The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- Current Contents/Social and Behavioral Sciences®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)
- Scopus™
- Social Sciences Citation Index®

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, Mata, NetCourse, and Stata Press are registered trademarks of StataCorp LP.

The Stata Journal (2012)
12, Number 1, pp. 45–60

Age–period–cohort models in Stata

Peter D. Sasieni
Centre for Cancer Prevention
Wolfson Institute of Preventive Medicine
Queen Mary, University of London
London, UK
p.sasieni@qmul.ac.uk

Abstract. In this article, I describe and illustrate Stata programs that facilitate i) the fitting of smooth age–period–cohort models to event data and ii) the plotting of observed and fitted rates. The programs include postestimation functionality and flexibility to fit models not possible using Stata's `glm` command. What distinguishes this article from a recent *Stata Journal* article on age–period–cohort models by Rutherford, Lambert, and Thompson (2010, *Stata Journal* 10: 606–627) is that the emphasis here is on extrapolating the model fit to make projections into the future.

Keywords: st0245, apcspline, grmean, age–period–cohort models

1 Background

The study of trends in disease generally involves a mixture of descriptive plots and formal modeling of rates. Many diseases rates (whether of incidence or mortality) vary hugely with age, and for this reason, trends tend to concentrate on either age-specific rates or age-standardized rates. While the general interest is to consider trends over time, there is also interest in seeing whether those trends can best be attributed to trends in the age-specific rates corresponding to different birth cohorts.

For many diseases, the key risk factors have largely been determined before the disease is common, whether that be in utero, in childhood, or in young adulthood. Thus it is common to approximate the rates over disease in the Lexis diagram as a function of age (at the time of the event) combined with a function of period (or the date of the event) and a function of cohort (or the date of birth).

2 Age–period–cohort models

Age–period–cohort (APC) models have long been used in demography and medical statistics to describe the rate of mortality of incidence of a disease as a function of both age and period. The idea is to approximate the rate, which is a function of age and period, by an additive combination of function of age, period, and cohort:

$$\lambda(\text{age}, \text{period}) = g\left\{f_A\left(\text{age}\right) + f_P\left(\text{period}\right) + f_C\left(\text{cohort}\right)\right\}$$

where age = period − cohort. Traditionally, the functions f_A, f_P, and f_C are step-functions and g is the exponential function, but other choices are possible. Much of the literature concerns the linear components of the three functions being nonidentifiable because of the linear dependence between age, period, and cohort.

A good reference for APC models is Clayton and Schifflers (1987).

3 Fitting functions of age, period, and cohort

To fit an APC model in Stata, you would most naturally use the `poisson` command. Provided age, period, and cohort were each coded discretely, you might use the command

```
. poisson events i.age i.period i.cohort, exposure(pyears)
```

This command is fine if you are happy to use step-functions but it lacks flexibility. Additionally, it does not facilitate the visualization of the functions f_A, f_P, and f_C.

3.1 Natural cubic regression splines

The APC model is a special form of a generalized additive model (see Hastie and Tibshirani [1990]). There are many ways to flexibly estimate smooth functions of continuous variables in Stata. A fully flexible APC command in Stata could be written so that the user would specify what sort of smoother should be used for each of the three functions. Instead, we have simply allowed the use of natural regression splines.

A natural spline is the shape that would be taken by a flexible rod forced to pass through a number of points (or knots). The rod will take the form of a cubic spline—cubic between knots with continuous second derivatives at each knot and linear beyond the end knots. The adjective "regression" indicates that we use a limited number of knots and obtain the fit by (nonpenalized) regression. Such splines were made available in Stata in 1994 (Sasieni 1994). The Stata ado-files have been adapted slightly for use in the program described here. The command `mkspline`, which is part of Stata, will also generate a set of variables for cubic splines.

The number of knots determines the flexibility of the spline function. The default values used are arbitrary but were chosen to allow greater flexibility in fitting the age-effects (six interior knots) and to prevent over-fitting of the nonlinear period (five knots) and cohort (three knots) effects. The default gives more knots to the period than to the cohort effects because there tend to be relatively few events at the extremes of the cohort variable, whereas the events are more often approximately uniformly distributed over the period.

A trade-off exists between having the flexibility to capture the salient features of the cohort effect and having a parsimonious model. Even with millions of events, you may wish for a small number of knots. Each additional knot adds an explanatory variable to the regression model. Because the splines are forced to be linear beyond the end knots, a natural cubic spline with no internal knots is simply a straight line (linear function).

