

Homework 3

Christian Tillich

October 11, 2017

Setup 1

```
library(rjags)
library(jagstools)

# columns dfree , age, beck, ivhx, ndrugtx, race, treat, site
df <- read.table("../data/druguse.txt",header=T)
# create categorical predictor
df$ivhx.gr <- factor(df$ivhx)
# classical fit
CF <- glm(dfree ~ age+beck+ivhx.gr+ndrugtx+race+treat+site, data=df, family=binomial(link="logit"))
summary(CF) %>% coef %>% kable

# JAGS
code <- "
model {
  for (i in 1:575) {
    dfree[i] ~ dbern(p[i])
    dfree.rep[i] ~ dbern(p[i])
    check[i] <- equals(dfree.rep[i],dfree[i])
    check.y1[i] <- equals(dfree.rep[i],dfree[i])*equals(dfree[i],1)
    check.y0[i] <- equals(dfree.rep[i],dfree[i])*equals(dfree[i],0)
    logit(p[i]) <- beta0 + beta[1]*age[i]/10 + beta[2]*beck[i]/10 +
      beta[3]*equals(ivhx[i],2) + beta[4]*equals(ivhx[i],3) +
      beta[5]*ndrugtx[i]/10 + beta[6]*race[i] + beta[7]*treat[i] +
      beta[8]*site[i]
  }
  # predict probs at combination of covariates
  logit(P[1]) <- beta0+beta[1]*3.7+beta[2]*1.7+beta[4]+beta[5]*0.1+beta[7]
  logit(P[2]) <- beta0+beta[1]*2.7+beta[2]*1.7+beta[4]+beta[5]*0.6
  totch.y1 <- sum(check.y1[])
  totch.y0 <- sum(check.y0[])

  # priors
  beta0 ~ dnorm(0,0.001)
  for (j in 1:8) {
    beta[j] ~ dnorm(0,0.001)
    sig.beta[j] <- step(beta[j])
  }
}" %>% strsplit('\n') %>% unlist

INI <- list(
  list(beta0=0,beta=c(0,0,0,0,0,0,0,0))
  ,list(beta0=-2,beta=c(0,0,-0.5,-0.5,0,0,0,0))
)
```

```

R <- build.model(code, df, INI, params="beta",n.chains=2,n.adapt=500)

summary(R)$statistics %>% kable %>% print
summary(R)$quantiles %>% kable %>% print

# trace plot and Kernel density
#plot(R)
#gelman.diag(R)$psrf %>% kable %>% print
#gelman.plot(R)

# extract predicted probabilities, predictive concordance rates
#R1 <- build.model(code, df, INI, params=c('check','p'),n.chains=2,n.adapt=500)
#jagsresults(R1, c("check")) %>% kable %>% print
#jagsresults(R1, c("p")) %>% kable %>% print

```

Problem 1

The correct interpretation of `totch.y1` is (b). The variable `check` is true when a replicated observation matches the actual observation. `check.y1` is true when `check` is true **and** when the actual response was equal to 1. Thus (b)

Problem 2

Sensitivity is the percentage of response = 1 given predict = 1. Therefore, the answer is (b).

Problem 3

```

code %>%
{.[19] <- " sens <- sum(check.y1[]) / sum(dfree[]);.} %>%
build.model(df, INI, params = 'sens',n.chains=2,n.adapt=500) %>%
jagsresults('sens') %>%
kable

```

	mean	sd	2.5%	25%	50%	75%	97.5%
sens	0.2994004	0.04209	0.2176871	0.2721088	0.2993197	0.3265306	0.3809524

The posterior mean sensitivity is (b) over 0.25

Problem 4

```

code %>%
{.[19] <- " cont <- P[1]/P[2]";.} %>%
build.model(df, INI, params = 'cont',n.chains=2,n.adapt=500) %>%

```

```
jagsresults('cont') %>%
kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
cont	2.999551	0.808572	1.75472	2.421248	2.888427	3.449883	4.894105

The posterior mean for this contrast is ~3.0, so (b) over 2.5.

Problem 5

```
code %>%
{.[15] <- " logit(P[1]) <- beta0+beta[1]*3.7+beta[2]*1.7+beta[5]*0.1+beta[7];." %>%
{.[19] <- " cont <- P[1] / P[2];." %>%
build.model(df, INI, params = 'cont', n.chains=2, n.adapt=500) %>%
jagsresults('cont') %>%
kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
cont	4.821774	1.501864	2.540641	3.748079	4.601841	5.65275	8.329513

The posterior mean for the ratio $P[1] / P[2]$ is ~4.8, so (a) over 4.5.

