Pretending not to know

Matan Mazor1, Chaz Firestone2, & Ian Phillips2

1 Birkbeck, University of London

2 Johns Hopkins 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.

Correspondence concerning this article should be addressed to Matan Mazor, Malet St., London WC1E 7HX. E-mail: [mtnmzor@gmail.com](mailto:mtnmzor@gmail.com)

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:* pretense, theory of mind, metacognition, self-model, self-simulation

*Word count:* X

Pretending not to know

# Exp. 1: Battleship

## Methods

We report how we determined our sample size, all data exclusions (if any), all manipulations, and all measures in the study.

A detailed pre-registration can be accessed at [osf.io/v9zsb](https://osf.io/v9zsb)

### Participants.

The research complied with all relevant ethical regulations and was approved by the Research Ethics Committee of Johns Hopkins University. 500 Participants were recruited via Prolific (prolific.co) and gave their informed consent prior to their participation. They were selected based on their acceptance rate (>95%) and for being native English speakers. The entire experiment took approximately 20 minutes to complete. Participants’ pay was equivalent to an hourly wage of 9.50 USD, in addition to a bonus payment (0.20 - 2 USD, mean = 0.90).

### Procedure.

Participants were first instructed that the experiment, based on the game Battleships, had three parts, and that they could accumulate ‘points’ that would later translate to a monetary bonus payment. They were then presented with a leaderboard of previous players, and given the rules of the game:

“In the game Battleships, your task is to sink all ships located in a grid with as few clicks as possible. What makes the game difficult is that you can’t see the ships; all you can see is a grid of squares, and you have to guess where the ships are. To sink a ship, you need to click on all of the squares it is located in. If you hit part of a ship, the square will turn red. If there is no ship in the square, it will turn blue.”

We further explained that in this version of the game, ships can touch corners, but their sides can’t touch. This explanation was accompanied by a visual presentation of legal and illegal ship configurations.

After completing a comprehension question and a practice round, participants completed one ‘pretend’ and one ‘non-pretend’ block, each comprising five full games and one half game (see below for details). The order of pretend and non-pretend blocks was counterbalanced between participants. The allocation of boards (spatial configurations of ships; see Fig. 1A) to conditions was randomized between participants such that exactly one board was played in both pretend and non-pretend conditions, and this common board was different for different participants. The order of boards within a block was fully randomized, with the exception that half-games were always played last.

#### Non-pretend (normal) games.

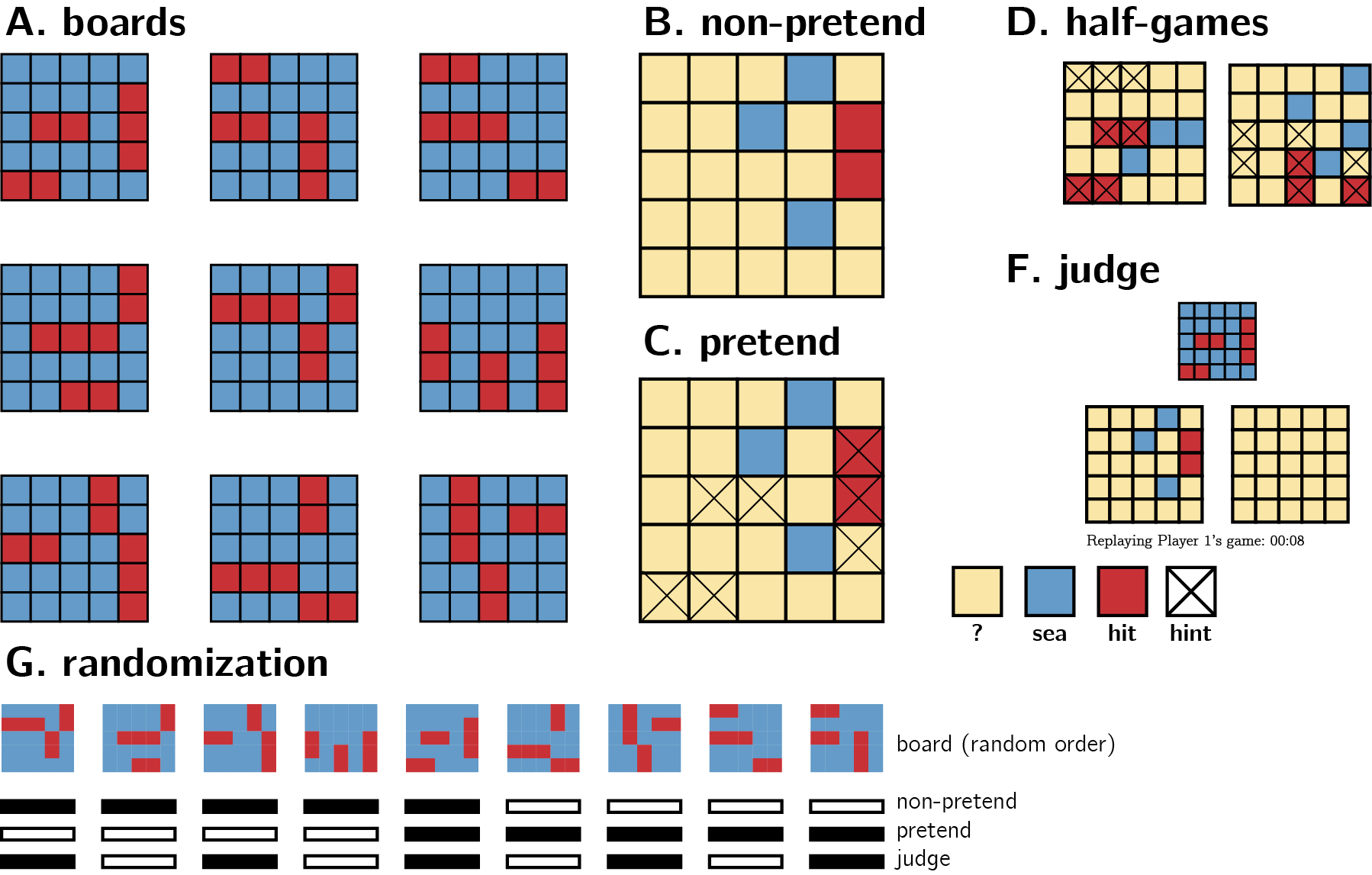
In non-pretend games (Fig. 1B), participants aimed to sink two 2-square patrol boats and one 3-square submarine with as few clicks as possible. An online counter of the number of clicks was displayed on the screen. After each game, feedback was given about the number of clicks and resulting number of points obtained.

