Lab 4.5

```
setwd("/Users/huiyuhu/Desktop/Study/UCLA_Biostat/BIOSTAT234/lab/Lab 4_5")
getwd()
```

```
## [1] "/Users/huiyuhu/Desktop/Study/UCLA_Biostat/BIOSTAT234/lab/Lab 4_5"
```

```
expit = function(a) exp(a)/(1+exp(a))
logit = function(a) log(a/(1-a))
#LOAD NECESSARY PACKAGES
library(R2jags)
```

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

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

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

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

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

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

```
library(knitr)
load("AddBurnin.RData")
library(lattice)
#install.packages("bayesplot")
library("bayesplot")
```

```
## This is bayesplot version 1.6.0
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```
#install.packages("ggmcmc")
library("ggmcmc")
```

```
## Warning: package 'ggmcmc' was built under R version 3.5.2
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: tidyr
```

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

```
library("ggplot2")
library(xtable)
# useful function
mysummary = function(invector) {
  c(mean(invector), sd(invector), quantile(invector, .025),
    quantile(invector,.975),
    length(invector[invector>0])/length(invector))
}
```

- Data

```
metadata = matrix(data = c(
    506,     21,    164,     20,
  20614,    222,   1575,     59,
  32279,    681,   3051,    212,
    234,     49,     59,     19,
    201,     28,     30,     11,
    568,     60,    116,     19,
   2035,    130,    549,     43,
    406,     32,     99,     17,
   2946,     55,    831,     83), byrow=T, ncol=4)
#metadata
colnames(metadata) = c("n0.HD", "y0.HD", "n1.HD", "y1.HD")
dim(metadata)
```

```
## [1] 9 4
```

```
kable(metadata) # table output in latex format
```

n0.HD	y0.HD	n1.HD	y1.HD
506	21	164	20
20614	222	1575	59
32279	681	3051	212
234	49	59	19
201	28	30	11
568	60	116	19
2035	130	549	43
406	32	99	17
2946	55	831	83
* brief	explorat	ion of t	he data

```
zzz = cbind(metadata[,2]/metadata[,1],metadata[,4]/metadata[,3])
ORs = (zzz[,2]/(1-zzz[,2]))/ (zzz[,1]/(1-zzz[,1]))
zzz = cbind(zzz, ORs)
zzz = round(zzz, 3)
kable(zzz)
```

