

acoustic_analysis

```
cur_exp = "exp2"
features = c("duration", "meanIntensity", "meanpit")
# info = c('participant', 'verb', 'condition', 'word', 'word_num')
info = c('participant', 'item_id', 'location_condition', 'word', 'word_num')
bRemove_outliers = 0
```

This the analysis for exp2. The parameters of all exps can be seen at https://github.com/Xinzh-Fang/prosody_study_exp/blob/master/tAll_exps.csv.

The trial-by-trial design of this exp can be seen at https://github.com/Xinzh-Fang/prosody_study_exp/blob/master/exp2/tAll_trials.csv

```
tAll_trials = read.csv(file.path '..', cur_exp, 'tAll_trials.csv'))

df0 = read.csv(paste0('measure_', cur_exp, '.csv'), header = T)
df0$location_condition = NA
df0$item_id = NA

for (iR in 1:nrow(df0)){
  df0$location_condition[iR] = as.character(tAll_trials[tAll_trials$trial_id == df0$trialId[iR], 'location_condition'])
  df0$item_id[iR] = as.character(tAll_trials[tAll_trials$trial_id == df0$trialId[iR], 'filler_or_item_id'])
}

df1 = df0[startsWith(df0$item_id, "item"),]

# df0 = read.csv("measure_nonrhyming_84total_60No_24Yes_20181210.csv", header = T)
# df0 = transform(df0, trialId=as.numeric(trialId))
# sort(df0$trialId, decreasing = FALSE)
# colnamesC(df1)

df2 = df1[df1$word != 'sp',]
# code for word_num
df2 <- df2 %>%
  dplyr::group_by(participant, trialId) %>%
  # dplyr::group_by(participant, question, trialId) %>%
  dplyr::mutate(word_num=1:dplyr::n()) %>%
  dplyr::select(c(info, features))

## Adding missing grouping variables: `trialId`
```

2 workers and 60 trials are included in this analysis.

```
# write.csv(df2, 'newdf.csv')
# code for getting Nth instance of question
# nthdf <- df1 %>%
#   group_by(participant, Verb, question, condition, word_num) %>%
#   mutate(Appearance=1:n())
# write.csv(nthdf, 'nthdf.csv')

# subsetting it to relevant Nth appearance
```

```

# workingdf <- nthdf %>%
#   filter (Appearance == 2)
#
# write.csv(workingdf, 'workingdf2.csv')

normalize_data = function(df, remove_outliers){
  for(col_name in features){
    if(!is.numeric(df[[col_name]])){
      df[[col_name]] = as.numeric(df[[col_name]])
    }
    df[[col_name]] = scale(df[[col_name]])
    # there is surge of na after the first colling of the above line. tested by print(sum(is.na(df_Agent)))
    # print(sum(is.na(df_Agent)))
  }
  for(col_name in features){

    if(remove_outliers){
      df = df[df[[col_name]]>-2 & df[[col_name]]<2,]
      # print(sum(is.na(df_Agent)))
    }
  }
  return(df)
}

# process_data = function(file_name){
process_data = function(df){
  # df <- read.csv(file_name, header = TRUE, fileEncoding="UTF-8", na.strings=c("", "NA", "--undefined--"))
  # df <- na.omit(df)

  #df = df[df$wordlabel != 'sp']
  # df$verb = as.factor(df$verb)

  # df_Agent = df[(df$location_condition=='Agent' | df$location_condition=='Verb') & df$word_num=='3',]
  #
  # df_Verb = df[(df$location_condition=='Verb' | df$location_condition=='Patient') & df$word_num=='5',]
  #
  # df_Patient = df[(df$location_condition=='Patient' | df$location_condition=='Agent') & df$word_num=='1',]

  df_Agent = df[(df$location_condition=='Agent' | df$location_condition=='Control') & df$word_num=='2',]
  # df_Agent inheri row hum from df

  df_Verb = df[(df$location_condition=='Verb' | df$location_condition=='Control') & df$word_num=='4',]

  df_Patient = df[(df$location_condition=='Patient' | df$location_condition=='Control') & df$word_num=='1',]

  # print(sum(is.na(df_Agent)))

  # relevant_columns = c('participant', 'verb', 'condition', 'duration', 'meanIntensity', 'meanpit')
  # df_Agent = df_Agent[relevant_columns]
  # df_Verb = df_Verb[relevant_columns]

