Bayesian Statistics in R

Lynd Bacon
Loma Buena Associates
Northwestern University

Bi[R]mingham R Meetup

April 24, 2015

Schema

- Bayes
- Why R for Bayes?
- MCMC and related techniques
- An example application: modeling stated preferences using choice-based conjoint data

Bayes Theorem

Consider two events, A and B

The probability that A occurs, given B, is given by Bayes Theorem:

$$p(A|B) = \frac{p(B|A)\,p(A)}{p(B)}$$

p(“A|B”) is the probability of A conditional on B

The “Inverse Probability Law” from Conditional Probabilities

$$p(B|A)=\frac{p(B \text{ and } A)}{p(A)} \qquad \text{Eqn.1}$$

$$p(A|B)=\frac{p(A \text{ and } B)}{p(B)} \qquad \text{Eqn.2}$$

$$p(B|A)p(A)=p(B \text{ and } A)=p(A \text{ and } B)=p(A|B)p(B) \qquad \text{Eqn.3}$$

$$p(B|A)=\frac{p(A|B)p(B)}{p(A)}, \text{ also } p(A|B)=\frac{p(B|A)p(A)}{p(B)} \qquad \text{Eqn.4}$$

Bayes Theorem, Again

$$p(\theta|D)=\frac{L(D|\theta)\,p(\theta)}{p(D)}$$

$$where:$$

$$p(\theta|D): posterior\ distribution\ of\ \theta$$

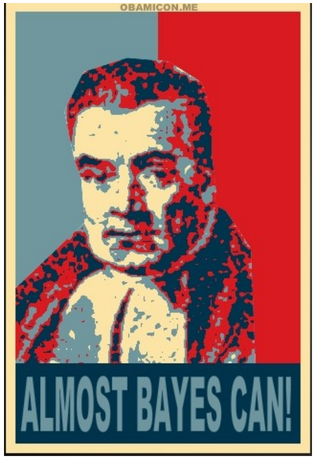
$$L(D|\theta): Likelihood$$

$$p(\theta): prior\ distribtion\ of\ \theta$$

$$p(D): data\ density$$

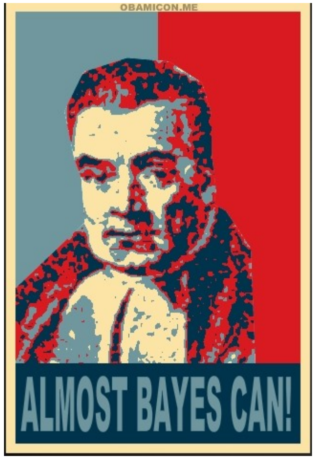
$$p(\theta|D)\propto L(D|\theta)\,p(\theta)$$

Why Bayes?



- Incorporate prior knowledge
- Shrinkage estimators
- Avoid asymptotic estimators
- Avoid NHST (loose the p values)
- Exploit information pooling, esp. with sparse data
- Estimate models not amenable to closed form solutions
- “Natural” treatment of missing values
- Get arbitrarily precise estimates

Why Not? (“inconveniences”)



- Full probability models
- Those “Dreadful Priors”
- Computational intensity
- Results “overload”
- Explaining Bayesian inference to non-Bayesians

Why do Bayes in R?

- Wide (and ever growing!) range of packages and algorithms
  - parametric, semi-parametric, and non-parametric Bayesian models
  - social science, biostat, econometric, machine learning applications
  - General tools, specific model methods, model evaluation tools
- Packages for learning or teaching Bayesian statistics, e.g. LearnBayes
- The R language is convenient for rapid prototyping of methods and algorithms
- R has superior methods for graphics and data visualization
- Packages that access algorithms (“sampling engines”) programmed using other languages, e.g.
  - BUGS, JAGS, STAN
  - C/C++, FORTRAN, Python, etc.
- Tools for accessing databases, proprietary file formats
- Large developer and user communities

