gnm: an R Package for Generalized Nonlinear Models

Heather Turner

Department of Statistics
University of Warwick, UK

Overview

- What is a generalized nonlinear model (GNM)?
- How does gnm fit GNMs?
- What are the key functions in gnm?
- Using gnm to fit a 'standard' GNM
- Using gnm to fit a custom GNM

Generalized Linear Models

- A GLM is made up of a **linear predictor**

$$\eta = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

and two functions

  - a **link** function that describes how the mean, $E(Y) = \mu$, depends on the linear predictor

$$g(\mu) = \eta$$

  - a **variance** function that describes how the variance, $Var(Y)$ depends on the mean

$$Var(Y) = \phi V(\mu)$$

where the **dispersion parameter** ϕ is a constant

Generalized Nonlinear Models

- A **generalized nonlinear model (GNM)** is the same as a GLM except that we have

$$g(\mu) = \eta(x; \beta)$$

  where $\eta(x; \beta)$ is nonlinear in the parameters β.
- Thus a GNM may also be considered as an extension of a nonlinear least squares model in which the variance of the response is allowed to depend on the mean.
- There a several models in the literature that fit within this framework.

Models for Contingency Tables

- Goodman's row-column association model for 2 way tables

$$\log \mu_{ij} = \alpha_i + \beta_j + \gamma_i \delta_j$$

- UNIDIFF model for 3 way tables

$$\log \mu_{ijk} = \alpha_{ik} + \beta_{jk} + \gamma_k \delta_{ij}$$

- Diagonal reference model for square tables

$$\mu_{ij} = w\gamma_i + (1 - w)\gamma_j$$

- These are specific examples with multiplicative terms

More Models with Multiplicative Terms

- AMMI model for Gaussian crop yields

$$\mu_{ij} = \alpha_i + \beta_j + \sigma_1\gamma_{1i}\delta_{1j} + \sigma_2\gamma_{2i}\delta_{2j}$$

- Lee-Carter model for (Quasi-)Poisson mortality rates

$$\log(\mu_{ay}/e_{ay}) = \alpha_a + \beta_a\gamma_y,$$

- Rasch-type model for Binomial voting data

$$\text{logit}(\mu_{rm}) = \alpha_r + \beta_r\gamma_m$$

- Stereotype model for ordered Multinomial data

$$\log\mu_{ic} = \beta_{0c} + \gamma_c(\beta_1 x_{1i} + \beta_2 x_{2i})$$

Other Models

- Although most standard applications have multiplicative terms, there is no restriction to such models.
- For example, gnm may be used to exponential decay models of the form

$$\mu = \alpha + \exp(\beta_1 + \gamma_1 x) + \exp(\beta_2 + \gamma_2 x)$$

which `nls` is unable to fit.

The gnm Function

- Models are specified via symbolic formulae
  - functions of class "nonlin" to specify nonlinear terms
- Single IWLS algorithm for all models
  - works with over-parameterized models
- Patterned after glm
  - similar arguments, returned objects, methods, etc

Model Specification

- Linear terms in the model may be specified in the usual way, e.g. `y ~ a + b + a:b`
- Nonlinear terms must be specified using functions of class `"nonlin"`
  - specify structure of term, possible also labels & starting values
  - provided functions: `Exp`, `Inv`, `Mult`, `MultHomog`, `Dref`
  - custom functions

Nesting and Instances

- Nonlin terms may be nested, e.g. for a UNIDIFF model:

$$\log \mu_{ijk} = \alpha_{ik} + \beta_{jk} + \exp(\gamma_k)\delta_{ij}$$

  the exponentiated multiplier is specified as

  `Mult(Exp(C), A:B)`

- Multiple instances e.g. in Goodman's RC(2) model:

$$\log \mu_{rc} = \alpha_r + \beta_c + \gamma_r\delta_c + \theta_r\phi_c$$

  may be specified using the `instances` function:

  `instances(Mult(A, B), 2)`

Arguments of "nonlin" Terms

- Arguments of "nonlin" terms need not be single variables, e.g. an exponential decay model

$$\mu = \alpha + \exp(\beta_1 + \gamma_1 x) + \exp(\beta_2 + \gamma_2 x)$$

  may be specified as

```
y ~ instances(Exp(1 + x), 2)
```

- Intercepts are not added to predictor arguments of "nonlin" terms by default

Working with Over-Parameterised Models

- gnm does not impose any identifiability constraints on the nonlinear parameters
  - the same model can be represented by an infinite number of parameterisations, e.g.

