German translation of the Artificial-Social-Agent questionnaire instrument for evaluating human-agent interaction

Underlying Analyses

Boleslav Khodakov

22 June, 2023

Contents

1 Introduction **1**

2 Data files summative_first_half_transformed.sav and summative_second_half_transformed.sav **2**

2.1 File data_culture_EN_DE.sav 3

3 Analyses results **3**

3.1 Correlation between English and German ASA Questionnaire 3

3.2 Variation Between English and German ASA Questionnaire 17

3.3 Comparison of Human-ASA Interaction between Different Cultural Backgrounds 36

References **42**

1 Introduction

This document presents statistical analyses of correlation and variation between English and German ASA questionnaires for item level, construct/dimension level, and short versions of the ASA questionnaire, as well as a comparison of human-ASA interaction between different cultural backgrounds, i.e., mixed international English-speaking group and bilingual group with German primary tongue.

Due to time constraints, the data transformed and analyzed in these files is only based on the first data collected between the 19th and 22nd of June. We are still aiming for 120 participants per questionnaire part for future work. Currently, data of 144 (72 per each half) participants is analyzed.

We use the following packages:

```
library(foreign) # Open various data files
library(nlme)    # Run multilevel linear models
library(car)     # Package linear regression
#install.packages("devtools")
#devtools::install_github("rasmusab/bayesian_first_aid")
library(BayesianFirstAid)  # Run Bayesian t-test
#install.packages("cmdstanr",
#   repos = c("https://mc-stan.org/r-packages/", getOption("repos")))
#devtools::install_github("rmcelreath/rethinking")
library(rethinking) # Run ulam
library(haven) # Use read_sav fuction
library(dplyr) # Use select function
library(knitr) # Get markdown file
library(tinytex) # Use TeX environment
library(rticles) # Use CTeX documents template
library(pander) # For pandering tables
panderOptions("table.alignment.default","left")
```

2 Data files summative_first_half_transformed.sav and summative_second_half_transformed.sav

The input data used in the analysis were transformed from two raw data files 'Final_ASA_German_Summative_First_
and 'Final_ASA_German_Summative_Second_Half_v1_2023_06_22_anonym_reduced.sav'. The detailed transformation from raw data to the input data files was explained in the markdown file 'Data Transformation Summative' (either for the first or second half).

We divided the 90 ASA items and participants into two groups to control fatigue effects. In the first group, human-ASA interaction evaluation data of the first 44 items (Construct 1-8: the first 12 constructs/dimensions) were collected from 72 bilingual participants with German primary tongue who are fluent German and fluent English speakers. In the second group, human-ASA interaction evaluation data of the last 46 items were collected from 82 bilingual participants with German primary tongue who are fluent German and fluent English speakers. The amount was artificially reduced to 72, since this codebase requires the two halves to have an equal amount of rows. Bilingual participants rated human-ASA interaction on 44/46 English items and corresponding German translations plus 14 attention control questions. All participants' evaluation data were included as they failed no attention control questions. We removed irrelevant data, e.g., attention control questions, just retaining scores of English items and corresponding German translations, also with 'AgentID' and 'agentName'. The

steps above were conducted and explained in the markdown files 'Transformation from raw data to the input data files', resulting in a two data file 'summative_first_half_transformed.sav' and 'summative_second_half_transformed.sav'. Up to this step, rating scores of 44/46 English items and corresponding German translations were ready for further analysis.

```
# Load data for first half
data01 <- data.frame(read_sav("summative_first_half_transformed.sav"))
# Select 44 item scores for English and German translation scores
d1 <- select(data01, Q_E_HLA1:Q_DE_R_AE4)
```

```
# Load data for second half
data02 <- data.frame(read_sav("summative_second_half_transformed.sav"))
# Select 46 item scores for English and German translation scores
d2 <- select(data02, Q_E_UE1:Q_DE_UAI4)
```

2.1 File data_culture_EN_DE.sav

Based on human-ASA interaction evaluation scores of 532 mixed international English-speaking participants in a previous study (accessible at https://osf.io/hxpsg) and 144 (72 per half) bilingual participants in this study, data_culture_EN_DE.sav is going to be used for exploring the differences in English questionnaire scores between these two cultural backgrounds. It consists of human-ASA evaluation on 24 English constructs and related dimensions for 14 ASAs by the participants from two cultural backgrounds.

```
data_culture <- data.frame(read_sav("data_culture_EN_DE.sav"))
```

3 Analyses results

3.1 Correlation between English and German ASA Questionnaire

We combined the scores of 44 items and 46 items as well as their corresponding translations in data frames 'd1' and 'd2'. Then we calculated ICC values for the 90 items. The multilevel model that we fit on the data set is a random intercept model. This model includes a fixed intercept (~1) and participant as a random intercept, indicated by random = ~1|id. Here, 'id' indicates the participant code for 72 bilingual participants whose scores were used to calculate ICC values.

3.1.1 ICC values for 90 items

We combined the scores of 44 items and 46 items as well as their corresponding translations in data frames 'd1' and 'd2'. Then we calculated ICC values for the 90 items. The multilevel model that

we fit on the data set is a random intercept model. This model includes a fixed intercept (~1) and participant as a random intercept, indicated by random = ~1|id. Here, 'id' indicates the participant code for 72 bilingual participants whose scores were used to calculate ICC values. We calculated ICC as: $\rho_I = \frac{\tau^2}{\tau^2+\sigma^2}$ whereby τ^2 is the variance between participants, and σ^2 is the variance within the score of individual (Finch, Bolin, and Kelley 2019). For the ICC calculation we defined the *getICC* function.

```
getICC <-function(model)
# Function for ICC value calculation using multilevel linear model
{
  vc.model <- VarCorr(model)
  # Estimated variances and correlations between the random-effects terms
  sigma_var <-as.numeric(vc.model[2,1])
  # Variance within the groups
  tau_var <- as.numeric(vc.model[1,1])
  # Variance between the groups
  icc <- tau_var/(tau_var + sigma_var)
  # Calculate ICC value
  return(icc)
}
```

Data frames 'd1' and 'd2' both have 72 data points, which we combined in single data frame.

```
# Combine evaluation scores of 44 items and 46 items for all participants
d_total <- cbind(select(d1,Q_E_HLA1:Q_E_R_AE4), select(d2,Q_E_UE1:Q_E_UAI4),
           select(d1,Q_DE_HLA1:Q_DE_R_AE4),select(d2,Q_DE_UE1:Q_DE_UAI4))
```

Next, we defined a function to run a multilevel model and obtain the associated ICC value for that model. As input, this function accepts the scores in both languages and the participant ID number. Before the model can be fitted this input data is transformed into a long format. The function returns ICC in value.

```
getLME <-function(s_1,s_2)
# Function for a linear mixed-effects model
{
  id<-rownames(s_2)
  # Row names that represent the ID number of each participant
  score_German<- data.frame(id, s_1, language= 1)
  # Transform German scores from wide format to long format and label as 1
  Score_English<- data.frame(id, s_2, language= 2)
  # Transform English scores from wide format to long format and label as 2
```

```
  Score_total <- rbind(score_German, Score_English)
  # Combine German and English scores in the long format
  m0 <- lme(score ~ 1, data = Score_total, random = ~1|id, method = "ML")
  # Linear mixed-effects model with a fixed intercept and
  # a random intercept of participant's ID number
  return(getICC(m0))
}
```

With the *getLME* function defined, the next step is to use this function to calculate the ICC value for each of the 90 ASA questionnaire items, and in addition, calculate the grand mean of these 90 ICC values. When going to the list of ASAQ items, we use the fact that in the data frame the first 90 columns present the results of the English ASAQ version and the last 90 columns present the results of the German ASAQ version.

```
calculate_item_ICC_values <- function(data, n=90){

  l_ICC <- data.frame(ItemID = double(), Item = character(), icc = double())

  # Numbers of columns in d_total
  German_column_offset <- ncol(data) /2


  # The value of n is equal to the number of columns divided by 2.
  for (i in 1:n)
  # Go step by step to 90 items of the ASA questionnaire,
  # whereby i is the ASA questionnaire item number
  {

    # Select scores of German version of ASAQ item i
    score_German <- data.frame(score=data[,i + German_column_offset])

    # Select scores of English version of ASAQ items i
    score_English <- data.frame(score=data[,i])


    # Calculated ICC and add it to the list of ICC values,
    # with ID number of the ASA questionnaire item
    l_ICC <- rbind(l_ICC, data.frame (i, icc = getLME(score_German, score_English)))
```

```
  }
  return(l_ICC)
}
```

```
l_ICC <- calculate_item_ICC_values(d_total)

l_ICC$Item = colnames(select(d_total,Q_E_HLA1:Q_E_UAI4)) # Add name code for each item
pander(l_ICC, caption = "All participants - ICC values for 90 items")
```

