Tweeting Ram Navami Model Estimates

```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> # Tweeting #RamNavami
> # Using R version 4.0.5 (2020-03-31) -- "Shake and Throw"
> # April 26, 2021
>
> # Set Working Directory
>
> setwd("C://Replication_Data")
>
> # Open Libraries
> # Note: statnet version 2019.6, created on 2019-06-13
> # Note: informR Version 1.0-5 created on 2015-03-09.
>
> library(statnet)
Loading required package: tergm
Loading required package: ergm
Loading required package: network
network: Classes for Relational Data
Version 1.16.1 created on 2020-10-06.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
```

```
                    Mark S. Handcock, University of California -- Los Angeles
                    David R. Hunter, Penn State University
                    Martina Morris, University of Washington
                    Skye Bender-deMoll, University of Washington
 For citation information, type citation("network").
 Type help("network-package") to get started.


ergm: version 3.11.0, created on 2020-10-14
Copyright (c) 2020, Mark S. Handcock, University of California -- Los Angeles
                    David R. Hunter, Penn State University
                    Carter T. Butts, University of California -- Irvine
                    Steven M. Goodreau, University of Washington
                    Pavel N. Krivitsky, UNSW Sydney
                    Martina Morris, University of Washington
                    with contributions from
                    Li Wang
                    Kirk Li, University of Washington
                    Skye Bender-deMoll, University of Washington
                    Chad Klumb
                    Michal Bojanowski, Kozminski University
                    Ben Bolker
Based on "statnet" project software (statnet.org).
For license and citation information see statnet.org/attribution
or type citation("ergm").

NOTE: Versions before 3.6.1 had a bug in the implementation of the bd() constraint which distorted the
sampled distribution
somewhat. In addition, Sampson's Monks datasets had mislabeled vertices. See the NEWS and the documentation
for more details.

NOTE: Some common term arguments pertaining to vertex attribute and level selection have changed in 3.10.0.
See terms help for more
details. Use 'options(ergm.term=list(version="3.9.4"))' to use old behavior.

Loading required package: networkDynamic

networkDynamic: version 0.10.1, created on 2020-01-16
Copyright (c) 2020, Carter T. Butts, University of California -- Irvine
                    Ayn Leslie-Cook, University of Washington
                    Pavel N. Krivitsky, University of Wollongong
```

```
                    Skye Bender-deMoll, University of Washington
                    with contributions from
                    Zack Almquist, University of California -- Irvine
                    David R. Hunter, Penn State University
                    Li Wang
                    Kirk Li, University of Washington
                    Steven M. Goodreau, University of Washington
                    Jeffrey Horner
                    Martina Morris, University of Washington
Based on "statnet" project software (statnet.org).
For license and citation information see statnet.org/attribution
or type citation("networkDynamic").


tergm: version 3.7.0, created on 2020-10-15
Copyright (c) 2020, Pavel N. Krivitsky, UNSW Sydney
                    Mark S. Handcock, University of California -- Los Angeles
                    with contributions from
                    David R. Hunter, Penn State University
                    Steven M. Goodreau, University of Washington
                    Martina Morris, University of Washington
                    Nicole Bohme Carnegie, New York University
                    Carter T. Butts, University of California -- Irvine
                    Ayn Leslie-Cook, University of Washington
                    Skye Bender-deMoll
                    Li Wang
                    Kirk Li, University of Washington
                    Chad Klumb
Based on "statnet" project software (statnet.org).
For license and citation information see statnet.org/attribution
or type citation("tergm").

Loading required package: ergm.count

ergm.count: version 3.4.0, created on 2019-05-15
Copyright (c) 2019, Pavel N. Krivitsky, University of Wollongong
                    with contributions from
                    Mark S. Handcock, University of California -- Los Angeles
                    David R. Hunter, Penn State University
Based on "statnet" project software (statnet.org).
For license and citation information see statnet.org/attribution
```

```
or type citation("ergm.count").

NOTE: The form of the term 'CMP' has been changed in version 3.2 of 'ergm.count'. See the news or
help('CMP') for more information.

Loading required package: sna
Loading required package: statnet.common

Attaching package: 'statnet.common'

The following object is masked from 'package:base':

    order

sna: Tools for Social Network Analysis
Version 2.6 created on 2020-10-5.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
 For citation information, type citation("sna").
