STAT 431 — Applied Bayesian Analysis — Course Notes

Introduction to JAGS and Model Graphs

Fall 2022

JAGS Software

Software can automate Gibbs sampling and extend it to handle non-conjugacy.

We will use **JAGS** — Just Another Gibbs Sampler:

`https://mcmc-jags.sourceforge.io/`

This web site links to a download site which also has an important document: the JAGS user manual.

JAGS has its own interface — a simple command line — but we will instead run it from within R using package `rjags`.

Note: Make sure to first download and install JAGS (as standalone software), then install `rjags` as an R package.

(Other JAGS-related R packages exist — see JAGS user manual.)

Remark: JAGS was based on the earlier BUGS (Bayesian inference Using Gibbs Sampling) project that produced WinBUGS, which later spawned OpenBUGS.

BEWARE: JAGS is not fully compatible with WinBUGS/OpenBUGS. OpenBUGS code may need modifications to be used as JAGS code.

To run JAGS you need:

- the data
- a specification of the model (including priors)
- the initial values (for all chains)

R/JAGS Example 3.6:

Population Proportion

JAGS Model Specification

The model (including prior) is specified in the `model` block:

```
model {
  ... statements ...
}
```

Many statements have the form

```
variablename ~ distribution(args)
```

Example:

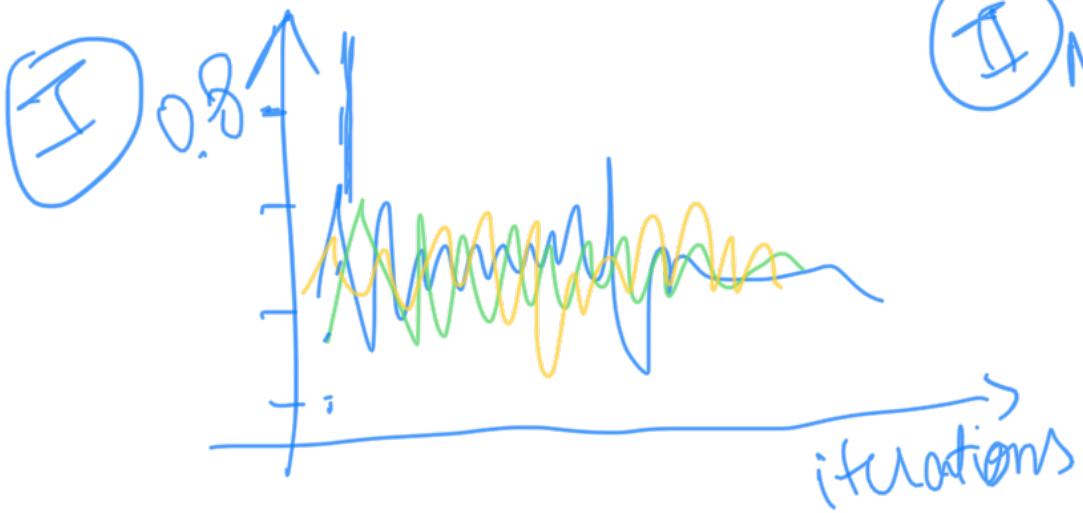
$$
\begin{aligned}
Y \mid \theta &\sim \text{Binomial}(n = 10,\ \theta) \\
\theta &\sim \text{Beta}(\alpha = 2,\ \beta = 1)
\end{aligned}
$$

```
model {
  y ~ dbin(theta, 10)
  theta ~ dbeta(2, 1)
}
```

- the order of the statements *does not* matter
- the order of the arguments *does* matter
- variables do not need to be "declared"

There is a **graph** for this model:

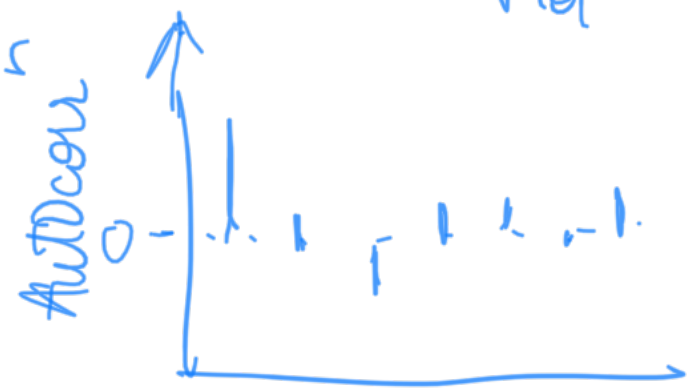
[Draw model graph ...]



→ Assess convergence
3 chains for θ values

II Multi-modality → not here

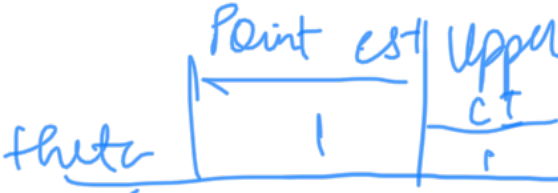
Autocorrelation plot



The graph says that:

- θ and y are both specified as random variables
- θ is specified marginally (unconditionally)
- y is specified conditionally on θ

III Gelman-Rubin stat plot
→ 1 when chain are sampling from posterior



≠ Note : Time-series SE < 1/20 of SD

Variables may be specified together as a **vector** or as an **array**.

Eg: If `y` is a vector, `y[i]` is its `i`th element.

Vectors can be handled easily using a `for` statement.
E.g. if `y` has `n` elements,

```
for (i in 1:n) {

  y[i] ~ distribution(args)

}
```

Important: `n` must be a constant (non-random).

⇒ Gibbs sampling requires atleast 2 params
⇒ JAGS → Gibbs next best thing

Example:

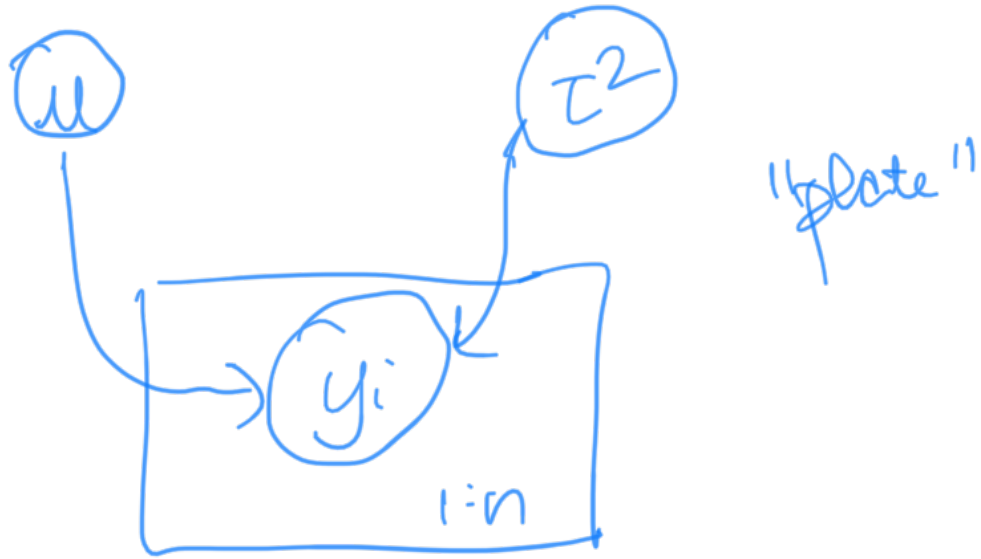
$$Y_1, \ldots, Y_n \mid \mu, \tau^2 \sim iid\ \text{Normal}(\mu,\ 1/\tau^2)$$

$$\left.\begin{aligned} \mu &\sim \text{Normal}(0, 100) \\ \tau^2 &\sim \text{Gamma}(0.01, 0.01) \end{aligned}\right\} \text{independent}$$

```
model {
  for (i in 1:n) {
    y[i] ~ dnorm(mu, tausq)
  }
  mu ~ dnorm(0, 0.01)
  tausq ~ dgamma(0.01, 0.01)
}
```

The model graph:

[Draw model graph ...]



