

# STAT641 - Homework 10

Marcel Gietzmann-Sanders

## Problem 1

(a)

```
mtcars$cwt = mtcars$wt - mean(mtcars$wt)
mtcars$cqsec = mtcars$qsec - mean(mtcars$qsec)

model1 = lm(mpg ~ cwt + cqsec, data=mtcars)
model2 = lm(mpg ~ cwt + cqsec + cwt * cqsec, data=mtcars)

(summary(model1))

##
## Call:
## lm(formula = mpg ~ cwt + cqsec, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.3962 -2.1431 -0.2129  1.4915  5.7486
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.0906     0.4589  43.776 < 2e-16 ***
## cwt         -5.0480     0.4840 -10.430 2.52e-11 ***
## cqsec        0.9292     0.2650   3.506  0.0015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.596 on 29 degrees of freedom
## Multiple R-squared:  0.8264, Adjusted R-squared:  0.8144
## F-statistic: 69.03 on 2 and 29 DF,  p-value: 9.395e-12
```

```
(summary(model2))
```

```
##
## Call:
## lm(formula = mpg ~ cwt + cqsec + cwt * cqsec, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.0143 -1.9740  0.0877  1.3045  5.3115
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   19.9161     0.4818  41.340 < 2e-16 ***
## cwt           -5.0274     0.4819 -10.432 3.75e-11 ***
## cqsec          0.8595     0.2707   3.175 0.00363 **
## cwt:cqsec     -0.5897     0.5188  -1.137 0.26526
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.583 on 28 degrees of freedom
## Multiple R-squared:  0.8341, Adjusted R-squared:  0.8163
## F-statistic: 46.92 on 3 and 28 DF,  p-value: 4.777e-11
```

The  $p$  value for the interaction term is quite high at 26.5% and adding the term did little to reduce our residual standard error (as compared to model 1) so I'd say that we don't really seem to need this interaction term.

(b)

```
AIC(model1)
AIC(model2)
```

Model	AIC
Model 1	156.7
Model 2	157.3

These AIC are extremely close. However if ignore their closeness then model 1 is ever so slightly better with the smaller AIC of the two.

## Problem 2

(a)

We will want to report on the credible sets for each of our parameters and there is a *beta* for the interaction term as well. Therefore we must include it in model 2.

```
model
{
  for (i in 1:N) {
    y[i] ~ dnorm(mu[i], 1 / sigma ^ 2)
    mu[i] = (
      beta[1]
      + beta[2] * cwt[i] + beta[3] * cqsec [i]
      + beta[4] * cwt[i] * cqsec[i]
    )
  }

  beta[1:4] ~ dmnorm(
    beta.prior.mean,
    beta.prior.precision
  )
  sigma ~ dunif(0, 20)
}
```

	Mean	SD	95% Credible Set	MCMC Error
$\beta_1$	4.62	5.1	-5.49, 14.72	0.029
$\beta_2$	5.49	9.8	-13.83, 24.9	0.056
$\beta_3$	0.86	0.29	0.289, 1.42	0.002
$\beta_4$	-0.59	0.55	-1.68, 0.49	0.003
$\sigma$	2.7	0.38	2.08, 3.58	0.003

```
library(rjags)
beta.prior.mean = rep(0, 4)
beta.prior.precision = solve(10^5 * diag(4))
data = list(
  y=mtcars$mpg,
  N=nrow(mtcars),
  cwt=mtcars$cwt,
  cqsec=mtcars$cqsec,
  beta.prior.mean=beta.prior.mean,
  beta.prior.precision=beta.prior.precision
)
inits = list(
  list(beta=rep(0, 4), sigma=1),
  list(beta=rep(1, 4), sigma=2),
  list(beta=rep(-1, 4), sigma=2)
)
fname = "hw10_model_2.txt"
n.iter = 10000
thin = 1

model2 = jags.model(
  file=fname,
```

```

data=data,
inits=inits,
n.chains=3,
n.adapt=1000,
quiet=TRUE
)
variables = c("beta", "sigma")
samples = coda.samples(
  model2, variables, n.iter=n.iter, thin=thin
)
summary(samples)