 Type help(package="sna") to get started.

Loading required package: tsna

statnet: version 2019.6, created on 2019-06-13
Copyright (c) 2019, Mark S. Handcock, University of California -- Los Angeles
                    David R. Hunter, Penn State University
                    Carter T. Butts, University of California -- Irvine
                    Steven M. Goodreau, University of Washington
                    Pavel N. Krivitsky, University of Wollongong
                    Skye Bender-deMoll
                    Martina Morris, University of Washington
Based on "statnet" project software (statnet.org).
For license and citation information see statnet.org/attribution
or type citation("statnet").

--- Please select a CRAN mirror for use in this session ---

There are updates for the following statnet packages on CRAN:
     Installed ReposVer Built
tsna "0.3.1"   "0.3.3"  "4.0.5"
Restart R and use "statnet::update_statnet()" to get the updates.
> library(informR)
```

```
Loading required package: abind
Loading required package: relevent
Loading required package: trust
Loading required package: coda
relevent: Relational Event Models
Version 1.0-4 created on March 9, 2015.
copyright (c) 2007, Carter T. Butts, University of California-
Irvine
 For citation information, type citation("relevent").
 Type help(package="relevent") to get started.

informR: Sequence Statistics for Relational Event Models
Version 1.0-5 created on 2015-03-09.
copyright (c) 2010, Christopher Steven Marcum, University of California-
Irvine
 For citation information, type citation("informR").
 Type help(package="informR") to get started.

>
> # Read Data
>
> Attributes_Two_or_More_Hashtags <- read.csv("Attributes_Two_or_More_Hashtags.csv", header=TRUE)
> Attributes_Top_30_Hashtags <- read.csv("Attributes_Top_30_Hashtags.csv", header=TRUE)
> Edges_Two_or_More_Hashtags <- read.csv("Edges_Two_or_More_Hashtags.csv", header=TRUE)
> Edges_Top_30_Hashtags <- read.csv("Edges_Top_30_Hashtags.csv", header=TRUE)
> eventlist <- read.csv("Eventlist_RamNavami.csv", header=TRUE)
>
> # Convert Data to Network Form
>
> Hashtag_Edges_Two_or_More_Matrix <- as.matrix(Edges_Two_or_More_Hashtags, directed=FALSE)
> Hashtag_Edges_Two_or_More_Network <- as.network(Hashtag_Edges_Two_or_More_Matrix, bipartite=1668,
directed=FALSE)
> Hashtag_Edges_Top_30_Matrix <- as.matrix(Edges_Top_30_Hashtags, directed=FALSE)
> Hashtag_Edges_Top_30_Network <- as.network(Hashtag_Edges_Top_30_Matrix, bipartite=1389, directed=FALSE)
>
> # Attach Attributes to Network
>
> Hashtag_Edges_Two_or_More_Network %v% "B1_Followed" <- Attributes_Two_or_More_Hashtags$B1_Followed
> Hashtag_Edges_Two_or_More_Network %v% "B1_Followers" <- Attributes_Two_or_More_Hashtags$B1_Followers
> Hashtag_Edges_Two_or_More_Network %v% "B1_Tweets" <- Attributes_Two_or_More_Hashtags$B1_Tweets
```

```
> Hashtag_Edges_Two_or_More_Network %v% "B2_DirectlyReferncesRamNavami" <-
Attributes_Two_or_More_Hashtags$B2_DirectlyReferncesRamNavami
> Hashtag_Edges_Two_or_More_Network %v% "B2_ReligiousConnotation" <-
Attributes_Two_or_More_Hashtags$B2_ReligiousConnotation
> Hashtag_Edges_Two_or_More_Network %v% "B2_Covid" <- Attributes_Two_or_More_Hashtags$B2_Covid
>
> Hashtag_Edges_Top_30_Network %v% "B1_Followed" <- Attributes_Top_30_Hashtags$B1_Followed
> Hashtag_Edges_Top_30_Network %v% "B1_Followers" <- Attributes_Top_30_Hashtags$B1_Followers
> Hashtag_Edges_Top_30_Network %v% "B1_Tweets" <- Attributes_Top_30_Hashtags$B1_Tweets
> Hashtag_Edges_Top_30_Network %v% "B2_DirectlyReferncesRamNavami" <-
Attributes_Top_30_Hashtags$B2_DirectlyReferncesRamNavami
> Hashtag_Edges_Top_30_Network %v% "B2_ReligiousConnotation" <-
Attributes_Top_30_Hashtags$B2_ReligiousConnotation
> Hashtag_Edges_Top_30_Network %v% "B2_Covid" <- Attributes_Top_30_Hashtags$B2_Covid
>
> # Estimate Logit Model of Binary Network
>
> Model_01 <- ergm(Hashtag_Edges_Top_30_Network ~ edges
+ + nodecov("B2_DirectlyReferncesRamNavami")
+ + nodecov("B2_ReligiousConnotation")
+ + nodecov("B2_Covid"),
+ control=control.ergm(12345)
+ )
Starting maximum pseudolikelihood estimation (MPLE):
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Finished MPLE.
