wk7-hw-bayes

Part 1

1. Use at least 2 different methods proving that the groups in section 3.3 of part 1 of the workshop are different.

```
myDataFrame = read.csv( file=paste(dataPath,"wk7/TwoGroupIQ.csv",sep="/") )
y = as.numeric(myDataFrame[,"Score"])
x = as.numeric(as.factor(myDataFrame[,"Group"]))
(xLevels = levels(as.factor(myDataFrame[,"Group"])))
```

```
## [1] "Placebo"    "Smart Drug"
```

```
(Ntotal = length(y))
```

```
## [1] 120
```

```
# Specify the data in a list, for later shipment to JAGS:
dataList = list(
    y = y ,
    x = x ,
    Ntotal = Ntotal ,
    meanY = mean(y) ,
    sdY = sd(y)
)
dataList
```

```
## $y
##   [1] 102 107  92 101 110  68 119 106  99 103  90  93  79  89 137 119 126
##  [18] 110  71 114 100  95  91  99  97 106 106 129 115 124 137  73  69  95
```

```
##  [35] 102 116 111 134 102 110 139 112 122  84 129 112 127 106 113 109 208
##  [52] 114 107  50 169 133  50  97 139  72 100 144 112 109  98 106 101 100
##  [69] 111 117 104 106  89  84  88  94  78 108 102  95  99  90 116  97 107
##  [86] 102  91  94  95  86 108 115 108  88 102 102 120 112 100 105 105  88
## [103]  82 111  96  92 109  91  92 123  61  59 105 184  82 138  99  93  93
## [120]  72
##
## $x
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [106] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## $Ntotal
## [1] 120
##
## $meanY
## [1] 104.1333
##
## $sdY
## [1] 22.43532
```

Model params.

```
# Use the description for Stan from file "ch16_2.stan"
modelString = "
data {
    int<lower=1> Ntotal;
    int x[Ntotal];
    real y[Ntotal];
    real meanY;
    real sdY;
}
transformed data {
    real unifLo;
    real unifHi;
    real normalSigma;
```

```
    real expLambda;
    unifLo = sdY/1000;
    unifHi = sdY*1000;
    normalSigma = sdY*100;
    expLambda = 1/29.0;
}
parameters {
    real<lower=0> nuMinusOne; //New: definition of additional parameter nu
    real mu[2];
    real<lower=0> sigma[2];
}
transformed parameters {
    real<lower=0> nu;           //New: new parameter nu
    nu=nuMinusOne+1;           //New: shifting nu to avoid zero
}
model {
    sigma ~ uniform(unifLo, unifHi);
    mu ~ normal(meanY, normalSigma);
    nuMinusOne~exponential(expLambda);      //New: exponential prior for nu
    for (i in 1:Ntotal) {
        y[i] ~ student_t(nu, mu[x[i]], sigma[x[i]]);
    }

}
"
```

save

```
stanDsoRobust <- stan_model( model_code=modelString )
```

```
## In file included from D:/R-3.4.4/library/BH/include/boost/config.hpp:39:0,
##                  from D:/R-3.4.4/library/BH/include/boost/math/tools/config.hpp:13,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core/var.hpp:7,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core/gevv_vvv_vari.hpp:5,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core.hpp:12,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/mat.hpp:4,
```

```
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math.hpp:4,
##                  from D:/R-3.4.4/library/StanHeaders/include/src/stan/model/model_header.hpp:4,
##                  from file3a4c33b37610.cpp:8:
## D:/R-3.4.4/library/BH/include/boost/config/compiler/gcc.hpp:186:0: warning: "BOOST_NO_CXX11_RVALUE_REFERENCES"
redefined
##  #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
## cc1plus.exe: warning: unrecognized command line option "-Wno-ignored-attributes"
```

```
save(stanDsoRobust,file=paste(dataPath,"DSORobust1.Rds",sep="/"))
```

run

```
parameters = c( "mu" , "sigma" , "nu" )     # The parameters to be monitored
adaptSteps = 500               # Number of steps to "tune" the samplers
burnInSteps = 1000
nChains = 4
thinSteps = 1
numSavedSteps<-5000
# Get MC sample of posterior:

stanFitRobust <- sampling( object=stanDsoRobust ,
                     data = dataList ,
                     pars=c('nu','mu', 'sigma'),
                     chains = 3,
                     cores= 3,
                     iter = 50000,
                     warmup = 300,
                     thin = 1 )
```

save

```
save(stanFitRobust,file=paste(dataPath,"StanRobustFit2Groups.Rdata",sep="/"))
# load(paste(dataPath,"StanRobustFit2Groups.Rdata",sep="/"))
```

