Statistical Methods for Accelerometer Data

Phuong Tran

Supervised by Dr Chris Drovandi

Queensland University of Technology

AMSI
AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Australian Government
Department of Education and Training

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

1 Introduction

As an older branch of statistics with a soaring list of advancements in the 21st century, Bayesian statistical methods have been widely used in many fields. The ability to provide clear, coherent and direct inferences for unknown parameters by making use of all of the available data of Bayesian methods is highly praised by many practitioners. Bayesian statistics also offer several major advantages such as incorporating prior knowledge into the analysis, quantifying uncertainties in the model and the parameter values, as well as allowing great flexibility in modelling which is incredibly useful in complex models (Kruschke 2015). However, applying Bayesian methods to analysing data can be quite challenging for health practitioners. Due to the development of application tools implementing Bayesian methods, it is now feasible for non-statisticians to learn and make use of the numerous advantages Bayesian modelling has to offer (Depaoli, Clifton, and Cobb 2016). This research looks at two particular software packages, JAGS and Stan, developed specifically for the analysis of Bayesian hierarchical models and compares their performance on two datasets. Through this report, readers will be introduced to the concept of Bayesian inference, Markov Chain Monte Carlo methods as well as the use of two software packages that implement these methods.

2 Bayesian inference

2.1 Mathematical framework

The mathematical foundation of Bayesian statistics is a theorem invented by Thomas Bayes, an English mathematician and a Presbyterian minister in the 18^{th} century (McGrayne 2011). Bayes' theorem shows the relationship between the marginal probabilities and the conditional probabilities when data is taken into account (Kruschke 2015). The theorem underlies all the principles of inferences and decisions in Bayesian statistics. Bayes' theorem is expressed in equation (1).

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{1}$$

In equation (1), θ denotes the unobserved quantity, which is also the parameter of interest and y is the observed data. The posterior density, which is the probability of the unknown parameter θ given the known data y, is $p(\theta|y)$. The joint probability distribution $p(\theta, y)$ for θ and y can be expressed as the product of the prior $p(\theta)$ and the likelihood $p(y|\theta)$ of the known data y being observed given the value of θ. $p(y) = \int_\theta p(y|\theta)p(\theta)d\theta$ denotes the marginal probabilities of data y over all the possible values of θ. In Bayesian statistics, probability models are applied in the process of data analysis which allows uncertainty in inferences to be measured with probability. (Gelman 2005)

2.2 Challenge to implement Bayesian data analysis

The aim in Bayesian data analysis is to be able to make inferences from the available data about the unknown parameter θ using the the posterior distribution $p(\theta|y)$. A substantial challenge to analytically determining the posterior distribution from Bayes' rule is to compute the marginal likelihood $p(y)$. In order to illustrate the issue for the case when the parameter of interest θ is continuous, equation (1) can be expanded into equation (2).

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta} \tag{2}$$

The integral in equation (2) can be impossible to evaluate analytically for non-trivial models and deterministic numerical integration methods do not perform well when θ is high-dimensional. In some cases, the issue can be addressed with a conjugate-prior approach where models are restricted to simple likelihood functions with corresponding formulas for the prior distributions so that a tractable integral is produced. Nevertheless, the conjugate-prior approach does not work in realistically complex models in practice. Many approaches of approximation have been developed to estimate the posterior distribution instead of analytical approaches. One powerful approximation approach, which involves random sampling of the posterior distribution, is regarded as the class of Markov Chain Monte Carlo (MCMC) methods. These methods include numerous algorithms that can generate representative parameter values from the posterior distribution of complex models without evaluating the integral in equation (2). The statistical tools that apply these algorithms have played a major role in making Bayesian statistics accessible for users in practice. (Kruschke 2015)

3 Approximation of the posterior distribution

In order to utilise the software that apply MCMC methods, it is helpful to understand the essential idea of these methods. MCMC appoaches aim to accurately estimate the posterior distribution, without evaluating the integral, by generating a large representative sample of the distributions using random sampling.

