



---

# *Journal of Statistical Software*

January 2019, Volume 88, Issue 2.

doi: 10.18637/jss.v088.i02

---

## **JASP: Graphical Statistical Software for Common Statistical Designs**

**Jonathon Love**  
University of Newcastle

**Ravi Selker**  
University of Amsterdam

**Maarten Marsman**  
University of Amsterdam

**Tahira Jamil**  
University of Amsterdam

**Damian Dropmann**  
University of Amsterdam

**Josine Verhagen**  
University of Amsterdam

**Alexander Ly**  
University of Amsterdam

**Quentin F. Gronau**  
University of Amsterdam

**Martin Šmíra**  
Masaryk University

**Sacha Epskamp**  
University of Amsterdam

**Dora Matzke**  
University of Amsterdam

**Anneliese Wild**  
University of Amsterdam

**Patrick Knight**  
University of Amsterdam

**Jeffrey N. Rouder**  
University of California, Irvine  
University of Missouri

**Richard D. Morey**  
Cardiff University

**Eric-Jan Wagenmakers**  
University of Amsterdam

---

### **Abstract**

This paper introduces **JASP**, a free graphical software package for basic statistical procedures such as  $t$  tests, ANOVAs, linear regression models, and analyses of contingency tables. **JASP** is open-source and differentiates itself from existing open-source solutions in two ways. First, **JASP** provides several innovations in user interface design; specifically, results are provided immediately as the user makes changes to options, output is attractive, minimalist, and designed around the principle of progressive disclosure, and analyses can be peer reviewed without requiring a “syntax”. Second, **JASP** provides some of the recent developments in Bayesian hypothesis testing and Bayesian parameter estimation. The ease with which these relatively complex Bayesian techniques are available in **JASP** encourages their broader adoption and furthers a more inclusive statistical reporting practice. The **JASP** analyses are implemented in R and a series of R packages.

*Keywords:* **JASP**, statistical software, Bayesian inference, graphical user interface, basic statistics.

---

## 1. Introduction

The speed of scientific progress greatly benefits from the availability of free and open-source software. The need for such software is particularly acute in the arena of statistical (re)analysis, for at least three reasons. First, the availability of the software’s source code enables researchers to probe the adequacy of the underlying algorithms to establish whether or not a program’s results are trustworthy. Second, open-source software ensures the freedom of researchers to extend and adjust existing work. Finally, open-source software can be shared freely, from one researcher to the next, without cost. In contrast, proprietary software is generally provided without source code for review, without permission to modify it, and with substantial licensing costs.

Unfortunately, proprietary software continues to be entrenched in the field of statistics, something which holds true especially for “basic statistics”, that is, statistical methods for widely used procedures such as  $t$  tests, ANOVAs, linear regression, and contingency tables. The canonical example of a proprietary statistical software package is SPSS (IBM Corporation 2013). The SPSS program has a long history and is popular in education as well as in empirical research. A Google Scholar search for academic articles that contain the term “SPSS” produces over 100,000 results for the year 2014 alone. Given that SPSS is just one of many proprietary statistical packages used within science, this suggests that proprietary software maintains a substantial market share. This suggestion is corroborated by the Reproducibility Project (Open Science Collaboration 2015), a large-scale project that sought to replicate 100 high-profile psychology studies; out of the 100 replication attempts carried out by different psychology groups around the world, as many as 84 used proprietary software for their data analysis (<https://osf.io/ezcuw/wiki/home/>).

Thus, the current state of affairs in the field of software for basic statistics is decidedly suboptimal: Proprietary software remains entrenched, despite the clear and present drawbacks both for statistical education and for scientific progress. At the same time, it is obvious that science stands to gain considerably from adopting open-source alternatives, as this (1) enables peer review of the software; (2) facilitates the development and adoption of new statistical methods; and (3) reduces costs to the scientific and university communities.

To further the use of open-source statistical software, a number of programs and environments have been developed, the most high profile of these being the R programming language (R Core Team 2018). Although R is the program of choice for statisticians and methodologists, its command line interface and steep learning curve are sometimes believed to be a barrier to its broader adoption. This concern is most pressing for students and applied researchers, that is, for people who use statistics only occasionally, and for basic problems (Valero-Mora and Ledesma 2012). These occasional users do not require the full flexibility that R has to offer, and their statistical demands can be met by software with a graphical user interface. In light of this, a number of projects have developed graphical open-source statistical software for basic statistics. A select list includes PSPP (The GNU Foundation 2015), SOFA (Paton-Simpson & Associates 2015), RKWard (Rödiger *et al.* 2012), Deducer (Fellows 2012), and R Commander (Fox 2005). Continuing in this tradition, we have developed JASP: a graphical, open-source statistical platform for performing common statistical tasks, designed to be simple and intuitive to use, and available for Windows, Mac OS X and Linux.

There are two features that set JASP apart from existing software. First, JASP provides a number of innovations in user interface design. These include providing results to the user

in real-time as they make changes, providing minimalist, attractive output which is ready to publish, and allowing the user to see exactly what interface options were used to specify an analysis, without requiring the use of a “syntax”. Indeed, we believe that **JASP**’s unique user interface approach might inspire similar efforts in the future. Second, **JASP** provides a series of Bayesian analyses for basic statistical tests, featuring both parameter estimation and default Bayes factor hypothesis testing (Jeffreys 1961; Kass and Raftery 1995). These Bayesian analyses echo their classical counterparts, thereby encouraging the exploration of alternative statistical methodology and a more inclusive style of statistical reporting. To date, the Bayesian methods have not been made available in graphical statistical software.

The outline of this article is as follows. We first list the analyses available in **JASP** and describe the **JASP** user interface. We then explain **JASP**’s unique design philosophy and illustrate the use of **JASP** with a concrete example. Subsequently we explain how **JASP** is implemented.