Print

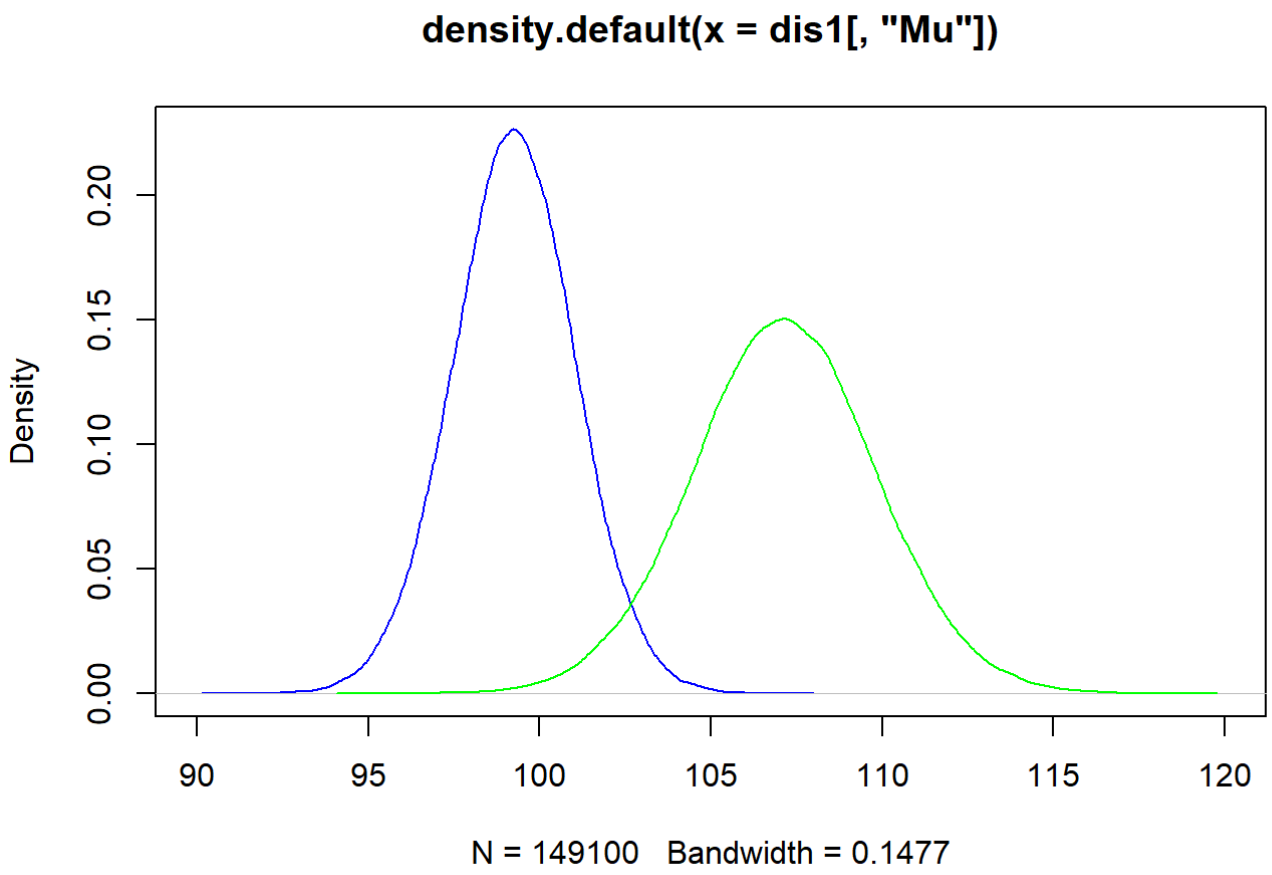
```
print(stanFitRobust)
```

```
## Inference for Stan model: 13367de882eb2ef7e53d40ca8e14e25e.
## 3 chains, each with iter=50000; warmup=300; thin=1;
## post-warmup draws per chain=49700, total post-warmup draws=149100.
##
##             mean se_mean   sd    2.5%     25%     50%     75%   97.5%
## nu          3.89    0.01 1.70    1.94    2.82    3.51    4.49    8.03
## mu[1]      99.25    0.00 1.79   95.70   98.07   99.26  100.45  102.77
## mu[2]     107.15    0.01 2.69  101.86  105.36  107.14  108.93  112.45
## sigma[1]   11.34    0.01 1.74    8.30   10.13   11.21   12.41   15.13
## sigma[2]   17.96    0.01 2.73   13.05   16.05   17.82   19.71   23.73
## lp__     -451.36    0.01 1.64 -455.40 -452.19 -451.02 -450.15 -449.20
##           n_eff Rhat
## nu       102308    1
## mu[1]    149100    1
## mu[2]    149100    1
## sigma[1] 117279    1
## sigma[2] 134181    1
## lp__      66907    1
##
## Samples were drawn using NUTS(diag_e) at Tue May 15 18:29:58 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Matches with workshop.

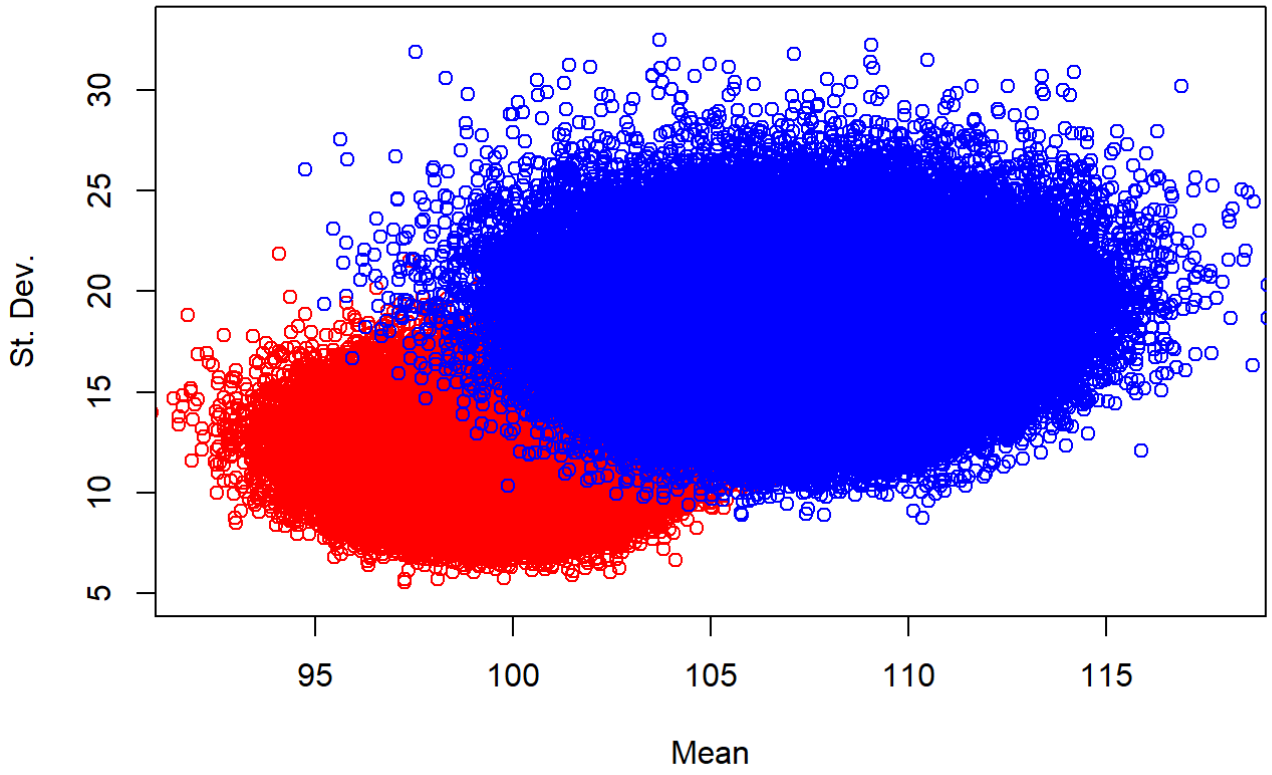
```
dis1<-cbind(Mu=rstan::extract(stanFitRobust,pars="mu[1]")$'mu[1]',
            Sigma=rstan::extract(stanFitRobust,pars="sigma[1]")$'sigma[1]')
dis2<-cbind(Mu=rstan::extract(stanFitRobust,pars="mu[2]")$'mu[2]',
            Sigma=rstan::extract(stanFitRobust,pars="sigma[2]")$'sigma[2]')
denDis1<-density(dis1[,"Mu"])
denDis2<-density(dis2[,"Mu"])
```