```

```

# df_Patient = df_Patient[relevant_columns]
print(sum(is.na(df[df$word != 'sp',])))
# df1[(df1$meanpit == '--undefined--') && (df1$word != 'sp'),]
# it seems that the only undefined is meanpitch for sp

# print(df_Verb)

df_Verb = normalize_data(df_Verb, bRemove_outliers)
df_Agent = normalize_data(df_Agent, bRemove_outliers)
df_Patient = normalize_data(df_Patient, bRemove_outliers)
# print(sum(is.na(df_Agent)))

# return(list(df_Agent_duration, df_Agent_meanIntensity, df_Agent_meanpit, df_Patient_duration, df_Patient_meanIntensity))
return(list(df_Verb, df_Agent, df_Patient))
}

#
# file_name = 'newdf.csv'

# c(df_Agent_duration, df_Agent_meanIntensity, df_Agent_meanpit, df_Patient_duration, df_Patient_meanIntensity)
# c(df_Verb, df_Agent, df_Patient) %<-% process_data(file_name)
c(df_Verb, df_Agent, df_Patient) %<-% process_data(df2)

## [1] 0

combine_datasets = function(Agent,Verb,Patient){
  Agent$condition = mapvalues(Agent$location_condition,c('Agent'),c('contrast'))
  Verb$condition = mapvalues(Verb$location_condition,c('Verb'),c('contrast'))
  Patient$condition = mapvalues(Patient$location_condition,c('Patient'),c('contrast'))

  Agent$Location = 'Agent'
  Verb$Location = 'Verb'
  Patient$Location = "Patient"

  return(rbind(Agent,Verb,Patient))
}

summarize_data = function(d, feature){
  # http://www.cookbook-r.com/Graphs/Plotting\_means\_and\_error\_bars\_\(ggplot2\)/
  return(summarySE(d,measurevar=feature ,groupvars=c('Location','condition'))))
}

plot_data = function(d,feature, title){
  print(ggplot(d, aes(x=Location, y=get(feature), fill=condition)) +
    geom_bar(position=position_dodge(), stat="identity") +
    geom_errorbar(aes(ymin=get(feature)-ci, ymax=get(feature)+ci),
      width=.2,
      position=position_dodge(.9))+
    xlab("Location") +
    ylab(paste0("normalized ", feature)) +
    scale_fill_hue(name="location_condition",

