STA365_homework2_code

Ruike Xu

3/29/2022

Question 2

```
# install.packages("rjags")
# install.packages("coda")

set.seed(1006562550)
library(coda)
```

```
## Warning: package 'coda' was built under R version 4.1.3
```

```
library(R2jags)
```

```
## Warning: package 'R2jags' was built under R version 4.1.3
```

```
## Loading required package: rjags
```

```
## Warning: package 'rjags' was built under R version 4.1.3
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
##
## Attaching package: 'R2jags'
```

```
## The following object is masked from 'package:coda':
##
##     traceplot
```

```
library(lattice)
```

$$Warren: \ \ \tau(1-\theta)+\theta(1-\lambda)$$

$$Trump(\psi): \ \ \lambda\theta+(1-\tau)(1-\theta)$$

```
# Model construction with likelihood function and prior
model.JAGS = function(){
    y ~ dbinom(psi, n)
    theta ~ dbeta(3, 7)
    tau ~ dbeta(6, 4)
    lambda ~ dbeta(7, 3)
    psi <- theta * lambda + (1 - theta) * (1 - tau)
  }
```

The model we are constructing is intended to predict which candidate is likely to win the 2024 US presidential election (Donald Trump or Elizabeth Warren), which is based on binomial likelihood and beta prior. θ is the probability that the voter claims to vote for trump (might lying), thus, $1-\theta$ is the probability that the person claims to vote for Warren. λ is the percentage of the population that is actually willing to vote for Trump while τ is the percentage of the population that is actually willing to vote for Warren. We can detect the 'Quasi Liberal' and 'Quasi Republican' as $1-\lambda$ and $1-\tau$. To be more specific, the percentage of the population that claimed they would vote for Trump/Warren but ended voting the opposite.

Therefore, we can derive that the actual probability of a person voting for Trump as $\psi : \lambda\theta + (1-\tau)(1-\theta)$ and the actual probability of a person voting for Warren as $\tau(1-\theta) + \theta(1-\lambda)$

To define τ and λ, we would assume more Republican people are less willing to claim their feelings in Liberal states, so τ ~ Beta(6, 4) and λ ~ Beta(7,3)

```
# Simulating data
n = 100000
y = 30000
data.JAGS = list(y = y, n = n)
```

```
# Randomly select the initial values
inits.JAGS = function(){
  return(list(theta=rbeta(1, 3, 7),tau=rbeta(1, 6, 4),lambda=rbeta(1, 7, 3)))
}
```

```
# Select parameters that will be simulated with MCMC model
para.JAGS = c("theta", "tau", "lambda", "psi")
```

```
# Fit the MCMC model, number of iterations = 90000, burn in = 10000
fit.JAGS = jags(data=data.JAGS,inits=inits.JAGS,
                parameters.to.save = para.JAGS,
                n.chains=1,
                n.iter=90000,
                n.burnin=10000,
                model.file=model.JAGS)
```

