Homework 2

Christian Tillich

September 30, 2017

Problem Setup

Linear Reference

```
# columns prestige, educ,log2inc,women
df.prestige <- read.table("../data/prestige.txt",header=T)
# conventional least squares
LM <- lm(prestige ~ education + log2inc +women, data=df.prestige)
LM %>% summary %>% coef %>% kable
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-110.9658241	14.8429281	-7.476006	0.0000000
education	3.7305078	0.3543830	10.526767	0.0000000
log2inc	9.3146664	1.3265151	7.021908	0.0000000
women	0.0468951	0.0298989	1.568459	0.1199974

Bayesian Comparison

```
# JAGS
code <- "
model {
  for (i in 1:102) {
    prestige[i] ~ dnorm(mu[i],tau)
    e[i] <- prestige[i]-mu[i]
    mu[i] <- beta[1]+beta[2]*education[i]+beta[3]*log2inc[i]+beta[4]*women[i]
  }
  for (j in 1:4) {beta[j] ~ dnorm(0,0.001)}
  tau ~ dgamma(1,0.001)
  sig <- 1/sqrt(tau)
}" %>% strsplit('\n') %>% unlist


build.model <- function(
   model
  ,data = df.prestige
  ,inits = list(
      list(beta=c(0,0,0,0),tau=1)
     ,list(beta=c(-10,0,0,0),tau=0.1)
   )
  ,params = c("beta","sig")
  ,...
){
```

```r
  M <- jags.model(textConnection(paste(model, collapse="\n")), data=data,inits=inits, quiet=T,...)
  R <- coda.samples(M,params,n.iter=25000)
  R
}
```

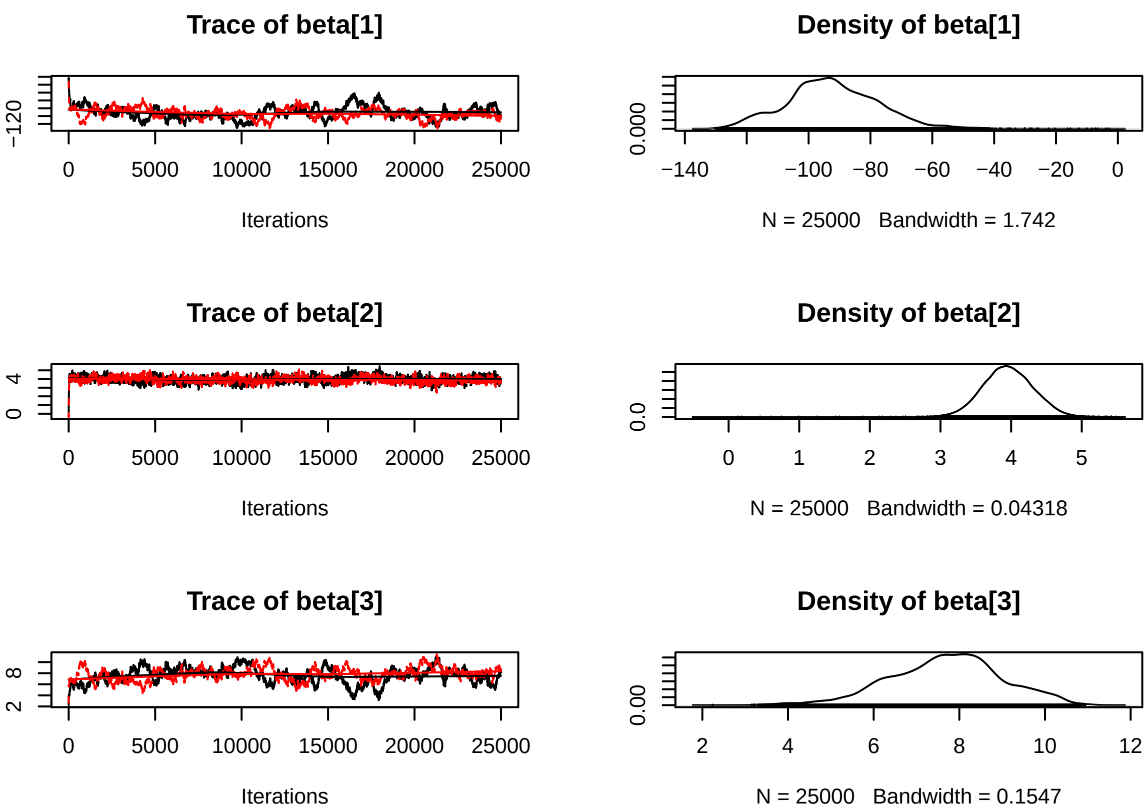
```r
mdl <- build.model(code, n.chains=2, n.adapt=500)

