ADVANCED BAYESIAN MODELING

EXAMPLE OF ROBUST ANALYSIS WITH THE t DISTRIBUTION:

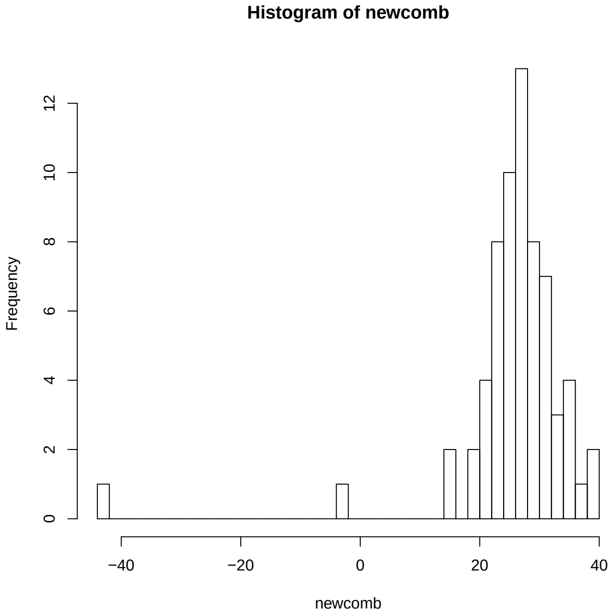
JAGS ANALYSIS WITH FIXED ν

Recall the `newcomb` data:

66 measurements that are shifted and scaled times for light to travel the same known distance (in air)

```
> library(MASS)  # has newcomb data set

> hist(newcomb, breaks=50)
```


Histogram of newcomb
Frequency
0
2
4
6
8
10
12
−40
−20
0
20
40
newcomb

We seek inference about the mean of the measurements.

For the whole data:

```
> mean(newcomb)
[1] 26.21212

> median(newcomb)
[1] 27
```

For the data without the outliers (which are obs. 2 and 54):

```
> mean(newcomb[-c(2,54)])
[1] 27.75
```

Perhaps a robust model can include the outliers without allowing them to unduly influence the mean estimate.

Robust Model

We try a t_2 model:

$$
\begin{aligned}
y_1, \ldots, y_{66} \mid \mu, \sigma^2 &\sim \text{iid } t_2(\mu, \sigma^2) \\
p(\mu) &\propto 1 \qquad\qquad -\infty < \mu < \infty \\
p(\sigma^2) &\propto (\sigma^2)^{-1} \qquad\qquad \sigma^2 > 0
\end{aligned}
$$

Why $\nu = 2$? Arbitrary, but small enough to make outliers highly probable, yet large enough that the mean exists.

JAGS needs proper priors, so try

$$
\begin{aligned}
y_1, \ldots, y_{66} \mid \mu, \sigma^2 &\sim \text{iid } t_2(\mu, \sigma^2) \\
\mu &\sim \mathrm{N}(0, 10000^2) \\
\sigma^2 &\sim \mathrm{Gamma}(0.00001, 0.00001)
\end{aligned}
$$

(This choice of priors would provide partial conjugacy, if the auxiliary variables trick is used.)

JAGS Analysis

In file `newcomb1.bug`:

```
model {

  for(i in 1:length(y)) {
    y[i] ~ dt(mu, 1/sigmasq, 2)
    yrep[i] ~ dt(mu, 1/sigmasq, 2)
  }

  mu ~ dnorm(0, 0.00000001)
  sigmasq ~ dgamma(0.00001, 0.00001)

}
```

JAGS uses `dt` to specify the t distribution, with parameters μ, $1/\sigma^2$, and ν, in that order.

(No need to define a `sigmasqinv` node.)

