Introduction to R

Mandy Vogel

University Leipzig

November 3, 2015

Overview

Recap glms

The famous O-Ring example

Linear Mixed Models or Multilevel Models

Table of Contents I

Generalized Linear Models

Linear modeling assumes constant variance and normally distributed errors. Certain kinds of respond variables lack these constraints. GLMs are excellent at dealing with it.

```
> m.lm <- lm(bweight ~ hyp, data=births)
> m.glm <- glm(bweight ~ hyp, family=gaussian, data=birt
```

give the same answer. The model formula is the same for both, but for `glm()` it is necessary to specify the family of likelihoods which will be used to fit the model.
The `glm()` function allows us to fit other models including logistic regression and Poisson regression.

Examples families with canonical links

- binomial(link = ”logit”)
- gaussian(link = ”identity”)
- poisson(link = ”log”)
- quasibinomial(link = ”logit”)
- quasipoisson(link = ”log”)

Odds

$$p = 1 \qquad \omega = \infty$$

$$p = 0.99 \qquad \omega = 99$$

$$p = 0.5 \qquad \omega = 1$$

$$p = 0.1 \qquad \omega = 0.\overline{1}$$

$$p = 0.01 \qquad \omega = 0.\overline{01}$$

$$p = 0 \qquad \omega = 0$$

Predicting Low Birth Weight

How it looks in R:

```
> m.bin1 <- glm(lowbw ~ hyp, family=binomial, data=births)
> summary(m.bin1)
Call:
glm(formula = lowbw ~ hyp, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8067  -0.4430  -0.4430  -0.4430   2.1773

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.2721     0.1661 -13.682  < 2e-16 ***
hyphyper      1.3166     0.3111   4.232 2.32e-05 ***
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 366.92  on 499  degrees of freedom
Residual deviance: 350.84  on 498  degrees of freedom
```

Interpreting the Cœfficients

- the back transformed intercepts are interpreted as probabilities

```
> invlogit(coef(m.bin1)[1])
(Intercept)
  0.09345794
```

Understanding the Cœfficients

- the `Effect()` and the `allEffects()` functions (`effects` package) are a convenient way to access effects from a glm

```
> Effect("hyp",m.bin1)

 hyp effect
hyp
    normal      hyper
0.09345794 0.27777778
```

Simple Logistic Regression

- modelling the probability of low birth weight dependent on gestational age

```
> m.bin5 <- glm(lowbw ~ gestwks, family=binomial, data=births)
```

- and as math formula

$$\log\left(\frac{\Pr(\text{lowbw})}{1-\Pr(\text{lowbw})}\right) = \beta_0 + \beta_1 \cdot \text{gestwks} + \epsilon$$

Simple Logistic Regression

```
> summary(m.bin5)

Call:
glm(formula = lowbw ~ gestwks, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0873  -0.3623  -0.2223  -0.1369   2.9753

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  31.8477     4.0574   7.849 4.18e-15 ***
gestwks      -0.8965     0.1084  -8.272  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 360.38  on 489  degrees of freedom
Residual deviance: 205.75  on 488  degrees of freedom
  (10 observations deleted due to missingness)
AIC: 209.75

Number of Fisher Scoring iterations: 6
```

Understanding the Cœfficients

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(31.8477 + -0.8965 \cdot \text{gestwks})$$

- the cœfficient for `gestwks` is best interpretable if we use it as argument to the exponential function

```
> exp(coef(m.bin5)[2])
   gestwks
0.4080114
```

this way it is interpretable as odds ratio for low birth weight for a difference of 1 week of gestational age

Effects for logistic regression

```
> Effect("gestwks",m.bin5)

 gestwks effect
gestwks
        25         30         35         40
0.99992022 0.99299324 0.61574996 0.01779725
> Effect("gestwks",m.bin5,xlevels = list(gestwks = c(20,30,40)))

 gestwks effect
gestwks
        20         30         40
0.99999910 0.99299324 0.01779725
```

The Challenger Disaster Example

In January 1986, the space shuttle Challenger exploded shortly after launch. An investigation was launched into the cause of the crash and attention focused on the rubber O-ring seals in the rocket boosters. At lower temperatures, rubber becomes more brittle and is a less effective sealant. At the time of the launch, the temperature was 31°F. Could the failure of the O-rings have been predicted? In the 23 previous shuttle missions for which data exists, some evidence of damage due to blow by and erosion was recorded on some O-rings. Each shuttle had two boosters, each with three O-rings. For each mission, we know the number of O-rings out of six showing some damage and the launch temperature.(faraway)
`http://www.history.com/topics/challenger-disaster/videos/engineering-disasters---challenger`

The Challenger Disaster Example

- the data are given in the data frame `orings` in the `faraway` package
- after loading we have a look at the first six lines

```
> library(faraway)
> data(orings)
> head(orings)
  temp damage
1   53      5
2   57      1
3   58      1
4   63      1
5   66      0
6   67      0
```

- we see that every shuttle mission has its own row (but not every O-ring)

The Challenger Disaster Example

- that is not a problem: one way of defining a binary response variable in a glm is to form a two-column matrix with the first column representing the number of "successes" y and the second column the number of "failures" n-y.

```
> m.or <- glm(cbind(damage,6-damage) ~ temp,
+                              family=binomial, orings)
```

- we see that every shuttle mission has its own row (but not every O-ring)

The Challenger Disaster Example

- the output looks familiar:

```
> summary(m.or)
Call:
glm(formula = cbind(damage, 6 - damage) ~ temp,
     family = binomial, data = orings)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9529  -0.7345  -0.4393  -0.2079   1.9565
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.66299    3.29626   3.538 0.000403 ***
temp        -0.21623    0.05318  -4.066 4.78e-05 ***
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 38.898  on 22  degrees of freedom
Residual deviance: 16.912  on 21  degrees of freedom
AIC: 33.675
```

- remember, the response is a probability. Therefore our model describes the probability of a damaged O-ring depending on the temperature

Understanding the Cœfficients

- this relationship is described by

$$\Pr(\text{damage}) = \text{logit}^{-1}(11.66299 + -0.21623 \cdot \text{temp})$$

- the intercept

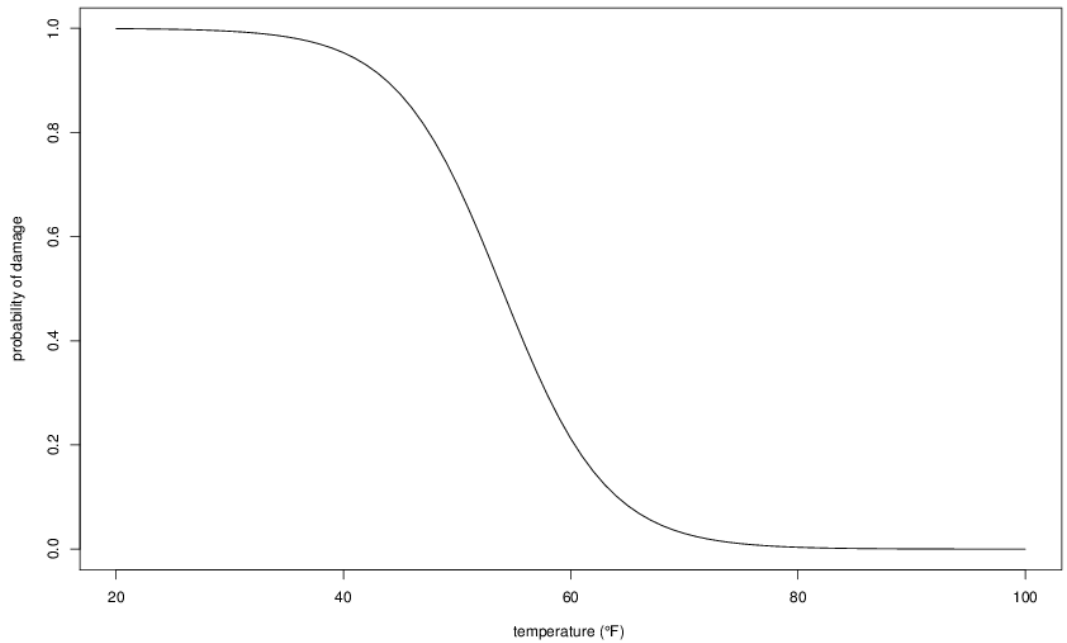
```
> invlogit(coef(m)[1])
(Intercept)
  0.9999914
```

  is interpretable as the probability for a damaged O-ring at a temperature of 0°F

- the parameter for temperature describes how fast the probability decreases with increasing temperature

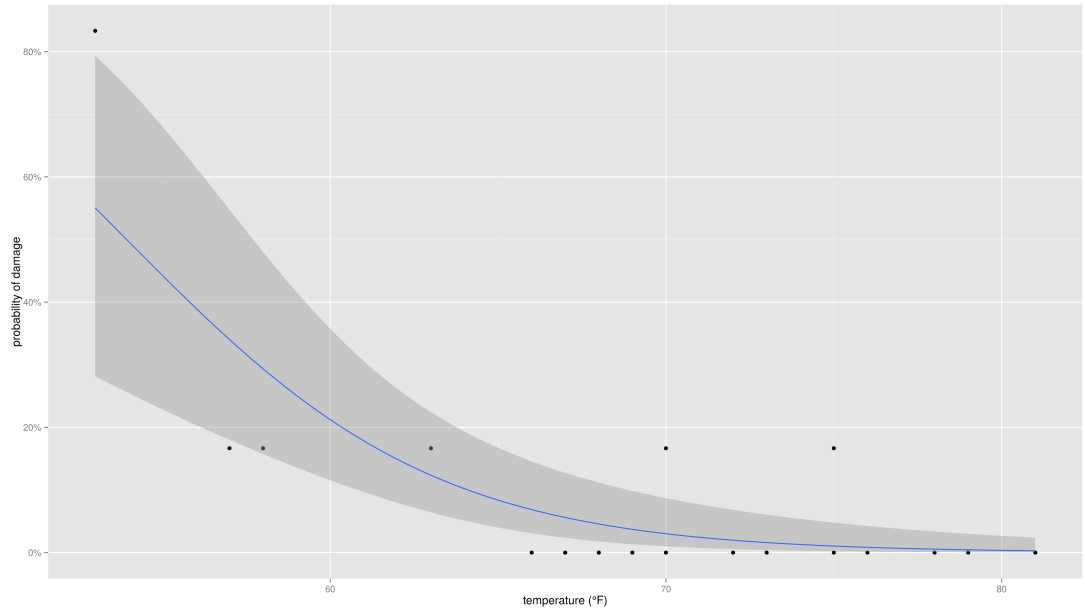
plot

```
> tf <- 20:100
> pd <- predict(m,newdata=list(temp=tf), type="response")
> plot(tf,pd,type="l",
+  xlab=expression(paste("temperature (",degree,"F)",sep=" ")),
+  ylab="probability of damage")
```



ggplot graphics

and the same plot made with ggplot



ggplot graphics

```
> orings$trials <- 6
> ggplot(orings,aes(x=temp,y=damage/trials)) +
+     geom_point() +
+     geom_smooth(method = "glm", family = "binomial", a
+     xlab(expression(paste("temperature (",degree,"F)")
+     ylab("probability of damage") +
+     scale_y_continuous(labels = percent)
```

