# standard options

Other options include: MCMCpack and MCMC R Package

Most standard

BUGS (Bayesian inference Using Gibbs Sampling)

\ref{The BUGS Project | MRC Biostatistics Unit}{ http://www.mrc-bsu.cam.ac.uk/software/bugs/ }

* openBUGS/winBUGS seem to be dead projects.
* Good to test on small datasets
* **winBUGS 1.4.3**(<https://www.mrc-bsu.cam.ac.uk/software/bugs/the-bugs-project-winbugs/>)
* **openBUGS 3.2.3** For a version that BUGS (BRugs) that sits within the R statistical package, <http://www.openbugs.net/w/Downloads>
* <https://m-clark.github.io/bayesian-basics/appendix.html#bugs-example>

JAGS (Just Another Gibbs Sampler)

\ref{ Just Another Gibbs Sampler } { http://mcmc-jags.sourceforge.net/} ;

- easy to use

- JAGS is (unfortunately not) vectorized.

- <https://www.r-bloggers.com/getting-started-with-jags-rjags-and-bayesian-modelling/>

BEST (Bayesian ESTimation supersedes the t test)

* BEST uses the JAGS package (Plummer, 2003) to produce samples from the posterior distribution of each parameter of interest. You will need to download JAGS from http://sourceforge. net/projects/mcmc-jags/ and install it before running BEST.

<https://cran.r-project.org/web/packages/BEST/vignettes/BEST.pdf>

* <http://doingbayesiandataanalysis.blogspot.com/2013/06/new-r-package-for-best-bayesian.html>
* <http://www.indiana.edu/~kruschke/BEST/>
* <https://www.youtube.com/watch?v=fhw1j1Ru2i0&feature=youtu.be>

STAN (what is the meaning?)

\ref{Project Home Page}{}

* <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

compile your model into a C++ program, which runs much faster

tan automatically converts hierarchical models you specify to compiled C++ code and uses a variant of Hamiltonian Monte Carlo to drastically speed up calculations in order to perform Bayesian inference on multi-level models. The Stan Reference Manual (Modeling Language Manual) contains many examples of actual Stan code for implementing various models.

<https://github.com/stan-dev/stan/releases/download/v2.17.0/stan-reference-2.17.0.pdf>

`differences' between the three: JAGS and Stan

<https://www.r-bloggers.com/jags-and-stan/>

<http://www.sumsar.net/files/academia/user_2015_tutorial_bayesian_data_analysis_short_version.pdf>

see example

●What?

○ Bayesian data analysis is a flexible method to fit any type of statistical model.

○ Maximum likelihood is a special case of Bayesian model fitting.

● Why?

○ Makes it possible to define highly custom models.

○ Makes it possible to include information from many sources, for example, data and expert knowledge.

○ Quantifies and retains the uncertainty in parameter estimates and predictions.

● How?

○ R! Using ABC, MCMCpack, JAGS, STAN, R-inla, etc.

How Bayesian inference works

<https://www.youtube.com/watch?v=5NMxiOGL39M>

<http://brohrer.github.io/how_bayesian_inference_works.html>

<https://docs.google.com/presentation/d/1325yenZP_VdHoVj-tU0AnbQUxFwb8Fl1VdyAAUxEzfg/edit>

<https://www.youtube.com/watch?v=lNrpPNk6InU>

Bayesian inference for Poisson data

POISSON

<https://www.statisticshowto.datasciencecentral.com/poisson-distribution/>

<https://www.umass.edu/wsp/resources/poisson/>

<http://mathworld.wolfram.com/PoissonDistribution.html>

<https://stattrek.com/probability-distributions/poisson.aspx> GOOD

Rjags

<https://www.coursera.org/lecture/mcmc-bayesian-statistics/jags-model-poisson-regression-VWH44>

<https://www4.stat.ncsu.edu/~reich/st590/code/PoissonGamma>

<https://georgederpa.github.io/teaching/countModels.html>

BUGS

<http://www.dtic.mil/dtic/tr/fulltext/u2/a624076.pdf>

STAN

<https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

rstan

<https://github.com/stan-dev/rstan/wiki/Installing-RStan-on-Windows>

rtools

R packages

rstan

rstanarm and brms.

**Parameter vs hyperparametre**

[**https://en.wikipedia.org/wiki/Hyperparameter**](https://en.wikipedia.org/wiki/Hyperparameter)

**Bayesian statistics**

an statistics, a hyperparameter is a parameter of a prior distribution; the term is used to distinguish them from parameters of the model for the underlying system under analysis.

For example, if one is using a beta distribution to model the distribution of the parameter p of a Bernoulli distribution, then:

p is a parameter of the underlying system (Bernoulli distribution), and

α and β are parameters of the prior distribution (beta distribution), hence hyperparameters.

One may take a single value for a given hyperparameter, or one can iterate and take a probability distribution on the hyperparameter itself, called a hyperprior.

(Regular) parameters are those that would be required to describe the physical process itself, and would be determined by the laws of the actual system you are modeling. In other words, they are properties of the thing being modeled, not properties of the model itself.

Hyperparameters are those that are not parameters in the actual physical process, and are only parameters in the model. You would have only introduced them to make your model "work" in the presence of finite data and/or finite computation time. If you had infinite power to measure or compute anything, hyperparameters would no longer exist in your model, since they wouldn't be describing any physical aspect of the actual system.

A hyperparameter is simply a parameter that impacts, completely or partly, other parameters. They do not directly solve the optimization problem you face, but rather optimize parameters that can solve the problem (hence the hyper, because they are not part of the optimization problem, but rather are "addons").