MRC | Cognition and Brain Sciences Unit

Probabilistic Computing and Bayesian Statistical Analysis

Fawad Jamshed & Jonathan Fawcett

Bayesian Interest Group

3rd December 2013

Overview

Fawad

- What are probabilistic languages?
  - What is Infer.Net?
  - What is Church?

Jonathan

- How can Bayesian inference be used in our data analysis?
  - How is Bayesian inference implemented in R?
  - What are some sample applications?

Part 1:

WHAT ARE PROBABILISTIC LANGUAGES ?

Probabilistic Programming Languages

- Probabilities describe degrees of belief, and probabilistic inference describe rational reasoning under uncertainty.
- A probabilistic programming language is a high-level language that makes it easy for a developer to define probability models and then "solve" these models automatically.

What does it mean to perform inference automatically?

- In a probabilistic programming, users specify a probabilistic model in its entirety. A simulation is a computer program that takes some initial conditions as an input.
- Then it uses the programmer's assumptions about the interactions between these variables to produce forecasts.
- The probabilistic language's runtime environment runs the program both forward and backward from causes to effects (data) and backward from the data to the causes.

Observations and the Posterior Distribution

- A prior represents your understanding of the system before you make a particular set of observations.
- The corresponding posterior represents your understanding of the system after you have made the observations.
- The more evidence you have, the less important your prior belief is.

Probabilistic Programming Languages

- Probabilistic programming languages, unifies general purpose programming with probabilistic modelling
- Application areas include scientific modelling, information retrieval, bioinformatics, vision, semantic web, business intelligence, security, human cognition, and more.
- Probabilistic programming systems include BUGS, IBAL, BLOG, Infer.NET, Church, and STAN amongst others.

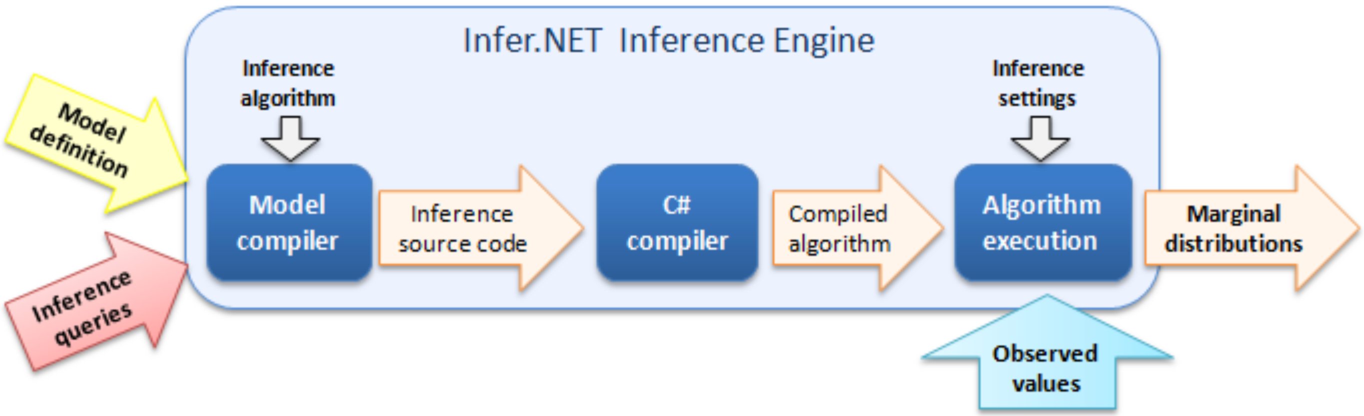
Infer.Net

- Infer.NET is a framework for running Bayesian inference in graphical models. It can also be used for probabilistic programming.
- Infer.NET is being developed in the Machine Learning and Perception group at Microsoft Research Cambridge by Tom Minka and his team.
- Infer.NET is used on wide variety of domains including information retrieval, bioinformatics, vision etc.

Infer.Net

- Is developed using ground up so can be scalable to large models and datasets.
- Effectively act as a compiler that converts code directly into source code with no overhead costs.
- Can be used with any dot net language that includes C sharp, C++, visual basics etc
- Commercial use of Infer.NET is limited to Microsoft.

Infer.Net Model



Overview of working

- Create a Model
  - An Infer.NET application is built around a probabilistic model, which defines the random variables and how they are related
- Observe Random Variables
  - you observe one or more of the model's random variables by assigning values to their observed value properties
- Infer Posteriors
  - A posterior incorporates the information from the prior and the observations, and represents your new and presumably improved knowledge of the variable's value.
- Use the Posteriors
  - Use the posterior as the variable's new prior. Make some additional observations. Compute a new posterior

Code Example

- Variables
  - Variable<bool> firstCoin = Variable.Bernoulli(0.5);
  - Variable<bool> secondCoin = Variable.Bernoulli(0.5)
  - Variable<bool> bothHeads = firstCoin & secondCoin;
- Inferring
  - InferenceEngine ie = new InferenceEngine(); Console.WriteLine("Probability both coins are heads: "+ie.Infer(bothHeads));
- Output
  - Probability both coins are heads: Bernoulli(0.25)

Benefits of Infer.net

- **Rich modelling language**
  - Support for univariate and multivariate variables, both continuous and discrete.
- **Multiple inference algorithms**
  - Built-in algorithms include Expectation Propagation, Belief Propagation (a special case of EP), Variational Message Passing and Gibbs sampling etc.

