German and Dutch Translations of the Artificial-Social-Agent Questionnaire Instrument for Evaluating Human-Agent Interactions

Correlation and Variation between English and German ASA Questionnaire

Nele Albers, Boleslav A. Khodakov

18 June, 2024

Contents

Introduction **1**

Load packages **1**

Define constants **2**

Load preprocessed data **2**

Analyses results **2**
- Correlation between English and German ASA Questionnaire . . . 2
  - ICC values for 90 items . . . 2
  - Removing English Prefix 'Q_E_' . . . 7
  - ICC values for 24 constructs and related dimensions . . . 7
  - ICC values between English and German scores for the short version of ASA questionnaire . . . 10
- Variation Between English and German ASA Questionnaire . . . 13
  - Mean score differences for 90 items . . . 13
  - Mean score differences for 24 constructs and related dimensions . . . 20
  - Mean score differences between English and German short version of ASA questionnaire . . . 24

References **27**

Introduction

This document presents statistical analyses of the correlation and variation between the English and German ASA questionnaires for the item level, construct/dimension level, and short version of the ASA questionnaire. The code is based on the one by Li et al. (2023).

Required files: summative_first_half_transformed_german.sav, summative_second_half_transformed_german.sav.

Authored by Nele Albers, Andrea Bönsch, Jonathan Ehret, Boleslav A. Khodakov, and Willem-Paul Brinkman.

Load packages

Let's load the packages that we need.

```r
library(BayesianFirstAid)  # Run Bayesian t-test
library(car) # Package linear regression
library(dplyr) # Use select function
library(foreign) # Open various data files
library(formatR)  # For formatting
library(haven) # Use read_sav fuction
library(knitr) # Get markdown file
library(nlme)    # Run multilevel linear models
library(pander) # For pandering tables
library(rethinking) # Run ulam

panderOptions("table.alignment.default","left")
panderOptions('round',2)
```

Define constants

And let's define a few constants that we use throughout.

```r
NUM_ITEMS_SHORT_ASAQ = 24
NUM_ITEMS_FULL_ASAQ = 90
NUM_PRECISION = 4  # Precision for determining ICC classification
```

Load preprocessed data

The input data used in the analysis are the preprocessed data files 'summative_first_half_transformed_german.sav' and 'summative_second_half_transformed_german.sav'.

Let's first load the data from the first half.

```r
# Load data for first half
d1 <- data.frame(read_sav("summative_first_half_transformed_german.sav"))
```

And we also load the data from the second half of the questionnaire.

```r
# Load data for second half
d2 <- data.frame(read_sav("summative_second_half_transformed_german.sav"))
```

Analyses results

Correlation between English and German ASA Questionnaire

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 the 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.

```r
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 120 data points, which we combine in a 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
{
  # Row names that represent the ID number of each participant
  id<-rownames(s_2)
  # Transform German scores from wide format to long format and label as 1
  score_German<- data.frame(id, s_1, language= 1)
   # Transform English scores from wide format to long format and label as 2
  Score_English<- data.frame(id, s_2, language= 2)
  # Combine German and English scores in the long format
  Score_total <- rbind(score_German, Score_English)
  # Linear mixed-effects model with a fixed intercept and
  # a random intercept of participant's ID number
  m0 <- lme(score ~ 1, data = Score_total, random = ~1|id, method = "ML")
  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 through 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.

First we define a function to calculate the ICC values.

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

  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])

    # Calculate 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)
}
```

