The Stata Journal (2019)
19, Number 3, pp. 645–666 DOI: 10.1177/1536867X19874240

intcount: A command for fitting count-data models from interval data

Stephen Pudney
Health Economics and Decision Science
School of Health and Related Research
University of Sheffield
Sheffield, UK
steve.pudney@sheffield.ac.uk

Abstract. In this article, I describe a community-contributed command, `intcount`, that fits one of several regression models for count data observed in interval form. The models available are Poisson, negative binomial, and binomial, and they can be fit in standard or zero-inflated form. I illustrate the command with an application to analysis of data from the UK Understanding Society survey on the demand for healthcare services.

Keywords: st0571, intcount, count data, interval data, zero-inflated, interpolation, Understanding Society

1 Introduction

Many survey variables are naturally nonnegative integer-valued counts, for example, the number of times an action or event has occurred within a given observation period. Count-data regression models based on distributions, such as the Poisson and negative binomial models, are widely used to analyze these variables.

But complications arise when survey questions are not designed to reveal the count exactly. Survey designers sometimes argue that questions may yield more reliable (albeit less detailed) data if they ask the respondent to place the count within one of a number of prespecified intervals, rather than to report a specific figure.

Interval observation of count data causes difficulty in the estimation of count-data regressions, because most available software requires the count to be observed exactly. Therefore, there is a need for estimation procedures that can account for coarse interval observation.[1] Furthermore, many types of descriptive or policy analysis require exact rather than interval counts, so some form of imputation or interpolation is required.

In this article, I describe a new command for interval estimation of a number of count-data models, and I report results from an illustrative application. Section 2 sets out the estimation approach and the range of available models. Section 3 details the

1. A Stata command, `intreg`, already exists for interval estimation of the regression model for a continuous dependent variable such as income, so `intcount` serves to widen the range of models for which interval estimation is possible. Note, however, that `incount` has more prediction and interpolation options than `intreg`.

syntax of `intcount` and the linked `predict` command that can be used for various types of postestimation imputation. Section 4 presents an application to healthcare data from the UK Understanding Society survey. Section 5 concludes.

2 Interval-observed count-data models

2.1 Basic setup

Let $Y_i \geq 0$ be the ith observation on a dependent variable that takes nonnegative integer values. Y_i may be bounded or unbounded. However, our observations are not on Y_i itself but rather an interval in which Y_i lies. Consequently, we have two observed dependent variables, $[L_i, U_i]$, with the property that

$$L_i \leq Y_i \leq U_i$$

The numerical values of the interval bounds $[L_i, U_i]$ vary across observations, but they are assumed to be observed and strictly exogenous. The two bounds may be equal for some observations where Y_i is fully observed, and, for unbounded distributions like the Poisson and negative binomial, the upper bound U_i may be infinite for some observations.

A set of explanatory covariates appears in a vector $\boldsymbol{X}_i$, and we assume a known parametric form for the discrete conditional probability function $f(\cdot)$ and corresponding distribution function $F(\cdot)$, defined for any nonnegative integer y:

$$\begin{aligned} \Pr(Y_i = y|\boldsymbol{X}_i) &= f(y|\boldsymbol{X}_i) \\ \Pr(Y_i \leq y|\boldsymbol{X}_i) &= F(y|\boldsymbol{X}_i) \end{aligned}$$

The conditional probability of observing the event $L_i \leq Y_i \leq U_i$ is

$$\begin{aligned} \Pr(L_i \leq Y_i \leq U_i|\boldsymbol{X}_i) &= F(U_i|\boldsymbol{X}_i) - F(L_i - 1|\boldsymbol{X}_i) \\ &= \sum_{y=L_i}^{U_i} f(y|\boldsymbol{X}_i) \end{aligned} \tag{1}$$

where $F(L_i - 1|\boldsymbol{X}_i)$ is understood to be zero for $L_i = 0$.

2.2 Alternative base distributions

The model is completed by specifying a parameterized functional form for the distribution function $F(\cdot|\boldsymbol{X}_i)$. The command offers nine possibilities formed from three alternative base models and three options for zero inflation. If we leave aside the possibility of zero inflation, the available models for $F(\cdot|\boldsymbol{X}_i)$ are as follows:

The Poisson model is

$$f(y|\boldsymbol{X}_i) = e^{-\lambda_i}\lambda_i^y/y! \tag{2}$$

where λ_i is the conditional mean function $E(Y_i|\boldsymbol{X}_i)$ parameterized as $e^{X_i\boldsymbol{\beta}}$. The conditional mean and variance of the count variable are both equal to λ_i.

The binomial model is

$$f(y|\boldsymbol{X}_i) = \left(\begin{array}{c} M_i \\ y \end{array} \right) p_i^y (1-p_i)^{M_i - y} \tag{3}$$

where M_i is the known maximum possible value, which may vary exogenously across observations, and p_i is the binomial probability, parameterized as $p_i = (1 - e^{-X_i\boldsymbol{\beta}})^{-1}$. The conditional mean function is $E(Y_i|\boldsymbol{X}_i) = M_i p_i$. This specification may be appropriate when there is a natural upper limit to survey responses (for example, to the question "on how many days last month did you use cannabis?").

The negative binomial model is derivable as the Poisson-gamma mixture

$$y \mid \nu \sim \text{Poisson}(\lambda_i \nu) \qquad \nu \sim \text{gamma}\left(\frac{1}{\alpha}, \alpha\right)$$

where $\lambda_i = e^{X_i\boldsymbol{\beta}}, \alpha > 0$. This gives a distribution for y with mean λ_i and variance $1 + \alpha\lambda_i$. Note that, in the terminology of Cameron and Trivedi (2013), this is the NB2 parameterization of the negative binomial regression model and is consistent with the specification implemented in the Stata `zinb` command. The ML estimation procedure treats $\ln \alpha$ as an unrestricted constant parameter.

2.3 Zero inflation

In some count-data applications, standard forms like the binomial, Poisson, and negative binomial are found to understate the frequency of zero counts. One way of dealing with this is to use a double hurdle or mixture process, where some individuals have a degenerate zero count with probability 1, while others have a count drawn from a standard distribution such as the Poisson.

