

Supplementary materials for

Baayen, Vasishth, Kliegl, and Bates (2016), The cave of shadows: addressing the human factor with generalized additive mixed models. *Journal of Memory and Language*.

This document describes how data sets discussed in this paper can be obtained, and presents the R-code for the models and corresponding figures and tables.

Required software: R ($\geq 3.2.2$) and R packages `devtools` (1.9.1 or later), `itsadug` (2.2 or later), `lattice` (0.20-33 or later), `lme4` (1.1-10 or later), `plotfunctions` (1.0 or later), `mgcv` (1.8-12 or later), `Rcurl` (1.95-4.7 or later), and `xtable` (1.7-4 or later).

Contents

1 Preliminaries	2
2 Introduction to the generalized additive mixed model	3
2.1 Figure 1	3
2.2 Figure 5 and the model for the vietnamese dataset	6
3 The KKL dataset	8
3.1 Table 1 (main text) and Table 3 (appendix)	8
3.2 Table 4 (appendix)	12
3.3 Figure 9	14
3.4 Figure 10	17
3.5 Figure 11	18
3.6 Figure 12	19
3.7 Figure 13	21
3.8 Table 2	22
4 The baldey dataset	24
4.1 Model and code for Table 5	24
4.2 Table 6: the model with linear effects only	26
4.3 Figure 14	28
5 The poems dataset	31
5.1 The full GAMM	31
5.2 A decompositional GAMM	32
5.3 A model with linear effects only	33
5.4 Figure 15	35
5.5 Figure 16	38
5.6 Figure 17 and corresponding LMMs	39
5.7 Figure 18	42

1 Preliminaries

Once R is started up, the abovementioned packages must be activated. This accomplished in the following code snippet:

```
require(mgcv)
require(lme4)
require(xtable)
require(itsadug)
require(plotfunctions)
require(devtools)
require(RCurl)
require(RePsychLing)
```

The `RePsychLing` package is currently available on [github](#). It is installed and made available as follows:

```
devtools::install_github("dmbates/RePsychLing", build_vignettes=FALSE)
require(RePsychLing)
```

All data sets analysed in this document are available in the `RePsychLing` package, with the exception of the `baldey` dataset, which is available at http://www.mirjamernestus.nl/Ernestus/Baldey/baldey_data.zip. After downloading and unpacking the zip file in a subfolder named `data`, the database can be loaded into R.

```
baldey = read.table("data/baldey.txt", header=TRUE)
```

One figure requires R code available on [github](#):

```
source(textConnection(getURL(
  "https://gist.github.com/mages/5339689/raw/576263b8f0550125b61f4ddba127f5aa00fa2014/add.alpha.R"))))
```

Generalized additive models for the larger data sets (`baldey`, `poems`) may take up to several hours to fit.

The R code and R output given below make it possible to replicate all results reported in our manuscript, including tables and figures. Technical details on the figures illustrating regression splines in section 2 of the manuscript are not included as they are not part of actual data analysis.

2 Introduction to the generalized additive mixed model

2.1 Figure 1

```
data("KKL")
dfr123 = KKL[KKL$subj==123,]

# set up one row of three plots
par(mfrow=c(1,3), oma=c(2,2,2,2), mar=c(5,5,2,1))

# two regression models with polynomials
poly10.lm = lm(lrt ~ poly(trial,9), data = dfr123)
poly20.lm = lm(lrt ~ poly(trial,19), data = dfr123)

# make left panel
cx=1.8
plot(dfr123$trial, dfr123$lrt,
     pch=19, cex=0.5, cex.lab=cx+0.3, cex.axis=cx,
     xlab="trial", ylab="log RT")
lines(dfr123$trial, predict(poly10.lm),
      col="steelblue2", lwd=3, lty=1)
lines(dfr123$trial, predict(poly20.lm),
      col="indianred", lwd=3, lty=1)
mtext("polynomials", 3, 1.5, cex=cx-0.5)
legend(0.1, 5.4,
      legend = c("degree 9", "degree 19"),
      lty = rep(1, 2), lwd = rep(3,3),
      col = c("steelblue2", "indianred"), bty = "n", cex = cx+0.2)

# regression splines with 10 basis functions
dfr123.gam = bam(lrt ~ s(trial, bs="cr", k=10),
                data = dfr123, method="fREML")
dfr123$GamCR = predict(dfr123.gam)
dfr123.gamTPRS = bam(lrt ~ s(trial, m=2, bs="tp", k=10),
                    data = dfr123, method="fREML")
dfr123$GamTPRS = predict(dfr123.gamTPRS)

# make center panel
plot(dfr123$trial, dfr123$lrt,
     pch=19, cex=0.5, cex.lab=cx+0.3, cex.axis=cx,
     xlab="trial", ylab="log RT")
lines(dfr123$trial, dfr123$GamCR,
      col="steelblue2", lwd=3)
lines(dfr123$trial, dfr123$GamTPRS,
      col="indianred", lwd=3)
legend(0.1, 5.4,
      legend=rev(c("cr", "tp")),
      lty=rep(1, 2), lwd=rep(3,2),
      col=c("indianred", "steelblue2"), bty="n", cex=cx+0.2)
mtext("regression splines with 10 basis functions", 3, 1.5, cex=cx-0.5)

# regression splines with 20 basis functions
```

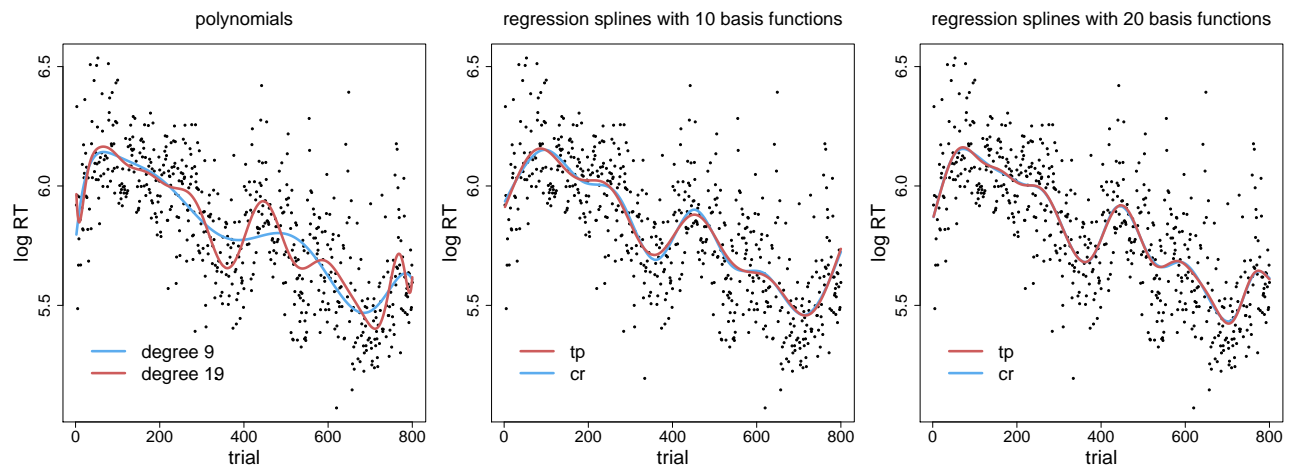
```

dfr123.gam      = bam(lrt ~ s(trial, bs="cr", k=20),
                      data=dfr123, method="fREML")
dfr123$GamCR    = predict(dfr123.gam)
dfr123.gamTPRS = bam(lrt ~ s(trial, m=2, bs="tp", k=20),
                      data=dfr123, method="fREML")
dfr123$GamTPRS  = predict(dfr123.gamTPRS)

# make right panel
plot(dfr123$trial, dfr123$lrt,
     pch=19, cex=0.5, cex.lab=cx+0.3, cex.axis=cx,
     xlab="trial", ylab="log RT")
lines(dfr123$trial, dfr123$GamCR,
      col="steelblue2", lwd=3)
lines(dfr123$trial, dfr123$GamTPRS,
      col="indianred", lwd=3)
mtext("regression splines with 20 basis functions", 3, 1.5, cex=cx-0.5)
legend(0.1, 5.4,
      legend=rev(c("cr", "tp")),
      lty=rep(1, 2), lwd=rep(3, 2),
      col=c("indianred", "steelblue2"), bty="n", cex=cx+0.2)

```

Figure 1



AIC values for the models discussed with respect to Figure 1.

```

# a model with a linear effect of trial
AIC(lm(lrt ~ trial, data=dfr123))

[1] -206.4587

# the models with polynomials
AIC(lm(lrt ~ poly(trial, 9), data=dfr123))

```

```

[1] -274.739

AIC(lm(lrt ~ poly(trial, 19), data=dfr123))

[1] -332.2674

# the models with restricted cubic splines
AIC(bam(lrt ~ s(trial, bs="cr", k=10), data=dfr123, method="fREML"))

[1] -316.748

AIC(bam(lrt ~ s(trial, bs="cr", k=20), data=dfr123, method="fREML"))

[1] -331.5204

# the models with thin plate regression splines
AIC(bam(lrt ~ s(trial, bs="tp", k=10), data=dfr123, method="fREML"))

[1] -311.4005

AIC(bam(lrt ~ s(trial, bs="tp", k=20), data=dfr123, method="fREML"))

[1] -334.0553

```

Comparison of restricted cubic splines with 10 and 20 basis functions.

```

dfr123.10 = bam(lrt ~ s(trial, k=10), data=dfr123, method="fREML")
dfr123.20 = bam(lrt ~ s(trial, k=20), data=dfr123, method="fREML")
anova(dfr123.10, dfr123.20, test="F")

```

Analysis of Deviance Table

Model 1: lrt ~ s(trial, k = 10)

Model 2: lrt ~ s(trial, k = 20)

	Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
1	624.36	21.944				
2	618.58	20.778	5.7799	1.1664	6.0082	5.676e-06

