

Analyses and plots for Bartek, Lewis, Vasishth and Smith, 2009

Version dated June 2, 2009

Contents

1	Description, paradigm details	3
2	Required libraries	3
3	GG SPR initial data prep	3
4	Accuracy	4
5	Trimming	10
6	Plots of RTs by condition, subject, and item	12
6.1	All the subjects	12
6.2	Accurate subjects only	15
6.3	Set up contrasts	18
7	BB SPR Initial data prep	20
8	Accuracy	21
9	Trimming	27
10	Plots of RTs by condition, subject, and item	28
10.1	All the subjects	28

<i>CONTENTS</i>	2
10.2 Accurate subjects only	31
10.3 Set up contrasts	34
11 Frequency and length processing	36
11.1 Spill-over preprocessing	37
11.2 Set up contrasts	46
12 Frequency and length processing	49
13 Merge accuracy	49
13.1 Spill-over preprocessing	50
13.2 Set up contrasts	58
14 HPD plots	63
14.1 HPD plots for locality effects within matrix and embedded conditions	63
14.2 HPD plots for embedding effects	68

1 Description, paradigm details

This is a replication of Grodner & Gibson 2005 Experiment 2 using SPR, with new materials in which all words are 4-6 letters, and have a frequency >50-per-million in the American National Corpus First Release.

Run Fall 2006 and Winter 2007.

2 Required libraries

```
> library(reshape)
> library(xtable)
> library(lattice)
> library(Hmisc)
> library(lme4)
> source("shravan-rprofile.r")
> options(device = "pdf", prompt = " ", continue = " ")
```

3 GG SPR initial data prep

Create figs subdirectory.

```
if (!is.element("figs", system("ls -d"))[1]) {
  system("mkdir figs")
}
```

Merge RT data with length and frequency statistics for each word

```
raw.spr.data <- read.table("gg-spr06-data.txt")
colnames(raw.spr.data) <- c("subj", "expt", "item", "condition", "roi",
  "word", "RT", "embedding",
  "intervention")
exp.items <- subset(raw.spr.data, expt == "E1")
exp.items$word <- tolower(as.character(exp.items$word))
exp.items$word <- as.character(sub("\\.", "", exp.items$word))
lex <- read.table("anc-lexicon.txt", quote = "", skip = 1)
colnames(lex) <- c("word", "freq")
lex$word <- tolower(as.character(lex$word))
exp.items <- merge(exp.items, lex, by = c("word"), all.x = TRUE)
exp.items$len <- nchar(as.character(exp.items$word))
```

Merge comprehension Q accuracy with data. GG-SPR output did not have the comprehension questions defined as a separate ROI with a column for accuracy. Thus, for this experiment (but not BB-SPR or either eyetracking expt), accuracy data was created in a different file and read in.

```

q.data ← read.table("gg-spr06-question-data.txt")
colnames(q.data) ← c("subj", "expt", "item", "condition", "roi",
  "response", "correct",
  "RT")
q.data ← q.data[, c(1, 2, 3, 4, 7), ]
q.data$correct ← ifelse(q.data$correct == 1, "correct", "incorrect")
d ← merge(exp.items, q.data)

```

Recode position so the first position is 1, not zero. Then code the critical verb in each condition and subset out the critical verb.

```
d$roi ← as.integer(as.character(d$roi)) + 1
```

```

mean.freq ← mean(d$freq, na.rm = TRUE)
mean.len ← mean(d$len, na.rm = TRUE)
d$cfreq ← d$freq - mean.freq
d$clen ← (1/d$len) - mean.len

```

Make item and subject categorical factors, not numerical values:

```

d$item ← factor(d$item)
d$subj ← factor(d$subj)

```

4 Accuracy

Count accuracy and errors for each subject:

```

d.melt ← melt(d, measure = "correct", variable_name = "measure", na.rm
  = FALSE)
subj.acc ← cast(d.melt, subj ~ value)
subj.acc$incorrect[is.na(subj.acc$incorrect)] ← 0
subj.acc$percent ← round(100 * subj.acc$correct / (subj.acc$correct +
  subj.acc$incorrect))

```

```
xtable(subj.acc, caption = "Accuracies for each subject.")
```

```

xtable(sort_df(subj.acc, vars = "percent"), caption = "Accuracies for
  each subject, sorted by accuracy from worst to best.")

```

Who are the bad/lazy subjects?

```
d$good.subj ← subj.acc$percent[d$subj] > 70
```

	subj	correct	incorrect	percent
1	1	398	176.00	69.00
2	2	423	137.00	76.00
3	3	428	128.00	77.00
4	4	434	142.00	75.00
5	5	436	117.00	79.00
6	6	372	174.00	68.00
7	7	422	152.00	74.00
8	8	1568	732.00	68.00
9	9	312	263.00	54.00
10	10	421	155.00	73.00
11	11	417	136.00	75.00
12	12	450	119.00	79.00
13	13	388	186.00	68.00
14	14	456	119.00	79.00
15	15	377	198.00	66.00
16	16	470	106.00	82.00
17	17	453	100.00	82.00
18	18	395	174.00	69.00
19	19	416	158.00	72.00
20	20	391	184.00	68.00
21	21	435	140.00	76.00
22	22	487	89.00	85.00
23	23	377	176.00	68.00
24	24	458	111.00	80.00
25	25	460	114.00	80.00
26	26	405	170.00	70.00
27	27	416	159.00	72.00
28	28	450	126.00	78.00
29	29	423	149.00	74.00
30	30	430	139.00	76.00
31	31	439	135.00	76.00
32	32	445	130.00	77.00
33	33	501	74.00	87.00
34	34	450	126.00	78.00
35	35	407	146.00	74.00
36	36	426	143.00	75.00
37	37	472	102.00	82.00
38	38	357	218.00	62.00
39	39	497	78.00	86.00
40	40	389	187.00	68.00
41	41	413	159.00	72.00
42	42	366	203.00	64.00
43	43	451	123.00	79.00
44	44	375	200.00	65.00
45	45	464	111.00	81.00
46	46	321	255.00	56.00
47	47	462	91.00	84.00
48	48	428	141.00	75.00
49	49	412	162.00	72.00

Table 1: Accuracies for each subject.

	subj	correct	incorrect	percent
9	9	312	263.00	54.00
46	46	321	255.00	56.00
38	38	357	218.00	62.00
42	42	366	203.00	64.00
44	44	375	200.00	65.00
15	15	377	198.00	66.00
6	6	372	174.00	68.00
8	8	1568	732.00	68.00
13	13	388	186.00	68.00
20	20	391	184.00	68.00
23	23	377	176.00	68.00
40	40	389	187.00	68.00
1	1	398	176.00	69.00
18	18	395	174.00	69.00
26	26	405	170.00	70.00
19	19	416	158.00	72.00
27	27	416	159.00	72.00
41	41	413	159.00	72.00
49	49	412	162.00	72.00
10	10	421	155.00	73.00
7	7	422	152.00	74.00
29	29	423	149.00	74.00
35	35	407	146.00	74.00
4	4	434	142.00	75.00
11	11	417	136.00	75.00
36	36	426	143.00	75.00
48	48	428	141.00	75.00
2	2	423	137.00	76.00
21	21	435	140.00	76.00
30	30	430	139.00	76.00
31	31	439	135.00	76.00
3	3	428	128.00	77.00
32	32	445	130.00	77.00
28	28	450	126.00	78.00
34	34	450	126.00	78.00
5	5	436	117.00	79.00
12	12	450	119.00	79.00
14	14	456	119.00	79.00
43	43	451	123.00	79.00
24	24	458	111.00	80.00
25	25	460	114.00	80.00
45	45	464	111.00	81.00
16	16	470	106.00	82.00
17	17	453	100.00	82.00
37	37	472	102.00	82.00
47	47	462	91.00	84.00
22	22	487	89.00	85.00
39	39	497	78.00	86.00
33	33	501	74.00	87.00

Table 2: Accuracies for each subject, sorted by accuracy from worst to best.

Count accuracy and errors for each item:

```
item.acc ← cast(d.melt, item ~ value)
item.acc$incorrect[is.na(item.acc$incorrect)] ← 0
item.acc$percent ← round(100 * item.acc$correct/(item.acc$correct +
  item.acc$incorrect))
```

```
xtable(item.acc, caption = "Accuracies for each item.")
```

	item	correct	incorrect	percent
1	1	489	497.00	50.00
2	2	727	338.00	68.00
3	3	432	425.00	50.00
4	4	850	16.00	98.00
5	5	925	279.00	77.00
6	6	551	426.00	56.00
7	7	721	483.00	60.00
8	8	758	126.00	86.00
9	9	770	219.00	78.00
10	10	1106	20.00	98.00
11	11	831	294.00	74.00
12	12	876	289.00	75.00
13	13	726	260.00	74.00
14	14	333	472.00	41.00
15	15	717	112.00	86.00
16	16	606	260.00	70.00
17	17	713	280.00	72.00
18	18	656	269.00	71.00
19	19	877	85.00	91.00
20	20	865	44.00	95.00
21	21	712	337.00	68.00
22	22	939	55.00	94.00
23	23	950	19.00	98.00
24	24	526	451.00	54.00
25	25	874	0.00	100.00
26	26	133	880.00	13.00
27	27	1068	21.00	98.00
28	28	505	413.00	55.00
29	29	872	97.00	90.00
30	30	735	346.00	68.00

Table 3: Accuracies for each item.

```
xtable(sort_df(item.acc, vars = "percent"), caption = "Accuracies for
  each item, sorted by accuracy from worst to best.")
```

	item	correct	incorrect	percent
26	26	133	880.00	13.00
14	14	333	472.00	41.00
1	1	489	497.00	50.00
3	3	432	425.00	50.00
24	24	526	451.00	54.00
28	28	505	413.00	55.00
6	6	551	426.00	56.00
7	7	721	483.00	60.00
2	2	727	338.00	68.00
21	21	712	337.00	68.00
30	30	735	346.00	68.00
16	16	606	260.00	70.00
18	18	656	269.00	71.00
17	17	713	280.00	72.00
11	11	831	294.00	74.00
13	13	726	260.00	74.00
12	12	876	289.00	75.00
5	5	925	279.00	77.00
9	9	770	219.00	78.00
8	8	758	126.00	86.00
15	15	717	112.00	86.00
29	29	872	97.00	90.00
19	19	877	85.00	91.00
22	22	939	55.00	94.00
20	20	865	44.00	95.00
4	4	850	16.00	98.00
10	10	1106	20.00	98.00
23	23	950	19.00	98.00
27	27	1068	21.00	98.00
25	25	874	0.00	100.00

Table 4: Accuracies for each item, sorted by accuracy from worst to best.

Count accuracy and errors for each condition

```
condition.acc ← cast(d.melt, condition ~ value)
condition.acc$incorrect[is.na(condition.acc$incorrect)] ← 0
condition.acc$percent ← round(100 *
  condition.acc$correct/(condition.acc$correct +
    condition.acc$incorrect))
```

```
xtable(condition.acc, caption = "Accuracies for each condition.")
```

	condition	correct	incorrect	percent
1	a	2910	776.00	79.00
2	b	3623	875.00	81.00
3	c	4009	1012.00	80.00
4	d	3529	1172.00	75.00
5	e	3811	1806.00	68.00
6	f	3961	2172.00	65.00

Table 5: Accuracies for each condition.

4.1 Self-paced reading spill-over preprocessing

Compute spillover information.

```
matrix ← subset(d, select = c(subj, condition, item, word, roi, RT,
  len, freq, correct))
matrix$roi ← matrix$roi + 1
matrix$prev.RT ← matrix$RT
matrix$prev.len ← matrix$len
matrix$prev.freq ← matrix$freq
matrix$RT ← NULL
matrix$len ← NULL
matrix$freq ← NULL
d ← merge(d, matrix, by = c("subj", "item", "condition", "roi",
  "correct"), all.x = TRUE)
d$word ← d$word.x
d$word.x ← NULL
d$word.y ← NULL
```

```
mean.prev.len ← mean(d$prev.len, na.rm = TRUE, trim = 0)
mean.prev.freq ← mean(d$prev.freq, na.rm = TRUE, trim = 0)
d$prev.cfreq ← d$prev.freq - mean.prev.freq
d$prev.clen ← (1/d$prev.len) - mean.prev.len
```

5 Trimming

Now we do a quick and dirty trim of the RTs by 3SDs (this replicates the original Grodner & Gibson (2005) procedure).

```
d.melt <- melt(d, measure = "RT", variable_name = "measure", na.rm = FALSE)
full.table <- length(d.melt$value)
```

```
RT.sd <- cast(d.melt, condition ~ ., function(x) c(m = round(mean(x)),
  sd = sd(x)))
RT.sd$high.cutoff <- RT.sd$m + 3 * RT.sd$sd
RT.sd$low.cutoff <- RT.sd$m - 3 * RT.sd$sd
for (c in levels(d.melt$condition)) {
  d.melt$value[(d.melt$condition == RT.sd$condition[c]) &
    (d.melt$value > RT.sd$high.cutoff[c])] <- NA
  d.melt$value[(d.melt$condition == RT.sd$condition[c]) &
    (d.melt$value < RT.sd$low.cutoff[c])] <- NA
}
d.melt <- d.melt[!is.na(d.melt$value), ]
trimmed.table <- print(length(d.melt$value))
```

```
[1] 35188
```

```
print((full.table - trimmed.table)/full.table)
```

```
[1] 0
```

```
rtlimits <- c(0, 1200)
sdlimits <- c(0, 1200)
```

```
d.melt <- gg.d.melt
```

```
ci <- function(scores) {
  m <- mean(scores)
  stderr <- se(scores)
  len <- length(scores)
  upper <- m + stderr
  lower <- m - stderr
  return(data.frame(lower = lower, upper = upper))
}
plot.intervals <- function(xrange, lower, upper, width, lty = "solid",
  col = "grey", lwd = 1) {
  for (j in 1:(length(xrange))) {
```

```

        lines(x = rep(xrange[j], 2), y = c(lower[j], upper[j]), lty = 11
              lty, col = col, lwd = lwd,
              type = "l")
        arrows(x0 = as.numeric(xrange[j]), y0 = lower[j], x1 = 12
              as.numeric(xrange[j]), y1 = upper[j],
              length = width, angle = 90, code = 3, lty = lty, col = col, 13
              lwd = lwd)
    }
}
add.axes.and.title ← function(title, line = 0.8) {
    axis(1, at = 1:3, labels = c("none", "PP", "RC"))
    axis(1, at = 4:6, labels = c("none", "PP", "RC"))
    lab.pos ← rtlimits[1] - diff(rtlimits) * 0.23
    text(labels = c("Matrix", "Embedded"), x = c(2, 5), y = c(lab.pos, 20
    lab.pos), cex = 1.1)
    mtext(title, side = 3, line = line, cex = 1.2 * 0.8, font = 2)
}

