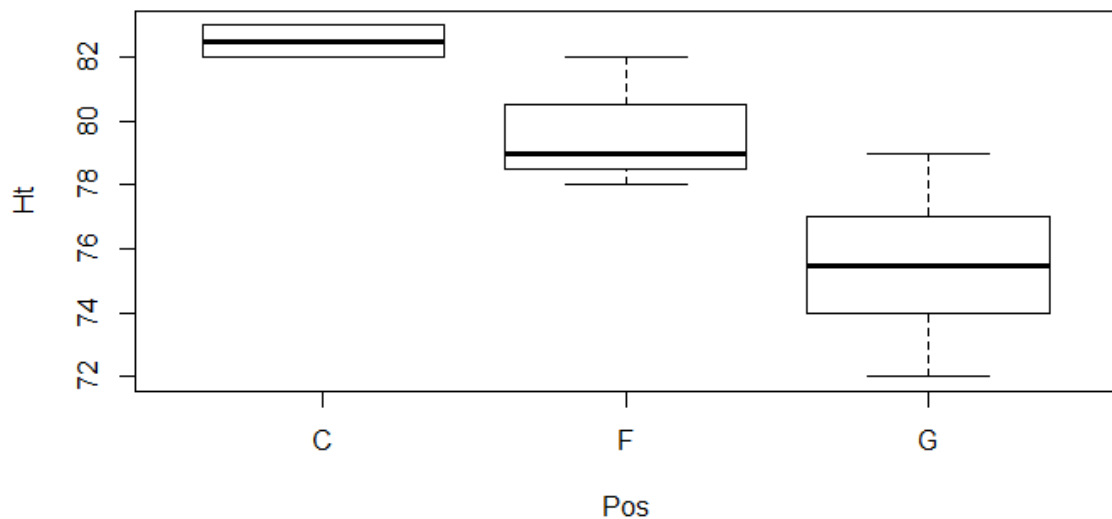


Assignment 6

1.

```
> players = read.csv(file="illinimensbb.csv", header = TRUE)
> plot(Ht ~ Pos, data = players)
> head(players)
   X      Player Ht Pos MIN FGM FGA BLK
1  0      DJ williams 79  G 150  17  40   5
2  1      Jaylon Tate 75  G 408  18  47   2
3  2      Kipper Nichols 78  F 286  39  86   7
4  3      Te'Jon Lucas 72  G 642  47 115   1
5  5 Jalen Coleman-Lands 75  G 850  94 257   4
6 12      Leron Black 79  F 627 101 223   3
> levels(players$Pos)
[1] "C" "F" "G"
> unclass(players$Pos)
[1] 3 3 2 3 3 2 3 3 1 3 1 3 3 2 3
attr(,"levels")
[1] "C" "F" "G"
```



Yes, there is a relationship between position and height, which matches the common sense of basketball. C has relatively high height, F has relatively medium height, and G has relatively low

height.

2.(a)

```
model {  
  for (i in 1:length(FGM)) {  
    FGM[i] ~ dbin(prob[i], FGA[i])  
    logit(prob[i]) <- betapos[pos[i]] + betaheight *  
heightscaled[i]  
    FGMrep[i] ~ dbin(prob[i], FGA[i])  
  }  
  for (j in 1:max(pos)) {  
    betapos[j] ~ dt(0, 0.01, 1)  
  }  
  betaheight ~ dt(0, 0.16, 1)  
}  
  
> d1 <- list(FGM = players$FGM, FGA = players$FGA, pos = un  
class(players$Pos), heightscaled = as.vector(scale(player  
s$Ht, scale = 2*sd(players$Ht))))  
> inits1 <- list(list(betapos=c(10,10,10), betaheight=10),  
list(betapos=c(10,10,-10), betaheight=-10), list(betapos=  
c(10,-10,10), betaheight=-10), list(betapos=c(10,-10,-10),  
betaheight=10))  
> library(rjags)  
Loading required package: coda  
Linked to JAGS 4.3.0  
Loaded modules: basemod,bugs  
> m1 <- jags.model("players.bug",d1, inits1, n.chains=4, n.  
adapt=10000)  
Compiling model graph
```

```

    Resolving undeclared variables
    Allocating nodes
  Graph information:
    Observed stochastic nodes: 15
    Unobserved stochastic nodes: 19
    Total graph size: 116