$$\begin{aligned}\log \mu_{rc} &= \alpha_r + \beta_c + \gamma_r \delta_c \\ &= \alpha_r + \beta_c + (2\gamma_r)(0.5\delta_c) \\ &= \alpha_r + \beta_c + \gamma_r^{'} \delta_c^{'}\end{aligned}$$

  - gnm will return one of these parameterisations, at random
- Rules for constraining nonlinear parameters not required
- Fitting algorithm must be able to handle singular matrices

Parameter Estimation

- Wish to estimate the predictor

$$\eta = \eta(\beta)$$

which is nonlinear, so we have a **local** design matrix

$$X(\beta) = \frac{\partial \eta}{\partial \beta}$$

where X is not of full rank, due to over-parameterisation

- Use maximum likelihood estimation: want to solve the likehood score equations

$$U(\beta) = \nabla l(\beta) = 0$$

Fitting Algorithm

- Use a two stage procedure:
  - one-parameter-at-a-time Newton method to update nonlinear parameters
  - full Newton-Raphson to update all parameters but with the Moore-Penrose pseudoinverse $(X^T W X)^-$
- Starting values are obtained in two ways:

  for the linear parameters use estimates from a `glm` fit

  for the nonlinear parameters generate randomly

  - parameterisation determined by the starting values of nonlinear parameters

Estimating Identifiable Parameter Combinations

- Prior to fitting
  - using arguments `constrain` and `constrainTo`
- After fitting
  - estimate simple contrasts using `getContrasts`
  - estimate linear combinations of parameters using se
- Both `getContrasts` and `se` check estimability first

Example: Yaish Data

- Study of social mobility by Yaish (1998, 2004)
- 3-way contingecny table classified by:

  `orig` father's social class (7 levels)
  `dest` son's social class (7 levels)
  `educ` son's education level (5 levels)

UNIDIFF Model

- In a UNIDIFF model

$$\log \mu_{ijk} = \alpha_{ik} + \beta_{jk} + \exp(\gamma_k)\delta_{ij}$$

  $\exp(\gamma_k)$ is the strength of association over dimension indexed by i and j.

- Fit to yaish data:

```
> unidiff <- gnm(Freq ~ educ*orig + educ*dest
    + Mult(Exp(educ), orig:dest),
    ofInterest = "[.]educ",
    family = poisson,
    data = yaish, subset = (dest != 7))
```

Summary of Fitted UNIDIFF Model

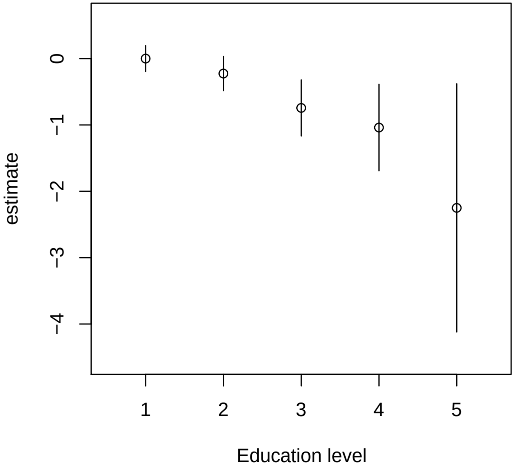
```
Call:
gnm(formula = Freq ~ educ * orig + educ * dest + Mult(Exp(educ),
    orig:dest), ofInterest = "[.]educ", family = poisson, data = yaish,
    subset = (dest != 7))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0286  -0.6402  -0.1048   0.5813   2.7459

Coefficients of interest:
                              Estimate Std. Error z value Pr(>|z|)
Mult(Exp(.), orig:dest).educ1  -0.4531         NA      NA       NA
Mult(Exp(.), orig:dest).educ2  -0.6785         NA      NA       NA
Mult(Exp(.), orig:dest).educ3  -1.1965         NA      NA       NA
Mult(Exp(.), orig:dest).educ4  -1.4920         NA      NA       NA
Mult(Exp(.), orig:dest).educ5  -2.7026         NA      NA       NA

Std. Error is NA where coefficient has been constrained or is unidentified

Residual deviance: 200.33 on 116 degrees of freedom
AIC: 1140.4
```

