Lecture8 Diagnostics

Witold Wolski

March 29, 2017

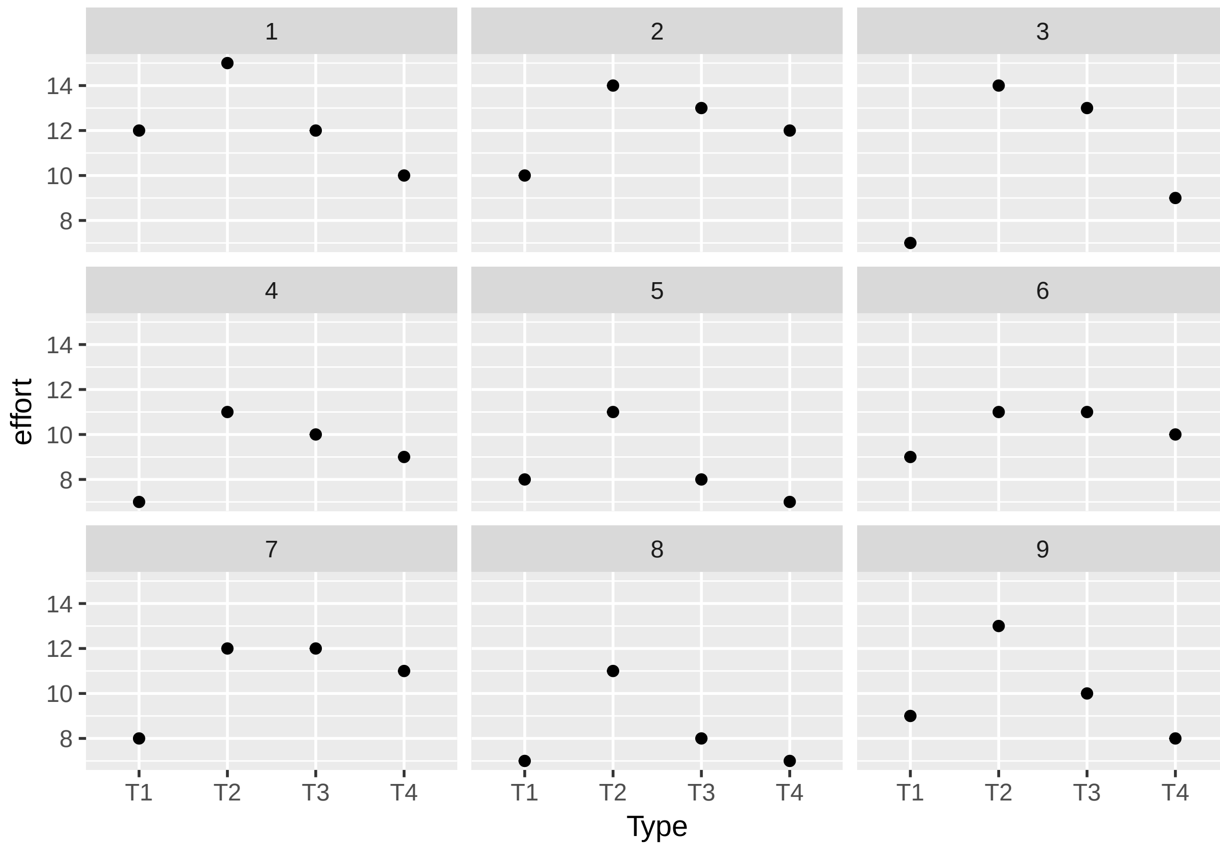
```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(ggplot2)
```

Start with a plot of the data, to look for outliers

```
load('MAS473.RData')
attach(ergoStool)
qplot(Type, effort, facets=~Subject,data=ergoStool)
```




```
fm1<-lmer(effort~Type -1 + (1|Subject),ergoStool)
```

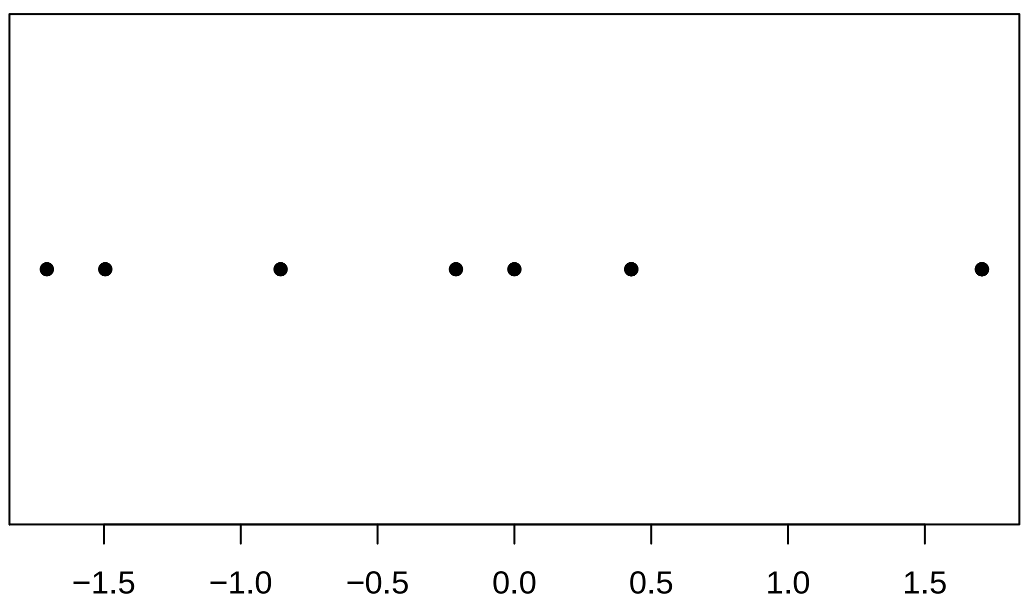
3.5.6 assessing the fitted model

Expected values of random effects

```r
ranef(fm1)
```

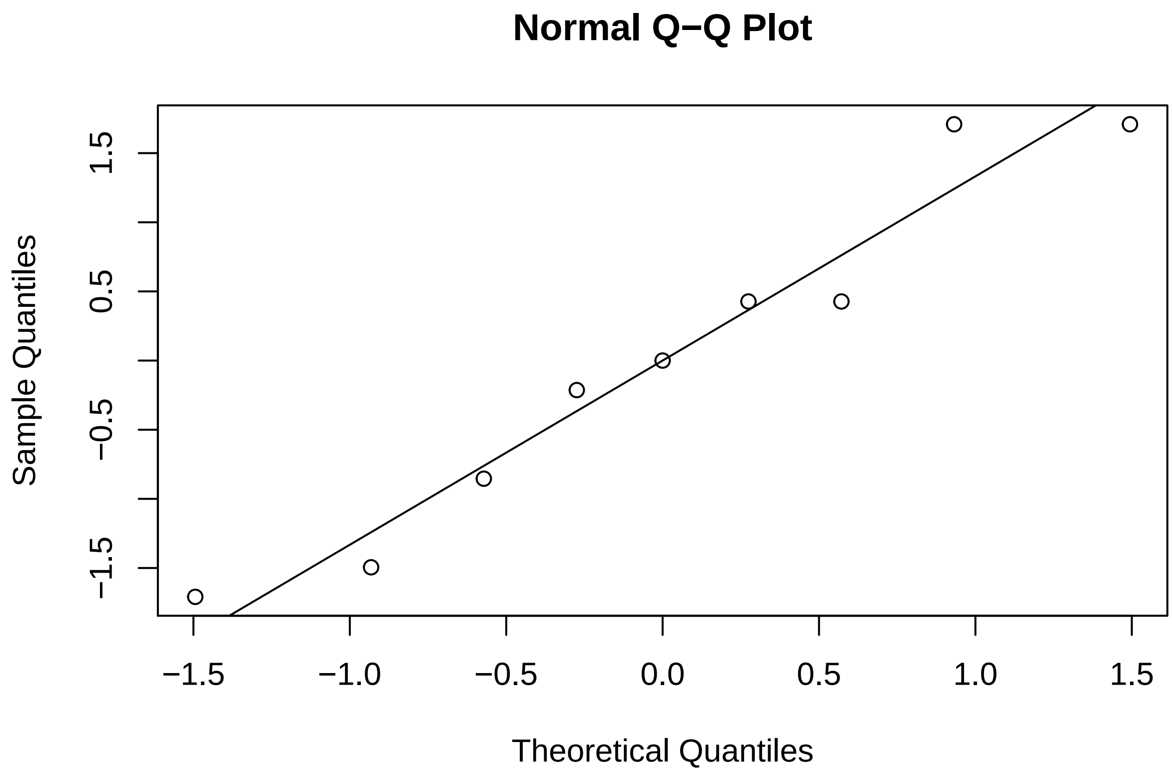
```
## $Subject
##   (Intercept)
## 1   1.7087162
## 2   1.7087162
## 3   0.4271791
## 4  -0.8543581
## 5  -1.4951267
## 6   0.0000000
## 7   0.4271791
## 8  -1.7087162
## 9  -0.2135895
```

```r
plot(ranef(fm1)$Subject, pch=16)
```



Check assumption that random effects are normally distributed

```r
qqnorm(unlist(ranef(fm1)))
abline(0,1.332) # Reference gradient is estimated standard deviation of random effect
```

