```r
require(devtools)
```

```
## Loading required package: devtools
## Loading required package: usethis
```

```r
devtools::install_github("bayesball/ProbBayes")
```

```
## Skipping install of 'ProbBayes' from a github remote, the SHA1 (805e6220) has not changed since last
##   Use `force = TRUE` to force installation
```

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```r
require(ProbBayes)
```

```
## Loading required package: ProbBayes
## Loading required package: LearnBayes
## Loading required package: shiny
```

```r
require(tidyverse)
```

```
## Loading required package: tidyverse
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
require(runjags)
```

```
## Loading required package: runjags
##
## Attaching package: 'runjags'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
require(coda)
```

```
## Loading required package: coda
```

```r
crcblue <- "#2905a1"
CEData <- read.csv("CEsample.csv", header = T, sep = ",")
```

Lab 5: Conditional means priors in Bayesian linear regression

Author:________(Insert your name here) ________

Total Grade for Lab 5: /15

Comments (optional)

Template for lab report

Instructions: This is the template you will use to type up your responses to the exercises. To produce a document that you can print out and turn in just click on Knit PDF above. All you need to do to complete the lab is to type up your BRIEF answers and the R code (when necessary) in the spaces provided below.

It is strongly recommended that you knit your document regularly (minimally after answering each exercise) for two reasons.

1. Ensure that there are no errors in your code that would prevent the document from knitting.
2. View the instructions and your answers in a more legible, attractive format.

```
# Any text BOTH preceded by a hashtag AND within the ```{r} ``` code chunk is a comment.
# R indicates a comment by turning the text green in the editor, and brown in the knitted
# document.
# Comments are not treated as a command to be interpreted by the computer.
# They normally (briefly!) describe the purpose of your command or chunk in plain English.
# However, for this class, they will have a different goal, as the text above and below
# each chunk should sufficiently describe the chunk's contents.
# For this class, comments will be used to indicate where your code should go, or to give
# hints for what the code should look like.
```

This lab explores the conditional means priors for the regression coefficients in Bayesian linear regression.

The conditional means priors in the CE example

- The linear relationship:

$$\mu_i = \beta_0 + \beta_1 x_i. \tag{1}$$

  - 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:

$$\mu_1 \sim \text{Normal}(m_1, s_1) \tag{2}$$

    e.g. if $x_1 = 10$, the mean $\mu_1 = \beta_0 + \beta_1(1) \sim \text{Normal}(8, 2)$

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

$$\mu_2 \sim \text{Normal}(m_2, s_2) \tag{3}$$

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

- Assuming independence:

$$\pi(\mu_1, \mu_2) = \pi(\mu_1)\pi(\mu_2) \tag{4}$$

- 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:

$$\beta_1 = \frac{\mu_2 - \mu_1}{x_2 - x_1}, \tag{5}$$

$$\beta_0 = \mu_1 - x_1\Big(\frac{\mu_2 - \mu_1}{x_2 - x_1}\Big). \tag{6}$$

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

$$\mu_1 = \beta_0 + \beta_1(10) \sim \text{Normal}(8, 2), \tag{7}$$

$$\mu_2 = \beta_0 + \beta_1(12) \sim \text{Normal}(11, 2). \tag{8}$$

Specifying a conditional means prior in JAGS

The `modelString` sample script for specifying a conditional means prior is given below.

```
## write the model
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))
}
"
```

```
y <- as.vector(CEData$log_TotalExp)
x <- as.vector(CEData$log_TotalIncome)
N <- length(y)
the_data <- list("y"=y, "x"=x, "N"=N,
                 "m1"=8, "g1"=2, "x1"=10,
                 "m2"=11, "g2"=2, "x2"=12,
                 "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))
}
```

Exercise 1: Provide the hyperparameter values in Equation (7) and Equation (8) in terms of `the_data` for the conditional means prior. You can simply write down R script when specifying the list of `the_data` as your answer.

Grade for Exercise 1: /3

Comments:

```
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)
```

Exercise 2: Run the complete JAGS script and perform MCMC diagnostics.

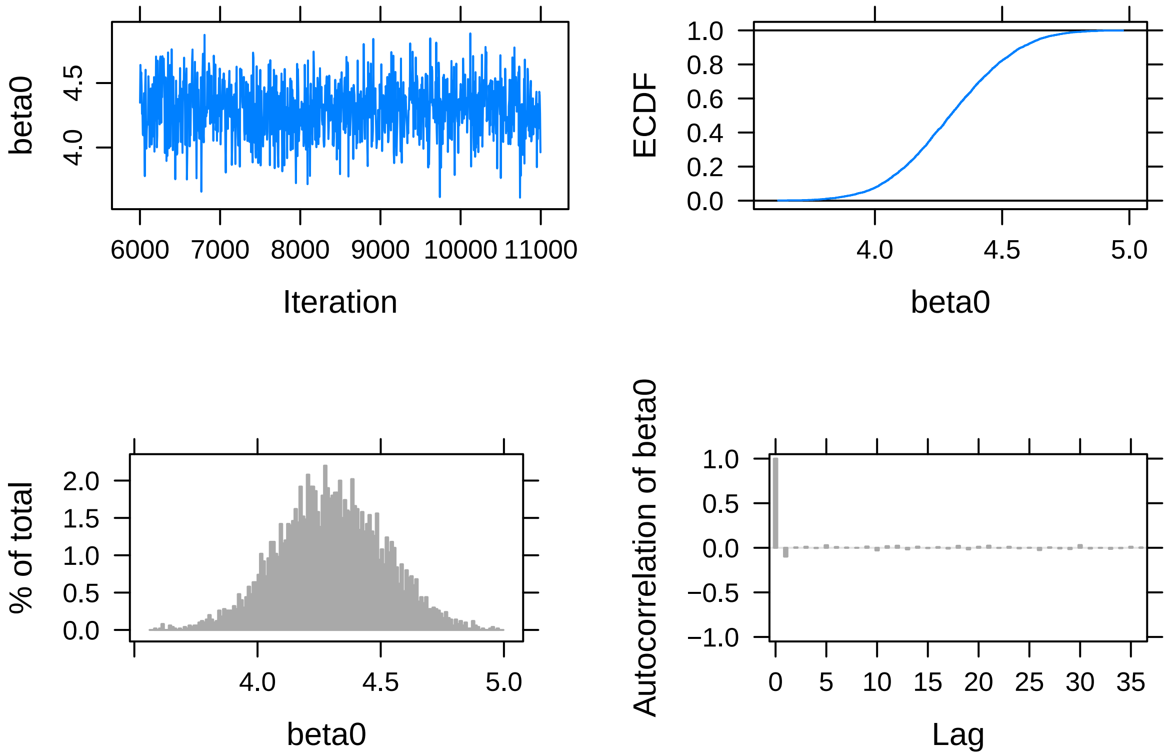
```
## Loading required namespace: rjags

## Compiling rjags model...
## Calling the simulation using the rjags method...
## Note: the model did not require adaptation
## Burning in the model for 5000 iterations...
## Running the model for 5000 iterations...
## Simulation complete
## Calculating summary statistics...