```
plot(denDis1,col="blue",xlim=c(90,120))
lines(denDis2,col="green")
```



Plots

```
plot(dis1,xlim=c(92,118),ylim=c(5,33),col="red",xlab="Mean",ylab="St. Dev.")
points(dis2,col="blue")
```



##Two different tests.

Ttest to compare means with different group vars. Reject null here.

```
t.test(dis1[,1],dis2[,1], var.equal=F, paired=FALSE)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  dis1[, 1] and dis2[, 1]
## t = -944.62, df = 259770, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.913784 -7.881012
## sample estimates:
## mean of x mean of y
##  99.25166 107.14906
```

Test two: let's run a regression to see if the samples have significant impact on mu values.

```
test_dta = data.frame(mu = c(dis1[,1],dis2[,1]),
                 group = c(rep(1,length(dis1[,1])),rep(2,length(dis2[,1]))))
                 )


head(test_dta)
```

```
##          mu group
## 1 101.77434     1
## 2  96.56251     1
## 3 100.75549     1
## 4  97.60919     1
## 5  95.62384     1
## 6  98.43982     1
```

GLM

```
glm1 = glm(mu ~ as.factor(group), data = test_dta)
summary(glm1)
```

```
##
## Call:
## glm(formula = mu ~ as.factor(group), data = test_dta)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -12.4110   -1.4432    0.0012    1.4422   11.9441
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       99.251659   0.005912 16789.1   <2e-16 ***
## as.factor(group)2  7.897398   0.008360   944.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 5.210728)
##
##     Null deviance: 6203430  on 298199  degrees of freedom
## Residual deviance: 1553829  on 298198  degrees of freedom
## AIC: 1338504
##
## Number of Fisher Scoring iterations: 2
```

Significant p values! Can see as well

Part 2

2. Analyze convergence of MCMC in section 5.1.4 of part 2 of the workshop, try to adjust parameters and rerun the process to obtain the a better quality of MCMC.

```
df = read.csv(file=paste(dataPath,"wk7/HierLinRegressData.csv",sep="/"))
head(df)
```

```
##   Subj    X     Y
## 1    1 60.2 145.6
## 2    1 61.5 157.3
## 3    1 61.7 165.6
## 4    1 62.3 158.8
```

```
## 5    1 67.6 196.1
## 6    1 69.2 183.9
```

```
dataList<-list(Ntotal=length(df$Y),
               y=df$Y,
               x=df$X,
               Ngroups = max(df$Subj),
               group = df$Subj
)
```

Model desc

```
modelStringPanel = "
data {
    int<lower=1> Ntotal;
    vector[Ntotal] y;
    vector[Ntotal] x;
    int<lower=1> Ngroups;
    int<lower=1, upper=Ngroups> group[Ntotal];
}
transformed data {
    real meanY;
    real sdY;
    vector[Ntotal] zy; // normalized y
    real meanX;
    real sdX;
    vector[Ntotal] zx; // normalized x
    meanY = mean(y);
    sdY = sd(y);
    zy = (y - meanY) / sdY;
    meanX = mean(x);
    sdX = sd(x);
    zx = (x - meanX) / sdX;
}
parameters {
    real<lower=0> zsigma;
```

