Calculating Bayes factors with R and JASP

Jeffrey R. Stevens

9 May 2019

Disclaimer: I am not a Bayesian statistics expert. This guide is not meant to accurately reflect the data analysis process in terms of establishing *a priori* hypotheses, checking test assumptions, or following proper statistical practice. It is only meant to illustrate general techniques for calculating Bayes factors. Please read the software documentation and Bayes factor literature to learn how to properly implement these techniques.

Import data

JASP

Open the data file: *File* → *Open* → *Computer* → *Browse* → `bf_data.csv`.

R

Open the `bayes_factor_workshop.Rproj` and open `src` → `bf_analysis.R` or create a new R script and type the text.

```r
# Load packages -----------------------------------------------------------
library(BayesFactor)  # needed to calculate Bayes factors
library(here)  # needed to assign working directory for relative paths
library(tidyverse)  # needed for importing, processing, and plotting data

# Import data -------------------------------------------------------------
datafile <- read_csv(here("data/bf_data.csv"), col_types = "nnffnnnn")
  # import data with columns 3 and 4 as factors and the other columns as numerics
datafile$condition <- factor(datafile$condition, levels = c("control", "experimental"))
  # reorder condition levels
```

Calculating Bayes factors from the data

One sample t-test

A one sample t-test tests the hypothesis that the mean of a sample differs from a known value (μ). In our data set, we have a column of data that represents the proportion of times that each participant has selected one of two options (`choice`). We may want to test if the participants are choosing randomly. Because there are two options, we would want test if the sample mean differs from 0.5.

- H_1: $\mu \neq 0.5$
- H_0: $\mu = 0.5$

JASP

To visualize the data, go to *Descriptives* → *Descriptive Statistics*, add `choice` to *Variables*, and click *Plots* → *Boxplots* → *Jitter Element*.

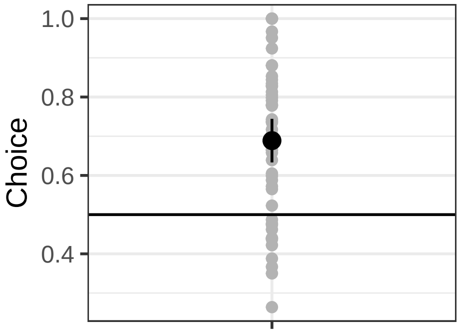
To conduct the frequentist one sample t-test, go to *T-Tests* → *One Sample T-Test*, add `choice` to *Variables*, and change the *Test value* to 0.5. You can alter the type of test and hypothesis direction and output normality assumption checks, effect sizes, descriptive statistics, plots, etc.

The Bayesian one sample t-test is under *T-Tests* → *Bayesian One Sample T-Test*. Otherwise, everything else is the same.

R

To visualize the data, with mean and 95% confidence interval:

```
ggplot(datafile, aes(x = "", y = choice)) +
  geom_point(col = "grey70") +  # plot individual points in grey
  stat_summary(fun.data = "mean_cl_normal") +  # add mean and 95% CI
  geom_hline(yintercept = 0.5) +  # add horizontal line at 0.5
  labs(x = NULL, y = "Choice") +  # label axes
  theme_bw()  # change themes
```



For the frequentist one sample t-test:

```
freq_onesample <- t.test(datafile$choice, mu = 0.5)  # frequentist one sample t-test
freq_onesample
```

```
##
##  One Sample t-test
##
## data:  datafile$choice
## t = 6.8234, df = 49, p-value = 1.246e-08
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.6331534 0.7443266
## sample estimates:
## mean of x
##   0.68874
```

The mean choice proportion is 0.689, and the p-value for the test that this mean does not equal 0.5 is p=1.25^{-8}.

The Bayesian t-test is almost exactly the same but uses the `ttestBF` function.

```r
bayes_onesample <- ttestBF(datafile$choice, mu = 0.5)  # Bayesian one sample t-test
bayes_onesample
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 1021239 ±0%
##
## Against denominator:
##   Null, mu = 0.5
## ---
## Bayes factor type: BFoneSample, JZS
```

The Bayes factor is 1021239. `r` controls the scale of the prior distribution and 0.707 is a standard value for this parameter, but it can be changed with the `rscale` argument. `±0%` represents the error percentage of the Bayes factor estimate. This error can be reduced by increasing the number of Monte Carlo samples run, using the `iteration` argument.

If we want to extract the Bayes factor, we use the `extractBF` function and add `$bf` to get just the Bayes factor. Try the function without the `$bf` to see the output.

```r
extractBF(bayes_onesample)$bf  # extract BF
```

```
## [1] 1021239
```

Holy cow, our Bayes factor is 1,021,239! So there is over a million times more evidence for H_1 than H_0. We have extreme evidence for our hypothesis that choice proportions are different from 0.5.

Independent samples t-test

An independent samples t-test tests that the mean of two independent samples differ from each other. In our data set, we have a column of data indicating whether each participant was in the experimental or control group (`condition`). We may want to test if the participants in the experimental condition have different choice proportions than those in the control condition.

- H_1: $\mu_{control} \neq \mu_{experimental}$
- H_0: $\mu_{control} = \mu_{experimental}$

JASP

To visualize the data, go to *Descriptives* → *Descriptive Statistics*, add `choice` to *Variables* and `condition` to *Split*, and click *Plots* → *Boxplots* → *Jitter Element*.

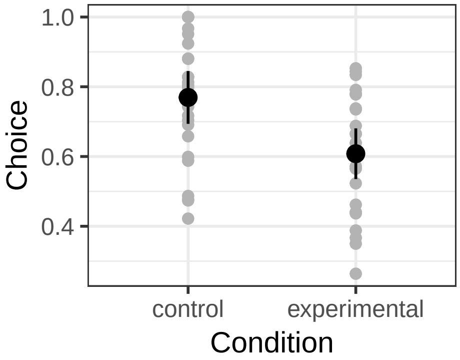
To conduct the frequentist independent samples t-test, go to *T-Tests* → *Independent Samples T-Test*, add `choice` to *Dependent Variables*, and add `condition` to *Grouping Variable*. You can alter the type of test and hypothesis direction and output assumption checks, effect sizes, descriptive statistics, plots, etc.

The Bayesian one independent t-test is under *T-Tests* → *Bayesian Independent Samples T-Test*. Otherwise, everything else is the same.

R

To visualize the data, with means and 95% confidence intervals:

```r
ggplot(datafile, aes(x = condition, y = choice)) +
  geom_point(col = "grey70") +  # plot individual points in grey
  stat_summary(fun.data = "mean_cl_normal") +  # add mean and 95% CIs
  labs(x = "Condition", y = "Choice") +  # label axes
  theme_bw()  # change themes
```



For the frequentist independent samples t-test:

```
freq_independ <- t.test(choice ~ condition, data = datafile)  # frequentist t-test
freq_independ
```

```
##
##  Welch Two Sample t-test
##
## data:  choice by condition
## t = 3.1747, df = 47.923, p-value = 0.002621
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.05914698 0.26349302
## sample estimates:
##     mean in group control mean in group experimental
##                   0.76940                    0.60808
```

The mean choice proportion for the control group is 0.769 and for the experimental group is 0.608, and the p-value for the test that the means are not equal is p=0.003.

The Bayesian t-test is almost exactly the same but uses the `ttestBF` function.

```
bayes_independ <- ttestBF(formula = choice ~ condition, data = datafile)  # Bayesian t-test
bayes_independ
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 14.0967 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

The Bayes factor is 14.1, so we have strong evidence for our hypothesis that choice proportions are different in the experimental and control groups.

Paired samples t-test

A paired samples t-test tests that the mean of two dependent/paired samples differ from each other. In our data set, we have columns of data indicating each participant's pre-intervention ratings (`pre`) and post-intervention ratings (`post`). We may want to test if participants' ratings differ before and after the intervention.

- H_1: $\mu_{pre} \neq \mu_{post}$
- H_0: $\mu_{pre} = \mu_{post}$

JASP

Visualizing the raw data in JASP is not straightforward, so we will skip it, and you can plot the means and CIs after running the analysis.

To conduct the frequentist paired samples t-test, go to *T-Tests* → *Paired Samples T-Test* and add `pre` and then `post` to *Variables*. You can alter the type of test and hypothesis direction and output assumption checks, effect sizes, descriptive statistics, plots, etc.

The Bayesian paired samples t-test is under *T-Tests* → *Bayesian Paired Samples T-Test*. Otherwise, everything else is the same.

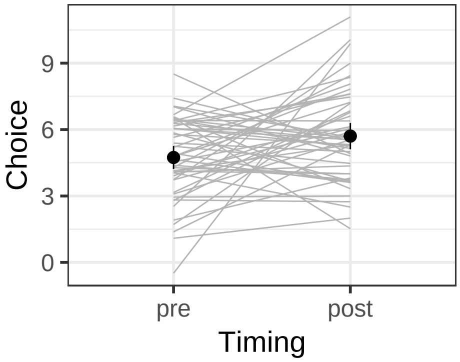
R

To visualize the data, we first need to rearrange the data:

```
prepost_data <- gather(select(datafile, participant, condition, pre, post),
  key = "pre_post", value = "rating", -participant, - condition) %>%
  # convert from wide to long data format
  mutate(participant = as.factor(participant),  # convert participant to factor
         pre_post = as.factor(pre_post))  # convert pre_post to factor
prepost_data$pre_post <- factor(prepost_data$pre_post, levels = c("pre", "post"))
  # reorder pre_post levels
```

Now we can plot the individual data points, means, and 95% CIs:

```
ggplot(prepost_data, aes(x = pre_post, y = rating)) +
  geom_line(aes(group = participant), col = "grey70", size = 0.25) +  # plot individual data
  stat_summary(fun.data = "mean_cl_normal", size = 0.25) +  # add mean and between-subjects
  # 95% CIs; note these are not accurate for a within-subjects comparison
  labs(x = "Timing", y = "Choice") +  # label axes
  theme_bw()  # change themes
```



For the frequentist paired samples t-test:

```
freq_paired <- t.test(datafile$pre, datafile$post, paired = TRUE)  # frequentist t-test
freq_paired
```

```
##
##  Paired t-test
##
## data:  datafile$pre and datafile$post
## t = -2.4037, df = 49, p-value = 0.02006
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.7750789 -0.1585211
## sample estimates:
## mean of the differences
##                 -0.9668
```

The mean difference in ratings is -0.967, and the p-value for the test that the means are not equal is p=0.02.

The Bayesian t-test is almost exactly the same but uses the `ttestBF` function.

```r
bayes_paired <- ttestBF(datafile$pre, datafile$post, paired = TRUE)  # Bayesian t-test
bayes_paired
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 2.07768 ±0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

The Bayes factor is 2.1, so we do not have evidence for our hypothesis that ratings differ before or after the intervention. We also do not have enough evidence to state that there is no difference. We need to collect more data to reach a threshold of evidence. One thing that is interesting about this example is that the frequentist approach found a 'significant' result, but the Bayesian analysis did not find enough evidence to support one hypothesis over another. This is common with p-values between 0.01-0.05.

ANOVA

An analysis of variance tests that the mean of two or more samples differ from each other. In our data set, we have a column of data indicating each participant's gender (`gender`). We may want to test if the participants' choice proportions differ according to their gender and whether there is an interaction between gender and condition. *Note: Due to unequal sample sizes, the homogeneity of variance assumption of ANOVAs is clearly violated—this example is for illustrative purposes only.*

JASP

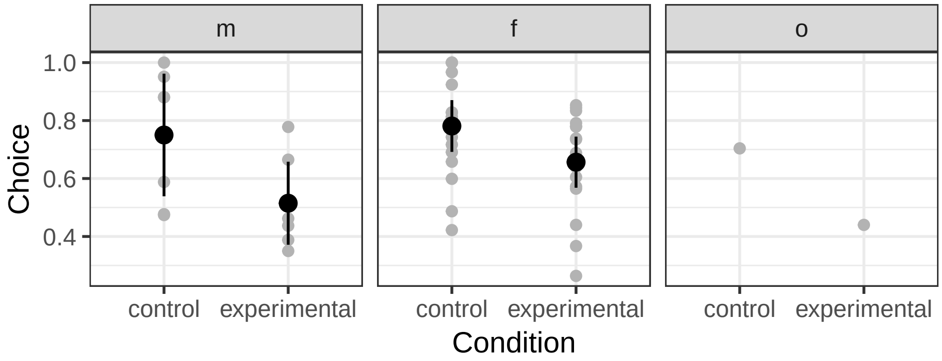
Again, we cannot visualize the data in JASP until after the analysis. To conduct a fixed effects ANOVA, go to *ANOVA* → *ANOVA* and enter `choice` as a *Dependent Variable* and `gender` and `condition` into a *Fixed Factors*. There are lots of options to specify the model, check assumptions, plot results, and run contrasts, post-hoc tests, and nonparametric tests.

The Bayesian ANOVA is under *ANOVA* → *Bayesian ANOVA*. Otherwise, everything else is the same.

