University of Basel

Faculty of Psychology

Bachelor's thesis presented to the Department of Psychology of the University of Basel for the degree of Bachelor of Science in Psychology

Testing the Meta-Analytic Stability and Change Model:

A Simulation Approach to Assess Parameter Recovery

Author: Gisin Sabine

Matriculation number: 07-644-164

Correspondence email: sabine.gisin@unibas.ch

Examiner: Professor of the Faculty of Psychology Dr. Mata Rui

Center for Cognitive and Decision Sciences

Submission date:

15.11.2024

Acknowledgement

I would like to thank Dr. Alexandra Bagaïni for providing the original simulation script that I was allowed to adapt for the purpose of the parameter recovery studies described in this thesis.

Declaration of Independent Authorship

I attest with my signature that I have written this work independently and without outside help. I also attest that the information concerning the sources used in this work is true and complete in every respect.
All sources that have been quoted or paraphrased have been marked accordingly.

Additionally, I affirm that any text passages written with the help of AI-supported technology are marked as such, including a reference to the AI-supported program used.

This paper may be checked for plagiarism and use of AI-supported technology using the appropriate software. I understand that unethical conduct may lead to a grade of 1 or “fail” or expulsion from the study program.

Date and Signature

15.11.2024

Abstract

This thesis examines the performance and sensitivity of the Meta-Analytic Stability and Change (MASC) model in evaluating the temporal stability of behavioral patterns and personality constructs through a series of simulation studies. Aimed at enhancing the reliability of meta-analytic research in psychology, the study varies key dataset characteristics, including sample size, maximum retest interval, and the presence of moderating effects such as age and contextual domains. The findings underscore the MASC model's robustness in simpler setups but highlight challenges in complex contexts involving moderating variables. This work helps to provide guidelines for researchers employing the MASC model in meta-analyses, particularly in contexts where moderating effects are significant.

keywords: meta-analytic stability and change model, simulation study, parameter recovery, test-retest, age differences

Testing the Meta-Analytic Stability and Change Model:

A Simulation Approach to Assess Parameter Recovery

Introduction

Can we extrapolate from a person's openness to new experiences at age twenty to their openness as an aging adult, or from a child's self-control to their healthy living as a grown-up? Are individuals who frequently make risky financial decisions more likely to engage in unprotected sex, and will this general risk taking remain a constant over the course of their lives? Can affect be considered more stable over time than well-being? Psychologists may ask questions such as these in an effort to disentangle the temporal development of personality traits, to compare these trajectories over time across constructs, or to better understand the role that different domains such as health or finances play in what might be viewed as an overarching construct (Anusic & Schimmack, 2016; Bagaïni et al., 2023; Mata et al., 2018; Moffitt et al., 2011). Understanding the stability of these traits over time has implications for everything from public education campaigns to economic policymaking, particularly in guiding how individuals might respond to changing life circumstances. Estimating these temporal trajectories, however, is an ongoing challenge and one promising approach to address it is posed by the Meta-Analytic Stability and Change model (MASC; Anusic & Schimmack, 2016), but no published simulation study has yet assessed the accuracy of MASC estimates. With my thesis, I contribute to closing this gap by performing parameter recovery studies that test the MASC model in simulation scenarios founded on constructs central to personality research.

In what follows, I will first introduce the Meta-Analytic Stability and Change model (MASC) by describing Past Work on the Meta-Analytic Stability and Change Model, I will then give a Rationale for Testing Parameter Recovery with Simulation Studies and conclude the introduction with an Overview of the Current Studies.

Past Work on the Meta-Analytic Stability and Change Model

It is undisputed in current research that personality constructs are characterized by both stability and change (Caspi et al., 2005; Costa & McCrae, 2018; Seifert et al., 2022). Quantifying these fundamental aspects is therefore of particular interest, and the importance of meta-analyses in this endeavor has been emphasized, as they provide a means of integrating findings from a range of longitudinal studies across

different retest intervals, countries, and measurement techniques (Bleidorn et al., 2022.; Gurevitch et al., 2018; Roberts et al., 2006).

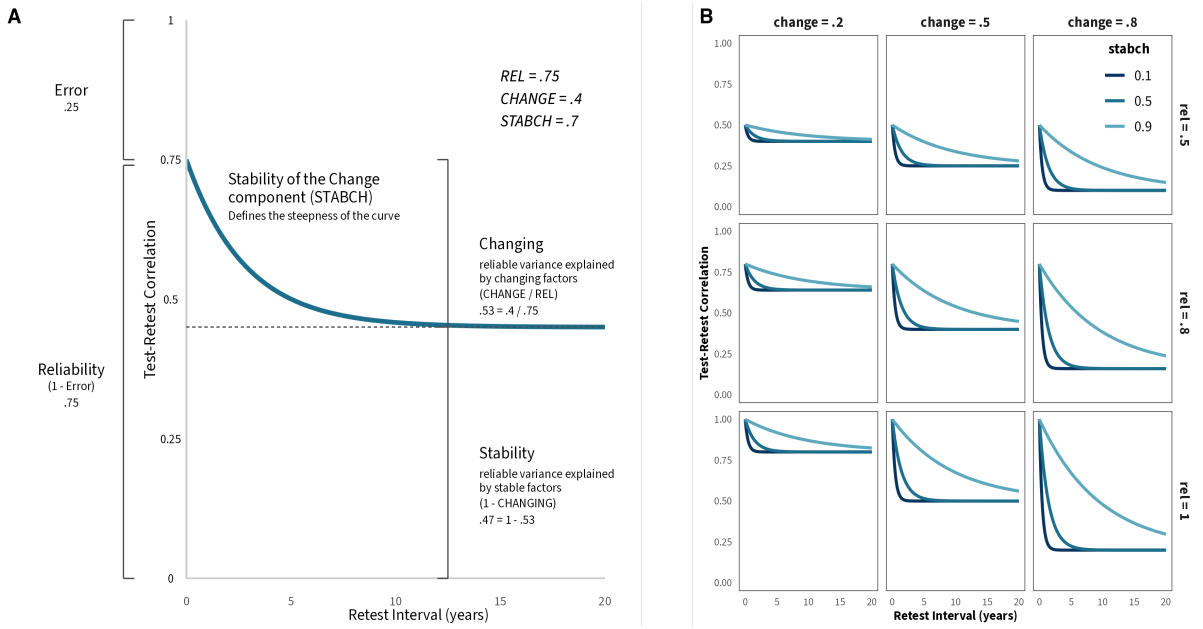
Typically, test-retest correlations are used to measure the consistency of measures over time (or their rank-order stability; see, e.g. Roberts & DelVecchio, 2000). Since these correlations do not account for the length of the retest interval, various attempts have been made to develop models that quantify stability and change as a function of the retest interval. Building on these works, Anusic and Schimmack (2016) proposed the Meta-Analytic Stability and Change (MASC) model. MASC estimates the three parameters *reliability*, *change* and *stability of change* for meta-analytic data. While reliability estimates the proportion of reliable variance as opposed to error variance, change quantifies the amount of reliable variance that can be attributed to changing (as opposed to stable) factors and stability of change accounts for the rate at which that change occurs (Anusic & Schimmack, 2016). By modeling these parameters, MASC provides not only descriptive insights but also an interpretable causal framework for understanding how constructs change over time (see Anusic & Schimmack, Figure 1, p.768). Integrating a term that distinguishes between reliable between-person variance (i.e., true change, as captured by the reliability parameter in MASC) is essential for meta-analytic models. Without this, error variance or measurement error—undisputedly present—may be overlooked, leading to distorted results (Anusic & Schimmack).

MASC describes this temporal development in a curve, shown in Figure 1. It depicts the trajectory that test-retest correlations follow over the years as retest intervals increase and retest correlations decay until they reach a non-zero asymptote, where only the stable part of the reliable measurement variance remains (e.g., a person's childhood environment). As elaborated in Figure 1B, the shape of the curve is determined by the combination of the change and stability of change parameters, while reliability defines its intercept.

In their study, Bagaïni et al. (2023) assess temporal stability of risk preference measures using MASC, thus implementing it in a Bayesian statistical framework—such an approach is recommended when working with a non-linear model. This is especially the case when sample sizes are varying and data is hierarchical which in combination can be problematic in a frequentist framework, but is computationally tractable and conceptually simple in a Bayesian one (Kruschke, 2010; Kruschke & Liddell, 2018; Schad et al., 2021).

Figure 1

MASC-Function for Different Parameter Combinations



Note. Reprinted from Bagaïni, A., Liu, Y., Kapoor, M., Son, G., Bürkner, P., Tisdall, L., & Mata, R. (2023, July 11). Meta-Analyses of the Temporal Stability and Convergent Validity of Risk Preference Measures. https://doi.org/10.31234/osf.io/d7nuj

Past work on MASC has presented formal models that can be empirically fitted to data. However, these approaches have two limitations. First, the results presented by Bagaïni et al. (2023) show considerable heterogeneity in the temporal patterns of risk preference measures, as well as differences in the reliability of the measures. Second, the number of measures per category and the length of maximum retest intervals in these categories is subject to imbalance, as is to be expected of real-world applications of such a model.

Rationale for Testing Parameter Recovery with Simulation Studies

It is critical to understand and assess the robustness of model performance in estimating these parameters, particularly given the heterogeneity and imbalances observed in empirical applications like

those by Bagaïni et al. (2023). As emphasized by Heathcote et al. (2015), testing the adequacy of parameter recovery is an essential step in evaluating model performance. This involves assessing the precision of parameter estimates under varying conditions, either for the available data in an empirical study or for datasets of different sizes and characteristics.

In the latter case, simulation studies are the method of choice for addressing key methodological questions: What sample size is required to detect a given effect? How many measurement waves are needed to achieve a desired level of precision in parameter estimates? As Boulesteix et al. (2020) and Morris et al. (2019) highlight, simulation studies are invaluable tools for systematically exploring these questions, offering insights that are not always accessible through empirical data alone. By treating data simulations as empirical experiments designed to answer methodological questions, such studies allow for controlled testing of model assumptions and performance across a range of plausible scenarios (Siepe et al., 2023).

Overview of the Current Studies

To evaluate the sensitivity and performance of the MASC model in a Bayesian framework, as done by Bagaïni et al. (2023), I designed simulation studies addressing two main questions: First, how accurately can the model estimate its parameters? In a Bayesian context, this involves comparing the posterior mean to the known “true” parameter values. And second, how informative is the model? Specifically, how much does fitting the data reduce prior uncertainty (Schad et al., 2021)?

When deciding on a workflow, I considered two potentially conflicting goals. On the one hand, understanding model performance across different parameter configurations is essential (Morris et al., 2019; Schad et al., 2021; Siepe et al., 2023). On the other hand, the greatest practical value often comes from aligning simulation settings with real-world studies (Heathcote et al., 2015). To reconcile these aims, I implemented a two-step approach involving three simulation studies: Simulation Study 1, part of Step 1, evaluates parameter recovery across different configurations. Simulation Study 2 and Simulation Study 3, forming Step 2, match simulation settings to a real-world meta-analysis (Bagaïni et al., 2023).

Step 1: Intercept-Only Models

I simulated test-retest correlation datasets reflecting the properties of psychological constructs described in the literature (Anusic & Schimmack, 2016; Bagaïni et al., 2023). These datasets varied by sample size, maximum retest interval, and balance (equal vs. unequal numbers of retest correlations across intervals). These baseline models (so-called intercept-only models) excluded moderating effects like age or domain to focus on general parameter accuracy. This step, which aims to assess baseline accuracy and identify deviations is described in Simulation Study 1: General Setup and Accuracy of Intercept-Only Model Scenarios. After the description of the general setup—which sets the base for all simulation setups discussed in this thesis—the central approach for evaluating the accuracy of the recovered parameters is introduced. This so-called *corridor of stability* is based on the work by Schönbrodt and Perugini (2013) and offers a way of defining minimum sample sizes or required length of retest intervals present in a sample to estimate the parameters for a given precision and confidence level. By establishing a foundational understanding of the model's performance under controlled conditions, this study serves as a benchmark for subsequent studies.

Step 2: Simulated Risk Preference Dataset with Moderating Effects

Next, I focused on risk preference, a construct of interest due to its variability across age and domain. Age and domain are fundamental dimensions in understanding the development and expression of psychological traits over time. Age captures the dynamic nature of individual growth, stability and change across the lifespan, reflecting developmental trajectories shaped by both biological and environmental factors (Anusic & Schimmack, 2016). Domain, on the other hand, highlights the contextual specificity of behavior, as individuals may exhibit distinct patterns of stability and change depending on the specific area of life being examined, such as health, finances, or ethics (Bagaïni et al., 2023). Together, these dimensions offer a nuanced lens for investigating how psychological constructs evolve and diversify over time.

My simulation scenarios included fixed age and domain effects with and without interactions, reflecting trends such as the stability peak in middle age or the low reliability of ethical domains compared to smoking or alcohol consumption (Bagaïni et al., 2023; Anusic & Schimmack, 2016). Parameter accuracy in more complex scenarios is assessed in Simulation Study 2: Model Accuracy in Setups with Moderating Effects, again using the corridor of stability introduced above. Building on Simulation Study 1, this study

enlarges the scope and aims to provide insights into scenarios closer to real-world data, where temporal patterns are influenced by respondents' age and the context—or domain—surrounding a given personality construct. By incorporating moderating effects such as age and domain, I evaluate how the MASC model performs under these added complexities, assessing its ability to accurately estimate parameters when influential factors are present.

Finally, in Simulation Study 3: Model Fit in Setups with Moderating Effects I focus on the overall model fit which I discuss by following an approach suggested by Schad et al. (2021). Here, model accuracy is scaled by *posterior contraction*—a value for quantifying uncertainty reduction from the prior to the posterior distribution—which translates to the gain in information that the model can extract from the given data.This further broadens the scope by moving from the accuracy assessment of each individual parameter to evaluate the gain in information that the model as a whole can provide.

Scope and Contributions of the Current Research

All three simulation studies are described in a method, result and a brief discussion section. The Generall Discussion in the last part of this thesis integrates the insights from all individual studies.

My thesis helps to evaluate how accurately the MASC parameters can be estimated under different conditions, providing a clearer sense of how much confidence we can have in values reported in past research, and more importantly, these insights aim to guide the planning of future work that will likely consider increasingly specific constructs and more detailed interactions. By understanding the strengths and limitations of the MASC model, this thesis will hopefully contribute to refining its application and expanding its utility in exploring stability and change in personality constructs.

Simulation Study 1: General Setup and Accuracy of Intercept-Only Model Scenarios

The first simulation study aims to establish a foundational understanding of the MASC model's performance under controlled conditions. By focusing on intercept-only models without moderating effects, the baseline accuracy of parameter estimation can be assessed. This study serves as a benchmark for subsequent simulations, helping to identify any inherent biases or limitations in the model's ability to

recover known parameters. Specifically, I expect to determine the sample sizes and retest intervals required to achieve desirable levels of precision in parameter estimates.

To structure this assessment, I drew inspiration from Schönbrodt and Perugini's (2013) concept of the corridor of stability, which they used to evaluate at what sample sizes simulated parameter estimates stabilize around the true values. By defining corridors with specific widths and confidence levels, they provided a nuanced way to discuss parameter accuracy across different sample sizes. Adopting a similar approach, I constructed corridors around the true parameter values to systematically evaluate the MASC model's accuracy. This methodology helps in identifying the minimum data requirements and contributes to a more precise understanding of the model's capabilities in estimating parameters accurately.

Method

Meta-Analytic-Stability-and-Change Model (MASC)

MASC is a non-linear model introduced by Anusic and Schimmack (2016) for the meta-analytic analysis of longitudinal data. Using test-retest correlations r_{t2-t1}, it estimates three parameters: *rel* (reliability), the proportion of total between-person variance that can be attributed to systematic variance as opposed to measurement error or noise, *change*, the proportion of this variance that is subject to change over time, and *stabch* (stability of change), the rate at which this change occurs. MASC is formalized as:

$$r_{t2-t1} = rel \times (change \times (stabch)^{time} - 1) + 1)$$

Data Simulation and Bayesian Model Specifications

To test model accuracy, datasets of different sizes and with different maximum retest intervals were simulated in the R statistical environment (R Development Core Team., 2024). Test-retest correlations were computed from a set of true values, using the MASC function and adding normally distributed noise (mean = 0, sigma = 0.1).

"True" parameter values for the intercept-only models were selected based on findings from the literature to represent a range of psychological constructs with differing temporal dynamics (Anusic & Schimmack, 2016; Bagaïni et al., 2023). Specifically, I chose personality, affect, and risk preference due to

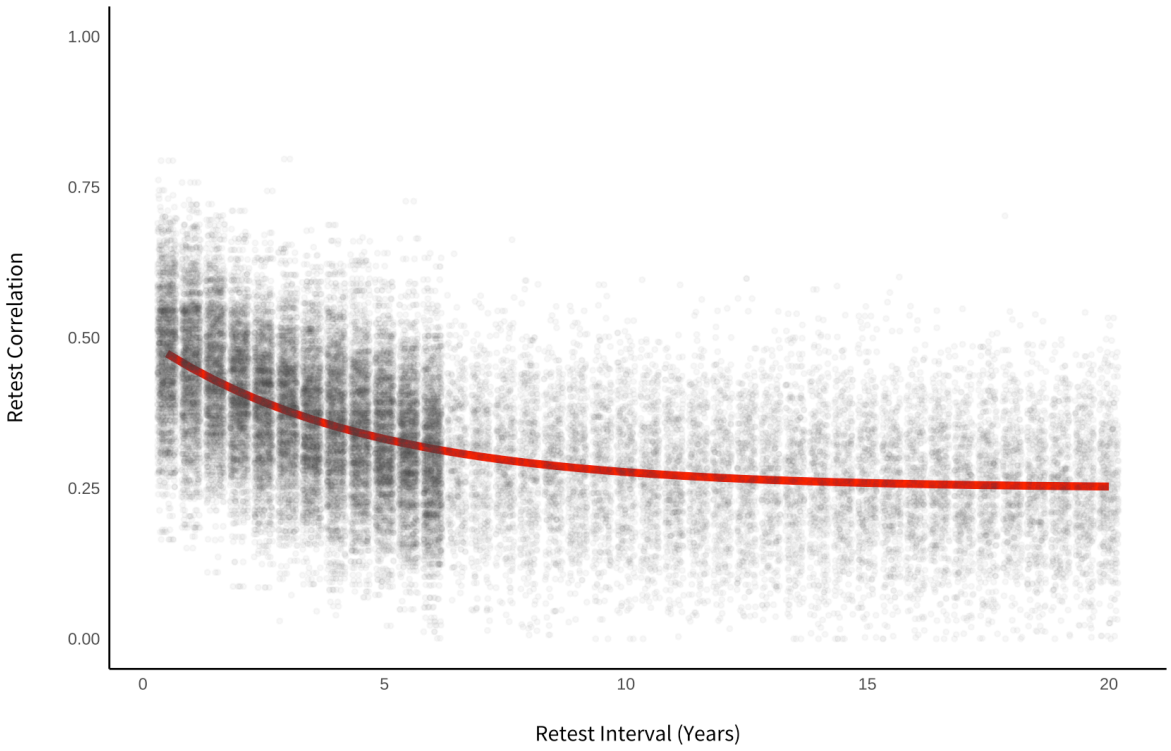
their varying levels of reliability, change, and stability of change, which allows for testing the MASC model across diverse scenarios. For personality, I set rel = 0.7, change = 0.2, and stabch = 0.25, reflecting Anusic and Schimmack's (2016) findings that personality traits have high reliability, moderate change, and low stability of change, indicating that changes occur but stabilize relatively quickly. For affect, I used rel = 0.6, change = 0.6, and stabch = 0.9, based on literature suggesting that affect has lower reliability, higher levels of change, and high stability of change (Anusic & Schimmack, 2016). This combination reflects constructs that change significantly but do so slowly over time. Risk preference was assigned rel = 0.5, change = 0.5, and stabch = 0.8, informed by Bagaïni et al. (2023), who found moderate reliability and change with relatively high stability of change. This represents constructs that exhibit moderate changes occurring steadily over time. By selecting these parameter values, I aimed to simulate datasets that capture the diversity of temporal patterns found in psychological constructs, providing a robust test of the MASC model's ability to recover parameters accurately across different conditions. For each setup, an example dataset was plotted before the simulation to visually confirm sanity (see Figure 2).

