Bayesian Inference Project - Master in Statistics for Data Science 2024/25

Carlos Fernandez Pascual Antia Enriquez Yurrebaso
Miguel Santos Pascual Nicolas Carrizosa Arias

Abalone Dataset

Abalones, also known as ear shells or sea ears, are marine gastropod mollusks from the family *Haliotidae*. The inner layer of their shell is composed of nacre, which in many cases is highly iridescent, giving rise to a range of strong, changeable colors, which make the shells attractive as decorative elements or as material for jewelry. It is found that the economic value of the abalone is directly related to their age. However, determining the age of abalone is a cumbersome and expensive process that consists on cutting the shell through the cone, staining it, and then counting the number of rings through a microscope.

References

https://archive.ics.uci.edu/dataset/1/abalone https://en.wikipedia.org/wiki/Abalone

http://ijeais.org/wp-content/uploads/2020/11/IJAAR201103.pdf

https://mpra.ub.uni-muenchen.de/91210/1/MPRA_paper_91210.pdf

Objective

The objective of this case study is to predict the number of rings found in the shell of an abalone by performing bayesian regression on simple measurements of the shell.

Dataset

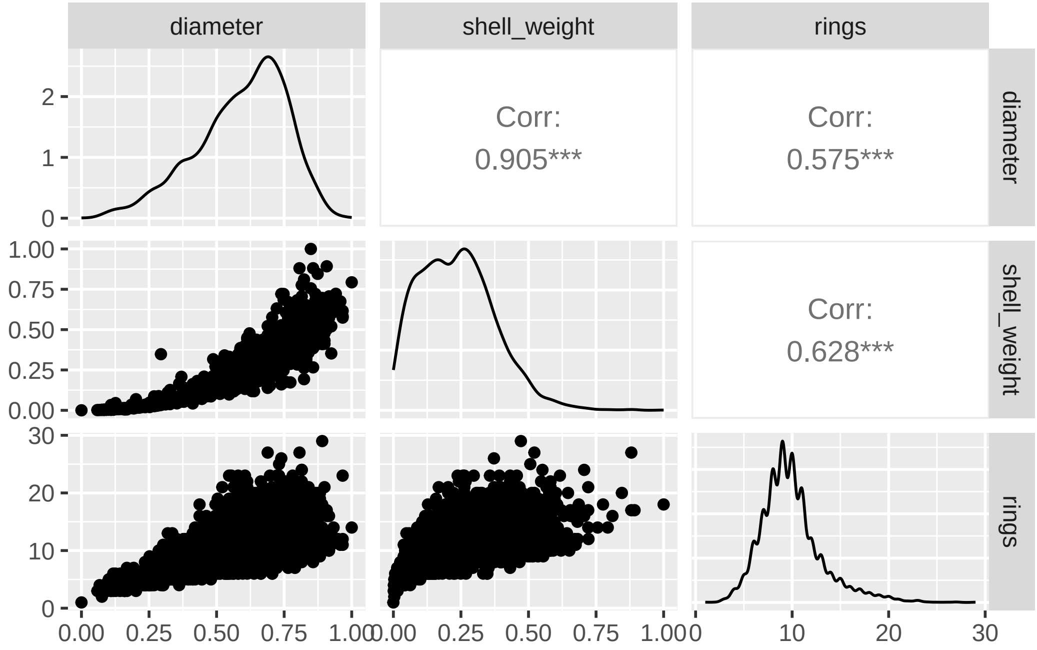
For the analysis, data was taken from the *Abalone* dataset, with measurements from 4000 mollusks. The following variables from the dataset were considered:

- **Diameter (mm)**: the length of the shell perpendicular to its main axis.
- **Shell weight (g)**: weight of the shell, after being emptied and dried.
- **Rings**: number of rings in the shell.

The reading and preprocessing of the data was performed through the following code:

```r
# Load necessary libraries
library(ggplot2)
library(rjags)
library(bayesplot)
library(gridExtra)

# Read the dataset
df <- read.csv("abalone.data", header=FALSE)

# Define column names
colnames <- c("sex","length","diameter","height",
              "whole_weight","shucked_weight","viscera_weight",
              "shell_weight","rings")
colnames(df) <- colnames

# Shuffle observations
df <- df[sample(nrow(df)), ]

# Select variables
df <- df[, c("diameter", "shell_weight", "rings")]

# Min-Max normalization function
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# Apply normalization
numeric_variables <- c("diameter", "shell_weight")
df[numeric_variables] <- as.data.frame(lapply(df[numeric_variables], normalize))

# Ensure the dependent variable is integer
df$rings <- as.integer(df$rings)

# Define variables as vectors
x1 <- df$diameter
x2 <- df$shell_weight
y <- df$rings
```

The predictor variables were normalized so that the results of the model wouldn't be hindered by the difference in scale of said covariates. The new variables result are shown below.



It is important to note that all correlations are clearly different from zero, and particularly positive. This prior knowledge will guide the choice of priors in the Bayesian approach, favoring positive associations between the predictors and the objective variable. In addition, the correlation between the two predictors is approximately 0.9, which is considerably high and must be taken into account during the modeling process, as it may influence parameter estimation and model interpretation.

