the
WINNOWER

MATHEMATICS

Using Bayes Factors to Get the Most out of Linear Regression: A Practical Guide Using R

ALEXANDER ETZ

ABSTRACT

This guide is for readers who want to make rich inferences from their data. It does not contain anything new with regard to R code or theoretical development, but it does piece together information in an easy to follow guide. An article was recently published in a journal that is probably not well known by most researchers, *Multivariate Behavioral Research,* where the authors discuss the theoretical background of Bayesian inference and provide a simple web calculator for linear regression (Rouder & Morey, 2012; http://pcl.missouri.edu/bf-reg). More recently, R code has been developed (Morey, Rouder, & Jamil, 2015) and compiled to conduct many different types of Bayesian analyses (Morey, 2015; http://bayesfactorpcl.r-forge.r-project.org/). However, this leaves information scattered across websites and journal pages, and so it is not easily taken in by a researcher who doesn't know where to start with either Bayesian theory or R software. This guide has been put together in a way that a researcher who has never used Bayesian inference or R software could conduct, report, and interpret a Bayesian linear regression. It skips over the complicated derivations and mathematics behind the theory, and focuses on why Bayes factors are valuable and desirable to researchers. It then shows step-by-step how R software can be used for Bayesian model comparison as well as parameter estimation in linear regression problems.

READ REVIEWS

WRITE A REVIEW

CORRESPONDENCE:
etz.alexander@gmail.com

DATE RECEIVED:
June 10, 2015

KEYWORDS:
linear regression, bayes factor, model selection, bayesian statistics, r programming

INTRODUCTION

Researchers often want to make claims that two or more groups of participants show the same (or close to the same) performance on a task. For example, a psychologist may seek to demonstrate that an effect is equally strong for men and women. The conventional way to demonstrate this is to collect data on men and women and compare their scores; if the difference between the group means is close to zero and a nonsignificant p-value is returned, the researcher states that there was no evidence for an effect of gender.

But notice the phrasing, “there was *no evidence for an effect* of gender” is not the same claim as, “there was *evidence for no effect* of gender”. This language contortion must be performed because classic inference procedures using p-values cannot be used to support a null-hypothesis (e.g., no difference between groups; Dienes, 2014; Fisher, 1955). As Fisher (1966) famously states,

> it should be noted that the null hypothesis is never proved or established, but is possibly disproved, in the course of experimentation. Every experiment may be said to exist only in order to give the facts a chance of disproving the null hypothesis … The notion of an error of the so-called ‘second kind,’ due to accepting the null hypothesis ‘when it is false’ … has no meaning with reference to simple tests of significance. (as cited in Royall, 1997, p. 73)

In other words, the null hypothesis has merely survived until its next test. This is a major limitation for conventional statistics, leading researchers to often find themselves on “the wrong side of the null hypothesis” (Rouder & Morey, 2012, pg. 878). Researchers want so strongly to be able to assert that two groups are equivalent that they will ignore this limitation of p-values and continue to accept the null hypothesis without proper justification.

For example, researchers analyzing the effect of race on participants judgment of perceived probability to commit aggressive acts wrote: “Interestingly, variation in the slopes for Afrocentric features was not significant, suggesting no individual differences in the degree to which higher aggression probability estimates were given for targets with more Afrocentric facial features.” These researchers have committed a statistical sin by accepting the null hypothesis that the slopes are equivalent after achieving a non-significant result.

Another example, this time the researchers are more explicit in their conclusion: “Testing the variance components in this mixed model revealed that the variance of the gun by target race effect across participants was not significantly different from 0, meaning that the shooter effect is basically the same for all participants. This suggests that one might look in vain for individual differences that might moderate the effect.” I won’t identify the papers here because the quotes are sufficient to stand alone without pointing fingers at any particular researchers.

These quotes are exemplars of a widespread problem: The literature of every field is littered with more or less explicit forms of accepting the null hypothesis when the p-value test is not significant. P-value tests have a fatal asymmetry built-in to them: Either reject the null hypothesis or else say there was simply not sufficient evidence to reject it. In the words of Carl Sagan, “Absence of evidence is not evidence of absence.” However, there is a type of inference that can overcome this and many other limitations of conventional inference, and it is called the Bayes factor (Kass & Raftery, 1995).

WHAT IS A BAYES FACTOR?

A Bayes factor is a measure of the relative predictive success of two competing models. A researcher must state what they believe (or what their theory predicts) about the true value of a parameter, known as a prior probability distribution over a parameter, to indicate her uncertainty about that parameter.[1]

For example, Marty might think the most likely value for a parameter is 30, but he recognizes that there is uncertainty in his guess; he thinks the parameter’s value could plausibly range from 10 to 50, with values beyond those bounds being highly implausible to him. Marty’s belief could be represented by a normal distribution with $\mu = 30$ and $\sigma = 10$. Other analysts can specify their prior differently. For example, Biff might think the parameter value of 60 is most likely, but it could plausibly range from as low as 40 to as high as 80. Biff’s belief could be represented by a normal distribution with $\mu = 60$ and $\sigma = 10$. If the results of the experiment indicate that the best estimate for the parameter is equal to 50, then Biff did a much better job predicting the data than did Marty.

This can be intuitively shown by noting that the parameter estimate lies well inside Biff’s range of plausible values whereas it is just on the cusp of Marty’s. You can see this in the graph below, where Marty’s belief is represented by the curve in red and Biff’s belief is represented by the curve in blue. The dotted line represents the data obtained in the experiment. Notice how much higher the blue curve is than the red curve when they cross the dotted line. **Bayes factors quantify *precisely how much better* Biff’s prior belief distribution predicted the data than did Marty’s.**

