Lecture 10.3
Mixed-effects models

Nicola Romanò
nicola.romano@ed.ac.uk
13-11-2019

浙江大学爱丁堡大学联合学院
ZJU-UoE Institute

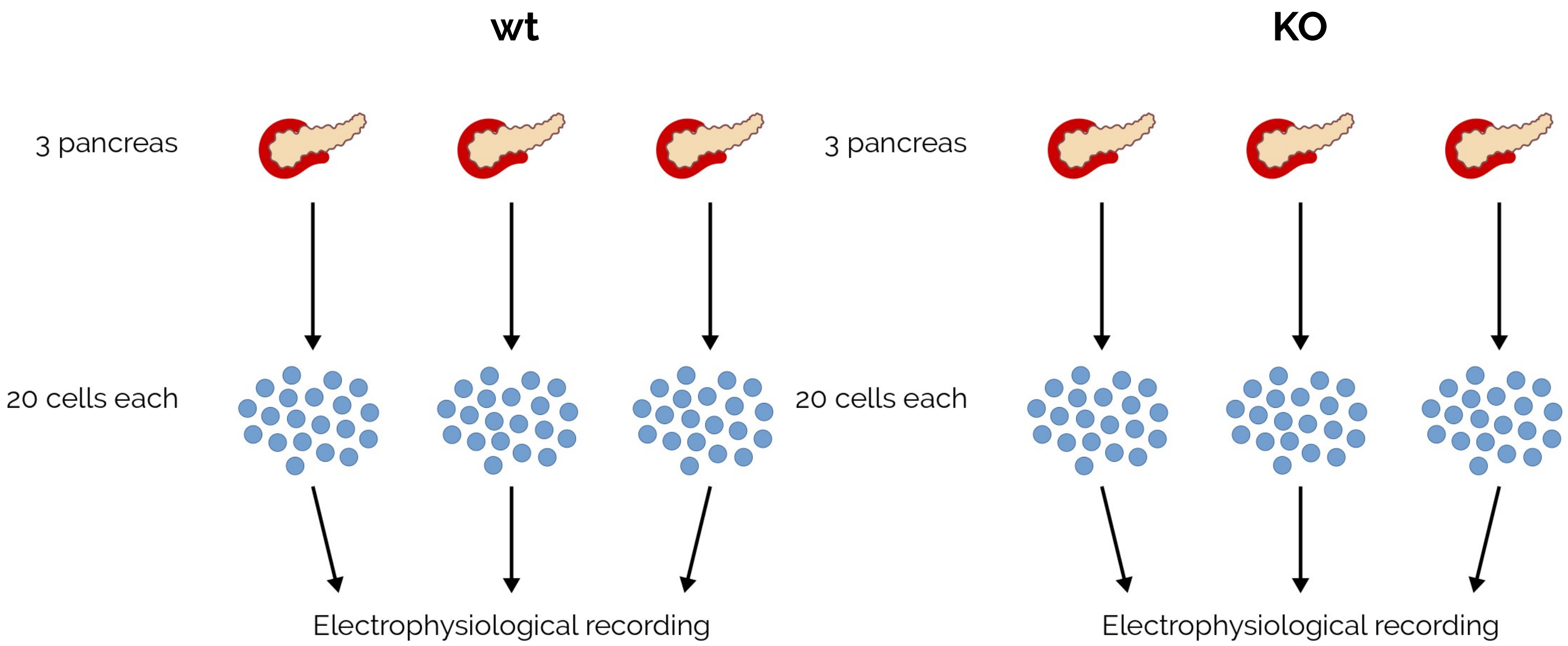
Learning objectives

At the end of this lecture you should be able to:

- Identify fixed and random effects in an experimental scenario
- Describe the problem associated with ignoring random effects or considering them as fixed
- Give a basic explanation of what a mixed effect model is and code it in R*
- Interpret the output of a mixed effect model*

* These last two points will be complemented by what you will learn in the next R workshop.

