

Intro to Bayesian Analysis

Michael Hansen



Topics

- ▶ Part 1
 - ▶ Probability from a Bayesian Perspective
 - ▶ Differences with NHST
 - ▶ Theory of Bayesian Statistics
- ▶ Part 2
 - ▶ Gibbs sampling, MCMC
 - ▶ Examples – R and BUGS

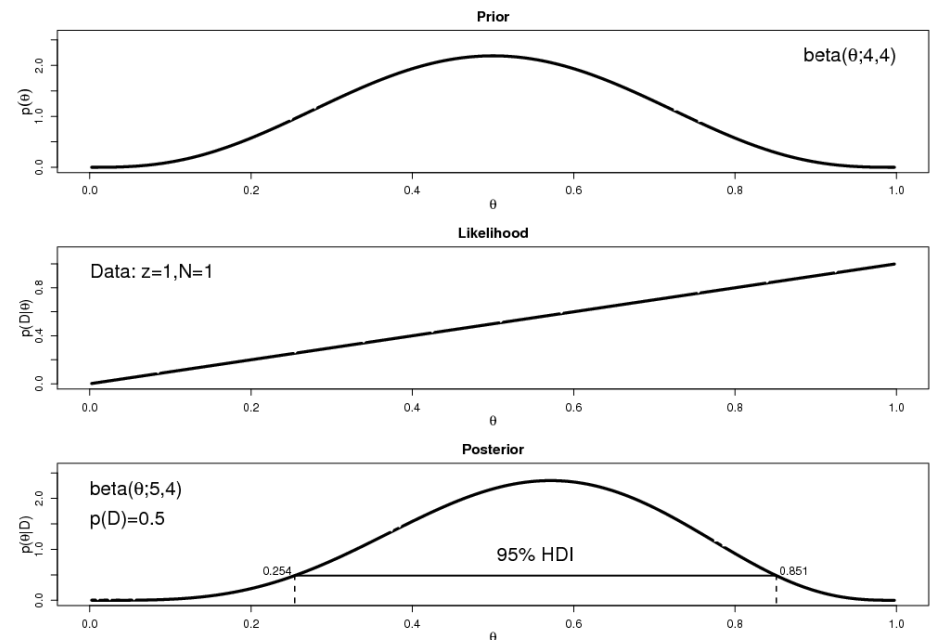
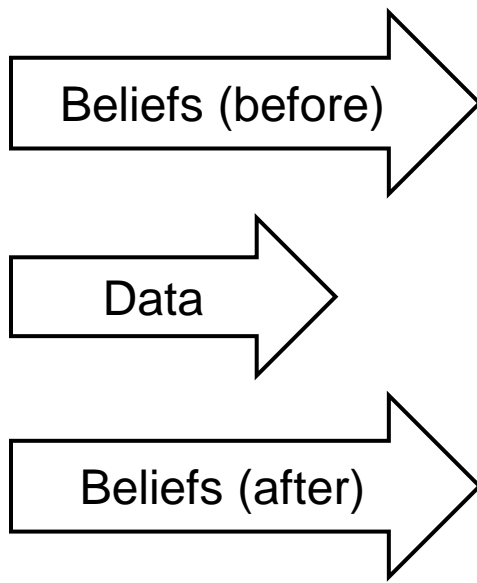


Part 1 - The Theory



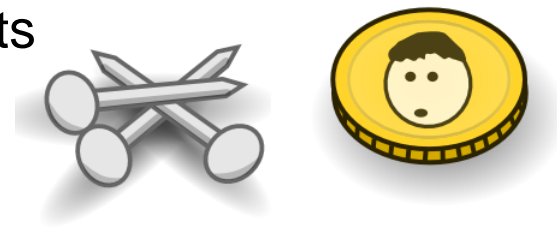
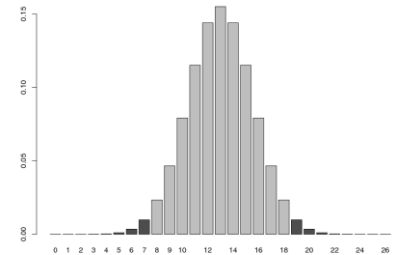
Probability

- ▶ Measure of uncertainty
- ▶ Subjective beliefs or long-run relative frequencies?
 - ▶ Math is the same – probability distributions over sample space
- ▶ Bayesian analysis transforms beliefs



Null Hypothesis Significance Testing

- ▶ Goal of inference: determine significance of a parameter value
 - ▶ Calculate p value, reject if $p < 0.05$ (5% chance of false alarm)
 - ▶ Problem – p value calculation depends on experimenter intentions
 - ▶ Was sample size set by design or by chance?
 - ▶ Edge cases exist where this can change significance
- ▶ Confidence Intervals
 - ▶ Range where a parameter would be significant
 - ▶ Still depends on p value, so...
 - ▶ No information about degree of uncertainty in a parameter value
- ▶ Prior knowledge
 - ▶ Difficult or impossible to use in standard tests



