Methods 3: Multilevel Statistical Modeling and Machine Learning

Week 3: *Generalized linear mixed effects models*
September 28, 2021

by: Lau Møller Andersen

Messages

- Practical exercise due 23.59 tomorrow
- Make sure to add your GitHub repository – a few are still missing: https://cryptpad.fr/pad/#/2/pad/edit/U21qNTbLgfkRiGZU1bnmDE2o/
- Remember, Class 2 (10-12) will be in 1453-116 tomorrow

RECAP on pooling

SLEEP STUDY EXAMPLE

https://psyteachr.github.io/stat-models-v1/introducing-linear-mixed-effects-models.html

Learning goals and outline –

Linear Mixed Effects Models (LMM)

1) Why can it be a good idea to do mixed effects modelling?

2) Understanding the basics of multilevel modelling

   - also known as linear mixed effects modelling

3) Appreciating the difference between the different levels of effects

   – or *random* and *fixed* effects, as they are also called

4) Understanding the concept of pooling (none, complete and partial)

Pooling - summary

- Complete pooling
  - ignoring a categorical predictor (e.g. *subject*)
- No pooling
  - model each level of the categorical predictor separately
- Partial pooling
  - we model both an average and each level of the categorical predictor (e.g. *subject*)

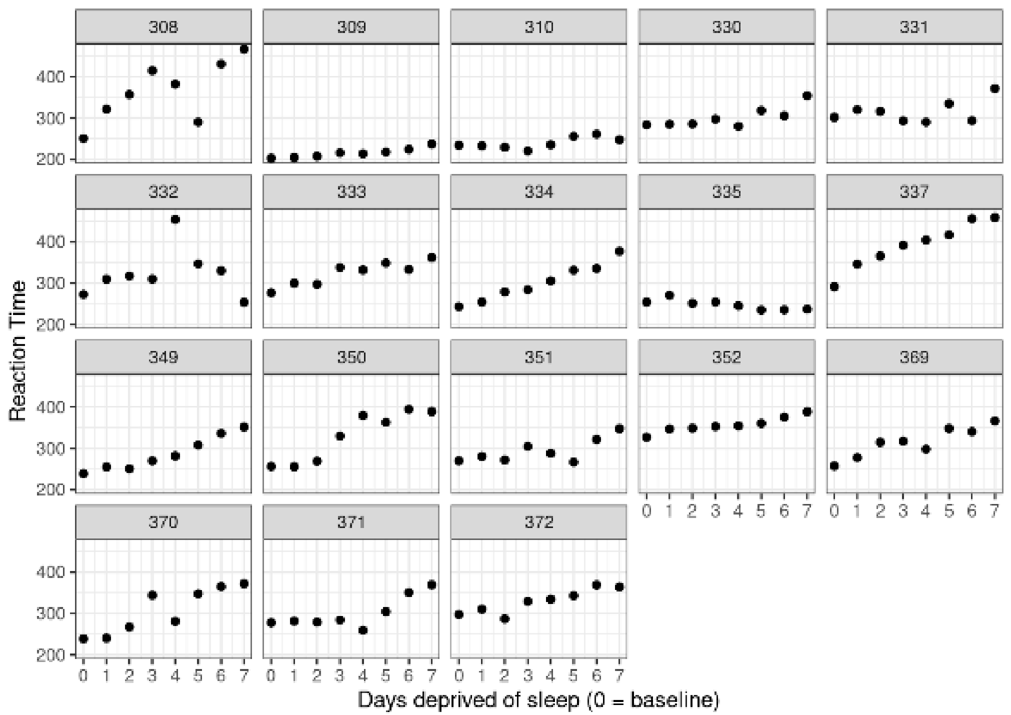
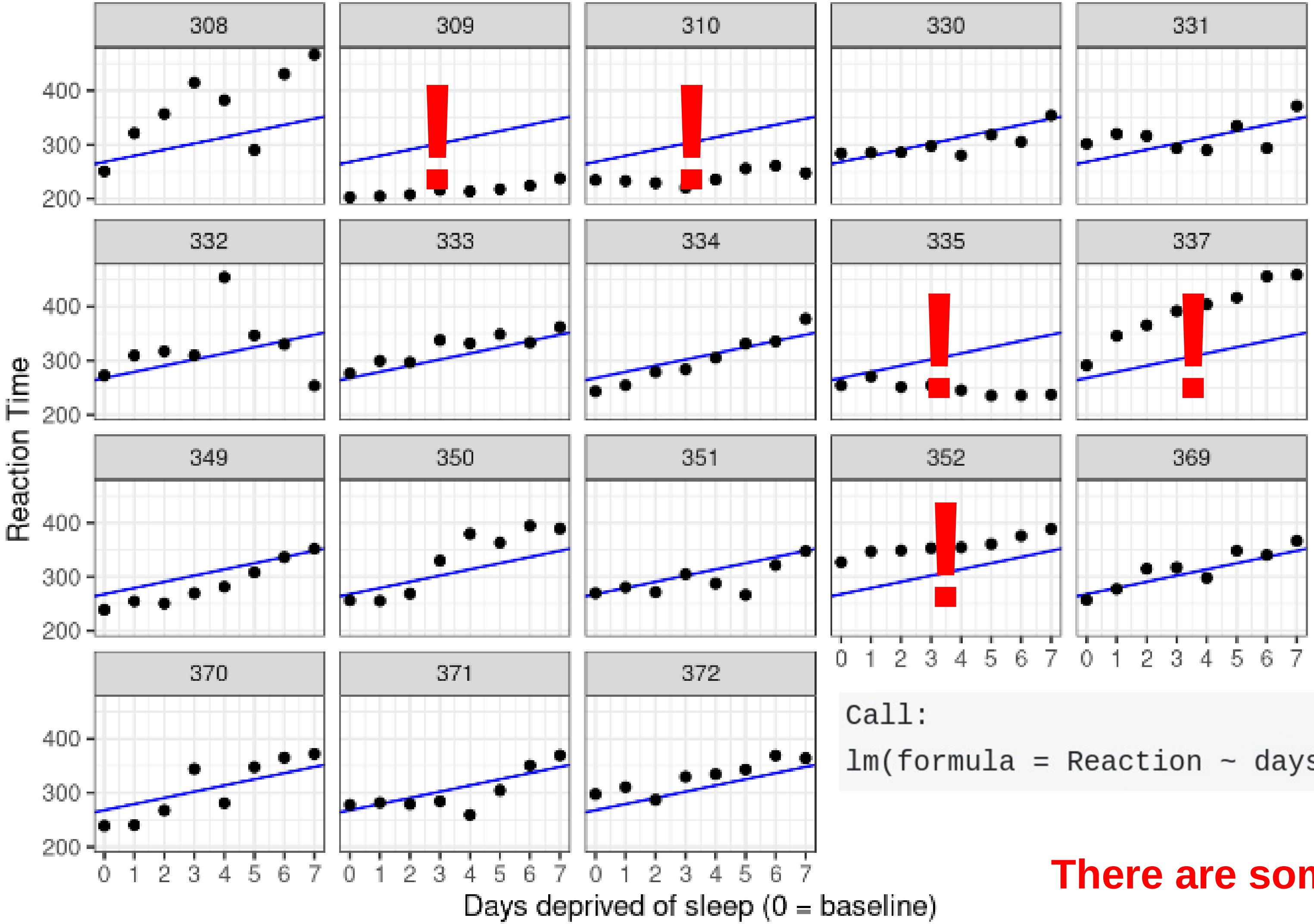


Figure 5.3: *Data from Belenky et al. (2003), showing reaction time at baseline (0) and after each day of sleep deprivation.*

(Barr, University of Glasgow)

COMPLETE POOLING




```
Coefficients:
               Estimate
(Intercept)     267.967
days_deprived    11.435
```

```
Call:
lm(formula = Reaction ~ days_deprived, data = sleep2)
```

There are some bad fits

(Barr, University of Glasgow)

```
lm(formula = Reaction ~ days_deprived + Subject + days_deprived:Subject,
    data = sleep2)
```

```
## Coefficients:
##                               Estimate
## (Intercept)                   288.2175
## days_deprived                  21.6905
## Subject309                    -87.9262
## Subject310                    -62.2856
## Subject330                    -14.9533
## Subject331                      9.9658
## Subject332                     27.8157
```

... and the remaining 12 subjects

```
## days_deprived:Subject309 -17.3334
## days_deprived:Subject310 -17.7915
## days_deprived:Subject330 -13.6849
## days_deprived:Subject331 -16.8231
## days_deprived:Subject332 -19.2947
## days_deprived:Subject333 -10.8151
```

... and the remaining 12 subjects

NO POOLING

(Barr, University of Glasgow)

