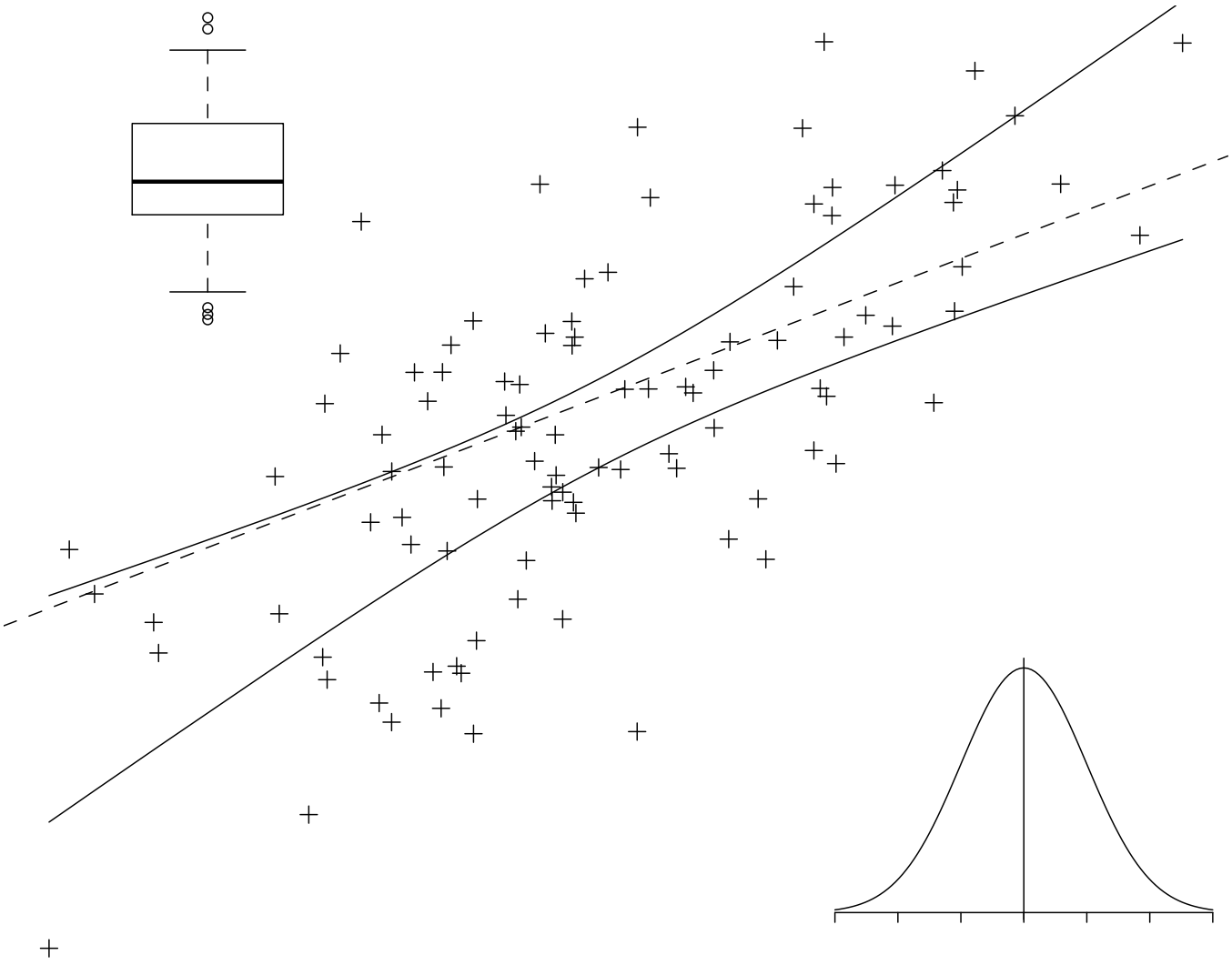
Mixed effects models — 2016



Oliver Kirchkamp

Contents

1 Introduction

1.1 Scope

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.

- John K. Kruschke , Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. Academic Press, 2nd Edition, 2014.
- A. C. Davison, D. V. Hinkley, Bootstrap Methods and their Application, Cambridge University Press, 1997
- Bradley Efron and Robert J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, 1994

1.2 Terminology

Fixed-Effects, Random-Effects, Mixed-Models FE-only-model:

$$y_j = \beta X_j + u_j \qquad \text{with } u_j \sim N(0, \sigma^2 \mathbf{I}_n)$$

ME-model: There are $i \in \{1 \dots M\}$ groups of data, each with $j \in \{1 \dots n_i\}$ observations. Groups i represent one or several factors.

$$y_{ij} = \beta X_{ij} + \gamma_i Z_{ij} + u_{ij} \qquad \text{with } \gamma_i \sim N(0, \Psi), u_{ij} \sim N(0, \Sigma_i)$$

- Fixed effects: The researcher is interested in specific effects β. The researcher does not care/is ignorant about the type of distribution of β. If β is a big object, estimation can be expensive.
- Random effects: The researcher only cares about the distribution of γ. It is sufficient to estimate only the parameters of the distribution of γ (e.g. its standard deviation). This is less expensive/makes more efficient use of the data.
- Mixed model: includes a mixture of fixed and random factors

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.

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

Problems 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 $1 \dots 12$
- k group $1 \dots 10$
- t time $1 \dots 8$
- 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.

1.3 A small Example

We will illustrate most of our examples with R. Input and output will be shown in a frame, like this:

```
1+1

[1] 2
```

To accomodate those of you how are already familiar with Stata, we will also (sometimes) have a look at the Stata notation. Stata input will be shown like this:

```
display 1+1
2
```

Regardless whether you work with R or with Stata, it is strongly recommended that you try out the code yourself.

1.4 Starting R

During this course we will use 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(Ecdat)
library(car)
library(Hmisc)
load(file="data/me.Rdata")
```

1.5 A small example

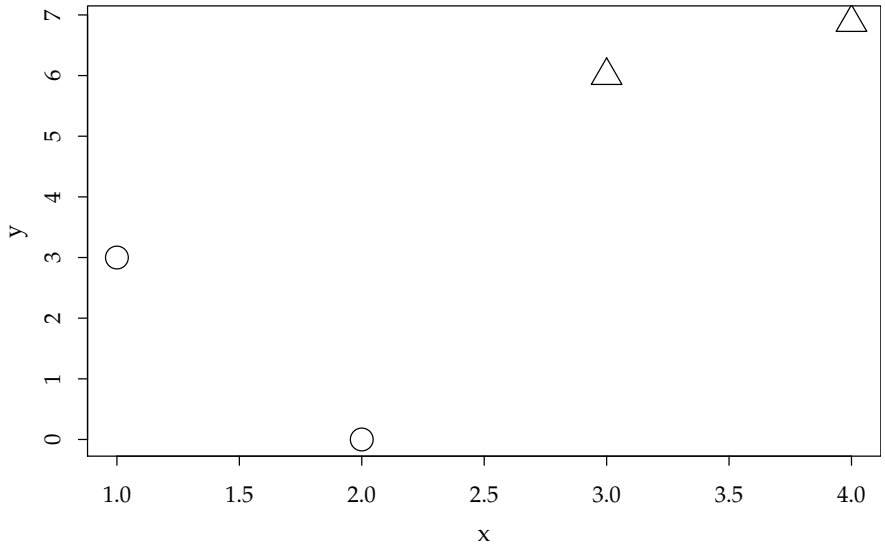
The following figure shows the predicted relationship for the various methods. The dataset is very simple. We have only four observations, two from two groups each. The first groups (shown as circles) are at the bottom left of the diagram, the second group (triangles) are top right.

```
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
```

```
plot(y ~ x,data=simple,pch=i,cex=2)
```



In Stata we could input the data in a similar way:

```
input i x y
1 1 3
1 2 0
2 3 6
2 4 6.8744
end
```

Let us now look at the following models:

- Standard OLS:

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

- Between OLS:

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

- 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.

- 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 $\epsilon_{ik} \sim N(0, \sigma)$

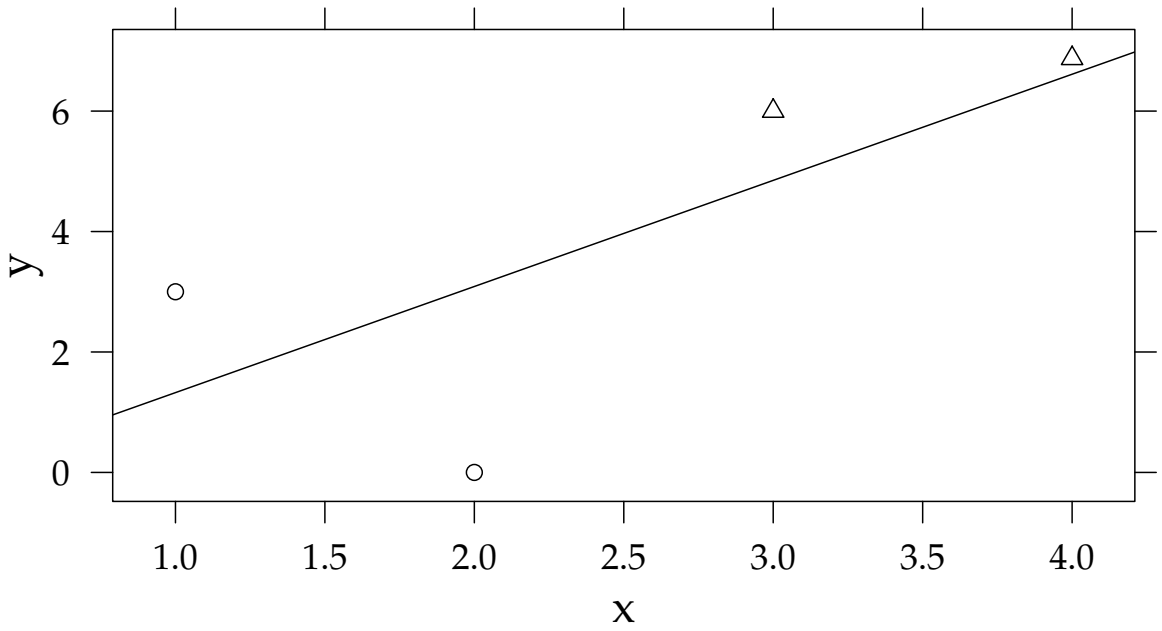
Let us now try to use these models in R and in Stata:

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)
```

or in Stata:

```
regress y x
```



Between OLS:

$$y_i = \beta_0 + \beta_1 x_i + \nu_i$$

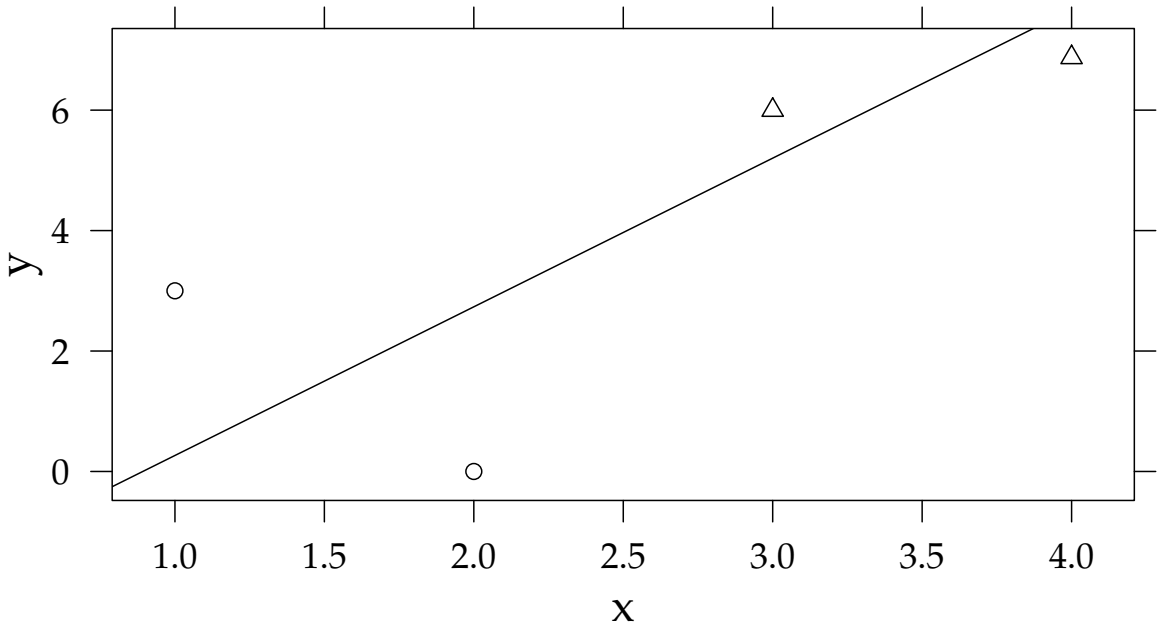
with $\nu_i \sim N(0, \sigma)$

```
betweenSimple <- with(simple,aggregate(simple,list(i),mean))
betweenSimple

  Group.1   x      y i
1       1 1.5 1.5000 1
2       2 3.5 6.4372 2

betweenOLS <- lm (y ~ x,data=betweenSimple)
```

Turning to Stata we see that Stata has a different attitude towards datasets (yet). While in R we usually work with different datasets which are stored in variables, in Stata we

only work with one dataset at a time. Any manipulation of the dataset means that we either lose the old dataset, or we have to explicitely save it somewhere. We can either save it as a file on disk and later retrive it there, or we store it in a special place, called *preserve*.

```
preserve
collapse y x,by(i)
regress y x
restore
```



Fixed effects for groups i:

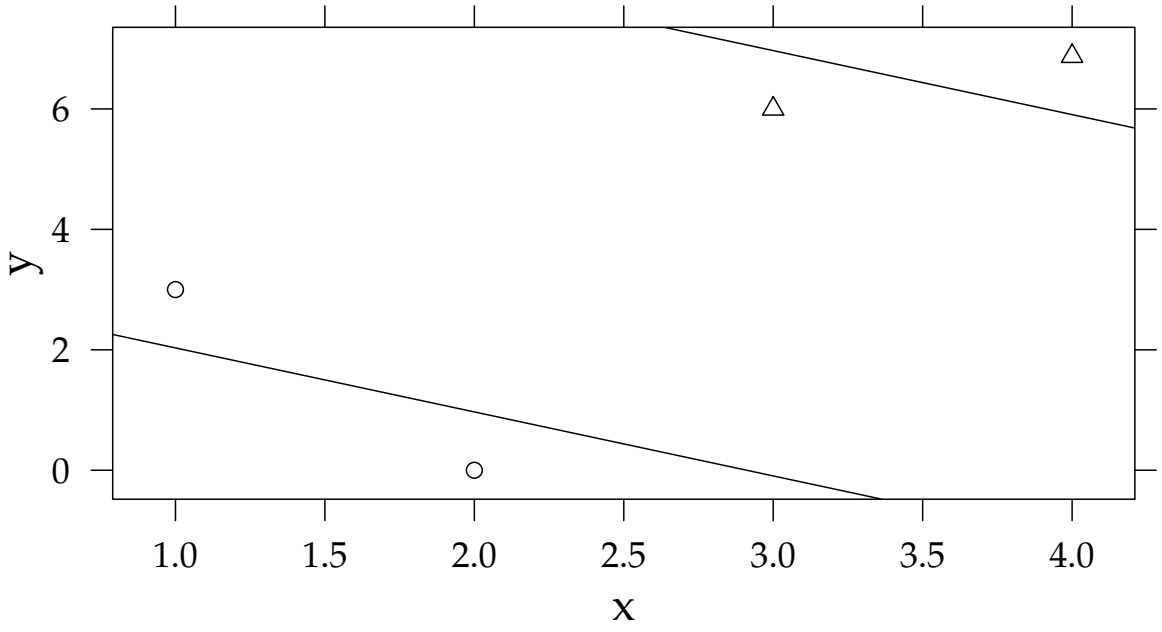
$$y_{ik} = \beta_0 + \beta_1 x_{ik} + \sum_i \gamma_i d_i + \epsilon_{ik}$$

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)
```

```
xi i.i,noomit
regress y x _Ii*,noconstant
```



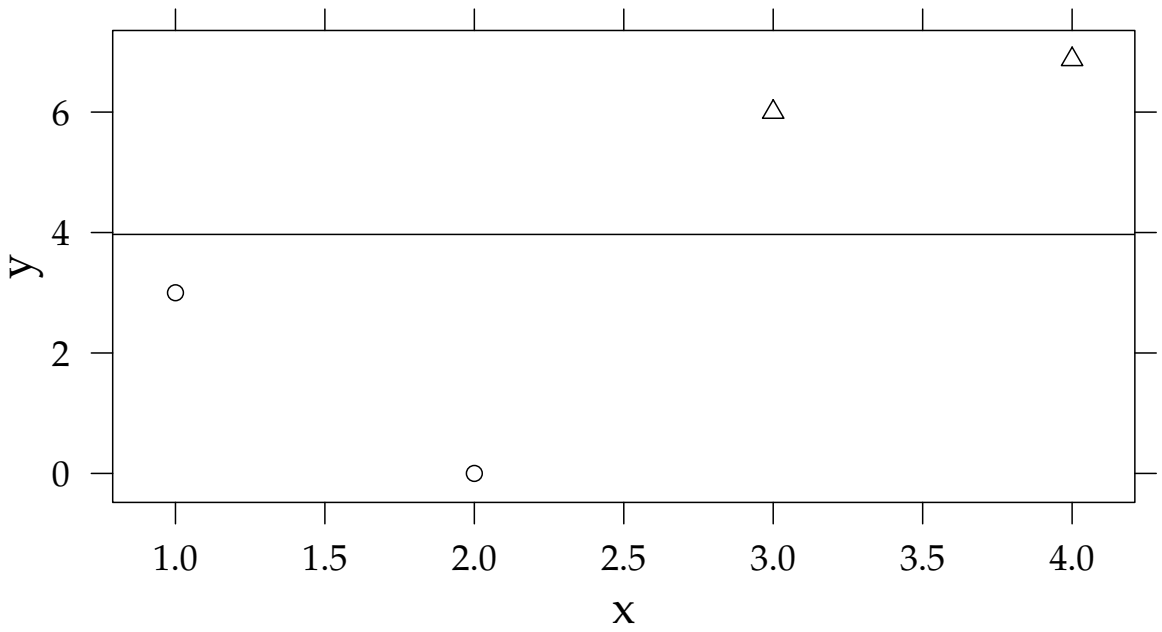
Random effects for groups i:

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

with $\nu_i \sim N(0, \sigma_\nu)$ and $\epsilon_{ik} \sim N(0, \sigma)$

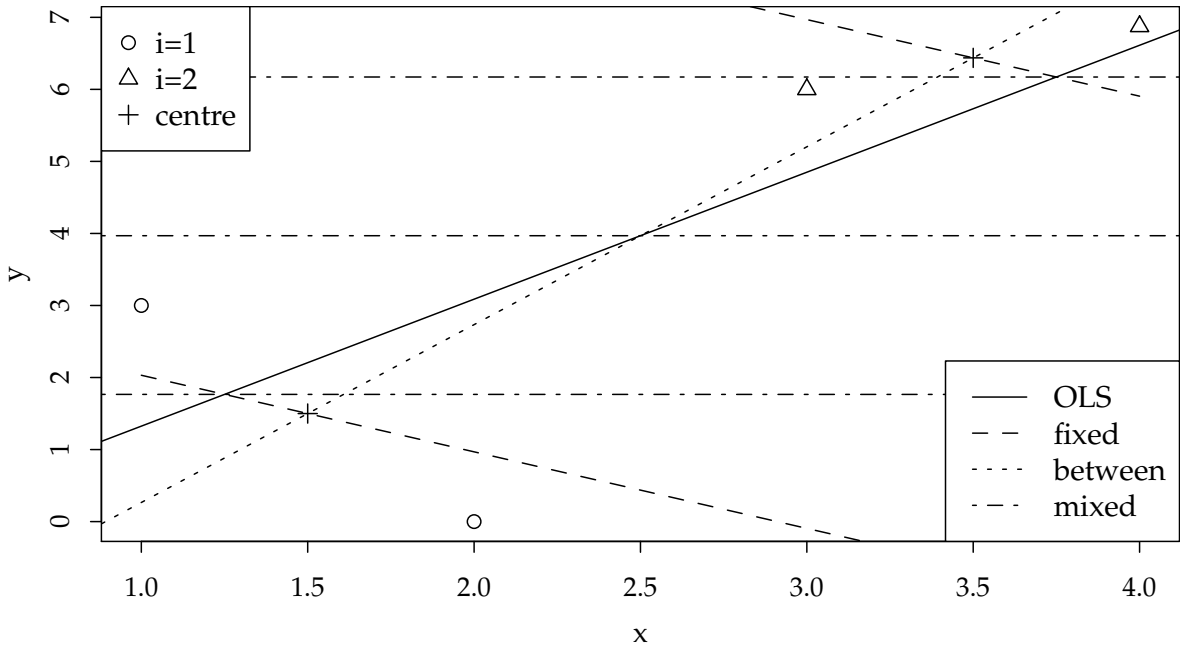
```
library(lme4)
ranef <- lmer(y ~ x + (1|i),data=simple)
```

