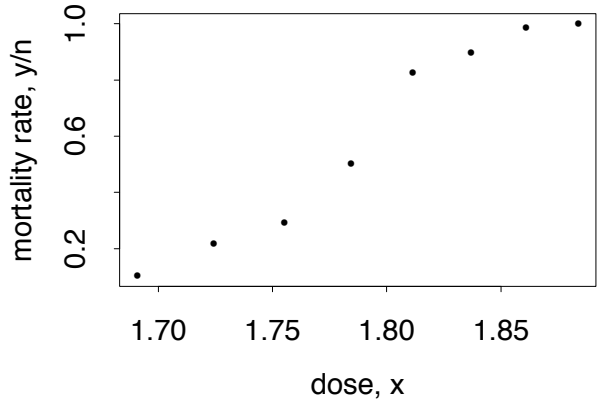
Generalised Linear regression Models

- Specification of Bayesian GLMs follows straightforwardly from previous discussion of linear models
- No closed form solution available, but straightforward to obtain samples from posterior using MCMC

Example: Beetles

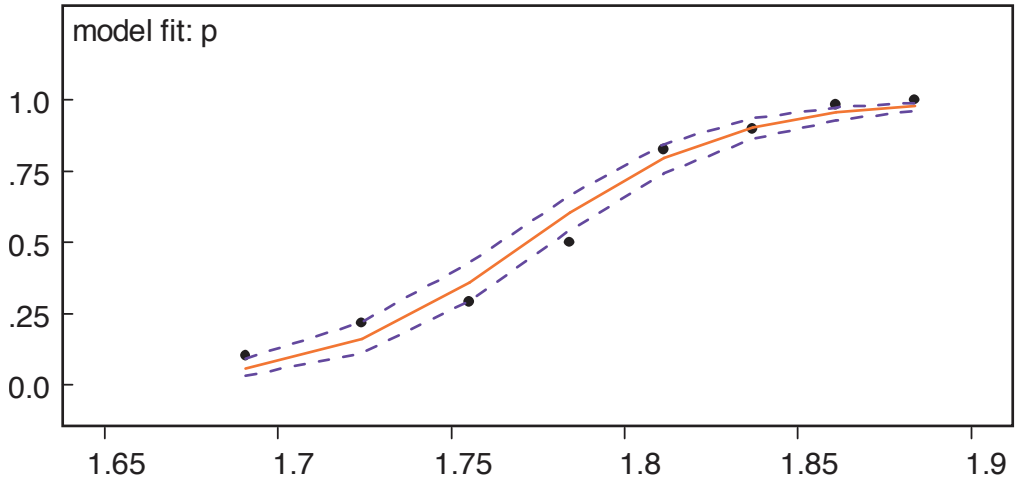
Dobson (1983) analyses binary dose-response data from a bioassay experiment in which the numbers of beetles killed after 5 hour exposure to carbon disulphide at N=8 different concentrations are recorded.



We start by fitting a logistic regression model

$$\begin{aligned} y_i &\sim \text{Binomial}(p_i, n_i) \\ \text{logit} p_i &= \alpha + \beta(x_i - \overline{x}) \\ \alpha &\sim \text{Normal}(0, 10000) \\ \beta &\sim \text{Normal}(0, 10000) \end{aligned}$$

Beetles: logistic regression model fit (red = posterior mean of p_i; blue = 95% interval; black dots = observed rate y_i/n_i)

