

Bayesian linear regression

Jingchen (Monika) Hu

Vassar College

MATH 347 Bayesian Statistics

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS

Review: the Normal model

- When you have continuous outcomes, you can use a Normal model:

$$Y_i \mid \mu, \sigma \stackrel{i.i.d.}{\sim} \text{Normal}(\mu, \sigma), \quad i = 1, \dots, n. \quad (1)$$

- This model assumes each observation follows the same Normal density with mean μ and standard deviation σ .
- Suppose now you have another continuous variable available, x_i . And you want to use the information in x_i to learn about Y_i .
 - Y_i is the log of expenditure of CU's
 - x_i is the log of total income of CU's
- Is the model in Equation (1) flexible to include x_i ?

An observation specific mean

- We can adjust the model in Equation (1) to Equation (2), where the common mean μ is replaced by an observation specific mean μ_i :

$$Y_i \mid \mu_i, \sigma \stackrel{\text{ind}}{\sim} \text{Normal}(\mu_i, \sigma), \quad i = 1, \dots, n. \quad (2)$$

- How to link μ_i and x_i ?

Linear relationship between the mean and the predictor

- One basic approach: use a linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n. \quad (3)$$

- x_i 's are known constants.
- β_0 and β_1 are unknown parameters.
- Interpretation:
 - 1 the linear function $\beta_0 + \beta_1 x_i$ is the **expected outcome** with x_i
 - 2 β_0 is the **intercept**: then **expected outcome** when $x_i = 0$
 - 3 β_1 is the **slope**: the increase in the **expected outcome** when x_i increases by 1 unit

Linear relationship between the mean and the predictor

- One basic approach: use a linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n. \quad (3)$$

- x_i 's are known constants.
- β_0 and β_1 are unknown parameters.
- Interpretation:
 - 1 the linear function $\beta_0 + \beta_1 x_i$ is the **expected outcome** with x_i
 - 2 β_0 is the **intercept**: then **expected outcome** when $x_i = 0$
 - 3 β_1 is the **slope**: the increase in the **expected outcome** when x_i increases by 1 unit
- Bayesian approach:

Linear relationship between the mean and the predictor

- One basic approach: use a linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n. \quad (3)$$

- x_i 's are known constants.
- β_0 and β_1 are unknown parameters.
- Interpretation:
 - 1 the linear function $\beta_0 + \beta_1 x_i$ is the **expected outcome** with x_i
 - 2 β_0 is the **intercept**: then **expected outcome** when $x_i = 0$
 - 3 β_1 is the **slope**: the increase in the **expected outcome** when x_i increases by 1 unit
- Bayesian approach:
 - 1 assign a prior distribution to $(\beta_0, \beta_1, \sigma)$
 - 2 perform inference
 - 3 summarize posterior distribution of these parameters

The simple linear regression model

- To put everything together, a linear regression model:

$$Y_i \mid x_i, \beta_0, \beta_1, \sigma \stackrel{\text{ind}}{\sim} \text{Normal}(\beta_0 + \beta_1 x_i, \sigma), \quad i = 1, \dots, n. \quad (4)$$

- Alternatively:

$$Y_i = \mu_i + \epsilon_i, \quad (5)$$

$$\mu_i = \beta_0 + \beta_1 x_i, \quad (6)$$

$$\epsilon_i \stackrel{\text{i.i.d.}}{\sim} \text{Normal}(0, \sigma) \quad i = 1, \dots, n. \quad (7)$$

- What assumptions does this model make?

The simple linear regression model

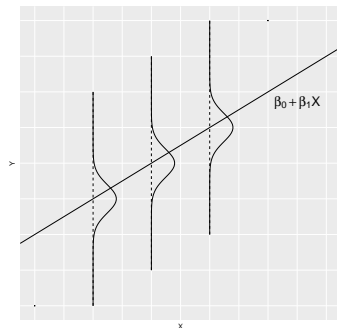


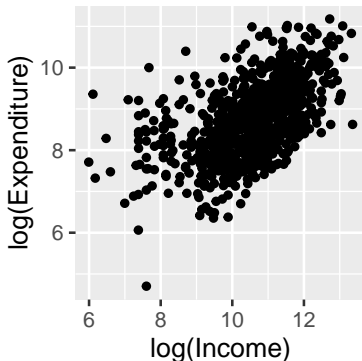
Figure 1: Display of linear regression model. The line represents the unknown regression line $\beta_0 + \beta_1 x$ and the normal curves represent the distribution of the response Y about the line.

The simple linear regression model cont'd

```

CEData <- read.csv("CEsample.csv", header = T, sep = ",")
g1 <- ggplot(CEData, aes(x = log_TotalIncome, y = log_TotalExp)) +
  geom_point(size=1) +
  labs(x = "log(Income)", y = "log(Expenditure)") +
  theme_grey(base_size = 10, base_family = "")
g1

```



Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS

The CE sample

The CE sample comes from the 2017 Q1 CE PUMD: 4 variables, 994 observations.

Variable	Description
log(Expenditure)	Continuous; CU's total expenditures in last quarter (log)
log(Income)	Continuous; the amount of CU income before taxes in past 12 months (log)
Rural	Binary; the urban/rural status of CU: 0 = Urban, 1 = Rural
Race	Categorical; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS

A SLR for the CE sample

- For now, we focus on a simple linear regression:

$$Y_i \mid \mu_i, \sigma \stackrel{\text{ind}}{\sim} \text{Normal}(\mu_i, \sigma), \quad (8)$$

$$\mu_i = \beta_0 + \beta_1 x_i. \quad (9)$$

Variable	Description
log(Expenditure)	Continuous; CU's total expenditures in last quarter (log)
log(Income)	Continuous; the amount of CU income before taxes in past 12 months (log)

- Remarks:

- 1 Y_i is log(Expenditure), and x_i is log(Income).
- 2 The intercept β_0 : the **expected** log(Expenditure) μ_i for a CU i that has zero log(Income) (i.e. $x_i = 0$).
- 3 The slope β_1 : the change in the **expected** log(Expenditure) μ_i when the log(Income) of CU i increases by 1 unit.'

A weakly informative prior

- Sometimes one has limited prior information about the regression parameters β_0 and β_1 and/or the standard deviation σ .
- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0, \beta_1)\pi(\sigma). \quad (10)$$

A weakly informative prior

- Sometimes one has limited prior information about the regression parameters β_0 and β_1 and/or the standard deviation σ .
- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0, \beta_1)\pi(\sigma). \quad (10)$$

- 1 Assuming independence between β_0 and β_1 :

$$\pi(\beta_0, \beta_1) = \pi(\beta_0)\pi(\beta_1), \quad (11)$$

$$\beta_0 \sim \text{Normal}(\mu_0, s_0), \quad (12)$$

$$\beta_1 \sim \text{Normal}(\mu_1, s_1). \quad (13)$$

e.g. $\text{Normal}(0, 100)$.

A weakly informative prior

- Sometimes one has limited prior information about the regression parameters β_0 and β_1 and/or the standard deviation σ .
- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0, \beta_1)\pi(\sigma). \quad (10)$$

- 1 Assuming independence between β_0 and β_1 :

$$\pi(\beta_0, \beta_1) = \pi(\beta_0)\pi(\beta_1), \quad (11)$$

$$\beta_0 \sim \text{Normal}(\mu_0, s_0), \quad (12)$$

$$\beta_1 \sim \text{Normal}(\mu_1, s_1). \quad (13)$$

e.g. $\text{Normal}(0, 100)$.

- 2 Assigning a weakly informative prior for the standard deviation σ :

$$1/\sigma^2 \sim \text{Gamma}(a, b). \quad (14)$$

e.g. $\text{Gamma}(1, 1)$.

Full conditional derivation?

