

Bayesian Data Analysis

Daniel Paulin & Nicolò Margaritella

University of Edinburgh



Semester 2, 2020/2021

With thanks to Jonathan Gair, Rubén Amorós-Salvador, Ken Newman, Vanda Inácio and
Natalia Bochkina for much of the material

Outline

- 1 Normal linear regression (recall)
- 2 Departures from Normality: Poisson data
- 3 Departures from Normality: Binary or Bernoulli Data
- 4 Basic structure of GLMs
- 5 Frequentist inference for GLMs
- 6 Bayesian inference for the Poisson GLM
- 7 Bayesian inference for the Binomial GLM
- 8 Categorical Covariates
- 9 Computation of posterior and predictive probabilities
- 10 Bayesian GLM: from JAGS to INLA

Section 1

Normal linear regression (recall)

Generalised Linear Models

Normal linear regression (recall)

↪ In Linear Regression we assumed that a quantity of primary interest, a random variable Y , has an expected value that is a linear combination of another quantity, x , or a vector of quantities, \mathbf{x} .

We label Y =response variable, x 's=covariates. (Y =random variable, y =observed value.)

↪ $\mathbb{E}[Y_i] = \mu_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_p x_{i,p}$, $i=1, \dots, n$.

↪ Abbreviated notation for $\mu_i = \mathbf{x}_i^T \boldsymbol{\beta}$, where

$$\mathbf{x}_i = \begin{bmatrix} 1 \\ x_{i,1} \\ \vdots \\ x_{i,p} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

↪ This is a model for the **mean** structure of a random variable Y .

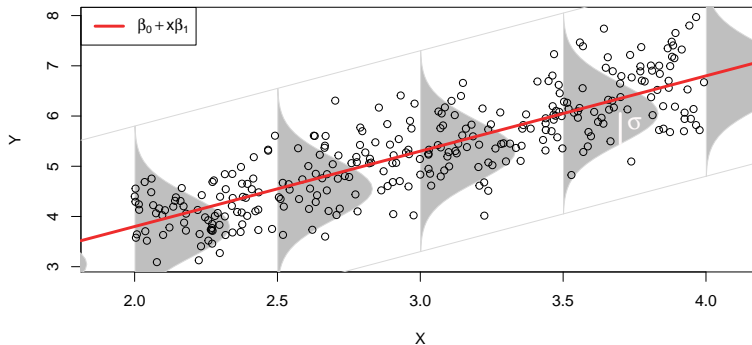
Generalised Linear Models

Normal linear regression

↪ We further assumed a probability distribution of Y_i given \mathbf{x}_i and β :

$$Y_i | \mathbf{x}_i, \beta, \sigma^2 \sim \text{Normal}(\mathbf{x}_i^T \beta, \sigma^2), \quad i = 1, \dots, n$$

With a common variance σ^2 .



Generalised Linear Models

Normal linear regression

↪ We can then estimate β and σ^2 by frequentist (classical) methods, namely maximum likelihood:

$$\begin{aligned}\hat{\beta} &= (X^T X)^{-1} X^T \mathbf{y} \\ \hat{\sigma}^2 &= \frac{\sum_{i=1}^n (y_i - \mathbf{x}_i^T \hat{\beta})^2}{n}\end{aligned}$$

where

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,p} \end{bmatrix}$$

Generalised Linear Models

Normal linear regression

↪ Or by Bayesian estimation of β and σ^2 , with the posterior distribution:

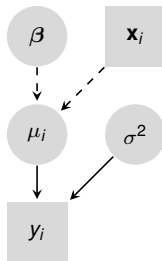
$$\begin{aligned} p(\beta, \sigma^2 \mid \mathbf{y}) &= \frac{p(\beta, \sigma^2, \mathbf{y})}{m(\mathbf{y})} = \frac{f(\mathbf{y} \mid \beta, \sigma^2) \pi(\beta, \sigma^2)}{\int \dots \int f(\mathbf{y} \mid \beta, \sigma^2) \pi(\beta, \sigma^2) d\beta_0 \dots d\sigma^2} \\ &\propto f(\mathbf{y} \mid \beta, \sigma^2) \pi(\beta, \sigma^2) \end{aligned}$$

Generalised Linear Models

Normal linear regression

↪ Example of a full model and directed acyclic graph (DAG) representation:

$$\begin{aligned}
 y_i \mid \mu_i, \sigma^2, \mathbf{x}_i &\sim N(\mu_i, \sigma^2), \quad i = 1, \dots, n \\
 \mu_i &= \mathbf{x}_i^T \boldsymbol{\beta} \\
 \beta_j &\sim N(0, \sigma_{\beta_j}^2), \quad j = 0, \dots, p \\
 \sigma &\sim \text{Unif}(0, u_\sigma)
 \end{aligned}$$



Section 2

Departures from Normality: Poisson data

Generalised Linear Models

Departures from Normality: Poisson data

- An example is the number of “satellite” male crabs that a female Horseshoe crab attracts in addition to a “primary” male crab, y , compared to the width of the carapace (the larger outer shell), x .

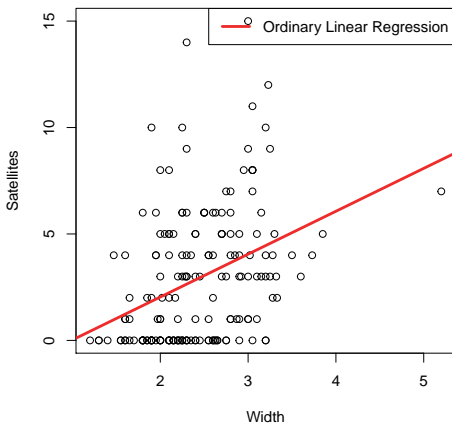


¹pictures NWF; Jane Brockwell

Generalised Linear Models

Departures from Normality: Poisson data

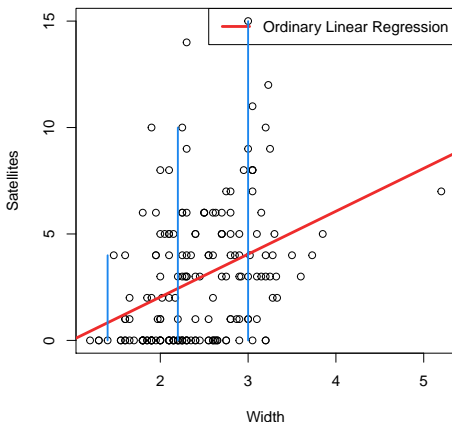
Female Horseshoe Crab's Satellites vs Carapace Width



Generalised Linear Models

Departures from Normality: Poisson data

Female Horseshoe Crab's Satellites vs Carapace Width



- ↪ Discrete response
- ↪ Lower boundary
- ↪ Heteroskedasticity (non-constant variance)

Generalised Linear Models

Departures from Normality: Poisson data

- ↪ We may consider the data to come from a Poisson distribution.
- ↪ Frequentist usual Normal linear regression assumes that the variance of y_i is a constant, i.e., $\text{Var}(y_i) = \sigma^2$, but in a Poisson distribution $\text{Var}(y_i) = E(y_i) = \mu_i$.
- ↪ But modeling μ_i as in the Normal model does not ensure positive mean, i.e.

$$\mu_i = \beta_0 + \beta_1 x_i > 0$$

is not guaranteed.