		ORs
0.042	0.122	3.208
0.011	0.037	3.575
0.021	0.069	3.465

		ORs
0.209	0.322	1.793
0.139	0.367	3.577
0.106	0.164	1.658
0.064	0.078	1.245
0.079	0.172	2.423
0.019	0.100	5.833

```
# #Meta-analysis model 1
# sink("syncope1.txt")
# cat("
#     model
#     {
#     for( i in 1:npapers ) {
#       y0[i] ~ dbin(pie0[i],n0[i])
#       y1[i] ~ dbin(pie1[i],n1[i])
#       logit(pie0[i]) = alpha + beta[i] - delta[i]/2
#       logit(pie1[i]) = alpha + beta[i] + delta[i]/2
#       beta[i]  ~ dnorm(0 , sigmainv2)
#       delta[i] ~ dnorm(d0, tauinv2  )
#       OR[i]    = (pie1[i]/(1 - pie1[i])) / (pie0[i]/(1 - pie0[i]))
#     }
#     alpha ~ dnorm(a, b)
#     d0    ~ dnorm(0, d)
#     sigmainv2 ~ dgamma(c1,c2)
#     tauinv2   ~ dgamma(f1,f2)
#   sigma = 1/sqrt(sigmainv2)
#   tau   = 1/sqrt(tauinv2)
#     }
#     ",fill = TRUE)
# sink()
```

```
# Prior parameters
npapers = 9
a = -2.75  #
b = 1/2
d = 1/2
c1 = f1 = 3
c2 = f2 = 2


parameters= c(
  "pie0", "pie1", "alpha", "sigma",
  "tau", "d0", "OR"
  )

priordata = list( npapers = npapers, a=a, b=b, d = d, c1=c1, f1 = f1,
                  c2 = c2, f2 = f2, y0 = metadata[,2], n0 = metadata[,1],
                  y1 = metadata[,4], n1 = metadata[,3]
                  )

inits = rep(list(list(
  beta   = rep(0,npapers),
  delta   = rep(0,npapers),
  alpha = 0,
  d0 = 0,
  sigmainv2 = 1,
  tauinv2 = 1
)), 5)

run1 = jags(priordata, inits, parameters, "syncope1.txt",
            n.chains=5, n.iter=1100, n.burnin=0, n.thin=1)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 18
##    Unobserved stochastic nodes: 22
##    Total graph size: 172
##
## Initializing model
```

```
#proc.time()
names(run1)
```

```
## [1] "model"              "BUGSoutput"         "parameters.to.save"
## [4] "model.file"         "n.iter"             "DIC"
```

```
Output1=AddBurnin(run1$BUGSoutput$sims.array,burnin=100,n.thin=1)

names(Output1)
```

```
## [1] "Burnin.sims.array"  "Burnin.sims.matrix" "Burnin.Summary"
```

```
print(Output1$Burnin.Summary)
```

```
##               mu.vect     sd.vect         2.5%        97.5%    P>0
## OR[1]      3.28152307 1.0335121983   1.715494557   5.77698472 1.0000
## OR[2]      3.60069134 0.5131837821   2.711378533   4.70435930 1.0000
## OR[3]      3.47035047 0.2704095914   2.963527700   4.03165381 1.0000
## OR[4]      1.95055112 0.5916659000   1.033772061   3.34450206 1.0000
## OR[5]      3.30497252 1.2653032023   1.442442796   6.38243506 1.0000
## OR[6]      1.80925519 0.4886686551   1.027109176   2.91957439 1.0000
## OR[7]      1.32892143 0.2352890431   0.910679414   1.86005824 1.0000
## OR[8]      2.55748818 0.7833267186   1.366296538   4.39590962 1.0000
## OR[9]      5.60924452 0.9961772492   3.904893454   7.77106146 1.0000
## alpha     -2.27695241 0.5350856907  -3.195452794  -0.80552378 0.0000
## d0         0.95478813 0.2536144685   0.433063646   1.45236309 0.9996
## deviance 118.36136224 5.6953492359 108.953413863 130.77645354 1.0000
## pie0[1]    0.04217266 0.0086135175   0.027368291   0.06061087 1.0000
## pie0[2]    0.01084687 0.0007153002   0.009447197   0.01225802 1.0000
## pie0[3]    0.02111981 0.0007827218   0.019617365   0.02265842 1.0000
## pie0[4]    0.20286243 0.0256098397   0.155543823   0.25597454 1.0000
## pie0[5]    0.13858724 0.0232133823   0.095561150   0.18732663 1.0000
## pie0[6]    0.10334216 0.0124598803   0.080024313   0.12916601 1.0000
## pie0[7]    0.06312590 0.0054251345   0.053028993   0.07400500 1.0000
## pie0[8]    0.07795311 0.0126958729   0.055137861   0.10489972 1.0000
## pie0[9]    0.01944272 0.0025245322   0.014821342   0.02470850 1.0000
## pie1[1]    0.12105928 0.0244099469   0.076977324   0.17286601 1.0000
## pie1[2]    0.03780690 0.0045956400   0.029488436   0.04750392 1.0000
## pie1[3]    0.06955263 0.0044962975   0.060870361   0.07848301 1.0000
## pie1[4]    0.32290883 0.0557306643   0.220010473   0.43631814 1.0000
## pie1[5]    0.33275807 0.0756556381   0.198460889   0.49064192 1.0000
## pie1[6]    0.16899195 0.0325242971   0.109754674   0.23782061 1.0000
## pie1[7]    0.08145968 0.0113072750   0.060095136   0.10515455 1.0000
## pie1[8]    0.17232824 0.0351316012   0.109383178   0.24737339 1.0000
## pie1[9]    0.09840549 0.0100448298   0.079583901   0.11887253 1.0000
## sigma      0.99049262 0.2506500554   0.656887101   1.62147465 1.0000
## tau        0.69392954 0.1437903028   0.479870450   1.02292794 1.0000
```

```
colnames(Output1$Burnin.sims.matrix)
```

```
##  [1] "OR[1]"    "OR[2]"    "OR[3]"    "OR[4]"    "OR[5]"    "OR[6]"   
##  [7] "OR[7]"    "OR[8]"    "OR[9]"    "alpha"    "d0"       "deviance"
## [13] "pie0[1]"  "pie0[2]"  "pie0[3]"  "pie0[4]"  "pie0[5]"  "pie0[6]" 
## [19] "pie0[7]"  "pie0[8]"  "pie0[9]"  "pie1[1]"  "pie1[2]"  "pie1[3]" 
## [25] "pie1[4]"  "pie1[5]"  "pie1[6]"  "pie1[7]"  "pie1[8]"  "pie1[9]" 
## [31] "sigma"    "tau"
```

