Introduction to Bayesian Statistics and Markov Chain Monte Carlo Estimation

PSYC 943: Fundamentals of Multivariate Modeling

Lecture 12: November 13, 2013

Today's Class

- An introduction to Bayesian statistics:
  - What it is
  - What it does
  - Why people use it

- An introduction to Markov Chain Monte Carlo (MCMC estimation)
  - How it works
  - Features to look for when using MCMC
  - Why people use it

AN INTRODUCTION TO BAYESIAN STATISTICS

Bayesian Statistics: The Basics

- Bayesian statistical analysis refers to the use of models where some or all of the parameters are treated as **random components**
  - Each parameter comes from some type of distribution

- The likelihood function of the data is then augmented with an additional term that represents the likelihood of the **prior distribution** for each parameter
  - Think of this as saying each parameter has a certain likelihood – the height of the prior distribution

- The final estimates are then considered summaries of the **posterior distribution** of the parameter, conditional on the data
  - In practice, we use these estimates to make inferences, just as we have when using the non-Bayesian approaches we have used throughout this class (e.g., maximum likelihood/least squares)

Bayesian Statistics: Why It Is Used

- Bayesian methods get used because the ***relative*** accessibility of one method of estimation (MCMC – to be discussed shortly)

- There are three main reasons why people use MCMC:

1. Missing data
   - Multiple imputation: MCMC is used to estimate model parameters then "impute" data
   - More complicated models for certain types of missing data

2. Lack of software capable of handling large sized analyses
   - Have a zero-inflated negative binomial with 21 multivariate outcomes per 18 time points?

3. New models/generalizations of models not available in software
   - Have a new model?
   - Need a certain link function not in software?

Bayesian Statistics: Perceptions and Issues

- Historically, the use of Bayesian statistics has been controversial
  - The use of certain prior distributions can produce results that are biased or reflect subjective judgment rather than objective science

- Most MCMC estimation methods are **computationally intensive**
  - Until recently, very few methods available for those who aren't into programming in FORTRAN or C++

- Understanding of what Bayesian methods are and how they work is limited outside the field of mathematical statistics
  - Especially the case in the social sciences

- Over the past 15 years, Bayesian methods have become widespread – making new models estimable and becoming standard in some social science fields (quantitative psychology and educational measurement)

HOW BAYESIAN METHODS WORK

How Bayesian Statistics Work

- The term Bayesian refers to Thomas Bayes (1701-1761)
  - Formulated Bayes' Theorem

- Bayesian methods rely on Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

  - $P(A)$ is the **prior distribution (pdf) of A → WHY THINGS ARE BAYESIAN**
  - $P(B)$ is the **marginal distribution (pdf) of B**
  - $P(B|A)$ is the **conditional distribution (pdf) of B, given A**
  - $P(A|B)$ is the **posterior distribution (pdf) of A, given B**

- Bayes' Theorem Example…

Imagine a patient takes a test for a rare disease (present 1% of the population) that has a 95% accuracy rate…what is the probability the patient actually has the disease?

Bayes' Theorem Example

Imagine a patient takes a test for a rare disease (present 1% of the population) that has a 95% accuracy rate…what is the probability the patient actually has the disease?

- D = the case where the person actually has the disease
- ND = the case where the person does not have the disease
- + = the test for the disease is positive

The question is asking for: P(D|+)

From Bayes' Theorem:

$$P(D|+) = \frac{P(+|D)P(D)}{P(+)}$$

What we know:

$$P(D) = .01$$
$$P(+|D) = .95$$

Back to Distributions

- We don't know $P(+)$ directly from the problem, but we can figure it out if we recall how distributions work:

- $P(+)$ is a marginal distribution
- $P(+|D)$ is a conditional distribution

- We can get to the marginal by summing across the conditional:

$$P(+) = P(+|D)P(D) + P(+|ND)P(ND)$$
$$= .95 * .01 + .05 * .99 = .059$$

- So, to figure out the answer, if a person tests positive for the disease, the **posterior probability** they actually have the disease is:

$$P(D|+) = \frac{P(+|D)P(D)}{P(+)} = \frac{.01 * .99}{.059} = .17$$

A (Perhaps) More Relevant Example

- The old-fashioned Bayes' Theorem example I've found to be difficult to generalize to your actual data, so…

- Imagine you administer an IQ test to a sample of 50 people
  - y_p = person p's IQ test score

- To put this into a linear-models context, the empty model for Y:

$$y_p = \beta_0 + e_p$$

Where $e_p \sim N(0, \sigma_e^2)$

- From this empty model, we know that:
  - β_0 is the mean of the Y (the mean IQ)
  - σ_e^2 is the sample variance of Y
  - The conditional distribution of Y is then: $f\left(y_p \middle| \beta_0, \sigma_e^2\right) \sim N(\beta_0, \sigma_e^2)$

Non-Bayesian Analysis (i.e., Frequentist Approach)

- Up to this point in the class, we have analyzed these data using ML

- For ML, we maximized the joint likelihood of the sample with respect to the two unknown parameters β_0 and σ_e^2

$$L(\beta_0, \sigma_e^2) = \prod_{p=1}^{N} f\left(y_p \middle| \beta_0, \sigma_e^2\right) = \prod_{p=1}^{N} \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{\left(y_p - \beta_0\right)^2}{2\sigma_e^2}\right)$$

- Here, using PROC MIXED, I found:

$$\beta_0 = 102.769$$
$$\sigma_e^2 = 239.490$$

- Also, I found:

$$-2LogL = 415.8$$

Setting up a Bayesian Approach

- The (fully) Bayesian approach would treat each parameter as a random instance from some **prior distribution**

- Let's say you know that this version of the IQ test is supposed to have a mean of 100 and a standard deviation of 15
  - So β_0 should be 100 and σ_e^2 should be 225

- Going a step further, let's say you have seen results for administrations of this test that led you to believe that the mean came from a normal distribution with a SD of 2.13
  - This indicates the prior distribution for the **mean**…or

$$f(\beta_0) \sim N(100, 2.13^2)$$

- Let's also say that you don't really have an idea as for the distribution of the variance, but you have seen it range from 200 to 400, so we can come up with a prior distribution for the **variance** of:

$$f(\sigma_e^2) \sim U(200, 400)$$

- Here the prior is a uniform distribution meaning all values from 200 to 400 are equally likely

More on the Bayesian Approach

- The Bayesian approach is now to seek to find the **posterior distribution** of the parameters given the data:

$$f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- We can again use Bayes' Theorem (but for continuous parameters):

$$f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right) = \frac{f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0, \sigma_e^2)}{f\left(\boldsymbol{y}_p\right)} = \frac{f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2)}{f\left(\boldsymbol{y}_p\right)}$$

- Because $f\left(\boldsymbol{y}_p\right)$ essentially is a constant (which involves integrating across β_0 and σ_e^2 to find its value), this term is often referred to as:

$$f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right) \propto f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2)$$

- The symbol $\propto$ is read as "is proportional to" – meaning it is the same as when multiplied by a constant
  - So it is the same for all values of β_0 and σ_e^2

Unpacking the Posterior Distribution

- $f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right)$ is the **conditional distribution** of the data given the parameters – we know this already from our linear model (slide 12)

$$f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) = \prod_{p=1}^{N} f\left(y_p \middle| \beta_0, \sigma_e^2\right) = \prod_{p=1}^{N} \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{\left(y_p - \beta_0\right)^2}{2\sigma_e^2}\right)$$

- $f(\beta_0)$ is the **prior distribution** of β_0, which we decided would be $N(100, 2.13^2)$, giving the height of any β_0:

$$f(\beta_0) = \frac{1}{\sqrt{2\pi\sigma_{\beta_0}^2}} \exp\left(-\frac{\left(\beta_0 - \mu_{\beta_0}\right)^2}{2\sigma_{\beta_0}^2}\right)$$

$$= \frac{1}{\sqrt{2\pi * 2.13^2}} \exp\left(-\frac{(\beta_0 - 100)^2}{2 * 2.13^2}\right)$$

Unpacking the Posterior Distribution

- $f(\sigma_e^2)$ is the **prior distribution** of σ_e^2, which we decided would be $\mathrm{U}(200,400)$, giving the height of any value of σ_e^2 as:

$$f(\sigma_e^2) = \frac{1}{b_{\sigma_e^2} - a_{\sigma_e^2}} = \frac{1}{400 - 200} = \frac{1}{200} = .005$$

- Some useful terminology:
  - The parameters of the model (for the data) get prior distributions
  - The prior distributions each have parameters – these parameters are called **hyper-parameters**
  - The hyper-parameters are not estimated in our example, but could be – giving us a case where we would call our priors **empirical priors**
    - **AKA random intercept variance**

Up Next: Estimation (first using non-MCMC)

- Although MCMC is commonly thought of as the only method for Bayesian estimation, there are several other forms

- The form analogous to ML (where the value of the parameters that maximize the likelihood or log-likelihood) is called **Maximum a Posteriori estimation (MAP)**
  - The term modal comes from the maximum point coming at the peak (the mode) of the posterior distribution

- In practice, this functions similar to ML, only instead of maximizing the joint likelihood of the data, we now have to worry about the prior:

$$f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right) = \frac{f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2)}{f\left(\boldsymbol{y}_p\right)} \propto f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2)$$

- Because it is often more easy to work with, the log of this is often used:

$$\log\left(f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right)\right) \propto \log f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) + \log f(\beta_0) + \log f(\sigma_e^2)$$