```

```

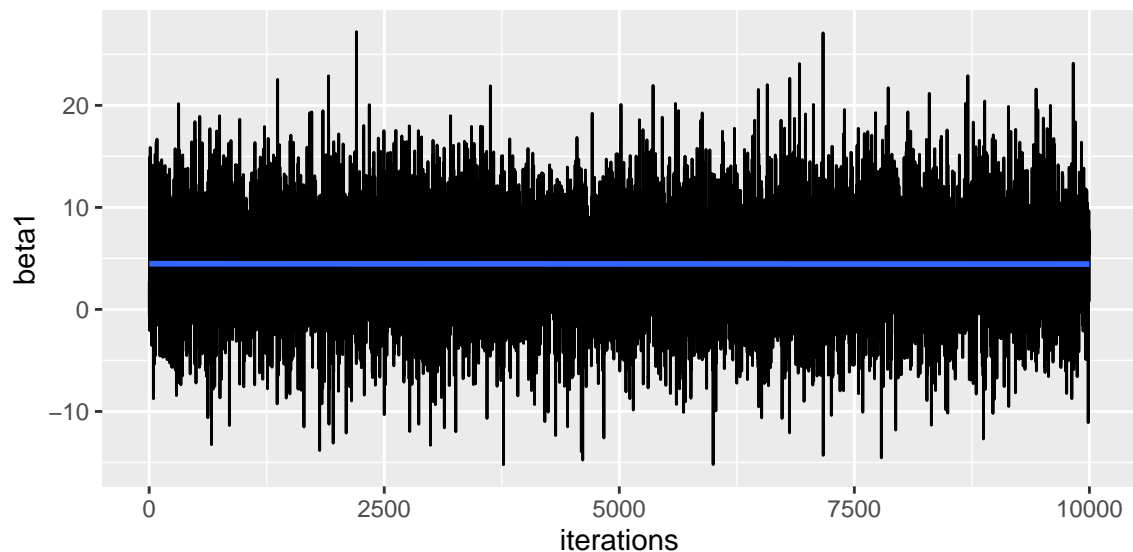
library(ggplot2)
df = data.frame(
  beta1=as.numeric(samples[[1]][,"beta[1]"]),
  beta2=as.numeric(samples[[1]][,"beta[2]"]),
  beta3=as.numeric(samples[[1]][,"beta[3]"]),
  beta4=as.numeric(samples[[1]][,"beta[4]"]),
  sigma=as.numeric(samples[[1]][,"sigma"]),
  iterations=seq(1,n.iter)
)

```

```

(
  ggplot(df, aes(x=iterations,y=beta1))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta1")
)

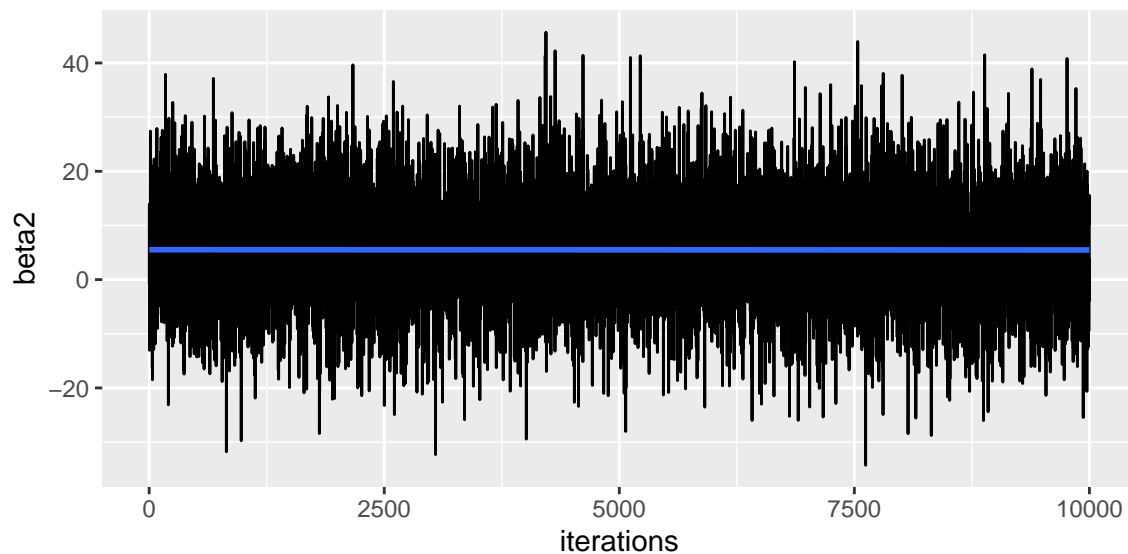
```