Parasite Infection Example

- the binary response variable is parasite infection (infected or not)
- the explanatory variables are weight and age (continuous)
- and sex (categorical)
- we want to investigate if there is a different effect of age for each of the sexes on the outcome variable

```
> infection <- read.table("infection.txt",header=T)
> summary(infection)
    infected            age              sex
 Min.   :0.000   Min.   :  2.00   Min.   :0.000
 1st Qu.:0.000   1st Qu.: 46.00   1st Qu.:0.000
 Median :0.000   Median : 84.50   Median :1.000
 Mean   :0.324   Mean   : 93.69   Mean   :0.514
 3rd Qu.:1.000   3rd Qu.:139.25   3rd Qu.:1.000
 Max.   :1.000   Max.   :200.00   Max.   :1.000
```

Parasite Infection Example

```
> m.inf <- glm(infected~age*sex,family=binomial,
+                                  data=infection)
> summary(m.inf)

Call:
glm(formula = infected ~ age * sex, family = binomial, data = inf

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0411  -0.7307  -0.4363   0.6632   2.3215

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.000513   0.413639  -7.254 4.05e-13 ***
age           0.015657   0.003176   4.929 8.25e-07 ***
sexfemale     0.116664   0.553956   0.211   0.8332
age:sexfemale 0.011050   0.004612   2.396   0.0166 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Parasite Infection Example

- so for male at a age of 0 there is a probability of

```
> invlogit(coef(m.inf)[1])
(Intercept)
 0.04740269
```

- for females is the probability at age 0

```
> invlogit(coef(m.inf)[1]+coef(m.inf)[3])
(Intercept)
 0.05295775
```

Parasite Infection Example

- so what about the slope?
- for males the underlying model is the following

$$\Pr(\text{infection}) = \text{logit}^{-1}(-3.000513 + 0.015657 \cdot \text{age})$$

- for females the slope is almost twice as high

$$\Pr(\text{infection}) = \text{logit}^{-1}(-2.883849 + 0.02670685 \cdot \text{age})$$

- we can compare them by looking at the age where the probability to be infected is 50%

Parasite Infection Example

- this is the case when $-3.000513 + 0.015657 \cdot \text{age} = 0$ respectively $-2.883849 + 0.02670685 \cdot \text{age} = 0$; you can do it by hand or use R

```
> ## male
> solve(0.015657,3.000513)
[1] 191.6404
> ## female
> solve(0.02670685,2.883849)
[1] 107.9816
```

- `solve()` solves systems of linear equations in the form A*x=b, where A is the matrix of cœfficients and b are the (negative) intercepts, here we have the special case with just one equation

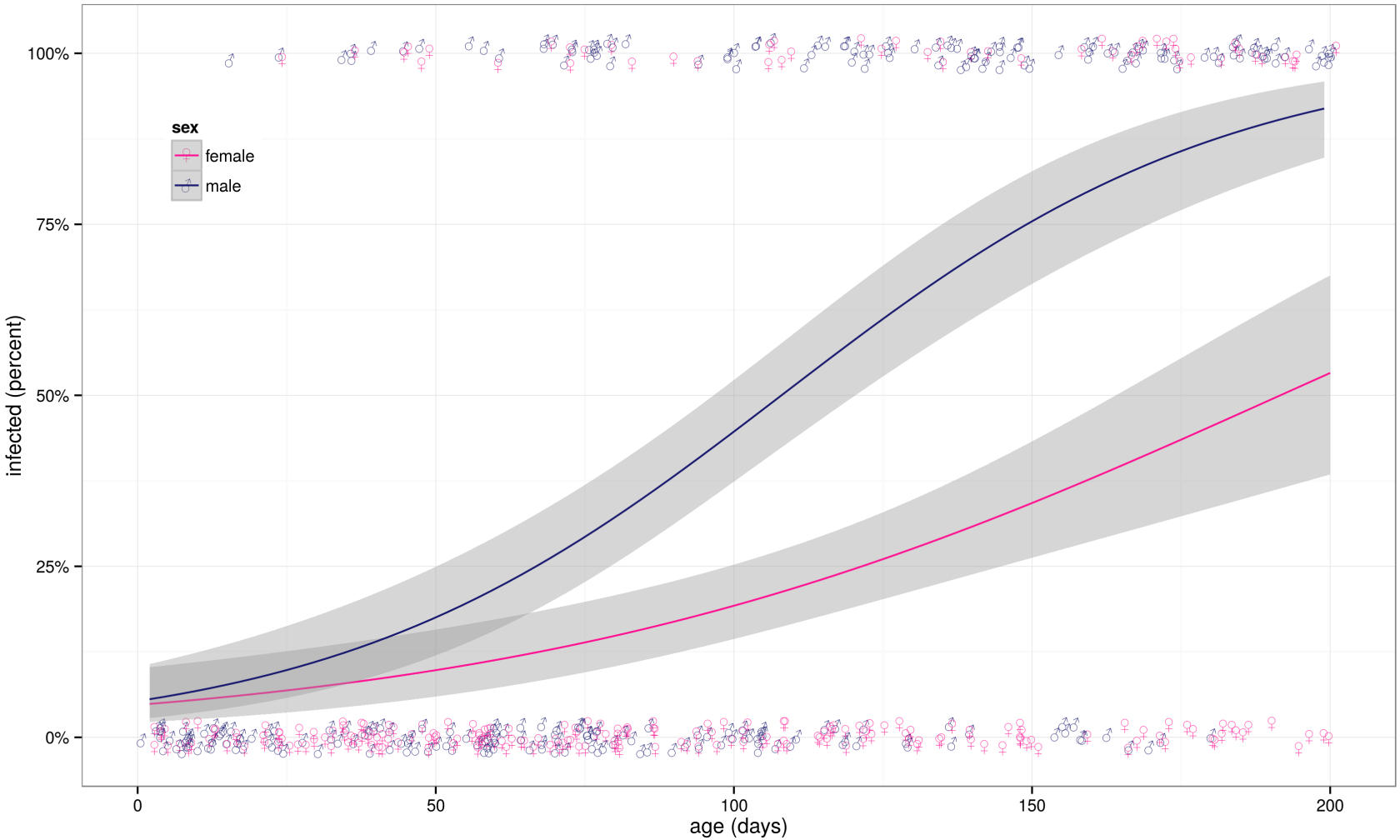
Parasite Infection Example

- you can also use the `allEffects()` function (part of the `effects` package), which give you the probabilities for being infected on several ages for both sexes

```
> allEffects(m)
 model: infected ~ age * sex


 age*sex effect
     sex
age             0          1
  2   0.04883687 0.05570148
  24  0.06756215 0.09596497
  46  0.09276694 0.16038932
  68  0.12610300 0.25582483
  90  0.16918450 0.38219715
  112 0.22322468 0.52680374
  134 0.28853152 0.66704908
  156 0.36399154 0.78286130
  178 0.44679328 0.86645480
  200 0.53265591 0.92110968
```

Parasite Infection Example



Exercise

Try to reproduce the plot! Hints:

1. set up a ggplot object, think about the æsthetics (`aes()`). Which quality of the graph you wanna set to which variable?
2. begin with the lines (`geom_smooth()`)
3. add the points (`geom_jitter()`; do not think about the symbols in the first place; try to adjust the width and height appropriately)
4. change the colour of the lines and points (`scale_colour_manual()`); I used midnightblue for male and deeppink for female
5. change the symbols (`scale_shape_manual()`); use

```
values = c("male" = "\u2642","female" = "\u2640")
```

   as values
6. set the axes titles
7. change to text of the y axis to percentage
8. etc

Exercise

1. load the data using

   ```
   cuse <- read.table("http://data.princeton.edu/wws509/datasets/c
   header=TRUE)
   ```

2. you can find a short description here:
   `http://data.princeton.edu/wws509/datasets/#cuse`
3. model the use of contraceptiva dependent on age, education, and the wish for more children
4. what are the probabilities for each age group? For each education level? for the levels of WantsMore?
5. add an interaction effect age:wants More
6. how does the effects change?
7. does the interaction term improves the model significantly? (look at the help for `anova.glm`)

Table of Contents I

Mixed Model I

- the mixed model looks very similar to ANOVA:

$$Y_{ij} = \mu + \alpha_j + S_i + Error_{ij}$$

- the difference is the assumption:
  - μ is a fixed effect: no estimate of variability
  - α_j are also fixed effects: no estimate of variability
  - S_i are random effects have normal distribution with zero mean
  - E_{ij} are normally distributed with mean zero (as usual)

Mixed Model I

- this also can be written as follows:
  - $Y_{ij} = [\pi_{0i} + \pi_{2i} \cdot \mathrm{TIME}_{ij}] + [\epsilon_{ij}]$ (structural part + stochastic part)
  - Y_{ij} is the value of subject i at time j as a linear function of TIME
  - π_{0i} and π_{2i} are individual parameters that characterize its shape for the ith subject and where

$$\pi_{0i} = \gamma_{00} + \gamma_{01}\mathrm{X}_i + \xi_{0i}$$

and

$$\pi_{1i} = \gamma_{10} + \gamma_{11}\mathrm{X}_i + \xi_{1i}$$

where

  - γ_{00} and γ_{10} the level-2 intercepts, represent the population average initial status and rate of change where $\mathrm{X} = 0$

Mixed Model II

- the level-2 slopes γ_{01} and γ_{11} represent the treatment effect X on the intercept and the slope respectively
- ξ_{0i} and ξ_{1i}, the level-2 residuals represent those portions of initial status or rate of change that are unexplained at level-2, they represent deviations of the individual change trajectories around their respective group average trends
- ξ_{0i} and ξ_{1i} are assumed to be independently draw from a bivariate normal distribution with mean 0 and variances σ_0^2 and σ_1^2

packages

- load the `lme4` package
- the example data are can be found in the `nlmeU` package

```
> require(lme4)
Lade nötiges Paket: lme4
Lade nötiges Paket: Matrix
> require(nlmeU)
Lade nötiges Paket: nlmeU

Attache Paket: 'nlmeU'