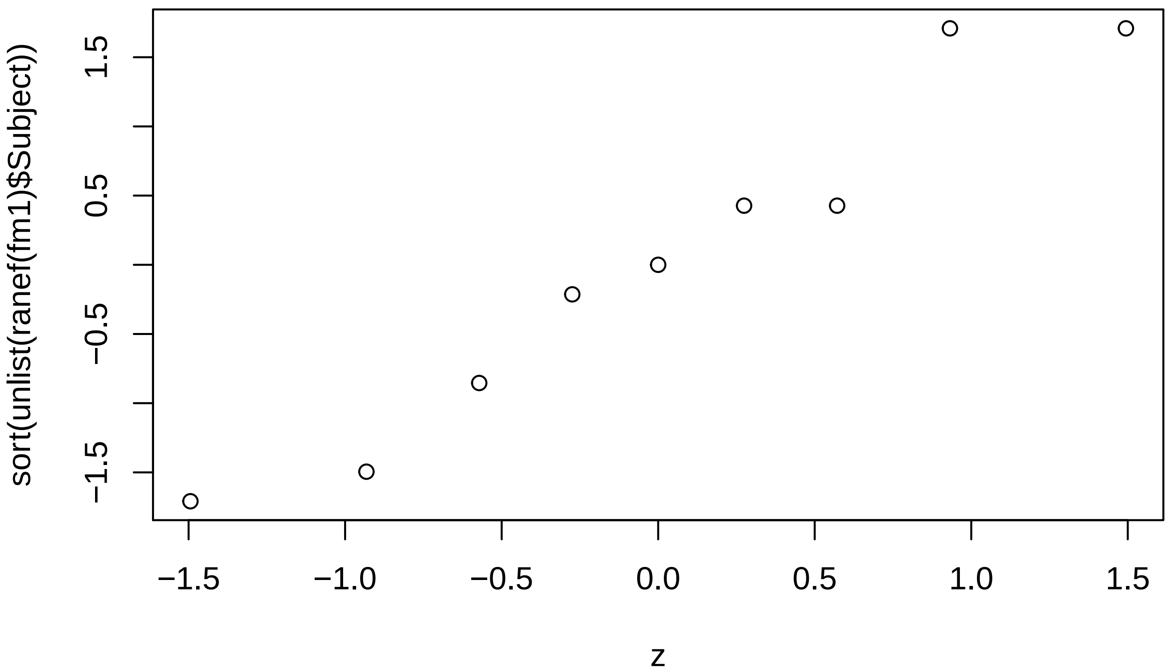


For background, R uses these quantiles for the QQ plot Do ?ppoints for more details

```
(qu<-(1:9 - 3/8)/(9 + (1-3/8)-3/8))
```

```
## [1] 0.06756757 0.17567568 0.28378378 0.39189189 0.50000000 0.60810811
## [7] 0.71621622 0.82432432 0.93243243
```

```
z<-qnorm(qu)
plot(z,sort(unlist(ranef(fm1)$Subject)) )
```



Fitted values

```r
fitted(fm1)
```

```
##         1         2         3         4         5         6         7
## 10.264272 14.153161 12.486494 10.930938 10.264272 14.153161 12.486494
##         8         9        10        11        12        13        14
## 10.930938  8.982735 12.871624 11.204957  9.649401  7.701197 11.590086
##        15        16        17        18        19        20        21
##  9.923420  8.367864  7.060429 10.949318  9.282651  7.727096  8.555556
##        22        23        24        25        26        27        28
## 12.444444 10.777778  9.222222  8.982735 12.871624 11.204957  9.649401
##        29        30        31        32        33        34        35
##  6.846839 10.735728  9.069062  7.513506  8.341966 12.230855 10.564188
##        36
##  9.008633
```

Residuals (not standardaised)

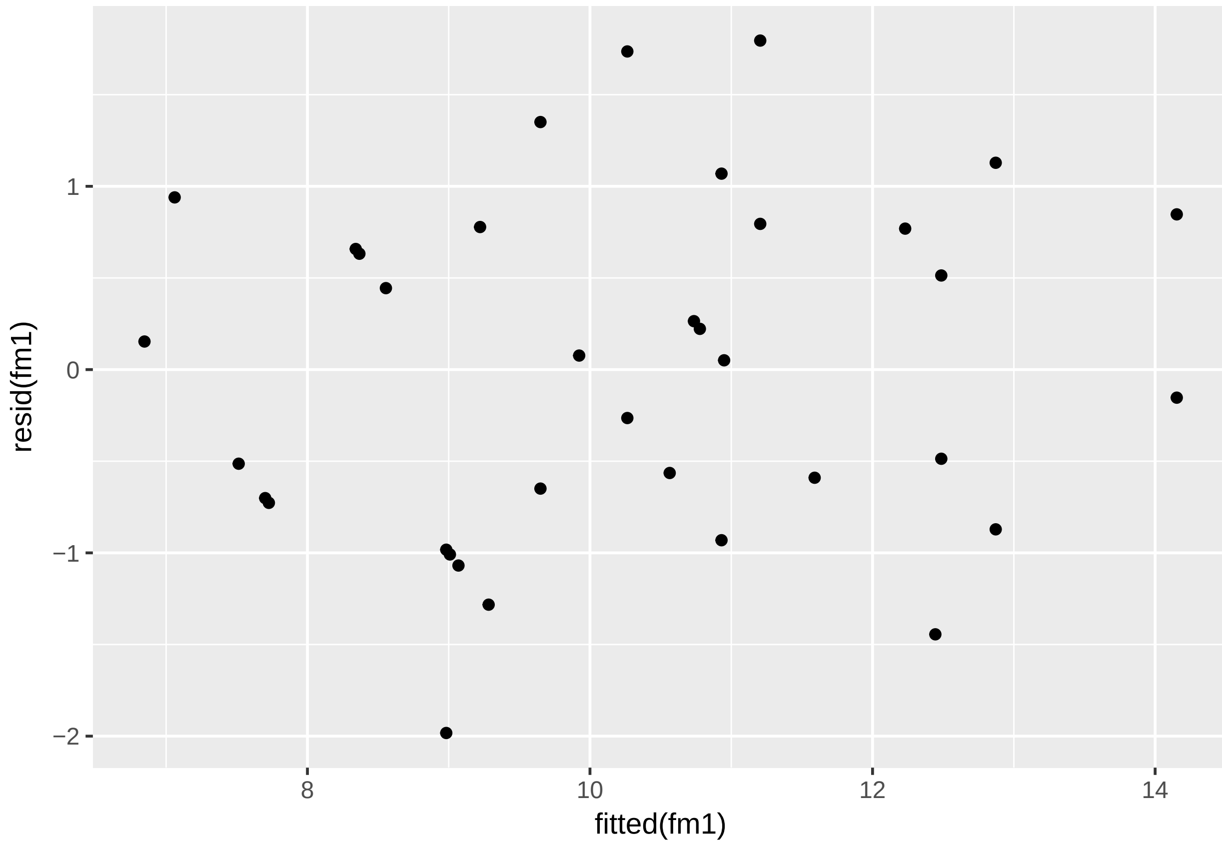
```r
resid(fm1)
```

```
##           1           2           3           4           5           6
##  1.73572821  0.84683932 -0.48649401 -0.93093846 -0.26427179 -0.15316068
##           7           8           9          10          11          12
##  0.51350599  1.06906154 -1.98273461  1.12837650  1.79504316 -0.64940128
##          13          14          15          16          17          18
## -0.70119744 -0.59008633  0.07658034  0.63213590  0.93957115  0.05068226
##          19          20          21          22          23          24
## -1.28265107 -0.72709552  0.44444444 -1.44444444  0.22222222  0.77777778
```

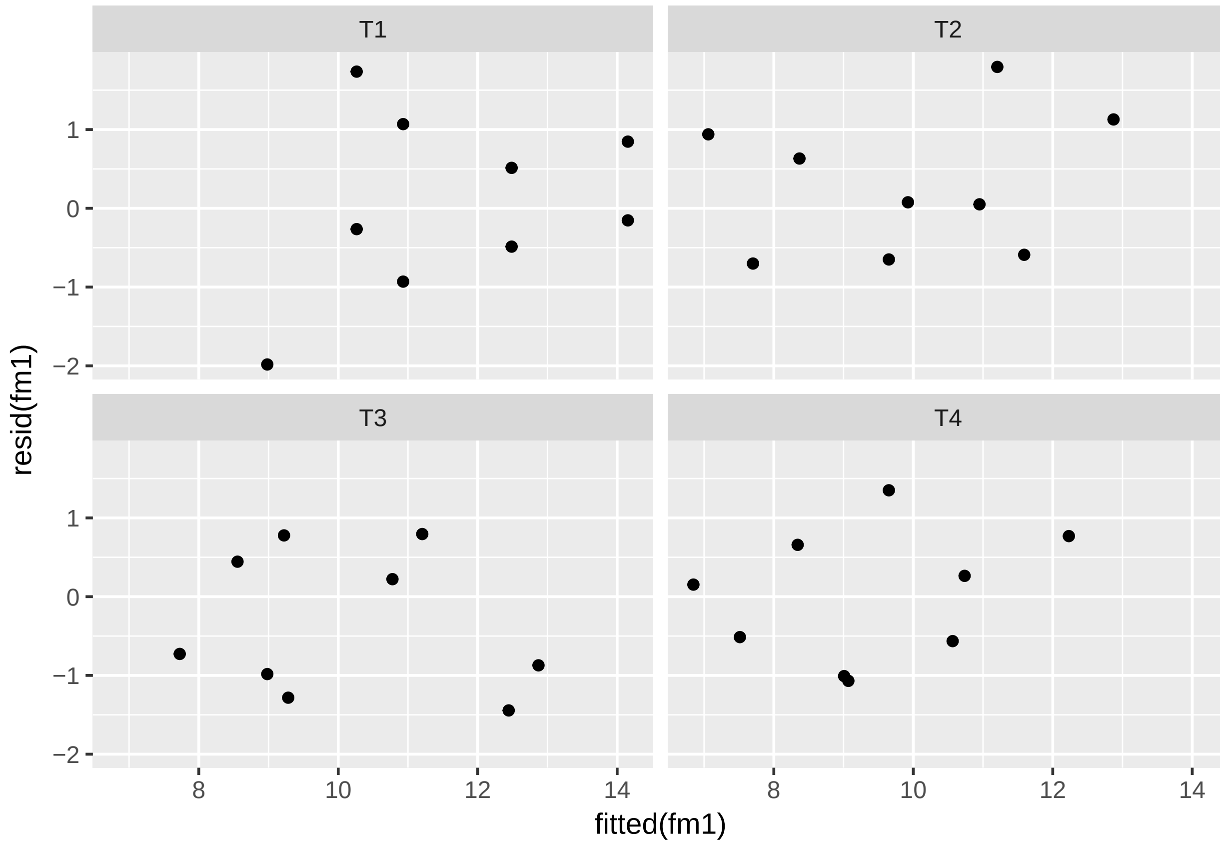
```
##          25          26          27          28          29          30
## -0.98273461 -0.87162350  0.79504316  1.35059872  0.15316068  0.26427179
##          31          32          33          34          35          36
## -1.06906154 -0.51350599  0.65803397  0.76914508 -0.56418825 -1.00863269
```

