MTurk Pupillometry Analysis (Exposure)

Zach Burchill

February 14, 2018

Contents

1 Quick recap 1

2 The GAMM stuff 2
2.1 Setting up the data frame . . . 2
2.2 Logic of the model . . . 3
2.3 Potential issues: . . . 3
2.3.1 Setting the k basis functions . . . 3
2.3.2 Including linear *and* non-linear versions of variables . . . 4
2.3.3 Finnicky details in GAM models . . . 4
2.4 The model . . . 4

3 GAMM plots 5

4 Visualization of predicted values 7

5 Overall thoughts/feelings/questions 7

6 Results from older experiments 10
6.1 Exp1 . . . 10
6.2 Exp1b: Different wording . . . 12

7 Results with a different trial and block predictor 13
7.1 Similar results from previous experiments . . . 16
7.2 Exp1 . . . 16
7.3 Exp1b: Different wording . . . 19

1 Quick recap

In this somewhat replication of C&G'04, we had participants listen to chunks of a story read by their exposure talkers between each block of the RT task, with the intent that this would give them even more exposure to adapt to, letting us study accent adaptation over a longer scale. We had planned to look at how pupil dilation corresponds to accent adaptation after this behavioral study turned out ok.

Although the experiments without the exposure passages seemed encouraging, the results from the experiments with exposure passages interspersed with the C&G task were somewhat confusing–each time the participants came back to the task, it seemed that their response times reset, especially for the first one or two trials. These non-linearities prompted us to approach the problem with GAM models. We discussed using the three exposure blocks for each condition to make predictions about the test block, but I can't quite remember the comparisons we had planned.

Regardless, I've implemented a few different sorts of predictions that the GAM model could make about the test block—namely how the model would predict the responses if the participants heard the test block with the same talker as exposure (i.e. participants in the unaccented condition heard the same unaccented talker for test) and if all the participants heard the accented talker in test (as was done in the actual experiment).

Predicting the data in the test block given only the exposure (during which each participant only heard one talker) lets us establish a baseline to compare the actual results.

2 The GAMM stuff

2.1 Setting up the data frame

Just the nitty-gritty about what I used in the GAM model. Feel free to skip.

```
d.gamm <- dat_out2 %>%
  filter(Condition %in% c("accented","unaccented") &
           Run %in% c("Exp2v1","ExpPilotv1","Exp2v2")) %>%
  within(., {
    Word <- factor(Word)
    WorkerId <- factor(WorkerId)
    # sum coding for accent condition
    Condition <- factor(Condition, levels = c("unaccented","accented"))
    contrasts(Condition) <- cbind("accented" = c(-1,1))
    # Make the unordered version of Condition
    Condition.unordered <- Condition
    # Make the ordered version
    Condition <- ordered(Condition)
    Trial <- TrialInExperiment-max(Trial[Block=="0"])
    Trial.c <- Trial-12.5
    BlockTrial <- Trial-floor((Trial-1)/8)*8
    BlockTrial.c <- BlockTrial-mean(BlockTrial)
  }) %>%
  filter(Block %in% c("1","2","3")) %>%
  within(., {
    # Make Block an ordered factor with helmert coding
    Block <- factor(Block)
    Block <- as.ordered(Block)
    contrasts(Block) <- contr.helmert(3)
    colnames(contrasts(Block)) <- c("Block:2v1","Block:3v12")

    # Make a continuous variable `Block`
    BlockContinuous <- ifelse(Block=="1",1,
                       ifelse(Block=="2",2,
                       ifelse(Block=="3",3,NA)))

    # Make a factor of the combination of `Block` and `Condition`
    BlockXCondition <- factor(paste0(Condition,"-",Block))

    # sum contrast code List (counterbalancing nuisance factor)
    List <- factor(List)
    contrasts(List) <- contr.sum(nlevels(List))
    colnames(contrasts(List)) <- rownames(contrasts(List))[1:(length(levels(List))-1)]

    # sum code ListID
    # Currently considers the new runs completely unrelated
    ListID <- sapply(List,function(x){as.numeric(strsplit(as.character(x),"_")[[1]][3])},
                            USE.NAMES = FALSE)
    ListID <- factor(ifelse(Run=="Exp2v2",
                             paste0(ListID,"_new"),
                             ListID))
    contrasts(ListID) <- contr.sum(nlevels(ListID))

    #sum code ListOrder
    ListOrder <- factor(sapply(List,function(x){strsplit(as.character(x),"_")[[1]][4]},
                             USE.NAMES = FALSE))
    contrasts(ListOrder) <- contr.sum(nlevels(ListOrder))
    }) %>%
  as.data.frame()
```

2.2 Logic of the model

I begin by making my assumptions clear (or as clear as I can). The model will predict `AdjustedRT` (the adjusted reaction times) with the following linear predictors:

Linear predictors:

- ~~Block number (`BlockContinuous`), 1-3, so we can extend our predictions to Block 4, as a linear predictor~~
- Exposure condition (`Condition`), a binary centered ordered factor (1 and -1), as a linear predictor (so that we can more easily test differences between conditions)
- ~~The interaction of `Block` and `Condition`, as a linear predictor~~

Non-linear predictors:

- **Block number (BlockContinuous), 1-3, so we can extend our predictions to Block 4, as a linear predictor.** I decided to not have Block number as a linear *and* non-linear predictor.
- Within-block trial number (`BlockTrial`), 1-8, then centered around the mean (i.e. 4.5), as a continuous variable, as a non-linear predictor
- The interaction of `Block` and `BlockTrial`, so that different smooths for `BlockTrial` are fitted to each level of `Block`, as a non-linear predictor. Because `Block` is an ordered factor, the smooths for each level of the factor will be a "reference" smooth for the first level and "difference" smooths for all the other levels, representing the difference from the reference. (I'm not actually sure how/if the coding of the factor actually does anything if each level gets its own smooth. In fact, I'm pretty sure the coding doesn't matter at all in the interaction, based on examples from the package)
- The interaction of `Condition` and `BlockTrial`, so that different smooths for `BlockTrial` are fitted to each level of `Condition`
- The interaction of `BlockTrial` and the combination of `Condition` and `Block`, so that different smooths for `BlockTrial` are fitted to each combination of `Condition` and `Block`

Random effects:

- **Maybe:** Factor smooths for `BlockTrial` each participant (`WorkerId`), so that each participant has their own smooth fitted for `BlockTrial` (also maybe: the same smoothing parameter for each participant)
- Random intercepts for the audio clips (i.e. a cross of sentence and talker)

2.3 Potential issues:

2.3.1 Setting the k basis functions

`BlockTrial` only has 8 values (1-8) and `BlockContinuous` only has 3 (1-3). Jacolien van Rij says the following:

> "Note that the model will base the number of base functions (reflected in the `edf` of the summary) on the data with the setting for `k` as upper bound. *By default, the value of `k` for `s()` is around 9,*

> *and for `te()` and `ti()` 5 per dimension.* Importantly, the value of `k` should be at most one less than the number of unique data points, otherwise it will fit the density of that predictor.

The `mgcv` documentation says the following:

> "When setting up models in the `mgcv` package, using `s` or `te` terms in a model formula, `k` must be chosen: the defaults are essentially arbitrary.

To me, this means that since our variables have so few different values, we should manually set `k` so that it's one less than the number of unique data points for all non-linear predictors, and completely ignore defaults.

2.3.2 Including linear *and* non-linear versions of variables

In the current model, I've included both linear and non-linear versions of `BlockContinuous` and `Condition`.