R

To visualize the data, with means and 95% confidence intervals:

```
ggplot(datafile, aes(x = condition, y = choice)) +
  facet_wrap(~gender) +  # facet by gender
  geom_point(col = "grey70") +  # plot individual points in grey
  stat_summary(fun.data = "mean_cl_normal") +  # add mean and 95% CIs
  labs(x = "Condition", y = "Choice") +  # label axes
  theme_bw()  # change themes
```



For the frequentist ANOVA:

```
freq_aov <- summary(aov(choice ~ condition * gender, data = datafile))  # frequentist ANOVA
freq_aov
```

```
##                  Df Sum Sq Mean Sq F value  Pr(>F)
## condition         1 0.3253  0.3253  10.144 0.00266 **
## gender            2 0.1024  0.0512   1.596 0.21423
## condition:gender  2 0.0359  0.0179   0.559 0.57580
## Residuals        44 1.4110  0.0321
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the main effect of condition is p=0.003, the main effect of gender is p=0.214, and the interaction is p=0.576. We use the `summary` function to get the ANOVA table. Try running the code without it to see the default `aov` output.

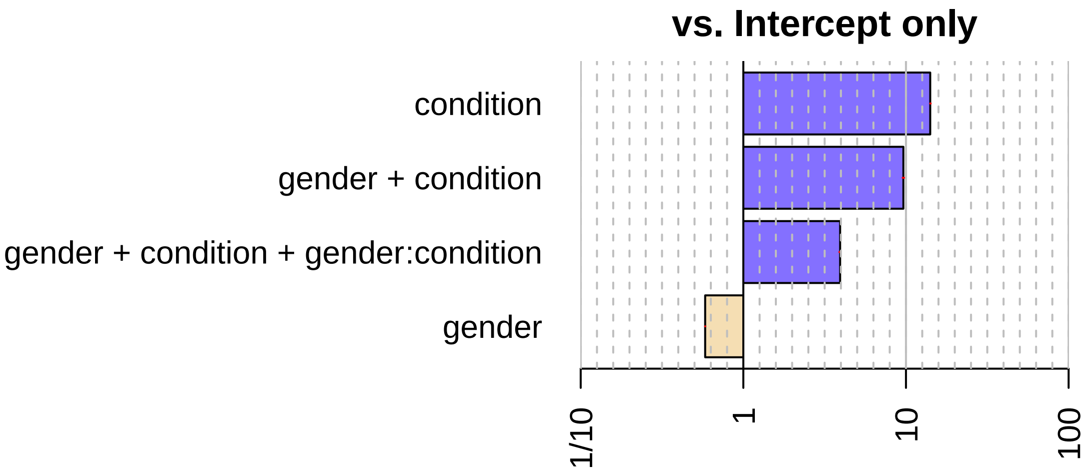
The Bayesian t-test is almost exactly the same but uses the `anovaBF` function.

```
bayes_aov <- anovaBF(choice ~ condition * gender, data = datafile)  # Bayesian ANOVA
bayes_aov
```

```
## Bayes factor analysis
## --------------
## [1] gender                                : 0.5824818 ±0.02%
## [2] condition                             : 14.0967   ±0%
## [3] gender + condition                    : 9.64021   ±1.1%
## [4] gender + condition + gender:condition : 3.936143  ±1.08%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

The highest Bayes factor is 14.1 (for the model with condition `[2]`), so we have strong evidence for our hypothesis that choice proportions are different in the experimental and control groups. Note that this Bayes factor compares the model with condition (only) to a null model with intercept only. So there is 14 times more evidence for the model with condition than the model with nothing. The model with gender and condition [3] has a Bayes factor of 9.6 and the model with gender, condition, and their interaction [4] has a Bayes factor of 3.9. We can plot the Bayes factors.

```
plot(bayes_aov)  # plot ANOVA results
```



Though there is evidence that the model with condition and gender is better than the null, the model with condition only [2] has a higher Bayes factor. How do we compare models? You can just divide the Bayes factors with the alternative model in the numerator and the null in the denominator. In this case, we can divide the highest Bayes factor [2] by the other two supported models [3] and [4].

```
bayes_aov[2] / bayes_aov[3:4]  # calculate BFs relative to best model
```

```
##             denominator
## numerator   gender + condition gender + condition + gender:condition
##   condition           1.462281                              3.581349
```

Though the condition only model has the highest Bayes factor, we do not have evidence that it is better than the next best model (condition and gender)—but we have moderate evidence that it is better than the model with the interaction.

Repeated measures/mixed ANOVA

A repeated measures analysis of variance tests that the mean of two or more dependent samples differ from each other. In our data set, we have columns of ratings taken before (pre) and after (post) an intervention. We will use these columns for the JASP analysis. For R, we have created a long version of the data set with a column of data indicating each participant's ratings (rating) and a column indicating whether the rating was before or after the participant's intervention (pre_post). We may want to test if the participants' ratings differ according to whether they were taken before or after the intervention and whether there is an interaction between pre-post and condition.

