mkduration: An Easy Way to Create Duration Variables in Binary Cross-Sectional Time Series Data

Andrew Q. Philips[*]

February 17, 2020

Abstract: In cross-sectional time series data with a dichotomous dependent variable (B-CSTS), failing to account for duration dependence when it exists can lead to faulty inferences. A common solution is to include duration dummies, polynomials or splines to proxy for duration dependence. Yet creating these is not easy for the common practitioner. In this manuscript I introduce `mkduration`, a straightforward way to generate a duration variable using B-CSTS data in Stata. This command can handle missing data, and can also output the duration variable in commonly-used formats such as splines or polynomials.

Word Count: (abstract) 91; approximately 3000 (manuscript)
Keywords: binary cross-sectional time series; event history; duration

[*] andrew.philips@colorado.edu. Assistant Professor, Department of Political Science, University of Colorado Boulder, UCB 333, Boulder, CO 80309-0333.

Introduction

It is well-known that when modeling a dichotomous dependent variable in cross-sectional time series data (or B-CSTS), failing to account for duration dependence—the phenomenon by which the occurrence of an event at time t in unit i may make the re-occurrence of an event at future time point more or less likely—can have severe consequences for estimation (Beck, Katz and Tucker 1998). At best, failing to model such dependence may induce serial autocorrelation, leading to standard errors that are anti-conservative. At worst, it can produce omitted variable bias even if the included regressors are unrelated to the omitted duration dependence.

A common approach recommended by Beck, Katz and Tucker (1998) when dealing with B-CSTS data when the occurrence of events are relatively rare is to estimate a logistic regression with duration dummies in order to proxy for any duration dependence.[1] While alternative approaches exist (c.f., Zorn 2000; Box-Steffensmeier and Jones 2004), in general the Beck, Katz and Tucker (1998) approach is very popular due to its simplicity; as of February 2020 this article had over 2800 citations. Despite these suggestions, creating the duration variable from the dichotomous dependent variable is not straightforward. For one, such time-since-last-event variables are not as simple as techniques like including the lag of a series. Moreover, missing data can lead to additional complications since it is unknown whether an event has occurred during this period. And, creating functional forms of duration dependence through tools such as splines is a technique unfamiliar to many social scientists (Carter and Signorino 2010).

In this paper I introduce `mkduration`, an easy way to generate duration variables for B-CSTS data in Stata using a single command. It can also handle missing data, as well as producing several different functional forms of duration commonly used in the literature. In the paragraphs that follow, I first discuss duration dependence in the context of B-CSTS data, then introduce the `mkduration` command. I illustrate the utility of this command through an example using data from Bapat and Zeigler (2016).

[1] In lieu of duration dummies, other recommendations include turning durations into splines or polynomial terms (Beck, Katz and Tucker 1998; Carter and Signorino 2010).

Duration dependence with B-CSTS

Consider a simple B-CSTS dataset in long form, like the one shown in Table 1. y_{it} is a dichotomous dependent variable that does not occur relatively often. This is commonly modeled using a generalized linear model with a logistic link in order to account for the dichotomous nature of the dependent variable (Beck, Katz and Tucker 1998). The problem that arises is with duration dependence, which exists if the $\Pr(y_{it}) = 1$ changes based on how long the last event occurs. This is shown by the duration variable in Table 1, which records the time-to-event in the data.[2] Failing to model duration dependence implies a constant hazard rate, meaning that the probability of event re-occurrence does not change over time. In other words, event occurrences are independent from one another. In real-world data however, such an assumption is probably almost always violated. For instance, duration dependence has been argued to exist in topics as varied as conflict onsets (Clare 2010; Bapat and Zeigler 2016), pursuit of nuclear weapons (Way and Weeks 2014), and firm-level bankruptcies (Hillegeist et al. 2004).