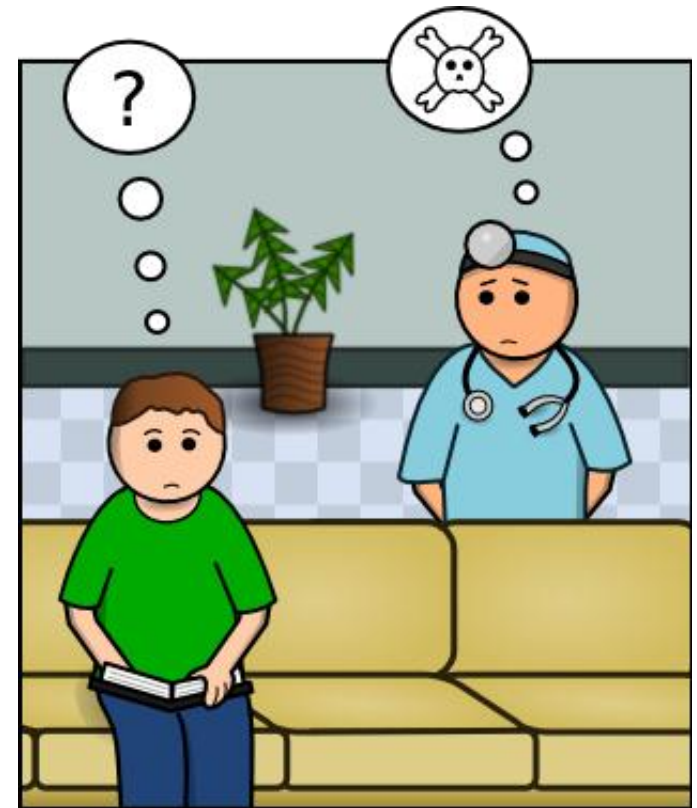
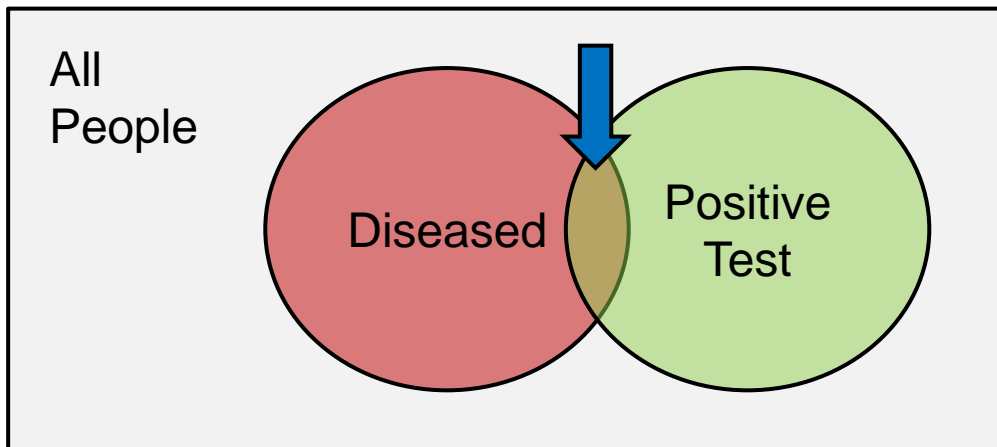
Advantages of Bayesian Analysis

- ▶ Prior knowledge is naturally represented
 - ▶ Prior must be acceptable to a skeptical scientific audience
 - ▶ Easy to swap out priors and re-run the analysis
- ▶ Uncertainty is front-and-center
 - ▶ Parameters have degrees of certainty rather than significant or not significant
 - ▶ Judgment is left to the user (decision theory, HDI, ROPE)
- ▶ Model comparison is simple
 - ▶ Highly complex models are naturally penalized (diffuse posterior)
- ▶ No corrections needed for multiple comparisons
 - ▶ Multi-dimensional posterior can be freely sliced



Deriving Bayes' Rule (1/2)

- ▶ A blood test for a disease has a 1% false alarm rate
- ▶ Typically, 1 in 100,000 have the disease
- ▶ You receive a positive test
 - ▶ Do you have the disease?



Deriving Bayes' Rule (2/2)

▶ $P(\text{disease}|\text{test}) = P(\text{test}|\text{disease}) * P(\text{disease})$

normalized

▶ Do you have the disease?

▶ $(0.99 \times 0.00001) / (0.99 \times 0.00001 + 0.01 \times 0.9999) = 0.00099$

▶ Less than 1 in 1,000 chance

▶ What if you re-did the test?

