

Analysis of replication of Grodner English eyetracking data

Shravan Vasishth
University of Potsdam Germany
vasishth@acm.org

Version dated February 9, 2007

Contents

1	Data preparation	1
2	Data analysis	3
2.1	At the embedded verb (verb1)	3
2.2	Second verb in embedded conditions (v2)	27
3	Conditional Regression Information	28
	• Regression out count: Number of times the interest area (IA) was exited to a lower IA position before a higher IA was fixated in the trial.	
	• Regression out full count: Number of times the interest area was exited to a lower IA position during the trial.	
	• First run duration: This contains the total duration of fixations until the gaze is first directed away from the IA, regardless of whether any material downstream in the sentence has been viewed.	
	• First fixation progressive: 1 if no material downstream in the sentence was viewed before the first fixation in this IA, 0 otherwise.	

1 Data preparation

```
> data <- read.table("grodner_standardNoAss.txt", header = TRUE)
> dataSPR <- read.table("../SPR/data3.txt")
> colnames(dataSPR) <- c("subj", "expt", "item", "condition", "roi",
+   "word", "RT", "embedding", "intervention")
> dataSPR$roi <- dataSPR$roi + 1
> lex <- read.table("~/Data/Corpora/anc-lexicon.txt", quote = "",
+   skip = 1)
```

```

> colnames(lex) <- c("word", "freq")
> lex <- subset(lex, is.element(word, dataSPR$word))
> dataSPR$freq <- as.integer(lapply(dataSPR$word, function(x) (lex$freq[lex$word ==
+   as.character(x)][1])))
> dataSPR$len <- nchar(as.character(dataSPR$word))
> freqdata <- data.frame(item = dataSPR$item, condition = dataSPR$condition,
+   roi = dataSPR$roi, word = dataSPR$word, freq = dataSPR$freq,
+   len = dataSPR$len)
> map <- function(vec, from, to) {
+   newVec <- vec
+   for (i in 1:length(from)) {
+     newVec[vec == from[i]] <- to[i]
+   }
+   return(newVec)
+ }
> attach(freqdata)
> rawdataIndex <- paste(item, condition, roi, "-")
> rawdataWords <- word
> detach(freqdata)
> attach(data)
> emoutputIndex <- paste(item, condition, roi, "-")
> detach(data)
> data$word <- map(emoutputIndex, rawdataIndex, as.character(rawdataWords))
> lex <- read.table("~/Data/Corpora/anc-lexicon.txt", quote = "",
+   skip = 1)
> colnames(lex) <- c("word", "freq")
> lex <- subset(lex, is.element(word, data$word))
> data$freq <- as.integer(lapply(data$word, function(x) (lex$freq[lex$word ==
+   as.character(x)][1])))
> data$len <- nchar(as.character(data$word))
> data <- subset(data, (item != 25 & item != 2))
> data$subject <- factor(data$subject)
> data$trial <- ordered(data$trial)
> data$item <- factor(data$item)
> data$freq <- log(data$freq)

```

Spillover information (only for SFD, FFD, FPRT):

```

> maxspill <- 1
> temp <- cbind(data$roi, data$SFD)
> nrows <- dim(temp)[1]
> spillover <- matrix(0, nrow = nrows)
> for (j in 1:maxspill) {
+   for (i in 1:nrows) {
+     if (temp[i, 1] <= j) {
+       spillover[i, 1] <- NA
+     }
+     if (temp[i, 1] > j) {
+       spillover[i, 1] <- temp[(i - j), 2]
+     }
+   }
+   data <- data.frame(data, n.SFD = spillover)
+ }
> temp <- cbind(data$roi, data$FFD)
> nrows <- dim(temp)[1]

```

```

> spillover <- matrix(0, nrow = nrows)
> for (j in 1:maxspill) {
+   for (i in 1:nrows) {
+     if (temp[i, 1] <= j) {
+       spillover[i, 1] <- NA
+     }
+     if (temp[i, 1] > j) {
+       spillover[i, 1] <- temp[(i - j), 2]
+     }
+   }
+   data <- data.frame(data, n.FFD = spillover)
+ }
> temp <- cbind(data$roi, data$FPRT)
> nrows <- dim(temp)[1]
> spillover <- matrix(0, nrow = nrows)
> for (j in 1:maxspill) {
+   for (i in 1:nrows) {
+     if (temp[i, 1] <= j) {
+       spillover[i, 1] <- NA
+     }
+     if (temp[i, 1] > j) {
+       spillover[i, 1] <- temp[(i - j), 2]
+     }
+   }
+   data <- data.frame(data, n.FPRT = spillover)
+ }

```

2 Data analysis

2.1 At the embedded verb (verb1)

Subset the embedded verb data:

```

> data$criticalv1 <- ifelse((data$condition == "a" & data$roi ==
+   3) | (data$condition == "b" & data$roi == 6) | (data$condition ==
+   "c" & data$roi == 8) | (data$condition == "d" & data$roi ==
+   6) | (data$condition == "e" & data$roi == 9) | (data$condition ==
+   "f" & data$roi == 11), "yes", "no")
> d <- subset(data, criticalv1 == "yes")
> d$criticalv1 <- factor(d$criticalv1)
> d$cfreq <- d$freq - mean(d$freq, na.rm = TRUE)
> d$clen <- d$len - mean(d$len, na.rm = TRUE)

```

Five nested orthogonal contrasts:

Main prediction is that PP and RC modification will result in increased fixation durations at the verb.

```

> d$c1 <- ifelse(d$condition %in% c("a", "b", "c"), 1, -1)
> d$c2 <- ifelse(d$condition == "a", -2, ifelse(d$condition %in%
+   c("b", "c"), 1, 0))

```

	a	b	c	d	e	f
c1	-1	-1	-1	1	1	1
c2	-2	1	1	0	0	0
c3	0	-1	1	0	0	0
c4	0	0	-2	1	1	
c5	0	0	0	-1	1	

Table 1: Orthogonal contrasts.

```
> d$c3 <- ifelse(d$condition == "b", -1, ifelse(d$condition ==
+ "c", 1, 0))
> d$c4 <- ifelse(d$condition == "d", -2, ifelse(d$condition %in%
+ c("e", "f"), 1, 0))
> d$c5 <- ifelse(d$condition == "e", -1, ifelse(d$condition ==
+ "f", 1, 0))
> library(reshape)
> d.rs <- melt(d, id = c("subject", "condition", "roi", "qacc",
+ "c1", "c2", "c3", "c4", "c5", "item", "cfreq", "c1en", "word",
+ "n.SFD", "n.FFD", "n.FPRT"), measure = c("FFD", "FFP", "SFD",
+ "FPRT", "RBRT", "RPD", "RRT", "RRTR", "RRTP", "ROC", "ROFC",
+ "LPRT", "TFT"), variable_name = "times", preserve.na = FALSE)
> d.rs <- subset(d.rs, value > 0)
```

Means and CIs:

```
> standard <- c("SFD", "FFD", "FPRT", "RBRT", "RPD", "RRT", "RRTR",
+ "RRTP", "TFT", "LPRT")
> accumulator <- c()
> for (m in standard) {
+   d.cast <- cast(d.rs, condition ~ ., function(x) c(M = round(mean(x)),
+     ci = round(ci(x))), subset = times == m)
+   d.cast <- data.frame(d.cast, measure = m)
+   accumulator <- rbind(accumulator, d.cast)
+ }
> d.cast <- accumulator
> d.cast$cond <- as.numeric(d.cast$condition)
> names(d.cast$cond) <- "Condition"
> levels(d.cast$cond) <- letters[1:6]
> d.cast$measure <- factor(d.cast$measure)
> library(Hmisc)
```

Hmisc library by Frank E Harrell Jr

Type library(help='Hmisc'), ?Overview, or ?Hmisc.Overview') to see overall documentation.

