Homework 3 Suglia

Elena Suglia

11/17/2019

Question 1

Using the "litterbags.csv" data set, create and run a model in JAGS that corresponds to this one:

```
lmer(N_min_rate~Celastrus + (1|Plot), data=litterbags)
```

This data set contains nitrogen mineralization rates at 7 different plots where the treatment was whether or not the invasive liana Celastrus orbiculatus was present or not.

In your answer, please include the model code and the means and standard deviations of the intercept, slope, group-level (i.e. among-plot) variance and individual-level (within-plot) variance. Briefly report what you did to check model convergence.

Read in data

```
d = read_csv("litterbags.csv")
```

Model code in text file

```
 model {

   # Likelihood
  for (i in 1:n) {
    y[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[Plot[i]] + b*x[i] # regression with intercept for each plot
  }

  # group-level model
  for (j in 1:7) { # 7 plots
    a[j] ~ dnorm(mu.a, tau.a) # separate intercepts for every plot, no pooling of information
  }

  #priors
  mu.a ~ dnorm(0, 0.1) # overall mean
  sigma.a ~ dnorm(0, 0.5)T(0,) # weakly informative prior on the group-level standard deviation
  tau.a <- pow(sigma.a, -2) # group-level precision

  # individual-level priors
  b ~ dnorm(0, 0.1) #  prior for slope
  sigma.y ~ dnorm(0, 0.1)T(0,) # weakly informative prior on the individual-level sd
  tau.y <- pow(sigma.y, -2) # individual-level precision
}
```

Define data needed for model

```
length(unique(d$Plot)) # 7 plots
d2.data <- list(n=nrow(d), y=d$N_min_rate, x=d$Celastrus,
plot.index=d$Plot, n.plots=length(unique(d$Plot)))
```

Have JAGS generate starting values

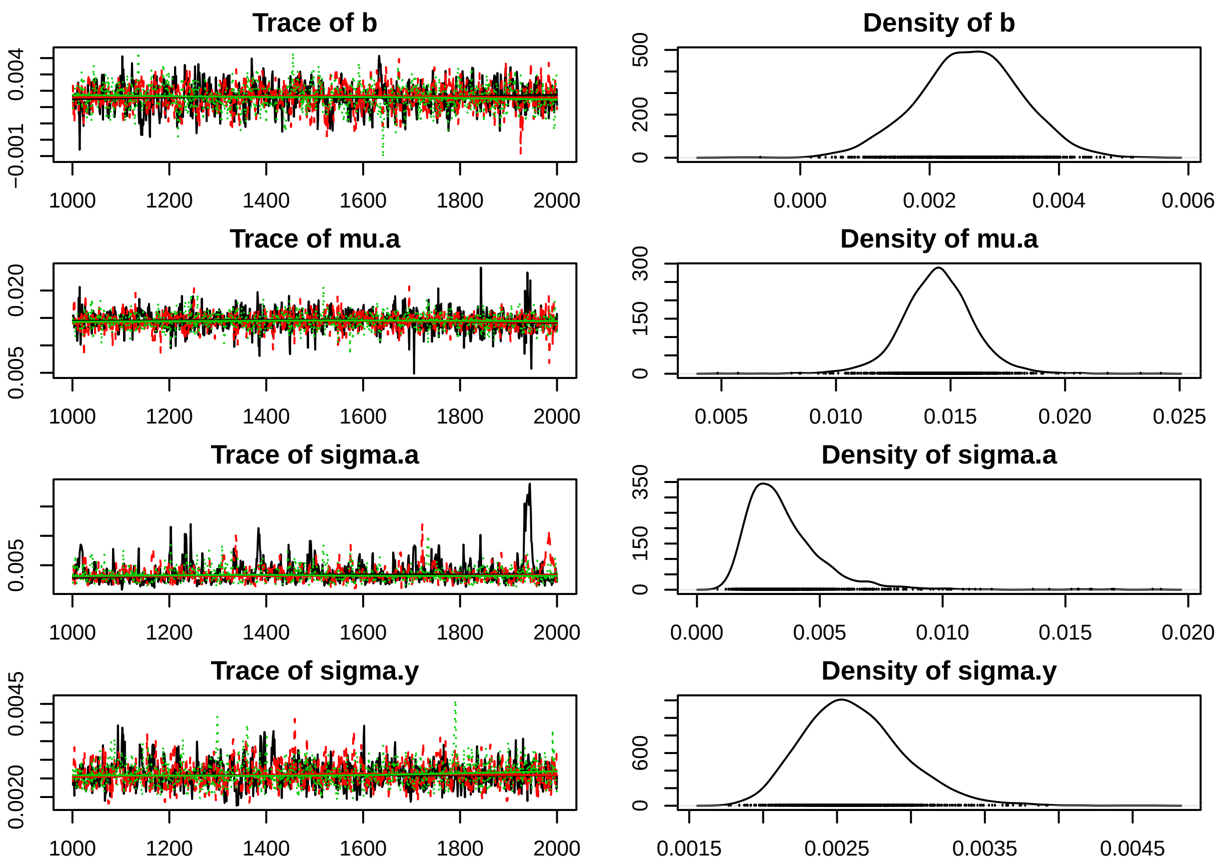
```
d2.inits <- list(list(sigma.a = 1, sigma.y=2), list(sigma.a = 2,
sigma.y=1), list(sigma.a = 5, sigma.y=0.4))
d2 <- jags.model("litterbags_jags_hw3.txt", data=d2.data, inits = d2.inits,
n.chains=3, n.adapt=1000)
```

Do a trial run with some parameters, and check convergence

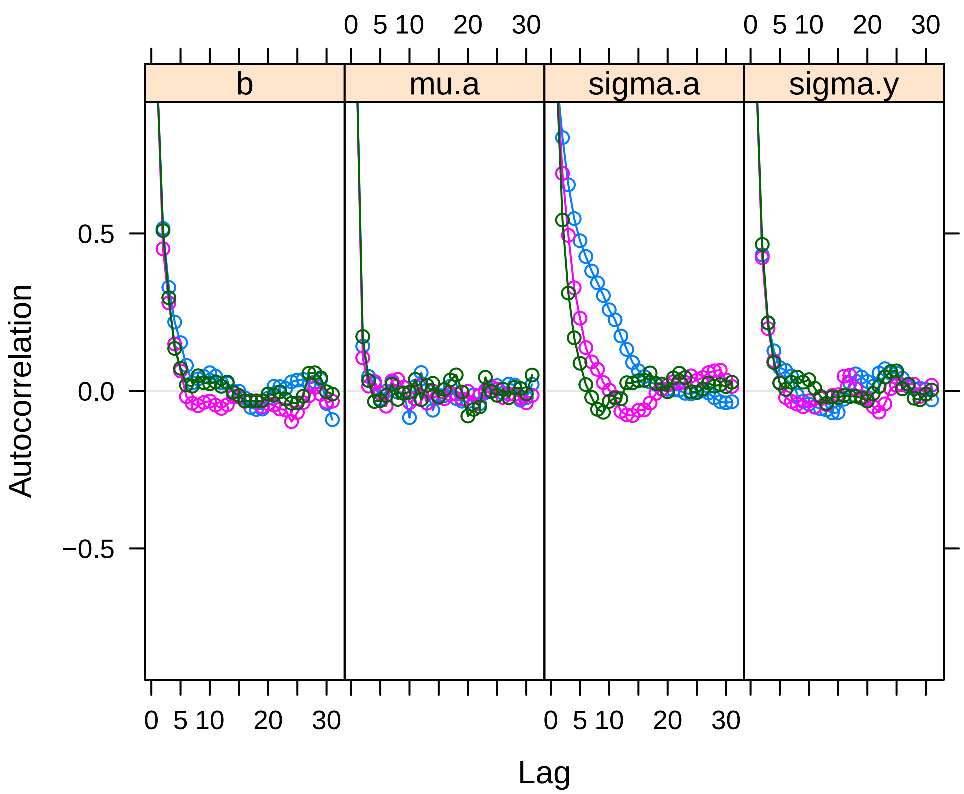
```
d2.samp <- coda.samples(d2, c("b", "sigma.y", "mu.a", "sigma.a"), n.iter=1000)
summary(d2.samp)
```

```
##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean        SD  Naive SE Time-series SE
## b       0.002613 0.0007992 1.459e-05      2.671e-05
## mu.a    0.014442 0.0015870 2.898e-05      3.339e-05
## sigma.a 0.003557 0.0016984 3.101e-05      8.202e-05
## sigma.y 0.002616 0.0003446 6.291e-06      1.010e-05
##
## 2. Quantiles for each variable:
##
##              2.5%      25%      50%      75%    97.5%
## b       0.001025 0.002110 0.002628 0.003136 0.004138
## mu.a    0.011243 0.013498 0.014455 0.015364 0.017573
## sigma.a 0.001612 0.002467 0.003180 0.004195 0.007713
## sigma.y 0.002041 0.002375 0.002584 0.002820 0.003364
```

```
par(mar=rep(2, 4))
plot(d2.samp)
```




```r
acfplot(d2.samp)
```



```r
gelman.diag(d2.samp)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## b              1.00       1.00
## mu.a           1.01       1.01
## sigma.a        1.06       1.11
## sigma.y        1.00       1.00
##
## Multivariate psrf
##
## 1.01
```