Nested design



wt

KO

3 pancreas

3 pancreas

20 cells each

20 cells each

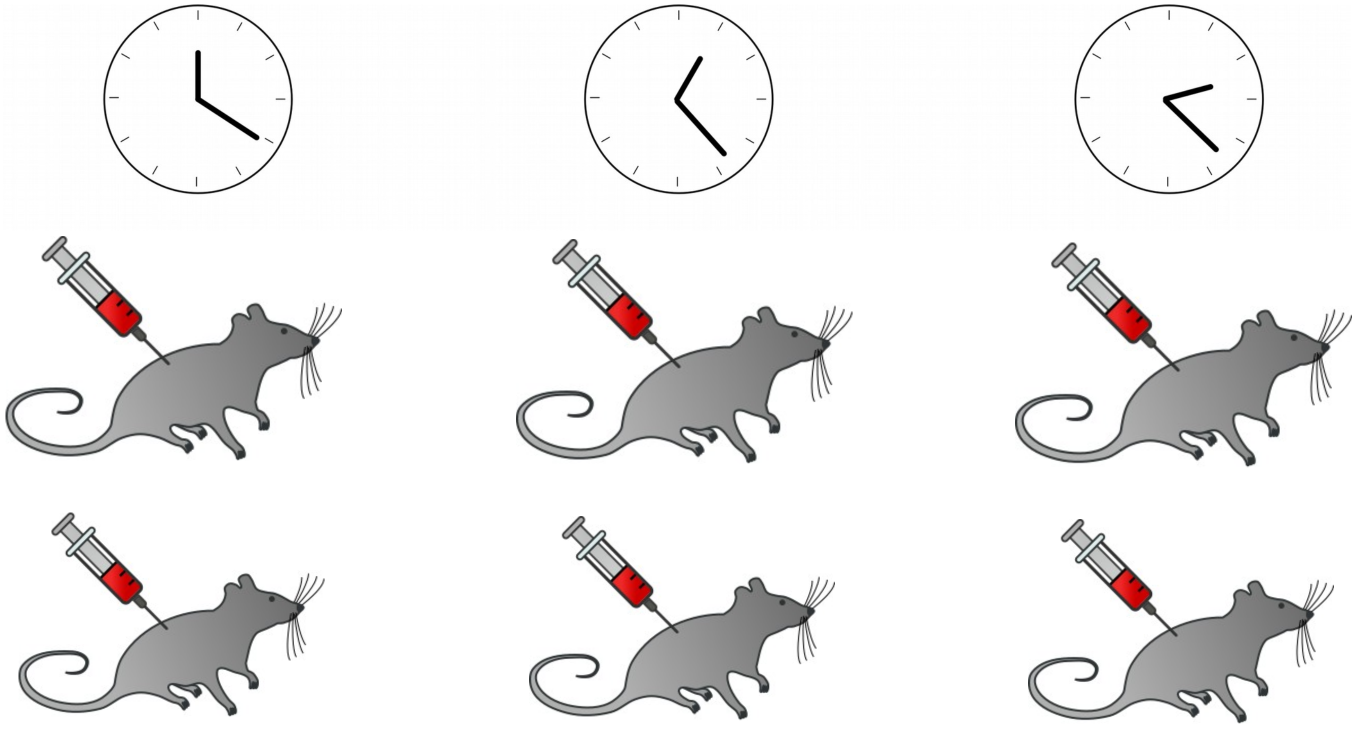
Electrophysiological recording

Electrophysiological recording

n = 60 cells from 3 pancreas. Record membrane potential.

What factors are influencing the response?

Repeated measures design



Males (n = 10)

Females (n = 10)

n = 20 mice (10 M, 10 F). Record glucose level over time.

What factors are influencing the response?

Fixed and random effects

Two types of factors influence our outcome:

- Factors that we want to study ← **Fixed effects**
- “Nuisance” factor ← **Random effect**

The linear models we have seen so far only contain fixed effects.

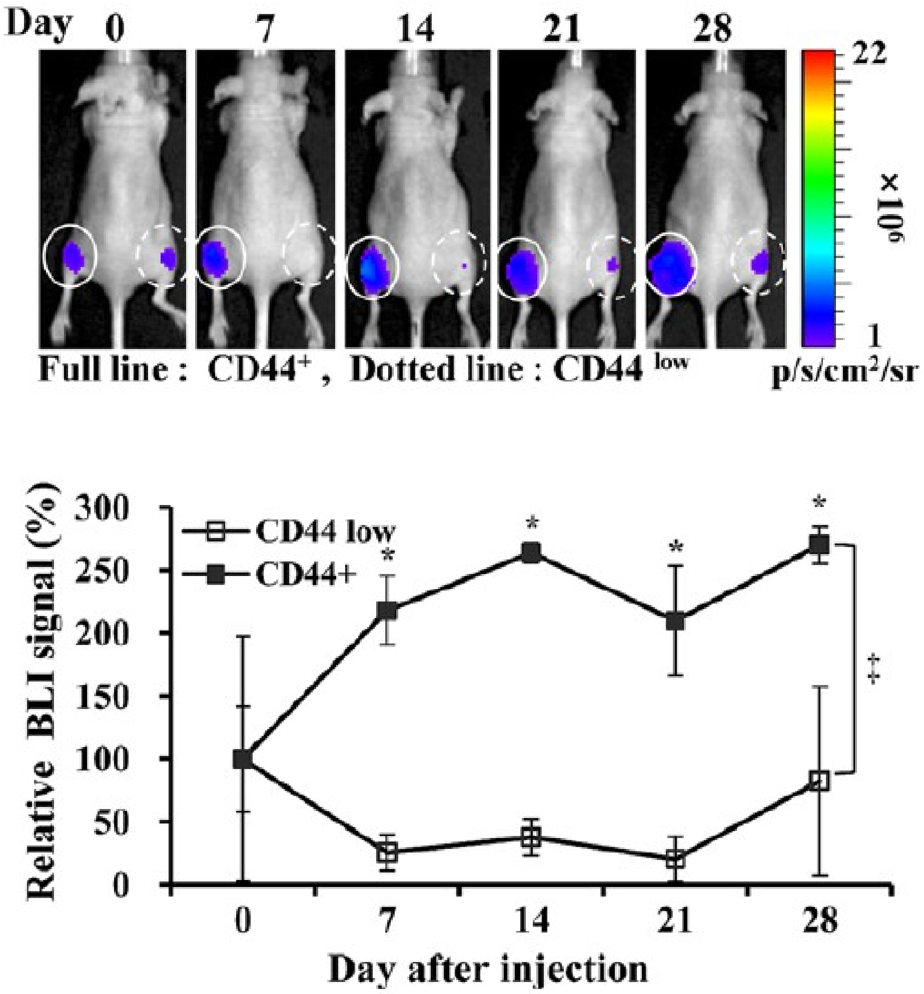
Random effects are also contributing to data variance, but depend on the specific sample, so estimating coefficients for them is not useful.