Grid Searching for the MAP Estimate of $\boldsymbol{\beta_0}$

- To demonstrate, let's imagine we know $\sigma_e^2 = 239.490$
  - Later we won't know this…when we use MCMC

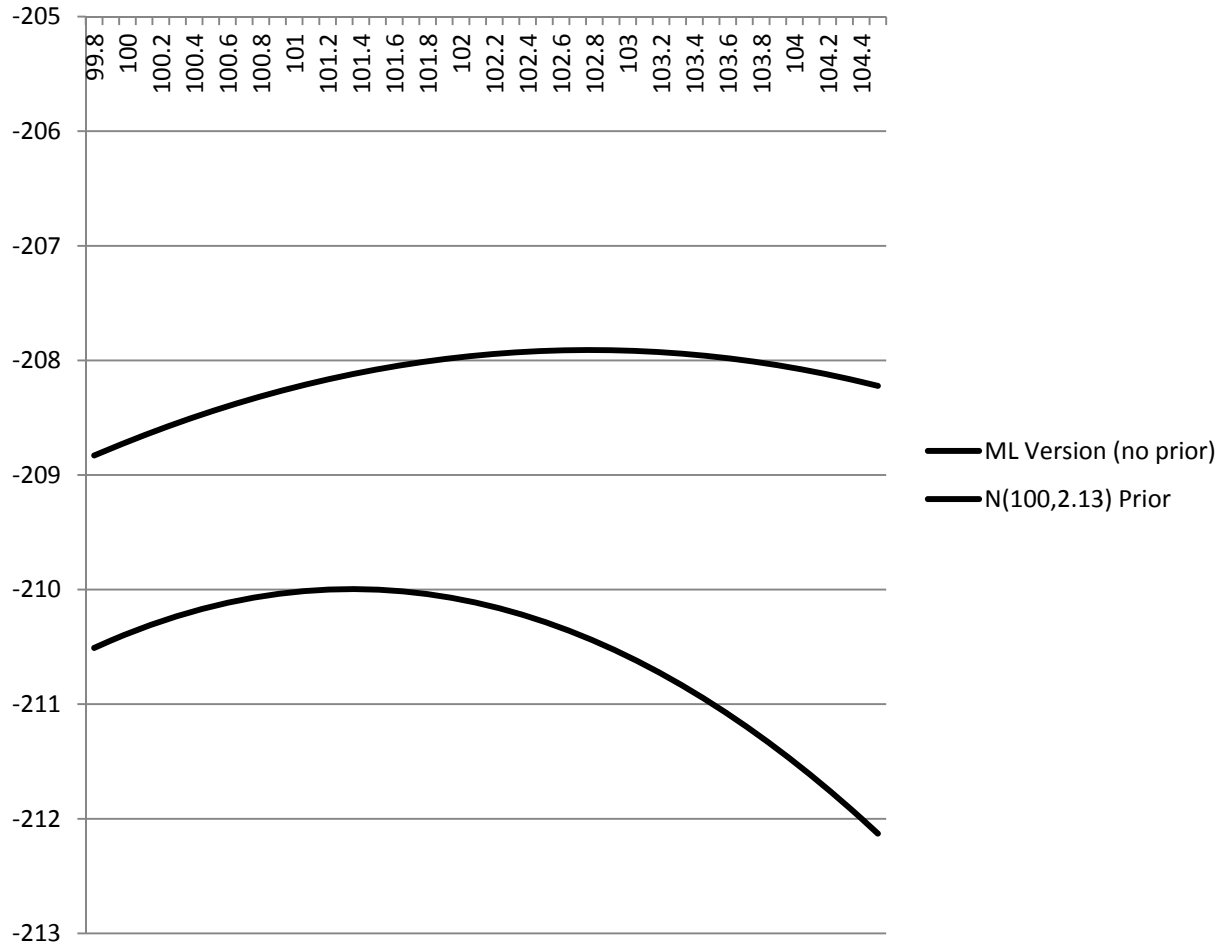
- We will use Excel to search over a grid of possible values for β_0

- In each, we will use $\log f\left(\boldsymbol{y}_p \middle| \beta_0\right) + \log f\left(\beta_0\right)$

- As a comparison, we will also search over the ML log likelihood function $\log f\left(\boldsymbol{y}_p \middle| \beta_0\right)$

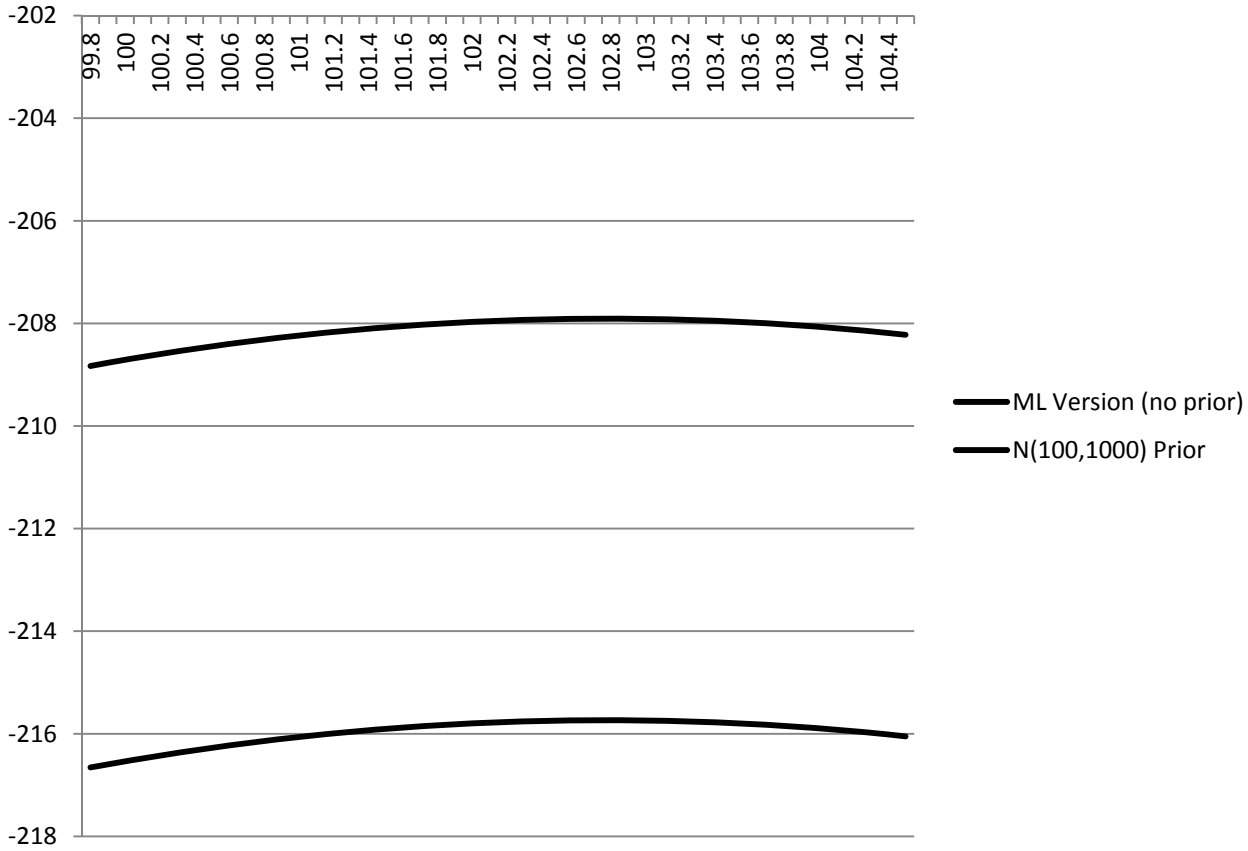
ML v. Prior for β_0 of N(100, 2.13^2)

- Maximum for ML: 102.8
- Maximum for Bayes: 101.4 (estimate is closer to mean of prior)