Benefits of Infer.net

- **Designed for large scale inference**
  - Infer.NET compiles models into inference source code which can be executed independently with no overhead.
  - It can also be integrated directly into your application
  - In addition, the source code can be viewed, stepped through, profiled or modified as needed, using standard development tools.

Benefits of Infer.net

- **User-extendable**
  - Infer.NET uses a plug-in architecture which makes it open-ended and adaptable.
  - custom code can be written and freely mixed with the built-in functionality, minimising the amount of extra work that is needed.

What Infer.Net can not do

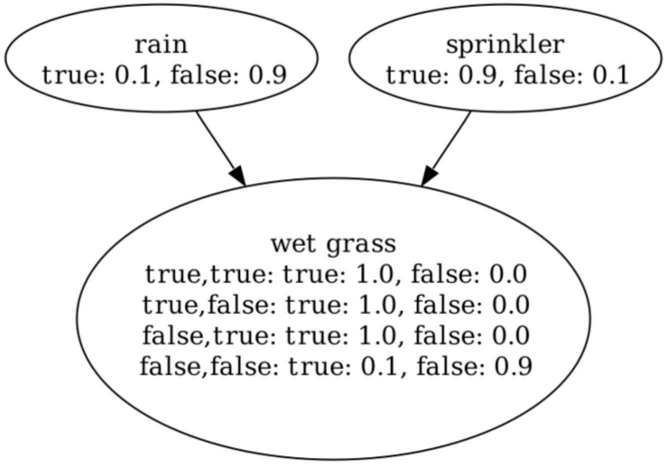
- Non-parametric models (e.g. Dirichlet process) are not supported but may be supported in future releases.

What is Church?

- Based on Scheme
- Currently used for AI and linguistic applications
- Recursion and Memoization
- Various implementations
  - MIT-Church
  - Bher
  - Stochastic Matlab

Very different syntax (*Sprinkler Problem*)

```
(define rain
  (flip .1))
(define sprinkler
  (flip .9))
(define wet
  (if (or rain sprinkler)
   true (flip .1)))

wet
```



Part 2:

HOW CAN BAYESIAN INFERENCE BE USED IN OUR DATA ANALYSIS?

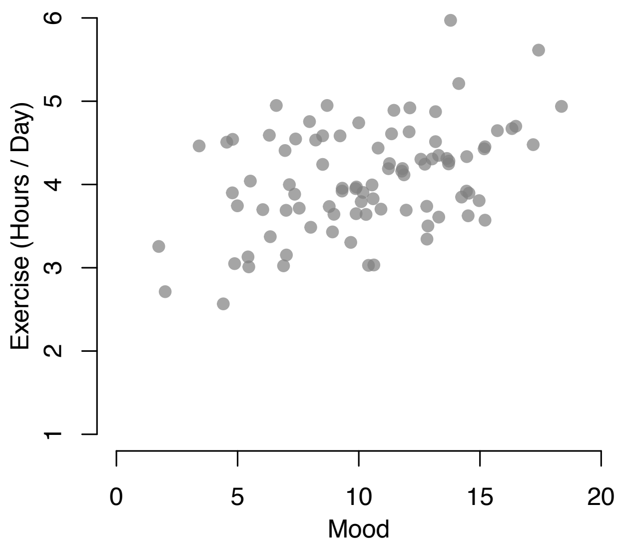
Example: Exercise and Mood

Example: Exercise and Mood

- Observational study relating daily exercise to mood in individuals suffering from a past depressive episode
  - 90 participants
  - Exercise: *Avg. # hours per day*
  - Happiness Scores: *0 = Low Happiness, 20 = High Happiness*

Example: Exercise and Mood

- Observational study relating daily exercise to mood in individuals suffering from a past depressive episode
  - 90 participants
  - Exercise: *Avg. # hours per day*
  - Happiness Scores: *0 = Low Happiness, 20 = High Happiness*



Choosing the right research question

- **Our question:** In our target population, does the slope of the linear relationship between exercise and mood differ from 0?

Choosing the right research question

- **Our question:** In our target population, does the slope of the linear relationship between exercise and mood differ from 0?
- **Traditional statistics:** Assuming that there is no relationship between exercise and mood, if we were to repeat the experiment many times, what proportion of times would we obtain a slope that is as great as or greater than the slope observed our experiment?

Choosing the right research question

- **Our question:** In our target population, does the slope of the linear relationship between exercise and mood differ from 0?
- **Traditional statistics:** Assuming that there is no relationship between exercise and mood, if we were to repeat the experiment many times, what proportion of times would we obtain a slope that is as great as or greater than the slope observed our experiment?
- **Bayesian statistics:** Given the current data and our prior beliefs, how credible is it that the slope between exercise and mood differs from 0?