Model employed

The objective of this work was to predict the number of rings in the shell of abalone from data from their physical dimensions.

In order to do so, it was assumed that the number of rings, y, follow a Poisson distribution of parameter lambda, as the number of rings is an integer variable that can be understood as a number of counts:

$$y_i \sim Pois(\lambda_i),\, with\, \lambda \in (0, \infty), \quad \forall\, i \in \{1, ..., n\},$$

where n is the total number of individuals.

From this assumption, the goal was to predict λ through a Poisson regression of y based on the values from the predictor variables (remember that the expected value for a Poisson variable is the parameter of the distribution):

$$\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon_i = \beta^T \mathbf{x} + \epsilon_i \iff \lambda_i = e^{\beta^T \mathbf{x} + \epsilon_i}, \quad \forall i \in \{1, \dots, n\}$$

where ϵ_i has been introduced as an uncertainty for the model to better adjust to a real life scenario.

As prior distributions for the parameters of the model, since there is no information about them, they were assumed to be the following. However, as it was thought to have positive correlations, it was decided to set the mean on 0.5.

$$\beta_i \sim N(0.5, 1), \quad \forall, i \in 1, , ..., , n, \epsilon_i \sim N(0, \tau), \quad \forall, i \in 1, , ..., , n, \tau \sim \Gamma(1, 1),$$

where $\tau = \frac{1}{\sigma^2}$ is the precision.

From this, the likelihood obtained is the following:

$$\begin{aligned}
\mathbb{P}(data|\beta_0, \beta_1, \beta_2, \sigma^2, \epsilon) &= \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i, X_i = x_i|\beta_0, \beta_1, \beta_2, \sigma^2, \epsilon) \\
&= \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i|X_i = x_i, \beta_0, \beta_1, \beta_2, \sigma^2, \epsilon)\mathbb{P}(X_i = x_i|\beta_0, \beta_1, \beta_2, \sigma^2, \epsilon) \\
&= \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i|\lambda_i = e^{\beta^t x}) \cdot 1 \\
&= \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i|\lambda_i = e^{\beta^t x})
\end{aligned}$$

Finally, the posterior distribution is found to be proportional to the product of the likelihood function times the prior distribution:

$$f(\theta|data) \propto \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i|\theta) f_\theta(\theta),$$

where $\theta = (\beta_0, \beta_1, \beta_2, \sigma^2, \epsilon)$ are the parameters of the model.

Since this is very difficult to obtain an analytical form for this posterior distribution, the Gibbs Sampler method was employed through the JAGS language (Just Another Gibbs Sampler) by means of the pakcage *rjags*.

Implementation of the model

The following code was employed to implement the model in R:

```
# Define the JAGS model
model_string <- "
model {
  for (i in 1:N) {
    y[i] ~ dpois(lambda[i])  # Likelihood
    # Log-link with error
    log(lambda[i]) <- beta0 + beta1*x1[i] + beta2*x2[i] + epsilon[i]

    epsilon[i] ~ dnorm(0, tau)  # Normally distributed error term
  }

  # Priors for the coefficients
  beta0 ~ dnorm(0.5, 1)
  beta1 ~ dnorm(0.5, 1)
  beta2 ~ dnorm(0.5, 1)

  # Precision (inverse of variance) for error term
  tau ~ dgamma(1, 1)

  # Convert precision to standard deviation
  sigma <- 1 / sqrt(tau)
}
"

# Save the model to a temporary file
writeLines(model_string, con = "model.jags")

# Prepare data for JAGS
data_jags <- list(y = as.vector(y), x1 = x1, x2 = x2, N = length(y))

# Initial values
inits <- function() list(beta0 = rnorm(1, 0, 1), beta1 = rnorm(1, 0, 1),
                         beta2 = rnorm(1, 0, 1), tau = rgamma(1, 1, 1))

# Load the model into JAGS (one chain)
jags_model <- jags.model("model.jags", data = data_jags, inits = inits,
                         n.chains = 1, n.adapt = 1000)

# Extract samples using MCMC
update(jags_model, 1000)  # Burn-in
samples <- coda.samples(jags_model,
                        variable.names = c("beta0", "beta1", "beta2", "sigma"),
                        n.iter = 100000)
```

```
# Summary of results
summary(samples)
```