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

## Finished running the simulation
```

```
summary(posterior)
```

```
##         Lower95    Median   Upper95      Mean         SD Mode        MCerr
## beta0 3.8778212 4.2955599 4.7122059 4.2983857 0.21312886   NA 0.0027302981
## beta1 0.3821787 0.4237202 0.4604438 0.4234985 0.02000952   NA 0.0002584769
## sigma 0.6932358 0.7248726 0.7564801 0.7251520 0.01627039   NA 0.0002180821
##       MC%ofSD SSeff       AC.10 psrf
## beta0     1.3  6093 -0.02941145   NA
## beta1     1.3  5993 -0.02842126   NA
## sigma     1.3  5566  0.02043386   NA
```

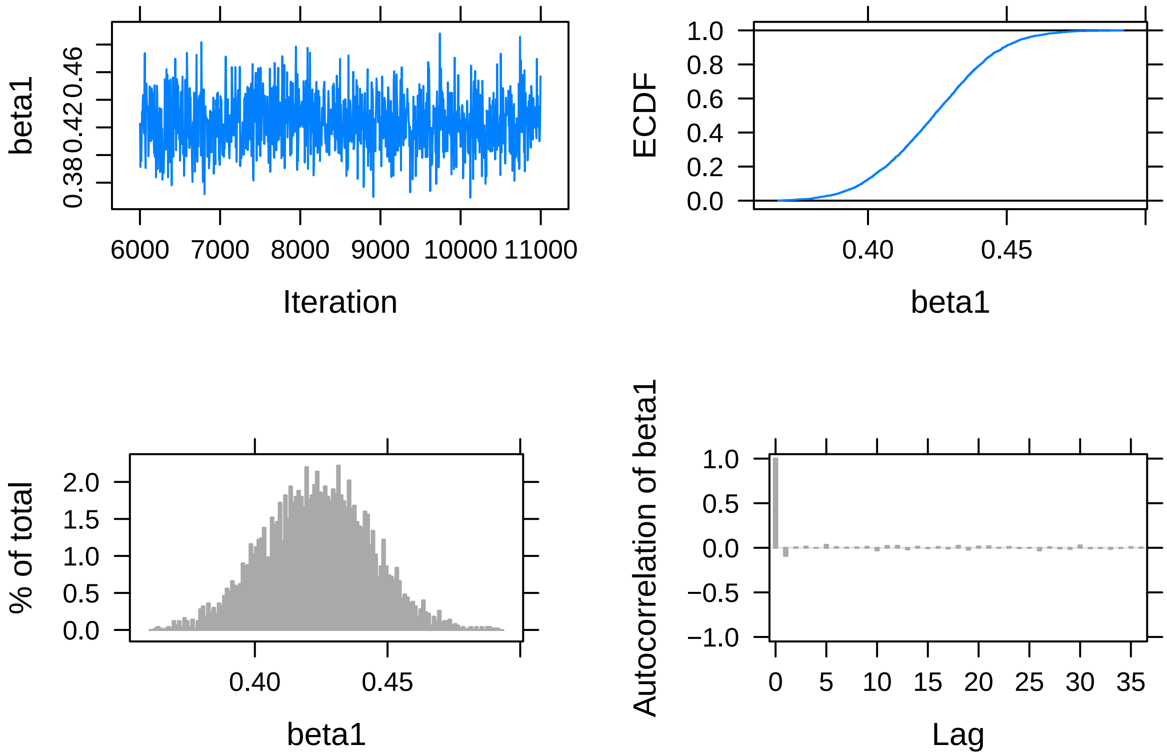
```
plot(posterior, vars = "beta0")
```

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



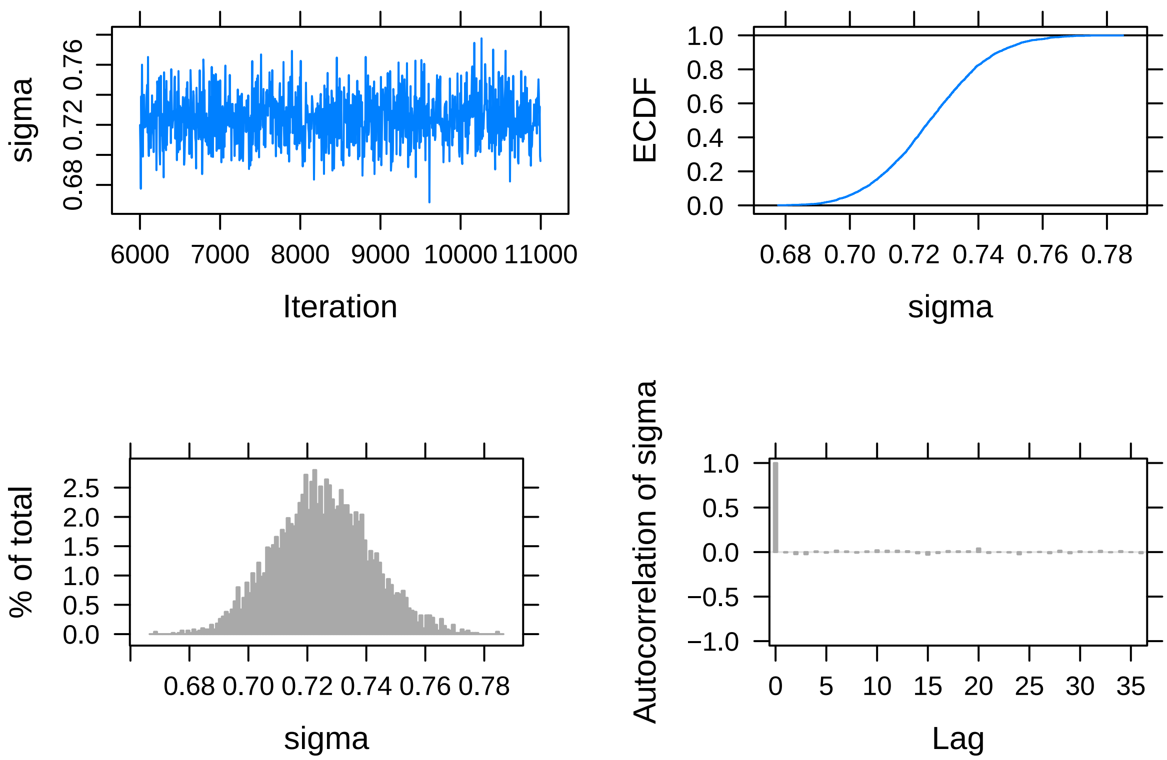