Table 1: Durations in B-CSTS data

Unit	Time	Event (y_{it})	Duration
1	1	0	1
1	2	0	2
1	3	1	3
1	4	0	1
1	5	1	2
2	1	0	1
2	2	1	2
2	3	0	1
2	4	0	2
2	5	0	3
3	1	0	1
3	2	1	2
3	3	1	1
3	4	0	1
3	5	0	2
⋮	⋮	⋮	⋮

[2] Table 1 is an example of left-censored data, which is a reasonable strategy if data are balanced (e.g., all units i enter at time $t = 1$; alternative strategies may be advisable if some units enter into the sample at different times (Beck, Katz and Tucker 1998).

Failing to model duration dependence when it exists can leadd to a number of problems. At best, the estimator will be inefficient and the standard errors will be incorrect; at worst, inconsistent estimates may result, since, in effect, failing to include duration when it exists is a form of omitted variable bias (Beck, Katz and Tucker 1998).

Beck, Katz and Tucker (1998) note that a straightforward way to continue to model B-CSTS data in the logistic framework—but also account for duration dependence—is to simply create a time-since-last-event variable (i.e., the duration variable shown in Table 1), which is then turned into a vector of dummy variables. These are then included in the logit-GLM:

$$\Pr(y_{it} = 1 | \mathbf{x}_{it}, \boldsymbol{\kappa}_i) = \frac{1}{1 + \exp\left[-(\mathbf{x}_{it}\boldsymbol{\beta} + \boldsymbol{\kappa}_{it}\boldsymbol{\gamma})\right]} \tag{1}$$

Now, in addition to the standard covariates ($\mathbf{x}_{it}$ is a matrix of k regressors with dimensions $(N \cdot T) \times k$ with coefficients $\boldsymbol{\beta}$), $\boldsymbol{\kappa}_{it}$ is now included, and is a matrix of duration dummies with coefficients $\boldsymbol{\gamma}$.[3] An illustration of these dummy variables are shown in Table 2. For instance, $\kappa_1 = 1$ if the duration variable is equal to 1, $\kappa_2 = 1$ if the duration variable is equal to 2, and so on.

Since it is likely that some $\boldsymbol{\kappa}_i$ may be perfectly collinear with y_{it}, separation is likely to lead to estimation issues when using maximum likelihood; this will force Stata to drop the collinear dummy variables (Carter and Signorino 2010). To alleviate this, Carter and Signorino (2010) advocate for a simple approach of incorporating duration, duration squared, and duration cubed in the model instead of either splines (which are another approach that Beck, Katz and Tucker (1998) recommend) or dummy variables. While some consider $\boldsymbol{\kappa}_{it}$ to be nuisance parameters (Beck 2010), others contend that it is important to discuss and interpret—typically using plots—the estimated dependence function as something of theoretical interest (Carter and Signorino 2010; Williams 2016).[4] Regardless, both lines of reasoning agree that it is necessary to include some functional form of duration in the

[3] Only d dummy variables are needed, where $d = \text{Max(Duration)}$. Note that either the constant or one of the dummy variables must be dropped in order to estimate Equation 1.

[4] Moreover, the estimated duration can also be used to check if the assumption of proportional hazards has been violated (Carter and Signorino 2010).

Table 2: Duration dummy variables

Unit	Time	Event (y_{it})	Duration	κ_1	κ_2	κ_3
1	1	0	1	1	0	0
1	2	0	2	0	1	0
1	3	1	3	0	0	1
1	4	0	1	1	0	0
1	5	1	2	0	1	0
2	1	0	1	1	0	0
2	2	1	2	0	1	0
2	3	0	1	1	0	0
2	4	0	2	0	1	0
2	5	0	3	0	0	1
3	1	0	1	1	0	0
3	2	1	2	0	1	0
3	3	1	1	1	0	0
3	4	0	1	1	0	0
3	5	0	2	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

model in order to account for duration dependence.

