hnrGamms

```r
require(tidyverse)
require(mgcv)
require(mgcViz)
require(itsadug)
```

Load prepared hnr data

```r
setwd("C:/Users/Helen/Desktop/Stats/Pruned3_big")

hnrData = read.csv("hnr_ready_for_gamms.csv")

hnrData$speaker = as.factor(hnrData$speaker)
hnrData$label = as.factor(hnrData$label)
```

GAM for hnr

```r
m1hnr=bam(hnr ~ label + te(time, by=label) + s(speaker, bs="re")
         + s(time, speaker, bs="fs", m=1), data=hnrData)
```

```r
summary(m1hnr)
```

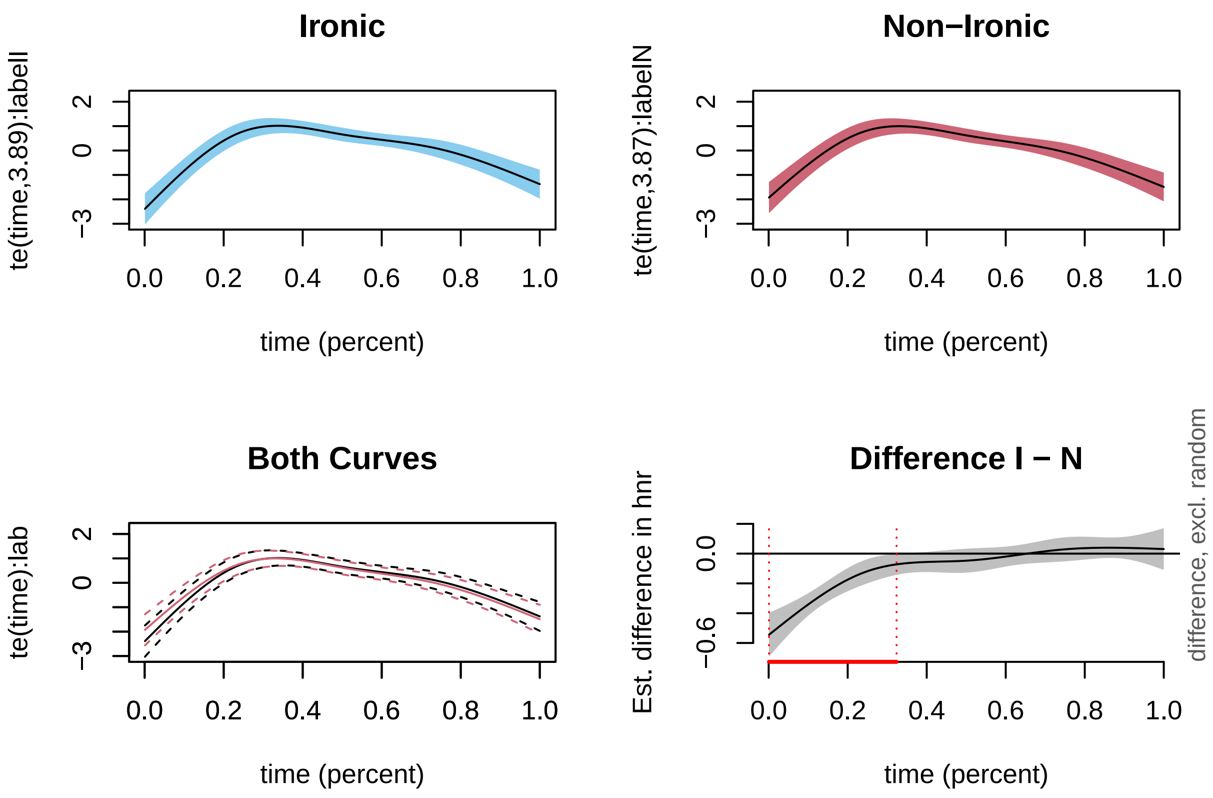
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## hnr ~ label + te(time, by = label) + s(speaker, bs = "re") +
##     s(time, speaker, bs = "fs", m = 1)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.36835    0.37552   27.61  < 2e-16 ***
## labelN       0.08447    0.01877    4.50 6.81e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                    edf  Ref.df         F p-value
## te(time):labelI  3.894   3.920    48.378  <2e-16 ***
## te(time):labelN  3.870   3.899    40.113  <2e-16 ***
## s(speaker)       5.391  11.000     0.966  <2e-16 ***
## s(time,speaker) 79.357 107.000 19070.733  0.0219 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## R-sq.(adj) =  0.0717   Deviance explained = 7.19%
## fREML = 1.402e+06  Scale est. = 36.384    n = 435900
```

summary and plots