2.2 Figure 5 and the model for the vietnamese dataset

```
data(vietnamese)

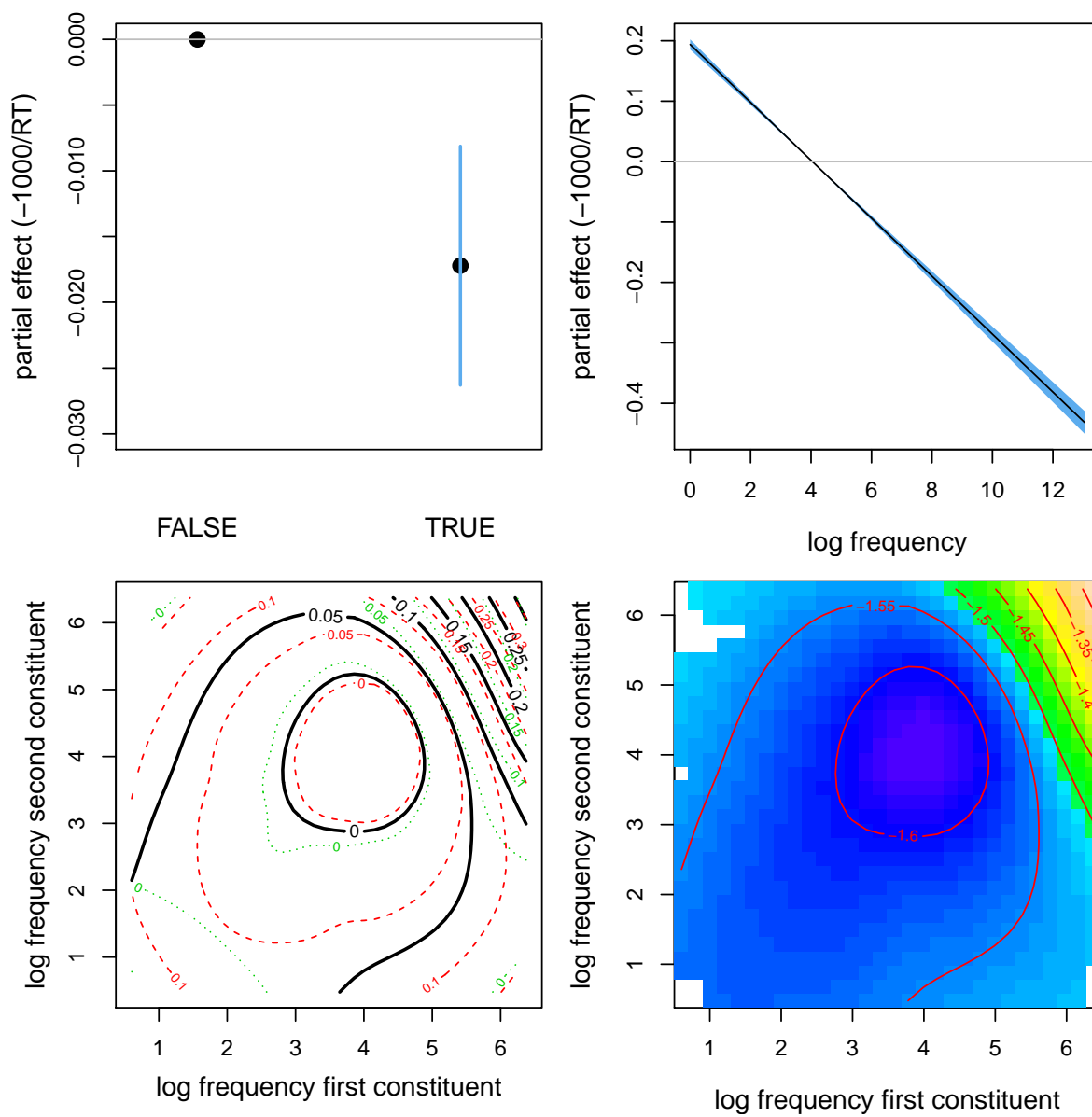
# a GAM for this single-subject data set
vietnamese.gam = bam(RTinv ~ MidLevelTone +
                     s(LogFreq) +
                     te(LogFreqSyl1, LogFreqSyl2),
                     data=vietnamese, method="ML")

# a summary of the parametric part of the model
summary(vietnamese.gam)$p.table
# a summary of the smooths in the model
summary(vietnamese.gam)$s.table

# the corresponding figure
par(mfrow = c(2,2), mar = c(4,4,1,1))
plot(c(0.5,2.5), c(0, -0.017215), xlim=c(0, 3), ylim=c(-0.03, 0), xaxt="n",
     xlab=" ", ylab="partial effect (-1000/RT)", pch=19, cex=1.5)
abline(h=0, col="gray")
lines(c(2.5, 2.5), c(-0.017215+2*0.004544, -0.017215-2*0.004544), lwd=2, col="steelblue2")
mtext(c("FALSE", "TRUE"), side=1, line=2.5, at=c(0.5, 2.5))

plot(vietnamese.gam, select=1, scheme=1, shade.col="steelblue2", rug=FALSE,
     xlab="log frequency", ylab="partial effect (-1000/RT)")
abline(h=0, col="gray")
plot(vietnamese.gam, select=2, rug=FALSE,
     xlab="log frequency first constituent", ylab="log frequency second constituent",
     main=" ")
vis.gam(vietnamese.gam, view=c("LogFreqSyl1", "LogFreqSyl2"),
        color="topo", plot.type="contour", too.far=0.1,
        xlab="log frequency first constituent", ylab="log frequency second constituent",
        main=" ")
```

Figure 5.



3 The KKL dataset

3.1 Table 1 (main text) and Table 3 (appendix)

```
data(KKL)
# identify starting point for time series of RTs
KKL$FirstTrial = KKL$first==1
# scale trial and soa
KKL$Trial = scale(KKL$trial)
KKL$Soa = scale(KKL$SOA)
# add a quadratic term for trial
KKL$TrialQ = KKL$Trial^2
# define interaction of size by orientation
KKL$Int = interaction(KKL$size, KKL$cardinal)
```

A sequence of linear mixed models and pairwise likelihood ratio tests:

```
kk1.lmer0 = lmer(lrt ~ size * (spt + obj + grv) * orn +
  (1|subj)+
  (0+spt|subj)+
  (0+grv|subj)+
  (0+obj|subj)+
  (0+orn|subj)+
  (0+spt_orn|subj)+
  poly(Soa,2,row=TRUE),
  data=KKL, REML=FALSE)

fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients

# add in Trial and its interactions with size and orn
kk1.lmer1 = lmer(lrt ~ size * (spt + obj + grv) * orn +
  (1|subj)+
  (0+spt|subj)+
  (0+grv|subj)+
  (0+obj|subj)+
  (0+orn|subj)+
  (0+spt_orn|subj)+
  Trial*(size+orn)+
  poly(Soa,2,row=TRUE),
  data=KKL, REML=FALSE)

fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients

anova(kk1.lmer0, kk1.lmer1)

Data: KKL
Models:
kk1.lmer0: lrt ~ size * (spt + obj + grv) * orn + (1 | subj) + (0 + spt |
kk1.lmer0:      subj) + (0 + grv | subj) + (0 + obj | subj) + (0 + orn |
kk1.lmer0:      subj) + (0 + spt_orn | subj) + poly(Soa, 2, raw = TRUE)
kk1.lmer1: lrt ~ size * (spt + obj + grv) * orn + (1 | subj) + (0 + spt |
kk1.lmer1:      subj) + (0 + grv | subj) + (0 + obj | subj) + (0 + orn |
kk1.lmer1:      subj) + (0 + spt_orn | subj) + Trial * (size + orn) + poly(Soa,
kk1.lmer1:      2, raw = TRUE)
```


	Df	AIC	BIC	logLik	deviance	Chisq	Chi	Df	Pr(>Chisq)
kk1.lmer0	25	-25088	-24865	12569	-25138				
kk1.lmer1	28	-25885	-25636	12970	-25941	802.96		3	< 2.2e-16

add in quadratic effect of Trial

```
kk1.lmer2 = lmer(lrt ~ size * (spt + obj + grv) * orn +
  (1|subj)+
  (0+spt|subj)+
  (0+grv|subj)+
  (0+obj|subj)+
  (0+orn|subj)+
  (0+spt_orn|subj)+
  Trial*(size+orn)+
  TrialQ*(size+orn)+
  poly(Soa,2,raw=TRUE),
  data=KKL, REML=FALSE)
```

fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients

```
anova(kk1.lmer1, kk1.lmer2)
```

Data: KKL

Models:

```
kk1.lmer1: lrt ~ size * (spt + obj + grv) * orn + (1 | subj) + (0 + spt |
kk1.lmer1:   subj) + (0 + grv | subj) + (0 + obj | subj) + (0 + orn |
kk1.lmer1:   subj) + (0 + spt_orn | subj) + Trial * (size + orn) + poly(Soa,
kk1.lmer1:   2, raw = TRUE)
kk1.lmer2: lrt ~ size * (spt + obj + grv) * orn + (1 | subj) + (0 + spt |
kk1.lmer2:   subj) + (0 + grv | subj) + (0 + obj | subj) + (0 + orn |
kk1.lmer2:   subj) + (0 + spt_orn | subj) + Trial * (size + orn) + TrialQ *
kk1.lmer2:   (size + orn) + poly(Soa, 2, raw = TRUE)
           Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
kk1.lmer1 28 -25885 -25636  12970   -25941
kk1.lmer2 31 -26174 -25899  13118   -26236 295.77    3 < 2.2e-16
```

and allow slope of Trial to vary by subject

```
kk1.lmer3 = lmer(lrt ~ size * (spt + obj + grv) * orn +
  (1+Trial|subj)+
  (0+spt|subj)+
  (0+grv|subj)+
  (0+obj|subj)+
  (0+orn|subj)+
  (0+spt_orn|subj)+
  Trial*(size+orn)+
  TrialQ*(size+orn)+
  poly(Soa,2,raw=TRUE),
  data=KKL, REML=FALSE)
```

fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients

```
anova(kk1.lmer2, kk1.lmer3)
```

Data: KKL

Models:

```
kk1.lmer2: lrt ~ size * (spt + obj + grv) * orn + (1 | subj) + (0 + spt |
```

```

kkl.lmer2:      subj) + (0 + grv | subj) + (0 + obj | subj) + (0 + orn |
kkl.lmer2:      subj) + (0 + spt_orn | subj) + Trial * (sze + orn) + TrialQ *
kkl.lmer2:      (sze + orn) + poly(Soa, 2, raw = TRUE)
kkl.lmer3: lrt ~ sze * (spt + obj + grv) * orn + (1 + Trial | subj) + (0 +
kkl.lmer3:      spt | subj) + (0 + grv | subj) + (0 + obj | subj) + (0 +
kkl.lmer3:      orn | subj) + (0 + spt_orn | subj) + Trial * (sze + orn) +
kkl.lmer3:      TrialQ * (sze + orn) + poly(Soa, 2, raw = TRUE)
      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
kkl.lmer2 31 -26174 -25899 13118 -26236
kkl.lmer3 33 -26989 -26696 13528 -27055 818.5      2 < 2.2e-16

```

show the model (Table 3 in the manuscript)

```
print(summary(kkl.lmer3),corr=FALSE)
```

