Final assessment part 1: Correlation between the orginal English and the newly translated Dutch ASA Questionnaire

Johan Hensman

21 June, 2023

Contents

1 Introduction

This document presents statistical analyses of correlation between English and Dutch ASA questionnaires for item level, construct/dimension level, and short versions of the ASA questionnaire as reported in the paper Code based on Fengxiang Li with adaptations made to Dutch.

We use the following packages:

```r
library(foreign) # Open various data files
library(nlme)    # Run multilevel linear models
library(car)     # Package linear regression
library(haven) # Use read_sav fuction
library(dplyr) # Use select function
library(knitr) # Get markdown file
library(tinytex) # Use TeX environment
library(pander) # For pandering tables
panderOptions("table.alignment.default","left")
```

2 Data files

The input data for this analysis is based on 3 files. These aren't the fully raw data collected on Qualtrics as it pre-processed to anonymize and delte columns that are not needed/allowed to publish.

2.1 Final_ASA_Dutch_Round_1_First_Half_anonym.sav"

The first survey was split in two to prevent fatigue. In the first group first 12 constructs were asked and the second group the last 12 constructs. The participants rated 44 and 46 questionnaire items each and 14 attention checks that all had to be correct. The participants had the option to not recommend their data, but this was of importance when creating the questionnaire. In this final assessment all data will be used. this file contains data of the first 12 constructs

2.2 Final_ASA_Dutch_Round_1_Second_Half_anonym.sav

this file contains data from the last 12 constructs

2.3 Final_ASA_Dutch_Round_2_anonym.sav

After the first round was performed 37 items needed to be retranslated as they had a low correlation value. We received 37 new translation as well as 27 alternative translations that needed to be evaluated. The results of that survey is saved in this file.

```r
dataset <- read_sav("Final_ASA_Dutch_Round_1_First_Half_anonym.sav")
dataset2 <- read_sav("Final_ASA_Dutch_Round_1_Second_Half_anonym.sav")
dataset3 <- read_sav("Final_ASA_Dutch_Round_2_anonym.sav")
#Importing data
```

```r
#"dataset" contains ASA question items from the first 12 constructs
#"dataset2" contains ASA question items from the last 12 constructs
```

The data has some columns that are not necessary for this analysis, so we only select the attention checks and the questionnaire items of the survey.

```r
dataset <- data.frame(select(dataset, AC_English_1:Q_ITEMS_438.0))
dataset2 <- data.frame(select(dataset2, AC_English_1:Q42_442))
dataset3 <- data.frame(select(dataset3, AC_Dutch_1:English_UAI3))
#Selecting only relevant question items, which include attention checks
#and ASA questionnaire items
```