Table 1: All participants - ICC values for 90 items

i	icc	Item
1	0.785	Q_E_HLA1
2	0.8998	Q_E_HLA2
3	0.7699	Q_E_HLA3
4	0.7361	Q_E_HLA4
5	0.6751	Q_E_HLB1
6	0.6237	Q_E_HLB2
7	0.7755	Q_E_HLB3
8	0.8043	Q_E_HLB4
9	0.7978	Q_E_HLB5
10	0.694	Q_E_NA1
11	0.6196	Q_E_NA2
12	0.6491	Q_E_NA3
13	0.617	Q_E_NA4
14	0.682	Q_E_NA5
15	0.896	Q_E_NB1
16	0.7519	Q_E_NB2
17	0.7917	Q_E_NB3
18	0.6119	Q_E_AAS1
19	0.5568	Q_E_AAS2
20	0.7114	Q_E_AAS3
21	0.6126	Q_E_AU1
22	0.4496	Q_E_AU2
23	0.6261	Q_E_AU3
24	0.7092	Q_E_PF1
25	0.5804	Q_E_PF2
26	0.4782	Q_E_PF3
27	0.5431	Q_E_AL1

i	icc	Item
28	0.8967	Q_E_AL2
29	0.8262	Q_E_R_AL3
30	0.6235	Q_E_AL4
31	0.8574	Q_E_AL5
32	0.3082	Q_E_AS1
33	0.2689	Q_E_AS2
34	0.7527	Q_E_AS3
35	0.5597	Q_E_APP1
36	0.744	Q_E_R_APP2
37	0.8843	Q_E_APP3
38	0.6611	Q_E_UAA1
39	0.7027	Q_E_UAA2
40	0.3393	Q_E_R_UAA3
41	0.8042	Q_E_R_AE1
42	0.7794	Q_E_AE2
43	0.7166	Q_E_AE3
44	0.4961	Q_E_R_AE4
45	0.4911	Q_E_UE1
46	0.5671	Q_E_UE2
47	0.2907	Q_E_UE3
48	0.6429	Q_E_UT1
49	0.4347	Q_E_UT2
50	0.5686	Q_E_UT3
51	0.7403	Q_E_UAL1
52	0.4532	Q_E_UAL2
53	0.5793	Q_E_UAL3
54	0.5584	Q_E_UAL4
55	0.6259	Q_E_UAL5
56	0.6321	Q_E_UAL6
57	0.5202	Q_E_AA1
58	0.3927	Q_E_AA2
59	0.6596	Q_E_AA3
60	0.6271	Q_E_R_AC1
61	0.6463	Q_E_R_AC2
62	0.7909	Q_E_R_AC3
63	0.758	Q_E_R_AC4
64	0.4817	Q_E_AI1
65	0.7158	Q_E_AI2

i	icc	Item
66	0.6752	Q_E_R_AI3
67	0.7512	Q_E_AI4
68	0.7773	Q_E_AT1
69	0.7451	Q_E_AT2
70	0.8209	Q_E_R_AT3
71	0.743	Q_E_SP1
72	0.7205	Q_E_SP2
73	0.6125	Q_E_SP3
74	0.7687	Q_E_IIS1
75	0.7251	Q_E_IIS2
76	0.5317	Q_E_IIS3
77	0.8366	Q_E_IIS4
78	0.6466	Q_E_AEI1
79	0.7725	Q_E_AEI2
80	0.503	Q_E_R_AEI3
81	0.7267	Q_E_AEI4
82	0.7378	Q_E_R_AEI5
83	0.4973	Q_E_UEP1
84	0.5745	Q_E_UEP2
85	0.7135	Q_E_UEP3
86	0.6264	Q_E_UEP4
87	0.7	Q_E_UAI1
88	0.2868	Q_E_UAI2
89	0.5812	Q_E_UAI3
90	0.6946	Q_E_UAI4

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(l_ICC$icc),digits=4),round(sd(l_ICC$icc),digits=4),
           round(min(l_ICC$icc),digits=4),round(max(l_ICC$icc),digits=4))
# Calculate the grand mean, standard deviation,
# minimum and maximum values of ICC values of 90 items
description <- cbind(Variable, Value) # Descriptive statistics of ICC values of 90 items

# Print results
pander(description, caption = paste("All participants - Descriptive",
                                    "statistics of ICC values of 90 items"))
```

Table 2: All participants - Descriptive statistics of ICC values of 90 items

Variable	Value
Grand_mean	0.6513
SD	0.1427
Minimum	0.2689
Maximum	0.8998

For the assessment of the correlation between the English and German ASA Questionnaire, we followed Cicchetti's classification of ICC categories (Cicchetti 1994). Then we get the categories of ICC classifications and number of ICC values in classification category.

```
Classification <- c("Excellent","Good","Fair","Poor")
ICC_Range <- c("0.75-1.00","0.60-0.74","0.40-0.59","0-0.39")
# Categories of ICC classifications by Cicchetti (1994)
n_item <- length(l_ICC$icc) # Number of ICC values
round_ICC <- round(l_ICC$icc, digits=4) # Round ICC values
Number <- c(length(l_ICC[which(round_ICC>=0.75&round_ICC<=1),]$icc),
            length(l_ICC[which(round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(l_ICC[which(round_ICC>=0.40&round_ICC<=0.59),]$icc),
            length(l_ICC[which(round_ICC>=0.00&round_ICC<=0.39),]$icc))
# Calculate number of ICC values in classification category
Percentage <- c(round(Number[1]/n_item,digits=4)*100, round(Number[2]/n_item,digits=4)*100,
                round(Number[3]/n_item,digits=4)*100, round(Number[4]/n_item,digits=4)*100)
# Calculate percentage of ICC values in classification category
ICC_category <- cbind(Classification,ICC_Range,Number,Percentage)

# Print results
pander(ICC_category, caption = "Categories of ICC classifications and
       number of ICC values in classification category for 90 items")
```

Table 3: Categories of ICC classifications and number of ICC values in classification category for 90 items

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	24	26.67
Good	0.60-0.74	35	38.89

Classification	ICC_Range	Number	Percentage
Fair	0.40-0.59	21	23.33
Poor	0-0.39	5	5.56

3.1.2 Removing English Prefix 'Q_E_'

For easier legibility of the code below, and for better compatibility with the legacy codebase (from the Chinese translation creation/validation), the Prefix 'Q_E_' is removed from English items (e.g. 'HLA1' instead of 'Q_E_HLA1'). The German item-prefixes ('Q_DE_') remain.

```
for ( col in 1:90){
    colnames(d_total)[col] <-  sub("Q_E_", "", colnames(d_total)[col])
}
```

3.1.3 ICC values for 24 constructs and related dimensions

We combined the scores of Construct1-8 (first half) and Construct 9-19 (second half), as the input data for the correlation analysis for 24 constructs/dimensions. Then we called the function *getLME* to calculate ICC values for each construct/dimension.

```
German_column_offset = ncol(d_total)/2

i <- which(names(d_total)%in%c("HLA1","HLB1","NA1","NB1","AAS1","AU1","PF1","AL1",
          "AS1","APP1","UAA1","R_AE1","UE1","UT1","UAL1","AA1","R_AC1","AI1","AT1",
          "SP1","IIS1","AEI1","UEP1","UAI1"))
# 'i' is a vector with the column number of the first English
# version of item of the construct/dimension

k1 <- c(ncol(select(d_total, HLA1:HLA4)),ncol(select(d_total, HLB1:HLB5)),
        ncol(select(d_total, NA1:NA5)),ncol(select(d_total, NB1:NB3)),
        ncol(select(d_total, AAS1:AAS3)),ncol(select(d_total, AU1:AU3)),
        ncol(select(d_total, PF1:PF3)),ncol(select(d_total, AL1:AL5)),
        ncol(select(d_total, AS1:AS3)),ncol(select(d_total, APP1:APP3)),
        ncol(select(d_total, UAA1:R_UAA3)),ncol(select(d_total, R_AE1:R_AE4)))
# 'k1' is a vector with the number of questionnaire items of each
# construct/dimension for Construct 1-8
# Note that we assume here that construct/dimension items are
# adjacent columns in the data frame
```

```
k2 <- c(ncol(select(d_total, UE1:UE3)),ncol(select(d_total, UT1:UT3)),
        ncol(select(d_total, UAL1:UAL6)),ncol(select(d_total, AA1:AA3)),
        ncol(select(d_total, R_AC1:R_AC4)),ncol(select(d_total, AI1:AI4)),
        ncol(select(d_total, AT1:R_AT3)),ncol(select(d_total, SP1:SP3)),
        ncol(select(d_total, IIS1:IIS4)),ncol(select(d_total, AEI1:R_AEI5)),
        ncol(select(d_total, UEP1:UEP4)),ncol(select(d_total, UAI1:UAI4)))
# 'k2' is a vector with the number of questionnaire items of each
# construct/dimension for Construct 9-19

k = c(k1,k2)
# Combine k1 and k2 into a single vector with the number of questionnaire
# items of each construct/dimension of the entire ASAQ
h <- cbind.data.frame(i,k)
# Combine i and k into a data frame, whereby i indicates the column number
# of the first English item of a construct and k the total number of adjacent
# questionnaire items associated with the construct

l_ICC <- data.frame(ConstructID=double(), Construct=character(), icc=double())
# Initialize output of ICC values of 24 constructs/dimensions

for( p in 1:24 )
# Go step by step to 24 constructs/dimensions of the ASA questionnaire
{
  i <- h[p,1]
  # Column number of the first ASAQ item in English of the construct/dimension
  j <- i+ German_column_offset
  # The column number of the first ASAQ item in the
  # German version of the construct/dimension
  k <- h[p,2]
  # The number of ASAQ items associate to the construct/dimension
  s_German <- data.frame(d_total[,j:(j+k-1)])
  # Select the scores of all the ASAQ items in German
  # associated with the construct/dimension
  s_English <- data.frame(d_total[,i:(i+k-1)])
  # Select the score of all the ASAQ items in English associated
  # with the construct/dimension
  average_s_German <- data.frame(rowMeans(s_German))
  # Calculate the mean score of ASAQ items in German associated
  # with the construct/dimension per participant
```