Linear mixed model fit by maximum likelihood ['lmerMod']

```

Formula: lrt ~ sze * (spt + obj + grv) * orn + (1 + Trial | subj) + (0 +
      spt | subj) + (0 + grv | subj) + (0 + obj | subj) + (0 +
      orn | subj) + (0 + spt_orn | subj) + Trial * (sze + orn) +
      TrialQ * (sze + orn) + poly(Soa, 2, raw = TRUE)
Data: KKL

```

	AIC	BIC	logLik	deviance	df.resid
	-26988.9	-26695.5	13527.5	-27054.9	53732

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-5.2071	-0.6203	-0.0982	0.5029	6.4808

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
subj	(Intercept)	0.0261682	0.16177	
	Trial	0.0026194	0.05118	-0.27
subj.1	spt	0.0044635	0.06681	
subj.2	grv	0.0011434	0.03381	
subj.3	obj	0.0006558	0.02561	
subj.4	orn	0.0058135	0.07625	
subj.5	spt_orn	0.0011040	0.03323	
	Residual	0.0344439	0.18559	

Number of obs: 53765, groups: subj, 86

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	5.659e+00	1.753e-02	322.9
sze	1.843e-01	3.501e-02	5.3
spt	7.408e-02	7.698e-03	9.6
obj	4.259e-02	4.524e-03	9.4
grv	-8.957e-04	5.122e-03	-0.2
orn	1.375e-02	9.109e-03	1.5
Trial	-4.299e-02	5.755e-03	-7.5
TrialQ	1.532e-02	9.031e-04	17.0
poly(Soa, 2, raw = TRUE)1.0	-1.009e-02	8.036e-04	-12.6
poly(Soa, 2, raw = TRUE)2.0	1.853e-02	8.991e-04	20.6
sze:spt	4.837e-02	1.540e-02	3.1
sze:obj	-1.177e-02	9.048e-03	-1.3

size:grv	-3.750e-02	1.024e-02	-3.7
size:orn	3.931e-02	1.785e-02	2.2
spt:orn	2.028e-02	6.499e-03	3.1
obj:orn	9.050e-03	7.168e-03	1.3
grv:orn	1.093e-02	7.192e-03	1.5
size:Trial	1.835e-02	1.145e-02	1.6
orn:Trial	2.785e-02	3.225e-03	8.6
size:TrialQ	-9.166e-05	1.790e-03	-0.1
orn:TrialQ	-6.078e-03	4.794e-03	-1.3
size:spt:orn	-1.425e-02	1.300e-02	-1.1
size:obj:orn	-3.466e-03	1.434e-02	-0.2
size:grv:orn	-4.680e-02	1.438e-02	-3.3

fit warnings:
fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients

The summary of `kk1.lmer3` is reported as **Table 3** in the Appendix. The above log likelihood ratio tests are summarized in **Table 1**.

3.2 Table 4 (appendix)

Table 4 (in the appendix) reports the summary of the following GAMM:

```
kkl.gamD = bam(lrt ~ size * (spt + obj + grv) * orn +
               s(Trial, subj, bs="fs", m=1)+
               s(subj, spt, bs="re") +
               s(subj, grv, bs="re") +
               s(subj, obj, bs="re") +
               s(subj, orn, bs="re") +
               s(subj, spt_orn, bs="re") +
               s(Trial, by=Int)+
               s(Soa),
               AR.start=FirstTrial, rho=0.15,
               data=KKL, method="fREML", discrete=TRUE)
summary(kkl.gamD)
```

Family: gaussian
Link function: identity

Formula:

```
lrt ~ size * (spt + obj + grv) * orn + s(Trial, subj, bs = "fs",
    m = 1) + s(subj, spt, bs = "re") + s(subj, grv, bs = "re") +
    s(subj, obj, bs = "re") + s(subj, orn, bs = "re") + s(subj,
    spt_orn, bs = "re") + s(Trial, by = Int) + s(Soa)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.6850584	0.0189221	300.445	< 2e-16
size	0.1839522	0.0378441	4.861	1.17e-06
spt	0.0729251	0.0079223	9.205	< 2e-16
obj	0.0410758	0.0041140	9.984	< 2e-16
grv	-0.0004952	0.0049377	-0.100	0.920111
orn	0.0374653	0.0141694	2.644	0.008193
size:spt	0.0482519	0.0158446	3.045	0.002326
size:obj	-0.0088449	0.0082283	-1.075	0.282407
size:grv	-0.0366134	0.0098757	-3.707	0.000210
size:orn	0.0095644	0.0283388	0.338	0.735741
spt:orn	0.0213368	0.0063789	3.345	0.000824
obj:orn	0.0082761	0.0068235	1.213	0.225181
grv:orn	0.0077985	0.0068488	1.139	0.254848
size:spt:orn	-0.0097831	0.0127578	-0.767	0.443181
size:obj:orn	-0.0075086	0.0136470	-0.550	0.582183
size:grv:orn	-0.0482719	0.0136977	-3.524	0.000425

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Trial,subj)	557.867	774.000	661.847	< 2e-16
s(spt,subj)	77.793	86.000	31.591	< 2e-16
s(grv,subj)	47.606	84.000	2.164	1.82e-10
s(obj,subj)	29.571	84.000	1.340	0.00114
s(orn,subj)	44.080	84.000	1.148	< 2e-16
s(spt_orn,subj)	47.373	84.000	1.315	1.68e-10

```
s(Trial):Intbig.cardinal      8.349   8.679  13.402 < 2e-16
s(Trial):Intsmall.cardinal    8.053   8.489   9.837 1.44e-14
s(Trial):Intbig.diagonal      5.887   6.574   5.856 8.00e-06
s(Trial):Intsmall.diagonal    7.997   8.453   8.862 6.39e-08
s(Soa)                        5.602   6.747 103.115 < 2e-16
```

```
R-sq.(adj) = 0.532   Deviance explained = 54%
fREML = -15136   Scale est. = 0.032571   n = 53765
```

```
AIC(kkl.gamD)
```

```
[1] -31890
```

3.3 Figure 9

```
# For Figure 9, we first define a function for acf plots with lattice
acfs = function(m=dat.gamA, s=selectedSubjects, mydata=dat, add_NA=FALSE,
               ci=0.95, rawplot=FALSE) {
  mydata$resids = resid(m)
  res = list()
  n.used = rep(0,length(s))
  for (i in 1:length(s)) {
    tmp = mydata[mydata$subj == s[i],]
    if (!add_NA) {
      x = acf(tmp$resids, plot=rawplot)
    } else {
      v = rep(NA, max(tmp$trial))
      v[tmp$trial] = tmp$resids
      x = acf(v, na.action=na.pass, plot=rawplot)
    }
    res[[i]] = data.frame(
      lag = x$lag,
      acf = x$acf,
      subj = factor(s[i]))
    n.used[i] = x$n.used
  }
  Dfr = do.call(rbind, res)

  civec = qnorm((1 + ci)/2)/sqrt(n.used)

  theplot = xyplot(acf ~ lag | subj, type = "h", data = Dfr, #col.line = "black",
    panel = function(x, y, subscripts, ...) {
      panel.abline(h = civec[panel.number()], col.line = "grey")
      panel.abline(h = -civec[panel.number()], col.line = "grey")
      panel.abline(h = 0, col.line = "black")
      d=Dfr[subscripts,]
      panel.xyplot(d$lag, d$acf,
        col=ifelse(abs(d$acf)>civec[panel.number()] & d$lag > 0,
          "indianred", "steelblue2"),
        ...)
    },
    #strip = strip.custom(bg = "grey90"),
    #par.strip.text = list(cex = 0.8),
    xlab="lag", ylab="autocorrelation",
    layout=c(4,1,1))

  return(list(vals=Dfr, plot=theplot))
}
# a LMM
kk1.lmer = lmer(lrt ~ sze * (spt + obj + grv) * orn +
  (1|subj)+
  (0+spt|subj)+
  (0+grv|subj)+
  (0+obj|subj)+
  (0+orn|subj)+
```

```

(0+spt_orn|subj)+
poly(Soa,2,raw=TRUE), data=KKL, REML=FALSE)

# make plot for selected subjects
selectedSubjects = c(3, 43, 136, 123)
x = acfs(kkl.lmer, selectedSubjects, KKL)
print(x[[2]]) # this produces the plot

```

Figure 9, lower panels.

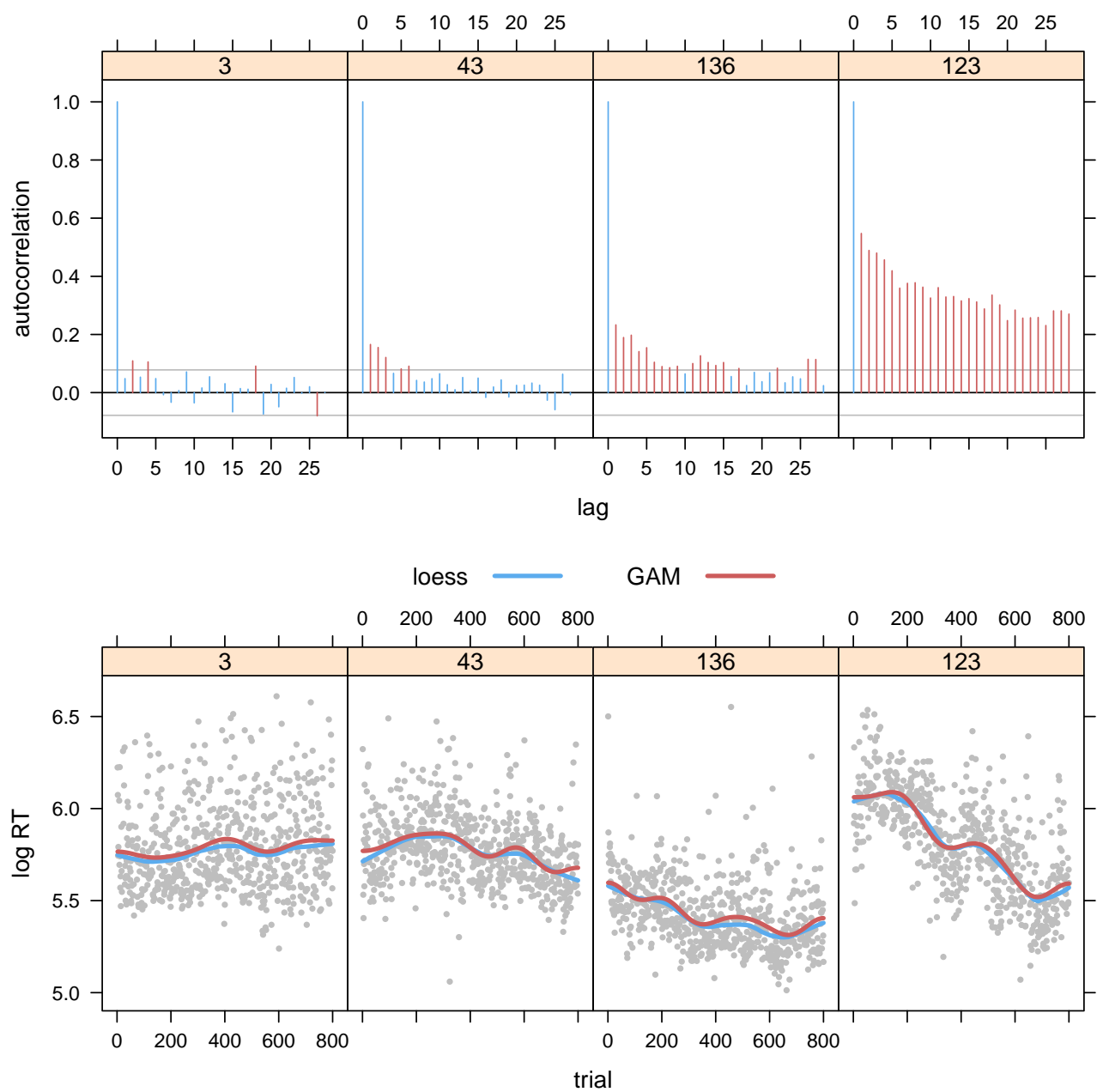
```

selectedSubjects = c(3, 43, 136, 123)
dfr = droplevels(KKL[KKL$subj %in% selectedSubjects,])
dfr$subj = ordered(dfr$subj, selectedSubjects)
dfr.gam = bam(lrt~1+s(trial, subj, bs="fs", m=1), data=dfr, method="fREML")
dfr$Gam = predict(dfr.gam)

keys = list(space="top", text=list(c("loess", "GAM")), columns=2,
          lines=list(lty=c(1,1), col=c("steelblue2", "indianred"), lwd=3))
xyplot(lrt ~ trial | subj, data = dfr, #col.line = "indianred",
  panel = function(x, y, subscripts, ...) {
    panel.xyplot(x, y, pch=19, col="gray", cex=0.4, ...)
    panel.loess(x, y, lwd=3, span=0.3, col="steelblue2", ...)
    d = dfr[subscripts,]
    panel.lines(d$trial, d$Gam, col="indianred", lwd=3)
  },
  #strip = strip.custom(bg = "grey90"),
  #par.strip.text = list(cex = 0.8),
  xlab="trial", ylab="log RT",
  layout=c(4,1,1),
  key=keys)

```

Figure 9.

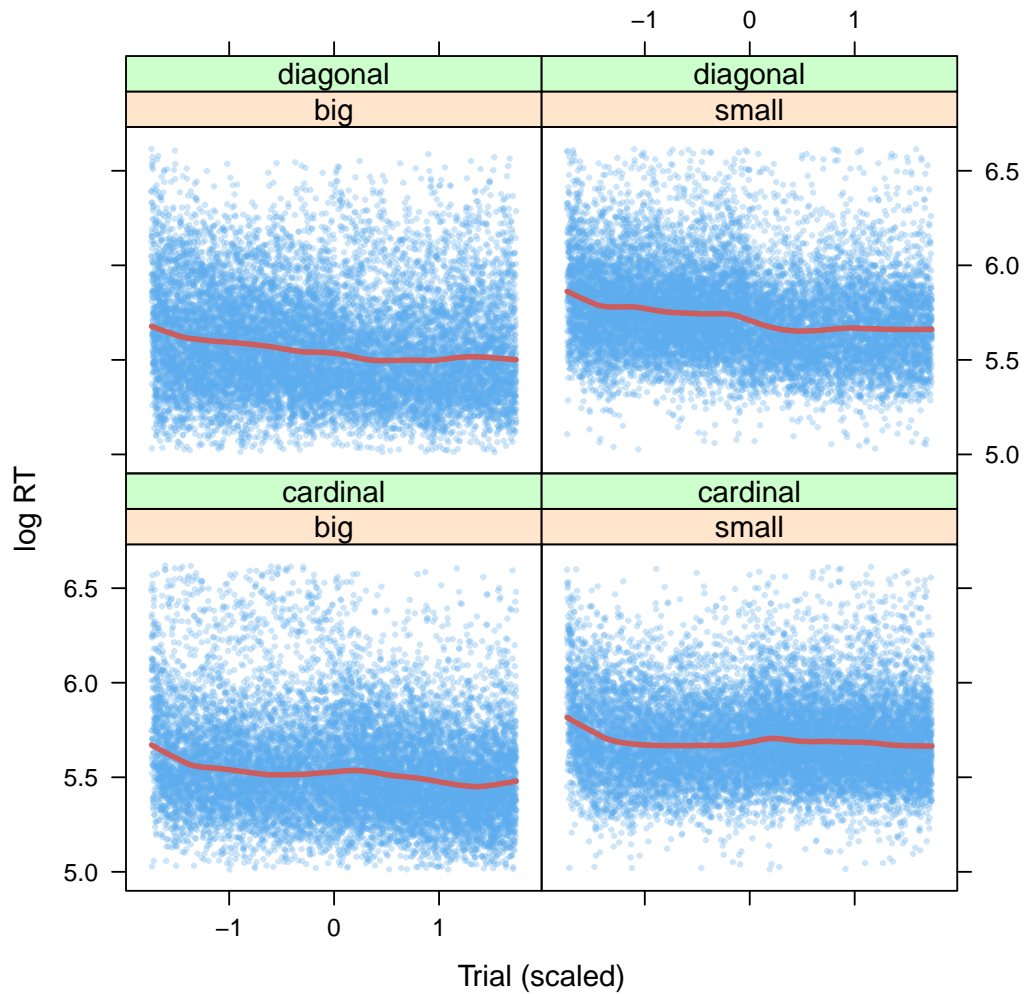


3.4 Figure 10

```
selectedSubjects = c(3, 43, 136, 123)
myColor="steelblue2"
add.alpha <- function(col=NULL, alpha=1){
  if(missing(col)) stop("Please provide a vector of colours.")
  apply(sapply(col, col2rgb)/255, 2,
        function(x) rgb(x[1], x[2], x[3], alpha=alpha))
}

myColorAlpha = add.alpha(myColor, alpha=0.3)

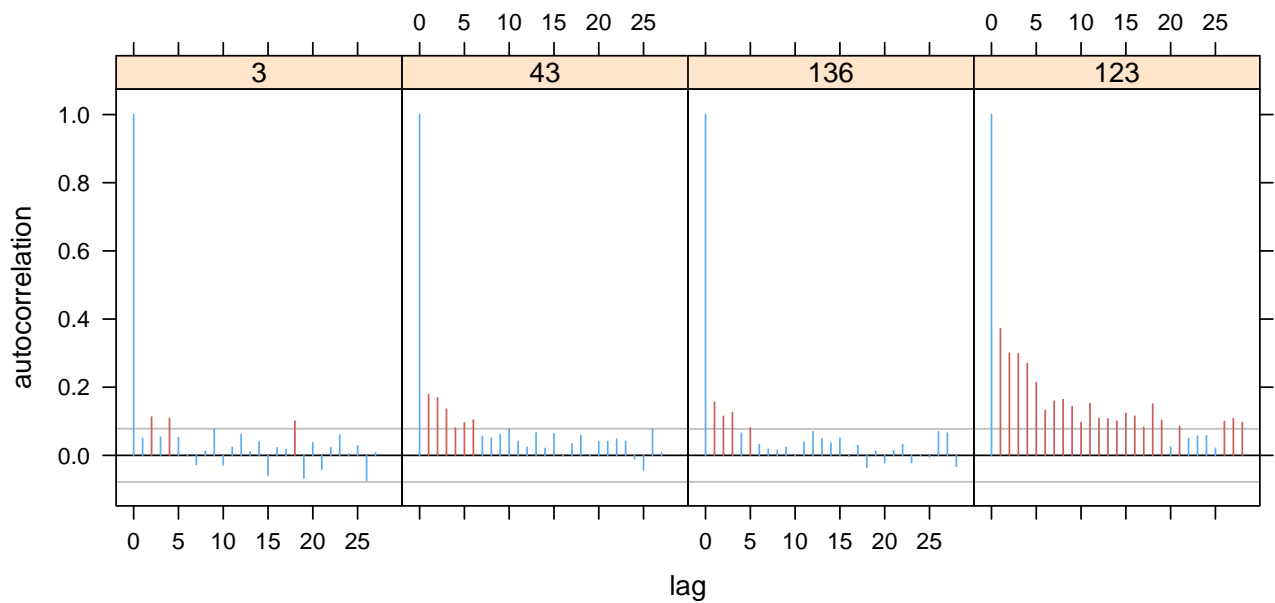
xyplot(lrt~Trial|size+cardinal, data=KKL, type=c("p", "smooth"),span=0.2,
       col.line="indianred", cex=0.3, pch=19, col=myColorAlpha, lwd=3,
       xlab="Trial (scaled)", ylab="log RT")
```



3.5 Figure 11