Plot level 1 fitted values against residuals

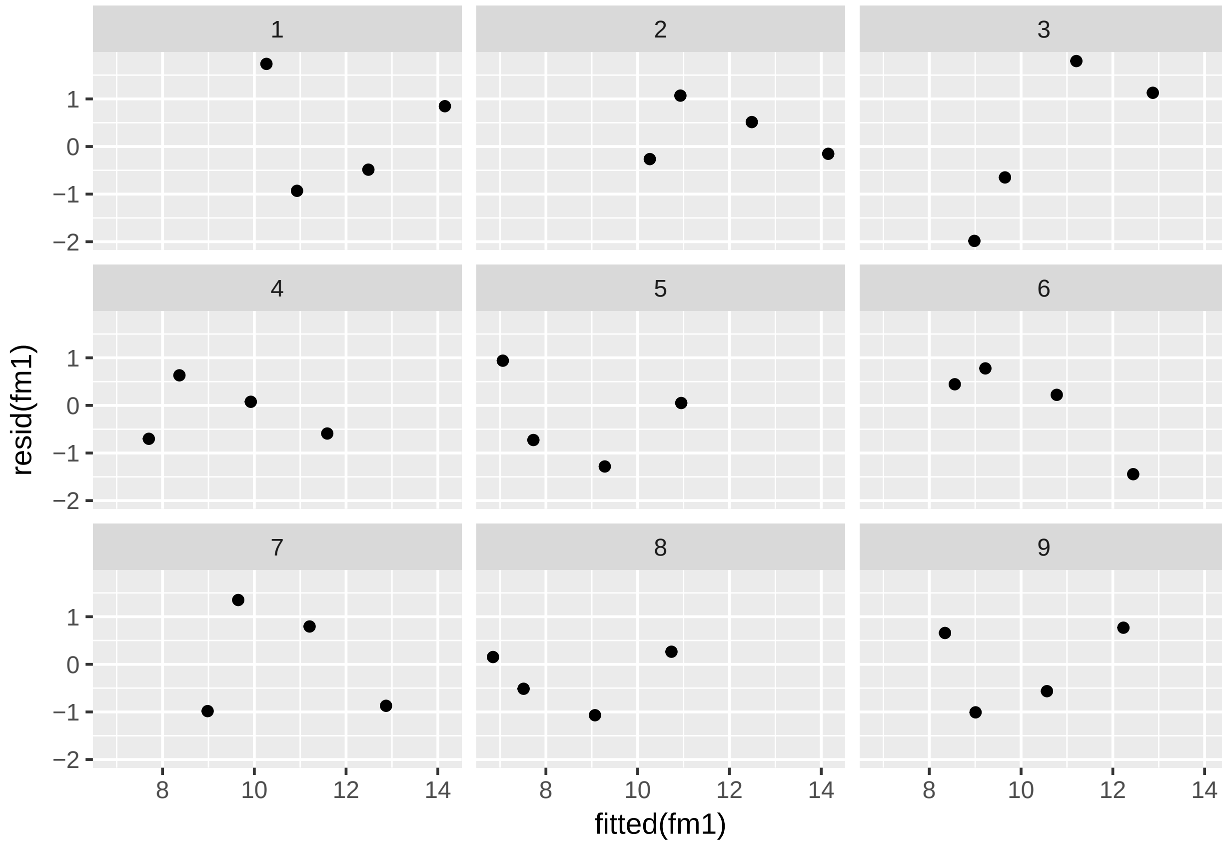
```
qplot(fitted(fm1),resid(fm1), data = ergoStool)
```




```
qplot(fitted(fm1), resid(fm1), facets=~Type, data = ergoStool) # Residual plots by stool type
```

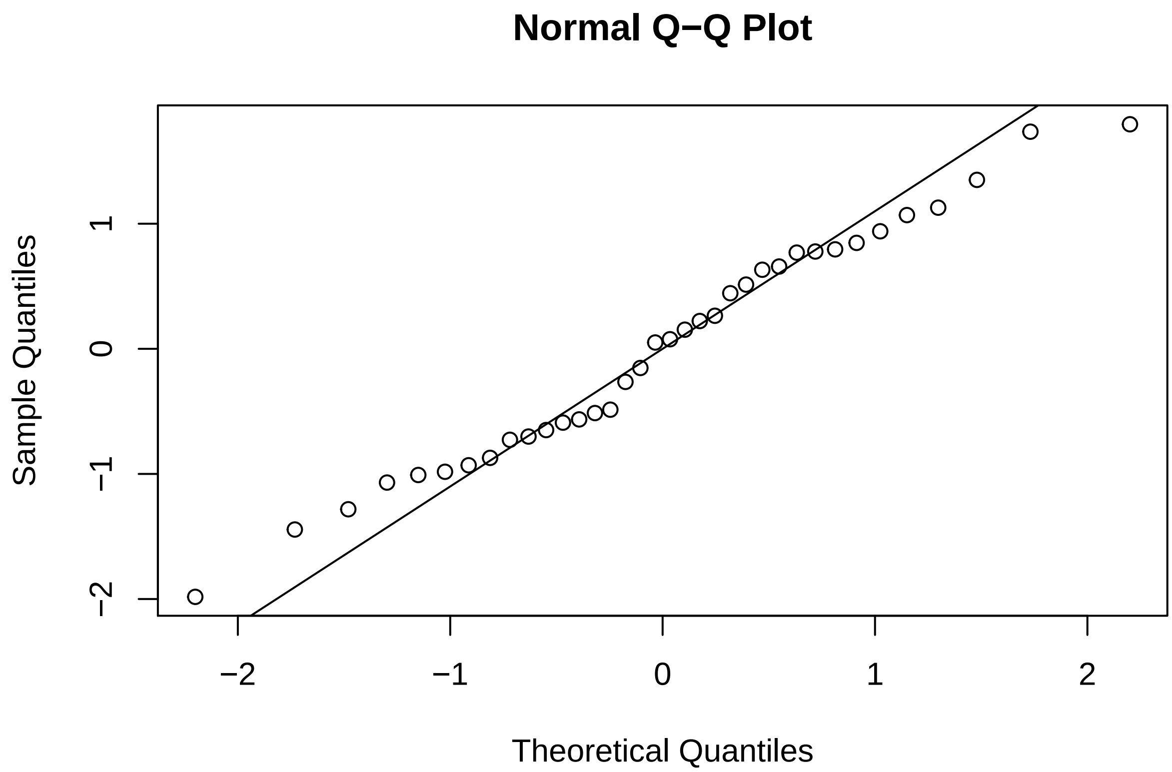



```
qplot(fitted(fm1), resid(fm1), facets=~Subject, data = ergoStool) # Residual plots by subject
```



Check assumption that errors are normally distributed

```
qqnorm(resid(fm1))
#qqline(resid(fm1))
abline(0,1.1003)
```