JASP

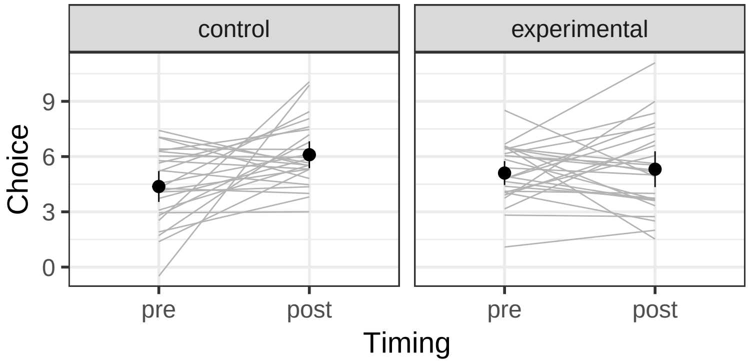
Again, we cannot visualize the data in JASP until after the analysis. To conduct a fixed effects ANOVA, go to *ANOVA* → *Repeated Measures ANOVA* and add pre and post to *Repeated Measures Cells* and condition to *Between Subjects Factors*. There are lots of options to specify the model, check assumptions, plot results, and run contrasts, post-hoc tests, and nonparametric tests.

The Bayesian ANOVA is under *ANOVA* → *Bayesian Repeated Measures ANOVA*. Otherwise, everything else is the same.

R

To visualize the data, with means and 95% confidence intervals:

```
ggplot(prepost_data, aes(x = pre_post, y = rating)) +
  facet_wrap(~condition) +  # facet by condition
  geom_line(aes(group = participant), col = "grey70", size = 0.25) +  # plot individual data
  stat_summary(fun.data = "mean_cl_normal", size = 0.25) +  # add mean and between-subjects
  # 95% CIs; note these are not accurate for a within-subjects comparison
  labs(x = "Timing", y = "Choice") +  # label axes
  theme_bw()  # change themes
```



For the frequentist ANOVA:

```
freq_rm <- summary(aov(rating ~ condition * pre_post + Error(participant /
  (condition * pre_post)), data = prepost_data))  # frequentist mixed effects ANOVA
freq_rm
```

```
##
## Error: participant
##           Df Sum Sq Mean Sq F value Pr(>F)
## condition  1   0.02   0.019   0.005  0.944
## Residuals 48 182.64   3.805
##
## Error: participant:pre_post
##                    Df Sum Sq Mean Sq F value Pr(>F)
## pre_post            1  23.37   23.37   6.102 0.0171 *
## condition:pre_post  1  14.35   14.35   3.747 0.0588 .
## Residuals          48 183.83    3.83
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the main effect of pre-post is p=0.017, condition is p=0.944, and the interaction is p=0.059.

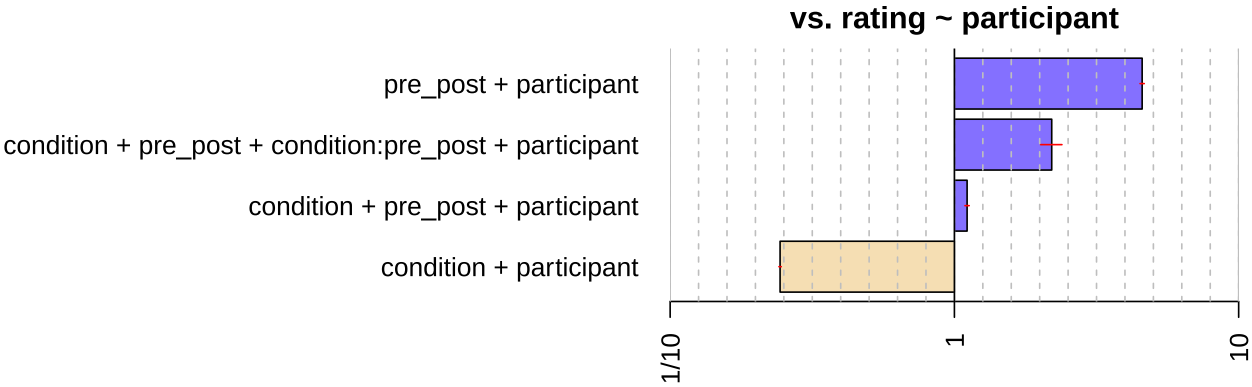
Here is the Bayesian ANOVA, where `whichRandom` specifies the random variable(s).