Bayes and Simulation Methods

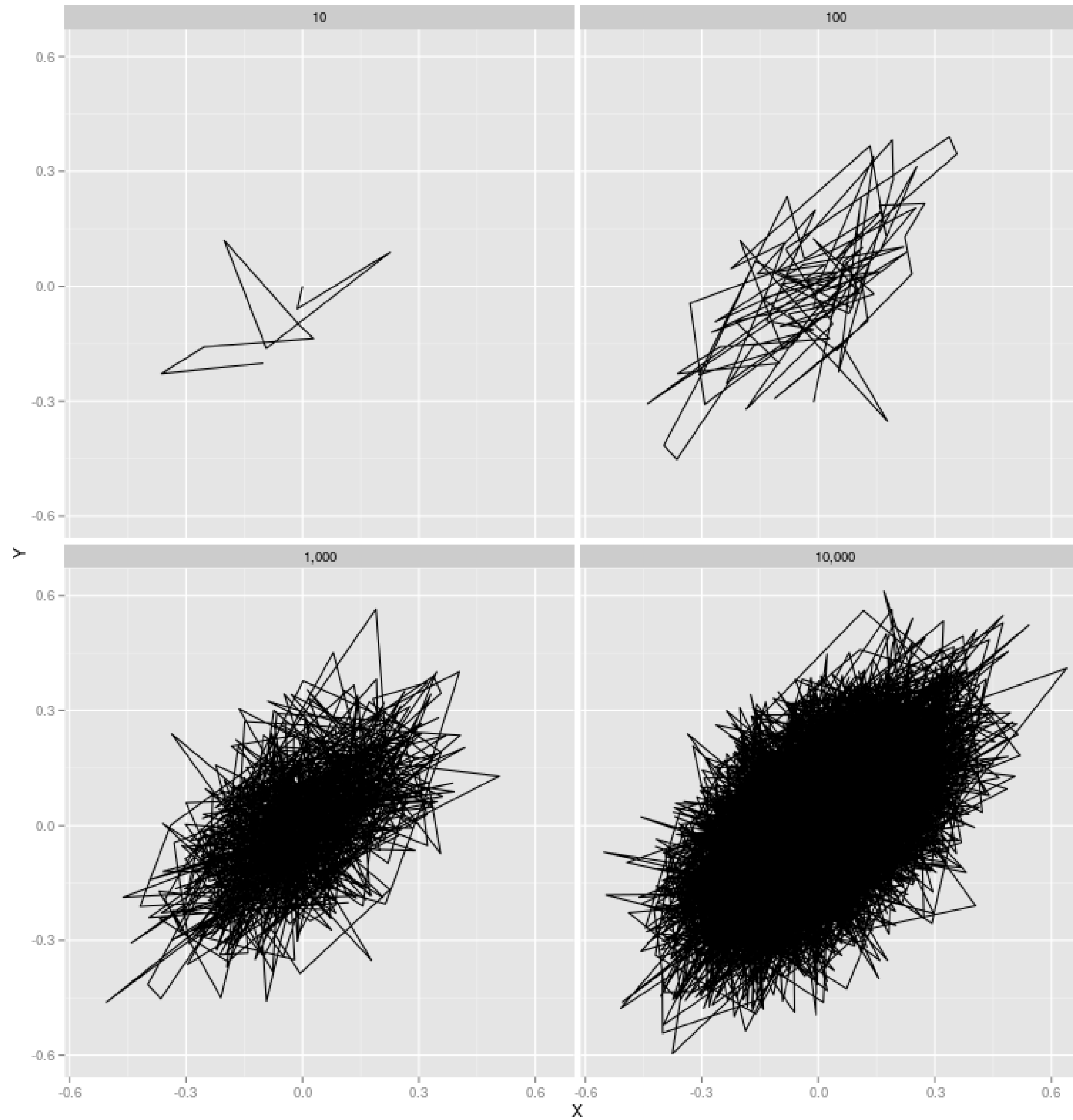
- Model parameters estimated by “sampling” from conditional posteriors (or proxies for them)
- Methods generally referred to as Markov Chain Monte Carlo (MCMC)
  - Markov process: values generated depend on the current state of the simulation algorithm
  - Sample “chains” of conditional, randomly selected values from stationary posterior distributions
  - Several flavors depending on the posterior densities of interest

