

DISCLAIMER:

These materials are based on the following tutorial paper, currently in press:

Garofalo S., Finotti G., Orsoni M., Giovagnoli S., Benassi M. (2024) Testing Bayesian Informative Hypotheses in 5 steps with R and JASP. Advances in Methods and Practices in Psychological Science. doi: 10.1177/25152459241260259

Please refer to the full paper for more information.

Example 1: 2x2 Anova

Let's assume we want to test the efficacy of a new anxiety treatment on different levels of symptomatology. To this aim, we recruit 200 volunteers suffering from anxiety-related disorders equally divided into two groups, one presenting high levels of symptoms and one presenting low levels of symptoms. We then randomly assign half of each group to the experimental group (receiving a real drug treatment) and the other half to the control group (receiving a placebo treatment). We measure the anxiety level before and after the treatment and we compute an index of treatment efficacy by subtracting the score recorded before the treatment from the score recorded after the treatment (pre-post) so that a positive score would indicate an improvement (i.e., a reduction) in anxiety, a negative score would indicate an increase in anxiety, and 0 no difference between before and after the treatment.

We thus have a 2x2 factorial design with treatment (Drug/Placebo) and symptoms (High/Low) as between-subjects independent variables and the treatment efficacy index as the dependent variable.

The Anova dataset

The dataset is composed of five columns: "ID" (containing participants' identification number), "treatment" (Drug/Placebo), "symptoms" (High/Low)", and "score" (containing a continuous variable reporting the treatment efficacy score). Of note, since the bain package requires all the levels of a factorial design to be stored in one single variable, an additional column named "groups" containing all four groups (specifically, Drug High, Drug Low, Placebo High, and Placebo Low) is needed.

The hypotheses

Let's assume we want to compare and contrast a set of three hypotheses. The first hypothesis poses that the two groups receiving a real treatment (Drug) show a positive increase (> 0) in the treatment efficacy index and (&) that the strength of such treatment, intended as the difference between the two groups (Drug/Placebo), is similar regardless of symptomatology level (High/Low). This hypothesis can formally be represented as:

$$H_1: (\mu_{\text{Drug.High}}, \mu_{\text{Drug.Low}}) > 0 \ \& \ (\mu_{\text{Drug.High}} - \mu_{\text{Placebo.High}}) = (\mu_{\text{Drug.Low}} - \mu_{\text{Placebo.Low}})$$

The second hypothesis poses again that the two groups receiving a real treatment (Drug) show a positive increase (> 0) in the treatment efficacy index and (&) that the strength of such treatment, intended as the difference between the two (Drug/Placebo), is higher for the group with High symptoms, as compared to the group with Low symptoms:

$$H_2: (\mu_{\text{Drug.High}}, \mu_{\text{Drug.Low}}) > 0 \ \& \ (\mu_{\text{Drug.High}} - \mu_{\text{Placebo.High}}) > (\mu_{\text{Drug.Low}} - \mu_{\text{Placebo.Low}})$$

The third hypothesis poses that there are comparable scores regardless of treatment and symptomatology level:

$$H_3: \mu_{\text{Drug.High}} = \mu_{\text{Placebo.High}} = \mu_{\text{Drug.Low}} = \mu_{\text{Placebo.Low}}$$

For more information on how to define informative hypotheses, see the paragraph “Characteristics of the informative hypotheses”.

Step-by-step tutorial in JASP

Preliminary steps

To run the Bayesian informative hypothesis testing (bain) module in JASP, you first need to add the “bain” module to the software. To do that: open JASP, click on the “+” icon at the top right (do not confuse it with the “+” sign to “add a computed column”), and select the “BaIn” module. This will now add a “BaIn” icon to the top menu (Figure 2). After loading the dataset (Step 1a), you can click on the icon to see the list of all the bain analyses currently available (Figure 3).