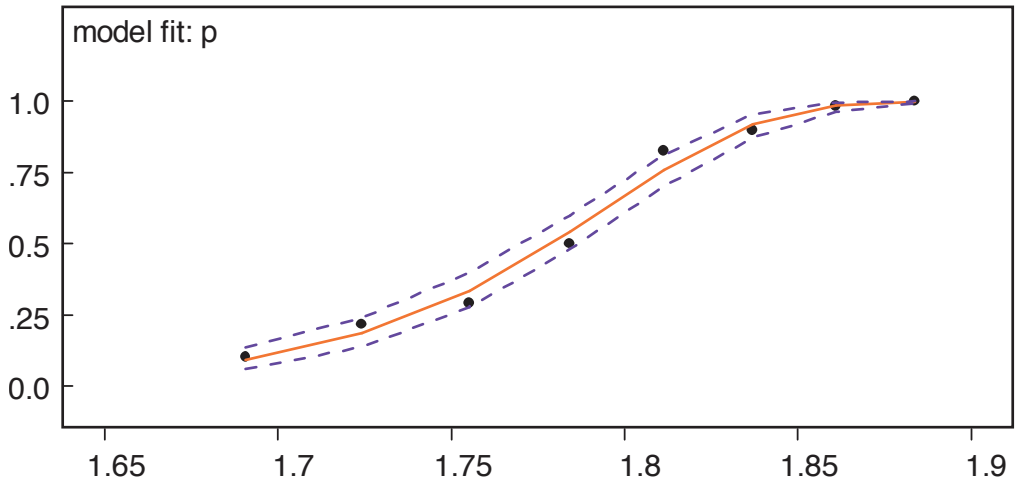


dose level i	obs. rate y_i/n_i	posterior mean of p_i	95% interval
1	0.10	0.06	(0.03, 0.09)
2	0.22	0.16	(0.11, 0.22)
3	0.29	0.36	(0.29, 0.43)
4	0.50	0.61	(0.54, 0.67)
5	0.83	0.80	(0.74, 0.85)
6	0.90	0.90	(0.86, 0.94)
7	0.98	0.96	(0.93, 0.97)
8	1.00	0.98	(0.96, 0.99)

Some evidence of lack of fit at extremes, so try alternative complementary log-log link function

$$\begin{aligned} y_i &\sim \text{Binomial}(p_i, n_i) \\ \text{cloglog} p_i &= \alpha + \beta(x_i - \overline{x}) \\ \alpha &\sim \text{Normal}(0, 10000) \\ \beta &\sim \text{Normal}(0, 10000) \end{aligned}$$

Beetles: cloglog regression model fit (red = posterior mean of p_i; blue = 95% interval; black dots = observed rate y_i/n_i)



dose level i	obs. rate y_i/n_i	posterior mean of p_i	95% interval
1	0.10	0.09	(0.06, 0.14)
2	0.22	0.19	(0.14, 0.24)
3	0.29	0.34	(0.28, 0.40)
4	0.50	0.54	(0.48, 0.60)
5	0.83	0.76	(0.70, 0.81)
6	0.90	0.92	(0.87, 0.95)
7	0.98	0.98	(0.96, 0.99)
8	1.00	1.00	(0.99, 1.00)

Can write probit model in two different ways

$$\text{probit}p_i \quad = \quad \alpha + \beta(x_i - \overline{x})$$

or

$$p_i \quad = \quad \Phi(\alpha + \beta(x_i - \overline{x}))$$

In WinBUGS , either

```
probit(p[i]) <- alpha + beta*(x[i]-mean(x[]))
```

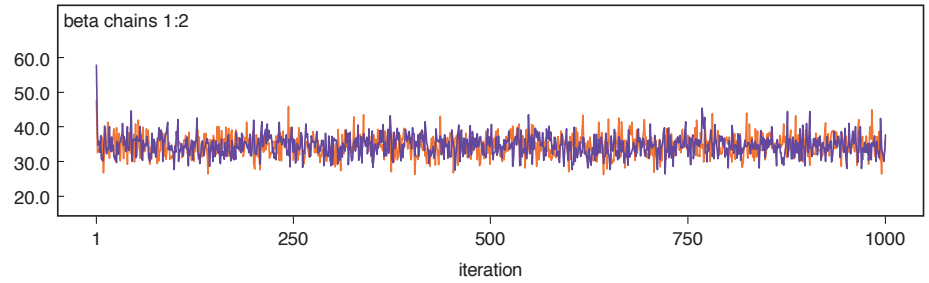
or

```
p[i] <- phi(alpha + beta*(x[i]-mean(x[])))
```

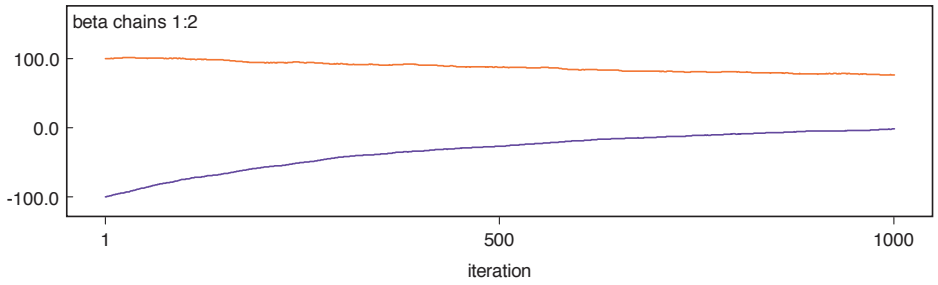
The second way is *slower*, but can be *more robust* to numerical problems.