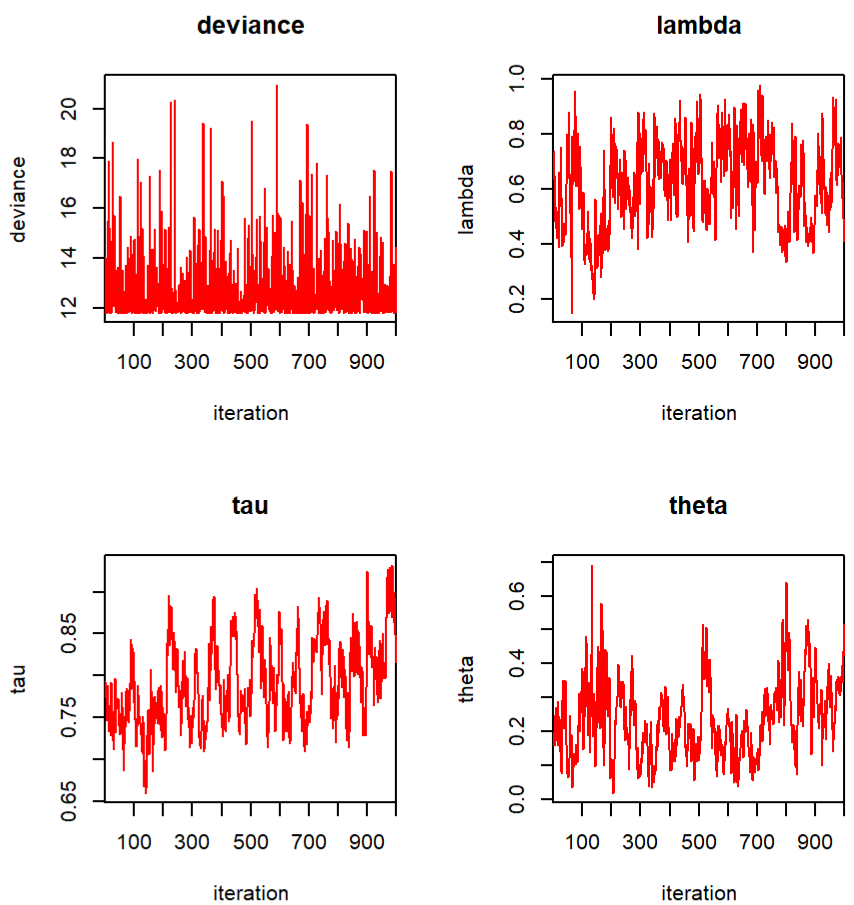
```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1
##    Unobserved stochastic nodes: 3
##    Total graph size: 15
##
## Initializing model
```

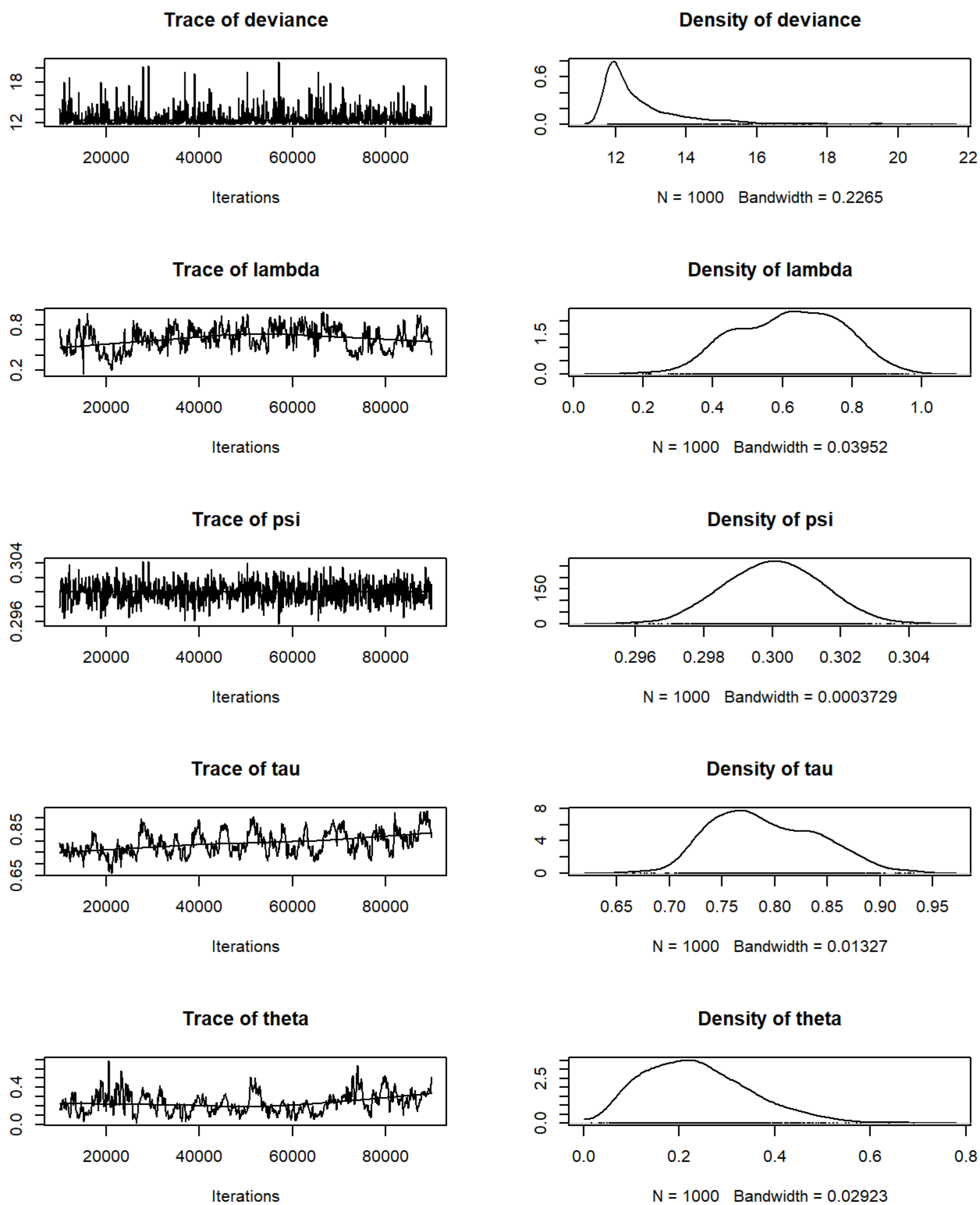
```
# Print model fit summary
print(fit.JAGS)
```

```
## Inference for Bugs model at "C:/Users/Ruike Xu/AppData/Local/Temp/Rtmpgtofxy/model862071d15d2
a.txt", fit using jags,
##  1 chains, each with 90000 iterations (first 10000 discarded), n.thin = 80
##  n.sims = 1000 iterations saved
##          mu.vect sd.vect   2.5%    25%    50%    75%  97.5%
## lambda     0.620   0.148  0.348  0.507  0.625  0.732  0.880
## psi        0.300   0.001  0.297  0.299  0.300  0.301  0.303
## tau        0.791   0.050  0.713  0.752  0.784  0.830  0.891
## theta      0.236   0.110  0.067  0.152  0.226  0.308  0.478
## deviance  12.724   1.272 11.791 11.890 12.245 13.030 16.160
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 0.8 and DIC = 13.5
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
# Generate traceplots
traceplot(fit.JAGS,mfrow=c(2,3),ask=FALSE)
```