```
  average_s_English <- data.frame(rowMeans(s_English))
  # Doing the same but now for English version of the items
  colnames(average_s_German) <- c("score") # Rename German mean column
  colnames(average_s_English) <- c("score") # Rename English mean column
  l_ICC <- rbind(l_ICC, data.frame (p, icc = getLME(average_s_German, average_s_English)))
  # Call function 'getLME' for ICC value calculation
}
l_ICC$Construct = c('HLA','HLB','NA','NB','AAS','AU','PF','AL','AS','APP',
'UAA','AE','UE','UT','UAL','AA','AC','AI','AT','SP','IIS','AEI','UEP','UAI')
# Add construct/dimension name code
pander(l_ICC, caption = "ICC values for 24 constructs/dimensions")
```

Table 4: ICC values for 24 constructs/dimensions

p	icc	Construct
1	0.907	HLA
2	0.8893	HLB
3	0.8317	NA
4	0.9113	NB
5	0.7615	AAS
6	0.7704	AU
7	0.731	PF
8	0.9263	AL
9	0.5814	AS
10	0.8497	APP
11	0.7219	UAA
12	0.8166	AE
13	0.5111	UE
14	0.7168	UT
15	0.8288	UAL
16	0.6597	AA
17	0.8155	AC
18	0.8095	AI
19	0.9177	AT
20	0.8388	SP
21	0.8732	IIS
22	0.8768	AEI
23	0.8024	UEP
24	0.8311	UAI

p	icc	Construct

```
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(l_ICC$icc),digits=4),round(sd(l_ICC$icc),digits=4),
           round(min(l_ICC$icc),digits=4),round(max(l_ICC$icc),digits=4))
# Calculate the grand mean, standard deviation, minimum and
# maximum values of ICC values of 24 constructs/dimensions
description <- cbind(Variable, Value)
# Descriptive statistics of ICC values of 24 constructs/dimensions

# Print results
pander(description, caption = "Descriptive statistics of ICC values
of 24 constructs/dimensions")
```

Table 5: Descriptive statistics of ICC values of 24 constructs/dimensions

Variable	Value
Grand_mean	0.7991
SD	0.1045
Minimum	0.5111
Maximum	0.9263

```
Classification <- c("Excellent","Good","Fair","Poor")
ICC_Range <- c("0.75-1.00","0.60-0.74","0.40-0.59","0-0.39")
# Categories of ICC classifications by Cicchetti (1994)
n_item <- length(l_ICC$icc) # Number of ICC values
round_ICC <- round(l_ICC$icc, digits=2) # Round ICC values
Number <- c(length(l_ICC[which(round_ICC>=0.75&round_ICC<=1),]$icc),
            length(l_ICC[which(round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(l_ICC[which(round_ICC>=0.40&round_ICC<=0.59),]$icc),
            length(l_ICC[which(round_ICC>=0.00&round_ICC<=0.39),]$icc))
# Calculate number of ICC values in classification category
Percentage <- c(round(Number[1]/n_item,digits=4)*100, round(Number[2]/n_item,digits=4)*100,
                round(Number[3]/n_item,digits=4)*100, round(Number[4]/n_item,digits=4)*100)
# Calculate percentage of ICC values in classification category
ICC_category <- cbind(Classification,ICC_Range,Number,Percentage)
```

```
# Print results
pander(ICC_category, caption = "Categories of ICC classifications and number
       of ICC values in classification category for 24 constructs/dimensions")
```

Table 6: Categories of ICC classifications and number of ICC values in classification category for 24 constructs/dimensions

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	18	75
Good	0.60-0.74	4	16.67
Fair	0.40-0.59	2	8.33
Poor	0-0.39	0	0

3.1.4 ICC values between English and German scores for the short version of ASA questionnaire

The last ICC calculation is for the ASAQ items of the short version of the ASAQ. The procedure is similar to ICC calculation of the 90 items, only this time, we select only the relevant 24 items first.

```
s_German <- select(d_total,Q_DE_HLA2,Q_DE_HLB5,Q_DE_NA4,Q_DE_NB3,Q_DE_AAS1,Q_DE_AU1,Q_DE_PF1,
      Q_DE_AL2,Q_DE_AS1,Q_DE_APP1,Q_DE_UAA1,Q_DE_R_AE1,Q_DE_UE2,Q_DE_UT3,Q_DE_UAL1,
      Q_DE_AA2,Q_DE_R_AC1,Q_DE_R_AI3,Q_DE_AT1,Q_DE_SP2,Q_DE_IIS2,Q_DE_R_AEI3,Q_DE_UEP3,Q_DE_UAI4)
# Select German versions of the 24 representative ASAQ items
s_English <- select(d_total,HLA2,HLB5,NA4,NB3,AAS1,AU1,PF1,AL2,AS1,APP1,UAA1,
      R_AE1,UE2,UT3,UAL1,AA2,R_AC1,R_AI3,AT1,SP2,IIS2,R_AEI3,UEP3,UAI4)
# Select English versions of the 24 representative ASAQ items
ss <- cbind(s_German,s_English)
# Combine German and English scores

n <- ncol(ss) # Numbers of all columns in ss
English_column_offset <- n /2

l_ICC <- data.frame(ID=double(), Item=character(), icc=double())
# Initialize output of ICC values of 24 representative items
for (i in 1:24)
# Go step by step to 24 representative items of the ASA questionnaire
{
  score_German <- data.frame(score=ss[,i])
```

```r
  # Select German scores of the ASAQ item
  score_English <- data.frame(score=ss[,i+ English_column_offset])
  # Select English scores of the ASAQ item
  l_ICC <- rbind(l_ICC, data.frame (i, icc = getLME(score_German, score_English)))
  # Call function 'getLME' for ICC value calculation
}
l_ICC$Item <- colnames(s_English) # Add item name code
pander(l_ICC, caption = "ICC values for 24 representative items")
```

Table 7: ICC values for 24 representative items

i	icc	Item
1	0.8998	HLA2
2	0.7978	HLB5
3	0.617	NA4
4	0.7917	NB3
5	0.6119	AAS1
6	0.6126	AU1
7	0.7092	PF1
8	0.8967	AL2
9	0.3082	AS1
10	0.5597	APP1
11	0.6611	UAA1
12	0.8042	R_AE1
13	0.5671	UE2
14	0.5686	UT3
15	0.7403	UAL1
16	0.3927	AA2
17	0.6271	R_AC1
18	0.6752	R_AI3
19	0.7773	AT1
20	0.7205	SP2
21	0.7251	IIS2
22	0.503	R_AEI3
23	0.7135	UEP3
24	0.6946	UAI4

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(l_ICC$icc),digits=4),round(sd(l_ICC$icc),digits=4),
           round(min(l_ICC$icc),digits=4),round(max(l_ICC$icc),digits=4))
# Calculate the grand mean, standard deviation, minimum
# and maximum values of ICC values of 24 representative items
description <- cbind(Variable, Value)
# Descriptive statistics of ICC values of 24 representative items

# Print results
pander(description, caption = "Descriptive statistics of ICC values
of 24 representative items")
```

Table 8: Descriptive statistics of ICC values of 24 representative items

Variable	Value
Grand_mean	0.6656
SD	0.1413
Minimum	0.3082
Maximum	0.8998

```r
Classification <- c("Excellent","Good","Fair","Poor")
ICC_Range <- c("0.75-1.00","0.60-0.74","0.40-0.59","0-0.39")
# Categories of ICC classifications by Cicchetti (1994)
n_item <- length(l_ICC$icc) # Number of ICC values
round_ICC <- round(l_ICC$icc, digits=2) # Round ICC values
Number <- c(length(l_ICC[which(round_ICC>=0.75&round_ICC<=1),]$icc),
            length(l_ICC[which(round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(l_ICC[which(round_ICC>=0.40&round_ICC<=0.59),]$icc),
            length(l_ICC[which(round_ICC>=0.00&round_ICC<=0.39),]$icc))
# Calculate number of ICC values in classification category
Percentage <- c(round(Number[1]/n_item,digits=4)*100, round(Number[2]/n_item,digits=4)*100,
                round(Number[3]/n_item,digits=4)*100, round(Number[4]/n_item,digits=4)*100)
# Calculate percentage of ICC values in classification category
ICC_category <- cbind(Classification,ICC_Range,Number,Percentage)

# Print results
```

```
pander(ICC_category, caption = "Categories of ICC classifications and number
       of ICC values in classification category for 24 representative items")
```

Table 9: Categories of ICC classifications and number of ICC values in classification category for 24 representative items

Classification	ICC_Range	Number	Percentage
Excellent	0.75-1.00	6	25
Good	0.60-0.74	12	50
Fair	0.40-0.59	4	16.67
Poor	0-0.39	2	8.33

3.2 Variation Between English and German ASA Questionnaire

The results were reported in the subsection of Variation Between English and German ASA Questionnaire. The mean score differences between the English and German questionnaires are estimates for absolute accuracy in score equivalence between the two languages. 95% credible interval of mean paired difference was calculated by Bayesian paired t-test, for item level, construct and dimension level, and the short version of the ASA questionnaire. We used the combined input data of both halves.

