LENA Evaluation

Alejandrina Cristia

Contents

Basic stats

To start with, we read the confusion matrices in. In all cases, the columns are the LENA-returned labels, and the rows are the reference/what human annotators said. In the reference, CHI stands for any child; FEM for any female adult, MAL for any mal adult. Overlaps within class (CHI with another CHI) are treated as if it was a sample of the pure class. Overlaps across human voices of different classes are marked with / – that is CHI/FEM indicates frames for which there is overlap across a child and a female adult. We've generated one confusion matrix per subcorpus, plus one for the whole corpora (including the 5 subcorpora). Numbers in these original matrices refer to 100 ms frames.

```
ber=read.table(paste0(thisdir,"BER","_cm.txt"),header=T)
row=read.table(paste0(thisdir,"ROW","_cm.txt"),header=T)
sod=read.table(paste0(thisdir,"SOD","_cm.txt"),header=T)
war=read.table(paste0(thisdir,"WAR","_cm.txt"),header=T)
all=read.table(paste0(thisdir,"all","_cm.txt"),header=T)

#remove empty rows
ber=ber[rownames(ber)[rowSums(ber)>0],]
row=row[rownames(row)[rowSums(row)>0],]
war=war[rownames(war)[rowSums(war)>0],]
sod=sod[rownames(sod)[rowSums(sod)>0],]
all=sod[rownames(all)[rowSums(all)>0],]

sumref=rowSums(all)
```

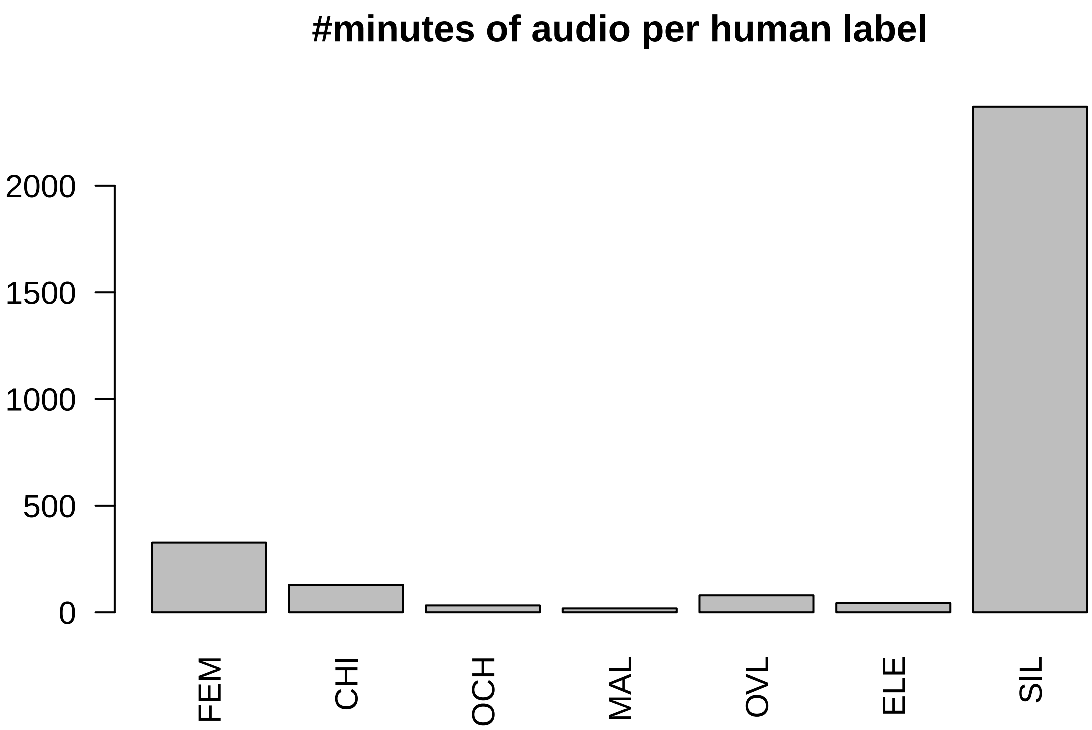
Prevalence of tags within human annotation (intro to results)

How much data has been annotated as being of each of the categories? Remember that each frame is 100 ms; so we divide by 10 to have seconds, and by 60 to have minutes.

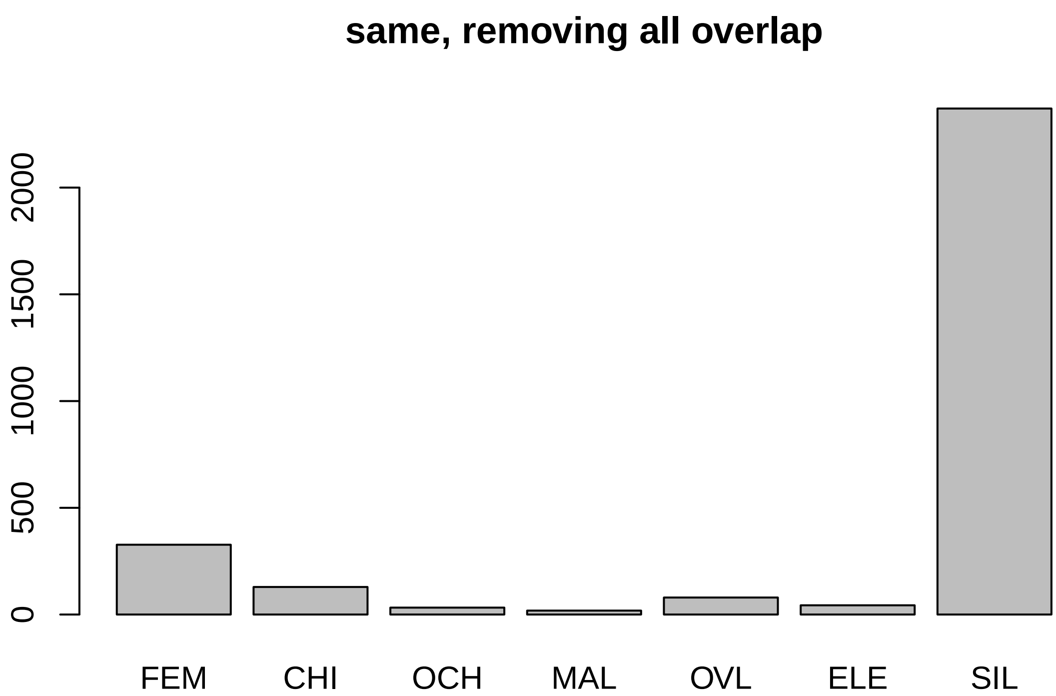
```
n_min_human=rowSums(all,na.rm=T)/10/60

round(n_min_human/sum(n_min_human)*100)
```

```
## FEM CHI OCH MAL OVL ELE SIL
##  11   4   1   1   3   1  79
```

```
barplot(n_min_human,las=2, main="#minutes of audio per human label")
```



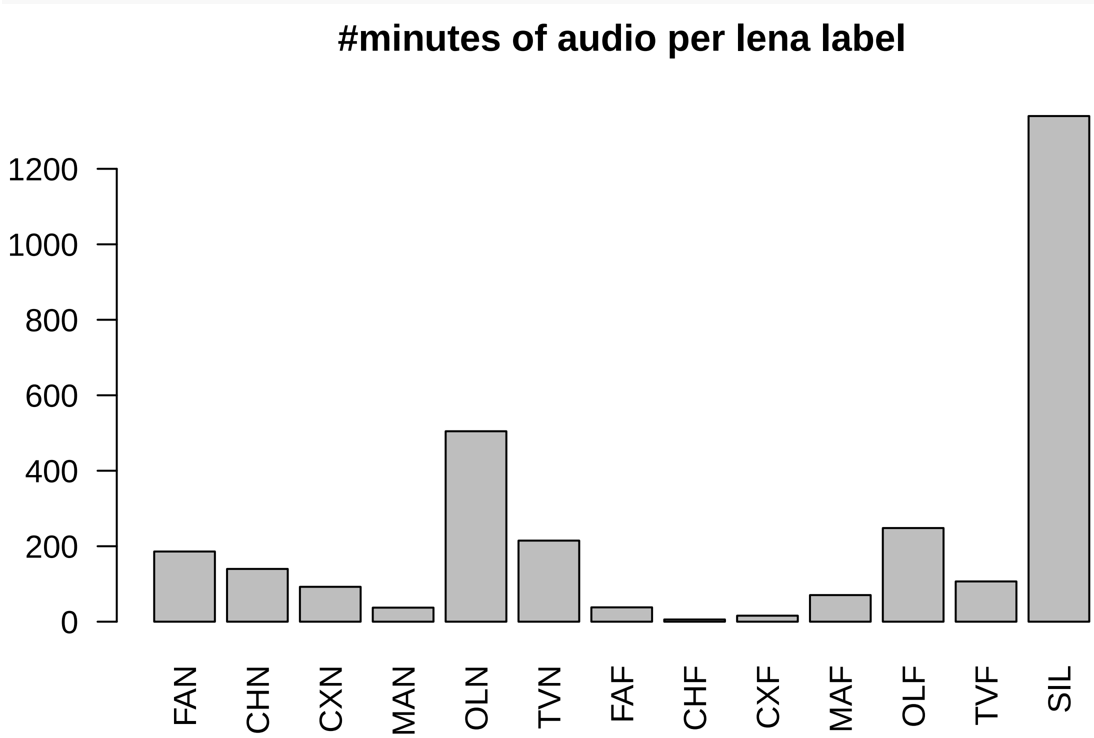

```
barplot(n_min_human[grep("/",names(n_min_human),invert=T)], main="same, removing all overlap")
```