Common MCMC Sampling Methods

- Gibbs
  - Geman & Geman, 1984; Gelfand & Smith, 1990
  - Generate random variables from distributions indirectly, without having to calculate densities
  - Used when it's possible to sample directly (“draw”) from marginal posterior distributions
- Metropolis, Metropolis Hastings
  - Metropolis & Ulam, 1949; Hastings, 1970
  - Posteriors can't easily be sampled directly, but can be evaluated
  - Samples from a “proposal” or “jump” distribution determine whether new posterior values are to be accepted or rejected.

Ex: “Gibbsed” Bivariate Normal

```
biVarGibbs.f=function(n,r){
  dmat=matrix(nrow=n,ncol=2)
  x=0
  y=0
  d=(1-sqrt(1-r^2))
  dmat[1,]=c(x,y)
  for (i in 2:n){
    x = rnorm(1,r*y,d)
    y = rnorm(1,r*x,d)
    dmat[i,]=c(x,y)
  }
  dmat
}
```



Ex: r=0.5
10, 100, 1,000, and 10,000 draws

Ex: Gibbsing Linear Regression

$$y_i = \beta x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

$A\ normal\ prior\ on\ \beta:$

$$p(\beta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-1}{2\sigma^2}(\beta - \bar{\beta})^2\right]$$

$Prior\ on\ \sigma^2:$

$inverse\ gamma,\ inverse\ \chi^{2,}$ or $uninformative$

$likelihood:$

$$p(y_i|\beta,\ sigma^{2,} x_i) = \prod_{i=1}^{I} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-1}{2\sigma^2}(y_i - \beta x_i)^2\right]$$

$Joint\ posterior:$

$$p(\beta, \sigma^2 | y_i, x_i) \propto p(y_i|\beta, \sigma^{2,} x_i)\, p(\beta)\, p(\sigma^2)$$