```

```

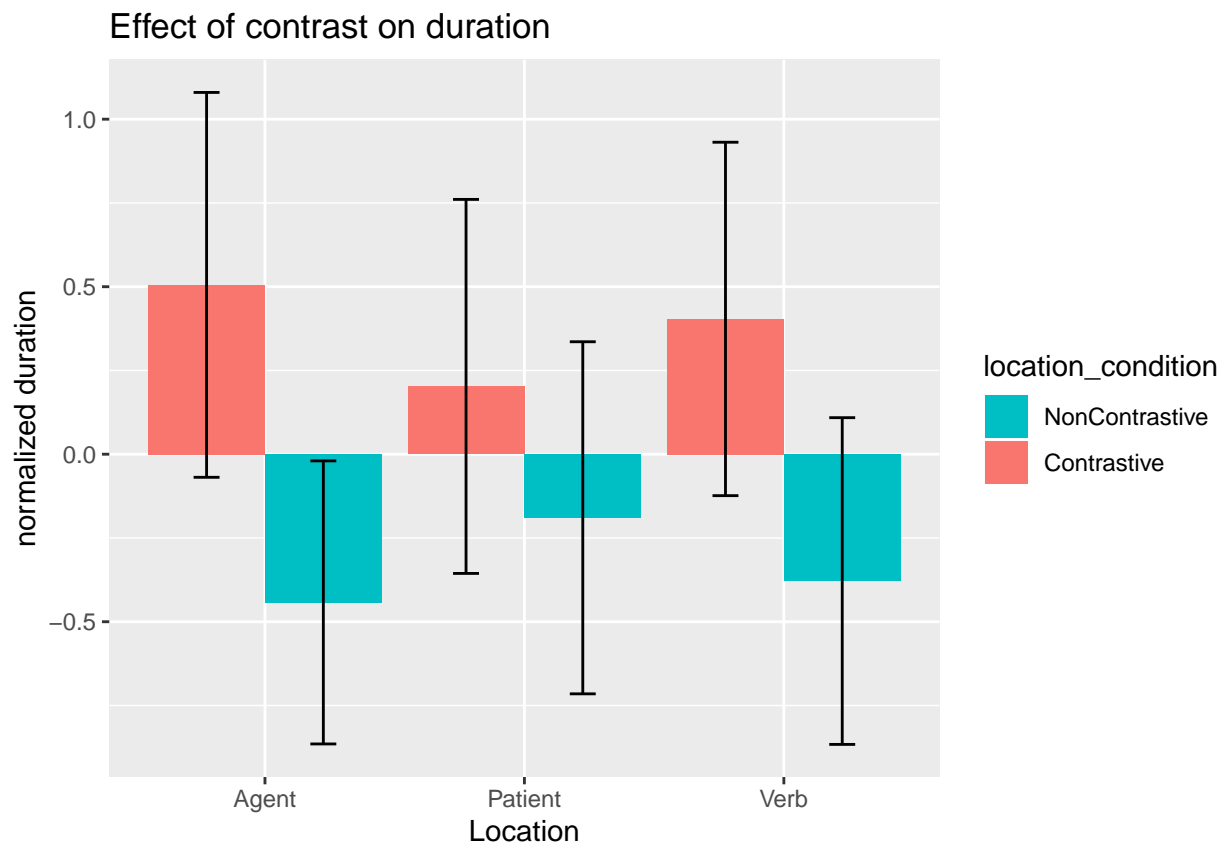
        breaks=c("Control", "contrast"),
        labels=c("NonContrastive", "Contrastive")) +
        ggtitle(title))
}
for (iF in features){
  print(iF)

  combined_dataset = combine_datasets(df_Agent, df_Verb, df_Patient)
  summarized_dataset= summarize_data(combined_dataset, iF)

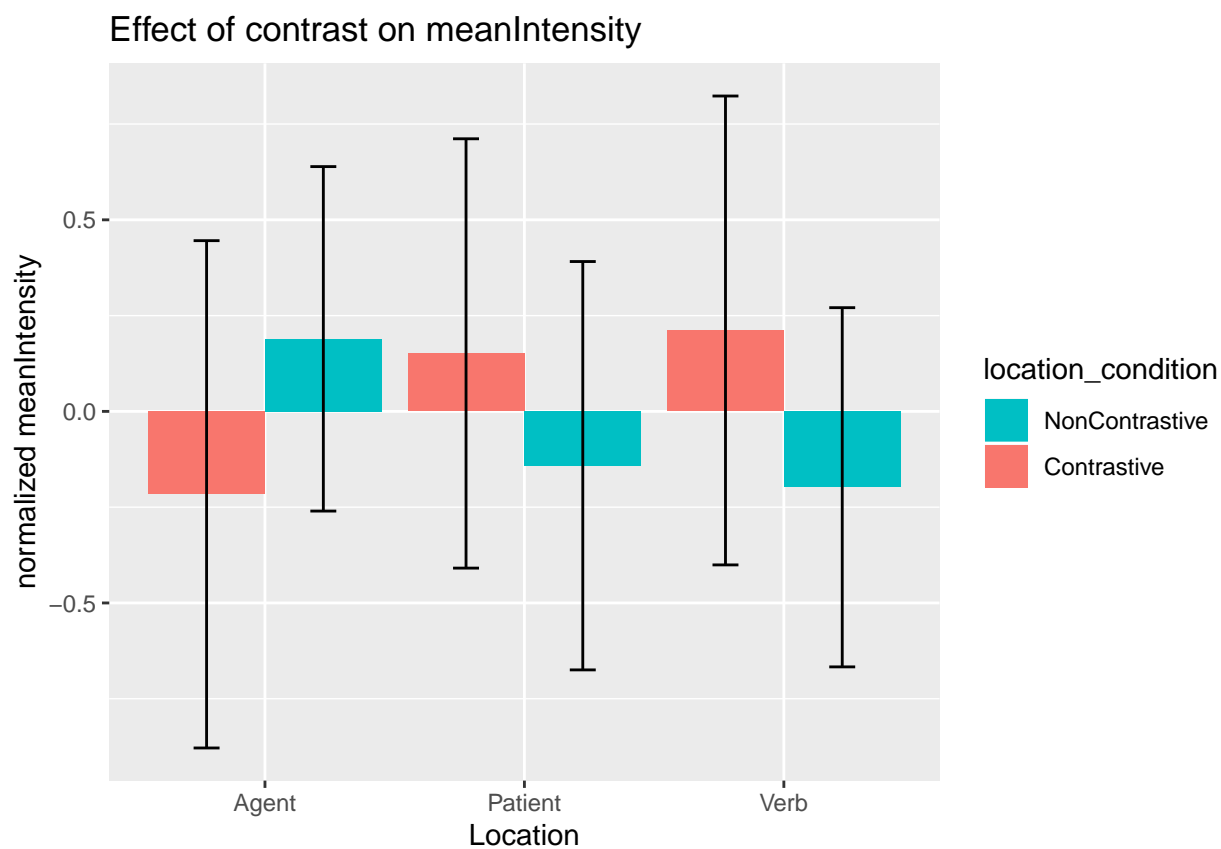
  plot_data(summarized_dataset,iF, title= paste0('Effect of contrast on ', iF))
}