Add more iterations and include all parameters of interest

```r
d2.samp <- coda.samples(d2, c("a", "b", "sigma.y", "mu.a", "sigma.a", "y.hat"),
n.iter=5000, thin=5)
```

Look at the results

```r
d2.summ <- summary(d2.samp)
d2.stats <- as.data.frame(d2.summ$statistics)
head(d2.stats, 100)
```

```
##                  Mean           SD     Naive SE Time-series SE
## a[1]      0.018064523 0.0011421164 2.085210e-05   2.120725e-05
## a[2]      0.012926552 0.0010826086 1.976564e-05   2.023522e-05
## a[3]      0.017671907 0.0012642898 2.308267e-05   2.412744e-05
## a[4]      0.012157906 0.0011192934 2.043541e-05   2.086360e-05
## a[5]      0.014941101 0.0011006553 2.009512e-05   2.028074e-05
## a[6]      0.012637983 0.0011322999 2.067287e-05   2.067069e-05
## a[7]      0.012722819 0.0011138184 2.033545e-05   2.033488e-05
## b         0.002671389 0.0008174323 1.492420e-05   1.533336e-05
## mu.a      0.014516050 0.0015507554 2.831279e-05   2.935845e-05
## sigma.a   0.003508709 0.0015679564 2.862684e-05   3.530598e-05
## sigma.y   0.002614988 0.0003468084 6.331826e-06   6.472411e-06
## y.hat[1]  0.020735912 0.0011709493 2.137851e-05   2.182648e-05
## y.hat[2]  0.020735912 0.0011709493 2.137851e-05   2.182648e-05
## y.hat[3]  0.020735912 0.0011709493 2.137851e-05   2.182648e-05
## y.hat[4]  0.018064523 0.0011421164 2.085210e-05   2.120725e-05
## y.hat[5]  0.018064523 0.0011421164 2.085210e-05   2.120725e-05
## y.hat[6]  0.018064523 0.0011421164 2.085210e-05   2.120725e-05
## y.hat[7]  0.015597941 0.0010773428 1.966950e-05   1.966807e-05
## y.hat[8]  0.015597941 0.0010773428 1.966950e-05   1.966807e-05
## y.hat[9]  0.015597941 0.0010773428 1.966950e-05   1.966807e-05
## y.hat[10] 0.012926552 0.0010826086 1.976564e-05   2.023522e-05
## y.hat[11] 0.012926552 0.0010826086 1.976564e-05   2.023522e-05
## y.hat[12] 0.012926552 0.0010826086 1.976564e-05   2.023522e-05
## y.hat[13] 0.020343296 0.0012238090 2.234359e-05   2.234106e-05
## y.hat[14] 0.020343296 0.0012238090 2.234359e-05   2.234106e-05
```

```
## y.hat[15] 0.020343296 0.0012238090 2.234359e-05   2.234106e-05
## y.hat[16] 0.017671907 0.0012642898 2.308267e-05   2.412744e-05
## y.hat[17] 0.017671907 0.0012642898 2.308267e-05   2.412744e-05
## y.hat[18] 0.014829295 0.0011030265 2.013842e-05   2.058756e-05
## y.hat[19] 0.014829295 0.0011030265 2.013842e-05   2.058756e-05
## y.hat[20] 0.014829295 0.0011030265 2.013842e-05   2.058756e-05
## y.hat[21] 0.012157906 0.0011192934 2.043541e-05   2.086360e-05
## y.hat[22] 0.012157906 0.0011192934 2.043541e-05   2.086360e-05
## y.hat[23] 0.012157906 0.0011192934 2.043541e-05   2.086360e-05
## y.hat[24] 0.017612490 0.0010973523 2.003482e-05   2.024061e-05
## y.hat[25] 0.017612490 0.0010973523 2.003482e-05   2.024061e-05
## y.hat[26] 0.017612490 0.0010973523 2.003482e-05   2.024061e-05
## y.hat[27] 0.014941101 0.0011006553 2.009512e-05   2.028074e-05
## y.hat[28] 0.014941101 0.0011006553 2.009512e-05   2.028074e-05
## y.hat[29] 0.014941101 0.0011006553 2.009512e-05   2.028074e-05
## y.hat[30] 0.015309372 0.0011272386 2.058047e-05   2.058195e-05
## y.hat[31] 0.015309372 0.0011272386 2.058047e-05   2.058195e-05
## y.hat[32] 0.015309372 0.0011272386 2.058047e-05   2.058195e-05
## y.hat[33] 0.012637983 0.0011322999 2.067287e-05   2.067069e-05
## y.hat[34] 0.012637983 0.0011322999 2.067287e-05   2.067069e-05
## y.hat[35] 0.012637983 0.0011322999 2.067287e-05   2.067069e-05
## y.hat[36] 0.015394208 0.0011191942 2.043360e-05   2.013723e-05
## y.hat[37] 0.015394208 0.0011191942 2.043360e-05   2.013723e-05
## y.hat[38] 0.015394208 0.0011191942 2.043360e-05   2.013723e-05
## y.hat[39] 0.012722819 0.0011138184 2.033545e-05   2.033488e-05
## y.hat[40] 0.012722819 0.0011138184 2.033545e-05   2.033488e-05
## y.hat[41] 0.012722819 0.0011138184 2.033545e-05   2.033488e-05
```

```r
y.hat.rows <- grep("y.hat", rownames(d2.stats))
y.hat <- d2.stats$Mean[y.hat.rows] # To get fitted values of the model,
# look at the stats and pull out the rows corresponding to y.hat
resids <- d$N_min_rate - y.hat
# for convenience, we can then put those values into our data frame and examine them
d <- cbind(d, y.hat = y.hat, resid = resids)
```

Some summary statistics:

The means and standard deviations of:

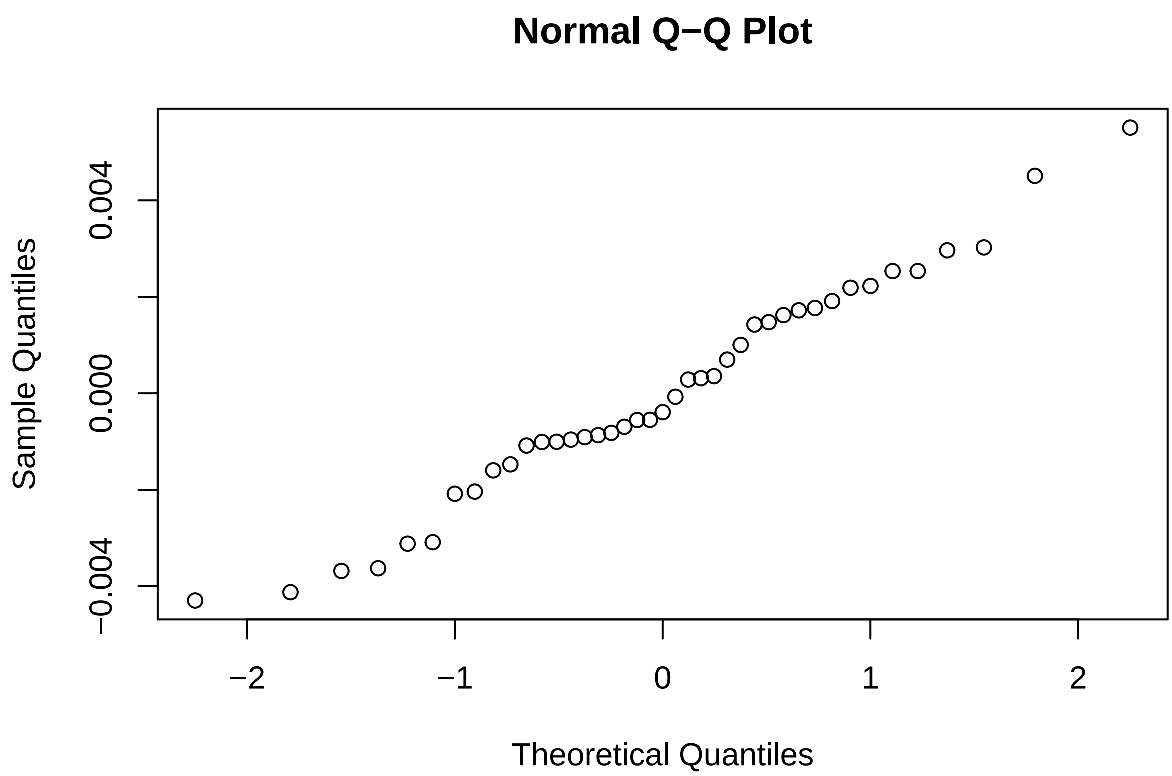
- slope = b
- intercept = mu.a
- group-level (i.e. among-plot) variance = sigma.a
- individual-level (within-plot) variance = sigma.y

```
##                   Mean           SD
## a[1]       0.018064523 0.0011421164
## a[2]       0.012926552 0.0010826086
## a[3]       0.017671907 0.0012642898
## a[4]       0.012157906 0.0011192934
## a[5]       0.014941101 0.0011006553
## a[6]       0.012637983 0.0011322999
## a[7]       0.012722819 0.0011138184
## b          0.002671389 0.0008174323
## mu.a       0.014516050 0.0015507554
```