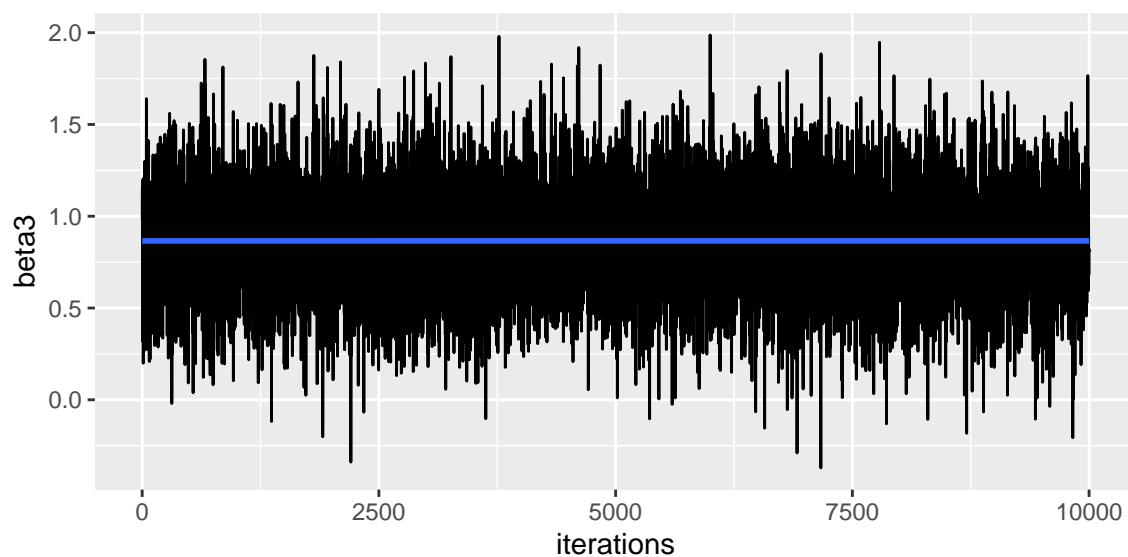
```

(
  ggplot(df, aes(x=iterations,y=beta2))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta2")
)

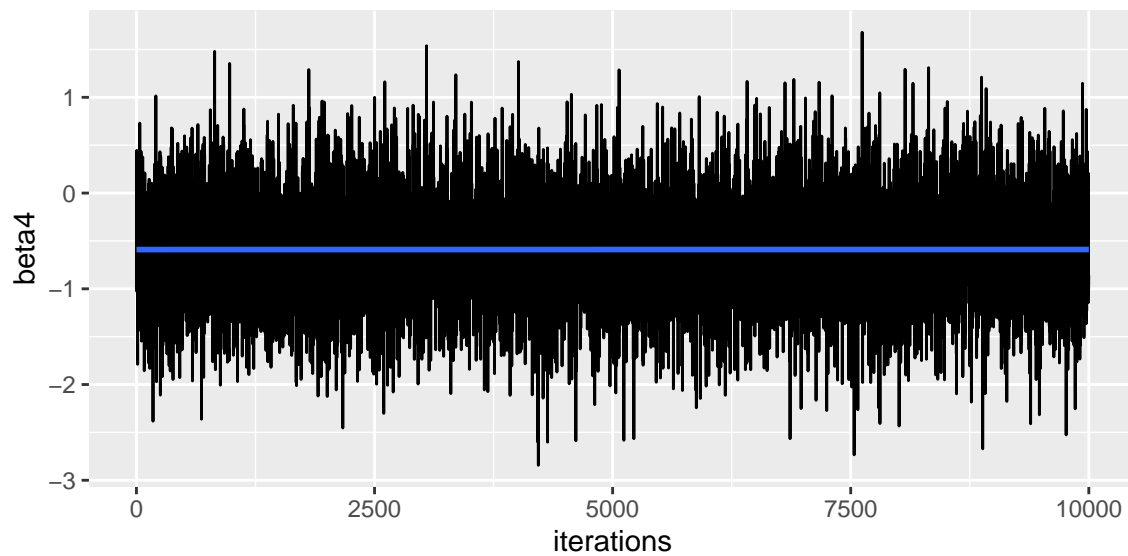
```