NOTE:Hmisc no longer redefines [.factor to drop unused levels when subsetting. To get the old behavior of Hmisc type dropUnusedLevels().

```
> sfdplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+ 1), data = subset(d.cast, measure == "SFD"), nx = TRUE, asp = 1,
+ type = "b", ylim = range(subset(d.cast, measure == "SFD")$ci.lower,
+ subset(d.cast, measure == "SFD")$ci.upper), pch = c(1),
+ lty = 1:5, col = 1, cex = 0.3, xlab = "Condition", ylab = "Reading time [ms]",
+ scales = list(x = list(at = seq(1, 6, by = 1), labels = levels(d.cast$condition))),
```

```

+   main = "Single-fixation duration")
> ffdplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "FFD"), nx = TRUE, asp = 1,
+   type = "b", ylim = range(subset(d.cast, measure == "FFD")$ci.lower,
+   subset(d.cast, measure == "FFD")$ci.upper), pch = c(1),
+   lty = 1:5, col = 1, cex = 0.3, xlab = "Condition", ylab = "Reading time [ms]",
+   scales = list(x = list(at = seq(1, 6, by = 1), labels = levels(d.cast$condition))),
+   main = "First-fixation duration")
> fprtplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "FPRT"), nx = TRUE,
+   asp = 1, type = "b", ylim = range(subset(d.cast, measure ==
+   "FPRT")$ci.lower, subset(d.cast, measure == "FPRT")$ci.upper),
+   pch = c(1), lty = 1:5, col = 1, cex = 0.3, keys = "lines",
+   xlab = "Condition", ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+   6, by = 1), labels = levels(d.cast$condition))), main = "First-pass reading time")
> rbrtplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "RBRT"), nx = TRUE,
+   asp = 1, type = "b", ylim = range(subset(d.cast, measure ==
+   "RBRT")$ci.lower, subset(d.cast, measure == "RBRT")$ci.upper),
+   pch = c(1), lty = 1:5, col = 1, cex = 0.3, keys = "lines",
+   xlab = "Condition", ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+   6, by = 1), labels = levels(d.cast$condition))), main = "Right-bounded reading time")
> rpdplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "RPD"), nx = TRUE, asp = 1,
+   type = "b", ylim = range(subset(d.cast, measure == "RPD")$ci.lower,
+   1000, subset(d.cast, measure == "RPD")$ci.upper), pch = c(1),
+   lty = 1:5, col = 1, cex = 0.3, keys = "lines", xlab = "Condition",
+   ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+   6, by = 1), labels = levels(d.cast$condition))), main = "Regression path duration")
> rrtplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "RRT"), nx = TRUE, asp = 1,
+   type = "b", ylim = range(subset(d.cast, measure == "RRT")$ci.lower,
+   subset(d.cast, measure == "RRT")$ci.upper), pch = c(1),
+   lty = 1:5, col = 1, cex = 0.3, keys = "lines", xlab = "Condition",
+   ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+   6, by = 1), labels = levels(d.cast$condition))), main = "Rereading time")
> rrtrplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "RRTR"), nx = TRUE,
+   asp = 1, type = "b", ylim = range(subset(d.cast, measure ==
+   "RRTR")$ci.lower, subset(d.cast, measure == "RRTR")$ci.upper),
+   pch = c(1), lty = 1:5, col = 1, cex = 0.3, keys = "lines",
+   xlab = "Condition", ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+   6, by = 1), labels = levels(d.cast$condition))), main = "Rereading time, regressive")
> rrtpplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "RRTP"), nx = TRUE,
+   asp = 1, type = "b", ylim = range(subset(d.cast, measure ==
+   "RRTP")$ci.lower, subset(d.cast, measure == "RRTP")$ci.upper),
+   pch = c(1), lty = 1:5, col = 1, cex = 0.3, keys = "lines",
+   xlab = "Condition", ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+   6, by = 1), labels = levels(d.cast$condition))), main = "Rereading time, progressive")
> lprrtplot <- xYplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+   1), data = subset(d.cast, measure == "LPRT"), nx = TRUE,
+   asp = 1, type = "b", ylim = range(subset(d.cast, measure ==
+   "LPRT")$ci.lower, subset(d.cast, measure == "LPRT")$ci.upper),
+   pch = c(1), lty = 1:5, col = 1, cex = 0.3, keys = "lines",
+   xlab = "Condition", ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,

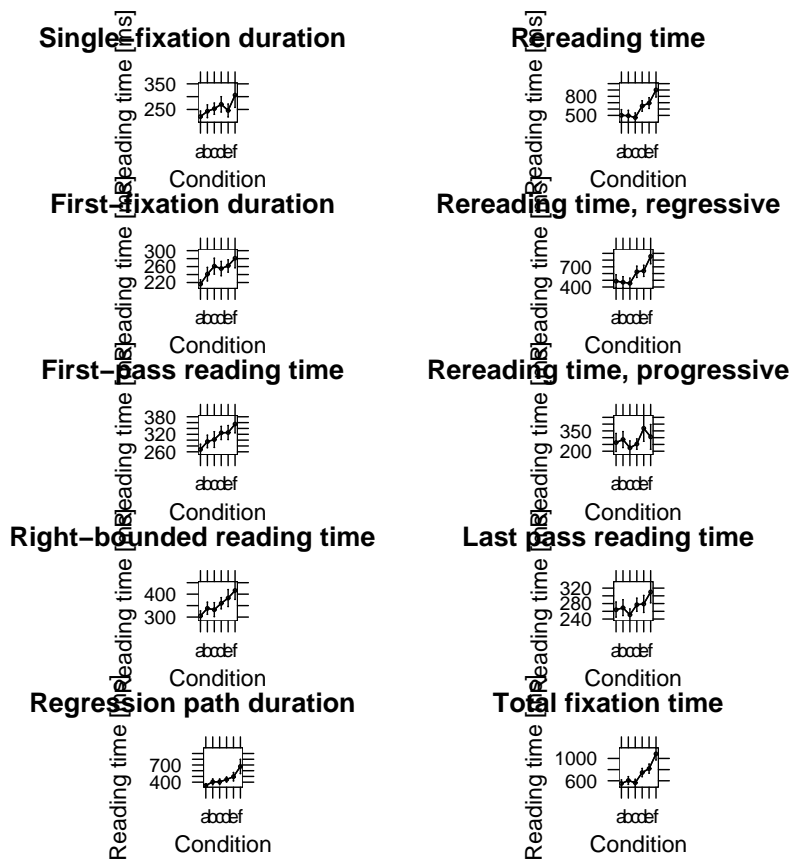
```

```

+       6, by = 1), labels = levels(d.cast$condition))), main = "Last pass reading time")
> tftplot <- xyplot(Cbind(M, ci.lower, ci.upper) ~ cond, layout = c(1,
+       1), data = subset(d.cast, measure == "TFT"), nx = TRUE, asp = 1,
+       type = "b", ylim = range(subset(d.cast, measure == "TFT")$ci.lower,
+       subset(d.cast, measure == "TFT")$ci.upper), pch = c(1),
+       lty = 1:5, col = 1, cex = 0.3, keys = "lines", xlab = "Condition",
+       ylab = "Reading time [ms]", scales = list(x = list(at = seq(1,
+       6, by = 1), labels = levels(d.cast$condition))), main = "Total fixation time")

> print(sfdplot, more = TRUE, position = c(0, 0, 1, 1), split = c(1,
+       1, 2, 5))
> print(ffdplot, more = TRUE, position = c(0, 0, 1, 1), split = c(1,
+       2, 2, 5))
> print(fprtplot, more = TRUE, position = c(0, 0, 1, 1), split = c(1,
+       3, 2, 5))
> print(rbrtplot, more = TRUE, position = c(0, 0, 1, 1), split = c(1,
+       4, 2, 5))
> print(rpdplot, more = TRUE, position = c(0, 0, 1, 1), split = c(1,
+       5, 2, 5))
> print(rrtplot, more = TRUE, position = c(0, 0, 1, 1), split = c(2,
+       1, 2, 5))
> print(rrtrplot, more = TRUE, position = c(0, 0, 1, 1), split = c(2,
+       2, 2, 5))
> print(rrtpplot, more = TRUE, position = c(0, 0, 1, 1), split = c(2,
+       3, 2, 5))
> print(lprtplot, more = TRUE, position = c(0, 0, 1, 1), split = c(2,
+       4, 2, 5))
> print(tftplot, more = TRUE, position = c(0, 0, 1, 1), split = c(2,
+       5, 2, 5))

```

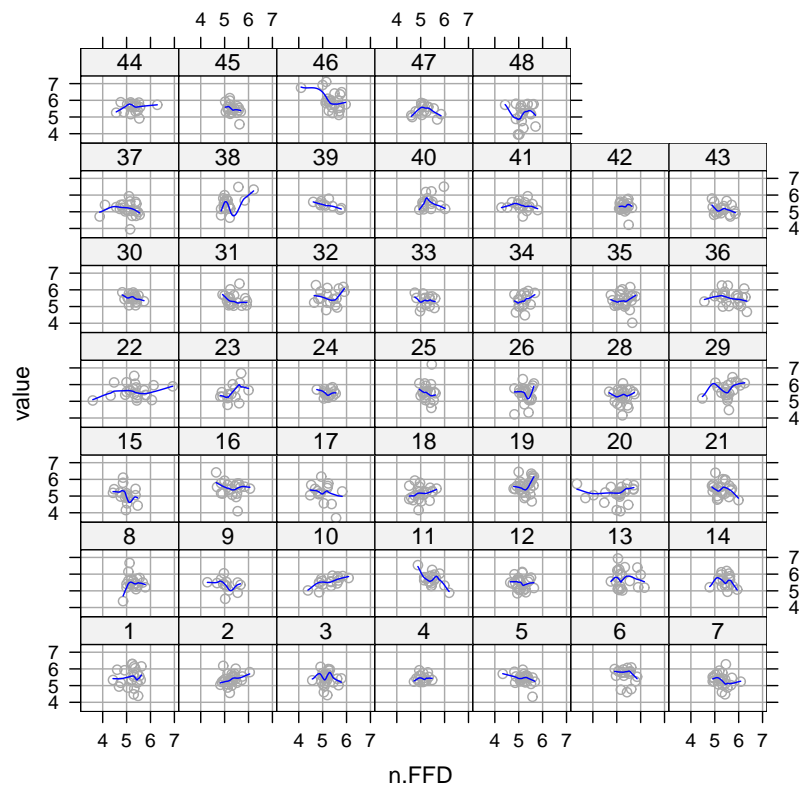


Spillover in SFD (too little data).