Here, we change the column names of such that the ASA questionnaire items are renamed according to which language they are in, and which construct they belong to. In the Legend Q1.txt you can find the questions that they belonged to.

```r
#Here, all the column names are changed so that ASA questionnaire items are
#named according to their construct.
colnames(dataset)[59:102] <- c("D_HLA1", "D_HLA2", "D_HLA3", "D_HLA4", "D_HLB1",
                               "D_HLB2", "D_HLB3", "D_HLB4", "D_HLB5", "D_NA1",
                               "D_NA2", "D_NA3", "D_NA4", "D_NA5", "D_NB1",
                               "D_NB2", "D_NB3", "D_AAS1", "D_AAS2", "D_AAS3",
                               "D_AU1", "D_AU2", "D_AU3", "D_PF1", "D_PF2",
                               "D_PF3", "D_AL1", "D_AL2", "D_R_AL3", "D_AL4",
                               "D_AL5", "D_AS1", "D_AS2", "D_AS3", "D_APP1",
                               "D_R_APP2", "D_APP3", "D_UAA1", "D_UAA2",
                               "D_R_UAA3", "D_R_AE1", "D_AE2", "D_AE3",
                               "D_R_AE4")
#Changing columns of Dutch ASA question items in dataset

colnames(dataset2)[61:106] <- c("D_UE1", "D_UE2", "D_UE3", "D_UT1", "D_UT2",
                                "D_UT3", "D_UAL1", "D_UAL2", "D_UAL3", "D_UAL4",
                                "D_UAL5", "D_UAL6", "D_AA1", "D_AA2", "D_AA3",
                                "D_R_AC1", "D_R_AC2", "D_R_AC3", "D_R_AC4",
                                "D_AI1", "D_AI2", "D_R_AI3", "D_AI4", "D_AT1",
                                "D_AT2", "D_R_AT3", "D_SP1", "D_SP2", "D_SP3",
                                "D_IIS1", "D_IIS2", "D_IIS3", "D_IIS4",
                                "D_AEI1", "D_AEI2", "D_R_AEI3", "D_AEI4",
                                "D_R_AEI5", "D_UEP1", "D_UEP2", "D_UEP3",
                                "D_UEP4", "D_UAI1", "D_UAI2", "D_UAI3",
```

```r
                                "D_UAI4")
#Changing columns of Dutch ASA question items in dataset2

colnames(dataset)[8:51] <- c("HLA1", "HLA2", "HLA3", "HLA4", "HLB1", "HLB2",
                             "HLB3", "HLB4", "HLB5", "NA1", "NA2", "NA3", "NA4",
                             "NA5", "NB1", "NB2", "NB3", "AAS1", "AAS2", "AAS3",
                             "AU1", "AU2", "AU3", "PF1", "PF2", "PF3", "AL1",
                             "AL2", "R_AL3", "AL4", "AL5", "AS1", "AS2", "AS3",
                             "APP1", "R_APP2", "APP3", "UAA1", "UAA2", "R_UAA3",
                             "R_AE1", "AE2", "AE3", "R_AE4")
#Changing columns of English ASA question items in dataset

colnames(dataset2)[8:53] <- c("UE1", "UE2", "UE3", "UT1", "UT2", "UT3", "UAL1",
                              "UAL2", "UAL3", "UAL4", "UAL5", "UAL6", "AA1",
                              "AA2", "AA3", "R_AC1", "R_AC2", "R_AC3", "R_AC4",
                              "AI1", "AI2", "R_AI3", "AI4", "AT1", "AT2",
                              "R_AT3", "SP1", "SP2", "SP3", "IIS1", "IIS2",
                              "IIS3", "IIS4", "AEI1", "AEI2", "R_AEI3", "AEI4",
                              "R_AEI5", "UEP1", "UEP2", "UEP3", "UEP4", "UAI1",
                              "UAI2", "UAI3", "UAI4")
#Changing columns of English ASA question items in dataset2
```

As the data got imported from Qualtrics not all cells had numeric values. So, we remove all text from the cells and make the text numerical.

```r
dataset <- as.data.frame(lapply(dataset[1:102],
                              function(y) as.numeric(gsub('[a-zA-Z]', '', y))))
dataset2 <- as.data.frame(lapply(dataset2[1:106],
                              function(y) as.numeric(gsub('[a-zA-Z]', '', y))))
dataset3 <- as.data.frame(lapply(dataset3[1:115],
                              function(y) as.numeric(gsub('[a-zA-Z]', '', y))))
#Transform the data to numeric representation
```

In this section we can omit the attention checks and only keep the quuestionnaire items.

```r
d1 <- as.data.frame(select(dataset, HLA1:R_AE4, D_HLA1:D_R_AE4))
# Select scores of 44 English items and corresponding Dutch translations
d2 <- as.data.frame(select(dataset2, UE1:UAI4, D_UE1:D_UAI4))
# Select scores of 46 English items and corresponding Dutch translations
```

```r
d3 <- as.data.frame(select(dataset3, English_HLA1:English_UAI3, Dutch_HLA:Dutch_UAI3))
# Select scores of 37 English items and corresponding Dutch translations
```

