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

Lisa M. DeBruine[1] & Daniël Lakens[2]
Affiliation

Abstract

In many scientific fields researchers rely on hypothesis tests to determine whether empirical observations support predictions. In a well-specified hypothesis test, a theoretical 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 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 supported. 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. 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 supported, falsified, or that the results are inconclusive.

We propose that the gold standard in well-specified hypothesis test should be a statistical prediction that is machine readable. This means that a computer can evaluate whether a statistical prediction is supported 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. In this manuscript we demonstrate how statistical predictions can be made machine readable. We believe that there are two broad arguments for a move to machine readable hypothesis tests. The

Lisa DeBruine https://orcid.org/0000-0002-7523-5539
Daniël Lakens https://orcid.org/0000-0002-0247-239X
Correspondence concerning this article should be addressed to Lisa M. DeBruine, 62 Hillhead Street, Glasgow, G12 8QB. E-mail: lisa.debruine@glasgow.ac.uk

first argument is that by specifying hypothesis tests in a format that can be read and evaluated by a machine, tests of predictions and the conclusions derived from these tests will become transparent, statistically falsifiable, and rigorous. This will help to alleviate poor practices in how scientists currently 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 meta-analyses), and find and access this information (e.g., to answer meta-scientific questions about the proportion of statistical results in the scientific literature that support the prediction).

Poor practices when testing predictions

As a concrete example of a typical hypothesis test in the published literature, DeBruine (2012) 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 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 the two-person bargaining game. From this, we can conclude that the evaluation rule to determine if the statistical prediction was confirmed or not was whether the morphing manipulation had an effect on either the trust measure, or the reciprocation measure.

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 it becomes clear what the author inferred as support, but this is never clearly specified before the conclusion is reported. The second problem is that is is not clearly specified what would support the hypothesis and what would statistically falsify the hypothesis. We can infer that

the prediction is supported when either of the two tests is significant at an alpha level of 0.05, without correcting for multiple comparisons, but this is not explicitly stated. 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) and the analysis strategy does not follow unequivocally from the introduction and methods section.

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). A theoretical prediction is operationalized in a specific study, and the operationalized prediction is evaluated based on a statistical prediction.

We distinguish the following components of a statistical prediction: 1) a statistical test, 2) a test result, and 3) one or more criterion values that the test result is compared to. 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). 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 (Dienes, 2019; Neyman, Pearson, & Pearson, 1933).

As highlighted above, one limitation of the current practice is that researchers often do not explicitly state what would support or falsify their prediction. To be able to unambiguously evaluate a hypothesis, researchers need to specify evaluation rules for when they will interpret statistical test results as support for a prediction, falsification of a prediction, or when a result will be treated as inconclusive. There are different statistical approaches that can be used to statistically conclude a prediction is falsified.

One approach, known as equivalence testing (Lakens, Scheel, & Isager, 2018), 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 supported when we can statistically conclude

the observed mean difference is greater than zero, and not statistically smaller than the smallest effect size we care about. The prediction would be falsified if the effect is statistically smaller than the smallest effect size of interest, and inconclusive if we can neither conclude the 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 2.5% alpha level), and our smallest effect size of interest is 0.2, then we can conclude that we have supported our prediction if the lower bound of our 97.5% CI is larger than 0 and the upper bound is not smaller than 0.2. Our prediction is falsified if the upper bound of our 97.5% CI is 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 supported by the data. We envision machine readable hypothesis tests are part of a completely reproducible workflow. Computer scripts will load the raw data, and if needed prepare the analytic data from the raw data (e.g., outlier removal, transformations, computing sum scores). The statistical test is automatically run on the 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 supported, falsified, or inconclusive.

Box 1 provides a concrete example of a machine-readable statistical prediction for the study by DeBruine (2002) described above. It is written using JSON, which is an open-standard file format that can be used to transmit data. Because it is an open-standard file format, it can easily be converted into any other data file format such as YAML or JATS, 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 prediction.

The top level list 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 aspects of a study that we would like to be able to retrieve. Here, we will focus on the aspects of the study that are required to make statistical predictions machine readable, for which we need to specify the hypotheses, the analyses, and the evaluation criteria for each prediction.

A study could contain multiple hypotheses, but our example contains only one. Each hypothesis (Box 1, lines 3-50) consists of a verbal human-readable description (5), one or more criteria (6-39) to evaluate analysis results, and rules to determine if the results support (40-44) or falsify (45-49) the hypothesis that is automatically generated if the data is available. Criteria need an id to be able to reference them in the evaluations. 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 (7-14) specifies that if the statistical result “conf.int[1]” from “trust_analysis” is “>” than “0” then the criterion “trust_lowbound” evaluates to “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 supported. The support (40-44) and falsify (45-49) sub-components describe rules to determine support or falsification from the criteria conclusions, and consist of three elements. The description element contains verbal descriptions of the decision rules for concluding the hypothesis is supported or falsified. The evaluation element contains a logical version referencing the criteria IDs. For example, "`(trust_lowbound & trust_highbound) | (recip_lowbound & recip_highbound)`" (42) means that the support conclusion (43) will be set to "true" if the first two criteria are both true, or if the last two criteria are both true, while "`!trust_highbound & !recip_highbound`" (47) means that the falsify conclusion (48) will be set to "true" if both of these criteria are false.

Each analysis is specified in the analysis component (52-81). An analysis consists of an id to reference the statistical test when evaluating the criteria, a reference to the function or script used to run the analysis (func), a list of parameters that need to be specified in the function (params), and a list of named results 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 (version 3.6.0). For each t-test, we need to specify values for "x" and "y" (columns in a dataset called "kin"), "paired" (the type of t-test), and "conf.level" (to change the default value for this test to .99).

Each dataset can be specified in the data component (82-112). A dataset consists of an id to reference the dataset in analyses and information about how to obtain the data (e.g., doi, url). The codebook (87-104) contains descriptions of each column, and it is even possible to include the data values (105-110) in this component. Below, we present a simple version of a codebook, but the descriptors for each column can be arbitrarily detailed and automatically extracted from existing data structures, such as SPSS files.

Box 1. Example JSON file illustrating a machine-readable statistical prediction.

```
 1. {
 2.   "name": "Kinship and Prosocial Behaviour",
 3.   "hypotheses": [
 4.     {
 5.       "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.",
 6.       "criteria": [
 7.         {
 8.           "id": "trust_lowbound",
 9.           "analysis_id": "trust_analysis",
10.           "result": "conf.int[1]",
```

```
          "operator": ">",
          "comparator": 0,
          "conclusion": true
        },
        {
          "id": "trust_highbound",
          "analysis_id": "trust_analysis",
          "result": "conf.int[2]",
          "operator": ">",
          "comparator": 0.2,
          "conclusion": true
        },
        {
          "id": "recip_lowbound",
          "analysis_id": "recip_analysis",
          "result": "conf.int[1]",
          "operator": ">",
          "comparator": 0,
          "conclusion": false
        },
        {
          "id": "recip_highbound",
          "analysis_id": "recip_analysis",
          "result": "conf.int[2]",
          "operator": ">",
          "comparator": 0.2,
          "conclusion": false
        }
      ],
      "support": {
        "description": "The hypothesis is supported 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": "(trust_lowbound & trust_highbound) | (recip_lowbound &
         recip_highbound)",
        "conclusion": true
      },
      "falsify": {
        "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": "!trust_highbound & !recip_highbound",
        "conclusion": false
      }
    }
```

```
  ],
  "analyses": [
    {
      "id": "trust_analysis",
      "software": "R version 3.6.0 (2019-04-26)",
      "func": "t.test",
      "params": {
        "x": "kin$trust_self",
        "y": "kin$trust_non",
        "paired": true,
        "conf.level": 0.975
      },
      "results": {
        "conf.int": [0.02, 0.98]
      }
    },
    {
      "id": "recip_analysis",
      "software": "R version 3.6.0 (2019-04-26)",
      "func": "t.test",
      "params": {
        "x": "kin$recip_self",
        "y": "kin$recip_non",
        "paired": true,
        "conf.level": 0.975
      },
      "results": {
        "conf.int": [-0.51 0.43]
      }
    }
  ],
  "data": [
    {
      "id": "kin",
      "doi": "10.17605/OSF.IO/F7QWS",
      "url": "https://osf.io/ewfhs/",
      "codebook": [
        {
          "name": "trust_self",
          "description": "Number of trusting moves towards self-morphs"
        },
        {
          "name": "trust_other",
          "description": "Number of trusting moves towards self-morphs"
        },
```

```
 96.         {
 97.           "name": "recip_self",
 98.           "description": "Number of reciprocating moves towards other-morphs"
 99.         },
100.         {
101.           "name": "recip_other",
102.           "description": "Number of reciprocating moves towards other-morphs"
103.         }
104.       ],
105.       "values": {
106.         "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],
107.         "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],
108.         "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],
109.         "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]
110.       }
111.     }
112.   ]
113. }
```