```
xtset i
xtmixed y  x || i:
estimates store mixed
```



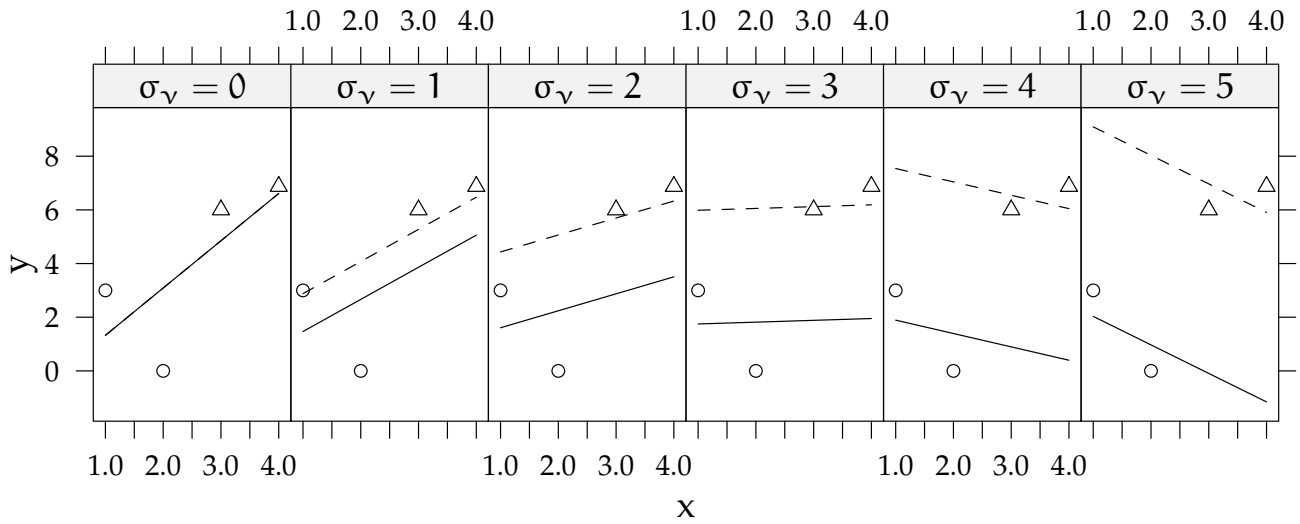
Here is a picture of the different estimated regression lines:

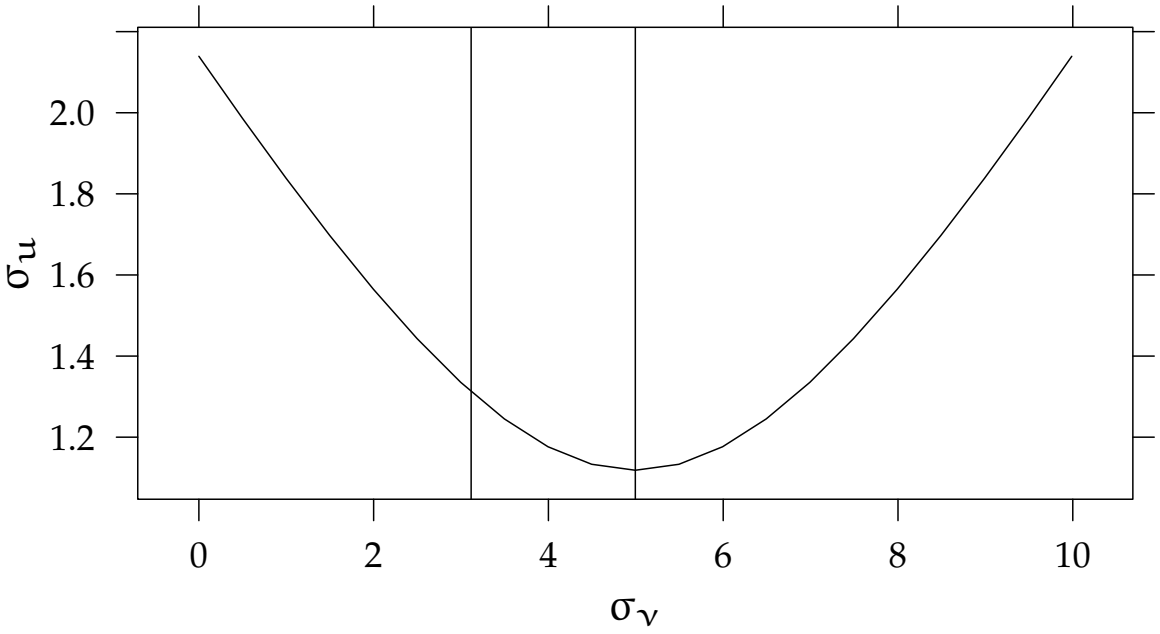
```
plot(y ~ x,data=simple,pch=i)
points(betweenSimple[,c("x","y")],pch=3)
abline(ol)
abline(betweenOLS,lty=3)
qq <- sapply(unique(simple$i),function(g)
             lines(predict(fixef,newdata=within(simple,i<-g)) ~ x,
                   data=simple,lty=2))
abline(fixef(ranef),lty=4)
qq<-apply(coef(ranef)$i,1,abline,lty=4)
```



We see that, depending on the model, we can get anything between a positive and a negative relationship.

The following graph illustrates the transition from the pure OLS model (left) to the pure fixed effects model (right):





OLS allows no fixed effect ($\sigma_\nu = 0$), all the variance goes into u.
Fixed effects puts no restriction on σ_ν, hence σ_u can be minimal.
Random effects is in between.

- 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 mixed effects yields an estimation between the fixed effect and the (pooled) OLS estimation.

$$
\begin{array}{lrcll}
\text{between} & y_i & = & \beta_0+\beta_1 x_i & +\nu_i \\
\text{OLS} & y_{ik} & = & \beta_0+\beta_1 x_{ik} & +\epsilon_{ik} \\
\text{mixed effects} & y_{ik} & = & \beta_0+\beta_1 x_{ik} & +\nu_i+\epsilon_{ik} \\
\text{fixed effects} & y_{ik} & = & \beta_0+\beta_1 x_{ik} & +\sum_i \gamma_i d_i+\epsilon_{ik}
\end{array}
$$

	ν_i	ϵ_{ik}
between	finitely expensive	cheap (no cost)
OLS	0 (infinitely expensive)	finitely expensive
mixed effects	finitely expensive	finitely expensive
fixed effects	cheap (no cost)	finitely expensive

2 6 different methods - 6 different results

2.1 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
- Mixed 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))
```

We also save the data to do the same exercise in Stata:

```
write.csv(data,file="csv/methods6.csv",row.names=FALSE)
```

2.2 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
```

```
clear
insheet using csv/methods6.csv
regress y x
estimates store ols

Source |       SS       df       MS              Number of obs =     300
-------------+------------------------------           F(  1,   298) =    0.33
       Model |  29.5423583     1  29.5423583           Prob > F      =  0.5683
    Residual |  26980.8152   298  90.5396484           R-squared     =  0.0011
-------------+------------------------------           Adj R-squared = -0.0023
       Total |  27010.3576   299  90.3356441           Root MSE      =  9.5152
```

```
------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.071069   1.875056     0.57   0.568     -2.61896    4.761099
       _cons |  -3.826152   1.092081    -3.50   0.001     -5.97532   -1.676985
------------------------------------------------------------------------------
```

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

2.3 Clustered OLS

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

```
library(geepack)
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)   -3.826   3.730 1.052    0.305
x              1.071   0.968 1.224    0.269

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    89.94   39.66

Correlation: Structure = independenceNumber of clusters:   6   Maximum cluster size: 50
```

```
regress y x,cluster(i)
```

```
Linear regression                                      Number of obs =     300
                                                       F(  1,     5) =    1.02
                                                       Prob > F      =  0.3596
                                                       R-squared     =  0.0011
                                                       Root MSE      =  9.5152

                                     (Std. Err. adjusted for 6 clusters in i)
------------------------------------------------------------------------------
             |               Robust
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
```

```
-------------+----------------------------------------------------------------
           x |   1.071069   1.062213     1.01   0.360    -1.659436    3.801575
       _cons |  -3.826152   4.092947    -0.93   0.393    -14.34741    6.695103
-------------------------------------------------------------------------------
```

- 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 $\mathbf{X}$.
- Estimation of $\Sigma_{\beta\beta}$ 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, 1993, Regression standard errors in clustered samples, STB 13. .

2.4 Between OLS

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

```
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
```

```
preserve
collapse x y,by(i)
regress y x
restore
```

```
      Source |       SS       df       MS              Number of obs =       6
-------------+------------------------------           F(  1,     4) =    0.00
       Model |   .01975652     1   .01975652           Prob > F      =  0.9905
    Residual |  490.144171     4  122.536043           R-squared     =  0.0000
-------------+------------------------------           Adj R-squared = -0.2499
       Total |  490.163928     5  98.0327855           Root MSE      =   11.07

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.715449   135.0998     0.01   0.990    -373.3816    376.8125
       _cons |  -4.150513   68.15495    -0.06   0.954     -193.379     185.078
------------------------------------------------------------------------------
```

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

2.5 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",])

        1         2         3         4         5         6
 0.009656 -0.210645  2.944480 -0.898667  1.828669  2.622907

wilcox.test(estBetax)


Wilcoxon signed rank test

data:  estBetax
V = 16, p-value = 0.3
alternative hypothesis: true location is not equal to 0
```

```
statsby, by(i) clear: regress y x
signrank _b_x=0
```

```
Wilcoxon signed-rank test

        sign |      obs   sum ranks    expected
-------------+--------------------------------
    positive |        4          16        10.5
    negative |        2           5        10.5
```

```
        zero |        0           0           0
-------------+---------------------------------
         all |        6          21          21

unadjusted variance       22.75
adjustment for ties        0.00
adjustment for zeros       0.00
                     ----------
adjusted variance         22.75

Ho: _b_x = 0
             z =   1.153
    Prob > |z| =   0.2489
```

Why do the results of Stata and R differ? R can calculate the results provided by Stata, too, if it is called as follows:

```
wilcox.test(estBetax,exact=FALSE,correct=FALSE)


Wilcoxon signed rank test

data:  estBetax
V = 16, p-value = 0.2
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 $\mathbf{X}_i$.
- σ_β can be estimated as the standard deviation of β_i.
- Efficiency $\rightarrow$ less efficient than fixed or mixed effects, since we do not exploit any relative differences.

2.6 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
```

```
-7.617 -2.186  0.064  2.194  7.103

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
```

```
xi i.i,noomit
regress y x _Ii*,noconstant
estimates store fixed
```

```
      Source |       SS       df       MS              Number of obs =     300
-------------+------------------------------           F(  7,   293) =  470.08
       Model |  27778.2215     7  3968.31736           Prob > F      =  0.0000
    Residual |  2473.46655   293  8.44186537           R-squared     =  0.9182
-------------+------------------------------           Adj R-squared =  0.9163
       Total |   30251.688   300   100.83896           Root MSE      =  2.9055

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |    1.06256   .5763195     1.84   0.066    -.0716903    2.196811
       _Ii_1 |  -.4465983   .5209705    -0.86   0.392    -1.471917    .5787203
       _Ii_2 |  -2.878954   .5032844    -5.72   0.000    -3.869465   -1.888443
       _Ii_3 |   -20.6976   .5052905   -40.96   0.000    -21.69206   -19.70314
       _Ii_4 |  -9.253381   .4935815   -18.75   0.000     -10.2248   -8.281966
       _Ii_5 |   4.231748   .4864067     8.70   0.000     3.274454    5.189041
       _Ii_6 |   6.113573     .50987    11.99   0.000     5.110101    7.117044
------------------------------------------------------------------------------
```

- Estimation of β is consistent if residuals (ϵ_{it}) are uncorrelated with $\mathbf{X}$. This is a weaker requirement, since, with fixed effects, residuals are only ϵ_{ikt}, not ν_i.
- Estimation of $\Sigma_{\beta\beta}$ is consistent.
- The procedure loses 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*

2.7 Mixed effects

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

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

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i)
   Data: data
REML criterion at convergence: 1522
Random effects:
 Groups   Name        Std.Dev.
 i        (Intercept) 9.89
 Residual             2.91
Number of obs: 300, groups:  i, 6
Fixed Effects:
(Intercept)            x
      -3.82         1.06
```

```
summary(mixed)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i)
   Data: data

REML criterion at convergence: 1522

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.6156 -0.7473  0.0279  0.7553  2.4467

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
```

```
xtmixed y x || i:
```

```
Mixed-effects REML regression                   Number of obs      =       300
Group variable: i                               Number of groups   =         6

                                                Obs per group: min =        50
                                                               avg =      50.0
                                                               max =        50


                                                Wald chi2(1)       =      3.40
Log restricted-likelihood = -761.07079          Prob > chi2        =    0.0652

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.062575   .5763129     1.84   0.065    -.0669773    2.192128
       _cons |  -3.821877   4.052465    -0.94   0.346    -11.76456     4.12081
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
i: Identity                  |
                  sd(_cons) |   9.892476   3.133668      5.317009    18.40529
-----------------------------+------------------------------------------------
               sd(Residual) |   2.905489   .1200246      2.679516    3.150518
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =   675.22 Prob >= chibar2 = 0.0000
```

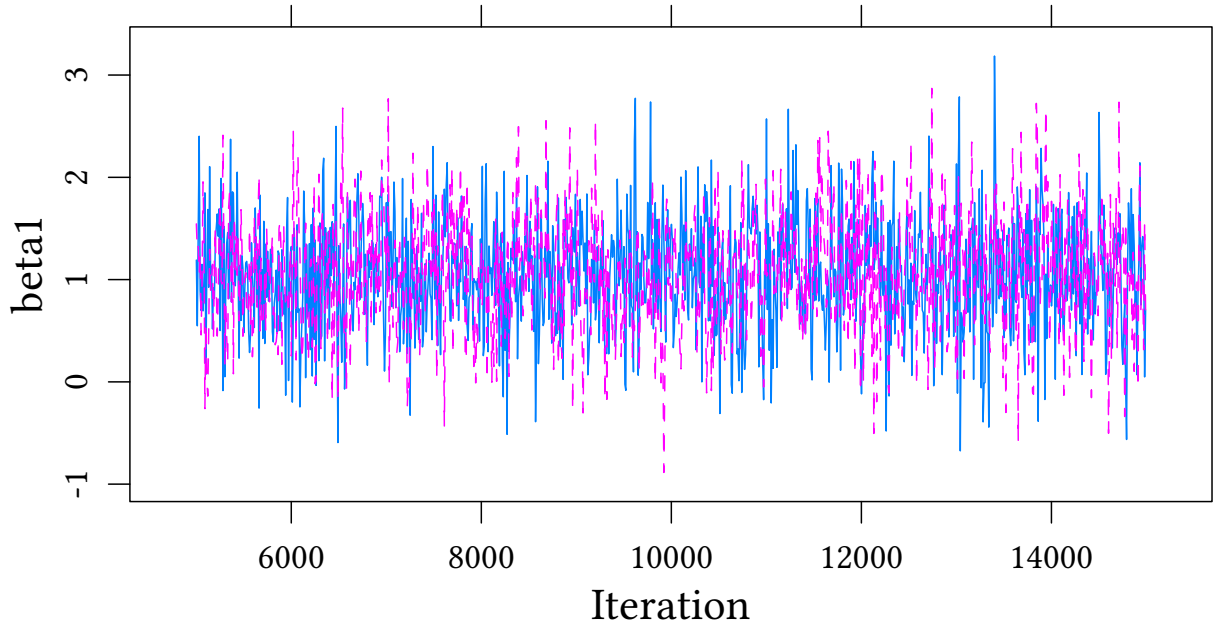
- Estimation of β is consistent if residuals ν_i and ϵ_{it} are uncorrelated with $\mathbf{X}$.
- This is a stronger requirement than with fixed effects, since we also impose a restriction on ν_i.

  (e.g., what, if participants self select into treatments?)
- Estimation of $\Sigma_{\beta\beta}$ may require the bootstrap.

2.8 Bayes and mixed effects

```
modelM <- 'model {
 for (i in 1:length(y)) {
        y[i] ~ dnorm(beta0+beta1*x[i]+nu[group[i]],tau)
 }
 for (j in 1:max(group)) {
    nu[j] ~ dnorm(0,tauNu)
 }
 beta0   ~ dnorm (0,.0001)
 beta1   ~ dnorm (0,.0001)
 tau     ~ dgamma(.01,.01)
 tauNu   ~ dgamma(.01,.01)
 sd     <- sqrt(1/tau)
 sdNu   <- sqrt(1/tauNu)
}'
dataList<-list(y=data$y,x=data$x,group=as.numeric(data$i))
bayesM<-run.jags(model=modelM,data=dataList,
                 monitor=c("beta0","beta1","sd","sdNu"))
```

```
plot(bayesM,"trace",vars=c("beta1"))
```




```
summary(bayesM)

      Lower95 Median Upper95  Mean    SD  Mode     MCerr MC%ofSD SSeff    AC.10
beta0 -9.4995  -3.01    3.66 -3.01 3.535 -3.37 0.717445    20.3    24 0.97649
beta1 -0.0623   1.06    2.20  1.05 0.571  1.04 0.010634     1.9  2880 0.06234
sd     2.6876   2.91    3.15  2.91 0.120  2.91 0.000883     0.7 18486 0.00561
sdNu   5.4302  10.21   19.21 11.13 4.167  9.23 0.067185     1.6  3846 0.05365
      psrf
beta0 1.07
beta1 1.00
sd    1.00
sdNu  1.00
```

```
fixef(mixed)