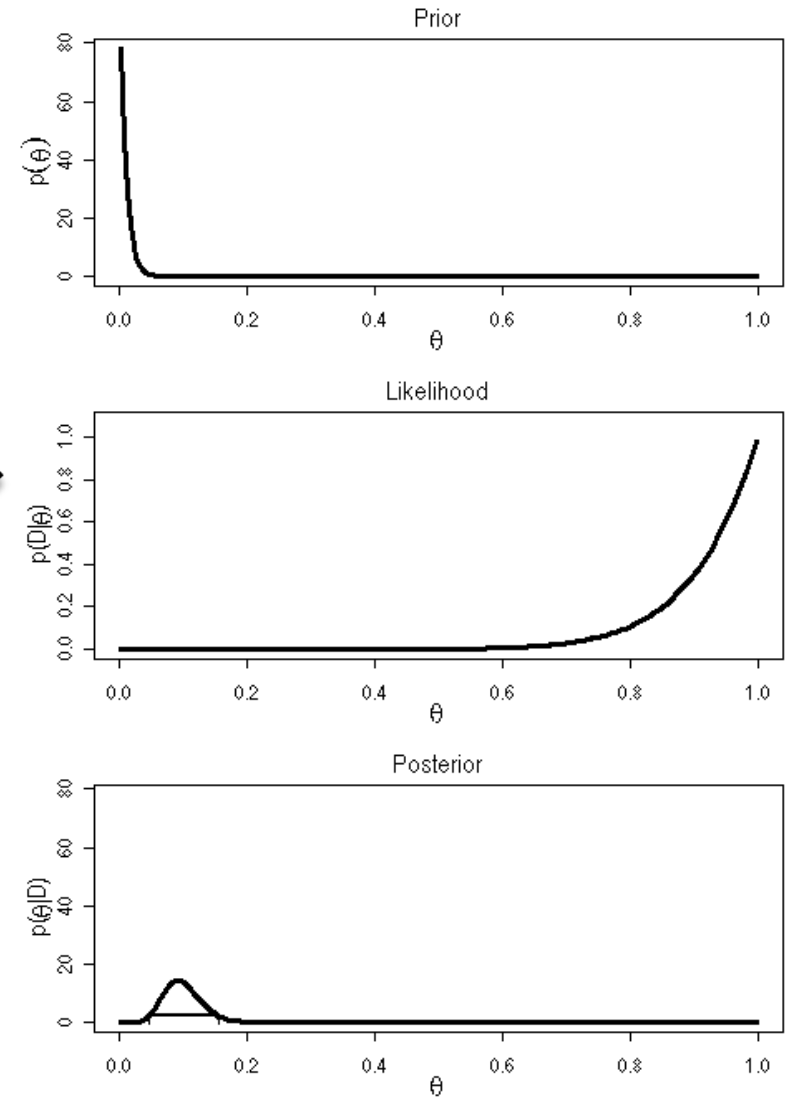
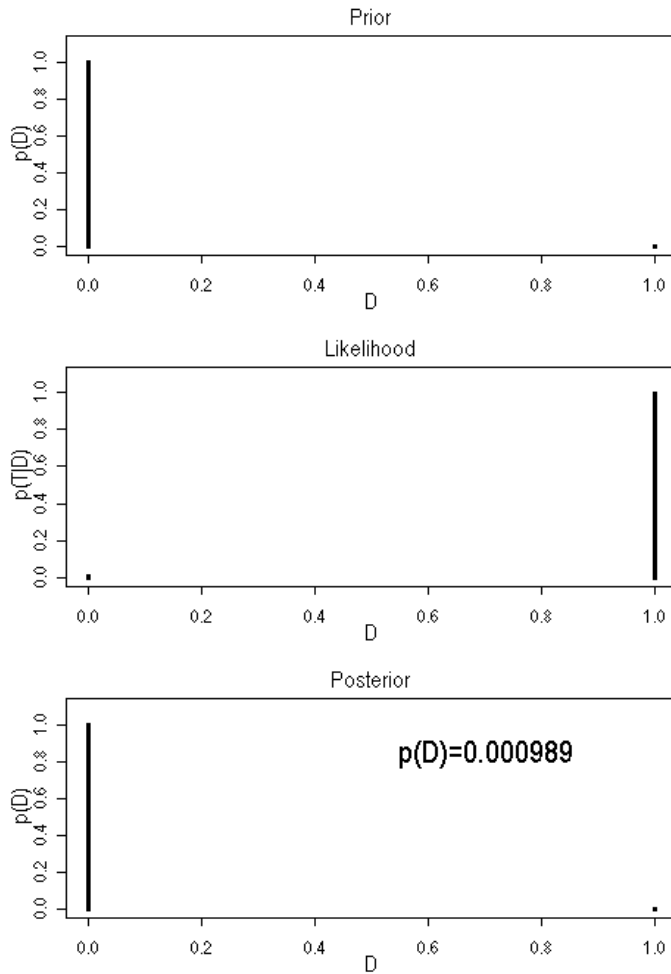
▶ Positive result – Less than 1 in 10