We manually checked which translation had the highest ICC value and picked them to create the variable with the best values. The chosen translations and their corresponding ICC value can be found in the same folder of this codebase under Best_ICC_and_translations.xlsx. To confirm these values we calculate the ICC values again from the datafiles. So we combine the necessary columns from each dataset.

```r
d_total1 <- cbind(select(d1,HLA1), select(d3,English_HLA1),
                 select(d1,HLA3:HLB3), select(d3,English_HLB2),
                 select(d1,HLB5:NA1), select(d3,English_NA1:English_NB1),
                 select(d1, NB2),  select(d3, English_NB3), select(d1, AAS1),
                 select(d3, English_AAS1:English_AAS2), select(d1, AU1:AU2),
                 select(d3, English_AU1), select(d1, PF1:PF3),
                 select(d3, English_AL1), select(d1, AL2:AL5),
                 select(d3, English_AS1), select(d1, AS2),
                 select(d3, English_AS2), select(d1,APP1:R_UAA3),
                 select(d3,English_AE1:English_AE2), select(d1, AE3:R_AE4),
                 select(d2, UE1), select(d3,English_UE1), select(d2, UE3:UT1),
                 select(d3, English_UT1), select(d2, UT3:UAL1),
                 select(d3, English_UAL1), select(d2, UAL3:AA2),
                 select(d3, English_AA2), select(d2, R_AC1:AI4),
                 select(d3, English_AT1), select(d2, AT2:R_AT3),
                 select(d3, English_SP1), select(d2, SP2:SP3),
                 select(d3, English_IIS1: English_AEI1),
                 select(d2, AEI2:R_AEI5),
                 select(d3, English_UEP), select(d2, UEP2:UAI1),
                 select(d3, English_UAI1: English_UAI3))
#select all English items from their correspoding datasets.


d_total2 <- cbind(select(d1,D_HLA1), select(d3,Dutch_HLA),
                 select(d1,D_HLA3:D_HLB3), select(d3,Dutch_HLB2),
                 select(d1,D_HLB5:D_NA1), select(d3,Dutch_NA1),
                 select(d3,Dutch_NA2_Alt1), select(d3,Dutch_NA3),
                 select(d3,Dutch_NA4_Alt2:Dutch_NB1),select(d1, D_NB2),
                 select(d3,Dutch_NB3_Alt2) , select(d1, D_AAS1),
                 select(d3, Dutch_AAS1), select(d3, Dutch_AAS2_Alt4),
                 select(d1, D_AU1:D_AU2), select(d3, Dutch_AU1_Alt1),
```

```r
                  select(d1, D_PF1:D_PF3), select(d3, Dutch_AL1_Alt1),
                  select(d1, D_AL2: D_AL5), select(d3, Dutch_AS1_Alt1),
                  select(d1, D_AS2), select(d3, Dutch_AS2_Alt3),
                  select(d1,D_APP1:D_R_UAA3), select(d3, Dutch_AE1_Alt1),
                  select(d3, Dutch_AE2), select(d1, D_AE3:D_R_AE4),
                  select(d2, D_UE1), select(d3, Dutch_UE1_Alt2),
                  select(d2, D_UE3:D_UT1), select(d3, Dutch_UT1_Alt1),
                  select(d2, D_UT3:D_UAL1), select(d3, Dutch_UAL1_Alt1),
                  select(d2, D_UAL3:D_AA2), select(d3, Dutch_AA2),
                  select(d2, D_R_AC1:D_AI4), select(d3, Dutch_AT1_Alt1),
                  select(d2, D_AT2:D_R_AT3), select(d3, Dutch_SP),
                  select(d2, D_SP2:D_SP3), select(d3, Dutch_IIS1_Alt2),
                  select(d3, Dutch_IIS2_Alt2), select(d3, Dutch_IIS3_Alt2),
                  select(d3, Dutch_IIS4: Dutch_AEI1_Alt1),
                  select(d2, D_AEI2:D_R_AEI5), select(d3, Dutch_UEP1_Alt1),
                  select(d2, D_UEP2:D_UAI1),
                  select(d3, Dutch_UAI1: Dutch_UAI3))
#select all Dutch items from their correspoding datasets.

d_total_Item <- cbind(d_total1, d_total2)
# Combine evaluation scores of the best items across the 3 data files.
```