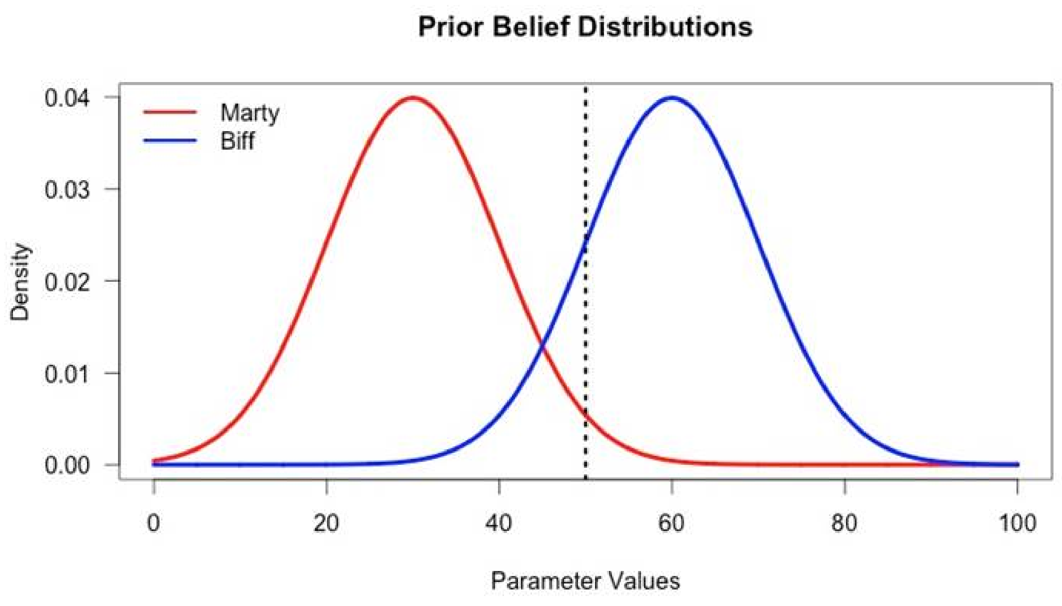


Figure 1. Prior belief distributions for Marty (red) and Biff (blue). Marty specified a Normal (m=30, sd =10) and Biff specified a Normal (m=60, sd=10). The dotted line is the data.

Marty and Biff have approximately represented their prior beliefs by normal curves, and in doing so they have made predictions about what data they might find in the experiment. They can quantify their relative predictive ability for the data by calculating the predictive success of each prior belief distribution and then dividing one by the other to obtain a ratio. This ratio of the prior distributions' predictive success is the Bayes factor. I won't go into the technical details of deriving and calculating the Bayes factor, but I point the interested reader to the BayesFactor package's blog for an easy to follow introduction (Morey, 2014; http://bayesfactor.blogspot.co.uk/2014/02/the-bayesfactor-package-this-blog-is.html). Additionally, I will not go into the justification and defense of personalistic probabilities; additional reading on this can be found in Dienes (2008), Lindley (2000), and Savage (1954).

BAYES FACTORS HAVE MANY DESIRABLE PROPERTIES

1) Bayes factors solve the problem of inferential asymmetry by treating the null hypothesis as just an ordinary model. Just as Biff and Marty assigned normal distributions to their beliefs, the null-hypothesis assigns all of its probability to a single null point.[2] Since the null hypothesis has no special status for Bayes factors, it can get the same treatment as any other model. This means that Bayes factors have *perfect inferential symmetry*: Support for the null model is possible, support for the alternative model is possible, and inconclusive support that favors neither model is possible.

2) There is no penalty for optional stopping. The Bayes factor is a ratio of (marginal) likelihoods, and the likelihood principle tells us that the sampling plan is irrelevant; only the actual data obtained influence the likelihood function, and thus only the data actually obtained factor into the quantification of evidence (A. Edwards, 1972; Royall, 1997). *How this evidence is obtained* does not factor into the likelihood function. This means that the researcher can "collect data until a point has been proven or disproven, or the data collector runs out of time, money, or patience" (W. Edwards, Lindman, & Savage, 1963).

A related rule that follows from the likelihood principle is that multiple comparisons are not problematic for Bayesian inference, unlike conventional inference. The data alone determine the strength of evidence, not how many other places we looked or what day the theory was thought up or how many different ways the data were analyzed (Dienes, 2008).

3) Bayes factors give the researcher a directly interpretable number that quantifies the evidence provided by the data for one model (hypothesis) or another. A Bayes factor of 10 means that the data are 10 times more probable under one model (hypothesis) than another. This number, and its interpretation, does not depend on stopping intention, sample size, when the hypothesis was specified, or how many comparisons were made. A Bayes factor of 10 is a Bayes factor of 10 is a Bayes factor of 10. Always. This gives the analyst immense flexibility, while maintaining a simple and straightforward dissemination of results.[3]

4) The prior probability distribution is often cited as the subjective element interfering with the general adoption of Bayesian inference, but it is actually a boon. It has been proposed that if only we could develop an objective prior (Berger, 2006; also see commentaries) we could avoid the criticism and bickering amongst ourselves, and get back to doing the science we love. As it turns out, attempts to specify a purely objective prior distribution have so far been unsuccessful, and some contend that it is not desirable or even possible in the first place (Goldstein, 2006; also see commentaries). In fact, deriving a purely objective prior has been likened to the Holy Grail, "The goal is elusive, the exercise is fruitless, and the enterprise is ultimately diverting our energies from *doing* quality statistics (Fienberg, 2006 pg. 431; emphasis original).

For many types of analysis, specifying an appropriate prior probability distribution can be tough. However, it is through this specification that researchers can intimately connect data to theory (Dienes, 2008), move toward more quantitative theories (Vanpaemel, 2010), and move toward a more quantitative scientific enterprise as a whole. Bayes factors elegantly link a researcher's theory to empirical data; "subjective" priors could be more aptly called "contextual" or "conditional" priors (Christen, 2006). See Dienes (2008, 2011), Lee and Wagenmakers (2005), and Wagenmakers (2007) for more detailed exposition on prior probabilities and Bayes factors.

HOW CAN BAYES FACTORS BE CALCULATED AND USED?

This introduction to Bayes factors may be interesting to the reader who has made it this far, but if one does not know how to use a method then there is no point reading or talking about it. For the rest of this guide I will give an outline of how Bayes factors can be used for model comparison in linear regression problems. This guide will explain how to install R and the accompanying software R Studio. It will then explain step by step how to import real data and use the BayesFactor R package (Morey et al., 2015) to conduct, interpret, and report Bayesian regression model comparisons. Later in this demonstration I will use a default dataset that comes with R (R Core Team, 2015) called stackloss. **(All R code in this guide is highlighted green, since the Winnower does not allow code tags at the moment.)**

In order to calculate a Bayes factor one must define the parameter's prior probability distribution for every model under consideration (like Marty and Biff did above). This can be the hardest part of Bayesian inference. One way of doing so is to use a prior that has desirable properties, so-called "default" priors that are appropriate in many situations (see Rouder & Morey, 2012, specifically p.883 for details. That paper gets pretty derivation and equation-intense, so feel free to skip it unless you want more detail).

