Problem 1 Continue working with the denim dataset and build a model with supplier as a random effect (full data). a. Test whether a random effect should be included in the model in two equivalent ways: i. Write a code from scratch where you obtain 1000 bootstrap samples from your model's response under the null hypothesis (no random effects present), refit the model with each simulated response, and obtain a likelihood ratio test statistic for each replication. Then, compare them with the one obtain from the original problem for an empirical p-value.

→ the p-value decreased to 1.

```
library(faraway)
library(faraway)
library(lme4)
```

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

```
library(ggplot2)
data(coagulation, package="faraway")
de <- data.frame(denim)
# REML and MLE
m0 <- lm(waste ~ 1, de)
mm1 <- lmer(waste ~ 1+(1|supplier), de) # REML
mm1.ml <- lmer(waste ~ 1+(1|supplier), de, REML=FALSE) # MLE
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(mm1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: waste ~ 1 + (1 | supplier)
##    Data: de
##
## REML criterion at convergence: 702.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9095 -0.4363 -0.1669  0.3142  6.3817
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  supplier (Intercept)  0.6711  0.8192
##  Residual             97.3350  9.8658
## Number of obs: 95, groups:  supplier, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    6.997      1.078    6.49
```

```
summary(mm1.ml)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: waste ~ 1 + (1 | supplier)
##    Data: de
##
##      AIC      BIC   logLik deviance df.resid
##    710.0    717.7   -352.0    704.0       92
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8877 -0.4498 -0.1806  0.3021  6.4246
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  supplier (Intercept)  0.00    0.000
##  Residual             96.84    9.841
## Number of obs: 95, groups:  supplier, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    6.977      1.010    6.91
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
# Testing for random effect "operator"

lrt.1 <- as.numeric(2*(logLik(mm1.ml)-logLik(m0))) # lrt
pvalue <- pchisq(lrt.1, 1, lower=FALSE)
data.frame(lrt.1, pvalue)
```

```
##          lrt.1    pvalue
## 1 1.136868e-13 0.9999997
```

```
## Parametric Bootstrap
library(faraway)
library(lme4)

y <- simulate(m0) #Simulate from the distribution under the null
lrt.vec <- numeric(1000)
set.seed(123)
for(i in 1:1000){
  y <- unlist(simulate(m0))
  b0 <- lm(y ~ 1)
  b1 <- lmer(y ~ 1 + (1|supplier), de, REML=FALSE)
  lrt.vec[i] <- as.numeric(2*(logLik(b1)-logLik(b0)))
}
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(lrt.vec)
```

```
##     Min.  1st Qu.  Median    Mean  3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.21631 0.01096 9.89373
```

```
mean(lrt.vec < 0.00001)
```

```
## [1] 0.722
```

```
phat = mean(lrt.vec > lrt.1)
phat
```

```
## [1] 0.278
```

```
sqrt(phat*(1-phat)/1000)
```

```
## [1] 0.01416743
```

\ ii. Repeat this process using exactRLRT(). To receive full credit, provide the appropriate hypotheses and conclusion to your test.

```
## Testing Random Effects

library(RLRsim)
#exactLRT(mm1.ml, m0)
exactRLRT(mm1) # works when a single random effect is used in the model
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.029383, p-value = 0.3481
```

\ Note: Section 10.2 provides an example similar to what you are asked to do here. b. Compute confidence intervals for the random effect SDs. To do this, use parametric bootstrap; specifically: i. Write a code from scratch where you obtain 1000 bootstrap samples from your model's response, refit these models with these simulated responses (based on your model) and extract sigmahat, the estimated standard deviation for the random effect, for each replication. Then, use the vector of estimates to construct a 98% confidence interval.

```
######################################
# Confidence and Prediction Intervals
######################################

## Predicting random effects and
# comparing with fixed effects


m1 <- lm(waste ~ supplier, de)
model.tables(aov(m1)) # supplier as fixed effect
```

```
## Tables of effects
##
##  supplier
##          1      2      3       4    5
##     -2.454  1.855 -2.145  0.5126  3.4
## rep 22.000 22.000 19.000 19.0000 13.0
```

```r
ranef(mm1) # supplier as random effect
```

```
## $supplier
##   (Intercept)
## 1 -0.32586905
## 2  0.24163713
## 3 -0.25080763
## 4  0.05703166
## 5  0.27800790
##
## with conditional variances for "supplier"
```

```r
(cc <- model.tables(aov(m1)))
```

```
## Tables of effects
##
##  supplier
##          1      2      3       4    5
##     -2.454  1.855 -2.145  0.5126  3.4
## rep 22.000 22.000 19.000 19.0000 13.0
```

```r
cc[[1]]$supplier/ranef(mm1)$supplier #shrinkage estimators
```

```
##   (Intercept)
## 1    7.530985
## 2    7.676701
## 3    8.553421
## 4    8.988545
## 5   12.230160
```

```r
# BLUPs
predict(mm1, re.form=~0)[1]
```

```
##       1
## 6.99706
```

```r
predict(mm1, newdata=data.frame(supplier="1"))
```

```
##        1
## 6.671191
```

```r
fixef(mm1)+ranef(mm1)$supplier
```

```
##   (Intercept)
## 1    6.671191
## 2    7.238698
## 3    6.746253
## 4    7.054092
## 5    7.275068
```

```r
##############################
## Confident Intervals
##############################

VarCorr(mm1)
```

```
##  Groups   Name        Std.Dev.
##  supplier (Intercept) 0.81918
```

```
##  Residual             9.86585
```

```
group.sd <-as.data.frame(VarCorr(mm1))$sdcor[1]
resid.sd <-as.data.frame(VarCorr(mm1))$sdcor[2]
pv <- numeric(1000)
for(i in 1:1000){
  y <-unlist(simulate(mm1))
  bmod <-refit(mm1, y)
  pv[i] <-predict(bmod, re.form=~0)[1] + rnorm(n=1, sd=group.sd) +
    rnorm(n=1, sd=resid.sd)
}
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```r
quantile(pv, c(0.01, 0.99))
```

```
##        1%       99%
## -16.14596  31.75727
```

```r
confint(mm1, method = "boot") #direct method, doesn't require for loop
```

```
## Computing bootstrap confidence intervals ...

##
## 264 message(s): boundary (singular) fit: see help('isSingular')

##                2.5 %    97.5 %
## .sig01      0.000000  3.460480
## .sigma      8.398082 11.400223
## (Intercept) 5.024741  9.206378
```

```r
##############################
## Prediction interval
##############################

for(i in 1:1000){
  y <-unlist(simulate(mm1, use.u=TRUE))
  bmod <-refit(mm1, y)
  pv[i] <-predict(bmod, newdata=data.frame(supplier="1")) +
    rnorm(n=1, sd=resid.sd)
}
```

```
## boundary (singular) fit: see help('isSingular')

## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```r
quantile(pv, c(0.01, 0.99))
```

```
##         1%       99%
## -17.57209  29.70712
```

\

ii. Repeat this process using confint() with method=bootstrap and confirm the results obtained are very similar to those in part i. To receive full credit, provide an interpretation for the interval obtained.

```r
confint(mm1, method = "boot") #direct method, doesn't require for loop
```