As the columns of the 2nd round didnt had the proper naming, we renamed all columns again. (most values contained xx_xx_Alt 1 as there were multiple translation, but we want only the proper tags in the end.)

```r
colnames(d_total_Item)[1:180] <- c("HLA1", "HLA2", "HLA3", "HLA4", "HLB1", "HLB2",
                              "HLB3", "HLB4", "HLB5", "NA1", "NA2", "NA3", "NA4",
                              "NA5", "NB1", "NB2", "NB3", "AAS1", "AAS2", "AAS3",
                              "AU1", "AU2", "AU3", "PF1", "PF2", "PF3", "AL1",
                              "AL2", "R_AL3", "AL4", "AL5", "AS1", "AS2", "AS3",
                              "APP1", "R_APP2", "APP3", "UAA1", "UAA2", "R_UAA3",
                              "R_AE1", "AE2", "AE3", "R_AE4","UE1", "UE2", "UE3",
                              "UT1", "UT2", "UT3", "UAL1", "UAL2", "UAL3",
                              "UAL4", "UAL5", "UAL6", "AA1", "AA2", "AA3",
                              "R_AC1", "R_AC2", "R_AC3", "R_AC4", "AI1", "AI2",
                              "R_AI3", "AI4", "AT1", "AT2", "R_AT3", "SP1",
                              "SP2", "SP3", "IIS1", "IIS2", "IIS3", "IIS4",
                              "AEI1", "AEI2", "R_AEI3", "AEI4", "R_AEI5", "UEP1",
                              "UEP2", "UEP3", "UEP4", "UAI1", "UAI2", "UAI3",
```

```
                                "UAI4", "D_HLA1", "D_HLA2", "D_HLA3", "D_HLA4",
                                "D_HLB1", "D_HLB2", "D_HLB3", "D_HLB4", "D_HLB5",
                                "D_NA1", "D_NA2", "D_NA3", "D_NA4", "D_NA5",
                                "D_NB1", "D_NB2", "D_NB3", "D_AAS1", "D_AAS2",
                                "D_AAS3", "D_AU1", "D_AU2", "D_AU3", "D_PF1",
                                "D_PF2","D_PF3", "D_AL1", "D_AL2", "D_R_AL3",
                                "D_AL4", "D_AL5", "D_AS1", "D_AS2", "D_AS3",
                                "D_APP1", "D_R_APP2", "D_APP3", "D_UAA1",
                                "D_UAA2", "D_R_UAA3", "D_R_AE1", "D_AE2", "D_AE3",
                                "D_R_AE4", "D_UE1", "D_UE2", "D_UE3", "D_UT1",
                                "D_UT2", "D_UT3", "D_UAL1", "D_UAL2", "D_UAL3",
                                "D_UAL4", "D_UAL5", "D_UAL6", "D_AA1", "D_AA2",
                                "D_AA3", "D_R_AC1", "D_R_AC2", "D_R_AC3",
                                "D_R_AC4", "D_AI1", "D_AI2", "D_R_AI3", "D_AI4",
                                "D_AT1", "D_AT2", "D_R_AT3", "D_SP1", "D_SP2",
                                "D_SP3", "D_IIS1", "D_IIS2", "D_IIS3", "D_IIS4",
                                "D_AEI1", "D_AEI2", "D_R_AEI3", "D_AEI4",
                                "D_R_AEI5", "D_UEP1", "D_UEP2", "D_UEP3",
                                "D_UEP4", "D_UAI1", "D_UAI2", "D_UAI3", "D_UAI4")
```

we already prepare also the data we need for the construct level. As construct level icc is only valid if all questions from 1 construct are answered by the same person, we had to take almost all values from the First dataset(which was split in two). There was one construct where all questions were asked in round 2 and resulted in a higher ICC value. So the construct level data contains 1 construct (with the tag IIS) from Round 2 and the rest originate from round 1. The necessary columns were renamed to show the correct tags.

```
d_total1 <- cbind(select(d1,HLA1:R_AE4), select(d2,UE1:SP3),
                  select(d3, English_IIS1: English_IIS4), select(d2, AEI1:UAI4))
#select all English items from their correspoding datasets.

d_total2 <- cbind(select(d1,D_HLA1:D_R_AE4), select(d2,D_UE1:D_SP3),
                  select(d3, Dutch_IIS1_Alt2), select(d3, Dutch_IIS2_Alt2),
                  select(d3, Dutch_IIS3_Alt2), select(d3, Dutch_IIS4),
                  select(d2, D_AEI1:D_UAI4))
#select all Dutch items from their correspoding datasets.

d_total_Construct <- cbind(d_total1, d_total2)
#combine the items into 1.
```

```r
colnames(d_total_Construct)[74:77] <- c("IIS1", "IIS2", "IIS3", "IIS4")
colnames(d_total_Construct)[164:167] <-c("D_IIS1", "D_IIS2", "D_IIS3", "D_IIS4")
# rename the necessary columns.
```

All scores that have an R_ in front have to be reversed. This was not necessary for the item-level ICC value, but it is for the construct level as question like 'I don't like [the agent]' and 'I like [the agent]' will have opposite values so questions like 'i don't like [the agent]' are reversed.

```r
for (i in grep("R_",colnames(d_total_Construct))){
# Find column number of reversing-scoring items and translations
  d_total_Construct[[i]][] <- d_total_Construct[[i]][]*(-1)
  # Reverse scores of reverse-scoring items and translations
}
```

3 Analyses results reported in Section Results

3.1 function/model for ICC calculation

We fit a multilevel model with a fixed intercept (~1) and participants as the random intercept. $\rho_I = \frac{\tau^2}{\tau^2+\sigma^2}$ is the formula needed to calculate the ICC where τ^2 is the variance between participants, and σ^2 is the variance within the score of individual.

```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)
}
```

Next, we defined a function to run a multilevel mode. It takes as input the score from both languages and returns the ICC 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_Dutch<- data.frame(id, s_1, language= 1)
  # Transform Dutch 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_Dutch, Score_English)
  # Combine Dutch 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))
}
```

3.2 ICC values for 90 items

Now the function and the data is prepared to calculate the ICC values for the 90 questionnaire items. The grand mean, standard deviation and the range is also presented Note: the first half contains all English data and the second half contains all Dutch data.