3.2.1 Mean score differences for 90 items

We used the Bayesian pairwise t-test to estimate the difference in ASAQ items score between the English and the German version. First we define function establish sample means and standard deviation, next relevant information is extracted from output date produced by Bayesian t-test.

```
getBAYES <-function(ID, ss_1, ss_2, B_output)
# Function to obtain mean, and sd values of ss_1 (German)
# and ss_2 (English), and relevant information from
# Bayesian t-test output stored in B_output,
# this is take from the 1 line for Bayes output
# which relates to the estimation of the means and mean difference
# ID is the identification number added in the return data
# frame row to identify an item or construct
{ l <- data.frame(ID,
                  mean_German = mean(ss_1), # Mean of German translation
                  sd_German = sd(ss_1), # Standard deviation of German translation
                  mean_English = mean(ss_2), # Mean of English item
```

```
                    sd_English = sd(ss_2), # Standard deviation of English item
                    mean_diff = as.numeric(B_output[["stats"]][1,1]), # Mean of mu difference
                    sd_diff = as.numeric(B_output[["stats"]][1,2]), # Standard deviation
                    HDIlo = as.numeric(B_output[["stats"]][1,5]), # HDIlo
                    HDIup = as.numeric(B_output[["stats"]][1,6]), # HDIup
                    n_eff = as.numeric(B_output[["stats"]][1,16]),# n_eff
                    Rhat = as.numeric(B_output[["stats"]][1,15]), # Rhat
                    P_posterior = max(B_output[["stats"]][1,8],   # %<comp
                                      B_output[["stats"]][1,7]),  # %>comp
                    zero_excl = ifelse((as.numeric(B_output[["stats"]][1,5])>0)   # HDIlo
                                       | (as.numeric(B_output[["stats"]][1,6])<0),# HDIup
                                       '*','')
#add  "*"  marker if the low bound of HDI is large than zero,
# or the upper bound is smaller than zero
  )
return(l)  # Line 1 in the bayes.t.test output of mu_diff
}
```

With the function *getBAYES* defined, we now go examine for ASAQ item the difference between German and English scores.

```
item_list <- data.frame(Item=character(),ID=double(),mean_German=double(),
                        sd_German=double(),mean_English=double(),sd_English=double(),
                        mean_diff=double(),sd_diff=double(),HDIlo=double(),
                        HDIup=double(),zero_excl=character())
# Initialize output of Items with credible bias indication

set.seed(1) # Make sure that estimations of Bayesian analyses remain the same
n <- ncol(d_total)
# Numbers of all columns in d_total, i.e. English and German scores combined
German_column_offset <- n /2
# Offset for the column position of the first German ASAQ items

for (i in 1:90)
# Go step by step to 90 ASA questionnaire items
{
  score_German <- d_total[,i+ German_column_offset] # German scores
  score_English <- d_total[,i] # English item scores
  fit <- bayes.t.test(score_German, score_English, paired = TRUE)
  # conduct a Bayesian paired t-test on the German and English score of ASAQ item
```

```
  item_list <- rbind(item_list, getBAYES(i, score_German, score_English, fit))
  # store results from Bayesian analysis in a list to print later
}

# Print results
item_list$Item = colnames(select(d_total,HLA1:UAI4))
# Add item name code
pander(select(item_list,ID,mean_German,sd_German,mean_English,sd_English,Item),
       caption = "Items with credible bias indication (Part 1)")
```

Table 10: Items with credible bias indication (Part 1)

ID	mean_German	sd_German	mean_English	sd_English	Item
1	-1.153	1.976	-1.319	2.082	HLA1
2	-1.319	2.068	-1.306	2.074	HLA2
3	-1.069	2.065	-1.014	2.133	HLA3
4	-1.167	2.021	-1.014	2.017	HLA4
5	-0.7083	1.909	-0.5417	1.993	HLB1
6	-0.05556	1.868	-0.05556	1.883	HLB2
7	0.02778	2.028	-0.3333	2.116	HLB3
8	-0.5694	1.898	-0.625	1.857	HLB4
9	-0.1667	2.007	-0.1667	1.943	HLB5
10	-1.153	1.962	-0.9861	2.146	NA1
11	-0.2917	1.953	-0.5	2.049	NA2
12	-0.5972	2.18	-0.6667	2.182	NA3
13	-0.4444	1.971	-0.5556	2.013	NA4
14	0.5694	1.626	0.4167	1.774	NA5
15	-1.361	2.138	-1.319	2.122	NB1
16	0.1944	1.82	-0.08333	1.766	NB2
17	0.125	2.007	-0.01389	2.052	NB3
18	1.333	1.454	1.181	1.457	AAS1
19	1.306	1.553	1.25	1.381	AAS2
20	1.208	1.547	1.278	1.324	AAS3
21	1.389	1.369	1.444	1.299	AU1
22	1.028	1.601	1.264	1.565	AU2
23	1.542	1.373	1.653	1.128	AU3
24	1.542	1.383	1.347	1.247	PF1
25	1.444	1.5	1.514	1.035	PF2

ID	mean_German	sd_German	mean_English	sd_English	Item
26	1.194	1.37	1.333	1.163	PF3
27	0.3611	1.779	0.5556	1.727	AL1
28	0.8472	1.789	0.75	1.79	AL2
29	1.125	1.913	1.431	1.806	R_AL3
30	1.667	1.175	1.694	1.043	AL4
31	0	2.13	-0.4306	2.041	AL5
32	0.7083	1.707	-0.3889	1.804	AS1
33	1.111	1.588	0.5972	1.589	AS2
34	0.7917	1.744	0.7639	1.858	AS3
35	-0.08333	1.904	-0.2778	1.893	APP1
36	-0.5	2.062	-0.1111	1.896	R_APP2
37	-1.111	1.896	-1.181	1.894	APP3
38	1.486	1.363	1.542	1.244	UAA1
39	1.556	1.161	1.417	1.33	UAA2
40	0.8472	1.948	1.292	1.699	R_UAA3
41	1	1.823	0.8056	1.962	R_AE1
42	1.319	1.573	1.389	1.478	AE2
43	1.417	1.47	1.514	1.363	AE3
44	1.181	1.731	1.653	1.503	R_AE4
45	2.222	0.8429	1.875	1.363	UE1
46	2	1.007	1.972	0.9341	UE2
47	1.681	1.432	1.111	1.757	UE3
48	-0.2917	1.665	-0.3611	1.495	UT1
49	1.111	1.24	0.8194	1.336	UT2
50	0.7222	1.324	0.5694	1.626	UT3
51	0.04167	1.842	-0.1528	1.821	UAL1
52	0.02778	1.678	0.3056	1.692	UAL2
53	-0.02778	1.784	-0.4861	1.8	UAL3
54	0.7639	1.496	1.236	1.409	UAL4
55	0.3056	1.58	0.5694	1.5	UAL5
56	1.181	1.377	1.028	1.627	UAL6
57	1.75	1.402	2	1.187	AA1
58	1.681	1.072	1.514	1.394	AA2
59	2	1.061	2	1.233	AA3
60	1.958	1.272	1.944	1.32	R_AC1
61	1.931	1.248	1.958	1.144	R_AC2
62	1.667	1.374	1.5	1.463	R_AC3
63	1.653	1.602	1.708	1.56	R_AC4

ID	mean_German	sd_German	mean_English	sd_English	Item
64	0.2778	1.738	0.3889	1.534	AI1
65	0.2917	1.772	0.5694	1.694	AI2
66	0.9722	1.838	1.083	1.766	R_AI3
67	-0.375	1.682	-0.2917	1.748	AI4
68	1.417	1.527	1.361	1.613	AT1
69	1.375	1.551	1.278	1.475	AT2
70	1.681	1.555	1.708	1.665	R_AT3
71	-0.09722	1.737	-0.125	1.703	SP1
72	-0.7083	1.674	-0.6806	1.83	SP2
73	-1.097	1.745	-1.111	1.675	SP3
74	0.2361	1.468	0.2361	1.534	IIS1
75	0.25	1.508	0.4167	1.59	IIS2
76	0.2083	1.609	0.2083	1.768	IIS3
77	0.1111	1.588	0.1944	1.516	IIS4
78	-0.5	1.936	-1.014	1.804	AEI1
79	-1.069	2.009	-0.9444	1.822	AEI2
80	-0.1667	2.09	-0.7083	1.887	R_AEI3
81	-0.6111	1.954	-0.625	1.857	AEI4
82	-1.111	1.976	-1.111	1.873	R_AEI5
83	1.333	1.434	0.8472	1.859	UEP1
84	0.5556	1.669	0.125	1.891	UEP2
85	1.194	1.633	1.069	1.639	UEP3
86	0.9167	1.882	0.7778	2.016	UEP4
87	0.875	1.83	0.8333	1.884	UAI1
88	1.375	1.409	0.9583	1.326	UAI2
89	1.25	1.34	1.528	1.414	UAI3
90	0.3056	1.933	0.2778	1.809	UAI4

```
pander(select(item_list,ID,mean_diff,sd_diff,HDIlo,HDIup,Item),
       caption = "Items with credible bias indication (Part 2)")
```

Table 11: Items with credible bias indication (Part 2)

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
1	-4.914e-07	0.0001633	-0.0003248	0.0003181	HLA1
2	1.285e-06	0.0001002	-0.0002038	0.0001879	HLA2
3	-8.517e-07	0.0001772	-0.0003469	0.0003476	HLA3

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
4	-7.265e-06	0.0005441	-0.0008341	0.0008628	HLA4
5	-0.1391	0.1768	-0.4783	0.2149	HLB1
6	0.01798	0.1893	-0.3654	0.3781	HLB2
7	0.08815	0.1491	-0.1058	0.4446	HLB3
8	0.05089	0.1394	-0.2201	0.3267	HLB4
9	-0.004547	0.1476	-0.2925	0.2878	HLB5
10	0.003393	0.1133	-0.218	0.2407	NA1
11	0.2979	0.1458	0.02135	0.5886	NA2
12	0.03887	0.1584	-0.28	0.356	NA3
13	0.2413	0.1339	-0.02323	0.5015	NA4
14	0.155	0.159	-0.1625	0.4644	NA5
15	-8.627e-07	0.000107	-0.0002118	0.000207	NB1
16	0.2789	0.1479	-0.01856	0.5684	NB2
17	0.1595	0.1495	-0.1387	0.4495	NB3
18	0.04637	0.09268	-0.1002	0.2745	AAS1
19	3.977e-06	0.00023	-0.0004387	0.0004689	AAS2
20	-0.0443	0.1192	-0.2837	0.1856	AAS3
21	6.795e-07	0.0001653	-0.0003397	0.0003082	AU1
22	0.0005624	0.01847	-0.03684	0.04215	AU2
23	-6.183e-07	0.0001356	-0.0002642	0.0002711	AU3
24	0.1871	0.1156	-0.03539	0.4177	PF1
25	2.283e-06	0.000472	-0.0006836	0.0006892	PF2
26	-0.02556	0.1129	-0.2626	0.1904	PF3
27	-0.1674	0.1767	-0.5201	0.1791	AL1
28	-8.224e-07	9.663e-05	-0.0001894	0.0001897	AL2
29	2.799e-07	0.0004812	-0.0006503	0.0006288	R_AL3
30	4.156e-06	0.000363	-0.0005603	0.0005966	AL4
31	0.434	0.1208	0.2023	0.6756	AL5
32	1.132	0.2213	0.6934	1.56	AS1
33	0.4865	0.2025	0.08911	0.888	AS2
34	1.564e-05	0.0005292	-0.0008055	0.0008471	AS3
35	-0.01255	0.09685	-0.2049	0.188	APP1
36	-0.2761	0.1423	-0.5553	-0.004548	R_APP2
37	2.441e-07	0.0001018	-0.000199	0.0002023	APP3
38	1.199e-05	0.0004642	-0.0006311	0.0006737	UAA1
39	0.1416	0.1138	-0.08143	0.3608	UAA2
40	-0.3583	0.1679	-0.685	-0.02982	R_UAA3
41	7.434e-07	0.0002001	-0.0003736	0.0003823	R_AE1