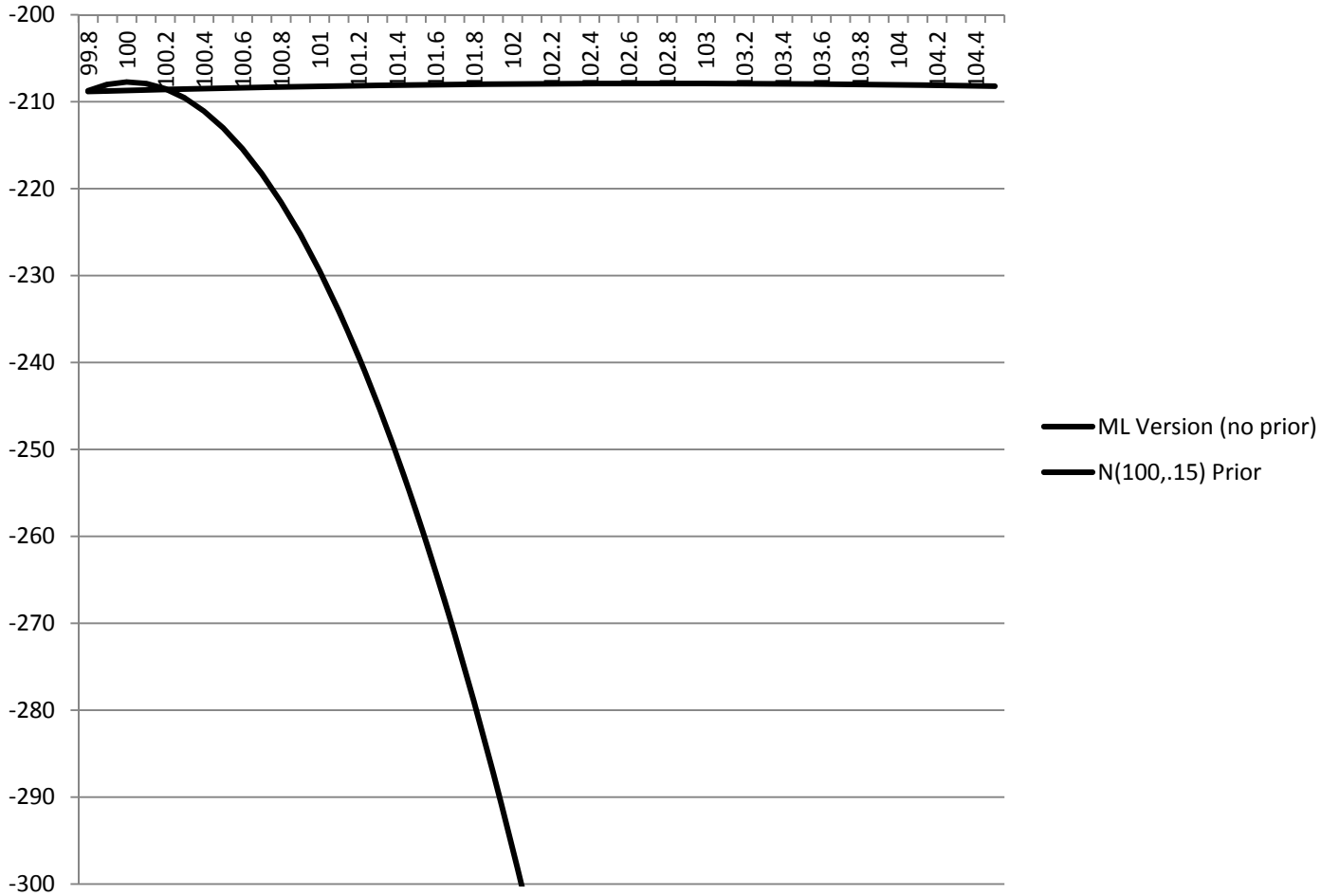
ML vs. Prior for β_0 of N(100, 1000^2)

- Maximum for ML: 102.8
- Maximum for Bayes: 102.8



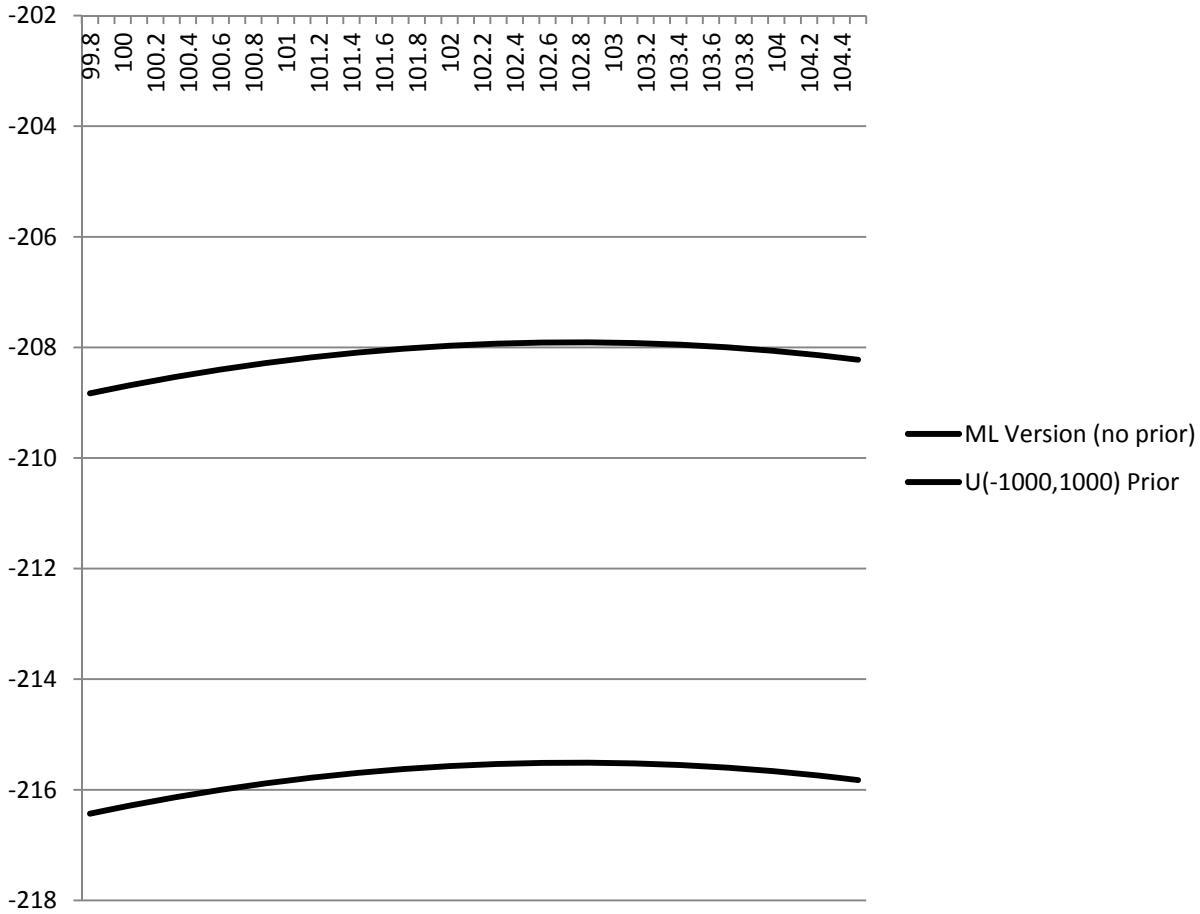
ML vs. Prior for β_0 of N(100, 0.15^2)

- Maximum for ML: 102.8
- Maximum for Bayes: 100



ML vs. Prior for β_0 of U(-1000,1000)

- Maximum for ML: 102.8
- Maximum for Bayes: 102.8



Summarizing Bayesian So Far

- Bayesian → parameters have prior distributions

- Estimation in Bayesian → MAP estimation is much like estimation in ML, only instead of likelihood of data, now have to add in likelihood for prior of all parameters
  - But…MAP estimation may be difficult as figuring out derivatives for gradient function (for Newton Raphson) are not always easy
  - Where they are easy: **Conjugate** priors → prior distributions that are the same as the posterior distribution (think multilevel with normal outcomes)

- Priors can be **informative** (highly peaked) or **uninformative** (not peaked)
  - Some uninformative priors will give MAP estimates that are equal to ML

- Up next: estimation by brute force: Markov Chain Monte Carlo

MARKOV CHAIN MONTE CARLO ESTIMATION: THE BASICS

How Estimation Works (More or Less)

- Most estimation routines do one of three things:

1. **Minimize Something:** Typically found with names that have "least" in the title. Forms of least squares include "Generalized", "Ordinary", "Weighted", "Diagonally Weighted", "WLSMV", and "Iteratively Reweighted." Typically the estimator of last resort…

2. **Maximize Something:** Typically found with names that have "maximum" in the title. Forms include "Maximum likelihood", "ML", "Residual Maximum Likelihood" (REML), "Robust ML". Typically the gold standard of estimators (and we now know why).

3. **Use Simulation to Sample from Something:** more recent advances in simulation use resampling techniques. Names include "Bayesian Markov Chain Monte Carlo", "Gibbs Sampling", "Metropolis Hastings", "Metropolis Algorithm", and "Monte Carlo". Used for complex models where ML is not available or for methods where prior values are needed.

How MCMC Estimation Works

- MCMC estimation works by taking samples from the posterior distribution of the data given the parameters:

$$f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right) = \frac{f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2)}{f\left(\boldsymbol{y}_p\right)}$$

  - How is that possible? We don't know $f(\boldsymbol{y}_p)$…but…we'll see…

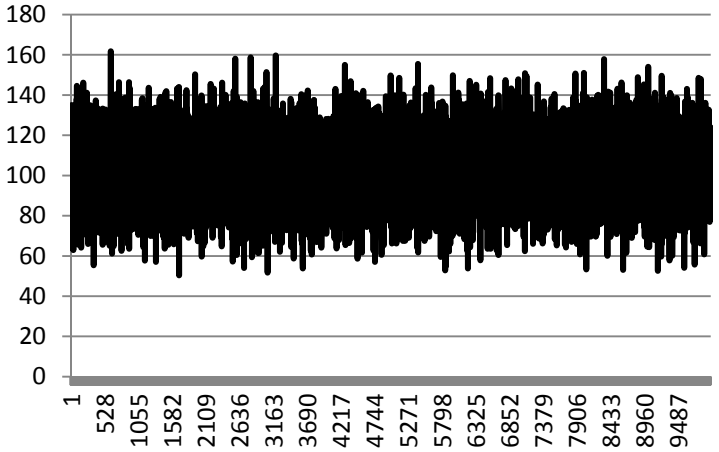
- After enough values are drawn, a rough shape of the distribution can be formed
  - From that shape we can take summaries and make them our parameters (i.e., mean)

- How the sampling mechanism happens comes from several different algorithms that you will hear about, the most popular being:
  - **Gibbs Sampling**: used when $f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right)$ is known
    - Parameter values are drawn and kept throughout the chain
  - **Metropolis-Hastings (within Gibbs):** used when $f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right)$ is unknown
    - Parameter values are proposed, then either kept or rejected
    - SAS PROC MCMC uses the latter
    - TRIVIA NOTE: The Metropolis algorithm comes from Chemistry (in 1950)

