Supplemental Material

load data

```
load('data2.rda')
str(data2)
```

```
## 'data.frame':    40890 obs. of  20 variables:
##  $ X                   : int  1 12 14 15 16 2 3 4 5 6 ...
##  $ Subject             : Factor w/ 55 levels "101","102","103",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AgeMonths           : int  115 115 115 115 115 115 115 115 115 115 ...
##  $ Sentence            : Factor w/ 54 levels "Welche Ärzte schägt der Koch?",..: 1 1 1 1 1 1 1 1 1 1
##  $ TypeOfCue           : Factor w/ 3 levels "AgCa","Agre",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ nrCue               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EarlyAvailabilityCue: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Condition           : Factor w/ 9 levels "AgCaObj","AgCaPas",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ TypeOfQuestion      : Factor w/ 3 levels "object","passive",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Timebin200          : int  100 300 500 700 900 1100 1300 1500 1700 1900 ...
##  $ ACC                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Group               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ nrRed               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nrYellow            : int  60 60 60 60 60 60 60 60 60 60 ...
##  $ TDDiff              : int  -60 -60 -60 -60 -60 -60 -60 -60 -60 -60 ...
##  $ SubjectQuestion     : Factor w/ 165 levels "101.object","102.object",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ TypeOfQuestionO     : Ord.factor w/ 3 levels "object"<"passive"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ seqinv              : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ SeqStart            : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
##  $ QuestionGroup       : Factor w/ 6 levels "object.1","passive.1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
sort(names(data2))
```

```
##  [1] "ACC"                  "AgeMonths"            "Condition"
##  [4] "EarlyAvailabilityCue" "Group"                "nrCue"
##  [7] "nrRed"                "nrYellow"             "QuestionGroup"
## [10] "Sentence"             "seqinv"               "SeqStart"
## [13] "Subject"              "SubjectQuestion"      "TDDiff"
## [16] "Timebin200"           "TypeOfCue"            "TypeOfQuestion"
## [19] "TypeOfQuestionO"      "X"
```

Load packages

```
# For GAMMs:
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-22. For overview type 'help("mgcv-package")'.
```

```
# For GAMM interpretation and visualization:
library(itsadug)
```

```
## Loading required package: plotfunctions
```

```
## Loaded package itsadug 2.3 (see 'help("itsadug")' ).
```

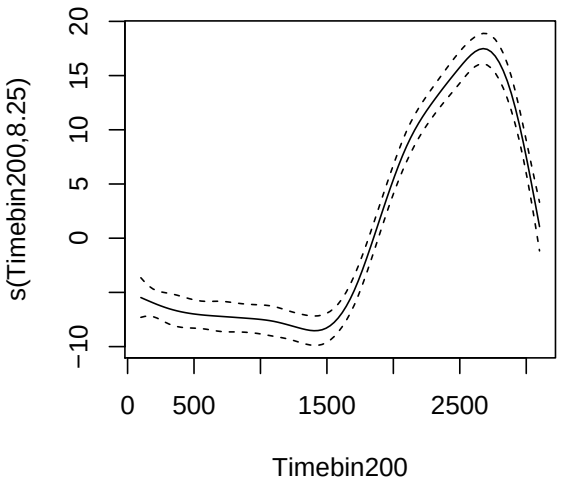
GROUP DIFFERENCES PER QUESTION TYPE

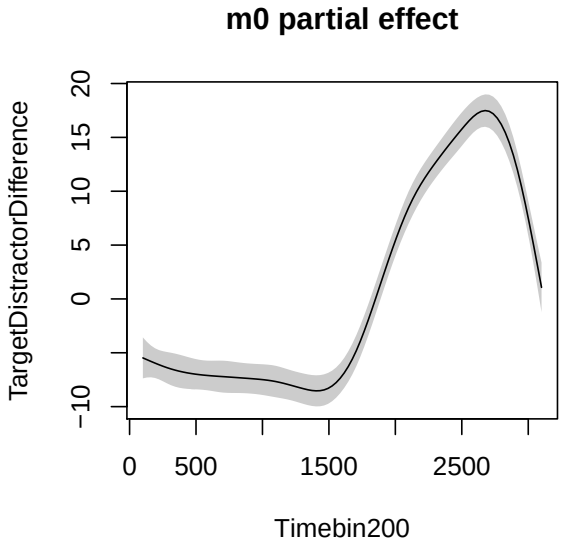
First model for smooth over time

```
m0 = bam(TDDiff ~ s(Timebin200), data=data2, gc.level=2, method='ML')
summary(m0)
```

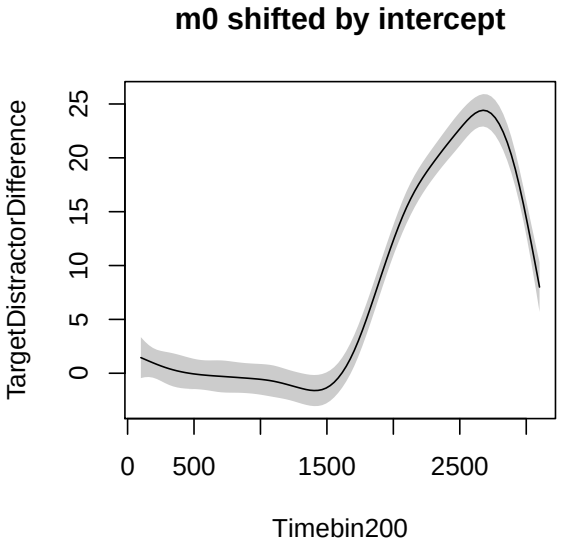
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.9267     0.2544   27.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df   F p-value
## s(Timebin200) 8.247  8.844 149  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0312   Deviance explained = -1.91e+03%
## -ML = 2.1915e+05  Scale est. = 2644.1    n = 40890
```

```
par(mfrow=c(1,2))
plot.gam(m0)
plot(m0,rug=F, shade=T, seWithMean=T, ylab='TargetDistractorDifference', main='m0 partial effect')
```






```
ic=m0$coef["(Intercept)"]
plot(m0,rug=F, shade=T, seWithMean=T, shift = ic, ylab='TargetDistractorDifference', main='m0 shifted by
```



Adding random slopes

```
str(data2)
```

```
## 'data.frame':    40890 obs. of  20 variables:
##  $ X                 : int  1 12 14 15 16 2 3 4 5 6 ...
##  $ Subject           : Factor w/ 55 levels "101","102","103",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AgeMonths         : int  115 115 115 115 115 115 115 115 115 115 ...
##  $ Sentence          : Factor w/ 54 levels "Welche Ärzte schägt der Koch?",..: 1 1 1 1 1 1 1 1 1 1
```

```
##  $ TypeOfCue           : Factor w/ 3 levels "AgCa","Agre",..: 1 1 1 1 1 1 1 1 1 ...
##  $ nrCue               : int  1 1 1 1 1 1 1 1 1 ...
##  $ EarlyAvailabilityCue: int  0 0 0 0 0 0 0 0 0 ...
##  $ Condition           : Factor w/ 9 levels "AgCaObj","AgCaPas",..: 1 1 1 1 1 1 1 1 1 ...
##  $ TypeOfQuestion      : Factor w/ 3 levels "object","passive",..: 1 1 1 1 1 1 1 1 1 ...
##  $ Timebin200          : int  100 300 500 700 900 1100 1300 1500 1700 1900 ...
##  $ ACC                 : int  1 1 1 1 1 1 1 1 1 ...
##  $ Group               : int  1 1 1 1 1 1 1 1 1 ...
##  $ nrRed               : int  0 0 0 0 0 0 0 0 0 ...
##  $ nrYellow            : int  60 60 60 60 60 60 60 60 60 ...
##  $ TDDiff              : int  -60 -60 -60 -60 -60 -60 -60 -60 -60 -60 ...
##  $ SubjectQuestion     : Factor w/ 165 levels "101.object","102.object",..: 1 1 1 1 1 1 1 1 1 ...
##  $ TypeOfQuestionO     : Ord.factor w/ 3 levels "object"<"passive"<..: 1 1 1 1 1 1 1 1 1 ...
##  $ seqinv              : int  0 1 1 1 1 1 1 1 1 ...
##  $ SeqStart            : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
##  $ QuestionGroup       : Factor w/ 6 levels "object.1","passive.1",..: 1 1 1 1 1 1 1 1 1 ...
```

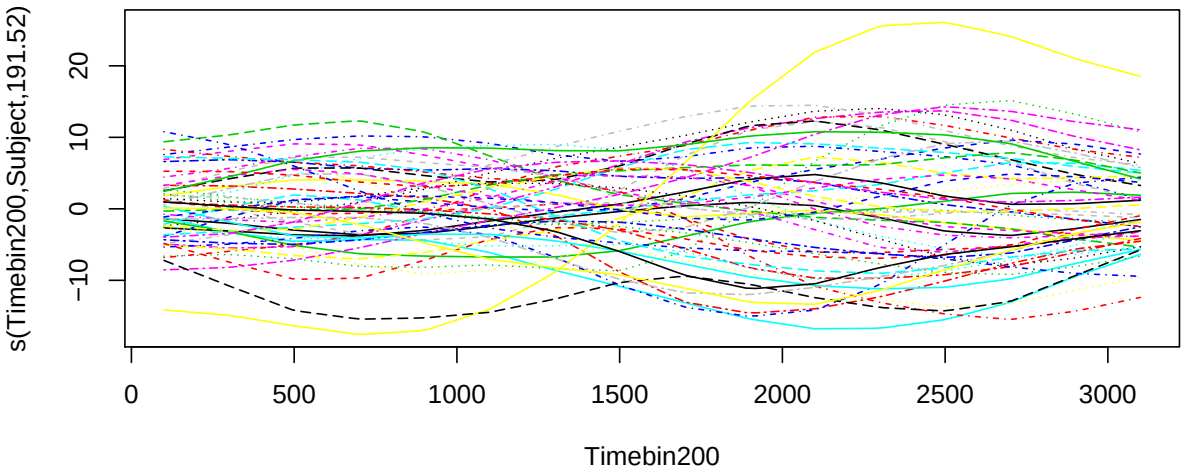
```
data2$Subject <- as.factor(data2$Subject)
m1 = bam(TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs='fs', m=1), data=data2, gc.level=2, method=
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
summary(m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs = "fs", m = 1)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.9299     0.7345   9.435   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                            edf  Ref.df      F p-value
## s(Timebin200)            8.198   8.765 41.061  <2e-16 ***
## s(Timebin200,Subject) 191.520 494.000  1.804  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0518   Deviance explained = -1.11e+03%
## -ML = 2.1888e+05  Scale est. = 2587.7    n = 40890
```