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
42	3.84e-07	0.0001435	-0.0002822	0.0002845	AE2
43	-0.1081	0.1246	-0.3515	0.1402	AE3
44	-0.4268	0.1885	-0.7862	-0.04713	R_AE4
45	1.562e-06	0.0001328	-0.0002686	0.0002579	UE1
46	-1.094e-07	0.0001267	-0.0002468	0.000249	UE2
47	0.1485	0.1469	-0.08986	0.4596	UE3
48	0.0897	0.1231	-0.15	0.3382	UT1
49	0.05856	0.1194	-0.1196	0.356	UT2
50	0.1195	0.1432	-0.1601	0.4001	UT3
51	0.1587	0.1424	-0.1103	0.4465	UAL1
52	-0.1236	0.1903	-0.5057	0.2433	UAL2
53	0.3641	0.1681	0.02752	0.6859	UAL3
54	-0.4468	0.135	-0.7111	-0.1793	UAL4
55	-0.2144	0.1309	-0.4802	0.03236	UAL5
56	0.07504	0.1469	-0.2146	0.3606	UAL6
57	-1.085e-06	0.0001748	-0.0003404	0.0003515	AA1
58	-0.04004	0.1136	-0.2654	0.1929	AA2
59	5.628e-06	0.000344	-0.0005929	0.0005855	AA3
60	0.0224	0.106	-0.194	0.2452	R_AC1
61	-3.508e-07	0.0001206	-0.0002335	0.0002422	R_AC2
62	-4.972e-07	0.0001084	-0.0002133	0.000211	R_AC3
63	-5.136e-07	0.0001374	-0.0002777	0.0002653	R_AC4
64	-0.08299	0.1508	-0.3748	0.2174	AI1
65	-0.1947	0.1274	-0.4467	0.03403	AI2
66	-0.07953	0.1371	-0.3545	0.1891	R_AI3
67	-0.09015	0.1428	-0.3726	0.1865	AI4
68	-6.779e-06	0.0004003	-0.0006597	0.0006325	AT1
69	0.1	0.1103	-0.1158	0.3163	AT2
70	1.079e-06	0.000113	-0.0002186	0.0002266	R_AT3
71	0.01766	0.1326	-0.2393	0.2834	SP1
72	-0.01004	0.1517	-0.3074	0.2917	SP2
73	-2.607e-06	0.0005107	-0.0008808	0.0008561	SP3
74	5.196e-07	0.0001154	-0.0002267	0.0002272	IIS1
75	-6.519e-06	0.0004692	-0.0006947	0.0006943	IIS2
76	0.02526	0.1841	-0.3296	0.3918	IIS3
77	-5.109e-07	0.000126	-0.0002473	0.0002485	IIS4
78	0.4896	0.1785	0.1512	0.8532	AEI1
79	-0.06326	0.1231	-0.3384	0.16	AEI2

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
80	0.1242	0.1822	-0.2016	0.5145	R_AEI3
81	-0.04277	0.09991	-0.2584	0.1509	AEI4
82	0.02972	0.1055	-0.1753	0.2473	R_AEI5
83	0.1337	0.1182	-0.08489	0.377	UEP1
84	0.2927	0.1544	-0.002137	0.6033	UEP2
85	0.09116	0.09984	-0.09669	0.2904	UEP3
86	-2.927e-06	0.0002206	-0.0004439	0.0004221	UEP4
87	-0.002012	0.02238	-0.04488	0.04515	UAI1
88	0.4287	0.1854	0.05832	0.7872	UAI2
89	-0.2203	0.1352	-0.4867	0.03812	UAI3
90	0.1061	0.1266	-0.1273	0.3737	UAI4

```
pander(select(item_list,ID,n_eff,Rhat,P_posterior,zero_excl,Item),
       caption = "Items with credible bias indication (Part 3)")
```

Table 12: Items with credible bias indication (Part 3)

ID	n_eff	Rhat	P_posterior	zero_excl	Item
1	20879	1.001	0.5014		HLA1
2	19734	1	0.5071		HLA2
3	19832	1.001	0.5024		HLA3
4	8996	1.13	0.5001		HLA4
5	17834	1	0.7857		HLB1
6	17928	1	0.539		HLB2
7	342	1.061	0.7011		HLB3
8	19942	1	0.6423		HLB4
9	18648	1.001	0.5117		HLB5
10	12948	1	0.5186		NA1
11	14677	1	0.9847	*	NA2
12	16076	1	0.6013		NA3
13	17057	1	0.9658		NA4
14	18558	1	0.8377		NA5
15	20485	1	0.5041		NB1
16	17967	1.001	0.9693		NB2
17	17909	1	0.861		NB3
18	1840	1.005	0.6817		AAS1
19	20169	1.001	0.5087		AAS2

ID	n_eff	Rhat	P_posterior	zero_excl	Item
20	18548	1.001	0.6444		AAS3
21	20935	1.001	0.5003		AU1
22	2792	1.005	0.5124		AU2
23	20458	0.9999	0.501		AU3
24	18031	1	0.9489		PF1
25	20653	1.028	0.5025		PF2
26	10010	1	0.5742		PF3
27	16930	1	0.8321		AL1
28	21439	1	0.5053		AL2
29	15371	1.038	0.5022		R_AL3
30	18287	1.03	0.5001		AL4
31	18696	1	0.9997	*	AL5
32	17626	1	1	*	AS1
33	16981	1.001	0.9911	*	AS2
34	3257	1.131	0.5029		AS3
35	15543	1	0.5601		APP1
36	6892	1.001	0.9824	*	R_APP2
37	21085	1	0.5002		APP3
38	6185	1.001	0.5057		UAA1
39	17996	1	0.895		UAA2
40	17394	1.001	0.9857	*	R_UAA3
41	16017	1.003	0.5038		R_AE1
42	23507	1	0.5032		AE2
43	18903	1.001	0.8104		AE3
44	15771	1	0.9877	*	R_AE4
45	22792	1	0.5048		UE1
46	25006	1	0.5001		UE2
47	2315	1.018	0.8806		UE3
48	15222	1.001	0.7778		UT1
49	561	1.028	0.664		UT2
50	18399	1	0.8016		UT3
51	14377	1	0.8712		UAL1
52	8296	1	0.7415		UAL2
53	15980	1.001	0.9868	*	UAL3
54	17846	1	0.9996	*	UAL4
55	14572	1.001	0.9544		UAL5
56	12667	1	0.6964		UAL6
57	19608	1	0.5022		AA1

ID	n_eff	Rhat	P_posterior	zero_excl	Item
58	7674	1.003	0.6652		AA2
59	14027	1.031	0.502		AA3
60	13283	1.001	0.5939		R_AC1
61	20458	0.9999	0.502		R_AC2
62	24491	1	0.5033		R_AC3
63	20387	1	0.5009		R_AC4
64	17174	1	0.7114		AI1
65	5007	1	0.9511		AI2
66	16578	1	0.7248		R_AI3
67	18722	1	0.7362		AI4
68	6468	1.021	0.5008		AT1
69	18403	1	0.8202		AT2
70	21121	1.001	0.5041		R_AT3
71	18280	1	0.5513		SP1
72	18547	1	0.5232		SP2
73	6889	1.043	0.5033		SP3
74	19589	1.001	0.5024		IIS1
75	15927	1.025	0.5043		IIS2
76	16834	1	0.5583		IIS3
77	30309	1	0.5002		IIS4
78	17026	1.001	0.9967	*	AEI1
79	2979	1.012	0.6819		AEI2
80	4547	1.001	0.7482		R_AEI3
81	14231	1.003	0.6672		AEI4
82	13617	1.001	0.6124		R_AEI5
83	9075	1	0.885		UEP1
84	13954	1.001	0.9744		UEP2
85	10266	1	0.8258		UEP3
86	19864	1	0.5037		UEP4
87	2405	1.017	0.5277		UAI1
88	18810	1	0.9875	*	UAI2
89	8951	1	0.9559		UAI3
90	9734	1.001	0.8181		UAI4

```
# Calculate Grand mean information across the statistics obtained from 90 items
Variable <- c("mean_German","sd_German","mean_English","sd_English",
              "mean_diff","sd_diff","minimum_diff","maximum_diff",
```

```
              "n_zero_excl","percent_zero_excl")
# Define the names of the statistics

Grand_mean <- c(mean(item_list$mean_German),mean(item_list$sd_German),
                mean(item_list$mean_English),mean(item_list$sd_English),
                mean(abs(item_list$mean_diff)),mean(item_list$sd_diff),
                min(item_list$mean_diff),max(item_list$mean_diff),
                sum(item_list$zero_excl=="*"),round(sum(item_list$zero_excl=="*")
                /length(item_list$ID),digits=4)*100)
# Calculate the grand means of mean_German, sd_German, mean_English, sd_English,
# sd_diff, grand mean of the absolute value of mean differences, number of items
# with credible bias indication, and percentage of these items

# Print results
GrandMean <- cbind(Variable, Grand_mean)
pander(GrandMean, caption = "Grand mean of 90 items")
```