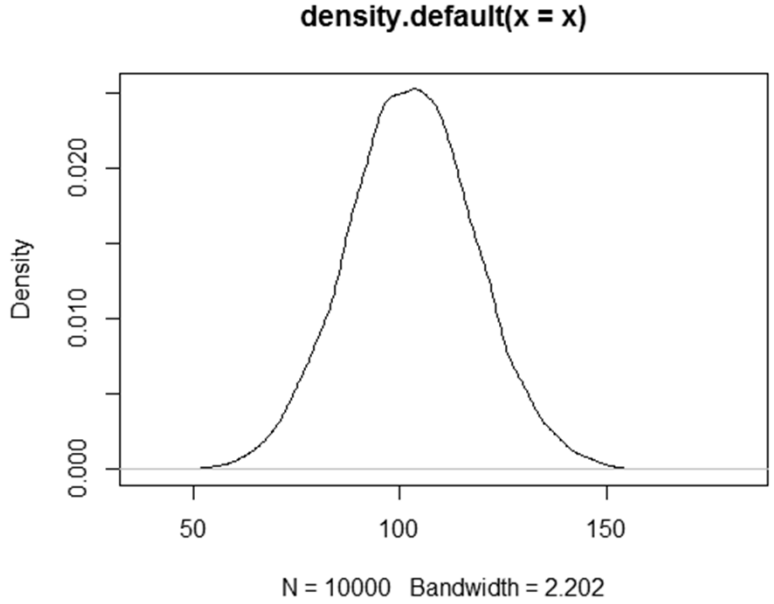
- In some fields (Physics in particular), MCMC estimation is referred to as Monte Carlo estimation

Sampling Example

- Imagine I wanted to get the shape of a distribution similar to our IQ example (with a mean of 102.8 and a variance of 239.5)
  - This is essentially Gibbs Sampling
- I will open Excel and draw 10,000 random values from N(102.8, 239.5)
  - You can do this by typing "=norminv(rand(),102.8,SQRT(239.5))"




```
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  44.62   92.52  102.80  102.80  113.20  176.70
> var(x)
[1] 240.7343
```



MCMC Estimation with MHG

- The Metropolis-Hastings algorithm works a bit differently than Gibbs sampling:

1. Each parameter (here β_0 and σ_e^2) is given an initial value
2. In order, a new value is proposed for each model parameter from some distribution:

$$\beta_0^* \sim Q(\beta_0^* | \beta_0); \sigma_e^{2^*} \sim Q\left(\sigma_e^{2^*} \middle| \sigma_e^2\right)$$

3. The proposed value is then accepted as the current value with probability $\max(r_{MHG}, 1)$:

$$r_{MHG} = \frac{f\left(\boldsymbol{y}_p \middle| \beta_0^*, \sigma_e^{2^*}\right) f(\beta_0^*) f\left(\sigma_e^{2^*}\right) Q\left(\beta_0 \middle| \beta_0^*\right) Q\left(\sigma_e^2 \middle| \sigma_e^{2^*}\right)}{f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2) Q(\beta_0^* | \beta_0) Q\left(\sigma_e^{2^*} \middle| \sigma_e^2\right)}$$

4. The process continues for a pre-specified number of iterations (more is better)

Notes About MHG

- The constant in the denominator of the posterior distribution:

$$f\left(\beta_0, \sigma_e^2 \middle| \boldsymbol{y}_p\right) = \frac{f\left(\boldsymbol{y}_p \middle| \beta_0, \sigma_e^2\right) f(\beta_0) f(\sigma_e^2)}{f\left(\boldsymbol{y}_p\right)}$$

…cancels when the ratio is formed

- The proposal distributions $Q(\beta_0^* | \beta_0)$ and $Q\left(\sigma_e^{2^*} \middle| \sigma_e^2\right)$ can literally be any statistical distribution
  - The trick is picking ones that make the chain "converge" quickly
  - Want to find values that lead to moderate number of accepted parameters
  - SAS PROC MCMC/WINBUGS don't make you pick these

- Given a long enough chain, the final values of the chain will come from the posterior distribution
  - From that you can get your parameter estimates

Introducing…SAS PROC MCMC

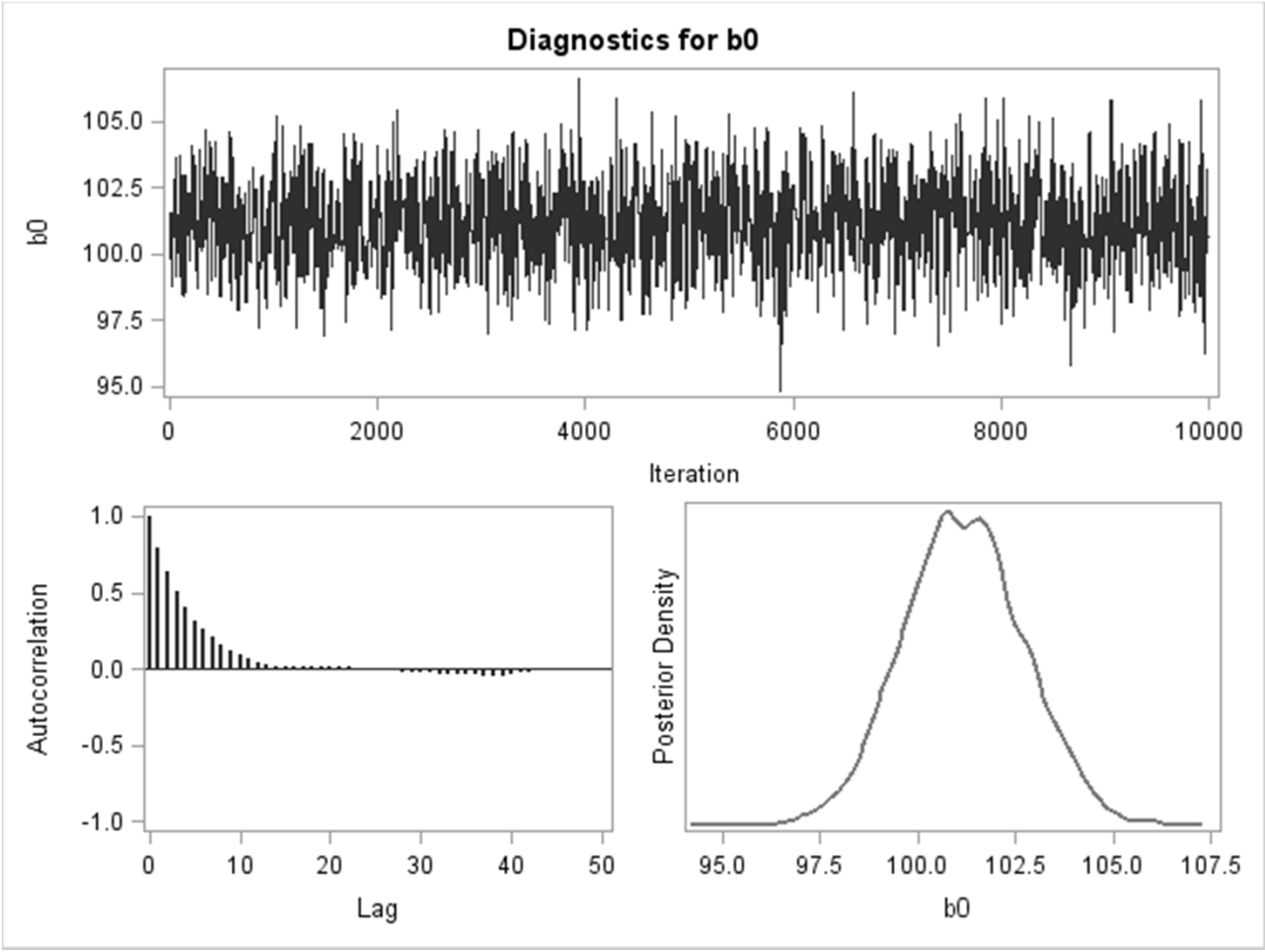
```
*INITIAL PROC MCMC RUN: PRIORS FROM SLIDE 13 - but without burnin period;
PROC MCMC DATA=work.normalgen OUTPOST=work.outpost SEED=10252012
          NBI=0 THIN=1 NMC=10000 DIC;
    PARMS b0, errorvar;
    PRIOR b0 ~ N(100,SD=2.13);
    PRIOR errorvar ~ UNIFORM(200,400);

    condmean_y = b0;
    MODEL y ~ N(condmean_y,VAR=errorvar);
RUN;
```

- SEED: random number seed (same number = same output)
- NBI: number of burn in iterations (more on this soon)
- THIN: thinning interval (more on this soon)
- NMC: number of total iterations
- PARMS: list **all** model parameters
- PRIOR: specify priors for each parameter
- MODEL: specify model for the data (note: MODEL is different from previous SAS PROCs in that you must specify distribution)

