Limitations . . . . .	13
<b>Mixed models</b>	<b>13</b>
Motivation/Introduction . . . . .	13
The model . . . . .	14
Plotting tools . . . . .	17
Validity of the assumptions . . . . .	22
Contrasts and post-hoc . . . . .	24
<b>Multivariate ANOVA (MANOVA)</b>	<b>26</b>
Motivation . . . . .	26
Reshaping the data . . . . .	26
Manova . . . . .	30
<b>Mapping results on a scalp</b>	<b>32</b>
<b>(minimal) Bibliography</b>	<b>34</b>

## 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 (face)
- Measure: Area around the component P170

Setting parameters, importing the data:

```
rm(list=ls())

load("./dataset/datiEEG.Rdata")

# dati2 is the same as dati, only selecting:
# ( Chan = "O1" ) & ( Condition = "f" or "n" )
load("./dataset/dati2EEG.Rdata")

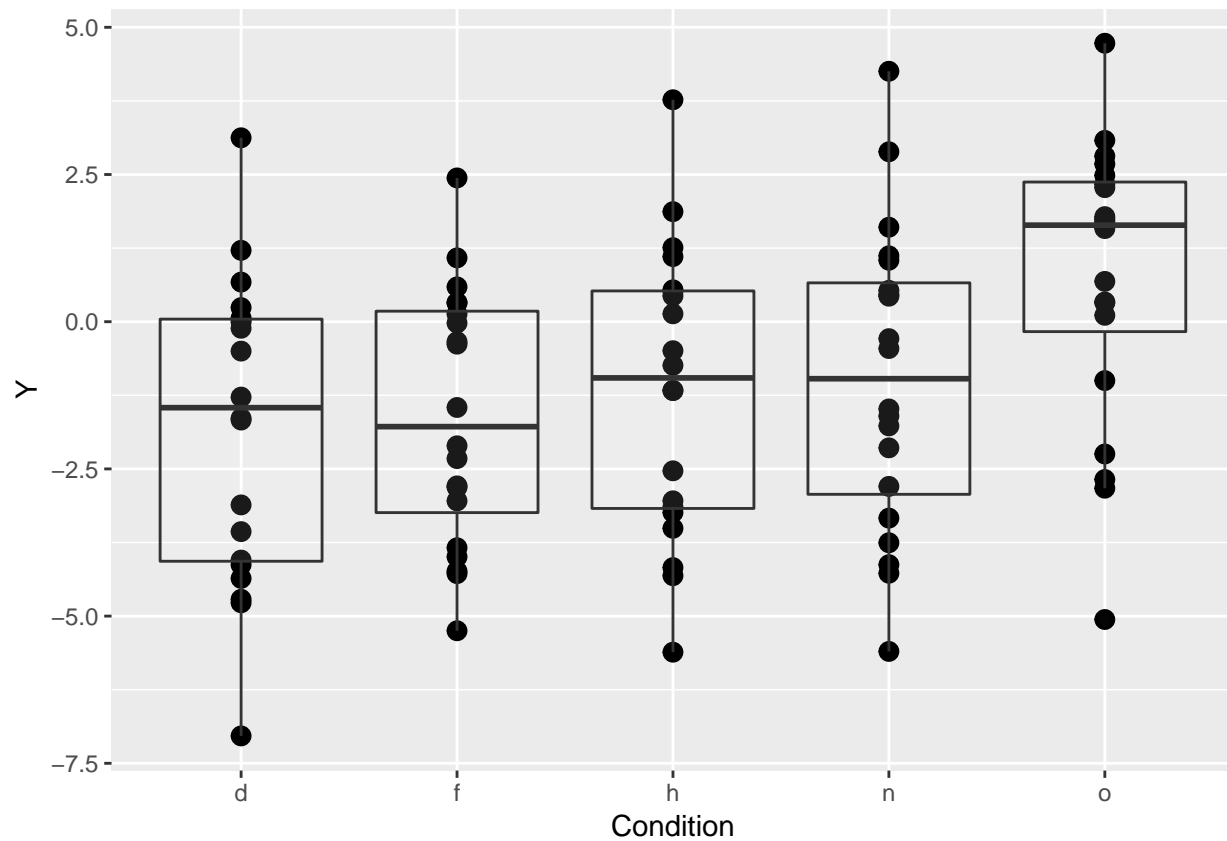
# VERY IMPORTANT:
contrasts(dati$Chan) <- contr.sum(6)
contrasts(dati$Condition) <- contr.sum(5)
contrasts(dati$Subj) <- contr.sum(nlevels(dati$Subj))

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

## Motivation (EDA)

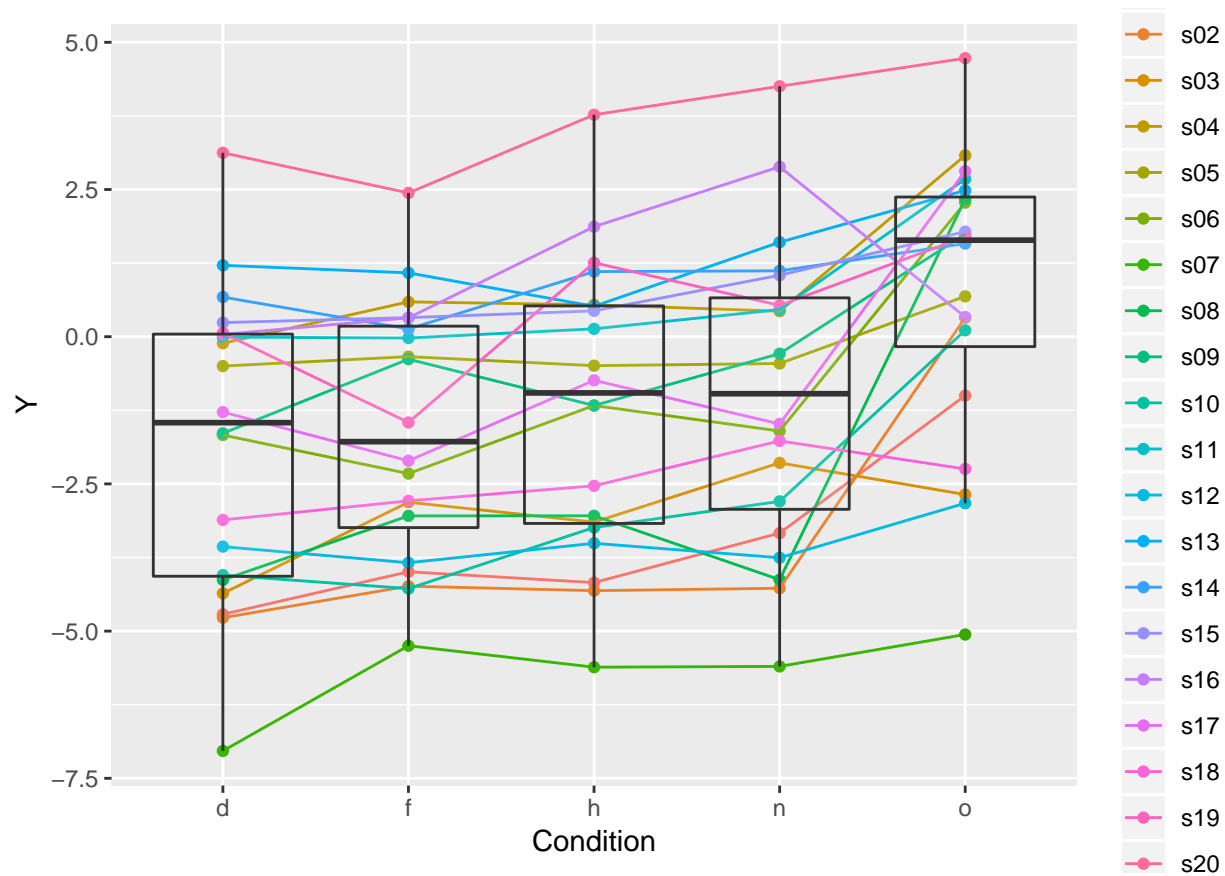
For Channel O1:

```
library(ggplot2)
p <- ggplot(subset(dati, Chan=="O1"), aes(Condition, Y))
p+geom_point(size = 3) +geom_boxplot(alpha=.1)
```



Is there a specificity of the subject?

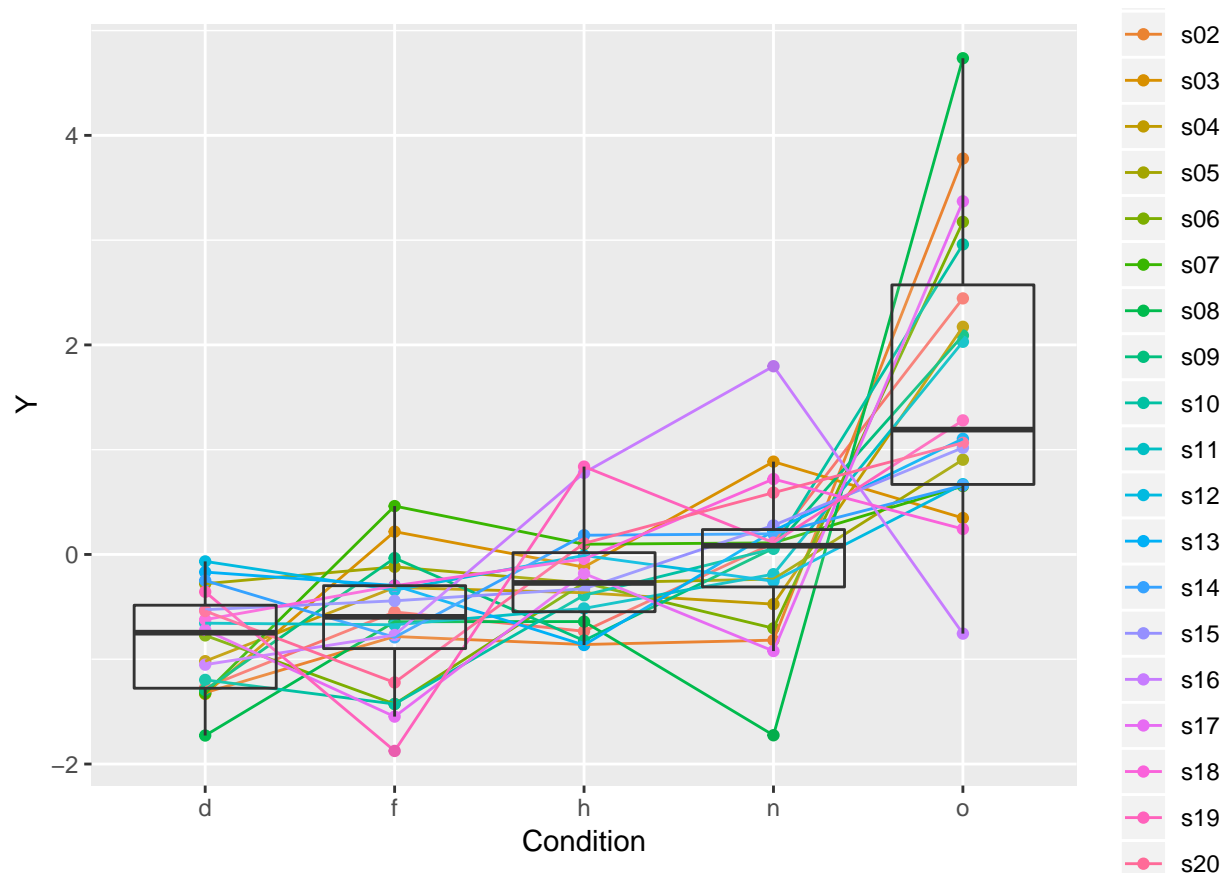
```
dati01=subset(dati,Chan=="01")
library(ggplot2)
p <- ggplot(dati01,aes(Condition,Y))
p+geom_point(aes(group = Subj, colour = Subj))+
  geom_line(aes(group = Subj, colour = Subj))+
  geom_boxplot(alpha=.1)
```



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](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](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, matched observations

Let consider the reduced problem: channel `Chan=="O1` 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 difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2303616 1.0380084
## sample estimates:
## mean of the differences
##                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)
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 phase.

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): ANOVA F-tests on an essentially perfect fit
## are unreliable
```

```
## Analysis of Variance Table
##
## Response: Y
##
```

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

## 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)
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 DFd           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))
```

```
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	DFd	F	p	p<.05
## 2	Condition	4	76	78.6974657	2.998429e-26	*
## 3	Lateral	1	19	0.8340652	3.725436e-01	
## 4	ChanL	2	38	56.1729241	4.478690e-12	*
## 5	Condition:Lateral	4	76	0.1443064	9.649821e-01	
## 6	Condition:ChanL	8	152	35.1954651	5.609013e-31	*
## 7	Lateral:ChanL	2	38	2.8073524	7.292266e-02	
## 8	Condition:Lateral:ChanL	8	152	2.6449974	9.620701e-03	*

```
## ges
## 2 0.2075450604
## 3 0.0070380089
## 4 0.1749978790
## 5 0.0001853800
## 6 0.0159782333
## 7 0.0040220980
## 8 0.0006202045
##
## $`Mauchly's Test for Sphericity`
##
```

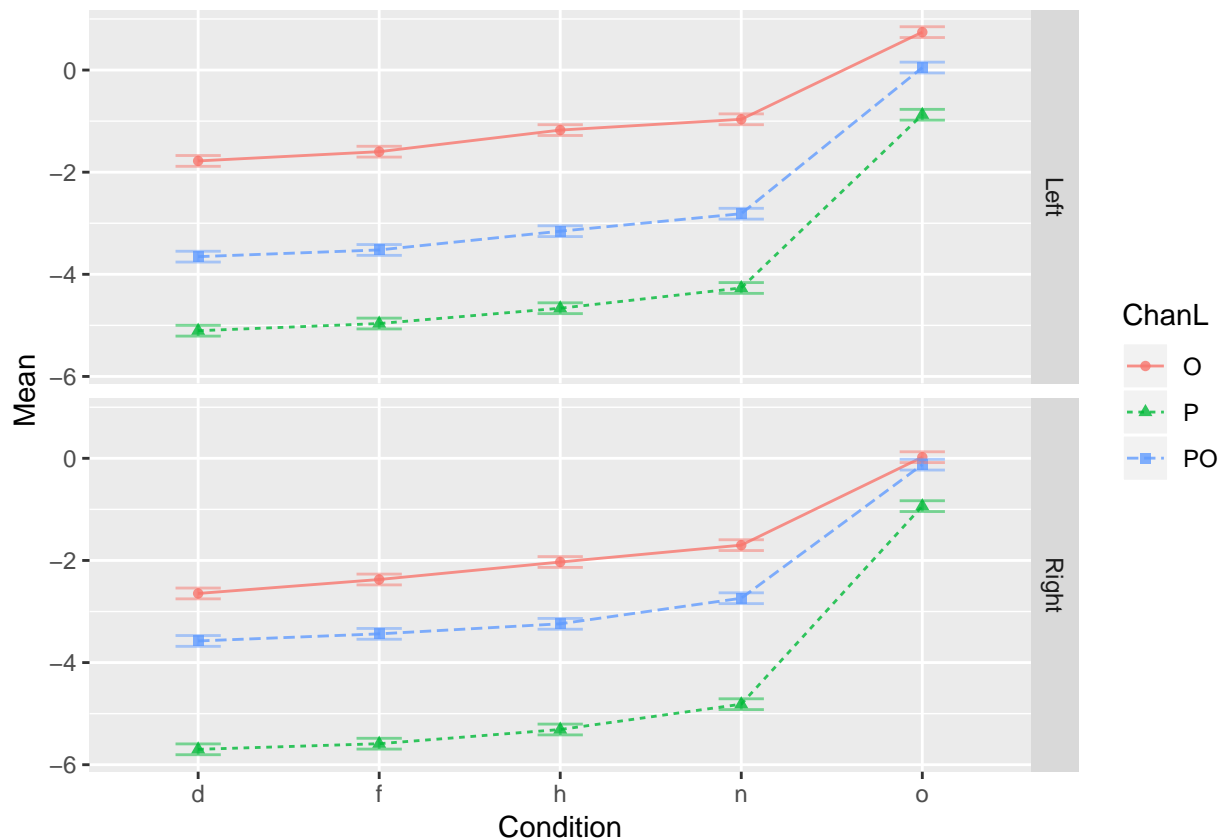
	Effect	W	p	p<.05
## 2	Condition	6.754172e-02	4.802965e-07	*
## 4	ChanL	6.299144e-01	1.561471e-02	*
## 5	Condition:Lateral	9.051612e-03	1.045354e-13	*
## 6	Condition:ChanL	1.397323e-05	7.887294e-21	*
## 7	Lateral:ChanL	8.087935e-01	1.480945e-01	
## 8	Condition:Lateral:ChanL	3.462436e-05	2.626994e-18	*

```
##
## $`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=ChanL,row=Lateral)
```



## Sphericity

The sphericity assumption 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 stricter) condition.

### Compound symmetry

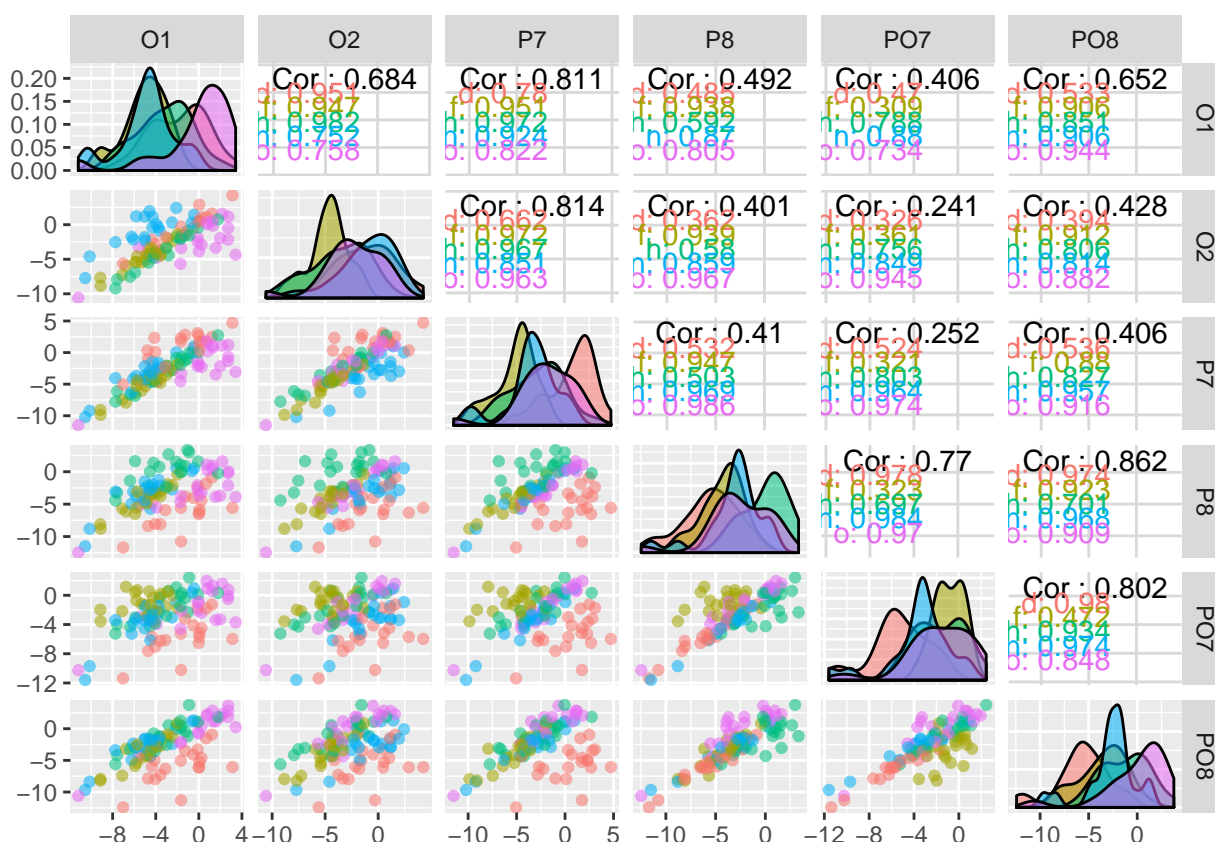
Compound symmetry is met if all the covariances (the off-diagonal elements of the covariance matrix) are equal and all the variances are equal in the populations being sampled. Provided the observed covariances are roughly equal in our samples (and the variances are OK too) we can be pretty confident that compound symmetry is not violated.