2.3.3 Finnicky details in GAM models

There's also the possibility that something about the way I fit the model is unoptimal. There are a bunch of finnicky things that I tried to account for, but I'm no expert, so I might have gotten them wrong. For example, I tried to set the k basis functions as well as I understand them, but I'm not 100% confident in them. Likewise, choices such as whether to center certain variables, whether to use factors vs. continuous predictors vs. ordered factors, etc.

The aspect I'm least sure is the `d` parameter in `mgcv::ti`. The predictors in my interactions are *not* on the same scale, and I *think* the default for `mgcv::ti` assumes that, but I can't really make heads or tails of the documentation.

2.4 The model

This model is only fit to correct responses.

```r
m.full.block_contin <- gam(AdjustedRT ~
      # Linear predictors
      Condition + #BlockContinuous + BlockContinuous:Condition +
      # Non linear predictors
      ti(BlockContinuous, k=3) +
      ti(BlockTrial, k=7) +
      ti(BlockContinuous, by=Condition, k=3) +
      ti(BlockTrial, BlockContinuous, k=3) +
      ti(BlockTrial, by=Condition, k=7) +
      ti(BlockTrial, BlockContinuous, by=Condition, k=3) +
      # Random effects
      s(WorkerId, bs="re") + #Random intercept
      s(WorkerId, BlockTrial, bs="re"),  # Random slope
    data = d.gamm %>% filter(BinaryCorrect==1)
      )
summary(m.full.block_contin)
```

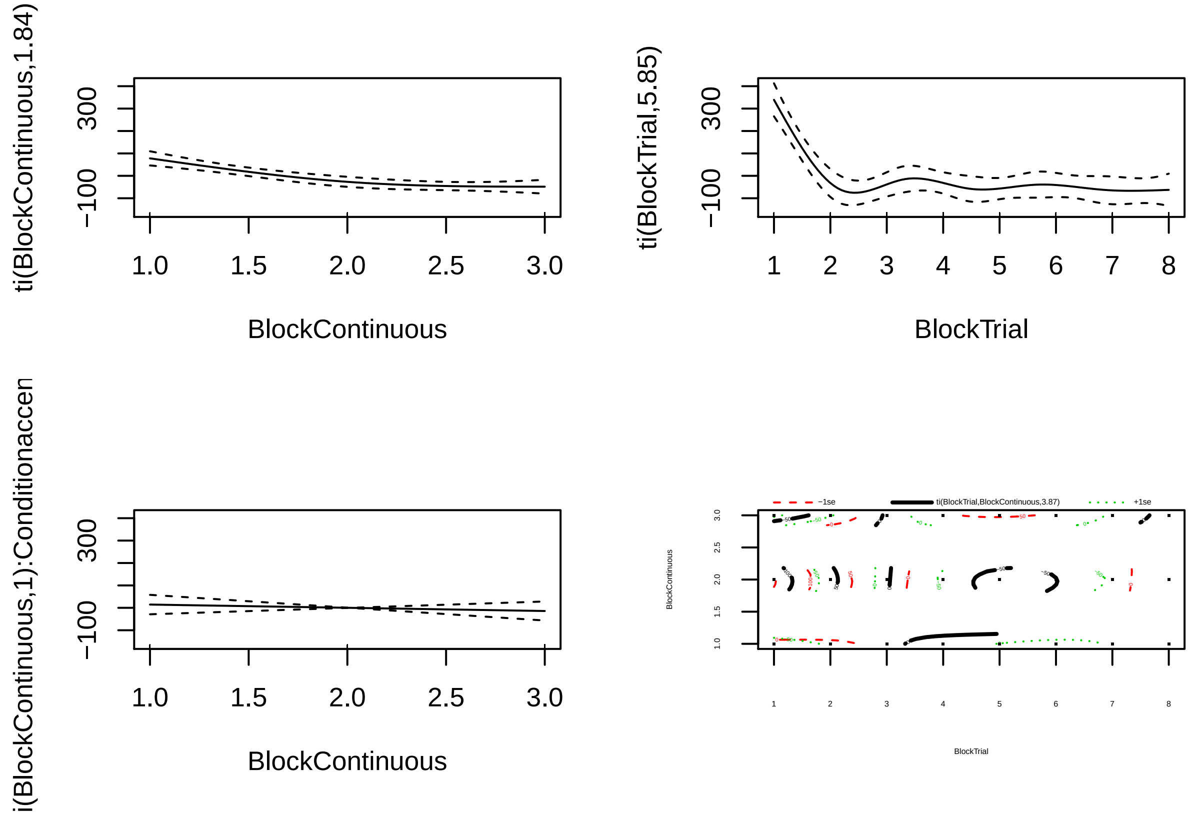
```
Family: gaussian
Link function: identity

Formula:
AdjustedRT ~ Condition + ti(BlockContinuous, k = 3) + ti(BlockTrial,
    k = 7) + ti(BlockContinuous, by = Condition, k = 3) + ti(BlockTrial,
    BlockContinuous, k = 3) + ti(BlockTrial, by = Condition,
    k = 7) + ti(BlockTrial, BlockContinuous, by = Condition,
    k = 3) + s(WorkerId, bs = "re") + s(WorkerId, BlockTrial,
    bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   134.25      44.83   2.995  0.00278 **
Condition.L   108.52      63.40   1.712  0.08710 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                                                  edf Ref.df      F
ti(BlockContinuous)                             1.839  1.974 13.709
ti(BlockTrial)                                  5.847  5.971 16.155
ti(BlockContinuous):Conditionaccented           1.000  1.000  0.449
ti(BlockTrial,BlockContinuous)                  3.872  3.973  3.643
ti(BlockTrial):Conditionaccented                4.627  5.301  2.093
ti(BlockTrial,BlockContinuous):Conditionaccented  2.684  3.339  0.895
s(WorkerId)                                    80.391 92.000 41.558
s(WorkerId,BlockTrial)                         51.683 92.000 40.695
                                                p-value
ti(BlockContinuous)                            8.83e-06 ***
ti(BlockTrial)                                  < 2e-16 ***
ti(BlockContinuous):Conditionaccented           0.50299
ti(BlockTrial,BlockContinuous)                  0.00475 **
ti(BlockTrial):Conditionaccented                0.06585 .
ti(BlockTrial,BlockContinuous):Conditionaccented  0.35281
s(WorkerId)                                     < 2e-16 ***
s(WorkerId,BlockTrial)                         8.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.326   Deviance explained = 37.6%
GCV = 1.676e+05  Scale est. = 1.5508e+05  n = 2060
```