Figure 2. Adding the bain module to JASP

It is important to know that many analyses require generating a series of random numbers for computational purposes. The bain package, in particular, uses sampling to compute Bayes factors and posterior model probabilities. To get reproducible results it is possible to set a specific seed number from which the series of pseudorandom numbers is generated. Knowing the seed and the generator it is always possible to reproduce the same output. Otherwise, you can change it to any number you like. In the “Additional Options” section you will see that, by default, the seed is set to “123”. A good practice to ensure the stability of your results is to repeat the analysis with different seeds and check whether the results are coherent. For this reason, we will use different seeds in the JASP and R examples and ensure that, although slightly different, the same trend should emerge from both analyses.

Step 1a: Load the dataset

The first step is to load a file containing the Anova dataset described, available on the OSF page <https://osf.io/dez9b/>. To do that, go to the main menu icon on the top left, select “Open → Computer”, and browse to the folder on your computer in which you downloaded the file, then select

and load the “dataset_anova.txt” file. You will now visualize a spreadsheet containing the entire dataset (Figure 3).

The screenshot shows a statistical software interface. At the top is a menu bar with icons and labels for various analysis types: Descriptives, T-Tests, ANOVA, Mixed Models, Regression, Frequencies, Factor, Bain, Machine Learning, and Meta-Analysis. Below the menu is a data table with 19 rows and 6 columns. The columns are labeled ID, treatment, symptoms, group, and score. The data shows 19 observations, all with 'Drug' treatment and 'High' symptoms, grouped into 'Drug.High'. The 'score' column contains numerical values ranging from approximately 4.18 to 8.57. To the right of the data table is a panel titled 'T-Tests' which is currently selected. This panel lists three options: Welch's T-Test, Paired Samples T-Test, and One Sample T-Test. Below this panel is another panel titled 'ANOVA' which lists 'ANOVA' and 'ANCOVA'. At the bottom of the right panel is a 'Regression' section listing 'Linear Regression' and 'Structural Equation Modeling'.

	ID	treatment	symptoms	group	score
1	1	Drug	High	Drug.High	6.577389762
2	2	Drug	High	Drug.High	5.031160555
3	3	Drug	High	Drug.High	8.192775974
4	4	Drug	High	Drug.High	7.405105742
5	5	Drug	High	Drug.High	6.584040367
6	6	Drug	High	Drug.High	6.150964392
7	7	Drug	High	Drug.High	4.182012613
8	8	Drug	High	Drug.High	5.099813284
9	9	Drug	High	Drug.High	5.548375473
10	10	Drug	High	Drug.High	5.318485958
11	11	Drug	High	Drug.High	8.988195568
12	12	Drug	High	Drug.High	7.200966086
13	13	Drug	High	Drug.High	8.408124169
14	14	Drug	High	Drug.High	7.258807152
15	15	Drug	High	Drug.High	8.43647567
16	16	Drug	High	Drug.High	4.955970632
17	17	Drug	High	Drug.High	7.102502699
18	18	Drug	High	Drug.High	7.150986471
19	19	Drug	High	Drug.High	8.352017131

Figure 3. Loading the dataset and selecting the analysis

Step 2a: Fit the model

By selecting “Bain → Anova” in the top menu (Figure 3), you will be presented with a new graphical user interface that can be used to select the variables of interest (Figure 4). Set the variable “score” as the dependent variable and the variable “group” as the fixed factor.

