Mixed Linear Models

Case studies on speech rate modulations in spontaneous speech

LSA Summer Institute 2009, UC Berkeley

Florian Jaeger

University of Rochester

Managing speech rate

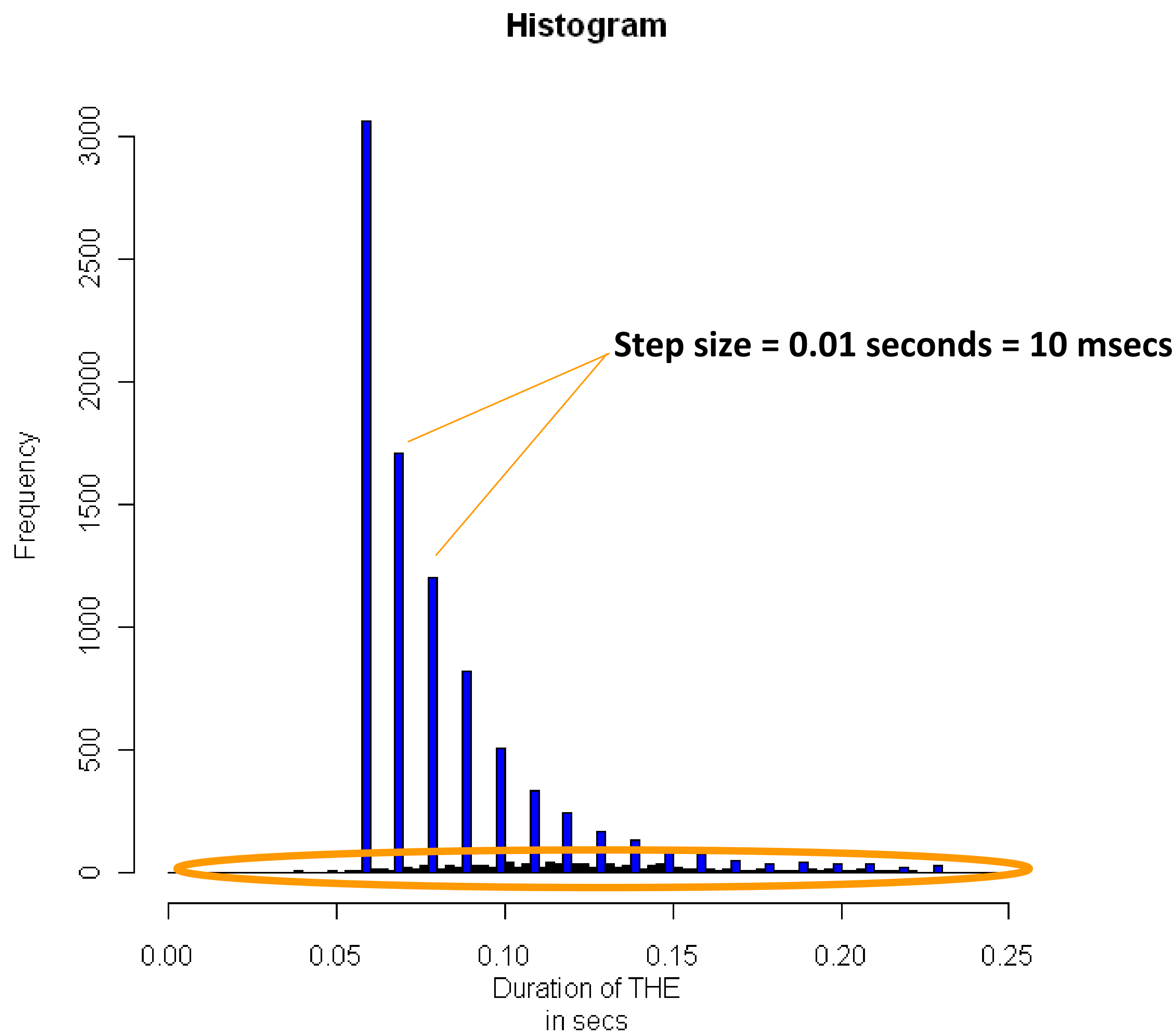
- How do speakers determine how fast to talk at a given moment?

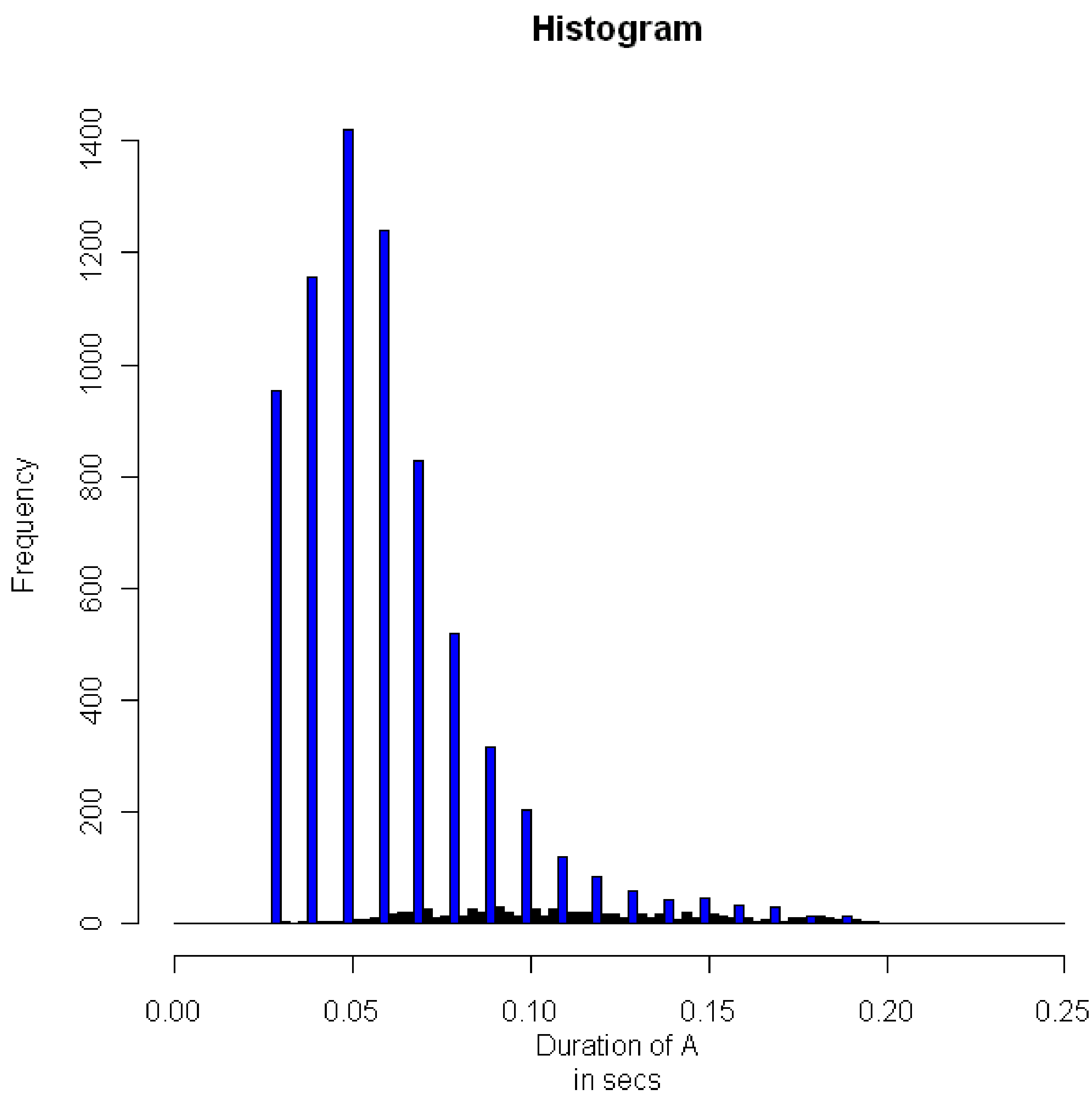
- Beyond speech rate difference between speakers, speech rate could be used strategically
  - … to slow down when **planning/retrieving difficult *upcoming* material** in order to avoid disfluency
  - … to slow down if the ***current* word is unexpected** to provide more signal to the interlocutors
- Speech rate may also be affected by **segmental** or **supra-segmental interference**.