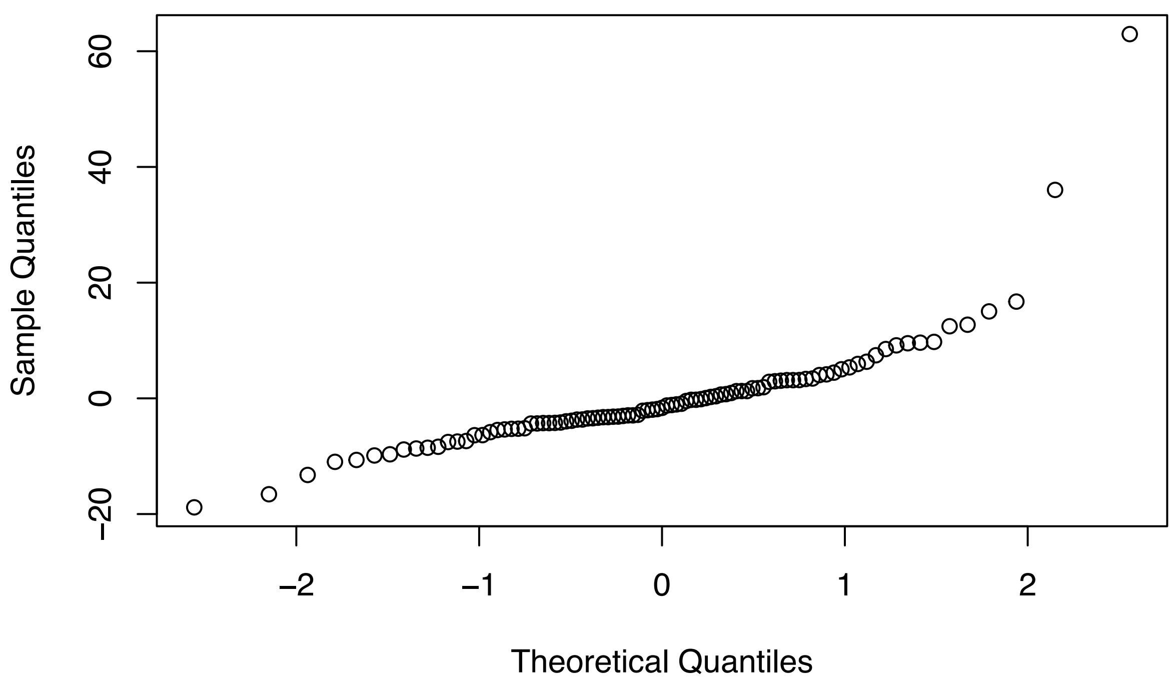
```
## Computing bootstrap confidence intervals ...

##
## 257 message(s): boundary (singular) fit: see help('isSingular')

##                 2.5 %    97.5 %
## .sig01       0.000000  3.571739
## .sigma       8.335020 11.264003
## (Intercept) 4.793705  9.341964
```

With 95% confidence interval, Intercept = [4.7937, 9.34], sig01 = [0, 3.57] and sigma = [8.33, 11.26]

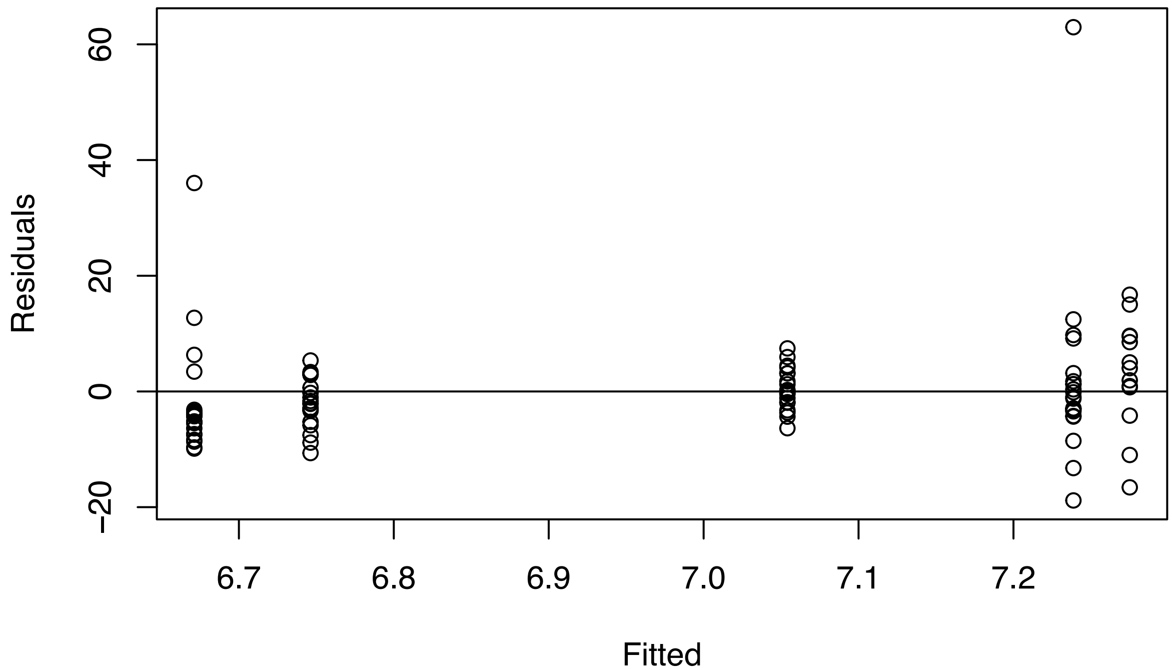
\ c. Estimate the effect of each supplier in your model (Bayesian approach) and if only one supplier will be used, choose the best. In addition, compute the Best Linear Unbiased Predictors for each supplier.

↳ this is in p.17.

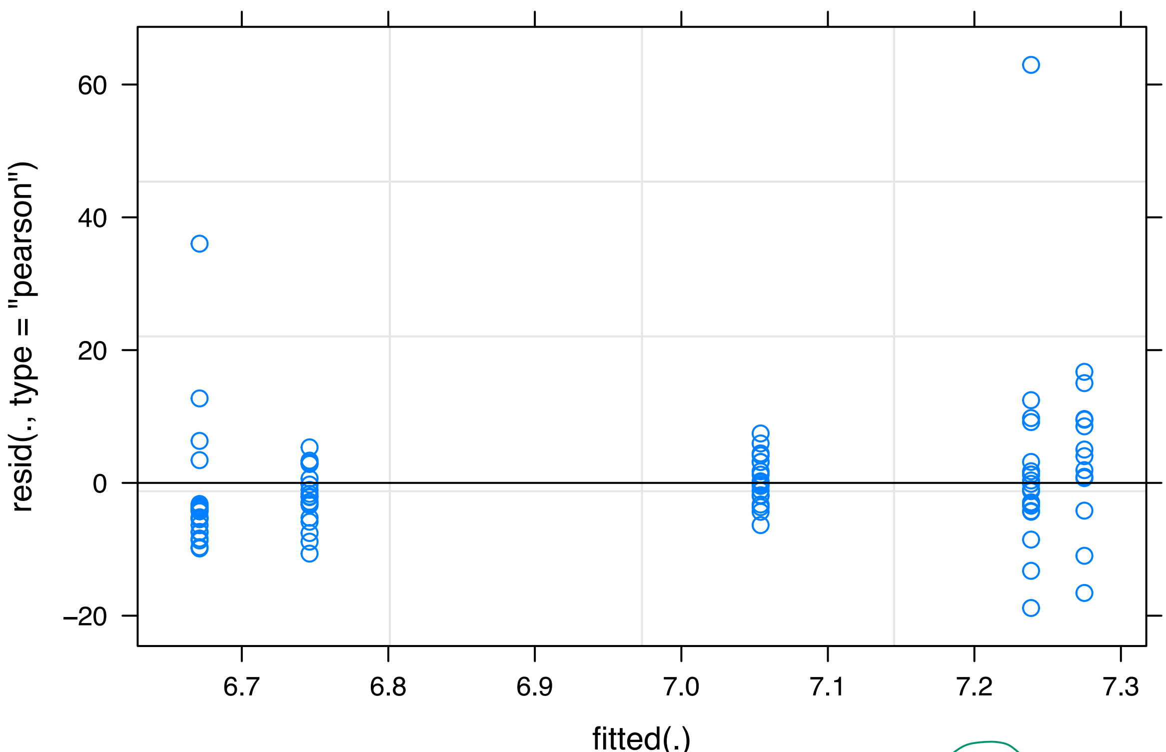
```r
###############################
# Diagnostics
###############################

qqnorm(residuals(mm1),main="")
```




```r
plot(fitted(mm1),residuals(mm1),xlab="Fitted",ylab="Residuals")
abline(h=0)
```



```
plot(mm1)
```



\ d. There are two ouliers in the data. Remove them and redo a.ii., b.ii., and c. Comment on the differences observed.