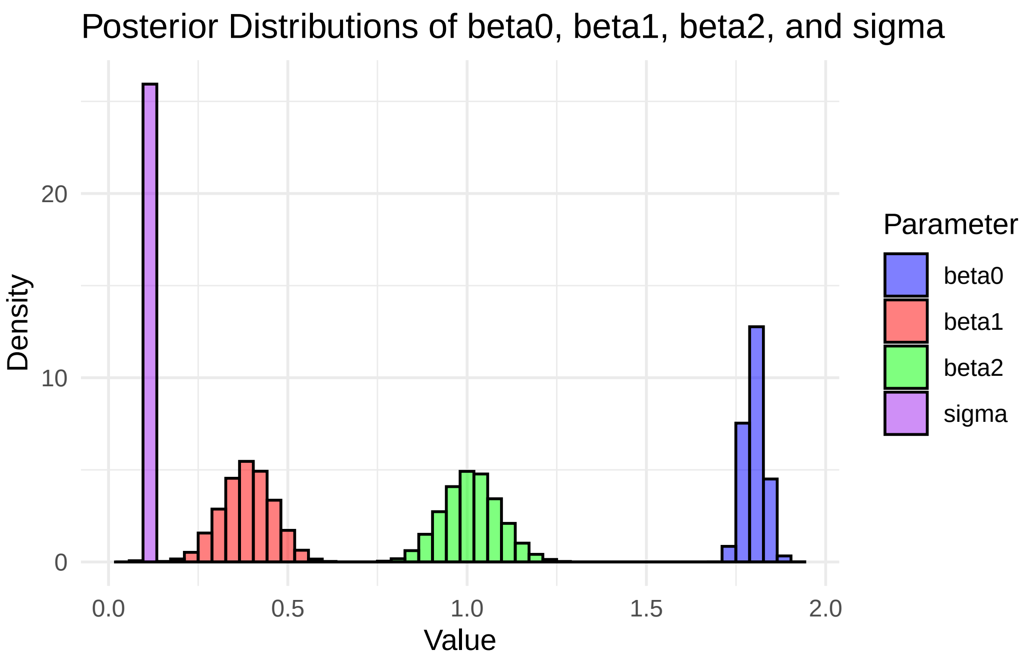
```
Iterations = 2001:102000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1e+05

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

        Mean       SD  Naive SE Time-series SE
beta0 1.8012 0.028307 8.952e-05      0.0012101
beta1 0.3877 0.072067 2.279e-04      0.0034688
beta2 1.0107 0.079522 2.515e-04      0.0032064
sigma 0.1065 0.003889 1.230e-05      0.0000717

2. Quantiles for each variable:

         2.5%    25%    50%    75%  97.5%
beta0 1.74629 1.7821 1.8009 1.8202 1.8572
beta1 0.24628 0.3397 0.3882 0.4371 0.5256
beta2 0.85679 0.9568 1.0109 1.0636 1.1691
sigma 0.09916 0.1039 0.1065 0.1092 0.1143
```

To feed the JAGS with the model, it needs to be written as a string, where the equations of the model are declared, as well as the prior distributions for the parameters. The data also needs to be properly prepared, and initial values for the parameters of the model need to be provided. Then, the model is called using as arguments the string with the formulation of the model, the initial values of the parameters and the data. Finally, samples are extracted from the model using a Monte Carlo Markov Chain. A burn-in of 1000 iterations was chosen to start obtaining samples in a region where the model is already stabilized, and then 1000000 samples were obtained.

From the summary of the obtained samples, we can see the mean values for each parameters as well as their standard deviation, and the confidence intervals based on the quantiles of the sample obtained (the most external values define the 95% confidence interval for the parameters).

