

MCMC Diagnostics

Checking if something is wrong instead of confirming
that all is good

© Dave Campbell 2009

Merck vaccine blocks cervical cancer

Final-stage study proves 100% effective

Breakthrough could lift sagging company, which has been hit hard by Vioxx withdrawal

LINDA A. JOHNSON
ASSOCIATED PRESS

TRENTON, N.J. — The first large study of an experimental cervical cancer vaccine found it was 100-per-cent effective, in the short term, at blocking the most common cause of the disease, the vaccine's maker said yesterday.

Merck's genetically engineered vaccine prevents cervical cancer by blocking infection from the human papilloma virus strains that cause 70 per cent of cervical cancers.

Other types of HPV, which is sexually transmitted, also can cause cervical cancer and painful genital warts. About

20 million Americans have some form of HPV.

The final-stage study of the vaccine included 10,559 sexually active women ages 16 to 26 in the United States and 12 other countries who were not infected with the HPV strains 16 or 18. Half got three vaccine doses over six months; half got dummy shots.

Among those still virus-free after the six months, none who received the vaccine developed either cervical cancer or precancerous lesions likely to turn cancerous over an average two years of followup, compared with 21 who got dummy shots.

"To have 100-per-cent efficacy is something that you have very

rarely," Dr. Eliav Barr, Merck's head of clinical development for the vaccine called Gardasil, told Associated Press. "It's a breakthrough," said Dr. Gloria Bachmann, director of The Women's Health Institute at

The immune system clears most such infections in a year or two, but several types of HPV

ing out the champagn

Merck shares rose

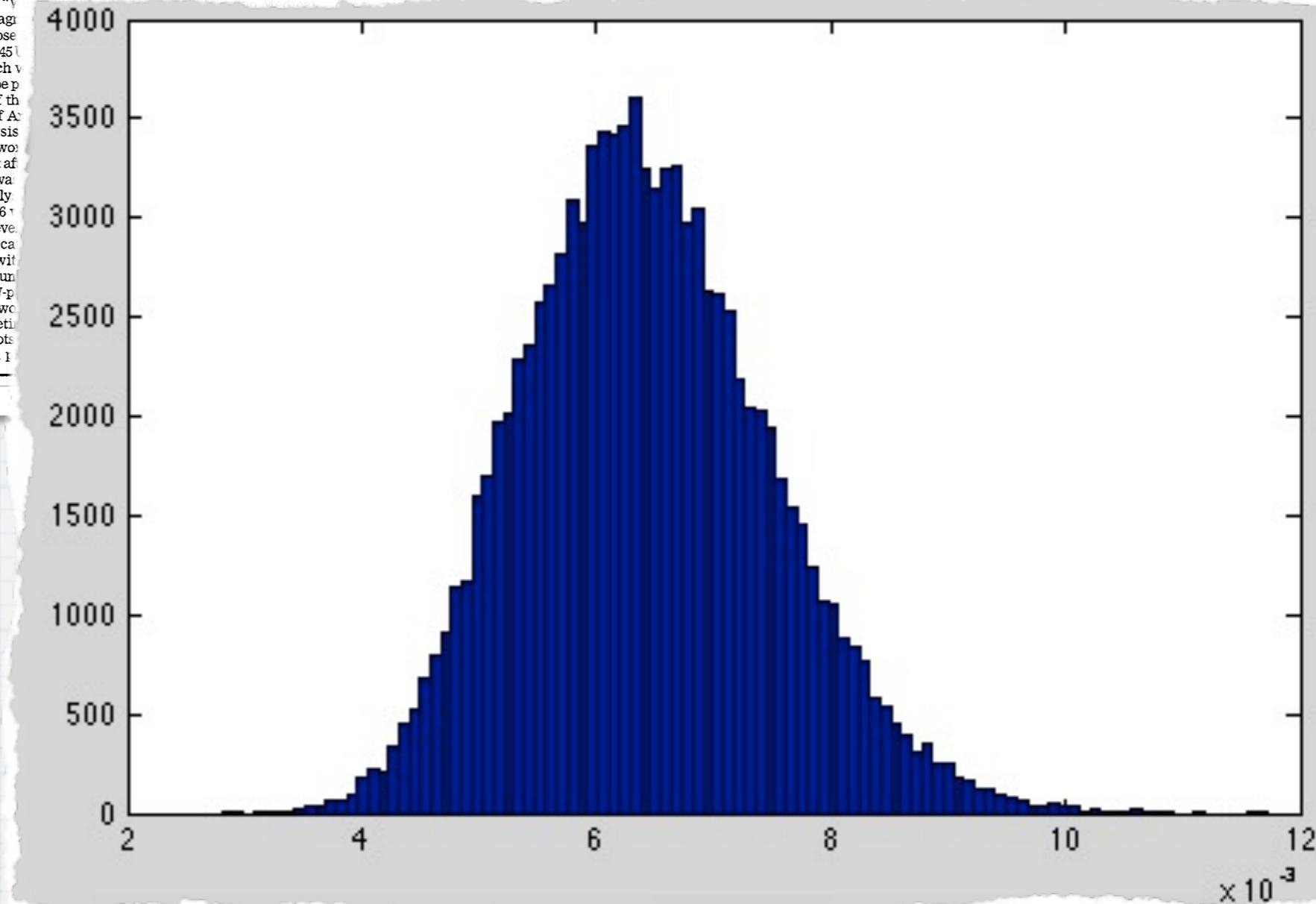
2.1 per cent, to \$27.45.

