Biostat234 LAB 1

Name: Huiyu Hu

Questions:

Q1

What information is in the chapter 5 tables of the jags user manual?

- Tables listed the functions with their description as well as how to use them.
- Table 5.1: Base functions listed in reverse order of precedence - power, arithmetic, comparison, logical
- Table 5.2 - 5.7: Functions in the bugs module, including scalar, probability, link, scalar-valued, matrix-valued, canonical functions.

Q2

What information is in the distributions chapter of the jags user manual? Recite briefly the tables in this chapter.

- In bugs module, real-valued (table 6.1), discrete-valued (table 6.2) and multivariate (table 6.3) distributions was introduced with name, usage, density and range. Table 6.4 describes some distrubutions with aliases.

Q3.

See the WinBUGS examples at http://www.mrc-bsu.cam.ac.uk/wp-content/uploads/WinBUGS_Vol1.pdf (http://www.mrc-bsu.cam.ac.uk/wp-content/uploads/WinBUGS_Vol1.pdf) . (Or: install WinBUGS on your computer! See the Help menu, Examples volume I, roughly 9th example down.) What models (plural) does the Stacks example use? Specify each model. (There are 6 models).

- There are 6 models with different error structure:
- linear regression:

1. yi ~ Normal(μi, τ), normal error model
2. yi ~ Double exp(μi, τ), double exponential error model
3. yi ~ t(μi, τ, d), t distribution with df = 4

- Ridge-regression:

4. with normally distributed error, normal error model
5. double-exponential distribution error
6. t-distributed (t4) error, t distribution with df = 4

Q4.

Turn in properly formatted output from your regression model.

```
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
```

```
setwd("/Users/huiyuhu/Desktop/Study/UCLA_Biostat/BIOSTAT234/lab/Lab 1")
getwd()
```

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

```
x     = c(1,2,3,4,5)
y     = c(1,3,3,3,5)
N     = 5
x.bar = 3
jags.data = list("x","y","N","x.bar")
jags.params = c("alpha", "beta", "tau", "sigma")
jags.inits = function(){
    list("alpha" = 0, "beta" = 1, "tau" = 1)
}
lab1.sim = jags(jags.data, jags.inits, jags.params,
              model.file = "model_origin.txt",
              n.chains = 3, n.iter = 11000, n.burnin = 1000)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 36
##
## Initializing model
```

- Table showed that the mean and sd of the alpha, beta, deviance, sigma, tau.

```
knitr::kable(lab1.sim$BUGSoutput$summary[,c("mean","sd","2.5%","50%","97.5%")])
```