Implementing Bayesian statistics

- The R programming language
  - Free and open-source (http://cran.r-project.org)
  - Embedded mathematical / graphical functions
  - Great community support
  - Several GUIs and IDEs available
  - Loads of free statistical packages

Bayesian statistics using R

- OpenBUGS (**B**ayesian Inference **U**sing **G**ibbs **S**ampling)
- JAGS (**J**ust **A**nother **G**ibbs **S**ampler)
- Stan (named for Stanislaw Ulam; uses a variant of Hamiltonian Monte Carlo Sampling)

Bayesian statistics using Stan

- Using Stan in R (rstan package)
  - Load your data into R
  - Specify your model using Stan
    - Data
    - Parameters
    - Model
  - Call the stan function
  - Plot or summarize the posterior distribution

Linear regression

- The relationship between X and Y, defined by the equation:

$$Y = \alpha + \beta * X + \varepsilon$$

- Where
  - Y: Criterion/Dependent Variable
  - X: Predictor/Independent Variable
  - α: Intercept (Y when X=0)
  - β: Slope (unit change in Y per unit change in X)
  - ε: Residual error term

Stan: Specifying your model

```
data {
        // This section contains information pertaining
        // to the data collected in your study
}
parameters {
        // This section contains information pertaining
        // to what you hope to estimate
}
model {
        // This section contains your assumptions and the
        // relationships amongst your data/parameters
}
```

Stan: Specifying the Data

```
data {
  int<lower=0> N;        // This refers to your sample size
  vector[N] x;           // This will be a vector containing X
  vector[N] y;           // This will be a vector containing Y
}
```

N: A non-negative integer

x: A vector of real numbers containing N elements

y: A vector of real numbers containing N elements

Stan: Specifying the Parameters

```
parameters {
  real intercept;                    // Your intercept
  real slope;                        // Your slope
  real<lower=0> sigma;               // Unexplained variance
}
```

intercept: The intercept (α) from the linear regression
slope: The slope (β) from the linear regression
sigma: The error term (ε) from the linear regression

Stan: Specifying the Model

```
model {
  intercept ~ normal(0, 20);      // Prior for intercept
  slope ~ normal(0, 20);          // Prior for slope
  sigma ~ uniform(0,1000);        // Prior for error term
  y ~ normal(intercept + slope * x, sigma);      // Model for y
}
```

intercept: The intercept is modeled as *N*(M=0, SD=20)

slope: The slope is modeled as *N*(M=0, SD=20)

sigma: The error term is modeled as a uniform distribution ranging from 0 to 1000

y: The DV is modeled as a Normal distribution with M equal to the linear equation and SD equal to sigma

Stan: Complete Model

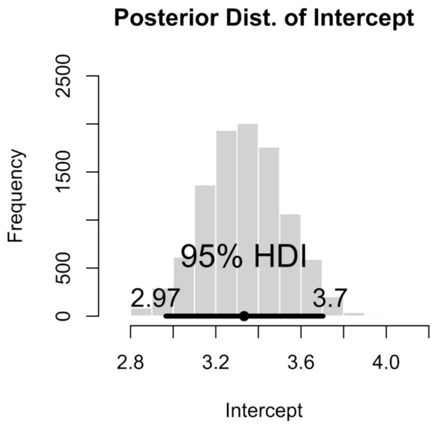
```
linear_model =
"
data {
  int<lower=0> N;                                  // Sample size
  vector[N] x;                                     // Vector containing X
  real y[N];                                       // Vector containing Y
} parameters {
  real intercept;                                  // Your intercept
  real slope;                                      // Your slope
  real<lower=0> sigma;                             // Unexplained variance
} model {
  intercept ~ normal(0, 20);                       // Prior for intercept
  slope ~ normal(0, 20);                           // Prior for slope
  sigma ~ uniform(0,1000);                         // Prior for error term
  y ~ normal(intercept + slope * x, sigma);        // Model for y
}
"
```

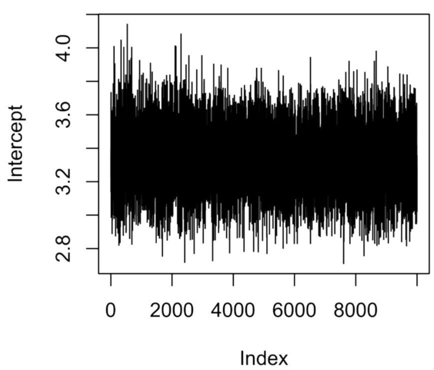
Stan: Complete Model

```
library(rstan)

your_dat <- read.table(‘datafile.csv’, header=TRUE, sep=‘,’)

model_dat <- list(N = nrow(your_dat ), x = your_dat$x, y = your_dat$y)

linear_fit <- stan(
  model_code = linear_model # Variable containing model specified earlier
  , data = model_dat
  , iter = 10000)

fit_parameters <- extract(linear_fit, permute=TRUE)
```