A **plate** indicates that the variables on it are repeated as vector/array elements.

The length of a *data* vector `y` can be referenced as `length(y)`:

```
model {
  for (i in 1:length(y)) {
    y[i] ~ dnorm(mu, tausq)
  }
  ...
}
```

This avoids the need to specify the length as a separate data value.

Improper priors, such as

$$\mu \sim 1\, d\mu$$

are NOT allowed in JAGS.

You can instead use a "vague" proper prior, such as

```
mu ~ dnorm(0, 0.000001)
```

though whether this is "vague" enough depends on the context.

What if we want inference about σ^2 (not just τ^2)?

We could use

```
model {
  for (i in 1:length(y)) {
    y[i] ~ dnorm(mu, tausq)
  }
  mu ~ dnorm(0, 0.000001)
  tausq ~ dgamma(0.01, 0.01)

  sigmasq <- 1/tausq
}
```

deterministic rel^n

A statement

```
variablename <- expression
```

represents a **deterministic** relationship: the variable is computed according to the expression.

The expression may use the usual four arithmetic symbols, the unary minus, the power symbol `^`, and certain scalar functions — see JAGS user manual.

Deterministic relationships are *hollow* lines in the model graph:

[Draw model graph ...]

Important: Variables having data values are *not allowed* on the left-hand side of a deterministic relationship (<-).

Eg: Suppose data `y[i]` needs to be log-transformed to normality.

NOT allowed:

```
y[i] <- exp(z[i])
z[i] ~ dnorm(mu, tausq)
```

What about this?

```
z[i] <- log(y[i])
z[i] ~ dnorm(mu, tausq)
```

NOT allowed, because a variable is allowed on the left hand side of at most one statement (in the `model` block).

What about this?

```
log(y[i]) ~ dnorm(mu, tausq)
```

Again, NOT allowed by JAGS.

So what should we do? ...

Use a preceding `data` block to transform data:

```
data {
  for (i in 1:length(y)) {
    z[i] <- log(y[i])
  }
}
model {
  for (i in 1:length(z)) {
    z[i] ~ dnorm(mu, tausq)
  }
  ...
}
```

The `data` block is executed (once) before the model is run.

Example: Wikipedia Article Modifications

For $n = 10$ random English Wikipedia articles, time (days) since last modification is recorded:

$$Y_1, \ldots, Y_{10} \sim iid \quad \text{(given parameters)}$$

We want to know about the *median* time, and perhaps the variation.

Preliminary analysis (Q-Q plot) suggests approximate normality on the *log* scale:

$$\ln(Y_1), \ldots, \ln(Y_{10}) \mid \mu, \sigma^2 \sim iid \ \text{Normal}(\mu, \sigma^2)$$

Consider prior

$$\left.\begin{aligned} \mu &\sim \text{Normal}(0, \sigma_0^2) \\ \sigma^2 &\sim \text{InvGamma}(\alpha, \beta) \end{aligned}\right\} \text{ independent}$$

We will try

$$\sigma_0^2 = 10000 \qquad \alpha = 0.0001 \qquad \beta = 0.0001$$

because they produce a "vague" prior.

We might want inference for μ, σ^2, and

$$e^{\mu} \quad = \quad \text{(conditional) median of } Y$$

(Why is that the median?)

Remember, JAGS uses

$$\tau^2 \quad = \quad 1/\sigma^2$$

Graph:

[Draw model graph ...]

Note: Use a square (instead of a circle) for values that are "constants," i.e. not on the left hand side of any `~` or `<-`.

