# Research Audit for the Transparent Psi Project (TPP)

James E. Kennedy
March 19, 2023 — Final
See copyright notice at the end.

## Table of Contents

# 1. Background and Scope of the Audit

The Transparent Psi Project (TPP) was intended as a potential model and learning experience for optimal research practices. The TPP is, to my knowledge, the only research study in psychology that was designed to use methods that are comparable to my experience in regulated clinical research. These methods include formal software validation, measures to prevent experimenter fraud, developing the operating characteristics for Bayesian analyses, and research audits. TPP will hopefully become a useful model for high quality research methodology.

The purpose of this audit is to apply to the TPP my experience with research audits in regulated clinical research. I initiated the audit after reviewing the final reports of previous audits of TPP by an IT auditor and two research auditors, all with academic backgrounds. In reviewing their final reports, it was clear that their expectations were very different than mine. The previous audits did not have the range of topics, details, emphasis on protocol deviations, and edginess consistent with my expectations for an audit. My hope is to provide an audit that demonstrates useful expectations based on experience outside of academic research.

Information about my background may be useful in putting this audit on context. From 1974 to 1979 I worked at the Institute for Parapsychology, with a heavy emphasis on statistics and methodology. Six months into that position, I discovered that the Director who had hired me was fraudulently manipulating the experimental results. After exposing his fraud (which was not easy, as described in Kennedy, 2017), many months were spent investigating the extent of the fraud. In 1982 I obtained an MSPH (Public Health), including courses in biostatistics. From 1982 to 1990 I did environmental work that involved integrating science, engineering, and law. Knowledge of data analysis was useful, including in some legal actions and helping academic scientists understand the fundamental differences between academic science and law.

In 1991 I shifted careers and began doing data analysis for academic medical research at Duke University. In 1995 I switched to analyzing data for clinical trials for pharmaceutical research. Much to my surprise (and still not recognized by most academic psychologists), the research methodology in regulated clinical research was much better than in academic research. I worked in regulated medical research at four different organizations until retiring in 2011. My later positions were managing the software infrastructure for data management and analyses for regulated clinical trials. This included being the first person that a regulatory auditor wanted to interview.

One major lesson from this diverse experience was that the easy, tempting fraud that was exposed in parapsychology and would be possible in most academic research could not happen in the environment of constant double and triple checking in regulated clinical research. I much preferred an environment that prevented fraud than one that ignored the possibility of fraud or had the high stress and lost productivity of exposing fraud.

In 2012 Caroline Watt and I began operating the KPU Study Registry. My main role is to review submitted registrations to assure that they provide adequate information. I have also written papers about methodological problems that are still not widely recognized in psychology (e.g., Kennedy, 2016)

**Scope of the Audit**

The goals of a good academic research audit have two components: (a) verify that recognized good research practices were used, and (b) verify that the actions and practices specified in the preregistration and/or protocol were properly implemented. The topics described in this audit reflect my views about good research practices.

An audit in regulated clinical research has another major component that unfortunately cannot be expected for academic research. Regulatory guidelines for clinical trials define an audit as:

> A systematic and independent examination of trial-related activities and documents to determine whether the evaluated trial-related activities were conducted, and the data were recorded, analyzed, and accurately reported according to the protocol, sponsor's standard operating procedures (SOPs), good clinical practice (GCP), and the applicable regulatory requirement(s). (US FDA, 2018, page 3)

Any organization that has a role in clinical data collection, data processing, or data analysis is expected to have written SOPs that implement good research practices (US FDA, 2007, 2018). SOPs cover everything that involves the data including setting up workstations and servers, access permissions, software documentation, software validation, audit trails, user training, database locks, and much more. An auditor will review the SOPs to verify good practices and also verify that key steps, such as software validation, were done in accordance with the SOPs. Such SOPs are typically far outside the thought process and practices of academic researchers—but would be very valuable for implementing consistently good research practices.

Good research practices include documentation. The working assumption in an audit is that if a needed practice is not documented, it was not done. Also, any seemingly minor, easily overlooked discrepancy found in an audit should be pursued because such discrepancies sometimes are a manifestation of significantly deficient research practices. An audit report should describe all the topics that were considered, including those that were well handled, as well as the problematic cases.

Good research practices should make it very difficult or impossible for one person acting alone to unintentionally or intentionally (fraudulently) bias the data with little chance of detection. This standard is routinely implemented in typical double-blind, randomized clinical trials. The data collection and analysis software and the data are validated and in a fixed state (with the database formally locked and unchangeable) before the study is unblinded. In addition, expected quality control practices in the research culture include documented independent checks of all data collection, processing, and analysis steps.

For comparison, academic research is typically unblind and often has situations in which fraud by one person acting alone would be easy and tempting with little chance of detection. Like others who have written about experimenter fraud, I think that academic fraud occurs much more frequently than is detected. It is well established that replication and peer review are generally ineffective at detecting or deterring experimenter fraud, and that arguments based on assumptions about rational behavior are not applicable (Kennedy, 2017).

Fortunately, methods to prevent fraud and unintentional errors are relatively easy to implement once they become research habits and an expected part of the research culture. These methods should be considered basic quality control measures.

When a regulatory auditor asks the carefully worded question "how do you know that the programmer did not change the data?" it does not matter whether a change was intentional or unintentional. Documented evidence that the data were not changed is expected.

Some of the routine reliability measures in regulated medical research may not be needed in academic research. One notable example is independent validation of data analysis software. Avoiding analysis errors is very important in medical research. However, a common strategy in academic research at present is to make the analysis scripts publicly available as part of open science practices and assume that others will verify the validity of the analysis.  Analysis errors are primarily a matter of personal embarrassment and credibility in this situation, not life or death with potential class action lawsuits. Therefore, a lack of documented validation for analysis software is not treated as a serious problem in this audit.

However, the validation of data collection software is important because data collection errors cannot be fixed after the fact.

Regulated clinical research involves more than audits. Key steps are (a) review, discussion, and agreement with the regulatory agency about the protocol before the study begins, (b) audits of key research processes after the studies have been completed, and (c) the regulatory agency receives copies of the data and conducts their own analyses. Registered reports in psychology may have steps analogous to (a) and possibly (c) above, but systematic standards have not been developed and most psychological studies do not use the registered reports process. Audits that include aspects of study design and analysis will often be appropriate for academic research. Analysis of data in an academic audit could be considered optional.

This audit was begun on January 22, 2023. At that time data collection and the primary analyses for the study were complete.  The report of the study had been accepted for publication as a Registered Report in *Royal Society Open Science*. The stage 2 preprint was available. The final report was posted during the audit on February 1, 2023 (https://doi.org/10.1098/rsos.191375). Thus, the audit was completed after the final report was published.

The audit was done remotely, using information from:

- the preregistration at https://osf.io/a6ew3
- the published report posted at https://doi.org/10.1098/rsos.191375
- the OSF project website https://osf.io/jk2zf/, specifically https://osf.io/3e9rg/,
- the associated GitLab software website https://gitlab.com/gyorgypakozdi/psi,
- the associated GitHub data website https://github.com/kekecsz/transparent-psi-results/tree/master/live_data, and
- information obtained in emails from the principle investigator (PI) Zoltán Kekecs, including copies of files that are listed as available on OSF but with OSF links rarely functioning.