In next page.

```
library(magrittr)
de2 <- subset(de, resid(mm1)< 20)
dim(de)
```

```
## [1] 95  2
```

```r
dim(de2)
```

```
## [1] 93  2
```

\

```r
# REML and MLE
mm11 <- lmer(waste ~ 1+(1|supplier), de2) # REML
mm11.ml <- lmer(waste ~ 1+(1|supplier), de2, REML=FALSE) # MLE
summary(mm11)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: waste ~ 1 + (1 | supplier)
##    Data: de2
##
## REML criterion at convergence: 603.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.99119 -0.48597 -0.08981  0.49970  2.60002
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  supplier (Intercept)  5.718   2.391
##  Residual             37.292   6.107
## Number of obs: 93, groups:  supplier, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    6.155      1.246   4.938
```

```r
summary(mm11.ml)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: waste ~ 1 + (1 | supplier)
##    Data: de2
##
##      AIC      BIC   logLik deviance df.resid
##    612.0    619.6   -303.0    606.0       90
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.93251 -0.48712 -0.09076  0.51638  2.56200
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  supplier (Intercept)  4.057   2.014
##  Residual             37.325   6.109
## Number of obs: 93, groups:  supplier, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    6.128      1.105   5.547
```

9.841 (p. 2)

From 9.841 to 6.109, it decreases.

#a2 ii. Repeat this process using exactRLRT(). To receive full credit, provide the appropriate hypotheses and conclusion to your test.

```r
## Testing Random Effects

library(RLRsim)
#exactLRT(mm11.ml, m0)
exactRLRT(mm11) # works when a single random effect is used in the model
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 4.5674, p-value = 0.0108
```

//

#b2 b. Compute confidence intervals for the random effect SDs. To do this, use parametric bootstrap; ii. Repeat this process using confint() with method=bootstrap and confirm the results obtained are very similar to those in part i. To receive full credit, provide an interpretation for the interval obtained.

```r
confint(mm11, method = "boot") #direct method, doesn't require for loop
```

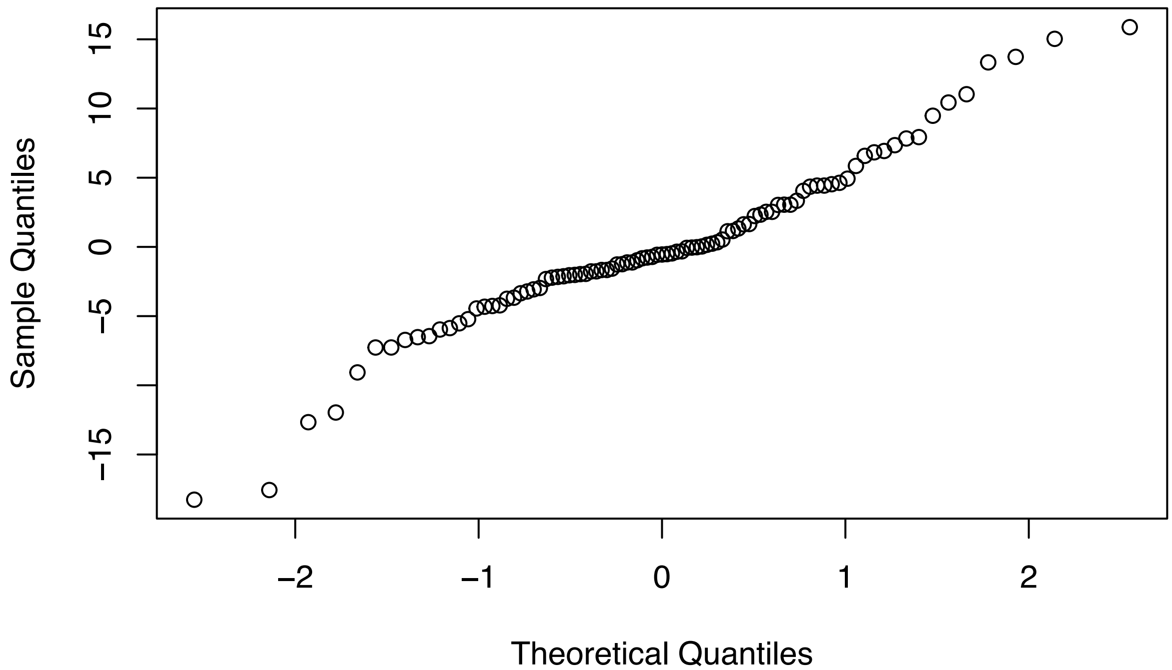
```
## Computing bootstrap confidence intervals ...
```

```
##
## 57 message(s): boundary (singular) fit: see help('isSingular')
```

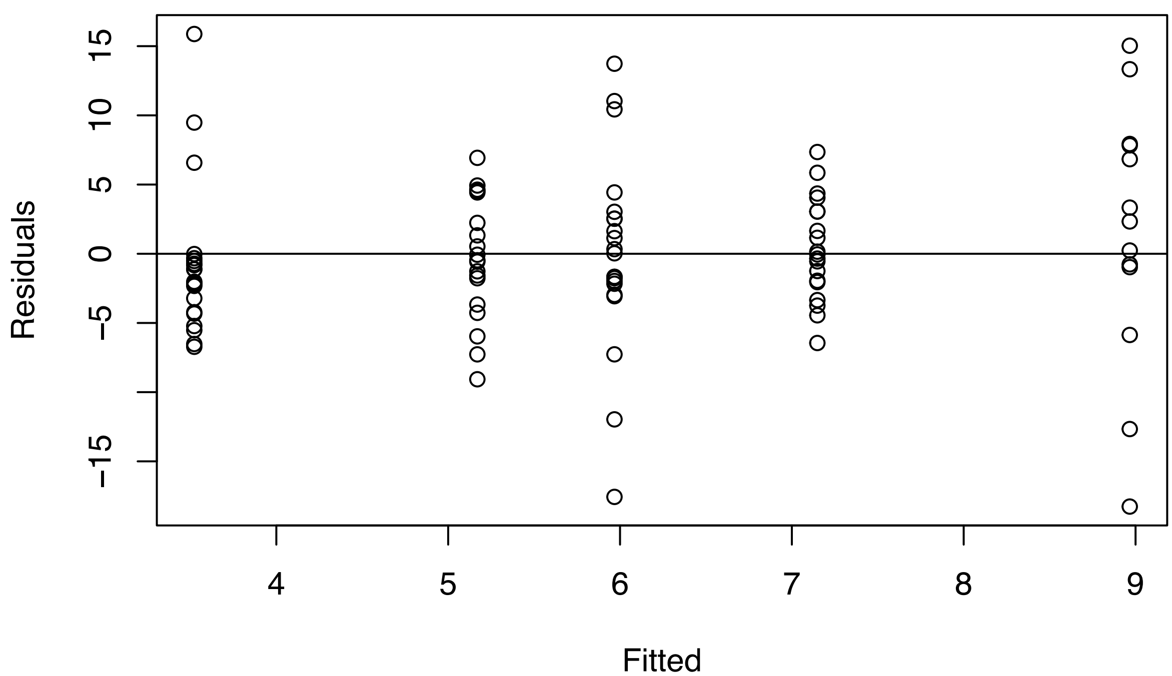
```
##                2.5 %   97.5 %
## .sig01      0.000000 4.469656
## .sigma      5.290828 7.076106
## (Intercept) 3.655324 8.664662
```