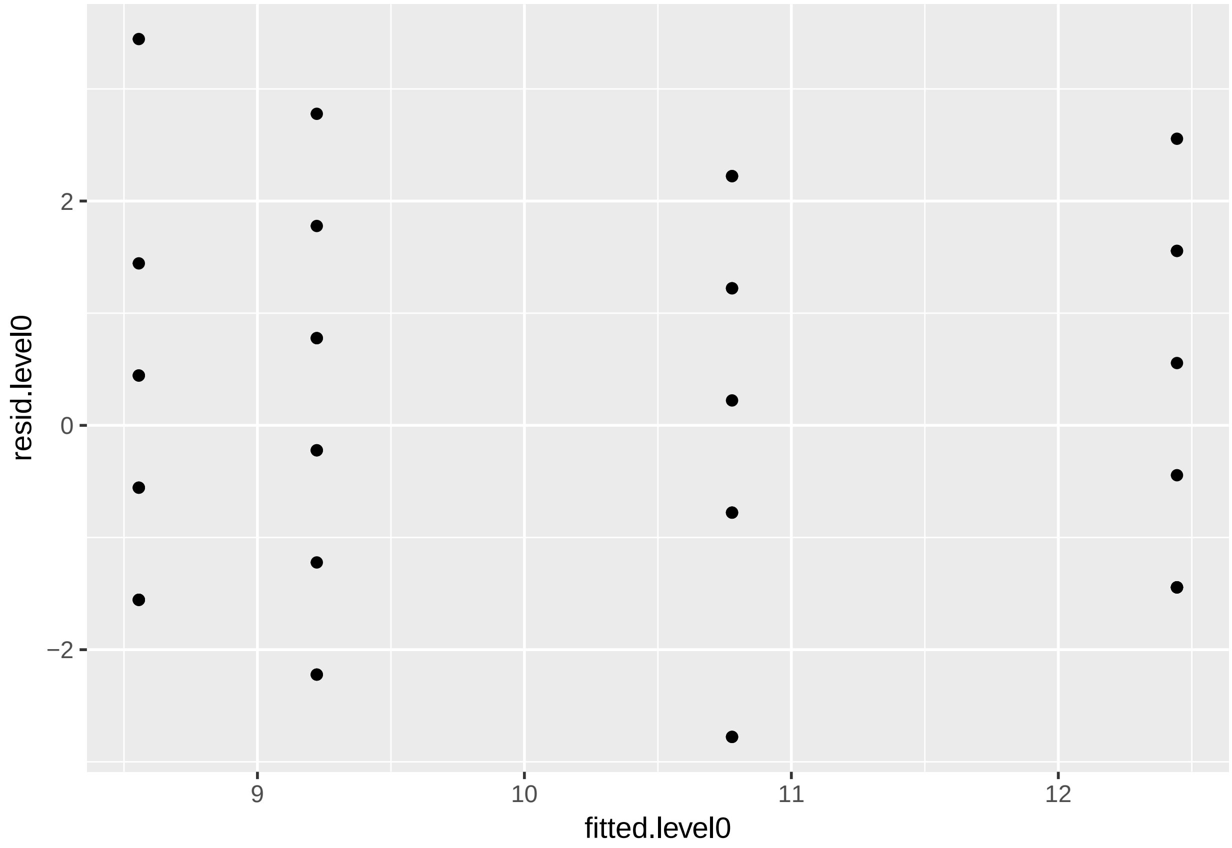
Plot level 0 fitted values against residuals

```
(fitted.level0<-fm1@pp$X %*% fixef(fm1))
```

```
##          [,1]
## 1   8.555556
## 2  12.444444
## 3  10.777778
## 4   9.222222
## 5   8.555556
## 6  12.444444
## 7  10.777778
## 8   9.222222
## 9   8.555556
## 10 12.444444
## 11 10.777778
## 12  9.222222
## 13  8.555556
## 14 12.444444
## 15 10.777778
## 16  9.222222
## 17  8.555556
## 18 12.444444
## 19 10.777778
## 20  9.222222
## 21  8.555556
## 22 12.444444
```

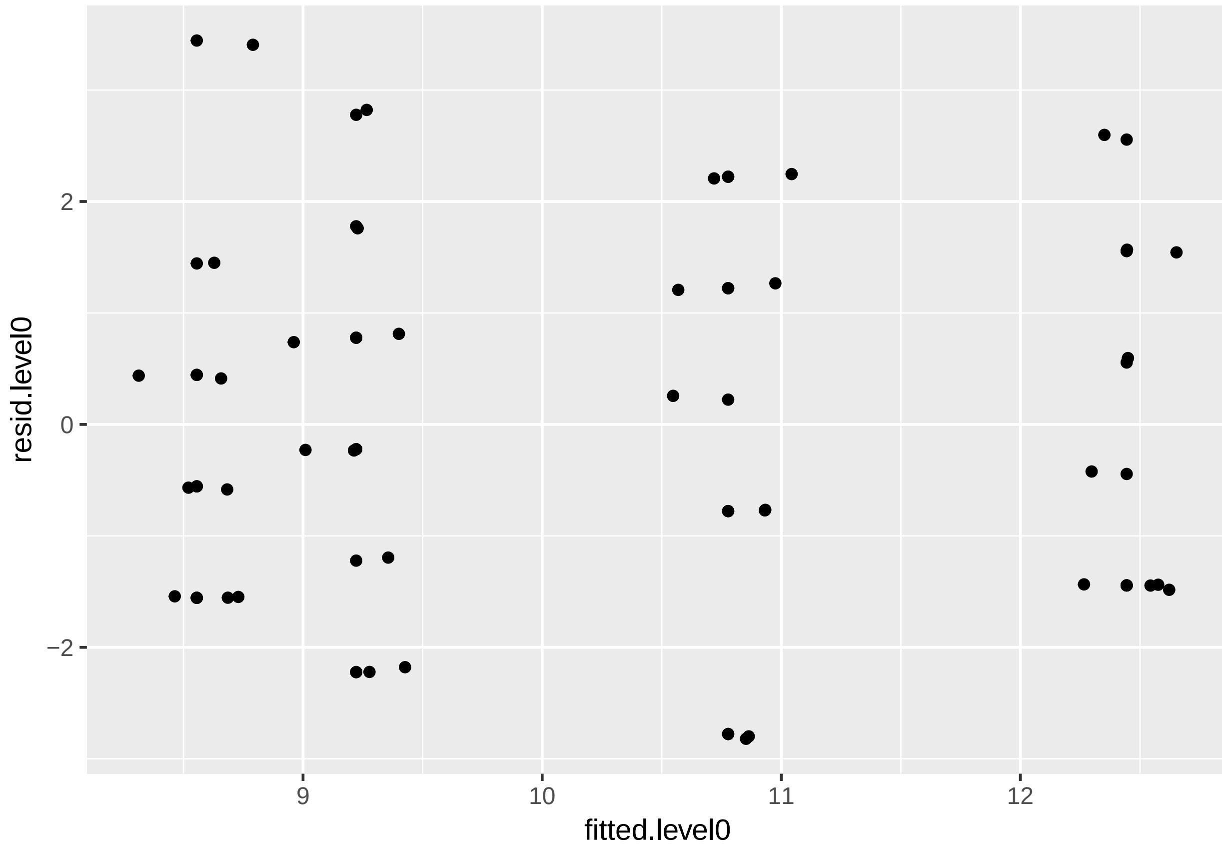
```
## 23 10.777778
## 24  9.222222
## 25  8.555556
## 26 12.444444
## 27 10.777778
## 28  9.222222
## 29  8.555556
## 30 12.444444
## 31 10.777778
## 32  9.222222
## 33  8.555556
## 34 12.444444
## 35 10.777778
## 36  9.222222
```

```r
resid.level0<-effort-fitted.level0
qplot(fitted.level0,resid.level0)
```



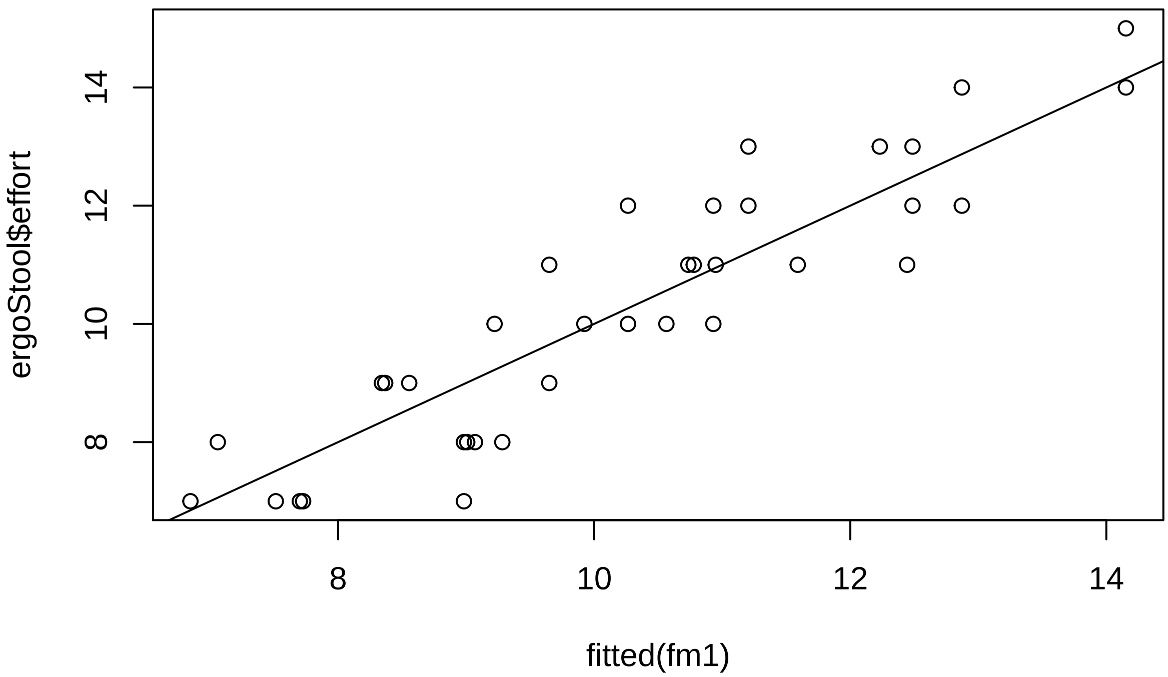
Can jitter the points to make easier to see

```r
qplot(fitted.level0,resid.level0)+geom_jitter()
```