As with a regulatory audit, an initial draft report was prepared and shared with the organization being audited. They had an opportunity to correct any misunderstandings and to add and clarify information. The PI pointed out several clarifications and appeared to have read the draft audit report carefully. This audit was conducted voluntarily, with no compensation.

### Keep in Mind

This study attempted to implement research practices that are unusual for psychological research. Some people were performing roles that they had never done before and without any precedent or guidelines within their work experience. It is a significant learning experience on many fronts.

This audit points out certain deficiencies. Those should not be taken as personal criticisms in this situation. The challenge is to think about how research practices could be developed that avoid the deficiencies.

### References

Kennedy, J.E. (2017). Experimenter fraud: What are appropriate methodological standards? *Journal of Parapsychology*, 81, 63-72. https://jeksite.org/psi/jp17.pdf

Kennedy, J.E. (2016). Is the methodological revolution in psychology over or just beginning? *Journal of Parapsychology*, 80, 56-68. https://jeksite.org/psi/jp16.pdf

US FDA (2007). Guidance for Industry: *Computerized Systems Used in Clinical Investigations*. https://www.fda.gov/media/70970/download

US FDA (2018). Guidance for Industry: *E6(R2) Good Clinical Practice: Integrated Addendum to ICH E6(R1)*. https://www.fda.gov/media/93884/download

Lessons and recommendations from this audit are discussed at https://jeksite.org/psi/tpp_audit_lessons.pdf.

# 2. Summary

The evaluations of the protocol, preregistration, and study plans were all rated **very good**, except that consistency with previous research was rated **good** and could have been better.

The documentation related to the participants and experimenters were rated **very good**. The documentation related to the computer system and software were mixed with **not adequate** ratings for monitoring the data collection server, tracking modifications to the data collection software, documentation of data analysis code, and documentation of needed validation when the data collection software was installed on a different server.

The handling of consent, disclosures, and privacy for the participants and experimenters were all rated **very good**.

The handling of modifications to the protocol and preregistration were rated as **minimally adequate**. A small amount of test data were included with the "live" study data and were removed with data analysis programming code that was changed after preregistration without appropriate documentation. There also was no documentation about why, when, and by whom test data were collected during the live data collection.

The handling of protocol deviations was rated as **adequate**. Significant protocol deviations included: (a) the server did not track accesses and unexpected events, (b) the GitLab software repository was not synced with the application server, and (c) auditors were not independent of the laboratories in the study. These topics were noted in the published report, but not explicitly (transparently) described as protocol deviations. These deviations relate to other inadequacies in the management of the computer systems.

Given the automated data collection and the security features in the study design, the data collection software programming is the only feasible possibility for undetectable unintentional or intentional data alterations in favor of ESP. Key aspects of the management of the data collection server and software were rated as **not adequate**, and when combined, represent serious deficiency. However, several factors make the possibility of undetectable data alterations from the programming unlikely. These include: (a) the study outcome of strong support for the null model implies that any errors or fraud would have canceled a true precognitive effect, which would be difficult to achieve and involve unlikely motivations for a study like this, (b) the pre-study validation of the data collection software would detect unintentional or intentional programming errors, and (c) the hiring of an IT auditor would provide a strong disincentive for programming fraud.

However, **if the study would have found evidence for ESP, concerns about the protocol deviations and inadequate computer system management would have been much greater**. The probability of programming bias with the inadequately managed computer system would then be balanced against the probability of ESP.

The consistency between the preregistration and the published report was **very good**, except that discussing protocol deviations was **adequate.**

# 3. Protocol, Preregistration, and Study Plans

**Questions**

A. Does the study reasonably implement and test the theoretical hypothesis?
Evaluation: **Very good**.

B. Is the study reasonably consistent with the previous research being confirmed?
Evaluation: **Good, but could have been better.**

C. Does the preregistration eliminate researcher flexibility to adapt or bias the confirmatory results?
Evaluation: **Very good.**

D. Are the planned statistical analyses appropriate?
Evaluation: **Very good.**

E. Is the study vulnerable to experimenter expectancy effects or other biases in the performance of the participants?
Evaluation: **Very Good.**

These topics are optimally reviewed during the study design. However, they are also appropriately reviewed after data collection when information about actual study conduct and outcome can be incorporated. Some of the topics are beyond a typical research audit and could be addressed in a review for publication—but they are also appropriate for a thorough audit of academic psychological research.

**Evaluations**

***A. Does the study reasonably implement and test the theoretical hypothesis?***

The theoretical hypothesis is that a person can predict future random stimuli. The study procedure directly implements and tests that hypothesis, with no ambiguity about the procedure or measures. Evaluation: very good.

***B. Is the study reasonably consistent with the previous research being confirmed?***

The usual assumption for parapsychological experiments is that ESP is a widespread human ability. The initial studies by Daryl Bem (2011) were based on this assumption and the reported results were consistent with it. A subsequent meta-analysis (Bem et al., 2016) similarly reported evidence for the effect. However, very few of the studies were formal preregistered confirmatory research.

Post hoc speculations about failed replications in psychological and parapsychological research often propose that an effect is precariously dependent on participant populations, cultural conditions, experimental environment, etc. However, if researchers are unable to identify and control research conditions to obtain reasonably reliable effects in formal preregistered confirmatory research, the evidence for the effect is unconvincing.

The design of this experiment appears to be intended to duplicate the conditions of the initial experiments to the extent possible for a multi-center study conducted about a decade or more later. This included a consensus design process and substantial effort to train the experimenters, including submitting videos of simulated conduct of the study. For consistency with Bem's studies, this study administered questionnaires that were used by Bem even though the questionnaires were not needed in this study. At the same time, there were differences from the original studies, including different cultures, different stimuli, and different testing conditions.

One of the most notable differences from previous studies is that participants were often tested in groups rather than individually. This could introduce distractions and/or self-consciousness, particularly given the sexual context of the task. According to the lab notebooks (described in the next section), about 28% of the participants were tested alone, and about 51% were tested in groups of 4 to 14 participants.

The instructions to experimenters (https://osf.io/6uan5) specified that dividers prevent viewing another participant's computer screen, but did not suggest keeping the participants separated for privacy to the extent possible. In looking at the trial videos by the experimenters for the three labs that contributed the largest amounts of data (about 68%), it appeared there was little effort to separate the participants, even when greater separation and privacy appeared to be possible. I had expected more effort to maintain separation and privacy in the study, including data collection computers on opposite sides of a room and facing in different directions.

The published report does not provide information about the number and size of testing groups, or analyses to show that ESP scoring was unrelated to the size of the testing group. Given the overall slightly negative hit rate for the ESP test, it is unlikely that participants tested alone (or any other subset of the data) had a positive effect that was diluted by other chance data. However, this difference in study design from previous research is noteworthy and could be empirically evaluated, as was done for experimenter effects described below.