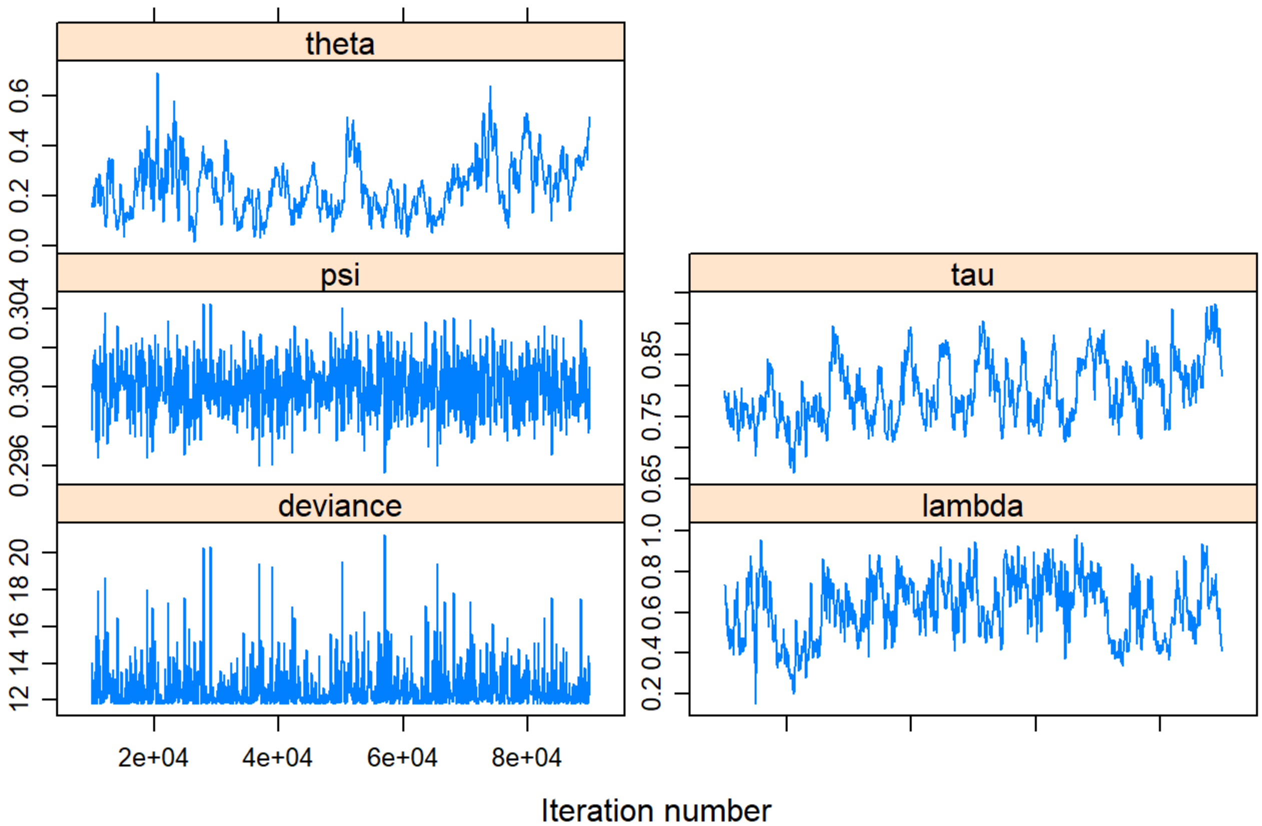

```
# Plot MCMC object to show all parameter densities along with traceplots
fit.JAGS.mcmc = as.mcmc(fit.JAGS)
plot(fit.JAGS.mcmc,ask=FALSE)
```