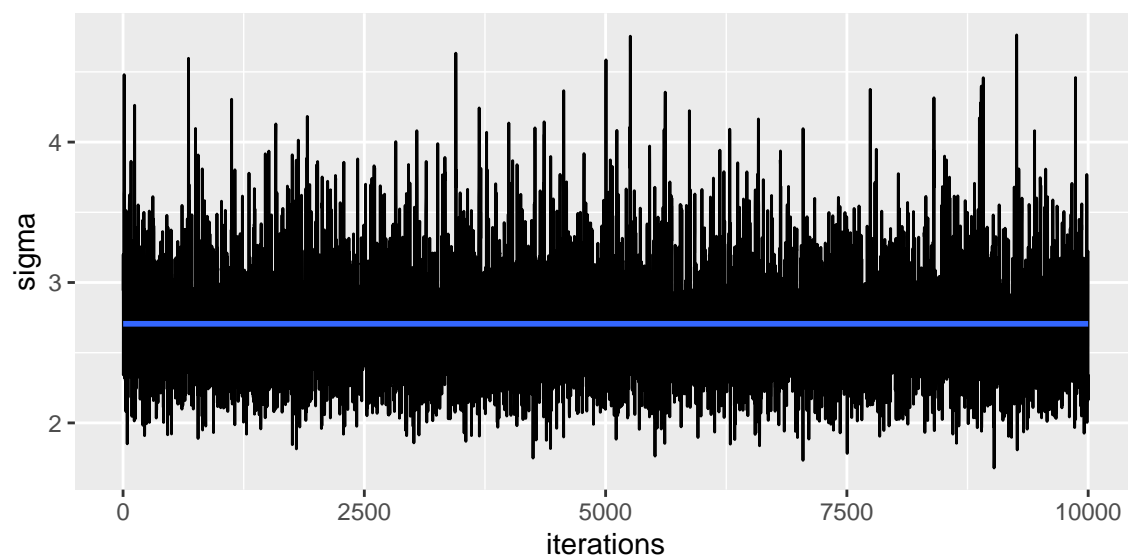
```
(
  ggplot(df, aes(x=iterations,y=beta3))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta3")
)
```



```
(
  ggplot(df, aes(x=iterations,y=beta4))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta4")
)
```



```
(  
  ggplot(df, aes(x=iterations,y=sigma))  
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")  
  + ylab("sigma")  
)
```



(b)

Model 1:

```
model
{
  for (i in 1:N) {
    y[i] ~ dnorm(mu[i], 1 / sigma ^ 2)
    mu[i] = (
      beta[1]
      + beta[2] * cwt[i] + beta[3] * cqsec [i]
    )
  }

  beta[1:3] ~ dmnorm(
    beta.prior.mean,
    beta.prior.precision
  )
  sigma ~ dunif(0, 20)
}
```