Natural cubic splines were used in APC models by Sasieni and Adams (1999, 2000) for drawing inference on the impact of cervical screening on cervical cancer rates. Carstensen (2007) has written about their use more generally. Rutherford et al. (2010) has provided software in Stata for fitting APC models using natural cubic splines.

4 Nonidentifiable

As mentioned previously, there is a very large literature regarding the linear components of the age, period, and cohort functions being not identifiable. This makes no difference in terms of the fitted rates, but it does cause difficulties in interpreting trends. In particular, it is not possible to say whether an increase in age-specific rates over time is due to a period or a cohort effect. Although Stata will identify the collinearity and drop a term from the regression model, this does not help to interpret the individual effects.

One partial solution to the problem is to remove the linear term from both the period and the cohort effect and to relabel it as "drift". This is straightforward when there is no nonlinear term in either period (yielding an age–cohort model) or cohort (yielding an age–period model), but care must be given as to what one means by "removing the linear component" in the general case. To see this, consider $y = x^2$, a function that has no linear component in x. Now suppose that you use $z = x - c$. Then $y = z^2 + 2cz + c^2$, which does include a linear term in z.

The linear cohort constraint could be obtained by $f_C(c0) = f_C(c1)$ for two values of cohort; these might be the minimum and maximum values of the year of birth. But we might want to avoid the extremes of year of birth and use instead $p0 - 75$ and $p1 - 60$, where $p0$ is the first period (year) with observations and $p1$ is the last. Another way to remove the linear component would be to regress each of the variables used to estimate the cohort effect (that is, the basis) on the year of birth and replace each variable by the resulting residual. Here too one might use either a simple regression or a weighted regression, taking into account the number of events at each observation.

When considering the estimated age, period, and cohort effects, an arbitrary constraint must be placed on the constant (it is included in age) and the linear components. We do this by centering period at the mean year (of the observed counts) and by centering cohort at the weighted mean year of birth, with weights proportional to the observed counts at each year of birth.

5 Extrapolation—projection into the future

We might think that the constraint used to make the model identifiable is unimportant because it does not affect the model fit. This is true for the model itself, but the constraints will often affect how we extrapolate the model to project future rates. While it seems sensible to use the cohort effect estimated for someone born in 1955 (say) to project their rates at the age of 70, it is not clear that we should project the trend in

the period effect into the future. A sensible option might be to extrapolate existing cohort effects into the future, allow the drift to continue but attenuate it, and stop all nonlinear period effects. That is, the period effect 10 years into the future would be assumed to be the same as the current period effect (after removing the drift). Similarly, cohorts not yet born (or not yet included in the model) will have the same cohort effect as the youngest birth cohort contributing data to the model. Although this sort of extrapolation seems more reasonable than extrapolation of cubic splines beyond the domain of the observed data, it does require the imposition of arbitrary constraints.

In the `apcspline` command, constraints are imposed by centering period at the mean year (of the observed counts) and by centering cohort at the weighted mean year of birth, with weights proportional to the observed counts at each year of birth. `apcfit` has options that allow the user to specify the centering of each variable.

5.1 Technical note

To prevent unstable extrapolation of cohort effects for future generations, the cohort variable is replaced by $C\text{max} + \log(C - C\text{max} + 1)$ for $C > C\text{max}$, where C is the year of birth and Cmax is the 99th percentile of the weighted distribution of year of birth (weighted by observed counts). (As a result of this transformation, age, period, and cohort are no longer linearly dependent beyond Cmax. Because 1% of the events are born beyond Cmax, this slightly changes the model.)

The exact choice of transformation is arbitrary: we could have chosen Cmax to be some other percentile (such as the 98th) or simply replaced C by Cmax for all $C > C$max; however, in practice it is important to make Cmax less than the maximum year of birth, and it seems reasonable to allow the transform to increase albeit slowly with increasing year of birth. The effect of the transformation can be seen by comparing the cohort effect estimated by `apcspline` with that estimated by `apcfit` using data on colorectal cancer incidence in Great Britain. Although the fitted counts are similar, the cohort effects (illustrated in figure 1) are quite different. In particular, the command `apcfit` suggests that those born in 1995–2000 have five-fold higher risk than those born in 1900–1950.

Because spline basis functions are defined for values of period and cohort for which there are no data, there is no computing issue regarding extrapolation of the model. In practice, however, without additional information, it is impossible to predict the nonlinear cohort and period effects. We might wish to assume that the drift is attenuated over time—it is not reasonable to assume that rates will continue to double every 20 years for the next 200 years!