One difficulty with implementing the advice above is that incorporating some functional form of duration dependence requires the creation of a duration variable, which is far less straightforward than taking lags or including time dummmies in standard CSTS data with a continuous dependent variable. This difficulty is compounded if some data are missing. In addition, many social scientists are unfamiliar with techniques such as splines (Carter and Signorino 2010). Below, I show a straightforward way to create a duration variable, even in the presence of missing data, using the command `mkduration`.

Accounting for dependence with `mkduration`

The command is as follows:

Syntax

```
mkduration eventvar, [dname() force polynomial spline() nknots()]
```

This command requires the specification of a single variable, `eventvar`, which is a dichotomous dependent variable where "1" indicates the presence of some event occurring at time *t* for unit *i*, and “0” indicates the absence of this event (i.e., `eventvar` will become the dependent variable in the logit model). The data must also first be set using `xtset`.

Options

`dname()` is an option to name the duration variable that will be generated by mkduration. By default, the duration variable is called simply “duration”.

`force` is an is an option to force the creation of duration data when gaps in time are present for one or more units. By default, gaps in the series are handled by replacing the duration variable with missings until the next event. Specifying force will fill in the duration variable. To do this, it must assume that no event occurred during the gap. This is described in greater detail in the example below.

`polynomial` is an option to automatically generate the variables duration, duration squared, and duration cubed, that can be used to proxy for duration dependence as suggested by Carter and Signorino (2010). The three resulting variables will be named “dname” (whatever the user selects for the duration name—or “duration” by default), “dname2” and “dname3”.

`spline()` is an option to create a spline to model duration dependence, as suggested by Beck, Katz and Tucker (1998). Note that one can specify either the polynomial or spline options, not both. The resulting spline variables to include in the model will include a “_spl1”, “_spl2” (and so on) suffix. Users can choose from the following:

- `spline(linear)` creates a linear spline. In effect, a piecewise linear model is run across the duration series, which can then be included in the logistic regression model. By default, five knots are used, meaning that duration will be split into six equal segments based on the percentiles of the data; the first knot will be placed at about the 16.66 percentile, the next at the 33.33rd percentile, and so on. The assumption with a linear spline is that duration has a linear (but not necessarily constant) effect on the probability of an event occurring between the knots. Increasing the number of knots allows for a more flexible relationship, but has the potential to overfit the data.

- `spline(cubic)` creates a restricted cubic spline that creates a linear function before the first knot, allows for a cubic polynomial function from the second and subsequent knots, and data past the final knot is assumed linear. By default, five knots are used, and are placed along percentiles of duration, as recommended by Harrell Jr (2015), which is the default in Stata.

`nknots()` is an option—to use if `spline()` is specified—to define the number of knots to include. By default, five knots are included for both the linear and cubic splines, although this can be changed anywhere from between three to seven knots.[5] Less knots are often more efficient, but offer less flexibility in modeling duration dependence. Greater numbers of knots increase flexibility, at the cost of (potentially) decreasing efficiency and overfitting duration.

Example

For an applied example I use data from Bapat and Zeigler (2016), who examine how the presence of terrorists may precipitate a "preventive" conflict from one state to another, especially if terrorists are hindering economic factors such as growth and resource revenues.[6] As they state in their article, conflict involvement between country dyads is likely to exhibit duration dependence over time, which is why they include a cubic polynomial of duration in their model results. We can construct the duration variable as follows using the dependent variable of a preventative conflict in year t from dyad country A towards country B (`initiatemid11`), being sure to first `xtset` the data:

```
xtset dyad year
mkduration initiatemid11
```

By default the generated duration variable is called "duration", although this can

[5] Users wanting more knots, or more precise placements of the knots, should instead generate the duration variable and create splines using Stata's `mkspline` command (which is the same command as that used for `spline()`, but offers greater flexibility).