Das folgende Objekt ist maskiert 'package:lme4':

    sigma
> head(armd)
  subject treat.f visual0 miss.pat time.f time visual tp
2       1  Active      59     --XX   4wks    4     55  1
3       1  Active      59     --XX  12wks   12     45  2
5       2  Active      65     ----   4wks    4     70  1
6       2  Active      65     ----  12wks   12     65  2
```

Linear Mixed Models - Example

$$\begin{aligned}\text{VISUAL}_{it} = \beta_0 &+ \beta_1 \cdot \text{VISUAL0}_i + \beta_2 \cdot \text{TIME}_{it} \\ &+ \beta_3 \cdot \text{TREAT}_i \\ &+ \beta_4 \cdot \text{TREAT}_i \cdot \text{TIME}_{it} \\ &+ b_{0i} + \epsilon_{it}\end{aligned}$$

where the terms

- on the left-hand side
  - VISUAL_{it} denotes the visual acuity in patent i at time t
- in the fixed-effects part
  - VISUAL0_i is the value of visual acuity at baseline
  - TIME_{it} time of measurement t
  - TREAT_i is the treatment indicator
  - $\text{TREAT}_i \cdot \text{TIME}_{it}$ is their interaction

Linear Mixed Models

$$\begin{aligned}\text{VISUAL}_{it} = \beta_0 &+ \beta_1 \cdot \text{VISUAL0}_i + \beta_2 \cdot \text{TIME}_{it} \\ &+ \beta_3 \cdot \text{TREAT}_i \\ &+ \beta_4 \cdot \text{TREAT}_i \cdot \text{TIME}_{it} \\ &+ b_{0i} + \epsilon_{it}\end{aligned}$$

and the parameters

- in the fixed-effects part
  - β_0 is the overall intercept
  - β_1 describes the change in the mean visual acuity due to an unit increase in visual acuity at baseline
  - β_2 change due to a one week change in time
  - β_3 gives an overall treatment effect
  - β_4 describes an additional change due to a one week change in time for patients treated

Linear Mixed Models

$$\begin{aligned}\text{VISUAL}_{it} = \beta_0 &+ \beta_1 \cdot \text{VISUAL0}_i + \beta_2 \cdot \text{TIME}_{it} \\ &+ \beta_3 \cdot \text{TREAT}_i \\ &+ \beta_4 \cdot \text{TREAT}_i \cdot \text{TIME}_{it} \\ &+ b_{0i} + \epsilon_{it}\end{aligned}$$

the terms

- in the random-effects part
  - b_{0i} is the patient-specific random intercept assumed to be normally distributed with mean 0 and variance d_{11}
  - ϵ_{it} is a residual random error assumed to be normally distributed with mean 0 and variance σ^2

Random intercept

- formally speaking
  - the random intercept b_{0i} is a subject-specific deviation from the fixed intercept β_0
  - b_{0i} and β_0 are coupled
  - they both contribute to the subject-specific intercept

nlme and lme4 packages

- there are mainly the lme4 and the nlme package for fitting hierarchical models
- nlme (lme())is older (programmed in S3) and provides a wider range of variance structures
- lme4 (lmer())is older (programmed in S4), can be used for large-scale computational problems
- lmer() especially suitable to fit LMMs with crossed random effects but also hierarchical nested structures
- they are not compatible, because they are programmed in different systems the methods dœs not match

lme4 packages - lmer()

- `lmer()` function is the central function to fit linear mixed models
- the general syntax is `lmer(formula, syntax)`
- the formula itself comprises again the continuous response variable on the left-hand side and the explanatory variables on the right-hand side, they are separated by a $\sim$
- the fixed-effects on the right-hand side consists of one or more terms separated by $+$ symbols and they have the same syntax as in the classical formula

lme4 packages - lmer()

- in contrast the Z-terms are enclosed in parenthesis and are used to specify the part of the model involving random effects
- every Z-term is associated with a grouping factor

```
> f1 <- formula(visual ~ visual0 + time + treat.f + ## main eff
+                  treat.f:time + ## interaction
+                  (1|subject))  ## random effect
> f1
visual ~ visual0 + time + treat.f + treat.f:time + (1 | subject
```

- note that the formula of the Z-term – if not explicitly defined — implicitly assumes an inclusion of the intercept, you have to explicitly exclude it if you do not want one

Random effects

Syntax Z-term(s)	comment
$(1\|g1)$	Random intercepts only, 1x1 D matrix
$(z1\|g1)$	Random intercepts and slopes, general D matrix
$(1\|g1) + (0 + z1\|g1)$	Random intercepts and slopes, diagonal D matrix
$(1\|g1) + (1\|g12)$	Nested random effects $(g1/g12)$
$(1\|g1) + (1\|g1 : g2)$	Nested random effects $(g1/g2)$
$(1\|g1/g2)$	Nested random effects $(g1/g2)$
$(1\|g1) + (1\|g2)$	Crossed random effects

Extractor functions for a `lmerMod`

model fit component	extractor function
Summary	`m1.sum <- summary(m1)`
Print	`show(m1)`
Estimation method	`isREML(m1)`
$\hat{\beta}$ coefficients	`fixef(m1)`
$\hat{\beta}$, se($\hat{\beta}$), t-test	`coef(m1.sum)`
$Var(\hat{\beta})$	`vcov(m1)`
$\hat{b}$	`ranef(m1)`
$\hat{\beta} + coupled(\hat{b})$	`coef(m1)`

Confidence intervals

- to compute confidence intervals the `profile()` function can be used
- they can extracted by using `confint()` on the resulting object

```
> m1.prall <- profile(m1)
> confint(m1.prall)
                           2.5 %      97.5 %
.sig01                 7.9291593 10.01110528
.sigma                 8.1598181  9.11262421
(Intercept)            4.0454125 14.53187448
visual0                0.7390912  0.91376265
time                  -0.2571424 -0.16726552
treat.fActive         -5.3533430  0.50956354
time:treat.fActive    -0.1153109  0.01624458
```

Confidence intervals

- to compute confidence intervals the `profile()` function can be used
- they can extracted by using `confint()` on the resulting object

```
> confint(logProf(m1.prall))
                            2.5 %      97.5 %
.lsig01                 2.0705467  2.30369486
.lsigma                 2.0992218  2.20966072
(Intercept)             4.0454125 14.53187448
visual0                 0.7390912  0.91376265
time                   -0.2571424 -0.16726552
treat.fActive          -5.3533430  0.50956354
time:treat.fActive     -0.1153109  0.01624458
```

Testing random effects

- `exactRLRT()` from the `RLRsim` package for testing random effects
- arguments:
  - `m` - model-fit object of class `lmerMod`
  - `m0, mA` - only for models with multiple variance components where `mA` provides the model under the alternative hypothesis, model under the null

```
> require(RLRsim)
Lade nötiges Paket: RLRsim
> exactRLRT(m1)

simulated finite sample distribution of RLRT.

(p-value based on 10000 simulated values)

data:
RLRT = 249.97, p-value < 2.2e-16
```

Example m1 - random intercept

- fit model including three main effects and one interaction term as fixed effects
- random intercept

```
> f1 <- formula(visual ~ visual0 + time + treat.f + ## main effects
+                 treat.f:time + ## interaction
+                 (1|subject))  ## random effect
> m1 <- lmer(f1, data = armd)
```

Example m1 - extract fixed effects

- extract the fixed effects

```
> fixef(m1)
       (Intercept)             visual0                time      treat.fActive
         9.28807836          0.82643987         -0.21221595         -2.42200013
time:treat.fActive
        -0.04959058
```

Example m1 - extract fixed effects

- extract the fixed effects including SEs and t-tests

```
> coef(summary(m1))
                      Estimate Std. Error   t value
(Intercept)         9.28807836 2.68188866  3.463260
visual0             0.82643987 0.04466700 18.502245
time               -0.21221595 0.02292950 -9.255150
treat.fActive      -2.42200013 1.49996662 -1.614703
time:treat.fActive -0.04959058 0.03356171 -1.477594
```

Example m1 - random effects

```
> head(ranef(m1)[[1]])
  (Intercept)
1  -2.4160726
2   7.4638445
3  -6.2569087
4   4.2421197
6   0.3213384
7  10.1155516
```

Example m1 - broom package

```
> require(broom)
> glance(m1)
     sigma    logLik      AIC      BIC deviance df.residual
1 8.627515 -3288.986 6591.971 6625.326 6565.681         860
> m1.tidy <- tidy(m1)
> head(m1.tidy)
    group level        term  estimate
1 subject     1 (Intercept)  6.872006
2 subject     2 (Intercept) 16.751923
3 subject     3 (Intercept)  3.031170
4 subject     4 (Intercept) 13.530198
5 subject     6 (Intercept)  9.609417
6 subject     7 (Intercept) 19.403630
> table(m1.tidy$term)

       (Intercept)             visual0                time      treat.fActive
               234                 234                 234                 234
time:treat.fActive
               234
```

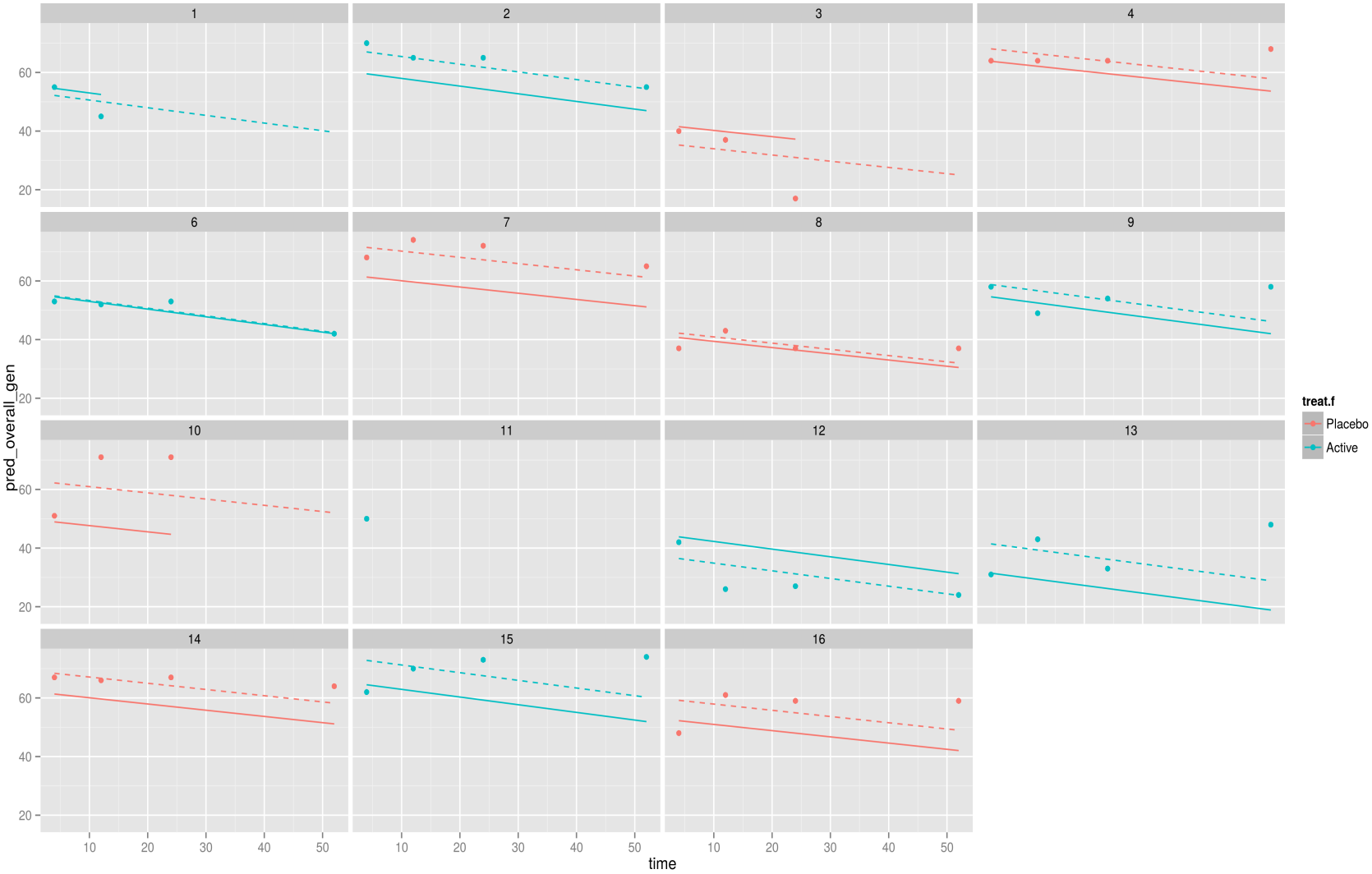
Example m1 - different kinds of predictions