```

6 Plots of RTs by condition, subject, and item

6.1 All the subjects

```

plot.data ← d.melt[d.melt$roi <= 11, ]
reorder.frame ← data.frame(condition = c(rep("a", 11), rep("b", 11),
  rep("c", 11), rep("d",
  11), rep("e", 11), rep("f", 11)), roi = rep(1:11, 6), plot.order =
  c(1, 5, 11, NA,
  NA, NA, NA, NA, NA, NA, NA, 1, 5, 8, 9, 10, 11, NA, NA, NA, NA, NA,
  1, 5, 6, 7, 8,
  9, 10, 11, NA, NA, NA, NA, 1, 2, 3, 4, 5, 11, NA, NA, NA, NA, NA, 1, 2,
  3, 4, 5, 8, 9,
  10, 11, NA, NA, 1:11))
plot.data ← merge(plot.data, reorder.frame)

```

```

color1 ← gray(0.6)
color2 ← gray(0.3)
color3 ← "black"
lty1 ← "dashed"
lty2 ← "solid"

```

```

trim.value ← 60000
rt.mean ← cast(plot.data, condition + plot.order ~ ., function(x) c(m =
  round(mean(x)),
  CI = round(ci(x))), subset = ((value < trim.value)))
rt.mean ← rt.mean[!(rt.mean$plot.order == "NA"), ]
rt.mean ← rt.mean[-c(4, 11, 20, 27, 37), ]
rt.sd.subj ← cast(d.melt, condition + subj ~ ., function(x)
  round(sd(x)), subset = ((value <
  trim.value)))
rt.sd ← cast(d.melt, condition ~ ., function(x) round(mean(x, na.rm =
  TRUE)))
dummy ← data.frame(region = c(1:11), m = rep(-999, 11))
subtitle ← "Error bars are one standard error."

```

```

plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
  cex.lab = 1.5, cex.main = 1.5,
  cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
  "Experiment 1 self-paced reading times",
  sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), line = 1, cex = 0.7, c("The", "(admini-",
  "who", "the)",
  "nurse", "(who", "was", "(from", "the", "clinic)", "supervised"))
mtext(side = 1, at = c(1:11), line = 2, cex = 0.7, c("", "strator", "",
  "", "", "", "",
  "", "", "", ""))
legend(x = rt.mean$plot.order[1], y = 1000, legend = c("condition A",
  "condition B", "condition C",

```

```

"condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$CI.lower[rt.mean$plot.order == 11],
  upper = rt.mean$CI.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(2)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "d"
dom1 <- 1:5
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type <- c(15)
color <- color1
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "e"
dom1 <- 1:5
dom2 <- 8:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(17)

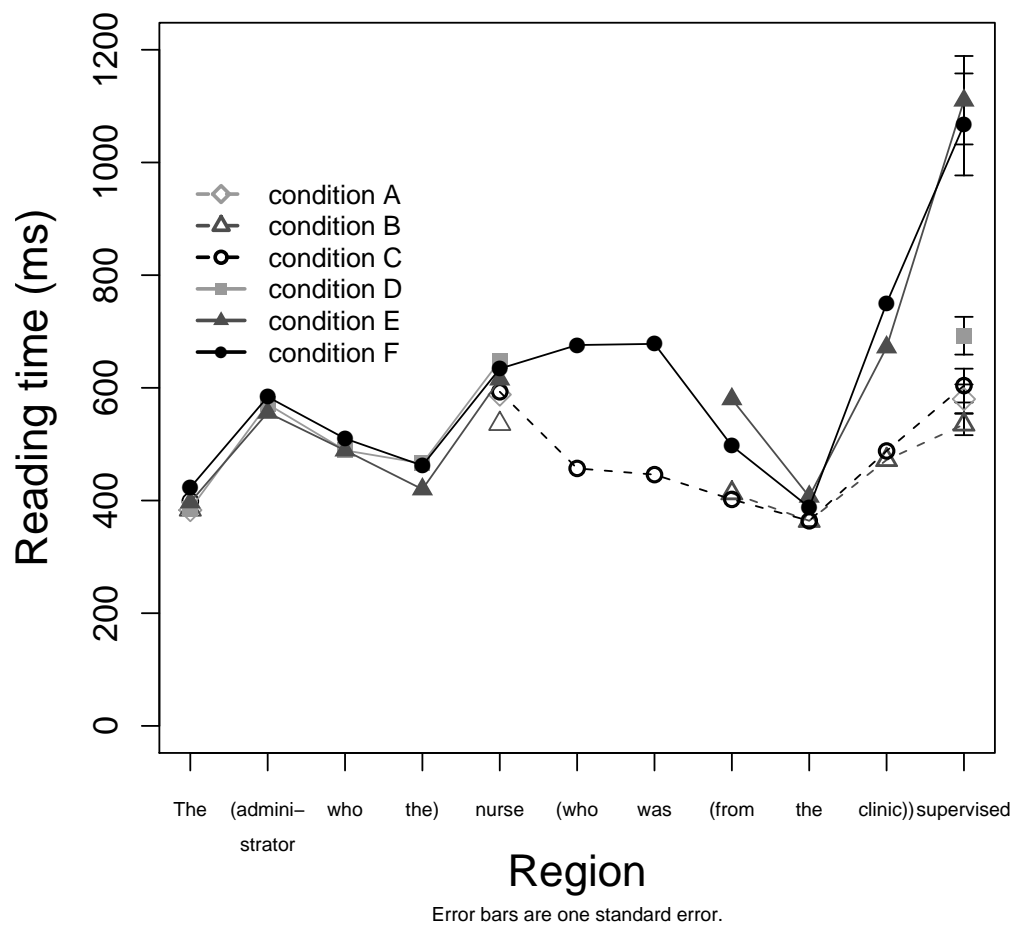
```

```

color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")

```

Experiment 1 self-paced reading times



6.2 Accurate subjects only

Now let's look at the RT pattern for the accurate subjects (defined above).

```
trim.value ← 60000
rt.mean ← cast(plot.data, condition + plot.order ~ ., function(x) c(m =
  round(mean(x)),
  CI = round(ci(x))), subset = ((value < trim.value) & (good.subj ==
  TRUE)))
rt.mean ← rt.mean[!(rt.mean$plot.order == "NA"), ]
rt.mean ← rt.mean[-c(4, 11, 20, 27, 37), ]
```

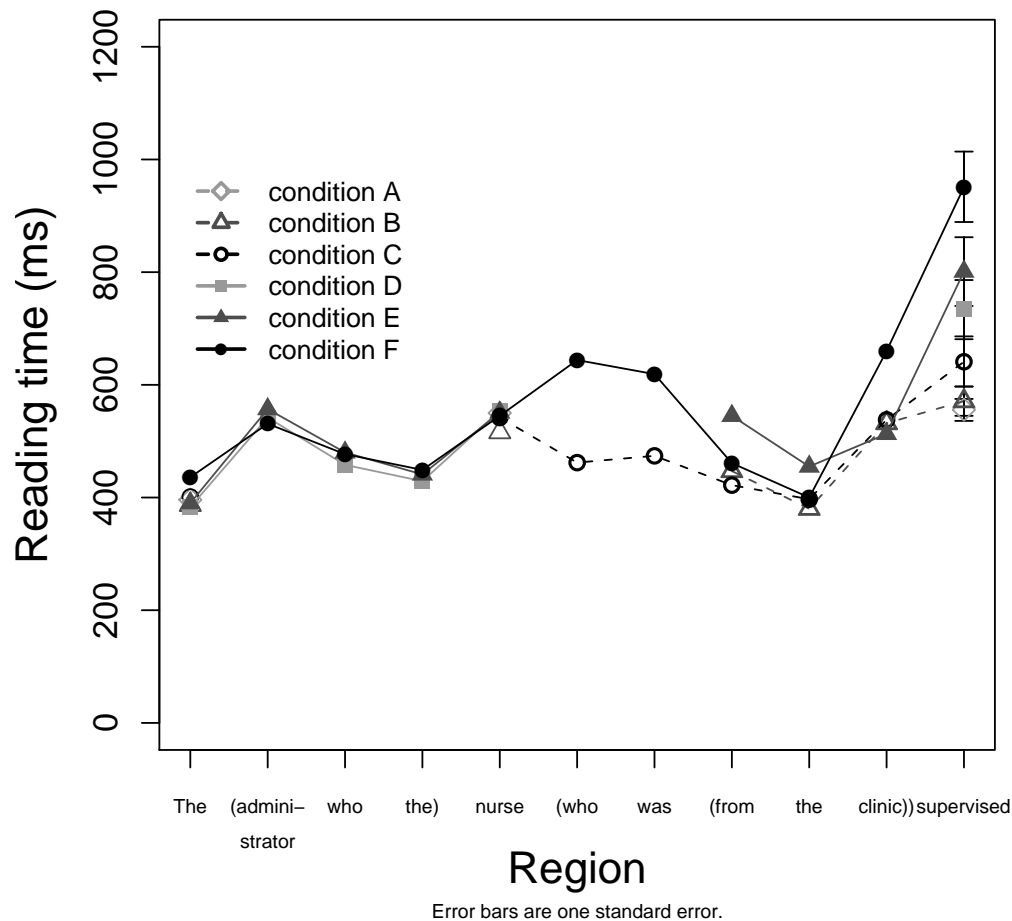
```
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
  cex.lab = 1.5, cex.main = 1.5,
  cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
  "Experiment 1 self-paced reading times",
  sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), line = 1, cex = 0.7, c("The", "(admini-",
  "who", "the)",
  "nurse", "(who", "was", "(from", "the", "clinic)", "supervised"))
mtext(side = 1, at = c(1:11), line = 2, cex = 0.7, c("", "strator", "",
  "", "", "", "",
  "", "", "", ""))
legend(x = rt.mean$plot.order[1], y = 1000, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$CI.lower[rt.mean$plot.order == 11],
  upper = rt.mean$CI.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition ← "a"
point.type ← c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1)
condition ← "b"
dom1 ← 8:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(2)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "c"
```

```

dom1 ← 5:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type ← c(1)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "d"
dom1 ← 1:5
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type ← c(15)
color ← color1
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "e"
dom1 ← 1:5
dom2 ← 8:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(17)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")

```

Experiment 1 self-paced reading times



```
d <- d.melt
```

1

```
d$criticalv1 <- ifelse((d$condition == "a" & d$roi == 3) | (d$condition == "b" & d$roi == 6) | (d$condition == "c" & d$roi == 8) | (d$condition == "d" & d$roi == 6) | (d$condition == "e" & d$roi == 9) | (d$condition == "f" & d$roi == 11), "yes", "no")
d <- subset(d, criticalv1 == "yes")
d$criticalv1 <- factor(d$criticalv1)
full.table.v1 <- print(length(d$value))
```

1

2

3

4

5

6

```
[1] 1863
```

1

6.3 Set up contrasts

It's often nice to be able to interpret the contrast coefficients of contrasts as if they represent the estimated mean difference between the two groups represented by the contrast. In order to ensure this, the contrasts must be normalized as follows (e.g. see p. B-13 in Maxwell & Delaney second edition). This new function will also check to be sure that the contrasts are orthogonal, and they are in fact contrasts.

```
normalize.and.check.contrasts ← function(contr) {
  for (col in 1:ncol(contr)) {
    pos.score ← mean(contr[contr[, col] > 0, col])
    neg.score ← mean(contr[contr[, col] < 0, col])
    diff ← pos.score - neg.score
    contr[, col] ← contr[, col]/diff
  }
  if (!isTRUE(all.equal(mean(apply(contr, MARGIN = 2, FUN = mean)),
    0)))
    message("WARNING!!! Contrast matrix contains non-contrasts!!")
  m ← t(contr) %*% contr
  should.be.diag ← m - diag(diag(m))
  check.diag ← all(sapply(as.vector(should.be.diag), FUN =
    function(x) {
      return(all.equal(x, 0))
    })))
  if (!check.diag)
    message("WARNING!!! Contrast matrix is non-orthogonal.")
  return(contr)
}
```

Now set up and normalize the first contrast set.

```
conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1.orig ← cbind(c1 = c(-1, -1, -1, 1, 1, 1), c2 = c(-2, 1, 1,
  0, 0, 0), c3 = c(0,
  -1, 1, 0, 0, 0), c4 = c(0, 0, 0, -2, 1, 1), c5 = c(0, 0, 0, 0, -1,
  1))
rownames(contr.set1.orig) ← conditions
contr.set1 ← normalize.and.check.contrasts(contr.set1.orig)
```

Let's take a look at the codings:

```
xtable(contr.set1.orig, caption = "Original coding of contrast set 1.")
```

```
xtable(contr.set1, caption = "Normalized coding of contrast set 1.")
```

For the second contrast set we only have two contrasts of interest and use the R contrast function to find orthogonal contrasts to fill out the set.