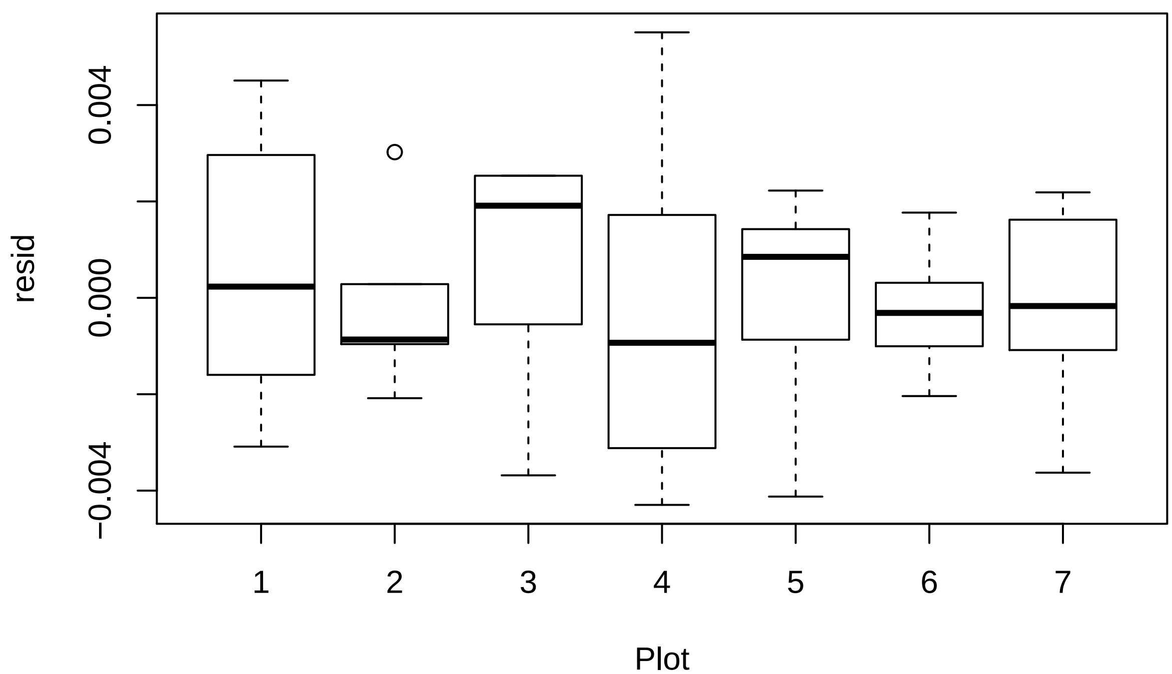
```
## sigma.a    0.003508709 0.0015679564
## sigma.y    0.002614988 0.0003468084
## y.hat[1]   0.020735912 0.0011709493
## y.hat[2]   0.020735912 0.0011709493
## y.hat[3]   0.020735912 0.0011709493
## y.hat[4]   0.018064523 0.0011421164
## y.hat[5]   0.018064523 0.0011421164
## y.hat[6]   0.018064523 0.0011421164
## y.hat[7]   0.015597941 0.0010773428
## y.hat[8]   0.015597941 0.0010773428
## y.hat[9]   0.015597941 0.0010773428
## y.hat[10]  0.012926552 0.0010826086
## y.hat[11]  0.012926552 0.0010826086
## y.hat[12]  0.012926552 0.0010826086
## y.hat[13]  0.020343296 0.0012238090
## y.hat[14]  0.020343296 0.0012238090
## y.hat[15]  0.020343296 0.0012238090
## y.hat[16]  0.017671907 0.0012642898
## y.hat[17]  0.017671907 0.0012642898
## y.hat[18]  0.014829295 0.0011030265
## y.hat[19]  0.014829295 0.0011030265
## y.hat[20]  0.014829295 0.0011030265
## y.hat[21]  0.012157906 0.0011192934
## y.hat[22]  0.012157906 0.0011192934
## y.hat[23]  0.012157906 0.0011192934
## y.hat[24]  0.017612490 0.0010973523
## y.hat[25]  0.017612490 0.0010973523
## y.hat[26]  0.017612490 0.0010973523
## y.hat[27]  0.014941101 0.0011006553
## y.hat[28]  0.014941101 0.0011006553
## y.hat[29]  0.014941101 0.0011006553
## y.hat[30]  0.015309372 0.0011272386
## y.hat[31]  0.015309372 0.0011272386
## y.hat[32]  0.015309372 0.0011272386
## y.hat[33]  0.012637983 0.0011322999
## y.hat[34]  0.012637983 0.0011322999
## y.hat[35]  0.012637983 0.0011322999
## y.hat[36]  0.015394208 0.0011191942
## y.hat[37]  0.015394208 0.0011191942
## y.hat[38]  0.015394208 0.0011191942
## y.hat[39]  0.012722819 0.0011138184
## y.hat[40]  0.012722819 0.0011138184
## y.hat[41]  0.012722819 0.0011138184
```

See how well the model ran

```r
qqnorm(d$resid)
```




```r
boxplot(resid~Plot, data=d)
```




```r
# Now that random intercepts for plot are included in the model,
# does it look like the residuals differ strongly by plot?
```

Look at DIC

```
d2.DIC <- dic.samples(d2, n.iter=5000, thin=5)
d2.DIC
```

```
## Mean deviance:  -372.9
## penalty 9.217
## Penalized deviance: -363.6
```

Question 2

Fit a hierarchical (multilevel) model in JAGS using the data sets "immunity.csv" and "patient_age.csv". The model should include "immune.level" as the response variable, "time" as an individual-level predictor, and "age" as a group-level predictor.

These data are repeated measures on individual patients, so each patient is a "group" in the data set. Each row in the data set "immunity.csv" is one observation on one patient. Each row in the data set "patient_age.csv" is the age of each patient at time the study began – so this is a group level predictor, with one row of data per patient. In other words, here "age" is analogous to county-level bedrock uranium content in the Gelman & Hill radon example.

Hints: You can use the column "patient" in "immunity.csv" to index the random intercept for patient. You can then use the column "age" in "patient_age.csv" in the group-level regression that explains some of the variation in the random intercepts. There is a model like this on page 361 of Gelman & Hill.

In your answer, include the model you created. Also report the means and standard deviations of: - the slopes of the individual-data-point-level and "group"-level (i.e. patient-level) regressions - the individual-data-point-level and "group"-level variance parameters

Read in data

```
im = read_csv("immunity.csv")
pa = read_csv("patient_age.csv")
```

Model to put in a txt file:

```
model {
  for(i in 1:n){
    y[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[patient.index[i]] + b*x[i]
  }
  b ~ dnorm(0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0, 100)

  for(j in 1:n.patients){
    a[j] ~ dnorm(a.hat[j], tau.a)
a.hat[j] <- g.0 + g.1*age[j]
  }
  g.0 ~ dnorm(0, .0001)
  g.1 ~ dnorm(0, .0001)
```

```
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}
```

Define data needed for model

```
im.data <- list(y=im$immune.level, x=im$time, patient.index = im$patient, age=pa$age, n.patients=length
```

Allow JAGS generate starting values

```
im.inits <- list(list(sigma.a = 1, sigma.y=2), list(sigma.a = 2, sigma.y=1),
list(sigma.a = 5, sigma.y=0.4))
im.model <- jags.model("immunity_jags_hw3.txt", data=im.data, inits = im.inits,
n.chains=3, n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 435
##    Unobserved stochastic nodes: 105
##    Total graph size: 2157
##
## Initializing model
```

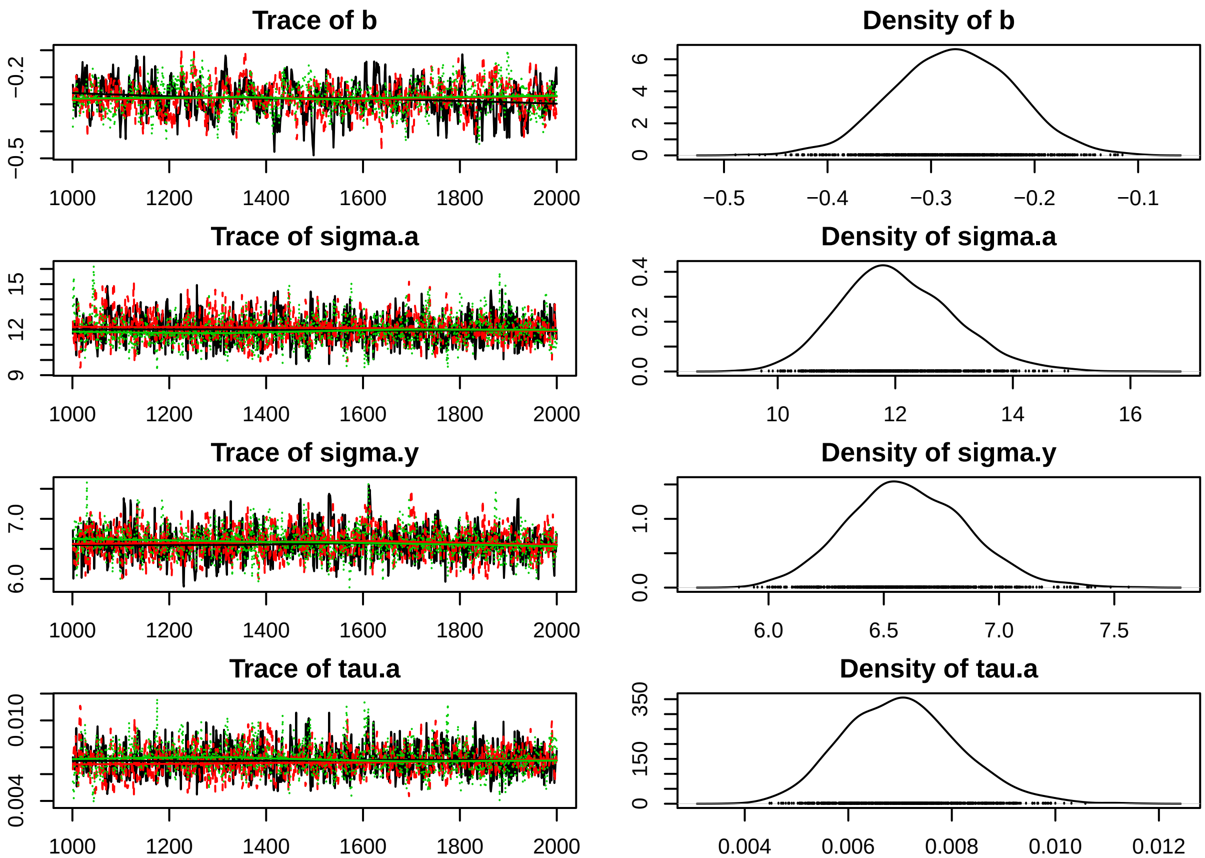
Do a trial run with some parameters, and check convergence

```
im.samp <- coda.samples(im.model, c("b", "sigma.a", "sigma.y", "tau.a"), n.iter=1000)
summary(im.samp)
```