MCMC methods explore the parameter space by generating a Markov chain whose limiting distribution is given by the posterior distribution to be approximated. MCMC methods generate correlated samples from the posterior distribution, which can subsequently be used to estimate quantities with respect to the posterior distribution such as expected values. Two important elements of MCMC methods in practice are the burn-in period and the effective sample size. The burn-in period is required to discard the initial samples generated such that the Markov chain has not yet converged to the posterior distribution. It is crucial that the burn-in period for the MCMC chain is sufficiently long so that the chain can reach convergence and the representative samples are only taken from the posterior distribution. The quality of the MCMC sample can be assessed by the effective sample size (ESS), which is the number of effectively independent samples obtained from the MCMC sampling output.The ESS takes into account that MCMC generates only correlated samples from the posterior distribution. A low ESS, relative to the number of MCMC samples, indicates the chain is poorly mixed whereas a high ESS, particularly when it is close to the total sample size, shows good mixing and that the samples adequately represent the posterior distribution.

There are several systems that apply the MCMC methods to estimate the posterior distributions. The two application tools examined in this project were JAGS and Stan. While the two systems possess some fundamental differences in their underlying algorithms, which will be discussed in the coming sections, they both allow the implementation process of the MCMC methods to be relatively simple. This makes the two software packages suitable for analysing data in complex hierarchical models. When using these two statistical tools, users are only required to provide the data, specify the prior distributions $p(\theta)$ and the statistical model believed to have generated data y. The algorithms will then produce a large random sample of θ from the posterior distribution $p(\theta|y)$ which allows users to access all the properties of the posterior distribution such as the mean or credible intervals of the distribution so that they can make inferences about the parameters of interest. (Kruschke 2015)

4 Statistical packages: JAGS and Stan

The two chosen statistical tools to examine in this project are JAGS and Stan. Both software packages apply the principle of MCMC methods to directly sample from the posterior distribution. One common advantage of JAGS and Stan is that they can work on various operating systems, such as Windows, Mac OX, Unix and Linux, and they can be used in conjunction with popular software such as MATLAB, R, Python for summarising and visualising the output of the MCMC procedure.. For this project, JAGS and Stan were implemented in the R environment as they were designed to work closely with the R language. (Depaoli, Clifton, and Cobb 2016)

This section provides a brief discussion on the underlying mechanisms and the components that are essential to understand to use JAGS and Stan.

4.1 JAGS

JAGS is a statistical tool applying the MCMC methods to sample directly from probability distributions for complex hierarchical models. JAGS is short for Just Another Gibbs Sampler (Plummer 2015). When Gibbs sampling is implemented, the posterior samples for each parameter are generated from the conditional distribution with the remaining parameters fixed to their current values. The process is carried out for all the parameters, one at a time, and repeated until convergence is reached for all parameters. (S. Geman and D. Geman 1984; Yildirim 2012) Moreover, JAGS also implements other sampling strategies when sampling from the conditionals cannot be done directly. (Plummer 2015)

The four key elements in the implementation process of JAGS include defining the data, specifying the model, compiling the model and initialising the model. Users are only required to define the data and specify the model in the format required by JAGS. The specifications can be done in text files, script files, strings or functions. They also have the option to initialise the model by specifying initial values for the MCMC chains and choosing which parameters to supply initial values for. However, the model will be converted and compiled by JAGS. Any errors in the model syntax will be identified at this compilation stage. Subsequent to the compilation process, JAGS goes through an adaptation phase in which JAGS will decide the most appropriate samplers for each parameter. The sampling process in JAGS takes place after this adaptation phase. The users can specify the total number of iterations for sampling as well as the number of iterations to discard (burn-in period). JAGS allows user to manually set monitors on parameters to record their simulated values in order to carry out diagnostic tests and assess summary statistics of the posterior distributions of those parameters. (Depaoli, Clifton, and Cobb 2016)

In the R environment, there are a number of packages to call JAGS such as rjags, r2jags, runjags (Plummer 2016; Su and Yajima 2015; Denwood 2016). Several packages have also been developed for the convergence diagnostics of the MCMC sampling outcomes. In this project, the package coda (Plummer et al. 2006) was used to assess convergence of the MCMC chains. Several resources with instructions and examples on how to implement these packages are easily found, which makes JAGS a useful tool to for beginners to implement Bayesian statistics.