#### Pretend games.

Participants in pretend games were given the same explanation of Battleships, and played a practice round. However, they were then given an additional twist:

“This time your goal is different. In this round, we’re going to tell you where the ships are, but **we want you to act like you don’t know this information**. We’ve marked the ships’ locations with a cross, so you’ll know where they are the whole time; but your job is to play the game as if these hints aren’t there. To see how good you are at this, we’re going to compare your games to the games of people who actually had no hints, and see how similar they are. We will measure where and when you clicked; if your clicks look similar to people who played like normal (trying to reveal all ships with as few clicks as possible, but without any hints), you’ll get bonus points. But if your games look different, you won’t get these bonus points. Your number of clicks in this part will not affect your bonus. Only your ability to play like you had no hints.”

We informed participants that both the location and timing of their cell clicks will be measured. After one practice round and one comprehension question, participants played five pretend games (Fig. 1C), followed by one pretend half-game. Each game was followed by a short message, reminding them that a game that looks similar to the games of participants who had no hints would be awarded 10 bonus points.



*Figure* *1.*  Experimental Design. See Methods for details.

#### Half games.

In order to directly compare participants’ pretend and non-pretend games for identical belief states (genuine or pretended ignorance about where the ships are hidden), participants completed one pretend and one non-pretend game given a partly finished board with the content of 7 cells already revealed (see Fig. 1D). We designed our half games to produce a strong expectation to find a ship in specific cells, but not in others. The assignment of half-completed boards to pretend and non-pretend conditions was randomized between participants.

#### Judge trials.

In the final part of the experiment, participants observed the games of previous players and tried to determine who had hints and who didn’t. On each trial, two empty grids were presented side by side, with a smaller grid on top, displaying the hidden positions of ships on the grid (Fig. 1F). The two grids corresponded to the true games of two previous players who played a version of the top grid either as pretenders or as non-pretenders. Only games shorter than one minute were chosen for presentation in this part. For non-pretend games, only games from the group of participants that pretended in the second block (and played normally in the first block) were chosen for presentation in this part. Judge participants observed a real time replay of the two grids, showing not only where participants clicked, but also when. After making a decision, participants were informed whether they would receive the 10 bonus points, or alternatively, whether the pretender would receive them in the event the pretender managed to trick them.

A more detailed description of the study procedure is provided in the study [pre-registration document](https://osf.io/v9zsb). Readers are also invited to try a [demo of the experiment](https://jatos.mindprobe.eu/publix/NervzpM0Y0z).

