Improving Transparency, Falsifiability, and Rigour by Making Hypothesis Tests Machine Readable

Daniel Lakens[1] & Lisa M. DeBruine[2]

[1] School of Innovation Sciences, Eindhoven University of Technology
[2] Institute of Neuroscience and Psychology, University of Glasgow

Abstract

Making scientific information machine-readable greatly facilitates its re-use. Many scientific articles have the goal to test a hypothesis, so making the tests of statistical predictions easier to find and access could be very beneficial. We propose an approach that can be used to make hypothesis tests machine readable. We believe there are two benefits to specifying a hypothesis test in a way that a computer can evaluate whether the statistical prediction is corroborated or not. First, hypothesis tests will become more transparent, falsifiable, and rigorous. Second, scientists will benefit if information related to hypothesis tests in scientific articles is easily findable and re-usable, for example when performing meta-analyses, during peer review, and when examining meta-scientific research questions. We examine what a machine readable hypothesis test should look like, and demonstrate the feasibility of machine readable hypothesis tests in a real-life example using the fully operational prototype R package scienceverse.

Keywords: hypothesis testing, machine readability, metadata, scholarly communication
Word count: 4790

In many scientific fields researchers rely on hypothesis tests to determine whether empirical observations corroborate predictions. In a well-specified hypothesis test, a hypothesis is used to derive predictions, which are operationalized when designing a specific study, and translated into a testable statistical hypothesis. Data is collected, and the statistical hypothesis is corroborated or not. Although this process sounds relatively straightforward, hypothesis tests are performed rather poorly in practice. First, statistical hypotheses are

Both authors contributed equally to the manuscript. First authorship was determined based on a Great League trainer battle between the authors in Pokemon Go.

Correspondence concerning this article should be addressed to Daniel Lakens, ATLAS 9.402, 5600 MB, Eindhoven, The Netherlands. E-mail: D.Lakens@tue.nl

stated verbally, but these verbal descriptions rarely sufficiently constrain flexibility in the data analysis. Second, there is a lack of transparency about which statistical tests in the results section are related to the predictions in the introduction section, and which pattern of results should be observed to conclude that a prediction is corroborated. Finally, researchers typically only implicitly specify what would lead them to act as if their prediction is confirmed (i.e., typically a *p*-value smaller than 0.05), and rarely specify what would lead them to act as if their prediction is falsified. Currently, it is often only possible to indirectly infer the authors' decision criteria, leading to disagreement about whether new patterns of results from replications should be considered to support or refute the hypothesis.

By contrast, a well-specified hypothesis test states the statistical hypothesis for each prediction in a way that eliminates flexible implementations, clearly links predictions derived from the theoretical hypothesis to statistical tests, and gives unambiguous criteria to conclude the prediction is corroborated, falsified, or that the results are inconclusive. When we refer to falsifiability, we limit ourselves to the falsification of statistical predictions, not entire theories. A specific operationalization of a theoretical prediction always requires auxiliary hypotheses, and if a statistical hypothesis is falsified, it remains unclear whether the problem lies with the theory, or the auxiliaries (Meehl, 1990). Additionally, while machine readability is no guarantee that a hypothesis test is logically or statistically free from error, it provides reviewers and readers a way to unambiguously assess this, avoiding problems of interpretation.

We propose that the gold standard for well-specified hypothesis tests should be a statistical prediction that is machine readable. This means that a computer can evaluate whether a statistical prediction is corroborated (or not) based on clearly articulated evaluation criteria and the observed data. Computers do not handle ambiguity well, and making a hypothesis test machine readable guarantees that it is specified precisely. While some of the improvements we suggest could also be achieved through careful verbal descriptions of mutually exclusive and exhaustive decision criteria in manuscripts and preregistrations, we believe that there are two broad arguments for a move to machine readable hypothesis tests. The first argument is that by specifying hypothesis tests in a format that can be read and evaluated by a machine, tests of statistical predictions and the conclusions derived from these tests will become more transparent, statistically falsifiable, and rigorous. This provides a first step to improve the currently poor practices scientists use to test hypotheses. The second argument is that the benefits of making data FAIR (findable, accessible, interoperable, and reusable) also apply to statistical predictions. If all aspects required to evaluate the test of a statistical prediction are machine-readable, we can easily reuse this information (e.g., when performing a z-curve analysis, effect size meta-analysis, or *p*-curve analysis), and find and access this information (e.g., to answer meta-scientific questions about the proportion of statistical results in the scientific literature that corroborate the prediction). Although achieving all benefits of machine readable hypothesis tests might take many decades, and will require extensive collaboration, coordination, and standardization, we believe machine readable hypothesis tests as they can be implemented based on the approach and R package outlined in this manuscript can already lead to immediate improvements in research practices.

Poor practices when testing predictions

As a concrete example of a typical hypothesis test in the published literature, DeBruine (2002) posited the theoretical prediction that people would exhibit higher levels of prosocial behavior towards those who physically resemble them, which follows from the idea that actions are influenced by an implicit evaluation of relatedness based on phenotypic similarity. Physical resemblance was manipulated by morphing face photographs with either the participant's own face (self morphs) or another person's face (other morphs). There were two versions of this manipulation: faces were morphed in shape only (n = 11) or in both shape and color (n = 13). Prosocial behavior was measured as the choice to trust or reciprocate trust in a monetary trust game where the first player could decide whether to trust the second player to split money and the second player, if trusted, could decide whether to reciprocate this trust by splitting the money equally or selfishly. The theoretical hypothesis was operationalized, and the operationalized prediction stated that people playing a trust game would trust and reciprocate more when playing with a person who was represented by a self morph than by an other morph. The statistical prediction was tested by counting the number of trusting and reciprocating responses participants made to self and other morphs and then performing a *t*-test on these counts, separately analyzed for the shape morphs and the shape-colour morphs. The statistical results indicated that participants made more trust responses to self morphs than to other morphs for both morph types. However, there were no differences in how often they reciprocated their partners' trust. The conclusion drawn from this study was that these results show that facial resemblance can increase prosocial behaviour. It was noted that the fact that an effect was observed for the trust measure, but not for the reciprocation measure, could perhaps be explained by the different pay-off structures in this particular game.