Contrast c6 is interaction of embedding and locality. Contrast c7 is interaction of embedding and modification type Contrasts 8-10 are not theoretically interesting; weights are assigned by R.

	c1	c2	c3	c4	c5
a	-1.00	-2.00	0.00	0.00	0.00
b	-1.00	1.00	-1.00	0.00	0.00
c	-1.00	1.00	1.00	0.00	0.00
d	1.00	0.00	0.00	-2.00	0.00
e	1.00	0.00	0.00	1.00	-1.00
f	1.00	0.00	0.00	1.00	1.00

Table 6: Original coding of contrast set 1.

	c1	c2	c3	c4	c5
a	-0.50	-0.67	0.00	0.00	0.00
b	-0.50	0.33	-0.50	0.00	0.00
c	-0.50	0.33	0.50	0.00	0.00
d	0.50	0.00	0.00	-0.67	0.00
e	0.50	0.00	0.00	0.33	-0.50
f	0.50	0.00	0.00	0.33	0.50

Table 7: Normalized coding of contrast set 1.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set2.orig ← cbind(c6 = c(2, -1, -1, -2, 1, 1), c7 = c(0, 1, -1,
0, -1, 1))
contrasts(conditions, how.many = 5) ← contr.set2.orig
contr.set2.orig ← contrasts(conditions)
colnames(contr.set2.orig) ← c("c6", "c7", "c8", "c9", "c10")
contr.set2 ← normalize.and.check.contrasts(contr.set2.orig)

```

Let's take a look at the codings:

```
xtable(contr.set2.orig, caption = "Original coding of contrast set 2.")
```

	c6	c7	c8	c9	c10
a	2.00	0.00	0.11	-0.46	-0.52
b	-1.00	1.00	-0.56	0.25	-0.34
c	-1.00	-1.00	-0.41	-0.47	0.33
d	-2.00	0.00	0.69	-0.01	-0.17
e	1.00	-1.00	0.01	0.71	0.01
f	1.00	1.00	0.17	-0.02	0.69

Table 8: Original coding of contrast set 2.

```
xtable(contr.set2, caption = "Normalized coding of contrast set 2.")
```

Now we merge in the contrasts to the data frame:

	c6	c7	c8	c9	c10
a	0.75	0.00	0.15	-0.64	-0.76
b	-0.38	0.50	-0.77	0.35	-0.50
c	-0.38	-0.50	-0.56	-0.66	0.48
d	-0.75	0.00	0.94	-0.01	-0.24
e	0.38	-0.50	0.02	0.98	0.02
f	0.38	0.50	0.23	-0.02	1.00

Table 9: Normalized coding of contrast set 2.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1 ← as.data.frame(contr.set1)
contr.set1$condition ← conditions
contr.set2 ← as.data.frame(contr.set2)
contr.set2$condition ← conditions
d ← merge(d, contr.set1, by = c("condition"))
d ← merge(d, contr.set2, by = c("condition"))

```

```

gg.spr.data ← d
write.table(gg.spr.data, file = "gg-spr-analysis-input.txt", sep =
  "\t", quote = FALSE,
  row.names = FALSE, col.names = TRUE)

```

7 BB SPR Initial data prep

Merge RT with length and frequency statistics for each word.

```

raw.spr.data ← read.table("bb-spr06-data.txt")
colnames(raw.spr.data) ← c("subj", "expt", "item", "condition", "roi",
  "word", "correct",
  "RT")
exp.items ← subset(raw.spr.data, expt == "E1")

```

```

no.questions ← subset(exp.items, roi != "?")
no.questions ← no.questions[, c(1, 2, 3, 4, 5, 6, 8)]
q.data ← exp.items[exp.items$roi == "?", ]
q.data ← q.data[, c(1, 2, 3, 4, 7), ]
q.data$correct ← ifelse(q.data$correct == 1, "correct", "incorrect")
exp.items ← no.questions
exp.items$word ← tolower(exp.items$word)
exp.items$word ← as.character(sub("\\.", "", exp.items$word))

```

```

exp.items$len ← nchar(as.character(exp.items$word))

```

```
exp.items$word ← gsub("saved", "save", exp.items$word, fixed = TRUE) 1
exp.items$word ← gsub("traded", "trade", exp.items$word, fixed = TRUE) 2
exp.items$word ← gsub("typed", "type", exp.items$word, fixed = TRUE) 3
exp.items$word ← gsub("parked", "park", exp.items$word, fixed = TRUE) 4
exp.items$word ← gsub("walked", "walk", exp.items$word, fixed = TRUE) 5
exp.items$word ← gsub("hated", "hate", exp.items$word, fixed = TRUE) 6
```

```
lex ← read.table("anc-lexicon.txt", quote = "", skip = 1) 1
colnames(lex) ← c("word", "freq") 2
lex$word ← tolower(as.character(lex$word)) 3
exp.items ← merge(exp.items, lex, by = c("word"), all.x = TRUE) 4
```

```
d ← merge(exp.items, q.data) 1
```

Remove subject 702 because this subject did not complete the experiment.

```
d ← d[d$subj != "702", ] 1
```

Remove item 9. It was improperly designed.

```
d ← d[d$item != "9" & d$item != "13" & d$item != "36", ] 1
```

Recode position so the first position is 1, not zero. Then code the critical verb in each condition and subset out the critical verb.

```
d$roi ← as.integer(as.character(d$roi)) + 1 1
```

```
mean.freq ← mean(d$freq, na.rm = TRUE) 1
mean.len ← mean(d$len, na.rm = TRUE) 2
d$cfreq ← d$freq - mean.freq 3
d$clen ← (1/d$len) - mean.len 4
```

Make item and subject categorical factors, not numerical values:

```
d$item ← factor(d$item) 1
d$subj ← factor(d$subj) 2
```

8 Accuracy

Count accuracy and errors for each subject:

```
d.melt <- melt(d, measure = "correct", variable_name = "measure", na.rm  
  = FALSE) 1  
subj.acc <- cast(d.melt, subj ~ value) 2  
subj.acc$incorrect[is.na(subj.acc$incorrect)] <- 0 3  
subj.acc$percent <- round(100 * subj.acc$correct/(subj.acc$correct + 4  
  subj.acc$incorrect))
```

```
xtable(subj.acc, caption = "Accuracies for each subject.") 1
```

```
xtable(sort_df(subj.acc, vars = "percent"), caption = "Accuracies for  
  each subject, sorted by accuracy from worst to best.") 1
```

Who are the bad/lazy subjects?

```
d$good.subj <- subj.acc$percent[d$subj] > 70 1
```

	subj	correct	incorrect	percent
1	1	294	35.00	89.00
2	2	317	14.00	96.00
3	3	285	53.00	84.00
4	4	292	38.00	88.00
5	5	330	0.00	100.00
6	6	336	0.00	100.00
7	7	319	10.00	97.00
8	8	271	60.00	82.00
9	9	328	10.00	97.00
10	10	293	37.00	89.00
11	11	1240	80.00	94.00
12	12	305	31.00	91.00
13	13	291	38.00	88.00
14	14	316	15.00	95.00
15	15	306	32.00	91.00
16	17	315	15.00	95.00
17	18	232	104.00	69.00
18	19	295	34.00	90.00
19	20	240	91.00	73.00
20	21	326	12.00	96.00
21	22	203	40.00	84.00
22	23	277	53.00	84.00
23	101	330	0.00	100.00
24	102	308	28.00	92.00
25	103	305	24.00	93.00
26	104	317	14.00	96.00
27	105	328	10.00	97.00
28	106	1206	114.00	91.00
29	107	316	14.00	96.00
30	108	320	16.00	95.00
31	109	276	53.00	84.00
32	110	304	27.00	92.00
33	111	338	0.00	100.00
34	112	263	67.00	80.00
35	113	300	30.00	91.00
36	114	327	9.00	97.00
37	115	281	48.00	85.00
38	116	331	0.00	100.00
39	117	308	30.00	91.00
40	119	330	0.00	100.00
41	120	323	13.00	96.00
42	121	307	22.00	93.00
43	122	331	0.00	100.00
44	123	300	38.00	89.00
45	128	275	12.00	96.00
46	129	315	23.00	93.00
47	704	331	0.00	100.00

Table 10: Accuracies for each subject.

	subj	correct	incorrect	percent
17	18	232	104.00	69.00
19	20	240	91.00	73.00
34	112	263	67.00	80.00
8	8	271	60.00	82.00
3	3	285	53.00	84.00
21	22	203	40.00	84.00
22	23	277	53.00	84.00
31	109	276	53.00	84.00
37	115	281	48.00	85.00
4	4	292	38.00	88.00
13	13	291	38.00	88.00
1	1	294	35.00	89.00
10	10	293	37.00	89.00
44	123	300	38.00	89.00
18	19	295	34.00	90.00
12	12	305	31.00	91.00
15	15	306	32.00	91.00
28	106	1206	114.00	91.00
35	113	300	30.00	91.00
39	117	308	30.00	91.00
24	102	308	28.00	92.00
32	110	304	27.00	92.00
25	103	305	24.00	93.00
42	121	307	22.00	93.00
46	129	315	23.00	93.00
11	11	1240	80.00	94.00
14	14	316	15.00	95.00
16	17	315	15.00	95.00
30	108	320	16.00	95.00
2	2	317	14.00	96.00
20	21	326	12.00	96.00
26	104	317	14.00	96.00
29	107	316	14.00	96.00
41	120	323	13.00	96.00
45	128	275	12.00	96.00
7	7	319	10.00	97.00
9	9	328	10.00	97.00
27	105	328	10.00	97.00
36	114	327	9.00	97.00
5	5	330	0.00	100.00
6	6	336	0.00	100.00
23	101	330	0.00	100.00
33	111	338	0.00	100.00
38	116	331	0.00	100.00
40	119	330	0.00	100.00
43	122	331	0.00	100.00
47	704	331	0.00	100.00

Table 11: Accuracies for each subject, sorted by accuracy from worst to best.

Count accuracy and errors for each item:

```
item.acc ← cast(d.melt, item ~ value)
item.acc$incorrect[is.na(item.acc$incorrect)] ← 0
item.acc$percent ← round(100 * item.acc$correct/(item.acc$correct +
  item.acc$incorrect))
```

```
xtable(item.acc, caption = "Accuracies for each item.")
```

	item	correct	incorrect	percent
1	1	634	12.00	98.00
2	2	445	199.00	69.00
3	3	554	78.00	88.00
4	4	632	15.00	98.00
5	5	616	12.00	98.00
6	7	603	28.00	96.00
7	8	646	0.00	100.00
8	10	607	25.00	96.00
9	11	594	0.00	100.00
10	14	618	13.00	98.00
11	16	593	106.00	85.00
12	17	508	150.00	77.00
13	18	563	69.00	89.00
14	19	693	0.00	100.00
15	20	505	135.00	79.00
16	21	559	58.00	91.00
17	22	508	138.00	79.00
18	24	569	76.00	88.00
19	25	632	0.00	100.00
20	26	628	10.00	98.00
21	28	622	33.00	95.00
22	29	578	53.00	92.00
23	30	640	48.00	93.00
24	31	635	10.00	98.00
25	33	607	10.00	98.00
26	34	611	89.00	87.00
27	35	681	27.00	96.00

Table 12: Accuracies for each item.

```
xtable(sort_df(item.acc, vars = "percent"), caption = "Accuracies for
  each item, sorted by accuracy from worst to best.")
```

	item	correct	incorrect	percent
2	2	445	199.00	69.00
12	17	508	150.00	77.00
15	20	505	135.00	79.00
17	22	508	138.00	79.00
11	16	593	106.00	85.00
26	34	611	89.00	87.00
3	3	554	78.00	88.00
18	24	569	76.00	88.00
13	18	563	69.00	89.00
16	21	559	58.00	91.00
22	29	578	53.00	92.00
23	30	640	48.00	93.00
21	28	622	33.00	95.00
6	7	603	28.00	96.00
8	10	607	25.00	96.00
27	35	681	27.00	96.00
1	1	634	12.00	98.00
4	4	632	15.00	98.00
5	5	616	12.00	98.00
10	14	618	13.00	98.00
20	26	628	10.00	98.00
24	31	635	10.00	98.00
25	33	607	10.00	98.00
7	8	646	0.00	100.00
9	11	594	0.00	100.00
14	19	693	0.00	100.00
19	25	632	0.00	100.00

Table 13: Accuracies for each item, sorted by accuracy from worst to best.

Count accuracy and errors for each condition

```
condition.acc ← cast(d.melt, condition ~ value)
condition.acc$incorrect[is.na(condition.acc$incorrect)] ← 0
condition.acc$percent ← round(100 *
  condition.acc$correct/(condition.acc$correct +
    condition.acc$incorrect))
```

```
xtable(condition.acc, caption = "Accuracies for each condition.")
```

	condition	correct	incorrect	percent
1	a	2059	116.00	95.00
2	b	2676	231.00	92.00
3	c	3066	239.00	93.00
4	d	2166	234.00	90.00
5	e	2950	210.00	93.00
6	f	3164	364.00	90.00

Table 14: Accuracies for each condition.

8.1 Self-paced reading spill-over preprocessing

Compute spillover information.

```
matrix ← subset(d, select = c(subj, condition, item, word, roi, RT,
  len, freq, correct))
matrix$roi ← matrix$roi + 1
matrix$prev.RT ← matrix$RT
matrix$prev.len ← matrix$len
matrix$prev.freq ← matrix$freq
matrix$RT ← NULL
matrix$len ← NULL
matrix$freq ← NULL
d ← merge(d, matrix, by = c("subj", "item", "condition", "roi",
  "correct"), all.x = TRUE)
d$word ← d$word.x
d$word.x ← NULL
d$word.y ← NULL
```

```
mean.prev.len ← mean(d$prev.len, na.rm = TRUE, trim = 0)
mean.prev.freq ← mean(d$prev.freq, na.rm = TRUE, trim = 0)
d$prev.cfreq ← d$prev.freq - mean.prev.freq
d$prev.clen ← (1/d$prev.len) - mean.prev.len
```

9 Trimming

Now we do a quick and dirty trim of the RTs by 3SDs (this replicates the original Grodner & Gibson (2005) procedure).

```
d.melt <- melt(d, measure = "RT", variable_name = "measure", na.rm =
  FALSE)
full.table <- length(d.melt$value)
```

```
RT.sd <- cast(d.melt, condition ~ ., function(x) c(m = round(mean(x)),
  sd = sd(x)))
RT.sd$high.cutoff <- RT.sd$m + 3 * RT.sd$sd
RT.sd$low.cutoff <- RT.sd$m - 3 * RT.sd$sd
for (c in levels(d.melt$condition)) {
  d.melt$value[(d.melt$condition == RT.sd$condition[c]) &
    (d.melt$value > RT.sd$high.cutoff[c])] <- NA
  d.melt$value[(d.melt$condition == RT.sd$condition[c]) &
    (d.melt$value < RT.sd$low.cutoff[c])] <- NA
}
d.melt <- d.melt[!is.na(d.melt$value), ]
trimmed.table <- print(length(d.melt$value))
```

```
[1] 24035
```

```
print((full.table - trimmed.table)/full.table)
```

```
[1] 0
```

```
bb.d.melt <- d.melt
```

```
rtllimits <- c(0, 500)
sdlimits <- c(0, 500)
```

```
d.melt <- bb.d.melt
```

10 Plots of RTs by condition, subject, and item

10.1 All the subjects

```

plot.data ← d.melt[d.melt$roi <= 11, ]
reorder.frame ← data.frame(condition = c(rep("a", 11), rep("b", 11),
  rep("c", 11), rep("d",
  11), rep("e", 11), rep("f", 11)), roi = rep(1:11, 6), plot.order =
  c(1, 5, 11, NA,
  NA, NA, NA, NA, NA, NA, NA, 1, 5, 8, 9, 10, 11, NA, NA, NA, NA, NA,
  1, 5, 6, 7, 8,
  9, 10, 11, NA, NA, NA, 1, 2, 3, 4, 5, 11, NA, NA, NA, NA, NA, 1, 2,
  3, 4, 5, 8, 9,
  10, 11, NA, NA, 1:11))
plot.data ← merge(plot.data, reorder.frame)

```

```

trim.value ← 60000
rt.mean ← cast(plot.data, condition + plot.order ~ ., function(x) c(m =
  round(mean(x)),
  CI = round(ci(x))), subset = ((value < trim.value)))
rt.mean ← rt.mean[!(rt.mean$plot.order == "NA"), ]
rt.mean ← na.delete(rt.mean)
dummy ← data.frame(region = c(1:11), m = rep(-999, 11))
subtitle ← "Error bars are one standard error."