// #c

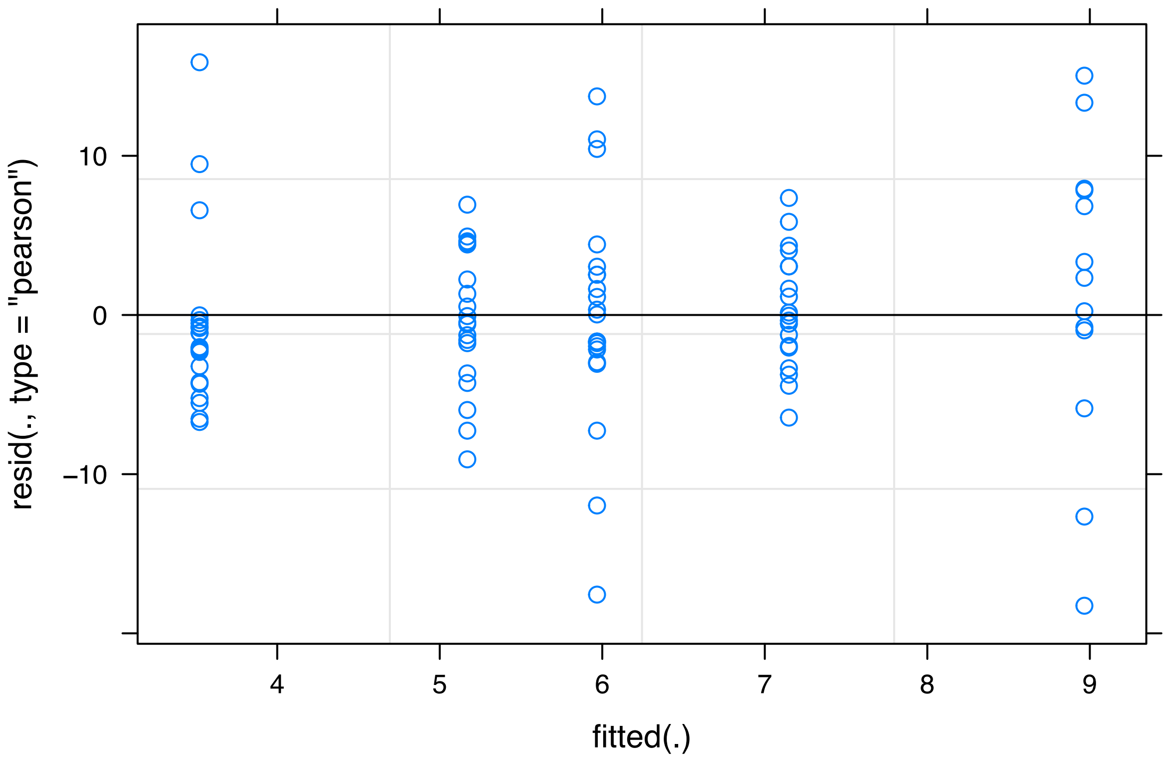
```r
##############################
# Diagnostics
##############################

qqnorm(residuals(mm11),main="")
```




```
plot(fitted(mm11),residuals(mm11),xlab="Fitted",ylab="Residuals")
abline(h=0)
```




```
plot(mm11)
```



\ Problem 2 The coagulation dataset from package faraway comes from a study of blood coagulation times. 24 animals were randomly assigned to four different diets and the samples were taken in a random order. Construct a mixed-effects model using diet as your random effect and do the following: a. Repeat problem 1 a.ii. in this context. b. Repeat problem 1 b.ii. in this context. \

```
data.frame(coagulation, package="faraway")
```

The comments are same with #1.

```
##    coag diet package
## 1    62    A faraway
## 2    60    A faraway
## 3    63    A faraway
## 4    59    A faraway
## 5    63    B faraway
## 6    67    B faraway
## 7    71    B faraway
## 8    64    B faraway
## 9    65    B faraway
## 10   66    B faraway
## 11   68    C faraway
## 12   66    C faraway
## 13   71    C faraway
## 14   67    C faraway
## 15   68    C faraway
## 16   68    C faraway
## 17   56    D faraway
## 18   62    D faraway
## 19   60    D faraway
## 20   61    D faraway
## 21   63    D faraway
## 22   64    D faraway
## 23   63    D faraway
```

```
## 24   59    D faraway
```

```r
summary(coagulation)
```

```
##       coag        diet 
##  Min.   :56.00   A:4  
##  1st Qu.:61.75   B:6  
##  Median :63.50   C:6  
##  Mean   :64.00   D:8  
##  3rd Qu.:67.00        
##  Max.   :71.00
```

```r
# REML and MLE
mm111 <- lmer(coag ~ 1+(1|diet), coagulation) # REML
mm111.ml <- lmer(coag ~ 1+(1|diet), coagulation, REML=FALSE) # MLE
summary(mm111)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: coag ~ 1 + (1 | diet)
##    Data: coagulation
## 
## REML criterion at convergence: 115.8
## 
## Scaled residuals: 
##      Min       1Q   Median       3Q      Max 
## -2.18491 -0.59921  0.09332  0.54078  2.17508 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  diet     (Intercept) 11.692   3.419   
##  Residual              5.599   2.366   
## Number of obs: 24, groups:  diet, 4
## 
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    64.01       1.78   35.96
```

```r
summary(mm111.ml)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: coag ~ 1 + (1 | diet)
##    Data: coagulation
## 
##      AIC      BIC   logLik deviance df.resid 
##    124.6    128.2    -59.3    118.6       21 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -2.2098 -0.6216  0.1244  0.5454  2.1958 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  diet     (Intercept) 8.526    2.920   
##  Residual             5.599    2.366   
## Number of obs: 24, groups:  diet, 4
## 
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   64.016      1.542   41.52
```

#1 a.ii.

ii. Repeat this process using exactRLRT(). To receive full credit, provide the appropriate hypotheses and conclusion to your test.

```
## Testing Random Effects

library(RLRsim)
#exactLRT(mm111.ml, m0)
exactRLRT(mm111) # works when a single random effect is used in the model
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 14.618, p-value = 1e-04
```

\ # 1 b.ii.

b. Compute confidence intervals for the random effect SDs. To do this, use parametric bootstrap;

ii. Repeat this process using confint() with method=bootstrap and confirm the results obtained are very similar to those in part i. To receive full credit, provide an interpretation for the interval obtained.

```
confint(mm111, method = "boot") #direct method, doesn't require for loop
```

```
## Computing bootstrap confidence intervals ...
##
## 15 message(s): boundary (singular) fit: see help('isSingular')
```

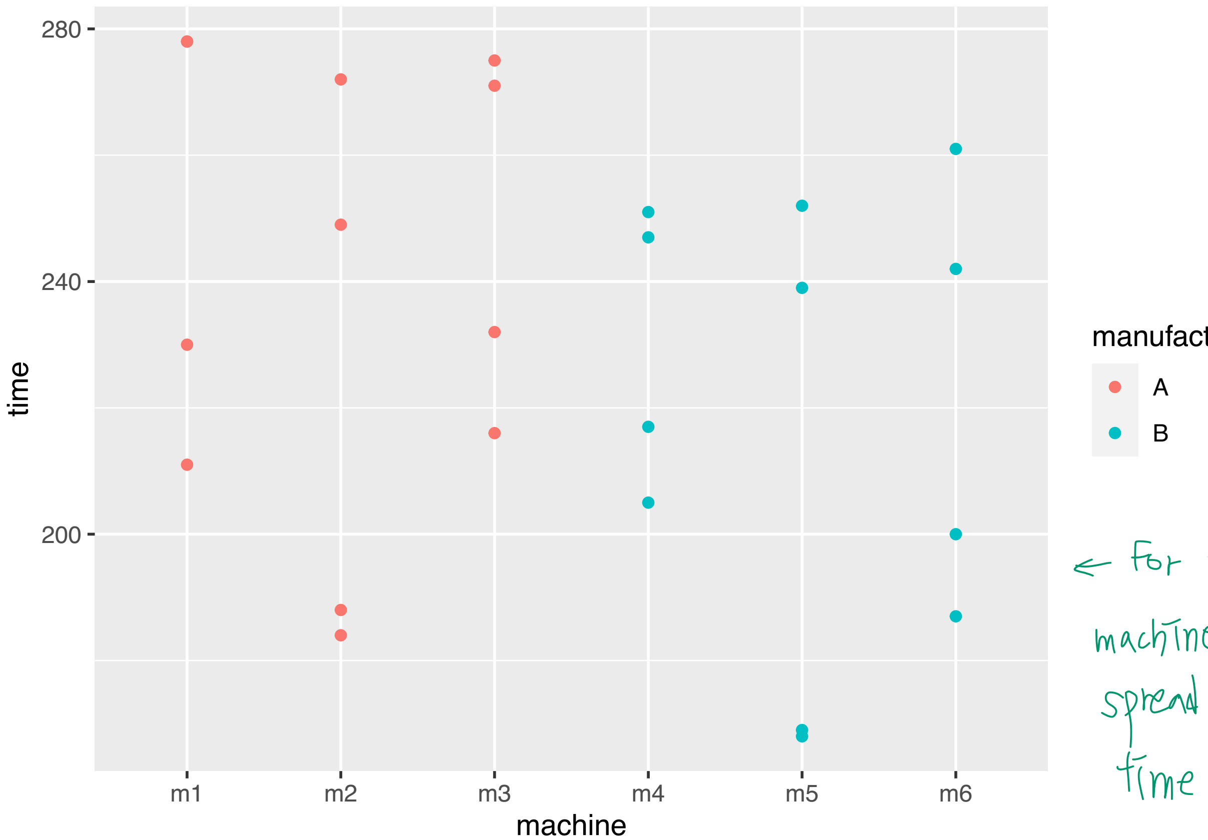
```
##                 2.5 %    97.5 %
## .sig01       0.000000  6.091221
## .sigma       1.612764  2.962358
## (Intercept) 60.232759 67.307839
```