```
par(mfrow=c(1,1))
plot(m1, select=2)
```




```
# comparing models
m0a= bam(TDDiff ~ s(Timebin200), data=data2, gc.level=2, method='fREML')
m1a= bam(TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs='fs', m=1), data=data2, gc.level=2, method=
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(m0a,m1a)
```

```
## m0a: TDDiff ~ s(Timebin200)
##
## m1a: TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs = "fs", m = 1)
##
## Chi-square test of fREML scores
## -----
##    Model    Score Edf Difference    Df  p.value Sig.
## 1    m0a 219143.5   3
## 2    m1a 218877.3   5    266.297 2.000  < 2e-16  ***
##
## AIC difference: 689.16, model m1a has lower AIC.
```

including conditions

```
m2 = bam(TDDiff ~ s(Timebin200, by=TypeOfQuestion) + TypeOfQuestion + s(Timebin200, Subject, bs='fs', m=
summary(m2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200, by = TypeOfQuestion) + TypeOfQuestion +
##     s(Timebin200, Subject, bs = "fs", m = 1)
##
```

```
## Parametric coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.5480     0.8102   1.911   0.0561 .
## TypeOfQuestionpassive     6.1573     0.6092  10.107   <2e-16 ***
## TypeOfQuestionsubject    10.3092     0.6076  16.967   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                           edf  Ref.df     F p-value
## s(Timebin200):TypeOfQuestionobject      7.303   8.265 24.40  <2e-16 ***
## s(Timebin200):TypeOfQuestionpassive     7.930   8.673 45.03  <2e-16 ***
## s(Timebin200):TypeOfQuestionsubject     6.428   7.522 35.27  <2e-16 ***
## s(Timebin200,Subject)                 193.856 494.000  1.85  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0678   Deviance explained = -880%
## -ML = 2.1856e+05  Scale est. = 2544.2    n = 40890
```

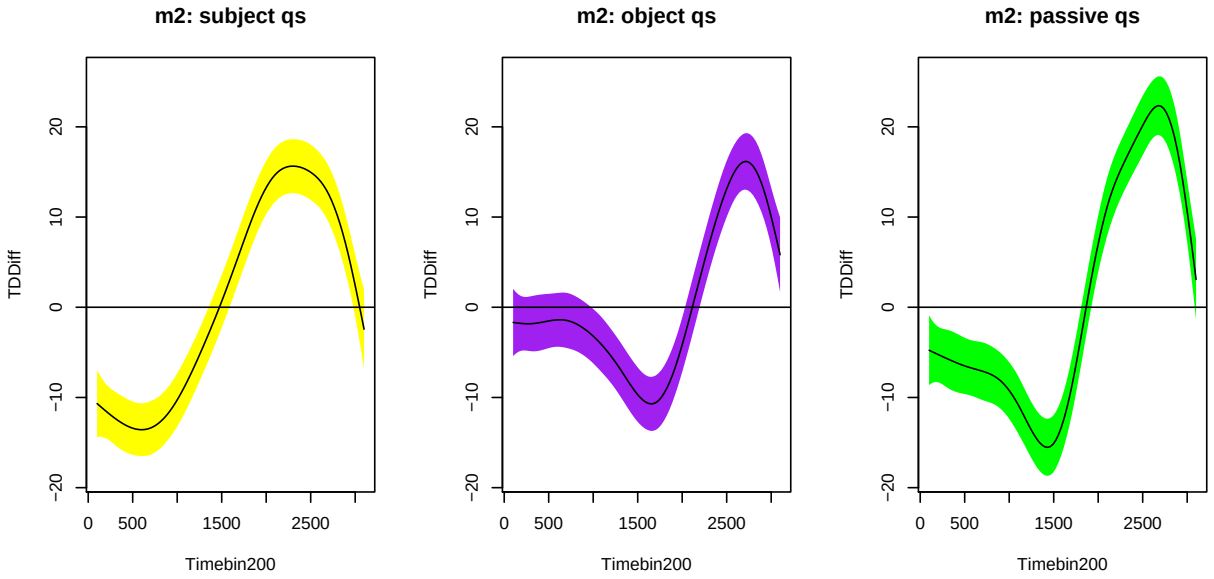
```
compareML(m1,m2)
```

```
## m1: TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs = "fs", m = 1)
##
## m2: TDDiff ~ s(Timebin200, by = TypeOfQuestion) + TypeOfQuestion +
##     s(Timebin200, Subject, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##   Model    Score Edf Difference    Df  p.value Sig.
## 1    m1 218880.4   5
## 2    m2 218557.8  11    322.551 6.000  < 2e-16  ***
##
## AIC difference: 673.91, model m2 has lower AIC.
```

```
save(m2, file = "m2.rda" )
```

visualizing both patterns: partial effects

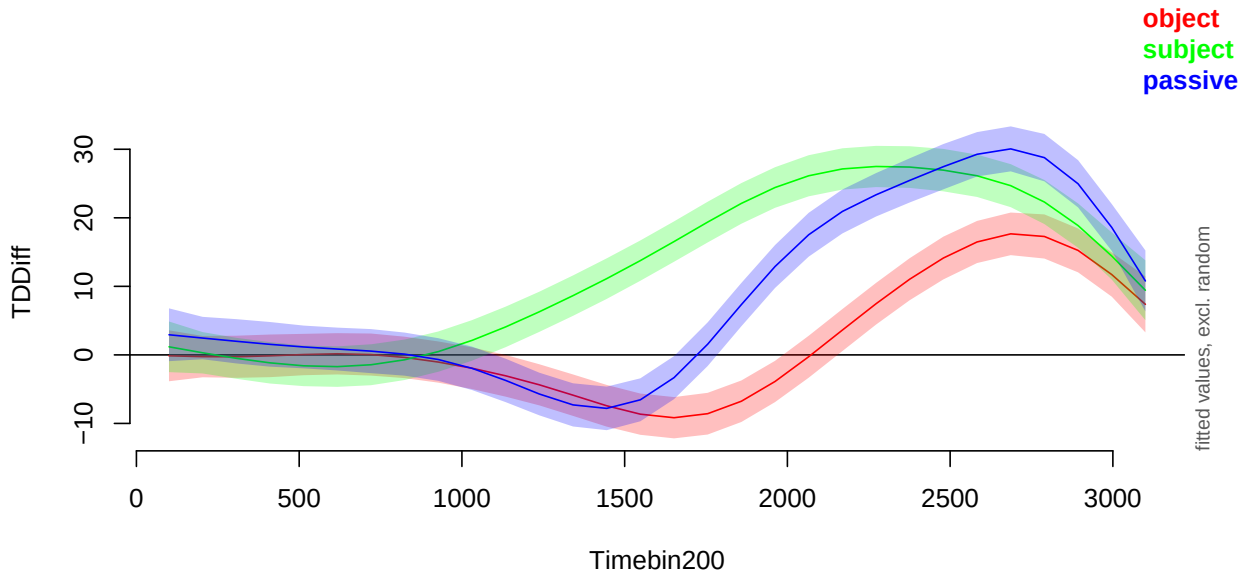
```
par(mfrow=c(1,3))
plot(m2, select=3, shade=T, rug=F, shade.col='yellow', seWithMean=T, ylab='TDDiff', main='m2: subject qs
plot(m2, select=1, shade=T, rug=F, shade.col='purple', seWithMean=T, ylab='TDDiff', main='m2: object qs
plot(m2, select=2, shade=T, rug=F, shade.col='green', seWithMean=T, ylab='TDDiff', main='m2: passive qs
```



visualizing both patterns: complete effects

```
par(mfrow=c(1,1))
plot_smooth(m2, view="Timebin200", plot_all="TypeOfQuestion", rm.ranef=TRUE)
```

```
## Summary:
##  * TypeOfQuestion : factor; set to the value(s): object, passive, subject.
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * Subject : factor; set to the value(s): 337. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```



Adding lines random effects per subject per condition

```
data2$SubjectQuestion = interaction(data2$Subject,data2$TypeOfQuestion)

m3 = bam(TDDiff ~ s(Timebin200, by=TypeOfQuestion) + TypeOfQuestion + s(Timebin200, SubjectQuestion, bs=
summary(m3)
```

```
compareML(m2, m3)
```

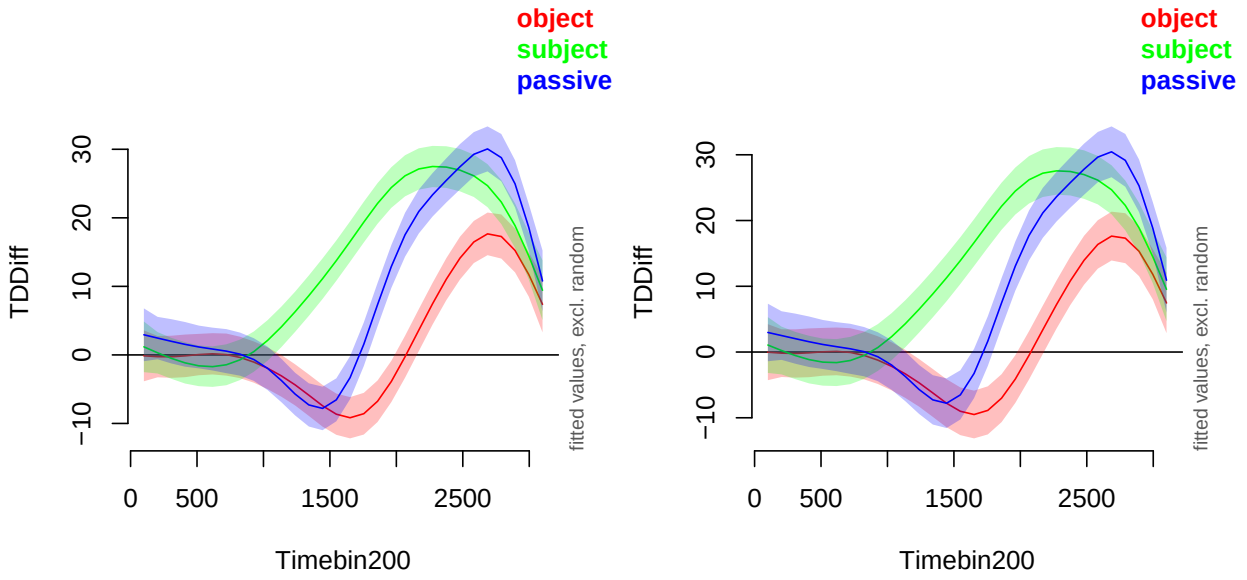
```
## m2: TDDiff ~ s(Timebin200, by = TypeOfQuestion) + TypeOfQuestion +
##     s(Timebin200, Subject, bs = "fs", m = 1)
##
## m3: TDDiff ~ s(Timebin200, by = TypeOfQuestion) + TypeOfQuestion +
##     s(Timebin200, SubjectQuestion, bs = "fs", m = 1)
##
## Model m3 preferred: lower ML score (178.077), and equal df (0.000).
## -----
##   Model    Score Edf Difference    Df
## 1    m2 218557.8  11
## 2    m3 218379.7  11   -178.077 0.000
##
## AIC difference: 559.69, model m3 has lower AIC.
```

```
save(m3, file = "m3.rda" )

par(mfrow=c(1,2))
plot_smooth(m2, view="Timebin200", plot_all="TypeOfQuestion", rm.ranef=TRUE)
```

```
## Summary:
##  * TypeOfQuestion : factor; set to the value(s): object, passive, subject.
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * Subject : factor; set to the value(s): 337. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

