Bayesian Modelling Project

Bayesian tools applied to Radiocarbon Measurements

Elisa Quadrini

1. Introduction

The world of physics is broad and includes many different applications. This project aims to explore, using a Bayesian approach, the Material Science field that deals with dating various types of archaeological findings through the technique of radiocarbon (C14) measurement.

The idea is to build a regression model based on a dataset, "*p3k14c_2022.06*", taken from the official radiocarbon measurement website https://www.p3k14c.org, which collected data from several thousand findings from sites all over the world. The goal is to understand which of the following variables influence the errors associated with the estimation of the sample age, expressed in years before present, based on C14 (radiocarbon) measurements.

1.1 Dataset description

In order to carry out a meaningful Bayesian analysis, where the observed data do not dominate the posterior too strongly due to the huge sample size, and there is not a significant amount of missing data, I decided to focus on only two sites: one in France and the other in Jordan. In this way, I expect to create two groups of observations that show some differences due to the fact that the data, spanning from age of 8000 years BP to 30000 years BP, were collected in different parts of the world.

The total number of observation is $n = 175$, while the number of variables selected, limited to those considered to be meaningful, is *4*. Here follows the variables description:

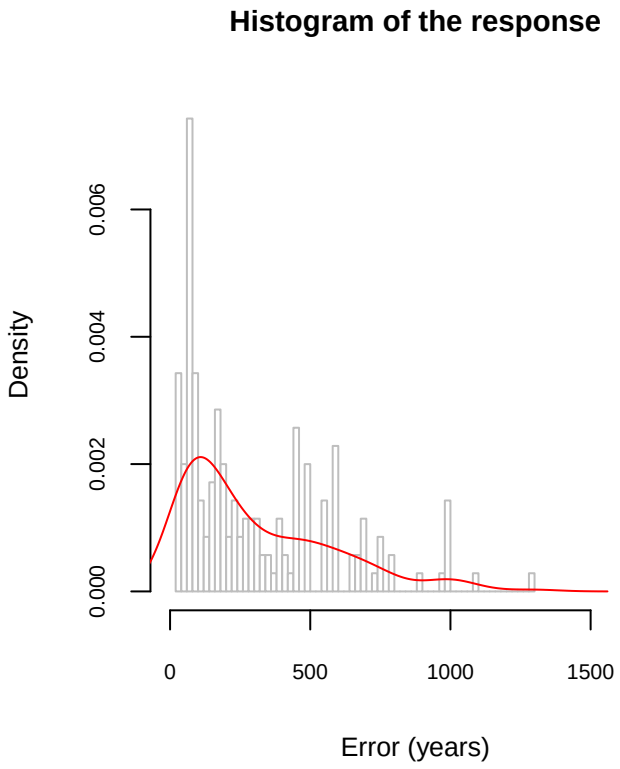
- **Age**: numerical variable; age of the fragment based on the C14 measurement, expressed in years before present.
- **Material**: categorical variable with three levels (*bone, charcoal, wood*); material of the sample.
- **Method**: categorical variable with three levels (*AMS, Radiometric, Unknown*); type of method used to perform C14 measurements. The label "Unknown" indicates that, for that observation, there is no available information about which method was used.
- **SiteName**: categorical variable with two levels (*Abri Pataud, Ain Ghazal*); site from which the sample comes.

The response variable we aim to model through linear regression is **Error**. This variable represents the one sigma standard error of the radiocarbon age measurement associated with the i-th observation, which could depend on various factors such as the treatment of the material before calibration, the age, the geographical location, and so on.

1.2 Goal of the analysis

Through this model, we aim to investigate whether there is a way to better understand which factors most strongly influence the measurement error, and whether something can be done to perform a more accurate estimation of the true age of the fragment.

In order to assess which model best fits our data, we first examine the distribution of the measurement errors we aim to model:



2. Model specification

Based on the plot, we observe that the distribution is right-skewed and defined on the positive real line. This suggests that a Generalized Linear Model (GLM) with a Gamma distribution and a suitable link function is appropriate.

After testing several link functions, the identity link proved to be the most suitable for capturing the relationship between the variables in my model. It allows for a direct interpretation of coefficients, but requires an explicit check for mean positivity and, in a Bayesian setting with priors on multiple parameters, may increase model flexibility to the point of producing an overly complex fit that reflects the specificities of the current dataset rather than general patterns.

We therefore adopt a Bayesian Gamma regression model of the form:

$$Y_i \sim \text{Gamma}(\alpha, \beta_i) \quad \text{with} \quad \mathbb{E}[Y_i] = \mu_i = \mathbf{x}_i^\top \boldsymbol{\beta}, \quad \beta_i = \frac{\alpha}{\mu_i}, \quad y_i > 0, \quad \alpha > 0, \quad \beta_i > 0$$

Then, given n observations, we derive the sampling distribution as

$$p(\mathbf{y} \mid \boldsymbol{\beta}, \alpha) = \prod_{i=1}^{n} \frac{(\alpha/\mu_i)^{\alpha}}{\Gamma(\alpha)} y_i^{\alpha-1} e^{-(\alpha/\mu_i)y_i}$$