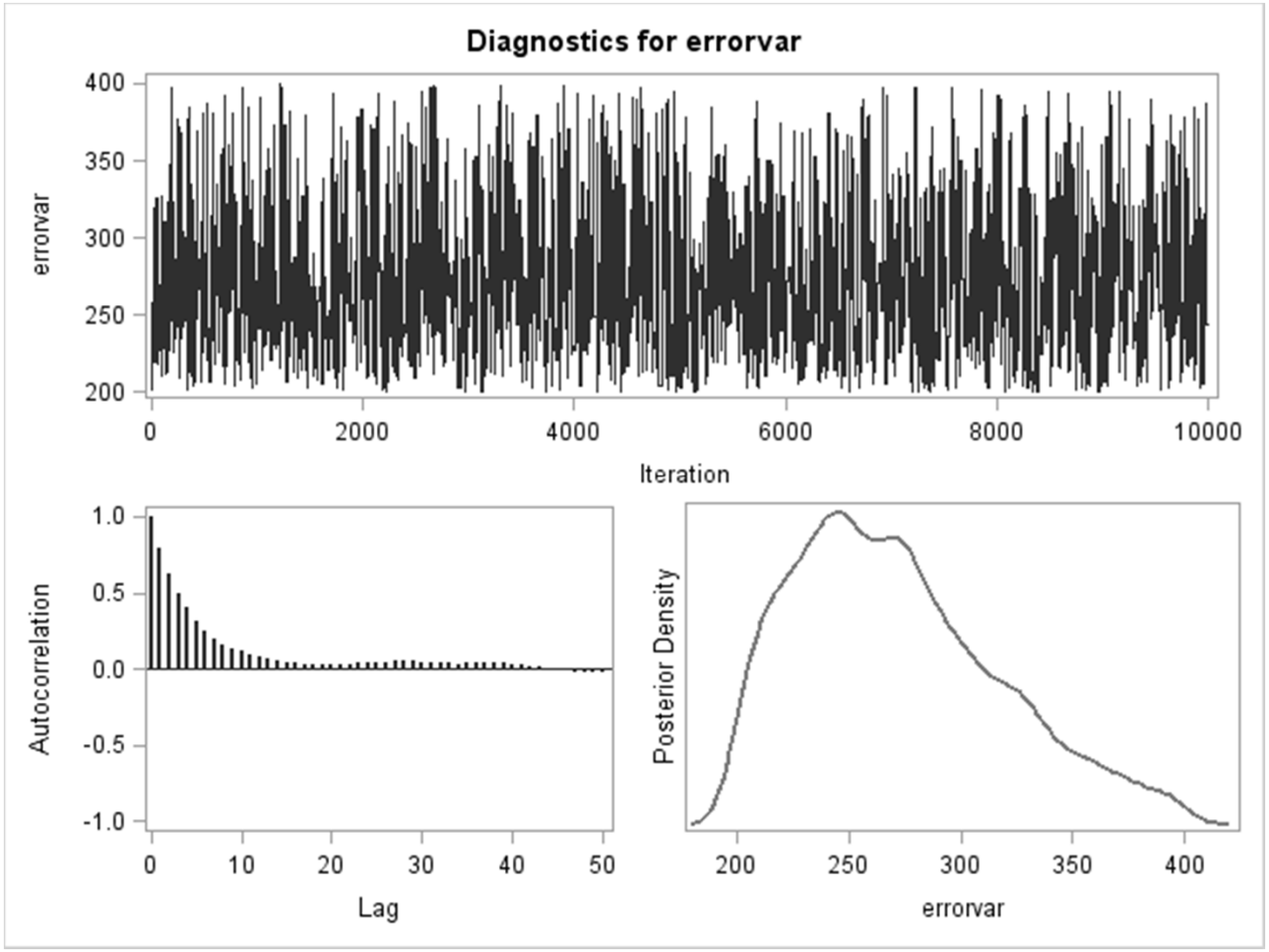
Iteration History from SAS

	Iteration	b0	errorvar	Log Prior Density	Log-Likelihood Value	Log Posterior Density
1	1	101.2	205.7	-7.1297	-208.5	-215.6
2	2	100.3	258.5	-6.9864	-208.5	-215.5
3	3	101.6	201.0	-7.2473	-208.5	-215.7
4	4	100.2	256.3	-6.9781	-208.6	-215.6
5	5	100.2	256.3	-6.9781	-208.6	-215.6
6	6	99.8318	228.9	-6.9765	-208.9	-215.9
7	7	99.8318	228.9	-6.9765	-208.9	-215.9
8	8	100.5	224.4	-7.0026	-208.5	-215.5
9	9	100.5	224.4	-7.0026	-208.5	-215.5
10	10	100.5	224.4	-7.0026	-208.5	-215.5
11	11	100.5	224.4	-7.0026	-208.5	-215.5
12	12	100.5	224.4	-7.0026	-208.5	-215.5
13	13	100.5	224.4	-7.0026	-208.5	-215.5
14	14	100.5	224.4	-7.0026	-208.5	-215.5
15	15	98.8215	275.7	-7.1264	-209.6	-216.7
16	16	98.8215	275.7	-7.1264	-209.6	-216.7
17	17	101.4	318.8	-7.1909	-209.0	-216.2
18	18	101.4	318.8	-7.1909	-209.0	-216.2
19	19	100.4	228.1	-6.9899	-208.6	-215.6
20	20	100.4	228.1	-6.9899	-208.6	-215.6
21	21	101.4	306.1	-7.1903	-208.8	-215.9
22	22	99.9909	280.8	-6.9734	-208.9	-215.9
23	23	99.9909	280.8	-6.9734	-208.9	-215.9
24	24	99.9909	280.8	-6.9734	-208.9	-215.9
25	25	101.6	219.7	-7.2484	-208.2	-215.4
26	26	101.6	219.7	-7.2484	-208.2	-215.4
27	27	101.6	219.7	-7.2484	-208.2	-215.4
28	28	101.6	219.7	-7.2484	-208.2	-215.4
29	29	101.6	219.7	-7.2484	-208.2	-215.4
30	30	101.6	219.7	-7.2484	-208.2	-215.4
31	31	101.6	219.7	-7.2484	-208.2	-215.4
32	32	101.6	219.7	-7.2484	-208.2	-215.4
33	33	101.6	219.7	-7.2484	-208.2	-215.4
34	34	101.7	245.9	-7.3065	-208.0	-215.3
35	35	101.7	245.9	-7.3065	-208.0	-215.3
36	36	101.7	245.9	-7.3065	-208.0	-215.3
37	37	101.7	245.9	-7.3065	-208.0	-215.3
38	38	101.7	245.9	-7.3065	-208.0	-215.3
39	39	101.7	245.9	-7.3065	-208.0	-215.3

Examining the Chain and Posteriors



Examining the Chain and Posteriors



Practical Specifics in MCMC Estimation

- A **burn-in** period is used where a chain is run for a set number of iterations before the sampled parameter values are used in the posterior distribution

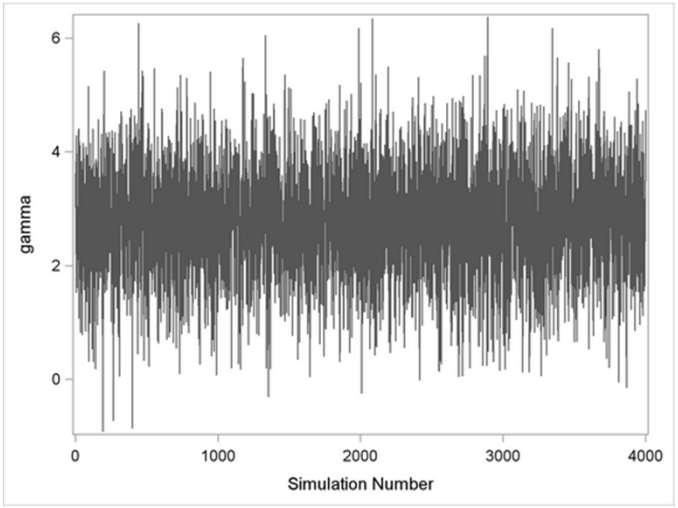
- Because of the rejection/acceptance process, any two iterations are likely to have a high correlation (called **autocorrelation**) → posterior chains use a **thinning interval** to take every Xth sample to reduce the autocorrelation
  - A high autocorrelation may indicate the standard error of the posterior distribution will be smaller than it should be

- The **chain length** (and sometimes number of chains) must also be long enough so the rejection/acceptance process can reasonably approximate the posterior distribution

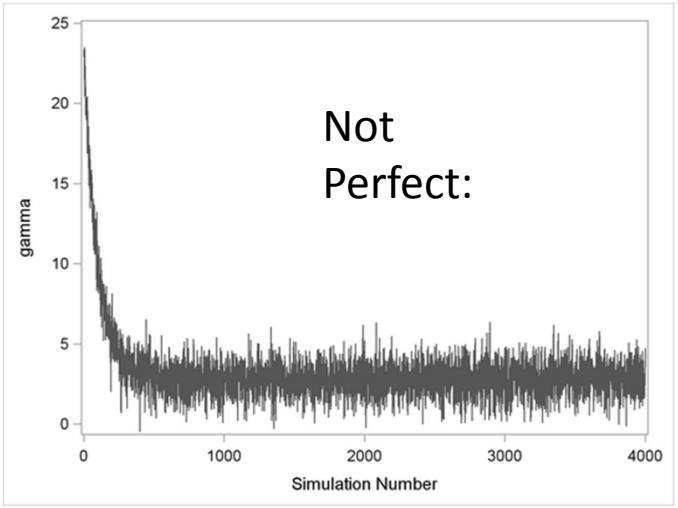
- How does one what values to pick for these? Output diagnostics
  - Trial. And. Error.