```

```

Initializing model

```

```

| ++++++ | 10
0%
> update(m1, 10000)
| ***** | 10
0%
> x1 <- coda.samples(m1, c("betapos","betaheight"), n.iter
=20000)
| ***** | 10
0%
> gelman.diag(x1, autoburnin=FALSE)
Potential scale reduction factors:

```

	Point est.	Upper C.I.
betaheight	1	1
betapos[1]	1	1
betapos[2]	1	1
betapos[3]	1	1

```

Multivariate psrf

```

```

1
> x1 <- coda.samples(m1,c("betapos","betaheight","prob","
FGMrep"),n.iter=20000)
| ***** | 10
0%
> effectiveSize(x1[,1:4])
FGMrep[1] FGMrep[2] FGMrep[3]
55216.07 78660.93 54402.33
FGMrep[4]
36619.37

```

```

(b)

```

```

> summary(x1[,16:19])

```

```

Iterations = 40001:60000

```

Thinning interval = 1
Number of chains = 4
Sample size per chain = 20000

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

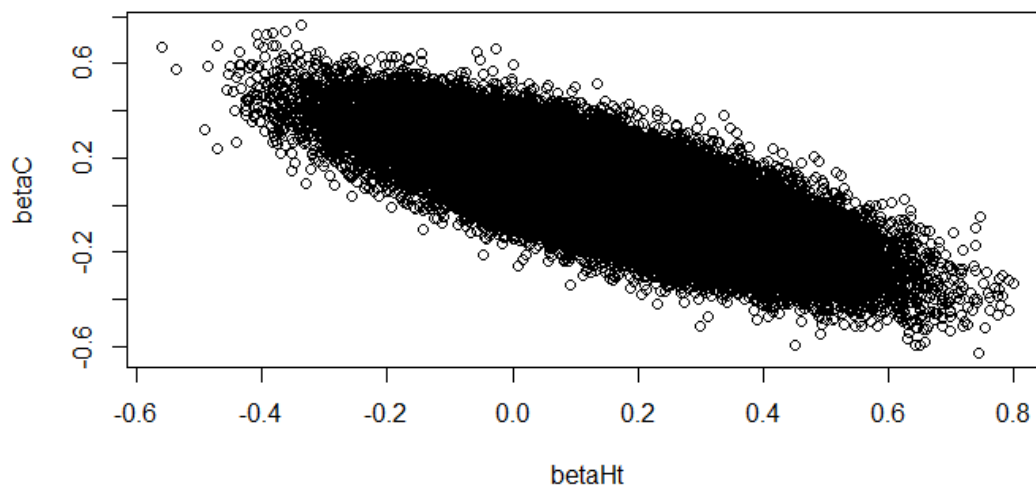
	Mean	SD	Naïve SE	Time-series SE
betaheight	0.15676	0.17142	0.0006061	0.0018037
betapos[1]	0.05114	0.16459	0.0005819	0.0016183
betapos[2]	-0.20796	0.11431	0.0004041	0.0009169
betapos[3]	-0.34284	0.06868	0.0002428	0.0004893