Custom function:

```
> xyloless = function(fmla = Rating ~ Frequency | Subject, data = weight,
+   span = 2/3, symbolcolor = "darkgrey", linecolor = "blue",
+   xlabel = "", ylabel = "") {
+   y = strsplit(as.character(fmla), "[|]")
+   if (xlabel == "")
+     xlabel = y[[3]][1]
+   if (ylabel == "")
+     ylabel = y[[2]]
+   require("lattice", quietly = TRUE, character = TRUE)
+   xyplot(fmla, data = data, xlab = xlabel, ylab = ylabel, panel = function(x,
+     y) {
+       panel.grid(h = -1, v = -1)
+       panel.xyplot(x, y, col.symbol = symbolcolor)
+       panel.loess(x, y, span = span, col.line = linecolor)
+     })
+ }
```

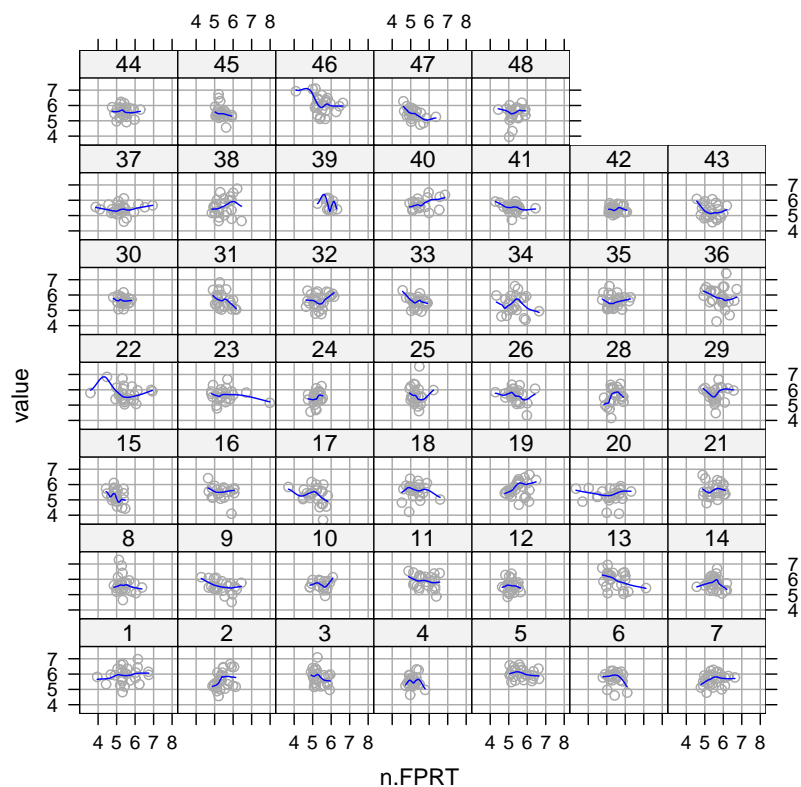
```
> tempdata <- subset(d.rs, times == "FFD" & n.FFD > 0)
> tempdata$value <- log(tempdata$value)
> tempdata$n.FFD <- log(tempdata$n.FFD)
> source("~/Data/BaayenData/functions/xyloless.r")
> print(xyloless(value ~ n.FFD | subject, tempdata))
```



In FFD, as reading time increases, the amount of spillover decreases:

```
> library(lme4)
> plot(confint(lmList(value ~ n.FFD | subject, tempdata), pooled = TRUE),
+       order = 1)

> tempdata <- subset(d.rs, times == "FPRT" & n.FPRT > 0)
> tempdata$value <- log(tempdata$value)
> tempdata$n.FPRT <- log(tempdata$n.FPRT)
> print(xylowess(value ~ n.FPRT | subject, tempdata))
```



In FPRT too, as reading time increases, the amount of spillover decreases:

```
> plot(confint(lmList(value ~ n.FPRT | subject, tempdata), pooled = TRUE),
+       order = 1)
```

```
> library(lme4)
> sfddata <- subset(d.rs, times == "SFD")
> print(summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 +
+   c5 + cfreq + clen + +(1 | subject) + (1 | item), data = sfddata,
+   control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+     msVerbose = F, EMverbose = F))))
```

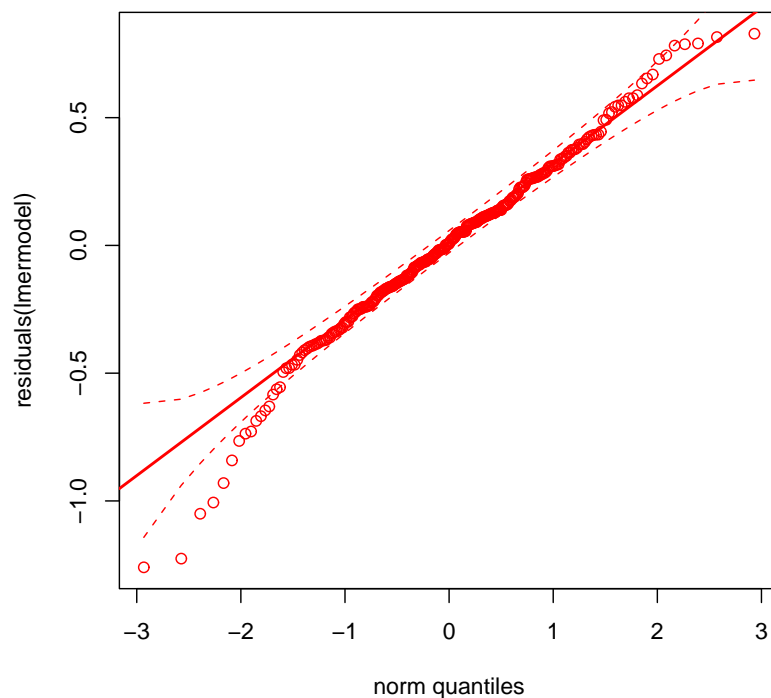
Linear mixed-effects model fit by REML Formula: $\log(\text{value}) \sim c1 + c2 + c3 + c4 + c5 + \text{cfreq} + \text{clen} + (1 | \text{subject}) + (1 | \text{item})$ Data: sfddata AIC BIC logLik MLdeviance REMLdeviance
309.1 346.0 -144.6 244.2 289.1 Random effects: Groups Name Variance Std.Dev. subject (Intercept) 0.0146643 0.121096 item (Intercept) 0.0006755 0.025990 Residual 0.1255644 0.354351
number of obs: 297, groups: subject, 44; item, 28

Fixed effects: Estimate Std. Error t value (Intercept) 5.490268 0.031138 176.32 c1 -0.076587 0.023039 -3.32 c2 0.043184 0.018681 2.31 c3 0.034359 0.032495 1.06 c4 0.013195 0.023709 0.56 c5 0.085111 0.048285 1.76 cfreq -0.015670 0.019508 -0.80 clen 0.008927 0.014396 0.62

Correlation of Fixed Effects: (Intr) c1 c2 c3 c4 c5 cfreq c1 -0.243 c2 0.010 0.021 c3 -0.050 -0.081 -0.076 c4 0.140 -0.189 -0.009 -0.001 c5 0.071 -0.123 -0.006 0.016 0.107 cfreq -0.165 0.096 0.008 0.009 -0.020 0.077 clen 0.130 0.082 0.023 -0.026 0.037 -0.005 0.203