[6] For brevity, I use a more simplified model than those in Bapat and Zeigler (2016).

be changed using `dname()`. The duration variable can quickly be turned into dummy variables κ_{it} and included in the model, the results of which are shown in Table 3, Model 1:

```
tabulate duration, generate(kappa_)
logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc kappa_1-kappa_18, noconstant
```

As is clear from the results, many of the κ_{it} dummies are statistically significant; a likelihood-ratio test comparing Model 1 against a restricted model with no κ_{it} fails to reject the null hypothesis that the restriction is valid. Note too that we failed to obtain estimates for κ_{16} and κ_{18}; in both cases, there were no conflicts that occurred either 16 or 18 years after a conflict onset, so separation results.

Instead of including duration dummies, we can use the recommendation of Carter and Signorino (2010) and create a cubic polynomial term of duration using the `polynomial` option. We can also change the name of the resulting duration variables using the `dname()` option:

```
mkduration initiatemid11, polynomial dname(dpoly)
logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dpoly dpoly2 dpoly3
```

These results are shown in Model 2 in Table 3. As an additional functional form choice, users can choose to model duration using splines:

```
* linear spline with 4 knots
mkduration initiatemid11, spline(linear) dname(dlinear) nknots(4)
logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dlinear_spl1 - dlinear_spl4


* cubic splines (5 knots default)
mkduration initiatemid11, spline(cubic) dname(dcubic)
logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dcubic_spl1 - dcubic_spl4
```

Each command shows (respectively), a piecewise linear spline with four knots (meaning that duration will be partitioned into quintiles), and a restricted cubic spline, using the

Table 3: Different approaches to account for duration

	(1)		(2)		(3)		(4)		(5)	
	Duration		Cubic		Linear		Cubic		Cubic Polynomial	
	Dummies		Polynomial		Spline		Spline		(with `force`)	
Terrorists	1.747***	(0.325)	1.771***	(0.324)	1.800***	(0.324)	1.775***	(0.324)	1.794***	(0.324)
Δ State Resource Rents	-0.112	(0.339)	-0.258	(0.326)	-0.287	(0.330)	-0.254	(0.325)	-0.243	(0.324)
Relative State Capacity	-0.480***	(0.151)	-0.483***	(0.148)	-0.477***	(0.149)	-0.477***	(0.149)	-0.485***	(0.148)
Constant			-3.061***	(0.570)	-3.222***	(0.620)	-3.391***	(0.573)	-3.112***	(0.567)
Duration Variables										
κ_1	-3.711***	(0.455)								
κ_2	-4.612***	(0.436)								
κ_3	-4.744***	(0.465)								
κ_4	-5.511***	(0.643)								
κ_5	-5.170***	(0.571)								
κ_6	-4.782***	(0.497)								
κ_7	-5.551***	(0.646)								
κ_8	-5.274***	(0.583)								
κ_9	-5.142***	(0.577)								
κ_{10}	-4.474***	(0.472)								
κ_{11}	-5.645***	(0.760)								
κ_{12}	-5.599***	(0.763)								
κ_{13}	-4.466***	(0.493)								
κ_{14}	-4.530***	(0.525)								
κ_{15}	-5.968***	(1.033)								
κ_{16}	N/A	(.)								
κ_{17}	-5.127***	(0.749)								
κ_{18}	N/A	(.)								
Duration			-0.901***	(0.261)					-0.877***	(0.257)
Duration^2			0.112***	(0.036)					0.108***	(0.035)
Duration^3			-0.004***	(0.001)					-0.004***	(0.001)
Spline 1					-0.621**	(0.250)	-0.511**	(0.205)		
Spline 2					-0.060	(0.170)	2.517	(1.709)		
Spline 3					0.079	(0.130)	-4.250	(4.020)		
Spline 4					-0.099	(0.118)	0.407	(4.197)		
N	2986		3278		3278		3278		3282	
Log Lik.	-315.389		-322.434		-324.449		-322.915		-326.277	

Note: Dependent variable is equal to one if dyad country A initiated preventative conflict towards B in year t, zero otherwise. Logistic regression with standard errors in parentheses. Two-tailed tests. * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

default of five knots. The results for these two options are shown in Models 3 and 4 in Table 3.

