# From fixed to random effects

## Bayesian statistics 4 – random and mixed effects models

Frédéric Barraquand (CNRS, IMB)

21/11/2023

# 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

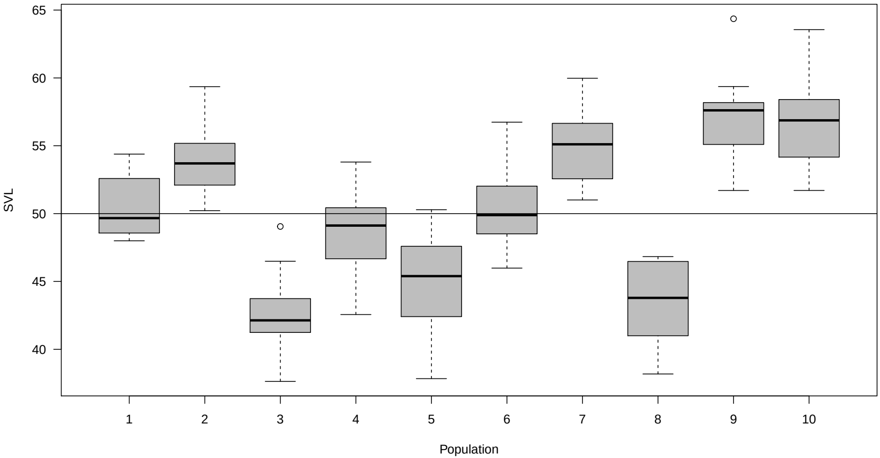
# Back to Snout-Vent Length (SVL) Snake data

From Kéry (2010) & TD 2

```
### Data generation
# same as TD2 but number of groups x 2
npop <- 10                  # Number of populations: now choose 10 rather than 5
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 ESSENTIAL
```

# The data: Snout-vent length in snakes

```
boxplot(y ~ x, col = "grey", xlab = "Population", ylab = "SVL", main = "", las = 1)
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 $\mu$ 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 $\mu$ you need to remove a group

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

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

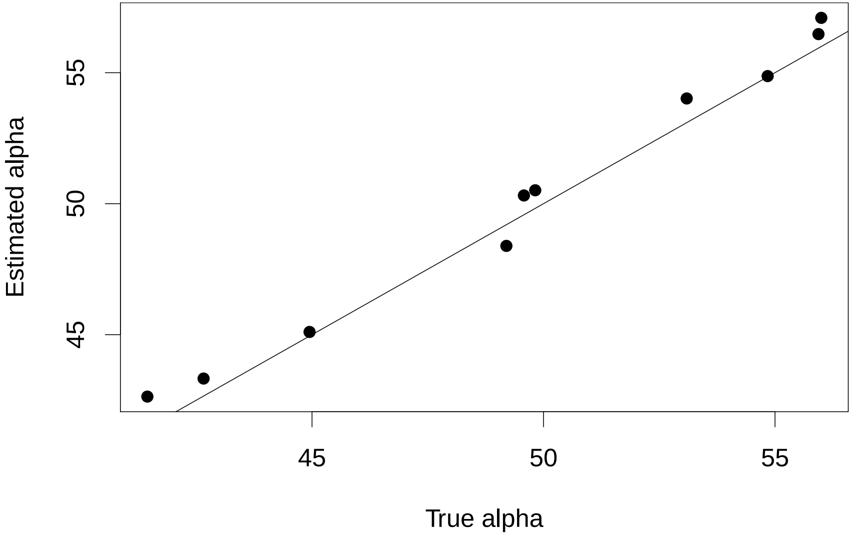
where $k[i]$ returns in which group is $i$. How we coded this JAGS.

# Running again the ANOVA

```
## List of 2
##  $ y: num [1:120] 50.2 52.2 48.5 54.4 49.2 ...
##  $ 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 variance is huge.
    }
    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

```
### 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: 639.3646
## Random effects:
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 5.275
##  Residual             3.036
## Number of obs: 120, groups:  pop, 10
## Fixed Effects:
## (Intercept)
##       50.31
```

