The Spatial Model Functions

December 9, 2008

1 Introduction

2 Method

2.1 The Poisson Model

The followings are the theoretical functions:

Let Y_k denotes the number of cancer cases in group k. P_k denotes the number of people (population size). λ_k denotes the rate of getting cancer.

We have

$$Y_k \sim Poisson(P_k \lambda_k),$$

we won't put time under our consideration at the beginning.

$$\log(\lambda_k) = \mu + x_k \beta + U_k + V_k,$$

where μ is the intercept, it could be $\log(P_k)$;

x_k is the covariates i.e income variable;

β is the log relative risk;

U_k is the spatial random variable and

$$U_k \sim N(0, \sigma_u^2)$$

V_k is the non-spatial random variable and

$$V_k \sim N(0, \sigma_v^2)$$

Therefore, we will fit a *glm* (generalized linear model) as:

$$model = glm(cases \sim offset(logpop) + factor(age)factor(sex))$$

2.2 The R Functions

Currently the diseasemapping package contains 6 functions:

1. `''formatCases'' <- function(casedata)`

   We get the data sets from different kinds of sources. Therefore, we need to clean and format the data set for the future manipulation. The function dealing with two scenarios: when there is `group` column in the data set or when there is `AGE_SEX_GROUP` column in data set. Essentially, we want to add in two columns : sex and age.

$M0\_5$
$M5\_9$
$M10\_14$
$\vdots$
$M85PLUS$

Group is usually a column looks like:

AGE_SEX_Group is usually a column of numbers composed with 3 digits: Male are the numbers begin with 1, the rest two digits are corresponding to the age. i.e:

100	$M0\_4$
101	$M5\_9$
102	$M10\_14$
$\vdots$	$\vdots$
117	$M85PLUS$

Female is the same, but starting at 200.

2. `''formatPopulation'' <- function(popdata)`

   This function is similar to the formatCases function. In some of the population data set, there will be a group of the people who are age 85 and plus, marked as: $M85PLUS$ or $F85PLUS$ for male and female respectively. The first part of the formatPopulation function is to change them to $M85\_89$ or $F85\_89$, so that they will have the same format as the other sex and age group, and that will simplifies our manipulation later. The second part of the function is to reshape the population data set using the function `reshape`. So that it changes the population data set from the wide format to the long format, and it will have the same format as the case data set. This step make the merge of case data set and population data set easier in the future calculations.

3. `''getRates'' <-`
   `function(casedata, popdata, formula, family=poisson,`
   `minimumAge=0, maximumAge=100, S=c("M", "F"))`

   This is the function we constructed to fit a Generalized Linear Model (`glm` function) to the case data set and the population data set. Then we get out estimated coefficients for the future prediction. The argument `casedata` and `popdata` are the places where we put case data set and population data set correspondingly. `formula` is the place we fit a formula function in the `glm` model. i.e : $age * sex$. `family` is the place we choose what distribution we want to fit in the `glm` model. By default we set it to be Poisson model, it can be binomial as well. $minimumAge = 0$ and $maximumAge = 100$ are the arguments to control the sample's age range. By default it is $age \in (0, 100)$, which includes all the cases. You can change the range to (0, 10) for chose all the kids' data set. `S` is in charge of control what kind of sex group we want to choose. By default it choose both the male and female.

4.
   ```
   ''getSMR'' <-
   function(model, population, cases=NULL, regionCode="CSDUID",
     regionCodeCases="CSD2006")
   ```

   Function `getSMR` is used to do the prediction using the model we get from the `getRates` function. Eventually, it returns a shape file (if the population data set we insert the function is a shape file) contains everything of the original population data set and several columns : expected, logExpected, cases, SMR. expected are the expected case numbers we estimated using the model we get from `getRates` function. logExpected are the log values of the expected values. cases are the number of cases we got from the case data set (if we fit a poisson distribution it will be the aggregate number of cases by different Decent Areas. SMR is the rate of observed number of cases over expected number of cases.

   The argument `model` is the glm model estimated coefficients we got from the `getRates` function. `population` is the population data set, it can be a shape file. `cases` is the number of case data set. `regionCodeCases` and `regionCode` are used to indicate the region for case data and population data sets respectively.