- The sampling model:

$$Y_i \mid x_i, \beta_0, \beta_1, \sigma \stackrel{ind}{\sim} \text{Normal}(\beta_0 + \beta_1 x_i, \sigma). \quad (15)$$

- The joint likelihood:

$$\begin{aligned} L(\beta_0, \beta_1, \sigma) &= \prod_{i=1}^n \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2} \right\} \right] \\ &\propto (1/\sigma^2)^{n/2} \exp \left\{ -\frac{1/\sigma^2}{2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \right\} \end{aligned}$$

Full conditional derivation? cont'd

- The joint posterior:

$$\begin{aligned}\pi(\beta_0, \beta_1, 1/\sigma^2 | y) &\propto (1/\sigma^2)^{n/2} \exp \left\{ -\frac{1/\sigma^2}{2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \right\} \\ &\times \exp \left\{ -\frac{1}{2s_0^2} (\beta_0 - \mu_0)^2 \right\} \exp \left\{ -\frac{1}{2s_1^2} (\beta_1 - \mu_1)^2 \right\} \\ &\times (1/\sigma^2)^{a-1} \exp(-b(1/\sigma^2))\end{aligned}$$

- While it is possible to further simplify and recognize full conditional posterior distribution, we will rely on JAGS to perform the MCMC for us.

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS

JAGS script for the SLR model

```
modelString <-"
model {
  ## sampling
  for (i in 1:N){
    y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
  }

  ## priors
  beta0 ~ dnorm(mu0, g0)
  beta1 ~ dnorm(mu1, g1)
  invsigma2 ~ dgamma(a, b)
  sigma <- sqrt(pow(invsigma2, -1))
}
"
```

JAGS script for the SLR model cont'd

- Pass the data and hyperparameter values to JAGS:

```

y <- as.vector(CEData$log_TotalExp)
x <- as.vector(CEData$log_TotalIncome)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 0.0001,
                 "mu1" = 0, "g1" = 0.0001,
                 "a" = 1, "b" = 1)

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}

```

JAGS script for the SLR model cont'd

- Run the JAGS code for this model:

```
posterior <- run.jags(modelString,  
                      n.chains = 1,  
                      data = the_data,  
                      monitor = c("beta0", "beta1", "sigma"),  
                      adapt = 1000,  
                      burnin = 5000,  
                      sample = 5000,  
                      thin = 1,  
                      inits = initsfunction)
```


JAGS output for the SLR model

- Obtain posterior summaries of all parameters:

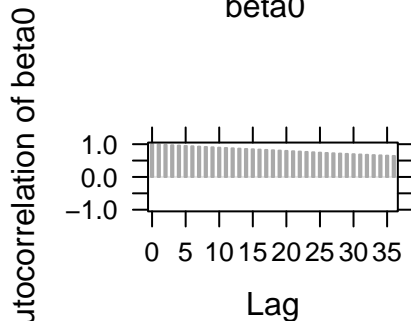
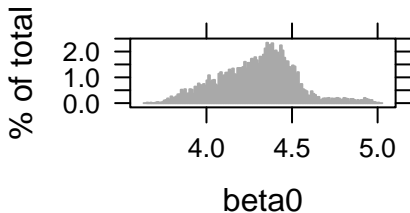
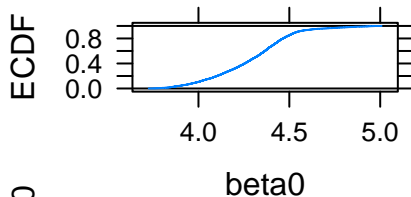
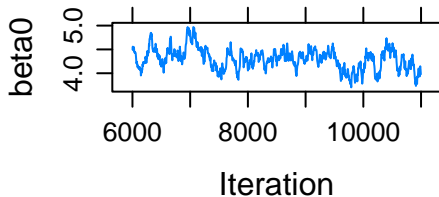
```
summary(posterior)
```

```
##           Lower95      Median Upper95      Mean      SD Mode      MCerr
## beta0  3.771420  4.3136450  4.655760  4.2947731  0.22275382   NA  0.0421194486
## beta1  0.388552  0.4218125  0.471380  0.4237176  0.02091679   NA  0.0039634093
## sigma  0.694060  0.7249680  0.757081  0.7251875  0.01624723   NA  0.0002175128
##           MC%ofSD SSeff      AC.10 psrf
## beta0      18.9      28 0.89361268   NA
## beta1      18.9      28 0.89351502   NA
## sigma       1.3    5579 0.02058294   NA
```

JAGS output for the SLR model cont'd

```
plot(posterior, vars = "beta0")
```

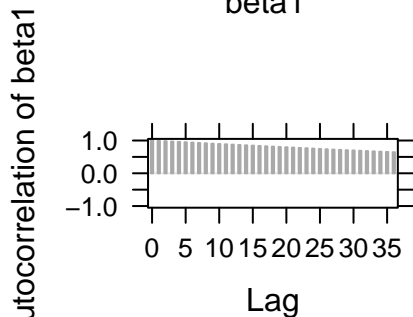
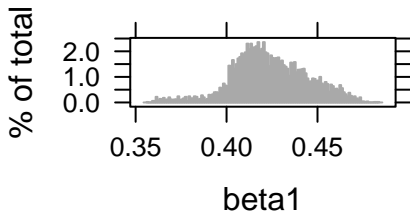
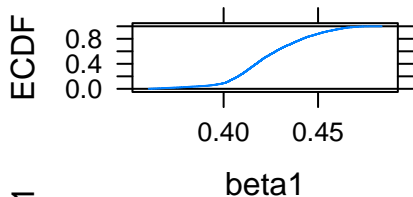
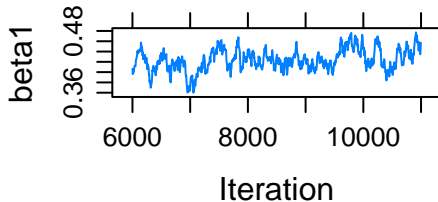
```
## Generating plots...
```



JAGS output for the SLR model cont'd

```
plot(posterior, vars = "beta1")
```

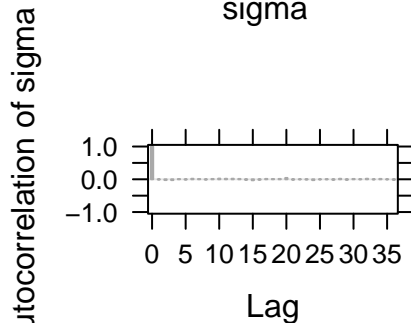
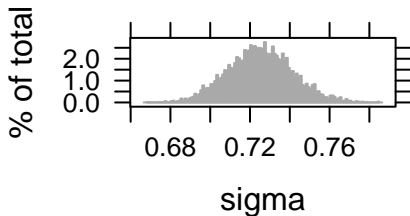
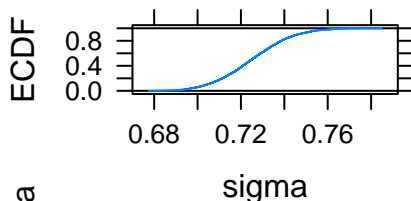
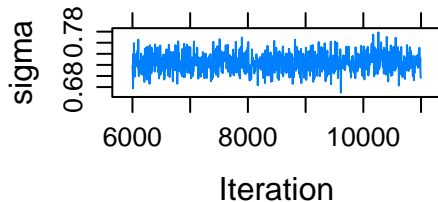
```
## Generating plots...
```



JAGS output for the SLR model cont'd

```
plot(posterior, vars = "sigma")
```

```
## Generating plots...
```



New JAGS script for the SLR model

Setting `thin = 50`, to get rid of the stickiness in β_0 and β_1 .

```
posterior_new <- run.jags(modelString,  
  n.chains = 1,  
  data = the_data,  
  monitor = c("beta0", "beta1", "sigma"),  
  adapt = 1000,  
  burnin = 5000,  
  sample = 5000,  
  thin = 50,  
  inits = initsfunction)
```

New JAGS output for the SLR model

- Obtain posterior summaries of all parameters:

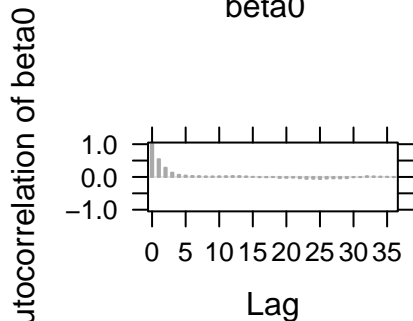
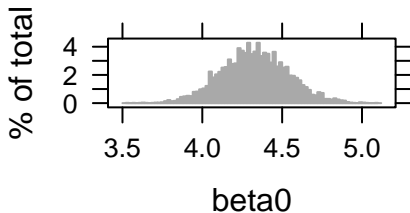
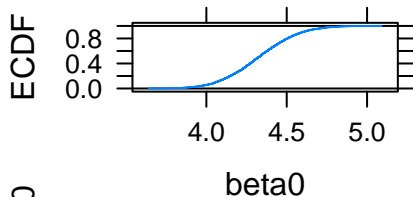
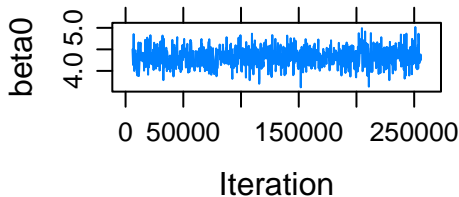
```
summary(posterior_new)
```

##	Lower95	Median	Upper95	Mean	SD	Mode	MCerr
## beta0	3.926200	4.3251800	4.753090	4.3269816	0.21092686	NA	0.0054707075
## beta1	0.381057	0.4208825	0.458968	0.4207384	0.01977946	NA	0.0004930936
## sigma	0.693623	0.7254665	0.757397	0.7255765	0.01624891	NA	0.0002297942
##	MC%ofSD	SSEff	AC.500	psrf			
## beta0	2.6	1487	0.018912301	NA			
## beta1	2.5	1609	0.018555972	NA			
## sigma	1.4	5000	0.003076245	NA			

New JAGS output for the SLR model cont'd

```
plot(posterior_new, vars = "beta0")
```

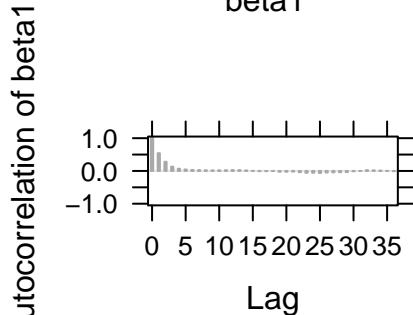
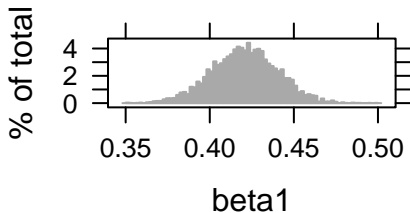
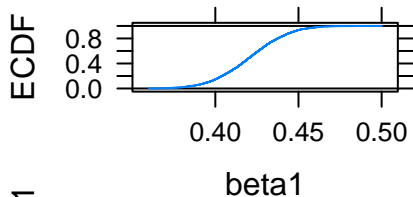
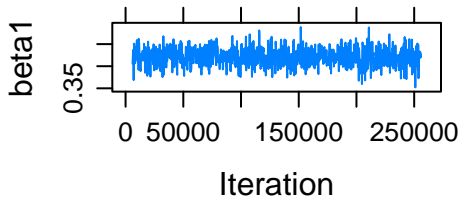
```
## Generating plots...
```



New JAGS output for the SLR model cont'd

```
plot(posterior_new, vars = "beta1")
```

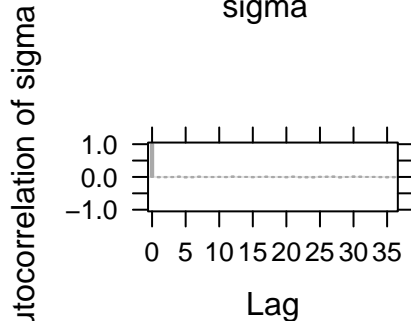
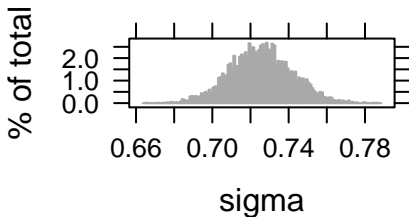
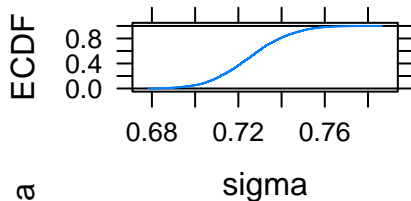
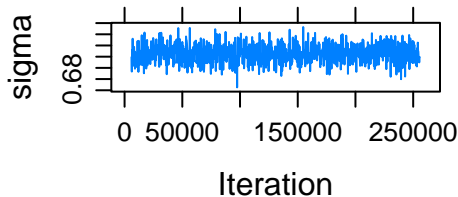
```
## Generating plots...
```



New JAGS output for the SLR model cont'd

```
plot(posterior_new, vars = "sigma")
```

```
## Generating plots...
```



Interpretation of regression coefficients

- The intercept β_0 :

For a CU with $\log(\text{Income})$ of \$0 (i.e. Income of \$1), the expected $\log(\text{Expenditure})$ is \$4.327 (the median), and it falls in the interval (\$3.926, \$4.753) with 90% posterior probability.

Interpretation of regression coefficients

- The intercept β_0 :

For a CU with $\log(\text{Income})$ of \$0 (i.e. Income of \$1), the expected $\log(\text{Expenditure})$ is \$4.327 (the median), and it falls in the interval (\$3.926, \$4.753) with 90% posterior probability.

- The slope β_1 :

With every \$1 increase in the $\log(\text{Income})$ of a CU, its $\log(\text{Expenditure})$ increases by \$0.421 (the median). In addition, this increase in the $\log(\text{Expenditure})$ falls in the interval (\$0.381, \$0.459) with 90% posterior probability.

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR**
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS

Simulate fits from the regression model

- The SLR model assumes

$$E(Y) = \beta_0 + \beta_1 x. \quad (16)$$

- Each pair of values (β_0, β_1) corresponds to a line $\beta_0 + \beta_1 x$ in the space of values of x and y .

Simulate fits from the regression model

- The SLR model assumes

$$E(Y) = \beta_0 + \beta_1 x. \quad (16)$$

- Each pair of values (β_0, β_1) corresponds to a line $\beta_0 + \beta_1 x$ in the space of values of x and y .
- Using posterior mean $\tilde{\beta}_0$ and $\tilde{\beta}_1$, one can find a “best” line of fit through the data

$$y = \tilde{\beta}_0 + \tilde{\beta}_1 x. \quad (17)$$

Simulate fits from the regression model

- The SLR model assumes

$$E(Y) = \beta_0 + \beta_1 x. \quad (16)$$

- Each pair of values (β_0, β_1) corresponds to a line $\beta_0 + \beta_1 x$ in the space of values of x and y .
- Using posterior mean $\tilde{\beta}_0$ and $\tilde{\beta}_1$, one can find a “best” line of fit through the data

$$y = \tilde{\beta}_0 + \tilde{\beta}_1 x. \quad (17)$$

- What about the uncertainty of the line estimate?

Simulate fits from the regression model cont'd

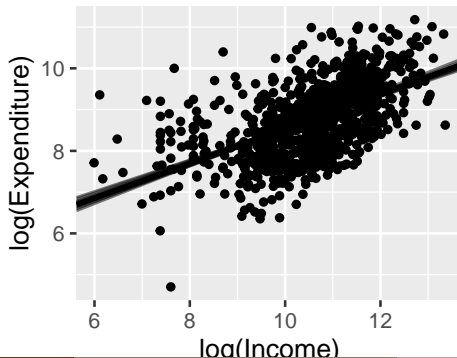
- To learn about the uncertainty of the line estimate, one can draw a sample of J rows from the matrix of posterior draws of (β_0, β_1) and collect the line estimates

$$\tilde{\beta}_0^{(j)} + \tilde{\beta}_1^{(j)} x, \quad j = 1, \dots, J. \quad (18)$$

```
post <- as.mcmc(posterior_new)
post_means <- apply(post, 2, mean)
post <- as.data.frame(post)
```


Simulate fits from the regression model cont'd

```
ggplot(CEData, aes(log_TotalIncome, log_TotalExp)) +
  geom_point(size=1) +
  geom_abline(data=post[1:10, ],
             aes(intercept=beta0, slope=beta1), alpha = 0.5) +
  geom_abline(intercept = post_means[1],
             slope = post_means[2], size = 1) +
  ylab("log(Expenditure)") + xlab("log(Income)") +
  theme_grey(base_size = 10, base_family = "")
```