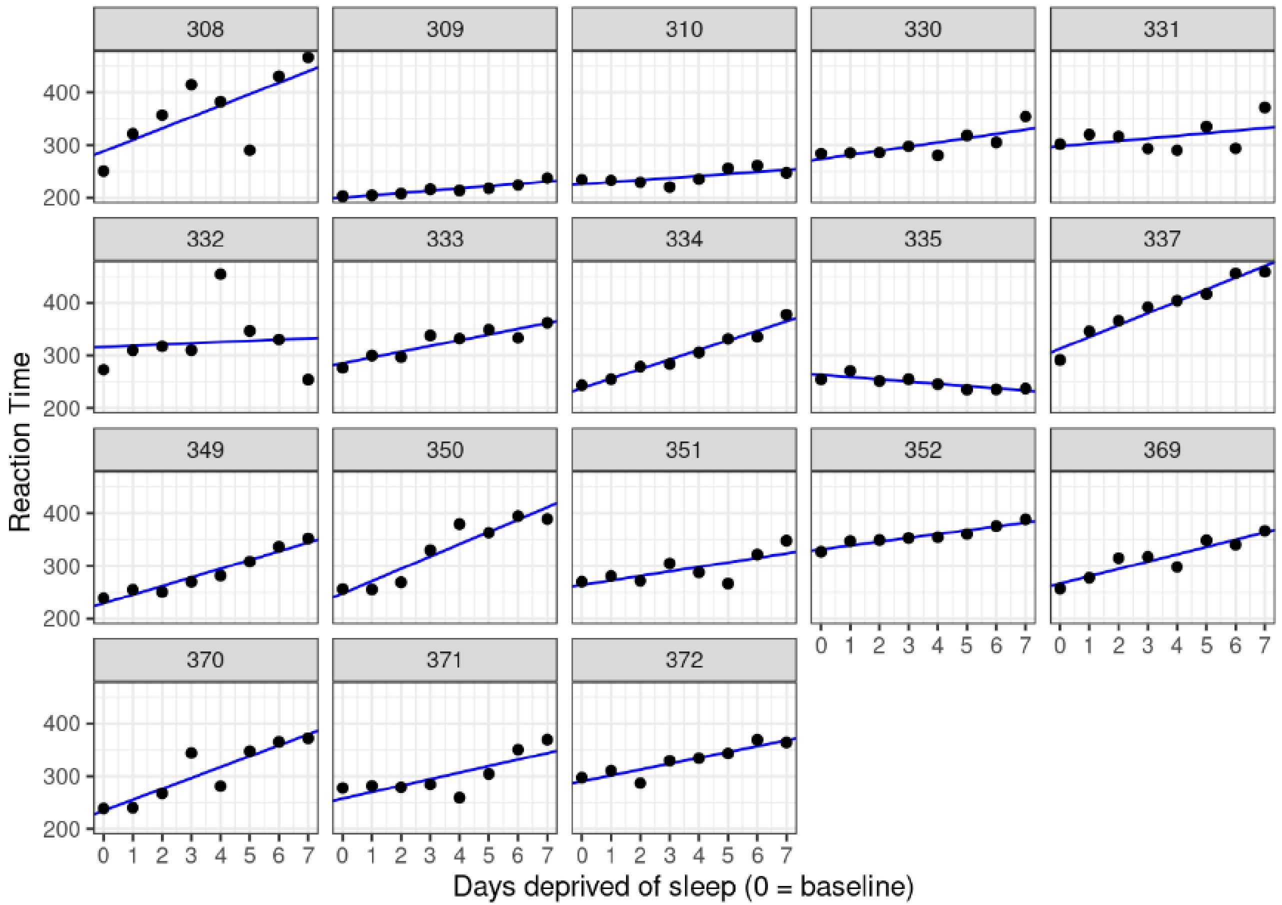


Figure 5.5: Data plotted against fits from the no-pooling approach.

NO POOLING

Good fits now:

What are the limits of this model?

(Barr, University of Glasgow)

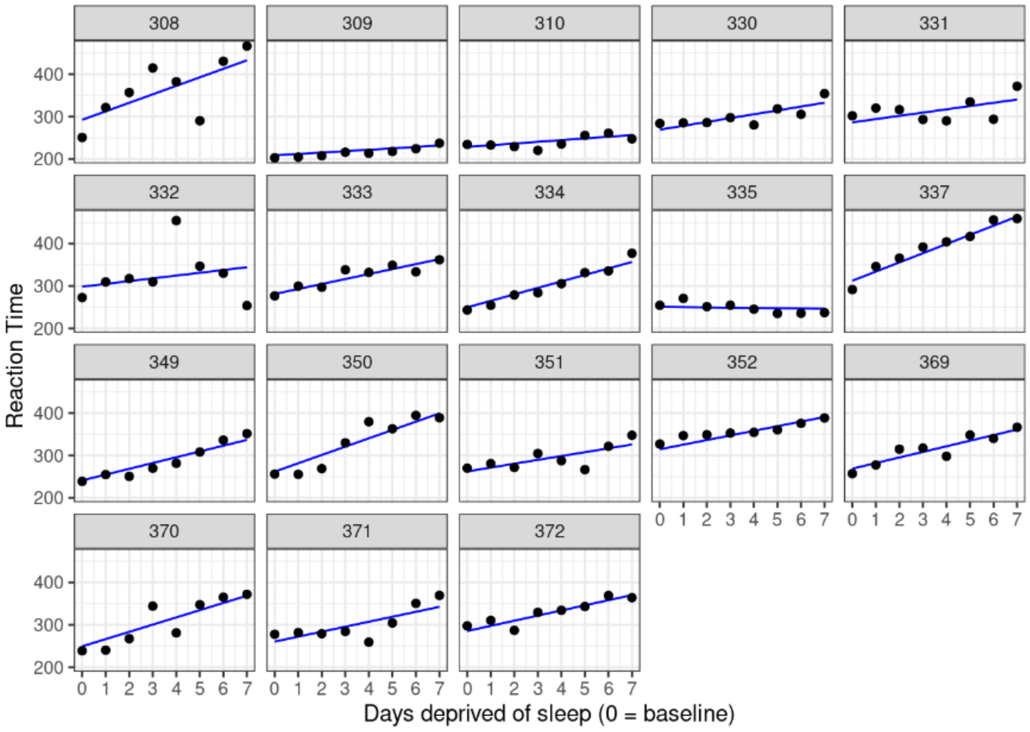


Figure 5.6: Data plotted against predictions from a partial pooling approach.

PARTIAL POOLING

```
Fixed effects:
             Estimate Std. Error t value
(Intercept)   267.967      8.266  32.418
days_deprived  11.435      1.845   6.197
```

```
ranef(pp_mod)[["Subject"]]
```

	(Intercept)	days_deprived
308	24.4992891	8.6020000
309	-59.3723102	-8.1277534
310	-39.4762764	-7.4292365
330	1.3500428	-2.3845976

```
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ days_deprived + (days_deprived | Subject)
   Data: sleep2
```


(Barr, University of Glasgow)

No pooling vs partial pooling

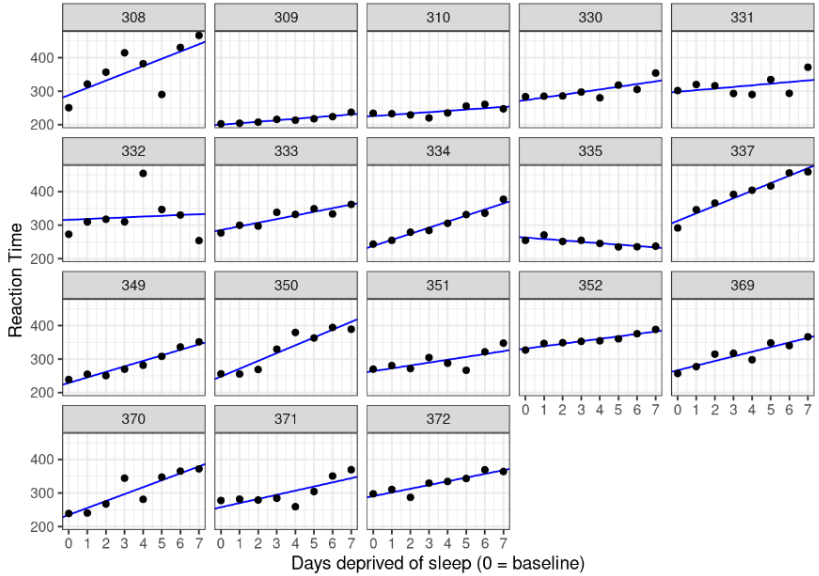


Figure 5.5: Data plotted against fits from the no-pooling approach.

