acoustic_analysis

```
cur_exp = "exp1"
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 exp1. The parameters of all exps can be seen at https://github.com/Xinzhu-Fang/prosody_study_exp/blob/master/tAll_exps.csv.

The trial-by-trial design of this exp can be seen at https://github.com/Xinzhu-Fang/prosody_study_exp/blob/master/exp1/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$trialId[iR],'locat
  df0$item_id[iR] =  as.character(tAll_trials[tAll_trials$trial_id == df0$trialId[iR],'filler_or_item_i
  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`
```

30 workers and 820 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_Agen
    # 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=='5

  # 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)))
```

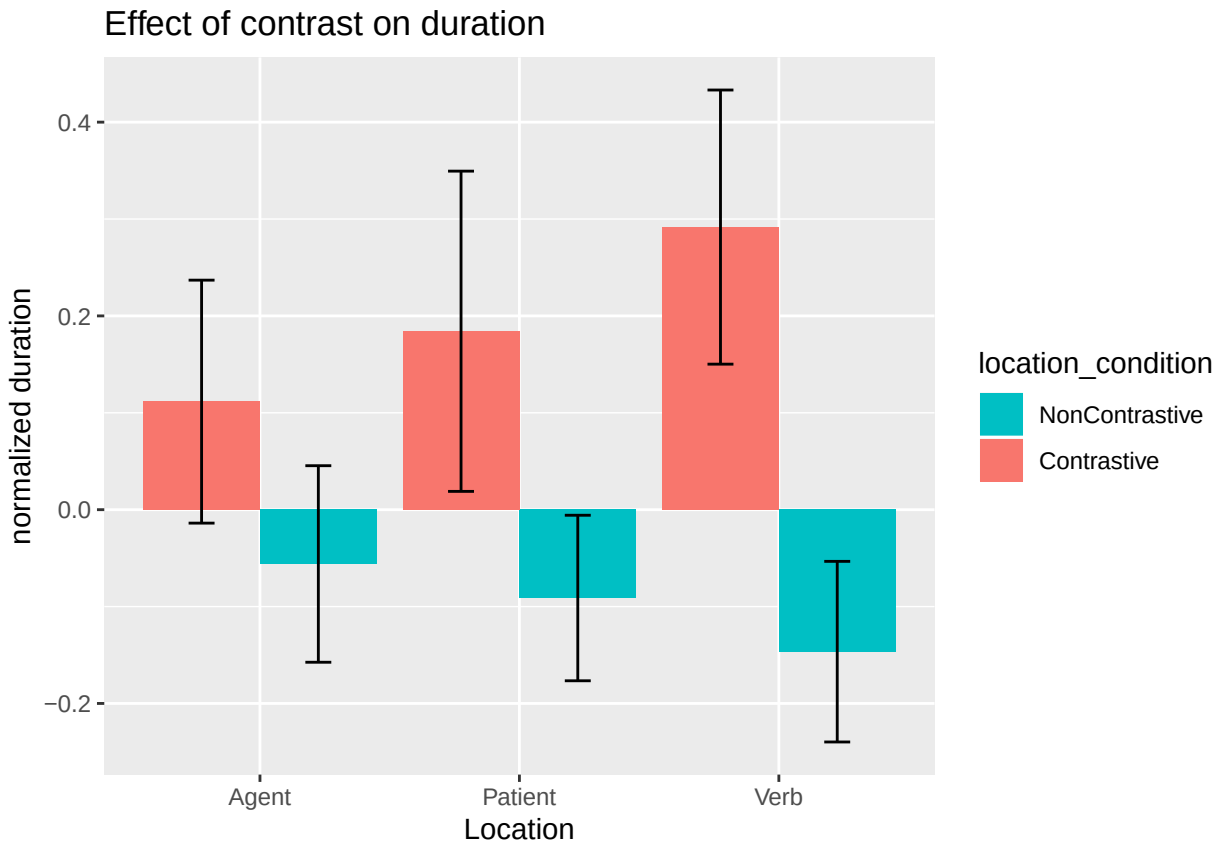
```
  # return(list(df_Agent_duration, df_Agent_meanIntensity, df_Agent_meanpit, df_Patient_duration, df_Pa
  return(list(df_Verb, df_Agent, df_Patient))
}