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

# Classical random effect modelling II

```
## $pop
##    (Intercept)
## 1   0.21274774
## 2   3.63046882
## 3  -7.41853225
## 4  -1.82601034
## 5  -5.04292289
## 6   0.03558689
## 7   4.50071632
## 8  -6.77164655
## 9   6.60775311
## 10  6.07183915
##
## 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 $\mu_\alpha$ too, we can set it to zero if there is an overall mean $\mu$ 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] 50.2 52.2 48.5 54.4 49.2 ...
##  $ 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 quant's
 }
 mu ~ dnorm(0,0.001)            # Hyperprior for grand mean svl
 sigma.alpha ~ dunif(0, 10)     # Hyperprior for sd of population effects
 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
 tau.alpha <- 1 / (sigma.alpha * sigma.alpha)
 tau.res <- 1 / (sigma.res * sigma.res)
}
")
```

# Fitting the model I

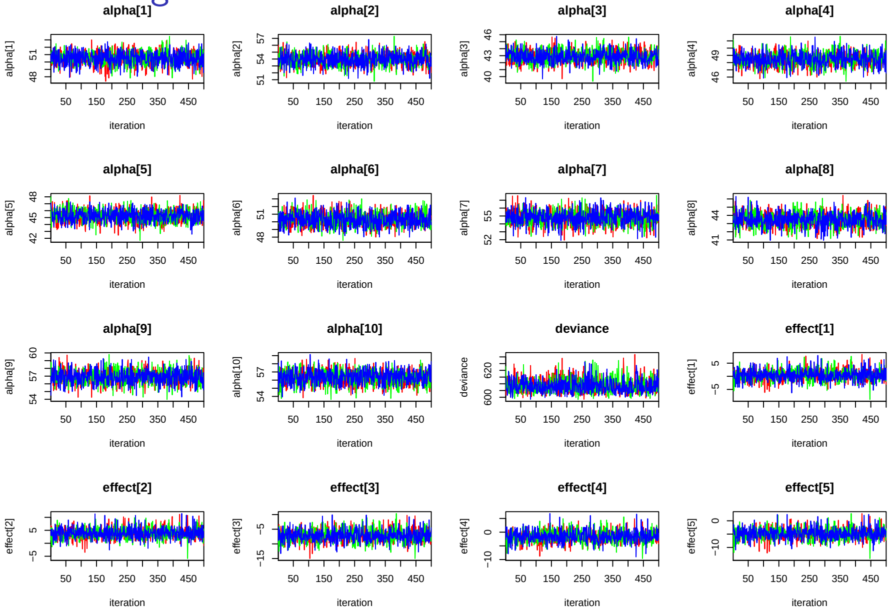
```
# Inits function
inits <- function(){ list(mu = runif(1, 0, 100), sigma.alpha = rlnorm(1), sigma.res

# 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.chains = nc,
            n.burnin = nb, n.iter = ni)
```

```
## 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]     50.511   0.897  48.613 49.958 50.534 51.128 52.184 1.001  1500
## alpha[2]     53.969   0.908  52.171 53.370 53.965 54.570 55.772 1.001  1300
## alpha[3]     42.902   0.908  41.184 42.309 42.848 43.491 44.694 1.001  1500
## alpha[4]     48.465   0.901  46.655 47.893 48.469 49.042 50.296 1.002  1100
## alpha[5]     45.242   0.891  43.562 44.632 45.249 45.823 46.977 1.001  1500
## alpha[6]     50.336   0.910  48.666 49.668 50.353 50.974 52.179 1.001  1500
## alpha[7]     54.795   0.895  52.988 54.210 54.811 55.384 56.573 1.004   900
## alpha[8]     43.502   0.859  41.739 42.941 43.498 44.071 45.147 1.001  1500
## alpha[9]     56.900   0.899  55.149 56.276 56.887 57.495 58.716 1.000  1500
## alpha[10]    56.355   0.861  54.601 55.778 56.370 56.938 57.940 1.003   790
## effect[1]     0.451   2.108  -3.587 -0.929  0.446  1.724  4.899 1.001  1500
## effect[2]     3.909   2.092   0.081  2.585  3.869  5.195  8.281 1.000  1500
## effect[3]    -7.159   2.083 -11.042 -8.489 -7.193 -5.977 -2.783 1.001  1500
## effect[4]    -1.595   2.106  -5.622 -2.950 -1.674 -0.296  2.522 1.002  1200
## effect[5]    -4.818   2.078  -8.861 -6.102 -4.843 -3.548 -0.564 1.001  1500
## effect[6]     0.276   2.108  -3.771 -1.125  0.211  1.602  4.484 1.001  1500
## effect[7]     4.735   2.100   0.607  3.333  4.711  6.029  9.035 1.001  1500
```

