

Advanced Probabilistic Modelling

Bayesian regression models

Irantzu Barrio Beraza

- ① Statistical Bayesian Modelling
- ② Bayesian Inference in Linear Models
- ③ Bayesian Inference in Generalized Linear Models

Statistical Bayesian Modelling

So far we have done inference on the parameter of models with a single variable of interest unrelated to other variables:

- Proportion p of sick people in a city? $X \sim Ber(p)$
- Average number λ of patients admitted to a hospital during a hour? $Z \sim Po(\lambda)$
- Mean weight μ of a fish? $Y \sim N(\mu, \sigma)$

We have made inference about one or two of the parameters of univariate distributions.

But, explaining real life needs more complex models.

- In general, a model is a small-scale representation of reality:
 - > either a description of reality,
 - > a tool to understand the reality or
 - > a tool for predicting future behavior.

-The best feature of a model is to be as accurate as possible in their task of representing reality

- Statistics allows us to incorporate the variability present in real life in our models through randomness.
- Still: “essentially, all models are wrong, but some of them are useful” (Box, 1987).
- “The problem formulation is more essential than its own solution, which may simply be a mathematical or experimental skill” (Albert Einstein).

- Once we have the data collected, we have to model.
- When we model a real problem we ask ourselves
 - > what do we want to explain?
 - > and based on what?
- This classifies the variables into:
 - > **Response variables:** the ones we want to explain
 - > **Explanatory** (or independent, or predictor) variables: those that serve to explain the response variables.

- How do we build a model that reflects the situation we want to analyze?
- Most statistical models have a structure of the type:
 - > **Response variable** to be explained.
 - > A **systematic component** that contains the “general” information of the system under study, and is expressed as a **combination of explanatory variables** in the form of a parametric equation. It thus indicates how the explanatory variables affect the response.
 - > A random component that reflects the intrinsic variability in each particular situation (in each data).

- Depending on the type of variable, the explanatory variables are:
 - > Qualitative \Rightarrow Factors (with their corresponding “levels”)
 - **Fixed effects** (if factor levels are preset in advance: Sex)
 - **Random effects** (if the factor levels are a random sample of the possible levels of that factor: person)
 - > Quantitative \Rightarrow Covariates

- Often the systematic component is expressed as a lineal combination (but can also be non-linear).
- If the response variable is normal and the relationship is linear, we have a **linear model**
- Example: explain a person's weight by its height and age.

$$\mu_Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

- Interpretation is not so clear when the explanatory variables are discrete or quantitative but we observe them categorized. In those cases, *dummy* variables are introduced, allowing us to describe unique and numerically the status of a discrete variable.
- If the response variable belongs to the exponential family (binomial, Bernoulli, gamma, Poisson, etc.) we have a **generalized linear model**.

Bayesian Inference in Linear Models

- When modelling a real practical situation, we are often faced with the problem of explaining a continuous (normal) response variable as a function of one or several covariates (Linear Models), one or several factors that explain it (ANOVA), or the situation in which the explanatory variables are both factors and covariates (ANCOVA).
- We can unify all situations as a **Regression Linear Model**, introducing the factors as indicator variables.
- We ask ourselves if each of the covariates (including the indicators that mark the levels of the qualitative variable) is part of the model, and then we look for the best model that explains our response variable among all the possible combinations of covariates.

The complete model with covariates, factors (in the form of indicator variables) and interactions has the form:

$$\begin{aligned}
 Y_i = & \beta_0 + \beta_1 X_{i1} + \dots + \beta_p X_{ip} \\
 & + \gamma_1 D_{i1} + \dots + \gamma_q D_{iq} + \\
 & + \delta_{11} X_{i1} D_{i1} + \dots + \delta_{1q} X_{i1} D_{iq} + \\
 & + \dots + \\
 & + \delta_{p1} X_{ip} D_{i1} + \dots + \delta_{pq} X_{ip} D_{iq} + \\
 & + \epsilon_i; \quad \epsilon_i \sim N(0, \sigma) \quad \forall i = 1, \dots, n \\
 Y_i \sim & N(\mu_i = \mathbf{X}\beta, \sigma) \quad \forall i = 1, \dots, n
 \end{aligned}$$

- Objective: to estimate the parameters $\theta = (\beta, \sigma^2)$, being

$$\beta = (\beta_0, \beta_1, \dots, \beta_p, \gamma_1, \dots, \gamma_q, \delta_{11}, \dots, \delta_{pq})$$

- At the inferential level in both classical and Bayesian there are analytical solutions for the parameter estimators.

