Comparison of ICCs using IRT and CTT parameters

Diego Figueiras[1] & John T. Kulas[1]

[1] Montclair State University

Author Note

Add complete departmental affiliations for each author here. Each new line herein must be indented, like this line.

Enter author note here.

The authors made the following contributions. Diego Figueiras: Conceptualization, Writing - Original Draft Preparation, Writing - Review & Editing; John T. Kulas: Writing - Review & Editing.

Correspondence concerning this article should be addressed to Diego Figueiras, Postal address. E-mail: figueirasd1@montclair.edu

Abstract

One or two sentences providing a **basic introduction** to the field, comprehensible to a scientist in any discipline.

Two to three sentences of **more detailed background**, comprehensible to scientists in related disciplines.

One sentence clearly stating the **general problem** being addressed by this particular study.

One sentence summarizing the main result (with the words "**here we show**" or their equivalent).

Two or three sentences explaining what the **main result** reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more **general context**.

Two or three sentences to provide a **broader perspective**, readily comprehensible to a scientist in any discipline.

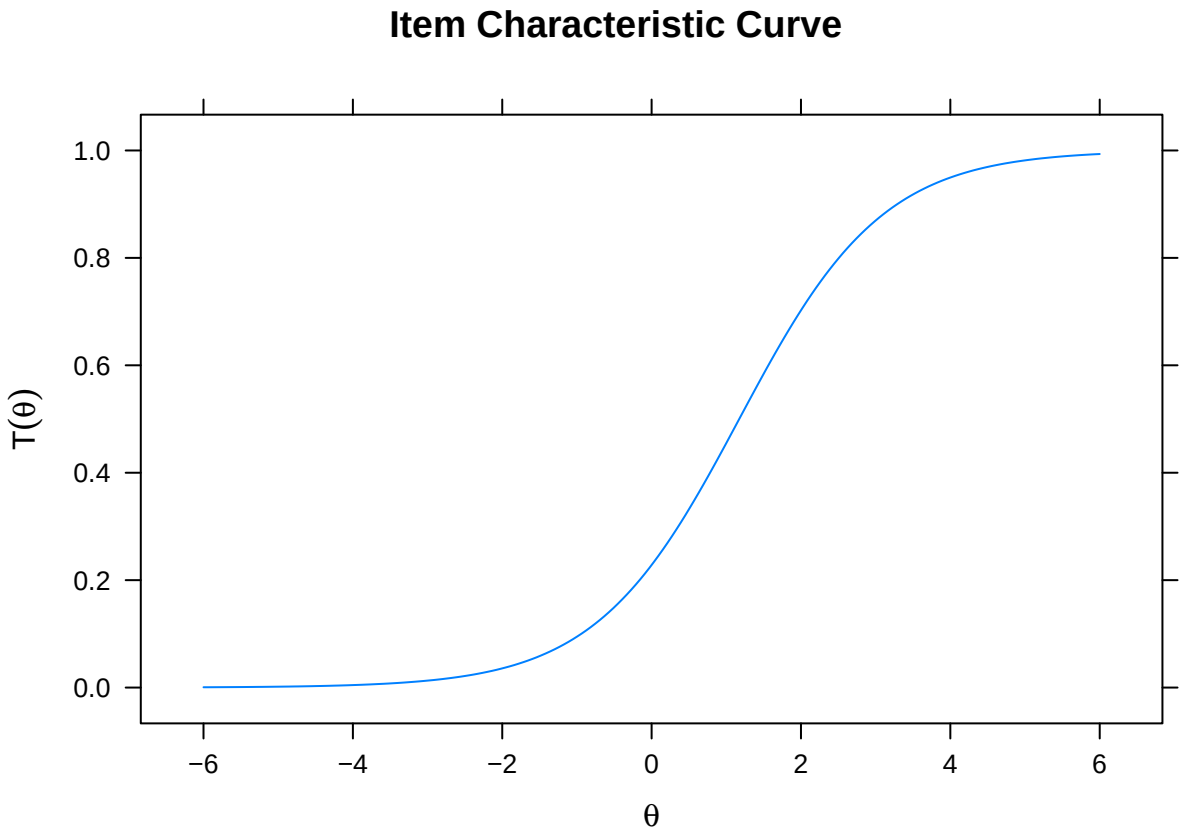
Keywords: keywords

Word count: X

Comparison of ICCs using IRT and CTT parameters

Introduction

Item characteristic curves are very often used by psychometricians to showcase and analyze the attributes of the item on a test or assessment. The x-axis shows a wide range of trait levels (ranging from high to low on the trait), while the y-axis displays probabilities of getting the item correct that range from 0 to 1. Each item has a curve. By looking at it, we can know the likelihood with which respondents of any trait level would answer any item correctly.If the curve is leaning towards the lower end of the trait level, this indicates that it is easy to answer the item correctly. On the contrary, if the curve is leaning towards the higher end of the trait level, this indicates that the item is difficult. If the curve is steep, this indicates high discrimination among respondents; if it is flat, it indicates no discrimination.



Psychometricians who examine ICCs usually do it while using Item Response Theory and Rasch models. From a Classical Test Theory (CTT) frame of thinking, the difficulty of an item is determined by looking at the p-values of the items, while discrimination is determined by checking the Cronbach alpha and the corrected item total correlations. Psychometricians who look at these CTT parameters don't typically use them to plot ICCs.There is no reason for this to happen, since ICCs based on CTT parameters could provide information as valuable as those based on IRT or Rasch without the need of being familiar with these models. Fan states in summary that IRT and CTT "... framework produce very similar item and person statistics" (p.379).