```
cbPalette = c("#88CCEE", "#CC6677","#DDCC77","#117733","#332288","#AA4499",
              "#44AA99","#999933","#882255","#661100","#6699CC","#888888")

par(mfrow=c(2,2))
plot(m1hnr, select=1, main = "Ironic", shade=TRUE, xlab="time (percent)",
     shade.col = cbPalette[1])
plot(m1hnr, select=2, main = "Non-Ironic", shade=TRUE, xlab="time (percent)",
     shade.col = cbPalette[2])
plot(m1hnr, select=1, ylab="te(time):lab", xlab="time (percent)",
     main = "Both Curves");
par(new=TRUE);
plot(m1hnr, select=2, ylab="", xlab="", main="", col=cbPalette[2])
plot_diff(m1hnr, view="time", shade=TRUE, xlab="time (percent)",
          comp=list(label=c("I", "N")))
```



```
hnrData$pred = predict(m1hnr)
```

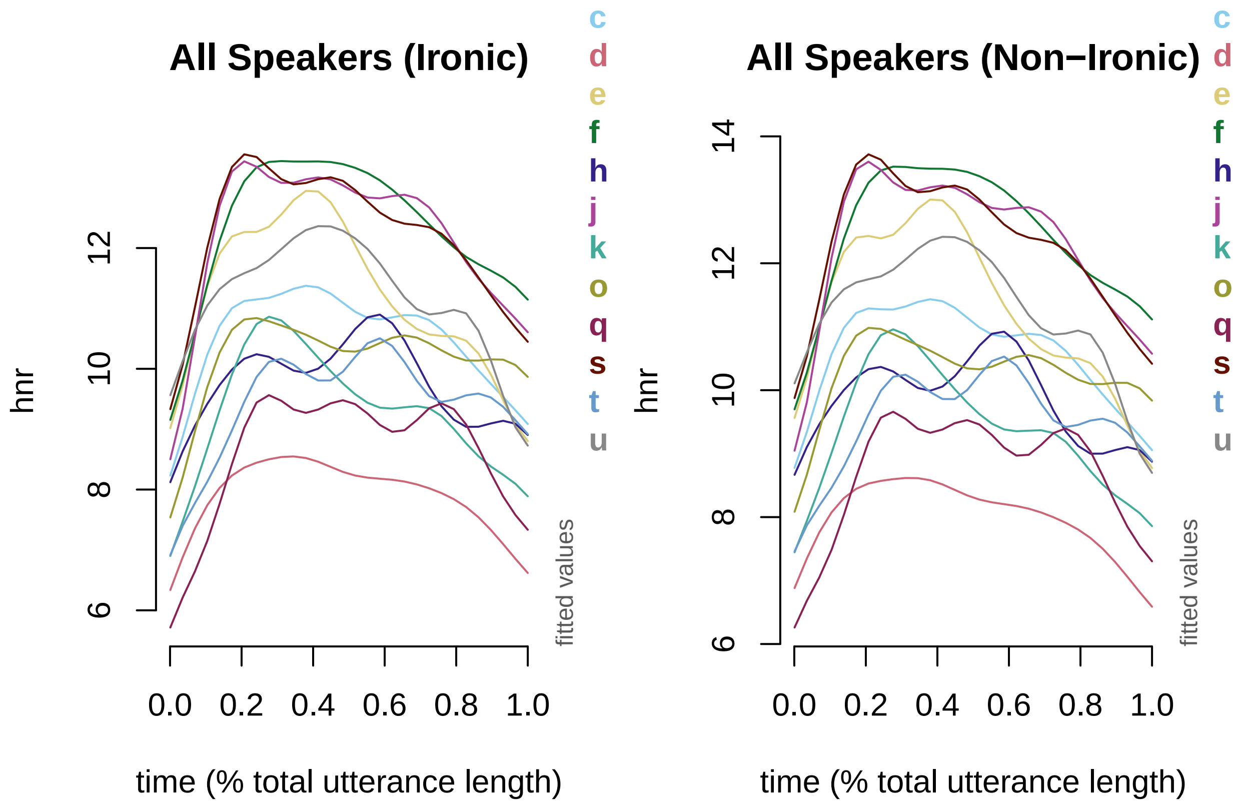
```r
par(mfrow=c(1, 2))

