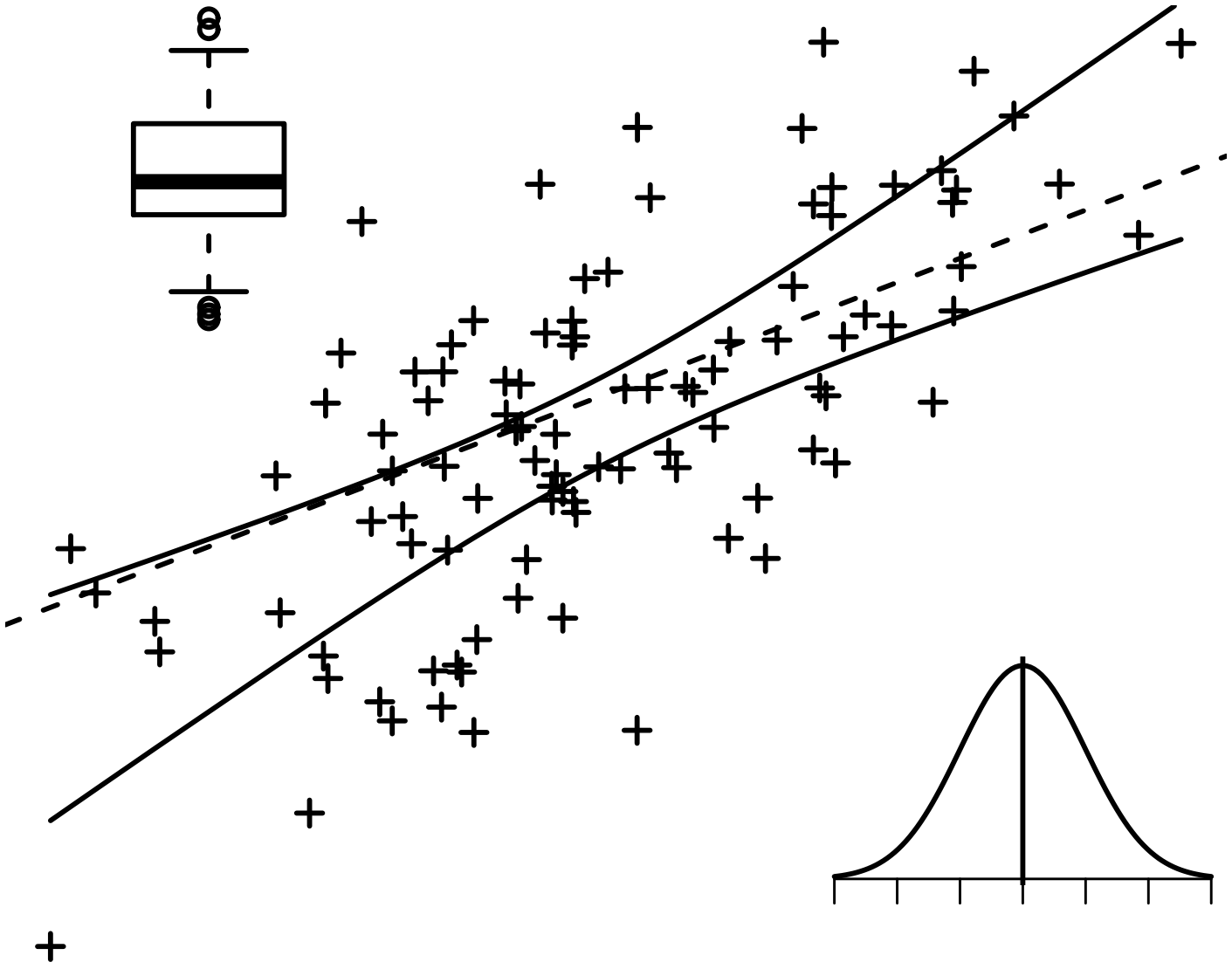
Mixed effects models — 2010



Oliver Kirchkamp

Contents

1 Introduction

Purpose of this handout In this handout you find all slides from the lecture (in a more printer friendly version). You also find (most of) the examples in R I plan to use in the lecture. Attached to the PDF file you find some datasets.

Homepage: http://www.kirchkamp.de/

Literature:

- Jose C. Pinheiro and Douglas M. Bates, Mixed Effects Models in S and S-Plus. Springer, 2002.
- Julian J. Faraway, Extending the Linear Model with R. Chapman & Hall, 2006.

Terminology Depending on the field, mixed effects models are known under different names:

- Mixed effects models
- Random effects models
- Hierarchical models
- Multilevel models

Why mixed effects models?

- Repeated observation of the same unit:
  - as part of a panel outside the lab
  - participant in the experiment
  - group of participants in an experiments
- Reasons for repeated observations:
  - within observational unit (participant/group) comparisons
  - study the dynamics of a process (market behaviour, convergence to equilibium,...)
  - allow "learning of the game"

A possible experiment Example: Repeated public good game

Question: is there a decay of contributions over time?

- participants in a group of four can contribute to a public good
- 8 repetitions
- random matching in groups of 12
- observe 10 matching groups (120 participants)

In our raw data we have $12 \times 8 \times 10 = 960$ observations.

Problems

- Repeated measurements
- always the same participants
- always the same matching groups

Observations are correlated — OLS requires uncorrelated ϵ

Solution A (inefficient):

- Aggregate over matching groups, use conservative tests (χ^2, rank-sum)

Disadvantage:

- Loss of power
- Control of individual properties only through randomisation

  (groups/participants might have different and known (even controlled) properties)

It would be nice to know:

- What is the treatment effect (in the example: the effect of time)
- What is an effect due to other observables (e.g. gender, risk aversion, social preferences)
- What is the heterogeneity of participants (due to unobservable differences)
- What is the heterogeneity of groups (e.g. due to contamination in the experiment)

Alternative (more efficient):

- Models with mixed effects

This example: OLS, fixed effects and random effects Indices:

- i individuals $1 \ldots 12$
- k group $1 \ldots 10$
- t time $1 \ldots 8$

- Standard OLS:
  $y_{ikt} = \beta_0 + \beta_1 x_{1,ikt} + \beta_2 x_{2,ikt} + \epsilon_{ikt}$
  with $\epsilon_{ikt} \sim N(0, \sigma)$
- Fixed effects for participants $i \times k$:
  $y_{ikt} = \beta_0 + \beta_1 x_{1,ikt} + \beta_2 x_{2,ikt} + \sum_{i,k} \gamma_{ik} d_{ik} + \epsilon_{ikt}$
  with $\epsilon_{ikt} \sim N(0, \sigma)$
- Random effects for participants $i \times k$:
  $y_{ikt} = \beta_0 + \beta_1 x_{1,ikt} + \beta_2 x_{2,ikt} + \nu_{ik} + \epsilon_{ikt}$
  with $\nu_{ik} \sim N(0, \sigma_\nu)$ and $\epsilon_{ikt} \sim N(0, \sigma)$

Fixed effects

\+ captures individual heterogeneity

− only describes heterogeneity in the sample (this is not a problem if sample heterogeneity is experimentally controlled, e.g. fixed effect for treatment 1 vs. treatment 2)

− less stable since many coefficients have to be estimated

\+ makes no distributional assumptions on heterogeneity

− can be fooled by spurious correlation among X and ν_{ik}

\+ unbiased if ν_{ik} and X are dependent

Random effects

\+ captures individual heterogeneity

\+ estimates heterogeneity in the population

\+ more stable since fewer coefficients have to be estimated

− makes distributional assumptions on heterogeneity

\+ exploits independence of ν_{ik} and X (if it can be assumed)

− biased if ν_{ik} and X are dependent

Terminology

$$y_{ikt} = \beta_0 + \beta_1 x_{1,ikt} + \beta_2 x_{2,ikt} + \nu_{ik} + \epsilon_{ikt}$$

- Random effects — units ik are selected randomly from a population. The effect is that the mean y depends on the choice of ik.
- Hierarchical/multilevel model — first we explain variance on the level of ik, then on the level of ikt.

2 Examples

During this course we will set one common variable, load a few libraries and load some data. The data is attached to the online version of this PDF:

```
bootstrapsize <- 100
library(lme4)
library(Ecdat)
library(car)
library(Hmisc)
library(geepack)
load(file = "data/me.Rdata")
```

2.1 A very simple example

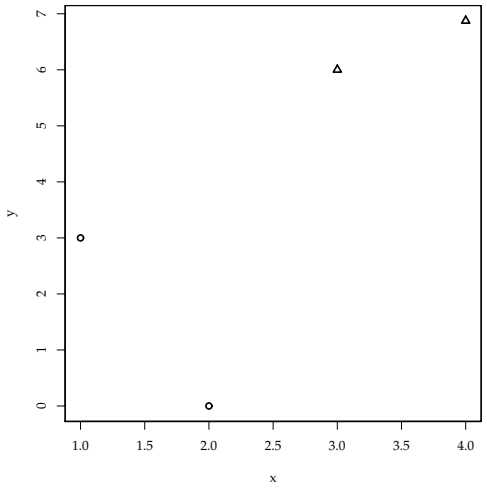
The following figure shows the predicted relationship for the various methods. The dataset is very simple. We have only four observations, two from two groups each. The first groups (shown as circles) are at the bottom left of the diagram, the second group (triangles) are top right.

```
simple <- as.data.frame(cbind(x = c(1, 2, 3, 4), y = c(3,
    0, 6, 6.8744), i = c(1, 1, 2, 2)))
```

```
simple
```

```
  x      y i
1 1 3.0000 1
2 2 0.0000 1
3 3 6.0000 2
4 4 6.8744 2
```

```
par(mar = c(4, 4, 0, 0))
plot(y ~ x, data = simple, pch = i)
```



- Standard OLS:

  $y_{ik} = \beta_0 + \beta_1 x_{ik} + \epsilon_{ik}$ with $\epsilon_{ik} \sim N(0, \sigma)$

  ```
  ol <- lm(y ~ x, data = simple)
  ```

- Between OLS:

  $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma)$

  ```
  betweenSimple <- with(simple, aggregate(simple, list(i),
      mean))
  betweenOLS <- lm(y ~ x, data = betweenSimple)
  ```

- Fixed effects for groups i:

  $y_{ik} = \beta_0 + \beta_1 x_{ik} + \sum_i \gamma_i d_i + \epsilon_{ik}$ with $\epsilon_{ik} \sim N(0, \sigma)$

  We also call the fixed effects model a "within" model, since only variance within the same group i matters.

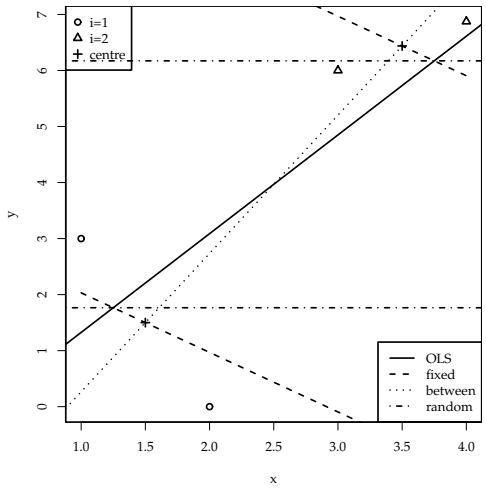
  ```
  fixef <- lm(y ~ x + as.factor(i), data = simple)
  ```

- Random effects for groups i:

  $y_{ik} = \beta_0 + \beta_1 x_{ik} + \nu_i + \epsilon_{ik}$ with $\nu_i \sim N(0, \sigma_\nu)$ and $\epsilon_{ik} \sim N(0, \sigma)$

  ```
  ranef <- lmer(y ~ x + (1 | i), data = simple)
  ```

```
par(mar = c(4, 4, 0, 0))
plot(y ~ x, data = simple, pch = i)
points(betweenSimple[, c("x", "y")], pch = 3)
legend("topleft", c("i=1", "i=2", "centre"), pch = 1:3,
    bg = "white")
abline(ol)
abline(betweenOLS, lty = 3)
qq <- sapply(unique(simple$i), function(g) lines(predict(fixef,
    newdata = within(simple, i <- g)) ~ x, data = simple,
    lty = 2))
qq <- sapply(ranef@ranef, function(r) abline(a = fixef(ranef)[1] +
    r, b = fixef(ranef)[2], lty = 4))
legend("bottomright", c("OLS", "fixed", "between", "random"),
    lty = 1:4)
```



- The between OLS estimator neglects any variance within groups and fits a line through the center (marked with a $+$) of each group.
- pooled OLS neglects any group specific effect and estimates a steeply increasing line. In a sense, OLS imposes an infinitely high cost on the fixed effect ν_i (setting them to 0) and, under this constraint, minimizes ϵ_{ikt}.

  Pooled OLS yields an estimation between the between OLS and the fixed effects estimator.