2. Quantiles for each variable:

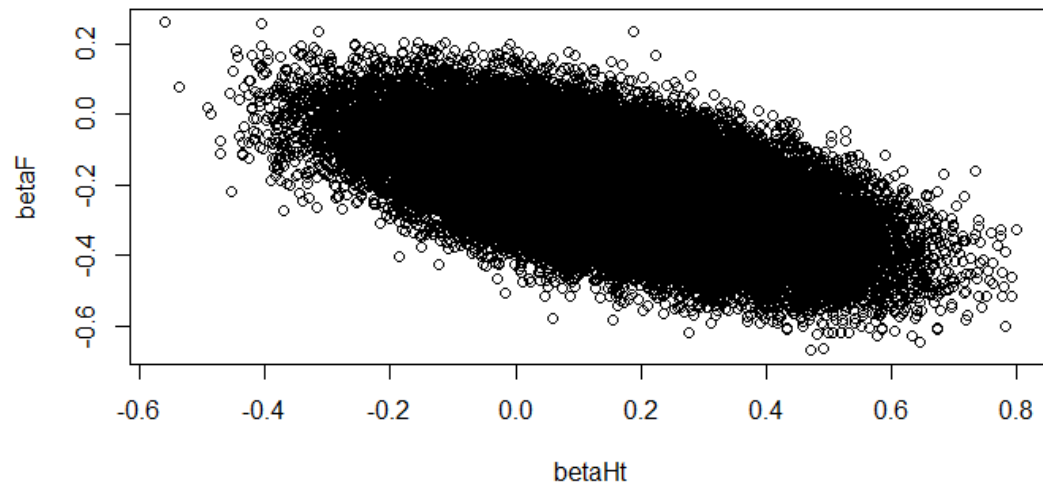
	2.5%	25%	50%	75%	97.5%
betaheight	-0.1801	0.04137	0.15757	0.2723	0.49318
betapos[1]	-0.2698	-0.05999	0.05031	0.1622	0.37764
betapos[2]	-0.4320	-0.28499	-0.20787	-0.1319	0.01752
betapos[3]	-0.4780	-0.38891	-0.34283	-0.2963	-0.20831

(c)(i)

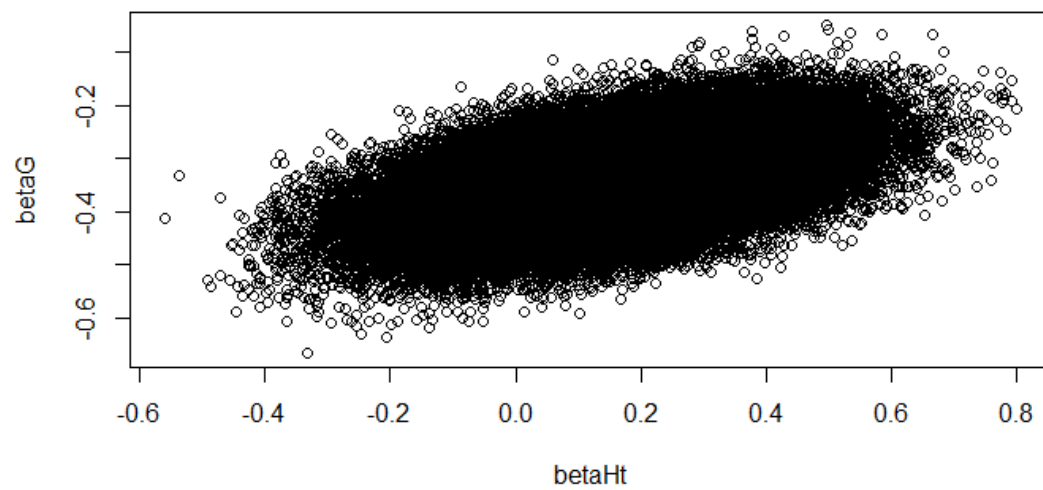
```
> plot(unlist(x1[,16]),unlist(x1[,17]), xlab = "betaHt", ylab="betaC")  
> plot(unlist(x1[,16]),unlist(x1[,18]), xlab = "betaHt", ylab="betaF")  
> plot(unlist(x1[,16]),unlist(x1[,19]), xlab = "betaHt", ylab="betaG")
```



(ii)



