2008 Q2

November

(a)

This is a 2^k fractional factorial design, so we need to specify the contrast to be `contr.sum`.

```
# read and preprocess data: factorization
d <- read.table('circuit.txt', header = T)
cols <- names(d)[1:8]
d[cols] <- lapply(d[cols], factor)
# set contrast option
options(contrasts = c('contr.sum', 'contr.poly'))
```

```
# check the alias structure
m <- lm(y ~ blocks + a*b*c*d*e*f*g, data = d)
dr <- alias(m)$Complete[, "blocks1"]
dr[dr %in% c("1", "-1")]
```

```
##          b1:c1:d1          a1:d1:f1       b1:c1:e1:f1       d1:e1:f1:g1
##                 1                 1                -1                -1
##    a1:b1:c1:f1:g1 a1:b1:c1:d1:e1:g1                g1             a1:e1
##                 1                -1                 1                -1
```

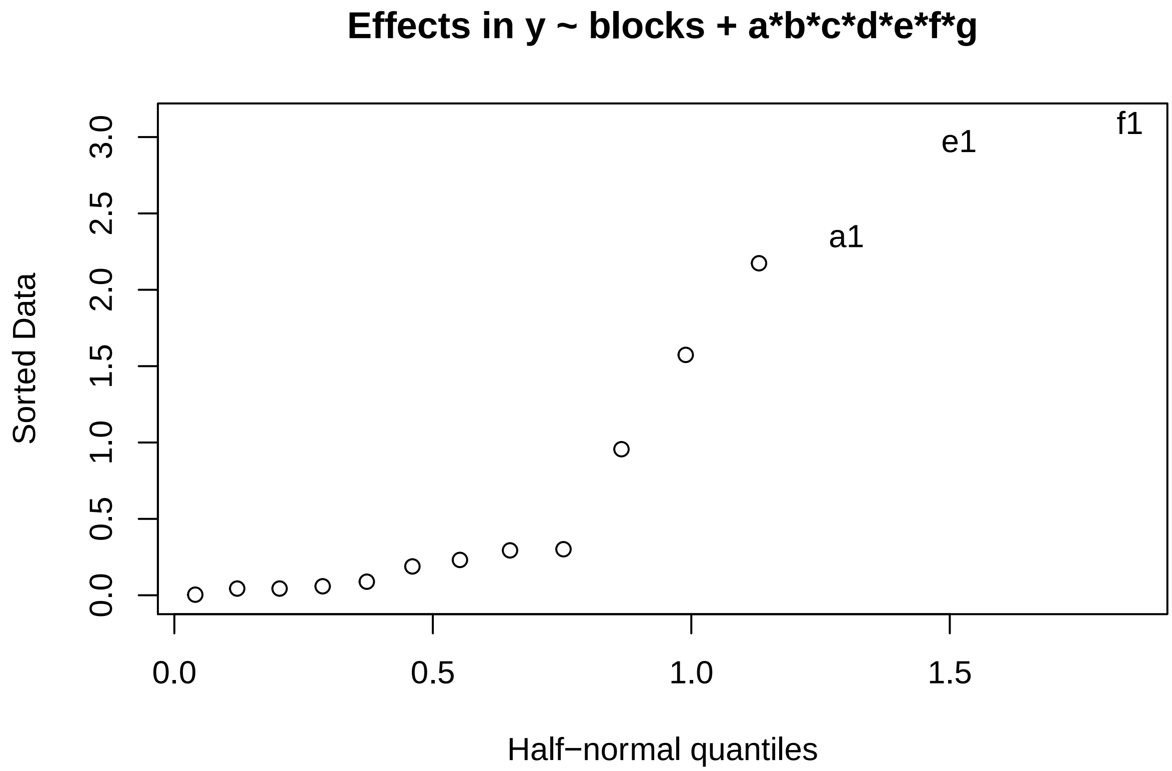
```
# try to find a better block generator
m0 <- lm(y ~ a*b*c*d*e*f*g, data = d)
aliases(m0)
```

```
##
##  a = e:g = b:c:f = d:f:g = b:c:d:e = a:d:e:f = a:b:c:d:g = a:b:c:e:f:g
##  b = a:c:f = c:d:g = a:c:d:e = b:d:e:f = a:b:e:g = c:e:f:g = a:b:d:f:g
##  c = a:b:f = b:d:g = a:b:d:e = c:d:e:f = a:c:e:g = b:e:f:g = a:c:d:f:g
##  d = e:f = b:c:g = a:f:g = a:b:c:e = a:d:e:g = a:b:c:d:f = b:c:d:e:f:g
##  e = d:f = a:g = a:b:c:d = b:c:f:g = a:b:c:e:f = b:c:d:e:g = a:d:e:f:g
##  f = d:e = a:b:c = a:d:g = b:c:e:g = a:e:f:g = b:c:d:f:g = a:b:c:d:e:f
##  g = b:c:d = a:d:f = b:c:e:f = d:e:f:g = a:b:c:f:g = a:b:c:d:e:g = a:e
##  a:b = c:f = c:d:e = b:e:g = a:c:d:g = b:d:f:g = a:b:d:e:f = a:c:e:f:g
##  a:c = b:f = b:d:e = c:e:g = a:b:d:g = c:d:f:g = a:c:d:e:f = a:b:e:f:g
##  b:c = a:f = d:g = a:d:e = e:f:g = b:c:d:e:f = a:b:c:e:g = a:b:c:d:f:g
##  a:d = f:g = b:c:e = a:e:f = d:e:g = b:c:d:f = a:b:c:g = a:b:c:d:e:f:g
##  b:d = c:g = a:c:e = b:e:f = a:c:d:f = a:b:f:g = a:b:d:e:g = c:d:e:f:g
##  c:d = b:g = a:b:e = c:e:f = a:b:d:f = a:c:f:g = a:c:d:e:g = b:d:e:f:g
##  b:e = a:c:d = b:d:f = a:b:g = c:f:g = a:c:e:f = c:d:e:g = a:b:d:e:f:g
##  c:e = a:b:d = c:d:f = a:c:g = b:f:g = a:b:e:f = b:d:e:g = a:c:d:e:f:g
```