Stopping at the initial estimate.
Evaluating log-likelihood at the estimate.
> summary(Model_01)
Call:
ergm(formula = Hashtag_Edges_Top_30_Network ~ edges + nodecov("B2_DirectlyReferncesRamNavami") +
    nodecov("B2_ReligiousConnotation") + nodecov("B2_Covid"),
    control = control.ergm(12345))

Iterations:  6 out of 20

Monte Carlo MLE Results:
                                    Estimate Std. Error MCMC % z value Pr(>|z|)
edges                               -2.87876    0.04195      0 -68.629   <1e-04 ***
nodecov.B2_DirectlyReferncesRamNavami -0.34249    0.04884      0  -7.012   <1e-04 ***
```

```
nodecov.B2_ReligiousConnotation        0.55975    0.04679      0  11.963   <1e-04 ***
nodecov.B2_Covid                       0.17301    0.08883      0   1.948   0.0515 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 57767  on 41670  degrees of freedom
 Residual Deviance: 21325  on 41666  degrees of freedom

AIC: 21333    BIC: 21368    (Smaller is better.)
>
> # Estimate Binary ERGM
>
> Model_02 <- ergm(Hashtag_Edges_Top_30_Network ~
+ nodecov("B2_DirectlyReferncesRamNavami") +
+ nodecov("B2_ReligiousConnotation") +
+ nodecov("B2_Covid") +
+ b1degree(2) +
+ edges,
+ control=control.ergm(12345)
+ )
Starting maximum pseudolikelihood estimation (MPLE):
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Finished MPLE.
Starting Monte Carlo maximum likelihood estimation (MCMLE):
Iteration 1 of at most 20:
Optimizing with step length 0.338468011445743.
The log-likelihood improved by 4.145.
Iteration 2 of at most 20:
Optimizing with step length 0.456074327409425.
The log-likelihood improved by 3.416.
Iteration 3 of at most 20:
Optimizing with step length 0.55067519287861.
The log-likelihood improved by 2.749.
Iteration 4 of at most 20:
Optimizing with step length 1.
The log-likelihood improved by 1.634.
Step length converged once. Increasing MCMC sample size.
Iteration 5 of at most 20:
Optimizing with step length 1.
The log-likelihood improved by 0.05197.
```

```
Step length converged twice. Stopping.
Finished MCMLE.
Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 .
This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the
mcmc.diagnostics() function.
> summary(Model_02)
Call:
ergm(formula = Hashtag_Edges_Top_30_Network ~ nodecov("B2_DirectlyReferncesRamNavami") +
    nodecov("B2_ReligiousConnotation") + nodecov("B2_Covid") +
    b1degree(2) + edges, control = control.ergm(12345))

Iterations:  5 out of 20

Monte Carlo MLE Results:
                                    Estimate Std. Error MCMC % z value Pr(>|z|)
nodecov.B2_DirectlyReferncesRamNavami -0.34681    0.04789      0  -7.241   <1e-04 ***
nodecov.B2_ReligiousConnotation        0.56221    0.04913      0  11.443   <1e-04 ***
nodecov.B2_Covid                       0.17773    0.08748      0   2.031   0.0422 *
b1deg2                                -0.52797    0.06848      0  -7.710   <1e-04 ***
edges                                 -2.88774    0.04398      0 -65.658   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 57767  on 41670  degrees of freedom
 Residual Deviance: 21260  on 41665  degrees of freedom

AIC: 21270    BIC: 21313    (Smaller is better.)