Datasets were simulated for n_{cor} = 25, 50, 500, 1000, 2000, 3000, 5000 and 10,000, covering a wide range of sample sizes to assess the MASC model's performance across different data availability scenarios. The selection of these values serves three purposes: First, n_{cor} = 25 and 50 represent relatively small datasets common in meta-analyses where limited studies are available. Testing these sizes allowed me to evaluate the minimum data requirements for reliable parameter estimation and to explore the lower boundaries of the model's effectiveness. Second, including intermediate sizes helps to observe the gradual improvement in parameter recovery as the sample size increases. This incremental scaling provides insights into how the model's accuracy evolves with additional data, highlighting any nonlinear effects in parameter estimation precision. Third, n_{cor} = 5,000 and 10,000 represent large datasets that, while less common, are increasingly feasible with the growing availability of large-scale studies and data-sharing practices. Simulating these upper limits helps to determine if and when additional data yield diminishing returns in estimation accuracy, informing researchers about the practicality of pursuing extremely large sample sizes.

In a first round, all datasets contained the same number of test-retest correlations for all retest intervals from 0.5 to 20 years in steps of 0.5 years (balanced datasets). The choice of a 20-year maximum

Figure 2

Simulated Data Set for Risk Preference (Unbalanced)



Note. For unbalanced data sets 2/3 of the data points have a retest interval of less than 6 years.

retest interval aligns with previous studies (Bagaïni et al., 2023) and represents a realistic upper limit for longitudinal psychological research.

In a second round, 2/3 of the retest correlations had retest intervals of less than 6 years (unbalanced datasets)—reflecting the pattern described by Bagaïni et al. (2023). As these produced comparable results while being closer to real world data, the different sized datasets described in this thesis are unbalanced.

Once the sample size needed for desirable model accuracy was determined, it was fixed, and datasets were simulated for maximum retest intervals ranging from 1 to 20 years in steps of 0.5 years. These datasets were balanced.

Following the approach of Bagaïni et al. (2023), I adopted a Bayesian framework to estimate the MASC parameters with the brms package (Bürkner, 2017). At this stage, the model accounted for no moderating effects. The values were obtained using Monte Carlo simulation and extracted as the mean of the posterior distribution. The distance to the true values was then calculated.

Priors were set in a non-informative way—as the goal was to allow a wide range of possible parameter estimates—and mirrored the priors used by Bagaïni et al. (2023) in their analysis. To ensure the robustness of the results while keeping the computational cost manageable, 50 simulation rounds were run for each setup.

Code Availability

All simulation scripts as well as figures and tables used for the evaluation of results referenced in this thesis can be reproduced running the code provided in Appendix C.

Assessing Model Accuracy

Following the approach of Schönbrodt and Perugini (2013), who constructed a corridor of stability around true parameter values to assess at what sample size simulated estimates remain within acceptable bounds, I defined similar corridors to evaluate model accuracy in my simulations. To ensure that these corridors reflect realistic ranges for parameter estimates in personality research, I chose their widths based on confidence intervals (CIs) and highest density intervals (HDIs) reported in the literature. Specifically, I defined three corridor half-widths: ±0.07, ±0.12, and ±0.2—meaning that for a true parameter value of 0.5, the estimated values must fall within [0.43, 0.57], [0.38, 0.62], and [0.3, 0.7], respectively, to meet the accuracy criteria. A deviation less than or equal to 0.07 represents ideal precision and aligns with ranges observed in empirical studies. For example, Anusic and Schimmack's (2016) reliability estimate for personality traits is $M = 0.72$ with a 95% CI of [0.65, 0.78], and Bagaïni et al.'s (2023) estimate for smoking frequency is $M = 0.84$ with a 95% HDI of [0.78, 0.90]. These intervals reflect narrow ranges similar to my strictest corridor.

Conversely, a deviation of 0.2 may seem large but remains more conservative than some reported estimates. For instance, Bagaïni et al. (2023) found a reliability estimate for the ethical domain with *M* = 0.64 and a wide 95% HDI of [0.36, 0.91]. This suggests that even broader corridors are applicable in certain contexts. Although the CIs and HDIs informing these corridor widths are influenced by the sample sizes of the original studies, they provide valuable reference points.

Confidence levels were defined based on the percentage of estimates from all 50 simulation rounds that remained within the corridor. Results are primarily reported for the 95% confidence level, as it provides a stringent criterion for model accuracy, but calculations for the 90% and 80% levels were also performed and can be found in the appendices.

Results

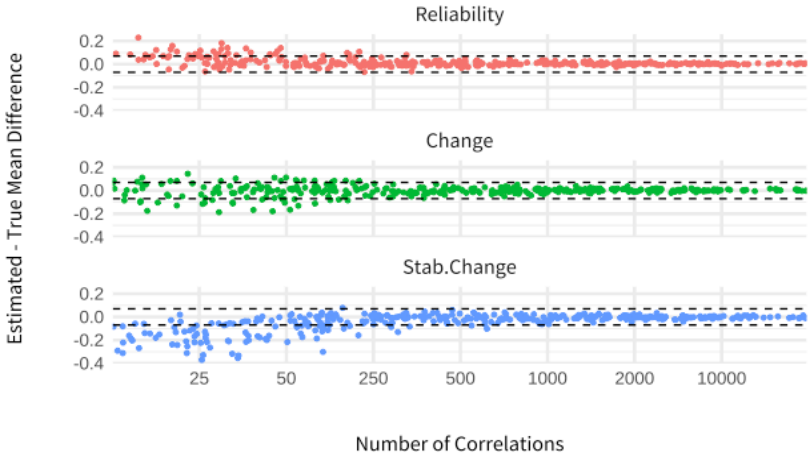
Corridor of Stability: Risk Preference and Affect Dataset

For the simulated risk preference dataset, the accuracy criteria are largely met and the differences between true and estimated values remain within the corridor of stability, even with a strict threshold such as the "ideal corridor" defined by a half-width of ±0.07, but a closer look at the estimated parameter values shows that the model tends to underestimate the change and especially the stability of change parameter for small datasets (see Figure 3). For this parameter, at n_{cor}= 25 only a small portion of estimated values are inside the stability corridor (18%), at n_{cor}= 50 about half of the estimates (52%) are inside the corridor, and starting from n_{cor}= 250 a clear majority (>86%) are inside the corridor (see Appendix A, Table A1).

The reliability parameter is the easiest to estimate accurately, requiring n_{cor}= 250 for an accurate estimate (with a corridor half-width of ±0.07 and a 95% confidence level). For the change and stability of change parameters, n_{cor} = 500 and n_{cor}= 1,000, respectively, are needed to meet the accuracy criterion (see Appendix A, Table A2). The results for the simulated affect dataset generally align with these estimates, but for the change parameter a relatively high number of datapoints (n_{cor}= 2,000) is required to meet the accuracy criterion (Appendix A, Figure A3, Tables A4 and A5).

Figure 3

Risk Prefence Dataset: Corridor of Stability w = ±0.07



Note. Differences between true and estimated parameter values are shown for the unbalanced risk preference dataset. The three panels represent reliability, change, and stability of change parameters, with estimates plotted against the number of test-retest correlations. Dashed lines indicate the ±0.07 corridor.

Corridor of Stability: Personality Dataset

Although following the trends described for the risk preference and affect setups, the stability of change parameter is overestimated rather than underestimated and does not fully meet the strictest accuracy criterion, which expects estimated parameter values to be within ±0.07 of the true value. For this criterion, at n_{cor}= 50 only 8% of the estimated values for stability of change are within the corridor and 88% for n_{cor}= 10,000, failing to meet the required 95% (see Appendix A, Figure A6, Tables A7 and A8).

Maximum Retest Intervals Required for Accurate Estimates

Simulations of the affect, personality, and risk preference constructs showed that MASC parameters could be estimated with moderate precision (95% within ±0.12) when the maximum retest interval was between 3 and 5 years. For high precision (95% within ±0.07), a maximum retest interval of 10 to 12 years was required. In all simulation scenarios, the reliability parameter was the first to meet the accuracy criterion. With a sufficiently large dataset, it was accurately estimated even with a maximum retest interval of just one year. For the affect and risk preference constructs, the parameters for change and stability of change were underestimated when the maximum retest interval was below 3 to 5 years. In contrast, for the personality construct, these parameters were overestimated under the same conditions. In particular, for the personality construct, the stability of change parameter did not fully meet the accuracy criterion even with maximum retest intervals of up to 20 years (see Appendix A, Figures and Tables A9—A14).

Discussion

A clear pattern emerges across the simulations: The reliability parameter is consistently the most accurately estimated, regardless of parameter values or retest interval scenarios. For most intercept-only models, also the change and stability of change parameter can be estimated very accurately, when datasets with n_{cor}= 1,000 to 2,000 are provided. This demonstrates a baseline accuracy, showing that the MASC model can effectively estimate its parameters across different true parameter values when equipped with moderately large datasets.

For maximum retest intervals, measurement waves spaced 10—12 years apart guarantee high precision. This aligns with theory suggesting that data should be available at least until the changing part of the retest correlations has diminished and only stable factors remain, which, for most constructs will be true after 10—12 years.

A deviation from these patterns presented the stability of change parameter in the personality construct, which didn't meet the accuracy criteria until n_{cor} = 10,000 or a maximum retest interval of 20 years. The explanation for this behavior might be found in the combination of parameter values that characterize the construct. As a low proportion of change meets a low stability of change—resulting in a

steep curve—the asymptote, where only the stable part of the variance remains, is quickly reached. Thus, even with large datasets, only a fraction of the data can account for the steepness of the curve. A similar situation is mentioned by Schad et al. (2021) for drift-diffusion models, where one of the model parameters may depend on a small number of data points despite a very large overall data set. This explains why the parameter in question may be estimated with greater uncertainty than the overall sample size would suggest.

A possible solution in cases where a rapid decrease in changing factors is expected is to set more informative priors, which would reduce the amount of data needed to confirm low stability of change. More generally, the results confirm that estimates tend to revert toward the prior (set at $M = 0.5$ with $SD = 0.73$) unless sufficient data are available to inform the model of a more extreme value (see, e.g., McElreath, 2016). This highlights the importance of using more informative priors—and, by extension, more theoretically grounded prior knowledge—in situations where only smaller datasets are available.

Simulation Study 2: Model Accuracy in Setups with Moderating Effects

Building upon the findings from Simulation Study 1, which assessed the MASC model's parameter recovery in intercept-only scenarios, Simulation Study 2 aimed to evaluate the model's performance in more complex where moderating effects are present. Specifically, this study examined how the inclusion of age and domain as moderators affects the accuracy of parameter estimation in the MASC model.

The main moderators investigated in the literature for the MASC model were age, domain, and gender (Anusic & Schimmack, 2016; Bagaïni et al., 2023). However, gender was consistently reported to have no detectable effect on the stability of psychological constructs. Although gender differences may exist within the constructs themselves—for instance, in risk preference (Liu et al., 2023; Mata et al., 2016)—these differences seem to not impact the parameters of stability and change as modeled by MASC. Therefore, gender effects were not simulated in this study.

To create realistic simulation scenarios for age and domain patterns, I incorporated three central tendencies described in the literature. First, many personality constructs show a peak of temporal stability in midlife, resulting in an inverted U-shaped relationship between test-retest correlations and age (Bagaïni

et al., 2023; Bleidorn et al., 2022; Seifert et al., 2022). Second, as highlighted by Bagaïni et al. (2023), there appears to be a negative linear effect of age on stability, with the proportion of changing factors increasing as individuals enter old age. Third, reliability often varies across domains within a construct, leading to differences in the intercept values of retest trajectories (Bagaïni et al., 2023).

These main effects are additive in nature and were simulated accordingly. By keeping the noise level and prior settings consistent with those in the intercept-only scenarios from Simulation Study 1, I ensured a valid comparison between studies. The model was set up to account for additive effects of age and domain.

However, interactions between moderating effects have also been observed in the literature. Based on the predictions for the three model parameters reported by Bagaïni et al. (2023), interactions between age and domain effects appear to be present, though they do not follow a consistent pattern. To explore this further, I simulated a second scenario incorporating similar interactions. Specifically, I added interaction terms between quadratic age and domain to examine their influence on both the change and stability of change parameters.

By simulating both additive and interactive moderating effects, Simulation Study 2 aimed to assess the MASC model's ability to accurately estimate parameters in more complex scenarios that resemble real-world data. Understanding how these moderating effects impact parameter recovery is crucial for researchers who wish to apply the MASC model to complex datasets where temporal patterns are influenced by both individual factors (like age) and contextual factors (like domain), as well as their interactions.

Method

In Simulation Study 2, I adapted the dataset sizes to better suit the increased complexity introduced by moderating effects and their interactions. Compared to Simulation Study 1, I omitted the $\boldsymbol{n_{cor}}$ = 25 dataset because such a small number of test-retest correlations is less informative in scenarios involving moderating effects such as age and domain. In contrast, I additionally included $\boldsymbol{n_{cor}}$ = 15,000 and—for the interactive setup—$\boldsymbol{n_{cor}}$ = 20,000 to account for the possibility that more complex models may require very large sample sizes to achieve accurate parameter estimates. Further, the use of large amounts of data is empirically grounded: Bagaïni et al. (2023) analyzed a database comprising approximately 72,000 test-retest

correlations. Although they pooled the data and fitted the model separately for three categories—effectively reducing the sample size for each model-fitting—their work underscores the relevance of large datasets in this area of research. Therefore, I aimed to reflect the scale of data encountered in such comprehensive studies and to explore whether increasing the sample size leads to a plateau-effect or further enhances the accuracy of parameter estimates in complex models.

Main Moderating Effects

The model was set up to account for additive effects of age on the parameters change and stability of change by adding or subtracting weights relative to a baseline age of 40 years. This baseline represents midlife, a period where many psychological constructs reach peak stability (Bleidorn et al., 2022; Seifert et al., 2022). For the change parameter, the weights were 0.04 for quadratic age and 0.035 for linear age. Quadratic age was modeled to have an effect of -0.03 on the stability of change parameter. These weights reflect empirical findings suggesting that both linear and non-linear age effects influence how constructs change over time, with stability increasing as individuals grow-older, until mid-life, and decreasing again, as they age. Bagaïni et al. (2023) highlight that old age is associated with even lower stability than adolescence and young adulthood, differing from the ever-increasing stability that Anusic & Schimmack (2016) report. I follow the former in my modeling to capture the observed decrease in stability during older adulthood.

The reliability parameter was modeled to be influenced by quadratic age—to form the tipical U-shaped pattern—plus domain-specific weights, ranging from 0.4 to 0.65 compared to the baseline value of 0.5. This variation captures differences in measurement reliability across domains, as observed in previous research (Bagaïni et al., 2023).

To verify the adequacy of the simulated dataset, the plot showing parameter values for different domains and age bins was recreated from the example in Bagaïni et al. (2023) and compared (see Appendix B, Figure B1). This comparison confirmed that the simulated data captured the central patterns observed in empirical studies, supporting the validity of the simulation approach.

Interactions between Moderating Effects

An overview of the parameter values used by Bagaïni et al. (2023) in their meta-analysis shows that, in addition to central features such as different reliability across domains, the inverted U-shape of the retest correlations as a function of age, there are also interaction effects between age and domain (Figure 4B). To explore this, I simulated a scenario incorporating the main moderating effects described above, adding weights for interaction terms between quadratic age and domain to examine their influence on both change and the stability of the change parameter. The aim of my simulation was to model these patterns and create a setup with comparable variability of parameter values when grouped by age and domain. Visual inspection confirmed that the central patterns were present and that there was realistic parameter variability, particularly for the parameters change and stability of change across domains (Figure 4A).

To further ensure the validity of the simulated data, I recreated central plots of the original datasets which visually confirmed that they shared important tendencies with the results presented by Bagaïni et al (2023). Specifically, I plotted retest trajectories for different age groups as a function of the retest interval and for different retest intervals as a function of age (see Appendix B, Figures B4 and B5). These sanity checks focused on the category propensity, as Bagaïni et al. reported clear trends in this part of the data.