Learning about the expected response

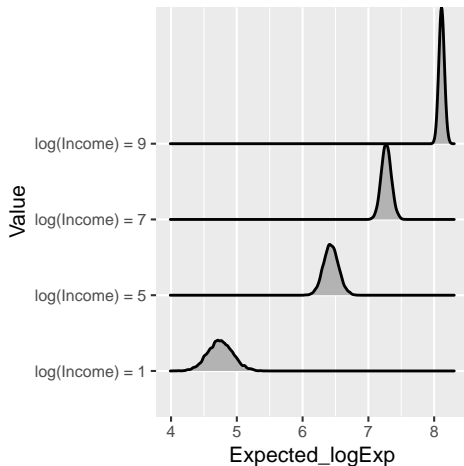
- What if one wants to learn about the expected log expenditure of a CU for a specific log income value?
- One could obtain a simulated sample from the posterior of $\beta_0 + \beta_1 x$ by computing $E(Y) = \beta_0 + \beta_1 x$ on each of the simulated pairs from the posterior of (β_0, β_1) .

```
post <- as.data.frame(post)
one_expected <- function(x){
  lp <- post[ , "beta0"] + x * post[ , "beta1"]
  data.frame(Value = paste("log(Income) =", x),
             Expected_logExp = lp)
}

df <- map_df(c(1, 5, 7, 9), one_expected)
```

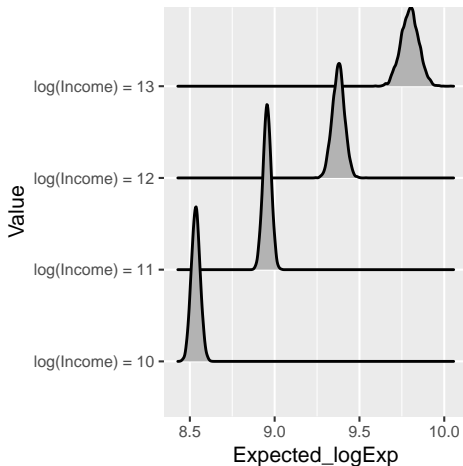
Learning about the expected response cont'd

```
require(ggribes)
ggplot(df, aes(x = Expected_logExp, y = Value)) +
  geom_density_ridges() +
  theme_grey(base_size = 8, base_family = "")
```



Learning about the expected response cont'd

```
df <- map_df(c(10, 11, 12, 13), one_expected)
ggplot(df, aes(x = Expected_logExp, y = Value)) +
  geom_density_ridges() +
  theme_grey(base_size = 8, base_family = "")
```



Learning about the expected response cont'd

```
df %>% group_by(Value) %>%
  summarize(P05 = quantile(Expected_logExp, 0.05),
            P50 = median(Expected_logExp),
            P95 = quantile(Expected_logExp, 0.95))
```

```
## # A tibble: 4 x 4
##   Value          P05    P50    P95
##   <chr>        <dbl> <dbl> <dbl>
## 1 log(Income) = 10  8.49  8.53  8.58
## 2 log(Income) = 11  8.91  8.96  8.99
## 3 log(Income) = 12  9.32  9.38  9.43
## 4 log(Income) = 13  9.71  9.80  9.88
```

e.g. for a CU of $\log(\text{Income})$ of \$5, the posterior median of the expected $\log(\text{Expenditure})$ is \$6.43, and the probability that the expected $\log(\text{Expenditure})$ falls between \$6.25 and \$6.62 is 90%.

Prediction of future responses

- To predict future $\log(\text{Expenditure})$ for a CU given its $\log(\text{Income})$:

$$Y_i \mid x_i, \beta_0, \beta_1, \sigma \stackrel{\text{ind}}{\sim} \text{Normal}(\beta_0 + \beta_1 x_i, \sigma). \quad (19)$$

- variability in β_0 and β_1 in the expected value $\beta_0 + \beta_1 x$
- variability in the sampling model in Equation (19)

Prediction of future responses

- To predict future $\log(\text{Expenditure})$ for a CU given its $\log(\text{Income})$:

$$Y_i \mid x_i, \beta_0, \beta_1, \sigma \stackrel{\text{ind}}{\sim} \text{Normal}(\beta_0 + \beta_1 x_i, \sigma). \quad (19)$$

- variability in β_0 and β_1 in the expected value $\beta_0 + \beta_1 x$
- variability in the sampling model in Equation (19)
- For a large number of S :

simulate $E[y]^{(1)} = \beta_0^{(1)} + \beta_1^{(1)} x \rightarrow$ sample $\tilde{y}^{(1)} \sim \text{Normal}(E[y]^{(1)}, \sigma^{(1)})$

simulate $E[y]^{(2)} = \beta_0^{(2)} + \beta_1^{(2)} x \rightarrow$ sample $\tilde{y}^{(2)} \sim \text{Normal}(E[y]^{(2)}, \sigma^{(2)})$

\vdots

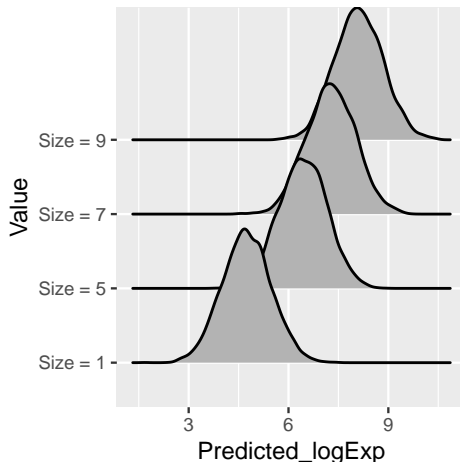
simulate $E[y]^{(S)} = \beta_0^{(S)} + \beta_1^{(S)} x \rightarrow$ sample $\tilde{y}^{(S)} \sim \text{Normal}(E[y]^{(S)}, \sigma^{(S)})$

Prediction of future responses cont'd

```
one_predicted <- function(x){  
  lp <- post[ , "beta0"] + x * post[ , "beta1"]  
  y <- rnorm(5000, lp, post[ , "sigma"])  
  data.frame(Value = paste("Size =", x),  
             Predicted_logExp = y)  
}  
df <- map_df(c(1, 5, 7, 9), one_predicted)
```


Prediction of future responses cont'd

```
require(ggribes)
ggplot(df, aes(x = Predicted_logExp, y = Value)) +
  geom_density_ridges() +
  theme_grey(base_size = 9, base_family = "")
```



Prediction of future responses cont'd

```
df %>% group_by(Value) %>%
  summarize(P05 = quantile(Predicted_logExp, 0.05),
            P50 = median(Predicted_logExp),
            P95 = quantile(Predicted_logExp, 0.95))
```

```
## # A tibble: 4 x 4
##   Value      P05    P50    P95
##   <chr>    <dbl> <dbl> <dbl>
## 1 Size = 1  3.54  4.73  5.97
## 2 Size = 5  5.25  6.43  7.62
## 3 Size = 7  6.05  7.28  8.48
## 4 Size = 9  6.91  8.10  9.34
```

- Recall that for a CU of $\log(\text{Income})$ of \$5, the probability that the expected $\log(\text{Expenditure})$ falls between \$6.25 and \$6.62 is 90%.
- A 90% prediction interval for the $\log(\text{Expenditure})$ is (\$5.24, \$7.61), which is wider than the posterior interval estimate for the expected $\log(\text{Expenditure})$.
- The predictive distribution incorporates the sizable uncertainty in the $\log(\text{Expenditure})$ given the $\log(\text{Income})$ represented by the sampling standard deviation σ .

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors**
- 7 A multiple linear regression, and MCMC simulation by JAGS

Subjective prior: standardization

- To put different variables on similar scales.

