Untitled

2024-05-16

```
aggregated_data <- read.csv("AggregatedData1.csv")
daily_data <- read.csv("Daily_AggregatedData1.csv")
```

```
cluster1_day <- daily_data[, -c(3,4,5,6,7)]

y <- cluster1_day$Cluster.1
x1 <- cluster1_day$toy        # time of year
x2 <- cluster1_day$weekend    # weekend
x3 <- cluster1_day$temp       # temperature
```

```
library(rjags)
```

```
## Loading required package: coda

## Linked to JAGS 4.3.2

## Loaded modules: basemod,bugs
```

For daily data of cluster 1:

```
model_string <- "model {
  # Priors for the coefficients
  beta0 ~ dnorm(0, 0.01)
  beta1 ~ dnorm(0, 0.01)
  beta2 ~ dnorm(0, 0.01)
  beta3 ~ dnorm(0, 0.01)
  beta4 ~ dnorm(0, 0.01)
  beta5 ~ dnorm(0, 0.01)

  # Prior for the precision (inverse of variance)
  tau ~ dgamma(0.01, 0.01)

  # Initial value for y[1]
  y[1] ~ dnorm(0, 0.01)

  # Likelihood
  for (t in 2:N) {
    y[t] ~ dnorm(mu[t], tau)
    mu[t] <- beta0 + beta1 * y[t-1] +                                   # intercept and y[
             beta2 * sin(x1[t]) + beta3 * cos(x1[t]) +                  # time of year x1
             beta4 * x2[t] +                                            # weekend x2
             beta5 * x3[t]                                              # temperature x3
```

```
             }

}

"
```

```
datalist <- list(N = length(y), x1=x1, x2=x2, x3=x3)
```

```
model <- jags.model(file = textConnection(model_string),
                    data = datalist, n.chains = 3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 356
##    Total graph size: 3806
##
## Initializing model
```

```
update(model, n.iter = 1000)

Nrep = 10000

posterior_sample <- coda.samples(model,
                       variable.names = c("tau", "beta0", "beta1", "beta2", "beta3", "beta4", "beta5"),
                       n.iter = Nrep)
```

```
summary(posterior_sample)
```

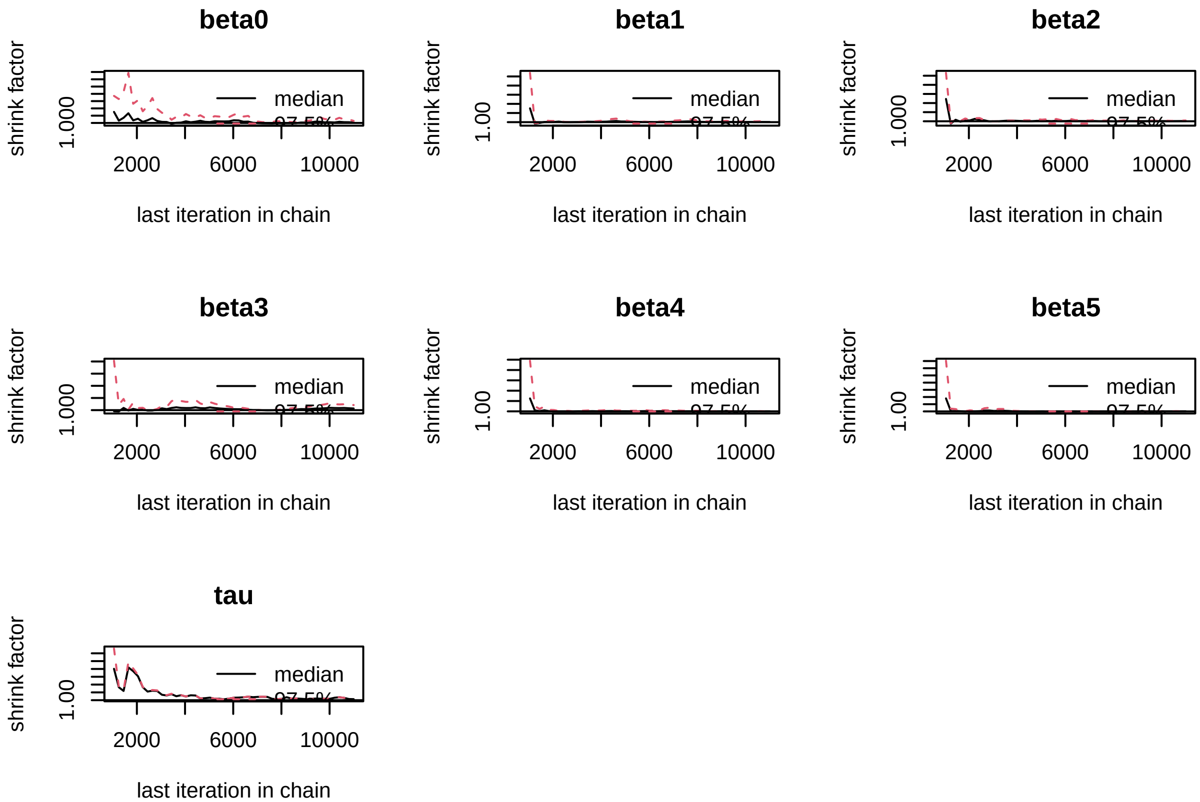
```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean     SD Naive SE Time-series SE
## beta0 -0.01574  9.946  0.05742        0.05843
## beta1  0.06046  9.923  0.05729        0.05729
## beta2  0.01285  9.987  0.05766        0.05766
## beta3  0.09654  9.986  0.05765        0.05765
## beta4  0.01966 10.019  0.05785        0.05759
## beta5 -0.02551 10.022  0.05786        0.05708
## tau    0.96827  9.358  0.05403        0.05524
##
## 2. Quantiles for each variable:
##
```

```
##              2.5%        25%        50%       75%  97.5%
## beta0  -1.960e+01 -6.712e+00 -1.128e-02 6.697e+00 19.435
## beta1  -1.922e+01 -6.561e+00  2.268e-02 6.689e+00 19.579
## beta2  -1.945e+01 -6.749e+00  1.589e-02 6.778e+00 19.533
## beta3  -1.960e+01 -6.660e+00  9.989e-02 6.905e+00 19.553
## beta4  -1.955e+01 -6.744e+00 -6.011e-02 6.765e+00 19.729
## beta5  -1.957e+01 -6.808e+00 -2.940e-02 6.787e+00 19.503
## tau    5.630e-157  1.621e-58  5.665e-29 1.824e-11  4.164
```

```
gelman.diag(posterior_sample)
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## beta0       1.00       1.00
## beta1       1.00       1.00
## beta2       1.00       1.00
## beta3       1.00       1.00
## beta4       1.00       1.00
## beta5       1.00       1.00
## tau         1.01       1.01
##
## Multivariate psrf
##
## 1
```

```
gelman.plot(posterior_sample)
```