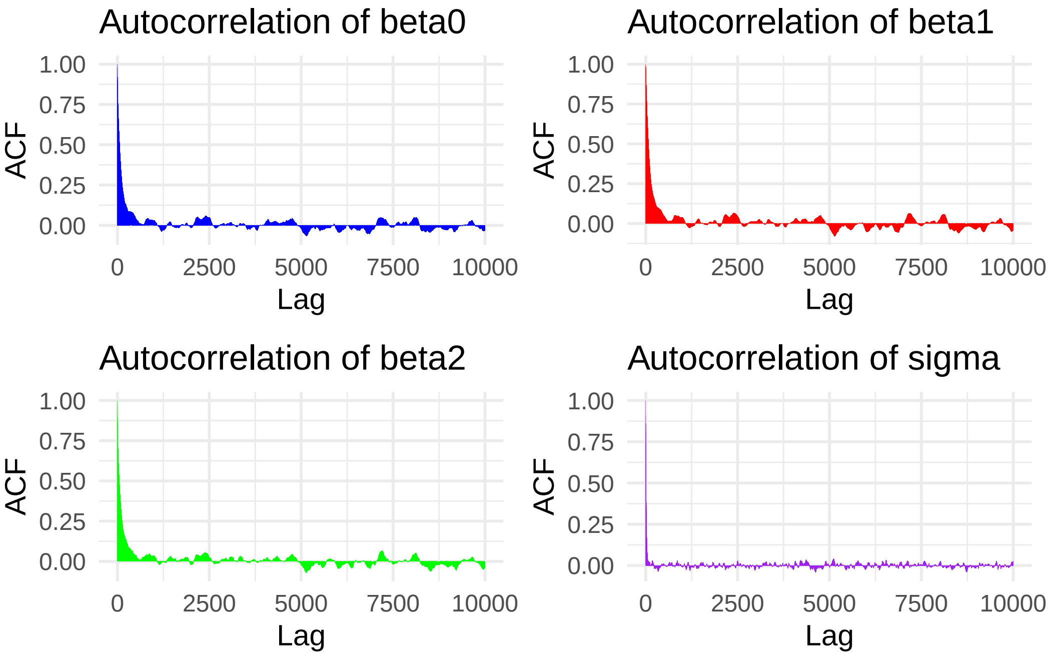
```
# Convert samples to dataframe
mcmc_samples <- as.data.frame(as.matrix(samples[[1]]))

# Rename columns for better visualization
colnames(mcmc_samples) <- c("beta0", "beta1", "beta2", "sigma")
```

```
# Histogram of the posteriors
ggplot(mcmc_samples) +
  geom_histogram(aes(x = beta0, y = ..density.., fill = "beta0"),
                 bins = 50, alpha = 0.5, color = "black") +
  geom_histogram(aes(x = beta1, y = ..density.., fill = "beta1"),
                 bins = 50, alpha = 0.5, color = "black") +
  geom_histogram(aes(x = beta2, y = ..density.., fill = "beta2"),
                 bins = 50, alpha = 0.5, color = "black") +
  geom_histogram(aes(x = sigma, y = ..density.., fill = "sigma"),
                 bins = 50, alpha = 0.5, color = "black") +
  scale_fill_manual(values = c("beta0" = "blue", "beta1" = "red",
                              "beta2" = "green", "sigma" = "purple")) +
  labs(title = "Posterior Distributions of beta0, beta1, beta2, and sigma",
       x = "Value", y = "Density", fill = "Parameter") +
  theme_minimal()
```



```
# ACF plots
acf_plot <- function(samples, param_name, color, max_lag = 10000) {
  acf_result <- acf(samples, plot = FALSE, lag.max = max_lag)
  acf_data <- data.frame(lag = acf_result$lag, acf = acf_result$acf)

  ggplot(acf_data, aes(x = lag, y = acf)) +
    geom_bar(stat = "identity", fill = color, alpha = 0.6) +
    labs(title = paste("Autocorrelation of", param_name), x = "Lag", y = "ACF") +
    theme_minimal()
}
```

```
acf_beta0 <- acf_plot(mcmc_samples$beta0, "beta0", "blue")
acf_beta1 <- acf_plot(mcmc_samples$beta1, "beta1", "red")
acf_beta2 <- acf_plot(mcmc_samples$beta2, "beta2", "green")
acf_sigma <- acf_plot(mcmc_samples$sigma, "sigma", "purple")
grid.arrange(acf_beta0, acf_beta1, acf_beta2, acf_sigma, ncol = 2, nrow = 2)
```



From the plot of the posterior sampled distributions for the coefficients, two main conclusions can be drawn. First of all, all β coefficients of the model are positive. From this, it can be concluded that all the covariates studied (*shell diameter* and *shell weight*) have a positive marginal effect on the number of rings present in the shell of the abalote, that is, if one is held constant, when the other one increases, the expected number of rings increases. This happens because the model fitted is a regression on the parameter λ, the expected value for the variable y (the number of rings), which follows a Poisson distribution of parameter λ.

This conclusion makes sense, as, the bigger the values for those covariates, the bigger is the shell (both variables refer to the physical size of the animal), and thus there would be more space for new rings to grow, and also the abalone would be expected to be older (and it is known that there is a positive correlation between the age of an abalone an the number of rings). From the magnitude of the coefficients β_1 and β_2, it can be concluded that the effect that the diameter of the shell (β_1) has on the expected number of rings is lower than the effect of the weight (β_2), since their marginal effects are given by e^{β_i}, $i = 1, 2$.

Finally, the distribution for the parameter σ^2 is found to be very focused on the mean value (that would be of around $(0.1066)^2 = 0.0114$), which is very close to zero. This is good news, since it means that the variance of the model has decreased significantly and the model has arrived at a place where the uncertainty can be modelled in a much more deterministic way (both because the variance of the process is pretty much constant and because its value is very close to zero). It can be the concluded that the model applied was able to minimize the induced noise.

As for the autocorrelations of the model, it can be seen that there are is a lot of correlation present for all β coefficients. This happens because the covariates *shell diameter* and *shell weight* are very highly correlated, as both of them refer to the same physical information: the size of the shell. In the case for σ, there isn't a lot of correlation with previous iterations, because it is a parameter associated to the uncertainty ϵ, which is supposed to be an independent individual uncertainty that has no relationship with anything else in the model.