## 2. Available analyses

The development of **JASP** started in 2013 with support from a grant from the European Research Council. As of version 0.7 (September 2015), **JASP** provides descriptive statistics along with the following basic analysis methods:

- $t$  tests for one-sample, paired, and grouped designs.
- ANOVA for grouped and repeated measures designs.
- ANCOVA.
- Linear regression.
- Tests of correlation.
- Contingency tables.

As mentioned above, **JASP** not only provides the classical implementation of these tests but also features their Bayesian equivalents. The Bayesian  $t$  tests are based on the work of Jeffreys (1961, see also Ly *et al.* 2016; Rouder *et al.* 2009; Wetzels *et al.* 2009). The Bayesian ANOVA, ANCOVA, and linear regression are based on the work of Liang *et al.* (2008, see also Rouder and Morey 2012; Rouder *et al.* 2012; Wetzels *et al.* 2012). The Bayesian tests of correlation are based on the work of Jeffreys (1961, see also Ly *et al.* 2016), and the Bayesian contingency tables are based on the work of Gunel and Dickey (1974, see also Jamil *et al.* 2017). For many applied researchers, these tests represent the workhorses of their discipline. Future releases of **JASP** will include additional procedures such as log linear regression and logistic regression.

## 3. The JASP user interface

The **JASP** user interface is depicted in Figure 1. The **JASP** user interface is divided down the middle, with data visible in spreadsheet form to the left, and the results visible to the right. Analyses are available from the menus along the top, and selecting these produces

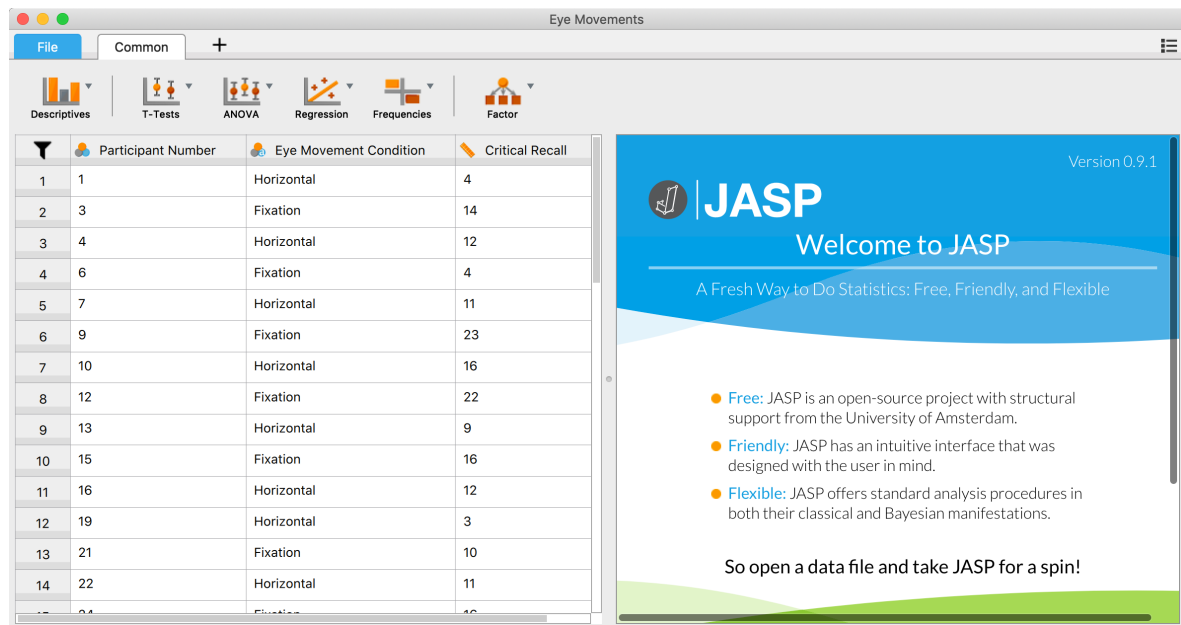


Figure 1: A screenshot of the **JASP** user interface.

output that is displayed in the right hand panel. The output of a new analysis is appended immediately below the outcome of the preceding analysis, a property that will be explored further in the worked example provided in Section 5.

## 4. JASP's design philosophy

As indicated above, **JASP** is not the first project to provide open-source, graphical software for performing basic statistics. One feature that sets **JASP** apart from these earlier software packages is its unique design philosophy, consisting of three principles that are briefly discussed below.

### 4.1. Design principle 1: Immediate feedback

**JASP** is designed around the principle of immediate or direct feedback. In this approach, the user interface responds immediately to user input. For example, creating an analysis in **JASP** produces a results table immediately, even before all the options have been fully specified. As the user makes changes to the options, the results reflecting these new options automatically appear in the right panel. In contrast, changing an analysis in a typical statistical package requires discarding the previous output and re-running it. **JASP** saves the user from having to continually perform this “book-keeping” task. This approach also invites exploration; for example, students can easily explore the way that different options affect the results. Additionally, immediate feedback is also “forgiving”; if the user mis-specifies some part of the analysis, they need only change that option to the correct value, and the results will be corrected. In contrast, typical statistical software requires that the user discards their earlier analysis, and re-runs it. In sum, the principle of immediate feedback allows **JASP** to prevent redundant output and clarifies the relation between statistical input and output.

## 4.2. Design principle 2: Attractive, minimalist output

Considerable effort was invested to have **JASP** produce tables and graphs that are attractive and clean, containing only the relevant information. Specifically, **JASP** produces tables in American Psychological Association (APA) format. These tables are attractive, easy to read, and suitable for publication “as is”. Indeed, it is possible to simply copy and paste the results from **JASP** into a graphical word processor such as Microsoft Word or LibreOffice. In contrast, many existing statistical packages do not produce appropriately formatted tables, and therefore require the user to perform an additional step such as manually transposing the values from the output into their writing software. This activity is not just time consuming but also error prone. Similarly, **JASP** produces graphs that are clean and easy to understand. Just as the tables, these graphs can be copied and pasted into a writing software and offered for publication as is.