▶ Negative result – Less than 1 in 100,000

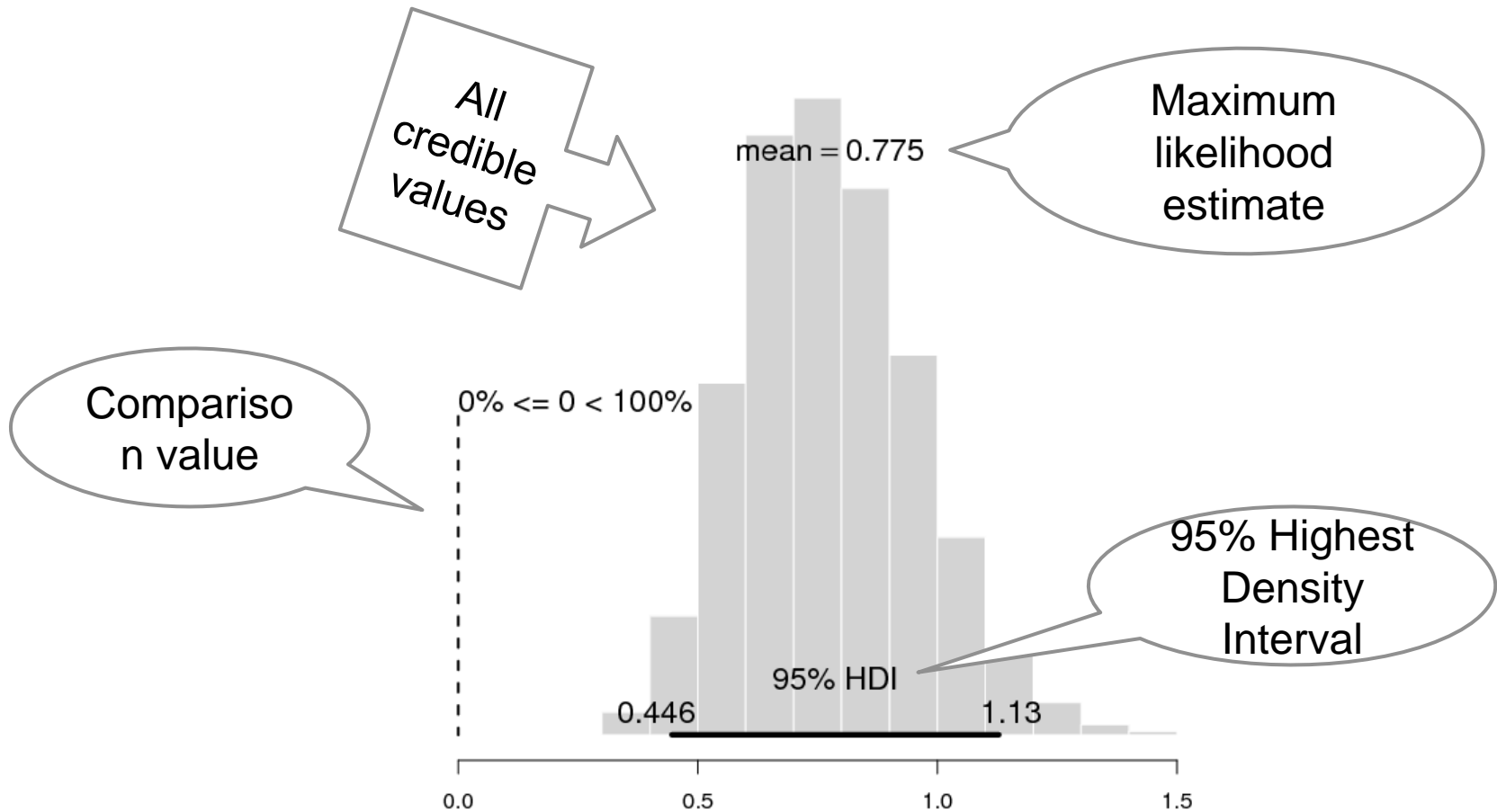
▶ Probability changes with prior knowledge