Best Output Diagnostics: the Eye Ball Test

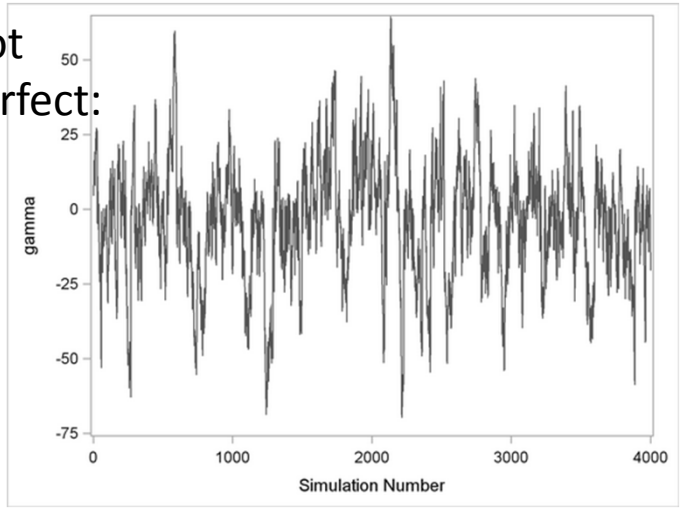
Perfect:



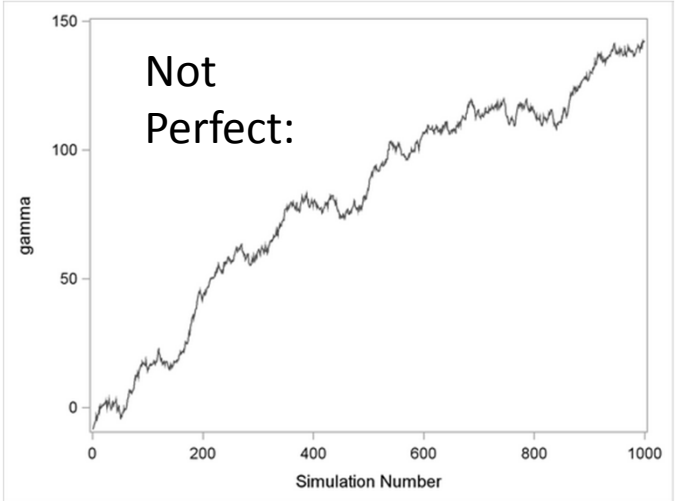
Not Perfect:



Not Perfect:



Not Perfect:



Output Diagnostic Statistics

Geweke Diagnostics		
Parameter	z	Pr > |z|
b0	-0.4489	0.6535
errorvar	1.0657	0.2865

Posterior Autocorrelations				
Parameter	Lag 1	Lag 5	Lag 10	Lag 50
b0	0.7942	0.3163	0.0929	-0.0074
errorvar	0.7914	0.3189	0.1158	-0.0280

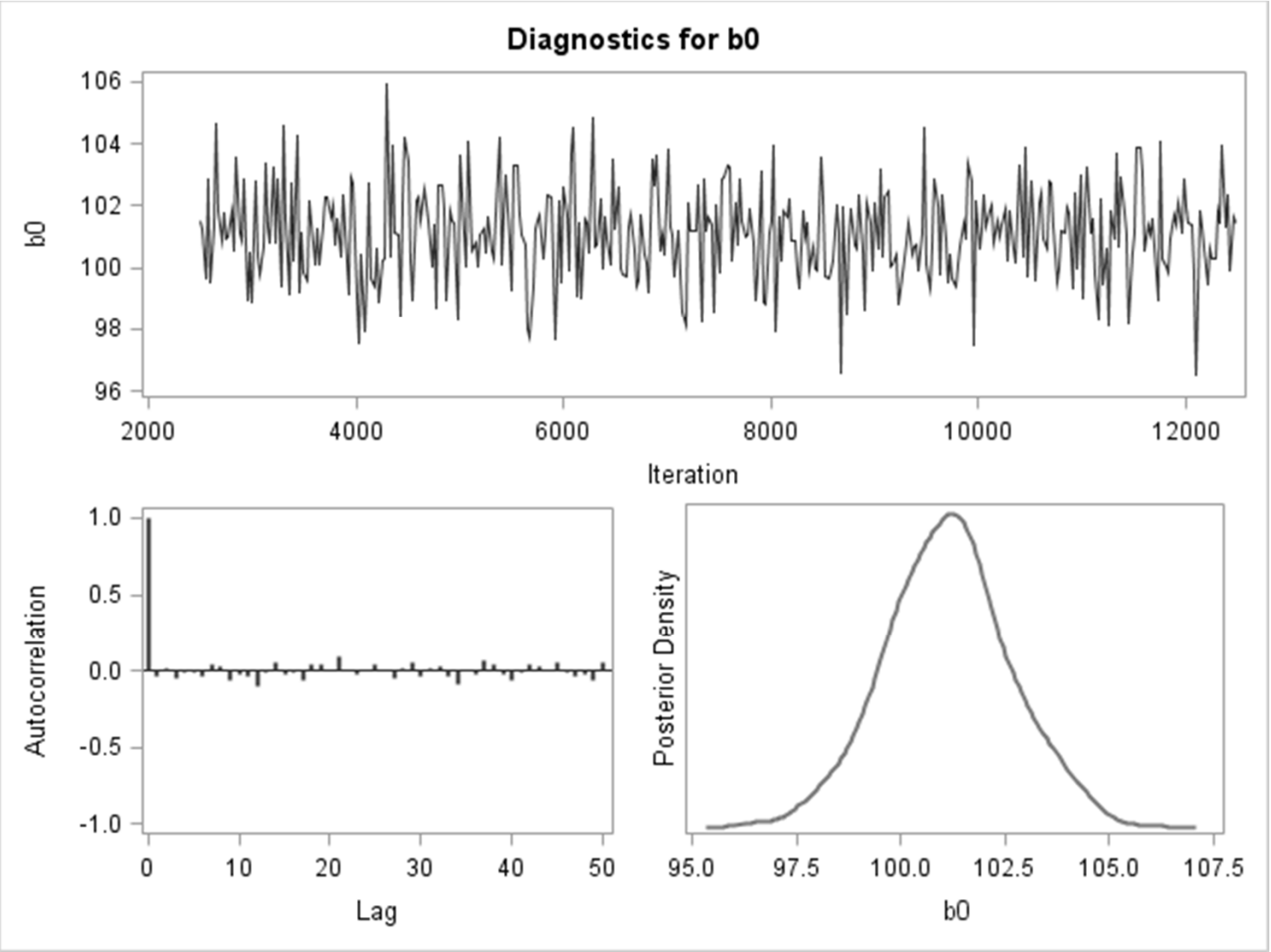
Rerunning Our Analysis: Burn-in and Thinning Interval

- Burn-in = 2500; thinning interval = 25; chain length = 10000

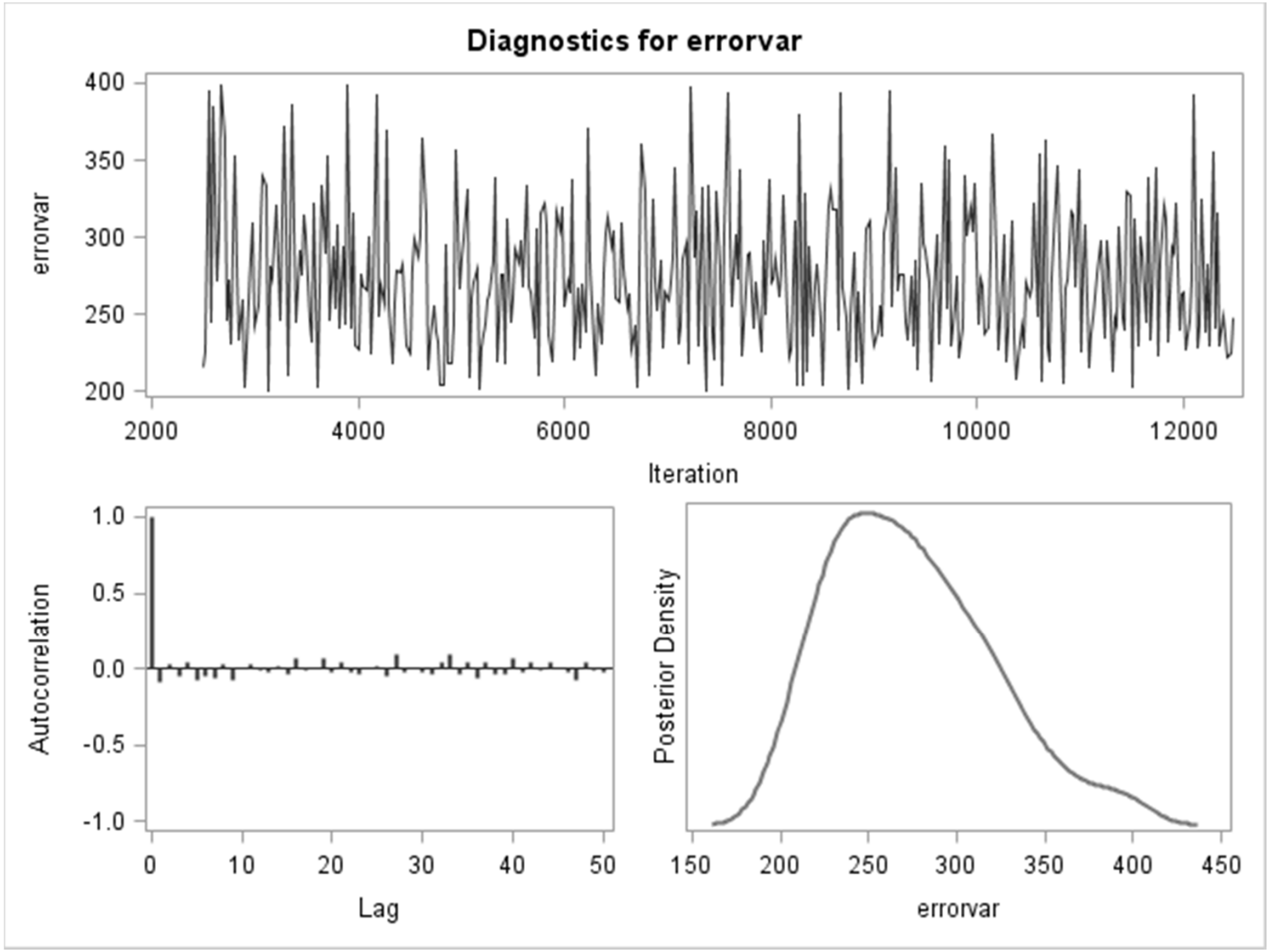
Posterior Autocorrelations				
Parameter	Lag 1	Lag 5	Lag 10	Lag 50
b0	-0.0402	-0.0128	-0.0198	0.0493
errorvar	-0.0832	-0.0751	-0.0033	-0.0188

