Aldo Benini

Technical note:

"Runs of relief"

- A data management technique for humanitarian logistics analysts to identify multiple-day delivery runs *when short gaps during runs are allowed*

Abstract

In one of the first quantitative studies of local disaster relief allocation, Benini, Conley, Dittemore et al. (2006)[1] model the effects of survivor needs and logistics factors on decisions to ship relief to communities affected by the Pakistan earthquake in October 2005. The basic analytic unit is the combination of receiving local government area and delivery date during the relief operation. This leads to models of the occurrence and size of daily deliveries.

Critics may argue that such models capture factors influencing decisions that for the most part are short-term or tactical. To estimate effects on operational decisions, deliveries possibly stretching over several consecutive days – known as a delivery run or spell – may be a more appropriate study object.

Identifying such runs efficiently when short gaps within are legitimate (e.g. during weekends) was a challenge that proved insurmountable at the time of our study. More recently, Cox publicized a technique that achieves this (Cox 2007). I demonstrate it for the benefit of humanitarian logistics analysts, using the same Pakistan data. The 1,386 one-day deliveries can be collapsed into 486 runs if gaps shorter than three days are permitted. The longest spell lasted 32 days and had five gaps.

[1] Available at http://aldo-benini.org/Level2/humanitarian_data_analysis.htm. A shorter version appeared in the journal "Disasters", vol. 33(1).

Table of contents

Motivation: Study operational, rather than tactical, decisions

"Runs" equal "spells"

A humanitarian data analysis team (Benini, Conley et al. 2006) analyzed factors that influenced the decisions to ship relief goods to earthquake-affected communities in Pakistan in 2005 and 2006. The basic unit for which observed deliveries were calculated for the purpose of statistical modeling was the combination of receiving local government unit and delivery day, in short a "Union Council day". Separate models were estimated for food, shelter and clothing, construction material and tool deliveries.

In numerous cases, a Union Council area would receive shipments in one or another of those three commodity groups on several subsequent days. The sequence of days with deliveries to an area may be called a "run" or "spell" – I use these terms interchangeably -, with minimum length one. For example, a sequence of ten days' uninterrupted deliveries to a Union Council area would constitute a 10-day run or 10-day spell. The time between delivery spells is termed, in this paper, a "gap".

Tactical and operational perspectives

Some may argue that the choice of uniformly one-day long basic units may capture factors influencing *tactical* decisions, but loses sight of *operational* ones that meaningfully contributed to the strategic objectives of "Operation Winter Race" in 2005 and 2006, such as early stocking with food and tents of mountainous village communities to which access would be lost later in winter. In this line, defining entire delivery runs as basic units and calculating their quantities of interest for subsequent analyses may seem to be a more promising research strategy.

Why short gaps may occur

The gaps between subsequent runs sometimes were very short – one or two days only. In such a case, and again depending on the objectives, the analyst might want to consider whether it was more appropriate to assume one delivery spell disturbed by small gaps, rather than several distinct spells.

We have no data as to why short gaps occurred. Plausibly, they were caused, among others, by

- Temporarily low stock at regional hubs
- Hub-to-area transport factors, such as vehicle availability, road openings, weather
- Receiving ground crew factors, such as switching between receiving and warehousing, forwarding and distribution, movement to other receiving points within the Union area, rest and recreation
- Receiving community factors, such as time for local deliberations and dispute resolution, labor availability, Friday communal prayers.

A data management challenge and its solution

At the time, the challenge of efficiently determining such runs if short gaps were to be allowed, say of 1 or 2 days maximum, kept us from investigating this avenue. In other words, analytical alternatives were not explored because a prior *data management* problem seemed insurmountable within our time budget. This kind of self-limitation is always regrettable, regardless of whether run data would later have presented their own order of analytical challenges.

Subsequently, Nicholas Cox, at Durham University, UK, presented a technique to efficiently identify spells with gaps of permissible length (Cox 2007). I demonstrate its application to our Pakistan relief data as a case study for others who may want to create spell-based humanitarian relief data tables with permitted gaps, whatever the specific contents and analysis objectives. Cox's paper is written for STATA users, but his generic ideas can be implemented in other data management applications as well, if less conveniently, such as in MS Excel with some VBA programming or nested if-formulae.