A model containing only fixed effects is called a **fixed effect model**

A model containing only random effects is called a **random effect model**

A **mixed effect model** contains both fixed and random effects.

Fixed and random effects



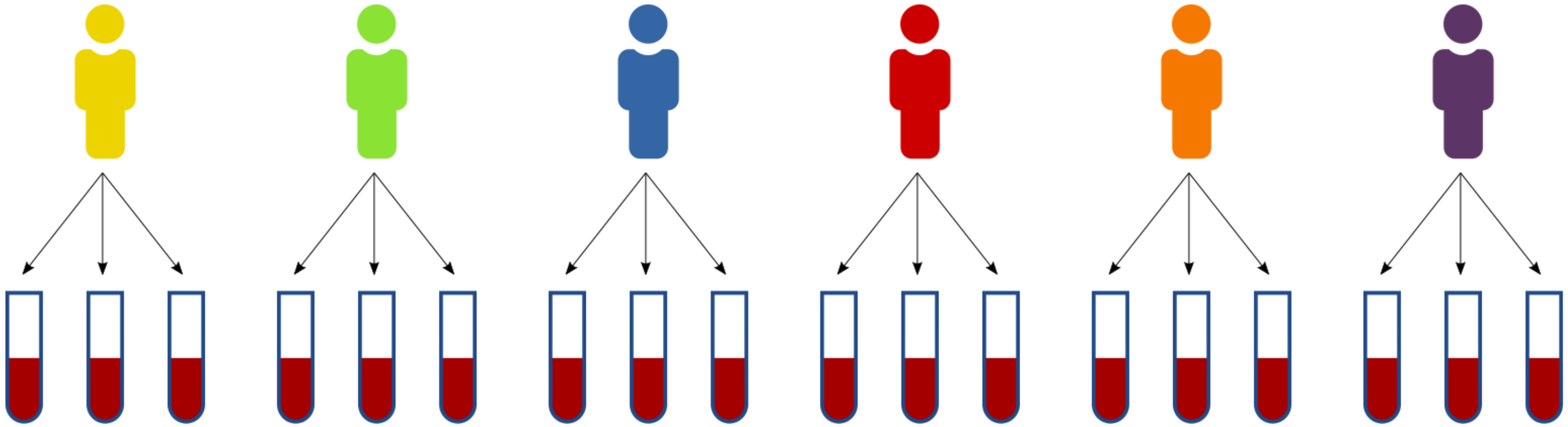
Kim and colleagues implanted two different types of cancer cell (genetically modified so that they emit luminescence) in the left and right mammary gland of mice.

They then imaged the mice over four weeks to determine the growth of the tumoral mass.

What are the fixed and the random effects?

A simple example

I measure the level of thyroxine in six individuals, in three different occasions.



Let’s ignore the nested structure of the samples and treat them as independent

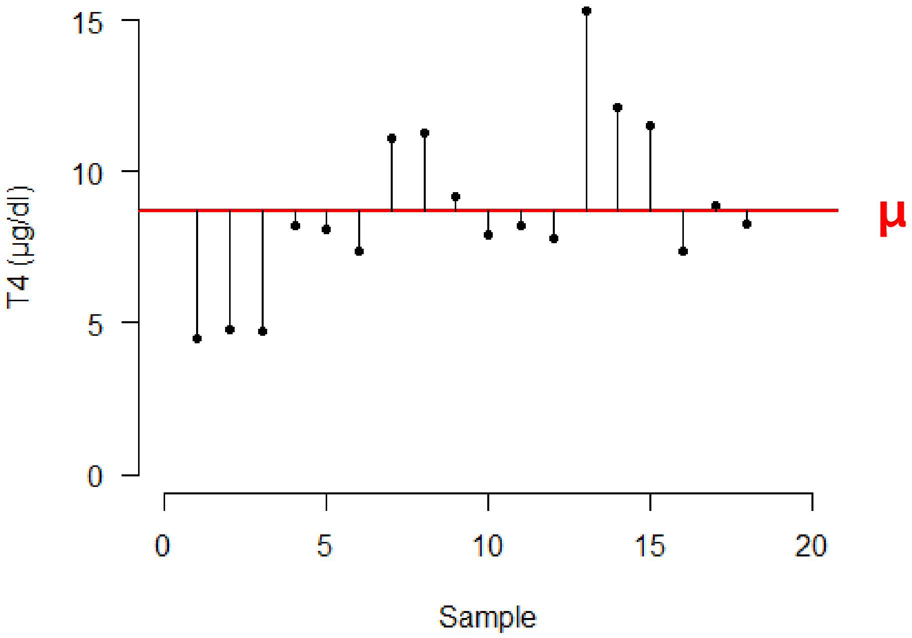
Important: we are only doing this to explain why mixed effects models are useful, you should NOT ignore nesting!