Examine residuals:

```
> library(car)
> qq.plot(residuals(lmermodel))
```



Print p values:

```
> xtable(pvals.fnc(lmermodel)$anova)
```

Custom function for HPD intervals:

```
> printHPDs <- function(m) {
+   M <- rowMeans(HPDinterval(m)[2:6, ])
+   HPDs <- data.frame(M = M, ci.lower = HPDinterval(m)[2:6,
+     1], ci.upper = HPDinterval(m)[2:6, 2])
+   print(xtable(HPDs, digits = 4))
+ }
> plotHPDs <- function(m, measure) {
+   M <- rowMeans(HPDinterval(m)[2:6, ])
+   HPDs <- data.frame(contrasts = paste("c", 1:5, sep = ""),
+     M = M, ci.lower = HPDinterval(m)[2:6, 1], ci.upper = HPDinterval(m)[2:6,
+     2])
+   print(HPDs)
+   HPDs$contrasts <- as.numeric(HPDs$contrasts)
+   names(HPDs$contrasts) <- "Contrasts"
+   levels(HPDs$contrasts) <- paste("c", 1:5, sep = "")
+   print(xYplot(Cbind(M, ci.lower, ci.upper) ~ contrasts, layout = c(1,
+     1), data = HPDs, nx = TRUE, asp = 1, type = "p", pch = c(1),
```

```

+       col = 1, cex = 0.3, xlab = "Contrasts", ylab = "Estimates",
+       scales = list(x = list(at = seq(1, 5, by = 1), labels = levels(HPDs$contrasts))),
+       main = paste("Highest posterior density intervals,",
+         measure, sep = " "))
+ }

> library(coda)
> m1 <- mcmcscamp(lmermodel, 50000)
> printHPDs(m1)

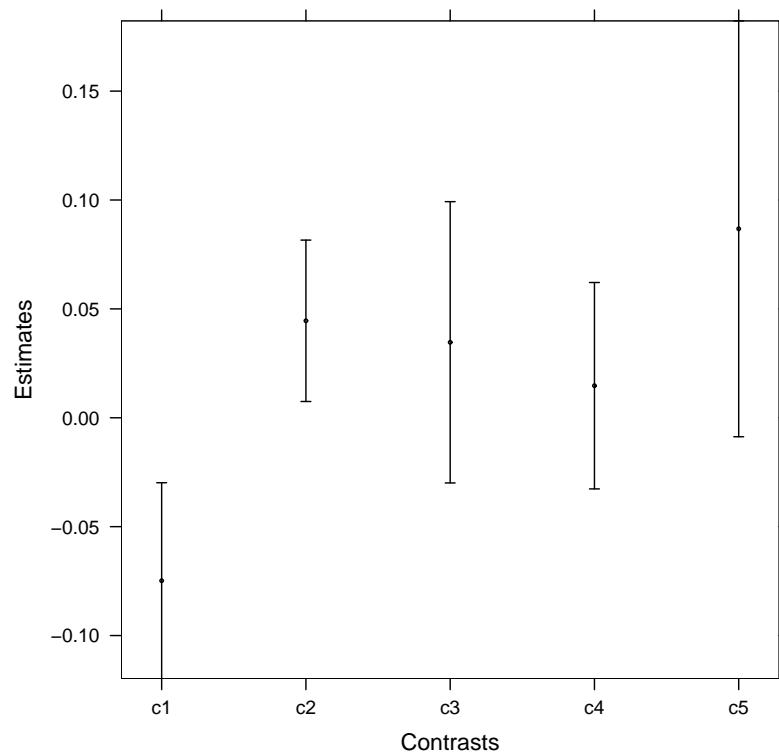
```

	M	ci.lower	ci.upper
c1	-0.0748	-0.1198	-0.0298
c2	0.0445	0.0075	0.0816
c3	0.0347	-0.0299	0.0992
c4	0.0147	-0.0327	0.0621
c5	0.0868	-0.0087	0.1823

```
> plotHPDs(m1, "Single fixations")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.07483058	-0.119832808	-0.02982835
c2	c2	0.04453334	0.007476004	0.08159068
c3	c3	0.03465730	-0.029927716	0.09924231
c4	c4	0.01472467	-0.032668744	0.06211809
c5	c5	0.08677040	-0.008711119	0.18225193

Highest posterior density intervals, Single fixations



First fixation durations:

```
> ffddata <- subset(d.rs, times == "FFD" & n.FFD > 0)
> print(summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 +
+   c5 + cfreq + clen + log(n.FFD) + (1 | subject) + (1 | item),
+   data = ffddata, control = list(gradient = FALSE, niterEM = 0,
+   maxIter = 200, msVerbose = F, EMverbose = F))))
```

Linear mixed-effects model fit by REML

Formula: $\log(\text{value}) \sim c1 + c2 + c3 + c4 + c5 + \text{cfreq} + \text{clen} + \log(\text{n.FFD}) + (1 | \text{subject}) + (1 |$

Data: ffddata

AIC	BIC	logLik	MLdeviance	REMLdeviance
1106	1159	-542	1028	1084

Random effects:

Groups	Name	Variance	Std.Dev.
subject	(Intercept)	0.0242494	0.155722
item	(Intercept)	0.0022880	0.047833
Residual		0.1626065	0.403245

number of obs: 942, groups: subject, 47; item, 28

Fixed effects:

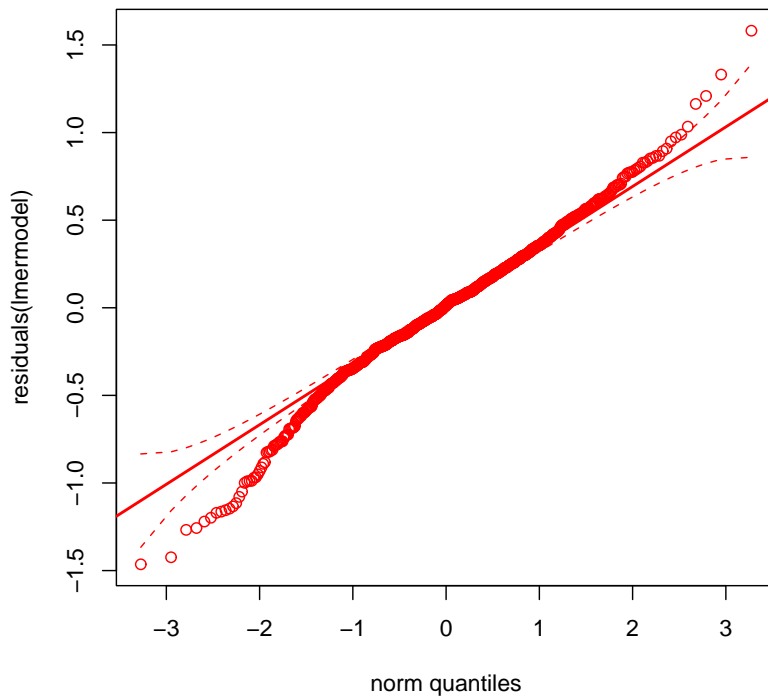
	Estimate	Std. Error	t value
(Intercept)	5.540e+00	2.120e-01	26.130
c1	-5.756e-02	1.335e-02	-4.312
c2	3.611e-02	1.280e-02	2.822
c3	4.400e-02	2.299e-02	1.914
c4	2.572e-02	1.301e-02	1.976
c5	2.087e-02	2.474e-02	0.844
cfreq	3.929e-05	1.399e-02	0.003
clen	2.017e-02	1.058e-02	1.905
log(n.FFD)	-1.824e-02	4.003e-02	-0.456

Correlation of Fixed Effects:

	(Intr)	c1	c2	c3	c4	c5	cfreq	clen
c1	-0.042							
c2	0.040	0.035						
c3	0.051	0.000	0.003					
c4	0.012	-0.105	0.003	0.000				
c5	0.053	-0.013	-0.001	0.005	0.013			
cfreq	-0.012	0.028	0.018	0.005	-0.016	-0.010		
clen	0.042	0.001	0.036	-0.003	-0.043	0.006	0.347	
log(n.FFD)	-0.991	0.039	-0.038	-0.051	-0.005	-0.053	0.012	-0.045

Examine residuals:

```
> library(car)
> qq.plot(residuals(lmermodel))
```



Print p values:

```
> xtable(pvals.fnc(lmermodel)$anova)
```

```
> m <- mcmcscamp(lmermodel, 50000)
```

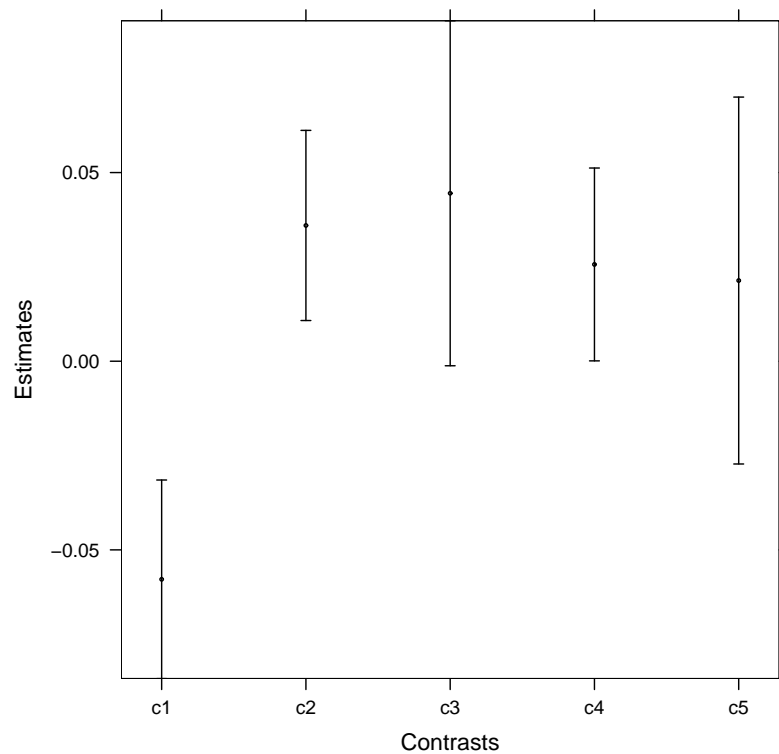
```
> printHPDs(m)
```

	M	ci.lower	ci.upper
c1	-0.0578	-0.0841	-0.0315
c2	0.0360	0.0108	0.0612
c3	0.0445	-0.0012	0.0902
c4	0.0256	0.0001	0.0512
c5	0.0214	-0.0272	0.0700