For subset of cluster 1:

```
cluster1_data <- aggregated_data[, -c(2,3,4,5,6)]

num_rows <- nrow(cluster1_data)

# Take a random sample of 3000 rows from cluster1_data
set.seed(123)  # Setting seed for reproducibility
sample_indices <- sample(num_rows, 10000)
sampled_data <- cluster1_data[sample_indices, ]

y <- sampled_data$Cluster.1
x1 <- sampled_data$toy          # time of year
x2 <- sampled_data$weekend      # weekend
x3 <- sampled_data$temp         # temperature
x4 <- sampled_data$tod          # time of day
```

```
model_string2 <- "model {
  # Priors for the coefficients
  beta0 ~ dnorm(0, 0.01)
  beta1 ~ dnorm(0, 0.01)
  beta2 ~ dnorm(0, 0.01)
  beta3 ~ dnorm(0, 0.01)
```

```
  beta4 ~ dnorm(0, 0.01)
  beta5 ~ dnorm(0, 0.01)
  beta6 ~ dnorm(0, 0.01)
  beta7 ~ dnorm(0, 0.01)
  beta8 ~ dnorm(0, 0.01)
  beta9 ~ dnorm(0, 0.01)

  # Prior for the precision (inverse of variance)
  tau ~ dgamma(0.01, 0.01)
  sigma <- 1 / sqrt(tau)

  # Initial value for y[1]
  y[1] ~ dnorm(0, 0.01)

  # Likelihood
  for (t in 2:N) {
    y[t] ~ dnorm(mu[t], tau)
    mu[t] <- beta0 + beta1 * y[t-1] +                                              # intercept and y[t
             beta2 * sin(x1[t]) + beta3 * cos(x1[t]) +                             # time of year x1
             beta4 * x2[t] +                                                       # weekend x2
             beta5 * x3[t] +                                                       # temperature x3
             beta6 * x4[t] + beta7 * x4[t]^2 + beta8 * x4[t]^3 + beta9 * x4[t]^4     # time of day x4
             }

}

"
```

```
datalist <- list(N = length(y), x1=x1, x2=x2, x3=x3, x4=x4)
```

```
model <- jags.model(file = textConnection(model_string2),
                    data = datalist, n.chains = 3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 10011
##    Total graph size: 71815
##
## Initializing model
```

```
update(model, n.iter = 1000)

Nrep = 10000

posterior_sample <- coda.samples(model,
                       variable.names = c("tau", "beta0", "beta1", "beta2", "beta3", "beta4", "beta5",
                                          "beta6", "beta7", "beta8", "beta9"),
                       n.iter = Nrep)
```

```r
summary(posterior_sample)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean     SD Naive SE Time-series SE
## beta0 -0.043995 10.045  0.05800        0.05786
## beta1  0.005088 10.089  0.05825        0.05798
## beta2  0.062300 10.051  0.05803        0.05803
## beta3  0.048800  9.986  0.05765        0.05765
## beta4 -0.057834 10.057  0.05807        0.05783
## beta5  0.036889  9.992  0.05769        0.05769
## beta6 -0.082941 10.001  0.05774        0.05668
## beta7  0.037919  9.981  0.05763        0.05733
## beta8 -0.096393  9.874  0.05701        0.05701
## beta9  0.020788 10.078  0.05818        0.05818
## tau    1.104410 10.702  0.06179        0.06179
##
## 2. Quantiles for each variable:
##
##              2.5%        25%        50%       75%  97.5%
## beta0  -1.988e+01 -6.807e+00 -3.728e-02 6.791e+00 19.351
## beta1  -1.971e+01 -6.802e+00 -2.473e-02 6.740e+00 19.821
## beta2  -1.950e+01 -6.771e+00  3.211e-03 6.822e+00 19.886
## beta3  -1.924e+01 -6.758e+00  4.093e-02 6.815e+00 19.639
## beta4  -1.987e+01 -6.812e+00 -3.748e-02 6.759e+00 19.571
## beta5  -1.961e+01 -6.743e+00  5.549e-02 6.740e+00 19.568
## beta6  -1.959e+01 -6.907e+00 -8.182e-02 6.688e+00 19.673
## beta7  -1.955e+01 -6.709e+00  8.917e-02 6.859e+00 19.504
## beta8  -1.959e+01 -6.729e+00 -5.821e-02 6.665e+00 19.027
## beta9  -1.979e+01 -6.713e+00  7.962e-03 6.755e+00 19.747
## tau    1.872e-159  2.922e-59  3.429e-29 1.669e-11  5.644
```

```r
gelman.diag(posterior_sample)
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## beta0       1.00       1.00
## beta1       1.00       1.00
## beta2       1.00       1.00
## beta3       1.00       1.00
## beta4       1.00       1.00
## beta5       1.00       1.00
## beta6       1.00       1.00
## beta7       1.00       1.00
```

```
## beta8        1.00       1.00
## beta9        1.00       1.00
## tau          1.01       1.01
##
## Multivariate psrf
##
## 1
```