- ↪ One solution to this is to model the logarithm of μ_i :

$$\ln(\mu_i) = \beta_0 + \beta_1 x_i$$

Then

$$\mu_i = \exp(\beta_0 + \beta_1 x_i) > 0$$

Generalised Linear Models

Departures from Normality: Poisson data

↪ The likelihood of the Poisson model with mean $\mu_i = \exp(\beta_0 + \beta_1 x_i)$ is:

$$f(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x}) \propto \prod_i e^{-\mu_i} \mu_i^{y_i} = \prod_i e^{-\exp(\beta_0 + \beta_1 x_i)} (\exp(\beta_0 + \beta_1 x_i))^{y_i}$$

↪ Under the frequentist paradigm, maximum (log)likelihood with respect to β_0 and β_1 :

$$\ln(f(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x})) \propto \sum_i (-\exp(\beta_0 + \beta_1 x_i) + y_i(\beta_0 + \beta_1 x_i))$$

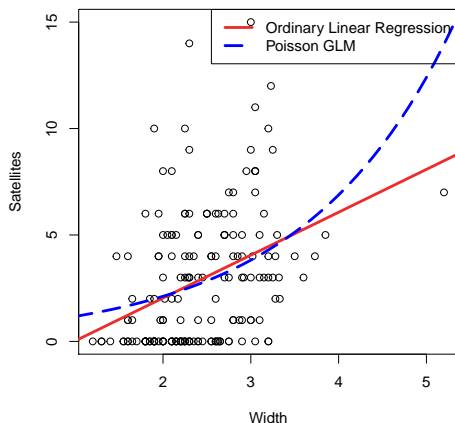
↪ Under the Bayesian paradigm, a prior distribution for β_0 and β_1 must be defined and the posterior of the parameters is found in the usual way:

$$p(\beta_0, \beta_1 \mid \mathbf{y}, \mathbf{x}) \propto f(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x}) \pi(\beta_0, \beta_1)$$

Generalised Linear Models

Departures from Normality: Poisson data

Female Horseshoe Crab's Satellites vs Carapace Width



Section 3

Departures from Normality: Binary or Bernoulli Data

Generalised Linear Models

Departures from Normality: Binary or Bernoulli Data

- ↪ Binary Data: Success or Failure, 1 or 0: $Y \sim \text{Bernoulli}(\theta)$. Thus $\mathbb{E}[Y] = \mu = \theta$.
- ↪ The probability of success, θ , may be a function of covariates.
- ↪ Can you think of some examples of Binary Data and corresponding covariates that could affect θ ?
- ↪ For example, is a randomly chosen Scot in favour of Scottish independence? Let $\theta = \text{Pr}(\text{in favour}) = \text{Pr}(Y = 1)$. What factors might affect θ ?
- ↪ Suppose $x = \text{Age}$ is a factor. What might be wrong with the following model?

$$\mu_i \equiv \theta_i = \beta_0 + \beta_1 \text{Age}_i$$

Generalised Linear Models

Departures from Normality: Binary or Bernoulli Data

- ↪ Again a solution for this is to construct a 1:1 function of μ , $g(\mu)$, as a linear combination of covariates,

$$g(\mu_i) \equiv g(\theta_i) = \eta_i = \beta_0 + \beta_1 \text{Age}_i$$

where the inverse of g maps into the allowable range of θ :

$$0 < [g^{-1}(\beta_0 + \beta_1 \text{Age}_i) = \theta_i] < 1$$

- ↪ A particular function that can be used for this case is the *logit* function, $\ln(x/(1 - x))$:

$$g(\theta_i) = \ln(\theta_i/(1 - \theta_i)) = \beta_0 + \beta_1 \text{Age}_i$$

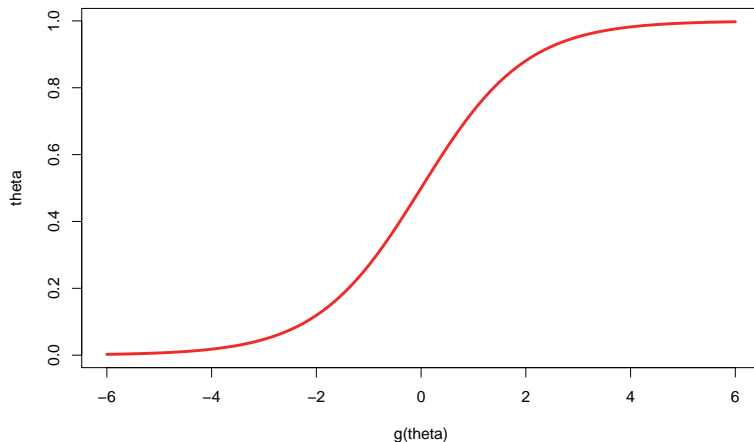
Then

$$0 < [g^{-1}(\beta_0 + \beta_1 \text{Age}_i) \equiv \theta_i = \frac{1}{1 + \exp(-\beta_0 - \beta_1 \text{Age}_i)}] < 1$$

Generalised Linear Models

Departures from Normality: Binary or Bernoulli Data

Inverse-Logit function



Generalised Linear Models

Departures from Normality: Binary or Bernoulli Data

↪ The likelihood of the Bernoulli model with probability of success $\theta_i = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_i)}$ is:

$$f(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x}) = \prod_i \theta_i^{y_i} (1 - \theta_i)^{(1-y_i)} = \prod_i \left(1 + e^{-\beta_0 - \beta_1 x_i}\right)^{-y_i} \left(1 + e^{\beta_0 + \beta_1 x_i}\right)^{-(1-y_i)}$$

↪ Under the frequentist paradigm, maximum (log)likelihood with respect to β_0 and β_1 :

$$\ln(f(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x})) = \sum_i (y_i(\beta_0 + \beta_1 x_i) - \ln(1 + \exp(\beta_0 + \beta_1 x_i)))$$

↪ Under the Bayesian paradigm, a prior distribution for β_0 and β_1 must be defined and the posterior of the parameters is found in the usual way:

$$p(\beta_0, \beta_1, \mid \mathbf{y}, \mathbf{x}) \propto f(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x}) \pi(\beta_0, \beta_1)$$

Section 4

Basic structure of GLMs

Generalised Linear Models

Basic Structure

Generalised Linear Models extend the Normal Linear Model in two ways:

- 1 The distribution of Y is a member of the *Exponential Family Distributions*.
- 2 A 1:1 'link' function of $\mathbb{E}[Y] = \mu$, $g(\mu)$, is a linear combination of covariates, $g(\mu) = \mathbf{x}^T \boldsymbol{\beta}$

$$\begin{aligned} Y &\sim f(y \mid \mu, \dots) \\ g(\mu) &= \eta \\ \eta &= \mathbf{x}^T \boldsymbol{\beta} \end{aligned}$$

Generalised Linear Models

Exponential Family of Distributions

- Exponential Family of Distributions include Normal, Poisson, Bernoulli, Binomial, Multinomial, Exponential, Gamma, Beta, Lognormal, Dirichlet.
- Precise Definition for Single Parameter Case

$$f_Y(y \mid \theta) = h(y) \exp(\eta(\theta) \cdot T(y) - A(\theta))$$

- Precise Definition for Multiple Parameter Case

$$f_Y(y \mid \theta) = h(y) \exp\left(\sum_{i=1}^s \eta_i(\theta) T_i(y) - A(\theta)\right)$$

- $h(y)$ and $T(y)$ are functions of y
- $\eta(\theta)$ and $A(\theta)$ are functions of θ
- More on this topic can be found in Generalised Regression Models course.
- NOTE: Exponential family distributions have conjugate priors!

Generalised Linear Models

Two Examples: Exponential Family of Distributions

$$f_Y(y \mid \theta) = h(y) \exp(\eta(\theta) \cdot T(y) - A(\theta))$$

↪ Poisson(θ)²

$$f_Y(y \mid \theta) = \frac{\exp(-\theta)\theta^y}{y!} = \frac{1}{y!} \exp(\ln(\theta) \cdot y - \theta)$$

↪ Exponential(θ)³

$$f_Y(y \mid \theta) = \theta \exp(-\theta y) = \exp(-\theta \cdot y - (-\ln(\theta)))$$

²where $h(y)=1/y!$; $\eta(\theta)=\ln(\theta)$; $T(y)=y$; and $A(\theta)=\theta$.

³where $h(y)=1$; $\eta(\theta)=-\theta$; $T(y)=y$; and $A(\theta)=-\ln(\theta)$.