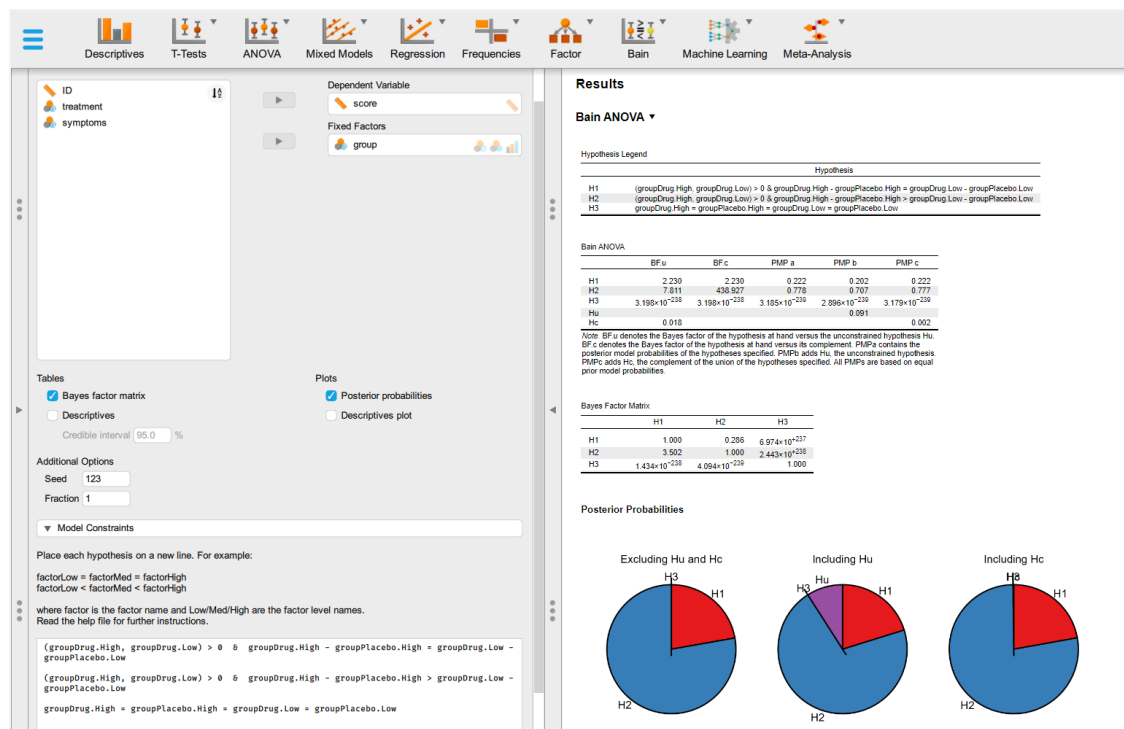


Figure 4. Selecting the variables, defining the informative hypothesis, and visualizing the results

Step 3a: Define and test informative hypotheses with bain

We can now set the hypotheses defined above and compare them. To do that, you can type or copy and paste the text reported below in the “Model Constraints” box that you can find at the bottom of the left panel (Figure 4):

```
(groupDrug.High, groupDrug.Low) > 0 & groupDrug.High - groupPlacebo.High = groupDrug.Low - groupPlacebo.Low
(groupDrug.High, groupDrug.Low) > 0 & groupDrug.High - groupPlacebo.High > groupDrug.Low - groupPlacebo.Low
groupDrug.High = groupPlacebo.High = groupDrug.Low = groupPlacebo.Low
```

Then press Ctrl+Enter (if you use a Windows PC) or Cmd+Return (if you use a Mac). JASP will start testing the informative hypothesis specified and the results will be presented on the right panel, where a table called “Bain Anova” will appear.

Step-by-step tutorial in R

There are two ways to proceed with this example: (1) open a new R script and use the following lines of code to play along with the tutorial; or (2) open and use the file “BaIn_AnovainR.R” available on the OSF page (<https://osf.io/dez9b/>), which contains all the steps and code reported below.

Preliminary steps

Clear workspace

Before starting any new analysis, it is generally considered good practice to clear all the variables, datasets, functions, etc. possibly loaded in the R environment. To do that, you can use the `rm()` function as follows:

```
rm(list=setdiff(ls(), "..."))
```

Install and/or load the necessary packages

Once R is installed, it comes with several built-in functions, like `sum()` which returns the sum of the values inserted between the brackets, or `sqrt()` which returns the square root of all the values present in its arguments. To know what a function is used for, you can type “?” before the function name (i.e., `?sum`) in the R console to open the help. Sets of functions that work together are usually grouped and contained in so-called packages (or libraries). For instance, `sum()` and `sqrt()` are part of the “base” package, which is automatically loaded when you use R. Packages for more specific analysis or plots have to be directly installed and loaded. For this tutorial, you need to install and load the “bain” package (Hojtink, Mulder, et al., 2019), which contains all the functions needed to run Bayesian informative hypotheses testing. To do so, you can use the `install.packages()` function by running in R or RStudio the following line:

```
install.packages("bain")
```

This line of code is needed only if you have not already installed this package. Once a package is installed on your computer, you can use the `library()` function to load it and access its functions. This loading needs to be done every time you start a new R session.

```
library(bain)
```

These two steps are required each time you need to use functions that are not part of “base” R but need to be loaded from an external package. In this tutorial, a few more packages will be used to generate plots or manipulate the data. These packages are not strictly required but can be very useful for several purposes. Each time we suggest one of these packages we will explain their purpose and how to use them. However, the same results could be obtained with different packages.