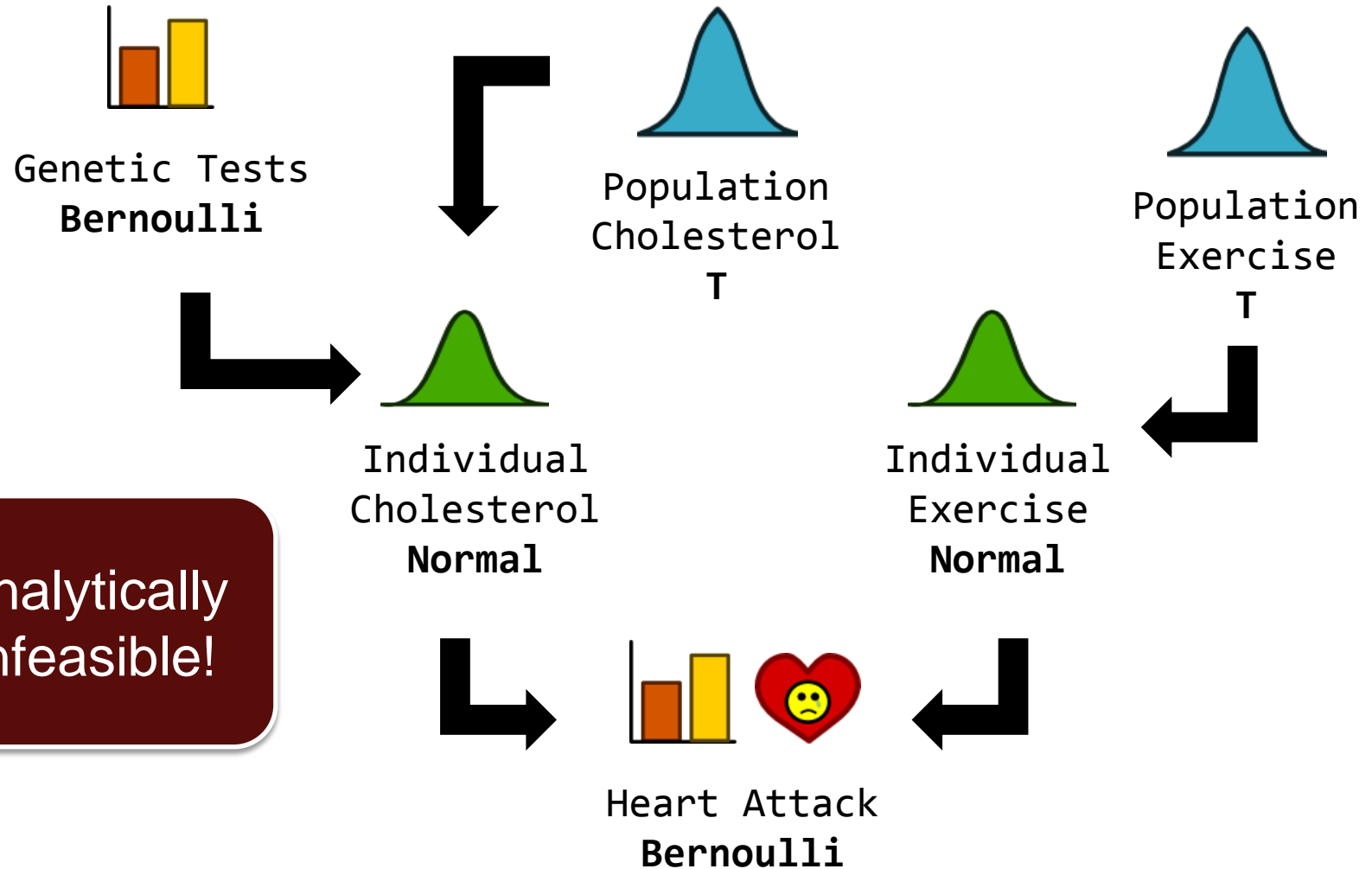
Making Things Fuzzy



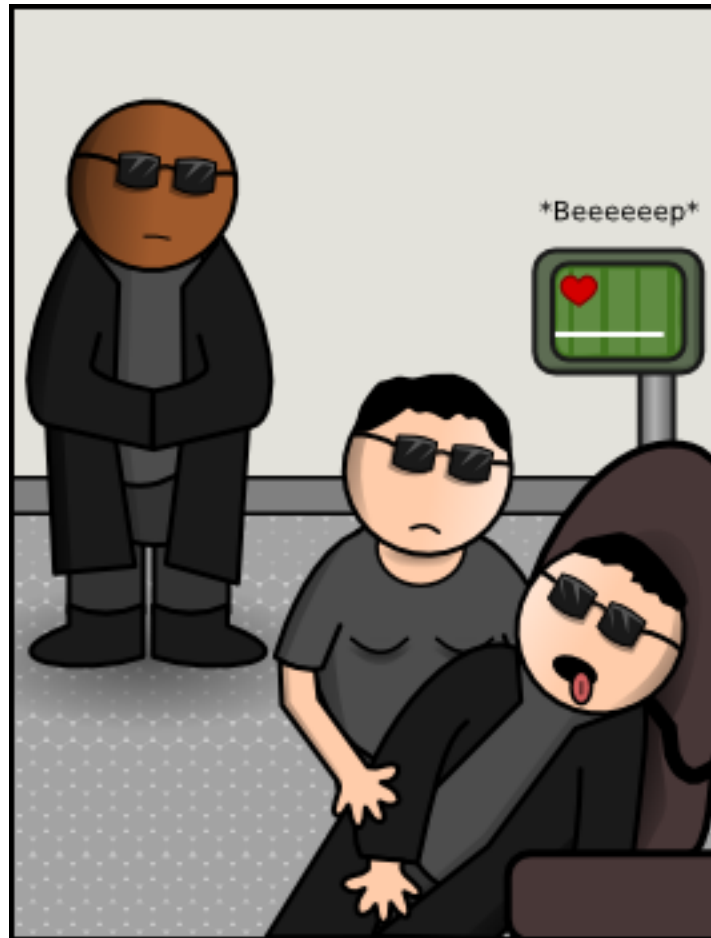
Interpreting Bayesian Results



Hierarchical Modeling with Parameters



Part 2 - The Real World



Markov Chain Monte Carlo Method

▶ Metropolis Algorithm

- ▶ Chains explore the posterior via random walk (converges in the limit)
- ▶ Proposal distribution controls how jumps are accepted/rejected

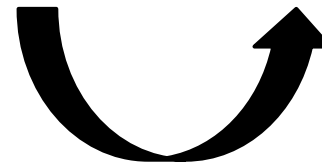
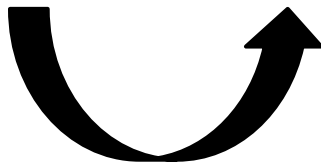
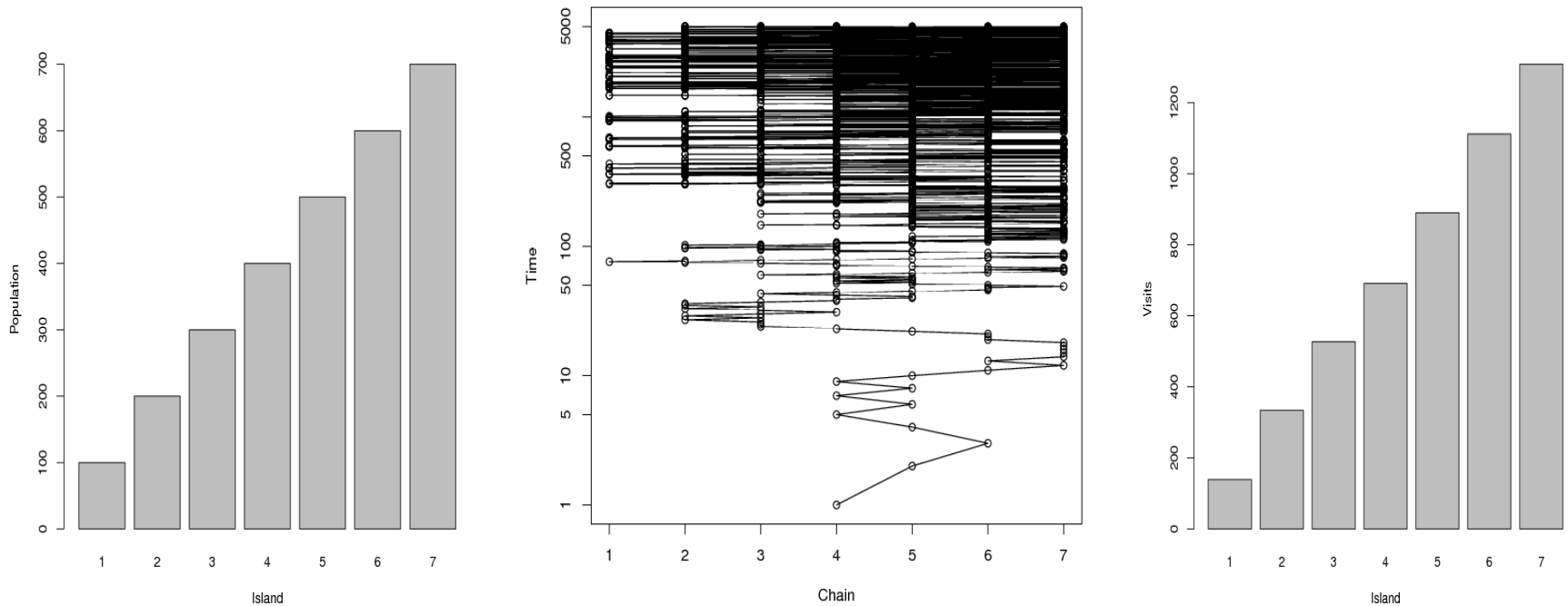
▶ Gibbs Sampling