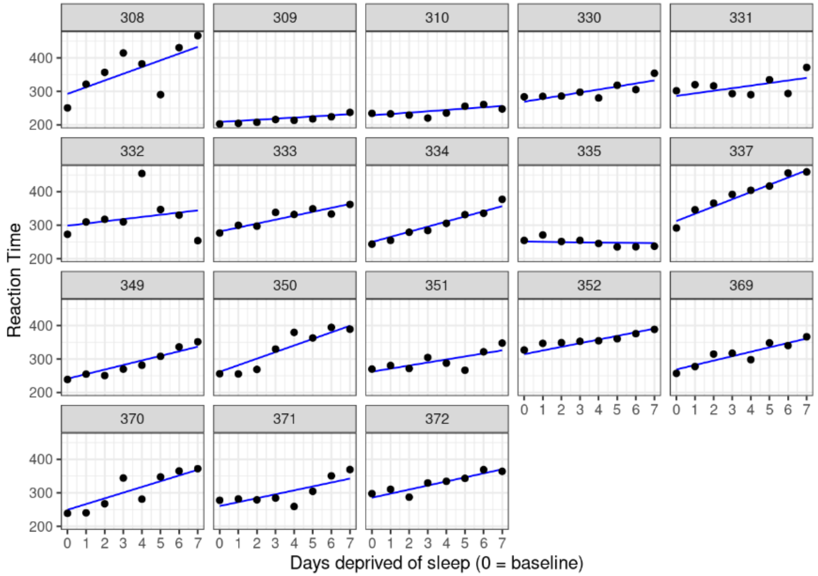


Figure 5.6: Data plotted against predictions from a partial pooling approach.

Both model the individual variance – but only one is generalisable outside the subject pool

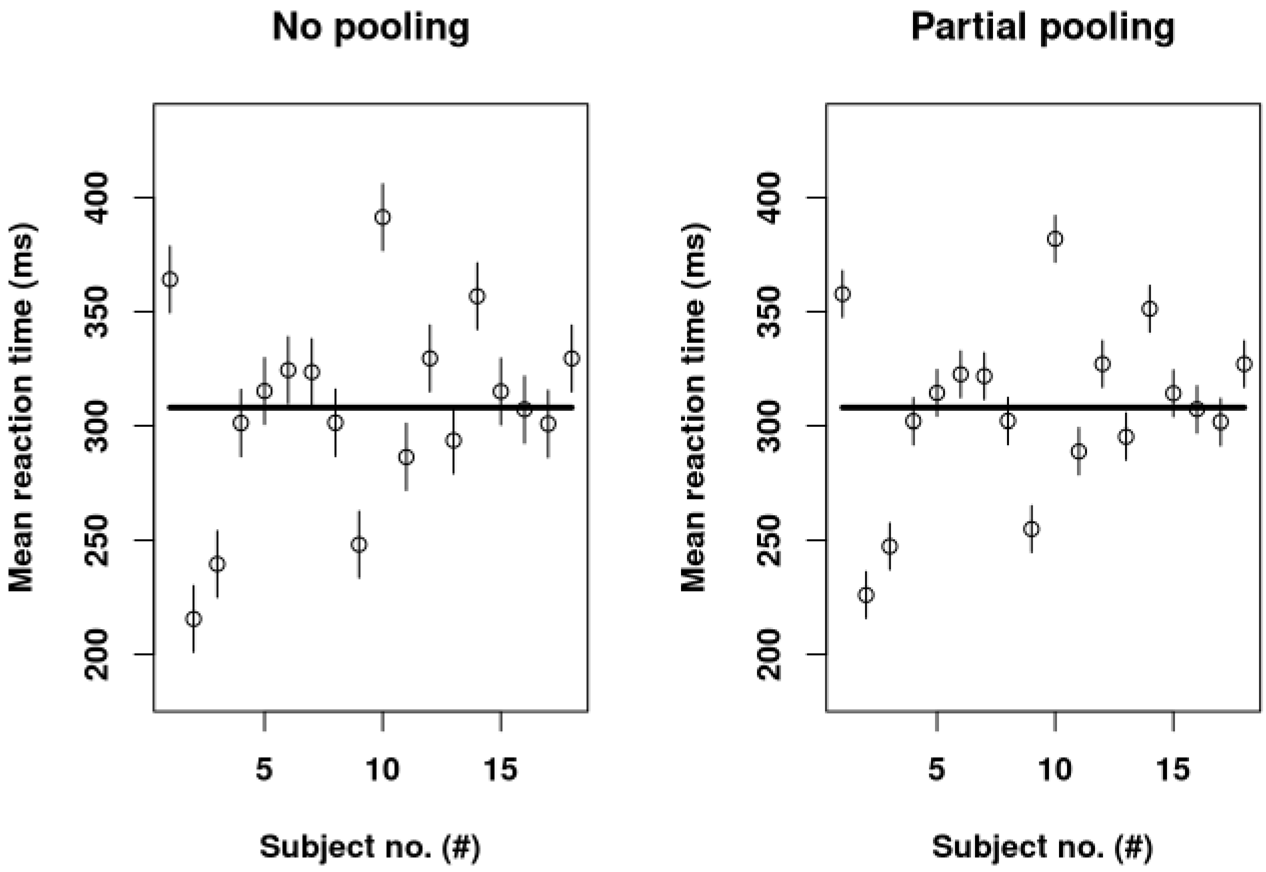
(Barr, University of Glasgow)

Partial pooling

(Gelman and Hill, 2006 (12.1))

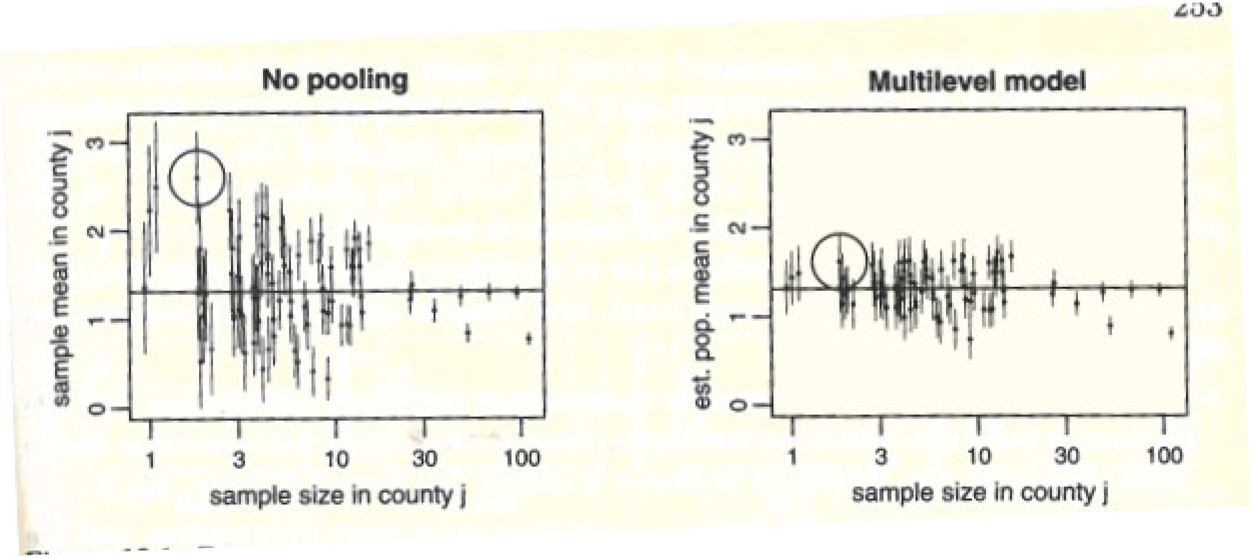
$$\hat{\alpha}_j^{multilevel} \approx \frac{\frac{n_j}{\sigma_y^2}\bar{y}_j + \frac{1}{\sigma_\alpha^2}\bar{y}_{all}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}$$

another scary looking thing...



What is the advantage of the partial pooling model?