The default is to assume that the drift between $k-1$ and k years after the last period is multiplied by a damping factor of 0.92^k. This default value is chosen so that the drift after 8 years will be approximately half of what it is during the observation period. Note that the average damping in successive 5-year periods is 22% (after 1–5 years), 48% (6–10 years), 66% (11–15) years, 78% (16–20 years), and 85% (21–25 years). Thus the amount of geometric damping is similar over 1–25 years to the amount of linear damping used by the NordPred and empirically validated by Møller et al. (2003).

As indicated earlier, for the purpose of extrapolation, the nonlinear period effects beyond the last observed period are assumed to be the same as those in the last observed period. Although birth cohort effects carry through into the future as cohorts age, future cohorts are assumed to have the same effect as the last observed cohort. These assumptions are similar to those made by the NordPred R package based on the work of Møller et al. (2003).

6 Link functions for Poisson regression

The `poisson` command in Stata does not provide any flexibility with regard to the link function. If anything other than the canonical logarithmic link function is required, we must use the `glm` command with the `family(poisson)` option. The `glm` command fits a model to the number of events rather than to the rates, and hence the offset/exposure is not handled as we might wish. With link g, we have

$$g(\mu_i) = \text{offset}_i + X_i^{'}\beta$$

or

$$g(\mu_i) = \ln(\text{exposure}_i) + X_i^{'}\beta$$

where μ_i is the expected number of events for the ith observation. Note, however, that the rate is equal to μ/exposure. Hence, we might wish to fit the model

$$g(\mu_i/\text{exposure}_i) = X_i^{'}\beta$$

Unless $g(x) \equiv \ln(x)$, these models are not the same. I have not modified `glm` so that the `exposure()` option (as opposed to the `offset()` option) is handled differently, but I have modified its use within the `apcspline` command so that different link functions can be used on the rates.

The linear predictor can take any real value. If the link is such that the fitted means are not guaranteed to be nonzero, the model may be ill-posed and some form of regularization may be necessary to fit the model. This can be done most easily by adding some number (Z) to the observed counts. The addition of such a background is similar to the use of ridge-regression or a Bayesian prior to smooth the model. Formally,

$$g(\mu_i/\text{exposure}_i + Z_i) = X_i^{'}\beta$$

If Z depends at most on age, it can be viewed as a background number of events based on a background rate common to all observations. Reasonable values for Z will depend on the extent to which the data are divided into small cells (5-year age bands, 1-year age bands, annual data, monthly data, etc.), but choosing Z based on the mean rate seems reasonable.

The `apcspline` command allows the user to specify a background number. A default background is also available, based on the square root of the product of the exposure times a weighted average of the age-specific empirical rate (weight 98) and the overall empirical rate (weight 2). The square root corresponds to the standard deviation of the number of cases.

7 The apcspline command

The `apcspline` command fits an APC model of the form

$$N \sim \text{Poisson}(\mu)$$

$$g\left(\frac{\mu}{\text{exposure}}\right) = f_A(\text{age}) + f_P(\text{period}) + f_C(\text{cohort}) + \beta\text{drift}$$

where g is the link function and f_A, f_P, and f_C are natural cubic splines.

7.1 Syntax

`apcspline` *depvar agevar periodvar* [*if*] [*in*] [*weight*] [`,` `exposure(`*varname_e*`)` `link(`*linkname*`)` `scale(x2|dev|#)` `regularize` `background(`*exp*`)` `damping(#)` `nkage(#)` `nkperiod(#)` `nkcohort(#)`]

The cohort variable is calculated as "period − age", and it is for the user to ensure that both period and age are in the same units. If age is recorded in 5-year groups and period is recorded in single years, then the difference in the value of age for group 40–44 compared with group 35–39 should be 5.

Many of the options available to `glm` or `poisson` related to standard error, maximization, and reporting may also be used.

7.2 Options

`exposure(`*varname_e*`)` specifies the variable relative to which rates are calculated (see the discussion in section 6 above).

`link(`*linkname*`)` specifies the link function (see section 6 above) as in the `glm` command except that the exposure is handled differently for power links. The default is `link(log)`.

`scale(x2|dev|#)` overrides the default scale parameter; see [R] **glm**.

`regularize` specifies that a default number of events be added to each observation before fitting the model. The default is a noninteger variable that is different for each value of age. The variable `regular` is equal to $\sqrt{0.98 \times E_{\text{age}} + E_0/50}$ where E_{age} is the expected count given the age and E_0 is the expected count based on the

overall rate. The background numbers are stored in `_Ibackground` and subtracted back to obtain the fitted values.

