

HW 5

Nathan Hawkins

2/26/2021

```
crack <- read.table("C:/Users/nateh/Downloads/cracker.dat", header = TRUE)
```

1 Different slopes and intercepts for each treatment

```
mdl <-  
  "model {  
  
    for(i in 1:14){  
      sales[i] ~ dnorm(mu[i], 1/vv);  
      mu[i] <- b0[tmt[i]] + b1[tmt[i]]*presales[i];  
    }  
  
    for(i in 1:3){  
      b0[i] ~ dnorm(25, .001);  
      b1[i] ~ dnorm(0, .001);  
    }  
  
    vv ~ dgamma(1.5, .5);  
  
  }  
  
"  
  
writeLines(mdl, 'ancova.txt')  
sales <- crack$sales  
presales <- crack$presales  
tmt <- crack$tmt  
  
data.jags <- c('sales', 'presales', 'tmt')  
parms <- c('b0', 'b1', 'vv')  
  
ancova.sim <- jags(data= data.jags, parameters.to.save = parms,  
                  model.file = 'ancova.txt', inits = NULL,  
                  n.iter = 50000, n.thin = 2, n.burnin = 1000,  
                  n.chains = 5)  
  
## module glm loaded  
  
## Compiling model graph  
##   Resolving undeclared variables
```

```
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 14
## Unobserved stochastic nodes: 7
## Total graph size: 84
##
## Initializing model

ancova.sim

## Inference for Bugs model at "ancova.txt", fit using jags,
## 5 chains, each with 50000 iterations (first 1000 discarded), n.thin = 2
## n.sims = 122500 iterations saved
##      mu.vect sd.vect  2.5%  25%  50%  75% 97.5% Rhat  n.eff
## b0[1]   13.297   5.912  1.635  9.462 13.287 17.098 25.099 1.001 120000
## b0[2]    9.883   4.355  1.282  7.074  9.866 12.671 18.586 1.001  60000
## b0[3]    7.955   4.192 -0.343  5.263  7.944 10.629 16.315 1.001 120000
## b1[1]    1.074   0.252  0.570  0.912  1.074  1.237  1.574 1.001 120000
## b1[2]    0.995   0.165  0.664  0.890  0.996  1.102  1.322 1.001  57000
## b1[3]    0.772   0.163  0.446  0.668  0.773  0.877  1.094 1.001 120000
## vv       3.625   1.567  1.560  2.518  3.305  4.373  7.560 1.001  46000
## deviance 57.241   4.193 50.979 54.141 56.607 59.653 67.101 1.001 120000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 8.8 and DIC = 66.0
## DIC is an estimate of expected predictive error (lower deviance is
## better).
```

2

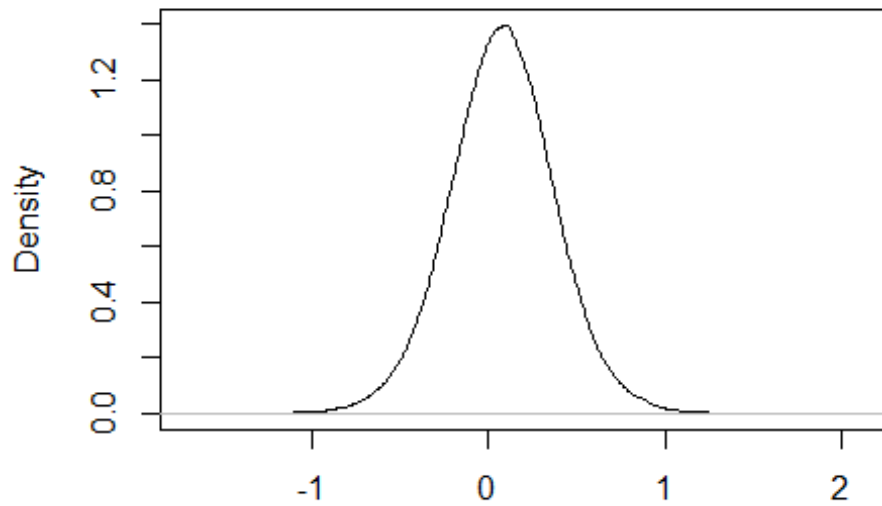
Check the differences between the slopes. There doesn't appear to be any significant difference in slope.

```
sims <- as.mcmc(ancova.sim)
chains <- as.matrix(sims)
sims <- as.mcmc(chains)

b11 <- chains[,4]
b12 <- chains[,5]
b13 <- chains[,6]

plot(density(b11 - b12))
```