Additionally, analysis results in **JASP** are typically minimalist. A classical  $t$  test might produce only the  $t$  statistic, degrees of freedom, and a  $p$  value. The user can digest these values and then decide whether they would like to test an assumption or see an additional value such as effect size. Upon selecting an additional option, **JASP** immediately adds another column to the results table, or adds an assumption table. The user can then consider and digest these new values. In this way **JASP** allows for *progressive disclosure* of results. Users can gradually build up the results at their own pace, and in the way that interests them, preventing the situation where an analysis may produce so many results that the user is overwhelmed.

In sum, **JASP** presents statistical output in a way that is easy to digest and publish. Attractive, clean tables and graphs are publication-ready. Furthermore, **JASP** output is additionally minimalist, allowing for progressive disclosure of results, thereby reducing the possibility for confusion.

## 4.3. Design principle 3: Perfect transparency

In science, it is often desirable to re-examine statistical analyses that were performed previously. This need can arise in communication between different researchers, such as when a reviewer or collaborator wishes to know the exact input options by which a particular output was produced; but the need is also present for an individual researcher who does not have perfect memory and may well forget the details of an analysis conducted months or even years in the past.

To address this need, statistical software often provides a past record of the options that were specified in conducting particular analysis. Typically, graphical statistical packages construct this record in the following way:

1. The user specifies the options for the analysis using the user interface.
2. An intermediate syntax is generated from the user interface options.
3. This intermediate syntax is passed to some sort of “statistical engine” to perform the analysis.

Users wishing to keep a record of the analysis options retain the intermediate syntax from step 2. Those who wish to review the analysis at a later point in time are required to “decode” and

understand the intermediate syntax; it is typically impossible to automatically return to the user interface options from step 1. This can be problematic, because users often understand analyses in terms of the user interface options from step 1, and the intermediate syntax is something they are not necessarily familiar with. Those wishing to review analyses are then forced to learn the intermediate syntax in order to work backwards to what the user interface options would have been.

In contrast, **JASP** eliminates the need for the intermediate syntax in step 2. In order to examine what options were used to create a particular analysis, a user simply selects the results in the output panel, and the user interface for that analysis will reappear with all of the buttons, lists, and check boxes populated with what was originally specified for the analysis. Hence, the user does not need to decode a syntax to understand which options were selected. Moreover, the user can then go on to make subsequent changes to the old analysis, picking up exactly where they left off. In addition, the user can save the statistical analyses in the **JASP** format, a format that includes the data, the input options, and the resulting output. This creates a perfectly transparent, immediately accessible record which facilitates collaboration, review, adjustment, and storage. Hence, **JASP** addresses the need for a permanent analysis record without the need for an underlying syntax.

It should be acknowledged that some researchers may prefer software that outputs intermediate syntax, as it allows them to learn the underlying language. For example, **R Commander** (Fox 2005), **Deducer** (Fellows 2012), and **RKward** (Rödiger *et al.* 2012) all output intermediate R code. This has the advantage that people can be progressively exposed to, and come to learn the R programming language. In contrast, teaching people how to use R is not a priority for the **JASP** project, as there are already many excellent tools available.

In sum, **JASP** yields transparent, reproducible analyses without requiring any syntax whatsoever; by selecting a particular component of the output, the user is immediately presented with the input options that gave rise to it. This way entire sequences of analyses can be saved, shared, adjusted, and retrieved, without requiring that users learn and understand syntax.

## 5. JASP example

The following example illustrates the functionality of **JASP** through a  $t$  test example based on an empirical data set. We use **JASP** to reanalyze a subset of the data from a recent adversarial collaboration that focused on the ostensibly beneficial effects of horizontal eye movements on memory (Matzke *et al.* 2015). Matzke and colleagues presented participants with a list of study words for a subsequent free-recall memory test (i.e., a test that requires participants to recall as many words from a study list, regardless of their order). Immediately following the study phase, one group of participants was requested to fix their gaze on a dot in the middle of the screen, whereas the other group was requested to perform a short series of horizontal eye movements. The hypothesis under scrutiny holds that the number of correctly recalled words is higher in the horizontal condition than in the fixation condition; consequently Matzke and colleagues tested this hypothesis with a one-sided independent samples Bayesian  $t$  test (e.g., Wetzels *et al.* 2009). The complete data set is available on the Open Science Framework at <http://openscienceframework.org/project/pXT3M/>. Here we reanalyze the free recall data using the **JASP** implementation of a classical  $t$  test as well as a Bayesian  $t$  test.