`background(`*exp*`)` specifies the background rate of events added to each observation (dependent on age) so as to stabilize the estimates. Regularization can make the model converge after fewer iterations and can even enable a model to fit when it would not otherwise (particularly when using a link function that does not ensure that the rates are positive). The program first tries to generate `_Ibackground=`*exp*. If this generates an error, it tries `_Ibackground=regular` *exp* where `regular` is the variable generated by `regularize`. Thus `background(0.5)` adds 0.5 to every observation, `background(*0.5)` multiplies the default regularization by 0.5, and `background(*1+0.5)` adds 0.5 to the default regularization. Note that the larger the value of `_Ibackground`, the greater the "smoothing".

`damping(#)` specifies the level of geometric shrinking of the drift beyond the last observation point. During the observed period, the linear predictor is increased by `_b[drift]` for each unit increase in the period. For k units of time after the last observation, the linear predictor is increased by `_b[drift]` $\times$ `damping()`^{k} for a unit increase in the drift. By default, `damping()` is set to 0.92 so that 8 years after the last observation, the drift effect is only about half of what it is during the observation period.

Without damping, increasing (or decreasing) trends would be predicted to go on forever. To achieve this effect, use `damping(1)`.

`nkage(#)`, `nkperiod(#)`, and `nkcohort(#)` specify the number of interior knots in the natural cubic splines used for each of the functions f_A, f_P, and f_C, respectively. The default values are 6, 5, and 3, respectively. Specifying `nkperiod(0)` results in the age–cohort model. Specifying `nkcohort(0)` results in the age–period model. Specifying `nkperiod(0)` and `nkcohort(0)` results in the age–drift model.

8 Postestimation commands

Although you can use any of the postestimation commands available after fitting a `glm` model, I urge you to use them with caution. Certain `apcspline`-specific behavior is described here as well as some `apcspline`-specific options for `predict`.

The standard `predict` after `glm` cannot adjust for the `background()`. I have adapted the predictions so that the fitted means are adjusted for the `background()`. I have also introduced an option to obtain the estimated rate, defined as the mean divided by the exposure. By using the option `rate(#)`, we can generate rates per #. Thus, for instance, `rate(1e5)` yields rates per 100,000. Other options of `predict glm` have not been altered. Thus the standard residual will be true, but the Pearson residual will be an estimate of $(O-E)/(E+b)^{1/2}$ where O is the observed count, E is its expectation, and b is the background.

Using `predict`, we can also obtain estimates of f_A, f_P, and f_C by specifying `age`, `period`, and `cohort`, respectively. In fact, the option `period` yields the relative rate as a function of period—that is, $\exp\{f_P(\text{period})\}$—and similarly, cohort yields $\exp\{f_C(\text{cohort})\}$. The option `age` gives the estimated rate as a function of age when the period and cohort relative rates are 1, that is, $\exp\{\_const + f_A(\text{age})\}$. The `rate(#)` option can be used with the `age` option.

Strictly speaking, $\exp\{f_P(\text{period})\}$ would not be a relative risk if the link were not logarithmic. Thus the functions f_P and f_C are difficult to interpret when estimated based on a noncanonical link. For this reason, `predict` fits a canonical model (that is, one with a logarithmic link) to the fitted means of the model specified by `apcspline` whenever a noncanonical link is used. Thus the period and cohort effects estimated using `predict` can always be interpreted as relative risks even if the link function is not logarithmic.

Consider $y1 = \exp\{\ln(5) + x\ln(2)/10\}$ and $y2 = \{5^{0.2} + (10^{0.2} - 5^{0.2})x/10\}^5$. Both functions are equal to 5 and 10 at $x = 0$ and $x = 10$, respectively. At $x = 5$, $y1 = 7.071$ and $y2 = 7.156$, so the functions are almost indistinguishable on (0,10). But at $x = 20$, $y1 = 20$ and $y2 = 18.4$; and at $x = 70$, $y1 = 640$ and $y2 = 177$. Thus for moderate trends, the difference between the logarithmic and the power 0.2 links in terms of fitted values to the observed numbers of events will be minimal, but the impact on long-term extrapolation could be considerable. The ease with which relative risk functions can be interpreted suggests that using these to summarize the drift, period, and cohort effects is sensible even if a different link is used for projections.

