Assignment 2: ECO 545 (Fall 2020)

I pledge on my honor that I have not given or received any unauthorized assistance.
Name: Deepesh Kumar Lall Roll No.: 170236

---

1. **Solution: (a),(b),(c)** The code for `1.a`, `1.b`, and `1.c`, is in the appendix and works straight forward for the summary report for data in table using an available library `psych`.

   The result of each code's output on console is also shown below the code along with the command required to run them respectively. The .R files are also to be placed in the same folder as shown in the folder tree.

   (d) It can be noted that the results of both the part from (b) and (c) are exactly same as reason being the MCMC for censored data will give us the same best fit curve as OLS model as it slope doesnt have any link function effect in the CRM model in constrast to the result shown in Q3. part (f) where a link function factor comes into the equation and bring marginal difference. There by both the method gives the same $]beta$. Additionally the MCMC stablizes early in the process as the prior density explains the tobit results alredy, thereby giving a range of low valued standard deviation and then eventually zero.

   □

2. **Solution:** The code for Q2 is present in the appendix and all the results as asked of each part from `(a)` to `(i)` are printed as the output on the console of R. This .R script also generates a `q2_result.Rdata` file with the final feature vector and response variable numerically saved for Q3. □

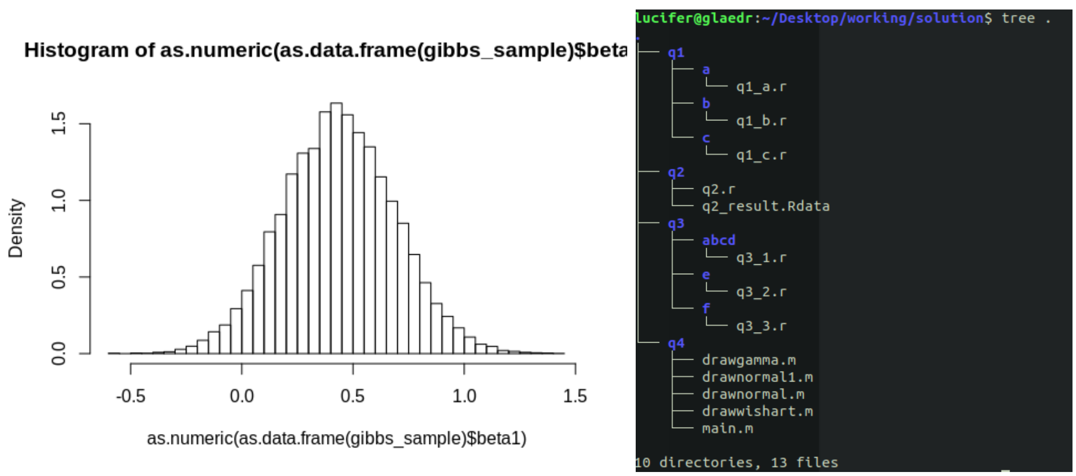

Figure 1: Reference figures.

3. **Solution: (a)** The binary probit model can be expressed in terms of latent variable as folows:

$$z_i = x_i'\beta + \epsilon_i, \quad \epsilon_i \sim N(0,1)$$

$$y_i = \begin{cases} 1, & \text{if } z_i > 0 \\ 0, & \text{if } z_i \leq 0 \end{cases}$$

   In order to find the $P(y_i = 0)$ and $P(y_i = 1)$ we can write it as:

$$P(y_i = 0) = P(z_i \leq 0) = P(\epsilon_i \leq -x_i'\beta)$$

$$P(y_i = 0) = \Phi(-x_i'\beta) = 1 - \Phi(x_i'\beta)$$

Similarly for $P(y_i = 1)$,

$$P(y_i = 0) = \Phi(x_i'\beta)$$

Thereby we can write the likelihood of this model as:

$$f(y|\beta, X) = \prod_{i=1}^{n} \{[\Phi(x_i'\beta)]^{y_i}[1 - \Phi(x_i'\beta)]^{1-y_i}\}$$

The Function for the above likelihood of data is implemented in the code and is available for use in subsequent parts i.e. for finding optimal β.

(b), (c), (d), (e), (f) These parts are all coded and result are all displayed on the console and respective output are also shown in the appendix below. The implementation of Gibbs sampling and long iteration continuously print the loop no for keeping a check on the progress and the final output format can be easily checked referring to the appendix. The handler for deciding to change the β from MLE to OLS are also added by just changing the value assignment to `beta_non_mc` in code corresponding to (b) part. Also the handlers are added to plot the result histogram(as shown in image) of Gibbs sample in beginning of code corresponding to (f) part.

Also the answer for (f) is different from that of (c) as there's a difference of factor of $\phi(\beta_0 + \beta_1 x_1 + .. + \beta_k x_k)$ in there observations. As because fitting a linear model in a censored data will pose no restriction for $Y = \Phi(\beta_0 + \beta_1 x_1 + .. + \beta_k x_k)$ rather a linear model will fit according to the best fit hyperplane i.e. $Y = \beta_0 + \beta_1 x_1 + .. + \beta_k x_k$ with no effect of link function. But as Gibbs sampling will give us β as sampled from the Link function's Y thereby will have a factor of β_j, as we can see below for linear models:

$$\frac{\partial Y}{\partial \beta_j} = \beta_j$$

but for Probit model Sample from Gibbs sampling having the regression results for Y with Link function on the hyperplane we have:

$$\frac{\partial Y}{\partial \beta_j} = \beta_j \phi(\beta_0 + \beta_1 x_1 + .. + \beta_k x_k)$$

□

4. **Solution:** The MATLAB code of this part is also present in the folder and added to this report in appendix and output of terminal are also shown below the code. The code need to be runned with all the folder in the same directory or say with all the code of appendix in the same folder but in different file for draw functions.

□

APPENDIX

1. **Code for Question 1.a**: (This Code is written in R.)

```
set.seed(1)
library(readxl)
library(dplyr)
library(psych)

table <- read_excel("Desktop/working/adoption.xlsx")
table$winbet = table$winbet - 125

table = table[,2:16]

stats <- psych::describe(table)
print(stats)
```