```
> plotHPDs(m, "First fixation durations")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.05780156	-8.412037e-02	-0.03148275
c2	c2	0.03598174	1.078588e-02	0.06117759
c3	c3	0.04451395	-1.202894e-03	0.09023079
c4	c4	0.02564836	9.161385e-05	0.05120511
c5	c5	0.02138435	-2.723480e-02	0.07000349

Highest posterior density intervals, First fixation durations



First pass reading times:

```
> fprtdata <- subset(d.rs, times == "FPRT" & n.FPRT > 0)
> print(summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 +
+   c5 + cfreq + clen + log(n.FPRT) + (1 | subject) + (1 | item),
+   data = fprtdata, control = list(gradient = FALSE, niterEM = 0,
+   maxIter = 200, msVerbose = F, EMverbose = F))))
```

Linear mixed-effects model fit by REML

Formula: $\log(\text{value}) \sim c1 + c2 + c3 + c4 + c5 + cfreq + clen + \log(n.FPRT) + (1 | \text{subject}) + (1$

Data: fprtdata

	AIC	BIC	logLik	MLdeviance	REMLdeviance
	1505	1561	-741.7	1427	1483

Random effects:

Groups	Name	Variance	Std.Dev.
subject	(Intercept)	0.0288107	0.16974
item	(Intercept)	0.0034187	0.05847
Residual		0.1923446	0.43857

number of obs: 1135, groups: subject, 47; item, 28

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	5.74914	0.16774	34.27
c1	-0.07639	0.01311	-5.82
c2	0.03124	0.01296	2.41
c3	0.01408	0.02280	0.62
c4	0.01701	0.01289	1.32
c5	0.03674	0.02343	1.57

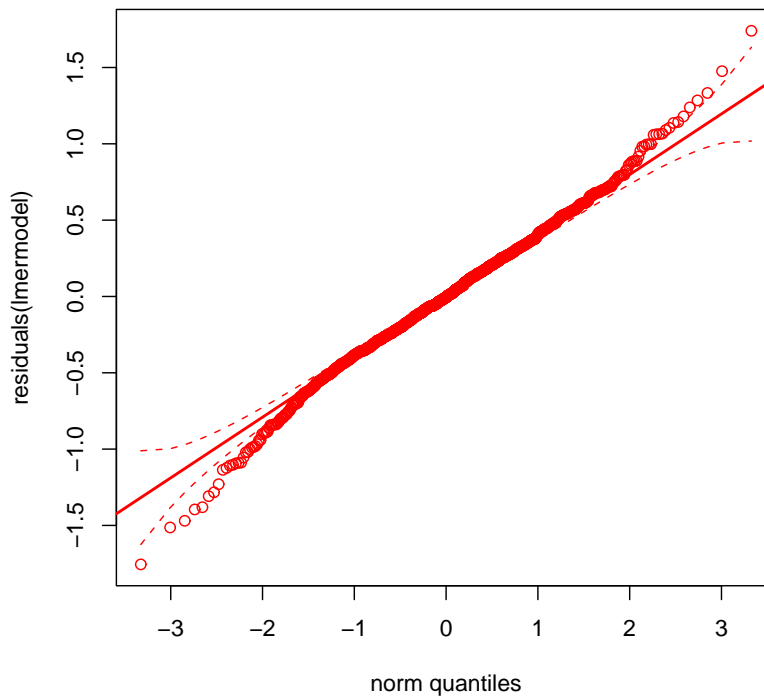
```
cfreq      -0.01565    0.01481   -1.06
clen       0.02979    0.01122    2.65
log(n.FPRT) -0.02365    0.03060   -0.77
```

Correlation of Fixed Effects:

	(Intr)	c1	c2	c3	c4	c5	cfreq	clen
c1		0.056						
c2	-0.126		0.017					
c3	-0.003	0.001	0.002					
c4	-0.013	-0.050	0.005	0.000				
c5	0.071	0.014	-0.010	0.001	-0.011			
cfreq	-0.015	0.000	0.012	0.003	0.003	0.002		
clen	0.020	-0.002	0.014	0.003	-0.013	0.020	0.348	
log(n.FPRT)	-0.984	-0.058	0.131	0.003	0.017	-0.073	0.017	-0.020

Examine residuals:

```
> qq.plot(residuals(lmermodel))
```



Print p values:

```
> xtable(pvals.fnc(lmermodel)$anova)
```

```
> m <- mcmcscamp(lmermodel, 50000)
```

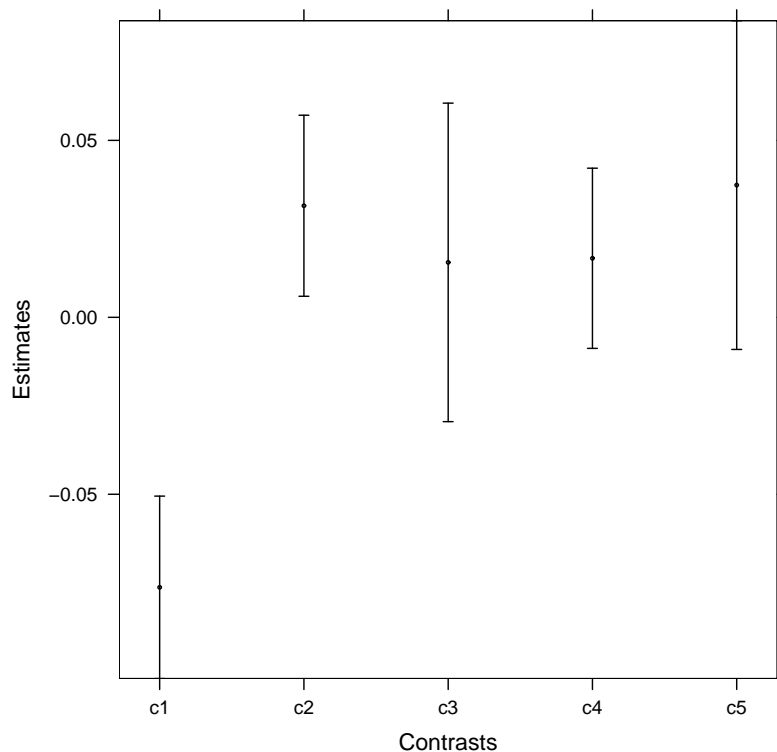
```
> printHPDs(m)
```

	M	ci.lower	ci.upper
c1	-0.0763	-0.1021	-0.0505
c2	0.0315	0.0059	0.0571
c3	0.0155	-0.0295	0.0605
c4	0.0167	-0.0088	0.0421
c5	0.0374	-0.0091	0.0838

```
> plotHPDs(m, "First-pass reading times")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.07629948	-0.102077463	-0.05052150
c2	c2	0.03152588	0.005946124	0.05710564
c3	c3	0.01552263	-0.029478996	0.06052425
c4	c4	0.01668229	-0.008777127	0.04214171
c5	c5	0.03736475	-0.009078395	0.08380789

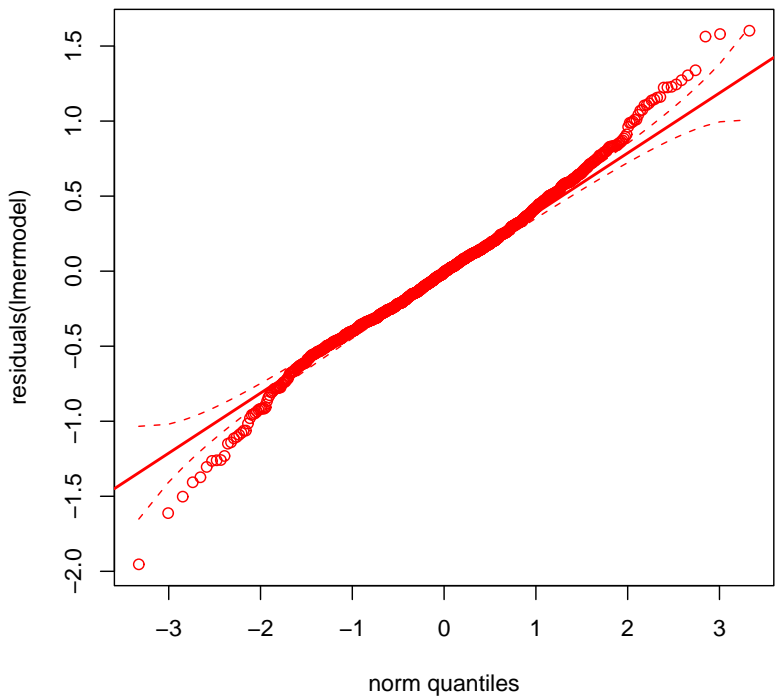
Highest posterior density intervals, First-pass reading times



Right-bounded RT:

```
> summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 + c5 +
+   cfreq + clen + (1 | subject) + (1 | item), data = d.rs, subset = times ==
+   "RBRT", control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+   msVerbose = F, EMverbose = F)))