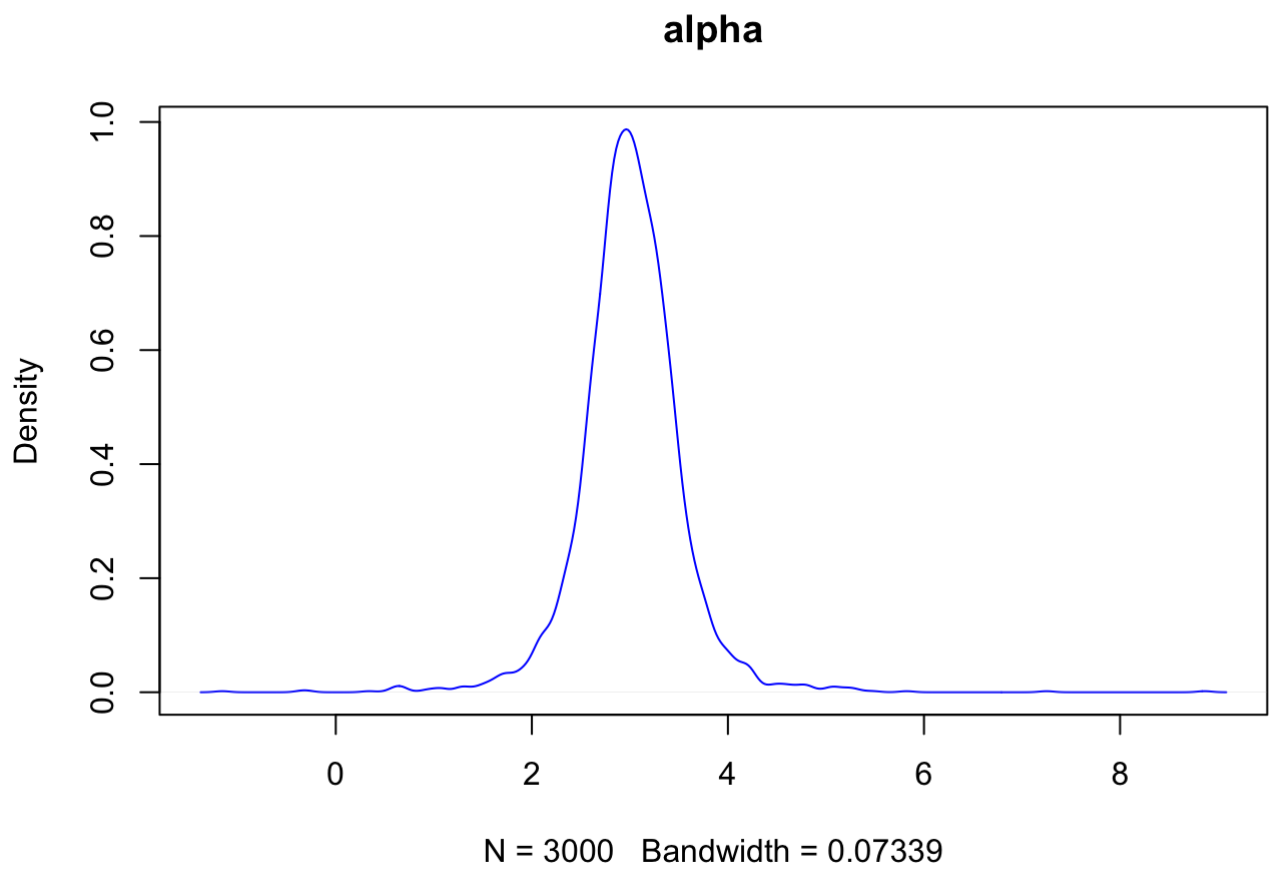
	mean	sd	2.5%	50%	97.5%
alpha	3.0114052	0.5325039	2.0199693	3.0045249	4.035463
beta	0.8258689	0.3053191	0.2353453	0.8144974	1.475537
deviance	12.7387690	3.2256244	8.8089462	11.9875748	20.758951
sigma	0.9914065	0.5675100	0.4607261	0.8523066	2.306064
tau	1.6651792	1.2280529	0.1880430	1.3766016	4.711015

- Graphs discribing the distribution of plpha beta tau sigma were shown as below.

