Homework 9

Shao-Ting Chiu (UIN:433002162)

11/17/22

Table of contents

Computational Environment

```
using Pkg
Pkg.activate("hw9")
using Distributions
using DataFrames
using Plots
using DelimitedFiles
using LinearAlgebra
using Statistics
using ProtoStructs
using CSV
import Random
Random.seed!(2022)
```

```
  Activating project at `~/Documents/GitHub/STAT638_Applied-Bayes-Methods/hw/hw9`
```

```
Random.TaskLocalRNG()
```

Description

- Course: STAT638, 2022 Fall
- Deadline: 2022/11/17, 12:00 pm

Read Chapter 9 in the Hoff book. Then do Problems 9.1 and 9.2 in Hoff.

For both regression models, please include an intercept term (β_0).

In 9.1(b), please replace "max" by "min". (This is not listed in the official book errata, but appears to be a typo.)

For 9.2, the azdiabetes.dat data are described in *Exercise* 6 of Chapter 7 (see errata).

Problem 9.1

Extrapolation: The file `swim.dat` contains data on the amount of time in seconds, it takes each of four high school swimmers to swim 50 yards. Each swimmer has 6 times, taken on a biweekly basis.

(a)

Perform the following data analysis for each swimmer separately:

1. Fit a linear regression model of swimming time as the response and week as the explanatory variable. To formulate your prior, use the information that competitive times for this age group generally range from 22 to 24 seconds.
2. For each swimmer j, obtain a posterior predictive distribution for Y_j^*, their time if they were to swim 2 weeks from the last recorded time.

- Suppose a linear model

$$Y = X\beta + \epsilon$$

$$Y_i = x_{i,1}\beta_1 + x_{i,2}\beta_2 + \epsilon_i$$

- $Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_6 \end{bmatrix}$. A swimmer's record of 6. Series in time
- $X = \begin{bmatrix} x_{1,1} & x_{1,2} \\ \vdots & \vdots \\ x_{6,1} & x_{6,2} \end{bmatrix}$

  - $x_{j,1}$: jth record with swim score in the range of 22 to 24 second
  - $x_{j,2}$: Weeks of training

- $\beta = \begin{bmatrix}\beta_1 \\ \beta_2\end{bmatrix}$.

  - $\mu_0 = \begin{bmatrix}23 \\ 0\end{bmatrix}$
    * The prior expectation of intercept of y is 23.
  - $\beta_0 \sim N_p(\mu_0, \Sigma_0)$.
    1. FCD: $\beta|y, \sigma^2 \sim N_p(\beta_n, \Sigma_n)$
    2. $\Sigma_n^{-1} = \Sigma_0^{-1} + \frac{X^T X}{\sigma^2}$
    3. $\beta_n = \Sigma_n(\Sigma_0^{-1}\beta_0 + \frac{X^T y}{\sigma^2})$

Prior setting

- $\Sigma_0 = \begin{bmatrix}0.1 & 0 \\ 0 & 0.1\end{bmatrix}$
  - There is uncertainty about β estimation.
  - Covariance of time and intersept is believe as 0
- $\sigma^2 \sim IG(\nu_0/2, \nu_0\sigma_0^2/2)$
  - FCD: $\sigma^2|y, \beta \sim IG((\nu_0 + n)/2, (\nu_0\sigma_0^2) + SSR(\beta)/2)$
- $SSR(\beta) = (y - X\beta)^T(y - X\beta)$