And then we use this function to compute the ICC values.

```
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	0.83	Q_E_HLA1
2	0.91	Q_E_HLA2
3	0.82	Q_E_HLA3
4	0.76	Q_E_HLA4
5	0.72	Q_E_HLB1
6	0.58	Q_E_HLB2
7	0.74	Q_E_HLB3
8	0.76	Q_E_HLB4
9	0.76	Q_E_HLB5
10	0.73	Q_E_NA1
11	0.67	Q_E_NA2
12	0.71	Q_E_NA3
13	0.67	Q_E_NA4
14	0.7	Q_E_NA5
15	0.91	Q_E_NB1
16	0.76	Q_E_NB2
17	0.76	Q_E_NB3
18	0.62	Q_E_AAS1
19	0.62	Q_E_AAS2
20	0.61	Q_E_AAS3
21	0.61	Q_E_AU1
22	0.49	Q_E_AU2
23	0.65	Q_E_AU3
24	0.75	Q_E_PF1
25	0.57	Q_E_PF2
26	0.52	Q_E_PF3
27	0.6	Q_E_AL1

i	icc	Item
28	0.89	Q_E_AL2
29	0.85	Q_E_R_AL3
30	0.61	Q_E_AL4
31	0.87	Q_E_AL5
32	0.3	Q_E_AS1
33	0.33	Q_E_AS2
34	0.67	Q_E_AS3
35	0.62	Q_E_APP1
36	0.75	Q_E_R_APP2
37	0.88	Q_E_APP3
38	0.73	Q_E_UAA1
39	0.76	Q_E_UAA2
40	0.38	Q_E_R_UAA3
41	0.78	Q_E_R_AE1
42	0.79	Q_E_AE2
43	0.73	Q_E_AE3
44	0.58	Q_E_R_AE4
45	0.58	Q_E_UE1
46	0.46	Q_E_UE2
47	0.41	Q_E_UE3
48	0.62	Q_E_UT1
49	0.44	Q_E_UT2
50	0.63	Q_E_UT3
51	0.76	Q_E_UAL1
52	0.4	Q_E_UAL2
53	0.58	Q_E_UAL3
54	0.62	Q_E_UAL4
55	0.66	Q_E_UAL5
56	0.65	Q_E_UAL6
57	0.57	Q_E_AA1
58	0.44	Q_E_AA2
59	0.7	Q_E_AA3
60	0.65	Q_E_R_AC1
61	0.69	Q_E_R_AC2
62	0.7	Q_E_R_AC3
63	0.74	Q_E_R_AC4
64	0.61	Q_E_AI1
65	0.67	Q_E_AI2
66	0.59	Q_E_R_AI3
67	0.7	Q_E_AI4
68	0.78	Q_E_AT1
69	0.74	Q_E_AT2
70	0.86	Q_E_R_AT3
71	0.8	Q_E_SP1
72	0.73	Q_E_SP2
73	0.57	Q_E_SP3
74	0.73	Q_E_IIS1
75	0.76	Q_E_IIS2
76	0.56	Q_E_IIS3
77	0.74	Q_E_IIS4
78	0.71	Q_E_AEI1
79	0.78	Q_E_AEI2

i	icc	Item
80	0.63	Q_E_R_AEI3
81	0.74	Q_E_AEI4
82	0.77	Q_E_R_AEI5
83	0.53	Q_E_UEP1
84	0.58	Q_E_UEP2
85	0.63	Q_E_UEP3
86	0.62	Q_E_UEP4
87	0.68	Q_E_UAI1
88	0.38	Q_E_UAI2
89	0.54	Q_E_UAI3
90	0.69	Q_E_UAI4

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(l_ICC$icc),digits=2),round(sd(l_ICC$icc),digits=2),
           round(min(l_ICC$icc),digits=2),round(max(l_ICC$icc),digits=2))
# 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	0.66
SD	0.13
Minimum	0.3
Maximum	0.91

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 each classification category.

```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 = NUM_PRECISION)  # Round ICC values
Number <- c(length(l_ICC[which(round_ICC >= 0.75 & round_ICC <=
    1), ]$icc), length(l_ICC[which(round_ICC >= 0.6 & round_ICC <=
    0.7499), ]$icc), length(l_ICC[which(round_ICC >= 0.4 & round_ICC <=
    0.5999), ]$icc), length(l_ICC[which(round_ICC >= 0 & round_ICC <=
    0.3999), ]$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)
```

```r
# 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	25	27.78
Good	0.60-0.74	41	45.56
Fair	0.40-0.59	20	22.22
Poor	0-0.39	4	4.44

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.

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

ICC values for 24 constructs and related dimensions

We combined the scores of Construct 1-8 (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.

```r
German_column_offset = ncol(d_total)/2

# 'i' is a vector with the column number of the first
# English version of item of the construct/dimension
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"))

# '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
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)))
```

```
# 'k2' is a vector with the number of questionnaire items
# of each construct/dimension for Construct 9-19
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)))

# Combine k1 and k2 into a single vector with the number of
# questionnaire items of each construct/dimension of the
# entire ASAQ
k = c(k1, k2)
# 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
h <- cbind.data.frame(i, k)

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

# Go step by step to 24 constructs/dimensions of the ASA
# questionnaire
for (p in 1:NUM_ITEMS_SHORT_ASAQ) {
    # Column number of the first ASAQ item in English of
    # the construct/dimension
    i <- h[p, 1]
    # The column number of the first ASAQ item in the
    # German version of the construct/dimension
    j <- i + German_column_offset
    # The number of ASAQ items associate to the
    # construct/dimension
    k <- h[p, 2]
    # Select the scores of all the ASAQ items in German
    # associated with the construct/dimension
    s_German <- data.frame(d_total[, j:(j + k - 1)])
    # Select the score of all the ASAQ items in English
    # associated with the construct/dimension
    s_English <- data.frame(d_total[, i:(i + k - 1)])
    # Calculate the mean score of ASAQ items in German
    # associated with the construct/dimension per
    # participant
    average_s_German <- data.frame(rowMeans(s_German))
    # Doing the same but now for English version of the
    # items
    average_s_English <- data.frame(rowMeans(s_English))
    colnames(average_s_German) <- c("score")  # Rename German mean column
    colnames(average_s_English) <- c("score")  # Rename English mean column
    # Call function 'getLME' for ICC value calculation
```

```r
    l_ICC <- rbind(l_ICC, data.frame(p, icc = getLME(average_s_German,
        average_s_English)))
}
# Add construct/dimension name code
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")
pander(l_ICC, caption = "ICC values for 24 constructs/dimensions")
```