Contrasts of Strength Parameters

```
> unidiffContrasts <- getContrasts(unidiff, ofInterest(unidiff))
> summary(unidiffContrasts, digits = 2)

Model call:  gnm(formula = Freq ~ educ * orig + educ * dest +
    Mult(Exp(educ), orig:dest), ofInterest = "[.]educ",
    family = poisson, data = yaish, subset = (dest != 7))
                                  estimate   SE quasiSE quasiVar
    Mult(Exp(.), orig:dest).educ1     0.00 0.00   0.098   0.0095
    Mult(Exp(.), orig:dest).educ2    -0.23 0.16   0.129   0.0166
    Mult(Exp(.), orig:dest).educ3    -0.74 0.23   0.212   0.0449
    Mult(Exp(.), orig:dest).educ4    -1.04 0.34   0.326   0.1063
    Mult(Exp(.), orig:dest).educ5    -2.25 0.95   0.936   0.8754
Worst relative errors in SEs of simple contrasts (%):  -0.9 1.4
Worst relative errors over *all* contrasts (%):  -3.6 2.1
```

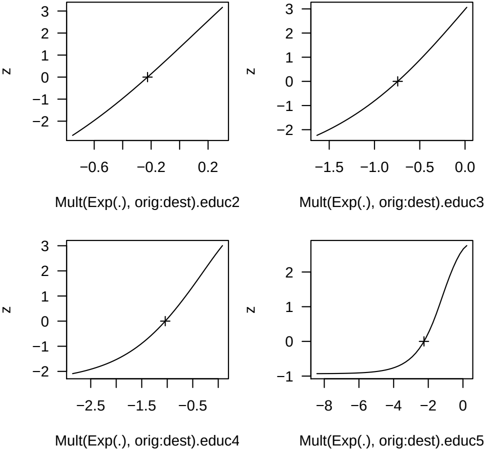
Contrasts Plot

```
plot(unidiffContrasts, xlab = "Education Level", levelNames = 1:5)
```



Profiling

```
unidiff2 <- update(unidiff, constrain = "[.]educ1")
prof <- profile(unidiff2, ofInterest(unidiff2), trace = TRUE)
plot(prof)
```



Profile Confidence Intervals

```
> conf <- confint(prof)
> print(conf, digits = 2)
2.5 % 97.5 %
Mult(Exp(.), orig:dest).educ1    NA     NA
Mult(Exp(.), orig:dest).educ2  -0.6    0.1
Mult(Exp(.), orig:dest).educ3  -1.5   -0.2
Mult(Exp(.), orig:dest).educ4  -2.6   -0.3
Mult(Exp(.), orig:dest).educ5  -Inf   -0.7
```

Example: Marriage Data

- The Living in Ireland Surveys were conducted 1994-2001
- For five 5-year cohorts of women born between 1950 and 1975 we have the following data
  - year of (first) marriage
  - year and month of birth
  - social class
  - highest level of education attained
  - year highest level of education was attained

Discrete-time Hazard Models

- For discrete-time the **hazard** of marriage occuring at time t is defined as

$$h(t) = P(T = t | T \geq t)$$

- We can model the hazard using models of the form

$$\operatorname{logit}(h(t|\boldsymbol{x}_{it})) = \alpha(age_{it}) + \boldsymbol{x}'_{it}\boldsymbol{\beta}$$

Episode-splitting

- To estimate the discrete-time hazard model we generate an **event history** for each observation
- Pseudo observations are created at each time point from time 0 up to marriage or censoring - this is known as **episode-splitting**
- The parameters can then be estimated by logistic regression of a marriage indicator at each time point (married $= 1$, unmarried $= 0$)