plot_smooth(m1hnr, view="time", cond=list("label"="I"), plot_all=c("speaker"),
            main = "All Speakers (Ironic)", col = cbPalette[1:12],
            rm.ranef=FALSE, se=0,
            xlab="time (% total utterance length)")
```

```
## Summary:
##  * label : factor; set to the value(s): I.
##  * time : numeric predictor; with 30 values ranging from 0.000755 to 1.000000.
##  * speaker : factor with 12 values; set to the value(s): c, d, e, f, h, j, k, o, q, s, ...
```

```r
plot_smooth(m1hnr, view="time", cond=list("label"="N"), plot_all=c("speaker"),
            main = "All Speakers (Non-Ironic)", col = cbPalette[1:12],
            rm.ranef=FALSE, se=0,
            xlab="time (% total utterance length)")
```

```
## Summary:
##  * label : factor; set to the value(s): N.
##  * time : numeric predictor; with 30 values ranging from 0.000755 to 1.000000.
##  * speaker : factor with 12 values; set to the value(s): c, d, e, f, h, j, k, o, q, s, ...
```



```r
par(mfrow=c(2, 2))

plot_smooth(m1hnr, view="time", cond=list("speaker"="c"), plot_all=c("label"),
            main = "Speaker C", col = cbPalette[1:2], rm.ranef=FALSE, se=0,
            xlab="time (% total utterance length)")
```

```
## Summary:
##  * label : factor; set to the value(s): I, N.
##  * time : numeric predictor; with 30 values ranging from 0.000755 to 1.000000.
##  * speaker : factor; set to the value(s): c.
```

