Module 3: The JAGS software with simple examples

Andrew Parnell, School of Mathematics and Statistics,
University College Dublin

Learning outcomes

- LO 1
- LO 2
- LO 3

Simple example in JAGS

In later modules we will start using JAGS to fit models like this.
The code is much simpler than the previous R version:

```
library(rjags)
modelstring ='
  model {
    # Likelihood
    x ~ dnorm(theta,1/pow(0.8,2))
    # Prior
    theta ~ dnorm(2.3,1/pow(0.5,2))
  }
'
# Set up data
data=list(x=3.1)
# Run jags
model=jags.model(textConnection(modelstring), data=data)
output=coda.samples(model=model,variable.names=c("theta"),
# Plot output
plot(density(output[[1]]))
```

Pos

1.0 –

0.8 –

0.6 –

ensity
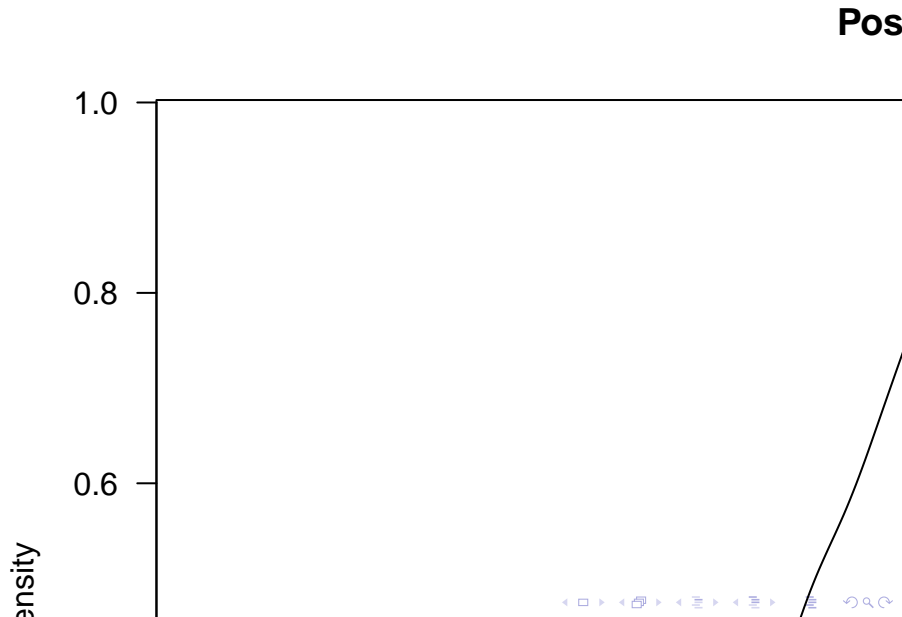
What are the assumptions involved in this example?

- We've assumed that the normal distribution is appropriate for the likelihood and the prior
- We've only observed one data point. What if we observed many?
- We've assumed that the likelihood standard deviation is fixed at 0.8

A more complicated JAGS version

```
modelstring ='
  model {
    # Likelihood
    for(i in 1:n) { x[i] ~ dnorm(theta,1/pow(sd,2)) }
    # Prior
    theta ~ dnorm(2.3,1/pow(0.5,2))
    sd ~ dunif(0,100)
  }
'
# Set up data
data=list(x=c(3.1,2.7,4.2,3.6),n=4)
# Run jags
model=jags.model(textConnection(modelstring), data=data)
output=coda.samples(model=model,variable.names=
                        c("theta","sd"), n.iter=1000)
```

Now have four data points, two parameters and a prior for each

What if the observations aren't normal?

No problem! We just choose distributions which are appropriate for the type of data

```
modelstring ='
  model {
    # Likelihood
    for(i in 1:n) {
      x[i] ~ dgamma(alpha,beta)
    }
    # Prior
    alpha ~ dunif(0,100)
    beta ~ dunif(0,100)
    # Mean
    mean <- alpha/beta
  }
'
data=list(x=c(3.1,2.7,4.2,3.6),n=4)
model=jags.model(textConnection(modelstring), data=data)
output=coda.samples(model=model,variable.names=c("mean"),n
```

How do I specify the prior distribution?

There are several choices when it comes to specifying prior distributions:

- *Informative*, when there is information from a previous study, or other good external source, e.g $\theta \sim N(2.3, 0.5^2)$
- *Vague*, when there is only weak information, perhaps as to the likely range of the parameter e.g. $\theta \sim U(0, 100)$
- *Flat*, when there is no information at all about a parameter (very rare). In JAGS, write `theta ~ dflat()`

Choosing the prior and choosing the likelihood are very similar problems

Choosing likelihoods and priors

When creating Bayesian models it's helpful to know a lot of probability distributions. The ones we will use most are:

Distribution	Range	Useful for:
Normal, $N(\mu, \sigma^2)$	$(-\infty, \infty)$	A good default choice
Uniform, $U(a, b)$	(a, b)	Vague priors when we only know the range of the parameter
Binomial, $Bin(k, \theta)$	$[0, k]$	Count or binary data restricted to have an upper value
Poisson, $Po(\lambda)$	$[0, \infty)$	Count data with no upper limit
Gamma, $Ga(\alpha, \beta)$	$(0, \infty)$	Continuous data with a lower bound of zero
Multivariate	$(-\infty, \infty)$	Multivariate

Creating the posterior distribution

- In the very simple example, I was able to calculate the posterior distribution in just a couple of lines of R code
- When we have lots of parameters, and complicated prior distributions, we have to resort to *simulation*
- This means that we obtain *samples* from the posterior distribution rather than creating the probability distribution directly
- JAGS uses Markov chain Monte Carlo (MCMC) to create these samples. We will talk about this a bit more in later lectures/discussion

Summarising the posterior distribution

- Because we obtain samples from the posterior distribution, we can create any quantity we like from them
- e.g. we can obtain the mean or standard deviation simply from combining the samples together
- We can create quantiles e.g. 50% for the median
- We can create a Bayesian *credible interval* (CI) by calculating lower and upper quantiles
- When the posterior distribution is messy (e.g. multi-modal) we can use a *highest posterior density* (HPD) region

Example:

From the earlier simple example. First 5 posterior samples

```
output[[1]][1:5]
```

```
## [1] 2.653882 1.911947 2.509492 3.111370 2.805593
```

The mean and standard deviation:

```
c(mean(output[[1]]),sd(output[[1]]))
```

```
## [1] 2.5254535 0.4239647
```

A 95% credible interval

```
quantile(output[[1]],probs=c(0.025,0.975))
```

```
##     2.5%    97.5%
## 1.684793 3.353078
```