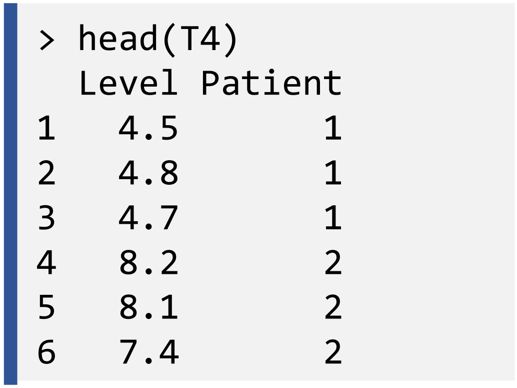
A simple example

For each sample i we measure the level of thyroxine, T_i

$$T_i = \mu + \varepsilon_i, \qquad i = 1, \dots, 18, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

We assume that ε_i are independent errors, coming from a normal distribution





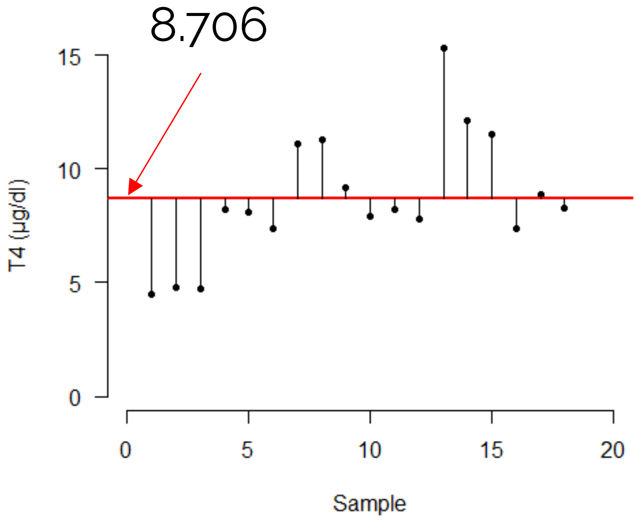
```
> head(T4)
  Level Patient
1   4.5       1
2   4.8       1
3   4.7       1
4   8.2       2
5   8.1       2
6   7.4       2
```