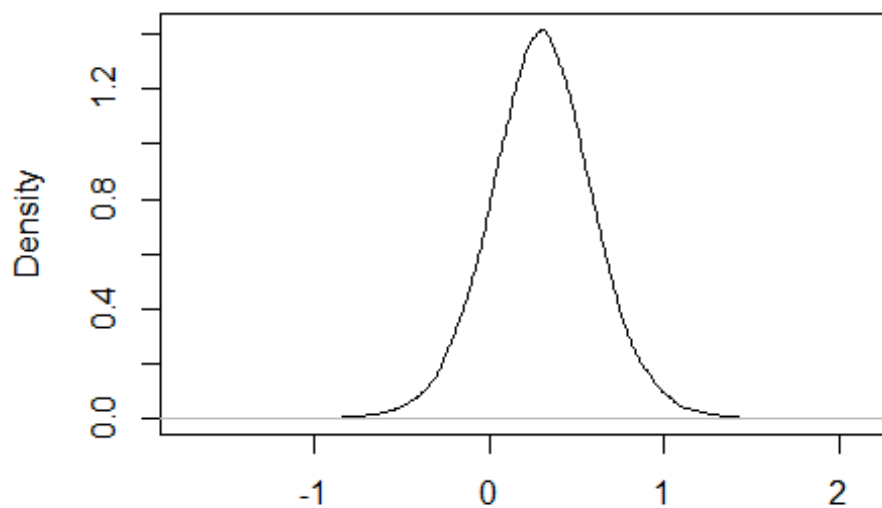
density.default(x = b11 - b12)



N = 122500 Bandwidth = 0.02497

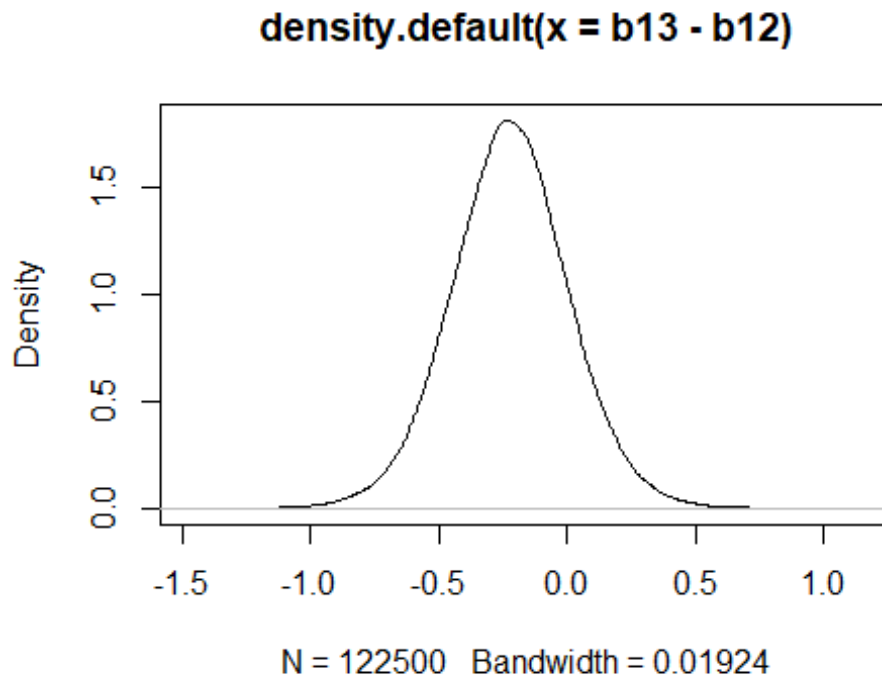
```
plot(density(b11 - b13))
```

density.default(x = b11 - b13)



