Doing Bayesian Data Analysis
Workshop for WIM 2011

John K. Kruschke

WIM 2011 Workshop

Agenda

- Why go Bayesian.
- Bayes' Rule, Grid Approximation, & R.
- Markov Chain Monte Carlo & BUGS.
- Linear Regression.
- (Hierarchical Models, if only there were time)
Why Go Bayesian?

Oh honey I’m searching for love that is true,
But driving through fog is so dang hard to do.
Please paint me a line on the road to your heart,
I’ll rev up my pick up and get a clean start.
Why Go Bayesian?

- In null hypothesis significance testing (NHST), $p$ values are ill-defined.
  - $p$ values depend on stopping intention.
  - $p$ values depend on intended other data collection (e.g., in multiple comparisons).
- NHST provides no good indication of range of uncertainty.
  - Confidence intervals have limits determined by $p$ values, so are themselves ill-defined.
  - Confidence intervals have no distributional info.
- NHST makes power analysis “virtually unknowable”.
- NHST provides no measure of evidence in favor of the null.
- NHST suffers from sampling to a foregone conclusion.
- NHST ignores prior knowledge.

The road to NHST is paved with good intentions.

The $p$ value is the probability of the obtained sample statistic, or a result more extreme, from the null hypothesis, if the intended experiment were repeated many times.
The distribution of $t$ when the intended experiment is repeated many times

Null Hypothesis:
Groups are identical

Many simulated repetitions of the intended experiment

Fixed N=6 per group (x2 groups)

$t_{\text{crit}} = 2.23$
The intention to collect data until the end of the week

Null Hypothesis: Groups are identical

Many simulated repetitions of the intended experiment

Fixed Duration=2 weeks (x6/week)

$t_{crit} = 2.45$
An example of a $t$ test:

**Data:**

Group 1: 5.70 5.40 5.75 5.25 4.25 4.74; $M_1 = 5.18$
Group 2: 4.55 4.98 4.70 4.78 3.26 3.67; $M_2 = 4.32$

$t = 2.33$

---

**Can the null hypothesis be rejected?** To answer, we must know the intention of the data collector.
- We ask the research assistant who collected the data. The assistant says, “I just collected data for two weeks. It’s my job. I happened to get 6 subjects in each group.”
- We ask the graduate student who oversaw the assistant. The student says, “I knew we needed 6 subjects per group, so I told the assistant to run for two weeks, because we usually get about 6 subjects per week.”
- We ask the lab director, who says, “I told my graduate student to collect 6 subjects per group.”

Therefore, for the lab director, $t = 2.33$ **rejects the null hypothesis** (because $p < .05$), but for the research assistant who actually collected the data, $t = 2.33$ **fails to reject** the null hypothesis (because $p > .05$).

---

Two labs collect the same data:

**Lab A:** Collect data until $N=6$ per group.

**Data:**

Group 1: 5.70 5.40 5.75 5.25 4.25 4.74; $M_1 = 5.18$
Group 2: 4.55 4.98 4.70 4.78 3.26 3.67; $M_2 = 4.32$

$t = 2.33$

**Lab A: Reject the null.**

---

**Lab B:** Collect data for two weeks.

**Data:**

Group 1: 5.70 5.40 5.75 5.25 4.25 4.74; $M_1 = 5.18$
Group 2: 4.55 4.98 4.70 4.78 3.26 3.67; $M_2 = 4.32$

$t = 2.33$

**Lab B: Do not reject the null.**
The intention to examine data thoroughly

Many experiments involve multiple groups, and **multiple comparisons** of means.

Example: Consider 2 different drugs from chemical family A, 2 different drugs from chemical family B, and a placebo group. Lots of possible comparisons...

Problem: With every test, there is possibility of false alarm! False alarms are bad; therefore, keep the experimentwise false alarm rate down to 5%.
Multiple Corrections for Multiple Comparisons

Begin: Is goal to identify the best treatment?
Yes: Use Hsu’s method.
No: Contrasts between control group and all other groups?
   Yes: Use Dunnett’s method.
   No: Testing all pairwise and no complex comparisons (either planned or post hoc) and choosing to test only some pairwise comparisons post hoc?
      Yes: Use Tukey’s method.
      No: Are all comparisons planned?
         Yes: Use Scheffe’s method.
         No: Is Bonferroni critical value less than Scheffe critical value?
            Yes: Use Bonferroni’s method.
            No: Use Scheffe’s method (or, prior to collecting the data, reduce the number of contrasts to be tested).

Adapted from Maxwell & Delaney (2004). Designing experiments and analyzing data: A model comparison perspective. Erlbaum.