Given that interpreting the various approaches to duration in Table 3 is not that straightforward, we can instead plot the dummy variables, splines and/or cubic polynomials, to better understand the underlying nature of duration dependence in the data (Carter and Signorino 2010; Williams 2016). Here, we will quietly run each model, and generate the predicted probability of conflict across duration, while setting all other covariates to their means:

```
qui logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc kappa_1-kappa_15 kappa_17, nocon
adjust terrorgroup1 lnnatchange1 lnrpc, by(duration) generate(pr_dum) pr
qui logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dpoly dpoly2 dpoly3
adjust terrorgroup1 lnnatchange1 lnrpc, by(duration) generate(pr_poly) pr
qui logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dlinear_spl1 -dlinear_spl4
adjust terrorgroup1 lnnatchange1 lnrpc, by(duration) generate(pr_lspline) pr
qui logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dcubic_spl1 - dcubic_spl4
adjust terrorgroup1 lnnatchange1 lnrpc, by(duration) generate(pr_cspline) pr


lab var pr_dum "Duration Dummies"
lab var pr_cspline "Cubic Spline"
lab var pr_poly "Cubic Polynomial"
lab var pr_lspline "Linear Spline"
twoway line pr_dum duration, sort || line pr_cspline duration, sort ///
 || line pr_poly duration, sort || line pr_lspline duration, sort ytitle("Pr(y = 1)")
```

The resulting plot of these durations is shown in Figure 1. The duration for conflict appears to be non-monotonic; it quickly declines until around six or seven years after a conflict, rises slightly until just over a decade after a conflict, then continues a slow decline. Figure 1 also shows how the inclusion of the duration dummies, especially in the context of separation (which occurred for κ_{16} and κ_{18}) can result in "bumpy" durations. The other three strategies—a cubic polynomial, cubic spline, or linear spline—appear to be largely the same.