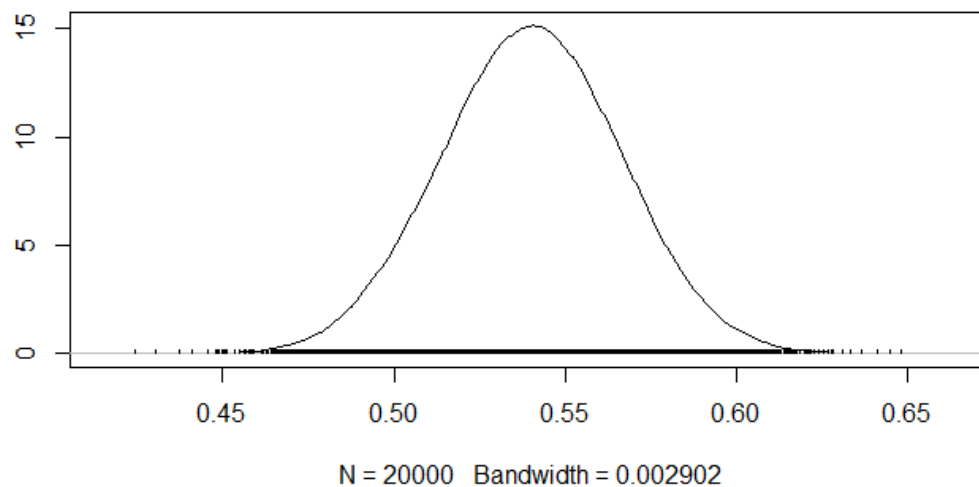
(iii)



β_C and β_{Ht} are highly negative correlated, β_F and β_{Ht} are moderately negative correlated, β_G and β_{Ht} are moderately positive correlated.

(d)

```
> densplot(x1[,28])
```



(e)

```
> mean(as.numeric(unlist(x1[,17]))>as.numeric(unlist(x1[,
19])))
[1] 0.9758125
```

Posterior probability $\beta_C > \beta_G$ is 0.9758125.

```
> clargerG = mean(as.numeric(unlist(x1[,17]))>as.numeric
(unlist(x1[,19])))
> clargerG / (1-clargerG)
[1] 40.34367
```

This is strong data evidence of $\beta_C > \beta_G$.

(f)

```
> probs <- as.matrix(x1)[,paste("prob[",1:nrow(players),"
", sep="")]
> FGMrep <- as.matrix(x1)[,paste("FGMrep[",1:nrow(player
s),"", sep="")]
> Tchi <- numeric(nrow(FGMrep))
> Tchirep <- numeric(nrow(FGMrep))
> for (s in 1:nrow(FGMrep)) {
+ Tchi[s] <- sum((players$FGM - players$FGA * probs[s,])^2
/ (players$FGA*probs[s,]*(1-probs[s,])))
+ Tchirep[s] <- sum((FGMrep[s,]-players$FGA*probs[s,])^2 /
(players$FGA*probs[s,]*(1-probs[s,])))
+ }
```

```
> mean(Tchirep >= Tchi)
[1] 0.7781125
```

It does not indicate any evidence of problems.

(g)

```
> dic.samples(m1,200000)
| ***** | 10
0%
Mean deviance: 77.92
penalty 3.987
Penalized deviance: 81.9
```

Effective number of parameters: about 4

Plummer's DIC: 81.9

The effective number of parameters is almost the same as the actual number of regression coefficients.

(h)

```
model {
  for (i in 1:length(FGM)) {
    FGM[i] ~ dbin(prob[i], FGA[i])
    logit(prob[i]) <- betapos[pos[i]]
    FGMrep[i] ~ dbin(prob[i], FGA[i])
  }
  for (j in 1:max(pos)) {
    betapos[j] ~ dt(0, 0.01, 1)
  }
}

> d1 <- list(FGM = players$FGM, FGA = players$FGA, pos = un
class(players$Pos))
```

```
> inits1 <- list(list(betapos=c(10,10,10)),list(betapos=c
(10,10,-10)), list(betapos=c(10,-10,10)), list(betapos=c
(10,-10,-10)))
> m1 <- jags.model("players.bug", d1, inits1, n.chains=4,
n.adapt=10000)
```

```
Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 15
  Unobserved stochastic nodes: 18
  Total graph size: 69
```

Initializing model

```
|+++++| 10
0%
> update(m1,10000)
|*****| 10
0%
> dic.samples(m1,200000)
|*****| 10
0%
Mean deviance: 77.77
penalty 3.008
Penalized deviance: 80.78
```

According to Plummer's DIC, there is not much difference between this new model and the original model, but new model is slightly better. Height variable might not be an important explanatory variable.