- ▶ Conjoint parameter distribution for proposal distribution
- ▶ Used by BUGS software

```
j = (rand() < 0.5) ?  
    i - 1 : i + 1  
if (pop[j] > pop[i]):  
    i = j  
else:  
    p_move = (pop[j] / pop[i])  
    i = (rand() <= p_move) ?  
        j : i
```

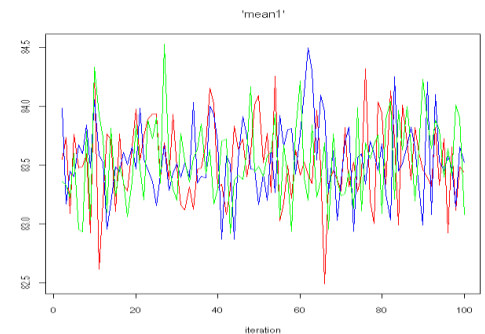
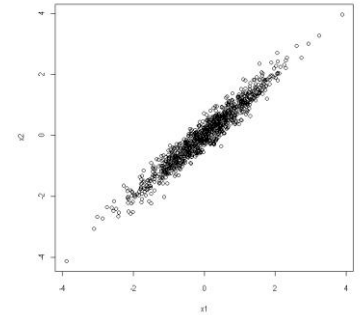


Metropolis Algorithm



Metropolis and Gibbs “gotchas”

- ▶ Metropolis algorithm
 - ▶ Requires tuning of proposal distribution
 - ▶ “Clumpier” trajectories due to rejected jumps
- ▶ Gibbs Sampling
 - ▶ Can get stuck when parameters are highly correlated
 - ▶ Must be able to derive conditional probabilities of each parameter on the other and generate samples
- ▶ Both
 - ▶ Chains should be checked for autocorrelation (thinning)
 - ▶ May take time to find bulk of posterior (burn-in)



Example – Parameter Recovery

R Code

```
# Generate parameters
mean1 = runif(1, 0, 100)
mean2 = runif(1, 0, 100)
prec1 = runif(1, 0, 1)
prec2 = runif(1, 0, 1)

# Generate data
y1 = rnorm(s, mean1,
           1/sqrt(prec1))

y2 = rnorm(s, mean2,
           1/sqrt(prec2))
```

BUGS Code

```
model {
  # Likelihood
  for (i in 1:numY1) {
    y1[i] ~ dnorm(mean1, prec1)
  }

  for (i in 1:numY2) {
    y2[i] ~ dnorm(mean2, prec2)
  }

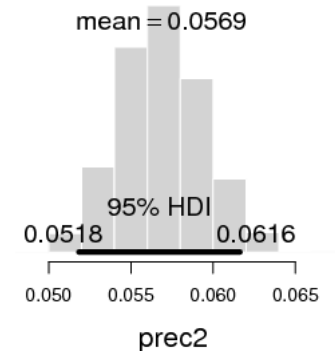
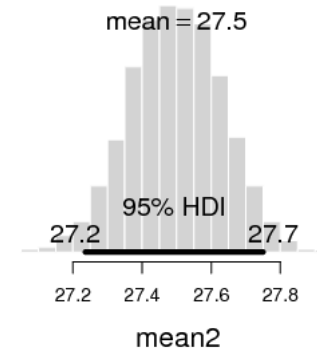
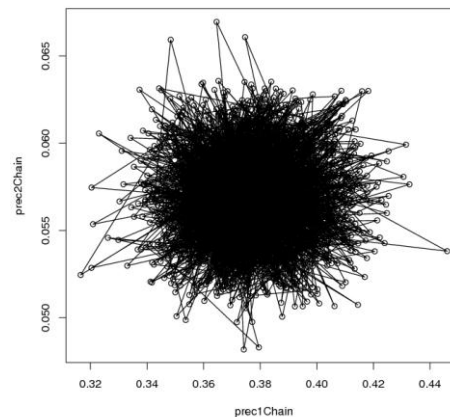
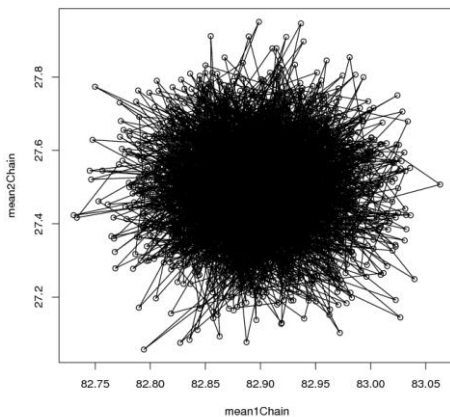
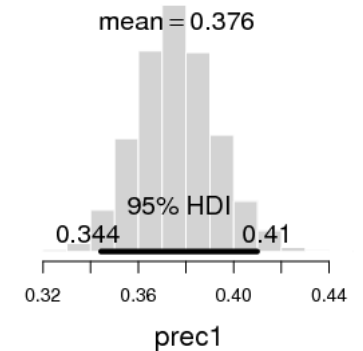
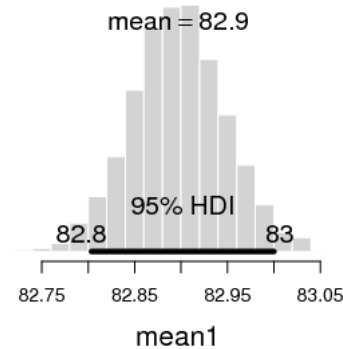
  # Priors
  mean1 ~ dunif(0, 100)
  mean2 ~ dunif(0, 100)
  prec1 ~ dunif(0, 1)
  prec2 ~ dunif(0, 1)
}
```



Example – Parameter Recovery (dense data)

Real Values (1000 samples)

- mean1: 82.89689
- prec1: 0.3512249
- mean2: 27.37247
- prec2: 0.05239301



Parameter Chains

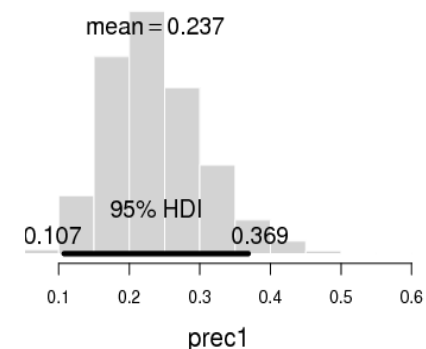
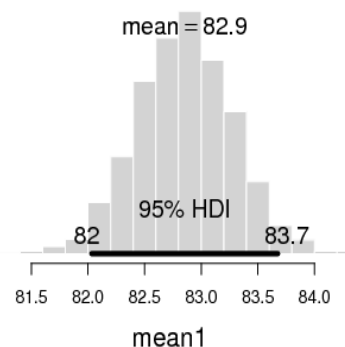
Burn-in: 100 steps
Steps per chain: 1000
Thinning: 2



Example – Parameter Recovery (sparse data)

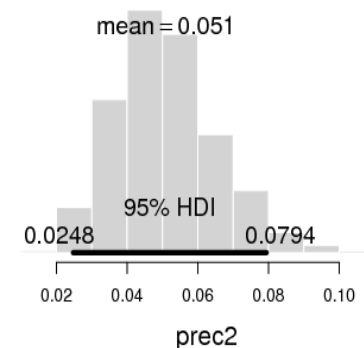
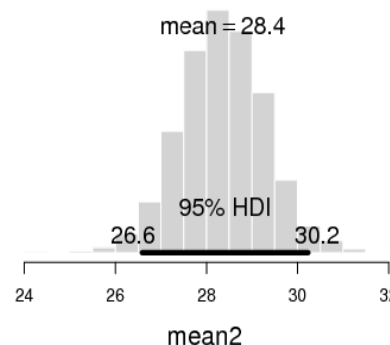
Real Values (25 samples)

- mean1: 82.89689
- prec1: 0.3512249
- mean2: 27.37247
- prec2: 0.05239301



HDI Width Increase

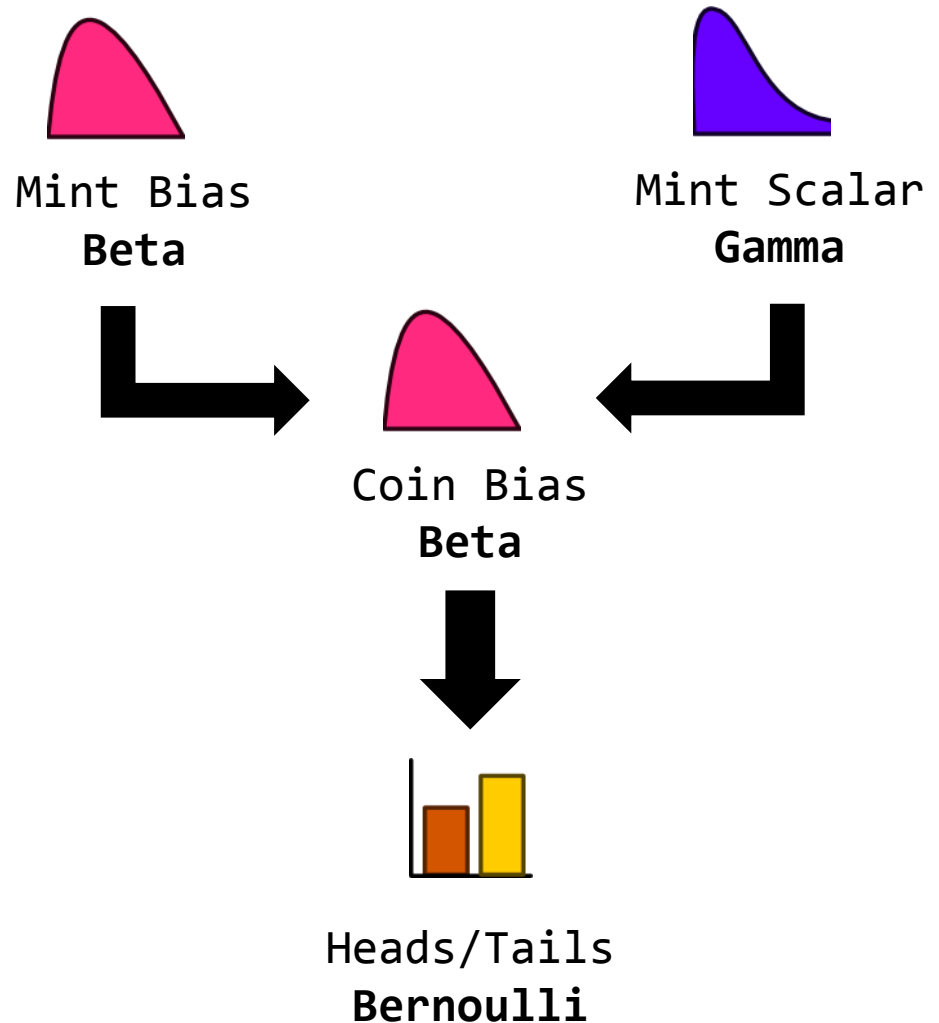
- mean1: 8.5x
- prec1: 3.97x
- mean2: 7.2x
- prec2: 5.57x



Burn-in: 100 steps
Steps per chain: 1000
Thinning: 2

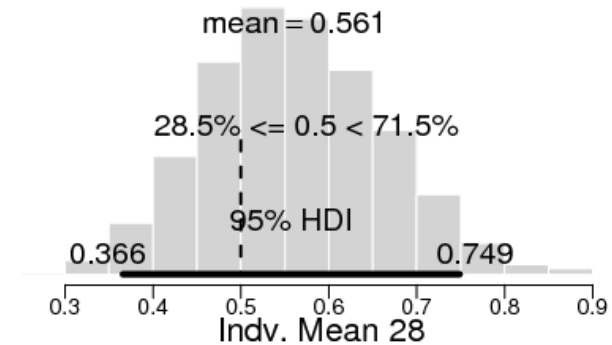
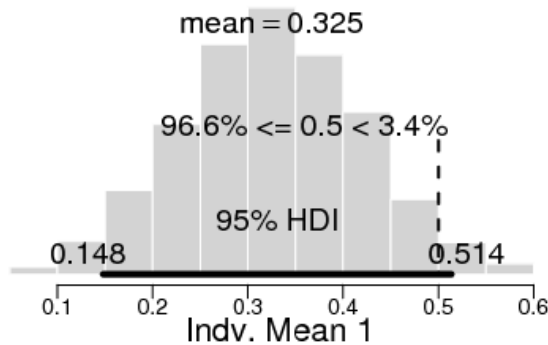
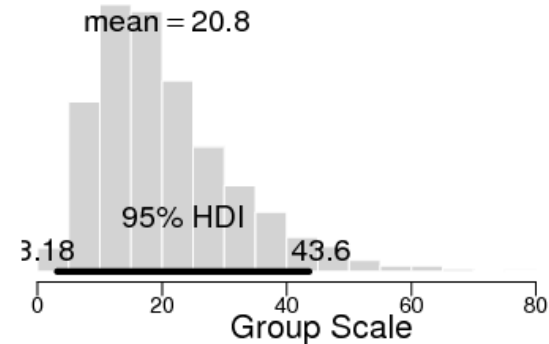
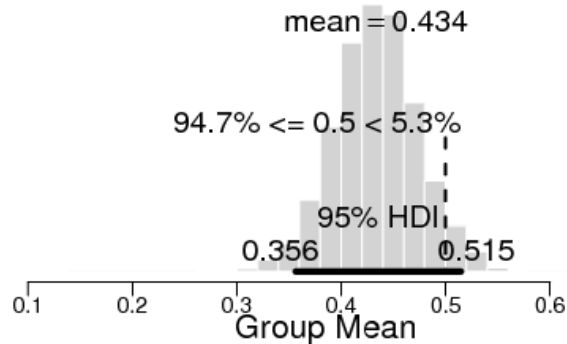
Mutually Informing Data

```
model {  
  for (i in 1:numFlips) {  
    f[i] ~ dbern(cb[c[i]])  
  }  
  
  for (i in 1:numCoins) {  
    cb[i] ~ dbeta(cbA, cbB)  
  }  
  
  # Hyper-priors  
  cbA <- cbMn * cbSc1 + 1  
  cbB <- (1 - cbMn) * cbSc1 + 1  
  cbMn ~ dbeta(mintA, mintB)  
  cbSc1 ~ dgamma(mintS, mintR)  
  
  # Priors...  
}
```



Example – Therapeutic Touch

- ▶ Can TT practitioners detect “energy field” better than chance?
- ▶ 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5,



Questions?