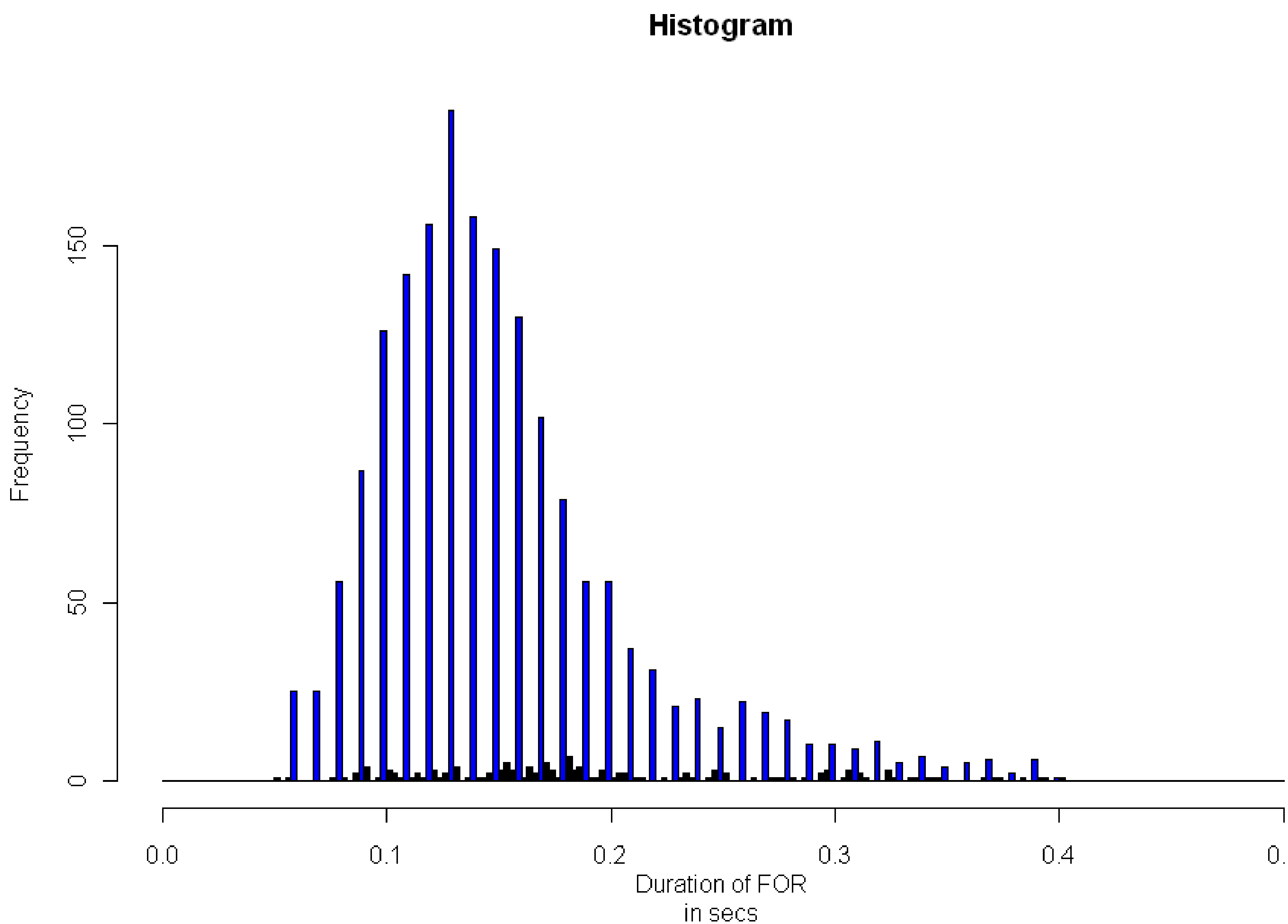
Corpus & Data

- Switchboard corpus
  - 357 speakers
  - 650 dialogues
  - 800k words
  - 100k utterances
  - *Automatically time-aligned transcription* (40k words hand-corrected)

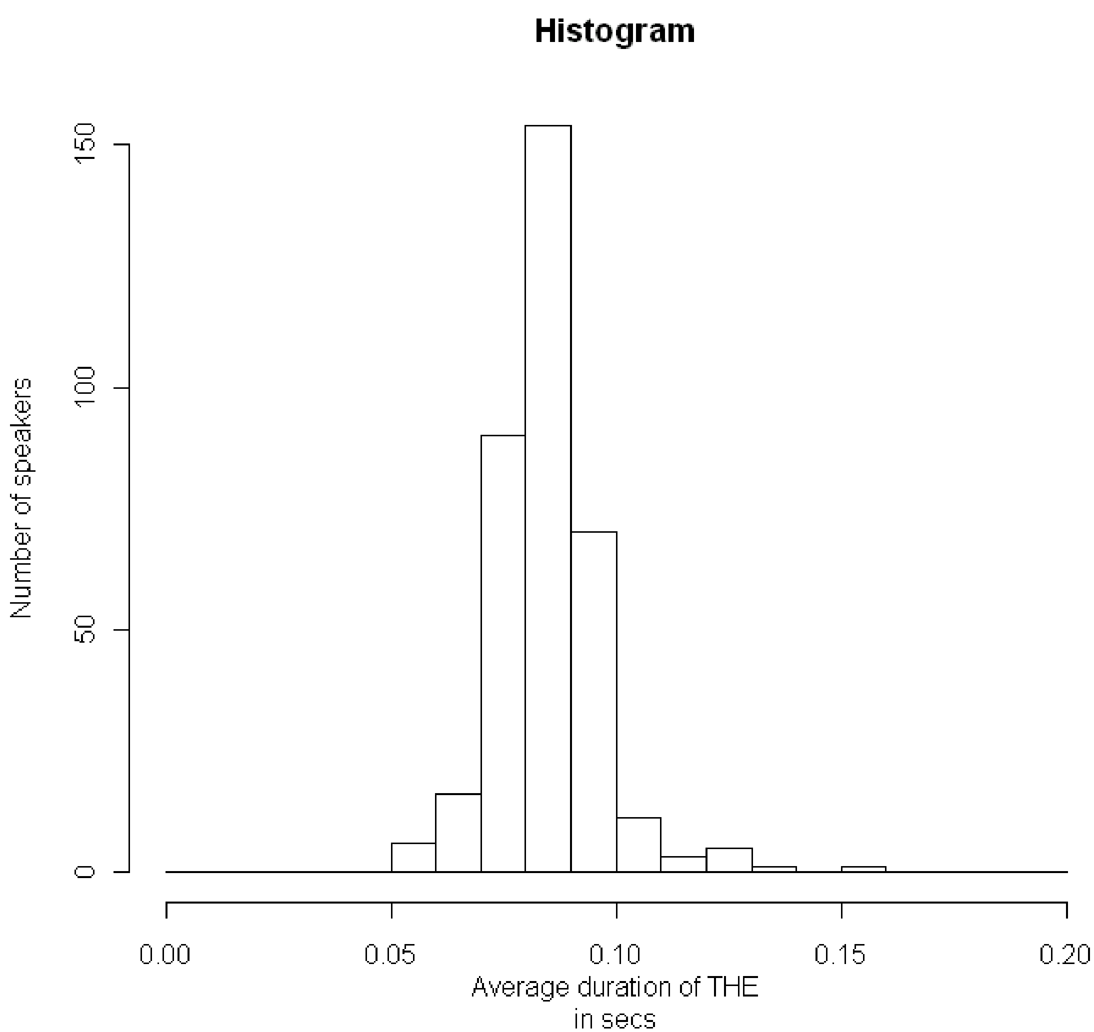
- **Today:**
  - High frequency function word: *the, a, they, it,* etc.