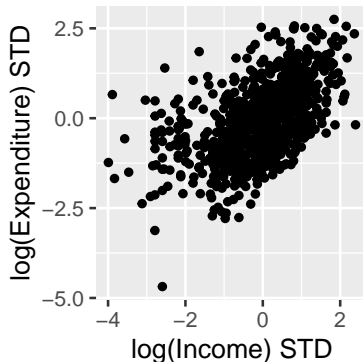
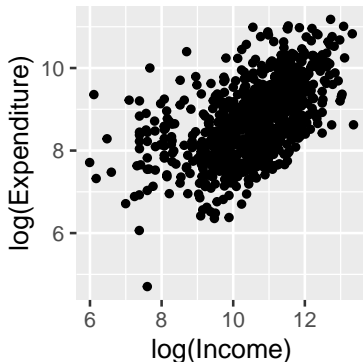
$$y_i^* = \frac{y_i - \bar{y}}{s_y}, x_i^* = \frac{x_i - \bar{x}}{s_x} \quad (20)$$

- Use the `scale` command to standardize.

```
CEData$log_TotalExpSTD <- scale(CEData$log_TotalExp)
CEData$log_TotalIncomeSTD <- scale(CEData$log_TotalIncome)
```

Subjective prior: standardization cont'd

```
g2 = ggplot(CEDData, aes(x = log_TotalIncomeSTD, y = log_TotalExpSTD)) +
  geom_point(size=1) +
  xlab("log(Income) STD") + ylab("log(Expenditure) STD") +
  theme_grey(base_size = 10, base_family = "")
grid.arrange(g1, g2, ncol=2)
```



Subjective prior: SLR model after standardization

- A standardized value represents the number of standard deviations that the value falls above or below the mean.
- What does $x_i^* = -2$ mean? What does $y_i^* = 1$ mean?

Subjective prior: SLR model after standardization

- A standardized value represents the number of standard deviations that the value falls above or below the mean.
- What does $x_i^* = -2$ mean? What does $y_i^* = 1$ mean?
- The SLR model after standardization:

$$Y_i^* \mid \mu_i^*, \sigma \stackrel{\text{ind}}{\sim} \text{Normal}(\mu_i^*, \sigma), \quad (21)$$

$$\mu_i^* = \beta_0 + \beta_1 x_i^*. \quad (22)$$

- 1 Y_i^* is standardized log(Expenditure), and x_i^* is standardized log(Income).
- 2 The intercept β_0 : the **expected** standardized log(Expenditure) μ_i^* for a CU i that has the **average** log(Income) (i.e. $x_i^* = 0$).
- 3 The slope β_1 : the change in the **expected** standardized log(Expenditure) μ_i^* when the standardized log(Income) x_i^* of CU i increases by 1 unit, or when the log(Income) variable increases by one standard deviation.
- 4 The slope β_1 equals to the correlation between x_i^* and y_i^* : positive β_1 vs negative β_1 , absolute value indicates the strength.

Subjective prior: a subjective prior

- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \quad (23)$$

Subjective prior: a subjective prior

- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \quad (23)$$

- Prior on the intercept β_0 :

- 1 If one believes a CU of average $\log(\text{Income})$ will also have an average $\log(\text{Expenditure})$.

$$\beta_0 \sim \text{Normal}(0, s_0). \quad (24)$$

- 2 The standard deviation s_0 reflects how confident the person believes in the guess of $\beta_0 = 0$. e.g. $\text{Normal}(0, 1)$.

Subjective prior: a subjective prior

- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \quad (23)$$

- Prior on the intercept β_0 :

- If one believes a CU of average $\log(\text{Income})$ will also have an average $\log(\text{Expenditure})$.

$$\beta_0 \sim \text{Normal}(0, s_0). \quad (24)$$

- The standard deviation s_0 reflects how confident the person believes in the guess of $\beta_0 = 0$. e.g. $\text{Normal}(0, 1)$.

- Prior on the slope β_1 :

- The slope β_1 represents the correlation between the predictor and the response.

$$\beta_1 \sim \text{Normal}(\mu_1, s_1). \quad (25)$$

- μ_1 represents one's best guess of the correlation, and s_1 represents the sureness of this guess. e.g. $\text{Normal}(0.7, 0.15)$.

Subjective prior: a subjective prior cont'd

- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \quad (26)$$

Subjective prior: a subjective prior cont'd

- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \quad (26)$$

- Prior on the standard deviation σ : weakly informative

$$1/\sigma^2 \sim \text{Gamma}(1, 1). \quad (27)$$

Subjective prior: a subjective prior cont'd

- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \quad (26)$$

- Prior on the standard deviation σ : weakly informative

$$1/\sigma^2 \sim \text{Gamma}(1, 1). \quad (27)$$

- The informative/subjective prior for $(\beta_0, \beta_1, \sigma)$ is defined as

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma), \quad (28)$$

$$\beta_0 \sim \text{Normal}(0, 1), \quad (29)$$

$$\beta_1 \sim \text{Normal}(0.7, 0.15), \quad (30)$$

$$1/\sigma^2 \sim \text{Gamma}(1, 1). \quad (31)$$

Subjective prior: JAGS script for the standardized SLR model

```
modelString <-"
model {
  ## sampling
  for (i in 1:N){
    y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
  }

  ## priors
  beta0 ~ dnorm(mu0, g0)
  beta1 ~ dnorm(mu1, g1)
  invsigma2 ~ dgamma(a, b)
  sigma <- sqrt(pow(invsigma2, -1))
}
"
```

Subjective prior: JAGS script for the standardized SLR model cont'd

- Pass the data and hyperparameter values to JAGS:

```

y <- as.vector(CEData$log_TotalExpSTD)
x <- as.vector(CEData$log_TotalIncomeSTD)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 1,
                 "mu1" = 0.7, "g1" = 44.4,
                 "a" = 1, "b" = 1)

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}

```

JAGS script for the SLR model cont'd

- Run the JAGS code for this model:

```
posterior_sub <- run.jags(modelString,  
  n.chains = 1,  
  data = the_data,  
  monitor = c("beta0", "beta1", "sigma"),  
  adapt = 1000,  
  burnin = 5000,  
  sample = 5000,  
  thin = 1,  
  inits = initsfunction)
```


Subjective prior: JAGS output for the SLR model

- Obtain posterior summaries of all parameters:

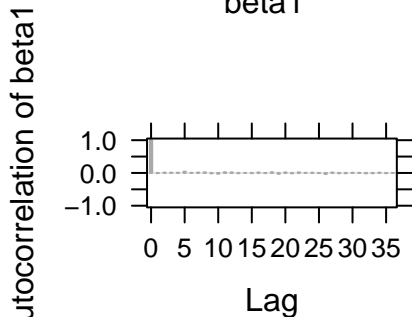
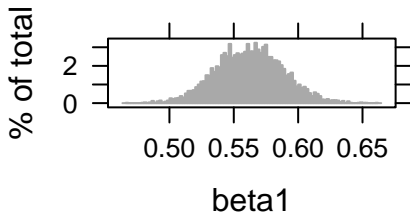
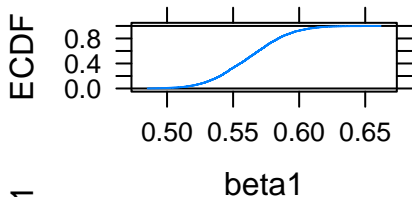
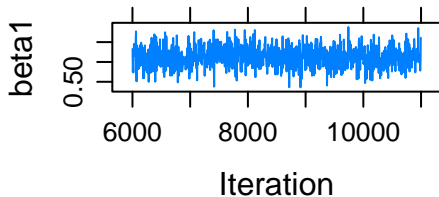
```
summary(posterior_sub)
```

##	Lower95	Median	Upper95	Mean	SD	Mode
## beta0	-0.0484962	-0.0000908726	0.0544673	-5.334241e-05	0.02637737	NA
## beta1	0.5131890	0.5623200000	0.6144910	5.622705e-01	0.02616385	NA
## sigma	0.7958380	0.8314665000	0.8684720	8.318869e-01	0.01866642	NA
##	MCerr	MC%ofSD	SSEff	AC.10	psrf	
## beta0	0.0003561888	1.4	5484	-0.02064139	NA	
## beta1	0.0003700127	1.4	5000	-0.02451218	NA	
## sigma	0.0002502895	1.3	5562	0.02083092	NA	

