

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'])
  df0$present_num[iR] = as.numeric(rownames(tAll_trials[tAll_trials$trial_id == df0$trialId[iR],]))
}

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`
```

29 workers and 779 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_with_yes = function(df){

  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_Pa
    return(list(df_Verb, df_Agent, df_Patient))
}

process_data_without_yes = function(df){
  df_Agent = df[ df$location_condition!='Control' & df$word_num=='2',]
  # df_Agent inheri row hum from df

  df_Verb = df[ df$location_condition!='Control' & df$word_num=='4',]

  df_Patient = df[ df$location_condition!='Control' & df$word_num=='5',]

  df_Agent$location_condition = mapvalues(df_Agent$location_condition, from=c("Patient", "Verb"), to=c('
  df_Verb$location_condition = mapvalues(df_Verb$location_condition, from=c("Agent", "Patient"), to=c('
  df_Patient$location_condition = mapvalues(df_Patient$location_condition, from=c("Agent", "Verb"), to=

  # 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_Pa
  return(list(df_Verb, df_Agent, df_Patient))
}

# c(df_Verb, df_Agent, df_Patient) %<-% process_data_with_yes(df2)
c(df_Verb, df_Agent, df_Patient) %<-% process_data_without_yes(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))
}

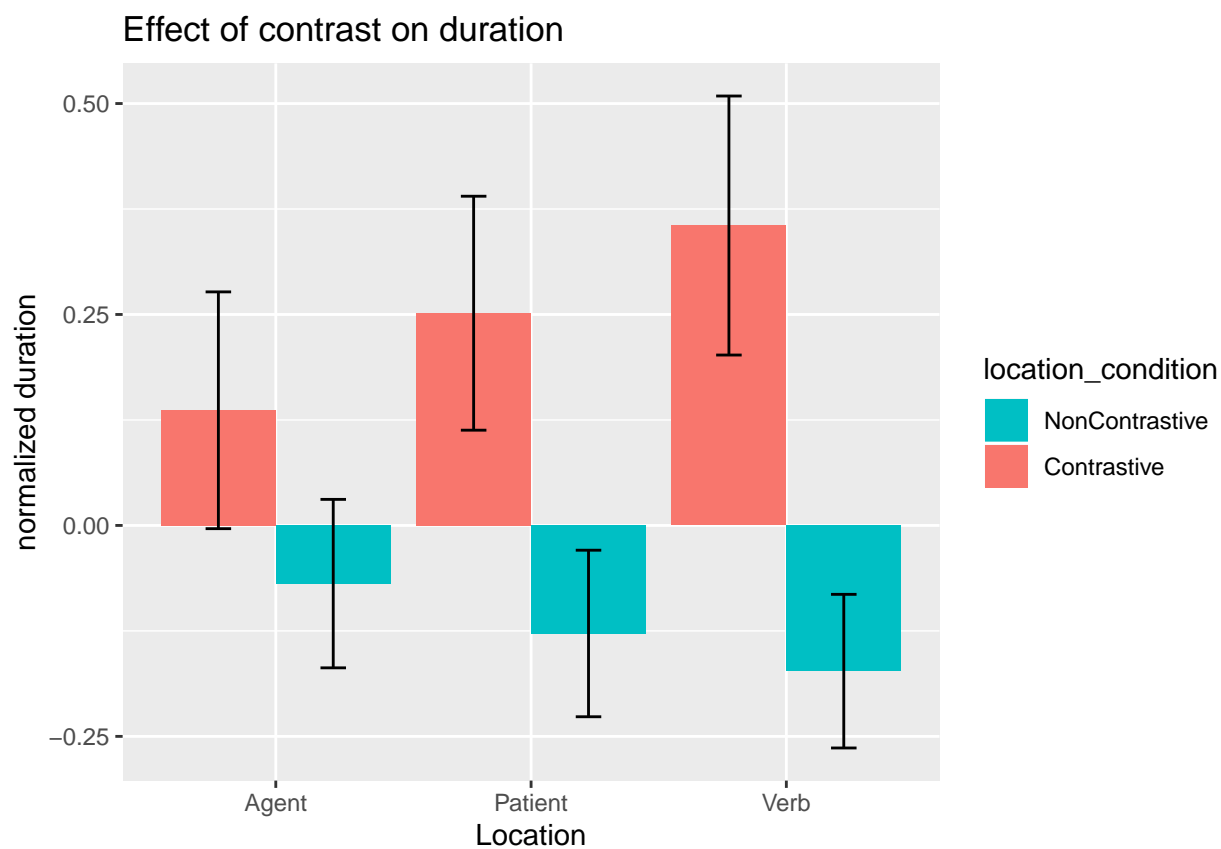
for (iF in features){
  print(iF)

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

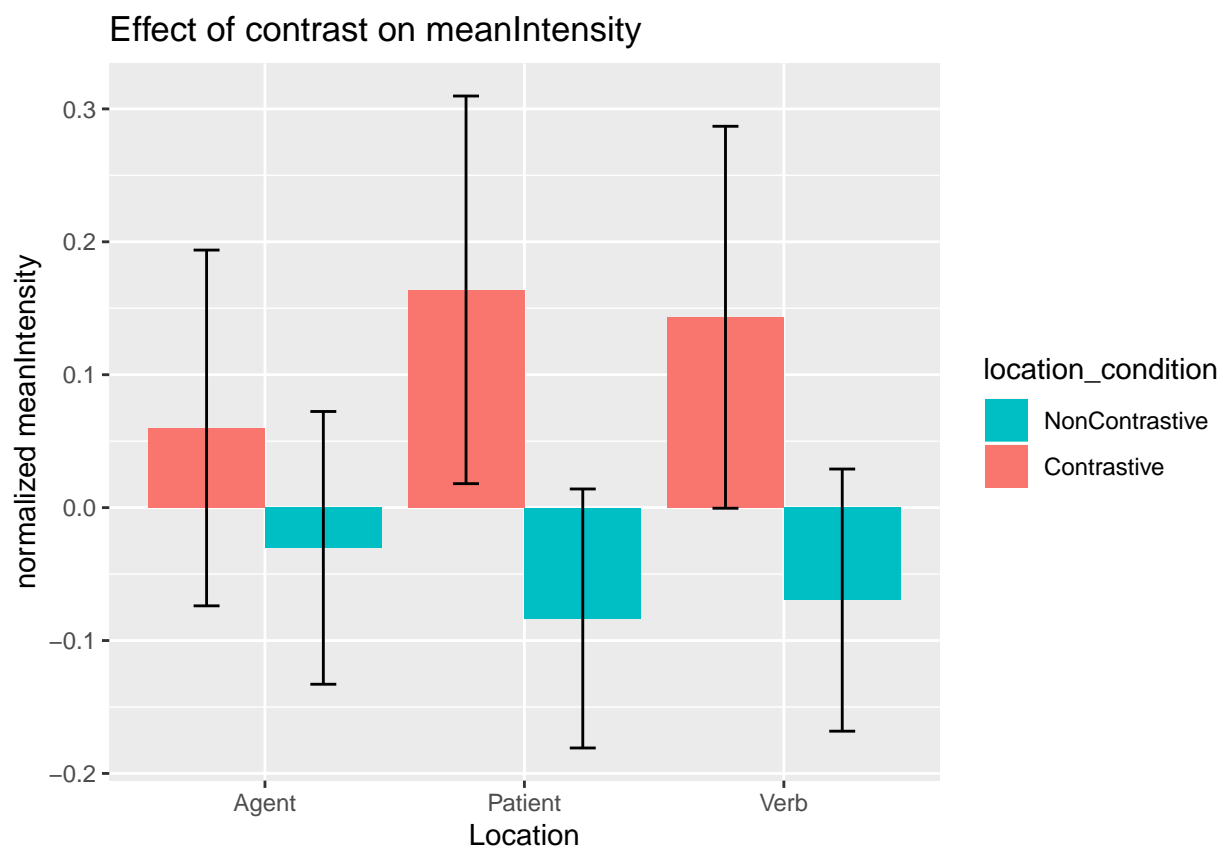
  plot_data(summarized_dataset,iF, title= paste0('Effect of contrast on ', iF))
}