Why Go Bayesian?

☑ In null hypothesis significance testing (NHST), \( p \) values are ill-defined.
   ☑ \( p \) values depend on stopping intention.
   ☑ \( p \) values depend on intended other data collection (e.g., in multiple comparisons).
   ● NHST provides no good indication of range of uncertainty.
      • Confidence intervals have limits determined by \( p \) values, so are themselves ill-defined.
      • Confidence intervals have no distributional info.
   ● NHST makes power analysis “virtually unknowable”.
   ● NHST provides no measure of evidence in favor of the null.
   ● NHST suffers from sampling to a foregone conclusion.
   ● NHST ignores prior knowledge.
Bayes' Rule, Grid Approximation, and R

I'll love you forever in every respect
(I'll marginalize all your glaring defects)
But if you could change some to be more like me
I'd love you today unconditionally

Conjoint, marginal and conditional probability
Conjoint, marginal and conditional probability

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td>.12</td>
<td>.20</td>
<td>.01</td>
<td>.04</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.18</td>
<td>.48</td>
<td>.21</td>
<td>.12</td>
</tr>
</tbody>
</table>


Conjoint probabilities: \( p(e,h) \)

For example, \( p(e=\text{blue}, h=\text{black}) = .03 \)
### Conjunct, marginal and conditional probability

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td>.12</td>
<td>.20</td>
<td>.01</td>
<td>.04</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.18</td>
<td>.48</td>
<td>.21</td>
<td>.12</td>
</tr>
</tbody>
</table>

**Marginal probabilities: p(e)**  
For example, $p(e=\text{blue}) = .36 = \sum_h p(e=\text{blue},h)$

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td>.12</td>
<td>.20</td>
<td>.01</td>
<td>.04</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.18</td>
<td>.48</td>
<td>.21</td>
<td>.12</td>
</tr>
</tbody>
</table>

**Marginal probabilities: p(h)**  
For example, $p(h=\text{black}) = .18 = \sum_e p(e,h=\text{black})$
### Conjoint, marginal and conditional probability

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
</tr>
<tr>
<td>Brown</td>
<td>.12</td>
<td>.20</td>
<td>.01</td>
<td>.04</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
</tr>
</tbody>
</table>

**Marginal probabilities:**

\[ p(h) \text{ without info about } e \]

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
</tr>
<tr>
<td>Brown</td>
<td>.12</td>
<td>.20</td>
<td>.01</td>
<td>.04</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Hair Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>.18</td>
<td>.48</td>
<td>.21</td>
<td>.12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Conditional probabilities:**

\[ p(h \mid e=\text{blue}) \text{ is } p(h) \text{ with info that } e=\text{blue} \]
**Conjoint, marginal and conditional probability**

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Black</td>
<td>Brunette</td>
<td>Blond</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
<td>.36</td>
</tr>
<tr>
<td>Brown</td>
<td>.12</td>
<td>.24</td>
<td>.01</td>
<td>.04</td>
<td>.37</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
<td>.27</td>
</tr>
<tr>
<td></td>
<td>.18</td>
<td>.48</td>
<td>.21</td>
<td>.12</td>
<td></td>
</tr>
</tbody>
</table>

**Conditional probabilities:**

\[
p(h | e=\text{blue}) = \frac{p(h,e=\text{blue})}{p(e=\text{blue})}
\]

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Black</td>
<td>Brunette</td>
<td>Blond</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td>.03/.36 = .08</td>
<td>.14/.36 = .39</td>
<td>.16/.36 = .45</td>
<td>.03/.36 = .08</td>
<td>.36/.36 = 1</td>
</tr>
<tr>
<td>Brown</td>
<td>.12/.36 = .37</td>
<td>.24/.36 = .74</td>
<td>.01/.36 = .03</td>
<td>.04/.36 = .12</td>
<td>.37/.36 = 1</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td>.03/.36 = .09</td>
<td>.14/.36 = .45</td>
<td>.04/.36 = .12</td>
<td>.05/.36 = .15</td>
<td>.27/.36 = .77</td>
</tr>
<tr>
<td></td>
<td>.18/.36 = .53</td>
<td>.48/.36 = 1.48</td>
<td>.21/.36 = .60</td>
<td>.12/.36 = .33</td>
<td>.87/.36 = 2.47</td>
</tr>
</tbody>
</table>
Conjoint, marginal and conditional probability

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.16</td>
<td>.03</td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td>.12</td>
<td>.20</td>
<td>.01</td>
<td>.04</td>
</tr>
<tr>
<td>Hazel/Green</td>
<td></td>
<td>.03</td>
<td>.14</td>
<td>.04</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.18</td>
<td>.48</td>
<td>.21</td>
<td>.12</td>
</tr>
</tbody>
</table>