Geweke Diagnostics		
Parameter	z	Pr > |z|
b0	1.7722	0.0764
errorvar	1.7334	0.0830

Chain Plots


Diagnostics for b0
b0
106
104
102
100
98
96
2000
4000
6000
8000
10000
12000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
95.0
97.5
100.0
102.5
105.0
107.5
b0

Chain Plots


Diagnostics for errorvar
errorvar
400
350
300
250
200
2000
4000
6000
8000
10000
12000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
150
200
250
300
350
400
450
errorvar

Converged? If Yes…now onto the parameters

Posterior Summaries						
Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
b0	400	101.1	1.4841	100.1	101.1	102.0
errorvar	400	273.9	45.2561	238.7	268.5	302.3

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
b0	0.050	98.1372	104.1	98.2694	104.2
errorvar	0.050	204.1	384.7	200.2	359.7

$\beta_0 = 101.1;$
$\sigma_e^2 = 273.9$

Model Comparison?
Have to use DIC

Deviance Information Criterion	
Dbar (posterior mean of deviance)	417.636
Dmean (deviance evaluated at posterior mean)	416.748
pD (effective number of parameters)	0.888
DIC (smaller is better)	418.524

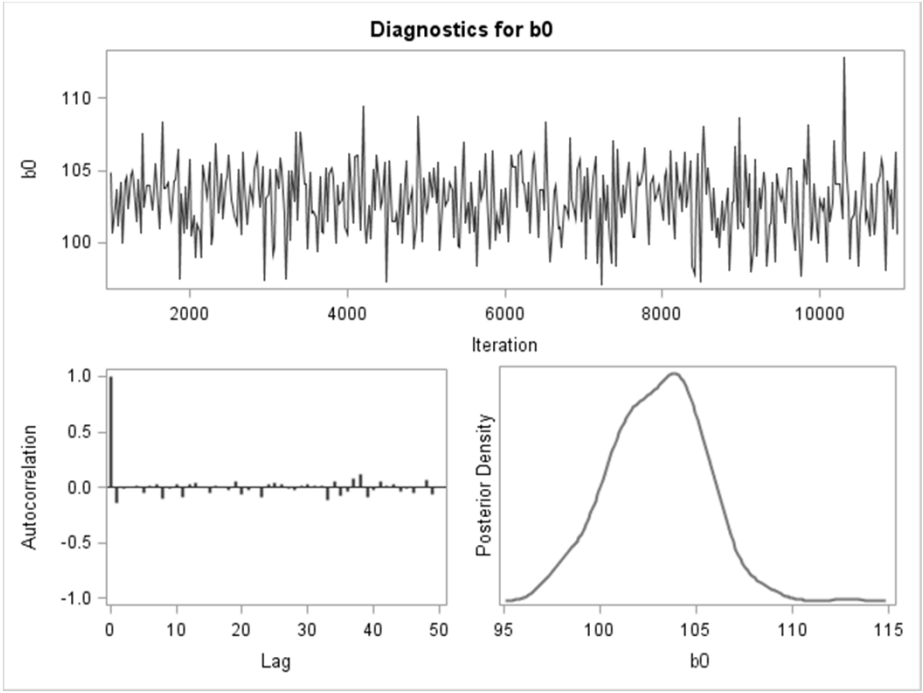
Changing Up the Prior

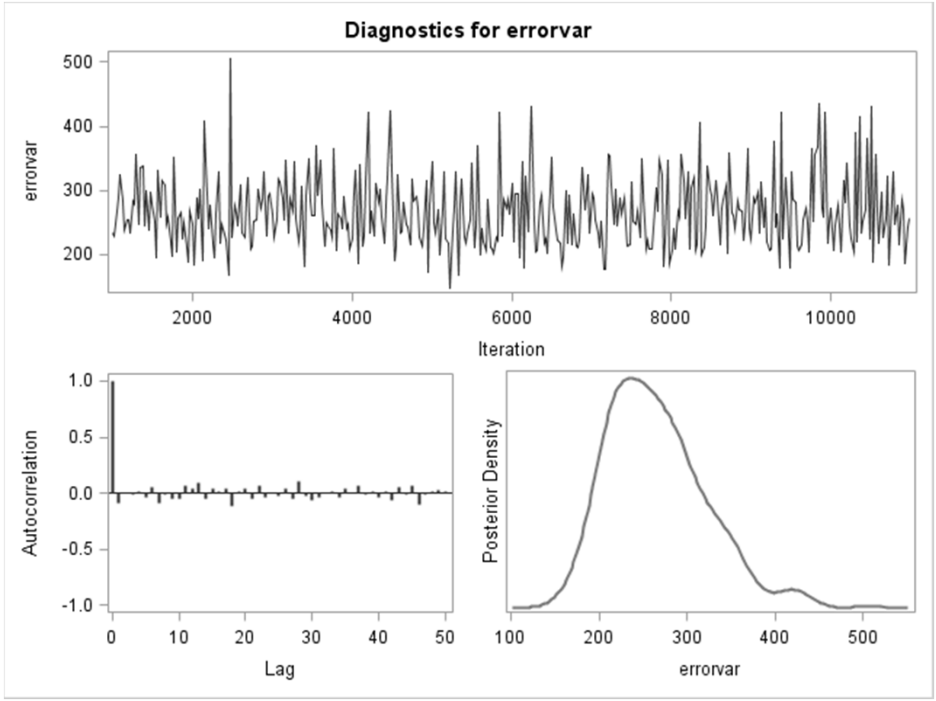
- To demonstrate how changing the prior affects the analysis, we will now try a few prior distributions for our parameters
- Prior: $\beta_0 \sim U(-10000,10000); \sigma_e^2 \sim U(0,5000)$

Posterior Summaries						
Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
b0	400	103.0	2.4016	101.3	103.1	104.6
errorvar	400	266.0	55.6614	223.5	256.8	296.5

Deviance Information Criterion	
Dbar (posterior mean of deviance)	418.004
Dmean (deviance evaluated at posterior mean)	416.093
pD (effective number of parameters)	1.911
DIC (smaller is better)	419.915

Chain Plots


Diagnostics for b0
b0
110
105
100
2000
4000
6000
8000
10000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
95
100
105
110
115
b0


Diagnostics for errorvar
errorvar
500
400
300
200
2000
4000
6000
8000
10000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
100
200
300
400
500
errorvar