Histogram
Step size = 0.01 seconds = 10 msecs
Frequency
3000
2500
2000
1500
1000
500
0
0.00
0.05
0.10
0.15
0.20
0.25
Duration of THE
in secs

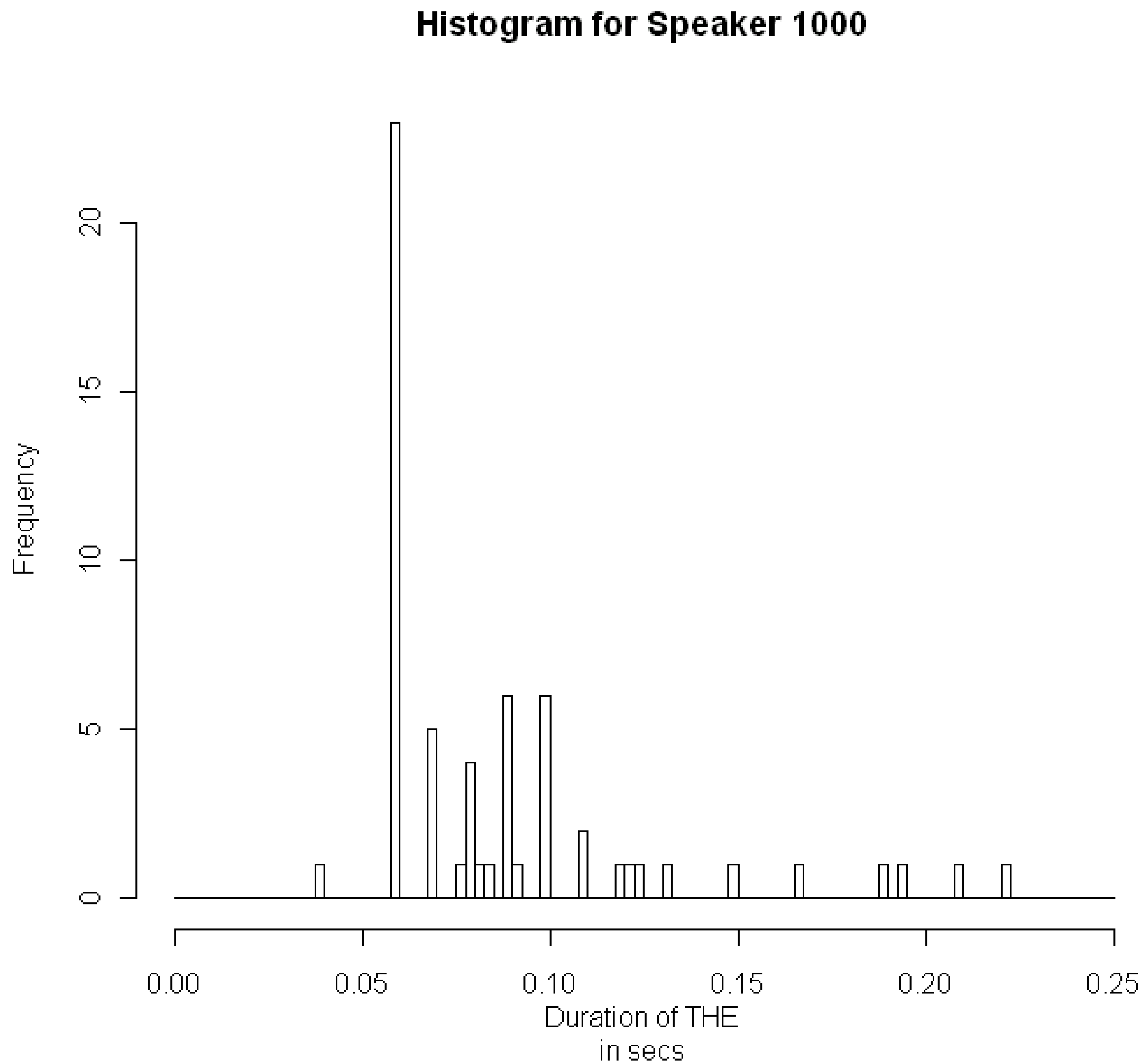

Histogram
Frequency
0
200
400
600
800
1000
1200
1400
0.00
0.05
0.10
0.15
0.20
0.25
Duration of A
in secs


Histogram
Frequency
0
50
100
150
0.0
0.1
0.2
0.3
0.4
0.5
Duration of FOR
in secs

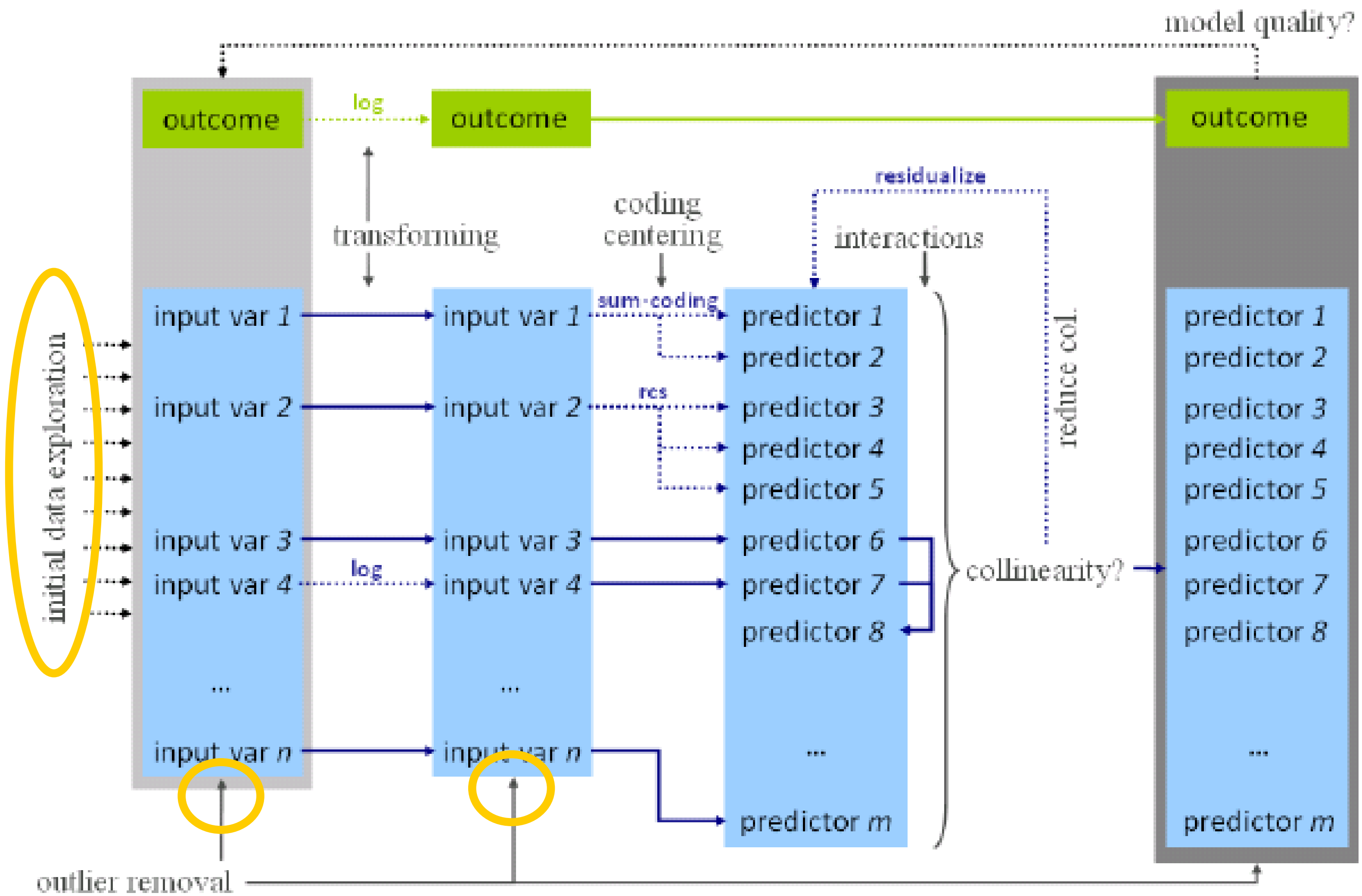
Speakers vary



Instances within speakers vary



Preparing the data



Subset-ing (1): Missing information

- Exclude cases with missing variable information:

```
d <- subset(d,
   SpeechRate > 0 &
   !is.na(ID_duration) &
   ID_duration > 0 &
   WORDpreceding != "" &
   WORDfollowing != ""
)
```