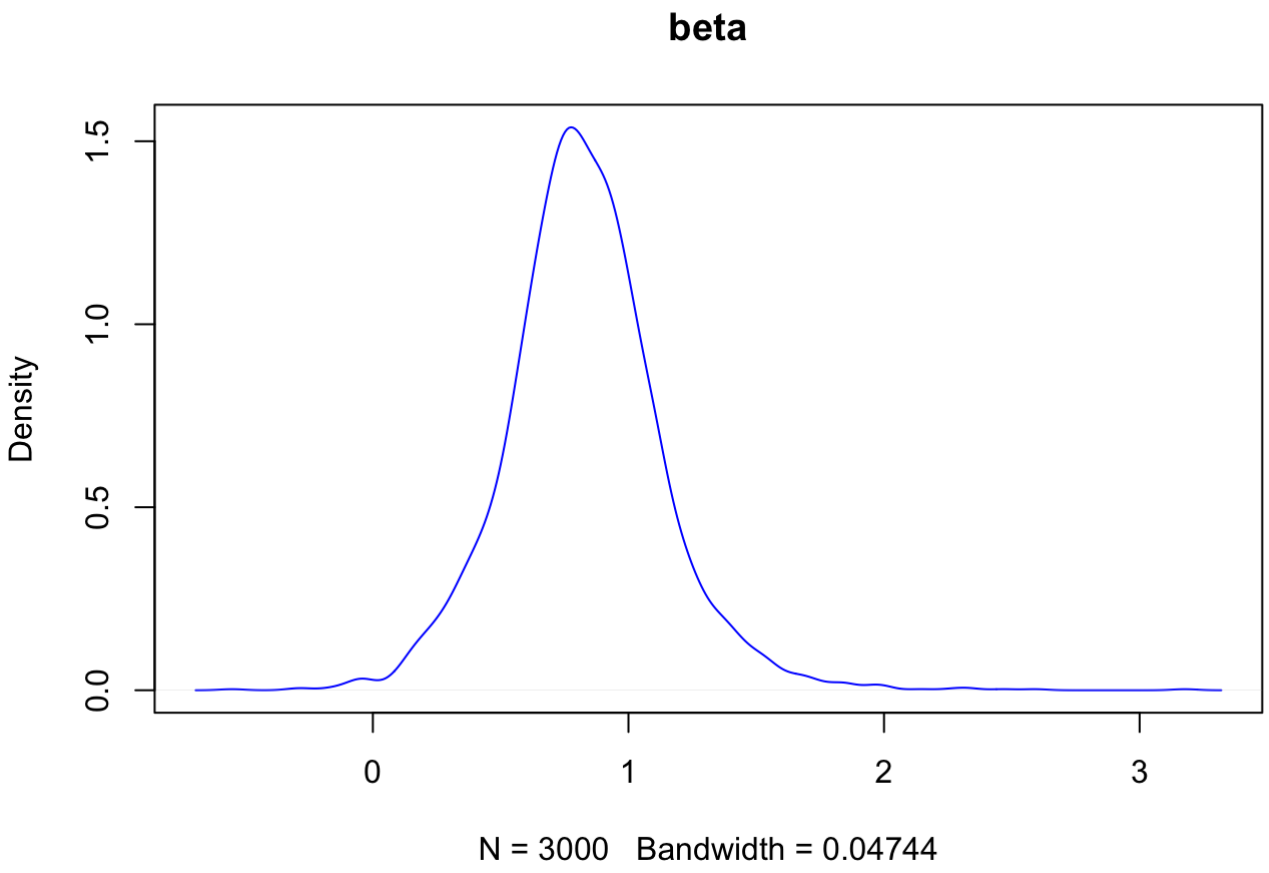
```
temp1 = lab1.sim$BUGSoutput$sims.matrix
head(temp1[,1])
```

```
## [1] 2.609909 2.504498 2.999217 2.441233 2.832345 2.939019
```

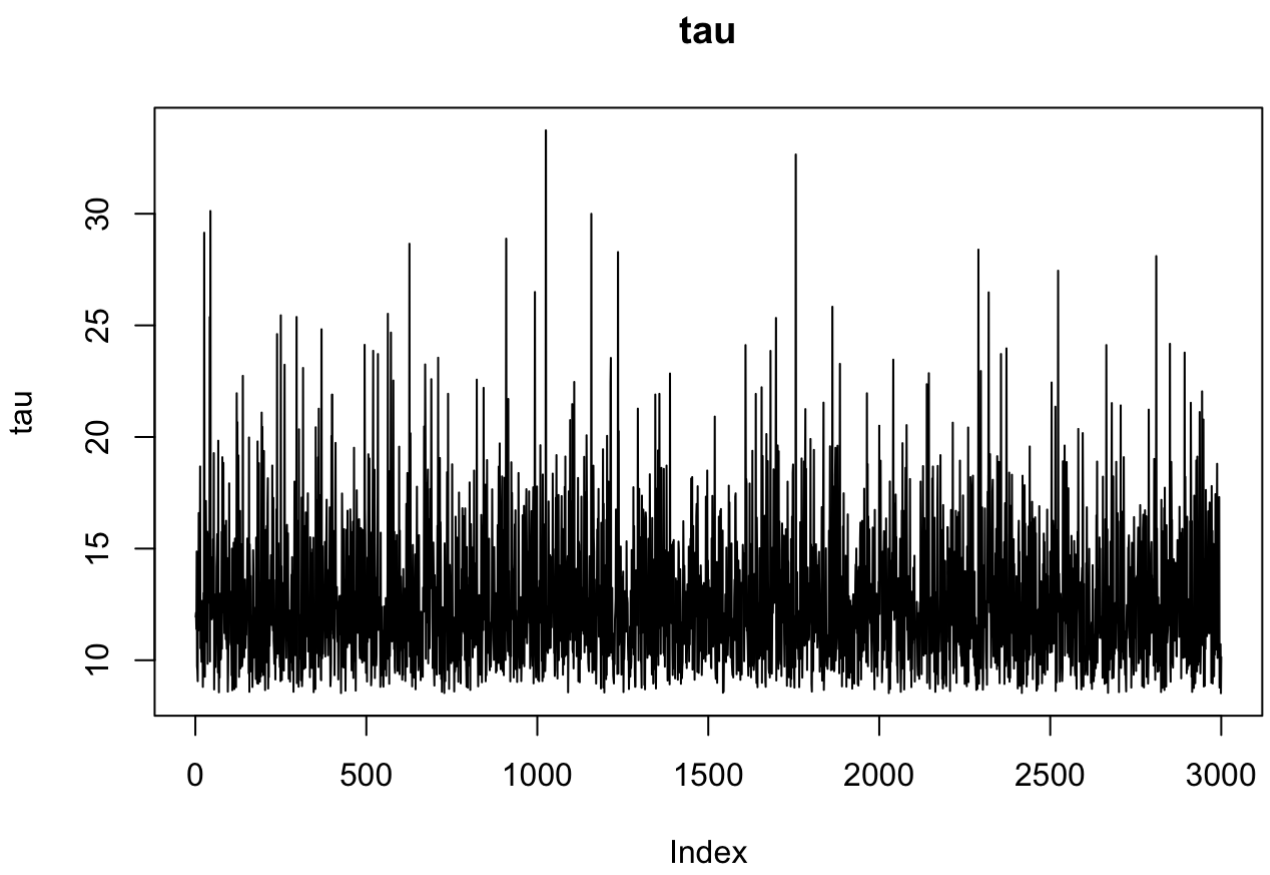
```
alpha = temp1[,1]
beta  = temp1[,2]
tau   = temp1[,3]
sigma = temp1[,4]

plot(density(alpha), main = "alpha", col = "blue")
```



