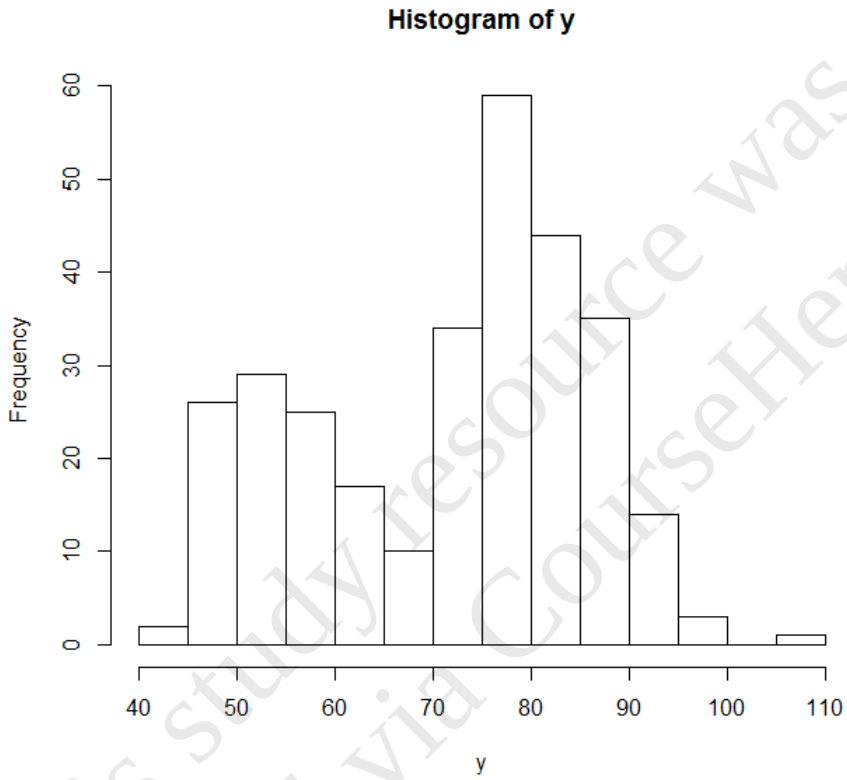
MAST30027: Modern Applied Statistics

Assignment 6

Due: 1:00 pm Fri 23 October (week 12)

This assignment is worth 3 1/3% of your total mark.

Here is a histogram of 299 observations of the time between eruptions for the Old Faithful geyser in Yellowstone National Park. The data can be found in the file `geyserdata.txt` on the subject website.



We will model this data using a mixture of two normals, which has density

$$f(x) = \pi\sigma_1^{-1}\phi((x-\mu_1)/\sigma_1) + (1-\pi)\sigma_2^{-1}\phi((x-\mu_2)/\sigma_2)$$

where ϕ is the standard normal density, $\pi \in [0,1]$, $\mu_i \in \mathbb{R}$ and $\sigma_i \in (0,\infty)$.

(a) Show that if $A \sim \text{bin}(1,\pi)$, $X_1 \sim N(\mu_1,\sigma_1^2)$ and $X_2 \sim N(\mu_2,\sigma_2^2)$, all independent of each other, then $Y = AX_1 + (1-A)X_2$ has density f.

Solution: Conditioning on A we have

$$\begin{aligned}
\mathbb{P}(Y \le y) &= \mathbb{P}(Y \le y|A=0)(1-\pi) + \mathbb{P}(Y \le y|A=1)\pi \\
&= \mathbb{P}(X_2 \le y)(1-\pi) + \mathbb{P}(X_1 \le y)\pi \\
&= \mathbb{P}((X_2-\mu_2)/\sigma_2 \le (y-\mu_2)/\sigma_2)(1-\pi) + \mathbb{P}((X_1-\mu_1)/\sigma_1 \le (y-\mu_1)/\sigma_1)\pi \\
&= \Phi((y-\mu_2)/\sigma_2)(1-\pi) + \Phi((y-\mu_1)/\sigma_1)\pi
\end{aligned}$$