In order to avoid computational problems, in the implementation of the algorithm we will use the log-sampling distribution instead:

$$\log p(\mathbf{y} \mid \boldsymbol{\beta}, \alpha) = \sum_{i=1}^{n} \left[\alpha \log\left(\frac{\alpha}{\mu_i}\right) + (\alpha - 1)\log y_i - \frac{\alpha}{\mu_i} y_i - \log\Gamma(\alpha) \right]$$

2.1 Prior Distributions

The Gamma distribution is parameterized in terms of a shape parameter α and a mean μ_i , which depends on the vector of covariates for the i-th observation x_i and the parameter vector β. So we need to assign two priors:

- $\boldsymbol{\beta} \sim \mathcal{N}_p(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ with density:

  $$p(\boldsymbol{\beta}) = \frac{1}{(2\pi)^{p/2}\,|\boldsymbol{\Sigma}_\beta|^{1/2}} \exp\left(-\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)^\top \boldsymbol{\Sigma}_\beta^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}_\beta) \right)$$

  with log transformation: $\log p(\boldsymbol{\beta}) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_\beta| - \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)^\top \boldsymbol{\Sigma}_\beta^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)$

- $\alpha \sim \text{Gamma}(a, b)$ with density:

  $$p(\alpha) = \frac{b^a}{\Gamma(a)}\, \alpha^{a-1} e^{-b\alpha}, \quad \alpha > 0$$

  with log transformation: $\log p(\alpha) = a\log b - \log\Gamma(a) + (a-1)\log\alpha - b\alpha$.

2.2 Posterior distribution

The joint posterior distribution is given by the Bayes theorem by the formula

$$p(\boldsymbol{\beta}, \alpha \mid \mathbf{y}) \propto p(\mathbf{y} \mid \boldsymbol{\beta}, \alpha) \cdot p(\boldsymbol{\beta}) \cdot p(\alpha)$$

The log-posterior is given by:

$$\log p(\boldsymbol{\beta}, \alpha \mid \mathbf{y}) \propto \log p(\mathbf{y} \mid \boldsymbol{\beta}, \alpha) + \log p(\boldsymbol{\beta}) + \log p(\alpha)$$

Where each term has been defined before.

3. Model implementation

To implement the model, since we do not have either conjugate or semi-conjugate priors, we need to adopt a Metropolis algorithm in order to perform posterior inference.

3.1 Metropolis Hastings for Posterior approximation

In this context, the Metropolis-Hastings scheme allows us to generate samples from the posterior distribution of the regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ and the rate parameter α by iteratively proposing candidate values for each parameter and accepting or rejecting them based on an acceptance probability.

At each iteration $s = 1, \ldots, S$ the following steps are carried out

1. For each parameter $j = 1, \ldots, p$:

   1.1 Propose new β_j:

   - If $s = 1$: draw from a normal centered at $\boldsymbol{\beta}^{(0)}$:

   $$\beta_j^* \sim \mathcal{N}(\beta_j^{(0)}, \sigma_j^{2(0)})$$

   We need to specify this case because in the first iteration we need the proposed value to be sampled from a normal with the initialized hyperparameters.

   - If $s > 1$: use an *adaptive proposal*, where each component β_j^* is drawn as:

   $$\beta_j^* \sim \mathcal{N}(m_j^{(s-1)},\ s_j^{2(s-1)}) \qquad \text{where} \qquad m_j^{(s-1)} = \frac{1}{s-1}\sum_{\ell=1}^{s-1}\beta_j^{(\ell)}, \quad s_j^{2(s-1)} = \frac{1}{s-1}\sum_{\ell=1}^{s-1}\left(\beta_j^{(\ell)} - m_j^{(s-1)}\right)^2$$

   This allows the algorithm to speed up the convergence, that would take otherwise a huge burn-in period.

   1.2 The acceptance ratio for β_j is:

   $$\log r_j = \log p(\beta_j^*, \boldsymbol{\beta_{-j}}^{(s-1)}, \alpha^{(s-1)} \mid \mathbf{y}) - \log p(\boldsymbol{\beta}^{(s-1)}, \alpha^{(s-1)} \mid \mathbf{y}) + \log q(\beta_j^{(s-1)} \mid \beta_j^*) - \log q(\beta_j^* \mid \beta_j^{(s-1)})$$

   1.3 Accept or reject the proposed parameter:

   $$\beta_j{}^{(s)} = \begin{cases} \beta^* & \text{with probability } \min(1, \exp(\log r_j)) \\ \beta_j{}^{(s-1)} & \text{otherwise} \end{cases}$$

2. Propose new α^* using a *Gamma adaptive proposal*:

   $$\alpha^* \sim \text{Gamma}(\delta, \delta/\alpha^{(s-1)})$$

   This maintains $\mathbb{E}[\alpha^*] = \alpha^{(s-1)}$, meaning that the new proposed value comes from a gamma distribution centered in the value of α of the previous iteration .