Subset-ing (2): Stratification

- Only words in the center of prosodic phrases of sufficiently long clauses:

```
d <- subset(d,
   TOPlength > 4 &
   ID_spWindowSyllables > 7 &
   ID_spWindowSyllables < 40 &
   ID_spWindowSyllablePosition > 3 &
   ID_spWindowSyllables - ID_spWindowSyllablePosition > 3
)
```

- Exclude disfluent words:

```
d <- subset(d,
   Dform != 1
)
```

Subset-ing (3):

- Exclude outliers based on distributional information:

```
d<- subset(d,
   abs(scale(lSpeechRate)) < 2.5 &
   abs(scale(ID_duration)) < 2.5
)
```

Data

- 9,460 *the*
- 7,685 *a*
- 5, 876 *I*
- 5,443 *that* (determiner)
- 3,605 *it*
- 2,290 *they*
- 1,930 *for*
- 1,730 *we*
- …

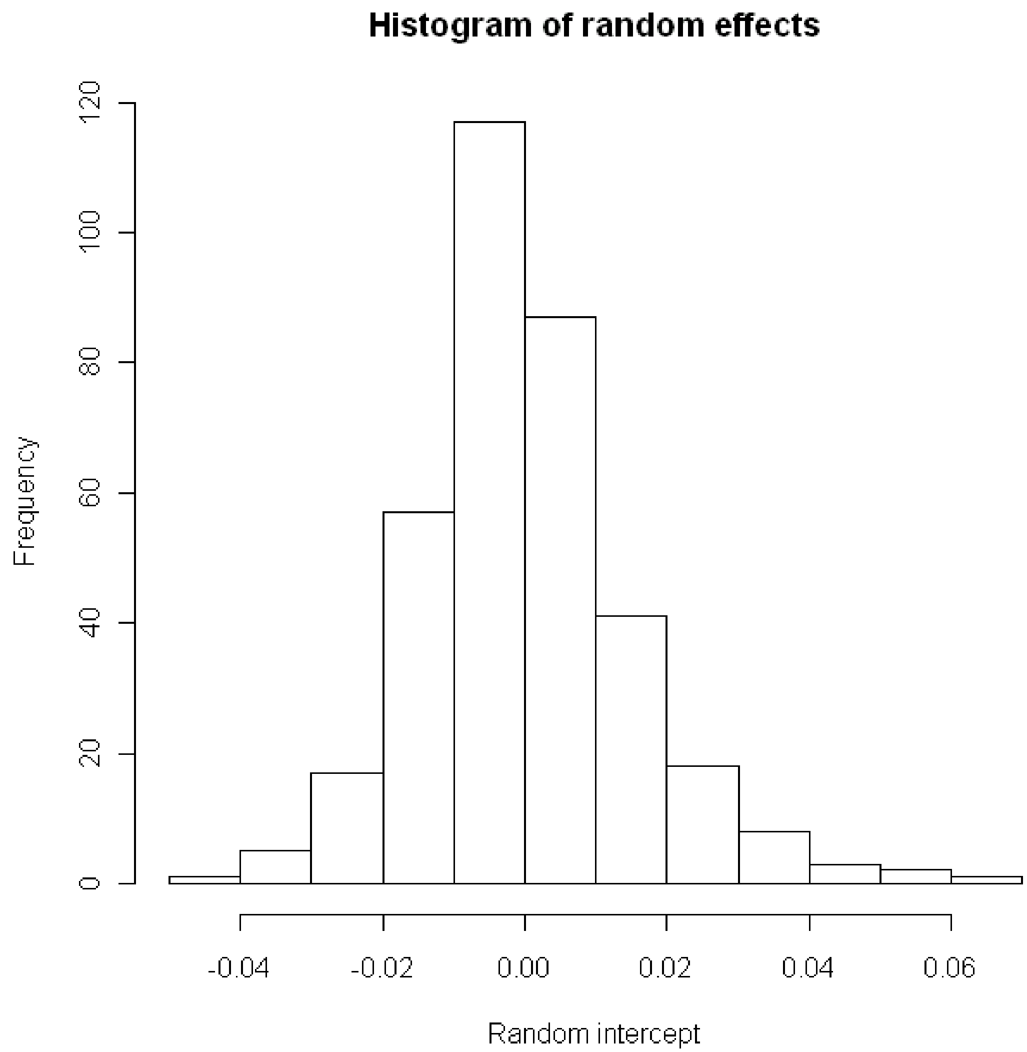
Syntactic annotation available

A simple model

```
> lmer(log(ID_duration) ~ lSpeechRate + (1 | Speaker_ID), the)
Linear mixed model fit by REML
Formula: log(ID_duration) ~ lSpeechRate + (1 | Speaker_ID)
   Data: the
  AIC  BIC logLik deviance REMLdev
 5144 5173  -2568     5121    5136
Random effects:
 Groups     Name        Variance  Std.Dev.
 Speaker_ID (Intercept) 0.0011172 0.033424
 Residual               0.0997008 0.315754
Number of obs: 9460, groups: Speaker_ID, 357

Fixed effects:
            Estimate Std. Error t value
(Intercept) -2.04607    0.02964  -69.04
lSpeechRate -0.28866    0.01807  -15.97
```

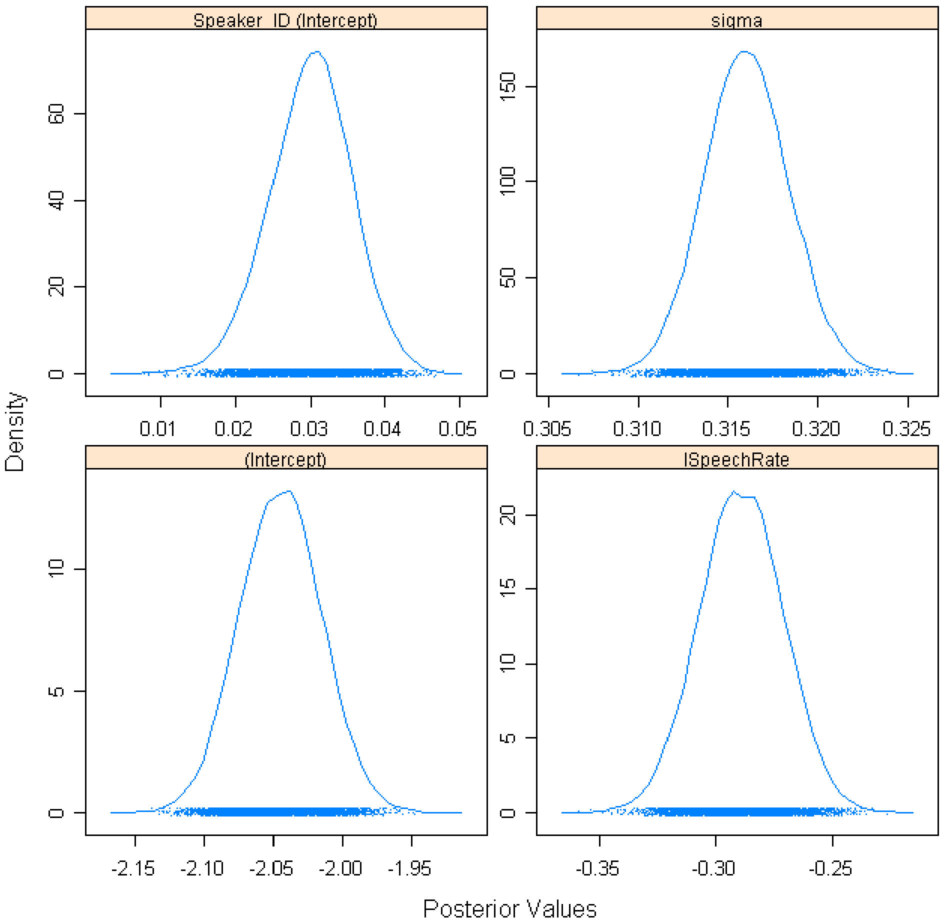
Interpretation?