Output for Question 1.a: (This output is generated on R console.)

```
> source('~/Desktop/working/q1_a.r')
        vars    n    mean     sd median trimmed  mad  min  max range skew kurtosis   se
year       1 2692 2007.42   0.49   2007 2007.40 0.00 2007 2008     1 0.33    -1.89 0.01
winbet     2 2692  110.26 375.29      0   32.20 0.00    0 7675  7675 9.17   131.50 7.23
male       3 2692    0.24   0.43      0    0.17 0.00    0    1     1 1.23    -0.50 0.01
female     4 2692    0.46   0.50      0    0.45 0.00    0    1     1 0.15    -1.98 0.01
bay        5 2692    0.21   0.41      0    0.14 0.00    0    1     1 1.42     0.01 0.01
black      6 2692    0.09   0.28      0    0.00 0.00    0    1     1 2.97     6.84 0.01
brown      7 2692    0.08   0.27      0    0.00 0.00    0    1     1 3.10     7.60 0.01
buckskin   8 2692    0.07   0.25      0    0.00 0.00    0    1     1 3.47    10.02 0.00
dun        9 2692    0.07   0.25      0    0.00 0.00    0    1     1 3.49    10.18 0.00
gray      10 2692    0.07   0.25      0    0.00 0.00    0    1     1 3.40     9.54 0.00
pinto     11 2692    0.06   0.23      0    0.00 0.00    0    1     1 3.87    12.99 0.00
redroan   12 2692    0.04   0.20      0    0.00 0.00    0    1     1 4.48    18.04 0.00
sorrel    13 2692    0.17   0.38      0    0.09 0.00    0    1     1 1.75     1.06 0.01
age       14 2692    2.22   1.54      2    2.09 1.48    0   10    10 0.77     0.27 0.03
training  15 2692    0.18   0.38      0    0.10 0.00    0    1     1 1.69     0.87 0.01
```

Code for Question 1.b: (This Code is written in R.)

```
set.seed(1)
library(readxl)
library(dplyr)

table <- read_excel("Desktop/working/adoption.xlsx")
table$winbet = table$winbet - 125

X = table[,4:16]
Y = table$winbet
X = as.matrix(X)
X = cbind(1,X)
colnames(X)[1] <- "Intercept"
beta = solve(t(X) %*% X) %*% t(X) %*% Y
```

```
print(beta)

error = Y - X %*% beta
print(paste0("standard deviation : ",sd(error) ** 2) )
```

Output for Question 1.b: (This output is generated on R console.)

```
> source('~/Desktop/working/q1_b.r')
               [,1]
Intercept 134.274867
male       33.237989
female     10.227898
bay      -117.961861
black    -112.303165
brown    -132.329130
buckskin  -81.300646
dun       121.852364
gray     -116.848316
pinto     -62.457603
redroan   -36.086347
sorrel   -125.272181
age         3.140796
training  187.987287
[1] "standard deviation : 131865.95336334"
```

Code for Question 1.c: (This Code is written in R.)

```
set.seed(1)
library(readxl)
library(dplyr)
library(truncnorm)
library(invgamma)
library(psych)

burnin = 2
loopLength = 50

table <- read_excel("Desktop/working/adoption.xlsx")
table$winbet = table$winbet - 125
table = table[,3:16]

feature_vector = table[2:14]
y = table$winbet
feature_vector = as.matrix(feature_vector)
feature_vector = cbind(1,feature_vector)
colnames(feature_vector)[1] <- "Intercept"

n = dim(feature_vector)[1]
k = dim(feature_vector)[2]

beta.not = numeric(k)
B.not = diag(100, nrow = k, ncol = k)
beta.intial = mvtnorm::rmvnorm(1, mean = beta.not, sigma = B.not)
sigma2.not = rinvgamma(1, shape = 0.5, scale = 0.5)
ind_C = which(y==0)
```

```
beta.value = matrix(0,nrow = loopLength, ncol = 14)
sigma.value = matrix(0,nrow = loopLength, ncol = 1)

colnames(beta.value) <-
    c("beta1","beta2","beta3","beta4","beta5","beta6","beta7","beta8","beta9","beta10","beta11","beta12",
"beta13","beta14")
colnames(sigma.value) <- c("sigma2")

get.z = function(beta, sigma2){
  for (i in 1:length(ind_C))
  {
    index = ind_C[i]
    y[index] = rtruncnorm(1, a=-Inf, b=0, mean = as.vector(t(feature_vector[index,])) %*%
        beta, sd = sigma2^(1/2))
  }
  return(y)
}

get.beta = function(sigma2,z_dash)
{
  B1 = solve((1/sigma2) * t(feature_vector) %*% feature_vector + solve(B.not))
  beta.bar = B1 %*% ((1/sigma2) * t(feature_vector) %*% as.matrix(z_dash) + solve(B.not) %*%
      beta.not)
  beta_updated = mvtnorm::rmvnorm(1, mean = beta.bar, sigma = B1)

  return(beta_updated)
}

get.sigma2 = function(z_dash, beta)
{
  alpha1 = 1 + n
  delta1 = 1 + t(as.matrix(z_dash) - feature_vector%*%t(beta)) %*% (as.matrix(z_dash) -
      feature_vector%*%t(beta))
  sigma2_updated = rinvgamma(1, shape = alpha1/2 , scale = delta1/2)

  return(sigma2_updated)
}

y = get.z(beta.intial,sigma2.not)
beta.prev = get.beta(sigma2 = sigma2.not, z_dash = y)
sigma2.prev = get.sigma2(z_dash = y, beta = beta.prev)
y= get.z(beta = beta.prev,sigma2 = sigma2.prev)

beta.value[1,] = beta.prev
sigma.value[1,] = sigma2.prev

for(iter in 2:loopLength){
  beta.next = get.beta(sigma2 = sigma2.prev, z_dash = y)
  sigma2.next = get.sigma2(z_dash = y, beta = beta.next)
  y = get.z(beta = beta.next,sigma2 = sigma2.next)

  beta.value[iter,] = beta.next
  sigma.value[iter,] = sigma2.next

  beta.prev = beta.next
```

```
  sigma2.prev = sigma2.next
}

beta.value = beta.value[burnin:loopLength,]
sigma.value = sigma.value[burnin:loopLength,]

stats <- psych::describe(beta.value)
print(stats)
stats2 <- psych::describe(sigma.value)
print(stats2)
```

Output for Question 1.c: (This output is generated on R console.)

```
> source('~/Desktop/working/q1_c.r')
       vars     n    mean sd  median trimmed mad    min     max range  skew kurtosis se
beta1     1 10001  134.27  0  134.27  134.27   0  134.27  134.27     0 -0.01    -0.01  0
beta2     2 10001   33.24  0   33.24   33.24   0   33.24   33.24     0 -0.01     0.07  0
beta3     3 10001   10.23  0   10.23   10.23   0   10.23   10.23     0  0.01    -0.07  0
beta4     4 10001 -117.96  0 -117.96 -117.96   0 -117.96 -117.96     0  0.00     0.10  0
beta5     5 10001 -112.30  0 -112.30 -112.30   0 -112.30 -112.30     0 -0.02     0.00  0
beta6     6 10001 -132.33  0 -132.33 -132.33   0 -132.33 -132.33     0 -0.01    -0.06  0
beta7     7 10001  -81.30  0  -81.30  -81.30   0  -81.30  -81.30     0 -0.01     0.05  0
beta8     8 10001  121.85  0  121.85  121.85   0  121.85  121.85     0 -0.03     0.05  0
beta9     9 10001 -116.85  0 -116.85 -116.85   0 -116.85 -116.85     0  0.01     0.01  0
beta10   10 10001  -62.46  0  -62.46  -62.46   0  -62.46  -62.46     0  0.00     0.00  0
beta11   11 10001  -36.09  0  -36.09  -36.09   0  -36.09  -36.09     0  0.01     0.02  0
beta12   12 10001 -125.27  0 -125.27 -125.27   0 -125.27 -125.27     0 -0.01    -0.06  0
beta13   13 10001    3.14  0    3.14    3.14   0    3.14    3.14     0  0.02     0.00  0
beta14   14 10001  187.99  0  187.99  187.99   0  187.99  187.99     0  0.05     0.07  0
   vars     n mean sd median trimmed mad min max range skew kurtosis se
X1    1 10001    0  0      0       0   0   0   0     0  0.1    -0.06  0
```