Methods

We used the formulas presented by Kulas, Smith, and Xu (2017).

Study 2 simulates a bunch of test data and then we generate ICCs based on the IRT model and then we compare that to our CTT estimates. ## Participants

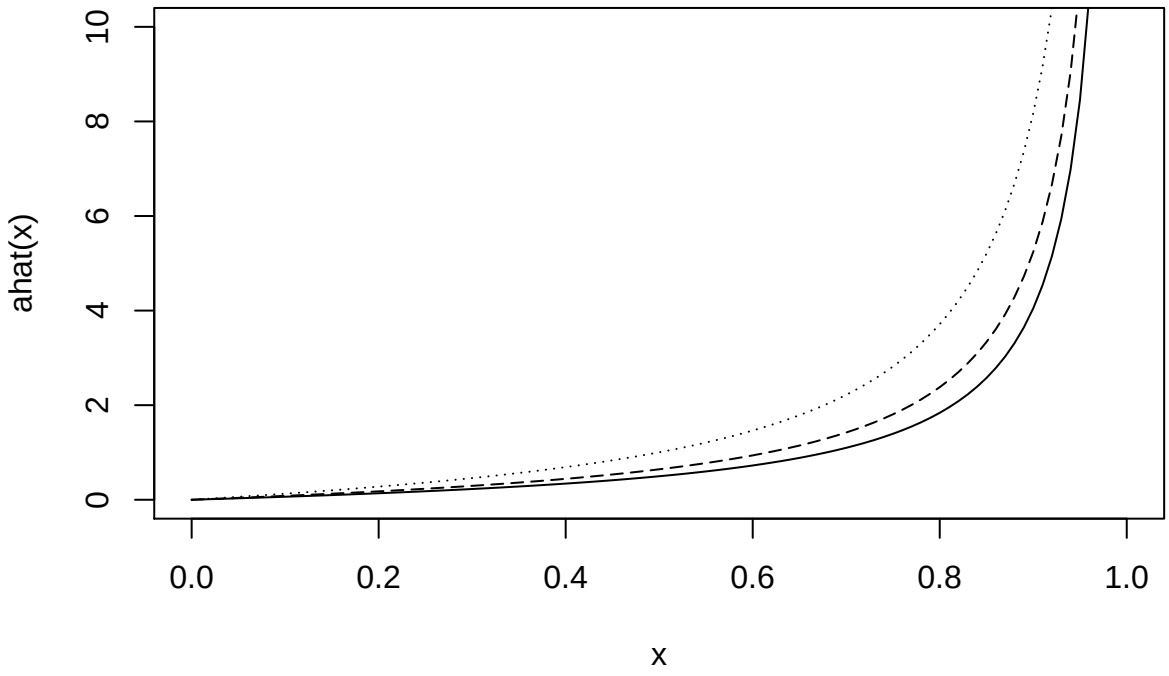
Material

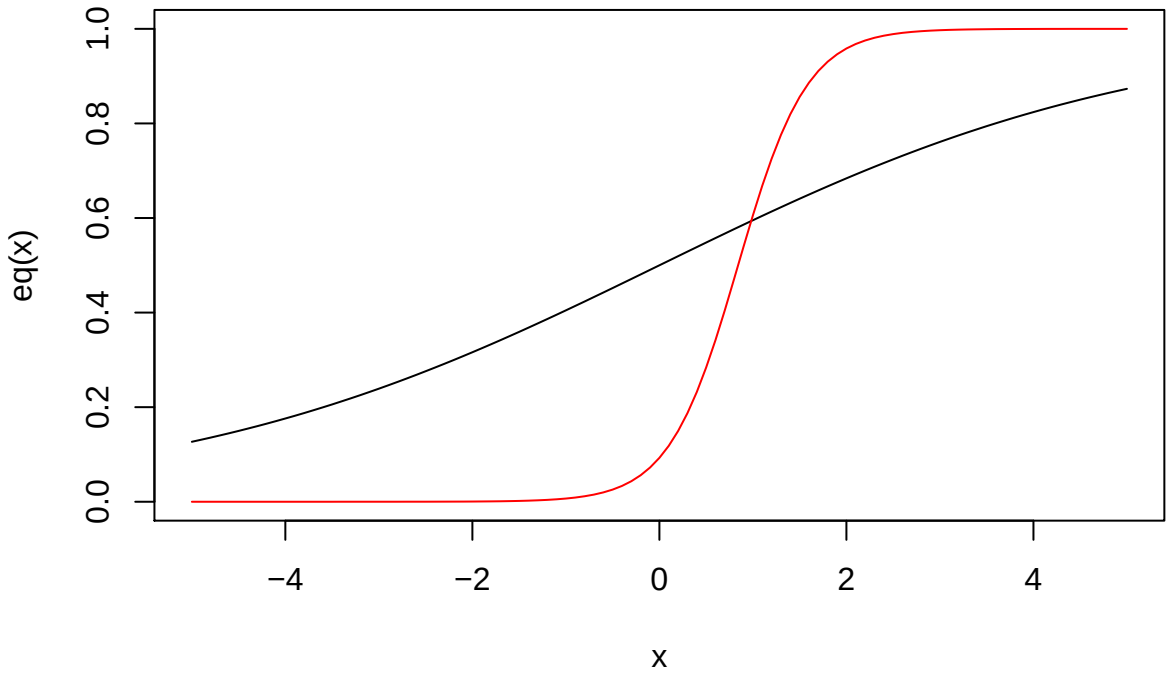
Procedure

Data analysis

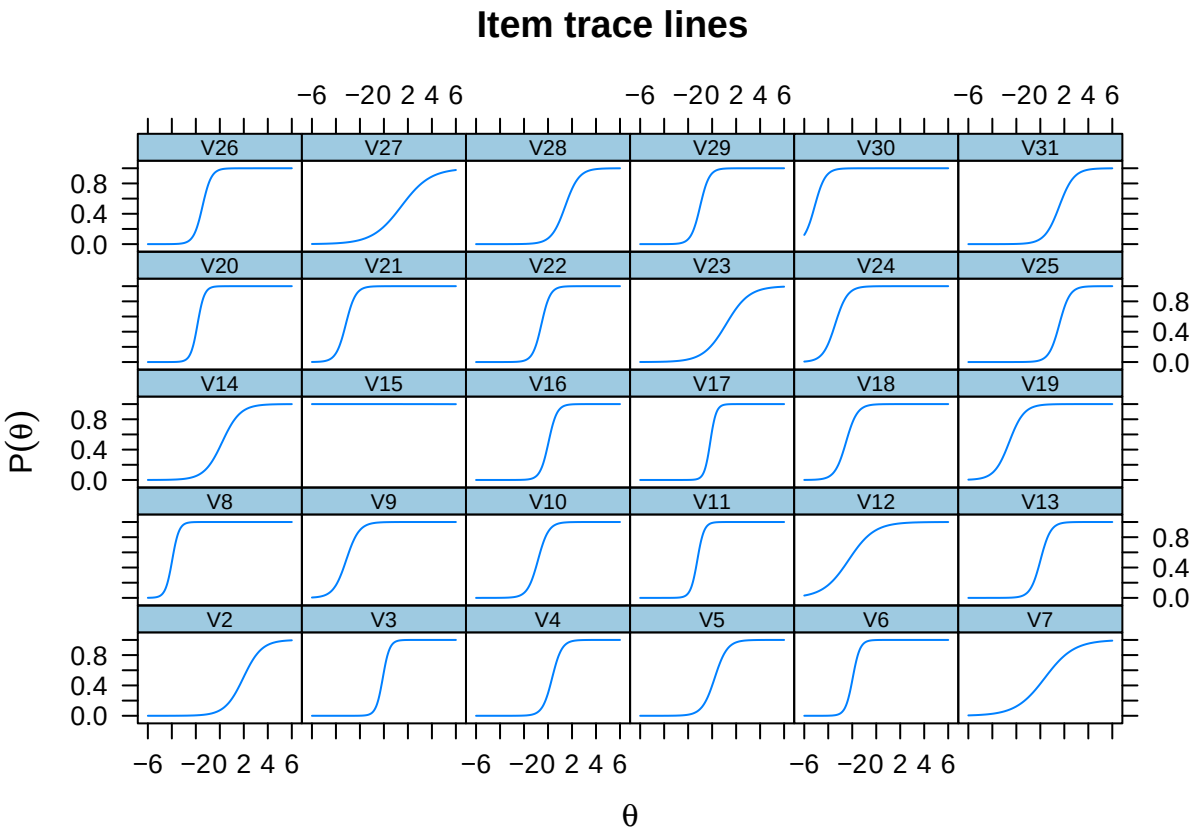
We used R (Version 4.0.3; R Core Team, 2020) and the R-packages *dplyr* (Version 1.0.7; Wickham et al., 2021), *DT* (Version 0.19; Xie, Cheng, & Tan, 2021), *forcats* (Version 0.5.1; Wickham, 2021a), *formattable* (Version 0.2.1; Ren & Russell, 2021), *ggplot2* (Version 3.3.5; Wickham, 2016), *jpeg* (Version 0.1.9; Urbanek, 2021), *knitr* (Version 1.33; Xie, 2015), *markdown* (Version 1.1; Allaire, Horner, Xie, Marti, & Porte, 2019; Xie, Allaire, & Grolemund, 2018; Xie, Dervieux, & Riederer, 2020), *officer* (Version 0.3.19; Gohel, 2021), *papaja* (Version 0.1.0.9997; Aust & Barth, 2020), *pdftools* (Version 3.0.1; Ooms, 2021),

psych (Version 2.1.6; Revelle, 2021), *purrr* (Version 0.3.4; Henry & Wickham, 2020), *readr* (Version 2.0.1; Wickham & Hester, 2021), *readxl* (Version 1.3.1; Wickham & Bryan, 2019), *reticulate* (Version 1.20; Ushey, Allaire, & Tang, 2021), *rmarkdown* (Version 2.10; Xie et al., 2018, 2020), *shiny* (Version 1.6.0; Chang et al., 2021), *stringr* (Version 1.4.0; Wickham, 2019), *tibble* (Version 3.1.4; Müller & Wickham, 2021), *tidyr* (Version 1.1.3; Wickham, 2021b), *tidyverse* (Version 1.3.1; Wickham, Averick, et al., 2019), and *tinytex* (Version 0.33; Xie, 2019) for all our analyses.