Now with different sample sizes



(Gelman and Hill, 2006)

What is the advantage of the partial pooling model?

$$\hat{\alpha}^{multilevel} \approx \frac{\frac{n_j}{\sigma_y^2}\bar{y}_j + \frac{1}{\sigma_\alpha^2}\bar{y}_{all}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}$$

(Gelman and Hill, 2006)

$\hat{\alpha}_j$: estimated mean for subject j

n_j: sample size for subject j

σ_y^2: within-subject variance

σ_α^2: variance around the average

$\bar{y}_j$: unpooled estimate of subject j

$\bar{y}_{all}$: the pooled estimate

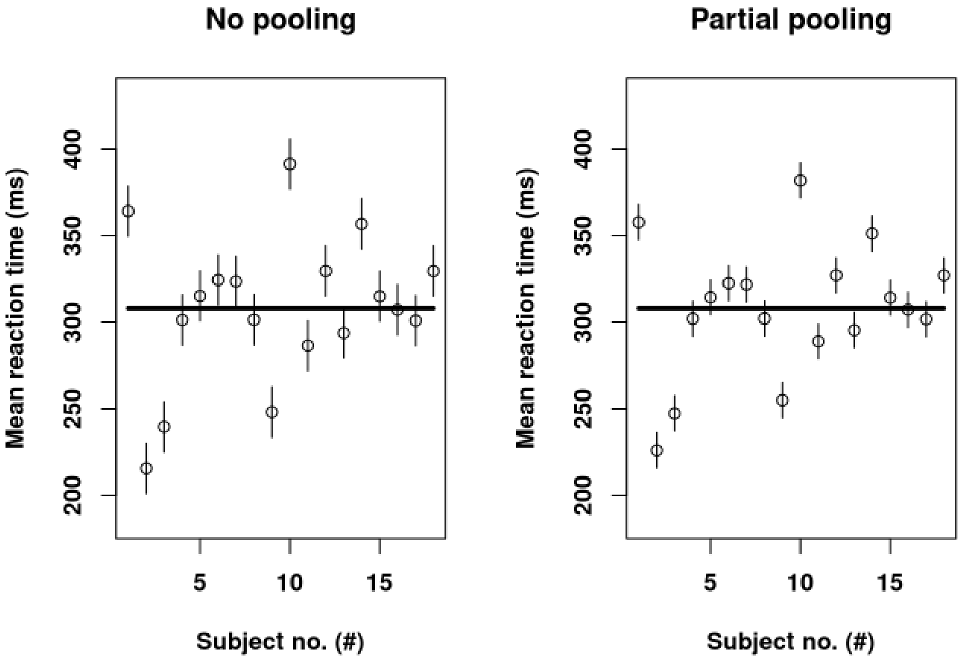
Discuss in small groups

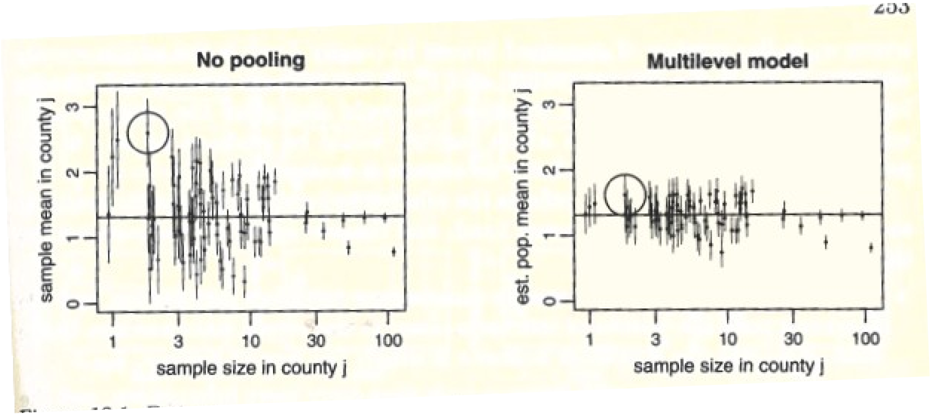
What happens to the estimated mean, $\hat{a}_j$, when n_j:

1) increases?
2) decreases?
3) is 0?
4) goes towards infinity?

Same *n* for each subject

Different *n* for each county



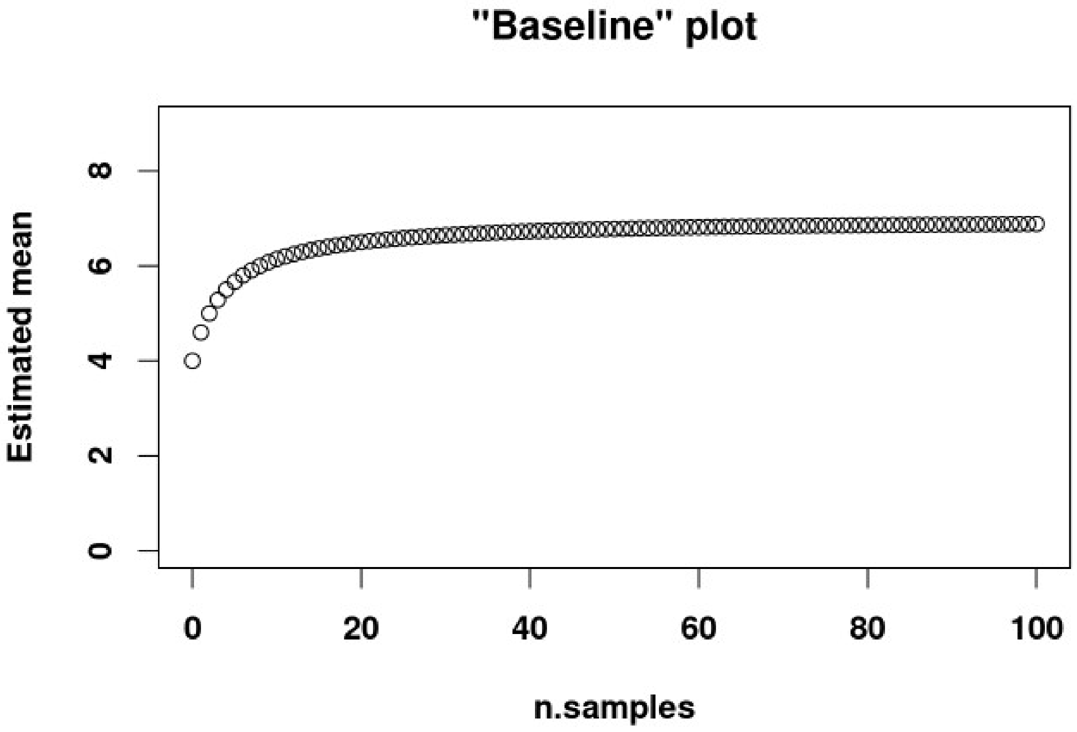


(Gelman and Hill, 2006)

```
estimate.multilevel.mean <- function(n.j, sigma.y, sigma.mean, y.j, y.al
l)
{
    alpha <- ((n.j / sigma.y^2) * y.j + (1 / sigma.mean^2) * y.all) /
             ( (n.j / sigma.y^2) + (1 / sigma.mean^2))
    return(alpha)
}
```

```
## "baseline"

sigma.y <- 3
y.j <- 7
sigma.mean <- 1.5
y.all <- 4
ns <- 0:100
```