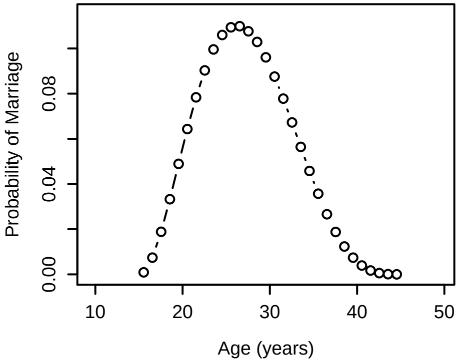
Blossfeld and Huinink Model

- Blossfeld and Huinink (Am. J. Sociol., 1991) propose the following linear baseline

$$\alpha(age_{it}) = c + \beta_l \log(age_{it} - 15) + \beta_r \log(45 - age_{it})$$

  - describes the nature of the time dependence
  - fixes the support of the hazard to be 15 to 45 years

BH Model



Nonlinear Discrete-time Hazard Model

- An obvious extension of the BH model is to treat the endpoints as parameters

$$\alpha(age_{it}) = c + \beta_l \log(age_{it} - \alpha_l) + \beta_r \log(\alpha_r - age_{it})$$

  - nonlinear
  - can't specify with standard `"nonlin"` functions

Variables and Predictors

- A `"nonlin"` function creates a list of arguments for the internal function `nonlinTerms`
- Nonlinear terms are considered as functions of variables and predictors

$$\beta_l \log(age_{it} - \alpha_l) + \beta_r \log(\alpha_r - age_{it})$$

- Create `"nonlin"` function `Bell` with argument `x`, which returns the arguments

```
predictors = list(slope = 1, endpoint = 1),
variables = list(substitute(x))
```

Term-specific Issues

- Would like to use same function for both "log-excess" terms, so add argument

  ```
  side = "left"
  ```

- Need to constrain endpoints to avoid undefined log values, so define

  ```
  constraint <- ifelse(side == "right",
                       max(x) + 1e-5, min(x) - 1e-5)
  ```

Term

- The `term` argument of `nonlinTerms` takes labels for the predictors and variables and returns a deparsed expression of the term:

```
term = function(predLabels, varLabels) {
    paste(predLabels[1], " * log(",
        " -"[side == "right"], varLabels[1], " + ",
        " -"[side == "left"], constraint,
        " + exp(", predLabels[2], "))")
    }
```

Parameter Labels

- Default parameter labels are taken from the `predictor` names, here `slope` and `endpoint`
- To make parameter labels unique, save call to `Bell`:

  ```
  call <- sys.call()
  ```

  and specify `call` argument to `nonlinTerms`

  ```
  call = as.expression(call)
  match = c(0, 0)
  ```

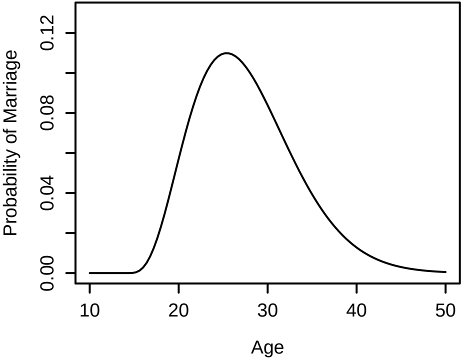
Complete Function

```
Bell <- function(x, side = "left"){
    call <- sys.call()
    constraint <- ifelse(side == "right",
                         max(x) + 1e-5, min(x) - 1e-5)
    list(predictors = list(slope = 1, endpoint = 1),
         variables = list(substitute(x)),
         term = function(predLabels, varLabels) {
             paste(predLabels[1], " * log(",
                   " -"[side == "right"], varLabels[1], " + ",
                   " -"[side == "left"], constraint,
                   " + exp(", predLabels[2], "))")
         },
     call = as.expression(call),
     match = c(0, 0)
     )
 }
 class(Bell) <- "nonlin"
```

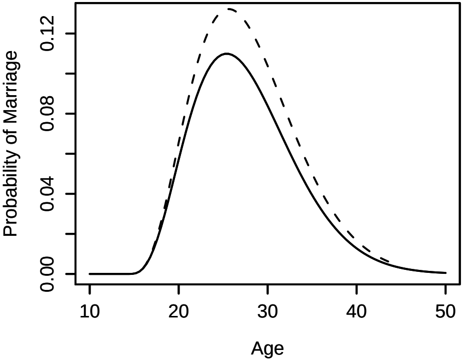
Summary of Extended Model

```
Call:
gnm(formula = marriages/lives ~ Bell(age, side = "left") + Bell(age,
    side = "right"), family = binomial, data = fulldata, weights = lives,
    start = c(-20, 3, 0, 3, 0))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8098  -0.4441  -0.3224  -0.1528   4.0483

Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -118.5395         NA      NA       NA
Bell(age, side = "left")slope        3.6928         NA      NA       NA
Bell(age, side = "left")endpoint    -0.1432         NA      NA       NA
Bell(age, side = "right")slope      24.8623         NA      NA       NA
Bell(age, side = "right")endpoint    4.0247         NA      NA       NA

Std. Error is NA where coefficient has been constrained or is unidentified

Residual deviance: 12553 on 31004 degrees of freedom
AIC: 12748
```