### Data analysis.

## Results

We designed our analyses to explore subjects’ capacity for self-simulation under a counterfactual knowledge state, and the limits of this capacity. We focused on where subjects clicked and when, and asked whether this differed between pretend and non-pretend games. All analyses were pre-registered unless otherwise specified. In our pre-registration document, we committed to separately analyzing participants according to whether they pretended before or after completing a non-pretend block. Whenever there is a mismatch between the two groups, we focus on the first block of each participant (only pretend games from participants who pretended first, and only standard games from participants who played normally first). We do so to ensure that any successful pretending is not due to memory of one’s own behavior in a previous block, and that non-pretend games are not biased by experience with the pretend block.

*Hypothesis 1 (game duration)*: Pretend games were longer than non-pretend games by 9 seconds on average. This was true in the pretend-first group (, ), and in the pretend-second groups (, ).

*Hypothesis 2 (first click latency)*: The first click in pretend games took longer to execute by 559 ms. This was true in the pretend-first group (, ), but not in the pretend-second groups (, ). As per our pre-registration document, we focused our analysis on the first blocks only, using a between-subject t-test. This test revealed a significant difference in the first click latency (, 95% CI , , ).

*Hypothesis 3 (number of clicks)*:

To sink all ships, players had to click on at least 7 and at most 25 squares. A simulated player that clicks randomly had a median total click number of 23, and a near-optimal greedy player that consistently selected the square with the highest objective probability of containing a ship had a median total click number of 14. Among our players, the median number of clicks was 16 in both pretend and non-pretend games.

No significant difference in the total number of clicks was observed in the pretend-first group (, ), but a significant difference was observed in the pretend-second group (, ). Note that the two effects point in opposite directions, reflecting a tendency to make fewer clicks in the second block of the game. Crucially, when focusing on the first block only, We observed no significant difference in the number of clicks between the two conditions (, ).

In 62 pretend games from 20 players, games were completed after 7 clicks only, without ever missing a ship. This never happened in non-pretend games. We assumed that these participants did not follow the task instructions, and excluded them from all analyses.

*Hypotheses 4-6 (click latency by click outcome)*: Trials will be classified based on whether they resulted in hitting a ship (*hit*) or not (*miss*). We will then contrast median click latency between hits and misses separately for pretend and non-pretend games. We will test for a difference in trial latency as a function of outcome on pretend games, on non-pretend games, and for an interaction effect between click outcome (hit or miss) and condition (pretend or non-pretend) on click latency. The same analysis will then be repeated by classifying trials on the basis of whether the previous trial resulted in a hit or a miss (*Hypothesis 5*), and whether the next trial will result in a hit or a miss (*Hypothesis 6*).

# Exp. 2: Hangman

## Methods

We report how we determined our sample size, all data exclusions (if any), all manipulations, and all measures in the study.