The block generator used in the experiment is `BCD`, and the issue is that it will confound with the main effect `G`, which is undesirable. To find a block generator that will not confound with main effects, and even not with two-way interaction terms, we can check the alias structure. From the results, we can see that `ACD` or `ABD` would be good choices, since they would confound with only one two-way interaction term, and the others are three-way or higher order.

(b)

```
# check the halfnorm plot
halfnorm(-2 * coef(m)[-1], labs = names(coef(m)[-1]), nlab = 3)
title(main = "Effects in y ~ blocks + a*b*c*d*e*f*g")
```




```r
# aliases of A, E, F
dr <- alias(m)$Complete[, "a1"]
dr[dr %in% c("1", "-1")]
```

```
##            e1:g1          b1:c1:f1          d1:f1:g1       b1:c1:d1:e1
##               -1                 1                 1                -1
##       a1:d1:e1:f1    a1:b1:c1:d1:g1 a1:b1:c1:e1:f1:g1
##               -1                 1                -1
```

```r
dr <- alias(m)$Complete[, "e1"]
dr[dr %in% c("1", "-1")]
```

```
##          d1:f1          a1:g1    a1:b1:c1:d1    b1:c1:f1:g1 a1:b1:c1:e1:f1
##             -1             -1             -1             -1              1
## b1:c1:d1:e1:g1 a1:d1:e1:f1:g1
##              1              1
```

```r
dr <- alias(m)$Complete[, "f1"]
dr[dr %in% c("1", "-1")]
```

```
##            d1:e1          a1:b1:c1          a1:d1:g1       b1:c1:e1:g1
##               -1                 1                 1                -1
##       a1:e1:f1:g1    b1:c1:d1:f1:g1 a1:b1:c1:d1:e1:f1
##               -1                 1                -1
```

The top3 promising main effects are: A, E, F. For this problem, I assume all three-way interactions terms are not important. Factor A is aliased with EG, factor E is aliased with DF and AG, and factor F is aliased with DE. We can not separate the effect of these three main effects with the two-way interaction effects, so it is likely that these two-way interaction terms are also important. Further analysis should concentrate on A, E, F.

