Universiteit Leiden

The Netherlands

Computer Science & Economics

Using AI-Powered Data Extraction to Improve Reproducibility in Scientific Literature

Stijn Jacobus Pleunes

Supervisors:
Dr. T.D.P. Heyman
Dr. E.P.L. van Nieuwenburg

BACHELOR THESIS
Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

09/06/2025

Abstract

This thesis investigates whether artificial intelligence (AI)-powered data extraction can improve reproducibility in scientific literature by enhancing existing tools: *statcheck* and the *GRIM test*. Statcheck is an automated tool that checks whether reported p-values in null hypothesis significance testing (NHST) results are consistent with the accompanying test type, test statistic, and degrees of freedom, but it only works when results are reported in strict APA format. The GRIM (Granularity-Related Inconsistency of Means) test checks whether reported means are mathematically possible, assuming they are based on integer data and the reported sample size. However, it currently lacks any form of automation.

This thesis develops and tests two Python scripts that integrate AI-powered data extraction to improve these tools. The AI-powered statcheck script substantially improves the detection rate of NHST results by identifying them even when they are not reported in strict APA format, while maintaining high accuracy (97-99%). In a subset of articles where all NHST results were present in the `HTML` version of the paper, the script achieved a detection rate of 98.4%.

The AI-powered GRIM script is the first documented attempt to automate the GRIM test, using AI to automatically extract mean values and sample sizes. However, its true positive rate is low, as the relevant GRIM components – such as the mean, sample size, and whether the mean is based on integer data – are often spread out across different parts of the text. Another significant contributing factor to the low true positive rate is that the AI model often struggles to tell whether a mean value is actually GRIM-applicable.

Of the two scripts, only the AI-powered statcheck script currently performs well enough for practical use, showing strong results in both detection percentage and accuracy. Future research should focus on improving the GRIM script's reasoning capabilities and exploring a memory structure approach to help the model link relevant information that is spread out across the text.

Keywords: AI-powered data extraction, statcheck, GRIM test, statistical consistency, NHST, reproducibility

Acknowledgements

I would like to sincerely thank both my supervisors, Dr. Tom Heyman and Dr. Evert van Nieuwenburg, for their flexibility and guidance throughout this project. In particular, I want to thank Tom for introducing me to the world of metascience and bringing this incredibly interesting project to my attention. Tom, you were an amazing supervisor; I could not have asked for better guidance in this project and I am forever grateful for the collaboration that we have had.

I would also like to thank Michèle Nuijten, the creator of the original statcheck tool, for her interest in this project and for providing the data necessary to conduct my analysis.

Finally, I want to thank Nicholas Brown and James Heathers, the creators of the GRIM test, for their shared interest in this project and for providing useful data that made this analysis possible.

Project Links

The full code, a detailed README, and the raw research data are available at the following locations:

- **GitHub (code and README):** https://github.com/s3275744/bachelorThesis
- **OSF (raw `.csv` files):** https://osf.io/ae2pu/

Contents

Thesis Overview

This bachelor thesis was conducted under the supervision of Dr. T.D.P. Heyman and Dr. E.P.L. van Nieuwenburg at the Leiden Institute of Advanced Computer Science (LIACS), as part of the Computer Science & Economics bachelor's programme. This thesis explores how *Artificial Intelligence (AI)-powered* data extraction can enhance reproducibility in scientific literature.

This thesis is structured into three main parts and a general conclusion. The first part provides general context, while the second and third part cover two existing tools, *statcheck* and *the GRIM test*, which can be used to detect potential inconsistencies in scientific literature. This thesis explores if AI can be implemented to improve these existing methods. The final part consists of a general conclusion, which summarises the most important findings of this thesis.

- **Part I: General Context and Background:** covers the importance of both replicability and reproducibility in scientific literature.
- **Part II: statcheck:** covers statcheck, an existing tool designed to automatically identify inconsistencies in null hypothesis significance testing (NHST) results.
- **Part III: The GRIM Test:** covers the GRIM test, a test that can be used to identify inconsistencies in reported means based on sample size and granularity (the smallest possible difference between two values in a dataset).
- **Part IV: General Conclusion:** summarises the key findings of this thesis and reflects on the potential of AI-powered data extraction in improving reproducibility methods.

Part I

General Context and Background

1.1 Context

1.1.1 Reproducibility vs. Replicability

Reproducibility and replicability are two terms that have often been confused [34]. In this thesis, the following definitions will be upheld: *reproducibility* refers to the ability to obtain consistent results using the same input data, computational methods, and conditions of analysis [18]. In contrast, *replicability* refers to the ability to obtain consistent results when a study is repeated using new data. This involves conducting a different experiment that addresses the same research question, to see whether similar findings can be achieved [20]. Both replicability and reproducibility are important concepts for validating scientific findings, but they serve different purposes [18].

As a researcher aiming to validate findings in an article, when choosing to validate through reproducibility, access to the original dataset that the authors used is required. However, these datasets are not always publicly available [17]. In these cases, it is still possible to validate the reproducibility to some extent, using only the data that is present in the paper itself. Examples of such methods are statcheck and the GRIM test.

1.1.2 Statcheck

Statcheck is an R package developed to automatically detect inconsistencies in statistical reporting, specifically in *null hypothesis significance tests (NHST)* reported in APA style (American Psychological Association (2010) [3]). It recalculates the p-value using the reported test type (e.g., t, F, χ^2), test statistic, and degrees of freedom. If the reported and recalculated p-values do not match, statcheck will flag the result as an *inconsistency* [21]. A more elaborate explanation of statcheck can be found in part II of this thesis.

This tool currently has multiple limitations. Its biggest limitation is its detection percentage; Nuijten et al. (2017) [23] state that statcheck's detection percentage is 61.2%. This is because statcheck only detects NHST results if they are reported exactly according to APA guidelines. This thesis aims to improve this percentage by integrating AI into this tool, enabling the detection of non-APA reporting as well.

1.1.3 The GRIM Test

The GRIM (Granularity-Related Inconsistency of Means) test is a simple mathematical check that evaluates whether reported mean values of integer-based data (e.g., survey responses, Likert scales) are mathematically possible given the stated sample size [8]. A more elaborate explanation of the GRIM test can be found in part III of this thesis.

Currently, there is no way to automatically extract a mean value along with its corresponding sample size from text. This thesis aims to provide such a method by developing an AI-powered script that automatically extracts these values and performs the GRIM test.

1.1.4 Potential Impact on Scientific Integrity

Artificial Intelligence (AI) refers to the development of computer systems capable of performing tasks that typically require human intelligence, such as learning, reasoning, problem-solving, and language understanding [35]. Integrating AI into existing methods has the potential to improve scientific integrity by allowing both authors and reviewers to use automated methods, powered by AI, which can scan the document for potential (statistical) reporting inconsistencies.

For **authors**, automated AI-powered tools serve as a pre-submission check, helping to identify (statistical) inconsistencies before submission.

For **reviewers**, automated AI-powered tools provide an additional layer of scrutiny. This allows reviewers to quickly identify any remaining statistical reporting inconsistencies, without having to recalculate each individual value. Given that reviewers often have limited time to properly review an article [15], the use of such tools would be useful, since it allows reviewers to properly focus on the broader methodology of the article.

1.2 Questionable Research Practices

Questionable Research Practices (QRPs) encompass "a range of activities that intentionally or unintentionally distort data in favour of a researcher's own hypotheses" [13]. Examples of such activities are (a) failing to report all of a study's dependent measures, (b) whether to collect more data after looking to see whether the results were significant, and (c) "rounding off" a p-value (e.g., reporting that a p-value of .054 is less than .05) [16]. The third example is a QRP that can be detected using statcheck, as long as the test type, test statistic, degrees of freedom, and tail ('one' or 'two') have not been tampered with.

Wicherts et al. (2011) [40] found that statistical inconsistencies and decision errors were more prevalent in studies where data were not shared, suggesting a potential link between QRPs and reluctance to share data. However, a more recent study by Claessen et al. (2023) [10] did not find robust empirical evidence for this link, as they were unable to replicate the results of Wicherts et al. (2011) [40].

1.2.1 Impact on Scientific Integrity

The prevalence of QRPs hurt scientific integrity by increasing the rate of type I errors, also known as false positives. Simmons et al. (2011) [37] demonstrated that with the current (insufficient) standards for disclosing details of data collection and analysis, the prevalence of false positives become vastly more likely: "In fact, it is unacceptably easy to publish 'statistically significant' evidence consistent with *any* hypothesis."

Simmons et al. (2011) [37] showed through simulations that the cumulative effects of common QRPs can inflate the actual false-positive rate well beyond the nominal α level of a 5% threshold, sometimes exceeding 60% when multiple QRPs are combined.

1.2.2 Solution

To mitigate the prevalence of QRPs, Simmons et al. (2011) [37] proposed a set of six requirements for authors and four guidelines for reviewers, aimed at increasing transparency and reducing the likelihood of false-positive results. These recommendations focus on predefining data collection protocols, fully reporting all variables and conditions, and checking that results do not depend on subjective or arbitrary choices.

However, these guidelines were established prior to the recent advancements in AI. Given the new opportunities presented by AI, we strongly encourage researchers to explore new methods or develop tools to further reduce the prevalence of QRPs. This thesis is also contributes to that effort: subsection 1.3 discusses specific methods we have investigated to mitigate QRPs through AI-powered scripts.

1.3 Developments in AI

1.3.1 Advancements in Large Language Models

AI has undergone rapid advancements in recent years, driven by breakthroughs in areas such as deep learning, natural language processing (NLP), and reinforcement learning [11]. One major development is the advent of large language models (LLMs), such as the *Generative Pretrained Transformer (GPT)* models created by OpenAI [27]. GPT models are capable of tasks such as generating human-like text, content creation, summarisation, data analysis, and data extraction [2].

1.3.2 Data Extraction Capabilities

In the case of this thesis, the data extraction capabilities of GPT models are particularly relevant. These models have shown capabilities of transforming unstructured data into structured data [32]. Specifically, in the case of this thesis, they show potential to automatically identify and extract relevant statistical data from scientific texts, and format them in a structured manner. This is exactly what is needed for the use cases described in this thesis: improving statcheck & the GRIM test.

1.4 Research Question

The GRIM test and statcheck both have several limitations. For instance, statcheck relies heavily on structured reporting (in APA style) and the GRIM test is currently not automated. The recent advancements in AI offer possibilities to overcome those limitations.

The following central research question arises:

"Can AI-powered data extraction improve existing methods, such as statcheck and the GRIM test, to provide a more effective approach for detecting and highlighting inconsistencies in scientific literature?"

This research question can be split into two separate research questions:

1. **Research Question 1 - statcheck:** "Can AI-powered data extraction be used to improve statcheck by allowing for a greater detection rate of NHST results, while maintaining the high level of accuracy (96-99%, [23]) demonstrated by the current statcheck tool?"
2. **Research Question 2 - the GRIM test:** "Can AI-powered data extraction be used to automate the GRIM test, which currently still needs to be carried out manually?"

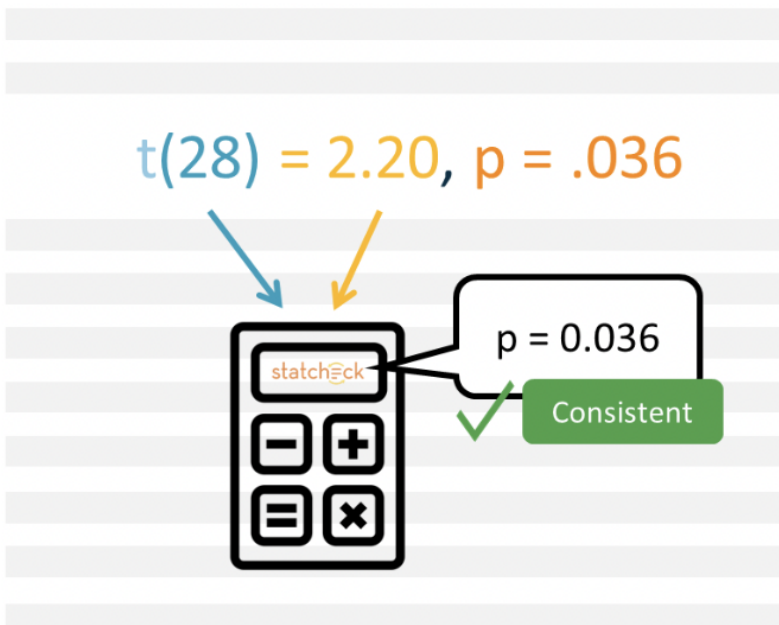
Part II
statcheck

2 Literature Review

Statcheck was first introduced by Michèle B. Nuijten et al. (2016) [21]. It is an R package developed to detect statistical reporting inconsistencies in scientific literature. It works as an automatic tool, meaning it can detect statistical tests without human input. Nuijten describes statcheck as a "'spellchecker' for statistics" [25]. It checks whether the reported p-value matches its accompanying test type, test statistic, and degrees of freedom [25].

Statcheck searches for null-hypothesis significance tests (NHST) in APA style (e.g., `t(28) = 2.20, p < .05`). It recalculates the p-value using the reported test type, test statistic, and degrees of freedom. If the reported and recalculated p-values do not match, statcheck will flag the result as an *inconsistency*.

Figure 1: A visualisation of how statcheck works (image directly sourced from [25])



Null-hypothesis significance testing evaluates whether observed data differs from what would be expected if no effect exists between groups [33]. Statcheck recognises the following null-hypothesis significance tests: Pearson correlations (r), t-, F-, χ^2-, z-, and Q-tests [25].

2.1 Background and Motivation for Development

Statcheck was designed in response to the growing awareness that many published scientific papers, particularly in psychology, contain statistical inconsistencies: errors where the reported p-value does not match the corresponding test statistic and degrees of freedom. This issue has been reported in numerous articles [4, 5, 7, 9, 14, 38, 40]. These studies have shown that roughly half of all published psychological papers contain at least one statistical reporting inconsistency, and that around one in seven papers contain a *gross inconsistency*.

Two types of inconsistencies can be distinguished. A *regular inconsistency* occurs when the reported p-value is incorrect but still leads to the same conclusion regarding (in)significance. In contrast, a *gross inconsistency* occurs when the incorrect p-value leads to a different conclusion about statistical significance, i.e., when a reported p-value is significant but the recalculated p-value is not, or vice versa. These inconsistencies, especially gross inconsistencies, can have significant consequences, as they may affect the conclusions drawn from a study and lead to false claims about statistical (in)significance.

2.1.1 The Steps of statcheck

Statcheck roughly follows four steps when executing its procedure [21]:

Step 1: Conversion

First, the user is asked to provide a file. This file is then converted from its original format (`PDF` or `HTML`) into plain text. `HTML` files to plain text usually convert accurately, while the conversion from `PDF` files to plain text can sometimes be problematic due to typesetting issues. This is because some journals use images or signs such as “<”, “>”, or “=”, instead of the actual character.

Step 2: Extraction

Once a document is converted to plain text, statcheck extracts r, t, F, χ^2 and z statistics (version 1.0.1., [12]), with the accompanying degrees of freedom (df) and p-value. In later versions, statcheck was updated to also extract Q-tests (from meta-analyses) [22]. Statcheck is an automated tool and it is programmed to search for specified strings of text. This means that it can only detect results that are reported exactly in APA style. If a result is not reported according to these standards, statcheck is not able to extract this result.

Step 3: Recalculation

Statcheck calculates a range of valid p-values based on the extracted test statistic, test type, and degrees of freedom. By default, it assumes that the tests are two-tailed, when distributions are symmetric, except for F- and χ^2-tests, which are considered one-tailed. Statcheck takes numeric rounding into account; hence, a valid range of p-values has to be calculated to account for the rounding of test statistics.

p-Value Range Calculation Example

For a reported t-test with `t(30) = 1.96` and `p = 0.059`, statcheck calculates a valid p-value range between the largest and the smallest possible numbers that still round to `1.96`.

- **Lower bound:** `t = 1.964999...` gives a p-value of `0.05873`.
- **Upper bound:** `t = 1.955` gives a p-value of `0.05996`.

Step 4: Comparison

Finally, statcheck compares the recalculated valid p-value range with the reported p-value. Since the reported p-value of `0.059` falls between the recalculated range `0.05873 to 0.05996`, the test is consistent in the example above. If the reported p-value does not fall within the calculated range, statcheck flags the result as an inconsistency. Statcheck can also differentiate between regular inconsistencies and gross inconsistencies.

To take into account one-sided tests, there is an option to try to identify and correct for one-tailed tests. When this option is enabled, statcheck scans the entire text of the article for the words "one-tailed," "one-sided," or "directional." If a result is initially marked as inconsistent, but the article mentions one of these terms, *and* the result would have been consistent if it were one-sided, then the result is marked as consistent.

Note that statcheck does not take into account p-values that are adjusted for multiple testing (e.g., a Bonferroni correction). When it detects an inconsistency in such cases, it cannot assess whether a correction has been applied. As a result, statcheck may (incorrectly) report an inconsistency even though the reported p-value is correct after adjustment.

One-Tailed Consistency Example

Statcheck detects the following result: `t(18) = 2.09, p < .05`. If this result were two-tailed, it would be considered inconsistent, since the two-tailed p-value for the largest possible test statistic that still rounds to 2.09 (i.e., $t = 2.0949\ldots$) is at least 0.0506. Therefore, the result cannot be considered significant under a two-tailed test. However, when one-tailed detection is enabled, if either of the following words are present *anywhere in the paper*: "one-tailed," "one-sided," or "directional," statcheck re-evaluates the result under the assumption of a one-tailed test. The one-tailed p-value for the previously described result equals 0.0253, which means that the result is considered consistent.

Note that statcheck does not check for these terms in the immediate context of the reported test result but searches for them anywhere in the paper. As a result, one-tailed detection may be triggered even if the term refers to a different analysis.

2.1.2 Initial Findings Using Statcheck

When statcheck was first introduced by Nuijten et al. (2016) [21], it was used to assess 258,105 p-values in eight flagship psychology journals. The population of interest in this study was all NHST results reported according to APA guidelines in the full text of articles published between 1985 and 2013. Note that only articles following APA guidelines for reporting statistical results were included.

In addition, the study was limited to articles that were available in `HTML` format, rather than a more common format: `PDF`. This decision was made due to typesetting issues that can occur when converting `PDF` files to plain text, as discussed in subsection 2.1.1 (Step 1).

Results

Table 1 provides an overview of the articles that were selected for this study.

Table 1: Summary of articles reporting NHST results across different journals (table results directly sourced from [21])

Journal	**Subfield**	**Years included**	**No. of articles**	**No. of articles with NHST results**	**No. of NHST results**	**Median no. of NHST results per article with NHST results**
PLOS	General	2000-2013	10,299	2,487 (24.1%)	31,539	9
JPSP	Social	1985-2013	5,108	4,346 (85.1%)	101,621	19
JCCP	Clinical	1985-2013	3,519	2,413 (68.6%)	27,429	8
DP	Developmental	1985-2007	3,379	2,607 (77.2%)	37,658	11
JAP	Applied	1985-2013	2,782	1,638 (58.9%)	15,134	6
PS	General	2003-2013	2,307	1,681 (72.9%)	15,654	8
FP	General	2010-2013	2,139	702 (32.8%)	10,149	10
JPEG	Experimental	1985-2013	1,184	821 (69.3%)	18,921	17
Total			30,717	16,695 (54.4%)	258,105	11

Note. Journals: PLOS = Public Library of Science; JPSP = Journal of Personality and Social Psychology; JCCP = Journal of Consulting and Clinical Psychology; DP = Developmental Psychology; JAP = Journal of Applied Psychology; PS = Psychological Science; FP = Frontiers in Psychology; JPEG = Journal of Experimental Psychology: General.

Table 2: Prevalence of inconsistencies in analysed articles (table results directly sourced from [21])

No. of articles with NHST results	**No. of results**	**Inconsistencies (%)**	**Gross inconsistencies (%)**	**Articles with at least one inconsistency (%)**	**Articles with at least one gross inconsistency (%)**
16,695	258,105	9.7	1.4	49.6	12.9

Table 2 shows the results of applying statcheck on the selected articles. These results show that roughly half (49.6%) of the articles using null hypothesis significance testing (NHST) contained at least one inconsistent p-value (8,273 of the 16,695 articles) and 12.9% (2,150) of the articles contained at least one *gross inconsistency*.

2.1.3 The Validity of Statcheck

In a different study, Nuijten et al. (2017) [23] conducted an experiment to evaluate the validity of statcheck. Statcheck's performance was measured by two key metrics: (a) sensitivity and (b) specificity [1]. In calculating sensitivity and specificity, the following terminology was used [6]:

Sensitivity represents the *true positive rate*: the proportion of *true* (gross) inconsistencies that were also flagged by statcheck as such:

$$\textbf{sensitivity} = \frac{TP}{TP + FN} \tag{1}$$

Specificity represents the *true negative rate*: the proportion of results that are *truly* not (grossly) inconsistent, and statcheck correctly did not flag them as (gross) inconsistencies:

$$\textbf{specificity} = \frac{TN}{TN + FP} \tag{2}$$

Together, sensitivity and specificity provide a measure of statcheck's **accuracy**: the ability to correctly differentiate between consistent and (grossly) inconsistent results, or more mathematically:

$$\textbf{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Note: TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

Ideally, the accuracy should be 100%, which would mean that there are no false positives or false negatives.

Data Used

As a reference set, the same sample from Wicherts et al. (2011) [40] was used, where NHST results from 49 articles published in the *Journal of Experimental Psychology: Learning, Memory, and Cognition* (JEP:LMC) and the *Journal of Personality and Social Psychology* (JPSP) were manually coded. This dataset included only test results that were (a) uniquely reported, (b) complete (i.e., test statistic, degrees of freedom, and p-value reported), and (c) that were reported as being significant (i.e., `p < 0.05`) in the main text or in the tables in the results section. This means that, for example, an NHST result reported in the abstract or footnote is not included in this dataset. Furthermore, only results from t, F, or χ^2-tests were included in this dataset, meaning correlations (r), z-, and Q-tests were not included. The total set consisted of 1,148 NHST results. One article with 28 NHST results was excluded from the reference set, as it was retracted due to misconduct. The final reference set consisted of 48 articles and 1,120 NHST results.

Results

Using statcheck version 1.2.2, statcheck was able to automatically detect 685 of the 1,120 NHST results (**61.2%**). The results of the sensitivity and specificity analysis for statcheck version 1.2.2 can be found in the table below. This table only shows results that are both present in the manually coded validate file as well as in the statcheck output.