\ Problem 3 Data on the cutoff times of lawnmowers may be found in the dataset lawn (package faraway). Three machines were randomly selected from those produced by manufacturers A and B. Each machine was tested twice at low speed and high speed. a. Make plots of the data and comment.

```
library(faraway)
la = data.frame(lawn)
head(la)
```

```
##   manufact machine speed time
## 1        A      m1     L  211
## 2        A      m1     H  278
## 3        B      m4     L  205
## 4        B      m4     H  247
## 5        A      m1     L  230
## 6        A      m1     H  278
```

```
library(ggplot2)
ggplot(la, aes(x=machine, y=time, color = manufact))+geom_point()
```



← For the "A", machines are spread in high time.

b. Fit a fixed effects model for the cutoff time response using just the main effects of the three predictors. Explain why not all effects can be estimated

← Because the other effects are written in non-numeric values.

```
library(lme4)
#library(Matrix)
mm1 <- lmer(time ~ 1+(1|machine), la)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(mm1) # observe that the variance of the supplier is the same
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: time ~ 1 + (1 | machine)
##    Data: la
##
## REML criterion at convergence: 232.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7671 -0.7496  0.1542  0.6879  1.3639
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  machine  (Intercept)    0      0.00
##  Residual             1234     35.13
## Number of obs: 24, groups:  machine, 6
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)  230.083      7.171   32.08
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

\\\\

c. (c-1)Fit a mixed effects model with manufacturer and speed as main effects along with their interaction and machine as a random effect. (c-2)Using the Laird-Ware notation, write down the model in matrix form and the structure of D, the variance-covariance matrix for the random effects (including its dimension). Also, provide the dimensions of matrices/vectors X,Z, beta, and gamma. \ (c-1)

```
mm2 = lmer(time~manufact*speed + (1|machine),data=la)
summary(mm2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: time ~ manufact * speed + (1 | machine)
##    Data: la
##
## REML criterion at convergence: 168.4
##
## Scaled residuals:
##      Min       1Q  Median       3Q      Max
## -1.0909 -0.6740 -0.1291  0.6661  1.5405
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  machine  (Intercept) 145.2    12.05
##  Residual             132.3    11.50
## Number of obs: 24, groups:  machine, 6
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)      270.500      8.394  32.225
## manufactB        -21.833     11.871  -1.839
## speedL           -60.333      6.641  -9.085
## manufactB:speedL   2.667      9.392   0.284
##
## Correlation of Fixed Effects:
##             (Intr) mnfctB speedL
## manufactB   -0.707
## speedL      -0.396  0.280
## mnfctB:spdL  0.280 -0.396 -0.707
```

(c-2)

L-W notation

$$y = \underbrace{X\beta}_{\text{fixed effect}} + \underbrace{Z\gamma}_{\text{random effect}} + \epsilon$$

Random-effects coefficients

with 94 observations.

y: 94×1

X: 94×3

β: 3×1

Z: 94×2

γ: 2×1

\ (c-2) ~~See the last page.~~

d. Use the model with interactions obtained in the previous part. What would be the SD of the times observed for the following situations:

Answer

e. Assume that the same machine were tested at the same speed.

```
mm3 = lmer(time~manufact*speed + (1|machine) + (1|speed),data=la)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly uniden
##  - Rescale variables?
```

```
summary(mm3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: time ~ manufact * speed + (1 | machine) + (1 | speed)
```

```
##    Data: la
##
## REML criterion at convergence: 168.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0909 -0.6740 -0.1291  0.6661  1.5405
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  machine  (Intercept) 145.2    12.05
##  speed    (Intercept) 114.6    10.71
##  Residual             132.3    11.50
## Number of obs: 24, groups:  machine, 6; speed, 2
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)       270.500     13.604  19.883
## manufactB         -21.833     11.871  -1.839
## speedL            -60.333     16.533  -3.649
## manufactB:speedL    2.667      9.392   0.284
##
## Correlation of Fixed Effects:
##             (Intr) mnfctB speedL
## manufactB   -0.436
## speedL      -0.608  0.112
## mnfctB:spdL  0.173 -0.396 -0.284
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

\ ii. Assume instead that different machines were sampled from the same manufacturer and tested at the same speed once only.

```
mm4 = lmer(time~manufact*speed + (1|machine) + (1|manufact) + (1|speed),data=la)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

```
summary(mm4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: time ~ manufact * speed + (1 | machine) + (1 | manufact) + (1 |
##     speed)
##    Data: la
##
## REML criterion at convergence: 168.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0909 -0.6740 -0.1291  0.6661  1.5405
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev.
##  machine  (Intercept) 145.2    12.05
##  manufact (Intercept) 441.8    21.02
##  speed    (Intercept) 110.4    10.51
##  Residual             132.3    11.50
## Number of obs: 24, groups:  machine, 6; manufact, 2; speed, 2
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)      270.500     24.954  10.840
## manufactB        -21.833     32.008  -0.682
## speedL           -60.333     16.277  -3.707
## manufactB:speedL   2.667      9.391   0.284
##
## Correlation of Fixed Effects:
##             (Intr) mnfctB speedL
## manufactB   -0.641
## speedL      -0.326  0.042
## mnfctB:spdL  0.094 -0.147 -0.288
## optimizer (nloptwrap) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

\ e. Test whether the interaction term of the model can be removed. If so, go on to test the two main fixed effects terms.

```
library(stats)
mm6 = lmer(time~ (1|machine) + (1|manufact) + (1|speed),data=la)
anova(mm6, mm2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: la
## Models:
## mm6: time ~ (1 | machine) + (1 | manufact) + (1 | speed)
## mm2: time ~ manufact * speed + (1 | machine)
##     npar    AIC    BIC   logLik deviance  Chisq Df Pr(>Chisq)
## mm6    5 213.01 218.90 -101.507   203.01
## mm2    6 202.91 209.97  -95.453   190.91 12.107  1  0.0005022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova
```

```
## function (object, ...)
## UseMethod("anova")
## <bytecode: 0x7f8f999f3068>
## <environment: namespace:stats>
```

f. Check whether there is any variation between machines.

```
mm7 = lmer(time~manufact*speed + (1|manufact) + (1|speed),data=la)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Hessian is numerically singular: parameters are not uniquely determined
```

```r
summary(mm4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: time ~ manufact * speed + (1 | machine) + (1 | manufact) + (1 |
##     speed)
##    Data: la
##
## REML criterion at convergence: 168.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0909 -0.6740 -0.1291  0.6661  1.5405
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  machine  (Intercept) 145.2    12.05
##  manufact (Intercept) 441.8    21.02
##  speed    (Intercept) 110.4    10.51
##  Residual             132.3    11.50
## Number of obs: 24, groups:  machine, 6; manufact, 2; speed, 2
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)       270.500     24.954  10.840
## manufactB         -21.833     32.008  -0.682
## speedL            -60.333     16.277  -3.707
## manufactB:speedL    2.667      9.391   0.284
##
## Correlation of Fixed Effects:
##              (Intr) mnfctB speedL
## manufactB   -0.641
## speedL      -0.326  0.042
## mnfctB:spdL  0.094 -0.147 -0.288
## optimizer (nloptwrap) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

\ g. Fit a model with speed as the only fixed effect and manufacturer as a random effect with machines also as a random effect nested within manufacturer. Compare the variability between machines with the variability between manufacturers. Std. Dev. Residual 0.5108

Std. Dev. Residual 0.5108

So they are same