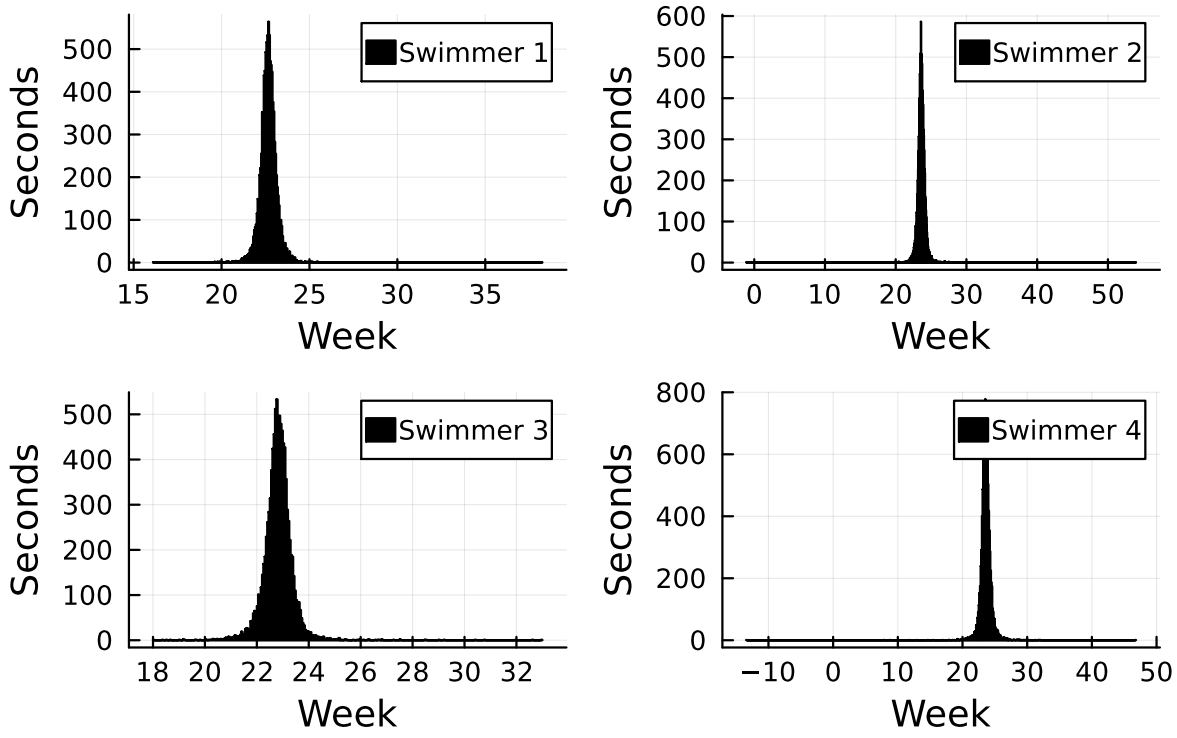
```
ys = readdlm("data/swim.dat")
```

```
4×6 Matrix{Float64}:
 23.1  23.2  22.9  22.9  22.8  22.7
 23.2  23.1  23.4  23.5  23.5  23.4
 22.7  22.6  22.8  22.8  22.9  22.8
 23.7  23.6  23.7  23.5  23.5  23.4
```

```
"""
Problem 9.1 (a)
"""

@proto struct SwimmingModel
    S = 1000 # Number of sampling
    # Data
```

```
    y
    n = length(y) # number of records
    # Model
    X = hcat( ones(n), collect(0:2:10) )
    p = size(X)[2]
    # Prior
      = MvNormal([23., 0.], [0.1 0; 0 0.1])
      = 1.
     ² = 0.2
end


function SSR( , y, X)
    ssrV = (y - X* )' * (y - X* )
    return sum(ssrV)
end

function  _FCD( ², m::SwimmingModel)
    Σ  =( m.  .Σ^-1 + m.X' * m.X /  ²)^-1
      = Σ *(m.  .Σ^-1 * m.  .  + m.X' * m.y /  ²)
    return MvNormal(vec( ), Hermitian(Σ ))
end

function  ²_FCD( , m::SwimmingModel)
     = (m.   + m.n)/2
     = (m.  *m.  ²) + SSR( , m.y, m.X)
    return InverseGamma( ,  )
end

function pred(X, m::SwimmingModel)
    # Sampling vector
     smp = zeros(m.S, length(m.  . ))
     ²smp = zeros(m.S)
    y = zeros(m.S)
    # Init
     smp[1,:] = rand(m.  )
     ²smp[1] = m.  ²
    y[1] = m.y[1]
    for i in 2:m.S
         smp[i,:] = rand( _FCD( ²smp[i-1], m))
         ²smp[i] = rand( ²_FCD( smp[i-1,:], m))
```

```
        # Predict
        y[i] =  smp[i,:]' * X + rand(Normal(0.,  ²smp[i]))
    end

    return (y=y,  = smp,  ²= ²smp)
end

j_swim = 1
ms = [ SwimmingModel(y = hcat(ys[i,:]), S=10000 ) for i in 1:size(ys)[1] ]
ys_pred = zeros(size(ys)[1], ms[1].S)
X_pred = [1,12]

for i in eachindex(ms)
    ys_pred[i,:] = pred([1,12], ms[i]).y
end

## Plotting
p = [histogram(ys_pred[i,:], label="Swimmer $i", color="black",
    xlabel="Week", ylabel="Seconds"
    ) for i in 1:size(ys)[1]]
plot(p...)
```



(b)

The coach of the team has to decide which of the four swimmers will compete in a swimming meet in 2 weeks. Using your predictive distributions, compute $Pr(Y_j^* = \max\{Y_1^*, \ldots, Y_4^*\}|Y)$ for each swimmer j, and based on this make a recommendation to the coach.

```
am = argmax(ys_pred, dims=1)

y_count = zeros(1, size(ys)[1])

for a in am
    y_count[a[1]] += 1
end

pmax = vec(y_count ./ length(am))

## Recommendation
ds = DataFrame( Dict("Swimmer"=> collect(1:size(ys)[1]),"Pr(Y_i is max)" => pmax ))
```

	Pr(Y_i is max)	Swimmer
	Float64	Int64
1	0.023	1
2	0.4773	2
3	0.0424	3
4	0.4573	4

```
Swimmer 2 is the most probable winner.
```

Problem 9.2

Model selection: As described in Example 6 of Chapter 7, the file `azdiabetes.dat` contains data on health-related variables of a population of 532 women. In this exercise we will be modeling the conditional distribution of glucose level (`glu`) as a linear combination of the other variables, excluding the variable `diabetes`.