- To make inference about the parameters of the linear model:
 - > **Data information via the Likelihood:** $P(\mathbf{y}, \mathbf{x}|\theta)$.
 - > The **priori distribution of the parameters:** $P(\theta)$.
 - > The **posterior distribution of the parameters** via Bayes' Theorem:

$$P(\theta|\mathbf{y}, \mathbf{x}) = \frac{P(\mathbf{y}, \mathbf{x}, \theta)}{P(\mathbf{y}, \mathbf{x})} = \frac{P(\theta)P(\mathbf{y}, \mathbf{x}|\theta)}{P(\mathbf{y}, \mathbf{x})} = \frac{P(\theta)P(\mathbf{y}, \mathbf{x}|\theta)}{\int P(\theta)P(\mathbf{y}, \mathbf{x}|\theta)d\theta}$$

- The information provided by the experiment

$$\mathbf{y} = (y_1, \dots, y_n) \quad \text{with } Y_i \sim N(\mu_i, \sigma),$$

and the relationship between the response variable and the covariates $\mu_i = \mathbf{X}\beta$ can be expressed through the likelihood:

$$l(\beta, \sigma^2) = P(\mathbf{y}, \mathbf{x}|\theta) = (2\pi\sigma^2)^{-n/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{x}\beta)' (\mathbf{y} - \mathbf{x}\beta) \right\}.$$

- We can use an **uninformative improper prior** distribution (indicating little or no knowledge) about the parameters:

$$P(\beta, \sigma^2) \propto \underbrace{1 \times 1 \times \dots \times 1}_{\text{length of the vector of parameters } \beta} \times \frac{1}{\sigma^2}.$$

- Then, the **posterior distribution** of the parameters is proportional to

$$\begin{aligned}P(\beta, \sigma^2 | \mathbf{y}, \mathbf{X}) &\propto P(\beta, \sigma^2) P(\mathbf{y}, \mathbf{X} | \beta, \sigma) \\&\propto \frac{1}{\sigma^2} (2\pi\sigma^2)^{-n/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) \right\}\end{aligned}$$

As we have seen before, we can obtain the posterior distribution of all parameters:

$$P(\beta, \sigma^2 | \mathbf{y}, \mathbf{x}) = P(\beta | \mathbf{y}, \mathbf{x}, \sigma^2) P(\sigma^2 | \mathbf{y}, \mathbf{x})$$

where,

$$\begin{aligned} P(\beta | \mathbf{y}, \mathbf{x}, \sigma^2) &= N_k(\hat{\beta}, (\mathbf{X}'\mathbf{X})^{-1}\sigma^2) \\ P(\sigma^2 | \mathbf{y}, \mathbf{x}) &= Inv - \chi^2(n - k, \hat{\sigma}^2) \end{aligned}$$

being,

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y} \\ \hat{\sigma}^2 &= \frac{1}{n-k}(\mathbf{y} - \mathbf{X}\hat{\beta})'(\mathbf{y} - \mathbf{X}\hat{\beta}) \end{aligned}$$

where k = length of the vector of parameters β .

- We consider a study on the extinction of birds (Ramsey and Schafer, 1997; Pimm et al. 1988)
- Measurements on breeding pairs of land-bird species were collected from 16 islands around Britain over the course of several decades.
- For each species, the dataset contains `TIME`, the average time of extinction on the islands where it appeared, `NESTING`, the average number of nesting pairs, `SIZE`, the size of the species (large or small), and `STATUS`, the migratory status of the species (migrant or resident).
- The objective is to fit a model that describes the variation in the time of extinction of the bird species in terms of the covariates `NESTING`, `SIZE`, and `STATUS`.
- This dataset is available as `birdextinct` in the `LearnBayes` package.

```
library(LearnBayes)
data(birdextinct)
summary(birdextinct[,2:5])
```