Note the importance of centering the covariate (dose) in this example to reduce correlations between the parameters

History plot for slope, β : Centred covariate

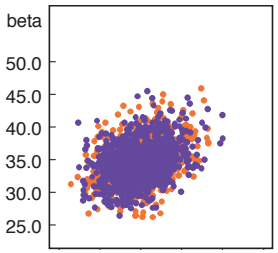


History plot for slope, β : Uncentred covariate

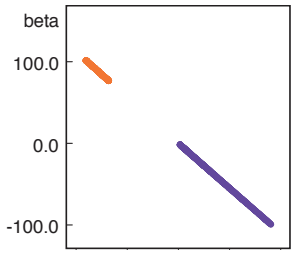


Bivariate scatter plot showing correlation between sampled values of α and β

Centered covariate



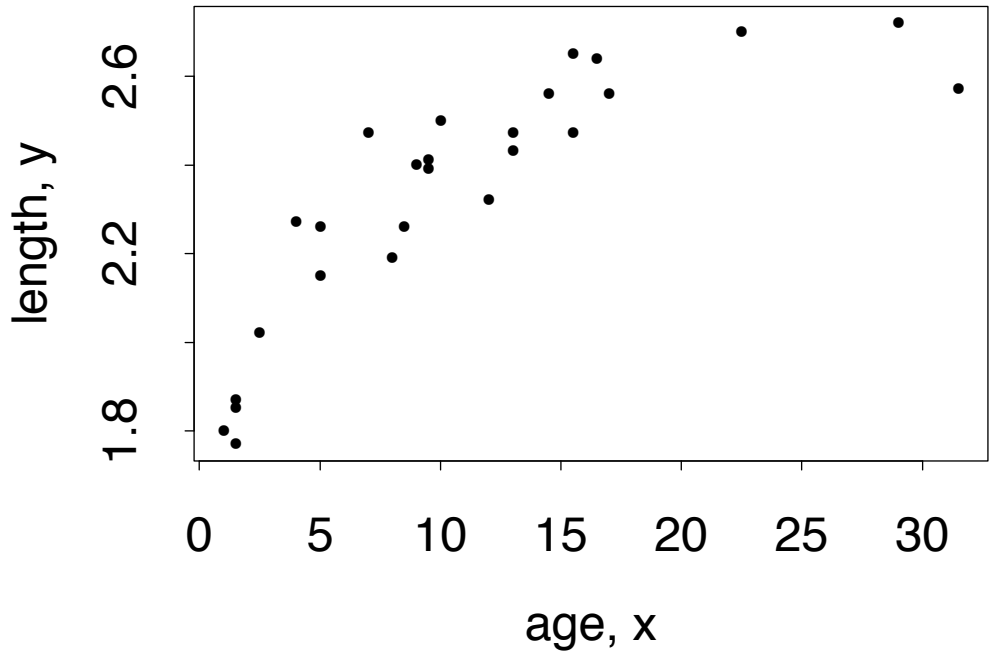
Uncentred covariate



Non linear regression models

Example: Dugongs

Carlin and Gelfand (1991) consider data on length (y_i) and age (x_i) measurements for 27 dugongs (sea cows) captured off the coast of Queensland



A frequently used nonlinear growth curve with no inflection point and an asymptote as x_i tends to infinity is

$$\begin{aligned} y_i &\sim \text{Normal}(\mu_i, \sigma^2) \\ \mu_i &= \alpha - \beta\gamma^{x_i} \end{aligned}$$

where $\alpha, \beta > 0$ and $\gamma \in (0, 1)$

Vague prior distributions with suitable constraints may be specified as e.g.

$$\begin{aligned} \alpha &\sim \text{Uniform}(0, 100) \\ \beta &\sim \text{Uniform}(0, 100) \\ \gamma &\sim \text{Uniform}(0, 1) \end{aligned}$$

Alternatively, vague Normal priors with appropriate bounds could be specified for α and β, e.g.

$$\begin{aligned} \alpha &\sim \text{Normal}(0, 10000)I(0,) \\ \beta &\sim \text{Uniform}(0, 10000)I(0,) \end{aligned}$$

