Estimating the Relative Index of Inequality in *R*

Jamie C. Sergeant
Nuffield College, University of Oxford, UK

jamie.sergeant@nuffield.oxford.ac.uk

January 6, 2005

1 Introduction

This paper is a brief guide to help users exploit the *R* add-on package `RII`. It is not designed to provide an in depth discussion of relative index of inequality estimation, nor to provide arguments in favour of using one method of estimation over others. For more detail on the estimation method implemented in the package and described in this paper, and for more detail on relative index of inequality estimation in general, see Sergeant & Firth (2004).

The relative index of inequality (RII) is used to compare rates of incidence, usually of death or disease, between those with lowest and highest socio-economic status. Suppose that every individual in some population of interest has a socio-economic rank x, scaled to range between 0 (the lowest) and 1 (the highest) and that the rate of incidence of the outcome of interest (per unit exposure) is $f(x)$ for individuals of social rank x. The RII is defined as $f(0)/f(1)$, the ratio of incidence rates for the (often notional) pair of individuals at the very bottom and top of the socio-economic scale. In practice $f(x)$ is unknown and must be estimated from available data, where individuals are typically categorized into k ordered social classes so that x is interval-censored.

2 The Model

As well as categorized into k social classes, the population under study may also be partitioned into l groups which represent different levels of some standardizing variable. For example, the standardizing variable could be age and the l groups could be the age of the individuals under study in, say, five or ten year intervals. Without loss of generality, assume that age is present as a standardizing variable. Also, for ease of exposition, assume that the outcome of interest is death.

Suppose that the amount of exposure in age group j within social class i is t_{ij}, and that d_{ij} deaths are observed in this class/group intersection during the period under study. Here t_{ij} could represent, for example, the number of person-years at risk, the mid-study period population or the number of individuals at risk at the start of the study period. The death rate for an individual of social rank x and in age group j is modelled as $f(x)\exp(\beta_j)$, allowing the death rate to vary multiplicatively between age groups. With this development the RII is still meaningfully defined as $f(0)/f(1)$; the ratio of death rates for individuals in the same age group and at opposite ends of the social scale. Setting $\beta_1 \equiv 0$ gives a baseline group for comparative purposes.

The number of deaths in each class/age combination is modelled as a Poisson random variable with mean equal to the average value of the age-specific death rate on that class. An estimate of $f(x)$ as a natural cubic spline is obtained by maximizing a penalized log likelihood arising from this Poisson formulation, with a smoothing parameter, $\lambda \geq 0$, controlling how severely roughness in the estimate is penalized. Maximization of the penalized log likelihood takes place over the k coefficients that specify the spline $f(x)$ and the $l-1$ age parameters $\beta_2, \ldots, \beta_l$.

2.1 Example

The dataset `LSDeaths` included with the `RII` package is taken from Sergeant & Firth (2004) and contains data from the UK Office for National Statistics Longitudinal Study (LS). It is a dataframe with 24 rows, giving the number of males dead at the end of observation and at risk at the start of observation in six social classes and four age groups for the observation period 1996 to 2000:

```
> library(RII)
> data(LSDeaths)
> LSDeaths

   class   age Deaths AtRisk
1      V 25-34     19   1956
2     IV 25-34     36   5754
3   IIIM 25-34     66  12189
4   IIIN 25-34     21   4169
5     II 25-34     36   9778
6      I 25-34      6   2679
7      V 35-44     27   1379
8     IV 35-44     62   4515
9   IIIM 35-44    143  11160
10  IIIN 35-44     43   3139
11    II 35-44    125  11671
12     I 35-44     15   2644
13     V 45-54     72   1377
14    IV 45-54    166   4084
15  IIIM 45-54    387   9530
16  IIIN 45-54     95   2456
17    II 45-54    253   9114
18     I 45-54     59   2095
19     V 55-64    164   1289
20    IV 55-64    492   3851
21  IIIM 55-64    752   7410
22  IIIN 55-64    219   2240
23    II 55-64    504   6341
24     I 55-64     77   1397
```

Social class ranges from V (the lowest) to I (the highest) and was recorded, together with age, at the 1991 UK census. To estimate the RII for these data it is first necessary to produce cross-tabulations by social class and age:

```
> LSdead <- xtabs(Deaths ~ class + age, data = LSDeaths)
> LSatrisk <- xtabs(AtRisk ~ class + age, data = LSDeaths)
> LSdead

      age
class  25-34 35-44 45-54 55-64
  V       19    27    72   164
  IV      36    62   166   492
  IIIM    66   143   387   752
  IIIN    21    43    95   219
  II      36   125   253   504
  I        6    15    59    77

> LSatrisk

      age
class  25-34 35-44 45-54 55-64
  V     1956  1379  1377  1289
  IV    5754  4515  4084  3851
  IIIM 12189 11160  9530  7410
  IIIN  4169  3139  2456  2240
  II    9778 11671  9114  6341
  I     2679  2644  2095  1397
```