For my own interest and as an optional part of this audit, I did a simple evaluation of the relationship between ESP performance and group testing. The raw data are records for one trial in chronological order of generation by any participant. When participants are tested in a group, the output records for their trials are intermingled. Counting the number of other participants who contributed trials in the database for the same experimenter during the test session for each participant gives an indication of which participants were tested alone and which were in groups. No suggestion of an effect was found for a simple regression predicting ESP hit rate from the number of other participants being tested at the same time, or for a t-test comparing hit rates for those with zero others being tested versus four or more others (Welch t = -.085, df = 1476.6, 710 tested alone, 772 tested in groups with four or more others, p-value = 0.93 two-sided, participants with 24 or more completed trials, includes data collected after stopping criterion was reached). The unit of analysis was participant for these analyses.

Overall, this experiment appears to be an acceptably close replication effort given the need for a multi-center confirmatory study. If the effects reported by Bem are a widespread, reasonably robust human ability, this study should detect them. It appears that group testing could have been handled better in both the conduct and reporting of the study, but that did not affect the study conclusions.
Evaluation: good, but could have been better.

### *C. Does the preregistration eliminate researcher flexibility to adapt or bias the confirmatory results?*

The table below summarizes possible sources of flexibility or bias and how those were handled in the preregistration (https://osf.io/a6ew3). The handling of potential bias was optimal (beyond good) in all cases. Finding the preregistered analysis code was time consuming due to OSF's counterintuitive processes. The analysis code is at https://osf.io/v2nm6/files/osfstorage.
Evaluation: very good.

| **Source of potential flexibility and bias** | **What was specified in the preregistration** |
|---|---|
| Data collection start and stop for inclusion | Start and stop for included data specified. Stop based on number of trials in sequential analyses. |
| Handling incomplete data and dropouts | All completed erotic trials before the stopping criteria will be included, including when a session is not completed. |
| Data exclusions | All completed trials included. |
| Data processing adjustments and transformations | No data processing adjustments or transformations were specified. |
| Exploratory versus confirmatory | Confirmatory and pre-planned exploratory analyses clearly specified. |
| Statistical analysis | Analysis scripts included in preregistration. |
| Criteria for acceptable evidence for confirmatory analysis | Specific statistical tests, direction of effect, and criteria specified based on magnitude of Bayes factors (25) and confidence interval of a mixed model logistic regression. |
| Ambiguous results from low power | High power (>.95) for both null and alternative models. Very low probability of inconclusive or ambiguous outcome. |
| Interpretation of results | Interpretation of different possible outcomes included in preregistration. |

### *D. Are the planned statistical analyses appropriate?*

The study plans included simulations to evaluate the operating characteristics (inferential error rates) of the planned confirmatory analyses. This provides high confidence in the planned analyses and in the researchers' understanding of the planned analyses. The simulations also neutralize most statistical debates. The simulations included the possibility that an effect may be produced by a minority of participants rather than the usual statistical assumption that all participants contribute equally.

The study plans for making inferences required that four statistical tests all support either the null or the alternative models. These tests included three Bayesian analyses with different prior probability distributions and a frequentist mixed model. The operating characteristics simulations verified that this analysis plan was feasible with high power.
Evaluation: very good.

### *E. Is the study vulnerable to experimenter expectancy effects or other biases in the performance of the participants?*

Traditionally in parapsychology, the experimenter's positive attitude about ESP and personal enthusiasm have been considered important (Palmer & Millar, 2015; Schmeidler, 1997). The experimenter is assumed to create positive expectations and to motivate the participants to perform well. Negative results by skeptics are proposed to be due to the experimenter communicating skeptical expectations.

TPP attempted to make the experimenters' interactions with the participants as uniform as possible. Detailed instructions were provided for the experimenters (https://osf.io/6uan5; https://osf.io/uarfx; https://osf.io/9xwah). The experimenters were required to submit a video of a simulated test session, that included addressing questions and creating optimistic expectations. The lead investigators reviewed the videos and only approved data collection after they were satisfied that an experimenter was adequately trained and did not communicate negative expectations.

The experimenters who interacted with the participants and the senior researchers at the lab also completed a sheep-goat questionnaire. The traditional assumption that experimenter attitude is important was evaluated with a post hoc analysis that found no relationship between experimenter sheep-goat scores and ESP performance. The published analysis appears to me to be not optimal because it included a separate intercept for each experimenter as well as the sheep-goat score. A sheep-goat effect could be spread across the slope and intercept parameters, with low power to detect an effect. However, I did a simple linear model analysis that similarly showed nothing remotely suggestive of an effect.
Evaluation: very good.

## References

Bem, D. (2011). Feeling the future: Experimental evidence for anomalous retroactive influenceson cognition and affect. *Journal of Personality and Social Psychology*, 100(3), 407–425. https://www.apa.org/pubs/journals/features/psp-a0021524.pdf

Bem, D., Tressoldi, P., Rabeyron, T., & Duggan, M. (2016). Feeling the future: A meta-analysis of 90 experiments on the anomalous anticipation of random future events. *F1000Research*, https://doi.org/10.12688%2Ff100research.7177.2

Palmer, J., & Millar, B. (2015). Experimenter effects in parapsychological research. In Etzel Cardeña, John Palmer, and David Marcusson-Clavertz (Eds.) *Parapsychology: A handbook for the 21st century*. Jefferson NC: McFarland.

Schmeidler, G.R. (1997). Psi-Conducive Experimenters and Psi-Permissive Ones. *European Journal of Parapsychology*, 13, 83-94. https://jeksite.org/others/gs1997ejp.pdf

# 4. Documentation During Study Conduct

**Questions**

Were appropriate documentation and records kept for:

A. experimenter selection, experimenter conduct, and laboratory processes;
Evaluation: **Very Good.**

B. individual participant records;
Evaluation: **Very Good.**

C. initial validation of data collection software;
Evaluation of initial validation using first server: **Good.**
Evaluation of validation using second server: **Not adequate.**

D. security and access to the data collection server;
Evaluation: **Good.**

E. unexpected events, and troubleshooting of the data collection server and software;
Evaluation: **Not adequate.**

F. tracking modifications to the data collection software and verification that unauthorized modifications did not occur;
Evaluation: **Not adequate**.

G. documentation of data analysis code.
Evaluation: **Not adequate.**

This section considers only the completeness and clarity of the documentation. The implications of the documented information, such as protocol deviations, are discussed in other sections.

**Evaluations**

***A. Experimenter selection, experimenter conduct, and laboratory processes.***

The instructions for the experimenters were clear and detailed (https://osf.io/6uan5; https://osf.io/uarfx; https://osf.io/9xwah). The use of trial videos of the experimenters doing a mock session seem very useful, including showing the facilities at each site as well as the performance of the experimenters. That fact that over half of the experimenters had to submit a second video after more practice indicates the value of the videos.
Evaluation: very good.

***B. Individual participant records.***

The experimenters filled out an online lab notebook form for each test session. The sessions included 1 to 14 participants tested at one time. The notebooks or forms included checklists and questions about possible protocol deviations and computer problems. The information in the notebooks was put in a spreadsheet and made publicly available as part of the study materials (https://osf.io/myjsw).