```
x = acfs(kkl.lmer3, selectedSubjects, KKL)
print(x[[2]])
```

```
load("models/kkl.lmer3.rda")
x = acfs(kkl.lmer3, selectedSubjects, KKL)
pdf("figs/Figure11.pdf", he=4, wi=8)
print(x[[2]])
dev.off()
```



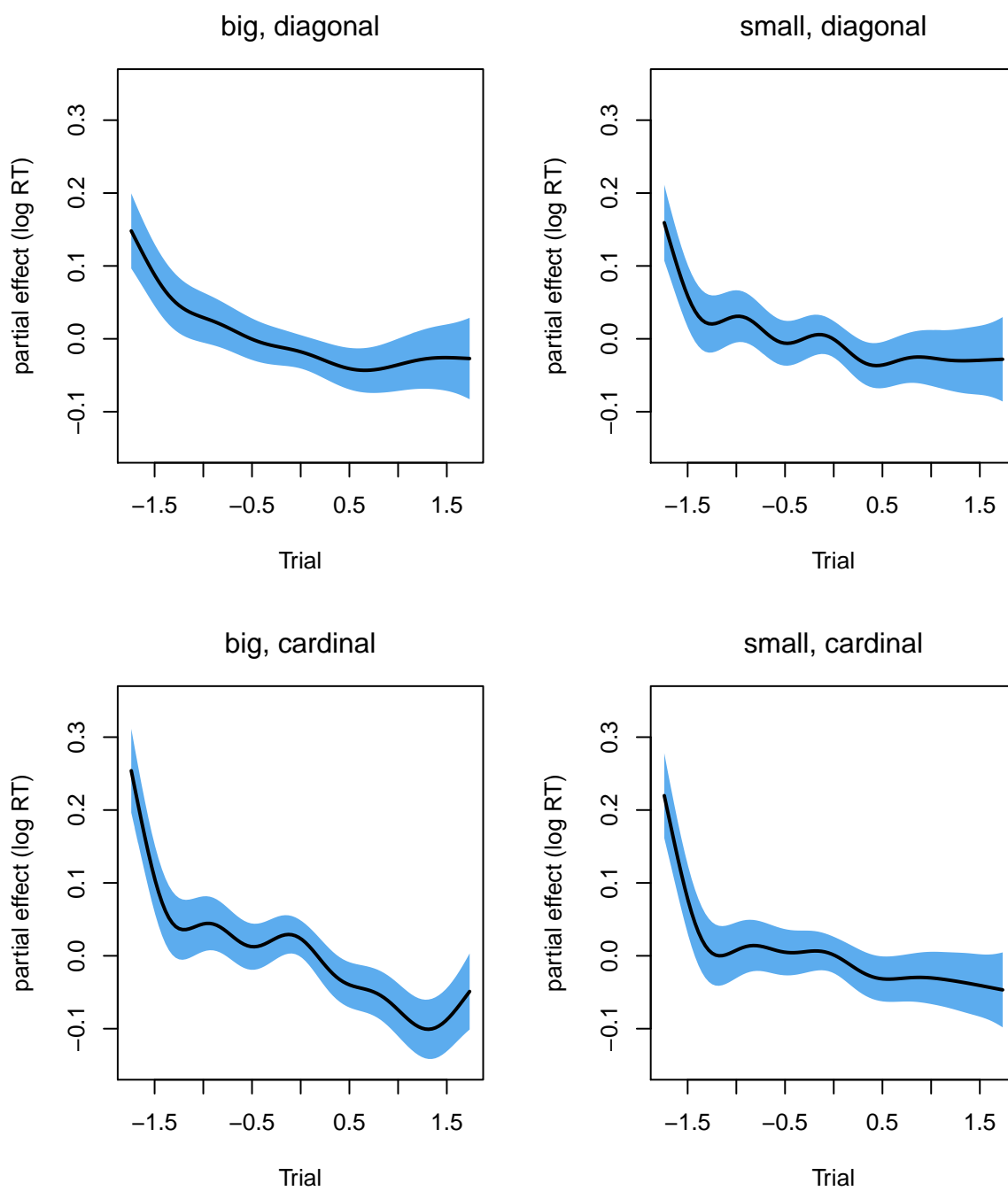
3.6 Figure 12

```
par(mfrow=c(2,2),mar=c(5,5,3,1),oma=rep(2,4))
ylim=c(-0.15, 0.35)
plot(kkl.gamD, select=7, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("big, cardinal", 3, 1.0)
plot(kkl.gamD, select=8, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("small, cardinal", 3, 1.0)
plot(kkl.gamD, select=9, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("big, diagonal", 3, 1.0)
plot(kkl.gamD, select=10, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("small, diagonal", 3, 1.0)
```

```
pdf("figs/Figure12.pdf", he=8,wi=7)
par(mfrow=c(2,2),mar=c(5,5,3,1),oma=rep(2,4))
ylim=c(-0.15, 0.35)
plot(kkl.gamD, select=9, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("big, diagonal", 3, 1.0)
plot(kkl.gamD, select=10, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("small, diagonal", 3, 1.0)
plot(kkl.gamD, select=7, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("big, cardinal", 3, 1.0)
plot(kkl.gamD, select=8, rug=F, ylim=ylim, scheme=1,
     xlab="Trial", ylab="partial effect (log RT)",
     shade.col="steelblue2", col="indianred", lwd=2)
mtext("small, cardinal", 3, 1.0)

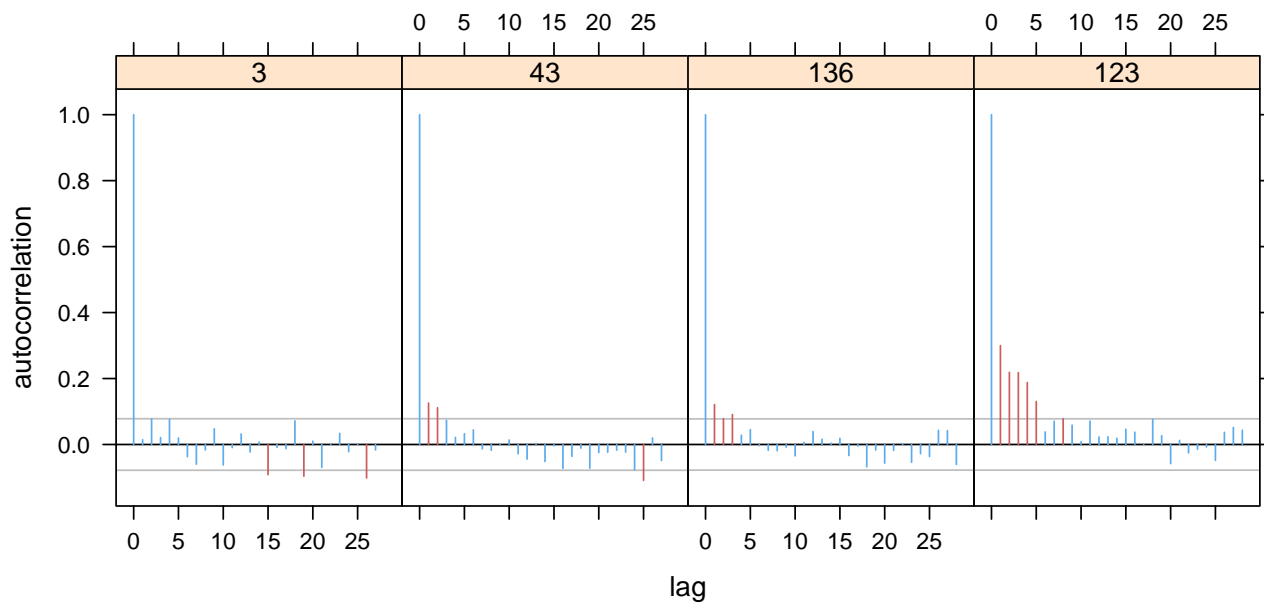
dev.off()
```

Figure 12.



3.7 Figure 13

```
x = acfs(kk1.gamD, selectedSubjects, KKL)  
print(x[[2]])
```



3.8 Table 2

Table 2 is based on the following models:

```
# reference model
kkl.ref.gam <- bam(lrt ~ sze * (spt + obj + grv) * orn +
  s(subj, bs="re") +
  s(subj, spt, bs="re") +
  s(subj, grv, bs="re") +
  s(subj, obj, bs="re") +
  s(subj, orn, bs="re") +
  s(subj, spt_orn, bs="re") +
  poly(Soa,2,raw=TRUE),
  data=KKL, method="fREML", discrete=TRUE)

# 'linear' model
kkl.lin.gam <- bam(lrt ~ sze * (spt + obj + grv) * orn +
  s(subj, bs="re") +
  s(subj, Trial, bs="re") +
  s(subj, spt, bs="re") +
  s(subj, grv, bs="re") +
  s(subj, obj, bs="re") +
  s(subj, orn, bs="re") +
  s(subj, spt_orn, bs="re") +
  Trial*(sze+orn)+
  TrialQ*(sze+orn)+
  poly(Soa,2,raw=TRUE),
  data=KKL, method="fREML", discrete=TRUE)

# model with factor smooths
kkl.fs.gam = bam(lrt ~ sze * (spt + obj + grv) * orn +
  s(Trial, subj, bs="fs", m=1)+
  s(subj, spt, bs="re") +
  s(subj, grv, bs="re") +
  s(subj, obj, bs="re") +
  s(subj, orn, bs="re") +
  s(subj, spt_orn, bs="re") +
  Trial*(sze+orn)+
  TrialQ*(sze+orn)+
  s(Soa),
  data=KKL, method="fREML", discrete=TRUE)

# model with smooths for the interactions with trial
kkl.trial.gam = bam(lrt ~ sze * (spt + obj + grv) * orn +
  s(Trial, subj, bs="fs", m=1)+
  s(subj, spt, bs="re") +
  s(subj, grv, bs="re") +
  s(subj, obj, bs="re") +
  s(subj, orn, bs="re") +
  s(subj, spt_orn, bs="re") +
  s(Trial, by=Int)+
  s(Soa),
  data=KKL, method="fREML", discrete=TRUE)
```

Model comparison was carried out with compareML from the itsadug package.

```
compareML(kkl.ref.gam, kkl.lin.gam)$table
```

	Model	Score	Edf	Chisq	Df	p.value	Sig.
1	kkl.ref.gam	-12495.77	27				
2	kkl.lin.gam	-13422.25	34	926.484	7.000	< 2e-16	***

```
compareML(kkl.lin.gam, kkl.fs.gam)$table
```

	Model	Score	Edf	Difference	Df	p.value	Sign.
1	kkl.lin.gam	-13422.25	34				
2	kkl.fs.gam	-14500.08	31	-1077.822	3.000	<NA>	

```
compareML(kkl.fs.gam, kkl.trial.gam)$table
```

	Model	Score	Edf	Chisq	Df	p.value	Sig.
1	kkl.fs.gam	-14500.08	31				
2	kkl.trial.gam	-14582.64	33	82.565	2.000	< 2e-16	***

```
AIC(kkl.ref.gam)
```

```
[1] -26009.55
```

```
AIC(kkl.lin.gam)
```

```
[1] -28047.29
```

```
AIC(kkl.fs.gam)
```

```
[1] -30876.72
```

```
AIC(kkl.trial.gam)
```

```
[1] -31040.29
```

These stats are brought together in Table 2.

4 The baldey dataset

4.1 Model and code for Table 5

For the baldey dataset, we restricted ourselves to correct responses to words.

```
baldey = droplevels(baldey[baldey$response=="correct" &  
  baldey$word_status=="word",])
```

A transform $-1000/RT$ for the reaction times brings a distribution with a strong rightward skew closer to normality. Outlier reaction times (less than 600 ms) are removed.

```
baldey = baldey[baldey$RT > 600,] # 0.4% data loss  
baldey$RTinv = -1000/baldey$RT
```

We add information about the initial trials for the subjects, and then scale trial.

```
mintrials = tapply(baldey$trial, baldey$subject, min)  
baldey$MinTrial = mintrials[as.character(baldey$subject)]  
baldey$AR.start = baldey$MinTrial==baldey$trial
```

We log-transform lemma frequency to avoid outlier effects of high-frequency ‘outliers’.