Problem 0.

- Repeat everything with the reparameterized model. Now how is the convergence?

```
# Here is a reparameterized model.
#Meta-analysis model
# sink("syncope1_repara.txt")
# cat("
#     model
#     {
#     for( i in 1:npapers ) {
#     y0[i] ~ dbin(pie0[i],n0[i])
#     y1[i] ~ dbin(pie1[i],n1[i])
#     logit(pie0[i]) = beta[i] - delta[i]/2
#     logit(pie1[i]) = beta[i] + delta[i]/2
#     beta[i]  ~ dnorm(alpha , sigmainv2)
#     delta[i] ~ dnorm(d0, tauinv2  )
#     OR[i]    = (pie1[i]/(1 - pie1[i])) / (pie0[i]/(1 - pie0[i]))
#     }
#     alpha ~ dnorm(a, b)
#     d0    ~ dnorm(0, d)
#     sigmainv2 ~ dgamma(c1,c2)
#     tauinv2   ~ dgamma(f1,f2)
#     sigma = 1/sqrt(sigmainv2)
#     tau   = 1/sqrt(tauinv2)
#     }
#     ",fill = TRUE)
# sink()
```

```
#repeat everything
run2 = jags(priordata, inits, parameters, "syncope1_repara.txt",
            n.chains=5, n.iter=1100, n.burnin=0, n.thin=1)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 18
##    Unobserved stochastic nodes: 22
##    Total graph size: 163
## 
## Initializing model
```

```
names(run2)
```

```
## [1] "model"              "BUGSoutput"         "parameters.to.save"
## [4] "model.file"         "n.iter"             "DIC"
```

```
Output2=AddBurnin(run2$BUGSoutput$sims.array,burnin=100,n.thin=1)

names(Output2)
```

```
## [1] "Burnin.sims.array"  "Burnin.sims.matrix" "Burnin.Summary"
```

```
print(Output2$Burnin.Summary)
```

```
##               mu.vect      sd.vect        2.5%       97.5%    P>0
## OR[1]      3.24160557 1.0299909582   1.735843541   5.66618994 1.0000
## OR[2]      3.58889589 0.5159162894   2.688746742   4.67949301 1.0000
## OR[3]      3.46787065 0.2795256549   2.957445802   4.03870996 1.0000
## OR[4]      1.92731910 0.5718167695   1.013914741   3.21876678 1.0000
## OR[5]      3.31213177 1.2691523085   1.446347513   6.32553279 1.0000
## OR[6]      1.79979438 0.5042180087   0.998355129   2.92420215 1.0000
## OR[7]      1.33314873 0.2308163784   0.932895927   1.82838180 1.0000
## OR[8]      2.53838142 0.7571747264   1.365101457   4.29412388 1.0000
## OR[9]      5.58668019 0.9804469243   3.922958119   7.69212282 1.0000
## alpha     -2.41636761 0.3119902127  -3.011689418  -1.79271173 0.0000
## d0         0.94926046 0.2509758465   0.457522434   1.43765452 0.9998
## deviance 118.69637779 5.8401599017 109.019209587 132.11021823 1.0000
## pie0[1]    0.04240143 0.0087615875   0.027057522   0.06151906 1.0000
## pie0[2]    0.01084800 0.0007120925   0.009498113   0.01226281 1.0000
## pie0[3]    0.02113158 0.0008077963   0.019559824   0.02274064 1.0000
## pie0[4]    0.20305719 0.0258179042   0.154280744   0.25591252 1.0000
## pie0[5]    0.13788852 0.0235572826   0.094015574   0.18711191 1.0000
## pie0[6]    0.10333854 0.0127448348   0.079656588   0.12940563 1.0000
## pie0[7]    0.06318436 0.0053455718   0.053283566   0.07394836 1.0000
## pie0[8]    0.07770189 0.0128294523   0.054704803   0.10458436 1.0000
## pie0[9]    0.01942317 0.0024913345   0.014779539   0.02458960 1.0000
## pie1[1]    0.12032911 0.0243786303   0.076817910   0.17138082 1.0000
## pie1[2]    0.03768993 0.0046277529   0.029258058   0.04730836 1.0000
## pie1[3]    0.06953253 0.0045922343   0.060835088   0.07871897 1.0000
## pie1[4]    0.32081173 0.0556798405   0.216890669   0.43474608 1.0000
## pie1[5]    0.33153625 0.0731258186   0.195013162   0.48328534 1.0000
## pie1[6]    0.16811242 0.0337963706   0.107689322   0.23822134 1.0000
## pie1[7]    0.08180668 0.0111608735   0.061420337   0.10468444 1.0000
## pie1[8]    0.17105766 0.0355763913   0.107862641   0.24561627 1.0000
## pie1[9]    0.09802257 0.0101840685   0.079390132   0.11888454 1.0000
## sigma      0.93285789 0.1933459369   0.642032846   1.39099752 1.0000
## tau        0.69337059 0.1463985631   0.471211014   1.03665480 1.0000
```

```
colnames(Output2$Burnin.sims.matrix)
```

```
##  [1] "OR[1]"    "OR[2]"    "OR[3]"    "OR[4]"    "OR[5]"    "OR[6]"
##  [7] "OR[7]"    "OR[8]"    "OR[9]"    "alpha"    "d0"       "deviance"
## [13] "pie0[1]"  "pie0[2]"  "pie0[3]"  "pie0[4]"  "pie0[5]"  "pie0[6]"
## [19] "pie0[7]"  "pie0[8]"  "pie0[9]"  "pie1[1]"  "pie1[2]"  "pie1[3]"
## [25] "pie1[4]"  "pie1[5]"  "pie1[6]"  "pie1[7]"  "pie1[8]"  "pie1[9]"
## [31] "sigma"    "tau"
```