Interpretation of random effects

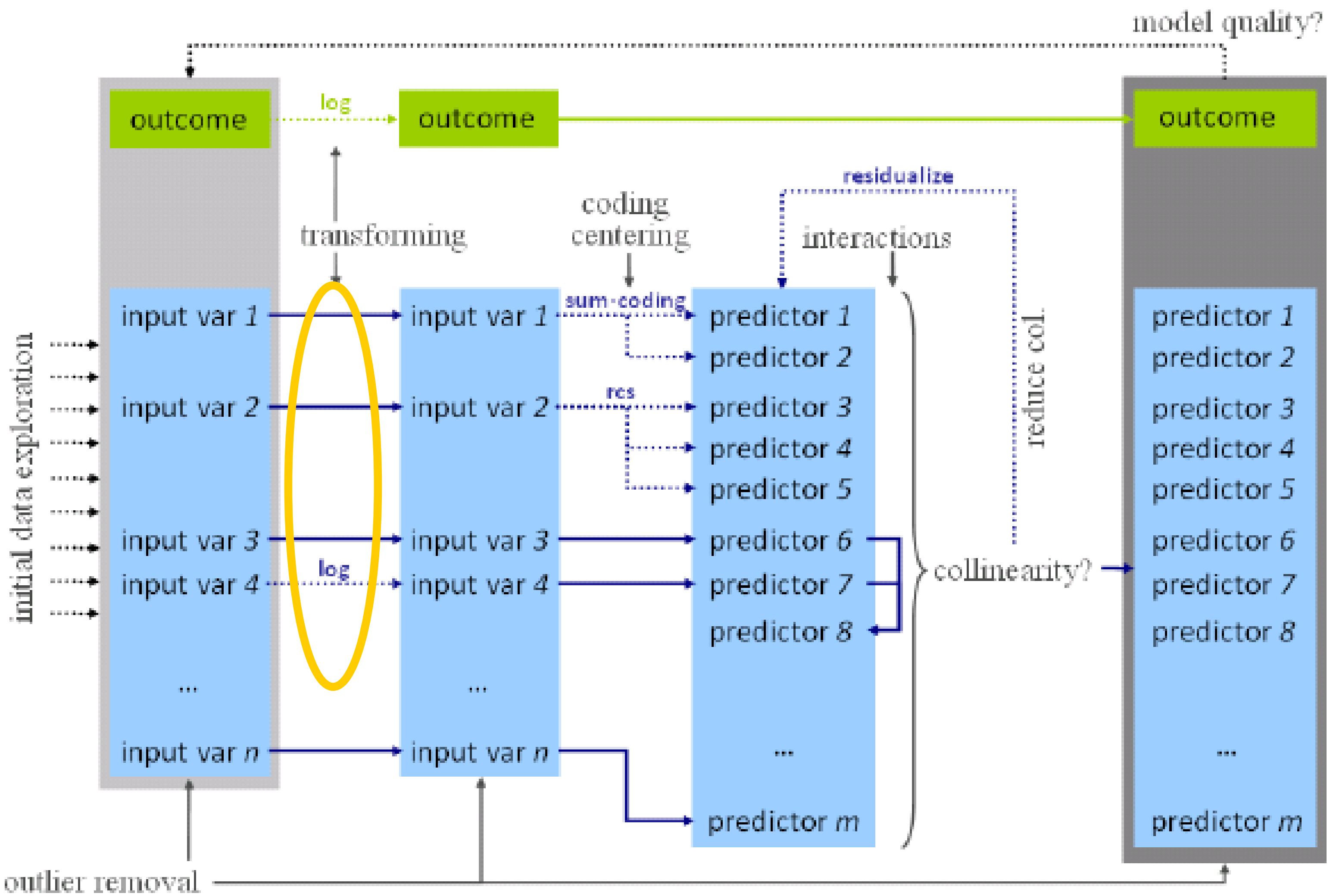


MCMC sampling

```
$fixed
            Estimate MCMCmean HPD95
(Intercept)  -2.0461  -2.0450    -2
lSpeechRate  -0.2887  -0.2892    -0

$random
      Groups        Name Std.Dev. M
1 Speaker_ID (Intercept)   0.0334
2   Residual               0.3158
```



Preparing the data



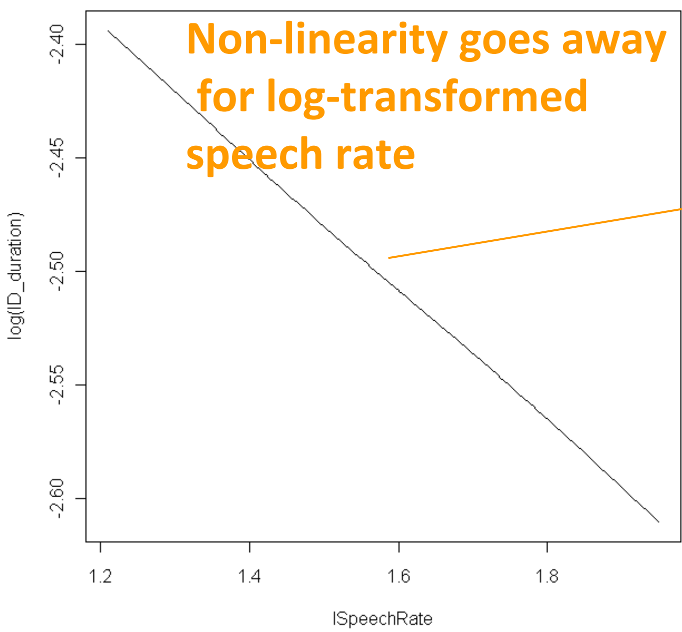
Was log-transform of speech rate justified?