Prevalence of tags within LENA annotation (not discussed in paper, just fyi)

How much data has been automatically tagged as being of each of the categories? Remember that each frame is 100 ms; so we divide by 10 to have seconds, and by 60 to have minutes.

```
n_min_lena=colSums(all,na.rm=T)/10/60

barplot(n_min_lena,las=2, main="#minutes of audio per lena label")
```



Diarization analyses

While confusion matrices are useful to understand patterns of errors, it is sometimes desirable to have indices of performance that are more specific to the task of diarization. To this end, we use a number of metrics provided by pyannote.metrics.

THIS NEEDS TO BE UPDATED TO THE 201906 RESULTS

```
read.csv(paste0(thisdir,"/gold/mapped/diaer__report.csv"))->py
dim(py) #873 clips
```

```
## [1] 704  11
```

```
summary(py)
```

```
##                            item     diarization.error.rate..
##  BER_0396_005220_005340.rttm:  1   Min.   :    0.00
##  BER_0396_008100_008220.rttm:  1   1st Qu.:   59.37
##  BER_0396_008820_008940.rttm:  1   Median :   96.14
##  BER_0396_010140_010260.rttm:  1   Mean   :  167.96
##  BER_0396_011160_011280.rttm:  1   3rd Qu.:  107.66
##  BER_0396_012900_013020.rttm:  1   Max.   :10059.52
##  (Other)                    :698
##      total              correct            correct..
##  Min.   :    0.00   Min.   :   0.000   Min.   :  0.00
##  1st Qu.:    0.00   1st Qu.:   0.000   1st Qu.: 37.09
##  Median :   14.86   Median :   6.295   Median : 47.48
##  Mean   :   53.12   Mean   :  24.675   Mean   : 47.95
##  3rd Qu.:   46.98   3rd Qu.:  20.785   3rd Qu.: 58.64
##  Max.   :18696.97   Max.   :8685.490   Max.   :100.00
##                                        NA's   :265
##   false.alarm        false.alarm..      missed.detection
##  Min.   :    0.000   Min.   :    0.00   Min.   :   0.000
##  1st Qu.:    3.422   1st Qu.:   18.10   1st Qu.:   0.000
##  Median :   12.945   Median :   40.12   Median :   1.485
##  Mean   :   46.735   Mean   :  186.32   Mean   :  11.400
##  3rd Qu.:   34.148   3rd Qu.:   87.51   3rd Qu.:   7.418
##  Max.   :16450.590   Max.   :10059.52   Max.   :4012.690
##                      NA's   :265
##  missed.detection..   confusion         confusion..
##  Min.   :  0.000    Min.   :   0.00   Min.   : 0.00
##  1st Qu.:  7.615    1st Qu.:   0.00   1st Qu.:16.44
##  Median : 18.346    Median :   2.64   Median :29.30
##  Mean   : 23.662    Mean   :  17.04   Mean   :28.39
##  3rd Qu.: 32.463    3rd Qu.:  13.06   3rd Qu.:39.69
##  Max.   :100.000    Max.   :5998.79   Max.   :63.77
##  NA's   :265                          NA's   :265
```

```
#294 FA, MI, confusion are NA because no speech at all in the clip

round(sum(is.na(py$confusion..))/dim(py)[1]*100)  # percentage of clips with no speech REPORTED IN PAPE
```

```
## [1] 38
```

```
# DER and FA can be very high when there is little speech in a clip
# so we NA anything that is more than 2 SDs above the mean
py$false.alarm..[py$false.alarm..> (mean(py$false.alarm..,na.rm=T) + 2 * sd(py$false.alarm..,na.rm=T))]
```

```r
py$diarization.error.rate[py$diarization.error.rate> (mean(py$diarization.error.rate,na.rm=T) + 2 * sd(p

summary(py) #the exclusion above affected 314-294=20 clips in FA, 7 in DIAER
```

```
##                             item     diarization.error.rate..
##  BER_0396_005220_005340.rttm:  1   Min.   :    0.00
##  BER_0396_008100_008220.rttm:  1   1st Qu.:   59.37
##  BER_0396_008820_008940.rttm:  1   Median :   96.14
##  BER_0396_010140_010260.rttm:  1   Mean   :  167.96
##  BER_0396_011160_011280.rttm:  1   3rd Qu.:  107.66
##  BER_0396_012900_013020.rttm:  1   Max.   :10059.52
##  (Other)                    :698
##      total              correct            correct..
##  Min.   :    0.00   Min.   :   0.000   Min.   :  0.00
##  1st Qu.:    0.00   1st Qu.:   0.000   1st Qu.: 37.09
##  Median :   14.86   Median :   6.295   Median : 47.48
##  Mean   :   53.12   Mean   :  24.675   Mean   : 47.95
##  3rd Qu.:   46.98   3rd Qu.:  20.785   3rd Qu.: 58.64
##  Max.   :18696.97   Max.   :8685.490   Max.   :100.00
##                                        NA's   :265
##   false.alarm         false.alarm..     missed.detection
##  Min.   :    0.000   Min.   :   0.00   Min.   :   0.000
##  1st Qu.:    3.422   1st Qu.:  17.82   1st Qu.:   0.000
##  Median :   12.945   Median :  39.62   Median :   1.485
##  Mean   :   46.735   Mean   :  92.70   Mean   :  11.400
##  3rd Qu.:   34.148   3rd Qu.:  84.70   3rd Qu.:   7.418
##  Max.   :16450.590   Max.   :1765.27   Max.   :4012.690
##                      NA's   :273
##  missed.detection..   confusion         confusion..
##  Min.   :  0.000    Min.   :   0.00   Min.   : 0.00
##  1st Qu.:  7.615    1st Qu.:   0.00   1st Qu.:16.44
##  Median : 18.346    Median :   2.64   Median :29.30
##  Mean   : 23.662    Mean   :  17.04   Mean   :28.39
##  3rd Qu.: 32.463    3rd Qu.:  13.06   3rd Qu.:39.69
##  Max.   :100.000    Max.   :5998.79   Max.   :63.77
##  NA's   :265                          NA's   :265
##  diarization.error.rate
##  Min.   :   0.00
##  1st Qu.:  58.14
##  Median :  95.66
##  Mean   : 106.91
##  3rd Qu.: 105.27
##  Max.   :1312.21
##  NA's   :9
```

```r
py$DER=py$false.alarm..+py$missed.detection..+py$confusion..

round(sum(is.na(py$false.alarm..))/dim(py)[1]*100)
```

```
## [1] 39
```

```r
round(sum(is.na(py$DER))/dim(py)[1]*100)
```

```
## [1] 39
```

```r
#other fixes and addtions
py=subset(py,item!="TOTAL")
py$cor=substr(py$item,1,3)
py$cor[grep("[0-9]",py$cor)]<-"tsi"
py$cor=factor(py$cor)

py$child=substr(py$item,1,8)
py$child[py$cor=="tsi"]=substr(py$item[py$cor=="tsi"],1,3)

# add in age info
spreadsheet = read.csv(paste0(thisdir,"/ACLEW_list_of_corpora.csv"), header=TRUE, sep = ",")
spreadsheet$child=paste0(spreadsheet$labname,"_",ifelse(nchar(spreadsheet$aclew_id)==3,paste0("0",spreac
spreadsheet = spreadsheet[,c("child","age_mo_round")]
colnames(spreadsheet) = c("child","age")
spreadsheet_tsi = read.csv(paste0(thisdir,"/anon_metadata.csv"), header=TRUE, sep = ",")
spreadsheet_tsi = spreadsheet_tsi[c("id","age_mo")]
colnames(spreadsheet_tsi) = c("child","age")
age_id = rbind(spreadsheet, spreadsheet_tsi)
py=merge(py,age_id,by.x="child",by.y="child",all.x=T)
```