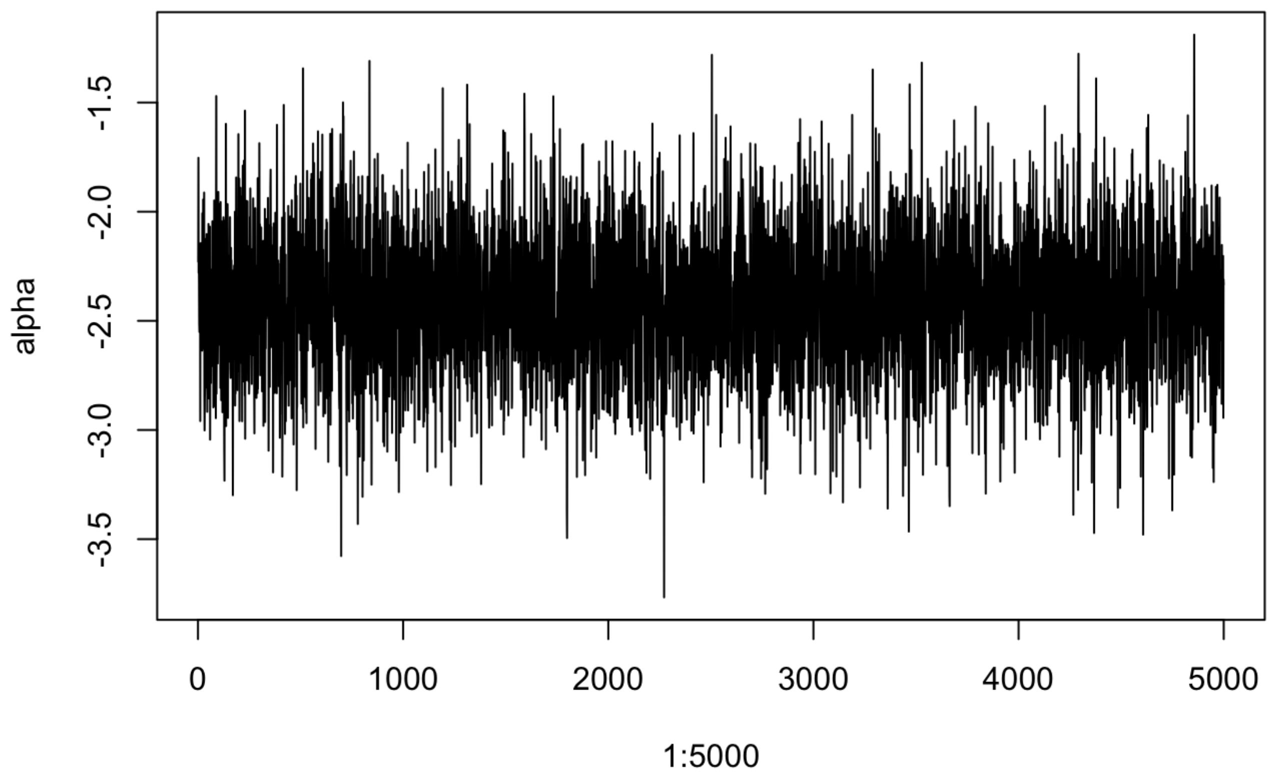
```
maindata = Output2$Burnin.sims.matrix[,c(10,11,31,32)]
dim(maindata)  #5000,4
```

```
## [1] 5000    4
```