Assess general fit of model

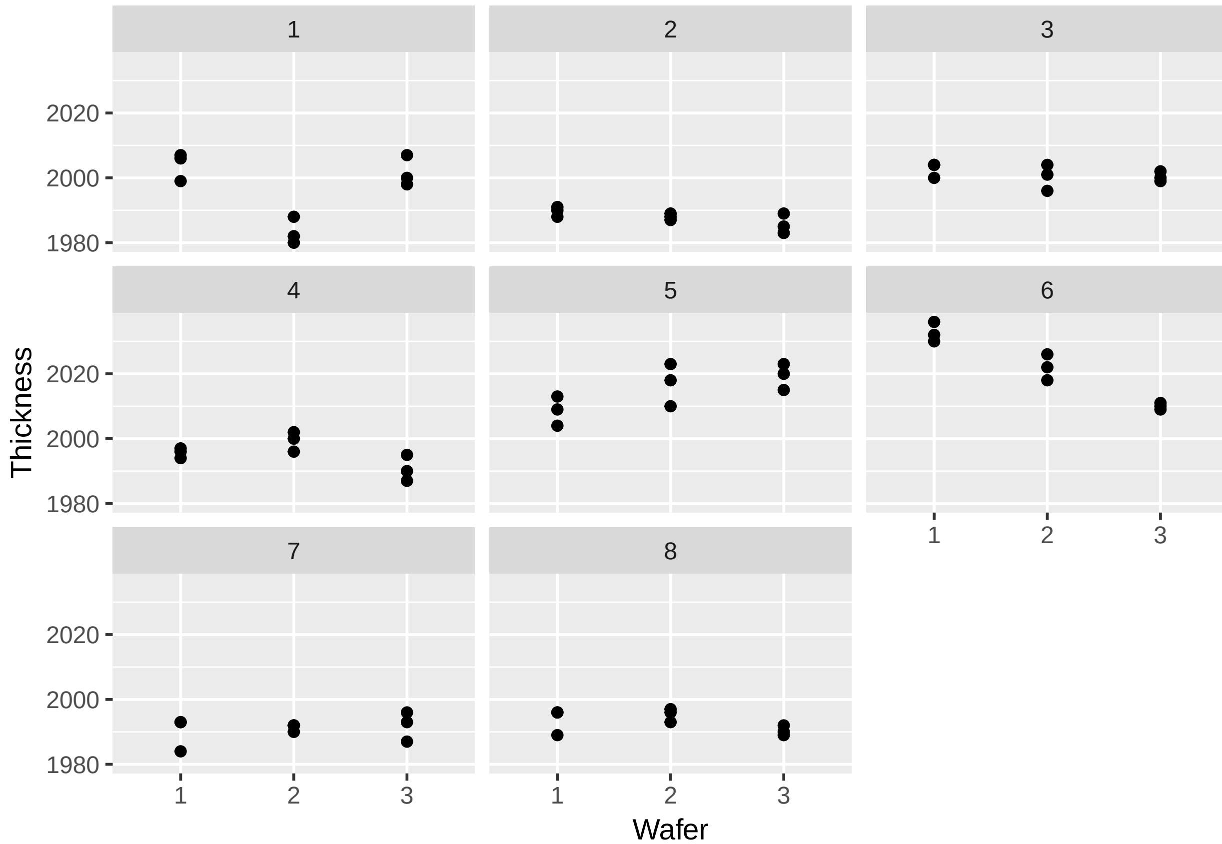
```
plot(fitted(fm1),ergoStool$effort)
abline(0,1)
```



Section 4.1.1 Oxides example

Plot the data first to check for outlying lots, wafers within lots, sites within wafers

```
attach(Oxide)
qplot(Wafer, Thickness, facets=~Lot, data=Oxide)
```




```r
fm1<-lmer(Thickness~1+(1|Lot/Wafer),data=Oxide)
summary(fm1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Thickness ~ 1 + (1 | Lot/Wafer)
##    Data: Oxide
##
## REML criterion at convergence: 454
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8746 -0.4991  0.1047  0.5510  1.7922
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Wafer:Lot (Intercept)  35.87    5.989
##  Lot       (Intercept) 129.91   11.398
##  Residual               12.57    3.545
## Number of obs: 72, groups:  Wafer:Lot, 24; Lot, 8
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 2000.153      4.232   472.7
```

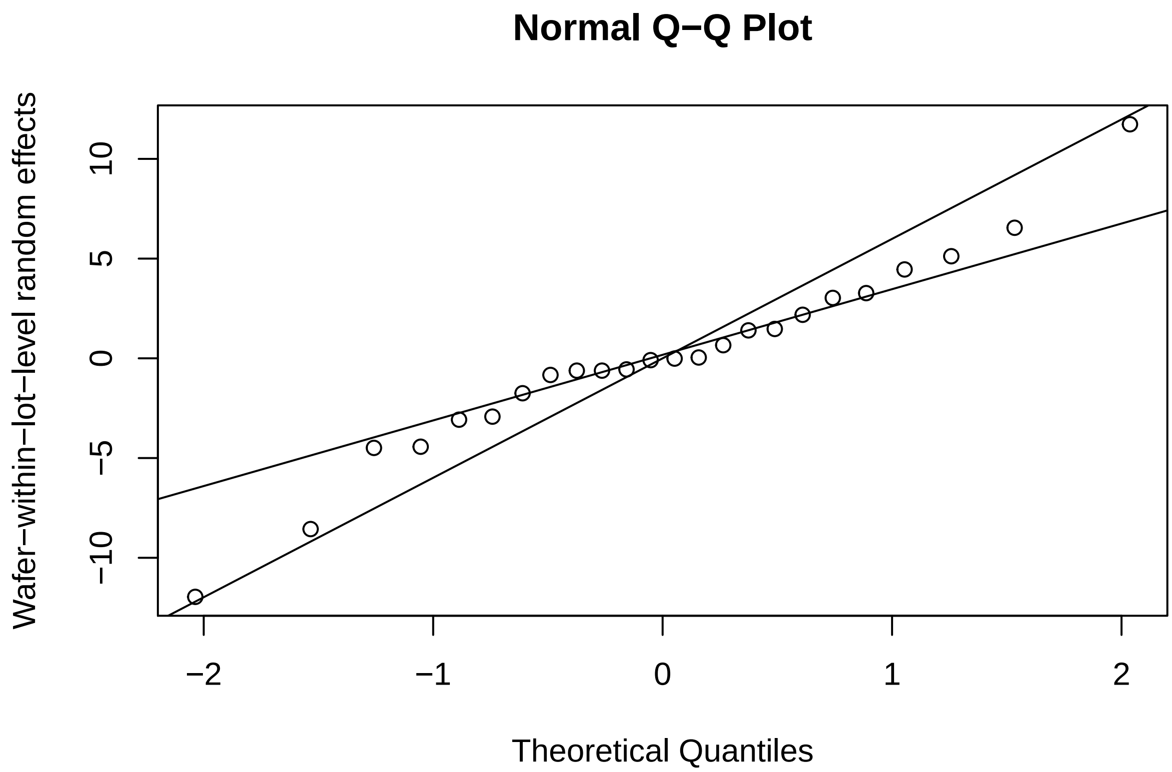
Estimated random effects

```r
ranef(fm1)
```

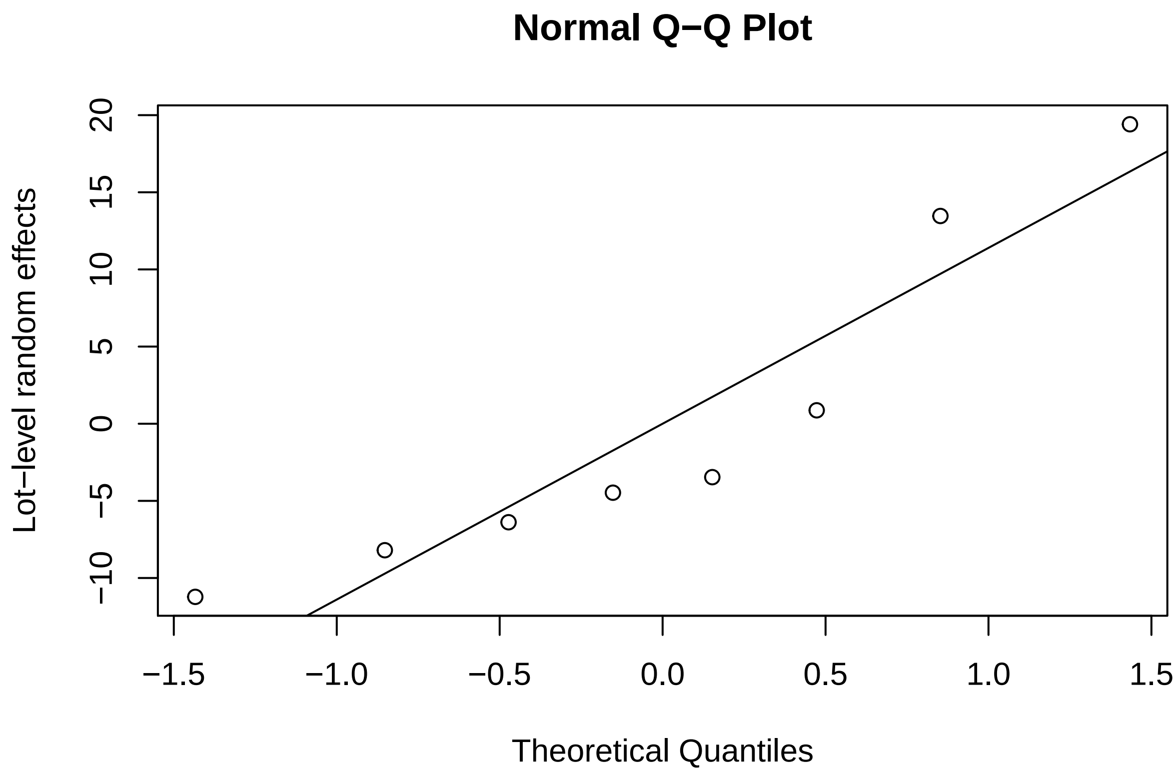
```
## $`Wafer:Lot`
##      (Intercept)
## 1:1   6.54599243
## 1:2   0.65859294
## 1:3   1.47281908
## 1:4  -0.01350901
## 1:5  -4.43183625
## 1:6  11.73499147
## 1:7  -1.74943356
## 1:8  -0.09019648
## 2:1 -11.95893879
## 2:2  -0.83374023
## 2:3  -0.61644735
## 2:4   3.26962395
## 2:5   3.02982956
## 2:6   2.18405923
## 2:7  -0.55556703
## 2:8   1.40213668
## 3:1   4.45672600
## 3:2  -2.92300666
## 3:3  -0.61644735
## 3:4  -4.49050850
## 3:5   5.11909599
## 3:6  -8.56073955
## 3:7   0.04136623
## 3:8  -3.07486281
##
## $Lot
##   (Intercept)
## 1  -3.4634693
## 2 -11.2216405
## 3   0.8690159
## 4  -4.4710240
## 5  13.4634497
## 6  19.4080225
## 7  -8.1989764
## 8  -6.3853779
```