N = 122500 Bandwidth = 0.02489

```
plot(density(b13 - b12))
```



```
quantile(b11-b12, probs = c(.025, .975))
##          2.5%          97.5%
## -0.5183146  0.6760433

quantile(b11-b13, probs = c(.025, .975))
##          2.5%          97.5%
## -0.2917701  0.9058877

quantile(b13-b12, probs = c(.025, .975))
##          2.5%          97.5%
## -0.6826974  0.2371405
```

3 Same slope and intercept for each treatment

```
library(R2jags)
head(crack)

##   tmt presales sales
## 1   1       21    38
## 2   1       26    39
## 3   1       22    36
## 4   1       28    45
```

```
## 5 1 19 33
## 6 2 34 43

mdl2 <-
  "model {

    for(i in 1:14){
      sales[i] ~ dnorm(mu[i], 1/vv);
      mu[i] <- b0 + b1*presales[i];
    }

    b0 ~ dnorm(25, .001);
    b1 ~ dnorm(0, .01);
    vv ~ dgamma(1.5, .5);

  }

"

writeLines(mdl2, 'ancova2.txt')
sales <- crack$sales
presales <- crack$presales
tmt <- crack$tmt

data.jags <- c('sales', 'presales', 'tmt')
parms <- c('b0', 'b1', 'vv')

ancova2.sim <- jags(data= data.jags, parameters.to.save = parms,
  model.file = 'ancova2.txt', inits = NULL,
  n.iter = 15000, n.thin = 2, n.burnin = 1000,
  n.chains = 5)

## Warning in jags.model(model.file, data = data, inits = init.values,
n.chains =
## n.chains, : Unused variable "tmt" in data

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 14
##   Unobserved stochastic nodes: 3
##   Total graph size: 63
##
## Initializing model

ancova2.sim
```

```
## Inference for Bugs model at "ancova2.txt", fit using jags,
## 5 chains, each with 15000 iterations (first 1000 discarded), n.thin = 2
## n.sims = 35000 iterations saved
##          mu.vect sd.vect   2.5%   25%   50%   75%   97.5%  Rhat n.eff
## b0      15.295   5.551  4.402 11.575 15.312 18.986 26.303 1.001 30000
## b1       0.743   0.221  0.304  0.595  0.742  0.890  1.178 1.001 35000
## vv      17.108   3.728 11.057 14.444 16.689 19.298 25.627 1.001 35000
## deviance 93.229   3.335 88.535 90.776 92.626 95.048 101.310 1.001 35000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 5.6 and DIC = 98.8
## DIC is an estimate of expected predictive error (lower deviance is
## better).
```

This second model gives a higher DIC (98.9) than the first model (66.1). I believe there should be a slope for each intercept.

4

Using 28 as a comparison point. Plots of differences in the values at presales = 28 shows that there are significant differences between the treatments. $b_{11} > b_{12} > b_1 > 3$.

SAS Code

```
data cracker; infile 'C:/Users/nateh/Downloads/cracker.dat'; input tmt presales sales; run;

proc mcmc data = cracker outpost = 'C:/Users/nateh/Documents/Stat
451/ancova_sas.sas7bdat' seed = 1234 nmc = 1000000 nbi= 1000 monitor=(parms) thin =
100 diagnostics=(rl ess autocorr) dic propcov=quanew; array b0[3] b01-b03; array b1[3]
b11-b13; parms b01 0; parms b02 0; parms b03 0; parms b11 0; parms b12 0; parms b13
0; parms vv 20; prior b01: ~ normal(0, var = 1000); prior b02: ~ normal(0, var = 1000);
prior b03: ~ normal(0, var = 1000); prior b11: ~ normal(0, var = 1000); prior b12: ~
normal(0, var = 1000); prior b13: ~ normal(0, var = 1000); prior vv ~ gamma(1.5, scale =
2); mu = b0[tmt]+b1[tmt]*presales; model sales ~ normal(mu, var = vv); run;
```