```
bayes_rm <- anovaBF(rating ~ condition * pre_post + participant,
  data = prepost_data, whichRandom = "participant")  # Bayesian mixed effects ANOVA
bayes_rm
```

```
## Bayes factor analysis
## --------------
## [1] condition + participant                                 : 0.2436283 ±0.93%
## [2] pre_post + participant                                  : 4.576034  ±1.7%
## [3] condition + pre_post + participant                      : 1.108311  ±1.71%
## [4] condition + pre_post + condition:pre_post + participant : 2.199641  ±8.64%
##
## Against denominator:
##   rating ~ participant
## ---
## Bayes factor type: BFlinearModel, JZS
```

The highest Bayes factor is 4.6 (for the model with condition [2]), so we have moderate evidence for our hypothesis that ratings are different before and after the intervention. The model with pre-post and condition and the interaction [4] has a Bayes factor of 2.2. We can plot the Bayes factors.

```
plot(bayes_rm)  # plot Bayesian results
```



Correlation

A correlation tests the extent to which, as one variable increases, another variable tends to increase. In our data set, we have a column of data indicating each participant's reaction time (rt) and a column indicating each participant's choice proportion (choice). We may want to test if the participants' reaction times correlate with choices.

JASP

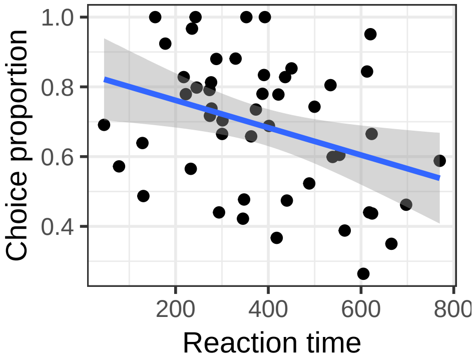
Again, we cannot visualize the data in JASP until after the analysis. To conduct a correlation, go to *Regression* → *Correlation Matrix* and enter choice and rt as *Variables*. You can alter the type of test and hypothesis direction and output various statistics, plots, etc.

The Bayesian ANOVA is under *Regression* → *Bayesian Correlation Matrix* or *Bayesian Correlation Pairs*. Otherwise, everything else is the same.

R

To visualize the data with a 95% confidence interval band:

```
ggplot(datafile, aes(x = rt, y = choice)) +
  geom_point() +  # plot individual points in grey
  geom_smooth(method = "lm") +  # plot regression line and CI
  labs(x = "Reaction time", y = "Choice proportion") +  # label axes
  theme_bw()  # change themes
```



For the frequentist correlation:

```
freq_cor <- cor.test(datafile$rt, datafile$choice)  # frequentist correlation test
freq_cor
```

```
##
##  Pearson's product-moment correlation
##
## data:  datafile$rt and datafile$choice
## t = -2.5483, df = 48, p-value = 0.01408
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.56888731 -0.07396064
## sample estimates:
##        cor
## -0.3452018
```

The correlation coefficient is -0.345, and the p-value is p=0.014.

The Bayesian correlation is almost exactly the same but uses the `correlationBF` function.

```
bayes_cor <- correlationBF(datafile$rt, datafile$choice)  # Bayesian correlation test
bayes_cor
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.333 : 4.82018 ±0%
##
## Against denominator:
##   Null, rho = 0
## ---
## Bayes factor type: BFcorrelation, Jeffreys-beta*
```

The Bayes factor is 4.8, suggesting that we have moderate evidence for a correlation.

Linear regression

A linear regressions tests for a linear relationship between two or more variables. In our data set, we have columns of data indicating each participant's choice proportion (`choice`), reaction time (`rt`), and age (`age`). We may want to test whether reaction times or age predict choice.

JASP

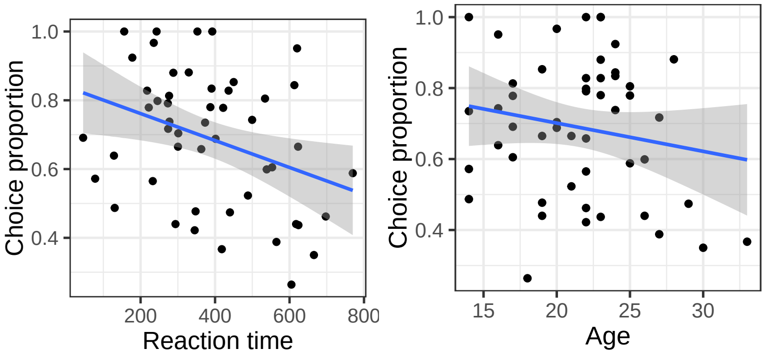
Again, we cannot visualize the data in JASP until after the analysis. To conduct a regression, go to *Regression* → *Linear Regression* and enter `choice` in *Dependent Variable* and `rt` and `age` in *Covariates*. There are lots of options to specify the model, check assumptions, and calculate other statistics.

The Bayesian ANOVA is under *Regression* → *Bayesian Linear Regression*. Otherwise, everything else is the same.

R

To visualize the data with a 95% confidence interval band:

```r
ggplot(datafile, aes(x = rt, y = choice)) +
  geom_point(size = 4) +  # plot individual points in grey
  geom_smooth(method = "lm", size = 2) +  # plot regression line and CI
  labs(x = "Reaction time", y = "Choice proportion") +  # label axes
  theme_bw(base_size = 30)  # change themes and increase axis font sizes
ggplot(datafile, aes(x = age, y = choice)) +
  geom_point(size = 4) +  # plot individual points in grey
  geom_smooth(method = "lm", size = 2) +  # plot regression line and CI
  labs(x = "Age", y = "Choice proportion") +  # label axes
  theme_bw(base_size = 30)  # change themes and increase axis font sizes
```



For the frequentist linear regression:

```r
freq_lm <- summary(lm(choice ~ rt + age, data = datafile))  # frequentist regression
freq_lm
```

```
##
## Call:
## lm(formula = choice ~ rt + age, data = datafile)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35968 -0.15543  0.01536  0.15909  0.32444
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9200280  0.1376773   6.682 2.48e-08 ***
## rt          -0.0003645  0.0001602  -2.275   0.0275 *
## age         -0.0042119  0.0063613  -0.662   0.5111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1866 on 47 degrees of freedom
## Multiple R-squared:  0.1273, Adjusted R-squared:  0.09017
## F-statistic: 3.428 on 2 and 47 DF,  p-value: 0.04076
```

Reaction time predicts choice with a p-value of p=0.028, but age does not predict choice.

The Bayesian linear regression is almost exactly the same but uses the `regressionBF` function.

```
bayes_lm <- regressionBF(choice ~ rt + age, data = datafile)  # Bayesian regression
bayes_lm
```

```
## Bayes factor analysis
## --------------
## [1] rt       : 3.707926  ±0%
## [2] age      : 0.5313747 ±0%
## [3] rt + age : 1.496802  ±0.01%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

The Bayes factor for reaction time is 3.7, suggesting moderate evidence for a relationship between reaction time and choice.

Converting frequentist statistics to Bayes factors

Sometimes it can be useful to calculate Bayes factors from existing analyses (e.g., if the full data are not available). This is possible for binomial tests, t-tests, one-way ANOVAs, correlations, and regressions if you have the test statistics and sample sizes.

In JASP, click the + to the right of *Common* and select *Summary Stats*. From there, you can choose your test and input the proper values to calculate a Bayes factor.

In R, you can use the set of `tstat` functions to do the same.

One-sample t-test

```
t.test(datafile$choice, mu = 0.5)  # frequentist t-test
```

```
##
##  One Sample t-test
##
## data:  datafile$choice
## t = 6.8234, df = 49, p-value = 1.246e-08
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.6331534 0.7443266
## sample estimates:
## mean of x
##   0.68874
```

```
ttest.tstat(t = 6.8234, n1 = 50, simple = TRUE)  # convert t statistic to BF
```

```
##     B10
## 1021377
```

Check if this matches the Bayes factor that we found previously.

One-way ANOVA

```
summary(aov(choice ~ condition, data = datafile))  # frequentist ANOVA
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)   
## condition    1 0.3253  0.3253   10.08 0.00262 **
## Residuals   48 1.5492  0.0323                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
oneWayAOV.Fstat(F = 10.08, N = 25, J = 2, simple = TRUE)  # convert F statistic to BF
```

```
##      B10 
## 14.10317
```

Linear regression

```
summary(lm(choice ~ rt, data = datafile))  # frequentist linear regression
```

```
## 
## Call:
## lm(formula = choice ~ rt, data = datafile)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.33855 -0.15583  0.01664  0.15168  0.35441 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.8397330  0.0647990  12.959   <2e-16 ***
## rt          -0.0003920  0.0001538  -2.548   0.0141 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1855 on 48 degrees of freedom
## Multiple R-squared:  0.1192, Adjusted R-squared:  0.1008 
## F-statistic: 6.494 on 1 and 48 DF,  p-value: 0.01408
```

```
linearReg.R2stat(N = 50, p = 1, R2 = 0.0164, simple = TRUE)  # convert R^2 statistic to BF
```

```
##       B10 
## 0.3924225
```