Step 1a: Load the dataset

There are several options to load the file containing your data. The easiest one is to use the RStudio graphical user interface by clicking on “File → Import dataset” and choosing your file type (e.g., text, excel, SPSS, etc., see Figure 7). For this example, the dataset is saved in a .txt file named “dataset_anova.txt”. To open it, go to “File → Import dataset → From Text (base)” and browse to the folder on your computer that contains the “dataset_anova.txt” file (downloaded from OSF page <https://osf.io/dez9b/>); then click on “Import” to load the file. If you accept the default values, your dataset will now appear in the R Environment with the same name as the original filename.

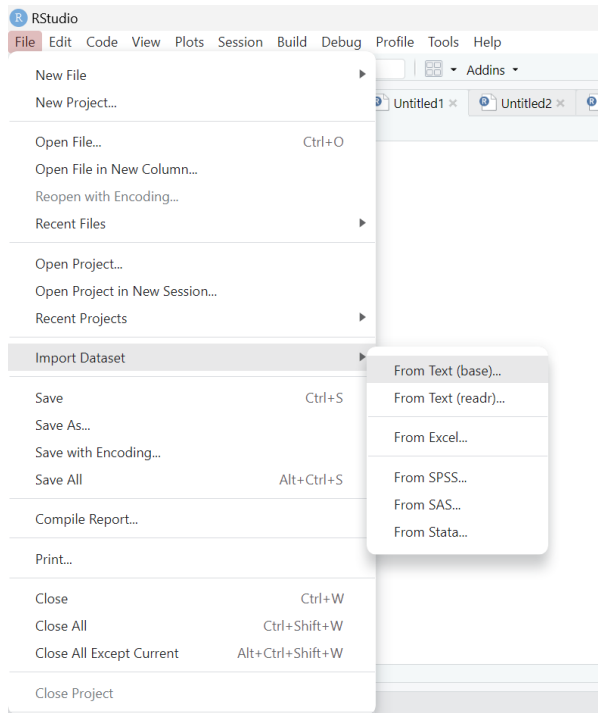


Figure 7. Loading the dataset from the graphical user interface of RStudio

If you prefer to use R code for loading a file, there are dedicated packages containing functions that can be used based on the file extension (for example, a .txt file can be loaded with the `read.table()` function). An overview of such functions would go beyond the scope of this tutorial but in the R script available online you will find a few commented lines that can be used for this purpose.

Once the file has been loaded, an object named "dataset_anova" should appear in your R environment. By clicking on this object, the entire dataset will appear (Figure 8) containing the variables described above in "The Anova dataset" paragraph. The `str()` function can be used to familiarize with your dataset. This function shows the structure of your dataset and has the additional benefit of showing the data type for each variable.

```
str(dataset_anova)
```