The lab notebooks accounted for 2155 participants. The published report indicates that data were collected for a total and 2220 participants, with data from 2115 used in the final report. Participants who started after the study stop criterion had been reached were excluded. The discrepancies were probably due to two factors. In some cases, computer problems caused the session for a participant to be restarted, which could appear as two participants in the database. Also, participants were tested in groups of up to 14. It is likely that some miscounts occurred in the lab notebook. It is also possible that the notebook was not filled out for a session.
Evaluation: very good.

### *C. Initial validation of data collection software.*

The first report by the IT auditor (https://osf.io/dx4nw) described (a) finding a significant programming error, (b) several recommended programming improvements, and (c) a list of various testing that was done and did not find problems. The second report (https://osf.io/ex56a) reviewed and accepted the resulting programming changes. Ideally, more details could have been provided about certain tests, such as the multi-user tests, and information about the dates of the tests might have been useful.
Evaluation: good.

However, as noted in section 8 below, the initial server that was validated was not used for the data collection in the study. No software validation with the second server is reported.
Evaluation of documentation for validation with second server: not adequate.

### *D. Security and access to the data collection server.*

The final report by the IT auditor (https://osf.io/p62gw) described (a) the failure of the server operating system to keep a log of user accesses due to an apparent error in a configuration file, and (b) placing the data collection application on the root of the server means that anyone with access to the application has the capability to make changes to the application files that would not be detected with operating system processes. The implications of the deficiencies could have been described more clearly for nontechnical readers.
Evaluation: good.

### *E. Unexpected events, and troubleshooting of the data collection server and software.*

The laboratory notebooks (https://osf.io/myjsw) reported about 50 cases of computer problems, including crashes, timeouts, and program restarts. Instances of computer problems were reported at 9 of the 10 laboratories. In some cases the test session for the participant was terminated and in other cases the session was started over. The database has one participant_ID code (hashed as 6fca1daa-de1b-4ce6-8ea0-02525d75d65e) that is used for two different people, one male and one female.

The computer problems usually occurred for just one participant when a group of participants were tested. This suggests that the problems may be related to unanticipated actions by a participant, rather than general internet connectivity problems. Problems after hitting browser refresh were noted. My experiments with the demo version of the software found that the session starts over if the browser refresh is clicked. The PI stated that starting the experiment over was the expected behavior for a browser refresh and that the participants should not press the refresh button during the experiment. A warning about the browser refresh was not described in the written instructions for the experimenters.

The problem cases reported in the notebooks could not be checked on the server because the incorrect configuration of the server did not keep the expected logs of unexpected events on the server. The IT auditor's reports did not mention the reported computer problems.
Evaluation: not adequate.

### *F. Tracking modifications to the data collection software and verification that unauthorized modifications did not occur.*

As described in section 8 below, the preregistration stated that a GitLab repository for the data collection software would be continuously synced with the application server and allow the auditors to check the state of the software at any time during the study. However, in a significant protocol deviation, the GitLab repository was not synced with the application server. There is no indication that the IT auditor directly accessed the application server and checked the software during data collection.

After data collection was complete, the IT auditor did a retrospective evaluation of the application server. As described in section 8 below, the final report by the IT auditor (https://osf.io/p62gw) did not discuss the potential for threats due to file changes that were not tracked by Git and how that might be related to the inappropriate root and system access of the programmers and lack of tracking of server accesses and unexpected events.
Evaluation: not adequate.

### *G. Documentation of data analysis code.*

Different versions of files with data analysis code are available on OSF. The files do not have internal documentation that describes how they differ from other files with the same name (https://osf.io/v2nm6/files/osfstorage; https://osf.io/skqvt; https://osf.io/x59vm). The OSF audit trail links for changes to the files gave error messages and were unusable. I was only able to figure out what changes were made by downloading different files and comparing the contents with a text editor. Notably, program changes after preregistration included adding two codes for test lab ids to be excluded from the analyses. The PI pointed out that the addition of the two test lab ids was done on January 24, 2020, soon after data collection started on January 14, and the change was tracked in the GitHub software repository at https://github.com/kekecsz/Transparent_psi_RR_materials/commit/4e4e2e4df561205fc707837add79c9e5bcbc4eb3. This change did not have an explanation in GitHub as is good practice.

A file with data analysis code that will be made available for research transparency and/or for research audits should have standalone internal documentation. The code should be prepared with the expectation that the file will be widely distributed without the context of version control in a repository or a folder structure that identifies different stages.

The documentation should include the purpose of the code, the names of the programmers who developed the code, the date of last modification, and a brief change log that lists changes after the code initially was put to significant use, such preregistered or used to draw inferences about data. A copy of each version that had significant use should be kept. Adding a sequential number to the file name makes the sequence of revisions very clear, particularly when no version control system is used or a version control system is unreliable, as appears to be the case with OSF.

Additional comments about the documentation of the analysis code is given in section 6 below.

Evaluation: not adequate.

# 5. Consent, Disclosures, and Privacy

**Questions**

A. Were disclosures and consent handled appropriately for the participants?
Evaluation: **Very Good**.

B. Were appropriate measures taken for participant privacy?
Evaluation: **Very Good**.

C. Were disclosures, consent, and privacy handled appropriately for the experimenters?
Evaluation: **Very Good**.

**Evaluations**

***A. Were disclosures and consent handled appropriately for the participant?***

The experimenters were trained to disclose to the participant the nature of the experiment and the experimental task, and to give the participant opportunities to withdraw. The data collection software also provided similar information with multiple opportunities to withdraw, including after seeing an example of an explicitly sexual image that would be displayed in the study. The software required consent to continue. The experimenters were also trained to respond to most questions by participants and to debrief the participants.
Evaluation: very good.

***B. Were appropriate measures taken for participant privacy?***

Privacy was a significant factor because the participants were asked to enter their sexual orientation into the computer. The experimenters were trained to avoid looking at the computer screens for the participants to protect their privacy. The data files that were made public had date-time, experimenter, and site fields hashed. Data was also uploaded in batches so that individual participants could not be identified.
Evaluation: very good.

***C. Were disclosures, consent, and privacy handled appropriately for the experimenters?***

The site-PI and individual experimenters filled out a consent form (https://osf.io/b46wa) that included information about what would happen with the information they provided. The site-PI had the option of remaining anonymous, which one site chose. The individual experimenters also had additional disclosures on the instructions for making videos about the trial sessions (https://osf.io/uarfx).
Evaluation: very good.

# 6. Modifications to the Protocol and Preregistration

**Question**

Were all modifications to the protocol and/or preregistration appropriately documented and explained?

Evaluation: **Minimally Adequate.**

**Evaluation**

The published report and a document on the OSF project website (https://osf.io/45e82) identified "two minor" changes from the preregistered protocol. One was a correction of the preregistered analysis code for a non-crucial statistic about side preferences. The other modification was that the planned video of the experimenter describing the test was based on two mock participants. Due to the Covid 19 pandemic, some sites were limited to two people in a room. The video process was modified to allow only one mock participant in these cases.