```

```

color1 ← gray(0.6)
color2 ← gray(0.3)
color3 ← "black"
lty1 ← "dashed"
lty2 ← "solid"

```

```

plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
  cex.lab = 1.5, cex.main = 1.5,
  cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
  "Experiment 3 self-paced reading times",
  sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), cex = 0.7, line = 1, c("The", "(sports",
  "that", "the)",
  "child", "(who", "was", "(from", "the", "school))", "played"))
legend(x = rt.mean$plot.order[1], y = 300, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$CI.lower[rt.mean$plot.order == 11],

```

```

    upper = rt.mean$CI.upper[rt.mean$plot.order == 11], width = 0.05,
      lwd = 1, col = "black",
      lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(2)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "d"
dom1 <- 1:5
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type <- c(15)
color <- color1
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "e"
dom1 <- 1:5
dom2 <- 8:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(17)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$m[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,

```

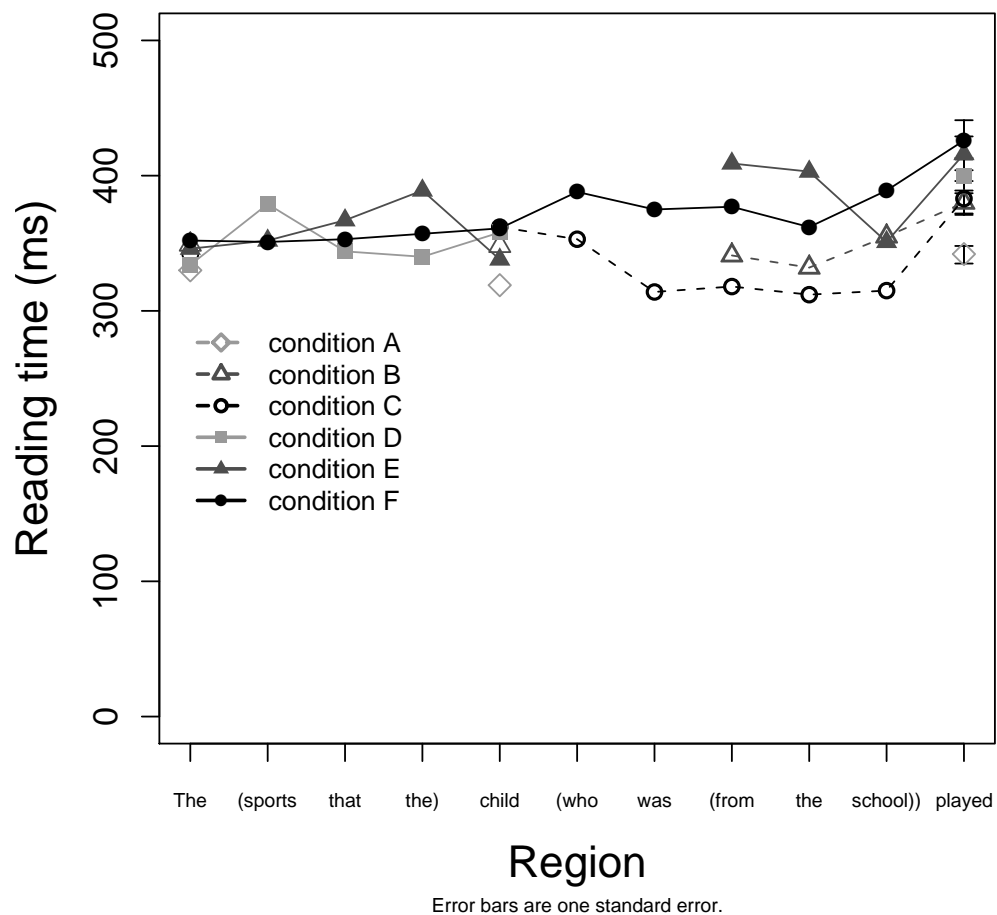
```

lwd = 1, type = "o")
condition <- "f"
dom1 <- 1:11
range1 <- rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order]
point.type <- c(16)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$m[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")

```

53
54
55
56
57
58
59
60

Experiment 3 self-paced reading times



10.2 Accurate subjects only

Now lets look at the RT pattern for the accurate subjects (defined above).

```

trim.value ← 60000
rt.mean ← cast(plot.data, condition + plot.order ~ ., function(x) c(m =
  round(mean(x)),
  CI = round(ci(x))), subset = ((value < trim.value) & (good.subj ==
  TRUE)))
rt.mean ← rt.mean[!(rt.mean$plot.order == "NA"), ]
rt.mean ← rt.mean[-c(4, 11, 20, 27, 37), ]

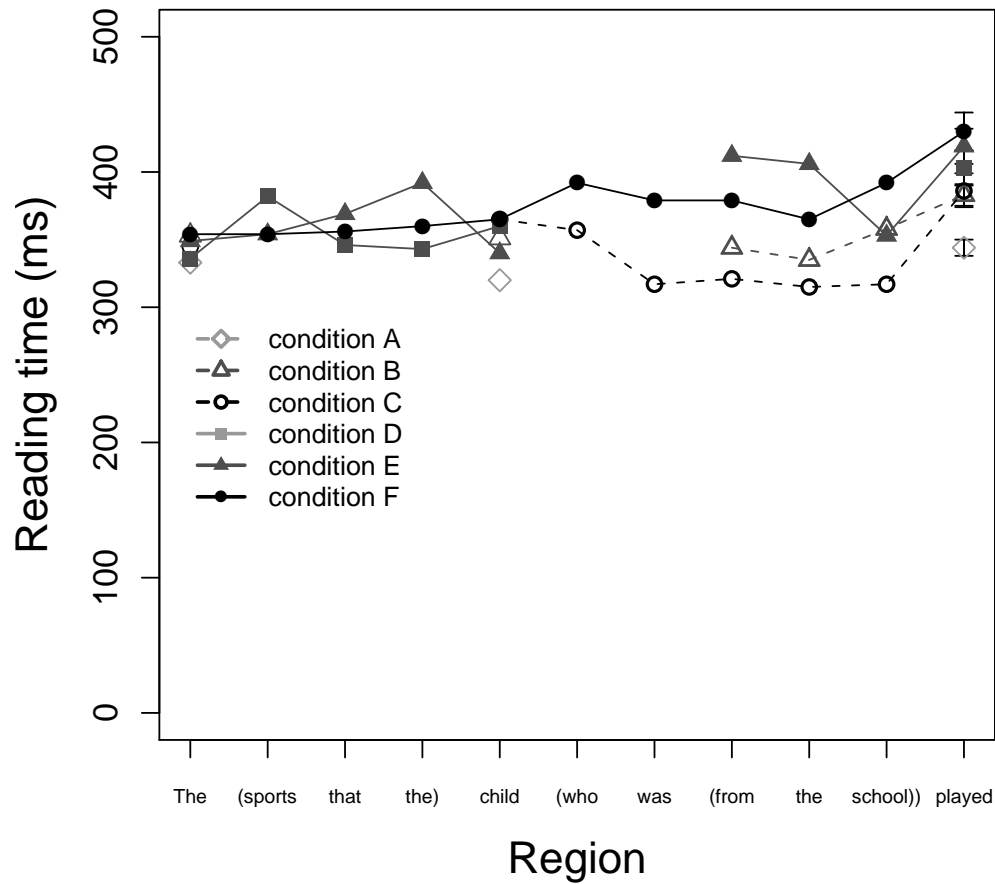
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
  cex.lab = 1.5, cex.main = 1.5,
  cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
  "Experiment 3 self-paced reading times",
  sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), cex = 0.7, line = 1, c("The", "(sports",
  "that", "the)",
  "child", "(who", "was", "(from", "the", "school))", "played"))
legend(x = rt.mean$plot.order[1], y = 300, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$CI.lower[rt.mean$plot.order == 11],
  upper = rt.mean$CI.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition ← "a"
point.type ← c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1)
condition ← "b"
dom1 ← 8:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(2)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$m[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "c"
dom1 ← 5:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
```

```

point.type ← c(1)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$m[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "d"
dom1 ← 1:5
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
                  < 6]
point.type ← c(15)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$m[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "e"
dom1 ← 1:5
dom2 ← 8:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
                  < 6]
range2 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order
                  > 7]
point.type ← c(17)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$m[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = "solid", col = color, pch =
      point.type, lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$m[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$m[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")

```

Experiment 3 self-paced reading times



```
d <- d.melt
```

1

```
d$criticalv1 <- ifelse((d$condition == "a" & d$roi == 3) | (d$condition == "b" & d$roi == 6) | (d$condition == "c" & d$roi == 8) | (d$condition == "d" & d$roi == 6) | (d$condition == "e" & d$roi == 9) | (d$condition == "f" & d$roi == 11), "yes", "no")
d <- subset(d, criticalv1 == "yes")
d$criticalv1 <- factor(d$criticalv1)
full.table.v1 <- print(length(d$value))
```

1

2

3

4

5

6

```
[1] 2005
```

1

10.3 Set up contrasts

It's often nice to be able to interpret the contrast coefficients of contrasts as if they represent the estimated mean difference between the two groups represented by the contrast. In order to ensure this, the contrasts must be normalized as follows (e.g. see p. B-13 in Maxwell & Delaney second edition). This new function will also check to be sure that the contrasts are orthogonal, and they are in fact contrasts.

```
normalize.and.check.contrasts ← function(contr) {
  for (col in 1:ncol(contr)) {
    pos.score ← mean(contr[contr[, col] > 0, col])
    neg.score ← mean(contr[contr[, col] < 0, col])
    diff ← pos.score - neg.score
    contr[, col] ← contr[, col]/diff
  }
  if (!isTRUE(all.equal(mean(apply(contr, MARGIN = 2, FUN = mean)),
    0)))
    message("WARNING!!! Contrast matrix contains non-contrasts!!")
  m ← t(contr) %*% contr
  should.be.diag ← m - diag(diag(m))
  check.diag ← all(sapply(as.vector(should.be.diag), FUN =
    function(x) {
      return(all.equal(x, 0))
    })))
  if (!check.diag)
    message("WARNING!!! Contrast matrix is non-orthogonal.")
  return(contr)
}
```

Now set up and normalize the first contrast set.

```
conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1.orig ← cbind(c1 = c(-1, -1, -1, 1, 1, 1), c2 = c(-2, 1, 1,
  0, 0, 0), c3 = c(0,
  -1, 1, 0, 0, 0), c4 = c(0, 0, 0, -2, 1, 1), c5 = c(0, 0, 0, 0, -1,
  1))
rownames(contr.set1.orig) ← conditions
contr.set1 ← normalize.and.check.contrasts(contr.set1.orig)
```

Let's take a look at the codings:

```
xtable(contr.set1.orig, caption = "Original coding of contrast set 1.")
```

```
xtable(contr.set1, caption = "Normalized coding of contrast set 1.")
```

For the second contrast set we only have two contrasts of interest and use the R contrast function to find orthogonal contrasts to fill out the set.

Contrast c6 is interaction of embedding and locality. Contrast c7 is interaction of embedding and modification type Contrasts 8-10 are not theoretically interesting; weights are assigned by R.

	c1	c2	c3	c4	c5
a	-1.00	-2.00	0.00	0.00	0.00
b	-1.00	1.00	-1.00	0.00	0.00
c	-1.00	1.00	1.00	0.00	0.00
d	1.00	0.00	0.00	-2.00	0.00
e	1.00	0.00	0.00	1.00	-1.00
f	1.00	0.00	0.00	1.00	1.00

Table 15: Original coding of contrast set 1.

	c1	c2	c3	c4	c5
a	-0.50	-0.67	0.00	0.00	0.00
b	-0.50	0.33	-0.50	0.00	0.00
c	-0.50	0.33	0.50	0.00	0.00
d	0.50	0.00	0.00	-0.67	0.00
e	0.50	0.00	0.00	0.33	-0.50
f	0.50	0.00	0.00	0.33	0.50

Table 16: Normalized coding of contrast set 1.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set2.orig ← cbind(c6 = c(2, -1, -1, -2, 1, 1), c7 = c(0, 1, -1,
0, -1, 1))
contrasts(conditions, how.many = 5) ← contr.set2.orig
contr.set2.orig ← contrasts(conditions)
colnames(contr.set2.orig) ← c("c6", "c7", "c8", "c9", "c10")
contr.set2 ← normalize.and.check.contrasts(contr.set2.orig)

```

Let's take a look at the codings:

```
xtable(contr.set2.orig, caption = "Original coding of contrast set 2.")
```

	c6	c7	c8	c9	c10
a	2.00	0.00	0.11	-0.46	-0.52
b	-1.00	1.00	-0.56	0.25	-0.34
c	-1.00	-1.00	-0.41	-0.47	0.33
d	-2.00	0.00	0.69	-0.01	-0.17
e	1.00	-1.00	0.01	0.71	0.01
f	1.00	1.00	0.17	-0.02	0.69

Table 17: Original coding of contrast set 2.