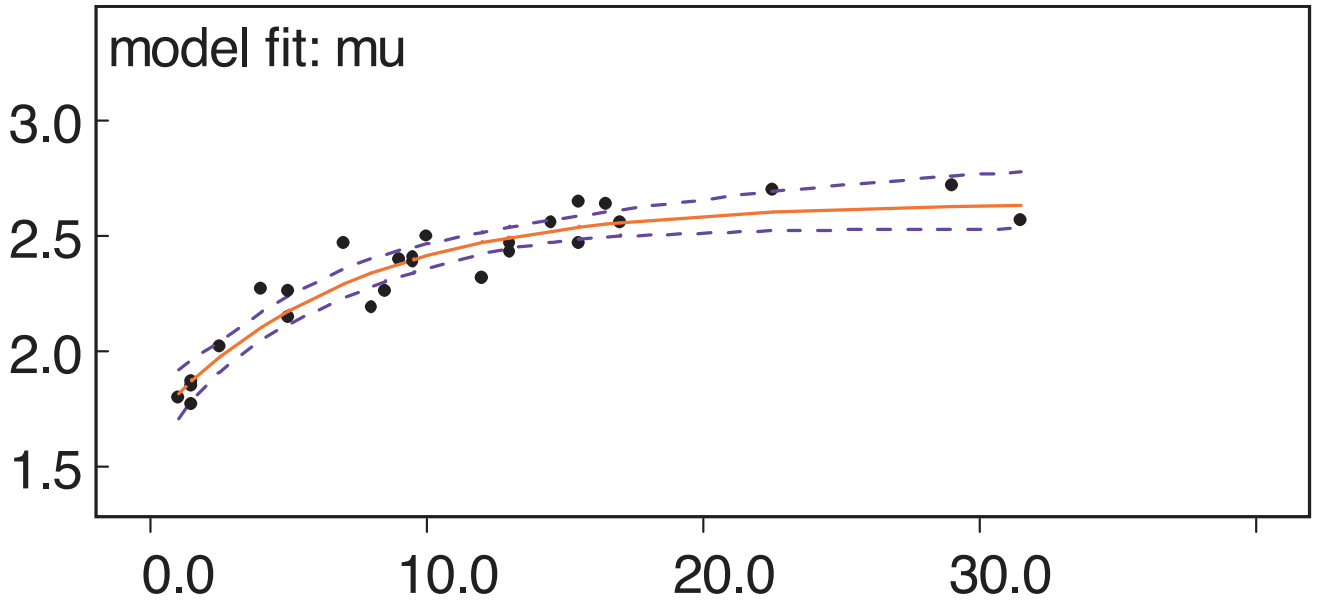
For the sampling variance, could specify uniform prior log variance or log sd scale

$$\log \sigma \sim \text{Uniform}(-10, 10)$$

or gamma prior on precision scale

$$1/\sigma^2 \sim \text{Gamma}(0.001, 0.001)$$

Dugongs: model fit (red = posterior mean of μ_i; blue = 95% interval)



Making predictions

- Important to be able to predict unobserved quantities for
  - 'filling-in' missing or censored data
  - model checking - are predictions 'similar' to observed data?
  - making predictions!

- Easy in MCMC/WinBUGS; just specify a stochastic node without a data-value - it will be automatically predicted

- Provides automatic imputation of missing data

- Easiest case is where there is no data at all: just 'forward sampling' from prior, *Monte Carlo* methods

Example: Dugongs — prediction

Suppose we want to project beyond current observations, eg at ages 35 and 40

Could explicitly set up predictions

```
for (i in 1:N){
    y[i]   ~   dnorm( mu[i], inv.sigma2 )
    mu[i] <-   alpha - beta * pow(gamma, x[i])
    }
mu35  <-   alpha - beta * pow(gamma, 35)
mu40  <-   alpha - beta * pow(gamma, 40)
y35    ~   dnorm( mu35, inv.sigma2 )
y40    ~   dnorm( mu40, inv.sigma2 )
```

Interval around μ_{40} will reflect uncertainty concerning fitted parameters

Interval around y_{40} will additionally reflect sampling error σ and uncertainty about σ

