From fixed to random effects

Bayesian statistics 4 – random and mixed effects models

Frédéric Barraquand (CNRS, IMB)

09/11/2021

Some things that we learned the last time(s)

Session 3

- MCMC = Monte Carlo + Markov Chain
- Requires two types of convergence to compute an posterior means or posterior distribution
- JAGS uses the Gibbs sampler, a multicomponent variant of the Metropolis algorithm
- The Gibbs sampler allows to sample parameter-rich models

Session 2

- T-tests, ANOVA and the likes can be framed as the General Linear Model
- The Linear Model $Y = X\beta + E$ is easily fitted with JAGS
- Uncertainties in effects $\rightarrow$ posteriors

Back to Snout-Vent Length (SVL) Snake data

From Kéry (2010) & TD 2

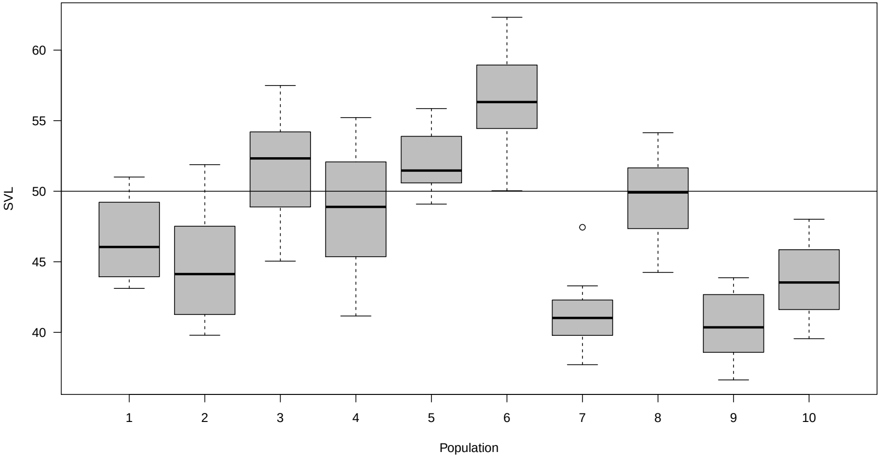
```
### Data generation
# same as TD2 but number of groups x 2
npop <- 10                    # Number of populations: now choose 10 rath
nsample <- 12                     # Number of snakes in each
n <- npop * nsample               # Total number of data points

pop.grand.mean <- 50                  # Grand mean SVL
pop.sd <- 5                   # sd of population effects about mean
pop.means <- rnorm(n = npop, mean = pop.grand.mean, sd = pop.sd)
sigma <- 3                    # Residual sd
eps <- rnorm(n, 0, sigma)         # Draw residuals

x <- rep(1:npop, rep(nsample, npop))
X <- as.matrix(model.matrix(~ as.factor(x)-1))
y <- as.numeric(X %*% as.matrix(pop.means) + eps) # as.numeric is E
```

The data: Snout-vent length in snakes

```
boxplot(y ~ x, col = "grey", xlab = "Population", ylab = "SVL", mai
abline(h = pop.grand.mean)
```



Questions that we could ask

- Effect of being in population *i*
- Is there more variation between populations or more residual variation?

The one-way ANOVA: from code to maths

$J = 10$ Groups. Notations

$$Y_{ij} = \alpha_i + \epsilon_{ij}, \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

Practical if $i = 1, ..., I$ is the same number of individuals per group.
$n = I \times J$.

The one-way ANOVA: from code to maths

$J = 10$ Groups. Notations

$$Y_{ij} = \alpha_i + \epsilon_{ij}, \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

Practical if $i = 1, ..., I$ is the same number of individuals per group.
$n = I \times J$.

or again

$$Y_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2), \mu_{ij} = \alpha_i$$

By that we mean that $\mathbb{E}(Y_{ij}) = \alpha_i$.

The one-way ANOVA: from code to maths

$J = 10$ Groups. Notations

$$Y_{ij} = \alpha_i + \epsilon_{ij}, \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

Practical if $i = 1, ..., I$ is the same number of individuals per group. $n = I \times J$.

or again

$$Y_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2), \mu_{ij} = \alpha_i$$

By that we mean that $\mathbb{E}(Y_{ij}) = \alpha_i$.

Note: if you have an overall mean μ you need to remove a group

The one-way ANOVA: from code to maths

$J = 10$ Groups. Notations

$$Y_{ij} = \alpha_i + \epsilon_{ij}, \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

Practical if $i = 1, ..., I$ is the same number of individuals per group. $n = I \times J$.

or again

$$Y_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2), \mu_{ij} = \alpha_i$$

By that we mean that $\mathbb{E}(Y_{ij}) = \alpha_i$.

Note: if you have an overall mean μ you need to remove a group

or again with $i = 1, ..., n$

$$Y_i \sim \mathcal{N}(\mu_i, \sigma^2), \mu_i = \alpha_{k[i]}$$

where $k[i]$ returns in which group is i. How we coded this JAGS.

Running again the ANOVA

```
## List of 2
##  $ y: num [1:120] 44.4 49.6 45.1 51 46.4 ...
##  $ x: int [1:120] 1 1 1 1 1 1 1 1 1 1 ...
```

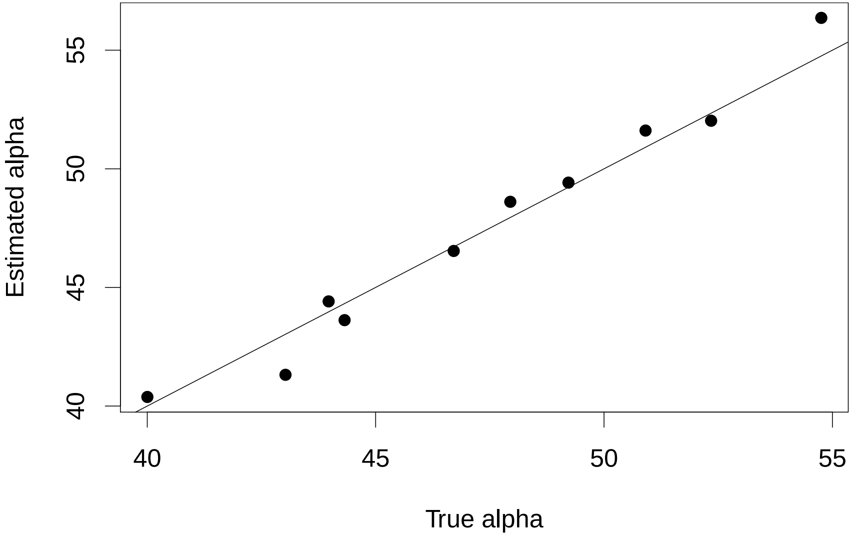
```
# Specify model in BUGS language
cat(file = "anova.txt", "
    model {

    # Priors
    for (i in 1:10){            # Implicitly define alpha as a vector
    alpha[i] ~ dnorm(0, 0.001) # Beware that a mean at 0 only works because
    }
    sigma ~ dunif(0, 100)

    # Likelihood
    for (i in 1:120) {
    y[i] ~ dnorm(mean[i], tau)
    mean[i] <- alpha[x[i]]
    }

    # Derived quantities
    tau <- 1 / ( sigma * sigma)
```

Estimated effects vs theoretical effects



Classical random effect modelling I

```r
### Restricted maximum likelihood (REML) analysis using R
library('lme4')                    # Load lme4

pop <- as.factor(x)                # Define x as a factor and call it pop

lme.fit <- lmer(y ~ 1 + 1 | pop, REML = TRUE)
lme.fit                         # Inspect results
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ 1 + 1 | pop
## REML criterion at convergence: 650.1976
## Random effects:
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 5.041
##  Residual             3.200
## Number of obs: 120, groups:  pop, 10
## Fixed Effects:
## (Intercept)
```

Classical random effect modelling II

```
##       47.47
```

```r
ranef(lme.fit)                    # Print random effects
```

```
## $pop
##    (Intercept)
## 1   -0.8446384
## 2   -2.9232184
## 3    4.0643402
## 4    1.1482505
## 5    4.4691200
## 6    8.6668906
## 7   -5.9309288
## 8    1.8830870
## 9   -6.8293818
## 10  -3.7035208
##
## with conditional variances for "pop"
```

Classical random effect model - maths

$$Y_i \sim \mathcal{N}(\mu_i, \sigma^2), \mu_i = \alpha_{k[i]}$$

where $k[i]$ returns in which group is i. What's missing?

Classical random effect model - maths

$$Y_i \sim \mathcal{N}(\mu_i, \sigma^2), \mu_i = \alpha_{k[i]}$$

where $k[i]$ returns in which group is i. What's missing?

i.i.d. observations. And then?

Classical random effect model - maths

$$Y_i \sim \mathcal{N}(\mu_i, \sigma^2), \mu_i = \alpha_{k[i]}$$

where $k[i]$ returns in which group is i. What's missing?

i.i.d. observations. And then?

We estimate the variance of the random effects

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$$

(we have to specify a mean too)

Random effect model in a Bayesian framework I

```r
# Bundle and summarize the data set passed to JAGS
str(bdata <- list(y = y, x = x, npop = npop, n = n))
```

```
## List of 4
##  $ y   : num [1:120] 44.4 49.6 45.1 51 46.4 ...
##  $ x   : int [1:120] 1 1 1 1 1 1 1 1 1 1 ...
##  $ npop: num 10
##  $ n   : num 120
```

Random effect model in a Bayesian framework II

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

# Priors and some derived things
for (i in 1:npop){
    alpha[i] ~ dnorm(mu, tau.alpha)     # Prior for population means
    effect[i] <- alpha[i] - mu  # Population effects as derived quan
 }
 mu ~ dnorm(0,0.001)            # Hyperprior for grand mean svl
 sigma.alpha ~ dunif(0, 10)     # Hyperprior for sd of population ef
 sigma.res ~ dunif(0, 10)       # Prior for residual sd

# Likelihood
 for (i in 1:n) {
    y[i] ~ dnorm(mean[i], tau.res)
    mean[i] <- alpha[x[i]]
 }

# Derived quantities
```

Fitting the model I

```
# Inits function
inits <- function(){ list(mu = runif(1, 0, 100), sigma.alpha = rlno

# Params to estimate
params <- c("mu", "alpha", "effect", "sigma.alpha", "sigma.res")

# MCMC settings
nb <- 1000  ;  nc <- 3  ;  ni <- 2000  ;  nt <- 2

# Call JAGS, check convergence and summarize posteriors
out2 <- jags(bdata, inits, params, "re.anova.txt", n.thin = nt, n.c
             n.burnin = nb, n.iter = ni)
```

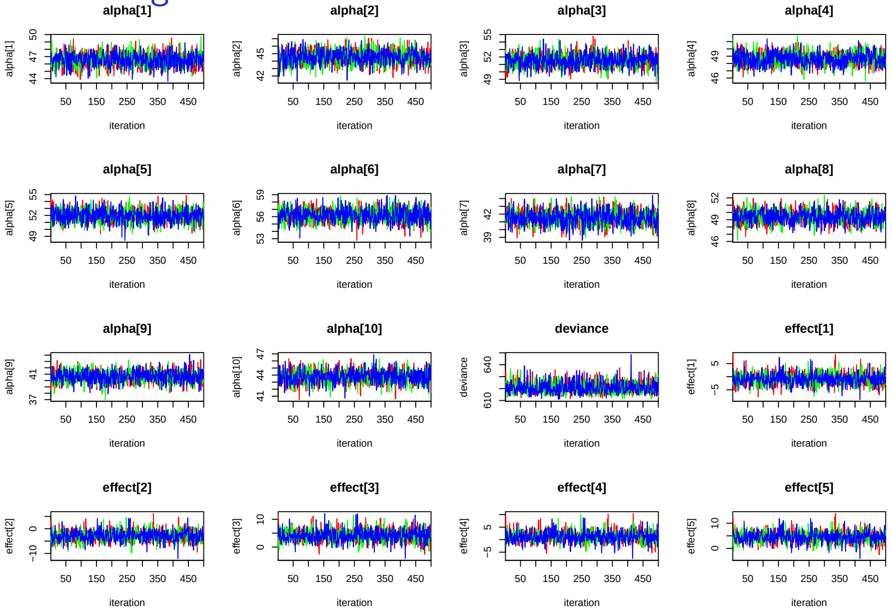
Fitting the model II

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 120
##    Unobserved stochastic nodes: 13
##    Total graph size: 273
##
## Initializing model
```

Model diagnostics I

```
traceplot(out2,mfrow=c(4,4))
```

Model diagnostics II



Model results I

```r
print(out2,dig=3)
```

```
## Inference for Bugs model at "re.anova.txt", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded), n.thin = 2
##  n.sims = 1500 iterations saved
##             mu.vect sd.vect    2.5%    25%    50%    75%  97.5%  Rhat n.eff
## alpha[1]     46.594   0.930  44.684 45.980 46.588 47.249 48.464 1.002  1200
## alpha[2]     44.547   0.939  42.715 43.919 44.574 45.179 46.332 1.001  1500
## alpha[3]     51.548   0.912  49.811 50.942 51.554 52.174 53.316 1.000  1500
## alpha[4]     48.647   0.888  47.045 48.052 48.616 49.257 50.412 1.002   810
## alpha[5]     51.955   0.932  50.164 51.334 51.944 52.564 53.834 1.000  1500
## alpha[6]     56.186   0.929  54.409 55.539 56.210 56.838 58.016 1.000  1500
## alpha[7]     41.534   0.929  39.700 40.946 41.529 42.152 43.341 1.001  1500
## alpha[8]     49.379   0.895  47.641 48.785 49.366 49.972 51.231 1.000  1500
## alpha[9]     40.611   0.937  38.711 40.008 40.634 41.247 42.399 1.000  1500
## alpha[10]    43.734   0.917  41.999 43.094 43.742 44.349 45.648 1.001  1500
## effect[1]    -0.796   2.177  -4.967 -2.182 -0.798  0.556  3.489 1.003  1400
## effect[2]    -2.843   2.165  -7.068 -4.212 -2.893 -1.506  1.391 1.000  1500
## effect[3]     4.158   2.153   0.007  2.758  4.103  5.523  8.591 1.000  1500
## effect[4]     1.257   2.185  -2.820 -0.187  1.209  2.561  5.624 1.003  1200
## effect[5]     4.565   2.139   0.411  3.212  4.532  5.927  8.709 1.000  1500
## effect[6]     8.796   2.182   4.678  7.342  8.779 10.185 13.353 1.002  1500
## effect[7]    -5.856   2.156 -10.102 -7.273 -5.788 -4.526 -1.667 1.001  1500
```

Model results II

```
## effect[8]      1.989   2.149  -2.222   0.598   1.946   3.342   6.418 1.001  1500
## effect[9]     -6.779   2.158 -10.938  -8.163  -6.770  -5.399  -2.398 1.000  1500
## effect[10]    -3.656   2.188  -8.157  -5.095  -3.692  -2.337   0.878 1.000  1500
## mu            47.390   2.001  43.232  46.146  47.436  48.652  51.305 1.001  1500
## sigma.alpha    5.812   1.452   3.562   4.675   5.640   6.668   9.199 1.006   310
## sigma.res      3.227   0.219   2.847   3.077   3.209   3.369   3.696 1.002  1500
## deviance     620.795   4.802 613.389 617.307 620.047 623.532 631.576 1.001  1500
##
## 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 = 11.5 and DIC = 632.3
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Comparison of variance estimates

```
### Well, comparison of sigma's...
VarCorr(lme.fit)
```

```
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 5.0409
##  Residual             3.2005
```

```
out2$BUGSoutput$mean$sigma.res #true value is 3
```

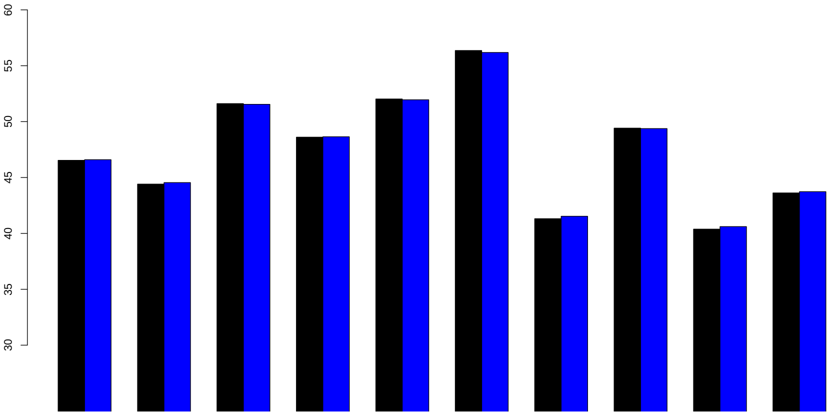
```
## [1] 3.226858
```

```
out2$BUGSoutput$mean$sigma.alpha #true value is 5
```

```
## [1] 5.811853
```

Comparison of fixed and random effects

```
## Plotting shrinkage
alpha_mean2 = out2$BUGSoutput$mean$alpha
barplot(t(matrix(c(alpha_mean,alpha_mean2),ncol=2,nrow=10)),beside=T
```



Re-running the analysis with more shrinkage I

Now we assume a prior $\sigma_\alpha \sim \text{Gamma}(100, 50)$.

More details on the Gamma distribution

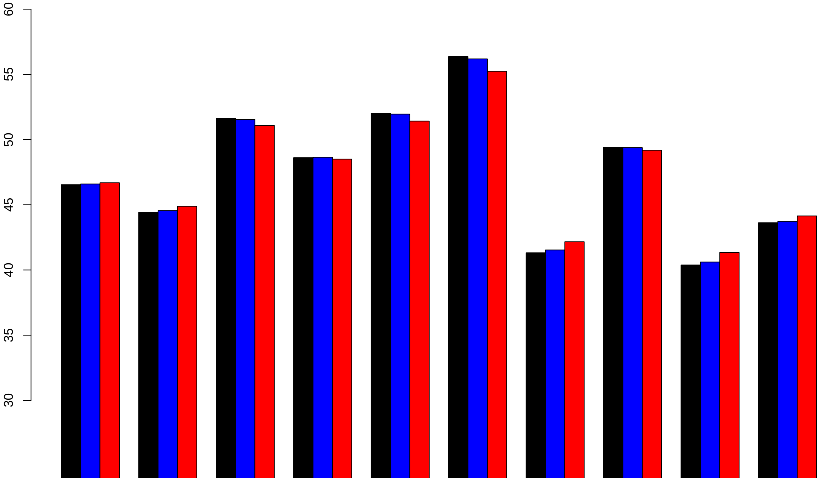
- $X \sim \text{Gamma}(a, b)$ with a = shape, b = rate = $\frac{1}{\theta}$ where θ is scale.
- Properties: $\mathbb{E}(X) = a\theta = 100/50 = 2$ and $\mathbb{V}(X) = a\theta^2 = \frac{100}{2500} = 0.04$ so that $\text{SD}(X) = 0.2$.

```
alpha_mean3 = out3$BUGSoutput$mean$alpha
out3$BUGSoutput$mean$sigma.alpha
```

```
## [1] 2.451122
```

```
barplot(t(matrix(c(alpha_mean,alpha_mean2,alpha_mean3),ncol=3,
```

Re-running the analysis with more shrinkage II



To wrap up – ANOVA vs REs in a Bayesian setup

- The difference between ANOVA and random effects models boils down to estimation of variance between group-level effects

To wrap up – ANOVA vs REs in a Bayesian setup

- The difference between ANOVA and random effects models boils down to estimation of variance between group-level effects
- (usually confusing because ANOVA *does* some variance partitioning)

To wrap up – ANOVA vs REs in a Bayesian setup

- The difference between ANOVA and random effects models boils down to estimation of variance between group-level effects
- (usually confusing because ANOVA *does* some variance partitioning)
- ANOVA e.g. $\alpha_j \sim \mathcal{N}(50, 100)$

To wrap up – ANOVA vs REs in a Bayesian setup

- The difference between ANOVA and random effects models boils down to estimation of variance between group-level effects
- (usually confusing because ANOVA *does* some variance partitioning)
- ANOVA e.g. $\alpha_j \sim \mathcal{N}(50, 100)$
- RE model $\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$ and $\sigma_\alpha \sim \mathcal{U}(0, 10)$ or $\sigma_\alpha \sim \text{Exp}(0, 1/5)$

To wrap up – ANOVA vs REs in a Bayesian setup

- The difference between ANOVA and random effects models boils down to estimation of variance between group-level effects
- (usually confusing because ANOVA *does* some variance partitioning)
- ANOVA e.g. $\alpha_j \sim \mathcal{N}(50, 100)$
- RE model $\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$ and $\sigma_\alpha \sim \mathcal{U}(0, 10)$ or $\sigma_\alpha \sim \text{Exp}(0, 1/5)$
- RE model with more shrinkage $\sigma_\alpha \sim \text{Gamma}(100, 50)$.

More material

- Shrinkage aka partial pooling is a property of mixed models, not Bayesian estimation (though you can top it up using informative priors)
- Kruschke's post on parameterizing the Gamma distribution

Bonus: fun and pretty snakes

Figure 1: *Vipera ursinii* Benny Trapp (CC BY)

Super épisode de La méthode scientifique sur France Culture, 08/11/2021