```
> ## no random effects
> armd$pred_overall_gen <- predict(m1, type='response', re.form=NA)
> ## all random effects
> armd$pred_overall <- predict(m1, type='response')
> ## specified random effect
> armd$pred_subj <- predict(m1, type='response', re.form=~ (1|subject)
> ##
this changes row order from the fit!!!
> armd_pred <- merge(armd, as.data.frame(table(subject=armd$subject)))
```

Example m1 - different kinds of predictions

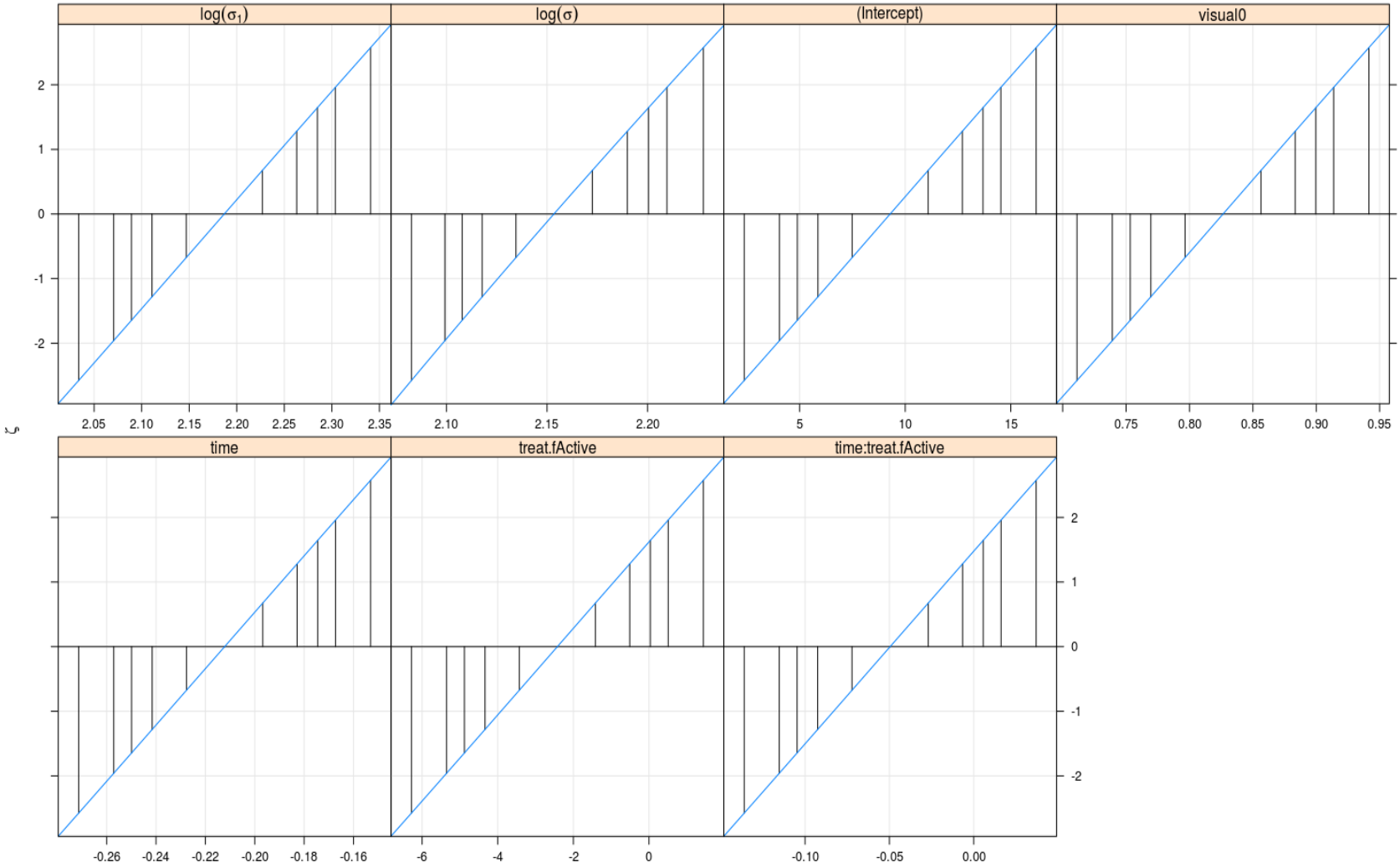
```
> head(armd_pred)
  subject treat.f visual0 miss.pat time.f time visual tp pred_overall_gen
1       1  Active      59     --XX   4wks    4     55  1          54.57880
2       1  Active      59     --XX  12wks   12     45  2          52.48435
3      10 Placebo      49     ---X   4wks    4     51  1          48.93477
4      10 Placebo      49     ---X  12wks   12     71  2          47.23704
5      10 Placebo      49     ---X  24wks   24     71  3          44.69045
6     100 Placebo      54     --X-   4wks    4     32  1          53.06697
  pred_overall pred_subj Freq
1     52.16273  52.16273    2
2     50.06828  50.06828    2
3     62.22367  62.22367    3
4     60.52595  60.52595    3
5     57.97935  57.97935    3
6     36.41897  36.41897    3
```

Observed and predicted values



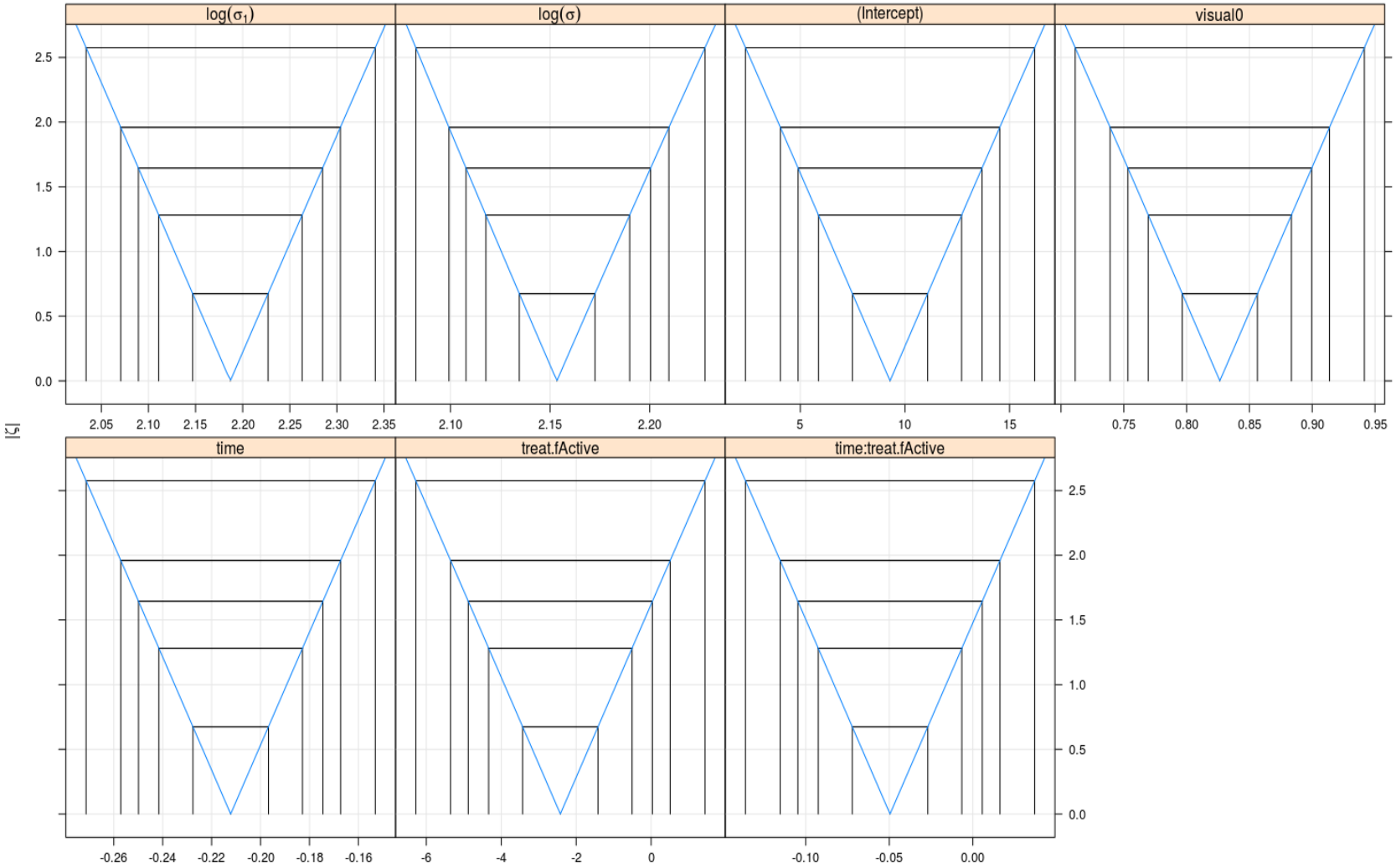
Diagnostics