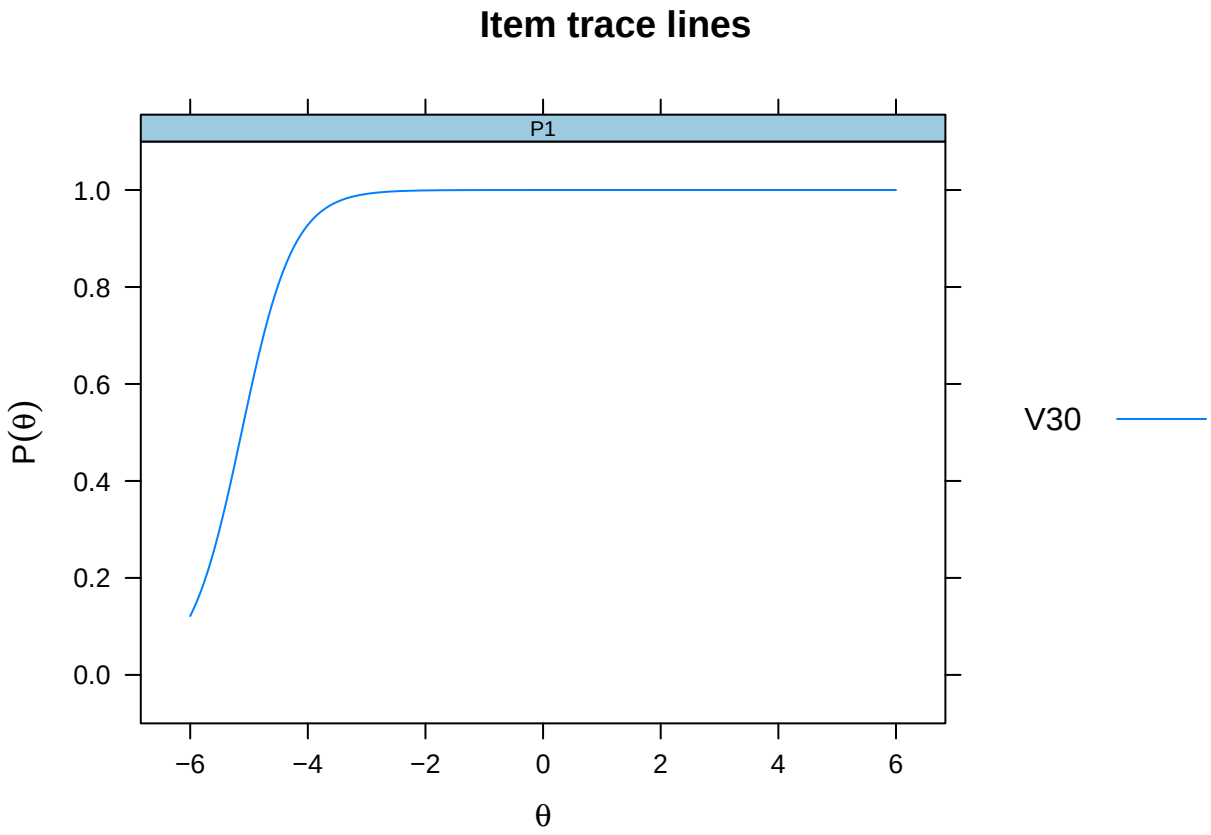



```
## Iteration: 1, Log-Lik: -98116.710, Max-Change: 4.09843Iteration: 2, Log-Lik: -93555.3
##
## Calculating information matrix...
```



78

79 ## Iteration: 1, Log-Lik: -98116.710, Max-Change: 4.09843Iteration: 2, Log-Lik: -93555.3




```
## Warning in mean.default(data$v10): argument is not numeric or logical: returning
## NA

## Warning in alpha(data): Some items were negatively correlated with the total scale an
## should be reversed.
## To do this, run the function again with the 'check.keys=TRUE' option

## Some items ( V15 ) were negatively correlated with the total scale and
## probably should be reversed.
## To do this, run the function again with the 'check.keys=TRUE' option

##
## Reliability analysis
```