(Intercept)           x
      -3.82        1.06
```

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

2.9 The power of the 6 methods

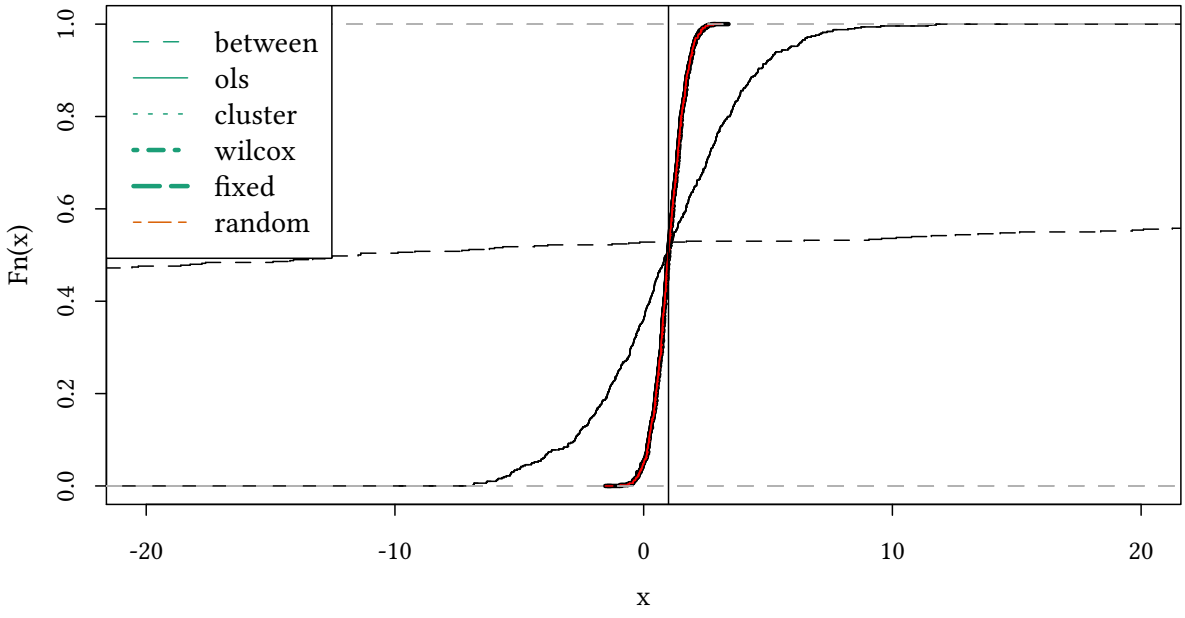
We repeat the above exercise 500 times. Each time we look at the estimated coefficient β_x and at the p-value of testing $\beta_x = 0$ against $\beta_x \neq 0$.

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:

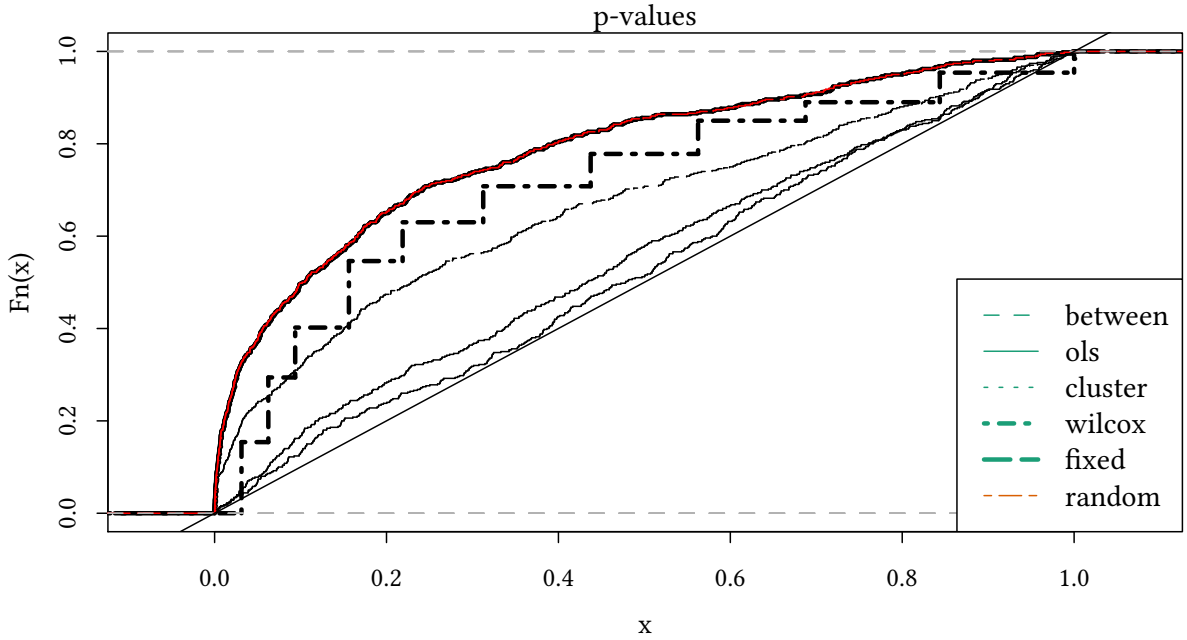


The good news is: All estimators seem to be unbiased (although the *between* estimator has a huge variance here). Also *OLS* and *clustered OLS* are not very efficient. Sometimes they estimate values that are far away from the true value $\beta_x = 1$.

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	23.80	15.40	37.40	37.60



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

```
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
```

```
print.xtable(xtable(rbind(apply(pval,1,function(x) mean(x<.05))*100)),includ
```

between	ols	cluster	wilcox	fixed	random.x
6.80	7.80	23.80	15.40	37.40	37.60

```
simul2 <- mclapply(1:500,function(x) exa(seed=x,I=6,T=20))
pval2 <- sapply(simul2,function(x) x["p",1:6])
print.xtable(xtable(rbind(apply(pval2,1,function(x) mean(x<.05))*100)),inclu
```

between	ols	cluster	wilcox	fixed	random.x
4.80	6.20	20.60	8.00	19.20	20.00

```
simul3 <- mclapply(1:500,function(x) exa(seed=x,I=6,T=6))
pval3 <- sapply(simul3,function(x) x["p",1:6])
print.xtable(xtable(rbind(apply(pval3,1,function(x) mean(x<.05))*100)),inclu
```

between	ols	cluster	wilcox	fixed	random.x
4.80	5.20	15.40	3.20	8.20	9.00

- More noisy data → fewer significant results.

→ lower p-values in wilcox, fixed and random.

- "Cluster" appears to be much more significant – this can not be correct.

```
print.xtable(xtable(rbind(apply(pval,1,function(x) mean(x<.05))*100)),inc
```

between	ols	cluster	wilcox	fixed	random.x
6.80	7.80	23.80	15.40	37.40	37.60

```
simul5 <- mclapply(1:500,function(x) exa(seed=x,I=500,T=2))
pval5 <- sapply(simul5,function(x) x["p",1:6])
print.xtable(xtable(rbind(apply(pval5,1,function(x) mean(x<.05))*100)),in
```

between	ols	cluster	wilcox	fixed	random.x
3.40	7.60	6.80	21.60	54.00	55.40

3 Estimation results

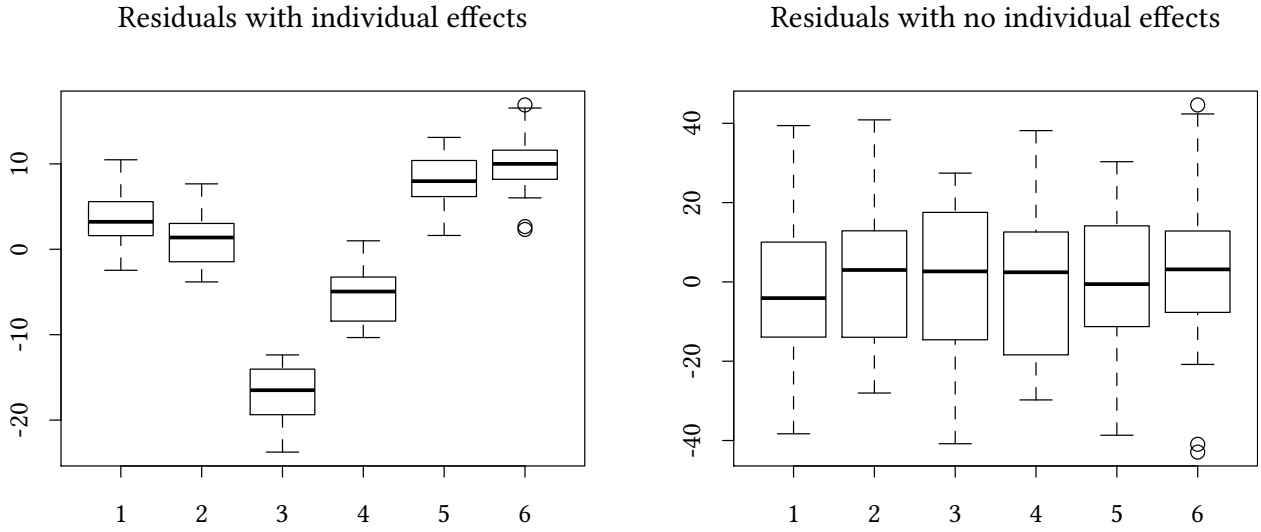
3.1 Residuals

3.2 OLS residuals

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

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

```
ols2 <- lm (y2 ~ x,data=data)
with(data,boxplot(residuals(ols) ~ i,main="Residuals with individual effects"))
with(data,boxplot(residuals(ols2) ~ i,main="Residuals with no individual effects"))
```

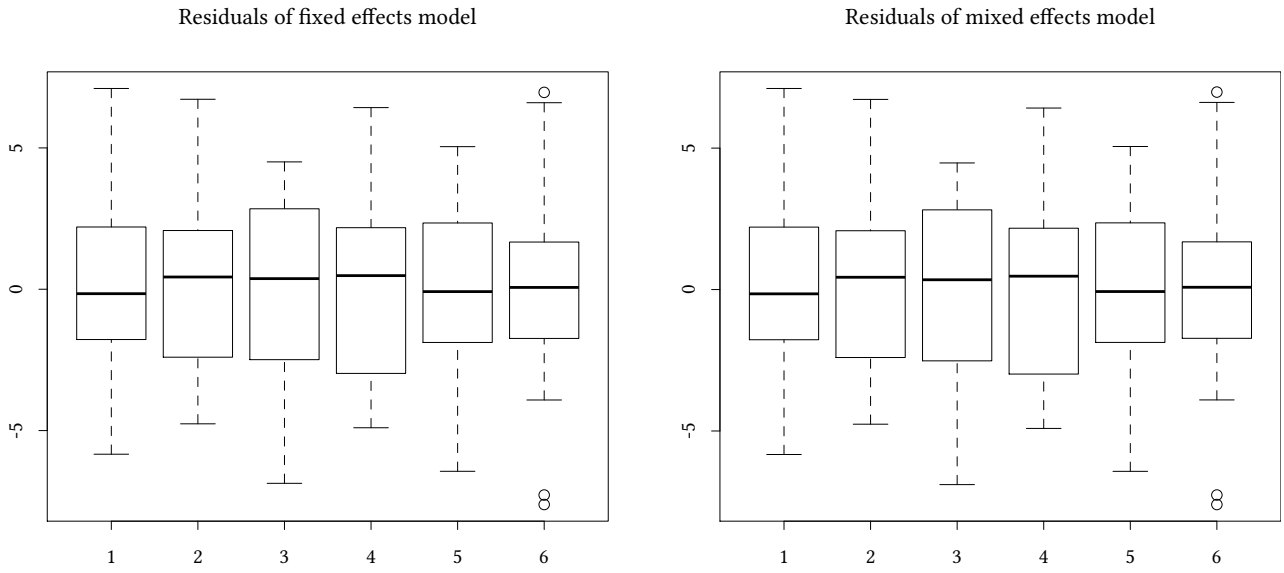



```
clear
insheet using csv/methods6.csv
regress y x
predict resid,residuals
graph box resid,over(i)
regress y2 x
predict resid2,residuals
graph box resid2,over(i)
```

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.

3.3 Fixed- and mixed effects residuals

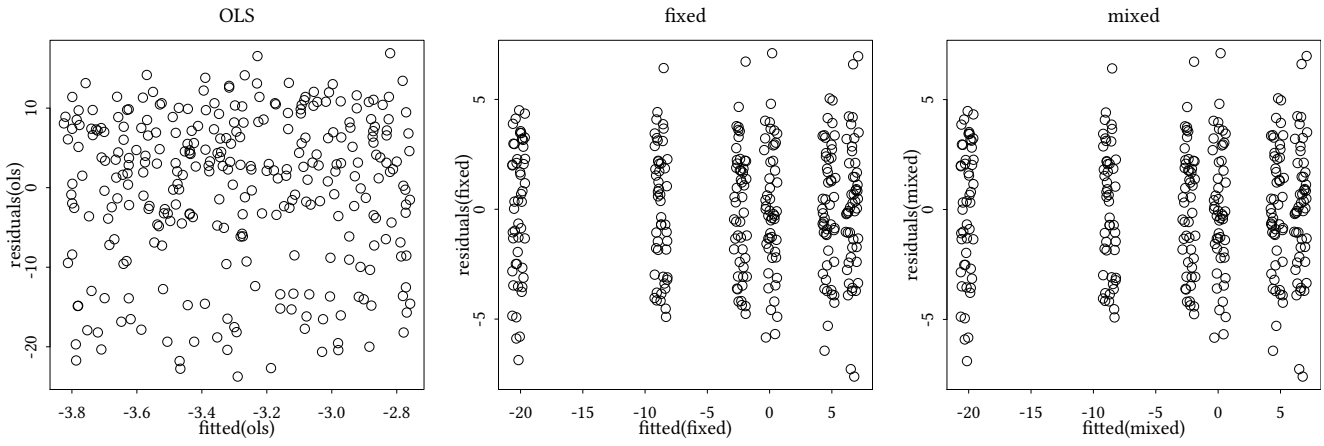
```
with(data,boxplot (residuals(fixed) ~ i,main="Residuals of fixed effects model"))
with(data,boxplot (residuals(mixed) ~ i,main="Residuals of mixed effects model"))
```



3.4 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:

```
plot(residuals(ols) ~ fitted(ols),main="OLS")
plot(residuals(fixed) ~ fitted(fixed),main="fixed")
plot(residuals(mixed) ~ fitted(mixed),main="mixed")
```



Exercise 3.1 *What can you say about the distribution of residuals of your estimates for* `ex1.csv`*?*

3.5 Estimated standard errors

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

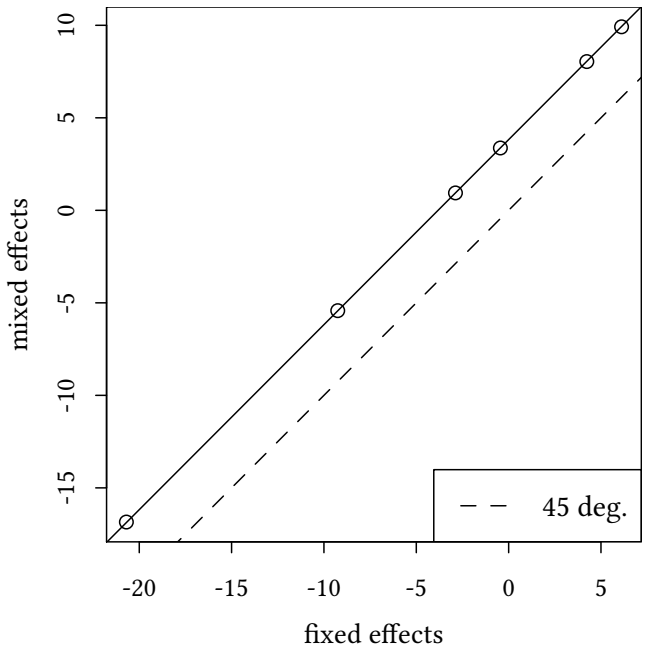
Let us compare the estimated standard errors of the residuals ϵ_{ikt}

```
summary(ols)$sigma

[1] 9.515232

summary(fixed)$sigma

[1] 2.905489

sigma(mixed)

[1] 2.905489
```

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

3.6 Estimated effects

```
plot(coef(fixed)[-1],ranef(mixed)$i[,"(Intercept)"],
     xlab="fixed effects",ylab="mixed effects")
abline(a=-mean(coef(fixed)[-1]),b=1)
abline(a=0,b=1,lty=2)
legend("bottomright","45 deg.",lty=2)
```



We see that the estimated effects for the fixed effects and for the mixed effects model are similar. Since the RE model contains an intercept, the dots may not be on the 45° line.

3.7 Information criteria

$$\mathrm{AIC} = -2\log \mathrm{L} + 2\mathrm{k}$$

```
AIC(ols)

[1] 2207.093

AIC(fixed)

[1] 1500.241
```

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

```
AIC(mixed)

[1] 1530.142
```

```
mixedML <- update(mixed,REML=FALSE)
AIC(mixedML)

[1] 1535.413
```

```
estimates restore ols
estat ic
```

```
-----------------------------------------------------------------------------
       Model |    Obs    ll(null)   ll(model)     df          AIC         BIC
-------------+---------------------------------------------------------------
         ols |    300   -1100.711   -1100.546      2     2205.093      2212.5
-----------------------------------------------------------------------------
               Note:  N=Obs used in calculating BIC; see [R] BIC note
```

```
estimates restore fixed
estat ic
```

```
-----------------------------------------------------------------------------
       Model |    Obs    ll(null)   ll(model)     df          AIC         BIC
-------------+---------------------------------------------------------------
       fixed |    300           .   -742.1206      7     1498.241    1524.168
-----------------------------------------------------------------------------
               Note:  N=Obs used in calculating BIC; see [R] BIC note
```

```
estimates restore mixed
estat ic
```

```
-----------------------------------------------------------------------------
       Model |    Obs    ll(null)   ll(model)     df          AIC         BIC
-------------+---------------------------------------------------------------
       mixed |    300           .   -761.0708      4     1530.142    1544.957
-----------------------------------------------------------------------------
               Note:  N=Obs used in calculating BIC; see [R] BIC note
```

3.8 Hausman test

- Fixed effects estimator is consistent but inefficient
- Mixed 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 mixed 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(fixef(random))
    fNames <- names(coef(fixed))
    timevarNames <- intersect(rNames,fNames)
    k <- length(timevarNames)
    rV <- vcov(random)
    rownames(rV)=rNames
    colnames(rV)=rNames
    bDiff <- (fixef(random))[timevarNames] - coef(fixed)[timevarNames]
    vDiff <- vcov(fixed)[timevarNames,timevarNames] - rV[timevarNames,timevarNames]
    H <- as.numeric(t(bDiff) %*% solve(vDiff) %*% bDiff)
    c(H=H,p.value=pchisq(H,k,lower.tail=FALSE))
}
hausman(fixed,mixed)

           H      p.value
0.0000290819 0.9956972182
```