	Mean	SD	95% Credible Set	MCMC Error
$\beta_1$	3.56	5.0	-6.32, 13.4	0.029
$\beta_2$	-5.05	0.51	-6.04, -4.05	0.003
$\beta_3$	0.93	0.28	0.38, 1.48	0.002
$\sigma$	2.71	0.37	2.09, 3.56	0.003

```
beta.prior.mean = rep(0, 3)
beta.prior.precision = solve(10^5 * diag(3))
data = list(
  y=mtcars$mpg,
  N=nrow(mtcars),
  cwt=mtcars$cwt,
  cqsec=mtcars$cqsec,
  beta.prior.mean=beta.prior.mean,
  beta.prior.precision=beta.prior.precision
)
inits = list(
  list(beta=rep(0, 3), sigma=1),
  list(beta=rep(1, 3), sigma=2),
  list(beta=rep(-1, 3), sigma=2)
)
fname = "hw10_model_1.txt"

model1 = jags.model(
  file=fname,
  data=data,
  inits=inits,
  n.chains=3,
  n.adapt=1000,
  quiet=TRUE
)
variables = c("beta", "sigma")
samples = coda.samples(
```

```

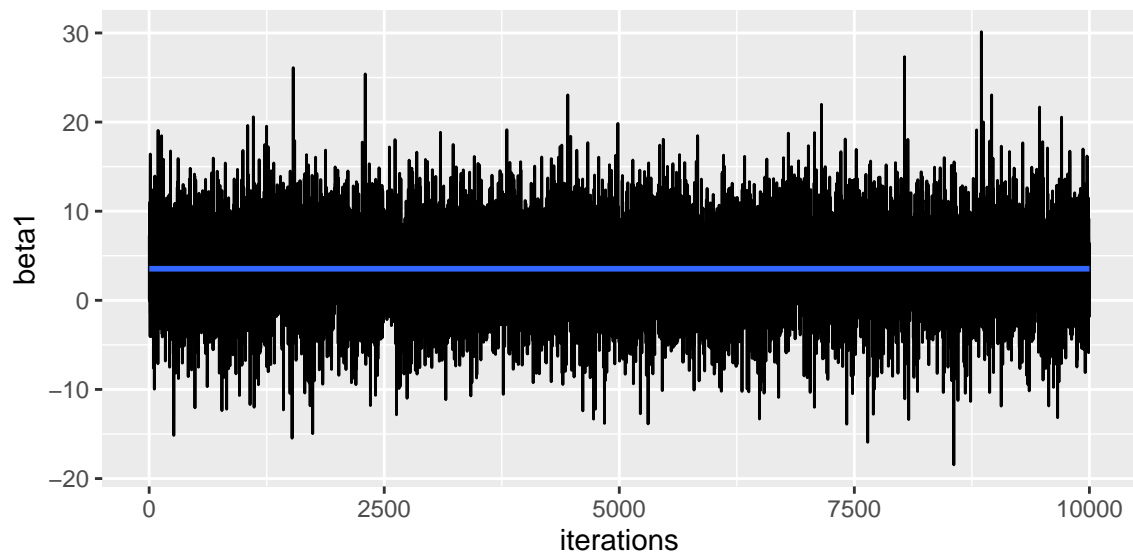
    model1, variables, n.iter=n.iter, thin=thin
)

summary(samples)

df = data.frame(
  beta1=as.numeric(samples[[1]][, "beta[1]"]),
  beta2=as.numeric(samples[[1]][, "beta[2]"]),
  beta3=as.numeric(samples[[1]][, "beta[3]"]),
  sigma=as.numeric(samples[[1]][, "sigma"]),
  iterations=seq(1,n.iter)
)

(
  ggplot(df, aes(x=iterations,y=beta1))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta1")
)