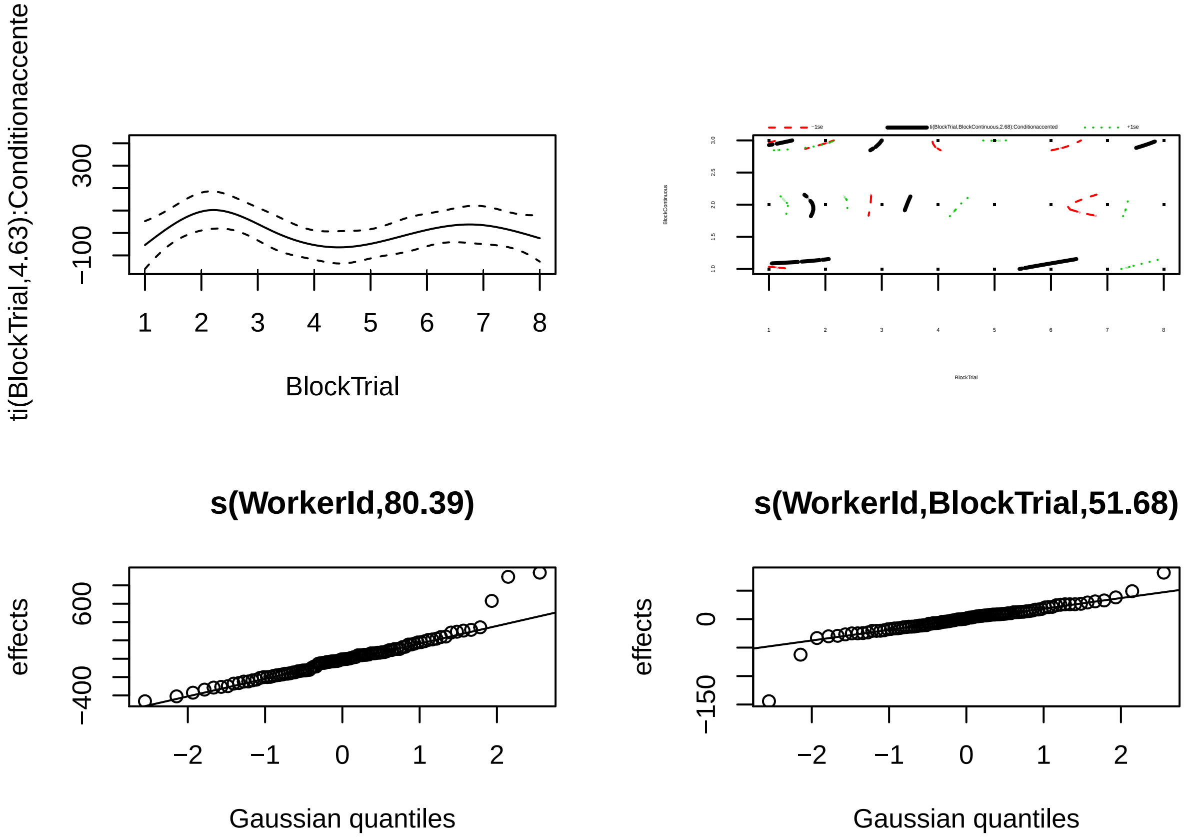
3 GAMM plots

These are plots of the components of the model. I can't really decipher the interaction plots, but the splines seem relatively interpretable.

I made the `Condition` variable an ordered factor in the model—from what I understand, this means that the splines that have `:Conditionaccented` in the y axis are "difference" smooths—how the `accented` splines differ from the `unaccented` ones.

It seems that `Condition` doesn't really change the slope of the effect of Block number (`BlockContinuous`), although it seems to be doing something with the effect of the trial number within the block (`BlockTrial`).


ti(BlockContinuous,1.84)
300
−100
1.0
1.5
2.0
2.5
3.0
BlockContinuous
ti(BlockTrial,5.85)
300
−100
1
2
3
4
5
6
7
8
BlockTrial
i(BlockContinuous,1):Conditionaccen
300
−100
1.0
1.5
2.0
2.5
3.0
BlockContinuous
−1se
ti(BlockTrial,BlockContinuous,3.87)
+1se
BlockContinuous
3.0
2.5
2.0
1.5
1.0
1
2
3
4
5
6
7
8
BlockTrial



4 Visualization of predicted values

5 Overall thoughts/feelings/questions

These results don't strike me as *super* encouraging, at least in the sense that they don't *seem* to suggest that there's adaptation in the *exposure* phase, which might matter for the pupillometry. However, it might be interesting to compare these results (which come from the experiments where there were passive listening sections between blocks) to the results where there were no passive listening blocks (which seemed to have much stronger results).

Like I said earlier, I'm less certain of how to use the actual test block data to measure adaptation in the context of a GAM model. Would it be fair to compare the actual results with the predicted values?

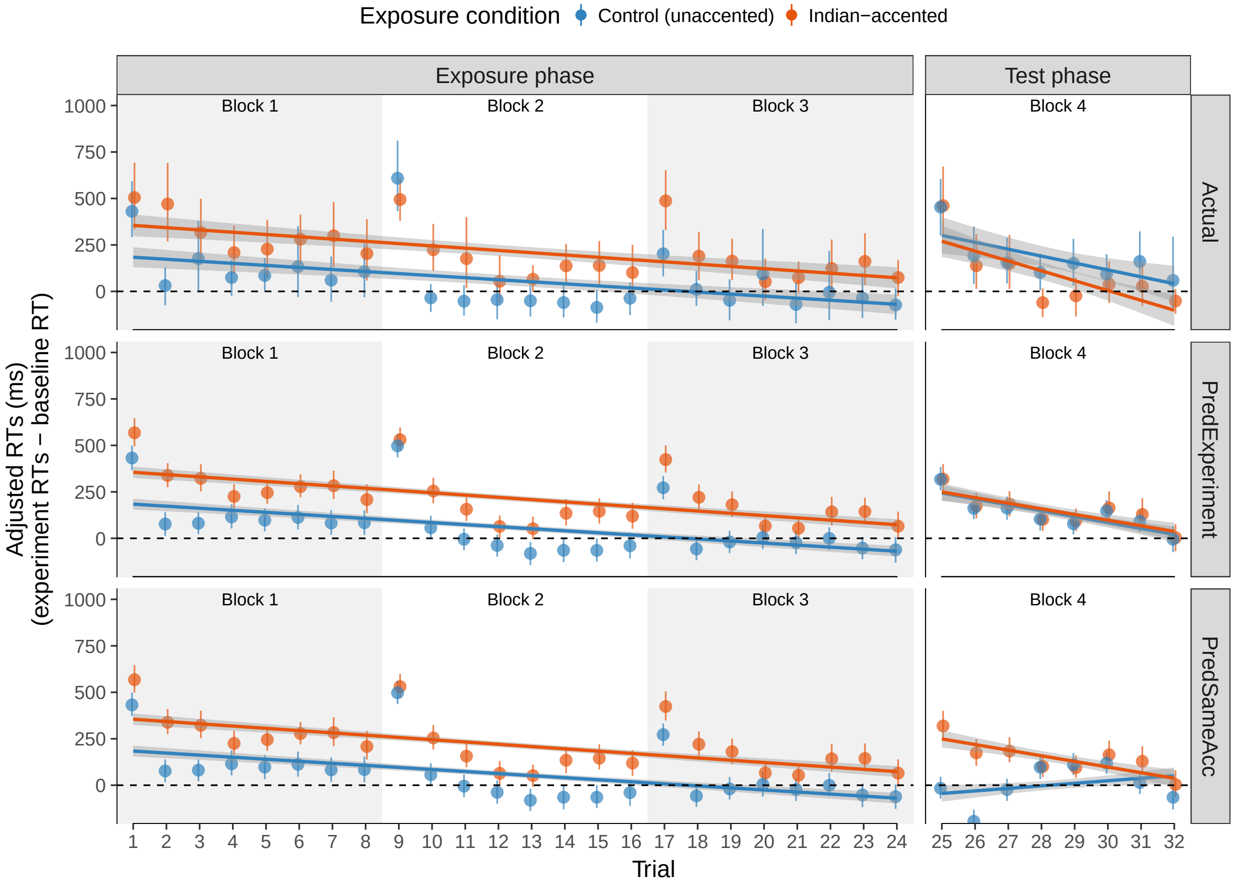


Figure 1: The adjusted RTs for correct trials in the main experiment, for both the actual data (Actual), and the predicted adjusted RTs for two scenarios: one emulating a situation in which all participants hear the accented talker in the fourth block (PredExperiment, as in the real experiment), and when they hear the same talker they heard in exposure (PredSameAcc). The predicted exposure values for PredExperiment and PredSameAcc are identical. The predicted data for the fourth block was generated by a model fit to the exposure blocks.

