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
```

```
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$trialId[iR],'locati
  df0$item_id[iR] =  as.character(tAll_trials[tAll_trials$trial_id == df0$trialId[iR],'filler_or_item_i
}

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)
colnames(df1)
```

```
##  [1] "participant"        "studyName"          "studyNameNah"
##  [4] "expId"              "month"              "date"
##  [7] "year"               "trialNah"           "trialId"
## [10] "word"               "wordlabel"          "phonelength"
## [13] "duration"           "silence"            "durasil"
## [16] "begin"              "meanpit"            "maxpitch"
## [19] "maxPitTime"         "minpitch"           "minPitTime"
## [22] "firstpitch"         "secondpitch"        "thirdpitch"
## [25] "fourthpitch"        "meanIntensity"      "maxIntensity"
## [28] "location_condition" "item_id"
```

```
# code for word_num
df2 <- df1 %>%
  dplyr::group_by(participant, trialId) %>%
  # dplyr::group_by(participant, question, trialId) %>%
  dplyr::mutate(word_num=1:dplyr::n()) %>%
  dplyr::select(c(info, features))
```

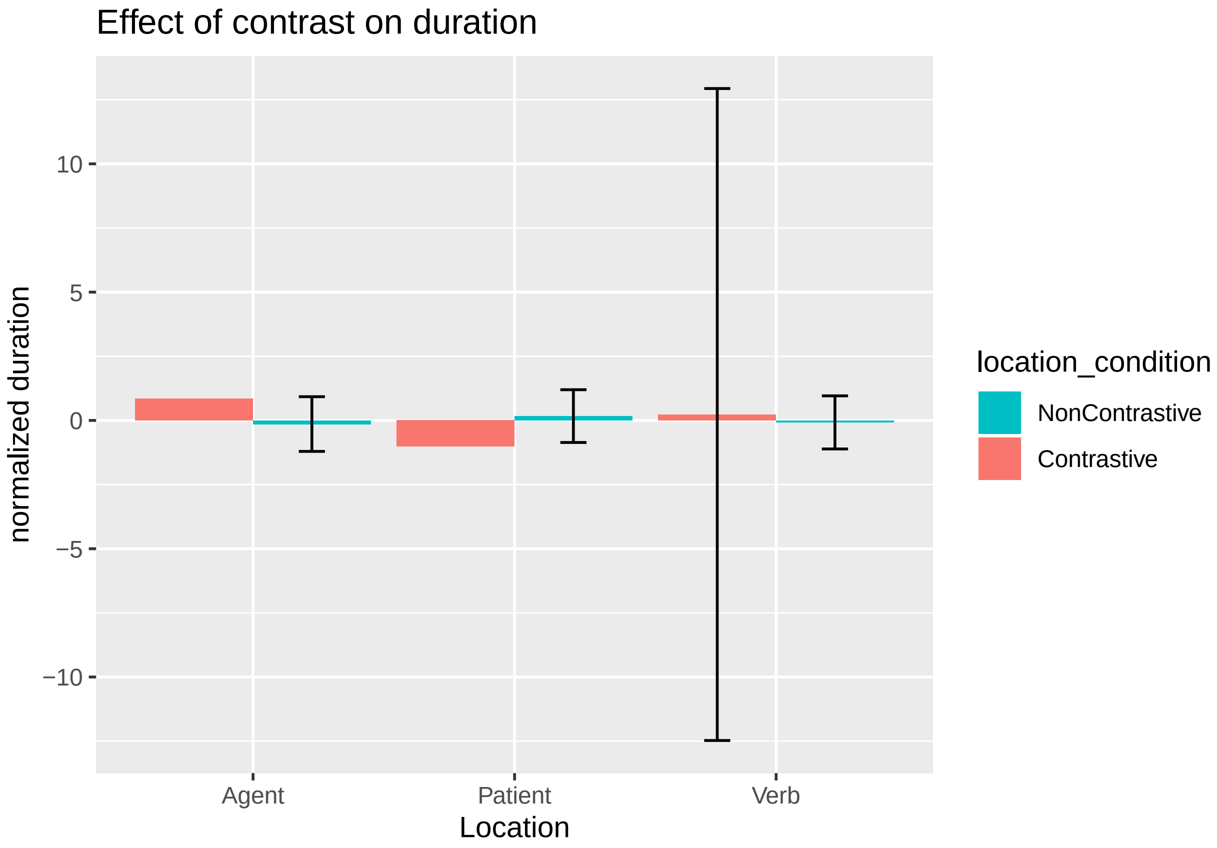
```
## Adding missing grouping variables: `trialId`
```