Subjective prior: JAGS output for the SLR model con'td

```
plot(posterior_sub, vars = "beta1")
```

```
## Generating plots...
```



Conditional means prior: a conditional means prior

- We have seen two methods for constructing a prior on the regression coefficient parameters:
 - ① Weakly informative/vague prior on a model on the original data
 - ② Subjective/informative prior on a model on standardized data

Conditional means prior: a conditional means prior

- We have seen two methods for constructing a prior on the regression coefficient parameters:
 - ① Weakly informative/vague prior on a model on the original data
 - ② Subjective/informative prior on a model on standardized data
- A third approach:
 - ① On a model on the original data
 - ② Stating prior beliefs about the expected response value conditional on specific values of the predictor variable

Conditional means prior: a conditional means prior

- We have seen two methods for constructing a prior on the regression coefficient parameters:
 - 1 Weakly informative/vague prior on a model on the original data
 - 2 Subjective/informative prior on a model on standardized data
- A third approach:
 - 1 On a model on the original data
 - 2 Stating prior beliefs about the expected response value conditional on specific values of the predictor variable
- Assuming independence:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0, \beta_1)\pi(\sigma). \quad (32)$$

- The linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i. \quad (33)$$

Conditional means prior: a conditional means prior cont'd

- The linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i. \quad (34)$$

- Easier to formulate prior opinion about the mean values, μ_i
- For predictor value x_1 , one can construct a Normal prior for the mean value μ_1 :

$$\mu_1 \sim \text{Normal}(m_1, s_1) \quad (35)$$

e.g. if $x_1 = 10$, the mean $\mu_1 = \beta_0 + \beta_1(10) \sim \text{Normal}(8, 2)$

Conditional means prior: a conditional means prior cont'd

- The linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i. \quad (34)$$

- Easier to formulate prior opinion about the mean values, μ_i
- For predictor value x_1 , one can construct a Normal prior for the mean value μ_1 :

$$\mu_1 \sim \text{Normal}(m_1, s_1) \quad (35)$$

e.g. if $x_1 = 10$, the mean $\mu_1 = \beta_0 + \beta_1(10) \sim \text{Normal}(8, 2)$

- Similarly, for predictor value x_2 , one can construct a Normal prior for the mean value μ_2 :

$$\mu_2 \sim \text{Normal}(m_2, s_2) \quad (36)$$

e.g. if $x_2 = 15$, the mean $\mu_2 = \beta_0 + \beta_1(12) \sim \text{Normal}(11, 2)$

Conditional means prior: a conditional means prior cont'd

- Assuming independence:

$$\pi(\mu_1, \mu_2) = \pi(\mu_1)\pi(\mu_2) \quad (37)$$

- One can then solve β_0 and β_1 in $\mu_i = \beta_0 + \beta_1 x_i$ given μ_1, μ_2, x_1, x_2 :

$$\beta_1 = \frac{\mu_2 - \mu_1}{x_2 - x_1}, \quad (38)$$

$$\beta_0 = \mu_1 - x_1 \left(\frac{\mu_2 - \mu_1}{x_2 - x_1} \right). \quad (39)$$

Conditional means prior: a conditional means prior cont'd

- Assuming independence:

$$\pi(\mu_1, \mu_2) = \pi(\mu_1)\pi(\mu_2) \quad (37)$$

- One can then solve β_0 and β_1 in $\mu_i = \beta_0 + \beta_1 x_i$ given μ_1, μ_2, x_1, x_2 :

$$\beta_1 = \frac{\mu_2 - \mu_1}{x_2 - x_1}, \quad (38)$$

$$\beta_0 = \mu_1 - x_1 \left(\frac{\mu_2 - \mu_1}{x_2 - x_1} \right). \quad (39)$$

- Currently, we have $x_1 = 10, x_2 = 12$, and

$$\mu_1 = \beta_0 + \beta_1(10) \sim \text{Normal}(8, 2), \quad (40)$$

$$\mu_2 = \beta_0 + \beta_1(12) \sim \text{Normal}(11, 2). \quad (41)$$

Conditional means prior: JAGS script

```
modelString <-"
model {
  ## sampling
  for (i in 1:N){
    y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
  }

  ## priors
  beta1 <- (mu2 - mu1)/(x2 - x1)
  beta0 <- mu1 - x1*(mu2 - mu1)/(x2 - x1)
  mu1 ~ dnorm(m1, g1)
  mu2 ~ dnorm(m2, g2)
  invsigma2 ~ dgamma(a, b)
  sigma <- sqrt(pow(invsigma2, -1))
}
"
```

Outline

- 1 Introduction: Adding a continuous predictor variable
- 2 The CE sample
- 3 A simple linear regression for the CE sample
- 4 MCMC simulation by JAGS for the SLR model
- 5 Bayesian inferences with SLR
- 6 More on priors
- 7 A multiple linear regression, and MCMC simulation by JAGS**

The CE sample

The CE sample comes from the 2017 Q1 CE PUMD. \ 4 variables, 994 observations.

Variable	Description
log(Expenditure)	Continuous; CU's total expenditures in last quarter (log)
log(Income)	Continuous; the amount of CU income before taxes in past 12 months (log)
Rural	Binary; the urban/rural status of CU: 0 = Urban, 1 = Rural
Race	Categorical; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race

How can we include additional information about the urban/rural status and race category to predict a CU's log(Expenditure)?

A multiple linear regression model

- Similar to SLR, MLR assumes an observation specific mean μ_i for Y_i :
$$Y_i \mid \mu_i, \sigma \stackrel{ind}{\sim} \text{Normal}(\mu_i, \sigma), \quad i = 1, \dots, n. \quad (42)$$

A multiple linear regression model

- Similar to SLR, MLR assumes an observation specific mean μ_i for Y_i :

$$Y_i \mid \mu_i, \sigma \stackrel{ind}{\sim} \text{Normal}(\mu_i, \sigma), \quad i = 1, \dots, n. \quad (42)$$

- In addition, MLR assumes the mean of Y_i is a linear function of **all** predictors:

$$\mu_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_r x_{i,r}. \quad (43)$$

- ▶ $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,r})$ is a vector of **known** predictors for observation i
- ▶ $\beta = (\beta_0, \beta_1, \dots, \beta_r)$ is a vector of **unknown** regression coefficient parameters (shared among all observations)

Regression coefficient interpretation

$$\mu_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_r x_{i,r}. \quad (44)$$

- When all r predictors are continuous,
 - ▶ What does β_0 mean?
 - ▶ What does each $\beta_j, j = 1, \cdots, r$ mean?

Regression coefficient interpretation

$$\mu_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_r x_{i,r}. \quad (44)$$

- When all r predictors are continuous,
 - ▶ What does β_0 mean?
 - ▶ What does each $\beta_j, j = 1, \dots, r$ mean?
- In the CE example, the predictors are not all continuous.

Variable	Description
log(Income)	Continuous ; the amount of CU income before taxes in past 12 months (log)
Rural	Binary ; the urban/rural status of CU: 0 = Urban, 1 = Rural
Race	Categorical ; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race

Adding a binary predictor

Variable	Description
$\log(\text{Expenditure})$	Continuous; CU's total expenditures in last quarter (log)
Rural	Binary ; the urban/rural status of CU: 0 = Urban, 1 = Rural

- While it is possible to consider Rural as a continuous variable: change by one unit from urban to rural. . .
- It is much more common to consider it as a binary categorical variable to classify two groups:
 - ▶ The urban group
 - ▶ The rural group
- Such classification puts an emphasis on the **difference of the expected outcomes** between the two groups.

With only one binary predictor

- For simplicity, consider a simplified regression model with a single predictor: the binary indicator for rural area x_i .

$$\mu_i = \beta_0 + \beta_1 x_i = \begin{cases} \beta_0, & \text{the urban group;} \\ \beta_0 + \beta_1, & \text{the rural group.} \end{cases} \quad (45)$$

- The expected outcome μ_i for CUs in the urban group: β_0 .
- The expected outcome μ_i for CUs in the rural group: $\beta_0 + \beta_1$.
- β_1 represents the **change in the expected outcome** μ_i from the urban group to the rural group.