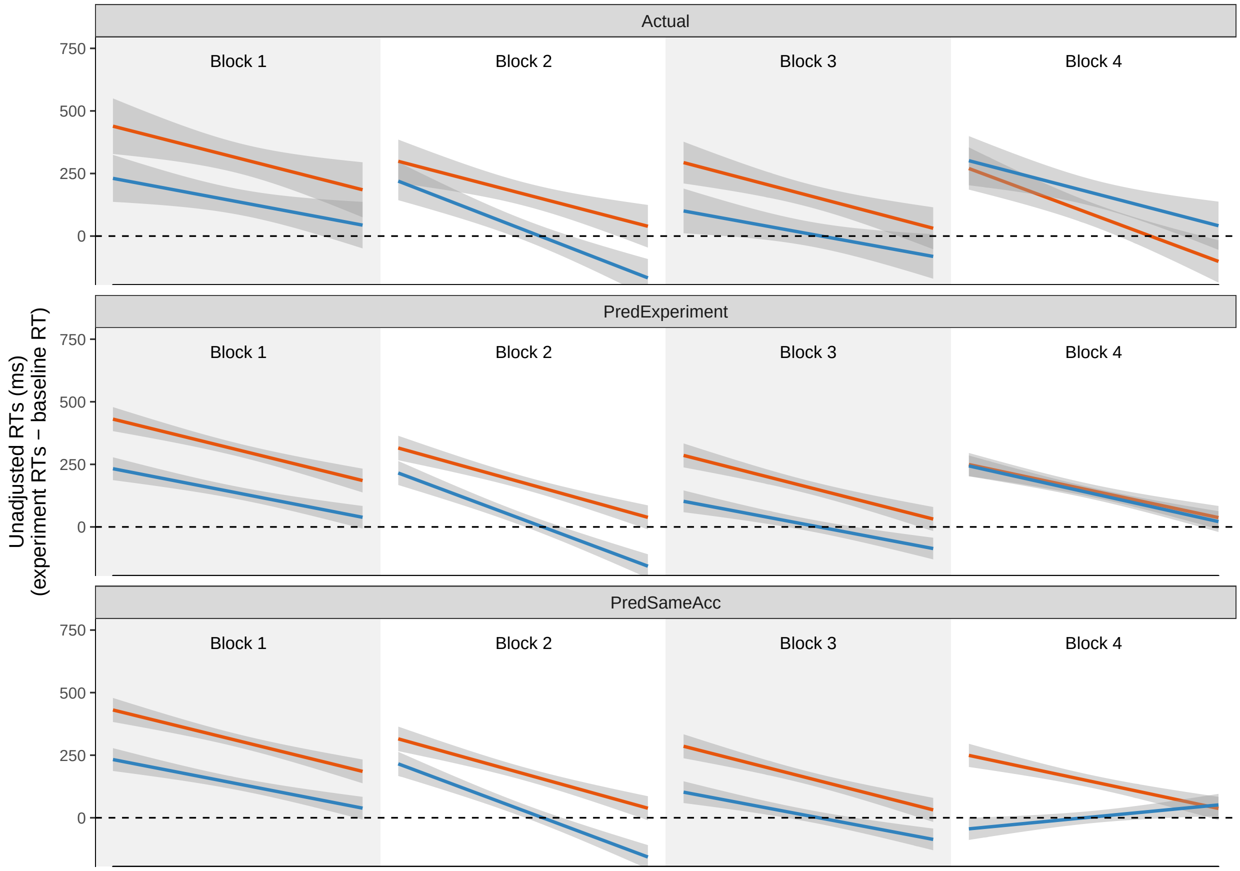


Figure 2: The linear trends per Block for the actual adjusted RTs (Actual), the predicted RTs for when all participants hear the accented voice (PredExperiment), and the predicted RTs for the scenario where participants would hear a fourth block in the same accent as they heard in exposure (PredSameAcc). The predicted exposure values for PredExperiment and PredSameAcc are identical.