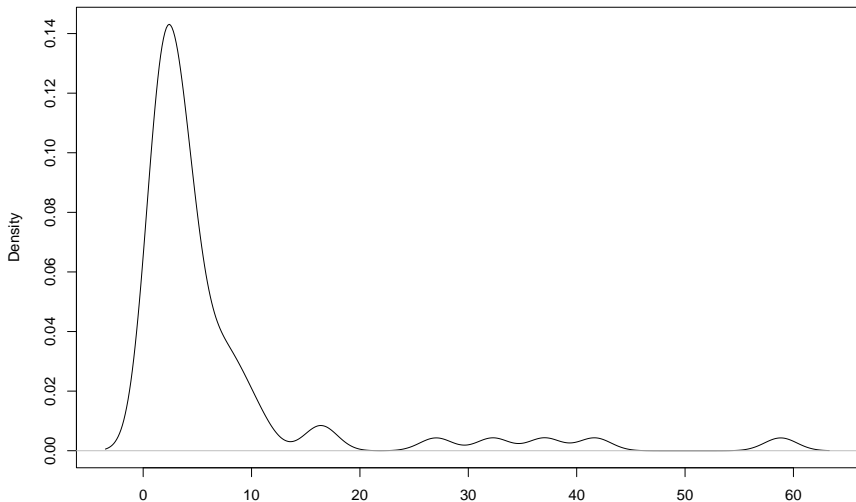
##	time	nesting	size	status
##	Min. : 1.000	Min. : 1.000	Min. :0.0000	Min. :0.0000
##	1st Qu.: 1.907	1st Qu.: 1.448	1st Qu.:0.0000	1st Qu.:0.0000
##	Median : 3.180	Median : 2.750	Median :1.0000	Median :1.0000
##	Mean : 6.957	Mean : 3.417	Mean :0.5484	Mean :0.6935
##	3rd Qu.: 6.989	3rd Qu.: 4.670	3rd Qu.:1.0000	3rd Qu.:1.0000
##	Max. :58.824	Max. :11.620	Max. :1.0000	Max. :1.0000

Extinction of Birds (III)

| 20

```
plot(density(birdextinct$time))
```

density.default(x = birdextinct\$time)



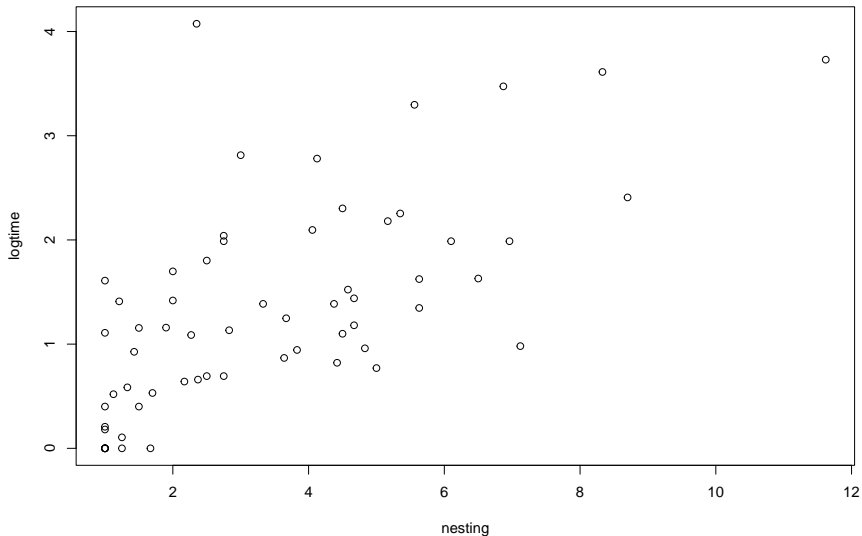
```
birdextinct$logtime <- log(birdextinct$time)
birdextinct$size <- factor(birdextinct$size, levels = c(0,1),
                           labels=c("small", "large"))
birdextinct$status <- factor(birdextinct$status, levels = c(0,1),
                             labels=c("migrant", "resident"))
summary(birdextinct[,4:6])
```

```
##      size      status      logtime
## small:28  migrant :19  Min.      :0.0000
## large:34  resident:43  1st Qu.:0.6455
##                                     Median :1.1569
##                                     Mean    :1.3284
##                                     3rd Qu.:1.9413
##                                     Max.    :4.0746
```