Set up data and initializations for four chains:

```
> d1 <- list(y = newcomb)

> inits1 <- list(list(mu=1000, sigmasq=1000000),
+                list(mu=1000, sigmasq=0.01),
+                list(mu=-1000, sigmasq=1000000),
+                list(mu=-1000, sigmasq=0.01))
```

```
> library(rjags)
...

> m1 <- jags.model("newcomb1.bug", d1, inits1, n.chains=4, n.adapt=1000)
...

> update(m1, 1000)  # burn-in
  |**************************************************| 100%

> x1 <- coda.samples(m1, c("mu","sigmasq"), n.iter=2000)
  |**************************************************| 100%
```

Convergence diagnostics are adequate (not shown).

```
> x1 <- coda.samples(m1, c("mu","sigmasq","yrep"), n.iter=2000)
  |**************************************************| 100%

> effectiveSize(x1[, c("mu","sigmasq")])
      mu  sigmasq
7124.320 5016.147
```

```
> summary(x1[, c("mu","sigmasq")])

...

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

         Mean     SD Naive SE Time-series SE
mu      27.39 0.6136 0.006861       0.007352
sigmasq 14.84 4.0664 0.045464       0.059188

2. Quantiles for each variable:

          2.5%   25%   50%   75% 97.5%
mu      26.202 26.99 27.39 27.81 28.60
sigmasq  8.518 11.92 14.22 17.14 24.17
```

Compare the approximate posterior mean and 95% central posterior interval for μ:

$$27.4 \qquad\qquad (26.2,\ 28.6)$$

to the ordinary sample mean and t-interval with the outliers:

$$\bar{y} \;\approx\; 26.2 \qquad\qquad \bar{y} \;\pm\; t_{65,0.025} \cdot s/\sqrt{n} \;\approx\; (23.6,\ 28.9)$$

and without the outliers:

$$27.75 \qquad\qquad (26.5,\ 29.0)$$

Outliers have a relatively small influence on the robust Bayesian analysis.

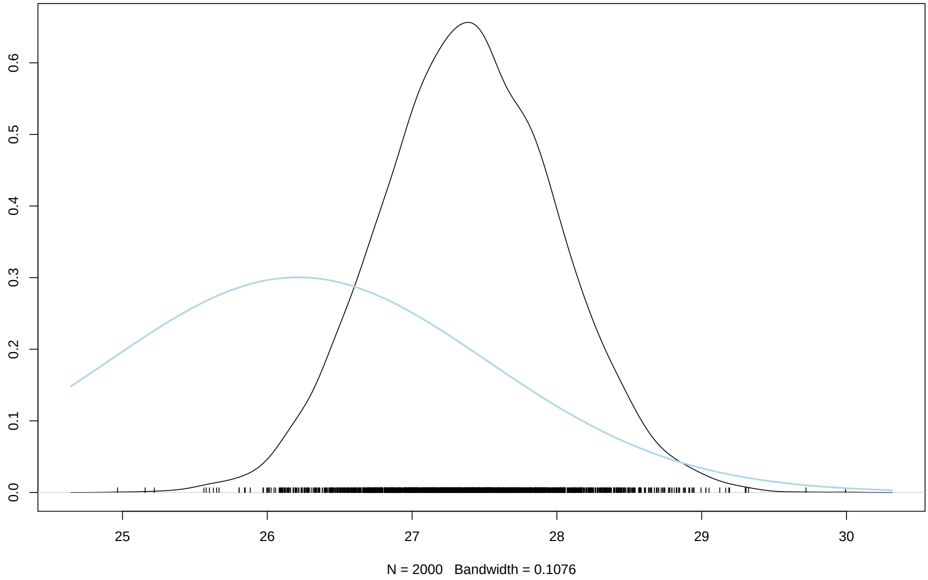
Let's plot the marginal posterior density of μ:

```
> densplot(x1[,"mu"])
```

and, for comparison, overplot the (exact) marginal posterior density of μ under the normal sample model:

```
> dtscaled <- function(x, mu, sigma, nu) dt((x-mu)/sigma, nu)/sigma

> curve(dtscaled(x, mean(newcomb), sd(newcomb)/sqrt(length(newcomb)),
+                length(newcomb)-1), add=TRUE, col="lightblue", lwd=2)
```


0.0
0.1
0.2
0.3
0.4
0.5
0.6
25
26
27
28
29
30
N = 2000 Bandwidth = 0.1076

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

```
> yrep <- as.matrix(x1)[, paste("yrep[",1:length(newcomb),"]", sep="")]

> mean(apply(yrep, 1, min) <= min(newcomb))
[1] 0.0935

> mean(apply(yrep, 1, max) >= max(newcomb))
[1] 0.919375
```

Noteworthy, but not quite indicative of problems.

(Compare results for the normal sample model in BDA3, Sec. 6.3.)