### Participants.

### Material.

### Procedure.

### Data analysis.

We used R (Version 4.2.2; R Core Team, 2021) and the R-packages *}egg* [@}R-egg], *afex* (Version 1.2.0; Singmann, Bolker, Westfall, Aust, & Ben-Shachar, 2021), *BayesFactor* (Version 0.9.12.4.4; Morey & Rouder, 2018), *brms* (Version 2.18.0; Bürkner, 2017, 2018), *broom* (Version 1.0.1; Robinson, Hayes, & Couch, 2021), *caret* (Version 6.0.93; Kuhn, 2022), *cocor* (Version 1.1.4; Diedenhofen & Musch, 2015), *coda* (Version 0.19.4; Plummer, Best, Cowles, & Vines, 2006), *cowplot* (Version 1.1.1; Wilke, 2020), *dplyr* (Version 1.0.10; Wickham, François, Henry, & Müller, 2021), *fddm* (Version 0.5.2; Foster, 2022), *forcats* (Version 0.5.2; Wickham, 2021a), *ggplot2* (Version 3.4.0; Wickham, 2016), *ggrepel* (Version 0.9.2; Slowikowski, 2021), *gridExtra* (Version 2.3; Auguie, 2017), *groundhog* (Version 2.1.0; Simonsohn & Gruson, 2022), *jsonlite* (Version 1.8.3; Ooms, 2014), *kernlab* (Version 0.9.31; Karatzoglou, Smola, Hornik, & Zeileis, 2004), *knitr* (Version 1.41; Xie, 2015), *lattice* (Version 0.20.45; Sarkar, 2008), *lme4* (Version 1.1.31; Bates, Mächler, Bolker, & Walker, 2015), *lmerTest* (Version 3.1.3; Kuznetsova, Brockhoff, & Christensen, 2017), *lsr* (Version 0.5.2; Navarro, 2015), *ltm* (Version 1.2.0; Rizopoulos, 2006), *magrittr* (Version 2.0.3; Bache & Wickham, 2020), *MASS* (Version 7.3.58.1; Venables & Ripley, 2002), *Matrix* (Version 1.5.3; Bates & Maechler, 2021), *memoise* (Version 2.0.1; Wickham, Hester, Chang, Müller, & Cook, 2021), *MESS* (Version 0.5.9; Ekstrøm, 2020), *modeest* (Poncet, 2019), *msm* (Version 1.7; Jackson, 2011), *osfr* (Wolen et al., 2020), *papaja* (Version 0.1.1; Aust & Barth, 2020), *png* (Version 0.1.8; Urbanek, 2013), *polycor* (Version 0.8.1; Fox, 2022), *ppcor* (Kim, 2015), *pracma* (Borchers, 2022), *psych* (Version 2.2.9; Revelle, 2022), *purrr* (Version 0.3.5; Henry & Wickham, 2020), *pwr* (Version 1.3.0; Champely, 2020), *RColorBrewer* (Version 1.1.3; Neuwirth, 2014), *Rcpp* (Eddelbuettel & Balamuta, 2018; Version 1.0.9; Eddelbuettel & François, 2011), *readr* (Version 2.1.3; Wickham & Hester, 2021), *reticulate* (Version 1.26; Ushey, Allaire, & Tang, 2021), *rstatix* (Version 0.7.1; Kassambara, 2021), *stringr* (Version 1.4.1; Wickham, 2019), *svglite* (Version 2.1.0; Wickham et al., 2022), *thesisdown* (Ismay & Solomon, 2021), *tibble* (Version 3.1.8; Müller & Wickham, 2021), *tidyr* (Version 1.2.1; Wickham, 2021b), *tidyverse* (Version 1.3.2; Wickham et al., 2019), *tinylabels* (Version 0.2.3; Barth, 2022), *varhandle* (Mahmoudian, 2020), *xtable* (Version 1.8.4; Dahl, Scott, Roosen, Magnusson, & Swinton, 2019), and *zoo* (Version 1.8.11; Zeileis & Grothendieck, 2005) for all our analyses.

## Results

# Discussion

# References

Auguie, B. (2017). *gridExtra: Miscellaneous functions for "grid" graphics*. Retrieved from <https://CRAN.R-project.org/package=gridExtra>

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

Bache, S. M., & Wickham, H. (2020). *Magrittr: A forward-pipe operator for r*. Retrieved from <https://CRAN.R-project.org/package=magrittr>

Barth, M. (2022). *tinylabels: Lightweight variable labels*. Retrieved from <https://cran.r-project.org/package=tinylabels>

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. <https://doi.org/10.18637/jss.v067.i01>

Bates, D., & Maechler, M. (2021). *Matrix: Sparse and dense matrix classes and methods*. Retrieved from <https://CRAN.R-project.org/package=Matrix>

Borchers, H. W. (2022). *Pracma: Practical numerical math functions*. Retrieved from <https://CRAN.R-project.org/package=pracma>

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. <https://doi.org/10.18637/jss.v080.i01>

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*(1), 395–411. <https://doi.org/10.32614/RJ-2018-017>

Champely, S. (2020). *Pwr: Basic functions for power analysis*. Retrieved from <https://CRAN.R-project.org/package=pwr>

Dahl, D. B., Scott, D., Roosen, C., Magnusson, A., & Swinton, J. (2019). *Xtable: Export tables to LaTeX or HTML*. Retrieved from <https://CRAN.R-project.org/package=xtable>

Diedenhofen, B., & Musch, J. (2015). Cocor: A comprehensive solution for the statistical comparison of correlations. *PLoS ONE*, *10*(4), e0121945. <https://doi.org/10.1371/journal.pone.0121945>

Eddelbuettel, D., & Balamuta, J. J. (2018). Extending extitR with extitC++: A Brief Introduction to extitRcpp. *The American Statistician*, *72*(1), 28–36. <https://doi.org/10.1080/00031305.2017.1375990>

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, *40*(8), 1–18. <https://doi.org/10.18637/jss.v040.i08>

Ekstrøm, C. T. (2020). *MESS: Miscellaneous esoteric statistical scripts*. Retrieved from <https://CRAN.R-project.org/package=MESS>

Foster, K. B. (2022). *Fddm: Fast implementation of the diffusion decision model*. Retrieved from <https://CRAN.R-project.org/package=fddm>

Fox, J. (2022). *Polycor: Polychoric and polyserial correlations*. Retrieved from <https://CRAN.R-project.org/package=polycor>

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

Ismay, C., & Solomon, N. (2021). *Thesisdown: An updated r markdown thesis template using the bookdown package*.