Variable explanation:

- "item": filename

- "diarization.error.rate": sum of false.alarm.., missed.detection.., and confusion..
- "total": total duration of voiced frames (counting overlapping speech repeatedly if necessary), in seconds
- "correct": duration of diarization that is correct, in seconds

- "correct..": percentage of duration that is correct

- "false.alarm": duration of segments where the system said there was speech, whereas in fact there was not

- "false.alarm..": percentage of duration with false alarms

- "missed.detection": duration of segments where there was speech but the system missed it

- "missed.detection..": percentage of duration with missed speech

- "confusion": duration of segments assigned to the wrong speaker

- "confusion..": percentage of duration with assignment to the wrong speaker
- cor: corpus ID
- child: child ID

Performance in 3 standard speech tech metrics

```r
length(py$false.alarm..) #total N of clips
```

```
## [1] 812
```

```r
summary(py$false.alarm..)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   19.59   44.53  104.84   91.03 1765.27     280
```

```r
summary(py$missed.detection..)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   7.615  18.461  23.820  32.829 100.000     265
```

```r
summary(py$confusion..)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   15.86   28.69   27.89   38.98   63.77     265
```

Confusion matrices

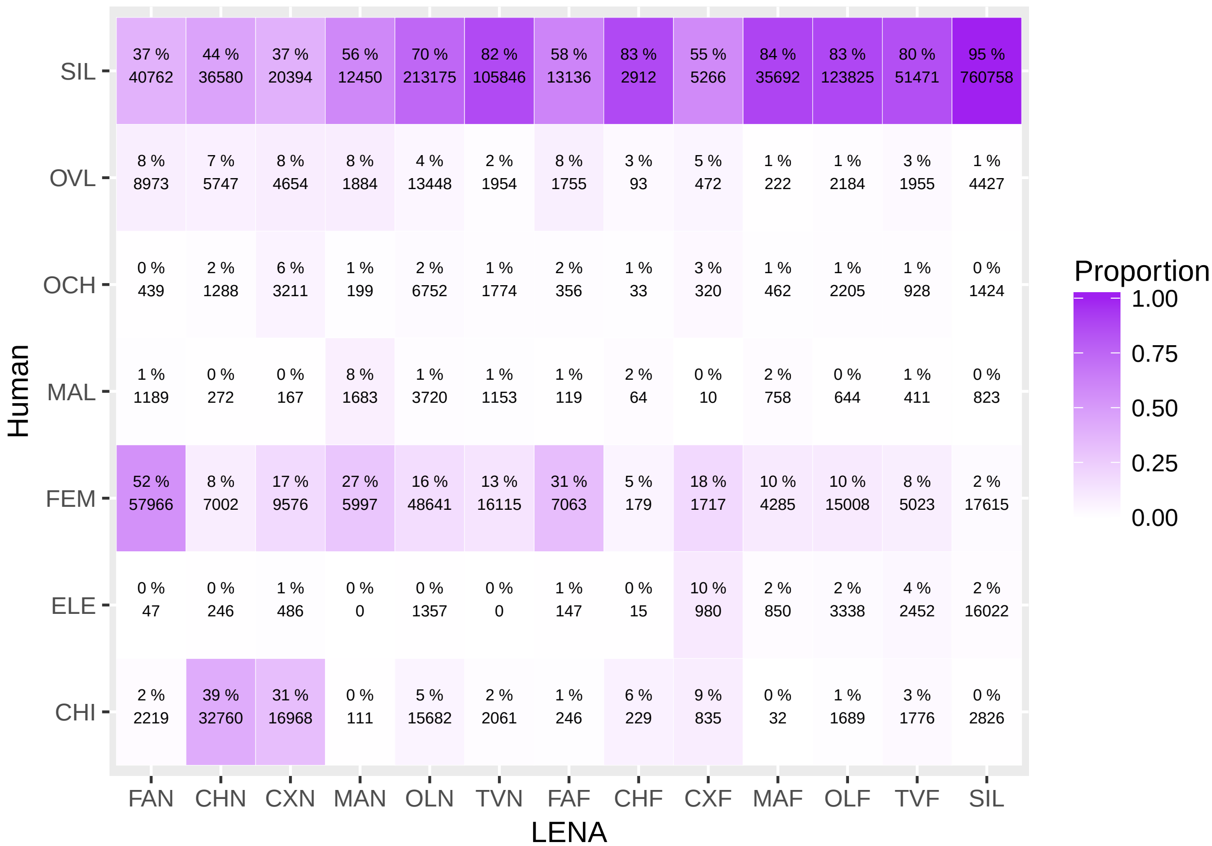
Precision massive ana

Here we just plot the most fine-grained analysis, it is precision because we divide by the sum of the columns –> the LENA categories

```r
prop_cat=data.frame(apply(all,2,dodiv)*100) #generates precision because columns
colSums(prop_cat)
```

```
## FAN CHN CXN MAN OLN TVN FAF CHF CXF MAF OLF TVF SIL
## 100 100 100 100 100 100 100 100 100 100 100 100 100
```

```r
stack(all)->stall
colnames(stall)<-c("n","lena")
stall$human=rownames(all)
stall$pr=stack(prop_cat)$values


ggplot(data = stall, mapping = aes(y = stall$human, x=stall$lena)) +
  geom_tile(aes(fill= rescale(stall$pr)), colour = "white") +
  geom_text(aes(label = paste(round(stall$pr),"%")), vjust = -1,size=2) +
  geom_text(aes(label = stall$n), vjust = 1,size=2) +
  scale_fill_gradient(low = "white", high = "purple", name = "Proportion") +
  xlab("LENA") + ylab("Human")
```



Recall massive ana

Here we just plot the most fine-grained analysis, it is precision because we divide by the sum of the rows –> the human categories

```
prop_cat=data.frame(t(apply(all,1,dodiv)*100)) #generates recall because rows
rowSums(prop_cat)
```

```
## FEM CHI OCH MAL OVL ELE SIL
## 100 100 100 100 100 100 100
```

```
stall$pr=stack(prop_cat)$values

ggplot(data = stall, mapping = aes(y = stall$human, x=stall$lena)) +
  geom_tile(aes(fill= rescale(stall$pr)), colour = "white") +
  geom_text(aes(label = paste(round(stall$pr),"%")), vjust = -1,size=2) +
  geom_text(aes(label = stall$n), vjust = 1,size=2) +
  scale_fill_gradient(low = "white", high = "purple", name = "Proportion") +
  xlab("LENA") + ylab("Human")
```

Overlap

To simplify the discussion, I'd like to first deal with the problem of overlap. It transpires that overlaps are rare, affecting 3% of frames. LENA's accuracy for detecting these overlaps can be conveyed by asking what proportion of frames with overlap get a LENA overlap label (28.1527382, 4.5720985 for OLF and OLN respectively); versus what proportion of frames that do not have an overlap in the referent nonetheless get a LENA overlap label (16.5119117, 8.3726927 for OLF and OLN respectively). My reading of these results is that the overlap detection system is not very good in that it only catches 24% of the cases of overlap (and also false-alarms 9% of non-overlaps as if they were overlaps). Nonetheless, since overlap is overall rare, we'll set aside overlap in the reference completely, by removing from consideration frames with overlap.

QUESTION: In what follows, we could take them into account in the denominator (because they still exist) or we could remove them from consideration altogether (remove lines from confusion matrix in which there is overlap; remove LENA's overlap-label columns). –> **decision**: take them into account.