Let the conditional probability of a degenerate zero be given by the linear index model

$$\Pr(\text{degenerate } 0|\boldsymbol{X}_i) = \pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma})$$

where $\boldsymbol{X}_{i1}$ is a subvector of $\boldsymbol{X}_i$. The distribution of Y among the nondegenerate population is $g(y|\boldsymbol{X}_{i2}\boldsymbol{\beta})$, where $\boldsymbol{X}_{i2}$ is another subvector of $\boldsymbol{X}_i$. Then the mixture distribution of Y is

$$f(y|\boldsymbol{X}_i) = \begin{cases} \pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma}) + \{1 - \pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma})\} g(0|\boldsymbol{X}_{i2}\boldsymbol{\beta}) & \text{if } y = 0 \\ \{1 - \pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma})\} g(y|\boldsymbol{X}_{i2}\boldsymbol{\beta}) & \text{if } y > 0 \end{cases}$$

The probability of the observed interval $[L_i, U_i]$ is again given by (1).

The `intcount` command offers three options for zero inflation:

standard model: $\pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma}) = 0$

logit: $\pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma}) = \{1 + \exp(-\boldsymbol{X}_{i1}\boldsymbol{\gamma})\}^{-1}$

probit: $\pi(\boldsymbol{X}_{i1}\boldsymbol{\gamma}) = \Phi(\boldsymbol{X}_{i1}\boldsymbol{\gamma})$

In practice, estimates of the logit and probit variants are usually almost identical apart from scaling of the $\boldsymbol{\gamma}$ coefficients, which are larger by a factor of approximately $\pi/\sqrt{3}$.

2.4 Estimation

Estimation is by maximum likelihood (ML), with probabilities of the form (1) used to construct the log-likelihood function. By default, numerical optimization of the log likelihood is carried out using Stata's modified Newton–Raphson optimizer; other algorithms can be substituted if you have difficulty in obtaining convergence (see StataCorp [2017, 639–686] for details). Optimization is based on the `lf0` evaluator, so log-likelihood derivatives are approximated by finite differences.

Experience to date suggests that this works well in most cases. Difficulties are most likely to be encountered with overspecified models involving zero inflation that is not required by the data, in which case one or more parameters in the coefficient vector $\boldsymbol{\gamma}$ will explode. Similar convergence difficulties may be found also in zero-inflated specifications where zero inflation is required empirically for a group with certain values for the variables $\boldsymbol{X}_{i2}$ but not for other sample groups. These convergence problems are usually easy to spot, and the required model respecification is obvious.

Occasionally (usually in the more heavily parameterized zero-inflated specifications), the optimizer reaches a difficult region with almost flat likelihood or discontinuous approximate derivatives. Often, these problems can be resolved by passing down the estimates from a simpler specification as starting values for the optimization—for example, a model without zero inflation or with constant zero inflation or a Poisson model as a simpler alternative to the negative binomial.

2.5 Prediction and imputation

The estimates provided by `intcount` may often be useful for imputation, and the `predict` command available with `intcount` offers options. Particularly useful options are the interval-conditional mean predictor $Y_i^* = E(Y_i|L_i \leq Y_i \leq U_i, \boldsymbol{X}_i)$ and the interval-conditional random draw, Y_i^+, which is a realization of the distribution of $Y_i|L_i \leq Y_i \leq U_i, \boldsymbol{X}_i$. Two common situations illustrate their use.

One is where we would like to use the unobserved variable Y_i as a covariate in another model—for example, a regression of some dependent variable W_i on Y_i and $\boldsymbol{X}_i$. But Y_i is unobserved, and we know only that it lies within an interval $[L_i, U_i]$. Then `intcount` can be used to fit a count-data model for Y_i on $\boldsymbol{X}_i$ and compute the interval-conditional

mean predictor Y_i^*. The use of Y_i^* as a proxy for Y_i introduces an imputation error proportional to $(Y_i - Y_i^*)$ into the regression residual term, but it is straightforward to show that $E\{(Y_i - Y_i^*)|Y_i^*, \boldsymbol{X}_i\} = 0$, so the residual is orthogonal to the constructed proxy for Y_i, and the regression of W_i on $Y_i, \boldsymbol{X}_i$ therefore gives unbiased coefficients under standard classical assumptions (provided the count-data model for $Y_i|\boldsymbol{X}_i$ is well specified). This is a better solution to the imputation problem than the common practice of using interval midpoints. However, it can be improved further by making random draws Y_i^+ and using single or multiple imputation.[2]

Another common application is where exact values for Y are needed within some complex policy simulation. Again, multiple random draws Y_i^+ can be used in place of the unobserved Y_i, and the policy calculations averaged across replications. The healthcare cost analysis by Davillas and Pudney (2019) is an example of this.

3 The intcount command

3.1 Syntax

```
intcount depvar1 depvar2 [indepvars] [if] [in] [weight]
    [, [poisson|binomial(#|varname)|negbin]
    inflate(varlist|_cons[, offset(varname) noconstant]) noconstant probit
    offset(varname) exposure(varname) from(matname) difficult]
```

3.2 Description

`intcount` is a community-contributed command that fits a range of count-data models when some of or all the observations on the dependent variable are intervals containing the count, rather than the count itself. The models are based on Poisson, binomial, or negative binomial distributions, possibly with zero inflation. It thus covers some of the same ground as existing Stata commands `poisson`, `nbreg`, `binreg`, `zip`, and `zinbreg` but allows for interval-form data.

depvar1 and *depvar2* are variables that specify the upper and lower limits L_i and U_i of the interval containing the unobserved true count Y_i. The covariates $\boldsymbol{X}_{i1}$ for the core Poisson, binomial, or negative binomial model are specified in *indepvars*; an intercept will automatically be included unless the `noconstant` option is used.

3.3 Output

`intcount` returns ML estimates of the parameters of a count-data model, allowing for the possibility that some of or all the observations on the dependent variable have the form of an interval containing the count, rather than the count itself.

2. See Manski and Tamer (2002) for a much fuller and more general discussion of inference from interval data.

3.4 Options

`poisson`, the default, specifies the Poisson base model defined by (2).

`binomial(`*#* | *varname*`)` specifies the binomial model (3). If the count limit M_i is constant across observations, *#* gives that fixed positive number; otherwise, *varname* specifies a variable containing M_i.

`negbin` specifies the negative binomial model.

At most, one of the options `poisson`, `binomial()`, or `negbin` may appear.