R/JAGS Example 3.7:

(Transformed) Normal Sample

Convergence Assessment

- TA job: school assistantship
- RCL: STAT or any other STAT course for now
- start

- Initial Values

  ... should be **overdispersed** — some should be chosen far away from the values you expect to see for the parameters

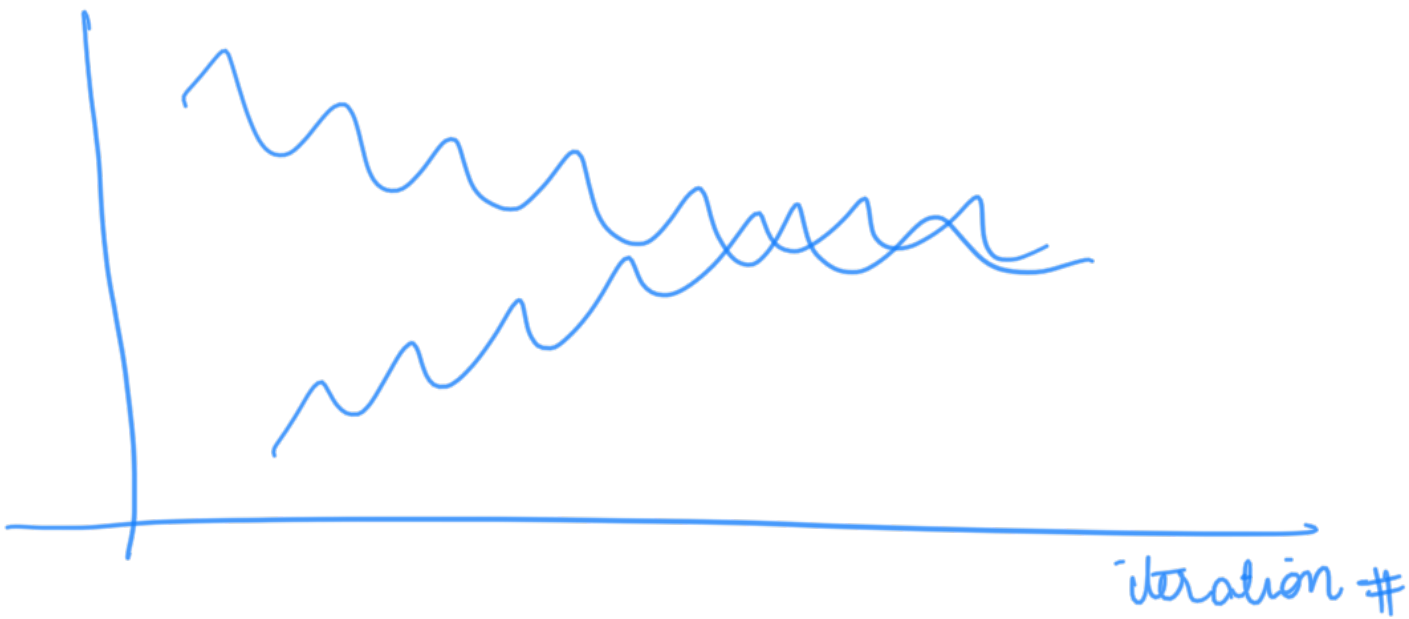
e.g. Multi-modal data
speed of convergence:

- Trace (History) Plots

  check for convergence time and speed of "mixing"