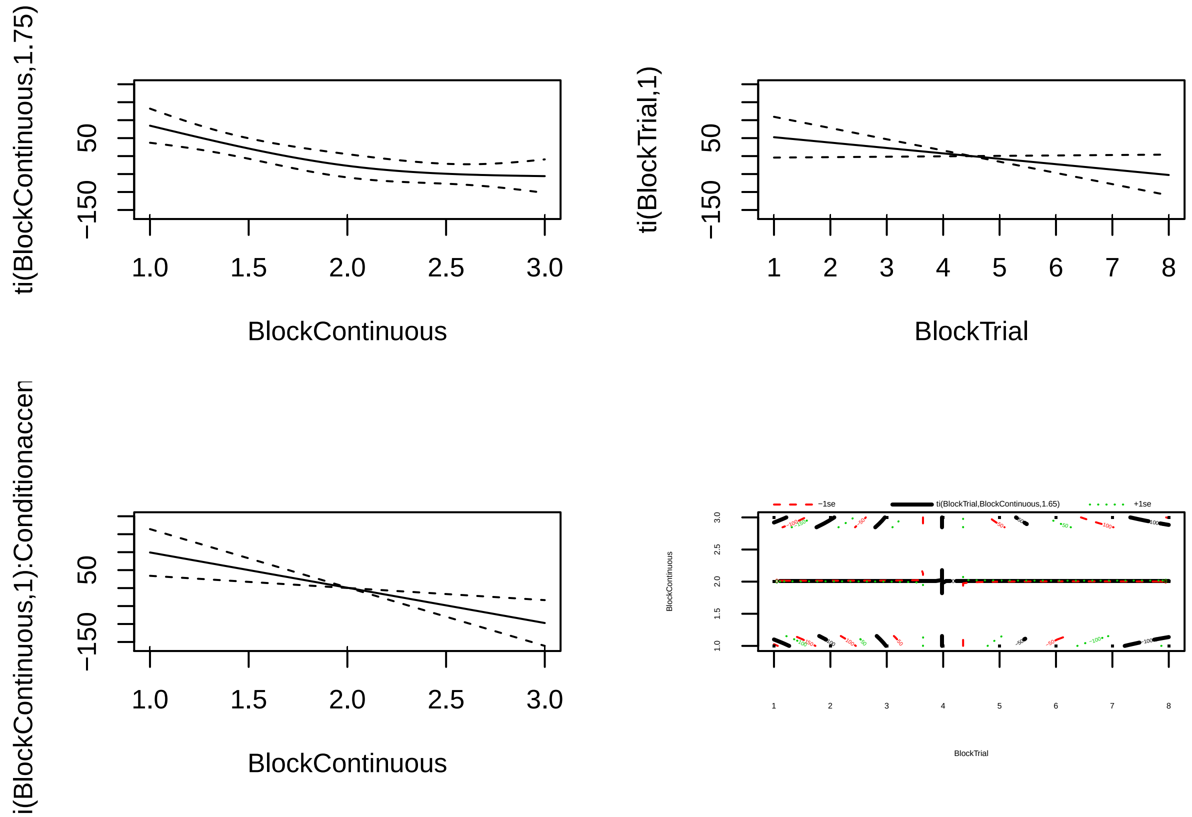
6 Results from older experiments

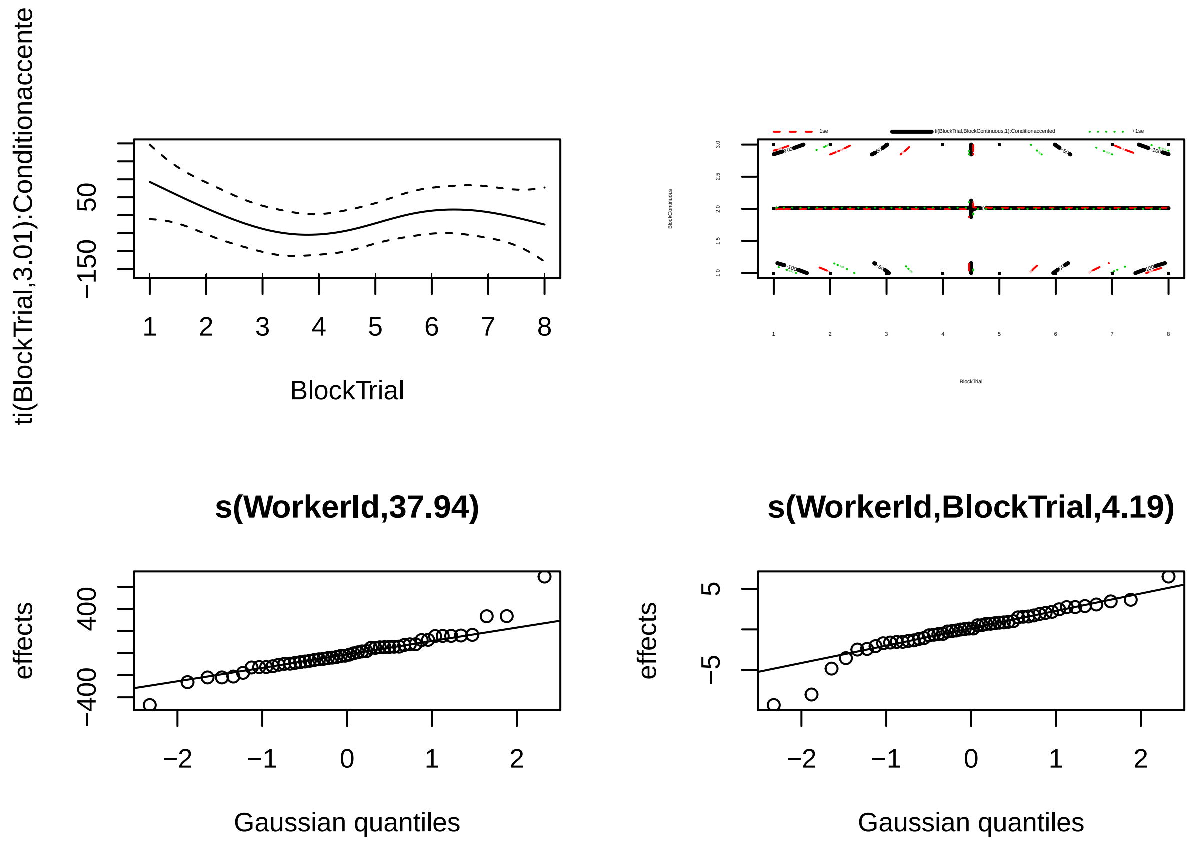
6.1 Exp1

```
m.ex1.full.block_contin <- gam(AdjustedRT ~
      # Linear predictors
      Condition + #BlockContinuous + BlockContinuous:Condition +
      # Non linear predictors
      ti(BlockContinuous, k=3) +
      ti(BlockTrial, k=7) +
      ti(BlockContinuous, by=Condition, k=3) +
      ti(BlockTrial, BlockContinuous, k=3) +
      ti(BlockTrial, by=Condition, k=7) +
      ti(BlockTrial, BlockContinuous, by=Condition, k=3) +
      # Random effects
      s(WorkerId, bs="re") + #Random intercept
      s(WorkerId, BlockTrial, bs="re"),  # Random slope
    data = d.gamm.ex1 %>% filter(BinaryCorrect==1)
      )
summary(m.ex1.full.block_contin)
```

```
Family: gaussian
Link function: identity

Formula:
AdjustedRT ~ Condition + ti(BlockContinuous, k = 3) + ti(BlockTrial,
    k = 7) + ti(BlockContinuous, by = Condition, k = 3) + ti(BlockTrial,
    BlockContinuous, k = 3) + ti(BlockTrial, by = Condition,
    k = 7) + ti(BlockTrial, BlockContinuous, by = Condition,
    k = 3) + s(WorkerId, bs = "re") + s(WorkerId, BlockTrial,
    bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    99.63      31.67   3.145 0.001706 **
Condition.L   158.95      44.79   3.548 0.000405 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                                                    edf Ref.df      F
ti(BlockContinuous)                               1.750  1.937  7.581
ti(BlockTrial)                                    1.000  1.000  3.436
ti(BlockContinuous):Conditionaccented             1.000  1.000  9.314
ti(BlockTrial,BlockContinuous)                    1.647  1.875 11.459
ti(BlockTrial):Conditionaccented                  3.011  3.666  1.479
ti(BlockTrial,BlockContinuous):Conditionaccented  1.000  1.000  6.022
s(WorkerId)                                      37.939 48.000  5.424
s(WorkerId,BlockTrial)                            4.191 48.000  0.186
                                                 p-value
ti(BlockContinuous)                              0.00247 **
ti(BlockTrial)                                   0.06407 .
ti(BlockContinuous):Conditionaccented           0.00233 **
ti(BlockTrial,BlockContinuous)                  0.00019 ***
ti(BlockTrial):Conditionaccented                0.16849
ti(BlockTrial,BlockContinuous):Conditionaccented 0.01429 *
s(WorkerId)                                     < 2e-16 ***
s(WorkerId,BlockTrial)                          0.43744
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.267   Deviance explained = 30.2%
GCV = 1.9613e+05  Scale est. = 1.8646e+05  n = 1086
```

```
#summary(m..full.block_contin)
plot(m.ex1.full.block_contin,pages=2)
```





6.2 Exp1b: Different wording

```
m.wording.full.block_contin <- gam(AdjustedRT ~
      # Linear predictors
      Condition + #BlockContinuous + BlockContinuous:Condition +
      # Non linear predictors
      ti(BlockContinuous, k=3) +
      ti(BlockTrial, k=7) +
      ti(BlockContinuous, by=Condition, k=3) +
      ti(BlockTrial, BlockContinuous, k=3) +
      ti(BlockTrial, by=Condition, k=7) +
      ti(BlockTrial, BlockContinuous, by=Condition, k=3) +
      # Random effects
      s(WorkerId, bs="re") + #Random intercept
      s(WorkerId, BlockTrial, bs="re"),  # Random slope
    data = d.gamm.wording %>% filter(BinaryCorrect==1)
      )
summary(m.wording.full.block_contin)
```