   2.1 Compute the log-posterior at the current and proposed values:

   $$\log p_{\text{curr}} = \log p(\boldsymbol{\beta}^{(s)}, \alpha^{(s-1)} \mid \mathbf{y}) \qquad \log p_{\text{prop}} = \log p(\boldsymbol{\beta}^{(s)}, \alpha^* \mid \mathbf{y})$$

   2.2 Compute the proposal density ratio (needed because the proposal for α is not symmetric):

   $$\log q_{\text{ratio}} = \log q(\alpha^{(s-1)} \mid \alpha^*) - \log q(\alpha^* \mid \alpha^{(s-1)}) \qquad \text{with} \quad q(\alpha^* \mid \alpha^{(s-1)}) = \text{Gamma}(\delta, \delta/\alpha^{(s-1)})$$

   2.3 Compute the Metropolis-Hastings acceptance log-ratio for α:

   $$\log r_\alpha = \log p_{\text{prop}} - \log p_{\text{curr}} + \log q_{\text{ratio}}$$

   2.4 Accept or reject the proposed parameter:

$$\alpha^{(s)} = \begin{cases} \alpha^* & \text{with probability } \min(1, \exp(\log r_\alpha)) \\ \alpha^{(s-1)} & \text{otherwise} \end{cases}$$

At the end of the process, we collect S samples:

$$\{(\boldsymbol{\beta}^{(s)}, \alpha^{(s)})\}_{s=1}^{S}$$

that we can use to approximate the joint posterior distribution of $(\boldsymbol{\beta}, \alpha \mid \boldsymbol{y})$.

I opted for non-informative priors by setting the hyperparameters $a = 1$ and $b = 0.01$, which yield a prior variance of 100^2 for the parameter α. For the regression coefficients $\boldsymbol{\beta}$, I specified a prior with mean vector equal to zero and covariance matrix $100I_p$, reflecting vague prior information. As initial values for $\boldsymbol{\beta}$ to facilitate convergence, I used the ordinary least squares estimates $\boldsymbol{\beta}_{\text{ols}}$ obtained from a standard linear regression model, while for the shape parameter α, I initialized with $\alpha_0 = 2$, since preliminary posterior credible intervals suggested this was a reasonable and sufficiently close starting point. The implementation of the algorithm is the following:

```r
# ----- INPUT -----

n <- nrow(X)
p <- ncol(X)
y <- mod_data$Error

# Priors
mu_beta <- rep(0, p)
Sigma_beta <- diag(100, p)
a <- 1
b <- 0.01

# MCMC settings
S <- 10000
beta_curr <- as.vector(lm(y ~ X - 1)$coefficients)
alpha_curr <- 2
samples <- matrix(NA, nrow = S, ncol = p+1)
colnames(samples) <- c(paste0("beta", 1:p), "alpha")

m_j <- rep(0, p)
s2_j <- rep(1, p)
delta <- 100


# ----- LOG POSTERIOR FUNCTION -----

log_posterior <- function(beta, alpha, y, X, mu_beta, Sigma_beta, a, b) {
  mu <- as.vector(X %*% beta)
  if (any(mu <= 0) || alpha <= 0) {
      cat("Invalid proposal: some mu <= 0 or alpha <= 0\n")
      return(-Inf)
    }
  loglik <- sum(dgamma(y, shape = alpha, rate = alpha / mu, log = TRUE))
  logprior_beta <- dmvnorm(beta, mean = mu_beta, sigma = Sigma_beta, log = TRUE)
  logprior_alpha <- dgamma(alpha, shape = a, rate = b, log = TRUE)
  return(loglik + logprior_beta + logprior_alpha)
```

```
}

# ----- MCMC LOOP -----
set.seed(42)

for (s in 1:S) {

  # Proposal for each component of beta
  for (j in 1:p) {

    if (s > 1) {
      m_j[j] <- mean(samples[1:(s - 1), j])
      s2_j[j] <- var(samples[1:(s - 1), j])
      if (is.na(s2_j[j]) || s2_j[j] == 0) s2_j[j] <- 1e-6
      beta_j_prop <- rnorm(1, mean = m_j[j], sd = sqrt(s2_j[j]))
    } else {
      beta_j_prop <- rnorm(1, mean = beta_curr[j], sd = sqrt(0.01))
    }

    beta_prop <- beta_curr
    beta_prop[j] <- beta_j_prop

    log_post_curr <- log_posterior(beta_curr, alpha_curr, y, X, mu_beta, Sigma_beta, a, b)
    log_post_prop <- log_posterior(beta_prop, alpha_curr, y, X, mu_beta, Sigma_beta, a, b)

    log_q_ratio <- dnorm(beta_curr[j], mean = m_j[j], sd = sqrt(s2_j[j]), log = TRUE) -
               dnorm(beta_j_prop, mean = m_j[j], sd = sqrt(s2_j[j]), log = TRUE)

    log_r <- log_post_prop - log_post_curr + log_q_ratio

    if (is.finite(log_r) && log(runif(1)) < log_r) { beta_curr[j] <- beta_j_prop }
  }

  # Proposal for alpha
  alpha_prop <- rgamma(1, shape = delta, rate = delta / alpha_curr)

  log_post_curr <- log_posterior(beta_curr, alpha_curr, y, X, mu_beta, Sigma_beta, a, b)
  log_post_prop <- log_posterior(beta_curr, alpha_prop, y, X, mu_beta, Sigma_beta, a, b)

  log_q_ratio <- dgamma(alpha_curr, shape = delta, rate = delta / alpha_prop, log = TRUE) -
                 dgamma(alpha_prop, shape = delta, rate = delta / alpha_curr, log = TRUE)

  log_r_alpha <- log_post_prop - log_post_curr + log_q_ratio

  if (is.finite(log_r_alpha) && log(runif(1)) < log_r_alpha) {
    alpha_curr <- alpha_prop
  }

  # ----- OUTPUT -----
  samples[s, ] <- c(beta_curr, alpha_curr)
}
thin <- 10
samples_thin <- samples[seq(1, nrow(samples), by = thin), ]
```

```
##                          mean CI_lower.2.5% CI_upper.97.5%
## (Intercept)        252.701434    252.697071     252.705340
## Age                217.281209    217.231731     217.342822
## MethodRadiometric   25.815447     25.663442      26.023557
## MethodUnknown       92.945620     92.630407      93.132490
## SiteNameAin Ghazal 121.332513    121.186385     121.458119
## Materialcharcoal   -47.064736    -47.342892     -46.882544
## Materialwood       -61.608817    -62.296117     -60.794302
## alpha                2.701465      2.206536       3.263313
```