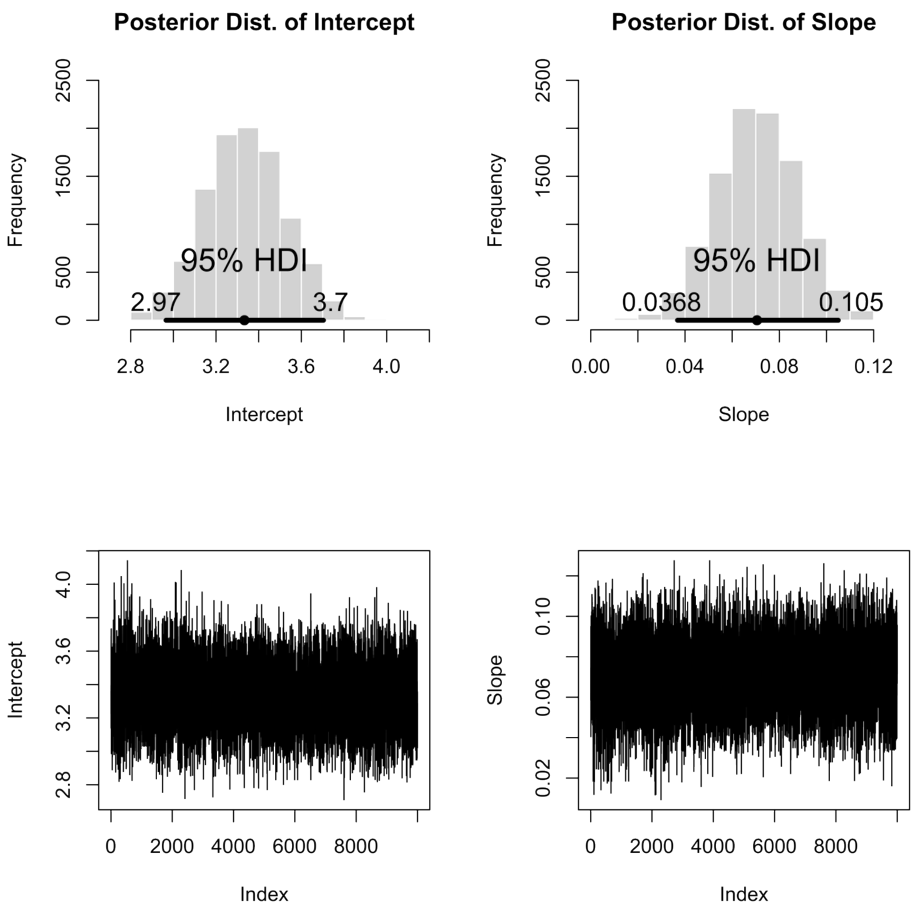
Stan: Interpreting the Results

Stan: Interpreting the Results

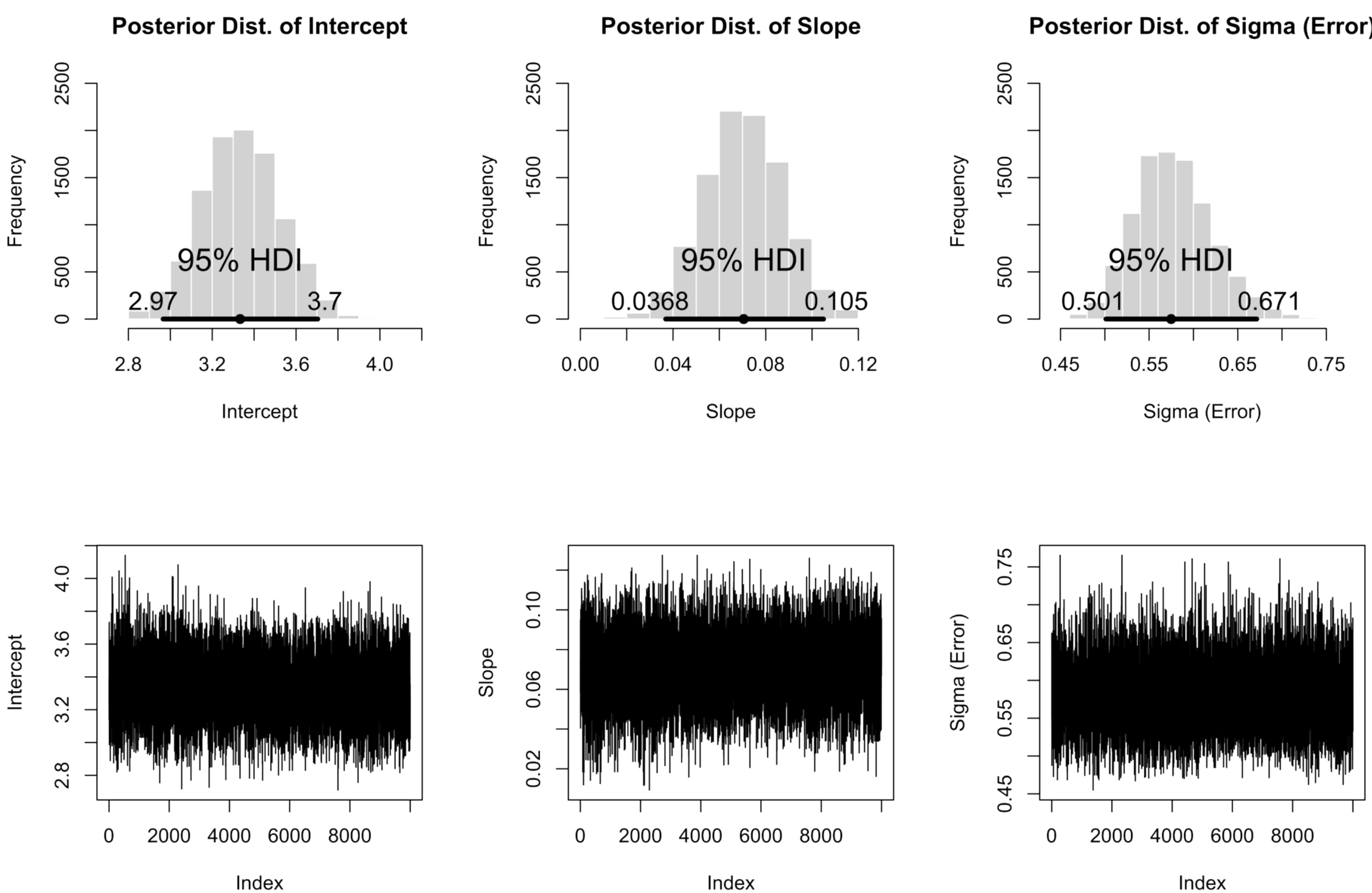




