A Handbook of Statistical Analyses Using R — 2nd Edition

Brian S. Everitt and Torsten Hothorn

CHAPTER 12

Analysing Longitudinal Data I: Computerised Delivery of Cognitive Behavioural Therapy – Beat the Blues

12.1 Introduction

12.2 Analysing Longitudinal Data

12.3 Analysis Using R

We shall fit both random intercept and random intercept and slope models to the data including the baseline BDI values (`pre.bdi`), `treatment` group, `drug` and `length` as fixed effect covariates. Linear mixed effects models are fitted in R by using the `lmer` function contained in the **lme4** package (Bates and Sarkar, 2012, Pinheiro and Bates, 2000, Bates, 2005), but an essential first step is to rearrange the data from the 'wide form' in which they appear in the `BtheB` data frame into the 'long form' in which each separate repeated measurement and associated covariate values appear as a separate row in a *data.frame*. This rearrangement can be made using the following code:

```
R> data("BtheB", package = "HSAUR2")
R> BtheB$subject <- factor(rownames(BtheB))
R> nobs <- nrow(BtheB)
R> BtheB_long <- reshape(BtheB, idvar = "subject",
+      varying = c("bdi.2m", "bdi.3m", "bdi.5m", "bdi.8m"),
+      direction = "long")
R> BtheB_long$time <- rep(c(2, 3, 5, 8), rep(nobs, 4))
```

such that the data are now in the form (here shown for the first three subjects)

```
R> subset(BtheB_long, subject %in% c("1", "2", "3"))
```

	drug	length	treatment	bdi.pre	subject	time	bdi
1.2m	No	>6m	TAU	29	1	2	2
2.2m	Yes	>6m	BtheB	32	2	2	16
3.2m	Yes	<6m	TAU	25	3	2	20
1.3m	No	>6m	TAU	29	1	3	2
2.3m	Yes	>6m	BtheB	32	2	3	24
3.3m	Yes	<6m	TAU	25	3	3	NA
1.5m	No	>6m	TAU	29	1	5	NA
2.5m	Yes	>6m	BtheB	32	2	5	17
3.5m	Yes	<6m	TAU	25	3	5	NA
1.8m	No	>6m	TAU	29	1	8	NA
2.8m	Yes	>6m	BtheB	32	2	8	20
3.8m	Yes	<6m	TAU	25	3	8	NA

The resulting *data.frame* `BtheB_long` contains a number of missing values

```
R> data("BtheB", package = "HSAUR2")
R> layout(matrix(1:2, nrow = 1))
R> ylim <- range(BtheB[,grep("bdi", names(BtheB))],
+                na.rm = TRUE)
R> tau <- subset(BtheB, treatment == "TAU")[,
+      grep("bdi", names(BtheB))]
R> boxplot(tau, main = "Treated as Usual", ylab = "BDI",
+          xlab = "Time (in months)", names = c(0, 2, 3, 5, 8),
+          ylim = ylim)
R> btheb <- subset(BtheB, treatment == "BtheB")[,
+      grep("bdi", names(BtheB))]
R> boxplot(btheb, main = "Beat the Blues", ylab = "BDI",
+          xlab = "Time (in months)", names = c(0, 2, 3, 5, 8),
+          ylim = ylim)
```

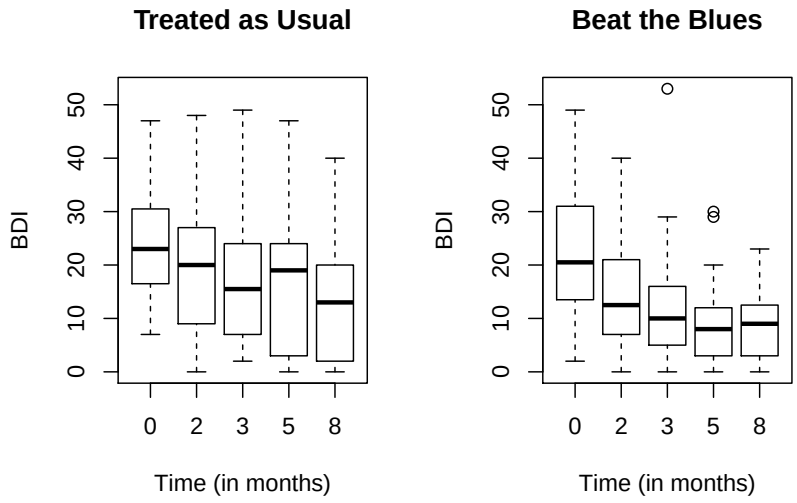

Figure 12.1 Boxplots for the repeated measures by treatment group for the `BtheB` data.