Jackson, C. H. (2011). Multi-state models for panel data: The msm package for R. *Journal of Statistical Software*, *38*(8), 1–29. Retrieved from <https://www.jstatsoft.org/v38/i08/>

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). Kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, *11*(9), 1–20. <https://doi.org/10.18637/jss.v011.i09>

Kassambara, A. (2021). *Rstatix: Pipe-friendly framework for basic statistical tests*. Retrieved from <https://CRAN.R-project.org/package=rstatix>

Kim, S. (2015). *Ppcor: Partial and semi-partial (part) correlation*. Retrieved from <https://CRAN.R-project.org/package=ppcor>

Kuhn, M. (2022). *Caret: Classification and regression training*. Retrieved from <https://CRAN.R-project.org/package=caret>

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, *82*(13), 1–26. <https://doi.org/10.18637/jss.v082.i13>

Mahmoudian, M. (2020). *Varhandle: Functions for robust variable handling*. Retrieved from <https://CRAN.R-project.org/package=varhandle>

Morey, R. D., & Rouder, J. N. (2018). *BayesFactor: Computation of bayes factors for common designs*. Retrieved from <https://CRAN.R-project.org/package=BayesFactor>

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

Navarro, D. (2015). *Learning statistics with r: A tutorial for psychology students and other beginners. (Version 0.5)*. Adelaide, Australia: University of Adelaide. Retrieved from <http://ua.edu.au/ccs/teaching/lsr>

Neuwirth, E. (2014). *RColorBrewer: ColorBrewer palettes*. Retrieved from <https://CRAN.R-project.org/package=RColorBrewer>

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between JSON data and r objects. *arXiv:1403.2805 [Stat.CO]*. Retrieved from <https://arxiv.org/abs/1403.2805>

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, *6*(1), 7–11. Retrieved from <https://journal.r-project.org/archive/>

Poncet, P. (2019). *Modeest: Mode estimation*. Retrieved from <https://CRAN.R-project.org/package=modeest>

R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>

Revelle, W. (2022). *Psych: Procedures for psychological, psychometric, and personality research*. Evanston, Illinois: Northwestern University. Retrieved from <https://CRAN.R-project.org/package=psych>

Rizopoulos, D. (2006). Ltm: An r package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1–25. Retrieved from <https://doi.org/10.18637/jss.v017.i05>

Robinson, D., Hayes, A., & Couch, S. (2021). *Broom: Convert statistical objects into tidy tibbles*. Retrieved from <https://CRAN.R-project.org/package=broom>

Sarkar, D. (2008). *Lattice: Multivariate data visualization with r*. New York: Springer. Retrieved from <http://lmdvr.r-forge.r-project.org>

Simonsohn, U., & Gruson, H. (2022). *Groundhog: Version-control for CRAN, GitHub, and GitLab packages*. Retrieved from <https://CRAN.R-project.org/package=groundhog>

Singmann, H., Bolker, B., Westfall, J., Aust, F., & Ben-Shachar, M. S. (2021). *Afex: Analysis of factorial experiments*. Retrieved from <https://CRAN.R-project.org/package=afex>

Slowikowski, K. (2021). *Ggrepel: Automatically position non-overlapping text labels with ’ggplot2’*. Retrieved from <https://CRAN.R-project.org/package=ggrepel>

Urbanek, S. (2013). *Png: Read and write PNG images*. Retrieved from <https://CRAN.R-project.org/package=png>

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

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with s* (Fourth). New York: Springer. Retrieved from <https://www.stats.ox.ac.uk/pub/MASS4/>

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

Wickham, H., Henry, L., Pedersen, T. L., Luciani, T. J., Decorde, M., & Lise, V. (2022). *Svglite: An ’SVG’ graphics device*. Retrieved from <https://CRAN.R-project.org/package=svglite>

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

Wickham, H., Hester, J., Chang, W., Müller, K., & Cook, D. (2021). *Memoise: ’Memoisation’ of functions*. Retrieved from <https://CRAN.R-project.org/package=memoise>

Wilke, C. O. (2020). *Cowplot: Streamlined plot theme and plot annotations for ’ggplot2’*. Retrieved from <https://CRAN.R-project.org/package=cowplot>

Wolen, A. R., Hartgerink, C. H. J., Hafen, R., Richards, B. G., Soderberg, C. K., & York, T. P. (2020). osfr: An R interface to the open science framework. *Journal of Open Source Software*, *5*(46), 2071. <https://doi.org/10.21105/joss.02071>

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

Zeileis, A., & Grothendieck, G. (2005). Zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, *14*(6), 1–27. <https://doi.org/10.18637/jss.v014.i06>