# Model results II

```
## effect[8]     -6.559   2.096 -10.620  -7.878  -6.494  -5.244  -2.565 1.000  1500
## effect[9]      6.839   2.122   2.700   5.498   6.820   8.184  11.113 1.001  1500
## effect[10]     6.295   2.105   2.234   4.904   6.258   7.566  10.619 1.001  1500
## mu            50.060   1.940  46.092  48.911  50.030  51.312  53.755 1.001  1500
## sigma.alpha    5.917   1.406   3.796   4.845   5.681   6.778   9.166 1.001  1500
## sigma.res      3.074   0.208   2.713   2.928   3.060   3.205   3.503 1.000  1500
## deviance     608.576   5.088 600.782 604.817 607.892 611.520 620.235 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 = 12.9 and DIC = 621.5
## 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) 5.2746
##  Residual             3.0361
```

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

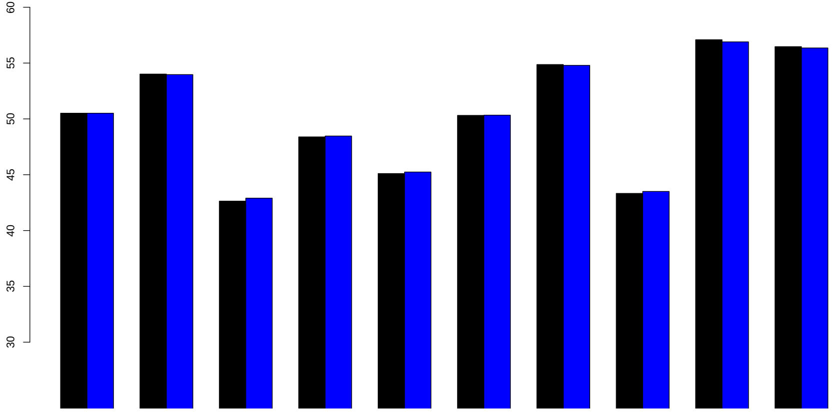
```
## [1] 3.073867
```

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

```
## [1] 5.916902
```

# 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=TRUE,col=c("blac
```



# 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

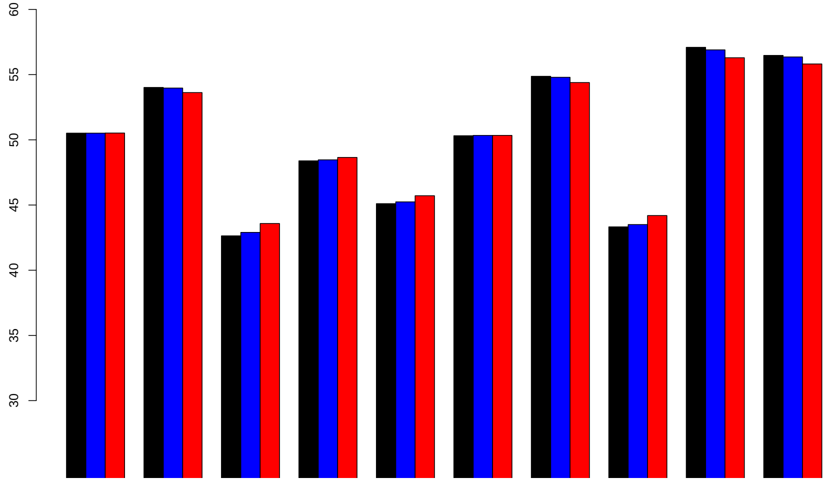
- $X \sim \text{Gamma}(a, b)$ with $a$ = shape, $b$ = rate $= \frac{1}{\theta}$ where $\theta$ is scale.
- Properties: $\mathbb{E}(X) = a\theta = 100/50 = 2$ and $\mathbb{V}(X) = a\theta^2 = \frac{100}{2500} = 0.04$ so that $\text{SD}(X) = 0.2$.

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

```
## [1] 2.497129
```

```
barplot(t(matrix(c(alpha_mean,alpha_mean2,alpha_mean3),ncol=3,nrow=10)),beside=TRUE
```

# 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