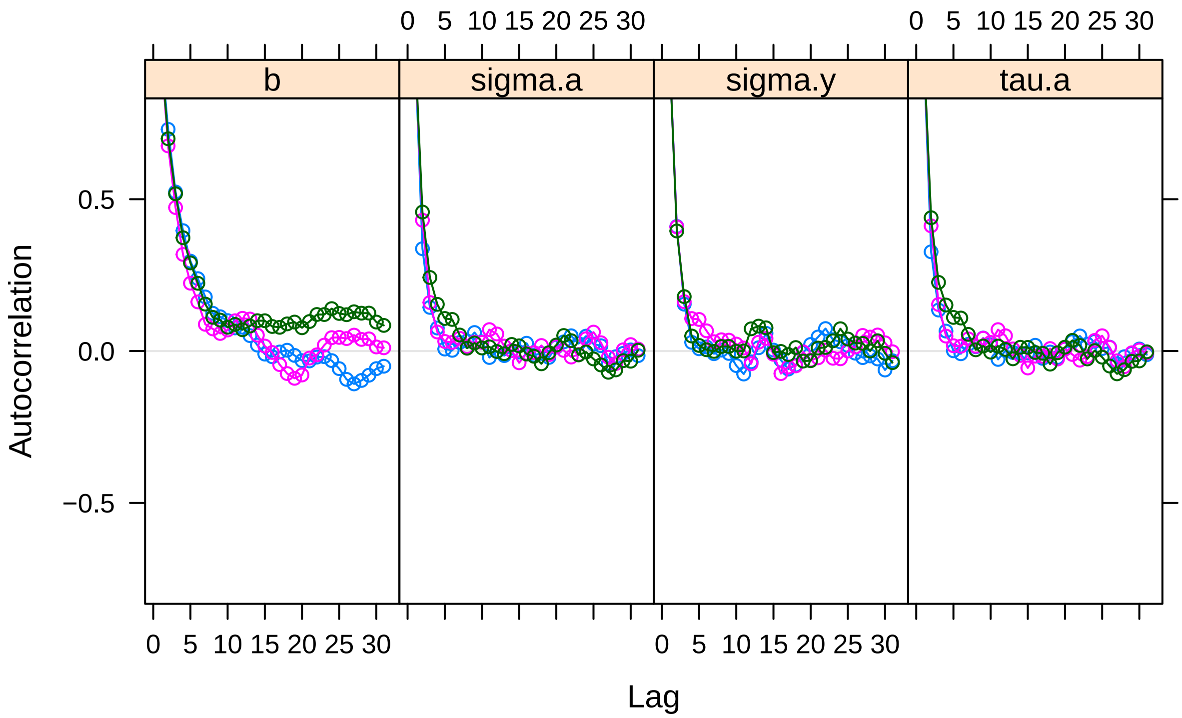
```
##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean       SD  Naive SE Time-series SE
## b       -0.27784 0.058101 1.061e-03      2.597e-03
## sigma.a 12.03230 0.966203 1.764e-02      2.740e-02
## sigma.y  6.60342 0.258334 4.717e-03      7.246e-03
## tau.a    0.00704 0.001118 2.041e-05      3.104e-05
##
## 2. Quantiles for each variable:
##
##              2.5%      25%       50%       75%     97.5%
## b       -0.392220 -0.31657 -0.277449 -0.237923 -0.165210
```

```
## sigma.a 10.313572 11.36808 11.953358 12.659123 14.067227
## sigma.y  6.126944  6.42748  6.590448  6.777528  7.131008
## tau.a    0.005053  0.00624  0.006999  0.007738  0.009401
```

```r
par(mar=rep(2, 4))
plot(im.samp)
```




```r
acfplot(im.samp)
```




```r
gelman.diag(im.samp)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## b              1.00       1.01
## sigma.a        1.01       1.02
## sigma.y        1.00       1.01
## tau.a          1.01       1.03
##
## Multivariate psrf
##
## 1.01
```

Add more iterations and include all parameters of interest

```r
im.samp.full <- coda.samples(im.model, c("a", "b", "tau.a", "sigma.a", "sigma.y",
"y.hat"), n.iter=5000, thin=5)
```