Problem 6

```
code %>%
append(" r[i] <- (dfree[i] - p[i])/(p[i]*(1-p[i]))^0.5", 5) %>%
build.model(df, INI, params = 'r', n.chains=2, n.adapt=500, n.iter=1000) %>%
jagsresults('r') %>%
as.data.frame %>%
{.[1] == max(.[1]) | .[1] == min(.[1]) %>%
#filter(mean == max(mean) | mean == min(mean)) %>%
kable(row.names=TRUE)
```

	mean	sd	2.5%	25%	50%	75%	97.5%
r[7]	4.794995	2.0774423	2.077964	3.376048	4.326474	5.712460	10.1148964
r[551]	-1.233075	0.2586898	-1.819831	-1.381965	-1.198514	-1.050433	-0.8190872

Observations #7 and #551 are the max and min standardized residuals, respectively.

Setup 2

```
library(rjags)
df <- data.frame(
```

```

n = c(1379,638,213,254)
,y = c(24, 35, 21,30)
,z = c(32, 56, 26,47)
,score = c(1,3,5,6)
)
df %<>% mutate(not.y = n-y)
df %<>% mutate(not.z = n-z)

# classical binomial regression fit
CF1=glm(cbind(y, not.y) ~ score, data = df, family=binomial)
CF2=glm(cbind(z, not.z) ~ score, data = df, family=binomial)
summary(CF1) %>% coef %>% kable
summary(CF2) %>% coef %>% kable

# JAGS
code <- "
model {
  for (i in 1:4) {
    y[i] ~ dbin(p[i],n[i])
    y.rep[i] ~ dbin(p[i],n[i])
    check[i] <- step(y.rep[i]-y[i]) + 0.5*equals(y.rep[i],y[i])
    logit(p[i]) <- beta[1]+beta[2]*score[i]
  }

  # priors
  for (j in 1:2) {beta[j] ~ dnorm(0,0.001)}
}" %>% strsplit('\n') %>% unlist

INI <- list(
  list(beta=c(0,0))
  ,list(beta=c(0,0.5))
)

```

Problem 7

```

build.model(code, df, INI, params = 'check', n.chains = 2, n.adapt = 500) %>%
  jagsresults('check') %>%
  kable

```

	mean	sd	2.5%	25%	50%	75%	97.5%
check[1]	0.77382	0.4627277	0	0	1	1	1.5
check[2]	0.15873	0.3970427	0	0	0	0	1.5
check[3]	0.46645	0.5549735	0	0	0	1	1.5
check[4]	0.74032	0.4819761	0	0	1	1	1.5

```

# M <- jags.model(inits=INI,data=D,n.chains=2,n.adapt=500, file="binom.jag")
# R <- coda.samples(M,c("beta"),n.iter=5000)
# summary(R)
# gelman.diag(R)

```

Observation # 2 actually causes the most concern, and the model tends to underpredict the true result.

Problem 8

```
code %>%
  append('m[i] <- n[i]*p[i]', 7) %>%
  append('dv[i] <- 2*(y[i]*log(y[i]/m[i])+(n[i]-y[i])*log((n[i]-y[i])/(n[i]-m[i])))', 8) %>%
  append('tot.dv <- 2*sum(dv[])', 13) %>%
  build.model(df, INI, params = c('tot.dv', 'dv'), n.chains = 2, n.adapt = 500) %>%
  jagsresults(c('tot.dv', 'dv')) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
dv[1]	1.3106110	1.6626916	0.0014609	0.1605540	0.6814964	1.8479317	5.947259
dv[2]	2.0060090	1.5292552	0.0544226	0.8492890	1.6855323	2.8197168	5.751911
dv[3]	0.3444194	0.4696934	0.0003856	0.0364876	0.1626717	0.4663643	1.671483
dv[4]	1.1271089	1.4585401	0.0012283	0.1339458	0.5744035	1.5721517	5.245793
tot.dv	9.5762967	3.9248579	5.7175066	6.7705612	8.3681501	11.1283900	20.083287

The mean posterior deviance is ~ 9.6 , so (a) over 5.

Problem 9

```
INI <- list(
  list(beta=c(0,0), gam=c(0,0))
, list(beta=c(0,0.5), gam=c(0,0.5))
)
```

```
code %>%
  append('z[i] ~ dbin(p2[i], n[i])', 4) %>%
  append('logit(p2[i]) <- gam[1] + gam[2]*score[i]', 8) %>%
  append('for(j in 1:2) {gam[j] ~ dnorm(0, 0.001)}', 13) %>%
  append('r <- step(gam[2] - beta[2])', 14) %>%
  build.model(df, INI, params = c('beta', 'gam', 'r'), n.chains = 2, n.adapt = 500) %>%
  jagsresults(c('beta', 'gam', 'r')) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	-4.2802170	0.2090432	-4.7014570	-4.4177152	-4.2746111	-4.1373586	-3.8836357
beta[2]	0.3994441	0.0499377	0.3023527	0.3656101	0.3994229	0.4326269	0.4976207
gam[1]	-3.9426602	0.1770150	-4.2984736	-4.0594516	-3.9415071	-3.8217897	-3.6008941
gam[2]	0.4224258	0.0426302	0.3383956	0.3939409	0.4225720	0.4506137	0.5060889
r	0.6389200	0.4803185	0.0000000	0.0000000	1.0000000	1.0000000	1.0000000

The probability is that the trend slope for hypertension is greater than CHD is $\sim 63\%$, so (b) under 0.75.