summary(mdl)$statistics %>% kable
```

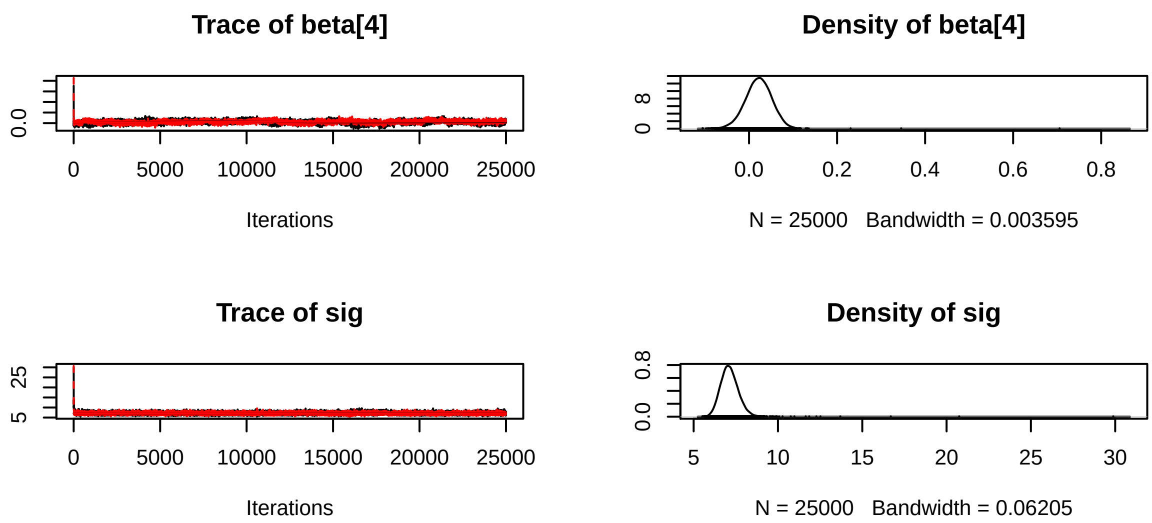
	Mean	SD	Naive SE	Time-series SE
beta[1]	-92.3636727	14.6871143	0.0656828	1.7958951
beta[2]	3.9547628	0.3591323	0.0016061	0.0134872
beta[3]	7.6946599	1.3045549	0.0058341	0.1844911
beta[4]	0.0217785	0.0301770	0.0001350	0.0018430
sig	7.1306606	0.5542947	0.0024789	0.0063944

```r
summary(mdl)$quantiles %>% kable
```

	2.5%	25%	50%	75%	97.5%
beta[1]	-120.1549729	-101.9477226	-93.0768837	-82.7764986	-62.3688427
beta[2]	3.2711505	3.7179153	3.9501569	4.1930782	4.6499575
beta[3]	4.9765929	6.8412300	7.7533876	8.5434045	10.1796458
beta[4]	-0.0387559	0.0023465	0.0222658	0.0419143	0.0782828
sig	6.2033359	6.7711275	7.0984541	7.4540050	8.2287050