The study, which was paid for by Merck, was to be presented today at a meeting of the American Society of Clinical Oncology.

A second analysis of the study showed that about 100 more women got the vaccine than did not. All of the 5,736 women who got the vaccine developed either cervical cancer or precancerous lesions, compared with 5,766 who got dummy shots.

Barr said the 97-per-cent efficacy was more "real world" than what patients sometimes delay follow-up shots.

"I see this as a real



hist(beta, 100)

The distribution of β , the probability of getting cancer without getting vaccinated.

1. Start with $\beta_{t-1} = j$
2. Propose a value $X | \beta_{t-1} = i$ from transition probability matrix Q_{ij} as a candidate for β_t
3. compute

$$\alpha_{ij} = \min\left(\frac{P(Y = y | \beta = j)P(\beta = j)P(X = j | \beta = i)}{P(Y = y | \beta = i)P(\beta = i)P(X = i | \beta = j)}, 1\right)$$
5. sample $u \sim \text{Unif}(0,1)$
6. If $u < \alpha_{ij}$ then accept the proposal and set $\beta_t = X$ and if not then set $\beta_t = \beta_{t-1}$.
7. Repeat (N times) until you obtain a sufficient sample from the distribution of $\beta | Y = y$

```

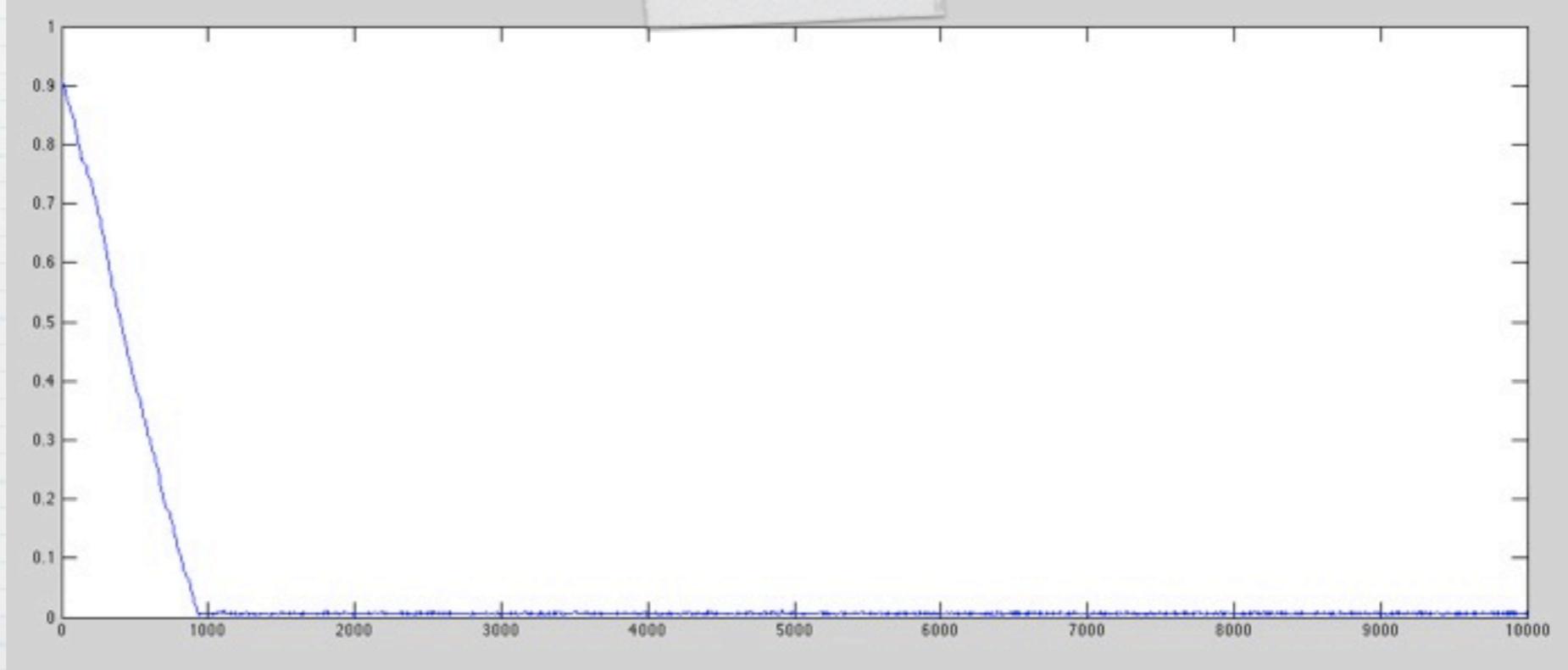
RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(clock)))

niter=10000;
beta=zeros(niter,1);
y=36;
N=5766;
stepvar=.004;
accepts=0;
beta(1)=.89;% although perhaps 36/5766 might be better;
for iter=2:niter
    % propose a value from an easy distribution
    x=unifrnd(beta(iter-1)-stepvar,beta(iter-1)+stepvar);
    % the ratio of un-normalized posteriors. Note that my proposal
    % distribution is symmetric so Q_{ij}=Q_{ji}
    alpha= binopdf(y,N,x)*(2-2*x) / ...
        (binopdf(y,N,beta(iter-1))*(2-2*beta(iter-1)));
    % make a decision
    if(rand<alpha)
        accepts=accepts+1;
        beta(iter)=x;
    else
        beta(iter)=beta(iter-1);
    end
end
acceptance_rate = accepts/niter