Look at the results

```r
im.summ <- summary(im.samp.full)[1]
im.stats <- as.data.frame(im.summ$statistics)
head(im.stats, 100)
```

```
##              Mean       SD   Naive SE Time-series SE
## a[1]    13.985637 2.685789 0.04903557     0.04822797
## a[2]    22.579105 2.486671 0.04540020     0.04539557
## a[3]    15.923610 3.185657 0.05816187     0.05814462
```

```
## a[4]   33.027715 3.158637 0.05766857     0.05768218
## a[5]    5.492767 3.230042 0.05897223     0.06081593
## a[6]   18.075757 4.340047 0.07923805     0.07756947
## a[7]   29.488743 2.454713 0.04481673     0.04482364
## a[8]    7.636434 4.279339 0.07812969     0.07905507
## a[9]   18.430103 2.516330 0.04594170     0.04595235
## a[10]  38.971508 2.643124 0.04825662     0.04825682
## a[11]  24.752304 2.518929 0.04598915     0.04374090
## a[12]  31.200272 2.515268 0.04592231     0.04592765
## a[13]  19.633690 4.374991 0.07987605     0.07989864
## a[14]  19.472783 2.925463 0.05341140     0.05680735
## a[15]  28.239154 2.872265 0.05244014     0.05269412
## a[16]   7.818127 4.423229 0.08075675     0.08078024
## a[17]  45.078438 2.731022 0.04986141     0.04987175
## a[18]  36.236255 3.597868 0.06568778     0.06365126
## a[19]  31.733133 3.640764 0.06647095     0.06668649
## a[20]  11.331969 4.340980 0.07925508     0.08133016
## a[21]  14.907715 5.859887 0.10698642     0.10535139
## a[22]  21.082813 3.131665 0.05717612     0.05617248
## a[23]  27.581184 2.815806 0.05140935     0.05141485
## a[24]  22.930401 4.356948 0.07954662     0.07792195
## a[25]   4.184149 2.939960 0.05367608     0.05261158
## a[26]  15.299371 3.222604 0.05883644     0.05976553
## a[27]  19.075497 3.207018 0.05855188     0.05854734
## a[28]  23.409760 3.248155 0.05930293     0.05882850
## a[29]  15.607281 2.681384 0.04895516     0.04978525
## a[30]  44.285363 2.528744 0.04616833     0.04613889
## a[31]  31.993277 2.493366 0.04552243     0.04436139
## a[32]  43.305745 2.538054 0.04633832     0.04632742
## a[33]  31.747260 2.549846 0.04655360     0.04865174
## a[34]  49.691355 5.802383 0.10593653     0.10596934
## a[35]  24.590890 2.520195 0.04601226     0.04535563
## a[36]  31.173688 2.576842 0.04704649     0.04788363
## a[37]  49.440314 2.585346 0.04720174     0.04720310
## a[38]   9.983150 2.727249 0.04979253     0.04876361
## a[39]  13.343008 2.441016 0.04456665     0.04544287
## a[40]  36.365403 2.469417 0.04508518     0.04509942
## a[41]  30.935581 2.728699 0.04981899     0.04961367
## a[42]  31.215218 2.690753 0.04912620     0.04914111
## a[43]  37.064756 2.956693 0.05398158     0.05876809
## a[44]  28.978894 3.602197 0.06576681     0.06820304
## a[45]  32.147317 2.905978 0.05305566     0.05400696
## a[46]  20.491019 3.224052 0.05886286     0.05876408
## a[47]  31.999143 3.183409 0.05812083     0.06013362
## a[48]  21.840721 5.793072 0.10576655     0.10578614
## a[49]  49.803930 3.637212 0.06640610     0.06641020
## a[50]  32.096781 3.738340 0.06825243     0.07036493
## a[51]  38.993075 3.688488 0.06734227     0.07783009
## a[52]  39.104374 4.368112 0.07975045     0.07832187
## a[53]  39.590454 3.695807 0.06747589     0.06746693
## a[54]  36.203257 2.920395 0.05331888     0.05646997
## a[55]  28.230116 4.397969 0.08029556     0.07870536
## a[56]  25.850410 4.391211 0.08017218     0.09084116
## a[57]  27.338072 3.217083 0.05873563     0.05873979
```

```
## a[58]   22.833687 3.620687 0.06610441     0.06670401
## a[59]   52.709165 3.204484 0.05850561     0.05850242
## a[60]   46.918882 2.906608 0.05306716     0.05306429
## a[61]   20.144795 2.889162 0.05274863     0.05266719
## a[62]   31.038587 4.361942 0.07963781     0.07842952
## a[63]   24.482818 3.140178 0.05733154     0.05732912
## a[64]   33.097876 2.494439 0.04554201     0.04526313
## a[65]   12.272876 4.381798 0.08000032     0.07992231
## a[66]   21.418869 2.866472 0.05233439     0.04785864
## a[67]   41.269193 5.848863 0.10678513     0.10681226
## a[68]   27.848533 2.999133 0.05475643     0.05291283
## a[69]    5.793798 4.376643 0.07990620     0.07989329
## a[70]   16.008739 2.863147 0.05227368     0.05226226
## a[71]   33.907490 2.956137 0.05397143     0.05696438
## a[72]   31.967600 3.612342 0.06595204     0.06523131
## a[73]   22.707033 3.222015 0.05882567     0.05790348
## a[74]   42.085480 2.948144 0.05382550     0.05369850
## a[75]   29.297585 3.621140 0.06611266     0.06612597
## a[76]   29.076702 2.497506 0.04559801     0.04724332
## a[77]   55.170056 2.720909 0.04967678     0.04757943
## a[78]   45.189503 3.663806 0.06689164     0.06232985
## a[79]   22.590349 4.328242 0.07902252     0.07902147
## a[80]   24.298991 5.963214 0.10887290     0.10889745
## a[81]   27.658258 2.683402 0.04899199     0.05062838
## a[82]    9.770331 2.932067 0.05353197     0.05570498
## a[83]   35.094013 2.739397 0.05001431     0.04925510
## a[84]   31.697719 2.871807 0.05243178     0.05158262
## a[85]    6.235850 4.305545 0.07860814     0.08002210
## a[86]   16.961399 4.403342 0.08039365     0.08041350
## a[87]   23.258416 3.550230 0.06481803     0.06766846
## a[88]   22.283135 2.541118 0.04639426     0.04621257
## a[89]   13.892146 2.498937 0.04562414     0.04569068
## a[90]    5.344756 2.688545 0.04908590     0.05216165
## a[91]   41.191199 3.668466 0.06697673     0.06339186
## a[92]   19.943439 3.177063 0.05800498     0.05734223
## a[93]   21.599137 3.187736 0.05819983     0.05820772
## a[94]   36.113998 2.915892 0.05323665     0.05085744
## a[95]   14.582691 3.175472 0.05797593     0.05797634
## a[96]   38.461166 2.727566 0.04979831     0.05104973
## a[97]   24.744516 3.250060 0.05933770     0.05935585
## a[98]   36.137976 4.415719 0.08061964     0.08034587
## a[99]   38.491756 3.225575 0.05889068     0.05890817
## a[100] 30.012492 3.253519 0.05940087     0.06277160
```

```r
y.hat.rows <- grep("y.hat", rownames(im.stats))
y.hat <- im.stats$Mean[y.hat.rows] # To get fitted values of the model,
# look at the stats and pull out the rows corresponding to y.hat
resids <- im$immune.level - y.hat
# for convenience, we can then put those values into our data frame and examine them
imres <- cbind(im, y.hat = y.hat, resid = resids)
```

Some summary statistics:

The means and standard deviations of:

- slope = b
- intercept = a
- group-level (i.e. among-patient) variance = sigma.a
- individual-level (within-patient; individual-data-point-level) variance = sigma.y
- y.hat?

```
woo1 = summary(im.samp.full)[1]
woo1$statistics[,c('Mean', 'SD')]
```

```
##                      Mean         SD
## a[1]        13.985636851 2.68578890
## a[2]        22.579104829 2.48667118
## a[3]        15.923610042 3.18565687
## a[4]        33.027714516 3.15863741
## a[5]         5.492767287 3.23004187
## a[6]        18.075756557 4.34004693
## a[7]        29.488743271 2.45471325
## a[8]         7.636434406 4.27933927
## a[9]        18.430102771 2.51633041
## a[10]       38.971507723 2.64312377
## a[11]       24.752303845 2.51892936
## a[12]       31.200272003 2.51526848
## a[13]       19.633689780 4.37499146
## a[14]       19.472783411 2.92546306
## a[15]       28.239153690 2.87226491
## a[16]        7.818127069 4.42322947
## a[17]       45.078437719 2.73102208
## a[18]       36.236255155 3.59786787
## a[19]       31.733132990 3.64076409
## a[20]       11.331968691 4.34097977
## a[21]       14.907714960 5.85988743
## a[22]       21.082813081 3.13166516
## a[23]       27.581183523 2.81580608
## a[24]       22.930401147 4.35694803
## a[25]        4.184149329 2.93995992
## a[26]       15.299371416 3.22260442
## a[27]       19.075496704 3.20701850
## a[28]       23.409760212 3.24815503
## a[29]       15.607281207 2.68138438
## a[30]       44.285363064 2.52874366
## a[31]       31.993277349 2.49336599
## a[32]       43.305745369 2.53805422
## a[33]       31.747260046 2.54984581
## a[34]       49.691355386 5.80238272
## a[35]       24.590890388 2.52019528
## a[36]       31.173687815 2.57684246
## a[37]       49.440314090 2.58534590
## a[38]        9.983149664 2.72724911
## a[39]       13.343008395 2.44101591
## a[40]       36.365403037 2.46941692
## a[41]       30.935580985 2.72869862
```

```
## a[42]      31.215218451 2.69075299
## a[43]      37.064756024 2.95669299
## a[44]      28.978894196 3.60219662
## a[45]      32.147317409 2.90597838
## a[46]      20.491018928 3.22405185
## a[47]      31.999143037 3.18340908
## a[48]      21.840720645 5.79307231
## a[49]      49.803930341 3.63721187
## a[50]      32.096780524 3.73833979
## a[51]      38.993075480 3.68848785
## a[52]      39.104374493 4.36811205
## a[53]      39.590454126 3.69580691
## a[54]      36.203256806 2.92039523
## a[55]      28.230116319 4.39796909
## a[56]      25.850410394 4.39121114
## a[57]      27.338071944 3.21708310
## a[58]      22.833686726 3.62068738
## a[59]      52.709165392 3.20448412
## a[60]      46.918881751 2.90660797
## a[61]      20.144795402 2.88916161
## a[62]      31.038586500 4.36194224
## a[63]      24.482817892 3.14017765
## a[64]      33.097875892 2.49443885
## a[65]      12.272876432 4.38179817
## a[66]      21.418868747 2.86647248
## a[67]      41.269193444 5.84886262
## a[68]      27.848533340 2.99913306
## a[69]       5.793798236 4.37664296
## a[70]      16.008739345 2.86314729
## a[71]      33.907489728 2.95613703
## a[72]      31.967600058 3.61234193
## a[73]      22.707032913 3.22201476
## a[74]      42.085479970 2.94814387
## a[75]      29.297585314 3.62113962
## a[76]      29.076701885 2.49750606
## a[77]      55.170055615 2.72090925
## a[78]      45.189503273 3.66380618
## a[79]      22.590348742 4.32824167
## a[80]      24.298991385 5.96321437
## a[81]      27.658258312 2.68340171
## a[82]       9.770330608 2.93206683
## a[83]      35.094013155 2.73939658
## a[84]      31.697719420 2.87180703
## a[85]       6.235850134 4.30554513
## a[86]      16.961399358 4.40334160
## a[87]      23.258416418 3.55022958
## a[88]      22.283134825 2.54111802
## a[89]      13.892146067 2.49893729
## a[90]       5.344755660 2.68854550
## a[91]      41.191199022 3.66846645
## a[92]      19.943438764 3.17706336
## a[93]      21.599136998 3.18773575
## a[94]      36.113997567 2.91589162
## a[95]      14.582690757 3.17547241
```

```
## a[96]       38.461165569 2.72756599
## a[97]       24.744515604 3.25005974
## a[98]       36.137975912 4.41571946
## a[99]       38.491755509 3.22557516
## a[100]      30.012492387 3.25351942
## b           -0.278612462 0.05880966
## sigma.a     11.975921671 0.95114014
## sigma.y      6.587527802 0.25755101
## tau.a        0.007104052 0.00112348
## y.hat[1]    13.707024389 2.67806637
## y.hat[2]    12.871187002 2.66256922
## y.hat[3]    12.035349615 2.65871484
## y.hat[4]    11.199512227 2.66655371
## y.hat[5]    10.363674840 2.68598346
## y.hat[6]     9.527837453 2.71675541
## y.hat[7]    22.300492367 2.47647375
## y.hat[8]    21.464654980 2.45409681
## y.hat[9]    20.628817593 2.44428199
## y.hat[10]   19.792980205 2.44718044
## y.hat[11]   18.957142818 2.46274728
## y.hat[12]   18.121305431 2.49074498
## y.hat[13]   17.285468044 2.53076101
## y.hat[14]   15.644997580 3.18073760
## y.hat[15]   14.809160192 3.17248175
## y.hat[16]   13.973322805 3.17402640
## y.hat[17]   13.137485418 3.18535730
## y.hat[18]   32.749102054 3.15149354
## y.hat[19]   31.913264667 3.13658702
## y.hat[20]   31.077427280 3.13156496
## y.hat[21]   28.569915118 3.17581301
## y.hat[22]    5.214154824 3.22031497
## y.hat[23]    4.378317437 3.19745298
## y.hat[24]    1.870805275 3.18701473
## y.hat[25]    1.034967888 3.20302317
## y.hat[26]   17.797144094 4.33813719
## y.hat[27]   16.961306707 4.33719010
## y.hat[28]   29.210130809 2.44048080
## y.hat[29]   28.374293421 2.40591719
## y.hat[30]   27.538456034 2.38394272
## y.hat[31]   26.702618647 2.37490687
## y.hat[32]   25.866781260 2.37895707
## y.hat[33]   25.030943872 2.39602696
## y.hat[34]   24.195106485 2.42584172
## y.hat[35]    7.357821944 4.28019009
## y.hat[36]    6.521984557 4.28758417
## y.hat[37]   18.151490309 2.50369903
## y.hat[38]   17.315652921 2.47381932
## y.hat[39]   16.479815534 2.45627962
## y.hat[40]   15.643978147 2.45134483
## y.hat[41]   14.808140760 2.45909084
## y.hat[42]   13.972303372 2.47939880
## y.hat[43]   13.136465985 2.51196406
## y.hat[44]   38.692895261 2.63413045
## y.hat[45]   37.857057874 2.61491222
```

```
## y.hat[46]  37.021220487 2.60751664
## y.hat[47]  36.185383099 2.61204411
## y.hat[48]  35.349545712 2.62843304
## y.hat[49]  33.677870938 2.69577354
## y.hat[50]  24.473691383 2.50372187
## y.hat[51]  23.637853996 2.46596438
## y.hat[52]  22.802016609 2.44040813
## y.hat[53]  21.966179221 2.42743850
## y.hat[54]  21.130341834 2.42725728
## y.hat[55]  20.294504447 2.43986731
## y.hat[56]  19.458667060 2.46507230
## y.hat[57]  30.921659541 2.50082513
## y.hat[58]  30.085822153 2.46541715
## y.hat[59]  29.249984766 2.44227179
## y.hat[60]  28.414147379 2.43173922
## y.hat[61]  27.578309992 2.43398318
## y.hat[62]  26.742472604 2.44896855
## y.hat[63]  25.906635217 2.47646405
## y.hat[64]  19.355077317 4.37122958
## y.hat[65]  16.847565155 4.37296237
## y.hat[66]  19.194170948 2.91639880
## y.hat[67]  18.358333561 2.89620933
## y.hat[68]  17.522496174 2.88668138
## y.hat[69]  16.686658787 2.88792047
## y.hat[70]  15.850821399 2.89991280
## y.hat[71]  27.960541227 2.86351468
## y.hat[72]  27.124703840 2.84440698
## y.hat[73]  26.288866453 2.83616647
## y.hat[74]  25.453029066 2.83888776
## y.hat[75]  24.617191678 2.85253949
## y.hat[76]   7.539514607 4.42058682
## y.hat[77]   6.703677219 4.41734957
## y.hat[78]  44.799825256 2.71801840
## y.hat[79]  43.963987869 2.68636451
## y.hat[80]  43.128150482 2.66603430
## y.hat[81]  42.292313095 2.65728768
## y.hat[82]  41.456475708 2.66023893
## y.hat[83]  39.784800933 2.70092964
## y.hat[84]  35.957642693 3.59181989
## y.hat[85]  34.285967918 3.57568623
## y.hat[86]  31.778455756 3.61648760
## y.hat[87]  31.454520527 3.63235142
## y.hat[88]  28.947008365 3.59924601
## y.hat[89]  28.111170978 3.60545127
## y.hat[90]  11.053356229 4.33952630
## y.hat[91]  10.217518842 4.33994713
## y.hat[92]  14.629102497 5.85854718
## y.hat[93]  20.804200619 3.12119372
## y.hat[94]  19.968363232 3.09627580
## y.hat[95]  17.460851070 3.08150461
## y.hat[96]  15.789176296 3.12187556
## y.hat[97]  27.302571061 2.80730729
## y.hat[98]  26.466733673 2.78910525
## y.hat[99]  25.630896286 2.78199503
```

```
## y.hat[100] 24.795058899 2.78606155
## y.hat[101] 23.123384125 2.82739938
## y.hat[102] 22.651788684 4.35243084
## y.hat[103] 19.308439135 4.36017081
## y.hat[104]  3.905536866 2.93127214
## y.hat[105]  3.069699479 2.91218737
## y.hat[106]  2.233862092 2.90371634
## y.hat[107]  1.398024704 2.90595186
## y.hat[108] -0.273650070 2.94232810
## y.hat[109] 15.020758953 3.21546566
## y.hat[110] 14.184921566 3.20044413
## y.hat[111] 13.349084179 3.19510880
## y.hat[112] 10.841572017 3.23726416
## y.hat[113] 18.796884241 3.19886766
## y.hat[114] 17.961046854 3.18082021
## y.hat[115] 17.125209467 3.17249657
## y.hat[116] 14.617697305 3.20618347
## y.hat[117] 23.131147749 3.23811383
## y.hat[118] 21.459472975 3.19997845
## y.hat[119] 19.787798200 3.20052502
## y.hat[120] 18.951960813 3.21535241
## y.hat[121] 15.328668744 2.67166281
## y.hat[122] 14.492831357 2.65012547
## y.hat[123] 13.656993970 2.64022768
## y.hat[124] 12.821156583 2.64210026
## y.hat[125] 11.985319195 2.65571831
## y.hat[126] 11.149481808 2.68090284
## y.hat[127] 44.006750602 2.51627405
## y.hat[128] 43.170913214 2.48684737
## y.hat[129] 42.335075827 2.46970422
## y.hat[130] 41.499238440 2.46510088
## y.hat[131] 40.663401053 2.47310739
## y.hat[132] 39.827563665 2.49360229
## y.hat[133] 38.991726278 2.52628164
## y.hat[134] 31.714664887 2.48101889
## y.hat[135] 30.878827500 2.45208072
## y.hat[136] 30.042990112 2.43561068
## y.hat[137] 29.207152725 2.43186210
## y.hat[138] 28.371315338 2.44089359
## y.hat[139] 27.535477951 2.46256454
## y.hat[140] 26.699640564 2.49654583
## y.hat[141] 43.027132906 2.52392148
## y.hat[142] 42.191295519 2.48939024
## y.hat[143] 41.355458132 2.46702294
## y.hat[144] 40.519620745 2.45715181
## y.hat[145] 39.683783357 2.45992728
## y.hat[146] 38.847945970 2.47530681
## y.hat[147] 38.012108583 2.50305807
## y.hat[148] 31.468647584 2.53687979
## y.hat[149] 30.632810196 2.50587276
## y.hat[150] 29.796972809 2.48702470
## y.hat[151] 28.961135422 2.48061278
## y.hat[152] 28.125298035 2.48673321
## y.hat[153] 27.289460647 2.50529413
```

```
## y.hat[154] 26.453623260 2.53602242
## y.hat[155] 49.412742923 5.80093891
## y.hat[156] 24.312277925 2.50659817
## y.hat[157] 23.476440538 2.47375985
## y.hat[158] 22.640603151 2.45320108
## y.hat[159] 21.804765764 2.44523162
## y.hat[160] 20.968928376 2.44997432
## y.hat[161] 20.133090989 2.46735587
## y.hat[162] 19.297253602 2.49711237
## y.hat[163] 30.895075352 2.56226303
## y.hat[164] 30.059237965 2.52624606
## y.hat[165] 29.223400578 2.50217922
## y.hat[166] 28.387563191 2.49040898
## y.hat[167] 27.551725803 2.49110965
## y.hat[168] 26.715888416 2.50427077
## y.hat[169] 25.880051029 2.52969785
## y.hat[170] 49.161701628 2.57209732
## y.hat[171] 48.325864241 2.54011833
## y.hat[172] 47.490026854 2.52011349
## y.hat[173] 46.654189466 2.51236885
## y.hat[174] 45.818352079 2.51699759
## y.hat[175] 44.982514692 2.53393189
## y.hat[176] 44.146677305 2.56292784
## y.hat[177]  9.704537202 2.71773330
## y.hat[178]  8.868699814 2.69668997
## y.hat[179]  8.032862427 2.68709020
## y.hat[180]  7.197025040 2.68905653
## y.hat[181]  6.361187653 2.70256372
## y.hat[182]  5.525350265 2.72744032
## y.hat[183] 13.064395933 2.42895127
## y.hat[184] 12.228558546 2.40105060
## y.hat[185] 11.392721158 2.38590404
## y.hat[186] 10.556883771 2.38375472
## y.hat[187]  9.721046384 2.39463763
## y.hat[188]  8.885208997 2.41837684
## y.hat[189]  8.049371610 2.45459938
## y.hat[190] 36.086790574 2.45761973
## y.hat[191] 35.250953187 2.43043663
## y.hat[192] 34.415115800 2.41586506
## y.hat[193] 33.579278413 2.41413342
## y.hat[194] 32.743441025 2.42526920
## y.hat[195] 31.907603638 2.44909689
## y.hat[196] 31.071766251 2.48525147
## y.hat[197] 30.656968522 2.71738365
## y.hat[198] 29.821131135 2.69087533
## y.hat[199] 28.985293748 2.67576168
## y.hat[200] 28.149456360 2.67223604
## y.hat[201] 27.313618973 2.68034413
## y.hat[202] 26.477781586 2.69998116
## y.hat[203] 30.936605988 2.68098795
## y.hat[204] 30.100768601 2.65929191
## y.hat[205] 29.264931214 2.64919326
## y.hat[206] 28.429093827 2.65082454
## y.hat[207] 27.593256439 2.66416421
```

```
## y.hat[208] 26.757419052 2.68903803
## y.hat[209] 36.786143562 2.94683419
## y.hat[210] 35.950306175 2.92416306
## y.hat[211] 35.114468787 2.91202367
## y.hat[212] 33.442794013 2.91975160
## y.hat[213] 32.606956626 2.93953478
## y.hat[214] 28.700281733 3.59680369
## y.hat[215] 27.864444346 3.58636705
## y.hat[216] 27.028606959 3.58459409
## y.hat[217] 31.868704947 2.89748337
## y.hat[218] 31.032867559 2.87906430
## y.hat[219] 30.197030172 2.87138764
## y.hat[220] 29.361192785 2.87453946
## y.hat[221] 28.525355398 2.88848430
## y.hat[222] 20.212406466 3.21917094
## y.hat[223] 19.376569078 3.21095284
## y.hat[224] 18.540731691 3.21241796
## y.hat[225] 17.704894304 3.22355308
## y.hat[226] 31.720530575 3.17832753
## y.hat[227] 30.884693188 3.16958773
## y.hat[228] 30.048855801 3.17065628
## y.hat[229] 29.213018413 3.18152329
## y.hat[230] 21.562108183 5.79288550
## y.hat[231] 49.525317878 3.63291949
## y.hat[232] 48.689480491 3.62573973
## y.hat[233] 47.853643104 3.62713764
## y.hat[234] 31.818168061 3.73404069
## y.hat[235] 30.982330674 3.72668610
## y.hat[236] 30.146493287 3.72767663
## y.hat[237] 38.714463018 3.68243697
## y.hat[238] 37.878625631 3.66988325
## y.hat[239] 37.042788243 3.66578757
## y.hat[240] 38.825762030 4.36608573
## y.hat[241] 37.989924643 4.36475800
## y.hat[242] 39.311841664 3.69199307
## y.hat[243] 38.476004277 3.68616171
## y.hat[244] 36.804329502 3.69979754
## y.hat[245] 35.924644344 2.90812917
## y.hat[246] 35.088806957 2.87823542
## y.hat[247] 34.252969569 2.85893641
## y.hat[248] 32.581294795 2.85286469
## y.hat[249] 30.909620020 2.89018580
## y.hat[250] 27.951503856 4.39724276
## y.hat[251] 26.279829082 4.40938483
## y.hat[252] 25.571797932 4.39047789
## y.hat[253] 24.735960545 4.39300373
## y.hat[254] 27.059459482 3.21151715
## y.hat[255] 26.223622095 3.20124998
## y.hat[256] 25.387784707 3.20068975
## y.hat[257] 24.551947320 3.20984153
## y.hat[258] 22.555074264 3.61778355
## y.hat[259] 21.719236876 3.61480330
## y.hat[260] 20.883399489 3.62042848
## y.hat[261] 52.430552930 3.19942892
```

```
## y.hat[262] 51.594715543 3.19072551
## y.hat[263] 50.758878156 3.19176558
## y.hat[264] 49.923040768 3.20253964
## y.hat[265] 46.640269289 2.89634183
## y.hat[266] 44.968594514 2.85947772
## y.hat[267] 44.132757127 2.85725441
## y.hat[268] 43.296919740 2.86591121
## y.hat[269] 41.625244965 2.91535569
## y.hat[270] 19.866182940 2.88067691
## y.hat[271] 18.194508165 2.85479031
## y.hat[272] 17.358670778 2.85814160
## y.hat[273] 16.522833391 2.87234633
## y.hat[274] 15.686996004 2.89724487
## y.hat[275] 28.252461876 4.32775652
## y.hat[276] 27.416624489 4.33304072
## y.hat[277] 24.204205430 3.13241766
## y.hat[278] 23.368368042 3.11568892
## y.hat[279] 22.532530655 3.10889834
## y.hat[280] 20.025018493 3.14832803
## y.hat[281] 32.819263429 2.48260862
## y.hat[282] 31.983426042 2.45524126
## y.hat[283] 31.147588655 2.44035410
## y.hat[284] 30.311751268 2.43817575
## y.hat[285] 29.475913881 2.44874012
## y.hat[286] 28.640076493 2.47188385
## y.hat[287] 27.804239106 2.50725860
## y.hat[288] 11.994263970 4.37643498
## y.hat[289]  9.486751808 4.36368038
## y.hat[290] 21.140256285 2.85656115
## y.hat[291] 20.304418898 2.83395060
## y.hat[292] 18.632744123 2.82147264
## y.hat[293] 17.796906736 2.83174952
## y.hat[294] 16.961069349 2.85292082
## y.hat[295] 40.990580981 5.84668500
## y.hat[296] 27.569920878 2.98993533
## y.hat[297] 26.734083490 2.96916803
## y.hat[298] 25.898246103 2.95879345
## y.hat[299] 25.062408716 2.95892091
## y.hat[300] 22.554896554 3.02175476
## y.hat[301]  5.515185774 4.37416840
## y.hat[302]  3.843511000 4.37592081
## y.hat[303] 15.730126882 2.85542325
## y.hat[304] 14.058452108 2.83437460
## y.hat[305] 13.222614721 2.84027808
## y.hat[306] 12.386777333 2.85708491
## y.hat[307] 11.550939946 2.88460451
## y.hat[308] 33.628877265 2.94812190
## y.hat[309] 32.793039878 2.93103318
## y.hat[310] 31.957202491 2.92450724
## y.hat[311] 31.121365104 2.92861470
## y.hat[312] 29.449690329 2.96843900
## y.hat[313] 31.688987596 3.60546656
## y.hat[314] 30.853150208 3.59054547
## y.hat[315] 29.181475434 3.58664678
```