```

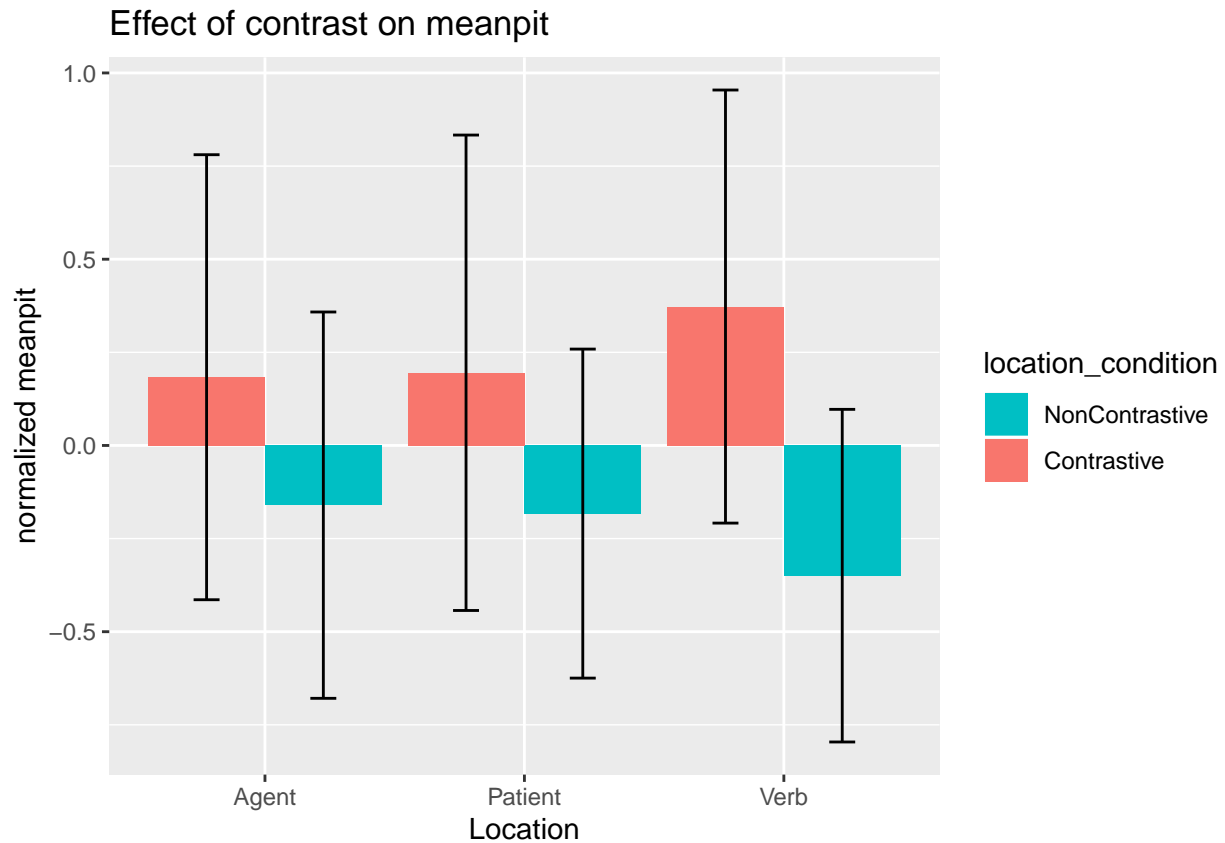
```
## [1] "duration"
```



```
## [1] "meanIntensity"
```



```
## [1] "meanpit"
```



```
run_regression = function(location, observation){
  cat(" \n###", observation, "of", location, " \n")
  r = lmer(get(observation) ~ location_condition + (1 + location_condition|participant) + (1 + location.
  # r = lmer(get(observation) ~ location_condition + (1 + location_condition | item_id), data=get(past
  print(summary(r))
  summary(r)
  cat(" \n")
}

for (iF in features){
  run_regression("Agent", iF)

  run_regression("Patient", iF)

  run_regression("Verb", iF)

}

##
## ### duration of Agent
## boundary (singular) fit: see ?isSingular
```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##   participant) + (1 + location_condition | item_id)
##   Data: get(paste0("df_", location))
##
## REML criterion at convergence: 78.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.20764 -0.60822 -0.05165  0.43369  2.07139
##
## Random effects:
##   Groups      Name                Variance Std.Dev.  Corr
##   item_id     (Intercept)          0.000e+00 0.000e+00
##              location_conditionControl 1.555e-12 1.247e-06  NaN
##   participant (Intercept)          1.016e-01 3.188e-01
##              location_conditionControl 8.372e-02 2.893e-01 -1.00
##   Residual                        7.705e-01 8.778e-01
## Number of obs: 30, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      0.5055     0.3253   1.554
## location_conditionControl -0.9478     0.3809  -2.489
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnC -0.816
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
## ### duration of Patient
## boundary (singular) fit: see ?isSingular
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##   participant) + (1 + location_condition | item_id)
##   Data: get(paste0("df_", location))
##
## REML criterion at convergence: 71.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.11689 -0.40247 -0.04134  0.33449  2.40993
##
## Random effects:
##   Groups      Name                Variance Std.Dev.  Corr
##   item_id     (Intercept)          0.1658   0.4072
##              location_conditionPatient 0.1923   0.4385   1.00
##   participant (Intercept)          0.6507   0.8067

```