```
# 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 = 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=='

  df_Agent = df[(df$location_condition=='Agent' | df$location_condition=='Control') & df$word_num=='3',
  # df_Agent inheri row hum from df

  df_Verb = df[(df$location_condition=='Verb'| df$location_condition=='Control') & df$word_num=='5',]

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

  # 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))
}




#
# file_name = 'newdf.csv'

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

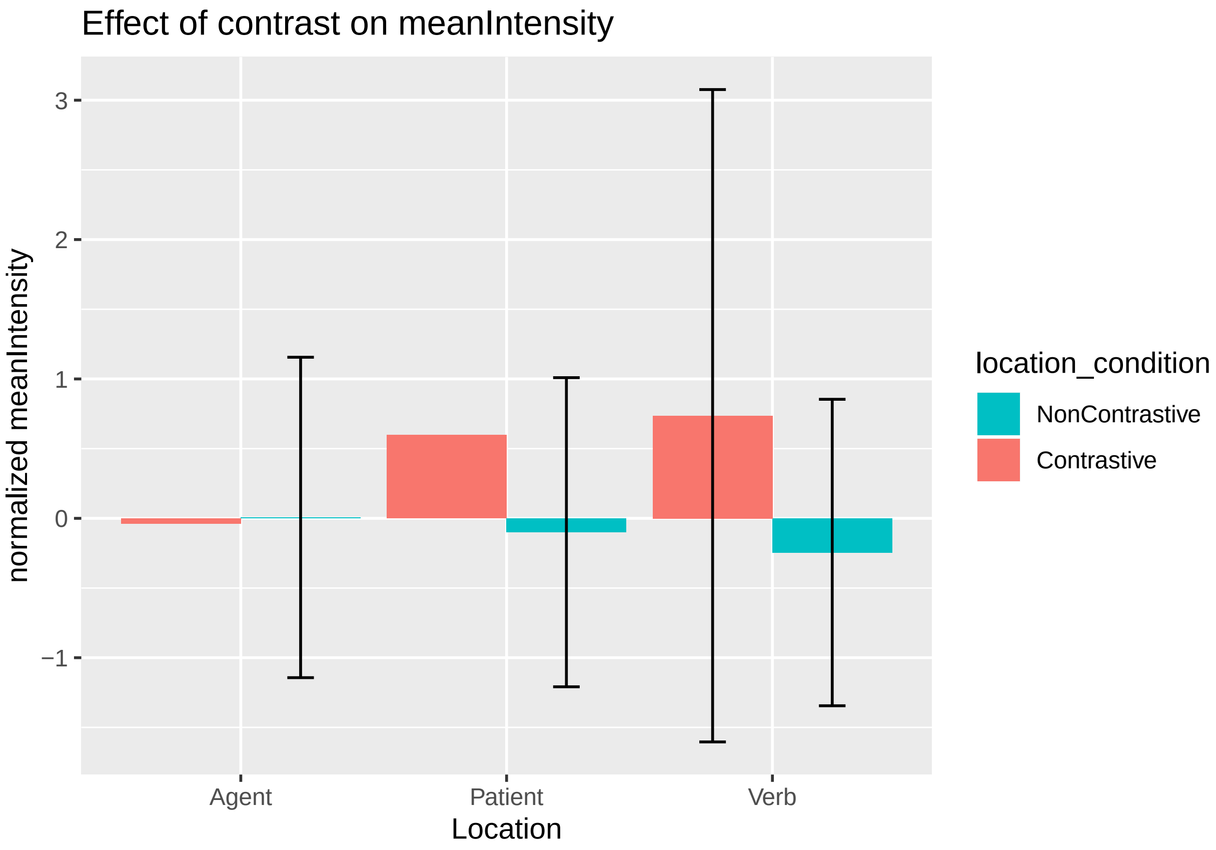
```
## [1] 0
```

```
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))
}
```