```
plot_smooth(m3, view="Timebin200", plot_all="TypeOfQuestion", rm.ranef=TRUE)
```

```
## Summary:
##  * TypeOfQuestion : factor; set to the value(s): object, passive, subject.
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * SubjectQuestion : factor; set to the value(s): 102.object. (Might be canceled as random effect, ch
##  * NOTE : The following random effects columns are canceled: s(Timebin200,SubjectQuestion)
##
```




```
compareML(m2,m3)
```

```
## m2: TDDiff ~ s(Timebin200, by = TypeOfQuestion) + TypeOfQuestion +
##     s(Timebin200, Subject, bs = "fs", m = 1)
##
## m3: TDDiff ~ s(Timebin200, by = TypeOfQuestion) + TypeOfQuestion +
##     s(Timebin200, SubjectQuestion, bs = "fs", m = 1)
##
## Model m3 preferred: lower ML score (178.077), and equal df (0.000).
## -----
##   Model    Score Edf Difference    Df
## 1    m2 218557.8  11
## 2    m3 218379.7  11   -178.077 0.000
##
## AIC difference: 559.69, model m3 has lower AIC.
```

Seperating intercept and smooth difference (binary factor is not centered, an ordered factor is)

```
data2$TypeOfQuestionO = as.ordered(data2$TypeOfQuestion)
contrasts(data2$TypeOfQuestionO) = 'contr.treatment'
m3c = bam(TDDiff ~ s(Timebin200, by=TypeOfQuestionO) + TypeOfQuestionO + s(Timebin200, SubjectQuestion,
summary(m3c)
save(m3c, file = "m3c.rda" )
```

Testing for group effects and question effects: Combined factor of TypeOfQuestion and Group

```
data2$QuestionGroup = interaction(data2$TypeOfQuestion,data2$Group)
levels(data2$QuestionGroup) # show levels
```

```r
contrasts(data2$QuestionGroup) = 'contr.treatment'
m4 = bam(TDDiff ~ s(Timebin200, by=QuestionGroup) + QuestionGroup + s(Timebin200, SubjectQuestion, bs='
```

```r
summary(m4)
```

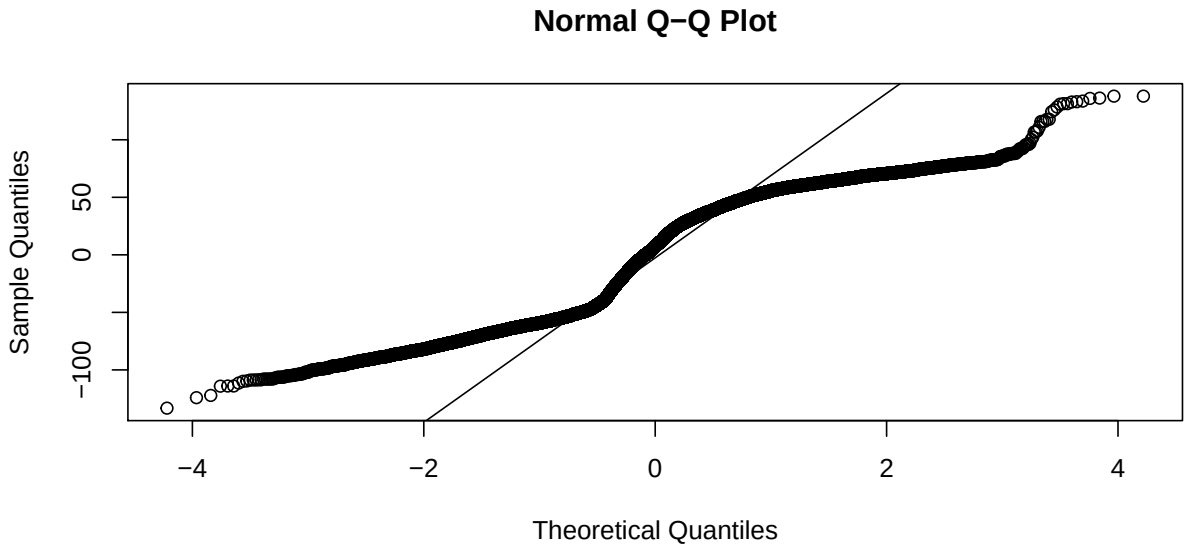
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200, by = QuestionGroup) + QuestionGroup +
##     s(Timebin200, SubjectQuestion, bs = "fs", m = 1)
##
## Parametric coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.035      1.274   2.382 0.017201 *
## QuestionGrouppassive.1     5.887      1.812   3.249 0.001159 **
## QuestionGroupsubject.1     9.680      1.808   5.354 8.63e-08 ***
## QuestionGroupobject.3     -4.322      2.066  -2.092 0.036456 *
## QuestionGrouppassive.3     2.951      2.070   1.426 0.153964
## QuestionGroupsubject.3     7.578      2.072   3.657 0.000256 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                          edf   Ref.df      F  p-value
## s(Timebin200):QuestionGroupobject.1    7.049    8.020 15.640  < 2e-16 ***
## s(Timebin200):QuestionGrouppassive.1   7.693    8.504 23.044  < 2e-16 ***
## s(Timebin200):QuestionGroupsubject.1   5.787    6.827 19.004  < 2e-16 ***
## s(Timebin200):QuestionGroupobject.3    5.171    6.151  5.786 4.30e-06 ***
## s(Timebin200):QuestionGrouppassive.3   6.484    7.521 13.219  < 2e-16 ***
## s(Timebin200):QuestionGroupsubject.3   4.845    5.788  8.782 4.06e-09 ***
## s(Timebin200,SubjectQuestion)        481.677 1479.000  1.167  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.087   Deviance explained = -577%
## -ML = 2.1839e+05  Scale est. = 2491.7    n = 40890
```

```r
save(m4, file = "m4.rda" )
compareML(m3c,m4)
```

```
## m3c: TDDiff ~ s(Timebin200, by = TypeOfQuestion0) + TypeOfQuestion0 +
##     s(Timebin200, SubjectQuestion, bs = "fs", m = 1)
##
## m4: TDDiff ~ s(Timebin200, by = QuestionGroup) + QuestionGroup +
##     s(Timebin200, SubjectQuestion, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##   Model    Score Edf Difference     Df   p.value Sig.
## 1   m3c 218440.4   9
## 2    m4 218394.2  20     46.177 11.000 5.758e-15  ***
##
## AIC difference: 50.31, model m4 has lower AIC.
```

Accounting for autocorrelation residuals:

```
data2 = data2[order(data2$Subject, data2$Sentence, data2$Timebin200, decreasing=F),]
data2$seqinv <- c(0L, data2$Timebin[-nrow(data2)]  <= data2$Timebin[-1])
data2$SeqStart = ifelse(data2$seqinv ==0, TRUE, FALSE)

qqnorm(resid(m4))
qqline(resid(m4))
```




```
rho4 <- start_value_rho(m4)
```

Model with correction for autocorrelation

```
m4c = bam(TDDiff ~ s(Timebin200, by=QuestionGroup) + QuestionGroup + s(Timebin200, SubjectQuestion, bs=
save(m4c, file = "m4c.rda" )
```

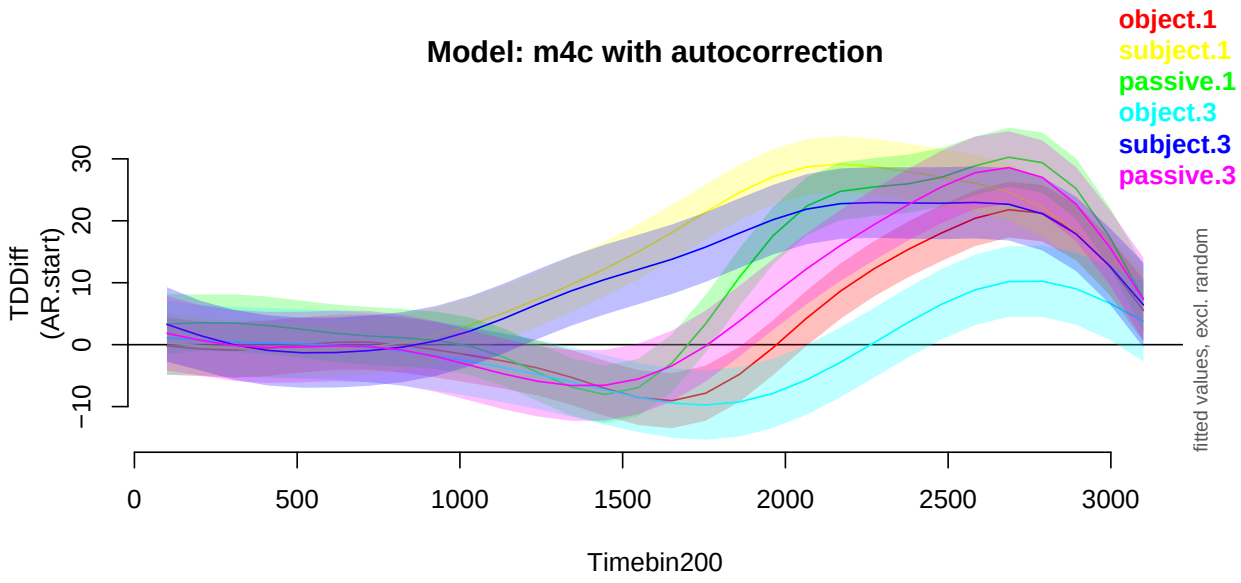
```
summary(m4c)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200, by = QuestionGroup) + QuestionGroup +
##     s(Timebin200, SubjectQuestion, bs = "fs", m = 1)
##
## Parametric coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.914      1.354   2.152 0.031372 *
## QuestionGrouppassive.1      5.826      1.954   2.981 0.002871 **
## QuestionGroupsubject.1      9.584      1.936   4.951  7.4e-07 ***
## QuestionGroupobject.3      -3.784      2.211  -1.711 0.087054 .
```