Table 13: Grand mean of 90 items

Variable	Grand_mean
mean_German	0.531481481481481
sd_German	1.66435067024113
mean_English	0.483950617283951
sd_English	1.66187273481957
mean_diff	0.108986344754192
sd_diff	0.0907279441720346
minimum_diff	-0.44684650500807
maximum_diff	1.13210495947046
n_zero_excl	11
percent_zero_excl	12.22

3.2.2 Mean score differences for 24 constructs and related dimensions

Next, step is to repeat the Bayesian *t*-test analysis but this time on a construct level. 95% credible interval of mean pairwise difference by Bayesian paired *t*-test was calculated for 24 constructs and related dimensions. It would reveal the variation between 24 English ASA constructs/dimensions and corresponding German translations. Before the *t*-test can be performed, we first have to calculate the construct score for each participant by taking the average score of the related ASAQ score. We have to do this both for the English and the German version of the ASAQ.

```r
con_list<-data.frame(Construct=character(),ID=double(),mean_German=double(),
                     sd_German=double(),mean_English=double(),sd_English=double(),
                     mean_diff=double(),sd_diff=double(),mean_diff=double(),
                     HDIlo=double(),HDIup=double(),zero_excl=character())
# Initialize output of Constructs/dimensions with credible bias indication

n <- ncol(d_total)
# Numbers of all columns in d_total, i.e. English and German scores combined
German_column_offset <- n /2
# Offset for the column position of the first German ASAQ items

for(p in 1:24)
# Go step by step to 24 constructs/dimensions
{
  i = h[p,1]
  # The column with the first English ASAQ item of the construct/dimension
  j = i+ German_column_offset
  # The column with the first German ASAQ item of the construct/dimension
  k = h[p,2] # The number of columns/items of the construct/dimension
  s_German <- data.frame(d_total[,j:(j+k-1)]) # Select German scores
  s_English <- data.frame(d_total[,i:(i+k-1)]) # Select English scores
  average_s_German <- data.frame(rowMeans(s_German))
  # German score means for each construct/dimension per participant
  average_s_English <- data.frame(rowMeans(s_English))
  # English score means for each construct/dimension per participant
  colnames(average_s_German) <- c("score")
  # Rename German mean column
  colnames(average_s_English) <- c("score")
  # Rename English mean column
  score <- data.frame(cbind(average_s_German,average_s_English))
  # Combine averaged scores of German and English constructs/dimensions
  score_German <- score[,1]
  # Select averaged scores of each German construct/dimension,
  # make sure data format is suitable for Bayesian paired t-test
  score_English <- score[,2]
  # Select averaged scores of each English construct/dimension,
  # make sure data format is suitable for Bayesian paired t-test
  fit <- bayes.t.test(score_German,score_English, paired = TRUE)
  # Conduct Bayesian t-test
```

```
  con_list <- rbind(con_list,getBAYES(p,score_German,score_English,fit))
  # Call function 'getBAYES' to obtain relevant information
  # from Bayesian t-test output and add result to output list
}

# Print results
con_list$Construct=c('HLA','HLB','NA','NB','AAS','AU','PF','AL','AS','APP',
'UAA','AE','UE','UT','UAL','AA','AC','AI','AT','SP','IIS','AEI','UEP','UAI')
# Add construct/dimension name code
pander(select(con_list,ID,mean_German,sd_German,mean_English,sd_English,Construct),
       caption = "Constructs/dimensions with credible bias indication (Part 1)")
```

Table 14: Constructs/dimensions with credible bias indication (Part 1)

ID	mean_German	sd_German	mean_English	sd_English	Construct
1	-1.177	1.923	-1.163	1.905	HLA
2	-0.2944	1.758	-0.3444	1.722	HLB
3	-0.3833	1.498	-0.4583	1.593	NA
4	-0.3472	1.694	-0.4722	1.681	NB
5	1.282	1.329	1.236	1.235	AAS
6	1.319	1.222	1.454	1.115	AU
7	1.394	1.147	1.398	0.9305	PF
8	0.8	1.417	0.8	1.352	AL
9	0.8704	1.404	0.3241	1.45	AS
10	-0.5648	1.558	-0.5231	1.56	APP
11	1.296	1.12	1.417	1.082	UAA
12	1.229	1.332	1.34	1.182	AE
13	1.968	0.8579	1.653	0.9999	UE
14	0.5139	1.12	0.3426	1.192	UT
15	0.3819	1.048	0.4167	1.073	UAL
16	1.81	0.9308	1.838	0.9674	AA
17	1.802	1.116	1.778	1.118	AC
18	0.2917	1.215	0.4375	1.287	AI
19	1.491	1.422	1.449	1.424	AT
20	-0.6343	1.397	-0.6389	1.488	SP
21	0.2014	1.21	0.2639	1.298	IIS
22	-0.6917	1.663	-0.8806	1.636	AEI
23	1	1.326	0.7049	1.523	UEP

ID	mean_German	sd_German	mean_English	sd_English	Construct
24	0.9514	1.066	0.8993	1.106	UAI

```
pander(select(con_list,ID,mean_diff,sd_diff,HDIlo,HDIup,Construct),
       caption = "Constructs/dimensions with credible bias indication (Part 2)")
```

Table 15: Constructs/dimensions with credible bias indication (Part 2)

ID	mean_diff	sd_diff	HDIlo	HDIup	Construct
1	-0.01636	0.09587	-0.2036	0.1749	HLA
2	0.03736	0.09469	-0.1482	0.2227	HLB
3	0.1644	0.08831	-0.004479	0.3454	NA
4	0.1372	0.08483	-0.03008	0.3027	NB
5	0.1048	0.1001	-0.09632	0.297	AAS
6	-0.1297	0.09254	-0.3124	0.05009	AU
7	0.0001719	0.09211	-0.1807	0.1794	PF
8	0.007056	0.06101	-0.1138	0.1263	AL
9	0.5203	0.137	0.2611	0.7985	AS
10	-0.0499	0.09854	-0.2462	0.1427	APP
11	-0.12	0.0957	-0.3111	0.06505	UAA
12	-0.1024	0.08774	-0.279	0.06596	AE
13	0.2787	0.104	0.07542	0.4832	UE
14	0.1176	0.1029	-0.07905	0.3239	UT
15	-0.03101	0.07408	-0.1725	0.1193	UAL
16	-0.0397	0.08686	-0.209	0.1296	AA
17	0.05154	0.06824	-0.0805	0.1875	AC
18	-0.1395	0.09203	-0.3202	0.04172	AI
19	0.05385	0.06399	-0.07057	0.1796	AT
20	-0.002784	0.09589	-0.1914	0.1863	SP
21	-0.1017	0.06486	-0.2291	0.02697	IIS
22	0.1018	0.08923	-0.07433	0.2733	AEI
23	0.208	0.0878	0.03981	0.3841	UEP
24	0.0502	0.07396	-0.09643	0.1935	UAI

```
pander(select(con_list,ID,n_eff,Rhat,P_posterior,zero_excl,Construct),
       caption = "Constructs/dimensions with credible bias indication (Part 3)")
```

Table 16: Constructs/dimensions with credible bias indication (Part 3)

ID	n_eff	Rhat	P_posterior	zero_excl	Construct
1	19560	1	0.5692		HLA
2	17847	1	0.6516		HLB
3	16690	1	0.9669		NA
4	18375	1	0.9478		NB
5	11214	1	0.8533		AAS
6	17986	1.001	0.9212		AU
7	19214	1	0.5022		PF
8	19085	1	0.5479		AL
9	19023	1	0.9999	*	AS
10	18616	1	0.6979		APP
11	19365	1	0.8966		UAA
12	17457	1	0.881		AE
13	12553	1	0.9969	*	UE
14	8439	1.001	0.876		UT
15	17944	1	0.6625		UAL
16	17759	1.001	0.6809		AA
17	18841	1	0.7794		AC
18	18602	1	0.9352		AI
19	17551	0.9999	0.8049		AT
20	18727	1	0.5113		SP
21	16146	1	0.9405		IIS
22	5780	1.001	0.8778		AEI
23	14001	1.001	0.9927	*	UEP
24	19261	0.9999	0.7524		UAI

```
# Determine grand (abs) means
Variable <- c("mean_German","sd_German","mean_English","sd_English",
              "mean_diff","sd_diff","minimum_diff","maximum_diff",
              "n_zero_excl","percent_zero_excl")
Grand_mean <- c(mean(con_list$mean_German),mean(con_list$sd_German),
                mean(con_list$mean_English),mean(con_list$sd_English),
                mean(abs(con_list$mean_diff)),mean(con_list$sd_diff),
                min(con_list$mean_diff),max(con_list$mean_diff),
                sum(con_list$zero_excl=="*"),round(sum(con_list$zero_excl=="*")
                /length(con_list$ID),digits=4)*100)
```

```
GrandMean <- cbind(Variable, Grand_mean)
# Calculate grand mean of mean_German, sd_German, mean_English, sd_English,
# sd_diff, grand mean of the absolute value of mean differences, number of
# constructs/dimensions with credible bias indication, and percentage of these
# constructs/dimensions
pander(GrandMean, caption = "Grand mean of 24 constructs/dimensions")
```