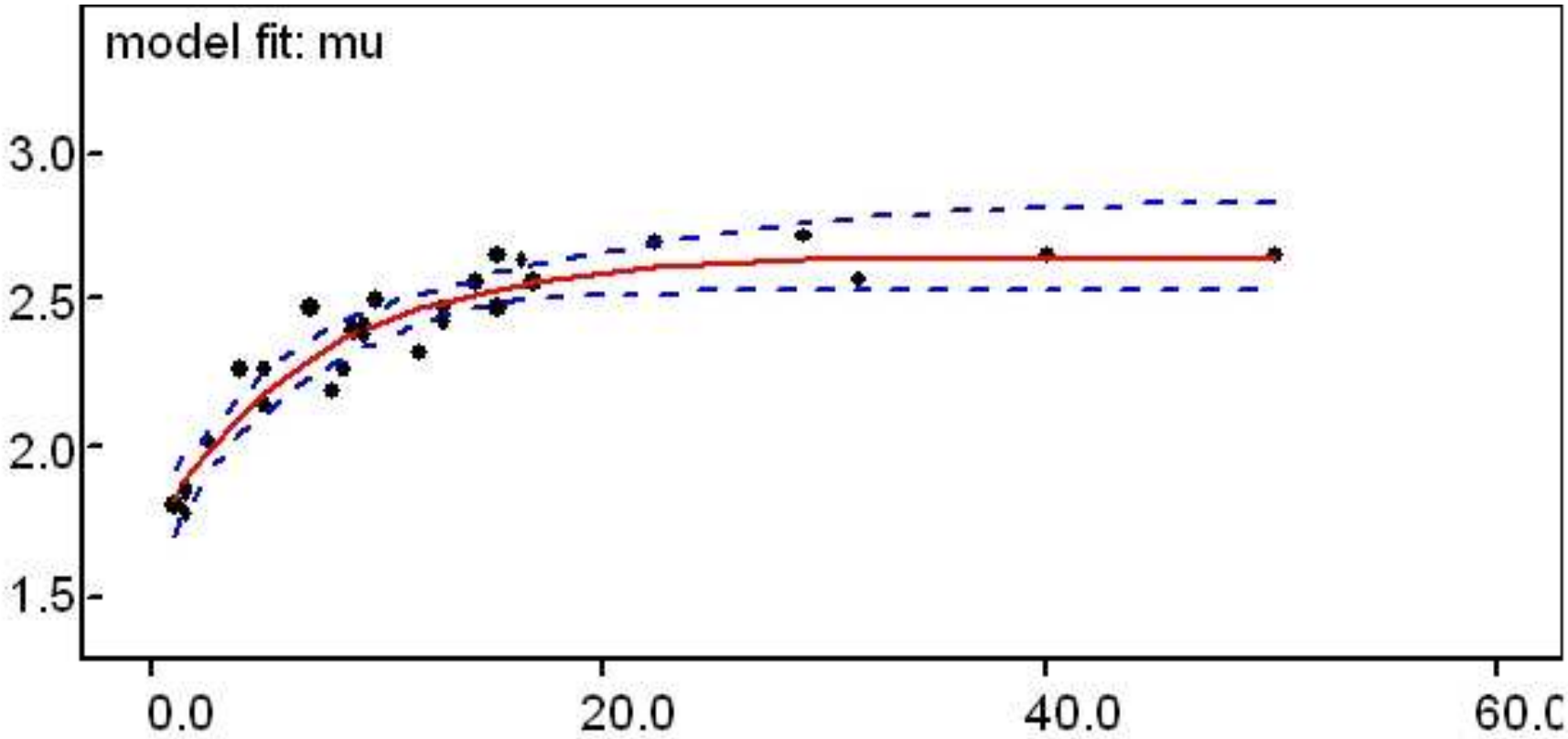
Dugongs: prediction as missing data

Easier to set up as missing data - WinBUGS automatically predicts it

```
list(x = c( 1.0,  1.5,  1.5,  1.5, 2.5,   4.0,  5.0,  5.0,  7.0,
              8.0,  8.5,  9.0,  9.5, 9.5,  10.0, 12.0, 12.0, 13.0,
             13.0, 14.5, 15.5, 15.5, 16.5, 17.0, 22.5, 29.0, 31.5, 35, 40),
       Y = c(1.80, 1.85, 1.87, 1.77, 2.02, 2.27, 2.15, 2.26, 2.47,
             2.19, 2.26, 2.40, 2.39, 2.41, 2.50, 2.32, 2.32, 2.43,
             2.47, 2.56, 2.65, 2.47, 2.64, 2.56, 2.70, NA, NA), N = 29)
```

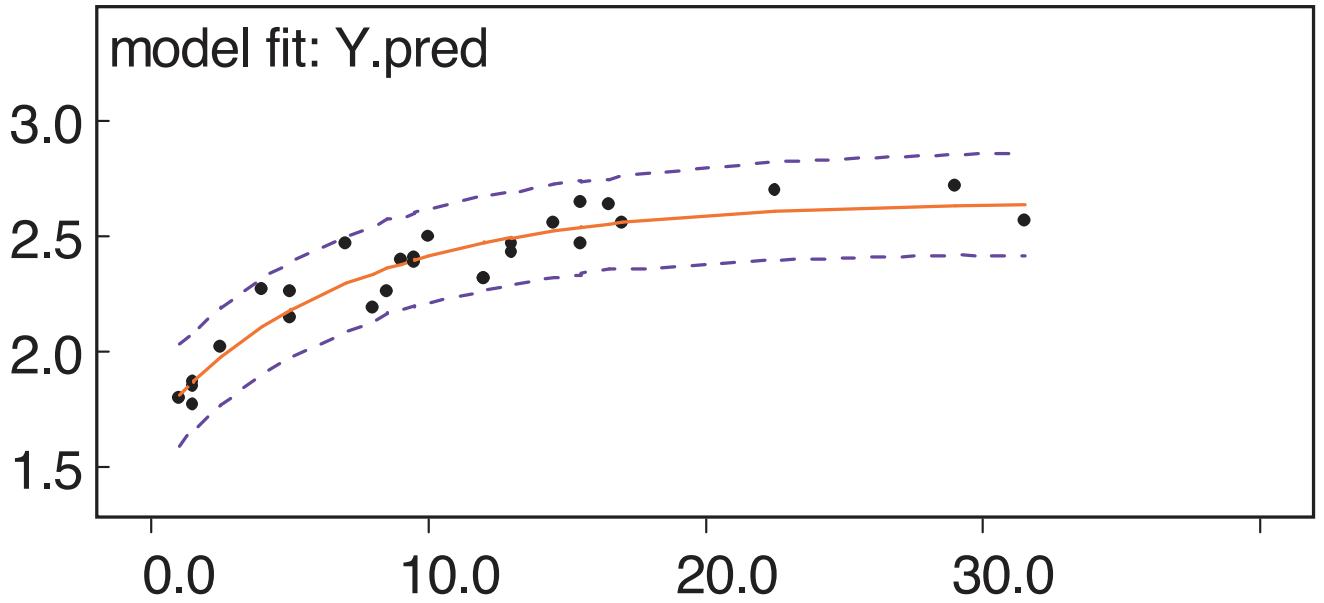
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[28]	2.651	0.07189	0.00423	2.533	2.642	2.815	1001	10000
Y[28]	2.651	0.1228	0.004537	2.415	2.648	2.902	1 001	10000
mu[29]	2.655	0.07825	0.004772	2.533	2.644	2.837	1001	10000
Y[29]	2.653	0.1275	0.005026	2.413	2.649	2.921	1001	10000