Generalised Linear Models

Link Function

$$g(\mu) = \eta = \mathbf{x}_i^T \boldsymbol{\beta}$$

- ↪ The Link Function g is not a unique function.
- ↪ g is usually (but not always) defined so that $g^{-1}(\mathbf{x}_i^T \boldsymbol{\beta})$ yields only allowable values for μ .
 - For example, the “identity” link function $g(\mu) = \mu$ can be specified for a Poisson but that does not ensure that $g^{-1}(\mathbf{x}_i^T \boldsymbol{\beta}) = \mathbf{x}_i^T \boldsymbol{\beta} > 0$. In practice, some constraints must be added to the estimates $\mathbf{x}_i^T \boldsymbol{\beta}$.

Generalised Linear Models

Exponential Family of Distributions and Link Functions

Some examples of link functions for certain distributions:

Distribution	$g(\mu) = \mathbf{x}^t \beta$
Normal(μ, σ^2)	Identity: $g(\mu) = \mu$
Poisson(μ)	Log: $g(\mu) = \ln(\mu)$
Bernoulli(μ)	Logit: $g(\mu) = \ln(\mu/(1 - \mu))$ C-log-log: $g(\mu) = \ln(-\ln(1 - \mu))$ Probit: $g(\mu) = \Phi^{-1}(\mu)$

Probit link: $\Phi(x)$ = CDF for a Standard Normal random variable:

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx$$

$\Phi : \mathcal{R}^1 \rightarrow [0, 1]$ and is 1:1, thus $\Phi^{-1} : [0, 1] \rightarrow \mathcal{R}^1$.

Section 5

Frequentist inference for GLMs

Generalised Linear Models

Frequentist Inference with R

- ↪ We will first discuss how to do frequentist inference for GLMs using the R function `glm`, and then show how to carry out a parallel Bayesian analysis using JAGS.
- ↪ We will look at two examples: a Poisson model (using the Horseshoe crabs data) and a Binomial model
- ↪ The `glm` function has multiple arguments:

```
glm(formula, family = gaussian, data, weights, subset,  
na.action, start = NULL, etastart, mustart, offset, ...)
```

Generalised Linear Models

Frequentist Inference with R

- ↪ `formula` is an expression for the mean structure that is similar to that for normal linear regression

```
formula = y ~ x           # a single covariate,  $b_0 + b_1 x$ 
formula = y ~ x1 + x2     # two covariates,  $b_0 + b_1 x_1 + b_2 x_2$ 
formula = y ~ x1 * x2     # with interac.:  $b_0 + b_1 x_1 + b_2 x_2 + b_3 (x_1 x_2)$ 
```

- ↪ While interpretation of the right-hand side is the same as for `lm`: $\mathbf{x}^T \beta$, the left-hand side is $g(\mu)$ (i.e. the function of $\mathbb{E}[Y \mid \mathbf{x}]$).

```
formula = ``g(\mu)`` ~ x1 + x2
```

Generalised Linear Models

Frequentist Inference with R

→ `family` specifies what exponential family distribution is used:

```
family=gaussian  # the default value
family=poisson
family=binomial
```

Each `family` has an implicit link function, $g(\mu)$ used by default. For example, here are the default link functions made explicit for some distributions:

```
gaussian(link = "identity")
poisson(link = "log")
binomial(link = "logit")
Gamma(link = "inverse")
inverse.gaussian(link = "1/mu^2")
```

→ One can override these defaults by typing an alternative link function between brackets. For example:

```
glm(formula=y ~ x, family=poisson(link=sqrt))
```

Generalised Linear Models

Frequentist Inference: Poisson and Horseshoe crabs

↪ The `crabs` data set has 6 attributes:

```
head(crabs) [, -1]
```

##	Colour	Spine	Weight	Width	Satellites
## 1	med	both.bad	28.3	3.05	8
## 2	dk.med	both.bad	26.0	2.60	4
## 3	dk.med	both.bad	25.6	2.15	0
## 4	dk	one.bad	21.0	1.85	0
## 5	med	both.bad	29.0	3.00	1
## 6	lt.med	one.bad	25.0	2.30	3

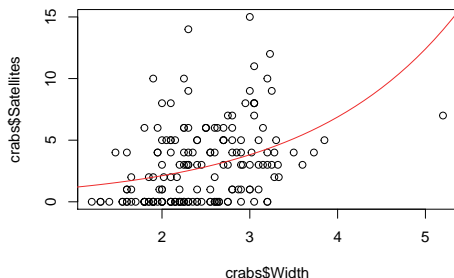
To begin we just focus on modelling the number of satellites as a function of carapace width.

Generalised Linear Models

Frequentist Inference: Poisson and Horseshoe crabs

R code for Poisson model

```
m1 <- glm(formula = Satellites ~ Width, family=poisson(link=log),
           data=crabs )
newdata <- data.frame(Width=seq(1,6,0.1))
pred1 <- predict(m1, newdata=newdata, type="response")
plot(crabs$Width, crabs$Satellites)
lines(newdata$Width, pred1, col="firebrick2")
```



Generalised Linear Models

Frequentist Inference: Poisson and Horseshoe crabs

Different link functions:

```
m1.identity <- glm(formula = Satellites ~ Width,
                    family=poisson(link=identity), data=crabs, start=c(0,1))
m1.sqrt <- glm(formula = Satellites ~ Width,
                family=poisson(link=sqrt), data=crabs )
tabb<-round(rbind(coef(m1),coef(m1.identity),coef(m1.sqrt)),2)
row.names(tabb)<-c("log","identity","sqrt")
tabb
```

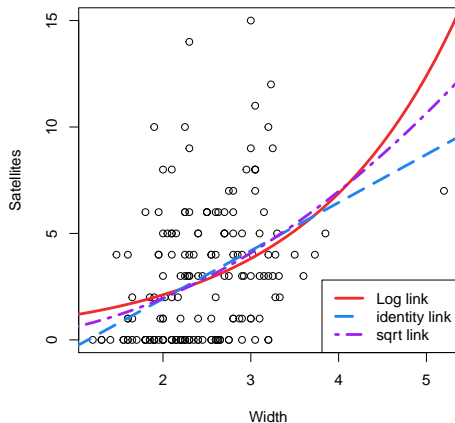
	(Intercept)	Width
log	-0.43	0.59
identity	-2.59	2.26
sqrt	0.16	0.62

→ Different link functions lead to different estimated parameters with very different interpretations.

Generalised Linear Models

Frequentist Inference: Poisson and Horseshoe crabs

Prediction for different link functions:



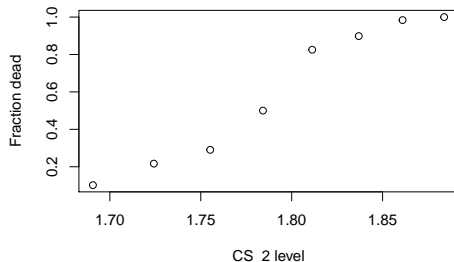
Generalised Linear Models

Frequentist Inference: Binomial and Dead Beetles

- Bliss (1935) presented results of an experiment on the fraction of confused flour beetles (*Tribolium confusum*) that were dead 5 hours after being exposed to gaseous Carbon Disulphide (CS_2). Eight different concentrations of CS_2 were used and about 60 beetles were exposed to each concentration.