Results

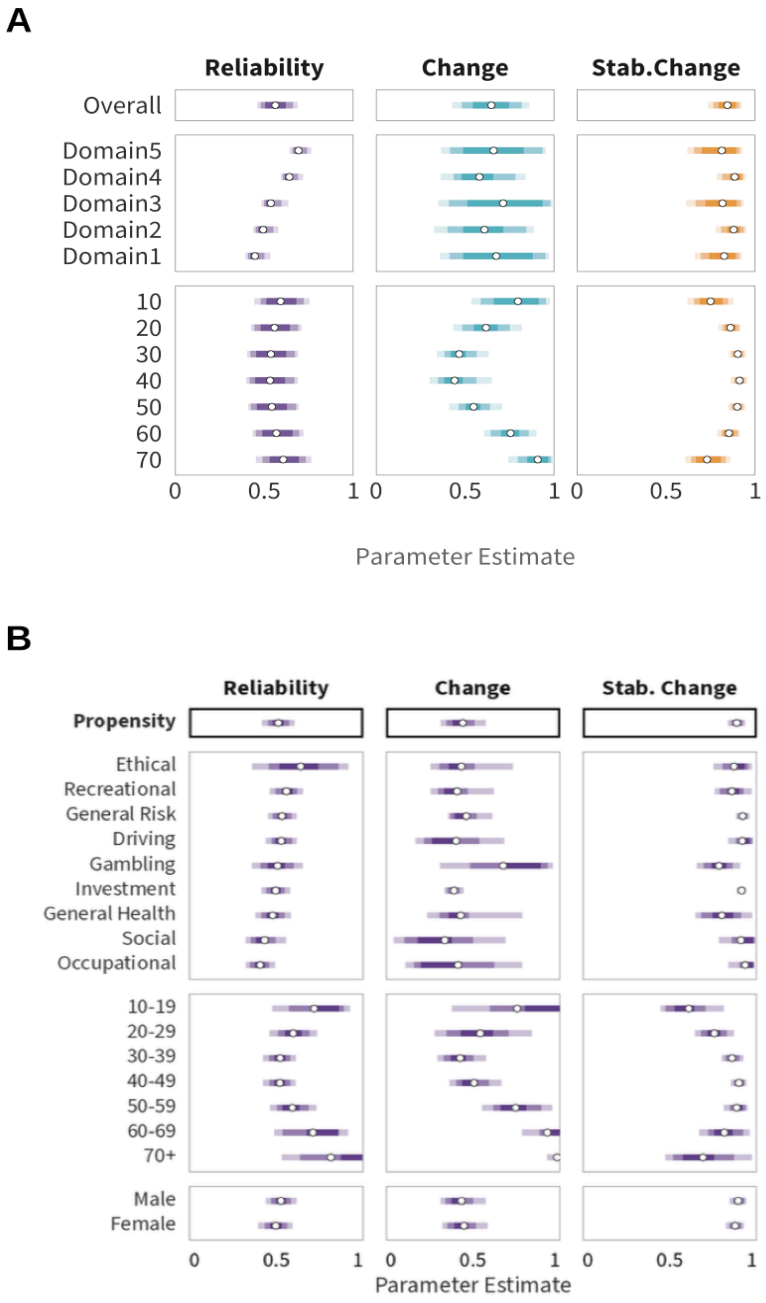
Additive Model with Age and Domain Effects

When age and domain effects are present and estimated by the non-linear MASC model, complexity increases and estimates for smaller datasets become much less precise (see Appendix B, Figure B1 and Tables B3 and B4).

At n_{cor} = 50, only 32% of the reliability estimates, 48% of the change estimates, and 14% of the stability of change estimates fell within the corridor fences of ±0.07. At n_{cor} = 500, 75% of the reliability and change estimates and 56% of the stability of change estimates met this accuracy threshold.

Figure 4

Comparison of Simulated (A) vs. Original Data (B)



Note. 7B Reprinted from Bagaïni, A., Liu, Y., Kapoor, M., Son, G., Bürkner, P., Tisdall, L., & Mata, R. (2023, July 11). Meta-Analyses of the Temporal Stability and Convergent Validity of Risk Preference Measures. https://doi.org/10.31234/osf.io/d7nuj

For large datasets with n_{cor} > 5,000, the precision was high and increased slightly (from 93% to 96% for the stability of change parameter). And importantly, the model met the strictest accuracy criteria (95% of all estimates within ± 0.07 for all three parameters) at n_{cor} = 10,000 (see Appendix B, Tables B2 and B3).

Setup with Interactions of Age and Domain

When examining the differences between true and estimated parameter values, this setup followed the additive pattern described above, where small datasets resulted in parameter estimates that spanned almost the entire range of possible values, but became more precise starting at n_{cor}= 2,000. However, Figure 5 shows that an important distinction emerged: This pattern held for the reliability and stability of change parameters, while the change parameter did not meet the accuracy criterion. Even with very large datasets of n_{cor}= 15,000 and 20,000 only about 80% of the estimates for the change parameter fell within the ±0.07 corridor, instead of the desired 95% (see Appendix B, Tables B7 and B8).

Discussion

The inclusion of additive and, particularly, interaction effects pose significant challenges to precisely estimating the parameters of the MASC model.

Such complexity is expected in real-world data. For example, we would not expect the trajectory of risk preferences for a person in their twenties deciding whether to use recreational drugs to mirror that of a fifty-year-old making stock market decisions. This illustrates that intercept-only models may not capture the nuanced variations within constructs like risk preference across different ages, meaning that modeling and evaluating moderating effects such as age and domain is crucial.

Figure 5

Interactive Effects Dataset: Corridor of Stability w= ±0.07

A closer examination of the individual parameters revealed that while the reliability parameter was consistently estimated accurately across all simulation setups, it required larger sample sizes as model complexity increased. In contrast, the results for the change and stability of change parameters were more varied. In the setup with only additive effects, a very large sample size (n_{cor} = 10,000) was sufficient to estimate all parameters within the predefined accuracy criteria. However, this was not the case for the setup involving interactive moderating effects. In this scenario, the change parameter could not be fully recovered, even with increased sample sizes.

Although the change parameter was influenced by the most effects—two additive and one interactive—compared to the stability of change parameter (one additive and one interactive) and the reliability parameter (two additive), the additional additive effect was already present in the additive setup and did not lead to less accurate estimates or necessitate larger samples compared to the other parameters.

This suggests that the change parameter may have become "overloaded" with effects in the interactions setup—a situation however, that could plausibly occur with real-world data.

Importantly, in the setup with interactive moderating effects, increasing the sample size beyond n_{cor} = 15,000 did not improve the estimation accuracy of the change parameter—it plateaued between n_{cor} = 15,000 and 20,000. This observation indicates that there is likely a point where adding more data cannot compensate for the challenges posed by high model complexity. Therefore, in highly complex models with multiple interacting effects, even large sample sizes may not suffice to achieve precise parameter estimates for certain parameters.

Simulation Study 3: Model Fit in Setups with Moderating Effects

Building upon the findings from the previous simulation studies, Simulation Study 3 aimed to evaluate the performance of the MASC model in complex scenarios involving moderating effects. Specifically, this study examined how well the model recovered true parameter values and how informative the data were in reducing uncertainty when moderating effects were present.

To comprehensively assess the model's performance at given sample sizes, it is essential to address two critical questions, as suggested by Schad et al. (2021): First, how well does the estimated posterior mean match the true value used to simulate the data? Second, how much is the uncertainty reduced from the prior to the posterior distribution?

While the first question focuses on the accuracy of parameter estimates, the second evaluates the informativeness of the data in terms of variance reduction. By investigating both aspects, this study provides a thorough evaluation of the MASC model's fit in setups with moderating effects.

Method

To address the first question of parameter recovery, the posterior difference was calculated for each parameter:

$$posterior\ difference = \mu\_post - \tilde{\theta}$$

For the second question regarding uncertainty reduction, the posterior contraction was computed:

$$s = 1 - \frac{\sigma^2_{post}}{\sigma^2_{prior}} = \frac{\sigma^2_{prior} - \sigma^2_{posterior}}{\sigma^2_{prior}}$$

Where σ^2_{post} is the variance of the posterior distribution, while σ^2_{prior} is the variance of the prior distribution, obtained by simulating 100,000 data points from the prior.

Ideally, the posterior contraction *s* should be close to one, indicating that the data is highly informative.

Negative values for posterior contraction can occur when the posterior variance is larger than the prior variance, meaning that uncertainty is increased from the prior to the posterior, which is not a desirable scenario but may be the case when we have too little data to reliably estimate some or all model parameters.

Both the posterior difference and posterior contraction were plotted against each other for each parameter, creating a two-dimensional diagnostic tool to assess model performance. In an ideal model fit:

- Posterior difference ≈ 0: The estimated parameters match the true values.
- Posterior contraction ≈ 1: There is a significant reduction in uncertainty.

This combined metric allows for a comprehensive evaluation of the model's ability to recover true parameters and the informativeness of the data.

Results

How do the more complex setups of the MASC model fit into this framework? To evaluate the performance for a given sample size, all n_{cor} = 50 parameter estimates were plotted on the above-described grid—Figure 6 illustrates the respective posterior differences as a function of posterior contraction for the additive and interaction setups with age and domain effects.

At n_{cor} = 50, the posterior differences ranged from -0.82 to +0.45 (*M* = 0.15, *SD* = 0.11), indicating non-ideal parameter recovery, and values below zero for posterior contraction were frequent, that is, information about the parameter deteriorated from the prior to the posterior. As a result, the overall sensitivity of the model was very poor.

At n_{cor} = 200, negative values for posterior contraction were still present, though less frequent , and were concentrated on estimates for the stability of change parameter. Most estimates showed stronger posterior contraction (*M* = 0.63, *SD* = 0.32), while posterior differences ranged from -0.5 to +0.43. Together, these results still indicate suboptimal model sensitivity.

From n_{cor} = 1,000, two key observations emerged. First, the posterior contraction was entirely on the positive side of the scale. While a contraction below 0.5 still represents a modest gain in insight, most estimates exceeded this value, particularly in larger datasets. There was also a clear trend: Lower posterior contraction values were associated with the stability of change parameter in the additive setup and the change parameter in the interaction setup, whereas estimates for the reliability parameter were more accurate in both cases. Second, the posterior differences were significantly reduced, ranging from -0.3 to +0.2 at n_{cor} = 1,000, and from -0.1 to +0.1 at n_{cor} = 15,000. In other words, the parameters were recovered with high accuracy, and the marked decrease in uncertainty, with values nearing one, indicated that the overall model sensitivity was very strong to ideal for datasets above n_{cor} = 3,000.

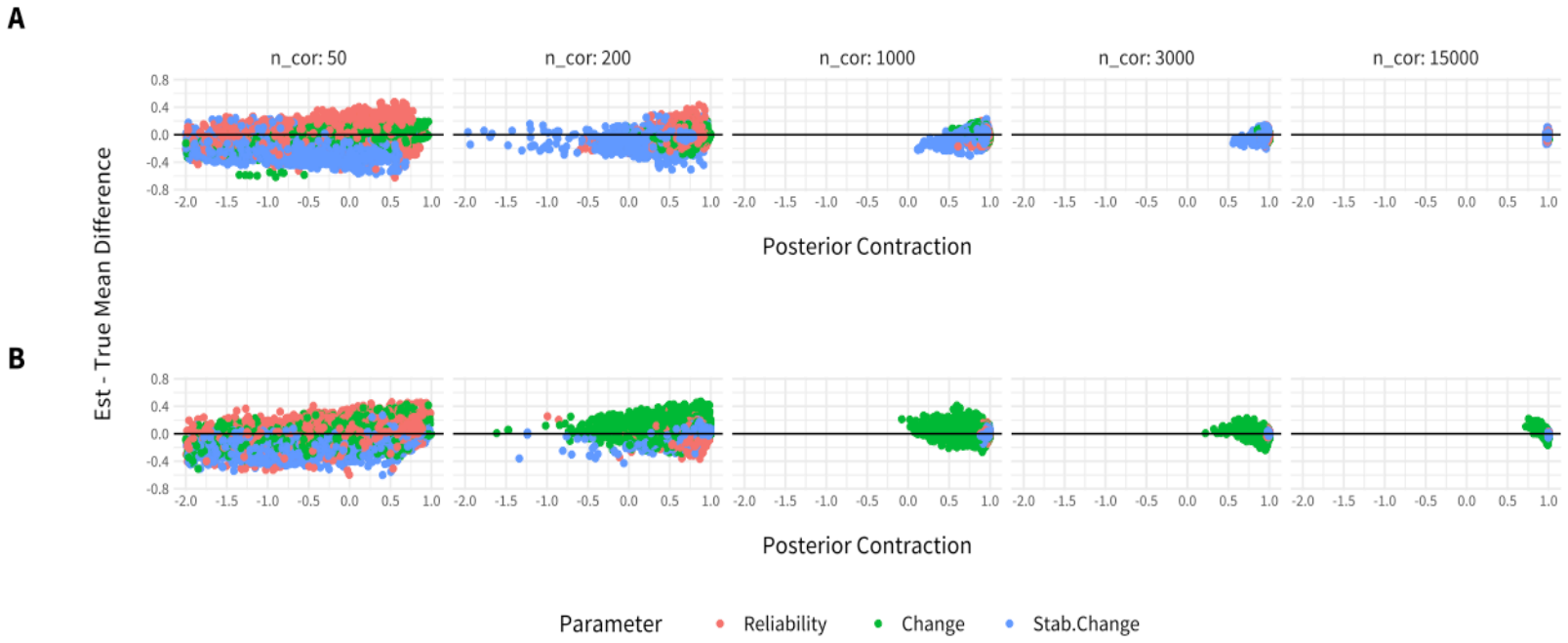
Discussion

The results of Simulation Study 3 demonstrate that the performance of the MASC model in setups with moderating effects of age and domain is highly dependent on sample size. For small datasets (n_{cor} = 50), the model exhibited poor parameter recovery, with posterior differences ranging widely and frequent negative posterior contractions, indicating that uncertainty increased from the prior to the posterior. This suggests that the model lacks sensitivity when data are limited and the complex moderating effects are present.

As the sample size increased to n_{cor} = 1,000 and above, two key observations emerged. First, the posterior contraction became entirely positive, signifying that the data provided sufficient information to reduce uncertainty from the prior to the posterior distribution. Second, the posterior differences narrowed significantly, indicating improved accuracy in parameter estimation. Notably, the reliability parameter was

Figure 6

MASC Model Fit for Different Sample Sizes



Note. Panel A shows the MASC model fit with additive effects for age and domain, while Panel B shows the model fit with interactions between age and

consistently estimated more accurately than the change and stability of change parameters across all sample sizes.

These findings emphasize the critical role of sample size in the performance of the MASC model, particularly in the presence of complex moderating effects. While small datasets may be insufficient for accurate parameter estimation, the challenges persist even with larger datasets for certain parameters. The difficulties in recovering the change and stability of change parameters suggest that these aspects of the model are more sensitive to complexity and may require larger sample sizes or alternative modeling strategies.

In practical terms, researchers should be aware that larger datasets are essential when working with complex models involving moderating effects. When obtaining large samples is not feasible, leveraging informative priors based on existing literature becomes crucial. This Bayesian approach can help improve model performance without the need for additional data—a significant advantage in fields like psychology and behavioral sciences, where data collection can be resource-intensive (Kruschke & Liddell, 2018; Schönbrodt & Perugini, 2013).

General Discussion

The aim of my thesis was to contribute to a nuanced understanding of the sensitivity of parameter estimates provided by the MASC model and to assess the model fit across various setups, especially in scenarios where effects of age and context—such as financial or health domains—are present. This focus is particularly relevant for empirical work, as the influence of such moderating variables is common in psychological research.

I ran a series of data simulations while systematically varying dataset characteristics such as sample size, maximum retest interval and presence or absence of moderating effects. Model and dataset setups relied on previous work (Anusic & Schimmack, 2016; Bagaïni et al., 2023), and the plausibility of datasets with moderating effects was carefully evaluated by recreating plots showing dataset characteristics from the meta-analysis conducted by Bagaïni et al. (2023) that served as a main reference point. When taking decisions on the type of metrics to rely on to evaluate results, I followed suggestions made by relevant publications (Schad et al., 2021; Schönbrodt & Perugini, 2013). My work can help to provide a guideline for

researchers desiring to use the MASC model for their meta-analyses on the temporal stability of behavioral patterns or personality constructs—be it for the purpose of looking at well-known dimensions like extraversion from a different angle or be it for investigating novel constructs such as climate fear or susceptibility for conspiracy theories.

In what follows, I will first present the Main Findings of the three simulation studies discussed in this thesis. Next, I will elaborate on Understanding the Findings of the Current Studies by discussing the implications of parameter uncertainty in the MASC model. Specifically, I will provide a theoretical illustration of its curve trajectories, including so-called uncertainty bands—ranges where alternative curve paths could traverse, considering the interactions of all parameter uncertainties. I will then address the Limitations of the Current Simulation Studies and conclude with the Conclusion and Implications for Future Research.

Main Findings

When integrating results from all data simulations performed to test the MASC-model, four main observations emerged.

First, my results suggest that in a scenario where no moderating effects are expected, MASC model estimates are very precise, surpassing the requirements I had defined based on the uncertainties reported in previous work. Dataset imbalance did not pose an issue, which led me to omit the results of perfectly balanced intercept-only models, as they can be considered too ideal and thus of little practical relevance.

Second, a rule of thumb emerged when varying the maximum retest interval: The time between the first and last measurement wave in at least some of the original studies included in the meta-analysis should be 10 to 12 years. When designing longitudinal panels, findings such as these may be relevant because following up participants over decades is costly and can be difficult because of the natural accumulation of dropouts due to changing life circumstances, declining motivation to contribute to research, illness, or death.

Third, when accounting for the effects of age and domain, the results became more mixed. While the proportion of reliable variance relative to error variance (parameter reliability) was precisely estimated across all simulation setups, estimating the proportion of this reliable variance that changes over time

(parameter change) and the rate at which this change occurs (parameter stability of change) was associated with greater uncertainty. The model fit results for datasets with moderating effects corroborated this observation. Since estimating these two parameters is central to the model, it is essential to evaluate whether the complexity of the data and prior knowledge allow for robust estimation of these parameters in future applications. This is especially important as accounting for moderating effects will be important in most scenarios, given that age trends in personality development and the variability of patterns across facets, traits, or domains are well established (Anusic & Schimmack, 2016; Bagaïni et al., 2023; Brandt et al., 2023; Mõttus et al., 2019; Seifert et al., 2022).

Forth and finally, the most complex scenario assessed in this thesis—a setup with additive and interactive moderating effects of age and domain on the model parameters—resulted in a plateau-effect where the change parameter could not be fully recovered even by increasing the very large dataset of 15,000 test-retest correlations further to 20,000. Whether or not adding even more data would have helped to finally estimating the parameter with high precision remains an open question.