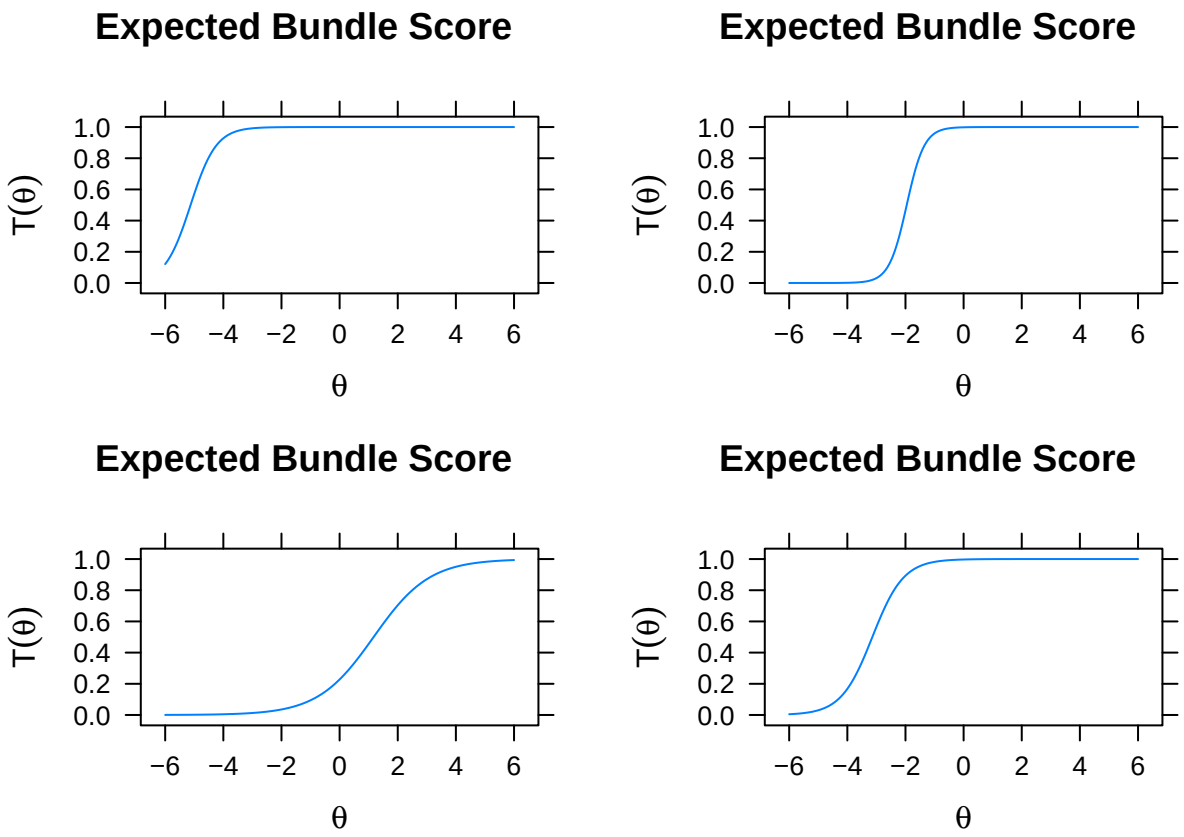
```
## Call: alpha(x = data)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.88      0.86    0.87      0.17 6.2 0.0015 0.65 0.17     0.15
##
##  lower alpha upper     95% confidence boundaries
## 0.88 0.88 0.89
##
##  Reliability if an item is dropped:
##     raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## V2       0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
## V3       0.87      0.85    0.86      0.16 5.7   0.0016 0.017  0.15
## V4       0.87      0.85    0.86      0.17 5.8   0.0016 0.018  0.15
## V5       0.88      0.85    0.87      0.17 5.8   0.0015 0.018  0.15
## V6       0.88      0.86    0.87      0.17 6.0   0.0015 0.019  0.15
## V7       0.88      0.86    0.87      0.17 6.1   0.0014 0.020  0.16
## V8       0.88      0.87    0.88      0.18 6.4   0.0015 0.018  0.17
## V9       0.88      0.86    0.87      0.18 6.2   0.0015 0.020  0.16
## V10      0.88      0.85    0.86      0.17 5.8   0.0015 0.018  0.15
## V11      0.88      0.85    0.86      0.17 5.8   0.0015 0.018  0.15
## V12      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
## V13      0.87      0.85    0.86      0.17 5.8   0.0016 0.018  0.15
## V14      0.88      0.86    0.87      0.17 5.9   0.0015 0.019  0.15
## V15      0.88      0.87    0.88      0.18 6.5   0.0015 0.018  0.17
## V16      0.87      0.85    0.86      0.16 5.7   0.0016 0.017  0.15
## V17      0.87      0.85    0.86      0.16 5.6   0.0016 0.017  0.15
## V18      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
```

```
## V19      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
## V20      0.88      0.86    0.87      0.17 5.9   0.0015 0.019  0.15
## V21      0.88      0.86    0.87      0.18 6.3   0.0015 0.019  0.16
## V22      0.87      0.85    0.86      0.16 5.7   0.0016 0.018  0.15
## V23      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.15
## V24      0.88      0.86    0.87      0.18 6.3   0.0015 0.019  0.16
## V25      0.88      0.86    0.87      0.17 6.1   0.0015 0.019  0.15
## V26      0.88      0.85    0.87      0.17 5.8   0.0015 0.019  0.15
## V27      0.88      0.86    0.87      0.17 6.1   0.0014 0.020  0.16
## V28      0.88      0.86    0.87      0.17 6.0   0.0015 0.019  0.15
## V29      0.88      0.85    0.86      0.17 5.8   0.0015 0.018  0.15
## V30      0.88      0.87    0.88      0.18 6.5   0.0015 0.018  0.17
## V31      0.88      0.86    0.87      0.17 6.1   0.0015 0.019  0.15
##
##  Item statistics