Silence

Next, we check LENA's accuracy to label silence as something other than voice. The absent of live human voices is extremely prevalent, with 79% of frames containing no human vocalization, what we call silence.

LENA has many categories that apply in this space (electronics, noise, silence) but since these were not separately categorized in our human annotation, we collapsed across all of them into one large category that we call silence.

Here, we would like to check, among the frames the human annotator also did not detect any voice for, what proportion were labeled as silence/TV/noise/FUZ by LENA ("silent frames" in the referent 53.4891128); versus what proportion of frames that do have some voice in the referent nonetheless get a LENA SILENCE label (11.4199712). My reading of these results is that the silence detection system does not over-apply these non-voice labels when the voice is present – in only 7% of cases where there is a voice present, did the system return one of these labels. However, it fails to return a "SILENCE" label for the vast majority of frames that truly do not contain voice (62% of frames that are truly silent according to the human annotation failed to get a silent-like LENA label).

Voice confusions

As can be imagined by the description above, a lot of frames get some LENA voicing label – that is, LENA returns a label of child or adult for many frames. Is the actual label that is applied appropriate? To assess this, we provide the simplified confusion matrices below.

For this first one, I calculated the percentages shown in the table taking as total the columns – meaning that this shows *precision*. For instance, out of all the frames where the LENA returned a CHN label, 65% did actually contain a child voice, 11% contained the voice of a female adult or male according tot he human annotator, and 25% did not have any voice (according to the human annotator).

% is Precision

```
prop_cat=data.frame(apply(all,2,dodiv)*100) #generates precision because columns
colSums(prop_cat)
```

```
## FAN CHN CXN MAN OLN TVN FAF CHF CXF MAF OLF TVF SIL
## 100 100 100 100 100 100 100 100 100 100 100 100 100
```

```
stack(all)->stol
colnames(stol)<-c("n","lena")
stol$human=rownames(all)
stol$pr=stack(prop_cat)$values

png("precision.png",width=5.5,height=5,units="in",res=300)
ggplot(data = stol, mapping = aes(y = stol$human, x=stol$lena)) +
  geom_tile(aes(fill= rescale(stol$pr)), colour = "white") +
  geom_text(aes(label = paste(round(stol$pr),"%")), vjust = -1,size=2) +
  geom_text(aes(label = stol$n), vjust = 2,size=2) +
  scale_fill_gradient(low = "white", high = "purple", name = "Proportion") +
  xlab("LENA") + ylab("Human")  +ggtitle("% is Precision (cols)")
dev.off()
```

```
## pdf
##   2
```

```
#we report precision-sil for far versus near
mean(stol$pr[substr(stol$lena,3,3)=="F" & substr(stol$lena,1,2)!="NO" & stol$human=="SIL"])
```

```
## [1] 73.82766
```

```
mean(stol$pr[substr(stol$lena,3,3)=="N"& substr(stol$lena,1,2)!="NO" & stol$human=="SIL"])
```

```
## [1] 54.19892
```

My interpretation of these results is that frame containing LENA's CHN and FAN do, more often than not, contain audio corresponding to those categories. CHF and MAF very seldom contain child/male adult material respectively. FAF and MAN are more intermediate in performance: FAF more often contains silence than female adult voice, but it still contains female voice a good third of the time. MAN contains a male voice 46% of the time, with 52% of the time actually consisting of silence or female voice.

For this next matrix, I calculated the percentages shown in the table taking as total the rows – meaning that this shows *recall*. For instance, out of all the frames that had a child voice (that did not overlap with others according to human annotators), 53% were labeled CHN by LENA; 23% were labeled OLN; 13% were labeled SIL.

% is Recall

```
prop_cat=data.frame(t(apply(all,1,dodiv)*100)) #generates recall because rows
rowSums(prop_cat)
```

```
## FEM CHI OCH MAL OVL ELE SIL
## 100 100 100 100 100 100 100
```

```
stack(all)->stol
colnames(stol)<-c("n","lena")
stol$human=rownames(all)
stol$pr=stack(prop_cat)$values

png("recall.png",width=5.5,height=5,units="in",res=300)

ggplot(data = stol, mapping = aes(y = stol$human, x=stol$lena)) +
  geom_tile(aes(fill= rescale(stol$pr)), colour = "white") +
  geom_text(aes(label = paste(round(stol$pr),"%")), vjust = -1,size=2) +
  geom_text(aes(label = stol$n), vjust = 2,size=2) +
  scale_fill_gradient(low = "white", high = "purple", name = "Proportion") +
  xlab("LENA") + ylab("Human")  +ggtitle("% is Recall (rows)")
dev.off()
```

```
## pdf
##   2
```

My interpretation of these results is that LENA did a fairly good job of labeling child frames as such; however, the performance of both female and male adult voices is lower: 30-32% are labeled as the "right" category, with the rest being labeled as other categories.

CVC and CTC

```
read.table(paste0(thisdir,"cvtc.txt"),header=T)->cvtc


png("cvc.png",width=5.5,height=5,units="in",res=300)

plot(CVC_n~CVC_gold,data=cvtc,pch=20,main="child voc counts (near only)",col=alpha("black",.2))
abline(lm(CVC_n~CVC_gold,data=cvtc))
abline(lm(CVC_n~CVC_gold,data=cvtc,subset=c(cvtc$CVC_n>0 & cvtc$CVC_gold>0)),lty=2)

dev.off()
```

```
## pdf
##   2
```

```
cor.test(cvtc$CVC_n,cvtc$CVC_gold)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cvtc$CVC_n and cvtc$CVC_gold
```

```
## t = 25.891, df = 701, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6592918 0.7350932
## sample estimates:
##       cor
## 0.6991519
```

```r
cor.test(cvtc$CVC_n[cvtc$CVC_n>0 & cvtc$CVC_gold>0],cvtc$CVC_gold[cvtc$CVC_n>0 & cvtc$CVC_gold>0])
```

```
##
##  Pearson's product-moment correlation
##
## data:  cvtc$CVC_n[cvtc$CVC_n > 0 & cvtc$CVC_gold > 0] and cvtc$CVC_gold[cvtc$CVC_n > 0 & cvtc$CVC_gol
## t = 22.011, df = 329, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7241755 0.8119793
## sample estimates:
##       cor
## 0.7717297
```

```r
# RER against human

sum(cvtc$CVC_gold>0)
```

```
## [1] 381
```

```r
no_human_zeros=cvtc[ cvtc$CVC_gold>0,]
rer=(no_human_zeros$CVC_n-no_human_zeros$CVC_gold)/no_human_zeros$CVC_gold*100
summary(rer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -100.00  -71.43  -44.83  -20.24  -13.33  800.00
```

```r
aer=(cvtc$CVC_n-cvtc$CVC_gold)
summary(aer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -47.000  -4.000   0.000  -2.013   0.000  30.000
```

```r
png("ctc.png",width=5.5,height=5,units="in",res=300)
plot(CTC_n~CTC_gold,data=cvtc,pch=20,main="Conversational turn counts (near only)",col=alpha("black",.2
abline(lm(CTC_n~CTC_gold,data=cvtc))
abline(lm(CTC_n~CTC_gold,data=cvtc,subset=c(cvtc$CTC_n>0 & cvtc$CTC_gold>0)),lty=2)
dev.off()
```

```
## pdf
##   2
```

```r
cor.test(cvtc$CTC_n,cvtc$CTC_gold)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cvtc$CTC_n and cvtc$CTC_gold
## t = 16.323, df = 701, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.4690471 0.5763683
## sample estimates:
##     cor
## 0.52479
```

```
cor.test(cvtc$CTC_n[cvtc$CTC_n>0 & cvtc$CTC_gold>0],cvtc$CTC_gold[cvtc$CTC_n>0 & cvtc$CTC_gold>0])
```

```
##
##  Pearson's product-moment correlation
##
## data:  cvtc$CTC_n[cvtc$CTC_n > 0 & cvtc$CTC_gold > 0] and cvtc$CTC_gold[cvtc$CTC_n > 0 & cvtc$CTC_gol
## t = 8.9075, df = 293, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3667540 0.5469815
## sample estimates:
##       cor
## 0.4616182
```

```
# RER against human

sum(cvtc$CTC_gold>0)
```

```
## [1] 408
```

```
no_human_zeros=cvtc[ cvtc$CTC_gold>0,]
rer=(no_human_zeros$CTC_n-no_human_zeros$CTC_gold)/no_human_zeros$CTC_gold*100
summary(rer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -100.00 -100.00  -82.95  -65.12  -58.27  300.00
```

```
aer=(cvtc$CTC_n-cvtc$CTC_gold)
summary(aer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -75.000 -14.000  -1.000  -8.222   0.000  25.000
```