Check model assumptions

```r
qqnorm(unlist(ranef(fm1)$`Wafer:Lot`),ylab="Wafer-within-lot-level random effects")
abline(0,5.9891)
qqline(unlist(ranef(fm1)$`Wafer:Lot`))
```

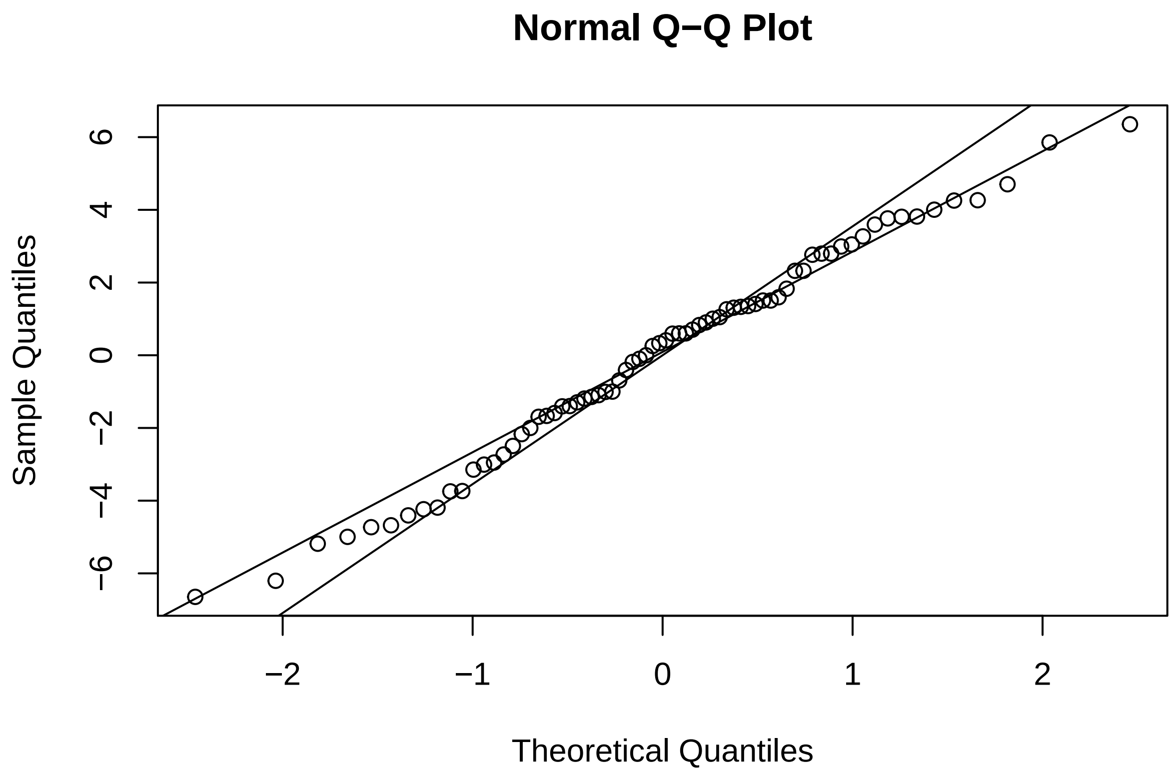



```
qqnorm(unlist(ranef(fm1)$Lot),ylab="Lot-level random effects")
abline(0,11.3967)
```

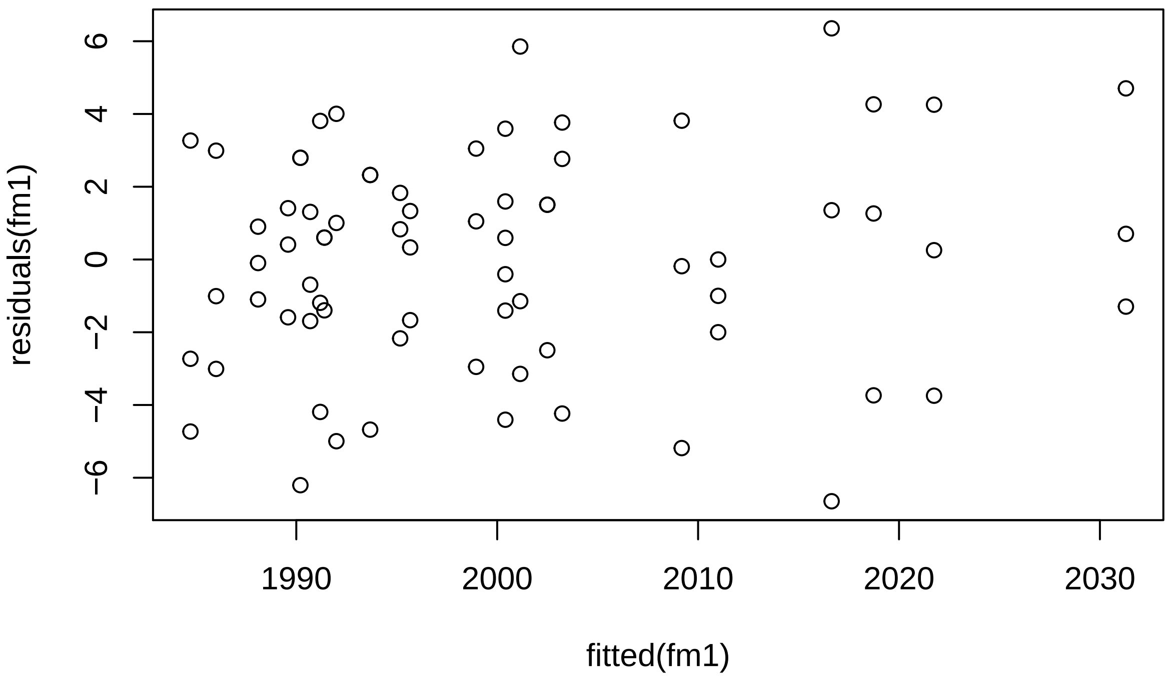


Doesn't look so good, but sample size is small

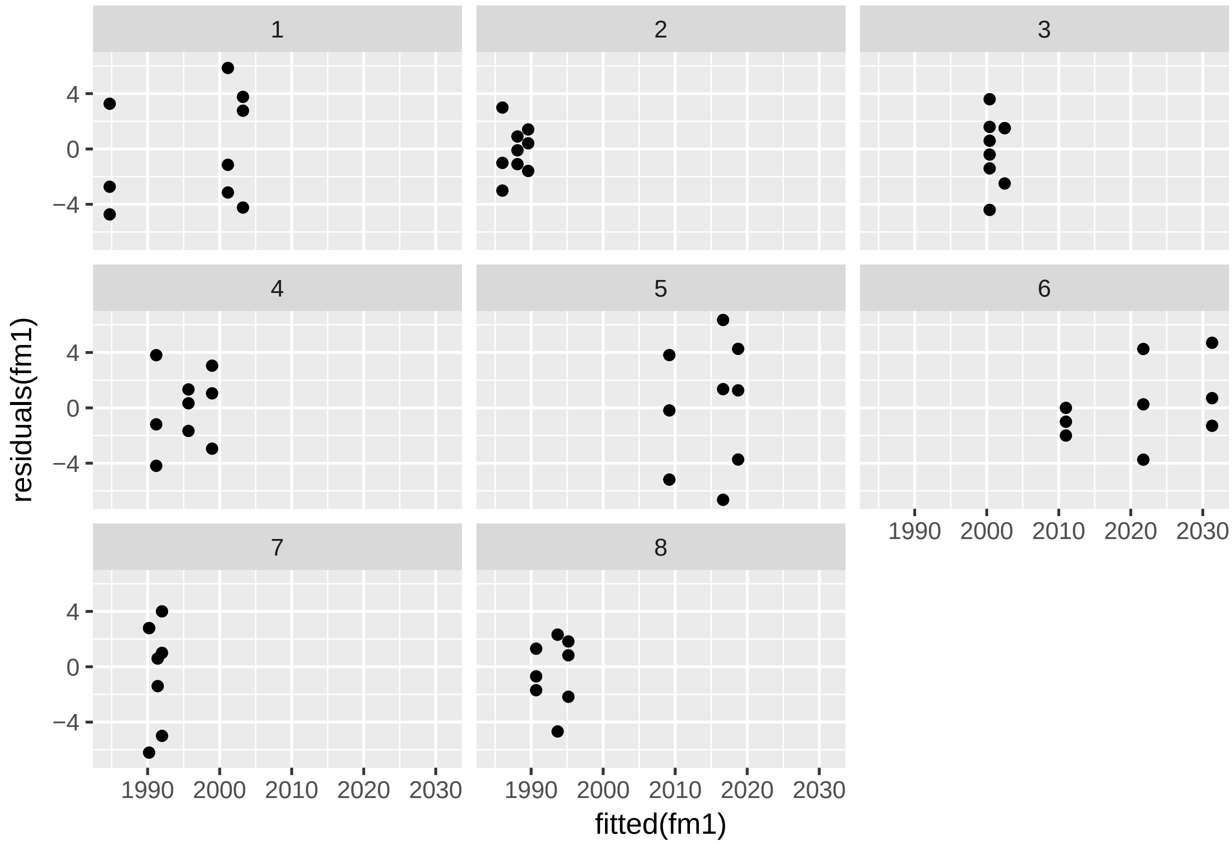
```
qqnorm(resid(fm1))
abline(0, 3.5453)
qqline(resid(fm1))
```




```
plot(fitted(fm1),residuals(fm1))
```