`inflate(`*varlist* | `_cons`[`, offset(`*varlist*`) noconstant`]`)` specifies the variables $\boldsymbol{X}_{i2}$ used as covariates in the zero-inflation model (if any). If `inflate()` is omitted, zero inflation is not used, and a standard count-data specification is estimated. If it appears as `inflate(_cons)`, the zero-inflation probability is estimated as a constant. If covariates are specified in *varlist*, an intercept will also be included unless the `noconstant` suboption is used.

`noconstant` suppresses the intercept term in the linear index $\boldsymbol{X}_{i1}\boldsymbol{\beta}$.

`probit` specifies that the zero-inflation model be of probit form. If omitted, the default is logit. The `probit` option may be used only if `inflate()` also appears.

`offset(`*varname*`)` includes *varname* in the model with the coefficient constrained to 1.

`exposure(`*varname*`)` includes ln(*varname*) in the model with the coefficient fixed at 1.

Standard options for controlling the ML optimization procedure can be included, most usefully:

`from(`*matname*`)` specifies the name of a single-row matrix containing user-supplied initial parameter values for the optimization. The column names should take the form `model:`*varname* and `model:_cons` for the coefficients and intercept in the linear index $\boldsymbol{X}_{i1}\boldsymbol{\beta}$ and `inflate:`*varname* and `inflate:_cons` for those in the index $\boldsymbol{X}_{i2}\boldsymbol{\gamma}$ of the zero-inflation mechanism. The column name for the $\ln(\alpha)$ parameter of the negative binomial model should be given as `/lnalpha` if running with Stata 15 or later or `lnalpha:_cons` for version 14 or earlier.[3] The vector may contain irrelevant elements because the vector is passed onto the ML optimizer with the `, skip` modifier.

`difficult` may occasionally help overcome convergence difficulties.

3. This is for consistency with `nbreg` and `zinb`—the column labeling of the $\ln(\alpha)$ parameter in the return vector `e(b)` from the `nbreg` and `zinb` changed between Stata 14 and 15. If a starting value for $\ln(\alpha)$ is supplied with the wrong labeling, it will be ignored by `intcount`.

3.5 predict

`predict` [*type*] *newvar* [*if*] [*in*] [`, n pr(`#|*varname* #|*varname*`)`
`ce(`#|*varname* #|*varname*`) mc(`#|*varname* #|*varname* [`,` *uniformvar*]`)`
`nooffset`]

Description

Following `intcount`, the `predict` command can be used to construct several measures conditional on covariate values, including the expected count, the probability of the count falling in a specified interval, and the expected value of the count, conditional on it lying in a specified interval. One can also generate a random draw of the interval-specific conditional count distribution. These `predict` options are particularly useful for interpolation purposes. The specified type of prediction is returned in *newvar* as a double precision variable.

Options

`n`, the default, gives a prediction of the count conditional only on the covariates.

`pr(`#|*varname* #|*varname*`)` is the predicted probability (conditional on covariate values) that the count lies in the interval defined by lower and upper limits that may each be a fixed number or a variable.

`ce(`#|*varname* #|*varname*`)` is the expectation of the count conditional on the covariates and the event that it lies in the interval defined by the two limits that may be variable or constant.

`mc(`#|*varname* #|*varname* [`,` *uniformvar*]`)` generates a single random draw from the distribution of y conditional on the event that it lies in the interval defined by the two specified limits. If the *uniformvar* option is not used, `intcount` will generate the required pseudo–random numbers itself without resetting the random-number seed. Optionally, the simulation can be controlled completely by passing a variable containing uniform pseudo–random numbers. The `mc()` option is useful for Monte Carlo simulation or imputation applications where distributional characteristics beyond the conditional mean are required.

`nooffset` causes offset or exposure adjustments to be ignored. By default, any offset or exposure adjustment used for estimation will also be incorporated in the predictions of type `pr()`, `ce()`, or `n`.

4 An application to healthcare demand

We apply the `intcount` command to data from wave 7 of the Understanding Society UK panel on the use of healthcare services. The questions distinguish three services:

consultations with a general practitioner (GP), attendance at a hospital outpatient (OP) clinic, and hospital inpatient (IP) stays.[4] The first two dependent variables come from the following survey questions:

> "In the last 12 months, approximately how many times have you talked to, or visited a GP or family doctor about your own health? Please do not include any visits to a hospital."

> "And in the last 12 months, approximately how many times have you attended a hospital or clinic as an out-patient or day patient?"

Responses to these questions are reported as one of five intervals: 0, [1–2], [3–5], [6–10], 11 or more. Figure 1 shows the two empirical distributions.

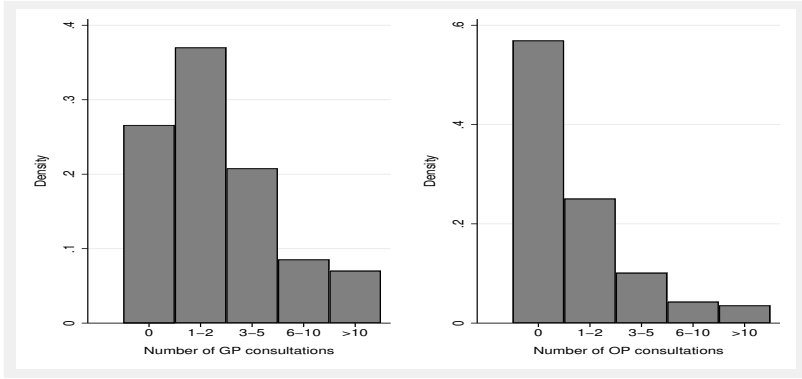

Figure 1. Distributions of the number of GP and OP consultations in the preceding 12 months (UK Household Longitudinal Study [UKHLS wave 7; $n = 6822$])

The third question is

> "In the last 12 months, in all, how many days have you spent in a hospital or clinic as an in-patient?" Answers are given as "exact" integers.

The distribution of responses, shown in the first panel of figure 2 (here plotted over 0–10 days), is typical of count data for rare events. There is a large mode at zero and a highly skewed and dispersed distribution of positive values—the sample maximum is 182 days in this case. This distribution can pose challenging modeling and computational problems. The second panel of figure 2 shows the distribution after we artificially group the responses to conform with the reporting intervals used in the GP and OP questions. Note that ex post grouping should not be assumed to coincide automatically with the answer that would have been provided by the respondent given an interval response scale—respondent behavior may be influenced by question design.