The first problem we can identify in this example is that it is not clear whether the operationalized prediction was confirmed if an effect was observed on both the trust measure and the reciprocation measure, or either of the two measures. From the conclusion the author draws, we can infer that the statistical prediction would be considered corroborated if the morphing manipulation had an effect on either the trust measure, or the reciprocation measure, or both. However, even if the decision rule can be inferred from the discussion, it is still not clear which patterns would be considered corroboration or falsification in future replications that might find similar but not identical patterns of results.

The second problem is that it is not clearly specified what would corroborate the hypothesis and what would statistically falsify the hypothesis. Although it is never explicitly stated, we can infer that the prediction would be corroborated when either of the two tests is significant at an alpha level of 0.05, without correcting for multiple comparisons. Furthermore, we can infer that a non-significant *p*-value is interpreted as the absence of any meaningful effect (even though this is a formally incorrect interpretation of a null hypothesis test).

The third problem is that there is a range of options when analyzing the data (e.g., pooling the two types of morphs in one analysis, or reporting two separate analyses by morph version). As is often the case when testing statistical predictions, no unique analysis strategy follows unequivocally from the introduction and methods section, which can lead to flexibility in the data analysis.

What Does a Formalized Test of a Prediction Look Like?

If we want to make hypothesis tests machine readable, we need to capture all essential aspects of a hypothesis test in a machine-readable data structure. A hypothesis test is a methodological procedure to evaluate a prediction that can be described on a conceptual level (e.g., people exhibit higher levels of prosocial behavior towards those who physically resemble them), an operationalized level (e.g., people playing a trust game make more trusting decisions when the person they play against is a self morph versus an other morph), and a statistical level (e.g., the average number of trust moves is statistically larger for games against self morphs than against other morphs in a dependent *t*-test).

When we evaluate the result of a statistical prediction, we need to perform a statistical test, retrieve the relevant test result, and compare this to one or more criterion values. For example, our statistical prediction might be that we will observe a positive difference in the means between two measurements, which will be examined in a dependent *t*-test, from which we will determine the lower and upper 97.5% confidence interval around the mean difference, which we will compare against a value of 0. Statistical hypotheses are probabilistic, and probabilistic hypotheses can be made falsifiable "by specifying certain rejection rules which may render statistically interpreted evidence 'inconsistent' with the probabilistic theory" (Lakatos, 1978, p. 25). A hypothesis test thus requires researchers to specify when the observed results of a statistical test will lead them to act as if their prediction is consistent with the data, inconsistent with the data, or inconclusive (Neyman & Pearson, 1933).

As highlighted above, one limitation of current practice when testing hypotheses is that researchers often do not explicitly state what would corroborate or falsify their prediction. To be able to unambiguously evaluate a hypothesis, researchers need to specify the rules they will use to evaluate whether statistical results corroborate a prediction, falsify it, or when the results are inconclusive. For example, in a 2x2 design, many different patterns of means across the four cells could be predicted (e.g., one of two main effects, or a specific pattern of the observed interaction effect), but the full pattern of possible results that would corroborate or falsify a prediction is seldom made explicit.

There are different approaches that can be used to statistically conclude that the prediction made in a study is falsified. In practice, corroborating or falsifying a statistical prediction in a single study is rarely sufficient to draw strong conclusions about a theory (Lakatos, 1978), and one should always keep random variation in mind when interpreting statistical results. One approach to conclude a prediction is falsified is known as equivalence testing (Lakens, Scheel, & Isager, 2018). An equivalence test requires researchers to specify a smallest effect size of interest, and tests if the presence of an effect that is large enough to be deemed interesting can be statistically rejected.

Continuing our example, we might conclude our prediction is corroborated when we can statistically conclude the observed mean difference for the trust measure, or the reciprocation measure, or both, is greater than zero, and neither are statistically smaller than the smallest effect size we care about. The prediction would be falsified if both effects are statistically smaller than the smallest effect size of interest, and inconclusive if we can neither conclude either effect is statistically greater than zero, nor statistically smaller than the smallest effect size we care about. If our statistical test is a dependent *t*-test, our test result is the upper and lower bound of a 97.5% confidence interval (i.e., a hypothesis test with a Bonferroni corrected alpha level of 2.5%), and our smallest effect size of interest is 0.2, we can conclude that we have corroborated our prediction if the lower bound of our 97.5% confidence intervals are larger than 0 and the upper bound is not smaller than 0.2. We decide that our prediction is falsified if the upper bound of our 97.5% confidence intervals are smaller than 0.2, and our data is inconclusive in all other situations.