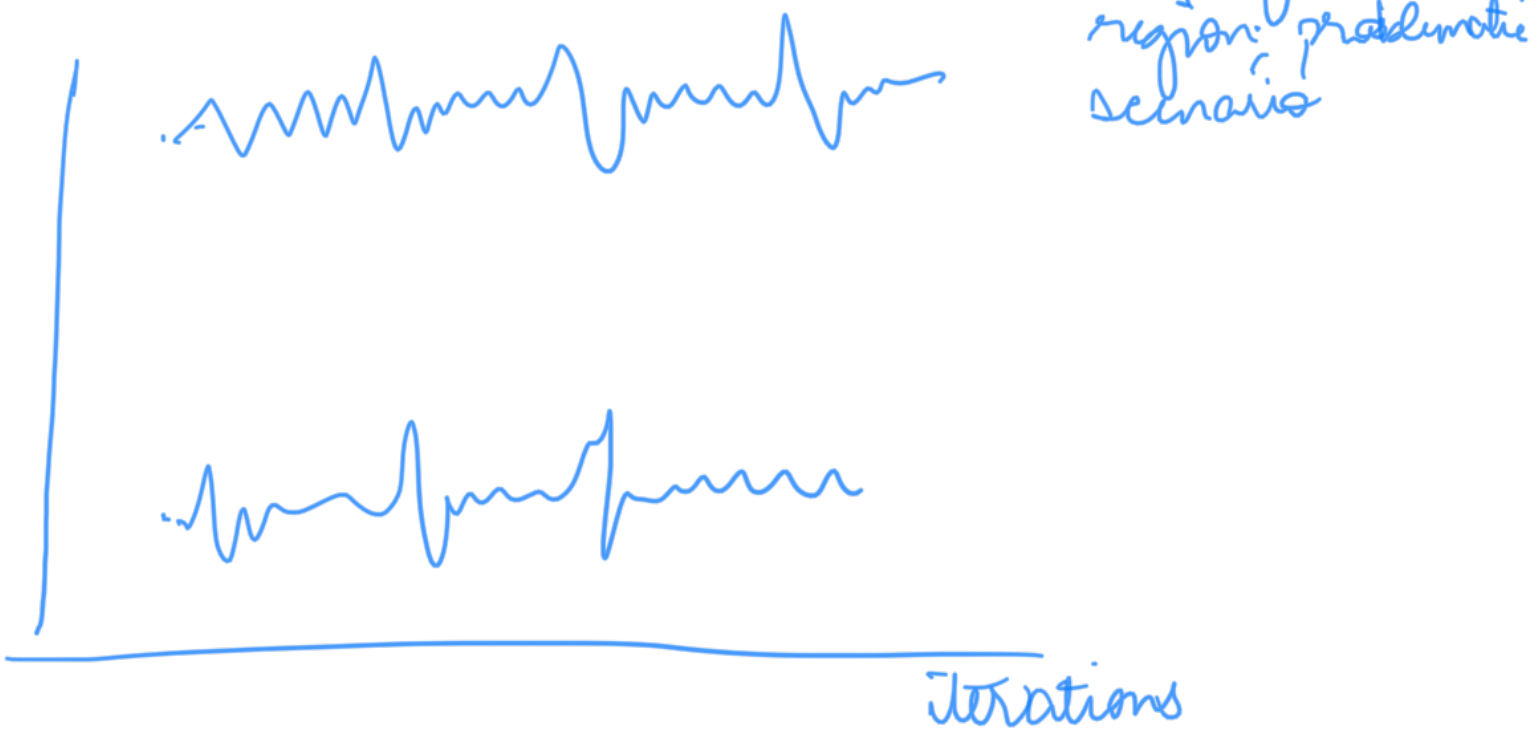
  Eg: showing transient (initialization) effects

  [Draw trace plot ...]