```r
la$sp<- ifelse(la$speed == "L", 0, 1)
mm8 = lmer(sp~1+(1|manufact) ,data=la)
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
summary(mm8)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: sp ~ 1 + (1 | manufact)
##    Data: la
##
## REML criterion at convergence: 37.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -0.9789 -0.9789  0.0000  0.9789  0.9789
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  manufact (Intercept) 0.0000   0.0000
##  Residual             0.2609   0.5108
## Number of obs: 24, groups:  manufact, 2
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   0.5000     0.1043   4.796
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
mm9 = lmer(sp~1+(1|machine) ,data=la)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(mm9)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: sp ~ 1 + (1 | machine)
##    Data: la
##
## REML criterion at convergence: 37.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.9789 -0.9789  0.0000  0.9789  0.9789
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  machine  (Intercept) 0.0000   0.0000
##  Residual             0.2609   0.5108
## Number of obs: 24, groups:  machine, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   0.5000     0.1043   4.796
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
anova(mm8, mm9)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Warning in optwrap(optimizer, devfun, x@theta, lower = x@lower, calc.derivs =
## TRUE, : convergence code 3 from bobyqa: bobyqa -- a trust region step failed to
## reduce q
```

```
## Data: la
## Models:
## mm8: sp ~ 1 + (1 | manufact)
## mm9: sp ~ 1 + (1 | machine)
##     npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## mm8    3 40.838 44.372 -17.419   34.838
## mm9    3 40.838 44.372 -17.419   34.838     0  0
```

\ h. Construct bootstrap confidence intervals for the terms of the previous model. Discuss whether the variability can be ascribed solely to manufacturers or to machines.

```
confint(mm8, method="boot")
```

It is not clear to consider solely on manufacturer or machines, as follows. This is because they are similar results.

```
## Computing bootstrap confidence intervals ...
##
## 335 message(s): boundary (singular) fit: see help('isSingular')
##                 2.5 %    97.5 %
## .sig01      0.0000000 0.2795422
## .sigma      0.3573229 0.6464336
## (Intercept) 0.2900579 0.7097737
```

```
confint(mm9, method="boot")
```

```
## Computing bootstrap confidence intervals ...
##
## 281 message(s): boundary (singular) fit: see help('isSingular')
##                 2.5 %    97.5 %
## .sig01      0.0000000 0.3305760
## .sigma      0.3405341 0.6567326
## (Intercept) 0.2930073 0.6894609
```

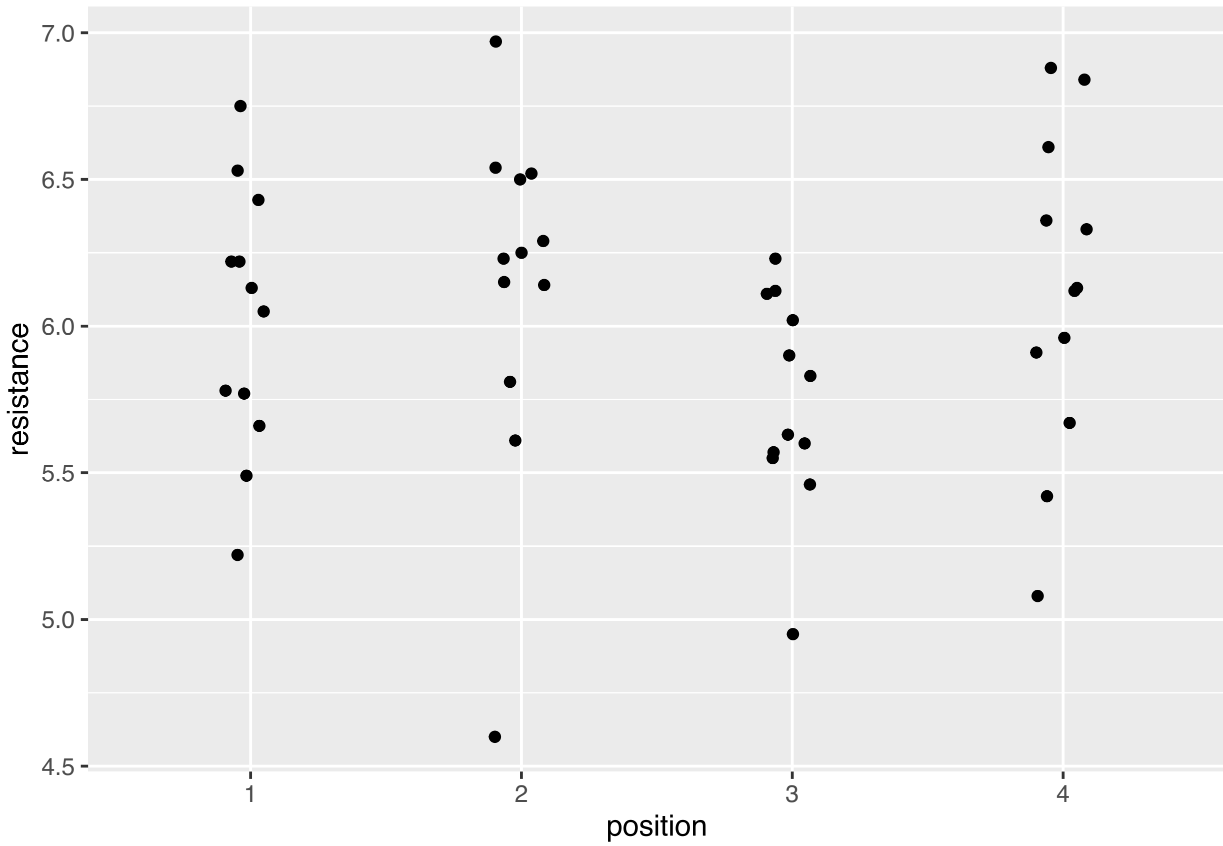
\

Problem 4 An experiment was conducted to optimize the manufacture of semiconductors. The semicond data has the resistance recorded on the wafer as the response. The experiment was conducted during four different time periods denoted by ET and three different wafers (Wafer) during each period. The position on the wafer is a factor with levels 1 to 4. The Grp variable is a combination of ET and Wafer. Analyze the data as a split plot experiment where ET and position are considered as fixed effects. Since the wafers are different in experimental time periods, the Grp variable should be regarded as the block or group variable. a. Plot the data appropriately and comment.

```
se<-data.frame(semicond)
summary(se)
```

```
##    resistance     ET     Wafer  position      Grp
##  Min.   :4.600   1:12   1:16   1:12     1/1    : 4
##  1st Qu.:5.652   2:12   2:16   2:12     1/2    : 4
##  Median :6.115   3:12   3:16   3:12     1/3    : 4
##  Mean   :6.003   4:12          4:12     2/1    : 4
##  3rd Qu.:6.300                          2/2    : 4
##  Max.   :6.970                          2/3    : 4
##                                         (Other):24
```

```
library(ggplot2)
ggplot(se, aes(x=position, y=resistance), color = Grp)+geom_point(position = position_jitter(width=0.1,
```



\ b. Fit a fixed effects model with an interaction between ET and position (no other predictors). What terms are significant? What is wrong with using this model to make inference about these predictors?