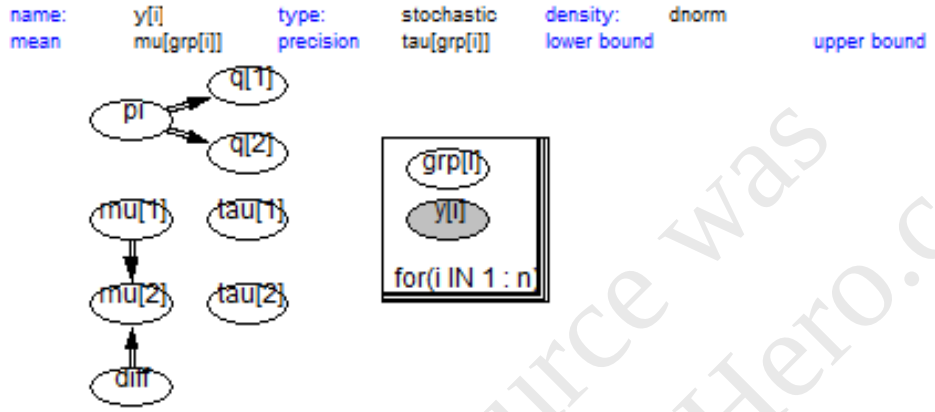
where Φ is the c.d.f. of a standard normal. Differentiating w.r.t. y gives the result.

(b) To build a Bayesian model for $Y(i)$, the i-th observation, we introduce variables $G(i)$ such that if $G(i) = j$ then $Y(i)$ has mean μ_j and precision $\tau_j = 1/\sigma_j^2$, for $j = 1, 2$. Moreover $\mathbb{P}(G(i) = 1) = \pi$ and $\mathbb{P}(G(i) = 2) = 1 - \pi$. Thus, to specify the model we need priors for π, μ_j and τ_j.

To avoid ambiguity, we suppose that $\mu_1 < \mu_2$. To achieve this we use a $N(60, 1000)$ prior for μ_1 and a $\Gamma(1, 1)$ prior for $\delta := \mu_2 - \mu_1$. (Note, μ_1 acts as a location parameter and δ as a scale parameter.)

For the τ_j we use $\Gamma(0.01, 0.01)$ priors, and for π a $\beta(3, 3)$ prior.

A WinBUGS doodle for this model is given below. Based on this, or otherwise, write a BUGS model for this problem. (Note that `q[]` is intended to be a probability vector that can be used to define a distribution using `dcat`.)



Solution: For the model we have

```
model;
{
   mu[1] ~ dnorm(60,0.001)
   tau[1] ~ dgamma(0.01,0.01)
   mu[2] <- mu[1] + diff
   tau[2] ~ dgamma(0.01,0.01)
   diff ~ dgamma(1,1)
   pi ~ dbeta(3,3)
   q[1] <- pi
   q[2] <- 1 - pi
   for( i in 1 : n ) {
      grp[i] ~ dcat(q[])
   }
   for( i in 1 : n ) {
      y[i] ~ dnorm(mu[grp[i]],tau[grp[i]])
   }
}
```

The data given on the subject webpage is already in the correct format. To check convergence of the MCMC algorithm using the BGR diagnostic we need at least two initial values, ideally some distance apart, for example:

```
list(mu = c(30, NA), tau = c(10, 1), diff = 70, pi = 0.2,
grp = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

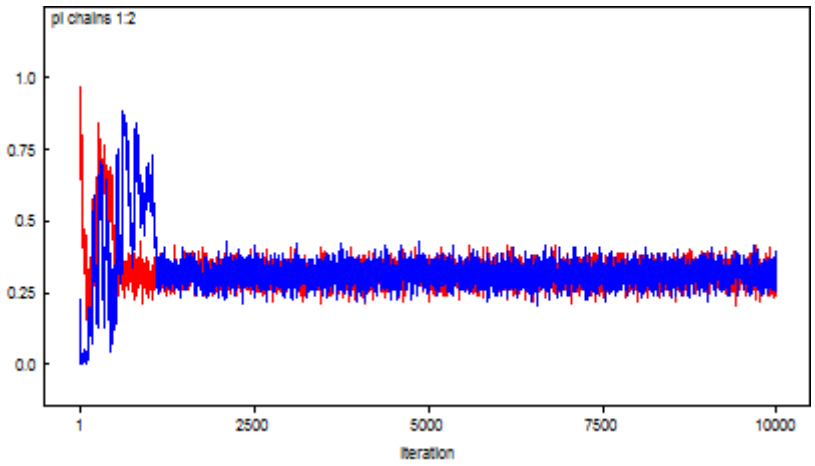
```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1))