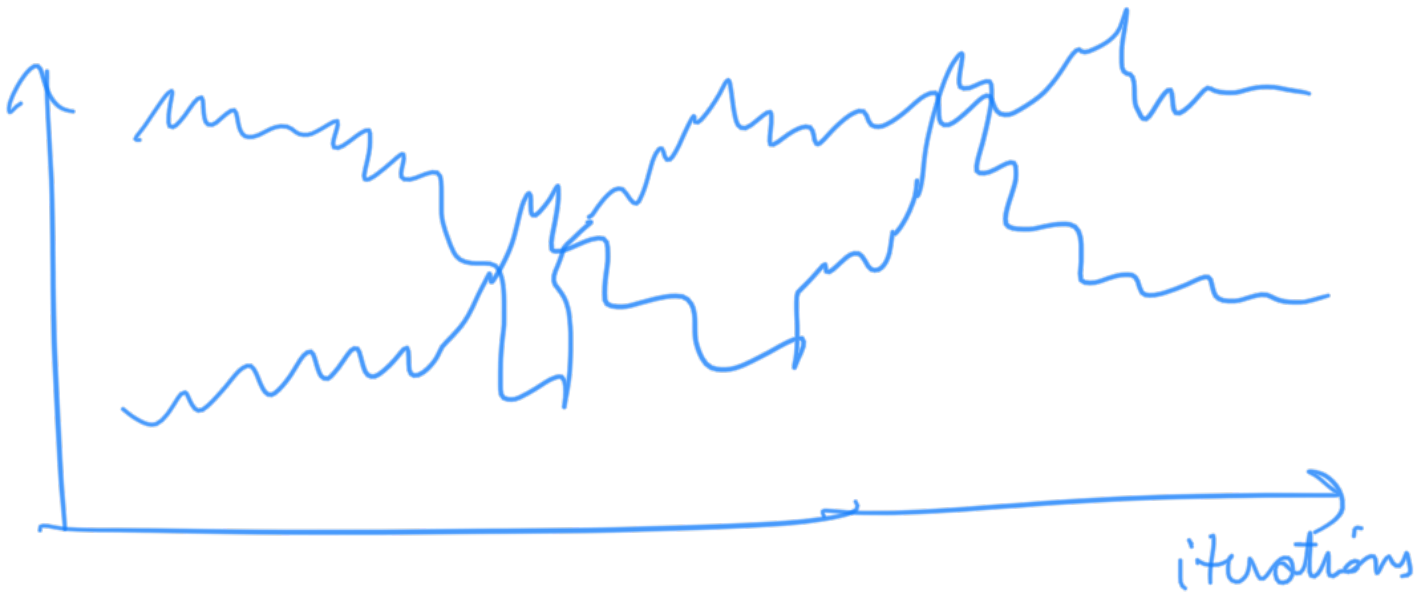
Eg: showing lack of convergence

[Draw trace plot ...]



Eg: showing slow mixing (which gives high MC error)

[Draw trace plot ...]