If compound symmetry is met, then sphericity is also met.

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 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.

Is compound symmetry met in our data?

```
# install.packages("GGally")
library(GGally)
Y=matrix(dati$Y,byrow = TRUE,nrow = 20*5)
Y=data.frame(Y)
names(Y)=levels(dati$Chan)
ggpairs(Y,aes(colour = dati$Condition[1:100], alpha = 0.4))
```



Not really! (correlations often differ)

### Sphericity

The sphericity assumption is that the all the variances of the differences are equal (in the population sampled). In practice, we'd expect the observed sample variances of the differences to be similar if the sphericity assumption was met.

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:

$$S_x^2 - S_y^2 = S_x^2 + S_y^2 - 2S_{xy}$$

Further reading: <http://homepages.gold.ac.uk/aphome/spheric.html>

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

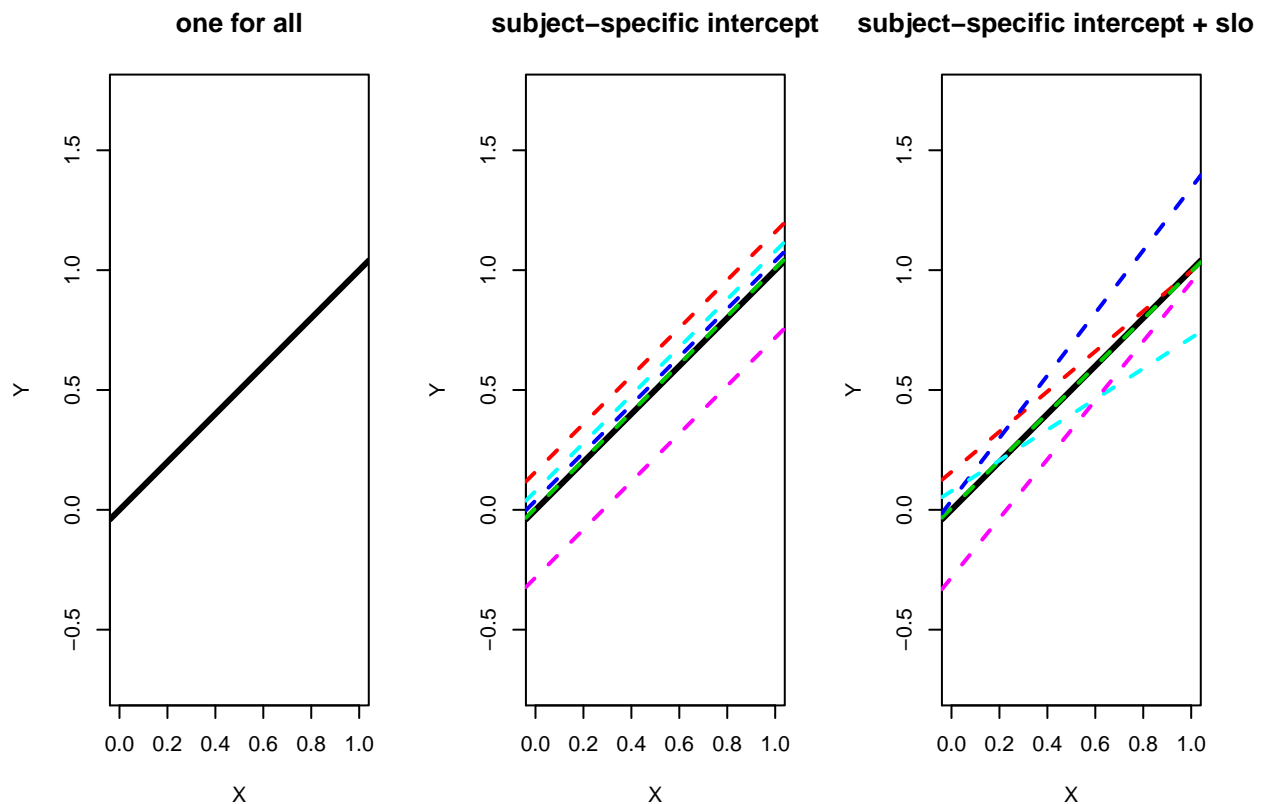
## (Further) Limitations

- (Design and) Data must be balanced
- Repeated Measures Anova doesn't allow for missing data (e.g. subjects/condition/channel cells)
- It only handle factors, no quantitative variables

Mixed model is a more flexible approach.

## Mixed models

### Motivation/Introduction



Mixed models allow for more flexible modelization.

I assume you are expert on mixed models, if not [https://en.wikipedia.org/wiki/Mixed\\_model](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](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 effect for Subject (Random Intercept)

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

```
library(lmerTest)
# 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 effect for Subject and Channel

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$Subj`.

```
mod2=lmer(Y~ 0+Lateral*ChanL*Condition +(0+Lateral*ChanL|Subj),data=dati)
summary(mod2)
```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Y ~ 0 + Lateral * ChanL * Condition + (0 + Lateral * ChanL |
##   Subj)
##   Data: dati
##
## REML criterion at convergence: 1935.8
##
## Scaled residuals:
##   Min      1Q  Median      3Q      Max
## -4.1899 -0.4579 -0.0040  0.5130  3.5464
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   Subj      LateralLeft         3.8560   1.9637
##            LateralRight        6.9945   2.6447    0.63
##            ChanL1              0.6787   0.8238    0.20  0.24
##            ChanL2              0.4654   0.6822   -0.47 -0.45 -0.87
##            Lateral1:ChanL1     0.1853   0.4305   -0.07  0.18 -0.18  0.02
##            Lateral1:ChanL2     0.1642   0.4052   -0.28 -0.14  0.29 -0.16 -0.82
##   Residual                    0.8518   0.9229
## Number of obs: 600, groups:  Subj, 20
##
## Fixed effects:
##
##              Estimate Std. Error      df t value
## LateralLeft      -2.516722    0.442312   19.000071  -5.690
## LateralRight     -2.946310    0.593772   18.999986  -4.962
## ChanL1           1.381018    0.191768   19.000039   7.201
## ChanL2          -1.490923    0.161576   19.025001  -9.227
## Condition1       -1.011749    0.075355  475.000008 -13.426
## Condition2       -0.849248    0.075355  475.000008 -11.270
## Condition3       -0.531396    0.075355  475.000008  -7.052
## Condition4       -0.151912    0.075355  475.000008  -2.016
## Lateral1:ChanL1    0.180370    0.110021   19.150923   1.639
## Lateral1:ChanL2    0.032409    0.105109   19.239316   0.308
## Lateral1:Condition1 0.015606    0.075355  475.000008   0.207
## Lateral1:Condition2 0.004035    0.075355  475.000008   0.054
## Lateral1:Condition3 0.049614    0.075355  475.000008   0.658
## Lateral1:Condition4 -0.013239    0.075355  475.000008  -0.176
## ChanL1:Condition1  0.149200    0.106568  475.000008   1.400
## ChanL2:Condition1 -0.167232    0.106568  475.000008  -1.569
## ChanL1:Condition2  0.213801    0.106568  475.000008   2.006
## ChanL2:Condition2 -0.204413    0.106568  475.000008  -1.918
## ChanL1:Condition3  0.278675    0.106568  475.000008   2.615
## ChanL2:Condition3 -0.233478    0.106568  475.000008  -2.191
## ChanL1:Condition4  0.169969    0.106568  475.000008   1.595
## ChanL2:Condition4 -0.166870    0.106568  475.000008  -1.566
## Lateral1:ChanL1:Condition1 0.022442    0.106568  475.000008   0.211
## Lateral1:ChanL2:Condition1 0.034156    0.106568  475.000008   0.321
## Lateral1:ChanL1:Condition2 -0.011970    0.106568  475.000008  -0.112
## Lateral1:ChanL2:Condition2 0.060872    0.106568  475.000008   0.571
## Lateral1:ChanL1:Condition3 -0.017273    0.106568  475.000008  -0.162
## Lateral1:ChanL2:Condition3 0.026281    0.106568  475.000008   0.247
## Lateral1:ChanL1:Condition4 -0.014015    0.106568  475.000008  -0.132