Now estimate the RII using a value of the smoothing parameter of 1 with the function RII.

```
> LSmodel1 <- RII(LSdead, LSatrisk, loglambda = 0)
> LSmodel1

RII estimate: 2.4654

Group effects:
 25-34  35-44  45-54  55-64
0.0000 0.8988 1.9836 2.9625
```

Note that the argument `loglambda = 0` rather than `loglambda = 1` is supplied as `loglambda` is the log of the value of the smoothing parameter λ. To fit the model with a smoothing parameter of 0, specify `loglambda = -Inf`:

```
> LSmodel2 <- RII(LSdead, LSatrisk, loglambda = -Inf)
> LSmodel2

RII estimate: 2.4653

Group effects:
 25-34  35-44  45-54  55-64
0.0000 0.8988 1.9836 2.9625
```

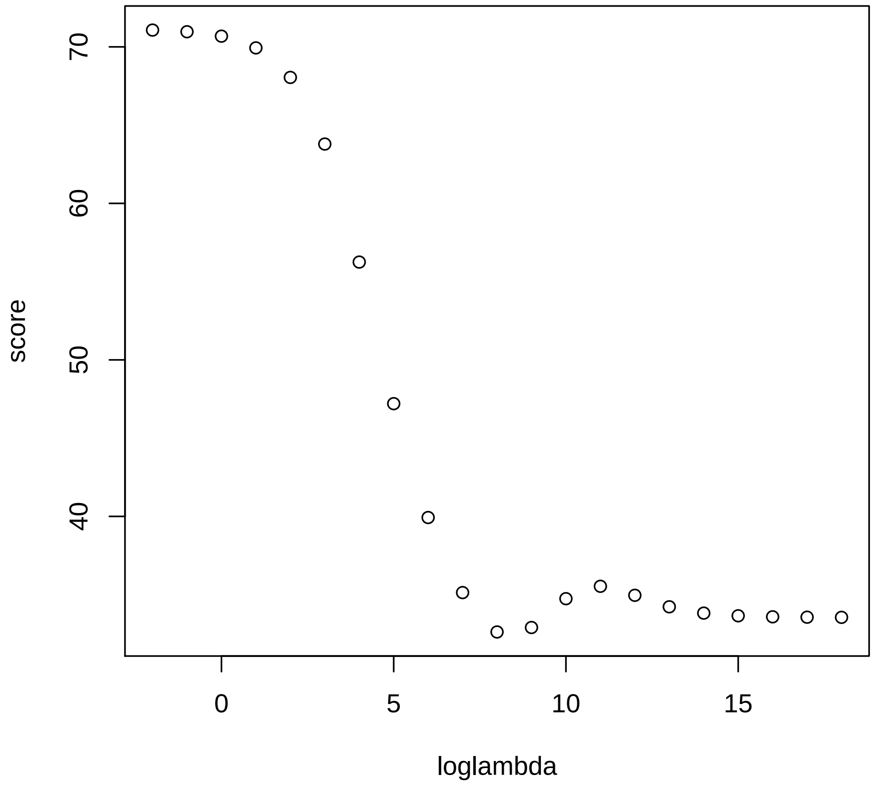
Setting `loglambda = Inf` will induce a linear fit and hence the component `par` of the model, instead of being a vector of k spline coefficients, will be a vector of length two giving the intercept and gradient of the linear fit:

```
> LSmodel3 <- RII(LSdead, LSatrisk, loglambda = Inf)
> LSmodel3$par

   intercept     gradient
 0.006785758 -0.003593981
```

3 Choosing the smoothing parameter

So far it has been assumed that a value of the smoothing parameter has been provided. In practice it is rarely obvious what value to use and so the function RII includes a data-driven mechanism for choosing a single 'optimum' value. This value is chosen by cross-validation. However, for the function to be able to find the value of λ which minimizes the cross-validation score, it must be supplied with a region in which to begin the search. This is where the function `RII.CVplot` comes in. Given a vector of values of $\log(\lambda)$, `RII.CVplot` will evaluate the cross-validation score at each value and plot the results. For example, with the `LSDeaths` data it appears that the value of $\log(\lambda)$ which minimizes the cross-validation score is in $[5, 10]$:

```
> RII.CVplot(LSdead, LSatrisk, loglambda = seq(-2, 18, len = 21))
```