>
> # Estimate Relation Event Model (REM)
>
> rawevents <- cbind(eventlist$Hashtag, eventlist$Sender)
> evls <- gen.evl(rawevents)
Warning message:
In gen.evl(rawevents) :
  Event History(ies): 2 has(have) less than 2 events. Removing from eventlist.Event History(ies): 5
has(have) less than 2 events. Removing from eventlist.Event History(ies): 12 has(have) less than 2 events.
Removing from eventlist.Event History(ies): 13 has(have) less than 2 events. Removing from eventlist.Event
History(ies): 15 has(have) less than 2 events. Removing from eventlist.Event History(ies): 19 has(have)
less than 2 events. Removing from eventlist.Event History(ies): 20 has(have) less than 2 events. Removing
from eventlist.Event History(ies): 22 has(have) less than 2 events. Removing from eventlist.Event
```

```
History(ies): 23 has(have) less than 2 events. Removing from eventlist.Event History(ies): 24 has(have)
less than 2 events. Removing from eventlist.Event History(ies): 25 has(have) less than 2 events. Removing
from eventlist.Event History(ies): 30 has(have) less than 2 events. Removing from eventlist.Event
History(ies): 31 has(have) less than 2 events. Removing from eventlist.E [... truncated]
> names(evls)
[1] "eventlist" "event.key"
> evls$event.key
      id  event.type
 [1,] "a" "bjaishreeram"
 [2,] "b" "bstaysafe"
 [3,] "c" "bramnavmi"
 [4,] "d" "bramnavami2021"
 [5,] "e" "bramanavami2021"
 [6,] "f" "bramnavmi2021"
 [7,] "g" "bstayhome"
 [8,] "h" "bram"
 [9,] "i" "bjayshreeram"
[10,] "j" "bwednesdaythought"
[11,] "k" "bjaishriram"
[12,] "l" "bhindu"
[13,] "m" "blordrama"
[14,] "n" "bayodhya"
[15,] "o" "badipurush"
[16,] "p" "brama"
[17,] "q" "blordram"
[18,] "r" "bnavratri"
[19,] "s" "bshriram"
[20,] "t" "bramayan"
[21,] "u" "bfestival"
[22,] "v" "bramanavami"
[23,] "w" "bhappyramnavami"
[24,] "x" "bindia"
[25,] "y" "bramayana"
[26,] "z" "bhappyramnavami2021"
[27,] "A" "bsitaram"
[28,] "B" "bhanuman"
[29,] "C" "bhinduism"
[30,] "D" "bradhe"
> alpha.ints <- gen.intercepts(evls, basecat="bramnavami2021")
>
> Hashtag_Characteristics <- c("c+d+e+f+v+w+z", "a+h+i+k+l+m+n+o+p+q+r+s+t+u+y+A+B+C+D", "b+g")
```

```
> Model_03.sforms <- gen.sformlist(evls, Hashtag_Characteristics)
Note:
         c+d+e+f+v+w+z S-form(s) contains special regex syntax.
         Using slow search methods.
 Note:
         a+h+i+k+l+m+n+o+p+q+r+s+t+u+y+A+B+C+D S-form(s) contains special regex syntax.
         Using slow search methods.
 Note:
         b+g S-form(s) contains special regex syntax.
         Using slow search methods.
> Model_03.ints <- slbind(Model_03.sforms, alpha.ints)
> Model_03.ints2 <- sldrop(Model_03.ints, c("bjaishreeram", "bramnavmi", "bjaishriram", "bram",
+                                           "bayodhya", "bindia", "blordrama", "bramayana", "bshriram", "bhindu",
"bsitaram",
+                                           "blordram", "bnavratri", "bstaysafe", "bramanavami2021",
"bramanavami", "bramnavmi2021",
+                                           "bfestival", "bwednesdaythought", "brama", "bstayhome", "badipurush",
"bhinduism",
+                                           "bhappyramnavami2021", "bhanuman", "bhappyramnavami", "bramayan",
"bjayshreeram",
+                                           "bradhe"))
> Model_03 <- rem(evls$eventlist, Model_03.ints2, estimator = "BPM",
+                  prior.param=list(mu = 0, sigma = 100, nu = 4))
> summary(Model_03)
Egocentric Relational Event Model (Ordinal Likelihood)

                                           Post.Mode  Post.SD Z value  Pr(>|z|)
c+d+e+f+v+w+z                               1.979805 0.159020 12.4500 < 2.2e-16 ***
a+h+i+k+l+m+n+o+p+q+r+s+t+u+y+A+B+C+D  0.080042 0.287880  0.2780     0.781
b+g                                         3.514402 0.532268  6.6027 4.037e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Null deviance: 15972.02 on 2348 degrees of freedom
Residual deviance: 15811.75 on 2345 degrees of freedom
        Chi-square: 160.2779 on 3 degrees of freedom, asymptotic p-value 0
AIC: 15817.75 AICC: 15817.76 BIC: 15835.03
Log posterior: -8208.816
Prior parameters: mu=0 sigma=100 nu=4
>
> # Extension of Logit Model of Binary Network
>
```

```
> Model_04 <- ergm(Hashtag_Edges_Top_30_Network ~ edges
+ + nodecov("B1_Followed")
+ + nodecov("B1_Followers")
+ + nodecov("B1_Tweets")
+ + nodecov("B2_DirectlyReferncesRamNavami")
+ + nodecov("B2_ReligiousConnotation")
+ + nodecov("B2_Covid"),
+ control=control.ergm(12345)
+ )
Starting maximum pseudolikelihood estimation (MPLE):
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Finished MPLE.