4. The data and a more comprehensive application are discussed in detail in Davillas and Pudney (2019).

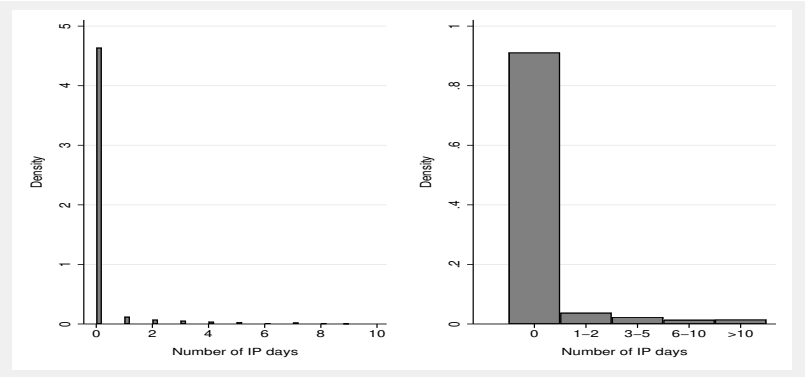

Figure 2. Distribution of the number of days as a hospital inpatient in the preceding 12 months, as observed and after grouping (UKHLS wave 7; $n = 6824$)

4.1 Hospital IP days: The effect of grouping

First, consider the choice of distributional form, using the original exact data. The `intcount` command can accommodate exact count data by setting the upper and lower limit variables equal to the exact count. The resulting estimates reproduce exactly those produced by `poisson` or `zip` for the Poisson model, `binreg` for the binomial model,[5] and `nbreg` or `zinb` for the negative binomial model. The covariates used in these models are simple demographics: a cubic in age `a` (measured in decades from an origin of 50 years), membership of any ethnic minority `nonw`, an indicator for the absence of any educational qualification `noed`, and another for degree-level education `degree`. The following code produced alternative gender-specific models, whose sample fit is summarized in table 1 using the Akaike information criterion (AIC) and Bayesian information criterion (BIC).

5. There appears to be no available Stata command for fitting the zero-inflated binomial model, and `intcount` now fills that gap.

```
. global Xvars "a a2 a3 nonw noed degree"
. //      No zero inflation
. forvalues i=0/1 {
  2.   intcount IP IP $Xvars if male== `i´, poisson vce(robust)
  3.   estat ic
  4.   intcount IP IP $Xvars if male== `i´, binomial(365) vce(robust)
  5.   estat ic
  6.   intcount IP IP $Xvars if male== `i´, negbin vce(robust)
  7.   estat ic
  8. }
. //      With zero inflation
. forvalues i=0/1 {
  2.   intcount IP IP $Xvars if male== `i´, inflate($Xvars)  poisson vce(robust)
  3.   estat ic
  4.   intcount IP IP $Xvars if male== `i´, inflate($Xvars) binomial(365)
> vce(robust)
  5.   estat ic
  6.   intcount IP IP $Xvars if male== `i´, inflate($Xvars) negbin vce(robust)
  7.   estat ic
  8. }
```

It is clear from table 1 that the negative binomial model is far superior in terms of sample fit to the Poisson and binomial models and also that zero inflation improves the fit substantially.

Table 1. AIC and BIC for zero-inflated versions of Poisson, binomial, and negative binomial count-data models, estimated separately by gender from exact data on days spent in hospital

Distributional form	Women		Men	
	AIC	BIC	AIC	BIC
Without zero inflation				
Poisson	91295	91350	71859	71913
Binomial	93874	93929	73634	73687
Negative binomial	21536	21599	13586	13647
With zero inflation				
Poisson	43165	43274	30237	30343
Binomial	45494	45604	31743	31850
Negative binomial	21456	21573	13443	13557

We now investigate the effect of data grouping by refitting the model using the artificially grouped form of the variable whose distribution is shown in figure 2. The code is as follows:

```
. forvalues i=0/1 {
  2.   intcount IP IP $Xvars1 if male== `i´, inflate($Xvars1) negbin vce(robust)
  3.   estimates store exact`i´
  4.   intcount lo_IP hi_IP $Xvars1 if male== `i´, inflate($Xvars1) negbin
> vce(robust)
  5.   estimates store grouped`i´
  6. }
. estout exact0 grouped0 exact1 grouped1, cells(b(star fmt(%7.3f))
> se(par)) starlevels(* .1 ** .05 *** .01) style(tex)
```

Table 2 compares the parameter estimates. There are substantial parameter differences, particularly for the age and education effects in the female sample.

Table 2. Estimates of zero-inflated negative binomial model fit from exact and artificially grouped data

Parameter (std. err.)	Women		Men	
	Exact	Grouped	Exact	Grouped
	Base model parameters			
age§	0.042	0.102*	0.078	0.177**
	(0.117)	(0.061)	(0.096)	(0.076)
age^2	0.057**	0.030**	0.024	0.006
	(0.027)	(0.015)	(0.028)	(0.020)
age^3	−0.001	0.006	0.005	0.001
	(0.014)	(0.008)	(0.011)	(0.008)
Nonwhite	0.155	0.067	−0.350	−0.260
	(0.192)	(0.101)	(0.232)	(0.180)
No education	0.092	0.027	0.203	0.114
	(0.173)	(0.112)	(0.209)	(0.173)
Degree	0.018	−0.229**	−0.753***	−0.660***
	(0.204)	(0.107)	(0.223)	(0.171)
Intercept	−0.426**	0.769***	0.713**	1.219***
	(0.206)	(0.186)	(0.299)	(0.273)
lnalpha	3.072***	1.267***	2.856***	1.601***
	(0.114)	(0.276)	(0.146)	(0.355)

Continued on next page

Parameter (std. err.)	Women		Men	
	Exact	Grouped	Exact	Grouped
	Zero-inflation parameters			
age§	0.899***	0.258***	−0.320***	−0.216***
	(0.169)	(0.054)	(0.094)	(0.051)
age^2	−0.539	−0.022**	−0.094***	−0.041***
	(0.627)	(0.010)	(0.031)	(0.013)
age^3	−0.263	−0.027***	−0.016	−0.003
	(0.170)	(0.007)	(0.010)	(0.006)
Nonwhite	−0.056	−0.052	−0.120	−0.015
	(0.276)	(0.079)	(0.166)	(0.110)
No education	−0.710*	−0.202**	−0.132	−0.106
	(0.393)	(0.086)	(0.171)	(0.107)
Degree	0.337	−0.011	0.034	0.051
	(0.279)	(0.081)	(0.178)	(0.114)
Intercept	−0.341	1.494***	0.695***	1.735***
	(0.377)	(0.214)	(0.248)	(0.272)