AWC

```
read.table("LENA_AWC_rel_v1_June.txt")->awc
colnames(awc)<-c("filename","gold","lena")
gsub("LUC","ROW",awc$filename)->awc$filename

awc$cor=substr(awc$filename,1,3)
awc$cor[grep("[0-9]",awc$cor)]<-"tsi"
awc$cor=factor(awc$cor)

awc$child=substr(awc$filename,1,8)
awc$child[awc$cor=="tsi"]=substr(awc$X[awc$cor=="tsi"],1,3)


merge(awc,age_id,by="child")->awc
```

```r
png("awc.png",width=5.5,height=5,units="in",res=300)

plot(gold~lena,data=awc,pch=20,main="AWC",col=alpha("black",.2))
abline(lm(gold~lena,data=awc))
abline(lm(gold~lena,data=awc,subset=c(awc$gold>0 & awc$lena>0)),lty=2)

dev.off()
```

```
## pdf
##   2
```

```r
cor.test(awc$gold,awc$lena)
```

```
##
##  Pearson's product-moment correlation
##
## data:  awc$gold and awc$lena
## t = 27.817, df = 598, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7139254 0.7839649
## sample estimates:
##       cor
## 0.7510504
```

```r
cor.test(awc$gold[awc$gold>0 & awc$lena>0],awc$lena[awc$gold>0 & awc$lena>0])
```

```
##
##  Pearson's product-moment correlation
##
## data:  awc$gold[awc$gold > 0 & awc$lena > 0] and awc$lena[awc$gold > 0 & awc$lena > 0]
## t = 16.545, df = 307, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6226821 0.7413460
## sample estimates:
##       cor
## 0.6865591
```

```r
# RER against human

sum(awc$gold>0)
```

```
## [1] 361
```

```r
no_human_zeros=awc[awc$gold>0,]
rer=(no_human_zeros$lena-no_human_zeros$gold)/no_human_zeros$gold*100
summary(rer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -100.00  -67.46  -17.78   55.04   40.67 7400.00
```

Variation as a function of corpus and child age

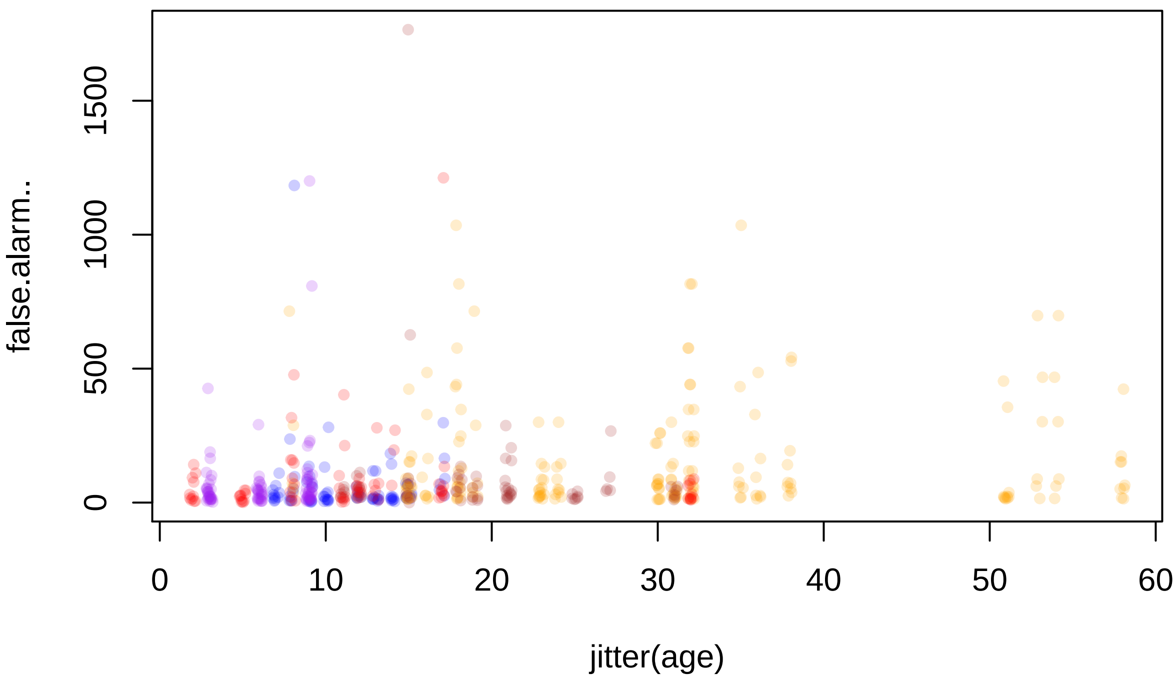
Diarization metrics

```
colfill<-c("blue","brown","red","orange","purple")
names(colfill)<-c("tsi","ROW","BER","SOD","WAR")
for(dv in c("false.alarm..","missed.detection..","confusion..")){
  if(dv=="false.alarm.."){ #note that the other two metrics are undefined when there is no speech
    print("all clips")
    mymodel=lmer(py[,dv]~cor*age+(1|child),data=py)
  print(summary(mymodel))
  print(Anova(mymodel))
  }
    print("only clips with some speech")
    mymodel=lmer(py[,dv]~cor*age+(1|child),data=py,subset=c(total>0))
  print(summary(mymodel))
  print(Anova(mymodel))

 plot(py[,dv]~jitter(age),data=py,col=alpha(colfill[py$cor],.2),pch=ifelse(py$total>0,20,3),ylab=dv)
}
```

```
## [1] "all clips"
## Linear mixed model fit by REML ['lmerMod']
## Formula: py[, dv] ~ cor * age + (1 | child)
##    Data: py
##
## REML criterion at convergence: 6957.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.2206 -0.3855 -0.1951  0.0653  9.0229
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  2674     51.71
##  Residual             30332    174.16
## Number of obs: 532, groups:  child, 52
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   79.566     93.267   0.853
## corROW        66.021    128.792   0.513
## corSOD         1.602    104.184   0.015
## cortsi        49.974    101.341   0.493
## corWAR       -56.618    114.343  -0.495
## age           -1.649      7.892  -0.209
## corROW:age    -1.490      9.013  -0.165
## corSOD:age     1.854      8.454   0.219
## cortsi:age     2.218      7.972   0.278
## corWAR:age    10.242     12.598   0.813
##
## Correlation of Fixed Effects:
##            (Intr) corROW corSOD cortsi corWAR age    crROW: crSOD: crts:g
## corROW     -0.724
```