Approximating Bayes factors from BICs

Currently, the `BayesFactor` package does not calculate Bayes factors for Generalized Linear Models (GLM). However, there is a way to estimate Bayes factors from the Bayesian Information Criterion (BIC) values of

GLMs, using the following equation (Wagenmakers, 2007):

$$BF = e^{\frac{BIC_{null} - BIC_{alternative}}{2}}$$

Though this is not available in JASP, it can be used in R to calculate Bayes factors. First, we need to define the function that converts BICs to BFs:

```
bic_bf10 <- function(null, alternative) {
  new_bf <- exp((null - alternative) / 2) # convert BICs to Bayes factor
  names(new_bf) <- NULL    # remove BIC label
  return(new_bf)  # return Bayes factor of alternative over null hypothesis
}
```

Next, we need to conduct our GLMs for our two models. We'll use the `lme4` package to run the GLMs. Here, we'll examine whether there is more evidence for the model including the interaction between condition and gender on reaction time compared to the model without the interaction.

```
library(lme4)  # needed to calculate mixed models
lm1 <- glm(rt ~ condition * gender, data = datafile, family = Gamma())  # with interaction
lm0 <- glm(rt ~ condition + gender, data = datafile, family = Gamma())  # no interation
```

Now we need to extract the BIC values for each model and convert them to a Bayes factor.

```
lm1_bic <- BIC(lm1)  # extract BIC
lm0_bic <- BIC(lm0)  # extract BIC
bic_bf10(lm0_bic, lm1_bic)  # convert BICs to BF
```

```
## [1] 0.02986607
```

We have very strong evidence for the model with no interaction.

References and resources

- *Psychonomic Bulletin & Review* special issue on Bayesian methods for advancing psychological science
- `BayesFactor` package
- JASP statistical software
- Bayesian Spectacles blog
- Understanding Bayes blog

- Anderson, D. R. (2008). Model Based Inference in the Life Sciences. New York: Springer.
- Etz, A., Gronau, Q. F., Dablander, F., Edelsbrunner, P. A., & Baribault, B. (2018). How to become a Bayesian in eight easy steps: An annotated reading list. *Psychonomic Bulletin & Review*, 25(1), 219–234. https://doi.org/10.3758/s13423-017-1317-5
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis (Third Edition). Boca Raton, FL: CRC Press.
- Kruschke, J. K. (2015). Doing Bayesian Data Analysis (Second Edition). New York: Academic Press.
- Rouder, J. N. (2014). Optional stopping: No problem for Bayesians. *Psychonomic Bulletin & Review*, 21(2), 301–308. https://doi.org/10.3758/s13423-014-0595-4
- Rouder, J. N., Haaf, J. M., & Vandekerckhove, J. (2018). Bayesian inference for psychology, part IV: parameter estimation and Bayes factors. *Psychonomic Bulletin & Review*, 25(1), 102–113. https://doi.org/10.3758/s13423-017-1420-7
- Schönbrodt, F. D., & Wagenmakers, E.-J. (2018). Bayes factor design analysis: Planning for compelling evidence. *Psychonomic Bulletin & Review*, 25(1), 128–142. https://doi.org/10.3758/s13423-017-1230-y

- Schönbrodt, F. D., Wagenmakers, E.-J., Zehetleitner, M., & Perugini, M. (2017). Sequential hypothesis testing with Bayes factors: Efficiently testing mean differences. *Psychological Methods*, 22(2), 322–339. https://doi.org/10.1037/met0000061
- Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11), 1359–1366. https://doi.org/10.1177/0956797611417632
- van Doorn, J., van den Bergh, D., Bohm, U., Dablander, F., Derks, K., Draws, T., … Wagenmakers, E.-J. (2019). The JASP guidelines for conducting and reporting a Bayesian analysis. *PsyArXiv.* https://doi.org/10.31234/osf.io/yqxfr
- Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems of p values. *Psychonomic Bulletin & Review*, 14(5), 779–804. https://doi.org/10.3758/BF03194105
- Wagenmakers, E.-J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Love, J., … Morey, R. D. (2018). Bayesian inference for psychology. Part I: Theoretical advantages and practical ramifications. *Psychonomic Bulletin & Review*, 25(1), 35–57. https://doi.org/10.3758/s13423-017-1343-3
- Wasserstein, R. L., & Lazar, N. A. (2016). The ASA's statement on p-values: Context, process, and purpose. *American Statistician*, 70(2), 129–133. https://doi.org/10.1080/00031305.2016.1154108