Table 17: Grand mean of 24 constructs/dimensions

Variable	Grand_mean
mean_German	0.604552469135802
sd_German	1.32387543203055
mean_English	0.552941743827161
sd_English	1.32997687991564
mean_diff	0.106920253828106
sd_diff	0.0888430384806325
minimum_diff	-0.139464658861835
maximum_diff	0.52025594974149
n_zero_excl	3
percent_zero_excl	12.5

3.2.3 Mean score differences between English and German short version of ASA questionnaire

As with ICC, we also conduct again difference analysis for representative ASAQ items in short version of ASAQ.

```
rep_list<-data.frame(Item=character(),ID=double(),mean_German=double(),
                     sd_German=double(),mean_English=double(),sd_English=double(),
                     mean_diff=double(),sd_diff=double(),HDIlo=double(),
                     HDIup=double(),zero_excl=character())
# Initialize output of Representative items with credible bias indication

n <- ncol(ss) # Numbers of all columns in ss
English_column_offset <- n /2

for (i in 1:24)
# Go step by step to 24 representative items of the ASA questionnaire
{
```

```
  score_German <- as.numeric(ss[,i]) # Select German scores
  score_English <- as.numeric(ss[,i+ English_column_offset]) # Select English scores
  fit<- bayes.t.test(score_German, score_English, paired = TRUE)
  rep_list <- rbind(rep_list, getBAYES(i, score_German, score_English, fit))
}

# Print results
rep_list$Item <- c('HLA2','HLB5','NA4','NB3','AAS1','AU1','PF1','AL2',
                   'AS1','APP1','UAA1','R_AE1','UE2','UT3','UAL1','AA2',
                   'R_AC1','R_AI3','AT1','SP2','IIS2','R_AEI3','UEP3','UAI4')
# Add item name code
pander(select(rep_list,ID,mean_German,sd_German,mean_English,sd_English,Item),
       caption = "Representative items with credible bias indication (Part 1)")
```

Table 18: Representative items with credible bias indication (Part 1)

ID	mean_German	sd_German	mean_English	sd_English	Item
1	-1.319	2.068	-1.306	2.074	HLA2
2	-0.1667	2.007	-0.1667	1.943	HLB5
3	-0.4444	1.971	-0.5556	2.013	NA4
4	0.125	2.007	-0.01389	2.052	NB3
5	1.333	1.454	1.181	1.457	AAS1
6	1.389	1.369	1.444	1.299	AU1
7	1.542	1.383	1.347	1.247	PF1
8	0.8472	1.789	0.75	1.79	AL2
9	0.7083	1.707	-0.3889	1.804	AS1
10	-0.08333	1.904	-0.2778	1.893	APP1
11	1.486	1.363	1.542	1.244	UAA1
12	1	1.823	0.8056	1.962	R_AE1
13	2	1.007	1.972	0.9341	UE2
14	0.7222	1.324	0.5694	1.626	UT3
15	0.04167	1.842	-0.1528	1.821	UAL1
16	1.681	1.072	1.514	1.394	AA2
17	1.958	1.272	1.944	1.32	R_AC1
18	0.9722	1.838	1.083	1.766	R_AI3
19	1.417	1.527	1.361	1.613	AT1
20	-0.7083	1.674	-0.6806	1.83	SP2
21	0.25	1.508	0.4167	1.59	IIS2

ID	mean_German	sd_German	mean_English	sd_English	Item
22	-0.1667	2.09	-0.7083	1.887	R_AEI3
23	1.194	1.633	1.069	1.639	UEP3
24	0.3056	1.933	0.2778	1.809	UAI4

```
pander(select(rep_list,ID,mean_diff,sd_diff,HDIlo,HDIup,Item),
       caption = "Representative items with credible bias indication (Part 2)")
```

Table 19: Representative items with credible bias indication (Part 2)

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
1	1.773e-07	9.999e-05	-0.0001949	0.0001963	HLA2
2	-0.004375	0.1478	-0.2893	0.2903	HLB5
3	0.2409	0.1342	-0.02199	0.5049	NA4
4	0.1607	0.1509	-0.1376	0.4545	NB3
5	0.04507	0.08933	-0.1033	0.2503	AAS1
6	-8.667e-07	0.0001633	-0.0003239	0.0003245	AU1
7	0.1884	0.1159	-0.04188	0.4145	PF1
8	-4.591e-07	9.756e-05	-0.0001963	0.0001868	AL2
9	1.13	0.2197	0.7019	1.56	AS1
10	-0.01256	0.09579	-0.2139	0.1751	APP1
11	6.367e-06	0.0004853	-0.0006613	0.0006497	UAA1
12	-1.354e-07	0.0001926	-0.0003839	0.0003657	R_AE1
13	8.7e-07	0.0001245	-0.0002416	0.0002458	UE2
14	0.1224	0.142	-0.16	0.4038	UT3
15	0.1584	0.1418	-0.1186	0.4363	UAL1
16	-0.04014	0.1142	-0.2704	0.1905	AA2
17	0.02148	0.1078	-0.1972	0.2481	R_AC1
18	-0.0787	0.135	-0.3507	0.1844	R_AI3
19	-2.082e-06	0.0003503	-0.0006175	0.0005974	AT1
20	-0.008639	0.1511	-0.31	0.284	SP2
21	2.134e-07	0.0004417	-0.0006989	0.0007248	IIS2
22	0.136	0.193	-0.212	0.5435	R_AEI3
23	0.09097	0.1004	-0.09702	0.2989	UEP3
24	0.1063	0.1259	-0.131	0.3664	UAI4

```
pander(select(rep_list,ID,n_eff,Rhat,P_posterior,zero_excl,Item),
       caption = "Representative items with credible bias indication (Part 3)")
```

Table 20: Representative items with credible bias indication (Part 3)

ID	n_eff	Rhat	P_posterior	zero_excl	Item
1	20463	1	0.5003		HLA2
2	17998	1	0.5121		HLB5
3	16972	1	0.9654		NA4
4	18008	1.001	0.8579		NB3
5	1370	1.004	0.6795		AAS1
6	21130	1	0.5015		AU1
7	16642	1	0.9479		PF1
8	23735	1	0.501		AL2
9	18022	1	1	*	AS1
10	16690	1	0.5595		APP1
11	11323	1.01	0.5007		UAA1
12	22611	1	0.5008		R_AE1
13	20624	1	0.5017		UE2
14	17571	1	0.8107		UT3
15	15507	1	0.8705		UAL1
16	6118	1.001	0.6677		AA2
17	15253	1	0.5866		R_AC1
18	18870	1	0.7226		R_AI3
19	23302	1.009	0.5005		AT1
20	18830	1	0.5242		SP2
21	5282	1.086	0.5017		IIS2
22	2383	1.007	0.7554		R_AEI3
23	11468	1	0.8255		UEP3
24	7832	1.001	0.8201		UAI4

```
# Calculate grand (abs) mean results
Variable <- c("mean_German","sd_German","mean_English","sd_English",
              "mean_diff","sd_diff","minimum_diff","maximum_diff",
              "n_zero_excl","percent_zero_excl")
Grand_mean <- c(mean(rep_list$mean_German),mean(rep_list$sd_German),
                mean(rep_list$mean_English),mean(rep_list$sd_English),
```

```
                mean(abs(rep_list$mean_diff)),mean(rep_list$sd_diff),
                min(rep_list$mean_diff),max(rep_list$mean_diff),
                sum(rep_list$zero_excl=="*"),round(sum(rep_list$zero_excl=="*")
                /length(rep_list$ID),digits=4)*100)
GrandMean <- cbind(Variable, Grand_mean)
# Calculate grand mean of mean_German, sd_German, mean_English, sd_English
# sd_diff, grand mean of the absolute value of mean differences, number of
# representative items with credible bias indication, and percentage of these items
pander(GrandMean, caption = "Grand mean of 24 representative items")
```

Table 21: Grand mean of 24 representative items

Variable	Grand_mean
mean_German	0.670138888888889
sd_German	1.64855479814062
mean_English	0.542824074074074
sd_English	1.66688938251241
mean_diff	0.106043058632004
sd_diff	0.0902868466339873
minimum_diff	-0.0786992379435328
maximum_diff	1.13007436594855
n_zero_excl	1
percent_zero_excl	4.17

3.3 Comparison of Human-ASA Interaction between Different Cultural Backgrounds

The results were reported in the subsection of Comparison of Human-ASA Interaction between Different Cultural Backgrounds. The analysis was based on human-ASA interaction evaluation of 532 mix international English-speaking participants in our previous study and 144 (72 per each half) bilingual participants with German mother tongue in this study, using the data file 'data_culture_EN_DE.sav'. We compared human-ASA interaction between these two cultural background populations mentioned above. Two-level linear regression model was implemented to explore construct and dimension score differences between two sample groups, with agent as random intercept to control for dependency of agent assignment. For the Bayesian analysis we used the rethinking package developed by Richard McElreath[1].