9 Example

The data we use to illustrate the `apcspline` command contain the number of cases of colorectal cancer in Great Britain in 5-year age bands for each year from 1975–2007 together with mid-year population estimates for 1975–2007 and population projections until 2030. The numbers of both cancers and population are separated by sex.

```
. use colorectal
(COLORECTAL)
. apcspline cases age year if sex==1, exposure(population)
Iteration 0:   log likelihood =  -17478.96
Iteration 1:   log likelihood = -4186.2951
Iteration 2:   log likelihood = -2774.7726
Iteration 3:   log likelihood = -2537.9356
Iteration 4:   log likelihood = -2487.4614
Iteration 5:   log likelihood = -2480.3359
Iteration 6:   log likelihood = -2480.2576
Iteration 7:   log likelihood = -2480.2575
Poisson regression                              Number of obs   =        594
                                                LR chi2(16)     = 1317571.33
                                                Prob > chi2     =     0.0000
Log likelihood = -2480.2575                     Pseudo R2       =     0.9962
```

_Icases	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	.1319719	.0014587	90.47	0.000	.1291129	.1348309
_IA1	.0005676	.0001326	4.28	0.000	.0003078	.0008274
_IA2	-.0002905	.0000956	-3.04	0.002	-.000478	-.0001031
_IA3	-.0000439	.0000351	-1.25	0.211	-.0001127	.0000248
_IA4	.000068	.0000159	4.27	0.000	.0000367	.0000992
_IA5	-.0000578	8.90e-06	-6.50	0.000	-.0000753	-.0000404
_IA6	.0001032	5.15e-06	20.04	0.000	.0000931	.0001133
_Idrift	.007759	.0001797	43.17	0.000	.0074067	.0081113
_IP1	-1.24e-06	.0000814	-0.02	0.988	-.0001607	.0001582
_IP2	.0000872	.0001064	0.82	0.413	-.0001214	.0002958
_IP3	-.0002902	.0001045	-2.78	0.006	-.0004951	-.0000853
_IP4	.0004373	.0000975	4.49	0.000	.0002462	.0006284
_IP5	-.0002914	.0000703	-4.14	0.000	-.0004292	-.0001536
_IC1	-.000019	3.83e-06	-4.96	0.000	-.0000265	-.0000115
_IC2	-.000013	5.38e-06	-2.42	0.015	-.0000236	-2.48e-06
_IC3	.0000353	3.69e-06	9.56	0.000	.000028	.0000425
_cons	-14.81817	.0841413	-176.11	0.000	-14.98309	-14.65326
ln(popula~n)	1	(exposure)				

```
. predict fit2
(option n assumed; predicted number of events)
. predict f_age, age
. predict f_cog, cohort
```

For comparison, we also use the `apcfit` command (also see figure 1).

```
. apcfit if sex==1 & case!=., cases(cases) period(year) age(age)
> poprisktime(population) cohort(cohort)
Iteration 0:   log likelihood = -36749.643
Iteration 1:   log likelihood = -7098.7844
Iteration 2:   log likelihood = -3332.0452
Iteration 3:   log likelihood = -2680.2755
Iteration 4:   log likelihood = -2555.3381
Iteration 5:   log likelihood = -2545.0924
Iteration 6:   log likelihood = -2544.9979
Iteration 7:   log likelihood = -2544.9978
Generalized linear models                         No. of obs      =       594
Optimization     : ML                             Residual df     =       579
                                                  Scale parameter =         1
Deviance         =  1259.785394                   (1/df) Deviance =  2.175795
Pearson          =  1291.749674                   (1/df) Pearson  =  2.231001
Variance function: V(u) = u                       [Poisson]
Link function    : g(u) = ln(u)                   [Log]
                                                  AIC             =  8.619521
Log likelihood   = -2544.997846                   BIC             = -2438.218
```

cases	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_spA1_intct	-9.682514	.0369489	-262.05	0.000	-9.754932	-9.610095
_spA2	3.750127	.0524424	71.51	0.000	3.647341	3.852912
_spA3	.8812742	.0499142	17.66	0.000	.7834442	.9791042
_spA4	-.00575	.0276824	-0.21	0.835	-.0600065	.0485066
_spA5	.1027369	.0051401	19.99	0.000	.0926624	.1128113
_spA6	.0472322	.0014886	31.73	0.000	.0443146	.0501498
_spP1	.022857	.0015183	15.05	0.000	.0198812	.0258328
_spP2	-.0025955	.0013071	-1.99	0.047	-.0051573	-.0000337
_spP3	.0094303	.0014652	6.44	0.000	.0065586	.012302
_spP4	.0084488	.001389	6.08	0.000	.0057263	.0111712
_spC1_ldrft	.0071056	.0001531	46.42	0.000	.0068056	.0074057
_spC2	-.1573066	.0260045	-6.05	0.000	-.2082744	-.1063388
_spC3	-.1485572	.0170126	-8.73	0.000	-.1819014	-.1152131
_spC4	-.1204548	.017805	-6.77	0.000	-.155352	-.0855577
_spC5	-.0614808	.0115918	-5.30	0.000	-.0842003	-.0387614
ln(popula~n)	1	(exposure)				

```
. predict fitapc
(option mu assumed; predicted mean cases)
(1422 missing values generated)
```