```

```
## Lateral1:ChanL2:Condition4 0.039652 0.106568 475.000008 0.372
## Pr(>|t|)
## LateralLeft 1.74e-05 ***
## LateralRight 8.65e-05 ***
## ChanL1 7.70e-07 ***
## ChanL2 1.87e-08 ***
## Condition1 < 2e-16 ***
## Condition2 < 2e-16 ***
## Condition3 6.26e-12 ***
## Condition4 0.04437 *
## Lateral1:ChanL1 0.11745
## Lateral1:ChanL2 0.76114
## Lateral1:Condition1 0.83602
## Lateral1:Condition2 0.95731
## Lateral1:Condition3 0.51060
## Lateral1:Condition4 0.86061
## ChanL1:Condition1 0.16216
## ChanL2:Condition1 0.11725
## ChanL1:Condition2 0.04540 *
## ChanL2:Condition2 0.05569 .
## ChanL1:Condition3 0.00921 **
## ChanL2:Condition3 0.02895 *
## ChanL1:Condition4 0.11139
## ChanL2:Condition4 0.11805
## Lateral1:ChanL1:Condition1 0.83330
## Lateral1:ChanL2:Condition1 0.74872
## Lateral1:ChanL1:Condition2 0.91062
## Lateral1:ChanL2:Condition2 0.56813
## Lateral1:ChanL1:Condition3 0.87131
## Lateral1:ChanL2:Condition3 0.80531
## Lateral1:ChanL1:Condition4 0.89543
## Lateral1:ChanL2:Condition4 0.71000
## ---
## 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
```

```
#alternative:
```

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

```
# +(1+Chan/Subj) (equivalent to +(Chan/Subj)) is a linear combination of +(0+Chan/Subj). I better lik
```

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

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Y
## Chisq Df Pr(>Chisq)
## Lateral 31.7515 2 1.274e-07 ***
## ChanL 81.2666 2 < 2.2e-16 ***
## Condition 1200.8753 4 < 2.2e-16 ***
```



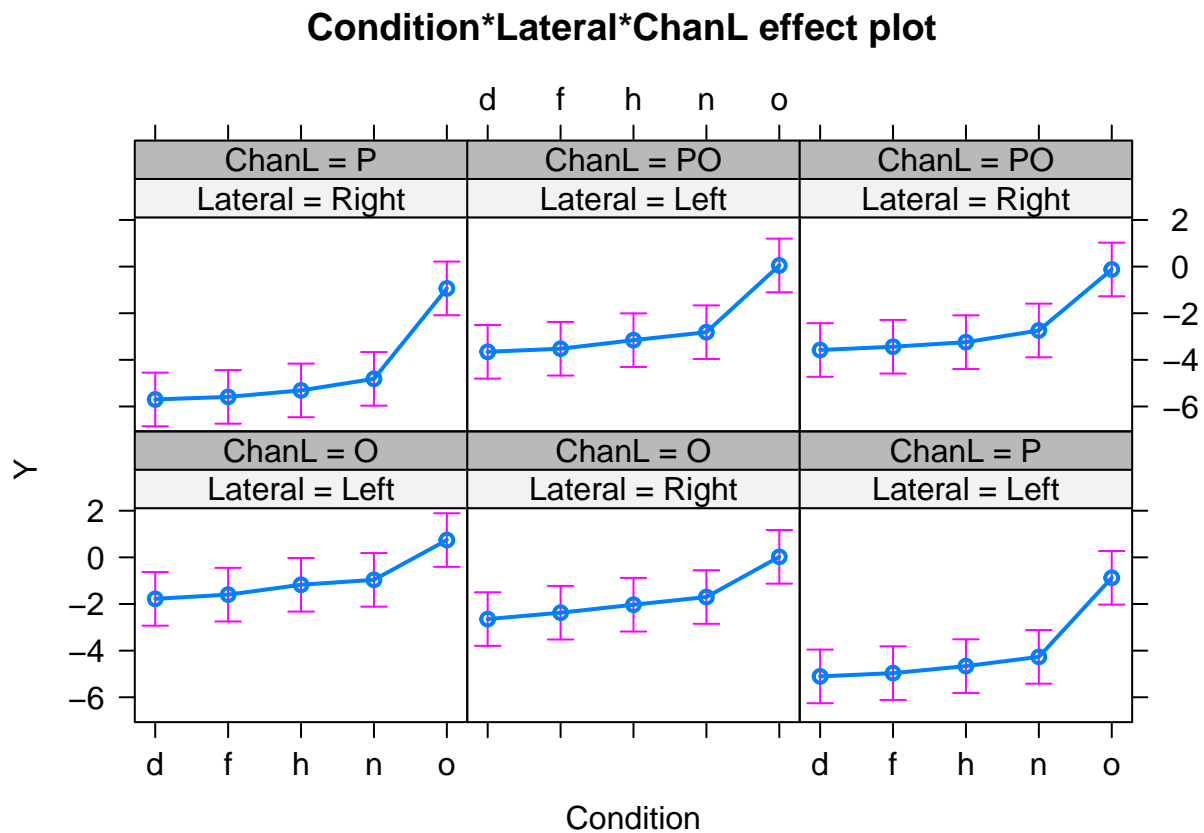
```
## Lateral:ChanL          7.7561  2    0.02069 *
## Lateral:Condition      0.8502  4    0.93160
## ChanL:Condition       74.4533  8    6.346e-13 ***
## Lateral:ChanL:Condition  2.8455  8    0.94367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# More flexible, but hard to fit:
# mod3=lmer(Y~ Condition*ChanL+Lateral +(0+Lateral/Subj)+(0+Condition/Subj)+(0+ChanL/Subj),data=dati)
#
```

## Plotting tools

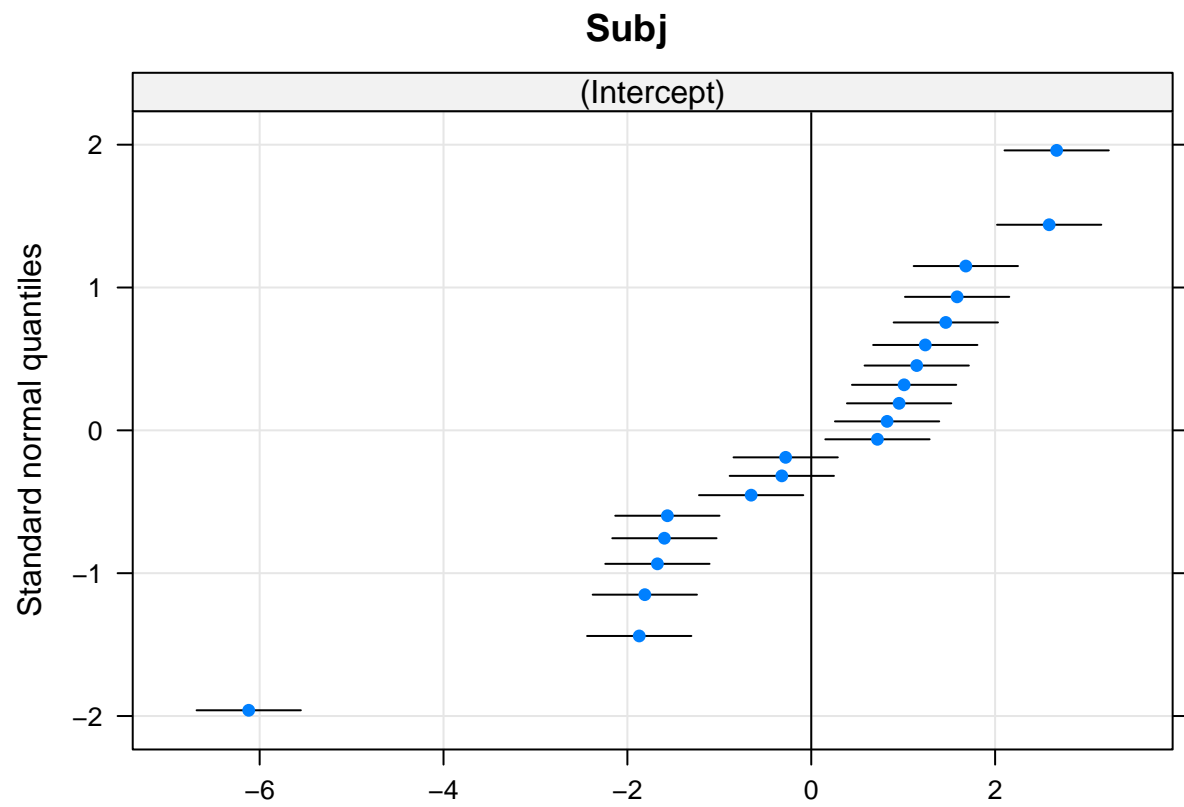
for the first model:

```
library(effects)
plot(allEffects(mod))
```



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

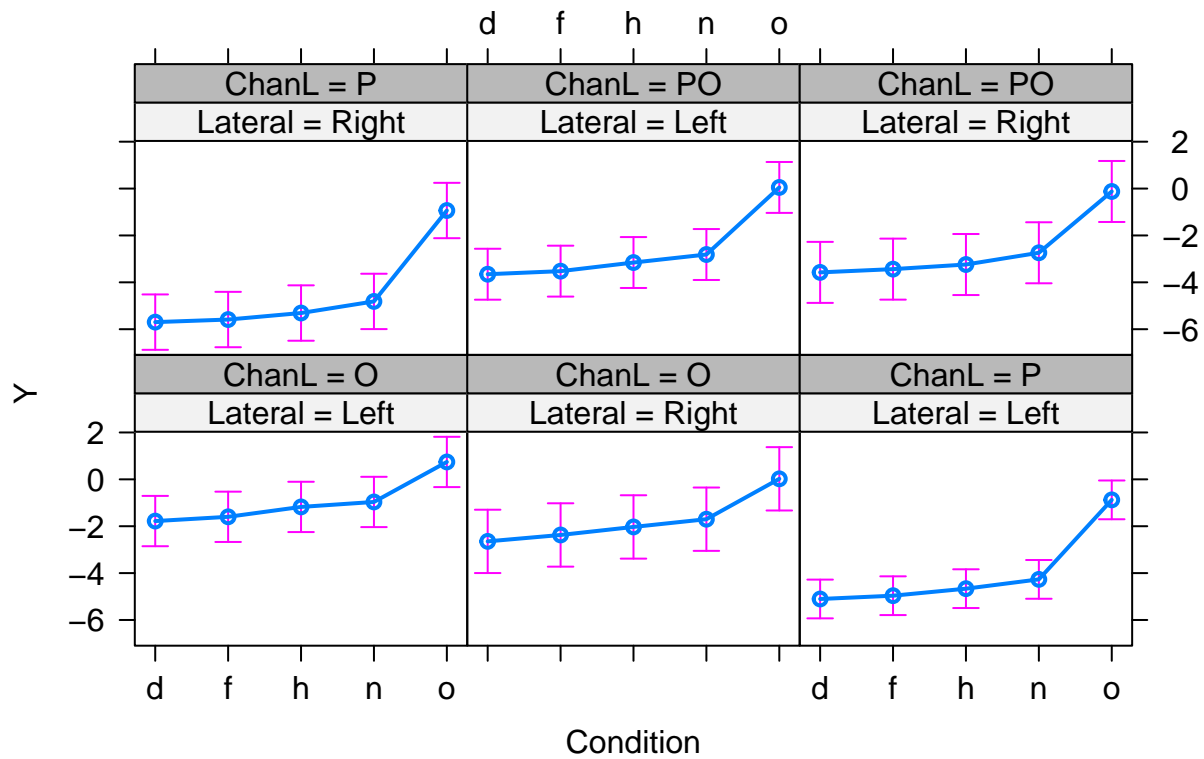
```
## $Subj
```



The second model:

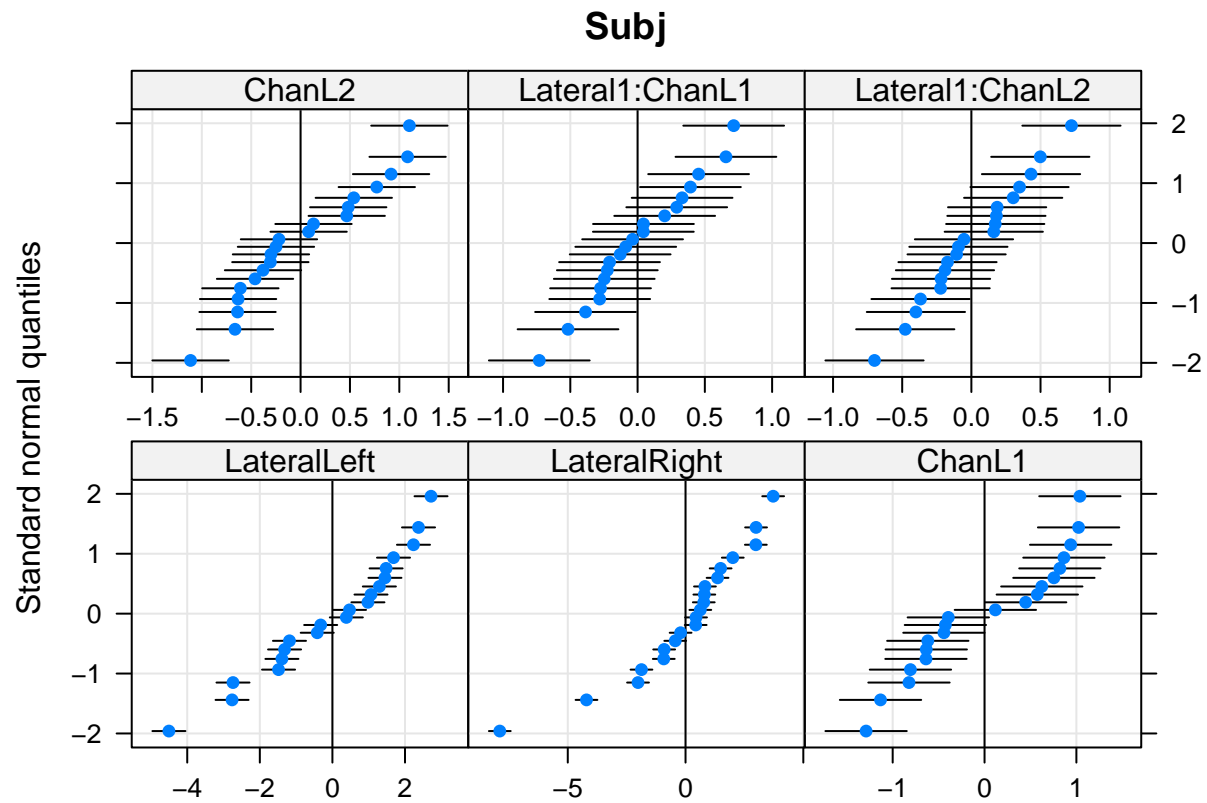
```
library(effects)  
plot(allEffects(mod2))
```

## Lateral\*ChanL\*Condition effect plot

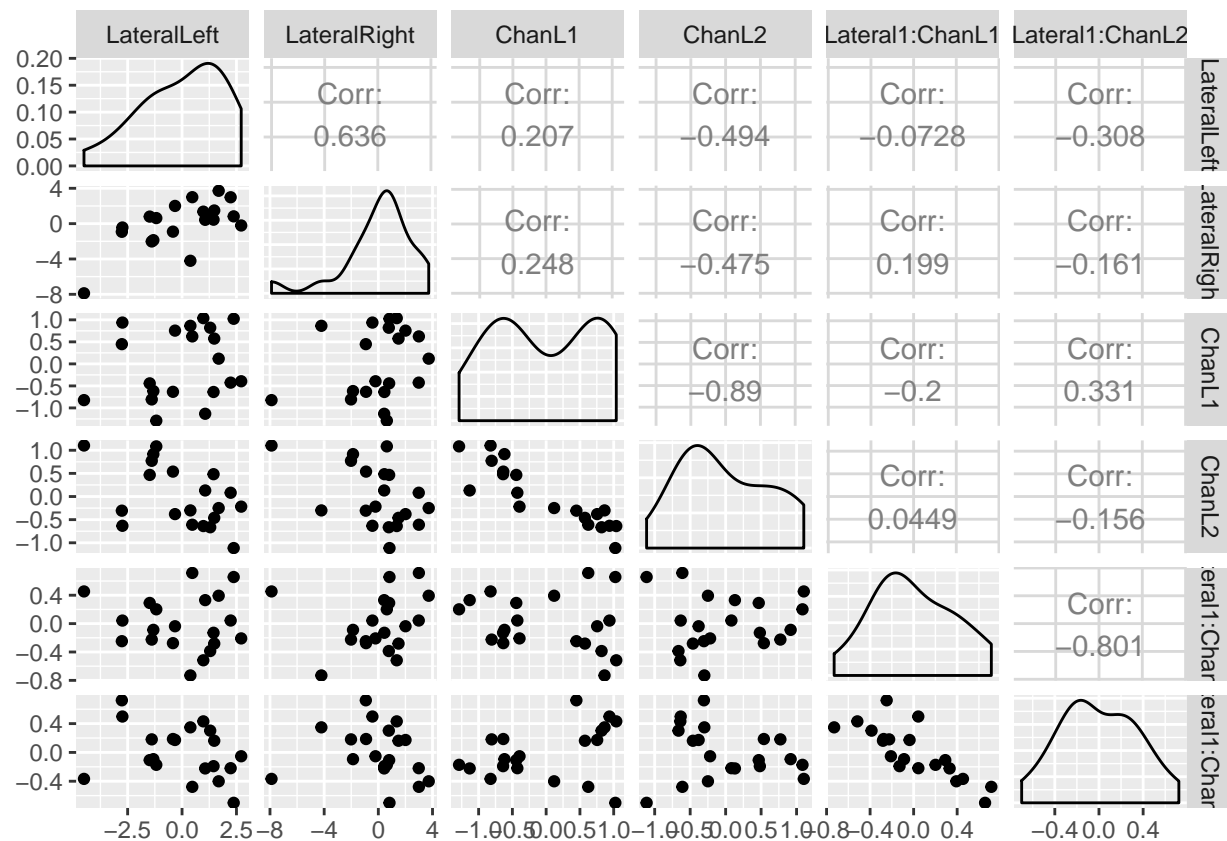