**Conditional probabilities:**

\[
p(h|e=brown) = \frac{p(h,e=brown)}{p(e=brown)}
\]

<table>
<thead>
<tr>
<th>Eye Color</th>
<th>Hair Color</th>
<th>Black</th>
<th>Brunette</th>
<th>Blond</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown</td>
<td></td>
<td>.12/.37</td>
<td>.20/.37</td>
<td>.01/.37</td>
<td>.04/.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=.32</td>
<td>=.54</td>
<td>=.03</td>
<td>=.11</td>
</tr>
</tbody>
</table>
Bayes’ rule

By definition, \( p(h \mid e) = \frac{p(e, h)}{p(e)} \) and \( p(e \mid h) = \frac{p(e, h)}{p(h)} \).

Hence \( p(h \mid e) p(e) = p(e, h) = p(e \mid h) p(h) \), and
Bayes’ rule

By definition, \( p(h \mid e) = \frac{p(e, h)}{p(e)} \) and \( p(e \mid h) = \frac{p(e, h)}{p(h)} \).

Hence \( p(h \mid e) p(e) = p(e, h) = p(e \mid h) p(h) \), and

\[
p(h \mid e) = p(e \mid h) \frac{p(h)}{p(e)}
\]
Bayes’ rule

\[ p(h \mid e) = p(e \mid h) \frac{p(h)}{p(e)} \]

So what’s the big deal? → Ramifications when applied to data and parameter values!

\[ p(\theta \mid D) = \frac{p(D \mid \theta) \cdot p(\theta)}{p(D)} \]

\[ \text{posterior} \quad \text{likelihood} \quad \text{prior} \quad \text{evidence} \]

↑ with info about \( D \)  \hspace{1cm} ↑ without info about \( D \)

Bayes’ rule applied to data and parameter values

<table>
<thead>
<tr>
<th>Data</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \theta )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( D )</td>
<td>( p(D,\theta) = p(D \mid \theta)p(\theta) )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>( p(\theta) )</td>
</tr>
</tbody>
</table>

Possible values of parameter

Possible values of data
Bayes’ rule applied to data and parameter values

<table>
<thead>
<tr>
<th>Data</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>θ</td>
</tr>
<tr>
<td>D</td>
<td>p(D,θ)=p(D</td>
</tr>
<tr>
<td>...</td>
<td>p(θ)</td>
</tr>
</tbody>
</table>

Marginal probabilities: 
$p(θ)$ without info about $D$  
a.k.a. “prior”

Bayes’ rule applied to data and parameter values

<table>
<thead>
<tr>
<th>Data</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>θ</td>
</tr>
<tr>
<td>D</td>
<td>p(D,θ)=p(D</td>
</tr>
<tr>
<td>...</td>
<td>p(θ)</td>
</tr>
</tbody>
</table>

Conditional probabilities: 
$p(θ|D)$ is $p(θ)$ with info about $D$  
a.k.a. “posterior”
Example: Estimating bias of a coin

**Possible data values:**
D=1 for “heads” and D=0 for “tails”

**Likelihood function:**
p(D=1|\(\theta\)) = \(\theta\) and p(D=0|\(\theta\)) = 1-\(\theta\)

hence p(D|\(\theta\)) = \(\theta^n(1-\theta)^{(1-D)}\)

**Prior on \(\theta\):**
\(\theta\) can be any value in interval [0,1], but for illustration we’ll restrict \(\theta\) to three possible values with these prior probabilities:
p(\(\theta = .25\)) = .25
p(\(\theta = .50\)) = .50
p(\(\theta = .75\)) = .25

Outcomes are nominal values, but re-coded as 1 and 0 for convenience in likelihood function.
Example: Estimating bias of a coin

Possible data values:
D=1 for "heads" and D=0 for "tails"

Likelihood function:
\[ p(D=1|\theta) = \theta \quad \text{and} \quad p(D=0|\theta) = 1-\theta \]
hence \[ p(D|\theta) = \theta^D(1-\theta)^{(1-D)} \]

Prior on \( \theta \):
\( \theta \) can be any value in interval [0,1],
but for illustration we’ll restrict \( \theta \) to
three possible values with these prior probabilities:
\[ p(\theta = .25) = .25 \]
\[ p(\theta = .50) = .50 \]
\[ p(\theta = .75) = .25 \]

Probability of data value is a function of the parameter \( \theta \).
D is an observed constant.
Example: Estimating bias of a coin

Prior on \( \theta \):
\( \theta \) can be any value in interval \([0,1]\), but for illustration we’ll restrict \( \theta \) to three possible values with these prior probabilities:

\[
\begin{align*}
    p(\theta = .25) &= .25 \\
    p(\theta = .50) &= .50 \\
    p(\theta = .75) &= .25
\end{align*}
\]

Flip the coin once and get “heads”: Data contain 1 Heads and 0 Tails.
Example: Estimating bias of a coin

Posterior at $\theta$ is prior, $p(\theta)$, times likelihood, $p(D|\theta)$, normalized by $p(D) = \sum_\theta p(D|\theta)p(\theta)$.