and in applying the `lmer` function these will be dropped. But notice it is only the missing values that are removed, *not* participants that have at least one missing value. All the available data is used in the model fitting process. The `lmer` function is used in a similar way to the `lm` function met in Chapter 6 with the addition of a random term to identify the source of the repeated measurements, here `subject`. We can fit the two models (**??**) and (**??**) and test which is most appropriate using

```
R> library("lme4")
```

```
R> BtheB_lmer1 <- lmer(bdi ~ bdi.pre + time + treatment + drug +
+      length + (1 | subject), data = BtheB_long,
+      REML = FALSE, na.action = na.omit)
R> BtheB_lmer2 <- lmer(bdi ~ bdi.pre + time + treatment + drug +
+      length + (time | subject), data = BtheB_long,
+      REML = FALSE, na.action = na.omit)
R> anova(BtheB_lmer1, BtheB_lmer2)
```

```
Data: BtheB_long
Models:
BtheB_lmer1: bdi ~ bdi.pre + time + treatment + drug + length + (1 | subject)
BtheB_lmer2: bdi ~ bdi.pre + time + treatment + drug + length + (time | subject)
            Df    AIC    BIC  logLik deviance  Chisq Chi Df
BtheB_lmer1  8 1887.5 1916.6 -935.75   1871.5
BtheB_lmer2 10 1891.0 1927.4 -935.52   1871.0 0.4542      2
            Pr(>Chisq)
BtheB_lmer1
BtheB_lmer2     0.7969
```

```
R> summary(BtheB_lmer1)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula:
bdi ~ bdi.pre + time + treatment + drug + length + (1 | subject)
   Data: BtheB_long

     AIC      BIC   logLik deviance df.resid
  1887.5   1916.6   -935.7   1871.5      272

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.6975 -0.5026 -0.0638  0.4124  3.8203

Random effects:
 Groups   Name        Variance Std.Dev.
 subject  (Intercept) 48.78    6.984
 Residual             25.14    5.014
Number of obs: 280, groups:  subject, 97

Fixed effects:
              Estimate Std. Error t value
(Intercept)     5.59239    2.24244   2.494
bdi.pre         0.63968    0.07789   8.212
time           -0.70476    0.14639  -4.814
treatmentBtheB -2.32908    1.67036  -1.394
drugYes        -2.82495    1.72684  -1.636
length>6m       0.19708    1.63832   0.120

Correlation of Fixed Effects:
            (Intr) bdi.pr time   trtmBB drugYs
bdi.pre     -0.682
time        -0.238  0.020
tretmntBthB -0.390  0.121  0.018
drugYes     -0.073 -0.237 -0.022 -0.323
length>6m   -0.243 -0.242 -0.036  0.002  0.157
```

Figure 12.2 R output of the linear mixed-effects model fit for the `BtheB` data.

The `summary` method for *lmer* objects doesn't print p-values for Gaussian mixed models because the degrees of freedom of the t reference distribution are

not obvious. However, one can rely on the asymptotic normal distribution for computing univariate p-values for the fixed effects using the `cftest` function from package **multcomp**. The asymptotic p-values are given in Figure 12.3.

```
R> cftest(BtheB_lmer1)

         Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = bdi ~ bdi.pre + time + treatment + drug + length +
    (1 | subject), data = BtheB_long, REML = FALSE, na.action = na.omit)

Linear Hypotheses:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept) == 0      5.59239    2.24244   2.494   0.0126
bdi.pre == 0          0.63968    0.07789   8.212 2.22e-16
time == 0            -0.70476    0.14639  -4.814 1.48e-06
treatmentBtheB == 0 -2.32908    1.67036  -1.394   0.1632
drugYes == 0         -2.82495    1.72684  -1.636   0.1019
length>6m == 0        0.19708    1.63832   0.120   0.9043
(Univariate p values reported)
```

Figure 12.3 R output of the asymptotic p-values for linear mixed-effects model fit for the `BtheB` data.

We can check the assumptions of the final model fitted to the `BtheB` data, i.e., the normality of the random effect terms and the residuals, by first using the `ranef` method to *predict* the former and the `residuals` method to calculate the differences between the observed data values and the fitted values, and then using normal probability plots on each. How the random effects are predicted is explained briefly in Section **??**. The necessary R code to obtain the effects, residuals and plots is shown with Figure 12.4. There appear to be no large departures from linearity in either plot.

```
R> layout(matrix(1:2, ncol = 2))
R> qint <- ranef(BtheB_lmer1)$subject[["(Intercept)"]]
R> qres <- residuals(BtheB_lmer1)
R> qqnorm(qint, ylab = "Estimated random intercepts",
+         xlim = c(-3, 3), ylim = c(-20, 20),
+         main = "Random intercepts")
R> qqline(qint)
R> qqnorm(qres, xlim = c(-3, 3), ylim = c(-20, 20),
+         ylab = "Estimated residuals",
+         main = "Residuals")
R> qqline(qres)
```