Table 3: Results of the sensitivity and specificity analysis of statcheck 1.2.2, with and without one-tailed test detection (table results directly sourced from [23])

	statcheck (default)				**statcheck (with automated one-tailed test detect.)**			
	TP	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**
Inconsistencies	34	26	625	0	29	19	632	5
Sensitivity	100%				85.3%			
Specificity	96.0%				97.1%			
Accuracy	96.2%				96.5%			
Inconsistencies (strict)*	52	8	625	0	47	1†	632	5
Sensitivity	100%				90.4%			
Specificity	98.7%				99.8%			
Accuracy	98.8%				99.1%			
Gross Inconsistencies	8	6	671	0	7	0	677	1
Sensitivity	100%				87.5%			
Specificity	99.1%				100%			
Accuracy	99.1%				99.9%			

† **Note:** In a recalculation of the percentages reported by Nuijten et al. (2017) [23], it became apparent that the FP-value for the one-tailed column in the strict analysis was incorrectly reported. The original article lists this value as 5, but the correct value should be 1. We reached out to Nuijten et al. (2017) [23] to address this inconsistency, and the table in their study has now been altered to display the correct value. This table also shows the corrected value.

* **Inconsistencies (strict)**: the following stricter criteria were applied: (a) results where a p-value is reported as `p = .000` (seven results) or (b) when a Huynh-Feldt correction[1] was applied, but the uncorrected degrees of freedom were reported (11 results), were considered true inconsistencies. The choice was made to consider `p = .000` as inconsistent, since a p-value can never be exactly zero. The APA prescribes that such results should be reported as "$p < .001$" [3]. Using these stricter criteria had no effect in flagging gross inconsistencies.

Table 3 shows that statcheck's sensitivity and specificity for detecting regular inconsistencies were high: between 85.3% and 100%, and between 96.0% and 100%, respectively, depending on which flagging criteria and settings were used. The overall accuracy of statcheck ranged from 96.2% to 99.9%.

The reason why statcheck's accuracy does not reach 100% under the stricter criteria is that, when one-tailed test detection was disabled, there were eight instances where a one-tailed test was conducted but not correctly identified, resulting in false positives. When one-tailed detection was enabled, there were five cases where statcheck was too lenient and failed to flag results that should

[1]The Huynh-Feldt correction adjusts the degrees of freedom in repeated measures ANOVA when the assumption of sphericity is violated, to provide a more accurate p-value.

have been marked as inconsistent. One false positive remained using one-tailed detection, which was caused by a rounding error in the one-tailed detection component of statcheck version 1.2.2 [23].

These results show that when statcheck is able to detect NHST results (i.e., results are reported in APA format), it is a reliable tool to correctly classify these results. It almost always correctly flags the results as either true (gross) inconsistencies or as true consistencies. However, it is important to once again stress that statcheck detected only 61.2% of the NHST results from the reference set, which highlights that statcheck's biggest limitation is its detection rate, which is limited by the fact that results must be reported according to APA standards.

2.1.4 Accounting for Corrections

As reported by Nuijten et al. (2016) [21] and described in subsection 2.1.2, roughly half (49.6%) of the articles using null hypothesis significance testing (NHST) contained at least one inconsistent p-value. A possible cause of the detection of inconsistent results could be the use of statistical corrections for multiple testing. Nuijten et al. (2017) [23] conducted a separate experiment to determine how many inconsistencies were detected due to the use of corrections for multiple testing, using a different dataset.

For example, take the Bonferroni correction for multiple testing. This correction is used to control the Type I error rate by dividing the level of significance (α) by the number of hypotheses tested. However, instead of dividing α, there are often cases in which researchers multiply the p-values by the number of tests. This then results in an internally inconsistent statistical result: the original test statistic and degrees of freedom no longer correspond to the reported (multiplied) p-value [23].

For instance, if you run three different tests, and you want to retain an overall α of .05, the Bonferroni corrected α for each of the tests is $\alpha = .05/3 = .01667$. However, if a researcher incorrectly adjusts the p-value instead, a test yielding p = 0.01 might be reported as p = 0.03 (0.01×3), meaning the original test statistic and degrees of freedom no longer correspond to the reported (multiplied) p-value.

Results

Table 4: Overview of NHST reporting and correction-associated inconsistencies (table results directly sourced from [23])

Correction type	**# APA reported NHST results in selected articles**	**# Inconsistent results**	**# Inconsistencies associated with correction**
Bonferroni	1,108	184	17 (9.2%)
Tukey	1,185	208	0 (0.0%)
Scheffé	898	135	0 (0.0%)
Greenhouse-Geisser	1,646	198	66 (33.3%)
Huynh-Feldt	769	73	14 (19.2%)
Total	**5,606**	**798**	**97 (12.2%)**

Table 4 shows that only 12.2% of inconsistencies are associated with the use of statistical corrections, which indicates that most inconsistencies are not related to these corrections. This indicates that most of the inconsistencies detected using statcheck are actual inconsistencies, meaning the author has made a mistake in reporting their results.

2.1.5 Limitations of statcheck

Statcheck has three notable limitations: (a) it can only detect results reported in APA style, (b) it is affected by typesetting issues, and (c) it does not account for statistical corrections.

APA Style

As already briefly mentioned in subsection 2.1.1 (Step 2): Statcheck is an automated tool and it is programmed to search for specified strings of text. This means that it can only detect results that are reported exactly in APA style. The official manual for statcheck [24] states that, for a result to be detected by Statcheck, it must be reported exactly as follows:

- `t(df) = value, p = value`
- `F(df1, df2) = value, p = value`
- `r(df) = value, p = value`
- $\chi^2(\texttt{df}, N = \texttt{value}) = \texttt{value}, p = \texttt{value}$ (N is optional, ΔG is also included, since it follows a χ^2 distribution)
- `Z = value, p = value`
- `Q(df) = value, p = value` (statcheck can read and distinguish between Q, Q_w (within) and Q_b (between))

Nuijten et al. (2016) [21] includes a great example of which results statcheck cannot detect:

> Limitations of *statcheck*: APA
>
> "It (statcheck) does not read results that deviate from the APA template. For instance, statcheck overlooks cases in which a result includes an effect size estimate in between the test statistic and the p-value (e.g., "$F(2{,}70) = 4.48$, $MSE = 6.61$, $p < .02$") or when two results are combined into one sentence (e.g., "$F(1, 15) = 19.9$ and 5.16, $p < .001$ and $p < .05$, respectively"). These restrictions usually also imply that statcheck will not read results in tables, since these are often incompletely reported."

Formatting and Contextual Limitations

As covered in subsection 2.1.1 (Step 1), there can be typesetting issues when converting a `PDF` file to plain text. This means the correct operator cannot be extracted and the p-value cannot be checked for consistency.

Furthermore, as covered in subsection 2.1.4, statcheck does not account for statistical corrections that have been (incorrectly)[2] applied.

[2] The APA manual does not discuss reporting of statistical corrections. Nuijten et al. (2017) [23] mention that they submitted feedback recommending future editions of the APA Publication Manual include specific examples on how to report these corrections in articles.

3 AI-Powered Methodology

The original statcheck tool requires that results must be reported in strict APA style. To overcome this limitation, a Python script integrating AI-powered data extraction was developed in this thesis. With this enhancement, results no longer need to comply with the APA reporting rules. Furthermore, this script also takes contextual understanding into account and adjusts its calculations accordingly. The script tries to automatically detect whether a test is one-tailed or two-tailed, as well as account for Huynh-Feldt/Greenhouse-Geisser statistical corrections when applicable, ensuring that degrees of freedom are adjusted based on the reported epsilon value. Currently, only Huynh-Feldt/Greenhouse-Geisser corrections are accounted for. Additional corrections could be added in future versions of the script.

The steps this Python script follows and how it functions are described below. The full code and detailed README for the project can be found on GitHub, as listed in the Project Links section.

3.1 How It Works

The AI-powered statcheck script uses the `GPT-4o-mini` AI model to extract relevant data and uses Python-based calculations. In typical NHST test reports, all relevant parameters (e.g., `test_type`, `test_value`, `p_value`) typically appear right next to each other in the text. This makes automated extraction relatively straightforward. The process involves the following steps:

1. **Central class:** the `StatcheckTester` class contains all methods for reading context from files, extracting reported statistical tests, recalculating a valid p-value range, comparison, and presenting results.

2. **Convert:** the `.pdf`, `.htm`, or `.html` file gets converted into plain text. `.txt` files are already in plain text.

3. **Segmentation and overlap:** the plain text is split into segments of 500 words each, with an overlap of 8 words between consecutive segments. Using segmentation, the script does a much better job of correctly identifying all statistical tests in the entire context. The overlap ensures that each statistical test is detected, even if the test spans multiple segments (*e.g.,* a test starting at the end of segment n and ending at the beginning of segment $n + 1$).

4. **Extract data:** The `extract_data_from_text` method uses the `GPT-4o-mini` AI model to identify and extract reported statistical tests from the text. This method transforms unstructured data (tests found in the text) into structured data: a Python list of dictionaries. Each extracted test is represented as a dictionary with the following keys:

   - `test_type`: One of `'r'`, `'t'`, `'f'`, `'chi2'`, `'z'` (string). `'Q'`-tests are not yet included but could be added in a future version of the script.
   - `df1`: First degree of freedom (float or integer). If not applicable, set to `None`.
   - `df2`: Second degree of freedom (float or integer). If not applicable, set to `None`.
   - `test_value`: The test statistic value (float).
   - `operator`: The operator used in the reported p-value (`'='`, `'<'`, `'>'`) (string).

- `reported_p_value`: The numerical value of the reported p-value (float).
- `epsilon`: Only applicable for Huynh-Feldt corrections (float). If not applicable, set to `None`.
- `tail`: `'one'` or `'two'` (string). Assume `'two'` unless explicitly stated.

The prompt used can be found in **Appendix E** or on the GitHub page (see Project Links, file: `testers/statcheck/config.py`).

5. **Apply statistical correction (if applicable):** Currently, the script can only account for Huynh-Feldt/Greenhouse-Geisser corrections. It automatically applies this correction under the following conditions:

   - `test_type == 'f'`,
   - `epsilon` is not `None`,
   - Both `df1` and `df2` are integers.

   If an `epsilon` value is reported but `df1` and `df2` are not integers, this may imply the degrees of freedom have already been adjusted by the `epsilon` value. In this case, the script does not reapply the correction.

   Note that, if a different correction than Huynh-Feldt/Greenhouse-Geisser has been applied, the script behaves like the original *statcheck* tool and calculates the p-value without accounting for the correction. This may result in false inconsistency detections.

6. ***p*-Value calculation:** The `calculate_p_value` method calculates a valid range of p-values (lower, upper) for each extracted test based on its parameters.

7. **Consistency checking:** the `compare_p_value` method checks whether the reported p-value falls within the range of the valid p-values (lower, upper). The script also makes a distinction between regular inconsistencies and gross inconsistencies.

8. **Processing results:** after extraction and testing, the results are added into a DataFrame and printed. Each test is displayed in a separate row with the following column headers:

   - **Consistent:** indicates whether the reported p-value falls within the valid recalculated range (`Yes` or `No`).
   - **APA Reporting:** displays the correct APA reporting of the detected test, regardless of how the test is reported in the context.
   - **Reported *p*-Value:** the p-value as originally reported in the text.
   - **Valid *p*-Value Range:** the range of valid p-values (lower, upper) based on the test type, test statistic, and degrees of freedom.
   - **Notes:** any additional information regarding the result, such as the presence of gross or regular inconsistencies or the usage of a statistical correction.

Using this approach, it should be possible to analyse results that are not reported in APA format, as well as try to incorporate relevant context, such as which tail is used.

4 Experiment

4.1 Data Used

To evaluate the AI-powered Python statcheck script, the dataset originally compiled by Wicherts et al. (2011) [40] was used. This dataset was also used by Nuijten et al. (2017) [23] to evaluate the validity of the original statcheck tool (refer to section 2.1.3).

The initial dataset consisted of 49 articles; however, nine of these were excluded for specific reasons. One article was retracted due to misconduct, and the remaining eight contained multiple NHST results reported in tables that were not available in the `HTML` version of the articles, making it impossible for the script to identify these results. The final dataset used for evaluation consisted of 40 articles and 869 NHST results.

The specific inclusion criteria that an NHST result must meet in order to be included in the dataset are described below, as well as in subsection 2.1.3. Results must be:

- Uniquely reported.
- Complete, with test statistic, degrees of freedom, and p-value reported.
- Reported as significant ($p < 0.05$) in the main text or tables in the results section. For example, NHST results reported in the abstract or footnote are not included.

The articles on which the data are based and on which the AI-powered Python statcheck script was run are published on the private web page https://osf.io/ske8z/, and can be shared by Nuijten et al. upon request.

4.2 Copyright Notice

The articles in this dataset are protected by copyright, meaning that their full content cannot be freely shared or used without permission from the respective copyright holders. This means that these articles may not be uploaded in the ChatGPT web-interface, since the data uploaded there may be used by OpenAI to train their models [28].

However, since the script created uses the ChatGPT API, the copyright restrictions are respected. Data provided via the API are not used by OpenAI to train or improve models, as is stated in OpenAI's Key Concepts documentation [29].

4.3 Procedure

The `testers/statcheck/main_multiple_runs.py` script was run on all 40 articles in the final dataset. This script automatically analyses each article three times, and uses the most frequent output for final analysis. This ensures a more consistent result than analysing an article just once. The full code for this script can be found on the aforementioned GitHub page (refer to Project Links). Each NHST result found in the output for an article was manually checked against the manually coded validate file from Wicherts et al. (2011) [40]. The following categories were logged:

- **Article:** Journal name and author(s) of the article.
- **# Significant results in validate file:** The total number of significant NHST results present in the manually coded validate file.

- **# Correctly identified by script:** The number of significant NHST results successfully identified by the script. That is, results that meet the inclusion criteria mentioned in both subsection 2.1.3 and 4.1.
- **# Missed results in validate file:** The number of significant NHST results that meet the inclusion criteria but were not included in the manually coded validate file.
- **# Missed by script:** The number of significant NHST results present in the manually coded validate file but not detected by the script.
- **True Positives (TP):** The number of results correctly flagged as a (gross) inconsistency.
- **False Positives (FP):** The number of results incorrectly flagged as a (gross) inconsistency.
- **True Negatives (TN):** The number of results correctly not flagged as a (gross) inconsistency.
- **False Negatives (FN):** The number of results incorrectly not flagged as a (gross) inconsistency.
- **Notes:** Any additional information regarding the result, such as the presence of gross or regular inconsistencies.

The detailed results of this analysis can be found in **Appendix A**. The filtered raw output of the script can be found in **Appendix B**. The `.csv` file of the complete raw output can be found on the OSF page of this project (see the Project Links section for the link). A list of the five missed NHST results by the manually coded validate file can be found in **Appendix C**.

Note that each result that is present in **Appendix C** – i.e., any result found by the script but not found in the manually validated code – was located in the original article to ensure it indeed is present in the text and is not a hallucination from the AI model.

5 Results

5.1 Detection Rate

Table 5 shows the results of applying the AI-powered statcheck script to the dataset. The table also shows a breakdown of the missed results: five cases were missed in the manually coded validate data file, and 64 cases were not identified by the script. 805 results were found in both outputs (manually coded validate data file and script), which corresponds to a detection percentage of **92.6%**. Table 6 provides further insights into the specific categories of missed NHST results by the script.

Table 5: The total number of NHST results detected by the AI-powered statcheck script

# NHST results	Count	Percentage detected
Total in validate data file	869	
Correctly identified by script	810	93.2%
Missed in validate data file	5	
Missed by script	64	
In both outputs	805	92.6%

Table 6: Categories of missed NHST results by the script

Category of missed NHST results	Count	Percentage of total missed
1. Incomplete NHST result (No p-value reported)	45	70.3%
2. Not extracted (Truly missed by script)	13	20.3%
3. Same result consecutively	5	7.8%
4. Altered result in validate data file	1	1.6%
Total missed results	**64**	**100%**

When taking a closer look at the specific categories of missed NHST results by the script, it becomes apparent that only 13 of the 64 results were truly missed (`Not extracted`). The largest category of missed results, `Incomplete NHST result`, accounted for 70.3% of the total missed results. However, these results should not have been included in the manually coded validate file, as they do not comply with the second inclusion criterion mentioned in subsection 4.1:

Inclusion criterion 2

"Results must be complete, with the test statistic, degrees of freedom, and p-value reported."

An example where such incomplete results are reported:

Example of incomplete NHST results

"In a two-way analysis of variance (ANOVA), response times were found to be affected both by n (n = 1–5), $F(1,4) = 11.53$, $MSE = 9,730.73$; and trial type, $F(1,4) = 50.56$, $MSE = 15,955.12$ ("yes" response time = 859 ms, "no" response time = 1,113 ms)." [39]

In this example, the p-values corresponding to the reported F-tests are missing, and therefore the results are considered incomplete.

Similarly, the category *Same result consecutively*, which accounted for 7.8% of the total missed results, should also not have been included in the manually coded validate file. These results fail to comply with the first inclusion criterion outlined in subsection 4.1:

Inclusion criterion 1

"Results must be uniquely reported."

An example of the same NHST result being reported consecutively is:

Example of same NHST result reported consecutively

"76% of the participants (55/72) selected the Black candidate regardless of qualifications, $\chi^2(1, N = 72) = 20.06$, $p < .001$. Replicating results from earlier studies, 84% of participants selected the candidate with the higher GPA when this candidate was Black, but this number dropped to 32% when that candidate was White, $\chi^2(1, N = 72) = 20.06$, $p < .001$." [19]

In this example, it even seems like a reporting error has been made. Two different results are described, yet the exact same χ^2-statistic and p-value are reported. Given the differences in the conditions being compared, it is highly unlikely that the same statistical values would be reported for both tests.

Finally, the category `Altered result in validate data file`, which accounted for 1.6% of the missed results, should also not have been included in the manually coded validate file. This is because the reported result is a transformation of the result present in the original article. The article reported the following: `F(1, 53) = 2.91, one-tailed`, which was manually converted to a t-value of 1.7059 in the validate file. Although this transformation is mathematically valid, as an F-test where the first degree of freedom is 1 is equivalent to a squared t-test ($F = t^2$), the script did not detect the transformed value because it was not reported as a t-test in the original article.

If we exclude these categories, the final dataset consists of 818 NHST results. Of these, the script successfully identified 805 results, resulting in a detection percentage of **98.4%**. Furthermore, the script also identified 5 additional NHST results that were not included in the manually coded validate file, although these specific results **indeed** meet the inclusion criteria. This shows that the script has detected even more results than accounted for in the reference set.

5.2 Sensitivity, specificity, and accuracy

Table 7 shows the sensitivity, specificity, and accuracy analysis results of the AI-powered Python script. For this table, only results that appear in both outputs (manually coded file and script) were included.

Table 7: Results of the sensitivity, specificity, and accuracy analysis where NHST results were both detected by the AI-powered statcheck script as well as present in the validated data file

AI-powered Python statcheck script with automatic one-tailed test detection				
	TP	**FP**	**TN**	**FN**
Inconsistencies	53	11	740	1
Sensitivity		98.15%		
Specificity		98.54%		
Accuracy		98.51%		
Inconsistencies (strict)*	60	4	740	1
Sensitivity		98.36%		
Specificity		99.46%		
Accuracy		99.38%		
Gross inconsistencies	9	1	795	0
Sensitivity		100.00%		
Specificity		99.87%		
Accuracy		99.88%		

Note: TP = True Positives; FP = False Positives; TN = True Negatives; FN = False Negatives.

* **Inconsistencies (strict)**: in 7 cases, the reported p-value was "$p = .000$". These were considered *true positives* in the strict analysis, since a p-value can never be exactly 0, whereas the validated file labelled them as *true negatives*.

In the strict analysis, four *false positives* remain. Three of these are due to incorrect operator extraction (e.g., ">" is extracted, but "<" is reported). The final remaining *false positive* is due to wrong tail detection (e.g., "two" extracted, but "one" used).

The analysis shows that the script has failed to correctly identify an inconsistency once, leading to a *false negative*. This occurred because results reported with two trailing zeros (e.g., "4.00") are interpreted by Python as having a single decimal place rather than two. This means that the valid range of p-values becomes too large. This issue has since been resolved, so if a reanalysis were to occur, the script should correctly flag this result as a *true positive*.

Comparing the AI-Powered Script With the Original Statcheck Tool

Table 8 shows a comparison of detection percentage, sensitivity, specificity, and accuracy for the AI-powered script and the original statcheck tool (default and one-tailed detection). Note that, if a reanalysis were to occur, the sensitivity of the AI-powered script would likely reach 100%, as the script has been updated to correctly handle trailing zeros. This issue previously caused the single false negative, which is why the current sensitivity is not at 100%.

Table 8: Comparison of different metrics for the AI-powered script and the original statcheck tool (default and one-tailed detection) using the strict analysis criteria. Best results are shown in bold.

	AI-powered script (automatic tail detection)	**Original statcheck (one-tail detection disabled)**	**Original statcheck (one-tail detection enabled)**
Inconsistencies (strict)			
Detection percentage	**98.4%**†	61.2%	61.2%
Sensitivity	98.4%	**100%**	90.4%
Specificity	99.5%	98.7%	**99.8%**
Accuracy	**99.4%**	98.8%	99.1%

† **Note:** This result is not directly comparable to the original statcheck tool due to differences in the dataset used. See Section 6 for details.

5.3 Runtime and Cost

The total runtime for processing 48 articles three times each is 6,358.51 seconds, resulting in an average runtime of **132.47 seconds per article**. The total cost for this analysis amounts to $0.90, which results in **$0.0188 per article** for three iterations. These costs reflect the usage of calling the OpenAI API, which is needed to run the AI-powered script. Note that these statistics are based on all 48 articles before the exclusion of eight `HTML` articles, which took place after this analysis.

6 Discussion

Excluding categories of NHST results that should not have been included in the manually coded validate file, the AI-powered script achieves a detection percentage of 98.4%. This is an improvement of 37.2 percentage points compared to the 61.2% detection rate reported in the validity study conducted by Nuijten et al. (2017) [23] for the original statcheck tool.

However, it is important to note that these results are not directly comparable, as this study used a smaller dataset than that of Nuijten et al. (2017) [23]. Specifically, the initial dataset consisted of 49 articles, of which nine were excluded – one due to retraction for misconduct, and eight because multiple NHST results were reported in tables that were unavailable in the `HTML` version. As a result, the final dataset used for evaluation consisted of 40 articles and 869 NHST results, of which an additional 51 were excluded due to failing to meet the inclusion criteria; the reasons for exclusion are summarised in Table 6. In contrast, Nuijten et al. (2017) [23] only excluded the article that had been retracted due to misconduct. In order to truly compare the AI-powered script and the original statcheck tool, both tools would need to be evaluated on the same dataset.

Given its performance, we argue that the AI-powered statcheck script should be used by both authors and reviewers as a quick, automatic, and reliable tool for checking NHST results reported in an article.