```
> require(lattice)
> xyplot(logProf(m1.prall),as.table = T)
```



Diagnostics

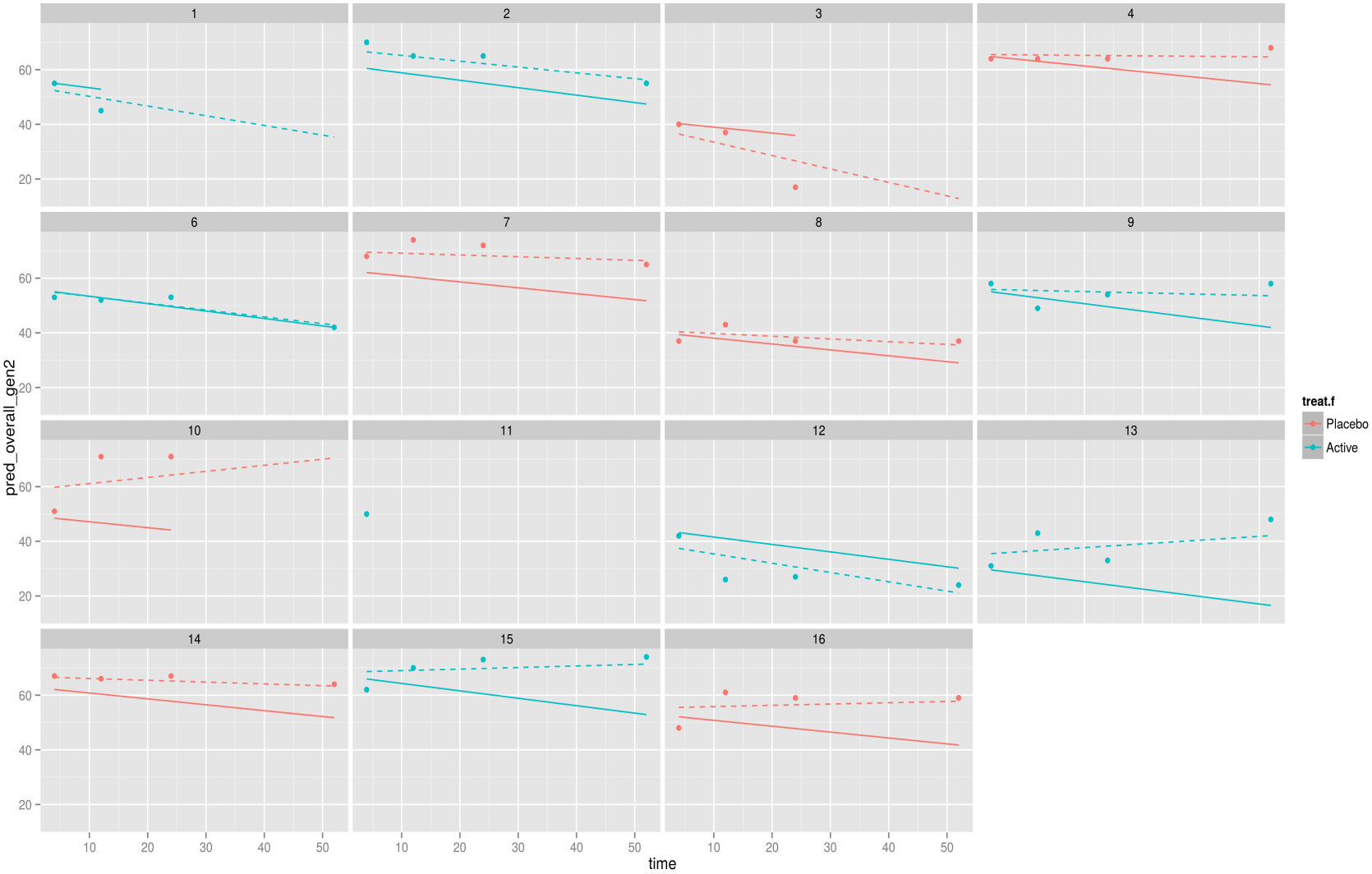
```
> xyplot(logProf(m1.prall),absVal = T,as.table = T)
```



Add random slope

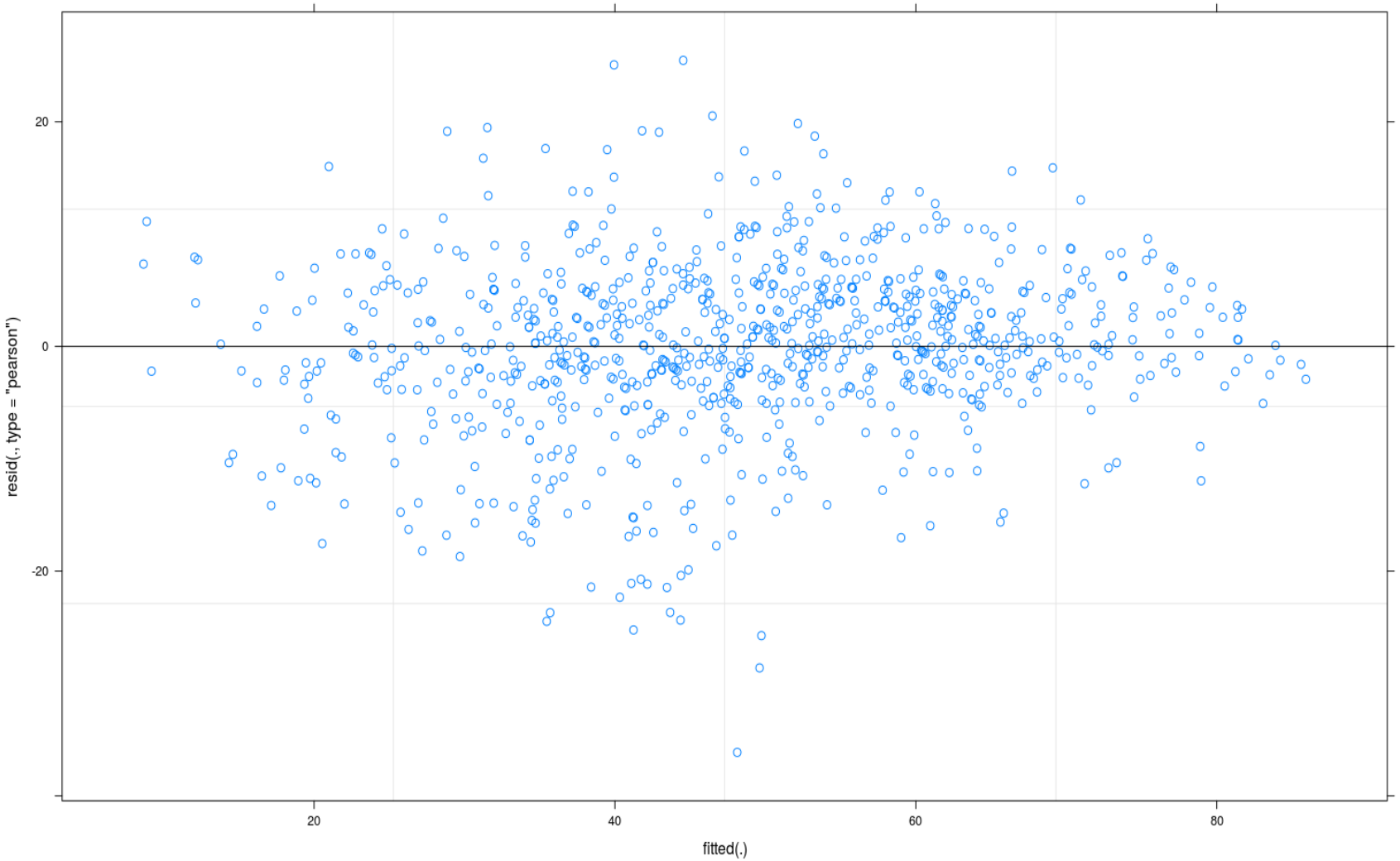
```
> ## add random slope
> f2 <- formula(visual ~ visual0 + time + treat.f + ## main
                  treat.f:time + ## interaction
                  (1 + time |subject))  ## random effect
> m2 <- lmer(f2, data = armd)
```

Observed and predicted values



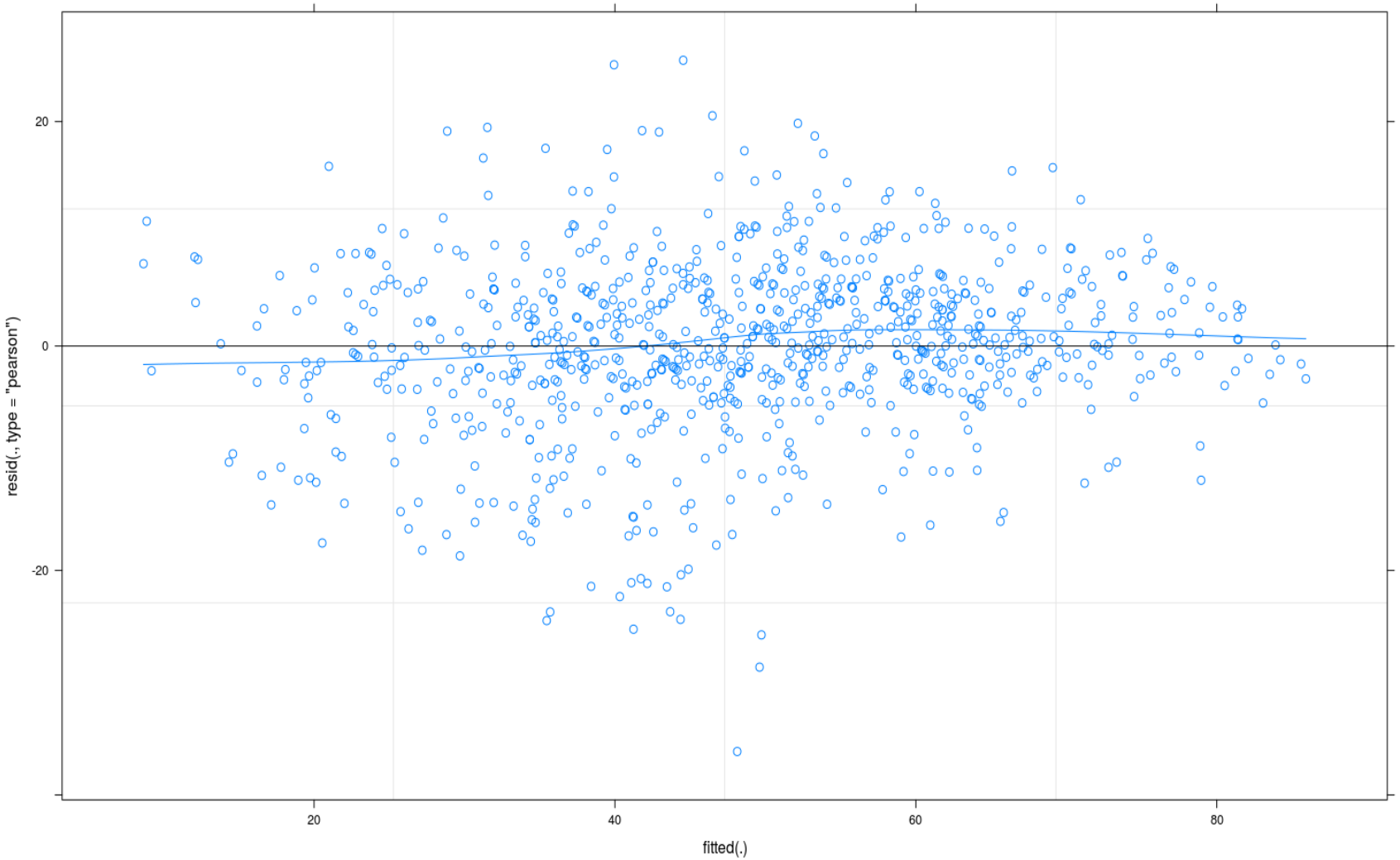
Default residual plot