Adding a multi-category categorical predictor

Variable	Description
$\log(\text{Expenditure})$	Continuous; CU's total expenditures in last quarter (log)
Race	Categorical ; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race

- It is common to consider it as a categorical variable to classify multiple groups:
 - ▶ How many groups? What are the groups?
- Such classification puts an emphasis on the **difference of the expected outcomes** between one group to **the reference group**.

With only one categorical predictor

- For simplicity, consider a simplified regression model with a single predictor: the race category of the reference person x_i .

$$\begin{aligned}\mu_i &= \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \beta_4 x_{i,4} + \beta_5 x_{i,5} \\ &= \begin{cases} \beta_0, & \text{White;} \\ \beta_0 + \beta_1, & \text{Black;} \\ \beta_0 + \beta_2, & \text{Native American;} \\ \beta_0 + \beta_3, & \text{Asian;} \\ \beta_0 + \beta_4, & \text{Pacific Islander;} \\ \beta_0 + \beta_5, & \text{Multi-race.} \end{cases} \end{aligned} \quad (46)$$

- What is the expected outcome μ_i for CUs in the White group?
- What is the expected outcome μ_i for CUs in the Asian group?
- What does β_5 represent?

Consider all predictors

- The linear combination of the log(income), rural indicator, and race category:

$$\begin{aligned}\mu_i^* = & \beta_0 + \beta_1 x_{i,income}^* + \beta_2 x_{i,rural} + \beta_3 x_{i,race_B} \\ & + \beta_4 x_{i,race_N} + \beta_5 x_{i,race_A} + \beta_6 x_{i,race_P} + \beta_7 x_{i,race_M}.\end{aligned}\tag{47}$$

- The MLR is written as

$$\begin{aligned}Y_i^* \mid \beta_0, \beta_1, \dots, \beta_7, \sigma, \mathbf{x}_i^* & \overset{ind}{\sim} \text{Normal}(\beta_0 + \beta_1 x_{i,income}^* + \beta_2 x_{i,rural} \\ & + \beta_3 x_{i,race_B} + \beta_4 x_{i,race_N} \\ & + \beta_5 x_{i,race_A} + \beta_6 x_{i,race_P} \\ & + \beta_7 x_{i,race_M}, \sigma).\end{aligned}\tag{48}$$

Note: * indicates standardized values.

Consider all predictors cont'd

$$\begin{aligned}\mu_i^* = & \beta_0 + \beta_1 x_{i,income}^* + \beta_2 x_{i,rural} + \beta_3 x_{i,race_B} \\ & + \beta_4 x_{i,race_N} + \beta_5 x_{i,race_A} + \beta_6 x_{i,race_P} + \beta_7 x_{i,race_M}.\end{aligned}\tag{49}$$

- What does intercept β_0 mean?
- What does regression coefficient β_1 mean?
- What about the expected outcome of a CU with Native American reference person, living in rural area with standardized $\log(\text{income}) = 1$?

A weakly informative prior

- With more than one predictors, the subjective prior and the conditional means prior are more difficult to specify. Why?
- Let's try giving a weakly informative prior.

A weakly informative prior

- With more than one predictors, the subjective prior and the conditional means prior are more difficult to specify. Why?
- Let's try giving a weakly informative prior.
- Assuming independence:

$$\pi(\beta_0, \beta_1, \dots, \beta_7, \sigma) = \pi(\beta_0, \beta_1, \dots, \beta_7)\pi(\sigma). \quad (50)$$

- ▶ Assuming independence among β_j 's:

$$\pi(\beta_0, \beta_1, \dots, \beta_7) = \prod_{j=0}^7 \pi(\beta_j), \quad (51)$$

$$\beta_j \sim \text{Normal}(\mu_j, s_j). \quad (52)$$

e.g. $\text{Normal}(0, 1)$.

- ▶ Assigning a weakly informative prior for the standard deviation σ :

$$1/\sigma^2 \sim \text{Gamma}(a, b). \quad (53)$$

e.g. $\text{Gamma}(1, 1)$.

JAGS script for the MLR model

- Need to standardize $\log(\text{Expenditure})$ and $\log(\text{Income})$.
- Also, need to create indicator variables (0 or 1) for each category of the categorical variable, except for the reference category.

```

CEData$log_TotalExpSTD <- scale(CEData$log_TotalExp)
CEData$log_TotalIncomeSTD <- scale(CEData$log_TotalIncome)

library(fastDummies)
## create indicator variable for Rural
CEData$Rural = fastDummies::dummy_cols(CEData$UrbanRural)[,names(fastDummies::
  == ".data_2")]

```

- Do this for Race as well, 5 indicator variables.

JAGS script for the MLR model cont'd

```

modelString <-"
model {
  ## sampling
  for (i in 1:N){
    y[i] ~ dnorm(beta0 + beta1*x_income[i] + beta2*x_rural[i] +
    beta3*x_race_B[i] + beta4*x_race_N[i] +
    beta5*x_race_A[i] + beta6*x_race_P[i] +
    beta7*x_race_M[i], invsigma2)
  }
  ## priors
  beta0 ~ dnorm(mu0, g0)
  beta1 ~ dnorm(mu1, g1)
  beta2 ~ dnorm(mu2, g2)
  beta3 ~ dnorm(mu3, g3)
  beta4 ~ dnorm(mu4, g4)
  beta5 ~ dnorm(mu5, g5)
  beta6 ~ dnorm(mu6, g6)
  beta7 ~ dnorm(mu7, g7)
  invsigma2 ~ dgamma(a, b)
  sigma <- sqrt(pow(invsigma2, -1))
}

```

JAGS script for the MLR model cont'd

- Pass the data and hyperparameter values to JAGS:

```

y = as.vector(CEData$log_TotalExpSTD)
x_income = as.vector(CEData$log_TotalIncomeSTD)
x_rural = as.vector(CEData$Rural)
x_race_B = as.vector(CEData$Race_Black)
x_race_N = as.vector(CEData$Race_NA)
x_race_A = as.vector(CEData$Race_Asian)
x_race_P = as.vector(CEData$Race_PI)
x_race_M = as.vector(CEData$Race_M)
N = length(y)  # Compute the number of observations

```

JAGS script for the MLR model cont'd

- Pass the data and hyperparameter values to JAGS:

```
the_data <- list("y" = y, "x_income" = x_income,
  "x_rural" = x_rural, "x_race_B" = x_race_B,
  "x_race_N" = x_race_N, "x_race_A" = x_race_A,
  "x_race_P" = x_race_P, "x_race_M" = x_race_M,
  "N" = N,
  "mu0" = 0, "g0" = 1, "mu1" = 0, "g1" = 1,
  "mu2" = 0, "g2" = 1, "mu3" = 0, "g3" = 1,
  "mu4" = 0, "g4" = 1, "mu5" = 0, "g5" = 1,
  "mu6" = 0, "g6" = 1, "mu7" = 0, "g7" = 1,
  "a" = 1, "b" = 1)
```

JAGS script for the MLR model cont'd

- Pass the data and hyperparameter values to JAGS:

```
initsfunction <- function(chain){  
  .RNG.seed <- c(1,2)[chain]  
  .RNG.name <- c("base::Super-Duper",  
                 "base::Wichmann-Hill")[chain]  
  return(list(.RNG.seed=.RNG.seed,  
              .RNG.name=.RNG.name))  
}
```

JAGS script for the MLR model cont'd

- Run the JAGS code for this model:

```
posterior_MLR <- run.jags(modelString,
  n.chains = 1,
  data = the_data,
  monitor = c("beta0", "beta1", "beta2",
              "beta3", "beta4", "beta5",
              "beta6", "beta7", "sigma"),
  adapt = 1000,
  burnin = 5000,
  sample = 5000,
  thin = 1,
  inits = initsfunction)
```

```
## Warning: Convergence cannot be assessed with only 1 chain
```