```

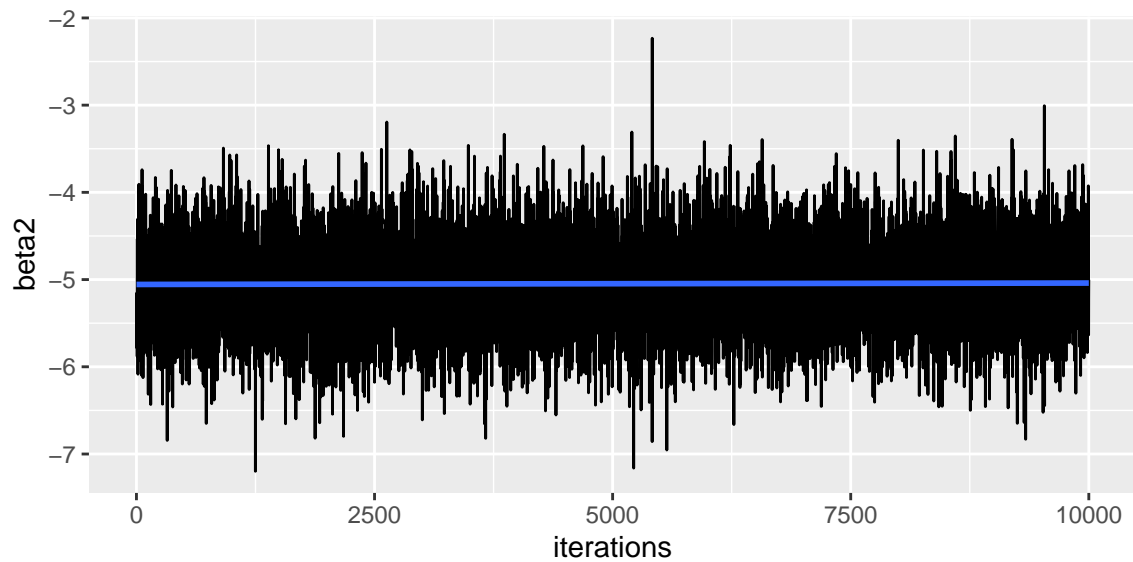


```

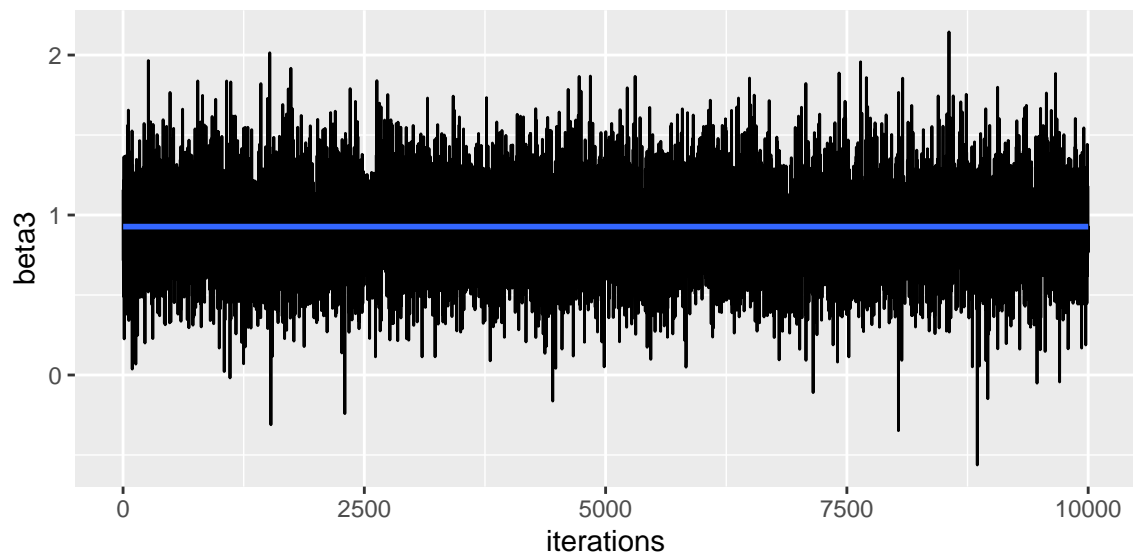
(
  ggplot(df, aes(x=iterations,y=beta2))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta2")
)