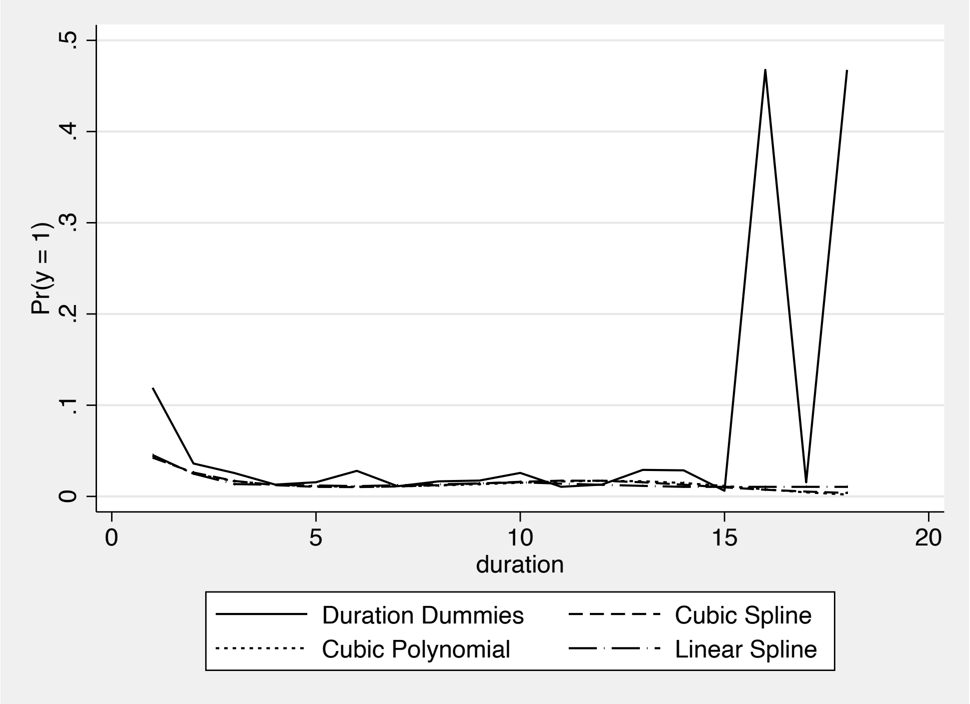

Figure 1: Different durations generated using `mkduration`

Missing data

One issue with duration dependence has to do with missing data. By default, `mkduration` will create a missing duration value for any instances between the missing values until after the next observed event. However, if the user is comfortable assuming that no events have occurred during the unobserved time period, they can use the `force` option to fill in the observed periods after the missing values, but before any observed event occurs. An illustration of this is shown in Table 4, where unit $i = 1$ is missing values of y_{it} for $t = 4, 5$, while unit $i = 2$ is missing the event variable at $t = 4$. By default (see the “Duration (default)” column), these missings will cause the duration variable to be missing until after the next observed events. In contrast, the “Duration (force)” column shows how the `force` option will account for the missing values when calculating the elapsed duration time.

Turning back to our example, we can use the `force` option to account for these missing values as follows (now calling the duration variable `dmiss`, and choosing to create a cubic

Table 4: `mkduration` and missing data

Unit	Time	Event (y_{it})	Duration (default)	Duration (force)
1	1	0	1	1
1	2	0	2	2
1	3	1	3	3
		[*missing* $t = 4, 5$]		
1	6	0	.	6
1	7	1	.	7
1	8	0	1	1
2	1	0	1	1
2	2	1	2	2
2	3	0	1	1
		[*missing* $t = 4$]		
2	5	0	.	3
2	6	0	.	4
⋮	⋮	⋮	⋮	⋮

polynomial), the results of which are shown in Model 5 in Table 3.

```
mkduration initiatemid11, force dname(dmiss) polynomial
logit initiatemid11 terrorgroup1 lnnatchange1 lnrpc dmiss dmiss2 dmiss3
```

Conclusion

In this paper, I have introduced `mkduration`, a simple, less error-prone way to create a duration variable in B-CSTS data when duration dependence is suspected. Replicating an example application using B-CSTS data, I have shown that this command allows users to easily account for duration dependence in whichever manner they choose, such as dummies, splines, or polynomials. Moreover, if users are willing to make additional assumptions, `mkduration` can easily account for dependence in the context of missing data.

References

Bapat, Navin A and Sean Zeigler. 2016. “Terrorism, dynamic commitment problems, and military conflict.” *American Journal of Political Science* 60(2):337–351.

Beck, Nathaniel. 2010. “Time is not a theoretical variable.” *Political Analysis* 18(3):293–294.

Beck, Nathaniel, Jonathan N Katz and Richard Tucker. 1998. “Taking time seriously: Time-series-cross-section analysis with a binary dependent variable.” *American Journal of Political Science* pp. 1260–1288.

Box-Steffensmeier, Janet M and Bradford S Jones. 2004. *Event history modeling: A guide for social scientists.* Cambridge University Press.

Carter, David B and Curtis S Signorino. 2010. “Back to the future: Modeling time dependence in binary data.” *Political Analysis* 18(3):271–292.

Clare, Joe. 2010. “Ideological fractionalization and the international conflict behavior of parliamentary democracies.” *International Studies Quarterly* 54(4):965–987.

Harrell Jr, Frank E. 2015. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis.* Springer.

Hillegeist, Stephen A, Elizabeth K Keating, Donald P Cram and Kyle G Lundstedt. 2004. “Assessing the probability of bankruptcy.” *Review of accounting studies* 9(1):5–34.

Way, Christopher and Jessica LP Weeks. 2014. “Making it personal: Regime type and nuclear proliferation.” *American Journal of Political Science* 58(3):705–719.

Williams, Laron K. 2016. “Long-term effects in models with temporal dependence.” *Political Analysis* 24(2):243–262.

Zorn, Christopher JW. 2000. “Modeling duration dependence.” *Political Analysis* 8(4):367–380.