However, two other changes were made to the final analysis code that were not in the preregistration. One change was modifications to Figure 2 to make it more clear. This change was clearly documented in the analysis code.

The other change was to exclude two lab ids from the analyses. This change was not described in the change document file above or in the analysis code file. It was found by comparing the preregistered version of the file (https://osf.io/v2nm6/files/osfstorage) with a later version of the file with the same name (https://osf.io/skqvt) when trying to figure out what changes had been made to analysis code files with the same name. Changing the data exclusion criteria directly affects the confirmatory analysis and is more significant than the change about side preferences documented above.

The data collection software had three categories for data collection: live, pilot, and test. Live was the category for data included in the study. However, some of the laboratory IDs apparently were used for testing with the live data. Testing the live data collection would be expected during the software validation before the study began. Testing live data collection during the study could be appropriate, but must be documented (a) to avoid retrospective decisions that certain data were test cases, (b) to avoid making errors in the process of excluding data, and (c) to explain fully all data exclusions.

The published study report stated (https://doi.org/10.1098/rsos.191375):

> All data were entered into the data analysis that is collected during the main study (not the pilot study), except for data generated during system tests. (The experimenter ID(s) of the test account(s) were specified in the preregistered analysis code.) The pilot data were not combined with the data collected in the main experiment and were not used in the confirmatory analyses. No other data were excluded from analysis for any reason. (page 13)

However, the final analysis code used for publication excluded two lab IDs that were added after preregistration. The preregistered analysis code excluded two lab IDs as test cases. A comment in the final analysis code where the four lab IDs were excluded said that "(these IDs were preregistered to be excluded)" (https://osf.io/jdukb). That comment and the published description are misleading because two lab ID codes were added to the exclusions after preregistration.

The PI pointed out that the addition of the two test lab ids was done on January 24, 2020, shortly after data collection started on January 14, and the change was tracked in the GitHub software repository at https://github.com/kekecsz/Transparent_psi_RR_materials/commit/4e4e2e4df561205fc707837add79c9e5bcbc4eb3. Thus, data had been collected when the change was made.

Verifying that the excluded data were actually test cases was not possible in this audit because the date-time, lab IDs, etc. are hashed to prevent identification. The excluded data for the two added lab IDs were only 81 records widely dispersed throughout the study and could not affect the study conclusions. The records appear consistent with test data, although it is not clear why such testing was done or who did it.

Good research practice would have been to clearly document at the time the initial testing occurred who, when, and why testing was done in the live data for the study and what lab ID codes would be removed from the data as test cases. The documentation in this study was not adequate, but the added exclusions do not impact the study outcome. Evaluation: minimally adequate.

*Note.*

Making some software changes for the final analysis compared to the preregistered code is acceptable if the changes are documented in the analysis code and are consistent with the writings and publications about the project.

# 7. Protocol Deviations

**Question**

Were all significant protocol deviations appropriately identified, documented, and explained?

Evaluation: **Adequate**

**Evaluation**

Optimal practice would be to acknowledge and explain significant protocol deviations in the report of the study or in the supplemental materials. A less transparent but acceptable alternative may be to discuss the methodological weakness without explicitly using the term protocol deviation.

***Potentially Significant Protocol Deviations***

*Protocol deviation: Server did not track accesses and unexpected events.*

As described in section 8 below, an apparent error in configuring the server resulted in the server operating system not keeping a log of accesses to the server and unexpected events on the server. This prevented the IT auditor from checking the server logs as specified in the preregistration. The published report on page 21, described this problem without specifically identifying it as a protocol deviation. The report said that “the IT auditor noted that due to the redundancies in verifying the integrity of the experimental software and data, it is still very unlikely that there would have been any undetected modifications in the experimental software and the data.” The present audit agrees with that conclusion, but for different reasons. The published report is consistent with the information available at the time it was written.

Evaluation: adequate.

*Protocol deviation: GitLab software repository not synced with application server.*

As described in section 8 below, the GitLab software repository was not synced with the application server as specified in the preregistration. The synced GitLab repository was intended to be used for software monitoring and verification that unauthorized changes were not made to the software after initial software validation. The lack of syncing was discovered as part of this audit at about the time the published study report went live online. Two places in the report needed to be modified. One change was made just before the report became live. The PI indicated that he would change the other place (on page 24). The situation is not identified as a clear protocol deviation. As described in section 8 below, mitigating factors make potential bias from these inadequate practices very unlikely.

Evaluation: minimally adequate.

*Protocol deviation: Auditors not independent of the laboratories in the study.*

The preregistration (https://osf.io/a6ew3) stated on page 15 that the IT auditor and two research auditors will be independent of the laboratories involved in the study. This point is repeated on page 8 of the published report (https://doi.org/10.1098/rsos.191375), which states “An IT auditor and two research auditors independent of the laboratories involved in the study were also involved in the project.”

However, the published report also states on page 16:

> Potential conflicts of interest may exist: one of the research auditors and the IT auditor work at University of Padova, where we also had a collaborating laboratory. The IT auditor has joint publications with the lead researcher of the University of Padova site.

Here too, the situation is not described as an explicit protocol deviation.

It is also notable that belief in ESP by the site-PI and experimenters was much higher at the University of Padova than for any of the other laboratories in the study (published report, Table 1, page 15).

Given the outcome of the TPP study, the potential biases of the affiliated auditors may be more beneficial than detrimental. Obtaining evidence for the null model in a study that has personnel with beliefs favorable to ESP may inspire more confidence in the conclusions. Also, the IT auditor’s knowledge of ESP research may have been beneficial for the pre-study software validation. The IT auditor recognized that generating the random target before the response in a precognition experiment is an error. A person not familiar with ESP research may not have recognized that as an important distinction. Evaluation: adequate.

**If the study would have found evidence for ESP, the potential implications of the above protocol deviations would have been different**, particularly when combined with other inadequate practices for managing the data collection server and software as described in section 8. The probability of bias from these inadequate practices would then have to be balanced against the probability of ESP.

### *Not Significant Protocol Deviations*

*Protocol deviation: Re-starting the ESP task.*

Different versions of the instructions for experimenters are on the OSF website. One version (https://osf.io/su4e5) emphasizes that the data collection software should not be re-started for a participant if the program crashes or jumps back to the start screen. Another version (https://osf.io/6uan5/) says to terminate the program if there are problems but does not emphasize not re-starting. The lab notebooks indicate that the ESP test was sometimes re-started for a participant when software problems occurred (for example, rows 196, 207, 208, 220, 290 in the spreadsheet for the lab notebooks). The instructions for experimenters may have been changed to make it more clear that a

participant should not be in the database for two sessions. However, there probably are cases with multiple sessions for one participant in the database. Though technically not consistent with the instructions, these cases cannot create a false-positive ESP effect if all ESP trials for a participant are included in the analysis as specified in the preregistered analysis plan. The published report notes that software crashes occurred and includes an evaluation of incomplete sessions.
Evaluation: Adequate.