```

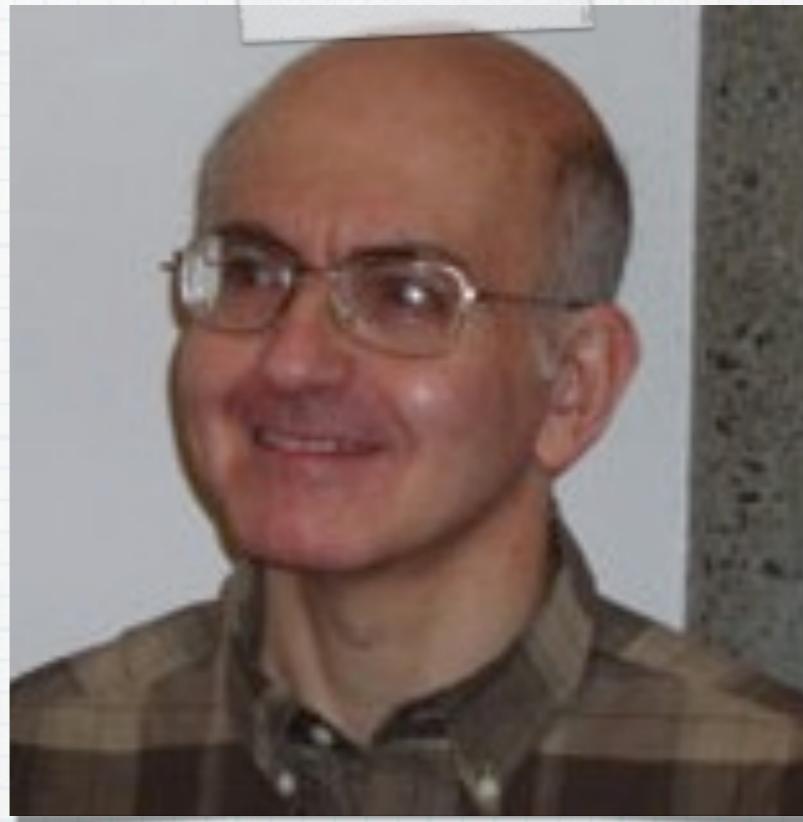
- * The trace plot shows us the path that the Markov Chain takes over time.
- * We use it to look for trends, or look for transience or recurrence

```
plot(beta);
```



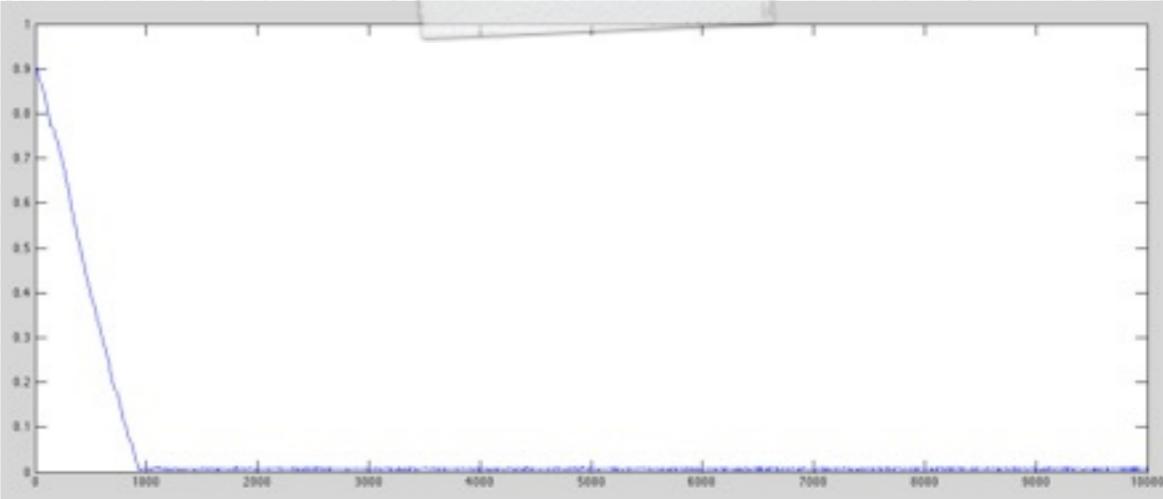
Convergence Diagnostics

- * Did it give us the right answer?
- * We check for ways that it has failed instead of a guarantee of success.