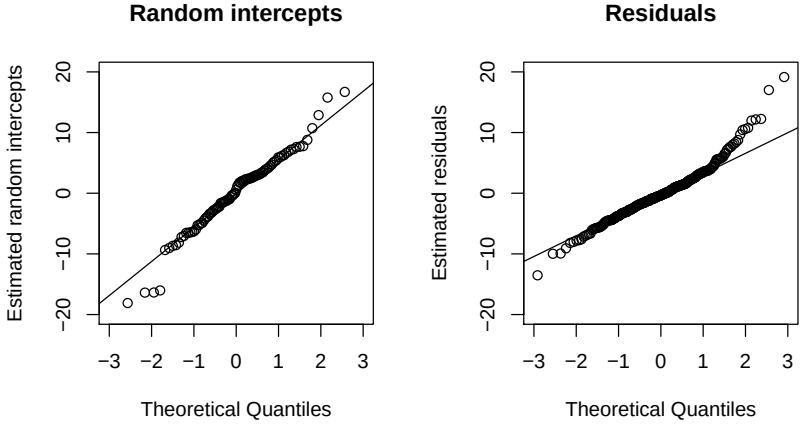

Figure 12.4 Quantile-quantile plots of predicted random intercepts and residuals for the random intercept model `BtheB_lmer1` fitted to the `BtheB` data.

```
R> bdi <- BtheB[, grep("bdi", names(BtheB))]
R> plot(1:4, rep(-0.5, 4), type = "n", axes = FALSE,
+       ylim = c(0, 50), xlab = "Months", ylab = "BDI")
R> axis(1, at = 1:4, labels = c(0, 2, 3, 5))
R> axis(2)
R> for (i in 1:4) {
+      dropout <- is.na(bdi[,i + 1])
+      points(rep(i, nrow(bdi)) + ifelse(dropout, 0.05, -0.05),
+             jitter(bdi[,i]), pch = ifelse(dropout, 20, 1))
+  }
```

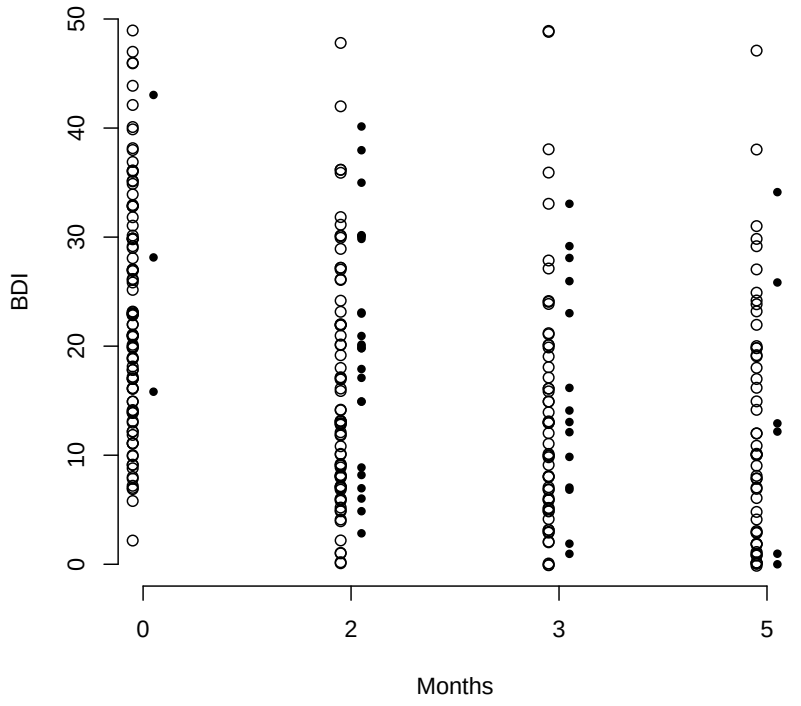

Figure 12.5 Distribution of BDI values for patients that do (circles) and do not (bullets) attend the next scheduled visit.

Bibliography

Bates, D. (2005), "Fitting linear mixed models in R," *R News*, 5, 27–30, URL `http://CRAN.R-project.org/doc/Rnews/`.

Bates, D. and Sarkar, D. (2012), ***lme4****: Linear Mixed-Effects Models Using S4 Classes*, URL `http://CRAN.R-project.org/package=lme4`, R package version 0.999375-42.

Pinheiro, J. C. and Bates, D. M. (2000), *Mixed-Effects Models in S and S-PLUS*, New York, USA: Springer-Verlag.