*Protocol deviation: Criteria for test data exclusions changed after preregistration.*

As described in section 6 above, data collection during the study included some test data that were excluded from the final analysis. The published report states that "The experimenter ID(s) of the test account(s) were specified in the preregistered analysis code" (page 13).  Similarly, a comment in the data analysis code states: "(these IDs were preregistered to be excluded)." However, the final analysis used for publication excluded two additional lab id codes that were added after preregistration. The excluded data were only 81 records and could not have altered the study outcome.
Evaluation: misleading descriptions.

# 8. Possibility of Bias and Unintentional or Intentional Data Alterations

**Questions**

A. Did the study design or conduct have potential sources of bias?
   Evaluation: **Nothing to add beyond section 3 above**.

B. At any time, could one person acting alone alter or fabricate data without detection?
   Evaluation: **Given the security measures in the study design, bias in the data collection software was essentially the only possibility for undetected data alterations. Fraudulent programming to produce a false precognitive effect would be trivial to implement and several key computer system security measures were not adequate, including two significant protocol deviations. The programmers had inappropriate full access to the server operating system, which would have allowed file changes without detection. Planned logging of server accesses to verify no unauthorized access was not done. The planned monitoring of software changes also was not done. Combined, these represent serious deficiency in managing the security of the application server.**

   **However, the possibility of undetectable data alterations from the programming was very unlikely because (a) the initial pre-study software validation would (and did) detect programming errors, whether unintentional or intentional, (b) an IT auditor was specifically hired to detect unauthorized software changes during the study, which would provide a strong disincentive to attempt programming fraud, (c) the study outcome of strong support of the null model would require fraudulently canceling a true precognitive effect, which would require significant effort and unlikely motivations for a study like this, and (d) the lead programmer for the project changed during the project, which would reduce the possibility that one programmer acting alone biased the data in a way that is undetectable.**

**Evaluations**

***A. Did the study design or conduct have potential sources of bias?***

As discussed below, the automated data collection process left no opportunities for the participants or the experimenters with the participants to create a false-positive bias in favor of ESP. The automated data collection process also reduced, but did not eliminate, the possibility of a false-negative bias in favor of the null model. As discussed in section 3 above, the experimenter who interacts with the participants could convey expectations that could affect a participant’s performance, and the experimental environment could be distracting or otherwise not appropriate for the test. Those topics are addressed in section 3 above. Another possibility would be an experimenter who fraudulently acted as a participant and made no effort to produce an ESP effect. However, the large number of experimenters and group testing make such possibilities unlikely to affect the study outcome.

The most direct threat of bias would come from the automated data collection system, as described below.

Evaluation: Nothing to add beyond section 3 above.

### *B. At any time, could one person acting alone alter or fabricate data without detection?*

Good research practices should make it very difficult or impossible for one person acting alone to unintentionally or intentionally (fraudulently) bias the data with little chance of detection. As noted in the introduction, this standard is routinely implemented in regulated medical research by double-blind, randomized clinical trials and a research culture that expects documented independent checks of all data collection, processing, and analyses steps.

However, academic research is typically unblind and often has situations in which fraud by one person acting alone would be easy and tempting with negligible chance of detection. The TPP had the explicit goal of doing more than the typical academic study to make unintentional or intentional biases difficult or impossible.

*Data Collection Software Design*

Data collection was done with a custom developed web-based application. As a frame of reference, the expectations for managing the integrity and security of computer systems and software for regulated clinical trials are described at US FDA (2007, 2023)

The TTP data collection software design included automatically uploaded a copy of the data to a third party publicly readable (live) repository as well as keeping a copy on the application server. The data was to be uploaded in batches of 200 lines, which was data for about 5 participants (41 lines per participant for 36 trials). Batch uploads were done to avoid the possibility that someone who knew when certain participants were tested could deduce the data for individual participants. The data in the repository would also have the date-time of data collection, the lab Id code, site-PI Id code, and experimenter Id code hashed to prevent identifying individual participants.

The automated processes for data collection and directly transmitting a copy of the data to a publicly readable third-party repository as well as keeping a copy on the application server would make undetectable data alterations subsequent to data collection very difficult or impossible. A fraudster would have to have access and make changes to the data on the application server and on the repository, and risk that no one was closely following the publicly readable repository data. Undetectable data alterations would be virtually impossible if the repository has version controls that cannot be circumvented, and/or the application server and repository were managed by different persons without mutual access (although the latter was not part of the design or implementation).

With this design, by far, the greatest threat for undetected data alterations would be in the programming of the data collection software. This includes unintentional or intentional (fraudulent) programming errors.

The data collection design could have been slightly more secure if the data had been uploaded to the repository after each trial and the delay in publicly displaying the data accomplished on the repository rather than in delaying the upload.
Evaluation of software design: **very good**.

*Physical Security of the Server*

Evaluation of the security of a computer system begins with the physical security. The web server was set up on the DigitalOcean cloud service. In terms of security, this is usually preferable to a server physically located on a university campus, and preferable to software distributed and run on workstations at each lab.
Evaluation: **very good**.

*Data Collection Software Programming*

The programming effort to intentionally bias data collection in this and other automated precognition studies would be easy and tempting, requiring only the addition or alteration of one line of code. Code added after the generation of the random target could have the logic that if the target is not equal to the response and the number from a random number function is above a certain value, set the target for the trial equal to the response. An unbiased version of the program could be used for initial testing, and replaced by the biased version at or shortly after the start of data collection for the study. The biased version could be swapped out again near the end of the study, or perhaps intermittently during the study. This fraud would not leave patterns or artifacts in the data, and would be impossible to detect from the data alone.

Measures to prevent such programming fraud and unintentional errors include formal software validation, controlled access to the server, and monitoring changes in the data collection software.

The data collection software in the TPP was custom developed by a lead programmer with some assistance from a second programmer.

On January, 14, 2020 the first data for the study was collected with the software.

On April 27, 2022 the last study data upload to the data repository was made.

In September, 2020 the second programmer became lead for the project when the previous lead programmer became no longer available.

*IT Auditor*

A person was hired (a) to validate the software before use in the study, and (b) to "oversee data integrity throughout the study," including (c) verifying that the software and data did not have unauthorized changes during the study, and (d) providing an overall system and data integrity report at the end of the study (https://osf.io/dhvrm). In the preregistration the person was called the "IT quality expert." In the published paper and on the OSF website for the project, the person is called the "IT auditor." The latter term is used here. However, the role was more quality control than auditing. Performing software validation and overseeing data integrity throughout the study are quality control functions. An auditor would verify that the quality control functions were adequately completed and documented, but would not implement the quality control functions.

The IT auditor produced three reports.

1. Pre-study validation, dated August 24, 2018 (https://osf.io/dx4nw)
2. Pre-study validation – review 2, dated September 25, 2018 (https://osf.io/ex56a)
3. Final report, dated May 24, 2022 (https://osf.io/p62gw)

Having an independent person specifically validate the software and verify the integrity of the computer systems is very valuable, particularly when programming is done unblind. This would provide a strong disincentive for programming fraud and should eliminate the possibility that programming fraud would be easy and tempting with little chance of being detected. The strengths and weaknesses of the content of the IT reports are described in comments below.