```

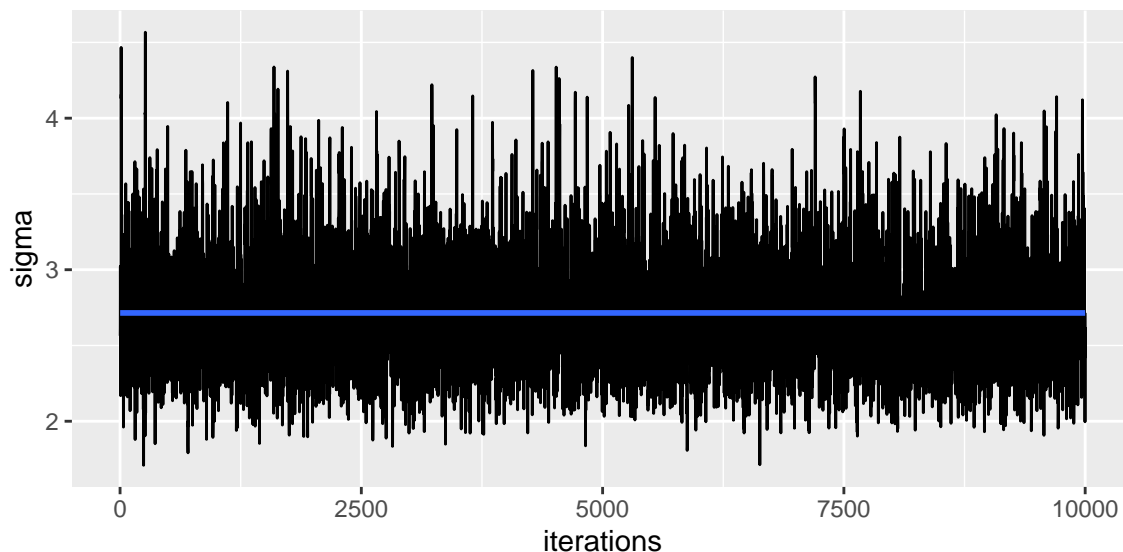




```
(
  ggplot(df, aes(x=iterations,y=beta3))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("beta3")
)
```



```
(
  ggplot(df, aes(x=iterations,y=sigma))
  + geom_line() + geom_smooth(se=FALSE) + xlab("iterations")
  + ylab("sigma")
)
```



```
for (i in 1:3) {
  (dic.samples(model1, n.iter=10000, thin=1, type="pD"))
  (dic.samples(model2, n.iter=10000, thin=1, type="pD"))
}
```

	$p_D$	DIC
Model 1	4.307, 4.331, 4.384	157.4, 157.5, 157.6
Model 2	5.438, 5.443, 5.417	158.3, 158.3, 158.3

There are very slight difference amongst the DIC for model 1 and amongst  $p_D$  for both models, but once again these differences are very slight.

In general the DIC vary by less than 1 so we're not hitting our threshold of 2 for considering these models to have any noticeable difference by DIC.

I'd end up preferring the simpler model just because the added complexity in model 2 doesn't seem to be getting us anywhere.

### Problem 3

If we apply the threshold of 2 to our AIC comparison as well then both of these methods lead to the same conclusion - these information criteria are not strongly differentiated enough to discern between the two models. Therefore if we take the "simplicity" is the best approach model 1 is chosen as preferable in both cases.

If we just take AIC at face value however then it does indicate a preference to model 1 on AIC alone - a conclusion we were not able to make with DIC.

#### Problem 4

Given we're dealing with a conjugate prior we might as well take advantage of it.

For some data:

$$y_i \sim \text{Poisson}(\lambda)$$

and a prior:

$$\lambda \sim \text{Gamma}(a, b)$$

our posterior is:

$$p(\lambda|\vec{y}) = \text{Gamma}(\sum y_i + a, n + b)$$

(a)

In this case our prior is:

$$\text{Gamma}(1, 1)$$

and our posterior is:

$$\text{Gamma}(36, 16)$$

```
(posterior_odds = (  
  (1 - pgamma(2.7, 36, rate=16))  
  / pgamma(2.7, 36, rate=16)  
))
```

```
## [1] 0.1343257
```

```
(prior_odds = (  
  (1 - pgamma(2.7, 1, rate=1))  
  / pgamma(2.7, 1, rate=1)  
))
```

```
## [1] 0.0720475
```

```
(bayes_factor = posterior_odds / prior_odds)
```

```
## [1] 1.864405
```

Given we are below 3 and above 1/3 this doesn't favor either M1 or M2.

(b)

In this case our prior is:

$$\text{Gamma}(1, 0.5)$$

and our posterior is:

$$\text{Gamma}(36, 15.5)$$

```
(posterior_odds = (  
  (1 - pgamma(2.7, 36, rate=15.5))  
  / pgamma(2.7, 36, rate=15.5)  
))
```

```
## [1] 0.1949666
```

```
(prior_odds = (  
  (1 - pgamma(2.7, 1, rate=1))  
  / pgamma(2.7, 1, rate=1)  
))
```

```
## [1] 0.0720475
```

```
(bayes_factor = posterior_odds / prior_odds)
```

```
## [1] 2.706084
```

Given we are below 3 and above 1/3 this also doesn't favor either M1 or M2.

(c)

We did get different bayes factors but not by enough to go from having no support in either direction to some support indicated for either M1 or M2.