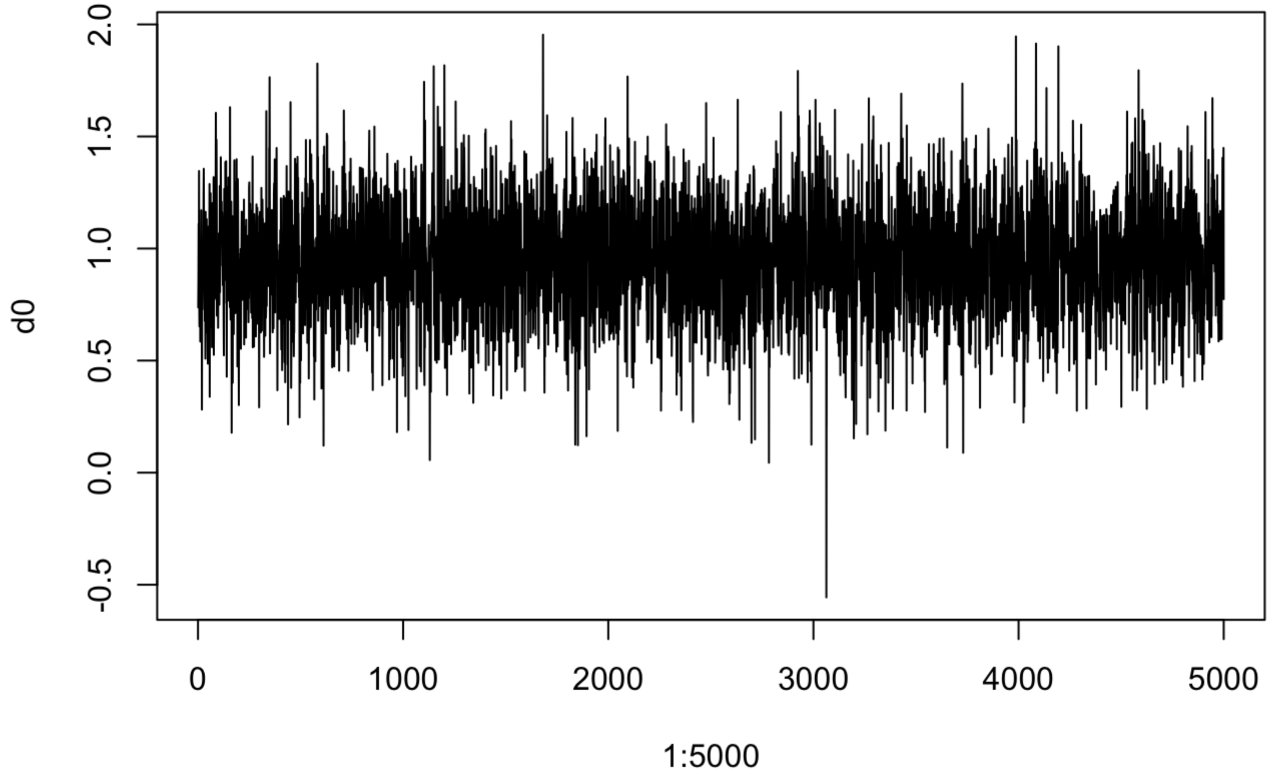
```
head(maindata)
```

```
##          alpha        d0     sigma       tau
## [1,] -2.229074 0.7376595 0.9077410 0.6865885
## [2,] -1.753210 0.8558480 0.9610270 0.8647577
## [3,] -2.279792 1.1099847 0.8063791 0.6833436
## [4,] -2.230678 0.8290282 0.7548588 0.7768187
## [5,] -2.314679 1.3458213 0.9682312 1.1083477
## [6,] -2.513681 0.7486087 1.0997126 0.5116249
```