An example of an extended delivery spell

Mohandri Union Council

Mohandri Union Council, an area with an estimated population in 2005 of 26,000, in Balakot Tehsil, Mansehra District, North West Frontier Province, was the destination of numerous relief deliveries. In fact, if deliveries from within all three commodity groups are viewed together, and gaps (= no deliveries in any commodity group) of one or two days within an ongoing spell are permissible, Mohandri registered exactly 14 delivery spells.

The longest relief delivery run recorded

I select this Union Council area here for this discussion because it was the beneficiary of the longest spell recorded during our 222-day observation period. This spell, lasting 32 days, occurred towards the end, after an 11-day spell and before a final 2-day spell, as shown schematically in the following table.

These deliveries took place between mid-April and mid-May 2006, a period no longer typical of the earlier emergency concerns in the wider response community. By March 2006, the emphasis had shifted towards the provision of reconstruction material and tools. In spring, many NGOs also dumped significant quantities of food, clothing and shelter material on affected communities, in preparation for their departure, and in a bid to avoid surrendering stocks to government agencies, which had started to demand stronger direct controls over relief goods.

In the following table, days with positive deliveries in any of the commodity groups are shaded dark gray. *Within*-spell gaps are in light gray. Gaps *between* spells, by definition at least three days without deliveries, are left unshaded. Quantities delivered are rounded up or down to metric tons.

Table 1: Sample spells of days with relief deliveries to a particular beneficiary unit, with permissible gaps within spells

Spell start and end dates	Days after disaster	Days in spell	Food	Shelter and clothing	Construction material and tools
[part of 11-day spell]	176				
	177		1		
	178				
	179				8
	180		2		3
April 7, 2006	181		2		
	182				
	183				
	184				
	185				
April 12, 2006	186	1	8		4
	187	2	10		3
	188	3			3
	189	4			
	190	5			
	191	6			22
	192	7			14
	193	8		1	8
	194	9	3	4	8
	195	10	30		
	196	11	18	4	10
	197	12			
	198	13	18		
	199	14			8
	200	15	9		4
	201	16		4	9
	202	17			
	203	18	10		10
	204	19			
	205	20	8		4
	206	21		3	17
	207	22			8
	208	23		2	15
	209	24		1	4
	210	25		3	14
	211	26			
	212	27		1	16
	213	28		1	12
	214	29		0	9
	215	30			5
	216	31	8		
May 13, 2006	217	32	8	1	7

	218			
	219			
	220			
May 17, 2006	221		3	13
May 18, 2006	222		1	3

As one can easily glean from the entries in the long spell, it had five gaps, four one-day, and one two-day. The continuity of deliveries was chiefly driven, during this particular period of time, by construction materials and tools, meaning if we did not consider the other two commodity groups, the 32-day long spell would remain the same; it would simply have two more one-day gaps, and one of the current one-day gaps would be two days' long.

While the basic concepts may be fairly intuitive, the point is to make the identification of spells efficient for data tables with numerous beneficiary units and observation periods of considerable length that together defy manual identification. Prior to carrying out any such algorithm, the data manager needs to know the definition of spells as a function of the research question in point.

Cox's generic scheme for identifying spells with permissible gaps

Basic concepts

Cox offers a very didactic approach to identifying spells, with or without permissible gaps. STATA users are encouraged to read his article, which offers advice also on dealing with spells of minimum required length, a topic not investigated here.

In generic definition, spells are recurrences of a particular variable value or of a condition over a segment of a sequence. Time series data is the most common situation, but a one-dimensional spatial sequence such as shops along a road may also vehicle spells, e.g. instances of restaurants, some of which may occupy adjacent buildings. Our case is defined by a particular condition: relief deliveries greater than zero on the days of a 222-day disaster response period.

For ordinary spells, and assuming that a variable for the underlying time or space has been created (e.g., "Days since the earthquake"), the principal elements to define and calculate are:

- The state or condition that defines the spell
- The beginning, or first observation, of the spell
- An unambiguous identifier that tags a distinct spell (usually a natural number)
- The end, or last observation, of the spell,

from which the length of the spell can easily be calculated. The number of distinct spells over the entire observation period equals the number of spell beginnings (unless the beginning of the first spell is explicitly outside).

The extension from one sequence to several – Cox, in time series manner, prefers “series” to “sequence” – adds no radically new conceptual elements. There will, of course, be a unit identifier, such as, in our case, for each participating local government area. For convenience, I suggest that one may create a numbering of the spells within each unit, and a differently named identifier for all spells in the entire panel data set.