NOTES: § Age measured in decades from an origin of 50.
Statistical significance: * = 10%, ** = 5%, *** = 1%

Figure 3 shows the implications of parameter differences for the estimated age profiles, plotting the probability of hospitalization $\Pr(y > 0|\texttt{age})$ against `age` in the range 16–85, with other covariates set to modal zero values. The relevant code is as follows:

```
. preserve
. replace age=.
(42,210 real changes made, 42,210 to missing)
. replace age=_n+15 if _n<=70
(70 real changes made)
. replace a=(age-50)/10
(42,210 real changes made, 42,140 to missing)
. replace a2=a^2
(42,209 real changes made, 42,140 to missing)
. replace a3=a*a2
(42,210 real changes made, 42,140 to missing)
. replace nonw=0
(30,402 real changes made)
. replace noed=0
(14,694 real changes made)
. replace degree=0
(18,067 real changes made)
. generate ll=0 if age<.
(42,147 missing values generated)
. generate uu=0 if age<.
(42,147 missing values generated)
```

```
. forvalues i=0/1 {
  2.   foreach d in exact grouped {
  3.     estimates restore `d´`i´
  4.     predict p`d´`i´ if age<=85,pr(1 .)
  5.   }
  6. }
(results exact0 are active now)
(results grouped0 are active now)
(results exact1 are active now)
(results grouped1 are active now)
. sort age
. twoway line pexact0 pgrouped0 age if age<=85 , lpattern(solid dash)
>   graphregion(fcolor(white) ilcolor(white) icolor(white) lcolor(white))
>   lcolor(black) name(p0, replace) xlabel(20(10)80) ylabel(0(0.05)0.2)
>   xscale(titlegap(3)) yscale(titlegap(3)) xtitle("Woman´s age")
>   legend(col(1) pos(5) ring(0) label(1 "exact")
>   label(2 "grouped")) ytitle("Pr(hospitalization)")
. twoway line pexact0 pgrouped0 age if age<=85 , lpattern(solid dash)
>   graphregion(fcolor(white) ilcolor(white) icolor(white) lcolor(white))
>   lcolor(black) name(p1, replace) xlabel(20(10)80) ylabel(0(0.05)0.2)
>   xscale(titlegap(3)) yscale(titlegap(3)) xtitle("Man´s age")
>   legend(col(1) pos(5) ring(0)label(1 "exact")
>   label(2 "grouped")) ytitle("Pr(hospitalization)")
. graph combine p0 p1
```

The estimated age profiles remain broadly similar after grouping, but they display more variability for the estimates based on exact data, so coarsening the counts to interval form has a smoothing effect in this example.

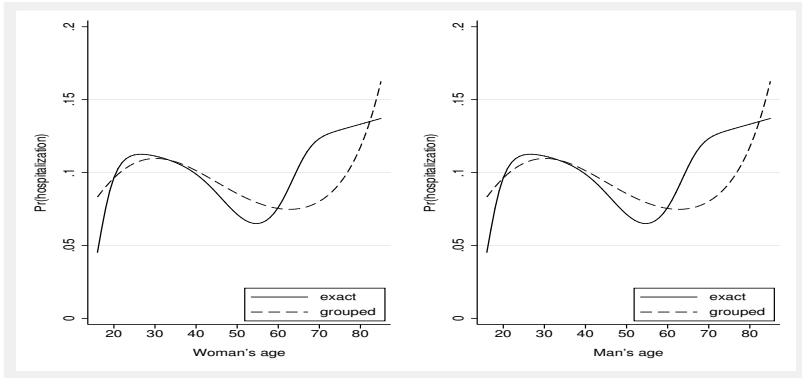

Figure 3. Predicted age profile of zero-count probability by age for ethnic majority woman and man with midlevel education

It is also striking in this application that grouping has a perverse effect on the standard errors. It is clear theoretically that recoding count data to coarser interval form must reduce statistical precision of the parameter estimator for a well-specified count-data model (this is easily confirmed empirically using Monte Carlo simulation by applying `intcount` to simulated counts in exact and grouped form). However, the

anticipated loss of precision may not occur for computed standard errors when the count-data model is misspecified. A poor model may do well in fitting the distribution of responses within broad intervals but much worse in fitting the distribution of exact counts within those intervals. Parameter estimates may be (asymptotically) biased differently for grouped and exact data, and the computed confidence intervals (that are not statistically valid for misspecified models) need not be wider for the interval estimates. This is what we find in table 2, where the interval estimates have robust standard errors that are almost always smaller (and in many cases much smaller).

4.2 Interpolated healthcare measures

The `intcount` command has been designed to be used for interpolation of the underlying count from coarse interval data. We now turn attention to the GP and OP variables, again taking the negative binomial as our basic model but considering both standard and zero-inflated (probit) variants. As covariates, we use dummy variables to allow for gender and ethnicity effects, a cubic in age, and a four-level categorization of educational attainment. Table 3 gives results and also includes estimates of the logit variant for the OP data. Comparison of the fourth and fifth columns of table 3 confirms that the choice between probit and logit specifications makes virtually no difference to the estimates except for scaling of the zero-inflation coefficients (which are larger in absolute value for the logit model by approximately $\sqrt{\pi^2/3} = 1.814$).