A simple example

Using the linear model notation we are familiar with

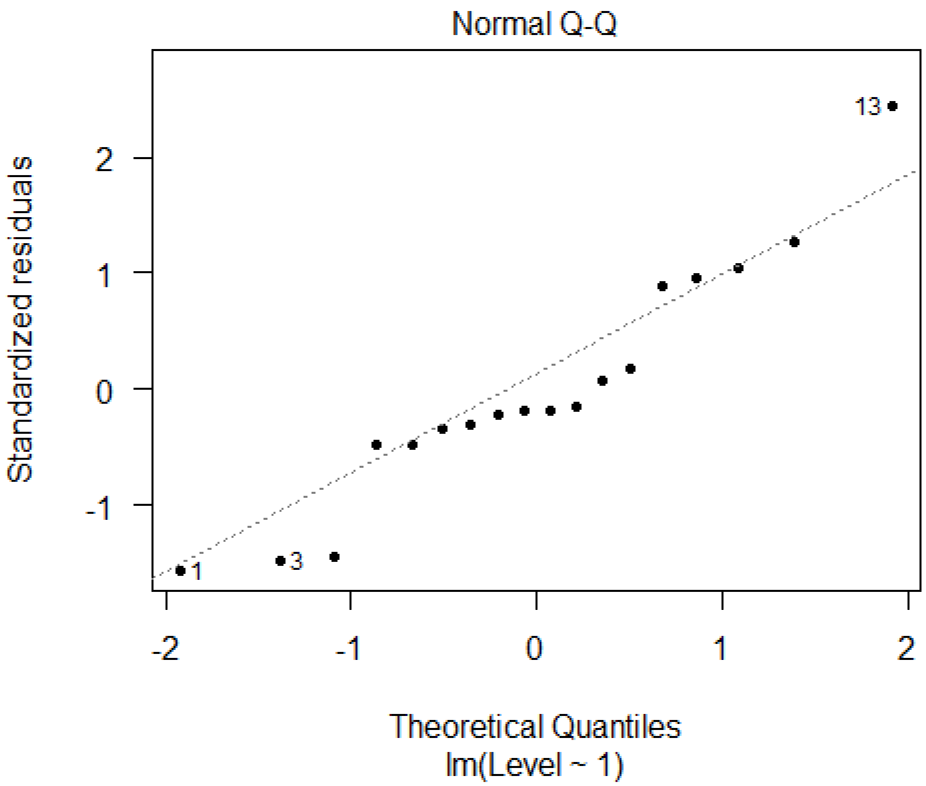
$$T = \beta_0 + \varepsilon$$

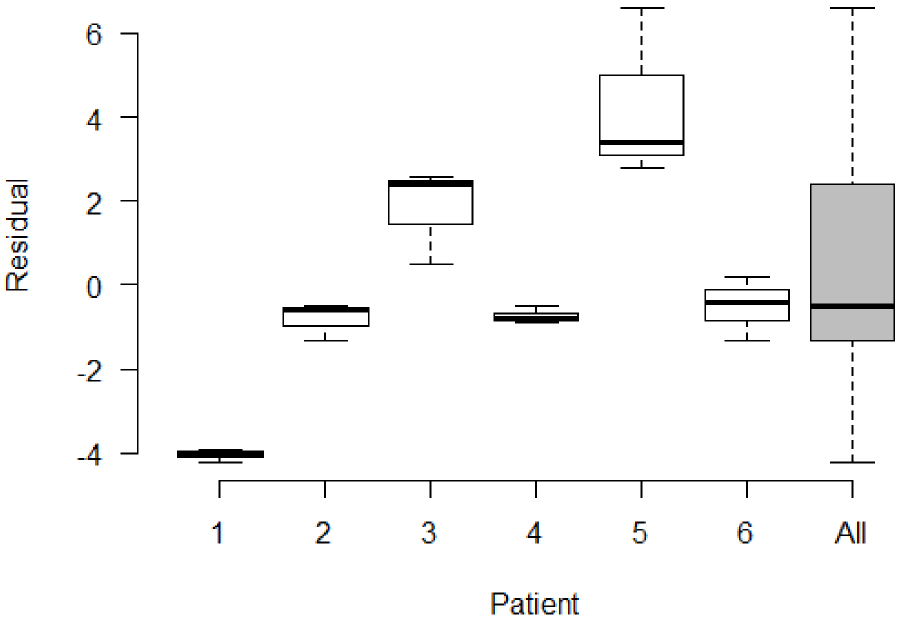
This is a model with no parameter (only intercept and error term)



```
# ~ 1 means we only model the intercept
model <- lm(Level ~ 1, data = T4)
summary(model)

Call:
lm(formula = Level ~ 1, data = T3)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.2056 -1.2056 -0.5056  1.9194  6.5944 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)    8.706      0.653   13.33 1.98e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 
0.1 ‘ ’ 1

Residual standard error: 2.77 on 17 degrees of
freedom
```

The problem is…



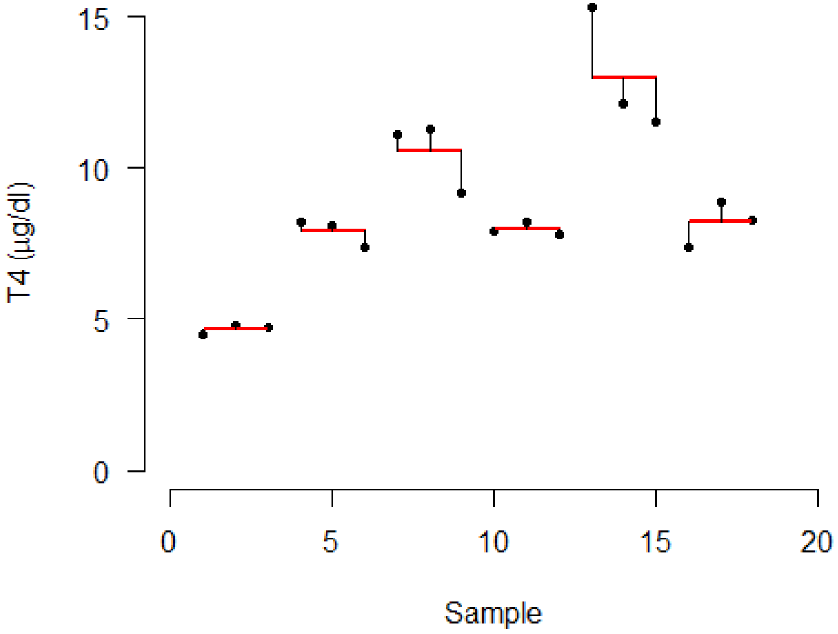


The “patient effect” has been incorporated in the residuals

We have artificially increased the within-patient variability

Improving the model

We could incorporate the patient in our model as a fixed effect. This will be represented as a factor with 6 levels, 1 to 6.



$$T_{ij} = \mu_i + \varepsilon_{ij}$$

$$i = 1, \dots, 6, \quad j = 1, 2, 3, \quad \varepsilon_{ij} \sim N(0, \sigma^2)$$

or

$$T = \beta_0 + \beta_1 P_2 + \beta_2 P_3 + \cdots + \varepsilon$$

Where P2 to P6 are dummy variables to represent the 6-level factor "patient"

$$P_2 = \begin{cases} 1, if\ Patient = 2 \\ 0, \qquad otherwise \end{cases}; \quad P_3 = \begin{cases} 1, if\ Patient = 3 \\ 0, \qquad otherwise \end{cases}; \quad \dots$$

Fixed-effects model

```
model.fix <- lm(Level ~ Patient, data = T4)
summary(model.fix)

Call:
lm(formula = Level ~ Patient, data = T4)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.46667 -0.41667  0.06667  0.28333  2.33333 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   4.6667     0.5936   7.861 4.50e-06 ***
Patient2      3.2333     0.8395   3.851  0.00230 ** 
Patient3      5.8667     0.8395   6.988 1.46e-05 ***
Patient4      3.3000     0.8395   3.931  0.00200 ** 
Patient5      8.3000     0.8395   9.886 4.05e-07 ***
Patient6      3.5333     0.8395   4.209  0.00121 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.028 on 12 degrees of freedom
Multiple R-squared:  0.9028,	Adjusted R-squared:  0.8622 
F-statistic: 22.28 on 5 and 12 DF,  p-value: 1.088e-05
```