From the posterior summary, we notice t that the variability of the estimated parameters is very low, as shown by the narrow 95% credible intervals; such behavior is expected given the use of adaptive proposal distributions, which are centered on the current estimates and help improve convergence by reducing unnecessary exploration of low-probability regions.

3.2 Spike and Slab variable selection

First of all, since we are considering a GLM model, let's perform a variable selection in order to assess if all the variables considered are significant. The method applied, through a JAGS implementation due to the complexity of the code, is a spike and slab procedure, on which we impose a prior of the form

$$\gamma_j \sim \text{Bern}(w), \quad \beta_j \mid \gamma_j = 1 \sim \mathcal{N}(\beta_{0j}, \sigma_{0j}^2), \quad \beta_j \mid \gamma_j = 0 = 0$$

This hierarchical prior allows for automatic variable selection: if $\gamma_j = 0$, the corresponding coefficient β_j is exactly zero (the "spike"); otherwise, it comes from a normal distribution centered at β_{0j} (the "slab"). The resulting joint prior can be written as

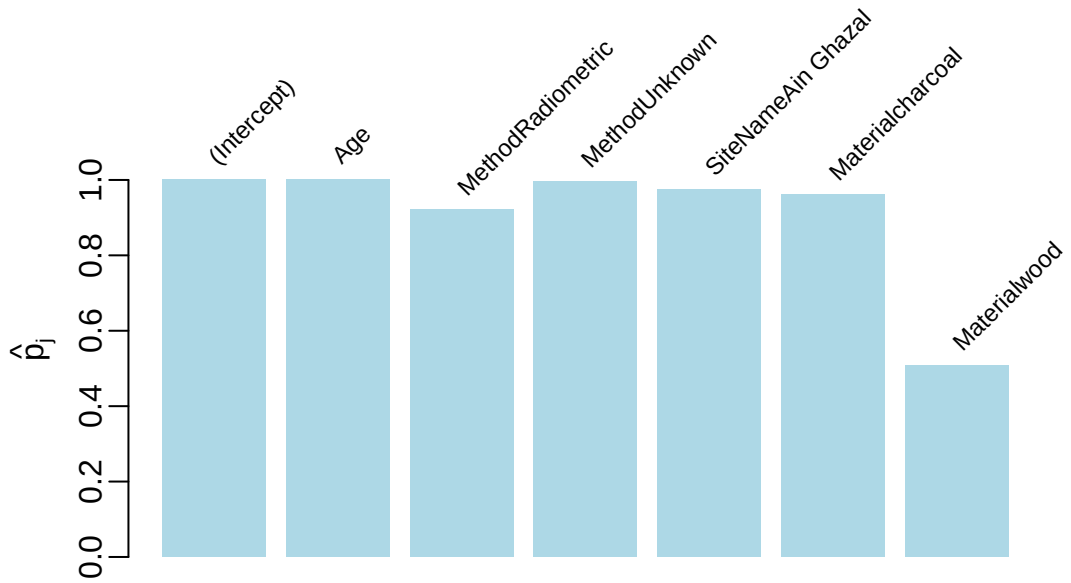
$$p(\beta_j, \gamma_j) = (1 - w)\, \delta_0 + w\, \mathcal{N}(\beta_j \mid \beta_{0j}, \sigma_{0j}^2)$$

where δ_0 is a point mass at zero. We fix $w = 0.5$ in order to be non informative.

```
jags_data <- list(y = y, X = X, n = n, p = p,
  w = 0.5,          # inclusion prior
  sigma2 = 0.01,      # slab variance
  a0 = 1,           # prior shape for alpha
  b0 = 0.01           # prior rate for alpha
)

model_string <- "
model {
  for (i in 1:n) {
    mu_raw[i] <- inprod(X[i, ], beta[])
    mu[i] <- max(mu_raw[i], 1e-3)
    y[i] ~ dgamma(alpha, alpha / mu[i])
  }

  for (j in 1:p) {
    gamma[j] ~ dbern(w)
    theta[j] ~ dnorm(0, 1 / sigma2)
    beta[j] <- gamma[j] * theta[j]
  }

  alpha ~ dgamma(a0, b0)
```

```
}
"

params <- c("beta", "gamma", "alpha")

inits <- function() {
  flag <- TRUE
  while (flag) {
    gamma <- rbinom(p, 1, 0.5)
    theta <- rnorm(p, 0, 1)
    beta <- gamma * theta
    mu <- as.vector(X %*% beta)

    flag <- all(mu > 0)
  }
  return(list(gamma = gamma, theta = theta, alpha = 1))
}

model_out <- jags(data = jags_data, inits = inits, parameters.to.save = params,
                  model.file = textConnection(model_string), n.chains = 2, n.iter = 10000, n.thin = 10)
```