```
plot(posterior, vars = "beta1")
```

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




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

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



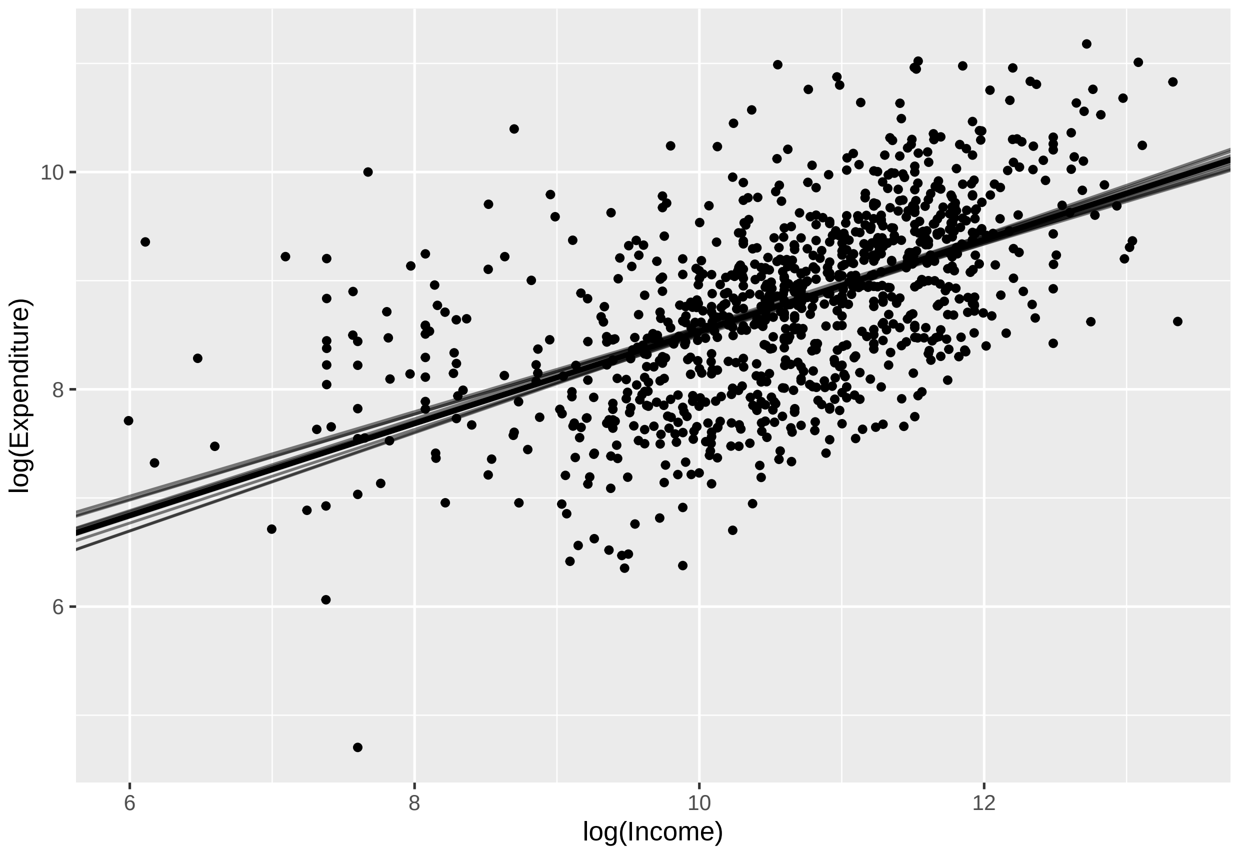
Grade for Exercise 2: /4

Comments:

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

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 = "")
```

Exercise 3: Interpret β_0 and β_1 in the context of the CE example.