Rouder and Morey (2012) outline a default prior distribution derived by Liang, Paulo, Molina, Clyde, & Berger (2008), which has the following desirable properties: 1) These default priors are location and scale invariant. This means that a Bayes factor wouldn't change if, say, temperature is measured in Kelvin, Celsius, or Fahrenheit. 2) They are also consistent. This means that if infinite data could be collected, the Bayes factor will favor the true model by an infinite ratio. 3) Finally, these priors are consistent in information. This means that if a predictor perfectly accounts for the behavior of an outcome variable (R^2 = 1) then the Bayes factor approaches infinity in favor of that predictor over a model without that predictor.

A quick note about default priors. Default priors can be useful in many instances, but it is important to always make sure they are consistent with one's theory or hypothesis. Model comparison only makes sense when the models being considered are theoretically relevant. If a theory clearly states a directional difference, then a directional prior can be used (Dienes, 2008). Directional priors do not have the same baggage as one-tailed tests do in conventional inference, and directional default priors are available for many situations (Morey & Wagenmakers, 2014).

With that out of the way, let's get on to the fun stuff! I am going to teach you how to go from raw data to Bayes factor using simple inputs in R. I strongly suggest using R Studio, which is a prettier and more user friendly than the base R program. This guide can be used by anyone who has a computer and an internet connection, simply copy and paste the code snippets and hit 'enter' to follow along.

GETTING STARTED WITH R

If you don't use R you can download it here: http://cran.r-project.org/bin/windows/base/ and then run it from a program called R Studio, which you can download here: http://www.rstudio.com/products/rstudio/download/. Ryne Sherman has more instructions and a simple intro course for getting started here: http://psy2.fau.edu/~shermanr/RCourse.html. You can simply copy and paste the code into the R Studio console as we go.

This guide is largely adapted from Morey's, "Using the 'BayesFactor' package" (2015, http://bayesfactorpcl.r-forge.r-project.org/), and I make no claim that any of this code is original. In fact, almost nothing written below is new. I am simply putting it all together in one place and trying to make it user-friendly for those who don't use R.

Load up R Studio and you'll see notes about what version of R you are using. You can ignore all of that. You'll notice a blinking cursor. This command line indicates that you are ready to start using R! Lines of code below that are preceded by ## indicates output from an R function.

If you have your own dataset that you'd like to use, R can import it from many types of files, including .txt, .csv, .sav (with the Foreign or Haven package), etc. For example, if you use the read.csv() command, you can import the data from any locally saved .csv file by using the file.choose() command. It would look like this:

```
my.data = read.csv(file.choose())
```

Be careful about capitalization, as R reads "my.data" as a different object than "My.data". When you hit enter it will bring up a menu you can navigate through to find your data file. Then check that it loaded correctly by typing:

```
summary(my.data)
```

I am going to use one of the datasets that R comes pre-equipped with called stackloss for illustrative purposes so all readers can follow along without having to download an external dataset or use their own. The stackloss dataset contains 3 predictors (water temperature, air flow, acid concentration) and 1 outcome (stack loss), a perfect dataset to use as an example of how to get started using regression Bayes factors (for more information about this data, type help(stackloss)).

To begin using the stackloss dataset, type the following:

```
data(stackloss)
```

This loads the data into the R workspace so that we can use it. Next, type:

```
stackloss
```

This shows the data values for each variable. To see descriptive statistics for this dataset type:

```
summary(stackloss)
```

This will display quartiles, means, and medians for each variable.

CONVENTIONAL LINEAR REGRESSION

Let's see what kind of result we would get if we used conventional linear regression with the stackloss dataset. Type the following:

```
conventionalLR = lm(stack.loss ~ ., data = stackloss)
```

This creates an object in R called conventionalLR that contains the summary information for the conventional linear regression. The function lm() says to conduct a linear regression using the stackloss data, using the variable stack.loss as the outcome, and using the rest of the variables as predictors. Now type:

```
summary(conventionalLR)
```

This gives us a summary of the linear regression output.

```
## summary(conventionalLR)
##
## Call:
## lm(formula = stack.loss ~ ., data = stackloss)
##
##    Residuals:
##         Min        1Q    Median        3Q       Max
##     -7.2377   -1.7117   -0.4551    2.3614    5.6978
##
##    Coefficients:
##                      Estimate   Std. Error   t value     Pr(>|t|)
##    (Intercept)       -39.9197     11.8960    -3.356      0.00375   **
##    Air.Flow            0.7156      0.1349     5.307      5.8e-05   ***
##    Water.Temp          1.2953      0.3680     3.520      0.00263   **
##    Acid.Conc.         -0.1521      0.1563    -0.973      0.34405
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'0.1 ' ' 1
##
## Residual standard error: 3.243 on 17 degrees of freedom
## Multiple R-squared:  0.9136,        Adjusted R-squared:  0.8983
## F-statistic:  59.9 on 3 and 17 DF,    p-value: 3.016e-09
```

Direct your attention to the coefficient summary, and you will see that two of the predictors appear to be significant (air flow and water temperature) and one predictor is non-significant (acid concentration), with an overall unadjusted R^2 = .91. We take this result to indicate that air flow and water temperature are highly predictive of the outcome variable, but we are left in the dark about acid concentration's relationship to the outcome variable. We do not know if there is evidence for excluding acid concentration from the model or if it is simply a noisy measure that did not reach significance.

BAYESIAN LINEAR REGRESSION

Bayesian regression and model comparison allows us to look at the predictive success of all possible models containing combinations of these variables. Since we have 3 predictors it won't get too unwieldy to look at all combinations. See http://bayesfactorpcl.r-forge.r-project.org/#regression for R commands that help you handle large numbers of variable combinations. It also has functions for many other kinds of Bayesian analyses.