3.(a)

```
model {
  for (i in 1:length(FGM)) {
```



```

FGM[i] ~ dpois(lambda[i])

log(lambda[i]) <- betapos[pos[i]] + betaHt *
heightstand[i]

FGMrep[i] ~ dpois(lambda[i])
}
for (j in 1:max(pos)) {
  betapos[j] ~ dnorm(0, 0.0001)
}
betaHt ~ dnorm(0, 0.0001)
}

```

(b)

```
> summary(x2[,16:19])
```

```

Iterations = 40001:50000
Thinning interval = 1
Number of chains = 4
Sample size per chain = 10000

```

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

	Mean	SD	Naïve SE
betaHt	0.2611	0.06253	0.0003127
betapos[1]	4.1728	0.12058	0.0006029
betapos[2]	4.1100	0.08258	0.0004129
betapos[3]	3.9926	0.05261	0.0002630

	Time-series SE
betaHt	0.0009979
betapos[1]	0.0018083
betapos[2]	0.0009525
betapos[3]	0.0005481

2. Quantiles for each variable:

	2.5%	25%	50%
betaHt	0.1408	0.2183	0.2606
betapos[1]	3.9348	4.0914	4.1742
betapos[2]	3.9446	4.0550	4.1109
betapos[3]	3.8878	3.9574	3.9930

	75%	97.5%
betaHt	0.3035	0.3849
betapos[1]	4.2553	4.4063
betapos[2]	4.1667	4.2693
betapos[3]	4.0277	4.0949

(c)

```
> quandata = quantile(unlist(x2[,16]),c(0.025,0.975))
> exp(quandata)
      2.5%      97.5%
1.151168 1.469511
```

Yes, the interval is between 115% and 147%, which indicates greater height is associated with a higher rate of blocking shots which is consistent with the common sense of basketball.

(d)

```
> lambdas <- as.matrix(x2)[,paste("lambda[",1:nrow(players),"]",sep="")]
> FGMrep <- as.matrix(x2)[,paste("FGMrep[",1:nrow(players),"]",sep="")]
> Tchi <- numeric(nrow(FGMrep))
> Tchirep <- numeric(nrow(FGMrep))
> for(s in 1:nrow(FGMrep)){
+ Tchi[s] <- sum((players$FGM - lambdas[s,])^2 / lambdas[s,])
+ Tchirep[s] <- sum((FGMrep[s,] - lambdas[s,])^2 / lambdas[s,])
+ }
> mean(Tchirep >= Tchi)
[1] 0
```

It is substantial evidence of a problem, probably overdispersion.

(e)(i)

```

> repcomp <- matrix(0,15,1)
> for(s in 1:15){
+ repcomp[s] = mean(as.numeric(unlist(x2[,s]))>=players$FGM[s])
+ }
> a <- cbind(as.matrix(players$Player),as.matrix(repcomp))
> a<- as.table(a)
A DJ williams      1
B Jaylon Tate      1
C Kipper Nichols   0.99885
D Te'Jon Lucas     0.07535
E Jalen Coleman-Lands 0
F Leron Black      0.001475
G Tracy Abrams     0
H Malcolm Hill     0
I Maverick Morgan   0.00015
J Aaron Jordan     1
K Mike Thorne Jr.  1
L Samson Oladimeji 1
M Clayton Jones    1
N Michael Finke    0.67795
O Alex Austin      1

```

(ii)

Jalen Coleman-Lands, Leron Black, Tracy Abrams, Malcolm Hill, Maverick Morgan

(iii)

7 out of 15 players got probability 1.

Blocks: 5, 2, 1, 8, 0, 0, 0

Minutes played: 150, 408, 165, 318, 8, 11, 20.

This is not surprising because those seven players do not have enough time to play so that they cannot show their block skills. For 2 of them who played over 300 minutes, some blocks were got but not many enough to be larger than the simulated blocks.