From fixed to random effects

Bayesian statistics 4 – random and mixed effects models

Frédéric Barraquand (CNRS, IMB)

15/11/2021

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 $Y = X\beta + E,\ E \sim \mathcal{N}(0, \Sigma)$ is easily fitted with JAGS
- Uncertainties in effects $\rightarrow$ posteriors

Back to Snout-Vent Length (SVL) Snake data

From Kéry (2010) & TD 2

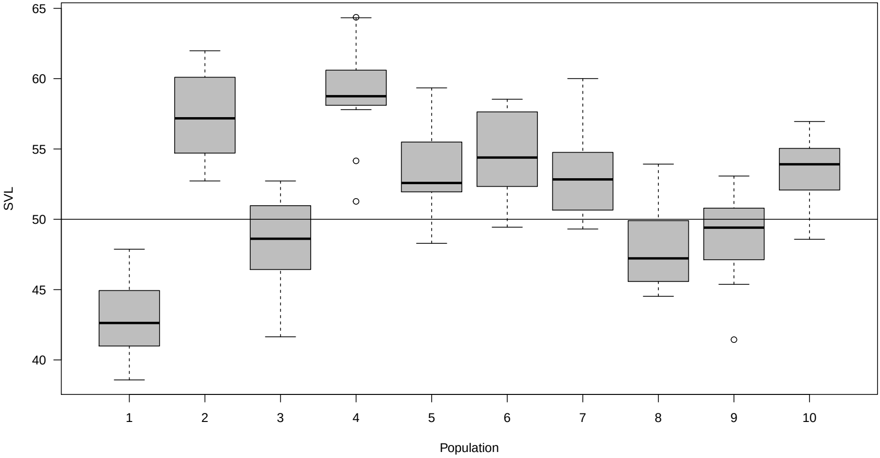
```
### Data generation
# same as TD2 but number of groups x 2
npop <- 10               # Number of populations: now choose 10 rath
nsample <- 12                # Number of snakes in each
n <- npop * nsample          # Total number of data points

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

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

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

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

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

or again

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

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

The one-way ANOVA: from code to maths

$J = 10$ Groups. Notations

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

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

or again

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

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

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

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

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

or again

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

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

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] 40.9 41.6 45.5 43.6 47.9 ...
##  $ x: int [1:120] 1 1 1 1 1 1 1 1 1 1 ...
```

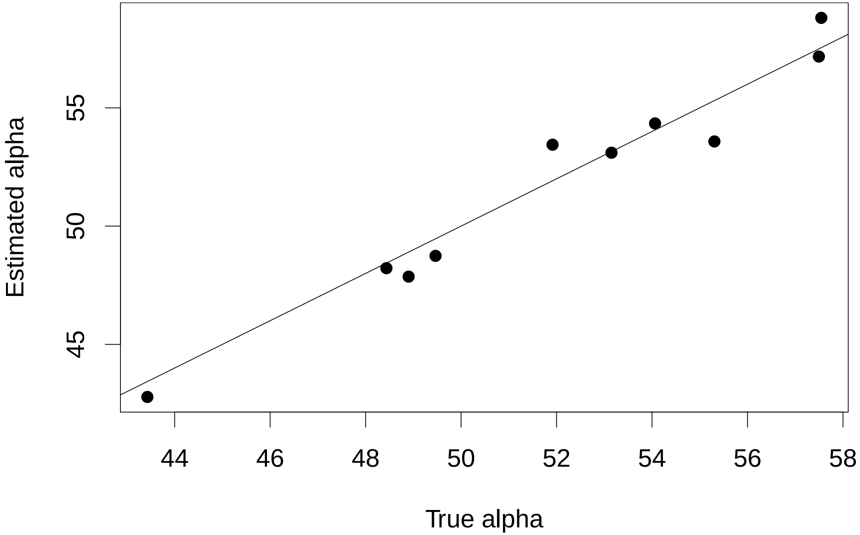
```
# Specify model in BUGS language
cat(file = "anova.txt", "
    model {

    # 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: 640.7965
## Random effects:
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 4.774
##  Residual             3.080
## Number of obs: 120, groups:  pop, 10
## Fixed Effects:
## (Intercept)
```