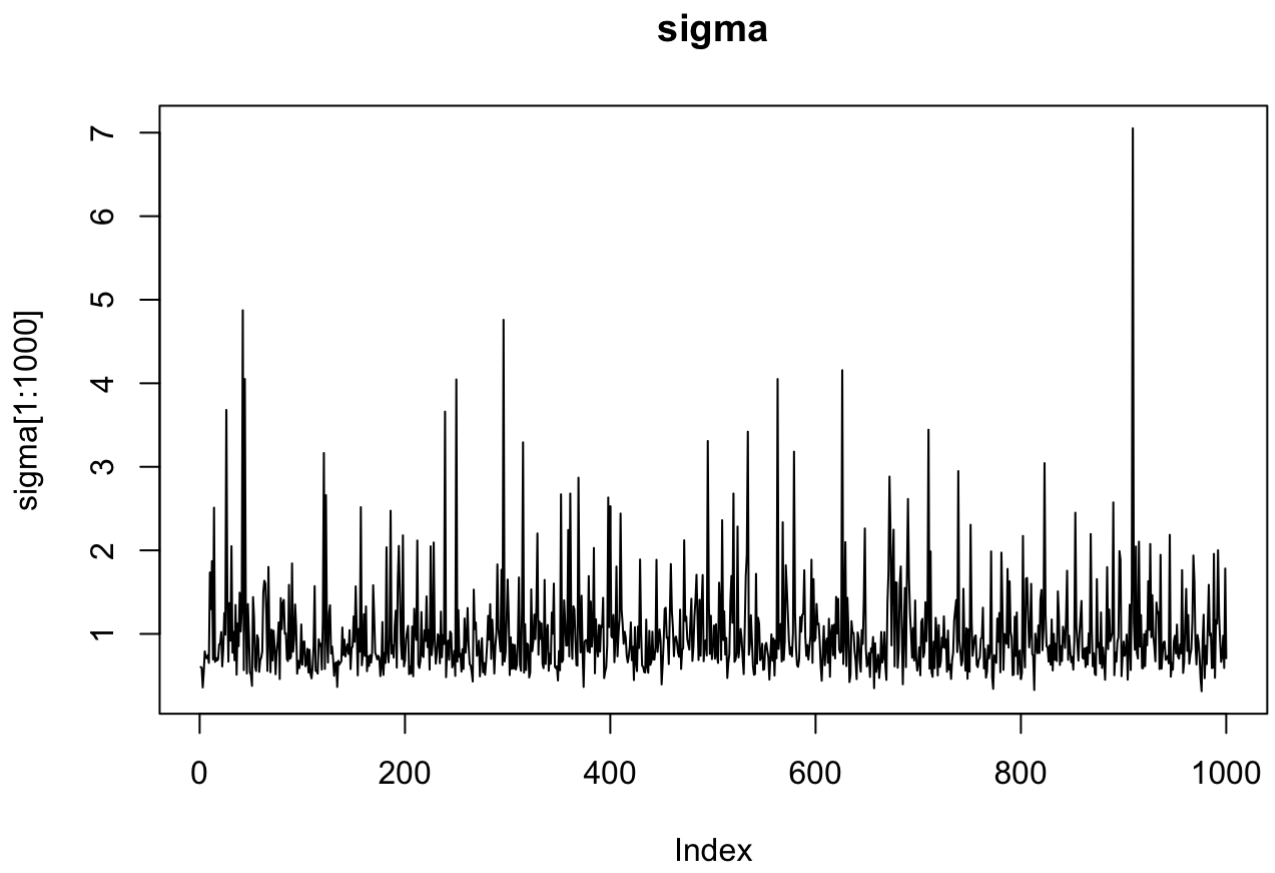
```
plot(density(beta), main = "beta", col = "blue")
```




```
plot(tau , type = "l", main = "tau")
```




```
plot(sigma[1:1000] , type = "l", main = "sigma")
```