Table 3. Estimates of negative binomial models for counts of GP and hospital OP consultations, estimated from grouped data

	GP consultations		Hospital OP consultations		
Parameter (std. err.)	No zero inflation	Probit inflation	No zero inflation	Probit inflation	Logit inflation
	Base model parameters				
age§	0.094***	0.068***	0.168***	0.065***	0.064***
	(0.009)	(0.009)	(0.014)	(0.016)	(0.016)
age^2	0.001	0.001	0.006*	0.003	0.003
	(0.002)	(0.002)	(0.003)	(0.004)	(0.004)
age^3	0.001	0.002**	0.001	0.006***	0.007***
	(0.001)	(0.001)	(0.002)	(0.002)	(0.002)
Male	−0.368***	−0.280***	−0.321***	−0.137***	−0.139***
	(0.015)	(0.016)	(0.023)	(0.027)	(0.027)
Minority	−0.139***	−0.130***	0.046*	0.012	0.016
	(0.017)	(0.018)	(0.027)	(0.031)	(0.031)
GCSE	−0.148***	−0.147***	−0.052	−0.085**	−0.084**
	(0.021)	(0.021)	(0.033)	(0.035)	(0.035)
A-level	−0.268***	−0.271***	−0.183***	−0.159***	−0.158***
	(0.024)	(0.025)	(0.039)	(0.042)	(0.042)

Continued on next page

Parameter (std. err.)	GP consultations: No zero inflation	GP consultations: Probit inflation	Hospital OP consultations: No zero inflation	Hospital OP consultations: Probit inflation	Hospital OP consultations: Logit inflation
Degree	−0.350***	−0.373***	−0.158***	−0.203***	−0.201***
	(0.020)	(0.021)	(0.032)	(0.035)	(0.034)
Intercept	1.525***	1.512***	0.616***	0.704***	0.702***
	(0.022)	(0.022)	(0.036)	(0.040)	(0.040)
$\ln(\alpha)$	0.153***	0.085***	1.146***	0.973***	0.973***
	(0.012)	(0.014)	(0.013)	(0.021)	(0.021)
Zero-inflation parameters					
age§		−0.621***		−0.731***	−1.424***
		(0.108)		(0.130)	(0.261)
age^2		−0.220**		−0.350***	−0.694***
		(0.086)		(0.096)	(0.182)
age^3		−0.024		−0.051**	−0.102***
		(0.021)		(0.021)	(0.038)
Male		4.645		0.730***	1.291***
		(79.355)		(0.080)	(0.154)
Minority		0.163*		−0.107	−0.161
		(0.096)		(0.067)	(0.114)
GCSE		−0.045		−0.218**	−0.367**
		(0.106)		(0.091)	(0.156)
A-level		−0.131		−0.005	0.008
		(0.128)		(0.095)	(0.161)
Degree		−0.421***		−0.254***	−0.435***
		(0.133)		(0.091)	(0.154)
Intercept		−6.221		−1.397***	−2.470***
		(79.355)		(0.131)	(0.256)
AIC	94783	94639	75310	75054	75055
BIC	94867	94799	75394	75214	75215

NOTES: § Age measured in decades from an origin of 50.
Statistical significance: * = 10%, ** = 5%, *** = 1%

We now compare two interpolation methods. If the observed interval is $[L_i, U_i]$, the conditional expectation predictor of the unobserved true count is $E(y|\boldsymbol{X}_i, L_i, U_i)$, and this is specified by the `ce()` option of the `predict` command.[6] The alternative is to generate a random draw from the conditional distribution $f(y|\boldsymbol{X}_i, L_i, U_i)$ using the `mc()` option. The following code generates the interpolations and plots their distributions (for the example of the OP count):

6. Note that we allow `predict` to generate the required random numbers; we could instead have passed down a variable containing uniform pseudo–random numbers.

```
. quietly intcount lo_OP hi_OP $Xvars, negbin inflate($Xvars) probit
. predict OP_ce if e(sample),ce(lo_OP hi_OP)
. predict OP_mc if e(sample),mc(lo_OP hi_OP)
. histogram OP_ce if OP_ce<=30,width(1)
>  graphregion(fcolor(white) ilcolor(white) icolor(white) lcolor(white))
>  name(OPce, replace) xlabel(0(5)30) ytitle("Density")
>  xscale(titlegap(3) range(0 30)) yscale(titlegap(3))
>  xtitle("Conditional mean count")
(bin=22, start=0, width=1)
. histogram OP_mc if OP_mc<=30,width(1)
>  graphregion(fcolor(white) ilcolor(white) icolor(white) lcolor(white))
>  name(OPmc, replace) xlabel(0(5)30) ytitle("Density")
>  xscale(titlegap(3) range(0 30)) yscale(titlegap(3))
>  xtitle("Conditional Monte Carlo count")
(bin=30, start=0, width=1)
. graph combine OPce OPmc
```

The distributions for the interpolated GP and OP counts are shown in figures 4 and 5; the `ce()` interpolator gives a much lumpier distribution than the `mc()` interpolator because it averages out random variation within intervals.

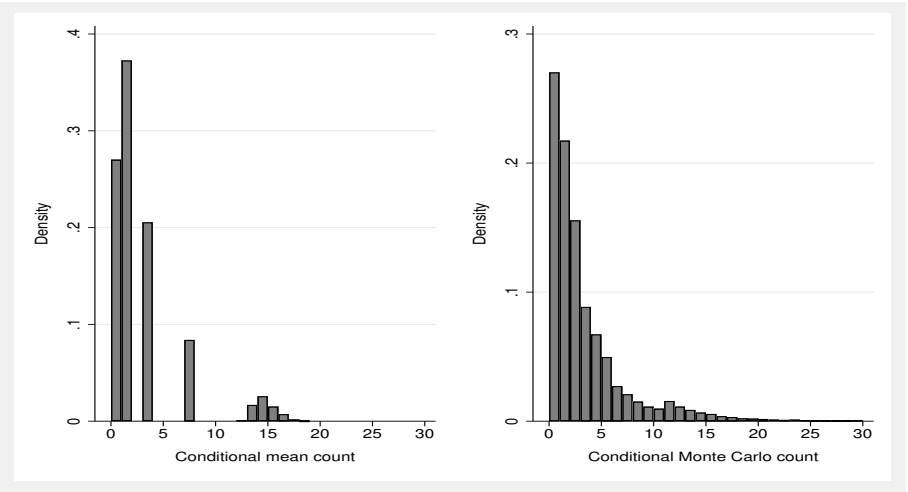

Figure 4. Distributions of GP consultation count with conditional expectation and Monte Carlo interpolation