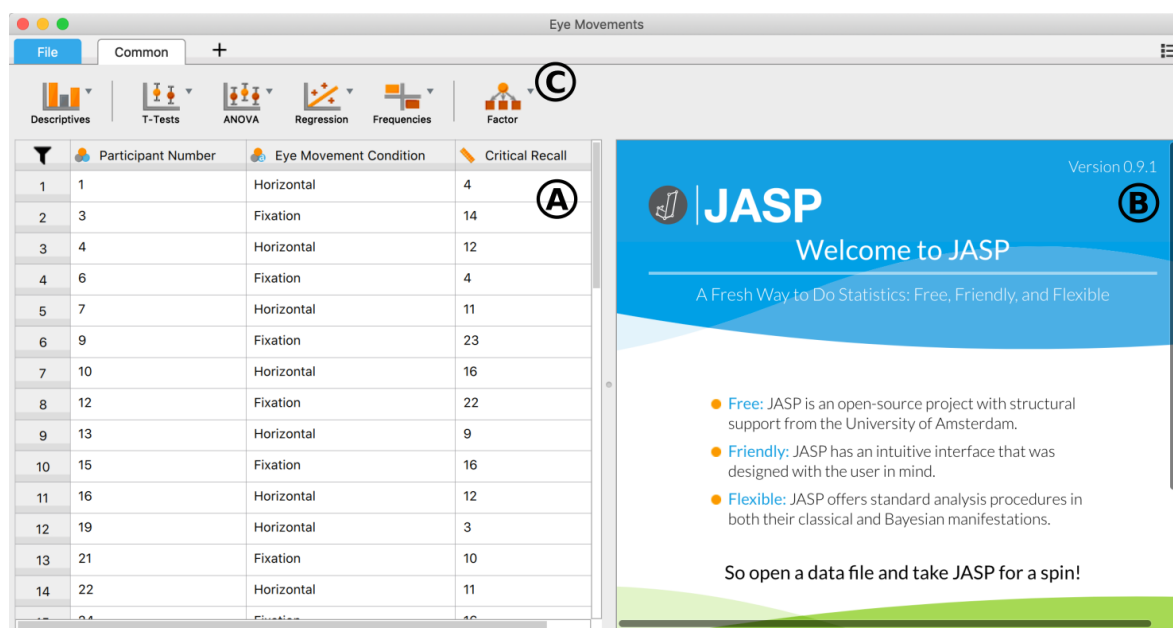


Figure 2: A screenshot of the **JASP** user interface. (A) Data display; (B) Output window; (C) Menus for analyses.

### 5.1. Data display and the analysis menu

As shown in Figure 2, **JASP** displays the data in a spreadsheet (A). Each column header contains an icon that represents the measurement level of the corresponding variable: nominal, ordinal, or continuous (Stevens 1946). When loading a data set, **JASP** employs a best guess to determine the type of the variables. The user can override the default variable types by clicking on the corresponding icon, and choosing an alternative from the menu. In the Matzke data set, the number of correctly recalled words is encoded in the “Critical Recall” column; the eye movement condition is encoded in the “Eye Movement Condition” column, with levels “Horizontal” and “Fixation”.

The right hand panel of Figure 2 shows the output panel that will be populated with results as soon as the user requests the desired analysis (B). The ribbon across the top of the user interface contains menus for the available statistical analyses (C).

### 5.2. Classical $t$ test

In order to perform a classical independent-samples  $t$  test, the user clicks the “T-Tests” menu, and then selects the “Independent Samples T-Test” option. The options displayed in the left hand panel of Figure 3 allow the user to specify the details of the analysis; the right hand panel shows the output. The standard  $t$  test table in the output panel is highlighted in white, signifying that it is the current analysis.

The options panel displays the list of available variables (i.e., columns from the data file; A). The user can assign these variables as the dependent variables or the grouping variable by dragging and dropping them in the appropriate box (B), or by using the assignment buttons. Here we assign “CriticalRecall” to the Dependent Variables box, and “EyeMovementCondi-

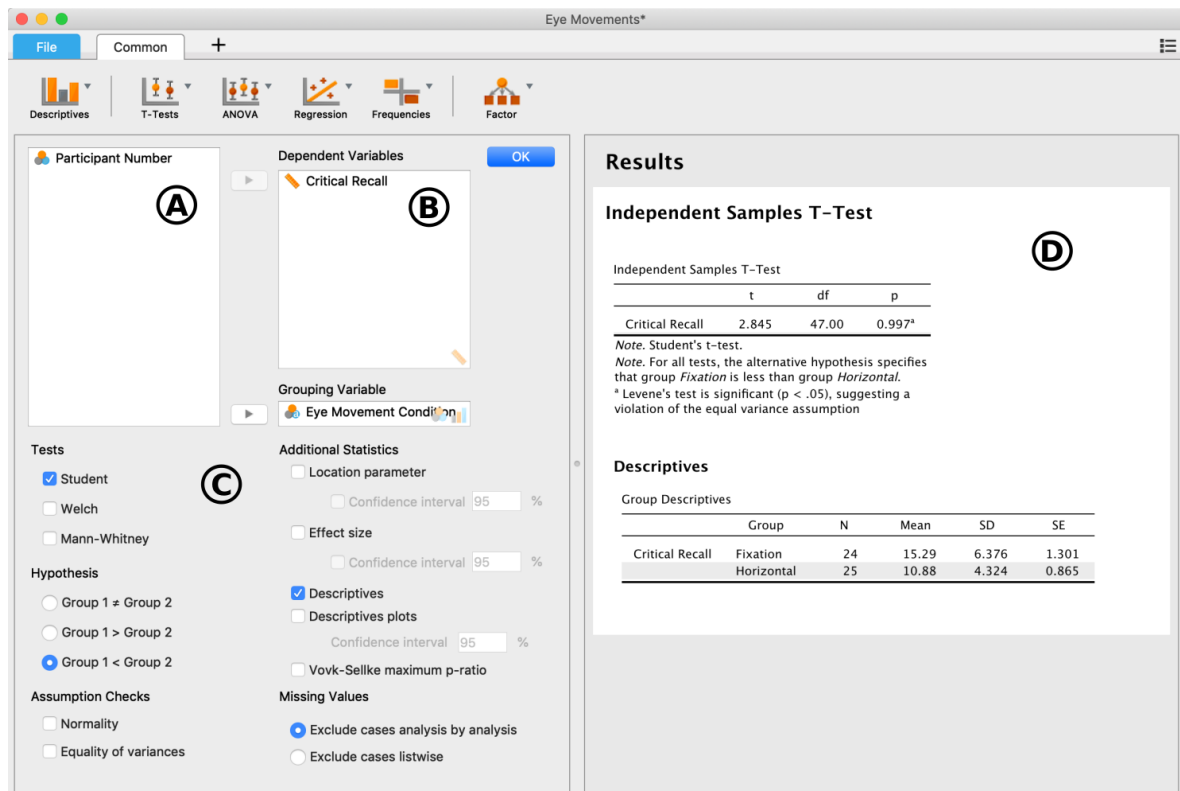


Figure 3: A classical  $t$  test in **JASP**. (A) List of available variables that can be assigned to the analysis; (B) The Dependent Variables box is assigned the variable “Critical Recall”; the Grouping Variable box is assigned the variable “Eye Movement Condition”; (C) Additional output options; (D) Output.

tion” to the Grouping variable box. The classical  $t$  test table is displayed automatically in the output panel (D).