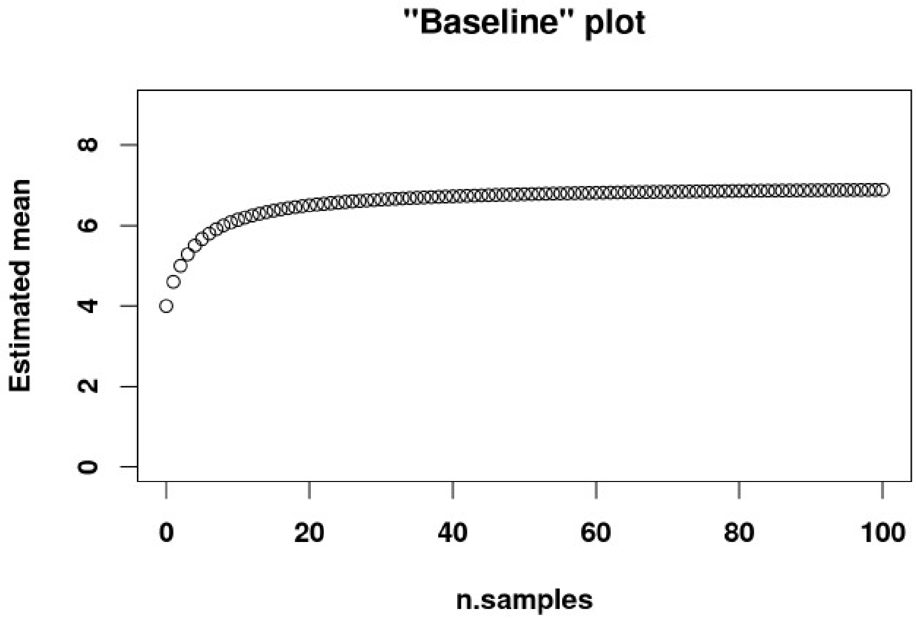


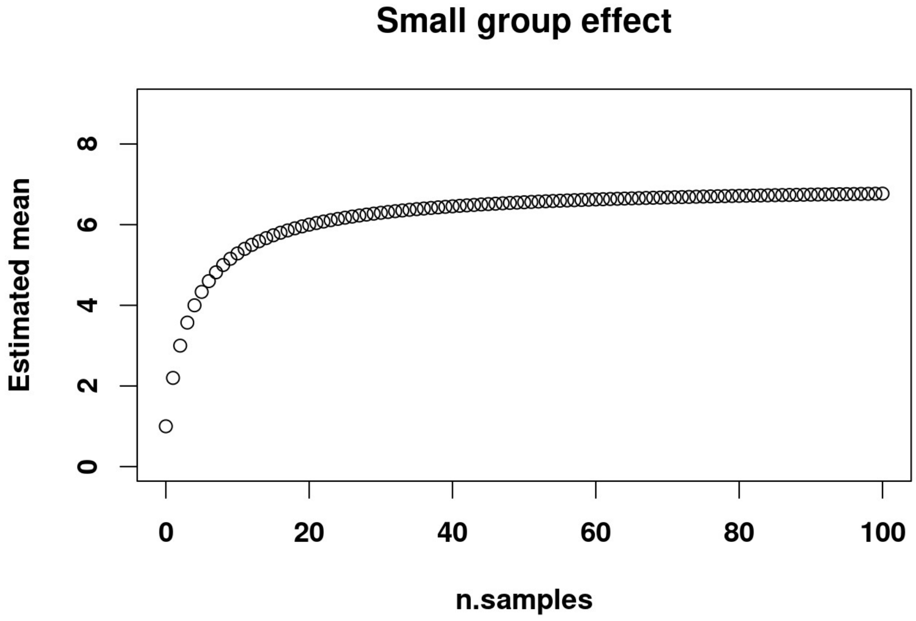
```r
## "baseline"

sigma.y <- 3
y.j <- 7
sigma.mean <- 1.5
y.all <- 4
ns <- 0:100
```

```r
## small group effect

sigma.y <- 3
y.j <- 7
sigma.mean <- 1.5
y.all <- 1
ns <- 0:100
```



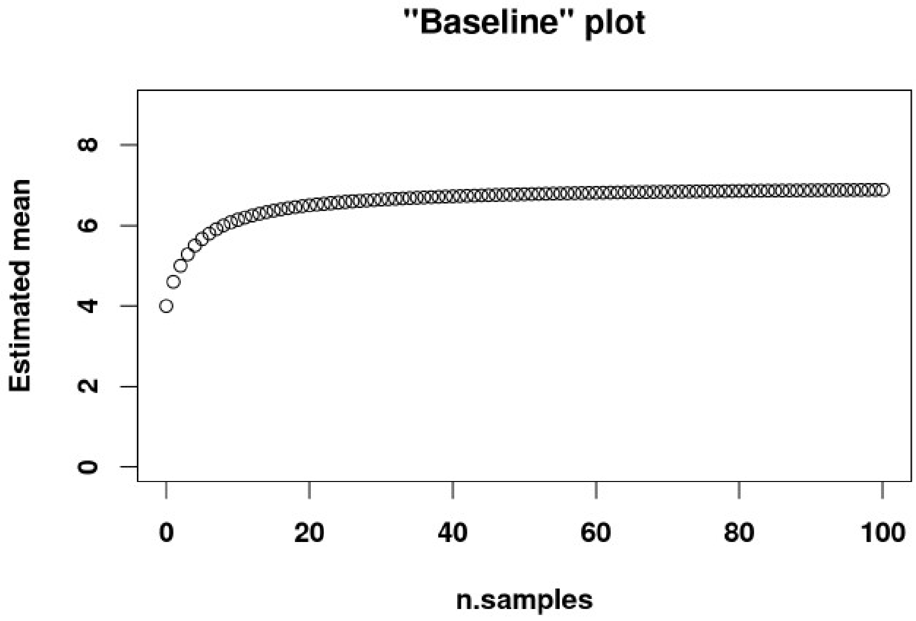


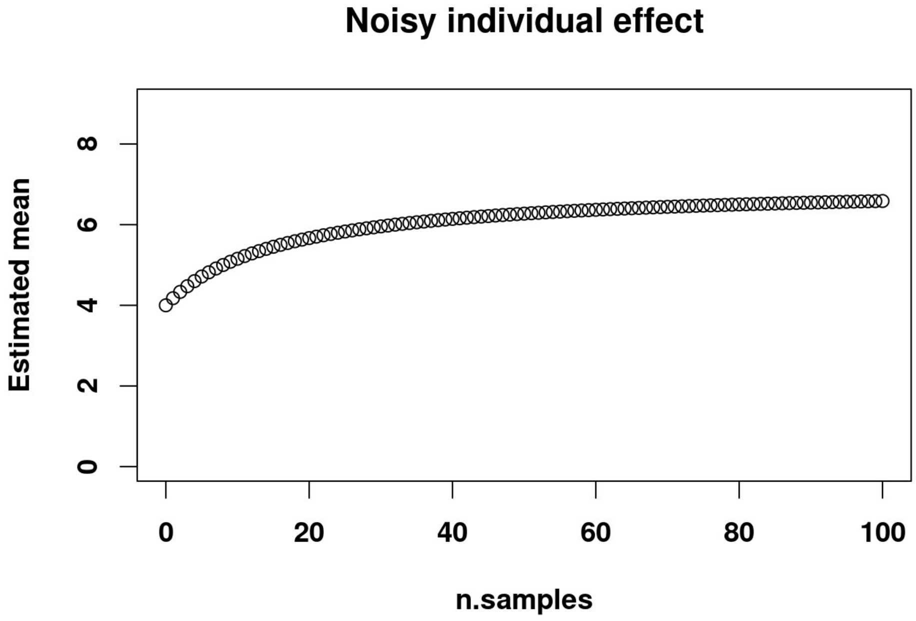
```
## "baseline"

sigma.y <- 3
y.j <- 7
sigma.mean <- 1.5
y.all <- 4
ns <- 0:100
```

```
## noisy individual effect

sigma.y <- 6
y.j <- 7
sigma.mean <- 1.5
y.all <- 4
ns <- 0:100
```