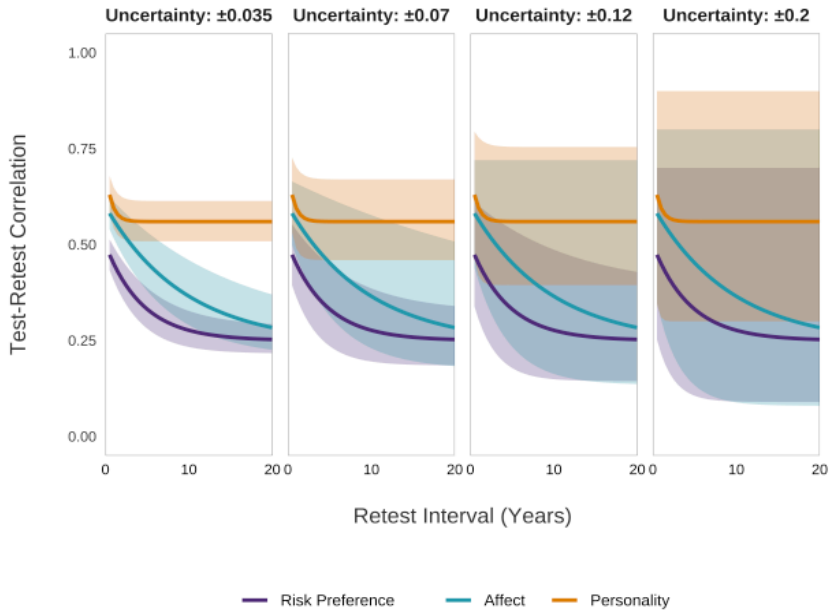
Understanding the Findings of the Current Simulation Studies

These findings provide an overview of the robustness and uncertainties associated with the MASC model across various parameter settings, sample sizes, and retest intervals. But what does a given level of uncertainty mean in practical terms, and how might it affect the interpretability of the estimated data? For the reliability parameter, interpretation is relatively straightforward, as it primarily represents the y-intercept of the MASC curve and is unaffected by time. Still, because reliability multiplies with the other parameters, its uncertainty can influence the entire curve trajectory. Further, the change and stability of change parameters also interact and are time-sensitive (see Figure 1), meaning that uncertainty in these parameters can accumulate over time.

Figure 7 is a theoretical illustration of the curve trajectories for affect, risk preference, and personality with their respective uncertainty bands—those represent the area where the curve could traject, given that the interacting parameters lie within a certain value-range, i.e. within a fixed accuracy threshold. The shown uncertainty levels correspond to those defined for the corridor of stability in this thesis as well as

Figure 7

Theoretical Illustration of Uncertainty Accumulation over Time



Note. The uncertainty bands represent four different accuracy thresholds (±0.035, ±0.07, ±0.12, ±0.2) for the parameters reliability, change and stability of change.

an additional, stricter scenario where all parameters have an uncertainty of ±0.035—a plausible setup, at least for the simulated intercept-only cases (see Appendix A, Table A3).

As shown, different parameter combinations lead to distinct patterns of uncertainty accumulation across constructs. The uncertainty band for risk preference starts narrow, then widens as the retest interval increases, eventually stabilizing. For affect, the uncertainty band also starts narrow but continues to widen consistently throughout the observed interval. In contrast, the personality band remains relatively parallel to its curve, indicating stable uncertainty across the retest interval. These patterns occur because constructs with high stability of change values decline slowly, remaining stable for a longer period—particularly when interacting with a large portion of changing variance, as in the affect construct. A slow decline leads to accumulating uncertainty, where deviations from the true parameter values are amplified over time.

Conversely, with lower stability of change and lower change (as in the personality construct), the decay function is steeper, and uncertainty accumulates less over time. In these cases, the uncertainty band remains relatively parallel to the true value curve, while it takes on a funnel shape when uncertainty accumulates, eventually stabilizing as the asymptote is approached.

This pattern has practical implications for deciding on an adequate level of accuracy. Constructs showing patterns similar to affect, where uncertainty accumulates over time, require a high level of accuracy. Figure 7 demonstrates that even a high precision level (here, ±0.07) results in overlap between constructs, making interpretation challenging. For risk preference, this same level of precision can reasonably estimate the parameters. On the other hand, what in this thesis is called medium precision (±0.12) introduces, taking into account the interacting quality of the parameters, a cumulative uncertainty that one could argue is too large, since it allows for curve trajectories that differ significantly. Clearly, at the lowest precision level (±0.2), the uncertainty bands across all constructs become so wide that meaningful interpretation is nearly impossible. This suggests that an uncertainty threshold should ideally not exceed ±0.07 to maintain clear distinctions between constructs. Note that these visualizations are based on intercept-only models; incorporating moderating effects would likely broaden the uncertainty bands, emphasizing the importance of setting appropriate accuracy thresholds.

Limitations of the Current Simulation Studies

Before discussing the implications of these findings, it is important to address several limitations of these simulation studies.

First, while I successfully modeled the effects of age and domain, one important predictor examined by Bagaïni et al. (2023) was sample structure. Simulating scenarios with varied panel intra- and interdependencies would be crucial, but this proved beyond the feasible scope of this thesis and could serve as a valuable starting point for future research.

Second, the parameter values used for dataset simulations were chosen to mimic three constructs that are likely important in behavioral research. However, the results were not entirely uniform, with the personality dataset showing greater uncertainty in the estimates of two out of the three parameters. While the reasons for this discrepancy could not be entirely established, one could argue that a fully factorial

design—which simulates all possible combinations of parameter values—would exhaustively explore patterns that emerge from unique parameter combinations. Morris et al. (2019) and Siepe et al. (2023) recommend using such designs whenever feasible. Research teams disposing of sufficient computational power could implement these fully factorial designs, leveraging the foundational work presented here.

Third, in this study, the level of noise and prior settings were kept constant. While this approach enhances comparability and allows for a clearer focus, future research could vary these characteristics to investigate their effects systematically. Testing the impact of more informative priors, in particular, could yield valuable insights for formulating recommendations on applying the MASC model.

Fourth, selecting appropriate metrics for evaluating the simulation experiments involved several theoretical considerations. Although these choices were informed, alternative metrics could also be relevant (see, e.g., Morris et al., 2019).

Finally, while simulation studies can reveal important insights into a model's behavior, they cannot fully anticipate all challenges that may arise with real-world data or capture its complexity in full. As noted by Boulesteix et al. (2020), this limitation is increasingly mitigated by advances in computational power, which make it possible to run simulations across a wider range of assumptions and parameter combinations. However, computational constraints remained a limiting factor in the present thesis. As a result, I adapted the sizes of the simulated datasets to meet the demands of scenarios with moderating effects, rather than rerunning simulations for intercept-only models including very large sample sizes—a step that might have been methodologically preferable.

Conclusion and Implications for Future Research

In conclusion, researchers who wish to use the MASC model for meta-analyses should ideally be equipped with very large datasets. For instance, Bagaïni et al. (2023) gathered over 70,000 test-retest correlations and made these publicly available, enabling the research community to build on their work. Open data can play a critical role in facilitating cumulative knowledge and allow for comprehensive model validation by different research teams (LeBel et al., 2017).

Estimating temporal patterns in psychological constructs remains a challenging endeavor. The nonlinearity and multiplicative nature of parameters, combined with the complexity of exponential decay functions, make these estimates delicate and prone to accumulating uncertainty. Prior knowledge can improve precision in estimates, but achieving robust priors requires a substantial data foundation (McElreath, 2016).

As larger, continuous datasets become available through mobile devices and online behavior tracking, there is a growing opportunity to apply dynamic, time-sensitive modeling techniques. Jebb et al. (2015) advocate for increasingly using time-series analyses for the interpretation of temporal patterns, also in the field of psychology. While this approach primarily holds promise on the level of empirical studies it might ultimately also shed light on the overarching constructs relevant for meta-analyses.

An ongoing challenge in studying personality development is to refine analysis techniques to account for inter-individual variability. Mõttus et al. (2019) elaborate that as individuals age, their personal trajectories might diverge due to differences in life experiences, social environments, and genetic predispositions. These divergences can challenge models that assume uniform trajectories across a population and emphasize the need to investigate the role moderating variables on the patterns and rates of stability and change in personality traits over time.

Simulation studies are an invaluable tool for understanding both the strengths and limitations of models like the MASC model. By systematically testing models across varying sample sizes, parameter ranges, and conditions, we gain insight into the specific contexts in which these models perform well and where they may fall short (Siepe et al., 2023). This approach highlights practical issues, such as the need for large datasets and robust priors, while also identifying areas where methodological refinements could further enhance model reliability. Ultimately, studies like this one advance our capacity to interpret temporal patterns in psychological constructs, building on a foundation for ever more precise and adaptable applications for the research of the future.

References

Anusic, I., & Schimmack, U. (2016). Stability and change of personality traits, self-esteem, and well-being: Introducing the meta-analytic stability and change model of retest correlations. *Journal of Personality and Social Psychology*, *110*(5), 766–781. https://doi.org/10.1037/pspp0000066

Bagaïni, A., Liu, Y., Kapoor, M., Son, G., Bürkner, P.-C., Tisdall, L., & Mata, R. (2023). *Meta-Analyses of the Temporal Stability and Convergent Validity of Risk Preference Measures*. https://doi.org/10.31234/osf.io/d7nuj

Bleidorn, W., Schwaba, T., Zheng, A., Hopwood, C. J., Sosa, S. S., Roberts, B. W., & Briley, D. A. (2022). *Personality Stability and Change: A Meta-Analysis of Longitudinal Studies*.

Boulesteix, A.-L., Groenwold, R. H., Abrahamowicz, M., Binder, H., Briel, M., Hornung, R., Morris, T. P., Rahnenführer, J., & Sauerbrei, W. (2020). Introduction to statistical simulations in health research. *BMJ Open*, *10*(12), e039921. https://doi.org/10.1136/bmjopen-2020-039921

Brandt, N. D., Drewelies, J., Willis, S. L., Schaie, K. W., Ram, N., Gerstorf, D., & Wagner, J. (2023). Beyond Big Five trait domains: Stability and change in personality facets across midlife and old age. *Journal of Personality*, *91*(5), 1171–1188. https://doi.org/10.1111/jopy.12791

Caspi, A., Roberts, B. W., & Shiner, R. L. (2005). Personality Development: Stability and Change. *Annual Review of Psychology*, *56*(1), 453–484. https://doi.org/10.1146/annurev.psych.55.090902.141913

Costa, P. T., & McCrae, R. R. (2018). *Personality Across the Life Span*.

Gurevitch, J., Koricheva, J., Nakagawa, S., & Stewart, G. (2018). Meta-analysis and the science of research synthesis. *Nature*, *555*(7695), 175–182. https://doi.org/10.1038/nature25753

Heathcote, A., Brown, S. D., & Wagenmakers, E.-J. (2015). An Introduction to Good Practices in Cognitive Modeling. In B. U. Forstmann & E.-J. Wagenmakers (Eds.), *An Introduction to Model-Based Cognitive Neuroscience* (pp. 25–48). Springer New York. https://doi.org/10.1007/978-1-4939-2236-9_2

Jebb, A. T., Tay, L., Wang, W., & Huang, Q. (2015). Time series analysis for psychological research: Examining and forecasting change. *Frontiers in Psychology*, *6*. https://doi.org/10.3389/fpsyg.2015.00727

Kruschke, J. K. (2010). What to believe: Bayesian methods for data analysis. *Trends in Cognitive Sciences*, *14*(7), 293–300. https://doi.org/10.1016/j.tics.2010.05.001

Kruschke, J. K., & Liddell, T. M. (2018). Bayesian data analysis for newcomers. *Psychonomic Bulletin & Review*, *25*(1), 155–177. https://doi.org/10.3758/s13423-017-1272-1

LeBel, E. P., Campbell, L., & Loving, T. J. (2017). Benefits of open and high-powered research outweigh costs. *Journal of Personality and Social Psychology*, *113*(2), 230–243. https://doi.org/10.1037/pspi0000049

Liu, Y., Bagaïni, A., Son, G., Kapoor, M., & Mata, R. (2023). Life-Course Trajectories of Risk-Taking Propensity: A Coordinated Analysis of Longitudinal Studies. *The Journals of Gerontology: Series B*, *78*(3), 445–455. https://doi.org/10.1093/geronb/gbac175

Mata, R., Frey, R., Richter, D., Schupp, J., & Hertwig, R. (2018). Risk Preference: A View from Psychology. *Journal of Economic Perspectives*, *32*(2), 155–172. https://doi.org/10.1257/jep.32.2.155

Mata, R., Josef, A. K., & Hertwig, R. (2016). Propensity for Risk Taking Across the Life Span and Around the Globe. *Psychological Science*, *27*(2), 231–243. https://doi.org/10.1177/0956797615617811

McElreath, R. (2016). *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC Press Taylor & Francis.

Moffitt, T. E., Arseneault, L., Belsky, D., Dickson, N., Hancox, R. J., Harrington, H., Houts, R., Poulton, R., Roberts, B. W., Ross, S., Sears, M. R., Thomson, W. M., & Caspi, A. (2011). A gradient of childhood self-control predicts health, wealth, and public safety. *Proceedings of the National Academy of Sciences*, *108*(7), 2693–2698. https://doi.org/10.1073/pnas.1010076108

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. https://doi.org/10.1002/sim.8086

Mõttus, R., Briley, D. A., Zheng, A., Mann, F. D., Engelhardt, L. E., Tackett, J. L., Harden, K. P., & Tucker-Drob, E. M. (2019). Kids becoming less alike: A behavioral genetic analysis of developmental increases in personality variance from childhood to adolescence. *Journal of Personality and Social Psychology*, *117*(3), 635–658. https://doi.org/10.1037/pspp0000194

R Core Team. (2021). *A language and environment for statistical computing* [Computer software]. R Foundation for Statistical Computing. https://www.R-project.org/

Roberts, B. W., & DelVecchio, W. F. (n.d.). *The Rank-Order Consistency of Personality Traits From Childhood to Old Age: A Quantitative Review of Longitudinal Studies*.

Roberts, B. W., Walton, K. E., & Viechtbauer, W. (2006). Patterns of mean-level change in personality traits across the life course: A meta-analysis of longitudinal studies. *Psychological Bulletin*, *132*(1), 1–25. https://doi.org/10.1037/0033-2909.132.1.1

Schad, D. J., Betancourt, M., & Vasishth, S. (2021). Toward a principled Bayesian workflow in cognitive science. *Psychological Methods*, *26*(1), 103–126. https://doi.org/10.1037/met0000275

Schönbrodt, F. D., & Perugini, M. (2013). At what sample size do correlations stabilize? *Journal of Research in Personality*, *47*(5), 609–612. https://doi.org/10.1016/j.jrp.2013.05.009

Seifert, I. S., Rohrer, J. M., Egloff, B., & Schmukle, S. C. (2022). *The Development of the Rank-Order Stability of the Big Five Across the Life Span*.

Siepe, B. S., Bartoš, F., Morris, T. P., Boulesteix, A.-L., Heck, D. W., & Pawel, S. (2023). *Simulation Studies for Methodological Research in Psychology: A Standardized Template for Planning, Preregistration, and Reporting* [Preprint]. PsyArXiv. https://doi.org/10.31234/osf.io/ufgy6

Appendix A: Tables and Figures Referenced in Simulation Study 1

Table A 1

Risk Preference Dataset: Percentage of Estimates within the Corridor of Stability by Number of Correlations

Number of Correlations	Reliability	Change	Stab.Change
25	48%	58%	18%
50	84%	62%	52%
250	98%	92%	86%
500	100%	100%	94%
1000	100%	100%	100%
2000	100%	100%	100%
10000	100%	100%	100%

Note. Calculations are based on a corridor half-width of w = ± 0.07.

Table A 2

Risk Preference Dataset: Accuracy Thresholds

	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**1000**	**250**	**250**	**500**	**250**	**250**	**250**	**250**	**50**
Reliability	250	50	25	250	50	25	50	25	25
Change	500	250	25	250	50	25	250	25	25
Stab.Change	1000	250	250	500	250	250	250	250	50

Note. Accuracy levels are calculated for different widths (w) of the corridor and diffent levels of confidence.

Values represent the required sample sizes for meeting the defined thresholds.

Figure A 3

Affect Dataset: Corridor of Stability w = ±0.07

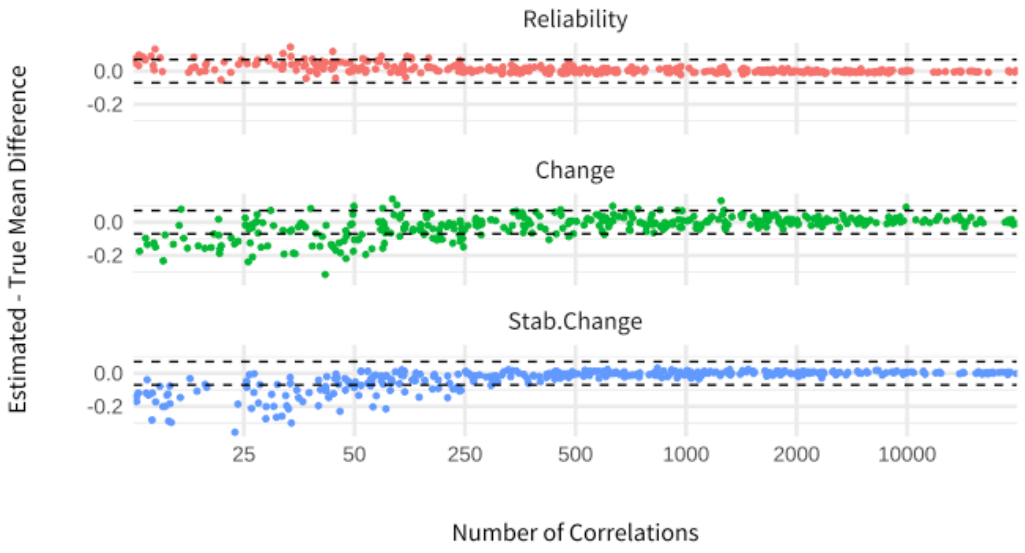


Table A 4

Affect Dataset: Percentage of Estimates within the Corridor of Stability by Number of Correlations

Number of Correlations	Reliability	Change	Stab.Change
25	78%	20%	14%
50	82%	56%	54%
250	100%	80%	96%
500	100%	86%	100%
1000	100%	94%	100%
2000	100%	98%	100%
10000	100%	100%	100%

Note. Calculations are based on a corridor half-width of w = ± 0.07.