```
xtable(contr.set2, caption = "Normalized coding of contrast set 2.")
```

Now we merge in the contrasts to the data frame:

	c6	c7	c8	c9	c10
a	0.75	0.00	0.15	-0.64	-0.76
b	-0.38	0.50	-0.77	0.35	-0.50
c	-0.38	-0.50	-0.56	-0.66	0.48
d	-0.75	0.00	0.94	-0.01	-0.24
e	0.38	-0.50	0.02	0.98	0.02
f	0.38	0.50	0.23	-0.02	1.00

Table 18: Normalized coding of contrast set 2.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1 ← as.data.frame(contr.set1)
contr.set1$condition ← conditions
contr.set2 ← as.data.frame(contr.set2)
contr.set2$condition ← conditions
d ← merge(d, contr.set1, by = c("condition"))
d ← merge(d, contr.set2, by = c("condition"))

```

```

bb.spr.data ← d
write.table(bb.spr.data, file = "bb-spr-analysis-input.txt", sep =
  "\t", quote = FALSE,
  row.names = FALSE, col.names = TRUE)

```

11 Frequency and length processing

```

gg.eye.data ← read.table("gg-eye06-standard-noass.txt", header = TRUE)
data.spr ← read.table("gg-spr06-data.txt")
colnames(data.spr) ← c("subject", "experiment", "item", "condition",
  "roi", "word", "RT",
  "embedding", "intervention")
data.spr$roi ← data.spr$roi + 1
words ← data.frame(item = data.spr$item, condition =
  as.character(data.spr$condition),
  roi = data.spr$roi, word = data.spr$word)
words ← unique(words)
words$word ← tolower(as.character(words$word))
words$word ← as.character(sub("\\.", "", words$word))

```

```

gg.eye.data ← merge(gg.eye.data, words, all.x = TRUE)

```

```

lex ← read.table("anc-lexicon.txt", quote = "", skip = 1)
colnames(lex) ← c("word", "freq")
lex$word ← tolower(as.character(lex$word))
gg.eye.data ← merge(gg.eye.data, lex, by = c("word"), all.x = TRUE)
gg.eye.data$len ← nchar(as.character(gg.eye.data$word))

```

```
gg.eye.data$correct ← gg.eye.data$qacc
```

1

```
data ← subset(gg.eye.data, (item != 25 & item != 2))
```

1

```
data ← data[data$subject != 11, ]
```

1

```
data ← data[data$subject != 29, ]
```

2

```
data ← data[data$subject != 30, ]
```

3

```
data ← data[data$subject != 33, ]
```

4

```
data$subject ← factor(data$subject)
```

1

```
data$trial ← ordered(data$trial)
```

2

```
data$item ← factor(data$item)
```

3

```
data$freq ← log(data$freq)
```

4

```
data$FFDX ← ifelse(data$FFD == data$FPRT | data$FFD == 0, NA, data$FFD)
```

1

11.1 Spill-over preprocessing

Compute spillover information.

```
matrix ← subset(data, select = c(subject, condition, item, trial, word,
  roi, LPRT, len,
  freq, correct))
```

1

```
matrix$roi ← matrix$roi + 1
```

2

```
matrix$prev.LPRT ← matrix$LPRT
```

3

```
matrix$prev.len ← matrix$len
```

4

```
matrix$prev.freq ← matrix$freq
```

5

```
matrix$LPRT ← NULL
```

6

```
matrix$len ← NULL
```

7

```
matrix$freq ← NULL
```

8

```
data ← merge(data, matrix, by = c("subject", "item", "condition",
  "trial", "roi", "correct"),
```

9

```
  all.x = TRUE)
```

10

```
data$word ← data$word.x
```

11

```
data$word.x ← NULL
```

12

```
data$word.y ← NULL
```

13

14

```
mean.prev.len ← mean(data$prev.len, na.rm ← TRUE, trim = 0)
```

1

```
mean.prev.freq ← mean(data$prev.freq, na.rm ← TRUE, trim = 0)
```

2

```
data$prev.cfreq ← data$prev.freq - mean.prev.freq
```

3

```
data$prev.clen ← (1/data$prev.len) - mean.prev.len
```

4

```
write.table(data, file = "gg-spillover-data.txt", sep = "\t", quote =
  FALSE, row.names = FALSE,
  col.names = TRUE)
```

```
data ← read.table(file = "gg-spillover-data.txt", header = TRUE)
data$subject ← factor(data$subject)
data$trial ← ordered(data$trial)
data$item ← factor(data$item)
```

```
d ← data
```

```
mean.freq ← mean(d$freq, na.rm = TRUE)
mean.len ← mean(d$len, na.rm = TRUE)
d$cfreq ← d$freq - mean.freq
d$clen ← (1/d$len) - mean.len
```

```
d$regressed ← !(d$RPD == d$RBRT)
d$ARPD ← ifelse(d$regressed, d$RPD - d$RBRT, NA)
write.table(d, file = "d-mat.txt", sep = "\t", quote = FALSE, row.names
  = FALSE, col.names = TRUE)
```

```
d.rs ← melt(d, id = c("subject", "condition", "item", "roi", "word",
  "cfreq", "clen",
  "prev.LPRT", "prev.clen", "prev.cfreq"), measure = c("FFD", "FFDX",
  "FFP", "SFD", "FPRT",
  "RBRT", "RRT", "RPD", "ARPD", "LPRT", "TFT"), variable_name =
  "times", na.rm = FALSE)
d.rs ← subset(d.rs, value > 0)
```

```
gg.eye.data ← d.rs
```

```
standard ← c("SFD", "FFD", "FPRT", "TFT")
accumulator ← c()
for (m in standard) {
  d.cast ← cast(gg.eye.data, condition + roi ~ ., 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[accumulator$roi < 12, ]
d.cast$measure ← factor(d.cast$measure)
```

```

plot.order = rep(c(1, 5, 11, NA, NA, NA, NA, NA, NA, NA, NA, 1, 5, 8,
  9, 10, 11, NA, NA,
  NA, NA, NA, 1, 5, 6, 7, 8, 9, 10, 11, NA, NA, NA, 1, 2, 3, 4, 5,
  11, NA, NA, NA, NA,
  NA, 1, 2, 3, 4, 5, 8, 9, 10, 11, NA, NA, 1:11), 4)
plot.data <- cbind(d.cast, plot.order)
plot.data <- plot.data[!(plot.data$plot.order == "NA"), ]
plot.data <- na.delete(plot.data)
dummy <- data.frame(region = c(1:11), m = rep(-999, 11))
subtitle <- "Error bars are one standard error."

```

```

color1 <- gray(0.6)
color2 <- gray(0.3)
color3 <- "black"
lty1 <- "dashed"
lty2 <- "solid"

```

```

pdf(file = "gg-eye06-FFD-all-roi.pdf", width = 10, height = 10)
rt.mean <- plot.data[plot.data$measure == "FFD", ]
rtlimits <- c(0, 400)
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
  cex.lab = 1.5, cex.main = 1.5,
  cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
  "Experiment 2 first-fixation durations",
  sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), line = 1, cex = 0.7, c("The", "(admini-",
  "who", "the)",
  "nurse", "(who", "was", "(from", "the", "clinic)", "supervised"))
mtext(side = 1, at = c(1:11), line = 2, cex = 0.7, c("", "strator", "",
  "", "", "", "",
  "", "", "", ""))
legend(x = rt.mean$plot.order[1], y = 400, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$ci.lower[rt.mean$plot.order == 11],
  upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]

```

```

point.type ← c(2)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "c"
dom1 ← 5:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type ← c(1)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "d"
dom1 ← 1:5
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type ← c(15)
color ← color1
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "e"
dom1 ← 1:5
dom2 ← 8:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(17)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
        condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
dev.off()

```

```
pdf
2
```

```
pdf(file = "gg-eye06-SFD-all-roi.pdf", width = 10, height = 10)
rt.mean <- plot.data[plot.data$measure == "SFD", ]
rtlimits <- c(0, 400)
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
     cex.lab = 1.5, cex.main = 1.5,
     cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
       "Experiment 2 single fixation durations",
     sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), line = 1, cex = 0.7, c("The", "(admini-",
  "who", "the)",
  "nurse", "(who", "was", "(from", "the", "clinic))", "supervised"))
mtext(side = 1, at = c(1:11), line = 2, c("", "strator", "",
  "", "", "", "",
  "", "", "", ""))
legend(x = rt.mean$plot.order[1], y = 400, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$ci.lower[rt.mean$plot.order == 11],
  upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(2)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
```

```

    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "d"
dom1 <- 1:5
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type <- c(15)
color <- color1
points(rt.mean$plot.order[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "e"
dom1 <- 1:5
dom2 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(17)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "f"
dom1 <- 1:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order]
point.type <- c(16)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
      rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
dev.off()

```

```

pdf
  2

```

```

pdf(file = "gg-eye06-FPRT-all-roi.pdf", width = 10, height = 10)
rt.mean <- plot.data[plot.data$measure == "FPRT", ]
rtlimits <- c(0, 400)
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
     cex.lab = 1.5, cex.main = 1.5,
     cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
       "Experiment 2 first-pass reading times",
     sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)

```

```

mtext(side = 1, at = c(1:11), line = 1, cex = 0.7, c("The", "(admini-", 8
      "who", "the)",
      "nurse", "(who", "was", "(from", "the", "clinic))", "supervised")) 9
mtext(side = 1, at = c(1:11), line = 2, cex = 0.7, c("", "strator", "", 10
      "", "", "", "",
      "", "", "", "")) 11
legend(x = rt.mean$plot.order[1], y = 400, legend = c("condition A", 12
      "condition B", "condition C",
      "condition D", "condition E", "condition F"), lwd = c(2), lty = 13
      c(lty1, lty1, lty1,
      lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2, 14
      color3), pch = c(5,
      2, 1, 15, 17, 16), bty = "n", cex = 0.9) 15
plot.intervals(xrange = rep(11, 6), lower = 16
      rt.mean$ci.lower[rt.mean$plot.order == 11],
      upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05, 17
      lwd = 1, col = "black",
      lty = "solid") 18
condition <- "a" 19
point.type <- c(5) 20
points(rt.mean$plot.order[rt.mean$condition == condition], 21
      rt.mean$M[rt.mean$condition ==
      condition], pch = point.type, cex = 1.2, col = color1) 22
condition <- "b" 23
dom1 <- 8:11 24
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 25
      > 7]
point.type <- c(2) 26
color <- color2 27
points(rt.mean$plot.order[rt.mean$condition == condition], 28
      rt.mean$M[rt.mean$condition ==
      condition], pch = point.type, col = color, cex = 1.2) 29
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type, 30
      lwd = 1, type = "o")
condition <- "c" 31
dom1 <- 5:11 32
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 33
      > 4]
point.type <- c(1) 34
color <- color3 35
points(rt.mean$plot.order[rt.mean$condition == condition], 36
      rt.mean$M[rt.mean$condition ==
      condition], pch = point.type, col = color, cex = 1.2) 37
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type, 38
      lwd = 1, type = "o")
condition <- "d" 39
dom1 <- 1:5 40
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 41
      < 6]
point.type <- c(15) 42
color <- color1 43
points(rt.mean$plot.order[rt.mean$condition == condition], 44
      rt.mean$M[rt.mean$condition ==
      condition], pch = point.type, col = color, cex = 1.2) 45
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type, 46
      lwd = 1, type = "o")

```

```

condition ← "e"
dom1 ← 1:5
dom2 ← 8:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(17)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
dev.off()

```

pdf
2

```

pdf(file = "gg-eye06-TFT-all-roi.pdf", width = 10, height = 10)
rt.mean ← plot.data[plot.data$measure == "TFT", ]
rtlimits ← c(0, 1200)
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
  cex.lab = 1.5, cex.main = 1.5,
  cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
    "Experiment 2 total fixation times",
  sub = subtitle, ylim = rtlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), line = 1, cex = 0.7, c("The", "(admini-",
  "who", "the)",
  "nurse", "(who", "was", "(from", "the", "clinic)", "supervised"))
mtext(side = 1, at = c(1:11), line = 2, cex = 0.7, c("", "strator", "",
  "", "", "", "",
  "", "", "", ""))
legend(x = rt.mean$plot.order[1], y = 1200, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)

```

```

plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$ci.lower[rt.mean$plot.order == 11],
  upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(2)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "d"
dom1 <- 1:5
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type <- c(15)
color <- color1
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = "solid", col = color, pch =
  point.type, lwd = 1, type = "o")
condition <- "e"
dom1 <- 1:5
dom2 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(17)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)

```

```

lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type, 56
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type, 57
      lwd = 1, type = "o")
condition <- "f" 58
dom1 <- 1:11 59
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order] 60
point.type <- c(16) 61
color <- color3 62
points(rt.mean$plot.order[rt.mean$condition == condition], 63
       rt.mean$M[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2) 64
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type, 65
      lwd = 1, type = "o")
dev.off() 66

```

```

pdf 1
2 2

```

```

d <- gg.eye.data 1

```

```

d$criticalv1 <- ifelse((d$condition == "a" & d$roi == 3) | (d$condition 1
== "b" & d$roi ==
6) | (d$condition == "c" & d$roi == 8) | (d$condition == "d" & 2
d$roi == 6) | (d$condition ==
"e" & d$roi == 9) | (d$condition == "f" & d$roi == 11), "yes", "no") 3
d <- subset(d, criticalv1 == "yes") 4
d$criticalv1 <- factor(d$criticalv1) 5
full.table.v1 <- print(length(d$value)) 6

```

```

[1] 9473 1

```

11.2 Set up contrasts

It's often nice to be able to interpret the contrast coefficients of contrasts as if they represent the estimated mean difference between the two groups represented by the contrast. In order to ensure this, the contrasts must be normalized as follows (e.g. see p. B-13 in Maxwell & Delaney second edition). This new function will also check to be sure that the contrasts are orthogonal, and they are in fact contrasts.

```

normalize.and.check.contrasts <- function(contr) { 1
  for (col in 1:ncol(contr)) { 2
    pos.score <- mean(contr[contr[, col] > 0, col]) 3
    neg.score <- mean(contr[contr[, col] < 0, col]) 4
    diff <- pos.score - neg.score 5
    contr[, col] <- contr[, col]/diff 6
  } 7

```

```

    if (!isTRUE(all.equal(mean(apply(contr, MARGIN = 2, FUN = mean)),
      0)))
      message("WARNING!!! Contrast matrix contains non-contrasts!!")
    m ← t(contr) %*% contr
    should.be.diag ← m - diag(diag(m))
    check.diag ← all(sapply(as.vector(should.be.diag), FUN =
      function(x) {
        return(all.equal(x, 0))
      }))
    if (!check.diag)
      message("WARNING!!! Contrast matrix is non-orthogonal.")
    return(contr)
  }

```

Now set up and normalize the first contrast set.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1.orig ← cbind(c1 = c(-1, -1, -1, 1, 1, 1), c2 = c(-2, 1, 1,
  0, 0, 0), c3 = c(0,
  -1, 1, 0, 0, 0), c4 = c(0, 0, 0, -2, 1, 1), c5 = c(0, 0, 0, 0, -1,
  1))
rownames(contr.set1.orig) ← conditions
contr.set1 ← normalize.and.check.contrasts(contr.set1.orig)

```

Let's take a look at the codings:

```

xtable(contr.set1.orig, caption = "Original coding of contrast set 1.")

```

	c1	c2	c3	c4	c5
a	-1.00	-2.00	0.00	0.00	0.00
b	-1.00	1.00	-1.00	0.00	0.00
c	-1.00	1.00	1.00	0.00	0.00
d	1.00	0.00	0.00	-2.00	0.00
e	1.00	0.00	0.00	1.00	-1.00
f	1.00	0.00	0.00	1.00	1.00

Table 19: Original coding of contrast set 1.