Table 4: ICC values for 24 constructs/dimensions

p	icc	Construct
1	0.92	HLA
2	0.88	HLB
3	0.87	NA
4	0.91	NB
5	0.75	AAS
6	0.77	AU
7	0.74	PF
8	0.94	AL
9	0.58	AS
10	0.87	APP
11	0.77	UAA
12	0.86	AE
13	0.6	UE
14	0.74	UT
15	0.81	UAL
16	0.71	AA
17	0.83	AC
18	0.8	AI
19	0.92	AT
20	0.84	SP
21	0.85	IIS
22	0.91	AEI
23	0.81	UEP
24	0.8	UAI

```r
# Define the names of the statistics
Variable <- c("Grand_mean", "SD", "Minimum", "Maximum")
# Calculate the grand mean, standard deviation, minimum and
# maximum values of ICC values of 24 constructs/dimensions
Value <- c(round(mean(l_ICC$icc), digits = 2), round(sd(l_ICC$icc),
    digits = 2), round(min(l_ICC$icc), digits = 2), round(max(l_ICC$icc),
    digits = 2))
# Descriptive statistics of ICC values of 24
# constructs/dimensions
description <- cbind(Variable, Value)

# 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	0.81
SD	0.09
Minimum	0.58
Maximum	0.94

And we classify the resulting ICC values again based on (Cicchetti 1994).

```
# Categories of ICC classifications by Cicchetti (1994)
Classification <- c("Excellent", "Good", "Fair", "Poor")
ICC_Range <- c("0.75-1.00", "0.60-0.74", "0.40-0.59", "0-0.39")
n_item <- length(l_ICC$icc)  # Number of ICC values
round_ICC <- round(l_ICC$icc, digits = NUM_PRECISION)  # Round ICC values
# Calculate number of ICC values in classification category
Number <- c(length(l_ICC[which(round_ICC >= 0.75 & round_ICC <=
    1), ]$icc), length(l_ICC[which(round_ICC >= 0.6 & round_ICC <=
    0.7499), ]$icc), length(l_ICC[which(round_ICC >= 0.4 & round_ICC <=
    0.5999), ]$icc), length(l_ICC[which(round_ICC >= 0 & round_ICC <=
    0.3999), ]$icc))
# Calculate percentage 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)
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	19	79.17
Good	0.60-0.74	3	12.5
Fair	0.40-0.59	2	8.33
Poor	0-0.39	0	0

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.

```
# Select German versions of the 24 representative ASAQ items
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 English 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)
# Combine German and English scores
ss <- cbind(s_German,s_English)

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

# Initialize output of ICC values of 24 representative items
l_ICC <- data.frame(ID=double(), Item=character(), icc=double())
# Go step by step to 24 representative items of the ASA questionnaire
for (i in 1:NUM_ITEMS_SHORT_ASAQ)
{
  # Select German scores of the ASAQ item
  score_German <- data.frame(score=ss[,i])
  # Select English scores of the ASAQ item
  score_English <- data.frame(score=ss[,i+ English_column_offset])
  # Call function 'getLME' for ICC value calculation
  l_ICC <- rbind(l_ICC, data.frame (i, icc = getLME(score_German, score_English)))
}
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	0.91	HLA2
2	0.76	HLB5
3	0.67	NA4
4	0.76	NB3
5	0.62	AAS1
6	0.61	AU1
7	0.75	PF1
8	0.89	AL2
9	0.3	AS1
10	0.62	APP1
11	0.73	UAA1
12	0.78	R__AE1
13	0.46	UE2
14	0.63	UT3
15	0.76	UAL1
16	0.44	AA2
17	0.65	R__AC1
18	0.59	R__AI3
19	0.78	AT1
20	0.73	SP2
21	0.76	IIS2
22	0.63	R__AEI3
23	0.63	UEP3
24	0.69	UAI4

```
# Define the names of the statistics
Variable <- c("Grand_mean","SD","Minimum","Maximum")
```

```r
# Calculate the grand mean, standard deviation, minimum
# and maximum values of ICC values of 24 representative items
Value <- c(round(mean(l_ICC$icc),digits=2),round(sd(l_ICC$icc),digits=2),
           round(min(l_ICC$icc),digits=2),round(max(l_ICC$icc),digits=2))
# Descriptive statistics of ICC values of 24 representative items
description <- cbind(Variable, Value)

# 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	0.67
SD	0.14
Minimum	0.3
Maximum	0.91

And we classify the resulting ICC values again based on (Cicchetti 1994).

```r
# Categories of ICC classifications by Cicchetti (1994)
Classification <- c("Excellent", "Good", "Fair", "Poor")
ICC_Range <- c("0.75-1.00", "0.60-0.74", "0.40-0.59", "0-0.39")
n_item <- length(l_ICC$icc)  # Number of ICC values
round_ICC <- round(l_ICC$icc, digits = NUM_PRECISION)  # Round ICC values
# Calculate number of ICC values in classification category
Number <- c(length(l_ICC[which(round_ICC >= 0.75 & round_ICC <=
    1), ]$icc), length(l_ICC[which(round_ICC >= 0.6 & round_ICC <=
    0.7499), ]$icc), length(l_ICC[which(round_ICC >= 0.4 & round_ICC <=
    0.5999), ]$icc), length(l_ICC[which(round_ICC >= 0 & round_ICC <=
    0.3999), ]$icc))
# Calculate percentage 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)
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	9	37.5
Good	0.60-0.74	11	45.83
Fair	0.40-0.59	3	12.5
Poor	0-0.39	1	4.17

Variation Between English and German ASA Questionnaire

The mean score differences between the English and German questionnaires are estimates for the absolute accuracy in score equivalence between the two languages. 95% credible intervals of mean paired differences were calculated using a 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.

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.