Evaluation of having an IT auditor: **very good**.

*Authorized Server Access*

The data collection software was initially set up on a server that was used for software development and validation, and for a pilot study. However, the server access and domain access expired, so a new server had to be set up.

On October 19, 2019 the PI set up the new server on the DigitalOcean cloud service, which was different that for the previous server. The PI set the root password and granted root (full system) access to the first lead programmer.

On January 20, 2020, the IT auditor was granted read-only access to the server.

On September 8, 2020, the PI created a sudo (access to system files) account for the second lead programmer when he took over the project.

On May, 10, 2022, the PI created a new sudo account for the second lead programmer after he forgot the password to his account.

On May 11, 2022, the PI granted sudo access to the account for the IT auditor to allow post-study evaluation of the server.

The root user account on the server for the first programmer remained active throughout the study because his role was phased out gradually and he continued to have a possible role for backup support and troubleshooting.

The final IT report pointed out that the data collection application was installed on the root folder of the server, and that:

> Using user root to run the application, or installing the application on folder /root is a bad practice and it is necessary to avoid it.
>
> Talking about the root user, I think it is necessary to consider a possible conflict of interest: if the author (or any collaborator that are involved in code and data analysis) is a root user, it could be very difficult to estimate and certify that the code and data collected was not changed during the data collection phase due to server access "behind the scene".
>
> Technically, it is always possible for the root user to access the server, changing something (like update a file and its last modified date, delete a file, upload a new one) and exiting after cleaning some access registers.
>
> It is not easy to block or find this corrupting behavior. (page 9)

In my experience working in industry, the operating system of a server is typically managed by an IT department in accordance with SOPs that limit access permissions to prevent users from circumventing or altering audit trails. Guidelines for computer systems for regulated clinical research specify carefully limited access and particularly restricted ability to alter the date-time records for events (US FDA 2007, 2023). Valid date-time records are the foundation of audit trails and overall system and data integrity. Programmers and other users should not have access to the operating system or configuration files that control the system logs. The IT staff typically do not have the specialized programming skills and knowledge of workflow to fraudulently change programming. The operating system logs and controls limit the ability of programmers to make undetected server accesses and file changes.

For TPP, the programmers had full access to the operating system, including the operation or lack of operation of the system logs and the ability to alter date-time records.

As noted in the final IT report, giving programmers full access to the server operating system would allow server access and file changes that cannot be detected with the operating system and does not provide the system security that is needed for good research practice. The ability to make undetected changes would occur even if the system access logs were set properly.

Evaluation of authorized server access: **not adequate.**

*Pre-Study Software Validation*

According to the pre-study validation reports, the IT auditor identified one major error in the data collection programming and other weakness that could be improved. The major error incorrectly selected the target before the trial began (rather than after the participant made the response as required for precognition) and would have allowed a computer-savvy participant to use the web browser to see the selected target for a trial before making a response. Other recommended improvements included encrypting the data transmitted to the repository and adding a step of internal data validation. The report also listed various tests that were done to verify that the software handled unexpected keystrokes and other potential problems.

The programmers made corrections that were reviewed with Git links and accepted by the IT auditor.

Evaluation of the initial pre-study software validation: **very good**.

Unfortunately, the server that was used for the pre-study validation was not the server that was used in the study. The data collection software had to be modified to correct the locations of some files when the new server was implemented on a different cloud service. In my experience in regulated medical research, additional software validation would be required if critical software was installed on a new server—and particularly if software changes were made and/or a different cloud service was used. Although obvious software disfunction is more likely than subtle biases, at a minimum some basic documented software validation should be done when critical software is installed on a new server. The PI stated that a pre-study validation of the data collection software on the new server was not done because funds were not available for a second pre-study validation.

Evaluation of pre-study validation after the new server: **not adequate**.

*Monitoring Server Access*

The Supplementary Materials (https://osf.io/m5b8x) part of the preregistration specified controlled access to the data collection server as described below:

> Records will be kept about the people who had change access to the server after the code of the experimental software was finalized, and why do/did they have access. The system password will be changed when a person who had change access to the server leaves the project. The system password will not be recorded in any form, instead, it will be memorized by the people who have change access to the server. The system password will not be provided to anyone other than the authorized individuals and care will be taken that the password cannot be overseen or overheard by unauthorized individuals. Server access will only be permitted with the approval of the Lead-PI and the project's programmer. The Lead-PI will keep a log of server accesses authorized by him. Unexpected logins to the server will be noted in the unexpected server event log. (page 35)
>
> …
>
> At the end of the study, the IT quality expert will submit a final software and data integrity report consisting of the pre-study software validation report and findings during the checking of software, database, and server logs. (page 36)

The study document with more detailed information about the role of the IT auditor was more explicit that "all server access will be logged by the server automatically" and "At the end of the data collection period of the project, the log of authorized accesses and server logs logging accesses will be noted, and any unauthorized accesses will be logged in the unexpected server event log kept on OSF" (https://osf.io/dhvrm).

The final IT report stated that the access logs on the server were not maintained due an apparent error in a server system configuration file. Thus, it was not possible to check or track direct server accesses during the study, or to verify that the only accesses were authorized by the Lead-PI. Although not identified as such in the IT report, **this is a significant protocol deviation.** The server configuration error was not discovered until after the study was complete, which means that the server was used for over two years with no one checking the logs. That is not good computer system management. Evaluation of monitoring server access: **not adequate.**

*Monitoring Software Changes*

The optimal practice for preventing unauthorized changes to programming code would be to have software tracking set up and controlled by an independent quality control person. Software tracking set up and controlled by the programmers being monitored is intrinsically more vulnerable to compromise.

In TPP, the programmer set up a Git repository that was relied upon for tracking changes to the data collection software. Git is a complicated system for software development that was designed for multiple programmers working simultaneously on a project. A Git repository is made on the server that has the production application software. The repository can track changes to the files and folder structure for the application. The repository can be copied or cloned to another server or to a workstation for a programmer. Changes can be made and tested in a remote repository and transferred to the production or local repository. Git identifies potential conflicts from changes to the same code by different programmers. GitHub and GitLab are two websites intended for online storing and working with Git repositories.

For TPP, the lead programmer established a Git repository on the data collection server and a copy on GitLab (https://gitlab.com/gyorgypakozdi/psi). The GitLab repository shows many changes to the software on the initial server used for development and testing (in 2018), and one set of changes when the new server was established (October, 2019).

The PI intended that the GitLab repository would be continuously or frequently synchronized with the application server and provide a reliable record of all changes to the software. The preregistration stated “Software code running on the server is continuously synchronized with a version-controlled code repository (GitLab). The GitLab account is shared with the auditors, who can verify at any time that the software code is unaltered” (https://osf.io/dkb4g page 16).

However, when the PI made inquiries to obtain information in response to questions during this audit, he discovered that software tracking was not implemented as was preregistered and as he intended and thought had been done. The GitLab repository was synced with the application server at the beginning of the project, but not throughout the study. Among other things, the auditors could not verify at any time that the software code was unaltered. **This is a significant protocol deviation.** There is no indication that any of the auditors attempted to verify the software code during the study.