Example: Estimating bias of a coin

Posterior is a compromise between prior and likelihood. Posterior shows reallocation of beliefs across candidate parameter values.
Example: Estimating bias of a coin

Prior has denser comb of parameter values.

More flips: 3 Heads and 9 Tails.

Posterior shows reallocation of beliefs across candidate parameter values.

Doing it in R

- You previously installed R, and copied program files, as instructed on the workshop website http://www.indiana.edu/~jkkteach/WorkshopWIM2010.html
- Invoke R from within Windows or WINE.
  In command window menu: Edit → Gui Preferences... → SDI (restart R if nec.)
- Help in R:
  help("keyword")
  If html help isn’t working, try
  help("keyword", h="text")
Doing it in R

• Change directory to the folder in which you have your R programs saved:
  File → Change Dir...

• Open the program **SimpleGraph.R**:
  File → Open Script...

```
x = seq( from = -2 , to = 2 , by = 0.1 )   # Specify vector of x values.
y = x^2                                    # Specify corresponding y values.
plot( x , y , type = "l" )                 # Make a graph of the x,y points.
dev.copy2eps( file = "SimpleGraph.eps" )   # Save the plot to an EPS file.
```
File → Open Script... BayesUpdate.R

```r
# Theta is the vector of candidate values for the parameter theta.
# nThetaVals is the number of candidate theta values.
# To produce the examples in the book, set nThetaVals to either 3 or 63.
# nThetaVals = 3
# Now make the vector of theta values:
Theta = seq( from = 1/(nThetaVals+1) , to = nThetaVals/(nThetaVals+1) ,
            by = 1/(nThetaVals+1) )

# pTheta is the vector of prior probabilities on the theta values.
# theta is a vector of values.
pTheta = pmin( Theta , 1-Theta ) # Makes a triangular belief distribution.
pTheta = pTheta / sum( pTheta ) # Makes sure that beliefs sum to 1.

# Specify the data in the book, use either
# Data = c(1,1,0,0,0,0,0,0,0,0) or Data = c(1,0,0,0,0,0,0,0,0,0).
# Data = c(1,1,0,0,0,0,0,0,0,0)
nHeads = sum( Data == 1 )
nTails = sum( Data == 0 )

# Compute the likelihood of the data for each value of theta:
pDataGivenTheta = Theta^nHeads * (1-Theta)^nTails

# Compute the posterior:
pData = sum( pDataGivenTheta * pTheta )
p ThetaGivenData = pDataGivenTheta * pTheta / pData # This is Bayes' rule!

# Plot the results.
```

J. K. Knobsch, WIM 2011 Workshop
# Theta is the vector of candidate values for the parameter theta.
# nThetaVals is the number of candidate theta values.
# To produce the examples in the book, set nThetaVals to either 3 or 63.
# Now make the vector of theta values:
Theta = seq( from = 1/(nThetaVals+1) , to = nThetaVals/(nThetaVals+1) ,
by = 1/(nThetaVals+1) )

# pTheta is the vector of prior probabilities on the theta values.
pTheta = pmin( Theta , 1-Theta ) # Makes a triangular belief distribution.
pTheta = pTheta / sum( pTheta ) # Makes sure that beliefs sum to 1.

# Specify the data. To produce the examples in the book, use either
# Data = c(1,1,1,0,0,0,0,0,0,0,0,0) or Data = c(1,0,0,0,0,0,0,0,0,0,0,0).
Data = c(1,1,1,0,0,0,0,0,0,0,0,0)
nHeads = sum( Data == 1 )

# Compute the likelihood of the data for each value of theta:
pDataGivenTheta = Theta^nHeads * (1-Theta)^nTails

# Compute the posterior:
pData = sum( pDataGivenTheta * pTheta )
pThetaGivenData = pDataGivenTheta * pTheta / pData # This is Bayes' rule!