Table A 5

Affect Dataset: Accuracy Thresholds

	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**2000**	**250**	**250**	**1000**	**250**	**50**	**500**	**50**	**50**
Reliability	250	25	25	250	25	25	50	25	25
Change	2000	250	50	1000	250	50	500	50	25
Stab.Change	250	250	250	250	250	50	250	50	50

Note. Accuracy levels are calculated for different widths (w) of the corridor and diffent levels of confidence.

Values represent the required sample sizes for meeting the defined thresholds.

Figure A 6

Personality Dataset: Corridor of Stability w= ±0.07

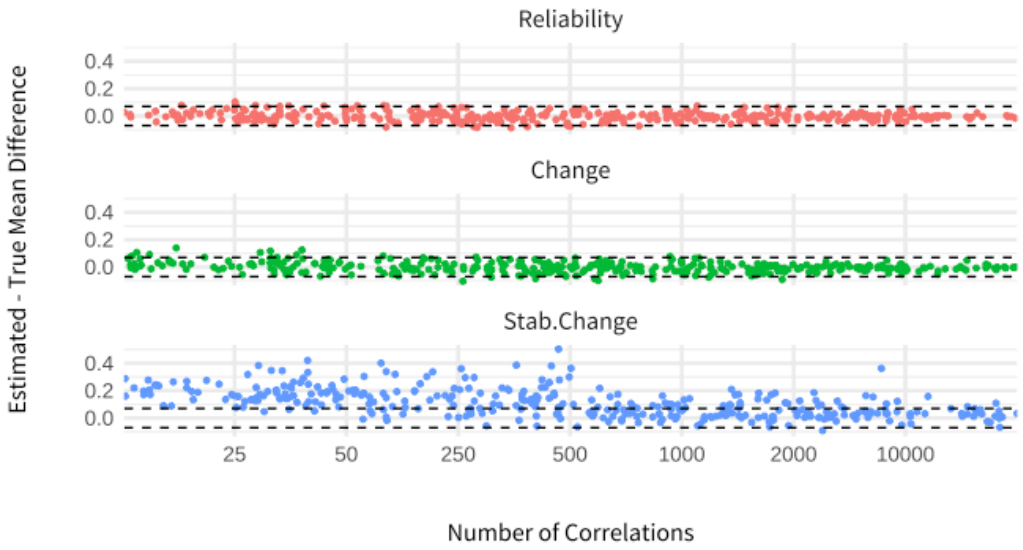


Table A 7

Personality Dataset: Percentage of Estimates within the Corridor of Stability by Number of Correlations

Number of Correlations	Reliability	Change	Stab.Change
25	96%	88%	2%
50	86%	90%	8%
250	96%	96%	26%
500	92%	88%	46%
1000	96%	96%	60%
2000	100%	100%	56%
10000	100%	100%	88%

Note. Calculations are based on a corridor half-width of w = ± 0.07.

Table A 8

Personality Dataset: Accuracy Thresholds

	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**>10,000**	**10000**	**1000**	**>10,000**	**10000**	**1000**	**10000**	**1000**	**1000**
Reliability	25	25	25	25	25	25	25	25	25
Change	250	25	25	250	25	25	25	25	25
Stab.Change	>10,000	10000	1000	>10,000	10000	1000	10000	1000	1000

Note. Accuracy levels are calculated for different widths (w) of the corridor and diffent levels of confidence.

Values represent the required sample sizes for meeting the defined thresholds.

Figure A 9

Risk Preference Dataset: Corridor of Stability w= ±0.07 for Different Maximum Retest Intervals

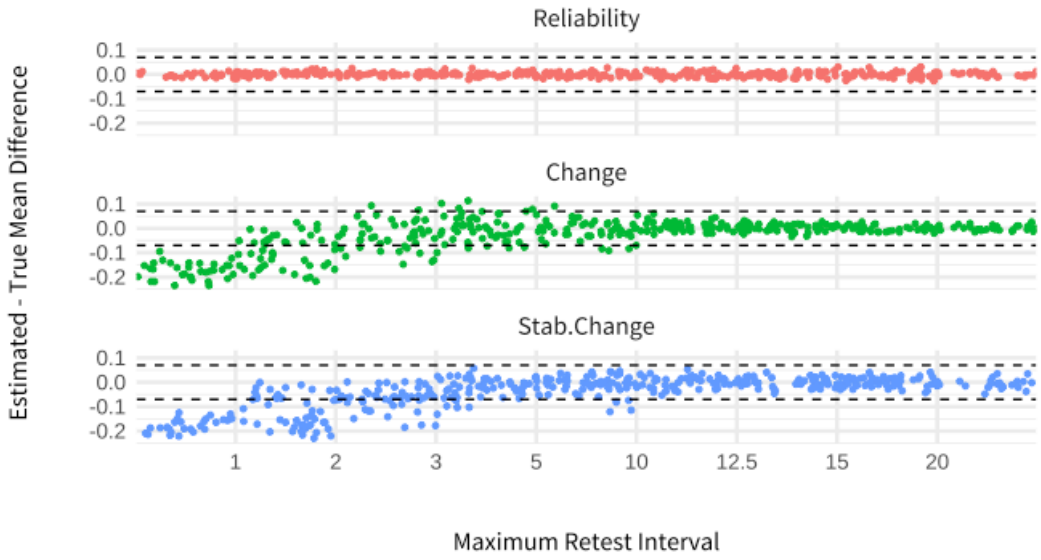


Table A 10

Risk Preference Dataset: Accuracy Thresholds for Maximum Retest Intervals

	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**10**	**3**	**2**	**10**	**3**	**2**	**3**	**2**	**2**
Reliability	1	1	1	1	1	1	1	1	1
Change	10	3	2	10	2	2	3	2	2
Stab.Change	10	3	2	10	3	2	3	2	2

Note. Maximum retest intervals are calculated for different corridor widths (w) and confidence levels. Values represent the maximum intervals within which the thresholds defined by the corridor of stability are met. Parameters include Reliability, Change, and Stability of Change (Stab.Change).

Figure A 11

Affect Dataset: Corridor of Stability w= ±0.07 for Different Maximum Retest Intervals

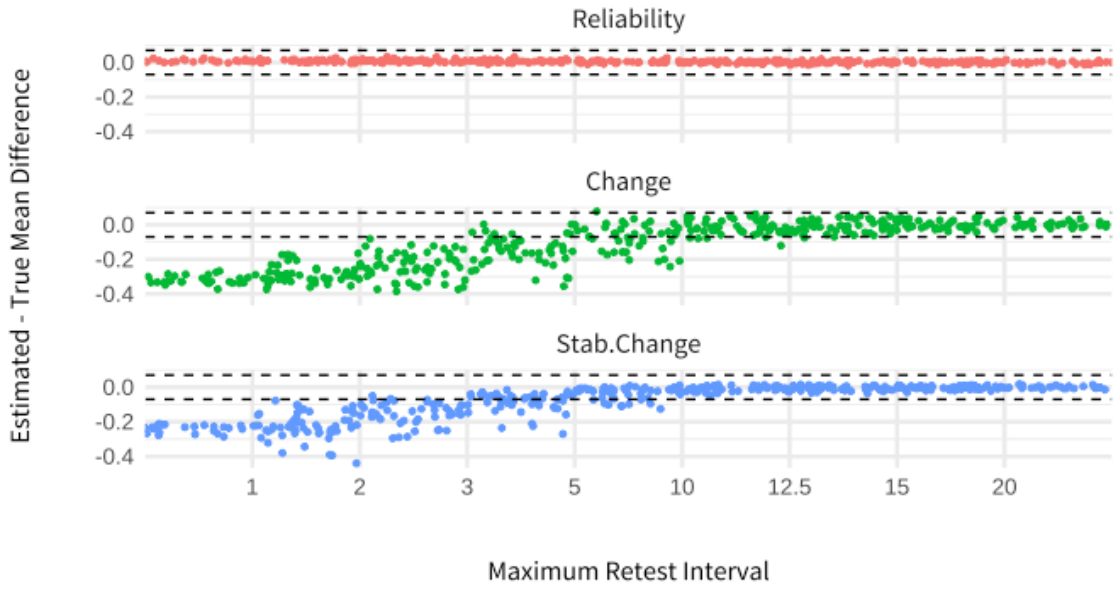


Table A 12

Affect Dataset: Accuracy Thresholds for Maximum Retest Intervals

	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**12.5**	**10**	**10**	**12.5**	**10**	**10**	**10**	**10**	**5**
Reliability	1.0	1	1	1.0	1	1	1	1	1
Change	12.5	10	10	12.5	10	10	10	10	5
Stab.Change	10.0	5	5	10.0	5	5	10	5	3

Note. Maximum retest intervals are calculated for different corridor widths (w) and confidence levels. Values represent the maximum intervals within which the thresholds defined by the corridor of stability are met. Parameters include Reliability, Change, and Stability of Change (Stab.Change).

Figure A 13

Personality Dataset: Corridor of Stability w= ±0.07 for Different Maximum Retest Intervals

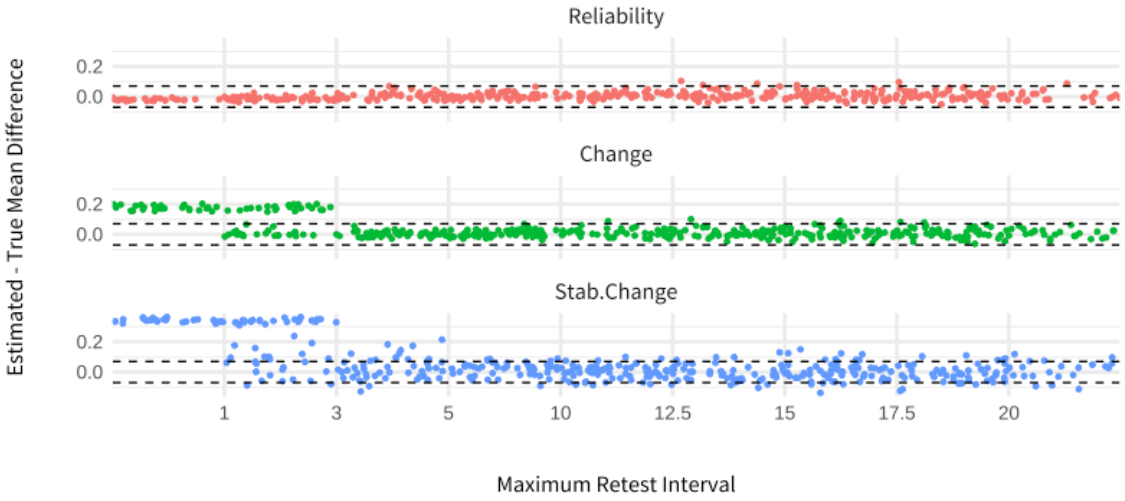


Table A 14

Personality Dataset: Accuracy Thresholds for Maximum Retest Intervals

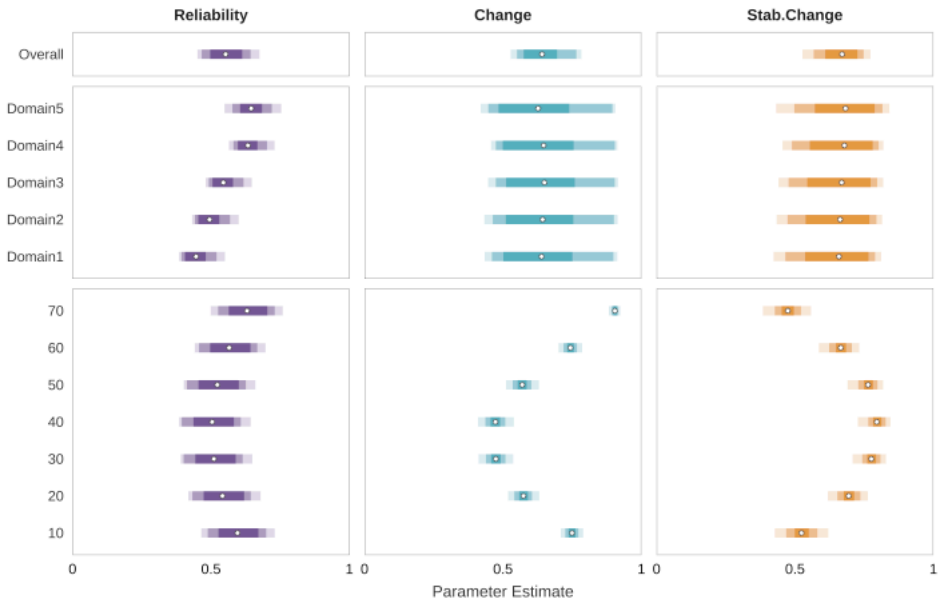
	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**>20**	**5**	**3**	**10**	**5**	**3**	**5**	**5**	**3**
Reliability	1	1	1	1	1	1	1	1	1
Change	3	3	1	3	3	1	3	3	1
Stab.Change	>20	5	3	10	5	3	5	5	3

Note. Maximum retest intervals are calculated for different corridor widths (w) and confidence levels. Values represent the maximum intervals within which the thresholds defined by the corridor of stability are met. Parameters include Reliability, Change, and Stability of Change (Stab.Change).

Appendix B: Tables and Figures Referenced in Simulation Study 2

Figure B 1

Simulated Additive Effects Dataset: Parameter Values after Fitting



Note. The visualization uses parameter values averaged over all simulated datasets.

Figure B2

Additive Effects Dataset: Corridor of Stability w=±0.07

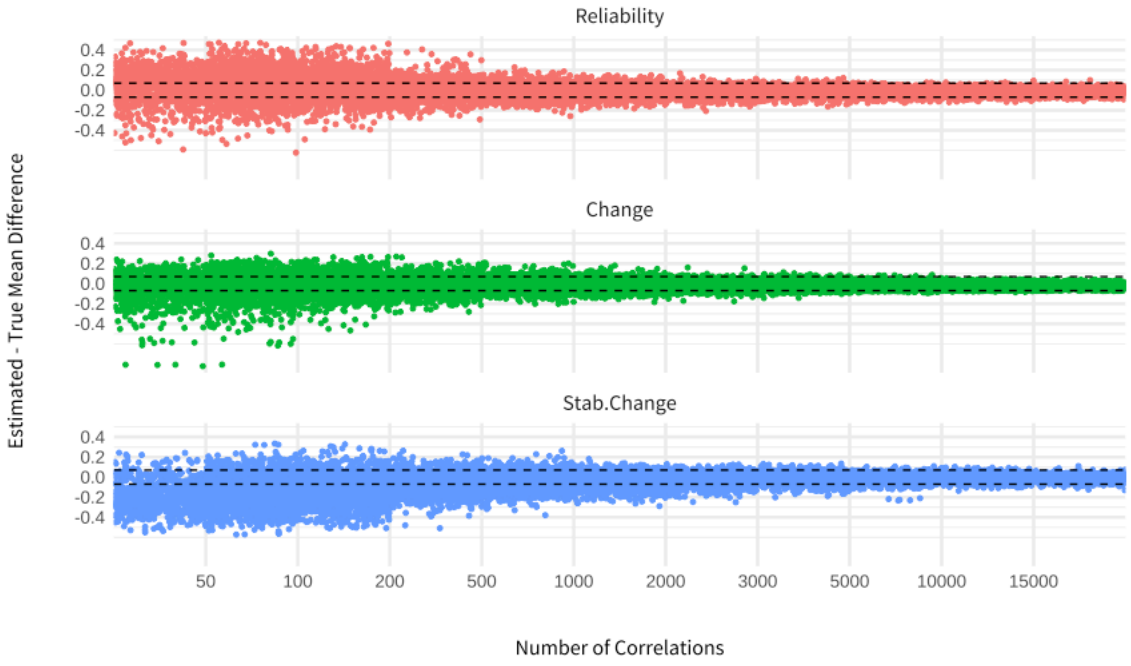


Table B 3

Additive Effects Dataset: Percentage of Estimates within the Corridor of Stability by Number of Correlations

Number of Correlations	Reliability	Change	Stab.Change
50	32%	48%	14%
100	38%	50%	21%
200	58%	64%	38%
500	75%	75%	56%
1000	85%	82%	69%
2000	90%	87%	79%
3000	92%	92%	84%
5000	94%	94%	93%
10000	95%	98%	96%
15000	96%	99%	96%

Note. Calculations are based on a corridor half-width of w = ± 0.07.

Table B 4

Additive Effects Dataset: Accuracy Thresholds

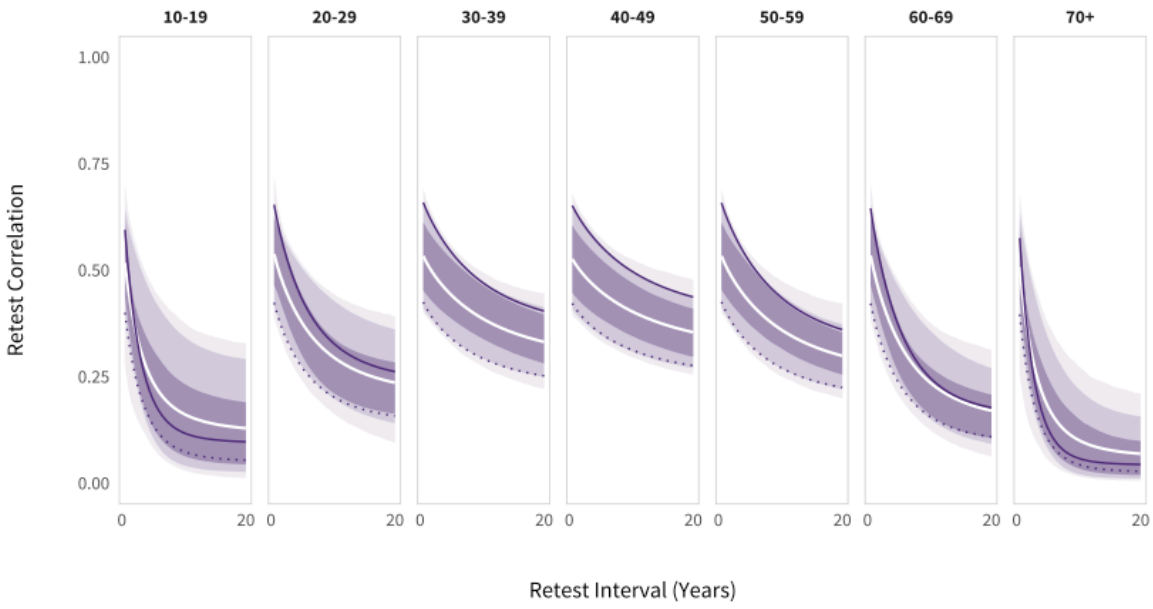
	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**10000**	**3000**	**500**	**5000**	**2000**	**500**	**3000**	**500**	**200**
Reliability	10000	1000	200	3000	500	200	1000	200	100
Change	10000	1000	200	3000	500	100	1000	200	50
Stab.Change	10000	3000	500	5000	2000	500	3000	500	200

Note. Accuracy levels are calculated for different widths (w) of the corridor and diffent levels of confidence.