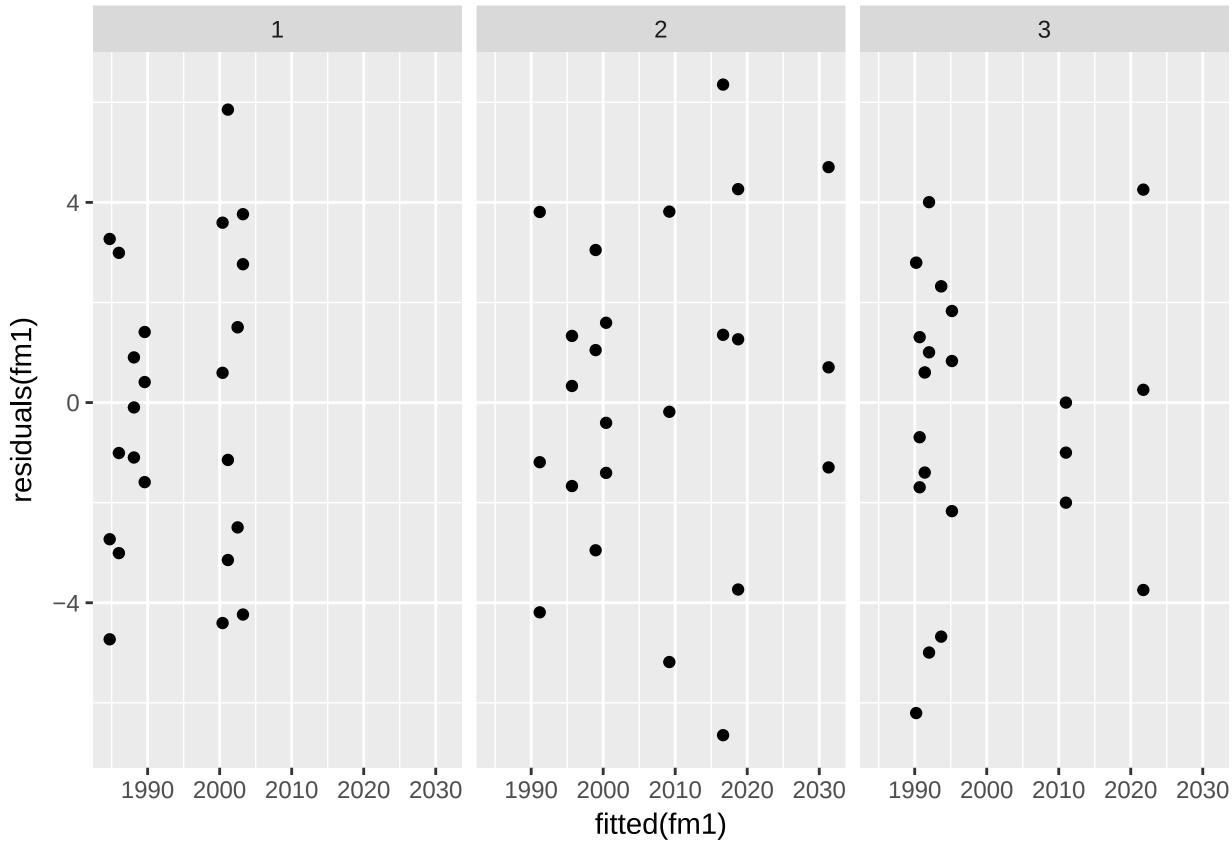



```
qplot(fitted(fm1), residuals(fm1), facets=~Lot, data=Oxide)
```



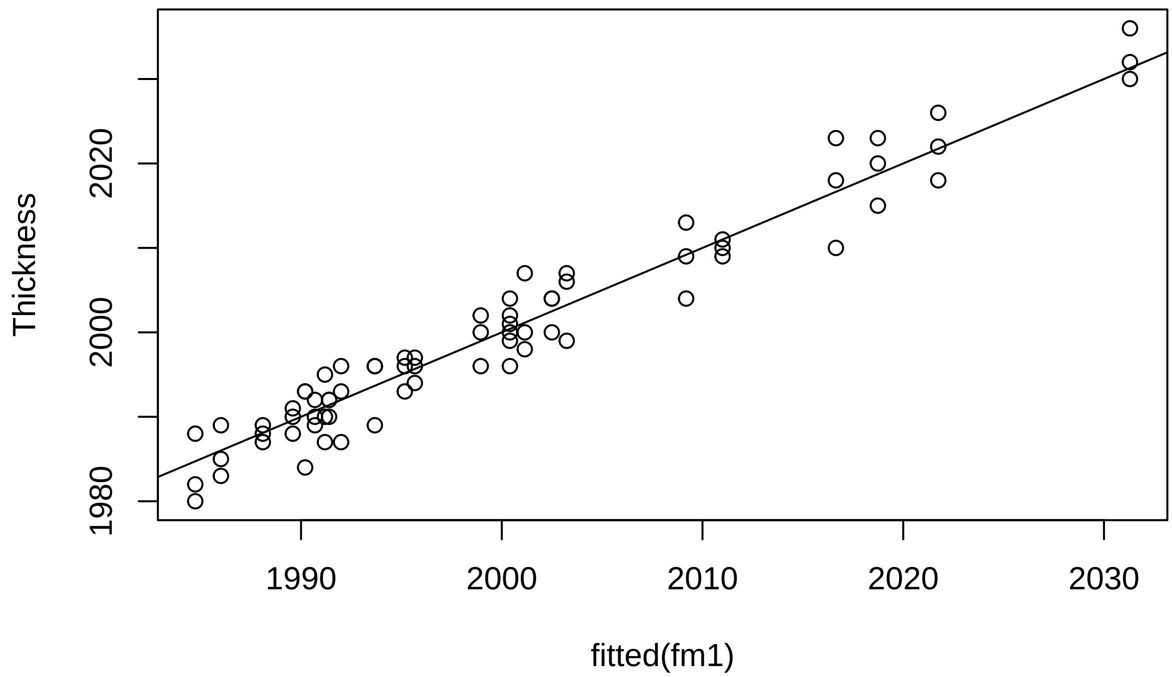
Why isn't the following plot helpful?

```
qplot(fitted(fm1),residuals(fm1), facets=~Wafer, data=Oxide)
```



Assess overall model fit

```
plot(fitted(fm1),Thickness)
abline(0,1)
```



Three levels of fitted values In matrix notation (Section 2.6), can extract Z with

```
(Zt<-fm1@pp$Zt) # this is transpose of Z
```

```
## 32 x 72 sparse Matrix of class "dgCMatrix"
##    [[ suppressing 72 column names '1', '2', '3' ... ]]
##
## 1:1 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:2 . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . .
## 1:3 . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . .
## 1:4 . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . .
## 1:5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:6 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:8 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:1 . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:2 . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . .
## 2:3 . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . .
## 2:4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . .
## 2:5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:6 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:8 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:1 . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:2 . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . .
## 3:3 . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . .
```

```
## 3:4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1
## 3:5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:6 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:8 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1   1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2   . . . . . . . . . 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . . .
## 3   . . . . . . . . . . . . . . . . . . 1 1 1 1 1 1 1 1 1 . . . . . . . .
## 4   . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 1 1 1 1 1
## 5   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 6   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 7   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 8   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
##
## 1:1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:5 . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 1:6 . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . .
## 1:7 . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . .
## 1:8 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . .
## 2:1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:5 . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2:6 . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . .
## 2:7 . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . .
## 2:8 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 .
## 3:1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:4 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:5 . . . . . . . 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . .
## 3:6 . . . . . . . . . . . . . . . . 1 1 1 . . . . . . . . . . . . . . . .
## 3:7 . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 . . . . . . .
## 3:8 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
## 1   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 2   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 3   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 4   1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
## 5   . 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . . . . . . . . . . .
## 6   . . . . . . . . . . 1 1 1 1 1 1 1 1 1 . . . . . . . . . . . . . . . .
## 7   . . . . . . . . . . . . . . . . . . . 1 1 1 1 1 1 1 1 1 . . . . . . .
## 8   . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 1 1 1 1
##
## 1:1 . .
## 1:2 . .
## 1:3 . .
## 1:4 . .
## 1:5 . .
## 1:6 . .
## 1:7 . .
```

```
## 1:8 . .
## 2:1 . .
## 2:2 . .
## 2:3 . .
## 2:4 . .
## 2:5 . .
## 2:6 . .
## 2:7 . .
## 2:8 . .
## 3:1 . .
## 3:2 . .
## 3:3 . .
## 3:4 . .
## 3:5 . .
## 3:6 . .
## 3:7 . .
## 3:8 1 1
## 1   . .
## 2   . .
## 3   . .
## 4   . .
## 5   . .
## 6   . .
## 7   . .
## 8   1 1
```