Extinction of Birds (V)

| 22

```
attach(birdextinct)  
plot(nesting, logtime)
```



```
tapply(logtime, status, mean)
```

```
##   migrant   resident  
## 0.8000648 1.5617959
```

```
tapply(logtime, size, mean)
```

```
##    small    large  
## 1.714829 1.010096
```

Least-squares fit for SIZE (frequentist approach)

| 24

```
fit <- lm(logtime ~ size, x=TRUE, y=TRUE)
summary(fit)

##
## Call:
## lm(formula = logtime ~ size, x = TRUE, y = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7148 -0.6086 -0.1903  0.3796  2.7196
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7148     0.1790   9.581 1.05e-13 ***
## sizelarge    -0.7047     0.2417  -2.916  0.00498 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9471 on 60 degrees of freedom
## Multiple R-squared:  0.1241, Adjusted R-squared:  0.1095
## F-statistic: 8.502 on 1 and 60 DF,  p-value: 0.004983
```


Joint posterior distribution of all parameters $\beta = (\beta_0, \beta_1)$:

$$P(\beta, \sigma^2 | \mathbf{y}, \mathbf{x}) = P(\beta | \mathbf{y}, \mathbf{x}, \sigma^2) P(\sigma^2 | \mathbf{y}, \mathbf{x})$$

where,

$$\begin{aligned} P(\beta | \mathbf{y}, \mathbf{x}, \sigma^2) &= N_k(\hat{\beta}, (\mathbf{X}'\mathbf{X})^{-1}\sigma^2) \\ P(\sigma^2 | \mathbf{y}, \mathbf{x}) &= Inv - \chi^2(n - k, \hat{\sigma}^2) \end{aligned}$$

being,

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y} \\ \hat{\sigma}^2 &= \frac{1}{n-k}(\mathbf{y} - \mathbf{X}\hat{\beta})'(\mathbf{y} - \mathbf{X}\hat{\beta}) \end{aligned}$$

where $k = 2$.

Computing the joint posterior distribution of β and σ in R

| 26

```
S2=sum(fit$residual^2)/fit$df.residual
sqrt(S2) # residual standard error: summary(fit)$sigma

## [1] 0.9470987

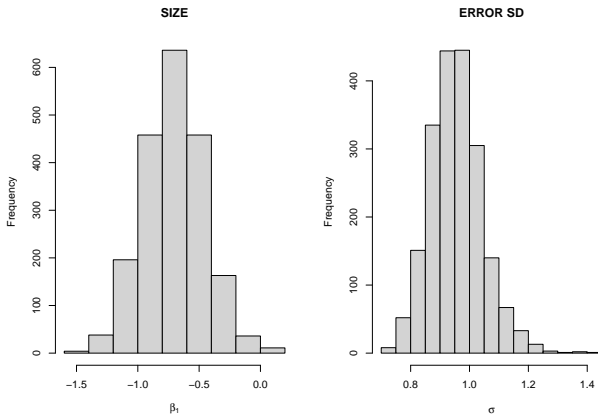
# Simulate from the decomposition of the joint
library("extraDistr")
sigma.sim <- rinvchisq(1, nu=fit$df.residual, tau=S2 )
vbeta <- vcov(fit)/S2
beta.sim <- rmnorm(1, mean=fit$coef, varcov=vbeta*sigma.sim)
```

Joint posterior distribution of β and σ

| 27

We can sample from the joint distribution using `blinreg` function:

```
theta.sample <- blinreg(fit$y, fit$x, 2000)
par(mfrow=c(1,2))
hist(theta.sample$beta[,2],main="SIZE", xlab=expression(beta[1]))
hist(theta.sample$sigma,main="ERROR SD", xlab=expression(sigma))
```



```
apply(theta.sample$beta,2,quantile,c(.025,.5,.975))
```

```
##          X(Intercept) Xsizelarge
## 2.5%          1.356801 -1.1841607
## 50%          1.716315 -0.7147859
## 97.5%         2.068997 -0.2297954
```

```
quantile(theta.sample$sigma,c(.025,.5,.975))
```

```
##          2.5%          50%          97.5%
## 0.7950566 0.9509763 1.1560229
```