```
# 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
getBAYES <-function(ID, ss_1, ss_2, B_output)
{ 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
                  # Add "*" marker if the low bound of HDI is large than zero,
                  # or the upper bound is smaller than zero
                  zero_excl = ifelse((as.numeric(B_output[["stats"]][1,5])>0)   # HDIlo
                                     | (as.numeric(B_output[["stats"]][1,6])<0),# HDIup
                                     '*','')
  )
return(l)
}
```

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

```
# Initialize output of Items with credible bias indication
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())

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

```r
# items
German_column_offset <- n/2

# Go step by step to 90 ASA questionnaire items
for (i in 1:NUM_ITEMS_FULL_ASAQ) {
    score_German <- d_total[, i + German_column_offset]  # German scores
    score_English <- d_total[, i]  # English item scores
    set.seed(1)  # Make sure that estimations of Bayesian analyses remain the same
    # Conduct a Bayesian paired t-test on the German and
    # English score of ASAQ item
    fit <- bayes.t.test(score_German, score_English, paired = TRUE)

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

# 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	-1.16	1.93	-1.29	1.98	HLA1
2	-1.32	2.05	-1.25	2.05	HLA2
3	-1.05	2.1	-1.07	2.08	HLA3
4	-1.2	1.93	-1.08	2.02	HLA4
5	-0.68	1.81	-0.62	1.85	HLB1
6	-0.05	1.81	-0.02	1.77	HLB2
7	-0.07	1.93	-0.38	2	HLB3
8	-0.55	1.77	-0.52	1.82	HLB4
9	-0.26	1.96	-0.15	1.85	HLB5
10	-1.35	1.91	-1.16	2	NA1
11	-0.38	1.94	-0.47	2.03	NA2
12	-0.8	2.12	-0.72	2.09	NA3
13	-0.5	1.91	-0.64	1.94	NA4
14	0.48	1.61	0.32	1.81	NA5
15	-1.51	2.06	-1.41	2.08	NB1
16	0.17	1.72	-0.02	1.71	NB2
17	0.02	1.92	-0.16	1.94	NB3
18	1.32	1.34	1.24	1.44	AAS1
19	1.26	1.49	1.22	1.35	AAS2
20	1.18	1.4	1.27	1.36	AAS3
21	1.35	1.33	1.51	1.24	AU1
22	1.24	1.43	1.32	1.4	AU2
23	1.57	1.26	1.64	1.23	AU3
24	1.3	1.4	1.15	1.25	PF1
25	1.43	1.46	1.39	1.22	PF2
26	1.11	1.35	1.22	1.3	PF3

ID	mean_German	sd_German	mean_English	sd_English	Item
27	0.28	1.72	0.36	1.69	AL1
28	0.61	1.83	0.62	1.76	AL2
29	0.94	1.94	1.25	1.85	R_AL3
30	1.64	1.17	1.58	1.06	AL4
31	-0.22	2.09	-0.51	2.07	AL5
32	0.57	1.58	-0.31	1.69	AS1
33	1.05	1.53	0.43	1.64	AS2
34	0.79	1.65	0.83	1.72	AS3
35	-0.22	1.78	-0.38	1.82	APP1
36	-0.46	2.03	-0.2	1.95	R_APP2
37	-1.06	1.93	-1.17	1.97	APP3
38	1.31	1.44	1.37	1.39	UAA1
39	1.44	1.3	1.3	1.44	UAA2
40	1.03	1.81	1.15	1.76	R_UAA3
41	0.81	1.77	0.72	1.85	R_AE1
42	1.17	1.53	1.18	1.46	AE2
43	1.24	1.58	1.32	1.46	AE3
44	1.18	1.68	1.56	1.51	R_AE4
45	2.02	1.12	1.75	1.46	UE1
46	1.76	1.2	1.77	1.26	UE2
47	1.72	1.46	1.23	1.68	UE3
48	-0.25	1.56	-0.28	1.5	UT1
49	1.06	1.28	0.96	1.25	UT2
50	0.71	1.4	0.52	1.61	UT3
51	-0.03	1.85	-0.17	1.8	UAL1
52	0.16	1.7	0.2	1.73	UAL2
53	0.15	1.76	-0.26	1.8	UAL3
54	0.91	1.5	1.23	1.41	UAL4
55	0.33	1.56	0.54	1.54	UAL5
56	0.93	1.59	1.02	1.64	UAL6
57	1.82	1.33	1.96	1.22	AA1
58	1.6	1.34	1.57	1.33	AA2
59	2.07	1.11	2	1.22	AA3
60	1.96	1.25	1.9	1.36	R_AC1
61	1.91	1.27	1.89	1.26	R_AC2
62	1.64	1.33	1.44	1.49	R_AC3
63	1.5	1.64	1.64	1.62	R_AC4
64	0.57	1.73	0.68	1.54	AI1
65	0.4	1.79	0.6	1.77	AI2
66	0.98	1.79	1.15	1.71	R_AI3
67	-0.18	1.83	-0.13	1.78	AI4
68	1.33	1.56	1.24	1.58	AT1
69	1.27	1.55	1.18	1.54	AT2
70	1.53	1.63	1.49	1.73	R_AT3
71	-0.13	1.82	-0.15	1.72	SP1
72	-0.62	1.8	-0.59	1.85	SP2
73	-1.11	1.73	-0.98	1.73	SP3
74	0.25	1.6	0.1	1.67	IIS1
75	0.23	1.56	0.34	1.63	IIS2
76	0.22	1.58	0.22	1.74	IIS3
77	0.2	1.63	0.18	1.6	IIS4
78	-0.32	2	-0.8	1.89	AEI1

ID	mean_German	sd_German	mean_English	sd_English	Item
79	-0.79	2.02	-0.89	1.87	AEI2
80	-0.07	2.14	-0.48	2.03	R_AEI3
81	-0.35	2.05	-0.49	1.92	AEI4
82	-0.98	2.03	-0.93	2.02	R_AEI5
83	1.26	1.48	0.91	1.79	UEP1
84	0.65	1.68	0.3	1.93	UEP2
85	0.91	1.71	0.95	1.69	UEP3