- Clustering is supposed to yield a better estimate for the standard errors, but does not change the estimate for the marginal effects.
- The fixed effect estimator neglects all the variance across groups and does not impose any cost on fixed effects. Here, the relationship within the two groups is decreasing on average, hence a negative slope is estimated.
- The random effect takes into account the ν_i and the ϵ_{ikt}. If the estimated slope is small (as with fixed effects) the ν_i are large (in absolute terms) and the ϵ_{ikt} are small, if the estimated slope is as large as with the OLS model, the ν_i are getting smaller but the ϵ_{ikt} are getting larger.

  The random effects yields an estimation between the fixed effect and the (pooled) OLS estimation.

2.2 A larger example

Consider the following relationship:
$y_{it} = x_{it} + \nu_i + \epsilon_{it}$
with $\nu_i \sim N(0, \sigma_\nu)$ and $\epsilon_{ikt} \sim N(0, \sigma)$
We simulate and test now the following methods

- between OLS
- pooled OLS
- clustered OLS
- non-parametric Wilcoxon test
- Fixed effects
- Random effects

```
set.seed(10)
I <- 6
T <- 50
i <- as.factor(rep(1:I, each = T))
ierr <- 15 * rep(rnorm(I), each = T)
uerr <- 3 * rnorm(I * T)
x <- runif(I * T)
y <- x + ierr + uerr
```

For comparison we will also construct a dependent variable *y2* without an individual specific random effect.

```
y2 <- x + 6 * uerr
```

We put them all in one dataset.

```
data <- as.data.frame(cbind(y, y2, x, i, ierr, uerr))
```

2.3 6 different methods - 6 different results

2.3.1 Pooled OLS

$$y_{it} = \beta_0 + \beta_1 x_{it} + \epsilon_{it} \quad \text{with } \epsilon_{ik} \sim N(0, \sigma)$$

```
ols <- lm(y ~ x, data = data)
summary(ols)
```

```
Call:
lm(formula = y ~ x, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-23.742  -4.895   2.283   7.343  16.896

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.826      1.092  -3.504  0.00053 ***
x              1.071      1.875   0.571  0.56828
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.515 on 298 degrees of freedom
Multiple R-squared: 0.001094,        Adjusted R-squared: -0.002258
F-statistic: 0.3263 on 1 and 298 DF,  p-value: 0.5683
```

- Estimation of β is consistent if residuals ϵ_{it} are uncorrelated with $\boldsymbol{X}$.
- With repeated observations (as in our case), estimation of σ is generally not consistent.

2.3.2 Clustered OLS

$$y_{it} = \beta_0 + \beta_1 x_{it} + \epsilon_{it} \quad \text{with } \epsilon_{ik} \sim N(0, \Sigma)$$

```
ols.cluster <- geeglm(y ~ x, id = i, data = data)
summary(ols.cluster)
```

```
Call:
geeglm(formula = y ~ x, data = data, id = i)

 Coefficients:
            Estimate Std.err  Wald Pr(>|W|)
(Intercept)  -3.8262  4.0798 0.880    0.348
x             1.0711  0.8007 1.789    0.181

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    89.94   36.74

Correlation: Structure = independenceNumber of clusters:   6   Maximum cluster s
```

- The estimated coefficients are the same as in the OLS model. Only the standard errors are different.
- Estimation of β is consistent if residuals (ϵ_{it}) are uncorrelated with $\boldsymbol{X}$.
- Estimation of Σ is better than with pooled OLS (still problematic for a small number of clusters. Convergence is $O(\sum_{j=1}^{C} N_j^2 / N^2)$

See Kézdi, Gábor, 2004, Robust Standard Error Estimation in Fixed-Effects Panel Models; Rogers, Regression standard errors in clustered samples, STB 13, .

2.3.3 Between OLS

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \text{with } \epsilon_i \sim N(0, \sigma)$$

```
data.between <- aggregate(data, list(data$i), mean)
ols.between <- lm(y ~ x, data = data.between)
summary(ols.between)
```

```
Call:
lm(formula = y ~ x, data = data.between)

Residuals:
      1       2       3       4       5       6
  3.341   0.942 -16.880  -5.413   8.087   9.922

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    -4.15      68.15   -0.06     0.95
x               1.72     135.10    0.01     0.99

Residual standard error: 11.1 on 4 degrees of freedom
Multiple R-squared: 4.03e-05,        Adjusted R-squared: -0.25
F-statistic: 0.000161 on 1 and 4 DF,  p-value: 0.99
```

- Estimation of β is consistent if residuals (ν_i) are uncorrelated with $\boldsymbol{X}$.
- Σ can not be estimated.
- The method is inefficient since the variance within a group is not exploited.

2.3.4 Non-parametric Wilcoxon Test

$$y_{it} = \beta_{0i} + \beta_{1i} x_{it} + \epsilon_{it} \quad \text{with } \epsilon_{ik} \sim N(0, \sigma_i)$$

```
estBetax <- sapply(by(data, list(i = data$i), function(data) lm(y ~
    x, data = data)), coef)["x", ]
mean(estBetax)
```

```
[1] 1.049
```

```
wilcox.test(estBetax)
```

```
        Wilcoxon signed rank test

data:  estBetax
V = 16, p-value = 0.3125
alternative hypothesis: true location is not equal to 0
```

- β can be estimated as the mean of the β_i as long as residuals ϵ_{it} are uncorrelated with $\boldsymbol{X}_i$.
- σ is not estimated.
- Efficiency $\rightarrow$ less efficient than fixed or random effects, since we do not exploit any relative differences.

2.3.5 Fixed effects

$$y_{it} = \beta_0 + \beta_1 x_{it} + \sum_i \gamma_i d_i + \epsilon_{it}$$

```
fixed <- lm(y ~ x + as.factor(i) - 1, data = data)
summary(fixed)
```

```
Call:
lm(formula = y ~ x + as.factor(i) - 1, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6167 -2.1856  0.0641  2.1945  7.1029

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
x                1.063      0.576    1.84    0.066 .
as.factor(i)1   -0.447      0.521   -0.86    0.392
as.factor(i)2   -2.879      0.503   -5.72  2.6e-08 ***
as.factor(i)3  -20.698      0.505  -40.96  < 2e-16 ***
as.factor(i)4   -9.253      0.494  -18.75  < 2e-16 ***
as.factor(i)5    4.232      0.486    8.70  2.4e-16 ***
as.factor(i)6    6.114      0.510   11.99  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.91 on 293 degrees of freedom
Multiple R-squared: 0.918,        Adjusted R-squared: 0.916
F-statistic:  470 on 7 and 293 DF,  p-value: <2e-16
```

- Estimation of β is consistent if residuals (ϵ_{it}) are uncorrelated with $\boldsymbol{X}$. This is a weaker requirement, since, with fixed effects, residuals are only ϵ_{ikt}, not ν_i.
- Estimation of σ is consistent.
- The procedure looses some efficiency, since all the d_i are exactly estimated (although we are not interested in d_i).

Exercise 2.1 *The file* `ex1.csv` *contains observations on* `x1`, `x2`, `y` *and a group variable* `group`. *You are interested in how* `x1` *and* `x2` *influence* `y`. *Estimate the following models, compare their coefficients and standard errors:*

- *Pooled OLS*
- *Pooled OLS with clustered errors*
- *Between OLS*
- *Fixed Effects*

2.3.6 Random effects

$$y_{it} = \beta_0 + \beta_1 x_{it} + \nu_i + \epsilon_{it}$$

```
random <- lmer(y ~ x + (1 | i), data = data)
summary(random)
```

```
Linear mixed model fit by REML
Formula: y ~ x + (1 | i)
   Data: data
  AIC  BIC logLik deviance REMLdev
 1530 1545   -761     1528    1522
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 97.86    9.89
 Residual              8.44    2.91
Number of obs: 300, groups: i, 6

Fixed effects:
            Estimate Std. Error t value
```

```
(Intercept)   -3.822      4.052   -0.94
x              1.063      0.576    1.84

Correlation of Fixed Effects:
  (Intr)
x -0.072
```

- Estimation of β is consistent if residuals v_i and ϵ_{it} are uncorrelated with $\boldsymbol{X}$.
- This is a stronger requirement than with fixed effects, since we also impose a restriction on v_i.

  (e.g., what, if participants self select into treatments?)

Exercise 2.2 *Have another look at the data from* `ex1.csv`*. Now also estimate a model with a random effect for groups.*

2.4 The power of the 6 methods

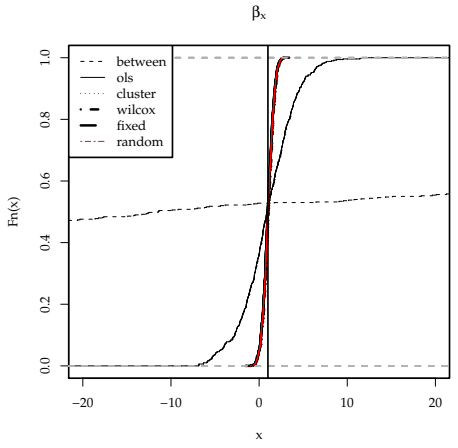
We repeat the above exercise 500 times. Each time we look at the estimated coefficient β_x and at the p-value of testing $\beta_x = 0$ against $\beta_x \neq 0$.

Note: the "true" $\beta_x = 1$

Here are mean and standard deviations for β_x for the six methods:

	between	ols	cluster	wilcox	fixed	random.x
mean	-11.71	0.94	0.94	1.00	1.00	1.00
sd	214.28	3.03	3.03	0.61	0.61	0.61

The figure shows the distribution of estimated β_x for the different methods:

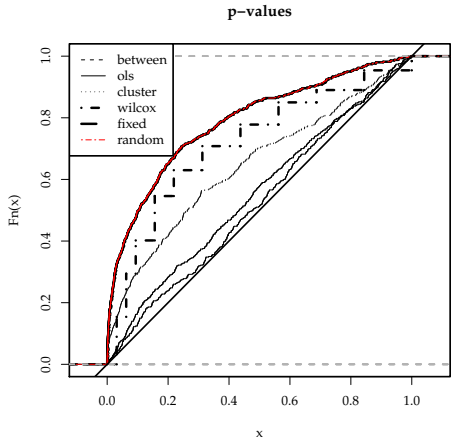


The good news is: All estimators seem to be unbiased (although the between estimator has a huge variance here). Also OLS and clustered OLS are not very efficient. Sometimes they estimate values that are far away from the true value $\beta_x = 1$.

Another desirable property of an estimator might be to find a significant effect if there is, indeed, a relationship.

Here is the relative frequency (in percent) to find in our simulation a p-value smaller than 5%:

between	ols	cluster	wilcox	fixed	random.x
6.80	7.80	20.60	15.40	37.40	37.60



Note that all five methods worked with the same data. Still, the fixed and random effects method were more successful in finding a significant relationship.

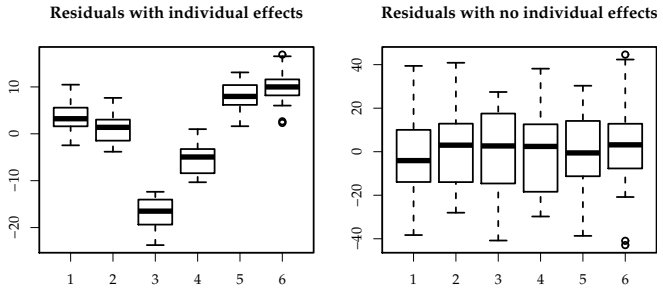
2.5 Residuals

In the above exercise we actually knew the correct relationship. How can we discover the need for a fixed- or random effects model from our data.

2.5.1 OLS residuals

Let us have a look at the residuals of the OLS estimation:

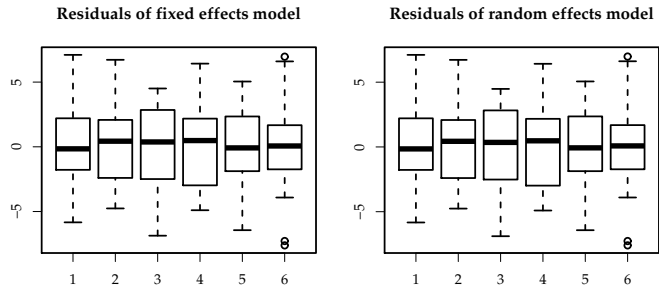
```
ols2 <- lm(y2 ~ x, data = data)
par(mfrow = c(1, 2), mar = c(4, 4, 4, 0))
boxplot(residuals(ols) ~ i, main = "Residuals with individual effects")
boxplot(residuals(ols2) ~ i, main = "Residuals with no individual effects")
```



The left graph shows the residuals for the model where we do have individual specific effects, the right graph shows residuals for the *y2* model without such effects.