After this, an standard GLM model for the parameter λ was performed in order to compare results. The equation of the model is the same, only that the parameter ϵ is excluded. The results are shown below:

```r
# GLM Regression Comparison
classic <- glm(y ~ x1 + x2, family = poisson)
```

```r
summary(classic)
```

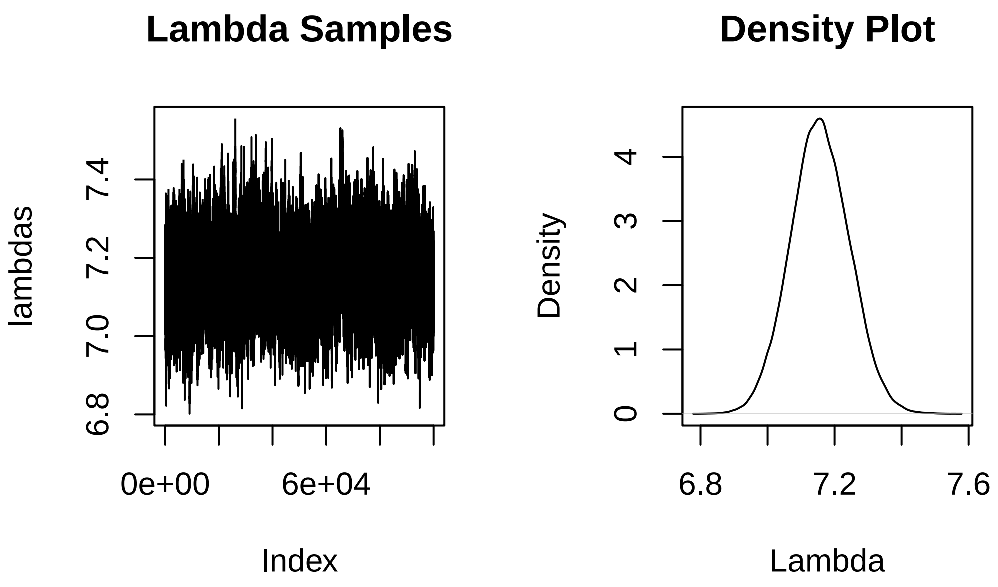
```
Call:
glm(formula = y ~ x1 + x2, family = poisson)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.81854    0.02690  67.609  < 2e-16 ***
x1           0.36233    0.06842   5.296 1.19e-07 ***
x2           1.02540    0.07530  13.617  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4139.3  on 4176  degrees of freedom
Residual deviance: 2459.0  on 4174  degrees of freedom
AIC: 19600

Number of Fisher Scoring iterations: 4
```

When compared with the coefficients obtained for the bayesian model, they are very similar, only diverging at most by 0.025 (in the case for β_1). In this case, there is no estimation of sigma, since the uncertainty ϵ wasn't considered in the frequentist case. It is notable to point out that the standard deviations for the coefficients in the bayesian case are are systematically bigger than the ones for the frequentist case. This may be because of the approach taken to arrive at the result: whereas the frequentist GLM arrives through deterministic equations, the bayesian method assumes a prior distribution for the parameters, which induces extra variability to the results.

Model predictions for the mean

Once the posterior distribution for the coefficients of the model was obtained, they were employed to obtain predictions for the expected number of rings found in a shell of an abalone, that is, predictions for λ, since the variable *rings* follows a Poisson distribution of parameter λ. For that, firstly a new observation was provided, samples of λ were calculated from the obtained coefficients and the given values for the new observation, and finally, predictions for the number of rings of the new individual are obtained through random generations from a poisson distribution for the correspondent values for λ.

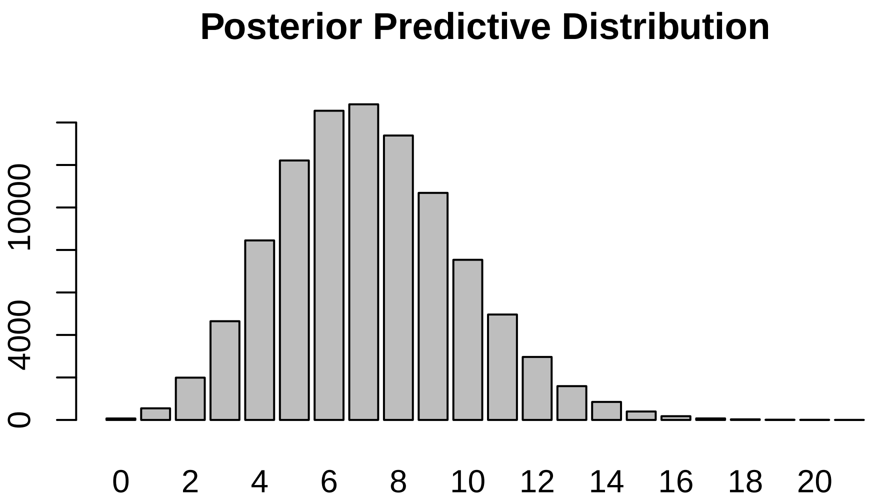
```r
# Prediction for a new observation
shell <- 0.05
diam <- 0.3

samp <- as.matrix(samples[[1]])

# Compute lambda for predictions
lambdas <- exp(samp[,1] + diam * samp[,2] + shell * samp[,3])


# Plot lambda samples
par(mfrow = c(1, 2))
plot(lambdas, type = 'l', main = "Lambda Samples")
density_estimate <- density(lambdas)
plot(density_estimate, main = "Density Plot", xlab = "Lambda", ylab = "Density")
```



```r
par(mfrow = c(1, 1))

summary(lambdas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  6.802   7.098   7.156   7.157   7.216   7.556 
```

```r
# Generate posterior predictive distribution
ypred <- rpois(n = length(lambdas), lambda = lambdas)

# Plot predictions
barplot(table(ypred), main = "Posterior Predictive Distribution")
```



Firstly, the posterior density for λ as well as the samples obtained for it are plotted, from which it can essentially be seen that, after choosing the proper burn-in and number of samples to obtain, that the samples obtained are stationary, and so the model has successfully converged to the desired values for λ.