Dugongs: projections



Dugongs: prediction as model checking

```
y.pred[i]   ~   dnorm( mu[i], inv.sigma2 )
```



Model comparison

What is the 'deviance'?

- For a likelihood $p(y|\theta)$, we define the deviance as

$$D(\theta) = -2 \log p(y|\theta) \tag{1}$$

- In WinBUGS the quantity `deviance` is automatically calculated, where θ are the parameters that appear in the stated sampling distribution of y

- The full normalising constants for $p(y|\theta)$ are included in `deviance`

- *e.g.* for Binomial data `y[i] ~ dbin(theta[i],n[i])`, the `deviance` is

$$-2 \sum_i \left[y_i \log \theta_i + (n_i - y_i) \log(1 - \theta_i) + \log \begin{pmatrix} n_i \\ r_i \end{pmatrix} \right]$$

What is the 'standardised deviance'?

- In generalised linear models the saturated deviance is (loosely) defined as $D(y)$ - the deviance with the observations substituted for their expectations
- We define the standardised deviance as $D(\theta) - D(y)$
- *e.g.* for Binomial data, Bayesian standardised deviance is

$$-2\sum_i \left[y_i \log \frac{\theta_i}{y_i/n_i} + (n_i - y_i) \log \frac{(1-\theta_i)}{(1-y_i/n_i)} \right]$$

- Just sum of deviance residuals
- This is a random quantity with a posterior distribution
- If model fits the data, expected to have χ^2_I distribution, where I is the dimensionality of θ.
- Can be used as absolute measure of fit
- In WinBUGS you currently need to calculate it yourself

Use of mean deviance as measure of fit

- Dempster (1974) suggested plotting posterior distribution of deviance
- Many authors suggested using posterior mean deviance $\overline{D} = \mathbb{E}[D]$ as a measure of fit
- Invariant to parameterisation of θ
- Robust, generally converges well
- But more complex models will fit the data better and so will have smaller $\overline{D}$
- Need to have some measure of 'model complexity' to trade off against $\overline{D}$

Bayesian model comparison using DIC

- Natural way to compare models is to use criterion based on trade-off between the fit of the data to the model and the corresponding complexity of the model
- Spiegelhalter et al (2002) proposed a Bayesian model comparison criterion based on this principle:

  Deviance Information Criterion, DIC = 'goodness of fit' + 'complexity'

- They measure fit via the deviance

$$D(\theta) = -2 \log L(\text{data}|\theta)$$

- Complexity measured by estimate of the 'effective number of parameters':

$$\begin{aligned} p_D &= E_{\theta|y}[D] - D(E_{\theta|y}[\theta]) \\ &= \overline{D} - D(\overline{\theta}); \end{aligned}$$

  i.e. posterior mean deviance minus deviance evaluated at the posterior mean of the parameters

- The DIC is then defined analagously to AIC as

$$\begin{aligned} \text{DIC} &= D(\overline{\theta}) + 2p_D \\ &= \overline{D} + p_D \end{aligned}$$

  Models with smaller DIC are better supported by the data

- DIC can be monitored in `WinBUGS` from `Inference/DIC` menu

- These quantities are easy to compute in an MCMC run
- Aiming for Akaike-like, cross-validatory, behaviour based on ability to make short-term predictions of a repeat set of similar data.
- Not a function of the marginal likelihood of the data, so *not* aiming for Bayes factor behaviour.
- Do not believe there is any 'true' model.
- p_D is not invariant to reparameterisation.
- p_D can be be negative! (not desirable)
- Alternative to p_D suggested

Could DIC be improved?