2. **Code for Question 2**: (This Code is written in R.)

```
library(readxl)
library(plyr)
library(dplyr)
library(modeest)

noOfEntries = 1240
marijuana <- read_excel("Desktop/working/marijuana.xlsx")

# Q2.a
print("================== Q2.a =========================")
y <- marijuana$q85
y[y=="Yes, legal"] = 1
y[y=="No, illegal"] = 0
y <- as.integer(y)
print(paste0("Count for y_i=1 (count, %): (",sum(y),", ",sum(y)/noOfEntries*100," %)"))
print(paste0("Count for y_i=0 (count, %): (",noOfEntries-sum(y),",
    ",100-sum(y)/noOfEntries*100," %)"))

cat("\n")
print("================== Q2.b =========================")
x2 <- marijuana$age
x2 <- as.integer(x2)
x3 <- marijuana$hh1
x3 <- as.integer(x3)
print(paste0("Mean of x2: ",mean(x2)))
print(paste0("Median of x2: ",median(x2)))
# print(paste0("Mode of x2: ",mfv(x2)))
print(paste0("Std. Dev. of x2: ",sd(x2)))
print(paste0("Max of x2: ",max(x2)))
print(paste0("Min of x2: ",min(x2)))
print(paste0("Mean of x3: ",mean(x3)))
print(paste0("Median of x3: ",median(x3)))
# print(paste0("Mode of x3: ",mfv(x3)))
print(paste0("Std. Dev. of x3: ",sd(x3)))
print(paste0("Max of x3: ",max(x3)))
print(paste0("Min of x3: ",min(x3)))

cat("\n")
print("================== Q2.c =========================")
x4 <- marijuana$`past use`
x4[x4=="Yes"] = 1
x4[x4=="No"] = 0
x4 <- as.integer(x4)
print(paste0("Count for x4=1 (count, %): (",sum(x4),", ",sum(x4)/noOfEntries*100," %)"))
print(paste0("Count for x4=0 (count, %): (",noOfEntries-sum(x4),",
    ",100-sum(x4)/noOfEntries*100," %)"))

cat("\n")
print("================== Q2.d =========================")
x5 <- marijuana$sex
x5[x5=="Male"] = 1
x5[x5=="Female"] = 0
x5 <- as.integer(x5)
print(paste0("Count for x5=1 (count, %): (",sum(x5),", ",sum(x5)/noOfEntries*100," %)"))
```

```r
print(paste0("Count for x5=0 (count, %): (",noOfEntries-sum(x5),",
    ",100-sum(x5)/noOfEntries*100," %)"))

x6 <- marijuana$parent
x6[x6=="Yes"] = 1
x6[x6=="No"] = 0
x6 <- as.integer(x6)
print(paste0("Count for x6=1 (count, %): (",sum(x6),", ",sum(x6)/noOfEntries*100," %)"))
print(paste0("Count for x6=0 (count, %): (",noOfEntries-sum(x6),",
    ",100-sum(x6)/noOfEntries*100," %)"))

cat("\n")
print("================= Q2.e ========================")
x7 <- marijuana$`marital status`
x7[x7=="Never been married"] = 1
x7[x7!=1] = 0
x7 <- as.integer(x7)
print(paste0("Count for x7=1 (count, %): (",sum(x7),", ",sum(x7)/noOfEntries*100," %)"))
print(paste0("Count for x7=0 (count, %): (",noOfEntries-sum(x7),",
    ",100-sum(x7)/noOfEntries*100," %)"))

x8 <- marijuana$`marital status`
x8[x8=="Divorced" | x8=="Separated" | x8=="Widowed"] = 1
x8[x8!=1] = 0
x8 <- as.integer(x8)
print(paste0("Count for x8=1 (count, %): (",sum(x8),", ",sum(x8)/noOfEntries*100," %)"))
print(paste0("Count for x8=0 (count, %): (",noOfEntries-sum(x8),",
    ",100-sum(x8)/noOfEntries*100," %)"))

cat("\n")
print("================= Q2.f ========================")
x9 <- marijuana$income
x9[x9=="Less than 10000" | x9=="10 to under 20000" | x9=="20 to under 30000" | x9=="30 to
    under 40000" | x9=="40 to under 50000"] = 1
x9[x9!=1] = 0
x9 <- as.integer(x9)
print(paste0("Count for x9=1 (count, %): (",sum(x9),", ",sum(x9)/noOfEntries*100," %)"))
print(paste0("Count for x9=0 (count, %): (",noOfEntries-sum(x9),",
    ",100-sum(x9)/noOfEntries*100," %)"))

x10 <- marijuana$income
x10[x10=="50 to under 75000" | x10=="75 to under 100000"] = 1
x10[x10!=1] = 0
x10 <- as.integer(x10)
print(paste0("Count for x10=1 (count, %): (",sum(x10),", ",sum(x10)/noOfEntries*100," %)"))
print(paste0("Count for x10=0 (count, %): (",noOfEntries-sum(x10),",
    ",100-sum(x10)/noOfEntries*100," %)"))

cat("\n")
print("================= Q2.g ========================")
x11 <- marijuana$educ
x11[x11=="Less than HS" | x11=="HS Incomplete" | x11=="HS" ] = 1
x11[x11!=1] = 0
x11 <- as.integer(x11)
print(paste0("Count for x11=1 (count, %): (",sum(x11),", ",sum(x11)/noOfEntries*100," %)"))
print(paste0("Count for x11=0 (count, %): (",noOfEntries-sum(x11),",
```

```
    ",100-sum(x11)/noOfEntries*100," %)"))

x12 <- marijuana$educ
x12[x12=="Some college" | x12=="Associate Degree"] = 1
x12[x12!=1] = 0
x12 <- as.integer(x12)
print(paste0("Count for x12=1 (count, %): (",sum(x12),", ",sum(x12)/noOfEntries*100," %)"))
print(paste0("Count for x12=0 (count, %): (",noOfEntries-sum(x12),",
    ",100-sum(x12)/noOfEntries*100," %)"))

cat("\n")
print("================= Q2.h ========================")
x13 <- marijuana$race
x13[x13=="White"] = 1
x13[x13!=1] = 0
x13 <- as.integer(x13)
print(paste0("Count for x13=1 (count, %): (",sum(x13),", ",sum(x13)/noOfEntries*100," %)"))
print(paste0("Count for x13=0 (count, %): (",noOfEntries-sum(x13),",
    ",100-sum(x13)/noOfEntries*100," %)"))

x14 <- marijuana$race
x14[x14=="Black"] = 1
x14[x14!=1] = 0
x14 <- as.integer(x14)
print(paste0("Count for x14=1 (count, %): (",sum(x14),", ",sum(x14)/noOfEntries*100," %)"))
print(paste0("Count for x14=0 (count, %): (",noOfEntries-sum(x14),",
    ",100-sum(x14)/noOfEntries*100," %)"))

cat("\n")
print("================= Q2.i ========================")
x15 <- marijuana$party
x15[x15=="Democrat"] = 1
x15[x15!=1] = 0
x15 <- as.integer(x15)
print(paste0("Count for x15=1 (count, %): (",sum(x15),", ",sum(x15)/noOfEntries*100," %)"))
print(paste0("Count for x15=0 (count, %): (",noOfEntries-sum(x15),",
    ",100-sum(x15)/noOfEntries*100," %)"))

x16 <- marijuana$party
x16[x16=="Republican"] = 1
x16[x16!=1] = 0
x16 <- as.integer(x16)
print(paste0("Count for x16=1 (count, %): (",sum(x16),", ",sum(x16)/noOfEntries*100," %)"))
print(paste0("Count for x16=0 (count, %): (",noOfEntries-sum(x16),",
    ",100-sum(x16)/noOfEntries*100," %)"))

cat("\n")
result =
    matrix(unlist(list(y,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16)),nrow=noOfEntries,ncol=16)
write.table(result, file="~/Desktop/working/q2_result.Rdata", row.names = F, col.names = F)
unlink("q2_result.Rdata")
```