Note that the fitted values from `apcfit` are only available for the observations that were used in the model fitting, whereas `predict` after `apcspline` provides estimated mean numbers for all observations. The fitted values that are provided by both commands are extremely similar.

```
. summarize fi* if case<. & sex==1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
fit2	594	886.3923	1057.698	.0264056	3572.244
fitapc	594	886.3923	1057	.0457746	3532.978

The estimated risks as a function of age are also very similar, but the cohort relative risks are quite different.

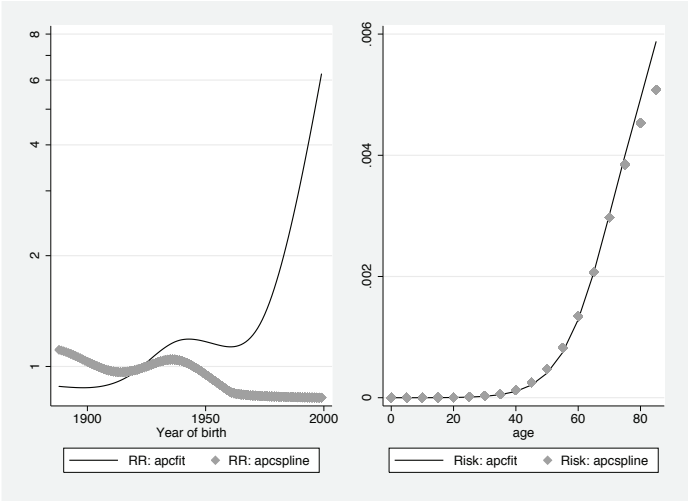

Figure 1. Comparison of the default output from `apcspline` with that from `apcfit`. The left-hand plot shows the estimated cohort effects, which are very different. In particular, in the `apcspline` model fit, the relative risk is always close to 1, whereas the `apcfit` gives an estimate that rises rapidly to beyond 5 for those born at the end of the twentieth century. It should be noted that the constraints imposed by the two programs are different: one could remove the drift from the `apcfit` cohort effect, but its tail behavior would still be quite different from the `apcspline` effect. The right-hand plot compares the age curve from both models. They are seen to be very similar.

The `apcspline` command can also be used to generate the bases for the splines, which can then be combined with other covariates or multiplied to produce interactions within a Poisson or `glm` model. For instance, if we wanted to have the same basic age curve for both sexes but with different period and cohort effects, we might use

```
. apcspline cases age year, exposure(population)
```
(*output omitted*)
```
. poisson cases i.sex _IA* i.sex#c._IC1 i.sex#c._IC2 i.sex#c._IC3
> i.sex#c._Idrift i.sex#c._IP1 i.sex#c._IP2 i.sex#c._IP3 i.sex#c._IP4
> i.sex#c._IP5
```
(*output omitted*)

10 Plotting rates

The command `grmean` allows for plotting of the observed and fitted rates against another variable with separate lines and symbols for different groups. Uniquely, this command has both the option `by()` and the option `over()`. The `by()` option allows different plots for separate subgroups within the same graph. The `over()` option allows plotting of rates for different subgroups within the same axes. The options `by()` and `over()` can be used simultaneously to allow rates for different subgroups on the same axes as well as different axes for a further subgroup.

`grmean` can be used for data that are not rates. The `nomean` option allows for plotting of the actual data rather than weighted mean values. This can be useful if we wish to use the `over()` option (see figure 2).