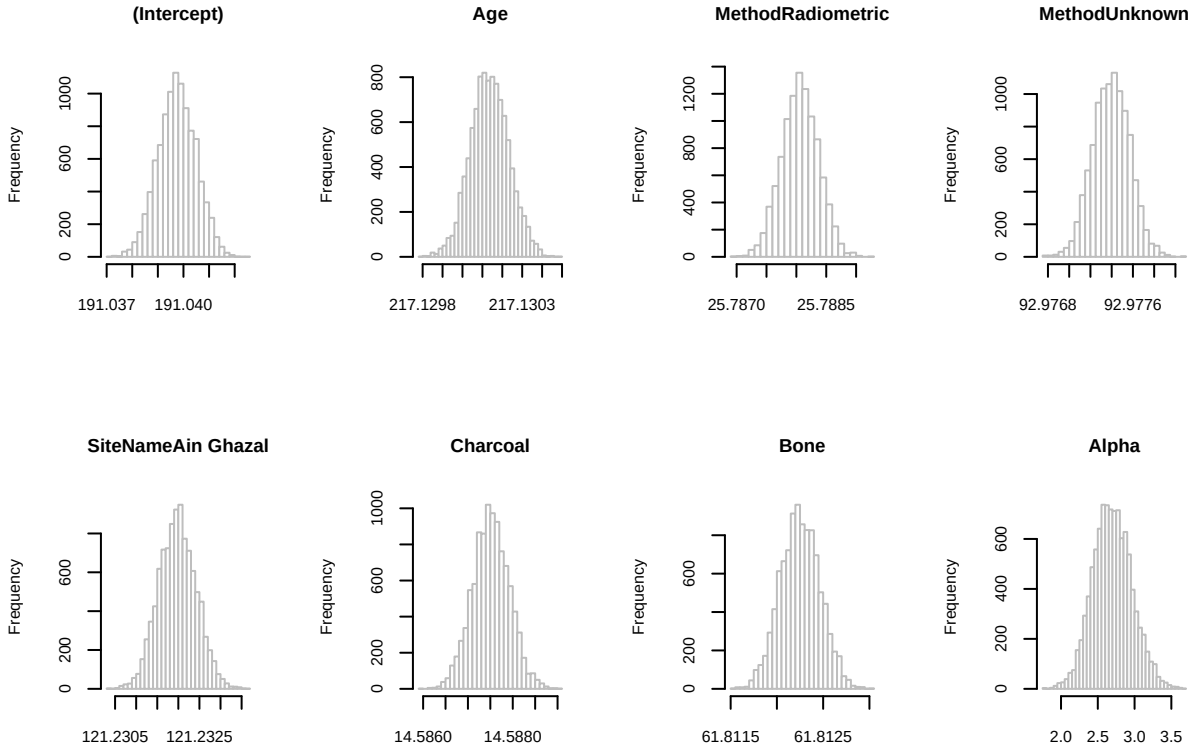
Notice that in this algorithm, since the linear predictor appears in the denominator of the rate parameter, we need to ensure that it remains strictly greater than zero significantly; otherwise, it would lead to computational issues.



We will implement the model using the same algorithm as before, but excluding the *wood* level of the categorical variable *Material*, since the spike and slab variable selection indicates that its posterior inclusion

probability is significantly lower than that of the other covariates, where $\hat{p}_{x_j} = \frac{1}{S}\sum_{s=1}^{S}\gamma_j^{(s)}$ is the approximated posterior distribution of model M_k.

```
# ----- Best model ------
mod_best <- mod_data[, c(1,2,3,4)]
mod_best$Charcoal <- ifelse(mod_data$Material == "charcoal", 1, 0)
mod_best$Bone <- ifelse(mod_data$Material == "bone", 1, 0)
```



The histograms representing the posterior distribution show highly concentrated estimates as the model implemented before, with even more extremely narrow 95% credible intervals, often at the fourth decimal. Alpha is more spreadly distributed: this comes from the fact that the β coefficients are directly associated with covariates and each observation provides information to update them. This stronger data influence leads to narrower posteriors for the betas, while alpha, having less direct information from the data, remains more uncertain.

Notice that all the covariates have a positive effect on the posterior expectation of intercept and no histogram contains the value 0, meaning that all the coefficients seem significant. The variable *Age* seems to be the one that most intact the expected value of the error, keeping the other variables constant.

These results are obtained under weakly informative priors, specifically a multivariate normal prior with large variance for $\boldsymbol{\beta}$, and a Gamma(1, 0.01) prior for α. The reduced uncertainty in the posterior is a result of the data's strong explanatory power, while the adaptive proposal mechanism improved the efficiency of the MCMC algorithm by focusing the exploration around high-density regions of the posterior.

What is missing is a meaningful interpretation of the results: from a scientific perspective, it is known that bone fragments tend to show lower measurement errors due to the type of treatment they undergo, which typically involves a larger amount of material compared to charcoal samples; this is not what is shown by the distribution of the coefficients. Moreover, the AMS method is often considered less precise than the

radiometric one, although this may depend on the type of fragment and the specific laboratory where the analysis was conducted.