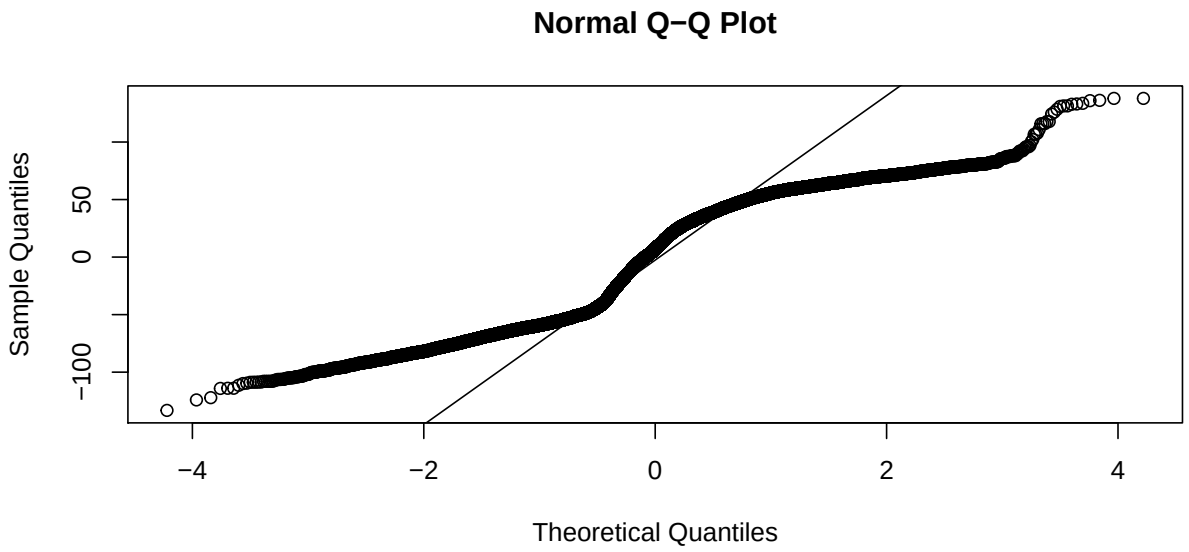
```
## QuestionGrouppassive.3    3.003      2.212   1.358 0.174530
## QuestionGroupsubject.3    7.566      2.221   3.407 0.000659 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                          edf   Ref.df      F  p-value
## s(Timebin200):QuestionGroupobject.1    8.264    8.804 18.280  < 2e-16 ***
## s(Timebin200):QuestionGrouppassive.1   8.599    8.923 28.501  < 2e-16 ***
## s(Timebin200):QuestionGroupsubject.1   7.395    8.349 16.958  < 2e-16 ***
## s(Timebin200):QuestionGroupobject.3    6.761    7.878  5.461 8.79e-07 ***
## s(Timebin200):QuestionGrouppassive.3   7.888    8.636 13.883  < 2e-16 ***
## s(Timebin200):QuestionGroupsubject.3   7.097    8.137  7.339 4.29e-10 ***
## s(Timebin200,SubjectQuestion)        313.106 1479.000  0.298 6.55e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0719   Deviance explained = -1.24e+03%
## -ML = 2.0178e+05  Scale est. = 2257.9    n = 40890
```

```
par(mfrow=c(1,1))
plot_smooth(m4c, view="Timebin200", plot_all="QuestionGroup", rm.ranef=TRUE, main= 'Model: m4c with aut
```

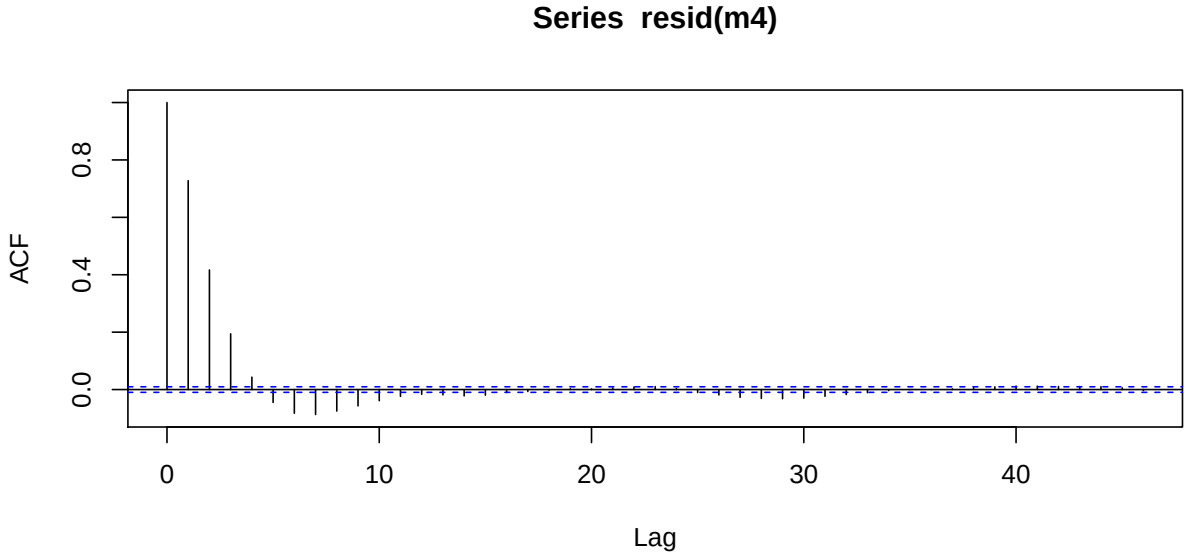
```
## Summary:
##  * QuestionGroup : factor; set to the value(s): object.1, object.3, passive.1, passive.3, subject.1,
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * SubjectQuestion : factor; set to the value(s): 102.object. (Might be canceled as random effect, cl
##  * NOTE : The following random effects columns are canceled: s(Timebin200,SubjectQuestion)
##
```




```
qqnorm(resid(m4))
qqline(resid(m4))
```




```
acf(resid(m4))
```



Tables and plots

```
par(mfrow=c(1,3), cex=1.1,mar=c(4,3,4,1))

#Subject, CI NH Children
newd <- get_difference(m4c, comp=list(QuestionGroup=c('subject.3', 'subject.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
```

```
##  * SubjectQuestion : factor; set to the value(s): 102.object. (Might be canceled as random effect, cl
##  * NOTE : The following random effects columns are canceled: s(Timebin200,SubjectQuestion)
##
```

```
# label in data frame newd which rows are significant:
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
# setup plot:
emptyPlot(c(0, 3000), c(-30,10), h0=0, v=c(1000,2500))
mtext('Subject questions', side=3, line=1)
mtext('Timebin (ms)', side=1, line=2)
# fill significant area:
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference > 0,
                                                           newd$difference-newd$CI,newd$difference+newd$
                                                            col='red', alpha=1, density=30)
# plot line + CI:
with(newd, plot_error(Timebin200, difference, CI))

#Object, CI NH Children
newd <- get_difference(m4c, comp=list(QuestionGroup=c('object.3', 'object.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

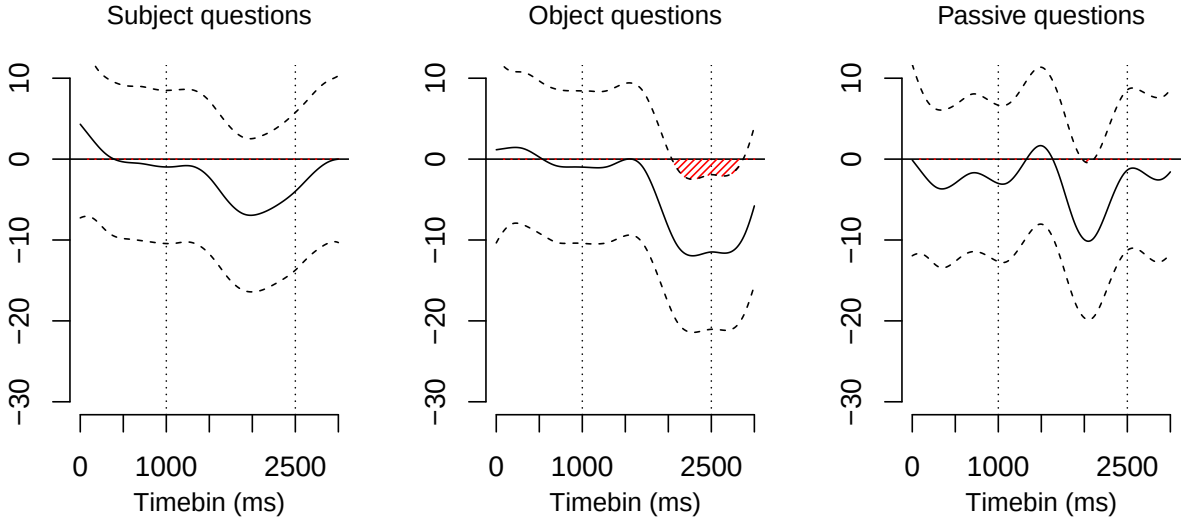
```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * SubjectQuestion : factor; set to the value(s): 102.object. (Might be canceled as random effect, cl
##  * NOTE : The following random effects columns are canceled: s(Timebin200,SubjectQuestion)
##
```

```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-30,10), h0=0, v=c(1000,2500))
mtext('Object questions', side=3, line=1)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference > 0,
                                                           newd$difference-newd$CI, newd$difference+newd
                                                            col='red', alpha=1, density=30)
with(newd, plot_error(Timebin200, difference, CI))