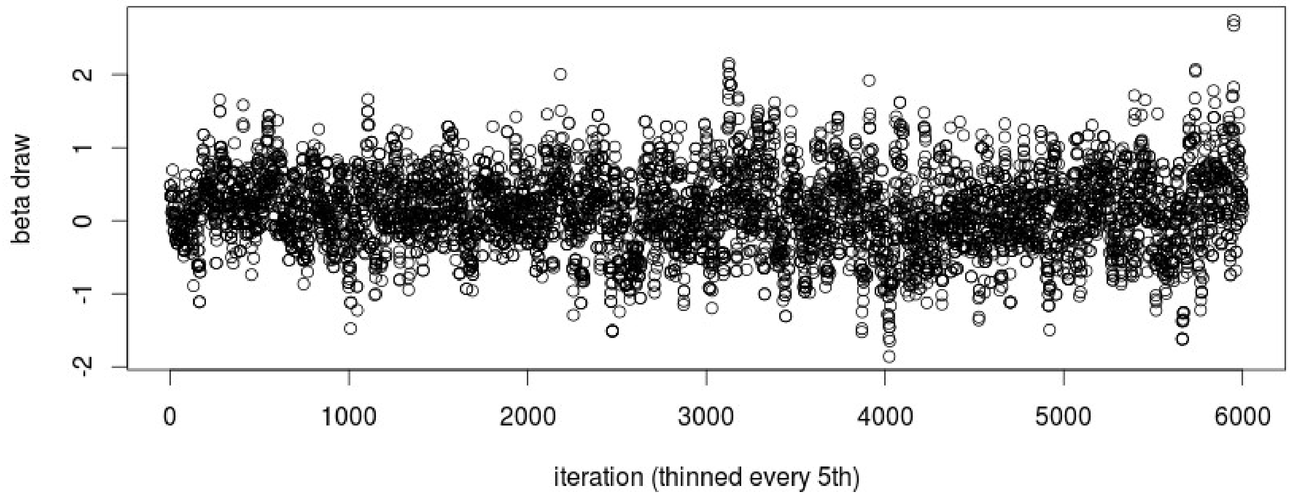
Gibbsed Linear Regression

- Specify initial values for β, σ^2
- Then:
  - Draw β given the data $\{y_i,x_i\}$ and the most recent draw of σ^2
  - Draw σ^2 given the data $\{y_i,x_i\}$ and the most recent draw of β
  - Repeat until done:
  - Retain for analysis draws after sufficient “burn-in”
    - Possibly after “thinning”
- Want to make this a binary probit model? Use *data augmentation*
  - Tanner & Wong, 1987
  - Albert & Chib, 1993 (binary, polychotomous responses)

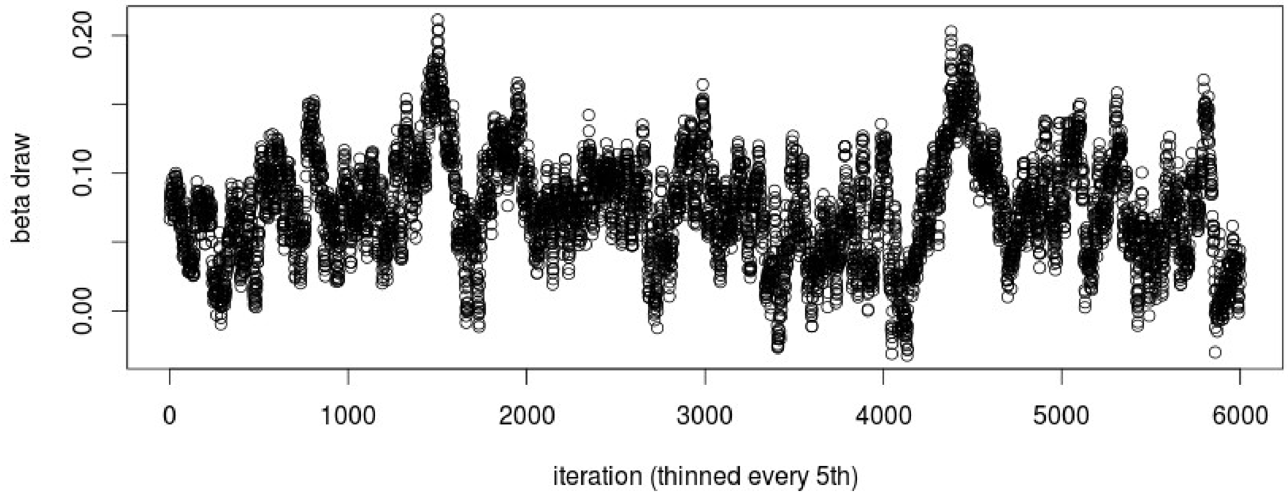
Model Evaluation

- Sufficient burn-in, “well behaved” chains
- Bayes Factor
- Posterior predictive checks
- MSE, MAE, DIC (w/ MVN posterior), log likelihood, root likelihood
- R tools include CODA, BOA, and methods in several packages
- The “sniff test”

Good Mixing, Bad Mixing



Pretty, pretty good.



Not so good.

An Example Application: Choice-Based Conjoint Measurement

- Conjoint measurement: a procedure for quantifying how elements of a product, service, or experience contribute to preference for it
- Stimuli are based on an experimental design and are presented in a series of questions
- Repeated measures
- Responses may be choices, rankings, ratings
- Most commonly used applied quantitative marketing science/research method

Example Choice-Based Conjoint Item

R packages used for choice task design include:

DoE.base
AlgDesign
ChoiceDes

Choice Set Mock-Up:

Curt's Outstanding Yurts Project, 2012

	1	2	3
Solar Fan	Yes	Yes	No
Doors	2	1	1
Windows	2	0	4
Fabric	Standard	Premium	Premium
Diameter (ft)	30	38	30
Price/ $ Sq Ft	50	70	40
I wouldn't buy any of these yurts X			