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Grade for Exercise 3: /4

Comments:

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

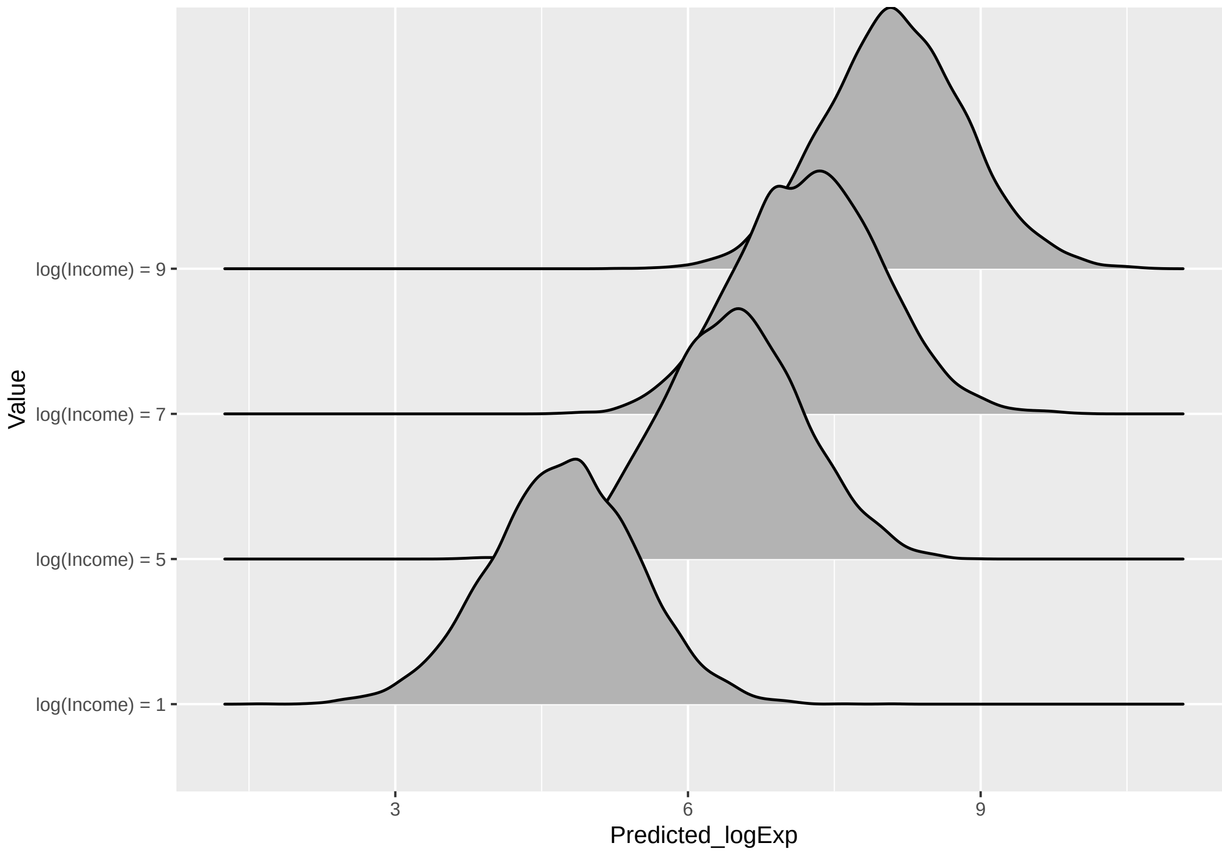
```
require(ggridges)
```

Exercise 4: Use the posterior samples of $(\beta_0, \beta_1, \sigma)$ and produce predicted values of future responses at $x = 1, 5, 7, 9$ and make a plot. What can you say about predicted log(Expenditure) for a CU of \$5 log(Income)? (Hint: check out lecture slides page $39 - 41$.)

```
## Loading required package: ggridges
```

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

```
## Picking joint bandwidth of 0.121
```




```
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 log(Income) = 1   3.46  4.73  5.94
## 2 log(Income) = 5   5.22  6.45  7.65
## 3 log(Income) = 7   6.05  7.25  8.47
## 4 log(Income) = 9   6.93  8.12  9.31
```

Grade for Exercise 4: /4

Comments: