Bayesian statistics 2 - The linear model

Frédéric Barraquand (CNRS, IMB)

October 25, 2022

What did we learn the last time?

- Bayes theorem applied to random variables representing your data and parameters leads to

$$\mathbb{P}(\theta|data) = \frac{\mathbb{P}(data|\theta)\mathbb{P}(\theta)}{\mathbb{P}(data)}$$

- Bayesian analysis differs essentially from frequentist analysis in that both observations and parameters are assumed to arise from random variables
- Credible intervals are more complicated to compute (for simple models) than confidence intervals but logically more straightforward
- Bayesian and frequentist statistics give similar estimates when *n* is very large, but prior influence grows as sample size decreases.

Revisiting the Linear model in a Bayesian setup

- T-test. $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = \mu_1 + \delta \times x_i$ and $x_i = 0$ if in group 1 and $x_i = 1$ if in group 2 ($\mu_2 = \mu_1 + \delta$)

Revisiting the Linear model in a Bayesian setup

- T-test. $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = \mu_1 + \delta \times x_i$ and $x_i = 0$ if in group 1 and $x_i = 1$ if in group 2 ($\mu_2 = \mu_1 + \delta$)
- Linear regression
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + bx_i$ with x_i a covariate (9.6,4.2,...)

Revisiting the Linear model in a Bayesian setup

- T-test. $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = \mu_1 + \delta \times x_i$ and $x_i = 0$ if in group 1 and $x_i = 1$ if in group 2 ($\mu_2 = \mu_1 + \delta$)
- Linear regression
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + bx_i$ with x_i a covariate (9.6,4.2,...)
- (One-way) analysis of variance (aka the T-test with more groups)
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i3}$ with x_{i1}, x_{i2}, x_{i3} binary (0 or 1)

Revisiting the Linear model in a Bayesian setup

- T-test. $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = \mu_1 + \delta \times x_i$ and $x_i = 0$ if in group 1 and $x_i = 1$ if in group 2 ($\mu_2 = \mu_1 + \delta$)
- Linear regression
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + bx_i$ with x_i a covariate (9.6,4.2,...)
- (One-way) analysis of variance (aka the T-test with more groups)
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i3}$ with x_{i1}, x_{i2}, x_{i3} binary (0 or 1)
- $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + \ldots$
  multiple linear regression (x_1 and x_2 real-valued), ANCOVA (x_1 binary and x_2 real-valued),...

Revisiting the Linear model in a Bayesian setup

- T-test. $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = \mu_1 + \delta \times x_i$ and $x_i = 0$ if in group 1 and $x_i = 1$ if in group 2 ($\mu_2 = \mu_1 + \delta$)
- Linear regression
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + bx_i$ with x_i a covariate (9.6,4.2,...)
- (One-way) analysis of variance (aka the T-test with more groups)
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i3}$ with x_{i1}, x_{i2}, x_{i3} binary (0 or 1)
- $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + ...$
  multiple linear regression (x_1 and x_2 real-valued), ANCOVA (x_1 binary and x_2 real-valued),...

All models that can be written

$$Y = X\beta + E$$

where X is called the *design matrix* and $Y = (Y_i)_{i \in 1:n}, \beta, E$ are vectors.

Revisiting the Linear model in a Bayesian setup

- T-test. $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = \mu_1 + \delta \times x_i$ and $x_i = 0$ if in group 1 and $x_i = 1$ if in group 2 ($\mu_2 = \mu_1 + \delta$)
- Linear regression
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + bx_i$ with x_i a covariate (9.6,4.2,...)
- (One-way) analysis of variance (aka the T-test with more groups)
  $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i3}$ with x_{i1}, x_{i2}, x_{i3} binary (0 or 1)
- $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, $\mu_i = a + b_1 x_{i1} + b_2 x_{i2} + ...$
  multiple linear regression (x_1 and x_2 real-valued), ANCOVA (x_1 binary and x_2 real-valued),...

All models that can be written

$$Y = X\beta + E$$

where X is called the *design matrix* and $Y = (Y_i)_{i \in 1:n}, \beta, E$ are vectors.

Also called the *General Linear Model*, and should not be confused with *GLM = Generalized Linear Model*.

Using JAGS through R2jags

JAGS = Just Another Gibbs Sampler.

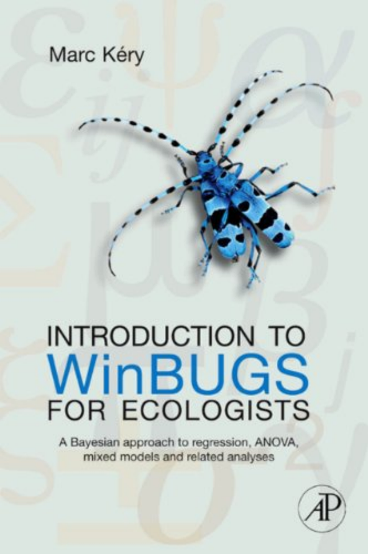
Update on BUGS = Bayesian analysis Using the Gibbs Sampler.
(for now we admit what is a Gibbs sampler, will be made clear in lecture 3)

BUGS started in 1989 in the MRC Biostatistics Unit, Cambridge. Followed by WinBUGS, OpenBUGS and now JAGS by Martyn Plummer.