# Plot the results.

Recap

- Conjoint, marginal, and conditional probabilities.
- Bayes’ rule: Re-allocating beliefs given data.
- Bayes’ rule applied to estimating bias of a coin (using grid approximation).
- Interactive introduction to R.
Markov Chain Monte Carlo and BUGS

You furtive posterior: coy distribution. Alluring, curvaceous, evading solution. Although I can see what you hint at is ample, I’ll settle for one representative sample.

Why Bayesian inference can be difficult

\[ p(\theta | D) = \frac{p(D | \theta) p(\theta)}{p(D)} \]
\[ = \frac{p(D | \theta) p(\theta)}{\sum_{\theta_c} p(D | \theta_c) p(\theta_c)} \text{ for discrete } \theta \]
\[ = \frac{p(D | \theta) p(\theta)}{\int d\theta \, p(D | \theta) p(\theta)} \text{ for continuous } \theta \]

Complicated mathematical forms... ...yield an intractable integral. Moreover, grid approximation will not work for models with many parameters.
Approximating the posterior by a random sample

Problem:
Analytical mathematical derivation is intractable, and grid approximation is intractable.

Solution:
- Approximate the posterior by a large random sample of representative points.
- This approach works because we can generate random values from the posterior without solving the integral for $p(D)$.

Approximating a distribution by a random sample

![Graphs showing approximation of a distribution with different sample sizes](image.png)
Approximating a distribution by a random sample

Any aspect of the underlying continuous distribution can be approximated from the sample: Mean, variance, percentiles, etc.

How to generate a random sample

Some distributions have known, efficient random-sample generators:
Uniform, normal, exponential, beta, gamma, etc.

Use the known random sample generators to indirectly sample from a complicated posterior distribution. There are several ways, of which we’ll consider two:
• Gibbs Sampling
• Metropolis Algorithm
Consider this posterior distribution on two parameters, $\theta_1$ and $\theta_2$. Goal is to generate a random sample without computing the integral $p(D)$.

**Gibbs Sampling**

1. Derive conditional distribution for each parameter: $p(\theta_i | D, \{ \theta_{j \neq i} \})$
2. Sample directly from each conditional distribution.
Gibbs Sampling

1. Derive conditional distribution for each parameter: \( p(\theta_i | D, \{\theta_j \neq i\}) \)
2. Sample directly from each conditional distribution.

Requires mathematical savvy.

Requires that the conditional distributions can be "directly" sampled.

Metropolis Algorithm

1. Propose a new parameter position, \( \theta_{\text{pro}} \), directly sampled from a known distribution such as a normal.
2. Decide whether to accept the proposed position. Decision rule only needs to evaluate \( p(D | \theta_{\text{pro}}) \) and \( p(\theta_{\text{pro}}) \), not \( p(D) \).
### Metropolis Algorithm

1. Propose a new parameter position, $\theta_{\text{pro}}$, directly sampled from a known distribution such as a normal.
2. Decide whether to accept the proposed position. Decision rule only needs to evaluate $p(D|\theta_{\text{pro}})$ and $p(\theta_{\text{pro}})$, not $p(D)$.

If the proposal distribution does not resemble the posterior distribution (e.g., proposal is too wide or too narrow), then many proposals are rejected and the sample gets very “clumpy”.

### Gibbs and Metropolis: Both “MCMC” Sampling

Both are “Markov chain Monte Carlo” (MCMC) methods:

- From current point (Markov chain), randomly generate a new point (Monte Carlo).

- Metropolis sampling suffers from rejected proposals and clumpiness.
- Gibbs sampling requires that conditional distributions can be derived and directly sampled, but it never has rejected proposals.
- Gibbs sampling can be inefficient and clumpy if parameters are strongly correlated in posterior: Hits a “wall” in every direction. But cleverly tuned Metropolis sampling can propose in correlated directions.
Doing it with BUGS

“BUGS” = Bayesian inference Using Gibbs Sampling but other sampling methods are incorporated.

R programming language  BRugs commands  BUGS executables

BUGS makes it easy. You specify only the
• prior function
• likelihood function
• starting position of the chains
and BUGS does the rest! You do no math, no
selection of sampling methods.

An example: Estimating the parameters
of a normal distribution

Normal distribution has
two parameters:
\( \mu \) and \( \sigma \).

Various values of \( \mu \) and \( \sigma \)
are reasonably consistent
with the three data
points shown at left.

Given \( D \), what should we believe about
\( \mu \) and \( \sigma \)?
An example: Estimating the parameters of a normal distribution

Given D, what should we believe about μ and σ?