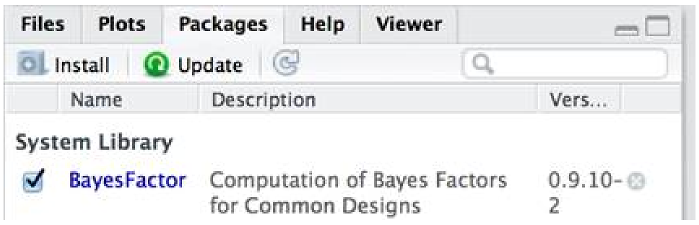
This method is not a built-in R package, so we'll have to download and install it. Thankfully, R Studio makes this incredibly simple. Type:

```
install.packages("BayesFactor")
```

R Studio uses this command to automatically look-up and install packages. Installing a package does not automatically mean it is ready for use. To be able to use the BayesFactor package type:

```
library("BayesFactor")
```

Now your BayesFactor package is loaded and ready for use. If you are using R Studio you can confirm this by looking in the bottom-right panel that the package BayesFactor has a checked box next to it.



	Name	Description	Vers...
System Library			
✓	BayesFactor	Computation of Bayes Factors for Common Designs	0.9.10-2

Figure 2. A screengrab of the bottom-right panel in R Studio. The BayesFactor package box is ticked, indicating that it is ready for use.

Using the BayesFactor regression function is just like using the lm() function. Type:

```
bayesianLR = regressionBF(stack.loss ~ ., data=stackloss)
```

As before, this creates an object in R that contains the information for the Bayesian regression. Now type:

```
bayesianLR
```

```
## Bayes factor analysis
## --------------
## [1] Air.Flow      : 1726839   ±0%
## [2] Water.Temp    : 45637.38  ±0%
## [3] Acid.Conc.    : 1.326714  ±0%
## [4] Air.Flow + Water.Temp                 : 17687511  ±0%
## [5] Air.Flow + Acid.Conc.                 : 295886    ±0%
## [6] Water.Temp + Acid.Conc.               : 8911.447  ±0%
## [7] Air.Flow + Water.Temp + Acid.Conc.    : 3471076   ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

Note the difference here, in that we don't need to use the summary() function. This gives us an output of all model combinations and their predictive success relative to an intercept-only (null) model. The numbers to the right of each model combination indicate how many times better that model accounts for the data than the null model. As is clear, every possible model (except a model with only acid concentration as a predictor) predicts the data much, much better than a null model.

R can do some nice cleaning up for us, so let's look at our data in order of highest to lowest Bayes factor relative to the null model. Type:

```
head(bayesianLR, n = 7)
```

```
## Bayes factor analysis
## --------------
## [1] Air.Flow + Water.Temp                 : 17687511  ±0%
## [2] Air.Flow + Water.Temp + Acid.Conc.    : 3471076   ±0%
## [3] Air.Flow                              : 1726839   ±0%
## [4] Air.Flow + Acid.Conc.                 : 295886    ±0%
## [5] Water.Temp                            : 45637.38  ±0%
## [6] Water.Temp + Acid.Conc.               : 8911.447  ±0%
## [7] Acid.Conc.                            : 1.326714  ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

The function head() takes the list of models and organizes it by the best performing and descending to worst performing. The n = 7 indicates how many models to show, starting from the top of the list. The default is 6, so if we want to show all of our combinations we have to manually enter that into the function. A similar function is tail() that lists the worst performing models.

We can visualize these comparisons by using the plot() command. This will display a plot in the bottom-right hand corner of R Studio. Type:

```
plot(bayesianLR)
```

This plot may be tiny and hard to read at first, so hit that “zoom” button above the plot panel in R studio to open it in a new window that can be expanded to better show the graph.

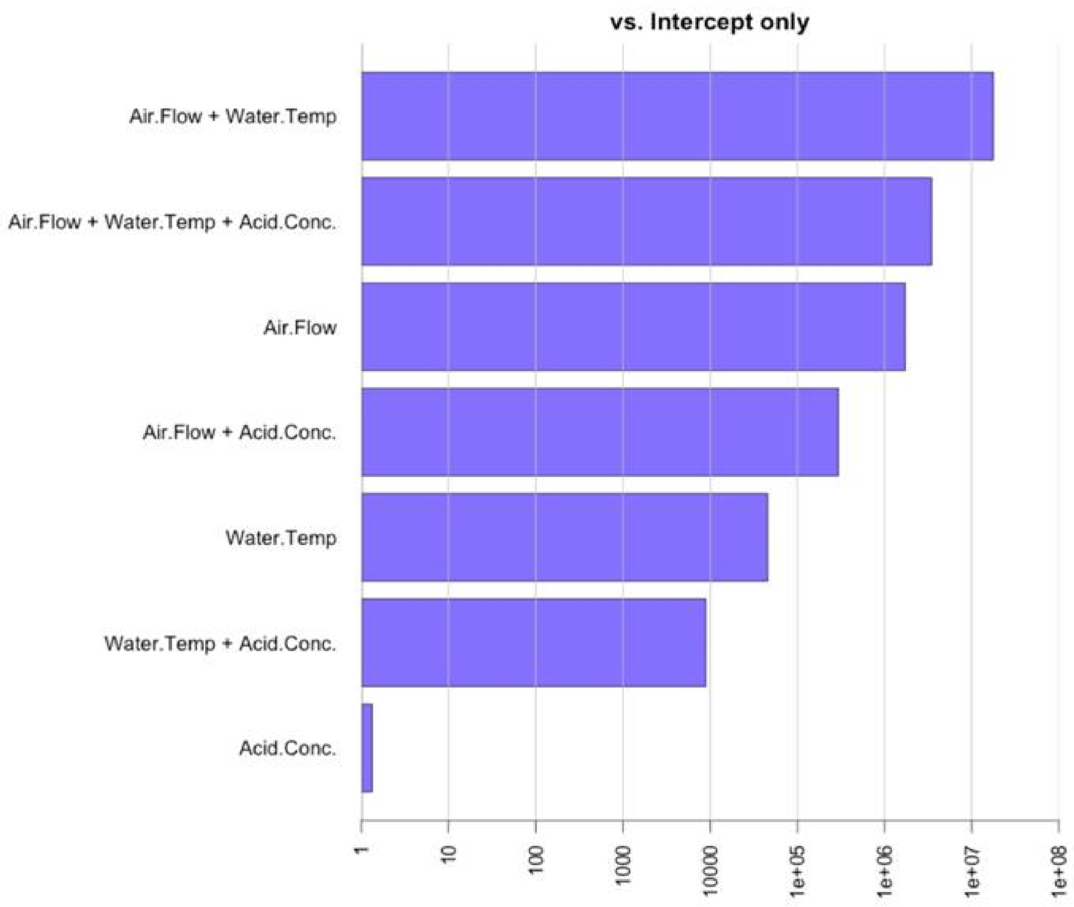


Figure 3. All possible combinations of predictors in the stackloss dataset. All models are preferred over an intercept-only null model.