Classical random effect modelling II

```
##       51.85
```

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

```
## $pop
##    (Intercept)
## 1    -8.761097
## 2     5.203301
## 3    -3.458541
## 4     6.790861
## 5     1.567910
## 6     2.484507
## 7     1.239695
## 8    -3.802799
## 9    -2.967132
## 10    1.703295
##
## 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

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

for $j = 1, ..., J$ (we have to specify a mean μ_α too, we can set it to zero if there is an overall mean μ though)

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] 40.9 41.6 45.5 43.6 47.9 ...
##  $ 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

```r
# 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)
```

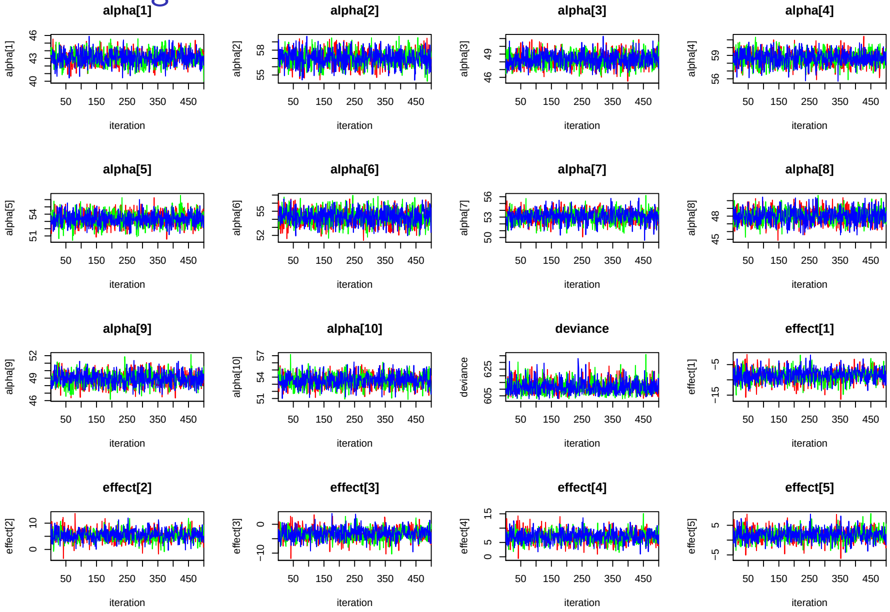
Fitting the model II

```
## 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

```
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]    43.047   0.890  41.401 42.437 43.047 43.647 44.780 1.000  1500
## alpha[2]    57.066   0.903  55.195 56.461 57.108 57.677 58.861 1.001  1500
## alpha[3]    48.325   0.870  46.679 47.713 48.302 48.914 50.051 1.001  1500
## alpha[4]    58.678   0.878  56.883 58.106 58.698 59.260 60.359 1.000  1500
## alpha[5]    53.368   0.881  51.641 52.780 53.374 53.940 55.144 1.002  1500
## alpha[6]    54.304   0.911  52.441 53.698 54.303 54.900 56.124 1.007   300
## alpha[7]    53.088   0.890  51.296 52.518 53.101 53.677 54.738 1.000  1500
## alpha[8]    47.994   0.888  46.290 47.407 47.996 48.592 49.820 1.002  1500
## alpha[9]    48.916   0.865  47.221 48.354 48.925 49.448 50.618 1.000  1500
## alpha[10]   53.537   0.890  51.764 52.939 53.529 54.123 55.242 1.000  1500
## effect[1]   -8.596   2.004 -12.663 -9.829 -8.614 -7.305 -4.545 1.006   490
## effect[2]    5.423   2.019   1.550  4.099  5.380  6.698  9.479 1.004   690
## effect[3]   -3.318   1.983  -7.298 -4.610 -3.321 -2.041  0.472 1.005   460
## effect[4]    7.036   1.992   3.072  5.786  6.980  8.319 11.053 1.006   620
## effect[5]    1.725   1.989  -2.174  0.449  1.692  2.993  5.889 1.004   770
## effect[6]    2.661   2.055  -1.397  1.319  2.603  4.033  6.722 1.005   460
## effect[7]    1.445   2.034  -2.462  0.149  1.412  2.697  5.658 1.005   480
```