```
    real<lower=0> nu;
    real zbeta0mu;
    real zbeta1mu;
    real<lower=0> zbeta0sigma;
    real<lower=0> zbeta1sigma;
    vector[Ngroups] zbeta0;
    vector[Ngroups] zbeta1;
}
transformed parameters {
    real<lower=0> sigma;
    real beta0mu;
    real beta1mu;
    vector[Ngroups] beta0;
    vector[Ngroups] beta1;
    // Transform to original scale:
    sigma = zsigma * sdY;
    beta0mu = meanY + zbeta0mu * sdY  - zbeta1mu * meanX * sdY / sdX;
    beta1mu = zbeta1mu * sdY / sdX;
    beta0 = meanY + zbeta0 * sdY  - zbeta1 * meanX * sdY / sdX; // vectorized
    beta1 = zbeta1 * sdY / sdX;                                 // vectorized
}
model {
    zsigma ~ uniform(0.001, 1000);
    nu ~ exponential(1/30.0);
    zbeta0mu ~ normal(0, 10.0^2);
    zbeta1mu ~ normal(0, 10.0^2);
    zbeta0sigma ~ uniform(0.001, 1000);
    zbeta1sigma ~ uniform(0.001, 1000);
    zbeta0 ~ normal(zbeta0mu, zbeta0sigma);  // vectorized
    zbeta1 ~ normal(zbeta1mu, zbeta1sigma);  // vectorized
    for (i in 1:Ntotal) {
        zy[i] ~ student_t(1+nu, zbeta0[group[i]] + zbeta1[group[i]] * x[i], zsigma);
    }
}"
```

Run

```
stanDsoRobustRegPanel <- stan_model( model_code=modelStringPanel )
```

```
## In file included from D:/R-3.4.4/library/BH/include/boost/config.hpp:39:0,
##                  from D:/R-3.4.4/library/BH/include/boost/math/tools/config.hpp:13,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core/var.hpp:7,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core/gevv_vvv_vari.hpp:5,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core.hpp:12,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/mat.hpp:4,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math.hpp:4,
##                  from D:/R-3.4.4/library/StanHeaders/include/src/stan/model/model_header.hpp:4,
##                  from file3a4c30c831ee.cpp:8:
## D:/R-3.4.4/library/BH/include/boost/config/compiler/gcc.hpp:186:0: warning: "BOOST_NO_CXX11_RVALUE_REFERENCES"
redefined
##  #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
## cc1plus.exe: warning: unrecognized command line option "-Wno-ignored-attributes"
```

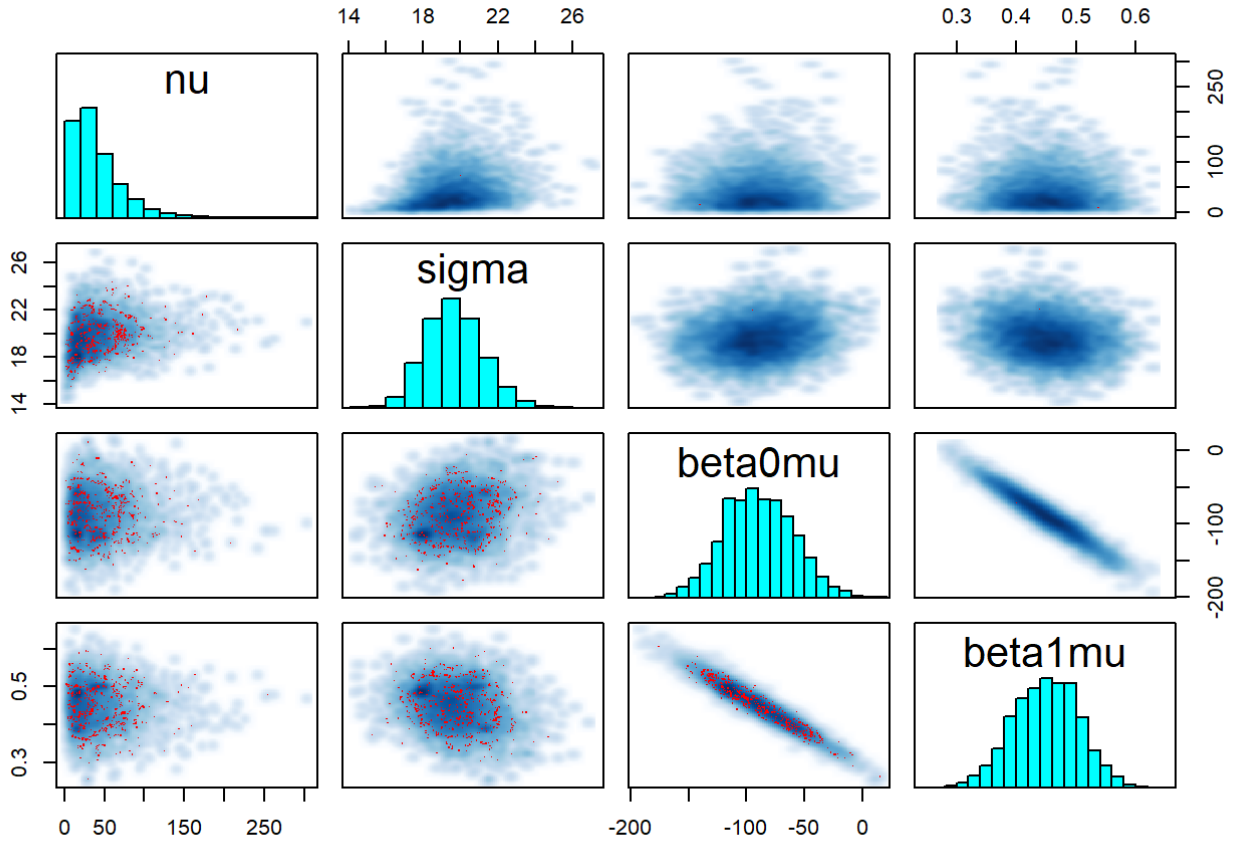
```
fit <- sampling (stanDsoRobustRegPanel,
                 data=dataList,
                 pars=c('nu', 'sigma', 'beta0mu', 'beta1mu', 'beta0', 'beta1',
                        'zbeta0sigma', 'zbeta1sigma'),
                 iter=5000,
                 chains = 4, cores = 4
                 )
```