Mean of pat. 1: 4.6667

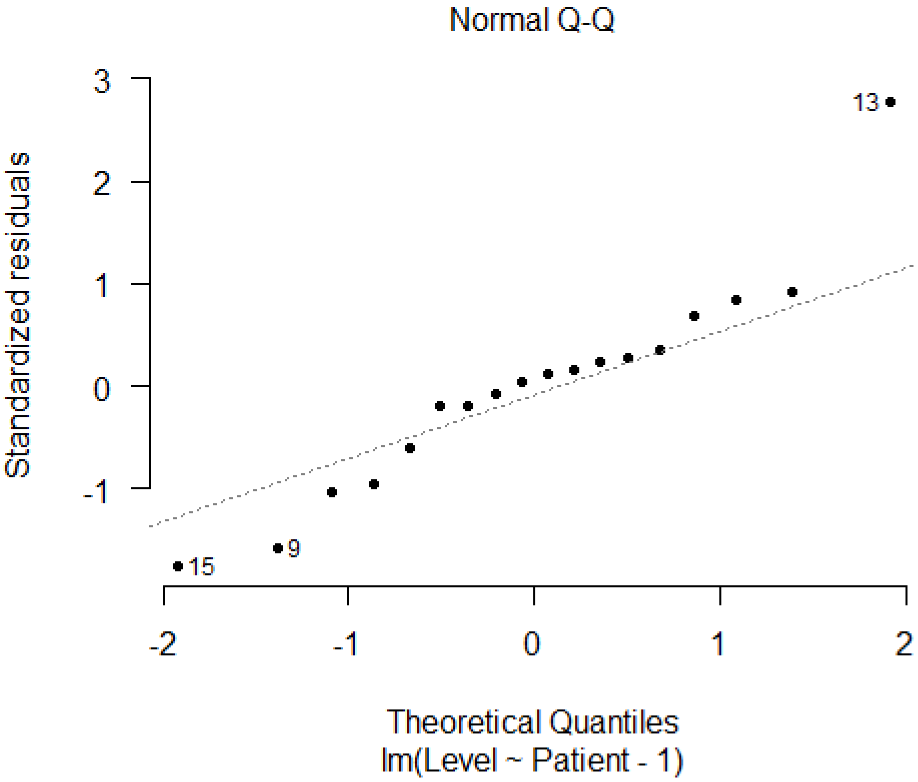
Mean of pat. 2: 4.6667 + 3.2333

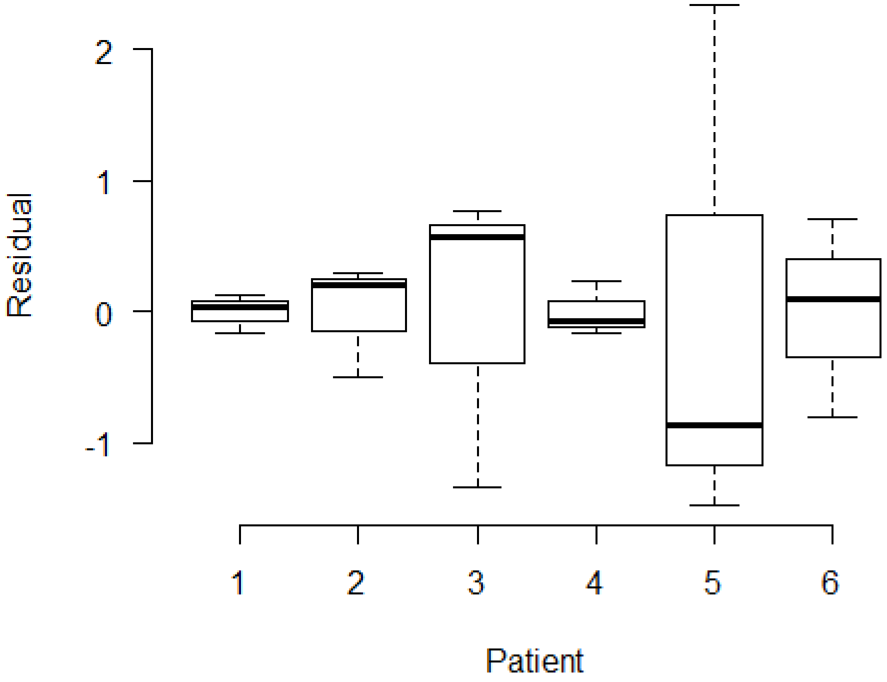
Mean of pat. 3: 4.6667 + 5.8667

…

~2.7 fold decrease!

Fixed-effects model residuals





The mean of the residuals is now centred around 0 for each patient!

Fixed-effects model

Is this model useful?

Not too much!

- Only models those 6 specific patients.
- Cannot extrapolate to a generic “patient 7”!
- Does not provide a measure of between-patient variability
- Needs 1 extra parameter for each new patient

Random effects

We can reparametrize our model and treat patients as random effects

$$T_{ij} = \mu + b_i + \varepsilon_{ij}, \qquad i = 1, \dots, 6, \qquad j = 1,2,3$$

$$b_i \sim N\left(0, \sigma_b^2\right), \qquad \varepsilon_{ij} \sim N(0, \sigma^2)$$

In this model, random effects

- are a **random variable**, drawn from a normal distribution
- are independent from the error (thus they have a different variance)

Observations coming from the same patient i share the same random effect b_i, and are therefore correlated, as they come from the same distribution.