- It would be better if WinBUGS used the posterior mean of the 'direct parameters' (eg those that appear in the WinBUGS distribution syntax) to give a 'plug-in' deviance, rather than the posterior means of the stochastic parents.

- Users are free to calculate this themselves: could dump out posterior means of 'direct' parameters in likelihood, then calculate deviance outside WinBUGS or by reading posterior means in as data and checking `deviance` in `node info`

- Lesson: need to be careful with highly non-linear models, where posterior means may not lead to good predictive estimates

- Same problem arises with mixture models

DIC is allowed to be negative - not a problem!

- A probability density $p(y|\theta)$ can be greater than 1 if has a small standard deviation
- Hence a deviance can be negative, and a DIC negative
- Only *differences* in DIC are important: its absolute size is irrelevant
- Suppose observe data (-0.01, 0.01)
- Unknown mean (uniform prior), want to choose between three models with $\sigma = 0.001, 0.01, 0.1$.

	Dbar	Dhat	pD	DIC
y1	177.005	176.046	0.959	177.964
y2	-11.780	-12.740	0.961	-10.819
y3	-4.423	-5.513	1.090	-3.332

- Each correctly estimates the number of unknown parameters.
- The middle model ($\sigma = 0.01$) has the smallest DIC, which is negative.

More on missing data in WinBUGS

Missing response data, assuming missing data mechanism is ignorable

- denote missing observations by `NA` in the data file
- specify response distribution (likelihood) as you would for complete data
- missing data are treated as additional unknown parameters

$\Rightarrow$ WinBUGS will automatically simulate values for the missing observations according to the specified likelihood distribution, conditional on the current values of all relevant unknown parameters

Ignorable missing response data is essentially a prediction problem — see earlier dugongs example

If missing data mechanism is informative

- need explicit model for missing data mechanism
- usually need informative priors on parameters of missing data model as no information in the data
- See Best *et al.* (1996) for one example

Missing covariate data

- denote missing observations by `NA` in the data file
- specify prior distribution for the covariate
  - e.g. if X is a continuous covariate containing some missing values, could specify $X_i \sim \text{Normal}(\mu, \sigma^2)$ or build regression model relating X_i to other observed covariates
  - can then assume vague priors for μ and σ^2; posterior distribution of μ and σ^2 will be informed by the observed part of the vector of X's
- WinBUGS will automatically simulate values from the posterior distribution of the missing covariates (which will depend on the prior for the X's and the likelihood contribution from the corresponding response variable)

Example: Childhood malaria in the Gambia

Diggle et al (2002)

Data:

- 2035 children in 65 villages in the Gambia
- Response: Binary indicator of presence of malarial parasites in blood sample taken from each child
- Covariates include: child's age and use of bed nets, inclusion/exclusion of village from primary health care system and greenness of surrounding vegetation (from satellite information)

Questions of interest include:

- Does sleeping under a bed net reduce risk of malaria?