Trace of deviance
Iterations
Density of deviance
N = 1000   Bandwidth = 0.2265
Trace of lambda
Iterations
Density of lambda
N = 1000   Bandwidth = 0.03952
Trace of psi
Iterations
Density of psi
N = 1000   Bandwidth = 0.0003729
Trace of tau
Iterations
Density of tau
N = 1000   Bandwidth = 0.01327
Trace of theta
Iterations
Density of theta
N = 1000   Bandwidth = 0.02923

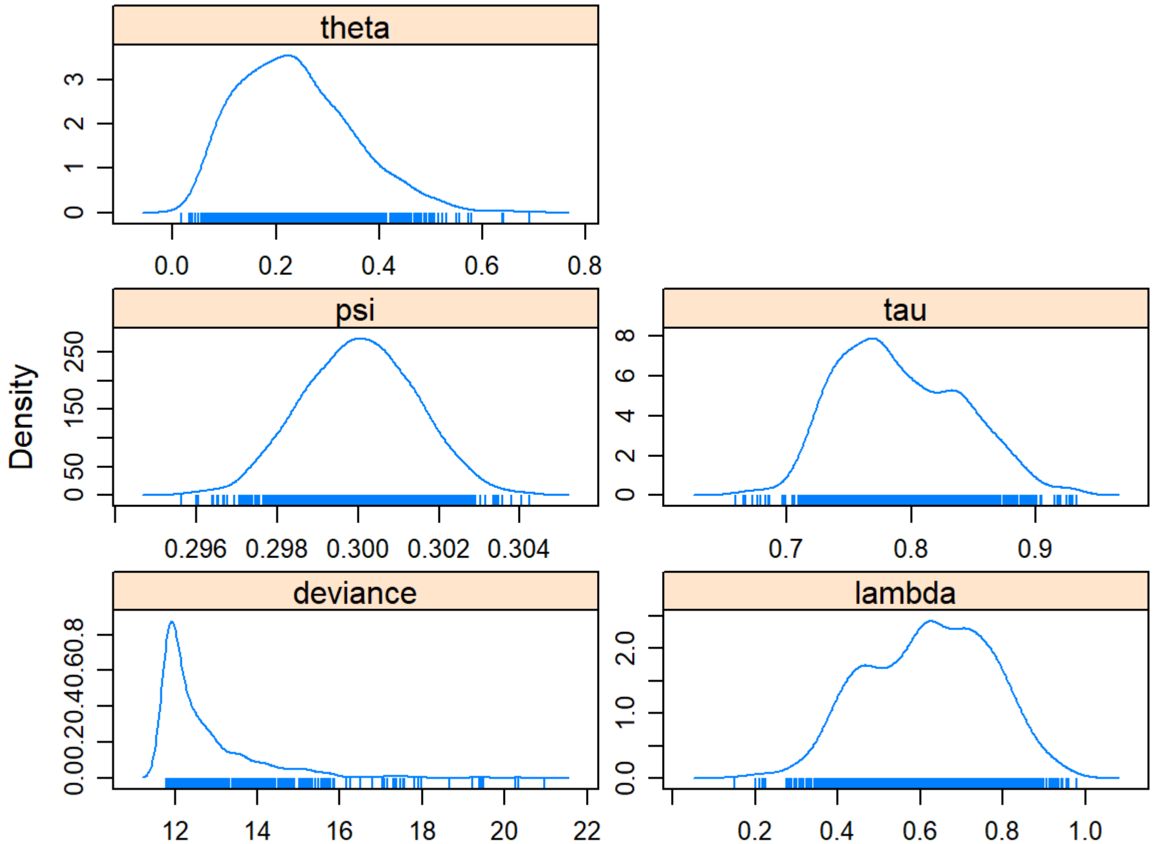
```r
# Summary as a MCMC object
summary(fit.JAGS.mcmc)
```

```
##
## Iterations = 10001:89921
## Thinning interval = 80
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean      SD  Naive SE Time-series SE
## deviance 12.7236 1.27180 4.022e-02      4.022e-02
## lambda    0.6201 0.14844 4.694e-03      2.407e-02
## psi       0.3000 0.00140 4.429e-05      4.429e-05
## tau       0.7912 0.04983 1.576e-03      8.174e-03
## theta     0.2364 0.10978 3.472e-03      1.788e-02
##
## 2. Quantiles for each variable:
##
##              2.5%     25%     50%     75%   97.5%
## deviance 11.79091 11.8904 12.2452 13.0302 16.1600
## lambda    0.34765  0.5065  0.6245  0.7321  0.8804
## psi       0.29735  0.2990  0.3000  0.3010  0.3027
## tau       0.71344  0.7523  0.7839  0.8302  0.8906
## theta     0.06721  0.1524  0.2263  0.3076  0.4782
```

```r
# Traceplots as a MCMC object
xyplot(fit.JAGS.mcmc,layout=c(2,3))
```




```
# Density plot as a MCMC object
densityplot(fit.JAGS.mcmc)
```



Part(d)

After we observe the results from part(c), the traceplots for the parameters seem fairly stationary, however, λ, τ, θ seems a bit less appropriate than ψ, but I think the variations are acceptable. When we identify the density plots, all parameters are in appropriate shapes as the priors we set up. Therefore, I think the model we constructed seems reasonable without any adjustments.

Part(e)

The estimated probability of Trump winning the 2024 US Election is 0.236, which is lower than 0.3. Since we set the prior that people that claimed to vote for Warren but vote for Trump are slightly higher than people who claimed to vote for Trump but vote for Warren, we would expect more Quasi Liberal to vote for Trump. However, the prior setting does not influence much due to the huge likelihood probability difference.

Question 3

Part(a)

$$\theta_1 = 1 + 0.065X_1 = 1 + 0.065 * 95 = 7.175$$
$$H_1 \sim Poisson(7.175)$$

```r
# Probability of Poisson distribution for at least 2 infections
P_atleast2 <- 1 - ppois(1, 7.175)
P_atleast2
```

```
## [1] 0.9937422
```

Therefore, the probability that Indonesia observes at least 2 infections in the given time period is 0.994

Part(b)

$$\theta_2 = 1 + 0.065X_2 = 1 + 0.065 * 150 = 10.75$$
$$H_2 \sim Poisson(10.75)$$

```r
# Probability of Poisson distribution for at least 28 infections
P_atleast28 <- 1 - ppois(27, 10.75)
P_atleast28
```

```
## [1] 8.378949e-06
```

Therefore, the probability that Singapore observes 28 or more infections in the given time period is approximately 0.