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
```

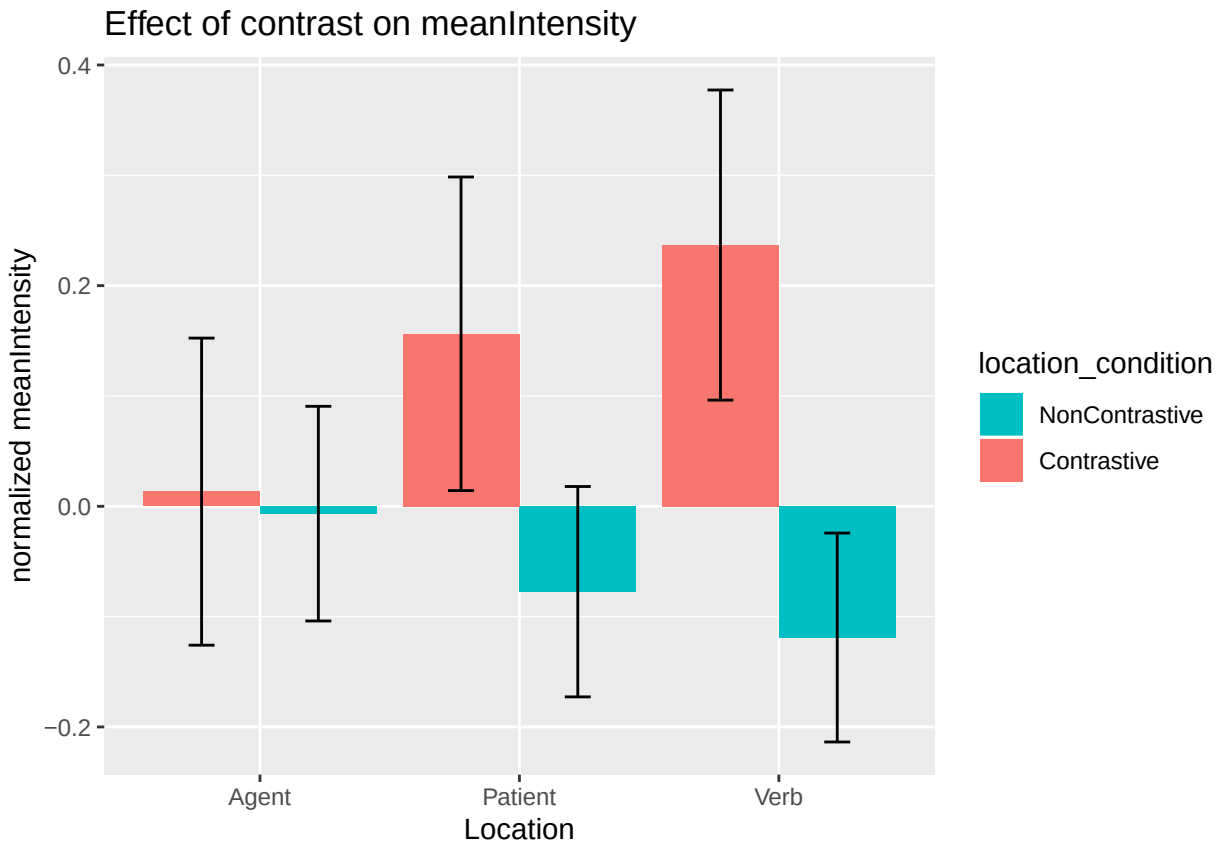
```
combine_datasets = function(Agent,Verb,Patient){
  Agent$condition = mapvalues(Agent$location_condition,c('Agent'),c('contrast'))
  Verb$condition = mapvalues(Verb$location_condition,c('Verb'),c('contrast'))
```