Choice and Utility

- “Utility” is a latent construct assumed to underlie stated or revealed preferences
- “Random Utility Theory:” preferences are determined by deterministic and stochastic components, e.g.

$$y_{ij}^{*} = X_{ij}\beta + \epsilon_{ij},$$

- Choice *models* describe expressed preferences, “RUMs”
  - May or may not mirror what goes on in peoples' heads
- Dan McFadden – Nobel Prize in economics
- Well known deviations and anomalies

A Hierarchical Choice Model

Choice responses from I respondents to K choice sets, each with J alternatives:

$$MN\ likelihood$$
$$y_{ijk}^{*} = X_{ijk}\beta_i + \epsilon_{ijk}$$
$$\beta_i \sim MVN\left(\bar{\beta}, V_{\beta}\right)$$
$$IW\ prior\ for\ V_{\beta}$$

We can add covariates that the β_i regression coefficients depend on, e.g.

$$\beta_i = \psi Z_i + u_i,\quad u_i \sim MVN\left(0, V_{\psi}\right)$$
$$with\ \psi_i \sim MVN,\quad IW\ prior\ for\ V_{\psi}$$

R packages for choice-based conjoint model estimation

- bayesm
- choiceModelR
- rstan
- rjags
- rbugs

Example Data: Tablet Design

- N=360
- Data collected using an online survey
- 36 choice sets with three alternatives in each
  - algorithmic design allowing estimation of brand name by price level effects
- Attributes (levels): RAM(3), screen size(3), processor speed(3), price(3), brand names(4)
- Participants selected the alternative in each set that they most preferred
- Additional data included brand ownership history, demographics, technology attitudes and behaviors
- Attribute "levels" effects coded in the design ("X") matrix
- Three brand X (linear) price regressors included

Model Estimation

- rhierMnlDP() from the R package bayesm
- 14 regressors (β's) , 2 β covariates
- 30,000 MCMC iterations, thinned every 5th
  - first (thinned) 5,000 as “burn-in”
  - last 1,000 for inferencing
- Of interest: inferences based on the conditional posteriors of β's and δ's.

Conjoint MCMC results

- rhierMnlDP() in the R package bayesm:

```
>fullrun2=rhierMnlDP(Data=Data2,Mcmc=mcmcrun1)
```

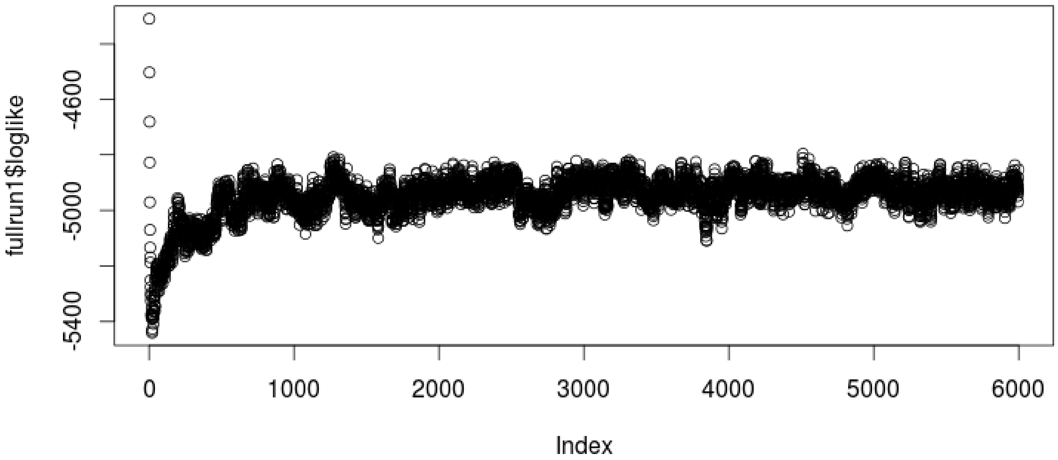
- ...returns (after a little time) results as “objects” in a list:

```
> names(fullrun2)
[1] "Deltadraw" "betadraw" "nmix" "alphadraw" "Istardraw" "adraw"
[7] "nudraw" "vdraw" "loglike"
```