```
baldey$LemmaFreq = log(baldey$CELEX_lemma_freq+1)
```

After scaling predictors,

```
baldey$LemmaFreq = log(baldey$CELEX_lemma_freq+1)  
baldey$LemmaFreqSc = as.numeric(scale(baldey$LemmaFreq))  
baldey$WordDurSc = as.numeric(scale(baldey$word_duration))  
baldey$TrialSc = as.numeric(scale(baldey$trial))
```

we fit a GAMM to the data.

```
baldey.gam <- bam(RTinv ~ gender +  
  s(LemmaFreqSc, by=gender, bs="cr") +  
  te(WordDurSc, TrialSc) +  
  s(word, bs="re") +  
  s(word, gender, bs="re") +  
  s(subject, WordDurSc, bs="re") +  
  s(session, subject, bs="fs", m=1),  
  AR.start=AR.start, rho=0.2,  
  data=baldey, discrete=TRUE)
```

The subtable of the parametric part of the model (see Table 5):

```
summary(baldey.gam)$p.table
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.0381929	0.06834953	-15.189467	5.555810e-52
gendermale	0.3732205	0.08909260	4.189131	2.805468e-05

The subtable of the smooths of the model (also in Table 5):


```
summary(baldehy.gam)$s.table
```

	edf	Ref.df	F	p-value
s(LemmaFreq):genderfemale	3.012133	3.430126	5.511694e+01	1.612847e-38
s(LemmaFreq):gendermale	1.014891	1.021960	1.247278e+02	3.398336e-29
te(word_duration,Trial)	8.705616	10.406449	2.271702e+01	4.892898e-44
s(word)	1735.807042	2777.000000	2.015884e+00	0.000000e+00
s(gender,word)	495.787366	5544.000000	1.108202e-01	1.967590e-07
s(word_duration,subject)	18.744049	19.000000	7.988325e+01	1.592307e-293
s(session,subject)	154.417238	178.000000	1.263994e+05	0.000000e+00

The fREML score,

```
baldehy.gam$gcv.ubre
```

```
fREML  
-14903.7
```

and the AIC:

```
AIC(baldehy.gam)
```

```
[1] -31728.01
```

4.2 Table 6: the model with linear effects only

The GAMM with linear effects only (Table 6).

```
baldey.lin <- bam(RTinv ~ gender +
  LemmaFreqSc*gender+
  WordDurSc*TrialSc +
  s(word, bs="re") +
  s(word, gender, bs="re") +
  s(subject, bs="re")+
  s(subject, WordDurSc, bs="re") +
  s(subject, session, bs="re"),
  data=baldey, discrete=TRUE)
```

The subtable of the parametric part of the model (Table 6).

```
summary(baldey.lin.gam)$p.table
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.942629376	0.0406017546	-23.216469	1.485951e-118
gendermale	0.237397743	0.0574021321	4.135696	3.545059e-05
LemmaFreqSc	-0.020886829	0.0016662431	-12.535283	5.478266e-36
WordDurSc	0.061461226	0.0070232829	8.751068	2.183294e-18
TrialSc	0.006232360	0.0008371858	7.444417	9.906287e-14
gendermale:LemmaFreqSc	0.002426149	0.0017620096	1.376922	1.685431e-01
WordDurSc:TrialSc	-0.005362875	0.0008329795	-6.438183	1.220947e-10

The subtable of the smooths of the model (Table 6).

```
summary(baldey.lin.gam)$s.table
```

	edf	Ref.df	F	p-value
s(word)	1668.12107	2777	2.790862e+00	2.023020e-214
s(gender,word)	451.76914	5544	1.021730e-01	4.817559e-09
s(subject)	17.93030	18	9.009013e+05	0.000000e+00
s(WordDurSc,subject)	18.73591	19	1.213828e+02	6.843180e-178
s(session,subject)	19.89525	20	7.170278e+05	0.000000e+00

The fREML and AIC scores:

```
baldey.lin.gam$gcv.ubre
```

```
fREML
-13027.88
```

```
AIC(baldey.lin.gam)
```

```
[1] -27582.51
```

Very similar results are obtained with the LMM:

```
baldey.lmer <- lmer(RTinv ~ gender +
                    LemmaFreqSc*gender+
                    WordDurSc*TrialSc +
                    (1+gender|word) +
                    (1+WordDurSc+session|subject),
                    data=baldey)
```

```
print(summary(baldey.lmer),corr=F)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: RTinv ~ gender + LemmaFreqSc * gender + WordDurSc * TrialSc +
  (1 + gender | word) + (1 + WordDurSc + session | subject)
Data: baldey
```

REML criterion at convergence: -26072.5

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-5.3489	-0.5975	-0.0130	0.6110	4.3265

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
word	(Intercept)	0.0034549	0.05878	
	gendermale	0.0009739	0.03121	-0.13
subject	(Intercept)	0.0254792	0.15962	
	WordDurSc	0.0009483	0.03079	-0.82
	session	0.0003140	0.01772	-0.22 0.06
Residual		0.0316786	0.17798	

Number of obs: 48385, groups: word, 2780; subject, 20

Fixed effects:

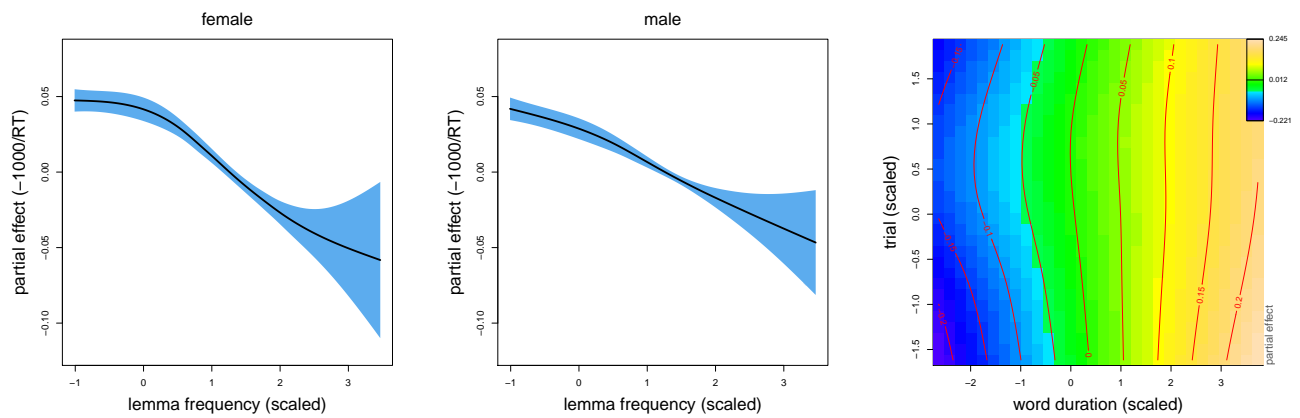
	Estimate	Std. Error	t value
(Intercept)	-0.8558065	0.0402314	-21.272
gendermale	0.0479604	0.0401551	1.194
LemmaFreqSc	-0.0208485	0.0016369	-12.737
WordDurSc	0.0619134	0.0070152	8.826
TrialSc	0.0062278	0.0008371	7.440
gendermale:LemmaFreqSc	0.0023540	0.0017618	1.336
WordDurSc:TrialSc	-0.0053851	0.0008328	-6.466

```
compareML(baldey.gam, baldey.lin)
```

	Model	Score	Edf	Chisq	Df	p.value	Sig.
1	baldey.lin	-13027.88	12				
2	baldey.gam	-14911.48	16	1883.594	4.000	< 2e-16	***

4.3 Figure 14

```
par(mfrow=c(1,3),mar=c(4,4,3,4))
plot(baldehy.gam, select=1, scheme=1, ylim=c(-0.12, 0.08), rug=FALSE,
     shade.col="steelblue2", col="indianred", lwd=2, xlab="lemma frequency",
     ylab="partial effect (-1000/RT)")
mtext("female", 3, 1)
plot(baldehy.gam, select=2, scheme=1, ylim=c(-0.12, 0.08), rug=FALSE,
     shade.col="steelblue2", col="indianred", lwd=2, xlab="lemma frequency",
     ylab="partial effect (-1000/RT)")
mtext("male", 3, 1)
pvisgam(baldehy.gam, select=3, view=c("WordDurSc", "TrialSc"),
        xlab="word duration (scaled)", ylab="trial (scaled)", main = " ")
```



Coefficient of variation for RTs and error rate.

```
library(RePsychLing)
data(baldey)
# analysis 1: error proportions and cv calculated for RTs exceeding 600 ms
baldey = baldey[baldey$RT > 600,]
baldey$RTinv=-1000/baldey$RT

incor = tapply(baldey$response=="incorrect", baldey$subject, sum)
corr = tapply(baldey$response=="correct", baldey$subject, sum)
pincor = incor/(incor+corr)

sds = tapply(baldey$RTinv, baldey$subject, sd)
means = tapply(baldey$RTinv, baldey$subject, mean)
cvs = sds/means

cor.test(pincor, cvs)
```

Pearson's product-moment correlation

```
data:  pincor and cvs
t = -2.2054, df = 18, p-value = 0.04067
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.75055279 -0.02349328
sample estimates:
      cor
-0.4612191
```

```
# analysis 2: error proportions calculated across all observations,  
# whereas cv is calculated over data points with RTs exceeding 600 ms  
data(baldey)
```

```
incor = tapply(baldey$response=="incorrect", baldey$subject, sum)  
corr = tapply(baldey$response=="correct", baldey$subject, sum)  
pincor = incor/(incor+corr)
```

```
baldey = baldey[baldey$RT > 600,]  
baldey$RTinv=-1000/baldey$RT  
sds = tapply(baldey$RTinv, baldey$subject, sd)  
means = tapply(baldey$RTinv, baldey$subject, mean)  
cvs = sds/means
```

```
cor.test(pincor, cvs)
```

Pearson's product-moment correlation

```
data: pincor and cvs  
t = -3.5808, df = 18, p-value = 0.002136  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
-0.8460326 -0.2833349  
sample estimates:  
cor  
-0.6449863
```

5 The poems dataset

5.1 The full GAMM

```
data(poems) # available in RePsychLing, which was required above
# log-transform and scale frequency
poems$Fre = scale(log(poems$WordFormFrequency+1))
# log-transform reading time
poems$Lrt = log(poems$ReadingTime)
# scale frequency
poems$FreSc = as.numeric(scale(poems$Fre))
# scale trial
poems$TrialSc = as.numeric(scale(poems$Trial))
# order by subject and trial, and mark initial trial as the start
# of a new time series; this is necessary for
poems = poems[order(poems$Subject, poems$Trial),]
pos = tapply(poems$Trial, poems$Subject, min)
poems$MinTrial = pos[as.character(poems$Subject)]
poems$Start = poems$MinTrial==poems$Trial
```

A model with a tensor product smooth of frequency by trial. (Fitting this model takes about 30 minutes, and producing the summary takes another 30 minutes.) This model is summarized in Table 7.