6.1 Limitations

The primary limitation in evaluating the results in terms of detection percentage, sensitivity, specificity, and accuracy is the lack of complete (100%) accuracy in the manually coded validate file. Several NHST results were included in the validate file despite not meeting the inclusion criteria (refer to Table 6). Additionally, this study has shown that five NHST results, correctly identified by the AI-powered script, were missing from the manually coded file even though they met the inclusion criteria. Since the manually coded file is not entirely reliable, the evaluation of the script's performance cannot be fully validated.

This script has not been tested on articles that include NHST results in tables. This is because there were no tables reported in the `HTML` versions of the articles. The full articles (including tables) were not available during this analysis, causing these articles to be excluded. Furthermore, the AI-powered script has the same typesetting issues as the original statcheck tool, as is covered in subsection 2.1.1 (Step 1).

Another limitation is that the manually coded validate file only covers t-, χ^2-, and F-tests. Other statistical tests supported by the script, such as Pearson correlations (r) and z-tests were not included in the evaluation, and thus it was not assessed how well the model performs on these types of tests. Furthermore, special cases like handling Huynh-Feldt corrections were rare in this dataset, making it difficult to assess how well the model performs in handling such situations in general.

7 Conclusion

This study has shown that the AI-powered script has similar results in terms of sensitivity, specificity, and accuracy as the original statcheck tool. Both tools score well in these metrics (98-100%). Although results are not entirely comparable due to differences in the dataset, this study has shown that creating an AI-powered statcheck script greatly increases its detection percentage. Even if the originally excluded articles were included in the total number of NHST results, the AI-powered script would still outperform the original statcheck tool, even if it were unable to detect *any* NHST results in those articles. The original statcheck tool detected 685 out of 1,120 NHST results, whereas the AI-powered script identified 805. Therefore, it can be confidently concluded that the AI-powered script outperforms the original statcheck tool in terms of detection percentage. The AI-powered script also performs well in automatically identifying whether a one- or two-tailed test was used, since only one false positive was caused by wrong tail detection, which is just one out of 805 total NHST results.

However, it is important to acknowledge that errors related to wrong tail detection which do *not* result in statistical inconsistencies, may go unnoticed. For example, if the script incorrectly identifies a test as two-tailed when it was actually one-tailed but the reported p-value remains consistent under both assumptions, this error will not be flagged. As a result, these wrong tail detections go unnoticed in the analysis, although it is also possible they did not occur at all.

7.1 Future Research

We recommend that future research begins by contacting Wicherts et al. (2011) [40] and reassessing the manually coded validate file to address its inaccuracies. Once a 100% accurate and reliable validate file has been created, a reanalysis of both tools should take place on the complete dataset (i.e, 48 articles). This reanalysis should include complete versions of the articles (e.g., `PDF` format with all tables), to ensure both tools have a fair opportunity to detect the NHST results, as the tables are missing from the `HTML` versions. The results from a more complete reanalysis would allow for a definitive comparison of the detection percentages, sensitivity, specificity, and accuracy of the AI-powered script and original statcheck tool. This reanalysis would also lead to a better reflection of real-world conditions, as articles containing table-based results were deliberately excluded in the current analysis.

In addition, it would be valuable to develop or use a more comprehensive manually coded file that includes a larger range of test types (e.g., Pearson correlations (r) and z-tests) and includes more rare or challenging cases, such as more statistical corrections and correction types beyond Huynh-Feldt.

Additionally, future research could involve incorporating more statistical corrections into the script. Currently, this script only automatically accounts for Huynh-Feldt/Greenhouse-Geisser corrections. This shows that it is possible to use AI to automatically adjust calculations based on statistical corrections, paving the way for other statistical corrections, such as Bonferroni, to be added.

Finally, `‘Q’`-tests could be added as an NHST test type.

Part III
The GRIM Test

8 Literature Review

The GRIM (Granularity-Related Inconsistency of Means) test is a statistical test developed to identify potential errors in the reporting of mean values in scientific research, created by Nicholas J.L. Brown & James A.J. Heathers (2016) [8]. The GRIM test checks the consistency of *reported mean values* with *mathematically possible mean values* given the sample size and scale used, usually integer or Likert-scale data. It identifies impossible means that cannot mathematically exist given the relevant parameters, indicating potential reporting errors.

To apply the GRIM test, the following variables should be considered: (a) reported mean, (b) the sample size, and (c) the type of data used, e.g. integer values obtained from a Likert scale. The test only works for means that are composed of integer data, since the calculation relies on the granularity of integer values to determine whether the mean is mathematically possible given the sample size. The test works by multiplying the reported mean with the sample size and then evaluating whether the product of this calculation could result from a plausible sum of individual data points [8].

Consider the following fictional extract, created to provide an example use case of the GRIM test:

> *GRIM test*: example use case
>
> Participants (N = 60) were randomly assigned to consume 250 ml of either "extra healthy water", containing essential vitamins and minerals (experimental condition, N = 30) or standard sparkling water (control condition, N = 30). Thirty minutes after consuming the beverage, all participants were asked to answer the question "How healthy do you currently feel?" using a 7-point Likert scale. The scale ranged from 1 (Not at all) to 7 (Extremely healthy). Participants in the "extra healthy water" condition reported a significantly higher level of perceived health (M = 6.02, SD = 1.12) compared to those in the control group (M = 4.12, SD = 1.45).

At first glance, these results seem plausible, but they are mathematically impossible. The GRIM test can be applied to detect such inconsistencies. Given that the data are collected on a 7-point Likert-scale, where participants can only choose integer values (1, 2, 3, etc.), and the sample size for both groups is 30, the reported mean must be a result of summing these integers and dividing by the sample size.

Consider the first group, the "extra healthy water" condition. To verify the consistency of the reported mean (M = 6.02), it is multiplied by the sample size (N = 30), resulting in a product of 180.6. The two integers that give a result closest to the reported mean of 6.02 are 180 and 181. However, neither 180 nor 181, when divided by the sample size (30), results in the reported mean of 6.02. Specifically, 180 divided by 30 yields a mean of exactly 6.00, and 181 divided by 30 yields approximately 6.03. Therefore, it is impossible to obtain a mean of 6.02 with any combination of integer responses in this scenario.

Now, consider the second group, the standard sparkling water condition. Again, the reported mean (M = 4.12) is multiplied by the sample size (N = 30), which results in a product of 123.6. The closest integers, 123 and 124, would yield means of 4.10 and approximately 4.13, respectively. Thus, just as with the first group, the reported mean of 4.12 cannot be achieved through any valid combination of responses.

These errors can occur for a number of different reasons, such as typographical mistakes, incorrect reporting of the number of participants excluded from analyses, rounding errors, or even deliberate fraud, where the author manipulates the results to achieve a desired outcome [8].

Such inconsistencies may also be the result of Questionable Research Practices (QRPs), in which researchers intentionally or unintentionally alter their analyses or reporting to obtain favourable results. For example, participants whose responses negatively impact the desired outcome might be purposefully excluded from the analysis, deliberately inflating or deflating mean values to support the research hypothesis [16].

8.1 What Is Granularity?

In the context of the GRIM test, granularity refers to "the numerical separation between possible values of the summary statistics" [8]. In simpler words, granularity describes the smallest possible difference between two values in a dataset. The formula for granularity can be expressed as:

$$\text{Granularity} = \frac{1}{(N \times L)} \tag{4}$$

where N is the number of participants and L is the number of items in the measure (i.e., the number of individual questions/statements used to compute the composite score). When no composite measure is used (i.e., $L = 1$), the granularity becomes $\frac{1}{N}$, meaning the smallest possible difference between two means is solely determined by the number of participants. This scenario can occur when only collecting one piece of information per participant.

Take our previous example: rating one's perception of health on a 7-point Likert-scale. In this case, since there is only one item being measured (perceived health), $L = 1$, and the granularity of the mean is dependent only on the sample size N. Take one of the groups, consisting of 30 participants, the granularity of the mean would be $\frac{1}{30} = 0.033$. This means that the smallest possible difference between the two average values reported in the study would be 0.033. If the mean reported in the study differs from a multiple of this granularity, there is an inconsistency present, which could be detected using the GRIM test.

8.2 Limitations of the GRIM Test

The GRIM test has two notable limitations: (a) the effectiveness of the test diminishes as the sample size increases, and (b) there is currently no automation of the test.

Diminished Effectiveness

The GRIM test can only be used to check for inconsistencies when the product of the number of participants N and the number of items L is smaller than 10^D, where D is the number of decimal places reported. The formula which checks if the GRIM test can be applied, can be expressed as:

$$(L \times N) < 10^D \tag{5}$$

In most scientific research, means are commonly rounded to two decimal places ($D = 2$). This means that the GRIM test can only be applicable when the product of $L \times N$ is less than 100. This is because as $L \times N$ approaches or exceeds 100, the differences between possible mean values become so small that, after rounding, all mean values within the valid range become possible. In such cases, the GRIM test can no longer identify impossible means.

Note that percentages reported to one decimal place can typically be tested for consistency with a sample size up to 1,000 (i.e., L = 1), as they are, in effect, fractions reported to three decimal places (e.g. 53.2% = 0.532) [8].

Figure 2: Plot of possible and impossible mean values *(D = 2)* (image directly sourced from [8])

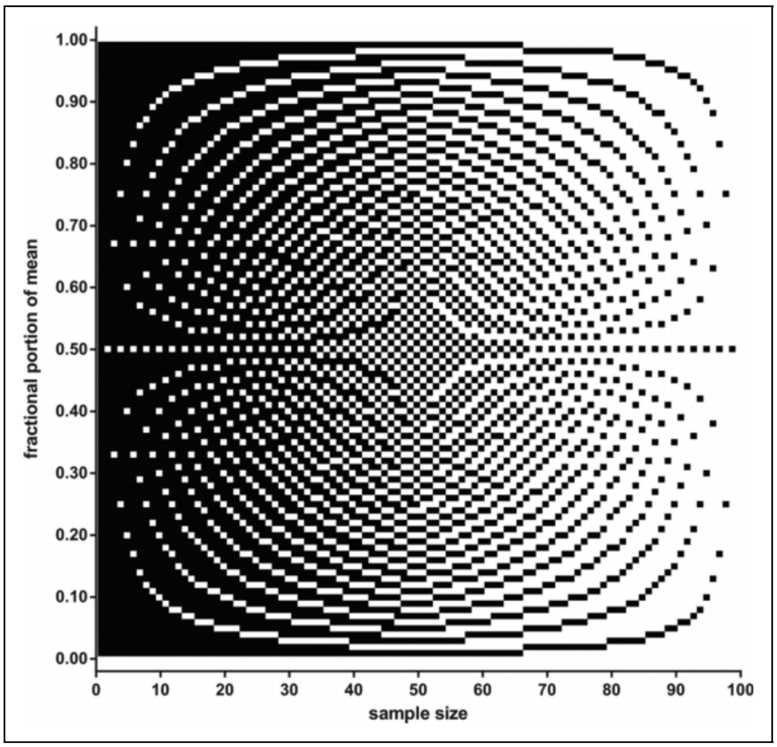


Figure 2 shows a plot that is a function of the sample size (represented on the x-axis) and the fractional portion (decimal part) of the mean (represented on the y-axis). In this figure, the black dots indicate mean values that are mathematically impossible given the sample size and rounding to two decimal places, whereas the white dots represent mean values that are possible. Note that in this figure, numbers ending in exactly 5 at the third decimal place (e.g., 1/8 = 0.125) were always rounded up. If such means were allowed to be either rounded up or down, a few extra white dots would appear at sample sizes that are multiples of 8.

As shown in Figure 2, as the sample size starts to increase towards 100, the number of white dots increases, meaning more mean values become possible, and thus the GRIM test's effectiveness diminishes.

Lack of Automation

Currently, there is no tool available that scans a document and automatically performs the GRIM test when applicable. This means that researchers and reviewers must manually check each reported mean for consistency. This manual process can be time-consuming and prone to human error. For example, Brown & Heathers (2016) [8] themselves mention that they have made a human error when analysing an article: "In one of the cases above, the data that we received showed that we had failed to completely understand the original article; what we had thought were inconsistencies in the means on a Likert-type measure were due to that measure being a multiple-item composite, and we had overlooked that it was correctly reported as such." [8]

Therefore, the development of such an automated tool could potentially minimise the risk of human error, as well as improve efficiency, as reviewers no longer have to perform the test manually.

9 AI-Powered Methodology

This thesis aims to create an automated implementation of the GRIM test by developing an AI-powered Python script, which uses AI to extract relevant data from a certain context and uses Python for its necessary calculations. This approach is very similar to the approach used for the statcheck AI-powered methodology described in Section 3. However, unlike statcheck's methodology, creating an AI-powered methodology for the GRIM test is more complex.

In statcheck, all relevant parameters (e.g., `test_type`, `test_value`, `p_value`) typically appear right next to each other in the text, making automated extraction relatively straightforward. However, GRIM-relevant parameters are often spread out across different sentences or even different paragraphs. Furthermore, for each mean value found, the AI model has to find the correlating sample size **and** determine if the mean value is GRIM-applicable (i.e., the mean value is composed of integer data).

Since the relevant parameters are more spread out for GRIM, the size of each segment has been set at a value of 1,000 words and the overlap is set at 200 words. Furthermore, since this model needs to properly understand the context in order to determine if a found mean value is GRIM applicable or not, the choice has been made to use the full `GPT-4o` model rather than the smaller `GPT-4o-mini` model, despite the fact that both input and output API tokens are 12.5 times more expensive for `GPT-4o` [30]. `GPT-4o` outperforms `GPT-4o-mini` in reasoning-related benchmarks such as the *Massive Multitask Language Understanding* benchmark (MMLU), where `GPT-4o` achieves an accuracy of 88.7% compared to 82.0% for `GPT-4o-mini`, and the *Discrete Reasoning Over Paragraphs* benchmark (DROP), where `GPT-4o` scores 88.4% compared to 79.7% for `GPT-4o-mini` [26].

The AI-powered methodology for the GRIM test includes a final reasoning step. In this step, the model reflects on each extracted mean value and provides a brief reasoning of why it considers the value to be GRIM-applicable.

Additionally, the following formula was implemented to determine whether the GRIM test can be theoretically applied due to sample size constraints:

$$N < 10^d \tag{6}$$

Here, N is the sample size and d is the number of decimal places in the reported mean. For example, for a mean value with two decimals, if the sample size exceeds a value of 100, the found mean gets removed as an entry. Note that this is a simplified version of the full formula (see formula 5). This is because initial attempts at detecting composite measures were unsuccessful, leading to the decision not to implement this functionality at this time.

The steps this Python script follows and how it functions are described below. The full code and detailed README for the project can be found on GitHub, as listed in the Project Links section.

9.1 How It Works

The AI-powered GRIM script uses the `GPT-4o` AI model to extract relevant data and uses Python-based calculations. The process involves the following steps:

1. **Central class:** the `GRIMTester` class contains all methods for reading context from files, extracting reported means and sample sizes, validating whether GRIM is applicable, performing

the GRIM test, and presenting results.

2. **Convert:** the `.pdf`, `.htm`, or `.html` file gets converted into plain text. `.txt` files are already in plain text.

3. **Segmentation and overlap:** the plain text is split into segments of 1,000 words each, with an overlap of 200 words between consecutive segments. This larger context window increases the chances that all relevant parameters are captured together in the same context window.

4. **Extract data:** the `extract_data_from_text` method uses the `GPT-4o` AI model to identify and extract reported means and sample sizes from each segment. The model is instructed to extract values only if:

   - The value is explicitly labelled as a `mean`.
   - The mean is based on integer data (e.g., Likert scales).
   - A sample size (`N`) is explicitly mentioned and clearly linked to the mean.

   Besides extracting relevant data, the model is also instructed to return a `reasoning string`, in which the model must provide a brief justification for why the identified mean value is considered GRIM-applicable.

   This method transforms unstructured data (tests found in the text) into structured data: a Python list of dictionaries. Each extracted test is represented as a dictionary with the following keys:

   - `reported_mean`: The mean value as reported in the article (float).
   - `sample_size`: The sample size associated with the reported mean (integer).
   - `discrete_reasoning`: A brief explanation of why the mean value is considered GRIM-applicable (string). For example: `"mean of 7-point Likert responses clearly linked to N = 28 in same sentence"`.

   The prompt used can be found in **Appendix F** or on the GitHub page (see Project Links, file: `testers/GRIM/config.py`).

5. **GRIM applicability check:** this check removes entries from the final output that are not theoretically testable using the GRIM formula. More specifically, the sample size must not exceed 10^d, where d is the number of decimal places in the reported mean. This is because *any* mean value where the sample size exceeds this threshold can be constructed from integer data. To reduce clutter in the final output, these entries are excluded form the final results.

6. **GRIM test:** the `grim_test` method calculates if the reported mean is mathematically possible given the sample size and number of decimal places.

7. **Processing results:** after extraction and testing, the results are added into a `DataFrame` and printed. Each test is displayed in a separate row with the following column headers:

   - **Consistent:** indicates whether the reported mean passed the GRIM test (`Yes` or `No`).

- **Reported Mean:** the original mean value as extracted from the text, including trailing zeros.
- **Sample Size:** the corresponding sample size.
- **Decimals:** the number of decimal places in the reported mean.
- **Reasoning:** the AI-generated explanation for why the reported mean was considered GRIM-applicable.

Using the approach, it should now be possible to automatically perform the GRIM and get an immediate overview of the results.

10 Experiment

10.1 Data Used

To evaluate the efficacy of the AI-powered GRIM Python script, a subset of the articles analysed by Brown & Heathers (2016) [8] was selected. Specifically, all articles containing GRIM-testable data from the journal *Psychological Science* ($N = 30$) were included, along with an additional 10 articles without GRIM-testable data, to verify if the model would return no values in such cases. A list of exactly which articles were analysed can be found in **Appendix** D.

Brown & Heathers were contacted with a request to share the GRIM-applicable means they had found in their analysis. However, their analysis results were not logged in a structured manner. Fortunately, they did annotate the original articles and highlighted (most) inconsistent GRIM-applicable mean values. These annotated articles served as the basis for this analysis, though they are not without their flaws. As Brown noted in an e-mail response: "There will surely be some that we missed, and probably a couple of false positives due to exhaustion or the use of multi-item measures."

10.2 Procedure

The `testers/GRIM/main.py` script was run on all 40 articles in dataset. The full code for this script can be found on the aforementioned GitHub page. The following categories were logged:

- **Author:** Author(s) of the article.
- **Title** The title of the article.
- **# Inconsistent mean values found by script:** The number of inconsistent *and* GRIM-applicable mean values, according to the script.
- **# Inconsistent mean values annotated by Brown & Heathers:** The number of mean values explicitly annotated as inconsistent by Brown & Heathers. Although their annotations include various notes (e.g., indicating that a mean was rounded up or down), only those clearly marked as inconsistent are counted in this category. All other comments are excluded from the analysis.
- **# Mean values in intersection:** The number of inconsistent mean values detected by the script *and* explicitly annotated as inconsistent by Brown & Heathers. This is considered the *most informative metric*, given the absence of a structured file containing all GRIM-applicable values.
- **Notes:** Additional information regarding why certain mean values were not found or any other findings worth mentioning.

Note that, unlike the statcheck analysis, this GRIM evaluation does not include categories such as true positives, false positives, true negatives, or false negatives. This is because the annotations do not serve as a definitive ground truth and the interpretation can be somewhat challenging, as is also confirmed by the e-mail response quoted earlier. Furthermore, the GRIM analysis is only run for one iteration, unlike the statcheck analysis, which was run in three iterations. This is because the output of the GRIM script is not deterministic, and repeated runs do not yield more consistent or meaningful insights.

The detailed results of this analysis can be found in **Appendix** D. The full raw output of the script is available on the OSF page of this project (see the Project Links section for the link).

11 Results

Table 9 shows the results of applying the AI-powered GRIM script to the dataset.

Table 9: Overview of GRIM-applicable articles and inconsistencies detected by the AI-powered script compared to annotations by Brown & Heathers

Category	**# Inconsistent mean values found by script**	**# Inconsistent mean values annotated by Brown & Heathers**	**# Mean values in intersection**
Category 1: Article contains GRIM-applicable means and at least one inconsistency was annotated by Brown & Heathers (N = 14)	46	61	20
Category 2: Article contains GRIM-applicable means but no inconsistencies were annotated by Brown & Heathers (N = 16)	58	0	0
Category 3: Article does not contain any GRIM-applicable means (N = 10)	13	0	0
TOTAL (N = 40)	**117**	**61**	**20**

When analysing only articles with GRIM-applicable data (i.e., Category 1 & 2), the script marks 104 mean values as inconsistent, of which only 20 (**19.2%**) are confirmed as inconsistent by Brown & Heathers. When adding articles that do not contain any GRIM-applicable data (i.e., Category 3), the number of inconsistent means marked by the script increase to 117, which causes the true positive rate to drop to **17.1%**.

In this analysis, Category 3 includes only 10 articles, whereas in the full dataset, it consists of 70 articles. If the full set were analysed, it is likely that the remaining 60 articles would cause the script to (falsely) flag additional inconsistent means, which would cause the overall true positive rate to further decrease. This shows that, in order to get *at least some* use out of the script results, it is important to know beforehand if your article contains GRIM-applicable data, otherwise the true positive rate will be so low that you will be overwhelmed by false positives.

Table 10, below, provides further insights into the specific categories of missed mean values by the script.

Table 10: Categories of missed mean values by the script

Category of missed mean value	Count	Percentage of total missed
Mean values in tables	22	53.7%
Regular misses	8	19.5%
Wrong N extracted	7	17.1%
Mean values reported as percentages	4	9.7%
Total missed results	**41**	**100%**

This table shows that roughly half (53.7%) of missed mean values occurred because the relevant mean values were presented in tables, which the script was unable to extract. This is due to how the table was inserted into the original PDF and how the current context window approach works. The article *Constructing Rich False Memories of Committing Crime* by J. Shaw [36] shows that the script *can* detect mean values presented in tables, as long as the tables are formatted in a way that allows the PDF-to-text conversion library (PyMuPDF) to extract them properly (refer to **Appendix D**). Furthermore, even when the script successfully converts a table to plain text, it still fails to identify the correct sample size if this information is not (explicitly) mentioned within the same context window. As a result, the script is unable to match the extracted mean value to its corresponding sample size.

The category `Wrong N extracted` shows that the script has a hard time matching the right sample size to its corresponding mean value, especially in clustered contexts where many means and sample sizes appear close together.

Finally, the script has not been programmed to interpret percentages as mean values.

11.1 Runtime and Cost

The total runtime for processing 40 articles is 1,124.63 seconds, resulting in an average runtime of **28.12 seconds per article**. The total cost for this analysis amounts to $2.03, which results in **$0.0508 per article** for one iteration.

12 Discussion

The results show that, even when articles are pre-filtered for containing GRIM-applicable mean values, the script still performs poorly, with a true positive rate of only about 1 in 5 (19.2%). This shows that the script often flags results as inconsistent when they are not. This is mostly because the script thinks a certain mean value is GRIM-applicable, despite that mean *not* being composed of integer data, but rather floating-point data.