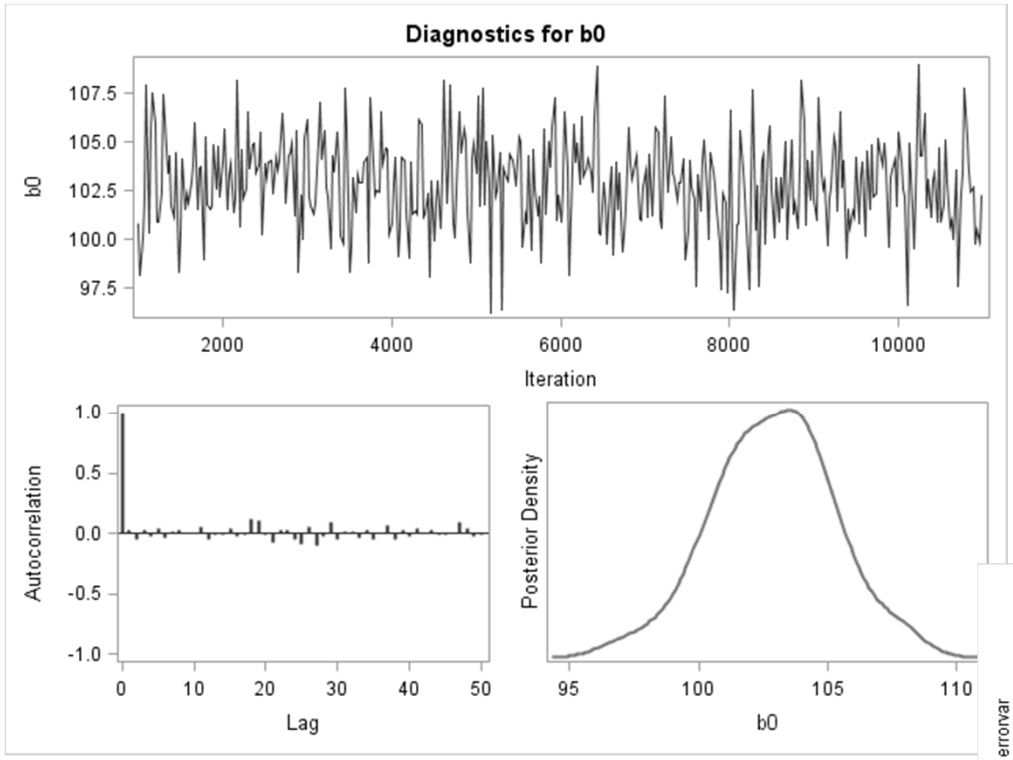
Changing Up the Prior

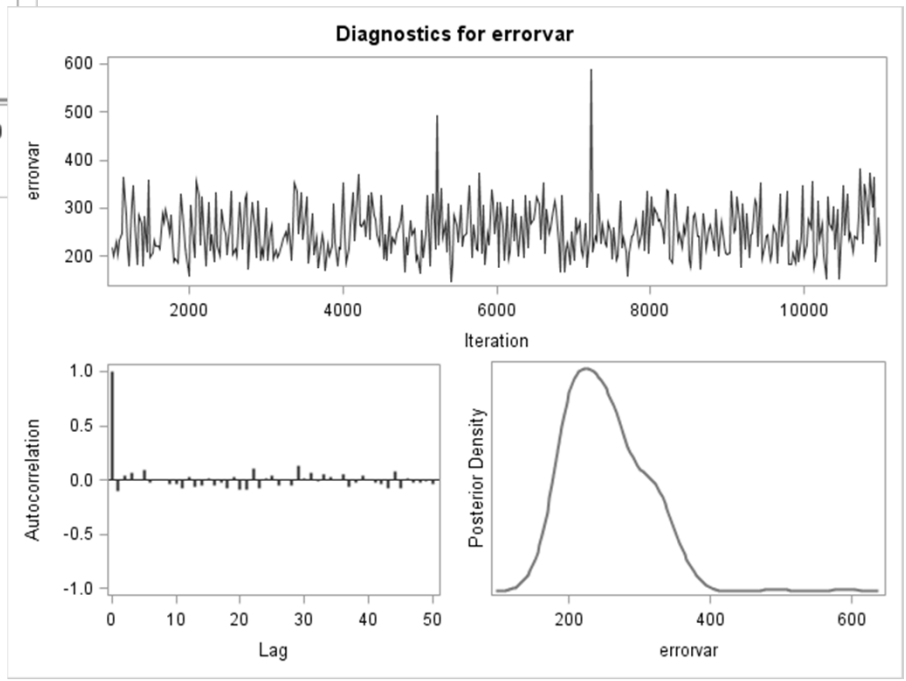
- Prior: $\beta_0 \sim N(0{,}100{,}000); \sigma_e^2 \sim \gamma^{-1}(shape = \frac{3}{10}, scale = \frac{10}{3})$

Posterior Summaries						
Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
b0	400	102.9	2.3533	101.3	102.9	104.3
errorvar	400	252.0	54.6698	210.1	244.9	286.7

Deviance Information Criterion	
Dbar (posterior mean of deviance)	417.973
Dmean (deviance evaluated at posterior mean)	415.885
pD (effective number of parameters)	2.088
DIC (smaller is better)	420.061

Chain Plots


Diagnostics for b0
b0
107.5
105.0
102.5
100.0
97.5
2000
4000
6000
8000
10000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
95
100
105
110
b0


Diagnostics for errorvar
errorvar
600
500
400
300
200
2000
4000
6000
8000
10000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
200
400
600
errorvar

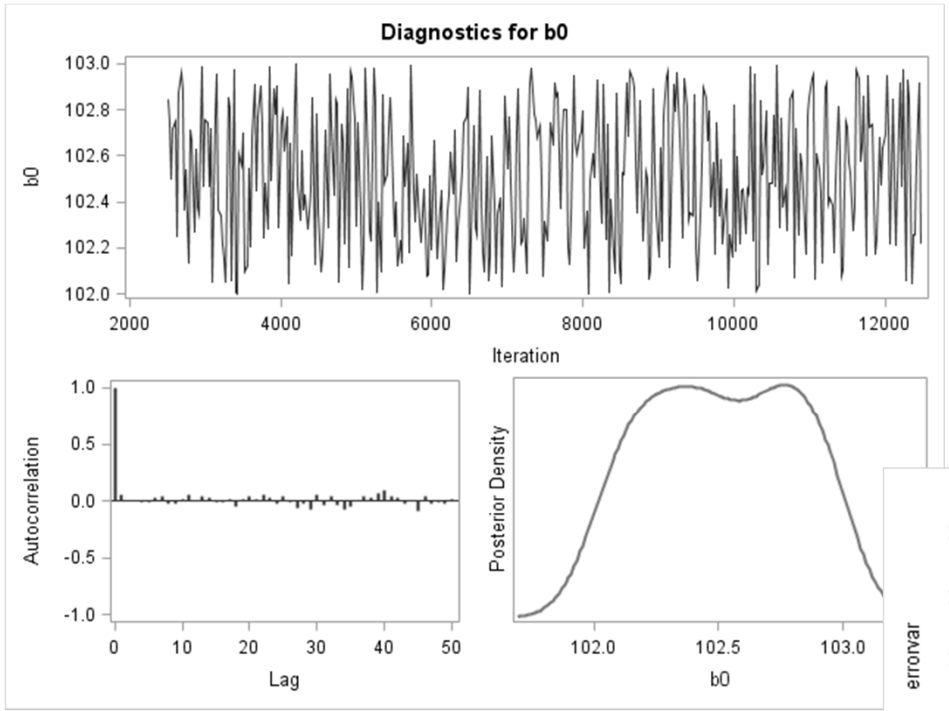
What About an Informative Prior?

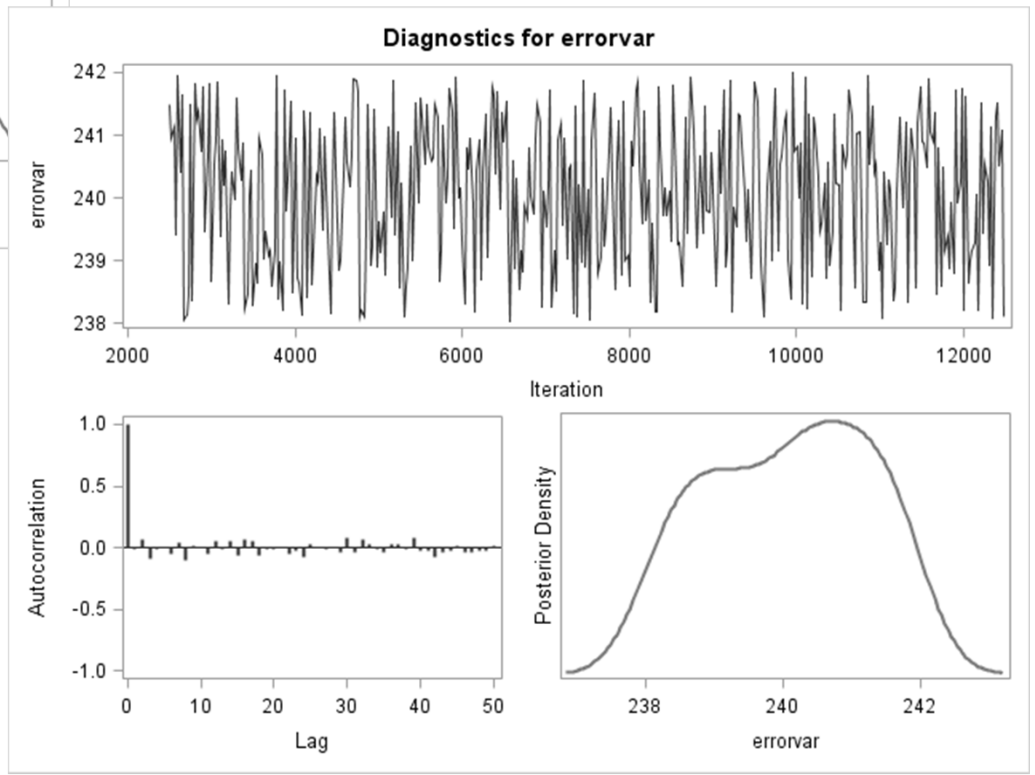
- Prior: $\beta_0 \sim U(102,103); \sigma_e^2 \sim U(238,242)$

Posterior Summaries						
Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
b0	400	102.5	0.2886	102.3	102.5	102.8
errorvar	400	240.1	1.1440	239.1	240.2	241

Deviance Information Criterion	
Dbar (posterior mean of deviance)	415.851
Dmean (deviance evaluated at posterior mean)	415.833
pD (effective number of parameters)	0.018
DIC (smaller is better)	415.869

Chain Plots


Diagnostics for b0
b0
103.0
102.8
102.6
102.4
102.2
102.0
2000
4000
6000
8000
10000
12000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
102.0
102.5
103.0
b0


Diagnostics for errorvar
errorvar
242
241
240
239
238
2000
4000
6000
8000
10000
12000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
238
240
242
errorvar

WRAPPING UP

Wrapping Up

- Today was an introduction to Bayesian statistics
  - Bayes = use of prior distributions on parameters
- We used two methods for estimation:
  - MAP estimation – far less common
  - MCMC estimation
    - Commonly, people will say Bayesian and mean MCMC – but Bayesian is just the addition of priors. MCMC is one way of estimating Bayesian models!
- MCMC is effective for most Bayesian models:
  - Model likelihood and prior likelihood are all that are needed
- MCMC is estimation by brute force:
  - Can be very slow, computationally intensive, and disk-space intensive
- But…MCMC runs multiple imputation…which is our next topic