```
m0 = lmer(resistance~ET + position +(1|Wafer) ,data=se)
summary(m0)
```

There are no significant terms as follows. "Wafer" can be important terms for predictor.

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: resistance ~ ET + position + (1 | Wafer)
##    Data: se
##
## REML criterion at convergence: 62.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.70953 -0.52345 -0.01676  0.64037  1.77121
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Wafer    (Intercept) 0.02303  0.1518
##  Residual             0.17108  0.4136
## Number of obs: 48, groups:  Wafer, 3
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  5.64375    0.18063  31.244
## ET2          0.34000    0.16886   2.013
## ET3          0.46167    0.16886   2.734
## ET4          0.70667    0.16886   4.185
```

```
## position2    0.11333    0.16886   0.671
## position3   -0.27333    0.16886  -1.619
## position4    0.08833    0.16886   0.523
##
## Correlation of Fixed Effects:
##           (Intr) ET2    ET3    ET4    postn2 postn3
## ET2       -0.467
## ET3       -0.467  0.500
## ET4       -0.467  0.500  0.500
## position2 -0.467  0.000  0.000  0.000
## position3 -0.467  0.000  0.000  0.000  0.500
## position4 -0.467  0.000  0.000  0.000  0.500  0.500
```

```
m1 = lmer(resistance~ET:position +(1|Wafer) ,data=se)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: resistance ~ ET:position + (1 | Wafer)
##    Data: se
##
## REML criterion at convergence: 58.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.30697 -0.57453 -0.07558  0.73383  1.43171
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Wafer    (Intercept) 0.02151  0.1467
##  Residual             0.19543  0.4421
## Number of obs: 48, groups:  Wafer, 3
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    6.54000    0.26891  24.320
## ET1:position1 -0.92667    0.36095  -2.567
## ET2:position1 -0.54667    0.36095  -1.515
## ET3:position1 -0.40333    0.36095  -1.117
## ET4:position1 -0.20000    0.36095  -0.554
## ET1:position2 -1.09000    0.36095  -3.020
## ET2:position2 -0.35333    0.36095  -0.979
## ET3:position2 -0.19333    0.36095  -0.536
## ET4:position2  0.01333    0.36095   0.037
## ET1:position3 -0.98667    0.36095  -2.734
## ET2:position3 -0.77333    0.36095  -2.143
## ET3:position3 -0.76667    0.36095  -2.124
## ET4:position3 -0.64333    0.36095  -1.782
## ET1:position4 -0.65333    0.36095  -1.810
## ET2:position4 -0.62333    0.36095  -1.727
## ET3:position4 -0.44667    0.36095  -1.237
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
```

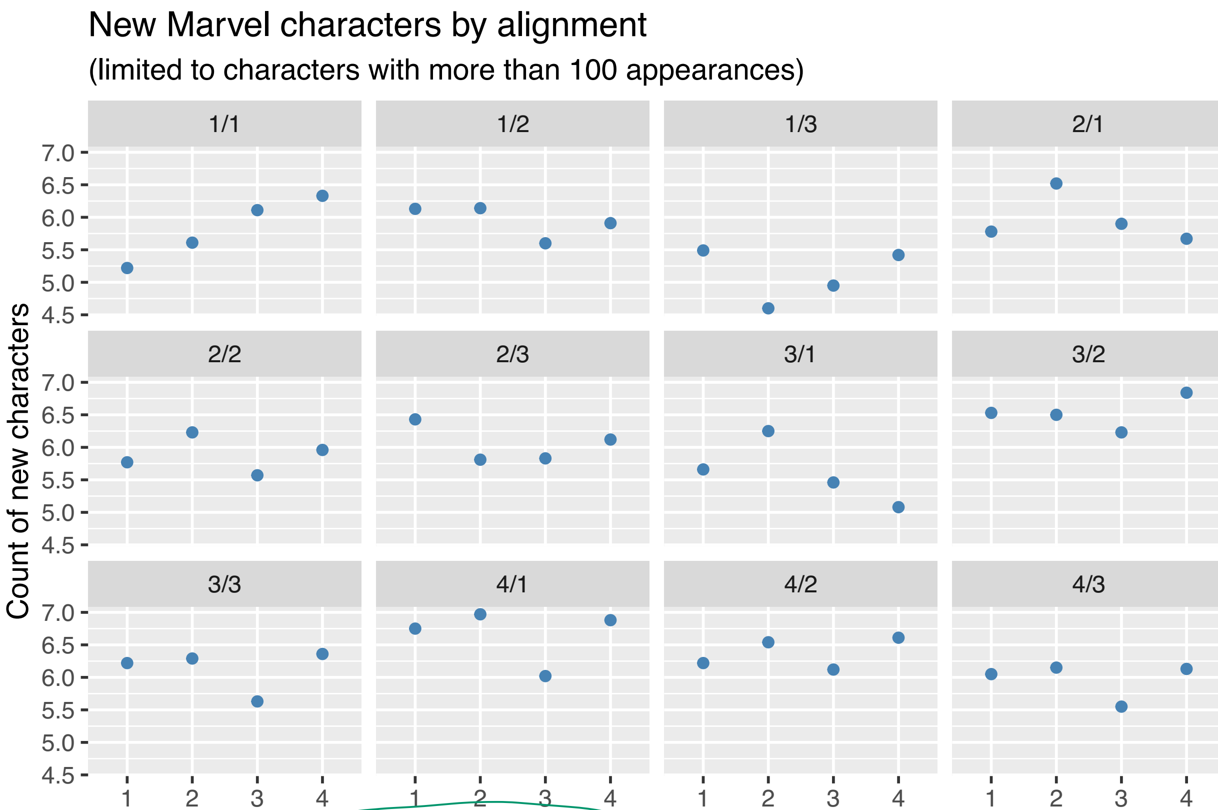
```
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

\ c. (c-1)Fit a model appropriate to the split plot design used here. Comment on the relative variation between and within the groups (Grp). (c-2)Using the Laird-Ware notation, write down the model in matrix form and the structure of D, the variance-covariance matrix for the random effects.

```
library(ggplot2)
ggplot(data = se, aes(position, resistance)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color="steelblue") +
  labs(title = "New Marvel characters by alignment",
       subtitle = "(limited to characters with more than 100 appearances)",
       y = "Count of new characters", x = "") +
  facet_wrap(~ Grp) #Within the Grp=ET*Wafer
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



(c-2) ~~See the last page~~. \ d. Test for the effect of position.

```r
m2 = lmer(resistance~position +(1|Wafer) +(1|ET) ,data=se)
summary(m2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: resistance ~ position + (1 | Wafer) + (1 | ET)
##    Data: se
##
## REML criterion at convergence: 64.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.67581 -0.53824 -0.00129  0.64187  1.62086
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ET       (Intercept) 0.07219  0.2687
##  Wafer    (Intercept) 0.02304  0.1518
##  Residual             0.17108  0.4136
## Number of obs: 48, groups:  ET, 4; Wafer, 3
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  6.02083    0.19996  30.110
## position2    0.11333    0.16886   0.671
## position3   -0.27333    0.16886  -1.619
## position4    0.08833    0.16886   0.523
##
```

(c-2)

L-W notation

Random-effects coefficients

$$y = \underbrace{X\beta}_{\text{fixed effect}} + \underbrace{Z\gamma + \epsilon}_{\text{random effect}}$$

with 94 observations.

y: 48×1

X: 48×3

β: 3×1

Z: 48×3

γ: 3×1

```
## Correlation of Fixed Effects:
##            (Intr) postn2 postn3
## position2 -0.422
## position3 -0.422  0.500
## position4 -0.422  0.500  0.500
```

\ e. Which level of ET results in the highest resistance? Can we be sure that this is really better than the second highest level?

```
m4 = lmer(resistance~ET +(1|Wafer) +(1|position) ,data=se)
summary(m4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: resistance ~ ET + (1 | Wafer) + (1 | position)
##    Data: se
##
## REML criterion at convergence: 61.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5927 -0.4760  0.0070  0.6588  1.5455
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  position (Intercept) 0.01710  0.1308
##  Wafer    (Intercept) 0.02303  0.1518
##  Residual             0.17108  0.4136
## Number of obs: 48, groups:  position, 4; Wafer, 3
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   5.6258     0.1619  34.750
## ET2           0.3400     0.1689   2.013
## ET3           0.4617     0.1689   2.734
## ET4           0.7067     0.1689   4.185
##
## Correlation of Fixed Effects:
##     (Intr) ET2    ET3
## ET2 -0.522
## ET3 -0.522  0.500
## ET4 -0.522  0.500  0.500
```

On the level 4, ET results in the highest level.
But it is not significant statistically.