Q5.

Change the prior precision for beta to 100, 10, 1, .1, .01, .001. The prior precision is the number 100 in the statement beta ~ dnorm(1,100) in your model program. Run the model for each of these values. What happens to the estimate of beta as the prior precision changes? To answer this,

- Change the prior precision:

```
# The model has been created in file.
# Model1: 100
# Model2: 10
# Model3: 1
# Model4: 0.1
# Model5: 0.01
# Model6: 0.001

# x     = c(1,2,3,4,5)
# y     = c(1,3,3,3,5)
# N     = 5
# x.bar = 3
# jags.data = list("x","y","N","x.bar")
# jags.params = c("alpha", "beta", "tau", "sigma")
# jags.inits = function(){
#     list("alpha" = 0, "beta" = 1, "tau" = 1)
# }

list <- list.files(pattern = "model\\d.txt")
hw1.sims <- list()
for (i in 1:length(list)) {
   hw1.sims[[i]] = jags(jags.data, jags.inits, jags.params,
                        model.file = list[i], n.chains = 3,
                        n.iter = 11000, n.burnin = 1000)
 }
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 37
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 37
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 36
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 37
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 37
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 5
##    Unobserved stochastic nodes: 3
##    Total graph size: 37
##
## Initializing model
```

a. Report a(n appropriately formatted) table of the posterior means and sds as a function of the prior precision.

```
beta_mean <- unlist(lapply(hw1.sims, function(x) x$BUGSoutput$summary["beta", "mean"]))
beta_sd <- unlist(lapply(hw1.sims, function(x) x$BUGSoutput$summary["beta", "sd"]))
table <- cbind(beta_mean, beta_sd)
prior_precision <- c(100, 10, 1, 0.1, 0.01, 0.001)
table1 <- cbind(prior_precision, table)
knitr::kable(table1)
```

prior_precision	beta_mean	beta_sd
1e+02	0.9672706	0.0947497
1e+01	0.8827067	0.2075913
1e+00	0.8207104	0.2980297
1e-01	0.8111712	0.3629223
1e-02	0.8101121	0.3632243
1e-03	0.8026536	0.7137336

b. As the prior precision goes to +infinity, what do you suppose the limit of the values of the estimate and sd are?

- Table above shows that as the tau decrease from 100 to 0.001, the beta also changes. Based on the trends showed on table above, as the prior precision (tau) goes to +infinity, beta is likely approaches 1 and sd of beta is likely to be near 0.
- Explaination: When prior precision go to +infinity, the variance becomes very low. Therefore, estimated beta will be centered at around 1 (since beta ~ dnorm(1, prior precision)).

c. The least squares estimate of beta is .8. What is the limit of the estimate as the prior precision goes to zero?

- According to previous result, as the prior precision goes to zero, the limit of the estimate will also be around ** 0.8 **.