Study the effect of the covariate STATUS

Species multiple bayesian linear regression

| 30

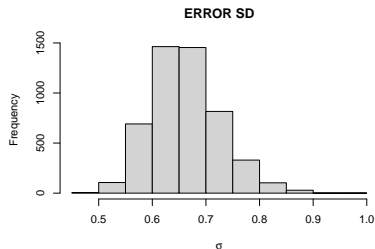
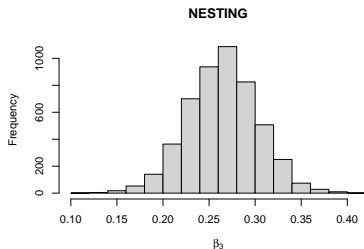
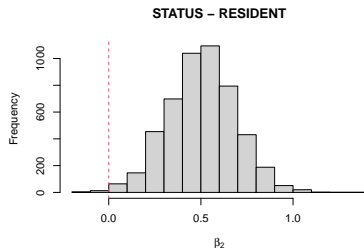
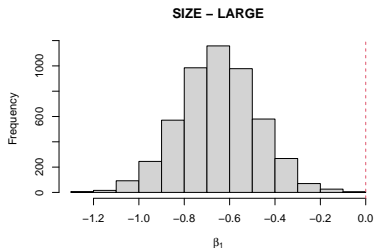
```
fit.2 <- lm(logtime ~ size + status + nesting, x=TRUE, y=TRUE)
summary(fit.2)
```

```
##
## Call:
## lm(formula = logtime ~ size + status + nesting, x = TRUE, y = '
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8410 -0.2932 -0.0709  0.2165  2.5167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.43087    0.20706   2.081 0.041870 *
## sizelarge     -0.65220    0.16667  -3.913 0.000242 ***
## statusresident  0.50417    0.18263   2.761 0.007712 **
## nesting        0.26501    0.03679   7.203 1.33e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
theta.multi.sample <- blinreg(fit.2$y, fit.2$x, 5000)
par(mfrow=c(2,2))
hist(theta.multi.sample$beta[,2],
      main="SIZE - LARGE", xlab=expression(beta[1]))
hist(theta.multi.sample$beta[,3],
      main="STATUS - RESIDENT", xlab=expression(beta[2]))
hist(theta.multi.sample$beta[,4],
      main="NESTING", xlab=expression(beta[3]))
hist(theta.multi.sample$sigma,
      main="ERROR SD", xlab=expression(sigma))
```

Species multiple bayesian linear regression

| 32




```
apply(theta.multi.sample$beta, 2, quantile, c(.025, .975, 0.5))
```

##	X(Intercept)	Xsizelarge	Xstatusresident	Xnesting
## 2.5%	0.0110419	-0.9934889	0.1494833	0.1889342
## 97.5%	0.8449834	-0.3160767	0.8649601	0.3385768
## 50%	0.4315846	-0.6503806	0.5072949	0.2654665

- To fit a Bayesian linear regression we can also use the function `stan_glm` from the `rstanarm` package.
- See for further reference (Muth et al, 2018).

- Some of the arguments for this function are the following:
 - > **family**: by default this function uses the Gaussian distribution as we do with the classical glm function to perform lm model.
 - > **prior**: the prior distribution for the regression coefficients, **by default the normal prior is used**. There are subset of functions used for the prior provided by rstanarm. If we want a flat uniform prior we set this to NULL.
 - > **prior_intercept**: prior for the intercept, can be normal, student t , or Cauchy. If we want a flat uniform prior we set this to NULL.
 - > **prior_aux**: prior fo auxiliary parameters such as the error standard deviation for the Gaussian family.
 - > **algorithm**: The estimating approach to use. The default is sampling MCMC.
 - > **iter**: is the number of iterations if the MCMC method is used, the default is 4000.