```

xtable(contr.set1, caption = "Normalized coding of contrast set 1.")

```

For the second contrast set we only have two contrasts of interest and use the R contrast function to find orthogonal contrasts to fill out the set.

Contrast c6 is interaction of embedding and locality. Contrast c7 is interaction of embedding and modification type Contrasts 8-10 are not theoretically interesting; weights are assigned by R.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))

```

	c1	c2	c3	c4	c5
a	-0.50	-0.67	0.00	0.00	0.00
b	-0.50	0.33	-0.50	0.00	0.00
c	-0.50	0.33	0.50	0.00	0.00
d	0.50	0.00	0.00	-0.67	0.00
e	0.50	0.00	0.00	0.33	-0.50
f	0.50	0.00	0.00	0.33	0.50

Table 20: Normalized coding of contrast set 1.

```

contr.set2.orig ← cbind(c6 = c(2, -1, -1, -2, 1, 1), c7 = c(0, 1, -1,
0, -1, 1))
contrasts(conditions, how.many = 5) ← contr.set2.orig
contr.set2.orig ← contrasts(conditions)
colnames(contr.set2.orig) ← c("c6", "c7", "c8", "c9", "c10")
contr.set2 ← normalize.and.check.contrasts(contr.set2.orig)

```

Let's take a look at the codings:

```

xtable(contr.set2.orig, caption = "Original coding of contrast set 2.")

```

	c6	c7	c8	c9	c10
a	2.00	0.00	0.11	-0.46	-0.52
b	-1.00	1.00	-0.56	0.25	-0.34
c	-1.00	-1.00	-0.41	-0.47	0.33
d	-2.00	0.00	0.69	-0.01	-0.17
e	1.00	-1.00	0.01	0.71	0.01
f	1.00	1.00	0.17	-0.02	0.69

Table 21: Original coding of contrast set 2.

```

xtable(contr.set2, caption = "Normalized coding of contrast set 2.")

```

	c6	c7	c8	c9	c10
a	0.75	0.00	0.15	-0.64	-0.76
b	-0.38	0.50	-0.77	0.35	-0.50
c	-0.38	-0.50	-0.56	-0.66	0.48
d	-0.75	0.00	0.94	-0.01	-0.24
e	0.38	-0.50	0.02	0.98	0.02
f	0.38	0.50	0.23	-0.02	1.00

Table 22: Normalized coding of contrast set 2.

Now we merge in the contrasts to the data frame:

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))

```

```

contr.set1 ← as.data.frame(contr.set1)
contr.set1$condition ← conditions
contr.set2 ← as.data.frame(contr.set2)
contr.set2$condition ← conditions
d ← merge(d, contr.set1, by = c("condition"))
d ← merge(d, contr.set2, by = c("condition"))

```

```

standard ← c("SFD", "FFD", "FFDX", "FPRT", "RBRT", "RPD", "RRT", "TFT",
             "LPRT", "ARPD")
accumulator ← c()
for (m in standard) {
  d.cast ← cast(d, 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)

```

```

gg.d.cast ← d.cast
gg.eye ← d

```

```

write.table(gg.eye, file = "gg-eye-analysis-input.txt", sep = "\t",
  quote = FALSE, row.names = FALSE,
  col.names = TRUE)

```

12 Frequency and length processing

```

bb.data ← read.table("bb-eye06-standard-noass.txt", header = TRUE)
bb.data$experiment ← bb.data$expt
data.spr ← read.table("bb-spr06-data.txt")
colnames(data.spr) ← c("subject", "experiment", "item", "condition",
  "roi", "word", "dummy",
  "RT")
data.spr ← data.spr[data.spr$experiment == "E1", ]
data.spr$roi ← as.integer(as.character(data.spr$roi)) + 1
words ← data.frame(item = data.spr$item, condition =
  as.character(data.spr$condition),
  roi = data.spr$roi, word = data.spr$word)
words ← unique(words)
words$word ← tolower(as.character(words$word))
words$word ← as.character(sub("\\.", "", words$word))

```

```
exp.items$len ← nchar(as.character(exp.items$word))
```

1

```
exp.items$word ← gsub("saved", "save", exp.items$word, fixed = TRUE)
exp.items$word ← gsub("traded", "trade", exp.items$word, fixed = TRUE)
exp.items$word ← gsub("typed", "type", exp.items$word, fixed = TRUE)
exp.items$word ← gsub("parked", "park", exp.items$word, fixed = TRUE)
exp.items$word ← gsub("walked", "walk", exp.items$word, fixed = TRUE)
exp.items$word ← gsub("hated", "hate", exp.items$word, fixed = TRUE)
```

1

2

3

4

5

6

```
bb.data ← merge(bb.data, words, all.x = TRUE)
```

1

```
lex ← read.table("anc-lexicon.txt", quote = "", skip = 1)
colnames(lex) ← c("word", "freq")
lex$word ← tolower(as.character(lex$word))
bb.data ← merge(bb.data, lex, by = c("word"), all.x = TRUE)
bb.data$len ← nchar(as.character(bb.data$word))
```

1

2

3

4

5

Translate frequency to words-per-million

13 Merge accuracy

```
fixreport ← read.table("bb-eye06-fixreportall.txt", header = TRUE, sep
= "\t")
fixreport ← fixreport[fixreport$experiment == "E1", ]
all.accuracy ← subset(fixreport, select = c(RECORDING_SESSION_LABEL,
itemnumber, questionACC))
accuracy ← unique(all.accuracy)
colnames(accuracy) ← c("subject", "item", "correct")
bb.data ← merge(bb.data, accuracy, by = c("subject", "item"), all.x =
TRUE)
```

1

2

3

4

5

6

Remove mistake in stimuli, item 9 had contextually inappropriate verb; 13 was missing critical verb; 36 had low frequency verb.

```
bb.data ← subset(bb.data, (item != 9 & item != 13 & item != 36))
```

1

```
data ← bb.data
```

1

```
data ← data[data$subject != "s702", ]
```

1

```

data$subject ← factor(data$subject)
data$trial ← ordered(data$trial)
data$item ← factor(data$item)
data$freq ← log(data$freq)

```

1
2
3
4

```

data$FFDX ← ifelse(data$FFD == data$FPRT, NA, data$FFD)

```

1

13.1 Spill-over preprocessing

Compute spillover information.

```

matrix ← subset(data, select = c(subject, condition, item, trial, word,
    roi, LPRT, len,
    freq, correct))
matrix$roi ← matrix$roi + 1
matrix$prev.LPRT ← matrix$LPRT
matrix$prev.len ← matrix$len
matrix$prev.freq ← matrix$freq
matrix$LPRT ← NULL
matrix$len ← NULL
matrix$freq ← NULL
data ← merge(data, matrix, by = c("subject", "item", "condition",
    "trial", "roi", "correct"),
    all.x = TRUE)
data$word ← data$word.x
data$word.x ← NULL
data$word.y ← NULL

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14

```

mean.prev.len ← mean(data$prev.len, na.rm = TRUE, trim = 0)
mean.prev.freq ← mean(data$prev.freq, na.rm = TRUE, trim = 0)
data$prev.cfreq ← data$prev.freq - mean.prev.freq
data$prev.clen ← (1/data$prev.len) - mean.prev.len

```

1
2
3
4

```

write.table(data, file = "bb-spillover-data.txt", sep = "\t", quote =
    FALSE, row.names = FALSE,
    col.names = TRUE)

```

1
2

```

data ← read.table(file = "bb-spillover-data.txt", header = TRUE)
data$subject ← factor(data$subject)
data$trial ← ordered(data$trial)
data$item ← factor(data$item)

```

1
2
3
4

```

d ← data

```

1

```

mean.freq ← mean(d$freq, na.rm = TRUE)
mean.len ← mean(d$len, na.rm = TRUE)
d$cfreq ← d$freq - mean.freq
d$clen ← (1/d$len) - mean.len

```

1
2
3
4

```

d$regressed ← !(d$RPD == d$RBRT)
d$ARPD ← ifelse(d$regressed, d$RPD - d$RBRT, NA)
write.table(d, file = "d-mat.txt", sep = "\t", quote = FALSE, row.names
  = FALSE, col.names = TRUE)

```

1
2
3

```

d.rs ← melt(d, id = c("subject", "condition", "item", "roi", "word",
  "cfreq", "clen",
  "prev.LPRT", "prev.clen", "prev.cfreq"), measure = c("FFD", "FFDX",
  "FFP", "SFD", "FPRT",
  "RBRT", "RRT", "RPD", "ARPD", "LPRT", "TFT"), variable_name =
  "times", na.rm = FALSE)
d.rs ← subset(d.rs, value > 0)

```

1
2
3
4

```

bb.eye.data ← d.rs

```

1

```

standard ← c("SFD", "FFD", "FPRT", "TFT")
accumulator ← c()
for (m in standard) {
  d.cast ← cast(d.rs, condition + roi ~ ., 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[accumulator$roi < 12, ]
d.cast$measure ← factor(d.cast$measure)
plot.order = rep(c(1, 5, 11, NA, NA, NA, NA, NA, NA, NA, 1, 5, 8, 9,
  10, 11, NA, NA, NA,
  NA, NA, 1, 5, 6, 7, 8, 9, 10, 11, NA, NA, NA, 1, 2, 3, 4, 5, 11,
  NA, NA, NA, NA, NA,
  1, 2, 3, 4, 5, 8, 9, 10, 11, NA, NA, 1:11), 4)
plot.data ← cbind(d.cast, plot.order)
plot.data ← plot.data[!(plot.data$plot.order == "NA"), ]
plot.data ← na.delete(plot.data)
dummy ← data.frame(region = c(1:11), m = rep(-999, 11))
subttile ← "Error bars are one standard error."
color1 ← gray(0.6)
color2 ← gray(0.3)
color3 ← "black"
lty1 ← "dashed"
lty2 ← "solid"
pdf(file = "bb-eye06-FFD-all-roi.pdf", width = 10, height = 10)
rt.mean ← plot.data[plot.data$measure == "FFD", ]
rtllimits ← c(0, 400)

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2, 27
     cex.lab = 1.5, cex.main = 1.5,
     cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main = 28
       "Experiment 4 first-fixation durations",
     sub = subtitle, ylim = rtlimits) 29
axis(side = 1, at = 1:11, labels = FALSE) 30
mtext(side = 1, at = c(1:11), cex = 0.7, line = 1, c("The", "(sports", 31
  "that", "the)",
  "child", "(who", "was", "(from", "the", "school))", "played")) 32
legend(x = rt.mean$plot.order[1], y = 600, legend = c("condition A", 33
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty = 34
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2, 35
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9) 36
plot.intervals(xrange = rep(11, 6), lower = 37
  rt.mean$ci.lower[rt.mean$plot.order == 11],
  upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05, 38
  lwd = 1, col = "black",
  lty = "solid") 39
condition <- "a" 40
point.type <- c(5) 41
points(rt.mean$plot.order[rt.mean$condition == condition], 42
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1) 43
condition <- "b" 44
dom1 <- 8:11 45
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 46
  > 7]
point.type <- c(2) 47
color <- color2 48
points(rt.mean$plot.order[rt.mean$condition == condition], 49
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2) 50
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type, 51
  lwd = 1, type = "o")
condition <- "c" 52
dom1 <- 5:11 53
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 54
  > 4]
point.type <- c(1) 55
color <- color3 56
points(rt.mean$plot.order[rt.mean$condition == condition], 57
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2) 58
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type, 59
  lwd = 1, type = "o")
condition <- "d" 60
dom1 <- 1:5 61
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 62
  < 6]
point.type <- c(15) 63
color <- color1 64
points(rt.mean$plot.order[rt.mean$condition == condition], 65
  rt.mean$M[rt.mean$condition ==

```

```

    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "e"
dom1 ← 1:5
dom2 ← 8:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(17)
color ← color2
points(rt.mean$plot.order[rt.mean$condition ==
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
dev.off()

```

pdf
2

```

pdf(file = "bb-eye06-SFD-all-roi.pdf", width = 10, height = 10)
rt.mean ← plot.data[plot.data$measure == "SFD", ]
rtlimits ← c(0, 400)
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2,
     cex.lab = 1.5, cex.main = 1.5,
     cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
       "Experiment 4 single fixation durations",
     sub = subtitle, ylim = rlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), cex = 0.7, line = 1, c("The", "(sports",
  "that", "the)",
  "child", "(who", "was", "(from", "the", "school))", "played"))
legend(x = rt.mean$plot.order[1], y = 400, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)