Setup 3

```
library(MASS)

# columns headed ofp, hosp, health,numchron,gendermale,school,privins
df <- read.table("../data/debtriv.txt",header=T)
df$health.gr <- factor(df$health)

#Different GLM models - Poisson and Neg. Bin.
CM1 <- glm(ofp ~ hosp+health.gr+numchron+gendermale+school+privins, data = df, family = poisson)
CM2 <- glm.nb(ofp ~ hosp+health.gr+numchron+gendermale+school+privins, data = df)

# JAGS Poisson
poisson <- "
model {
  for (i in 1:4406) {
    ofp[i] ~ dpois(mu[i])
    log(mu[i]) <- beta0 + beta[1]*hosp[i] + beta[2]*equals(health[i],2) +
      beta[3]*equals(health[i],3) + beta[4]*numchron[i] + beta[5]*gendermale[i] +
      beta[6]*school[i] + beta[7]*privins[i]
  }
  beta0 ~ dnorm(0,0.000001)
  for (i in 1:7){ beta[i] ~ dnorm(0,0.001)}
}" %>% strsplit('\n') %>% unlist

p.inits <- list(
  list(beta0=0,beta=c(0,0,0,0,0,0,0))
  ,list(beta0=1,beta=c(0.2,0.2,-0.2,0,0,0,0.2))
)
build.model(poisson, df, p.inits, params = 'beta', n.chains = 2, n.adapt = 500, n.iter=1000) %>%
  jagsresults('beta') %>%
  kable

# JAGS Neg-Bin
negbin <- "
model {
  for (i in 1:4406) {
    ofp[i] ~ dnegbin(p[i],theta)
    p[i] <- theta/(theta+mu[i])
    log(mu[i]) <- beta0 + beta[1]*hosp[i] + beta[2]*equals(health[i],2) +
      beta[3]*equals(health[i],3) + beta[4]*numchron[i] + beta[5]*gendermale[i] +
      beta[6]*school[i] + beta[7]*privins[i]
  }
  theta ~ dgamma(1,0.01)
  beta0 ~ dnorm(0,0.000001)
  for (i in 1:7){ beta[i] ~ dnorm(0,0.001)}
}" %>% strsplit('\n') %>% unlist

n.inits <- list(
  list(beta0=0,beta=c(0,0,0,0,0,0,0),theta=1)
  ,list(beta0=1,beta=c(0.2,0.2,-0.2,0,0,0,0.2),theta=2)
)

build.model(negbin, df, n.inits, params=c('beta','theta'), n.chains=2, n.adapt = 500, n.iter=1000) %>%
```

```
jagsresults(c('beta','theta')) %>%
kable
```

Problem 10

```
poisson %>%
  append('dev[i] <- 2*(ofp[i]*log(ofp[i]/mu[i]) - (ofp[i] - mu[i]))', 7) %>%
  append('tot.dev <- sum(dev[])', 9) %>%
  build.model(df, p.inits, params = 'tot.dev', n.chains = 2, n.adapt = 500, n.iter=1000) %>%
  jagsresults('tot.dev') %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
tot.dev	23175.99	4.100786	23169.96	23173.12	23175.22	23178.17	23185.94

The posterior mean for the total Poisson deviance is ~23000, so (a) under 24000

Problem 11

```
negbin %>%
  append('V[i] <- mu[i] + mu[i]^2/theta', 8) %>%
  append('r[i] <- (ofp[i] - mu[i])/V[i]^0.5', 9) %>%
  build.model(df, n.inits, params='r', n.chains=2, n.adapt = 500, n.iter=500) %>%
  jagsresults('r') %>%
  as.data.frame %>%
  arrange(desc(mean)) %>%
  head(2) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
	18.68181	1.0452437	16.81133	17.92455	18.69336	19.41725	20.66903
	12.87955	0.4853215	11.89407	12.53389	12.91930	13.24344	13.72860

The two largest residuals are shown above?

Problem 12

```
negbin %>%
  {.[10] <- 'theta ~ dunif(0, 10)'; .} %>%
  build.model(df, n.inits, params='theta', n.chains=2, n.adapt = 500, n.iter=5000) %>%
  jagsresults('theta') %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
theta	1.204443	0.0336081	1.13888	1.182041	1.20407	1.226391	1.271962

Posterior mean of theta is ~ 1.21 , so (b) under 1.25