(c)

```r
m1 <- lm(y ~ blocks + a * e * f, data = d)
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value   Pr(>F)
## blocks     1 18.901  18.901 10.5877 0.011634 *
## a          1 22.114  22.114 12.3873 0.007851 **
## e          1 35.373  35.373 19.8148 0.002135 **
## f          1 38.347  38.347 21.4809 0.001678 **
## a:f        1  0.363   0.363  0.2033 0.664006
## e:f        1  0.014   0.014  0.0077 0.932084
## a:e:f      1  0.032   0.032  0.0176 0.897594
## Residuals  8 14.281   1.785
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
alias(m1)
```

```
## Model :
## y ~ blocks + a * e * f
##
## Complete :
##       (Intercept) blocks1 a1 e1 f1 a1:f1 e1:f1 a1:e1:f1
## a1:e1  0          -1       0  0  0  0     0     0
```

```r
m2 <- lm(y ~ blocks + a + e + f, data = d)
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## blocks     1 18.901  18.901  14.153 0.0031432 **
## a          1 22.114  22.114  16.559 0.0018536 **
## e          1 35.373  35.373  26.488 0.0003199 ***
## f          1 38.347  38.347  28.715 0.0002307 ***
## Residuals 11 14.690   1.335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m1, m2)
```

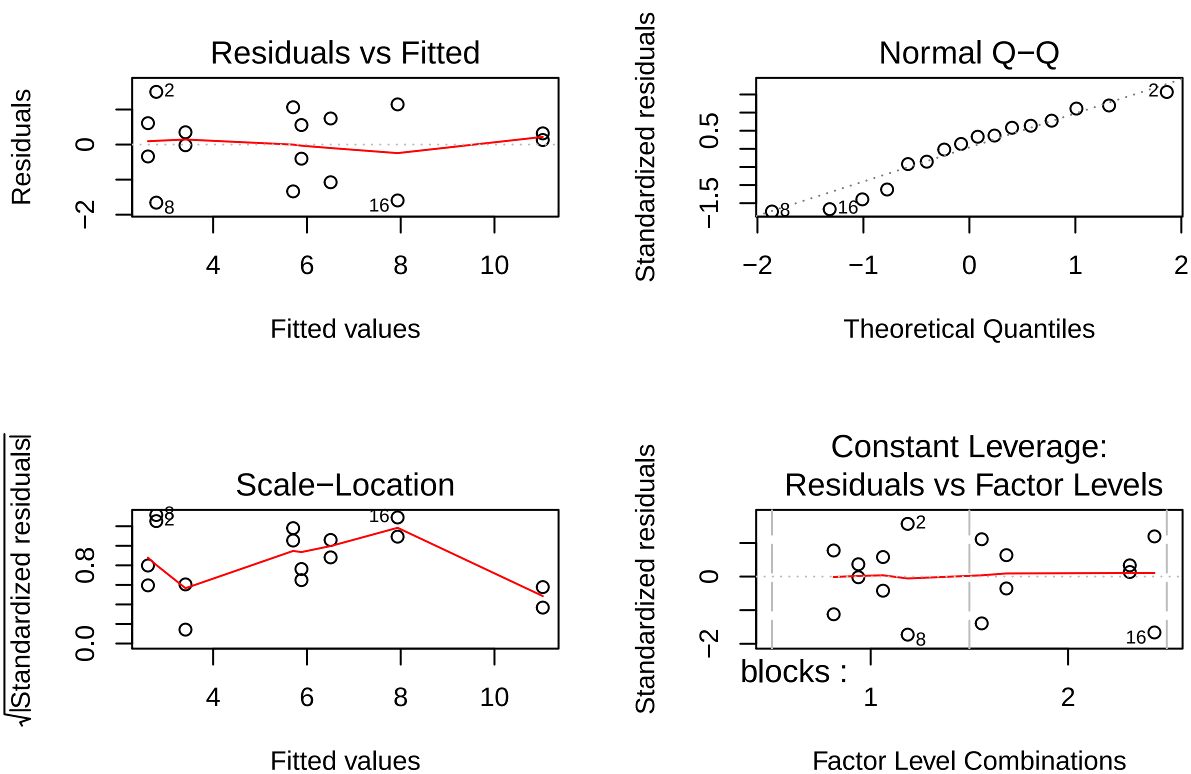
```
## Analysis of Variance Table
##
## Model 1: y ~ blocks + a * e * f
## Model 2: y ~ blocks + a + e + f
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      8 14.281
## 2     11 14.690 -3  -0.40832 0.0762 0.9711
```

```r
summary(m2)
```