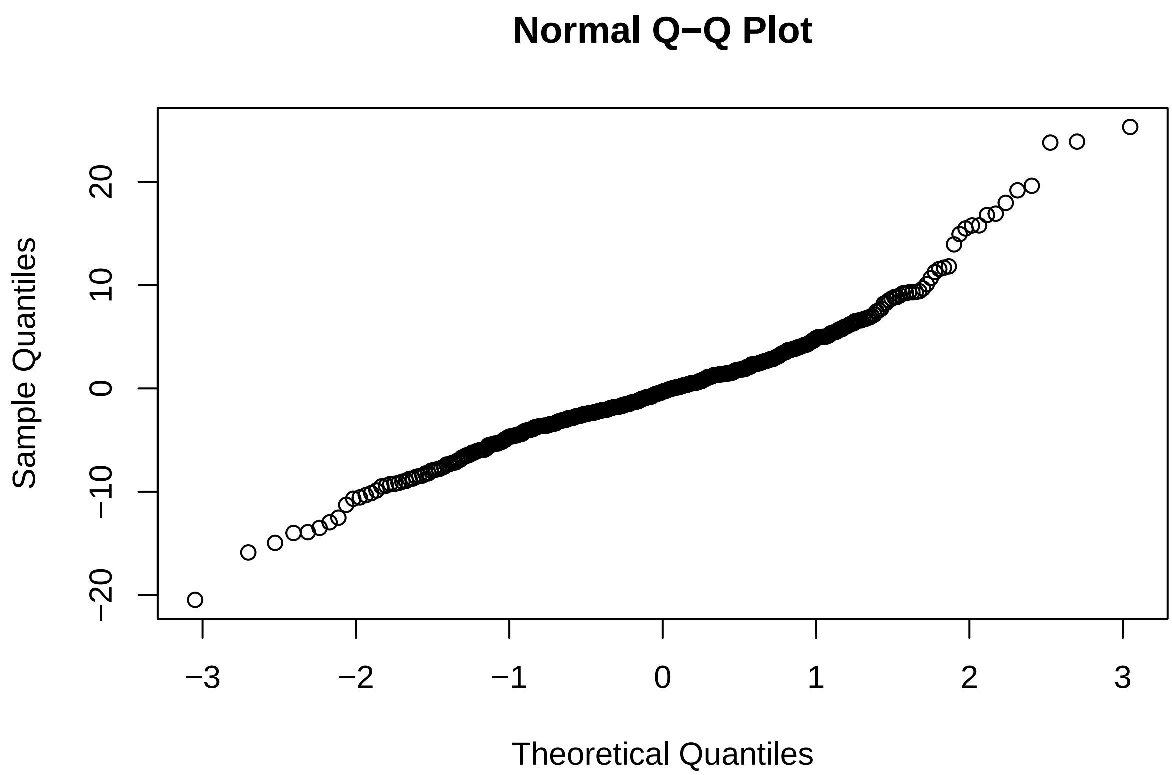
```
## y.hat[316] 22.428420451 3.21308839
## y.hat[317] 21.592583063 3.19266558
## y.hat[318] 19.085070902 3.18969737
## y.hat[319] 18.249233514 3.20817133
## y.hat[320] 41.806867508 2.93848419
## y.hat[321] 40.135192734 2.90495637
## y.hat[322] 39.299355346 2.90416842
## y.hat[323] 38.463517959 2.91408154
## y.hat[324] 37.627680572 2.93458730
## y.hat[325] 29.018972852 3.61892433
## y.hat[326] 28.183135465 3.61801047
## y.hat[327] 27.347298078 3.62569173
## y.hat[328] 28.798089423 2.48370440
## y.hat[329] 27.962252036 2.45031500
## y.hat[330] 27.126414649 2.42931142
## y.hat[331] 26.290577261 2.42101606
## y.hat[332] 25.454739874 2.42555929
## y.hat[333] 24.618902487 2.44286950
## y.hat[334] 23.783065100 2.47267855
## y.hat[335] 54.891443153 2.70843426
## y.hat[336] 54.055605765 2.67841856
## y.hat[337] 53.219768378 2.65979138
## y.hat[338] 51.548093604 2.65751421
## y.hat[339] 50.712256216 2.67389399
## y.hat[340] 49.876418829 2.70171996
## y.hat[341] 44.910890811 3.65752683
## y.hat[342] 44.075053424 3.64432240
## y.hat[343] 39.895866488 3.70578162
## y.hat[344] 22.311736280 4.32570358
## y.hat[345] 20.640061505 4.32726069
## y.hat[346] 24.020378923 5.96148047
## y.hat[347] 27.379645849 2.67376431
## y.hat[348] 26.543808462 2.65247642
## y.hat[349] 25.707971075 2.64282067
## y.hat[350] 24.872133688 2.64492446
## y.hat[351] 24.036296301 2.65875987
## y.hat[352] 23.200458913 2.68414551
## y.hat[353]  9.491718145 2.92036955
## y.hat[354]  8.655880758 2.89217711
## y.hat[355]  6.148368597 2.87169027
## y.hat[356]  5.312531209 2.88647754
## y.hat[357]  4.476693822 2.91189878
## y.hat[358] 34.815400693 2.72671135
## y.hat[359] 33.143725918 2.67659895
## y.hat[360] 32.307888531 2.66874073
## y.hat[361] 31.472051144 2.67253186
## y.hat[362] 30.636213757 2.68792307
## y.hat[363] 29.800376369 2.71471704
## y.hat[364] 31.419106958 2.86487740
## y.hat[365] 30.583269570 2.85127450
## y.hat[366] 29.747432183 2.84855479
## y.hat[367] 28.075757409 2.87576489
## y.hat[368] 26.404082634 2.94530142
## y.hat[369]  5.957237671 4.30182555
```

```
## y.hat[370]  4.285562897 4.29637856
## y.hat[371] 16.682786896 4.40008659
## y.hat[372] 15.011112121 4.39705475
## y.hat[373] 22.979803956 3.54673508
## y.hat[374] 22.143966569 3.54209423
## y.hat[375] 21.308129181 3.54623571
## y.hat[376] 22.004522363 2.52868534
## y.hat[377] 21.168684975 2.49933164
## y.hat[378] 20.332847588 2.48220107
## y.hat[379] 19.497010201 2.47754717
## y.hat[380] 18.661172814 2.48544004
## y.hat[381] 17.825335427 2.50576112
## y.hat[382] 16.989498039 2.53821191
## y.hat[383] 13.613533604 2.48579044
## y.hat[384] 12.777696217 2.45439539
## y.hat[385] 11.941858830 2.43540832
## y.hat[386] 11.106021443 2.42912021
## y.hat[387] 10.270184055 2.43562943
## y.hat[388]  9.434346668 2.45483416
## y.hat[389]  8.598509281 2.48644027
## y.hat[390]  5.066143197 2.67840928
## y.hat[391]  4.230305810 2.65559360
## y.hat[392]  3.394468423 2.64437758
## y.hat[393]  2.558631036 2.64490881
## y.hat[394]  1.722793649 2.65718023
## y.hat[395]  0.886956261 2.68103064
## y.hat[396] 40.912586560 3.66274120
## y.hat[397] 37.569237011 3.66763346
## y.hat[398] 36.733399624 3.69000496
## y.hat[399] 19.664826302 3.16597946
## y.hat[400] 17.157314140 3.11479483
## y.hat[401] 16.321476753 3.11757852
## y.hat[402] 15.485639366 3.13031935
## y.hat[403] 21.320524535 3.18051294
## y.hat[404] 20.484687148 3.16530810
## y.hat[405] 19.648849761 3.15989598
## y.hat[406] 17.977174987 3.17855944
## y.hat[407] 35.835385104 2.90419005
## y.hat[408] 34.999547717 2.87602337
## y.hat[409] 34.163710330 2.85848831
## y.hat[410] 32.492035555 2.85597767
## y.hat[411] 30.820360781 2.89677044
## y.hat[412] 14.304078295 3.16829866
## y.hat[413] 12.632403520 3.14806683
## y.hat[414] 11.796566133 3.15274650
## y.hat[415] 10.960728746 3.16726232
## y.hat[416] 38.182553106 2.71624040
## y.hat[417] 37.346715719 2.68970295
## y.hat[418] 35.675040944 2.67101949
## y.hat[419] 34.839203557 2.67911333
## y.hat[420] 34.003366170 2.69874150
## y.hat[421] 33.167528783 2.72965521
## y.hat[422] 24.465903141 3.24045497
## y.hat[423] 21.958390979 3.20168097
```