- Fraction dead vs exposure:



Generalised Linear Models

Frequentist Inference: Binomial and Dead Beetles

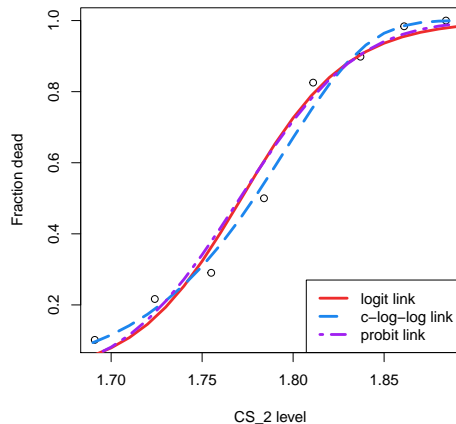
```
n.exposed <- c(59, 60, 62, 56, 63, 59, 62, 60)
CS2.level <- c(1.691, 1.724, 1.755, 1.784, 1.811, 1.837, 1.861, 1.884)
y.dead <- c(6, 13, 18, 28, 52, 53, 61, 60)
beetles <- data.frame(n.exposed, CS2.level, y.dead)
m.logit<- glm(cbind(y.dead, n.exposed-y.dead) ~ CS2.level,
              family=binomial(link=logit), data = beetles)
m.cloglog<- glm(cbind(y.dead, n.exposed-y.dead) ~ CS2.level,
                family=binomial(link=cloglog), data = beetles)
m.probit<- glm(cbind(y.dead, n.exposed-y.dead) ~ CS2.level,
                family=binomial(link=probit), data = beetles)
tabb2<-(round(rbind(coef(m.logit), coef(m.cloglog), coef(m.probit)), 3))
row.names(tabb2)<-c("logit", "cloglog", "probit")
colnames(tabb2)<-c("beta_0", "beta_1 (CS2.level)")
tabb2

##          beta_0 beta_1 (CS2.level)
## logit    -60.740             34.286
## cloglog  -39.522             22.015
## probit   -34.956             19.741
```

Generalised Linear Models

Frequentist Inference: Poisson and Horseshoe crabs

Prediction for different link functions:



Section 6

Bayesian inference for the Poisson GLM

Generalised Linear Models

Bayesian Inference

- As usual, for the Bayesian approach we need to specify priors for the unknown parameters. In the case of GLMs, that is primarily priors for the β s in the model for the expected response via the link function formulation: $g(\mu) = \mathbf{x}^T \beta$.
- The posterior distribution for β :

$$p(\beta \mid \mathbf{y}) \propto f(\mathbf{y} \mid \beta) \pi(\beta)$$

But now the β will appear in the likelihood via the link function. For example, with a Poisson and a single covariate:

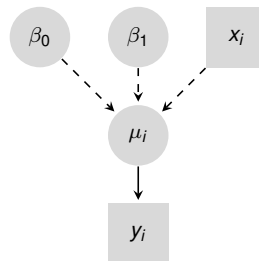
$$\begin{aligned} p(\mathbf{y} \mid \beta_0, \beta_1, \mathbf{x}) &\propto \prod_i e^{-\mu_i} \mu_i^{y_i} = \exp\left(-\sum_{i=1}^n \mu_i\right) \prod_i \mu_i^{y_i} \\ &= \exp\left(-\sum_i g^{-1}(\beta_0 + \beta_1 x_i)\right) \prod_i \left(g^{-1}(\beta_0 + \beta_1 x_i)\right)^{y_i} \end{aligned}$$

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ The Poisson full model for the Horseshoe Crabs and its DAG representation:

$$\begin{aligned}
 y_i \mid \mu_i, \mathbf{x}_i &\sim \text{Poisson}(\mu_i), & i = 1, \dots, n \\
 \log(\mu_i) &= \beta_0 + \beta_1 \text{Width}_i \\
 \beta_j &\sim \text{N}(0, \sigma_{\beta_j}^2), & j = 0, 1
 \end{aligned}$$



Generalised Linear Models

Bayesian Inference: Parameters and Priors

Data Centering

- High correlation between parameters can slow the MCMC. With many correlated parameters, the time necessary to generate enough effective samples for each parameter might be too long.
- Centering the covariates usually helps ($x - \bar{x}$), especially by reducing the dependence of β_0 with the rest of the β s.

Prior Selection

- The selection of “non-informative” priors is highly dependent on the possible values of the response variable, the covariates and the link function.
- Numerical issues might arise with too sparse priors for the β s and extreme initial values (see, for example, the problem of separation in logistic regression⁴)
- On the other hand, too concentrated priors can be too informative or even restrictive to the support of μ .
- A useful diagnostic for priors for GLMs is to examine the induced priors for the mean parameter μ for typical and extreme individuals.
- Always check sensitivity to priors!

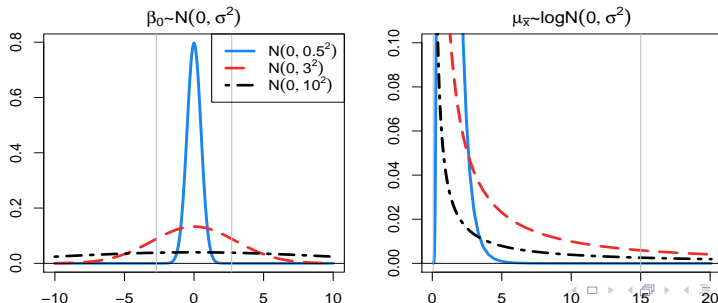
⁴ Gelman A, Carlin JB, Stern HS, et al. Bayesian Data Analysis, Third Edition, CRC Press. Chapter 16, section 16.3

Generalised Linear Models

Bayesian Inference: Parameters and Priors

- $\log(\mu_i) = \beta_0 + \beta_1(x_i - \bar{x})$ or equivalently $\mu_i = e^{\beta_0 + (x_i - \bar{x})\beta_1}$
- If the covariates are centered, β_0 determines the expected value of y_i for a typical individual ($x_i = \bar{x}$).

$$\mu_{\bar{x}} = g^{-1}(\beta_0 + \beta_1(\bar{x} - \bar{x})) = g^{-1}(\beta_0) = e^{\beta_0}$$
- For example, usual values for the number of Horseshoe crab satellites (y_i) are between 0 and 15. With a log-link function, we could try to be uninformative by giving high probability density for the induced variable $\mu_{\bar{x}} = e^{\beta_0}$ on the interval $[1/15, 15]$ and surroundings; or equivalently for β_0 on the interval $[-\ln(15), \ln(15)]$.



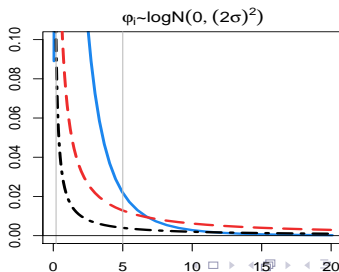
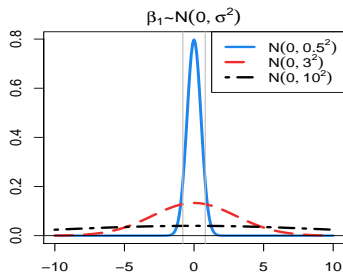
Generalised Linear Models

Bayesian Inference: Parameters and Priors

→ And for the prior of β_1 , with a log-link function, we have that

$$\mu_i = e^{\beta_0 + (x_i - \bar{x})\beta_1} = \mu_{\bar{x}} e^{(x_i - \bar{x})\beta_1} \equiv \mu_{\bar{x}} \varphi_i$$

- Therefore β_1 and x_i determine a factor φ_i that indicates, for any individual i , how many times larger μ_i is with respects of $\mu_{\bar{x}}$.
- Usual values of the width of Horseshoe crabs are between 1 and 5, so $\max |x_i - \bar{x}| \approx 2$. For extreme values of x_i , $\mu_{\bar{x}}$ will be multiplied or divided by a factor $\varphi_i = e^{2\beta_1}$.
- If we consider that an extreme individual can have a mean at most $\varphi_i = 5$ times the mean of a typical individual, then $\beta_1 = \ln(\varphi_i)/|x_{\text{extreme}} - \bar{x}|$ would be in $[-\ln(5)/2, \ln(5)/2]$.



Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

- ↪ Using JAGS, we repeat the analysis of the number of Horseshoe Crab Satellites as a function of carapace width, again assuming a Poisson distribution, in a Bayesian framework.
- ↪ The log link function is used and the covariate will be centred:
$$\log(\mu) = \beta_0 + \beta_1(\text{Width} - \overline{\text{Width}}).$$
- ↪ The priors for the β s will be $\text{Normal}(0, \tau)$, where $\tau=0.01$.
- ↪ For JAGS we need three components: a data block, initial values for the parameters, and a model statement.