Bayesian answer:

\[ p(\mu, \sigma \mid D) = \frac{p(D \mid \mu, \sigma)p(\mu, \sigma)}{p(D)} \]

**Posterior**

**Prior**

**Normal likelihood function**

*Bayesian inference is re-allocation of beliefs. Any estimated parameter has beliefs re-allocated from prior to posterior.*

---

Likelihood function:
The data, \( y_i \), come from a normal distribution with parameters \( \mu \) and \( \tau (\tau=1/\sigma^2) \).

Figure 15.2: A model of dependencies for several metric scores drawn from a single group. The normal distributions are parameterized by mean and precision (not standard deviation).
An example: Estimating the parameters of a normal distribution

Prior:
Without the data, we believe $\mu$ comes from a normal distribution with parameters $M$ and $T$ ($T=1/SD^2$).

Figure 15.2: A model of dependencies for several metric scores drawn from a single group. The normal distributions are parameterized by mean and precision (not standard deviation).
Digression: The gamma distribution
(to express prior belief about precision)

Figure 9.8: Examples of the gamma density distribution. The gamma(\(x, r\)) distribution is a probability density for \(x \geq 0\), given by gamma(\(x, r\)) = \(\frac{\Gamma(x)}{\Gamma(r)} x^{-r-1} e^{-\frac{x}{r}}\), where \(\Gamma(x)\) is the gamma function: \(\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt\).

The gamma function is a generalization of the factorial, because for positive integers, \(\Gamma(x) = (x-1)!\). In the specification of the distribution, \(x\) is called the “shape” parameter and \(r\) is called the “rate” (or “inverse scale”) parameter. The mean of the gamma distribution is \(m = x/r\), and the standard deviation of the gamma distribution is \(sd = \sqrt{3r}\). Hence \(v = m^2/sd^2\) and \(r = m/sd^2\).

In R, the gamma density is provided by dgamma(x, shape, rate=1), and the gamma function is provided by gamma(s). Conveniently, BUGS parameterizes the gamma distribution the same way as R, i.e., with shape and rate parameters in that order.

An example: Estimating the parameters of a normal distribution

Prior:
In general, it’s a conjoint prior on \(\mu\) and \(\tau\), which here happens to be specified as independent beliefs.

Figure 15.2: A model of dependencies for several metric scores drawn from a single group. The normal distributions are parameterized by mean and precision (not standard deviation).
An example: Estimating the parameters of a normal distribution

Goal: Have BUGS provide posterior estimate of \( \mu \) and \( \tau \).

Figure 15.2: A model of dependencies for several metric scores drawn from a single group. The normal distributions are parameterized by mean and precision (not standard deviation).

An example: Estimating the parameters of a normal distribution

In BUGS you specify only the
- prior function
- likelihood function
- starting position of the chains
and BUGS does the rest!
An example: Estimating the parameters of a normal distribution

BUGS language (not R!):

```plaintext
model {
  # Likelihood:
  for (i in 1 : N) {
    y[i] ~ dnorm( mu , tau )
  }
  # Prior:
  tau ~ dgamma( 0.01 , 0.01 )
  mu ~ dnorm( 0 , 1.0E-10 )
}
```

An example: Estimating the parameters of a normal distribution

BUGS language (not R!):

```plaintext
model {
  # Likelihood:
  for (i in 1 : N) {
    y[i] ~ dnorm( mu , tau )
  }
  # Prior:
  tau ~ dgamma( 0.01 , 0.01 )
  mu ~ dnorm( 0 , 1.0E-10 )
}
```
An example: Estimating the parameters of a normal distribution

- You previously installed BUGS, and copied program files, as instructed on the tutorial website http://www.indiana.edu/~jkkteach/WorkshopWIM2011.html
- Invoke R from within Windows or WINE.
- Change directory to the folder in which you have your R programs saved:
  
  File → Change Dir...

- Open the program **YmetricXsingleBrugs.R**:
  
  File → Open Script...

  You will also need programs **plotChains.R**, **plotPost.R** and **HDIofMCMC.R** in the same folder.

```r
graphics.off()
rm(list=ls(all=TRUE))
#----------------------------------------------
# THE MODEL.
modelstring = "
# BUGS model specification begins here...
model {
    # Likelihood:
    for ( i in 1 : N ) {
        y[i] ~ dnorm( mu , tau ) # tau is precision, not SD
    }
    # Prior:
    tau ~ dgamma( 0.01 , 0.01 )
    mu ~ dnorm( 0 , 1.0E-10 )
}
# ... end BUGS model specification
" # close quote for modelstring
writeLines(modelstring,con="model.txt")
modelCheck( "model.txt" )
#----------------------------------------------
```