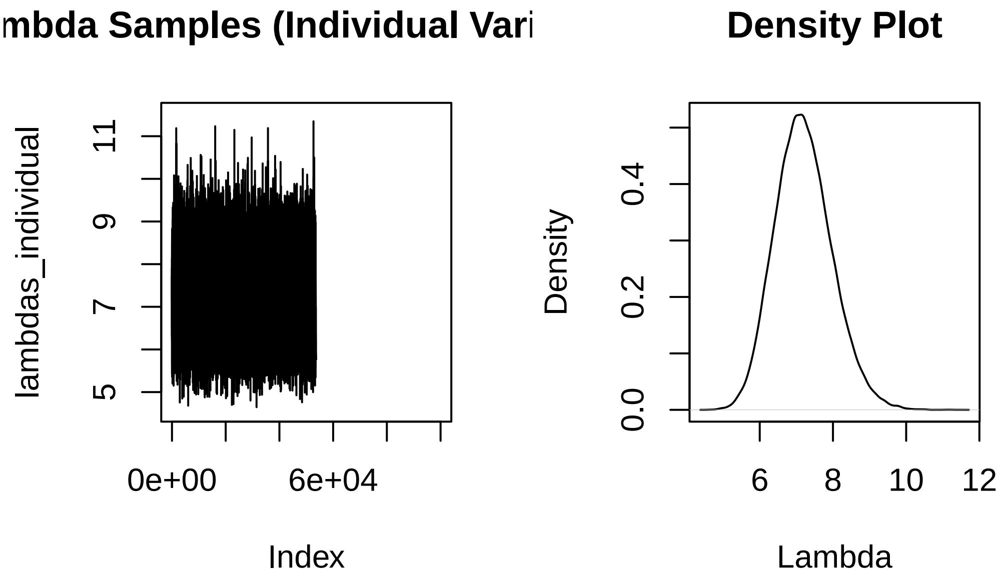
The mean value obtained for the predicted mean is $\lambda = 7.158$.

From this values obtained for λ, a posterior predictive distribution for the number of rings is generated, as can be seen in the last histogram.

Predictions for a new individual

Once the prediction for the mean was obtained, the same procedure was repeated for predictions of the precise number of rings of a new individual (not for its *expected* number of rings). As the quantity of interest is the precise value for the new individual, in this case the uncertainty for the prediction ϵ_{new} was included in the generation of the sample. The same new individual was generated to be able to compare the results.

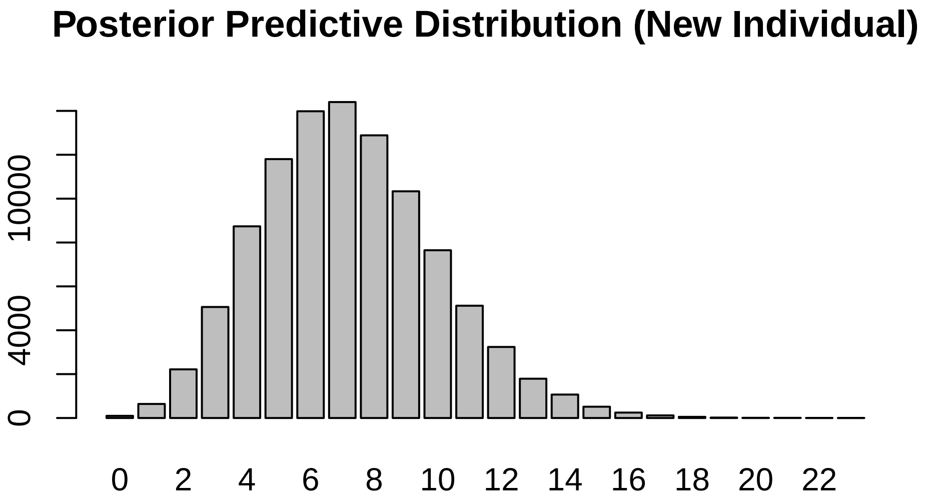
```
# New observation
shell <- 0.05
diam <- 0.3

samp <- as.matrix(samples[[1]])  # Extract MCMC samples
sigma_samp <- samp[, 4]  # Extract sigma samples

# Compute lambda with individual variability
 # Sample individual errors
epsilon_new <- rnorm(n = nrow(samp), mean = 0, sd = sigma_samp)
lambdas_individual <-exp(samp[, 1]+diam*samp[, 2]+shell*samp[, 3]+epsilon_new)

# Plot lambda samples with individual variability
par(mfrow = c(1, 2))
plot(lambdas_individual, type = 'l',
     main = "Lambda Samples (Individual Variability)")
density_estimate <- density(lambdas_individual)
plot(density_estimate, main = "Density Plot", xlab = "Lambda", ylab = "Density")
```



```r
par(mfrow = c(1, 1))

summary(lambdas_individual)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  4.584   6.662   7.163   7.201   7.696  11.505 
```

```r
# Generate posterior predictive distribution (accounting for individual variability)
ypred_individual <- rpois(n = length(lambdas_individual),
                          lambda = lambdas_individual)

# Plot predictions
barplot(table(ypred_individual),
        main = "Posterior Predictive Distribution (New Individual)")
```