2.5.2 Fixed- and random effects residuals

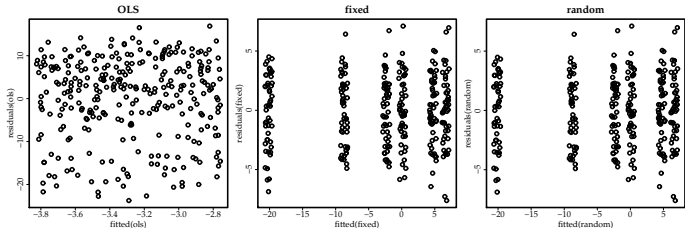
```
par(mfrow = c(1, 2), mar = c(4, 4, 4, 0))
boxplot(residuals(fixed) ~ i, main = "Residuals of fixed effects model")
boxplot(residuals(random) ~ i, main = "Residuals of random effects model")
```



2.5.3 Distribution of residuals over fitted values

Let us also look at the distribution of residuals over fitted values. We have to check that the standard error of residuals does not depend on X. One way to do this is to check that the standard error does not depend on $\hat{Y}$ which is linear in X:

```
par(mfrow = c(1, 3), mar = c(4, 6, 4, 0), mex = 0.5)
plot(residuals(ols) ~ fitted(ols), main = "OLS")
plot(residuals(fixed) ~ fitted(fixed), main = "fixed")
plot(residuals(random) ~ fitted(random), main = "random")
```



Exercise 2.3 *What can you say about the distribution of residuals of your estimates for* `ex1.csv`*?*

2.5.4 Estimated standard errors

$$y_{it} = \beta_0 + \beta_1 x_{it} + \nu_i + \epsilon_{it}$$

Let us compare the estimated standard errors of the residuals ϵ_{ikt}

```
summary(ols)$sigma
```

```
[1] 9.515
```

```
summary(fixed)$sigma
```

```
[1] 2.905
```

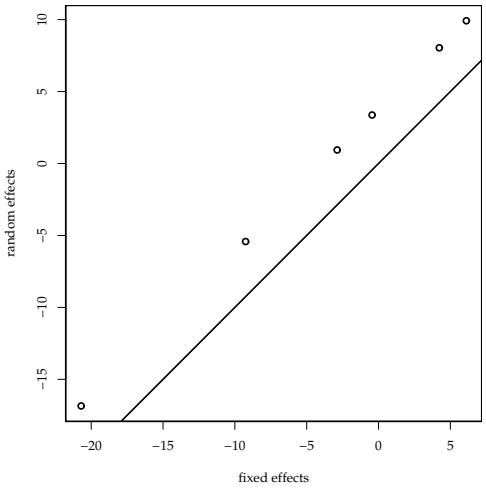
```
summary(random)@sigma
```

```
sigmaREML
    2.905
```

Here the estimated standard errors of the random and fixed effects model are similar. This need not be the case (and here is due to the fact that the sample is balanced).

2.5.5 Estimated effects

```
par(mar = c(4, 4, 0, 0))
plot(coef(fixed)[-1], ranef(random)$i[, "(Intercept)"],
    xlab = "fixed effects", ylab = "random effects")
abline(a = 0, b = 1)
```



We see that the estimated effects for the fixed effects and for the random effects model are similar. The variance of the random effects model is smaller.

2.5.6 Information criteria

$$AIC = -2 \log L + 2k$$

```
AIC(ols)
```

```
[1] 2207
```

```
AIC(fixed)
```

```
[1] 1500
```

When we want to compare the models, we have to use ML also for the random effects model. Usually random effects models are estimated with a different method, REML.

```
randomML <- update(random, REML = FALSE)
summary(randomML)@AICtab
```

```
  AIC  BIC logLik deviance REMLdev
 1535 1550 -763.7     1527    1522
```

2.6 Hausman test

- Fixed effects estimator is consistent but inefficient
- Random effects estimator is efficient, but only consistent if ν_i is not correlated with X.

In an experiment we can often rule out such a correlation through the experimental design. Then using random effects is not problematic. With field data matters can be less obvious.

If we don't know whether ν_i and X are correlated:

- Compare the time varying coefficents of fixed and random effects estimators:

$$\operatorname{var}(\hat{\beta}_{FE} - \hat{\beta}_{RE}) = \operatorname{var}(\hat{\beta}_{FE}) - \operatorname{var}(\hat{\beta}_{RE}) = \Psi$$

$$H = (\hat{\beta}_{FE} - \hat{\beta}_{RE})' \Psi^{-1} (\hat{\beta}_{FE} - \hat{\beta}_{RE}) \sim \chi^2_K$$

We can define a little function that compares two models:

```
hausman
```

```
function(fixed,random) {
  rNames <- names(random@fixef)
  fNames <- names(coef(fixed))
  timevarNames <- intersect(rNames,fNames)
  k <- length(timevarNames)
  rV <- vcov(random)
  rownames(rV)=rNames
  colnames(rV)=rNames
  bDiff <- (random@fixef)[timevarNames] - coef(fixed)[timevarNames]
  vDiff <- vcov(fixed)[timevarNames,timevarNames] - rV[timevarNames,timevarNames]
  (H <- t(bDiff) %*% solve(vDiff) %*% bDiff)
  c(H=H,p.value=pchisq(H,k,lower.tail=FALSE))
}
```

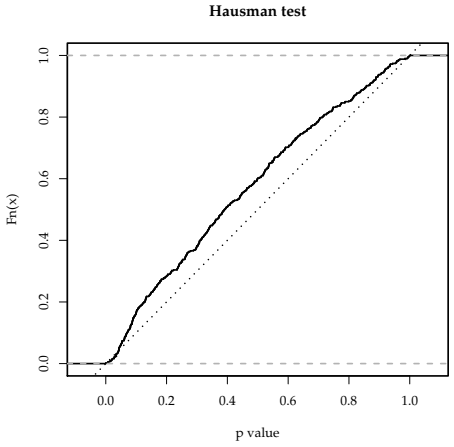
```
hausman(fixed, random)
```

```
        H    p.value
0.00002952 0.99566506
```

We see that in our example there is no reason not to use random effects. (For completeness: We are looking here at the difference of two variance-covariance matrices, hence it is possible that the Hausman statistic becomes negative)

Is the Hausman test conservative? The following graph extends the above Monte Carlo exercise. For each of the 500 simulated datasets we carry out a Hausman test and compare the random with the fixed effects model. The distribution of the estimated p-values is shown in the following graph.

```
hm1 <- sapply(simul1, function(x) x[, 7])
plot(ecdf(hm1["p", ]), do.points = FALSE, verticals = TRUE,
    main = "Hausman test", xlab = "p value")
abline(a = 0, b = 1, lty = 3)
```



Since (by construction of the dataset) there is no correlation between the random ν_i and the x, the p-value should be uniformly distributed between 0 and 1. We see that this is not the case. E.g. we obtain in 16.4% of all cases a p-value smaller than 10%.

Exercise 2.4 *Use a Hausman test to compare the fixed and the random effects model for the dataset* `ex1.csv`.

2.7 Testing random effects

Do we really have a random effect? How can we test this?

Idea: Likelihood ratio test (this works for testing random effects, this does not work very well if we want to test fixed effects).

generally

$$\chi^2_k \sim 2 \cdot (\log(L_{\text{large}}) - \log(L_{\text{small}}))$$

here

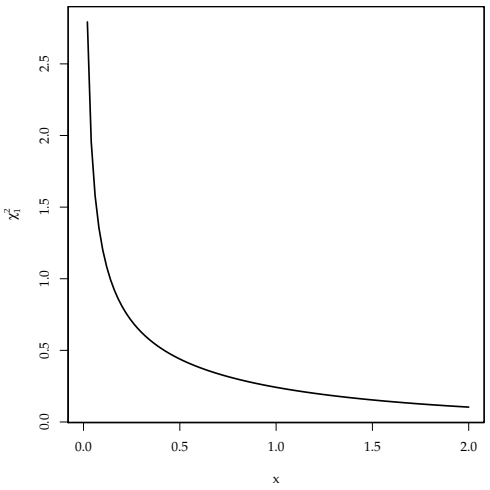
$$\chi^2_k \sim 2 \cdot (\log(L_{\text{RE}}) - \log(L_{\text{OLS}}))$$

```
teststat <- 2 * (logLik(randomML) - logLik(ols))[1]
```

```
   ML
673.7
```

This test statistic should be approximately χ^2 distributed as long as we are not at the boundary of the parameter space. When we test $\sigma^2_\nu = 0$ this is no longer the case. Nevertheless…

```
par(mar = c(4, 4, 0, 0))
plot(function(x) dchisq(x, 1), 0, 2, ylab = expression(chi[1]^2))
```



The χ^2 p-value would be

```
pchisq(teststat, 1, lower = FALSE)
```

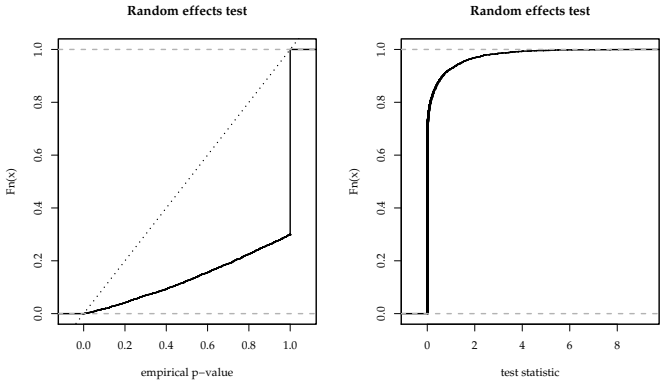
```
       ML
1.583e-148
```

Let us bootstrap the distribution:

```
set.seed(125)
dev <- replicate(5000, {
    by <- c(unlist(simulate(ols)))
    bols <- lm(by ~ x, data = data)
    brandom <- refit(randomML, by)
    LL <- 2 * (logLik(brandom) - logLik(bols))[1]
    c(LL = LL, pchisq = pchisq(LL, 1, lower = FALSE))
})
```

The bootstrapped distribution differs from the χ^2 distribution:

```
par(mfrow = c(1, 2))
plot(ecdf(dev["pchisq.ML", ]), do.points = FALSE, verticals = TRUE,
    xlab = "empirical p-value", main = "Random effects test")
abline(a = 0, b = 1, lty = 3)
plot(ecdf(dev["LL.ML", ]), do.points = FALSE, verticals = TRUE,
    xlab = "test statistic", main = "Random effects test")
```



We see that the assumption of a χ^2 distribution is rather conservative. If we manage to reject our Null (that there is no random effect) based on the χ^2 distribution, then we can definitely reject it based on the bootstrapped distribution. We might actually accept the Null too often. Hence, if we find a teststatistic which is still acceptable according to the χ^2 distribution (pooled OLS is ok), chances are that we could reject this statistic with the bootstrapped distribution.

We can, of course, use the bootstrapped value of the teststatistic and compare it with the value from our test:

```
mean(teststat < dev["LL.ML", ])
```

```
[1] 0
```

Note that we need many bootstrap replications to get reliable estimates for p-values.

2.7.1 Confidence intervals for fixed effects

We can't assume that estimated coefficients follow a normal distribution. To determine confidence intervals we have to bootstrap a sample of coefficients.

```
random.mc <- mcmcsamp(random, bootstrapsize)
HPDinterval(random.mc)$fixef
```

```
             lower  upper
(Intercept) -8.0498 0.5548
x           -0.2494 2.1356
attr(,"Probability")
[1] 0.95
```

Exercise 2.5 *The file* `ex2.csv` *contains observations on* `x1`*,* `x2`*,* `y` *and a group variable* `group`*. You are interested in how* `x1` *and* `x2` *influence* `y`*.*

- *In the fixed effects model: Is the group specific effect significant?*
- *In the random effects model: Is the group specific effect significant?*
- *Use a Hausman test to compare the fixed and the random effects model.*

3 A mixed effects model with unreplicated design

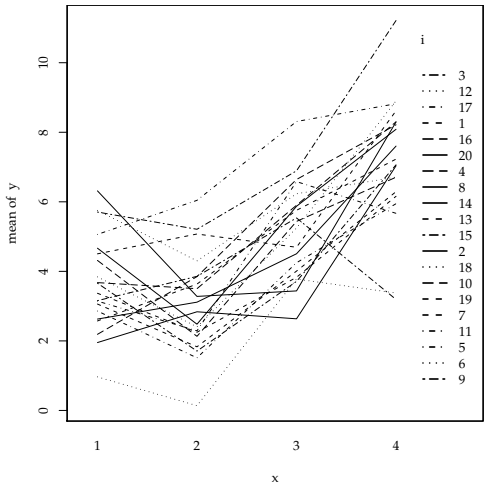
The dataset `dataM` shows the result of a (hypothetical) experiment where 20 different individuals i all solve 4 dif-

ferent tasks x. The dependent variable y shows the time needed by individual i for task x.

```
with(dataM, table(x, i))
```

```
   i
x   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
  1 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
  2 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
  3 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
  4 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
```

```
par(mar = c(4, 4, 0, 0))
with(dataM, interaction.plot(x, i, y))
```



One way to write the model:

$$y_{ij} = \beta_j + \nu_i + \epsilon_{ij}, \qquad i \in \{1,\dots,20\}, \quad j \in \{1,\dots 4\}$$

with $\nu_i \sim N(0,\sigma_\nu)$ and $\epsilon_{ij} \sim N(0,\sigma)$
An alternative way to write the model:

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\nu_i + \boldsymbol{\epsilon}_i, \qquad i \in \{1,\dots,20\}$$

with

$$\boldsymbol{y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ y_{i3} \\ y_{i4} \end{pmatrix}, \boldsymbol{X}_i = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{cell means}},$$

$$\boldsymbol{Z}_i = \boldsymbol{I} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \boldsymbol{\epsilon}_i = \begin{pmatrix} \epsilon_{i1} \\ \epsilon_{i2} \\ \epsilon_{i3} \\ \epsilon_{i4} \end{pmatrix}$$

Instead of using this specification, we could also use any other matrix of full rank. Common are the following:

$$\boldsymbol{X}_i = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}}_{\text{reference}}, \underbrace{\begin{pmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & 2 & -1 \\ 1 & 0 & 0 & 3 \end{pmatrix}}_{\text{Helmert}}$$

$$, \text{ or } \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix}}_{\text{sum}}$$

3.1 Estimation with different contrast matrices

3.1.1 First category as a reference

The default in R (and in Stata) is to use the first category as a reference.

```
data1 <- subset(dataM, i == 1)
mm <- model.matrix(y ~ x, data1)
```

```
as.data.frame(mm)
```

```
  (Intercept) x2 x3 x4
1           1  0  0  0
2           1  1  0  0
3           1  0  1  0
4           1  0  0  1
```

- Asymmetric treatment of categories (The effect of the first category is captured by the intercept. The effects of the remaining three treatments are relative to the intercept).
- x_1, x_2, and x_3 are not orthogonal to the intercept. Multiplied with the intercept the result is always different from zero.

```
as.data.frame(mm)
```

```
  (Intercept) x2 x3 x4
1           1  0  0  0
2           1  1  0  0
3           1  0  1  0
4           1  0  0  1
```

Let us check non-orthogonality:

```
c(mm[, 1] %*% mm[, 2:4], mm[, 2] %*% mm[, 3:4], mm[,
    3] %*% mm[, 4])
```

```
[1] 1 1 1 0 0 0
```

Here are the estimation results if we follow this approach:

```
r.lmer <- lmer(y ~ x + (1 | i), data = dataM)
print(r.lmer, correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: y ~ x + (1 | i)
   Data: dataM
 AIC   BIC logLik deviance REMLdev
 282 296.3   -135    267.5     270
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.1599   1.0770
 Residual             1.1697   1.0815
Number of obs: 80, groups: i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)   3.6697     0.3413  10.753
x2           -0.5980     0.3420  -1.748
x3            1.4924     0.3420   4.364
x4            3.5020     0.3420  10.240
```

```
as.data.frame(mm)
```

```
  (Intercept) x2 x3 x4
1           1  0  0  0
2           1  1  0  0
3           1  0  1  0
4           1  0  0  1
```

Linear combinations of the coefficients have a meaning:

If we are, e.g. interested in the mean of the second category, we add the intercept and the estimate of β_2:

```
r.lmer@fixef %*% mm[2, ]
```

```
        [,1]
[1,] 3.071748
```

3.1.2 Sum contrasts

Often it is interesting to immediately estimate an overall mean effect and then add contrasts that describe difference between treatments. Sum contrasts are one way to do this:

```
oldOpt <- getOption("contrasts")
options(contrasts = c(unordered = "contr.sum", ordered = "contr.poly"))
mm <- model.matrix(y ~ x, data1)
```

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1  1  0  0
2           1  0  1  0
3           1  0  0  1
4           1 -1 -1 -1
```

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1  1  0  0
2           1  0  1  0
3           1  0  0  1
4           1 -1 -1 -1
```

- Intercept: mean effect over all four treatments.
- Coefficient of *x1*: difference between the first and the fourth treatment.
- Coefficient of *x2*: difference between the second and the fourth treatment.
- Coefficient of *x3*: difference between the third and the fourth treatment.

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1  1  0  0
2           1  0  1  0
3           1  0  0  1
4           1 -1 -1 -1
```

Still, coefficients are not orthogonal.

```
c(mm[, 1] %*% mm[, 2:4], mm[, 2] %*% mm[, 3:4], mm[,
    3] %*% mm[, 4])
```

```
[1] 0 0 0 1 1 1
```

Here are the estimation results if we follow this approach:

```
s.lmer <- lmer(y ~ x + (1 | i), data = dataM)
print(s.lmer, correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: y ~ x + (1 | i)
   Data: dataM
   AIC   BIC logLik deviance REMLdev
 284.8 299.1 -136.4    267.5   272.8
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.1599   1.0770
 Residual             1.1697   1.0815
Number of obs: 80, groups: i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)   4.7689     0.2695  17.698
x1           -1.0991     0.2094  -5.248
x2           -1.6971     0.2094  -8.103
x3            0.3933     0.2094   1.878
```

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1  1  0  0
2           1  0  1  0
3           1  0  0  1
4           1 -1 -1 -1
```

Linear combinations of the coefficients have a meaning:

If we are, e.g. interested in the mean of the second category, we add the intercept and the estimate of β_2:

```
s.lmer@fixef %*% mm[2, ]
```

```
        [,1]
[1,] 3.071748
```

3.1.3 Helmert contrasts

Helmert contrasts are another way to do this.

```
options(contrasts = c(unordered = "contr.helmert", ordered = "contr.poly"))
mm <- model.matrix(y ~ x, data1)
```

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1 -1 -1 -1
2           1  1 -1 -1
3           1  0  2 -1
4           1  0  0  3
```

- Intercept: mean effect over all four treatments.
- Coefficient of *x1*: difference between the second and the first treatment.
- Coefficient of *x2*: difference between the third and the mean of the first two.
- Coefficient of *x3*: difference between the fourth and the mean of the other three.

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1 -1 -1 -1
2           1  1 -1 -1
3           1  0  2 -1
4           1  0  0  3
```

Furthermore, all variables are now uncorrelated.

```
c(mm[, 1] %*% mm[, 2:4], mm[, 2] %*% mm[, 3:4], mm[,
    3] %*% mm[, 4])
```

```
[1] 0 0 0 0 0 0
```

Here are the estimation results based on Helmert contrasts.

```
h.lmer <- lmer(y ~ x + (1 | i), data = dataM)
print(h.lmer, correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: y ~ x + (1 | i)
   Data: dataM
   AIC   BIC logLik deviance REMLdev
 288.4 302.7 -138.2    267.5   276.4
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.1599   1.0770
 Residual             1.1697   1.0815
Number of obs: 80, groups: i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.76886    0.26946  17.698
x1          -0.29900    0.17100  -1.748
x2           0.59715    0.09873   6.048
x3           0.80097    0.06981  11.473
```

```
options(contrasts = oldOpt)
```

```
as.data.frame(mm)
```

```
  (Intercept) x1 x2 x3
1           1 -1 -1 -1
2           1  1 -1 -1
3           1  0  2 -1
4           1  0  0  3
```

It is still possible to calculate the mean effect of, e.g. the second treatment:

```
h.lmer@fixef %*% mm[2, ]
```

```
        [,1]
[1,] 3.071748
```

3.1.4 Cell means contrasts

If we are not primarily interested in the overall mean effect, then cell means are a possibility:

```
mm <- model.matrix(y ~ x - 1, data1)
```

```
as.data.frame(mm)
```

```
  x1 x2 x3 x4
1  1  0  0  0
2  0  1  0  0
3  0  0  1  0
4  0  0  0  1
```

Now the four coefficients reflect the average effect of the four categories.

Here is the estimation result for cell means:

```
cm.lmer <- lmer(y ~ x - 1 + (1 | i), data = dataM)
print(cm.lmer, correlation = FALSE)
```

```
Linear mixed model fit by REML
Formula: y ~ x - 1 + (1 | i)
   Data: dataM
 AIC   BIC logLik deviance REMLdev
 282 296.3   -135    267.5     270
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.1599   1.0770
 Residual             1.1697   1.0815
Number of obs: 80, groups: i, 20

Fixed effects:
   Estimate Std. Error t value
x1   3.6697     0.3413  10.753
x2   3.0717     0.3413   9.001
x3   5.1622     0.3413  15.126
x4   7.1718     0.3413  21.014
```

```
as.data.frame(mm)
```

```
  x1 x2 x3 x4
1  1  0  0  0
2  0  1  0  0
3  0  0  1  0
4  0  0  0  1
```

It is still possible to calculate the mean effect of, e.g. the second treatment:

```
cm.lmer@fixef %*% mm[2, ]
```

```
        [,1]
[1,] 3.071748
```

3.2 Which statistics are affected by the type of contrasts?

3.2.1 t-statistics and p-values

As we see above, t-statistics (and, hence, p-values) depend very much on the way how the fixed effect enters the model. We should not use these statistics when we assess the influence of the factor.

```
models <- list(reference = r.lmer, sum = s.lmer, helmert = h.lmer,
    cellmeans = cm.lmer)
sapply(models, function(model) summary(model)@coefs[,
    "t value"])
```

```
            reference       sum   helmert cellmeans
(Intercept) 10.752786 17.697531 17.697537 10.752786
x2          -1.748497 -5.248044 -1.748497  9.000591
x3           4.363820 -8.103327  6.048400 15.125835
x4          10.239706  1.878044 11.473327 21.014148
```

3.2.2 Anova

As long as we keep the intercept, the anova is not affected. We should use the anova (with an intercept term) when we assess the influce of the factor.

```
sapply(models, function(model) anova(model))
```

```
        reference sum      helmert  cellmeans
Df      3         3        3        4
Sum Sq  200.3386  200.3386 200.3386 566.5205
Mean Sq 66.77953  66.77953 66.77953 141.6301
F value 57.09254  57.09254 57.09254 121.0854
```

The last representation (`cellmeans`) leads to a different anova. The reason is that the latter model is tested against $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ while the other two are only tested against $\beta_1 = \beta_2 = \beta_3 = \beta_4 =$ constant.

3.2.3 Information criteria

The change in the type of contrasts is a change in the fixed effect, hence (with REML) changes the likelihood of the model and, thus, also the AIC and BIC.

```
sapply(models, function(model) summary(model)@AICtab)
```

```
         reference sum       helmert   cellmeans
AIC      282.0249  284.7975  288.3811  282.0249
BIC      296.3171  299.0897  302.6732  296.3171
logLik   -135.0125 -136.3988 -138.1905 -135.0125
deviance 267.5197  267.5197  267.5197  267.5197
REMLdev  270.0249  272.7975  276.3811  270.0249
```

When we compare information criteria of different models, we have to take the same type of contrasts — at least as long as we use REML estimation.

With ML estimation the type of the contrasts does not matter for information criteria:

```
sapply(models, function(model) summary(update(model,
    REML = FALSE))@AICtab)
```

```
         reference sum       helmert   cellmeans
AIC      279.5197  279.5197  279.5197  279.5197
BIC      293.8119  293.8119  293.8119  293.8119
logLik   -133.7599 -133.7599 -133.7599 -133.7599
deviance 267.5197  267.5197  267.5197  267.5197
REMLdev  270.0249  270.0249  270.0249  270.0249
```

Likelihood ratio tests should, hence, be carried out with ML, not with REML.

Exercise 3.1 *The dataset* `ex3.csv` *contains three variables. g controls for the treatment group, x is an independent variable, and y is the dependent variable. You want to estimate*

$$y = \beta x + \sum_{g=1}^{G} d_g \gamma_g + u$$

where d_g is a dummy that is one for observations in group g and zero otherwise.

1. *Compare a simple OLS, a fixed effects, and a random effects model.*
2. *You are not primarily interested in the individual values of γ_g but you want to estimate the average value of γ_g. What is a simple way to obtain this in a fixed effects model?*
3. *How can you do this in a random effects model?*
4. *Compare the fixed effects with the random effects model with a Hausman test.*
5. *Now you suspect the following relationship:*

$$y = \gamma + \sum_{i=0}^{G} d_g \beta_g x + u\,.$$

Again, you are not interested in the individual values of β_g but you want to estimate an average effect. Compare the results of a fixed and random effects model.

4 Testing fixed effects

To test a fixed effect we can not use REML as an estimation procedure.

4.1 Anova

```
r.lmerML <- update(r.lmer, REML = FALSE)
r.lmerMLsmall <- update(r.lmerML, . ~ . - x)
r.anova <- anova(r.lmerMLsmall, r.lmerML)
r.anova
```

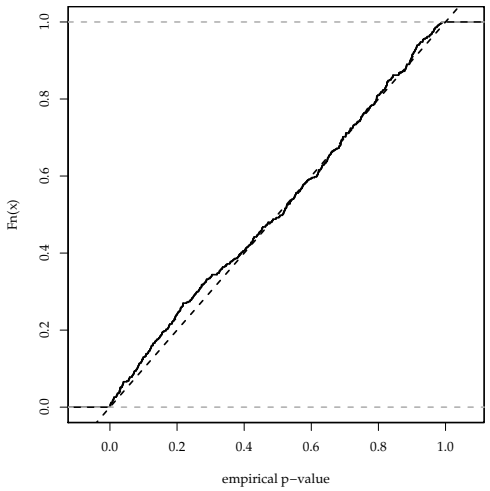
```
Data: dataM
```

```
Models:
r.lmerMLsmall: y ~ (1 | i)
r.lmerML: y ~ x + (1 | i)
             Df    AIC    BIC  logLik Chisq Chi Df Pr(>Chisq)
r.lmerMLsmall  3 357.13 364.28 -175.56
r.lmerML       6 279.52 293.81 -133.76 83.61      3  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let us check whether the assumption of a χ^2 distributed test statistic, which is made by `anova`, is really justified.