4.2 Stan

Similar to JAGS, Stan is a software package that generates representative samples directly from the posterior distributions of continuous parameters for hierarchical models. The name Stan stands for Sampling Through Adaptive Neighborhoods (Kruschke 2015). Stan applies a different method than JAGS to generate the Monte Carlo steps which can be more effective than the samplers in JAGS for large complex models. The method implemented in Stan is called Hamiltonian Monte Carlo (HMC).

Even though Stan is not always better or faster than JAGS, Stan uses C++ compiler which allows more programming flexibility. This aspect of Stan and the fact that its underlying mechanism for sampling is the HMC method make it useful for complex models where JAGS may display very poor mixing properties. Stan requires a bit more effort to learn than JAGS. However, Stan's extensive documentation with detailed examples (Stan Development Team 2016) and its convenient platform to communicate directly to the Stan development team about users' issues facilitates the learning process for Stan users very well.

As mentioned earlier, both JAGS and Stan apply MCMC methods which use random walks to explore the parameter space so that direct sampling from the posterior distributions can occur. Each new step from the current position in the walk is determined by a proposed jump, which is sampled randomly from a proposal distribution. The proposed jump is then accepted or rejected based on the relative density of the posterior at the proposed position and the current position. The HMC method in Stan employs the gradient of the log posterior in its sampling algorithm to guide the Monte Carlo steps of the random walks being more tuned towards the posterior distribution (Kruschke 2015; Stan Development Team 2016). This implementation of the derivatives of the log probability function helps Stan to reach convergence to the stationary distribution faster and to be more efficient with subsequent parameter exploration. (Stan Development Team 2016)

The project calls Stan with the rstan package in the R programming environment. Stan is more involved than JAGS in terms of model specification as it requires specific declarations of variables, parameters together with their types and restrictions on their domains. Before sampling can start, the model is translated and compiled into a dynamic shared object. In this phase, Stan would also compute the gradient functions for the Hamiltonian dynamics, which can take some time depending on the complexity of the model (Kruschke 2015). Similar to JAGS, users can specify the total number of iterations, the number of warm up iterations, which is the equivalent of burn-in in JAGS, the parameters to monitor as well as the initial values for parameters. There are also many packages to facilitate plots and diagnostics in Stan such as coda, ggmcmc, Shinystan (Plummer et al. 2006; Fernández-i-Marín 2016; Stan Development Team 2017).

5 Data

The performances of JAGS and Stan were made based on their sampling results on two datasets. A simulated dataset with a small number of data points and a large real dataset on accelerometer data (Trost et al. 2011). This section provides a description of the data used in this project.

5.1 Simulated dataset

Our simulated dataset consists of 120 data points with no missing values. The data follows a linear mixed-effects model with a continuous response variable. The data is treated as if it came from a study where 20 participants were asked to walk on a treadmill. At different time intervals, each participant's concentration level is measured and recorded so that the relationship between the amount of walking time and the level of concentration of a person can be investigated. The model for this dataset can be expressed as equation (3).

$$y_{ij} = \beta_0 + \beta_1 \times x_{ij} + \gamma_i + \epsilon_{ij} \tag{3}$$

In equation (3), a subject, indexed by the i subscript, is measured at observation j to record a score y. Subscript i goes from 0 to 20 and j goes from 1 to 6. The response y_{ij} is the concentration score of subject i at observation j; y_{ij} can be any value between 0 to 100 with 0 being no concentration and 100 being the highest concentration score. The covariate x in this model is treated as the number of minutes the person spent walking on the treadmill. The possible values for x_{ij} are 0, 4, 8, 12, 16, 20. The terms γ_i and ϵ_{ij} represent the subject effect and the error term respectively in each observation. The intercept β_0 and the slope β_1 are the two parameters of interest to be analysed with the two software packages.