#Passive, CI NH Children
newd <- get_difference(m4c, comp=list(QuestionGroup=c('passive.3', 'passive.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * SubjectQuestion : factor; set to the value(s): 102.object. (Might be canceled as random effect, cl
##  * NOTE : The following random effects columns are canceled: s(Timebin200,SubjectQuestion)
##
```

```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-30,10), h0=0, v=c(1000,2500))
mtext('Passive questions', side=3, line=1)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference > 0,
                                                           newd$difference-newd$CI, newd$difference+newd
                                                            col='red', alpha=1, density=30)
with(newd, plot_error(Timebin200, difference, CI))

title('Difference plots per Type of Question (children with CI vs. children with NH)', line= -1.5,outer=
```

Difference plots per Type of Question (children with CI vs. children with NH)



GROUP DIFFERENCES PER CUE TYPE FOR OBJECT QUESTIONS

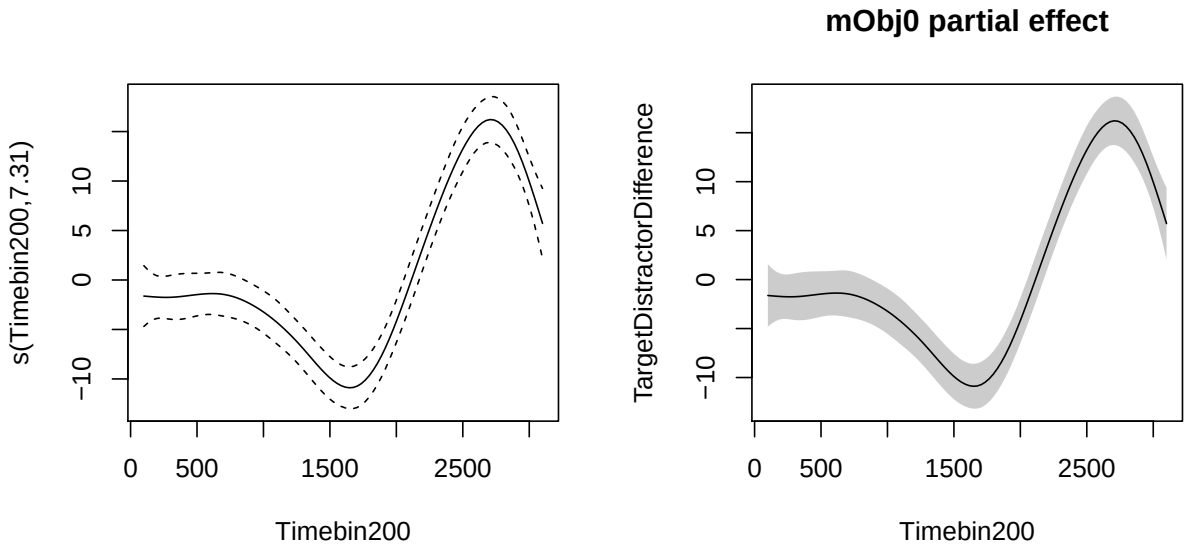
```
dataObj <- data2[data2$TypeOfQuestion == 'object',]
dataObj <- droplevels(dataObj)
```

First model for smooth over time

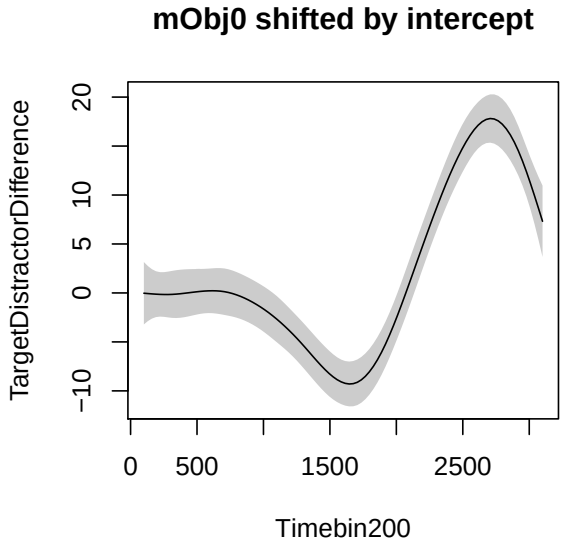
```
mObj0 = bam(TDDiff ~ s(Timebin200), data=dataObj, gc.level=2, method='ML')
summary(mObj0)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.5973     0.4341    3.68 0.000234 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(Timebin200) 7.308  8.308 41.27  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0233   Deviance explained = -3.9e+03%
## -ML =  76925  Scale est. = 2698.4    n = 14325
```

```
par(mfrow=c(1,2))
plot.gam(mObj0)
plot(mObj0,rug=F, shade=T, seWithMean=T, ylab='TargetDistractorDifference', main='mObj0 partial effect')
```




```
ic=mObj0$coef["(Intercept)"]
plot(mObj0,rug=F, shade=T, seWithMean=T, shift = ic, ylab='TargetDistractorDifference', main='mObj0 shi
```



Adding random slopes

```
str(dataObj)
```

```
## 'data.frame':    14325 obs. of  20 variables:
```

```
##  $ X                   : int  1 12 14 15 16 2 3 4 5 6 ...
##  $ Subject             : Factor w/ 55 levels "101","102","103",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AgeMonths           : int  115 115 115 115 115 115 115 115 115 115 ...
##  $ Sentence            : Factor w/ 18 levels "Welche Ärzte schägt der Koch?",..: 1 1 1 1 1 1 1 1 1 1
##  $ TypeOfCue           : Factor w/ 3 levels "AgCa","Agre",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ nrCue               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EarlyAvailabilityCue: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Condition           : Factor w/ 3 levels "AgCaObj","AgreObj",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ TypeOfQuestion      : Factor w/ 1 level "object": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Timebin200          : int  100 300 500 700 900 1100 1300 1500 1700 1900 ...
##  $ ACC                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Group               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ nrRed               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nrYellow            : int  60 60 60 60 60 60 60 60 60 60 ...
##  $ TDDiff              : int  -60 -60 -60 -60 -60 -60 -60 -60 -60 -60 ...
##  $ SubjectQuestion     : Factor w/ 55 levels "101.object","102.object",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ TypeOfQuestionO     : Ord.factor w/ 1 level "object": 1 1 1 1 1 1 1 1 1 1 ...
##  $ seqinv              : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ SeqStart            : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
##  $ QuestionGroup       : Factor w/ 2 levels "object.1","object.3": 1 1 1 1 1 1 1 1 1 1 ...
```

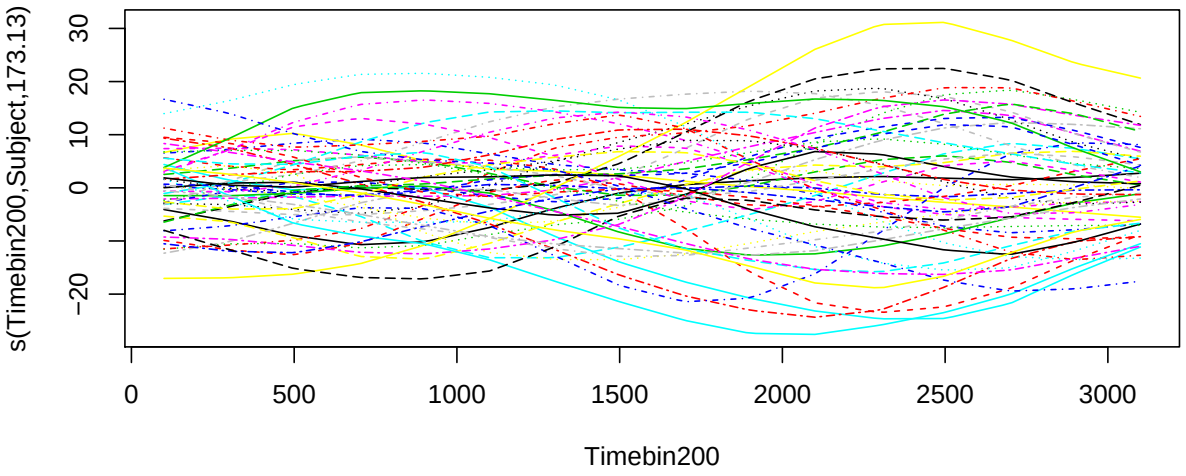
```
mObj1 = bam(TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs='fs', m=1), data=dataObj, gc.level=2, me
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
summary(mObj1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs = "fs", m = 1)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.435      1.048   1.369    0.171
##
## Approximate significance of smooth terms:
##                            edf  Ref.df      F p-value
## s(Timebin200)            7.229   8.158 17.515  <2e-16 ***
## s(Timebin200,Subject) 173.129 494.000  1.321  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.066   Deviance explained = -880%
## -ML =  76759  Scale est. = 2580.6    n = 14325
```

```
par(mfrow=c(1,1))
plot(mObj1, select=2)
```




```
# comparing models
mObj0a= bam(TDDiff ~ s(Timebin200), data=dataObj, gc.level=2, method='fREML')
mObj1a= bam(TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs='fs', m=1), data=dataObj, gc.level=2, me
```

```
## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has
## repeated 1-d smooths of same variable.
```