Given the limited accuracy of the script, it is worth questioning whether any meaningful time is actually saved by using it. Because of the high number of false positives, a lot of manual checking is still needed, which reduces the time-saving benefits of using the script in the first place.

12.1 Limitations

The biggest limitation of the current script is the one described just above: the AI model's (in)ability to determine whether a reported mean is derived from integer data only. The GRIM test can only be applied when the underlying data are integer-based (e.g., Likert scales), but the model frequently misclassifies floating-point data as GRIM-applicable. As a result, the script attempts to apply the GRIM test to these mean values, resulting in many false positives.

Another significant limitation of the script lies in its difficulty matching a mean value to its corresponding sample size. In many cases, these two values are spread out across several sentences or even paragraphs, which makes it difficult for the AI model to properly link these values together. In this study, the decision was made to use a segmentation-based approach, since processing an entire article at once almost always results in a context window that is too large. As a result, the script is unable to process all relevant mean values effectively, causing many results to be missed. However, using a segmentation-based approach includes its own caveat: the sample size and mean value are often not reported within the same (relatively small) segment. As a result, the model is unable to match them correctly, leading to missed or incorrect GRIM evaluations.

13 Conclusion

This study has shown that it is indeed possible to automate the GRIM test using AI-powered data extraction, but the performance is poor. This is due to the limitations mentioned in subsection 12.1. Despite efforts to improve the script's performance – such as using `GPT-4o` instead of `GPT-4o-mini`, using a segmentation-based approach to manage context, implementing a final reasoning step and applying a GRIM-applicability formula to exclude entries with overly large sample sizes – it still achieved a true positive rate of only 19.2%. This was the case even when the test was limited to only articles containing GRIM-applicable mean values. When the script is tested on a larger subset, which also includes articles without GRIM-applicable data, the true positive rate of the script decreases even further.

This script could serve as a quick and automated scan of an article, but users must be wary of its limitations. If you use this script as an author and want to check your own paper, you could run the tool and focus only on the mean values that you know are composed of integer values. The script, in its current state, is not yet accurate enough to serve as a standalone tool.

13.1 Future Research

Although the performance of the current implementation is quite limited, it provides a solid foundation for future improvements. We recommend that future research begins by exploring the performance of more advanced reasoning models, such as OpenAI's `o3` and `o4-mini` models. These models are "LLMs trained with reinforcement learning to perform reasoning. Reasoning models think before they answer, producing a long internal chain of thought before responding to the user," according to OpenAI [31].

Future research could also involve more general ways to attempt to improve the script's reasoning capabilities, such as altering the prompt and adding additional examples.

To address the limitation where mean values and sample sizes do not appear within the same context window, future work could explore the use of a memory structure. For example, such a structure could log mean values and sample sizes, along with a one-line description so the script knows what the values refer to. This memory structure could then be passed to the model alongside the next context window, potentially allowing for cross-segment value linking.

Future research could involve developing a way to detect the use of multi-item measurements. This could potentially reduce the number of false positives. As Brown himself stated in an e-mail reply: "You have to detect the multi-item measures efficiently or you will be generating lots of false positives with GRIM."

Furthermore, future research could explore whether there are any options for altering the settings of the current PDF-to-text conversion library (PyMuPDF) to allow for the extraction of all types of tables, or consider using a different library that supports more forms of table detection.

Finally, it could be investigated whether enabling the model to also check percentages adds meaningful value. However, researchers should be cautious, as this could potentially increase the number of false positives significantly. This effect potentially outweighs the benefits of detecting a few additional true positives.

All in all, these suggestions show that there is still room to improve and expand upon the current approach.

Part IV

General Conclusion

This thesis explored whether AI-powered data extraction can improve reproducibility tools in scientific literature, focusing on statcheck and the GRIM test.

The AI-powered statcheck script has shown that when all relevant parameters (e.g., `test_type`, `test_value`, `p_value`) are located close together in the text, AI-powered data extraction performs very well. The AI-powered statcheck script has achieved a significantly higher detection percentage than the original statcheck tool, while maintaining very high overall accuracy. Using AI-powered data extraction, results no longer need to be in strict APA format in order to be automatically extracted and recalculated.

In contrast, the AI-powered GRIM script performed poorly. GRIM-relevant parameters, such as `reported_mean`, `sample_size`, and whether the value is based on `integer_data`, are often distributed across different sentences or even different sections. The AI model struggled to identify and link these elements correctly, especially due to the segmentation approach used. This has led to many false positives and missed detections.

In summary, AI-powered data extraction is effective when key information appears in close context, but becomes significantly more challenging when the relevant information is spread out.

References

[1] Douglas G. Altman and J. Martin Bland. Diagnostic tests 1: sensitivity and specificity. *British Medical Journal*, 308:1552, 1994. doi: 10.1136/bmj.308.6943.1552.

[2] Inc. Amazon Web Services. What is gpt ai? - generative pre-trained transformers explained, n.d. URL https://aws.amazon.com/what-is/gpt/. Retrieved February 5, 2025.

[3] American Psychological Association. *Publication Manual of the American Psychological Association*. American Psychological Association, Washington, DC, 6th edition, 2010.

[4] M. Bakker and J. M. Wicherts. The (mis)reporting of statistical results in psychology journals. *Behavior Research Methods*, 43(3):666–678, 2011. doi: 10.3758/s13428-011-0089-5.

[5] M. Bakker and J. M. Wicherts. Outlier removal and the relation with reporting errors and quality of psychological research. *PLoS ONE*, 9(7):e103360, 2014. doi: 10.1371/journal.pone.0103360.

[6] Alireza Baratloo, Maryam Hosseini, Ahmed Negida, and Gamal El Ashal. Part 1: Simple definition and calculation of accuracy, sensitivity and specificity. *Emergency*, 3(2):48–49, 2015.

[7] D. Berle and V. Starcevic. Inconsistencies between reported test statistics and p-values in two psychiatry journals. *International Journal of Methods in Psychiatric Research*, 16(4):202–207, 2007. doi: 10.1002/mpr.225.

[8] N. J. L. Brown and J. Heathers. The grim test: A simple technique detects numerous anomalies in the reporting of results in psychology. *Social Psychological and Personality Science*, 8(4): 363–369, 2016. doi: 10.1177/1948550616673876.

[9] J. M. Caperos and A. Pardo. Consistency errors in p-values reported in spanish psychology journals. *Psicothema*, 25(3):408–414, 2013.

[10] Aline Claesen, Wolf Vanpaemel, Anne-Sofie Maerten, Thomas Verliefde, Francis Tuerlinckx, and Tom Heyman. Data sharing upon request and statistical consistency errors in psychology: A replication of wicherts, bakker and molenaar (2011). *PLOS ONE*, 18(4):e0284243, 2023. doi: 10.1371/journal.pone.0284243. URL https://doi.org/10.1371/journal.pone.0284243.

[11] CWRU Online Engineering. Advancements in artificial intelligence and machine learning, March 2024. URL https://online-engineering.case.edu/blog/advancements-in-artificial-intelligence-and-machine-learning. Retrieved February 5, 2025.

[12] Sacha Epskamp and Michèle B. Nuijten. statcheck: Extract statistics from articles and recompute p values, 2015. URL http://CRAN.R-project.org/package=statcheck. R package version 1.0.1.

[13] FORRT - Framework for Open and Reproducible Research Training. Questionable research practices or questionable reporting practices (qrps), November 2021. URL https://forrt.org/glossary/vbeta/questionable-research-practices-or-/. Retrieved December 13, 2024.

[14] E. Garcia-Berthou and C. Alcaraz. Incongruence between test statistics and p values in medical papers. *BMC Medical Research Methodology*, 4:13, 2004. doi: 10.1186/1471-2288-4-13.

[15] T. Hartonen and M.J. Alava. How important tasks are performed: peer review. *Scientific Reports*, 3:1679, 2013. doi: 10.1038/srep01679.

[16] Leslie K. John, George Loewenstein, and Drazen Prelec. Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science*, 23(5):524–532, 2012. doi: 10.1177/0956797611430953. Retrieved December 13, 2024.

[17] Tsuyoshi Miyakawa. No raw data, no science: another possible source of the reproducibility crisis. *Molecular Brain*, 13(1), 2020. doi: 10.1186/s13041-020-0552-2. URL https://doi.org/10.1186/s13041-020-0552-2.

[18] National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. National Academies Press, Washington, DC, 2019. doi: 10.17226/25303. URL https://doi.org/10.17226/25303.

[19] M. I. Norton, J. A. Vandello, and J. M. Darley. Casuistry and social category bias. *Journal of Personality and Social Psychology*, 87(6):817–831, 2004. doi: 10.1037/0022-3514.87.6.817.

[20] Brian A. Nosek, Tom E. Hardwicke, Hannah Moshontz, Aline Allard, Katherine S. Corker, Anna Dreber, Fiona Fidler, Joseph Hilgard, Melissa K. Struhl, Michèle B. Nuijten, Julia M. Rohrer, Felipe Romero, Anne M. Scheel, Laura D. Scherer, Felix D. Schönbrodt, and Simine Vazire. Replicability, robustness, and reproducibility in psychological science. *Annual Review of Psychology*, 73(1):719–748, 2021. doi: 10.1146/annurev-psych-020821-114157. URL https://doi.org/10.1146/annurev-psych-020821-114157.

[21] M. B. Nuijten, C. H. J. Hartgerink, M. a. L. M. Van Assen, S. Epskamp, and J. M. Wicherts. The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior Research Methods*, 48(4):1205–1226, 2016. doi: 10.3758/s13428-015-0664-2.

[22] Michele Nuijten. Github - michelenuijten/statcheck: A spellchecker for statistics, n.d. URL https://github.com/MicheleNuijten/statcheck. Retrieved October 16, 2024.

[23] Michèle B Nuijten, Marcel ALM van Assen, Chris HJ Hartgerink, Sacha Epskamp, and Jelte M Wicherts. The validity of the tool "statcheck" in discovering statistical reporting inconsistencies. November 2017. doi: 10.31234/osf.io/tcxaj. URL https://doi.org/10.31234/osf.io/tcxaj.

[24] Michèle Nuijten. Rpubs - manual statcheck 1.3.0, 2018. URL https://rpubs.com/michelenuijten/Statcheckmanual. Retrieved October 21, 2024.

[25] Michèle B. Nuijten and Sacha Epskamp. statcheck: Extract statistics from articles and recompute p-values, 2024. R package version 1.5.0. Web implementation at https://statcheck.io.

[26] OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Retrieved April 7, 2025.

[27] OpenAI. Openai models documentation, 2024. URL https://platform.openai.com/docs/models. Retrieved February 5, 2025.

[28] OpenAI. How your data is used to improve model performance, September 2024. URL https://help.openai.com/en/articles/6783458-how-your-data-is-used-to-improve-model-performance. Retrieved December 4, 2024.

[29] OpenAI. Key concepts: Understanding the openai api, December 2024. URL https://platform.openai.com/docs/concepts/tokens. Retrieved December 4, 2024.

[30] OpenAI. Api pricing, 2025. URL https://openai.com/api/pricing/. Retrieved April 18, 2025.

[31] OpenAI. Reasoning guide — openai platform documentation, 2025. URL https://platform.openai.com/docs/guides/reasoning?api-mode=responses. Retrieved April 18, 2025.

[32] OpenAI. Data extraction and transformation in elt workflows using gpt-4o as an ocr alternative, n.d. URL https://cookbook.openai.com/examples/data_extraction_transformation. Retrieved February 5, 2025.

[33] C. Pernet. Null hypothesis significance testing: a short tutorial. *F1000Research*, 4:621, August 2015. doi: 10.12688/f1000research.6963.3.

[34] Hans E. Plesser. Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11, 2018. doi: 10.3389/fninf.2017.00076. URL https://doi.org/10.3389/fninf.2017.00076.

[35] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, Upper Saddle River, NJ, 3rd edition, 2016. ISBN 978-0136042594.

[36] Julia Shaw and Stephen Porter. Constructing rich false memories of committing crime. *Psychological Science*, 26(3):291–301, 2015. doi: 10.1177/0956797614562862.

[37] Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. False-positive psychology. *Psychological Science*, 22(11):1359–1366, 2011. doi: 10.1177/0956797611417632. URL https://doi.org/10.1177/0956797611417632.

[38] C. L. S. Veldkamp, M. B. Nuijten, L. Dominguez-Alvarez, M. A. L. M. Van Assen, and J. M. Wicherts. Statistical reporting errors and collaboration on statistical analyses in psychological science. *PLoS ONE*, 9(6):e98900, 2014. doi: 10.1371/journal.pone.0098900.

[39] P. Verhaeghen, J. Cerella, and C. Basak. A working memory workout: how to expand the focus of serial attention from one to four items in 10 hours or less. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(6):1322–1337, 2004. doi: 10.1037/0278-7393.30.6.1322.

[40] J. M. Wicherts, M. Bakker, and D. Molenaar. Willingness to share research data is related to the strength of the evidence and the quality of reporting of statistical results. *PLoS ONE*, 6 (11):e26828, 2011. doi: 10.1371/journal.pone.0026828.

A Statcheck Script Analysis Data

Appendix A shows the analysis result of running the AI-powered Python statcheck script and comparing the results to the manually coded validate file, which can be found at https://osf.io/753qd/.

Table 11: Full analysis of articles and their NHST results

Article	**# Sig. results validate file**	**# Correctly iden. by script**	**# Missed in validate data**	**# Missed by script**	**TP**	**FP**	**TN**	**FN**	**Note**
JPSP Ames 2004	15	15	0	0	1	0	14	0	-
JEP Beaman 2004	16	16	0	0	2	1	13	0	1 FP due to Huynh-Feldt correction
JPSP Blair	29	29	0	0	1	0	28	0	-
JEP Carlson 2004	23	23	0	0	1	0	22	0	-
JEP Creel 2004	10	10	0	0	0	0	10	0	-
JEP Delaney 2004	39	37	0	2	7	0	30	0	2 results missed because they are the same test consecutively
JPSP Dijksterhuis	21	20	0	1	4	0	16	0	1 result missed because they are the same test consecutively
JEP Domangue 2004	25	24	0	1	1	1	22	0	1 result missed because they are the same test consecutively; 1 FP due to wrong operator extraction
JPSP Eagly	-	-	-	-	-	-	-	-	-
JPSP Eberhardt	27	27	0	0	1	0	26	0	-
JEP Estes 2004	9	9	0	0	0	0	9	0	-
JPSP Exline	8	8	0	0	0	0	8	0	-
JPSP Feeney	-	-	-	-	-	-	-	-	-

Article	**# Sig. results validate file**	**# Correctly iden. by script**	**# Missed in validate data**	**# Missed by script**	**TP**	**FP**	**TN**	**FN**	**Note**
JPSP Ferguson	33	33	0	0	1	7	25	0	7 FP p = .000, 1 TP actual inconsistency
JEP Folstein 2004	37	33	0	4	0	2	31	0	4 truly missed by script; 2 FP because of Huynh-Feldt
JEP Glanzer	32	31	0	1	1	1	29	0	1 FP due to wrong operator extraction; 1 truly missed by script
JPSP Golec	-	-	-	-	-	-	-	-	-
JEP Heit 2004	22	22	0	0	3	0	19	0	-
JPSP Hewig	9	9	0	0	0	7	2	0	7 FP due to Huynh-Feldt correction
JEP Hohlfeld 2004	-	-	-	-	-	-	-	-	-
JEP Jahn 2004	13	13	0	0	0	0	13	0	-
JPSP Johnson 2004	27	27	0	0	3	0	24	0	-
JPSP Koole 2004	30	30	1	1	2	0	28	0	The validate file fabricated 1 result (which was missed by the script) and it missed one TP gross inconsistency
JPSP Lord 2004	24	24	0	0	1	0	22	1	1 FN: "f(1, 57) = 4.0 < 0.05, 0.04890 to 0.05168"
JEP McKenzie 2004	7	7	0	0	0	1	6	0	1 FP due to wrong operator extraction
JPSP Meiser	24	24	0	0	0	0	24	0	-
JPSP Mikulincer	-	-	-	-	-	-	-	-	-
JEP Moscoso del Prado Martín 2004	-	-	-	-	-	-	-	-	-

Article	**# Sig. results validate file**	**# Correctly iden. by script**	**# Missed in validate data**	**# Missed by script**	**TP**	**FP**	**TN**	**FN**	**Note**
JPSP Mussweiler	8	8	0	0	4	1	3	0	1 FP (gross inconsistency) due to wrong tail detection (should be 1, 2 was used)
JEP Norris 2004	21	21	0	0	1	0	20	0	-
JPSP Norton	20	19	0	1	0	0	19	0	1 result missed because they are the same test consecutively
JEP Pexman	46	41	0	5	0	0	41	0	5 results truly missed
JEP Rapp 2004	33	33	0	0	3	0	30	0	-
JEP Rayner	35	32	0	3	0	0	32	0	3 truly missed by script
JEP Rinck 2004	11	11	0	0	0	0	11	0	-
JPSP Tamir	7	7	0	0	0	0	7	0	-
JPSP Tazelaar	-	-	-	-	-	-	-	-	-
JPSP Thrash	-	-	-	-	-	-	-	-	-
JEP Tillmann 2004	21	22	1	0	0	0	22	0	1 result truly missed by validate file
JEP Unsworth	30	30	0	0	0	0	30	0	-
JEP Van Zandt 2004	8	0	0	8	0	0	0	0	Script found no results, but there are results to be found because no p-value was reported for each test
JPSP Van Zomeren	16	19	3	0	4	0	15	0	3 significant chi2 results missed by validate file
JEP Verhaeghen	37	0	0	37	0	0	0	0	Script found no results, but there are results to be found because no p-value was reported for each test
JPSP Visser 2004	9	9	0	0	0	0	9	0	-

Article	# Sig. results validate file	# Correctly iden. by script	# Missed in validate data	# Missed by script	TP	FP	TN	FN	Note
JEP Ward 2004	37	37	0	0	1	0	36	0	-
JEP Winman 2004	8	8	0	0	0	0	8	0	-
JEP Yang	19	19	0	0	0	0	19	0	-
JPSP Jones 2004	23	23	0	0	2	0	21	0	-
TOTAL	**869**	**810**	**5**	**64**	**44**	**21**	**744**	**1**	-