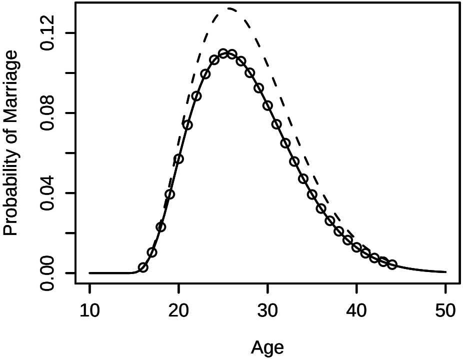
Example ‘Recoil’ Plot



Example 'Recoil' Plot



Example 'Recoil' Plot



Re-parameterization

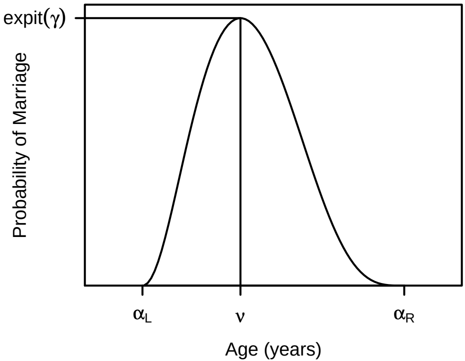
- The problem with aliasing can be overcome by re-parameterizing the model:

$$\begin{aligned}\alpha(age_{it}) = \gamma - \delta &\left\{(\nu - \alpha_l)\log\left(\frac{\nu - \alpha_l}{age_{it} - \alpha_l}\right)\right\} \\ &+ \delta\left\{(\alpha_r - \nu)\log\left(\frac{\alpha_r - \nu}{\alpha_r - age_{it}}\right)\right\}\end{aligned}$$

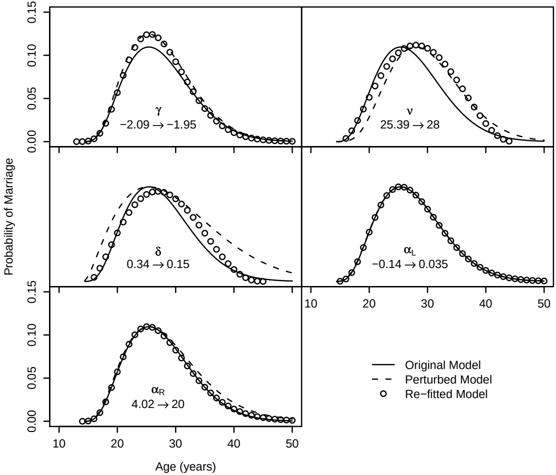
- A new `nonlin` function, `Surge`, is need to specify this term

Interpretation of Parameters

- The parameters of the new parameterisation have a more useful interpretation than before:



Recoil Plots for Reparameterised Model



Infinite Right Endpoint

- Having gone through a process of variable selection, the estimate for the right endpoint is 400 years!
- Letting the right end-point tends to infinity:

$$\alpha(age_{it}) = \gamma - \delta \left\{ (\nu - \alpha_l) \log \left(\frac{\nu - \alpha_l}{age_{it} - \alpha_l} \right) - age_{it} - \nu \right\}$$

  does not significantly increase the deviance
- An argument is added to `Surge` to specify whether the right endpoint should be estimated