```
compareML(mObj0a,mObj1a)
```

```
## mObj0a: TDDiff ~ s(Timebin200)
##
## mObj1a: TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs = "fs", m = 1)
##
## Chi-square test of fREML scores
## -----
##    Model    Score Edf Difference    Df  p.value Sig.
## 1 mObj0a 76922.51   3
## 2 mObj1a 76754.94   5    167.570 2.000  < 2e-16  ***
##
## AIC difference: 464.15, model mObj1a has lower AIC.
```

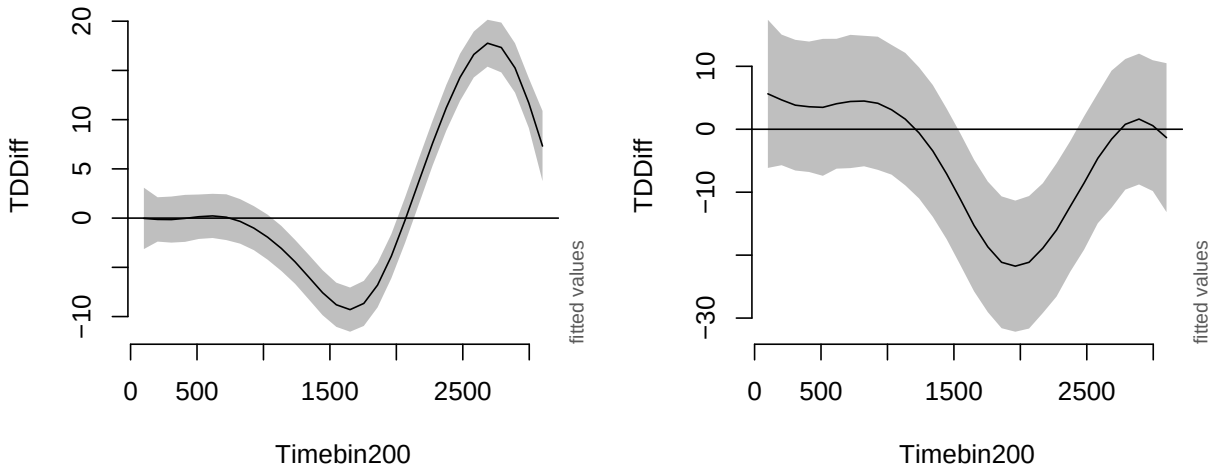
```
par(mfrow=c(1,2))
plot_smooth(mObj0, "Timebin200",eegAxis=F)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
```

```
plot_smooth(mObj1, "Timebin200", eegAxis=F, dropRanef="Subject")
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * Subject : factor; set to the value(s): 102.
## Warning in plot.window(...): "dropRanef" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "dropRanef" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "dropRanef" is
```

```
## not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "dropRanef" is
## not a graphical parameter

## Warning in box(...): "dropRanef" is not a graphical parameter

## Warning in title(...): "dropRanef" is not a graphical parameter
```



including conditions

```
mObj2 = bam(TDDiff ~ s(Timebin200, by=TypeOfCue) + TypeOfCue + s(Timebin200, Subject, bs='fs', m=1), da
summary(mObj2)
```

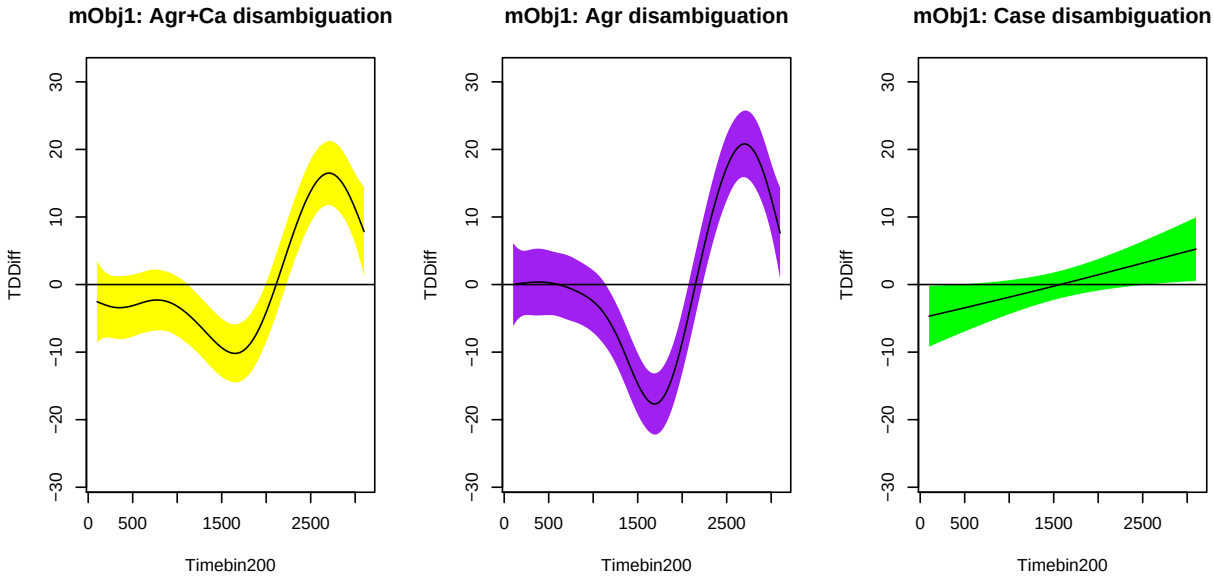
```
compareML(mObj1,mObj2)
```

```
## mObj1: TDDiff ~ s(Timebin200) + s(Timebin200, Subject, bs = "fs", m = 1)
##
## mObj2: TDDiff ~ s(Timebin200, by = TypeOfCue) + TypeOfCue + s(Timebin200,
##     Subject, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##    Model    Score Edf Difference    Df    p.value Sig.
## 1 mObj1 76758.77   5
## 2 mObj2 76727.01  11     31.760 6.000 8.649e-12  ***
##
## AIC difference: 70.61, model mObj2 has lower AIC.
```

```
save(mObj2, file = "mObj2.rda" )
```

visualizing both patterns: partial effects

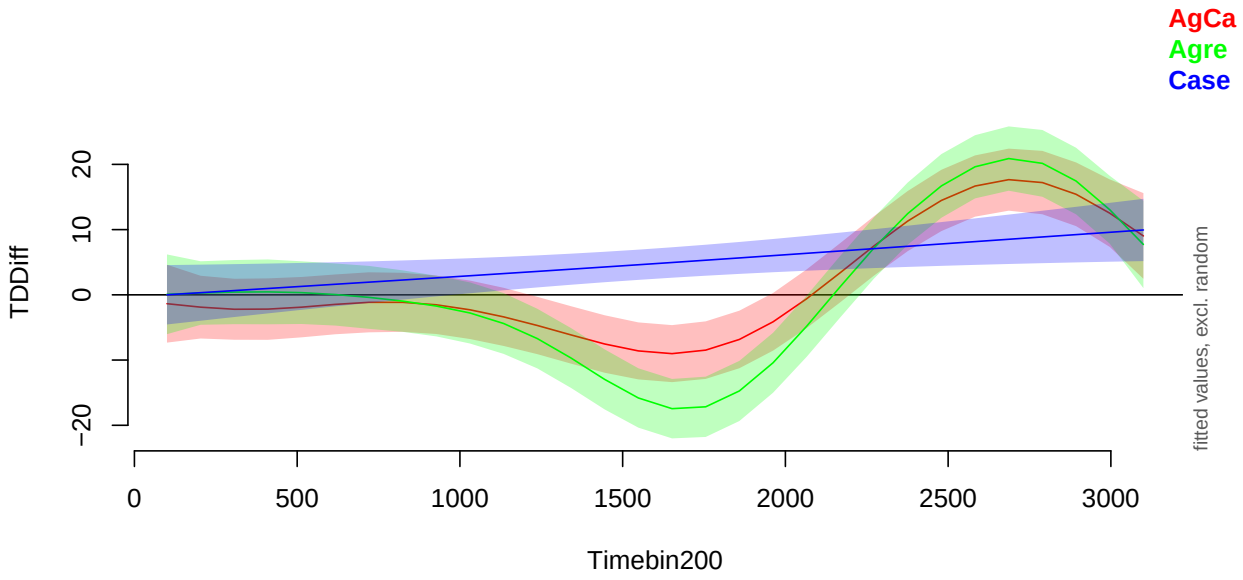
```
par(mfrow=c(1,3))
plot(mObj2, select=1, shade=T, rug=F, shade.col='yellow', seWithMean=T, ylab='TDDiff', main='mObj1: Agr
plot(mObj2, select=2, shade=T, rug=F, shade.col='purple', seWithMean=T, ylab='TDDiff', main='mObj1: Agr
plot(mObj2, select=3, shade=T, rug=F, shade.col='green', seWithMean=T, ylab='TDDiff', main='mObj1: Case
```



visualizing both patterns: complete effects

```
par(mfrow=c(1,1))
plot_smooth(mObj2, view="Timebin200", plot_all="TypeOfCue", rm.ranef=TRUE)
```

```
## Summary:
##  * TypeOfCue : factor; set to the value(s): AgCa, Agre, Case.
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```



Testing for group effects and Cue effects: Combined factor of TypeOfCue and Group

```
dataObj$CueGroup <- interaction(dataObj$TypeOfCue, dataObj$Group)

levels(dataObj$CueGroup) # show levels
contrasts(dataObj$CueGroup) = 'contr.treatment'
mObj3 = bam(TDDiff ~ s(Timebin200, by=CueGroup) + CueGroup + s(Timebin200, Subject, bs='fs', m=1), data=
summary(mObj3)
save(mObj3, file = "mObj3.rda" )
```

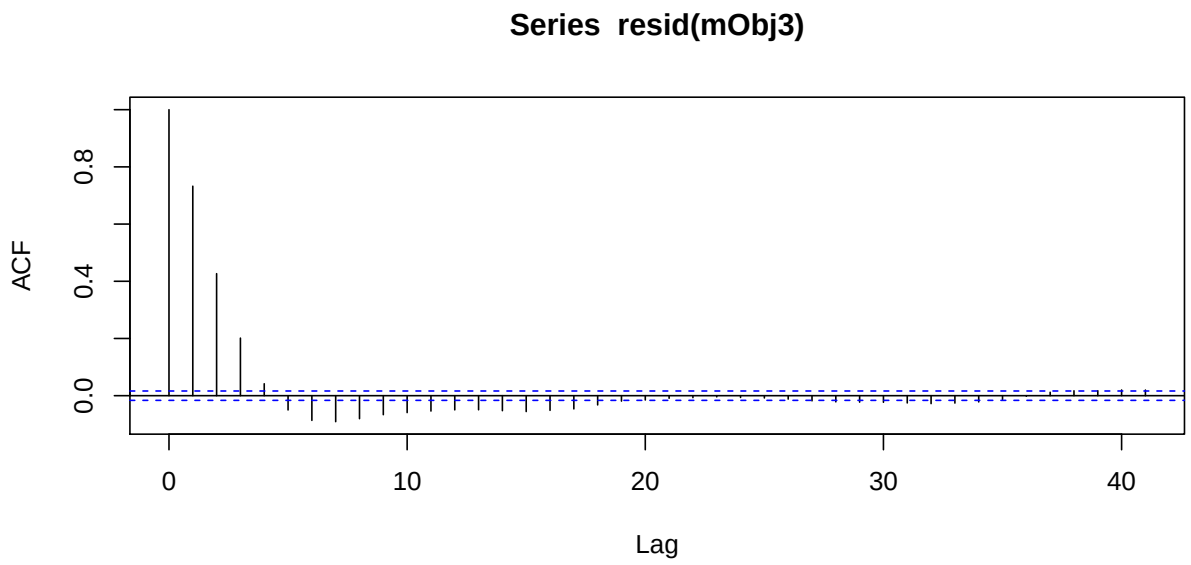
```
compareML(mObj2,mObj3)
```

```
## mObj2: TDDiff ~ s(Timebin200, by = TypeOfCue) + TypeOfCue + s(Timebin200,
##     Subject, bs = "fs", m = 1)
##
## mObj3: TDDiff ~ s(Timebin200, by = CueGroup) + CueGroup + s(Timebin200,
##     Subject, bs = "fs", m = 1)
##
## Chi-square test of ML scores
## -----
##   Model    Score Edf Difference    Df p.value Sig.
## 1 mObj2 76727.01  11
## 2 mObj3 76720.73  20      6.275 9.000   0.184
##
## AIC difference: 5.80, model mObj3 has lower AIC.
```