86	0.84	1.91	0.78	1.98	UEP4
87	0.96	1.77	0.88	1.83	UAI1
88	1.31	1.39	1.09	1.38	UAI2
89	1.18	1.41	1.35	1.46	UAI3
90	0.4	1.78	0.37	1.74	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	0	0	0	0	HLA1
2	0	0	0	0	HLA2
3	0	0	0	0	HLA3
4	-0.02	0.05	-0.14	0.06	HLA4
5	0	0.08	-0.17	0.16	HLB1
6	0.05	0.13	-0.21	0.3	HLB2
7	0.16	0.13	-0.06	0.42	HLB3
8	0.02	0.09	-0.16	0.21	HLB4
9	-0.06	0.11	-0.28	0.15	HLB5
10	0	0	0	0	NA1
11	0.14	0.1	-0.05	0.35	NA2
12	-0.09	0.1	-0.29	0.09	NA3
13	0.18	0.11	-0.02	0.4	NA4
14	0.17	0.12	-0.05	0.41	NA5
15	0	0	0	0	NB1
16	0.18	0.11	-0.03	0.39	NB2
17	0.17	0.12	-0.06	0.41	NB3
18	-0.01	0.07	-0.16	0.14	AAS1
19	0	0	0	0	AAS2
20	-0.06	0.09	-0.24	0.11	AAS3
21	0	0	0	0	AU1
22	0	0	0	0	AU2
23	0	0	0	0	AU3
24	0.02	0.05	0	0.16	PF1
25	0.02	0.06	-0.09	0.17	PF2
26	-0.09	0.11	-0.31	0.13	PF3
27	-0.05	0.12	-0.29	0.2	AL1
28	0	0	0	0	AL2
29	0	0	0	0	R_AL3
30	0.01	0.03	-0.03	0.06	AL4
31	0	0.04	0	0	AL5
32	0.89	0.17	0.56	1.22	AS1

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
33	0.58	0.15	0.28	0.88	AS2
34	0	0.08	-0.15	0.16	AS3
35	0.06	0.09	-0.13	0.24	APP1
36	-0.21	0.1	-0.4	-0.02	R_APP2
37	0	0	0	0	APP3
38	0	0	0	0	UAA1
39	0	0	0	0	UAA2
40	-0.11	0.13	-0.38	0.15	R_UAA3
41	0	0	0	0	R_AE1
42	0	0	0	0	AE2
43	-0.11	0.09	-0.29	0.06	AE3
44	-0.31	0.13	-0.56	-0.06	R_AE4
45	0	0	0	0	UE1
46	0	0.02	-0.04	0.05	UE2
47	0.08	0.08	-0.06	0.25	UE3
48	0.02	0.04	-0.05	0.11	UT1
49	0.04	0.08	-0.13	0.22	UT2
50	0.12	0.1	-0.07	0.32	UT3
51	0.12	0.1	-0.06	0.32	UAL1
52	-0.04	0.15	-0.34	0.27	UAL2
53	0.24	0.13	-0.01	0.5	UAL3
54	-0.21	0.11	-0.4	0	UAL4
55	-0.2	0.09	-0.38	-0.02	UAL5
56	-0.08	0.11	-0.3	0.13	UAL6
57	0	0	0	0	AA1
58	-0.02	0.05	-0.14	0.08	AA2
59	0	0	0	0	AA3
60	0	0	0	0	R_AC1
61	0	0	0	0	R_AC2
62	0	0	0	0	R_AC3
63	0	0	0	0	R_AC4
64	-0.07	0.09	-0.25	0.1	AI1
65	-0.17	0.09	-0.35	0.01	AI2
66	-0.07	0.1	-0.28	0.13	R_AI3
67	0	0.02	-0.05	0.05	AI4
68	0	0	0	0	AT1
69	0.07	0.07	-0.06	0.22	AT2
70	0	0	0	0	R_AT3
71	0	0	0	0	SP1
72	-0.02	0.08	-0.19	0.14	SP2
73	-0.03	0.06	-0.17	0.09	SP3
74	0	0	0	0	IIS1
75	0	0	0	0	IIS2
76	0.05	0.11	-0.17	0.27	IIS3
77	0	0	0	0	IIS4
78	0.47	0.13	0.22	0.72	AEI1
79	0	0	0	0	AEI2
80	0.05	0.08	-0.1	0.22	R_AEI3
81	0	0.04	-0.07	0.07	AEI4
82	0	0.04	-0.08	0.07	R_AEI5
83	0.09	0.1	-0.1	0.29	UEP1
84	0.18	0.1	-0.03	0.38	UEP2

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
85	0.06	0.09	-0.11	0.24	UEP3
86	0	0	0	0	UEP4
87	0	0	0	0	UAI1
88	0.21	0.14	-0.06	0.48	UAI2
89	-0.16	0.11	-0.39	0.05	UAI3
90	0.07	0.11	-0.14	0.28	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	20511	1	0.5		HLA1
2	21258	1	0.5		HLA2
3	20243	1	0.5		HLA3
4	2086	1.01	0.69		HLA4
5	15403	1	0.51		HLB1
6	12881	1	0.66		HLB2
7	1468	1	0.91		HLB3
8	18495	1	0.6		HLB4
9	14607	1	0.72		HLB5
10	18615	1	0.5		NA1
11	11511	1	0.93		NA2
12	14327	1	0.84		NA3
13	16242	1	0.96		NA4
14	17989	1	0.93		NA5
15	20309	1	0.5		NB1
16	17876	1	0.95		NB2
17	18420	1	0.92		NB3
18	9050	1	0.56		AAS1
19	20849	1	0.5		AAS2
20	18120	1	0.75		AAS3
21	21069	1	0.5		AU1
22	15647	1	0.5		AU2
23	19895	1	0.5		AU3
24	80	1.04	0.55		PF1
25	4851	1	0.66		PF2
26	17278	1	0.79		PF3
27	19092	1	0.65		AL1
28	21653	1	0.5		AL2
29	22540	1	0.51		R_AL3
30	316	1.12	0.57		AL4
31	17248	1	0.51		AL5
32	19107	1	1	*	AS1
33	19023	1	1	*	AS2
34	17205	1	0.52		AS3
35	17695	1	0.72		APP1
36	10784	1	0.99	*	R_APP2
37	21814	1	0.5		APP3
38	24718	1	0.5		UAA1

ID	n_eff	Rhat	P_posterior	zero_excl	Item
39	27083	1	0.5		UAA2
40	18391	1	0.8		R_UAA3
41	22399	1	0.5		R_AE1
42	21847	1.01	0.5		AE2
43	18589	1	0.9		AE3
44	8156	1	0.99	*	R_AE4
45	21099	1	0.5		UE1
46	10568	1	0.5		UE2
47	2093	1	0.86		UE3
48	3882	1	0.67		UT1
49	12723	1	0.67		UT2
50	16451	1	0.89		UT3
51	17194	1	0.9		UAL1
52	18801	1	0.59		UAL2
53	7867	1	0.98		UAL3
54	1648	1	0.98		UAL4
55	14848	1	0.98	*	UAL5
56	17846	1	0.76		UAL6
57	21937	1	0.5		AA1
58	8873	1	0.69		AA2
59	22515	1	0.5		AA3
60	22868	1	0.51		R_AC1
61	22133	1	0.5		R_AC2
62	21191	1	0.5		R_AC3
63	19877	1	0.51		R_AC4
64	16903	1	0.8		AI1
65	11712	1	0.98		AI2