Values represent the required sample sizes for meeting the defined thresholds.

Figure B 5

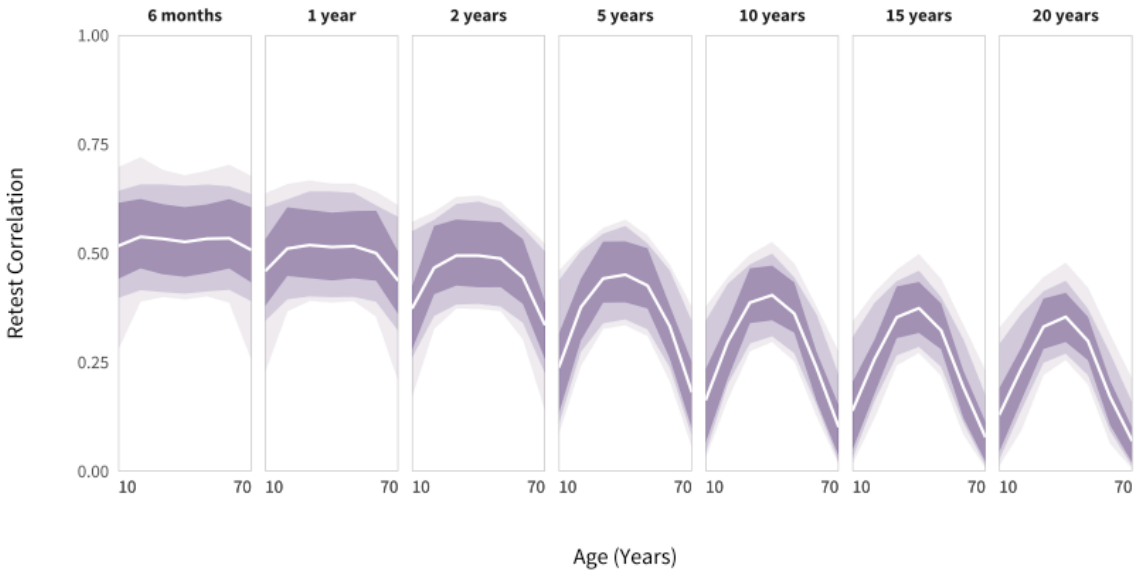
Interactive Effects Dataset: Retest Trajectories by Age Groups as a Function of the Retest Interval



Note. Data is averaged across domains and simulated datasets. The white line represents the average trajectory over all domains while the dotted and the solid line show *Domain 1* and *Domain 5* respectively. Shaded areas indicate 95%, 80% and 50% HDI.

Figure B 6

Interactive Effects Dataset: Retest Trajectories by Retest Intervals as a Function of Age



Note. Data is averaged across domains and simulated datasets. Shaded areas indicate 95%, 80% and 50% HDI.

Table B 7

Interactive Effects Dataset: Percentage of Estimates within the Corridor of Stability by Number of Correlations

Number of Correlations	Reliability	Change	Stab.Change
50	34%	35%	19%
100	44%	39%	44%
250	75%	39%	71%
500	85%	51%	80%
1000	93%	60%	89%
2000	98%	72%	96%
3000	99%	73%	98%
5000	100%	79%	99%
10000	100%	77%	100%
15000	100%	81%	100%
20000	100%	79%	100%

Note. Calculations are based on a corridor half-width of w = ± 0.07.

Table B 8

Interactive Effects Dataset: Accuracy Thresholds

	95%			90%			80%		
Parameter	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2	w = 0.07	w = 0.12	w = 0.2
Whole model	**>20000**	**5000**	**1000**	**>20000**	**2000**	**500**	**15000**	**1000**	**250**
Reliability	2000	500	250	1000	500	250	500	250	100
Change	>20000	5000	1000	>20000	2000	500	15000	1000	100
Stab.Change	2000	1000	250	2000	500	250	1000	250	250

Note. Accuracy levels are calculated for different widths (w) of the corridor and diffent levels of confidence.

Values represent the required sample sizes for meeting the defined thresholds.

Appendix C: Code for Data Simulation and Evaluation

Data Simulation Code C1: Intercept-Only Models

```
# DESCRIPTION ------------------------------------------------------------------

# In this script we simulate intercept-only datasets for the constructs risk preference,
# affect and personality for different sample sizes.
#
# Author(s): Alexandra Bagaini and Sabine Gisin
# Alexandra Bagaini provided the original simulation script for balanced simulation scenarios,
# Sabine Gisin adapted the original script to simulate unbalanced datasets and to store
# dataset informations.


# PACKAGES ---------------------------------------------------------------------
library(tidyverse)
library(tidyr)
library(brms)
library(tidybayes)
library(boot) # logit function
library(data.table)
library(gt)

# FUNCTIONS --------------------------------------------------------------------

masc <- function(rel, change, stabch, retest_interval) {

 retest <- rel * (change * (stabch^retest_interval - 1) + 1)

 return(retest)

}

inv_logit <- function(x) {inv.logit(x)}

# CREATING THE DATA SET --------------------------------------------------------

# setting up the true values of the parameters (Risk Preference)
rel <- 0.5
change <- 0.5
stabch <- 0.8
sigma =  0.1

# # setting up the true values of the parameters (Affect)
# rel <- 0.6
# change <- 0.6
# stabch <- 0.9
# sigma =  0.1

#setting up the true values of the parameters (Personality)
# rel <- 0.7
# change <- 0.2
# stabch <- 0.25
# sigma =  0.1

# data struct
retest_interval = seq(.5,20, by = .5) # 6 month intervals
panel = as.character(1:1000) # (this can vary)

# creating a dataset that has all of the combinations of the variables using the crossings() function
data_struct <- crossing(retest_interval = retest_interval,
                        panel = panel)
```

```
m1_data <- data_struct %>%
 mutate(
  # adding "true" param value to the dataset
  rel = rel,
  change = change,
  stabch = stabch,
  sigma = sigma,
  # calc retest
  retest = masc(rel = rel, #  rel ~ 1
          change = change, #  change ~ 1
          stabch = stabch, #  stabch ~ 1
          retest_interval = retest_interval)  + rnorm(n(), 0, sigma)) %>%
 mutate(retest = case_when(retest > 1 ~ 1,
                 retest < 0 ~ 0,
                 TRUE ~retest)) # retest correlations bounded between 0 & 1


true_masc <-data_struct %>%
 filter(panel == "1") %>%
 mutate(
  # adding "true" param value to the dataset
  rel = rel,
  change = change,
  stabch = stabch,
  sigma = sigma,
  # calc retest
  retest = masc(rel = rel, #  rel ~ 1
          change = change, #  change ~ 1
          stabch = stabch, #  stabch ~ 1
          retest_interval = retest_interval))

# # Weighted sampling
# # Separate the data into two subsets based on retest_interval
data_smaller_6 <- m1_data %>% filter(retest_interval <= 6)
data_larger_6 <- m1_data %>% filter(retest_interval > 6)
#
# # Define the number of observations to sample from each subset
num_smaller_6 <- round(2/3 * obs_num)
num_larger_6 <- obs_num - num_smaller_6
# # Perform the sampling
sample_smaller_6 <- data_smaller_6[sample(nrow(data_smaller_6), num_smaller_6, replace = TRUE), ]
sample_larger_6 <- data_larger_6[sample(nrow(data_larger_6), num_larger_6, replace = TRUE), ]

# # Combine the sampled data
sub_m1_data <- bind_rows(sample_smaller_6, sample_larger_6)


# DATA SIMULATION ----------------------------------------------------------

full_eval_zscore <- NULL

data_list <- NULL

true_vals <- tibble(
 rel = rel,
 change = change,
 stabch = stabch) %>%
 pivot_longer(1:3, values_to = "true_val", names_to = "nlpar")


for (n_sim in c(1:50)) {

 for (obs_num in c(25, 50, 250, 500, 1000, 2000, 3000, 5000, 10000)) {

  # Weighted sampling
  # Separate the data into two subsets based on retest_interval
  data_smaller_6 <- m1_data %>% filter(retest_interval <= 6)
  data_larger_6 <- m1_data %>% filter(retest_interval > 6)
```

```
# Define the number of observations to sample from each subset
num_smaller_6 <- round(2/3 * obs_num)
num_larger_6 <- obs_num - num_smaller_6

# Perform the sampling
sample_smaller_6 <- data_smaller_6[sample(nrow(data_smaller_6), num_smaller_6, replace = TRUE), ]
sample_larger_6 <- data_larger_6[sample(nrow(data_larger_6), num_larger_6, replace = TRUE), ]

# Combine the sampled data
sub_m1_data <- bind_rows(sample_smaller_6, sample_larger_6)

family <- brmsfamily(
 family = "student",
 link = "identity"
)


# setting up priors
# weakly informative priors
priors <-
 prior(normal(0, 1), nlpar="logitrel", class = "b") +
 prior(normal(0, 1), nlpar="logitchange", class = "b") +
 prior(normal(0, 1), nlpar="logitstabch", class = "b")


# setting up model
formula <- bf(
 retest ~ rel * (change * ((stabch^retest_interval) - 1) + 1),
 nlf(rel ~ inv_logit(logitrel)),
 nlf(change ~ inv_logit(logitchange)),
 nlf(stabch ~ inv_logit(logitstabch)),
 logitrel ~ 1,
 logitchange ~ 1,
 logitstabch ~ 1,
 nl = TRUE
)



# fit model (adjust arguments where needed; e.g., "control", "cores", "chains")
m1_fit_masc <- brm(
 formula = formula,
 prior = priors,
 family = family,
 data = sub_m1_data,
 cores = 2,
 chains = 2,
 iter = 3000,
 warmup = 1000,
 backend = "cmdstanr",
 control = list(max_treedepth = 10, adapt_delta = 0.95),
 init = "0"
)

# extracting param values
epred_rel <-  as.numeric(add_epred_draws(m1_fit_masc, newdata = tibble(retest_interval = 0, panel = "1"), nlpar = "rel")$.epred)
epred_change <-  as.numeric(add_epred_draws(m1_fit_masc, newdata = tibble(retest_interval = 0, panel = "1"), nlpar =
"change")$.epred)
epred_stabch <-  as.numeric(add_epred_draws(m1_fit_masc, newdata = tibble(retest_interval = 0, panel = "1"), nlpar =
"stabch")$.epred)

est_vals <- tibble(rel = epred_rel,
                   change = epred_change,
                   stabch = epred_stabch) %>%
 pivot_longer(rel:stabch,names_to = "nlpar", values_to = "val") %>%
 group_by(nlpar) %>%
 summarise(est_mean = mean(val),est_sd = sd(val))
```

```
  eval_zscore <- est_vals %>% left_join(true_vals, by = "nlpar")


  # compare estimated vs. true param val ( "posterior z-score")
  eval_zscore <- eval_zscore %>%
   mutate(post_zscore  = (est_mean-true_val)/est_sd,
        est_true_mean_diff = est_mean - true_val)


  # predicted MASC curve
  predict_retest <-  add_epred_draws(m1_fit_masc, newdata = data_struct %>% filter(panel == "1"))  %>%
   mean_qi() %>%
   rename(retest_predict = .epred)


  # storing model eval. metrics & data set info
  full_eval_zscore <- eval_zscore %>%
   mutate(     n_sim = n_sim,
           n_cor = obs_num) %>%
   bind_rows(full_eval_zscore)


  # saving data
  data_list <-  tibble(n_sim = n_sim,
                n_cor = obs_num,
                model_data = list(sub_m1_data),
                masc_pred = list(predict_retest)) %>%
   bind_rows(data_list)


  }
 }
}
```

Data Simulation Code C2: Maximum Retest Intervals

```
# DESCRIPTION ------------------------------------------------------------------

# In this script we simulate intercept-only datasets for the constructs risk preference,
# affect and personality for different maximum retest intervals.
#
# Author(s): Alexandra Bagaini and Sabine Gisin
# Alexandra Bagaini provided the original simulation script for balanced simulation scenarios,
# Sabine Gisin adapted the original script to simulate scenarios for different maximum retest intervals
# and to store dataset informations


# PACKAGES ---------------------------------------------------------------------
library(tidyverse)
library(tidyr)
library(brms)
library(tidybayes)
library(boot) # logit function
library(data.table)
library(gt)

# FUNCTIONS --------------------------------------------------------------------

masc <- function(rel, change, stabch, retest_interval) {

 retest <- rel * (change * (stabch^retest_interval - 1) + 1)

 return(retest)

}

inv_logit <- function(x) {inv.logit(x)}


# CREATING THE DATA SET --------------------------------------------------------
# setting up the true values of the parameters (Risk Preference)
rel <- 0.5
change <- 0.5
stabch <- 0.8
sigma =  0.1

# setting up the true values of the parameters (Affect)
# rel <- 0.6
# change <- 0.6
# stabch <- 0.9
# sigma =  0.1

# setting up the true values of the parameters (Personality)
# rel <- 0.7
# change <- 0.2
# stabch <- 0.25
# sigma =  0.1

# data struct
retest_interval = seq(.5,20, by = .5) # 6 month intervals
panel = as.character(1:5000) # (this can vary)


# DATA SIMULATION --------------------------------------------------------------

full_eval_zscore <- NULL

data_list <- NULL

true_vals <- tibble(
```

```
 rel = rel,
 change = change,
 stabch = stabch) %>%
 pivot_longer(1:3, values_to = "true_val", names_to = "nlpar")

obs_num <- 5000

for (n_sim in c(1:50)) {

 for (max_rt in c(1, 3, 5, 10, 12.5, 15, 17.5, 20)) {

  sub_m1_data <- m1_data %>% filter(retest_interval <= max_rt) %>%
   sample_n(obs_num, replace = TRUE)

  family <- brmsfamily(
   family = "student",
   link = "identity"
  )

  # setting up priors
  # weakly informative priors
  priors <-
   prior(normal(0, 1), nlpar="logitrel", class = "b") +
   prior(normal(0, 1), nlpar="logitchange", class = "b") +
   prior(normal(0, 1), nlpar="logitstabch", class = "b")

  # setting up model
  formula <- bf(
   retest ~ rel * (change * ((stabch^retest_interval) - 1) + 1),
   nlf(rel ~ inv_logit(logitrel)),
   nlf(change ~ inv_logit(logitchange)),
   nlf(stabch ~ inv_logit(logitstabch)),
   logitrel ~ 1,
   logitchange ~ 1,
   logitstabch ~ 1,
   nl = TRUE
  )

  # fit model (adjust arguments where needed; e.g., "control", "cores", "chains")
  m1_fit_masc <- brm(
   formula = formula,
   prior = priors,
   family = family,
   data = sub_m1_data,
   cores = 2,
   chains = 2,
   iter = 3000,
   warmup = 1000,
   backend = "cmdstanr",
   control = list(max_treedepth = 10, adapt_delta = 0.95, step_size = 0.02),
   init = "0",
   seed = 75672
  )

  ### 4. Extracting & Storing Model Diagnostics

  #Extract r_hats
  rhat_values <- rhat(m1_fit_masc, pars = c("b_logitrel_Intercept", "b_logitchange_Intercept", "b_logitstabch_Intercept"))   #pars =
NULL would return ALL rhats

  #extract divergent transitions
  np <- nuts_params(m1_fit_masc)
  str(np)
  # extract the number of divergence transitions
  divergent_transitions <- sum(subset(np, Parameter == "divergent__")$Value)

  #extract neff-ratio
  neff_ratio <- neff_ratio(m1_fit_masc, pars = c("b_logitrel_Intercept", "b_logitchange_Intercept", "b_logitstabch_Intercept"))
```

```
  #create tibble to combine model diagnostics
  model_diagnostics <- tibble(nlpar = names(rhat_values),
                      R_hat = rhat_values,
                      neff_ratio = neff_ratio[names(rhat_values)],
                      divergent_transitions = divergent_transitions) %>%# Align by names to ensure correct matching)
    mutate(nlpar = case_when(
      nlpar == "b_logitrel_Intercept" ~ "rel",
      nlpar == "b_logitchange_Intercept" ~ "change",
      nlpar == "b_logitstabch_Intercept" ~ "stabch"))


  # extracting param values
  epred_rel <-  as.numeric(add_epred_draws(m1_fit_masc, newdata = tibble(retest_interval = 0, panel = "1"), nlpar = "rel")$.epred)
  epred_change <-  as.numeric(add_epred_draws(m1_fit_masc, newdata = tibble(retest_interval = 0, panel = "1"), nlpar =
"change")$.epred)
  epred_stabch <-  as.numeric(add_epred_draws(m1_fit_masc, newdata = tibble(retest_interval = 0, panel = "1"), nlpar =
"stabch")$.epred)

  est_vals <- tibble(rel = epred_rel,
                change = epred_change,
                stabch = epred_stabch) %>%
    pivot_longer(rel:stabch,names_to = "nlpar", values_to = "val") %>%
    group_by(nlpar) %>%
    summarise(est_mean = mean(val),est_sd = sd(val))



  eval_zscore <- est_vals %>%
    left_join(true_vals, by = "nlpar") %>%
    left_join(model_diagnostics, by = "nlpar")



  # compare estimated vs. true param val ( "posterior z-score")
  eval_zscore <- eval_zscore %>%
    mutate(post_zscore  = (est_mean-true_val)/est_sd,
         est_true_mean_diff = est_mean - true_val)


  # predicted MASC curve
  predict_retest <-  add_epred_draws(m1_fit_masc, newdata = data_struct %>% filter(panel == "1"))  %>%
    mean_qi() %>%
    rename(retest_predict = .epred)


  # storing model eval. metrics & data set info
  full_eval_zscore <- eval_zscore %>%
    mutate(     n_sim = n_sim,
            max_rt = max_rt,
            n_cor = obs_num) %>%
    bind_rows(full_eval_zscore)


  # saving data
  data_list <-  tibble(n_sim = n_sim,
                n_cor = obs_num,
                max_rt = max_rt,
                model_data = list(sub_m1_data),
                masc_pred = list(predict_retest)) %>%
    bind_rows(data_list)


  }

 }
```