**JASP** provides the user with a range of additional output options (C). For instance, clicking the “Descriptives” option produces the “Group Descriptives” table in the output panel. This table contains various descriptive statistics such as the group sample sizes. Importantly, the Group Descriptives table also shows the order of the groups: Results for the Fixation condition are displayed first, and the results for the Horizontal condition are displayed second. This order corresponds to Group 1 and Group 2 in the Hypothesis option. Here we clicked the Group 1 < Group 2 option in order to test the order-restricted alternative hypothesis that participants in the Fixation condition are expected to recall fewer words on average than do participants in the Horizontal condition. As shown in the output panel, this one-sided test results in a  $p$  value of 0.997.

Note that clicking additional options immediately adds the corresponding results to the output panel, providing direct feedback to the user. Similarly, unchecking options removes the corresponding results from the output panel immediately. **JASP**’s behavior contrasts to that of most other graphical statistical software, where results are produced in a separate window, and only after the user has fully specified the analysis and clicked “OK”. Also note that the output tables conform to the publishing standards of the APA, saving the user the tedious



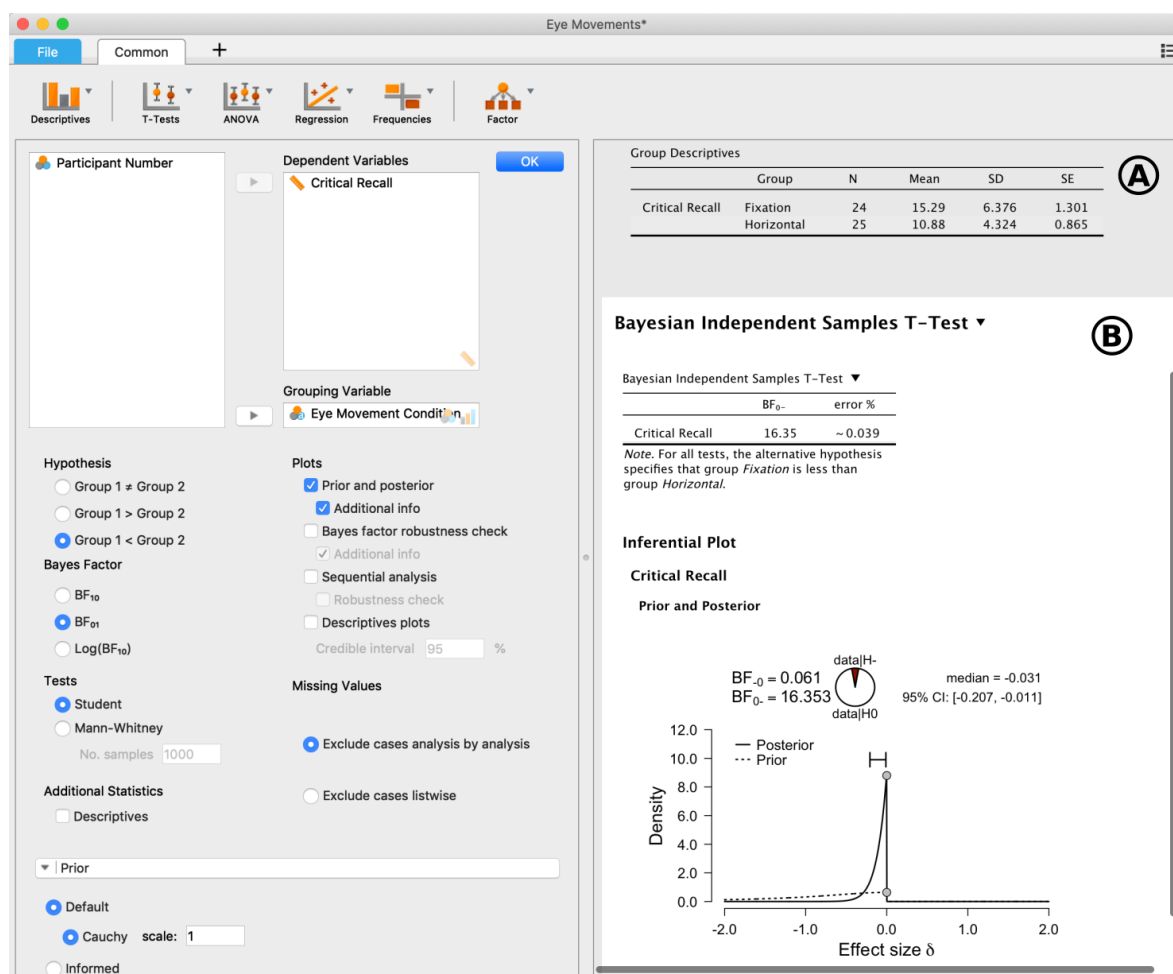


Figure 4: A Bayesian  $t$  test in **JASP**. (A) The results from the previous classical  $t$  test are shaded in gray; (B) The results from the current Bayesian  $t$  test are highlighted in white.

task of reformatting the output. Placing the mouse cursor over the table presents a menu with the option to copy, allowing the table to be pasted into a different program such as Microsoft Word or LibreOffice.