Extinction of birds with stan_glm

| 36

```
library(rstanarm)
bayes.lm <- stan_glm(logtime ~ size + status + nesting,
                     prior=NULL, prior_intercep=NULL,
                     prior_aux=NULL, seed=111, data=birdextinct)
```

Extinction of birds with stan_glm

| 37

```
summary(bayes.lm)
```

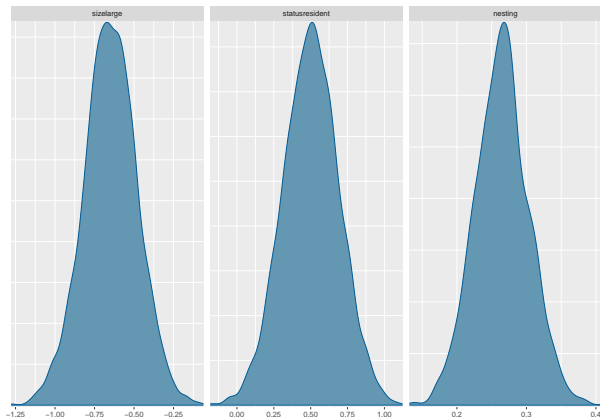
```
##
## Model Info:
## function:      stan_glm
## family:        gaussian [identity]
## formula:       logtime ~ size + status + nesting
## algorithm:     sampling
## sample:        4000 (posterior sample size)
## priors:        see help('prior_summary')
## observations:  62
## predictors:    4
```

```
##
## Estimates:
##              mean    sd  10%   50%   90%
## (Intercept)   0.4    0.2  0.2   0.4   0.7
## sizelarge    -0.6    0.2 -0.9  -0.7  -0.4
## statusresident 0.5    0.2  0.3   0.5   0.7
## nesting       0.3    0.0  0.2   0.3   0.3
## sigma         0.7    0.1  0.6   0.7   0.8
##
```

Extinction of birds with `stan_glm`

| 38

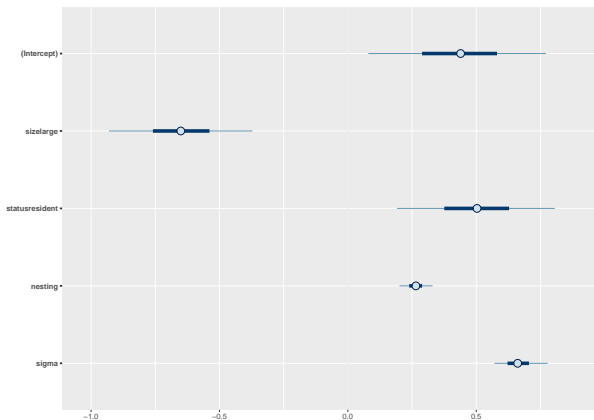
```
library(bayesplot)
mcmc_dens(bayes.lm,
  pars = c("sizelarge", "statusresident", "nesting"))
```



Posterior credible intervals

| 39

```
plot(bayes.lm)
```



Repeat the analysis considering a normal prior for β .

Help: <https://mc-stan.org/rstanarm/articles/priors.html>

Bayesian Inference in Generalized Linear Models

- Generalized Linear Models (GLM) **extend the linear regression model** in order to accommodate:
 - > non-normal responses, e.g. binomial data, frequency data, etc.
 - > and transformation to linearity
- Well known models are logistic regression, log-linear models for frequency tables, Poisson regression, Gamma regression, etc.
- The conceptual advantage is that **many data analytic problems with non-normal data are reduced to regression modelling.**

A Generalized linear model consists of:

- A set of Y_1, \dots, Y_n random variables (response) independent and identically distributed inside the Exponential family (https://en.wikipedia.org/wiki/Exponential_family).
- A set of explanatory variables X_1, X_2, \dots, X_p that along with a parametric vector $(\beta_0, \beta_1, \dots, \beta_p)$ form the linear predictor:

$$\eta_i = \beta_0 + \beta_1 X_{i1} + \dots + \beta_p X_{ip}, \quad i = 1, \dots, n$$

- A monotonic and differentiable function called *link function* $g()$, defining the relationship between the mean of the response $\mu_i = E(Y_i)$ and the linear predictor:

$$g(\mu_i) = \eta_i$$

- Equivalently: $E[Y_i] = g^{-1}(\beta_0 + \beta_1 X_{i1} + \dots + \beta_p X_{ip})$.