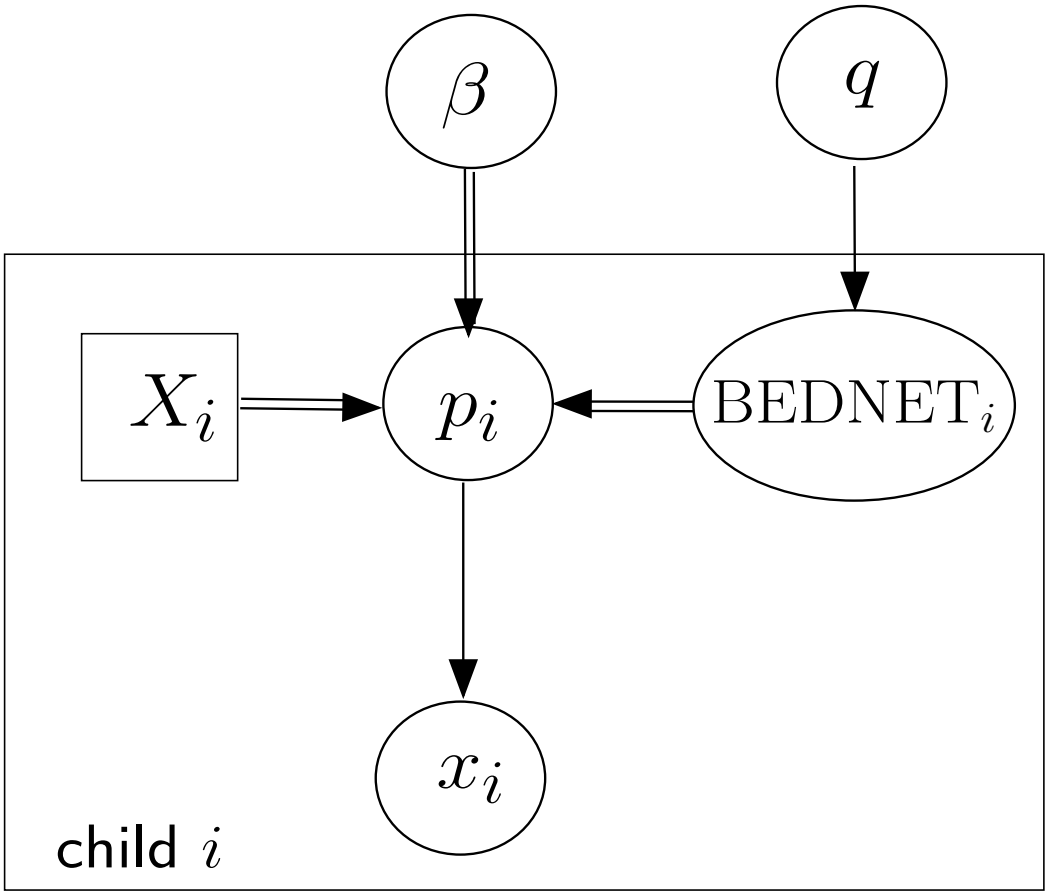
- Here we consider a slightly modified version of Diggle et al's dataset:
  - BEDNET = binary indicator of whether child sleeps under a (treated) bed net
  - Suppose the value of BEDNET is missing for 30% of children
- Consider 2 alternative models for the missing covariate:
  1. Probability of BEDNET = 1 is same for all children *a priori*

$$\begin{aligned} \text{BEDNET}_i &\sim \text{Bernoulli}(q) \\ q &\sim \text{Beta}(1,1) \end{aligned}$$

  2. Probability of BEDNET = 1 depends on whether or not village belongs to primary health care system (PHC)

$$\begin{aligned} \text{BEDNET}_i &\sim \text{Bernoulli}(q_i) \\ \text{logit}q_i &= \gamma_1 + \gamma_2 \text{PHC}_i; \quad (+ \text{ vague priors on } \gamma_1 \text{ and } \gamma_2) \end{aligned}$$

DAG for Model 1



WinBUGS code for model 1

```
model {
  for(i in 1:2035) {
    Y[i] ~ dbern(p[i])
    logit(p[i]) <- alpha + beta.age[AGE[i]] + beta.bednet*BEDNET[i] +
                   beta.green*(GREEN[i] - mean(GREEN[])) + beta.phc*PHC[i]
  }
  # model for missing exposure variable
  for(i in 1:2035) { BEDNET[i] ~ dbern(q)  } # prior model for whether or not child
                                             # i sleeps under treated bednet
  q ~ dbeta(1, 1) # vague prior (uniform) on prob of sleeping under treated bednet

  # vague priors on regression coefficients
  alpha ~ dflat()
  beta.bednet ~ dflat()
  ........etc.......

  # calculate odds ratios of interest
  OR.bednet <- exp(beta.bednet)      # odds ratio of malaria for children using
                                     # treated bednets
  PP.bednet <- step(0.8 - OR.bednet) # probability that using treated bed net
                                     # reduces risk of malaria by at least 20%
}
```

WinBUGS code for model 2

- Replace model for missing exposure variable by

```
# model for missing exposure variable
for(i in 1:2035) {
   BEDNET[i] ~ dbern(q[i])  # prior model for whether or not child i
                            # sleeps under treated bednet
   logit(q[i]) <- gamma[1] + gamma[2]*PHC[i] # allow prob of using treated
                                             # bednet to depend on whether
                                             # or not village belongs to
                                             # primary health care system
}
for(k in 1:2) {   gamma[k] ~ dflat()     }
OR.treated.phc <- exp(gamma[2])              # odds ratio of sleeping under
                                             # treated bednet for children
                                             # living in villages in the PHC
```

Results

	`OR.bednet`		`PP.bednet`	`OR.age2`	
	Mean	95% interval		Mean	95% interval
No missing data	0.57	(0.45, 0.72)	0.99	1.40	(1.06, 1.81)
Model 1	0.66	(0.49, 0.86)	0.93	1.39	(1.06, 1.80)
Model 2	0.64	(0.47, 0.83)	0.95	1.41	(1.06, 1.83)
Single imputation*	0.76	(0.61, 0.95)	0.68	1.40	(1.05, 1.80)
Complete case (exclude all cases with missing data)	0.63	(0.47, 0.83)	0.96	1.70	(1.20, 2.35)

*Imputed using observed proportion of bed net users