Stopping at the initial estimate.
Evaluating log-likelihood at the estimate.
> summary(Model_04)
Call:
ergm(formula = Hashtag_Edges_Top_30_Network ~ edges + nodecov("B1_Followed") +
    nodecov("B1_Followers") + nodecov("B1_Tweets") + nodecov("B2_DirectlyReferncesRamNavami") +
    nodecov("B2_ReligiousConnotation") + nodecov("B2_Covid"),
    control = control.ergm(12345))

Iterations:  5 out of 20

Monte Carlo MLE Results:
                                    Estimate Std. Error MCMC % z value Pr(>|z|)
edges                             -2.842e+00  4.246e-02      0 -66.930   <1e-04 ***
nodecov.B1_Followed               -6.888e-06  6.325e-06      0  -1.089   0.2762
nodecov.B1_Followers              -5.171e-08  5.321e-08      0  -0.972   0.3311
nodecov.B1_Tweets                 -4.447e-06  1.005e-06      0  -4.423   <1e-04 ***
nodecov.B2_DirectlyReferncesRamNavami -3.427e-01  4.886e-02      0  -7.015   <1e-04 ***
nodecov.B2_ReligiousConnotation    5.601e-01  4.680e-02      0  11.967   <1e-04 ***
nodecov.B2_Covid                   1.731e-01  8.885e-02      0   1.948   0.0514 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 57767  on 41670  degrees of freedom
 Residual Deviance: 21291  on 41663  degrees of freedom

AIC: 21305    BIC: 21366    (Smaller is better.)
>
```

```
> # Estension of Binary ERGM
>
> Model_05 <- ergm(Hashtag_Edges_Two_or_More_Network ~
+ nodecov("B2_DirectlyReferncesRamNavami") +
+ nodecov("B2_ReligiousConnotation") +
+ nodecov("B2_Covid") +
+ b1degree(2) +
+ edges,
+ control=control.ergm(12345)
+ )
Starting maximum pseudolikelihood estimation (MPLE):
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Finished MPLE.
Starting Monte Carlo maximum likelihood estimation (MCMLE):
Iteration 1 of at most 20:
Optimizing with step length 0.398630731053877.
The log-likelihood improved by 2.594.
Iteration 2 of at most 20:
Optimizing with step length 0.496012181494262.
The log-likelihood improved by 2.498.
Iteration 3 of at most 20:
Optimizing with step length 1.
The log-likelihood improved by 1.21.
Step length converged once. Increasing MCMC sample size.
Iteration 4 of at most 20:
Optimizing with step length 1.
The log-likelihood improved by 0.1791.
Step length converged twice. Stopping.
Finished MCMLE.
Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 .
This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the
mcmc.diagnostics() function.