```
> plot(m1)
```



Default residual plot

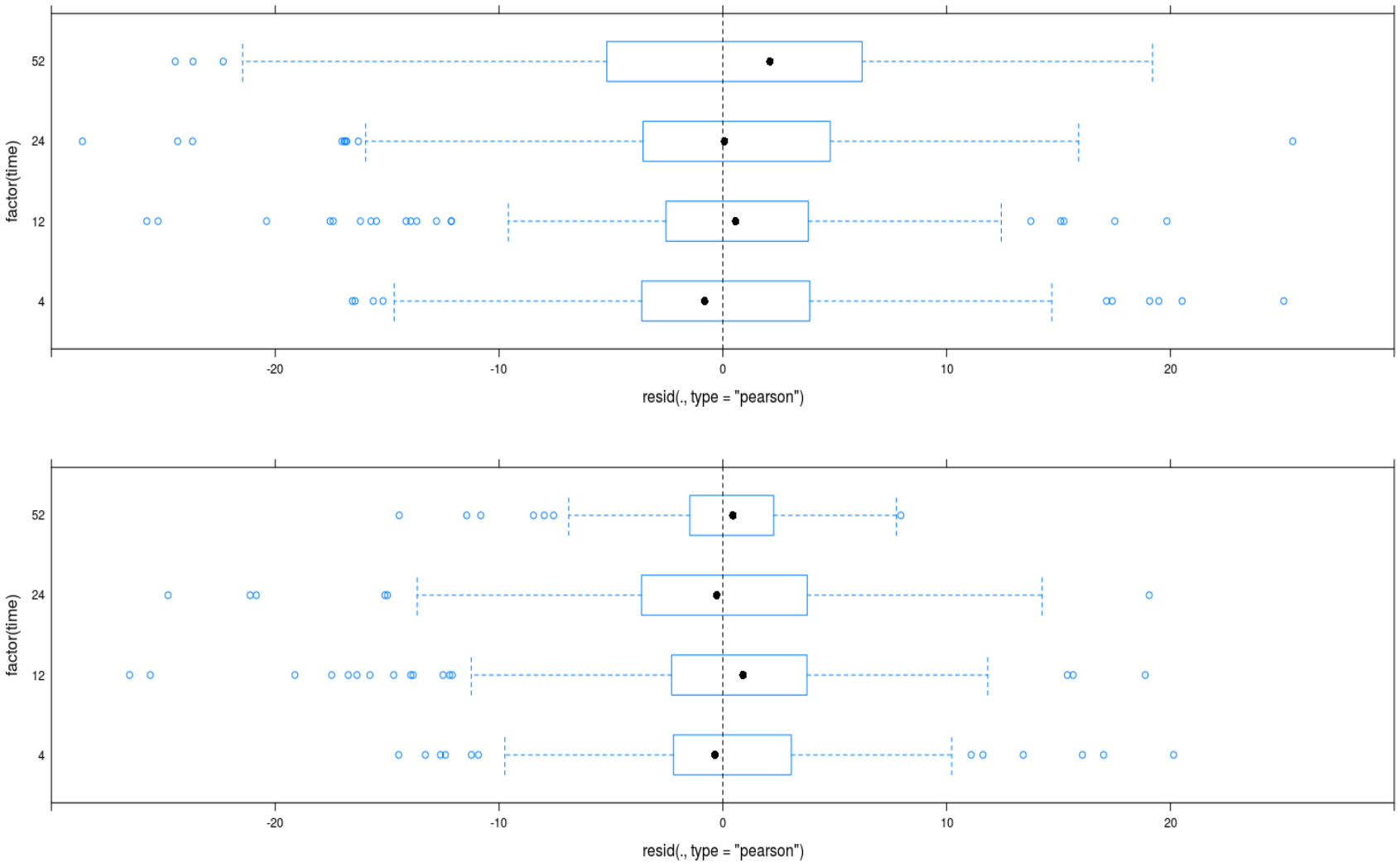
```
> plot(m1, type = c("p","smooth"))
```



Default residual plot

```
> p1 <- plot(m1, factor(time) ~ resid(., type = "pearson"),
+            abline=c(v=0), lty=2, xlim = c(-30,30))
> p2 <- plot(m2, factor(time) ~ resid(., type = "pearson"),
+            abline=c(v=0), lty=2, xlim = c(-30,30))
> require(gridExtra)
> grid.arrange(p1,p2,nrow = 2)
```

Default residual plot



The multcomp package

- testing several linear hypothesis simultaneously
- given a model the `glht()` function can do pairwise comparisons like Tukey's all pairwise comparisons or Dunnett's many-to-one
- different methods for p-value adjustment
- methods available for `lm`, `glm`, `mer` and other
- has its own plot method

The multcomp package

- the linear hypotheses can be set up using the `linfct` argument
- in general `linfct` is a matrix with as many columns as cœfficients in the data
- this matrix can also generated by a symbolic description
- or a predefined procedure as Dunnett or Tukey

The multcomp package - Example

- default behavior

```
> m1.mc

 General Linear Hypotheses

Linear Hypotheses:
                                Estimate
(Intercept) == 0                 4.40518
visual0 == 0                     0.82642
time.f.L == 0                   -7.71034
time.f.Q == 0                    0.29799
time.f.C == 0                    0.73909
treat.fActive == 0              -3.56133
time.f.L:treat.fActive == 0     -1.80230
time.f.Q:treat.fActive == 0     -0.09443
time.f.C:treat.fActive == 0     -0.84549
```

The multcomp package – Example

- default behavior - `summary()`

```
> summary(m1.mc)

 Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = visual ~ visual0 + time.f + treat.f + treat.f:time.f +
    (1 | subject), data = armd)

Linear Hypotheses:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept) == 0                 4.40518    2.63624   1.671   0.5157
visual0 == 0                     0.82642    0.04467  18.500   <0.001 ***
time.f.L == 0                   -7.71034    0.83659  -9.216   <0.001 ***
time.f.Q == 0                    0.29799    0.81759   0.364   0.9999
time.f.C == 0                    0.73909    0.80467   0.919   0.9585
treat.fActive == 0              -3.56133    1.32084  -2.696   0.0536 .
time.f.L:treat.fActive == 0     -1.80230    1.22545  -1.471   0.6669
time.f.Q:treat.fActive == 0     -0.09443    1.18388  -0.080   1.0000
time.f.C:treat.fActive == 0     -0.84549    1.15870  -0.730   0.9893
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

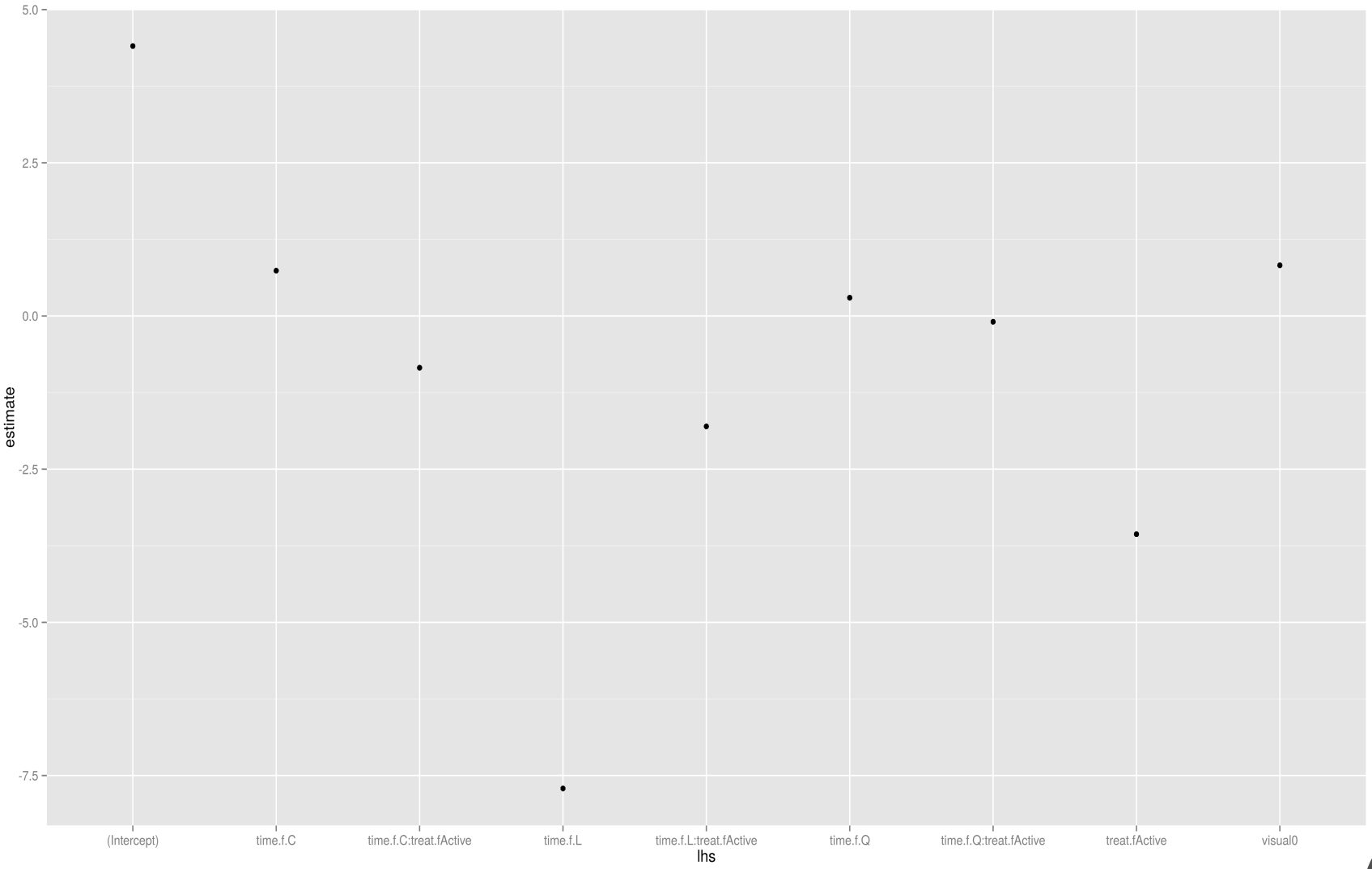
The multcomp package – Example

- default behavior - `broom`

```
> m1.mc.tidy <- tidy(summary(m1.mc))
> m1.mc.tidy
                      lhs rhs    estimate  std.error   statistic    p.value
1             (Intercept)   0  4.40518292 2.63624152  1.67100886 0.51568844
2                  visual0   0  0.82641634 0.04467229 18.49952831 0.00000000
3                time.f.L   0 -7.71034040 0.83659340 -9.21635336 0.00000000
4                time.f.Q   0  0.29799443 0.81758517  0.36448121 0.99992527
5                time.f.C   0  0.73909162 0.80466880  0.91850413 0.95848936
6           treat.fActive   0 -3.56132619 1.32084386 -2.69625071 0.05344545
7 time.f.L:treat.fActive   0 -1.80230304 1.22545080 -1.47072657 0.66686541
8 time.f.Q:treat.fActive   0 -0.09442963 1.18387660 -0.07976307 1.00000000
9 time.f.C:treat.fActive   0 -0.84549028 1.15870330 -0.72968661 0.98929805
```