```
l_ICC <- data.frame(ItemID = double(), Item = character(), icc = double())
# Initialize output of ICC values of 44 items
n <- ncol(d_total_Item)

# Numbers of columns in d1
Dutch_column_offset <- ncol(d_total_Item)/2
# Offset, the first column with scores of the Dutch version of ASAQ item
d0 <- data.frame(d_total_Item)


for (i in 1:90)
  # Go step by step to 90 items of the ASA questionnaire, whereby i is
  #the ASA questionnaire item number
{
  score_Dutch <- na.omit(data.frame(score=d0[,i + Dutch_column_offset ]))
  # Select scores of Dutch version of ASAQ item i
```

```r
  score_English <- na.omit(data.frame(score=d0[,i ]))
  # Select scores of English version of ASAQ items i
  iccScore <- getLME(score_Dutch, score_English)
  l_ICC <- rbind(l_ICC, data.frame (i, icc = iccScore))
  # Calculated ICC and add it to the list of ICC values,
  # with ID number of the ASA questionnaire item
}
l_ICC$Item = colnames(select(d0,HLA1:UAI4))
# give all items in the list the correct name
pander(l_ICC, caption = "ICC values for 90 items")
```

Table 1: ICC values for 90 items

i	icc	Item
1	0.6725	HLA1
2	0.4252	HLA2
3	0.6586	HLA3
4	0.7368	HLA4
5	0.6111	HLB1
6	0.3975	HLB2
7	0.7799	HLB3
8	0.5491	HLB4
9	0.6236	HLB5
10	0.6341	NA1
11	0.3608	NA2
12	0.6298	NA3
13	0.6054	NA4
14	0.4726	NA5
15	0.603	NB1
16	0.5097	NB2
17	0.4326	NB3
18	0.6291	AAS1
19	0.5327	AAS2
20	0.7904	AAS3
21	0.7509	AU1
22	0.7518	AU2
23	0.6966	AU3
24	0.8442	PF1
25	0.6643	PF2

i	icc	Item
26	0.6491	PF3
27	0.6732	AL1
28	0.8252	AL2
29	0.7139	R_AL3
30	0.7753	AL4
31	0.9249	AL5
32	0.5784	AS1
33	0.6752	AS2
34	0.595	AS3
35	0.7241	APP1
36	0.8079	R_APP2
37	0.633	APP3
38	0.6341	UAA1
39	0.7357	UAA2
40	0.3762	R_UAA3
41	0.6942	R_AE1
42	0.7005	AE2
43	0.7098	AE3
44	0.7142	R_AE4
45	0.6029	UE1
46	0.8083	UE2
47	0.696	UE3
48	0.6442	UT1
49	0.6662	UT2
50	0.7971	UT3
51	0.7106	UAL1
52	0.559	UAL2
53	0.6086	UAL3
54	0.7246	UAL4
55	0.5993	UAL5
56	0.6044	UAL6
57	0.7631	AA1
58	0.4828	AA2
59	0.5846	AA3
60	0.7388	R_AC1
61	0.9036	R_AC2
62	0.7605	R_AC3
63	0.8253	R_AC4

i	icc	Item
64	0.8023	AI1
65	0.8135	AI2
66	0.7973	R_AI3
67	0.7078	AI4
68	0.6247	AT1
69	0.7193	AT2
70	0.6371	R_AT3
71	0.632	SP1
72	0.7326	SP2
73	0.7511	SP3
74	0.808	IIS1
75	0.7088	IIS2
76	0.6341	IIS3
77	0.5529	IIS4
78	0.5169	AEI1
79	0.7272	AEI2
80	0.897	R_AEI3
81	0.7595	AEI4
82	0.8307	R_AEI5
83	0.5204	UEP1
84	0.6197	UEP2
85	0.7303	UEP3
86	0.6515	UEP4
87	0.7071	UAI1
88	0.6482	UAI2
89	0.7664	UAI3
90	0.4189	UAI4

```
#display all ICC values on item-level

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 ite
description <- cbind(Variable, Value) # Descriptive statistics of ICC values of 90 items
```

```r
# Print results
pander(description, caption = "Descriptive statistics of ICC values of 90 items")
```

Table 2: Descriptive statistics of ICC values of 90 items

Variable	Value
Grand_mean	0.67
SD	0.1187
Minimum	0.3608
Maximum	0.9249

For the assessment of the correlation between the English and Chinese ASA Questionnaire, we followed Cicchetti's classification of ICC categories.