```
. sysuse auto, clear
(1978 Automobile Data)
. lowess mpg weight, by(foreign) nograph gen(smooth)
. grmean mpg weight, over(foreign) nomean ti("grmean mpg weight,over(foreign)")
> sav(mpg1, replace)
(file mpg1.gph saved)
. grmean mpg smooth weight, over(foreign) nomean
> ti("grmean mpg smooth weight, over(foreign)") sav(mpg2, replace)
(file mpg2.gph saved)
. gr combine mpg1.gph mpg2.gph
```

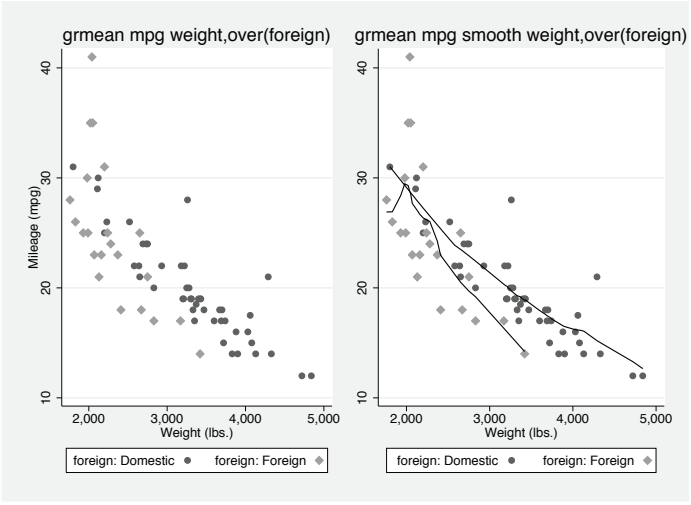

Figure 2. Illustration of the `over()` option of `grmean` (together with the `nomean` option) using `auto.dta`. Foreign and domestic cars are plotted using different symbols on the same axes.

The main application of `grmean` is for calculating and plotting directly standardized rates. The following commands were used to generate fitted rates from APC models.

```
apcspline case age year if sex==1, exposure(pop)
predict fit
apcspline case age year if sex==2, exposure(pop)
predict fit2
replace fit=fit2 if sex==2
generate rate_fit=1e5*fit/population

apcspline case age year if sex==1 & age>15, exposure(pop) link(power .2)
predict fitp
apcspline case age year if sex==2 & age>15, exposure(pop) link(power .2)
predict fitp2
replace fitp=fitp2 if sex==2
generate rate_fitp=1e5*fitp/population
```

Note that we could also use the `ir` option of `predict` to estimate the fitted rates directly:

```
. predict rate_fitp, rate(1e5) ir
. grmean rate rate_fit year, by(sex) over(Age_gp) standard(stpop)
> xlab(1975(10)2025)
```

The above commands would yield figure 3. Here the rates are age-standardized within age group. The observed data are plotted as dots, and the fitted data are joined as lines.

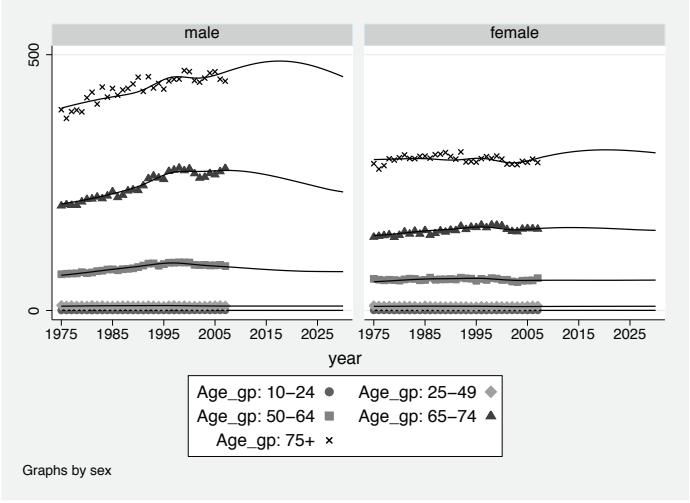

Figure 3. Trends in rates of colorectal cancer. The figure shows trends separately in males and females in different age bands. Within each age band, the rates are age-standardized, but the use of 15-year bands means that they convey information regarding age-specific rates.

The general syntax of `grmean` is

`grmean` *yvar_0* [*yvars*] *xvar* [*if*] [*in*] [`, standard(`*var*`) nomean over(`*varname* [`, total`]`) by(`*varlist* [`,` *byopts*]`) addplot(`*plot*`)` *twoway_options*]

Note that we can use more than one model fit, as shown in figure 4:

```
. grmean rate rate_fit rate_fitp year if sex==1, by(age)
```

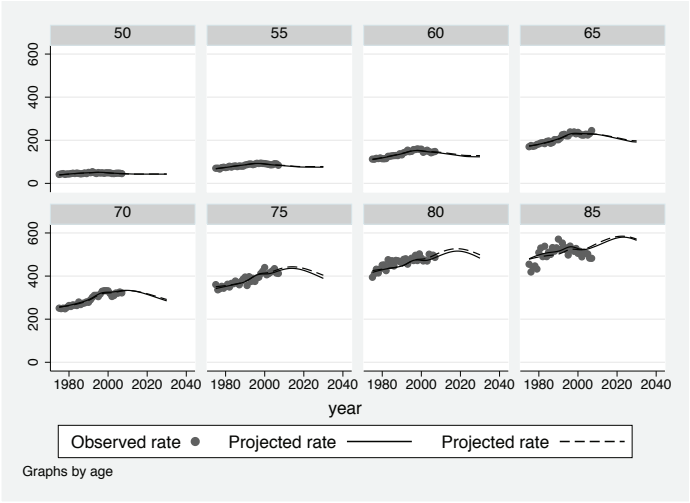

Figure 4. Observed and projected rates by 5-year age group. Both the projections based upon the logarithmic link (solid line) and the projections based on the power 0.2 link (dashed line) are shown. They are almost identical.

We can also do cohort plots (see figure 5):