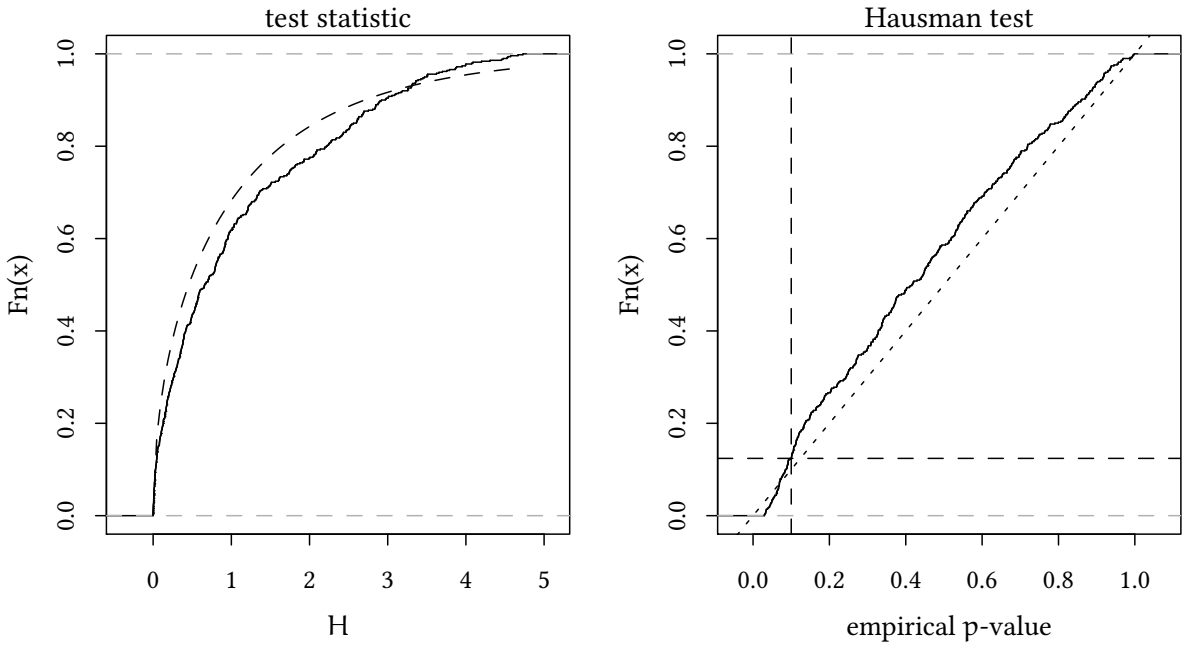
We see that in our example there is no reason not to use random effects. (For completeness: We are looking here at the difference of two variance-covariance matrices, hence it is possible that the Hausman statistic becomes negative)

```
insheet using csv/methods6.csv,clear
xi i.i,noomit
regress y x _Ii*,noconstant
est store fixed
xtmixed y  x || i:
est store mixed
hausman fixed mixed,equations(1:1)
```

```
                 ---- Coefficients ----
              |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
              |     fixed        mixed       Difference          S.E.
-------------+------------------------------------------------------------------
           x |     1.06256     1.062575       -.0000148        .0027541
------------------------------------------------------------------------------
                         b = consistent under Ho and Ha; obtained from regress
          B = inconsistent under Ha, efficient under Ho; obtained from xtmixed

    Test:  Ho:  difference in coefficients not systematic

                  chi2(1) = (b-B)'[(V_b-V_B)^(-1)](b-B)
                          =        0.00
                Prob>chi2 =      0.9957
```

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 mixed with the fixed effects model. The distribution of the estimated p-values is shown in the following graph.

```
par(mfrow=c(1,2),cex=.8)
hm1 <- sapply(simul1,function(x) x[,"h"])
plot(ecdf(hm1["x",]),do.points=FALSE,verticals=TRUE,xlab="$H$",main="test statistic")
plot(function(x) pchisq(x,df=1,lower=TRUE),xlim=range(hm1["x",]),lty=2,add=TRUE)
#
plot(ecdf(hm1["p",]),do.points=FALSE,verticals=TRUE,main="Hausman test",
     xlab="empirical $p$-value")
abline(a=0,b=1,lty=3)
p10<-mean(hm1["p",]<.1)
abline(h=p10,v=.1,lty=2)
```



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 12.4% of all cases a p-value smaller than 10%.

Exercise 3.2 *Use a Hausman test to compare the fixed and the mixed effects model for the dataset* `ex1.csv`.

3.9 Testing random effects

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

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

generally

$$2\cdot(\log(L_{\text{large}})-\log(L_{\text{small}}))\sim\chi^2_k \qquad \text{with } k=df_{\text{large}}-df_{\text{small}}$$

here

$$2\cdot(\log(L_{RE})-\log(L_{OLS}))\sim\chi^2_k \qquad \text{with } k=1$$

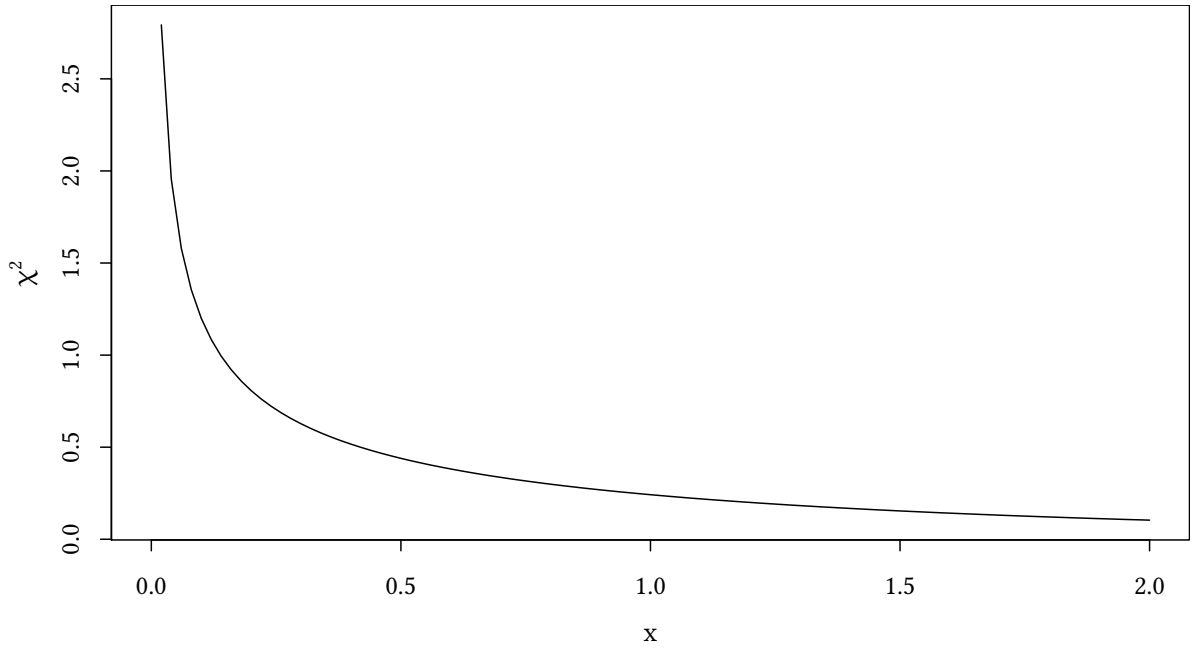
```
(teststat <- 2*(logLik(mixedML)-logLik(ols))[1])
```

```
[1] 673.6798
```

Under the Null 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…

```
xtmixed y x ||
est store ols
lrtest ols mixed
```

```
plot(function(x) dchisq(x,1),0,2,ylab="$\\chi^2$")
```



The p-value of the χ^2-test would be

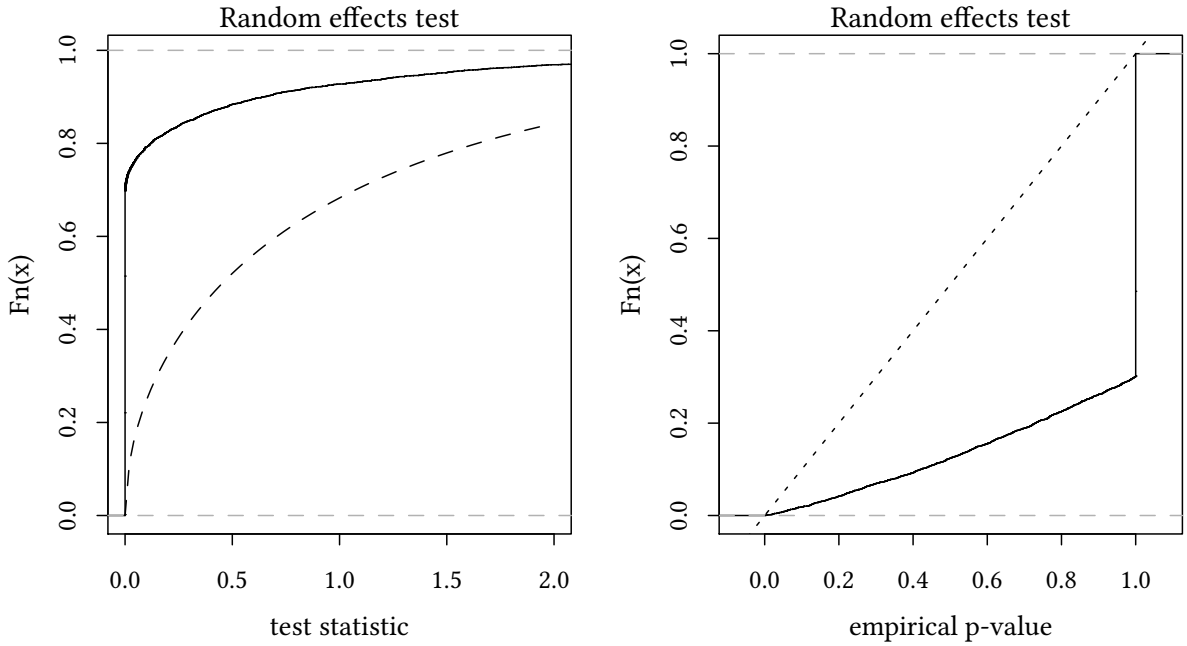
```
pchisq(teststat,1,lower=FALSE)

[1] 1.582576e-148
```

Let us bootstrap its distribution:

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

The bootstrapped distribution differs from the χ^2 distribution:

```
plot(ecdf(dev["LL",]),do.points=FALSE,verticals=TRUE,xlim=c(0,2),
     xlab="test statistic",main="Random effects test")
plot(function(x) pchisq(x,df=1,lower=TRUE),xlim=c(0,2),lty=2,add=TRUE)
#
plot(ecdf(dev["pchisq",]),do.points=FALSE,verticals=TRUE,
     xlab="empirical p-value",main="Random effects test")
abline(a=0,b=1,lty=3)
```



- The assumption of a χ^2 distribution is *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 accept the Null too often.

  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",])

[1] 0
```

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

3.10 Confidence intervals for fixed effects (in a ME model)

To determine confidence intervals for estimated coefficients we have to bootstrap a sample of coefficients.

```
summary(mixed)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i)
   Data: data
```

```
REML criterion at convergence: 1522.1

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.61559 -0.74732  0.02794  0.75531  2.44665

Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 97.861   9.892
 Residual              8.442   2.905
Number of obs: 300, groups:  i, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -3.8219     4.0525  -0.943
x             1.0626     0.5763   1.844

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

```
set.seed(123)
require(boot)
(mixed.boot <- bootMer(mixed,function(.) fixef(.),nsim=bootstrapsize))
```

```


Call:
bootMer(x = mixed, FUN = function(.) fixef(.), nsim = bootstrapsize)


Bootstrap Statistics :
     original     bias    std. error
t1* -3.821877 0.17365102   3.3944679
t2*  1.062575 0.04651233   0.6366886
```

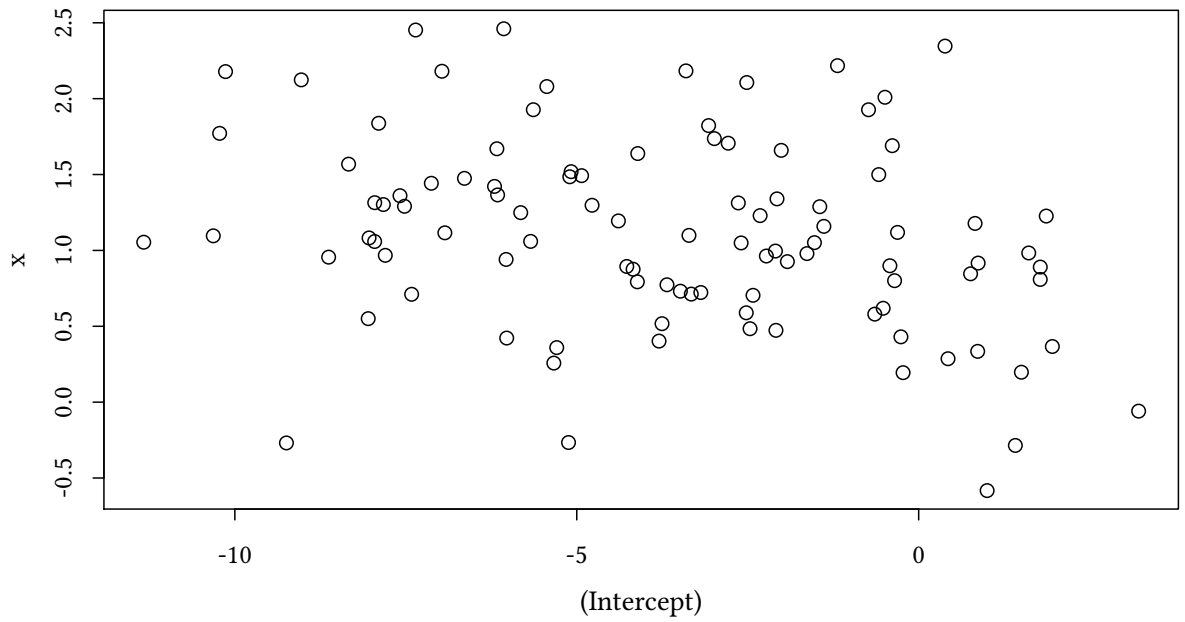
The *t* component of the *boot* object contains all the estimated fixed effects. We can also calculate manually *mean* and *sd*:

```
apply(mixed.boot$t,2,function(x) c(mean=mean(x),sd=sd(x)))

     (Intercept)         x
mean   -3.648226 1.1090875
sd      3.394468 0.6366886
```

$$\left(\texttt{bias} = \bar{\theta}_{BS} - \hat{\theta}\right)$$

```
boot.ci(mixed.boot,index=2,type=c("norm","basic","perc"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = mixed.boot, type = c("norm", "basic", "perc"),
    index = 2)

Intervals :
Level      Normal              Basic              Percentile
95%   (-0.232,  2.264 )   (-0.267,  2.401 )   (-0.276,  2.392 )
Calculations and Intervals on Original Scale
Some basic intervals may be unstable
Some percentile intervals may be unstable
```

```
plot(as.data.frame(mixed.boot))
```



Exercise 3.3 *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 mixed effects model: Is the group specific effect significant?*
- *Use a Hausman test to compare the fixed and the mixed effects model.*

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

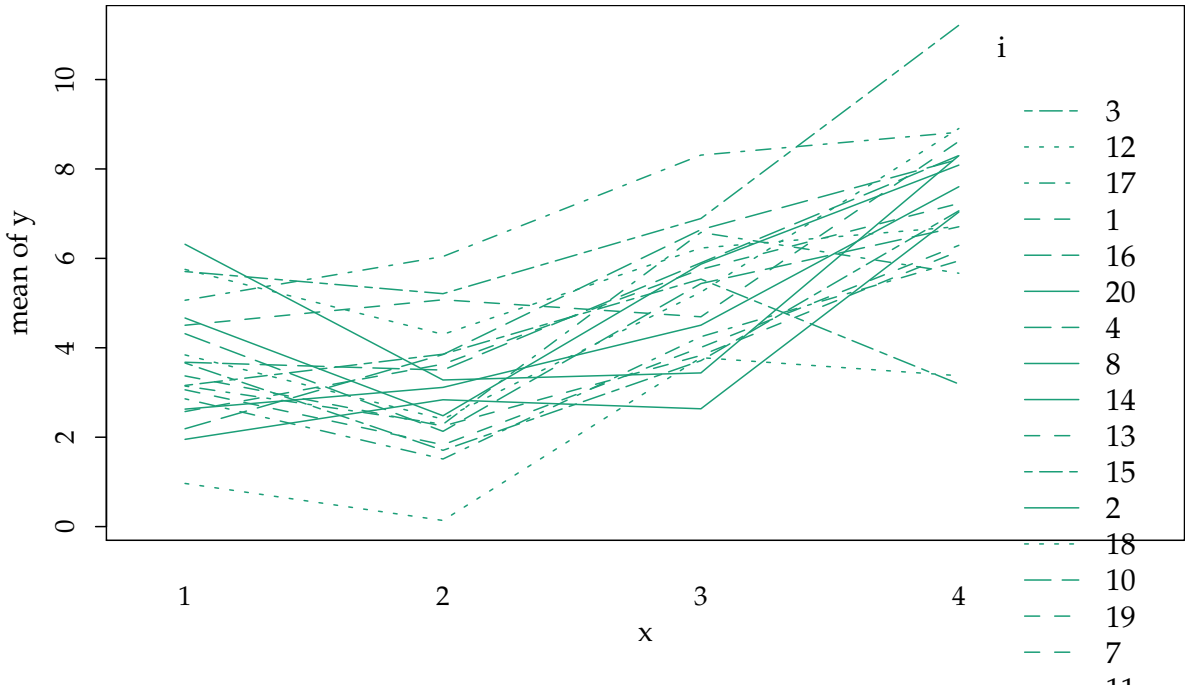
```
with(dataM,xtable(table(x,i)))
```

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

We have to keep in mind that *x* and *i* are factors, i.e. there are only three variables, not 4 + 20 dummies.

```
str(dataM)

'data.frame': 80 obs. of  3 variables:
 $ y: num  4.5 5.07 4.69 8.62 1.95 ...
 $ x: Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
 $ i: Factor w/ 20 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
```

```
with(dataM,interaction.plot(x,i,y))
```



```
write.csv(dataM,file="csv/dataM.csv",row.names=FALSE)
```

Stata can do something similar, although it is not trivial to apply (systematically) different linestyles for different groups.

```
clear
insheet using csv/dataM.csv
sort i x
graph twoway line y x
```

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:

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

with

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

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

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

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

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

4.1 Estimation with different contrast matrices

4.1.1 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.