- Start with the corresponding likelihood which contains the available information about the parameters: the coefficients of the linear predictor.
- For example, in the case of Poisson data with mean λ_i and logarithm link $g(\lambda_i) = \log(\lambda_i)$, the likelihood is:

$$l(\beta|\mathbf{y}, \mathbf{x}) = \prod_{i=1}^n \left(\frac{1}{y_i!} \right) \exp \{ y_i \mathbf{x}_i^t \beta - \exp(\mathbf{x}_i^t \beta) \}$$

- In frequentist statistics: maximum likelihood estimates are found using an iteratively reweighted least squares algorithm using either a Newton-Raphson method or the Fisher's scoring method.
- But, what about the Bayesian approach? Not much different conceptually.

- The Bayesian point of view is not a technique in the field of Statistics.
- It is another way of understanding and performing Statistics.
- And so, when our data bring us to Generalized Linear Models we can solve it using both Bayesian and Frequentist methods.
- Bayesian statistical analysis has benefited from the explosion of cheap and powerful desktop computing over the last two decades: MCMC.
- **Bayesian techniques can now be applied to complex modeling problems** where they could not have been applied previously.
- Bayesian perspective will probably continue to challenge, and perhaps supplant, traditional frequentist statistical methods which have dominated many disciplines of science for a long time.

- Conceptually the Bayesian specification is “straightforward”.
- Starting with the corresponding likelihood, we “only” need to **assign a prior for regression coefficients**.
- But, how to do this assignment? Not an easy answer.
- Easy option: to choose conjugate or non-informative independent priors. For instance normal or flat.
- But, as usual in Bayesian modelling, there are no closed form solution available for the posterior distribution of parameters.
- Here is where **Numerical techniques** come to rescue us by allowing us to obtain approximations of the posterior distribution.

- To conduct the Bayesian GLM in R, we can use the package `arm` which contains the `bayesglm` function (Gelman et al., 2010).
- We can also use the package `MCMCpack` also from R software which contains several functions to do so, like `MCMClogit`, `MCMCPoisson` or `MCMCprobit`.
- Nevertheless, there are more ways to perform Bayesian MCMC analysis in general which can also be used for GLM's. One of the most popular is BUGS (Bayesian Analysis using Gibbs Samplig) by Lunn et al. (2000).
<http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>
- BUGS has two ways to run it, WinBUGS being the most popular, and OpenBUGS being the other option.
- They can be used from R using the package `R2WinBUGS` or `BRugs` (the interface to the OpenBUGS).

- MCMC is not the only way to approximate posterior distributions.
- As obtaining posteriors is equivalent to integrate, numerical integration is another way to do it.
- Then, Laplace approximation can be used to numerically integrate.
- In fact the Integrated Nested Laplace Approximation (INLA) due to Rue et al. (2009) has become a fast and powerful tool lately.
- It can be easily connected with R: <http://www.r-inla.org/>

We consider a study in which the probability of children developing bronchitis or pneumonia in their first year of life is studied by type of feeding and sex (it can be found in the library faraway).

```
y<-rep(c(1,0,1,0,1,0,1,0,1,0,1,0),c(77,381,19,128,47,447,
                                         48,336,16,111,31,433))
sexn<-factor(rep(c("boy","girl"),c(1099,975)))
foodn<-factor(rep(c("Bottle","Suppl","Breast",
                    "Bottle","Suppl","Breast"),
                  c(458,147,494,384,127,464))))
db<-data.frame(y,sexn,foodn); summary(db)
```

```
##          y          sexn          foodn
##  Min.    :0.0000   boy :1099   Bottle:842
##  1st Qu.:0.0000   girl: 975   Breast:958
##  Median :0.0000                Suppl :274
##  Mean    :0.1148
##  3rd Qu.:0.0000
##  Max.    :1.0000
```

The `bayesglm` function represents a kind of short cut of the Bayesian approach to inference. Typically, the posterior is not used directly for making inferences. Instead, an empirical distribution is constructed based on draws from the posterior and that empirical distribution is what informs the inference(s).

