

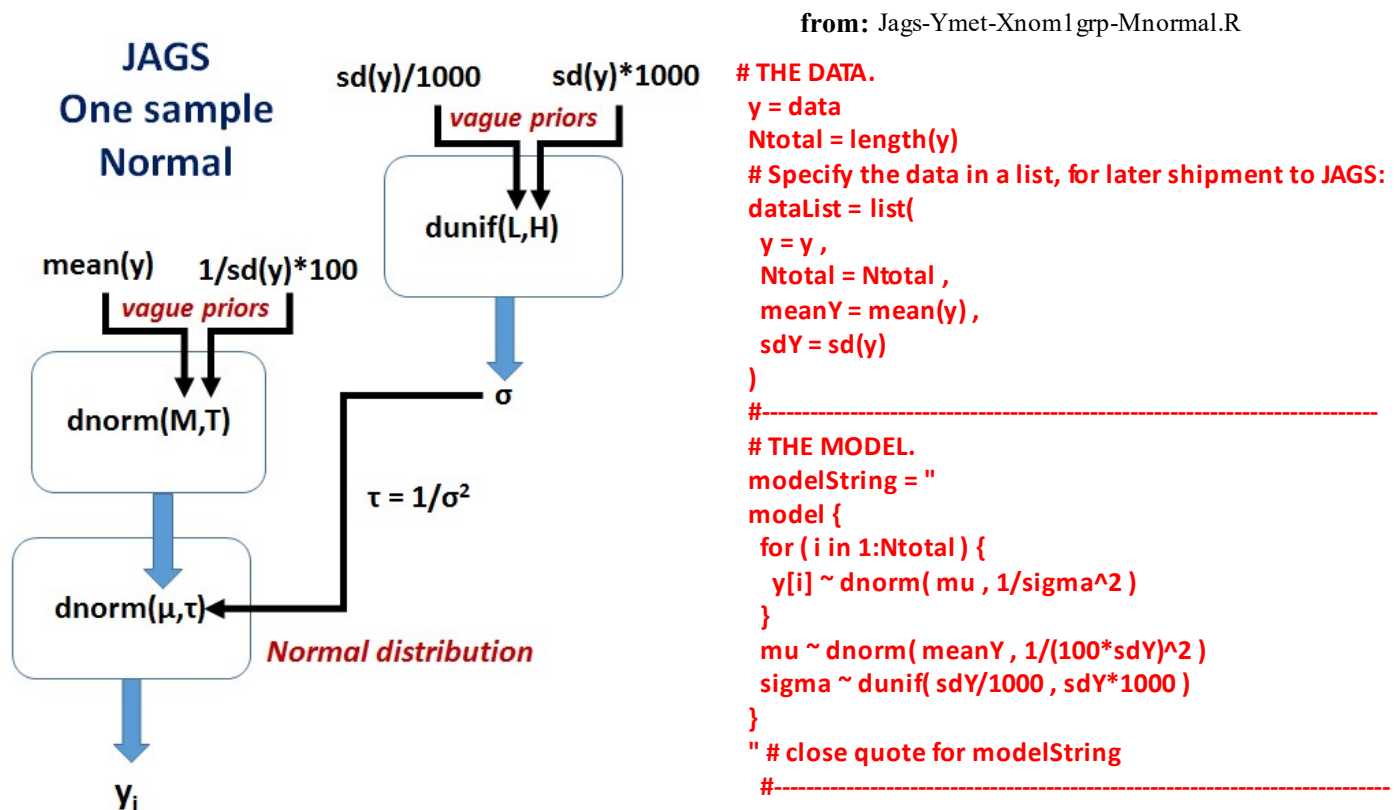
ORIGIN ≡ 0

Model Diagrams for MCMC in JAGS

Robust One Sample t-Test

In the text by J.K. Kruschke (K): *Doing Bayesian Data Analysis - A Tutorial with R, JAGS, and Stan*, strong emphasis is placed on the process of constructing suitable models for data. Moreover, constructing a model in graphical format is presented as a highly desirable prelude to formal specification in JAGS. Models are written in 2nd level files of his scaffold, inside function `genMCMC()`. Presented below is a modified summary of his suggested process showing the relationship between the graph and JAGS code. Slightly greater emphasis is placed here in the graphs on the coding functions for distributions in JAGS, and the explicit forms of uninformative or "vague" priors actually utilized in these models. Two models, "Normal" and "Robust" are analyzed below using K's data and scaffolds. The complete set of files are available at <https://sites.google.com/site/doingbayesiandataanalysis/>.