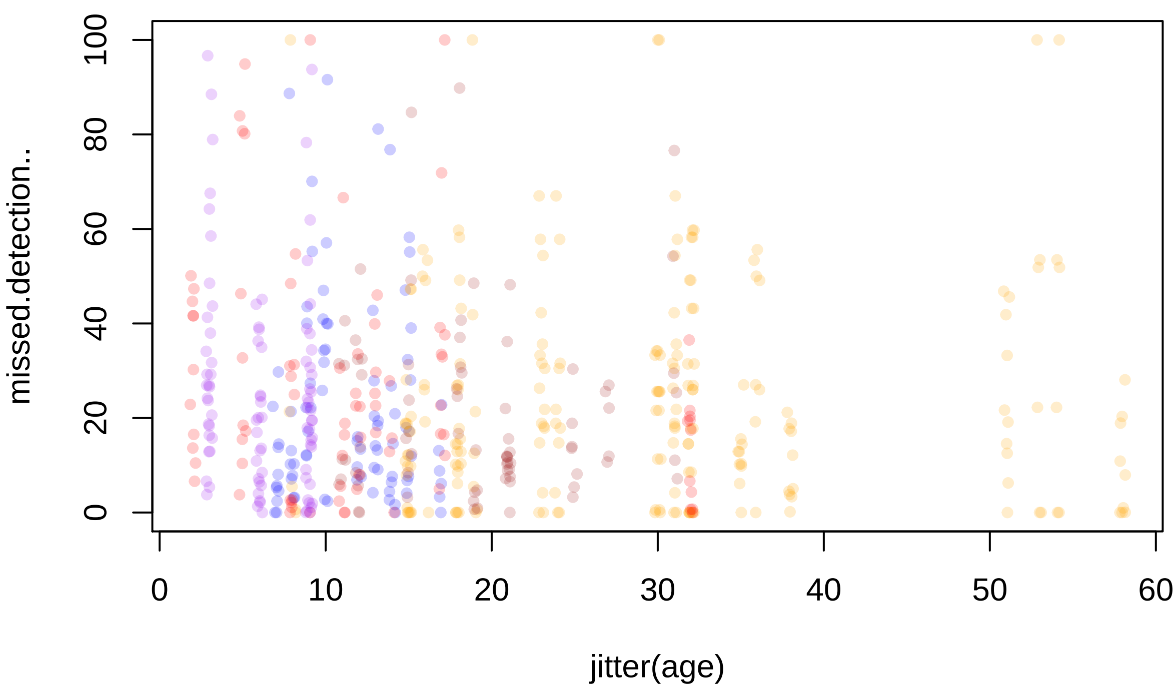
```
## corSOD     -0.895  0.648
## cortsi     -0.920  0.666  0.824
## corWAR     -0.816  0.591  0.730  0.751
## age        -0.963  0.697  0.862  0.886  0.785
## corROW:age  0.843 -0.929 -0.755 -0.776 -0.688 -0.876
## corSOD:age  0.899 -0.651 -0.936 -0.827 -0.733 -0.933  0.817
## cortsi:age  0.953 -0.690 -0.853 -0.925 -0.777 -0.990  0.867  0.924
## corWAR:age  0.603 -0.437 -0.540 -0.555 -0.910 -0.626  0.548  0.585  0.620
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: py[, dv]
##          Chisq Df Pr(>Chisq)
## cor     5.1918  4     0.2682
## age     0.1364  1     0.7119
## cor:age 1.4502  4     0.8354
## [1] "only clips with some speech"
## Linear mixed model fit by REML ['lmerMod']
## Formula: py[, dv] ~ cor * age + (1 | child)
##    Data: py
##  Subset: c(total > 0)
##
## REML criterion at convergence: 6957.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.2206 -0.3855 -0.1951  0.0653  9.0229
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  2674     51.71
##  Residual             30332    174.16
## Number of obs: 532, groups:  child, 52
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   79.566     93.267   0.853
## corROW        66.021    128.792   0.513
## corSOD         1.602    104.184   0.015
## cortsi        49.974    101.341   0.493
## corWAR       -56.618    114.343  -0.495
## age           -1.649      7.892  -0.209
## corROW:age    -1.490      9.013  -0.165
## corSOD:age     1.854      8.454   0.219
## cortsi:age     2.218      7.972   0.278
## corWAR:age    10.242     12.598   0.813
##
## Correlation of Fixed Effects:
##            (Intr) corROW corSOD cortsi corWAR age    crROW: crSOD: crts:g
## corROW     -0.724
## corSOD     -0.895  0.648
## cortsi     -0.920  0.666  0.824
## corWAR     -0.816  0.591  0.730  0.751
## age        -0.963  0.697  0.862  0.886  0.785
## corROW:age  0.843 -0.929 -0.755 -0.776 -0.688 -0.876
```

```
## corSOD:age  0.899 -0.651 -0.936 -0.827 -0.733 -0.933  0.817
## cortsi:age  0.953 -0.690 -0.853 -0.925 -0.777 -0.990  0.867  0.924
## corWAR:age  0.603 -0.437 -0.540 -0.555 -0.910 -0.626  0.548  0.585  0.620
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: py[, dv]
##          Chisq Df Pr(>Chisq)
## cor     5.1918  4     0.2682
## age     0.1364  1     0.7119
## cor:age 1.4502  4     0.8354
```



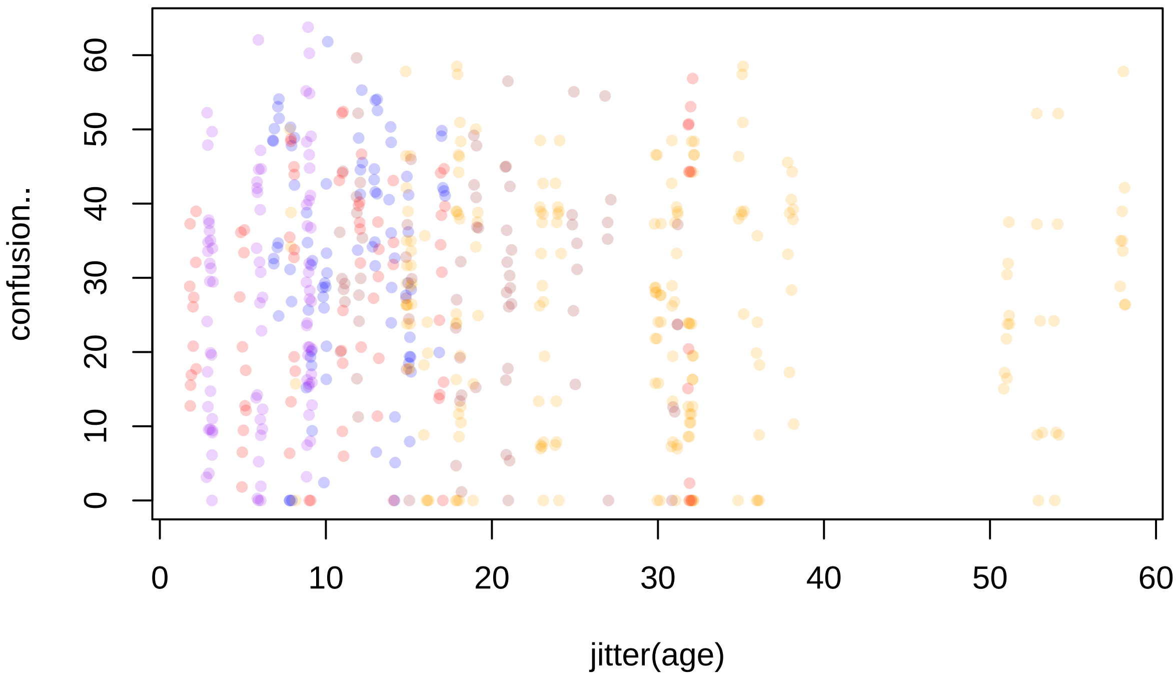

```
## [1] "only clips with some speech"
## Linear mixed model fit by REML ['lmerMod']
## Formula: py[, dv] ~ cor * age + (1 | child)
##    Data: py
##  Subset: c(total > 0)
##
## REML criterion at convergence: 4879.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6792 -0.6397 -0.2053  0.3769  3.6439
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  54.87    7.408
##  Residual             431.41   20.770
## Number of obs: 547, groups:  child, 53
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  22.9599    12.0692   1.902
## corROW       -3.3741    16.6706  -0.202
```

```
## corSOD       11.3383    13.4707   0.842
## cortsi       -0.1746    13.0481  -0.013
## corWAR       12.8055    14.8466   0.863
## age          -0.1816     1.0222  -0.178
## corROW:age    0.2558     1.1663   0.219
## corSOD:age   -0.4507     1.0968  -0.411
## cortsi:age    0.2144     1.0315   0.208
## corWAR:age   -1.3426     1.6397  -0.819
##
## Correlation of Fixed Effects:
##            (Intr) corROW corSOD cortsi corWAR age    crROW: crSOD: crts:g
## corROW     -0.724
## corSOD     -0.896  0.649
## cortsi     -0.925  0.670  0.829
## corWAR     -0.813  0.589  0.728  0.752
## age        -0.962  0.697  0.862  0.890  0.782
## corROW:age  0.843 -0.928 -0.756 -0.780 -0.686 -0.876
## corSOD:age  0.897 -0.649 -0.937 -0.830 -0.729 -0.932  0.817
## cortsi:age  0.954 -0.690 -0.854 -0.926 -0.775 -0.991  0.868  0.924
## corWAR:age  0.600 -0.434 -0.537 -0.555 -0.910 -0.623  0.546  0.581  0.618
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: py[, dv]
##          Chisq Df Pr(>Chisq)
## cor     2.4686  4     0.6503
## age     0.1573  1     0.6917
## cor:age 3.8907  4     0.4210
```




```
## [1] "only clips with some speech"
## Linear mixed model fit by REML ['lmerMod']
## Formula: py[, dv] ~ cor * age + (1 | child)
##    Data: py
##  Subset: c(total > 0)
```

```
##
## REML criterion at convergence: 4483.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6340 -0.6680  0.1025  0.6783  2.2388
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  48.76    6.983
##  Residual             199.66   14.130
## Number of obs: 547, groups:  child, 53
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)  3.374e+01  9.862e+00   3.422
## corROW       5.386e-01  1.364e+01   0.039
## corSOD      -1.051e+01  1.106e+01  -0.950
## cortsi      -6.757e+00  1.057e+01  -0.639
## corWAR      -1.285e+01  1.221e+01  -1.052
## age         -1.048e-02  8.368e-01  -0.013
## corROW:age  -2.638e-01  9.537e-01  -0.277
## corSOD:age   2.893e-01  9.013e-01   0.321
## cortsi:age   4.415e-04  8.428e-01   0.001
## corWAR:age   8.938e-01  1.356e+00   0.659
##
## Correlation of Fixed Effects:
##            (Intr) corROW corSOD cortsi corWAR age    crROW: crSOD: crts:g
## corROW     -0.723
## corSOD     -0.892  0.645
## cortsi     -0.933  0.675  0.832
## corWAR     -0.807  0.584  0.720  0.753
## age        -0.961  0.695  0.857  0.897  0.776
## corROW:age  0.843 -0.927 -0.752 -0.787 -0.681 -0.877
## corSOD:age  0.892 -0.645 -0.936 -0.833 -0.721 -0.928  0.815
## cortsi:age  0.954 -0.690 -0.851 -0.925 -0.771 -0.993  0.871  0.922
## corWAR:age  0.593 -0.429 -0.529 -0.554 -0.910 -0.617  0.542  0.573  0.613
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: py[, dv]
##          Chisq Df Pr(>Chisq)
## cor     5.1570  4     0.2716
## age     0.0076  1     0.9307
## cor:age 1.7413  4     0.7832
```