```
  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))
}
```

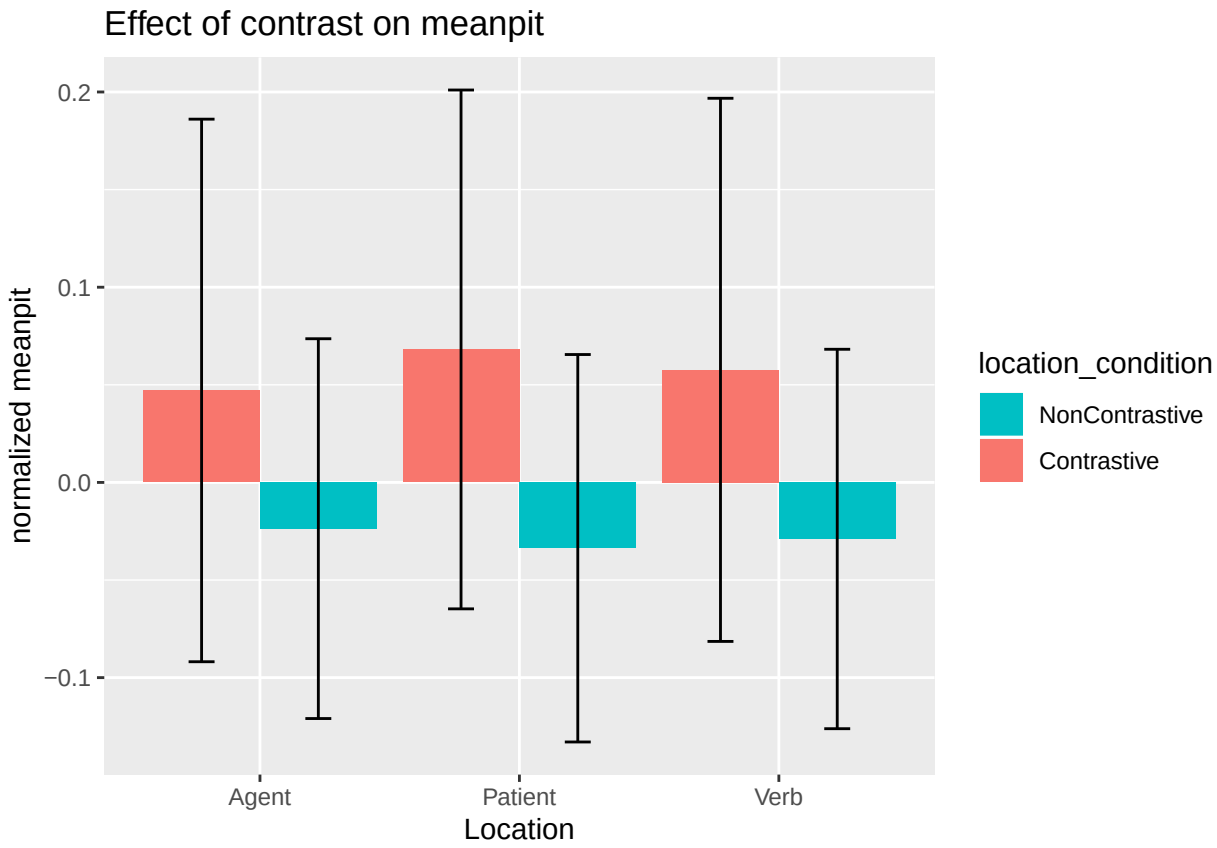
```
## [1] "duration"
```




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


Effect of contrast on meanIntensity
0.4
0.2
0.0
−0.2
normalized meanIntensity
Agent
Patient
Verb
Location
location_condition
NonContrastive
Contrastive

```
## [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: 1573.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2927 -0.4366 -0.0914  0.2637  8.7667
##
## Random effects:
##  Groups      Name                        Variance  Std.Dev. Corr
##  participant (Intercept)                 0.2680445 0.51773
##              location_conditionControl   0.0027208 0.05216  1.00
##  item_id     (Intercept)                 0.0004402 0.02098
##              location_conditionControl   0.0060055 0.07750  1.00
##  Residual                                0.6858009 0.82813
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                  0.11267    0.11375   0.991
## location_conditionControl -0.18247    0.08172  -2.233
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnC -0.212
## 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: 1438.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.7719 -0.3450 -0.0442  0.2728 11.5631
##
## Random effects:
##  Groups      Name                        Variance Std.Dev. Corr
##  participant (Intercept)                 0.359907 0.5999
##              location_conditionPatient   0.001755 0.0419   1.00
##  item_id     (Intercept)                 0.082855 0.2878
```

```
##              location_conditionPatient 0.025955 0.1611   0.86
##  Residual                                0.530864 0.7286
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                -0.07694    0.18631  -0.413
## location_conditionPatient   0.27309    0.10246   2.665
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnP 0.498
## 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: 1528.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0508 -0.4112 -0.0771  0.2813 12.4128
##
## Random effects:
##  Groups      Name                    Variance Std.Dev. Corr
##  participant (Intercept)             0.22502  0.4744
##              location_conditionVerb  0.03104  0.1762   1.00
##  item_id     (Intercept)             0.04972  0.2230
##              location_conditionVerb  0.03883  0.1970   0.17
##  Residual                            0.62700  0.7918
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)              -0.1608     0.1482  -1.085
## location_conditionVerb    0.4331     0.1245   3.478
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnV 0.182
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
```