Convert into normal matrix form, and take transpose

```
Z<-as.matrix(t(Zt))
```

Put the predicted random effects into single column vector

```
b.hat<-as.matrix(rbind(ranef(fm1)$`Wafer:Lot`,ranef(fm1)$Lot ))
```

Inner most level is 'level 2'

$$\hat{y_{ijk}} = \hat{beta} + \hat{b_i} + \hat{b_{ij}}$$

Calculate manually

```
fitted.level2<-as.matrix(fm1@pp$X %*% fixef(fm1) + Z%*%b.hat)
```

inner most level also given by the fitted command

```
fitted(fm1)
```

```
##        1        2        3        4        5        6        7        8
## 2003.235 2003.235 2003.235 1984.730 1984.730 1984.730 2001.146 2001.146
##        9       10       11       12       13       14       15       16
## 2001.146 1989.590 1989.590 1989.590 1988.097 1988.097 1988.097 1986.008
##       17       18       19       20       21       22       23       24
## 1986.008 1986.008 2002.495 2002.495 2002.495 2000.405 2000.405 2000.405
##       25       26       27       28       29       30       31       32
## 2000.405 2000.405 2000.405 1995.668 1995.668 1995.668 1998.951 1998.951
##       33       34       35       36       37       38       39       40
## 1998.951 1991.191 1991.191 1991.191 2009.184 2009.184 2009.184 2016.646
##       41       42       43       44       45       46       47       48
```

```
## 2016.646 2016.646 2018.735 2018.735 2018.735 2031.296 2031.296 2031.296
##       49       50       51       52       53       54       55       56
## 2021.745 2021.745 2021.745 2011.000 2011.000 2011.000 1990.204 1990.204
##       57       58       59       60       61       62       63       64
## 1990.204 1991.398 1991.398 1991.398 1991.995 1991.995 1991.995 1993.677
##       65       66       67       68       69       70       71       72
## 1993.677 1993.677 1995.170 1995.170 1995.170 1990.693 1990.693 1990.693
```

Check they are the same

```
cbind(fitted.level2, fitted(fm1))
```

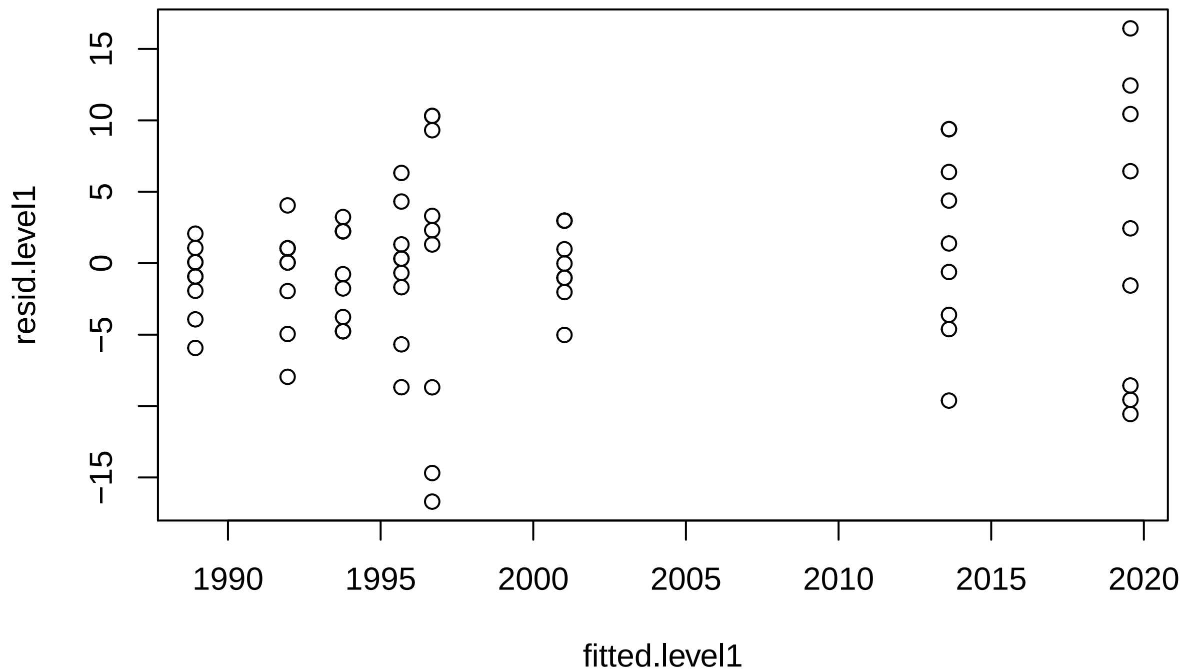
```
##        [,1]     [,2]
## 1  2003.235 2003.235
## 2  2003.235 2003.235
## 3  2003.235 2003.235
## 4  1984.730 1984.730
## 5  1984.730 1984.730
## 6  1984.730 1984.730
## 7  2001.146 2001.146
## 8  2001.146 2001.146
## 9  2001.146 2001.146
## 10 1989.590 1989.590
## 11 1989.590 1989.590
## 12 1989.590 1989.590
## 13 1988.097 1988.097
## 14 1988.097 1988.097
## 15 1988.097 1988.097
## 16 1986.008 1986.008
## 17 1986.008 1986.008
## 18 1986.008 1986.008
## 19 2002.495 2002.495
## 20 2002.495 2002.495
## 21 2002.495 2002.495
## 22 2000.405 2000.405
## 23 2000.405 2000.405
## 24 2000.405 2000.405
## 25 2000.405 2000.405
## 26 2000.405 2000.405
## 27 2000.405 2000.405
## 28 1995.668 1995.668
## 29 1995.668 1995.668
## 30 1995.668 1995.668
## 31 1998.951 1998.951
## 32 1998.951 1998.951
## 33 1998.951 1998.951
## 34 1991.191 1991.191
## 35 1991.191 1991.191
## 36 1991.191 1991.191
## 37 2009.184 2009.184
## 38 2009.184 2009.184
## 39 2009.184 2009.184
## 40 2016.646 2016.646
## 41 2016.646 2016.646
## 42 2016.646 2016.646
```

```
## 43 2018.735 2018.735
## 44 2018.735 2018.735
## 45 2018.735 2018.735
## 46 2031.296 2031.296
## 47 2031.296 2031.296
## 48 2031.296 2031.296
## 49 2021.745 2021.745
## 50 2021.745 2021.745
## 51 2021.745 2021.745
## 52 2011.000 2011.000
## 53 2011.000 2011.000
## 54 2011.000 2011.000
## 55 1990.204 1990.204
## 56 1990.204 1990.204
## 57 1990.204 1990.204
## 58 1991.398 1991.398
## 59 1991.398 1991.398
## 60 1991.398 1991.398
## 61 1991.995 1991.995
## 62 1991.995 1991.995
## 63 1991.995 1991.995
## 64 1993.677 1993.677
## 65 1993.677 1993.677
## 66 1993.677 1993.677
## 67 1995.170 1995.170
## 68 1995.170 1995.170
## 69 1995.170 1995.170
## 70 1990.693 1990.693
## 71 1990.693 1990.693
## 72 1990.693 1990.693
```

Level 1

$$\hat{y_{ijk}} = \hat{beta} + \hat{b_i}$$

```r
fitted.level1<-as.matrix(fm1@pp$X %*% fixef(fm1) + Z[,25:32]%*%b.hat[25:32,1])
resid.level1<-Thickness-fitted.level1
plot(fitted.level1, resid.level1)
```



Level 0 (not very informative, as only a single fixed effect parameter)

$$\hat{y_{ijk}} = b\hat{e}ta$$

```
(fitted.level0<-as.matrix(fm1@pp$X %*% fixef(fm1) ))
```

```
##        [,1]
## 1  2000.153
## 2  2000.153
## 3  2000.153
## 4  2000.153
## 5  2000.153
## 6  2000.153
## 7  2000.153
## 8  2000.153
## 9  2000.153
## 10 2000.153
## 11 2000.153
## 12 2000.153
## 13 2000.153
## 14 2000.153
## 15 2000.153
## 16 2000.153
## 17 2000.153
## 18 2000.153
## 19 2000.153
```

```
## 20 2000.153
## 21 2000.153
## 22 2000.153
## 23 2000.153
## 24 2000.153
## 25 2000.153
## 26 2000.153
## 27 2000.153
## 28 2000.153
## 29 2000.153
## 30 2000.153
## 31 2000.153
## 32 2000.153
## 33 2000.153
## 34 2000.153
## 35 2000.153
## 36 2000.153
## 37 2000.153
## 38 2000.153
## 39 2000.153
## 40 2000.153
## 41 2000.153
## 42 2000.153
## 43 2000.153
## 44 2000.153
## 45 2000.153
## 46 2000.153
## 47 2000.153
## 48 2000.153
## 49 2000.153
## 50 2000.153
## 51 2000.153
## 52 2000.153
## 53 2000.153
## 54 2000.153
## 55 2000.153
## 56 2000.153
## 57 2000.153
## 58 2000.153
## 59 2000.153
## 60 2000.153
## 61 2000.153
## 62 2000.153
## 63 2000.153
## 64 2000.153
## 65 2000.153
## 66 2000.153
## 67 2000.153
## 68 2000.153
## 69 2000.153
## 70 2000.153
## 71 2000.153
## 72 2000.153
```