J. K. Kruschke, WIM 2011 Workshop
# An example: Estimating the parameters of a normal distribution

# THE DATA.

# Generate random data from known parameter values:
set.seed(47405)
trueM = 100
trueSD = 15
y = round(rnorm(n=500, mean=trueM, sd=trueSD))  # R dnorm uses mean and SD
dataList = list(
    y = y,
    N = length(y)
)

# Get the data into BRugs: (default filename is data.txt).
modelData(bugsData(datalist))

# INITIALIZE THE CHAINS.

nchain = 3
modelCompile(numChains = nchain)

automaticInit = F  # TRUE or FALSE
if ( automaticInit ) {
    modelGenInits()  # automatically initialize chains from prior
} else {
    genInitList <- function() {  # manually initialize chains near the data
        list( mu = mean(datalist$y),
              tau = 1 / sd(datalist$y)^2 )
    }
    for ( chainIdx in 1 : nchain ) {
        modelInits(bugsInits( genInitList ) )
    }
}

---

J. K. Kruschke, WIM 2011 Workshop

79

---

J. K. Kruschke, WIM 2011 Workshop

80
An example: Estimating the parameters of a normal distribution

#---------------------------------
# RUN THE CHAINS

# burn in
BurnInSteps = 500
modelUpdate( BurnInSteps )
# actual samples
samplesSet( c( "mu" , "tau" ) )
stepsPerChain = 2000
thinStep = 1
modelUpdate( stepsPerChain , thin=thinStep )

#---------------------------------

An example: Estimating the parameters of a normal distribution

#---------------------------------
# EXAMINE THE RESULTS

filenamert = "YmetricXsingleBrugs"

source("plotChains.R")
muSum = plotChains("mu", saveplots=F , filenamert )
sigmaSum = plotChains( "tau", saveplots=F , filenamert )

muSample = samplesSample( "mu" )
tauSample = samplesSample( "tau" )
sigmaSample <- 1 / sqrt( tauSample ) # Convert precision to SD

source("plotPost.R")
windows()
plotPost( muSample , xlab="mu" , breaks=30 , main="Posterior" )
dev.copy2eps(file=paste(filenamert,"PostMu.eps",sep=""))

nPts = length(muSample) ; nPtsForDisplay = min( nPts , 2000 )
thinIdx = seq( 1 , nPts , nPts / nPtsForDisplay )
windows()
plot( muSample[thinIdx] , sigmaSample[thinIdx] , col="gray" ,
    xlab="mu" , ylab="sigma" , cex.lab=1.5 , main="Posterior" , log="y" )
points( mean(muSample) , mean(sigmaSample) , pch="*" , cex=2 )

text( mean(muSample) , mean(sigmaSample) ,
    bquote( .(round(mean(muSample),1)) *"* .(round(mean(sigmaSample),1)) ) ,
    adj=c(.5,-0.5) )
dev.copy2eps(file=paste(filenamert,"PostMuSigma.eps",sep=""))

#---------------------------------
An example: Estimating the parameters of a normal distribution

Result is *conjoint* distribution over all the parameters.

Marginal distribution of parameter, with summary values (shown only to 3 significant digits).
Highest Density Interval (HDI)

Points within the HDI have higher believability (probability density) than points outside the HDI.

The total probability of points within the 95% HDI is 95%.

Points outside the HDI may be deemed unbelievable or not credible.

An example: Estimating the parameters of a normal distribution

Marginal distribution of parameter, with summary values (shown only to 3 significant digits).

Values outside HDI may be deemed “not credible”.
Recap

- MCMC sampling from posterior distribution.
- Interactive introduction to BUGS.
- Applied to estimating the parameters of a normal distribution.

Linear Regression

*The agri-bank’s threaten’ to revoke my lease
If my field’s production don’t rapid increase.
Oh Lord how I wish I could divine the trend,
Will my furrows deepen and will my line end?*
For every individual, there is a predictor variable, \( x \), and a predicted variable, \( y \).

Linear regression models \( y \) as a random distribution around the line

\[
y_{\text{pred}} = \beta_0 + \beta_1 x
\]

Example of data generated from the model.

Three parameters:
- intercept is 10,
- slope is 2,
- SD is 2.

Vertical scatter is normally distributed, with same SD for all \( x \).
Linear Regression

Another example of data generated from the model.

Model does not describe distribution of \( x \), instead merely dependence of \( y \) on \( x \).

Vertical scatter is still normally distributed with same SD for all \( x \).

Bayesian Linear Regression

Given the data \( D=\{<x_i,y_i>\} \), what are credible estimates of slope, intercept, and standard deviation?