5.2 Accelerometer dataset

The accelerometer dataset used in this project consists of just over 35000 observations, after removing all the data points with missing values. The data comes from 222 participants between the age of 5 and 18 years completing 12 different activity trials at four different time points (Drovandi et al. 2015). The three types of trials in this study are sedentary activities (lying down, writing, computer game), lifestyle activities (sweeping, laundry, throw and catch, aerobics, basketball) and ambulatory activities (comfortable walk, brisk walk, brisk treadmill walk, running) (Trost et al. 2011). Four different cut-points were applied on each individual's performance of the 12 activities at each time point. These cut-points with their respective prediction equations are used to predict the type of activities undertaken based on the output of the accelerometer. The response variable in this dataset is a binary variable indicating whether the prediction was correct or not. The aim of the analysis is to determine which of the cut-points are most effective and how this might depend on the activity type and also the age of the participant.

The model we fit to this dataset is a logistic regression mixed effects model with a binary response where age is a continuous variable, cut-points and the type of activities are four-factored and twelve-factored variables respectively. The model considered both main effects and all two-way interactions which resulted in 63 parameters in total. A normal random intercept was also included for each participant.

The dataset was previously analysed for the effect of age (both main and interactions effects) (Drovandi et al. 2015). However, in this project, as our aim is to compare the overall performances of JAGS and Stan, the sampling results for the posterior distributions of all the 63 parameters were compared.

6 Sampling results

Both the JAGS and Stan approaches are used on the simulated dataset and the accelerometer dataset. When implementing JAGS and Stan to sample from the posterior distributions in each dataset, we apply the same model specifications, prior distributions and initial values for the MCMC chain. We also ran the MCMC chains for the same total number of iterations as well as the number of iterations to be discarded. We then compared the performances of the two software packages based on the effective sample size for each parameter, which is the equivalent number of independent samples produced from the MCMC output, which gives correlated samples, and their computational efficiency (the number of effective samples per unit of time).

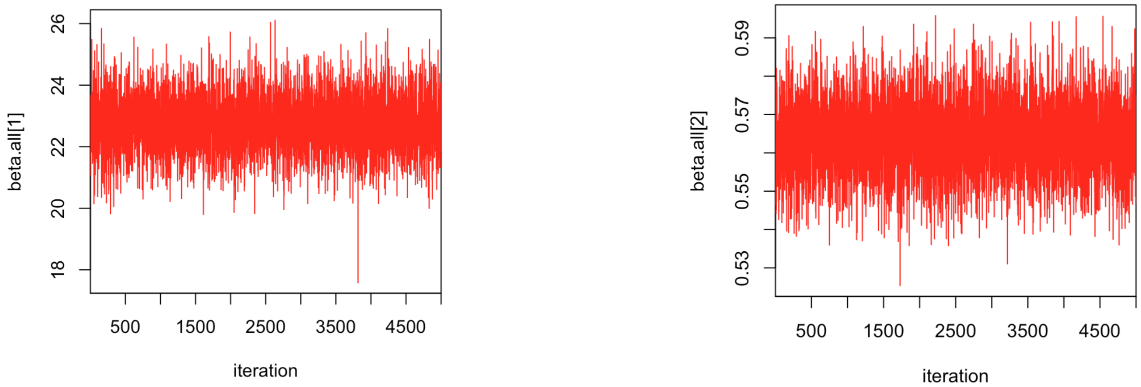
6.1 Results on simulated dataset

In order to apply the MCMC sampling methods on the simulated data, we ran the model for 6000 iterations with the first 1000 iterations being discarded. The diagnostic plots from both JAGS and Stan models show that convergence was reached for both parameters in both approaches. When assessing convergence for each parameter, we looked for a consistent pattern with minimum changes over a large number of iterations. Figure 1 is an illustration of successful convergence.

Figure 2 shows that both approaches arrived at very similar posterior distributions for both parameters in this example. Furthermore, table 1 shows that if the means of these posterior distributions were used as the point estimates for the two parameters of interest β_0 and β_1, then both JAGS and Stan gave the estimated values that were very close to the true values used to generate the data. This means that the sampling outcomes are reliable in this case.