B Filtered Raw Output

Appendix B shows the raw output of each article that is included in the final dataset used for evaluation. These results have been filtered to comply with the inclusion criteria mentioned in both subsection 2.1.3 and 4.1. Each paper has automatically been analysed three times, the most frequent output is shown below. The complete, unfiltered output can be downloaded as a `.csv` file on the OSF page: https://osf.io/ae2pu/files/osfstorage.

```
(1/48)
JPSP_Ames_2004_87_5_573_Strategies for social.htm
1       Yes          t(26) = 2.47       = 0.02      0.02016 to 0.02062
3       Yes          t(36) = 2.09       < 0.05      0.04327 to 0.04422
4       Yes          t(36) = 2.32       < 0.05      0.02583 to 0.02643
7       Yes       f(1, 78) = 7.15       < 0.01      0.00910 to 0.00915
8       Yes           t(78) = 2.1       = 0.04      0.03465 to 0.04372
11      Yes          t(48) = 3.71      < 0.001      0.00053 to 0.00055
12       No     f(1, 49) = 12.21      < 0.001      0.00102 to 0.00102
Recalculated p-value does not match the reported p-value.
16      Yes          t(27) = 3.33       < 0.01      0.00249 to 0.00255
17      Yes         t(28) = 10.08      < 0.001      0.00000 to 0.00000
18      Yes          t(20) = 5.04      < 0.001      0.00006 to 0.00006
19      Yes          t(20) = 3.58       < 0.01      0.00185 to 0.00190
20      Yes         t(28) = 13.02      < 0.001      0.00000 to 0.00000
21      Yes       f(1, 45) = 8.33       < 0.01      0.00596 to 0.00598
22      Yes          t(45) = 2.78       < 0.01      0.00780 to 0.00801
24      Yes       f(1, 46) = 6.11       = 0.02      0.01716 to 0.01725

(2/48)
JEP_Beaman_2004_30_5_1106_The irrelevant sound.htm
0       Yes          t(37) = -4.93      < 0.001      0.00002 to 0.00002
3       Yes       f(1, 36) = 86.41      < 0.001      0.00000 to 0.00000
4        No        f(2, 72) = 6.75      = 0.004      0.00205 to 0.00207
Recalculated p-value does not match the reported p-value.
6       Yes       f(2, 72) = 176.3      < 0.001      0.00000 to 0.00000
7       Yes        f(2, 72) = 4.86       = 0.01      0.01043 to 0.01052
8        No     f(1, 36) = 161.63      = 0.048      0.00000 to 0.00000
Recalculated p-value does not match the reported p-value.
9       Yes       f(1, 18) = 40.26      < 0.001      0.00001 to 0.00001
10      Yes       f(1, 38) = 86.72      < 0.001      0.00000 to 0.00000
11      Yes       f(1, 38) = 18.79      < 0.001      0.00010 to 0.00010
13      Yes           t(38) = 3.05      < 0.008      0.00410 to 0.00421
14      Yes       f(2, 70) = 64.41      < 0.001      0.00000 to 0.00000
16      Yes           t(36) = 7.84      < 0.002      0.00000 to 0.00000
17      Yes           t(36) = 3.41      < 0.004      0.00159 to 0.00164
18      Yes       f(2, 70) = 34.97      < 0.001      0.00000 to 0.00000
```

```
20     Yes       f(2, 70) = 4.48       < 0.02      0.01470 to 0.01483
21      No          t(35) = 1.97      < 0.028      0.05620 to 0.05739
Gross inconsistency: reported p-value and recalculated p-value differ in significance.

(3/48)
JPSP_Blair_2004_87_6_763_The automaticity of.htm
1      Yes      f(1, 60) = 90.37      < 0.001      0.00000 to 0.00000
2      Yes       f(1, 60) = 8.66       < 0.01      0.00461 to 0.00463
3      Yes      f(1, 60) = 28.91      < 0.001      0.00000 to 0.00000
4      Yes       f(1, 60) = 8.35       < 0.01      0.00535 to 0.00537
5      Yes       f(1, 28) = 64.9      < 0.001      0.00000 to 0.00000
6      Yes      f(1, 32) = 25.94      < 0.001      0.00002 to 0.00002
7      Yes      f(1, 60) = 71.95      < 0.001      0.00000 to 0.00000
8      Yes        f(1, 60) = 7.1       < 0.01      0.00964 to 0.01014
10     Yes     f(1, 69) = 109.49      < 0.001      0.00000 to 0.00000
11     Yes       f(1, 69) = 5.98      < 0.025      0.01698 to 0.01707
12     Yes      f(1, 69) = 30.41      < 0.001      0.00000 to 0.00000
14     Yes       f(1, 69) = 4.62       < 0.05      0.03501 to 0.03521
15     Yes       f(1, 69) = 70.0      < 0.001      0.00000 to 0.00000
16     Yes       f(1, 69) = 9.59       < 0.01      0.00282 to 0.00283
17     Yes      f(1, 95) = 72.02      < 0.001      0.00000 to 0.00000
18     Yes      f(1, 95) = 28.32      < 0.001      0.00000 to 0.00000
20     Yes       f(2, 95) = 9.07      < 0.001      0.00025 to 0.00025
21     Yes      f(1, 95) = 18.11      < 0.001      0.00005 to 0.00005
23     Yes       f(1, 95) = 5.41      < 0.025      0.02209 to 0.02221
24     Yes      f(1, 95) = 92.28      < 0.001      0.00000 to 0.00000
25     Yes       f(1, 95) = 8.66       < 0.01      0.00408 to 0.00410
26     Yes       f(1, 95) = 9.45       < 0.01      0.00275 to 0.00276
29     Yes      f(1, 69) = 38.37      < 0.001      0.00000 to 0.00000
30      No       f(1, 69) = 4.72       < 0.03      0.03316 to 0.03334
Recalculated p-value does not match the reported p-value.
31     Yes     f(1, 69) = 163.32      < 0.001      0.00000 to 0.00000
32     Yes      f(1, 69) = 14.28      < 0.001      0.00033 to 0.00033
33     Yes      f(1, 69) = 32.25      < 0.001      0.00000 to 0.00000
34     Yes       f(1, 69) = 4.04       < 0.05      0.04821 to 0.04848
35     Yes          t(67) = 2.23       < 0.05      0.02876 to 0.02945

(4/48)
JEP_Carlson_2004_30_6_1235_Intentional control of.htm
0      Yes      f(2, 58) = 21.89       < 0.01      0.00000 to 0.00000
1      Yes       f(2, 58) = 27.0       < 0.01      0.00000 to 0.00000
2      Yes     f(2, 58) = 535.75       < 0.01      0.00000 to 0.00000
7      Yes          t(33) = 6.14       < 0.01      0.00000 to 0.00000
8      Yes          t(33) = 3.05       < 0.01      0.00443 to 0.00455
9      Yes          t(33) = 5.55       < 0.01      0.00000 to 0.00000
```

```
10      Yes          t(22) = 5.14       < 0.01      0.00004 to 0.00004
11      Yes          t(27) = 4.39       < 0.01      0.00015 to 0.00016
13      Yes          t(24) = 3.45       < 0.01      0.00206 to 0.00211
14      Yes          t(24) = 2.89       < 0.01      0.00795 to 0.00814
15       No          t(15) = 2.75       < 0.01      0.01474 to 0.01504
Recalculated p-value does not match the reported p-value.
Consistent for one-tailed, inconsistent for two-tailed
18      Yes      f(2, 58) = 5.813      = 0.005      0.00500 to 0.00500
19      Yes     f(4, 116) = 5.286      = 0.001      0.00060 to 0.00060
20      Yes       f(2, 46) = 7.21       < 0.01      0.00188 to 0.00190
24      Yes       f(2, 58) = 7.89       < 0.01      0.00093 to 0.00094
27      Yes      f(2, 56) = 35.86       < 0.01      0.00000 to 0.00000
28      Yes      f(2, 56) = 30.83       < 0.01      0.00000 to 0.00000
33      Yes        f(1, 29) = 4.2       < 0.05      0.04831 to 0.05085
34      Yes      f(1, 29) = 12.01       < 0.01      0.00167 to 0.00167
35      Yes        f(1, 29) = 7.2       < 0.05      0.01165 to 0.01218
36      Yes       f(1, 29) = 13.5       < 0.01      0.00094 to 0.00098
37      Yes      f(1, 29) = 10.64       < 0.01      0.00283 to 0.00284
38      Yes       f(1, 29) = 7.65       < 0.01      0.00975 to 0.00980

(5/48)
JEP_Creel_2004_30_5_1119_Distant Melodies. Statistical.htm
0       Yes      f(1, 10) = 10.0       = 0.01      0.00998 to 0.01026    -
3       Yes         t(11) = 4.46      = 0.001      0.00095 to 0.00097    -
5       Yes     f(1, 10) = 16.71      = 0.002      0.00218 to 0.00219    -
7       Yes         t(11) = 4.47      = 0.001      0.00094 to 0.00095    -
8       Yes     f(1, 22) = 22.73     < 0.0001      0.00009 to 0.00009    -
9       Yes      f(1, 10) = 5.24      = 0.045      0.04500 to 0.04517    -
10      Yes      f(1, 10) = 9.58       = 0.01      0.01133 to 0.01136    -
11      Yes         t(11) = 5.07     = 0.0004      0.00036 to 0.00036    -
16      Yes         t(38) = 6.03     < 0.0001      0.00000 to 0.00000    -
17      Yes         t(38) = 4.14     < 0.0002      0.00018 to 0.00019    -

(6/48)
JEP_Delaney_2004_30_6_1219_Immediate and sustained.htm
0       Yes      f(3, 114) = 3.65       < 0.05      0.01466 to 0.01485
1       Yes       f(1, 38) = 10.5       < 0.01      0.00243 to 0.00254
2       Yes       f(1, 38) = 5.01       < 0.05      0.03106 to 0.03122
3       Yes       f(1, 38) = 10.5       < 0.01      0.00243 to 0.00254
4       Yes      f(1, 38) = 42.03      < 0.001      0.00000 to 0.00000
6        No       f(2, 76) = 4.86       < 0.01      0.01028 to 0.01037
Recalculated p-value does not match the reported p-value.
7       Yes        f(1, 18) = 9.8       < 0.01      0.00568 to 0.00588
8       Yes        f(2, 36) = 3.5       < 0.05      0.03916 to 0.04258
10      Yes      f(1, 18) = 31.96      < 0.001      0.00002 to 0.00002
```

```
13      No          t(18) = 1.99       < 0.05      0.06141 to 0.06260
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
Consistent for one-tailed, inconsistent for two-tailed
14     Yes       f(1, 21) = 4.62       < 0.05      0.04331 to 0.04351
17     Yes       f(3, 60) = 4.33       < 0.01      0.00785 to 0.00794
18     Yes       f(1, 20) = 4.39       < 0.05      0.04897 to 0.04920
19     Yes       f(1, 20) = 6.32       < 0.05      0.02057 to 0.02065
20     Yes      f(1, 20) = 55.47      < 0.001      0.00000 to 0.00000
21     Yes       f(1, 20) = 7.71       < 0.05      0.01162 to 0.01166
22     Yes       f(1, 60) = 5.46       < 0.05      0.02275 to 0.02287
23     Yes       f(3, 60) = 6.51      < 0.001      0.00069 to 0.00070
24     Yes      f(1, 22) = 71.95      < 0.001      0.00000 to 0.00000
25     Yes        f(1, 5) = 8.37       < 0.05      0.03403 to 0.03410
26     Yes          t(43) = 8.85      < 0.001      0.00000 to 0.00000
27     Yes     f(1, 28) = 104.15      < 0.001      0.00000 to 0.00000
29     Yes      f(1, 24) = 34.35      < 0.001      0.00000 to 0.00000
30      No       f(1, 24) = 7.52       < 0.01      0.01132 to 0.01137
Recalculated p-value does not match the reported p-value.
31      No        f(1, 24) = 5.9       < 0.01      0.02248 to 0.02353
Recalculated p-value does not match the reported p-value.
32      No       f(1, 24) = 7.76       < 0.01      0.01024 to 0.01029
Recalculated p-value does not match the reported p-value.
33     Yes       f(1, 28) = 7.78       < 0.01      0.00937 to 0.00942
34     Yes      f(1, 28) = 16.55      < 0.001      0.00035 to 0.00035
35     Yes        f(1, 28) = 6.4       < 0.05      0.01693 to 0.01772
36     Yes          t(14) = 5.56      < 0.001      0.00007 to 0.00007
38     Yes       f(1, 24) = 5.24       < 0.05      0.03110 to 0.03124
39     Yes      f(1, 24) = 20.49      < 0.001      0.00014 to 0.00014
40     Yes       f(1, 24) = 4.34       < 0.05      0.04793 to 0.04817
41      No       f(1, 28) = 4.55       < 0.01      0.04172 to 0.04193
Recalculated p-value does not match the reported p-value.
42     Yes       f(1, 28) = 4.74       < 0.05      0.03796 to 0.03815
43     Yes       f(1, 24) = 6.51       < 0.05      0.01748 to 0.01756
44      No       f(2, 35) = 8.43      < 0.001      0.00102 to 0.00103
Recalculated p-value does not match the reported p-value.

(7/48)
JPSP_Dijksterhuis_2004_87_5_586_Think different. The.htm
   Consistent   APA Reporting    Reported P-value Valid P-value Range
0      Yes          t(21) = 2.75       < 0.02      0.01187 to 0.01213
1      Yes        f(2, 54) = 3.4       < 0.05      0.03890 to 0.04251
3      Yes       f(1, 37) = 4.96       < 0.04      0.03203 to 0.03219
6       No       chi2(59) = 3.13       < 0.04      1.00000 to 1.00000
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
7       No       chi2(60) = 6.69       < 0.01      1.00000 to 1.00000
```

```
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
8      Yes      f(2, 133) = 4.69       < 0.02      0.01071 to 0.01081
10     Yes       f(1, 88) = 8.07       < 0.01      0.00558 to 0.00560
11     Yes        f(1, 84) = 4.6       < 0.04      0.03391 to 0.03584
12     Yes       f(1, 84) = 4.03       < 0.05      0.04778 to 0.04805
13     Yes       f(1, 94) = 6.42       < 0.02      0.01290 to 0.01297
17     Yes      f(2, 111) = 5.29       < 0.01      0.00636 to 0.00642
18      No      f(2, 111) = 2.91       < 0.03      0.05837 to 0.05892
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
19     Yes     f(1, 111) = 63.37     < 0.0001      0.00000 to 0.00000
21      No       f(2, 111) = 3.1       < 0.02      0.04671 to 0.05135
Recalculated p-value does not match the reported p-value.
22     Yes     f(1, 111) = 13.31      < 0.001      0.00040 to 0.00040
23     Yes      f(2, 111) = 4.19       < 0.02      0.01753 to 0.01769
25     Yes       f(1, 63) = 6.56       < 0.02      0.01280 to 0.01287
26     Yes       f(2, 63) = 4.32       < 0.02      0.01737 to 0.01753
27     Yes        f(1, 43) = 5.5       < 0.03      0.02311 to 0.02431
28     Yes       f(1, 42) = 9.58      < 0.005      0.00349 to 0.00350

(8/48)
JEP_Domangue_2004_30_5_1002_Effects of model-based.htm
   Consistent    APA Reporting    Reported P-value Valid P-value Range
0      Yes     f(2, 176) = 262.82      < 0.001      0.00000 to 0.00000
3      Yes        f(1, 90) = 4.63      = 0.034      0.03400 to 0.03419
4      Yes      f(2, 176) = 27.82      < 0.001      0.00000 to 0.00000
5      Yes        f(1, 88) = 17.3      < 0.001      0.00007 to 0.00008
6      Yes      f(2, 176) = 13.23      < 0.001      0.00000 to 0.00000
7      Yes       f(2, 176) = 85.5      < 0.001      0.00000 to 0.00000
8      Yes       f(1, 88) = 19.48      < 0.001      0.00003 to 0.00003
9      Yes      f(2, 176) = 79.09      < 0.001      0.00000 to 0.00000
10     Yes       f(1, 88) = 27.91      < 0.001      0.00000 to 0.00000
11     Yes       f(2, 176) = 3.68      = 0.027      0.02705 to 0.02731
12     Yes       f(2, 176) = 5.32      < 0.006      0.00568 to 0.00574
13     Yes      f(1, 104) = 13.11      < 0.001      0.00045 to 0.00046
14     Yes       f(3, 104) = 7.65      < 0.001      0.00011 to 0.00011
15      No       f(3, 104) = 5.67      < 0.001      0.00122 to 0.00124
Recalculated p-value does not match the reported p-value.
16     Yes       f(3, 104) = 3.69      = 0.014      0.01421 to 0.01439
17     Yes      f(1, 104) = 24.24      < 0.001      0.00000 to 0.00000
18     Yes       f(3, 104) = 9.73      < 0.001      0.00001 to 0.00001
20     Yes       f(5, 114) = 6.81      < 0.001      0.00001 to 0.00001
21     Yes      f(1, 114) = 18.99      < 0.001      0.00003 to 0.00003
22     Yes       f(5, 114) = 4.07      = 0.002      0.00193 to 0.00197
23      No       f(5, 114) = 2.77      < 0.021      0.02103 to 0.02142
Recalculated p-value does not match the reported p-value.
```

24	Yes	f(5, 114) = 8.82	< 0.001	0.00000 to 0.00000
25	Yes	f(5, 114) = 10.03	< 0.001	0.00000 to 0.00000
26	Yes	f(5, 114) = 14.11	< 0.001	0.00000 to 0.00000

(9/48)
EXCLUDE THIS ARTICLE
JPSP_Eagly_2004_87_6_796_Gender gaps in.htm

(10/48)
JPSP_Eberhardt_2004_87_6_876_Seeing black. Race.htm

	Consistent	APA Reporting	Reported P-value	Valid P-value Range
0	Yes	f(2, 36) = 5.98	< 0.01	0.00570 to 0.00574
1	Yes	f(2, 36) = 7.04	< 0.01	0.00262 to 0.00264
2	Yes	t(25) = 4.54	< 0.01	0.00012 to 0.00012
3	Yes	t(24) = 2.34	< 0.05	0.02763 to 0.02824
5	Yes	t(13) = 2.96	= 0.01	0.01095 to 0.01116
6	Yes	t(12) = 2.35	< 0.05	0.03638 to 0.03705
7	Yes	f(1, 46) = 11.89	< 0.01	0.00122 to 0.00122
8	Yes	f(1, 46) = 8.22	< 0.01	0.00622 to 0.00625
10	Yes	f(1, 46) = 12.02	< 0.01	0.00115 to 0.00115
12	Yes	f(1, 65) = 5.33	< 0.05	0.02409 to 0.02422
13	Yes	f(1, 65) = 4.96	< 0.05	0.02933 to 0.02949
14	Yes	f(1, 65) = 6.6	= 0.01	0.01219 to 0.01282
15	Yes	f(1, 53) = 15.24	< 0.01	0.00027 to 0.00027
16	No	f(1, 53) = 3.95	< 0.05	0.05191 to 0.05219

Gross inconsistency: reported p-value and recalculated p-value differ in significance.

17	Yes	f(1, 53) = 12.6	< 0.01	0.00080 to 0.00084
18	Yes	f(1, 53) = 9.74	< 0.01	0.00291 to 0.00292
19	Yes	f(1, 53) = 5.87	< 0.05	0.01880 to 0.01890
20	Yes	t(56) = 10.49	< 0.01	0.00000 to 0.00000
21	Yes	t(56) = 3.03	< 0.01	0.00364 to 0.00375
23	Yes	f(1, 55) = 16.82	< 0.01	0.00014 to 0.00014
24	Yes	f(1, 55) = 7.3	< 0.01	0.00893 to 0.00938
25	Yes	t(55) = 2.35	< 0.05	0.02212 to 0.02266
27	Yes	f(1, 76) = 6.35	= 0.01	0.01380 to 0.01387
28	Yes	f(1, 74) = 4.6	< 0.05	0.03430 to 0.03623
29	Yes	f(1, 36) = 4.78	< 0.05	0.03529 to 0.03547
31	Yes	f(1, 38) = 9.74	< 0.01	0.00343 to 0.00344
32	Yes	f(1, 74) = 8.12	< 0.01	0.00565 to 0.00568