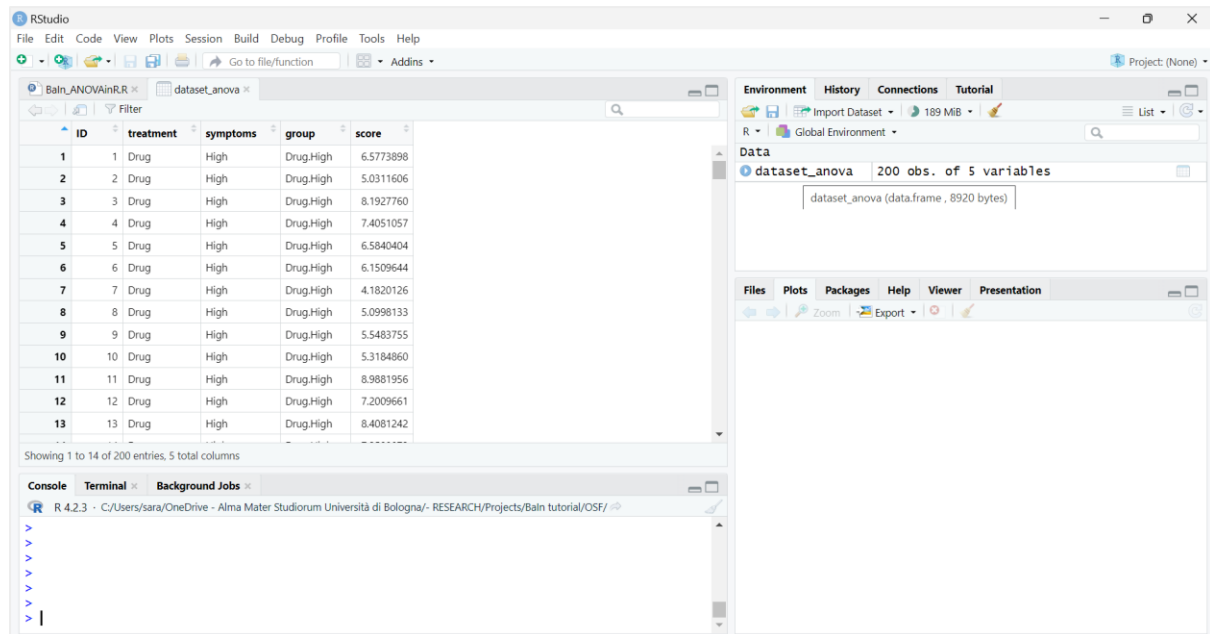


Figure 8. Dataset loaded in RStudio

Step 2a: Fit the model

The following code can be used to fit a linear model, with the function `lm()`, which returns estimated means for each level contained in the column “group”. The outcome of the fitted model will be stored in an object called “fit”. Note that the -1 is used to estimate the means for each level of the variables. The subsequent line uses the `coef()` function to store the means estimated by the model for each group and session in an object called “estimated”.

```
fit = lm(score ~ group - 1, data = dataset_anova)
```

Step 3a: Define and test informative hypotheses with bain

We can now set the hypotheses defined above and compare them. Before doing that, you can use the `coef()` function to take a look at the estimated means and their names, as these are the variable names to use to define the informative hypotheses:

```
coef(fit)
```

It is also important to know that many analyses require generating a series of random numbers for computational purposes. The “bain” package, in particular, uses sampling to compute Bayes factors and posterior model probabilities. To get reproducible results it is possible to use the `set.seed()` function, which sets a specific seed number from which the series of pseudorandom numbers is generated. Knowing the seed and the generator it is always possible to reproduce the same output. It can be set to any number you like. A good practice to ensure the stability of your results is to repeat the analysis with different seeds and check whether the results are coherent. For this reason, we may use different seeds in the JASP and R examples and ensure that, although slightly different, the same trend should emerge from both analyses. The following code can be used to set the seed to 123:

```
set.seed(123)
```

The following code can be used to test the informative hypotheses described above. The output is stored in an object called “results”.

```
results <- bain(x = fit,  
  hypothesis =  
  "(groupDrug.High, groupDrug.Low) > 0 & groupDrug.High - groupPlacebo.High = groupDrug.Low - groupPlacebo.Low;  
  groupDrug.High, groupDrug.Low) > 0 & groupDrug.High - groupPlacebo.High > groupDrug.Low - groupPlacebo.Low;  
  groupPlacebo.High = groupDrug.High = groupPlacebo.Low = groupDrug.Low")
```

In this code, the first argument “x” is a vector containing the estimated means for each group and condition, while the second argument “hypothesis” contains the hypotheses.

There are a few options available to display the results. The function print() returns in the console the main results, or bain Anova table:

```
print(results)
```

A pie chart displaying the three PMPs reported in Table 1 can be created using the pie() function or similar ones.