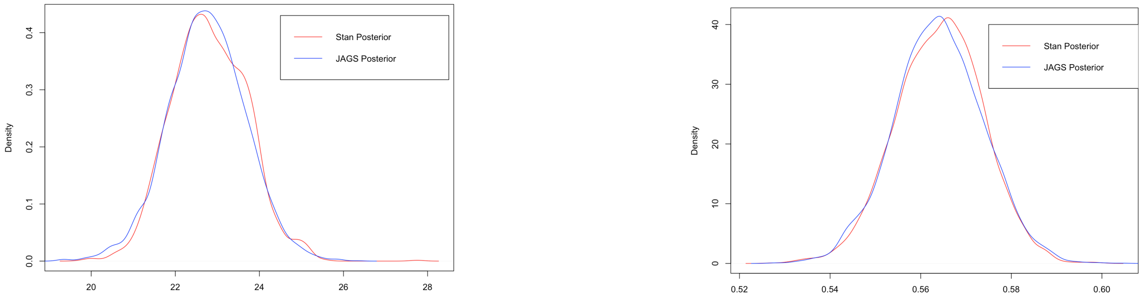
In term of efficiency, the sampling process took 0.435 seconds with JAGS and results in a perfect effective sample size of 5000 for both β_0 and β_1 (from 5000 iterations). On average, JAGS would give 1150 effective samples per 0.1 second for β_0 and β_1 respectively in this example. On the other hand, Stan took 8.314 seconds to complete the sampling process and produces an effective sample size of 274 and 1704 for β_0 and β_1 respectively (from 5000 iterations). Equivalently that means about 4 and 21 effective samples per 0.1 second for β_0 and β_1 respectively. Clearly, JAGS is more effective than STAN in this example.

Figure 1: Illustrations of convergence



(a) β_0 sampling (b) β_1 sampling

Figure 2: Posterior distributions obtained from JAGS and Stan



(a) Posterior distributions for β_0 sampling (b) Posterior distributions β_1 sampling

Table 1: Estimated parameters on simulated dataset

Parameter	β_0	β_1
Stan ESS	274	1704
JAGS ESS	5000	5000
Stan mean estimate	22.75	0.56
JAGS mean estimate	22.8	0.56
True value	23.8	0.57

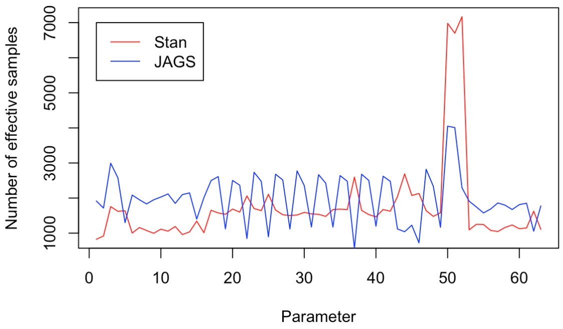
6.2 Results on accelerometer dataset

In order to decide on the number of iterations to run for both the JAGS and Stan models for this large dataset, we tried running the model in JAGS and Stan for a few different numbers of iterations, between 5000 iterations to 20000 iterations. We then inspected the diagnostic plot for each of the 63 parameters to see if convergence was reached. With the smaller total number of iterations or smaller number of discarded iterations, convergence did not occur for some parameters in either one of both of the models. Therefore, we decided to run the model for 20000 iterations with the first 5000 iterations being discarded so that convergence was reached for all the parameters. We then compared the final sampling outputs from the JAGS model and the Stan model.

Similar to the results on the simulated dataset, we found that the sampling outcomes from both approaches on this dataset also arrived at very similar posterior distribution for each parameter. Appendix 1 shows the mean of the posteriors obtained from JAGS and Stan for each parameter. From table 2 in appendix 1, we can see the point estimates of the parameters, which are the means of their respective posterior distributions, are very similar.

In term of sampling efficiency, figure 3 shows the number of ESS obtained from the models in JAGS and Stan. Stan produced a much higher ESS on some parameters, while JAGS produced a higher ESS of a large number of parameters. On top of that, computational time was a major advantage of JAGS for this dataset. The sampling process took just over two hours to complete in JAGS while it took around 31.5 hours in Stan. From appendix 1 it is obvious that the average number of effective samples per hour for each parameter is much larger from the sampling output in JAGS than Stan.

Figure 3: Effective Sample Size for 15000 iterations



Once again, JAGS produced a better sampling outcome for the majority of parameters in this dataset and was faster over all compared to Stan.