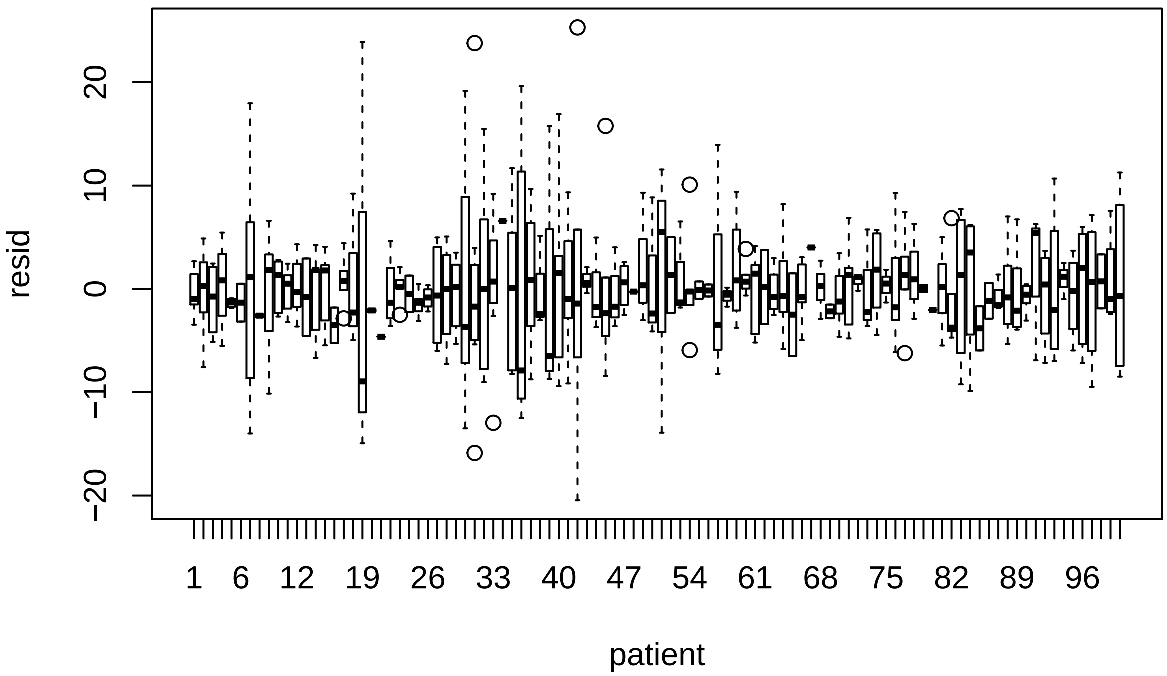
```
## y.hat[424] 21.122553592 3.20811585
## y.hat[425] 20.286716205 3.22420656
## y.hat[426] 35.859363450 4.41131127
## y.hat[427] 34.187688676 4.40130209
## y.hat[428] 38.213143047 3.21555234
## y.hat[429] 37.377305660 3.19180278
## y.hat[430] 34.033956111 3.19397261
## y.hat[431] 33.198118724 3.21878253
## y.hat[432] 29.733879924 3.24688991
## y.hat[433] 28.898042537 3.23334401
## y.hat[434] 27.226367763 3.23507580
## y.hat[435] 26.390530375 3.25033813
```

See how well the model ran

```
qqnorm(imres$resid)
```




```
boxplot(resid~patient, data=imres)
```




```
# Now that random intercepts for patient are included in the model,
# does it look like the residuals differ strongly by patient?
```

Get model DIC

```
im.DIC <- dic.samples(im.model, n.iter=5000, thin=5)
im.DIC
```

```
## Mean deviance:  2874
## penalty 94.57
## Penalized deviance: 2968
```