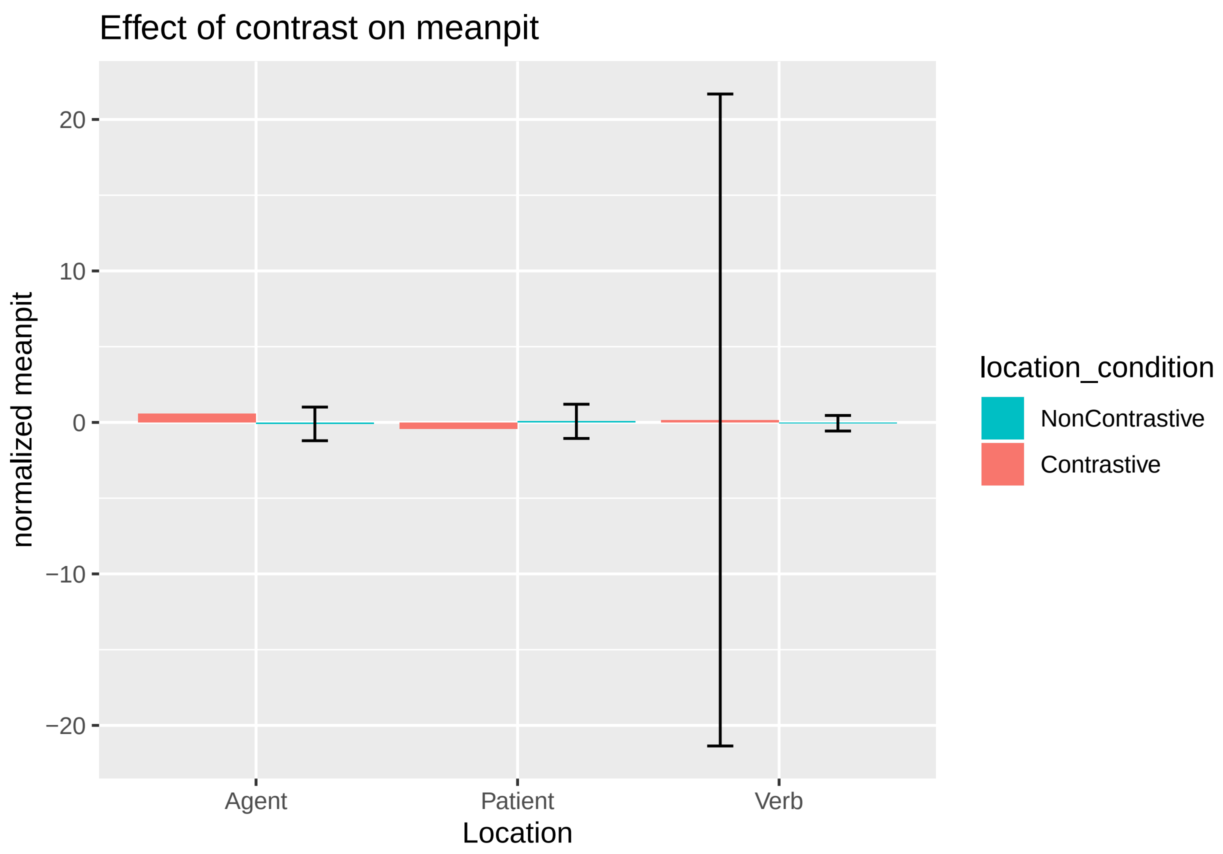
```
## [1] "duration"

## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs produced

## Warning: Removed 2 rows containing missing values (geom_errorbar).

## [1] "meanIntensity"

## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs produced
```



```
## Warning: Removed 2 rows containing missing values (geom_errorbar).

## [1] "meanpit"

## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs produced
```



```
## Warning: Removed 2 rows containing missing values (geom_errorbar).
```



```
run_regression = function(location,observation){
  cat("  \n###", observation, "of", location, "  \n")
  # r = lmer(get(observation) ~ location_condition + (1 + location_condition|participant) + (1 + locati
  r = lmer(get(observation) ~ location_condition  + (1 + location_condition | item_id), data=get(paste0
  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
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Hessian is numerically singular: parameters are not
## uniquely determined
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 13.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.85638 -0.53858 -0.03351  0.49024  0.98657
##
## Random effects:
##  Groups   Name                      Variance Std.Dev. Corr
##  item_id  (Intercept)               0.7901   0.8889
##           location_conditionControl 0.3757   0.6129   -0.20
##  Residual                           0.2717   0.5213
## Number of obs: 7, groups:  item_id, 3
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)                 0.6856     0.8809   0.778
## location_conditionControl  -0.8269     0.8265  -1.000
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnC -0.753
## convergence code: 0
##  Hessian is numerically singular: parameters are not uniquely determined
##
##
##
## ### duration of Patient
```

```
## 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, : Hessian is numerically singular: parameters are not
## uniquely determined
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 15.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.3002 -0.5057  0.0000  0.5880  1.1356
##
```

```
## Random effects:
##  Groups   Name                      Variance Std.Dev. Corr
##  item_id  (Intercept)               0.2459   0.4959
##           location_conditionPatient 0.5899   0.7681   -0.23
##  Residual                           0.7636   0.8738
## Number of obs: 7, groups:  item_id, 3
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)                 0.1689     0.4574   0.369
## location_conditionPatient  -0.9985     1.2256  -0.815
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnP -0.279
## convergence code: 0
## unable to evaluate scaled gradient
##  Hessian is numerically singular: parameters are not uniquely determined
##
##
##
## ### duration of Verb
```