cvc

```
cvtc$cor=substr(cvtc$filename,1,3)
cvtc$cor[substr(cvtc$filename,1,1)=="C"]<-"tsi"
cvtc$cor=factor(cvtc$cor)

cvtc$child=substr(cvtc$filename,1,8)
cvtc$child[cvtc$cor=="tsi"]<-substr(cvtc$filename[cvtc$cor=="tsi"],1,3)

merge(cvtc,age_id,by="child")->cvtc

mymodel<-lmer(CVC_gold~CVC_n*age*cor + (1|child), data=cvtc)
summary(mymodel)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: CVC_gold ~ CVC_n * age * cor + (1 | child)
##    Data: cvtc
##
## REML criterion at convergence: 5647.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.4030 -0.3165 -0.1642  0.2802  5.9030
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  0.92    0.9592
##  Residual             58.00    7.6158
## Number of obs: 812, groups:  child, 53
##
## Fixed effects:
```

```
##                    Estimate Std. Error t value
## (Intercept)         3.98586    2.83252   1.407
## CVC_n               0.01699    0.34482   0.049
## age                -0.16751    0.24264  -0.690
## corROW             -0.02193    4.04009  -0.005
## corSOD             -6.51905    3.26492  -1.997
## cortsi             -2.56909    3.43513  -0.748
## corWAR             -2.91222    3.63321  -0.802
## CVC_n:age           0.07982    0.02957   2.700
## CVC_n:corROW        0.88930    0.43087   2.064
## CVC_n:corSOD        1.12139    0.38318   2.927
## CVC_n:cortsi        0.88809    0.41826   2.123
## CVC_n:corWAR        1.67403    0.41069   4.076
## age:corROW          0.06356    0.27748   0.229
## age:corSOD          0.50648    0.26722   1.895
## age:cortsi          0.16715    0.24988   0.669
## age:corWAR          0.28685    0.41881   0.685
## CVC_n:age:corROW -0.06957    0.03176  -2.191
## CVC_n:age:corSOD -0.09262    0.03077  -3.010
## CVC_n:age:cortsi -0.08562    0.03051  -2.806
## CVC_n:age:corWAR -0.14416    0.04404  -3.274
```

```
##
## Correlation matrix not shown by default, as p = 20 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```r
Anova(mymodel)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CVC_gold
##                  Chisq Df Pr(>Chisq)
## CVC_n         762.1730  1  < 2.2e-16 ***
## age             0.1378  1   0.710436
## cor            11.1844  4   0.024568 *
## CVC_n:age       0.7186  1   0.396610
## CVC_n:cor      16.2180  4   0.002740 **
## age:cor         9.6119  4   0.047499 *
## CVC_n:age:cor  14.0914  4   0.007009 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#there is a 3-way interaction between age, corpus, and the predictive value of LENA's counts with respe
# to investigate this we fit the same reg within corpus

for(thiscor in levels(cvtc$cor)){
  print(thiscor)
 mymodel<-lmer(CVC_gold~CVC_n*age + (1|child), data=cvtc,subset=c(cor==thiscor))
print(summary(mymodel))
print(Anova(mymodel))
}
```

```
## [1] "BER"
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: CVC_gold ~ CVC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1000
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7770 -0.4192 -0.2206  0.3487  5.1915
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  0.00    0.000
##  Residual             44.26    6.653
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  3.93811    2.28819   1.721
## CVC_n        0.02817    0.29869   0.094
## age         -0.16420    0.19614  -0.837
## CVC_n:age    0.07904    0.02565   3.082
##
## Correlation of Fixed Effects:
##           (Intr) CVC_n  age
## CVC_n     -0.557
## age       -0.957  0.534
## CVC_n:age  0.533 -0.958 -0.559
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CVC_gold
##              Chisq Df Pr(>Chisq)
## CVC_n     113.3771  1  < 2.2e-16 ***
## age         1.1418  1   0.285278
## CVC_n:age   9.4971  1   0.002058 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "ROW"
## Linear mixed model fit by REML ['lmerMod']
## Formula: CVC_gold ~ CVC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1100.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7159 -0.2756 -0.1524  0.1559  3.5413
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  2.479   1.575
##  Residual             84.446   9.189
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  4.06568    3.66598   1.109
## CVC_n        0.88970    0.31243   2.848
## age         -0.11036    0.17175  -0.643
## CVC_n:age    0.01128    0.01403   0.804
##
## Correlation of Fixed Effects:
##           (Intr) CVC_n  age
## CVC_n     -0.539
## age       -0.956  0.492
## CVC_n:age  0.515 -0.961 -0.511
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CVC_gold
##              Chisq Df Pr(>Chisq)
## CVC_n     172.0008  1     <2e-16 ***
## age         0.0726  1     0.7876
## CVC_n:age   0.6468  1     0.4213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SOD"
## Linear mixed model fit by REML ['lmerMod']
## Formula: CVC_gold ~ CVC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1109.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9047 -0.4847 -0.1404  0.3426  4.6474
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  7.038   2.653
##  Residual             86.926   9.323
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -2.56105    2.43857  -1.050
## CVC_n        1.14124    0.20555   5.552
## age          0.34207    0.16806   2.035
## CVC_n:age   -0.01315    0.01048  -1.254
##
## Correlation of Fixed Effects:
##           (Intr) CVC_n  age
## CVC_n     -0.478
```

```
## age       -0.850  0.353
## CVC_n:age  0.467 -0.836 -0.488
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CVC_gold
##             Chisq Df Pr(>Chisq)
## CVC_n     67.3116  1  2.318e-16 ***
## age        2.6600  1     0.1029
## CVC_n:age  1.5726  1     0.2098
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "tsi"
## Linear mixed model fit by REML ['lmerMod']
## Formula: CVC_gold ~ CVC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1136.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5088 -0.4912 -0.2246  0.4608  2.8783
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  0.6722  0.8199
##  Residual             11.1244  3.3353
## Number of obs: 212, groups:  child, 13
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  1.142313   0.931286   1.227
## CVC_n        0.912781   0.104957   8.697
## age          0.007812   0.027856   0.280
## CVC_n:age   -0.005878   0.003334  -1.763
##
## Correlation of Fixed Effects:
##           (Intr) CVC_n  age
## CVC_n     -0.643
## age       -0.904  0.610
## CVC_n:age  0.588 -0.889 -0.667
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CVC_gold
##              Chisq Df Pr(>Chisq)
## CVC_n     242.7125  1    < 2e-16 ***
## age         1.4442  1    0.22946
## CVC_n:age   3.1094  1    0.07784 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "WAR"
## boundary (singular) fit: see ?isSingular
## Linear mixed model fit by REML ['lmerMod']
```

```
## Formula: CVC_gold ~ CVC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1083.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6257 -0.2436 -0.1614  0.3904  3.2265
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  0.00    0.000
##  Residual             78.71    8.872
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  0.98265    2.46135   0.399
## CVC_n        1.70718    0.25724   6.636
## age          0.13091    0.37010   0.354
## CVC_n:age   -0.06636    0.03764  -1.763
##
## Correlation of Fixed Effects:
##           (Intr) CVC_n  age
## CVC_n     -0.599
## age       -0.932  0.565
## CVC_n:age  0.577 -0.944 -0.617
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CVC_gold
##              Chisq Df Pr(>Chisq)
## CVC_n     225.5910  1     < 2e-16 ***
## age         0.8693  1     0.35115
## CVC_n:age   3.1080  1     0.07791 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ctc