TAKING STOCK SO FAR

Right now we have a lot of information available to us. We have the result of a conventional linear regression, the result of a Bayesian linear regression, and we know how use R to see which models perform the best when compared to a null model.

Our Bayesian regression indicates that the best fitting model is one that takes into account air flow and water temperature as predictors, with Bayes factor vs a null model = 17,687,511. This means that by including these two variables as predictors we can account for the data roughly seventeen-million to eighteen-million times better than the model that has no predictor variables. The next best model is the full model including air flow, water temperature, and acid concentration, with Bayes factor vs a null model = 3,471,076. This means that by including all predictor variables we account for the data roughly three-and-a-half-million times better than a model having no predictor variables.

Since every Bayes factor reported by this output is a ratio of the model’s predictive success relative to a null model, that means we can compare any two models by simply dividing one by the other (to cancel out the null model denominators). A useful comparison is see how much better the best model is than the rest of the models under consideration. Type:

```
bayesianLR2 = head(bayesianLR, n=7) / max(bayesianLR)

bayesianLR2
```

```
 ## Bayes factor analysis
 ## --------------
 ##    [1]    Air.Flow + Water.Temp                : 1              ±0%
 ##    [2]    Air.Flow + Water.Temp + Acid.Conc.   : 0.1962444      ±0%
 ##    [3]    Air.Flow                             : 0.0976304      ±0%
 ##    [4]    Air.Flow + Acid.Conc.                : 0.01672852     ±0%
 ##    [5]    Water.Temp                           : 0.002580204    ±0%
 ##    [6]    Water.Temp + Acid.Conc.              : 0.0005038271   ±0%
 ##    [7]    Acid.Conc.                           : 7.500854e-08   ±0%
 ##
 ## Against denominator:
 ##   stack.loss ~ Air.Flow + Water.Temp
 ## ---
 ## Bayes factor type: BFlinearModel, JZS
```

Note the difference here in the Against denominator spot; it has updated from intercept only model to the best fitting model (Air.Flow + Water.Temp).

Since we only have 7 models, we can easily look at how much better the best model is than every other model. If we had many more possible predictor combinations we could limit ourselves to comparing only the best or worst models. As is clear from the output above, the best model (air flow and water temperature) is about 1/.19 or ~ 5 times better than the next best model (full model). This means that the inclusion of the acid concentration predictor to the model does not add enough explanatory power to overcome the natural Bayesian penalties for increasing model complexity.

Additionally, the best model is 1/.09 or ~ 10 times better than the model with only air flow as a predictor. This means that the water temperature variable adds considerable explanatory power, enough to overcome the increased complexity penalty; thus, we have fairly strong evidence indicating that the water temperature variable should be included as a predictor in the model.

We can visualize these comparisons the same way we did before. Type:

```
plot(bayesianLR2)
```

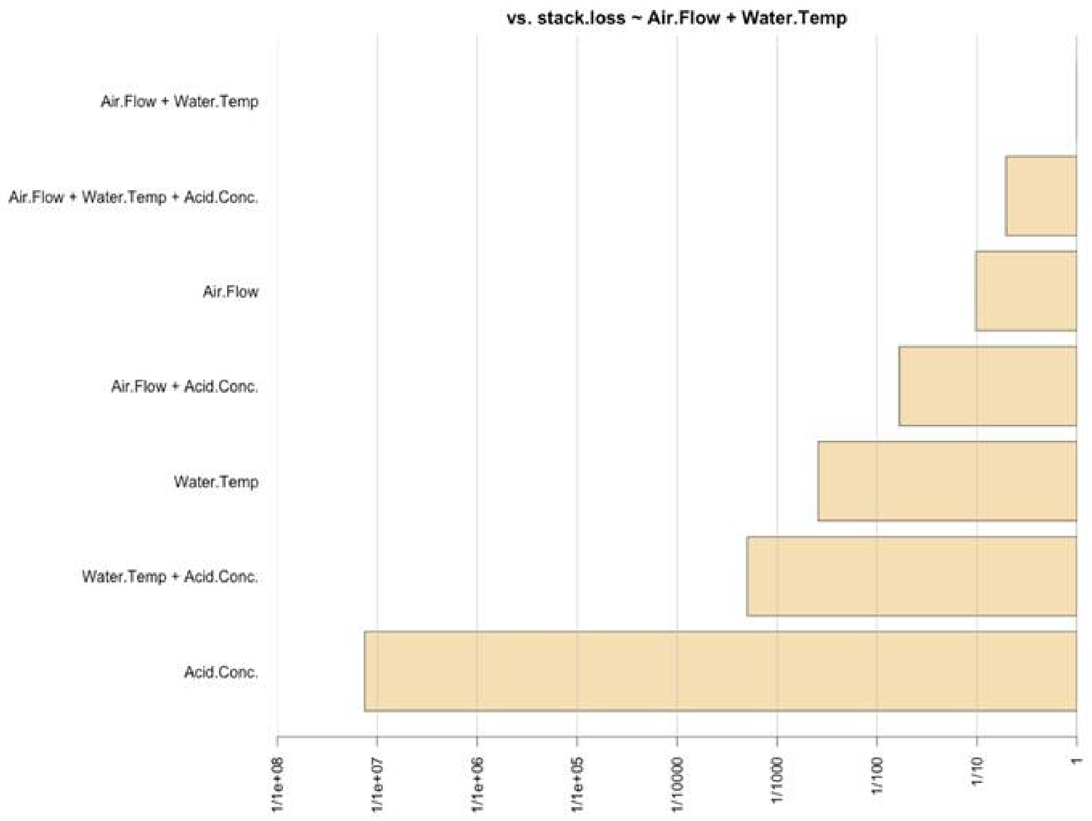


Figure 4. All models compared to the best fitting model. Blue bars (none pictured) would indicate a model that fits better than the best fitting model. Yellow bars indicate models that fit worse than the best fitting model.