Now let's inspect the goodness of the model:

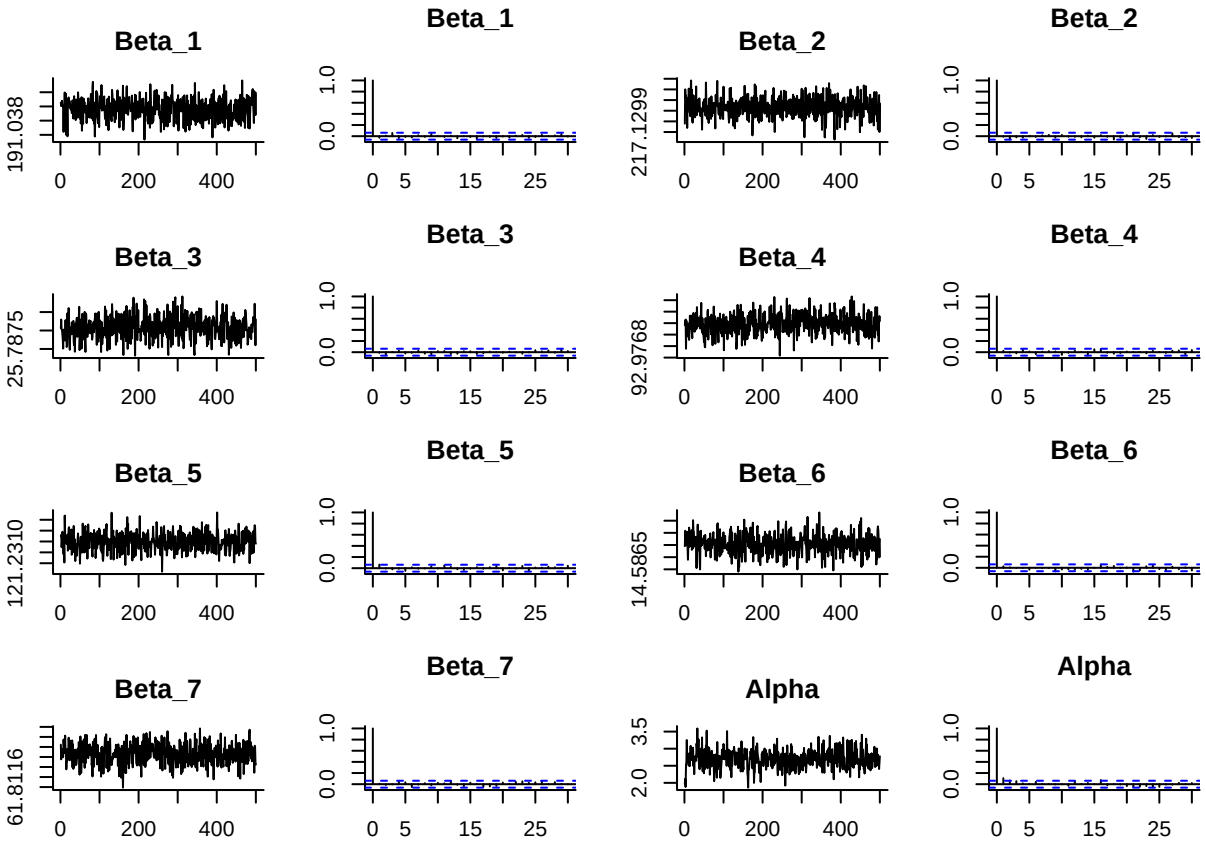
4. Diagnostic

4.1 Convergence: traceplots and autocorrelation

In Markov Chain Monte Carlo methods, the sequence of draws from the posterior distribution is dependent by construction: each draw depends on the one of the previous iteration.

But when we approximate a distribution empirically we would ideally use independent and identically distributed (i.i.d.) samples. Dependence within the chain reduces the effective number of independent draws, leading artificially narrow credible intervals because of the fact that the current iteration is not going to be far from the previous one. We need to investigate this aspect because the posterior histogram show very narrow credible intervals, which it may indicate either that this issue is actually occurring or alternatively that the algorithm explores only areas with high posterior probability.

Let's first inspect the trace plot of each sampled parameter, which is a graphical representation of the chain, along with the graphical representation of the autocorrelation coefficients:



What we aim to obtain are chains that do not exhibit any noticeable trends and are concentrated in a region with high posterior probability, as shown in these plots. This behavior is achieved thanks to the thinning that we have already applied to the chains in the MH algorithm, meaning that only one draw every k=10 iterations was stored, where k is the thinning interval.

Let's evaluate the *Effective Sample Size* to detect possible autocorrelation in the chains, and apply the *Geweke test* to assess whether a burn-in period is needed for the Markov chain to converge:

```
##     beta1     beta2     beta3     beta4     beta5     beta6     beta7     alpha
## 1000.0000 1000.0000 1000.0000 1000.0000  877.4044 1000.0000 1000.0000  718.9366
```

```
## Total number of iterations: 1000
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    beta1    beta2    beta3    beta4    beta5    beta6    beta7    alpha
##  1.79584  0.01406 -1.04626 -2.74376 -0.17056  2.63258  1.60117 -0.06160
```