```
set.seed(123)
empirP <- replicate(500, {
    r.lmerSim <- lmer(y ~ sample(x) + (1 | i), data = dataM,
        REML = FALSE)
    a <- anova(r.lmerMLsmall, r.lmerSim)
    c(Chisq = a[["Chisq"]][2], df = a[["Chi Df"]][2],
        pval = a[["Pr(>Chisq)"]][2])
})
```

```
par(mar = c(4, 4, 0, 0))
plot(ecdf(empirP["pval", ]), do.points = FALSE, verticals = TRUE,
    xlab = "empirical p-value", main = "")
abline(a = 0, b = 1, lty = 2)
```



The empirical frequency to get the χ^2 statistic we got above under the Null is

```
mean(r.anova[["Chisq"]][2] < empirP["Chisq", ])
```

```
[1] 0
```

So far everything looks good. For the dataset PBIB[1] (provided by the library(SASmixed)) things do not work out so well.

```
library(SASmixed)
data(PBIB)
```

Here is the `anova` for PBIB:

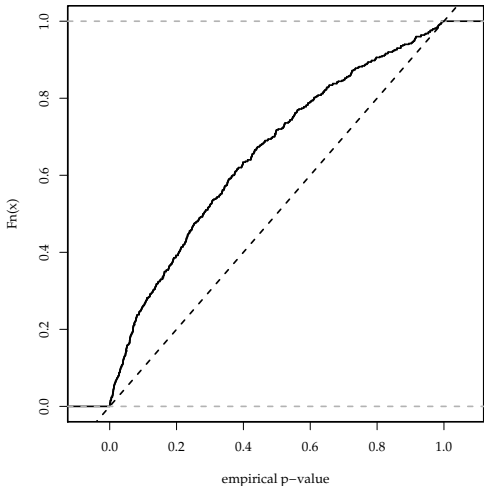
```
l.small <- lmer(response ~ 1 + (1 | Block), data = PBIB,
    REML = FALSE)
l.large <- lmer(response ~ sample(Treatment) + (1 | Block),
    data = PBIB, REML = FALSE)
pbib.anova <- anova(l.large, l.small)
pbib.anova
```

```
Data: PBIB
Models:
l.small: response ~ 1 + (1 | Block)
l.large: response ~ sample(Treatment) + (1 | Block)
        Df    AIC    BIC   logLik  Chisq Chi Df Pr(>Chisq)
l.small  3 52.152 58.435 -23.0759
l.large 17 52.082 87.686  -9.0412 28.070     14    0.01393 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now we bootstrap the distribution of the empirical p-values, provided that `Treatment` is entirely random:

```
empirP <- replicate(500, {
    l.largeSim <- lmer(response ~ sample(Treatment) +
        (1 | Block), data = PBIB, REML = FALSE)
    a <- anova(l.small, l.largeSim)
    c(Chisq = a[["Chisq"]][2], df = a[["Chi Df"]][2],
        pval = a[["Pr(>Chisq)"]][2])
})
```

```
par(mar = c(4, 4, 0, 0))
plot(ecdf(empirP["pval", ]), do.points = FALSE, verticals = TRUE,
    xlab = "empirical p-value", main = "")
abline(a = 0, b = 1, lty = 2)
```



The empirical frequency to get the χ^2 statistic we got above under the Null is

```
mean(pbib.anova[["Chisq"]][2] < empirP["Chisq", ])
```

```
[1] 0.056
```

With the help of `anova`, how often would we obtain an empirical p-value smaller 5%, if the variable `Treatment` does not matter at all?

```
mean(empirP["pval", ] < 0.05) * 100
```

```
[1] 14.8
```

[1] Littel, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), SAS System for Mixed Models, SAS Institute (Data Set 1.5.1)

4.2 Confidence intervals

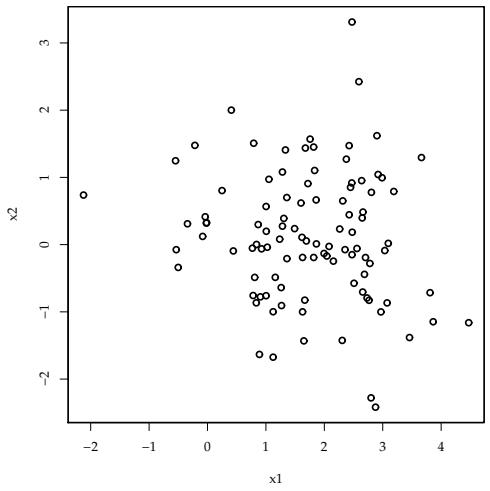
Let us have a look at the dataset *data3*. It is similar to *data*, except that now we have two fixed effects, x_1 and x_2.

```
random <- lmer(y ~ x1 + x2 + (1 | i), data = data3)
```

mcmcsamp generates a sample from the posterior distribution of parameters.

```
random.boot <- mcmcsamp(random, bootstrapsize)
```

```
par(mar = c(4, 4, 0, 0))
plot(t(random.boot@fixef[2:3, ]))
```



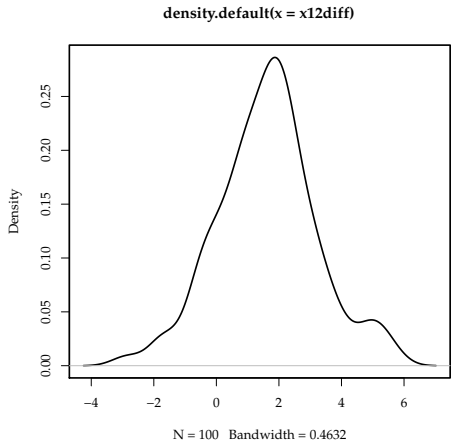
HPDinterval generates confidence intervals.

```
HPDinterval(random.boot)$fixef
```

```
                lower    upper
(Intercept) -2.1572418 3.995950
x1          -0.5473138 3.663168
x2          -1.6742225 2.000468
attr(,"Probability")
[1] 0.95
```

Functions of coefficients We can use the mcmc-Sample to look at linear functions of coefficients. Assume that in the above example we are interested in $\beta_{x_1} - \beta_{x_2}$.

```
x12diff <- with(as.data.frame(t(random.boot@fixef)),
    x1 - x2)
plot(density(x12diff))
```



Again, we can use *HPDinterval* to calculate confidence intervals.

```
HPDinterval(as.matrix(x12diff))
```

```
         lower    upper
[1,] -0.8372966 5.633797
attr(,"Probability")
[1] 0.95
```

4.3 Testing random effects

See section 2.7 above.

Exercise 4.1 *You look again at the* `ex3.csv` *(see exercise 3.1) and at the following model*

$$y = \beta x + \sum_{g=1}^{G} d_g \gamma_g + u$$

where d_g is a dummy that is one for observations in group g and zero otherwise.

1. *In a model with fixed effects for g: Does one have to include the fixed effect? Give a confidence interval for the average value (over groups g) of γ_g.*
2. *In a model with random effects for g: Does one have to include the random effect? Give a confidence interval for the average value (over groups g) of γ_g.*
3. *Now do the same for the following model:*

$$y = \gamma + \sum_{i=0}^{G} d_g \beta_g x + u\,.$$

5 Mixing fixed and random effects

A common situation in economic experiments is that different groups of participants are associated with different treaments. To measure the size of the treatment effect we want to introduce a fixed effect for the treatment. Can we also introduce a random effect for the groups (which are nested in the treatments)?

The dataset *ex4.csv* contains observations on a hypothetical experiment with 3 treatments and 108 participants in 9 groups. Each group contains 12 participants. Each participants stays for 10 periods in the experiment. Each group participates in only one treatment. Since participants within a group interact over these 10 periods we suspect that observations within a group are correlated.

```
ex4 <- read.csv("ex4.csv")
ex4[1:20, ]
```

```
   treat group pid period    y
1      A     1   1      1 11.2
2      A     1   1      2 11.3
3      A     1   1      3 11.2
4      A     1   1      4 11.1
5      A     1   1      5 11.4
6      A     1   1      6 10.5
7      A     1   1      7 11.0
8      A     1   1      8 10.3
9      A     1   1      9 10.4
10     A     1   1     10 10.1
11     A     1   2      1 12.7
12     A     1   2      2 12.9
13     A     1   2      3 12.0
14     A     1   2      4 12.6
15     A     1   2      5 12.2
16     A     1   2      6 12.2
17     A     1   2      7 12.2
18     A     1   2      8 11.4
19     A     1   2      9 12.2
20     A     1   2     10 12.0
```

Now let us estimate the treatment effect of `treat` and include a random effect for the `group` as well as a random effect for the participants `pid`.

```
r.mer <- lmer(y ~ treat - 1 + (1 | group) + (1 | pid),
    data = ex4)
fixef(r.mer)
```

```
   treatA    treatB    treatC
12.163889  4.458611  8.693056
```

R calculates two random effects, random effects for participants and for groups. Furthermore we have the estimated residuals.

```
str(ranef(r.mer))
```

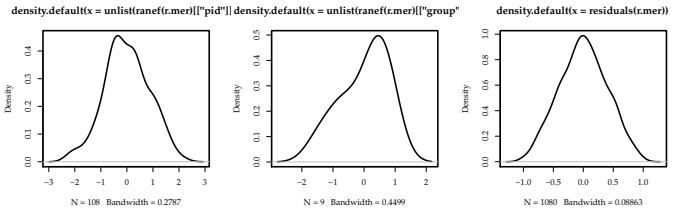
```
List of 2
 $ pid  :'data.frame':        108 obs. of  1 variable:
  ..$ (Intercept): num [1:108] -0.51 0.85 0.155 0.409 -1.215 ...
 $ group:'data.frame':        9 obs. of  1 variable:
  ..$ (Intercept): num [1:9] -0.7924 0.1581 0.6013 0.0543 0.4339 ...
 - attr(*, "class")= chr "ranef.mer"
```

```
str(residuals(r.mer))
```

```
num [1:1080] 0.339 0.439 0.339 0.239 0.539 ...
```

Here is the density of the estimated random effects and residuals:

```
par(mfrow = c(1, 3))
plot(density(unlist(ranef(r.mer)[["pid"]])))
plot(density(unlist(ranef(r.mer)[["group"]])))
plot(density(residuals(r.mer)))
```



Exercise 5.1 *Have another look at the dataset* `ex4.csv`*. You suspect that behaviour in the experiment changes over time (*`period`*).*

1. *Do you think that there is such an effect?*
2. *Is the effect linear?*
3. *Assume that the effect is linear, can you give a confidence interval for the size of the effect?*
4. *Is the magniture of the effect the same for all treatments?*

6 A mixed effects model with replicated design

The dataset *dataMR* shows the result of a (hypothetical) experiment where 20 different individuals i all solve 3 different tasks x. The dependent variable y shows the time needed by individual i for task x. In contrast to the experiment shown in *dataM* in this experiment participants took each task 4 times.

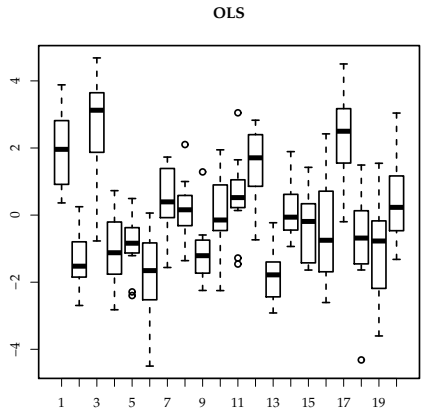
```
with(dataMR, table(x, i))
```

```
   i
x   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
  1 4 4 4 4 4 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4
  2 4 4 4 4 4 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4
  3 4 4 4 4 4 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4
```

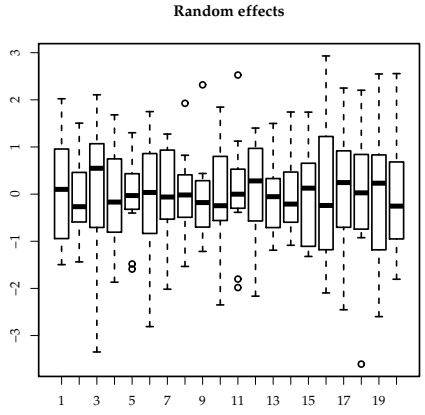
6.1 A model with one random effect

Let us compare residuals for each individual for an OLS with a random effects model:

```
ols <- lm(y ~ x, data = dataMR)
boxplot(residuals(ols) ~ i,
    data = dataMR, main = "OLS")
```




```
m1.lmer <- lmer(y ~ x + (1 |
    i), data = dataMR)
boxplot(residuals(m1.lmer) ~
    i, data = dataMR, main = "Random effects")
```



Visual comparison:

- heterogeneity among individuals

Alternative:

- Calculate the difference between the likelihoods of the two models and then bootstrap the distribution as we did above.