The multcomp package – Example

```
> ggplot(m1.mc.tidy, aes(x = lhs, y = estimate)) +
+   geom_point()
```



The multcomp package – Example

- confidence intervals - `broom`

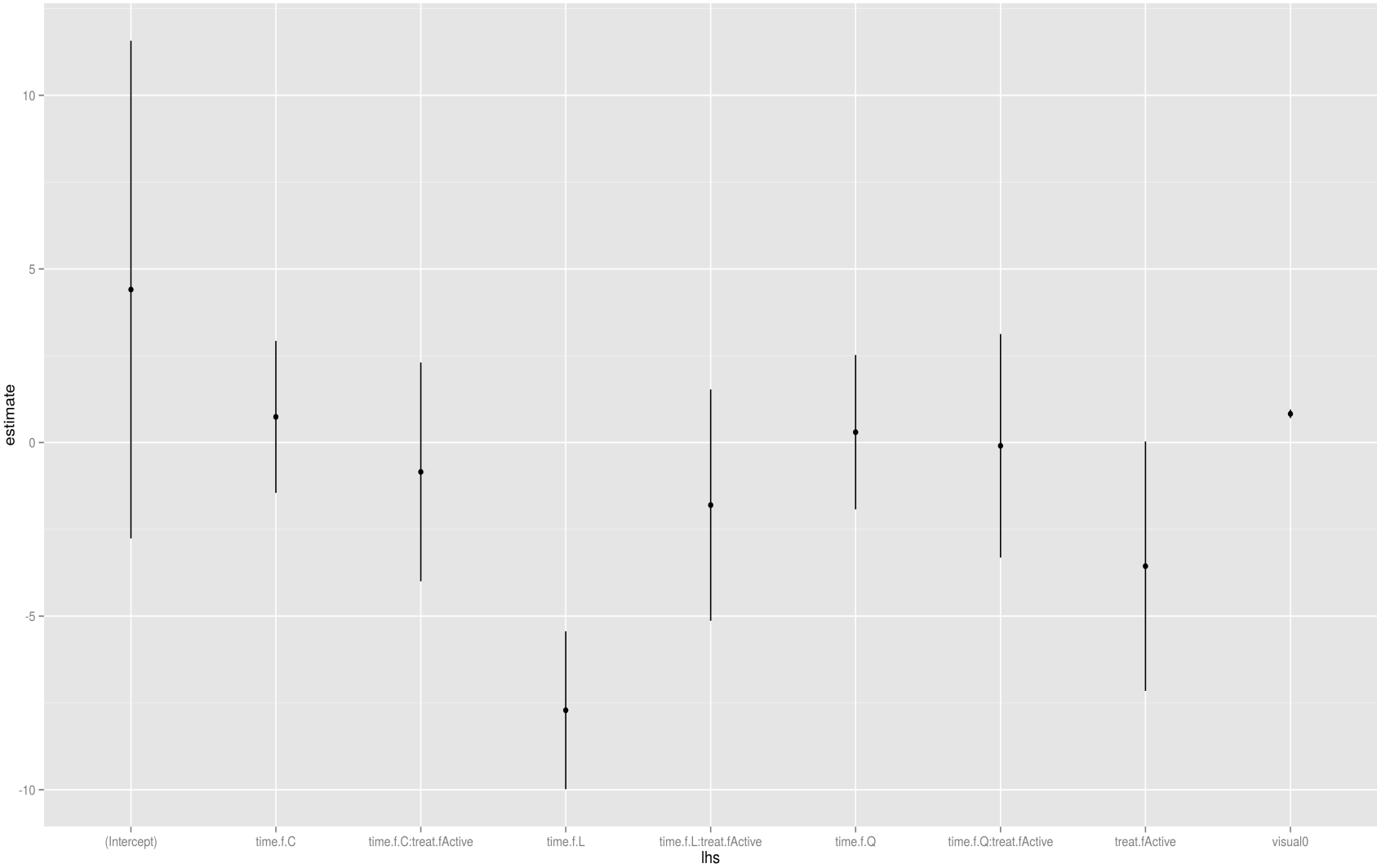
```
> m1.mc.ci <- tidy(confint(m1.mc))
> m1.mc.ci
                      lhs rhs    estimate   conf.low  conf.high
1             (Intercept)   0  4.40518292 -2.7642829 11.5746487
2                 visual0   0  0.82641634  0.7049265  0.9479061
3               time.f.L   0 -7.71034040 -9.9855218 -5.4351590
4               time.f.Q   0  0.29799443 -1.9254926  2.5214815
5               time.f.C   0  0.73909162 -1.4492684  2.9274516
6          treat.fActive   0 -3.56132619 -7.1534648  0.0308124
7 time.f.L:treat.fActive   0 -1.80230304 -5.1350127  1.5304066
8 time.f.Q:treat.fActive   0 -0.09442963 -3.3140750  3.1252157
9 time.f.C:treat.fActive   0 -0.84549028 -3.9966749  2.3056943
```

The multcomp package – Example

- plotting confidence intervals using ggplot

```
> ggplot(m1.mc.ci, aes(x = lhs,
+                      y = estimate,
+                      ymin = conf.low,
+                      ymax = conf.high)) +
+                        geom_point() +
+                        geom_pointrange()
```

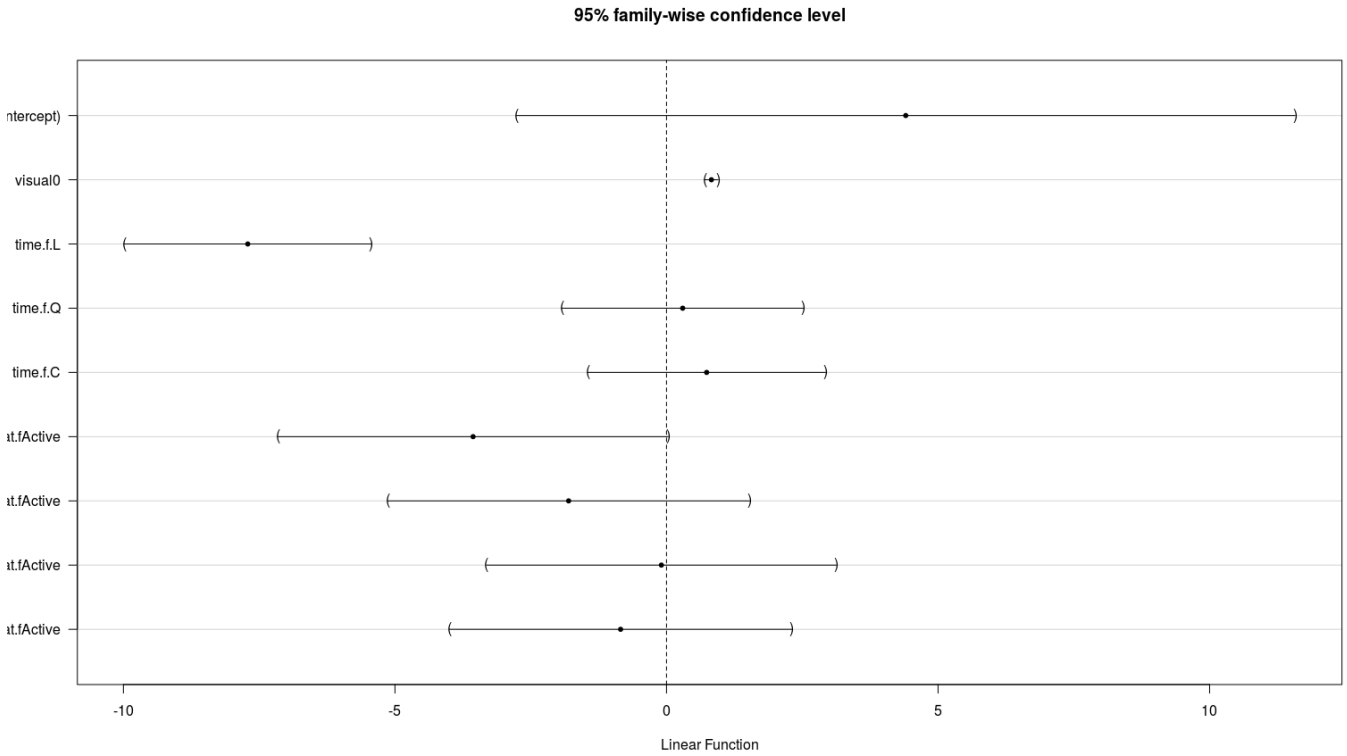
The multcomp package – Example



The multcomp package – Example

- default plot

```
> plot(m1.mc)
```



The multcomp package – Example

- defining contrasts: first refitting the model using time coded as factor

```
> f1 <- formula(visual ~ visual0 + time.f + treat.f + ## main effects
+                 treat.f:time.f + ## interaction
+                 (1|subject))  ## random effect
>
> m1 <- lmer(f1, data = armd)> plot(m1.mc)
```

The multcomp package - Example

- Tukey's pairwise comparisons

```
> glht(m1,linfct = mcp(time.f = "Tukey"))

 General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Linear Hypotheses:
                     Estimate
12wks - 4wks == 0     -1.030
24wks - 4wks == 0     -4.668
52wks - 4wks == 0     -9.904
24wks - 12wks == 0    -3.637
52wks - 12wks == 0    -8.874
52wks - 24wks == 0    -5.236

Warnmeldung:
In mcp2matrix(model, linfct = linfct) :
  covariate interactions found -- default contrast might be inappropriate
```

