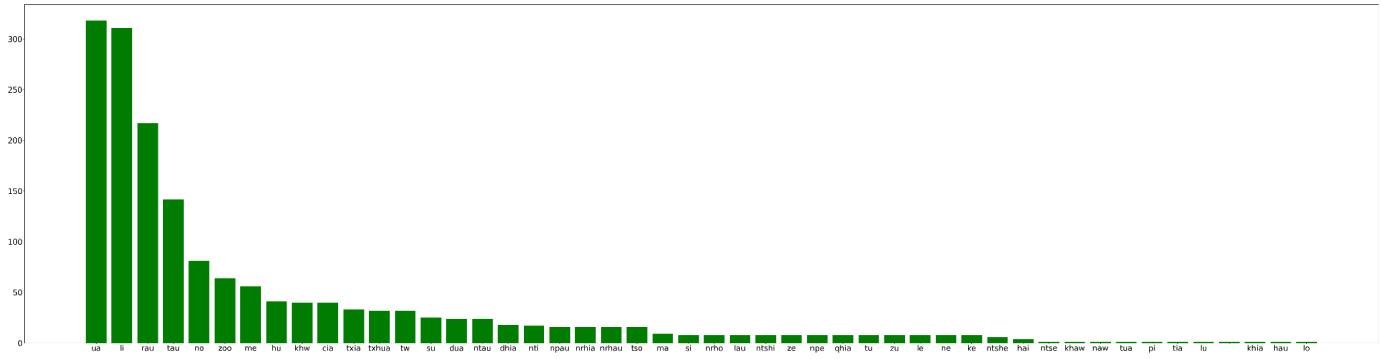
By-Word Modeling

In the following pages, titled Word_Type_Top6, GAMs are plotted for each of the eight tones in White Hmong, which each predict F0 (normalized with the ST-AvgF0 method) based on only normalized time, producing a smooth for each of the 7 possible values of NormalizedWord. For tokens where the word variable contains one of the six most common word types for the given tone, NormalizedWord is set to that word. Otherwise, NormalizedWord is X. The remainder of this preface will be used to justify this six-word threshold.

A relatively small number of words being modeled separately allows for models to be run quickly, in order to identify whether or not we should continue examining the tone contours of individual words. Second, to identify where exactly to place the threshold, the exact counts of each word type attested in the data for each tone were plotted, as is given below (the plot below is for mid-tone (33 Ø) words). In all cases except one[1], these plots showed a zipf-like distribution, in which the most common types are very common, and then at the other end there is a long tail of very rare words, which is further justification to have a cutoff threshold, at which point all less-common words are grouped into one type. The number six was chosen because, for all tones, it includes not just the most common words, but also some less-common words, allowing the models shown below not just to represent the words with the most data, but also some with fewer examples (while still containing enough examples to successfully model the words.



[1] The -d tone has two word types attested in the data, and thus did not show a zipf distribution.

Word_Type_Top6

Henry Heyden

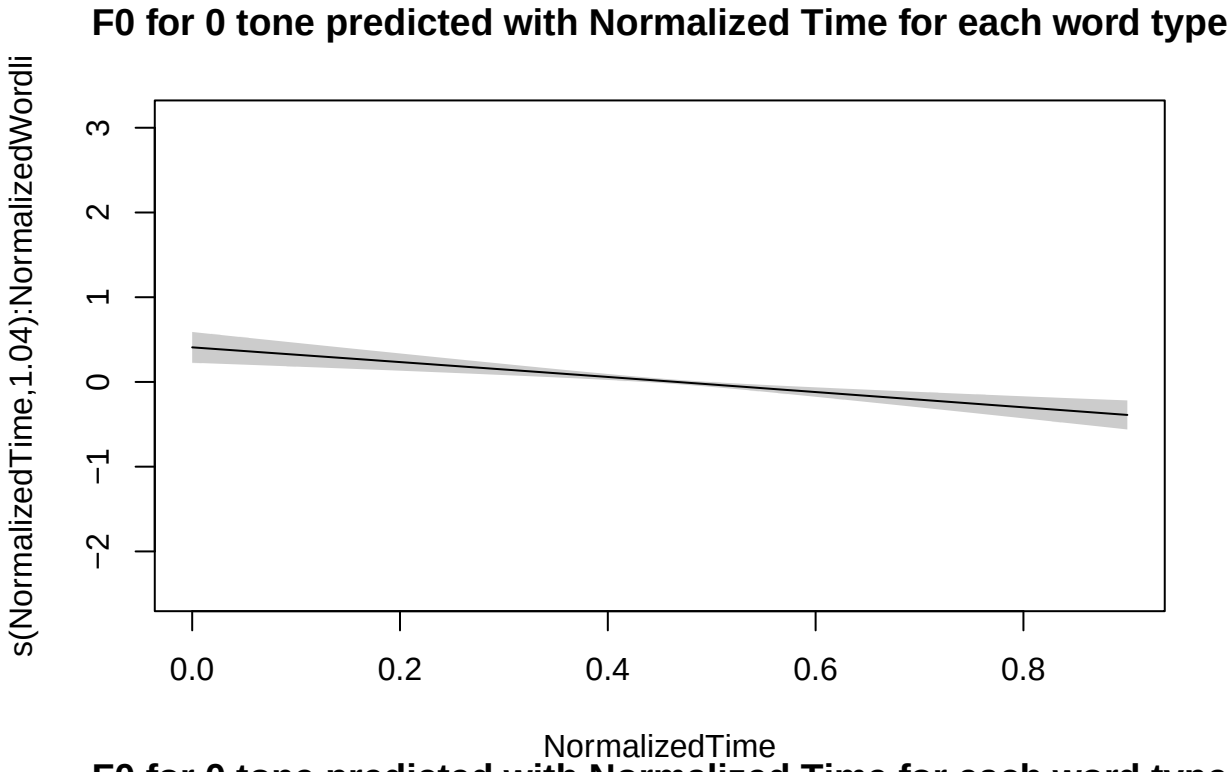
2025-08-13

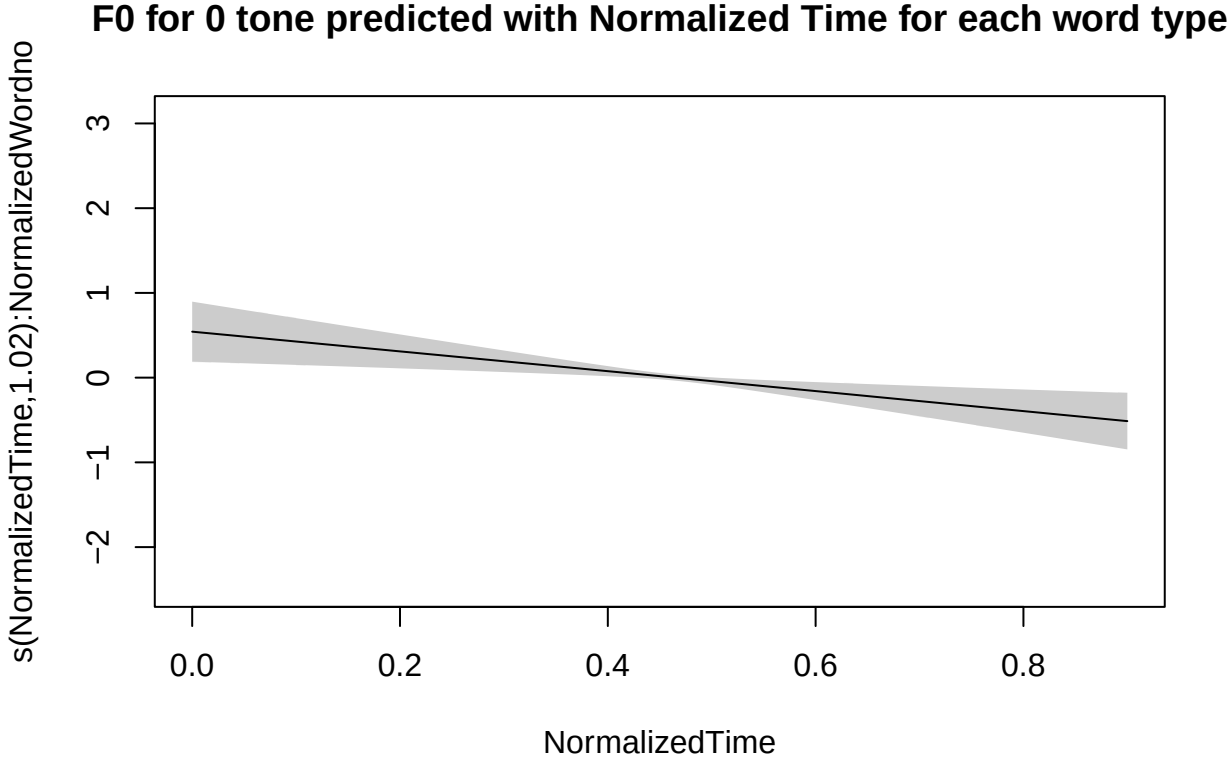
This file is for GAMs that are by = word

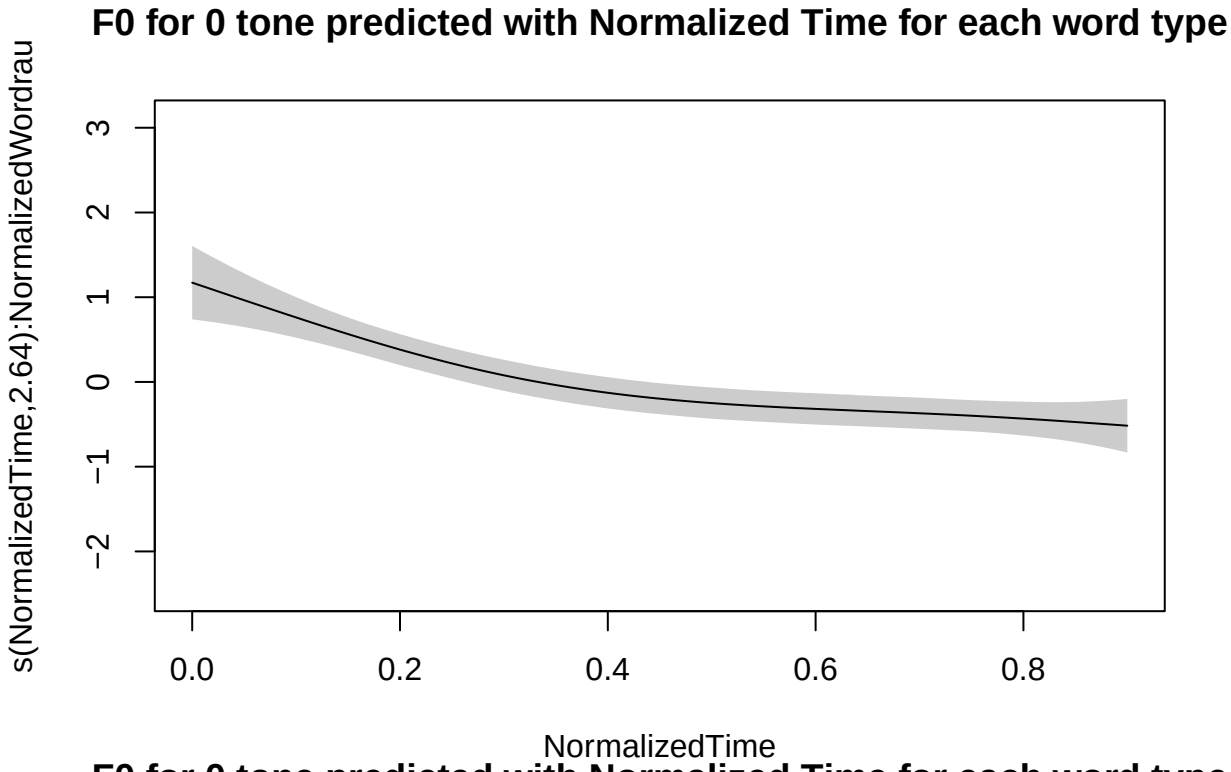
```
# Model
gamWord_0 = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongData0, method = 'REML')
summary(gamWord_0)
```