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ Data block and Initial Values

```
n <- length(crabs$Obs)
crabs.data <- list(n=n, Satellites=crabs$Satellites, Width=crabs$Width)

crabs.inits <- list(list(beta0=-1, beta1=-1),
                    list(beta0=0, beta1=1),
                    list(beta0=3, beta1=2))
```

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ Model statement

```
crabs.model <- "model {
  # Hyperparameters
  beta.mu.0    <- 0
  beta.tau.0   <- 0.01

  # prior
  beta0 ~ dnorm(beta.mu.0,beta.tau.0)
  beta1 ~ dnorm(beta.mu.0,beta.tau.0)

#Likelihood
  for(i in 1:n) {
    # Note: link function on LHS of fn assignment
    log(mu[i]) <- beta0+beta1*(Width[i]-mean(Width[]))
    Satellites[i] ~ dpois(mu[i])
  }
}"
```

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ Call from R to JAGS:

```
# Run JAGS to the completion of the "adaption" stage
results.crabs.A <- jags.model(file=textConnection(crabs.model),
                             data=crabs.data, inits=crabs.inits,
                             n.chains=3)

# Burn-in of 5000 iterations
update(results.crabs.A, n.iter=5000)

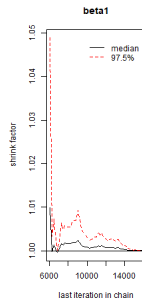
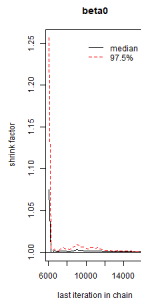
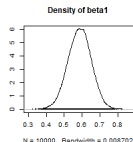
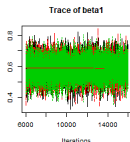
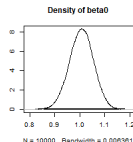
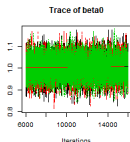
# Longer run for making inferences, assuming chains have converged
results.crabs.B <- coda.samples(results.crabs.A,
                                variable.names=c("beta0", "beta1"), n.iter=10000)
```

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

→ Diagnostics before looking at the summary statistics

```
# Trace plots and density
plot(results.crabs.B)
# Brooks-Gelman-Rubin statistic (want a value near 1)
gelman.plot(results.crabs.B)
```



Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ Other measures of "mixing"- effective sample size given 10,000

```
effectiveSize(results.crabs.B[[1]][, "beta0"])
```

```
##      var1  
## 5254.064
```

```
effectiveSize(results.crabs.B[[1]][, "beta1"])
```

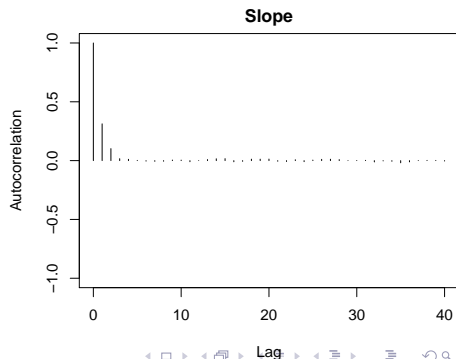
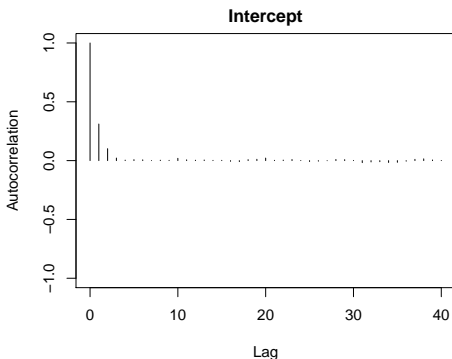
```
##      var1  
## 5236.409
```

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

→ And autocorrelation within each chain.

```
autocorr.plot(results.crabs.B[[1]][, "beta0"], main="Intercept")  
autocorr.plot(results.crabs.B[[1]][, "beta1"], main="Slope")
```



Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ Now look at the posteriors and compare to MLEs:

```
summary(results.crabs.B)
##              Mean          SD   Naive SE Time-series SE
## beta0  1.0067  0.04717  0.0002723      0.0003780
## beta1  0.5864  0.06452  0.0003725      0.0005275
```

```
## vs MLEs #####
##              Estimate Std. Error
## (Intercept)  1.00782    0.04691
## ctr.Width    0.58919    0.06500
```

- ↪ Monte Carlo error: error in the calculated mean due to random sampling. Can be reduced by increasing the chain length
- ↪ A “rule of thumb” (in the OpenBUGS manual) is to have a chain length such that the MC error is no more than 1/20th the size of the standard deviation of the parameter. ⁵

⁵The reduction is not proportional; in this case, double chain length reduces MC error by 28% ↻

Generalised Linear Models

Bayesian Inference: Poisson and Horseshoe Crabs

↪ Interpretation

```
# Joinning all the chains in one data.frame
results.crabs.output <- do.call(rbind.data.frame, results.crabs.B)
#interpretation
cat("E[exp(beta0)] is", round(mean(exp(results.crabs.output$beta0)), 3))

## E[exp(beta0)] is 2.739

cat("E[exp(beta1)] is", round(mean(exp(results.crabs.output$beta1)), 3))

## E[exp(beta1)] is 1.801
```

- ↪ 2.739 is the posterior expected number of satellite male crabs for a female with an average carapace width
- ↪ For a unit increase in the carapace width of a female crab, the expected number of her satellite male crabs increases of 1.801 times, i.e. a 80% increase.

Section 7

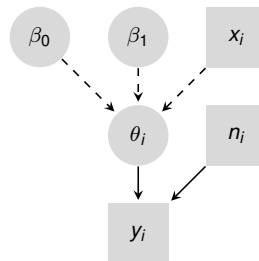
Bayesian inference for the Binomial GLM

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ The Binomial full model for the beetles and its DAG representation:

$$\begin{aligned}
 y_i \mid \mu_i, \mathbf{x}_i &\sim \text{Bin}(\theta_i, n_i), & i = 1, \dots, I \\
 g(\theta_i) &= \beta_0 + \beta_1 \text{Dose}_i \\
 \beta_j &\sim \text{N}(0, \sigma_{\beta_j}^2), & j = 0, 1
 \end{aligned}$$



Generalised Linear Models

Bayesian Inference: Parameters and Priors

- ↪ $g(\theta_i) = \ln(\theta_i/(1 - \theta_i)) = \beta_0 + \beta_1(x_i - \bar{x})$
- ↪ β_0 is the log odds of beetle death for the mean concentration \bar{x} of CS₂
- ↪ β_1 is the log odds-ratio (i.e. the expected change in the log odds) of beetle death for a one-unit increase in the concentration x of CS₂
- ↪ When setting priors we have to consider the minimum distance in x , $x_u - x_l$, for which we think the change of probability can happen. We then set a prior that ensures that $|\beta_1|$ can be larger than 10 over that minimum change distance.

$$-5 \approx \ln(0.005/(1 - 0.005)) = \beta_0 + \beta_1(x_l - \bar{x})$$

$$5 \approx \ln(0.995/(1 - 0.995)) = \beta_0 + \beta_1(x_u - \bar{x})$$

- ↪ For the prior of β_0 , we have to ensure that $|\beta_0|$ can be larger than $(\bar{x} - x_m)|\beta_1|$ ⁶ where $x_m = (x_u + x_l)/2$.
- ↪ When in doubt, try vaguer priors and check sensitivity. If results do not change, you are probably being uninformative enough.