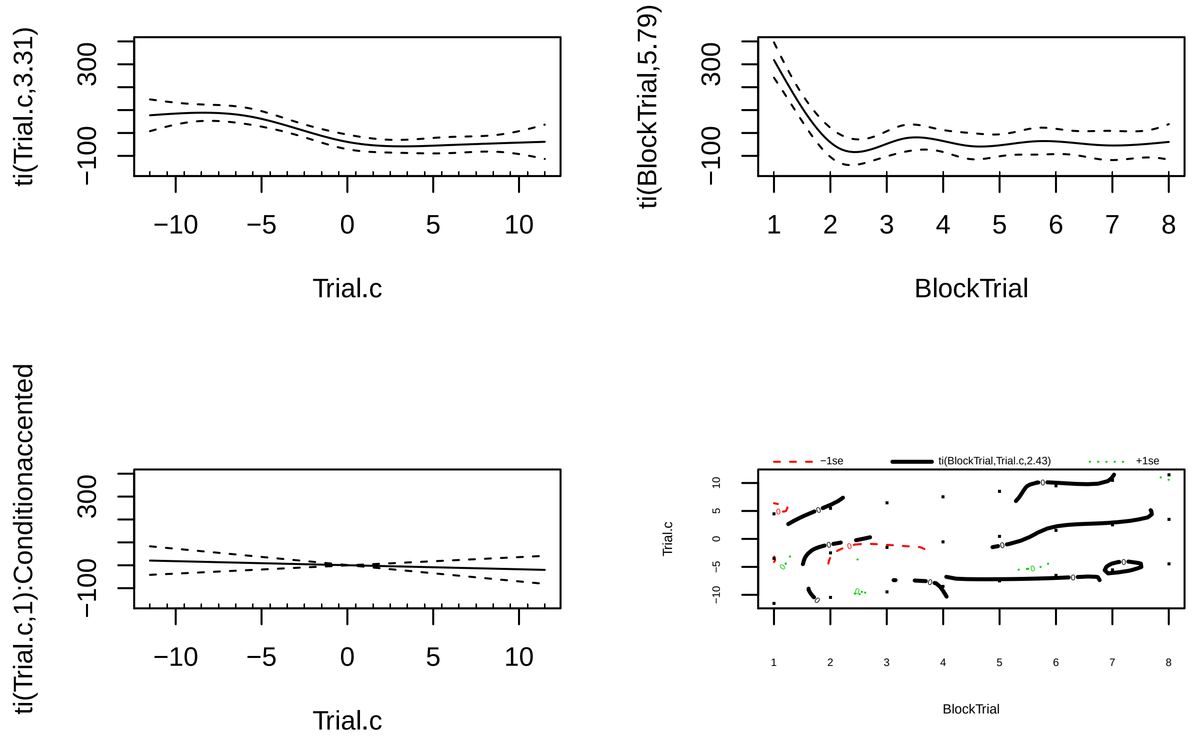
```
Family: gaussian
Link function: identity

Formula:
AdjustedRT ~ Condition + ti(BlockContinuous, k = 3) + ti(BlockTrial,
    k = 7) + ti(BlockContinuous, by = Condition, k = 3) + ti(BlockTrial,
    BlockContinuous, k = 3) + ti(BlockTrial, by = Condition,
    k = 7) + ti(BlockTrial, BlockContinuous, by = Condition,
    k = 3) + s(WorkerId, bs = "re") + s(WorkerId, BlockTrial,
    bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   134.85      37.65   3.581 0.000359 ***
Condition.L   126.40      53.25   2.374 0.017815 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                                                    edf Ref.df      F
ti(BlockContinuous)                               1.500  1.750 17.897
ti(BlockTrial)                                    5.260  5.723  1.489
ti(BlockContinuous):Conditionaccented             1.000  1.000  0.445
ti(BlockTrial,BlockContinuous)                    1.836  1.959  3.806
ti(BlockTrial):Conditionaccented                  3.872  4.570  1.374
ti(BlockTrial,BlockContinuous):Conditionaccented  2.147  2.621  1.617
s(WorkerId)                                      32.452 45.000 16.876
s(WorkerId,BlockTrial)                           20.131 45.000  8.076
                                                  p-value    
ti(BlockContinuous)                              3.03e-06 ***
ti(BlockTrial)                                     0.3712    
ti(BlockContinuous):Conditionaccented              0.5049    
ti(BlockTrial,BlockContinuous)                     0.0406 *  
ti(BlockTrial):Conditionaccented                   0.2095    
ti(BlockTrial,BlockContinuous):Conditionaccented   0.2012    
s(WorkerId)                                       < 2e-16 ***
s(WorkerId,BlockTrial)                           4.89e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.389   Deviance explained =   43%
GCV = 1.7405e+05  Scale est. = 1.622e+05  n = 1031
```

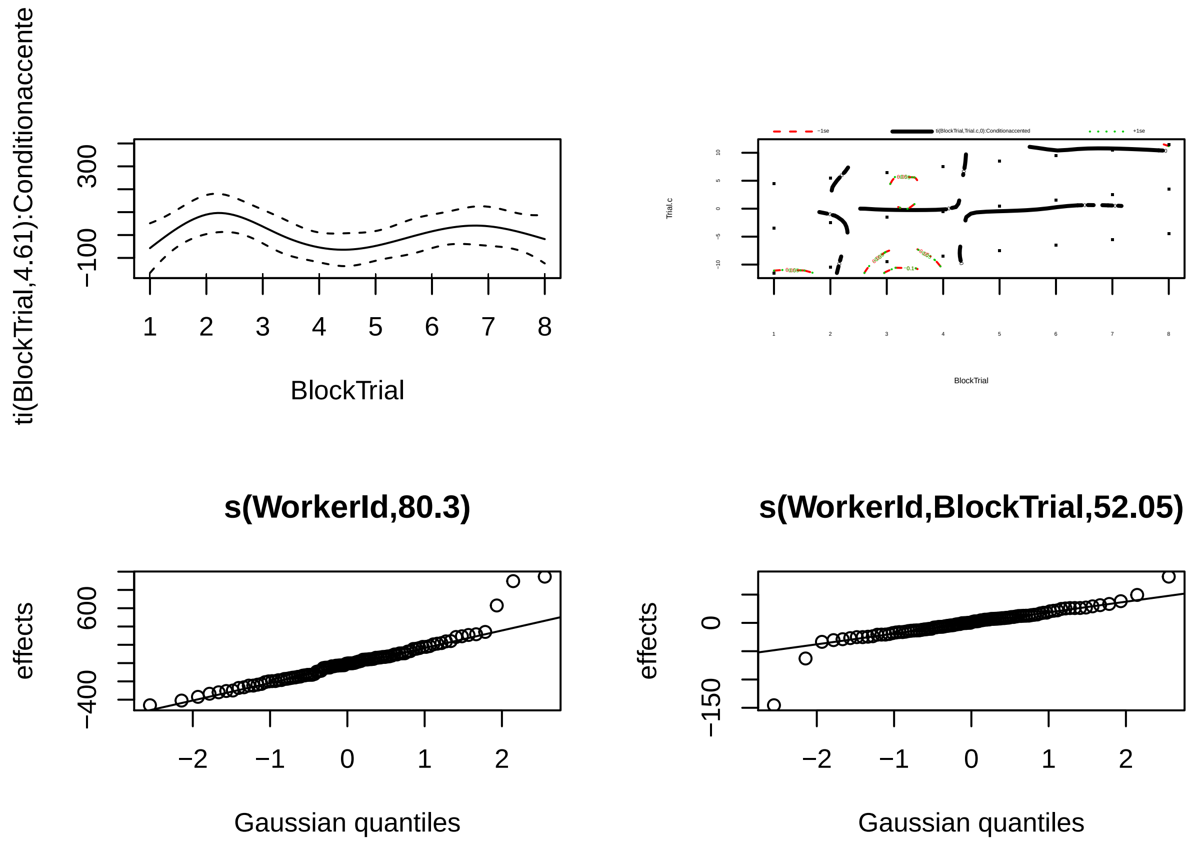
7 Results with a different trial and block predictor

```
m.full.no_block <- gam(AdjustedRT ~
      # Linear predictors
      Condition + #BlockContinuous + BlockContinuous:Condition +
      # Non linear predictors
      ti(Trial.c) +
      ti(BlockTrial, k=7) +
      ti(Trial.c, by=Condition) +
      ti(BlockTrial, Trial.c, k=7) +
      ti(BlockTrial, by=Condition, k=7) +
      ti(BlockTrial, Trial.c, by=Condition, k=7) +
      # Random effects
      s(WorkerId, bs="re") + #Random intercept
      s(WorkerId, BlockTrial, bs="re"),  # Random slope
    data = d.gamm %>% filter(BinaryCorrect==1)
      )
summary(m.full.no_block)
```

```
Family: gaussian
Link function: identity

Formula:
AdjustedRT ~ Condition + ti(Trial.c) + ti(BlockTrial, k = 7) +
    ti(Trial.c, by = Condition) + ti(BlockTrial, Trial.c, k = 7) +
    ti(BlockTrial, by = Condition, k = 7) + ti(BlockTrial, Trial.c,
    by = Condition, k = 7) + s(WorkerId, bs = "re") + s(WorkerId,
    BlockTrial, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   135.55      44.94   3.016  0.00259 **
Condition.L   108.80      63.39   1.716  0.08624 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                                              edf Ref.df      F  p-value
ti(Trial.c)                             3.306e+00  3.683  7.745 3.80e-05
ti(BlockTrial)                          5.791e+00  5.954 14.292 1.94e-15
ti(Trial.c):Conditionaccented           1.000e+00  1.000  0.433   0.5107
ti(BlockTrial,Trial.c)                  2.432e+00 13.000  0.332   0.0806
ti(BlockTrial):Conditionaccented        4.613e+00  5.285  2.029   0.0740
ti(BlockTrial,Trial.c):Conditionaccented 4.190e-06 14.000  0.000   1.0000
s(WorkerId)                             8.030e+01 92.000 42.128  < 2e-16
s(WorkerId,BlockTrial)                  5.205e+01 92.000 41.172 8.11e-05

ti(Trial.c)                             ***
ti(BlockTrial)                          ***
ti(Trial.c):Conditionaccented
ti(BlockTrial,Trial.c)                  .
ti(BlockTrial):Conditionaccented        .
ti(BlockTrial,Trial.c):Conditionaccented
s(WorkerId)                             ***
s(WorkerId,BlockTrial)                  ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.326   Deviance explained = 37.5%
GCV = 1.6752e+05  Scale est. = 1.552e+05  n = 2060
```

```
plot(m.full.no_block,pages=2)
```


