ADVANCED BAYESIAN MODELING

EXAMPLE OF ROBUST ANALYSIS WITH THE t DISTRIBUTION: **JAGS ANALYSIS WITH ν AS A PARAMETER**

Our previous Bayesian t-based model for the `newcomb` data set used $\nu = 2$.

What if we allow the data to choose ν adaptively, as a parameter?

As before, consider values of $1/\nu$ ranging from 0 (normal) to 1 (location-scale Cauchy).

Robust Model with Varying ν

$$
\begin{aligned}
y_1, \ldots, y_{66} \mid \mu, \sigma^2, \nu &\sim \text{iid } t_\nu(\mu, \sigma^2) \\
p(\mu) &\propto 1 && -\infty < \mu < \infty \\
p(\sigma^2) &\propto (\sigma^2)^{-1} && \sigma^2 > 0 \\
1/\nu &\sim \mathrm{U}(0, 1)
\end{aligned}
$$

Since JAGS needs proper priors, try

$$\begin{aligned}
y_1, \ldots, y_{66} \mid \mu, \sigma^2, \nu &\sim \text{iid } t_\nu(\mu, \sigma^2) \\
\mu &\sim \mathrm{N}(0, 10000^2) \\
\sigma^2 &\sim \mathrm{Gamma}(0.00001, 0.00001) \\
1/\nu &\sim \mathrm{U}(0, 1)
\end{aligned}$$

(This choice of priors still provides partial conjugacy for μ and σ^2, if the auxiliary variables trick is used.)

JAGS Analysis

In file `newcomb2.bug`:

```
model {

  for(i in 1:length(y)) {
    y[i] ~ dt(mu, 1/sigmasq, 1/nuinv)
    yrep[i] ~ dt(mu, 1/sigmasq, 1/nuinv)
  }

  mu ~ dnorm(0, 0.00000001)
  sigmasq ~ dgamma(0.00001, 0.00001)
  nuinv ~ dunif(0, 1)

}
```

```
> d2 <- list(y = newcomb)

> inits2 <- list(list(mu=1000, sigmasq=1000000, nuinv=0.01),
+                list(mu=1000, sigmasq=0.01, nuinv=0.99),
+                list(mu=-1000, sigmasq=1000000, nuinv=0.99),
+                list(mu=-1000, sigmasq=0.01, nuinv=0.01))
```

```
> library(rjags)
...

> m2 <- jags.model("newcomb2.bug", d2, inits2, n.chains=4, n.adapt=1000)
...

> update(m2, 1000)  # burn-in
  |**************************************************| 100%

> x2 <- coda.samples(m2, c("mu","sigmasq","nuinv"), n.iter=2000)
  |**************************************************| 100%
```

Convergence diagnostics are reasonable (not shown).

```
> x2 <- coda.samples(m2, c("mu","sigmasq","nuinv","yrep"), n.iter=2000)
  |**************************************************| 100%

> effectiveSize(x2[, c("mu","sigmasq","nuinv")])
      mu  sigmasq    nuinv
7039.802 2583.859 2590.482
```

```
> summary(x2[, c("mu","sigmasq","nuinv")])

...

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

           Mean     SD Naive SE Time-series SE
mu      27.3969 0.6103 0.006823       0.007347
sigmasq 14.9530 5.0490 0.056450       0.099878
nuinv    0.5245 0.1521 0.001701       0.002997

2. Quantiles for each variable:

           2.5%     25%    50%     75%   97.5%
mu      26.2188 26.9767 27.390 27.8033 28.5990
sigmasq  7.2752 11.3681 14.219 17.8233 26.7630
nuinv    0.2673  0.4114  0.511  0.6237  0.8596
```

This new analysis has approximate posterior mean and 95% central posterior interval for μ practically identical to the previous analysis (with $\nu = 2$).

Marginal posterior density of μ is also very similar (not shown).

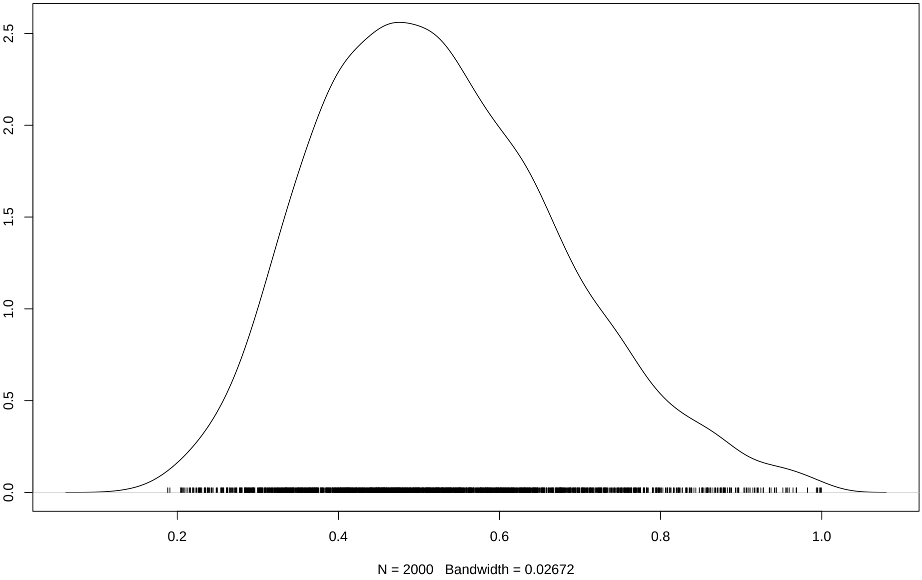
For $1/\nu$, approximate posterior mean and 95% central posterior interval are

$$0.52 \qquad\qquad\qquad\qquad (0.26,\ 0.86)$$

suggesting $\nu = 2$ is a reasonable value (but $\nu = \infty$ or $\nu = 1$ might not be).

To see the posterior density of $1/\nu$:

```
densplot(x2[,"nuinv"])
```


2.5
2.0
1.5
1.0
0.5
0.0
0.2
0.4
0.6
0.8
1.0
N = 2000 Bandwidth = 0.02672

Posterior predictive p-values based on max and min statistics:

```
> yrep <- as.matrix(x2)[, paste("yrep[",1:length(newcomb),"]", sep="")]

> mean(apply(yrep, 1, min) <= min(newcomb))
[1] 0.12975

> mean(apply(yrep, 1, max) >= max(newcomb))
[1] 0.912875
```

Slightly less extreme than before.