JAGS output for the MLR model

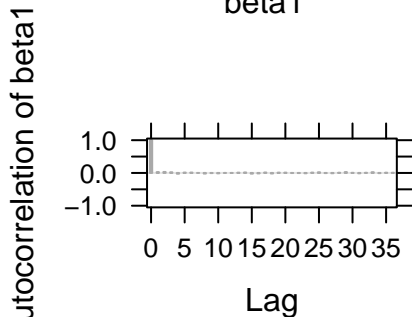
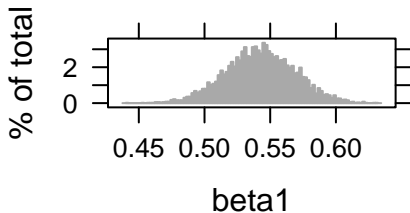
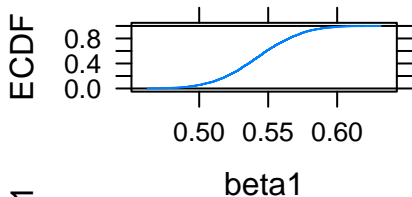
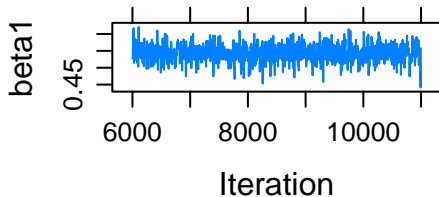
```
summary(posterior_MLR)
```

##		Lower95	Median	Upper95	Mean	SD	Mode
##	beta0	-0.0261884	0.03072615	0.0898729	0.03111402	0.02974211	NA
##	beta1	0.4889750	0.54241750	0.5926740	0.54237419	0.02648363	NA
##	beta2	-0.4918380	-0.26816300	-0.0262582	-0.26872848	0.11800343	NA
##	beta3	-0.3996750	-0.23509350	-0.0678321	-0.23338067	0.08548032	NA
##	beta4	-0.6071250	-0.01422395	0.5520270	-0.01764766	0.29634497	NA
##	beta5	-0.0922392	0.18444550	0.4434330	0.18556012	0.13734992	NA
##	beta6	-0.5323690	0.08998935	0.6842530	0.09247921	0.31739975	NA
##	beta7	-0.3609250	0.03685400	0.4281700	0.03163386	0.20378584	NA
##	sigma	0.7892530	0.82702250	0.8624090	0.82749520	0.01860583	NA
##		MCerr	MC%ofSD	SSeff	AC.10	psrf	
##	beta0	0.0005408114	1.8	3024	-0.004175826	NA	
##	beta1	0.0003934179	1.5	4532	-0.012275215	NA	
##	beta2	0.0018827201	1.6	3928	-0.001199134	NA	
##	beta3	0.0014087613	1.6	3682	-0.006798767	NA	
##	beta4	0.0041909508	1.4	5000	-0.024524053	NA	
##	beta5	0.0020743101	1.5	4384	-0.007574647	NA	
##	beta6	0.0044887103	1.4	5000	0.007878188	NA	
##	beta7	0.0030045388	1.5	4600	0.011216547	NA	
##	sigma	0.0002631261	1.4	5000	0.011646253	NA	

JAGS output for the MLR model cont'd

```
plot(posterior_MLR, vars = "beta1")
```

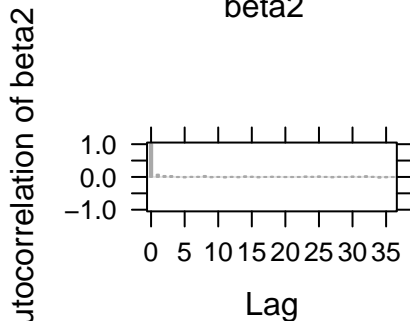
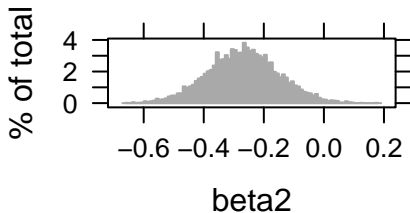
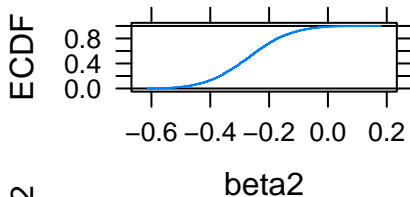
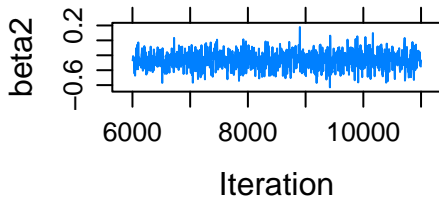
```
## Generating plots...
```



JAGS output for the MLR model cont'd

```
plot(posterior_MLR, vars = "beta2")
```

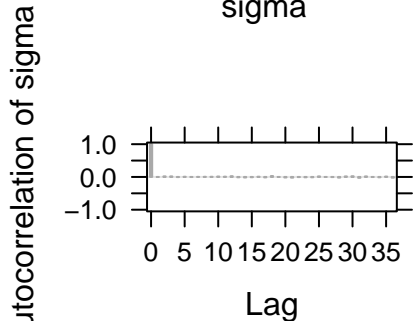
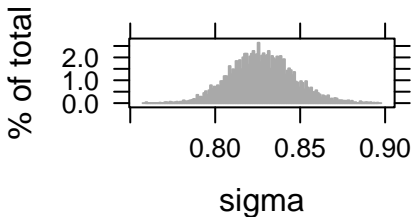
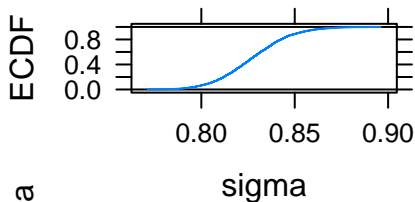
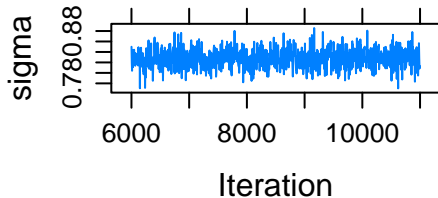
```
## Generating plots...
```



JAGS output for the MLR model cont'd

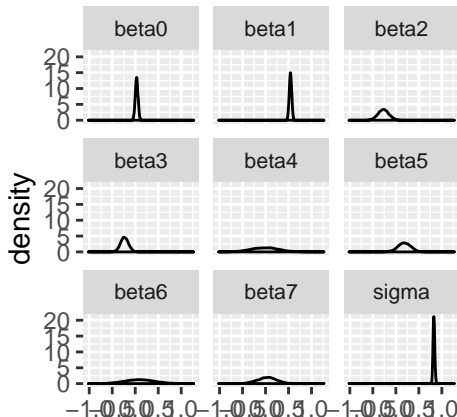
```
plot(posterior_MLR, vars = "sigma")
```

```
## Generating plots...
```



JAGS output for the MLR model cont'd

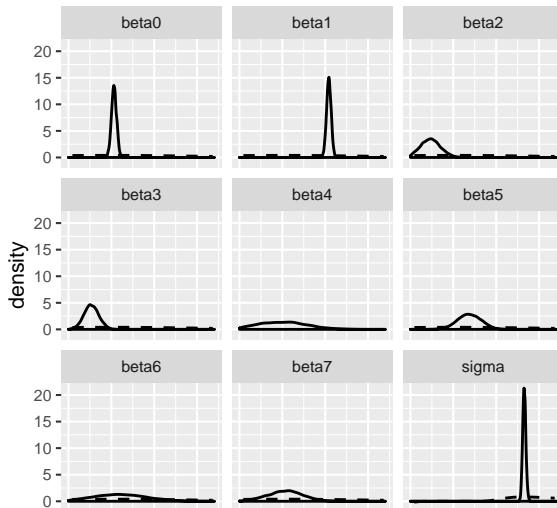
```
post <- as.mcmc(posterior_MLR)
post %>% as.data.frame %>%
  gather(parameter, value) -> post2
ggplot(post2, aes(value)) +
  geom_density() + facet_wrap(~ parameter, ncol = 3) +
  theme(strip.text.x = element_text(size=8))
```



JAGS output for the MLR model cont'd

Warning: Removed 548 rows containing non-finite values (stat_density).

Warning: Removed 72 rows containing missing values (geom_path).



Bayesian inferences with MLR

- Learning about the expected response
- Predictor of future responses
- Posterior predictive checks
- Experiment with several priors to show the impact on the posterior inference