```
## Warning: There were 625 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: There were 3 chains where the estimated Bayesian Fraction of Missing Information was low. See
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
# pairs
pairs(fit, pars=c('nu', 'sigma', 'beta0mu', 'beta1mu'))
```



Pairs plot shows high cor between betas.

improving convergence

Increase adaptive delta.

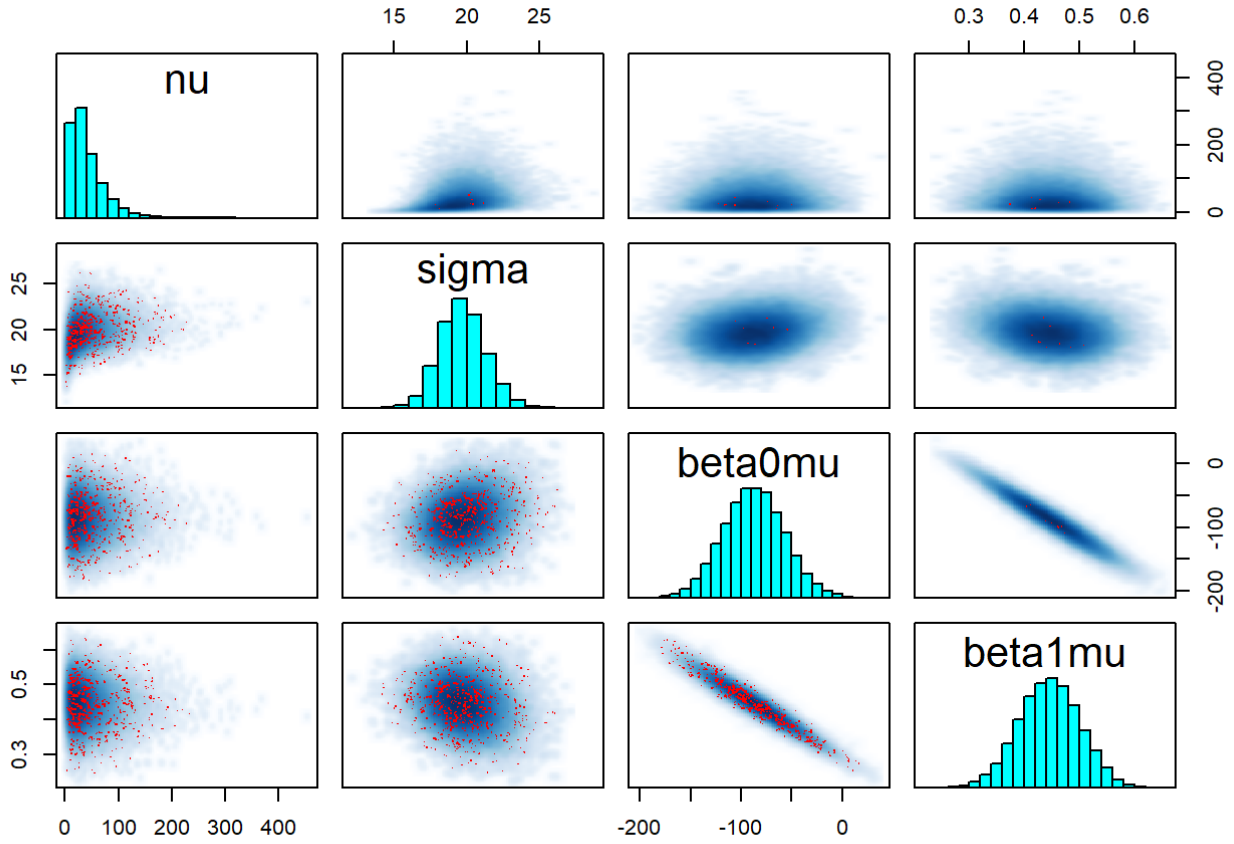
```
fit <- sampling (stanDsoRobustRegPanel,
                 data=dataList,
                 pars=c('nu', 'sigma', 'beta0mu', 'beta1mu', 'beta0', 'beta1',
                        'zbeta0sigma', 'zbeta1sigma'),
                 iter=50000,
                 chains = 4, cores = 4,
                 control = list(adapt_delta = 0.999, stepsize = 0.01, max_treedepth = 15)
                 )
```

```
## Warning: There were 1524 divergent transitions after warmup. Increasing adapt_delta above 0.999 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. See
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
pairs(fit, pars=c('nu', 'sigma', 'beta0mu', 'beta1mu'))
```



Still a lot of divergent points with a small step size and lhigh adapt_delta.

```
save(fit,file=paste(dataPath,"fitPanelModifControl11152016.Rdata",sep="/"))
```

Part 3

3. Consider data state.x77 from datasets used in multiple regression example in Statistical Analysis (MScA 31007).

Using LifeExp as response fit Gaussian and robust non-hierarchical regression models using Bayesian approach.

```
dta<-as.data.frame(state.x77)
colnames(dta)[4]<-"LifeExp"
colnames(dta)[6]<-"HSGrad"

head(dta)
```

```
##            Population Income Illiteracy LifeExp Murder HSGrad Frost   Area
## Alabama          3615   3624        2.1   69.05   15.1   41.3    20  50708
## Alaska            365   6315        1.5   69.31   11.3   66.7   152 566432
## Arizona          2212   4530        1.8   70.55    7.8   58.1    15 113417
## Arkansas         2110   3378        1.9   70.66   10.1   39.9    65  51945
## California      21198   5114        1.1   71.71   10.3   62.6    20 156361
## Colorado         2541   4884        0.7   72.06    6.8   63.9   166 103766
```

```
dim(dta)
```

```
## [1] 50  8
```

Let's do a simple regression.

```
## Create Stan data
dat <- list(N        = nrow(dta),
            p        = 8,
            Population  = dta$Population,
            Income     = dta$Income,
            Illiteracy      = dta$Illiteracy,
            Murder      = dta$Murder,
            HSGrad     = dta$HSGrad,
            Frost = dta$Frost,
            Area     = dta$Area,
            LifeExp = dta$LifeExp

            )
```

Model text.

```
modelStringPanel = "

data {
   // Define variables in data
   // Number of observations (an integer)
   int<lower=0> N;
   // Number of parameters
   int<lower=0> p;
   // Variables
   real LifeExp[N];
   real  Population[N];
   real  Income[N];
   real Illiteracy[N];
   real  Murder[N];
   real HSGrad[N];
   real  Frost[N];
   real Area[N];

 }

 parameters {
   // Define parameters to estimate
   real beta[p];

   // standard deviation (a positive real number)
   real<lower=0> sigma;
 }

 transformed parameters  {
   // Mean
   real mu[N];
 for (i in 1:N) {
     mu[i] <- beta[1] + beta[2]*Population[i] + beta[3]*Income[i] + beta[4]*Illiteracy[i] + beta[5]*Murder[i] + b
eta[6]*HSGrad[i] + beta[7]*Frost[i] + beta[8]*Area[i];
   }
```

```
 }

 model {
   // Prior part of Bayesian inference (flat if unspecified)

   // Likelihood part of Bayesian inference
     LifeExp ~ normal(mu, sigma);
 }

"
```

```
## Run Stan
resStan = stan(model_code = modelStringPanel, data = dat,
               chains = 3, iter = 3000, warmup = 500, thin = 10)
```

```
## In file included from D:/R-3.4.4/library/BH/include/boost/config.hpp:39:0,
##                  from D:/R-3.4.4/library/BH/include/boost/math/tools/config.hpp:13,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core/var.hpp:7,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core/gevv_vvv_vari.hpp:5,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/core.hpp:12,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math/rev/mat.hpp:4,
##                  from D:/R-3.4.4/library/StanHeaders/include/stan/math.hpp:4,
##                  from D:/R-3.4.4/library/StanHeaders/include/src/stan/model/model_header.hpp:4,
##                  from file3a4c51cb34cc.cpp:8:
## D:/R-3.4.4/library/BH/include/boost/config/compiler/gcc.hpp:186:0: warning: "BOOST_NO_CXX11_RVALUE_REFERENCES"
redefined
##  #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
## cc1plus.exe: warning: unrecognized command line option "-Wno-ignored-attributes"
##
## SAMPLING FOR MODEL '1c25c7f835d87f1e691e83fba3abf98c' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
```

```
##
##
## Iteration:    1 / 3000 [  0%]  (Warmup)
## Iteration:  300 / 3000 [ 10%]  (Warmup)
## Iteration:  501 / 3000 [ 16%]  (Sampling)
## Iteration:  800 / 3000 [ 26%]  (Sampling)
## Iteration: 1100 / 3000 [ 36%]  (Sampling)
## Iteration: 1400 / 3000 [ 46%]  (Sampling)
## Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Iteration: 3000 / 3000 [100%]  (Sampling)
##
##  Elapsed Time: 3.185 seconds (Warm-up)
##                7.48 seconds (Sampling)
##                10.665 seconds (Total)
##
##
## SAMPLING FOR MODEL '1c25c7f835d87f1e691e83fba3abf98c' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 3000 [  0%]  (Warmup)
## Iteration:  300 / 3000 [ 10%]  (Warmup)
## Iteration:  501 / 3000 [ 16%]  (Sampling)
## Iteration:  800 / 3000 [ 26%]  (Sampling)
## Iteration: 1100 / 3000 [ 36%]  (Sampling)
## Iteration: 1400 / 3000 [ 46%]  (Sampling)
## Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Iteration: 2600 / 3000 [ 86%]  (Sampling)
```