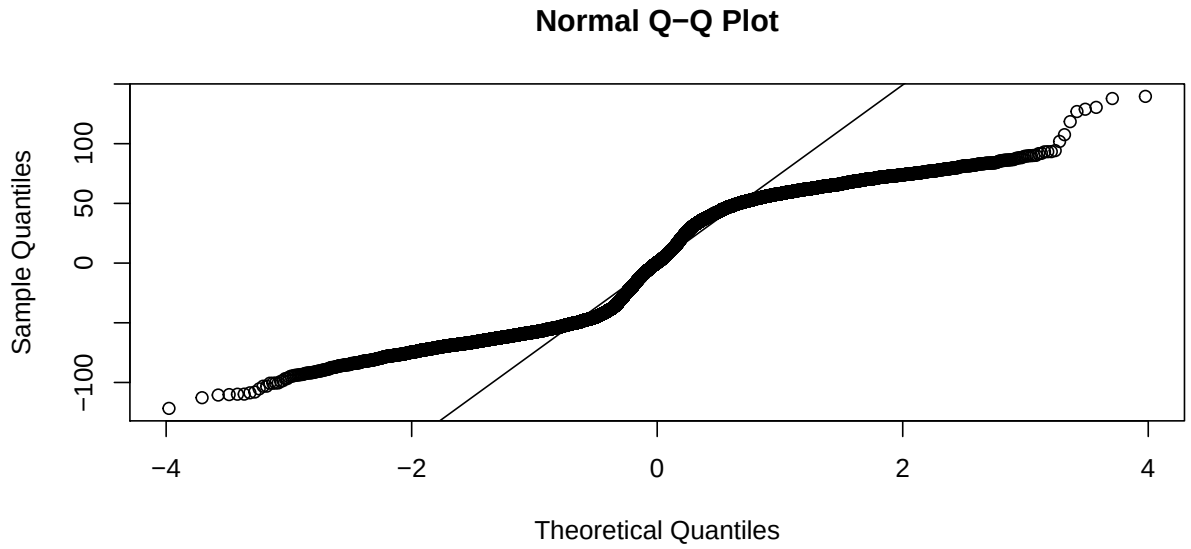
Accounting for autocorrelation residuals:

```
dataObj = dataObj[order(dataObj$Subject, dataObj$Sentence, dataObj$Timebin200, decreasing=F),]
dataObj$seqinv <- c(0L, dataObj$Timebin[-nrow(dataObj)]  <= dataObj$Timebin[-1])
```

```
dataObj$SeqStart = ifelse(dataObj$seqinv ==0, TRUE, FALSE)

acf(resid(mObj3))
```




```
qqnorm(resid(mObj3))
qqline(resid(mObj3))
```




```
rhoObj3 <- start_value_rho(mObj3)
```

Model with correction for autocorrelation

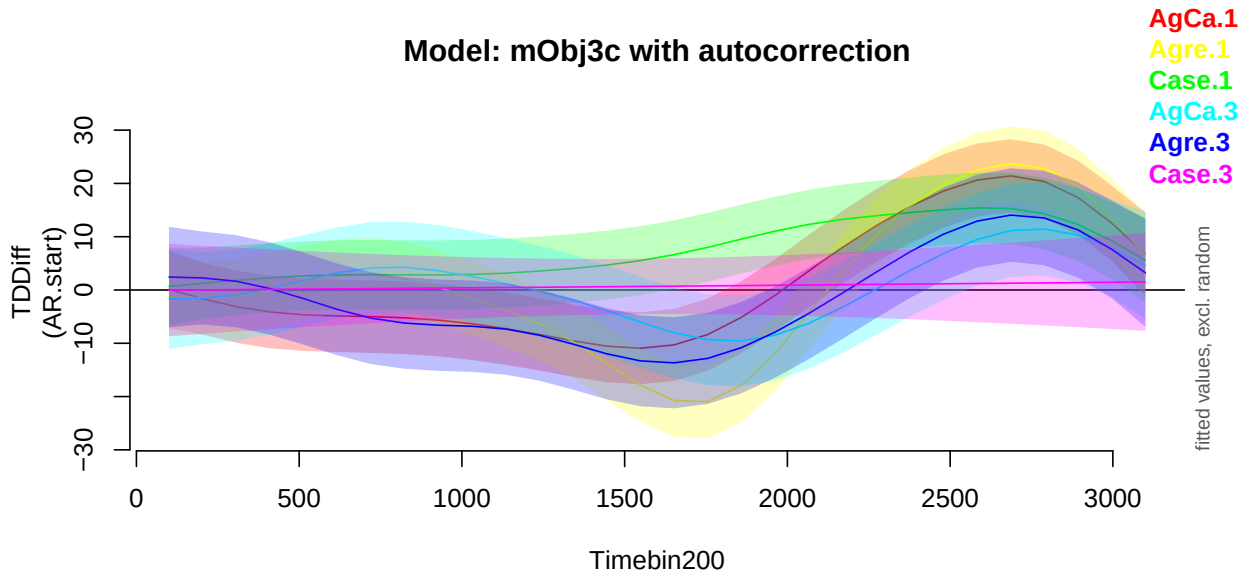
```
mObj3c = bam(TDDiff ~ s(Timebin200, by=CueGroup) + CueGroup + s(Timebin200, Subject, bs='fs', m=1), data
save(mObj3c, file = "mObj3c.rda" )
```

```
summary(mObj3c)
```

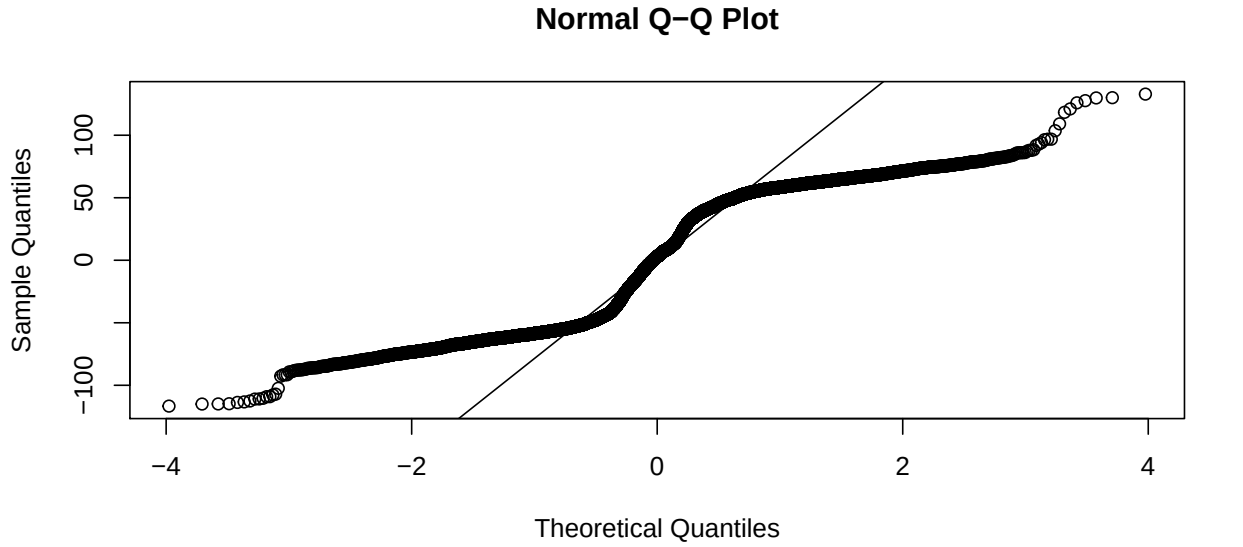
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## TDDiff ~ s(Timebin200, by = CueGroup) + CueGroup + s(Timebin200,
##     Subject, bs = "fs", m = 1)
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.9476     2.1473   0.441    0.659
## CueGroupAgre.1  -0.4172     2.8099  -0.148    0.882
## CueGroupCase.1   6.0618     2.8098   2.157    0.031 *
## CueGroupAgCa.3  -0.3702     3.4943  -0.106    0.916
## CueGroupAgre.3  -2.6726     3.4812  -0.768    0.443
## CueGroupCase.3  -0.2631     3.4049  -0.077    0.938
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                    edf  Ref.df      F  p-value
## s(Timebin200):CueGroupAgCa.1     7.102   8.232  8.271 1.68e-11 ***
## s(Timebin200):CueGroupAgre.1     8.060   8.775 15.489  < 2e-16 ***
## s(Timebin200):CueGroupCase.1     5.297   6.606  2.356  0.02018 *
## s(Timebin200):CueGroupAgCa.3     6.215   7.525  3.131  0.00195 **
## s(Timebin200):CueGroupAgre.3     6.442   7.727  4.375 4.66e-05 ***
## s(Timebin200):CueGroupCase.3     1.009   1.018  0.037  0.84323
## s(Timebin200,Subject)          117.799 493.000  0.347 1.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0588   Deviance explained = -1.38e+03%
## -ML =  70810  Scale est. = 2322.2    n = 14325
```