Computationally Evaluating Hypotheses

If a prediction is machine readable, it is possible to automatically determine if a prediction is corroborated by the data. Although computational reproducibility is becoming increasingly popular as user-friendly tools are continuously being developed, there are no existing solutions that make hypothesis tests machine readable and re-usable. We envision machine readable hypothesis tests as part of a completely reproducible workflow. Computer scripts will load the raw data, and if needed, create the analytic data from the raw data (e.g., outlier removal, transformations, computing sum scores according to pre-specified rules). The statistical tests are automatically performed on the analytic data, and the relevant test statistics are retrieved. These test statistics are compared against pre-specified criteria, based on decision rules that evaluate whether the prediction is corroborated, falsified, or inconclusive. All the information that is required to perform these operations is stored in a structured meta-data file.

We provide a vignette for a Quick Demo (see Open Practices section) with a concrete example of a machine-readable statistical prediction for the study by DeBruine (2002) described above. It is written using the fully operational prototype implemented in the R package **scienceverse** and produces a JSON file, which is an open-standard file format (in JavaScript Object Notation) that can be used to transmit data. Because it is an open-standard file format, it can easily be converted into any other open data file format (for example, the Journal Article Tag Suite), which in essence are all nested lists.. It can also be converted to a human-readable report, summarising the study with verbal descriptions and a list containing the conclusion for each statistical prediction.

In summary, to make statistical hypotheses machine readable, we need to identify the individual components that make it possible to evaluate a hypothesis test. Our example relies on a *statistical hypothesis* that is tested in an *analysis* that takes *data* as input and returns test *results*. Some of these tests results will be compared to *criteria*, used in the *evaluation* of the test result. The sections below describe how each component can be specified in a machine-readable format.

Setting up a study. The top level list (Box 1) contains components describing different aspects of the study, such as authors, hypotheses, materials, methods, data, and analyses. In the future we might be able to describe all meta-data pointing to information in a scientific article that we would like to be able to retrieve, but here we will focus on the aspects of the study that are required to make statistical predictions machine readable. To achieve this, we need a meta-data file that specifies the hypotheses, the analyses, and the evaluation criteria for each prediction.

The meta-data file is structured as a JSON object, which is a list of keys and values, separated by a colon. The list items are separated by commas and surrounded by curly brackets (see Box 1). The basic structure requires keys for the study name, info, authors, hypotheses, methods, data, and analyses. All values (except the name) default to an empty array “[]” where these components can be later added.

Box 1. The top-level structure of the machine-readable study description.

```
{
    "name": "Kinship and Prosocial Behaviour",
    "info": [],
    "authors": [],
    "hypotheses": [ ...Box 2... ],
    "methods": [],
    "data": [ ...Box 6... ],
    "analyses": [ ...Box 5... ]
}
```

Hypotheses. A study could contain multiple hypotheses, but our example contains only one. Each `hypothesis` (Box 2) consists of an `id` for referencing the hypothesis in other components, a verbal human-readable `description`, one or more `criteria` to evaluate analysis results, and rules to determine `corroboration` or `falsification` of the hypothesis. If the data are available, these rules are automatically evaluated and a `conclusion` of “corroborate”, “falsify”, or “inconclusive” is added.

Box 2. The hypothesis component.

```
    "hypotheses": [
        {
            "id": "self_pref",
            "description": "Cues of kinship will increase prosocial
                            behaviour. Cues of kinship will be
                            manipulated by morphed facial self-
                            resemblance. Prosocial behaviour will be
                            measured by responses in the trust game.
                            The prediction is that the number of
                            trusting AND/OR reciprocating moves will
                            be greater to self morphs than to other
                            morphs.",
            "criteria": [ ...Box3... ],
            "corroboration": { ...Box 4... },
            "falsification": { ...Box 4... },
            "conclusion": "corroborate"
        }
    ]
```

Criteria. Each criterion (Box 3) needs an `id` to be able to reference it in the evaluations and references a named `result` from an analysis with the id `analysis_id`. An `operator` and a `comparator` are provided for each criterion to specify the method of comparison (e.g., >, <, =, !=) and the comparison value (e.g., 0). For example, the first criterion specifies that if the statistical result "conf.int[1]" from "trust_analysis" is ">" than "0", then the criterion "t_lo" evaluates to a `conclusion` of "true". In other words, if we can statistically reject the null hypothesis (because the lower bound of the confidence interval does not overlap with 0), this criterion of our statistical prediction is corroborated. Although in essence this describes nothing more than what researchers do when they interpret test results, this decision process is now captured and made explicit in machine-readable code.

Box 3. Criteria for evaluation.

```
"hypotheses": [
    {
        ...
        "criteria": [
            {
                "id": "t_lo",
                "analysis_id": "trust",
                "result": "conf.int[1]",
                "operator": ">",
                "comparator": 0,
                "conclusion": true
            },
            {
                "id": "t_hi",
                "analysis_id": "trust",
                "result": "conf.int[2]",
                "operator": ">",
                "comparator": 0.2,
                "conclusion": true
            },
            {
                "id": "r_lo",
                "analysis_id": "recip",
                "result": "conf.int[1]",
                "operator": ">",
                "comparator": 0,
                "conclusion": false
            },
            {
                "id": "r_hi",
                "analysis_id": "recip",
                "result": "conf.int[2]",
                "operator": ">",
                "comparator": 0.2,
                "conclusion": true
            }
        ],
    },
    ...
]
```

Hypothesis Evaluation. The `corroboration` and `falsification` subcomponents (Box 4) describe rules to determine corroboration or falsification of a hypothesis from the criteria conclusions, and each consists of three elements. The `description` element contains verbal descriptions of the decision rules for concluding the hypothesis is corroborated or falsified. The `evaluation` element contains a logical version referencing the criteria `id`. For example, "`(t_lo & t_hi) | (r_lo & r_hi)`" means that the corroboration `result` will be set to "true" if the first two criteria are both true, or if the last two criteria are both true, while "`!t_hi & !r_hi`" means that the falsify conclusion will be set to "true" if both of these criteria are false (note that an exclamation mark means 'not').

Box 4. Corroboration and falsification rules.

```
"hypotheses": [
    {
        ...
        "corroboration": {
            "description": "The hypothesis is corroborated if the
                            97.5% CI lower bound is greater than 0
                            and the 97.5%  CI upper bound is
                            greater than 0.2 (the SESOI) for either
                            the trust or reciprocation moves.",
            "evaluation": "(t_lo & t_hi) | (r_lo & r_hi)",
            "result": true
        },
        "falsification": {
            "description": "The hypothesis is falsified if the
                            97.5% CI upper bound is smaller than
                            0.2 (the SESOI) for both trust and
                            reciprocation.",
            "evaluation": "!t_hi & !r_hi",
            "result": false
        },
    }
]
```

Analyses. Each analysis is specified in the `analysis` component (Box 5). An analysis consists of an `id` to reference the statistical test when evaluating the criteria and the `code` used to run the analysis. Once data are attached and the analyses are run, a list of named `results` can added (either manually or automatically by software such as scienceverse) to be referenced in the criteria. Each analysis can also contain additional information, such as the software used to perform the analysis. The example below specifies two *t*-tests, using the `t.test` function in R. In the working scienceverse prototype used in this manuscript, short analyses can be added directly, while longer analysis scripts that return a test result can be added by referencing an external analysis script.

Box 5. The analysis component.

```
"analyses": [
    {
        "id": "trust",
        "code": "    t.test(kin$trust_self, kin$trust_other,
                            paired = TRUE, conf.level = 0.975)",
        "software": "R version 4.0.2 (2020-06-22)",
        "results": {
            "statistic": 2.5045,
            "parameter": 23,
            "p.value": 0.0198,
            "conf.int": [0.0213, 0.9787],
            "estimate": 0.5,
            "null.value": 0,
            "stderr": 0.1996,
            "alternative": "two.sided",
            "method": "Paired t-test",
            "data.name": "kin$trust_self and kin$trust_other"
        }
    },
    {
        "id": "recip",
        "code": "    t.test(kin$recip_self, kin$recip_other,
                            paired = TRUE, conf.level = 0.975)",
        "software": "R version 4.0.2 (2020-06-22)",
        "results": {
            "statistic": -0.2138,
            "parameter": 23,
            "p.value": 0.8326,
            "conf.int": [-0.5089, 0.4256],
            "estimate": -0.0417,
            "null.value": 0,
            "stderr": 0.1949,
            "alternative": "two.sided",
            "method": "Paired t-test",
            "data.name": "kin$recip_self and kin$recip_other"
        }
    }
]
```

Data. Each dataset can be specified in the `data` component (Box 6). A dataset consists of an `id` to reference the dataset in analyses and other information such as how to obtain the data (e.g., `doi`, `url`). The `codebook` contains descriptions of each column, but it is even possible to include the `data` itself in this component. By storing the data underlying the reported analyses as nested lists in the same file together with good meta-data, a reported analysis could be completely reproduced in the future from a single file. Furthermore, it becomes very easy to perform additional analyses or sensitivity analyses on the data.

Box 6 contains a data component with a codebook created by scienceverse using the Psych-DS 0.1.0 format, which is currently still in development. The descriptors for each column can be arbitrarily detailed, or follow other meta-data formats. For other software that helps researchers to create and share machine-readable codebooks, see Arslan (2019).

Automatic Evaluation. Now that the prediction is specified in a machine readable format, it is possible for the statistical prediction to be evaluated automatically. Automatic evaluation of machine readable hypotheses has at least two useful functions during the peer review process. First, we foresee a future where researchers are required to submit fully computationally reproducible analysis scripts with their submissions. This will require editorial assistants or reviewers to check the computational reproducibility of the reported results in a manuscript. Machine-readable hypothesis tests would make this check a matter of running a single function. The scienceverse R package can do this for code written in R, and a machine-readable format makes it straightforward to create scripts that automatically run analyses in other languages.

Based on the information specified in the analyses, criteria, and data components, the `study_analyze` function in scienceverse reads in the analytic data, performs each analysis, and stores and evaluates the results. In the example above, running the `study_analyze` function will automatically load the data as the object "kin", and perform the "trust" analysis by running the analysis `t.test(x = kin$trust_self, y = kin$trust_other, paired = TRUE, conf.level = .975)`. The result of this analysis is automatically stored (e.g., the t.test function in R returns a list of named numbers, including "conf.int": [0.0213, 0.9787]). The criteria are then evaluated against the results of the analyses. For example, because the first number in the "conf.int" result (0.0213) is larger (">") than zero ("0"), the conclusion that this criterion is "true" will be stored (see Box 3).

After the `study_analyze` function has drawn conclusions about whether each criterion is met or not, based on the results of the analyses, the evaluation rules can be used to determine whether the prediction is corroborated, falsified, or neither (and thus the results are inconclusive). For the prediction to be corroborated, the criteria for "t_lo" and "t_hi" have to be met, and/or the criteria for "r_lo" and "r_hi" have to be met. Since the conclusions for "t_lo" and "t_hi" are both true, the prediction is corroborated, and because it is not true that both upper bounds for the confidence interval are smaller than 0.2, the prediction is not falsified. The overall `conclusion` is therefore that our statistical prediction is corroborated. It will typically be useful to create a human-readable summary. This can be done with the `study_save` function, which created output as presented in Figure 1 below. Such a human-readable summary would allow editorial assistants or reviewers to quickly check the computational reproducibility of the reported results.