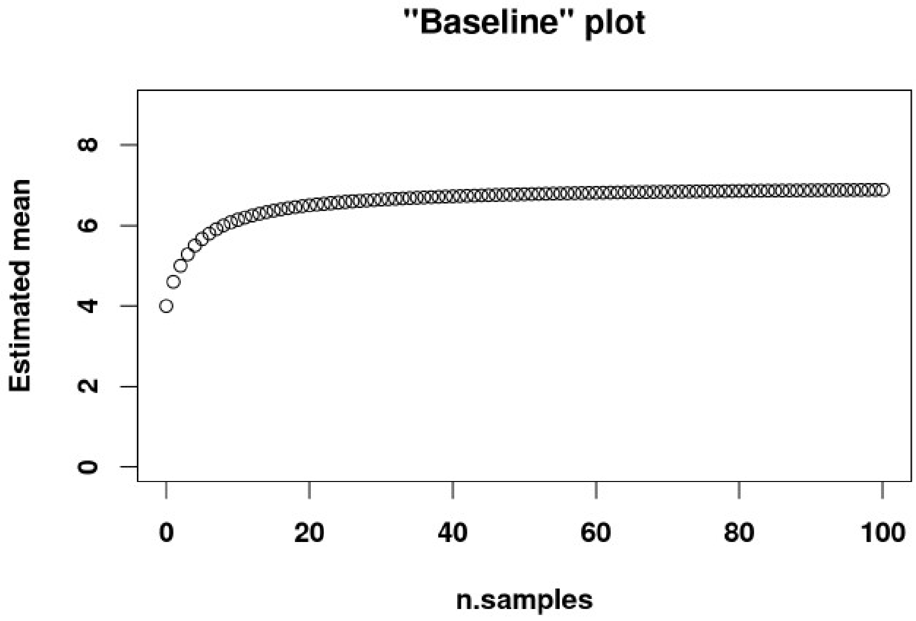


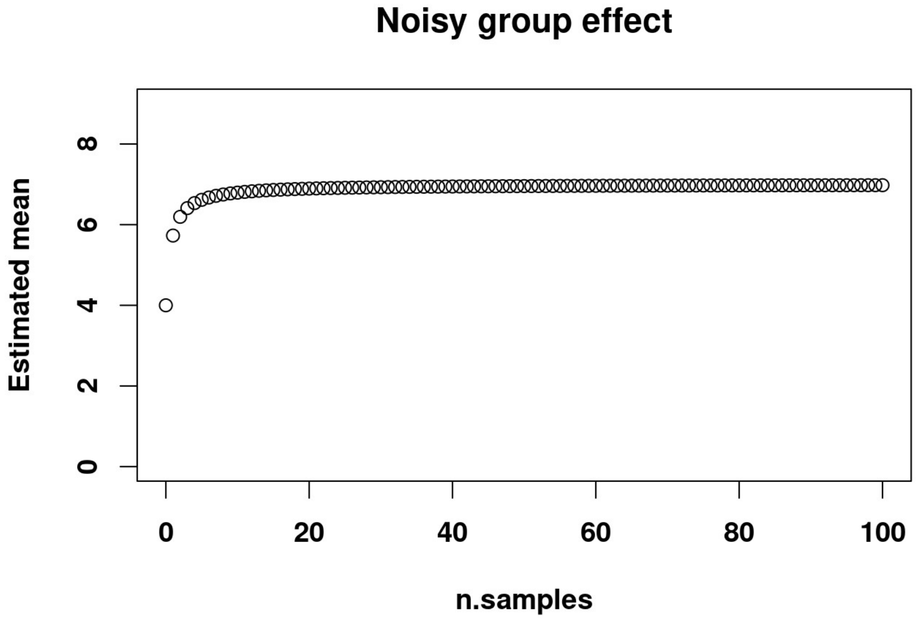
```r
## "baseline"

sigma.y <- 3
y.j <- 7
sigma.mean <- 1.5
y.all <- 4
ns <- 0:100
```

```r
## noisy group effect

sigma.y <- 3
y.j <- 7
sigma.mean <- 3.5
y.all <- 4
ns <- 0:100
```





FROM LAST WEEK

Motivation for multilevel modelling:

We want to use all the information in the data while fulfilling the assumptions necessary for the residuals

We can add:

Without letting small or uncertain samples unduly affect our group estimate

$$\hat{\alpha}^{multilevel} \approx \frac{\frac{n_j}{\sigma_y^2}\bar{y}_j + \frac{1}{\sigma_\alpha^2}\bar{y}_{all}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}$$

(Gelman and Hill, 2006 (12.1))

$\hat{\alpha}_j$: estimated mean for subject j

n_j: sample size for subject j

σ_y^2: within-subject variance

σ_α^2: variance around the average

$\bar{y}_j$: unpooled estimate of subject j

$\bar{y}_{all}$: the pooled estimate

Revisiting equation 12.1

Why is this not very interesting for purposes of fitting models?

Now with parameters (β)

$$\hat{\alpha}_j \approx \frac{\frac{n_j}{\sigma_y^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}} (\bar{y}_j - \beta \bar{x}_j) + \frac{\frac{1}{\sigma_\alpha^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}} \mu_\alpha$$

$\hat{\alpha}_j$: group level parameters

n_j: sample size for subject j

σ_y^2: within-subject variance

σ_α^2: variance around the average

$\bar{y}_j$: unpooled estimate of subject j

$\bar{y}_{all}$: the pooled estimate

$(\bar{y}_j - \beta \bar{x}_j)$: the unpooled estimate for the subject

μ_α: *mean*

Now on to generalized linear mixed models ...

Did you learn?

Linear Mixed Effects Models (LMM)

1) Why can it be a good idea to do mixed effects modelling?
2) Understanding the basics of multilevel modelling

    - also known as linear mixed effects modelling

3) Appreciating the difference between the different levels of effects

    – or *random* and *fixed* effects, as they are also called

4) Understanding the concept of pooling (none, complete and partial)

... but let's do a recap of the generalized linear model first

Learning goals

Generalized Linear Mixed Effects Models (GLMM)

1) Understanding that we can extend the scope of our multilevel modelling by using appropriate link functions and data distributions

2) Understanding the multilevel equivalent of the GLM

At least four ingredients needed

1) A data vector: $y=(y_1,...,y_n)$

2) Predictors: X and coefficients β, forming a linear predictor $X\beta$

3) A *link function* g: yielding a vector of transformed data $\hat{y}=g^{-1}(X\beta)$ that are used to model the data

4) A data distribution: $p(y|\hat{y})$

$$(X\beta=\beta_0+X_1\beta_1+...+X_k\beta_k)$$

(Gelman and Hill, 2006, Chapter 6)

Breaking all promises and going back to ***mtcars***

1) A data vector: $y = (y_1, \ldots, y_n)$

```
print(y <- mtcars$am)
```

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

2) Predictors: X and coefficients β, forming a linear predictor $X\beta$

```r
logistic.model <- glm(am ~ wt + 1, data=mtcars, family='binomial')
X <- model.matrix(logistic.model)
print(head(X))
```

```
##                   (Intercept)    wt
## Mazda RX4                   1 2.620
## Mazda RX4 Wag               1 2.875
## Datsun 710                  1 2.320
## Hornet 4 Drive              1 3.215
## Hornet Sportabout           1 3.440
## Valiant                     1 3.460
```

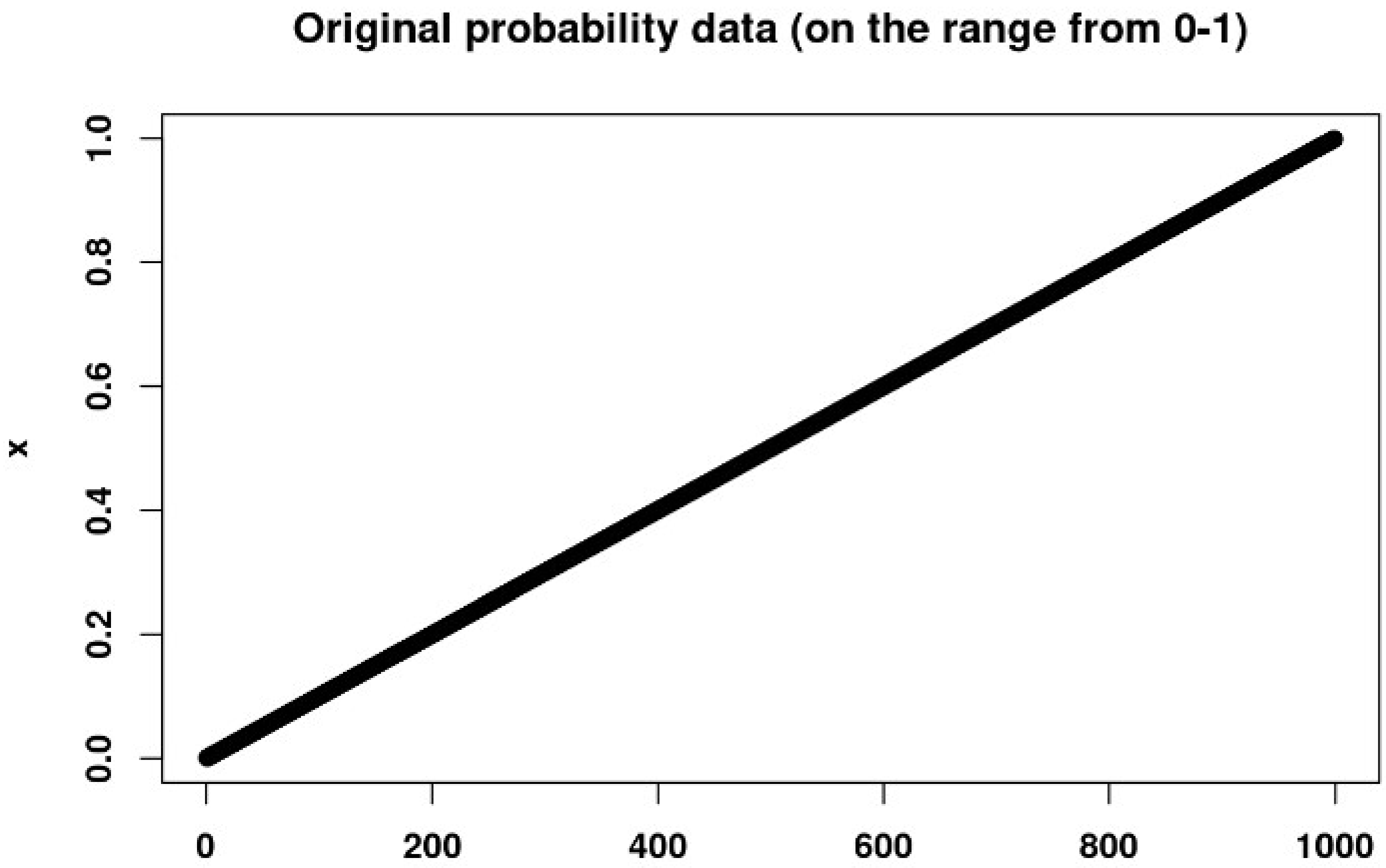
```r
print(beta.hat <- logistic.model$coefficients)
```

```
## (Intercept)          wt
##    12.04037    -4.02397
```