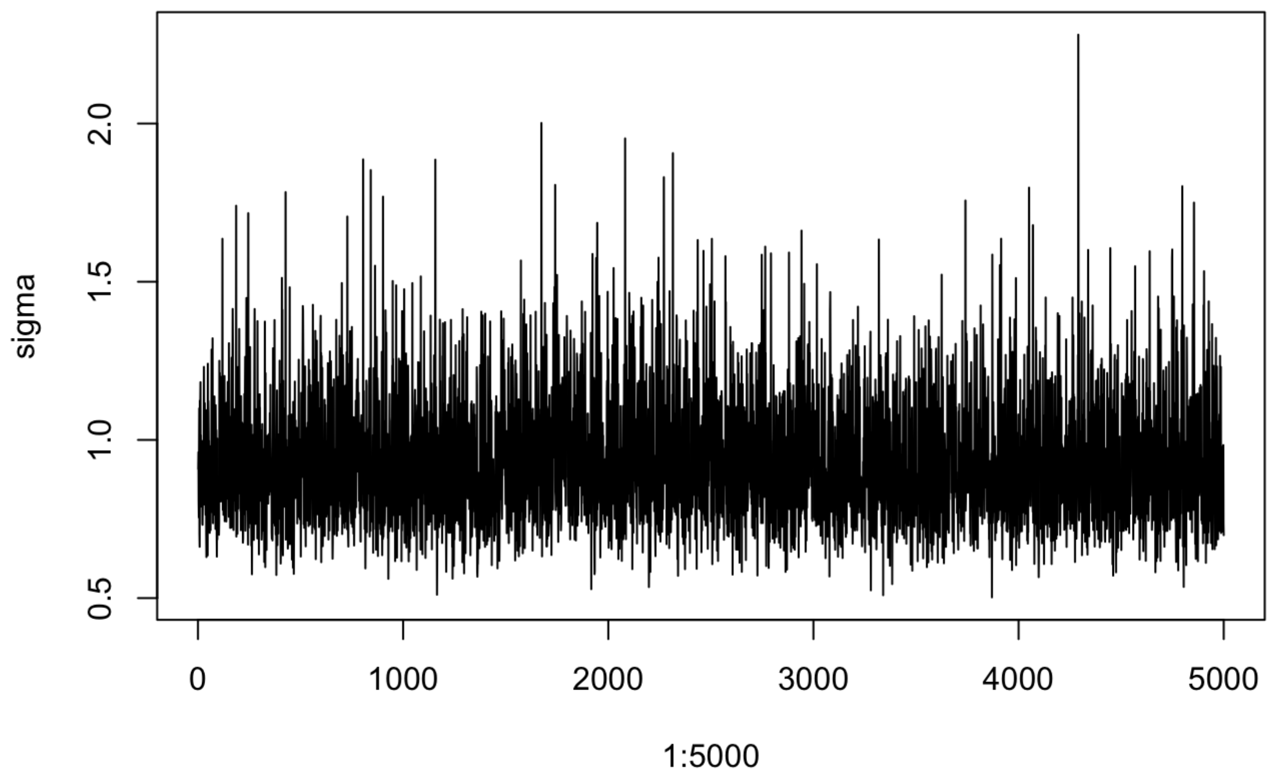
```
plot(1:5000, maindata[,1], ylab = "alpha", type="l")
```