⁶To work out this relationship, start by adding the two equations in this slide

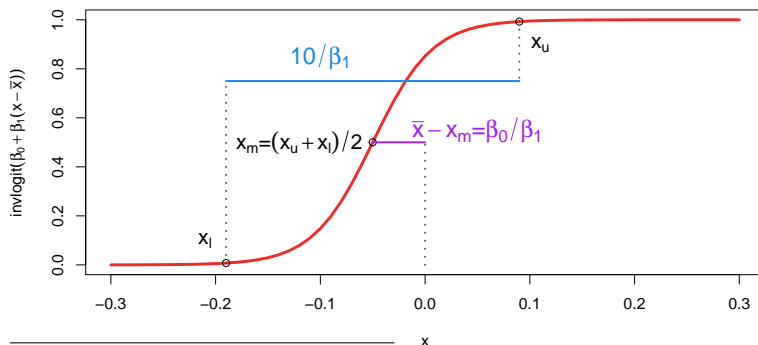
Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ β_1 determines how fast the change in probability happens with respects to x . β_0 determines the horizontal displacement of this change.⁷

$$-5 \approx \text{logit}(0.005) = \beta_0 + \beta_1(x_l - \bar{x})$$

$$5 \approx \text{logit}(0.995) = \beta_0 + \beta_1(x_u - \bar{x})$$



⁷Example with $\beta_0 = 1.75$ and $\beta_1 = 35$

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

- ↪ For the Bayesian analysis of beetle mortality rate, we need three components: data, initial values, and model statement (with priors and likelihood).
- ↪ The election of different initial values allows us to detect possible multimodalities of the model.
- ↪ If one has an idea of the range of possible values for β_0 and β_1 , five initial values corresponding to the 0.05, 0.25, 0.5, 0.75, 0.95 quantiles are one possibility.
- ↪ Another option is to randomly generate the initial values (from the priors or narrower distributions) using a function that generates a list of random initial values.

```
n <- length(n.exposed)
beetles.data <- list(n=n, n.exposed=n.exposed,
                    y.dead=y.dead, CS2.level=CS2.level*100)
beetles.inits <- function() {
  beta0 <- rnorm(1, 0, 1)
  beta1 <- rnorm(1, 0, 1)
  return( list(beta0=beta0, beta1=beta1) )
}
```

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ Model Statement

```
beetles.model <- "model {
  #Hyperparameters
  beta.mu.0    <- 0
  beta.tau.0   <- 0.0001

  # prior
  beta0 ~ dnorm(beta.mu.0,beta.tau.0)
  beta1 ~ dnorm(beta.mu.0,beta.tau.0)

  #Likelihood
  for(i in 1:n) {
    logit(mu[i]) <- beta0+beta1*(CS2.level[i]-mean(CS2.level[]))
    y.dead[i] ~ dbin(mu[i],n.exposed[i])
  }
}"
```

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ Call from R to JAGS

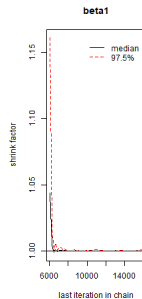
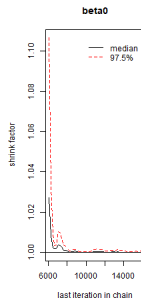
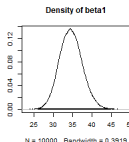
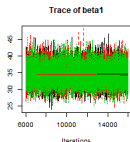
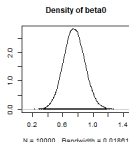
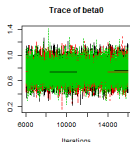
```
# Run JAGS to the completion of the "adaption" stage
results.beetles.A <- jags.model(file=textConnection(beetles.model),
data=beetles.data, inits=beetles.inits, n.chains=3)
# Burn-in of 5000 iterations
update(results.beetles.A, n.iter=5000)
# Longer run for making inferences, assuming chains have converged
results.beetles.B <- coda.samples(results.beetles.A,
variable.names=c("beta0", "beta1"), n.iter=10000)
```

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

→ Trace plots and Brooks-Gelman-Rubin (BGR) statistic

```
# Trace plots and density
plot(results.beetles.B)
# Brooks-Gelman-Rubin statistic (want a value near 1)
gelman.plot(results.beetles.B)
```



Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ Effective sample size

```
effectiveSize(results.beetles.B[[1]][, "beta0"])
```

```
##      var1  
## 4985.016
```

```
effectiveSize(results.beetles.B[[1]][, "beta1"])
```

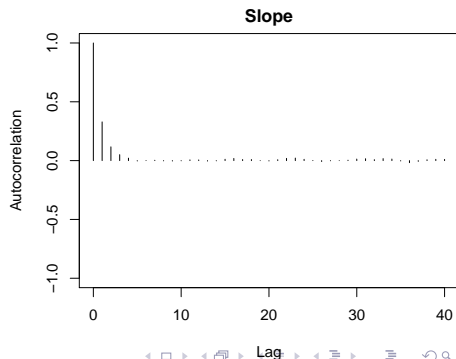
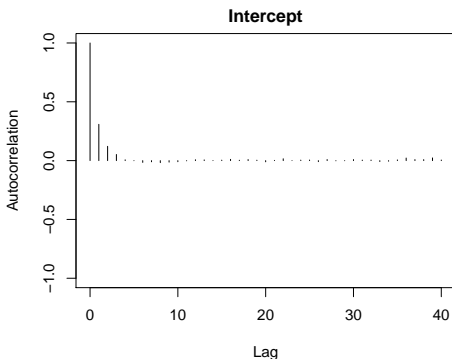
```
##      var1  
## 5046.473
```

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

→ Autocorrelation Function (ACF).

```
autocorr.plot(results.beetles.B[[1]][, "beta0"], main="Intercept")  
autocorr.plot(results.beetles.B[[1]][, "beta1"], main="Slope")
```



Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ Now look at the posteriors and compare to MLEs:

```
summary(results.beetles.B)
```

```
##           Mean      SD Naive SE Time-series SE
## beta0  0.7561 0.140 0.0008065      0.0011320
## beta1  0.3464 0.029 0.0001689      0.0002375
```

```
## vs MLEs #####
## (Intercept)      ctr.CS2
##      0.7474      0.34286
```

↪ The Monte Carlo error (Time-series SE) is much lower than 1/20th the size of the standard deviation of the parameters.

Generalised Linear Models

Bayesian Inference: Binomial and Dead Beetles

↪ Interpretation

```
# Joinning all the chains in one data.frame
results.beetles.output <- do.call(rbind.data.frame, results.beetles.B)
#interpretation
ilogit=function(x){ 1/(1+exp(-x)) }
#
cat("E[ilogit(beta0)] is", round(mean(ilogit(results.beetles.output$beta0))
## E[ilogit(beta0)] is 0.68

cat("E[exp(beta1)] is", round(mean(exp(results.beetles.output$beta1)), 3))
## E[exp(beta1)] is 1.415
```

- ↪ The estimated posterior mean probability of a dead beetle for the average CS2 level (1.79) is 0.68.
- ↪ for a one-unit increase in the CS2 level (*100) we expect a 41.5% increase in the odds of a dead beetle.

Section 8

Categorical Covariates

Generalised Linear Models

Bayesian Analysis: Categorical Covariates

- ↪ Sometimes a covariate is categorical with K possible values. For example, there are $K=4$ crab carapace colours: light medium, medium, dark medium, and dark.
- ↪ To include such covariates in a regression model, one must create $K - 1$ “indicator” variables.
- ↪ The effect of indicator (“dummy”) variables is to shift the model intercept up or down, and there are different ways to construct indicators.
- ↪ Each indicator below takes value=1 for the specific colour is assigned to (Light Medium, Medium and Dark Medium) and value=0 otherwise.

$$g(\mu) = \beta_0 + \beta_1(\text{Width} - \overline{\text{Width}}) + \beta_2 I_{LtMed} + \beta_3 I_{Med} + \beta_4 I_{DKMed}$$

Thus, for a light medium crab, $g(\mu) = \beta_0 + \beta_1(\text{Width} - \overline{\text{Width}}) + \beta_2$ and similarly for the other cases.