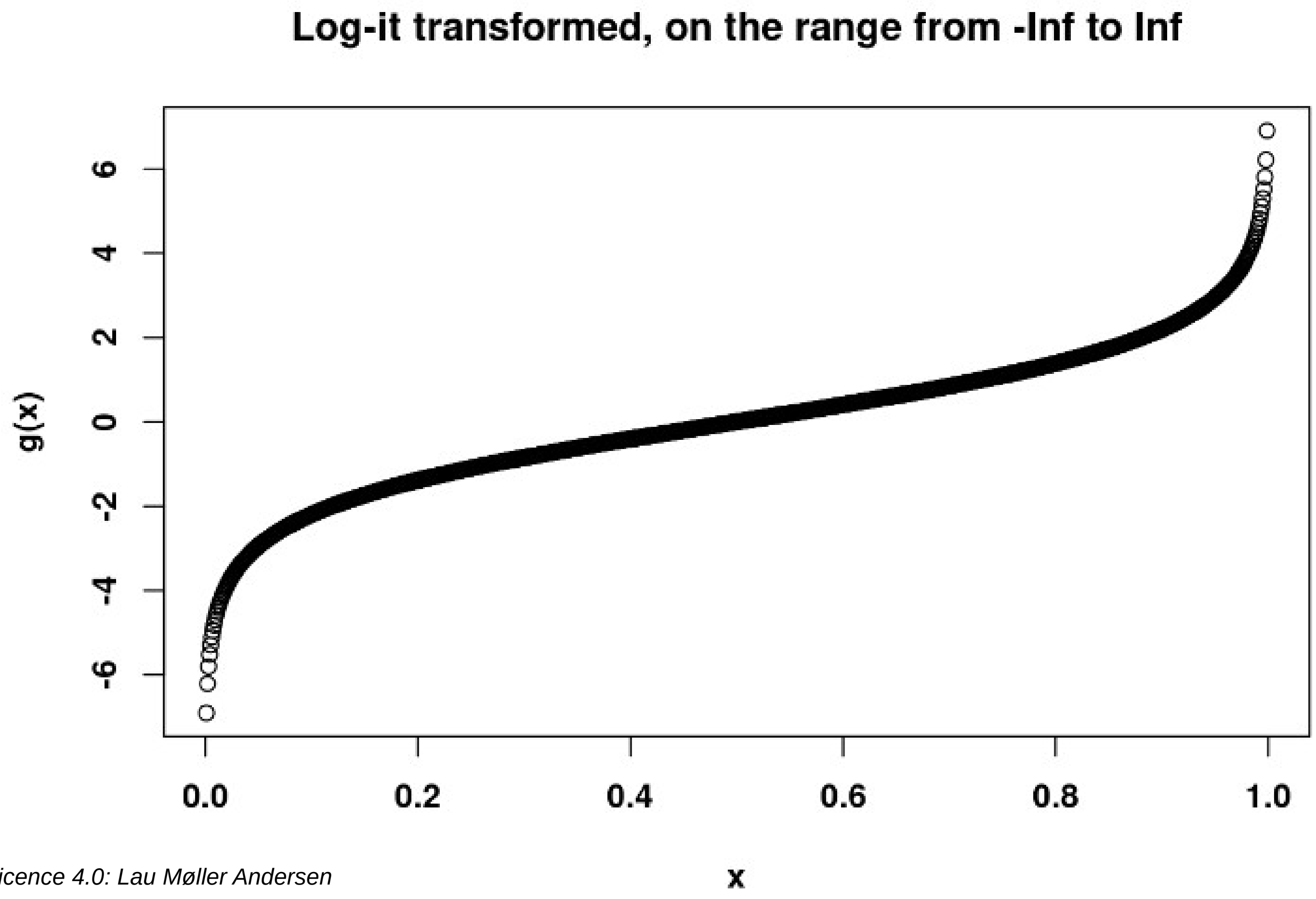
3) A *link function* g: yielding a vector of transformed data $\hat{y} = g^{-1}(X\beta)$ that are used to model the data

```r
g <- function(x) log(x / (1 - x)) ## logit
inv.g <- function(x) exp(x) / (1 + exp(x)) ##logit-¹
```

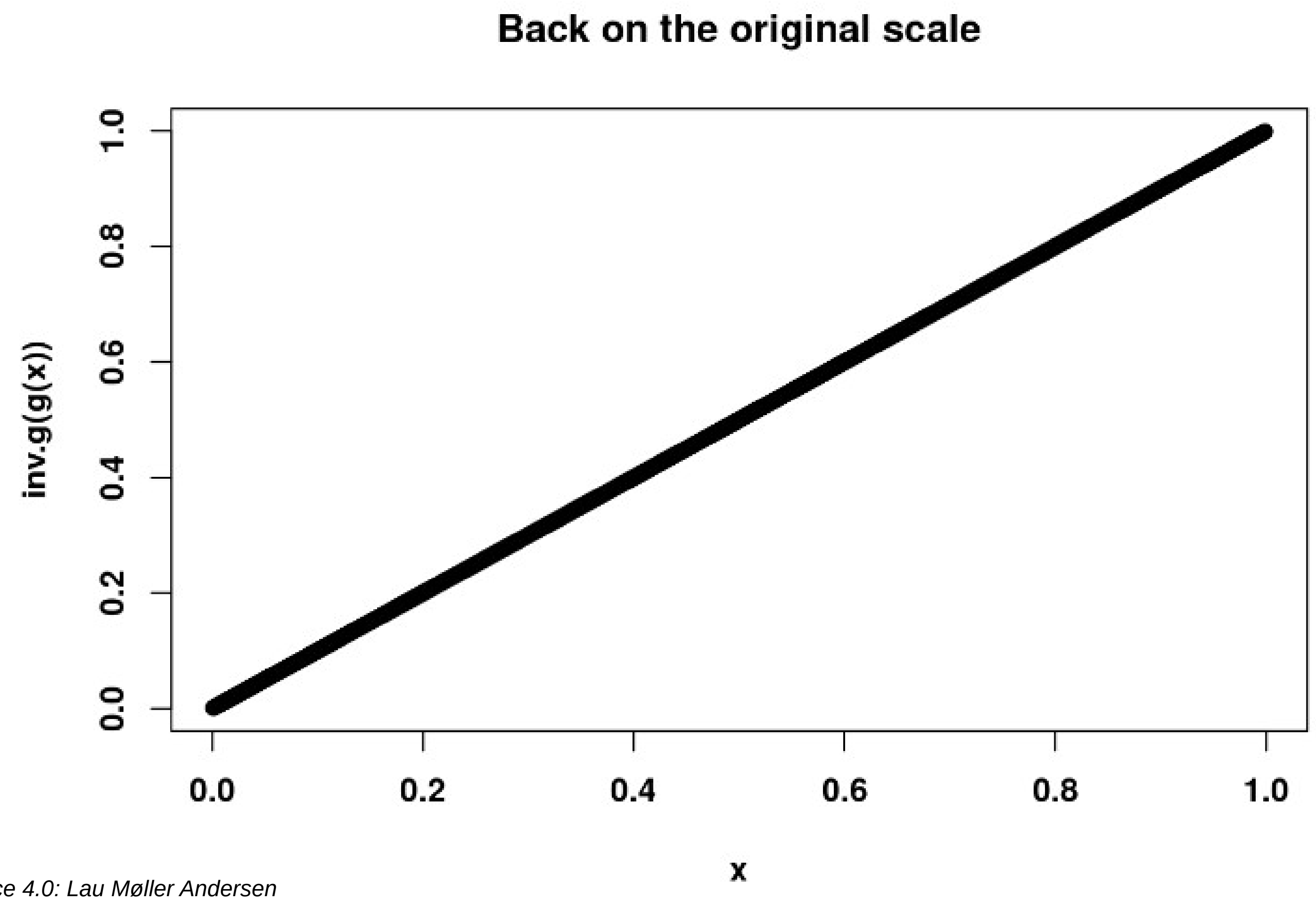
```r
x <- seq(0.001, 0.999, 0.001)
plot(x, main='Original probability data (on the range from 0-1)')
```



```r
plot(x, g(x), main='Log-it transformed, on the range from -Inf to Inf')
```



```r
plot(x, inv.g(g(x)), main='Back on the original scale')
```