```

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

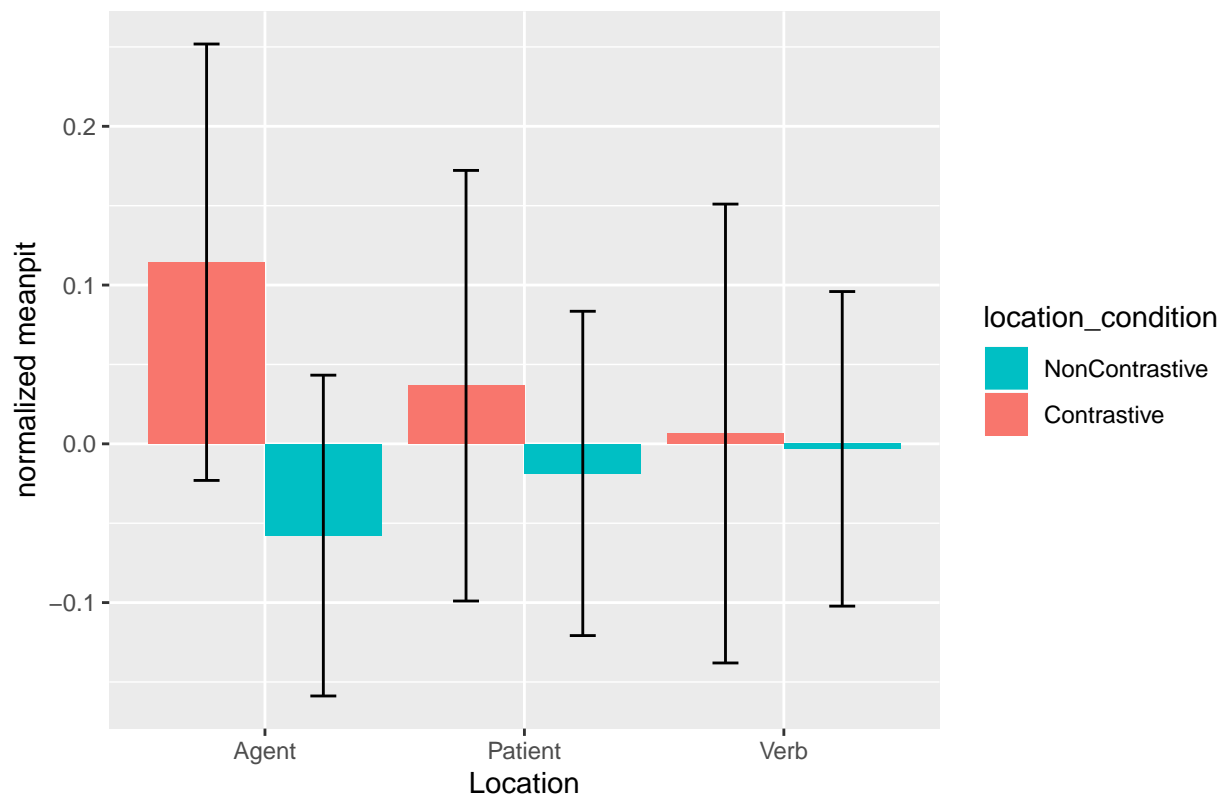


```
## [1] "meanIntensity"
```



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

Effect of contrast on 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: 1393.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5712 -0.6212 -0.0851  0.4639  5.1164
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   participant (Intercept)          0.50676  0.7119
##              location_conditionControl 0.02102  0.1450  -0.66
##   item_id     (Intercept)          0.05844  0.2417
##              location_conditionControl 0.02216  0.1489  -1.00
##   Residual                        0.55031  0.7418
## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      0.1280    0.1889   0.678
## location_conditionControl -0.2076    0.1030  -2.016
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnC -0.736
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
## ### duration of Patient
## 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: 1256.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.5386 -0.5002 -0.0827  0.3942  9.6576
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   participant (Intercept)          0.48388  0.6956
##              location_conditionPatient 0.03619  0.1902  -0.10
##   item_id     (Intercept)          0.12280  0.3504
##              location_conditionPatient 0.01369  0.1170   0.03
##   Residual                        0.41747  0.6461

```