R code

```
library(haven)
sasoutput <- read_sas('ancova_sas.sas7bdat')
sas.sims <- as.mcmc(sasoutput)
sas.chains <- as.matrix(sas.sims)

raftery.diag(sas.chains)
```

```

##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##          Burn-in  Total  Lower bound  Dependence
##          (M)      (N)    (Nmin)       factor (I)
## Iteration 1724    1724   3746         0.46
## b01       10     11196  3746         2.99
## b02        4     5038   3746         1.34
## b03        5     5672   3746         1.51
## b11        8    10004   3746         2.67
## b12        3     4520   3746         1.21
## b13        4     4635   3746         1.24
## vv         2     3865   3746         1.03
## LogPrior   2     3994   3746         1.07
## LogLike    8     9474   3746         2.53
## LogPost    3     4520   3746         1.21

colnames(sas.chains)

## [1] "Iteration" "b01"      "b02"      "b03"      "b11"      "b12"
## [7] "b13"      "vv"       "LogPrior" "LogLike"  "LogPost"

sasb01 = sas.chains[,2]
sasb02 = sas.chains[,3]
sasb03 = sas.chains[,4]

sasb11 = sas.chains[,5]
sasb12 = sas.chains[,6]
sasb13 = sas.chains[,7]

mean(sasb01)

## [1] 12.48432

mean(sasb02)

## [1] 9.415161

mean(sasb03)

## [1] 7.496664

mean(sasb11)

## [1] 1.107999

mean(sasb12)

## [1] 1.012992

mean(sasb13)

```

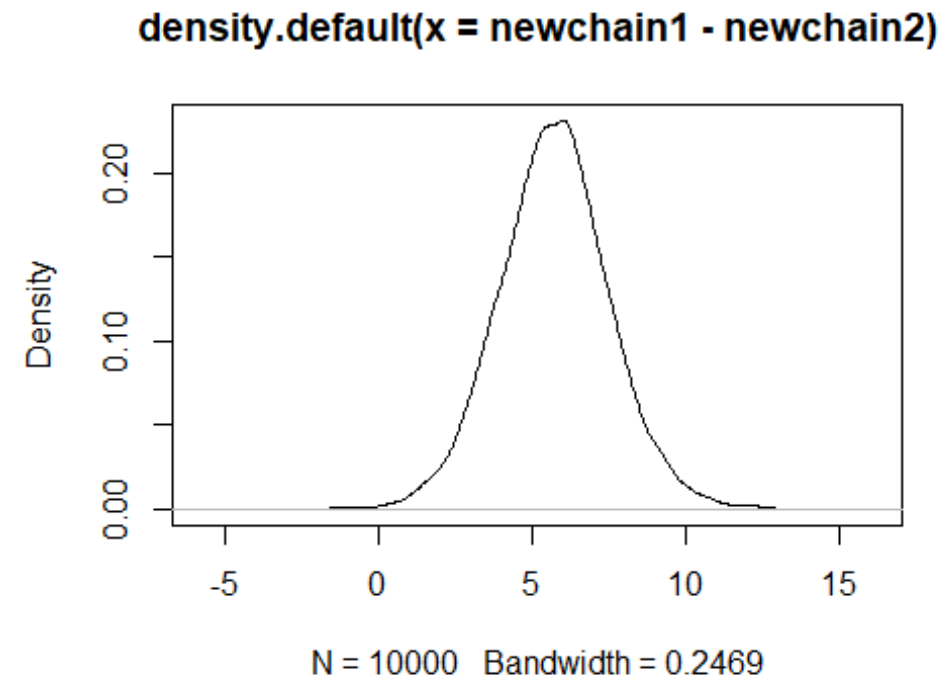
```
## [1] 0.7890095
```

```
newchain1 <- sasb11*28 + sasb01
```

```
newchain2 <- sasb12*28 + sasb02
```

```
newchain3 <- sasb13*28 + sasb03
```