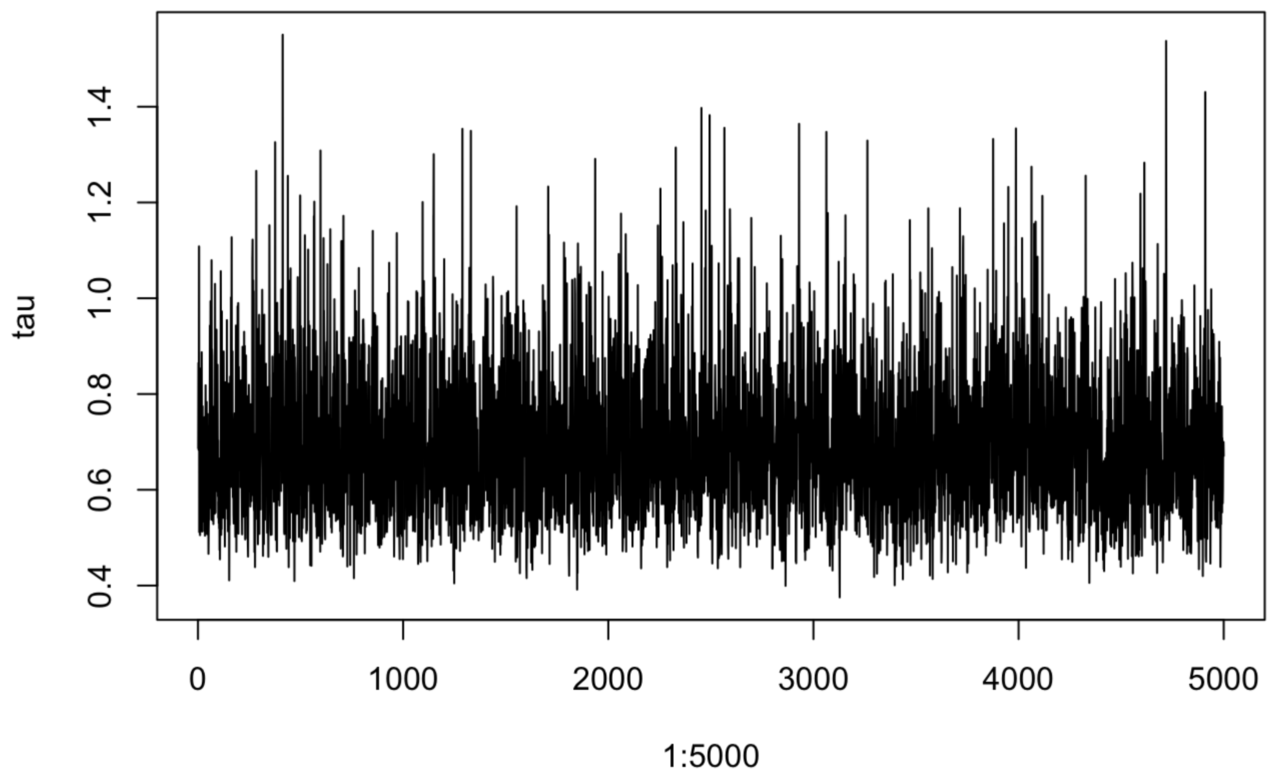

```
plot(1:5000, maindata[,2], ylab = "d0"   , type="l")
```




```
plot(1:5000, maindata[,3], ylab = "sigma", type="l")
```




```
plot(1:5000, maindata[,4], ylab = "tau"  , type="l")
```



- According to the new plots, the convergence became better, especially alpha.

Problem 1.

1a. What is the change in the model from first version to second version?

- The second version, we use $ _i^* N(,2)$.

1b. Is the model any different?