```
#plot random effects:
require(lattice)
qqmath(ranef(mod2, condVar=TRUE))
```

```
## $Subj
```

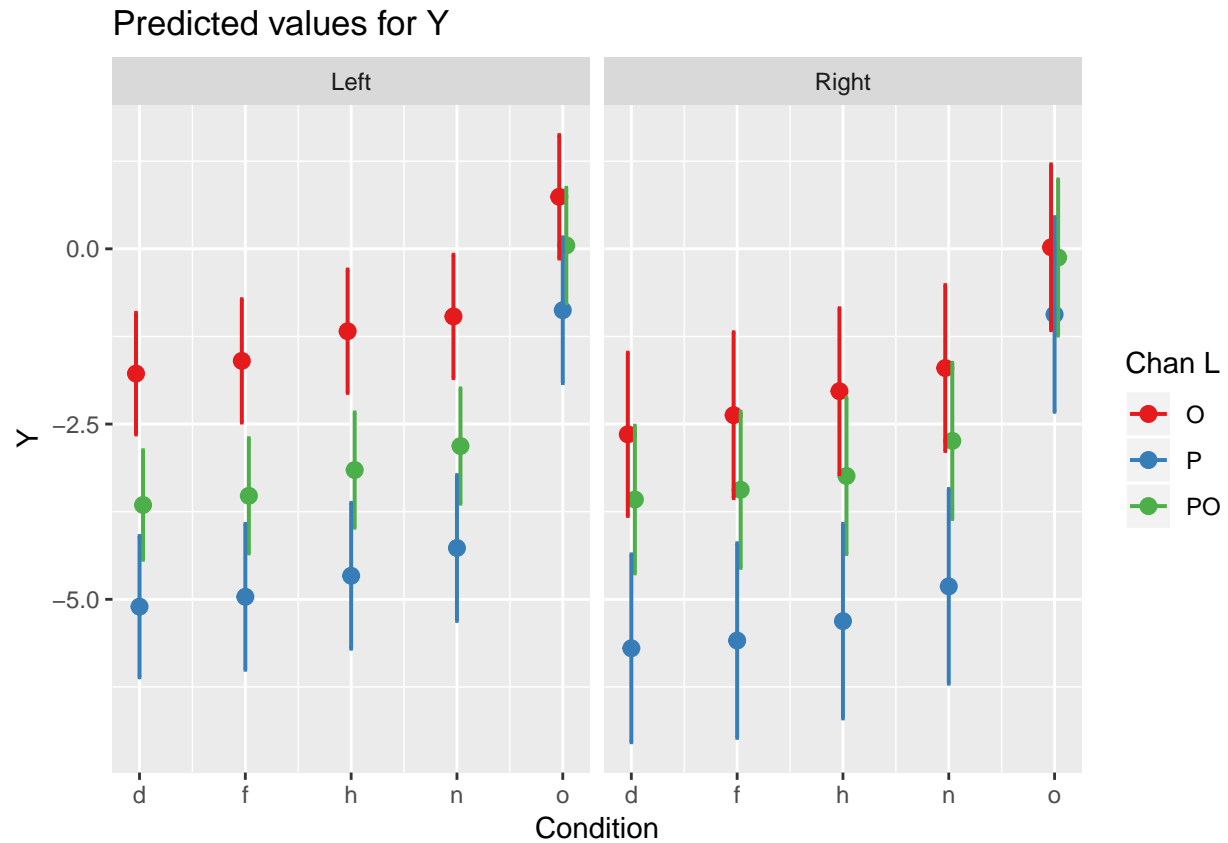


```
# scatter plot
ggpairs(ranef(mod2, condVar=TRUE)[[1]])
```



An alternative plotting tool:

```
library(sjPlot)
library(ggplot2)
plot_model(mod2, type = "pred", terms = c("Condition", "ChanL", "Lateral"))
```



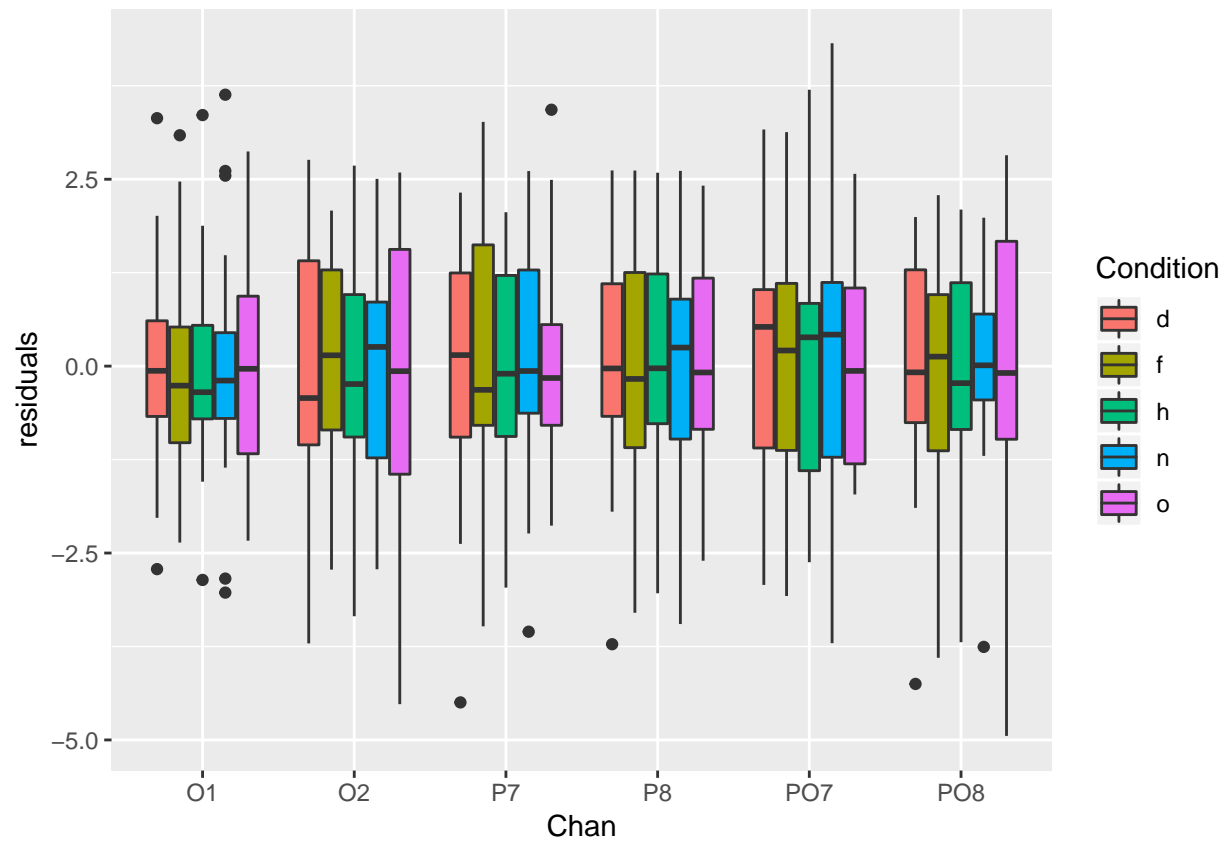
## Validity of the assumptions

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

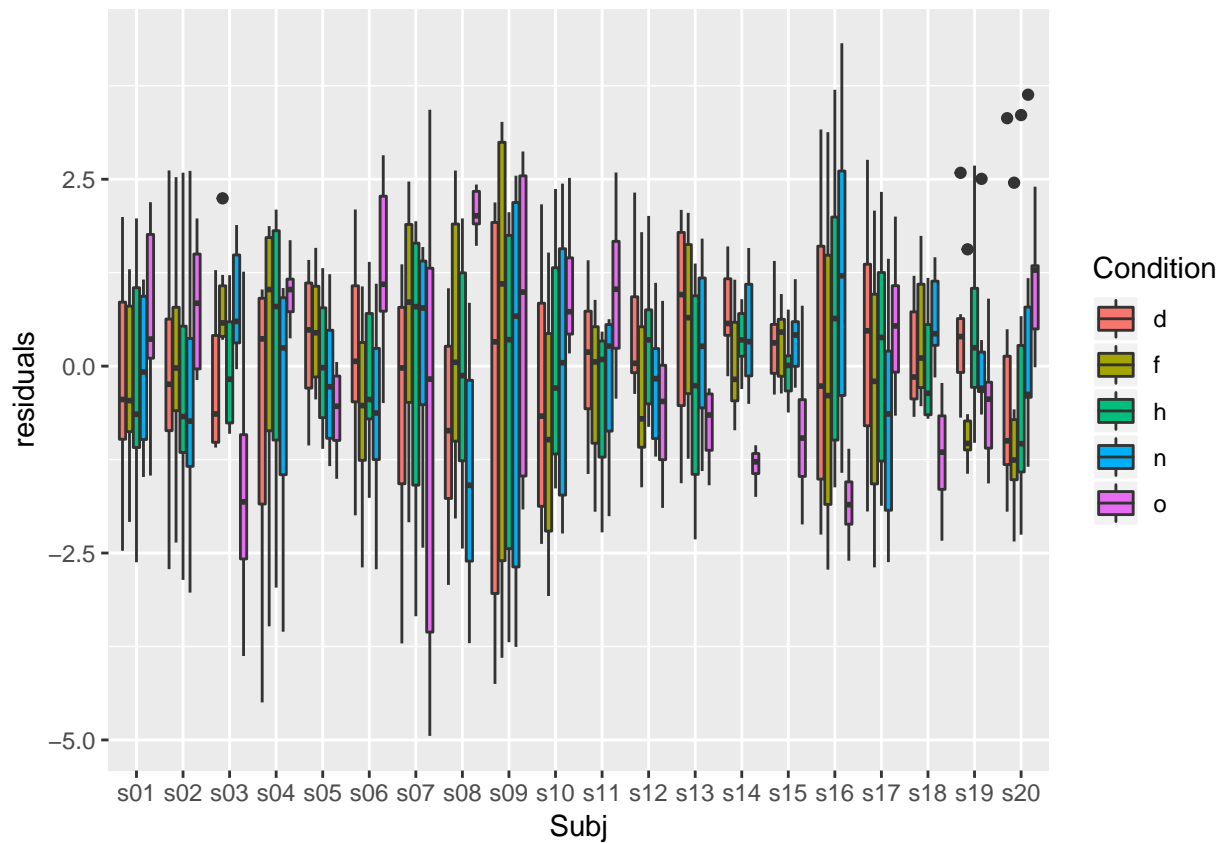
Please, do not test for normality, for homoscedasticity, sphericity etc.

Use Exploratory data Analysis, instead!

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



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



## Contrasts and post-hoc

### Post-hoc