(11/48)
JEP_Estes_2004_30_5_1082_The importance of.htm

0	Yes	chi2(1) = 6.9	< 0.01	0.00838 to 0.00886	-
1	Yes	f(1, 79) = 55.07	< 0.001	0.00000 to 0.00000	-
2	Yes	f(1, 12) = 28.96	< 0.001	0.00016 to 0.00017	-

```
3      Yes     f(1, 79) = 23.15      < 0.001      0.00001 to 0.00001    -
4      Yes     f(1, 12) = 32.64      < 0.001      0.00010 to 0.00010    -
5      Yes      f(1, 79) = 58.6      < 0.001      0.00000 to 0.00000    -
6      Yes     f(1, 12) = 25.67      < 0.001      0.00028 to 0.00028    -
7      Yes         t(79) = 8.75      < 0.001      0.00000 to 0.00000    -
8      Yes           t(6) = 6.5      < 0.001      0.00061 to 0.00066    -

(12/48)
JPSP_Exline_2004_87_6_894_Too proud to.htm
11     Yes         t(267) = 2.59       = 0.01      0.00998 to 0.01027
5     Yes      f(1, 210) = 4.34       < 0.05      0.03833 to 0.03855
1     Yes         t(161) = 3.86      < 0.001      0.00016 to 0.00017
22     Yes          t(161) = 2.3       < 0.05      0.01999 to 0.02580
34     Yes          t(152) = 3.2       < 0.01      0.00142 to 0.00197
1     Yes      f(1, 110) = 54.2      < 0.001      0.00000 to 0.00000
52     Yes     f(1, 110) = 25.01      < 0.001      0.00000 to 0.00000
53     Yes     f(1, 110) = 27.58      < 0.001      0.00000 to 0.00000

(13/48)
EXCLUDE THIS ARTICLE
JPSP_Feeney_2004_87_5_631_A secure base.htm

(14/48)
JPSP_Ferguson_2004_87_5_557_Liking is for.htm
   Consistent   APA Reporting   Reported P-value Valid P-value Range
0      Yes      f(1, 69) = 6.18      = 0.015      0.01531 to 0.01539
1      Yes     f(1, 16) = 11.84      = 0.003      0.00335 to 0.00336
3      Yes         t(69) = 1.68       < 0.05      0.04825 to 0.04923
4      Yes         t(69) = 1.73       < 0.05      0.04360 to 0.04450
7       No     f(1, 34) = 99.78        = 0.0      0.00000 to 0.00000
A p-value is never exactly 0. Recalculated p-value does not match the reported p-value.
8       No         t(18) = 4.96        = 0.0      0.00010 to 0.00010
A p-value is never exactly 0. Recalculated p-value does not match the reported p-value.
Consistent for one-tailed, inconsistent for two-tailed
10     Yes      f(1, 34) = 5.91       = 0.02      0.02043 to 0.02053
11     Yes      f(3, 32) = 3.03      = 0.044      0.04333 to 0.04379
12     Yes      f(1, 34) = 4.86      = 0.034      0.03427 to 0.03444
13     Yes      f(1, 34) = 8.48      = 0.006      0.00629 to 0.00631
14      No     f(1, 16) = 27.57        = 0.0      0.00008 to 0.00008
A p-value is never exactly 0.
Recalculated p-value does not match the reported p-value.
15     Yes          t(34) = 2.0       < 0.05      0.04814 to 0.05946
16     Yes      f(1, 18) = 9.87      = 0.006      0.00563 to 0.00565
17     Yes     f(1, 16) = 13.22      = 0.002      0.00222 to 0.00223
18      No      f(3, 14) = 5.45       < 0.01      0.01074 to 0.01081
```

```
Recalculated p-value does not match the reported p-value.

23     Yes      f(2, 51) = 3.85      = 0.028      0.02760 to 0.02784
24     Yes      f(1, 16) = 7.12      = 0.017      0.01680 to 0.01686
25     Yes      f(1, 18) = 6.02      = 0.025      0.02451 to 0.02461
26     Yes     f(1, 10) = 12.72      = 0.005      0.00512 to 0.00513
27     Yes     f(1, 10) = 22.28      = 0.001      0.00082 to 0.00082
28     Yes     f(1, 10) = 10.15       = 0.01      0.00971 to 0.00973
29     Yes      f(1, 10) = 6.88      = 0.025      0.02543 to 0.02551
31     Yes      f(1, 17) = 4.87      = 0.041      0.04127 to 0.04146
32     Yes      f(1, 20) = 8.71      = 0.008      0.00788 to 0.00791
33     Yes      f(1, 20) = 6.16       = 0.02      0.02202 to 0.02211
34     Yes          t(29) = 2.0       < 0.05      0.02475 to 0.03045
35     Yes         t(29) = 2.06      < 0.025      0.02398 to 0.02449
37      No     f(1, 56) = 15.36        = 0.0      0.00024 to 0.00024
A p-value is never exactly 0. Recalculated p-value does not match the reported p-value.
38      No     f(1, 56) = 16.94        = 0.0      0.00013 to 0.00013
A p-value is never exactly 0. Recalculated p-value does not match the reported p-value.
39     Yes      f(1, 56) = 6.91      = 0.011      0.01102 to 0.01107
40      No     f(1, 56) = 19.79        = 0.0      0.00004 to 0.00004
A p-value is never exactly 0. Recalculated p-value does not match the reported p-value.
41      No     f(1, 56) = 13.81        = 0.0      0.00047 to 0.00047
A p-value is never exactly 0. Recalculated p-value does not match the reported p-value.
42     Yes      f(1, 56) = 4.97       = 0.03      0.02974 to 0.02990

(15/48)
JEP_Folstein_2004_30_5_1026_Multidimensional rule, unidimensional.htm
1      Yes      f(1, 18) = 137.1     < 0.0001      0.00000 to 0.00000
2      Yes       f(4, 74) = 8.72     < 0.0001      0.00001 to 0.00001
3      Yes       f(2, 74) = 6.59      < 0.005      0.00231 to 0.00233
6      Yes       f(2, 36) = 4.22       < 0.05      0.02248 to 0.02266
7      Yes          t(18) = 3.26       < 0.01      0.00430 to 0.00440
8      Yes           t(28) = 3.6      < 0.005      0.00106 to 0.00138
9      Yes       f(1, 18) = 8.73       < 0.01      0.00847 to 0.00850
11     Yes       f(1, 37) = 65.0     < 0.0001      0.00000 to 0.00000
12     Yes       f(4, 74) = 14.0       < 0.01      0.00000 to 0.00000
13     Yes       f(2, 37) = 13.2     < 0.0001      0.00005 to 0.00005
14     Yes        f(4, 74) = 4.6       < 0.01      0.00210 to 0.00243
15     Yes       f(1, 18) = 12.8      < 0.005      0.00212 to 0.00218
16     Yes       f(1, 18) = 11.0      < 0.005      0.00377 to 0.00390
17     Yes       f(1, 18) = 24.8       < 0.01      0.00010 to 0.00010
18     Yes       f(1, 28) = 17.9     < 0.0005      0.00022 to 0.00023
19     Yes       f(1, 18) = 13.7      < 0.002      0.00161 to 0.00166
23     Yes     f(12, 444) = 2.44       < 0.05      0.00432 to 0.00449
25     Yes      f(4, 148) = 8.98     < 0.0001      0.00000 to 0.00000
```

```
26     Yes      f(4, 148) = 9.67     < 0.0001      0.00000 to 0.00000
27     Yes      f(4, 148) = 12.2       < 0.01      0.00000 to 0.00000
29     Yes       f(1, 37) = 8.01       < 0.01      0.00746 to 0.00749
30     Yes      f(6, 114) = 3.85       < 0.05      0.00154 to 0.00157
31      No       f(4, 76) = 4.29       = 0.05      0.00347 to 0.00352
Recalculated p-value does not match the reported p-value.
33     Yes       f(1, 19) = 4.71       < 0.05      0.04277 to 0.04297
34     Yes       f(1, 19) = 20.6     < 0.0005      0.00022 to 0.00023
36     Yes       f(1, 28) = 5.98       < 0.05      0.02097 to 0.02106
37     Yes      f(6, 168) = 2.42        < 0.1      0.02834 to 0.02895
38     Yes      f(5, 140) = 2.53        < 0.1      0.03137 to 0.03195
39     Yes        f(1, 28) = 6.2       < 0.05      0.01855 to 0.01942
40      No      f(6, 108) = 2.71       = 0.05      0.01704 to 0.01739
Recalculated p-value does not match the reported p-value.
41     Yes       f(5, 90) = 2.43        < 0.1      0.04060 to 0.04132
42     Yes       f(3, 84) = 3.98       < 0.05      0.01048 to 0.01061
43     Yes       f(3, 84) = 3.44       < 0.05      0.02031 to 0.02056

(16/48)
JEP_Glanzer_2004_30_6_1176_Six regularities of.htm
0      Yes       f(1, 59) = 8.87      = 0.004      0.00419 to 0.00421
1      Yes          t(59) = 4.77      < 0.001      0.00001 to 0.00001
2      Yes          t(59) = 6.34      < 0.001      0.00000 to 0.00000
4      Yes           t(59) = 2.4       < 0.02      0.01727 to 0.02214
7      Yes       f(1, 69) = 28.3      < 0.001      0.00000 to 0.00000
8      Yes      f(1, 69) = 86.84      < 0.001      0.00000 to 0.00000
9      Yes      f(1, 69) = 19.73      < 0.001      0.00003 to 0.00003
13     Yes          t(55) = 4.48      < 0.001      0.00004 to 0.00004
15     Yes          t(55) = 8.69      < 0.001      0.00000 to 0.00000
18     Yes       f(1, 55) = 6.11      = 0.017      0.01652 to 0.01661
19     Yes      f(1, 55) = 22.54      < 0.001      0.00002 to 0.00002
20     Yes       f(1, 55) = 4.36      = 0.041      0.04132 to 0.04155
24     Yes          t(35) = 5.27      < 0.001      0.00001 to 0.00001
26     Yes           t(35) = 3.4      < 0.002      0.00148 to 0.00195
29     Yes          t(35) = 8.76      < 0.001      0.00000 to 0.00000
30     Yes       f(1, 35) = 7.95      = 0.008      0.00785 to 0.00788
31     Yes       f(1, 35) = 7.99      = 0.008      0.00771 to 0.00774
32      No       f(1, 35) = 7.57      = 0.008      0.00931 to 0.00936
34     Yes          t(46) = 7.51      < 0.001      0.00000 to 0.00000
36     Yes          t(46) = 5.12      < 0.001      0.00001 to 0.00001
39     Yes       f(1, 50) = 8.33      < 0.006      0.00573 to 0.00576
40      No       f(1, 50) = 7.56      < 0.008      0.00826 to 0.00830
Recalculated p-value does not match the reported p-value.
43     Yes     f(1, 50) = 142.97      < 0.001      0.00000 to 0.00000
45     Yes          t(62) = 4.88      < 0.001      0.00001 to 0.00001
```

```
47     Yes          t(62) = 4.19      < 0.001      0.00009 to 0.00009
50     Yes       f(1, 64) = 8.55      = 0.005      0.00476 to 0.00478
52     Yes          t(62) = 3.53      < 0.001      0.00078 to 0.00080
53     Yes       f(1, 61) = 5.17      = 0.026      0.02644 to 0.02658
56     Yes          t(41) = 11.3      < 0.001      0.00000 to 0.00000
57     Yes          t(23) = 3.65      < 0.002      0.00132 to 0.00135
58     Yes       f(1, 64) = 5.97      = 0.017      0.01728 to 0.01736

(17/48)
EXCLUDE THIS ARTICLE
JPSP_Golec_2004_87_6_750_Understanding responses to.htm

0     Yes         f(1, 98) = 26.47      < 0.01       0.00000 to 0.00000
1     Yes     chi2(1, 946) = 11.28      < 0.01       0.00078 to 0.00079
2     Yes          chi2(1) = 11.57      < 0.01       0.00067 to 0.00067
3      No          chi2(1) = 5.02       < 0.01       0.02498 to 0.02513
Recalculated p-value does not match the reported p-value.
4     Yes           chi2(1) = 7.0       < 0.01       0.00793 to 0.00838

(18/48)
JEP_Heit_2004_30_5_1065_Modeling the effects.htm
0      Yes      f(2, 90) = 101.43      < 0.001      0.00000 to 0.00000
1       No      f(10, 450) = 2.99      < 0.001      0.00113 to 0.00118
Recalculated p-value does not match the reported p-value.
2      Yes       f(2, 90) = 40.38      < 0.001      0.00000 to 0.00000
4      Yes          t(45) = 7.32       < 0.001      0.00000 to 0.00000
5      Yes       f(5, 225) = 5.14      < 0.001      0.00017 to 0.00018
6       No         f(1, 78) = 9.1      < 0.001      0.00337 to 0.00354
Recalculated p-value does not match the reported p-value.
7      Yes        f(2, 78) = 65.2      < 0.001      0.00000 to 0.00000
8      Yes     f(5, 390) = 107.32      < 0.001      0.00000 to 0.00000
9      Yes      f(2, 156) = 39.65      < 0.001      0.00000 to 0.00000
10     Yes      f(10, 780) = 2.71       < 0.01      0.00278 to 0.00288
11     Yes     f(5, 390) = 111.75      < 0.001      0.00000 to 0.00000
12     Yes      f(1, 78) = 101.66      < 0.001      0.00000 to 0.00000
13     Yes       f(5, 390) = 3.76       < 0.01      0.00243 to 0.00248
14      No        f(1, 78) = 9.23      < 0.001      0.00323 to 0.00325
Recalculated p-value does not match the reported p-value.
15     Yes      f(2, 117) = 52.22      < 0.001      0.00000 to 0.00000
16     Yes         f(1, 78) = 8.7       < 0.01      0.00410 to 0.00430
17     Yes      f(5, 390) = 15.67      < 0.001      0.00000 to 0.00000
18     Yes          t(39) = 3.46        < 0.01      0.00130 to 0.00134
19     Yes          t(78) = 2.52        < 0.05      0.01360 to 0.01396
21     Yes       f(5, 39) = 32.09      < 0.001      0.00000 to 0.00000
22     Yes      f(10, 390) = 3.38      < 0.001      0.00030 to 0.00031
```

```
23     Yes        f(5, 195) = 3.6       < 0.01      0.00352 to 0.00428

(19/48)
JPSP_Hewig_2004_87_6_926_On the selective.htm
7       No     f(9, 504) = 2.25      = 0.036      0.01770 to 0.01824
Recalculated p-value does not match the reported p-value.
11      No     f(9, 504) = 2.49      = 0.033      0.00850 to 0.00877
Recalculated p-value does not match the reported p-value.
12     Yes     f(9, 504) = 2.09       < 0.05      0.02845 to 0.02929
13      No     f(9, 504) = 2.81      = 0.011      0.00310 to 0.00320
Recalculated p-value does not match the reported p-value.
16      No     f(9, 504) = 2.08       = 0.05      0.02929 to 0.03016
Recalculated p-value does not match the reported p-value.
17     Yes      f(1, 56) = 7.32      = 0.009      0.00900 to 0.00904
18      No      f(9, 504) = 2.3      = 0.035      0.01328 to 0.01797
Recalculated p-value does not match the reported p-value.
23      No     f(9, 513) = 2.27      = 0.032      0.01664 to 0.01715
Recalculated p-value does not match the reported p-value.
27      No     f(9, 513) = 3.11      = 0.023      0.00116 to 0.00120
Recalculated p-value does not match the reported p-value.

(20/48)
EXCLUDE THIS ARTICLE
JEP_Hohlfeld_2004_30_5_1012_Effects of additional.htm

(21/48)
JEP_Jahn_2004_30_5_969_Three turtles in.htm
0      Yes      f(1, 31) = 6.65      < 0.05       0.01485 to 0.01492    -
1      Yes      f(1, 22) = 5.85      < 0.05       0.02424 to 0.02435    -
4      Yes      f(1, 31) = 6.05      < 0.05       0.01964 to 0.01973    -
5      Yes      f(1, 22) = 5.43      < 0.05       0.02930 to 0.02943    -
7      Yes      f(1, 31) = 7.24      < 0.05       0.01136 to 0.01141    -
8      Yes      f(1, 22) = 4.37      < 0.05       0.04823 to 0.04846    -
12     Yes      f(1, 62) = 4.13      < 0.05       0.04629 to 0.04655    -
16     Yes       f(1, 31) = 4.5      < 0.05       0.04094 to 0.04308    -
17     Yes      f(1, 23) = 4.78      < 0.05       0.03913 to 0.03932    -
19     Yes      f(1, 62) = 4.68      < 0.05       0.03429 to 0.03447    -
20     Yes       f(1, 62) = 4.3      < 0.05       0.04113 to 0.04345    -
21     Yes     f(1, 31) = 11.97      < 0.01       0.00159 to 0.00160    -
22     Yes     f(1, 22) = 14.14      < 0.01       0.00108 to 0.00108    -

(22/48)
JPSP_Johnson_2004_87_5_615_Inferenes about the.htm
0      Yes     f(1, 211) = 118.38     < 0.0001      0.00000 to 0.00000
1      Yes       f(1, 211) = 6.07      < 0.015      0.01451 to 0.01459
```

```
2       Yes       f(1, 211) = 36.2     < 0.0001      0.00000 to 0.00000
3       Yes     f(1, 211) = 110.71     < 0.0001      0.00000 to 0.00000
4       Yes      f(1, 211) = 24.47     < 0.0001      0.00000 to 0.00000
5       Yes      f(1, 211) = 30.28     < 0.0001      0.00000 to 0.00000
6       Yes       f(1, 211) = 6.47      < 0.015      0.01166 to 0.01172
7       Yes     f(1, 211) = 623.09     < 0.0001      0.00000 to 0.00000
8       Yes     f(1, 211) = 161.21     < 0.0001      0.00000 to 0.00000
9       Yes      f(1, 211) = 49.15     < 0.0001      0.00000 to 0.00000
10      Yes       f(1, 211) = 8.48      < 0.004      0.00397 to 0.00399
11       No       f(1, 211) = 8.35      < 0.004      0.00425 to 0.00427
Recalculated p-value does not match the reported p-value.
12       No       f(1, 211) = 9.64      < 0.002      0.00216 to 0.00217
Recalculated p-value does not match the reported p-value.
13       No          t(209) = 3.69     < 0.0001      0.00028 to 0.00029
Recalculated p-value does not match the reported p-value.
16      Yes      f(1, 193) = 19.38     < 0.0001      0.00002 to 0.00002
17      Yes       f(1, 193) = 31.8     < 0.0001      0.00000 to 0.00000
18      Yes      f(1, 193) = 26.64     < 0.0001      0.00000 to 0.00000
19      Yes      f(1, 193) = 40.78     < 0.0001      0.00000 to 0.00000
23      Yes      f(1, 193) = 61.15     < 0.0001      0.00000 to 0.00000
24      Yes       f(1, 193) = 4.98       < 0.03      0.02672 to 0.02687
25      Yes      f(1, 193) = 80.57     < 0.0001      0.00000 to 0.00000
26      Yes      f(1, 193) = 11.68      < 0.001      0.00077 to 0.00077
27      Yes      f(1, 193) = 60.63     < 0.0001      0.00000 to 0.00000
31      Yes          t(194) = 5.16     < 0.0001      0.00000 to 0.00000
32      Yes          t(194) = 8.98     < 0.0001      0.00000 to 0.00000
33      Yes          t(193) = 4.39     < 0.0001      0.00002 to 0.00002
34      Yes          t(193) = 2.15       < 0.04      0.03240 to 0.03320

(23/48)
JPSP_Koole_2004_87_6_974_Getting a grip.htm
0       Yes       f(1, 78) = 9.81      < 0.003      0.00244 to 0.00245
2       Yes       f(2, 77) = 3.56       < 0.04      0.03306 to 0.03336
4       Yes       f(1, 21) = 4.85       < 0.04      0.03887 to 0.03905
6       Yes       f(2, 39) = 4.68       < 0.02      0.01502 to 0.01514
7       Yes       f(2, 39) = 3.56       < 0.04      0.03785 to 0.03817
8       Yes       f(2, 44) = 5.64      < 0.008      0.00657 to 0.00663
10      Yes       f(1, 22) = 6.89       < 0.02      0.01543 to 0.01550
15      Yes      f(1, 56) = 17.36      < 0.001      0.00011 to 0.00011
16      Yes       f(1, 53) = 7.77      < 0.008      0.00734 to 0.00738
17      Yes       f(1, 53) = 4.85       < 0.04      0.03193 to 0.03210
18      Yes       f(1, 53) = 7.77      < 0.008      0.00734 to 0.00738
19      Yes       f(1, 53) = 4.85       < 0.04      0.03193 to 0.03210
20      Yes      f(1, 53) = 12.91      < 0.002      0.00071 to 0.00072
21       No       f(1, 53) = 2.91       < 0.05      0.09361 to 0.09416
```

```
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
22     Yes      f(1, 56) = 5.84      < 0.02      0.01890 to 0.01900
23     Yes      f(1, 53) = 5.14      < 0.03      0.02741 to 0.02756
24     Yes      f(1, 53) = 6.97      < 0.02      0.01084 to 0.01089
30     Yes     f(1, 67) = 42.42     < 0.001      0.00000 to 0.00000
31     Yes    f(1, 67) = 174.72     < 0.001      0.00000 to 0.00000
33     Yes      f(1, 67) = 4.61      < 0.04      0.03531 to 0.03551
34     Yes      f(1, 67) = 8.26     < 0.006      0.00541 to 0.00544
35     Yes      f(1, 67) = 4.46      < 0.04      0.03832 to 0.03854
36     Yes     f(2, 69) = 15.38     < 0.001      0.00000 to 0.00000
37     Yes     f(1, 70) = 17.63     < 0.001      0.00008 to 0.00008
38      No      f(1, 70) = 4.24     < 0.001      0.04308 to 0.04332
Recalculated p-value does not match the reported p-value.
39     Yes      f(1, 67) = 5.11      < 0.03      0.02697 to 0.02711
40     Yes       f(1, 32) = 5.1      < 0.04      0.03012 to 0.03165
41     Yes         t(32) = 2.79      < 0.01      0.00870 to 0.00892
42     Yes         t(32) = 2.26      < 0.04      0.03042 to 0.03111
43     Yes         t(32) = 2.73      < 0.02      0.01009 to 0.01034

(24/48)
JPSP_Lord_2004_87_5_733_Houses built on.htm
0      Yes    f(1, 58) = 153.98     < 0.001      0.00000 to 0.00000
4      Yes      f(1, 57) = 6.48      < 0.05      0.01360 to 0.01367
5      Yes       f(1, 56) = 6.1      < 0.05      0.01618 to 0.01702
6      Yes      f(1, 56) = 6.05      < 0.05      0.01698 to 0.01706
7      Yes      f(1, 58) = 4.36      < 0.05      0.04108 to 0.04131
8      Yes       f(1, 57) = 4.0      < 0.05      0.04890 to 0.05168
11     Yes      f(2, 53) = 4.03      < 0.05      0.02338 to 0.02359
12     Yes         t(36) = 3.16      < 0.01      0.00315 to 0.00324
13     Yes      f(2, 53) = 5.08      < 0.01      0.00954 to 0.00963
14     Yes     f(1, 50) = 156.6     < 0.001      0.00000 to 0.00000
18     Yes      f(2, 48) = 3.45      < 0.05      0.03964 to 0.03999
19     Yes       f(2, 48) = 3.5      < 0.05      0.03649 to 0.03982
23     Yes      f(2, 49) = 3.82      < 0.05      0.02860 to 0.02885
24     Yes     f(4, 172) = 3.35      < 0.05      0.01129 to 0.01148
25      No      f(1, 86) = 3.36      < 0.05      0.07005 to 0.07046
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
29     Yes     f(2, 172) = 4.92      < 0.01      0.00832 to 0.00840
30     Yes      f(2, 86) = 3.18      < 0.05      0.04630 to 0.04674
31     Yes      f(1, 86) = 5.83      < 0.05      0.01783 to 0.01792
32     Yes      f(1, 86) = 5.15      < 0.05      0.02568 to 0.02582
33     Yes     f(2, 168) = 3.23      < 0.05      0.04182 to 0.04223
34     Yes     f(2, 167) = 3.15      < 0.05      0.04519 to 0.04563
36     Yes     f(2, 169) = 3.37      < 0.05      0.03654 to 0.03689
37     Yes      f(1, 169) = 6.6      < 0.05      0.01077 to 0.01137
```

```
40     Yes      f(1, 125) = 4.34       < 0.05      0.03915 to 0.03938

(25/48)
JEP_McKenzie_2004_30_5_947_Explaining purportedly irrational.htm
0     Yes        t(96) = 4.61       < 0.01      0.00001 to 0.00001
1     Yes     f(1, 15) = 16.9       < 0.05      0.00091 to 0.00094
3     Yes        t(48) = 3.11       < 0.01      0.00310 to 0.00319
4     Yes        t(46) = 3.14       < 0.01      0.00291 to 0.00299
5     Yes        t(29) = 2.36       = 0.03      0.02494 to 0.02550
6      No        t(29) = 2.65       < 0.01      0.01275 to 0.01305
Recalculated p-value does not match the reported p-value.
Consistent for one-tailed, inconsistent for two-tailed
8     Yes        t(90) = 3.32      = 0.001      0.00128 to 0.00132

(26/48)
JPSP_Meiser_2004_87_5_599_Cognitive processes in.htm
0      Yes       f(1, 38) = 6.1      = 0.018      0.01769 to 0.01857    -
1      Yes      f(1, 38) = 8.38      = 0.006      0.00624 to 0.00627    -
3      Yes      f(1, 38) = 14.6      < 0.001      0.00047 to 0.00049    -
4      Yes      f(1, 38) = 5.22      = 0.028      0.02793 to 0.02807    -
8      Yes     f(1, 76) = 23.95      < 0.001      0.00001 to 0.00001    -
9      Yes     f(1, 76) = 52.26      < 0.001      0.00000 to 0.00000    -
10     Yes      f(1, 76) = 4.41      = 0.039      0.03894 to 0.03916    -
12     Yes      f(1, 74) = 4.78      = 0.032      0.03187 to 0.03204    -
13     Yes     f(1, 76) = 10.06      = 0.002      0.00218 to 0.00219    -
14     Yes     f(1, 76) = 37.17      < 0.001      0.00000 to 0.00000    -
15     Yes      f(1, 76) = 8.32      = 0.005      0.00509 to 0.00511    -
16     Yes      f(1, 76) = 5.42      = 0.023      0.02251 to 0.02263    -
22     Yes      f(1, 99) = 9.09      = 0.003      0.00326 to 0.00327    -
23     Yes     f(1, 99) = 55.02      < 0.001      0.00000 to 0.00000    -
24     Yes      f(1, 99) = 5.79      = 0.018      0.01792 to 0.01802    -
25     Yes     f(1, 98) = 72.05      < 0.001      0.00000 to 0.00000    -
26     Yes     f(1, 33) = 23.13      < 0.001      0.00003 to 0.00003    -
27     Yes     f(1, 65) = 63.71      < 0.001      0.00000 to 0.00000    -
29     Yes       f(1, 99) = 5.0      = 0.028      0.02685 to 0.02836    -
30     Yes      f(1, 99) = 47.9      < 0.001      0.00000 to 0.00000    -
31     Yes      f(1, 98) = 6.78      = 0.011      0.01062 to 0.01068    -
32     Yes      f(1, 98) = 9.22      = 0.003      0.00306 to 0.00307    -
33     Yes     f(1, 65) = 12.85      = 0.001      0.00064 to 0.00065    -
35     Yes     f(1, 65) = 54.22      < 0.001      0.00000 to 0.00000    -

(27/48)
EXCLUDE THIS ARTICLE
JPSP_Mikulincer_2004_87_6_940_Attachment-related strategies during.htm
```

```
(28/48)
EXCLUDE THIS ARTICLE
JEP_Moscoso del Prado Martín_2004_30_6_1271_Morphological family size.htm

(29/48)
JPSP_Mussweiler_2004_87_6_832_The ups and.htm
0      No          t(14) = 2.0      < 0.03       0.05958 to 0.07149
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
Consistent for one-tailed, inconsistent for two-tailed
1      No     f(1, 44) = 5.58      < 0.02       0.02260 to 0.02271
Recalculated p-value does not match the reported p-value.
2      No     f(1, 44) = 4.04      < 0.05       0.05045 to 0.05073
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
3     Yes     f(1, 50) = 4.56      < 0.04       0.03755 to 0.03775
4      No      f(1, 50) = 5.7      < 0.02       0.02026 to 0.02132
Recalculated p-value does not match the reported p-value.
5     Yes     f(1, 50) = 7.76      < 0.01       0.00751 to 0.00754
6     Yes     f(1, 50) = 7.63      < 0.01       0.00799 to 0.00803
7      No     f(1, 50) = 5.71      < 0.02       0.02063 to 0.02073
Recalculated p-value does not match the reported p-value.

(30/48)
JEP_Norris_2004_30_5_1093_Retroactive effects of.htm
1      Yes      f(3, 36) = 8.14      < 0.01       0.00029 to 0.00029
2      Yes      f(3, 36) = 14.4      < 0.01       0.00000 to 0.00000
4      Yes      f(1, 24) = 13.6      < 0.01       0.00114 to 0.00117
5      Yes      f(2, 48) = 49.4      < 0.01       0.00000 to 0.00000
6      Yes      f(3, 72) = 62.2      < 0.01       0.00000 to 0.00000
8      Yes          t(29) = 2.4      = 0.01       0.01028 to 0.01290
9      Yes          t(29) = 1.8      = 0.04       0.03726 to 0.04535
10     Yes      f(1, 30) = 16.5      < 0.01       0.00032 to 0.00033
11     Yes      f(2, 60) = 98.7      < 0.01       0.00000 to 0.00000
12     Yes      f(3, 90) = 87.9      < 0.01       0.00000 to 0.00000
13     Yes          t(35) = 2.0      = 0.03       0.02396 to 0.02961
16     Yes       f(2, 60) = 6.0      < 0.01       0.00404 to 0.00439
17     Yes          t(35) = 3.3      < 0.01       0.00195 to 0.00255
18     Yes       f(1, 28) = 4.4      < 0.05       0.04397 to 0.04624
19     Yes      f(1, 28) = 19.1      < 0.01       0.00015 to 0.00016
20      No      f(1, 28) = 4.53      < 0.04       0.04214 to 0.04235
Recalculated p-value does not match the reported p-value.
21     Yes      f(3, 84) = 27.5      < 0.01       0.00000 to 0.00000
22     Yes         t(31) = 2.71      = 0.01       0.01073 to 0.01100
24     Yes         t(31) = 1.86      = 0.04       0.03584 to 0.03656
26     Yes       f(1, 23) = 7.5      = 0.01       0.01146 to 0.01195
27     Yes      f(3, 69) = 35.6      < 0.01       0.00000 to 0.00000
```