Random effects model

```
library(nlme) # Needs to be installed first!

model.rnd <- lme(Level ~ 1, data = T4, random = ~ 1 | Patient)
summary(model.rnd)

Linear mixed-effects model fit by REML
 Data: T4
       AIC      BIC    logLik
  73.59904 76.09868 -33.79952
```

Goodness-of-fit measures

```
Random effects:
 Formula: ~1 | Patient
        (Intercept) Residual
StdDev:    2.738572 1.028213

Fixed effects: Level ~ 1
               Value Std.Error DF  t-value p-value
(Intercept) 8.705556  1.143983 12 7.609863       0

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.2404308 -0.4756779 -0.0718532  0.2411250  2.4553010

Number of Observations: 18
Number of Groups: 6
```

What about fixed effects?

Let's imagine we have other 6 patients, with a different genotype.

Total 12 patients, 6 genotype A, 6 genotype B

A mixed effect model would be:

$$T_{ij} = \mu + \beta_k + b_{ki} + \varepsilon_{kij}, \qquad k = 1, 2, \qquad i = 1, \dots, 6, \qquad j = 1,2,3$$

$$b_{ki} \sim N\left(0, \sigma_b^2\right), \qquad \varepsilon_{kij} \sim N(0, \sigma^2)$$

Mixed effects model

```
model.mix <- lme(Level ~ Genotype, data = T4, random = ~ 1 | Patient)
summary(model.mix)

Linear mixed-effects model fit by REML
 Data: T4
       AIC      BIC    logLik
  186.2225 192.3279 -89.11124

Random effects:
 Formula: ~1 | Patient
        (Intercept) Residual
StdDev:     5.37896 1.903953

Fixed effects: Level ~ Genotype
                Value Std.Error DF  t-value p-value
(Intercept)  8.705556  2.241337 24 3.884090  0.0007
GenotypeB   12.811719  3.169729 10 4.041897  0.0024
 Correlation:
          (Intr)
GenotypeB -0.707

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.87642275 -0.43445309 -0.03279226  0.33914381  3.26399420

Number of Observations: 36
Number of Groups: 12
```

Random intercepts…

We have coded the random effect as: `~ 1 | Patient`

This means to assign a different intercept (indicated by the `1`) to each patient, whilst keeping the same slope.

Indeed:

```
coef(model.mix)

   (Intercept) GenotypeB
1     4.828582  12.81172
2     7.932294  12.81172
3    10.460059  12.81172
4     7.996288  12.81172
5    12.795843  12.81172
6     8.220267  12.81172
7    -1.021683  12.81172
8     7.202284  12.81172
9    11.553030  12.81172
10    7.138611  12.81172
11   20.184906  12.81172
12    7.176186  12.81172
```

… and random slopes

We could have specified the random effect as: `~ Genotype | Patient`

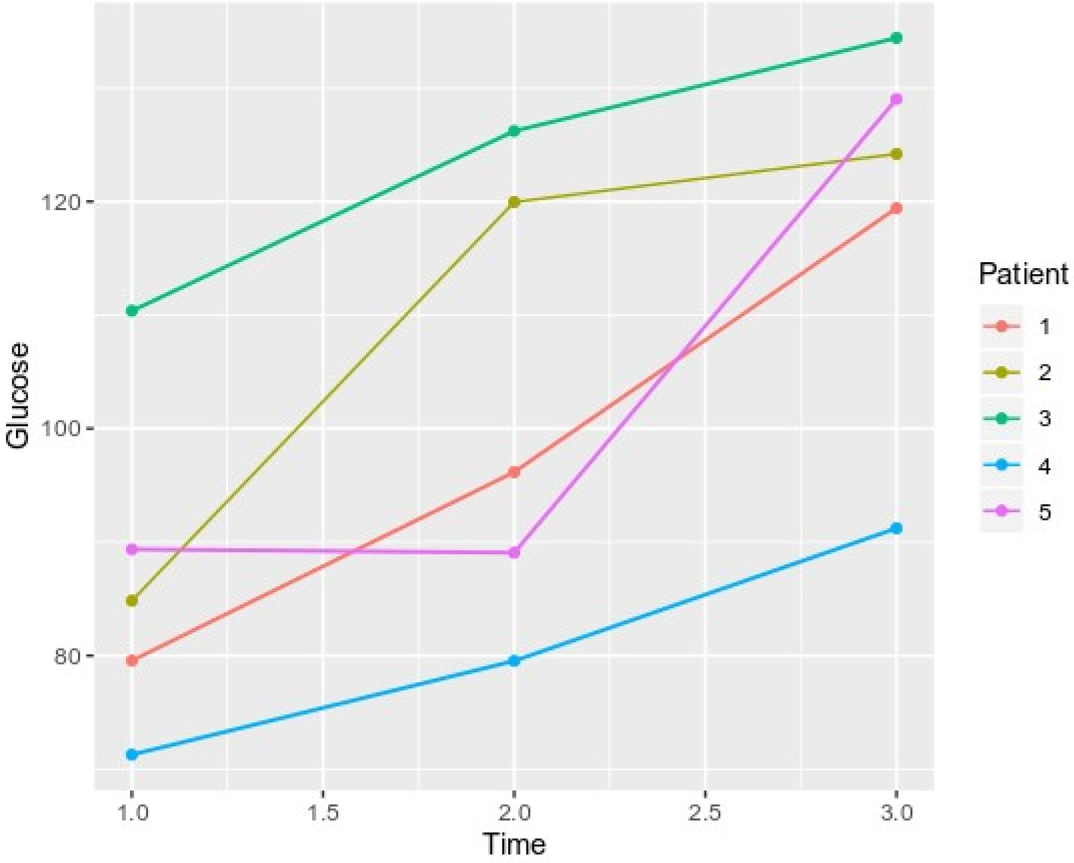
This would assign a different slope and intercept to each patient
(I could use `~ Genotype -1 | Patient` to only assign a different slope)