```

```

plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$ci.lower[rt.mean$plot.order == 11],
  upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(2)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "d"
dom1 <- 1:5
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type <- c(15)
color <- color1
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "e"
dom1 <- 1:5
dom2 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(17)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)

```

```

lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type, 54
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type, 55
      lwd = 1, type = "o")
condition <- "f" 56
dom1 <- 1:11 57
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order] 58
point.type <- c(16) 59
color <- color3 60
points(rt.mean$plot.order[rt.mean$condition == condition], 61
       rt.mean$M[rt.mean$condition ==
                 condition], pch = point.type, col = color, cex = 1.2) 62
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type, 63
      lwd = 1, type = "o")
dev.off() 64

```

```

pdf 1
2 2

```

```

pdf(file = "bb-eye06-FPRT-all-roi.pdf", width = 10, height = 10) 1
rt.mean <- plot.data[plot.data$measure == "FPRT", ] 2
rtlimits <- c(0, 400) 3
plot(m ~ region, data = dummy, xaxt = "n", type = "n", cex.axis = 1.2, 4
     cex.lab = 1.5, cex.main = 1.5,
     cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main = 5
     "Experiment 4 first-pass reading times",
     sub = subtitle, ylim = rlimits) 6
axis(side = 1, at = 1:11, labels = FALSE) 7
mtext(side = 1, at = c(1:11), cex = 0.7, line = 1, c("The", "(sports", 8
            "that", "the)",
            "child", "(who", "was", "(from", "the", "school)", "played)") 9
legend(x = rt.mean$plot.order[1], y = 400, legend = c("condition A", 10
            "condition B", "condition C",
            "condition D", "condition E", "condition F"), lwd = c(2), lty = 11
            c(lty1, lty1, lty1,
            lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2, 12
            color3), pch = c(5,
            2, 1, 15, 17, 16), bty = "n", cex = 0.9) 13
plot.intervals(xrange = rep(11, 6), lower = 14
              rt.mean$ci.lower[rt.mean$plot.order == 11],
              upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05, 15
              lwd = 1, col = "black",
              lty = "solid") 16
condition <- "a" 17
point.type <- c(5) 18
points(rt.mean$plot.order[rt.mean$condition == condition], 19
       rt.mean$M[rt.mean$condition ==
                 condition], pch = point.type, cex = 1.2, col = color1) 20
condition <- "b" 21
dom1 <- 8:11 22
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order 23
                  > 7]
point.type <- c(2) 24
color <- color2 25

```

```

points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$M[rt.mean$condition ==
condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
> 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$M[rt.mean$condition ==
condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "d"
dom1 <- 1:5
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
< 6]
point.type <- c(15)
color <- color1
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$M[rt.mean$condition ==
condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "e"
dom1 <- 1:5
dom2 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
< 6]
range2 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
> 7]
point.type <- c(17)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$M[rt.mean$condition ==
condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
condition <- "f"
dom1 <- 1:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order]
point.type <- c(16)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
       rt.mean$M[rt.mean$condition ==
condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
      lwd = 1, type = "o")
dev.off()

```

pdf
2

```
pdf(file = "bb-eye06-TFT-all-roi.pdf", width = 10, height = 10)
rt.mean <- plot.data[plot.data$measure == "TFT", ]
rtlimits <- c(0, 600)
plot(m ~ region, data = dummy, xaxt = "n", yaxt = "n", cex.axis = 1.2,
     cex.lab = 1.5, cex.main = 1.5,
     cex.sub = 0.7, xlab = "Region", ylab = "Reading time (ms)", main =
       "Experiment 4 total fixation times",
     sub = subtitle, ylim = rlimits)
axis(side = 1, at = 1:11, labels = FALSE)
mtext(side = 1, at = c(1:11), cex = 0.7, line = 1, c("The", "(sports",
  "that", "the)",
  "child", "(who", "was", "(from", "the", "school))", "played"))
legend(x = rt.mean$plot.order[1], y = 600, legend = c("condition A",
  "condition B", "condition C",
  "condition D", "condition E", "condition F"), lwd = c(2), lty =
  c(lty1, lty1, lty1,
  lty2, lty2, lty2), col = c(color1, color2, color3, color1, color2,
  color3), pch = c(5,
  2, 1, 15, 17, 16), bty = "n", cex = 0.9)
plot.intervals(xrange = rep(11, 6), lower =
  rt.mean$ci.lower[rt.mean$plot.order == 11],
  upper = rt.mean$ci.upper[rt.mean$plot.order == 11], width = 0.05,
  lwd = 1, col = "black",
  lty = "solid")
condition <- "a"
point.type <- c(5)
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, cex = 1.2, col = color1)
condition <- "b"
dom1 <- 8:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type <- c(2)
color <- color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "c"
dom1 <- 5:11
range1 <- rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 4]
point.type <- c(1)
color <- color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
  condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty1, col = color, pch = point.type,
  lwd = 1, type = "o")
condition <- "d"
```

```

dom1 ← 1:5
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
point.type ← c(15)
color ← color1
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = "solid", col = color, pch =
  point.type, lwd = 1, type = "o")
condition ← "e"
dom1 ← 1:5
dom2 ← 8:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  < 6]
range2 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order
  > 7]
point.type ← c(17)
color ← color2
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
lines(x = dom2, y = range2, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
condition ← "f"
dom1 ← 1:11
range1 ← rt.mean$M[rt.mean$condition == condition & rt.mean$plot.order]
point.type ← c(16)
color ← color3
points(rt.mean$plot.order[rt.mean$condition == condition],
  rt.mean$M[rt.mean$condition ==
    condition], pch = point.type, col = color, cex = 1.2)
lines(x = dom1, y = range1, lty = lty2, col = color, pch = point.type,
  lwd = 1, type = "o")
dev.off()

```

```

pdf
2

```

```
d ← bb.eye.data
```

```

d$criticalv1 ← ifelse((d$condition == "a" & d$roi == 3) | (d$condition
  == "b" & d$roi ==
    6) | (d$condition == "c" & d$roi == 8) | (d$condition == "d" &
      d$roi == 6) | (d$condition ==
        "e" & d$roi == 9) | (d$condition == "f" & d$roi == 11), "yes", "no")
d ← subset(d, criticalv1 == "yes")
d$criticalv1 ← factor(d$criticalv1)
full.table.v1 ← print(length(d$value))

```

```
[1] 9794
```

13.2 Set up contrasts

It's often nice to be able to interpret the contrast coefficients of contrasts as if they represent the estimated mean difference between the two groups represented by the contrast. In order to ensure this, the contrasts must be normalized as follows (e.g. see p. B-13 in Maxwell & Delaney second edition). This new function will also check to be sure that the contrasts are orthogonal, and they are in fact contrasts.

```
normalize.and.check.contrasts ← function(contr) {
  for (col in 1:ncol(contr)) {
    pos.score ← mean(contr[contr[, col] > 0, col])
    neg.score ← mean(contr[contr[, col] < 0, col])
    diff ← pos.score - neg.score
    contr[, col] ← contr[, col]/diff
  }
  if (!isTRUE(all.equal(mean(apply(contr, MARGIN = 2, FUN = mean)),
    0)))
    message("WARNING!!! Contrast matrix contains non-contrasts!!")
  m ← t(contr) %*% contr
  should.be.diag ← m - diag(diag(m))
  check.diag ← all(sapply(as.vector(should.be.diag), FUN =
    function(x) {
      return(all.equal(x, 0))
    })))
  if (!check.diag)
    message("WARNING!!! Contrast matrix is non-orthogonal.")
  return(contr)
}
```

Now set up and normalize the first contrast set.

```
conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1.orig ← cbind(c1 = c(-1, -1, -1, 1, 1, 1), c2 = c(-2, 1, 1,
  0, 0, 0), c3 = c(0,
  -1, 1, 0, 0, 0), c4 = c(0, 0, 0, -2, 1, 1), c5 = c(0, 0, 0, 0, -1,
  1))
rownames(contr.set1.orig) ← conditions
contr.set1 ← normalize.and.check.contrasts(contr.set1.orig)
```

Let's take a look at the codings:

```
xtable(contr.set1.orig, caption = "Original coding of contrast set 1.")
```

```
xtable(contr.set1, caption = "Normalized coding of contrast set 1.")
```

	c1	c2	c3	c4	c5
a	-1.00	-2.00	0.00	0.00	0.00
b	-1.00	1.00	-1.00	0.00	0.00
c	-1.00	1.00	1.00	0.00	0.00
d	1.00	0.00	0.00	-2.00	0.00
e	1.00	0.00	0.00	1.00	-1.00
f	1.00	0.00	0.00	1.00	1.00

Table 23: Original coding of contrast set 1.

	c1	c2	c3	c4	c5
a	-0.50	-0.67	0.00	0.00	0.00
b	-0.50	0.33	-0.50	0.00	0.00
c	-0.50	0.33	0.50	0.00	0.00
d	0.50	0.00	0.00	-0.67	0.00
e	0.50	0.00	0.00	0.33	-0.50
f	0.50	0.00	0.00	0.33	0.50

Table 24: Normalized coding of contrast set 1.

For the second contrast set we only have two contrasts of interest and use the R contrast function to find orthogonal contrasts to fill out the set.

Contrast c6 is interaction of embedding and locality. Contrast c7 is interaction of embedding and modification type Contrasts 8-10 are not theoretically interesting; weights are assigned by R.

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set2.orig ← cbind(c6 = c(2, -1, -1, -2, 1, 1), c7 = c(0, 1, -1,
0, -1, 1))
contrasts(conditions, how.many = 5) ← contr.set2.orig
contr.set2.orig ← contrasts(conditions)
colnames(contr.set2.orig) ← c("c6", "c7", "c8", "c9", "c10")
contr.set2 ← normalize.and.check.contrasts(contr.set2.orig)
```

Let's take a look at the codings:

```
xtable(contr.set2.orig, caption = "Original coding of contrast set 2.")
```

```
xtable(contr.set2, caption = "Normalized coding of contrast set 2.")
```

Now we merge in the contrasts to the data frame:

```

conditions ← as.factor(c("a", "b", "c", "d", "e", "f"))
contr.set1 ← as.data.frame(contr.set1)
contr.set1$condition ← conditions
contr.set2 ← as.data.frame(contr.set2)
contr.set2$condition ← conditions
```

	c6	c7	c8	c9	c10
a	2.00	0.00	0.11	-0.46	-0.52
b	-1.00	1.00	-0.56	0.25	-0.34
c	-1.00	-1.00	-0.41	-0.47	0.33
d	-2.00	0.00	0.69	-0.01	-0.17
e	1.00	-1.00	0.01	0.71	0.01
f	1.00	1.00	0.17	-0.02	0.69

Table 25: Original coding of contrast set 2.

	c6	c7	c8	c9	c10
a	0.75	0.00	0.15	-0.64	-0.76
b	-0.38	0.50	-0.77	0.35	-0.50
c	-0.38	-0.50	-0.56	-0.66	0.48
d	-0.75	0.00	0.94	-0.01	-0.24
e	0.38	-0.50	0.02	0.98	0.02
f	0.38	0.50	0.23	-0.02	1.00

Table 26: Normalized coding of contrast set 2.

```
d ← merge(d, contr.set1, by = c("condition"))
d ← merge(d, contr.set2, by = c("condition"))
```

6
7

```
standard ← c("SFD", "FFD", "FFDX", "FPRT", "RBRT", "RPD", "RRT", "TFT",
             "LPRT", "ARPD")
accumulator ← c()
for (m in standard) {
  d.cast ← cast(d, 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)
```

1
2
3
4
5
6
7
8
9
10
11
12
13

```
bb.d.cast ← d.cast
bb.eye ← d
```

1
2

```
write.table(d, file = "bb-eye-analysis-input.txt", sep = "\t", quote =
  FALSE, row.names = FALSE,
  col.names = TRUE)
```

1
2

```

rtlimits ← c(0, 1200)
sdlimits ← c(0, 1200)
pdf(file = "main-figure.pdf", width = 10, height = 10)
par(mfrow = c(3, 3))
par(bty = "n", xpd = NA)
par(mgp = c(2.8, 1, 0))
par(cex.sub = 1.1, font.sub = 3)
par(cex.axis = 1.1, cex.lab = 1.1)
par(las = 1)
source("original-gg05-spr.r")
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab = "",
     main = "", sub = "(data from Experiments 1 and 3)",
     ylim = rtlimits, xaxt = "n")
add.axes.and.title("Self-Paced Reading Times")
legend(x = 1.2, y = 1200, legend = c("Grodner & Gibson (2005)
  materials", "Short, high-frequency materials"),
      lty = c("solid", "solid"), col = c("black", gray(0.5)), bty = "n")
linetype ← "solid"
col ← "black"
d.melt ← subset(gg.spr.data, good.subj == TRUE)
source("spr-plots.r")
linetype ← "solid"
col ← gray(0.5)
d.melt ← subset(bb.spr.data, good.subj == TRUE)
source("spr-plots.r")
plot.eye.measure ← function(d.cast, plot.measure, linetype, color =
  "black") {
  plot.data ← subset(d.cast, measure == plot.measure)
  plot.intervals(xrange = 1:6, lower = plot.data$ci.lower, upper =
    plot.data$ci.upper,
    width = 0, lwd = 0.5, col = color, lty = "solid")
  lines(x = plot.data$condition[1:3], y = plot.data$M[1:3], lty =
    linetype, col = color,
    pch = c(20), lwd = 1, type = "o")
  lines(x = plot.data$condition[4:6], y = plot.data$M[4:6], lty =
    linetype, col = color,
    pch = c(20), lwd = 1, type = "o")
}
rtlimits ← c(0, 1200)
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab = "",
     main = "", sub = "(data from Experiments 2 and 4)",
     ylim = rtlimits, xaxt = "n")
add.axes.and.title("Total Fixation Times")
plot.eye.measure(gg.d.cast, "TFT", "solid", "black")
plot.eye.measure(bb.d.cast, "TFT", "solid", gray(0.5))
rtlimits ← c(0, 400)
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab =
  "Milliseconds", main = "",
  sub = "(data from Experiments 2 and 4)", ylim = rtlimits, xaxt =
  "n")
add.axes.and.title("First Fixation Durations")
plot.eye.measure(gg.d.cast, "FFDX", "solid")
plot.eye.measure(bb.d.cast, "FFDX", "solid", gray(0.5))
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab = "",
     main = "", sub = "(data from Experiments 2 and 4)",

```

```

ylim = rtlimits, xaxt = "n")
add.axes.and.title("Single Fixation Durations")
plot.eye.measure(gg.d.cast, "SFD", "solid")
plot.eye.measure(bb.d.cast, "SFD", "solid", gray(0.5))
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab = "",
     main = "", sub = "(data from Experiments 2 and 4)",
     ylim = rtlimits, xaxt = "n")
add.axes.and.title("First Pass Reading Times")
plot.eye.measure(gg.d.cast, "FPRT", "solid")
plot.eye.measure(bb.d.cast, "FPRT", "solid", gray(0.5))
rtlimits <- c(0, 1200)
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab = "",
     main = "", sub = "(data from Experiments 2 and 4)",
     ylim = rtlimits, xaxt = "n")
add.axes.and.title("Rereading Times")
plot.eye.measure(gg.d.cast, "RRT", "solid")
plot.eye.measure(bb.d.cast, "RRT", "solid", gray(0.5))
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab = "",
     main = "", sub = "(data from Experiments 2 and 4)",
     ylim = rtlimits, xaxt = "n")
add.axes.and.title("Regression Path Durations")
plot.eye.measure(gg.d.cast, "RPD", "solid")
plot.eye.measure(bb.d.cast, "RPD", "solid", gray(0.5))
plot(m ~ condition, data = dummy, type = "n", xlab = "", ylab =
     "Milliseconds", main = "",
     sub = "(data from Experiments 2 and 4)", ylim = rtlimits, xaxt =
     "n")
add.axes.and.title("Nonzero Regression Path Durations")
plot.eye.measure(gg.d.cast, "ARPD", "solid")
plot.eye.measure(bb.d.cast, "ARPD", "solid", gray(0.5))
dev.off()

```

```

pdf
2

```

14 HPD plots

Now we set up the plots.