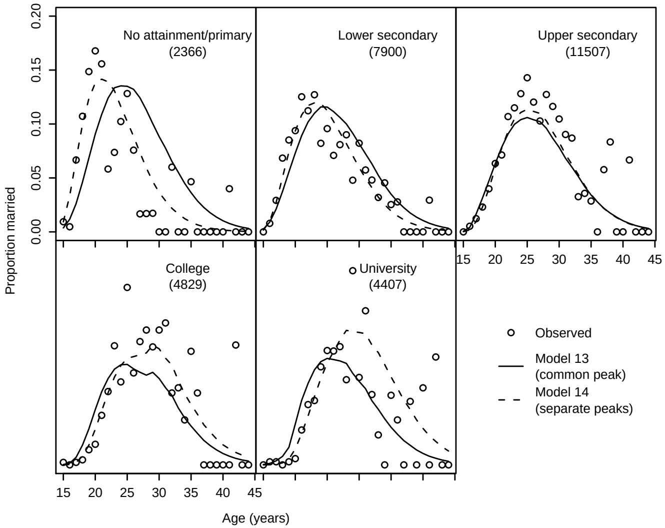
Refining the Model

- Checking the fit of the model over each covariate suggests some changes in the predictors
  - e.g. replacing the cohort factor by the nonlinear term

$$\theta \exp(\lambda(yrb_i - 1950))$$

- Residual analysis also suggests that both the scale and location of hazard vary between individuals

Fit over Education Levels



Linear Dependence of Peak Location

- Quantifying the education level by the average equivalent years in education *ed* a linear dependence of peak location on age can be incorporated as follows

$$\alpha(\boldsymbol{x}_{it}) = \gamma - \delta \left\{ (\nu_0 + \nu_1 ed_i - \alpha_l) \log \left(\frac{\nu_0 + \nu_1 ed_i - \alpha_l}{age_{it} - \alpha_l} \right) \right\}$$
$$+ \delta \left\{ age_{it} + \nu_0 + \nu_1 ed_i \right\}$$

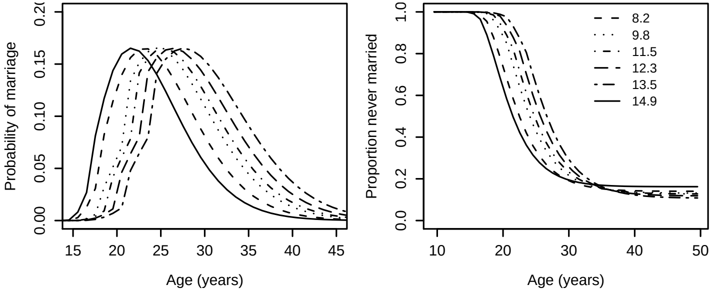
- An argument is added to `Surge` to specify the formula for the peak location

Final Model

```
Coefficients:
                                                    (Intercept)
                                                    -1.59971836
Surge(age, peakX = ~ . + YrsEduc, right = Inf).peakX(Intercept)
                                                    14.42125516
          Surge(age, peakX = ~ 1 + ., right = Inf).peakXYrsEduc
                                                     0.88430137
          Surge(age, peakX = ~ 1 + YrsEduc, right = Inf)fallOff
                                                     0.46183848
          Surge(age, peakX = ~ 1 + YrsEduc, right = Inf)leftAdj
                                                     0.16872262
                     Mult(., Exp(I(iyearb - 1950))).(Intercept)
                                                    -0.01991675
                               Mult(1, Exp(.)).I(iyearb - 1950)
                                                     0.19665983
                                                         InEduc
                                                    -1.46281777
                                                       PostEduc
                                                    -0.47859895
```

Hazard and Survival Curves

- For women born in 1950



Deviance = 11847 Residual d.f. = 31000

Interpretation

- $\hat{\alpha}_L = 13.86$ and the deviance is significantly increased if this is constrained to 15 years
- Peak location varies from 21.32 years (no education) to 27.60 years (university graduates)
- Peak hazard varies from 0.17 (b. 1950) through 0.16 (b. 1960) to 0.07 (b. 1970)

References

- More information about gnm can be found on www.warwick.ac.uk/go/gnm
- A comprehensive manual is distributed with the package
  `vignette("gnmOverview", package = "gnm")`
- A working paper on the marriage application is available at www.warwick.ac.uk/go/crism/research/2007

Acknowledgements

- The marriage data are from The Economic and Social Research Institute Living in Ireland Survey Microdata File (©Economic and Social Research Institute).
- We gratefully acknowledge Carmel Hannan for introducing us to this application and providing background on the data.