These are the fitted values

```
y.hat <- inv.g(X %*% beta.hat)
print(head(y.hat))
```

```
##                          [,1]
## Mazda RX4           0.8172115
## Mazda RX4 Wag       0.6157283
## Datsun 710          0.9373069
## Hornet 4 Drive      0.2897304
## Hornet Sportabout 0.1415972
## Valiant             0.1320944
```

```
print(head(y.hat - logistic.model$fitted.values))
```

```
##                   [,1]
## Mazda RX4            0
## Mazda RX4 Wag        0
## Datsun 710           0
## Hornet 4 Drive       0
## Hornet Sportabout    0
## Valiant              0
```

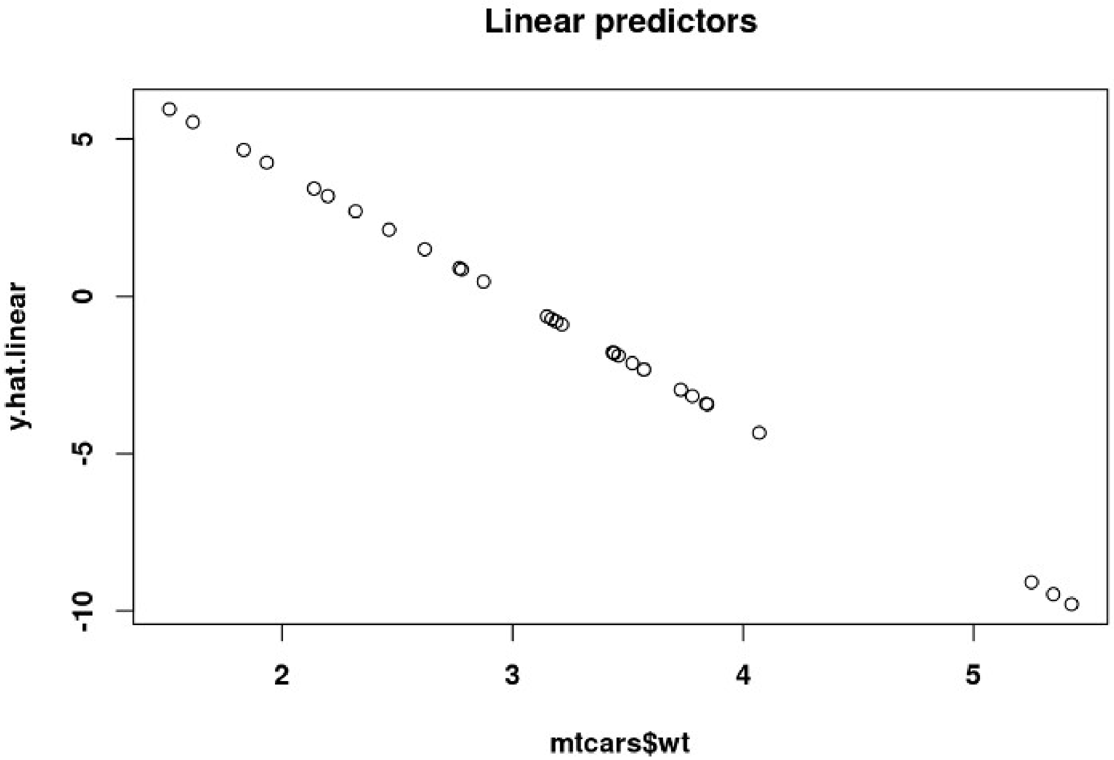
These are the linear predictors

```
y.hat.linear <- X %*% beta.hat

print(head(y.hat.linear - logistic.model$linear.predictors))
```

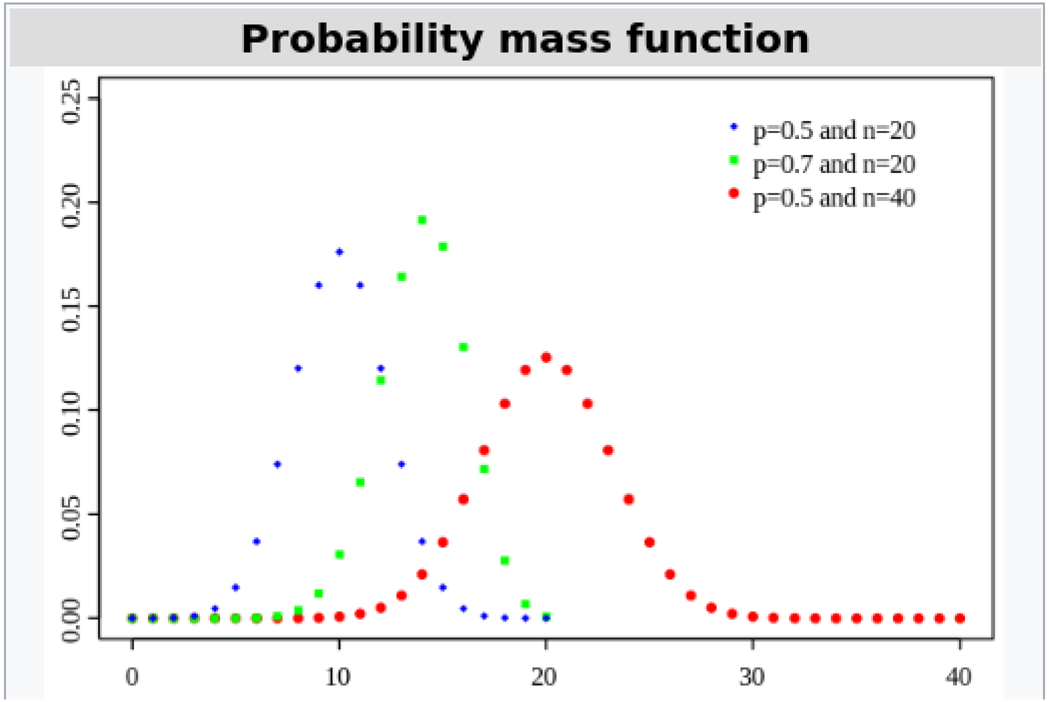
```
##                    [,1]
## Mazda RX4             0
## Mazda RX4 Wag         0
## Datsun 710            0
## Hornet 4 Drive        0
## Hornet Sportabout     0
## Valiant               0
```

Looks like a “normal” linear regression



4) A data distribution: $p(y|\hat{y})$

Binomial distribution



$$\Pr(y=1)=\hat{y}$$

https://en.wikipedia.org/wiki/Binomial_distribution

Some link functions

Usage

```
family(object, ...)

binomial(link = "logit")
gaussian(link = "identity")
Gamma(link = "inverse")
inverse.gaussian(link = "1/mu^2")
poisson(link = "log")
quasi(link = "identity", variance = "constant")
quasibinomial(link = "logit")
quasipoisson(link = "log")
```

Important difference from the general linear model

$$\xcancel{\hat{\beta} = (X^T X)^{-1} X^T Y \ \textbf{(least squares solution)}}$$

The least squares solution is the maximum likelihood estimation

Important difference from the general linear model

We also make maximum likelihood estimates for logistic regression, but there are no analytical solutions for those

Maximum likelihood estimate

Likelihood: $L(\theta \mid O) = \prod_{i=1}^{n} fx(x_i \mid \mu, \sigma^2)$

where μ and σ^2 are parameters describing a normal distribution

θ: the unknown parameters, e.g. $\hat{\beta}$ and $\hat{\sigma}^2$

O: the observations from a given sample

log-likelihood: $l(\theta \mid 0) = \ln(L(\theta \mid O))$

$MLE: \hat{\theta} = \arg\max l(\theta \mid O)$

The general linear mixed model (GLMM)

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}$$

$y: N\ x\ 1$ column vector

$X: N\ x\ p$ matrix of p predictor variables

$\beta: p\ x\ 1$ column vector of the first level regression coefficients

$Z: N\ x\ q$ design matrix for the q random effects

$u: q\ x\ 1$ column vector of the second-level effects

$\epsilon: N\ x\ 1$ column vector of the residuals

To generalize to non-linear functions At least four ingredients needed

1) A data vector: $y = (y_1, \ldots, y_n)$
2) Predictors: X and coefficients β, forming a linear predictor $X\beta$
3) A *link function* g: yielding a vector of transformed data $\hat{y} = g^{-1}(X\beta)$ that are used to model the data
4) A data distribution: $p(y \mid \hat{y})$

$$\left(\boldsymbol{X\beta = \beta_{0j} + X_{1j}\beta_{1j} + \ldots + X_{kj}\beta_{kj}}\right)$$

This time a $_j$ added to indicate that all of these are modelled at a second level as well

Did you learn?

Generalized Linear Mixed Effects Models (GLMM)

1) Understanding that we can extend the scope of our multilevel modelling by using appropriate link functions and data distributions

2) Understanding the multilevel equivalent of the GLM

References

- Gelman, A., Hill, J., 2006. Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press.