For cluster 2:

```r
cluster2_data <- aggregated_data[, -c(1,3,4,5,6)]

num_rows <- nrow(cluster2_data)

# Take a random sample of 3000 rows from cluster1_data
set.seed(123)  # Setting seed for reproducibility
sample_indices <- sample(num_rows, 10000)
sampled_data <- cluster2_data[sample_indices, ]

y <- sampled_data$Cluster.2
x1 <- sampled_data$toy        # time of year
x2 <- sampled_data$weekend    # weekend
x3 <- sampled_data$temp       # temperature
x4 <- sampled_data$tod        # time of day
```

```r
datalist <- list(N = length(y), x1=x1, x2=x2, x3=x3, x4=x4)
```

```r
model <- jags.model(file = textConnection(model_string2),
                    data = datalist, n.chains = 3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 10011
##    Total graph size: 71815
##
## Initializing model
```

```r
update(model, n.iter = 1000)

Nrep = 10000

posterior_sample <- coda.samples(model,
                       variable.names = c("tau", "beta0", "beta1", "beta2", "beta3", "beta4", "beta5",
                                          "beta6", "beta7", "beta8", "beta9"),
                       n.iter = Nrep)
```

```r
summary(posterior_sample)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean     SD Naive SE Time-series SE
## beta0 -0.083397  9.974  0.05758        0.05758
## beta1  0.049685  9.982  0.05763        0.05763
## beta2 -0.009574 10.026  0.05789        0.05789
## beta3  0.038107 10.021  0.05786        0.05786
## beta4  0.022735 10.061  0.05809        0.05809
## beta5 -0.037122  9.947  0.05743        0.05709
## beta6  0.028824 10.027  0.05789        0.05789
## beta7  0.007582  9.978  0.05761        0.05760
## beta8  0.019474 10.027  0.05789        0.05789
## beta9  0.048300 10.051  0.05803        0.05830
## tau    0.913490  9.540  0.05508        0.05508
##
## 2. Quantiles for each variable:
##
##              2.5%        25%        50%       75%  97.5%
## beta0  -1.953e+01 -6.746e+00 -1.098e-01 6.635e+00 19.491
## beta1  -1.963e+01 -6.635e+00  1.402e-01 6.816e+00 19.394
## beta2  -1.968e+01 -6.700e+00  5.204e-03 6.681e+00 19.585
## beta3  -1.931e+01 -6.661e+00 -2.919e-02 6.779e+00 19.747
## beta4  -1.971e+01 -6.722e+00 -7.127e-03 6.833e+00 19.776
## beta5  -1.964e+01 -6.729e+00 -9.477e-03 6.689e+00 19.576
## beta6  -1.962e+01 -6.756e+00  4.535e-02 6.728e+00 19.802
## beta7  -1.930e+01 -6.768e+00 -6.281e-02 6.710e+00 19.700
## beta8  -1.958e+01 -6.693e+00  2.473e-02 6.779e+00 19.663
## beta9  -1.975e+01 -6.722e+00  7.155e-02 6.854e+00 19.715
## tau    1.331e-160  9.413e-60  2.016e-29 1.432e-11  3.917
```

```r
gelman.diag(posterior_sample)
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## beta0       1.00       1.00
## beta1       1.00       1.00
## beta2       1.00       1.00
## beta3       1.00       1.00
## beta4       1.00       1.00
## beta5       1.00       1.00
## beta6       1.00       1.00
## beta7       1.00       1.00
```

```
## beta8       1.00       1.00
## beta9       1.00       1.00
## tau         1.02       1.02
##
## Multivariate psrf
##
## 1
```