The statistical prediction can be evaluated automatically, without human intervention, as long as the prediction is specified in a machine readable format. Based on the information specified in the analyses and criteria, a computer script can read in the data, perform each analysis specified in the JSON file, and store the results. For example, the "trust_analysis" can be performed by running the analysis `t.test(x = kin\$trust_self, y = kin$trust_other, paired = TRUE, conf.level = .975)` after the data is loaded as an object named "kin". We store the result of this analysis (e.g., the t.test function in R returns a list of named numbers, including "conf.int": [0.02, 0.98] in line 63). Based on the results of the analyses, the code can compare the results against the criteria. For example, because the first number in the "conf.int" result (0.02) is larger (">") than zero ("0"), we can store the conclusion that this criterion is "true" (line 13). After the code 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 supported, falsified, or neither (and thus the results are inconclusive). For the prediction to be supported, the criteria for trust_lowbound and trust_highbound have to be met, and/or the criteria for recip_lowbound and recip_highbound have to be met. Since the conclusions for trust_lowbound and trust_highbound are both true, the prediction is supported, 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 supported.

Benefits of Machine Readability

Are the benefits of making statistical predictions machine readable worth the extra effort? We believe so. First, machine-readable hypotheses completely remove ambiguity about what researchers predict and which criteria must be met to conclude a statistical

prediction hypothesis is supported. Although ambiguity can be removed in other ways, the formalized approach as illustrated in Box 1 is perfectly suited for this goal. Predictions are explicitly linked to the tests that are performed to decide if the prediction is supported or not, and the exact test is specified, which prevents flexibility in the data analysis. This method prevents researchers who will replicate the study from having to infer the criteria for support or falsification from the observed pattern of results and the conclusion. For example, it may be that both ordinal and disordinal interactions are consistent with the hypothesis, or that only ordinal interactions are.

While this method obviously cannot ensure that the logic behind predictions or criteria for support or falsification are correct, the process of writing a machine-readable statistical prediction has a secondary benefit of providing a framework for thinking through the logic of a prediction. If one finds it impossible to specify the ranges of results that will support or falsify a prediction, it is likely that the hypothesis is not yet well-specified enough for confirmatory hypothesis testing. Hypothesis tests are extremely formalized procedures to test predictions. Making this clear might make researchers realize they are not yet ready to test a hypothesis, and remind them of the importance and value of exploratory research.

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 Box 1 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. When a completely reproducible workflow is used, and data can be accessed, accessing the meta-data file could be sufficient to easily calculate effect sizes from the performed statistical tests for meta-analyses, and answer meta-scientific questions about, for example, the percentage of statistical predictions that are supported in a research area.

One more immediate use case for machine readable hypothesis tests is 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. Reviewers need to evaluate whether the analyses were conducted exactly as planned and whether conclusions follow from the predictions. This aspect of the peer review process is labor intensive and often ambiguous, but could be automated if the statistical predictions were machine readable.

Conclusions

Technological innovation makes it possible to communicate scientific information in digital formats that facilitate the use of this information. It is becoming increasingly accepted to expect that researchers share their data in a way that is FAIR (findable, accessible, interoperable, and reusable). This can only be accomplished by providing meta-

data alongside the data files, and software that helps researchers to share machine-readable meta-data is being developed (e.g., Arslan, 2019). We believe it is feasible and beneficial to make statistical predictions machine readable as well. This is a logical next step, with immediate benefits for the rigour of hypothesis tests, and the accessibility of important information related to hypothesis tests in scientific paper. Making statistical predictions machine readable is a logical next step towards a scientific literature that can be accessed not just visually, but computationally.

Acknowledgements

Author Contributions.

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.

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. doi:10.1177/2515245919838783

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

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

Dienes, Z. (2019). How do I know what my theory predicts? *Advances in Methods and Practices in Psychological Science*, *2*(4), 364–377. doi:10.1177/2515245919876960

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

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. doi:10.1177/2515245918770963

Neyman, J., Pearson, E. S., & Pearson, K. (1933). IX. 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. doi:10.1098/rsta.1933.0009