> qq.plot(residuals(lmermodel))
```



```
> xtable(pvals.fnc(lmermodel)$anova)
```

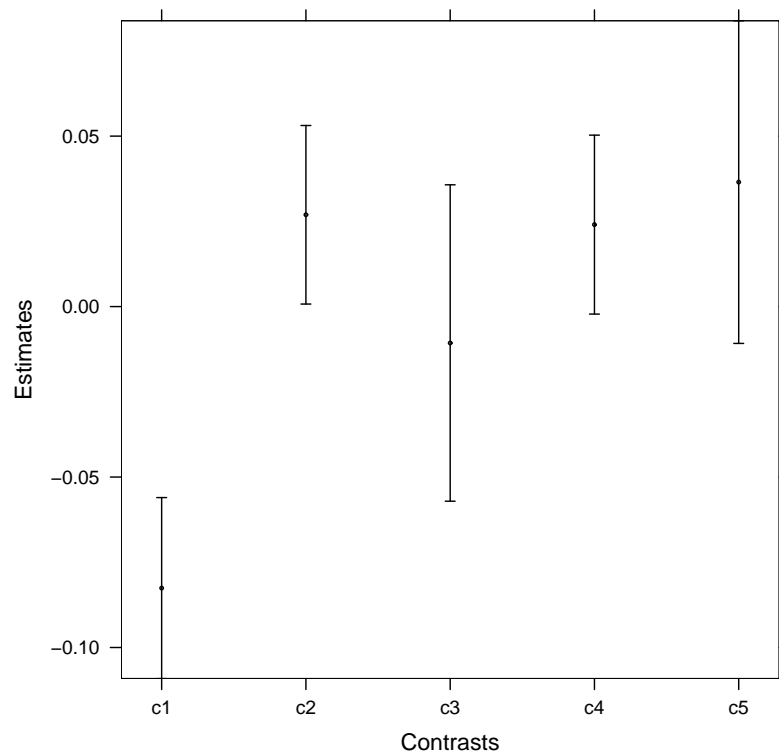
```
> m <- mcmcscamp(lmermodel, 50000)
```

```
> printHPDs(m)
```

	M	ci.lower	ci.upper
c1	-0.0826	-0.1091	-0.0560
c2	0.0269	0.0007	0.0531
c3	-0.0107	-0.0571	0.0357
c4	0.0240	-0.0022	0.0503
c5	0.0365	-0.0108	0.0838

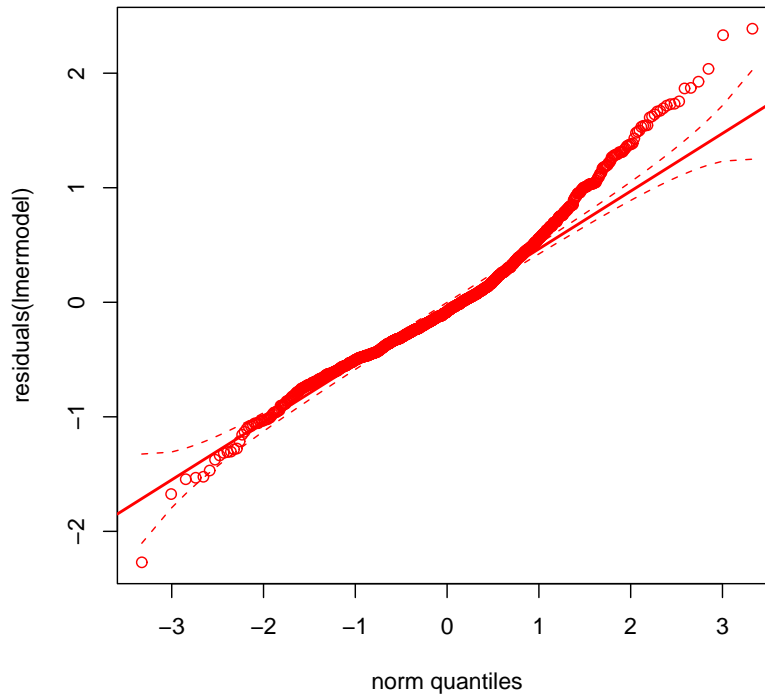
```
> plotHPDs(m, "Right-bounded reading time")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.08257370	-0.1091220001	-0.05602540
c2	c2	0.02692801	0.0007386516	0.05311737
c3	c3	-0.01068054	-0.0570819373	0.03572085
c4	c4	0.02404344	-0.0021996358	0.05028651
c5	c5	0.03651279	-0.0108079134	0.08383350

Highest posterior density intervals, Right-bounded reading time

```
> summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 + c5 +
+   cfreq + clen + (1 | subject) + (1 | item), data = d.rs, subset = times ==
+   "RPD", control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+   msVerbose = F, EMverbose = F)))

> qq.plot(residuals(lmermodel))
```



```
> xtable(pvals.fnc(lmermodel)$anova)
```

```
> m <- mcmcscamp(lmermodel, 50000)
```

```
> printHPDs(m)
```

```
% latex table generated in R 2.4.1 by xtable 1.4-3 package
```

```
% Fri Feb 9 16:08:41 2007
```

```
\begin{table}[ht]
```

```
\begin{center}
```

```
\begin{tabular}{rrrr}
```

```
\hline
```

```
& M & ci.lower & ci.upper \\\
```

```
\hline
```

```
c1 &  $-\$0.1191$  &  $-\$0.1550$  &  $-\$0.0832$  \\\
```

```
c2 & 0.0355 & 0.0011 & 0.0700 \\\
```

```
c3 &  $-\$0.0025$  &  $-\$0.0644$  & 0.0594 \\\
```

```
c4 & 0.0340 &  $-\$0.0007$  & 0.0687 \\\
```

```
c5 & 0.0832 & 0.0205 & 0.1459 \\\
```

```
\hline
```

```
\end{tabular}
```

```
\end{center}
```

```
\end{table}
```

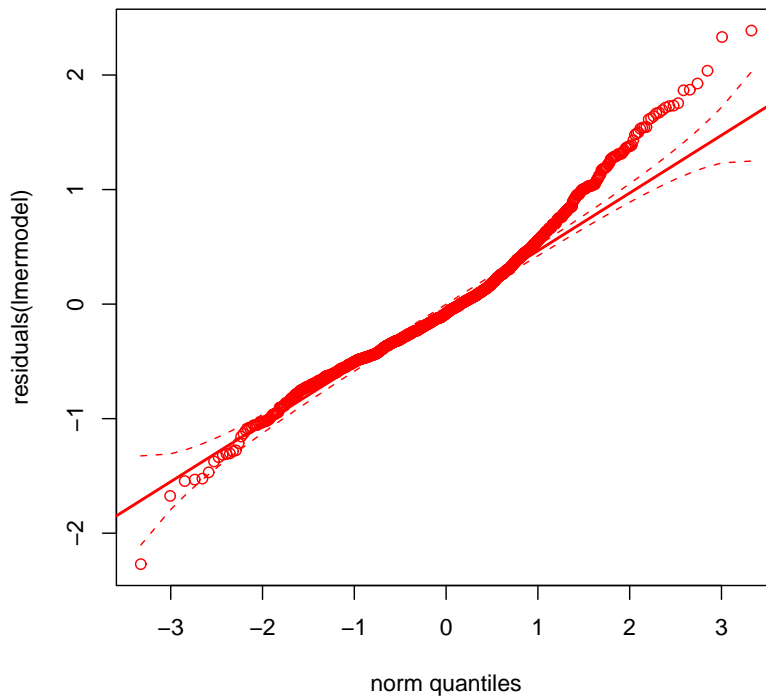
```
> plotHPDs(m, "Regression path duration")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.119083181	-0.1550140513	-0.08315231
c2	c2	0.035525509	0.0010903990	0.06996062
c3	c3	-0.002487348	-0.0643806244	0.05940593
c4	c4	0.034011585	-0.0007016739	0.06872484
c5	c5	0.083193683	0.0205072695	0.14588010

Regression-path duration:

```
> summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 + c5 +
+   cfreq + clen + (1 | subject) + (1 | item), data = d.rs, subset = times ==
+   "RPD", control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+   msVerbose = F, EMverbose = F)))
```

```
> qq.plot(residuals(lmermodel))
```



```
> xtable(pvals.fnc(lmermodel)$anova)
```

```
> m <- mcmc samp(lmermodel, 50000)
```

```
> printHPDs(m)
```

```
% latex table generated in R 2.4.1 by xtable 1.4-3 package
% Fri Feb 9 16:10:35 2007
```

```

\begin{table}[ht]
\begin{center}
\begin{tabular}{rrrr}
\hline
& M & ci.lower & ci.upper \\
\hline
c1 &  $-\$0.1186$  &  $-\$0.1535$  &  $-\$0.0837$  \\
c2 &  $0.0351$  &  $0.0007$  &  $0.0695$  \\
c3 &  $-\$0.0043$  &  $-\$0.0648$  &  $0.0562$  \\
c4 &  $0.0351$  &  $0.0005$  &  $0.0696$  \\
c5 &  $0.0836$  &  $0.0206$  &  $0.1465$  \\
\hline
\end{tabular}
\end{center}
\end{table}

```

```
> plotHPDs(m, "Regression path duration")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.11856795	-0.1534853559	-0.08365055
c2	c2	0.03512206	0.0007102287	0.06953388
c3	c3	-0.00430810	-0.0647666849	0.05615049
c4	c4	0.03508149	0.0005476052	0.06961537
c5	c5	0.08356416	0.0206402925	0.14648802