- Same model. Different parametrization, prior distribution for new betas also changed.

1c. Have any parameters changed meaning? Which are they, what are the changes?

- Yes. $$ has been changed.
- In the first model, $_i $ is a study level random effect modeling differences between studies, with mean of 0.
- In the second model, $_i^* $ has mean of α, which is an overall success parameter.

1d. Is the posterior of any of the parameters any different from the original model?

- The posterior of betas changed, since the prior distributions for betas were different.

1e. What are differences between the output of the first model and the second version?

- Second model showed better convergence. In addition, the α changed from -1.69 (model 1) to -2.41 (model 2).

Problem 2.

2a. What is the single parameter are we most interested in? Has it changed meaning between the two model versions? Use model 2 for the remainder of this problem.

- The key parameter that we are interested in is δ_0. δ_0 is the average, across papers, on the logit scale, of the log odds of the treatment (covariate) effect.
- The meaning of δ_0 has not been changed.

2b. What is your conclusion? Report your conclusion as an Odds Ratio and 95% interval. Do people with prior Heart Disease have better, the same or worse outcomes after visiting the emergency room for syncope compared to those without heart disease? Give both a quantitative and a qualitative answer.

- The OR mean is 2.68. The 95% interval is (1.59, 4.20). Prior of δ_0 is centered at zero. ORs greater than 1 indicate greater odds of a bad outcome, so people with prior Heart Disease have worse outcomes after visiting the emergency room for syncope compared to those without heart disease.

Problem 3.

Use the 2nd model for this problem. As sensitivity analysis for a meta-analysis, we rerun the analysis omitting each paper in turn. If there are n papers contributing data to the analysis, we run n additional analyses. A paper, that when omitted, changes our conclusions substantially is considered an influential paper.

```
n <- 8
parameters= c(
  "pie0", "pie1", "alpha", "sigma",
  "tau", "d0", "OR"
  )
inits = rep(list(list(
        beta   = rep(0, n),
        delta   = rep(0, n),
        alpha = 0,
        d0 = 0,
        sigmainv2 = 1,
        tauinv2 = 1
)), 5)

d_0 <- matrix(0,nrow = 9,ncol = 5)

for (i in 1:9) {
        metadata1 <- metadata[-i,]
        priordata1 = list(npapers = n, a=a, b=b, d = d, c1=c1, f1 = f1,
                          c2 = c2, f2 = f2, y0 = metadata1[,2], n0 = metadata1[,1],
                          y1 = metadata1[,4], n1 = metadata1[,3]
        )
        run3 = jags(priordata1, inits, parameters, "syncope1_repara.txt",
                    n.chains=5, n.iter=1100, n.burnin=0, n.thin=1)

        Output3=AddBurnin(run3$BUGSoutput$sims.array,burnin=100,n.thin=1)
        d01 = Output3$Burnin.sims.matrix[,10]
        d_0[i,] = round(mysummary(d01), 3)
}
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 20
##    Total graph size: 147
##
## Initializing model
```

```
colnames(d_0) = c("mean", "sd", "2.5%", "97.5%", "p")
rownames(d_0) = c("paper1", "paper2", "paper3", "paper4", "paper5", "paper6", "paper7",
"paper8", "paper9")
```

3a. Report a table of the most important inference and how it changes as we delete each paper in turn. Clearly label and format your table.

```
kable(d_0)
```

	mean	sd	2.5%	97.5%	p
paper1	0.933	0.274	0.387	1.455	0.999
paper2	0.908	0.268	0.354	1.425	0.999
paper3	0.907	0.272	0.362	1.427	0.999

	mean	sd	2.5%	97.5%	p
paper4	0.994	0.270	0.450	1.524	0.999
paper5	0.923	0.269	0.395	1.455	0.998
paper6	1.003	0.270	0.452	1.533	0.999
paper7	1.049	0.260	0.523	1.548	1.000
paper8	0.954	0.272	0.407	1.489	0.998
paper9	0.848	0.260	0.329	1.340	0.999

3b. Which paper is most influential?

- The paper 9 is most influential

3c. What is your conclusion? Is the final inference sensitive to omitting individual papers?

- Not very sensitive.