### 5.3. Bayesian $t$ test

The previous classical results could be obtained by any common statistical software. **JASP**, however, also provides users with the ability to conduct a Bayesian independent samples  $t$  test by clicking the “T-Tests” menu, and selecting the “Bayesian Independent Samples T-Test” option. **JASP** offers the Jeffreys-Zellner-Siow  $t$  test described by Rouder *et al.* (2009), which uses a scalable Cauchy distribution as a prior on the effect size, instantiating the assumption that effect sizes are likely to be small (Jeffreys 1961). **JASP** allows the specification of the analysis options, including the prior scale, as shown in the left hand panel of Figure 4; the right hand panel shows the output, where the Bayesian results (B) are appended to the earlier results of the classical  $t$  test (A). The current Bayesian results are highlighted in white; the results of the previous classical  $t$  test are now shaded in gray. Subsequent analyses accumulate,

one after another, in a similar fashion. For the present analysis, the standard output contains a Bayesian  $t$  test table that displays the Bayes factor, a Bayesian model selection measure that quantifies the relative plausibility of the data under the null hypothesis versus the alternative hypothesis (Berger 2006; Jeffreys 1961; Kass and Raftery 1995). If the option “Prior and posterior” is checked, then **JASP** will also report Bayesian parameter estimation results.

The functionality of the Dependent Variables and Grouping Variable boxes, as well as the Hypothesis, Additional Statistics, and Missing Values options resemble that of the classical analysis. Additionally, users can select whether they prefer to obtain Bayes factors that quantify evidence in favor of the alternative hypothesis ( $BF_{10}$ ) or Bayes factors that quantify evidence in favor of the null hypothesis ( $BF_{01}$ ); finally, users can choose to define one-sided alternatives (Morey and Rouder 2011; Morey and Wagenmakers 2014).

In the present example, we requested a one-sided Bayes factor in favor of the null hypothesis ( $BF_{01}$ ) and corresponding density plots of the prior and the posterior distributions. The output is shown in the right hand panel (B): the Bayes factor equals 16.35, indicating that the data are more than 16 times more likely under the null hypothesis than under the alternative hypothesis.

## 6. Implementation details

The **JASP** user interface is written in C++, analyses are implemented in the R programming language (R Core Team 2018), and the results panel is an instance of the **WebKit** (The WebKit Open Source Project 2015) browser with the resultant tables and plots rendered in HTML through JavaScript libraries built on top of **Backbone.js** (DocumentCloud 2015).

**JASP** runs as separate processes, with the **JASP** user interface running in one process, and the analyses running in separate background processes. This is depicted in Figure 5. When the user loads a data set such as a CSV file, the user interface (UI) process loads the data into interprocess shared memory, giving the background processes access to the data as well.

When the user creates an analysis, a new in-memory representation of the analysis is created, and the analysis is scheduled to be run. If there are no background processes available, for example, in the case that several analyses are already running, the analysis is added to a queue. If a background process is available, or becomes available, the waiting analysis is sent to the background process over shared memory, and a semaphore is set, wakening the process, and causing the analysis to be run. When the analysis completes, the results are also placed in shared memory, the UI process collects them from there, and the results panel is populated.

Analyses are performed in two stages; first the analysis is initialized, and subsequently, the analysis is run. These two stages are controlled by a scheduler, and may not necessarily occur in the same background process. The intention of the initialization step is to generate empty results, for example an empty table, immediately providing immediate feedback to the user. As such, initialization is designed to occur very quickly, and one of the background processes is reserved solely for initializations so that longer running analyses do not delay this feedback.

### 6.1. Analysis implementation

At present, each analysis in **JASP** is represented by an R function. When an analysis is initialized or run, the appropriate R function is called with arguments describing the user

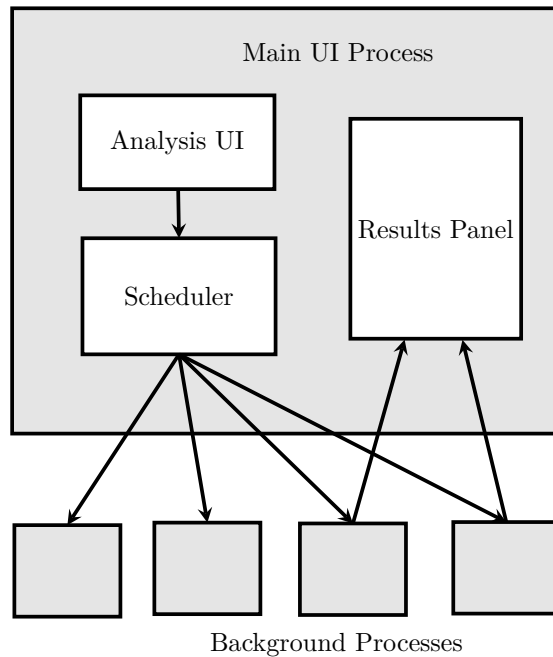


Figure 5: Conceptual overview of the internal **JASP** architecture. Analyses are scheduled, sent to background processes, and the results are sent to the results panel.

interface options, and whether the analysis is being initialized or run. The function is then able to access the data set residing in shared memory through several native functions provided by the **JASP** R environment through the **Rcpp** (Eddelbuettel and François 2011) and the **RInside** (Eddelbuettel and François 2015) packages. These native functions allow the analysis to request the data as a data frame. Having obtained the data in this way, the analysis marshals the **JASP** UI options to the argument forms that underlying R packages expect (A list of R packages used in **JASP** is available in Appendix A). Functions from these packages are called, results objects which are then marshaled into a nested structure of lists representing the tables and images. This nested list is then returned from the function, where it is converted into JSON and passed back to the **JASP** UI process. The **JASP** UI process passes the JSON representation of the results to the results panel, which uses JavaScript components written in Backbone.js to render these results into the results document as HTML elements. This produces tables and images in the results panel. Note that even though **JASP** borrows functionality from R, **JASP** installs as an independent program. A version of R is bundled with the **JASP** installation.

## 6.2. Callbacks

As described, the **JASP** R function returns the results as its return value. However, it is also possible for the analysis to return results while it is running. Similarly, it is possible for the analysis to respond to changes that the user makes to its options while it is running. This allows for two important use-cases. While **JASP** analyses are running, they can provide partial results as the analysis progresses, and they can receive and respond to changes to the analysis options. This is achieved through a callback mechanism. The analysis periodically

calls the callback, passing the intermediate results in as an argument, and the callback returns a value indicating whether the analysis options were changed by the user, and what those new options are. In this way, the user interface and in-progress analyses can communicate with one another, and both can respond accordingly.