ti(Trial.c,3.31)
300
100
−100
−10
−5
0
5
10
Trial.c
ti(BlockTrial,5.79)
300
100
−100
1
2
3
4
5
6
7
8
BlockTrial
ti(Trial.c,1):Conditionaccented
300
100
−100
−10
−5
0
5
10
Trial.c
−1se
ti(BlockTrial,Trial.c,2.43)
+1se
Trial.c
10
5
0
−5
−10
1
2
3
4
5
6
7
8
BlockTrial



7.1 Similar results from previous experiments

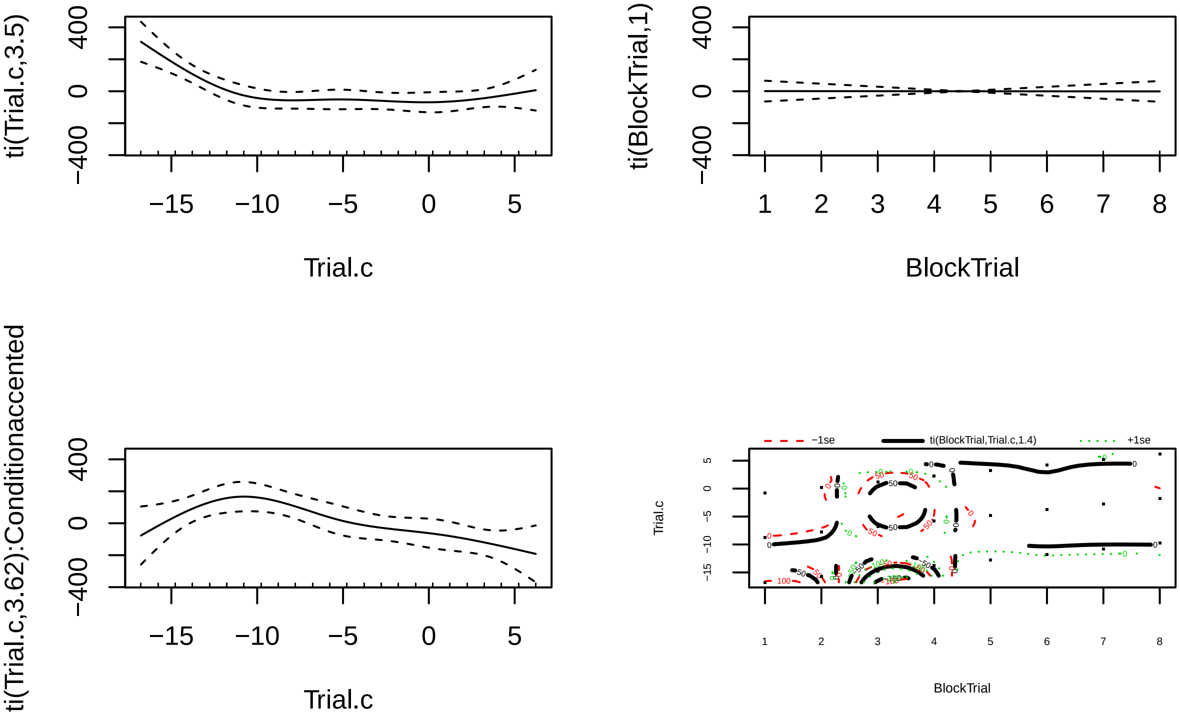
7.2 Exp1

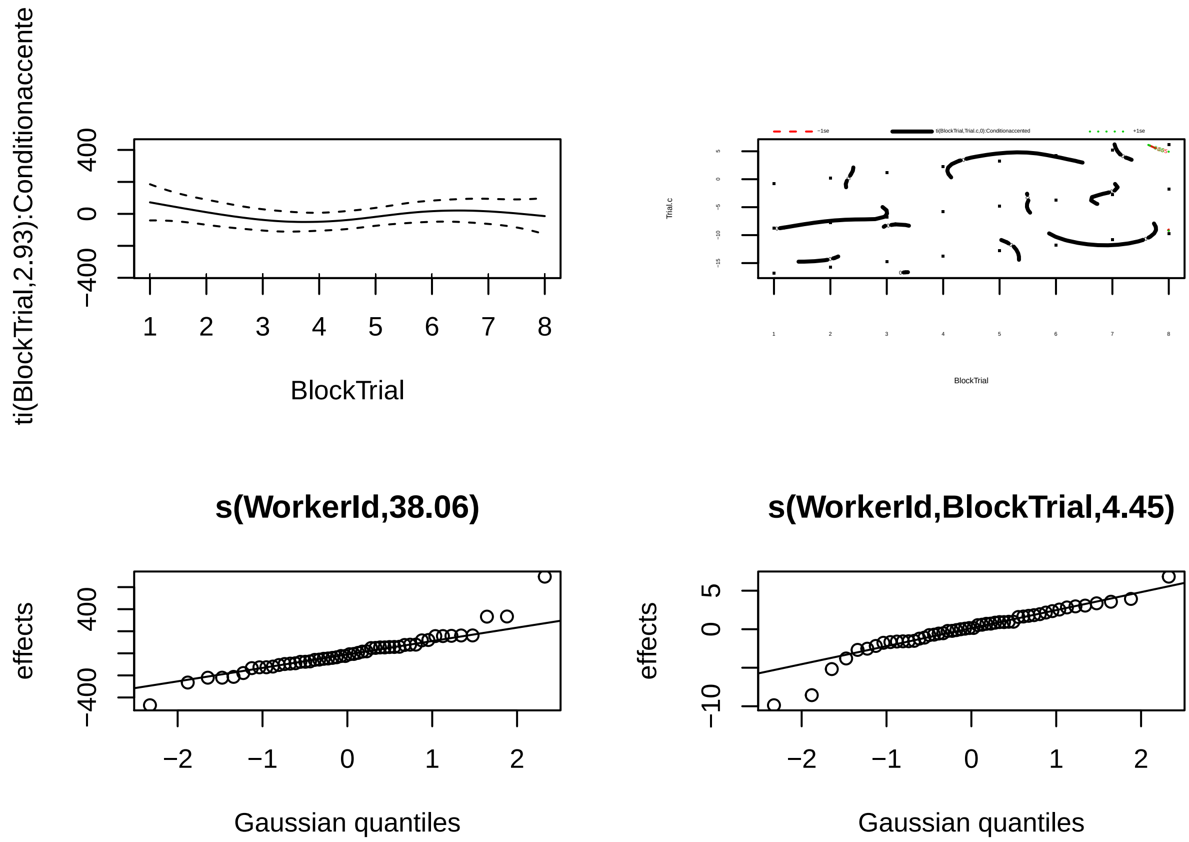
```
m.ex1.full.no_block <- gam(AdjustedRT ~
      # Linear predictors
      Condition + #BlockContinuous + BlockContinuous:Condition +
      # Non linear predictors
      ti(Trial.c) +
      ti(BlockTrial, k=7) +
      ti(Trial.c, by=Condition) +
      ti(BlockTrial, Trial.c, k=7) +
      ti(BlockTrial, by=Condition, k=7) +
      ti(BlockTrial, Trial.c, by=Condition, k=7) +
      # Random effects
      s(WorkerId, bs="re") + #Random intercept
      s(WorkerId, BlockTrial, bs="re"),  # Random slope
    data = d.gamm.ex1 %>% filter(BinaryCorrect==1)
      )
summary(m.ex1.full.no_block)
```

```
Family: gaussian
Link function: identity
```

```
Formula:
AdjustedRT ~ Condition + ti(Trial.c) + ti(BlockTrial, k = 7) +
    ti(Trial.c, by = Condition) + ti(BlockTrial, Trial.c, k = 7) +
    ti(BlockTrial, by = Condition, k = 7) + ti(BlockTrial, Trial.c,
    by = Condition, k = 7) + s(WorkerId, bs = "re") + s(WorkerId,
    BlockTrial, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)    98.65      31.98   3.085 0.002091 ** 
Condition.L   159.37      45.13   3.531 0.000432 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                                               edf Ref.df     F  p-value
ti(Trial.c)                              3.500e+00  3.821 7.410 1.03e-05
ti(BlockTrial)                           1.000e+00  1.000 0.001 0.977737
ti(Trial.c):Conditionaccented            3.616e+00  3.889 4.997 0.001995
ti(BlockTrial,Trial.c)                   1.398e+00 13.000 0.668 0.000893
ti(BlockTrial):Conditionaccented         2.933e+00  3.573 1.122 0.274882
ti(BlockTrial,Trial.c):Conditionaccented 2.003e-06 14.000 0.000 0.886171
s(WorkerId)                              3.806e+01 48.000 5.544  < 2e-16
s(WorkerId,BlockTrial)                   4.445e+00 48.000 0.207 0.442253

ti(Trial.c)                              ***
ti(BlockTrial)                              
ti(Trial.c):Conditionaccented            ** 
ti(BlockTrial,Trial.c)                   ***
ti(BlockTrial):Conditionaccented            
ti(BlockTrial,Trial.c):Conditionaccented    
s(WorkerId)                              ***
s(WorkerId,BlockTrial)                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.274   Deviance explained = 31.1%