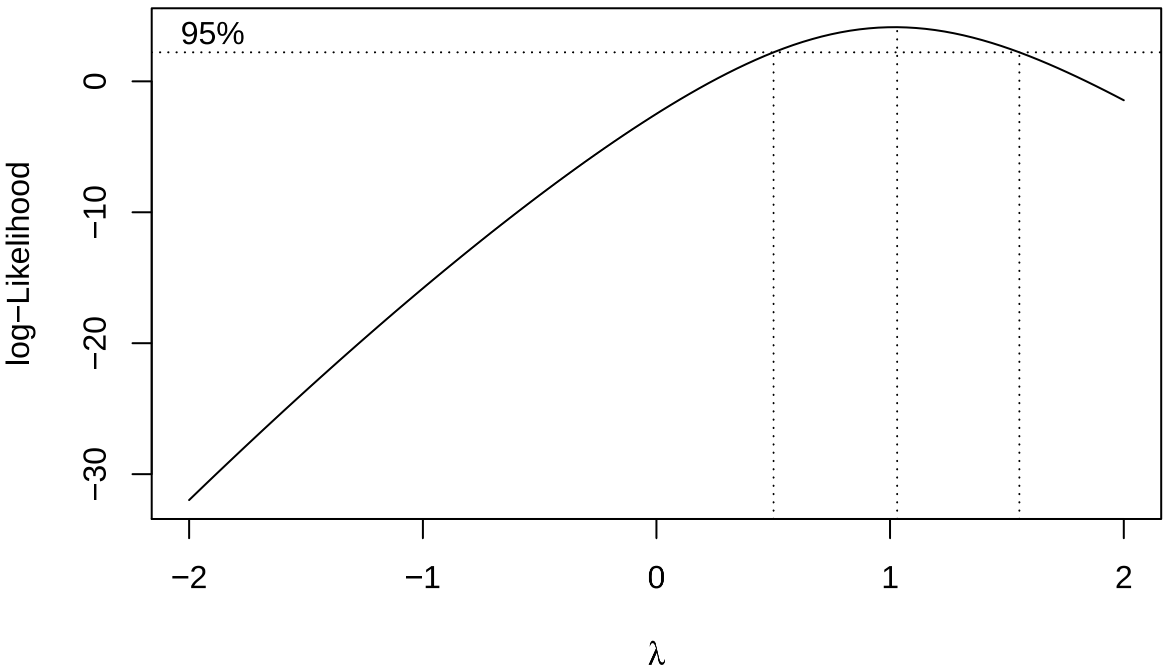
```
##
```

```
## Call:
## lm.default(formula = y ~ blocks + a + e + f, data = d)
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -1.6569 -0.5713  0.2244  0.6441  1.5031
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.7331     0.2889  19.845 5.81e-10 ***
## blocks1      -1.0869     0.2889  -3.762 0.003143 **
## a1           -1.1756     0.2889  -4.069 0.001854 **
## e1           -1.4869     0.2889  -5.147 0.000320 ***
## f1            1.5481     0.2889   5.359 0.000231 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.156 on 11 degrees of freedom
## Multiple R-squared:  0.8865, Adjusted R-squared:  0.8452
## F-statistic: 21.48 on 4 and 11 DF,  p-value: 3.729e-05
```

```r
# model diagnostics
par(mfrow = c(2, 2))
plot(m2)
```




```r
par(mfrow = c(1, 1))
boxcox(m2)
```




```r
linear.contrast(m2, term = a, contr.coefs = c(1, -1))
```

```
##   estimates        se   t-value     p-value  lower-ci  upper-ci
## 1  -2.35125 0.5778028 -4.069295 0.001853647 -3.622985 -1.079515
```

```r
linear.contrast(m2, term = e, contr.coefs = c(1, -1))
```

```
##   estimates        se   t-value      p-value  lower-ci  upper-ci
## 1  -2.97375 0.5778028 -5.146652 0.0003199055 -4.245485 -1.702015
```

```r
linear.contrast(m2, term = f, contr.coefs = c(1, -1))
```

```
##   estimates        se  t-value    p-value lower-ci upper-ci
## 1   3.09625 0.5778028 5.358662 0.00023074 1.824515 4.367985
```

Note that effect AE confounds with the block, so we cannot separate these two. The ANOVA table shows that the interaction terms are not significant, so I delete them and refit the model. Comparing `m1` and `m2`, I would use `m2` as the final model.

All three main effects are significant, with p-value being . . . .

The model diagnostic results look good.