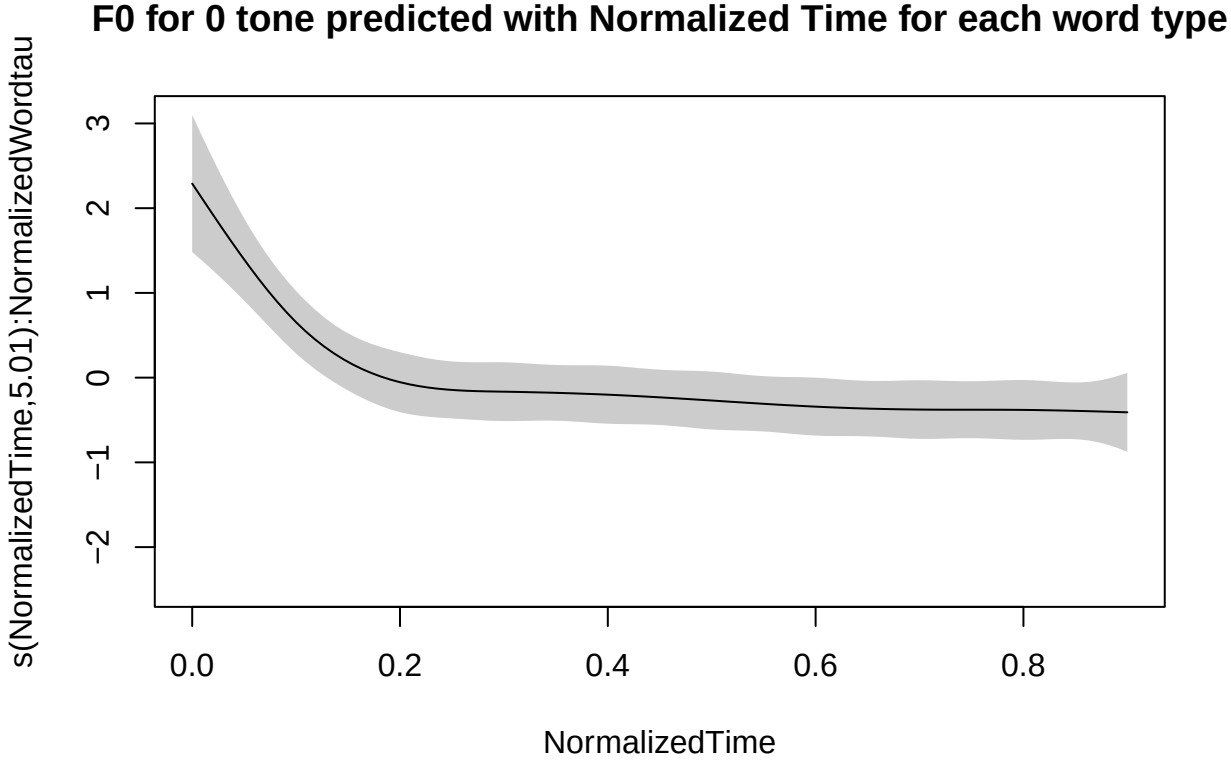
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.05520    0.02395   2.305   0.0212 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                       edf Ref.df      F  p-value
## s(NormalizedTime):NormalizedWordli  1.045  1.088 19.939 6.22e-06 ***
## s(NormalizedTime):NormalizedWordno  1.022  1.044  9.294  0.00216 **
## s(NormalizedTime):NormalizedWordrau 2.638  3.299 13.768  < 2e-16 ***
## s(NormalizedTime):NormalizedWordtau 5.006  6.163  6.031 3.27e-06 ***
## s(NormalizedTime):NormalizedWordua  3.523  4.378 62.154  < 2e-16 ***
## s(NormalizedTime):NormalizedWordX   3.331  4.144 53.838  < 2e-16 ***
## s(NormalizedTime):NormalizedWordzoo 1.063  1.124  4.070  0.03463 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0359   Deviance explained = 3.69%
## -REML =  41339  Scale est. = 9.2776    n = 16313
```

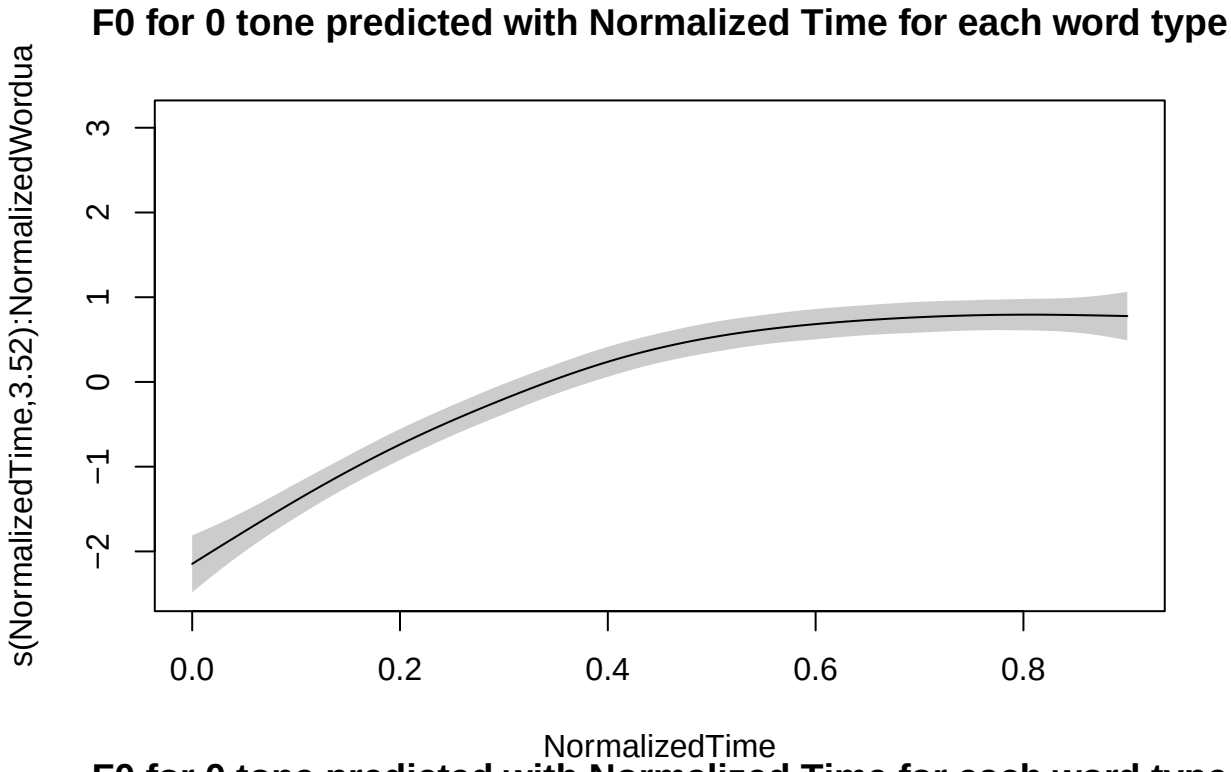
```
# Visualize Model
plot(gamWord_0, shade = TRUE, main = 'F0 for 0 tone predicted with Normalized Time for each word type')
```

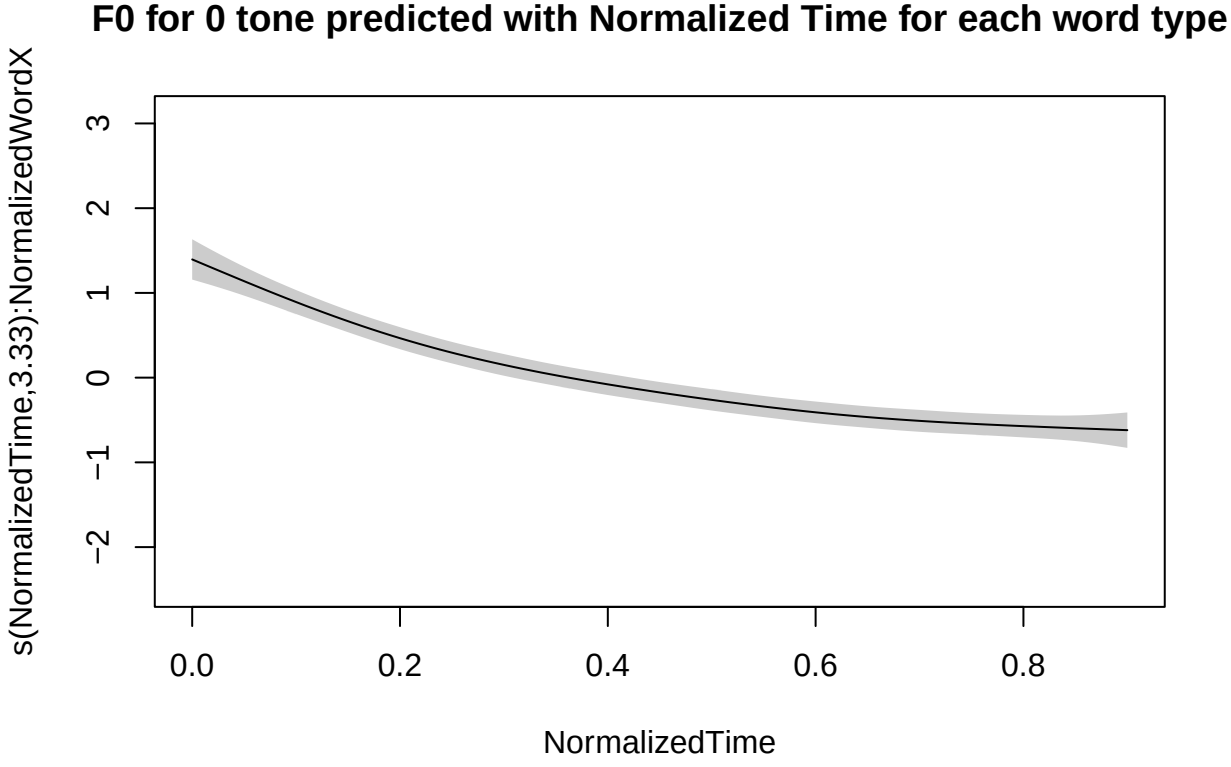

F0 for 0 tone predicted with Normalized Time for each word type
s(NormalizedTime,1.04):NormalizedWordli
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime

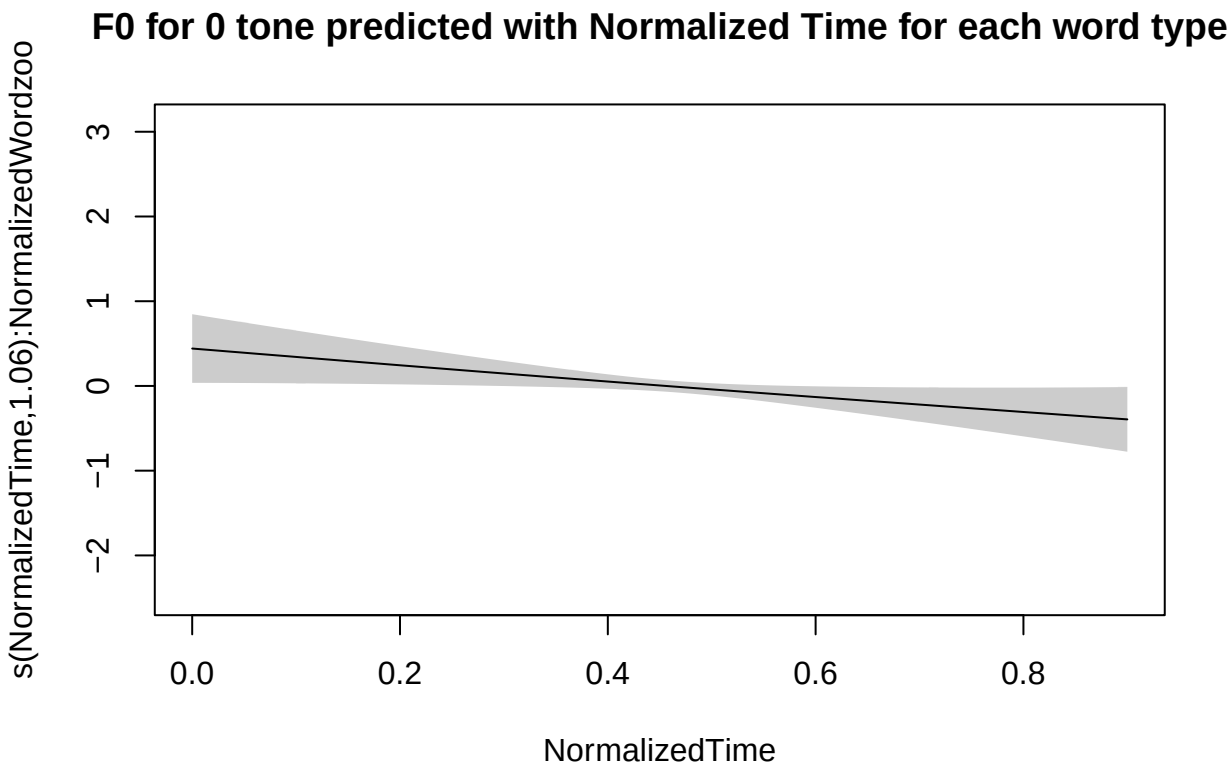

F0 for 0 tone predicted with Normalized Time for each word type
s(NormalizedTime,1.02):NormalizedWordno
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for 0 tone predicted with Normalized Time for each word type
s(NormalizedTime,2.64):NormalizedWordrau
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for 0 tone predicted with Normalized Time for each word type
s(NormalizedTime,5.01):NormalizedWordtau
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for 0 tone predicted with Normalized Time for each word type
s(NormalizedTime,3.52):NormalizedWordua
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for 0 tone predicted with Normalized Time for each word type
s(NormalizedTime,3.33):NormalizedWordX
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime

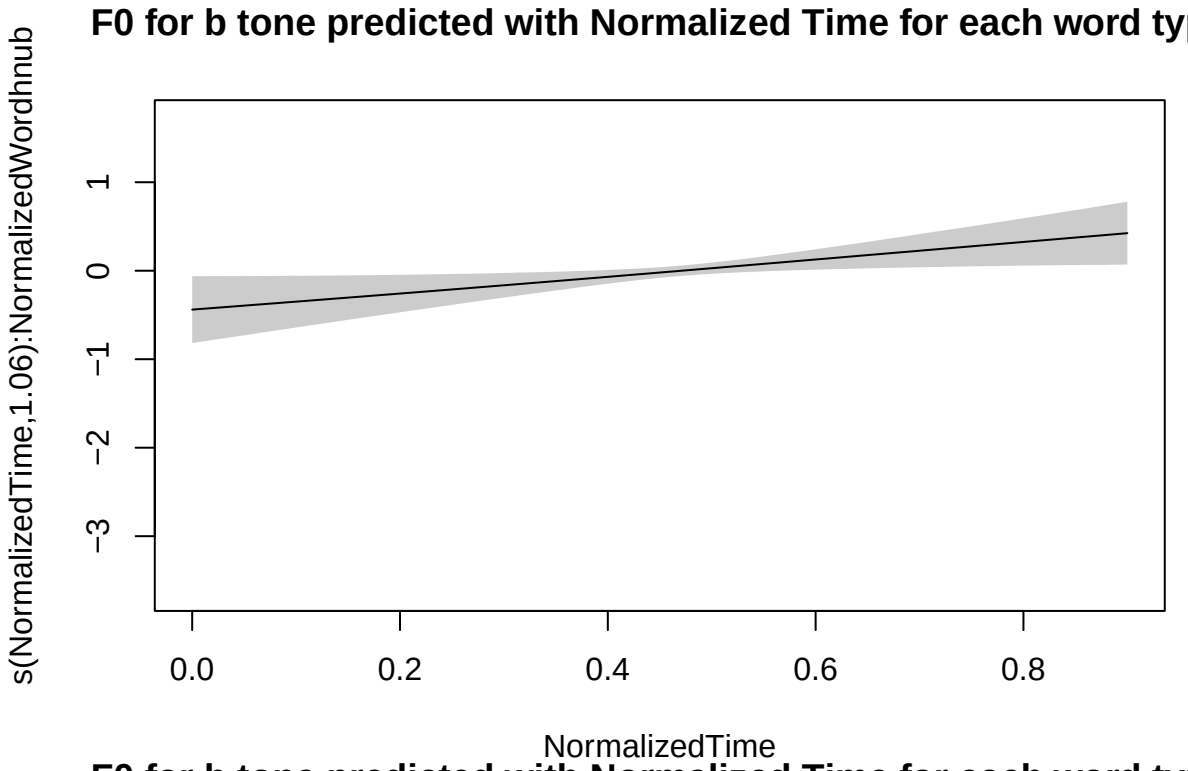


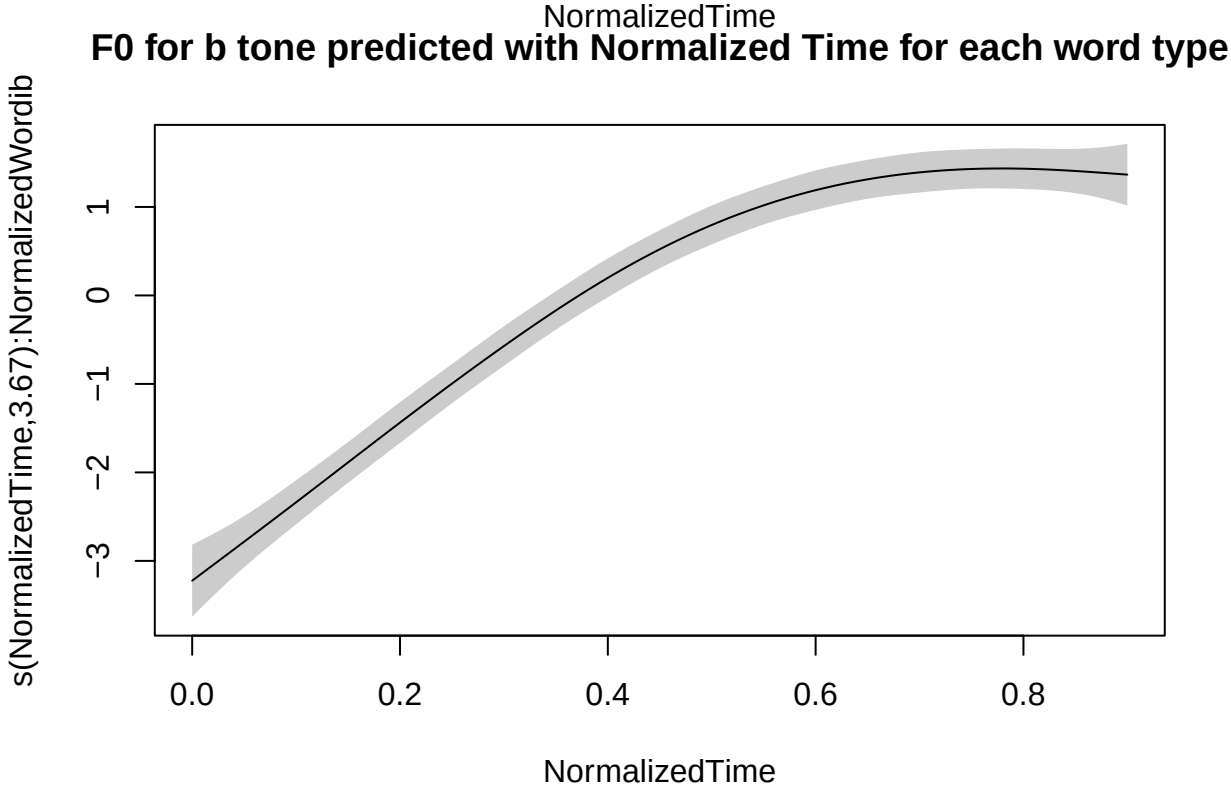

```
# Model
gamWord_b = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataB, method = 'REML')
summary(gamWord_b)
```

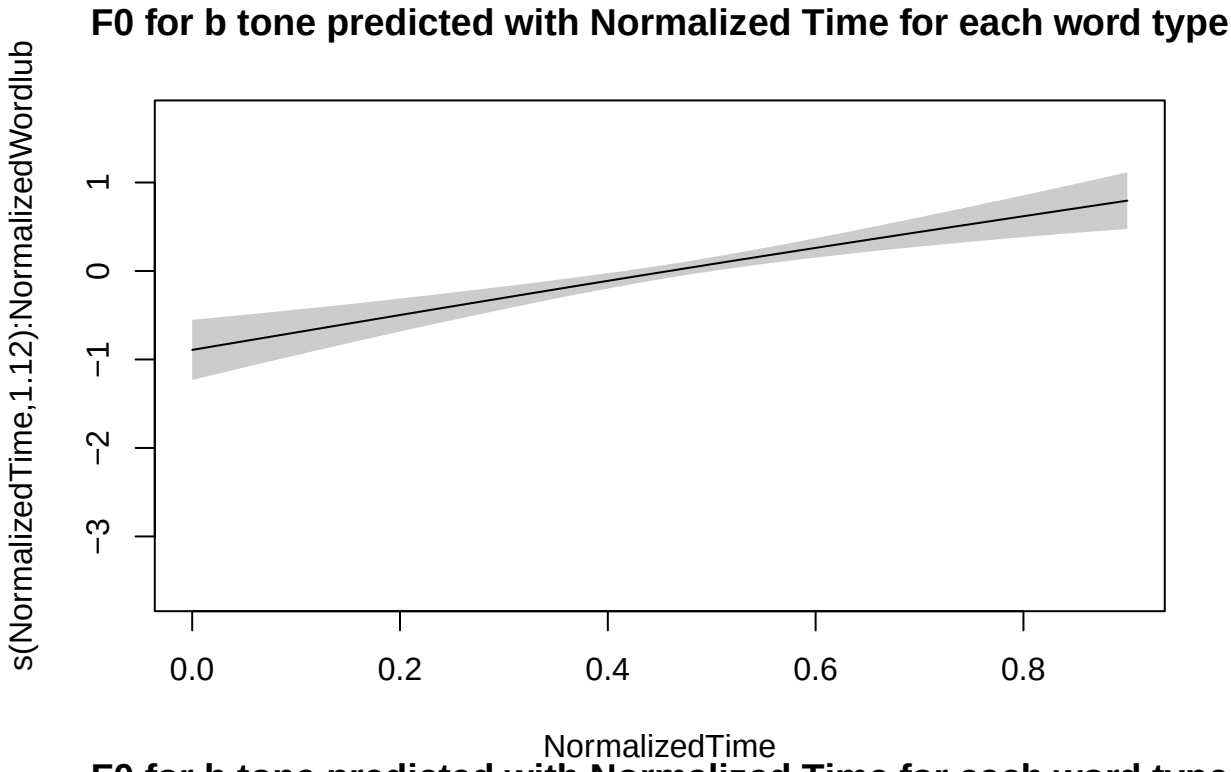
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.98247    0.02812   70.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                         edf Ref.df       F  p-value
## s(NormalizedTime):NormalizedWordhnub  1.056  1.110   5.511   0.0184 *
## s(NormalizedTime):NormalizedWordib    3.669  4.552 112.632  < 2e-16 ***
## s(NormalizedTime):NormalizedWordlub   1.124  1.237  22.907 7.40e-07 ***
## s(NormalizedTime):NormalizedWordnyob  1.316  1.565  17.514 1.56e-06 ***
## s(NormalizedTime):NormalizedWordteb   4.628  5.721   8.559  < 2e-16 ***
## s(NormalizedTime):NormalizedWordthiab 2.327  2.902   1.744   0.2229
## s(NormalizedTime):NormalizedWordX     5.686  6.888  22.987  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0491   Deviance explained = 5.03%
```

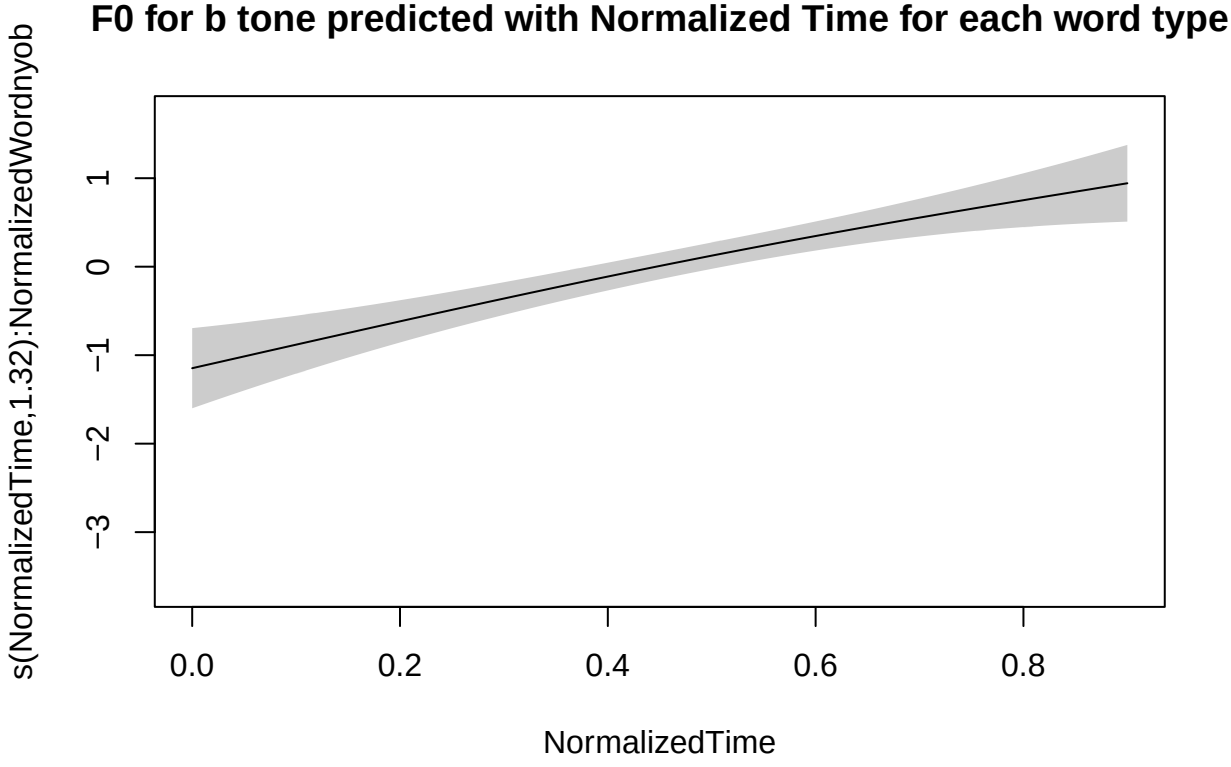
```
## -REML =  40364  Scale est. = 11.943    n = 15171
```

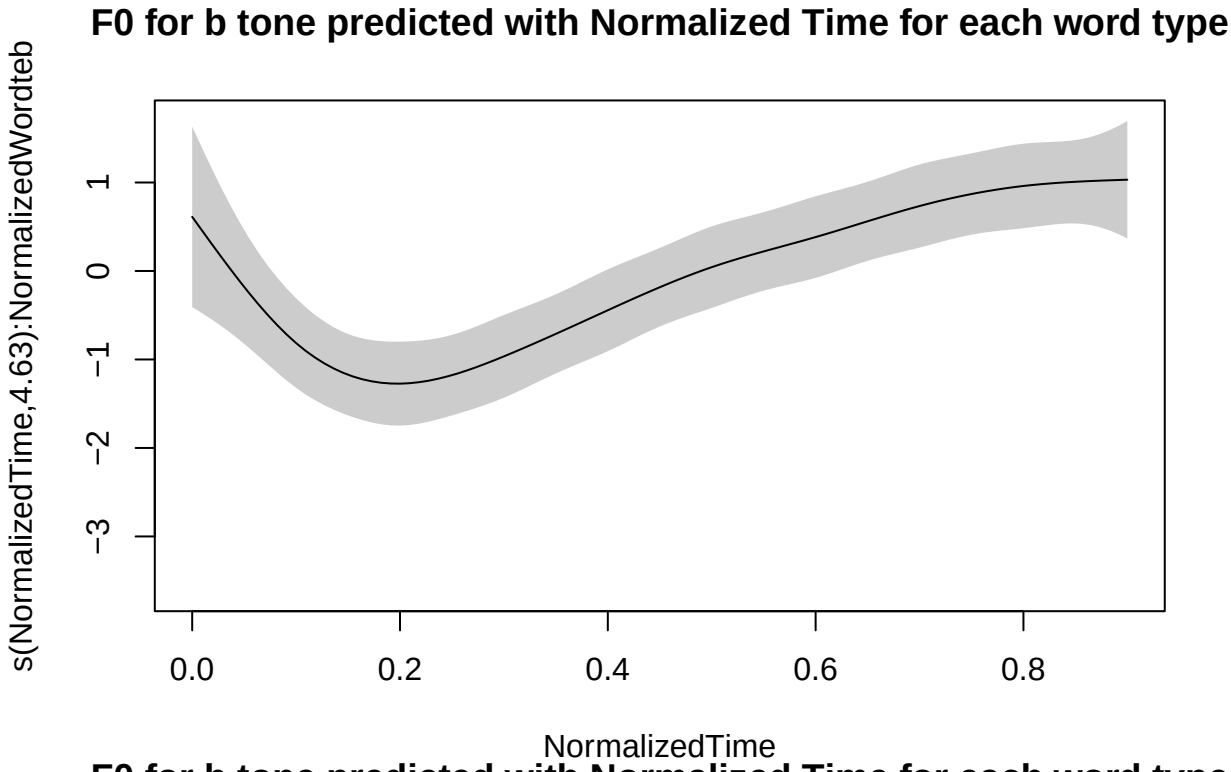
```
# Visualize Model
plot(gamWord_b, shade = TRUE, main = 'F0 for b tone predicted with Normalized Time for each word type')
```