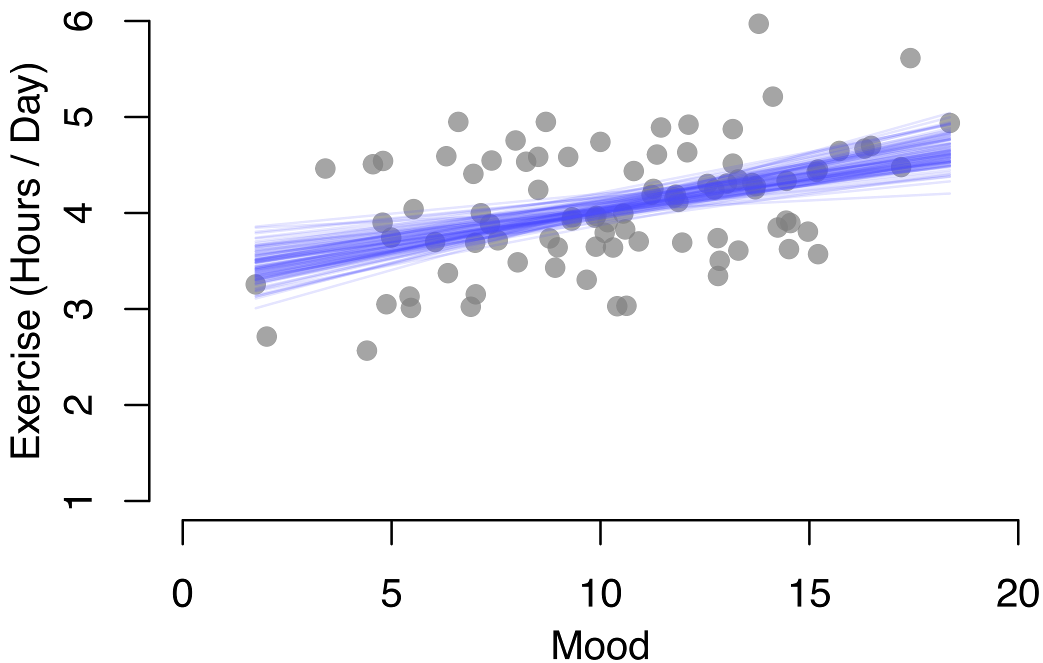
Stan: Interpreting the Results



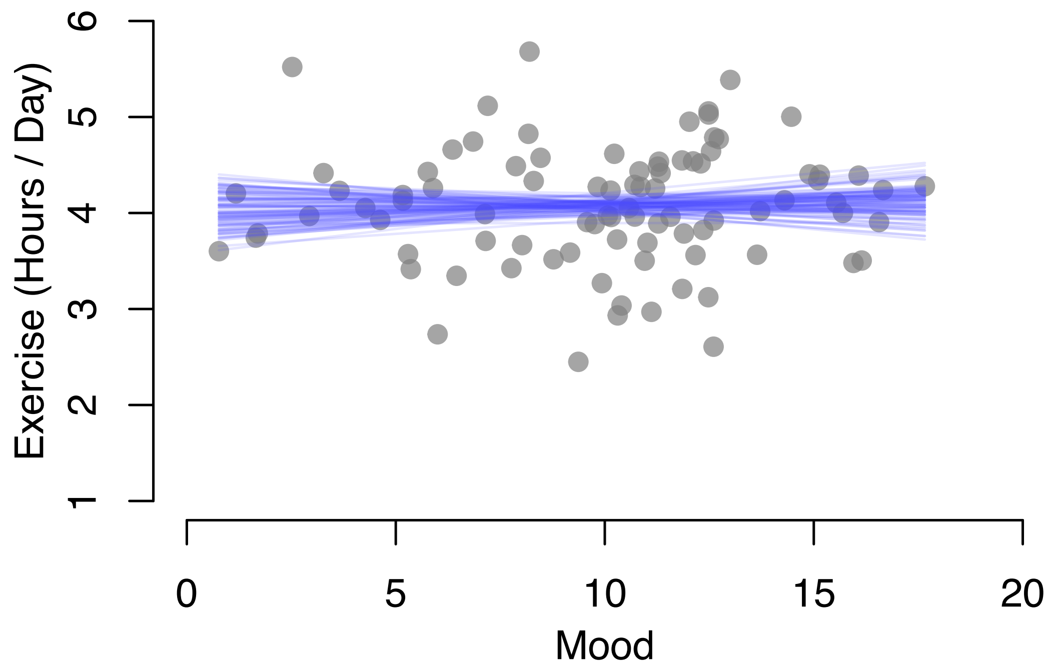
Stan: Interpreting the Results



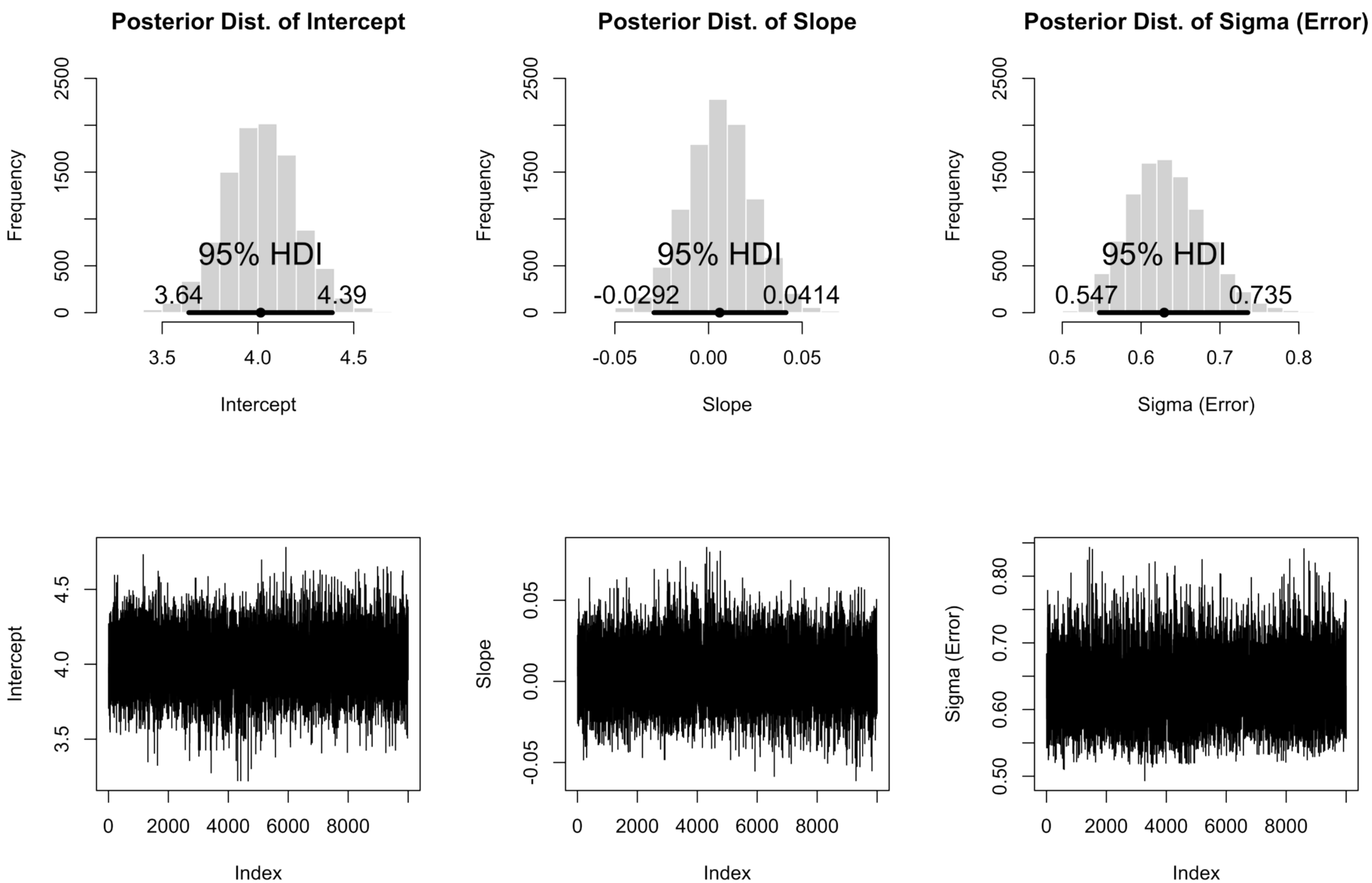
Stan: Interpreting the Results



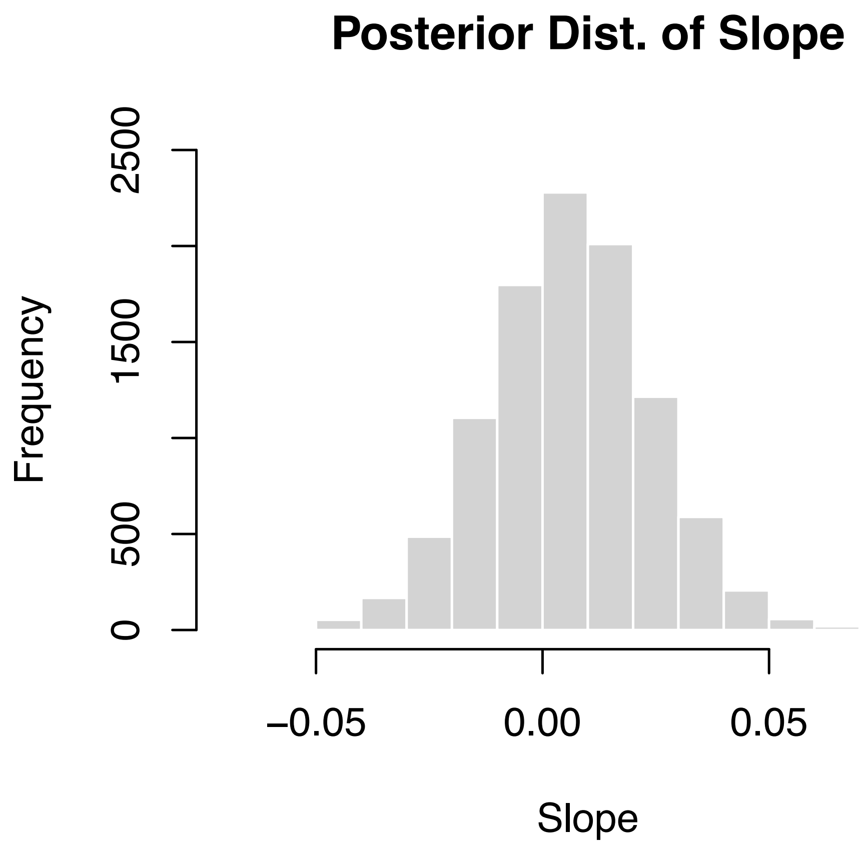
Stan: Supporting the "Null"



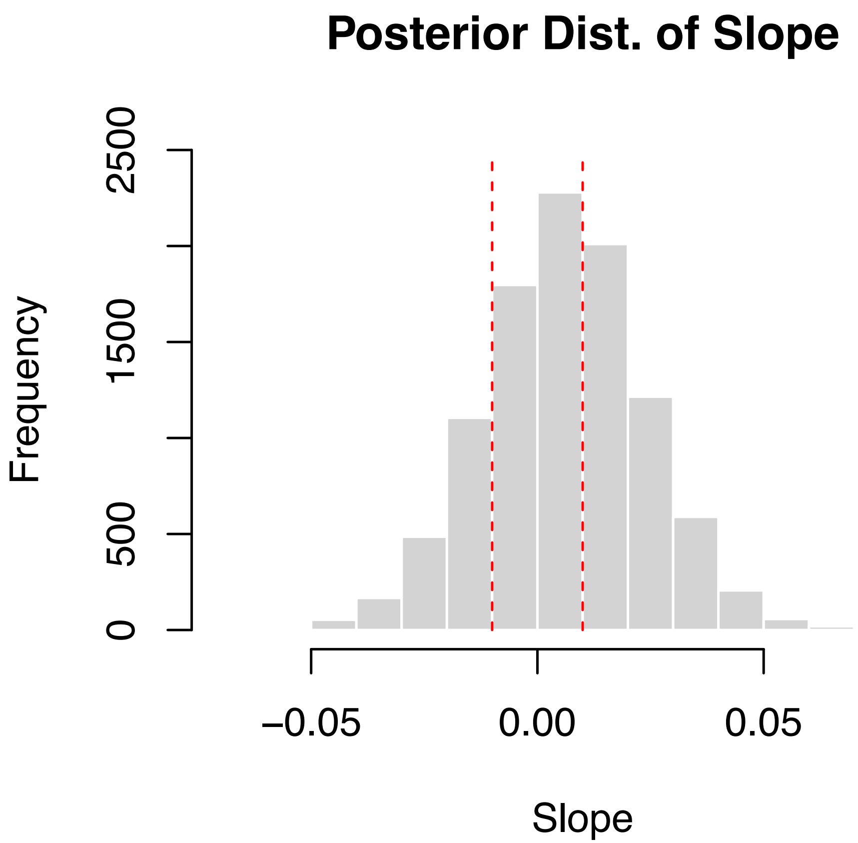
Stan: Supporting the "Null"



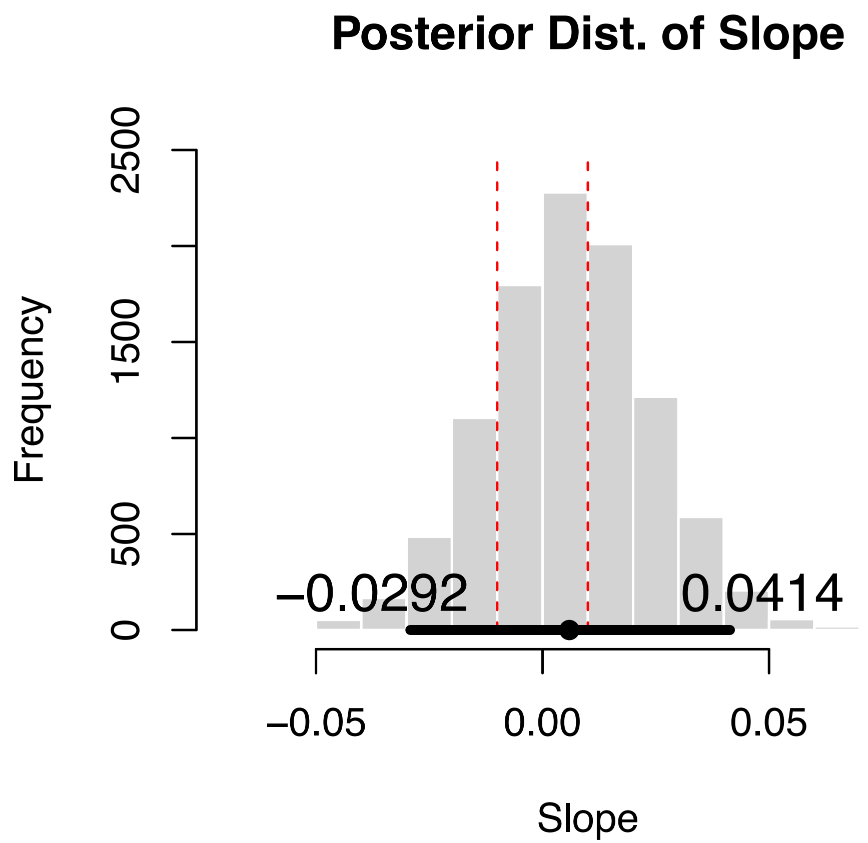
Stan: Regions of Practical Equivalence



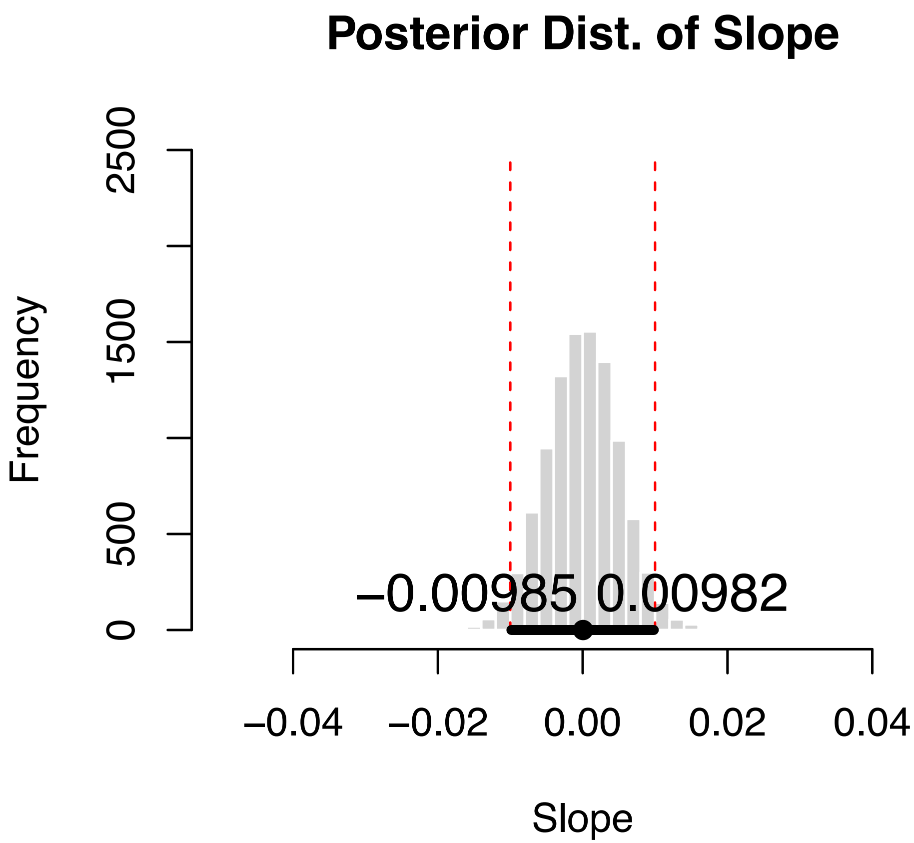
Stan: Regions of Practical Equivalence



Stan: Regions of Practical Equivalence



Stan: Regions of Practical Equivalence



The Benefits of Bayesian Analysis

- Explicates assumptions
- Scales well (e.g., multiple predictors, interactions)
- In line with actual behaviour
  - No need to consider stopping rule
  - No need to worry about multiplicity control
- Flexibility
  - “Build-your-own-analysis”
  - Can model outliers
  - Can model multicollinearity
  - Can deal with unequal N
  - Can deal with missing/censored data

References

- Dienes, Z. (2011). Bayesian Versus Orthodox Statistics: Which Side Are You On? Perspectives on Psychological Science, 6(3), 274–290.
- Goodman, N., Mansinghka, V., Roy, D., Bonawitz, K., & Tarlow, D. (2012). Church: a language for generative models. *arXiv preprint arXiv:1206.3255*.
- Kruschke, J. (2010). Doing Bayesian data analysis: A tutorial with R and BUGS.
- Stan Homepage: http://mc-stan.org
- R Homepage: http://cran.r-project.org