One Sample Analysis using assumed Normal Distribution for the data:



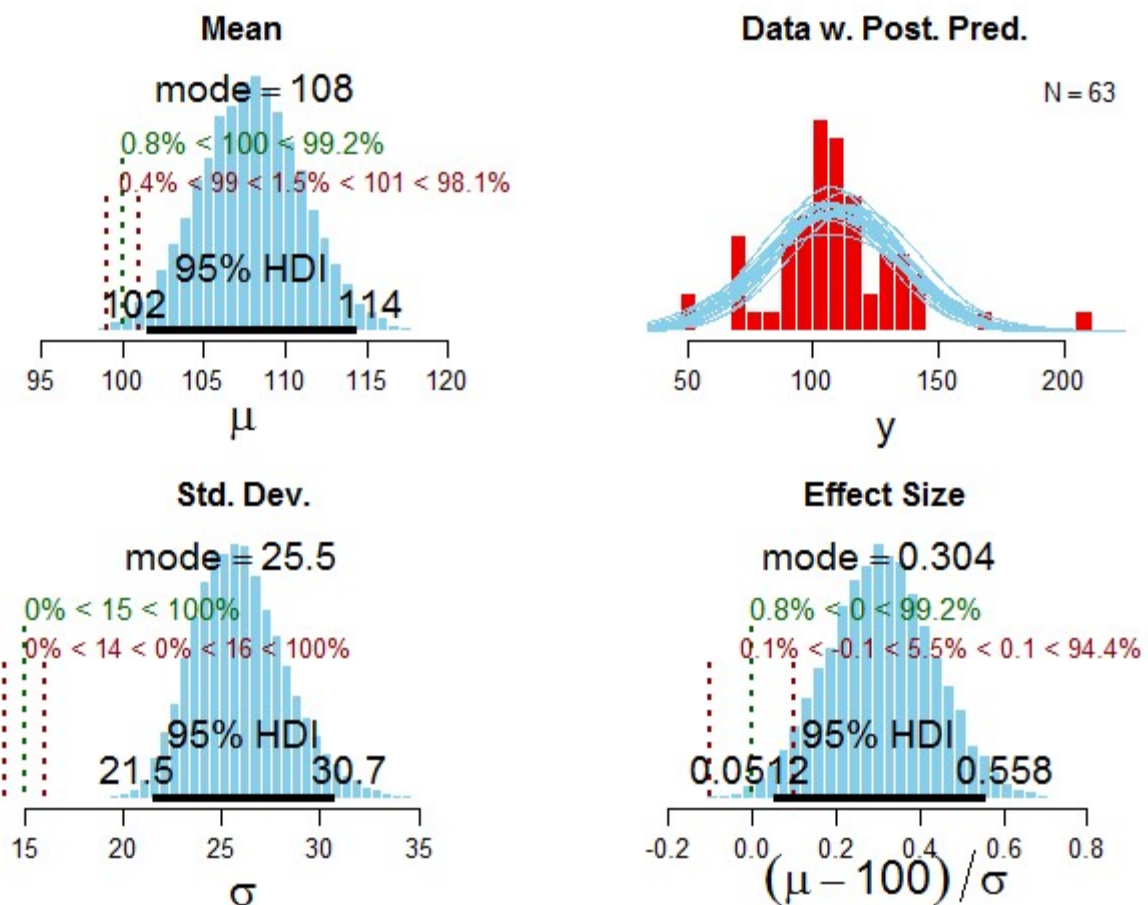
As a simplifying convention, the data in the scaffolds is always given the name y with each data value from a dataset indexed by subscript i , thus y_i . Commonly y_i are assumed to be derived from a Normal distribution implemented in JAGS script (similar but not identical with R script) as function `dnorm()` with two parameters (μ , τ). The mean μ is as we might expect. However, for technical reasons related to the underlying math, JAGS uses "precision" $\tau = 1/\sigma^2$ as its preferred measure of variability instead of standard deviation σ (or variance σ^2). Except for user inconvenience, perhaps, this is of little concern - until one forgets to make the conversion.