-

Single Chain diagnostic



- * How big should we make N?
How much should we discard for Burn-in?
- * Rule of thumb is a minimum of $N=50,000$ for a good estimate or $N \geq$ 'a lot'
- * Generally we discard at least 'a bunch' or 'half-ish'
- * Raftery Lewis tells us how big to make N based on our needs, and how much to discard as Burn-in

- * It applies to single chains and aims to detect non-convergence to the stationary distribution
- * We supply a quantile (q) that we wish to estimate with a desired precision (r) for its associated (s)% probability interval and a convergence tolerance (δ).
- * It provides bounds for the accuracy of the estimated quantiles of variables

* R-L gives us:

1. nprec (Total number of iterations that should be run)
2. Nburn (the suggested number to discard for burn-in)
3. K (Thinning interval) so that if we keep every k^{th} sample we would have an approximately independent sample.

- * The approach is based on 2 state Markov chain theory and sample size formulas for a binomial variance.
- * The Markov chain $\{\beta_t, t \geq 1\}$ is turned into a binary sequence $\{Z_t, t \geq 1\}$ of indicators of the event $\{\beta_t < \text{cutoff}\}$.
- * cutoff is the empirical quantile q

- * The algorithm then looks for the smallest thinning interval that makes this behaviour like an independent Markov chain.

- * The Burn-in is the number of iterations of the Markov Chain Z_t (the minimum value of t) that it takes for Z_t to approach within the δ of its estimated stationary distribution.

- * Finally, R-L gives us a guess at N_{\min} , the minimum number of samples we need to estimate our quantile to the precision that we desire.

- * User specifies a quantile (such as the 0.025) We will call this quantile 'q'
- * User specifies the desired degree of accuracy (default is $r=0.005$) \leftarrow width of the interval estimate for q
- * Probability of obtaining this accuracy (default is $s=95\%$) \leftarrow to make a 95% interval for q
- * Code gives you N_{min} -- the minimum size of T that you will need to achieve this

Software #1 : (Nearly) CODA for Matlab

- * Download a version of CODA in little pieces here:
- * <http://www.spatial-econometrics.com/gibbs/>
- * and here <http://www.spatial-econometrics.com/html/jplv7.zip>
- * or find it all as a zip on WebCT
- * Check the documentation here: <http://www.spatial-econometrics.com/html/mbook.pdf>

- * Know the quantile to know kind of precision should we expect for the it

```
>> quantile(beta(end/2:end), .975)  
  
ans =  
0.0088
```

- * The function:

```
result = raftery(beta,q,r,s)
```

- * Gives all the output as a struct, access it with `result` or specific fields by `result.nburn`

```
>> q=.975;r=.0005;s=.95;raftery(beta,q,r,s)

ans =

    meth: 'raftery'
    draws: 10000
    nvar: 1
        q: 0.9750
        r: 5.0000e-04
        s: 0.9500
    nburn: 1710
    nprec: 1
    kthin: 1
        kind: 249
        irl: 0.0046
    nmin: 374543
        n: 1711
```

```
>> q=.975;r=.0005;s=.95;raftery(beta,q,r,s)
```

```
ans =
```

```
meth: 'raftery'  
draws: 10000  
nvar: 1  
q: 0.9750  
r: 5.0000e-04  
s: 0.9500  
nburn: 1710  
nprec: 1  
kthin: 1  
kind: 249  
irl: 0.0046  
nmin: 374543  
n: 1711
```



```
>> quantile(beta(end/2:end), .025)
```

```
ans =
```

```
0.0045
```

```
>> q=.025;r=.0005;s=.95;raftery(beta,q,r,s)
```