Output for Question 2: (This output is generated on R console.)

```
> source('~/Desktop/working/q2.r')
[1] "================= Q2.a ========================"
```

```
[1] "Count for y_i=1 (count, %): (659, 53.1451612903226 %)"
[1] "Count for y_i=0 (count, %): (581, 46.8548387096774 %)"

[1] "================== Q2.b ========================="
[1] "Mean of x2: 50.5008064516129"
[1] "Median of x2: 52"
[1] "Std. Dev. of x2: 17.7579456052065"
[1] "Max of x2: 96"
[1] "Min of x2: 18"
[1] "Mean of x3: 2.71935483870968"
[1] "Median of x3: 2"
[1] "Std. Dev. of x3: 1.44032086704917"
[1] "Max of x3: 8"
[1] "Min of x3: 1"

[1] "================== Q2.c ========================="
[1] "Count for x4=1 (count, %): (587, 47.3387096774194 %)"
[1] "Count for x4=0 (count, %): (653, 52.6612903225806 %)"

[1] "================== Q2.d ========================="
[1] "Count for x5=1 (count, %): (603, 48.6290322580645 %)"
[1] "Count for x5=0 (count, %): (637, 51.3709677419355 %)"
[1] "Count for x6=1 (count, %): (358, 28.8709677419355 %)"
[1] "Count for x6=0 (count, %): (882, 71.1290322580645 %)"

[1] "================== Q2.e ========================="
[1] "Count for x7=1 (count, %): (235, 18.9516129032258 %)"
[1] "Count for x7=0 (count, %): (1005, 81.0483870967742 %)"
[1] "Count for x8=1 (count, %): (285, 22.9838709677419 %)"
[1] "Count for x8=0 (count, %): (955, 77.0161290322581 %)"

[1] "================== Q2.f ========================="
[1] "Count for x9=1 (count, %): (642, 51.7741935483871 %)"
[1] "Count for x9=0 (count, %): (598, 48.2258064516129 %)"
[1] "Count for x10=1 (count, %): (366, 29.5161290322581 %)"
[1] "Count for x10=0 (count, %): (874, 70.4838709677419 %)"

[1] "================== Q2.g ========================="
[1] "Count for x11=1 (count, %): (414, 33.3870967741935 %)"
[1] "Count for x11=0 (count, %): (826, 66.6129032258065 %)"
[1] "Count for x12=1 (count, %): (381, 30.7258064516129 %)"
[1] "Count for x12=0 (count, %): (859, 69.2741935483871 %)"

[1] "================== Q2.h ========================="
[1] "Count for x13=1 (count, %): (954, 76.9354838709677 %)"
[1] "Count for x13=0 (count, %): (286, 23.0645161290323 %)"
[1] "Count for x14=1 (count, %): (146, 11.7741935483871 %)"
[1] "Count for x14=0 (count, %): (1094, 88.2258064516129 %)"

[1] "================== Q2.i ========================="
[1] "Count for x15=1 (count, %): (422, 34.0322580645161 %)"
[1] "Count for x15=0 (count, %): (818, 65.9677419354839 %)"
[1] "Count for x16=1 (count, %): (357, 28.7903225806452 %)"
[1] "Count for x16=0 (count, %): (883, 71.2096774193548 %)"
```

3. **Code for Question 3 (a),(b),(c),(d)**: (This Code is written in R.)

```
library(readxl)
library(plyr)
library(dplyr)
library(modeest)
library(truncnorm)
library(invgamma)
library(psych)

noOfData = 1240

# Reading the data into y and X from .Rdata file created by q2.r
q2_result <- read.table("~/Desktop/working/q2_result.Rdata", quote="\"", comment.char="")
Y <- q2_result[,1]
X <- q2_result[,2:16]
X = as.matrix(X)
X = cbind(1,X)

colnames(X)[1] <- "Intercept"
for(i in 2:16){
  str=paste0("x",i)
  colnames(X)[i] <- str
}

# Summary Report
# print(summary(Y))
# print(summary(X))

###################### Q3.a #######################
# Function to get likelihood for given Y, X and beta.
get_ith_NLL <- function(y,X,beta){
  Phi = pnorm(X %*% beta)
  f = sum(y*log(Phi)) + sum((1-y)*log(1-Phi))
  f = -f
  return(f)
}

getNLL <- function(Y,X,beta,noOfData){
  result = 0
  for(i in 1:noOfData){
    result = result + get_ith_NLL(Y[i],X[i,],beta)
  }
  return(result)
}

# Likelihood function for given Y and rho as sample mean(i.e. MLE)
get_ith_mean_NLL <- function(y,rho){
  Phi = pnorm(rho)
  f = sum(y*log(Phi)) + sum((1-y)*log(1-Phi))
  f = -f
  return(f)
}

get_mean_NLL <- function(Y,rho,noOfData){
  result = 0
```

```
  for(i in 1:noOfData){
    result = result + get_ith_mean_NLL(Y[i],rho)
  }
  return(result)
}

###################### Q3.b ######################
# Computing the Sample mean from given dataset.
sample_mean = sum(Y)/noOfData
rho_dash_dash = qnorm(sample_mean, mean=0, sd=1)
rho_dash = rep(rho_dash_dash,noOfData)
# print(rho_dash_dash)

matequal <- function(x, y)
  is.matrix(x) && is.matrix(y) && dim(x) == dim(y) && all(x == y)

beta_MLE = solve(X[1:16,]) %*% rho_dash[1:16]
if(matequal(X %*% beta_MLE ,rho_dash)){
  # print("Matrix Invertible")
}else{
  # print(X %*% beta_MLE -rho_dash)
  # print("MLE matrix not invertible")
}
# print(beta_MLE)
# print(X %*% beta_MLE)

beta_OLS = solve(t(X) %*% X) %*% t(X) %*% Y
# print(beta_OLS)
# print(mean(X %*% beta_OLS))

# Decide Beta
beta_non_mcmc = beta_OLS
xi_time_beta_non_mcmc = mean(X %*% beta_non_mcmc)

rownames(beta_non_mcmc) <-
    t(c("beta1","beta2","beta3","beta4","beta5","beta6","beta7","beta8","beta9","beta10","beta11","beta12",
"beta13","beta14","beta15","beta16"))

print("Q3.b")
print(beta_non_mcmc)

###################### Q3.c ######################
covariateEffect = rep(0,max(X[,2]-min(X[,2])))
# Covariate effect of increasing age by 1 year.
for(i in min(X[,2]):(max(X[,2])-5)){
  covariateEffect[i-min(X[,2])] =
      pnorm(xi_time_beta_non_mcmc-mean(X[,2])*beta_non_mcmc[2]+(i+5)*beta_non_mcmc[2],
      mean=0, sd=1) -
      pnorm(xi_time_beta_non_mcmc-mean(X[,2])*beta_non_mcmc[2]+i*beta_non_mcmc[2], mean=0,
      sd=1)
}
cat("\n\n")
print("Covariate Effect of Age difference of 5yr (Q3.c)")
print(covariateEffect[1:(max(X[,2])-5-min(X[,2]))])
print(paste0("Mean of Age Covariate : ",mean(covariateEffect[1:(max(X[,2])-5-min(X[,2]))])))
```

```
###################### Q3.d #######################
binaryCovariateEffect = 0
binaryCovariateEffect =
    pnorm(xi_time_beta_non_mcmc-mean(X[,6])*beta_non_mcmc[6]+beta_non_mcmc[6], mean=0, sd=1)
    - pnorm(xi_time_beta_non_mcmc-mean(X[,6])*beta_non_mcmc[6], mean=0, sd=1)
cat("\n\n")
print(paste0("Binary covariate of parent (Q3.d) : ",binaryCovariateEffect))
```