In a JAGS Bayesian models, each parameter of the distribution of data y_i must be interpreted in terms of prior probabilities. Fixed assignments involving direct conversion of one parameter to another are shown with black arrows above, such as the direct conversion of σ to τ . Parameters values derived from prior probability distributions are shown with blue arrows. Prior distributions have parameters of their own, by convention indicated by Latin capital letters. The choice of what kind of distribution to use to inform priors is left to the researcher and may be driven by prior understanding of the data. Often, however, the research plan is to offer a sample of parameter values from "vague" (uninformative) prior distributions. Vague they may be, but JAGS requires that each prior distribution be explicitly set by values assigned to *their* parameters.

So reading the chain in the graph leading to precision parameter τ for the `dnorm()` distribution on data y_i , values of σ are drawn from a uniform distribution `dunif()` stocked with vague parameters L (for lower bound) and H (for upper bound). Parameters L & H are set suitably wide by scaling them (1000) to observed standard deviation of the data y . Parameter σ is converted to parameter τ by fixed assignment. Similarly, in the chain specifying μ , priors are derived from a prior distribution, in this case `dnorm()` with prior parameters M (mean) and T (precision, again). In similar fashion, M & T are set with wide properties scaled respectively to mean(y) for M, and low precision ($\text{sd}(y)*100$) relative to the data y for T.

Looking that The JAGS code, one can see that each blue arrow is represented by an assignment statement using the symbol \sim . Black arrows are explicitly set either within the function bracket itself, or in THE DATA block. I agree with K that making a diagram such as this for a proposed model is a helpful first step in producing workable JAGS code. Otherwise, it would be easy to mis-specify something.

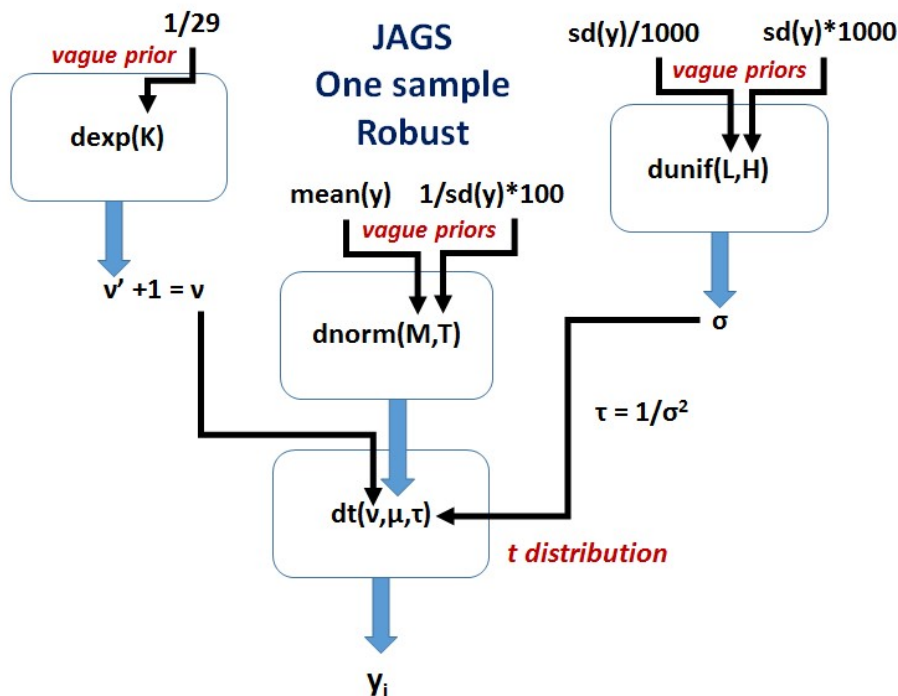
In running the above model in JAGS (previously in 010 MCMC), the following results were obtained:



Lack of fit of the data (red histogram) relative to the set of probable normal distributions (light blue in same graph) motivates use of a more robust distribution that has a higher hump near the mean, but doesn't flatten so much in the presence of data outliers. In the text, K offers suggests the t-distribution for this. Unlike previously encountered, perhaps, the t-distribution in Bayesian analysis allows for translocation of the mean and scaling of variance in addition to change in shape of the curve traditionally associated with "degrees of freedom". So accordingly, this t-distribution has three parameters: μ for translocation of the mean, σ as measure of variance, and ν called the "normality parameter" by K for change in curve shape. In this context, ν is not restricted to integers, but can be any real number. So, with three parameters to handle in the model, we need a diagram!

from: Jags-Ymet-Xnomlgrp-Mrobust.R

Robust Estimation with the t-distribution:



THE DATA.

y = data

Ntotal = length(y)

Specify the data in a list, for later shipment to JAGS:

dataList = list(

y = y ,

Ntotal = Ntotal ,

meanY = mean(y) ,

sdY = sd(y)

)

THE MODEL.

modelString = "

model {

for (i in 1:Ntotal) {

 y[i] ~ dt(mu , 1/sigma^2 , nu)

}

mu ~ dnorm(meanY , 1/(100*sdY)^2)

sigma ~ dunif(sdY/1000 , sdY*1000)

nu ~ dexp(1/30.0)