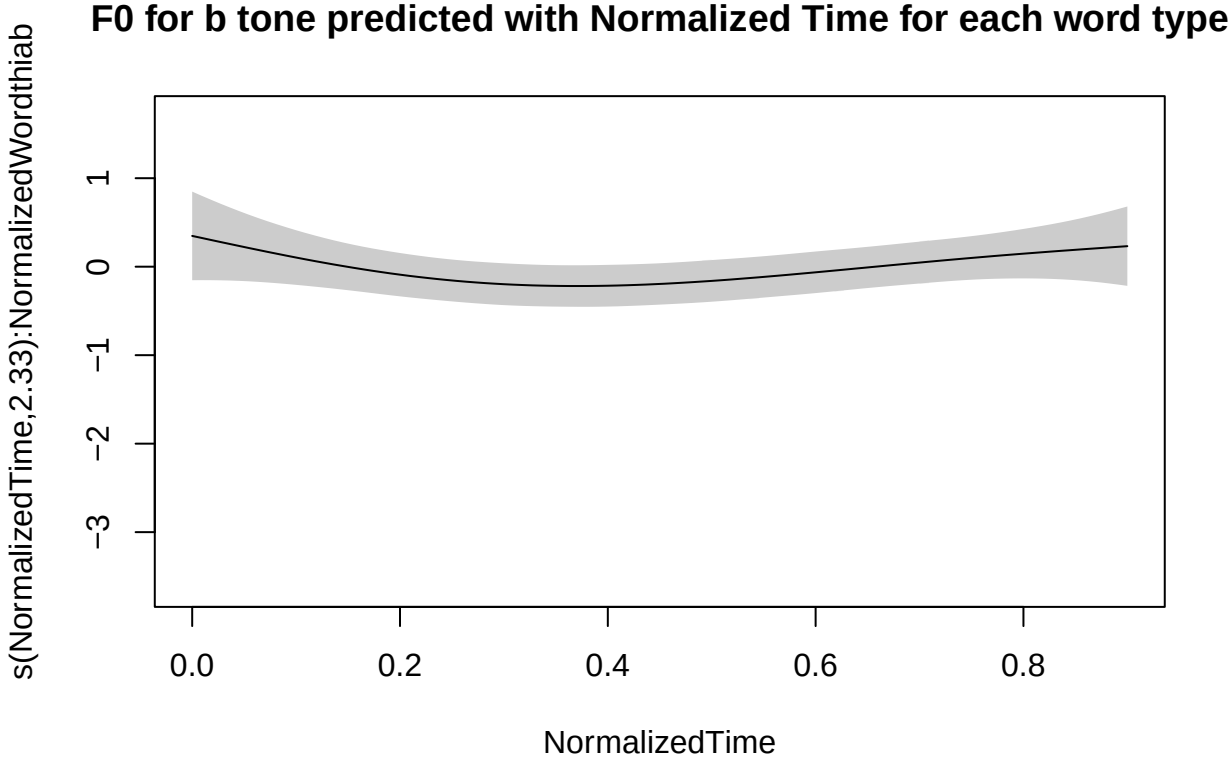


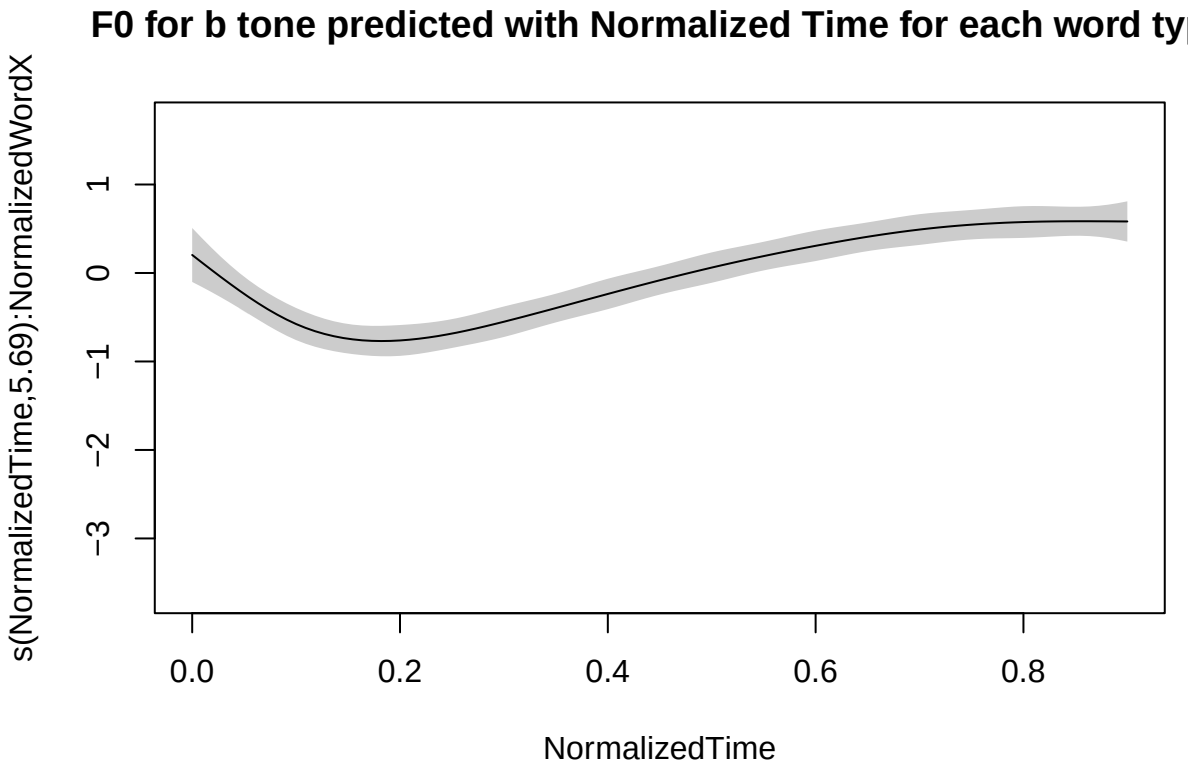



F0 for b tone predicted with Normalized Time for each word type
s(NormalizedTime,1.12):NormalizedWordlub
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for b tone predicted with Normalized Time for each word type
s(NormalizedTime,1.32):NormalizedWordnyob
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for b tone predicted with Normalized Time for each word type
s(NormalizedTime,4.63):NormalizedWordteb
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
NormalizedTime

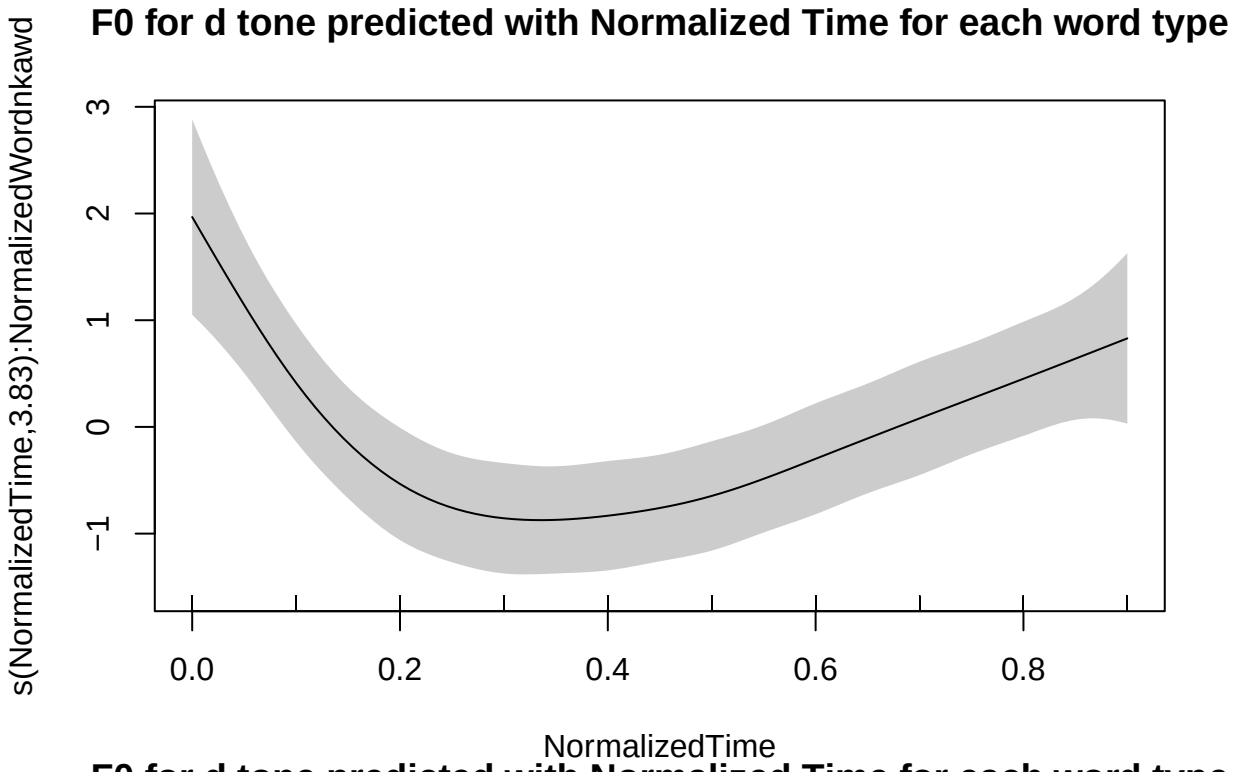

F0 for b tone predicted with Normalized Time for each word type
s(NormalizedTime,2.33):NormalizedWordthiab
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
NormalizedTime

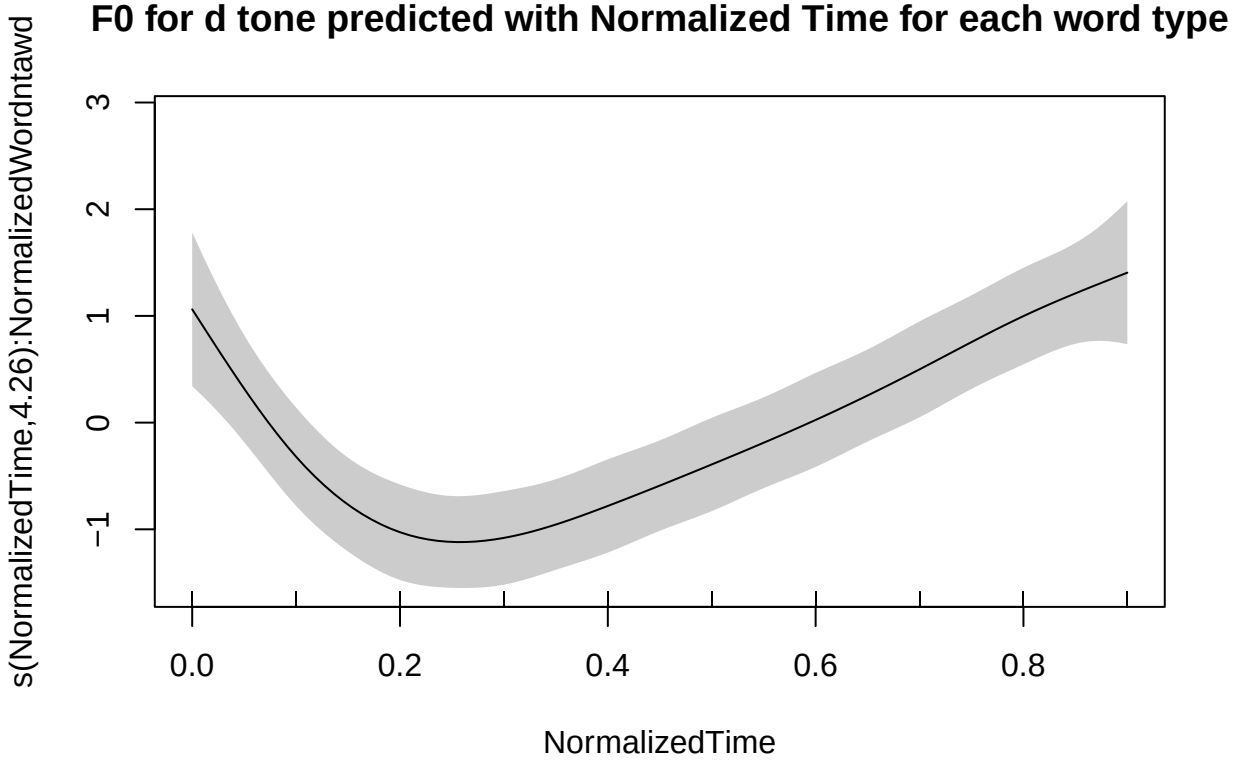



```
# Model
gamWord_d = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataD, method = 'REML')
summary(gamWord_d)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.90092    0.09449  -20.12   <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(NormalizedTime):NormalizedWordnkawd 3.833  4.753  6.282 2.64e-05 ***
## s(NormalizedTime):NormalizedWordntawd 4.256  5.259 10.059  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0928   Deviance explained = 10.2%
## -REML = 1898.7  Scale est. = 7.0423    n = 789
```

```
# Visualize Model
plot(gamWord_d, shade = TRUE, main = 'F0 for d tone predicted with Normalized Time for each word type')
```



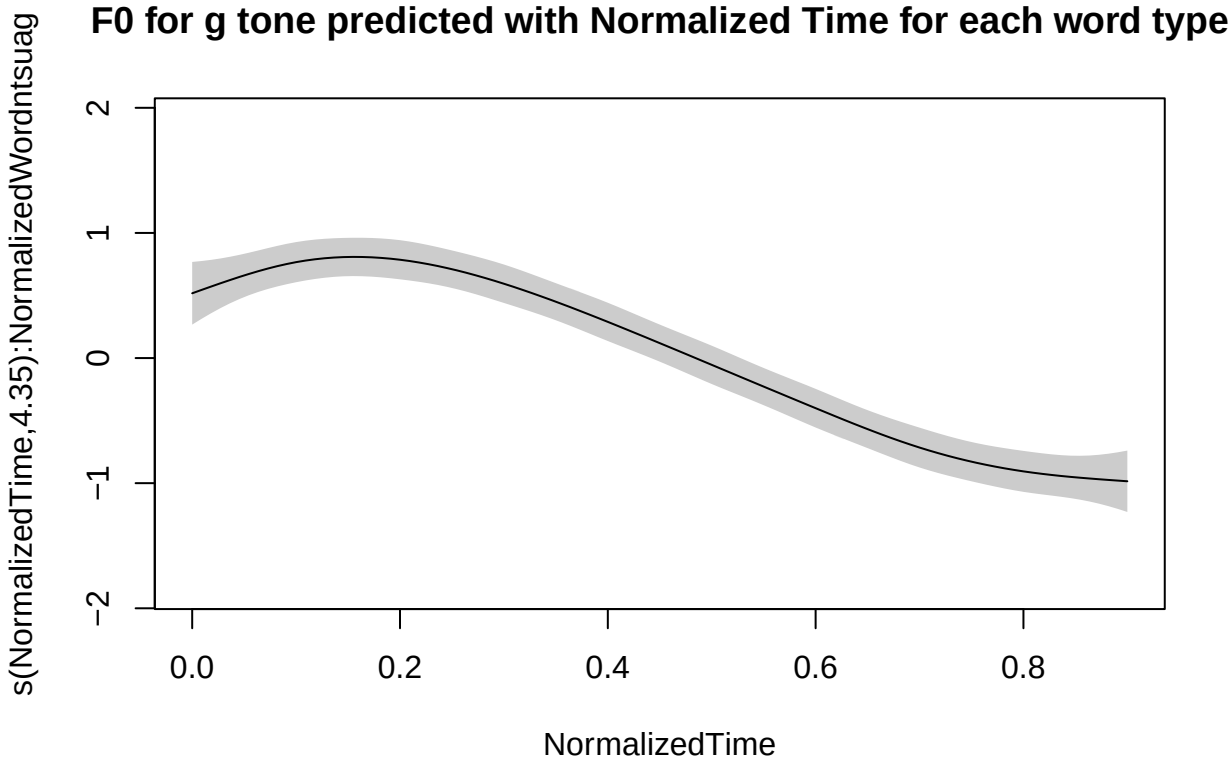


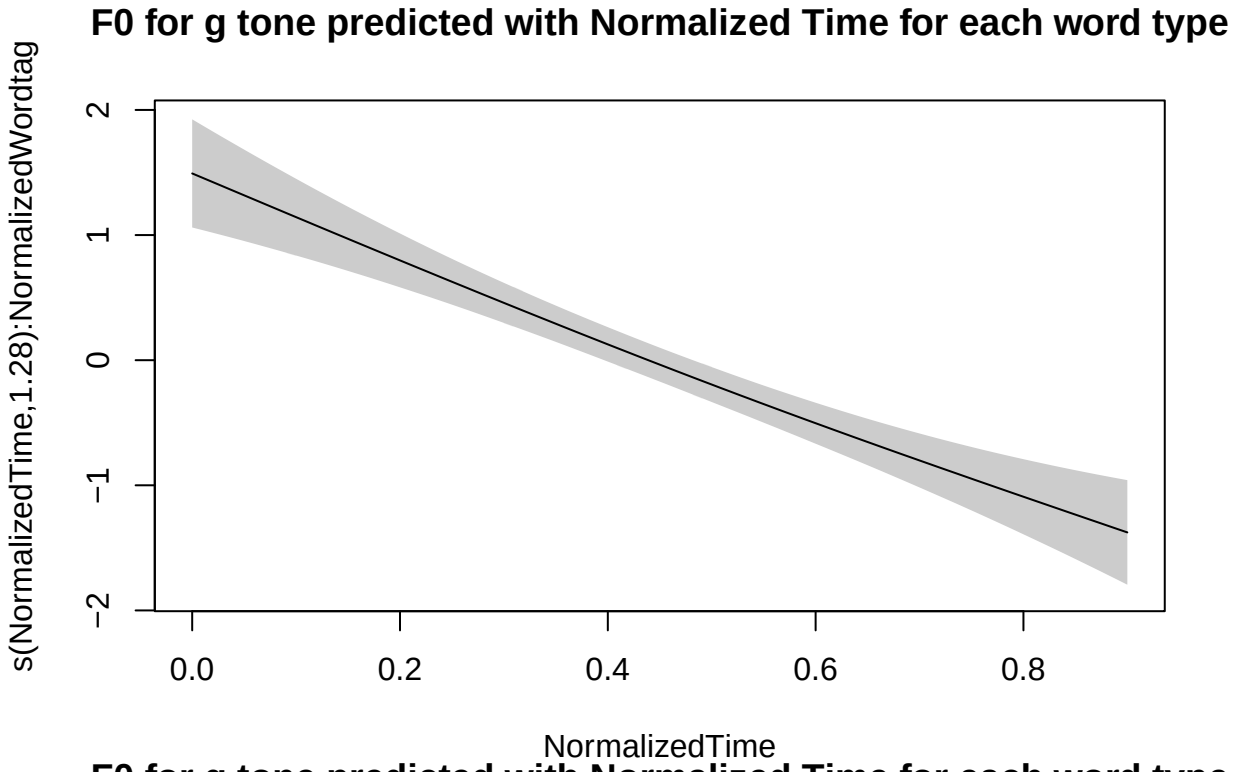

```
# Model
gamWord_g = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataG, method = 'REML')
summary(gamWord_g)
```