Here we look at another problem. So far we have a random effect for the intercept only.

$$y_{ij} = \beta_j + \nu_i + \epsilon_{ij}, \qquad i \in \{1,\dots,20\}, \quad j \in \{1,\dots 3\}$$

with $\nu_i \sim N(0,\sigma_\nu)$ and $\epsilon_{ij} \sim N(0,\sigma)$
The result was

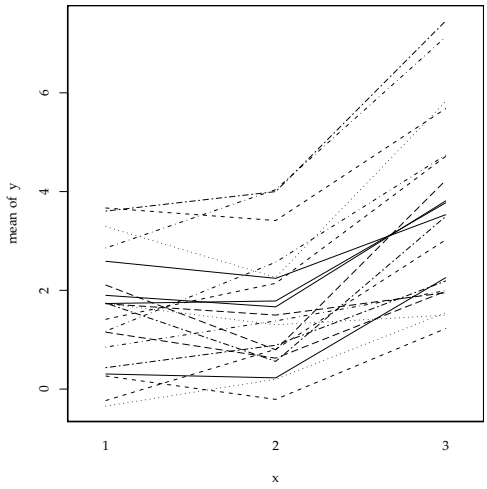
```
summary(m1.lmer)
```

```
Linear mixed model fit by REML
Formula: y ~ x + (1 | i)
   Data: dataMR
   AIC   BIC logLik deviance REMLdev
 819.5 836.9 -404.7    805.5   809.5
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.6796   1.2960
 Residual             1.3501   1.1620
Number of obs: 240, groups: i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.59859    0.31756   5.034
x2           0.01165    0.18372   0.063
x3           2.00393    0.18372  10.907

Correlation of Fixed Effects:
   (Intr) x2
x2 -0.289
x3 -0.289  0.500
```

6.2 Random effects for interactions

Is the above enough? Could it be that the effect of `x` itself varies with `i`, i.e. that we have to consider an interaction between `x` and `i` for the random effect?

```
par(mar = c(4, 4, 0, 0))
with(dataMR, interaction.plot(x, i, y, legend = FALSE))
```



The graph suggests that individuals `i` react differently to treatments `x`.

We estimate the following random effects model:

$$y_{ij} = \beta_j + \nu_i + \nu_{ij} + \epsilon_{ijk},$$
$$i \in \{1,\dots,20\}, \quad j \in \{1,\dots 3\}, \quad k \in \{1,\dots 4\}$$
with $\nu_i \sim N(0,\sigma_\nu)$, $\nu_{ij} \sim N(0,\sigma_{\nu'})$, and $\epsilon_{ij} \sim N(0,\sigma)$

```
m2.lmer <- lmer(y ~ x + (1 | i) + (1 | i:x), data = dataMR)
```

An equivalent (more compact) notation is the following:

```
m2.lmer <- lmer(y ~ x + (1 | i/x), data = dataMR)
m2.lmer
```

We could now use `anova` to compare the two models, although we can not be really sure whether the test statistics is really χ^2 distributed.

```
(anovaResult <- anova(m1.lmer, m2.lmer))
```

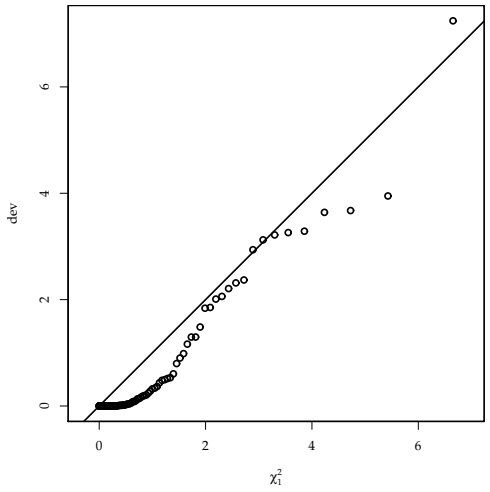
```
Data: dataMR
Models:
m1.lmer: y ~ x + (1 | i)
m2.lmer: y ~ x + (1 | i) + (1 | i:x)
        Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
m1.lmer  5 815.48 832.89 -402.74
m2.lmer  6 813.87 834.76 -400.94 3.6098      1    0.05744 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bootstrapping the distribution of the teststatistic we find that the approximation with the χ^2 distribution is not too bad. Again, a χ^2 test will usually be too conservative.

```
set.seed(125)
dev <- replicate(bootstrapsize, {
    by <- c(simulate(m1.lmer))
    b1 <- refit(m1.lmer, by)
    b2 <- refit(m2.lmer, by)
    2 * (logLik(b2) - logLik(b1))[1]
})
```

```
par(mar = c(4, 4, 0, 0))
qqplot(qchisq((1:bootstrapsize)/(bootstrapsize + 1),
    df = 1), dev, xlab = expression(chi[1]^2), asp = 1)
abline(a = 0, b = 1)
cat("p=", mean(anovaResult$Chisq[2] < dev))
```

```
p= 0.04
```



6.3 Interactions and replications

$$y_{ij} = \beta_j + \nu_i + \nu_{ij} + \epsilon_{ijk}, \quad i \in \{1,\dots,20\}, \quad j \in \{1,\dots 3\}, \quad k \in \{1,\dots 4\}$$

We need some replications k in order to distinguish between ν_{ij} and ϵ_{ijk}. The design need not be balanced, though.

6.4 More random interactions

$$y_{ij} = \beta_j + \nu_i + \nu_{ij} + \epsilon_{ijk}, \quad i \in \{1,\dots,20\}, \quad j \in \{1,\dots 3\}, \quad k \in \{1,\dots 4\}$$

In the above model we have made the following assumptions:

- All random interactions have the same variance $\sigma_{\nu'}$
- All random interaction terms are independent.

  This is a strong assumption. For any individual i it requires that the ν_{ij} are uncorrelated.

A more general model is the following:

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \qquad i \in \{1,\dots,20\}$$

with $\boldsymbol{b}_i \sim N(0, \boldsymbol{\Psi})$, $\boldsymbol{\epsilon} \sim N(0, \sigma^2\boldsymbol{I})$ and $\boldsymbol{\Psi}_{3\times 3}$ symmetric and positive definite.

Here is our $\boldsymbol{X}_i$ matrix, for a representative individual `i=1`:

```
dataMR1 <- subset(dataMR, i == 1)
as.data.frame(model.matrix(y ~ x, data = dataMR1))
```

```
   (Intercept) x2 x3
1            1  0  0
2            1  0  0
3            1  0  0
4            1  0  0
5            1  1  0
6            1  1  0
7            1  1  0
8            1  1  0
9            1  0  1
10           1  0  1
11           1  0  1
12           1  0  1
```

Note that random effects should have a mean of 0, anyway. Hence, there is no need to use contrast matrices which show averages of random effects. The simplest is the cell means specification for $\boldsymbol{Z}_i$.

```
as.data.frame(model.matrix(~x - 1, data = dataMR1))
```

```
   x1 x2 x3
1   1  0  0
2   1  0  0
3   1  0  0
4   1  0  0
5   0  1  0
6   0  1  0
7   0  1  0
8   0  1  0
9   0  0  1
10  0  0  1
11  0  0  1
12  0  0  1
```

```
(m3.lmer <- lmer(y ~ x + (x - 1 | i), data = dataMR))
```

```
Linear mixed model fit by REML
Formula: y ~ x + (x - 1 | i)
   Data: dataMR
   AIC   BIC logLik deviance REMLdev
 809.6 844.4 -394.8    785.9   789.6
Random effects:
 Groups   Name Variance Std.Dev. Corr
 i        x1   1.1354   1.0656
          x2   1.2365   1.1120   0.966
          x3   3.1816   1.7837   0.915 0.988
 Residual      1.1851   1.0886
Number of obs: 240, groups: i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.59859    0.26756   5.975
x2           0.01165    0.18380   0.063
x3           2.00393    0.26740   7.494

Correlation of Fixed Effects:
   (Intr) x2
x2 -0.292
x3  0.215  0.517
```

We see that the estimated standard deviations of the random effects differ among treatments and are highly correlated. We can compare the three models with the help of an `anova`.

```
anova(m1.lmer, m2.lmer, m3.lmer)
```

```
Data: dataMR
Models:
m1.lmer: y ~ x + (1 | i)
m2.lmer: y ~ x + (1 | i) + (1 | i:x)
m3.lmer: y ~ x + (x - 1 | i)
       Df    AIC    BIC  logLik   Chisq Chi Df Pr(>Chisq)
m1.lmer  5 815.48 832.89 -402.74
m2.lmer  6 813.87 834.76 -400.94  3.6098      1   0.057440 .
m3.lmer 10 805.85 840.66 -392.93 16.0225      4   0.002989 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The improvement in the log-likelihood is significant. Also the AIC would suggest that introdicing more parameters (as Ψ) is worth the effort. The BIC puts a higher penality on the additional parameters and would, hence, prefer the first model.

Exercise 6.1 *The dataset* `ex5.csv` *contains 8 variables. The treatment is coded as* `treatment`*, the id of the participant is stored in* `participant`*. Participants have different* `height`*,* `profession`*, and* `gender`*. Further controls are* `x1` *and* `x2`*. You are interested in the effect of* `treatment`*. Compare the following:*

1. *A (pooled) OLS model where you do not control for heterogeneity of participants (but you control for gender, height and profession),*
2. *a fixed effects model where you include a fixed effect for each participant,*
3. *a mixed model with a random effect for participants.*

What is the expected treatment effect from `B` *to* `C` *for a female, white collar worker of medium height? Can you give a confidence interval?*

7 Random effects for more than a constant

7.1 Models we studied so far

$$\begin{aligned} y_{ij} &= \beta_j + \nu_i + \epsilon_{ij}, \quad i \in \{1,\dots,20\}, j \in \{1,\dots 4\} \\ y_{ij} &= \beta_j + \nu_i + \nu_{ij} + \epsilon_{ijk}, \quad i \in \{1,\dots,20\}, \\ &\qquad j \in \{1,\dots 4\}, k \in \{1,\dots 3\} \\ \boldsymbol{y}_i &= \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i \in \{1,\dots,20\} \end{aligned}$$

In the previous examples, $\boldsymbol{X}$ and $\boldsymbol{Z}$ contained only treatment effects.

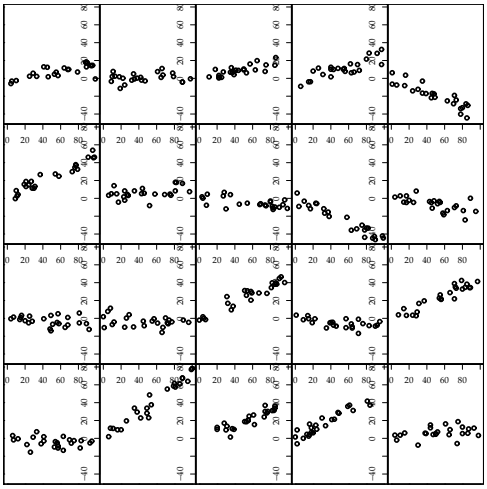
What, if $\boldsymbol{X}$ and $\boldsymbol{Z}$ also contain a linear variable, like a valuation, a cost, or time expired during the experiment?

Let us, in a first step, add a linear factor x_i to this model.

$$\begin{aligned} y_i &= \beta_1 + \beta_2 x_i + \epsilon_i, \quad i \in \{1,\dots,20\}, \epsilon_i \sim N(0,\sigma^2) \\ y_{ij} &= \beta_1 + \beta_2 x_i + \nu_i + \epsilon_{ij}, \\ &\qquad i \in \{1,\dots,20\}, j \in \{1,\dots 4\} \\ &\qquad \nu_i \sim N(0,\sigma_\nu^2), \epsilon_{ij} \sim N(0,\sigma^2) \end{aligned}$$

The dataset `dataII` contains information about 20 individuals which were divided into two groups. One of the two groups got treatment *a* (shown as a $+$ in the graph), the other not (shown as a $\circ$).

```
par(mfrow = c(4, 5), mar = c(0, 0, 0, 0))
qq <- with(dataII, sapply(unique(i), function(j) {
    plot(y ~ x, ylim = range(y), xlim = range(x), subset = i ==
        j)
}))
```

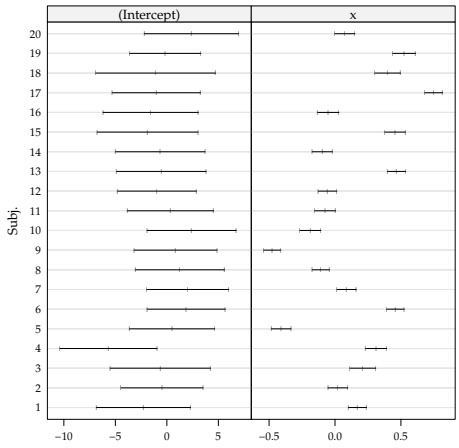


The figure suggests some systematic differences among participants i. Let us first estimate one OLS model for each participant.

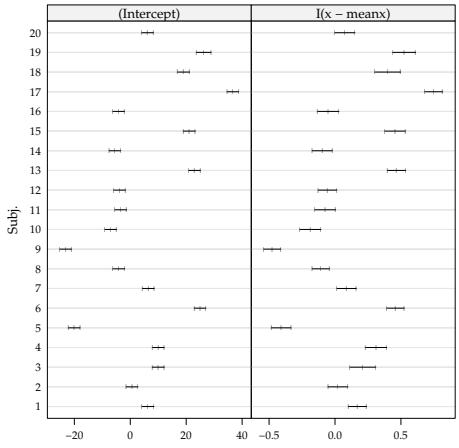
```
ols.list <- lmList(y ~ x | i, data = dataII)
aM <- with(dataII, aggregate(a, list(i), median))[, 2]
```

The following two graphs shows estimated confidence intervals for the coefficients. The left shows intervals for the above regression, the right shows intervals for a regression where x enteres "demeaned".

```
library(nlme)
iL <- intervals(ols.list)
attr(iL, "groupsName") <- "Subj."
print(plot(iL))
```




```
meanx <- mean(dataII$x)
ols2.list <- lmList(y ~ I(x -
    meanx) | i, data = dataII)
iL <- intervals(ols2.list)
attr(iL, "groupsName") <- "Subj."
print(plot(iL))
```



We see that scaling of the independent variable x has a considerable impact on the intervals for the intercept. Confidence intervals for x (or the demeaned version of x are not affected.

We do this exercise here to show that scaling the independent variables is not innocent. Here we will continue without scaling.

The above figure already suggests some randomness in the coefficient of x.

We compare the following models:

$$\begin{aligned} y_{ij} &= \beta_1 + \beta_2 x_i + \nu_i + \epsilon_{ij} \\ &\quad \nu_i \sim N(0, \sigma_\nu^2), \epsilon_{ij} \sim N(0, \sigma^2) \\ y_{ij} &= \beta_1 + (\beta_2 + \nu_i')x_i + \nu_i + \epsilon_{ij} \\ &\quad \nu_i' \sim N(0, \sigma_{\nu'}^2), \nu_i \sim N(0, \sigma_\nu^2), \epsilon_{ij} \sim N(0, \sigma^2) \end{aligned}$$

Let us start with the first model:

```
(r1.lmer <- lmer(y ~ x * a + (1 | i), data = dataII))
```

```
Linear mixed model fit by REML
Formula: y ~ x * a + (1 | i)
   Data: dataII
  AIC  BIC logLik deviance REMLdev
 3653 3678  -1820     3639    3641
Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 221.760  14.8916
 Residual              97.974   9.8982
Number of obs: 480, groups: i, 20

Fixed effects:
             Estimate Std. Error t value
(Intercept)  0.578952   4.887328   0.118
x           -0.005281   0.022644  -0.233
aTRUE       -1.107130   6.920993  -0.160
x:aTRUE      0.250553   0.032874   7.622

Correlation of Fixed Effects:
        (Intr) x      aTRUE
x       -0.234
aTRUE   -0.706  0.165
x:aTRUE  0.161 -0.689 -0.239
```

Now we estimate the second, a "multilevel" mixed effects model (with a random effect on the intercept but also on x).

```
(r2.lmer <- lmer(y ~ x * a + (x + 1 | i), data = dataII))
```

```
Linear mixed model fit by REML
Formula: y ~ x * a + (x + 1 | i)
   Data: dataII
  AIC  BIC logLik deviance REMLdev
 3055 3088  -1519     3035    3039
Random effects:
 Groups   Name        Variance   Std.Dev.     Corr
 i        (Intercept) 7.4473e-11 0.0000086298
          x           8.9713e-02 0.2995220532 0.000
 Residual             2.6063e+01 5.1051589141
Number of obs: 480, groups: i, 20

Fixed effects:
             Estimate Std. Error t value
(Intercept)  0.217634   0.677379   0.321
x            0.005387   0.095453   0.056
aTRUE       -0.674196   0.967777  -0.697
x:aTRUE      0.233632   0.135033   1.730

Correlation of Fixed Effects:
```

```
        (Intr) x      aTRUE
x       -0.108
aTRUE   -0.700  0.076
x:aTRUE  0.076 -0.707 -0.110
```

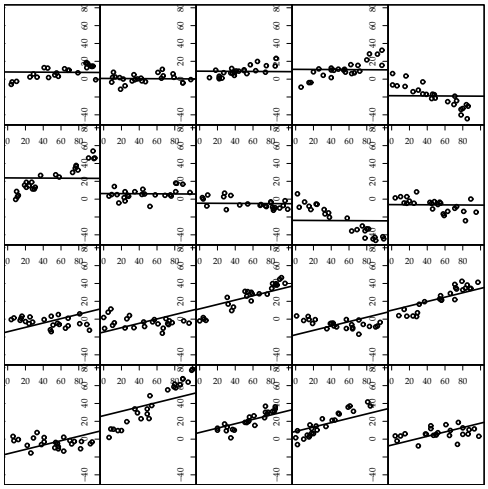
For the second model, let us calculate the slopes and intercepts of the best predictor for each individual:

```
aa1 <- r1.lmer@fixef["(Intercept)"] + aM * r1.lmer@fixef["aTRUE"] +
    r1.lmer@ranef
bb1 <- r1.lmer@fixef["x"] + aM * r1.lmer@fixef["x:aTRUE"]
aa2 <- r2.lmer@fixef["(Intercept)"] + aM * r2.lmer@fixef["aTRUE"] +
    r2.lmer@ranef[1:20]
bb2 <- r2.lmer@fixef["x"] + aM * r2.lmer@fixef["x:aTRUE"] +
    r2.lmer@ranef[21:40]
myPlot <- function(aa, bb) {
    par(mfrow = c(4, 5), mar = c(0, 0, 0, 0))
    qq <- with(dataII, sapply(unique(i), function(j) {
        plot(y ~ x, ylim = range(y), xlim = range(x),
            subset = i == j)
        abline(a = aa[j], b = bb[j])
    }))
}
```

The following graph shows the predicted values for each individual.

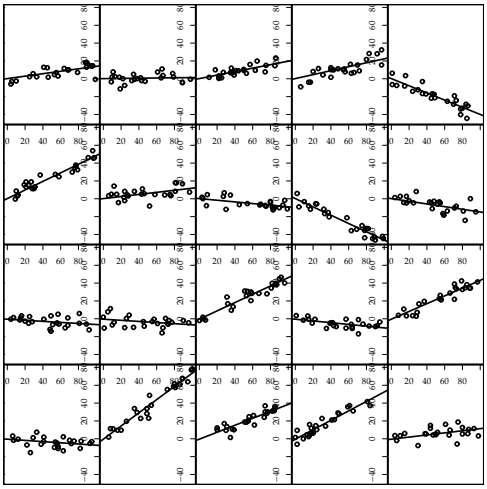
random effect only for intercept:

```
myPlot(aa1, bb1)
```



random effect for intercept and x:

```
myPlot(aa2, bb2)
```



Visual inspection suggests that the second model, which includes a random effect for x in addition to the random effect for the intercept, is more appropriate. More formally, we compare the two models with `anova`.

```
anova(r1.lmer, r2.lmer)
```

```
Data: dataII
Models:
r1.lmer: y ~ x * a + (1 | i)
r2.lmer: y ~ x * a + (x + 1 | i)
        Df    AIC    BIC  logLik Chisq Chi Df Pr(>Chisq)
r1.lmer  6 3651.1 3676.1 -1819.5
r2.lmer  8 3051.2 3084.6 -1517.6 603.9      2  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 7.1 *Have another look at the dataset* `ex5.csv`*. Now you suspect that the effect of* `x1` *might depend on the participant. Compare the following three models:*

- *a model where* `participant` *only affects the intercept,*
- *a model where* `participant` *only affects the slope of* `x1`*,*
- *a model where* `participant` *affects the slope of* `x1` *and the intercept.*

Which of these models do you prefer? Test formally!

8 Nonlinear models

As in section 2.2 we create an example dataset. The difference is that y is now a binary variable.

```
data <- read.csv("ex8.csv")
```

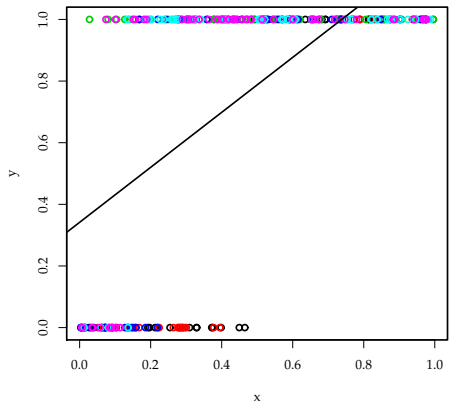
$$Y = 1 \Leftrightarrow x + \nu_i + \epsilon_{ik} > \texttt{crit}$$

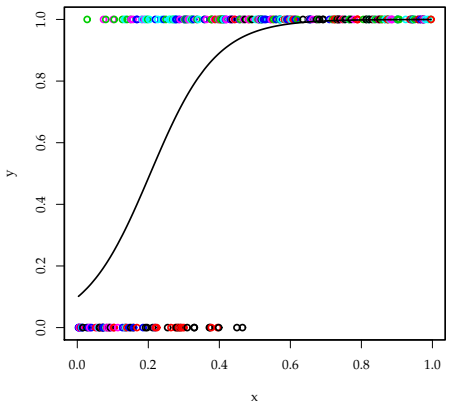
True relationship:

$$\Pr(Y = 1|x) \quad = \quad F(x, \beta, \nu)$$

8.1 Pooled linear regression

```
plot(y ~ x, data = data, col = i)
est.lm <- lm(y ~ x, data = data)
abline(est.lm)
```





8.2 Pooled logistic regression

$$Y = 1 \Leftrightarrow x + \epsilon_{ik} > \texttt{crit}$$

$$\Pr(Y = 1|x) \quad = \quad F(x, \beta)$$

```
data <- data[with(data, order(x, i)), ]
est.logit <- glm(y ~ x, family = binomial(link = "logit"),
    data = data)
summary(est.logit)
```

```
Call:
glm(formula = y ~ x, family = binomial(link = "logit"), data = data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.39122   0.02146   0.09098   0.44228   2.02723

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.2184     0.3855  -5.755 8.67e-09 ***
x            10.7891     1.4060   7.674 1.67e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 330.65  on 299  degrees of freedom
Residual deviance: 175.75  on 298  degrees of freedom
AIC: 179.75

Number of Fisher Scoring iterations: 7
```

```
plot(y ~ x, data = data, col = i)
lines(fitted(est.logit) ~ x, data = data)
```

8.3 Clustered logistic regression

```
data.oi <- data[order(data$i), ]
est.cluster <- geeglm(y ~ x, id = i, family = binomial(link = "logit"),
    data = data.oi)
summary(est.cluster)
```

```
Call:
geeglm(formula = y ~ x, family = binomial(link = "logit"), data = data.oi,
    id = i)

 Coefficients:
            Estimate Std.err   Wald   Pr(>|W|)
(Intercept)  -2.2184  0.7862  7.962    0.00478 **
x            10.7891  2.2654 22.682 0.00000191 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.613  0.7655

Correlation: Structure = independenceNumber of clusters:   6   Maximum cluster s
```

8.4 Non-parametric Wilcoxon test

```
estBetax <- sapply(by(data, list(i = data$i), function(data) glm(y ~
    x, family = binomial(link = "logit"), data = data)),
    coef)["x", ]
```

```
             a              b              c              d
32.235175970981 63.025582125904 -0.000000008405 40.055780646282
             e              f
97.119552216386 49.095677913257
```

```
wilcox.test(estBetax)
```

```
        Wilcoxon signed rank test

data:  estBetax
V = 20, p-value = 0.0625
alternative hypothesis: true location is not equal to 0
```

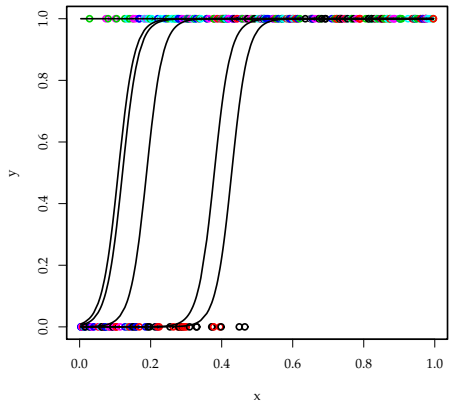
8.5 Fixed effects

$$Y = 1 \Leftrightarrow x + d_i + \epsilon_{ik} > \texttt{crit}$$