```r
plot_smooth(m1hnr, view="time", cond=list("speaker"="d"), plot_all=c("label"),
            main = "Speaker D", col = cbPalette[1:2], rm.ranef=FALSE, se=0,
            xlab="time (% total utterance length)")
```

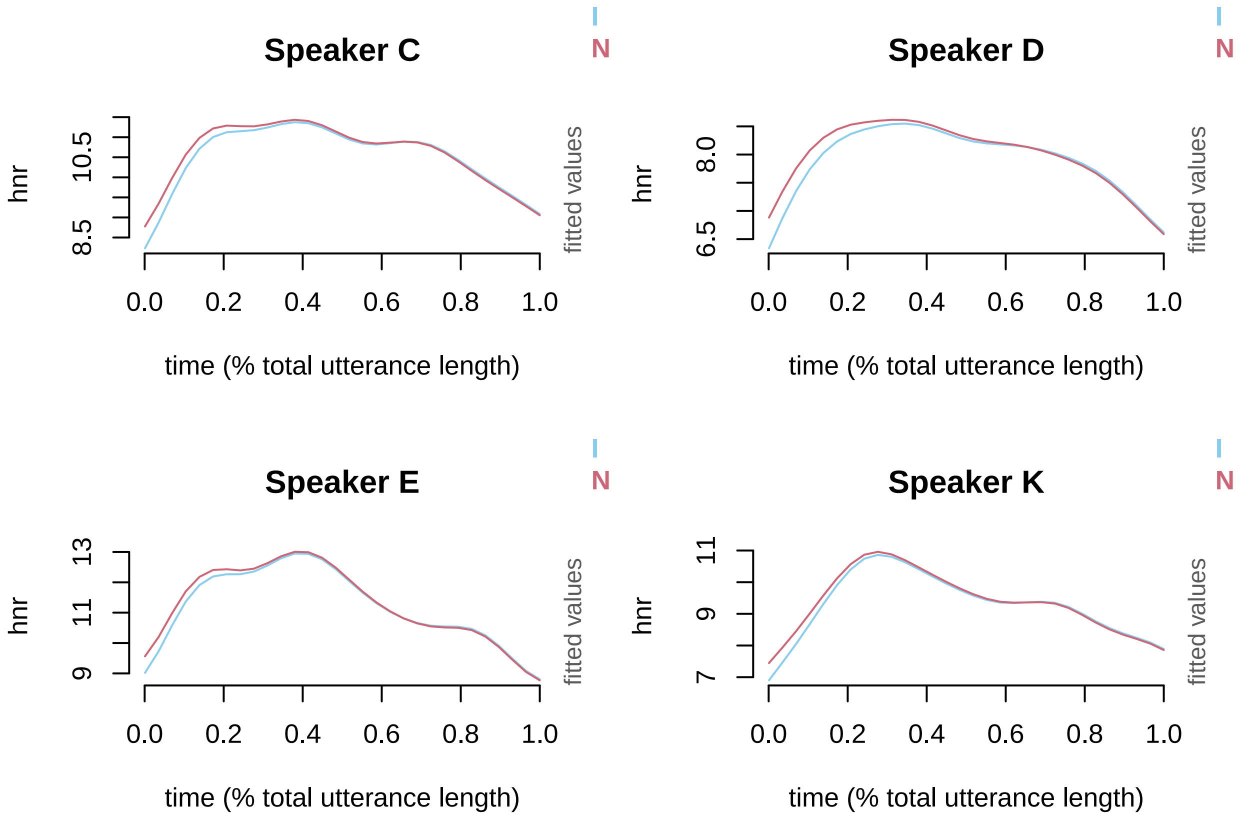
```
## Summary:
##  * label : factor; set to the value(s): I, N.
##  * time : numeric predictor; with 30 values ranging from 0.000755 to 1.000000.
##  * speaker : factor; set to the value(s): d.
```

```r
plot_smooth(m1hnr, view="time", cond=list("speaker"="e"), plot_all=c("label"),
            main = "Speaker E", col = cbPalette[1:2], rm.ranef=FALSE, se=0,
            xlab="time (% total utterance length)")
```

```
## Summary:
##  * label : factor; set to the value(s): I, N.
##  * time : numeric predictor; with 30 values ranging from 0.000755 to 1.000000.
##  * speaker : factor; set to the value(s): e.
```

```r
plot_smooth(m1hnr, view="time", cond=list("speaker"="k"), plot_all=c("label"),
            main = "Speaker K", col = cbPalette[1:2], rm.ranef=FALSE, se=0,
            xlab="time (% total utterance length)")
```

```
## Summary:
##  * label : factor; set to the value(s): I, N.
##  * time : numeric predictor; with 30 values ranging from 0.000755 to 1.000000.
##  * speaker : factor; set to the value(s): k.
```


Speaker C
I
N
hnr
8.5
10.5
fitted values
0.0
0.2
0.4
0.6
0.8
1.0
time (% total utterance length)
Speaker D
I
N
hnr
6.5
8.0
fitted values
0.0
0.2
0.4
0.6
0.8
1.0
time (% total utterance length)
Speaker E
I
N
hnr
9
11
13
fitted values
0.0
0.2
0.4
0.6
0.8
1.0
time (% total utterance length)
Speaker K
I
N
hnr
7
9
11
fitted values
0.0
0.2
0.4
0.6
0.8
1.0
time (% total utterance length)