\ f. Make a plot of the residuals and fitted values and interpret. Make a QQ plot and comment.

```
m5 = lmer(resistance~ET:position +(1|Wafer),data=se)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

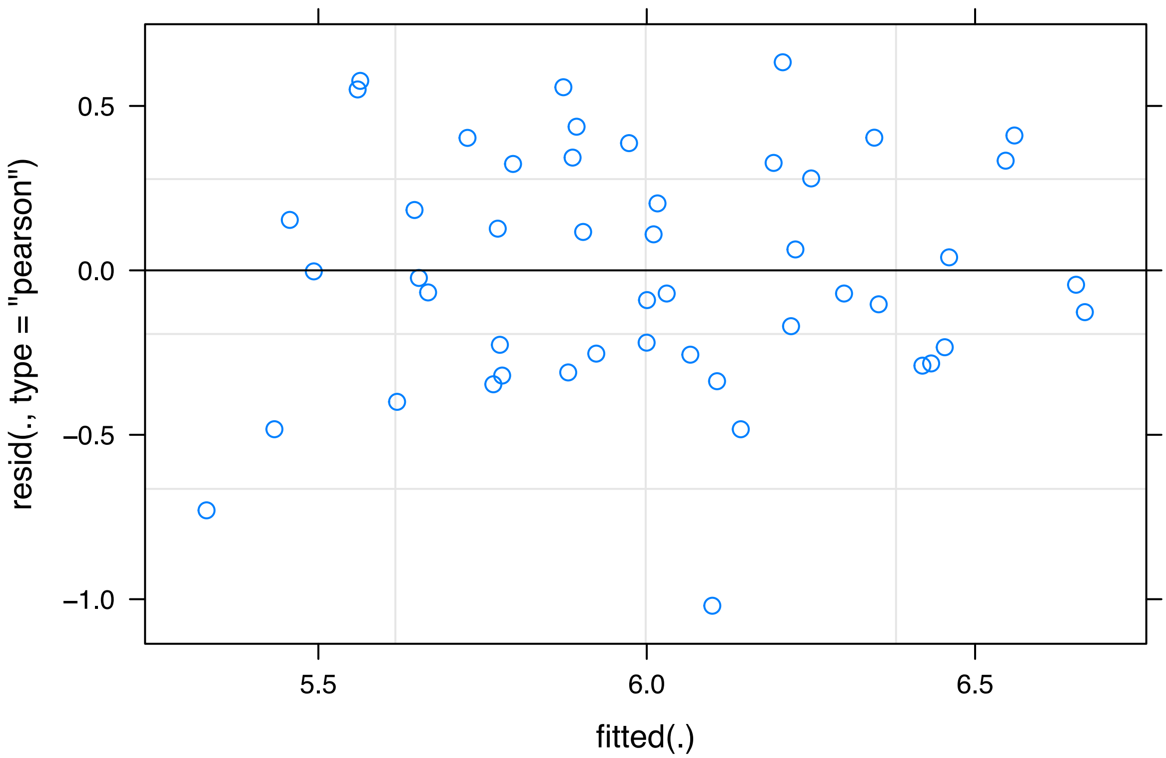
```
summary(m5)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: resistance ~ ET:position + (1 | Wafer)
##    Data: se
##
## REML criterion at convergence: 58.2
##
## Scaled residuals:
```

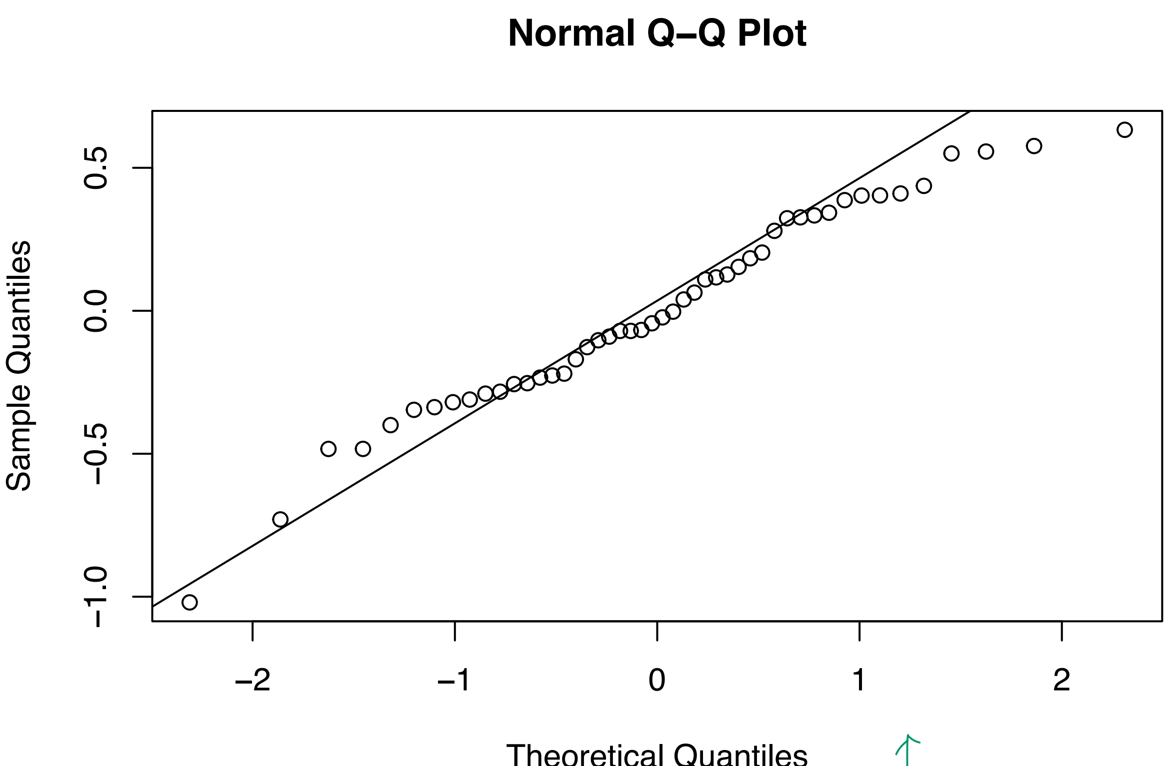
```
##      Min       1Q   Median       3Q      Max
## -2.30697 -0.57453 -0.07558  0.73383  1.43171
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Wafer    (Intercept) 0.02151  0.1467
##  Residual             0.19543  0.4421
## Number of obs: 48, groups:  Wafer, 3
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    6.54000    0.26891  24.320
## ET1:position1 -0.92667    0.36095  -2.567
## ET2:position1 -0.54667    0.36095  -1.515
## ET3:position1 -0.40333    0.36095  -1.117
## ET4:position1 -0.20000    0.36095  -0.554
## ET1:position2 -1.09000    0.36095  -3.020
## ET2:position2 -0.35333    0.36095  -0.979
## ET3:position2 -0.19333    0.36095  -0.536
## ET4:position2  0.01333    0.36095   0.037
## ET1:position3 -0.98667    0.36095  -2.734
## ET2:position3 -0.77333    0.36095  -2.143
## ET3:position3 -0.76667    0.36095  -2.124
## ET4:position3 -0.64333    0.36095  -1.782
## ET1:position4 -0.65333    0.36095  -1.810
## ET2:position4 -0.62333    0.36095  -1.727
## ET3:position4 -0.44667    0.36095  -1.237
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

Not that significant, but "position 1" is the most meaningful to predict.

```
plot(m5)
```




```
qqnorm(resid(m5))
qqline(resid(m5))
```



It follows well the normal distribution.