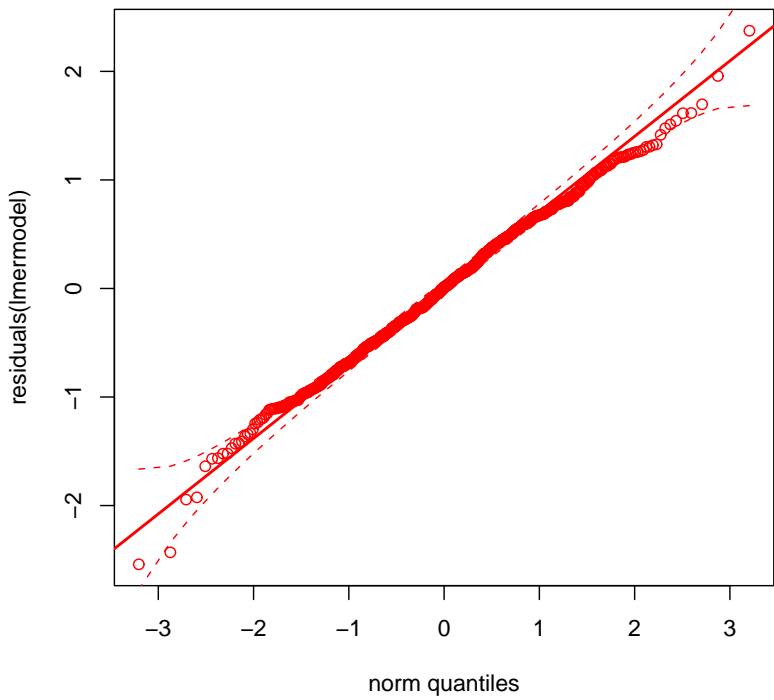
Re-reading time:

```

> summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 + c5 +
+   cfreq + clen + (1 | subject) + (1 | item), data = d.rs, subset = times ==
+   "RRT", control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+   msVerbose = F, EMverbose = F)))

```

```
> qq.plot(residuals(lmermodel))
```



```
> xtable(pvals.fnc(lmermodel)$anova)
```

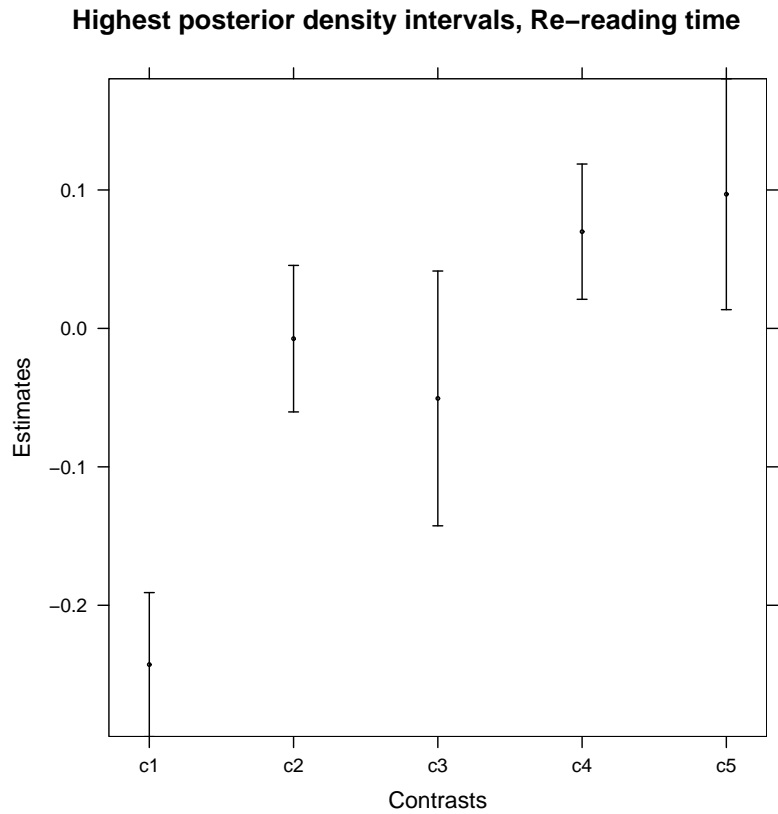
```
> m <- mcmcscamp(lmermodel, 50000)
```

```
> printHPDs(m)
```

	M	ci.lower	ci.upper
c1	-0.2428	-0.2949	-0.1908
c2	-0.0074	-0.0603	0.0455
c3	-0.0506	-0.1426	0.0414
c4	0.0699	0.0210	0.1187
c5	0.0969	0.0135	0.1803

```
> plotHPDs(m, "Re-reading time")
```

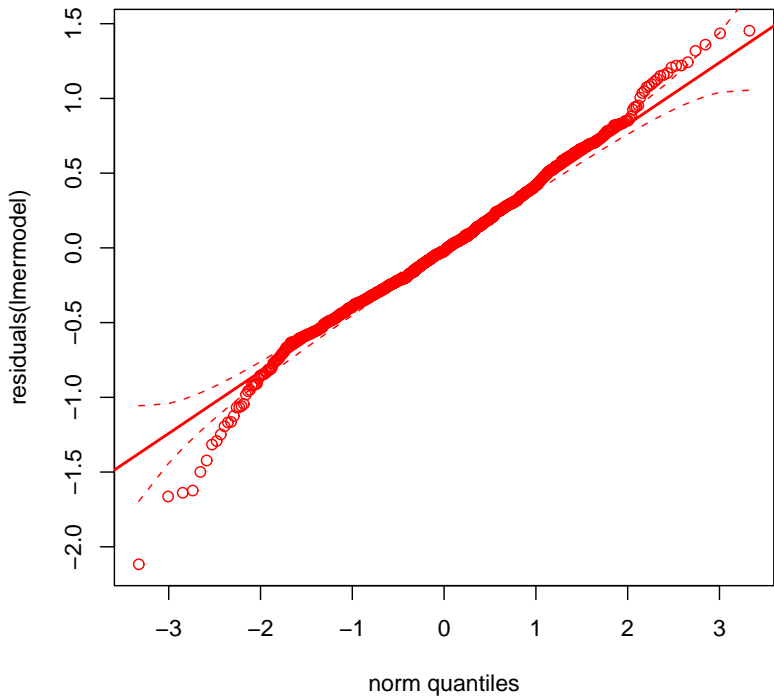
	contrasts	M	ci.lower	ci.upper
c1	c1	-0.242847681	-0.29486214	-0.19083322
c2	c2	-0.007416101	-0.06032913	0.04549693
c3	c3	-0.050578703	-0.14259741	0.04144000
c4	c4	0.069865798	0.02100688	0.11872472
c5	c5	0.096924795	0.01353649	0.18031310



Last-pass reading time:

```
> summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 + c5 +
+   cfreq + clen + (1 | subject) + (1 | item), data = d.rs, subset = times ==
+   "LPRT", control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+   msVerbose = F, EMverbose = F)))

> qq.plot(residuals(lmermodel))
```



```
> xtable(pvals.fnc(lmermodel)$anova)
```

```
> m <- mcmcscamp(lmermodel, 50000)
```