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 9.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8702 -0.6671  0.0000  0.3934  1.2416
##
## Random effects:
##  Groups   Name                   Variance Std.Dev. Corr
##  item_id  (Intercept)             1.18652 1.0893
##           location_conditionVerb 66.52584 8.1563   -1.00
##  Residual                         0.02247 0.1499
## Number of obs: 8, groups:  item_id, 3
##
## Fixed effects:
##                        Estimate Std. Error t value
## (Intercept)            -0.07673    0.63186  -0.121
## location_conditionVerb -3.72295    4.71894  -0.789
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnV -0.994
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
```

```
##
##
## ### meanIntensity of Agent
```

```
## 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 |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 16.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.83245  0.08751  0.22220  0.41805  0.59914
##
## Random effects:
##  Groups   Name                      Variance Std.Dev. Corr
##  item_id  (Intercept)               0.3468   0.5889
##           location_conditionControl 0.7962   0.8923   -0.78
##  Residual                           0.9419   0.9705
## Number of obs: 7, groups:  item_id, 3
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)                -0.1186     1.1337  -0.105
## location_conditionControl   0.1249     1.2618   0.099
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnC -0.913
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
##
##
##
## ### meanIntensity of Patient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Hessian is numerically singular: parameters are not
## uniquely determined
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
```

```
## REML criterion at convergence: 9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.31862 -0.07278  0.04055  0.16397  1.09571
##
## Random effects:
##  Groups   Name                      Variance Std.Dev. Corr
##  item_id  (Intercept)               1.33280  1.1545
##           location_conditionPatient 0.05121  0.2263   0.15
##  Residual                           0.05086  0.2255
## Number of obs: 7, groups:  item_id, 3
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)               -0.09962    0.67286  -0.148
## location_conditionPatient  0.09677    0.35695   0.271
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnP 0.018
## convergence code: 0
##  Hessian is numerically singular: parameters are not uniquely determined
##
##
##
## ### meanIntensity of Verb
```

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 19.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.32316 -0.51317 -0.03856  0.27755  1.79613
##
## Random effects:
##  Groups   Name                   Variance  Std.Dev.  Corr
##  item_id  (Intercept)            0.000e+00 0.0000000
##           location_conditionVerb 8.699e-07 0.0009327  NaN
##  Residual                        9.259e-01 0.9622235
## Number of obs: 8, groups:  item_id, 3
##
## Fixed effects:
##                        Estimate Std. Error t value
## (Intercept)             -0.2454     0.3928  -0.625
## location_conditionVerb   0.9814     0.7857   1.249
##
## Correlation of Fixed Effects:
```

```
##             (Intr)
## lctn_cndtnV -0.500
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
##
## ### meanpit of Agent
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 11.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.90137 -0.44207  0.02325  0.34417  1.07393
##
## Random effects:
##  Groups   Name                      Variance Std.Dev. Corr
##  item_id  (Intercept)               0.2153   0.4640
##           location_conditionControl 0.5272   0.7261   0.77
##  Residual                           0.1109   0.3331
## Number of obs: 7, groups:  item_id, 3
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)                 0.3813     0.4607   0.828
## location_conditionControl  -0.4791     0.5785  -0.828
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnC -0.203
##
##
## ### meanpit of Patient
```

```
## 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, : Hessian is numerically singular: parameters are not
## uniquely determined
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 12.8
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -0.9147 -0.4344  0.0000  0.2421  1.2994
##
## Random effects:
##  Groups   Name                      Variance Std.Dev. Corr
##  item_id  (Intercept)               1.2206   1.1048
##           location_conditionPatient 0.1826   0.4274   -0.09
##  Residual                           0.1823   0.4270
## Number of obs: 7, groups:  item_id, 3
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)                0.07128    0.66127   0.108
## location_conditionPatient -0.42813    0.66880  -0.640
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnP -0.100
## convergence code: 0
## unable to evaluate scaled gradient
##  Hessian is numerically singular: parameters are not uniquely determined
##
##
##
## ### meanpit of Verb
```

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## get(observation) ~ location_condition + (1 + location_condition |
##     item_id)
##    Data: get(paste0("df_", location))
##
## REML criterion at convergence: 14.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.1037 -0.4492  0.0000  0.3401  1.4373
##
## Random effects:
##  Groups   Name                   Variance Std.Dev. Corr
##  item_id  (Intercept)            0.0000   0.0000
##           location_conditionVerb 5.4997   2.3451    NaN
##  Residual                        0.2389   0.4888
## Number of obs: 8, groups:  item_id, 3
##
## Fixed effects:
##                        Estimate Std. Error t value
## (Intercept)            -0.05332    0.19955  -0.267
## location_conditionVerb  0.21328    1.70561   0.125
##
## Correlation of Fixed Effects:
##             (Intr)
## lctn_cndtnV -0.117
## convergence code: 0
## boundary (singular) fit: see ?isSingular
##
##
```

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