5.
   ```
   ''area'' <- function(sp)
   ```

   Function `area` is used to calculate the area of each region, return a vector of areas. The argument `sp` is a single spatial polygon object.

6.
   ```
   ''mergeBugsData'' <- function(x, bugsSummary, by.x = NULL, newcol="mean", ...)
   ```

   Function `mergeBugsData` is used to merge the results from the `bugs` function to the other data set i.e population data set.

   The argument `x` is the data set we want the result from the `bugs` to merge to. `bugsSummary` is the simulation result get from the `bugs` function. `by.x` is the common column in both the x and bugsSummary data set. `newcol` is the summary statistic that to be merged back to the data frame.

2.3 The Examples

We will take the ontario long cancer data set as an example. Due to confidential needs, we simulated the case (cancer) data set. For example, we have a case data set as:

¿ data(casedata) ¿ head(casedata)
library

Year	CSD2006	CD2006	PR2006	AGE_SEX_GROUP	Cases
1999	3501005	3501	35	210	3
1999	3501005	3501	35	213	1
1999	3501005	3501	35	214	6
1999	3501005	3501	35	215	1
1999	3501005	3501	35	216	3

¿ formatCases(casedata) ¿ head(casedata)
After applying the `formatCases` function, it will be:
So that we can see, there are two new columns: sex and age.

Year	CSD2006	CD2006	PR2006	AGE_SEX_GROUP	Cases	sex	age
1999	3501005	3501	35	210	3	F	50
1999	3501005	3501	35	213	1	F	65
1999	3501005	3501	35	214	6	F	70
1999	3501005	3501	35	215	1	F	75
1999	3501005	3501	35	216	3	F	80

Now let's look at the `formatPopulation` function:
For example, we have a population data set as:
¿ data(popdata) ¿ head(popdata@data)

CSDUID	CSDNAME	M0_4	M5_9	...	F0_4	F5_9
3501005	South Glengarry	295	355	...	285	320
3501011	South Stormont	310	380	...	265	345
3501012	Cornwall	1180	1295	...	1080	1210
3501020	South Dundas	230	285	...	240	300
3501030	North Dundas	295	315	...	310	285
3501042	North Stormont	200	220	...	180	220

After applying the `formatPopulation` function, it will be:
¿ formatPopulation(popdata)

CSDUID	CSDNAME	M0_4	...	F0_4	...	GROUP	POPULATION	AGE	SEX
3501005	South Glengarry	295	...	285	...	M0_4	295	0_4	M
3501011	South Stormont	310	...	265	...	M0_4	310	0_4	M
3501012	Cornwall	1180	...	1080	...	M0_4	1180	0_4	M
3501020	South Dundas	230	...	240	...	M0_4	230	0_4	M
3501030	North Dundas	295	...	310	...	M0_4	295	0_4	M
3501042	North Stormont	200	...	180	...	M0_4	200	0_4	M

There are four new columns added in: `Group`, `POPULATION`, `AGE` and `SEX`.
If we ran the following code:
cancerRates = getRates(casedata, popdata, age*sex)
The result of model will be the predicted coefficients of different age and sex groups:
cancerRates

```
Call:  glm(formula = formula1, family = family, data = newdata)

Coefficients:
  (Intercept)        age70_74        age60_64        age75_79        age80_84        age55_59
     -4.72155         0.18543        -0.38010         0.07425         0.44084        -0.86524
     age45_49        age40_44        age35_39        age30_34        age25_29        age20_24
     -1.99423        -2.76184        -3.64470        -5.13532        -4.74062        -6.79155
     age10_14          age0_4            sexF   age70_74:sexF   age60_64:sexF   age75_79:sexF
     -7.16576        -8.02353        -0.31172        -0.17597         0.01024        -0.02530
age85_89:sexF   age50_54:sexF   age45_49:sexF   age40_44:sexF   age35_39:sexF   age30_34:sexF
     -0.66813         0.42881         0.05375         0.81430         0.72865         1.25281
age15_19:sexF     age5_9:sexF   age10_14:sexF     age0_4:sexF
```

```
           NA              NA              NA              NA

Degrees of Freedom: 30 Total (i.e. Null);  0 Residual
Null Deviance:      38660
Residual Deviance: 3.535e-13    AIC: 278.9
```

3 Summary and Future Development

4 Bibliography