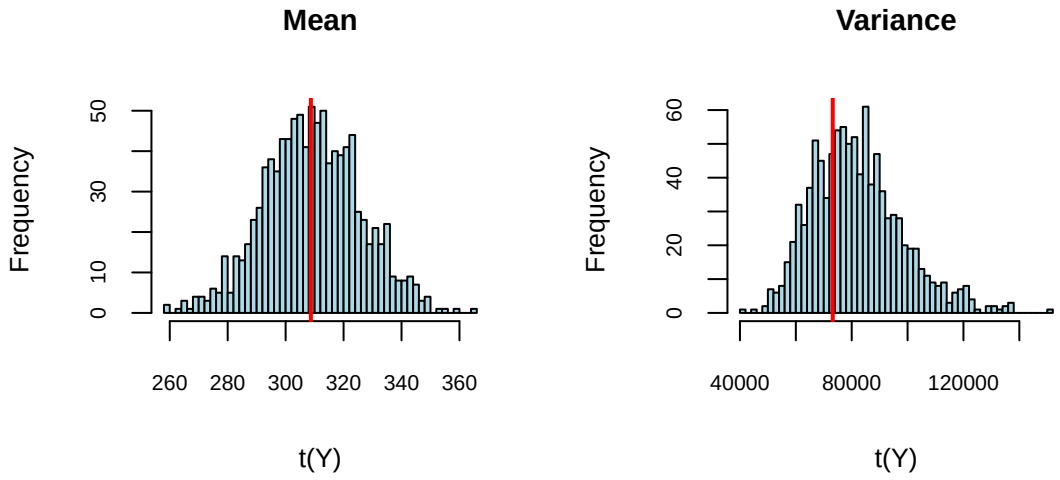
It seems that there are no major issues with the chain: only β_5 and α have an ESS that is lower than the total number of iterations but the Geweke test doesn't suggest rejecting the null hypothesis of convergence. This result was expected, as the proposal distributions and the algorithm initialization used in the model were designed to ensure fast convergence.

4.2 MC evaluation through Posterior Predictive model checking

A useful tool we can exploit once we have sample with S draws from the posterior distribution is to assess the goodness of the fit of the implemented model with respect to the observed data through a Monte Carlo approach.

We focus on specific statistics: in this case we will consider the mean and the variance. We compare the observed value of each statistic with its distribution empyrically constructed on S simulated samples from the approximated Posterior Predictive distribution of the model: if the observed value falls in the tails, the model may not fit well.

```r
# ----- PREDICTIVE SIMULATION -----
mu <- X %*% t(post.beta) # linear predictor simulated
y_pred <- matrix(NA, nrow = S, ncol = n)

# simulations from Gamma(alpha, rate = alpha / mu)
for (s in 1:S) {
  y_pred[s, ] <- rgamma(n, shape = post.alpha[s], rate = post.alpha[s]/mu[, s])
}
y_pred <- y_pred[seq(1, nrow(y_pred), by = 10), ]
```

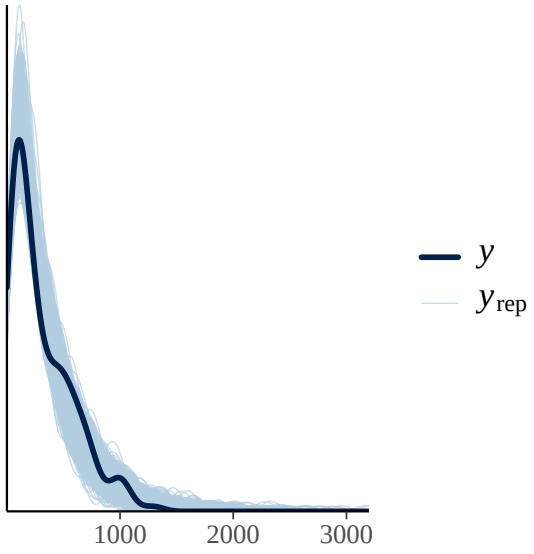


Notice that this approach cannot be taken as an overall measure of goodness, because it depends on the specific statistic we are considering. For this reason, I choose to use two statistics that in my opinion sufficiently summarize the posterior predictive distribution.

Since in both cases the observed statistic lies in the center of the distribution, the model seems reasonable for the data.

4.3 Predictive checks

Another tool to assess the goodness of the model is to visualize through the *ppc_dens_overlay* function if the observed response is well represented from the simulated data coming from the posterior predictive distribution:



From the plot we can conclude that the simulated data covers well the data used to fit the model, as the repetitions of the simulated y as the same form of the density of the observed y.

5. Conclusion

From our Bayesian model, we can conclude that we are able to reduce the measurement error associated with the radiocarbon (^{14}C) age. That is, our model helps improve the precision of the *estimated radiocarbon age* by modeling the error component and leveraging covariate information.

However, we cannot say the same for the true calendar age of the samples.

The model we constructed does not translate into a reduction of the uncertainty associated with the true calendar age of the samples: radiocarbon dating provides an estimated radiocarbon age based on the residual amount of ^{14}C in organic materials, the so called BP age, using its known exponential decay rate (with a half-life of approximately 5,730 years). Yet, because the atmospheric concentration of ^{14}C has varied over time due to natural and anthropogenic factors (such as solar activity and fossil fuel emissions), radiocarbon ages must be calibrated to obtain calendar ages.

This calibration is performed using externally derived calibration curves (e.g., IntCal20), which relate radiocarbon years to calendar years based on independent dating methods such as dendrochronology. The process is inherently nonlinear and introduces additional uncertainty: a single radiocarbon measurement may correspond to multiple possible calendar dates, especially in regions of the curve where atmospheric ^{14}C levels doesn't change rapidly.