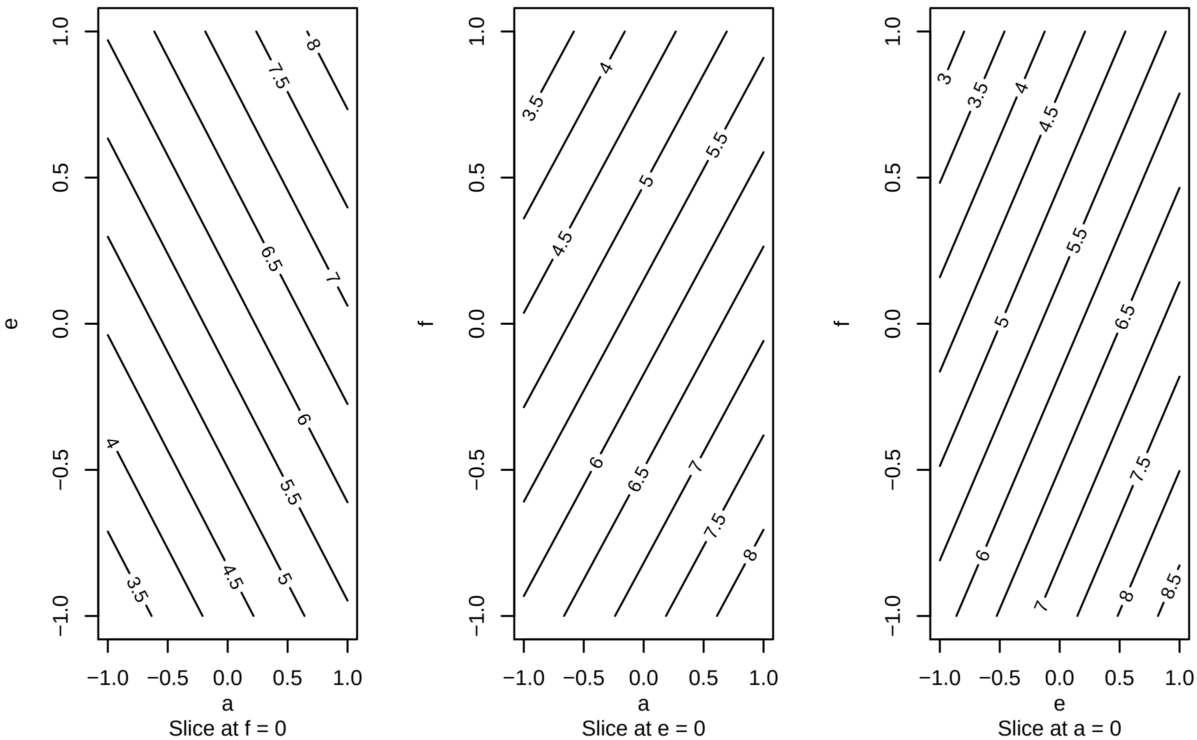
(d)

```r
# response surface
dd <- within(d, {
  a <- ifelse(a=='-1', -1, 1)
  e <- ifelse(e=='-1', -1, 1)
  f <- ifelse(f=='-1', -1, 1)
})
m.rsm <- rsm(y ~ blocks + FO(a, e, f), data = dd)
summary(m.rsm)
```

```
##
## Call:
## rsm(formula = y ~ blocks + FO(a, e, f), data = dd)
```

```
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   5.7331     0.2889 19.8446 5.811e-10 ***
## blocks1      -1.0869     0.2889 -3.7621 0.0031432 **
## a             1.1756     0.2889  4.0693 0.0018536 **
## e             1.4869     0.2889  5.1467 0.0003199 ***
## f            -1.5481     0.2889 -5.3587 0.0002307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.8865, Adjusted R-squared:  0.8452
## F-statistic: 21.48 on 4 and 11 DF,  p-value: 3.729e-05
##
## Analysis of Variance Table
##
## Response: y
##             Df Sum Sq Mean Sq F value   Pr(>F)
## blocks       1 18.901  18.901 14.1534 0.003143
## FO(a, e, f)  3 95.833  31.944 23.9208 4.02e-05
## Residuals   11 14.690   1.335
## Lack of fit  3  0.408   0.136  0.0762 0.971079
## Pure error   8 14.281   1.785
##
## Direction of steepest ascent (at radius 1):
##          a          e          f
##  0.4803641  0.6075418 -0.6325688
##
## Corresponding increment in original units:
##          a          e          f
##  0.4803641  0.6075418 -0.6325688
```

```
par(mfrow = c(1, 3))
contour(m.rsm, ~ a + e + f)
```




```
par(mfrow = c(1, 1))
```

From the contour plot, I don't think ≤ 1.5 is achievable with current settings.

(e)

(Note: For clarity of the report, part of the output results are hidden. And the utilized packages are as shown in the Appendix.)

Reminder

- Read through the problem, understand the problem, write down the solution sketch and highlight note, check understanding, then start coding.
- If there are covariates, start with plotting scatter plots.