- ↪ The category without a dummy variable (Dark) will be the reference category. This means that $\exp(\beta_0)$ is now interpreted as the expected number of satellite male crabs for a female crab with a dark carapace of average width.

Generalised Linear Models

Bayesian Inference: Categorical Covariates

↪ Data block ⁸

```
n <- length(crabs$Obs)
crabs.mult.data <- list(n=n, Satellites=crabs$Satellites,
                        Width=crabs$Width,
                        lt.med.Ind=crabs$Colour=="lt.med",
                        med.Ind=crabs$Colour=="med",
                        dk.med.Ind=crabs$Colour=="dk.med")
```

⁸If we do not provide initial values, the same initial values are set for all the chains. Only to be done if one knows that there is no multimodality in the posterior.

Generalised Linear Models

Bayesian Analysis: Categorical Covariates

↔ Model Statement

```
crabs.mult.model <- "model {
  #Hyperparameters
  beta.mu.0    <- 0
  beta.tau.0   <- 0.001
  # prior
  beta0        ~ dnorm(beta.mu.0,beta.tau.0)
  beta.width   ~ dnorm(beta.mu.0,beta.tau.0)
  beta.lt.med  ~ dnorm(beta.mu.0,beta.tau.0)
  beta.med     ~ dnorm(beta.mu.0,beta.tau.0)
  beta.dk.med  ~ dnorm(beta.mu.0,beta.tau.0)
  #Likelihood
  for(i in 1:n) {
    log(mu[i]) <- beta0+beta.width*(Width[i]-mean(Width[])) +
                  beta.lt.med*lt.med.Ind[i] + beta.med*med.Ind[i] +
                  beta.dk.med*dk.med.Ind[i]
    Satellites[i] ~ dpois(mu[i])
  }
}"
```

Generalised Linear Models

Bayesian Inference: Categorical Covariates

↪ Call from R to JAGS:

```
# Run JAGS to the completion of the "adaption" stage
results.crabs.mult.A <- jags.model(file=textConnection(crabs.mult.model),
                                data=crabs.mult.data,
                                n.chains=3)

# Burn-in of 5000 iterations
update(results.crabs.mult.A, n.iter=5000)

# Longer run for making inferences, assuming chains have converged
results.crabs.mult.B <- coda.samples(results.crabs.mult.A,
                                variable.names=c("beta0", "beta.width", "beta.lt.med",
                                                  "beta.med", "beta.dk.med"),
                                n.iter=10000)
```

Generalised Linear Models

Bayesian Inference: Categorical Covariates

→ Now look at the summary of the posteriors:

```
summary(results.crabs.mult.B)
```

##		Mean	SD	Naive SE	Time-series SE	
##	beta.dk.med	0.0103	0.18561	0.0010716	0.0058468	
##	beta.lt.med	0.4553	0.21197	0.0012238	0.0059372	
##	beta.med	0.2581	0.16901	0.0009758	0.0059801	
##	beta.width	0.5443	0.06831	0.0003944	0.0005911	
##	beta0	0.8159	0.15520	0.0008960	0.0054734	
##						
##		2.5%	25%	50%	75%	97.5%
##	beta.dk.med	-0.34411	-0.1164	0.006755	0.1345	0.3839
##	beta.lt.med	0.04373	0.3120	0.452149	0.5967	0.8727
##	beta.med	-0.06382	0.1417	0.255108	0.3699	0.5993
##	beta.width	0.40846	0.4989	0.544873	0.5907	0.6761
##	beta0	0.49562	0.7127	0.821404	0.9232	1.1067

→ We can immediately spot an increasing pattern towards lighter carapace colours.

Generalised Linear Models

Bayesian Inference: Categorical Covariates

↪ Interpretation

```
# Joining all the chains in one data.frame
results.crabs.mult.output <- do.call(rbind.data.frame, results.crabs.mult.B)
#interpretation
cat("E[exp(beta_LightMed)] is",
    round(mean(exp(results.crabs.mult.output$beta.lt.med)), 3))

## E[exp(beta_LightMed)] is 1.613

#... similarly for the other colours
```

- ↪ A female crab with a light-medium coloured carapace attracts on average 1.613 times more satellite male crabs than a female with dark coloured carapace, *while the other variables are held constant.*
- ↪ Equivalently, there is a 61% increase in the expected number of satellite male crabs attracted by a female crab with light medium carapace compared to a female with dark carapace, *while the other variables are held constant.*

Generalised Linear Models

Bayesian Analysis: Categorical Covariates

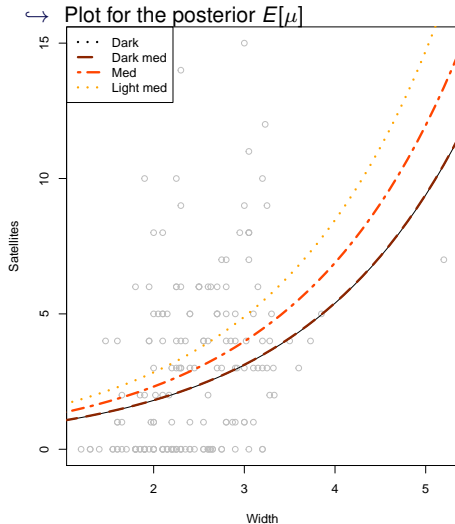
→ Plot for the posterior $E[\mu]$ (R code)

```
x <- seq(1, 6, 0.1)
m.crabs <- summary(results.crabs.mult.B)$statistics[, "Mean"]
plot(crabs$Width, crabs$Satellites, col="gray", xlab="Width", ylab="Satellites")
PostMean_mu=matrix(NA, 4, length(x))
#
for( k in 1:length(x) ){
  PostMean_mu[1,k]=mean(exp(results.crabs.mult.output$beta0+
    results.crabs.mult.output$beta.width*(x[k]-mean(crabs$Width))))

  PostMean_mu[2,k]=mean(exp(results.crabs.mult.output$beta0+
    results.crabs.mult.output$beta.width*(x[k]-mean(crabs$Width))+
    results.crabs.mult.output$beta.dk.med))
#...
}
lines(x, PostMean_mu[1,], lwd=4, lty=3)
lines(x, PostMean_mu[2,], col="orangered4", lwd=3, lty=5)
#...
legend("topleft", c("Dark", "Dark med", "Med", "Light med"), col = c("black",
"orangered4", "orangered", "orange"), lty = c(1,5,6,3), lwd = 3)
```


Generalised Linear Models

Bayesian Analysis: Categorical Covariates



Section 9

Computation of posterior and predictive probabilities

Generalised Linear Models

Bayesian Analysis: Direct calculations using the simulations

→ to easily extract simulation chains

```
library(runjags)
fit.crabs <- as.data.frame(combine.mcmc(results.crabs.mcmc.B))
# or use function do.call as shown in slide 71
```

→ It can be used to calculate probabilities, e.g. $Pr(\beta_{Med} > 0|y)$.

```
mean(fit.crabs$beta.med>0)
```

```
## [1] 0.9388333
```

→ One can also obtain simulations for the predictive distribution $\int f(y^*|\beta)p(\beta|y)d\beta$ by plugging-in posterior samples of β into the likelihood of the new observation y^* , for example $p(y^*|y, Width^* = 3, Colour^* = Med)$:

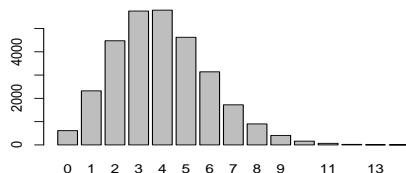
```
mu_star <- exp(fit.crabs$beta0 +
               fit.crabs$beta.width*(3-mean(crabs$Width)) +
               fit.crabs$beta.med)
y_star <- rpois(length(mu_star), mu_star)
```

