广义混合模型：其中心思想在于通过在线性预测项中引入随机效应，来体现同一对象 （“目标”组） 内数据的相关性和不同对象 （“目标”组） 内的异质性，这里的随机效应变量的分布是正态分布。
引入的原因是实际样本之间并不像假设的那样相互独立，而是往往样本间存在一定的相关性，如聚类关系或纵向关系。
聚类关系指不同的样本可能具有某种聚类关系，如来自同一个地区，具有同种信仰，而另一些样本来自另一个地区，具有另一种信仰等。
纵向关系指不同的样本来自与同一个采集个体，但只是采集时间不一致罢了。

这两种情况都会造成样本之间并非相互独立，因此要引入随机效应项来模拟替代这种相关关系。

A beginner’s guide to lmer

包lme4

This is just a basic introduction to `lmer` syntax for multilevel regression. As an example,

we’ll analyze the effect of different diets on chick growth. A typical call to `lmer` looks

something like this

```
> str(ChickWeight)
Classes ‘nfnGroupedData’, ‘nfGroupedData’, ‘groupedData’ and 'data.frame':      578 obs. of  4 variables:
 $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
 $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
 $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
 $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
```

这里的1和固定效应里的1一样表示拟合常数部分（截距），这里可以省略不写，有他无他一

"|"符号前面是影响斜率的随机变量，符号后面是影响截距的随机变量，多个变量之间用"+"号进行隔离。

这里将 Time 和 Chick 当作了随机效应，他们影响了模型的斜率和截距。

```
m <- lmer(weight ~ Time * Diet + (1 + Time | Chick), data=ChickWeight, REML=F)
```

1.The first argument is the model formula. The part to the left of the `~` is the outcome

variable (in this case `weight`, the chick weight in gm). The part to the right of the `~` has two

components:

> The fixed effects: `Time * Diet` which is a compact way of specifying all simple effects and interactions of time (number of days since birth) and diet. A less compact but more explicit way to writing that would be `Time + Diet + Time:Diet`

2.The random effects: `(1 + Time | Chick)` which allows individual chicks to vary

randomly in terms of their intercept (starting weight) and their effect of `Time` (weight

change over time, also called a “random slope”, but I think that terminology can get confusing when fitting models with nonlinear predictors).

The second argument is the data frame. Note that this makes it easy to fit models to subsets of the data – if you wanted to ignore diet 3, you could specify `data = subset(ChickWeight, Diet != "3")`.

The last argument is optional. The default in `lmer` is to fit models using the REML (REstricted Maximum Likelihood) criterion. There are good reasons for this, but we often use the likelihood ratio test to compare models based on log-likelhoods, so we should use the Maximum Likelihood (ML) criterion.

To see the results of the model, we can use the `summary()` function:

```
summary(m)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: weight ~ Time * Diet + (1 + Time | Chick)
##    Data: ChickWeight
##
##      AIC      BIC   logLik deviance df.resid
##   4824.2   4876.5  -2400.1   4800.2      566
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7508 -0.5693 -0.0401  0.4694  3.5415
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Chick    (Intercept) 103.61   10.179
##           Time         10.01    3.165   -0.99
##  Residual             163.36   12.781
## Number of obs: 578, groups:  Chick, 50
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  33.6541     2.8023  12.009
## Time          6.2799     0.7303   8.598
## Diet2        -5.0205     4.8072  -1.044
## Diet3       -15.4038     4.8072  -3.204
## Diet4        -1.7475     4.8145  -0.363
## Time:Diet2    2.3293     1.2508   1.862
## Time:Diet3    5.1430     1.2508   4.112
## Time:Diet4    3.2528     1.2515   2.599
##
## Correlation of Fixed Effects:
##            (Intr) Time   Diet2  Diet3  Diet4  Tm:Dt2 Tm:Dt3
## Time       -0.881
## Diet2      -0.583  0.513
## Diet3      -0.583  0.513  0.340
## Diet4      -0.582  0.513  0.339  0.339
## Time:Diet2  0.514 -0.584 -0.882 -0.300 -0.299
## Time:Diet3  0.514 -0.584 -0.300 -0.882 -0.299  0.341
## Time:Diet4  0.514 -0.584 -0.300 -0.300 -0.882  0.341  0.341
```