Box 6. The data component.

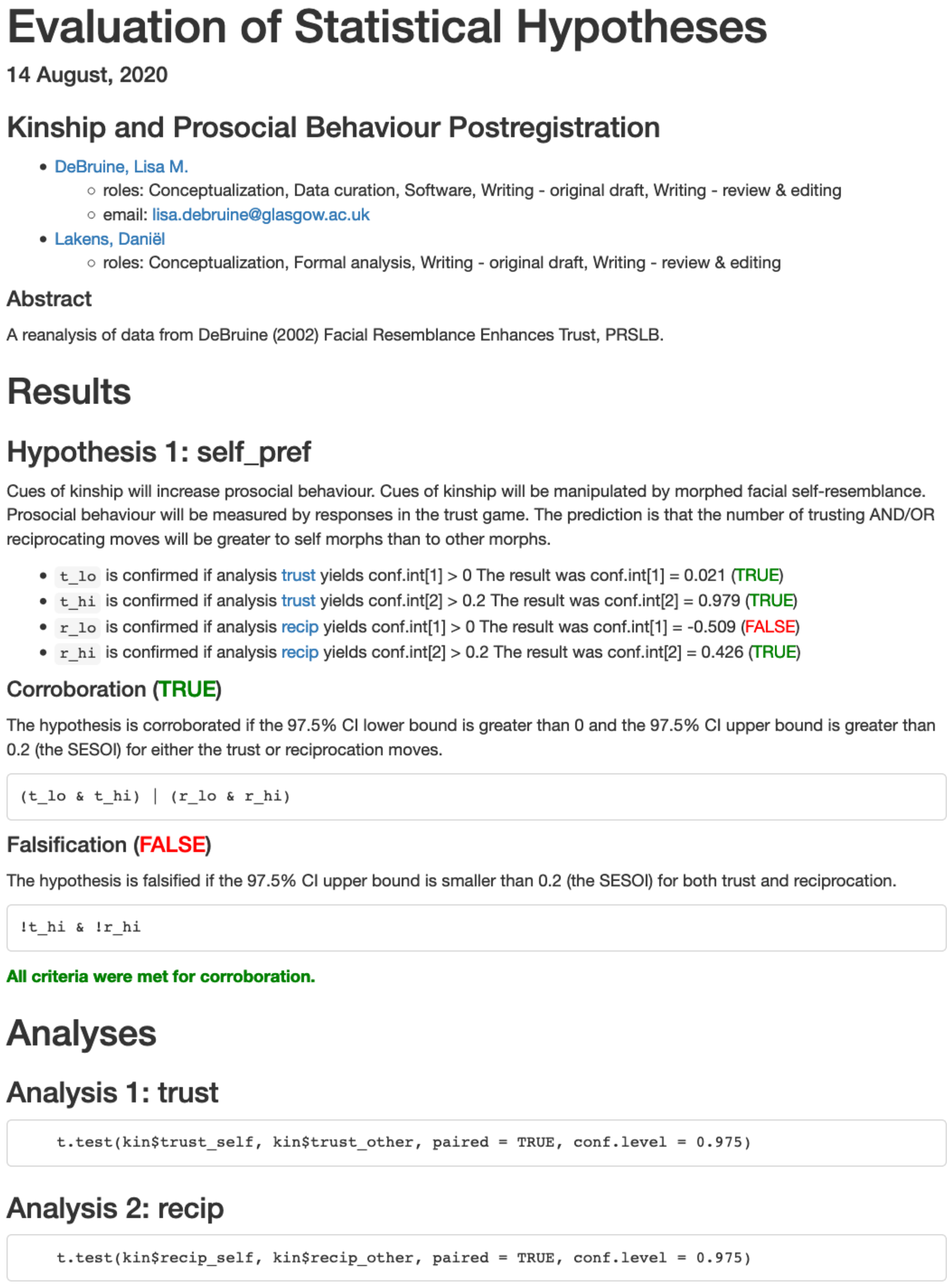
```
"data": [
    {
        "id": "kin",
        "codebook": {
            "@context": "https://schema.org/",
            "@type": "Dataset",
            "name": "kin",
            "schemaVersion": "Psych-DS 0.1.0",
            "url": "https://osf.io/ewfhs/",
            "variableMeasured": [
                {
                    "@type": "PropertyValue",
                    "name": "trust_self",
                    "description": "Trusting self-morphs",
                    "dataType": "int"
                },
                {
                    "@type": "PropertyValue",
                    "name": "trust_other",
                    "description": "Trusting other-morphs",
                    "dataType": "int"
                },
                {
                    "@type": "PropertyValue",
                    "name": "recip_self",
                    "description": "Reciprocating self-morphs",
                    "dataType": "int"
                },
                {
                    "@type": "PropertyValue",
                    "name": "recip_other",
                    "description": "Reciprocating other-morphs",
                    "dataType": "int"
                }
            ]
        },
        "data": {
            "trust_self": [1, 2, 2, 1, 1, 1, 1, 1, 2, 0, 2, 0,
                           1, 2, 2, 3, 2, 2, 1, 1, 2, 0, 0, 1],
            "trust_other": [1, 2, 2, 0, 1, 0, 0, 0, 1, 0, 1, 0,
                            1, 1, 1, 0, 1, 2, 2, 0, 0, 0, 2, 1],
            "recip_self": [0, 1, 3, 2, 1, 1, 1, 3, 3, 2, 3, 1,
                           1, 2, 3, 3, 3, 1, 1, 1, 3, 0, 3, 1],
            "recip_other": [1, 1, 2, 2, 3, 2, 1, 3, 3, 1, 3, 0,
                            1, 3, 3, 3, 3, 0, 3, 0, 1, 0, 3, 2]
        }
    }
],
```