> summary(Model_05)
Call:
ergm(formula = Hashtag_Edges_Two_or_More_Network ~ nodecov("B2_DirectlyReferncesRamNavami") +
    nodecov("B2_ReligiousConnotation") + nodecov("B2_Covid") +
    b1degree(2) + edges, control = control.ergm(12345))

Iterations:  4 out of 20
```

```
Monte Carlo MLE Results:
                                    Estimate Std. Error MCMC %  z value Pr(>|z|)
nodecov.B2_DirectlyReferncesRamNavami  2.18075    0.04030      1   54.110   <1e-04 ***
nodecov.B2_ReligiousConnotation        1.34354    0.03265      0   41.148   <1e-04 ***
nodecov.B2_Covid                       0.63962    0.05863      0   10.910   <1e-04 ***
b1deg2                                -0.14494    0.06521      0   -2.223   0.0262 *
edges                                 -5.99556    0.02836      0 -211.395   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 1193167  on 860688  degrees of freedom
 Residual Deviance:   58653  on 860683  degrees of freedom

AIC: 58663    BIC: 58722    (Smaller is better.)
>
> # Extension of Relational Event Model (REM)
>
> Model_06 <- rem(evls$eventlist, Model_03.ints, estimator = "BPM",
+                 prior.param=list(mu = 0, sigma = 100, nu = 4))
> summary(Model_06)
Egocentric Relational Event Model (Ordinal Likelihood)

                                    Post.Mode   Post.SD  Z value  Pr(>|z|)
bjaishreeram                        -0.041438  0.091678  -0.4520   0.65127
bstaysafe                           -1.571357  0.147622 -10.6445 < 2.2e-16 ***
bramnavmi                           -0.433921  0.095752  -4.5317 5.851e-06 ***
bramanavami2021                     -2.735213  0.250264 -10.9293 < 2.2e-16 ***
bramnavmi2021                       -1.886608  0.172070 -10.9642 < 2.2e-16 ***
bstayhome                           -1.794499  0.161117 -11.1378 < 2.2e-16 ***
bram                                -0.435923  0.102391  -4.2575 2.068e-05 ***
bjayshreeram                        -2.716651  0.250578 -10.8416 < 2.2e-16 ***
bwednesdaythought                   -2.084431  0.187661 -11.1074 < 2.2e-16 ***
bjaishriram                         -0.601415  0.105164  -5.7189 1.072e-08 ***
bhindu                              -1.072837  0.123817  -8.6647 < 2.2e-16 ***
blordrama                           -0.896214  0.116151  -7.7160 1.199e-14 ***
bayodhya                            -0.896214  0.116151  -7.7160 1.199e-14 ***
badipurush                          -2.116180  0.190328 -11.1186 < 2.2e-16 ***
brama                               -1.765981  0.163384 -10.8088 < 2.2e-16 ***
blordram                            -1.360518  0.138270  -9.8396 < 2.2e-16 ***
bnavratri                           -1.507121  0.146666 -10.2759 < 2.2e-16 ***
```

```
bshriram                                -1.119357  0.125984  -8.8849 < 2.2e-16 ***
bramayan                                -1.966650  0.178170 -11.0380 < 2.2e-16 ***
bfestival                               -1.721529  0.160331 -10.7373 < 2.2e-16 ***
bramanavami                             -1.700023  0.158882 -10.6999 < 2.2e-16 ***
bhappyramnavami                         -2.842101  0.265766 -10.6940 < 2.2e-16 ***
bindia                                  -0.868043  0.115006  -7.5478 4.419e-14 ***
bramayana                               -0.877345  0.115382  -7.6039 2.864e-14 ***
bhappyramnavami2021                     -2.292068  0.205981 -11.1276 < 2.2e-16 ***
bsitaram                                -1.246108  0.132214  -9.4249 < 2.2e-16 ***
bhanuman                                -1.939251  0.176051 -11.0153 < 2.2e-16 ***
bhinduism                               -1.721529  0.160331 -10.7373 < 2.2e-16 ***
bradhe                                  -2.459119  0.222308 -11.0618 < 2.2e-16 ***
c+d+e+f+v+w+z                            0.929290  0.168729   5.5076 3.638e-08 ***
a+h+i+k+l+m+n+o+p+q+r+s+t+u+y+A+B+C+D -1.024668  0.293461  -3.4917   0.00048 ***
b+g                                      4.017091  0.542391   7.4063 1.299e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Null deviance: 15972.02 on 2348 degrees of freedom
Residual deviance: 14563.4 on 2316 degrees of freedom
        Chi-square: 1408.628 on 32 degrees of freedom, asymptotic p-value 0
AIC: 14627.4 AICC: 14628.31 BIC: 14811.76
Log posterior: -10513.09
Prior parameters: mu=0 sigma=100 nu=4
>
```