Directly accessing and checking the application server would be required to verify the software code. Software change tracking was limited to the Git repository on the application server.

**The final IT report gives no indication that the IT auditor accessed and checked the server and software during the time data were being collected.** Given that the IT auditor's role included overseeing data integrity throughout the study (https://osf.io/dhvrm), my expectation for that role would include directly accessing and checking the server and software during the study. The IT auditor was given access to the server for such checking. The lack of logs for server accesses and unexpected events would presumably have been discovered if the IT auditor would have checked the server during the study.

The IT auditor did a retrospective evaluation of the application server after data collection was completed. The final report of the IT auditor noted that a change had been made to an application file that was not tracked in the Git repository.  The change was discovered when the "git status" command was run that compares the current state of the application files with the state in the repository.  The IT auditor apparently ran the git status command after data collection had stopped.

I do not have a working knowledge of Git and searched for information about key aspects of Git tracking. Git appears to be designed to give programmers control over the tracking of software changes. Git does not monitor folders and automatically track file changes. Such automatic tracking would be counterproductive for the frequent file changes when creating and editing programming code. Git is based on a programmer entering commands when the programmer wants Git to recognize and track a file change. Files can be directly changed outside Git processes and the changes will not be identified by the Git system until a Git command such as git status is run, as occurred with the TPP. If the git status command is not run, the changes will not be tracked.

Git is very complicated and includes several options for temporarily or permanently excluding a file from Git tracking. Updating a repository from another repository is a particularly complicated process. Extensive expertise in the many options, interactions of the options, and inner workings of Git would be required to confidently use Git to prevent programming fraud—or to know whether Git can be used reliably for that purpose.

The IT auditor's retrospective evaluation of the application server concluded that "After a complex analysis of the server, there is no evidence of a data breach nor of data or code integrity corruption. The data that has been collected is genuine and safely stored in the server."

The final IT report did not explicitly discuss the potential for threats due to file changes that were not tracked by Git. The final IT audit report says that no evidence of unauthorized changes was found, but also says that (a) the programmers could have made file changes that could not be detected given the programmers' (inappropriate) access to the operating system and (b) expected tracking of server accesses was not done. The lack of evidence for unauthorized software changes was not reconciled with the possibility of undetectable changes.

In the absence of an explicit detailed evaluation of the potential threats due to file changes not identified by Git, it is not possible to establish a degree of confidence in the monitoring of software changes. This is particularly true given that the software monitoring was not conducted as the PI intended and had preregistered, and the IT auditor apparently did not directly check the software and server during data collection. Evaluation of monitoring software changes: **not adequate**.

*Mitigating Factors*

Given that key components for the security of the server and software were not adequately handled, the rationale for confidence in the data must come from factors beyond the state of the server and software at the end of the study. Several factors mitigate the possibility of programming fraud in this study.

1. Probably most important, the simple fact that a person with knowledge of computer systems and programming was hired specifically to check the integrity of the programming for the study would provide a strong disincentive to attempt to alter the programming. This would achieve the top priority of eliminating situations when fraud by one person acting alone would be easy and tempting with little chance of getting caught.
2. The study outcome strongly supported the null model of no effect.  Fraud in this study would be to cancel a true precognitive effect. While the programming effort to fraudulently produce an apparent precognitive effect would be trivial, the effort to cancel a true precognitive effect would be more substantial. The cumulative hit rate for the data would need to be tracked and the programming adjusted to neutralize the effect. Unless carefully done, declines or oscillations in the hit rate would appear in the data.
3. A person having motivation to cancel a true precognitive effect in a study like this might be possible, but it is not likely.
4. Changing lead programmers during the study also reduces the possibility of programming fraud by one person acting alone. Collaboration on fraud has been very rare, at least among cases of detected fraud. Fraudulent code left by the first programmer could be detected by the second programmer. If a fraudulent program was removed by the first programmer, an inflection point for the effect would be apparent in the data. Similarly, an inflection in the data would be apparent if the second programmer implemented fraud. Checking for possible inflections or other patterns in the data was not part of this audit. The security value of changing programmers is somewhat reduced by the fact that the first programmer continued to have root access to the server throughout the study, even though he was no longer lead programmer.

## References

US FDA (2007). Guidance for Industry: *Computerized Systems Used in Clinical Investigations*. https://www.fda.gov/media/70970/download

US FDA (2023). Draft Guidance for Industry: *Electronic Systems, Electronic Records, and Electronic Signatures in Clinical Investigations: Questions and Answers.* https://www.fda.gov/media/166215/download (Use the final version when it become available.)

# 9. Consistency Between Preregistration and Study Report

### Questions

Verify that:

A. the study report has a direct link or URL to the preregistration and that the preregistration is irreversibly publicly available on a study registry;

B. all preregistered confirmatory hypotheses and analyses were included in the study report;

C. the study report has no ambiguity about whether each analysis is confirmatory, preregistered exploratory, or post hoc, and that these classifications are consistent with the preregistration;

D. the confirmatory statistical analyses in the study report were the preregistered analyses, including specific test and direction of effect;

E. all data exclusions, transformations, and other data modifications for the confirmatory analyses described in the study report were included in the preregistration;

F. any preregistered evaluation of data for dropouts was actually done and described in the study report;

G. key procedural steps are consistent in the study report and preregistration;

H. any deviations from the preregistration and protocol are appropriately described.

Evaluation: **All were handled very good in the published study report except, that three protocol deviations identified in this audit were handled in a way that is adequate.**

These evaluations may be considered as outside the scope of a research audit—and, a draft or final report of the study may not be available when an audit is conducted. However, these evaluations should be done before a study is published. It is efficient to include them in a research audit when possible because the auditor should already have a detailed knowledge of the preregistration.

### Evaluation

Biased inconsistencies between a preregistration and the subsequent published report occur to a surprising (shocking) degree (Claesen et al., 2021; Goldacre et al., 2019). Preregistration may be more useful for detecting bias than for preventing bias. Verifying consistency between a preregistration and published report is a necessary step for study integrity. Optimally, discrepancies will be found before publication rather than by a critical reviewer after publication.

All of the above questions were handled very well in the final report except for the last item. Three protocol deviations identified in this audit were handled in a way that is adequate, as described in section 7.

### References

Claesen, A., Gomes, S., Tuerlinckx, F., & Vanpaemel, W. (2021). Comparing dream to reality: An assessment of adherence of the first generation of preregistered studies. *Royal Society Open Science*, 8, 211037211037. http://doi.org/10.1098/rsos.211037

Goldacre, B., Drysdale, H., Dale, A. et al. (2019). COMPare: A prospective cohort study correcting and monitoring 58 misreported trials in real time. *Trials* 20, 118. https://doi.org/10.1186/s13063-019-3173-2 Also see, https://www.compare-trials.org/

Lessons and recommendations from this audit are discussed at https://jeksite.org/psi/tpp_audit_lessons.pdf.

## Copyright Notice