Output for Question 3 (a),(b),(c),(d): (This output is generated on R console.)

```
[1] "Q3.b"
             [,1]
beta1   0.676684111
beta2  -0.003135142
beta3  -0.021521593
beta4   0.292904735
beta5   0.056113591
beta6   0.018087723
beta7   0.040418173
beta8   0.018296975
beta9  -0.062850425
beta10 -0.059989957
beta11 -0.062632274
beta12 0.024412748
beta13 -0.016949194
beta14 -0.110569492
beta15 0.077085254
beta16 -0.166482154


[1] "Covariate Effect of Age difference of 5yr (Q3.c)"
 [1] -0.005152559 -0.005162597 -0.005172604 -0.005182579 -0.005192522 -0.005202434
     -0.005212313 -0.005222159 -0.005231973 -0.005241753 -0.005251501 -0.005261214
     -0.005270894
[14] -0.005280540 -0.005290151 -0.005299728 -0.005309270 -0.005318777 -0.005328248
    -0.005337684 -0.005347084 -0.005356448 -0.005365776 -0.005375067 -0.005384321
    -0.005393538
[27] -0.005402718 -0.005411860 -0.005420965 -0.005430031 -0.005439059 -0.005448049
    -0.005456999 -0.005465911 -0.005474783 -0.005483616 -0.005492410 -0.005501163
    -0.005509876
[40] -0.005518549 -0.005527181 -0.005535772 -0.005544321 -0.005552830 -0.005561297
    -0.005569722 -0.005578105 -0.005586446 -0.005594744 -0.005602999 -0.005611212
    -0.005619381
[53] -0.005627507 -0.005635589 -0.005643628 -0.005651622 -0.005659572 -0.005667477
    -0.005675338 -0.005683154 -0.005690924 -0.005698650 -0.005706329 -0.005713963
    -0.005721551
[66] -0.005729092 -0.005736587 -0.005744036 -0.005751437 -0.005758792 -0.005766099
    -0.005773359 -0.005780571
[1] "Mean of Age Covariate : -0.00548354807609352"


[1] "Binary covariate of parent (Q3.d) : 0.0062527813054063"
```

Code for Question 3 (e): (This Code is written in R.)

```r
library(readxl)
library(plyr)
library(dplyr)
library(modeest)
library(truncnorm)
library(invgamma)
library(psych)

noOfData = 1240
totalGibbsIteration = 20000
burnin = 5000

# Reading the data into y and X from .Rdata file created by q2.r
q2_result <- read.table("~/Desktop/working/q2_result.Rdata", quote="\"", comment.char="")
Y <- q2_result[,1]
X <- q2_result[,2:16]
X = as.matrix(X)
X = cbind(1,X)

colnames(X)[1] <- "Intercept"
for(i in 2:16){
  str=paste0("x",i)
  colnames(X)[i] <- str
}

n = dim(X)[1]
k = dim(X)[2]

beta.not = numeric(k)
B.not = diag(1, nrow = k, ncol = k)
z = matrix(rep(0,n))

draw.beta = function(z)
{
  B1 = solve(t(X) %*% X + solve(B.not))
  beta.bar = B1 %*% (t(X) %*% as.matrix(z) + solve(B.not) %*% beta.not)
  beta_updated = mvtnorm::rmvnorm(1, mean = beta.bar, sigma = B1)

  return(beta_updated)
}

draw.z = function(beta){
  for (i in 1:n)
  {
    if(Y[i]==0){
      z[i] = rtruncnorm(1, a=-Inf, b=0, mean = as.vector(t(X[i,])) %*% beta, sd = 1)
    }else{
      z[i] = rtruncnorm(1, a=0, b=+Inf, mean = as.vector(t(X[i,])) %*% beta, sd = 1)
    }
  }
  return(z)
}

beta.value = matrix(0,nrow = totalGibbsIteration, ncol = k)

beta.prev = mvtnorm::rmvnorm(1, mean = beta.not, sigma = B.not)
```

```
z = draw.z(beta.prev)
beta.value[1,] = beta.prev

for(iter in 2:totalGibbsIteration){
  # comment print below to hide progress
  print(paste0(iter," out of ",totalGibbsIteration))
  beta.prev = draw.beta(z)
  z = draw.z(beta.prev)
  beta.value[iter,] = beta.prev
}
print(psych::describe(beta.value[burnin:totalGibbsIteration,]))

write.table(beta.value, file="~/Desktop/working/q3_2_result.Rdata", row.names = F, col.names
     = F)
unlink("~/Desktop/working/q3_2_result.Rdata")
```