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)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i)
   Data: dataM
REML criterion at convergence: 270.0249
Random effects:
 Groups   Name        Std.Dev.
 i        (Intercept) 1.077
 Residual             1.082
Number of obs: 80, groups:  i, 20
Fixed Effects:
(Intercept)           x2           x3           x4
      3.670       -0.598        1.492        3.502
```

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:

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

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

As an alternative, we can change the reference category:

```
lmer(y ~ relevel(x,2) + (1|i),data=dataM)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ relevel(x, 2) + (1 | i)
   Data: dataM
REML criterion at convergence: 270.0249
Random effects:
 Groups   Name        Std.Dev.
 i        (Intercept) 1.077
 Residual             1.082
Number of obs: 80, groups:  i, 20
Fixed Effects:
   (Intercept)  relevel(x, 2)1  relevel(x, 2)3  relevel(x, 2)4
         3.072           0.598           2.090           4.100
```

First category as a reference can be done in Stata, too:

```
xtmixed y i.x || i:

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -135.01247
Iteration 1:   log restricted-likelihood = -135.01247

Computing standard errors:

Mixed-effects REML regression                   Number of obs      =        80
Group variable: i                               Number of groups   =        20

                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4


                                                Wald chi2(3)       =    171.28
Log restricted-likelihood = -135.01247          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |
          2  |  -.5979942   .3420046    -1.75   0.080    -1.268311    .0723226
          3  |   1.492447   .3420046     4.36   0.000     .8221299    2.162763
```

```
          4  |   3.502027   .3420046    10.24   0.000      2.83171    4.172344
             |
       _cons |   3.669742   .3412924    10.75   0.000     3.000821    4.338663
------------------------------------------------------------------------------


------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
i: Identity                  |
                  sd(_cons) |   1.077004   .2202311      .7213727     1.60796
-----------------------------+------------------------------------------------
               sd(Residual) |   1.081514    .101293      .9001391    1.299434
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    21.91 Prob >= chibar2 = 0.0000
```

Stata can use a different reference category, too:

```
xtmixed y b2.x || i:
```

```
Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -135.01247
Iteration 1:   log restricted-likelihood = -135.01247

Computing standard errors:

Mixed-effects REML regression                   Number of obs      =        80
Group variable: i                               Number of groups   =        20

                                                Obs per group: min =         4
                                                               avg =       4.0
                                                               max =         4


                                                Wald chi2(3)       =    171.28
Log restricted-likelihood = -135.01247          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |
          1  |   .5979942   .3420046     1.75   0.080    -.0723226    1.268311
          3  |   2.090441   .3420046     6.11   0.000     1.420124    2.760758
          4  |   4.100021   .3420046    11.99   0.000     3.429705    4.770338
             |
       _cons |   3.071748   .3412924     9.00   0.000     2.402827    3.740668
------------------------------------------------------------------------------


------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
```

```
i: Identity                  |
                  sd(_cons) |   1.077004   .2202311      .7213727     1.60796
-----------------------------+------------------------------------------------
               sd(Residual) |   1.081514    .101293      .9001391    1.299434
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    21.91 Prob >= chibar2 = 0.0000
```

4.1.2 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 ~ C(x,contr.sum),data1)
```

```
as.data.frame(mm)

  (Intercept) C(x, contr.sum)1 C(x, contr.sum)2 C(x, contr.sum)3
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 treatment and the mean.
- Coefficient of *x2*: difference between the second treatment and the mean.
- Coefficient of *x3*: difference between the third treatment and the mean.

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 ~ C(x,sum) + (1|i),data=dataM)
print(s.lmer,correlation=FALSE)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ C(x, sum) + (1 | i)
   Data: dataM
REML criterion at convergence: 272.7975
Random effects:
```

```
 Groups   Name        Std.Dev.
 i        (Intercept) 1.077
 Residual             1.082
Number of obs: 80, groups:  i, 20
Fixed Effects:
(Intercept)   C(x, sum)1   C(x, sum)2   C(x, sum)3
     4.7689      -1.0991      -1.6971       0.3933
```

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:

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

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

Sum contrasts can be done in Stata, too:

```
desmat x,dev(4)
list _x* if i==1
```

```
    +--------------------+
     | _x_1   _x_2   _x_3 |
     |--------------------|
  1. |    1      0      0 |
  2. |    0      1      0 |
  3. |    0      0      1 |
  4. |   -1     -1     -1 |
     +--------------------+
```

```
xtmixed y _x* || i:
```

```
Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -136.39877
Iteration 1:   log restricted-likelihood = -136.39877

Computing standard errors:

Mixed-effects REML regression                   Number of obs      =         80
Group variable: i                               Number of groups   =         20

                                                Obs per group: min =          4
                                                               avg =        4.0
                                                               max =          4


                                                Wald chi2(3)       =     171.28
```

```
Log restricted-likelihood = -136.39877          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         _x_1 |   -1.09912   .2094342    -5.25   0.000    -1.509603   -.6886364
         _x_2 |  -1.697114   .2094342    -8.10   0.000    -2.107598   -1.286631
         _x_3 |   .3933267   .2094342     1.88   0.060    -.0171568    .8038102
        _cons |   4.768862   .2694769    17.70   0.000     4.240697    5.297027
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
i: Identity                  |
                   sd(_cons) |   1.077004   .2202311      .7213726     1.60796
-----------------------------+------------------------------------------------
                sd(Residual) |   1.081514    .101293      .9001391    1.299434
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    21.91 Prob >= chibar2 = 0.0000
```

4.1.3 Helmert contrasts

Helmert contrasts are another way to show mean effects and differences between treatments.

```
mm <- model.matrix(y ~ C(x,contr.helmert),data1)
```

```
as.data.frame(mm)

  (Intercept) C(x, contr.helmert)1 C(x, contr.helmert)2 C(x, contr.helmert)3
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.

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 ~ C(x,contr.helmert) + (1|i) ,data=dataM)
print(h.lmer,correlation=FALSE)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ C(x, contr.helmert) + (1 | i)
   Data: dataM
REML criterion at convergence: 276.3811
Random effects:
 Groups   Name        Std.Dev.
 i        (Intercept) 1.077
 Residual             1.082
Number of obs: 80, groups:  i, 20
Fixed Effects:
         (Intercept)  C(x, contr.helmert)1  C(x, contr.helmert)2
              4.7689               -0.2990                0.5971
C(x, contr.helmert)3
              0.8010
```

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

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

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

Stata scales Helmert contrasts in a different way (we need the *desmat* package, which is not part of the standard installation).

```
desmat x,hel(b)
list _x* if i==1
```

```
    +-------------------------+
     | _x_1        _x_2   _x_3 |
     |-------------------------|
  1. |  .75           0      0 |
  2. | -.25    .6666667      0 |
  3. | -.25   -.3333333     .5 |
  4. | -.25   -.3333333    -.5 |
     +-------------------------+
```

```
xtmixed y _x* || i:
```

```
Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -135.01247
Iteration 1:   log restricted-likelihood = -135.01247

Computing standard errors:
```

```
Mixed-effects REML regression                   Number of obs      =         80
Group variable: i                               Number of groups   =         20

                                                Obs per group: min =          4
                                                               avg =        4.0
                                                               max =          4


                                                Wald chi2(3)       =     171.28
Log restricted-likelihood = -135.01247          Prob > chi2        =     0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        _x_1 |  -.5979942   .3420046    -1.75   0.080    -1.268311    .0723226
        _x_2 |   1.791444   .2961847     6.05   0.000     1.210932    2.371955
        _x_3 |   3.203876   .2792456    11.47   0.000     2.656565    3.751188
       _cons |   4.768862   .2694769    17.70   0.000     4.240697    5.297027
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
i: Identity                  |
                   sd(_cons) |   1.077004   .2202311      .7213727     1.60796
-----------------------------+------------------------------------------------
                sd(Residual) |   1.081514    .101293      .9001391    1.299434
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    21.91 Prob >= chibar2 = 0.0000
```

4.1.4 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 ['lmerMod']
Formula: y ~ x - 1 + (1 | i)
   Data: dataM
REML criterion at convergence: 270.0249
Random effects:
 Groups   Name        Std.Dev.
 i        (Intercept) 1.077
 Residual             1.082
Number of obs: 80, groups:  i, 20
Fixed Effects:
   x1     x2     x3     x4
3.670  3.072  5.162  7.172
```

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

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

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

Cell means contrasts can be done in Stata, too:

```
xi i.x,noomit
xtmixed y _Ix* ,noconstant|| i:
```

```
Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -135.01247
Iteration 1:   log restricted-likelihood = -135.01247

Computing standard errors:

Mixed-effects REML regression                   Number of obs      =         80
Group variable: i                               Number of groups   =         20

                                                Obs per group: min =          4
                                                               avg =        4.0
                                                               max =          4


                                                Wald chi2(4)       =     484.45
Log restricted-likelihood = -135.01247          Prob > chi2        =     0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _Ix_1 |   3.669742   .3412924    10.75   0.000     3.000821    4.338663
```

```
       _Ix_2 |   3.071748   .3412924     9.00   0.000     2.402827    3.740668
       _Ix_3 |   5.162188   .3412924    15.13   0.000     4.493268    5.831109
       _Ix_4 |   7.171769   .3412924    21.01   0.000     6.502848     7.84069
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
i: Identity                  |
                   sd(_cons) |   1.077004   .2202311      .7213727     1.60796
-----------------------------+------------------------------------------------
                sd(Residual) |   1.081514    .101293      .9001391    1.299434
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    21.91 Prob >= chibar2 = 0.0000
```

4.2 Which statistics are affected by the type of contrasts?

4.2.1 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 entire factor.

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

```
            reference       sum   helmert cellmeans
(Intercept) 10.752487 17.696738 17.696738 10.752487
x2          -1.748497 -5.248044 -1.748497  9.000341
x3           4.363820 -8.103328  6.048400 15.125415
x4          10.239706  1.878044 11.473327 21.013564
```

4.2.2 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 566.65
Mean Sq 66.77953  66.77953 66.77953 141.6625
F value 57.09254  57.09254 57.09254 121.113
```

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.

4.2.3 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.REML       sum.REML   helmert.REML cellmeans.REML
      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
df.resid   74.0000   74.0000   74.0000   74.0000
```

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

Exercise 4.1 *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.*

5 Testing fixed effects

5.1 Anova

```
summary(r.lmer)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i)
   Data: dataM

REML criterion at convergence: 270

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.06539 -0.47592  0.03843  0.56479  2.02980

Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.16     1.077
 Residual             1.17     1.082
Number of obs: 80, groups:  i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)   3.6697     0.3413  10.752
x2           -0.5980     0.3420  -1.748
x3            1.4924     0.3420   4.364
x4            3.5020     0.3420  10.240

Correlation of Fixed Effects:
   (Intr) x2     x3
x2 -0.501
x3 -0.501  0.500
x4 -0.501  0.500  0.500
```

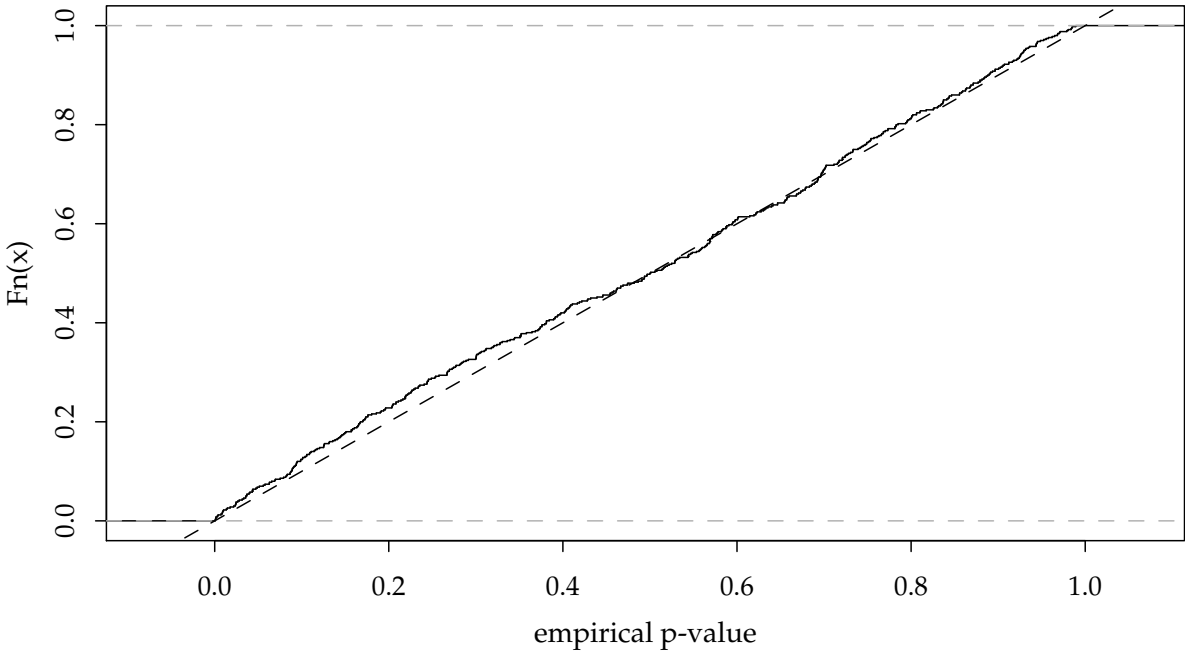
To test a fixed effect we can not use REML as an estimation procedure.

```
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 deviance  Chisq Chi Df Pr(>Chisq)
r.lmerMLsmall  3 356.77 363.92 -175.38   350.77
r.lmerML       6 279.52 293.81 -133.76   267.52 83.251      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])
  })
```

```
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:

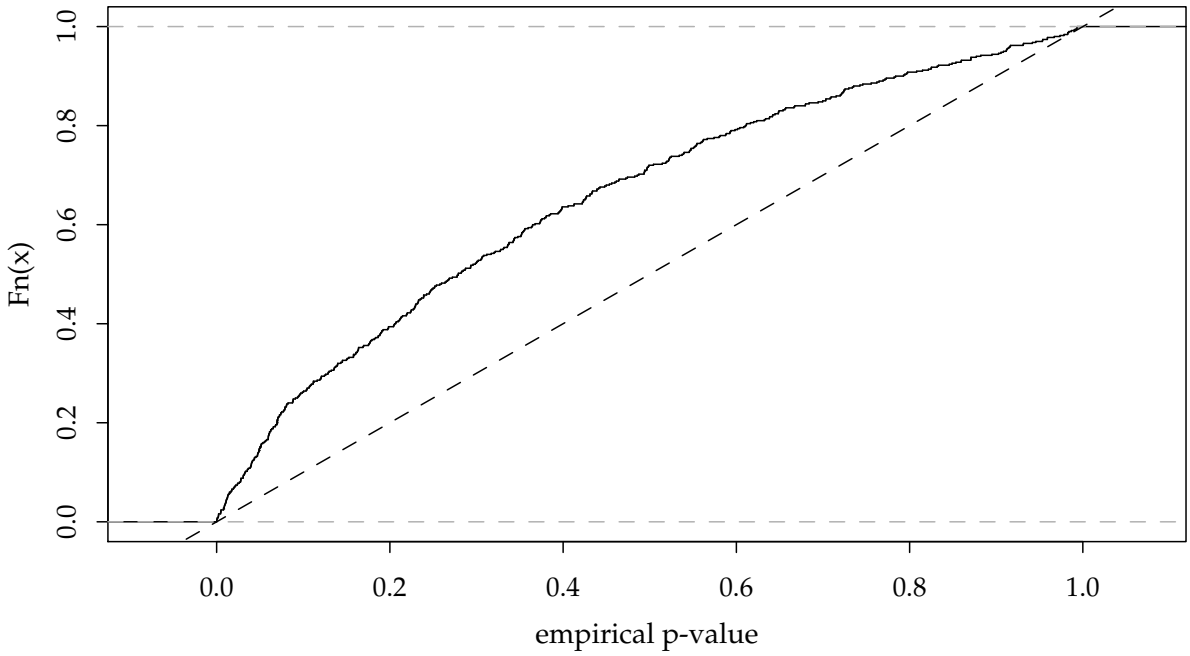
[1] Littel, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), SAS System for Mixed Models, SAS Institute (Data Set 1.5.1)

```
l.small <- lmer(response ~ 1 + (1|Block),data=PBIB,REML=FALSE)
l.large <- lmer(response ~ 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 ~ Treatment + (1 | Block)
        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
l.small  3 52.152 58.435 -23.076   46.152
l.large 17 56.571 92.174 -11.285   22.571 23.581     14    0.05144 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now we simulate 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])
})
```

```
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] 0.156
```

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",]<.05) * 100

[1] 14.8
```

5.2 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)
```

```
summary(random)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x1 + x2 + (1 | i)
   Data: data3

REML criterion at convergence: 1290.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.5297 -0.6688 -0.1560  0.5998  3.2360

Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 224.6    14.987
 Residual               8.1     2.846
Number of obs: 240, groups:  i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.19400    3.39003   0.352
x1           1.69234    0.64175   2.637
x2           0.05455    0.66086   0.082

Correlation of Fixed Effects:
   (Intr) x1
x1 -0.104
x2 -0.103  0.070
```

bootMer generates a boostrap sample for the parameters.

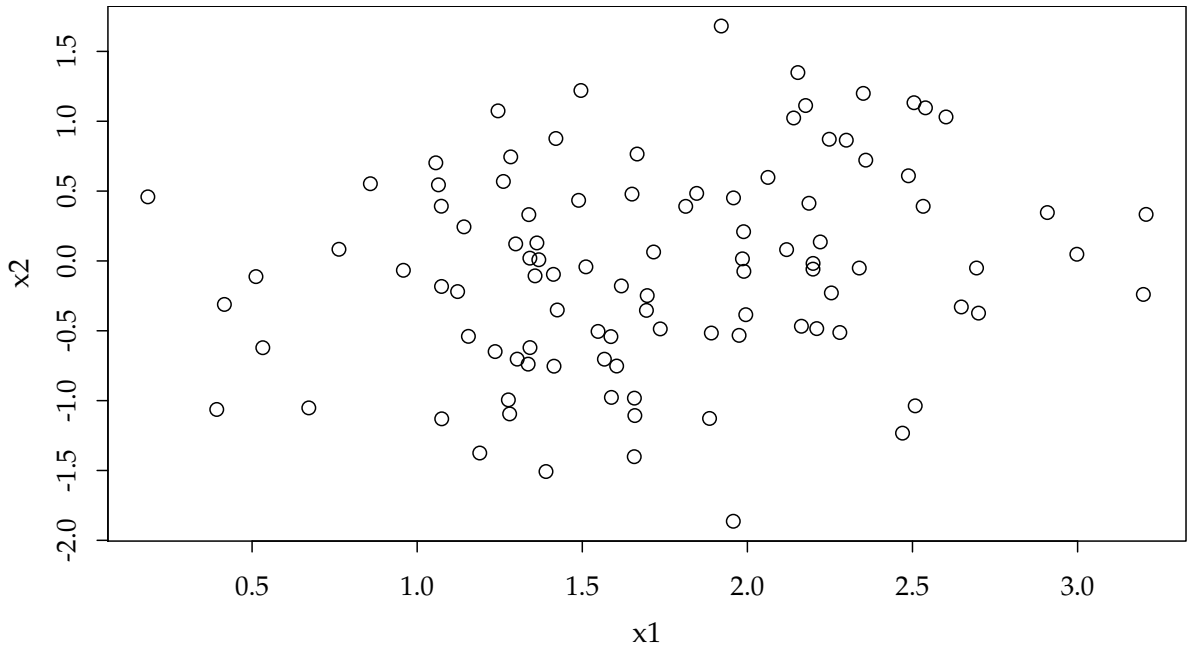
```
require(boot)
(random.boot <- bootMer(random,function(.) fixef(.),nsim=bootstrapsize))
```

```
Call:
bootMer(x = random, FUN = function(.) fixef(.), nsim = bootstrapsize)


Bootstrap Statistics :
      original      bias    std. error
t1* 1.19400144  0.45024665   2.9558641
t2* 1.69234221  0.02995304   0.6250481
t3* 0.05454998 -0.11202493   0.7256122
```

```
sqrt(diag(vcov(random)))
```

```
[1] 3.3900294 0.6417532 0.6608579
```

```
plot(as.data.frame(random.boot)[,2:3])
```




```
boot.ci(random.boot,index=2,type=c("norm","basic","perc"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = random.boot, type = c("norm", "basic", "perc"),
    index = 2)
```

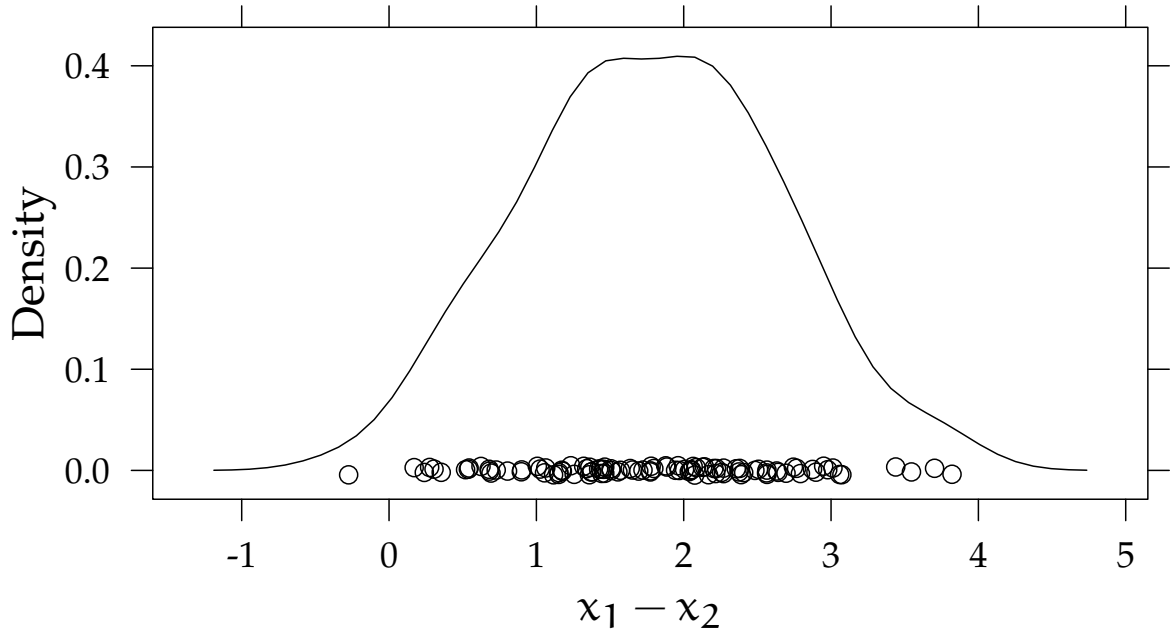
```
Intervals :
Level      Normal              Basic              Percentile
95%   ( 0.437,  2.887 )   ( 0.300,  2.979 )   ( 0.406,  3.085 )
Calculations and Intervals on Original Scale
Some basic intervals may be unstable
Some percentile intervals may be unstable
```

```
boot.ci(random.boot,index=3,type=c("norm","basic","perc"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = random.boot, type = c("norm", "basic", "perc"),
    index = 3)

Intervals :
Level      Normal              Basic              Percentile
95%   (-1.2556,  1.5887 )   (-1.1667,  1.5564 )   (-1.4473,  1.2758 )
Calculations and Intervals on Original Scale
Some basic intervals may be unstable
Some percentile intervals may be unstable
```

Functions of coefficients We can bootstrap also (linear and non-linear) functions of coefficients. Assume that in the above example we are interested in $\beta_{x_1} - \beta_{x_2}$.

```
(x12diff.boot <- bootMer(random,function(.) fixef(.) %*% c(0,1,-1),
                         nsim=bootstrapsize))



Call:
bootMer(x = random, FUN = function(.) fixef(.) %*% c(0, 1, -1),
    nsim = bootstrapsize)


Bootstrap Statistics :
    original   bias    std. error
t1* 1.637792 0.141978   0.8508993
```

```
densityplot(x12diff.boot$t,xlab="$x_1-x_2$")
```



Again, we can calculate confidence intervals.

```
boot.ci(x12diff.boot,type=c("norm","basic","perc"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 100 bootstrap replicates

CALL :
boot.ci(boot.out = x12diff.boot, type = c("norm", "basic", "perc"))

Intervals :
Level      Normal              Basic              Percentile
95%   (-0.172,  3.164 )   (-0.338,  3.066 )   ( 0.209,  3.614 )
Calculations and Intervals on Original Scale
Some basic intervals may be unstable
Some percentile intervals may be unstable
```

5.3 Testing random effects

See section 3.9 above.

Exercise 5.1 *You look again at the* `ex3.csv` *(see exercise 4.1) 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:*

$$y = \gamma + \sum_{g=1}^{G} d_g \beta_g x + u\,.$$

6 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("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
 [ reached getOption("max.print") -- omitted 12 rows ]
```

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)
summary(r.mer)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ treat - 1 + (1 | group) + (1 | pid)
   Data: ex4

REML criterion at convergence: 1619.3

Scaled residuals:
```

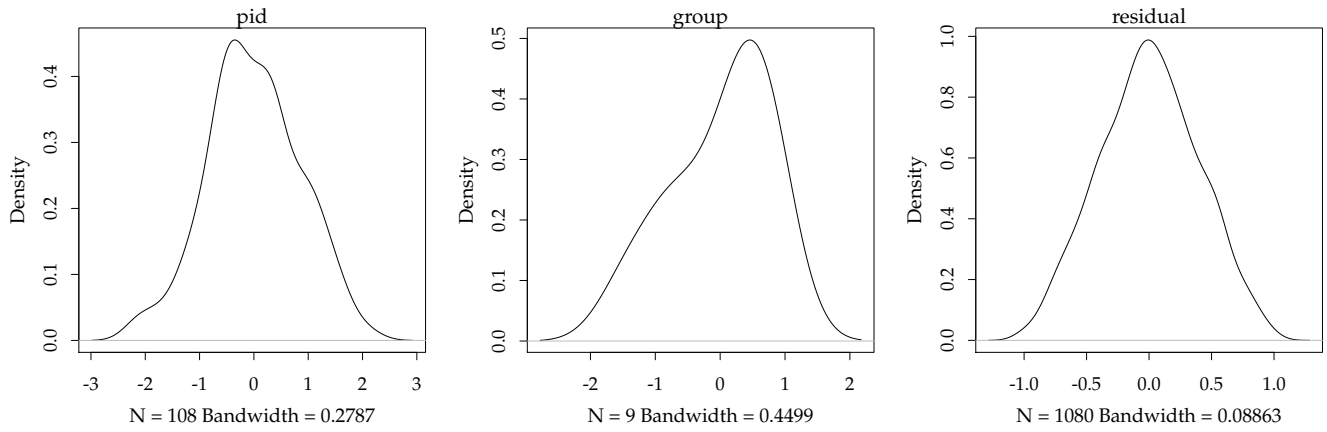
```
    Min      1Q  Median      3Q     Max
-2.4295 -0.6607 -0.0107  0.6683  2.4320

Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 0.8002   0.8946
 group    (Intercept) 0.8654   0.9303
 Residual             0.1756   0.4190
Number of obs: 1080, groups:  pid, 108; group, 9

Fixed effects:
       Estimate Std. Error t value
treatA  12.1639     0.5578  21.806
treatB   4.4586     0.5578   7.993
treatC   8.6931     0.5578  15.584

Correlation of Fixed Effects:
       treatA treatB
treatB 0.000
treatC 0.000  0.000
```

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] -0.7923 0.1581 0.6013 0.0543 0.4339 ...
 - attr(*, "class")= chr "ranef.mer"

str(residuals(r.mer))

 Named num [1:1080] 0.339 0.439 0.339 0.239 0.539 ...
 - attr(*, "names")= chr [1:1080] "1" "2" "3" "4" ...
```

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

```
plot(density(unlist(ranef(r.mer)[["pid"]])),main="pid")
plot(density(unlist(ranef(r.mer)[["group"]])),main="group")
plot(density(residuals(r.mer)),main="residual")
```



The same can be done in Stata:

```
clear
insheet using csv/ex4.csv
encode treat,gen(treatn)
xi i.treatn,noomit
xtmixed y _Itreatn*,noconstant || group:  || pid:
```

```
Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood =  -809.6717
Iteration 1:   log restricted-likelihood =  -809.6717

Computing standard errors:

Mixed-effects REML regression                   Number of obs      =      1080

------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+-------------------------------------------
          group |        9        120      120.0        120
            pid |      108         10       10.0         10
------------------------------------------------------------

                                                Wald chi2(3)       =    782.21
Log restricted-likelihood =  -809.6717          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  _Itreatn_1 |   12.16389   .5578344    21.81   0.000     11.07055    13.25722
  _Itreatn_2 |   4.458611   .5578344     7.99   0.000     3.365276    5.551946
  _Itreatn_3 |   8.693056   .5578344    15.58   0.000      7.59972    9.786391
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
```

```
------------------------------+------------------------------------------------
group: Identity               |
                   sd(_cons) |   .9302627    .289739      .5052321    1.712854
------------------------------+------------------------------------------------
pid: Identity                 |
                   sd(_cons) |   .8945568   .0649696      .7758666    1.031404
------------------------------+------------------------------------------------
                sd(Residual) |   .4189984   .0095031      .4007806    .4380443
-------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =  1915.72   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

Exercise 6.1 *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 magnitude of the effect the same for all treatments?*

7 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.

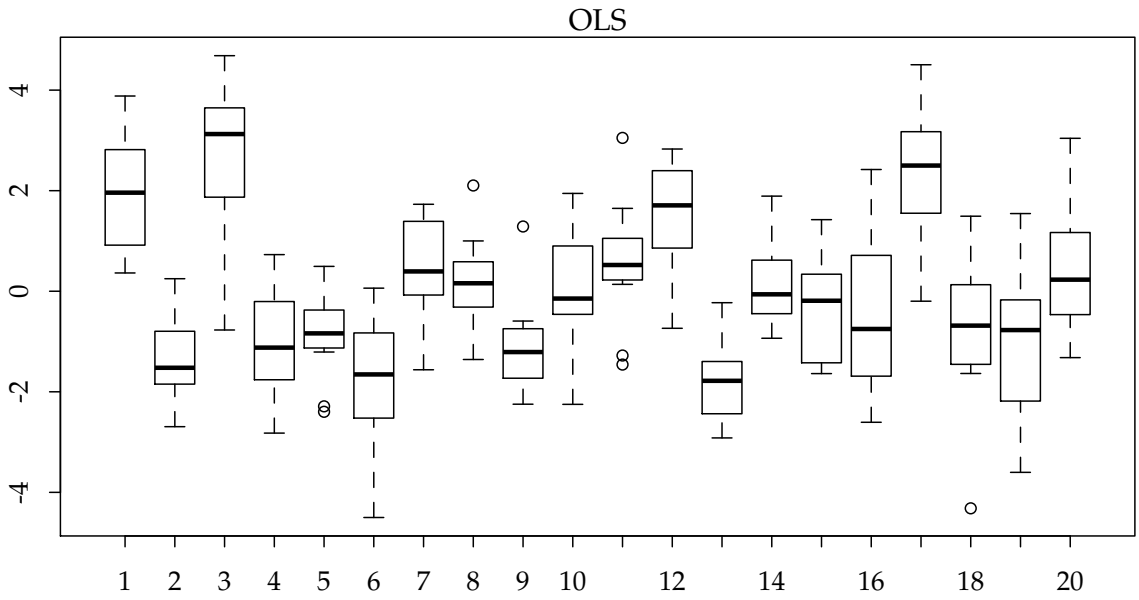
```
with(dataMR,table(x,i))
```

```
   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
 [ reached getOption("max.print") -- omitted 1 row ]
```

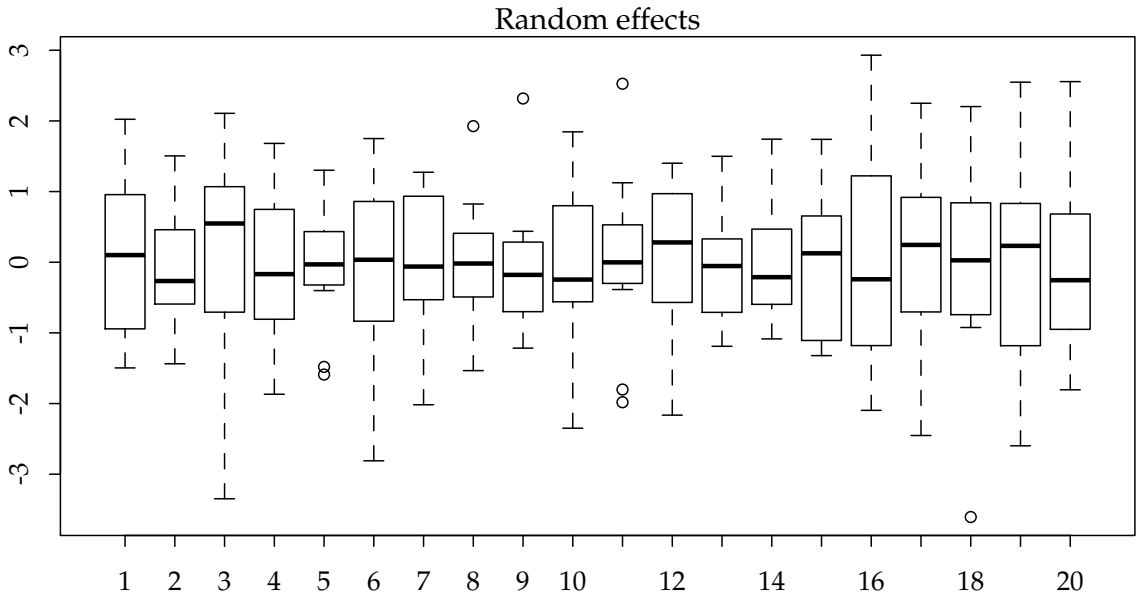
7.1 A model with one random effect

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")
```



Should we use a random effect?
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 ['lmerMod']
Formula: y ~ x + (1 | i)
   Data: dataMR

REML criterion at convergence: 809.5

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.10325 -0.57348 -0.01521  0.60176  2.52213

Random effects:
 Groups   Name        Variance Std.Dev.
 i        (Intercept) 1.68     1.296
 Residual             1.35     1.162
Number of obs: 240, groups:  i, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.59859    0.31758   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
```

This can also be done in Stata:

```
clear
insheet using csv/dataMR.csv
xtmixed y i.x || i:
```

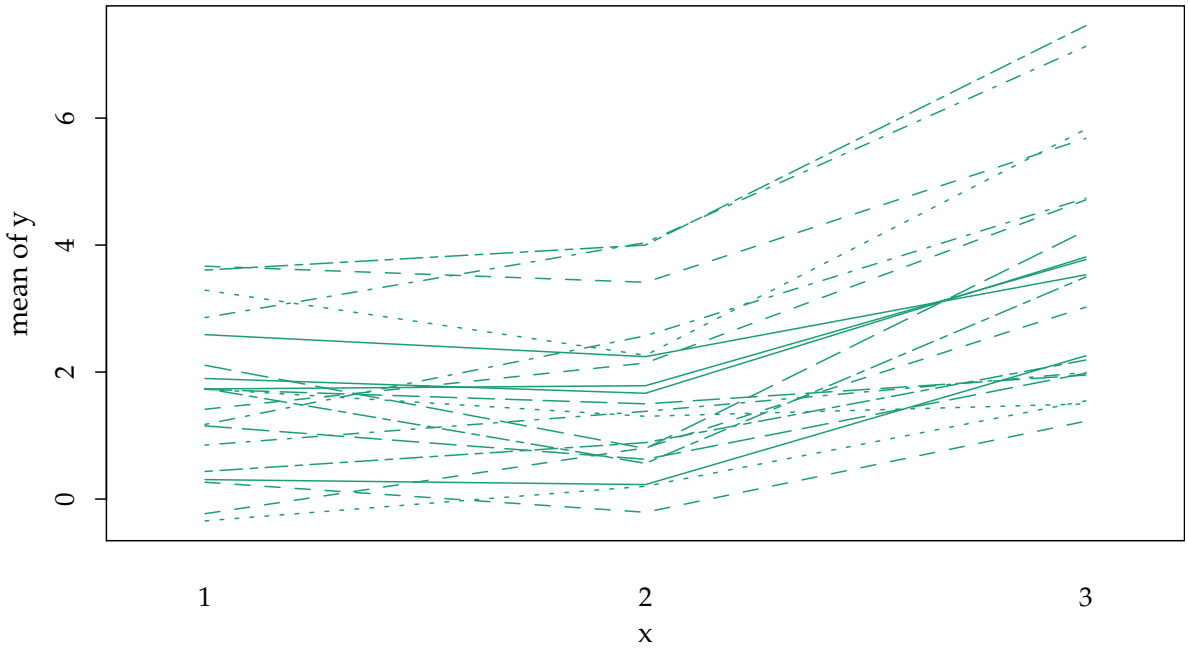
```
Mixed-effects REML regression                   Number of obs      =       240
Group variable: i                               Number of groups   =        20

                                                Obs per group: min =        12
                                                               avg =      12.0
                                                               max =        12


                                                Wald chi2(2)       =    157.71
```

```
Log restricted-likelihood = -404.73334          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |
          2  |   .0116518   .1837213     0.06   0.949    -.3484354     .3717389
          3  |   2.003926   .1837213    10.91   0.000     1.643838     2.364013
             |
       _cons |   1.598592   .3175831     5.03   0.000     .9761404     2.221043
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
i: Identity                  |
                  sd(_cons) |   1.296011   .2243625      .9231041    1.819562
-----------------------------+------------------------------------------------
               sd(Residual) |   1.161956   .0556476      1.057851    1.276306
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =   133.93 Prob >= chibar2 = 0.0000
```

7.2 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?

```
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))

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i) + (1 | i:x)
   Data: dataMR
REML criterion at convergence: 805.0675
Random effects:
 Groups   Name        Std.Dev.
 i:x      (Intercept) 0.4448
 i        (Intercept) 1.2748
 Residual             1.1010
Number of obs: 240, groups:  i:x, 60; i, 20
Fixed Effects:
(Intercept)           x2           x3
    1.59859      0.01165      2.00393
```

An equivalent (more compact) notation is the following:

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

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ x + (1 | i/x)
   Data: dataMR
REML criterion at convergence: 805.0675
Random effects:
 Groups   Name        Std.Dev.
 x:i      (Intercept) 0.4448
 i        (Intercept) 1.2748
 Residual             1.1010
Number of obs: 240, groups:  x:i, 60; i, 20
Fixed Effects:
(Intercept)           x2           x3
    1.59859      0.01165      2.00393
```

We could now use *anova* to compare the two models, although we can not really be 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/x)
        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
```