```

##           location_conditionPatient 0.1790    0.4231    -1.00
## Residual                        0.4117    0.6416
## Number of obs: 31, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      -0.1898     0.6265  -0.303
## location_conditionPatient  0.3868     0.4371   0.885
##
## Correlation of Fixed Effects:
##           (Intr)
## lctn_cndtnP -0.554
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
## ### duration of Verb
## boundary (singular) fit: see ?isSingular
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##   participant) + (1 + location_condition | item_id)
##   Data: get(paste0("df_", location))
##
## REML criterion at convergence: 66
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.81412 -0.52181  0.08692  0.68359  1.79678
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   item_id     (Intercept)          0.39260  0.6266
##              location_conditionVerb 0.07083  0.2661   1.00
##   participant (Intercept)          0.25394  0.5039
##              location_conditionVerb 0.24614  0.4961  -1.00
## Residual                        0.33482  0.5786
## Number of obs: 31, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      -0.3785     0.4960  -0.763
## location_conditionVerb  0.7286     0.4292   1.698
##
## Correlation of Fixed Effects:
##           (Intr)
## lctn_cndtnV -0.490
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##

```



```

## ### meanIntensity of Agent

## boundary (singular) fit: see ?isSingular

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##   participant) + (1 + location_condition | item_id)
##   Data: get(paste0("df_", location))
##
## REML criterion at convergence: 55.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5732 -0.6673  0.2096  0.4787  1.4181
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   item_id      (Intercept)          0.50014  0.7072
##              location_conditionControl 0.06376  0.2525  -0.95
##   participant (Intercept)          0.93903  0.9690
##              location_conditionControl 0.02372  0.1540  -1.00
##   Residual                        0.23324  0.4829
## Number of obs: 30, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      -0.2543    0.7819  -0.325
## location_conditionControl  0.4438    0.2434   1.823
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnC -0.704
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
## ### meanIntensity of Patient

## boundary (singular) fit: see ?isSingular

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##   participant) + (1 + location_condition | item_id)
##   Data: get(paste0("df_", location))
##
## REML criterion at convergence: 79.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.45209 -0.44317  0.08243  0.63055  1.61660
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr

```