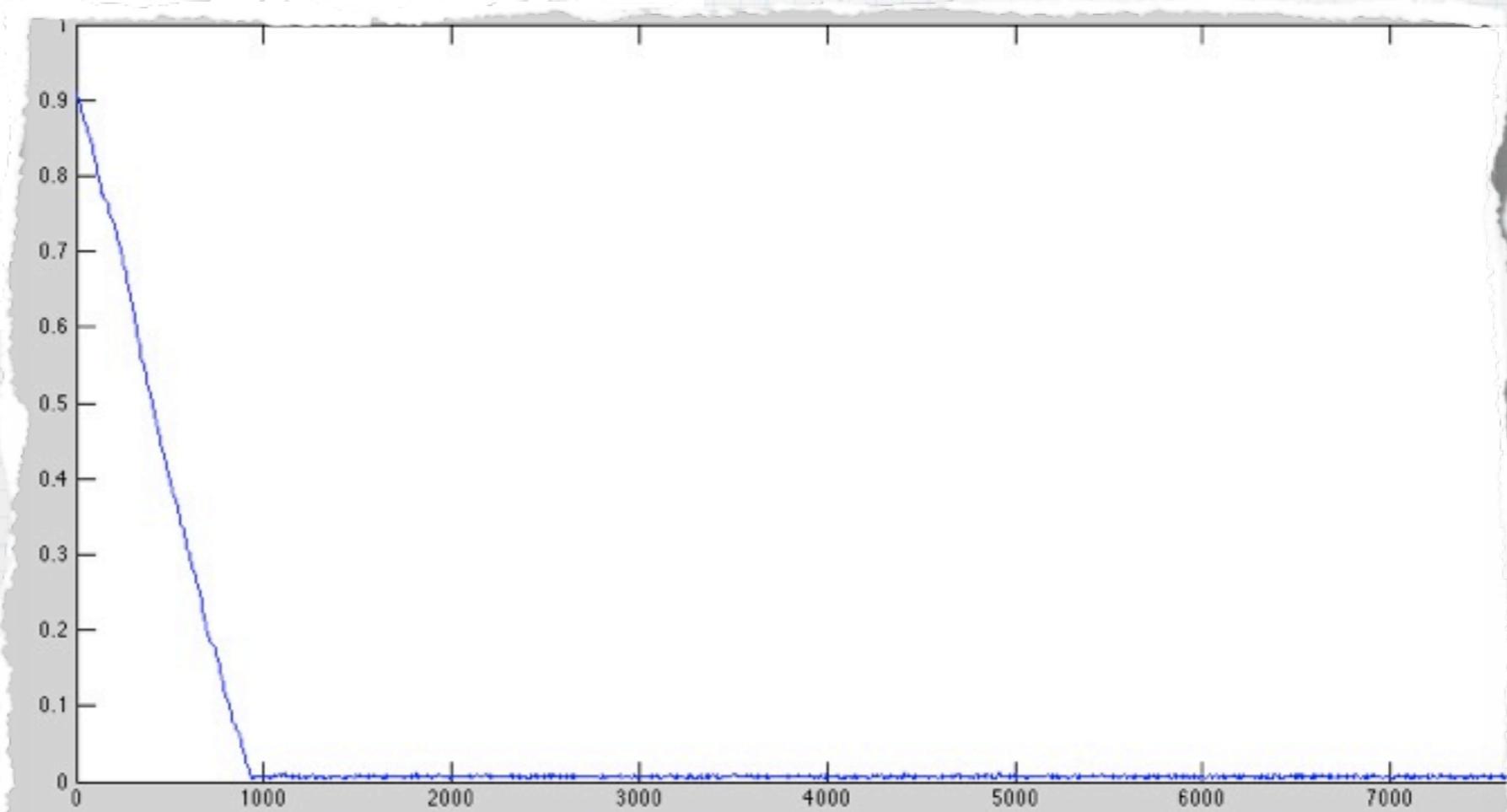
```
ans =
```

```
meth: 'raftery'  
draws: 10000  
nvar: 1  
q: 0.0250  
r: 5.0000e-04  
s: 0.9500  
nburn: 11  
nprec: 1155263  
kthin: 1  
kind: 6  
irl: 3.0845  
nmin: 374543  
n: 1155274
```

```
>> q=.025;r=.0005;s=.95;raftery(beta,q,r,s)
```

```
ans =
```

```
meth: 'raftery'  
draws: 10000  
nvar: 1  
q: 0.0250  
r: 5.0000e-04  
s: 0.9500  
nburn: 11  
nprec: 1155263  
kthin: 1  
kind: 6  
irl: 3.0845  
nmin: 374543  
n: 1155274
```



- * Here R-L told us that we're okay for the 97.5% quantile and essentially agrees with a visual burn-in check
- * For the 2.5% quantile, it suggests taking more samples.
- * 10,000 draws is generally not enough to be used in a publication.
- * 50,000 draws is a more typical rule of thumb.