```

## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)      -0.1854    0.2218  -0.836
## location_conditionPatient  0.3620    0.0895   4.045
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnP -0.063
##
##
## ### 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: 1422.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5903 -0.5460 -0.1116  0.3877  8.4649
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   participant (Intercept)          0.24022  0.4901
##               location_conditionVerb 0.17366  0.4167  -0.01
##   item_id     (Intercept)          0.03206  0.1790
##               location_conditionVerb 0.10970  0.3312   1.00
##   Residual                0.56880  0.7542
## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)      -0.1753    0.1347  -1.302
## location_conditionVerb  0.5005    0.1959   2.555
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnV 0.503
## 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: 899.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -5.3025 -0.4372  0.0609  0.5520  2.7598
##
## Random effects:
##   Groups      Name                Variance Std.Dev.  Corr
##   participant (Intercept)          0.668123 0.81739
##              location_conditionControl 0.005379 0.07334  1.00
##   item_id     (Intercept)          0.111989 0.33465
##              location_conditionControl 0.000212 0.01456 -1.00
##   Residual                        0.217983 0.46689
## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      -0.03550   0.23026  -0.154
## location_conditionControl -0.05731   0.04407  -1.301
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnC -0.020
## 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: 1150.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.06359 -0.55683 -0.05533  0.60969  3.15165
##
## Random effects:
##   Groups      Name                Variance Std.Dev.  Corr
##   participant (Intercept)          6.928e-01 8.323e-01
##              location_conditionPatient 1.625e-02 1.275e-01 -0.09
##   item_id     (Intercept)          0.000e+00 0.000e+00
##              location_conditionPatient 6.016e-10 2.453e-05  NaN
##   Residual                        3.492e-01 5.910e-01

```

```

## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)      -0.14034    0.16042  -0.875
## location_conditionPatient  0.28456    0.05755   4.944
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnP -0.138
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
## ### meanIntensity of Verb
## 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: 1051.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.5564 -0.5994  0.0043  0.5983  3.4662
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   participant (Intercept)          0.67397  0.8210
##               location_conditionVerb 0.03888  0.1972  -0.33
##   item_id     (Intercept)          0.08382  0.2895
##               location_conditionVerb 0.01352  0.1163   0.85
## Residual                        0.28286  0.5318
## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)      -0.12619    0.21401  -0.590
## location_conditionVerb  0.20622    0.08407   2.453
##
## Correlation of Fixed Effects:
##              (Intr)
## lctn_cndtnV 0.252
##
##
##
## ### meanpit 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: 1169.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.6163 -0.2486  0.0427  0.2902  4.5357
##
## Random effects:
##      Groups      Name                Variance Std.Dev. Corr
## participant (Intercept)            0.7225278 0.85002
##      location_conditionControl 0.0380160 0.19498 -0.30
## item_id      (Intercept)            0.0002211 0.01487
##      location_conditionControl 0.0082329 0.09074 -1.00
## Residual                        0.3565639 0.59713
## Number of obs: 581, groups: participant, 28; item_id, 4
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      0.03876   0.16675   0.232
## location_conditionControl -0.15569   0.07911  -1.968
##
## Correlation of Fixed Effects:
##      (Intr)
## lctn_cndtnC -0.302
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
## ### 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: 1402
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.7627 -0.4270 -0.0065  0.4620  3.4408
##
## Random effects:
##      Groups      Name                Variance Std.Dev. Corr
## participant (Intercept)            0.50132 0.70804
##      location_conditionPatient 0.35640 0.59699 -0.56
## item_id      (Intercept)            0.00000 0.00000
##      location_conditionPatient 0.00172 0.04148  NaN
## Residual                        0.53541 0.73172
## Number of obs: 581, groups: participant, 28; item_id, 4
##

```

```

## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      -0.02933    0.13956  -0.210
## location_conditionPatient  0.03015    0.13251   0.228
##
## Correlation of Fixed Effects:
##      (Intr)
## lctn_cndtnP -0.542
## 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: 1308.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.9838 -0.2786  0.0205  0.4185  4.1079
##
## Random effects:
##   Groups      Name                Variance Std.Dev.  Corr
##   participant (Intercept)          5.390e-01 7.342e-01
##               location_conditionVerb 4.095e-01 6.399e-01 -0.30
##   item_id     (Intercept)          3.233e-10 1.798e-05
##               location_conditionVerb 4.620e-09 6.797e-05 1.00
## Residual                        4.407e-01 6.639e-01
## Number of obs: 581, groups:  participant, 28; item_id, 4
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      -0.05830    0.14305  -0.408
## location_conditionVerb  0.02653    0.13590   0.195
##
## Correlation of Fixed Effects:
##      (Intr)
## lctn_cndtnV -0.324
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
# r = lmer(get(observation) ~ condition + (1 | participant) + (1 | verb), data=df)

```