(31/48)
JPSP_Norton_2004_87_6_817_Casuistry and social.htm

0	Yes	chi2(1) = 9.04	< 0.005	0.00263 to 0.00265
1	Yes	chi2(2) = 9.18	< 0.02	0.01013 to 0.01018
6	Yes	t(45) = 9.37	< 0.001	0.00000 to 0.00000
8	Yes	chi2(2) = 6.25	< 0.05	0.04383 to 0.04405
10	Yes	chi2(1) = 8.85	< 0.005	0.00292 to 0.00294
12	Yes	chi2(1) = 24.77	< 0.001	0.00000 to 0.00000
13	Yes	chi2(1) = 13.44	< 0.001	0.00025 to 0.00025
14	Yes	chi2(1) = 9.86	< 0.01	0.00168 to 0.00169
18	Yes	chi2(1) = 52.6	< 0.001	0.00000 to 0.00000
20	Yes	chi2(1) = 26.6	< 0.001	0.00000 to 0.00000
21	Yes	chi2(1) = 7.06	< 0.01	0.00786 to 0.00790
23	Yes	t(120) = 1.99	< 0.05	0.04831 to 0.04943
24	Yes	chi2(1) = 36.5	< 0.001	0.00000 to 0.00000
25	Yes	chi2(1) = 16.85	< 0.001	0.00004 to 0.00004
26	Yes	chi2(1) = 20.62	< 0.001	0.00001 to 0.00001
27	Yes	chi2(1) = 4.43	< 0.05	0.03521 to 0.03542
28	Yes	chi2(1) = 13.98	< 0.001	0.00018 to 0.00019
32	Yes	chi2(1) = 37.56	< 0.001	0.00000 to 0.00000
33	Yes	chi2(1) = 20.06	< 0.001	0.00001 to 0.00001

(32/48)
JEP_Pexman_2004_30_6_1252_Semantic ambiguity and.htm

0	Yes	f(1, 39) = 25.04	< 0.01	0.00001 to 0.00001
1	Yes	f(1, 56) = 5.7	< 0.05	0.01985 to 0.02090
2	Yes	f(1, 39) = 7.74	< 0.01	0.00826 to 0.00830
3	Yes	f(1, 56) = 4.19	< 0.05	0.04524 to 0.04549
4	Yes	f(1, 39) = 18.05	< 0.01	0.00013 to 0.00013
5	Yes	f(1, 56) = 4.41	< 0.05	0.04014 to 0.04036
6	Yes	f(1, 39) = 17.29	< 0.01	0.00017 to 0.00017
7	Yes	f(1, 56) = 17.29	< 0.01	0.00011 to 0.00011
8	Yes	f(1, 56) = 66.6	< 0.01	0.00000 to 0.00000
9	Yes	f(1, 39) = 106.78	< 0.01	0.00000 to 0.00000
11	Yes	f(1, 77) = 10.67	< 0.01	0.00162 to 0.00163
12	Yes	f(1, 77) = 120.6	< 0.01	0.00000 to 0.00000
13	Yes	f(1, 58) = 5.18	< 0.05	0.02649 to 0.02663
14	Yes	f(1, 77) = 130.85	< 0.01	0.00000 to 0.00000
15	Yes	f(1, 58) = 4.96	< 0.05	0.02976 to 0.02992
16	Yes	f(1, 31) = 18.86	< 0.01	0.00014 to 0.00014
19	Yes	f(1, 31) = 12.59	< 0.01	0.00126 to 0.00126
21	Yes	f(1, 31) = 41.31	< 0.01	0.00000 to 0.00000
22	Yes	f(1, 26) = 5.28	< 0.05	0.02980 to 0.02994
31	Yes	f(2, 50) = 17.32	< 0.01	0.00000 to 0.00000

```
33     Yes       f(2, 50) = 3.53       < 0.05      0.03665 to 0.03697
35     Yes          t(25) = 4.65       < 0.01      0.00009 to 0.00009
37     Yes          t(25) = 2.11       < 0.05      0.04457 to 0.04551
39     Yes          t(25) = 4.97       < 0.01      0.00004 to 0.00004
40     Yes          t(30) = 2.21       < 0.05      0.03450 to 0.03526
45     Yes       f(2, 50) = 9.77       < 0.01      0.00026 to 0.00026
47     Yes       f(2, 50) = 9.28       < 0.01      0.00037 to 0.00038
49     Yes          t(25) = 3.82       < 0.01      0.00078 to 0.00080
50     Yes          t(30) = 2.16       < 0.05      0.03847 to 0.03931
51     Yes          t(25) = 2.93       < 0.01      0.00705 to 0.00723
53     Yes          t(25) = 2.77       < 0.05      0.01030 to 0.01054
56     Yes          t(25) = 2.14       < 0.05      0.04186 to 0.04275
58     Yes          t(25) = 4.25       < 0.01      0.00026 to 0.00026
60     Yes          t(25) = 5.26       < 0.01      0.00002 to 0.00002
61     Yes          t(30) = 3.45       < 0.01      0.00166 to 0.00171
62     Yes           t(25) = 8.0      < 0.001      0.00000 to 0.00000
63     Yes          t(30) = 3.31       < 0.01      0.00240 to 0.00247
64     Yes          t(25) = 2.49       < 0.05      0.01956 to 0.02000
66     Yes          t(25) = 4.29       < 0.01      0.00023 to 0.00024
68     Yes          t(25) = 4.41       < 0.01      0.00017 to 0.00017
70     Yes          t(25) = 5.84       < 0.01      0.00000 to 0.00000

(33/48)
JEP_Rapp_2004_30_5_988_Interactive dimensions in.htm
0      Yes     f(1, 35) = 13.68      < 0.005      0.00074 to 0.00074
1      Yes      f(1, 19) = 6.62       < 0.05      0.01859 to 0.01867
2      Yes      f(1, 35) = 4.86       < 0.05      0.03407 to 0.03424
4      Yes         t(35) = 4.39      < 0.005      0.00010 to 0.00010
5      Yes         t(19) = 3.09       < 0.01      0.00596 to 0.00610
10     Yes      f(1, 35) = 4.53       < 0.05      0.04032 to 0.04053
12      No      f(1, 35) = 8.78      < 0.005      0.00543 to 0.00546
Recalculated p-value does not match the reported p-value.
14     Yes     f(1, 35) = 33.39      < 0.001      0.00000 to 0.00000
15     Yes     f(1, 18) = 16.26      < 0.005      0.00078 to 0.00078
16     Yes      f(1, 35) = 8.78       < 0.01      0.00543 to 0.00546
18     Yes         t(35) = 5.82      < 0.001      0.00000 to 0.00000
19      No         t(18) = 3.78      < 0.001      0.00136 to 0.00139
Recalculated p-value does not match the reported p-value.
Consistent for one-tailed, inconsistent for two-tailed
20     Yes         t(35) = 5.04      < 0.001      0.00001 to 0.00001
21     Yes         t(18) = 2.58       < 0.05      0.01868 to 0.01907
22     Yes         t(35) = 5.42      < 0.001      0.00000 to 0.00000
23      No         t(18) = 3.85      < 0.001      0.00116 to 0.00119
Recalculated p-value does not match the reported p-value.
Consistent for one-tailed, inconsistent for two-tailed
```

24	Yes	t(35) = 2.68	< 0.05	0.01101 to 0.01129	
26	Yes	t(35) = 2.53	< 0.05	0.01587 to 0.01625	
30	Yes	f(1, 35) = 9.24	< 0.005	0.00445 to 0.00447	
32	Yes	f(1, 35) = 11.48	< 0.005	0.00175 to 0.00176	
33	Yes	f(1, 19) = 13.04	< 0.005	0.00186 to 0.00186	
34	Yes	f(1, 35) = 6.19	< 0.05	0.01771 to 0.01780	
35	Yes	f(1, 19) = 6.61	< 0.05	0.01867 to 0.01874	
36	Yes	t(35) = 4.3	< 0.001	0.00011 to 0.00015	
37	Yes	t(19) = 3.19	< 0.005	0.00477 to 0.00488	
38	Yes	t(35) = 4.12	< 0.001	0.00022 to 0.00022	
39	Yes	t(19) = 2.86	< 0.01	0.00991 to 0.01013	
40	Yes	t(35) = 3.79	< 0.001	0.00056 to 0.00058	
41	Yes	t(19) = 4.17	< 0.001	0.00051 to 0.00053	
46	Yes	f(1, 70) = 5.0	< 0.05	0.02778 to 0.02932	
48	Yes	f(1, 70) = 4.85	< 0.05	0.03086 to 0.03103	
50	Yes	f(1, 70) = 5.69	< 0.05	0.01972 to 0.01982	
51	Yes	f(1, 38) = 4.51	< 0.05	0.04016 to 0.04037	

(34/48)
JEP_Rayner_2004_30_6_1290_The effect of.htm

1	Yes	t(32) = 2.32	< 0.05	0.02657 to 0.02718	-
2	Yes	t(24) = 2.55	< 0.05	0.01738 to 0.01777	-
3	Yes	t(35) = 3.43	< 0.01	0.00154 to 0.00159	-
5	Yes	f(2, 58) = 10.19	< 0.01	0.00016 to 0.00016	-
6	Yes	f(2, 70) = 3.44	< 0.05	0.03741 to 0.03775	-
7	Yes	f(2, 58) = 4.5	< 0.05	0.01460 to 0.01592	-
8	Yes	t(35) = 2.47	< 0.05	0.01831 to 0.01875	-
9	Yes	t(29) = 2.24	< 0.05	0.03256 to 0.03328	-
10	Yes	t(35) = 2.03	< 0.05	0.04948 to 0.05055	-
11	Yes	t(29) = 2.57	< 0.05	0.01539 to 0.01576	-
12	Yes	f(2, 70) = 6.45	< 0.01	0.00267 to 0.00270	-
13	Yes	f(2, 58) = 7.82	< 0.01	0.00098 to 0.00099	-
14	Yes	t(35) = 3.08	< 0.01	0.00396 to 0.00407	-
15	Yes	t(29) = 3.63	< 0.01	0.00107 to 0.00110	-
16	Yes	t(35) = 2.14	< 0.05	0.03897 to 0.03984	-
17	Yes	t(29) = 2.32	< 0.05	0.02728 to 0.02789	-
20	Yes	f(2, 70) = 7.89	< 0.01	0.00081 to 0.00082	-
21	Yes	f(2, 58) = 5.32	< 0.01	0.00753 to 0.00759	-
22	Yes	f(2, 70) = 5.17	< 0.01	0.00802 to 0.00809	-
23	Yes	f(2, 58) = 6.25	< 0.01	0.00347 to 0.00350	-
24	Yes	f(2, 70) = 5.78	< 0.01	0.00473 to 0.00477	-
25	Yes	f(2, 58) = 3.97	< 0.05	0.02411 to 0.02432	-
26	Yes	f(2, 70) = 27.74	< 0.01	0.00000 to 0.00000	-
27	Yes	f(2, 58) = 17.8	< 0.01	0.00000 to 0.00000	-

28	Yes	t(35) = 6.59	< 0.01	0.00000 to 0.00000	-
29	Yes	t(29) = 6.37	< 0.01	0.00000 to 0.00000	-
30	Yes	t(35) = 4.77	< 0.01	0.00003 to 0.00003	-
31	Yes	t(29) = 3.24	< 0.01	0.00296 to 0.00303	-
32	Yes	t(35) = 2.76	< 0.01	0.00902 to 0.00925	-
33	Yes	t(29) = 2.36	< 0.05	0.02494 to 0.02550	-
34	Yes	f(2, 70) = 19.71	< 0.01	0.00000 to 0.00000	-
35	Yes	f(2, 58) = 16.62	< 0.01	0.00000 to 0.00000	-
36	Yes	t(29) = 2.06	< 0.05	0.04797 to 0.04899	-

(35/48)
JEP_Rinck_2004_30_6_1211_The metrics of.htm

0	Yes	f(1, 27) = 4.38	< 0.05	0.04578 to 0.04601
1	Yes	f(1, 27) = 5.47	< 0.05	0.02693 to 0.02706
2	Yes	f(1, 27) = 6.13	< 0.05	0.01981 to 0.01990
3	Yes	t(27) = 2.48	< 0.05	0.01945 to 0.01989
4	Yes	t(27) = 2.18	< 0.05	0.03774 to 0.03856
5	Yes	t(27) = 2.91	< 0.05	0.00707 to 0.00724
7	Yes	f(1, 39) = 6.63	< 0.05	0.01390 to 0.01397
9	Yes	f(1, 39) = 4.48	< 0.05	0.04062 to 0.04083
10	Yes	f(1, 39) = 14.41	< 0.01	0.00050 to 0.00050
11	Yes	f(1, 39) = 12.31	< 0.01	0.00115 to 0.00115
12	Yes	f(1, 39) = 22.53	< 0.01	0.00003 to 0.00003

(36/48)
JPSP_Tamir_2004_87_6_913_Knowing good from.htm

2	Yes	f(1, 71) = 8.21	< 0.01	0.00546 to 0.00549
3	Yes	t(71) = 2.36	= 0.02	0.02077 to 0.02129
5	Yes	f(1, 71) = 7.49	< 0.05	0.00781 to 0.00785
6	Yes	f(1, 71) = 4.22	< 0.05	0.04351 to 0.04375
7	Yes	f(1, 82) = 8.55	< 0.01	0.00445 to 0.00448
8	Yes	f(1, 82) = 7.52	< 0.01	0.00747 to 0.00751
9	Yes	f(1, 82) = 4.47	< 0.05	0.03743 to 0.03764

(37/48)
JPSP_Tazelaar_2004_87_6_845_How to cope.htm
EXCLUDE THIS ARTICLE

(38/48)
JPSP_Thrash_2004_87_6_957_Inspiration. Core characteristics.htm
EXCLUDE THIS ARTICLE

(39/48)
JEP_Tillmann_2004_30_5_1131_Implicit learning of.htm

0	Yes	f(2, 66) = 30.29	< 0.0001	0.00000 to 0.00000	-

1	Yes	f(1, 66) = 19.73	< 0.01	0.00003 to 0.00003	-
2	Yes	f(1, 66) = 10.98	< 0.01	0.00149 to 0.00150	-
3	Yes	f(1, 66) = 30.24	< 0.0001	0.00000 to 0.00000	-
4	Yes	f(1, 66) = 12.99	< 0.001	0.00060 to 0.00060	-
5	Yes	f(1, 66) = 8.61	< 0.01	0.00458 to 0.00461	-
6	Yes	f(1, 66) = 8.92	< 0.01	0.00395 to 0.00397	-
7	Yes	t(11) = 8.0	< 0.001	0.00001 to 0.00001	-
8	Yes	t(11) = 4.45	< 0.001	0.00097 to 0.00099	-
9	Yes	t(11) = -4.0	< 0.01	0.00192 to 0.00227	-
10	Yes	t(11) = 4.04	< 0.01	0.00193 to 0.00197	-
11	Yes	f(1, 44) = 66.02	< 0.0001	0.00000 to 0.00000	-
12	Yes	f(1, 44) = 18.98	< 0.0001	0.00008 to 0.00008	-
13	Yes	f(1, 44) = 9.1	< 0.01	0.00414 to 0.00433	-
14	Yes	f(1, 44) = 19.71	< 0.0001	0.00006 to 0.00006	-
15	Yes	f(1, 44) = 54.98	< 0.0001	0.00000 to 0.00000	-
16	Yes	f(2, 88) = 3.9	< 0.05	0.02276 to 0.02495	-
18	Yes	f(1, 44) = 13.84	< 0.01	0.00056 to 0.00056	-
19	Yes	f(1, 44) = 6.43	< 0.05	0.01481 to 0.01488	-
22	Yes	t(11) = 7.03	< 0.001	0.00002 to 0.00002	-
23	Yes	t(11) = 3.26	< 0.01	0.00753 to 0.00767	-
24	Yes	t(11) = 4.21	< 0.01	0.00145 to 0.00147	-

(40/48)
JEP_Unsworth_2004_30_6_1302_Working memory capacity.htm

1	Yes	f(1, 47) = 32.3	< 0.01	0.00000 to 0.00000
2	Yes	f(1, 47) = 4.14	< 0.05	0.04742 to 0.04767
3	Yes	f(1, 47) = 6.49	< 0.05	0.01414 to 0.01421
4	Yes	f(1, 47) = 6.2	< 0.05	0.01597 to 0.01678
5	Yes	f(1, 47) = 45.98	< 0.01	0.00000 to 0.00000
6	Yes	f(1, 47) = 6.43	< 0.05	0.01457 to 0.01464
7	Yes	f(1, 47) = 6.62	< 0.05	0.01327 to 0.01333
8	Yes	f(1, 47) = 8.39	< 0.01	0.00570 to 0.00572
9	Yes	f(1, 47) = 269.98	< 0.01	0.00000 to 0.00000
10	Yes	f(1, 47) = 4.73	< 0.05	0.03462 to 0.03480
11	Yes	f(1, 47) = 4.14	< 0.05	0.04742 to 0.04767
13	Yes	f(1, 38) = 52.7	< 0.01	0.00000 to 0.00000
15	Yes	f(1, 38) = 7.87	< 0.01	0.00786 to 0.00790
16	Yes	f(1, 38) = 14.49	< 0.01	0.00050 to 0.00050
18	Yes	f(1, 38) = 28.73	< 0.01	0.00000 to 0.00000
22	Yes	f(1, 56) = 47.69	< 0.01	0.00000 to 0.00000
23	Yes	f(1, 56) = 45.81	< 0.01	0.00000 to 0.00000
24	Yes	f(1, 56) = 5.05	< 0.05	0.02851 to 0.02866
25	Yes	f(3, 56) = 7.49	< 0.01	0.00026 to 0.00027
27	Yes	f(3, 56) = 3.66	< 0.05	0.01752 to 0.01772
28	Yes	f(1, 56) = 102.2	< 0.01	0.00000 to 0.00000

29 Yes f(3, 56) = 11.9 < 0.01 0.00000 to 0.00000
30 Yes f(3, 60) = 2.78 < 0.05 0.04842 to 0.04900
31 Yes f(1, 56) = 5.76 < 0.05 0.01969 to 0.01980
32 Yes f(3, 56) = 3.11 < 0.05 0.03328 to 0.03368
33 Yes f(1, 56) = 49.41 < 0.01 0.00000 to 0.00000
34 Yes f(3, 56) = 8.81 < 0.01 0.00007 to 0.00007
35 Yes f(1, 56) = 5.02 < 0.05 0.02896 to 0.02912
37 Yes f(3, 56) = 3.15 < 0.05 0.03176 to 0.03213
38 Yes f(1, 56) = 74.91 < 0.01 0.00000 to 0.00000

(41/48)
JEP_Van Zandt_2004_30_5_1147_Response reversals in.htm
No results found.

(42/48)
PSP_Van Zomeren_2004_87_5_649_Put your money.htm
0 Yes f(1, 80) = 19.07 < 0.01 0.00004 to 0.00004
1 No f(1, 80) = 5.31 < 0.02 0.02373 to 0.02386
Recalculated p-value does not match the reported p-value.
3 Yes f(1, 80) = 7.44 < 0.01 0.00782 to 0.00786
9 Yes f(1, 80) = 25.13 < 0.01 0.00000 to 0.00000
10 No f(1, 80) = 4.81 < 0.03 0.03111 to 0.03128
Recalculated p-value does not match the reported p-value.
12 Yes chi2(13) = 34.12 < 0.01 0.00115 to 0.00116
13 Yes chi2(6) = 23.56 < 0.01 0.00063 to 0.00063
15 Yes f(1, 64) = 205.35 < 0.01 0.00000 to 0.00000
16 Yes f(1, 64) = 14.79 < 0.01 0.00028 to 0.00028
19 Yes f(1, 64) = 13.69 < 0.01 0.00045 to 0.00045
23 No f(1, 64) = 4.88 < 0.03 0.03067 to 0.03084
Recalculated p-value does not match the reported p-value.
28 Yes chi2(6) = 28.02 < 0.01 0.00009 to 0.00009
29 Yes f(1, 87) = 110.44 < 0.01 0.00000 to 0.00000
30 Yes f(1, 87) = 68.3 < 0.01 0.00000 to 0.00000
33 Yes f(1, 87) = 9.82 < 0.01 0.00235 to 0.00236
34 No f(1, 87) = 4.22 < 0.04 0.04283 to 0.04307
Recalculated p-value does not match the reported p-value.
35 Yes f(1, 87) = 5.7 < 0.02 0.01863 to 0.01965
37 Yes f(1, 87) = 7.3 < 0.01 0.00808 to 0.00850
39 Yes f(1, 87) = 4.97 < 0.03 0.02829 to 0.02844

(43/48)
JEP_Verhaeghen_2004_30_6_1322_A working memory.htm
No results found.