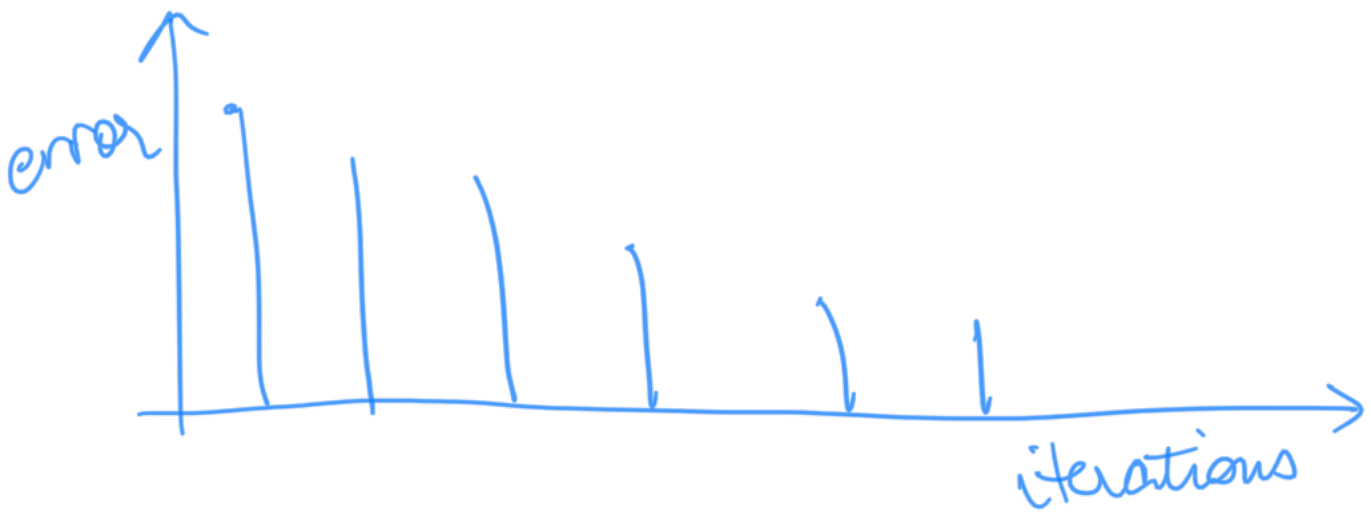


- Autocorrelation Plots

  assess serial dependence

  Eg: showing high correlations (which gives slow mixing)

  [Draw autocorrelation plot ...]



- Gelman-Rubin Statistic and Plots

The **Gelman-Rubin statistic**, or "potential scale reduction factor," helps to monitor convergence when using multiple chains with overdispersed starting points.

For a specific monitored quantity, it is essentially

$$R = \sqrt{\frac{\hat{V}}{W}}$$

where

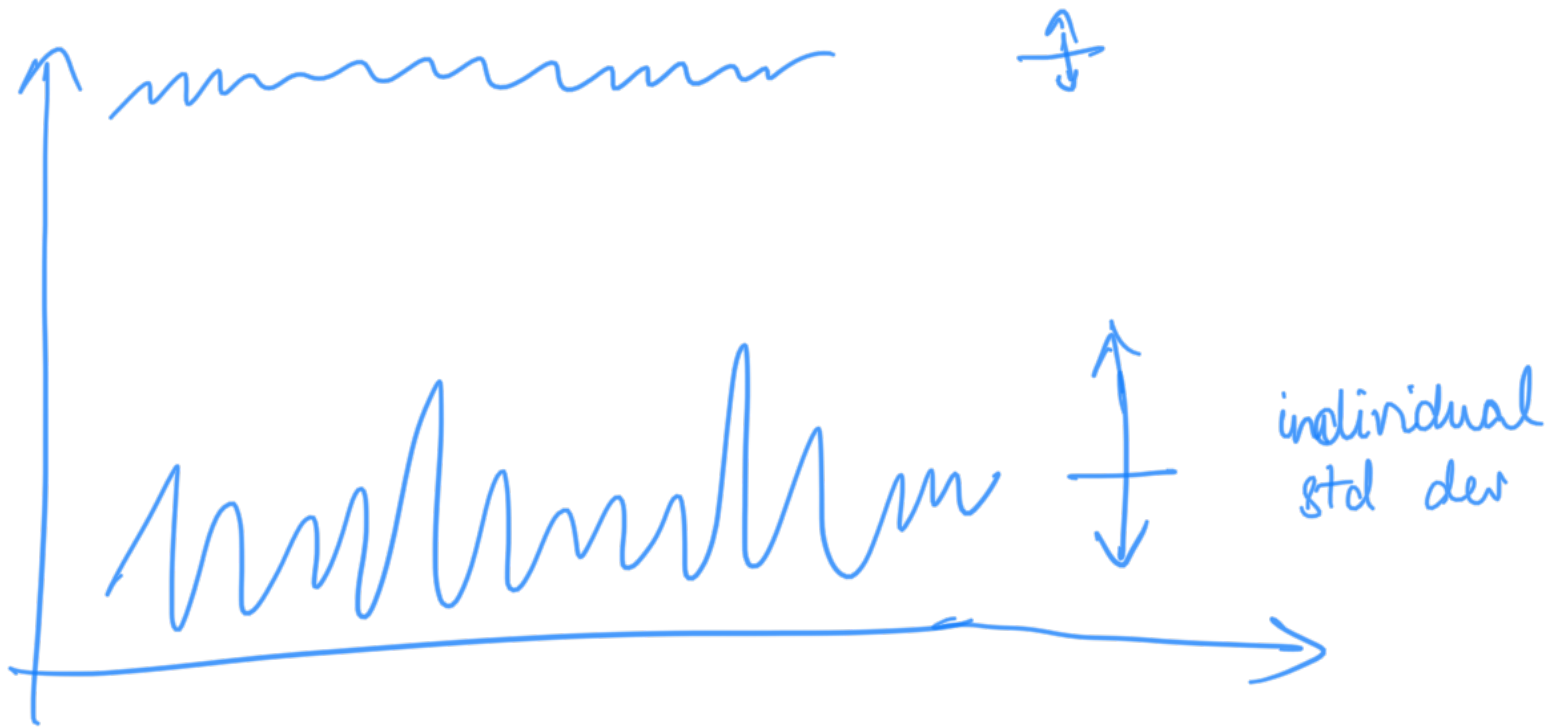
$\hat{V}$ = an estimated variance of the quantity based on all chains ("pooled")