Output for Question 3 (e): (This output is generated on R console.)

```
> source('~/Desktop/working/q3_2.r')
[1] "2 out of 20000"
[1] "3 out of 20000"
[1] "4 out of 20000"
[1] "5 out of 20000"
[1] "6 out of 20000"
[1] "7 out of 20000"
[1] "8 out of 20000"
[1] "9 out of 20000"
[1] "10 out of 20000"
[1] "11 out of 20000"
[1] "12 out of 20000"
[1] "13 out of 20000"
[1] "14 out of 20000"
[1] "15 out of 20000"
.
.
.
[1] "20000 out of 20000"
    vars     n  mean   sd median trimmed mad  min  max range skew kurtosis se
X1     1 15001 0.44 0.25  0.43    0.43 0.25 -0.59 1.43 2.03 0.03    0.02 0
X2     2 15001 -0.01 0.00 -0.01  -0.01 0.00 -0.02 0.00 0.02 0.02    0.04 0
X3     3 15001 -0.03 0.03 -0.03  -0.03 0.03 -0.14 0.11 0.25 0.03   -0.03 0
X4     4 15001 0.48 0.07  0.48    0.48 0.07 0.18 0.80 0.62 0.03    0.00 0
X5     5 15001 0.09 0.07  0.09    0.09 0.07 -0.23 0.34 0.57 0.00    0.02 0
X6     6 15001 0.03 0.09  0.03    0.03 0.09 -0.34 0.38 0.72 0.02   -0.05 0
X7     7 15001 0.07 0.11  0.07    0.07 0.11 -0.31 0.49 0.80 -0.02  -0.03 0
X8     8 15001 0.03 0.09  0.03    0.03 0.09 -0.32 0.41 0.73 0.00   -0.04 0
X9     9 15001 -0.10 0.10 -0.10  -0.10 0.10 -0.44 0.30 0.74 0.03   -0.11 0
X10   10 15001 -0.09 0.10 -0.09  -0.09 0.10 -0.45 0.28 0.73 0.01   -0.03 0
X11   11 15001 -0.10 0.09 -0.10  -0.10 0.09 -0.48 0.21 0.69 -0.03   0.04 0
X12   12 15001 0.04 0.09  0.04    0.04 0.09 -0.28 0.43 0.71 0.03    0.00 0
X13   13 15001 -0.02 0.11 -0.02  -0.02 0.11 -0.50 0.41 0.91 0.02    0.04 0
X14   14 15001 -0.18 0.14 -0.18  -0.18 0.14 -0.72 0.39 1.11 -0.01   0.07 0
X15   15 15001 0.13 0.08  0.13    0.13 0.08 -0.23 0.47 0.70 0.02   -0.02 0
X16   16 15001 -0.27 0.08 -0.28  -0.27 0.09 -0.61 0.06 0.67 0.02   -0.03 0
```

Code for Question 3 (f): (This Code is written in R.)

```
library(readxl)
library(plyr)
library(dplyr)
library(modeest)
library(truncnorm)
library(invgamma)

noOfData = 1240

q3_result <- read.table("~/Desktop/working/q3_2_result.Rdata", quote="\"", comment.char="")
gibbs_sample = q3_result[5000:20000,]
gibbs_sample = as.matrix(gibbs_sample)

# Reading the data into y and X from .Rdata file created by q2.r
q2_result <- read.table("~/Desktop/working/q2_result.Rdata", quote="\"", comment.char="")
Y <- q2_result[,1]
X <- q2_result[,2:16]
X = as.matrix(X)
X = cbind(1,X)

colnames(X)[1] <- "Intercept"
for(i in 2:16){
  str=paste0("x",i)
  colnames(X)[i] <- str
}

n = dim(X)[1]
k = dim(X)[2]

colnames(gibbs_sample)[1] <- "beta1"
for(i in 2:16){
  str=paste0("beta",i)
  colnames(gibbs_sample)[i] <- str
}

# Use below snippet for ploting
hist(as.numeric(as.data.frame(gibbs_sample)$beta1), freq=FALSE, breaks=40)

# Printing the Table output same as q3_2.r output for q3.d
# cat("Coefficient \t Mean \t S.D. \t Lower \t Upper \n")
# for(iter in 1:16){
#   cat(paste0("beta",iter),"\t", mean(gibbs_sample[,iter]),"\t", sd(gibbs_sample[,iter]),
    "\t", max(gibbs_sample[,iter]),"\t",min(gibbs_sample[,iter]),"\n")
# }
print(psych::describe(gibbs_sample))

# Covariate Effect of Age for a difference of 5
sum = 0
for(g in 1:15000){
  # Use print(g) to check progress.
  print(paste0(g," out of ",15000))
  for(p in 1:n){
    sum = sum + gibbs_sample[g,2]*dnorm(t(X[p,]) %*% gibbs_sample[g,], mean=0, sd=1)
  }
```

```
}

# Covariate Effect of Parent
sum = 0
for(g in 1:15000){
  # Use print(g) to check progress.
  print(paste0(g," out of ",15000))
  for(p in 1:n){
    sum = sum + pnorm(t(X[p,]) %*% gibbs_sample[g,] - X[p,6]*gibbs_sample[g,6] +
        gibbs_sample[g,6], mean=0, sd=1) - pnorm(t(X[p,]) %*% gibbs_sample[g,] -
        X[p,6]*gibbs_sample[g,6], mean=0, sd=1)
  }
}
print(paste0("Marginal Effect of Age : ",sum/(n*15000)))
print(paste0("5 yr Covariate Effect of age : ",sum/(n*15000)*5))
print(paste0("Marginal Effect of Parent : ",sum/(n*15000)))
```

Output for Question 3 (f): (This output is generated on R console.)

```
> source('~/Desktop/working/q3_3.r')
      vars     n  mean   sd median trimmed mad   min  max range  skew kurtosis se
beta1     1 15001  0.44 0.25   0.43    0.43 0.25 -0.59 1.43  2.03  0.03     0.02  0
beta2     2 15001 -0.01 0.00  -0.01   -0.01 0.00 -0.02 0.00  0.02  0.02     0.04  0
beta3     3 15001 -0.03 0.03  -0.03   -0.03 0.03 -0.14 0.11  0.25  0.03    -0.03  0
beta4     4 15001  0.48 0.07   0.48    0.48 0.07  0.18 0.80  0.62  0.03     0.00  0
beta5     5 15001  0.09 0.07   0.09    0.09 0.07 -0.23 0.34  0.57  0.00     0.02  0
beta6     6 15001  0.03 0.09   0.03    0.03 0.09 -0.34 0.38  0.72  0.02    -0.05  0
beta7     7 15001  0.07 0.11   0.07    0.07 0.11 -0.31 0.49  0.80 -0.02    -0.03  0
beta8     8 15001  0.03 0.09   0.03    0.03 0.09 -0.32 0.41  0.73  0.00    -0.04  0
beta9     9 15001 -0.10 0.10  -0.10   -0.10 0.10 -0.44 0.30  0.74  0.03    -0.11  0
beta10   10 15001 -0.09 0.10  -0.09   -0.09 0.10 -0.45 0.28  0.73  0.01    -0.03  0
beta11   11 15001 -0.10 0.09  -0.10   -0.10 0.09 -0.48 0.21  0.69 -0.03     0.04  0
beta12   12 15001  0.04 0.09   0.04    0.04 0.09 -0.28 0.43  0.71  0.03     0.00  0
beta13   13 15001 -0.02 0.11  -0.02   -0.02 0.11 -0.50 0.41  0.91  0.02     0.04  0
beta14   14 15001 -0.18 0.14  -0.18   -0.18 0.14 -0.72 0.39  1.11 -0.01     0.07  0
beta15   15 15001  0.13 0.08   0.13    0.13 0.08 -0.23 0.47  0.70  0.02    -0.02  0
beta16   16 15001 -0.27 0.08  -0.28   -0.27 0.09 -0.61 0.06  0.67  0.02    -0.03  0
[1] "1 out of 15000"
[1] "2 out of 15000"
[1] "3 out of 15000"
[1] "4 out of 15000"
[1] "5 out of 15000"
[1] "6 out of 15000"
[1] "7 out of 15000"
[1] "8 out of 15000"
[1] "9 out of 15000"
[1] "10 out of 15000"
[1] "11 out of 15000"
.
.
.
[1] "14999 out of 15000"
[1] "15000 out of 15000"
[1] "Marginal Effect of Age : 0.0111119790715667"
[1] "5 yr Covariate Effect of age : 0.0555598953578335"
[1] "Marginal Effect of Parent : 0.0111119790715667"
```

