***Bayes factor worksheet***

*Don’t worry if you can’t complete everything.* Some of the material here is adapted from Chapter 11 of my book:

Baguley, T. (2012). *Serious stats: A guide to advanced statistics for the behavioral sciences*. Palgrave Macmillan.

Alternatively there is similar coverage in:

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

However, much of the material is inspired by work by Rouder, Morey and colleagues – notably the BayesFactor package and support material for it:

Morey, R. D., & Rouder, J. N. (2015). *BayesFactor: Computation of Bayes factors for common designs*. R package version 0.9.11. <http://CRAN.R-project.org/package=BayesFactor>

**Installing the BayesFactor package**

Installing a new package in R Studio is fairly easy. First locate the “Packages” tab (usually in the bottom right window where the “Plots” tab is). Then select “Install” and start typing the name of the package you want). Select this package from the list and click the “Install” button. You may be prompted to restart R as part of the process (so save any work in progress first). Once installed, you can load the package in the usual ways (e.g., from its name in the “Packages” window of R Studio).

For support with BayesFactor take a look at the online manual and blog:

<http://bayesfactorpcl.r-forge.r-project.org>

<http://bayesfactor.blogspot.co.uk>

**Reprise: Effect of a single exposure to a US flag on voting intention**

|  |
| --- |
| **Q1**   1. Use the unit.prior.Bf.2s() function to reproduce the “*Single exposure to a US flag example*” Bayes factor. First download the functions from   <http://www2.ntupsychology.net/seriousstats/SeriousStatsAllfunctions.txt> [[1]](#footnote-1)  The summary statistics for the effect of the flag on voting intention are: *M*1-*M*2 = 0.142, *t*(181) = 2.02, SE = 0.0703, *p* < .04*.* The call to the function takes the form:  unit.prior.Bf.2s(t, n1, n2, scale.factor=1)  *Run the following call and note down the Bayes factor:*  unit.prior.Bf.2s(t=2.02, n1=91.5, n2=91.5, scale.factor=1)   1. The unit-information prior requires the scale factor is set to *r* = 1. A more general scaled-information prior can be obtained by increasing or decreasing *r*.   *What do you think will happen to the Bayes factor if you increase r?*  *What do you think will happen to the Bayes factor if you decrease r?*  Re-run the call with r = 0.5 and r = 2. *Did the Bayes factor change in the way you expected?*  **c)** Now run it again with r = 0.05 or 0.01 (or both). *Did the results surprise you in any way? Why do think the Bayes factor behaved in this way as you went from r = 2 down to r = 1, r = 0.5 to r = 0.05 or r = 0.01)?* |

|  |
| --- |
| **Q2**  **a)** Use the ttest.tstat() function to get a JZS Bayes factor from the BayesFactor package. (The package is designed to work from raw data so this bit is not very elegant). The call to the function takes the form:  log.bf <- ttest.tstat(t, n1, n2, rscale='medium')[['bf']]  exp(log.bf)  … or (if you prefer) it in one step:  exp(ttest.tstat(t, n1, n2, rscale='medium')[['bf']])  *Run the following call:*  exp(ttest.tstat(t=2.02, n1=91.5, n2=91.5, rscale='medium')[['bf']])  The ttest.tstat() is an internal function not normally called directly by users of the package. *Why do you think it returns the (natural) logarithm of the Bayes factor rather than the Bayes factor itself?*  **b)** Now try and reproduce the test of the Bayes factor test of the directional hypothesis. The call to the function is identical but requires an extra argument:  nullInterval=c(0,Inf)  e.g.,  exp(ttest.tstat(t=2.02, n1=91.5, n2=91.5, rscale='medium', nullInterval=c(0,Inf))[['bf']])  **c)** *What do you think will happen if the argument were* nullInterval=c(-Inf,0) *?* Try it. *Does the output match what you expected?*  **d)** *What do you think will happen if the argument were* nullInterval=c(-Inf,Inf) *?* Try it. *Does the output match what you expected?*  **f)** The  **-**Inf or Inf (the R primitive for inifinity) term in the nullInterval argument leaves that half of the prior density unconstrained, while 0 places no weight on that half of the prior density. This is perhaps an implausible because it is comparing a one-sided hypothesis (that ** is greater than/less than zero to a point null hypothesis – that ** = 0 exactly).  An alternative approach (if the point null is implausible) is to compare the hypothesis **hypothesis **< 0. This is relatively simple because odds ratios like Bayes factors are multiplicative. Thus the ratio of the Bayes factor for **> 0 to the ratio of the Bayes factor **< 0 is the Bayes factor testing these competing hypotheses.  *What is this Bayes factor?* |

1. You can also use load the functions directly into R using the source() function:

   source('http://www2.ntupsychology.net/seriousstats/SeriousStatsAllfunctions.txt') [↑](#footnote-ref-1)