```
source("hpd-plotting-support.r")
```

14.1 HPD plots for locality effects within matrix and embedded conditions

```

hpd <- read.table("all-HPD-intervals-untransformed.txt", header = TRUE)
pdf(file = "hpd-locality-untransformed.pdf", width = 10, height = 10)
par(mfrow = c(3, 3))

```

```

coef.labels ← c(c11 = "local vs. PP", c12 = "local vs. RC", c13 =
  "local vs. PP", c14 = "local vs. RC",
  c3 = "PP vs. RC", c2 = "local vs.\n nonlocal", c5 = "PP vs. RC", c4
    = "local vs.\n nonlocal")
coef.groups ← list(Matrix = c("c2", "c3"), Embedded = c("c4", "c5"))
group.labels ← NULL
group.y.positioning ← 0.25
cex.axis ← 1
cex.main ← 1.6
cex.point ← 0.8
cex.group.label ← 1.2
x.tick ← FALSE
spr.tft.lim ← c(-100, 300)
late.measure.lim ← c(-300, 800)
early.measure.lim ← c(-25, 100)

```

```

plot.new()
legend(x = 0, y = 0.5, legend = c("Grodner & Gibson (2005) materials",
  "Short, high-frequency materials"),
  lty = c("solid", "solid"), col = c("black", gray(0.5)), lwd = 1.5,
  bty = "n")
plot.hpds(hpd.intervals = hpd, measures = c("gg.spr", "bb.spr"),
  coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, cex.main =
    cex.main, x.tick = x.tick,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = spr.tft.lim, ylab =
    "Coefficient estimate: log (ms)",
  title = "Self-paced Reading", measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.TFT",
  "bb.eye.TFT"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, cex.main =
    cex.main, x.tick = x.tick,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = spr.tft.lim, ylab = "", title =
    "Total Fixation Time",
  measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FFDX",
  "bb.eye.FFDX"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = late.measure.lim, ylab =
    "Coefficient estimate: log(ms)",
  title = "First Fixation Duration", measure.col = c("black",
    grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.SFD",
  "bb.eye.SFD"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,

```

```

    group.labels = group.labels, ylim = early.measure.lim, ylab = "",
    title = "Single Fixation Duration",
    measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FPRT",
"bb.eye.FPRT"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label,
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = early.measure.lim, ylab = "",
title = "First-pass Reading Time",
measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RRT",
"bb.eye.RRT"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label,
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab =
"Coefficient estimate: log(ms)",
title = "Re-reading Time", measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RPD",
"bb.eye.RPD"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label,
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = "",
title = "Regression Path Duration",
measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.ARPD",
"bb.eye.ARPD"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label,
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = "",
title = "Nonzero Regression Path Duration",
measure.col = c("black", grey(0.4)))
dev.off()

```

```
pdf
2
```

```

hpd ← read.table("all-HPD-intervals-log-transformed.txt", header = TRUE)
pdf(file = "hpd-locality-log-transformed.pdf", width = 10, height = 10)
par(mfrow = c(3, 3))
spr.tft.lim ← c(-0.2, 0.4)
early.measure.lim ← c(-0.1, 0.4)
late.measure.lim ← c(-0.5, 1)
plot.new()
legend(x = 0, y = 0.5, legend = c("Grodner & Gibson (2005) materials",
"Short, high-frequency materials"),

```

```

    lty = c("solid", "solid"), col = c("black", gray(0.5)), lwd = 1.5,
    bty = "n")
plot.hpds(hpd.intervals = hpd, measures = c("gg.spr", "bb.spr"),
  coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, cex.main =
    cex.main, x.tick = x.tick,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = spr.tft.lim, ylab =
    "Coefficient estimate: log (ms)",
  title = "Self-paced Reading", measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.TFT",
  "bb.eye.TFT"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, cex.main =
    cex.main, x.tick = x.tick,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = spr.tft.lim, ylab = "", title =
    "Total Fixation Time",
  measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FFDX",
  "bb.eye.FFDX"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = late.measure.lim, ylab =
    "Coefficient estimate: log(ms)",
  title = "First Fixation Duration", measure.col = c("black",
    grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.SFD",
  "bb.eye.SFD"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = early.measure.lim, ylab = "",
  title = "Single Fixation Duration",
  measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FPRT",
  "bb.eye.FPRT"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = early.measure.lim, ylab = "",
  title = "First-pass Reading Time",
  measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RRT",
  "bb.eye.RRT"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = late.measure.lim, ylab =
    "Coefficient estimate: log(ms)",

```

```

title = "Re-reading Time", measure.col = c("black", grey(0.4))) 39
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RPD", 40
"bb.eye.RPD"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 41
cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label, 42
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = "", 43
title = "Regression Path Duration",
measure.col = c("black", grey(0.4))) 44
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.ARPD", 45
"bb.eye.ARPD"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 46
cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label, 47
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = "", 48
title = "Nonzero Regression Path Duration",
measure.col = c("black", grey(0.4))) 49
dev.off() 50

```

```

pdf 1
2 2

```

```

hpd ← read.table("all-HPD-intervals-untransformed.txt", header = TRUE) 1
pdf(file = "hpd-embedding-untransformed.pdf", width = 10, height = 10) 2
par(mfrow = c(3, 3)) 3
coef.labels ← c(c1 = "Embedding", c6 = "Embedding\n X Locality", c7 = 4
"Embedding\n X Modification")
coef.groups ← list(g1 = c("c1"), g2 = c("c6", "c7")) 5
group.labels = c(g1 = "Main effect", g2 = "Interactions") 6
spr.tft.lim ← c(-100, 350) 7
late.measure.lim ← c(-300, 600) 8
early.measure.lim ← c(-25, 80) 9
plot.new() 10
legend(x = 0, y = 0.5, legend = c("Grodner & Gibson (2005) materials", 11
"Short, high-frequency materials"),
lty = c("solid", "solid"), col = c("black", gray(0.5)), lwd = 1.5, 12
bty = "n")
plot.hpds(hpd.intervals = hpd, measures = c("gg.spr", "bb.spr"), 13
coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, cex.main = 14
cex.main, x.tick = x.tick,
cex.point = cex.point, cex.group.label = cex.group.label, 15
group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = spr.tft.lim, ylab = 16
"Coefficient estimate: log (ms)",
title = "Self-paced Reading", measure.col = c("black", grey(0.4))) 17
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.TFT", 18
"bb.eye.TFT"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, cex.main = 19
cex.main, x.tick = x.tick,
cex.point = cex.point, cex.group.label = cex.group.label, 20
group.y.positioning = group.y.positioning,

```

```

    group.labels = group.labels, ylim = spr.tft.lim, ylab = "", title =
      "Total Fixation Time",
    measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FFDX",
      "bb.eye.FFDX"), coef.groups = coef.groups,
    coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
      cex.main = cex.main,
    cex.point = cex.point, cex.group.label = cex.group.label,
      group.y.positioning = group.y.positioning,
    group.labels = group.labels, ylim = late.measure.lim, ylab =
      "Coefficient estimate: log(ms)",
    title = "First Fixation Duration", measure.col = c("black",
      grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.SFD",
      "bb.eye.SFD"), coef.groups = coef.groups,
    coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
      cex.main = cex.main,
    cex.point = cex.point, cex.group.label = cex.group.label,
      group.y.positioning = group.y.positioning,
    group.labels = group.labels, ylim = early.measure.lim, ylab = "",
      title = "Single Fixation Duration",
    measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FPRT",
      "bb.eye.FPRT"), coef.groups = coef.groups,
    coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
      cex.main = cex.main,
    cex.point = cex.point, cex.group.label = cex.group.label,
      group.y.positioning = group.y.positioning,
    group.labels = group.labels, ylim = early.measure.lim, ylab = "",
      title = "First-pass Reading Time",
    measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RRT",
      "bb.eye.RRT"), coef.groups = coef.groups,
    coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
      cex.main = cex.main,
    cex.point = cex.point, cex.group.label = cex.group.label,
      group.y.positioning = group.y.positioning,
    group.labels = group.labels, ylim = late.measure.lim, ylab =
      "Coefficient estimate: log(ms)",
    title = "Re-reading Time", measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RPD",
      "bb.eye.RPD"), coef.groups = coef.groups,
    coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
      cex.main = cex.main,
    cex.point = cex.point, cex.group.label = cex.group.label,
      group.y.positioning = group.y.positioning,
    group.labels = group.labels, ylim = late.measure.lim, ylab = "",
      title = "Regression Path Duration",
    measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.ARPD",
      "bb.eye.ARPD"), coef.groups = coef.groups,
    coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
      cex.main = cex.main,
    cex.point = cex.point, cex.group.label = cex.group.label,
      group.y.positioning = group.y.positioning,

```

```

    group.labels = group.labels, ylim = late.measure.lim, ylab = "",
    title = "Nonzero Regression Path Duration",
    measure.col = c("black", grey(0.4)))
dev.off()

```

```

pdf
2

```

14.2 HPD plots for embedding effects

```

hpd ← read.table("all-HPD-intervals-log-transformed.txt", header = TRUE)
pdf(file = "hpd-embedding-log-transformed.pdf", width = 10, height = 10)
par(mfrow = c(3, 3))
spr.tft.lim ← c(-0.2, 0.5)
early.measure.lim ← c(-0.1, 0.5)
late.measure.lim ← c(-0.2, 1)
plot.new()
legend(x = 0, y = 0.5, legend = c("Grodner & Gibson (2005) materials",
  "Short, high-frequency materials"),
  lty = c("solid", "solid"), col = c("black", gray(0.5)), lwd = 1.5,
  bty = "n")
plot.hpds(hpd.intervals = hpd, measures = c("gg.spr", "bb.spr"),
  coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, cex.main =
    cex.main, x.tick = x.tick,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = spr.tft.lim, ylab =
    "Coefficient estimate: log (ms)",
  title = "Self-paced Reading", measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.TFT",
  "bb.eye.TFT"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, cex.main =
    cex.main, x.tick = x.tick,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = spr.tft.lim, ylab = "", title =
    "Total Fixation Time",
  measure.col = c("black", grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FFDX",
  "bb.eye.FFDX"), coef.groups = coef.groups,
  coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick,
  cex.main = cex.main,
  cex.point = cex.point, cex.group.label = cex.group.label,
  group.y.positioning = group.y.positioning,
  group.labels = group.labels, ylim = late.measure.lim, ylab =
    "Coefficient estimate: log(ms)",
  title = "First Fixation Duration", measure.col = c("black",
    grey(0.4)))
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.SFD",
  "bb.eye.SFD"), coef.groups = coef.groups,

```

```

coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 26
  cex.main = cex.main,
coef.point = cex.point, cex.group.label = cex.group.label, 27
  group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = early.measure.lim, ylab = "", 28
  title = "Single Fixation Duration",
measure.col = c("black", grey(0.4))) 29
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.FPRT", 30
  "bb.eye.FPRT"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 31
  cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label, 32
  group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = early.measure.lim, ylab = "", 33
  title = "First-pass Reading Time",
measure.col = c("black", grey(0.4))) 34
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RRT", 35
  "bb.eye.RRT"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 36
  cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label, 37
  group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = 38
  "Coefficient estimate: log(ms)",
title = "Re-reading Time", measure.col = c("black", grey(0.4))) 39
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.RPD", 40
  "bb.eye.RPD"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 41
  cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label, 42
  group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = "", 43
  title = "Regression Path Duration",
measure.col = c("black", grey(0.4))) 44
plot.hpds(hpd.intervals = hpd, measures = c("gg.eye.ARPD", 45
  "bb.eye.ARPD"), coef.groups = coef.groups,
coef.labels = coef.labels, cex.axis = cex.axis, x.tick = x.tick, 46
  cex.main = cex.main,
cex.point = cex.point, cex.group.label = cex.group.label, 47
  group.y.positioning = group.y.positioning,
group.labels = group.labels, ylim = late.measure.lim, ylab = "", 48
  title = "Nonzero Regression Path Duration",
measure.col = c("black", grey(0.4))) 49
dev.off() 50

```

```

pdf 1
  2 2

```

```

options(prompt = "> ", continue = "+ ") 1

```