```
## Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Iteration: 3000 / 3000 [100%]  (Sampling)
##
##  Elapsed Time: 3.46 seconds (Warm-up)
##                15.217 seconds (Sampling)
##                18.677 seconds (Total)
##
##
## SAMPLING FOR MODEL '1c25c7f835d87f1e691e83fba3abf98c' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 3000 [  0%]  (Warmup)
## Iteration:  300 / 3000 [ 10%]  (Warmup)
## Iteration:  501 / 3000 [ 16%]  (Sampling)
## Iteration:  800 / 3000 [ 26%]  (Sampling)
## Iteration: 1100 / 3000 [ 36%]  (Sampling)
## Iteration: 1400 / 3000 [ 46%]  (Sampling)
## Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Iteration: 3000 / 3000 [100%]  (Sampling)
##
##  Elapsed Time: 2.341 seconds (Warm-up)
##                25.76 seconds (Sampling)
##                28.101 seconds (Total)
```

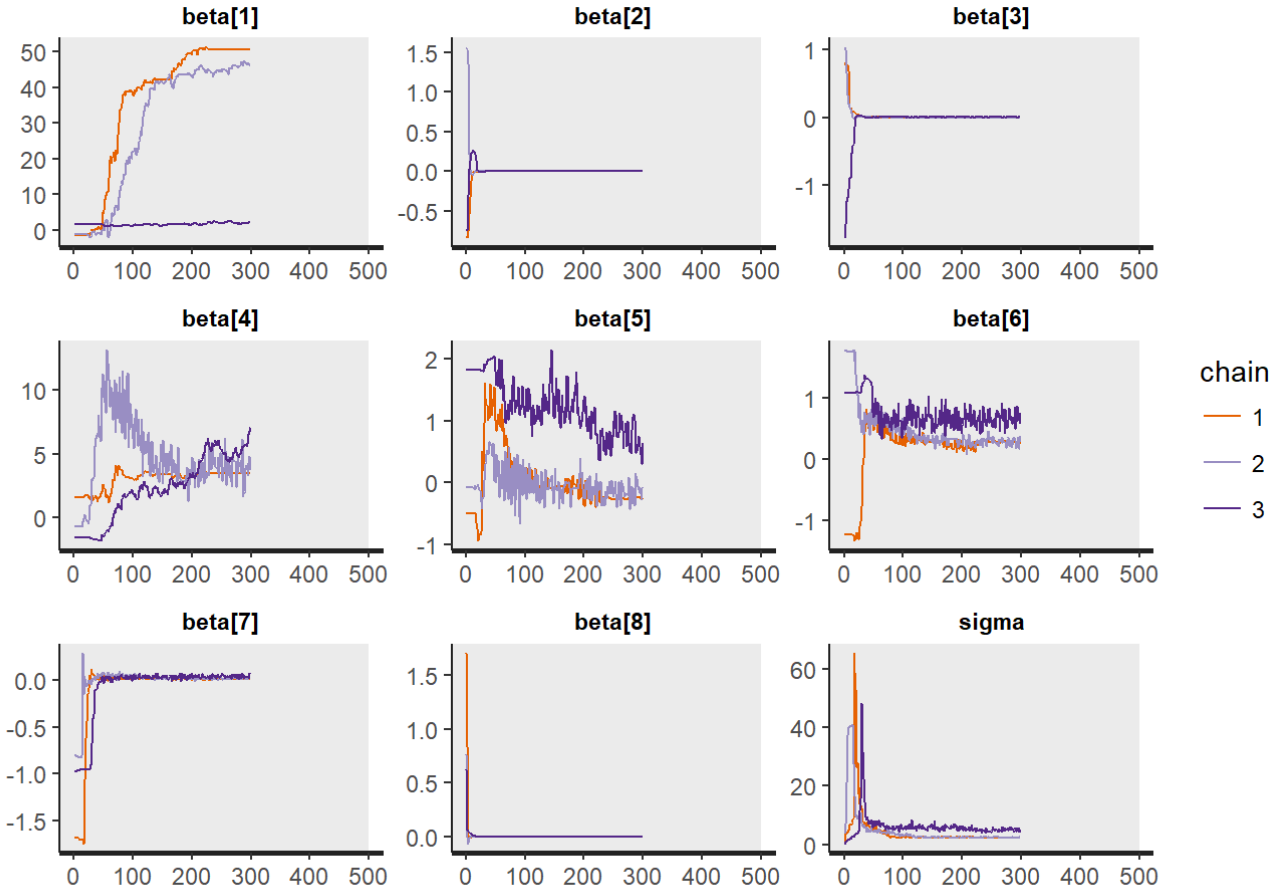
```
## Warning: There were 295 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: There were 340 transitions after warmup that exceeded the maximum treedepth. Increase max_treedepth a
bove 10. See
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Information was low. See
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Show traceplot
traceplot(resStan, pars = c("beta","sigma"), inc_warmup = TRUE)
```




```
#
print(resStan)
```

```
## Inference for Stan model: 1c25c7f835d87f1e691e83fba3abf98c.
## 3 chains, each with iter=3000; warmup=500; thin=10;
## post-warmup draws per chain=250, total post-warmup draws=750.
##
##           mean se_mean    sd    2.5%     25%    50%    75%  97.5% n_eff
## beta[1]  26.65   14.45 20.60    1.17    1.91  38.75  44.86  50.50     2
```

```
## beta[2]   0.00    0.00  0.00    0.00    0.00   0.00   0.00   0.00     5
## beta[3]   0.00    0.00  0.00    0.00    0.00   0.00   0.00   0.01     3
## beta[4]   3.87    0.77  2.03    0.48    2.95   3.48   4.68   9.67     7
## beta[5]   0.33    0.43  0.61   -0.36   -0.14   0.05   0.90   1.59     2
## beta[6]   0.46    0.13  0.20    0.19    0.29   0.37   0.63   0.88     2
## beta[7]   0.03    0.01  0.02    0.00    0.02   0.03   0.04   0.07     8
## beta[8]   0.00    0.00  0.00    0.00    0.00   0.00   0.00   0.00     2
## sigma     3.72    1.11  1.53    2.37    2.44   2.69   5.10   6.83     2
## mu[1]    67.42    0.19  1.63   64.14   66.55  67.52  68.27  70.65    75
## mu[2]    72.33    1.05  3.89   66.31   69.82  71.73  74.04  81.68    14
## mu[3]    72.02    0.25  1.56   68.28   71.17  72.06  73.03  75.04    40
## mu[4]    64.94    2.07  2.91   59.54   62.09  66.12  67.44  68.56     2
## mu[5]    72.91    0.48  2.57   68.35   71.37  72.65  74.22  78.95    28
## mu[6]    75.61    1.92  2.77   72.24   73.28  74.41  78.19  80.97     2
## mu[7]    75.44    0.61  1.89   72.54   74.08  75.09  76.56  79.79    10
## mu[8]    72.66    1.16  1.84   70.13   71.32  72.11  73.87  76.77     3
## mu[9]    70.84    1.07  2.05   67.73   69.59  70.38  71.58  76.06     4
## mu[10]   68.51    0.11  1.38   65.86   67.77  68.42  69.18  71.65   144
## mu[11]   77.19    1.15  2.82   73.24   75.16  76.42  78.78  83.78     6
## mu[12]   70.11    0.21  1.20   67.23   69.55  70.22  70.79  72.26    32
## mu[13]   73.25    1.86  2.77   69.75   71.11  72.33  75.25  79.23     2
## mu[14]   70.17    0.37  1.00   68.35   69.59  70.07  70.66  72.47     7
## mu[15]   70.98    0.64  1.20   67.92   70.36  71.25  71.81  72.69     4
## mu[16]   71.16    0.17  0.91   69.23   70.66  71.12  71.62  73.02    29
## mu[17]   65.39    1.32  2.08   61.23   63.61  66.09  67.02  68.25     2
## mu[18]   69.63    1.04  2.10   64.55   68.63  70.02  70.94  73.33     4
## mu[19]   68.73    1.72  2.55   63.19   66.81  69.69  70.66  71.90     2
## mu[20]   73.44    2.11  3.10   69.62   71.02  72.37  75.73  80.31     2
## mu[21]   74.00    0.37  1.40   71.30   73.27  73.96  74.66  77.24    14
## mu[22]   72.57    1.91  2.79   69.15   70.40  71.50  74.69  78.70     2
## mu[23]   70.80    0.99  1.64   66.71   69.84  71.34  71.94  72.96     3
## mu[24]   67.14    1.56  2.41   61.80   65.49  67.78  69.03  70.44     2
## mu[25]   67.58    0.41  1.33   64.19   67.07  67.81  68.33  69.81    11
## mu[26]   69.60    0.48  1.33   66.36   69.00  69.92  70.45  71.53     8
## mu[27]   70.79    0.65  1.20   67.77   70.14  71.08  71.65  72.47     3
## mu[28]   78.14    4.71  6.52   70.67   72.85  74.88  84.45  90.35     2
## mu[29]   72.76    0.14  1.20   70.33   72.06  72.70  73.31  75.36    73
```