66	19974	1	0.76		R_AI3
67	13486	1	0.51		AI4
68	21911	1	0.5		AT1
69	3557	1	0.81		AT2
70	20766	1	0.5		R_AT3
71	24475	1	0.5		SP1
72	15766	1	0.61		SP2
73	6616	1	0.67		SP3
74	19946	1	0.5		IIS1
75	22045	1	0.5		IIS2
76	13129	1	0.69		IIS3
77	21528	1	0.5		IIS4
78	18760	1	1	*	AEI1
79	18658	1	0.5		AEI2
80	9135	1	0.72		R_AEI3
81	15152	1.01	0.5		AEI4
82	14645	1	0.5		R_AEI5
83	15977	1	0.83		UEP1
84	14457	1	0.96		UEP2
85	17549	1	0.77		UEP3
86	20889	1	0.5		UEP4
87	14631	1	0.51		UAI1
88	18534	1	0.93		UAI2
89	16212	1	0.93		UAI3
90	17623	1	0.74		UAI4

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

# 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
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)

# 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	0.505925925925926
sd_German	1.66140192343459
mean_English	0.463240740740741
sd_English	1.66942692845849
mean_diff	0.0749652329869093
sd_diff	0.0599725281326737
minimum_diff	-0.30749569559454
maximum_diff	0.885519878373793
n_zero_excl	6
percent_zero_excl	6.67

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
# Initialize output of Constructs/dimensions with credible
# bias indication
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())

# Numbers of all columns in d_total, i.e. English and
```

```r
# German scores combined
n <- ncol(d_total)
# Offset for the column position of the first German ASAQ
# items
German_column_offset <- n/2

# Go step by step to 24 constructs/dimensions
for (p in 1:NUM_ITEMS_SHORT_ASAQ) {
    # The column with the first English ASAQ item of the
    # construct/dimension
    i = h[p, 1]
    # The column with the first German ASAQ item of the
    # construct/dimension
    j = i + German_column_offset
    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
    # German score means for each construct/dimension per
    # participant
    average_s_German <- data.frame(rowMeans(s_German))
    # English score means for each construct/dimension per
    # participant
    average_s_English <- data.frame(rowMeans(s_English))
    colnames(average_s_German) <- c("score")  # Rename German mean column
    colnames(average_s_English) <- c("score")  # Rename English mean column
    # Combine averaged scores of German and English
    # constructs/dimensions
    score <- data.frame(cbind(average_s_German, average_s_English))
    # Select averaged scores of each German
    # construct/dimension, make sure data format is
    # suitable for Bayesian paired t-test
    score_German <- score[, 1]
    # Select averaged scores of each English
    # construct/dimension, make sure data format is
    # suitable for Bayesian paired t-test
    score_English <- score[, 2]
    set.seed(1)  # Make sure that estimations of Bayesian analyses remain the same
    # Conduct Bayesian t-test
    fit <- bayes.t.test(score_German, score_English, paired = TRUE)
    # Call function 'getBAYES' to obtain relevant
    # information from Bayesian t-test output and add
    # result to output list
    con_list <- rbind(con_list, getBAYES(p, score_German, score_English,
        fit))
}

# Print results Add construct/dimension name code
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")
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	-1.18	1.91	-1.17	1.91	HLA
2	-0.32	1.67	-0.34	1.62	HLB
3	-0.51	1.51	-0.53	1.55	NA
4	-0.44	1.59	-0.53	1.63	NB
5	1.26	1.22	1.24	1.22	AAS
6	1.39	1.15	1.49	1.11	AU
7	1.28	1.12	1.25	1.01	PF
8	0.65	1.43	0.66	1.36	AL
9	0.8	1.32	0.32	1.39	AS
10	-0.58	1.58	-0.58	1.65	APP
11	1.26	1.2	1.27	1.21	UAA
12	1.1	1.33	1.19	1.21	AE
13	1.83	1	1.58	1.12	UE
14	0.51	1.1	0.4	1.14	UT
15	0.41	1.08	0.43	1.05	UAL
16	1.83	0.99	1.84	0.99	AA
17	1.75	1.13	1.72	1.19	AC
18	0.44	1.3	0.57	1.31	AI
19	1.38	1.45	1.3	1.44	AT
20	-0.62	1.52	-0.58	1.52	SP
21	0.22	1.25	0.21	1.37	IIS
22	-0.5	1.75	-0.72	1.72	AEI
23	0.91	1.36	0.73	1.49	UEP
24	0.96	1.01	0.92	1.06	UAI