### 6.3. State system

The **JASP** analysis environment also provides a state-system, allowing an analysis to save its state, and retrieve that state next time the analysis is run. This is useful in several situations, such as in the case that a user may have run a  $t$  test, and received a  $t$  statistic and a  $p$  value. They may then go on to subsequently check a check box requesting an effect size estimate. Without the ability to save and retrieve state, this analysis would need to run again from beginning to end, with the new option selected. This is inefficient, because the  $t$  statistic and  $p$  value had already been calculated the last time, and each change to the analysis results in the same values being calculated over and over again.

In contrast, the state system allows the analysis, as one of its very last steps, to save state; for example, it could save its  $t$  statistic and  $p$  value. Next time the analysis is run, the analysis can retrieve the state, examine what options have changed, and if appropriate simply use the  $t$  statistic and  $p$  value from last time. This allows the analysis to save time by only computing what needs to be computed. For longer running analyses, this feature can substantially reduce waiting times.

## 7. Concluding comments

**JASP** is a free and open-source statistical package for basic statistics. It is intuitive, user-friendly, and provides an innovative user experience. The adoption of **JASP** in preference to other proprietary statistical packages will enable peer review of the statistical algorithms, enable scientists to build on existing work, and reduce costs.

Additionally, **JASP** is based on a unique approach to graphical statistical software, providing immediate feedback, attractive and minimalist output, and perfect transparency. It is hoped that **JASP** will, beyond its immediate contribution as a software package, inspire a new generation of statistical software based around these principles. Finally, **JASP** is statistically inclusive. For the basic statistical scenarios at hand, **JASP** implements both the classical and the Bayesian approach. The Bayesian routines in **JASP** can be used for parameter estimation and for default Bayes factor hypothesis testing. Our aim is for **JASP** to unlock the recent Bayesian developments and make them accessible to a broader audience of researchers and students.

**JASP** is available for Windows, Mac OS X and Linux from the project website <https://jasp-stats.org/>, and the source code is available from the GitHub repository <https://github.com/jasp-stats/jasp-desktop>.

## References

- Berger JO (2006). “Bayes Factors.” In S Kotz, N Balakrishnan, C Read, B Vidakovic, NL Johnson (eds.), *Encyclopedia of Statistical Sciences*, volume 1, 2nd edition, pp. 378–386. John Wiley & Sons, Hoboken.

- DocumentCloud (2015). “**Backbone.js**.” URL <http://backbonejs.org/>.
- Eddelbuettel D, François R (2011). “**Rcpp**: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.
- Eddelbuettel D, François R (2015). ***RInside** 0.2.13*. URL <http://dirk.eddelbuettel.com/code/rinside.html>.
- Fellows I (2012). “**Deducer**: A Data Analysis GUI for R.” *Journal of Statistical Software*, **49**(8), 1–15. doi:10.18637/jss.v049.i08.
- Fox J (2005). “Getting Started with the **R Commander**: A Basic Statistics Graphical User Interface to R.” *Journal of Statistical Software*, **14**(9), 1–42. doi:10.18637/jss.v014.i09.
- Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks.
- Fox J, Weisberg S, Price B (2018). ***car**: Companion to Applied Regression*. R package version 3.0-2, URL <https://CRAN.R-project.org/package=car>.
- Friendly M (2017). ***vcdExtra**: vcd Extensions and Additions*. R package version 0.7-1, URL <https://CRAN.R-project.org/package=vcdExtra>.
- Gunel E, Dickey J (1974). “Bayes Factors for Independence in Contingency Tables.” *Biometrika*, **61**(3), 545–557. doi:10.2307/2334738.
- Hankin RKS (2016). ***hypergeo**: The Gauss Hypergeometric Function*. R package version 1.2-13, URL <https://CRAN.R-project.org/package=hypergeo>.
- IBM Corporation (2013). *IBM SPSS Statistics 22*. IBM Corporation, Armonk. URL <https://www.ibm.com/software/analytics/spss/>.
- Jamil T, Ly A, Marsman M, Love J, Morey RD, Wagenmakers EJ (2017). “Default “Gunel and Dickey” Bayes Factors for Contingency Tables.” *Behavior Research Methods*, **49**(2), 638–652. doi:10.3758/s13428-016-0739-8.
- Jeffreys H (1961). *Theory of Probability*. 3rd edition. Oxford University Press, New York.
- Kass RE, Raftery AE (1995). “Bayes Factors.” *Journal of the American Statistical Association*, **90**(430), 773–795. doi:10.2307/2291091.
- Lenth RV (2016). “Least-Squares Means: The R Package **lsmeans**.” *Journal of Statistical Software*, **69**(1), 1–33. doi:10.18637/jss.v069.i01.
- Liang F, Paulo R, Molina G, Clyde MA, Berger JO (2008). “Mixtures of g-Priors for Bayesian Variable Selection.” *Journal of the American Statistical Association*, **103**(481), 410–423. doi:10.1198/016214507000001337.
- Ly A, Verhagen AJ, Wagenmakers EJ (2016). “Harold Jeffreys’s Default Bayes Factor Hypothesis Tests: Explanation, Extension, and Application in Psychology.” *Journal of Mathematical Psychology*, **72**, 19–32. doi:10.1016/j.jmp.2015.06.004.