```
## mu[30]   73.54    0.36  1.53   71.20   72.54  73.30  74.34  77.30    18
## mu[31]   73.23    0.72  2.25   68.60   72.19  73.15  74.17  78.37    10
## mu[32]   73.88    1.01  2.35   70.52   72.18  73.36  75.16  79.68     5
## mu[33]   66.11    1.32  2.04   61.62   64.50  66.75  67.66  68.83     2
## mu[34]   70.02    1.41  2.55   63.31   69.03  70.65  71.65  73.51     3
## mu[35]   71.04    0.06  1.20   68.93   70.29  70.93  71.66  73.51   343
## mu[36]   67.58    1.25  1.84   63.95   65.95  68.27  69.11  70.12     2
## mu[37]   68.68    0.66  1.93   64.32   67.66  69.05  70.08  71.47     9
## mu[38]   69.68    1.18  2.02   64.82   68.55  70.27  71.05  72.48     3
## mu[39]   69.26    1.95  2.97   62.35   67.23  70.38  71.42  73.10     2
## mu[40]   67.30    1.41  2.20   62.03   65.92  68.02  68.89  70.23     2
## mu[41]   67.32    2.01  2.88   61.37   64.76  68.62  69.71  70.78     2
## mu[42]   66.95    0.87  1.48   63.75   65.85  67.38  68.08  69.01     3
## mu[43]   65.91    2.46  3.67   57.85   63.10  67.14  68.73  70.83     2
## mu[44]   73.48    0.25  1.91   70.17   72.32  73.29  74.38  78.28    60
## mu[45]   71.71    0.48  1.47   69.26   70.80  71.47  72.39  75.15     9
## mu[46]   70.82    0.59  1.25   68.98   69.97  70.57  71.41  74.00     5
## mu[47]   71.32    0.40  1.86   67.54   70.21  71.33  72.26  75.52    22
## mu[48]   65.03    2.31  3.19   59.32   61.74  66.49  67.79  68.88     2
## mu[49]   69.83    1.22  1.83   65.63   68.37  70.55  71.19  72.06     2
## mu[50]   74.32    1.77  2.63   71.17   72.22  73.34  76.45  79.75     2
## lp__    -80.16   16.42 23.21 -116.17 -105.24 -67.16 -59.69 -52.83     2
##         Rhat
## beta[1] 2.97
## beta[2] 1.22
## beta[3] 1.70
## beta[4] 1.62
## beta[5] 2.67
## beta[6] 1.81
## beta[7] 1.27
## beta[8] 1.96
## sigma   2.55
## mu[1]   1.04
## mu[2]   1.07
## mu[3]   1.06
## mu[4]   2.65
## mu[5]   1.06
```

```
## mu[6]   2.31
## mu[7]   1.26
## mu[8]   1.68
## mu[9]   1.43
## mu[10]  1.02
## mu[11]  1.28
## mu[12]  1.08
## mu[13]  2.02
## mu[14]  1.33
## mu[15]  1.31
## mu[16]  1.09
## mu[17]  1.78
## mu[18]  1.46
## mu[19]  1.96
## mu[20]  2.12
## mu[21]  1.19
## mu[22]  2.22
## mu[23]  1.54
## mu[24]  1.85
## mu[25]  1.34
## mu[26]  1.16
## mu[27]  1.33
## mu[28]  2.91
## mu[29]  1.05
## mu[30]  1.14
## mu[31]  1.31
## mu[32]  1.23
## mu[33]  1.77
## mu[34]  1.40
## mu[35]  1.03
## mu[36]  2.33
## mu[37]  1.26
## mu[38]  1.45
## mu[39]  1.95
## mu[40]  1.80
## mu[41]  2.23
## mu[42]  1.54
```

```
## mu[43]  1.99
## mu[44]  1.06
## mu[45]  1.10
## mu[46]  1.23
## mu[47]  1.11
## mu[48]  2.94
## mu[49]  1.86
## mu[50]  2.00
## lp__    2.92
##
## Samples were drawn using NUTS(diag_e) at Wed May 16 00:09:45 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
resStan@model_pars
```

```
## [1] "beta"  "sigma" "mu"    "lp__"
```

Checking plot

```
plot(resStan,pars=c('beta', 'sigma', 'mu'))
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```