```r
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	0.06	-0.11	0.11	HLA
2	0.03	0.07	-0.1	0.16	HLB
3	0.07	0.06	-0.06	0.2	NA
4	0.09	0.06	-0.03	0.21	NB
5	0.04	0.07	-0.1	0.19	AAS
6	-0.09	0.06	-0.21	0.03	AU
7	0.02	0.07	-0.11	0.14	PF
8	0	0.04	-0.09	0.08	AL
9	0.47	0.1	0.26	0.67	AS
10	0	0.07	-0.15	0.13	APP
11	-0.01	0.07	-0.15	0.13	UAA
12	-0.08	0.06	-0.2	0.03	AE
13	0.2	0.08	0.04	0.35	UE
14	0.09	0.07	-0.05	0.23	UT
15	-0.01	0.06	-0.13	0.1	UAL
16	0	0.06	-0.13	0.11	AA

ID	mean_diff	sd_diff	HDIlo	HDIup	Construct
17	0.03	0.05	-0.08	0.13	AC
18	-0.08	0.07	-0.22	0.07	AI
19	0.08	0.05	-0.02	0.18	AT
20	-0.06	0.08	-0.21	0.09	SP
21	-0.05	0.05	-0.14	0.06	IIS
22	0.17	0.06	0.05	0.3	AEI
23	0.12	0.07	-0.02	0.26	UEP
24	0.02	0.06	-0.09	0.14	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	16774	1	0.53		HLA
2	18940	1	0.66		HLB
3	16807	1	0.85		NA
4	18104	1	0.93		NB
5	17964	1	0.72		AAS
6	13869	1	0.94		AU
7	20193	1	0.62		PF
8	18789	1	0.54		AL
9	19816	1	1	*	AS
10	18332	1	0.53		APP
11	16592	1	0.57		UAA
12	17439	1	0.92		AE
13	14329	1	0.99	*	UE
14	17589	1	0.89		UT
15	19021	1	0.59		UAL
16	18618	1	0.52		AA
17	18363	1	0.7		AC
18	14009	1	0.86		AI
19	18450	1	0.95		AT
20	19032	1	0.77		SP
21	18338	1	0.83		IIS
22	11111	1	1	*	AEI
23	15818	1	0.96		UEP
24	16146	1	0.65		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")
# 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
```

```
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)
pander(GrandMean, caption = "Grand mean of 24 constructs/dimensions")
```

Table 17: Grand mean of 24 constructs/dimensions

Variable	Grand_mean
mean_German	0.575896990740741
sd_German	1.33195901229452
mean_English	0.5290625
sd_English	1.34375067260242
mean_diff	0.0759812987157153
sd_diff	0.0654433606212377
minimum_diff	-0.0913199706457842
maximum_diff	0.468603660303823
n_zero_excl	3
percent_zero_excl	12.5

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

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