Generalised Linear Models

Bayesian Analysis: Direct calculations using the simulations

↪ In this way we can plot the distribution $p(y^*|y, \text{Width}^* = 3, \text{Colour}^* = \text{Med})$

```
barplot(table(y_star) )
```



↪ And calculate predictive probabilities such as $Pr(y^* \geq 3|y, \text{Width}^* = 3, \text{Colour}^* = \text{Med})$

```
mean(y_star>=3)
```

```
## [1] 0.7530667
```

Section 10

Bayesian GLM: from JAGS to INLA

Generalised Linear Models

From JAGS to INLA: Categorical Covariates

- We repeat the analysis of the crab data with categorical covariates.
- In INLA, the log link function is the only available option for Poisson likelihood. This is the default link, so it does not have to be specified explicitly
- In INLA, the K-1 indicators are automatically created by the system when we use categorical variables in the regression formula

```
####Crab data with categorical covariates - INLA ####
#
#Creating centered Width covariate
crabs$Width.ctr=crabs$Width-mean(crabs$Width)
#Priors for the regression coefficients.
prior.beta <- list(mean.intercept = 0, prec.intercept = 0.01,
                   mean = 0, prec = 0.01)
#Fitting the model in INLA
#"control.fixed=prior.beta" sets regression coeff. priors.
m2.I <- inla(formula = Satellites ~ Width.ctr+Colour, family="poisson",
data=crabs, control.fixed=prior.beta)
```

Generalised Linear Models

From JAGS to INLA: Categorical Covariates

→ Now look at the summary of the posteriors:

```
summary(m2.I)
```

##		mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
##	(Intercept)	0.828	0.149	0.523	0.832	1.109	0.841	0
##	Width.ctr	0.546	0.068	0.410	0.547	0.678	0.548	0
##	Colourdk.med	0.003	0.180	-0.344	0.000	0.364	-0.004	0
##	Colourlt.med	0.453	0.208	0.045	0.452	0.863	0.452	0
##	Colourmed	0.248	0.163	-0.062	0.244	0.578	0.237	0

→ The summary statistics are essentially identical to what we got from JAGS.

Generalised Linear Models

From JAGS to INLA: Categorical Covariates

- Now we are going to plot the posterior mean of μ , as a function of the width, for a set of equally spaced widths, separately for each 4 colours.
- This can be done by including new rows in the dataset with the response variable (Satellites) set to NA.

```
#Create a sequence of widths at which the posterior means
#of the fitted values is computed
x=seq(1, 6, 0.1)
lx=length(x)

#Create new rows in the dataset for these widths,
#for all 4 colours, with the response Satellites set to NA
newdata <- rbind(data.frame(Width=x, Colour=as.factor("dk"),
                           Satellites=NA), data.frame(Width=x, Colour=as.factor("dk.med"),
                           Satellites=NA), data.frame(Width=x, Colour=as.factor("med"),
                           Satellites=NA), data.frame(Width=x, Colour=as.factor("lt.med"),
                           Satellites=NA))

#Join the new rows with the original dataframe,
#and include the centered width covariate
newdata=rbind(newdata, data.frame(Satellites=crabs$Satellites,
                                  Width=crabs$Width, Colour=crabs$Colour))
newdata$Width.ctr=newdata$Width-mean(crabs$Width)
```


Generalised Linear Models

From JAGS to INLA: Categorical Covariates

- We fit the model in INLA. Note that we need to set **control.predictor = list(compute = TRUE, link=1)**
- **link=1** tells INLA to compute the marginals and means of the fitted values μ_i for the rows where the response is set to NA
- normally, when the response is set to NA, the statistics in **inla.model\$summary.fitted.values** will refer to the linear predictor η_i , and not μ_i

*#whenever we use a model with link function that is not the identity
#and we want to compute the posterior marginal and mean of mu_i
#and not eta_i, we need to set link=1 in control.predictor*

```
m3.I <- inla(formula = Satellites ~ Width.ctr+Colour, family="poisson",
data=newdata, control.fixed=prior.beta,
control.predictor = list(compute = TRUE, link=1),
control.compute=list(config=TRUE))
```

Generalised Linear Models

From JAGS to INLA: Categorical Covariates

- We extract the posterior mean of the fitted values by `m3.I$summary.fitted.values$mean`, and plot them.

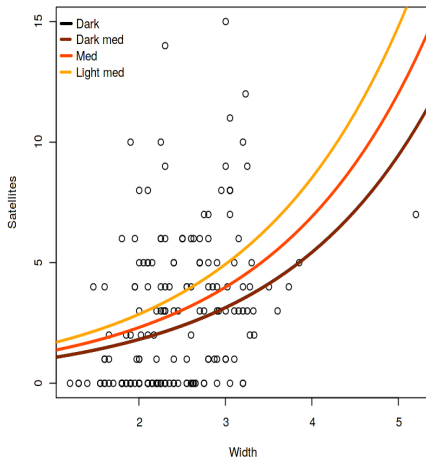
```
#Rows 1:lx contain the results for colour Light med,  
#rows lx+1:2*lx contain the results for colour Med, etc.
```

```
fittedvaluesm=m3.I$summary.fitted.values$mean
```

```
plot(crabs$Width, crabs$Satellites, col="black", xlab="Width",  
     ylab="Satellites")  
legend("topleft", c("Dark", "Dark med", "Med", "Light med"),  
       col = c("black", "orangered4", "orangered", "orange"),  
       lty = c(1,5,6,3), lwd = 3,bty = "n")  
lines(x, fittedvaluesm[1:lx], col="black",lwd=3)  
lines(x, fittedvaluesm[(lx+1):(2*lx)], col="orangered4", lwd=3, lty=5)  
lines(x, fittedvaluesm[(2*lx+1):(3*lx)], col="orangered", lwd=3, lty=6)  
lines(x, fittedvaluesm[(3*lx+1):(4*lx)], col="orange", lwd=3, lty=3)
```

Generalised Linear Models

From JAGS to INLA: Categorical Covariates



Generalised Linear Models

From JAGS to INLA: posterior and predictive probabilities

↪ We look at the posterior and posterior predictive probabilities

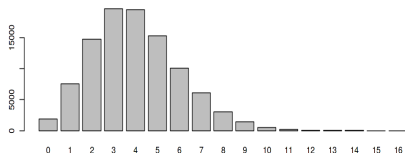
```
#We have already included the Width=3, Colour = Med
#in the newdata dataframe in row 51*2+21=123
#
#We only include this single linear predictor
#and the regression coefficient for colour med
#in the output of inla.posterior.sample by including
#selection=list(Predictor=123,Colourmed=1)
#
nbsamp=100000
crab.samples=inla.posterior.sample(n=nbsamp, result=m3.I,
                                   selection=list(Predictor=123,Colourmed=1))
#POSTERIOR PROBABILITY OF beta_MED > 0
#
beta.med.samples=inla.posterior.sample.eval(function(...) {Colourmed},
crab.samples)
mean(beta.med.samples>0)
#0.93691
```

Generalised Linear Models

From JAGS to INLA: posterior and predictive probabilities

↪ and $P(y^*|y, \text{width}^* = 3, \text{Colour}^* = \text{Med})$

```
predictor.samples=inla.posterior.sample.eval(function(...) {Predictor},
crab.samples)
post.pred.samples=rpois(n=nbsamp, lambda=exp(predictor.samples))
#
barplot(table(post.pred.samples))
#
mean(post.pred.samples>=3)
#0.75814
```



The R code containing all of the analyses in Lecture 4 using JAGS and INLA is available on [Learn](#).

Generalised Linear Models

Bayesian Analysis: further readings on Bayesian GLMs

- ↪ Reich BJ, Ghosh SK. Bayesian Statistical Methods. CRC press, 2019. Chapter 4, Section 4.3 and subsections.
- ↪ Gelman A, Carlin JB, Stern HS, et al. Bayesian Data Analysis, Third Edition. CRC press, 2014. Chapter 16, Introduction, Section 16.1, 16.2 "Offsets" subsection, and 16.4.