If it is more relevant to the problem to compare all models to one particular model, say, the full model with all predictors, we can do that too. To do this, type:

```
bayesianLR3 = head(bayesianLR, n = 7)/ bayesianLR["Air.Flow + Water.Temp + Acid.Conc.”]

bayesianLR3

plot(bayesianLR3)
```

```
## Bayes factor analysis
## --------------
##   [1]   Air.Flow + Water.Temp               :   5.095686      ±0%
##   [2]   Air.Flow + Water.Temp + Acid.Conc.  :   1             ±0%
##   [3]   Air.Flow                            :   .4974939      ±0%
##   [4]   Air.Flow + Acid.Conc.               :   0.08524331    ±0%
##   [5]   Water.Temp                          :   0.01314791    ±0%
##   [6]   Water.Temp + Acid.Conc.             :   0.002567344   ±0%
##   [7]   Acid.Conc.                          :   3.8222e-07    ±0%
##
## Against denominator:
```

```
##   stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.

## ---

## Bayes factor type: BFlinearModel, JZS
```

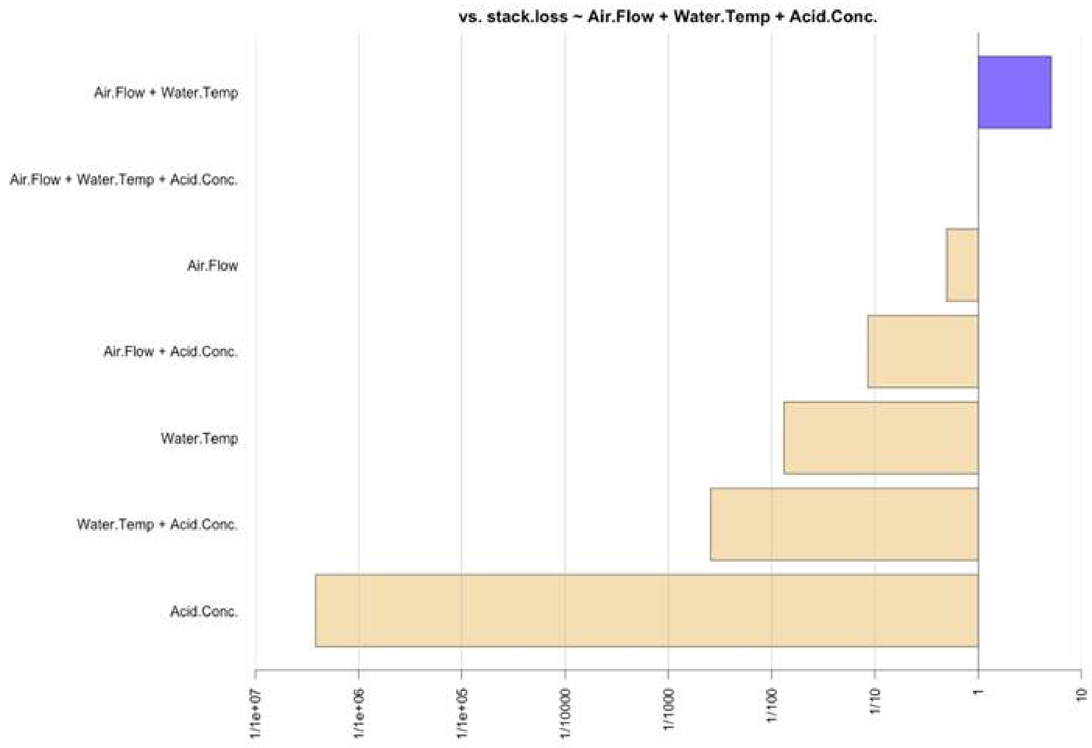


Figure 5. The full model with all predictors compared to all other models. Blue bars indicate a model that fits better than the full model. Yellow bars indicate models that fit worse than the full model.

If we wanted to focus in and see how dropping a predictor from the full model (top-down) impacts the fit, we could type:

```
bayesianLR4 = regressionBF(stack.loss ~ ., data = stackloss, whichModels = “top")

plot(bayesianLR4)
```