```
# Initialize output of Representative items with credible
# bias indication
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())

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

# Go step by step to 24 representative items of the ASA
# questionnaire
for (i in 1:NUM_ITEMS_SHORT_ASAQ) {
    score_German <- as.numeric(ss[, i])  # Select German scores
    score_English <- as.numeric(ss[, i + English_column_offset])  # Select English scores
    set.seed(1)  # Make sure that estimations of Bayesian analyses remain the same
    fit <- bayes.t.test(score_German, score_English, paired = TRUE)
    rep_list <- rbind(rep_list, getBAYES(i, score_German, score_English,
        fit))
}

# Print results Add item name code
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")
```

```
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	-1.32	2.05	-1.25	2.05	HLA2
2	-0.26	1.96	-0.15	1.85	HLB5
3	-0.5	1.91	-0.64	1.94	NA4
4	0.02	1.92	-0.16	1.94	NB3
5	1.32	1.34	1.24	1.44	AAS1
6	1.35	1.33	1.51	1.24	AU1
7	1.3	1.4	1.15	1.25	PF1
8	0.61	1.83	0.62	1.76	AL2
9	0.57	1.58	-0.31	1.69	AS1
10	-0.22	1.78	-0.38	1.82	APP1
11	1.31	1.44	1.37	1.39	UAA1
12	0.81	1.77	0.72	1.85	R_AE1
13	1.76	1.2	1.77	1.26	UE2
14	0.71	1.4	0.52	1.61	UT3
15	-0.03	1.85	-0.17	1.8	UAL1
16	1.6	1.34	1.57	1.33	AA2
17	1.96	1.25	1.9	1.36	R_AC1
18	0.98	1.79	1.15	1.71	R_AI3
19	1.33	1.56	1.24	1.58	AT1
20	-0.62	1.8	-0.59	1.85	SP2
21	0.23	1.56	0.34	1.63	IIS2
22	-0.07	2.14	-0.48	2.03	R_AEI3
23	0.91	1.71	0.95	1.69	UEP3
24	0.4	1.78	0.37	1.74	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	0	0	0	0	HLA2
2	-0.06	0.11	-0.28	0.15	HLB5
3	0.18	0.11	-0.02	0.4	NA4
4	0.17	0.12	-0.06	0.41	NB3
5	-0.01	0.07	-0.16	0.14	AAS1
6	0	0	0	0	AU1
7	0.02	0.05	0	0.16	PF1
8	0	0	0	0	AL2
9	0.89	0.17	0.56	1.22	AS1
10	0.06	0.09	-0.13	0.24	APP1
11	0	0	0	0	UAA1
12	0	0	0	0	R_AE1

ID	mean_diff	sd_diff	HDIlo	HDIup	Item
13	0	0.02	-0.04	0.05	UE2
14	0.12	0.1	-0.07	0.32	UT3
15	0.12	0.1	-0.06	0.32	UAL1
16	-0.02	0.05	-0.14	0.08	AA2
17	0	0	0	0	R_AC1
18	-0.07	0.1	-0.28	0.13	R_AI3
19	0	0	0	0	AT1
20	-0.02	0.08	-0.19	0.14	SP2
21	0	0	0	0	IIS2
22	0.05	0.08	-0.1	0.22	R_AEI3
23	0.06	0.09	-0.11	0.24	UEP3
24	0.07	0.11	-0.14	0.28	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	21258	1	0.5		HLA2
2	14607	1	0.72		HLB5
3	16242	1	0.96		NA4
4	18420	1	0.92		NB3
5	9050	1	0.56		AAS1
6	21069	1	0.5		AU1
7	80	1.04	0.55		PF1
8	21653	1	0.5		AL2
9	19107	1	1	*	AS1
10	17695	1	0.72		APP1
11	24718	1	0.5		UAA1
12	22399	1	0.5		R_AE1
13	10568	1	0.5		UE2
14	16451	1	0.89		UT3
15	17194	1	0.9		UAL1
16	8873	1	0.69		AA2
17	22868	1	0.51		R_AC1
18	19974	1	0.76		R_AI3
19	21911	1	0.5		AT1
20	15766	1	0.61		SP2
21	22045	1	0.5		IIS2
22	9135	1	0.72		R_AEI3
23	17549	1	0.77		UEP3
24	17623	1	0.74		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")
# Calculate grand mean of mean_German, sd_German,
```

```r
# 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
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)
pander(GrandMean, caption = "Grand mean of 24 representative items")
```

Table 21: Grand mean of 24 representative items

Variable	Grand_mean
mean_German	0.589236111111111
sd_German	1.65312756994999
mean_English	0.512152777777778
sd_English	1.66031204063477
mean_diff	0.0802283678677806
sd_diff	0.0606496554301999
minimum_diff	-0.0733363204263967
maximum_diff	0.885519878373793
n_zero_excl	1
percent_zero_excl	4.17

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.

Li, Fengxiang, Siska Fitrianie, Merijn Bruijnes, Amal Abdulrahman, Fu Guo, and Willem-Paul Brinkman. 2023. "Mandarin Chinese Translation of the Artificial-Social-Agent Questionnaire Instrument for Evaluating Human-Agent Interaction." *Frontiers in Computer Science* 5.