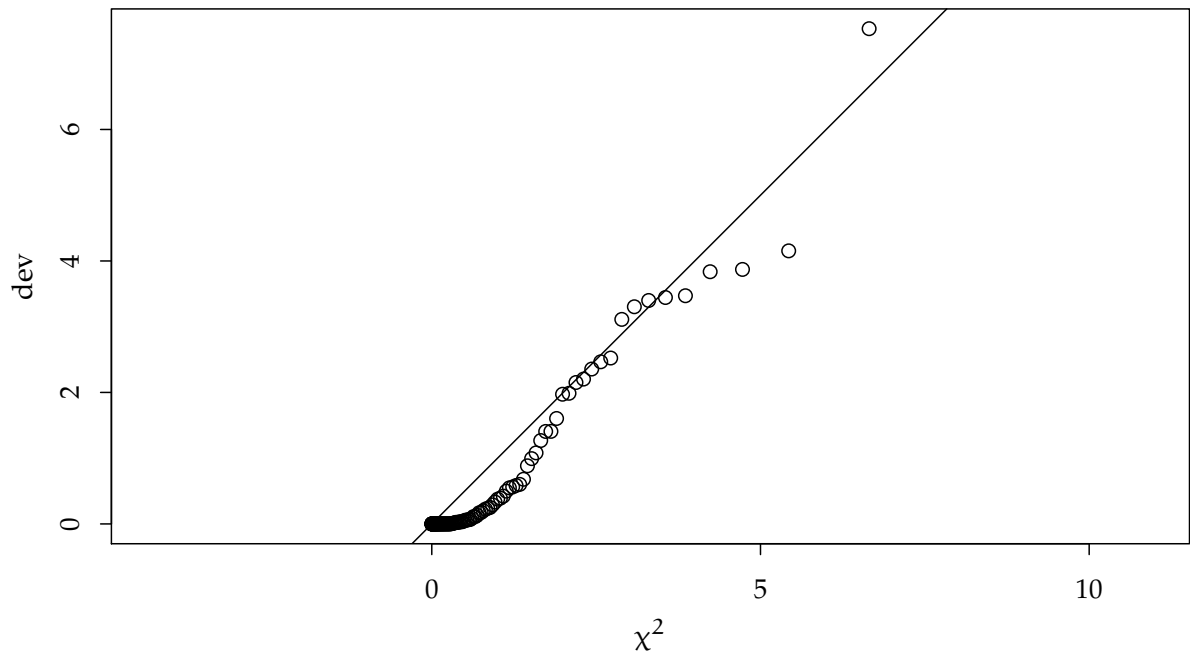
```
m1.lmer  5 815.47 832.87 -402.73   805.47
m2.lmer  6 813.82 834.70 -400.91   801.82 3.6524      1    0.05599 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
set.seed(125)
```

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

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

p= 0.04
```



I expect that this situation can be handled in Stata, too, but I do not know an elegant solution. Please let me know if you have an idea.

7.3 Interactions and replications

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

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

7.4 More random interactions

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

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:

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

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

```
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 ['lmerMod']
Formula: y ~ x + (x - 1 | i)
   Data: dataMR
REML criterion at convergence: 789.6428
Random effects:
 Groups   Name Std.Dev. Corr
 i        x1   1.066
          x2   1.112    0.97
          x3   1.784    0.92 0.99
 Residual      1.089
Number of obs: 240, groups:  i, 20
Fixed Effects:
(Intercept)           x2           x3
    1.59859      0.01165      2.00393
```

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/x)
m3.lmer: y ~ x + (x - 1 | i)
        Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
m1.lmer  5 815.47 832.87 -402.73   805.47
```

```
m2.lmer  6 813.82 834.70 -400.91   801.82  3.6524      1    0.05599 .
m3.lmer 10 805.82 840.62 -392.91   785.82 15.9991      4    0.00302 **
---
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 introducing 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.

I expect that this situation can be handled in Stata, too, but I do not know an elegant solution. Please let me know if you have an idea.

Exercise 7.1 *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.*
4. *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?*

```
I <- 20
T <- 12
K <- 3
```

8 Random effects for more than a constant

8.1 Models we studied so far

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

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

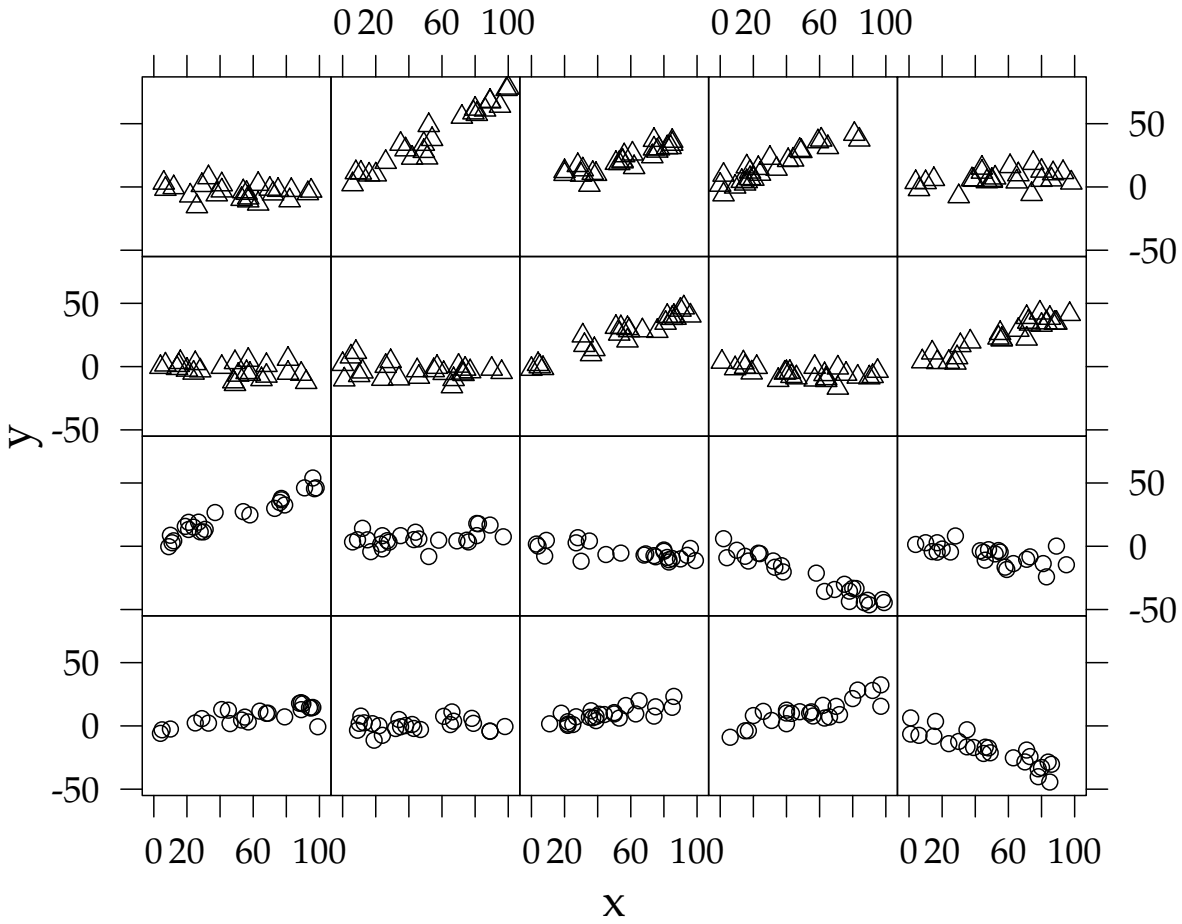
What, if $\mathbf{X}$ and $\mathbf{Z}$ also contain a linear variable, like a valuation, a cost, or time expired during the experiment?

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

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

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 ○).

```
xyplot(y ~ x | i,group=a,data=dataII,strip=FALSE,layout=c(5,4))
```

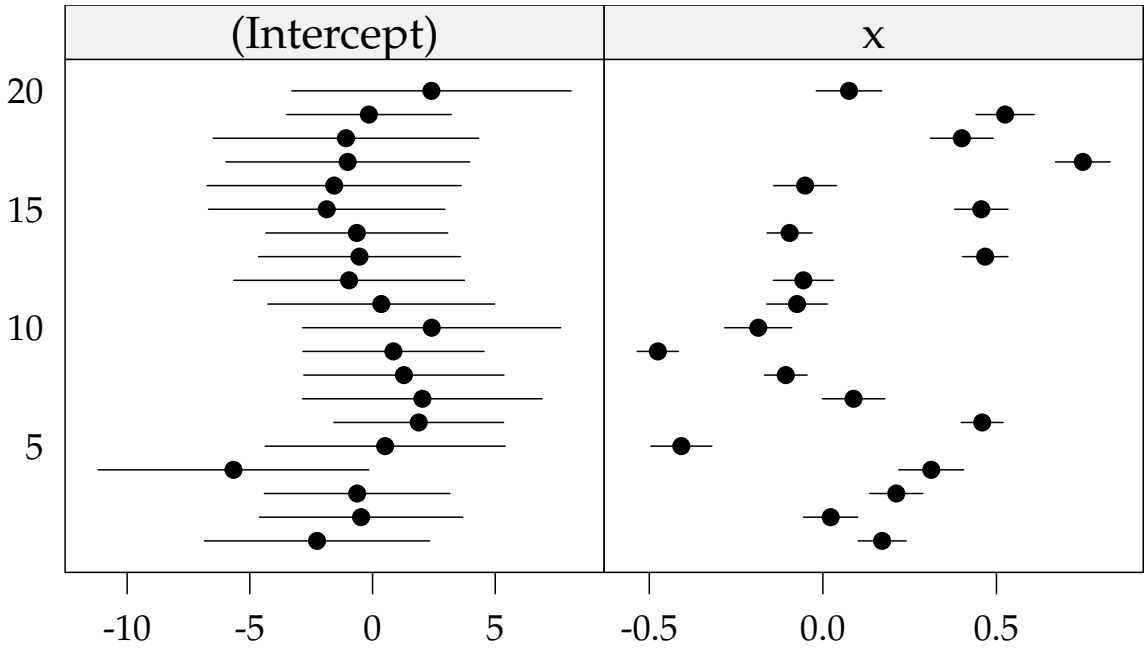


The figure suggests some systematic differences among participants i. Let us first estimate one OLS model for each participant.

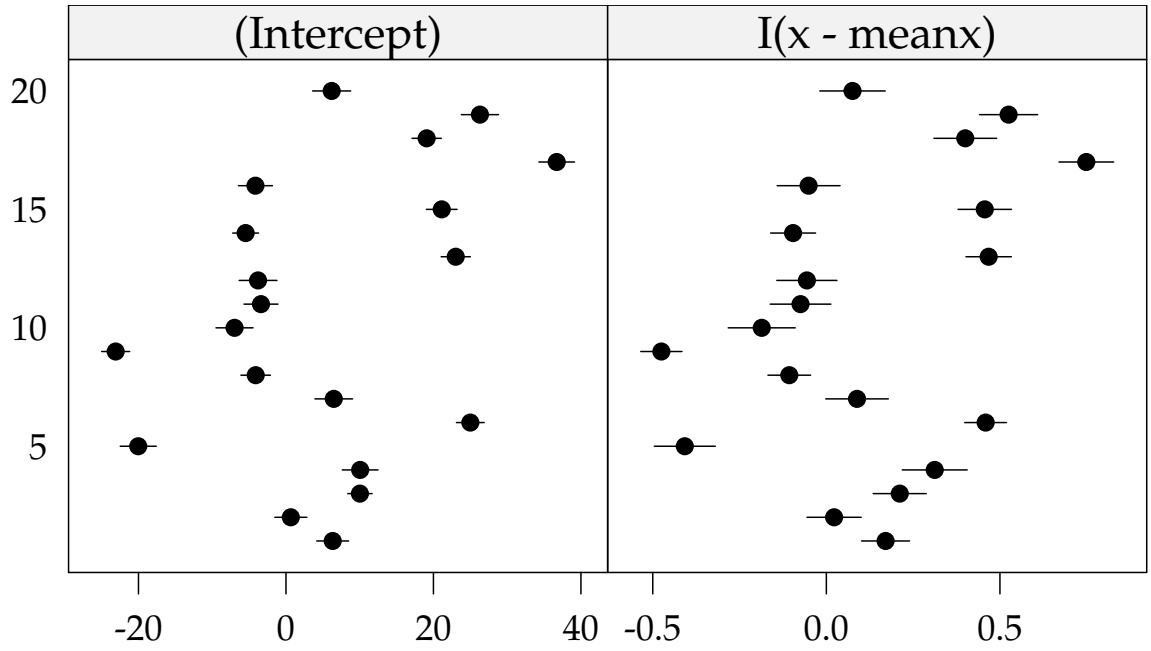
```
ols.list <- lmList(y ~ x | i,data=dataII)
aM <- with(dataII,aggregate(a,list(i),median))[,2]
```

The following two graphs shows estimated confidence intervals for the coefficients. The first shows intervals for the above regression, the second shows intervals for a regression where x enters "demeaned".

```
library(nlme)
library(plyr)
list2int<-function(l) rbind.fill(
    lapply(as.list(1:length(l)),function(i) {
    x<-l[[i]]
    cbind(data.frame(confint(x)),est=coef(x),i=i,n=names(coef(x)))}))
segplot( i ~ X2.5.. + X97.5.. | n,center=est,data=list2int(ols.list),scale=list(x="free"))
```




```
meanx <- mean(dataII$x)
ols2.list <- lmList(y ~ I(x - meanx) | i,data=dataII)
segplot( i ~ X2.5.. + X97.5.. | n,center=est,data=list2int(ols2.list),scale=list(x="free"))
detach(package:nlme)
```



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 ['lmerMod']
Formula: y ~ x * a + (1 | i)
   Data: dataII
REML criterion at convergence: 3640.566
Random effects:
 Groups   Name        Std.Dev.
 i        (Intercept) 14.892
 Residual              9.898
Number of obs: 480, groups:  i, 20
Fixed Effects:
(Intercept)            x        aTRUE      x:aTRUE
   0.578949    -0.005281    -1.107120     0.250552
```

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 ['lmerMod']
Formula: y ~ x * a + (x + 1 | i)
   Data: dataII
REML criterion at convergence: 3038.791
Random effects:
 Groups   Name        Std.Dev. Corr
 i        (Intercept) 0.0000
          x           0.2995    NaN
 Residual             5.1052
Number of obs: 480, groups:  i, 20
Fixed Effects:
(Intercept)            x        aTRUE      x:aTRUE
   0.217634     0.005387    -0.674196     0.233632
convergence code 0; 2 optimizer warnings; 0 lme4 warnings
```

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

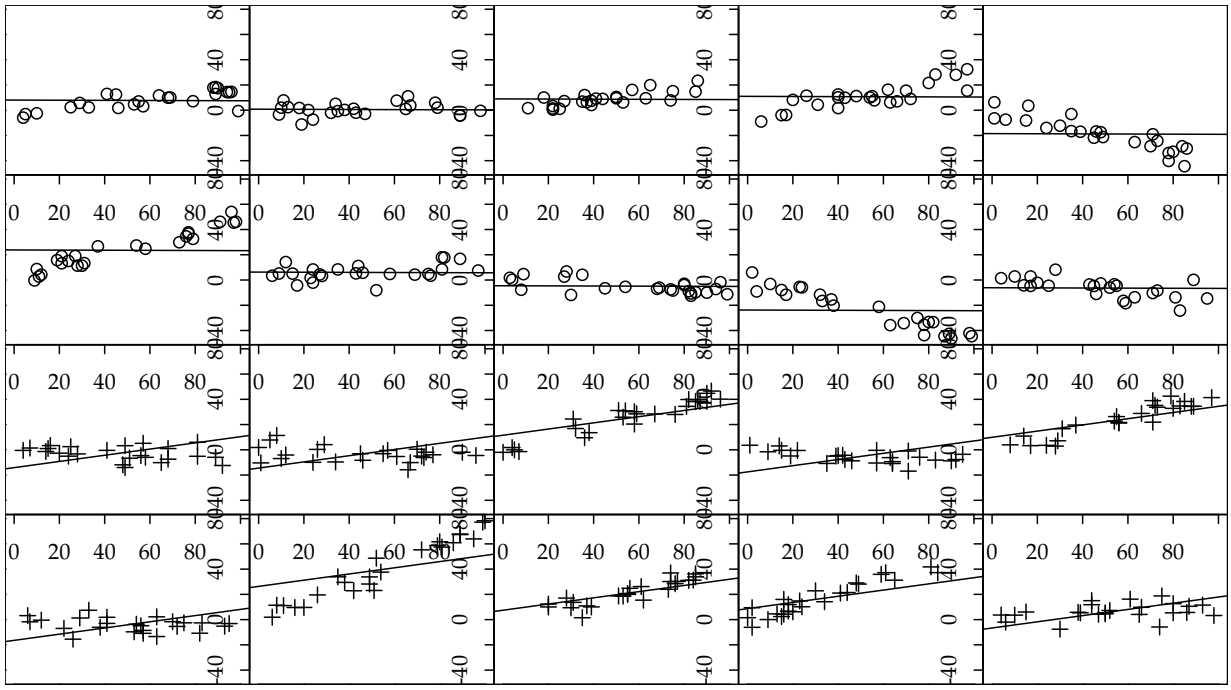
```
aa1 <- fixef(r1.lmer)["(Intercept)"]+aM*fixef(r1.lmer)["aTRUE"]+
    ranef(r1.lmer) [["i"]] [["(Intercept)"]]
bb1 <- fixef(r1.lmer)["x"]+aM*fixef(r1.lmer)["x:aTRUE"]
aa2 <- fixef(r2.lmer)["(Intercept)"]+aM*fixef(r2.lmer)["aTRUE"]+
    ranef(r2.lmer) [["i"]] [["(Intercept)"]]
bb2 <- fixef(r2.lmer)["x"]+aM*fixef(r2.lmer)["x:aTRUE"]+
    ranef(r2.lmer) [["i"]] [["x"]]
```

```
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,pch=1+2*as.numeric(a))
    abline(a=aa[j],b=bb[j])
  }))
  }
```

The following graph shows the predicted values for each individual.

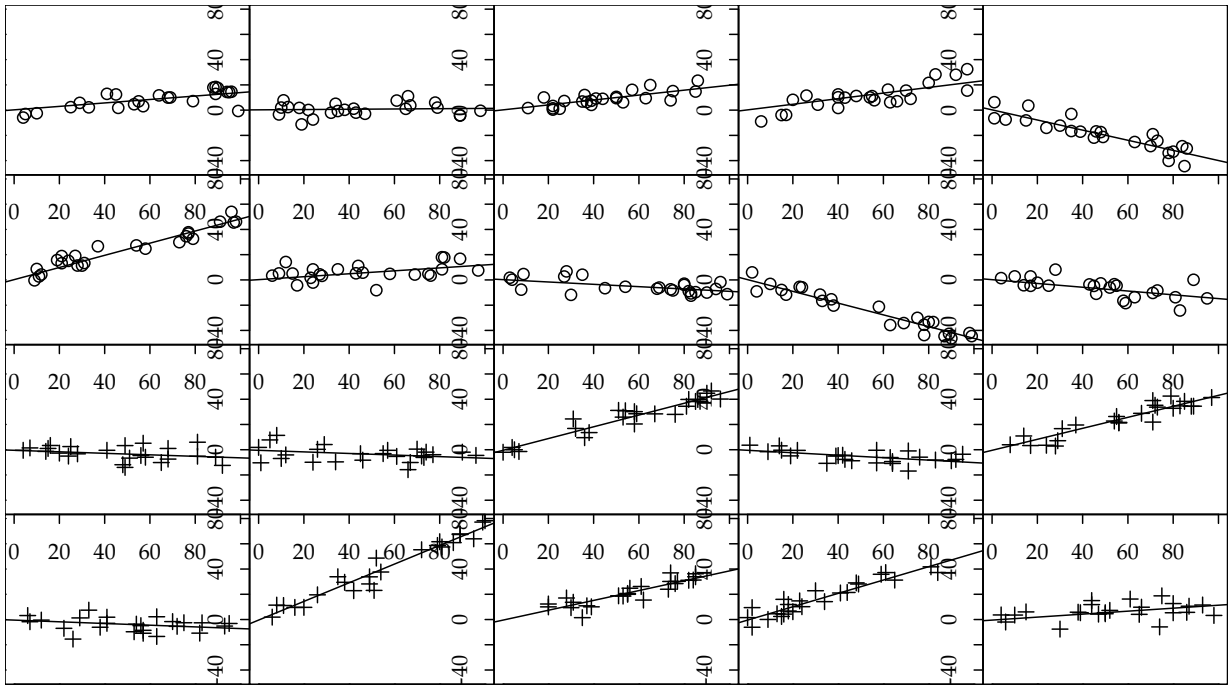
random effect only for intercept:

```
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 deviance Chisq Chi Df Pr(>Chisq)
r1.lmer  6 3651.0 3676.0 -1819.5   3639.0
r2.lmer  8 3051.1 3084.5 -1517.5   3035.1 603.9      2  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 8.1 *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!*

9 Nonlinear models

As in section 2.1 we create an example dataset. The difference is that y is now a binary variable.