##         n  raw.r std.r   r.cor  r.drop mean    sd
## V2  10000  0.366  0.36  0.3121  0.3092 0.12 0.325
## V3  10000  0.748  0.70  0.7108  0.6999 0.53 0.499
## V4  10000  0.665  0.62  0.6141  0.6074 0.39 0.488
## V5  10000  0.629  0.58  0.5730  0.5656 0.44 0.496
## V6  10000  0.395  0.47  0.4512  0.3613 0.96 0.203
## V7  10000  0.422  0.39  0.3453  0.3386 0.43 0.495
## V8  10000  0.041  0.13  0.0656  0.0383 1.00 0.014
## V9  10000  0.198  0.28  0.2306  0.1762 0.99 0.115
## V10 10000  0.627  0.61  0.6077  0.5717 0.74 0.436
## V11 10000  0.590  0.62  0.6151  0.5437 0.86 0.343
## V12 10000  0.341  0.35  0.3060  0.2802 0.86 0.343
```

```
## V13 10000  0.681  0.63  0.6307  0.6235 0.49 0.500
## V14 10000  0.545  0.51  0.4800  0.4718 0.45 0.498
## V15 10000 -0.003  0.07 -0.0044 -0.0049 1.00 0.010
## V16 10000  0.716  0.67  0.6708  0.6633 0.48 0.500
## V17 10000  0.763  0.71  0.7286  0.7179 0.56 0.497
## V18 10000  0.291  0.36  0.3174  0.2612 0.97 0.164
## V19 10000  0.303  0.36  0.3100  0.2668 0.96 0.201
## V20 10000  0.419  0.49  0.4751  0.3838 0.95 0.216
## V21 10000  0.160  0.25  0.1894  0.1450 0.99 0.077
## V22 10000  0.691  0.67  0.6672  0.6386 0.68 0.468
## V23 10000  0.434  0.41  0.3677  0.3608 0.27 0.443
## V24 10000  0.154  0.24  0.1777  0.1384 0.99 0.083
## V25 10000  0.431  0.42  0.3818  0.3810 0.10 0.306
## V26 10000  0.529  0.56  0.5525  0.4825 0.89 0.318
## V27 10000  0.375  0.35  0.3054  0.2985 0.26 0.438
## V28 10000  0.457  0.44  0.4022  0.3987 0.16 0.362
## V29 10000  0.622  0.62  0.6200  0.5709 0.80 0.403
## V30 10000  0.024  0.10  0.0314  0.0223 1.00 0.010
## V31 10000  0.430  0.41  0.3762  0.3733 0.14 0.345
##
## Non missing response frequency for each item
##        0    1 miss
## V2  0.88 0.12    0
## V3  0.47 0.53    0
## V4  0.61 0.39    0
## V5  0.56 0.44    0
## V6  0.04 0.96    0
```

```
## V7  0.57 0.43    0
## V8  0.00 1.00    0
## V9  0.01 0.99    0
## V10 0.26 0.74    0
## V11 0.14 0.86    0
## V12 0.14 0.86    0
## V13 0.51 0.49    0
## V14 0.55 0.45    0
## V15 0.00 1.00    0
## V16 0.52 0.48    0
## V17 0.44 0.56    0
## V18 0.03 0.97    0
## V19 0.04 0.96    0
## V20 0.05 0.95    0
## V21 0.01 0.99    0
## V22 0.32 0.68    0
## V23 0.73 0.27    0
## V24 0.01 0.99    0
## V25 0.90 0.10    0
## V26 0.11 0.89    0
## V27 0.74 0.26    0
## V28 0.84 0.16    0
## V29 0.20 0.80    0
## V30 0.00 1.00    0
## V31 0.86 0.14    0
```

```
## Warning: package 'gridExtra' was built under R version 4.0.5
```