To further support model selection, PMPs can also be compared via a Bayes factor matrix (Box 3) in which all hypotheses are compared with each other. The following code can be used to extract this information from the “results” object created in the previous steps:

```
results$BFmatrix
```

The following line of code can be used to print a table with descriptive statistics:

```
summary(results, ci = 0.95)
```

NOW ANSWER THESE QUESTIONS:

- 1) Which of a set of hypotheses is the best?
- 2) How much more likely a given hypothesis is relative to other possible explanations?
- 3) How much more likely a given hypothesis is relative to another specific hypothesis?

Example 2: Multiple linear regression

Let's assume we have already established that the previously tested anxiety treatment works (Drug > Placebo) and now wish to investigate if factors other than symptom severity can impact its effectiveness. To this aim, we want to evaluate on a continuous scale the impact of symptom severity, along with drug dosage and age (independent variables). To this aim, we recruited 100 volunteers currently undergoing such anxiety treatment and obtained the same treatment efficacy index used before (dependent variable). Since all our variables are continuous, multiple linear regression is a viable approach to test the influence that each independent variable, as well as their combination, can exert on the dependent variable.

The regression dataset

The dataset is composed of five columns: "ID" (containing participants' identification number), "treatment.effect" (containing the treatment efficacy score), "age" (containing participants' age), "dosage" (containing participants' drug dosage), and "symptoms" (containing participants' symptoms). All variables are on a continuous scale.

The hypotheses

The first hypothesis poses that dosage has a higher impact than symptoms, which has a higher impact than age on treatment effect:

$$H_1: \text{dosage} > \text{symptoms} > \text{age}$$

The second hypothesis poses that symptoms have a higher impact than dosage, which has a higher impact than age on treatment effect:

$$H_2: \text{symptoms} > \text{dosage} > \text{age}$$

The third hypothesis poses that dosage and symptoms have a comparably higher impact than age on treatment effect:

$$H_3: (\text{dosage}, \text{symptoms}) > \text{age}$$

For more on how to define informative hypotheses, see the paragraph "Characteristics of the informative hypotheses".

Step-by-step tutorial in JASP

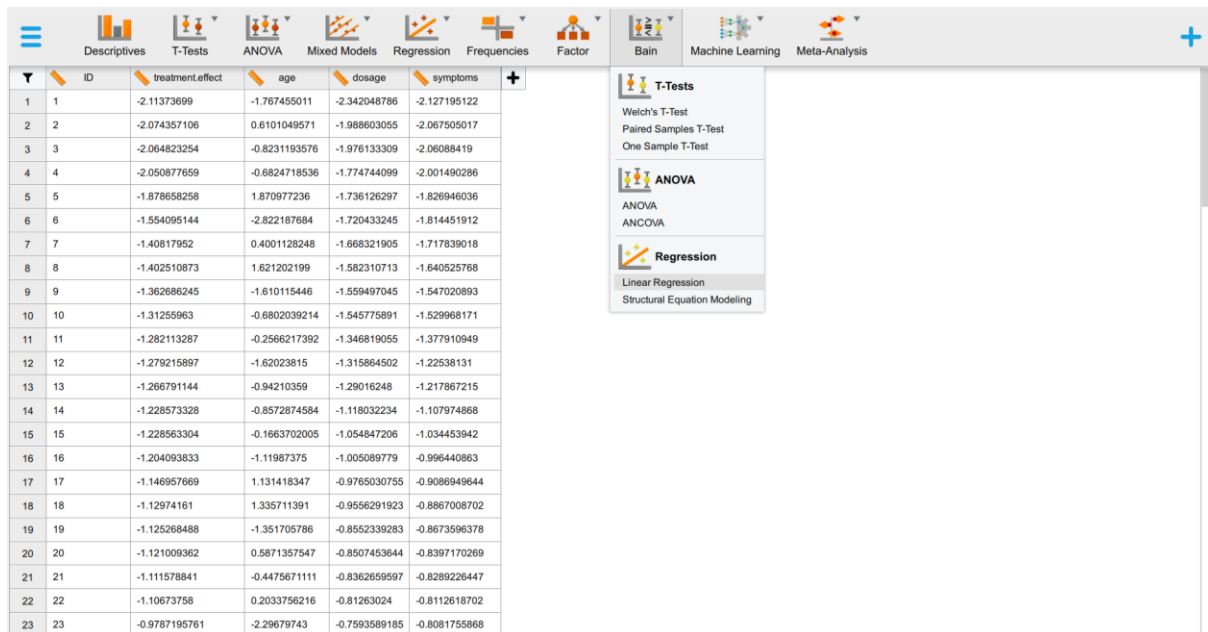
The analysis presented in this section is performed using the bain module on JASP (JASP Team, 2021). The file "bain_REGRESSIONinJASP.jasp" available on the OSF page (<https://osf.io/dez9b/>) contains all the steps reported below and can be directly opened in JASP.