```
poems.gam = bam(Lrt ~ te(Fre, TrialSc) +
                 s(Poem, bs="re") +
                 s(TrialSc, Subject, bs="fs", m=1) +
                 s(Subject, Fre, bs="re"),
                 AR.start = poems$Start, rho = 0.30,
                 data = poems, discrete=TRUE)
summary(poems.gam)
```

Family: gaussian
Link function: identity

Formula:

```
Lrt ~ te(Fre, TrialSc) + s(Poem, bs = "re") + s(TrialSc, Subject,
        bs = "fs", m = 1) + s(Subject, Fre, bs = "re")
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.04636	0.01739	347.7	<2e-16

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
te(Fre,TrialSc)	10.29	11.4	80.59	<2e-16
s(Poem)	81.19	87.0	19.45	<2e-16
s(TrialSc,Subject)	2163.53	2934.0	329.42	<2e-16
s(Fre,Subject)	304.11	326.0	14.88	<2e-16

R-sq.(adj) = 0.498 Deviance explained = 50.3%
fREML = 49642 Scale est. = 0.08918 n = 275996

5.2 A decompositional GAMM

A decompositional model with main effects for frequency and trial and a ‘tinsor’ product smooth for the interaction of frequency by trial (summarized in Table 8).

```
poems.gam2 = bam(Lrt ~ ti(Fre) + ti(TrialSc) + ti(Fre, TrialSc) +  
                  s(Poem, bs="re") +  
                  s(TrialSc, Subject, bs="fs", m=1) +  
                  s(Subject, Fre, bs="re"),  
                  AR.start = poems$Start, rho = 0.30,  
                  data = poems, discrete=TRUE)  
summary(poems.gam2)
```

Family: gaussian

Link function: identity

Formula:

```
Lrt ~ ti(Fre) + ti(TrialSc) + ti(Fre, TrialSc) + s(Poem, bs = "re") +  
      s(TrialSc, Subject, bs = "fs", m = 1) + s(Subject, Fre, bs = "re")
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.0461	0.0174	347.4	<2e-16

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
ti(Fre)	1.566	1.880	247.865	<2e-16
ti(TrialSc)	3.899	3.915	90.924	<2e-16
ti(Fre,TrialSc)	8.051	10.187	9.931	<2e-16
s(Poem)	81.180	87.000	19.405	<2e-16
s(TrialSc,Subject)	2163.630	2934.000	323.821	<2e-16
s(Fre,Subject)	304.082	326.000	14.874	<2e-16

R-sq.(adj) = 0.498 Deviance explained = 50.3%

fREML = 49636 Scale est. = 0.089175 n = 275996

5.3 A model with linear effects only

Table 9 summarizes a LMM for the `poems` data.

```
poems.lmer = lmer(Lrt ~ FreSc * TrialSc +
                  (1|Poem) +
                  (1+FreSc+TrialSc|Subject),
                  data = poems, REML=TRUE)
print(summary(poems.lmer),corr=FALSE)
```

Linear mixed model fit by REML ['lmerMod']
Formula: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 + FreSc + TrialSc | Subject)
Data: poems

REML criterion at convergence: 150493.3

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-5.9258	-0.6089	-0.1266	0.4605	5.6861

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
Subject	(Intercept)	0.059568	0.24407	
	FreSc	0.001176	0.03430	-0.61
	TrialSc	0.009619	0.09808	0.01 0.04
Poem	(Intercept)	0.002534	0.05034	
Residual		0.099396	0.31527	

Number of obs: 275996, groups: Subject, 326; Poem, 87

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	6.0389852	0.0145774	414.3
FreSc	-0.0525598	0.0019991	-26.3
TrialSc	-0.0782900	0.0054865	-14.3
FreSc:TrialSc	0.0038955	0.0006195	6.3

Informal comparison of `poems.gam` and `poems.lmer`:

```

# AIC scores
AIC(poems.gam)

[1] 92688.55

AIC(poems.lmer)

[1] 150517.3

# REML scores
poems.gam$gcv.ubre

      fREML
49642.31

# 150493.3 for lmer, see summary above
cor(fitted(poems.gam), poems$Lrt)^2

[1] 0.5030827

cor(fitted(poems.lmer), poems$Lrt)^2

[1] 0.4292628

```

5.4 Figure 15

```
# an ancillary function for acfs with lattice
acfs = function(dat=poems, model = poems.gam) {
  dat$Resid = resid_gam(model, AR_start=poems$Start, incl_na=TRUE)
  res=list()
  for (i in 1:nlevels(dat$Subject)) {
    tmp = dat[dat$Subject==levels(dat$Subject)[i],]
    acfmod = acf(tmp$Resid, na.action=na.pass, plot=F)
    acfobs = acf(tmp$Lrt, plot=F)
    res[[i]] = data.frame(lag = acfmod$lag,
                          acf = acfmod$acf,
                          acfObs = acfobs$acf)

    res[[i]]$subj = levels(dat$Subject)[i]
    res[[i]]$n.used.mod = acfmod$n.used
    res[[i]]$n.used.obs = acfobs$n.used
    res[[i]]$ci95obs = qnorm((1+0.95)/2)/sqrt(acfobs$n.used)
    res[[i]]$ci95mod = qnorm((1+0.95)/2)/sqrt(acfmod$n.used)
  }
  res = do.call(rbind, res)
  return(res)
}

acfdat = acfs()
```

```
load("models/acfdat.poems.rda")
```

The top panel of Figure 15 is produced by the following code snippet.

```
selectedSubjects = c("Subj265", "Subj176", "Subj19", "Subj183")
tmp = acfdat[acfdat$subj %in% selectedSubjects,]
tmp$subj = ordered(tmp$subj, selectedSubjects)
ci = unname(tapply(tmp$ci95obs, tmp$subj, mean))
theplotObs = xyplot(acfObs ~ lag | subj, type = "h", dat = tmp, ylim=c(0,1),
  panel = function(x, y, subscripts, ...) {
    panel.abline(h = ci[panel.number()], col.line = "grey")
    panel.abline(h = -ci[panel.number()], col.line = "grey")
    panel.abline(h = 0, col.line = "black")
    d=tmp[subscripts,]
    panel.xyplot(d$lag, d$acfObs,
      col=ifelse(abs(d$acfObs)> ci[panel.number()] & d$lag > 0,
        "indianred", "steelblue2"),
      ...)
  },
  xlab="lag", ylab="autocorrelation",
  layout=c(4,1,1))
print(theplotObs) # show plot
```

The center panel of Figure 15 has a smooth based on a simple GAM with only a factor smooth:

```
Lrt ~ s(Trial, Subject, bs="fs", m=1)
```

The following code produces the center panel:

```

selectedSubjects = c("Subj265", "Subj176", "Subj19", "Subj183")
dfr = droplevels(poems[poems$Subject %in% selectedSubjects,])
dfr$Subject = ordered(dfr$Subject, selectedSubjects)
dfr.gam = bam(Lrt~1+s(Trial, Subject, bs="fs", m=1), data=dfr, method="fREML")
dfr$Gam = predict(dfr.gam)

keys = list(space="top", text=list(c("loess", "GAM")), columns=2,
           lines=list(lty=c(1,1), col=c("steelblue2", "indianred"), lwd=3))
xyplot(Lrt ~ Trial | Subject, data = dfr,
       panel = function(x, y, subscripts, ...) {
         panel.xyplot(x, y, pch=19, col="gray", cex=0.4, ...)
         panel.loess(x, y, lwd=3, span=0.3, col="steelblue2", ...)
         d = dfr[subscripts,]
         panel.lines(d$Trial, d$Gam, col="indianred", lwd=3)
       },
       xlab="Trial", ylab="log RT",
       layout=c(4,1,1),
       key=keys)

```

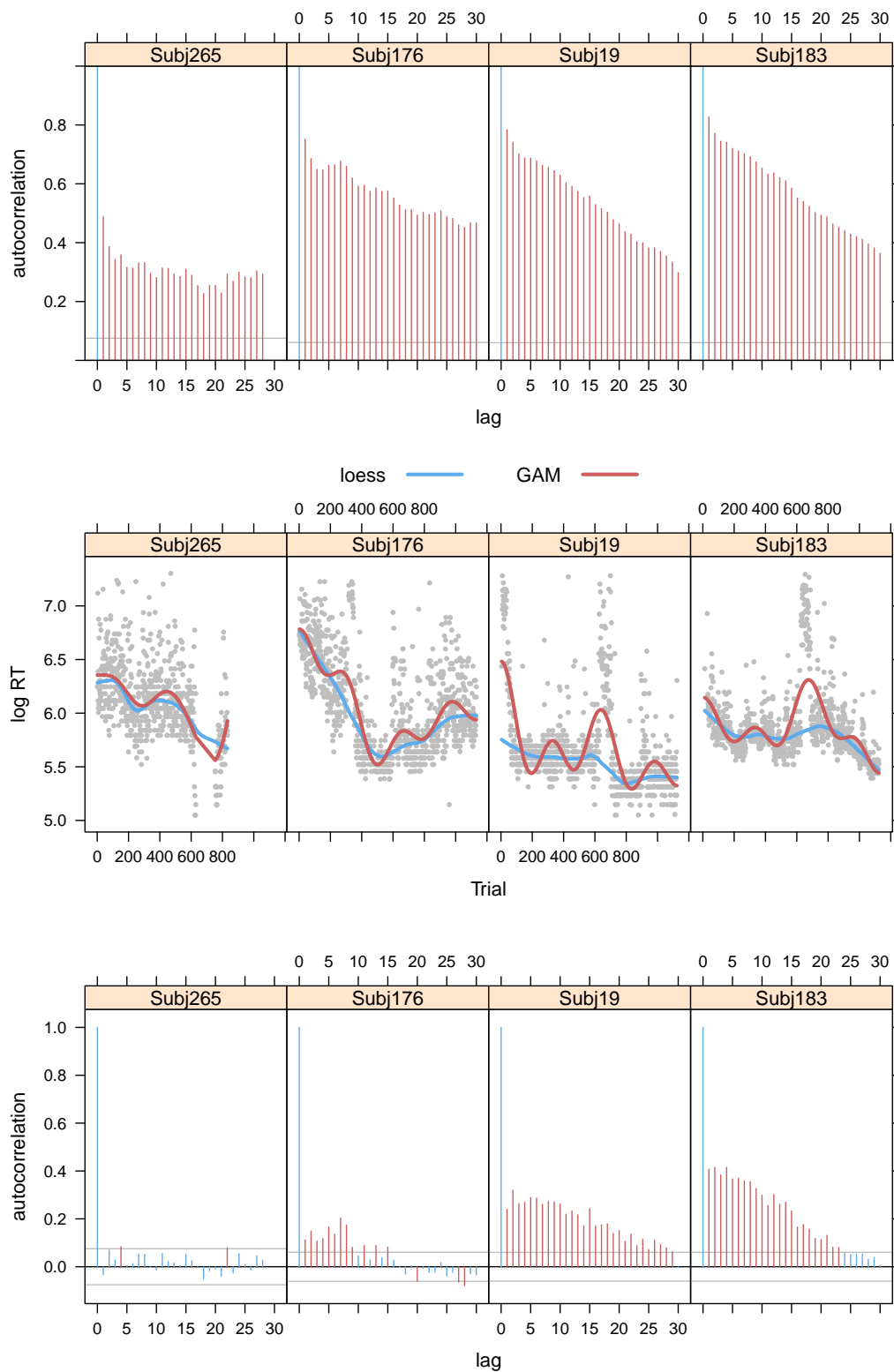
The bottom panel of Figure 15 is obtained in a similar way, but now using the model predictions.

