Fit Test Try 3

Blain Morin

December 21, 2017

$$Y_i = \alpha_{j[i]} + \sum \beta_p X_{pi} + \epsilon_i, for hospitals (i = 1 to N)$$

$$\alpha_j = a + \sum b_k W_{kj} + u_j, for markets (j = 1 to J)$$

Remove missing data for Stan:

```
###remove missing data colums
k12ReducedRG = k12ReducedRG %>%
  select(-reform, -joinnetwork)

###change data to only complete cases
k12ReducedRG = k12ReducedRG[complete.cases(k12ReducedRG),]

dim(k12ReducedRG)
```

```
## [1] 2920   28
```

Get m1 data ready for stan:

```
model1data = list(episode = k12ReducedRG$episode,
                  hrr = as.integer(as.factor(k12ReducedRG$hrr)),
                  K = length(unique(k12ReducedRG$hrr)),
                  N = nrow(k12ReducedRG))
```

```
model1stan = stan("model1.stan", data = model1data,  chains = 4, iter = 2000)
```

Compare max and min from stan simulations to observed data:

```
###Null model beta
print(model1stan, pars = c("beta[1]", "sigma_e", "sigma_w"))
```

```
## Inference for Stan model: model1.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean     sd     2.5%      25%      50%      75%
## beta[1] 19039.18   44.76  91.17 18819.88 18992.45 19055.41 19100.79
## sigma_e  2970.79   92.43 141.15  2700.45  2816.84  3034.63  3072.20
## sigma_w   347.68  362.63 528.40     0.99     4.68     9.74   884.44
##            97.5% n_eff Rhat
## beta[1] 19181.58     4 1.37
```

```
## sigma_e  3128.61     2 2.92
## sigma_w  1344.97     2 5.42
##
## Samples were drawn using NUTS(diag_e) at Fri Dec 22 12:17:20 2017.
## 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).
```

```
###Extract Maxes
model1maxes = extract(model1stan, pars = c("maximum"))

###Extract Mins
model1mins = extract(model1stan, pars = c("minimum"))

###Mean Max
mean(model1maxes$maximum)
```

```
## [1] 29918.34
```

```
summary(model1maxes$maximum)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27582   29194   29789   29918   30486   35694
```

```
###Mean Min
mean(model1mins$minimum)
```

```
## [1] 8315.222
```

```
summary(model1mins$minimum)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3730    7757    8437    8315    9014   10984
```

```
###Compare observed
summary(k12ReducedRG$episode)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7119   17126   19269   19076   21028   41469
```

Get m2 data ready for stan:

```
###rename columns
k12ReducedRG = k12ReducedRG %>%
  rename(cmirank = "Rank for Variable cmi",
         dshpctrank = "Rank for Variable dshpct",
         mdadjadmitrank = "Rank for Variable mdadjadmit")
```

```
### prepare data for STAN
model2data = list(episode = k12ReducedRG$episode,
                  hrr = as.integer(as.factor(k12ReducedRG$hrr)),
                  qstar = k12ReducedRG$qstarrating,
                  qieffort = k12ReducedRG$qieffort,
                  accredited = k12ReducedRG$jchaoaccredited,
                  urban = k12ReducedRG$urbanlocation,
                  mdaffiliation = k12ReducedRG$mdaffiliation,
```

```
                ownership = k12ReducedRG$ownershipstatus,
                bedsize = k12ReducedRG$hospitalbedsize,
                cmi = k12ReducedRG$cmirank,
                dsh = k12ReducedRG$dshpctrank,
                mdadjadmitrank = k12ReducedRG$mdadjadmitrank,
                K = length(unique(k12ReducedRG$hrr)),
                id = as.integer(as.factor(k12ReducedRG$Provider)),
                N = nrow(k12ReducedRG))
```

```
###run stan simulation

model2stan = stan("model2.stan", data = model2data, chains = 4, iter=2000, cores = 2)
```

Compare simulation max and mean to observed

```
print(model2stan, pars = c("beta[1]",
                           "beta[2]",
                           "beta[3]",
                           "beta[4]",
                           "beta[5]",
                           "beta[6]",
                           "beta[7]",
                           "beta[8]",
                           "beta[9]",
                           "beta[10]",
                           "beta[11]",
                           "sigma_e",
                           "sigma_w"))
```

```
## Inference for Stan model: model2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##              mean se_mean     sd     2.5%      25%      50%      75%
## beta[1]  14100.78   21.20 264.66 13569.17 13927.05 14096.52 14277.01
## beta[2]    -50.39    0.70  44.03  -135.85   -79.71   -50.19   -20.90
## beta[3]    114.22    1.30  82.42   -44.83    57.71   114.77   170.78
## beta[4]    375.67    1.58  99.70   179.31   311.30   377.72   444.29
## beta[5]    674.79   23.07 114.60   470.27   599.34   666.46   742.18
## beta[6]    109.17    0.85  53.45     6.17    73.46   109.16   145.58
## beta[7]    347.86   12.34  68.31   218.78   301.67   345.42   387.74
## beta[8]    572.88    7.82  76.59   420.87   521.29   572.83   624.30
## beta[9]   1577.58    0.69  43.64  1492.12  1548.37  1578.16  1606.05
## beta[10]   -30.01    1.17  35.79  -102.04   -53.33   -29.84    -6.02
## beta[11]   -81.23   10.05  41.65  -185.69  -101.32   -76.01   -53.52
## sigma_e   1753.72   20.80  64.40  1690.75  1720.92  1737.74  1756.94
## sigma_w    761.38   80.92 236.88    14.67   782.64   826.44   867.42
##              97.5% n_eff Rhat
## beta[1]  14622.45   156 1.03
## beta[2]     36.63  4000 1.00
## beta[3]    279.27  4000 1.02
## beta[4]    565.69  4000 1.00
```

```
## beta[5]    940.17    25 1.16
## beta[6]    211.39  4000 1.00
## beta[7]    500.64    31 1.12
## beta[8]    724.18    96 1.04
## beta[9]   1663.86  4000 1.02
## beta[10]    40.76   939 1.02
## beta[11]   -12.25    17 1.21
## sigma_e   1960.42    10 1.62
## sigma_w    943.45     9 1.82
##
## Samples were drawn using NUTS(diag_e) at Fri Dec 22 12:59:43 2017.
## 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).
```

```
###Reference for betas:

#      beta[1] + w[hrr[i]] + (intercept)
#      beta[2]*qstar[i] +
#      beta[3]*qieffort[i] +
#      beta[4]*accredited[i] +
#      beta[5]*urban[i] +
#      beta[6]*mdaffiliation[i] +
#      beta[7]*ownership[i] +
#      beta[8]*bedsize[i] +
#      beta[9]*cmi[i] +
#      beta[10]*dsh[i] +
#      beta[11]*mdadjadmitrank[i];



###Extract maxes
model2maxes = extract(model2stan, pars = c("maximum"))

###Extract mins
model2mins = extract(model2stan, pars = c("minimum"))

###Mean max
mean(model2maxes$maximum)
```

```
## [1] 10794490
```

```
summary(model2maxes$maximum)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  9030595 10410339 10738078 10794490 11109610 13083972
```

```
###Mean min
mean(model2mins$minimum)
```

```
## [1] 17130.12
```

```
summary(model2mins$minimum)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11110   15987   17166   17130   18303   22724
```

```r
###Compare observed
summary(k12ReducedRG$episode)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7119   17126   19269   19076   21028   41469
```