For cluster 3:

```r
cluster3_data <- aggregated_data[, -c(1,2,4,5,6)]

num_rows <- nrow(cluster3_data)

# Take a random sample of 3000 rows from cluster1_data
set.seed(123)  # Setting seed for reproducibility
sample_indices <- sample(num_rows, 10000)
sampled_data <- cluster3_data[sample_indices, ]

y <- sampled_data$Cluster.3
x1 <- sampled_data$toy        # time of year
x2 <- sampled_data$weekend    # weekend
x3 <- sampled_data$temp       # temperature
x4 <- sampled_data$tod        # time of day
```

```r
datalist <- list(N = length(y), x1=x1, x2=x2, x3=x3, x4=x4)
```

```r
model <- jags.model(file = textConnection(model_string2),
                    data = datalist, n.chains = 3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 10011
##    Total graph size: 71815
##
## Initializing model
```

```r
update(model, n.iter = 1000)

Nrep = 10000

posterior_sample <- coda.samples(model,
                       variable.names = c("tau", "beta0", "beta1", "beta2", "beta3", "beta4", "beta5",
                                          "beta6", "beta7", "beta8", "beta9"),
                       n.iter = Nrep)
```

```r
summary(posterior_sample)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean     SD Naive SE Time-series SE
## beta0  0.054397  9.943  0.05741        0.05741
## beta1  0.014526 10.047  0.05801        0.05836
## beta2 -0.002894 10.099  0.05831        0.05831
## beta3  0.136461  9.983  0.05764        0.05839
## beta4  0.045188  9.993  0.05770        0.05843
## beta5 -0.106667  9.940  0.05739        0.05664
## beta6 -0.030050  9.904  0.05718        0.05677
## beta7  0.044912  9.977  0.05760        0.05792
## beta8  0.019410 10.000  0.05773        0.05813
## beta9 -0.015533  9.987  0.05766        0.05735
## tau    1.002664 10.435  0.06025        0.06025
##
## 2. Quantiles for each variable:
##
##              2.5%        25%        50%       75% 97.5%
## beta0  -1.941e+01 -6.650e+00  4.968e-02 6.736e+00 19.69
## beta1  -1.969e+01 -6.723e+00  4.137e-02 6.815e+00 19.61
## beta2  -1.973e+01 -6.767e+00 -8.295e-02 6.774e+00 19.81
## beta3  -1.953e+01 -6.662e+00  1.332e-01 6.943e+00 19.51
## beta4  -1.971e+01 -6.663e+00  7.759e-02 6.749e+00 19.57
## beta5  -1.980e+01 -6.782e+00 -5.097e-02 6.600e+00 19.23
## beta6  -1.937e+01 -6.783e+00  4.361e-02 6.688e+00 19.14
## beta7  -1.941e+01 -6.680e+00  3.918e-02 6.776e+00 19.73
## beta8  -1.975e+01 -6.691e+00  5.114e-02 6.744e+00 19.70
## beta9  -1.956e+01 -6.661e+00 -6.803e-02 6.757e+00 19.44
## tau    7.850e-160  3.657e-59  1.547e-29 1.386e-11  4.69
```

Same problem.