Box 7. Results of data analysis.

```
"analyses": [
    {
        "id": "trust",
        ...
        "results": {
            "statistic": 2.5045,
            "parameter": 23,
            "p.value": 0.0198,
            "conf.int": [0.0213, 0.9787],
            "estimate": 0.5,
            "null.value": 0,
            "stderr": 0.1996,
            "alternative": "two.sided",
            "method": "Paired t-test",
            "data.name": "kin$trust_self and kin$trust_other"
        }
    },
    {
        "id": "recip",
        ...
        "results": {
            "statistic": -0.2138,
            "parameter": 23,
            "p.value": 0.8326,
            "conf.int": [-0.5089, 0.4256],
            "estimate": -0.0417,
            "null.value": 0,
            "stderr": 0.1949,
            "alternative": "two.sided",
            "method": "Paired t-test",
            "data.name": "kin$recip_self and kin$recip_other"
        }
    }
]
```

Benefits of Machine Readability

The example we describe above that uses the coding language R to specify analyses and our supplemental materials provide examples that use our R package, scienceverse. However, the use of R specifically, or any coding language, is not essential to the general idea of machine readable hypotheses. Much like the Brain Imaging Data Structure format (Gorgolewski et al., 2016), the proposed open format makes it possible to create data processing pipelines in any language. One can even create a JSON-formatted text file by hand in a text editor,



Evaluation of Statistical Hypotheses

14 August, 2020

Kinship and Prosocial Behaviour Postregistration

- DeBruine, Lisa M.
    - roles: Conceptualization, Data curation, Software, Writing - original draft, Writing - review & editing
    - email: lisa.debruine@glasgow.ac.uk
- Lakens, Daniël
    - roles: Conceptualization, Formal analysis, Writing - original draft, Writing - review & editing

Abstract

A reanalysis of data from DeBruine (2002) Facial Resemblance Enhances Trust, PRSLB.

Results

Hypothesis 1: self_pref

Cues of kinship will increase prosocial behaviour. Cues of kinship will be manipulated by morphed facial self-resemblance. Prosocial behaviour will be measured by responses in the trust game. The prediction is that the number of trusting AND/OR reciprocating moves will be greater to self morphs than to other morphs.

- `t_lo` is confirmed if analysis trust yields conf.int[1] > 0 The result was conf.int[1] = 0.021 (TRUE)
- `t_hi` is confirmed if analysis trust yields conf.int[2] > 0.2 The result was conf.int[2] = 0.979 (TRUE)
- `r_lo` is confirmed if analysis recip yields conf.int[1] > 0 The result was conf.int[1] = -0.509 (FALSE)
- `r_hi` is confirmed if analysis recip yields conf.int[2] > 0.2 The result was conf.int[2] = 0.426 (TRUE)

Corroboration (TRUE)

The hypothesis is corroborated if the 97.5% CI lower bound is greater than 0 and the 97.5% CI upper bound is greater than 0.2 (the SESOI) for either the trust or reciprocation moves.

```
(t_lo & t_hi) | (r_lo & r_hi)
```

Falsification (FALSE)

The hypothesis is falsified if the 97.5% CI upper bound is smaller than 0.2 (the SESOI) for both trust and reciprocation.

```
!t_hi & !r_hi
```

All criteria were met for corroboration.

Analyses

Analysis 1: trust

```
    t.test(kin$trust_self, kin$trust_other, paired = TRUE, conf.level = 0.975)
```

Analysis 2: recip

```
    t.test(kin$recip_self, kin$recip_other, paired = TRUE, conf.level = 0.975)
```

Figure 1. Example of machine readable output generated by scienceverse that shows the results and evaluation of the hypotheses.

and specify the result values manually. This could be a useful way to make the information in existing archives machine-readable, even if we don't have access to the original data or code. Currently, implementing machine readable hypothesis tests requires some effort, both in learning to specify explicit criteria for corroboration and falsification, as in programming knowledge to enter the meta-data. Future work should focus on making this process as easy as possible by providing detailed examples that users can follow, and by developing online forms that guide researchers through the creation of a scienceverse compatible JSON file.

We believe the benefits of making statistical predictions machine readable are worth the extra effort. First, machine-readable hypotheses remove ambiguity about what researchers predict and which criteria must be met to conclude a statistical hypothesis is corroborated. Predictions are explicitly linked to the tests that are performed to evalaute if the prediction is corroborated or not. The exact test is specified, which prevents flexibility in the data analysis. Furthermore, specifying the criteria for corroboration or falsification explicitly prevents future researchers who will replicate the study from having to infer which results would corroborate or falsify the original finding. Although machine readable hypotheses might feel extremely rigid, it is possible to specify a range of sensitivity analyses across which the prediction should hold.

Another benefit of making statistical hypotheses machine readable is that many important aspects of the hypothesis test become accessible, findable, and usable. This will benefit researchers in the future. We can imagine a utopian future where meta-data files such as the example in Boxes 1 to 7 are accessible by browsing to a website that consists of the DOI, appended by /meta (e.g., https://doi.org/10.1098/rspb.2002.2034/meta). Researchers can access these files to load all the information that is available about statistical predictions. For example, when a completely reproducible workflow is used, and data can be accessed as part of the meta-data file, the meta-data file should be sufficient to easily calculate or access effect sizes from the performed statistical tests for meta-analyses.