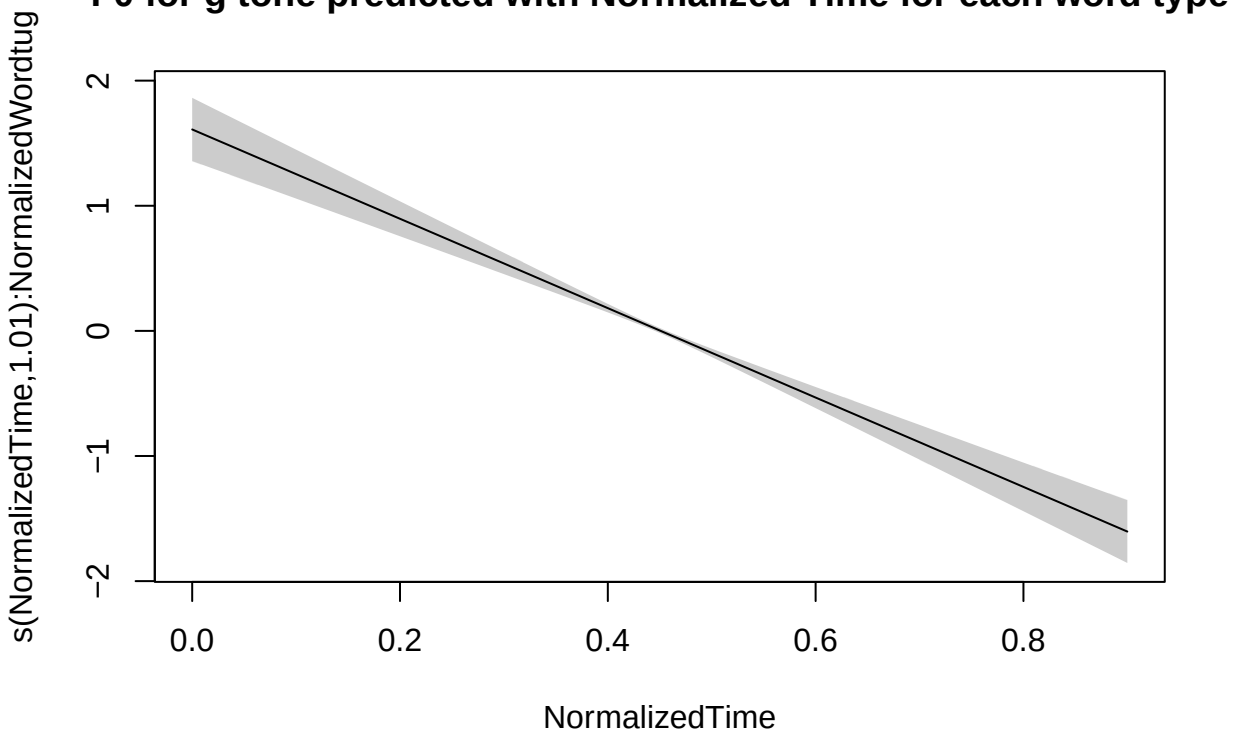
```
##
## Family: gaussian
```

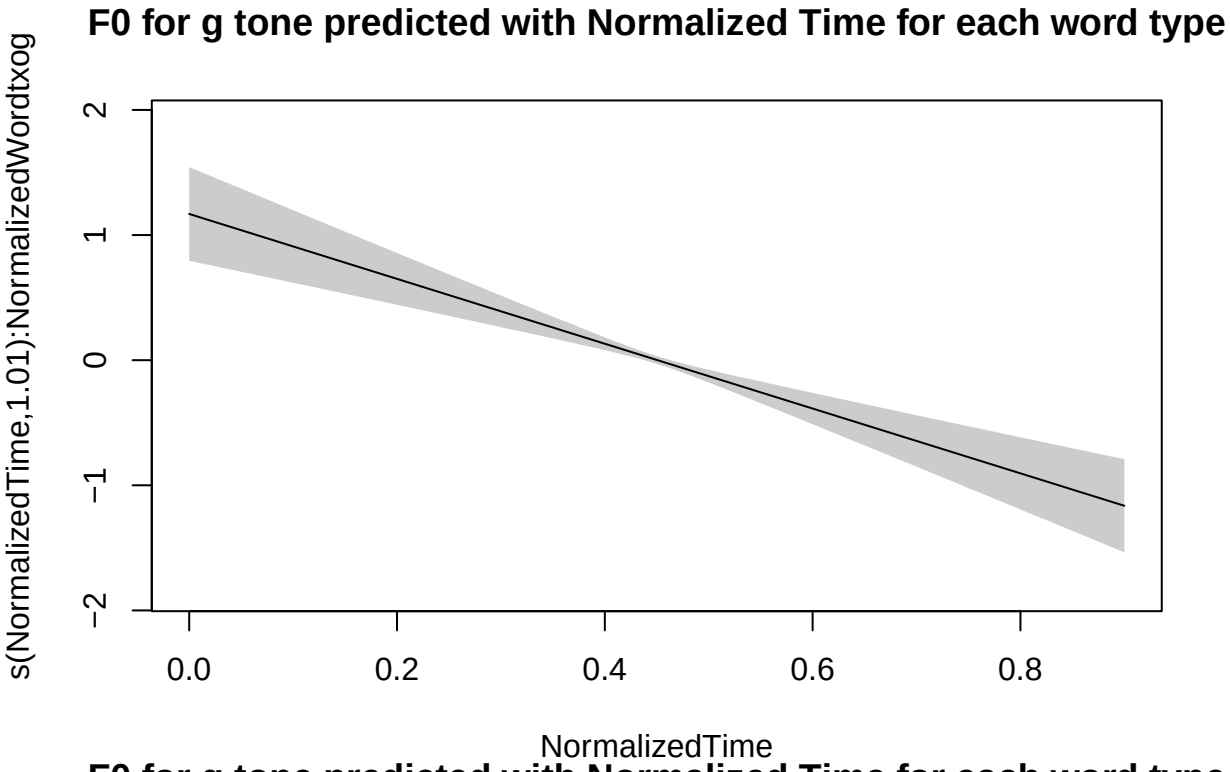
```
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.99282    0.02159   92.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                           edf Ref.df       F  p-value
## s(NormalizedTime):NormalizedWordntsuag 4.355  5.378  45.712  < 2e-16 ***
## s(NormalizedTime):NormalizedWordtag    1.282  1.512  36.806  < 2e-16 ***
## s(NormalizedTime):NormalizedWordtug    1.013  1.025 161.112  < 2e-16 ***
## s(NormalizedTime):NormalizedWordtxog   1.014  1.028  38.499  < 2e-16 ***
## s(NormalizedTime):NormalizedWordX      3.284  4.086  72.012  < 2e-16 ***
## s(NormalizedTime):NormalizedWordyog    3.085  3.834   6.362 6.28e-05 ***
## s(NormalizedTime):NormalizedWordzuag   2.940  3.656  10.552  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0744   Deviance explained = 7.59%
## -REML =  23826  Scale est. = 4.9875    n = 10711
```

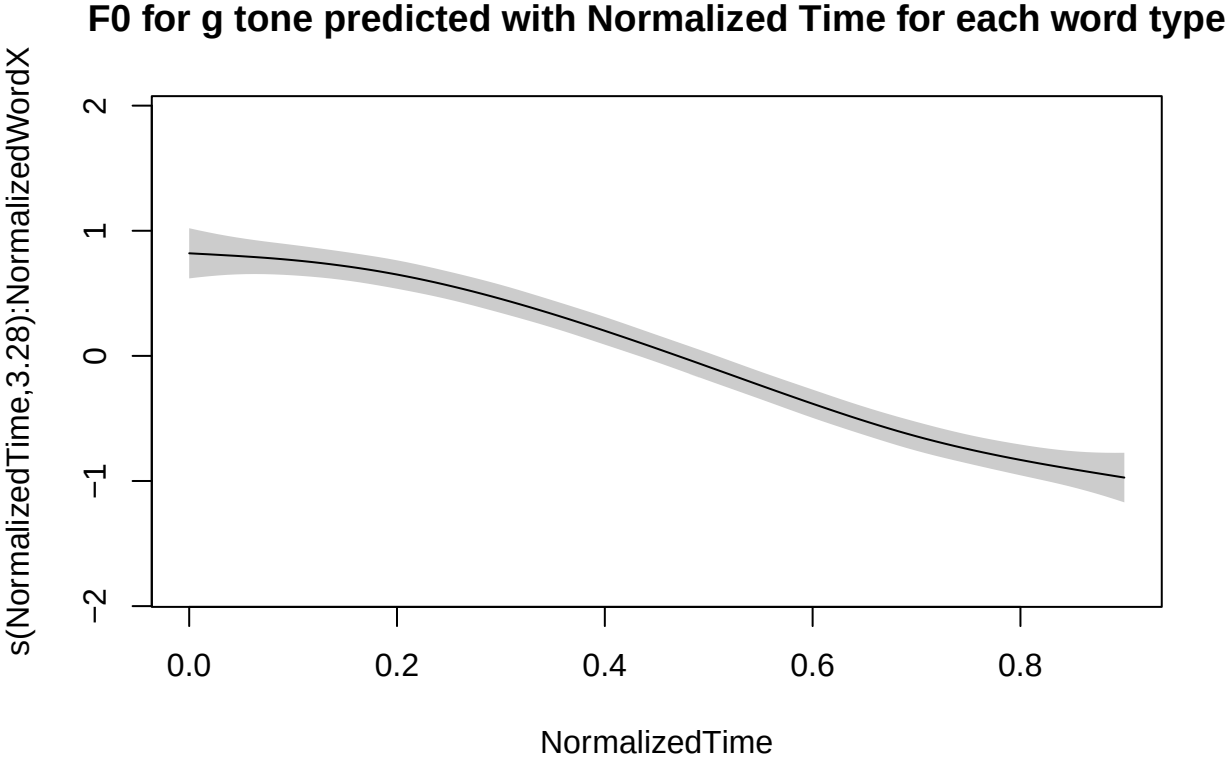
```r
# Visualize Model
plot(gamWord_g, shade = TRUE, main = 'F0 for g tone predicted with Normalized Time for each word type')
```

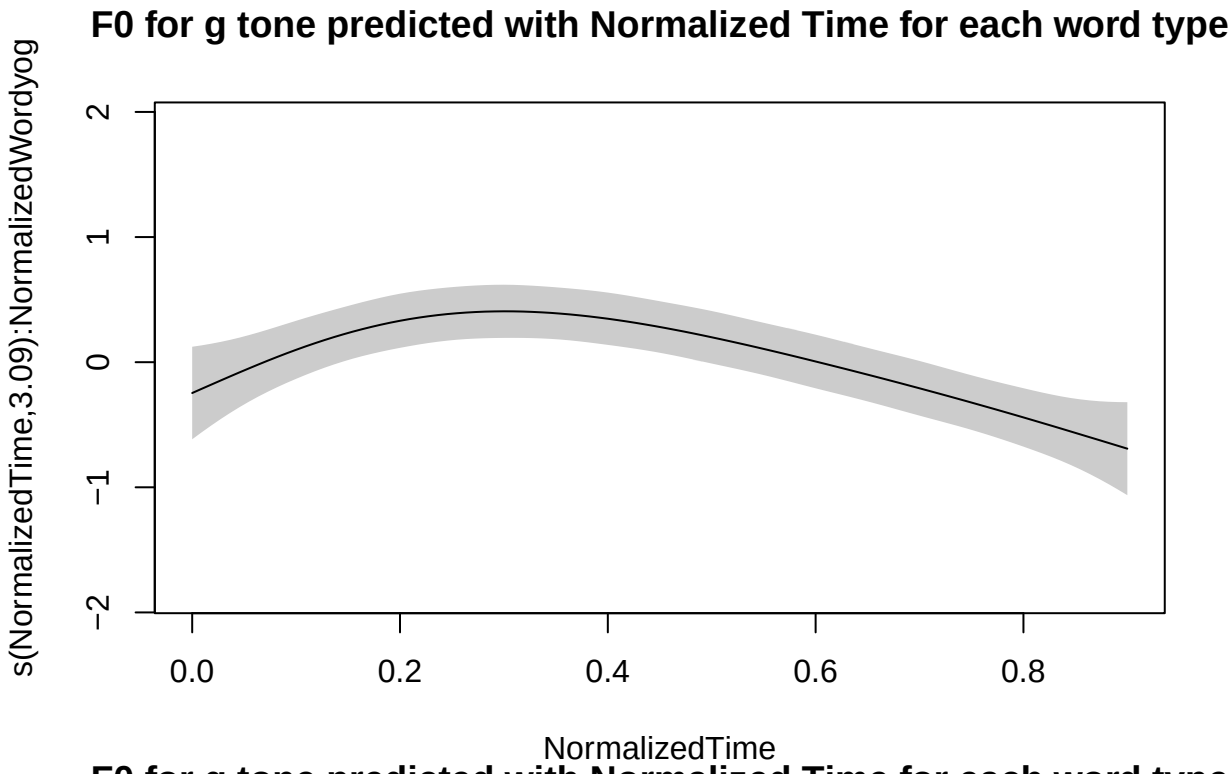


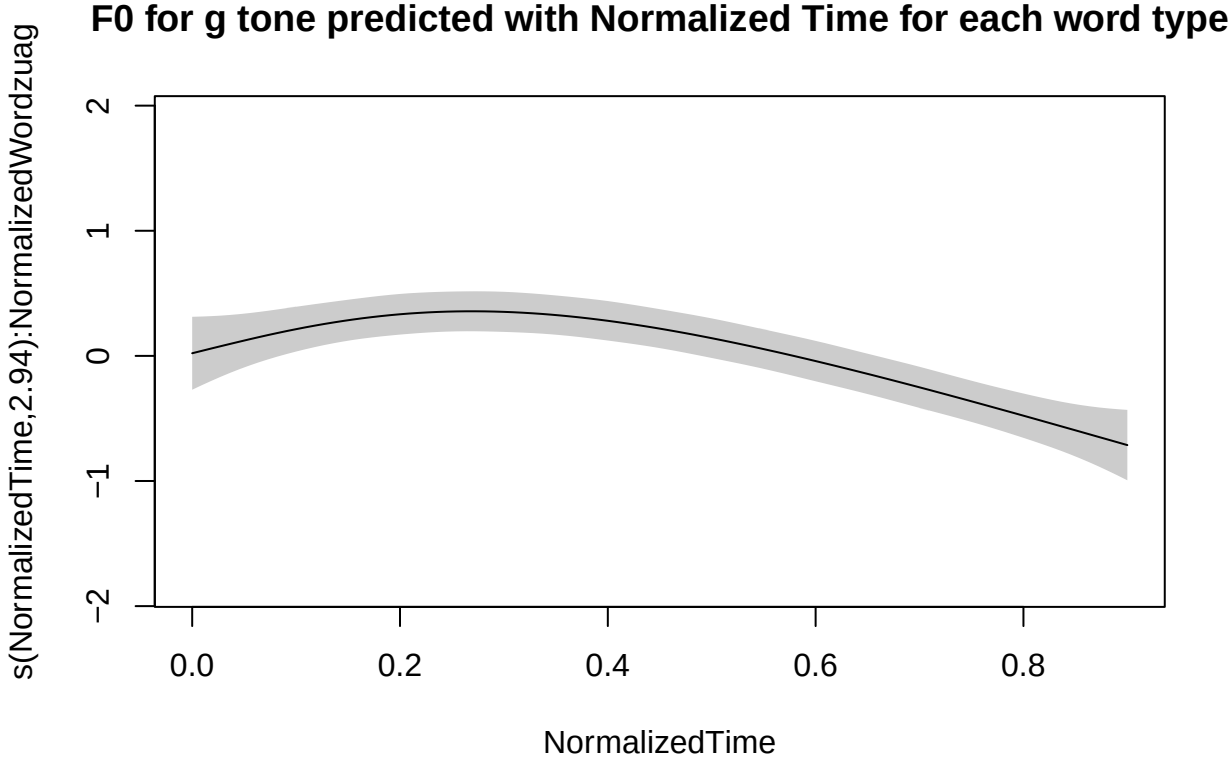

F0 for g tone predicted with Normalized Time for each word type
s(NormalizedTime,1.28):NormalizedWordtag
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for g tone predicted with Normalized Time for each word type
s(NormalizedTime,1.01):NormalizedWordtug
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for g tone predicted with Normalized Time for each word type
s(NormalizedTime,1.01):NormalizedWordtxog
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for g tone predicted with Normalized Time for each word type
s(NormalizedTime,3.28):NormalizedWordX
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime



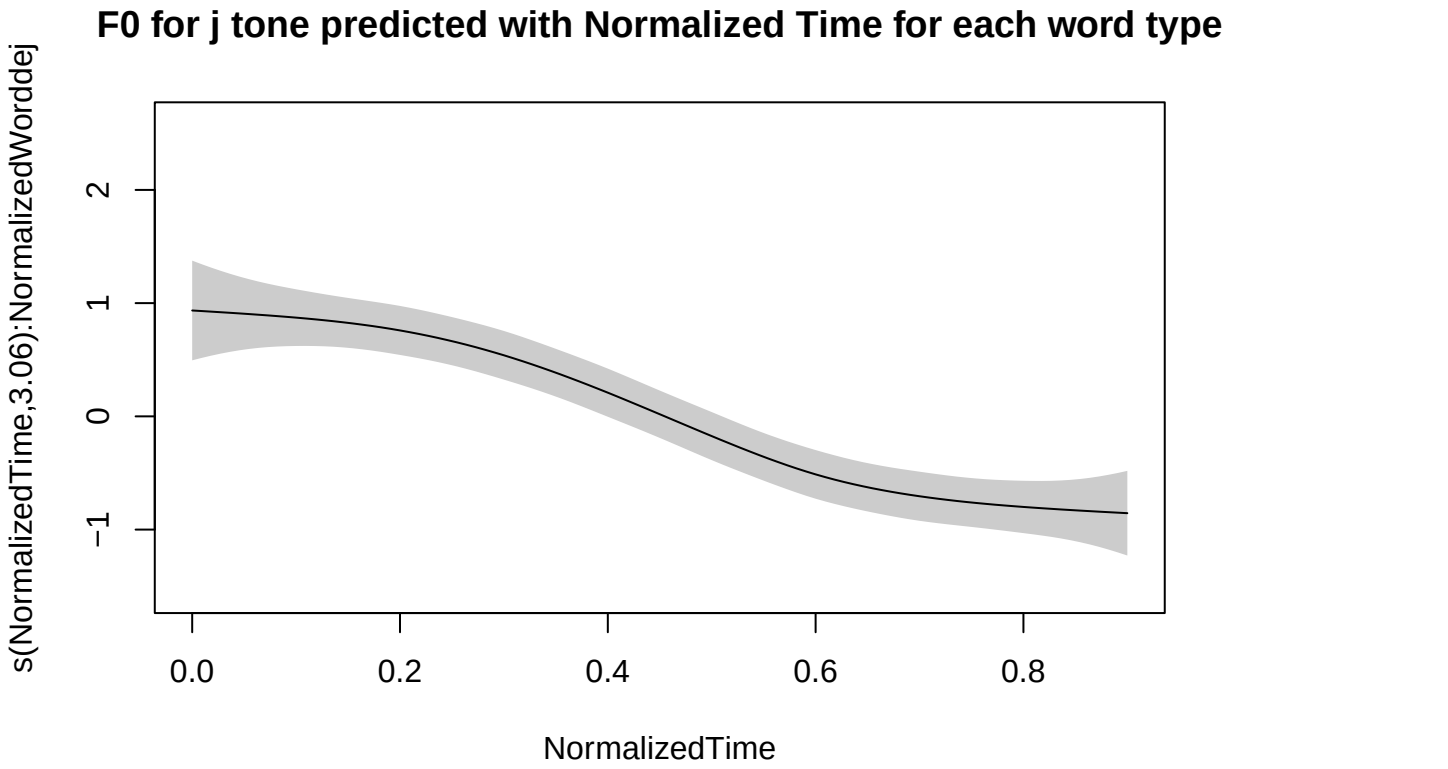


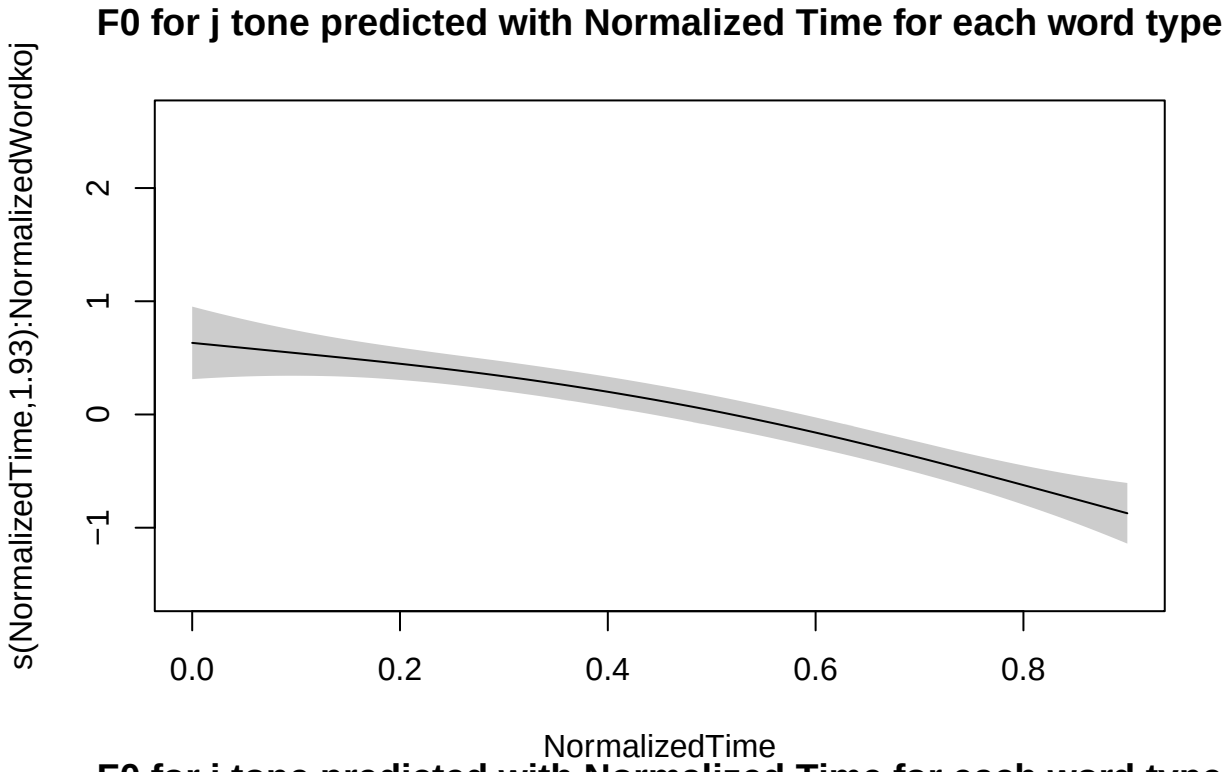

```
# Model
gamWord_j = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataJ, method = 'REML')
summary(gamWord_j)
```