```
library(multcomp)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
```

```
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
```

```
##
```

```
##      geyser
```



```
summary(glht(mod2, 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 ~ 0 + Lateral * ChanL * Condition + (0 + Lateral *
## ChanL | Subj), data = dati)
##
## Linear Hypotheses:
## Estimate Std. Error z value Pr(>|z|)
## f - d == 0 0.1625 0.1191 1.364 0.6509
## h - d == 0 0.4804 0.1191 4.032 <0.001 ***
## n - d == 0 0.8598 0.1191 7.217 <0.001 ***
## o - d == 0 3.5561 0.1191 29.846 <0.001 ***
## h - f == 0 0.3179 0.1191 2.668 0.0589 .
## n - f == 0 0.6973 0.1191 5.853 <0.001 ***
## o - f == 0 3.3936 0.1191 28.482 <0.001 ***
## n - h == 0 0.3795 0.1191 3.185 0.0126 *
## o - h == 0 3.0757 0.1191 25.814 <0.001 ***
## o - n == 0 2.6962 0.1191 22.629 <0.001 ***
## ---
## 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(mod2)[[1]])
contr <- rbind("n - o" = c(0,0,0,0,0,0,1,-1, rep(0,ncoeff-8)),
              "d - n" = c(0,0,0,0,0,0,-1, rep(0,ncoeff-7)),
              "f - n" = c(0,0,0,0,1,0,-1, rep(0,ncoeff-7)),
              "h - n" = c(0,0,0,0,0,1,-1, rep(0,ncoeff-7)))
compa= glht(mod2, linfct = contr)
summary(compa, test = adjusted("none"))
```

```
##
## Simultaneous Tests for General Linear Hypotheses
##
```

```
## Fit: lmer(formula = Y ~ 0 + Lateral * ChanL * Condition + (0 + Lateral *
##       ChanL | Subj), data = dati)
##
## Linear Hypotheses:
##           Estimate Std. Error z value Pr(>|z|)
## n - o == 0 -0.37948    0.11915  -3.185  0.00145 **
## d - n == 0  0.53140    0.07536   7.052 1.77e-12 ***
## f - n == 0 -0.48035    0.11915  -4.032 5.54e-05 ***
## h - n == 0 -0.31785    0.11915  -2.668  0.00764 **
## ---
## 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 ~ 0 + Lateral * ChanL * Condition + (0 + Lateral *
##       ChanL | Subj), data = dati)
##
## Linear Hypotheses:
##           Estimate Std. Error z value Pr(>|z|)
## n - o == 0 -0.37948    0.11915  -3.185  0.00466 **
## d - n == 0  0.53140    0.07536   7.052 < 0.001 ***
## f - n == 0 -0.48035    0.11915  -4.032 < 0.001 ***
## h - n == 0 -0.31785    0.11915  -2.668  0.02353 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

## Multivariate ANOVA (MANOVA)

### Motivation

ehi, 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' now compute the vectors of contrasts (one vector of reach channel, length equal to number of subjects):  
Fear 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(1,4),1]=c(1,-1)
contr[c(1,4)+5,2]=c(1,-1)
contr[c(1,4)+10,3]=c(1,-1)
contr[c(1,4)+15,4]=c(1,-1)
contr[c(1,4)+20,5]=c(1,-1)
contr[c(1,4)+25,6]=c(1,-1)
```

```
dim(contr)
```

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

```
head(contr)
```

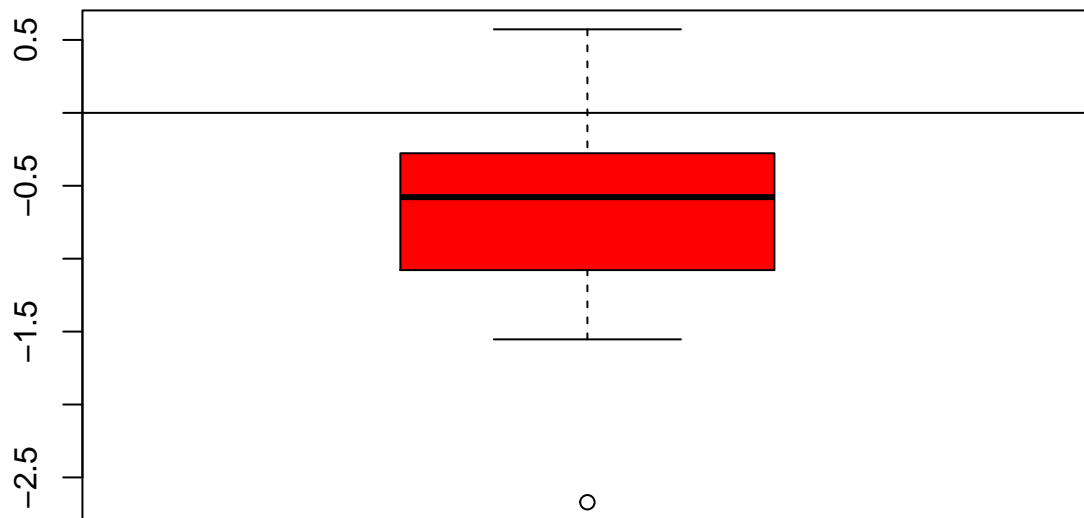
```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    0    0    0    0    0
## [3,]    0    0    0    0    0    0
## [4,]   -1    0    0    0    0    0
## [5,]    0    0    0    0    0    0
## [6,]    0    1    0    0    0    0
```

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

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

What we see in O1?

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



Same test as above, but under a different model

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

```
##
##  One Sample t-test
##
## data:  Yfn[, 1]
## t = -4.3327, df = 19, p-value = 0.0003587
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.0327953 -0.3599747
## sample estimates:
## mean of x
## -0.696385
```

We can run the analysis over all channels

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

```
## $01
##
##  One Sample t-test
##
## data:  newX[, i]
```

```

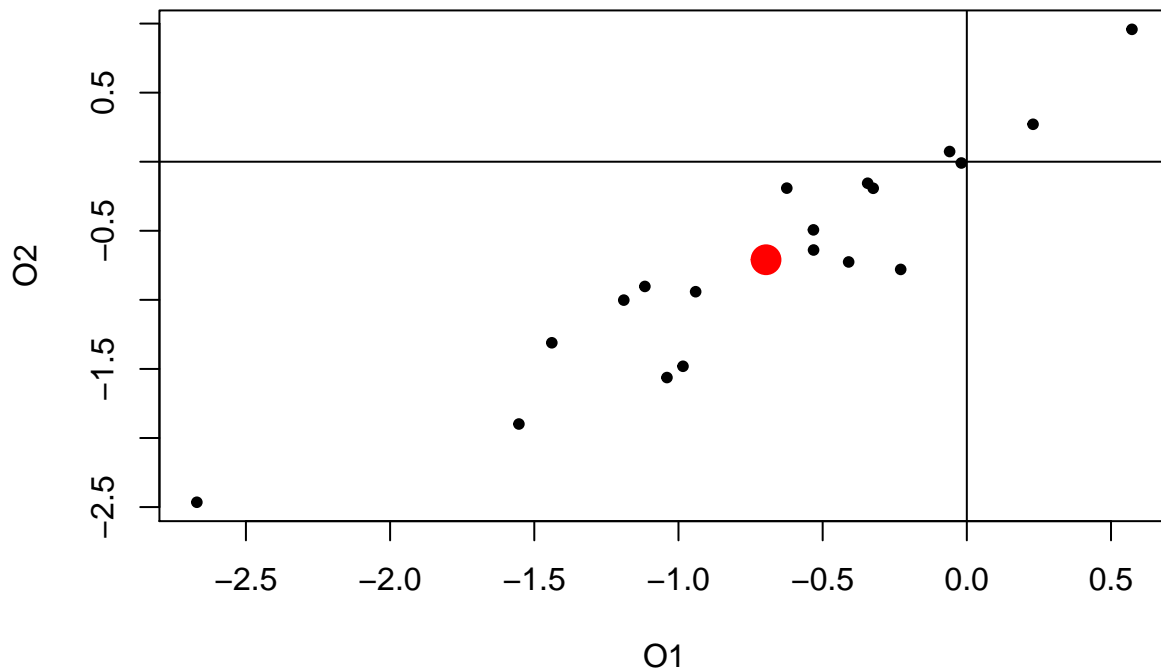
## t = -4.3327, df = 19, p-value = 0.0003587
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.0327953 -0.3599747
## sample estimates:
## mean of x
## -0.696385
##
##
## $O2
##
## One Sample t-test
##
## data: newX[, i]
## t = -3.9761, df = 19, p-value = 0.0008092
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.0831578 -0.3360722
## sample estimates:
## mean of x
## -0.709615
##
##
## $P7
##
## One Sample t-test
##
## data: newX[, i]
## t = -3.287, df = 19, p-value = 0.003877
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.0380084 -0.2303616
## sample estimates:
## mean of x
## -0.634185
##
##
## $P8
##
## One Sample t-test
##
## data: newX[, i]
## t = -5.1301, df = 19, p-value = 5.95e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.0889013 -0.4578487
## sample estimates:
## mean of x
## -0.773375
##
##
## $P07
##
## One Sample t-test

```