7 Discussion

This report provides a brief introduction to the framework behind Bayesian inference and the idea of MCMC sampling methods. It also gives an overview of the data as well as the statistical tools we used to implement MCMC methods in this project. The advancement of computer hardware's power together with the availability of statistical software like JAGS and Stan allows analysis that was not possible a few decades ago to take place. Our project implemented sampling in JAGS and Stan, assessed and compared the sampling outputs to inspect their efficiency in term of effective sampling and computational efficiency. In both of the examples reported, the MCMC sampling was more effectively delivered by JAGS than Stan. While both approaches arrived at more or less the same posterior distributions for the unknown parameters, JAGS gave a perfect sampling output in the more simple case of the linear mixed-effects simulated data and a much better overall performance in the more complicated case of accelerometer data with a logistic regression mixed effects model. In the more complicated dataset, Stan was able to get a better effective sample size on parameters that took JAGS a long time to converge. This could be an indication that the HMC methods implemented in Stan explores the parameter space more efficiently in complex cases where JAGS may not do so well. For future work, we would be interested to come across a dataset where JAGS does not perform so well so that we can really explore the advantages of the HMC sampling approach.

8 Acknowledgement

I would like to thank the Australian Mathematical Sciences Institute for giving me the opportunity to participate in the AMSI Vacation Research Scholarship program 2016/2017 and my supervisor, Dr Christopher Drovandi, for his meaningful directions and consistent support throughout the project.

A Appendix 1: Sampling results on accelerometer data

Table 2: Summary of sampling outputs from JAGS and Stan

Parameter	**STAN ESS**	**JAGS ESS**	**Stan ESS/hour**	**JAGS ESS/hour**	**Stan mean**	**JAGS mean**
1	825	1917	26	924	1.73	1.727
2	921	1716	29	827	0.03	0.029
3	1757	2993	56	1442	-0.5	-0.494
4	1623	2580	51	1243	-0.04	-0.045
5	1638	1302	52	627	-1.4	-1.412
6	1004	2085	32	1004	-1.98	-1.971
7	1161	1953	37	941	0.39	0.403
8	1076	1833	34	883	-0.03	-0.02
9	998	1951	32	940	-1.48	-1.466
10	1114	2030	35	978	-0.26	-0.24
11	1057	2118	33	1020	-0.97	-0.959
12	1190	1849	38	891	1.99	1.994
13	960	2099	30	1011	-2.1	-2.096
14	1035	2147	33	1034	0.33	0.335
15	1344	1401	43	675	-2.53	-2.524
16	1012	2010	32	968	-1.75	-1.743
17	1653	2497	52	1203	-0.01	-0.016
18	1577	2613	50	1258	0	0.004
19	1537	1122	49	540	0.38	0.389
20	1687	2503	53	1206	-0.07	-0.077
21	1600	2367	51	1140	0	0
22	2062	847	65	408	-6.27	-6.262
23	1702	2735	54	1317	0.02	0.01
24	1643	2476	52	1193	0	0.004
25	2108	895	67	431	-3.4	-3.394
26	1659	2680	53	1291	-0.49	-0.499
27	1526	2517	48	1212	0.64	0.642
28	1503	1119	48	539	-2	-1.988
29	1518	2775	48	1337	-0.16	-0.165
30	1592	2357	50	1135	0.37	0.366

Parameter	**STAN ESS**	**JAGS ESS**	**Stan ESS/hour**	**JAGS ESS/hour**	**Stan mean**	**JAGS mean**
31	1550	1175	49	566	-2.19	-2.183
32	1537	2668	49	1285	-0.01	-0.014
33	1478	2423	47	1167	0	0.004
34	1672	1174	53	565	0.3	0.31
35	1684	2640	53	1272	-0.09	-0.093
36	1672	2480	53	1194	0	0
37	2599	567	82	273	-5.98	-5.963
38	1649	2679	52	1290	-0.12	-0.127
39	1530	2503	48	1206	0.71	0.715
40	1466	1199	46	578	-0.71	-0.709
41	1673	2623	53	1263	0.17	0.169
42	1626	2478	52	1194	-1.56	-1.564
43	2042	1119	65	539	-4.36	-4.341
44	2690	1041	85	501	0.22	0.223
45	2074	1227	66	591	-1.57	-1.572
46	2131	724	67	349	-6.23	-6.218
47	1642	2820	52	1358	-0.26	-0.263
48	1478	2338	47	1126	-0.32	-0.316
49	1581	1167	50	562	-2.13	-2.123
50	6980	4049	221	1950	0.04	0.041
51	6700	4010	212	1932	0	0.003
52	7170	2294	227	1105	0.22	0.221
53	1095	1910	35	920	-0.01	-0.011
54	1252	1748	40	842	-0.09	-0.093
55	1248	1577	40	760	-0.26	-0.262
56	1078	1688	34	813	-0.02	-0.024
57	1046	1861	33	897	-0.1	-0.099
58	1162	1798	37	866	-0.06	-0.064
59	1233	1674	39	806	-0.27	-0.268
60	1126	1813	36	873	0	0
61	1148	1854	36	893	-0.18	-0.185
62	1626	1056	51	509	0.31	0.313
63	1108	1776	35	855	0.01	0.015