Data Simulation Code C3: Additive Effects of Age and Domain

```
# DESCRIPTION ----------------------------------------------------------------
# In this script we simulate datasets WITH ADDITIVE EFFECTS for the constructs risk preference,
# affect and personality.
# Author(s): Alexandra Bagaini and Sabine Gisin
# Alexandra Bagaini provided the original simulation script for balanced balenced intercept-only simulation scenarios,
# Sabine Gisin adapted the original script to simulate additive moderating effects.

# PACKAGES ----------------------------------------------------------------
library(tidyverse)
library(tidyr)
library(brms)
library(tidybayes)
library(boot) # logit function
library(data.table)
library(gt)

# FUNCTIONS ----------------------------------------------------------------

masc <- function(rel, change, stabch, retest_interval) {

 retest <- rel * (change * (stabch^retest_interval - 1) + 1)

 return(retest)

}

inv_logit <- function(x) {inv.logit(x)}


# CREATING THE DATA SET ----------------------------------------------------
# setting up the true values of the parameters (Risk Preference)
rel <- 0.5
change <- 0.5
stabch <- 0.8
sigma =  0.1

# setting up the true values of the parameters (Affect)
# rel <- 0.6
# change <- 0.6
# stabch <- 0.9
# sigma =  0.1

# setting up the true values of the parameters (Personality)
# rel <- 0.7
# change <- 0.2
# stabch <- 0.25
# sigma =  0.1

# Define the true parameter values
rel_values <- c(0.4, 0.45, 0.5, 0.6, 0.65) # Five different levels of rel
change <- 0.5
stabch <- 0.8
sigma <- 0.1

rel_age_effect <- 0.01
change_age_effect <- 0.04
change_age_eff_linear <- 0.035
stabch_age_effect <- -0.03

age_values <- seq(10, 70, by = 10)
age_dec_c <- (age_values-40)/10
age_quad_values <- age_dec_c^2

# Combine age and age_quad into a dataframe
```

```
age_df <- tibble(age = age_values, age_quad = age_quad_values)

# Define the domain values
domains <- c("Domain1", "Domain2", "Domain3", "Domain4", "Domain5")

# Generate example dataset for plotting
num_corr <- 10000  # Adjust number of correlations for the test

# Calculate the number of samples required
n_short_intervals <- round(2/3 * num_corr)
n_long_intervals <- num_corr - n_short_intervals

# Sample the retest intervals with added randomness
set.seed(123)  # For reproducibility
short_intervals <- sample(seq(0.5, 6, by = 0.5), n_short_intervals, replace = TRUE)
long_intervals <- sample(seq(6.5, 20, by = 0.5), n_long_intervals, replace = TRUE)
sample_retest_intervals <- c(short_intervals, long_intervals)

# Repeat age and domain values to match the length of sample_retest_intervals
repeated_age <- rep(age_values, length.out = length(sample_retest_intervals))
# Repeat domain values to match the length of sample_retest_intervals
repeated_domains <- rep(domains, length.out = length(sample_retest_intervals))


m2_data <- tibble(
 retest_interval = sample_retest_intervals,
 domain = repeated_domains,
 age = repeated_age
) %>%
 mutate(
  age_dec_c = (age - 40) / 10,
  age_quad = age_dec_c^2,
  rel = case_when(
   domain == "Domain1" ~ rel_values[1],
   domain == "Domain2" ~ rel_values[2],
   domain == "Domain3" ~ rel_values[3],
   domain == "Domain4" ~ rel_values[4],
   domain == "Domain5" ~ rel_values[5]
  ),
  change = change,
  stabch = stabch,
  rel_age_effect = rel_age_effect,
  change_age_effect = change_age_effect,
  change_age_eff_linear = change_age_eff_linear,
  stabch_age_effect = stabch_age_effect
 ) %>%
 mutate(
  bounded_rel = pmin(1, pmax(0, rel + rel_age_effect * age_quad)),
  bounded_change = pmin(1, pmax(0, change + change_age_effect * age_quad + change_age_eff_linear * age_dec_c)),
  bounded_stabch = pmin(1, pmax(0, stabch + stabch_age_effect * age_quad)),
  retest = masc(
   rel = bounded_rel,
   change = bounded_change,
   stabch = bounded_stabch,
   retest_interval = retest_interval
  ) + rnorm(n(), 0, sigma)
 ) %>%
 mutate(
  retest = case_when(
   retest > 1 ~ 1,
   retest < 0 ~ 0,
   TRUE ~ retest
  )
 ) %>%
 filter(!is.na(retest))

# Generate true_vals
true_vals <- crossing(
 domain = domains,
```

```
 age_df
) %>%
 mutate(
  rel = pmin(pmax(rep(rel_values, each = nrow(age_df)) + rel_age_effect * age_quad, 0), 1),
  change = pmin(pmax(change + change_age_effect * age_quad + change_age_eff_linear * age_dec_c, 0), 1),
  stabch = pmin(pmax(stabch + stabch_age_effect * age_quad, 0), 1)
 ) %>%
 pivot_longer(cols = c(rel, change, stabch), names_to = "nlpar", values_to = "true_val")


# Calculate prior variance for the computation of posterior contraction
# Simulate from the prior on the logit scale
set.seed(123)
logit_samples <- rnorm(100000, mean = 0, sd = 1)
prior_samples <- inv_logit(logit_samples)

# Calculate the empirical variance of the prior samples
prior_variance <- var(prior_samples)


# DATA SIMULATION ----------------------------------------------------------

full_eval_zscore <- NULL
data_list <- NULL

# Set seed once for reproducibility
set.seed(375)

for (n_sim in c(1:50)) {
 for (num_corr in c(50, 100, 200, 500, 1000, 2000, 3000, 5000, 7500, 10000, 15000)) {

  # Calculate the number of samples required for each segment
  n_short_intervals <- round(2/3 * num_corr)
  n_long_intervals <- num_corr - n_short_intervals

  # Sample the retest intervals

  short_intervals <- sample(seq(0.5, 6, by = 0.5), n_short_intervals, replace = TRUE)
  long_intervals <- sample(seq(6.5, 20, by = 0.5), n_long_intervals, replace = TRUE)
  sample_retest_intervals <- c(short_intervals, long_intervals)

  age <- rep(seq(10, 70, by = 10), length.out = num_corr)
  age_dec_c <- (age - 40) / 10
  age_quad <- age_dec_c^2
  # Repeat domain values to match the length of sample_retest_intervals
  repeated_domains <- rep(domains, length.out = length(sample_retest_intervals))


  m2_data <- tibble(
   retest_interval = sample_retest_intervals,
   domain = domain,
   age = age,
   age_dec_c = age_dec_c,
   age_quad = age_quad,
   rel = rep(rel_values, each = num_corr / length(domains)),
   change = rep(change, num_corr),
   stabch = rep(stabch, num_corr),
   rel_age_effect = rel_age_effect,
   change_age_effect = change_age_effect,
   stabch_age_effect = stabch_age_effect,
   change_age_eff_linear = change_age_eff_linear
  ) %>%
   mutate(
    bounded_rel = pmin(1, pmax(0, rel + rel_age_effect * age_quad)),
    bounded_change = pmin(1, pmax(0, change + change_age_effect * age_quad + change_age_eff_linear * age_dec_c)),
    bounded_stabch = pmin(1, pmax(0, stabch + stabch_age_effect * age_quad)),
    retest = masc(
     rel = bounded_rel,
     change = bounded_change,
```

```
      stabch = bounded_stabch,
      retest_interval = retest_interval
    ) + rnorm(n(), 0, sigma)
  ) %>%
  mutate(
    retest = case_when(
      retest > 1 ~ 1,
      retest < 0 ~ 0,
      TRUE ~ retest
    )
  )

    # Define newdata for parameter extraction
newdata <- crossing(
  retest_interval = 0,
  domain = unique(m2_data$domain),
  age_quad = unique(m2_data$age_quad)
) %>%
  left_join(m2_data %>% select(age_quad, age, age_dec_c) %>% distinct(), by = "age_quad")

family <- brmsfamily(
  family = "student",
  link = "identity"
)

# Setting up priors
# Weakly informative priors
priors <- prior(normal(0, 1), nlpar = "logitrel", class = "b") +
  prior(normal(0, 1), nlpar = "logitchange", class = "b") +
  prior(normal(0, 1), nlpar = "logitstabch", class = "b") +
  prior(cauchy(0,1), class = "sigma")

# Setting up model
formula <- bf(
  retest ~ rel * (change * ((stabch^retest_interval) - 1) + 1),
  nlf(rel ~ inv_logit(logitrel)),
  nlf(change ~ inv_logit(logitchange)),
  nlf(stabch ~ inv_logit(logitstabch)),
  logitrel ~ 1 + domain + age_quad + age_dec_c,     # Include random effects
  logitchange ~ 1 + domain + age_quad + age_dec_c,
  logitstabch ~ 1 + domain + age_quad + age_dec_c,
  nl = TRUE
)

# Fit model (adjust arguments where needed; e.g., "control", "cores", "chains")
m2_fit_masc <- brm(
  formula = formula,
  prior = priors,
  family = family,
  data = m2_data,
  cores = 2,
  chains = 2,
  iter = 3000,
  warmup = 1000,
  backend = "cmdstanr",
  control = list(max_treedepth = 12, adapt_delta = 0.95, step_size = 0.02),
  init = "0")

# Extracting parameter values
epred_rel <- as.numeric(add_epred_draws(m2_fit_masc, newdata = newdata, nlpar = "rel")$.epred)
epred_change <- as.numeric(add_epred_draws(m2_fit_masc, newdata = newdata, nlpar = "change")$.epred)
epred_stabch <- as.numeric(add_epred_draws(m2_fit_masc, newdata = newdata, nlpar = "stabch")$.epred)

# repeat domain and age_quad to match the number of samples
n_samples <- length(epred_rel) / nrow(newdata)
expanded_domain <- rep(newdata$domain, each = n_samples)
expanded_age_quad <- rep(newdata$age_quad, each = n_samples)
expanded_age <- rep(newdata$age, each = n_samples)
expanded_age_dec_c <- rep(newdata$age_dec_c, each = n_samples)
```

```
# Combine the parameter values with the corresponding domain and age_quad values
est_vals <- tibble(rel = epred_rel,
                   change = epred_change,
                   stabch = epred_stabch,
                   domain = expanded_domain,
                   age_quad = expanded_age_quad,
                   age_dec_c = expanded_age_dec_c,
                   age = expanded_age) %>%
  pivot_longer(rel:stabch, names_to = "nlpar", values_to = "val") %>%
  group_by(nlpar, domain, age_quad, age, age_dec_c) %>%
  summarise(est_mean = mean(val), est_sd = sd(val), .groups = 'drop')


eval_zscore <- est_vals %>% left_join(true_vals, by = c("nlpar", "domain", "age_quad", "age"))

# Compare estimated vs. true parameter values ("posterior z-score")
eval_zscore <- eval_zscore %>%
  mutate(post_zscore = (est_mean - true_val) / est_sd,
         est_true_mean_diff = est_mean - true_val,
         post_contraction = 1 - (est_sd^2 / prior_variance))  # Using empirical prior variance

# Extracting & Storing Model Diagnostics
rhat_values <- rhat(m2_fit_masc, pars = c("b_logitrel_Intercept", "b_logitchange_Intercept", "b_logitstabch_Intercept"))

np <- nuts_params(m2_fit_masc)
divergent_transitions <- sum(subset(np, Parameter == "divergent__")$Value)

neff_ratio <- neff_ratio(m2_fit_masc, pars = c("b_logitrel_Intercept", "b_logitchange_Intercept", "b_logitstabch_Intercept"))

model_diagnostics <- tibble(
  nlpar = names(rhat_values),
  R_hat = rhat_values,
  neff_ratio = neff_ratio[names(rhat_values)],
  divergent_transitions = divergent_transitions
) %>%
  mutate(nlpar = case_when(
    nlpar == "b_logitrel_Intercept" ~ "rel",
    nlpar == "b_logitchange_Intercept" ~ "change",
    nlpar == "b_logitstabch_Intercept" ~ "stabch"
  ))

eval_zscore <- eval_zscore %>%
  left_join(model_diagnostics, by = "nlpar") %>%
  mutate(n_sim = n_sim,
         n_cor = num_corr)

# Storing model evaluation metrics and dataset information
full_eval_zscore <- bind_rows(full_eval_zscore, eval_zscore)

# Define newdata for predicting the MASC curve with varying retest intervals
domain_levels <- unique(m2_data$domain)
age_quad_levels <- unique(m2_data$age_quad)
retest_intervals <- seq(0.5, 20, by = 0.5)

newdata_retest <- crossing(
  retest_interval = retest_intervals,
  domain = domain_levels,
  age_quad = age_quad_levels
) %>%
  left_join(m2_data %>% select(age_quad, age, age_dec_c) %>% distinct(), by = "age_quad")

# Predicted MASC curve
predict_retest <- add_epred_draws(m2_fit_masc, newdata = newdata_retest) %>%
  mean_qi() %>%
  rename(retest_predict = .epred)

# Saving data
data_list <- tibble(
```

```
    n_sim = n_sim,
    n_cor = num_corr,
    model_data = list(m2_data),
    masc_pred = list(predict_retest)
   ) %>%
    bind_rows(data_list) }
 }
}
```