- Matzke D, Nieuwenhuis S, Van Rijn H, Slagter HA, Van der Molen MW, Wagenmakers EJ (2015). “The Effect of Horizontal Eye Movements on Free Recall: A Preregistered Adversarial Collaboration.” *Journal of Experimental Psychology: General*, **144**(1), e1–e15. doi:10.1037/xge0000038.
- Meyer D, Zeileis A, Hornik K (2006). “The Strucplot Framework: Visualizing Multi-way Contingency Tables with `vcd`.” *Journal of Statistical Software*, **17**(3), 1–48. doi:10.18637/jss.v017.i03.
- Morey RD, Rouder JN (2011). “Bayes Factor Approaches for Testing Interval Null Hypotheses.” *Psychological Methods*, **16**, 406–419. doi:10.1037/a0024377.
- Morey RD, Rouder JN (2018). **BayesFactor**: *Computation of Bayes Factors for Common Designs*. R package version 0.9.12-4.2, URL <https://CRAN.R-project.org/package=BayesFactor>.
- Morey RD, Wagenmakers EJ (2014). “Simple Relation Between Bayesian Order-Restricted and Point-Null Hypothesis Tests.” *Statistics and Probability Letters*, **92**, 121–124. doi:10.1016/j.spl.2014.05.010.
- Open Science Collaboration (2015). “Estimating the Reproducibility of Psychological Science.” *Science*, **349**(6251). doi:10.1126/science.aac4716.
- Paton-Simpson & Associates (2015). “**SOFA Statistics**.” URL <http://www.sofastatistics.com/>.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Rödiger S, Friedrichsmeier T, Kapat P, Michalke M (2012). “**RKward**: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R.” *Journal of Statistical Software*, **49**(9), 1–34. doi:10.18637/jss.v049.i09.
- Rouder JN, Morey RD (2012). “Default Bayes Factors for Model Selection in Regression.” *Multivariate Behavioral Research*, **47**(6), 877–903. doi:10.1080/00273171.2012.734737.
- Rouder JN, Morey RD, Speckman PL, Province JM (2012). “Default Bayes Factors for ANOVA Designs.” *Journal of Mathematical Psychology*, **56**(5), 356–374. doi:10.1016/j.jmp.2012.08.001.
- Rouder JN, Speckman PL, Sun D, Morey RD, Iverson G (2009). “Bayesian *t* Tests for Accepting and Rejecting the Null Hypothesis.” *Psychonomic Bulletin & Review*, **16**(2), 225–237. doi:10.3758/pbr.16.2.225.
- Singmann H, Bolker B, Westfall J, Aust F (2018). **afex**: *Analysis of Factorial Experiments*. R package version 0.22-1, URL <https://CRAN.R-project.org/package=afex>.
- Stevens SS (1946). “On the Theory of Scales of Measurement.” *Science*, **103**(2684), 677–680. doi:10.1126/science.103.2684.677.
- The GNU Foundation (2015). “**PSPP**.” URL <https://www.gnu.org/software/pspp/>.

- The WebKit Open Source Project (2015). “**WebKit**.” URL <http://www.webkit.org/>.
- Valero-Mora PM, Ledesma R (2012). “Graphical User Interfaces for R.” *Journal of Statistical Software*, **49**(1), 1–8. doi:[10.18637/jss.v049.i01](https://doi.org/10.18637/jss.v049.i01).
- Wetzels R, Grasman RPPP, Wagenmakers EJ (2012). “A Default Bayesian Hypothesis Test for ANOVA Designs.” *The American Statistician*, **66**(2), 104–111. doi:[10.1080/00031305.2012.695956](https://doi.org/10.1080/00031305.2012.695956).
- Wetzels R, Raaijmakers JGW, Jakab E, Wagenmakers EJ (2009). “How to Quantify Support For and Against the Null Hypothesis: A Flexible **WinBUGS** Implementation of a Default Bayesian  $t$  Test.” *Psychonomic Bulletin & Review*, **16**(4), 752–760. doi:[10.3758/pbr.16.4.752](https://doi.org/10.3758/pbr.16.4.752).
- Wickham H (2011). “The Split-Apply-Combine Strategy for Data Analysis.” *Journal of Statistical Software*, **40**(1), 1–29. doi:[10.18637/jss.v040.i01](https://doi.org/10.18637/jss.v040.i01).
- Wickham H (2016). **ggplot2: Elegant Graphics for Data Analysis**. Springer-Verlag, New York. ISBN 978-3-319-24277-4.

## A. Available analyses and R packages used

Note that the R packages used by **JASP** will change as development progresses. An up-to-date list of the R packages used and their versions is maintained on the project website: <https://jasp-stats.org/r-package-list/>.

Analysis	R packages
Independent samples $t$ test	<b>stats</b> (R Core Team 2018)
Paired samples $t$ test	
One sample $t$ test	
ANOVA	<b>base</b> (R Core Team 2018)
Repeated measures ANOVA	<b>stats</b> (R Core Team 2018)
ANCOVA	<b>lsmeans</b> (Lenth 2016)
Linear regression	<b>car</b> (Fox <i>et al.</i> 2018; Fox and Weisberg 2011) <b>afex</b> (Singmann <i>et al.</i> 2018) <b>grid</b> (R Core Team 2018) <b>plyr</b> (Wickham 2011) <b>ggplot2</b> (Wickham 2016)
Correlation matrix	<b>stats</b> (R Core Team 2018)
Contingency tables	<b>stats</b> (R Core Team 2018) <b>vcd</b> (Meyer <i>et al.</i> 2006) <b>vcdExtra</b> (Friendly 2017)
Bayesian independent samples $t$ test	<b>BayesFactor</b> (Morey and Rouder 2018)
Bayesian paired samples $t$ test	
Bayesian one sample $t$ test	
Bayesian ANOVA	<b>BayesFactor</b> (Morey and Rouder 2018)
Bayesian repeated measures ANOVA	
Bayesian ANCOVA	
Bayesian linear regression	
Bayesian correlation matrix	<b>hypergeo</b> (Hankin 2016)
Bayesian correlation pairs	
Bayesian contingency tables	<b>BayesFactor</b> (Morey and Rouder 2018)

Table 1: Available analyses in **JASP**, and the R packages used.



**Affiliation:**

Jonathon Love  
University of Newcastle  
School of Psychology  
University Drive, Callaghan  
NSW 2308, Australia  
E-mail: [jon@thon.cc](mailto:jon@thon.cc)  
URL: <https://jona.thon.love/>