The multcomp package - Example

- Dunnett's

```
> glht(m1,linfct = mcp(time.f = "Dunnett"))

 General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts


Linear Hypotheses:
                   Estimate
12wks - 4wks == 0   -1.030
24wks - 4wks == 0   -4.668
52wks - 4wks == 0   -9.904

Warnmeldung:
In mcp2matrix(model, linfct = linfct) :
  covariate interactions found -- default contrast might be inappropriate
```

The multcomp package - Example

- taking interaction into account

```
> glht(m1,linfct = mcp(time.f = "Tukey", interaction_average = T))

 General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Linear Hypotheses:
                     Estimate
12wks - 4wks == 0      -1.698
24wks - 4wks == 0      -5.078
52wks - 4wks == 0     -11.274
24wks - 12wks == 0     -3.379
52wks - 12wks == 0     -9.576
52wks - 24wks == 0     -6.196
```

The multcomp package – Example

- extract contrast matrix

```
> glht(m1,linfct = mcp(time.f = "Tukey"))$linfct
              (Intercept) visual0  time.f.L   time.f.Q    time.f.C
12wks - 4wks            0       0 0.2198599 -0.7273795  1.19271985
24wks - 4wks            0       0 0.5496497 -1.3017908 -0.05679618
52wks - 4wks            0       0 1.3191593 -0.2310132  0.45436947
24wks - 12wks           0       0 0.3297898 -0.5744113 -1.24951604
52wks - 12wks           0       0 1.0992994  0.4963663 -0.73835038
52wks - 24wks           0       0 0.7695096  1.0707776  0.51116565
              treat.fActive time.f.L:treat.fActive time.f.Q:treat.fActive
12wks - 4wks              0                      0                      0
24wks - 4wks              0                      0                      0
52wks - 4wks              0                      0                      0
24wks - 12wks             0                      0                      0
52wks - 12wks             0                      0                      0
52wks - 24wks             0                      0                      0
              time.f.C:treat.fActive
12wks - 4wks                       0
24wks - 4wks                       0
52wks - 4wks                       0
24wks - 12wks                      0
52wks - 12wks                      0
52wks - 24wks                      0
attr(,"type")
[1] "Tukey"
Warnmeldung:
In mcp2matrix(model, linfct = linfct) :
  covariate interactions found -- default contrast might be inappropriate
```

The multcomp package – Example

- extract contrast matrix

```
> glht(m1,linfct = mcp(time.f = "Tukey", interaction_average = T))$linfct
              (Intercept) visual0  time.f.L   time.f.Q    time.f.C
12wks - 4wks            0       0 0.2198599 -0.7273795  1.19271985
24wks - 4wks            0       0 0.5496497 -1.3017908 -0.05679618
52wks - 4wks            0       0 1.3191593 -0.2310132  0.45436947
24wks - 12wks           0       0 0.3297898 -0.5744113 -1.24951604
52wks - 12wks           0       0 1.0992994  0.4963663 -0.73835038
52wks - 24wks           0       0 0.7695096  1.0707776  0.51116565
              treat.fActive time.f.L:treat.fActive time.f.Q:treat.fActive
12wks - 4wks              0              0.1099299             -0.3636898
24wks - 4wks              0              0.2748249             -0.6508954
52wks - 4wks              0              0.6595797             -0.1155066
24wks - 12wks             0              0.1648949             -0.2872057
52wks - 12wks             0              0.5496497              0.2481831
52wks - 24wks             0              0.3847548              0.5353888
              time.f.C:treat.fActive
12wks - 4wks              0.59635993
24wks - 4wks             -0.02839809
52wks - 4wks              0.22718473
24wks - 12wks            -0.62475802
52wks - 12wks            -0.36917519
52wks - 24wks             0.25558283
```

The multcomp package – Example

- recode model with interaction term

```
> f3 <- formula(visual ~ visual0 + ia +
+                 (1|subject))  ## random effect
> m3 <- lmer(f3, data = armd)
> summary(glht(m3,linfct = mcp(ia = "Tukey")))

 Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lmer(formula = visual ~ visual0 + ia + (1 | subject), data = armd)

Linear Hypotheses:
                                  Estimate Std. Error z value Pr(>|z|)
Active.4wks - Placebo.4wks == 0     -2.33744    1.63676  -1.428  0.82875
Placebo.12wks - Placebo.4wks == 0   -1.03042    1.13248  -0.910  0.98305
Active.12wks - Placebo.4wks == 0    -4.70386    1.64785  -2.855  0.07396 .
Placebo.24wks - Placebo.4wks == 0   -4.66789    1.14890  -4.063  0.00109 **
Active.24wks - Placebo.4wks == 0    -7.82501    1.66935  -4.687  < 0.001 ***
Placebo.52wks - Placebo.4wks == 0   -9.90419    1.17306  -8.443  < 0.001 ***
Active.52wks - Placebo.4wks == 0   -14.98150    1.70556  -8.784  < 0.001 ***
Placebo.12wks - Active.4wks == 0     1.30701    1.63713   0.798  0.99219
Active.12wks - Active.4wks == 0     -2.36642    1.16539  -2.031  0.43484
Placebo.24wks - Active.4wks == 0    -2.33046    1.64845  -1.414  0.83618
Active.24wks - Active.4wks == 0     -5.48758    1.19310  -4.599  < 0.001 ***
Placebo.52wks - Active.4wks == 0    -7.56675    1.66544  -4.543  < 0.001 ***
Active.52wks - Active.4wks == 0    -12.64406    1.24348 -10.168  < 0.001 ***
Active.12wks - Placebo.12wks == 0   -3.67344    1.64823  -2.229  0.31174
...
```

The multcomp package – Example

- change adjustment method

```
> summary(glht(m3,linfct = mcp(ia = "Tukey"),test = "fdr"))

 Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lmer(formula = visual ~ visual0 + ia + (1 | subject), data = armd)

Linear Hypotheses:
                                   Estimate Std. Error z value Pr(>|z|)
Active.4wks - Placebo.4wks == 0     -2.33744    1.63676  -1.428  0.82854
Placebo.12wks - Placebo.4wks == 0   -1.03042    1.13248  -0.910  0.98309
Active.12wks - Placebo.4wks == 0    -4.70386    1.64785  -2.855  0.07393 .
Placebo.24wks - Placebo.4wks == 0   -4.66789    1.14890  -4.063  0.00103 **
Active.24wks - Placebo.4wks == 0    -7.82501    1.66935  -4.687  < 0.001 ***
Placebo.52wks - Placebo.4wks == 0   -9.90419    1.17306  -8.443  < 0.001 ***
Active.52wks - Placebo.4wks == 0   -14.98150    1.70556  -8.784  < 0.001 ***
Placebo.12wks - Active.4wks == 0     1.30701    1.63713   0.798  0.99219
Active.12wks - Active.4wks == 0     -2.36642    1.16539  -2.031  0.43516
Placebo.24wks - Active.4wks == 0    -2.33046    1.64845  -1.414  0.83602
Active.24wks - Active.4wks == 0     -5.48758    1.19310  -4.599  < 0.001 ***
Placebo.52wks - Active.4wks == 0    -7.56675    1.66544  -4.543  < 0.001 ***
Active.52wks - Active.4wks == 0    -12.64406    1.24348 -10.168  < 0.001 ***
Active.12wks - Placebo.12wks == 0   -3.67344    1.64823  -2.229  0.31079
...
```

The multcomp package - Example

- specify contrasts of interest

```
> mt3.1 <- glht(m3,linfct = mcp(ia = c("Active.52wks - Active.24wks = 0",
+                           "Placebo.52wks - Active.4wks = 0")))
> mt3.2 <- glht(m3,linfct = mcp(ia = c("Active.52wks - Active.24wks = 1",
+                              "Placebo.52wks - Active.4wks = 1")))
> mt3.3 <- glht(m3,linfct = mcp(ia = c("Active.52wks - Active.24wks >= 1",
+                              "Placebo.52wks - Active.4wks >= 1")))
> mt3.3 <- glht(m3,linfct = mcp(ia = c("Active.52wks - Active.24wks <= 1",
+                              "Placebo.52wks - Active.4wks <= 1")))
```

The multcomp package - Example

- testing simultaneously for two factors

```
> mat1 <- glht(m1,linfct = mcp(time.f = "Tukey", interaction_average = T))$linfct
> mat2 <- glht(m1,linfct = mcp(treat.f = "Tukey", interaction_average = T))$linfc
> mat <- rbind(mat1,mat2)
> summary(glht(m1,linfct = mat))

 Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = visual ~ visual0 + time.f + treat.f + treat.f:time.f +
    (1 | subject), data = armd)

Linear Hypotheses:
                       Estimate Std. Error z value Pr(>|z|)
12wks - 4wks == 0       -1.6984     0.8125  -2.090   0.1869
24wks - 4wks == 0       -5.0777     0.8282  -6.131   <0.001 ***
52wks - 4wks == 0      -11.2741     0.8547 -13.190   <0.001 ***
24wks - 12wks == 0      -3.3793     0.8286  -4.078   <0.001 ***
52wks - 12wks == 0      -9.5757     0.8559 -11.188   <0.001 ***
52wks - 24wks == 0      -6.1964     0.8661  -7.154   <0.001 ***
Active - Placebo == 0   -4.2469     1.4329  -2.964   0.0189 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

The multcomp package - Example

- compare with

```
> tidy(summary(glht(m1,linfct = mcp(time.f = "Tukey",
+      interaction_average = T))))
            lhs rhs   estimate std.error  statistic      p.value
1  12wks - 4wks   0  -1.698421 0.8125024  -2.090358 1.560901e-01
2  24wks - 4wks   0  -5.077734 0.8281697  -6.131272 2.606585e-09
3  52wks - 4wks   0 -11.274125 0.8547392 -13.190134 0.000000e+00
4 24wks - 12wks   0  -3.379313 0.8286354  -4.078167 3.123938e-04
5 52wks - 12wks   0  -9.575704 0.8558612 -11.188384 0.000000e+00
6 52wks - 24wks   0  -6.196391 0.8661023  -7.154341 1.818767e-12
> tidy(summary(glht(m1,linfct = mcp(treat.f = "Tukey",
+      interaction_average = T))))
               lhs rhs  estimate std.error statistic     p.value
1 Active - Placebo   0 -4.246882  1.432863 -2.963914 0.003037535
>
```

The multcomp package - Example

- using model matrix

```
> tmp <- expand.grid(time.f = levels(armd$time.f),
+                    treat.f = levels(armd$treat.f))
> tmp$time.f <- factor(tmp$time.f,ordered = T)
> cm <- model.matrix(~ time.f * treat.f, data = tmp)
> cm <- cbind(cm, visual0 = 0)
> glht(m1,linfct = cm)

 General Linear Hypotheses

Linear Hypotheses:
       Estimate
1 == 0   -0.071
2 == 0    8.275
3 == 0    8.245
4 == 0    1.171
5 == 0    2.177
6 == 0   10.649
7 == 0    9.152
8 == 0   -1.401
```