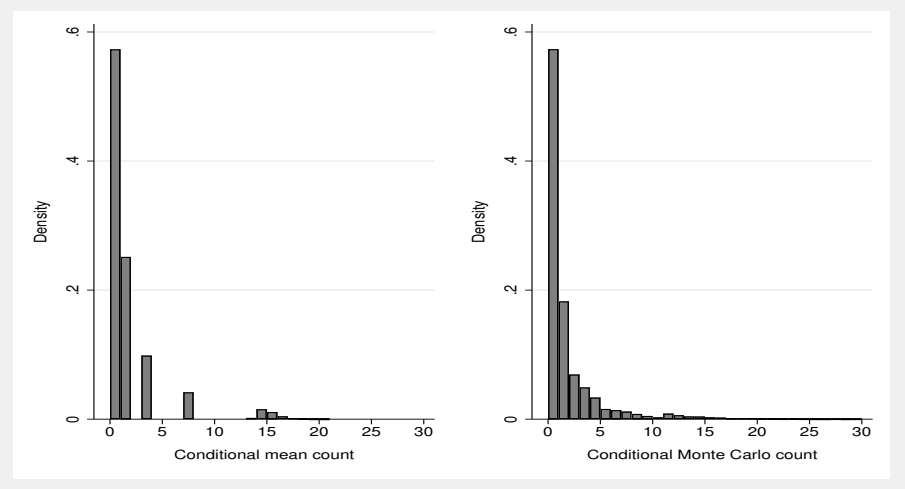

Figure 5. Distributions of OP consultation count with conditional expectation and Monte Carlo interpolation

Use of the `ce()` interpolator understates variance, so if other distributional features besides the conditional mean are of interest, the `mc()` interpolator is usually preferable. The following code produces the means and standard deviations shown in table 4. Within education or gender groups, the mean counts produced by `ce()` and `mc()` are similar (they would be essentially identical if we average many `mc()` interpolations or if there were a large sample within each education group). In contrast, cell-specific sample dispersion clearly confirms the downward bias in variance for the `ce()` interpolator.

```
. matrix mGP=J(8,4,.)
. matrix mOP=mGP
. foreach y in GP OP {
  2.   forvalues m=0/1 {
  3.     forvalues e=0/3 {
  4.       quietly summarize `y´_ce if e(sample)&educ==`e´&male==`m´
  5.       local r=2*`e´+1
  6.       local c=2*`m´+1
  7.       matrix m`y´[`r´,`c´]=r(mean)
  8.       local ++r
  9.       matrix m`y´[`r´,`c´]=r(sd)
 10.       quietly summarize `y´_mc if e(sample)&educ==`e´&male==`m´
 11.       local r=2*`e´+1
 12.       local c=2*`m´+2
 13.       matrix m`y´[`r´,`c´]=r(mean)
 14.       local ++r
 15.       matrix m`y´[`r´,`c´]=r(sd)
 16.     }
 17.   }
 18. }
. matrix m=mGP\mOP
. estout matrix(m, fmt(%5.2f)), style(tex)
```

Table 4. Means and standard deviations of GP and hospital OP consultations interpolated by alternative methods

Education level	Women		Men	
	`ce()`	`mc()`	`ce()`	`mc()`
	GP consultations			
None	4.28	4.31	3.36	3.34
	[4.98]	[5.53]	[4.25]	[4.39]
GCSE	3.38	3.41	2.41	2.42
	[4.13]	[4.51]	[3.42]	[3.59]
A-level	3.06	3.08	1.90	1.91
	[3.64]	[3.88]	[2.77]	[2.96]
Degree	2.80	2.82	1.99	2.01
	[3.44]	[3.65]	[2.73]	[2.88]
	OP consultations			
None	2.04	2.00	1.91	1.86
	[3.79]	[3.94]	[3.67]	[3.83]
GCSE	1.70	1.63	1.36	1.27
	[3.31]	[3.22]	[2.91]	[2.91]
A-level	1.54	1.49	1.03	0.94
	[3.07]	[3.08]	[2.42]	[2.34]
Degree	1.56	1.51	1.16	1.07
	[2.99]	[2.95]	[2.55]	[2.45]

NOTES: Group-specific standard deviations in square brackets.

4.3 Determinants of future healthcare demand

The UKHLS is a perpetual panel, and, in addition to healthcare use in wave 7, we can also observe a range of health measures and other characteristics at the wave 2 baseline. We use this rather than wave 1 as the baseline because a range of objective measurements was made by nurse interviewers at wave 2.

Our analysis dataset covers demographic covariates (age, gender); indicators of socioeconomic status (homeownership, log equivalized household income, education); and biometrics (waist–height ratio, grip strength, resting heart rate, lung function, HDL "good" cholesterol, hypertension). We fit standard negative binomial models from the interval data on GP and OP consultations. The following code produces three variants of the model for each dependent variable, and the parameter estimates are shown in table 5:

```
. global Xdem "male a a2"
. global Xses "h_own ln_income noed degree"
. global Xbio "whr grip pulse htfvc hdl hyper"
. quietly regress lo_GP lo_OP $Xdem $Xses $Xbio
. capture drop insamp
. generate byte insamp=e(sample)
. quietly intcount lo_GP hi_GP $Xdem $Xses if insamp, negbin
. estimates store GP1
. quietly intcount lo_GP hi_GP $Xdem $Xbio if insamp, negbin
. estimates store GP2
. quietly intcount lo_GP hi_GP $Xdem $Xses $Xbio if insamp, negbin
. estimates store GP3
. quietly intcount lo_OP hi_OP $Xdem $Xses if insamp, negbin
. estimates store OP1
. quietly intcount lo_OP hi_OP $Xdem $Xbio if insamp, negbin
. estimates store OP2
. quietly intcount lo_OP hi_OP $Xdem $Xses $Xbio if insamp, negbin
. estimates store OP3
. estout GP1 GP2 GP3 OP1 OP2 OP3, cells(b(star fmt(%7.3f))
> se(par)) starlevels(* .1 ** .05 *** .01) style(tex)
> stats(aic bic, fmt(%7.0f))
```