Data Simulation Code C4: Interactive Effects of Age and Domain

```
# DESCRIPTION ------------------------------------------------------------------

# In this script we simulate datasets with INTERACTIVE EFFECTS for the constructs risk preference,
# affect and personality.
#
# Author(s): Alexandra Bagaini and Sabine Gisin
# Alexandra Bagaini provided the original simulation script for balanced balenced intercept-only simulation scenarios,
# Sabine Gisin adapted the original script to simulate interactive moderating effects


# PACKAGES ---------------------------------------------------------------
library(tidyverse)
library(tidyr)
library(brms)
library(tidybayes)
library(boot) # logit function
library(data.table)
library(gt)

# FUNCTIONS --------------------------------------------------------------

masc <- function(rel, change, stabch, retest_interval) {

 retest <- rel * (change * (stabch^retest_interval - 1) + 1)

 return(retest)

}

inv_logit <- function(x) {inv.logit(x)}


# CREATING THE DATA SET ---------------------------------------------------------
# setting up the true values of the parameters (Risk Preference)
rel <- 0.5
change <- 0.5
stabch <- 0.8
sigma =  0.1

# setting up the true values of the parameters (Affect)
# rel <- 0.6
# change <- 0.6
# stabch <- 0.9
# sigma =  0.1

# setting up the true values of the parameters (Personality)
# rel <- 0.7
# change <- 0.2
# stabch <- 0.25
# sigma =  0.1

# Define the true parameter values
rel_values <- c(0.4, 0.45, 0.5, 0.6, 0.65) # Five different levels of rel
change <- 0.5
stabch <- 0.95
sigma <- 0.1

rel_age_effect <- 0.01
change_age_effect <- 0.04
change_age_eff_linear <- 0.035
stabch_age_effect <- -0.02

# Interaction effects: These should vary by domain
```

```
change_domain_interaction <- c(0.02, -0.01, 0.03, -0.02, 0.01)
stabch_domain_interaction <- c(-0.005, 0.005, -0.01, 0.005, -0.01)

age_values <- seq(10, 70, by = 10)
age_dec_c <- (age_values-40)/10
age_quad <- age_dec_c^2

# Combine age and age_quad into a dataframe
age_df <- tibble(age = age_values, age_quad = age_quad)

# Define the domain values
domains <- c("Domain1", "Domain2", "Domain3", "Domain4", "Domain5")

# Generate example dataset for plotting
num_corr <- 10000  # Adjust number of correlations for the test

# Calculate the number of samples required
n_short_intervals <- round(2/3 * num_corr)
n_long_intervals <- num_corr - n_short_intervals

# Sample the retest intervals with added randomness
set.seed(123)  # For reproducibility
short_intervals <- sample(seq(0.5, 6, by = 0.5), n_short_intervals, replace = TRUE)
long_intervals <- sample(seq(6.5, 20, by = 0.5), n_long_intervals, replace = TRUE)
sample_retest_intervals <- c(short_intervals, long_intervals)

# Repeat age values to match the length of sample_retest_intervals
repeated_age <- rep(age_values, length.out = length(sample_retest_intervals))

# Repeat domain values to match the length of sample_retest_intervals
repeated_domains <- rep(domains, length.out = length(sample_retest_intervals))

# Calculate prior variance for the computation of posterior contraction
# Simulate from the prior on the logit scale
set.seed(123)
logit_samples <- rnorm(100000, mean = 0, sd = 1)
prior_samples <- inv_logit(logit_samples)

# Calculate the empirical variance of the prior samples
prior_variance <- var(prior_samples)

# DATA SIMULATION ----------------------------------------------------------

full_eval_zscore <- NULL
data_list <- NULL

for (n_sim in c(1:50)) {
 for (num_corr in c(50, 100, 250, 500, 1000, 2000, 3000, 5000, 7500, 10000, 15000, 20000)) {

  # Calculate the number of samples required
  n_short_intervals <- round(2/3 * num_corr)
  n_long_intervals <- num_corr - n_short_intervals

  # Sample the retest intervals with added randomness
  short_intervals <- sample(seq(0.5, 6, by = 0.5), n_short_intervals, replace = TRUE)
  long_intervals <- sample(seq(6.5, 20, by = 0.5), n_long_intervals, replace = TRUE)
  sample_retest_intervals <- c(short_intervals, long_intervals)

  # Repeat age values to match the length of sample_retest_intervals
  repeated_age <- rep(age_values, length.out = length(sample_retest_intervals))

  # Repeat domain values to match the length of sample_retest_intervals
  repeated_domains <- rep(domains, length.out = length(sample_retest_intervals))

  # Create a dataframe with the simulated data
  m2_data <- tibble(
   retest_interval = sample_retest_intervals,
   domain = repeated_domains,
   age = repeated_age
```

```
  ) %>%
   mutate(
    age_dec_c = (age - 40) / 10,
    age_quad = age_dec_c^2,
    rel = case_when(
     domain == "Domain1" ~ rel_values[1],
     domain == "Domain2" ~ rel_values[2],
     domain == "Domain3" ~ rel_values[3],
     domain == "Domain4" ~ rel_values[4],
     domain == "Domain5" ~ rel_values[5]
    ),
    change = change,
    stabch = stabch,
    change_domain_effect = case_when(
     domain == "Domain1" ~ change_domain_interaction[1],
     domain == "Domain2" ~ change_domain_interaction[2],
     domain == "Domain3" ~ change_domain_interaction[3],
     domain == "Domain4" ~ change_domain_interaction[4],
     domain == "Domain5" ~ change_domain_interaction[5]
    ),
    stabch_domain_effect = case_when(
     domain == "Domain1" ~ stabch_domain_interaction[1],
     domain == "Domain2" ~ stabch_domain_interaction[2],
     domain == "Domain3" ~ stabch_domain_interaction[3],
     domain == "Domain4" ~ stabch_domain_interaction[4],
     domain == "Domain5" ~ stabch_domain_interaction[5]
    ),
    rel_age_effect = rel_age_effect,
    change_age_effect = change_age_effect,
    change_age_eff_linear = change_age_eff_linear,
    stabch_age_effect = stabch_age_effect
   ) %>%
   mutate(
    bounded_rel = pmin(1, pmax(0, rel + rel_age_effect * age_quad)),
    bounded_change = pmin(1, pmax(0, change + change_age_effect * age_quad + change_age_eff_linear * age_dec_c +
change_domain_effect * age_quad)),
    bounded_stabch = pmin(1, pmax(0, stabch + stabch_age_effect * age_quad + stabch_domain_effect * age_quad)),
    retest = masc(
     rel = bounded_rel,
     change = bounded_change,
     stabch = bounded_stabch,
     retest_interval = retest_interval
    ) + rnorm(n(), 0, sigma)
   ) %>%
   mutate(
    retest = case_when(
     retest > 1 ~ 1,
     retest < 0 ~ 0,
     TRUE ~ retest
    )
   ) %>%
   filter(!is.na(retest))

  # Define newdata for parameter extraction
  newdata <- crossing(
   retest_interval = 0,
   domain = unique(m2_data$domain),
   age_quad = unique(m2_data$age_quad)
  ) %>%
   left_join(m2_data %>% select(age_quad, age, age_dec_c) %>% distinct(), by = "age_quad")

  family <- brmsfamily(
   family = "student",
   link = "identity"
  )

  # Setting up priors
  # Weakly informative priors
  priors <- prior(normal(0, 1), nlpar = "logitrel", class = "b") +
```

```
  prior(normal(0, 1), nlpar = "logitchange", class = "b") +
  prior(normal(0, 1), nlpar = "logitstabch", class = "b") +
  prior(cauchy(0,1), class = "sigma")



# Setting up model
formula <- bf(
  retest ~ rel * (change * ((stabch^retest_interval) - 1) + 1),
  nlf(rel ~ inv_logit(logitrel)),
  nlf(change ~ inv_logit(logitchange)),
  nlf(stabch ~ inv_logit(logitstabch)),
  logitrel ~ 1 + domain * age_quad + age_dec_c,    # Include random effects
  logitchange ~ 1 + domain * age_quad + age_dec_c,
  logitstabch ~ 1 + domain * age_quad + age_dec_c,
  nl = TRUE
)

# Fit model (adjust arguments where needed; e.g., "control", "cores", "chains")
m2_fit_masc <- brm(
  formula = formula,
  prior = priors,
  family = family,
  data = m2_data,
  cores = 2,
  chains = 2,
  iter = 3000,
  warmup = 1000,
  backend = "cmdstanr",
  control = list(max_treedepth = 12, adapt_delta = 0.95, step_size = 0.02),
  init = "0")

# Extracting parameter values
epred_rel <- as.numeric(add_epred_draws(m2_fit_masc, newdata = newdata, nlpar = "rel")$.epred)
epred_change <- as.numeric(add_epred_draws(m2_fit_masc, newdata = newdata, nlpar = "change")$.epred)
epred_stabch <- as.numeric(add_epred_draws(m2_fit_masc, newdata = newdata, nlpar = "stabch")$.epred)

# Correctly repeat domain and age_quad to match the number of samples
n_samples <- length(epred_rel) / nrow(newdata)
expanded_domain <- rep(newdata$domain, each = n_samples)
expanded_age_quad <- rep(newdata$age_quad, each = n_samples)
expanded_age <- rep(newdata$age, each = n_samples)
expanded_age_dec_c <- rep(newdata$age_dec_c, each = n_samples)

# Combine the parameter values with the corresponding domain and age_quad values
est_vals <- tibble(rel = epred_rel,
                   change = epred_change,
                   stabch = epred_stabch,
                   domain = expanded_domain,
                   age_quad = expanded_age_quad,
                   age_dec_c = expanded_age_dec_c,
                   age = expanded_age) %>%
  pivot_longer(rel:stabch, names_to = "nlpar", values_to = "val") %>%
  group_by(nlpar, domain, age_quad, age, age_dec_c) %>%
  summarise(est_mean = mean(val), est_sd = sd(val), .groups = 'drop')


eval_zscore <- est_vals %>% left_join(true_vals, by = c("nlpar", "domain", "age_quad", "age"))

# Compare estimated vs. true parameter values ("posterior z-score")
eval_zscore <- eval_zscore %>%
  mutate(post_zscore = (est_mean - true_val) / est_sd,
         est_true_mean_diff = est_mean - true_val,
         post_contraction = 1 - (est_sd^2 / prior_variance))  # Using empirical prior variance

# Extracting & Storing Model Diagnostics
rhat_values <- rhat(m2_fit_masc, pars = c("b_logitrel_Intercept", "b_logitchange_Intercept", "b_logitstabch_Intercept"))

np <- nuts_params(m2_fit_masc)
```

```
  divergent_transitions <- sum(subset(np, Parameter == "divergent__")$Value)

  neff_ratio <- neff_ratio(m2_fit_masc, pars = c("b_logitrel_Intercept", "b_logitchange_Intercept", "b_logitstabch_Intercept"))

  model_diagnostics <- tibble(
   nlpar = names(rhat_values),
   R_hat = rhat_values,
   neff_ratio = neff_ratio[names(rhat_values)],
   divergent_transitions = divergent_transitions
  ) %>%
   mutate(nlpar = case_when(
    nlpar == "b_logitrel_Intercept" ~ "rel",
    nlpar == "b_logitchange_Intercept" ~ "change",
    nlpar == "b_logitstabch_Intercept" ~ "stabch"
   ))

  eval_zscore <- eval_zscore %>%
   left_join(model_diagnostics, by = "nlpar") %>%
   mutate(n_sim = n_sim,
        n_cor = num_corr)

  # Storing model evaluation metrics and dataset information
  full_eval_zscore <- bind_rows(full_eval_zscore, eval_zscore)

  # Define newdata for predicting the MASC curve with varying retest intervals
  domain_levels <- unique(m2_data$domain)
  age_quad_levels <- unique(m2_data$age_quad)
  retest_intervals <- seq(0.5, 20, by = 0.5)

  newdata_retest <- crossing(
   retest_interval = retest_intervals,
   domain = domain_levels,
   age_quad = age_quad_levels
  ) %>%
   left_join(m2_data %>% select(age_quad, age, age_dec_c) %>% distinct(), by = "age_quad")


  # Predicted MASC curve
  predict_retest <- add_epred_draws(m2_fit_masc, newdata = newdata_retest) %>%
   mean_qi() %>%
   rename(retest_predict = .epred)

  # Saving data
  data_list <- tibble(
   n_sim = n_sim,
   n_cor = num_corr,
   model_data = list(m2_data),
   masc_pred = list(predict_retest)
  ) %>%
   bind_rows(data_list)

  if (n_sim %% checkpoint_size == 0)  }
 }
}
```

Evaluation Script C5: Corridor of Stability

```
# DESCRIPTION ------------------------------------------------------------------

# In this script I compute the corridor of stability inspired by Schönbrodt and Perugini (2013).
#
# Author: Sabine Gisin


# PACKAGES ---------------------------------------------------------------------
library(tidyverse)
library(gt)

# EVALUATION SCRIPT ------------------------------------------------------------

# Load the simulated data
zscore_eval <- readRDS("zscore_eval.rds")

#get n_cor levels
n_cor_levels <- zscore_eval %>%
 arrange(n_cor) %>%
 distinct(n_cor) %>%
 pull(n_cor)


figure <- zscore_eval %>%
 mutate(n_cor = factor(n_cor, levels = n_cor_levels)) %>%
 ggplot(aes(n_cor, est_true_mean_diff, color = nlpar)) +
 geom_jitter(position = position_jitter(width = 1), size = 1, alpha = 1) +
 geom_hline(yintercept = 0.07, linetype = "dashed")+
 geom_hline(yintercept = -0.07, linetype = "dashed") +
 scale_y_continuous(breaks = c(-0.4, -0.2, 0, 0.2, 0.4)) +
 facet_wrap(~nlpar, nrow = 3) +
 labs(
  x = "Number of Correlations",
  y = "Estimated - True Mean Difference"
 ) +
 theme_minimal() +
 theme_minimal() +
 theme(
  plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 12),
  axis.title.x = element_text(family = "Source Sans 3", size = 12, margin = margin(t = 25, b = 20)),
  axis.title.y = element_text(family = "Source Sans 3", size = 12, margin = margin(r = 20)),
  axis.text.x = element_text(size = 10),  # Increase x-axis digits
  axis.text.y = element_text(size = 10),  # Increase y-axis digits
  axis.ticks = element_blank(),
  strip.text = element_text(size = 12, family = "Source Sans 3"),  # Adjust facet label size
  legend.position = "none",
  legend.title = element_blank(),
  legend.text = element_text(size = 12, family = "Source Sans 3"),
  plot.margin = margin(t = 20, r = 20, b = 20, l = 20),
  panel.grid.major.y = element_line(size = 0.8),
  panel.grid.major.x = element_line(size = 0.8)# Adjust grid size and color
 )


print(figure)
```

Evaluation Script C6: Percentages Table

```
# DESCRIPTION ------------------------------------------------------------------

# In this script I compute the table for percentages within the corridor for w = ±0.07.
#
# Author: Sabine Gisin

# PACKAGES ---------------------------------------------------------------------
library(tidyverse)
library(gt)

# FUNCTIONS --------------------------------------------------------------------

# Function to create the summary table
create_summary_table <- function(df) {
 df %>%
  group_by(nlpar, n_cor) %>%
  count(name = "n_obs") %>%
  left_join(
   df %>%
    group_by(nlpar, n_cor) %>%
    summarise(
     n_inside_corridor = sum(abs(est_true_mean_diff) <= convergence_criterion),
     .groups = 'drop'
    ),
   by = c("nlpar", "n_cor")
  ) %>%
  mutate(perc_inside_corridor = n_inside_corridor / n_obs)
}

# EVALUATION SCRIPT ------------------------------------------------------------

# Load the simulated data
zscore_eval <- readRDS("zscore_eval.rds")

# Define the convergence criterion
convergence_criterion <- 0.07

# Apply the function to create the summary table
summary_table <- create_summary_table(zscore_eval)

# Ensure all levels of n_cor are represented for each nlpar
all_n_cor <- unique(summary_table$n_cor)
all_nlpar <- unique(summary_table$nlpar)
expanded_grid <- expand.grid(n_cor = all_n_cor, nlpar = all_nlpar)

summary_table <- expanded_grid %>%
 left_join(summary_table, by = c("n_cor", "nlpar")) %>%
 replace_na(list(perc_outside_corridor = 0, n_obs = 0, n_inside_corridor = 0))

# Reshape the data to wide format
wide_summary_table <- summary_table %>%
 select(n_cor, nlpar, perc_inside_corridor) %>%
 pivot_wider(names_from = nlpar, values_from = perc_inside_corridor) %>%
 arrange(n_cor)

# Convert the summary table to a formatted table using gt
formatted_table <- wide_summary_table %>%
 gt() %>%
 cols_label(
  n_cor = "Number of Correlations",
  Change = "Change",
  Reliability = "Reliability",
  Stab.Change = "Stab.Change"
 ) %>%
 fmt_percent(
```

```
  columns = c("Reliability", "Change", "Stab.Change"),  # Format these columns as percentages
  decimals = 0
 ) %>%
 cols_align(
  align = "center",
  columns = everything()
 ) %>%
 tab_options(
  table.width = pct(100)
 ) %>%
 tab_style(
  style = list(
   cell_text(weight = "bold")
  ),
  locations = cells_column_labels(
   columns = everything()
  )
 )
```

Evaluation Script C7: Accuracy Thresholds Table

```
# DESCRIPTION ------------------------------------------------------------------

# In this script I compute the table for the different accuracy criteria.
#
# Author: Sabine Gisin

# PACKAGES ---------------------------------------------------------------------
library(tidyverse)
library(gt)

# FUNCTIONS --------------------------------------------------------------------
# Function to create summary table for a given parameter
create_summary_table <- function(df, convergence_criterion) {
 summary_table <- df %>%
  group_by(nlpar, n_cor) %>%
  count(name = "n_obs") %>%
  left_join(
   df %>%
    group_by(nlpar, n_cor) %>%
    summarise(
     n_outside_corridor = sum(abs(est_true_mean_diff) > convergence_criterion),
     .groups = 'drop'
    ),
   by = c("nlpar", "n_cor")
  ) %>%
  mutate(perc_outside_corridor = n_outside_corridor / n_obs * 100)

 return(summary_table)
}
# EVALUATION SCRIPT ------------------------------------------------------------
# Load the simulated data
zscore_eval <- readRDS("zscore_eval.rds")
# Different widths of the corridor and different percentage criteria
convergence_criteria <- c(0.07, 0.12, 0.2)
percentage_criteria <- c(5, 10, 20)
# Initialize an empty list to store results
results <- list()

# Loop through each combination of convergence_criterion and perc_crit
for (convergence_criterion in convergence_criteria) {
 for (perc_crit in percentage_criteria) {
  summary_table <- create_summary_table(zscore_eval, convergence_criterion)


  # Find the accuracy thresholds
  convergence_points_mean_diff <- summary_table %>%
   arrange(nlpar, n_cor) %>%
   filter((perc_outside_corridor < perc_crit & lead(perc_outside_corridor, default = Inf) < perc_crit) |
          (perc_outside_corridor < perc_crit & n_cor == max(n_cor, na.rm = TRUE)))


  # Handle cases where no n_cor meets the criteria
  if (nrow(convergence_points_mean_diff) == 0) {
   cut_off_mean_diff <- tibble(nlpar = unique(summary_table$nlpar), n_cor = NA_real_)
  } else {
   cut_off_mean_diff <- convergence_points_mean_diff %>%
    group_by(nlpar) %>%
    slice_min(n_cor, with_ties = FALSE) %>%
    summarize(nlpar = first(nlpar), n_cor = min(n_cor, na.rm = TRUE), .groups = 'drop')
  }


  # Calculate the cut-off point where all parameters converge
  if (n_distinct(cut_off_mean_diff$nlpar) == length(unique(zscore_eval$nlpar))) {
```

```
    cut_off_all <- cut_off_mean_diff %>%
     summarize(nlpar = "Whole model", n_cor = max(n_cor, na.rm = TRUE))
    if (cut_off_all$n_cor == -Inf) {
     cut_off_all$n_cor <- NA_real_
    }
   } else {
    cut_off_all <- tibble(nlpar = "Whole model", n_cor = NA_real_)  # Use NA_real_ for missing numeric value
   }


   # Combine individual cut-off points with the overall cut-off point
   final_cut_off_table <- bind_rows(cut_off_mean_diff, cut_off_all) %>%
    mutate(
     convergence_criterion = convergence_criterion,
     perc_crit = perc_crit
    )

   # Store the result
   results[[paste0("w_", convergence_criterion, "_perc_", perc_crit)]] <- final_cut_off_table
  }
}

# Combine all results into a single data frame
final_results <- bind_rows(results)

# Ensure the nlpar column is a factor with the desired order
final_results <- final_results %>%
 mutate(nlpar = factor(nlpar, levels = c("Whole model", "Reliability", "Change", "Stab.Change")))

max_value <- max(unique(zscore_eval$n_cor))


# Reshape the data to wide format
reshaped_results <- final_results %>%
 pivot_wider(names_from = c(convergence_criterion, perc_crit), values_from = n_cor) %>%
 arrange(nlpar) %>%
 mutate(across(starts_with("0.07"), ~ ifelse(is.na(.), paste0(">", max_value), .)))

# Create the gt table
formatted_table <- reshaped_results %>%
 gt() %>%
 # tab_header(
 #   title = "Sample Size Thresholds",
 #   subtitle = html("Risk Preference Dataset, Unbalanced<br>Percentage within the corridor")
 # ) %>%
 cols_label(
   nlpar = "Parameter",
   `0.07_5` = "w = 0.07",
   `0.07_10` = "w = 0.07",
   `0.07_20` = "w = 0.07",
   `0.12_5` = "w = 0.12",
   `0.12_10` = "w = 0.12",
   `0.12_20` = "w = 0.12",
   `0.2_5` = "w = 0.2",
   `0.2_10` = "w = 0.2",
   `0.2_20` = "w = 0.2"
 ) %>%
 cols_align(
   align = "center",
   columns = everything()
 ) %>%
 tab_spanner(
   label = "95%",
   columns = c(`0.07_5`, `0.12_5`, `0.2_5`)
 ) %>%
 tab_spanner(
   label = "90%",
   columns = c(`0.07_10`, `0.12_10`, `0.2_10`)
 ) %>%
```

```
 tab_spanner(
  label = "80%",
  columns = c(`0.07_20`, `0.12_20`, `0.2_20`)
 ) %>%
 tab_options(
  table.width = pct(100)
 ) %>%
 tab_style(
  style = list(
   cell_text(weight = "bold")
  ),
  locations = cells_body(
   columns = everything(),
   rows = nlpar == "Whole model"
  )
 ) %>%
 tab_style(
  style = cell_borders(sides = "right", color = "black", weight = px(2)),
  locations = list(
   cells_column_labels(columns = c(`0.2_5`)),
   cells_column_labels(columns = c(`0.2_10`)),
   cells_column_labels(columns = c(`0.2_20`))
  )
 ) %>%
 tab_style(
  style = cell_borders(sides = "right", color = "black", weight = px(2)),
  locations = list(
   cells_body(columns = c(`0.2_5`)),
   cells_body(columns = c(`0.2_10`)),
   cells_body(columns = c(`0.2_20`))
  )
 )
```

Appendix D: Declaration on the Use of Artificial Intelligence

AI Resource	Part of Thesis	Description of Use
Chat GPT	Whole Thesis	Conceptual, Feedback, R Code Feedback, Finetuning English
DeepL	Whole Thesis	Finetuning English