- * Raftery-Lewis, like all diagnostics is not perfect.
- * Changing quantiles will change the results.
- * We needed to look at the trace plot to understand what was going on and what might be going wrong.
- * We should listen to R-L if it says we need more draws
- * The R-L burn-in is conservative.

Geweke

- * The idea: Think of the markov chain as a time series and use time series methods to check for stationarity.
- * Useful when we want to know about convergence of the mean of β or a function of the sampled variables $g(\beta)$

- * Divide the chain in to the first $p_1\%$ and last $p_2\%$ of the draws
- * sampled values of $g(\beta_t^{(i)})$, $i \geq 1$ form a markov chain.
- * Assumption is that MCMC and function g imply the existence of a spectral density without discontinuity at frequency 0.

- * Estimate the asymptotic variance V_∞ of $g(\beta)$ by the spectral density of $\{g(\beta_t), t \geq 1\}$ at frequency zero and divide by N
- * Square root of V_∞ is the estimate of the standard error of the mean of $g(\beta)$, we call it Numerical Standard Error (NSE)
- * Then compare the means of $g(\beta)$ for the 2 parts of the chain and divide by $\sqrt{V_\infty}$

- * If we get a p-value of $\leq .05$ then we reject the hypothesis that the first $p_1\%$ and the last $p_2\%$ of the sample have the same mean.
- * Then we discard the first $p_1\%$ as burn in and try again.
- * Defaults are $p_1=10\%$ and $p_2=50\%$

- * There are several ways to estimate NSE
- * The spectral window is usually noisy so we smooth it a.k.a. we taper it.
- * Tapering let's us use a moving average with an interval that end our interval softly to deal with end point artifacts.

- * RNE tells us the number of draws needed to provide the same numerical accuracy if the draws were genuinely from an iid posterior sample <-- not as good as R-L Kthin

$$RNE = \frac{\text{var}(g(\beta))}{n * nse^2}$$

- * Values near 1 suggest that our samples are giving us results similar to what we'd get with an iid sample

- * RNE is the relative efficiency of the variance of the sample compared to it's asymptotic variance
- * If $RNE > 1$ sample variance in the chain is smaller than in the iid case and implies it will take longer to obtain convergence than in iid case. Here RNE is like R-L Kthin
- * If $RNE < 1$ our sample will require less samples to achieve the efficiency of the iid case

* Geweke by software part 1

```
>> coda(beta)
```

MCMC CONVERGENCE diagnostics

Based on sample size = 10000

Autocorrelations within each parameter chain

Variable	Lag 1	Lag 5	Lag 10	Lag 50
variable 1	0.998	0.991	0.982	0.912

Raftery-Lewis Diagnostics for each parameter chain

(q=0.0250, r=0.010000, s=0.950000)

Variable	Thin	Burn	Total(N)	(Nmin)	I-stat
variable 1	1	11	2900	937	3.095

Big differences between tapered and un-tapered suggests non iid we should use the tapered results

Geweke Diagnostics for each parameter chain

Variable	Mean	std dev	NSE iid	RNE iid	
variable 1	0.047697	0.153240	0.001532	1.000000	
Variable	NSE 4%	RNE 4%	NSE 8%	RNE 8%	NSE
15%	RNE 15%				
variable 1	0.027322	0.003146	0.033435	0.002101	
0.036496	0.001763				

Geweke Chi-squared test for each parameter chain

First 20% versus Last 50% of the sample

Variable	variable 1	N.S.E.	Chi-sq	Prob
NSE estimate	Mean			
i.i.d.	0.006451	0.000015	0.000000	
4% taper	0.006450	0.000030	0.000264	
8% taper	0.006450	0.000032	0.008188	
15% taper	0.006450	0.000032	0.042999	