Preliminary steps

Follow the "Preliminary steps" section of the JASP tutorial for Example A.

Step 1b: Load the dataset

Follow Step 1a of the JASP tutorial to load the file “dataset_regression.txt”. You will now visualize a spreadsheet containing the entire dataset (Figure 11) described above in “The regression dataset” paragraph.



The screenshot shows the JASP software interface. At the top, there is a menu bar with icons for Descriptives, T-Tests, ANOVA, Mixed Models, Regression, Frequencies, Factor, Bain, Machine Learning, and Meta-Analysis. Below the menu bar, a dataset is loaded, displaying a table with 23 rows and 6 columns: ID, treatment.effect, age, dosage, symptoms, and an empty column. The 'Bain' menu is open, showing options for T-Tests (Welch's T-Test, Paired Samples T-Test, One Sample T-Test), ANOVA (ANOVA, ANCOVA), and Regression (Linear Regression, Structural Equation Modeling).

	ID	treatment.effect	age	dosage	symptoms
1	1	-2.11373699	-1.767455011	-2.342048786	-2.127195122
2	2	-2.074357106	0.6101049571	-1.988603055	-2.067505017
3	3	-2.064823254	-0.8231193576	-1.976133309	-2.06088419
4	4	-2.050877659	-0.6824718536	-1.774744099	-2.001490286
5	5	-1.878658258	1.870977236	-1.736126297	-1.826946036
6	6	-1.554095144	-2.822187684	-1.720433245	-1.814451912
7	7	-1.40817952	0.4001128248	-1.668321905	-1.717839018
8	8	-1.402510873	1.621202199	-1.582310713	-1.640525768
9	9	-1.362686245	-1.610115446	-1.559497045	-1.547020893
10	10	-1.31255963	-0.6802039214	-1.545775891	-1.529968171
11	11	-1.282113287	-0.2566217392	-1.346819055	-1.377910949
12	12	-1.279215897	-1.62023815	-1.315864502	-1.22538131
13	13	-1.266791144	-0.94210359	-1.29016248	-1.217867215
14	14	-1.228573328	-0.8572874584	-1.118032234	-1.107974868
15	15	-1.228563304	-0.1663702005	-1.054847206	-1.034453942
16	16	-1.204093833	-1.11987375	-1.005089779	-0.996440863
17	17	-1.146957669	1.131418347	-0.9765030755	-0.9086949644
18	18	-1.12974161	1.335711391	-0.9556291923	-0.8867008702
19	19	-1.125268488	-1.351705786	-0.8552339283	-0.8673596378
20	20	-1.121009362	0.5871357547	-0.8507453644	-0.8397170269
21	21	-1.111578841	-0.4475671111	-0.8362659597	-0.8289226447
22	22	-1.10673758	0.2033756216	-0.81263024	-0.8112618702
23	23	-0.9787195761	-2.29679743	-0.7593589185	-0.8081755868

Figure 11. Loading the dataset and choosing the analysis in JASP

Step 2b: Fit the model

By selecting “Linear Regression” in the bain menu (Figure 11), you will be presented with a new graphical user interface that can be used to select the variables of interest (Figure 12). Set the variable “treatment.effect” as the dependent variable and the variables “age”, “dosage”, and “symptoms” as covariates.