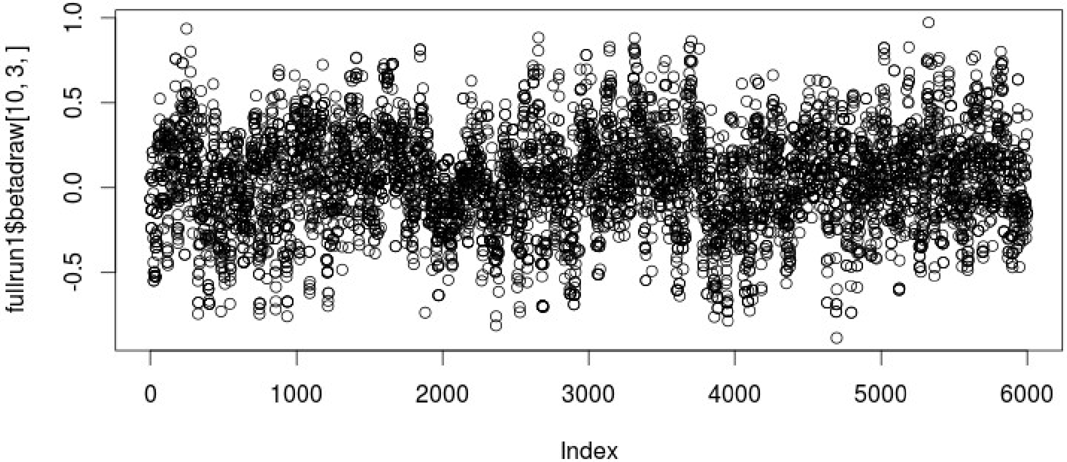
- Each of these objects contains one or more MCMC chains
  - betadraw, for example, is a 360 x 14 x 6,000 array of respondent's beta coefficient chains

Example Choice Model Chains

```
> plot(fullrun1$loglike)
```




```
> plot(fullrun1$betadraw[10,3,])
```



Beta Posteriors

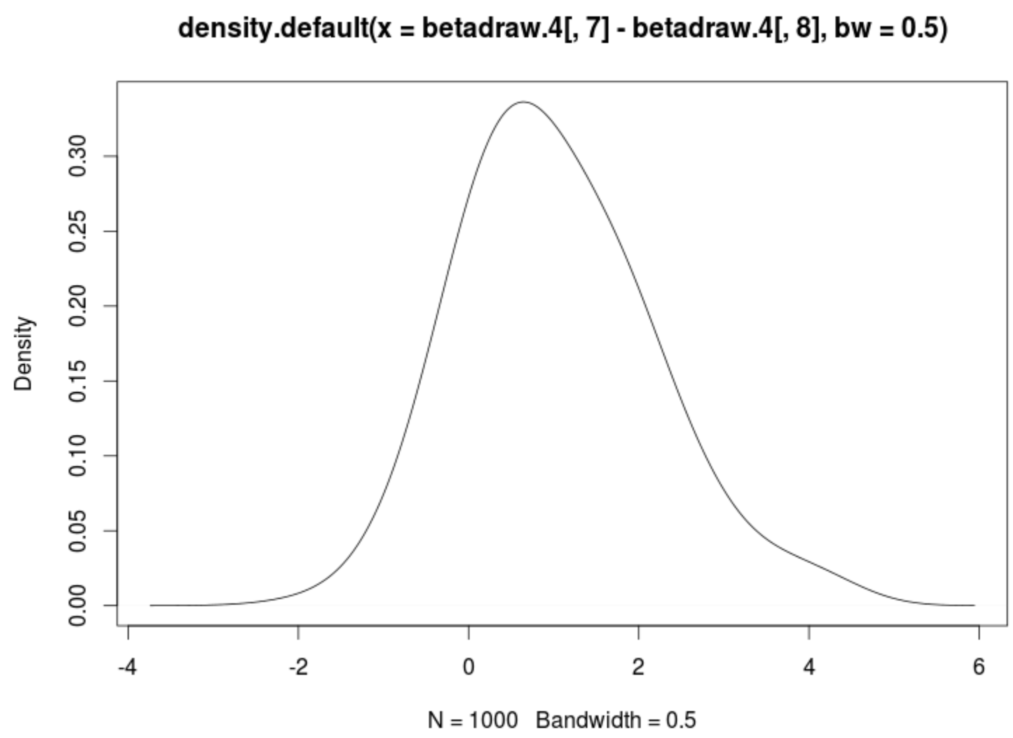
- Means over respondents and last 1,000 draws:

```
> round(betameans,3)
   RAM2v1    RAM3v1    SCR2v1    SCR3v1    CPU2v1    CPU3v1     PR2v1     PR3v1
   -0.244     0.498     0.067     0.585     1.058     1.306     0.275    -3.062

   BRD2v1    BRD3v1    BRD4v1 BRD2v1LPR BRD3v1LPR BRD4v1LPR
   -0.150     0.059    -0.423     0.054     0.082    -0.004
```