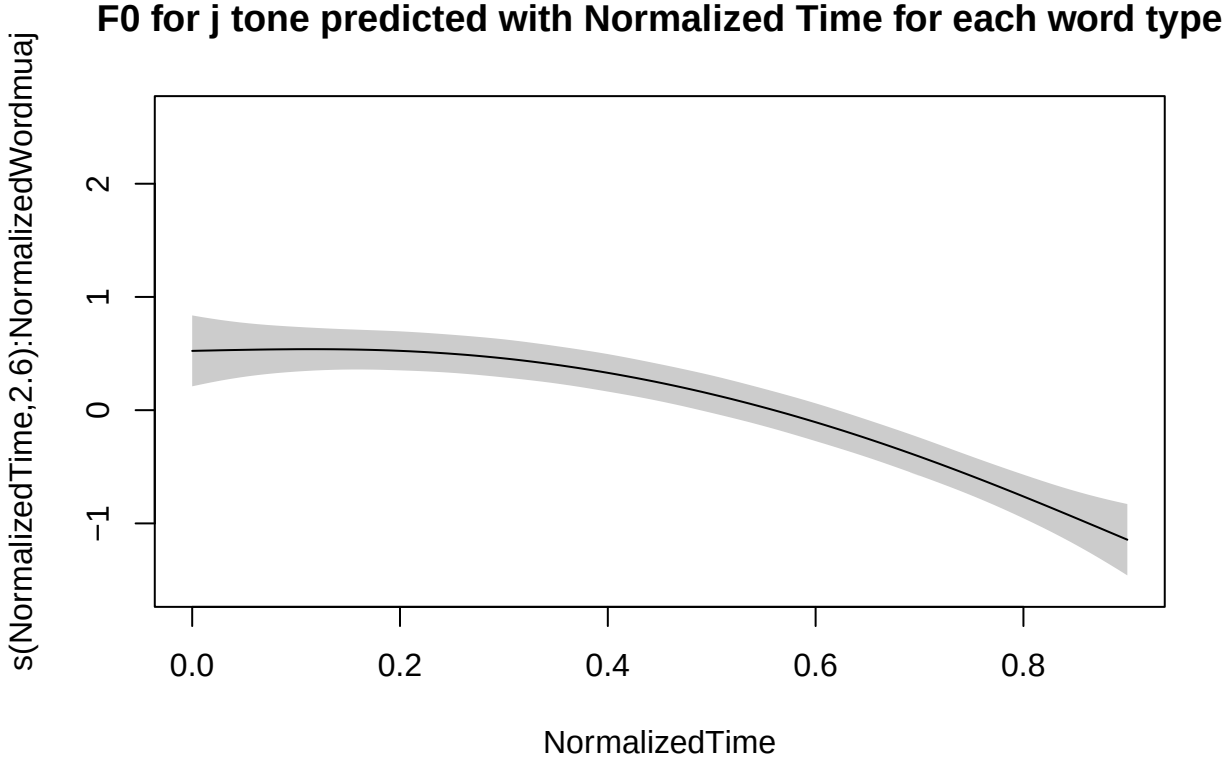
```
##
## Family: gaussian
```

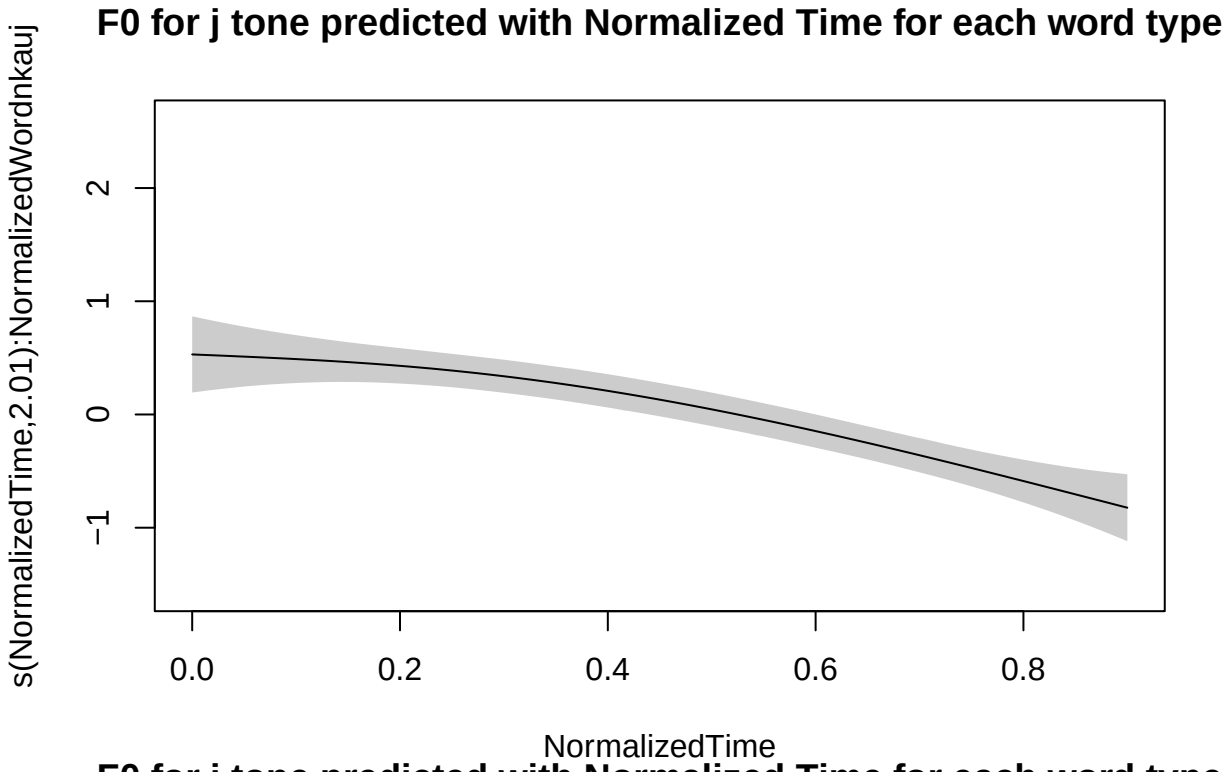
```
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.22515    0.01819   177.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                         edf Ref.df      F p-value
## s(NormalizedTime):NormalizedWorddej   3.057  3.815  22.24  <2e-16 ***
## s(NormalizedTime):NormalizedWordkoj   1.934  2.419  23.42  <2e-16 ***
## s(NormalizedTime):NormalizedWordmuaj  2.603  3.240  23.46  <2e-16 ***
## s(NormalizedTime):NormalizedWordnkauj 2.011  2.512  17.14  <2e-16 ***
## s(NormalizedTime):NormalizedWordpaj   4.180  5.185  78.46  <2e-16 ***
## s(NormalizedTime):NormalizedWordthiaj 1.018  1.036 307.80  <2e-16 ***
## s(NormalizedTime):NormalizedWordX     3.170  3.950 195.41  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.125   Deviance explained = 12.7%
## -REML =  25875  Scale est. = 4.0356    n = 12214
```

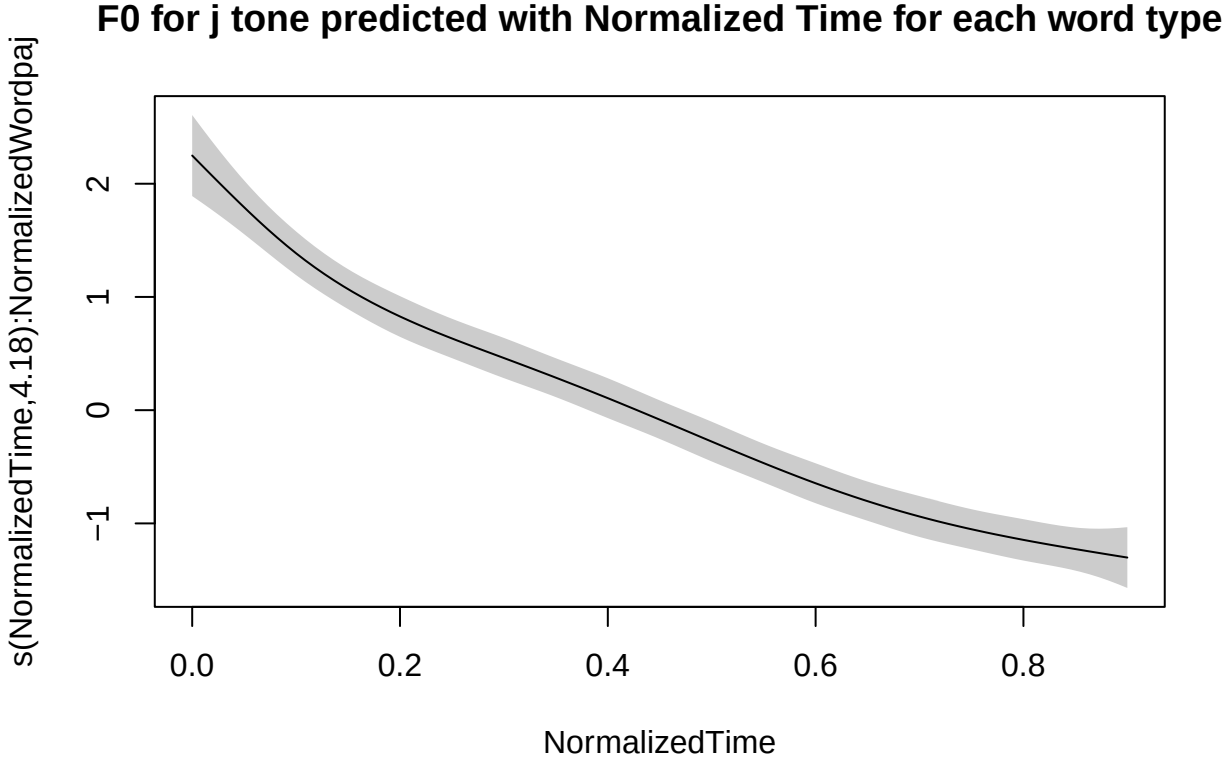
```r
# Visualize Model
plot(gamWord_j, shade = TRUE, main = 'F0 for j tone predicted with Normalized Time for each word type')
```

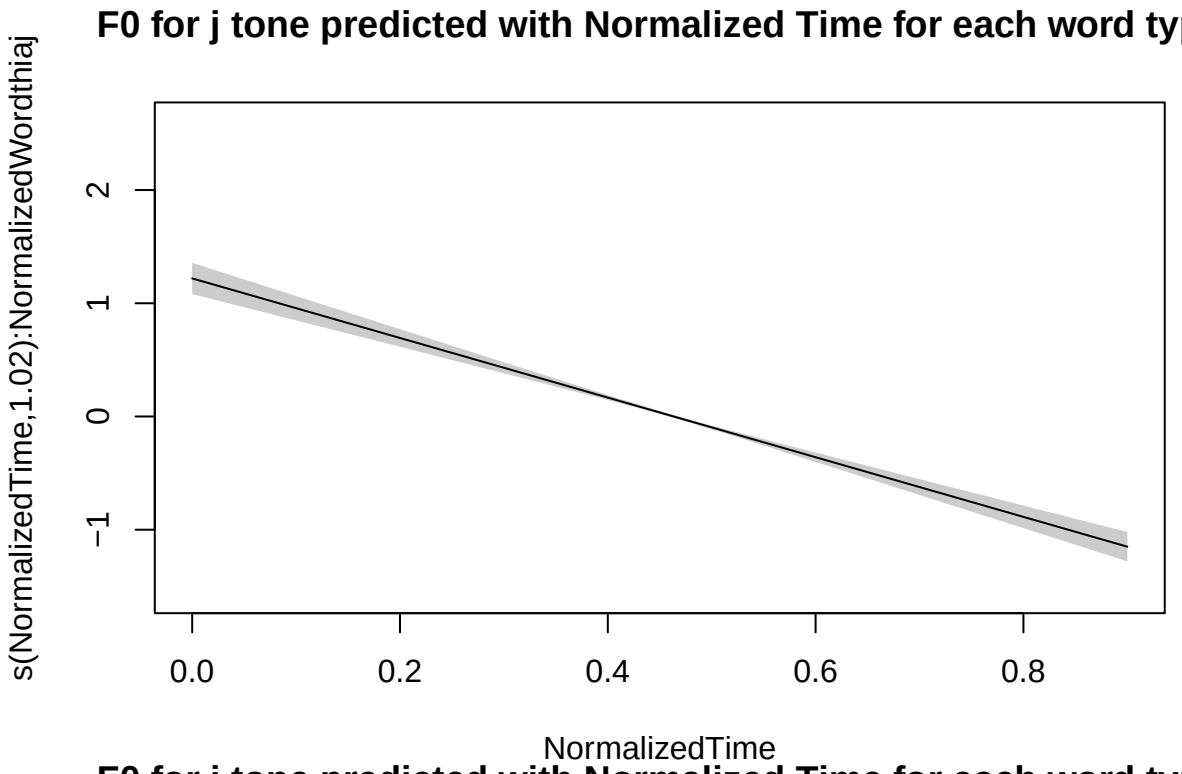


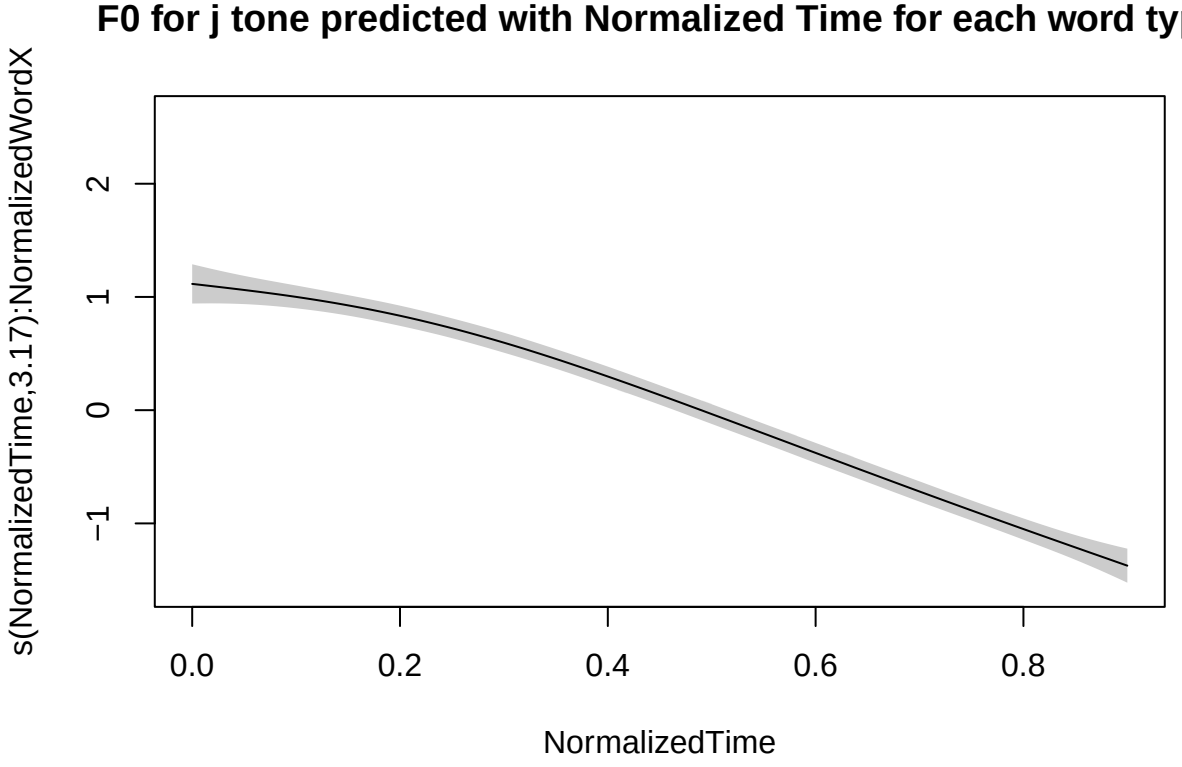

F0 for j tone predicted with Normalized Time for each word type
s(NormalizedTime,1.93):NormalizedWordkoj
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for j tone predicted with Normalized Time for each word type
s(NormalizedTime,2.6):NormalizedWordmuaj
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for j tone predicted with Normalized Time for each word type
s(NormalizedTime,2.01):NormalizedWordnkauj
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for j tone predicted with Normalized Time for each word type
s(NormalizedTime,4.18):NormalizedWordpaj
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
NormalizedTime



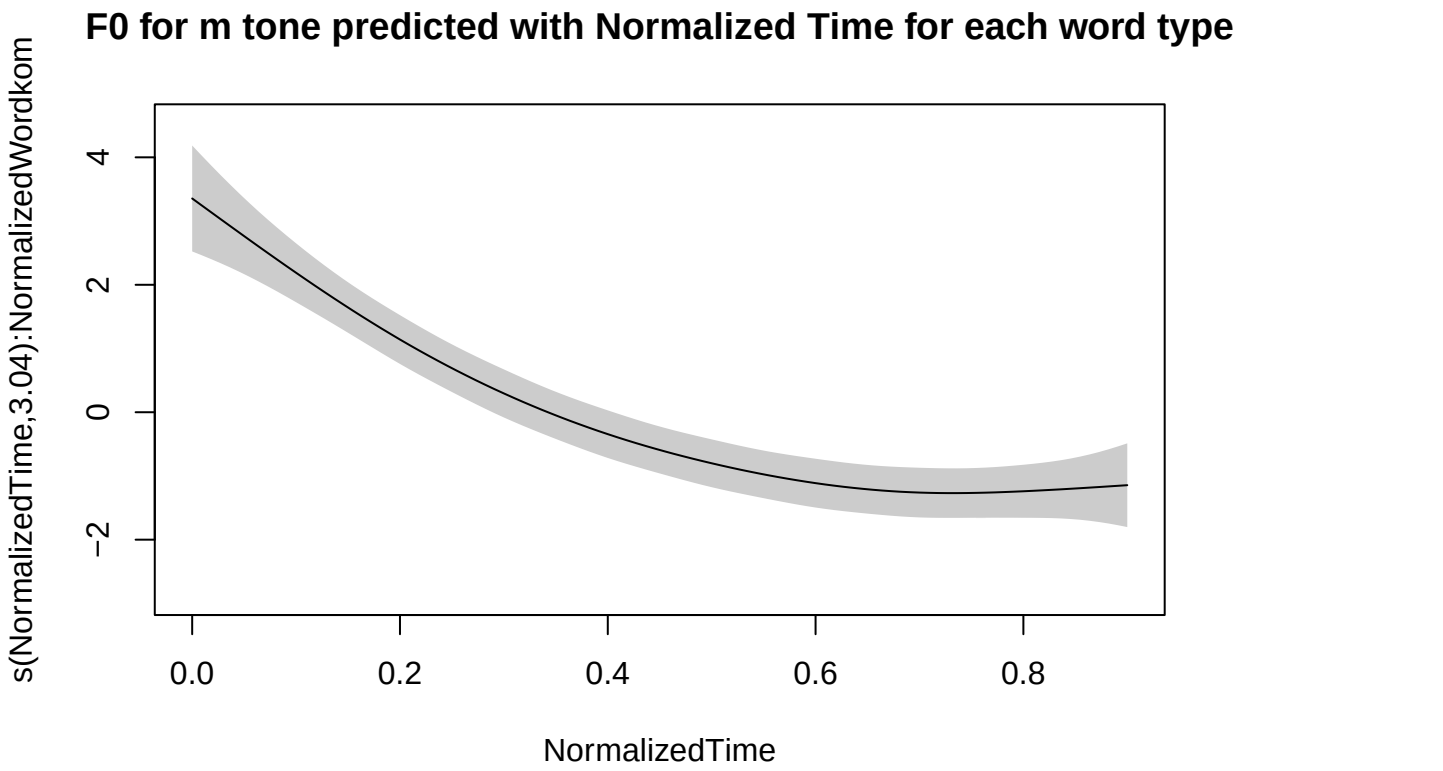


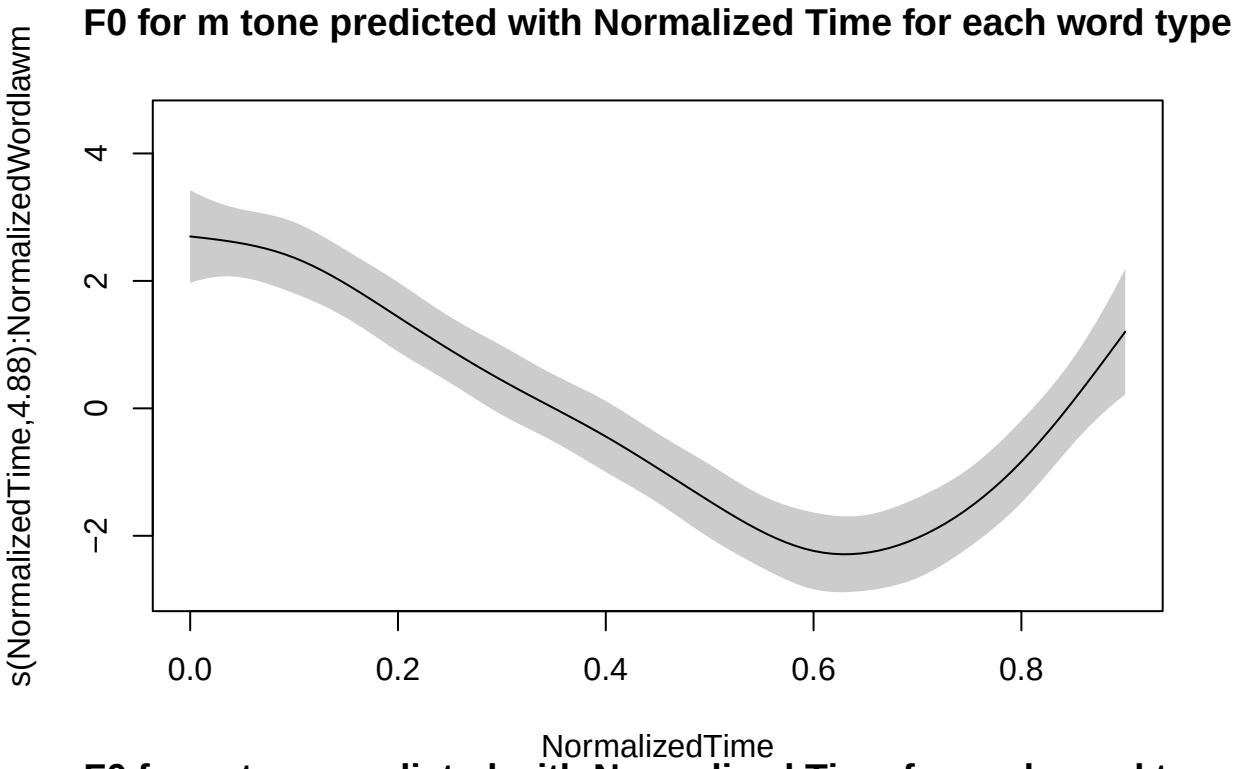

```
# Model
gamWord_m = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataM, method = 'REML')
summary(gamWord_m)
```