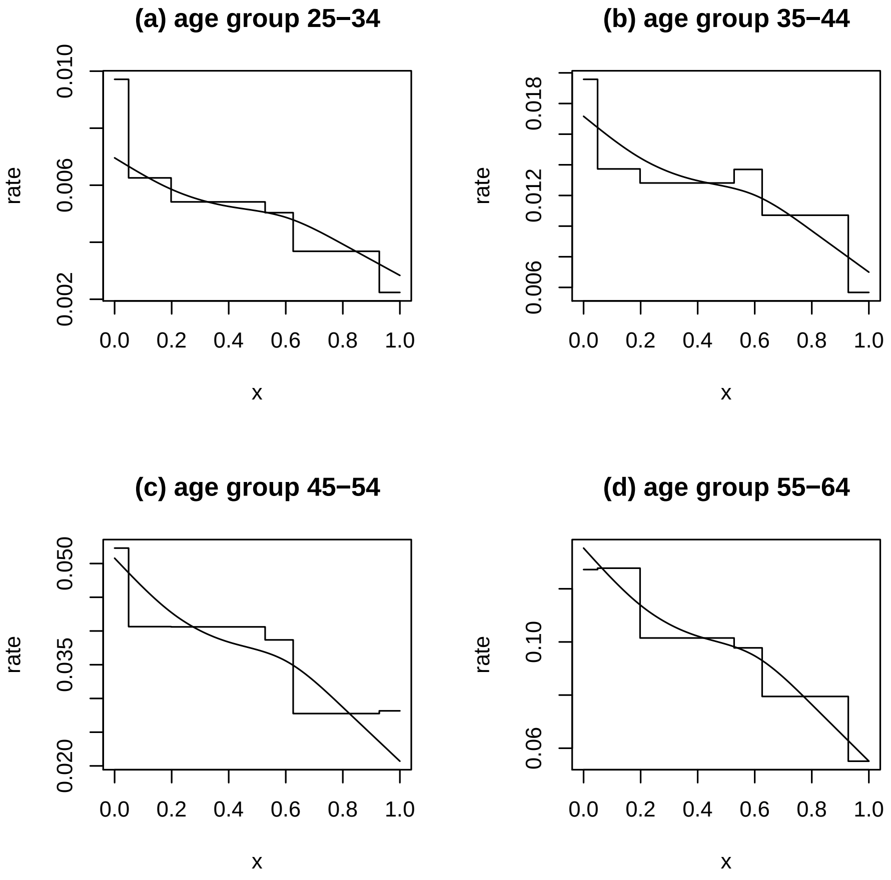
Hence, in light of this, a suitable value of the argument `grid` can be supplied to `RII` to localize the search for an optimum λ. `RII` takes the element of `grid` which produces the smallest value of the cross-validation score as the starting value for optimization over $\log(\lambda)$. The best value of $\log(\lambda)$ found is returned as the component `loglambda` of the fitted model:

```
> LSmodel4 <- RII(LSdead, LSatrisk, grid = seq(5, 10, len = 6))
> LSmodel4$loglambda

[1] 8.383986

> LSmodel4

RII estimate: 2.4537

Group effects:
 25-34  35-44  45-54  55-64
0.0000 0.9029 1.9877 2.9675
```

4 Viewing results

A method for the generic function `plot()` is provided for objects of class `RII`. For a specified age group, the empirical death rate is plotted together with the fitted rate. The standard graphical parameters can be exploited to produce attractive plots that illustrate the model in each age group:

```
> par(mfrow = c(2, 2))
> plot(LSmodel4, group = "25-34", main = "(a) age group 25-34")
> plot(LSmodel4, group = "35-44", main = "(b) age group 35-44")
> plot(LSmodel4, group = "45-54", main = "(c) age group 45-54")
> plot(LSmodel4, group = "55-64", main = "(d) age group 55-64")
```


(a) age group 25–34

(b) age group 35–44

(c) age group 45–54

(d) age group 55–64

The age parameters $\beta_1, \ldots, \beta_l$ are the component `group.effects` of the model. Taking their exponential gives the fitted death rate for each group relative to the first group (recall that $\beta_1 \equiv 0$):

```
> exp(LSmodel4$group.effects)

    25-34     35-44     45-54     55-64 
 1.000000  2.466781  7.298452 19.443572 
```

5 Standard errors

The `var` and `var.log` components of the fitted model give delta method approximations of the variance of the RII estimate and the variance of the log of the RII estimate respectively. However, these estimates are produced by ignoring the roughness penalty part of the penalized log likelihood and so should be treated as first approximations only. A better way to produce a standard error is by bootstrap methods. With the argument `se = TRUE` a bootstrap estimate of the standard error in the log of RII estimate is produced using `B` bootstrap samples. This can then be compared with the 'rough and ready' delta method approximation:

```
> LSmodel5 <- RII(LSdead, LSatrisk, loglambda = LSmodel4$loglambda,
+     se = TRUE, B = 1000)
> LSmodel5$se

[1] 0.1009219
```

```
> sqrt(LSmodel5$var.log)

[1] 0.1444773
```

Note that the specified `loglambda`, in this case the optimum value for the `LSDeaths` dataset, is used with each bootstrap dataset. This is unsatisfactory. Truer to the idea of the bootstrap is to search for a different optimum smoothing parameter for each dataset, which is done by supplying the argument `grid` rather than `loglambda` to `RII`. However, choosing a single `grid` suitable for all of the B bootstrap datasets plus the original data is not straightforward. Such a `grid` should allow the global minimum of the cross-validation score to be found for each dataset. With a satisfactory `grid` supplied, producing the standard error estimate with only a moderate value of `B`, e.g. 100 or 1000, can take a very long time.

References

Sergeant, J. C. & Firth, D. (2004). Relative index of inequality: definition, estimation and inference. In Preparation.