```
est.fixed <- glm(y ~ x + i, family = binomial(link = "logit"),
    data = data)
summary(est.fixed)
```

```
Call:
glm(formula = y ~ x + i, family = binomial(link = "logit"), data = data)

Deviance Residuals:
          Min             1Q         Median             3Q
-1.9025192211   0.0000000211   0.0000068046   0.0021905912
          Max
 1.8557222640

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -18.86       4.41   -4.27 0.000019 ***
x              44.06      10.19    4.32 0.000015 ***
ib              2.16       1.26    1.71  0.08691 .
ic             34.40    2456.85    0.01  0.98883
id             10.56       2.73    3.87  0.00011 ***
ie             13.54       3.23    4.20 0.000027 ***
if             14.08       3.50    4.03 0.000056 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 330.648  on 299  degrees of freedom
Residual deviance:  46.301  on 293  degrees of freedom
AIC: 60.3

Number of Fisher Scoring iterations: 20
```

```
plot(y ~ x, data = data, col = i)
qq <- sapply(c(coef(est.fixed)[-1:-2], 0), function(z) lines(plogis(cbind(1,
    data$x) %*% coef(est.fixed)[1:2] + z) ~ data$x))
```



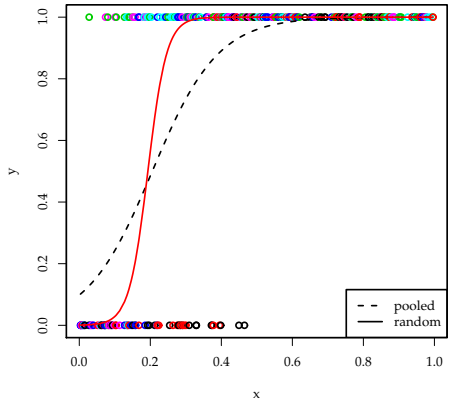
8.6 Random effects

$$Y = 1 \Leftrightarrow x + \nu_i + \epsilon_{ik} > \texttt{crit}$$

```
est.mer <- glmer(y ~ x + (1 | i), family = binomial(link = "logit"),
    data = data)
est.mer
```

```
Generalized linear mixed model fit by the Laplace approximation
Formula: y ~ x + (1 | i)
   Data: data
  AIC  BIC logLik deviance
 81.3 92.4  -37.7     75.3
Random effects:
 Groups Name        Variance Std.Dev.
 i      (Intercept) 40.1     6.33
Number of obs: 300, groups: i, 6

Fixed effects:
            Estimate Std. Error z value   Pr(>|z|)
(Intercept)    -7.31       3.08   -2.38      0.017 *
x              37.87       7.45    5.08 0.00000037 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
  (Intr)
x -0.520
```

```
plot(y ~ x, data = data, col = i)
lines(fitted(est.logit) ~ x, data = data, lty = 2)
lines(plogis(cbind(1, x) %*% cbind(fixef(est.mer))) ~
    x, data = data, col = "red")
legend("bottomright", c("pooled", "random"), lty = c(2,
    1))
```



Exercise 8.1 *The dataset in* `ex8b.csv` *contains data from a hypothetical study. We want to investigate how a control variable* `x1` *affects an outcome which can be either* `good` *or* `bad`*. We have several observations for each participant.*

1. *Estimate a pooled logistic model.*
2. *Estimate a logistic model with fixed effects for each participant.*
3. *Estimate a logistic model with a random effect for each participant. Compare the three models.*
4. *Does* `x1` *has an effect? Can you suggest a nonparametric test?*

9 Sample size

Before we run an experiment it would often be helpful to know something about the needed sample size. If we have at least some idea about the data generating process this can be done.

For a simple data generating processes there are formulas.

For more complicated ones we can simulate.

Let us assume we to investigate the impact of a stimulus on contribution in a public good game with random matching. The size of the interaction group is 4, the size of the matching group is 12, the experiment lasts for 10 periods. The endowment is between 0 and 10. From other experiments we expect an initial contribution of about 5 with a standard deviation of 3. We expect the contribution to decay by 0.2 units with a standard deviation of 1 from one period to the next.

- Define parameters of the simulation
- A function *player* provides the data we get from a single player:
- A function *group* combines a number of players to form a group.
- A function *groups* combines groups into the (random) experimental dataset.
- Apply random effects / Wilcoxon-Rank-Sum Test / ... to replicated versions of simulated datasets for experiments of different sizes.

Let us first define the parameters of our simulation.

```
meanContrib <- 5
sdContrib <- 3
meanChange <- -0.2
sdChange <- 1
effectSize <- 0.5
minContrib <- 0
maxContrib <- 10
periods <- 10
groupSize <- 12
```

A function *player* provides the data we get from a single player:

```
player <- function(pid = 1, gid = 1) {
    x <- round(rnorm(1, mean = meanContrib, sd = sdContrib) +
        rnorm(periods, mean = meanChange, sd = sdChange) +
        ifelse(effect, effectSize, 0), 0)
    cbind(gid = gid, pid = pid, period = 1:periods, contrib = pmin(pmax(minContrib,
        x), maxContrib), effect = effect)
}
```

A function *group* combines a number of players to form a group. Technically, we stack the players vertically, starting from an empty (*NULL*) matrix.

```
group <- function(gid = 1) {
    mGroupData <- NULL
    qq <- sapply(1:groupSize, function(p) mGroupData <<- rbind(mGroupData,
        player(p, gid)))
    mGroupData
}
```

Now we create the data for the hypothetical experiment.

```
groups <- function(numGroups) {
    allData <- NULL
    effect <<- FALSE
    sapply(1:(numGroups%/%2), function(gid) allData <<- rbind(allData,
        group(gid)))
    effect <<- TRUE
    qq <- sapply((numGroups%/%2 + 1):numGroups, function(gid) allData <<- rbin
        group(gid)))
    as.data.frame(allData)
}
```

Let us first check whether our simulation worked:

```
xx <- groups(2)
with(xx, table(pid))
```

```
pid
 1  2  3  4  5  6  7  8  9 10 11 12
20 20 20 20 20 20 20 20 20 20 20 20
```

```
with(xx, table(period))
```

```
period
 1  2  3  4  5  6  7  8  9 10
24 24 24 24 24 24 24 24 24 24
```

```
with(xx, table(gid, effect))
```

```
   effect
gid   0   1
  1 120   0
  2   0 120
```

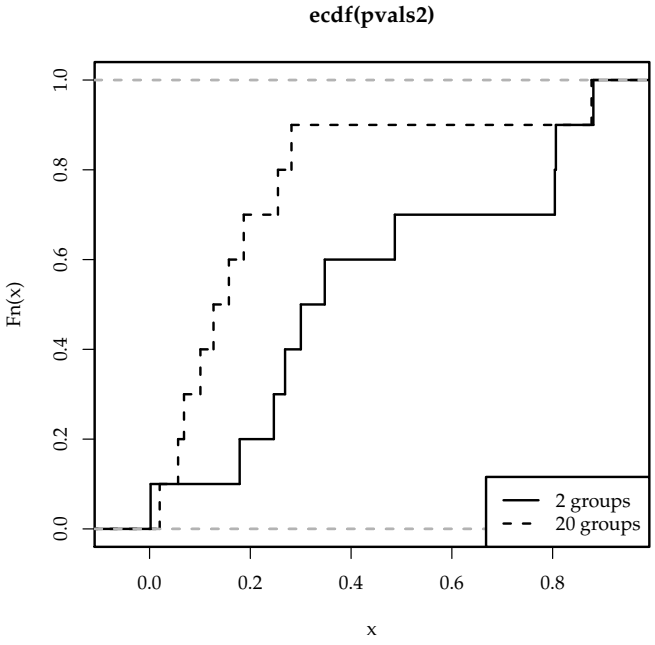
This looks fine. Now it is time to write a small function that calculates the statistics we care about for one simulated experiment. Let us assume that we care about p-values.

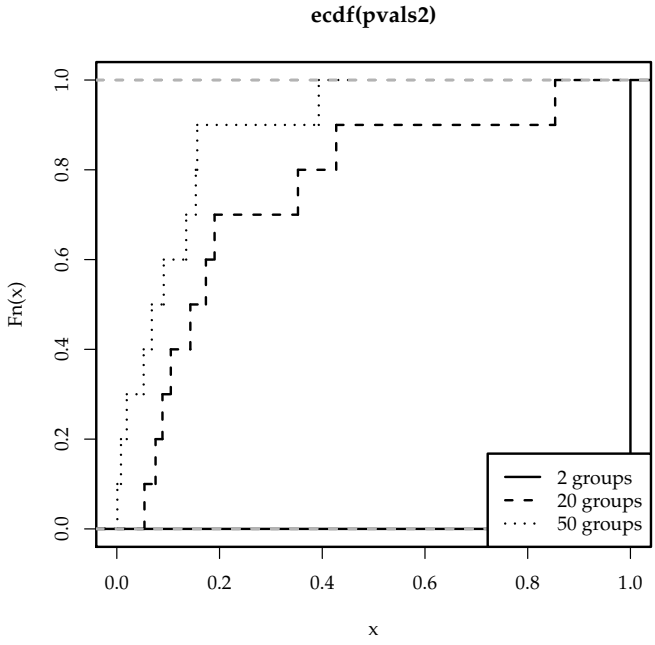
```
oneSimul <- function(groupNum) {
    xx <- groups(groupNum)
    est.mer <- lmer(contrib ~ effect + (1 | pid) + (1 |
        gid), data = xx)
    2 * pnorm(abs(summary(est.mer)@coefs["effect", "t value"]),
        lower = FALSE)
}
```

We use here t-statistics and assume that they follow a normal distribution. As pointed out above, this is a crude approximation. There are so many assumptions involved in this simulation that the mistake introduced by assuming normality is relatively small. The gain in computation time is large.

```
set.seed(123)
pvals2 <- replicate(10, oneSimul(2))
pvals20 <- replicate(10, oneSimul(20))
```

```
plot(ecdf(pvals2), do.p = FALSE, verticals = TRUE)
lines(ecdf(pvals20), do.p = FALSE, verticals = TRUE,
    lty = 2)
legend("bottomright", c("2 groups", "20 groups"), lty = 1:2,
    bg = "white")
```





Now let us assume that we are more conservative and want to apply a Wilcoxon rank sum test.

```
oneSimul <- function(groupNum) {
    xx <- groups(groupNum)
    wdata <- aggregate(xx, list(xx$gid), mean)
    wilcox.test(contrib ~ effect, data = wdata)$p.value
}
```

```
set.seed(123)
pvals2 <- replicate(10, oneSimul(2))
pvals20 <- replicate(10, oneSimul(20))
pvals50 <- replicate(10, oneSimul(50))
```

```
plot(ecdf(pvals2), do.p = FALSE, verticals = TRUE, xlim = c(0,
    1))
lines(ecdf(pvals20), do.p = FALSE, verticals = TRUE,
    lty = 2)
lines(ecdf(pvals50), do.p = FALSE, verticals = TRUE,
    lty = 3)
legend("bottomright", c("2 groups", "20 groups", "50 groups"),
    lty = 1:3, bg = "white")
```

Exercise 9.1 *You want to design a field experiment to test the effect of labour market qualification. Your dependent variable will be the salaries of your participants during the next five years. Each year you will have one observation for each participant. You assume that the qualification program will lead to an increase of the annual income of about 500$. You also assume that, within a participant, the standard deviation on the income from year to year is about 2 000$. Furthermore, you assume that across individuals the standard deviation of the income is about 20 000$.*

1. *How many participants do you need if 50% of your sample will participate in the qualification program? Assume that your significance level is 5%.*
2. *You know that you can put 300 participants into the qualification treatment. You can put any number into the control treatment. Can you expect significant results? If so, how large should your control group be?*

10 Exercises

Exercise 10.1 *The dataset* `exe1` *from the attached file* `me.Rdata` *provides data on a simple experiment.* `i` *denotes the individual,* `x` *is some independent stimulus,* `y` *is the reaction of the individual.*

1. *How many individuals are included? How many measurements do we have per individual?*
2. *Estimate a pooled OLS, between OLS, clustered OLS, fixed effects OLS, and a random effects OLS model. For each model provide a confidence interval for β_x. Also provide a non parameteric test whether the marginal effect of $\underline{x}$ is positive.*

Exercise 10.2 *Have a look at the data in* `exe2`*. The variable* `y` *is the reaction of the individual player* `player` *to different treatments* `treatment`*. The different periods are coded as* `period`*.*

1. *How many individuals are included? How many measurements do we have per individual? How many measurements do we have for each treatment?*
2. *What is the appropriate estimation procedure here?*
3. *Do you find any differences between treatments?*

4. *Do you find any differences between players?*
5. *Do you find a time trend?*
6. *One of your hypotheses is that treatments 1 to 4 yield a higher value of the dependent variable y. Can you confirm this hypothesis? Give a confindence interval of an appropriate statistic?*