The results are comparable to the previous case, where the model is at a point where it has converged and the samples for λ are stationary. A difference is that the distribution for the obtained λ is more right-skewed, however the mean value for λ is 7.158, as was expected in the predictions for a new individual: the point prediction should be the same (the predicted value for the mean), but the variability would change. The histogram for the prediction of the new individual is also provided.

Comparison of posterior distributions for the obtained parameter λ

After obtaining both posterior distributions for the correspondent λ, they were compared between them, with their histograms and their confidence intervals.

```r
#Get confidence intervals
summary(lambdas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  6.802   7.098   7.156   7.157   7.216   7.556 
```

```
quantile(lambdas, probs = c(0.025, 0.975))
```

```
    2.5%    97.5%
6.986889 7.331220
```

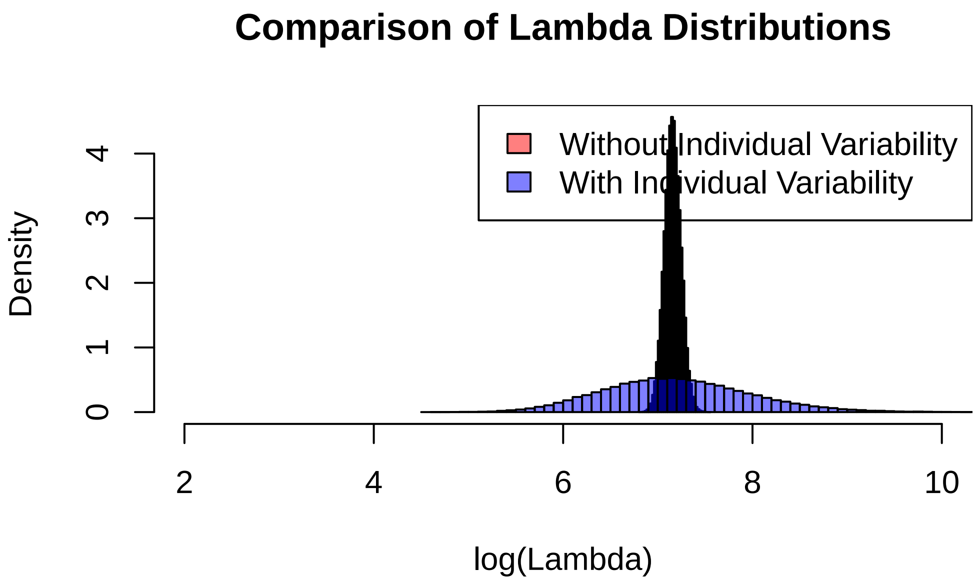
```
summary(lambdas_individual)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.584   6.662   7.163   7.201   7.696  11.505
```

```
quantile(lambdas_individual, probs = c(0.025, 0.975))
```

```
    2.5%    97.5%
5.798865 8.831579
```

```
# Plot both lambda distributions together
hist((lambdas), probability = TRUE, col = rgb(1, 0, 0, 0.5), xlim = c(2,10),
     main = "Comparison of Lambda Distributions",
     xlab = "log(Lambda)", ylab = "Density", breaks = 50)
hist((lambdas_individual), probability = TRUE, col = rgb(0, 0, 1, 0.5),
     add = TRUE, breaks = 50)
legend("topright", legend = c("Without Individual Variability",
                              "With Individual Variability"),
       fill = c(rgb(1, 0, 0, 0.5), rgb(0, 0, 1, 0.5)))
```



As for the confidence intervals for both obtained λ, the 95 % confidence interval for the λ fo the predicted mean value is $(6.987, 7.328)$ and for the λ for the predicted individual value is $(5.808, 8.825)$ (both intervals were obtained as the ones encapsulated by the 0.025 and the 0.975 quantiles of the respective samples).

This big difference in spread can be seen in the histogram, were the correspondent for the case without the individual variability is much more peaked and with way less dispersion, while the histogram for the case with individual variability is much more spread, exemplifying how estimating a precise value for an individual involves suffering greater uncertainty.