```r
poor <- data.frame(ItemID = double(), Item = character(), icc = double())
fair <- data.frame(ItemID = double(), Item = character(), icc = double())
good <- data.frame(ItemID = double(), Item = character(), icc = double())
excellent <- data.frame(ItemID = double(), Item = character(), icc = double())
#Create categorizations of ICC values, ranging from poor to excellent

for(i in 1:90){
  if(l_ICC$icc[i]>=0.75) {
    #If the ICC value is greater than 0.75, it is excellent
     excellent <- rbind(excellent, data.frame (i ,l_ICC$Item[i] ,
                                               icc = l_ICC$icc[i]))
  } else if(l_ICC$icc[i]>=0.60) {
    #If the ICC value is between 0.60 and 0.75, it is good
     good <- rbind(good, data.frame (i, l_ICC$Item[i], icc = l_ICC$icc[i]))
  } else if(l_ICC$icc[i]>=0.4) {
    #If the ICC value is between 0.4 and 0.6, it is fair
     fair <- rbind(fair, data.frame (i, l_ICC$Item[i], icc = l_ICC$icc[i]))
  } else {
    #If the ICC value is below 0.4, it is poor
     poor <- rbind(poor, data.frame (i, l_ICC$Item[i], icc = l_ICC$icc[i]))
  }
}



pander(poor, caption = "ICC values for poor items")
```

Table 3: ICC values for poor items

i	l_ICC.Item.i.	icc
6	HLB2	0.3975
11	NA2	0.3608
40	R_UAA3	0.3762

```r
pander(fair, caption = "ICC values for fair items")
```

Table 4: ICC values for fair items

i	l_ICC.Item.i.	icc
2	HLA2	0.4252
8	HLB4	0.5491
14	NA5	0.4726
16	NB2	0.5097
17	NB3	0.4326
19	AAS2	0.5327
32	AS1	0.5784
34	AS3	0.595
52	UAL2	0.559
55	UAL5	0.5993
58	AA2	0.4828
59	AA3	0.5846
77	IIS4	0.5529
78	AEI1	0.5169
83	UEP1	0.5204
90	UAI4	0.4189

```r
pander(good, caption = "ICC values for good items")
```

Table 5: ICC values for good items

i	l_ICC.Item.i.	icc
1	HLA1	0.6725
3	HLA3	0.6586
4	HLA4	0.7368

i	l_ICC.Item.i.	icc
5	HLB1	0.6111
9	HLB5	0.6236
10	NA1	0.6341
12	NA3	0.6298
13	NA4	0.6054
15	NB1	0.603
18	AAS1	0.6291
23	AU3	0.6966
25	PF2	0.6643
26	PF3	0.6491
27	AL1	0.6732
29	R_AL3	0.7139
33	AS2	0.6752
35	APP1	0.7241
37	APP3	0.633
38	UAA1	0.6341
39	UAA2	0.7357
41	R_AE1	0.6942
42	AE2	0.7005
43	AE3	0.7098
44	R_AE4	0.7142
45	UE1	0.6029
47	UE3	0.696
48	UT1	0.6442
49	UT2	0.6662
51	UAL1	0.7106
53	UAL3	0.6086
54	UAL4	0.7246
56	UAL6	0.6044
60	R_AC1	0.7388
67	AI4	0.7078
68	AT1	0.6247
69	AT2	0.7193
70	R_AT3	0.6371
71	SP1	0.632
72	SP2	0.7326
75	IIS2	0.7088
76	IIS3	0.6341

i	l_ICC.Item.i.	icc
79	AEI2	0.7272
84	UEP2	0.6197
85	UEP3	0.7303
86	UEP4	0.6515
87	UAI1	0.7071
88	UAI2	0.6482