```
. grmean rate rate_fit cohort if sex==1 & age>40, over(age) legend(row(3))
. grmean rate rate_fit age if sex==1 & age>40 & cohort>=1900 & cohort<1960,
> over(coh_g) legend(row(2)) xlab(45(10)85)
```

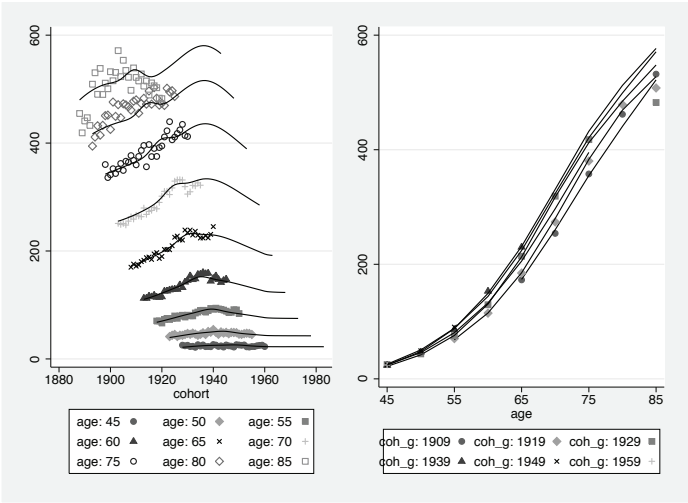

Figure 5. Cohort plots. In the left-hand panel, age-specific rates are plotted against year of birth. In the right-hand panel, rates plotted against age and fitted values corresponding to different 10-year birth cohorts are joined together.

As well as creating purely age-standardized plots, you can do

```
. grmean rate rate_fit year, by(sex) over(country) standard(stpop)
```

However, the command uses directly standardized rates, so

```
. grmean rate rate_fit cohort if sex==1, standard(stpop)
```

may not give the desired results if the age range is cohort dependent. Also the different ages available for the later cohorts for the fitted rates compared with the observed rates will mean that fit may appear to be poor in the above plot even though it is not.

Including the suboption `total` after `over()` allows an overall standardized rate to be plotted (within each level of `by()`).

```
. grmean rate rate_fit year, by(sex) standard(stpop) over(Age_gp, total)
```

11 References

Carstensen, B. 2007. Age–period–cohort models for the Lexis diagram. *Statistics in Medicine* 26: 3018–3045.

Clayton, D., and E. Schifflers. 1987. Models for temporal variation in cancer rates. II: Age–period–cohort models. *Statistics in Medicine* 6: 469–481.

Hastie, T. J., and R. J. Tibshirani. 1990. *Generalized Additive Models*. London: Chapman & Hall/CRC.

Møller, B., H. Fekjær, T. Hakulinen, H. Sigvaldason, H. H. Storm, M. Talbäck, and T. Haldorsen. 2003. Prediction of cancer incidence in the Nordic countries: Empirical comparison of different approaches. *Statistics in Medicine* 22: 2751–2766.

Rutherford, M. J., P. C. Lambert, and J. R. Thompson. 2010. Age–period–cohort modeling. *Stata Journal* 10: 606–627.

Sasieni, P. D. 1994. snp7: Natural cubic splines. *Stata Technical Bulletin* 22: 19–22. Reprinted in *Stata Technical Bulletin Reprints*, vol. 4, pp. 171–174. College Station, TX: Stata Press.

Sasieni, P. D., and J. Adams. 1999. Effect of screening on cervical cancer mortality in England and Wales: Analysis of trends with an age period cohort model. *British Medical Journal* 318: 1244–1245.

———. 2000. Analysis of cervical cancer mortality and incidence data from England and Wales: Evidence of a beneficial effect of screening. *Journal of the Royal Statistical Society, Series A* 163: 191–209.

About the author

Peter Sasieni is a professor of biostatistics and epidemiology in the Wolfson Institute of Preventive Medicine at Queen Mary, University of London. He has been a Stata user for many years and made several contributions to the *Stata Technical Bulletin*.