198

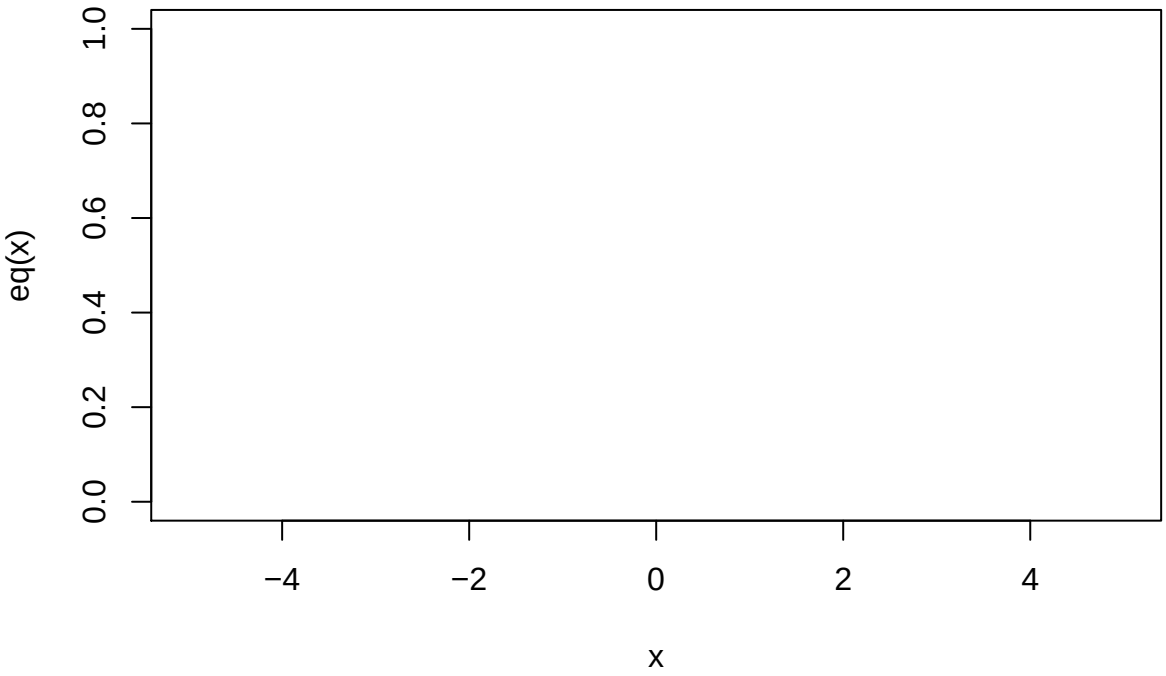
```
199 ## Warning in mean.default(data$i28): argument is not numeric or logical: returning
200 ## NA

201 ## Warning in alpha(data): Some items were negatively correlated with the total scale an
202 ## should be reversed.
203 ## To do this, run the function again with the 'check.keys=TRUE' option

204 ## Some items ( V15 ) were negatively correlated with the total scale and
205 ## probably should be reversed.
206 ## To do this, run the function again with the 'check.keys=TRUE' option

207 ##
208 ## Reliability analysis
```

```
## Call: alpha(x = data)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.88      0.86    0.87      0.17 6.2 0.0015 0.65 0.17     0.15
##
##  lower alpha upper     95% confidence boundaries
## 0.88 0.88 0.89
##
##  Reliability if an item is dropped:
##     raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## V2       0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
## V3       0.87      0.85    0.86      0.16 5.7   0.0016 0.017  0.15
## V4       0.87      0.85    0.86      0.17 5.8   0.0016 0.018  0.15
## V5       0.88      0.85    0.87      0.17 5.8   0.0015 0.018  0.15
## V6       0.88      0.86    0.87      0.17 6.0   0.0015 0.019  0.15
## V7       0.88      0.86    0.87      0.17 6.1   0.0014 0.020  0.16
## V8       0.88      0.87    0.88      0.18 6.4   0.0015 0.018  0.17
## V9       0.88      0.86    0.87      0.18 6.2   0.0015 0.020  0.16
## V10      0.88      0.85    0.86      0.17 5.8   0.0015 0.018  0.15
## V11      0.88      0.85    0.86      0.17 5.8   0.0015 0.018  0.15
## V12      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
## V13      0.87      0.85    0.86      0.17 5.8   0.0016 0.018  0.15
## V14      0.88      0.86    0.87      0.17 5.9   0.0015 0.019  0.15
## V15      0.88      0.87    0.88      0.18 6.5   0.0015 0.018  0.17
## V16      0.87      0.85    0.86      0.16 5.7   0.0016 0.017  0.15
## V17      0.87      0.85    0.86      0.16 5.6   0.0016 0.017  0.15
## V18      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
```

```
## V19      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.16
## V20      0.88      0.86    0.87      0.17 5.9   0.0015 0.019  0.15
## V21      0.88      0.86    0.87      0.18 6.3   0.0015 0.019  0.16
## V22      0.87      0.85    0.86      0.16 5.7   0.0016 0.018  0.15
## V23      0.88      0.86    0.87      0.17 6.1   0.0015 0.020  0.15
## V24      0.88      0.86    0.87      0.18 6.3   0.0015 0.019  0.16
## V25      0.88      0.86    0.87      0.17 6.1   0.0015 0.019  0.15
## V26      0.88      0.85    0.87      0.17 5.8   0.0015 0.019  0.15
## V27      0.88      0.86    0.87      0.17 6.1   0.0014 0.020  0.16
## V28      0.88      0.86    0.87      0.17 6.0   0.0015 0.019  0.15
## V29      0.88      0.85    0.86      0.17 5.8   0.0015 0.018  0.15
## V30      0.88      0.87    0.88      0.18 6.5   0.0015 0.018  0.17
## V31      0.88      0.86    0.87      0.17 6.1   0.0015 0.019  0.15
##
##  Item statistics
##         n  raw.r std.r   r.cor  r.drop mean    sd
## V2  10000  0.366  0.36  0.3121  0.3092 0.12 0.325
## V3  10000  0.748  0.70  0.7108  0.6999 0.53 0.499
## V4  10000  0.665  0.62  0.6141  0.6074 0.39 0.488
## V5  10000  0.629  0.58  0.5730  0.5656 0.44 0.496
## V6  10000  0.395  0.47  0.4512  0.3613 0.96 0.203
## V7  10000  0.422  0.39  0.3453  0.3386 0.43 0.495
## V8  10000  0.041  0.13  0.0656  0.0383 1.00 0.014
## V9  10000  0.198  0.28  0.2306  0.1762 0.99 0.115
## V10 10000  0.627  0.61  0.6077  0.5717 0.74 0.436
## V11 10000  0.590  0.62  0.6151  0.5437 0.86 0.343
## V12 10000  0.341  0.35  0.3060  0.2802 0.86 0.343
```

```
## V13 10000  0.681  0.63  0.6307  0.6235 0.49 0.500
## V14 10000  0.545  0.51  0.4800  0.4718 0.45 0.498
## V15 10000 -0.003  0.07 -0.0044 -0.0049 1.00 0.010
## V16 10000  0.716  0.67  0.6708  0.6633 0.48 0.500
## V17 10000  0.763  0.71  0.7286  0.7179 0.56 0.497
## V18 10000  0.291  0.36  0.3174  0.2612 0.97 0.164
## V19 10000  0.303  0.36  0.3100  0.2668 0.96 0.201
## V20 10000  0.419  0.49  0.4751  0.3838 0.95 0.216
## V21 10000  0.160  0.25  0.1894  0.1450 0.99 0.077
## V22 10000  0.691  0.67  0.6672  0.6386 0.68 0.468
## V23 10000  0.434  0.41  0.3677  0.3608 0.27 0.443
## V24 10000  0.154  0.24  0.1777  0.1384 0.99 0.083
## V25 10000  0.431  0.42  0.3818  0.3810 0.10 0.306
## V26 10000  0.529  0.56  0.5525  0.4825 0.89 0.318
## V27 10000  0.375  0.35  0.3054  0.2985 0.26 0.438
## V28 10000  0.457  0.44  0.4022  0.3987 0.16 0.362
## V29 10000  0.622  0.62  0.6200  0.5709 0.80 0.403
## V30 10000  0.024  0.10  0.0314  0.0223 1.00 0.010
## V31 10000  0.430  0.41  0.3762  0.3733 0.14 0.345
##
## Non missing response frequency for each item
##        0    1 miss
## V2  0.88 0.12    0
## V3  0.47 0.53    0
## V4  0.61 0.39    0
## V5  0.56 0.44    0
## V6  0.04 0.96    0
```

```
## V7  0.57 0.43    0
## V8  0.00 1.00    0
## V9  0.01 0.99    0
## V10 0.26 0.74    0
## V11 0.14 0.86    0
## V12 0.14 0.86    0
## V13 0.51 0.49    0
## V14 0.55 0.45    0
## V15 0.00 1.00    0
## V16 0.52 0.48    0
## V17 0.44 0.56    0
## V18 0.03 0.97    0
## V19 0.04 0.96    0
## V20 0.05 0.95    0
## V21 0.01 0.99    0
## V22 0.32 0.68    0
## V23 0.73 0.27    0
## V24 0.01 0.99    0
## V25 0.90 0.10    0
## V26 0.11 0.89    0
## V27 0.74 0.26    0
## V28 0.84 0.16    0
## V29 0.20 0.80    0
## V30 0.00 1.00    0
## V31 0.86 0.14    0
```