Therefore, even though our model improves the precision of the radiocarbon measurement, the total uncertainty on the true calendar age remains limited by the calibration process itself and cannot be reduced solely through statistical modeling of the measurement error.

6. Extra: DAG of the Bayesian Radiocarbon Model

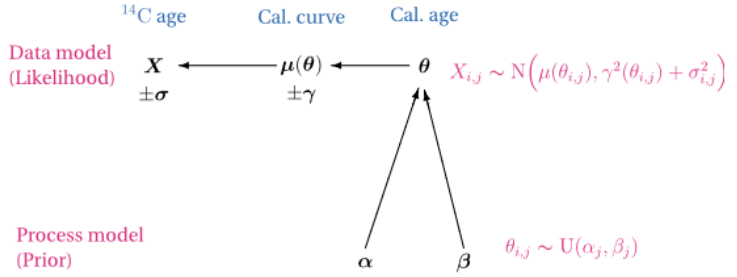
Reference: "Archaeometry - 2024 - Buck - Bayesian radiocarbon modelling for beginners" https://onlinelibrary.wiley.com/doi/10.1111/arcm.12998

The Bayesian radiocarbon model links the true (unknown) calendar ages of samples to the observed radiocarbon measurements, incorporating both measurement error and calibration uncertainty. It also allows the inclusion of stratigraphic prior information.

We define the following quantities:

$\theta_{i,j}$: true calendar age of sample i in context j; $\mu(\theta_{i,j})$: expected radiocarbon age from the calibration curve at $\theta_{i,j}$; $\gamma(\theta_{i,j})$: calibration curve uncertainty at $\theta_{i,j}$; $\sigma_{i,j}$: laboratory error for sample i; $x_{i,j}$: observed radiocarbon age (14C date) with the associated measurement error $\sigma_{i,j}$, realization of the r.v. of the real radiocarbon age $X_{i,j}$; α_j, β_j: start and end dates (calendar BP) of context j.

The dependencies between these variables can be summarized with the following DAG, known a priori:



`Prior on Calendar Dates:` If sample i is known to be within context j, we assume:

$$\theta_{i,j} \sim \text{Uniform}(\beta_j, \alpha_j) \quad (\text{with } \beta_j < \theta_{i,j} < \alpha_j)$$

Or, if stratigraphy gives ordering (e.g., samples $1 < 2 < 3$), we encode:

$$p(\boldsymbol{\theta}) \propto \mathbb{I}(\theta_3 > \theta_2 > \theta_1)$$

`Calibration Curve:` At each $\theta_{i,j}$, we get the expected radiocarbon age and error from the calibration curve through software like Oxcal:

$$\mu(\theta_{i,j}) \quad \text{and} \quad \gamma(\theta_{i,j})$$

`Likelihood (Measurement Model):` The observed radiocarbon age is modeled as:

$$x_{i,j} \sim \mathcal{N}\left(\mu(\theta_{i,j}), \sigma_{i,j}^2 + \gamma^2(\theta_{i,j})\right)$$

This reflects both the lab error and the uncertainty in the calibration curve at that calendar date (it is a normally distributed node).

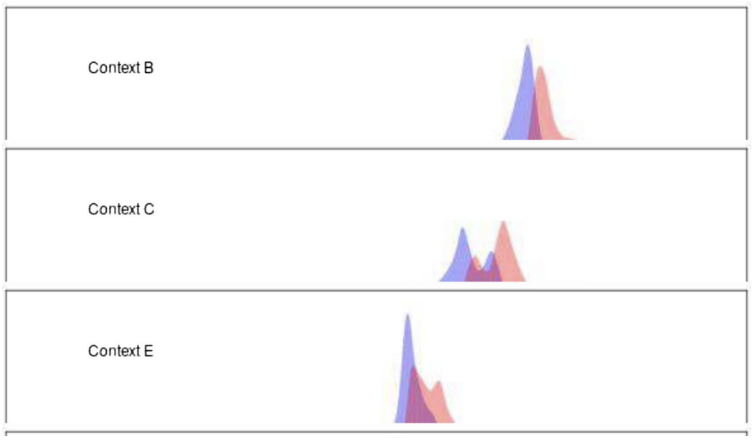
6.1 Posterior Inference

Combining prior and likelihood using Bayes' theorem: $p(\boldsymbol{\theta} \mid \boldsymbol{x}) \propto \prod_i \exp\left\{-\frac{(x_i-\mu(\theta_i))^2}{2(\sigma_i^2+\gamma^2(\theta_i))}\right\} \cdot p(\boldsymbol{\theta})$

and $p(\boldsymbol{\theta})$ includes stratigraphic priors or context boundaries.

Since the posterior cannot be calculated in closed form, we use Markov Chain Monte Carlo (MCMC) methods (e.g., via software like OxCal) to simulate from the posterior distribution of $\boldsymbol{\theta}$, allowing us to obtain credible intervals and probabilistic estimates of calendar ages.

A visual example:



This figure compares the posterior estimates for the start (*blue*) and end (*red*) calibrated dates of stratigraphic contexts B, C, E. The Bayesian model includes stratigraphic ordering, allowing the estimation of context to significantly improve, with some boundaries estimated to within 100–500 years.