list(mu = c(90, NA), tau = c(1, 10), diff = 10, pi = 0.8,
grp = c(2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2))
```

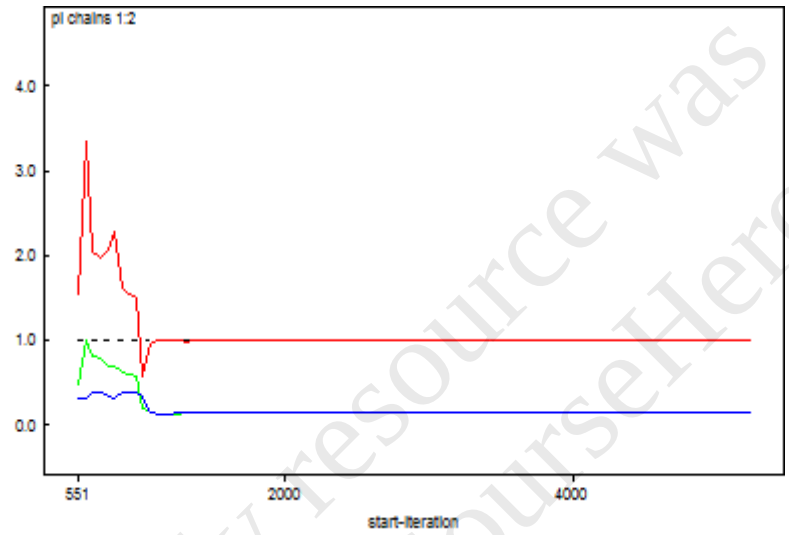
(c) Fit the model using WinBUGS. Use the BGR diagnostic for π to choose a burn-in period, and make sure that your sample size is large compared to the range of correlation in the trace.

Report the size of burn-in and sample size used, then give an estimate of $\mathbb{P}(Y > 100)$. What is the MC error for your estimate?

Solution: Two MCMC chains were run for 10,000 iterations. The traces and the BGR plot are given below. Based on these it looks like the chains have converged after 1,250 iterations. To be on the safe side we will use a burn-in of 2000 iterations.

Once they have converged the chains appear to be mixing well (the traces look like hairy caterpillars), so a sample of size 8000 should be large enough to overcome the autocorrelation present in the samples and give good estimates.





To estimate $\mathbb{P}(Y > 100)$ we need to add a few nodes to the model.

```
gnew ~ dcat(q[])
ynew ~ dnorm(mu[gnew],tau[gnew])
prob <- step(ynew - 100)
```

`gnew` and `ynew` give us a fitted/predicted value (they are the same thing in this case, as we have no predictor variables), then `prob` indicates when $Y > 100$. The posterior mean of `prob` will give us (an MCMC estimate of) the posterior probability that $Y > 100$.

We used a burn-in of 2000 and a sample size of 8000, as suggested by our convergence analysis of `pi`. To be thorough we could repeat the convergence analysis using `ynew`, but we do not do that here. Our estimate was 0.0025 with a MC error of 0.00039 (roughly 1 std dev). This was using a sample of size 16,000 (8000 from each chain). The MC error is relatively high (roughly 15% of the size of the estimate, so the true value could easily be anywhere between 0.0017 and 0.0033), but this is to be expected when trying to estimate such a small probability.