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

      y          x i
1 FALSE 0.01584211 a
2 FALSE 0.02709743 a
3 FALSE 0.06225133 a
4 FALSE 0.06354326 a
5 FALSE 0.07200932 a
6 FALSE 0.08970296 a
```

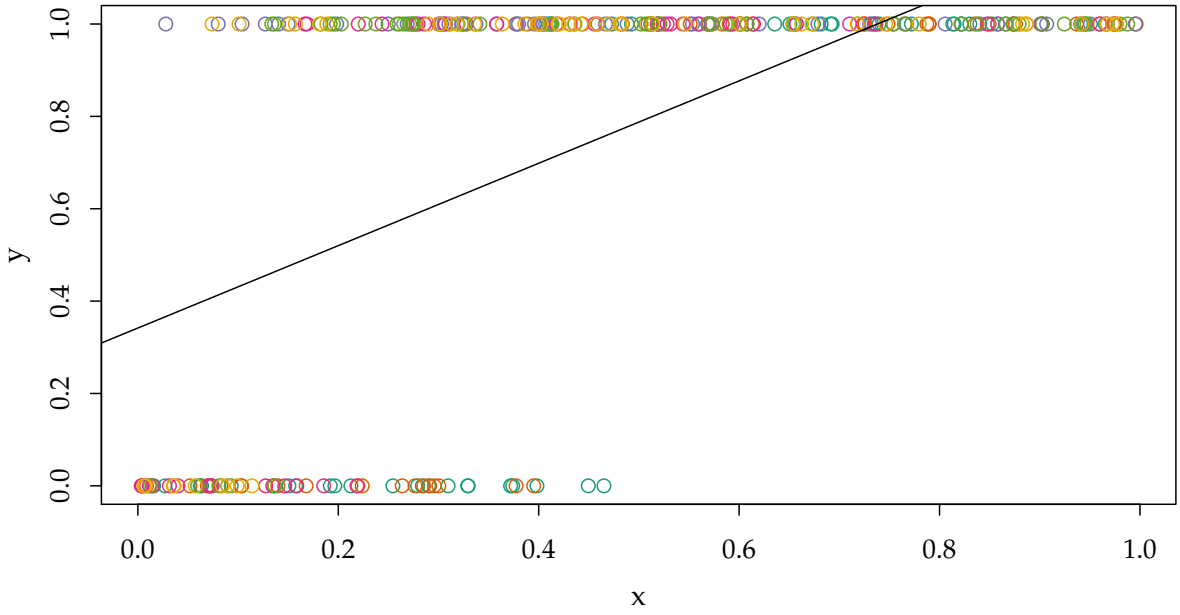
$$\Pr(Y_{ik}=1)=\mathcal{L}(\beta_0+\beta_1 X_{ik}+\nu_i)$$

True relationship:

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

9.1 Pooled linear regression

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



9.2 Pooled logistic regression

$$\Pr(Y_{ik} = 1) = \mathcal{L}(\beta_0 + \beta_1 X_{ik})$$

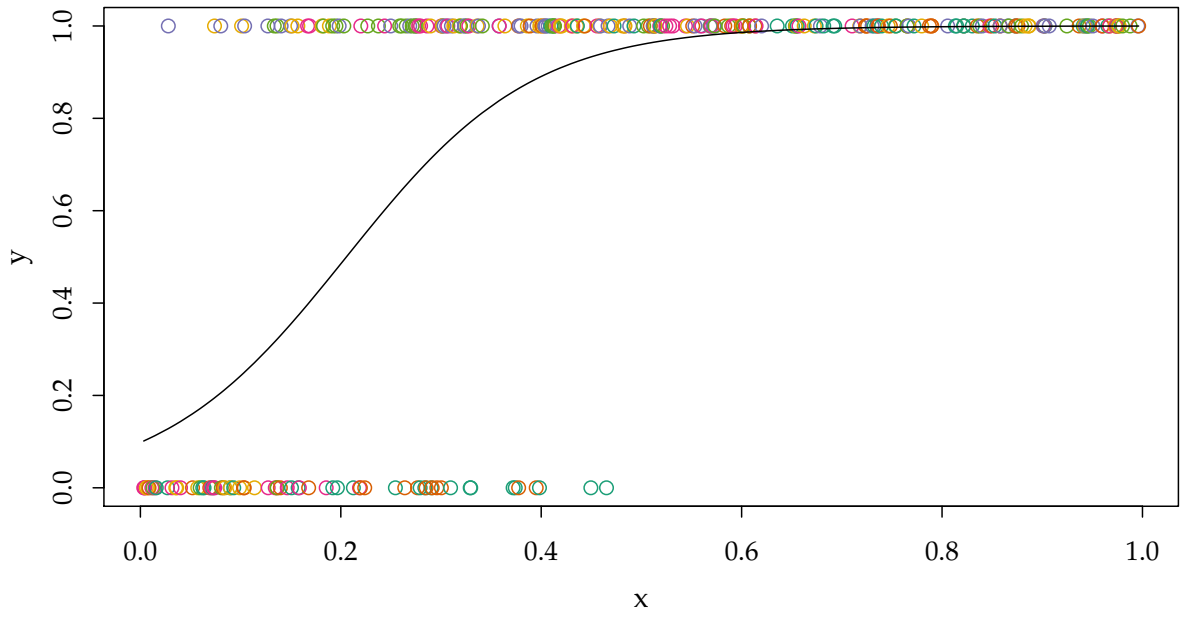
$$\Pr(Y = 1|x) \quad = \quad F(X, \beta)$$

```
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)
```



9.3 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  0.6385 12.07 0.000512 ***
x            10.7891  1.8658 33.44 7.36e-09 ***
```

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

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.613  0.4623

Correlation: Structure = independenceNumber of clusters:   6   Maximum cluster size: 50
```

9.4 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",]
wilcox.test(estBetax)
```

```

Wilcoxon signed rank test

data:  estBetax
V = 20, p-value = 0.06
alternative hypothesis: true location is not equal to 0
```

9.5 Fixed effects

$$\Pr(Y_{ik}=1)=\mathcal{L}(\alpha_i+\beta_1 X_{ik})$$

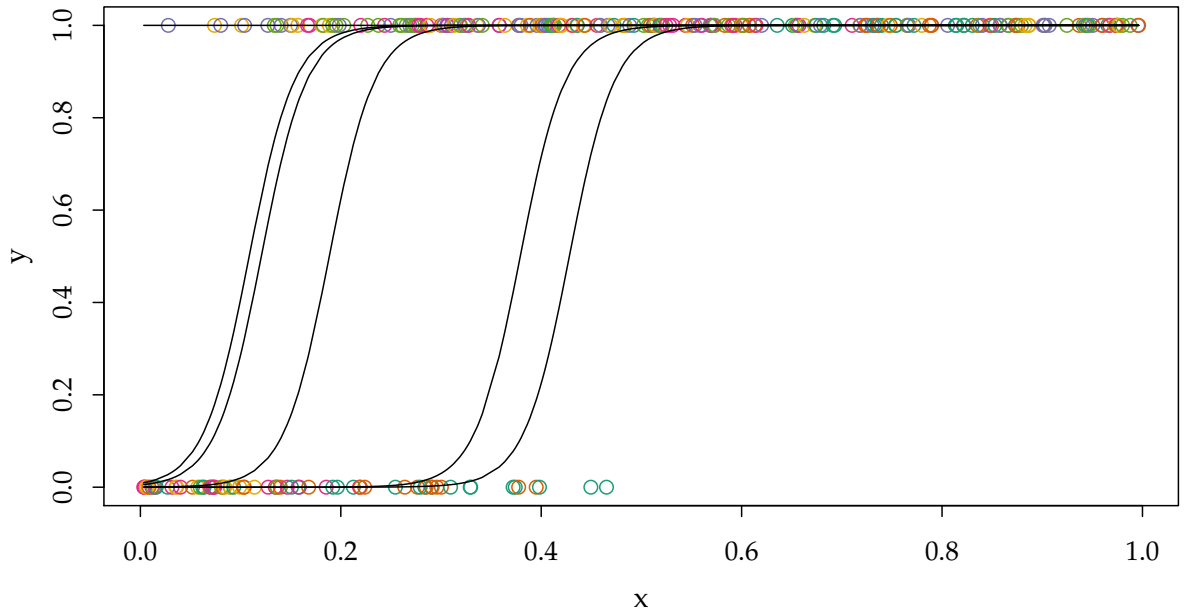
```
est.fixed <- glm (y ~ x + i,family=binomial(link="logit"),data=data)
```

```
summary(est.fixed)
```

```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9025   0.0000   0.0000   0.0022   1.8557

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -18.86       4.41   -4.27  1.9e-05 ***
x              44.06      10.19    4.32  1.5e-05 ***
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  2.7e-05 ***
if             14.08       3.50    4.03  5.6e-05 ***
---
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))
```



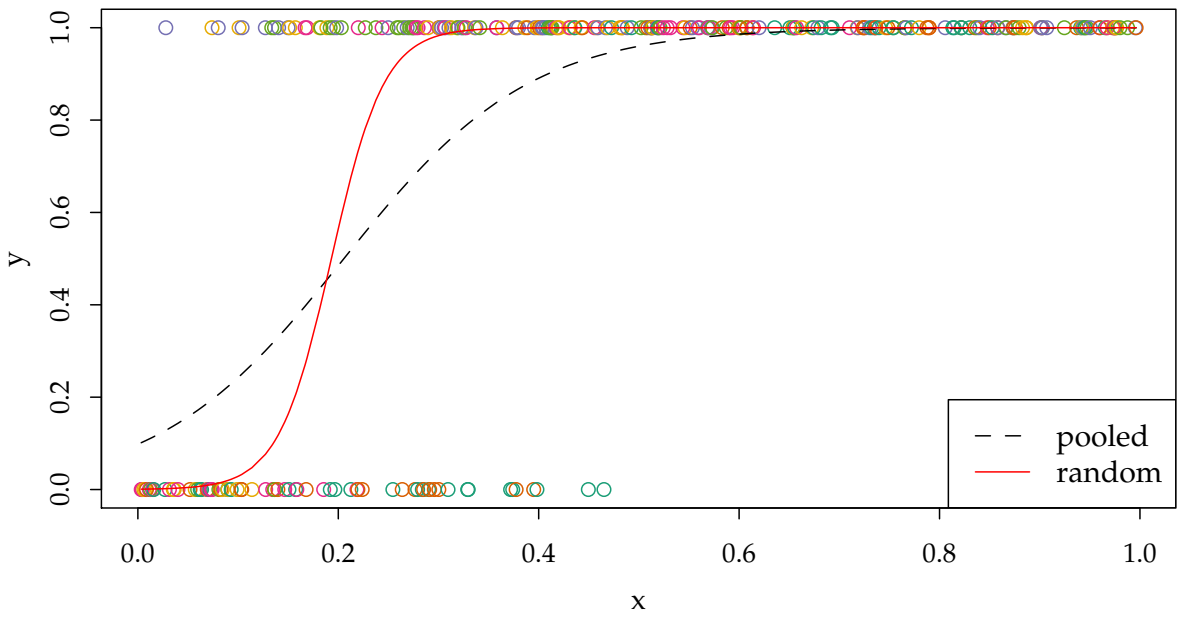
9.6 Random effects

$$\Pr(Y_{ik} = 1) = \beta_0 + \beta_1 X_{ik} + \nu_i$$

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

Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: y ~ x + (1 | i)
   Data: data
     AIC      BIC   logLik deviance df.resid
   81.33    92.45   -37.67    75.33      297
Random effects:
 Groups Name        Std.Dev.
 i      (Intercept) 6.33
Number of obs: 300, groups:  i, 6
Fixed Effects:
(Intercept)            x
      -7.31        37.87
```

```
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),col=c("black","red"))
```



9.7 Bayes and non-linear mixed effects

```
library(runjags)
modelL <- 'model {
 for (i in 1:length(y)) {
        y[i] ~ dbern(p[i])
        logit(p[i]) <- beta0+beta1*x[i]+nu[group[i]]
 }
 for (j in 1:max(group)) {
```

```
    nu[j] ~ dnorm(0,tauNu)
 }
 beta0   ~ dnorm (0,.0001)
 beta1   ~ dnorm (0,.0001)
 tauNu   ~ dgamma(.01,.01)
 sdNu   <- sqrt(1/tauNu)
}'
dataList<-list(y=as.numeric(data$y),x=data$x,group=as.numeric(data$i))
bayesL<-run.jags(model=modelL,modules="glm",data=dataList,
                 monitor=c("beta0","beta1","sdNu"))
```

```
summary(bayesL)

      Lower95 Median Upper95   Mean    SD   Mode  MCerr MC%ofSD SSeff  AC.10
beta0  -16.71 -7.502   2.125 -7.614 4.757 -7.502 0.1888     4.0   635 0.1209
```

```
beta1   29.52 40.110  55.513 41.026 6.967 39.346 1.6155    23.2    19 0.9789
sdNu     3.45  8.578  18.188  9.650 4.772  7.614 0.2429     5.1   386 0.1878
       psrf
beta0 1.008
beta1 1.104
sdNu  1.007
```

Compare with ML:

```
fixef(est.mer)

(Intercept)           x
     -7.314      37.868

summary(est.mer)$varcor

 Groups Name        Std.Dev.
 i      (Intercept) 6.33
```

Exercise 9.1 *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?*

10 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 <- -.2
sdChange <- 1
effectSize <- .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 <<- rbind(allData,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
```

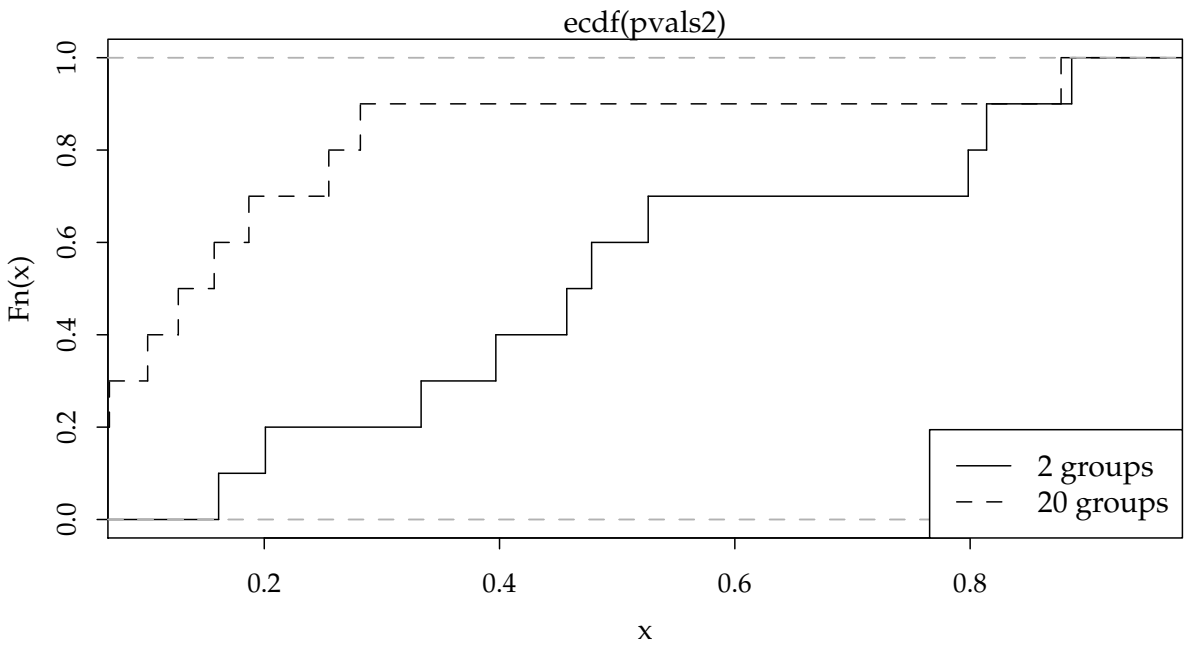
This looks fine. Now it is time to write a small function that calculates the statistics we care about for one simulated experiment. Let us assume that we care about p-values.

```
oneSimul <- function (groupNum) {
  xx <- groups(groupNum)
  est.mer <- lmer(contrib ~ effect + (1|pid) + (1|gid) ,data=xx)
  2*pnorm(abs(summary(est.mer)$coefficients["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")
```

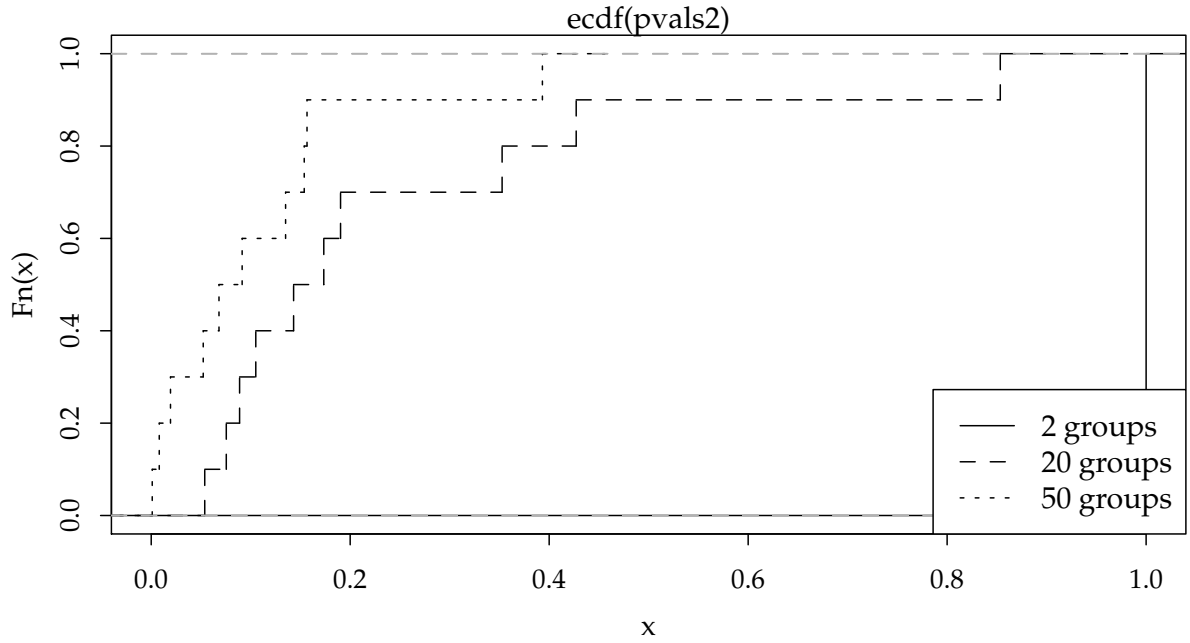


Now let us assume that we are more conservative and want to apply a Wilcoxon rank sum test.

```
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 10.1 *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?*

11 Exercises

Exercise 11.1 *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* x *is positive.*

Exercise 11.2 *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?*