JAGS: essentially identical syntax to BUGS, works on Linux Mac and Windows.

JAGS and WinBUGS are *softwares* that interpret a language.

Examples coming from Kéry's Introduction to WinBUGS



Marc Kéry

INTRODUCTION TO WinBUGS FOR ECOLOGISTS

A Bayesian approach to regression, ANOVA, mixed models and related analyses

Examples coming from Kéry's Introduction to WinBUGS

Falcons, again! (James Audubon drawing).

Examples coming from Kéry's Introduction to WinBUGS

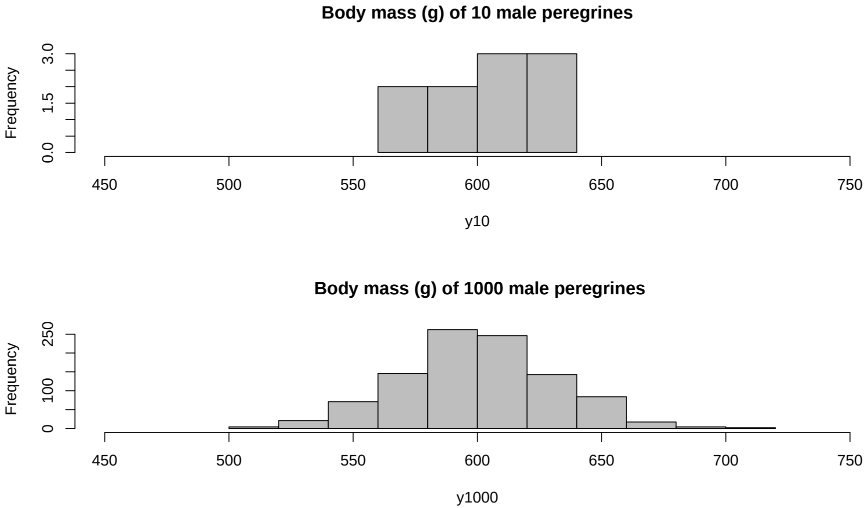
Peregine falcon. Fastest living stuff on Earth. Photo by Sumeet Moghe

Generating body size data

```
#Generate two samples of body mass measurements of male peregrines
y10 <- rnorm(n = 10, mean = 600, sd = 30) ## Sample of 10 birds
y1000 <- rnorm(n = 1000, mean = 600, sd = 30) ## Sample of 1000 birds

# Plot data
xlim = c(450, 750)
par(mfrow = c(2,1))
hist(y10, col = ’grey’, xlim = xlim, main = ’Body mass (g) of 10 male
    peregrines’)
hist(y1000, col = ’grey’, xlim = xlim, main = ’ Body mass (g) of 1000
    male peregrines’)
```

Generating body size data



Male mass distrib= f(sample size)

Feeding $Y_i \sim \mathcal{N}(\mu, \sigma^2)$ to JAGS

```r
### Analysis using R
summary(lm(y1000 ~ 1))

###Analysis using JAGS
library(R2jags)                     # Load R2jags

# Bundle and summarize the data set passed to JAGS
str(jags.data <- list(mass = y1000, nobs = length(y1000)))
```

Feeding $Y_i \sim \mathcal{N}(\mu, \sigma^2)$ to JAGS

```
# Specify model in BUGS language
cat(file = "model.txt", "
model {

## Priors
population.mean ~ dunif(0,5000)

# Normal parameterized by precision
precision <- 1 / population.variance
# Precision = 1/variance
population.variance <- population.sd * population.sd
population.sd ~ dunif(0,100)

## Likelihood
for(i in 1:nobs){
  mass[i] ~ dnorm(population.mean, precision)
}

}
")
```

Feeding $Y_i \sim \mathcal{N}(\mu, \sigma^2)$ to JAGS

```
# Initial values
inits <- function()list(population.mean = rnorm(1,600), population.sd =
    runif(1, 1, 30))

# Parameters to be monitored (= to estimate)
params <- c("population.mean", "population.sd", "population.variance")

# MCMC settings
nc <- 3          # Number of chains
ni <- 1000         # Number of draws from posterior (for each chain)
nb <- 1          # Number of draws to discard as burn-in
nt <- 1          # Thinning rate

# Call JAGS from R, check convergence and summarize posteriors
out <- jags(jags.data, inits, parameters=params, model.file="model.txt",
     n.chains=nc, n.thin=nt, n.iter=ni, n.burnin=nb, working.directory
    = getwd())
```

Feeding $Y_i \sim \mathcal{N}(\mu, \sigma^2)$ to JAGS

```
print(out, dig = 2)
```

And you should then observe

```
Inference for Bugs model at "model.txt", fit using jags,
 3 chains, each with 1000 iterations (first 1 discarded)
 n.sims = 2997 iterations saved
                    mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff
population.mean      600.37    0.94  598.55  599.74  600.35  600.98  602.26 1.00  2100
population.sd         31.04    0.69   29.75   30.57   31.02   31.52   32.36 1.00  1100
population.variance  964.15   42.86  884.85  934.77  962.52  993.42 1047.46 1.00  1100
deviance            9708.32    2.46 9706.46 9706.96 9707.71 9708.98 9713.51 1.05  3000

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 3.0 and DIC = 9711.3
DIC is an estimate of expected predictive error (lower deviance is better).
```

Now for the t-test

Let's go to the code!

Practicals: ANOVA.