While making hypothesis tests machine readable can obviously not ensure that statistical predictions are sensible or logically coherent, the process of writing a machine-readable statistical prediction could have a secondary benefit of providing a well-structured framework to think through and specify all important aspects of a statistical prediction. This might not be easy. Researchers might find it difficult to specify all required components in advance, or to specify the ranges of results that would corroborate or falsify a prediction. Sometimes a research idea is not yet well-specified enough to be tested in a confirmatory hypothesis test. Hypothesis tests are an extremely formalized procedure to make a decision whether a prediction is corroborated or not. If researchers realize they are actually not yet ready to make a falsifiable statistical prediction when creating a machine-readable hypothesis test, we would consider this a benefit as well (Scheel, Tiokhin, Isager, & Lakens, 2020). Researchers might then decide to estimate the population effect size instead of testing a falsifiable prediction. Alternatively, they might decide to perform additional studies that allow them to make a more falsifiable prediction. Specifying exploratory analyses in a machine-readable way still has benefits such as clarifying the source of statistical values in a manuscript and providing values for meta-analysis.

Use Cases

Registered Reports. We realize that several aspects of our proposal to make hypothesis tests machine readable sound futuristic. At the same time, we believe immediate use cases for machine-readable hypothesis tests already exist in the form of the Registered Report publication format (Chambers, 2019). Registered Reports require researchers to clearly specify their statistical prediction, and are developed to reduce flexibility in the statistical analyses. After Stage 1 review based on the introduction, methods, and analysis plan, researchers can receive an 'in principle acceptance'. They then collect the data, and submit a Stage 2 Registered Report that includes the results and conclusion. This should make it relatively easy for reviewers to compare planned and reported analyses. Peer reviewers might not always have the time to carefully check whether each reported analysis in the manuscript matches the planned analysis in the preregistration, and whether the conclusions in the manuscript follow from the test results. A machine readable hypothesis test can automatically generate reports that facilitate peer review.

Furthermore, whereas submission guidelines for Registered Reports require researchers to specify their analyses, researchers are typically not required to explain in advance when they would consider their hypotheses corroborated or falsified, while doing so would make it easier for reviewers to evaluate the severity of a statistical test (Lakens, 2019). In Registered Replication Reports published in AMPPS, authors are asked to explicitly specify when a replication corroborates the original finding. For example, in the analysis plan of the Registered Replication Report on Fischer, Castel, Dodd, and Pratt (2003) by Colling et al. (2020), available at https://osf.io/6a2ny/, a clear decision rule for corroboration is specified: "*If the congruency effect is positive and statistically significantly different from zero in the 500 ms and 750 ms delay conditions but not statistically significantly different from zero in the 250 ms and 1000 ms delay conditions, we will consider the findings of Fischer et al. (2003) to be replicated within the limits they propose.*"

Scienceverse illustrates one possible workflow where, after specifying the hypotheses at a Stage 1 submission, a machine-readable report can be produced. This report looks similar to Figure 1, without any of the lines containing color-coded true or false evaluations of the predictions. When the data is collected, it can be added to the meta-data file generated at Stage 1, the preregistered analyses can then be run, and a human-readable report can be generated as in Figure 1. This should make it relatively easy for reviewers to compare planned and reported analyses.

Power Analyses. To check the code in a preregistration, the scienceverse package has a function to simulate datasets by specifying the data structure for factorial designs (using the R-package faux, DeBruine, 2020). Another function generates a specified number of simulations, runs the analyses using the automatic evaluation procedure described above, and reports the total number of simulations for which each hypothesis was corroborated, falsified, or inconclusive. We provide an R script with an extended example of the study above that includes a power analysis in the supplemental materials (see the Open Practices section).

Meta-analyses. Researchers face several challenges when they want to examine research lines with meta-analytic techniques such as effect size meta-analysis, p-curve analysis (Simonsohn, Nelson, & Simmons, 2014), or z-curve analysis (Brunner & Schimmack, 2020).

First, many scientific papers do not report the results of statistical tests in sufficient detail to include these studies in a meta-analysis. Effect sizes are often not computed, and although researchers performing a meta-analysis can attempt to manually calculate effect sizes, this requires access to the means, standard deviations, correlations for within comparisons, and exact sample sizes for each condition, which are also often missing. Effect sizes can sometimes still be approximated from test statistics, but these are often not reported for non-significant results. The second problem a researcher performing a meta-analysis faces is a lack of transparency about which statistical test in the results section is related to the theoretical predictions in the introduction section. This can make it difficult to select the appropriate test to include in a meta-analysis.

The structured meta-study files we propose solve both these problems, as long as researchers 1) include the raw data in the meta-study file, and 2) specify for each hypothesis which statistical test result(s) will corroborate or falsify the predictions. In the online vignettes (see Open Practices section), we demonstrate how a z-curve and *p*-curve analysis can easily be performed based on the *p*-values stored in the results section of the meta-study file, and how the raw data across meta-study files can be used to identify shared variables across data sets and compute and analyze effect sizes in a meta-analysis. As meta-analyses will almost always include data from published papers that have no meta-data available, unless a concerted effort is made to catalog all published studies (for a noteworthy example, see Bosco, Field, Larsen, Chang, and Uggerslev (2020)), these benefits will at best apply to a subset of the studies included in a meta-analysis.