There is little evidence of a predictive role for socioeconomic status variables when the biometrics are included in the model, so we adopt variant (2), which uses only demographic and biometric covariates. Among the biometrics, only waist–height ratio and grip strength have a consistently significant impact, and the following code uses the `n predict` option to quantify those impacts by computing the mean predicted effect of adding 1 standard deviation to each in turn. The effects are substantial in terms of the potential cost to the public healthcare system: a uniform 1 standard deviation increase in waist–height ratio increases the consultation workload by 15% for GPs and 12% for hospital OP clinics. A similar increase in the grip strength measure is predicted to produce an 11% reduction in GP workloads and a 10% reduction for OP clinics.

```
. foreach c in GP OP {
  2.   foreach x in whr grip {
  3.     capture drop pred*
  4.     estimates restore `c´2
  5.     capture drop tmp
  6.     quietly generate double tmp=`x´
  7.     quietly predict pred0 if insamp,n
  8.     quietly summarize pred0, meanonly
  9.     scalar t0=r(mean)
 10.     quietly replace `x´=`x´+1
 11.     quietly predict pred1 if insamp,n
 12.     quietly summarize pred1, meanonly
 13.     scalar t1=r(mean)
 14.     display in gr "`c´: Impact of 1 sd increase in `x´: " %7.3f (t1-t0)
 15.     display in gr "Proportionate increase: " %5.1f 100*(t1-t0)/t0 "%"
 15.     quietly replace `x´=tmp
 16.   }
 17. }
(results GP2 are active now)
GP: Impact of 1 sd increase in whr:   0.344  ( 15.4%)
(results GP2 are active now)
GP: Impact of 1 sd increase in grip:  -0.246  (-11.0%)
(results OP2 are active now)
OP: Impact of 1 sd increase in whr:   0.156  ( 11.8%)
(results OP2 are active now)
OP: Impact of 1 sd increase in grip:  -0.126  ( -9.5%)
```

Table 5. 5-year-ahead predictive models of healthcare use

	GP consultations			OP consultations		
Coefficient	(1)	(2)	(3)	(1)	(2)	(3)
Male	−0.287***	−0.170**	−0.176**	−0.278***	−0.197*	−0.194
	(0.044)	(0.075)	(0.075)	(0.072)	(0.119)	(0.119)
Age§	0.090***	0.032*	0.035*	0.166***	0.144***	0.146***
	(0.015)	(0.018)	(0.019)	(0.025)	(0.030)	(0.032)
Age squared§	0.022***	0.026***	0.022**	0.034**	0.036**	0.036**
	(0.008)	(0.008)	(0.009)	(0.014)	(0.014)	(0.015)
Homeowner	−0.138**		−0.095	−0.071		−0.023
	(0.062)		(0.062)	(0.101)		(0.103)
ln(income)	−0.103**		−0.051	0.083		0.117*
	(0.041)		(0.042)	(0.068)		(0.069)
No qualification	0.122		0.097	0.069		0.065
	(0.080)		(0.080)	(0.134)		(0.133)
Degree	−0.015		0.006	−0.114		−0.113
	(0.047)		(0.047)	(0.077)		(0.077)
Waist–height ratio		0.143***	0.132***		0.111**	0.114**
		(0.027)	(0.027)		(0.045)	(0.045)
Grip strength		−0.117***	−0.109***		−0.100*	−0.111**
		(0.036)	(0.036)		(0.055)	(0.055)
Pulse rate		−0.010	−0.012		0.028	0.028
		(0.022)	(0.022)		(0.036)	(0.037)
Lung function		−0.039	−0.032		0.034	0.039
		(0.037)	(0.037)		(0.060)	(0.061)

Continued on next page

Coefficient	GP consultations			OP consultations		
	(1)	(2)	(3)	(1)	(2)	(3)
HDL cholesterol		−0.060**	−0.060**		0.016	0.010
		(0.025)	(0.025)		(0.041)	(0.041)
Hypertension		0.096*	0.096*		−0.054	−0.057
		(0.053)	(0.053)		(0.088)	(0.088)
Intercept	1.714***	0.745***	1.205***	−0.241	0.241***	−0.560
	(0.298)	(0.042)	(0.304)	(0.491)	(0.068)	(0.503)
$\ln(\alpha)$	−0.053	−0.084*	−0.087**	1.073***	1.068***	1.065***
	(0.043)	(0.043)	(0.043)	(0.044)	(0.044)	(0.044)
AIC	8866	8811	8811	7279	7276	7280
BIC	8921	8878	8903	7334	7343	7371

NOTES: § Age measured in decades from an origin of 50.
Statistical significance: * = 10%, ** = 5%, *** = 1%

5 Conclusions

Survey count data often come in interval form rather than exact counts. It is common for ad hoc methods to be used for modeling such data—for example, regression applied to midpoint interpolations, or ordered probit regression that does not exploit the known interval limits or the count nature of the data. In this article, I presented a new command, `intcount`, which allows the estimation of a range of count-data regression models from interval data without making arbitrary approximations. The postestimation `predict` command allows the use of the fitted model for many prediction purposes, including interpolation of the unobserved underlying exact count.

I illustrated the use of `intcount` with applications to data from the UK Understanding Society panel on the health service use. These applications demonstrate that interval observation need not be a barrier to econometric analysis.

6 Acknowledgments

I am grateful to Apostolos Davillas for help with preparing data from Understanding Society, which is an initiative funded by the Economic and Social Research Council and various government departments, with scientific leadership by the Institute for Social and Economic Research, University of Essex, and survey delivery by NatCen Social Research and Kantar Public. The research data are distributed by the UK Data Service. This work was supported by the Economic and Social Research Council through the project How can biomarkers and genetics improve our understanding of society and health? (grant ES/M008592/1), the Centre for Micro-Social Change (grant ES/L009153/1), and the Understanding Society study (grant ES/K005146/1). I am extremely grateful to the editors and an anonymous reviewer for comments that have greatly improved the code and its presentation in this article. The views expressed in this article, and any errors or omissions, are mine alone.

7 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 19-3
. net install st0571     (to install program files, if available)
. net get st0571         (to install ancillary files, if available)
```

8 References

Cameron, A. C., and P. K. Trivedi. 2013. *Regression Analysis of Count Data*. 2nd ed. Cambridge: Cambridge University Press.

Davillas, A., and S. Pudney. 2019. Baseline health and public healthcare costs five years on: A predictive analysis using biomarker data in a prospective household panel. Understanding Society Working Paper No. 2019-01, Economic & Social Research Council. https://www.iser.essex.ac.uk/research/publications/working-papers/iser/2019-01.pdf.

Manski, C. F., and E. Tamer. 2002. Inference on regressions with interval data on a regressor or outcome. *Econometrica* 70: 519–546.

StataCorp. 2017. *Stata 15 Mata Reference Manual*. College Station, TX.

About the author

Stephen Pudney is a professor of health econometrics in the Health Economics and Decision Science section in ScHARR, University of Sheffield, UK.