4. **Code for Question 4**: (This Code is written in MATLAB.)

```
%% Data Summary and reading
data =xlsread('Vella-Verbeek-Data.xlsx','Data');
[m,n] = size(data);
ID = data(:,1);
Year = data(:,2);
AG = data(:,3);
Black = data(:,4);
Bus = data(:,5);
Con = data(:,6);
Ent = data(:,7);
Exper = data(:,8);
Fin = data(:,9);
Hisp = data(:,10);
Hlth = data(:,11);
Hours = data(:,12);
Man = data(:,13);
Mar = data(:,14);
Min = data(:,15);
Nc = data(:,16);
Ne = data(:,17);
Occ1 = data(:,18);
Occ2 = data(:,19);
Occ3 = data(:,20);
Occ4 = data(:,21);
Occ5 = data(:,22);
Occ6 = data(:,23);
Occ7 = data(:,24);
Occ8 = data(:,25);
Occ9 = data(:,26);
Per = data(:,27);
Pro = data(:,28);
Pub = data(:,29);
Rur = data(:,30);
S = data(:,31);
School = data(:,32);
Tra = data(:,33);
Trad = data(:,34);
Union = data(:,35);
wage = data(:,36);
all_means = [mean(Hours) mean(School) mean(Exper) mean(wage)];
all_std = [std(Hours) std(School) std(Exper) std(wage)];
all_count = [sum(AG) sum(Black) sum(Bus) sum(Con) sum(Ent)...
    sum(Fin) sum(Hisp) sum(Hlth) sum(Man) sum(Mar) sum(Min) sum(Nc) sum(Ne)...
    sum(Occ1) sum(Occ2) sum(Occ2) sum(Occ3) sum(Occ4) sum(Occ5) sum(Occ6)...
    sum(Occ7) sum(Occ8) sum(Occ9) sum(Per) sum(Pro) sum(Pub) sum(Rur) sum(S)...
    sum(Tra) sum(Trad) sum(Union)];
percentage = all_count./m;
rowlabels = char('Hours','School','Exper','Wages');
rowlabels1 = char('AG','Black','Bus','Con','Ent','Fin','Hisp','Hlth'...
    ,'Man','Mar','Nc','Ne','Occ1','Occ2','Occ3','Occ4','Occ5','Occ6','Occ7'...
    ,'Occ8','Occ9','Per','Pro','Pub','Rur','S','Tra','Trad','Union');
fprintf("        Data Summary                 \n")
fprintf("_______________________________________________\n")
fprintf("     Mean      Standard Deviation  \n")
```

```
fprintf("_______________________________________________\n")
for i=1:4
    fprintf('%s %4.2f     %4.2f \n', rowlabels(i,:), all_means(i), all_std(i))
end
fprintf("_______________________________________________\n")
fprintf("     Counts    Percentage            \n")
fprintf("_______________________________________________\n")
for i=1:29
    fprintf('%s %4.2f   %4.2f  \n', rowlabels1(i,:),all_count(i),percentage(i)*100)
end
%% Panel data Model
Y = wage;
W = [ones(m,1) Exper];
X = [AG Black Bus Con Ent Fin Hisp Hlth Hours Man Mar Min Nc Ne Occ1 Occ2...
    Occ3 Occ4 Occ5 Occ6 Occ7 Occ8 Per Pro Pub Rur S School Trad Union];
nsim = 10000;
burn = 2500;
K1 = size(X,2);
K2 = size(W,2);
Beta = zeros(K1,nsim);
bis = zeros(K2,m/8,nsim);
Hu = zeros(1,nsim);
invD = zeros(K2,K2,nsim);
Do = inv(eye(K2));
vo = 6;
bo = zeros(K1,1);
Bo = 10*eye(K1);
invBo = inv(Bo);
invBobo = Bo\bo;
alphao = 6;
deltao = 3;
b1 = zeros(K2,1);
Beta(:,1) = ones(K1,1);
bis(:,:,1) = ones(K2,m/8);
h = waitbar(0,'Simulation in Progress');
for i=2:nsim
    Hu(1,i) = drawgamma(alphao,m,deltao,Y,W,X,Beta(:,i-1),bis(:,:,i-1));
    invD(:,:,i) = drawwishart(vo,Do,m/8,bis(:,:,i-1),K2);
    Beta(:,i) = drawnormal(X,Y,W,invD(:,:,i),Hu(1,i),invBo,invBobo,m,K1);
    bis(:,:,i) = drawnormal1(Hu(1,i),W,invD(:,:,i),Y,X,Beta(:,i),m,K2);
    waitbar(1/nsim);
end
close(h);
beta_means = mean(Beta(:,burn+1:nsim),2);
beta_std = std(Beta(:,burn+1:nsim),0,2);
bis_means = zeros(K2,m/8);
bis_std =zeros(K2,m/8);
for i=1:m/8
    bis_means(:,i) = [mean(bis(1,i,burn+1:nsim)); mean(bis(2,i,burn+1:nsim))];
    bis_std(:,i) = [std(bis(1,i,burn+1:nsim)); std(bis(2,i,burn+1:nsim))];
end
hu_means = mean(Hu,2);
D_means = [mean(invD(1,1,burn+1:nsim))...
    mean(invD(1,2,burn+1:nsim)); ...
    mean(invD(2,1,burn+1:nsim)) ...
    mean(invD(2,2,burn+1:nsim))];
```

```
D_std = [std(invD(1,1,burn+1:nsim))...
    std(invD(1,2,burn+1:nsim)); ...
    std(invD(2,1,burn+1:nsim))...
    std(invD(2,2,burn+1:nsim))];
hu_std = std(Hu,0,2);
mean_bis_mean = mean(bis_means,2);
mean_bis_std = mean(bis_std,2);
rowlabels2 = char('AG', 'Black', 'Bus','Con', 'Ent', 'Fin', 'Hisp', 'Hlth',...
    'Hours', 'Man', 'Mar', 'Min', 'Nc', 'Ne', 'Occ1', 'Occ2','Occ3','Occ4',...
    'Occ5', 'Occ6', 'Occ7', 'Occ8', 'Per', 'Pro', 'Pub', 'Rur', 'S', 'School'...
    ,'Trad', 'Union');
rowlabels3 =char('Intercept','Experience');
fprintf("             Bayesian Estimates                \n")
fprintf("_______________________________________________\n")
fprintf("          Mean    Standard Deviation  \n")
fprintf("_______________________________________________\n")
for i=1:K2
    fprintf('%s %4.2f    %4.2f \n',rowlabels3(i,:), mean_bis_mean(i),mean_bis_std(i));
end
for i=1:K1
    fprintf('%s %4.2f    %4.2f \n',rowlabels2(i,:), beta_means(i), beta_std(i));
end
fprintf("__________________________________________________\n")
fprintf("Mean of invD Matrix = \n")
disp(D_means)
fprintf("Standard Deviation of invD Matrix = \n")
disp(D_std)
fprintf("__________________________________________________\n")
fprintf("Mean of Hu = \n")
disp(hu_means)
fprintf("Standard Deviation of Hu = \n")
disp(hu_std)
%% Interpretation of Coefficients
fprintf('As the coefficient for Union membership is positive and statistically\n')
fprintf(' significant, it implies that the logarithm of wages of the employee\n')
fprintf(' increases by 0.08 if they are a part of Union as compared to when\n')
fprintf(' they are not.\n');
```

```
function [Beta] = drawnormal(X,Y,W,invD,Hu,invBo,invBobo,m,K1)
%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here
T=7;
sum = zeros(K1,K1);
sum1 = zeros(K1,1);
for i=1:T+1:m
     Wj = W(i:i+T,:);
     Yj = Y(i:i+T);
     Xj = X(i:i+T,:);
     B1j = Wj*(pinv(invD))*(Wj') + pinv(Hu)*eye(T+1);
     invB1j = B1j\eye(8);
     sum = sum + Xj'*(invB1j)*Xj;
     sum1 = sum1 + Xj'*(invB1j)*Yj;
end
B_sur = pinv(sum + invBo);
Btilde = B_sur*(invBobo + sum1);
```

```
L =chol(B_sur,'lower');
Beta = Btilde + (L*mvnrnd(zeros(K1,1), eye(K1))');
end
```

```
function [Hu] = drawgamma(alphao,m,deltao,Y,W,X,Beta,bis)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
alpha1 = alphao + m;
T = 7;
sum = 0;
z=1;
for i=1:T+1:m
    Yj = Y(i:i+T);
    Xj = X(i:i+T,:);
    Wj = W(i:i+T,:);
    bisj = bis(:,z);
    sum = sum + (Yj - Xj*Beta - Wj*bisj)'*(Yj - Xj*Beta - Wj*bisj);
    z=z+1;
end
delta1 = deltao + sum;
Hu = gamrnd(alpha1/2,2/delta1);
end
```

```
function [bis]= drawnormal1(Hu,W,invD,Y,X,Beta,m,K2)
%UNTITLED4 Summary of this function goes here
%   Detailed explanation goes here
T = 7;
z=1;
bis = zeros(K2,m/8);
for i=1:T+1:m
     Wj = W(i:i+T,:);
     Yj = Y(i:i+T);
     Xj = X(i:i+T,:);
     D1j = pinv(Hu*(Wj')*Wj + invD);
     bj = D1j*(Hu*(Wj')*(Yj - Xj*Beta));
     L = chol(D1j,'lower');
     bis(:,z) = bj + (L*mvnrnd(zeros(K2,1), eye(K2))');
     z=z+1;
end
end
```

```
function [invD] = drawwishart(vo,Do,n,bis,K2)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
v = vo + K2;
sum=zeros(K2,K2);
for i=1:n
    sum = sum + bis(:,i)*(bis(:,i))';
end
D1 = pinv(Do + sum);
invD = wishrnd(D1,v);
end
```