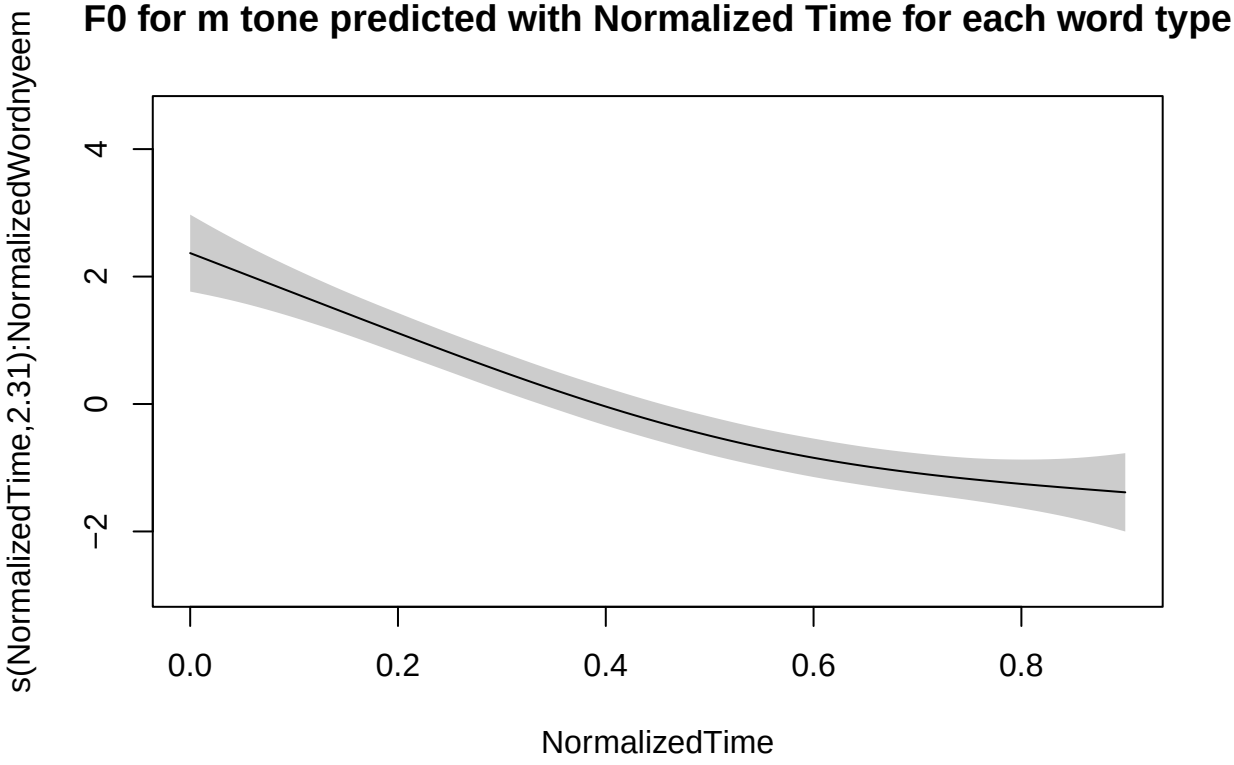
```
##
## Family: gaussian
```

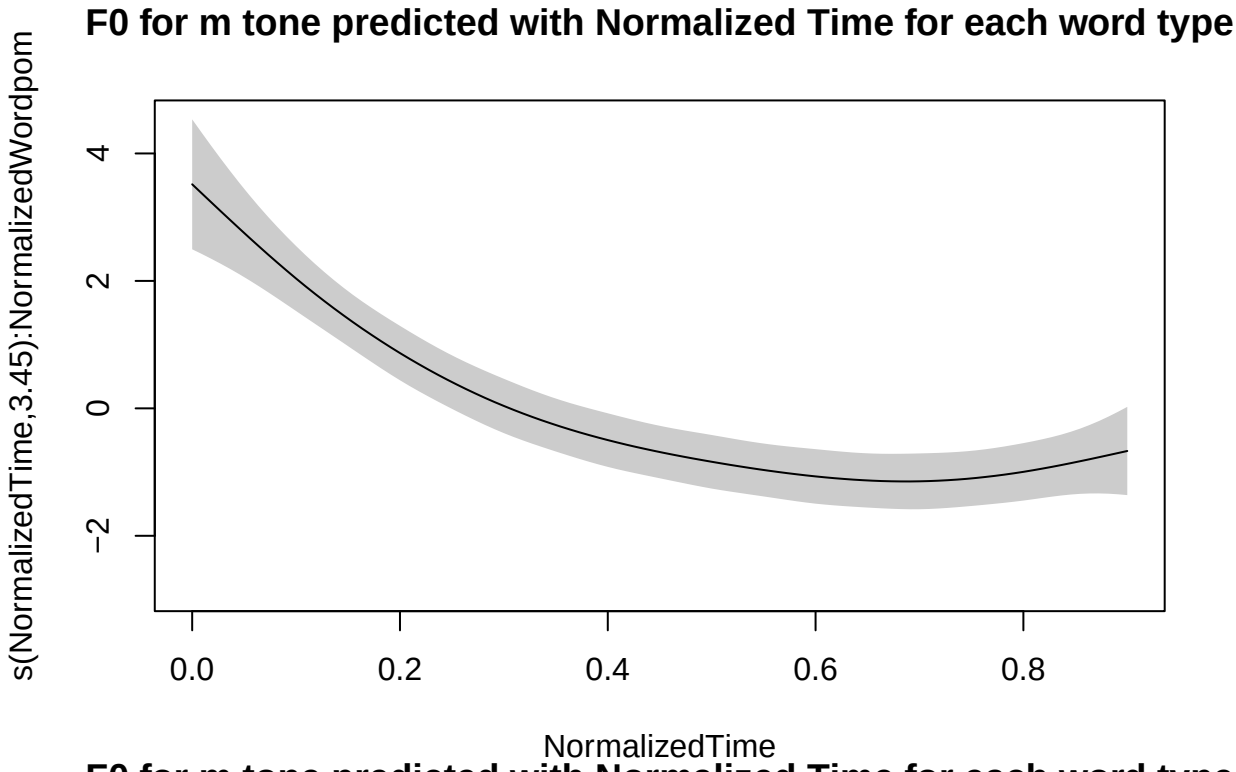
```
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.88625    0.03453  -83.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                         edf Ref.df      F  p-value
## s(NormalizedTime):NormalizedWordkom   3.036  3.789  28.57  < 2e-16 ***
## s(NormalizedTime):NormalizedWordlawm  4.885  5.984  25.76  < 2e-16 ***
## s(NormalizedTime):NormalizedWordnyeem 2.309  2.875  32.59  < 2e-16 ***
## s(NormalizedTime):NormalizedWordpom   3.452  4.309  18.36  < 2e-16 ***
## s(NormalizedTime):NormalizedWordthaum 2.051  2.565  33.07  < 2e-16 ***
## s(NormalizedTime):NormalizedWordtiam  1.780  2.225  14.36 2.58e-07 ***
## s(NormalizedTime):NormalizedWordX     4.361  5.391 215.80  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.122   Deviance explained = 12.3%
## -REML =  34077  Scale est. = 14.579    n = 12344
```

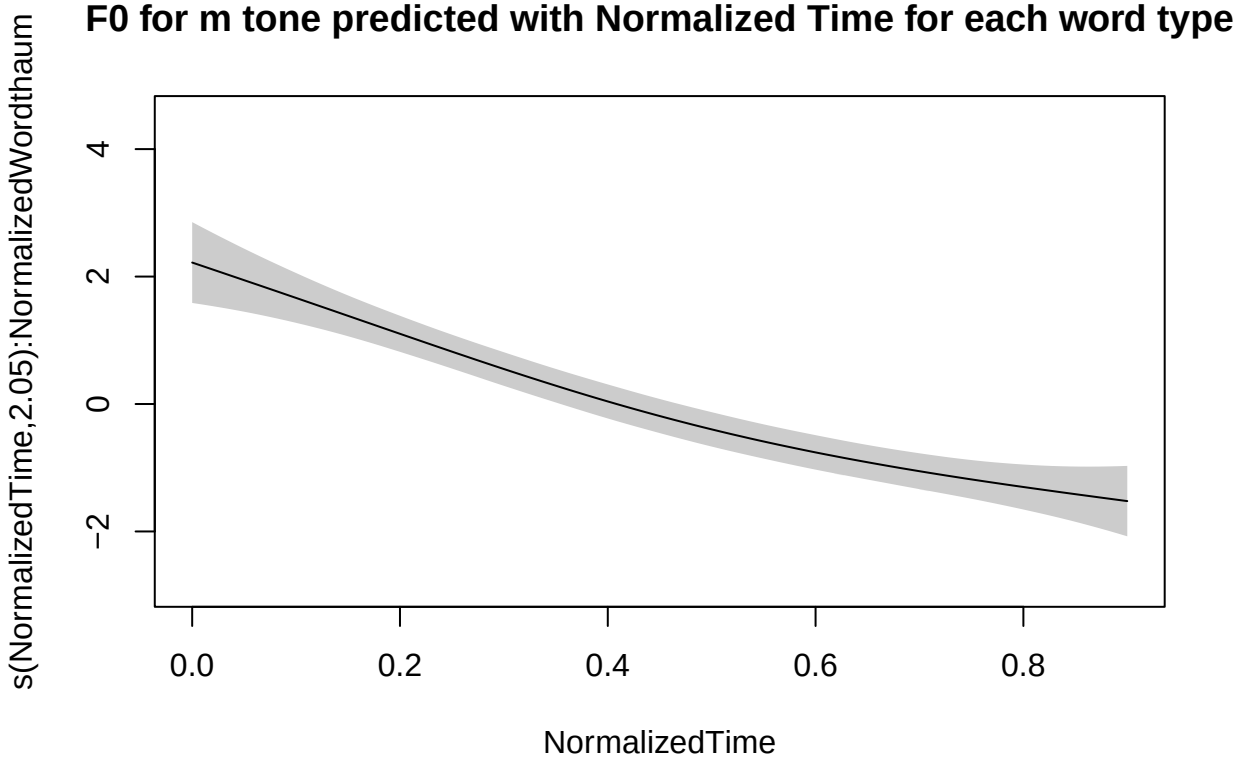
```r
# Visualize Model
plot(gamWord_m, shade = TRUE, main = 'F0 for m tone predicted with Normalized Time for each word type')
```

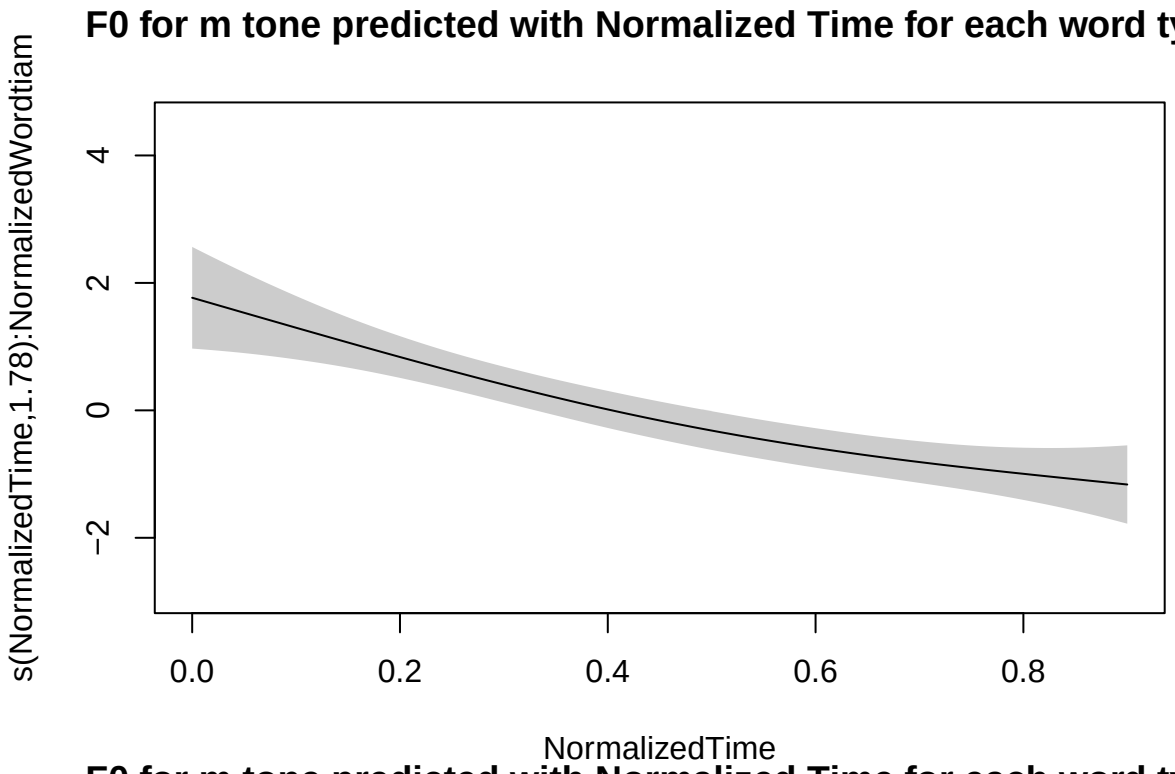


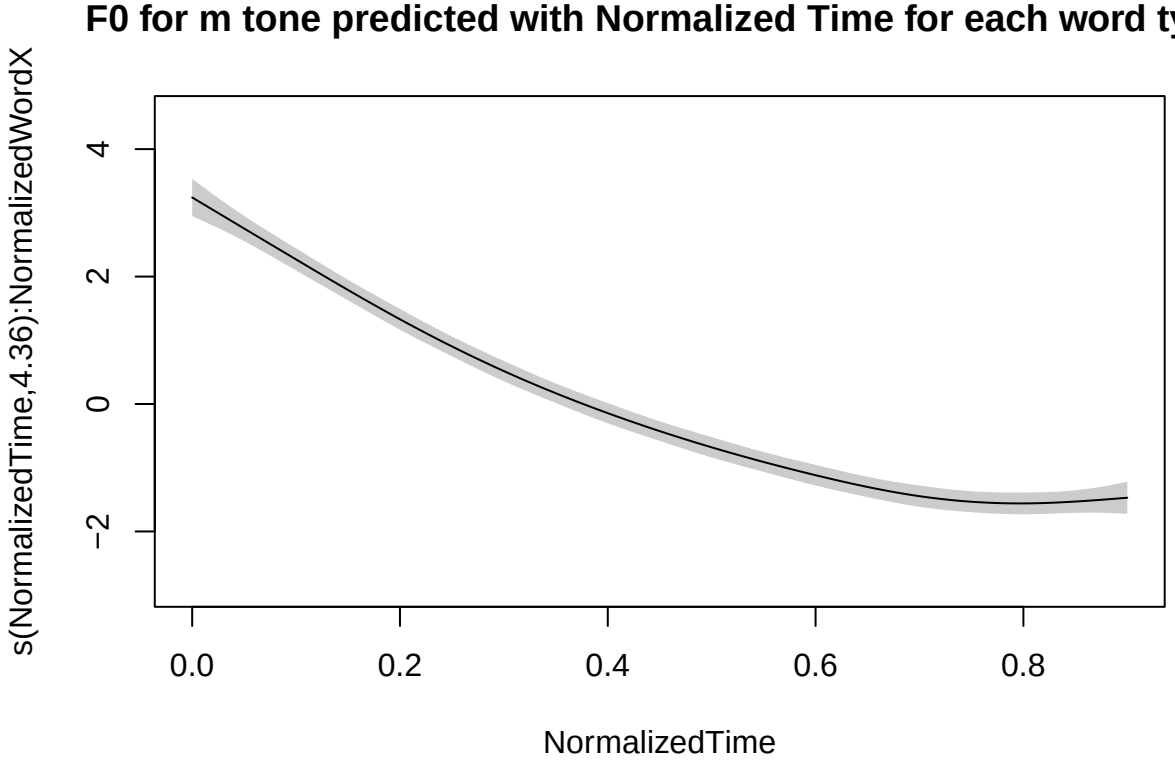

F0 for m tone predicted with Normalized Time for each word type
s(NormalizedTime,4.88):NormalizedWordlawm
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for m tone predicted with Normalized Time for each word type
s(NormalizedTime,2.31):NormalizedWordnyeem
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for m tone predicted with Normalized Time for each word type
s(NormalizedTime,3.45):NormalizedWordpom
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for m tone predicted with Normalized Time for each word type
s(NormalizedTime,2.05):NormalizedWordthaum
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
NormalizedTime



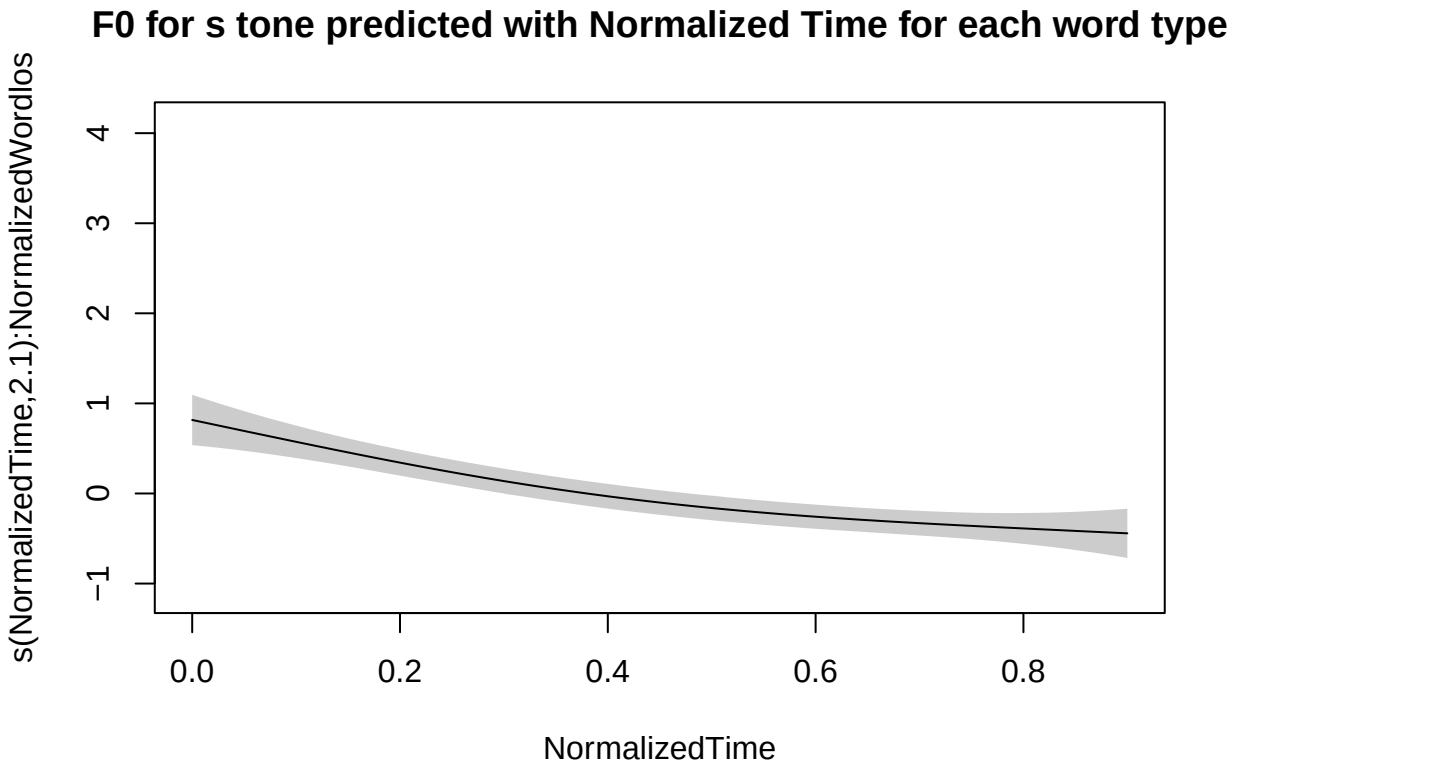


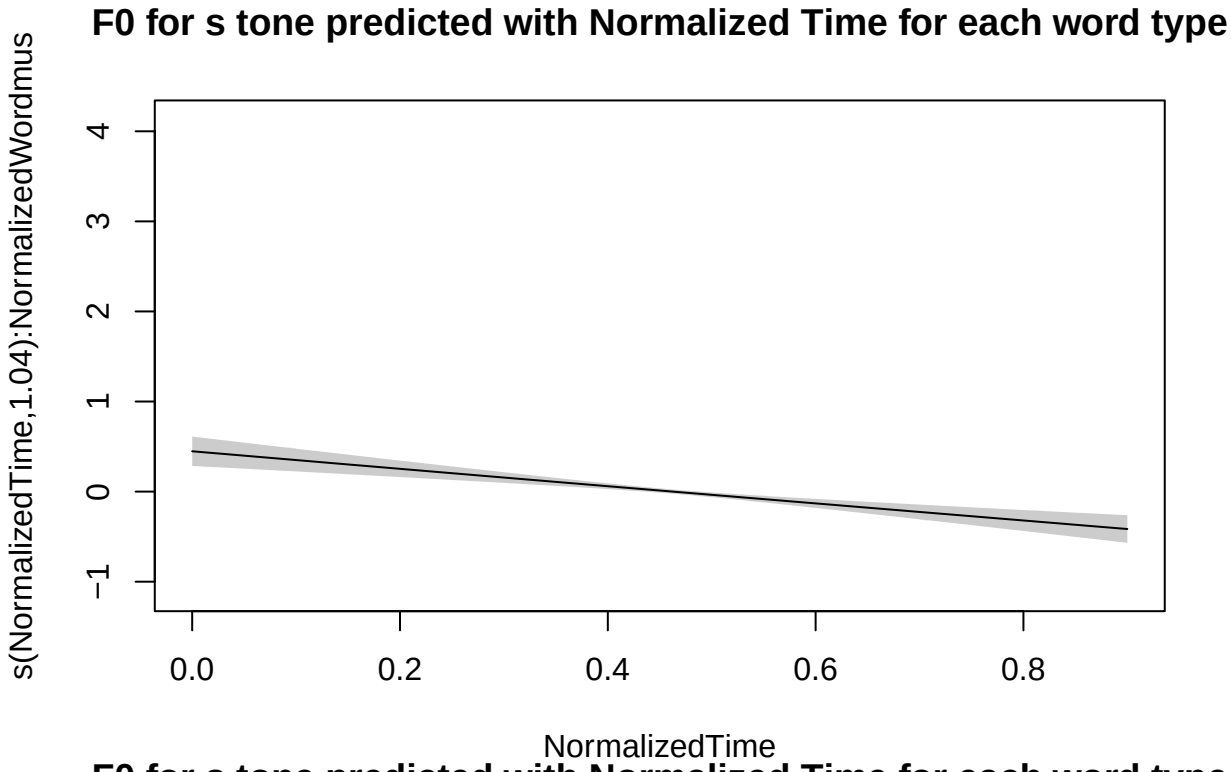

```
# Model
gamWord_s = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataS, method = 'REML')
summary(gamWord_s)
```