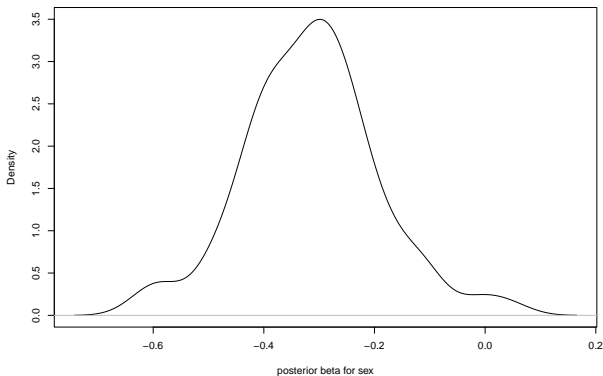
```
library(arm)
bm.1 <- bayesglm (y ~ sexn + foodn,
                  family = binomial(link="logit"),
                  prior.scale=Inf, prior.df=Inf)
# just a test: this should be identical to classical logit
# prior mean by default is 0

bm.2 <- bayesglm (y ~ sexn + foodn,
                  family = binomial(link="logit"))
# default Cauchy prior with scale 2.5

bm.3 <- bayesglm (y ~ sexn + foodn,
                  family = binomial(link="logit"),
                  prior.scale=2.5, prior.df=Inf)
```

We can retrieve the posterior distribution of all β parameters

```
plot(density(coef(sim(bm.3))[,2]), main="",  
      xlab="posterior beta for sex")
```



We can also retrieve the 95% credible interval for the coefficients

```
apply(coef(sim(bm.3)),2, quantile,c(.025,.5,.975))
```

##	(Intercept)	sexngirl	foodnBreast	foodnSuppl
## 2.5%	-1.844987	-0.58824046	-0.9551455	-0.5728774
## 50%	-1.599999	-0.32135165	-0.6714483	-0.1620121
## 97.5%	-1.391229	-0.05089078	-0.4118225	0.1830197

Recall, in Bayesian Statistics this credible interval is interpreted as: there is a 95% probability that the true population value of the coefficient for `girl` is between -0.55 and -0.05.

Install inla from <https://www.r-inla.org/download-install>

```
install.packages("INLA", repos=c(getOption("repos"),  
  INLA="https://inla.r-inla-download.org/R/stable"), dep=TRUE)  
# upgrade the package  
inla.upgrade()
```

```
library(INLA)
```

```
## Warning: package 'INLA' was built under R version 4.2.1
```

```
# INLA Method
```

```
bm.inla1 <- inla(y ~ sexn + foodn, data=db,  
  family = "binomial", control.compute = list(dic =
```

```
# INLA Method
# Output Posterior Estimates
round(bm.inla1$summary.fixed, 4)
```

##	mean	sd	0.025quant	0.5quant	0.975quant	mode
## (Intercept)	-1.6135	0.1124	-1.8379	-1.6121	-1.3967	NA
## sexngirl	-0.3130	0.1410	-0.5914	-0.3123	-0.0378	NA
## foodnBreast	-0.6700	0.1530	-0.9725	-0.6690	-0.3720	NA
## foodnSuppl	-0.1728	0.2056	-0.5860	-0.1695	0.2208	NA

- Once the inference has been done, we have to find the best model by selecting among the possible variables to be included in the model.
- One way (not the only one) to compare models in Bayesian statistics is using the DIC criterion. DIC is the counterpart of AIC or BIC to compare models. We can fit each model separately to calculate DIC or alternatively, all models can be simultaneously fitted.
- As usual, we select the best model as the one with the lowest DIC value.

```
bm.inla1$dic$dic
```

```
## [1] 1460.477
```

Note: this document is based on material kindly provided by Professor David Conesa of the University of Valencia and the Valencia Bayesian Research Group (<http://vabar.es/>)

- Albert, J. (2009). *Bayesian Computation with R*. Springer.