Output for Question 4: (This output is generated on MATLAB terminal.)

```
>> main
         Data Summary
-----------------------------------------------
      Mean      Standard Deviation
-----------------------------------------------
Hours 2191.26      566.35
School 11.77     1.75
Exper 6.51      2.83
Wages 1.65      0.53
-----------------------------------------------
     Counts    Percentage
-----------------------------------------------
AG    140.00    3.21
Black 504.00    11.56
Bus   331.00    7.59
Con   327.00    7.50
Ent   66.00    1.51
Fin   161.00    3.69
Hisp  680.00    15.60
Hlth  74.00    1.70
Man   1231.00    28.23
Mar   1914.00    43.90
Nc    68.00    1.56
Ne    1124.00    25.78
Occ1  829.00    19.01
Occ2  453.00    10.39
Occ3  399.00    9.15
Occ4  399.00    9.15
Occ5  233.00    5.34
Occ6  486.00    11.15
Occ7  934.00    21.42
Occ8  881.00    20.21
Occ9  401.00    9.20
Per   64.00    1.47
Pro   509.00    11.67
Pub   73.00    1.67
Rur   333.00    7.64
S     175.00    4.01
Tra   889.00    20.39
Trad  1529.00    35.07
Union 286.00   6.56
              Bayesian Estimates
-----------------------------------------------
           Mean    Standard Deviation
-----------------------------------------------
Intercept  -0.34    0.58
Experience 0.07    0.05
AG     0.07    0.06
Black  -0.10    0.81
Bus    0.04    0.04
Con    0.01    0.05
Ent    0.01    0.07
Fin    0.10    0.06
Hisp   0.08    0.71
```

```
Hlth   -0.06    0.05
Hours  -0.00    0.00
Man    0.07    0.04
Mar    0.05    0.02
Min    0.08    0.07
Nc     -0.09    0.07
Ne     0.04    0.10
Occ1   0.04    0.04
Occ2   0.03    0.04
Occ3   -0.04    0.04
Occ4   0.01    0.03
Occ5   0.01    0.03
Occ6   0.00    0.03
Occ7   -0.00    0.04
Occ8   0.01    0.08
Per    0.11    0.06
Pro    0.02    0.05
Pub    0.04    0.05
Rur    0.02    0.03
S      0.04    0.07
School 0.17    0.03
Trad   0.00    0.04
Union  0.08    0.02
--------------------------------------------------
Mean of invD Matrix =
    0.0394    0.1986
    0.1986    2.0101

Standard Deviation of invD Matrix =
    0.0244    0.1480
    0.1480    1.1512

--------------------------------------------------
Mean of Hu =
    9.9635

Standard Deviation of Hu =
    0.3889

As the coefficient for Union membership is positive and statistically
 significant, it implies that the logarithm of wages of the employee
 increases by 0.08 if they are a part of Union as compared to when
 they are not.
```