- Selected percentiles for respondent 4's betas:

```
> round(apply(betadraw.4,2,quantile,probs=probseq),3)
    RAM2v1 RAM3v1 SCR2v1 SCR3v1 CPU2v1 CPU3v1  PR2v1  PR3v1 BRD2v1 BRD3v1
5%  -1.047 -0.722 -0.182 -0.644 -0.362 -0.217 -0.718 -2.790 -4.908 -0.052
50% -0.461 -0.226  0.314 -0.114  0.164  0.372 -0.108 -1.048 -2.692  1.189
95%  0.121  0.420  0.854  0.464  0.758  0.995  0.436  0.175 -0.846  2.910

    BRD4v1 BRD2v1LPR BRD3v1LPR BRD4v1LPR
5%  -9.925    -1.375    -0.881    -1.261
50% -4.396    -0.402     0.128    -0.303
95% -1.865     0.398     0.979     0.554
```

Respondent 4: A (likely) Difference Between Price Level Betas?




```
> resp.4.ecdf=ecdf(betadraw.4[,7]-betadraw.4[,8])
> resp.4.ecdf(0)
[1] 0.177
```

Beta Covariates

Betas Regressed on prior ownership indicator:

```
> round(zowner.delta,3)
    RAM2v1 RAM3v1 SCR2v1 SCR3v1 CPU2v1 CPU3v1  PR2v1  PR3v1 BRD2v1 BRD3v1
5%  -0.261 -0.415 -0.181 -0.349 -0.207 -0.049 -0.211 -1.305 -0.494  0.862
50%  0.010 -0.152  0.019 -0.093  0.051  0.229  0.024 -0.637 -0.253  1.180
95%  0.283  0.121  0.223  0.154  0.321  0.517  0.245  0.099 -0.012  1.610

    BRD4v1 BRD2v1LPR BRD3v1LPR BRD4v1LPR
5%  -0.748    -0.512    -0.142    -0.100
50% -0.407    -0.195     0.140     0.194
95% -0.112     0.086     0.410     0.515
```

Betas Regressed on gender indicator:

```
> round(gender.delta,3)
    RAM2v1 RAM3v1 SCR2v1 SCR3v1 CPU2v1 CPU3v1  PR2v1  PR3v1 BRD2v1 BRD3v1
5%  -0.084 -0.016 -0.064 -0.002 -0.092 -0.097 -0.030 -0.720 -0.175 -0.241
50%  0.095  0.167  0.082  0.174  0.092  0.112  0.124 -0.254 -0.011 -0.056
95%  0.276  0.351  0.217  0.344  0.283  0.323  0.276  0.202  0.145  0.143

    BRD4v1 BRD2v1LPR BRD3v1LPR BRD4v1LPR
5%   0.395    -0.205    -0.329    -0.251
50%  0.585    -0.014    -0.150    -0.043
95%  0.787     0.174     0.022     0.134
```