(44/48)

```
JPSP_Visser_2004_87_6_779_Attitudes in the.htm
0      Yes       f(1, 46) = 4.58       < 0.04      0.03758 to 0.03778    -
1      Yes      f(1, 58) = 10.19       < 0.01      0.00228 to 0.00229    -
2      Yes       f(1, 46) = 4.71       < 0.05      0.03509 to 0.03528    -
3      Yes       f(1, 58) = 4.59       < 0.05      0.03627 to 0.03647    -
6      Yes       f(1, 77) = 5.29       < 0.03      0.02409 to 0.02422    -
7      Yes          t(77) = 2.18       < 0.04      0.03193 to 0.03270    -
8      Yes     f(1, 77) = 177.17      < 0.001      0.00000 to 0.00000    -
9      Yes       f(1, 37) = 5.47       < 0.05      0.02479 to 0.02492    -
10     Yes        f(1, 38) = 4.5       < 0.05      0.03944 to 0.04154    -

(45/48)
JEP_Ward_2004_30_6_1196_The effect of.htm
0      Yes       f(1, 35) = 98.36      < 0.01       0.00000 to 0.00000
1      Yes       f(4, 140) = 3.64      < 0.01       0.00740 to 0.00752
2      Yes       f(1, 35) = 55.74      < 0.01       0.00000 to 0.00000
4      Yes        f(1, 35) = 9.95      < 0.01       0.00329 to 0.00330
5      Yes        f(1, 35) = 9.52      < 0.01       0.00395 to 0.00397
6      Yes        f(1, 35) = 4.85      < 0.05       0.03424 to 0.03441
7       No        f(1, 35) = 7.23      < 0.01       0.01088 to 0.01093
Recalculated p-value does not match the reported p-value.
8      Yes       f(4, 140) = 12.0      < 0.01       0.00000 to 0.00000
9      Yes        f(1, 35) = 4.85      < 0.05       0.03424 to 0.03441
10     Yes      f(4, 140) = 14.53      < 0.01       0.00000 to 0.00000
11     Yes       f(4, 140) = 2.81      < 0.05       0.02763 to 0.02807
13     Yes      f(4, 140) = 29.56      < 0.01       0.00000 to 0.00000
15     Yes      f(4, 140) = 38.11      < 0.01       0.00000 to 0.00000
18     Yes       f(3, 57) = 10.06      < 0.01       0.00002 to 0.00002
21     Yes          t(16) = 10.36      < 0.01       0.00000 to 0.00000
22     Yes           t(18) = 6.09      < 0.01       0.00001 to 0.00001
23     Yes            t(29) = 3.9      < 0.01       0.00046 to 0.00060
24     Yes            t(17) = 7.0      < 0.01       0.00000 to 0.00000
25     Yes       f(1, 29) = 53.28      < 0.01       0.00000 to 0.00000
26     Yes       f(1, 29) = 49.46      < 0.01       0.00000 to 0.00000
27     Yes       f(1, 35) = 27.61      < 0.01       0.00001 to 0.00001
28     Yes      f(1, 29) = 302.15      < 0.01       0.00000 to 0.00000
29     Yes      f(2, 58) = 395.68      < 0.01       0.00000 to 0.00000
30     Yes      f(2, 58) = 118.96      < 0.01       0.00000 to 0.00000
31     Yes       f(1, 29) = 42.76      < 0.01       0.00000 to 0.00000
32     Yes       f(4, 116) = 3.89      < 0.01       0.00526 to 0.00535
33     Yes        f(1, 29) = 9.74      < 0.01       0.00405 to 0.00407
34     Yes       f(4, 116) = 4.21      < 0.01       0.00319 to 0.00324
35     Yes       f(4, 116) = 5.68      < 0.01       0.00033 to 0.00033
36     Yes     f(1, 29) = 1278.97      < 0.01       0.00000 to 0.00000
37     Yes        f(2, 58) = 21.0      < 0.01       0.00000 to 0.00000
```

38	Yes	f(2, 58) = 9.66	< 0.01	0.00024 to 0.00024
39	Yes	f(2, 58) = 17.69	< 0.01	0.00000 to 0.00000
40	Yes	f(4, 116) = 23.85	< 0.01	0.00000 to 0.00000
41	Yes	f(8, 232) = 3.58	< 0.01	0.00061 to 0.00062
42	Yes	f(2, 58) = 4.65	< 0.05	0.01334 to 0.01345
43	Yes	f(4, 116) = 366.93	< 0.01	0.00000 to 0.00000

(46/48)
JEP_Winman_2004_30_6_1167_Subjective probability intervals.htm

0	Yes	t(38) = 2.35	= 0.024	0.02379 to 0.02435	-
1	Yes	t(38) = 3.3	= 0.002	0.00184 to 0.00242	-
2	Yes	t(19) = 3.48	= 0.002	0.00248 to 0.00254	-
3	Yes	t(19) = 4.96	< 0.01	0.00009 to 0.00009	-
4	Yes	t(43) = 4.65	< 0.01	0.00003 to 0.00003	-
5	Yes	t(28) = 2.51	= 0.018	0.01792 to 0.01834	-
6	Yes	t(28) = 2.07	= 0.047	0.04729 to 0.04829	-
7	Yes	t(28) = 5.76	< 0.01	0.00000 to 0.00000	-

(47/48)
JEP_Yang_2004_30_5_1045_Knowledge partitioning in.htm

0	Yes	f(7, 322) = 42.21	< 0.01	0.00000 to 0.00000	-
3	Yes	f(1, 23) = 5.11	< 0.05	0.03348 to 0.03364	-
4	Yes	f(2, 92) = 48.08	< 0.01	0.00000 to 0.00000	-
5	Yes	f(2, 92) = 10.19	< 0.01	0.00010 to 0.00010	-
6	Yes	f(2, 46) = 15.48	< 0.01	0.00001 to 0.00001	-
7	Yes	f(1, 23) = 12.44	< 0.01	0.00180 to 0.00181	-
8	Yes	f(1, 23) = 16.47	< 0.01	0.00049 to 0.00049	-
9	Yes	f(1, 23) = 6.89	< 0.05	0.01511 to 0.01517	-
10	Yes	f(2, 46) = 10.52	< 0.01	0.00017 to 0.00017	-
11	Yes	f(7, 315) = 24.83	< 0.01	0.00000 to 0.00000	-
12	Yes	f(2, 30) = 9.28	< 0.01	0.00073 to 0.00073	-
13	Yes	f(1, 15) = 8.35	< 0.05	0.01121 to 0.01125	-
14	Yes	f(1, 15) = 8.59	< 0.05	0.01031 to 0.01034	-
15	Yes	f(1, 15) = 36.62	< 0.01	0.00002 to 0.00002	-
16	Yes	f(1, 15) = 52.42	< 0.01	0.00000 to 0.00000	-
17	Yes	f(2, 30) = 8.17	< 0.01	0.00147 to 0.00148	-
18	Yes	f(1, 15) = 8.63	< 0.05	0.01017 to 0.01020	-
19	Yes	f(1, 15) = 7.13	< 0.05	0.01744 to 0.01750	-
20	Yes	f(2, 60) = 6.24	< 0.01	0.00344 to 0.00347	-

(48/48)
JPSP_Jones_2004_87_5_665_How do I.htm

0	Yes	chi2(1) = 16.78	< 0.001	0.00004 to 0.00004
1	Yes	chi2(1) = 16.39	< 0.001	0.00005 to 0.00005
3	Yes	f(1, 63) = 55.62	< 0.001	0.00000 to 0.00000

```
4       Yes     chi2(1) = 2244.3      < 0.001      0.00000 to 0.00000
5       Yes      chi2(1) = 253.1      < 0.001      0.00000 to 0.00000
6       Yes      chi2(1) = 34.54      < 0.001      0.00000 to 0.00000
7       Yes      f(1, 63) = 9.77      = 0.003      0.00268 to 0.00269
8        No     f(4, 15) = 11.01      = 0.005      0.00023 to 0.00023
Recalculated p-value does not match the reported p-value.
10      Yes      chi2(1) = 11.01      < 0.001      0.00090 to 0.00091
11      Yes      chi2(1) = 32.43      < 0.001      0.00000 to 0.00000
12      Yes         t(50) = 2.57       < 0.05      0.01303 to 0.01337
13      Yes        t(108) = 2.04       < 0.05      0.04329 to 0.04430
14      Yes        t(108) = 2.01       < 0.05      0.04639 to 0.04746
17      Yes        t(107) = 8.23      < 0.001      0.00000 to 0.00000
20      Yes        t(108) = 3.64      < 0.001      0.00041 to 0.00043
21      Yes      f(1, 82) = 3.98       < 0.05      0.04923 to 0.04950
22      Yes      f(1, 82) = 6.55       = 0.01      0.01229 to 0.01236
23      Yes      f(1, 82) = 4.03       < 0.05      0.04786 to 0.04813
26      Yes          t(81) = 2.5       < 0.02      0.01266 to 0.01644
27      Yes         t(81) = 2.03       < 0.05      0.04512 to 0.04616
30       No         t(27) = 2.02       < 0.05      0.05286 to 0.05396
Gross inconsistency: reported p-value and recalculated p-value differ in significance.
Consistent for one-tailed, inconsistent for two-tailed
31      Yes         t(20) = 2.14       < 0.05      0.04442 to 0.04532
34      Yes         t(20) = 2.14       < 0.05      0.04442 to 0.04532
```

C Missed Results in Manually Coded Validate File

Appendix C shows a list of NHST results that meet the inclusion criteria mentioned in both subsection 2.1.3 and 4.1, but were not included in the manually coded validate file. These results were successfully detected by the Python script.

```
(23/48)
JPSP_Koole_2004_87_6_974_Getting a grip.htm
21      No       f(1, 53) = 2.91       < 0.05      0.09361 to 0.09416
Gross inconsistency: reported p-value and recalculated p-value differ in significance.

(39/48)
JEP_Tillmann_2004_30_5_1131_Implicit learning of.htm
22     Yes         t(11) = 7.03      < 0.001      0.00002 to 0.00002    -

(42/48)
PSP_Van Zomeren_2004_87_5_649_Put your money.htm
12     Yes      chi2(13) = 34.12      < 0.01       0.00115 to 0.00116
13     Yes       chi2(6) = 23.56      < 0.01       0.00063 to 0.00063
28     Yes       chi2(6) = 28.02      < 0.01       0.00009 to 0.00009
```

D GRIM Script Analysis Data

Appendix D summarises the GRIM test results per article.

Table 12: Comparison of GRIM test results with annotations by Brown & Heathers

Article	# Inconsistent (script)	# Inconsistent (annotated)	# In intersection	Notes
Eskine – A Bad Taste in the Mouth	6	7	0	All inconsistent mean values marked were in a table, which was not extracted
Ma-Kellams – Culturally Divergent Responses to Mortality Salience	10	2	2	-
Gable – Time Flies When You're Having Approach-Motivated Fun	1	1	1	-
Fessler – Friends Shrink Foes	0	8	0	4 percentages and 4 regular misses
Inagaki – Shared Neural Mechanisms Underlying Social Warmth and Physical Warmth	7	12	5	The remaining 7 mean values were considered "consistent" by the script, because the wrong N was extracted for those
Shariff – Free Will and Punishment	1	2	0	2 regular misses
Shaw – Constructing Rich False Memories of Committing Crime	6	15	6	6 inconsistent means in a table detected by the script, but another table with 8 results was missed. 1 regular miss in general text.
Oriña – Developmental and Dyadic Perspectives on Commitment in Adult Romantic Relationships	2	2	2	-
Moser – Mind Your Errors	1	1	1	-

Article	# Inconsistent (script)	# Inconsistent (annotated)	# In intersection	Notes
Pope – Round Numbers as Goals	0	1	0	1 regular miss
Kille – Tall, Dark, and Stable	5	1	1	-
Bilderbeck – Serotonin and Social Norms	3	3	2	1 result in a table missed
Patihis – Are the "Memory Wars" Over? A Scientist-Practitioner Gap in Beliefs About Repressed Memory	0	6	0	6 results in tables missed
Schroeder – The Sound of Intellect	4	0	0	No explicit mention of inconsistent means in annotated article
Greitemeyer – Denying Humanness to Others	10	0	0	-
Sylvers – Psychopathic Traits and Preattentive Threat Processing in Children	0	0	0	-
Birtel – Treating" Prejudice	17	0	0	-
Grant – Beneficiary or Benefactor	11	0	0	-
Kwan – Effects of Symptom Presentation Order on Perceived Disease Risk	0	0	0	-
Radel – Evidence of Motivational Influences in Early Visual Perception	2	0	0	-
Aspell – Turning Body and Self Inside Out	0	0	0	-
Kwang – Men Seek Social Standing, Women Seek Companionship	0	0	0	-

Article	# Inconsistent (script)	# Inconsistent (annotated)	# In intersection	Notes
van Gelder – Vividness of the Future Self Predicts Delinquency	0	0	0	-
Yap – The Ergonomics of Dishonesty	2	0	0	-
Hafenbrack – Debiasing the Mind Through Meditation	0	0	0	-
Noreen – Forgiving You Is Hard, but Forgetting Seems Easy	0	0	0	-
Dufau – A Thousand Words Are Worth a Picture	0	0	0	-
Engelhardt – Effects of Violent-Video-Game Exposure on Aggressive Behavior, Aggressive-Thought Accessibility, and Aggressive Affect Among	4	0	0	-
Hirsh-Pasek – The Contribution of Early Communication Quality to Low-Income Children's Language Success	0	0	0	-
Strohminger – Neurodegeneration and Identity	12	0	0	-
Akrami – Generalized Prejudice	0	0	0	-
Carter – A Single Exposure to the American Flag Shifts Support Toward Republicanism up to 8 Months Later	1	0	0	-

Article	# Inconsistent (script)	# Inconsistent (annotated)	# In intersection	Notes
Evans – Loosening the Link Between Childhood Poverty and Adolescent Smoking and Obesity	0	0	0	-
Imhoff – Facing Europe	0	0	0	-
Ireland – Language Style Matching Predicts Relationship Initiation and Stability	0	0	0	-
Kavanagh – When It's an Error to Mirror	0	0	0	-
Mancini – Visual Distortion of Body Size Modulates Pain Perception	0	0	0	-
Nagengast – Who Took the "×" out of Expectancy-Value Theory?	0	0	0	-
Nisbet – Underestimating Nearby Nature	12	0	0	-
Said – A Statistical Model of Facial Attractiveness	0	0	0	-
TOTAL	**117**	**61**	**20**	**-**

E Prompt for AI-Powered statcheck

Appendix E shows the statcheck related AI prompt used for data extraction.

```
STATCHECK_PROMPT: str = ("""
    You are an AI assistant that extracts statistical test results from scientific text.

    Please extract ALL statistical tests reported in the following text. For each test, extract the following components:

    - test_type: one of 'r', 't', 'f', 'chi2', 'z'.
    - df1: First degree of freedom (float or integer). If not applicable, set to None.
    - df2: Second degree of freedom (float or integer). If not applicable, set to None.
    - test_value: The test statistic value (float).
    - operator: The operator used in the reported p-value ('=', '<', '>').
    - reported_p_value: The numerical value of the reported p-value (float) if available, or 'ns' if reported as not significant.
    - epsilon (float): Only extract when a Huynh-Feldt correction is mentioned. If not applicable, set to None.
    - tail: 'one' or 'two'. Assume 'two' unless explicitly stated.

    Guidelines:

    - Do not extract any tests that does not EXPLICITY mention one of the predetermined test types (e.g., t, r, f, chi2, z).
    - Do not extract test that are incomplete (i.e., the minimal requirements are: test_type, df1, test_value, operator, reported_p_value).
    - IMPORTANT: EXTRACT THE CORRECT OPERATOR FROM THE P-VALUE (E.G., '=', '<', '>').
    - If you are not completely certain that a test meets the minimal requirements, do not extract it.
    - You must never infer or assume test types, degrees of freedom, or test values based on contextual clues, reported means, or p-values.
    - Be tolerant of minor typos or variations in reporting.
    - Recognize tests even if embedded in sentences or non-standard formats.
    - Pay special attention to distinguishing between chi-square tests ('    ', 'chi2') and F-tests.
    - Chi-square tests may also appear as "G-square", "G^2", or "G2". Use 'chi2' as test_type.
    - IMPORTANT: "rho" is not "r". Do not interpret "rho" as "r".
    - Extract both operator and numerical value for p-values using
```

```
      inequality signs.
    - Do not perform any calculations or inferences beyond whats explicitly stated.
    - A test may be split over multiple sentences. Extract correctly and carefully.
    - Treat commas in numbers as thousand separators, not decimal points.
    - For chi2 tests: do not extract the sample size (N).
    - Only F-tests require two degrees of freedom; others use df1 only.
    - Do not extract tests not described in this prompt (e.g., 'B' tests).
    - Only extract an epsilon value if explicitly mentioned AND if a Huynh-Feldt correction was applied.
    - IMPORTANT: EPSILON IS REPORTED AS ((ε)) OR (Epsilon).
    EPSILON IS NOT THE SAME AS ETA ((η)) OR ETA squared ((η²)).
      - EXAMPLE: F(1, 82) = 4.03, p <.05, (η) = .22 is NOT a Huynh-Feldt correction.
      DO NOT EXTRACT EPSILON, BECAUSE THIS IS NOT AN EPSILON VALUE, BUT AN ETA VALUE.
      - YOU NEVER EXTRACT ETA VALUES (η²) OR ETA (η) AS EPSILON.
      ONLY EXTRACT EPSILON VALUES (ε) OR (Epsilon) AS EPSILON!

      - You can also encounter NHST tests reported in a table. In these cases, the reported_p_value is often displayed using a symbol (e.g., * for p < 0.05, ** for p < 0.01, *** for p < 0.001).
      - In these cases, extract p < 0.05 for *, p < 0.01 for **, and p < 0.001 for ***.
        - EXAMPLE: 5.27 (2, 67)** in the column "F" should be extracted as:
        - test_type: "f"
        - df1: 2
        - df2: 67
        - test_value: 5.27
        - operator: "<"
        - reported_p_value: "0.01"
      - BUT, ONLY EXTRACT TESTS THAT HAVE A STAR SYMBOL. DO NOT EXTRACT INCOMPLETE TESTS WITHOUT A STAR SYMBOL, EVEN IF THEY ARE NHST TESTS. THIS IS BECAUSE THERE IS NO WAY TO DETERMINE THE REPORTED P-VALUE WITHOUT A STAR SYMBOL OR WITHOUT THE P-VALUE EXPLICITLY MENTIONED.
        - EXAPMLE: "F(1, 3184) = 2.20" - YOU DO NOT EXTRACT THIS TEST, BECAUSE IT IS INCOMPLETE. IT DOES NOT HAVE A STAR SYMBOL, AND THE OPERATOR IS NOT EXPLICITLY MENTIONED.
        You extract this test as:
        - DO NOT EXTRACT - CONTINUE

      - It is also possible that you enocounter a text that has
```

```
            typesetting issues: characters such as "<", ">", or "=" might
             not be properly extracted. If you encounter a NHST where
            everything is present except the operator, assume the
            operator is "<".
            - EXAMPLE: "F(1, 11) 83.93, p .001" - extract this as:
             - test_type: "f"
             - df1: 1
             - df2: 11
             - test_value: 83.93
             - operator: "<"
             - reported_p_value: "0.001"

            - EXAMPLE: "F(1, 15)
             6.1, p
             .05."
             Extract this as:
             - test_type: "f"
             - df1: 1
             - df2: 15
             - test_value: 6.1
             - operator: "<"
             - reported_p_value: "0.05"

    Format the result EXACTLY like this:

    tests = [
        {"test_type": <test_type>, "df1": <df1>, "df2": <df2>, "
            test_value": <test_value>,
         "operator": <operator>, "reported_p_value": <reported_p_value>,
         "epsilon": <epsilon>, "tail": <tail>}
    ]

    Now, extract the tests from the following text:

    {context}

    After reading the text above, read it again to ensure you understand
        the instructions.
    Then, extract the reported statistical tests as requested.
""")
```

F Prompt for AI-Powered GRIM Test

Appendix F shows the GRIM test related AI prompt used for data extraction.

```
GRIM_PROMPT: str = (
    """
    You are an extraction assistant. Your task is to extract only reported **means and their sample sizes** from the following scientific text. You must follow these rules strictly:

    ---

    **Extract only if ALL of the following are true:**
    - The value is explicitly labelled as a **mean** (e.g.,  M  = ... ,  mean  = ...  ).
    - The mean is clearly based on **integer-valued response data** (e.g., responses on Likert-type scales like 1-5, 1-7, etc.).
    - A specific **sample size (N)** is provided in the same sentence, or in a directly connected clause or phrase.
        - IMPORTANT: A SAMPLE SIZE IS ALWAYS AN INTEGER! DO NOT EXTRACT IF THE SAMPLE SIZE IS NOT AN INTEGER!
    - There is a **clear and direct correlation** between the reported mean and its corresponding sample size   do not guess or assume this link.
    - The mean is usable in the **GRIM test** (i.e., based on whole-number responses + a known sample size).
    - The source of the mean is explicitly mentioned (e.g., "mean of Likert-scale responses", "mean of 7-point scale", "mean survey response").
    - Only state that a Likert-scale was used if you see the word "Likert" or "scale" in the context!
      If you do not see either of these words, you may not assume that a Likert-scale was used!
      If this is not clear and there is no other indication that a mean is GRIM-applicable, do not extract it!

    - It is ONLY OKAY to derive sample sizes from other statistics (e.g., t-tests, ANOVA), if the sample size is not clearly mentioned,
      BUT ONLY IF: it is clear that the mean value is derived from DISCRETE INTEGER-BASED data.
    - t(23) can imply N=24. Keep in mind that for a t-test, the sample size is N = df + 1.
    - f(1, 60) can imply a total of N=62, but two groups of N=31 each.
      For an ANOVA, the sample size is N = df + k, where k is the number of groups.
      So ALWAYS look for the number of groups when you encounter ANOVA.
    - IMPORTANT: When you encounter an ANOVA, check the first degree of freedom (df1) and the second degree of freedom (df2).
```

```
    The first degree of freedom is the number of groups minus 1, and the second degree of freedom is the total sample size minus the number of groups.
    So if you see f(1, 60), it means that there are 2 groups (1 + 1) and a total sample size of 62 (60 + 2).
    So never assume the second degree of freedom + 1 is the sample size. Always check df1 to see how many groups there are!

- In your discrete_reasoning, only state that a Likert-scale was used if you see the word "Likert" or "scale" in the context!

---

**NEVER extract if ANY of the following are true:**
- The sample size is **not clearly linked** to the mean, or could refer to a different statistic or part of the study.
- It is a **median**, **mode**, **mean difference**, or **range**.
- It refers to **completion time**, percentages, or **continuous data** (e.g., durations, reaction times).
- It is a **statistical test value**: t, F, p, r, z,    , etc.
- The underlying response scale is not stated as **integer-based** or is ambiguous.

---

Additional rules:
- If the total sample is split into groups (e.g., experimental/control), extract group-level means and sample sizes separately.
- NEVER round mean values    extract them **exactly as reported**, preserving **all decimal places and trailing zeros** (e.g., keep `6.60`, not `6.6`).
- Do **not** perform any calculations. Only extract what is explicitly stated in the text.

---

IMPORTANT: The output must be a JSON-like list of dictionaries, formatted as follows:
YOU ARE NEVER ALLOWED TO CHANGE THIS FORMAT!
Output format:

tests = [
    {{
        "reported_mean": <mean>,
        "sample_size": <sample_size>,
        "discrete_reasoning": "<Why this mean is valid for GRIM (e.g., 'mean of 7-point Likert responses clearly linked to N = 28 in same sentence')>"
```

```
        }},
        ...
    ]

    ---

    Text:
    {context}

    Only return the list of tests. Do not explain anything else. Be
       strict, and only extract what is 100% valid under the criteria
       above.
    """
)
```