```r
plot(mdl)
```

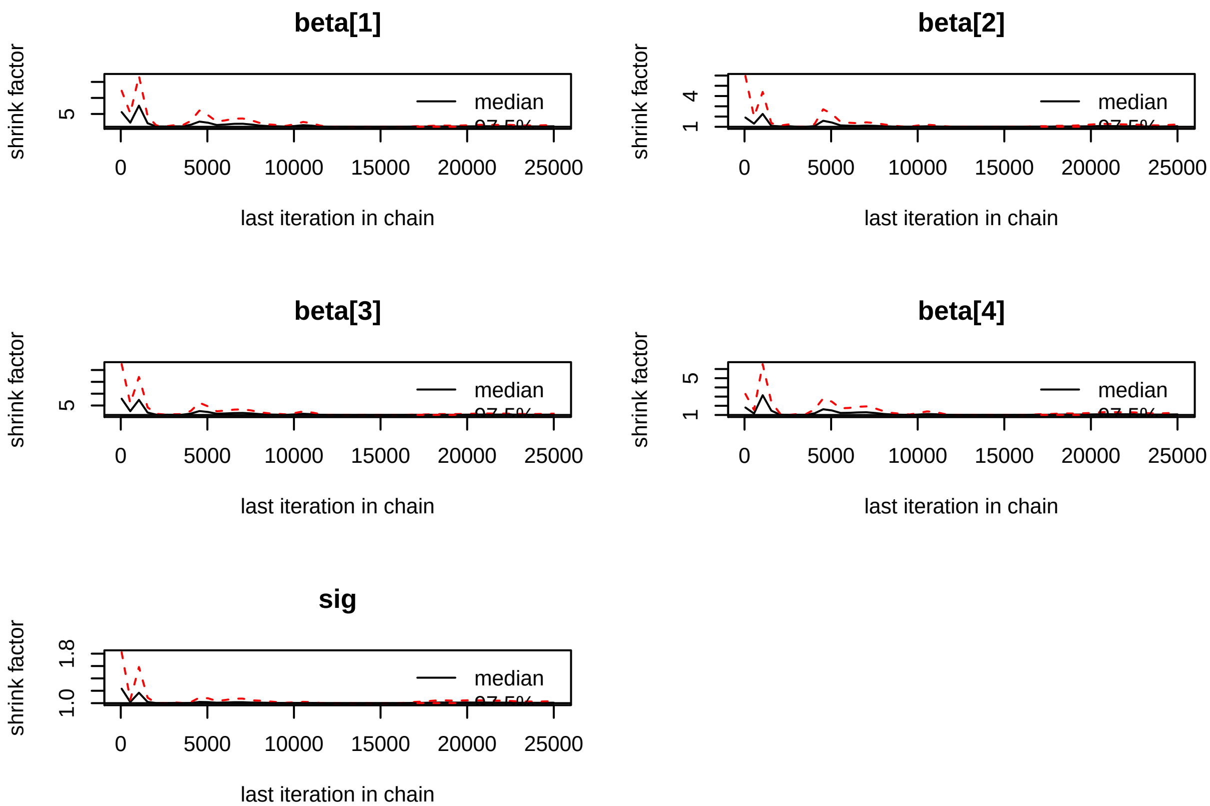

Trace of beta[1]
−120
0
5000
10000
15000
20000
25000
Iterations
Density of beta[1]
0.000
−140
−100
−80
−60
−40
−20
0
N = 25000 Bandwidth = 1.742
Trace of beta[2]
0
4
0
5000
10000
15000
20000
25000
Iterations
Density of beta[2]
0.0
0
1
2
3
4
5
N = 25000 Bandwidth = 0.04318
Trace of beta[3]
2
8
0
5000
10000
15000
20000
25000
Iterations
Density of beta[3]
0.00
2
4
6
8
10
12
N = 25000 Bandwidth = 0.1547




```r
gelman.diag(mdl)$psrf %>% as.data.frame %>% kable
```

	Point est.	Upper C.I.
beta[1]	1.157081	1.555669
beta[2]	1.056324	1.229736
beta[3]	1.159442	1.563336
beta[4]	1.063375	1.243130
sig	1.006695	1.030474

```r
gelman.plot(mdl)
```



Question 1

```r
code %>%
  {.[5] <- "    r[i] <- (prestige[i]-mu[i])/sig"; .} %>%
  build.model(params = c("r"), n.chains = 2, n.adapt = 500) %>%
  jagsresults(params="r") %>%
  as.data.frame %>%
  filter(mean == max(mean) | mean == min(mean)) %>%
  kable
```

mean	sd	2.5%	25%	50%	75%	97.5%
-2.468751	0.2110676	-2.884143	-2.610301	-2.467618	-2.326340	-2.062505
2.567313	0.2849938	2.018158	2.381020	2.566868	2.754981	3.120135

Above I show the residual extreme values.

Question 2

Yes, the 95% confidence interval is bounded at (2.0, 3.1)

Question 3