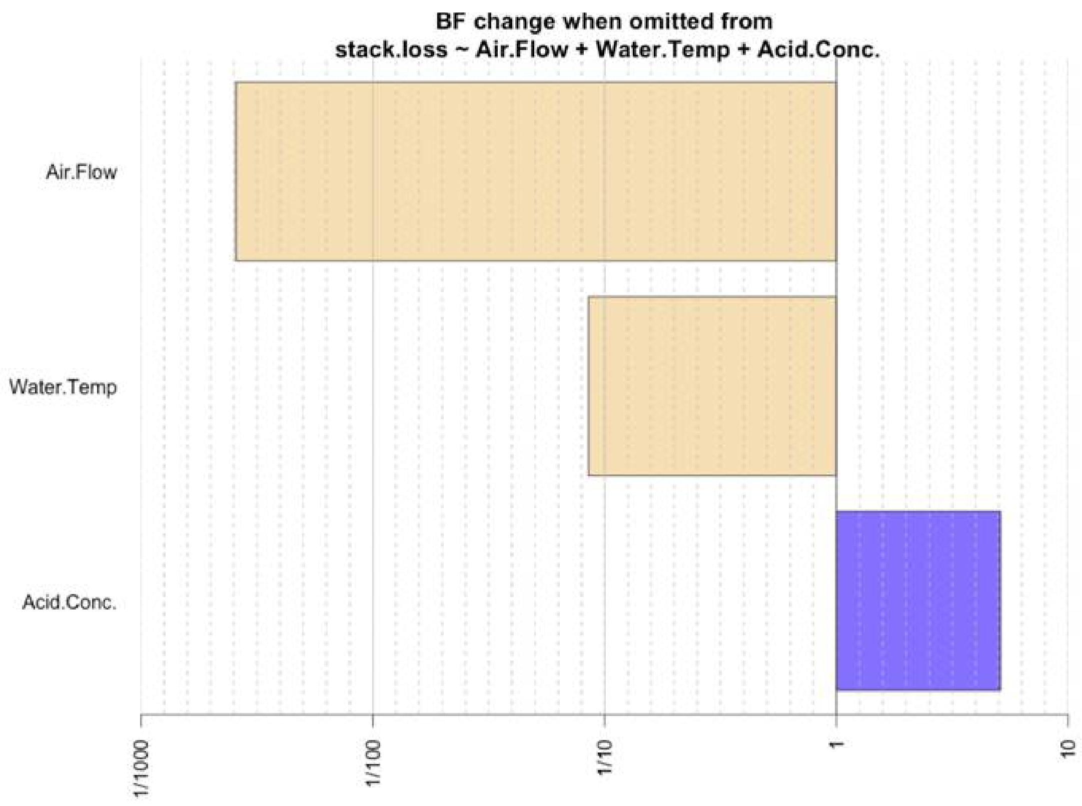


Figure 6. Top-down model comparisons. Each predictor is dropped from the full model one at a time. Blue bars indicate that the model fits better when that predictor is dropped. Yellow bars indicate that the model fits worse when that predictor is dropped.

This shows us that dropping air flow or water temperature from our full model gives us considerably worse fit, whereas dropping acid concentration gives us a slightly better fit.

Alternatively, we could progressively add predictors to the model until we reach the full model, and see how the fit is affected (bottom-up, starting with the worst fitting predictor). Type:

```
bayesianLR5 = regressionBF(stack.loss ~ ., data = stackloss, whichModels = "bottom")

bayesianLR5

plot(bayesianLR5)
```

This tells much the same story as the top-down model comparison so I won't include the output or graph, but it gives us converging evidence that the acid concentration variable should not be included in our models; acid concentration adds too much complexity and not enough explanatory power any way you slice it.

POSTERIOR ESTIMATION

Sometimes the end-goal of an analysis is not to simply compare different models, but to estimate the parameters to the best we can. It would not be feasible to report parameter estimates for every single possible model, especially in situations where more than 3 or 4 predictors are present. A more reasonable approach would be to pick 1 or 2 models to report that perform particularly well or are theoretically relevant. For example, researchers using the stackloss dataset could report the slope estimates for the full model and for the best performing model and see note any striking differences in slope estimates. Let's continue with our example, and find posterior slope estimates for our best fitting model, which includes air flow and water temperature as predictors.

The BayesFactor package has an easy way to obtain these estimates, but there are a few steps to getting to the posterior distribution. First we have to create a simplified version of our bayesianLR object that only contains the model of interest. To do this, we type:

```
bestModelBF = lmBF(stack.loss ~ Air.Flow + Water.Temp, data = stackloss)

bestModelBF

 ## Bayes factor analysis
 ## --------------
 ##     [1]     Air.Flow + Water.Temp    :     17687511     ±0%
 ##
 ## Against denominator:
 ##   Intercept only
 ## ---
 ## Bayes factor type: BFlinearModel, JZS
```

This sets up a single model comparison that stands alone from the rest. It is possible to set up single models like this to do focused comparisons using the / function. Next we need to sample from this model to obtain our slope estimates. We do this with the posterior() function. Type:

```
bestModelPosterior = posterior(bestModelBF, iterations = 10000)

summary(bestModelPosterior)

 ## Iterations = 1:10000
 ## Thinning interval = 1
 ## Number of chains = 1
 ## Sample size per chain = 10000
 ##
 ## 1. Empirical mean and standard deviation for each variable,
 ##    plus standard error of the mean:
 ##
 ##                      Mean        SD    Naive SE   Time-Series
 ##                                                            SE
 ##    Air.Flow        0.6629    0.1397    0.001397     0.001461
 ##    Water.Temp      1.2732    0.4025    0.004025     0.004025
 ##    sig2           12.5895    4.9039    0.049039     0.061206
 ##    g              51.6259  808.2256    8.082256     8.491609
 ##
 ## 2. Quantiles for each variable:
```

```
##
##                  2.5%      25%       50%       75%      97.5%
##   Air.Flow      0.3826    0.5723    0.6637    0.7538    0.9428
##   Water.Temp    0.4657    1.0194    1.2755    1.5290    2.0779
##   sig2          6.2417    9.2020   11.5508   14.8428   24.8592
##   g             0.8461    2.6476    5.5925   13.4549  167.8925
```

Note that it is important to use the summary() function here, and not simply call the raw bestPosteriorModel object. If we did that we would end up filling up our command console with a 1,000 row list that has all of the raw simulated samples. Don't do that.

This summary shows us the coefficient estimates for each of the predictors in the best fitting model, along with different quantiles of their respective distributions. If we wanted to report our best posterior estimate for each predictor with a credible interval we could simply report the coefficient estimate along with the 2.5% and 97.5% quantiles. For example, air flow predictor's best coefficient estimate is .66 with a 95% credible interval ranging from .38 to .94. We can interpret this credible interval to mean that, conditional on the model, there is a 95% chance that the true coefficient is between .38 and .94.

When reviewing the posterior coefficient estimates it can be helpful to recall the estimates from the conventional linear regression. I've provided an abbreviated output below for easy comparison:

```
conventionalLR = lm(stack.loss ~ ., data = stackloss)

summary(conventionalLR)

##   Coefficients:
##                  Estimate   Std. Error   t value    Pr(>|t|)
##   (Intercept)   -39.9197     11.8960     -3.356    0.00375  **
##   Air.Flow        0.7156      0.1349      5.307    5.8e-05  ***
##   Water.Temp      1.2953      0.3680      3.520    0.00263  **
##   Acid.Conc.     -0.1521      0.1563     -0.973    0.34405
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'0.1 ' ' 1
```

For this data set, the coefficient estimates are pretty close between the Bayesian model and linear model because we are only working with a few predictors. This will not always be the case for situations with more predictors, and it is not necessarily the case in general. We can construct rough 95% confidence intervals for our coefficients by multiplying the standard error from the output above by 2, and then adding and subtracting that value from the coefficient estimate.

Air flow in the above output has a standard error of approximately .135, so the confidence interval would span the range of .72 +/- .27, or from .45 to .99. This range is quite similar to the one we obtained from calculating a posterior (.38 — .94). Confidence intervals are interpreted to mean that if we were to repeat the experiment very many times, 95% of our intervals would capture the true coefficient value. This particular interval that we have constructed says nothing about the probability we have captured the true coefficient value. It merely says that 95% of intervals constructed using this same procedure would contain the true value.