```r
mymodel<-lmer(CTC_gold~CTC_n*age*cor + (1|child), data=cvtc)
summary(mymodel)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: CTC_gold ~ CTC_n * age * cor + (1 | child)
##    Data: cvtc
##
## REML criterion at convergence: 6450.2
##
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -4.1771 -0.5467 -0.2257  0.3398  5.4382
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)   6.057   2.461
##  Residual             158.890  12.605
## Number of obs: 812, groups:  child, 53
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)          5.58898    4.95235   1.129
## CTC_n                0.09868    0.76419   0.129
## age                  0.11761    0.42306   0.278
## corROW               9.23603    6.93112   1.333
## corSOD               3.64570    5.62487   0.648
## cortsi              -1.44350    5.86742  -0.246
## corWAR               1.03059    6.29300   0.164
## CTC_n:age            0.08031    0.05789   1.387
## CTC_n:corROW         1.53701    0.93637   1.641
## CTC_n:corSOD         1.02516    0.81733   1.254
## CTC_n:cortsi         0.45687    1.02774   0.445
## CTC_n:corWAR         1.44328    0.85522   1.688
## age:corROW          -0.46950    0.48191  -0.974
## age:corSOD          -0.03076    0.46044  -0.067
## age:cortsi          -0.13429    0.43348  -0.310
## age:corWAR          -0.19925    0.71553  -0.278
## CTC_n:age:corROW -0.08133    0.06427  -1.266
## CTC_n:age:corSOD -0.09429    0.06397  -1.474
## CTC_n:age:cortsi -0.07984    0.06295  -1.268
## CTC_n:age:corWAR -0.05478    0.08743  -0.627
```

```
##
## Correlation matrix not shown by default, as p = 20 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
Anova(mymodel)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CTC_gold
##                    Chisq Df Pr(>Chisq)
## CTC_n           284.7464  1  < 2.2e-16 ***
## age               0.1716  1  0.6787289
## cor              20.0269  4  0.0004933 ***
## CTC_n:age         0.0230  1  0.8793880
## CTC_n:cor        21.4899  4  0.0002532 ***
## age:cor           3.5874  4  0.4647073
## CTC_n:age:cor     2.3189  4  0.6773262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
for(thiscor in levels(cvtc$cor)){
  print(thiscor)
 mymodel<-lmer(CTC_gold~CTC_n*age + (1|child), data=cvtc,subset=c(cor==thiscor))
```

```r
print(summary(mymodel))
print(Anova(mymodel))
}
```

```
## [1] "BER"
## Linear mixed model fit by REML ['lmerMod']
## Formula: CTC_gold ~ CTC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1172.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0342 -0.5950 -0.5257  0.6232  3.8643
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)   0.1447  0.3805
##  Residual             144.1107 12.0046
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  5.49588    3.94735   1.392
## CTC_n        0.13685    0.71438   0.192
## age          0.12364    0.33744   0.366
## CTC_n:age    0.07774    0.05432   1.431
##
## Correlation of Fixed Effects:
##           (Intr) CTC_n  age
## CTC_n     -0.476
## age       -0.957  0.425
## CTC_n:age  0.483 -0.975 -0.465
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CTC_gold
##             Chisq Df Pr(>Chisq)
## CTC_n     51.8015  1  6.141e-13 ***
## age        1.3575  1     0.2440
## CTC_n:age  2.0484  1     0.1524
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "ROW"
## Linear mixed model fit by REML ['lmerMod']
## Formula: CTC_gold ~ CTC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1279
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0520 -0.5404 -0.3385  0.3169  4.0379
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)   1.186   1.089
##  Residual             293.279  17.125
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 14.578019   5.601367   2.603
## CTC_n        1.728860   0.719442   2.403
## age         -0.333407   0.265943  -1.254
## CTC_n:age   -0.007373   0.036992  -0.199
##
## Correlation of Fixed Effects:
##           (Intr) CTC_n  age
## CTC_n     -0.463
## age       -0.953  0.456
## CTC_n:age  0.416 -0.957 -0.455
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CTC_gold
##             Chisq Df Pr(>Chisq)
## CTC_n     58.7406  1  1.799e-14 ***
## age        2.2781  1     0.1312
## CTC_n:age  0.0397  1     0.8420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "SOD"
## Linear mixed model fit by REML ['lmerMod']
## Formula: CTC_gold ~ CTC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1275.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9746 -0.6302 -0.3613  0.5231  3.6284
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  34.45    5.869
##  Residual             271.04   16.463
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  9.48247    4.54214   2.088
## CTC_n        1.07335    0.38268   2.805
## age          0.07923    0.31058   0.255
## CTC_n:age   -0.01344    0.03601  -0.373
##
## Correlation of Fixed Effects:
```

```
##           (Intr) CTC_n  age
## CTC_n     -0.320
## age       -0.847  0.301
## CTC_n:age  0.221 -0.823 -0.300
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CTC_gold
##             Chisq Df Pr(>Chisq)
## CTC_n     19.3252  1  1.102e-05 ***
## age        0.0226  1     0.8806
## CTC_n:age  0.1393  1     0.7089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "tsi"

## boundary (singular) fit: see ?isSingular

## Linear mixed model fit by REML ['lmerMod']
## Formula: CTC_gold ~ CTC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1212
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0805 -0.7665 -0.1810  0.6096  2.5508
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)  0.00    0.000
##  Residual             16.69    4.086
## Number of obs: 212, groups:  child, 13
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)  4.1958692  0.9301850   4.511
## CTC_n        0.5734130  0.2189954   2.618
## age         -0.0201870  0.0288778  -0.699
## CTC_n:age    0.0001668  0.0078979   0.021
##
## Correlation of Fixed Effects:
##           (Intr) CTC_n  age
## CTC_n     -0.626
## age       -0.929  0.624
## CTC_n:age  0.550 -0.937 -0.624
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CTC_gold
##             Chisq Df Pr(>Chisq)
## CTC_n     56.8296  1  4.752e-14 ***
## age        0.7701  1     0.3802
```

```
## CTC_n:age  0.0004  1     0.9832
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "WAR"
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: CTC_gold ~ CTC_n * age + (1 | child)
##    Data: cvtc
##  Subset: c(cor == thiscor)
##
## REML criterion at convergence: 1153
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3820 -0.5437 -0.4877  0.4938  2.7507
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  child    (Intercept)   0.0     0.00
##  Residual             127.2    11.28
## Number of obs: 150, groups:  child, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  6.39580    2.90448   2.202
## CTC_n        1.59610    0.33972   4.698
## age         -0.04388    0.43292  -0.101
## CTC_n:age    0.01588    0.05796   0.274
##
## Correlation of Fixed Effects:
##           (Intr) CTC_n  age
## CTC_n     -0.503
## age       -0.927  0.500
## CTC_n:age  0.434 -0.913 -0.516
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: CTC_gold
##              Chisq Df Pr(>Chisq)
## CTC_n     146.4837  1     <2e-16 ***
## age         0.0022  1     0.9627
## CTC_n:age   0.0751  1     0.7841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```