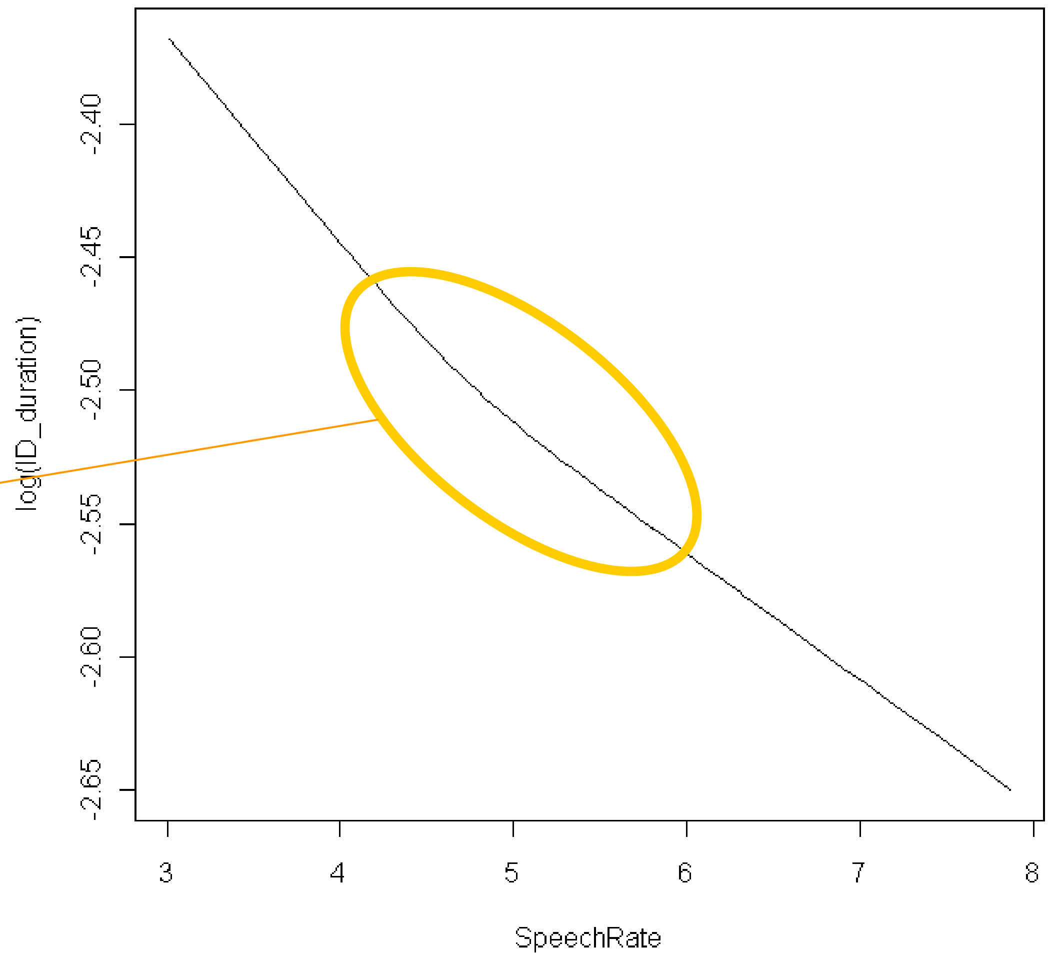
```
Linear mixed model fit by REML
Formula: log(ID_duration) ~ SpeechRate + (1 | Speaker_ID)
   Data: the
  AIC   BIC logLik deviance REMLdev
 5150 5179  -2571     5124    5142
Random effects:
 Groups     Name        Variance  Std.Dev.
 Speaker_ID (Intercept) 0.0011149 0.03339
 Residual               0.0997356 0.31581
Number of obs: 9460, groups: Speaker_ID, 357

Fixed effects:
            Estimate Std. Error t value
(Intercept) -2.22596    0.01864 -119.39
SpeechRate  -0.05602    0.00353  -15.87
```

cf. 5121
for log-transformed
speech rate

Other ways of testing the log-log-linearity assumption

```
l.rcs <- lmer(log(ID_duration) ~ rcs(SpeechRate, 4) + (1 |
   Speaker_ID), the)
plotLMER.fnc(l.rcs)
```





Let’s add some more controls

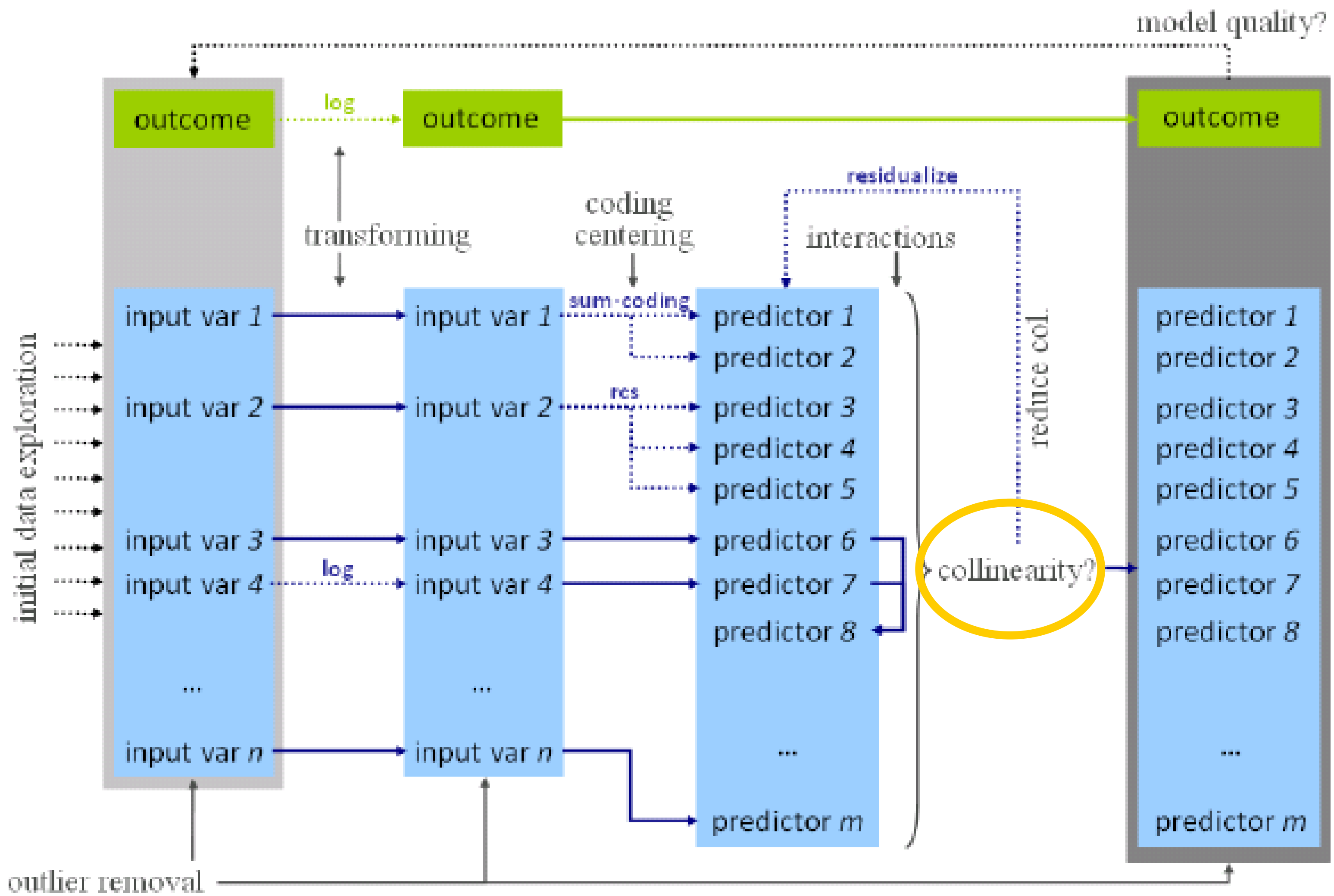
```
Formula: log(ID_duration) ~ lSpeechRate + Dpreceding +
   Dfollowing + (1 | Speaker_ID)
    Data: the
   AIC   BIC logLik deviance REMLdev
  4680 4723  -2334     4640    4668
Random effects:
 Groups     Name        Variance  Std.Dev.
 Speaker_ID (Intercept) 0.0011561 0.034002
 Residual               0.0947221 0.307770
Number of obs: 9460, groups: Speaker_ID, 357

Fixed effects:
            Estimate Std. Error t value
(Intercept) -2.12832    0.02915  -73.02
lSpeechRate -0.25013    0.01771  -14.12
Dpreceding   0.25645    0.01317   19.48
Dfollowing   0.34471    0.03221   10.70
```