```
pander(excellent, caption = "ICC values for excellent items")
```

Table 6: ICC values for excellent items

i	l_ICC.Item.i.	icc
7	HLB3	0.7799
20	AAS3	0.7904
21	AU1	0.7509
22	AU2	0.7518
24	PF1	0.8442
28	AL2	0.8252
30	AL4	0.7753
31	AL5	0.9249
36	R_APP2	0.8079
46	UE2	0.8083
50	UT3	0.7971
57	AA1	0.7631
61	R_AC2	0.9036
62	R_AC3	0.7605
63	R_AC4	0.8253
64	AI1	0.8023
65	AI2	0.8135
66	R_AI3	0.7973
73	SP3	0.7511
74	IIS1	0.808
80	R_AEI3	0.897
81	AEI4	0.7595
82	R_AEI5	0.8307
89	UAI3	0.7664

```
#Display all ASA question items in their corresponding classification
```

3.3 ICC values for 24 constructs and related dimensions

We first calculate the means of each construct/dimension and the we use in the same way as the item level ICC the LME function to calculate the ICC values for each construct.

```
Dutch_column_offset <- length(d_total_Construct)/2

i <- which(names(d_total_Construct)%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"))
#vector with the indexes of the first item of all constructs.

k <- c(ncol(select(d_total_Construct, HLA1:HLA4)),
       ncol(select(d_total_Construct, HLB1:HLB5)),
       ncol(select(d_total_Construct, NA1:NA5)),
       ncol(select(d_total_Construct, NB1:NB3)),
       ncol(select(d_total_Construct, AAS1:AAS3)),
       ncol(select(d_total_Construct, AU1:AU3)),
       ncol(select(d_total_Construct, PF1:PF3)),
       ncol(select(d_total_Construct, AL1:AL5)),
       ncol(select(d_total_Construct, AS1:AS3)),
       ncol(select(d_total_Construct, APP1:APP3)),
       ncol(select(d_total_Construct, UAA1:R_UAA3)),
       ncol(select(d_total_Construct, R_AE1:R_AE4)),
       ncol(select(d_total_Construct, UE1:UE3)),
       ncol(select(d_total_Construct, UT1:UT3)),
       ncol(select(d_total_Construct, UAL1:UAL6)),
       ncol(select(d_total_Construct, AA1:AA3)),
       ncol(select(d_total_Construct, R_AC1:R_AC4)),
       ncol(select(d_total_Construct, AI1:AI4)),
       ncol(select(d_total_Construct, AT1:R_AT3)),
       ncol(select(d_total_Construct, SP1:SP3)),
       ncol(select(d_total_Construct, IIS1:IIS4)),
       ncol(select(d_total_Construct, AEI1:R_AEI5)),
       ncol(select(d_total_Construct, UEP1:UEP4)),
       ncol(select(d_total_Construct, UAI1:UAI4)))
# 'k' is a vector with the  questionnaire items of each
```

```r
# construct/dimension

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

C_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 + Dutch_column_offset
  # The column number of the first ASAQ item in the
  # Dutch version of the construct/dimension
  k <- h[p,2]
  # The number of ASAQ items associate to the construct/dimension
  s_Dutch <- data.frame(d_total_Construct[,j:(j+k-1)])
  # Select the scores of all the ASAQ items in Dutch
  # associated with the construct/dimension
  s_English <- data.frame(d_total_Construct[,i:(i+k-1)])
  # Select the score of all the ASAQ items in English associated
  # with the construct/dimension
  average_s_Dutch <- data.frame(rowMeans(s_Dutch))
  # Calculate the mean score of ASAQ items in Dutch 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_Dutch) <- c("score") # Rename Dutch mean column
  colnames(average_s_English) <- c("score") # Rename English mean column
  C_ICC <- rbind(C_ICC, data.frame (p, icc = getLME(average_s_Dutch, average_s_English)))
  # Call function 'getLME' for ICC value calculation
}

C_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')
```

```r
# Add construct/dimension name code
pander(C_ICC, caption = "ICC values for 24 constructs/dimensions")
```

Table 7: ICC values for 24 constructs/dimensions

p	icc	Construct
1	0.8343	HLA
2	0.8974	HLB
3	0.5175	NA
4	0.5687	NB
5	0.625	AAS
6	0.8051	AU
7	0.8878	PF
8	0.9069	AL
9	0.8152	AS
10	0.8571	APP
11	0.8451	UAA
12	0.804	AE
13	0.8425	UE
14	0.7408	UT
15	0.8191	UAL
16	0.8252	AA
17	0.9322	AC
18	0.8879	AI
19	0.7136	AT
20	0.8637	SP
21	0.7918	IIS
22	0.9027	AEI
23	0.8041	UEP
24	0.7424	UAI

```r
# display the ICC values
```

The results are presented in the same way as the item-level ICC values.