Indeed:

```
coef(model.mix)

   (Intercept) GenotypeB
1     5.288198  10.08194
2     8.023964  12.26726
3    10.252063  14.04707
4     8.080372  12.31232
5    12.310939  15.69169
6     8.277798  12.47003
7     6.395764   5.22157
8     8.348595  11.63872
9     9.381706  15.03360
10    8.333475  11.58903
11   11.431396  21.76904
12    8.342398  11.61835
```

A final example

Increase in glucose levels after a meal.
15 observations from 5 subjects measured 3 times.



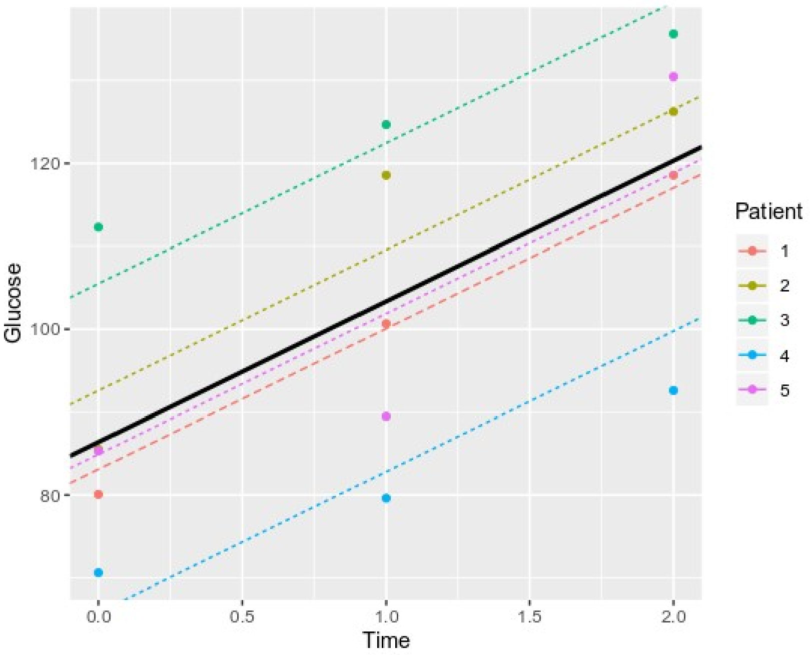
Random effects model

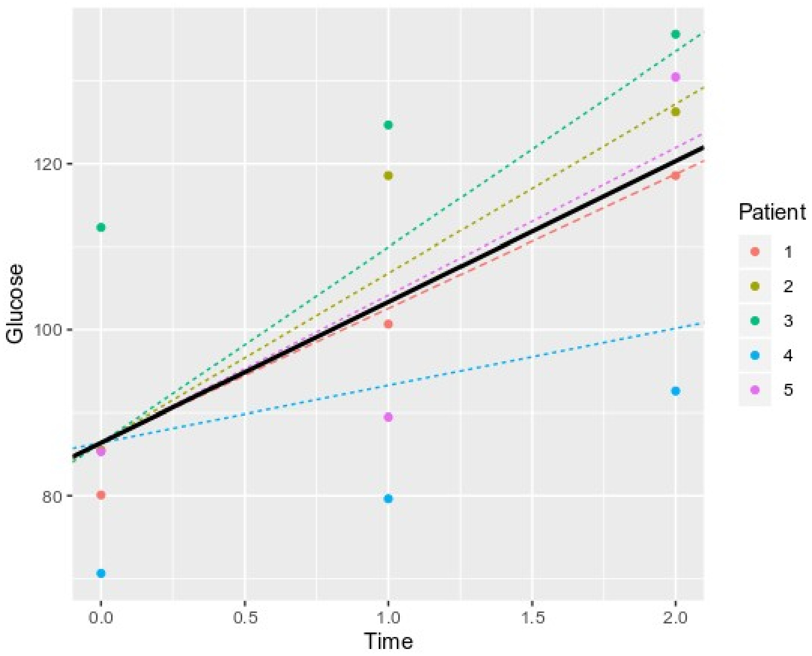
Random intercept (baseline glucose level) differs for each patient. Slope (rate of glucose change) is the same.

```
random = ~ 1 | Patient
```

Random intercept (baseline glucose level) is the same for each patient. Slope (rate of glucose change) changes

```
random = ~ Time - 1 | Patient
```





What about nested designs?

Similar syntax, but nesting should be indicated in the random effect, using `/`

For example, analysing the grades of different classes in different schools we may use `random = ~ 1 | School/Class`

For a much more in depth explanation and examples see R workshop #4, online on Friday!