Sometimes we just want the fixed effect parameter estimates (coefficients), which we can get by using `coef()`:

```
coef(summary(m))
```

```
##              Estimate Std. Error    t value
## (Intercept)  33.654113  2.8023022 12.0094514
## Time          6.279858  0.7303499  8.5984243
## Diet2        -5.020517  4.8071843 -1.0443779
## Diet3       -15.403787  4.8071843 -3.2043263
## Diet4        -1.747532  4.8145392 -0.3629697
```

```
## Time:Diet2    2.329278  1.2507884  1.8622481

## Time:Diet3    5.143013  1.2507884  4.1118171

## Time:Diet4    3.252804  1.2515387  2.5990434
```

One of the most challenging parts of fitting multilevel models is figuring our the right random effects. To understand the random effects, it can be helpful to look at the estimated variance and covariance block from the summary:

```
VarCorr(m)
```

```
##  Groups   Name        Std.Dev. Corr

##  Chick    (Intercept) 10.1789

##           Time        3.1645  -0.986

##  Residual             12.7811
```

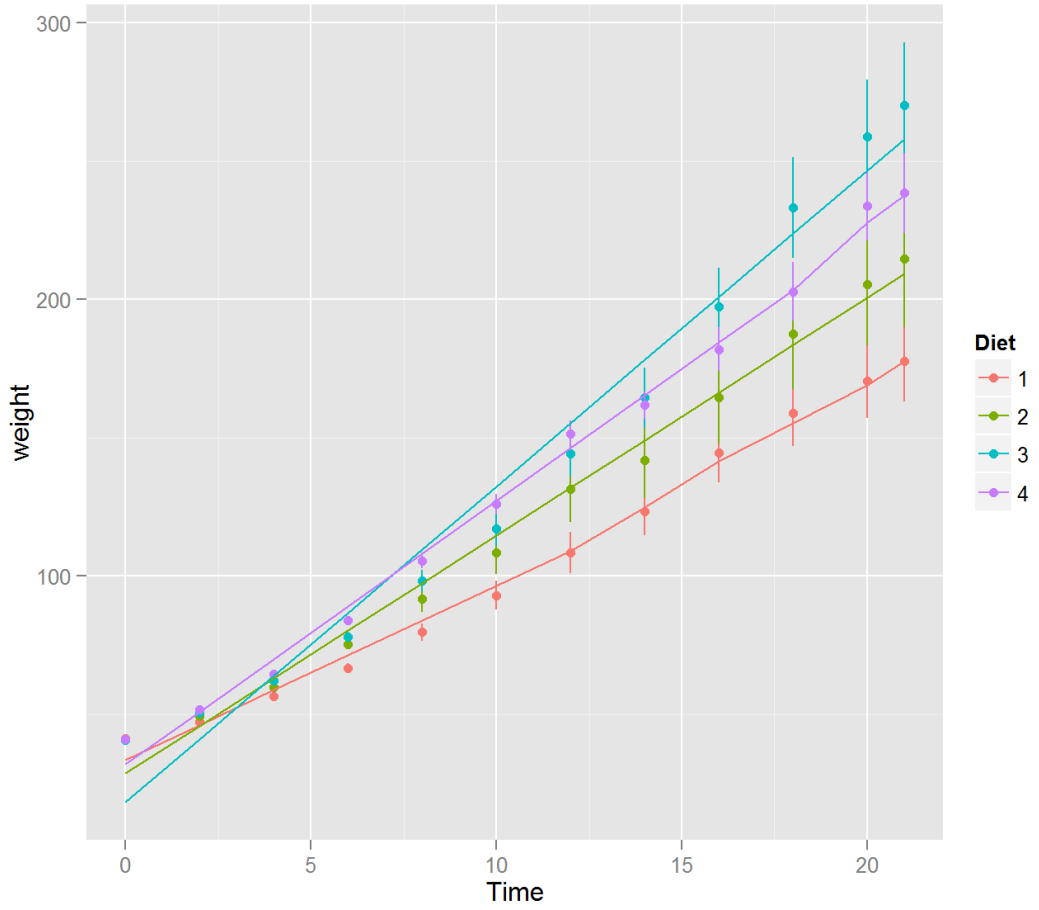
and we can get the actual random effect estimates using `resid(m)`

```
#to save space, only showing a summary of the structure and the first few value
s

str(resid(m))
```

```
##  Named num [1:578] 13.18 6.75 -0.68 -11.11 -14.54 ...

##  - attr(*, "names")= chr [1:578] "1" "2" "3" "4" ...
```

To make a (relatively) simple summary plot with model fit, use `fortify()` within `ggplot`:

```
ggplot(fortify(m), aes(Time, weight, color=Diet)) +

  stat_summary(fun.data=mean_se, geom="pointrange") +

  stat_summary(aes(y=.fitted), fun.y=mean, geom="line")
```


300
200
100
weight
0
5
10
15
20
Time
Diet
1
2
3
4