```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(C_ICC$icc) # Number of ICC values
```

```r
round_ICC <- round(C_ICC$icc, digits=2) # Round ICC values
Number <- c(length(C_ICC[which(round_ICC>=0.75&round_ICC<=1),]$icc),
            length(C_ICC[which(round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(C_ICC[which(round_ICC>=0.40&round_ICC<=0.59),]$icc),
            length(C_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 8: Categories of ICC classifications and number of ICC values in classification category for 24 constructs/dimensions The grand mean, standard deviation and the range is also presented

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

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(C_ICC$icc),digits=4),round(sd(C_ICC$icc),digits=4),
           round(min(C_ICC$icc),digits=4),round(max(C_ICC$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 9: Descriptive statistics of ICC values of 24 constructs/dimensions

Variable	Value
Grand_mean	0.8012
SD	0.1056
Minimum	0.5175
Maximum	0.9322

3.4 ICC values between English and Chinese 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_Dutch <- select(d_total_Item,D_HLA2,D_HLB5,D_NA4,D_NB3,D_AAS1,D_AU1,D_PF1,
                  D_AL2,D_AS1,D_APP1,D_UAA1, D_R_AE1,D_UE2,D_UT3,D_UAL1,D_AA2,
                  D_R_AC1,D_R_AI3,D_AT1,D_SP2,D_IIS2,D_R_AEI3,D_UEP3,D_UAI4)
# Select Chinese versions of the 24 representative ASAQ items
s_English <- select(d_total_Item,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
d_total_Short <- cbind(s_Dutch,s_English)
# Combine Chinese and English scores
n <- ncol(d_total_Short) # Numbers of all columns in d_total_Short
Dutch_column_offset <- n /2
s_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_Dutch <- data.frame(score=d_total_Short[,i])
# Select Chinese scores of the ASAQ item
score_English <- data.frame(score=d_total_Short[,i+ Dutch_column_offset])
# Select English scores of the ASAQ item
s_ICC <- rbind(s_ICC, data.frame (i, icc = getLME(score_Dutch, score_English)))
# Call function 'getLME' for ICC value calculation
}
```

```r
s_ICC$Item <- colnames(s_English) # Add item name code
pander(s_ICC, caption = "ICC values for 24 representative items")
```

Table 10: ICC values for 24 representative items

i	icc	Item
1	0.4252	HLA2
2	0.6236	HLB5
3	0.6054	NA4
4	0.4326	NB3
5	0.6291	AAS1
6	0.7509	AU1
7	0.8442	PF1
8	0.8252	AL2
9	0.5784	AS1
10	0.7241	APP1
11	0.6341	UAA1
12	0.6942	R_AE1
13	0.8083	UE2
14	0.7971	UT3
15	0.7106	UAL1
16	0.4828	AA2
17	0.7388	R_AC1
18	0.7973	R_AI3
19	0.6247	AT1
20	0.7326	SP2
21	0.7088	IIS2
22	0.897	R_AEI3
23	0.7303	UEP3
24	0.4189	UAI4

```r
Variable <- c("Grand_mean","SD","Minimum","Maximum")
# Define the names of the statistics
Value <- c(round(mean(s_ICC$icc),digits=4),round(sd(s_ICC$icc),digits=4),
           round(min(s_ICC$icc),digits=4),round(max(s_ICC$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 11: Descriptive statistics of ICC values of 24 representative items

Variable	Value
Grand_mean	0.6756
SD	0.1345
Minimum	0.4189
Maximum	0.897

```
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(s_ICC$icc) # Number of ICC values
round_ICC <- round(s_ICC$icc, digits=2) # Round ICC values
Number <- c(length(s_ICC[which(round_ICC>=0.75&round_ICC<=1),]$icc),
            length(s_ICC[which(round_ICC>=0.60&round_ICC<=0.74),]$icc),
            length(s_ICC[which(round_ICC>=0.40&round_ICC<=0.59),]$icc),
            length(s_ICC[which(round_ICC>=0.00&round_ICC<=0.39),]$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 12: 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	7	29.17
Good	0.60-0.74	12	50
Fair	0.40-0.59	5	20.83
Poor	0-0.39	0	0