```
code %>%
  append("  for (j in 1:4) {sig.beta[j] <- step(beta[j])}", 8) %>%
  build.model(params = c("beta", "sig.beta"), n.chains = 2, n.adapt = 500) %>%
  summary %>%
  .$statistics %>%
  kable
```

	Mean	SD	Naive SE	Time-series SE
beta[1]	-89.7915485	15.6792825	0.0701199	1.9904462
beta[2]	3.9991444	0.3784321	0.0016924	0.0184522
beta[3]	7.4594273	1.4038911	0.0062784	0.2154234
beta[4]	0.0182068	0.0314924	0.0001408	0.0021397
sig.beta[1]	0.0000000	0.0000000	0.0000000	0.0000000
sig.beta[2]	1.0000000	0.0000000	0.0000000	0.0000000
sig.beta[3]	1.0000000	0.0000000	0.0000000	0.0000000
sig.beta[4]	0.7272800	0.4453625	0.0019917	0.0214963

The intercept, education, and income measures all prove significant. The percentage of incombants who are women is not.

Question 4

```
code <- "
model {
  for (i in 1:102) {
    prestige[i] ~ dnorm(mu[i],tau[i])
    tau[i] <- tau.1*xi[i]
    xi[i] ~ dgamma(nu.1,nu.1)
    e[i] <- (prestige[i] - mu[i])*sqrt(xi[i])/sig
    mu[i] <- beta[1]+beta[2]*education[i]+beta[3]*log2inc[i]+beta[4]*women[i]
}
for (j in 1:4) {beta[j] ~ dnorm(0,0.001)}
tau.1 ~ dgamma(1,0.001)
nu ~ dexp(0.1)
nu.1 <- nu/2
sig <- 1/sqrt(tau.1)}
" %>% strsplit('\n') %>% unlist

inits = list(
      list(beta=c(0,0,0,0),tau.1=1, nu=10)
     ,list(beta=c(-10,0,0,0),tau.1=0.1, nu=1)
   )

code %>%
  build.model(params=c("beta","tau.1","nu"),inits=inits,n.chains=2,n.adapt = 500) %>%
  summary %>% .$statistics %>% kable
```

	Mean	SD	Naive SE	Time-series SE
beta[1]	-89.7690801	14.9236060	0.0667404	2.0910342
beta[2]	4.0381429	0.3623964	0.0016207	0.0186753
beta[3]	7.4323662	1.3422098	0.0060025	0.2103203
beta[4]	0.0151022	0.0304260	0.0001361	0.0020240
nu	16.7571214	10.5202284	0.0470479	0.3489903
tau.1	0.0231063	0.0042789	0.0000191	0.0000781

The posterior mean is ~17. So above.

Question 5

```
code %>%
  build.model(params=c("sig"),inits=inits,n.chains=2,n.adapt = 500) %>%
  summary %>% .$statistics %>%
  kable
```

Mean	6.6601576
SD	0.5830284
Naive SE	0.0026074
Time-series SE	0.0090623

The mean of sigma here is ~6.7. The mean of the fully-shared sigma from 2.4.3 is ~7.1. So it is lower than in the Normal model.

Question 6

```
df.salin <- read.table("../data/salin.txt",header=T)

out <- code %>%
  {.[8] <- "    mu[i] <- beta[1]+beta[2]*lag[i]+beta[3]*trend[i]+beta[4]*dis[i]"; .} %>%
  gsub("prestige","sal",.) %>% gsub("102","28", .) %>%
  build.model(
     data=df.salin
    ,inits=inits
    ,n.chains=2
    ,n.adapt=500
    ,params = c("beta","tau.1","nu")
  ) %>%
  summary %>% .$statistics

kable(out)
```

	Mean	SD	Naive SE	Time-series SE
beta[1]	15.5530659	4.9640692	0.0222000	0.8037097
beta[2]	0.7392892	0.0727257	0.0003252	0.0024575
beta[3]	-0.1445093	0.1625435	0.0007269	0.0084840

	Mean	SD	Naive SE	Time-series SE
beta[4]	-0.5200069	0.1848767	0.0008268	0.0285245
nu	5.9105035	7.4751882	0.0334301	0.2918960
tau.1	1.9418745	2.0404975	0.0091254	0.0523915

The posterior mean for nu is 5.91, so below 10.

Question 7