cf. 5121 for speech rate-only model

Pretty much unchanged

cf. -0.289 for speech rate-only model

Preparing the data

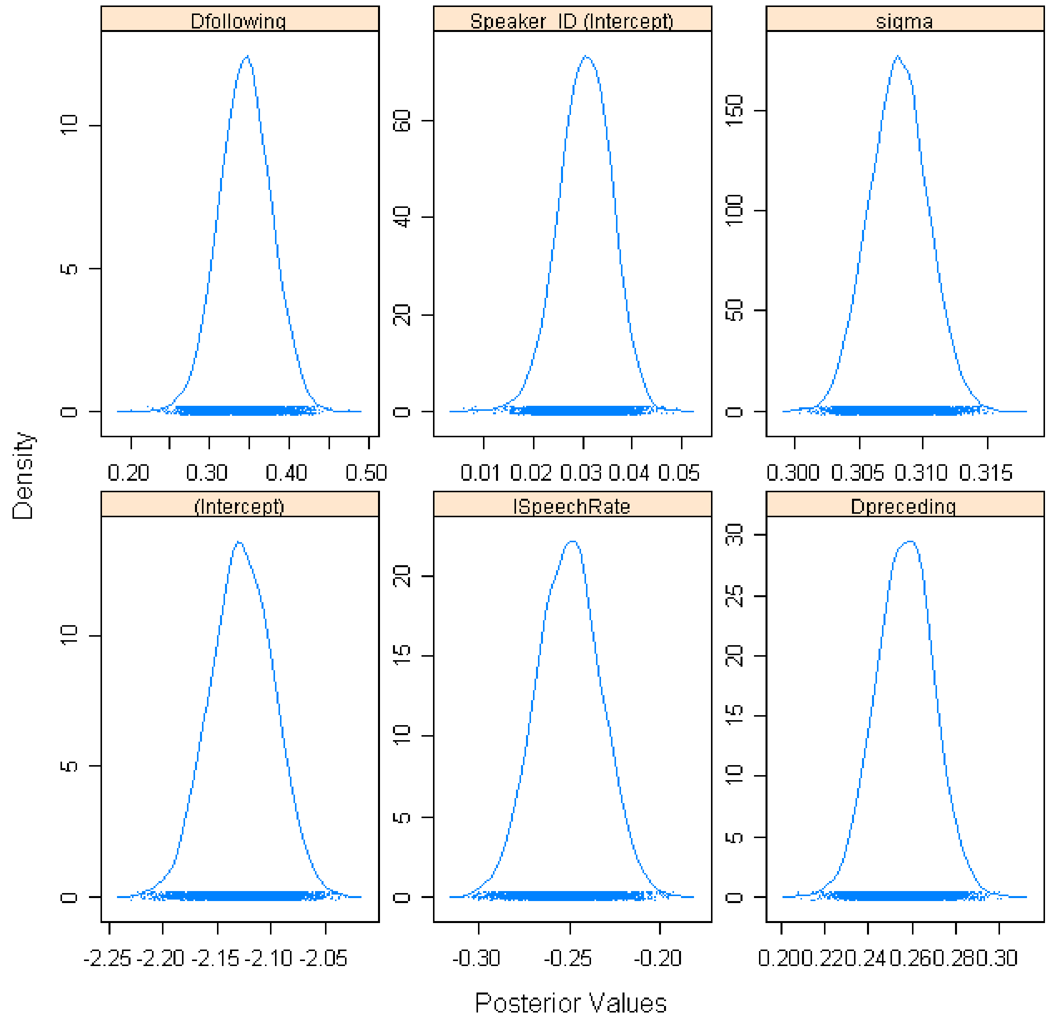


Collinearity?

```
Linear mixed model fit by REML
Formula: log(ID_duration) ~ lSpeechRate + Dpreceding +
   Dfollowing + (1 | Speaker_ID)

…
Correlation of Fixed Effects:
            (Intr) lSpchR Dprcdn
lSpeechRate -0.991
Dpreceding  -0.117  0.089
Dfollowing  -0.050  0.040  0.003
```

MCMC

```
$fixed
            Estimate MCMCmean HPD
(Intercept)  -2.1283  -2.1274
lSpeechRate  -0.2501  -0.2506
Dpreceding    0.2564   0.2565
Dfollowing    0.3447   0.3445

$random
      Groups        Name Std.Dev.
1 Speaker_ID (Intercept)   0.0340
2   Residual               0.3078
```



And some social variables

```
Formula: log(ID_duration) ~ lSpeechRate + Dpreceding +
   Dfollowing + SpeakerMale * lSpeakerAge + (1 | Speaker_ID)

…
Fixed effects:
                           Estimate Std. Error t value
(Intercept)               -2.0880825  0.0734325 -28.435
lSpeechRate               -0.2503733  0.0177274 -14.124
Dpreceding                 0.2564438  0.0131694  19.473
Dfollowing                 0.3449449  0.0322164  10.707
SpeakerMale                0.0023400  0.0963933   0.024
lSpeakerAge               -0.0117168  0.0189518  -0.618
SpeakerMale:lSpeakerAge    0.0003133  0.0271546   0.012
```

Collinearity!

```
…
 Effects:
            (Intr) lSpchR Dprcdn Dfllwn SpkrMl lSpkrA
lSpeechRate -0.384
Dpreceding  -0.039  0.090
Dfollowing  -0.020  0.040  0.003
SpeakerMale -0.643 -0.018 -0.011 -0.003
lSpeakerAge -0.917 -0.009 -0.008 -0.001  0.701
SpkrMl:lSpA  0.637  0.015  0.011  0.004 -0.997 -0.698
```

Definition of collinearity

- **Collinearity**: a predictor is collinear with other predictors in the model if there are high (partial) correlations between them.
- Even if a predictor is not highly correlated with any single other predictor in the model, it can be highly collinear with the combination of predictors $\rightarrow$ collinearity will affect the predictor
- This is not uncommon!
  - in models with many predictors
  - when several somewhat related predictors are included in the model (e.g. word length, frequency, age of acquisition)

Consequences of collinearity

- $\longrightarrow$ standard errors $\mathrm{SE}(\beta)$s of collinear predictors are biased (*in*flated).
  - $\longrightarrow$ *tends* to underestimate significance (but see below)
- $\longrightarrow$ coefficients β of collinear predictors become hard to interpret (though not biased)
  - 'bouncing betas': minor changes in data might have a major impact on βs
  - coefficients will flip sign, double, half
- $\longrightarrow$ coefficient-based tests don't tell us anything reliable about collinear predictors!

So what does collinearity do?

- Type II error increases → power loss
- Type I error does not increase much (5.165% Type I error for two predictors with $r > 0.9989$ in joined model vs. 5.25% in separate models; 20,000 simulation runs with 100 data points each)

```
set.seed(1)
n <- 100
M <- 20000
f <- function(n) {
 x <- runif(n)
 y <- x + rnorm(n,0,0.01)
 z <- rnorm(n,0,5)
 m <- lm(z ~ x + y)
 mx <- lm(z ~ x)
 my <- lm(z ~ y)
 signifmin <- ifelse(min(summary(m)$coef[2:3,4]) < 0.05, 1, 0)
 signifx <- ifelse(min(summary(mx)$coef[2,4]) < 0.05, 1, 0)
 signify <- ifelse(min(summary(my)$coef[2,4]) < 0.05, 1, 0)
 signifxory <- ifelse(signifx == 1 | signify == 1, 1, 0)
 return(c(cor(x,y),signifmin,signifx,signify,signifxory))
}
result <- sapply(rep(n,M), f)
sum(result[2,])/M # joined model returns >=1 spurious effect
sum(result[3,])/M
sum(result[4,])/M
sum(result[5,])/M # two individual models return >=1 spurious effect
min(result[1,])
```

Detecting collinearity

- Mixed model output in R comes with correlation matrix (cf. previous slide).
  - Partial correlations of fixed effects *in the model*.
- Also useful: correlation matrix (e.g. `cor()`; use Spearman option for categorical predictors) or `pairscor.fnc()` in `languageR` for visualization.
  - apply to predictors (not to untransformed input variables)!

```
> cor(lexdec[,c(2,3,10, 13)])
                  RT        Trial    Frequency      Length
RT         1.0000000 -0.052411295 -0.213249525  0.146738111
Trial     -0.0524113  1.000000000 -0.006849117  0.009865814
Frequency -0.2132495 -0.006849117  1.000000000 -0.427338136
Length     0.1467381  0.009865814 -0.427338136  1.000000000
```

Dealing with collinearity

- Good news: Estimates are only problematic for those predictors that are collinear.