}

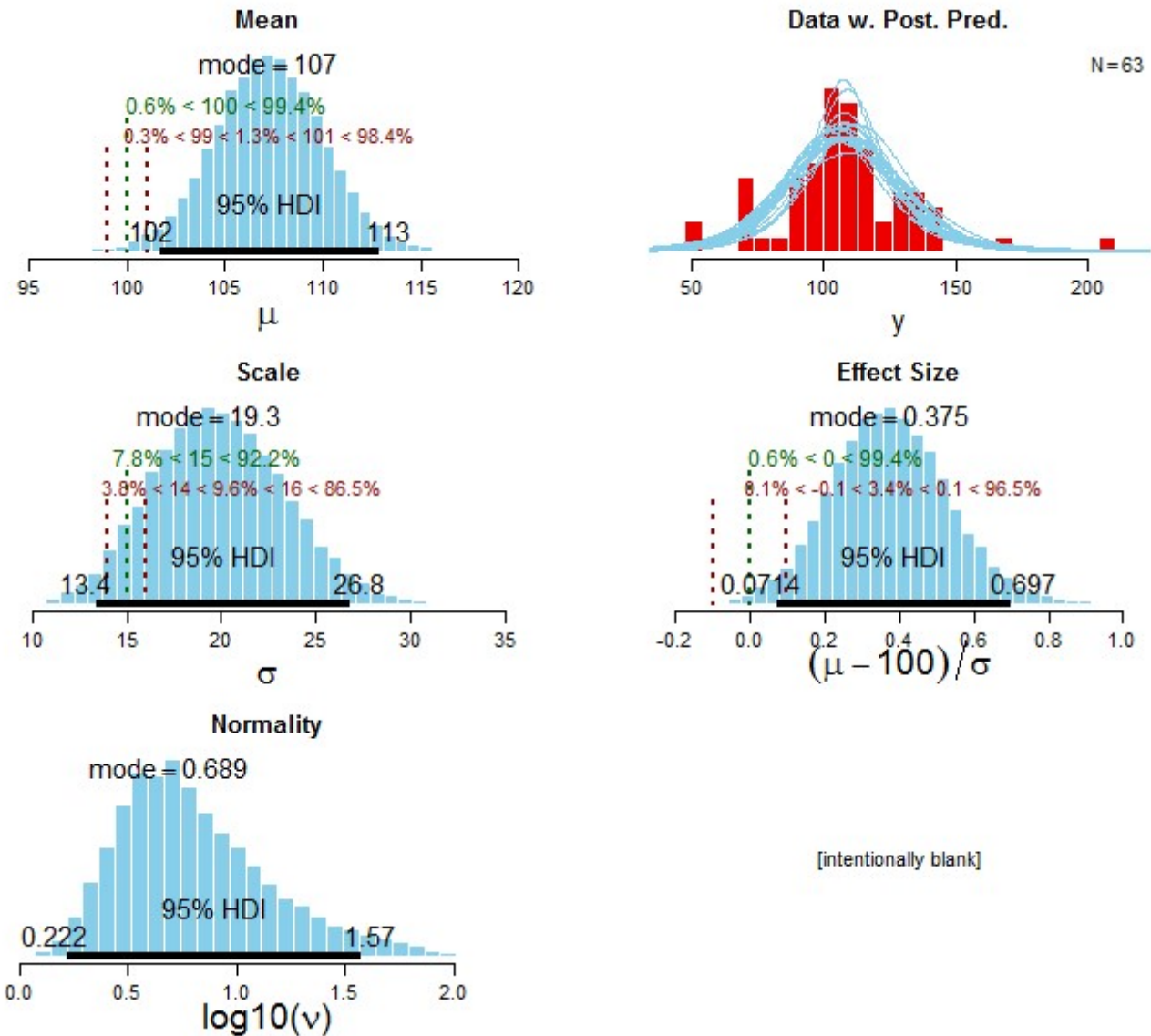
" # close quote for modelString

In the Robust model, the t-distribution specified by JAGS `dt()` function replaces the Normal distribution `dnorm()` for distribution of the data y_i . Parameters μ and σ are modeled in the prior just as before. So the only new thing is parameter ν . Here we see that parameter ν' is derived from an exponential distribution specified by JAGS function `dexp()`. This function has a lower bound of zero, but this is not permitted as a value for ν in `dt()`. So, like the conversion from σ to τ , ν' is converted to new parameter ν , by adding one to ν' . The vague prior set for parameter K in `dexp()` essentially sets ν to a high enough value that the t-distribution has the same properties as the Normal Distribution. If posterior estimates of ν are obtained at such high level, then the Robust model adds nothing over the Normal model for interpreting the data.

Along with all the diagnostics, the following MCMC results were obtained from JAGS:

> show(summaryInfo)

	Mean	Median	Mode	ESS	HDI _{mass}
mu	107.2188247	107.1986732	107.1967457	20000.0	0.95
sigma	19.8685986	19.7441150	19.3499467	8739.6	0.95
nu	9.6996929	5.5874225	3.4117924	3163.6	0.95
log10(nu)	0.8097367	0.7472115	0.6892981	5965.6	0.95
effSz	0.3746894	0.3696445	0.3750577	17266.9	0.95
	HDI _{low}	HDI _{high}	CompVal	PcntGtCompVal	
mu	101.65990729	112.8534621	100	99.375	
sigma	13.38947018	26.7893499	15	92.200	
nu	1.11302391	32.1646851	NA	NA	
log10(nu)	0.22202070	1.5697941	NA	NA	
effSz	0.07143793	0.6973514	0	99.375	
	ROPE _{low}	ROPE _{high}	PcntLtROPE	PcntInROPE	PcntGtROPE
mu	99.0	101.0	0.270	1.34	98.390
sigma	14.0	16.0	3.845	9.61	86.545
nu	NA	NA	NA	NA	NA
log10(nu)	NA	NA	NA	NA	NA
effSz	-0.1	0.1	0.095	3.37	96.535



Here we see from the Post Prediction Plot (top-right) a somewhat better fit with the data. The posterior distribution for v on \log_{10} scale has a mode of 0.689. A value of $\log_{10}(30) = 1.4771$ or higher indicates no better fit of the data for the Robust model versus the Normal model. In this case, we see some improvement with the Robust model. Improvement is also seen in comparing Comparison Values and ROPE percentages, and in widths of the HDI's. So a Robust is a somewhat better model of the data.

Pairwise plots of posterior parameter distributions, below, show independence of μ with σ and v , but noticeable correlation of σ with v . According to K, this makes sense, because data outliers involve a tradeoff between increasing variance with σ vs changing the shape of the t-distribution for fatter tails with v .

It is interesting to note that the Robust model, utilizing `dt()`, essentially reverts to a Normal model when the posterior probability of $v > 30$, or equivalently $\log_{10}(v) > 1.4771$. So, as general practice, there's no reason not to use the Robust model always.