Conclusions

Technological innovation makes it possible to communicate scientific findings in digital formats that allow for much easier re-use of scientific information contained in these digital files compared to traditional journal articles. As we move towards a time where researchers are expected to share their data in a way that is FAIR (findable, accessible, interoperable, and reusable), we believe it is feasible and beneficial to make the rest of research machine readable as well. We see machine-readable hypothesis tests as a logical development, with immediate benefits for the rigour of hypothesis tests. Increasing the accessibility of essential information related to hypothesis tests in scientific papers will also facilitate peer review, especially of Registered Reports, and facilitate meta-scientific research. Making statistical predictions machine readable will be an important next step towards a scientific literature that can be accessed not just visually, but also computationally.

Author Contributions

Both authors conceptualized the main idea, LMD wrote the Scienceverse software, and both authors wrote and revised this manuscript.

Daniel Lakens https://orcid.org/0000-0002-0247-239X

Lisa DeBruine https://orcid.org/0000-0002-7523-5539

Research Software

This paper and supplemental materials use the following open-source research software: R Core Team (2019); Wickham (2017); Bartoš and Schimmack (2020); Viechtbauer (2010); DeBruine (2020); Aust and Barth (2018); DeBruine and Lakens (2020).

Acknowledgements

We would like to thank Leo Tiokhin and Peder Isager for feedback on an earlier draft of this manuscript, and attendants of a hackathon at the Society for the Improvement of Psychological Science for their enthusiastic reception of the ideas behind machine-readable hypotheses.

Declaration of Conflicting Interests

The author(s) declared that there were no conflicts of interest with respect to the authorship or the publication of this article.

Funding

LMD is supported by European Research Council grant #647910. DL is funded by VIDI Grant 452-17-013 from the Netherlands Organisation for Scientific Research.

Open Practices

The code to reproduce this manuscript is available at https://github.com/scienceverse/machine-readable and the scienceverse R package is available from https://github.com/scienceverse/scienceverse/. Vignettes are available from https://scienceverse.github.io/scienceverse/articles/index.html.

References

Arslan, R. C. (2019). How to automatically document data with the codebook package to facilitate data reuse. *Advances in Methods and Practices in Psychological Science*, *2*(2), 169–187. https://doi.org/10.1177/2515245919838783

Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown.* Retrieved from https://github.com/crsh/papaja

Bartoš, F., & Schimmack, U. (2020). Zcurve: An r package for fitting z-curves. Retrieved from https://CRAN.R-project.org/package=zcurve

Bosco, F. A., Field, J. G., Larsen, K. R., Chang, Y., & Uggerslev, K. L. (2020). Advancing meta-analysis with knowledge-management platforms: Using metaBUS in psychology. *Advances in Methods and Practices in Psychological Science*, *3*(1), 124–137.

Brunner, J., & Schimmack, U. (2020). Estimating population mean power under conditions of heterogeneity and selection for significance. *Meta-Psychology*, *4*, 1–22. https://doi.org/10.15626/MP.2018.874

Chambers, C. (2019). What's next for registered reports? *Nature*, *573*, 187–189. https://doi.org/10.1038/d41586-019-02674-6

Colling, L. J., Szűcs, D., De Marco, D., Cipora, K., Ulrich, R., Nuerk, H.-C., … McShane, B. B. (2020). Registered Replication Report on Fischer, Castel, Dodd, and Pratt (2003). *Advances in Methods and Practices in Psychological Science*, *3*(2), 143–162. https://doi.org/10.1177/2515245920903079

DeBruine, L. (2002). Facial resemblance enhances trust. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *269*, 1307–1312. https://doi.org/10.1098/rspb.2002.2034

DeBruine, L. (2020). *Faux: Simulation for factorial designs.* Zenodo. https://doi.org/10.5281/zenodo.2669586

DeBruine, L., & Lakens, D. (2020). *Scienceverse: Machine-readable study descriptions.* Retrieved from https://github.com/scienceverse/scienceverse

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., … Poldrack, R. A. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Nature Scientific Data*, *3*(160044). https://doi.org/10.1038/sdata.2016.44

Lakatos, I. (1978). *The methodology of scientific research programmes: Volume 1: Philosophical papers.* Cambridge University Press.

Lakens, D. (2019). The value of preregistration for psychological science: A conceptual analysis. *Japanese Psychological Review*, *62*(3), 221–230.

Lakens, D., Scheel, A. M., & Isager, P. M. (2018). Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological*

Science, *1*(2), 259–269. https://doi.org/10.1177/2515245918770963

Meehl, P. E. (1990). Appraising and amending theories: The strategy of lakatosian defense and two principles that warrant it. *Psychological Inquiry*, *1*(2), 108–141.

Neyman, J., & Pearson, E. S. (1933). On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, *231*(694-706), 289–337. https://doi.org/10.1098/rsta.1933.0009

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Scheel, A. M., Tiokhin, L., Isager, P. M., & Lakens, D. (2020). Why hypothesis testers should spend less time testing hypotheses. *Perspectives on Psychological Science*. https://doi.org/10.31234/osf.io/vekpu

Simonsohn, U., Nelson, L. D., & Simmons, J. P. (2014). P-curve and effect size: Correcting for publication bias using only significant results. *Perspectives on Psychological Science*, *9*(6), 666–681. https://doi.org/10.1177/1745691614553988

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, *36*(3), 1–48. Retrieved from https://www.jstatsoft.org/v36/i03/

Wickham, H. (2017). *Tidyverse: Easily install and load the 'tidyverse'*. Retrieved from https://CRAN.R-project.org/package=tidyverse