```

## item_id      (Intercept)                0.4844630 0.69603
##              location_conditionPatient 0.0609440 0.24687 -0.63
## participant  (Intercept)                0.1922716 0.43849
##              location_conditionPatient 0.0002204 0.01485 -1.00
## Residual                    0.5779810 0.76025
## Number of obs: 31, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      -0.1418    0.5034  -0.282
## location_conditionPatient  0.2727    0.3005   0.908
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnP -0.439
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
## ### meanIntensity of Verb

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
## participant) + (1 + location_condition | item_id)
## Data: get(paste0("df_", location))
##
## REML criterion at convergence: 79.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.46772 -0.68973 -0.05721  0.74403  2.23234
##
## Random effects:
## Groups      Name                Variance Std.Dev. Corr
## item_id     (Intercept)          0.0001062 0.01030
##              location_conditionVerb 0.6403250 0.80020  -0.23
## participant (Intercept)          0.2073765 0.45539
##              location_conditionVerb 0.0097388 0.09869  1.00
## Residual                    0.5921822 0.76953
## Number of obs: 31, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      -0.1981    0.3751  -0.528
## location_conditionVerb  0.3894    0.4920   0.791
##

```

```

## Correlation of Fixed Effects:
##      (Intr)
## lctn_cndtnV -0.081
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
##
##
##
## ### meanpit of Agent

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00244259
## (tol = 0.002, component 1)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##   participant) + (1 + location_condition | item_id)
##   Data: get(paste0("df_", location))
##
## REML criterion at convergence: 36.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.77370 -0.42831  0.06722  0.59902  1.89019
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   item_id      (Intercept)          0.0586221 0.24212
##               location_conditionControl 0.0004221 0.02054  1.00
##   participant (Intercept)          1.9376774 1.39200
##               location_conditionControl 0.0510313 0.22590 -1.00
##   Residual                        0.1234128 0.35130
## Number of obs: 30, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)          0.1790    0.9962   0.180
## location_conditionControl -0.3393    0.2055  -1.651
##
## Correlation of Fixed Effects:
##      (Intr)
## lctn_cndtnC -0.806
## convergence code: 0
## Model failed to converge with max|grad| = 0.00244259 (tol = 0.002, component 1)
##
##
##
## ### meanpit of Patient

## boundary (singular) fit: see ?isSingular
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |

```

```

##      participant) + (1 + location_condition | item_id)
##      Data: get(paste0("df_", location))
##
## REML criterion at convergence: 81.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.0641 -0.5279 -0.2149  0.3049  2.2172
##
## Random effects:
##      Groups      Name                Variance Std.Dev.  Corr
## item_id      (Intercept)            0.000e+00 0.000e+00
##              location_conditionPatient 4.773e-10 2.185e-05  NaN
## participant (Intercept)            1.354e-01 3.679e-01
##              location_conditionPatient 2.914e-01 5.398e-01  1.00
## Residual                        7.468e-01 8.642e-01
## Number of obs: 31, groups:  item_id, 4; participant, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      -0.1829    0.3382  -0.541
## location_conditionPatient  0.4188    0.4923   0.851
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnP 0.316
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
## ### meanpit of Verb
##
## boundary (singular) fit: see ?isSingular
##
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##      participant) + (1 + location_condition | item_id)
##      Data: get(paste0("df_", location))
##
## REML criterion at convergence: 62.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.7255 -0.5180  0.1950  0.7267  1.2101
##
## Random effects:
##      Groups      Name                Variance Std.Dev.  Corr
## item_id      (Intercept)            0.09342  0.3056
##              location_conditionVerb 0.03081  0.1755  -1.00
## participant (Intercept)            0.16954  0.4118
##              location_conditionVerb 0.89703  0.9471   1.00
## Residual                        0.33801  0.5814
## Number of obs: 31, groups:  item_id, 4; participant, 2

```

```
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    -0.3496    0.3595  -0.972
## location_conditionVerb  0.7952    0.7071   1.125
##
## Correlation of Fixed Effects:
##           (Intr)
## lctn_cndtnV 0.631
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
# r = lmer(get(observation) ~ condition + (1 | participant) + (1 | verb), data=df)
```