```
## ### meanIntensity of Agent

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00311375
## (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: 1335
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.0703 -0.3656  0.1107  0.5107  2.3788
##
## Random effects:
##  Groups      Name                        Variance  Std.Dev. Corr
##  participant (Intercept)                 0.4542580 0.67399
##              location_conditionControl 0.0002648 0.01627  -1.00
##  item_id     (Intercept)                 0.1244253 0.35274
##              location_conditionControl 0.0044877 0.06699  1.00
##  Residual                                0.4406789 0.66384
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 0.04883    0.22175    0.22
## location_conditionControl -0.02119    0.06627   -0.32
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnC 0.231
## convergence code: 0
## Model failed to converge with max|grad| = 0.00311375 (tol = 0.002, component 1)
##
##
##
## ### 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: 1302.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6398 -0.5014 -0.0223  0.5071  3.2709
##
```

```
## Random effects:
##  Groups      Name                        Variance  Std.Dev. Corr
##  participant (Intercept)                 0.6254019 0.79082
##              location_conditionPatient 0.0138698 0.11777  -0.11
##  item_id     (Intercept)                 0.0139917 0.11829
##              location_conditionPatient 0.0005239 0.02289  1.00
##  Residual                                0.4118824 0.64178
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                -0.01938    0.16245  -0.119
## location_conditionPatient   0.23823    0.06083   3.916
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnP -0.070
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
## ### meanIntensity 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: 1224.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9234 -0.6321  0.0106  0.5564  3.4260
##
## Random effects:
##  Groups      Name                     Variance Std.Dev. Corr
##  participant (Intercept)              0.452120 0.67240
##              location_conditionVerb 0.034034 0.18448  0.09
##  item_id     (Intercept)              0.185153 0.43029
##              location_conditionVerb 0.002574 0.05073  1.00
##  Residual                             0.356731 0.59727
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)              -0.07318    0.25099  -0.292
## location_conditionVerb   0.34357    0.06753   5.088
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnV 0.294
```

```
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
## ### meanpit of Agent

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00470757
## (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: 1444.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3942 -0.2711  0.0033  0.2856  3.4793
##
## Random effects:
##  Groups      Name                        Variance Std.Dev. Corr
##  participant (Intercept)                 0.523040 0.72321
##              location_conditionControl   0.013180 0.11480  -0.49
##  item_id     (Intercept)                 0.009892 0.09946
##              location_conditionControl   0.006772 0.08229  -0.68
##  Residual                                0.535507 0.73178
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 0.04506    0.15307   0.294
## location_conditionControl -0.06215    0.07830  -0.794
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnC -0.452
## convergence code: 0
## Model failed to converge with max|grad| = 0.00470757 (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: 1411.8
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2562 -0.3226  0.0430  0.3327  4.0269
##
## Random effects:
##  Groups      Name                        Variance  Std.Dev. Corr
##  participant (Intercept)                 0.5002001 0.70725
##              location_conditionPatient 0.1222452 0.34964  -0.15
##  item_id     (Intercept)                 0.0004192 0.02047
##              location_conditionPatient 0.0010462 0.03235  1.00
##  Residual                                0.4920023 0.70143
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                -0.03172    0.13747  -0.231
## location_conditionPatient  0.08698    0.09140   0.952
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnP -0.188
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
## ### meanpit of Verb

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00563433
## (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: 1326.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3104 -0.2957  0.0534  0.3984  3.6805
##
## Random effects:
##  Groups      Name                     Variance Std.Dev. Corr
##  participant (Intercept)              0.568024 0.75367
##              location_conditionVerb 0.137021 0.37016  -0.23
##  item_id     (Intercept)              0.000655 0.02559
##              location_conditionVerb 0.004015 0.06337  -1.00
##  Residual                             0.419897 0.64799
## Number of obs: 610, groups:  participant, 29; item_id, 4
##
## Fixed effects:
```

```
##                         Estimate Std. Error t value
## (Intercept)             -0.02785    0.14518  -0.192
## location_conditionVerb  0.10013    0.09583   1.045
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnV -0.265
## convergence code: 0
## Model failed to converge with max|grad| = 0.00563433 (tol = 0.002, component 1)
##
##
```

```
# r = lmer(get(observation) ~ condition + (1 | participant) + (1 | verb), data=df)
```