Comparison for obtained predictions λ distributions

Finally, the same was compared for the prediction of the expected number of rings and the number of rings predicted for an individual.

```
#Get confidence intervals
summary(ypred)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   5.000   7.000   7.169   9.000  22.000 
```

```
quantile(ypred, probs = c(0.025, 0.975))
```

```
 2.5% 97.5% 
    2    13 
```

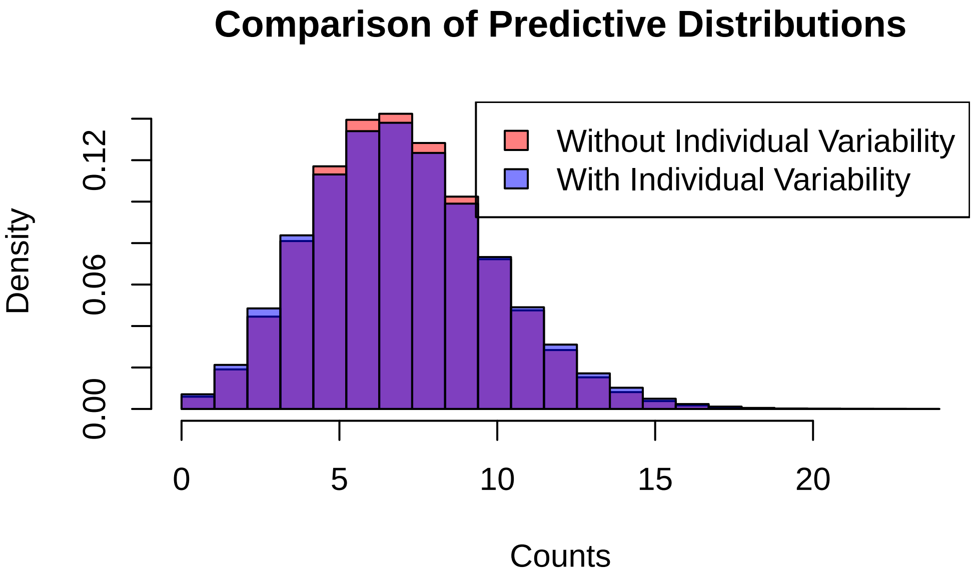
```
summary(ypred_individual)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   5.000   7.000   7.203   9.000  24.000 
```

```
quantile(ypred_individual, probs = c(0.025, 0.975))
```

```
 2.5% 97.5% 
    2    13 
```

```
# Plot both predictive distributions together
# Compute common break points based on the combined range of both datasets
common_breaks <- seq(min(c(ypred, ypred_individual)),
                     max(c(ypred, ypred_individual)), length.out = 24)

# Plot the first histogram
hist(ypred, probability = TRUE, col = rgb(1, 0, 0, 0.5),
     main = "Comparison of Predictive Distributions",
     xlab = "Counts", ylab = "Density", breaks = common_breaks)

# Overlay the second histogram with the same breaks
hist(ypred_individual, probability = TRUE, col = rgb(0, 0, 1, 0.5),
     add = TRUE, breaks = common_breaks)

# Add legend
legend("topright", legend = c("Without Individual Variability",
                              "With Individual Variability"),
       fill = c(rgb(1, 0, 0, 0.5), rgb(0, 0, 1, 0.5)))
```



The 95 % intervals in this case are found to be the same, both $(2, 13)$, as the possible values are restricted when dealing with integer variables. It can be seen that, even though the obtained λ had a very different distribution, the obtained predictions for the number of rings are more similar, at least in the centered range of most values. However, the frequency for these values in the interval may change.

When studying the histograms, they are fairly similar and are centered around the same value, since the mean value is the same for both. However, there is a crucial difference. The prediction with individual variability is smaller at the peak, and bigger at the tails. This is because, as had been commented before, the individual prediction provides an extra uncertainty that isn't present in the case for the prediction of the mean.

Conclusion

In this project, it was attempted to obtain a bayesian model based on Gibbs Sampling in order to predict the number of rings found in abalone shells from data from the shell diameter and the shell weight, assuming the number of rings follows a Poisson distribution of parameter λ and then performing a logistic regression on y (standard regression of the logarithm of λ), also considering an individual uncertainty ϵ. Gibbs Sampling was employed to obtain the posterior distribution for the coefficients of the model, as well as for the parameter σ that controls the uncertainty.

When comparing the results, the coefficients were found to be very similar to those of the frequentist GLM approach, even though the coefficients in the bayesian case are found to have a systematically higher standard deviation, probably due to the fact that prior distributions were assumed for the parameters of the model. It was also found that the diameter of the shell had a smaller effect on the number of rings in the shell than the weight of it, and that both covariates were positively related to the number of rings (when one is held constant and the other one increases, the number of rings increases).

As for the predictions for the mean number of rings of a new individual and for the exact number of rings for a new individual, the λ's in both cases were found to have a posterior distribution with the same mean (as was expected), but the variation in the case for the predictions for the exact value of a new individual was way larger, as was expected, since the extra uncertainty of that individual needs to be taken into account. This variation in the distribution of the parameter λ in both cases was nevertheless smoothed in the distribution for the predicted number of rings, where the histograms provided much more similar results, even though the exact predicted value for an individual still showed a greater dispersion.