```
code %>%
  {.[8] <- "    mu[i] <- beta[1]+beta[2]*lag[i]+beta[3]*trend[i]+beta[4]*dis[i]"; .} %>%
  gsub("prestige","sal",.) %>% gsub("102","28", .) %>%
  build.model(
     data=df.salin
    ,inits=list(beta=c(0,0,0,0),tau.1=1, nu=10)
    ,n.chains=2
    ,n.adapt=500
    ,params = c("xi")
  ) %>%
  jagsresults("xi") %>%
  as.data.frame %>%
  {.[.$mean == min(.$mean), ]} %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
xi[16]	0.2871533	0.3571829	0.0022268	0.0340667	0.127595	0.4336481	1.216192

Observation 16

Question 8

```
df.credit <- read.table("../data/creditcard.txt",header=T)

code <- "
model {
  for (i in 1:72) {
    # Core Model
    exp[i] ~ dnorm(mu[i],1/sig2[i])
    sig2[i] <- exp(h[i])
    e[i] <- (exp[i]-mu[i])/sqrt(sig2[i])

    # Centered Variables
    age.c[i] <- age[i] - mean(age[])
    income.c[i] <- income[i] - mean(income[])
    ownrent.c[i] <- ownrent[i] - mean(ownrent[])
    incomesq.c[i] <- incomesq[i] - mean(incomesq[])
```

```
    # Explanatory Parameters
    mu[i] <- beta[1] + beta[2]*age.c[i] + beta[3]*income.c[i] + beta[4]*ownrent.c[i] + beta[5]*incomesq
    h[i] <- gam[1] + gam[2]*age.c[i] + gam[3]*income.c[i] + gam[4]*ownrent.c[i] + gam[5]*incomesq.c[i]
  }
  for (j in 1:5) {
    beta[j] ~ dnorm(0,0.001)
    gam[j] ~ dnorm(0, 0.001)
  }
}" %>% strsplit('\n') %>% unlist

inits <- list(
  list(beta=c(0,0,0,0,0),gam=c(0,0,0,0,0)),
  list(beta=c(-10,0,0,0,0),gam=c(-1,0,0,0,0))
)
build.model(
   code
  ,data=df.credit
  ,inits = inits
  ,params = c('beta','gam','e')
  ,n.chains=2
  ,n.adapt=500
) %>%
  jagsresults(c('beta','gam')) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	130.9635688	18.3179414	93.8650385	119.3034637	131.3556061	143.4361769	165.5692645
beta[2]	-1.6589230	1.7084952	-4.3347293	-2.6322213	-2.0440454	-1.0385712	2.7673204
beta[3]	-1.3590189	22.6555829	-44.7769444	-18.6801012	-1.0570537	14.3261323	40.0376400
beta[4]	14.0311694	28.2829179	-40.2047437	-4.4725712	13.7366301	32.3487342	70.9779449
beta[5]	4.9528486	2.1492626	1.1842883	3.6568007	5.0221886	6.5115200	8.6039100
gam[1]	10.5153681	0.2038408	10.1349706	10.3751995	10.5087524	10.6489189	10.9357713
gam[2]	0.0074414	0.0318152	-0.0542012	-0.0139824	0.0073012	0.0287475	0.0701698
gam[3]	4.1390119	0.6680804	2.7486282	3.7132159	4.1487093	4.5707658	5.4379923
gam[4]	0.2912807	0.3957306	-0.4638624	0.0219227	0.2855998	0.5545393	1.0903301
gam[5]	-0.4089958	0.0849328	-0.5729958	-0.4646932	-0.4095410	-0.3549127	-0.2285479

Income and Squared-Income both seem to have a significant effect on heteroscedasticity. Rent and age do not.