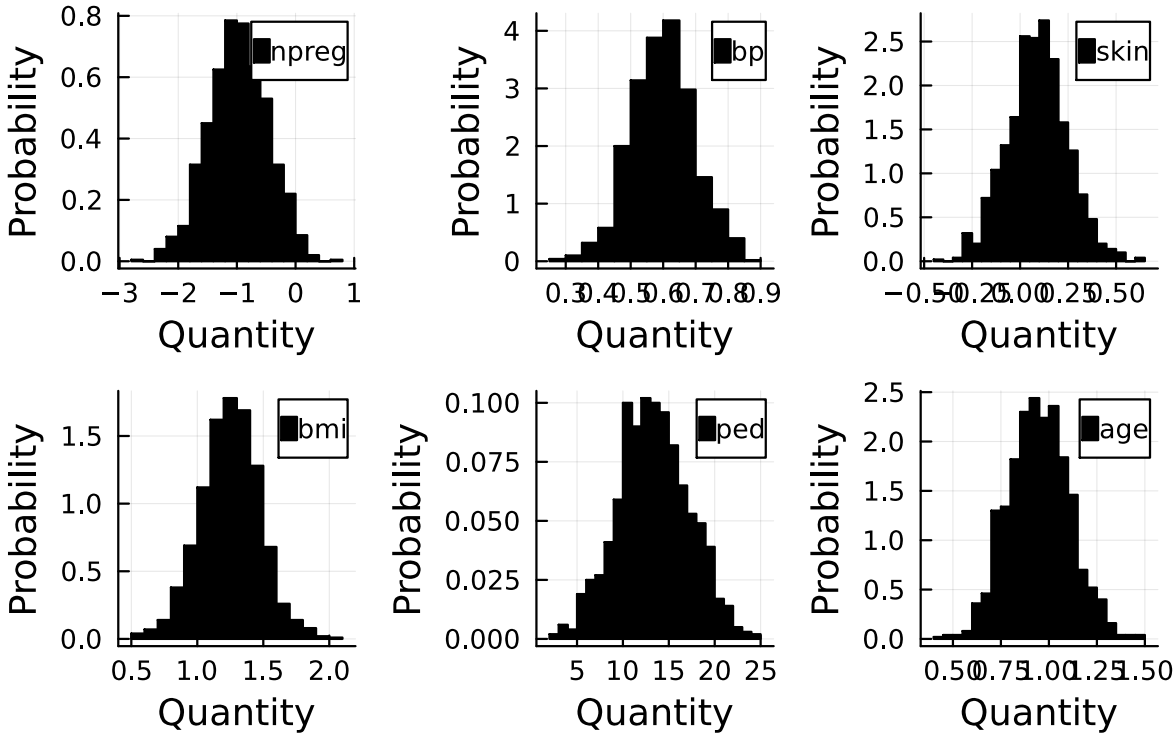
(a)

Fit a regression model using the g-prior with $g = n$, $\nu_0 = 2$ and $\sigma_0^2 = 1$. Obtain posterior confidence intervals for all of the parameters.

```
data = readdlm("data/azdiabetes.dat")
dt = data[1:end , 1:end-1]
y = float.(dt[2:end, 2])
X = float.(dt[2:end, 1:end .!= 2])
ns = data[1,1:end-1]
ns = ns[1:end .!=2]

@proto struct DiabetesModel
    S = 1000 # Number of sampling
    # Data
    y
    X
    n = length(y) # number of records
    p = size(X)[2]
    # Model
    # Prior
    g = n # g prior
      = 2.
     ² = 1.
      = MvNormal(zeros(p), g*  ²* (Hermitian(X'X))^-1)
```

```
end

function  _FCD( ², m::DiabetesModel)
    g = m.g; X = m.X
    Σ  = g/(g+1) *  ² * (X'X)^-1
      = g/(g+1) * ^( ², m)
      =  MvNormal( , Hermitian(Σ ))
    return
end

function ^( ², m::DiabetesModel)
    X = m.X; y = m.y
    return  ² * (X'X)^-1 * (X'y /  ²)
end


function  ²_FCD(m::DiabetesModel)
     = m.  + m.n / 2.
     = (m.  * m.  ² + SSR(m))/2.
    ² = InverseGamma( ,  )
    return  ²
end


function SSR(m::DiabetesModel)
    y = m.y; g = m.g; X=m.X
    return y'*(I - g/(g+1)*X*(X'X)^-1*X')*y
end


m = DiabetesModel(y=y, X=X )
 ²smp = zeros(m.S, 1)
 smp = zeros(m.S, size(m.X)[2])

for i in 1: m.S
     ²smp[i] = rand( ²_FCD(m))
     smp[i,:] = rand( _FCD( ²smp[i], m))
end

ps = [histogram( smp[:,i], xlabel="Quantity",
      ylabel="Probability",normalize=true, label="$(ns[i])", color="black")
```

```
      for i in 1:m.p]
plot(ps...)
```



(b)

Perform the model selection and averaging procedure described in Section 9.3. Obtain $Pr(\beta_j \neq 0|y)$, as well as posterior confidence intervals for all of the parameters. Compare to the results in part (a).