Model results II

```
## effect[8]     -3.648   1.973  -7.470  -4.922  -3.692  -2.372   0.324 1.005   600
## effect[9]     -2.726   1.976  -6.605  -3.948  -2.743  -1.460   1.092 1.004   450
## effect[10]     1.894   1.999  -2.147   0.641   1.858   3.138   5.928 1.004   490
## mu            51.643   1.845  47.984  50.445  51.699  52.809  55.295 1.005   470
## sigma.alpha    5.518   1.381   3.419   4.480   5.335   6.414   8.721 1.003  1500
## sigma.res      3.110   0.220   2.709   2.961   3.103   3.238   3.600 1.000  1500
## deviance     611.719   4.941 604.350 608.042 610.971 614.540 623.296 1.000  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 = 12.2 and DIC = 623.9
## 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) 4.7740
##  Residual             3.0801
```

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

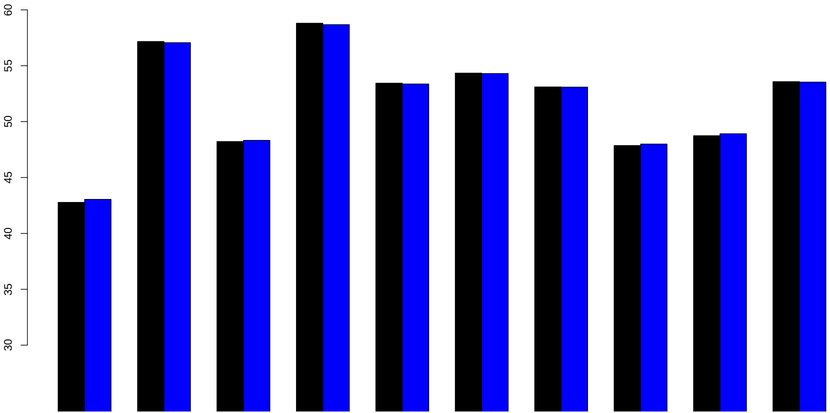
```
## [1] 3.109739
```

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

```
## [1] 5.517962
```

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

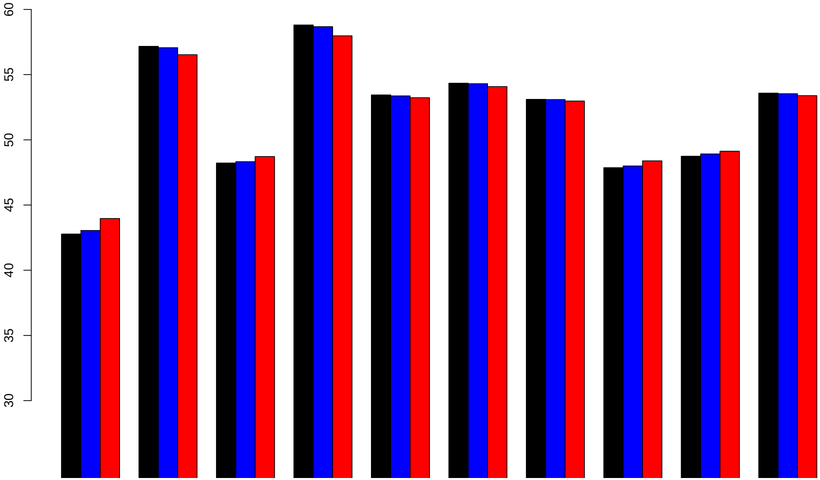
- $X \sim \text{Gamma}(a, b)$ with a = shape, b = rate $= \frac{1}{\theta}$ where θ is scale.
- Properties: $\mathbb{E}(X) = a\theta = 100/50 = 2$ and $\mathbb{V}(X) = a\theta^2 = \frac{100}{2500} = 0.04$ so that $\text{SD}(X) = 0.2$.

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

```
## [1] 2.408461
```

```
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