```
par(mfrow=c(1,1))
plot_smooth(mObj3c, view="Timebin200", plot_all="CueGroup", rm.ranef=TRUE, main= 'Model: mObj3c with au
```

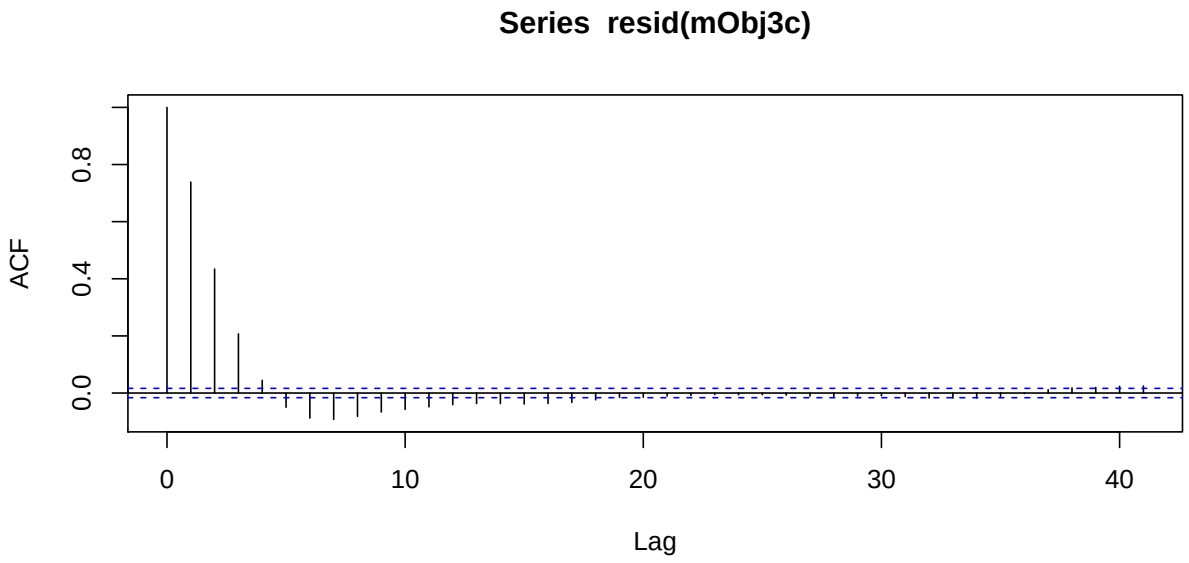
```
## Summary:
##  * CueGroup : factor; set to the value(s): AgCa.1, AgCa.3, Agre.1, Agre.3, Case.1, Case.3.
##  * Timebin200 : numeric predictor; with 30 values ranging from 100.000000 to 3100.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```




```
qqnorm(resid(mObj3c))
qqline(resid(mObj3c))
```




```
acf(resid(mObj3c))
```



Tables and plots

Difference plots per type of cue for children with NH´s object questions

```
par(mfrow=c(1,3), cex=1.1,mar=c(4,3,4,1))
### NH Children Agre vs. Case
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Case.1','Agre.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

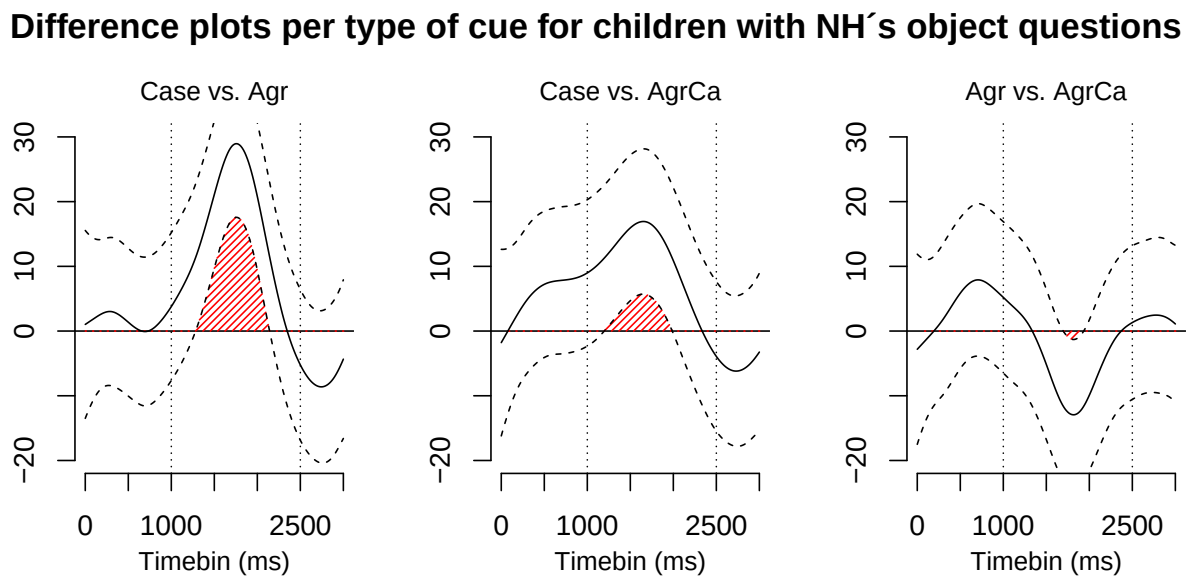
```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Case vs. Agr', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                             0, newd$difference-newd$CI, newd$differenc
          col='red', alpha=1, density=30)
with(newd, plot_error(Timebin200, difference, CI))

### NH Children Case vs. AgrCa
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Case.1','AgCa.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Case vs. AgrCa', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                           0, newd$difference-newd$CI, newd$differenc
          col='red', alpha=1, density=30)
with(newd, plot_error(Timebin200, difference, CI))

### NH Children Agre vs. AgrCa
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Agre.1', 'AgCa.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

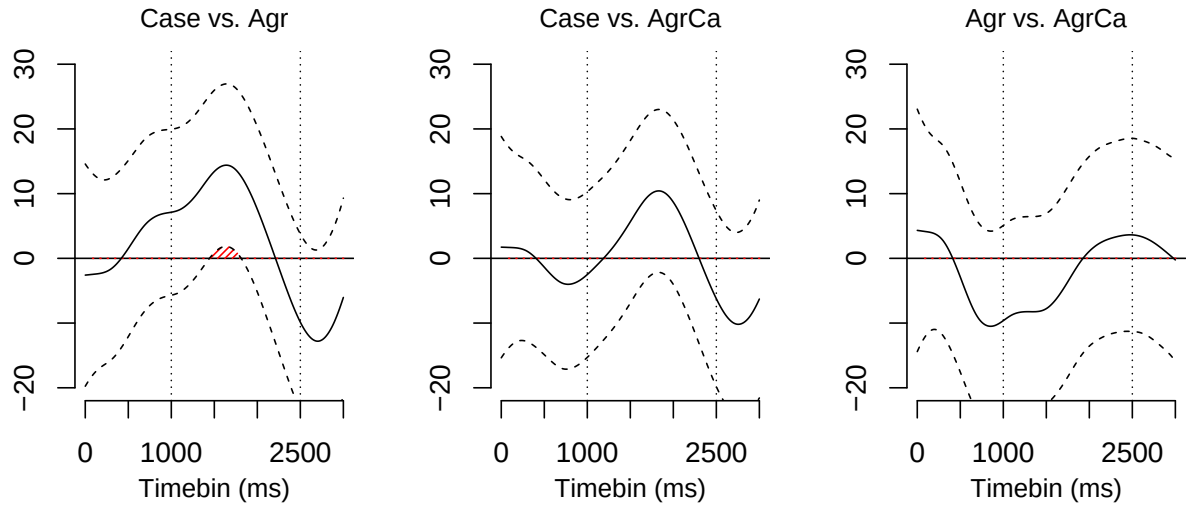
```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Agr vs. AgrCa', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                           0, newd$difference-newd$CI, newd$differenc
          col='red', alpha=1, density=30)
with(newd, plot_error(Timebin200, difference, CI))

title('Difference plots per type of cue for children with NH´s object questions', font.main=2, line= -1
```



Difference plots per type of cue for children with CIs object questions

```
######## per cue for children with CIs ################################3

par(mfrow=c(1,3), cex=1.1,mar=c(4,3,4,1))
```

```
### CI Children Agre vs. Case
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Case.3','Agre.3')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)

## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Case vs. Agr', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                           0, newd$difference-newd$CI, newd$differenc
with(newd, plot_error(Timebin200, difference, CI))

### CI Children Case vs. AgrCa
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Case.3','AgCa.3')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)

## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Case vs. AgrCa', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                           0, newd$difference-newd$CI, newd$differenc
with(newd, plot_error(Timebin200, difference, CI))

### CI Children Agre vs. AgrCa
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Agre.3', 'AgCa.3')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)

## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Agr vs. AgrCa', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                           0, newd$difference-newd$CI, newd$differenc
with(newd, plot_error(Timebin200, difference, CI))

title('Difference plots per type of cue for children with CIs object questions', font.main=2, line= -1.5
```



Difference plots per type of cue children with CI vs. children with NH

```
par(mfrow=c(1,3), cex=1.1,mar=c(4,3,4,1))

### children CI vs chidren NH per cue
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Case.3','Case.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Case', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                             0, newd$difference-newd$CI, newd$differenc
with(newd, plot_error(Timebin200, difference, CI))

### Agr
newd <- get_difference(mObj3c, comp=list(CueGroup=c('Agre.3','Agre.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('Agr', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
```

```
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                             0, newd$difference-newd$CI, newd$differenc
with(newd, plot_error(Timebin200, difference, CI))

### AgrCa
newd <- get_difference(mObj3c, comp=list(CueGroup=c('AgCa.3', 'AgCa.1')),
                       cond=list(Timebin200=seq(0,3000,length=200)), rm.ranef=TRUE, f=2.58)
```

```
## Summary:
##  * Timebin200 : numeric predictor; with 200 values ranging from 0.000000 to 3000.000000.
##  * Subject : factor; set to the value(s): 102. (Might be canceled as random effect, check below.)
##  * NOTE : The following random effects columns are canceled: s(Timebin200,Subject)
##
```

```
newd$f <- find_difference(newd$difference, newd$CI, as.vector=TRUE)
emptyPlot(c(0, 3000), c(-20,30), h0=0, v=c(1000,2500))
mtext('AgrCa', side=3, line=0.5)
mtext('Timebin (ms)', side=1, line=2)
fill_area(newd$Timebin200, ifelse(newd$f==FALSE, 0, ifelse(newd$difference >
                                                             0, newd$difference-newd$CI, newd$differenc
with(newd, plot_error(Timebin200, difference, CI))

title('Difference plots per type of cue children with CI vs. children with NH', font.main=2, line= -1.5
```