```

tmp = acfdat[acfdat$subj %in% selectedSubjects,]
tmp$subj = ordered(tmp$subj, selectedSubjects)
ci = unname(tapply(tmp$ci95mod, tmp$subj, mean))
theplot = xyplot(acf ~ lag | subj, type = "h", dat = tmp,
                 panel = function(x, y, subscripts, ...) {
                   panel.abline(h = ci[panel.number()], col.line = "grey")
                   panel.abline(h = -ci[panel.number()], col.line = "grey")
                   panel.abline(h = 0, col.line = "black")
                   d=tmp[subscripts,]
                   panel.xyplot(d$lag, d$acf,
                                col=ifelse(abs(d$acf)> ci[panel.number()] & d$lag > 0,
                                             "indianred", "steelblue2"),
                                ...)
                 },
                 xlab="lag", ylab="autocorrelation",
                 layout=c(4,1,1))
print(theplot)

```

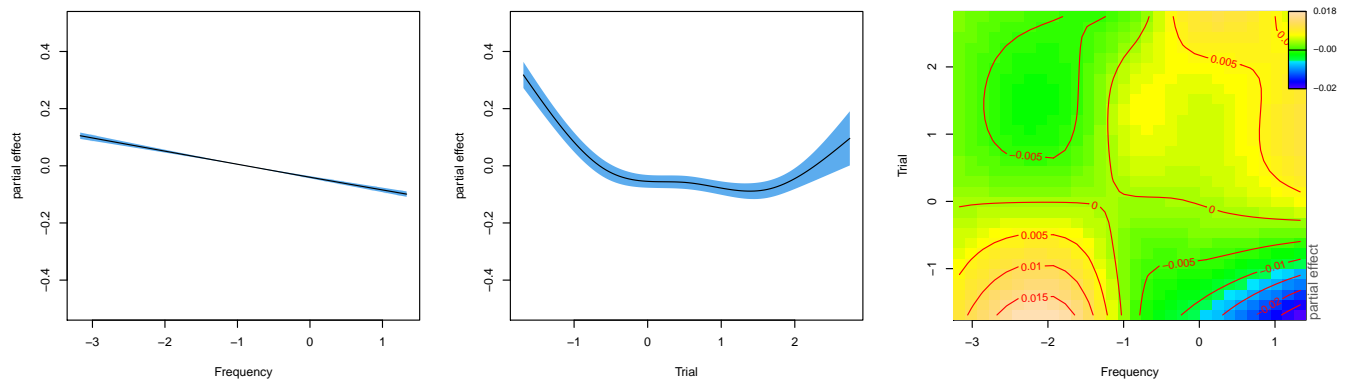
Figure 15.



5.5 Figure 16

This takes about 10 minutes to complete.

```
par(mfrow=c(1,3),oma=rep(0,4))
plot(poems.gam2, select=1, rug=FALSE,
     scheme=1, shade.col="steelblue2",
     ylab="partial effect", xlab="Frequency")
plot(poems.gam2, select=2, rug=FALSE,
     scheme=1, shade.col="steelblue2",
     ylab="partial effect", xlab="Trial")
pvisgam(poems.gam2, select=3, view=c("Fre", "TrialSc"),
        xlab="Frequency", ylab="Trial", main=" ")
```



5.6 Figure 17 and corresponding LMMs

First, we fit a sequence of models with more complex by-subject random effects structure, and show that by-subject random slopes for frequency receive good support.

```
# by-subject random intercepts
poems.lmer1 = lmer(Lrt ~ FreSc * TrialSc +
  (1|Poem) +
  (1|Subject),
  data = poems, REML=FALSE)

# by-subject random intercepts and by-subject random slopes for trial
poems.lmer2 = lmer(Lrt ~ FreSc * TrialSc +
  (1|Poem) +
  (1+TrialSc|Subject),
  data = poems, REML=FALSE)

# by-subject random intercepts and by-subject random slopes for trial
# and in addition by-subject random slopes for frequency
poems.lmer3 = lmer(Lrt ~ FreSc * TrialSc +
  (1|Poem) +
  (1+FreSc+TrialSc|Subject),
  data = poems, REML=FALSE)

# a simpler model without correlation parameters for frequency
poems.lmer4 = lmer(Lrt ~ FreSc * TrialSc +
  (1|Poem) +
  (1+TrialSc|Subject) +
  (0+FreSc|Subject),
  data = poems, REML=FALSE)

# compare the first three models
anova(poems.lmer1, poems.lmer2, poems.lmer3)

Data: poems
Models:
poems.lmer1: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 | Subject)
poems.lmer2: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 + TrialSc | Subject)
poems.lmer3: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 + FreSc + TrialSc | Subject)
      Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
poems.lmer1  7 172200 172274 -86093   172186
poems.lmer2  9 152934 153029 -76458   152916 19270.2      2 < 2.2e-16
poems.lmer3 12 150478 150605 -75227   150454  2461.7      3 < 2.2e-16
```

```

# and test for the correlation parameters for frequency
anova(poems.lmer4, poems.lmer3)

Data: poems
Models:
poems.lmer4: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 + TrialSc | Subject) +
poems.lmer4:      (0 + FreSc | Subject)
poems.lmer3: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 + FreSc + TrialSc | Subject)
              Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
poems.lmer4 10 150608 150713 -75294 150588
poems.lmer3 12 150478 150605 -75227 150454 133.66      2 < 2.2e-16

# note that the summary of poems.lmer3 indicates the frequency
# correlational structure is carried almost exclusively by
# the frequency by intercept correlation
print(summary(poems.lmer3), corr=FALSE)

Linear mixed model fit by maximum likelihood ['lmerMod']
Formula: Lrt ~ FreSc * TrialSc + (1 | Poem) + (1 + FreSc + TrialSc | Subject)
Data: poems

      AIC      BIC  logLik deviance df.resid
150478.3 150604.6 -75227.1 150454.3   275984

Scaled residuals:
    Min      1Q  Median      3Q      Max
-5.9261 -0.6089 -0.1266  0.4605  5.6860

Random effects:
Groups   Name              Variance Std.Dev. Corr
Subject (Intercept) 0.059408 0.24374
        FreSc      0.001172 0.03424 -0.61
        TrialSc     0.009589 0.09792  0.01  0.04
Poem    (Intercept) 0.002528 0.05028
Residual              0.099396 0.31527
Number of obs: 275996, groups: Subject, 326; Poem, 87

Fixed effects:
              Estimate Std. Error t value
(Intercept)   6.0389850  0.0145582   414.8
FreSc        -0.0525595  0.0019960   -26.3
TrialSc       -0.0782901  0.0054780   -14.3
FreSc:TrialSc  0.0038961  0.0006195     6.3

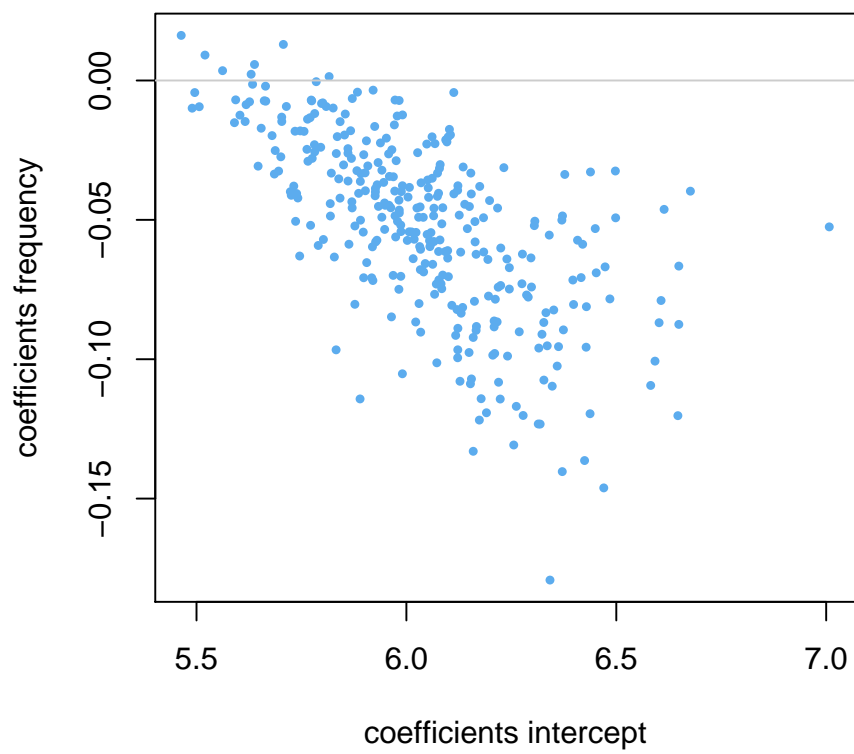
```

Figure 17 is obtained with the following code, extracting the coefficients from `poems.lmer3`:

```

coefs = coef(poems.lmer3)[[1]]
plot(coefs[,1], coefs[,2],
     xlab="coefficients intercept", ylab="coefficients frequency",
     pch=19, cex=0.5, col="steelblue2")
abline(h=0, col="gray80")

```

5.7 Figure 18

```
par(mfrow=c(1,2))
plot(poems.gam, select=1, rug=FALSE,
      xlab="Frequency", ylab="Trial", main=" ")
pvisgam(poems.gam, select=1, view=c("Fre", "TrialSc"),
         xlab="Frequency", ylab="Trial", main=" ")
```