Checklist

(1) Randomized or observational? Fit an adjusted or unadjusted model?
(2) Balanced or unbalanced?
(3) Data preprocessing. Factorization.
(4) For change from baseline problems, do remember to remove the baseline observation.
(5) `contr.sum` or `contr.treatment`? If 2^k factorial, we have to use `contr.sum`.
(6) Fit a large model, check model assumptions: typical 4 plots, residual plots, qqnorm for error, qqnorm for random effects, boxcox for transformation. If specified, check outliers. Check P17.1 in the assignments.R for fancy plots. But note that we need to refit the model and check whether the inference will change in order to determine outliers.
(7) Correlation structure? Which response to use? Interaction and polynomial terms? Random slope or random intercept? Try and use model diagnostics to help choose.
(8) If there are covariates, plot the scatter plot of the response vs. covariate. Check lecture code Chapter 17. Center the covariate for better interpretation.
(9) `anova()` or `Anova(, type = 2)`?
(10) Multiple comparison. Use `glht()`, use `linear.contrast()` and `fit.contrast()` to check. If using `intervals()`, then the contrast option needs to be `contr.treatment`.
(11) Copy the library code block to the appendix.

(12) Follow-up or use diff?
(13) If possible, plot fitted to check the goodness of the model.
(14) Always use `REML` to make inference.
(15) Do not factorized `fu`.
(16) Check previous problems to guide writing.
(17) Model selection: interaction terms, quadratic terms, random slope/intercept, correlation structure. AIC, or manual.
(18) Show all your findings and considerations, let the graders know your understandings.

Notes (1) `Anova(type=2)` means independent effect. (2) We can treat `month` as factor or numerics for variance reduction. Factor is more general. And if we want to try random slope, then we need to use numerics.

As shown in Figure 1a in Appendix, there is a non-linear relationship between `age` and `np.chg`, so I include the quadratic term. Also, for easier interpretation, I center `age` around 40.

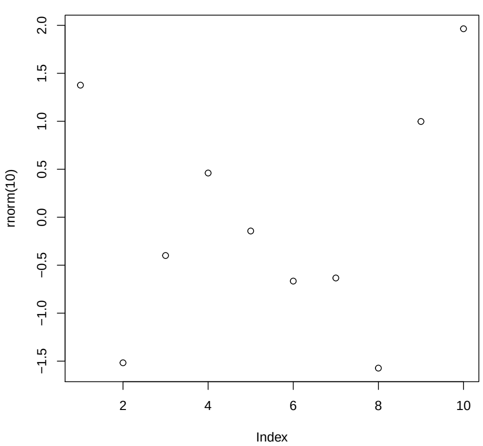
Packages

All R packages used in this problem are listed below.

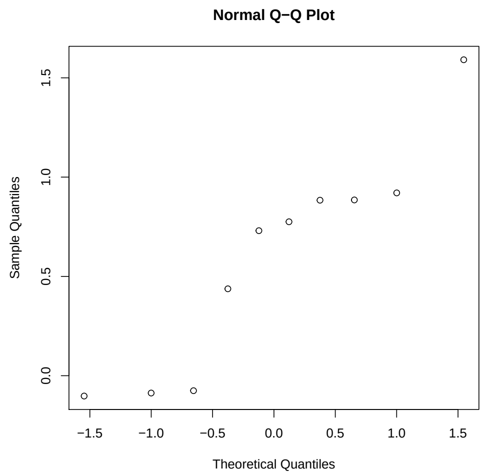
```r
library(gmodels)
library(MASS)
library(car)
library(dplyr)
```

Appendix

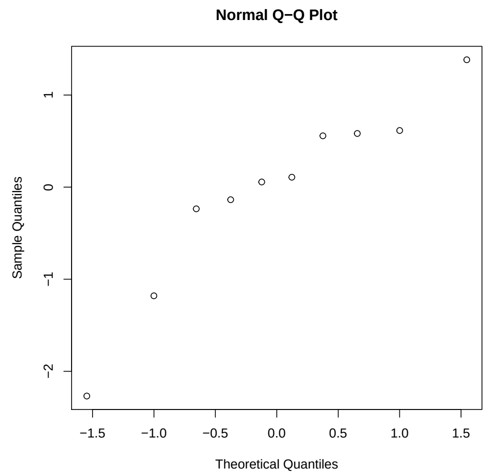
Figures



(a) residual plot of m.lme4



(b) qqnorm of residuals



(c) qqnorm of random effects

Figure 1: Model Diagnostics