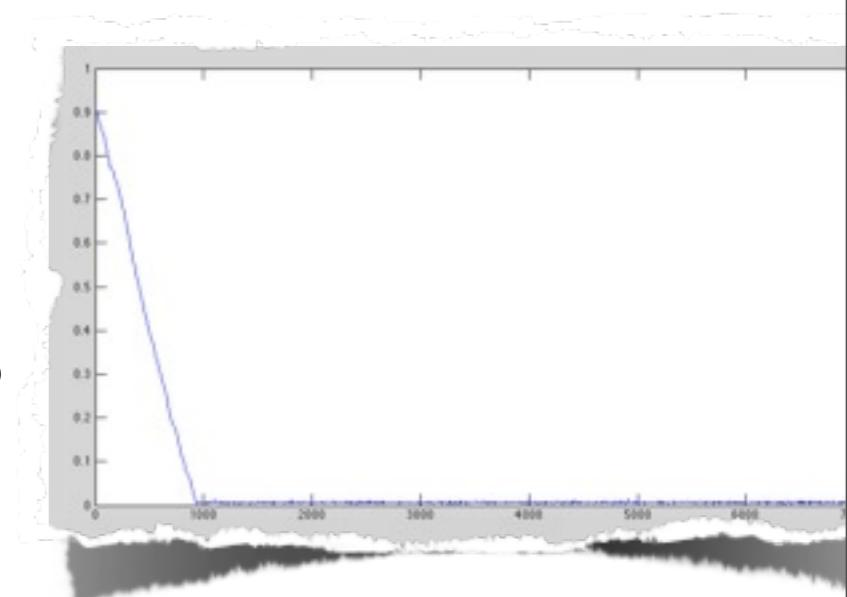
Geweke Diagnostics for each parameter chain

Variable	Mean	std dev	NSE iid	RNE iid	
variable 1	0.047697	0.153240	0.001532	1.000000	
Variable	NSE 4%	RNE 4%	NSE 8%	RNE 8%	NSE
15%	RNE 15%				
variable 1	0.027322	0.003146	0.033435	0.002101	
0.036496	0.001763				

Geweke Chi-squared test for each parameter chain

First 20% versus Last 50% of the sample

Variable	variable 1	N.S.E.	Chi-sq	Prob
NSE estimate	Mean			
i.i.d.	0.006451	0.000015	0.000000	
4% taper	0.006450	0.000030	0.000264	
8% taper	0.006450	0.000032	0.008188	
15% taper	0.006450	0.000032	0.042999	



Geweke after dicarding burn-in

```
>> coda(beta(end/2:end))
```

Geweke Diagnostics for each parameter chain

Variable	Mean	std dev	NSE iid	RNE iid	
variable 1	0.006450	0.001090	0.000015	1.000000	
Variable	NSE 4%	RNE 4%	NSE 8%	RNE 8%	NSE
15% RNE 15%					
variable 1	0.000030	0.267849	0.000032	0.233042	
0.000032	0.229741				

Geweke Chi-squared test for each parameter chain

First 20% versus Last 50% of the sample

Variable	variable 1	N.S.E.	Chi-sq	Prob
NSE estimate	Mean			
i.i.d.	0.006488	0.000019	0.022437	
4% taper	0.006478	0.000034	0.186554	
8% taper	0.006470	0.000034	0.178858	
15% taper	0.006462	0.000032	0.156402	

More generally, altering defaults

```
>> results = coda(beta,vnames,info)
```

- * info.q, info.r, info.s are R-L parameters
- * info.p1 info.p2 are Geweke parameters
- * fid is a file id for printing to a file

- * Geweke gives us:
- * NSE and RNE
- * A test and p-value for stationarity of the mean between 2 intervals
- * The asymptotic standard error of the mean.
- * Use it individually on each parameter
- * Can use it to compare two chains instead of parts of a single chain.