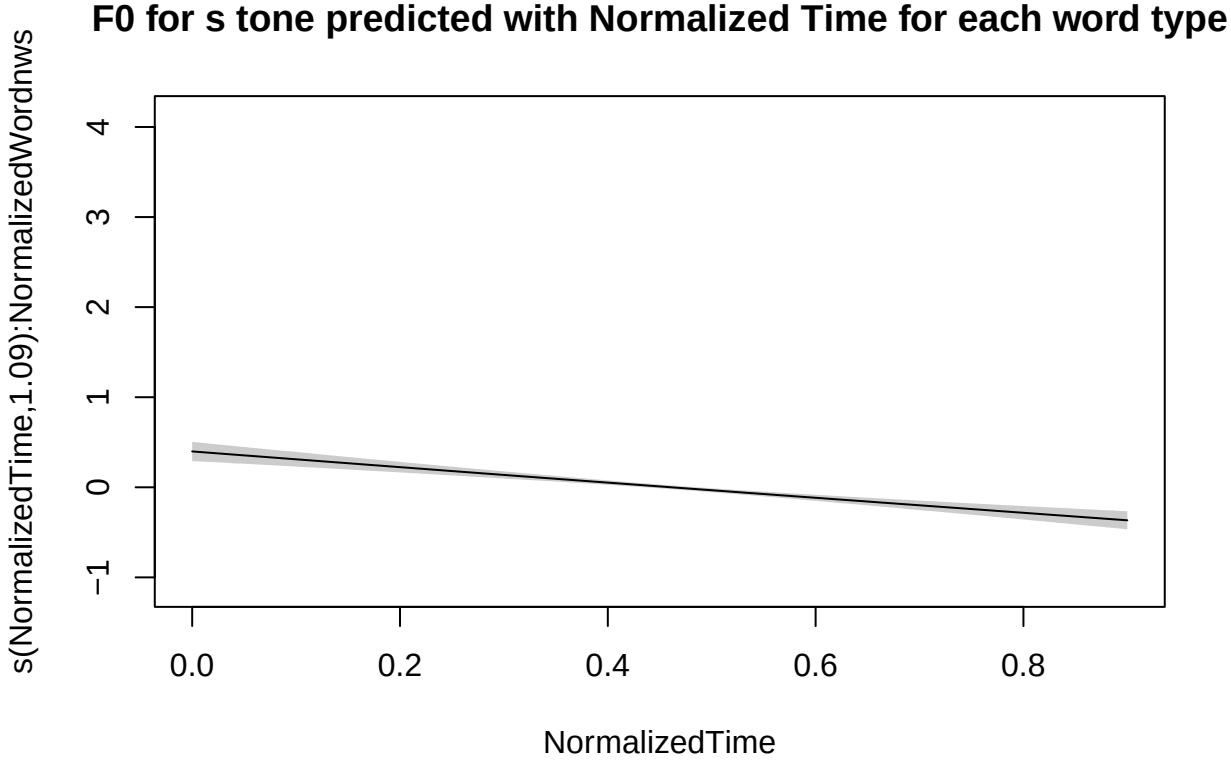
```
##
## Family: gaussian
```

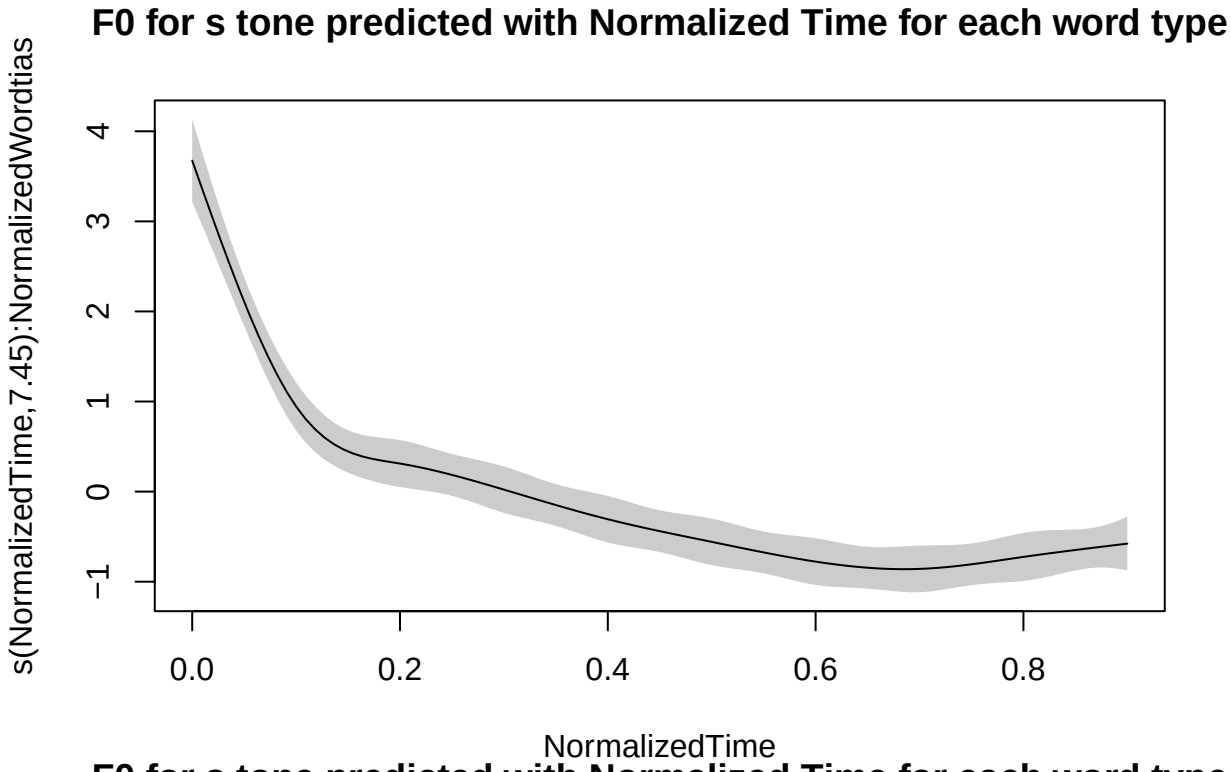
```
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.19066    0.01577  -75.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                        edf Ref.df      F p-value
## s(NormalizedTime):NormalizedWordlos  2.103  2.620  17.04  <2e-16 ***
## s(NormalizedTime):NormalizedWordmus  1.045  1.088  28.21  <2e-16 ***
## s(NormalizedTime):NormalizedWordnws  1.090  1.173  50.36  <2e-16 ***
## s(NormalizedTime):NormalizedWordtias 7.448  8.446  43.26  <2e-16 ***
## s(NormalizedTime):NormalizedWordtsis 3.235  4.038  39.54  <2e-16 ***
## s(NormalizedTime):NormalizedWordtus  2.885  3.603  34.49  <2e-16 ***
## s(NormalizedTime):NormalizedWordX    4.648  5.724 103.68  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0584   Deviance explained = 5.94%
## -REML =  50117  Scale est. = 5.4508    n = 22094
```

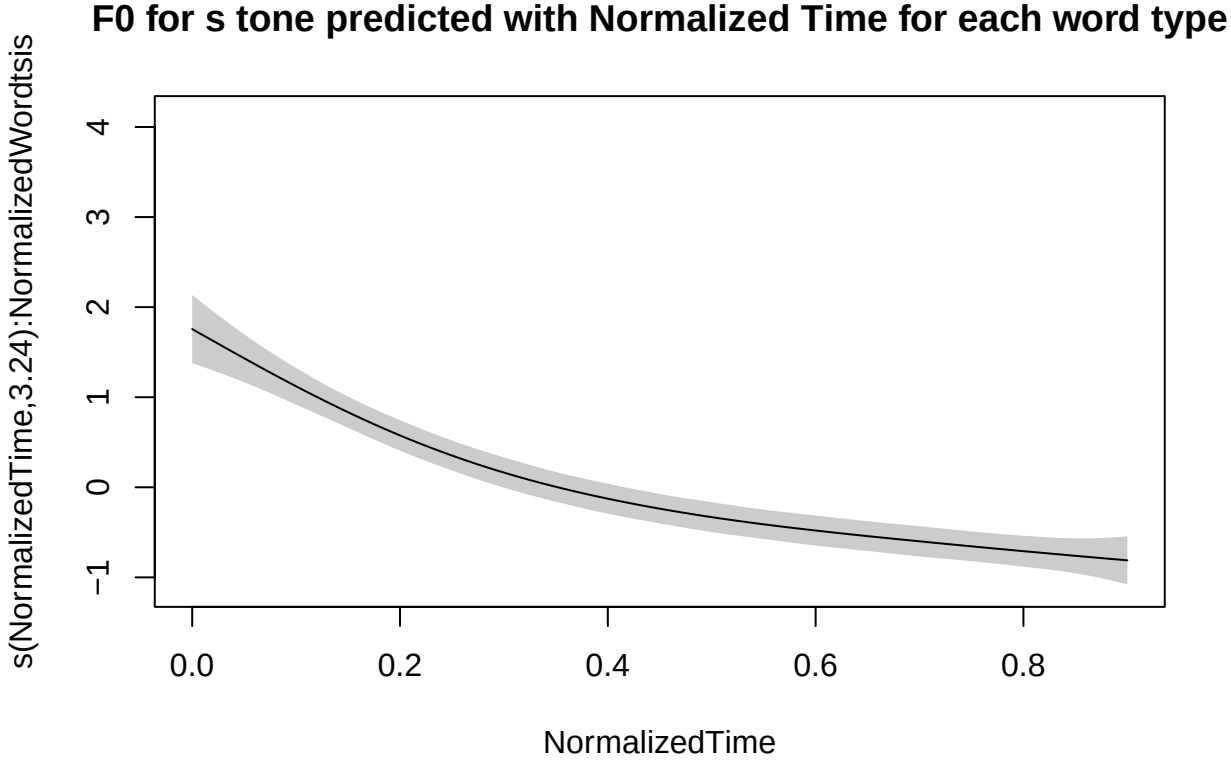
```r
# Visualize Model
plot(gamWord_s, shade = TRUE, main = 'F0 for s tone predicted with Normalized Time for each word type')
```

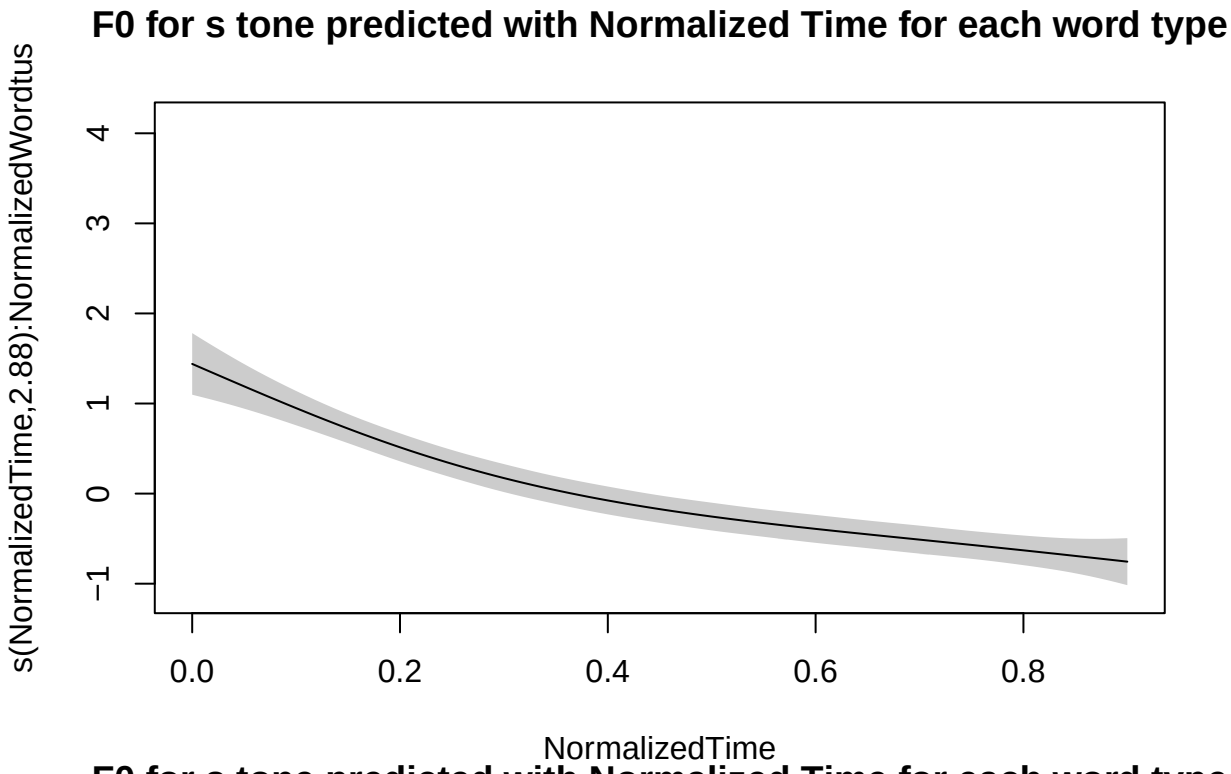


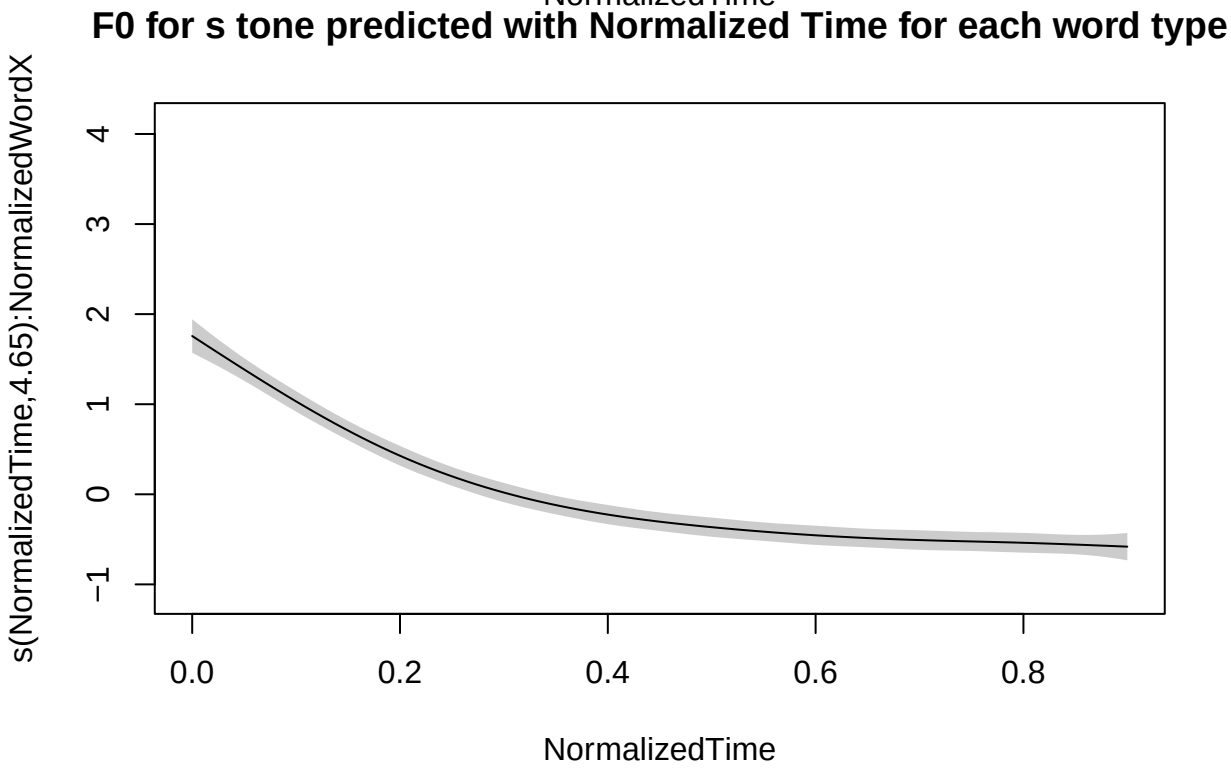

F0 for s tone predicted with Normalized Time for each word type
s(NormalizedTime,1.04):NormalizedWordmus
4
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for s tone predicted with Normalized Time for each word type
s(NormalizedTime,1.09):NormalizedWordnws
4
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for s tone predicted with Normalized Time for each word type
s(NormalizedTime,7.45):NormalizedWordtias
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for s tone predicted with Normalized Time for each word type
s(NormalizedTime,3.24):NormalizedWordtsis
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime



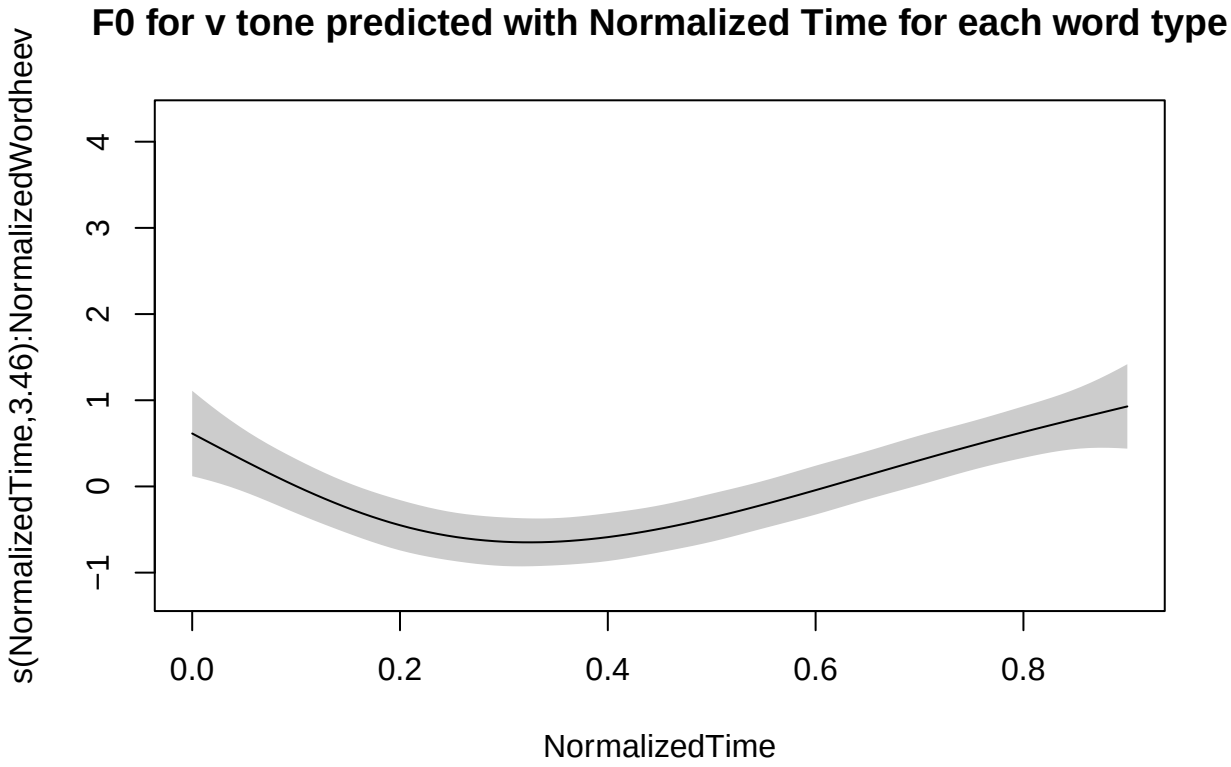


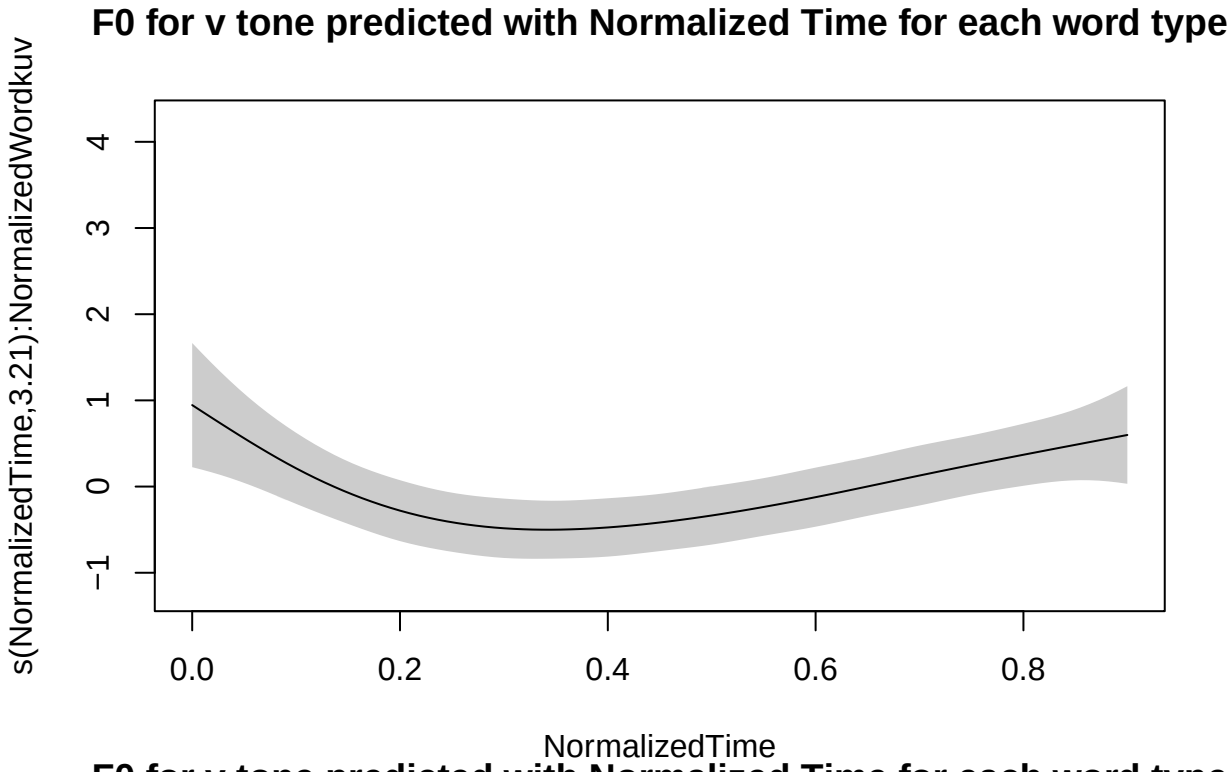

```
# Model
gamWord_v = gam(F0 ~ s(NormalizedTime, by = NormalizedWord), data = hmongDataV, method = 'REML')
summary(gamWord_v)
```

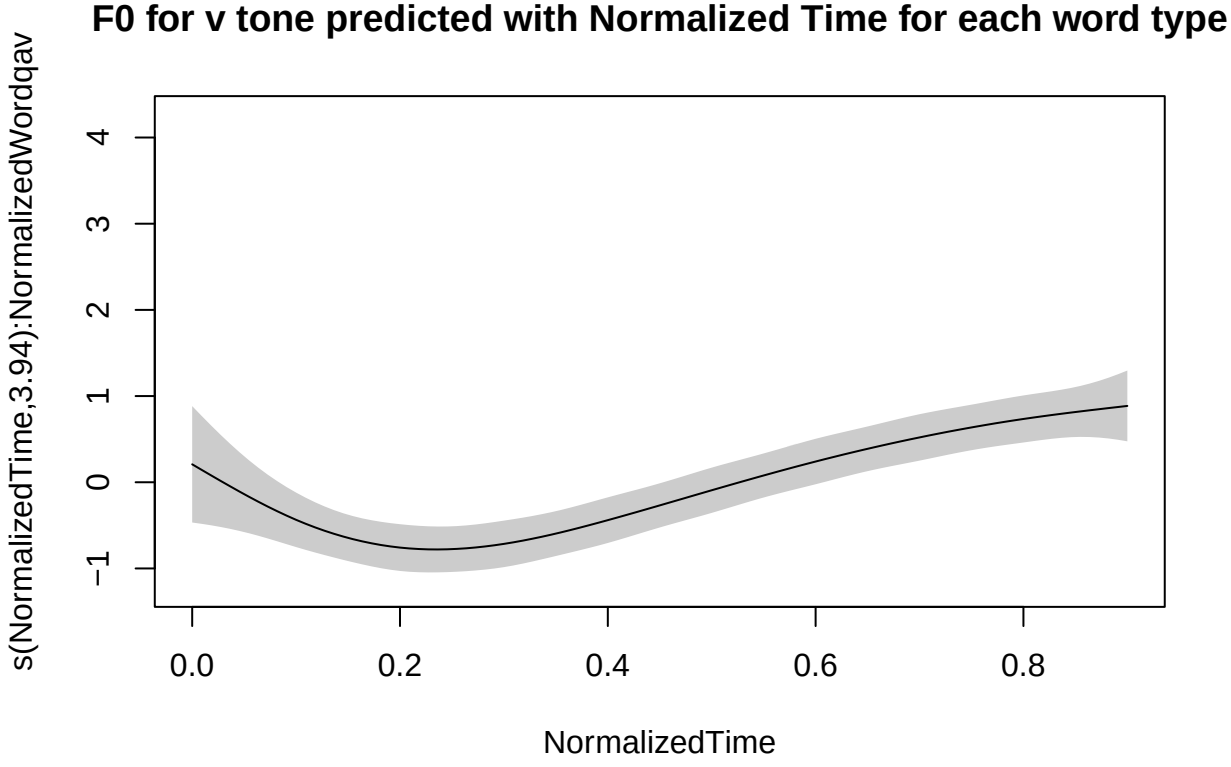
```
##
## Family: gaussian
```

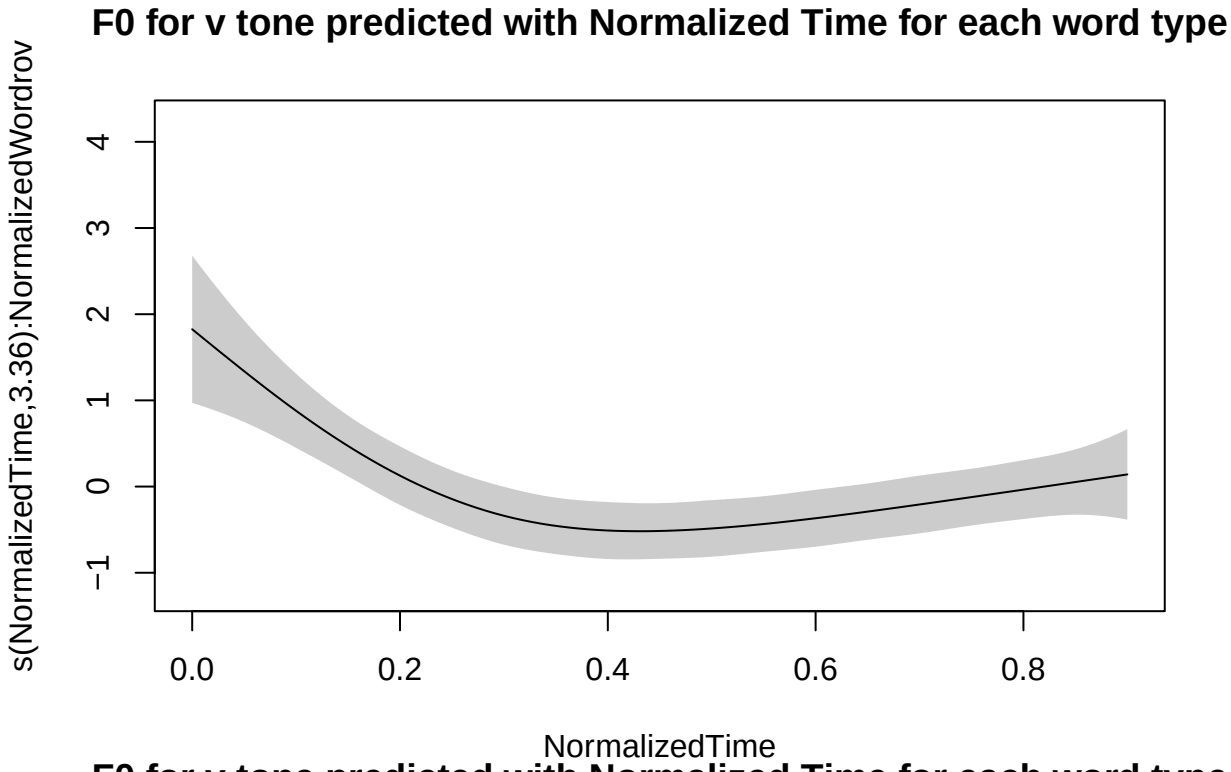
```
## Link function: identity
##
## Formula:
## F0 ~ s(NormalizedTime, by = NormalizedWord)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.27572    0.02613  -87.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                         edf Ref.df      F  p-value
## s(NormalizedTime):NormalizedWordheev 3.460  4.298  8.336  < 2e-16 ***
## s(NormalizedTime):NormalizedWordkuv  3.213  4.003  3.904 0.003622 **
## s(NormalizedTime):NormalizedWordqav  3.944  4.915 12.271  < 2e-16 ***
## s(NormalizedTime):NormalizedWordrov  3.357  4.192  5.646 0.000133 ***
## s(NormalizedTime):NormalizedWordtsov 5.991  7.225  9.420  < 2e-16 ***
## s(NormalizedTime):NormalizedWordX    5.389  6.568 26.082  < 2e-16 ***
## s(NormalizedTime):NormalizedWordyuav 2.702  3.362  4.647 0.002241 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0229   Deviance explained = 2.45%
## -REML =  43010  Scale est. = 11.085    n = 16392
```

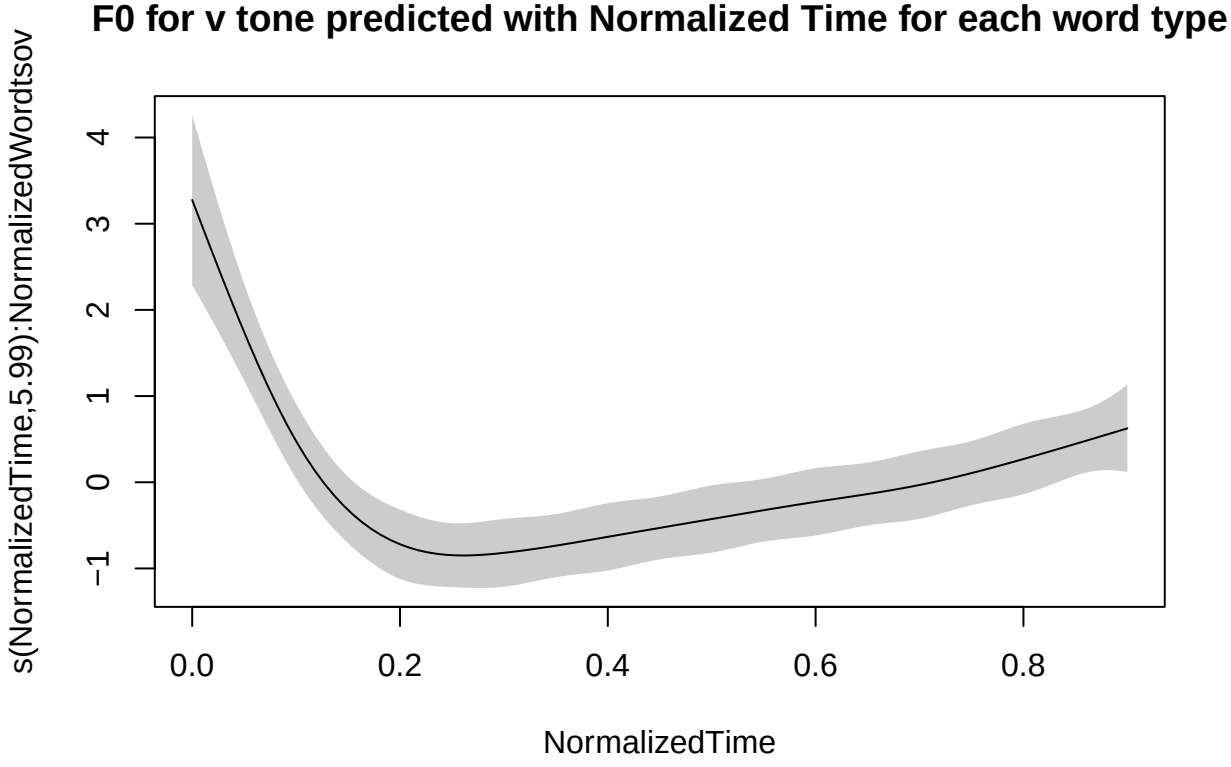
```r
# Visualize Model
plot(gamWord_v, shade = TRUE, main = 'F0 for v tone predicted with Normalized Time for each word type')
```

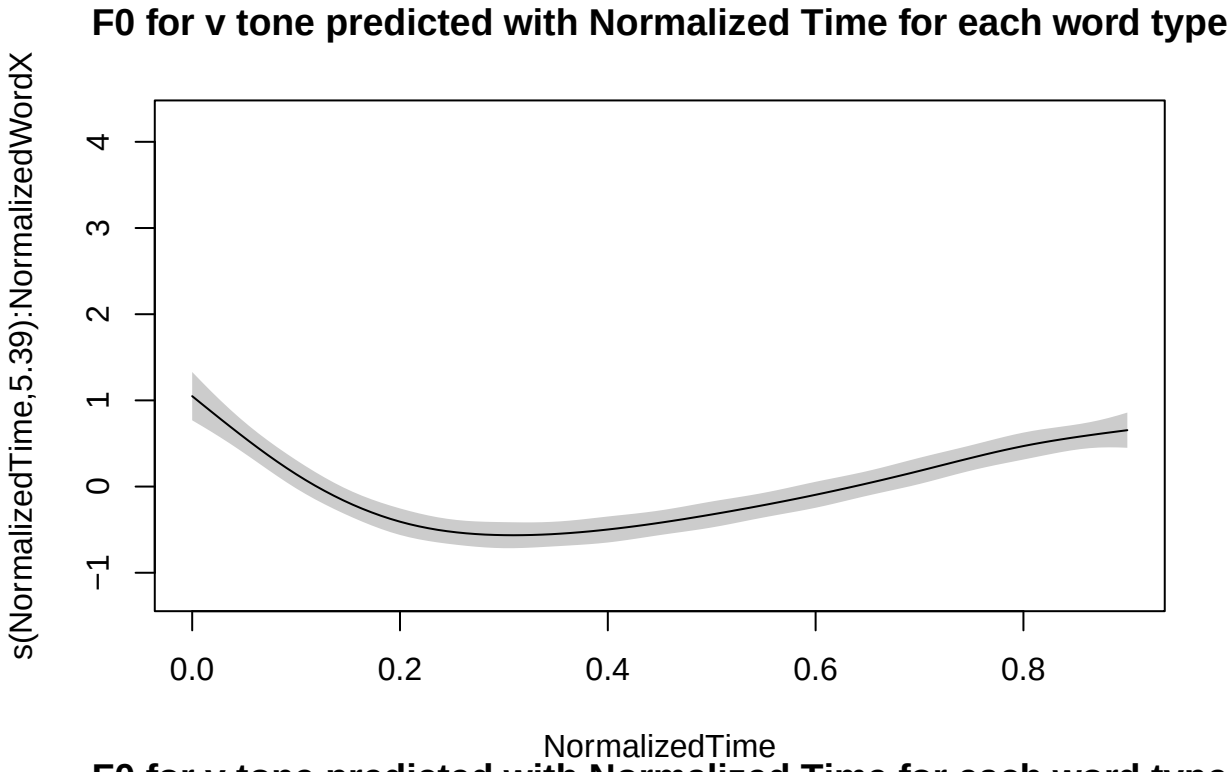


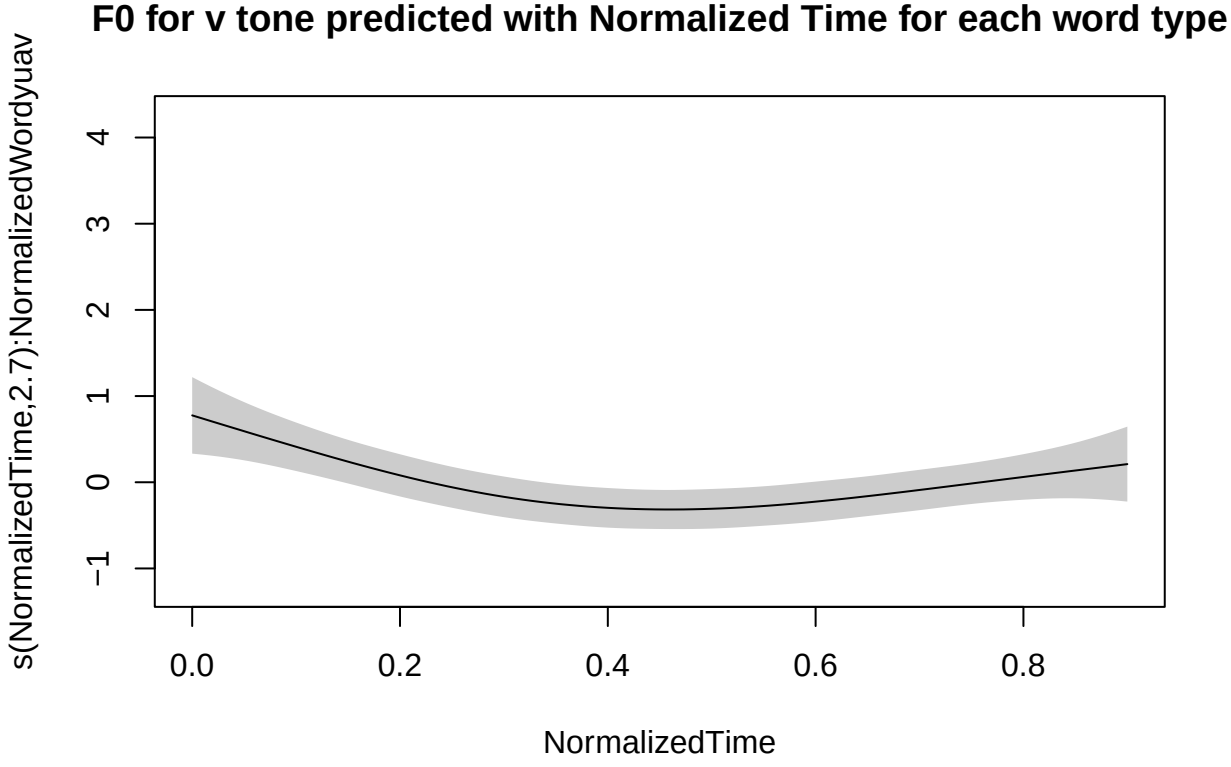

F0 for v tone predicted with Normalized Time for each word type
s(NormalizedTime,3.21):NormalizedWordkuv
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for v tone predicted with Normalized Time for each word type
s(NormalizedTime,3.94):NormalizedWordqav
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for v tone predicted with Normalized Time for each word type
s(NormalizedTime,3.36):NormalizedWordrov
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for v tone predicted with Normalized Time for each word type
s(NormalizedTime,5.99):NormalizedWordtsov
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for v tone predicted with Normalized Time for each word type
s(NormalizedTime,5.39):NormalizedWordX
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime


F0 for v tone predicted with Normalized Time for each word type
s(NormalizedTime,2.7):NormalizedWordyuav
−1
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
NormalizedTime