→ If collinearity is in the nuisance predictors (e.g. certain controls), nothing needs to be done.

- Somewhat good news: If collinear predictors are of interest but we are *not* interested in the direction of the effect, we can use ↷**model comparison** (rather than tests based on the standard error estimates of coefficients).

- If collinear predictors are of interest and we *are* interested in the direction of the effect, we need to reduce collinearity of those predictors.

Reducing collinearity

- **Centering**: reduces collinearity of predictor with intercept and higher level terms involving the predictor.
  - **pros:** easy to do and interpret; often improves interpretability of effects.
  - **cons:** none?
- **Re-express the variable** based on conceptual considerations (e.g. ratio of spoken vs. written frequency in `lexdec`; rate of disfluencies per words when constituent length and fluency should be controlled).
  - **pros:** easy to do and relatively easy to interpret.
  - **cons:** only applicable in some cases.

Collinearity is gone (nice)

```
…
 Correlation of Fixed Effects:
            (Intr) lSpchR Dprcdn Dfllwn cSpkrM clSpkA
lSpeechRate -0.991
Dpreceding  -0.117  0.090
Dfollowing  -0.050  0.040  0.003
cSpeakerMal  0.029 -0.036 -0.001  0.014
clSpeakerAg -0.003  0.001 -0.001  0.003  0.097
cSpkrMl:cSA -0.002  0.015  0.011  0.004  0.007 -0.094
```

After centering

```
tion) ~ lSpeechRate + Dpreceding + Dfollowing + cSpeakerMale
   * clSpeakerAge + (1 | Speaker_ID)
…
Fixed effects:
                            Estimate Std. Error t value
(Intercept)               -2.1280429  0.0291686  -72.96
lSpeechRate               -0.2503733  0.0177274  -14.12
Dpreceding                 0.2564438  0.0131694   19.47
Dfollowing                 0.3449449  0.0322164   10.71
cSpeakerMale               0.0034491  0.0078396    0.44
clSpeakerAge              -0.0115789  0.0136307   -0.85
cSpeakerMale:clSpeakerAge  0.0003133  0.0271546    0.01
```

Here: no change in significance
(social effects still insignificant)
but now we can trust the results

Driven by the phonological complexity of surrounding coda/onsets?

Addition of phonological complexity: $\chi^2(2)=577.5$, $p< 0.0001$

Removal of OCP effects: $\chi^2(3)=117.1$, $p< 0.0001$

Partial shadowed effect or collinearity?

```
Fixed effects:
                           Estimate Std. Error t value
(Intercept)               -2.529879   0.003873  -653.2
clSpeechRate              -0.287437   0.017237   -16.7
Dpreceding                 0.185212   0.013473    13.7
Dfollowing                 0.292674   0.031308     9.3
cOnsetPrecedingCodaOCP     0.019685   0.007426     2.7
cOnsetPrecedingOnsetOCP    0.065366   0.008069     8.1
cOnsetFollowingOnsetOCP    0.052071   0.007457     7.0
cCodaClusterPreceding     -0.095043   0.006164   -15.4
cOnsetClusterFollowing    -0.118048   0.006148   -19.2
cSpeakerMale               0.004391   0.007552     0.6
clSpeakerAge              -0.006175   0.013131    -0.5
cSpeakerMale:clSpeakerAge  0.002613   0.026160     0.1
```

Mild collinearity

```
Correlation of Fixed Effects:
            (Intr) clSpcR Dprcdn Dfllwn cOPCOC cOPOOC cOFOOC cCdClP …
clSpeechRat -0.024
Dpreceding  -0.218  0.098
Dfollowing  -0.082  0.047  0.007
cOnstPrCOCP -0.016 -0.016  0.072 -0.006
cOnstPrOOCP  0.030  0.016 -0.122 -0.005  0.088
cOnstFlOOCP -0.002 -0.001  0.003  0.018  0.006  0.003
cCdClstrPrc -0.060  0.057  0.284  0.010 -0.132 -0.064  0.001
cOnstClstrF -0.011  0.078  0.014  0.083 -0.011  0.005 -0.276  0.020
…
```

What to do if centering is not going to help?

- **Residualization**: Regress collinear predictor against combination of (partially) correlated predictors
  - usually using ordinary regression (e.g. `lm()`, `ols()`).
  - **pros:** systematic way of dealing with multicollinearity; directionality of (conditional) effect interpretable
  - **cons:** effect sizes hard to interpret; judgment calls: what should be residualized against what?

```
the$rOnsetFollowingOnsetOCP <- residuals(lm(cOnsetFollowingOnsetOCP
   ~ cOnsetClusterFollowing, the))
```

…

```
Correlation of Fixed Effects:
            (Intr) clSpcR Dprcdn Dfllwn cOPCOC cOPOOC rOFOOC cCdClP
clSpeechRat -0.024
Dpreceding  -0.218  0.098
Dfollowing  -0.082  0.047  0.007
cOnstPrCOCP -0.016 -0.016  0.072 -0.006
cOnstPrOOCP  0.030  0.016 -0.122 -0.005  0.088
rOnstFlOOCP -0.002 -0.001  0.003  0.018  0.006  0.003
cCdClstrPrc -0.060  0.057  0.284  0.010 -0.132 -0.064  0.001
cOnstClstrF -0.012  0.081  0.015  0.091 -0.010  0.006  0.002  0.021
```

…

Does availability affect pronunciation?

- Two measures of availability:
  - Frequency of next work
  - (trigram) predictability of next work

```
the$rlCndP_1forward <- residuals(lm(clCndP_1forward ~
   clFQfollowing, the))
l.avail.r <-  lmer(log(ID_duration) ~ clSpeechRate +
   Dpreceding + Dfollowing +
   cOnsetPrecedingCodaOCP + cOnsetPrecedingOnsetOCP +
   cOnsetFollowingOnsetOCP +
   cOnsetPrecedingCodaIdent + cOnsetPrecedingOnsetIdent +
   cOnsetFollowingOnsetIdent +
   cCodaClusterPreceding + cOnsetClusterFollowing +
   clFQfollowing + rlCndP_1forward +
   cSpeakerMale * clSpeakerAge +
   (1 | Speaker_ID) + (1 | WORDpreceding) + (1 |
   WORDfollowing), the)
```

Addition of availability:
$\chi^2(2)=32.3$, $p< 0.0001$

	Estimate	Std. Error	t value
(Intercept)	-2.501850	0.007983	-313.40
clSpeechRate	-0.283996	0.017188	-16.52
Dpreceding	0.051300	0.031609	1.62
Dfollowing	0.287069	0.076187	3.77
cOnsetPrecedingCodaOCP	0.052595	0.015801	3.33
cOnsetPrecedingOnsetOCP	-0.015448	0.015626	-0.99
cOnsetFollowingOnsetOCP	0.047243	0.011309	4.18
cOnsetPrecedingCodaIdent	-0.026780	0.054029	-0.50
cOnsetPrecedingOnsetIdent	0.043565	0.026797	1.63
cOnsetFollowingOnsetIdent	0.089541	0.053460	1.67
cCodaClusterPreceding	-0.089247	0.009703	-9.20
cOnsetClusterFollowing	-0.100809	0.008589	-11.74
clFQfollowing	**-0.010772**	**0.002747**	**-3.92**
rlCndP_1forward	**-0.008563**	**0.001988**	**-4.31**
cSpeakerMale	-0.002354	0.007571	-0.31
clSpeakerAge	-0.004993	0.013109	-0.38
cSpeakerMale:clSpeakerAge	0.001768	0.026105	0.07

Does redundancy affect pronunciation?