References

Denwood, M. J. (2016). "runjags: An R Package Providing Interface Utilities, Model Templates, Parallel Computing Methods and Additional Distributions for MCMC Models in JAGS". In: *Journal of Statistical Software* 71.9, pp. 1–25. DOI: `10.18637/jss.v071.i09`.

Depaoli, S., J. P. Clifton, and P. R. Cobb (2016). "Just Another Gibbs Sampler (JAGS): Flexible Software for MCMC Implementation". In: *Journal of Educational and Behavioral Statistics* 41.6, pp. 628–649.

Drovandi, C. C., C. Holmes, J. McGree, K. Mengersen, S. Richardson, and E. Ryan (2015). *A principled experimental design approach to Big Data analysis.* URL: `http://eprints.qut.edu.au.ezp01.library.qut.edu.au/87946/`.

Fernández-i-Marín, X. (2016). "ggmcmc: Analysis of MCMC Samples and Bayesian Inference". In: *Journal of Statistical Software* 70.9, pp. 1–20. DOI: `10.18637/jss.v070.i09`.

Gelman, A. (2005). *Bayesian data analysis.* 2nd ed. Boca Raton, Fla: Chapman & Hall/CRC.

Geman, S. and D. Geman (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 6.6, pp. 721–741. DOI: `10.1109/TPAMI.1984.4767596`. URL: `http://dx.doi.org/10.1109/TPAMI.1984.4767596`.

Kruschke, J. K. (2015). *Doing Bayesian data analysis: a tutorial with R, JAGS, and Stan.* Amsterdam: Academic Press.

McGrayne, S. B. (2011). *The theory that would not die: How bayes' rule cracked the enigma code, hunted down russian submarines, & emerged triumphant from two centuries of controversy.* New Haven [Conn.]: Yale University Press.

Plummer, M. (2015). *JAGS Version 4.0.0 user manual.* URL: `http://www.uvm.edu/~bbeckage/Teaching/DataAnalysis/Manuals/manual.jags.pdf`.

— (2016). *rjags: Bayesian Graphical Models using MCMC.* URL: `https://CRAN.R-project.org/package=rjags`.

Plummer, M., N. Best, K. Cowles, and K. Vines (2006). "CODA: Convergence Diagnosis and Output Analysis for MCMC". In: *R News* 6.1, pp. 7–11. URL: `https://journal.r-project.org/archive/`.

Stan Development Team (2016). *RStan: the R interface to Stan.* R package version 2.14.1. URL: `http://mc-stan.org/`.

— (2017). *shinystan: Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesian Models.* R package version 2.3.0. URL: `http://mc-stan.org/`.

Su, Y. and M. Yajima (2015). *R2jags: Using R to Run 'JAGS'.* URL: `https://CRAN.R-project.org/package=R2jags`.

Trost, S. G., P. D. Loprinzi, R. Moore, and K. A. Pfeiffer (2011). "Comparison of accelerometer cut points for predicting activity intensity in youth". In: *Medicine and Science in Sports and Exercise* 43.7, pp. 1360–1368. DOI: `10.1249/MSS.0b013e318206476e`.

Yildirim, I. (2012). *Bayesian Inference: Gibbs Sampling.* URL: `http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf`.