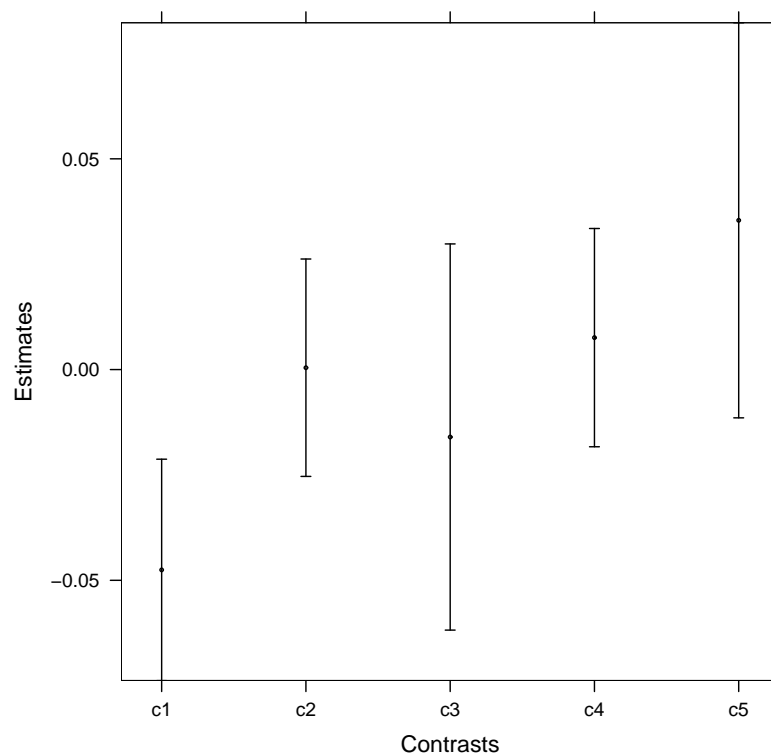
```
> printHPDs(m)
```

	M	ci.lower	ci.upper
c1	-0.0475	-0.0738	-0.0213
c2	0.0004	-0.0254	0.0262
c3	-0.0160	-0.0618	0.0298
c4	0.0076	-0.0183	0.0335
c5	0.0354	-0.0115	0.0823

```
> plotHPDs(m, "Last-pass reading time")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.0475317582	-0.07379736	-0.02126615
c2	c2	0.0004374313	-0.02535167	0.02622654
c3	c3	-0.0160050323	-0.06181620	0.02980613
c4	c4	0.0075744636	-0.01831881	0.03346774
c5	c5	0.0354098179	-0.01145743	0.08227707

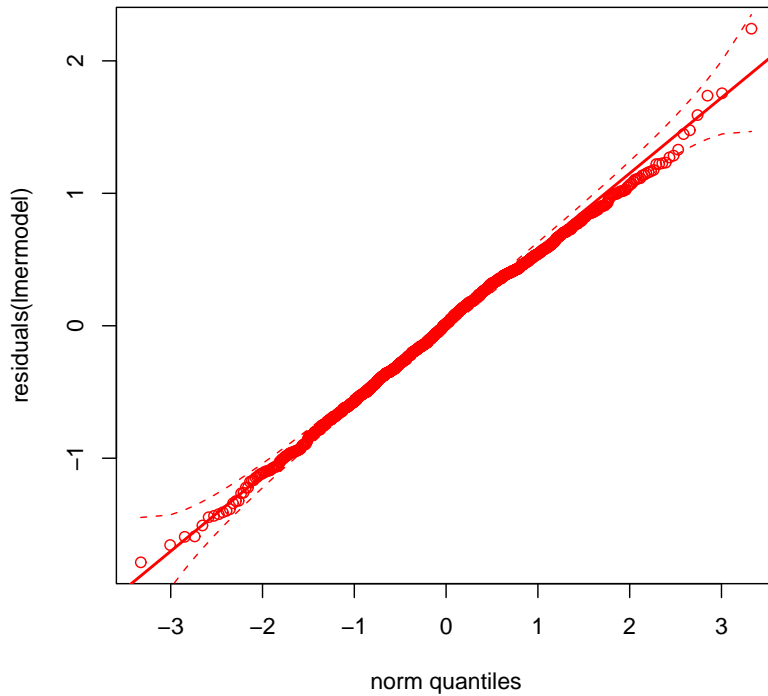
Highest posterior density intervals, Last-pass reading time



Total reading time:

```
> summary(lmermodel <- lmer(log(value) ~ c1 + c2 + c3 + c4 + c5 +
+   cfreq + clen + (1 | subject) + (1 | item), data = d.rs, subset = times ==
+   "TFT", control = list(gradient = FALSE, niterEM = 0, maxIter = 200,
+   msVerbose = F, EMverbose = F)))

> qq.plot(residuals(lmermodel))
```



```
> xtable(pvals.fnc(lmermodel)$anova)
```

```
> m <- mcmcscamp(lmermodel, 50000)
```

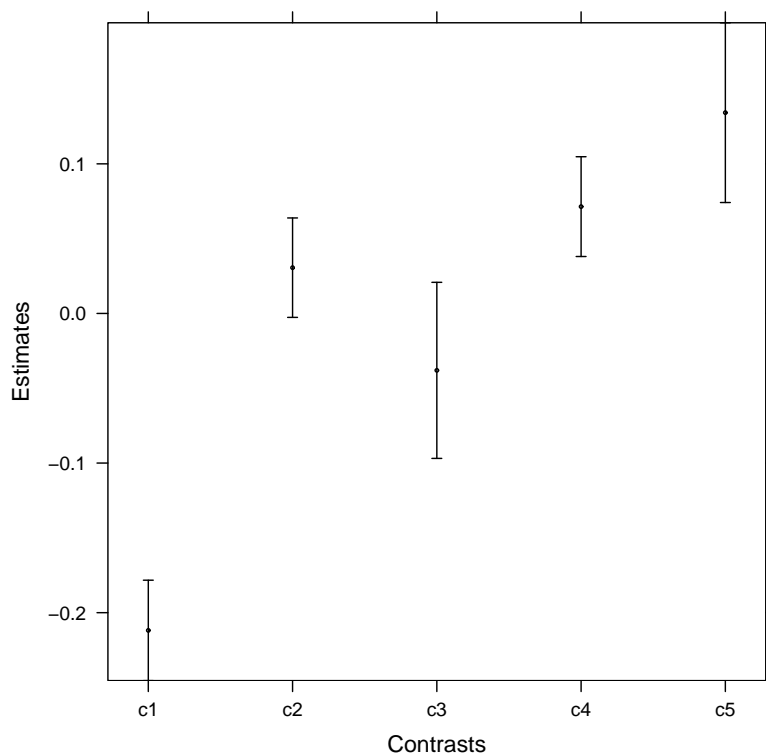
```
> printHPDs(m)
```

	M	ci.lower	ci.upper
c1	-0.2119	-0.2454	-0.1783
c2	0.0306	-0.0026	0.0638
c3	-0.0380	-0.0969	0.0208
c4	0.0714	0.0381	0.1048
c5	0.1342	0.0741	0.1943

```
> plotHPDs(m, "Total reading time")
```

	contrasts	M	ci.lower	ci.upper
c1	c1	-0.21185188	-0.245408328	-0.17829544
c2	c2	0.03059915	-0.002626912	0.06382520
c3	c3	-0.03803849	-0.096882111	0.02080512
c4	c4	0.07140919	0.038064283	0.10475409
c5	c5	0.13421830	0.074148018	0.19428858

Highest posterior density intervals, Total reading time



2.2 Second verb in embedded conditions (v2)

```
> data$criticalv2 <- ifelse((data$condition == "d" & data$roi ==
+   7) | (data$condition == "e" & data$roi == 10) | (data$condition ==
+   "f" & data$roi == 12), "yes", "no")
> dv2 <- subset(data, criticalv2 == "yes")
> dv2$condition <- factor(dv2$condition)
> dv2$cfreq <- dv2$freq - mean(dv2$freq, na.rm = TRUE)
> dv2$cflen <- dv2$len - mean(dv2$len, na.rm = TRUE)
```

The contrasts here are defined afresh. Note that the contrast names have nothing to do with the contrast names in the comparison above.

	d	e	f
c6	-2	1	1
c7	0	-1	1

Table 2: Orthogonal contrasts.

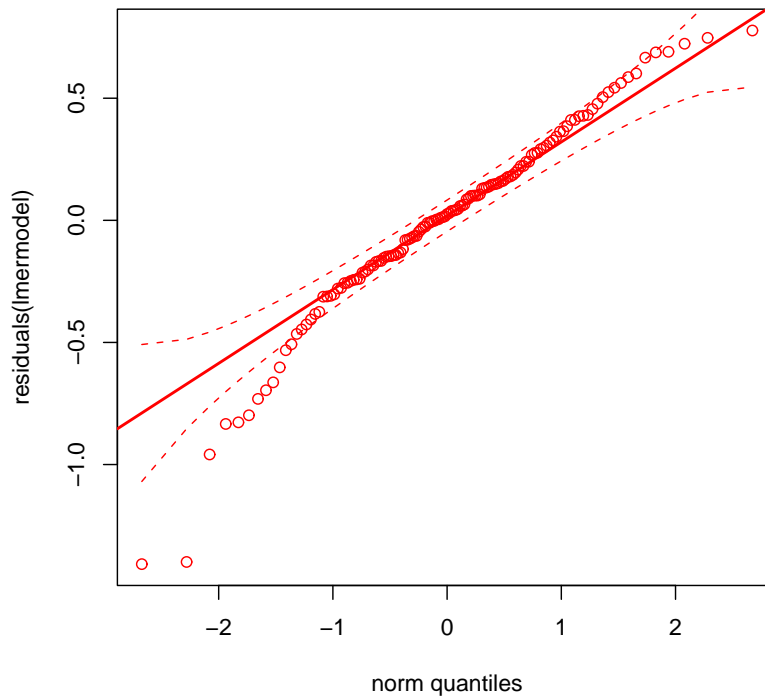
```
> dv2$c6 <- ifelse(dv2$condition == "d", -2, 1)
> dv2$c7 <- ifelse(dv2$condition == "e", -1, ifelse(dv2$condition ==
+   "f", 1, 0))
> dv2.rs <- melt(dv2, id = c("subject", "condition", "roi", "c6",
+   "c7", "item", "cfreq", "cflen", "word"), measure = standard,
+   variable_name = "times", preserve.na = FALSE)
> dv2.rs <- subset(dv2.rs, value > 0)
```

```

> summary(lmermodel <- lmer(log(value) ~ c6 + c7 + cfreq + clen +
+   (1 | subject), data = dv2.rs, subset = times == "SFD", control = list(gradient = FALSE,
+   niterEM = 0, maxIter = 200, msVerbose = F, EMverbose = F)))
> xtable(pvals.fnc(lmermodel)$anova)

> qq.plot(residuals(lmermodel))

```



The rest of this section is left as an exercise for Brian.

3 Conditional Regression Information