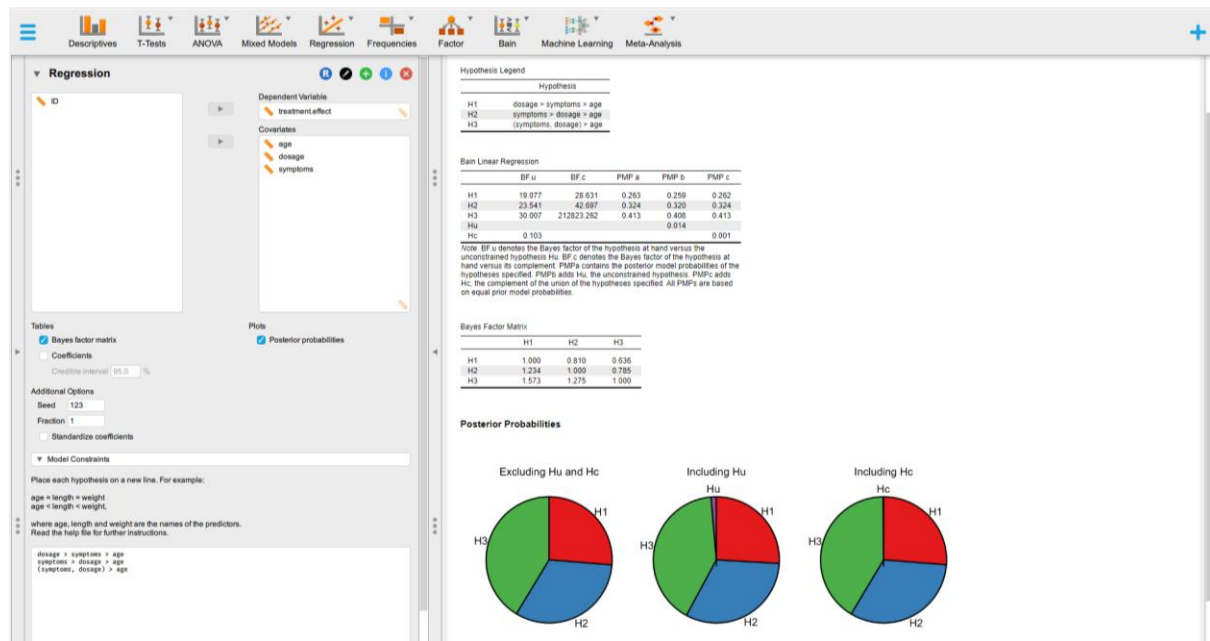


Figure 12. Definition of informative hypothesis and results

Step 3b: Define and test informative hypotheses with bain

Follow Step 3a of the JASP tutorial to set the hypotheses reported below, visualize the bain linear regression results, display the Bayes factor matrix, and plot the posterior model probabilities.

dosage > symptoms > age
symptoms > dosage > age
(symptoms, dosage) > age

Step-by-step tutorial in R

There are two ways to proceed with this example: (1) open a new R script and use the following lines of code to play along with the tutorial; or (2) open and use the file "Bain_REGRESSIONinR.R" available on the OSF page (<https://osf.io/dez9b/>), which contains all the steps and code reported below.

Preliminary steps

Follow the "Preliminary steps" section of the R tutorial for Example A.

Step 1b: Load the dataset

Follow Step 1a of the R tutorial to load the file "dataset_regression.txt" and visualize the dataset described above in "The regression dataset" paragraph.

Step 2b: Fit the model

The following code can be used to fit a linear model with the function `lm()`, which returns estimated coefficients for each independent variable.

```
fit <- lm(treatment.effect ~ age + dosage + symptoms, dataset_regression)
```

Step 3b: Define and test informative hypotheses with bain

Follow Step 3a of the R tutorial to test the hypotheses reported below, print the main results of the bain linear regression, display the Bayes factor matrix, and plot the posterior model probabilities (Figure 15).

```
set.seed(123)

results = bain( x = fit,
  hypothesis =
    "dosage > symptoms > age;
    symptoms > dosage > age;
    (symptoms, dosage) > age",
  standardize = TRUE)
```

NOW ANSWER THESE QUESTIONS:

- 4) Which of a set of hypotheses is the best?**
- 5) How much more likely a given hypothesis is relative to other possible explanations?**
- 6) How much more likely a given hypothesis is relative to another specific hypothesis?**