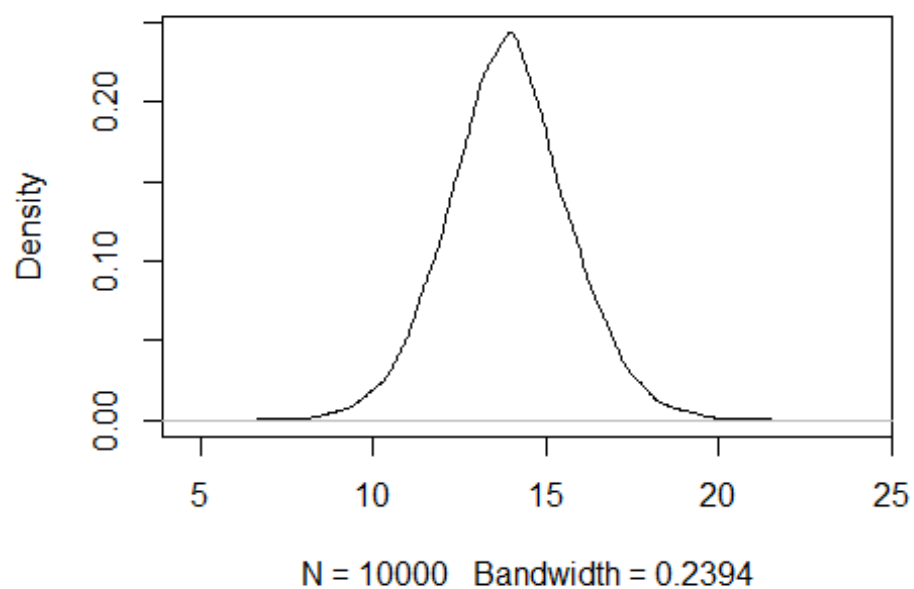
```
plot(density(newchain1 - newchain2))
```



```
plot(density(newchain1 - newchain3))
```

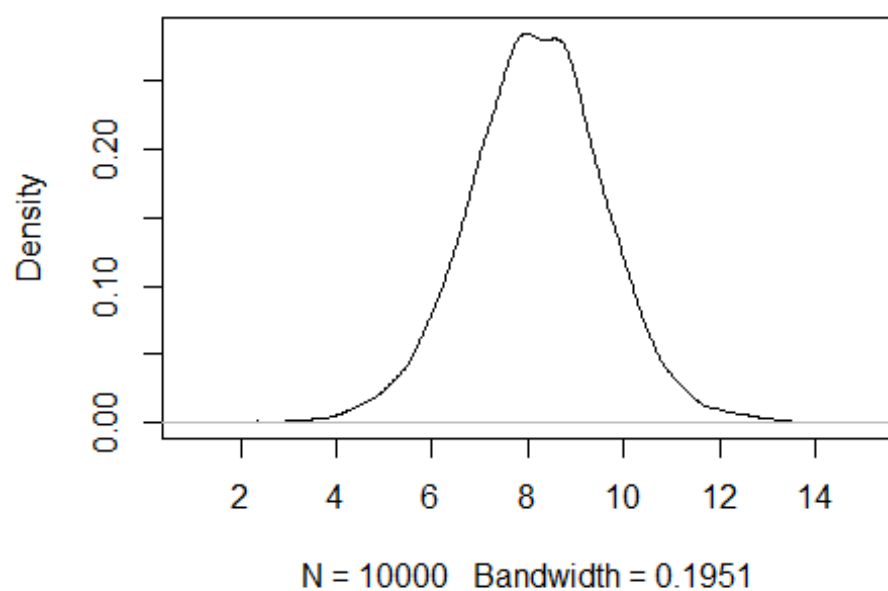


```
density.default(x = newchain1 - newchain3)
```



```
plot(density(newchain2 - newchain3))
```

```
density.default(x = newchain2 - newchain3)
```



```
quantile(newchain1 - newchain2, probs = c(.025, .975))
```

```
##      2.5%      97.5%  
## 2.149667 9.387946  
  
quantile(newchain1 - newchain3, probs = c(.025, .975))  
  
##      2.5%      97.5%  
## 10.40916 17.55988  
  
quantile(newchain2 - newchain3, probs = c(.025, .975))  
  
##      2.5%      97.5%  
## 5.328515 10.984614
```