W = average of the estimated variances from each chain individually

Do we remove Gelman (Burn-in)

Idea:

[Draw trace plot ...]



Recommend $R < 1.05$ to declare convergence.

The `coda` function `gelman.plot` plots R versus iterations used, which makes it easier to read how many iterations should be burned.

[Draw example plot ...]

Note: The version of the Gelman-Rubin statistic presented in BSM is not the same as the one produced by R, but it is used similarly.

Posterior Predictive Distributions

JAGS offers two ways to sample from a posterior predictive distribution:

not in dataset

- add a new variable to your model (e.g. "`ynew`")
- add a "missing" value in your data (using "`NA`")

Example: Wikipedia Article Modifications (continued)

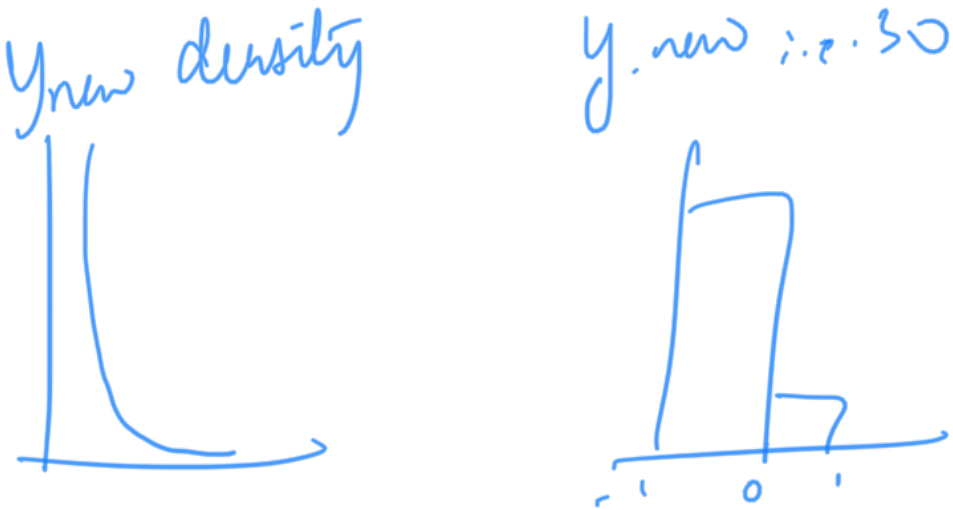
Want:

- a range containing the days since modification for 95% of all articles
- the estimated percentage of articles modified in the last 30 days

We will add a new variable called `ynew` and another variable indicating whether `ynew` is less than or equal to 30.

R/JAGS Example 3.8:

(Transformed) Normal Sample with Posterior Prediction



Notice: Certain transformations (including `log`) are allowed on the left hand side of a deterministic relationship:

```
log(ynew) <- znew
```

Q: Why didn't we use the "missing" value method?

A: Because the transformation of `y[i]` in the data block (on the right hand side of a `<-` statement) requires all non-missing values.