The interpretation of a confidence interval is a far cry from the interpretation of a Bayesian credible interval (i.e., 95% certainty the true value is within the interval .38-.94), and highlights one of the benefits of Bayesian inference we saw earlier: Bayesian inference provides directly interpretable answers to our questions. In this case, the question is, “What should I believe is the true value of the air flow coefficient?” and the answer is, “You can believe with 95% certainty that the true value lies between .38 and .94.” Further information can be gleaned from graphing and inspecting the posterior distribution’s shape, but that is not covered in this guide.

WRAP-UP

I started this guide with a problem that gives conventional statistics extreme difficulty but is strikingly simple for Bayesian analysis: Conventional statistics do not allow researchers to make claims about one of the models being tested (sometimes the only model). This inferential asymmetry is toxic to interpreting research results.

Bayes factors solve the problem of inferential asymmetry by treating all models equally, and have many other benefits: 1) No penalty for optional stopping or multiple comparisons. Collect data until you feel like stopping or run out of money and make as many model comparisons as you like; 2) Bayes factors give directly interpretable outputs. A Bayes factor means the same thing whether n is 10 or 10,000, and whether we compared 2 or 20 models. A credible interval ranging from .38 to .94 means that we should believe with 95% certainty that the true value lies in that range. 3) Prior probability distributions allow researchers to intimately connect their theories to empirical data. Prior probability distributions are what give Bayesian analysis its power to make predictions and test theories (Rouder, 2015). This was not elaborated on here but see Dienes (2008, 2014) and Vanpaemel (2010) for more discussion.

These benefits can be seen in the context of regression analysis. We were able to compare many models at once to assess relative fit, as well as make pointed comparisons between particular models. We were able to calculate the best estimates of our coefficients by simulating a posterior distribution and directly interpret our credible intervals. Now, with a little practice you should be able to compute, report, and interpret Bayes factors for your real research projects. Give it a try and you’ll never look back.

References

Berger, J. (2006). The case for objective Bayesian analysis. *Bayesian Analysis, 1*(3), 385-402.

Christen, J. A. (2006). Stop Using ‘subjective’to Refer to Bayesian Analyses (Comment on Articles by Berger and by Goldstein). *Bayesian Analysis, 1*(3), 421-422.

Dienes, Z. (2008).*Understanding psychology as a science: An introduction to scientific and statistical inference*. Palgrave Macmillan.

Dienes, Z. (2011). Bayesian versus orthodox statistics: Which side are you on?. *Perspectives on Psychological Science, 6*(3), 274-290.

Dienes, Z. (2014). Using Bayes to get the most out of non-significant results. *Frontiers in Psycholology, 5*. doi: 10.3389/fpsyg.2014.00781

Edwards, A. W. F. (1972). *Likelihood*. London, UK: Cambridge University Press.

Edwards, W., Lindman, H., & Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychological Review, 70*(3), 193-242.

Fienberg, S. E. (2006). When did Bayesian inference become" Bayesian"?. *Bayesian Analysis, 1*(1), 1-40.

Fisher, R. (1955). Statistical methods and scientific induction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 69-78.

Fisher, R. A. (1966). *The design of experiments (8th edn.).* Oliver and Boyd.

Goldstein, M. (2006). Subjective Bayesian analysis: principles and practice. *Bayesian Analysis, 1*(3), 403-420.

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association, 90*(430), 773-795.

Lee, M. D., & Wagenmakers, E. J. (2005). Bayesian statistical inference in psychology: comment on Trafimow (2003). *Psychological Review, 112*, 662-668.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., & Berger, J. O. (2008).Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association, 103*(481), 410-423.

Lindley, D. V. (2000). The philosophy of statistics. *Journal of the Royal Statistical Society: Series D (The Statistician), 49*(3), 293-337.

Morey, R. D. (2014, February 9). What is a Bayes factor? [Web Log Post]. Retrieve from http://bayesfactor.blogspot.co.uk/2014/02/the-bayesfactor-package-this-blog-is.html

Morey, R. D., & Rouder, J. N. (2011). Bayes factor approaches for testing interval null hypotheses. *Psychological Methods, 16*(4), 406-419.

Morey, R. D., Rouder, J.N., & Jamil, T. (2015). BayesFactor: Computation of Bayes factors for common designs. R package version 0.9.9.

Morey, R. D., & Wagenmakers, E. J. (2014). Simple relation between Bayesian order-restricted and point-null hypothesis tests. *Statistics & Probability Letters, 92*, 121-124.

Rouder, J. N. (2014). Optional stopping: No problem for Bayesians. *Psychonomic Bulletin & Review, 21*(2), 301-308.

Rouder, J. N. (2015, March 1). To Beware or To Embrace The Prior. [Web Log Post]. Retrieved from http://bayesfactor.blogspot.co.uk/2015/03/to-beware-or-to-embrace-prior.html

Rouder, J. N., & Morey, R. D. (2012). Default Bayes factors for model selection in regression. *Multivariate Behavioral Research, 47*(6), 877-903.

Royall, R. (1997). *Statistical evidence: A likelihood paradigm* (Vol. 71). CRC press.

Sanborn, A. N., & Hills, T. T. (2014). The frequentist implications of optional stopping on Bayesian hypothesis tests. *Psychonomic Bulletin & Review, 21*(2), 283-300.

Savage, L. J. (1954). *The foundations of statistics*. New York, NY: John Wiley & Sons Inc.

Vanpaemel, W. (2010). Prior sensitivity in theory testing: An apologia for the Bayes factor. *Journal of Mathematical Psychology, 54*(6), 491-498.

Wagenmakers, E. J. (2007). A practical solution to the pervasive problems of p-values. *Psychonomic Bulletin & Review, 14*(5), 779-804.

[1] The term “prior” here simply means it was not informed by the data. It does not necessarily mean it was specified prior to data collection.

[2] A null interval can be specified instead of a point null if relevant to the problem, see Morey

and Rouder, 2011 for details.

[3] It is true that these factors can influence the sampling distribution of the Bayes factor (Sanborn & Hills, 2014), however this fact does not influence the interpretation of the Bayes factor (Rouder, 2014).