[1] https://www.rdocumentation.org/packages/rethinking/versions/2.13

```r
cul_list <- data.frame(ConstructID=double(),mean_Ger=double(),sd_Ger=double(),
                       mean_Eng=double(),sd_Eng=double(),mean_diff=double(),
                       sd_diff=double(),lo2_5=double(),
                       hi97_5=double(),n_eff=double(),Rhat4=double(),
                       P_posterior=double(),zero_excl=character())
#Initialize output list of Construct/dimension differences between two cultural groups

for(j in 1:24)
# Go step by step to 24 constructs/dimensions of the ASA questionnaire
{
  data_culture$Culture <- (data_culture$Culture * -1) +1
  d_c<-subset(data_culture, ConstructID==j, select=c(AgentID, Culture, Rating))
  # select scores data for ASAQ construct j

  # Define the model we fit on the data. This is a multilevel model,
  # with agent as random intercept to control for
  # dependency of agent assignment, and culture as fixed effect
  m <- ulam(
    alist(
      #Likelihood
      Rating ~ dnorm(mu, sigma),

      #Linear model
      mu <- a + a_Agent[AgentID] + c_cult*Culture,

      #Adaptive prior
      a_Agent[AgentID] ~ dnorm(0, sigma_agent),

      #Hyper prior
      sigma_agent ~ dcauchy(0, 1),

      #Fixed priors
      a ~ dnorm(0, 2),
      c_cult ~ dnorm(0, 1),
      sigma ~ dcauchy(0, 1)
    ), data = d_c, iter = 50000, chains = 4, cores = 4, log_lik = TRUE,
    control=list(adapt_delta=.99)
  )
```

```r
# Calculate posterior probability
post_samples <- extract.samples(m, 1e4)
# Extract 10000 samples from the posterior distribution
c_cult <- as.numeric(post_samples$c_cult)
H0_post <- subset(c_cult, c_cult>0)
# Select samples with positive posterior values (positive bias)

H0_post_p <- length(H0_post)/1e4
# Calculate probability of a positive bias
H1_post_p <- 1 - H0_post_p
# Probability of a negative bias

d_c_Ger <- subset(d_c, Culture == 1)
# Subset of only German mother tongue sample

d_c_Eng <- subset(d_c, Culture == 0)
# Subset of only Mixed international sample

o <- precis(m,  depth=2, prob=.95)
l <- data.frame(ConstructID = j,
                mean_Ger = mean(d_c_Ger$Rating),
                sd_Ger = sd(d_c_Ger$Rating),
                mean_Eng = mean(d_c_Eng$Rating),
                sd_Eng = sd(d_c_Eng$Rating),
                mean_diff = as.numeric(o$mean[17]),
                sd_diff = as.numeric(o$sd[17]),
                lo2_5 = as.numeric(o$`2.5%`[17]),
                hi97_5 = as.numeric(o$`97.5%`[17]),
                n_eff = as.numeric(o$n_eff[17]),
                Rhat4 = as.numeric(o$Rhat4[17]),
                P_posterior = max(H0_post_p,H1_post_p),
                zero_excl = ifelse((as.numeric(o$`2.5%`[17])>0)
                             | (as.numeric(o$`97.5%`[17])<0),
                             '*','')
                )
# Line 17 in the precis output, are the results related to c_cul coefficient
cul_list <- rbind(cul_list, l)
# Store results in a list to print later on
```

```
}
```

The last step is the print the results of the model analysis.

```
# Print results
cul_list$Construct=c('HLA','HLB','NA','NB','AAS','AU','PF','AL','AS','APP',
'UAA','AE','UE','UT','UAL','AA','AC','AI','AT','SP','IIS','AEI','UEP','UAI')
# Add construct/dimension name code
pander(select(cul_list,ConstructID,mean_Ger,sd_Ger,mean_Eng,sd_Eng,Construct),
       caption="Construct/dimension differences between two cultural groups (Part 1)")
```

Table 22: Construct/dimension differences between two cultural groups (Part 1)

ConstructID	mean_Ger	sd_Ger	mean_Eng	sd_Eng	Construct
1	-1.163	1.905	-0.7533	2.013	HLA
2	0.04398	1.602	-0.3444	1.722	HLB
3	-0.4583	1.593	-0.2429	1.487	NA
4	-0.2932	1.555	-0.4722	1.681	NB
5	1.236	1.235	1.346	1.221	AAS
6	1.234	1.192	1.454	1.115	AU
7	1.398	0.9305	1.306	1.121	PF
8	0.7699	1.4	0.8	1.352	AL
9	0.3241	1.45	0.3164	1.488	AS
10	0.1986	1.489	-0.5231	1.56	APP
11	1.417	1.082	1.311	1.183	UAA
12	1.252	1.225	1.34	1.182	AE
13	1.653	0.9999	1.812	1.009	UE
14	0.4311	1.211	0.3426	1.192	UT
15	0.4167	1.073	0.5125	1.146	UAL
16	1.654	1.156	1.838	0.9674	AA
17	1.778	1.118	1.549	1.067	AC
18	0.6852	1.349	0.4375	1.287	AI
19	1.449	1.424	1.431	1.335	AT
20	-0.1629	1.508	-0.6389	1.488	SP
21	0.2639	1.298	0.648	1.145	IIS
22	-0.6684	1.705	-0.8806	1.636	AEI
23	0.7049	1.523	0.6245	1.285	UEP
24	0.7946	1.202	0.8993	1.106	UAI

```
pander(select(cul_list,ConstructID,mean_diff,sd_diff,lo2_5,hi97_5,Construct),
       caption="Construct/dimension differences between two cultural groups (Part 2)")
```

Table 23: Construct/dimension differences between two cultural groups (Part 2)

ConstructID	mean_diff	sd_diff	lo2_5	hi97_5	Construct
1	-0.3098	0.1631	-0.6287	0.0107	HLA
2	0.3374	0.1732	-0.00315	0.676	HLB
3	-0.1788	0.1513	-0.4744	0.1173	NA
4	0.132	0.1616	-0.1837	0.449	NB
5	-0.1149	0.1415	-0.392	0.1629	AAS
6	-0.2016	0.1344	-0.4635	0.06166	AU
7	0.08222	0.1315	-0.1758	0.3415	PF
8	-0.01383	0.1501	-0.3091	0.2787	AL
9	0.004234	0.1654	-0.3191	0.3293	AS
10	0.6846	0.1601	0.3702	0.9979	APP
11	0.08802	0.1347	-0.1752	0.3527	UAA
12	-0.06004	0.136	-0.3282	0.2056	AE
13	-0.1735	0.1213	-0.4102	0.06466	UE
14	0.1103	0.1383	-0.1599	0.3802	UT
15	-0.1025	0.136	-0.3694	0.1651	UAL
16	-0.1582	0.138	-0.4274	0.1139	AA
17	0.2048	0.1269	-0.0442	0.4534	AC
18	0.2735	0.1484	-0.01626	0.5649	AI
19	-0.0164	0.1411	-0.2918	0.26	AT
20	0.4338	0.1747	0.09099	0.777	SP
21	-0.3871	0.1348	-0.6514	-0.1222	IIS
22	0.1403	0.1744	-0.2009	0.4836	AEI
23	0.1101	0.1413	-0.1677	0.3869	UEP
24	-0.1216	0.1376	-0.3922	0.1474	UAI

```
pander(select(cul_list,ConstructID,n_eff,Rhat4,P_posterior,zero_excl,Construct),
       caption="Construct/dimension differences between two cultural groups (Part 3)")
```

Table 24: Construct/dimension differences between two cultural groups (Part 3)

ConstructID	n_eff	Rhat4	P_posterior	zero_excl	Construct
1	47475	1	0.9728		HLA
2	70136	1	0.9749		HLB
3	62287	1	0.8759		NA
4	57073	1	0.7918		NB
5	95959	1	0.7957		AAS
6	61368	1	0.9309		AU
7	102170	1	0.7296		PF
8	59277	1	0.5397		AL
9	97705	1	0.5101		AS
10	52354	1	1	*	APP
11	102939	1	0.7492		UAA
12	66828	1	0.6687		AE
13	110066	1	0.9283		UE
14	67864	1	0.7864		UT
15	120266	1	0.784		UAL
16	82842	1	0.8758		AA
17	123640	1	0.9488		AC
18	71666	1	0.9696		AI
19	55882	1	0.5536		AT
20	69061	1	0.9946	*	SP
21	84237	1	0.9977	*	IIS
22	68001	1	0.7878		AEI
23	77081	1	0.7877		UEP
24	56299	1	0.8084		UAI

```
# Print grand means
Variable <- c("mean_Ger","sd_Ger","mean_Eng","sd_Eng","mean_diff","sd_diff",
              "minimum_diff","maximum_diff","n_zero_excl","percent_zero_excl")
Grand_mean <- c(mean(cul_list$mean_Ger),mean(cul_list$sd_Ger),
                mean(cul_list$mean_Eng),mean(cul_list$sd_Eng),
                mean(abs(cul_list$mean_diff)),mean(cul_list$sd_diff),
                min(cul_list$mean_diff),max(cul_list$mean_diff),
                sum(cul_list$zero_excl=="*"),round(sum(cul_list$zero_excl=="*")
                /length(cul_list$ConstructID),digits=4)*100)
GrandMean <- cbind(Variable, Grand_mean)
```

```
# Calculate grand mean of mean_Ger, sd_Ger, mean_Eng, sd_Eng
# sd_diff, grand mean of the absolute value of mean differences, number of
# constructs/dimensions with credible bias indication,
# and percentage of these constructs/dimensions
pander(GrandMean, caption = "Grand mean of 24 constructs/dimensions between two cultural groups")
```

Table 25: Grand mean of 24 constructs/dimensions between two cultural groups

Variable	Grand_mean
mean_Ger	0.623231400259909
sd_Ger	1.34268276472619
mean_Eng	0.588012348579783
sd_Eng	1.32452566260208
mean_diff	0.184986958827542
sd_diff	0.146478689295446
minimum_diff	-0.38713100617282
maximum_diff	0.684623196611
n_zero_excl	3
percent_zero_excl	12.5

References

Cicchetti, Domenic V. 1994. "Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology." *Psychological Assessment* 6 (4): 284. https://doi.org/10.1037/1040-3590.6.4.284.

Finch, W Holmes, Jocelyn E Bolin, and Ken Kelley. 2019. *Multilevel Modeling Using r*. Crc Press.