Bayesian answer: Re-allocate credibility from prior.
Bayesian Linear Regression

Data consist of $x_i$ and $y_i$ for each individual

Likelihood function: Data $y_i$ are normally distributed around $\beta_0 + \beta_1 x_i$
Bayesian Linear Regression

Prior on intercept: without the data, $\beta_0$ is normally distributed with parameters $M_0, T_0$

Prior on slope: without the data, $\beta_1$ is normally distributed with parameters $M_1, T_1$
Bayesian Linear Regression

Prior on precision (1/SD²): without the data, τ is gamma distributed with parameters S,R

Three parameters with a conjoint prior, specified here as independent beliefs for convenience.
Bayesian Linear Regression in BUGS

model {
  for (i in 1 : Ndata) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + beta1 * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  beta1 ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}

J. K. Knusckie, WIM 2011 Workshop
Bayesian Linear Regression in BUGS

\[
\begin{align*}
\beta_0 + \beta_1 x_i &= \mu_i \\
\mu_i &\sim \text{normal}( \beta_0 + \beta_1 x_i, \tau ) \\
\beta_0 &\sim \text{dnorm}( 0, 1.0 \times 10^{-12} ) \\
\beta_1 &\sim \text{dnorm}( 0, 1.0 \times 10^{-12} ) \\
\tau &\sim \text{dgamma}( 0.001, 0.001 )
\end{align*}
\]

BUGS is not R. The assignment operator in BUGS is `<-` but R allows either `<-` or `=`.

```
model {
  for( i in 1:Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + beta1 * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  beta1 ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}
```

J. K. Knuschie, WIM 2011 Workshop

Bayesian Linear Regression in BUGS

```
model {
  for( i in 1:Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + beta1 * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  beta1 ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}
```

J. K. Knuschie, WIM 2011 Workshop
Bayesian Linear Regression in BUGS

model {
  for ( i in 1 : Ndata ) {
    y[i] ~ dnorm( mu[i] , tau )
    mu[i] <- beta0 + beta1 * x[i]
  }
  beta0 ~ dnorm( 0 , 1.0E-12 )
  beta1 ~ dnorm( 0 , 1.0E-12 )
  tau ~ dgamma( 0.001 , 0.001 )
}

J. K. Kruschke, WIM 2011 Workshop
Program for Linear Regression

- You previously installed BUGS, and copied program files, as instructed on the tutorial website http://www.indiana.edu/~jkkteach/WorkshopWIM2011.html
- Invoke R from within Windows or WINE.
- Change directory to the folder in which you have your R programs saved:
  - File → Change Dir...

- Open the program
  **SimpleLinearRegressionBrugs.R:**
  - File → Open Script...

You will also need the file
**HtWt30.csv** in the same folder

and **plotChains.R, plotPost.R** and **HDIofMCMC.R**
in the same folder.

---

Interactive discussion of programming details, including:

- Random data generator and **data structure**.
- Standardizing the data for MCMC efficiency.
- Initializing the chains “intelligently”.
- Transforming parameters from standardized scales to original scales (point to book).
- **Posterior prediction:** Random generation of simulated data using credible parameter values.
Linear Regression: Posterior Distribution

Posterior distribution is conjoint over *three* parameters, only two of which are shown here.

Posterior slopes and intercepts

Each line also has a corresponding SD (precision), not shown.
A slope of zero may be deemed not credible.

Vertical bars summarize simulated data as predicted from credible parameter values.
Posterior Predictions

Predictions are more uncertain for extrapolated x than for interpolated x

Recap

- Bayesian linear regression
  - Doing it with BUGS: Standardizing data
  - Correlated parameters in conjoint posterior
  - Posterior predictions show bowed extrapolation
Hierarchical Models

*Oh darlin’, for love, it’s on you I depend.*
Well, I s’pose Jack Daniels is also my friend.
*But you keep him locked up in the ladies’ loo.*
*S’pose that means his spirits depend on you too.*

Recap

- Hierarchical models applied to repeated measures
  - Simultaneous estimation of group and individual parameters
  - Different N observations for each individual
  - Doing it with BUGS: Burn-in and thinning
  - Posterior uncertainty affected by amount of data
  - “Shrinkage” of individual estimates: Mutually informed through hyperprior.
Conclusion

- Bayesian analysis overcomes many problems of 20th century NHST:
  - No $p$ values. No “corrections” for multiple comparisons. No pretending to fix sample size in advance. No pretending not to analyze data in advance. Etc. etc. etc!

- Bayesian data analysis provides flexible and rich inferences.
  - Direct interpretation of posterior. Conjoint posterior shows parameter correlations. Rational shrinkage of estimates. Etc. etc. etc!

- Straightforward to implement in R and BUGS.

- *It’s time to go Bayesian!*

J. K. Kruschke, WIM 2011 Workshop