315

316 **Results**

317 **Discussion**

References

Allaire, J., Horner, J., Xie, Y., Marti, V., & Porte, N. (2019). *Markdown: Render markdown with the c library 'sundown'*. Retrieved from https://CRAN.R-project.org/package=markdown

Aust, F., & Barth, M. (2020). *papaja: Create APA manuscripts with R Markdown*. Retrieved from https://github.com/crsh/papaja

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., ... Borges, B. (2021). *Shiny: Web application framework for r*. Retrieved from https://CRAN.R-project.org/package=shiny

Gohel, D. (2021). *Officer: Manipulation of microsoft word and powerpoint documents*. Retrieved from https://CRAN.R-project.org/package=officer

Henry, L., & Wickham, H. (2020). *Purrr: Functional programming tools*. Retrieved from https://CRAN.R-project.org/package=purrr

Kulas, J. T., Smith, J. A., & Xu, H. (2017). Approximate functional relationship between irt and ctt item discrimination indices: A simulation, validation, and practical extension of lord's (1980) formula. *Journal of Applied Measurement*, *18*(4), 393–407.

Müller, K., & Wickham, H. (2021). *Tibble: Simple data frames*. Retrieved from https://CRAN.R-project.org/package=tibble

Ooms, J. (2021). *Pdftools: Text extraction, rendering and converting of pdf documents*. Retrieved from https://CRAN.R-project.org/package=pdftools

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Ren, K., & Russell, K. (2021). *Formattable: Create 'formattable' data structures*. Retrieved from https://CRAN.R-project.org/package=formattable

Revelle, W. (2021). *Psych: Procedures for psychological, psychometric, and personality research*. Evanston, Illinois: Northwestern University. Retrieved from https://CRAN.R-project.org/package=psych

Urbanek, S. (2021). *Jpeg: Read and write jpeg images*. Retrieved from https://CRAN.R-project.org/package=jpeg

Ushey, K., Allaire, J., & Tang, Y. (2021). *Reticulate: Interface to 'python'*. Retrieved from https://CRAN.R-project.org/package=reticulate

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Wickham, H. (2019). *Stringr: Simple, consistent wrappers for common string operations*. Retrieved from https://CRAN.R-project.org/package=stringr

Wickham, H. (2021a). *Forcats: Tools for working with categorical variables (factors)*. Retrieved from https://CRAN.R-project.org/package=forcats

Wickham, H. (2021b). *Tidyr: Tidy messy data*. Retrieved from https://CRAN.R-project.org/package=tidyr

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., & Bryan, J. (2019). *Readxl: Read excel files*. Retrieved from https://CRAN.R-project.org/package=readxl

Wickham, H., François, R., Henry, L., & Müller, K. (2021). *Dplyr: A grammar of data manipulation*. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H., & Hester, J. (2021). *Readr: Read rectangular text data*. Retrieved from https://CRAN.R-project.org/package=readr

Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://yihui.org/knitr/

Xie, Y. (2019). TinyTeX: A lightweight, cross-platform, and easy-to-maintain latex distribution based on tex live. *TUGboat*, (1), 30–32. Retrieved from http://tug.org/TUGboat/Contents/contents40-1.html

Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide*. Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://bookdown.org/yihui/rmarkdown

Xie, Y., Cheng, J., & Tan, X. (2021). *DT: A wrapper of the javascript library 'datatables'*. Retrieved from https://CRAN.R-project.org/package=DT

Xie, Y., Dervieux, C., & Riederer, E. (2020). *R markdown cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://bookdown.org/yihui/rmarkdown-cookbook