GCV = 1.9491e+05  Scale est. = 1.8469e+05  n = 1086
```

```r
plot(m.ex1.full.no_block,pages=2)
```


ti(Trial.c,3.5)
400
0
−400
−15
−10
−5
0
5
Trial.c
ti(BlockTrial,1)
400
0
−400
1
2
3
4
5
6
7
8
BlockTrial
ti(Trial.c,3.62):Conditionaccented
400
0
−400
−15
−10
−5
0
5
Trial.c
−1se
ti(BlockTrial,Trial.c,1.4)
+1se
Trial.c
5
0
−5
−10
−15
1
2
3
4
5
6
7
8
BlockTrial



7.3 Exp1b: Different wording

```
m.wording.full.no_block <- gam(AdjustedRT ~
      # Linear predictors
      Condition + #BlockContinuous + BlockContinuous:Condition +
      # Non linear predictors
      ti(Trial.c) +
      ti(BlockTrial, k=7) +
      ti(Trial.c, by=Condition) +
      ti(BlockTrial, Trial.c, k=7) +
      ti(BlockTrial, by=Condition, k=7) +
      ti(BlockTrial, Trial.c, by=Condition, k=7) +
      # Random effects
      s(WorkerId, bs="re") + #Random intercept
      s(WorkerId, BlockTrial, bs="re"),  # Random slope
    data = d.gamm.ex1 %>% filter(BinaryCorrect==1)
      )
summary(m.wording.full.no_block)
```

```
Family: gaussian
Link function: identity

Formula:
AdjustedRT ~ Condition + ti(Trial.c) + ti(BlockTrial, k = 7) +
    ti(Trial.c, by = Condition) + ti(BlockTrial, Trial.c, k = 7) +
    ti(BlockTrial, by = Condition, k = 7) + ti(BlockTrial, Trial.c,
    by = Condition, k = 7) + s(WorkerId, bs = "re") + s(WorkerId,
    BlockTrial, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    98.65      31.98   3.085 0.002091 **
Condition.L   159.37      45.13   3.531 0.000432 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                                              edf Ref.df     F  p-value
ti(Trial.c)                             3.500e+00  3.821 7.410 1.03e-05
ti(BlockTrial)                          1.000e+00  1.000 0.001 0.977737
ti(Trial.c):Conditionaccented           3.616e+00  3.889 4.997 0.001995
ti(BlockTrial,Trial.c)                  1.398e+00 13.000 0.668 0.000893
ti(BlockTrial):Conditionaccented        2.933e+00  3.573 1.122 0.274882
ti(BlockTrial,Trial.c):Conditionaccented 2.003e-06 14.000 0.000 0.886171
s(WorkerId)                             3.806e+01 48.000 5.544  < 2e-16
s(WorkerId,BlockTrial)                  4.445e+00 48.000 0.207 0.442253

ti(Trial.c)                             ***
ti(BlockTrial)
ti(Trial.c):Conditionaccented           **
ti(BlockTrial,Trial.c)                  ***
ti(BlockTrial):Conditionaccented
ti(BlockTrial,Trial.c):Conditionaccented
s(WorkerId)                             ***
s(WorkerId,BlockTrial)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.274   Deviance explained = 31.1%
GCV = 1.9491e+05  Scale est. = 1.8469e+05  n = 1086
```