```
##
## data: newX[, i]
## t = -3.8837, df = 19, p-value = 0.0009992
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.0736034 -0.3216666
## sample estimates:
## mean of x
## -0.697635
##
##
## $P08
##
## One Sample t-test
##
## data: newX[, i]
## t = -3.1978, df = 19, p-value = 0.004737
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.1131969 -0.2324531
## sample estimates:
## mean of x
## -0.672825
```

## Manova

```
plot(Yfn[,1:2],pch=20)
abline(v=0)
abline(h=0)
points(mean(Yfn[,1]),mean(Yfn[,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/Multivariate_analysis_of_variance) [https://en.wikipedia.org/wiki/Hotelling%27s\\_T-squared\\_distribution](https://en.wikipedia.org/wiki/Hotelling%27s_T-squared_distribution)

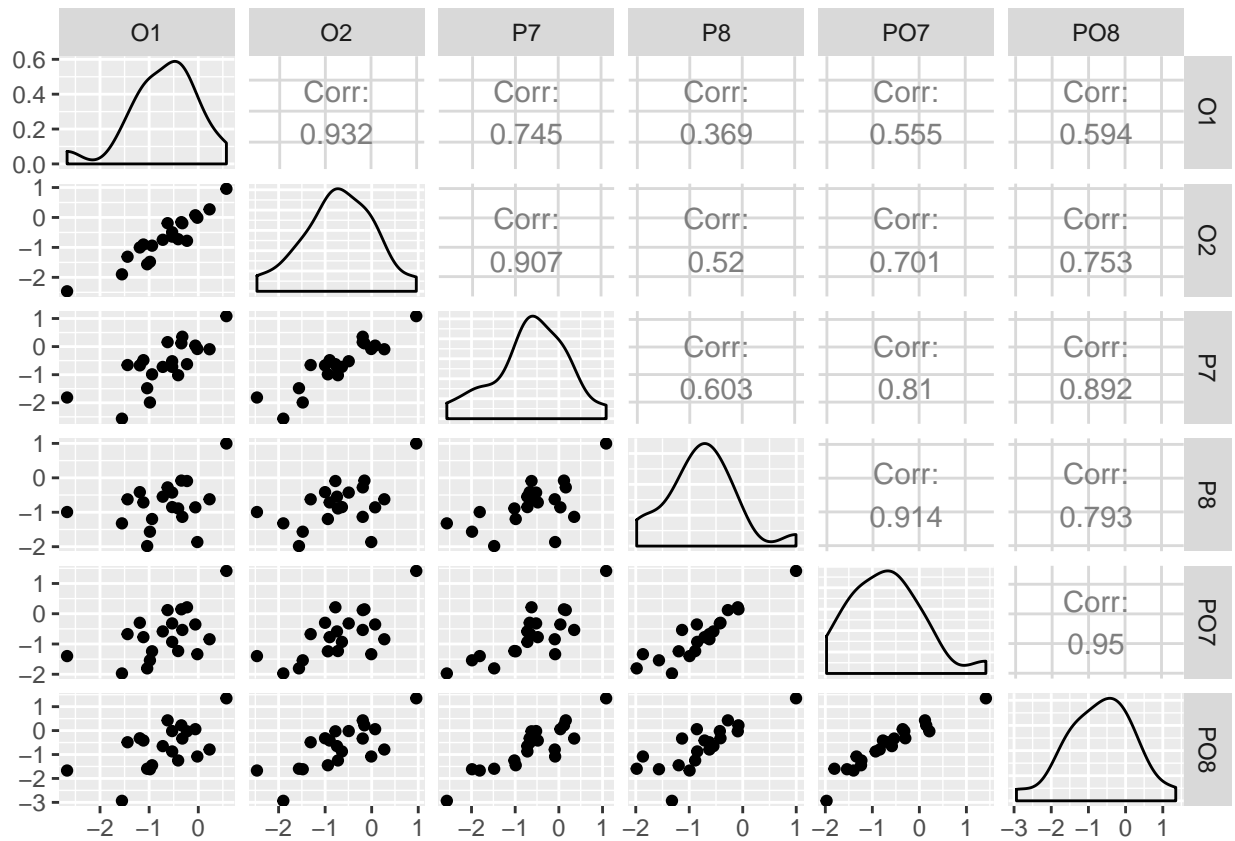
```
modman <- manova(Yfn ~ 1)
anova(modman)
```

```
## Analysis of Variance Table
##
##              Df  Pillai approx F num Df den Df    Pr(>F)
## (Intercept)  1 0.76928      7.78      6    14 0.0007989 ***
## Residuals    19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# equivalent to anova(modman,manova(Yfn ~ 0))
```

Assumptions: multivariate normality

```
subj=dati$Subj[(1:20)*30]
ggpairs(data.frame(Yfn))
```



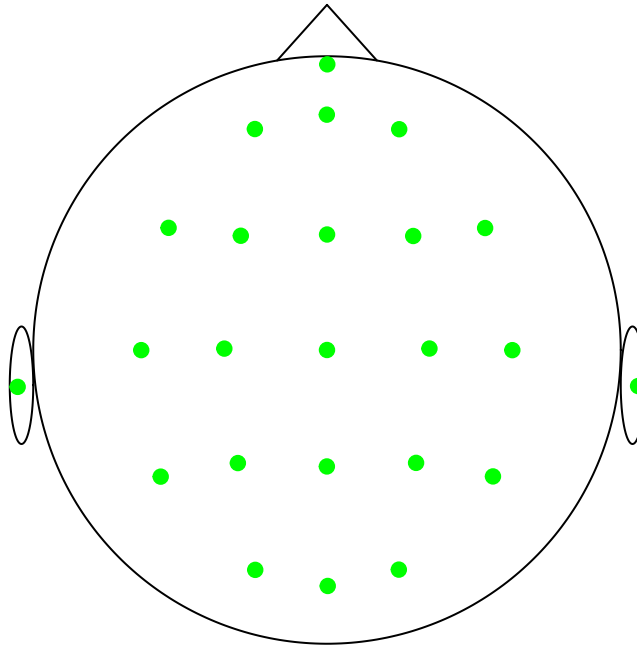
Not so bad, actually.

## Mapping results on a scalp

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

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



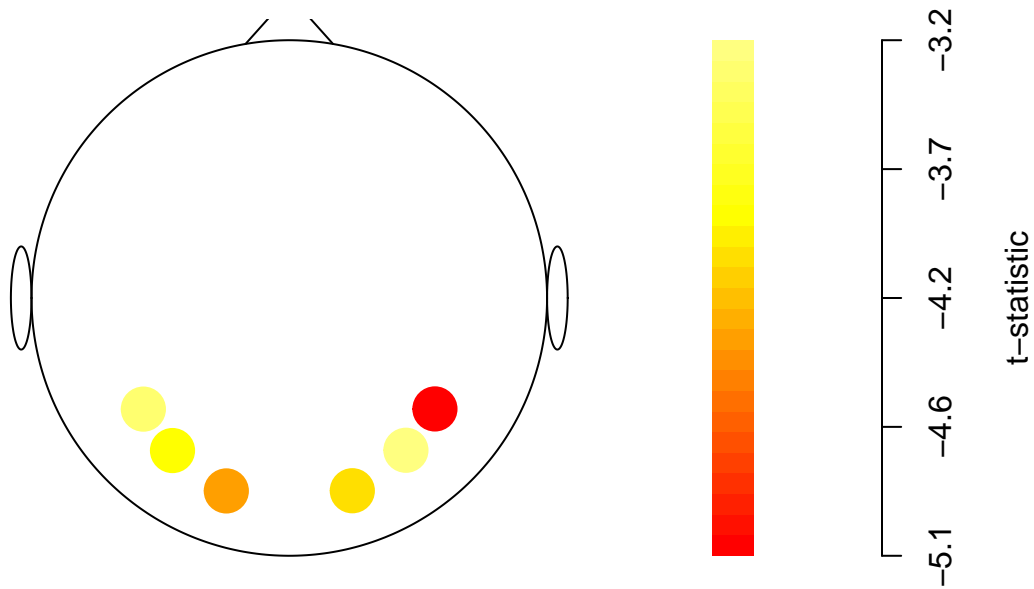


```
# get the t-statistic for each channel:
t_chan=sapply(uni_t,function(chan)chan$Statistic)
names(t_chan)=gsub("\\\\.t","",names(t_chan))

# 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))
```



If you like to play with library `ggplot2` this may help:

<http://www.matterdaddock.com/blog/2017/02/25/erp-visualization-creating-topographical-scalp-maps-part-1/>

## (minimal) Bibliography

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)

and 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/>

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

About Mixed models:

[http://webcom.upmf-grenoble.fr/LIP/Perso/DMuller/M2R/R\\_et\\_Mixed/documents/Bates-book.pdf](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>