Two passes through the data

For our purposes, Cox’s breakthrough contribution is the technique to identify spells with gaps of permissible length. His essential idea is that two passes through the data are an efficient procedure, with the second pass used to identify gaps as spells in their own right (Cox, op.cit., page 262 sqq). Then, by integrating the information on the uninterrupted initial spells and on the gaps, the liberally defined spells are identified.

Generic procedure

Assuming the data has been sorted first by unit, then by sequence (e.g. time), the generic procedure is:

1. Proceeding unit by unit, create an indicator variable for the start of each gap, i.e. set it = 1 if a gap starts with this observation, else = 0.
2. Within each unit, mark the gap start observation and any subsequent ones that are part of this gap with a gap identifier variable set to the running total of gap starts; set observations that are not part of gaps = 0.
3. Calculate the length of each gap, and create a variable that for all observations within each gap equals the length; else (i.e. for spells) set it to 0.
4. Define a, or use the already defined, maximum permissible gap length within spells, and define an indicator variable “inspell”, = 1 if the observation is part of a spell or of a permissible gap; else (i.e. of a longer than permissible gap) = 0.
5. Using “inspell”, create an indicator variable for the start of each of these liberal spells
6. Then, analogously to uninterrupted spells, create liberal spell identifiers using the running total of start indicators
7. Identify the ends and calculate the lengths of the liberal spells

Importance of sort orders

Care has to be exercised as to how the data is sorted for these operations; in the STATA implementation, Cox changes the sort order a number of times, but this may be best left to the users of particular applications.

It is important to remind us that this procedure achieves only the identification of spells with permissible gaps, but no other substantive results (unless one considers spell length distribution statistics to be of substantive interest); they say nothing about the correlates of the spells with or without gaps such as, in our case, the relief quantities shipped during spells. These are obtained in subsequent analyses.

An example in STATA

Why in STATA?

The fact that Cox chose the statistical package STATA (Stata Corporation 2003) to implement his two-pass technique is accidental to the problem at hand. As noted above, other applications, particularly spreadsheet programs, could be harnessed to achieve the same data management job.

However, some of STATA's features are particularly convenient for this task. Notably, as emphasized by Cox, they include subscripting, the prefix "by" (together with "by …, sort"), and the easy generation of binary variables based on conditions. The following assumes that you are a STATA user familiar with these features.

The purpose of this section is to highlight the STATA syntax of the critical steps in Cox's generic procedure, again by way of example. This includes important changes of the sort order needed at certain junctures. To do so, I use an edited log file and, for visual convenience, mark key passages with a yellow highlighter.

Data file and variables used for spell identification

The spells were identified for deliveries to 87 Union Council areas during a 222-day observation period after the disaster. The delivery data is contained in a so-called long-form file with 87 * 222 = 19,314 "Union Council days", each forming its own record. The data for October 2005 is considered incomplete, and we did not use the data for the first 23 days after the disaster in our earlier statistical analyses. Here this exclusion is not made.

The following variables are used:

Variable name	**Variable label**
uccode	Union Council (code)[Panel identifier]
dayssincedisaster	Days since disaster [Time variable]
HadDelivery	Union Council day with relief delivery
beginDeliverySpell	Begin of a delivery spell
DeliverySpellID_allUnions	ID delivery spell - all deliveries all Unions
DeliverySpellID_eachUnion	ID delivery spell within a given Union
DeliverySpellLengthDays	Length of delivery spell (days)
gapbegin	Start of a gap
gapIDwithinUnion	ID gap among gaps within same Union
gaplength	Length of gap
inspellGapsMax2days	Is part of delivery spell with gaps of max 2 days
begingSpellGapsMax2days	Start of delivery spell with permitted gaps
DeliverySpellID_allGapsMax2Days	ID spells with permitted gaps - all Union Councils
DelSpellLength_allGapsMax2Days	Length of spell with gaps of max 2 days
DelSpellTag_allGapsMax2Days	tag(DeliverySpellID_allGapsMax2 Days)

Selective code

Preliminaries

```
. * In order to work with the _by prefix, we need to ensure proper sorting:
.
. sort  uccode dayssincedisaster

. * For the readers not familiar with the Pakistan earthquake response situation, it may
be useful to point out that days with relief deliveries were relatively rare:

. tab  HadDelivery

      Union |
Council day |
with relief |
   delivery |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     17,928       92.82       92.82
          1 |      1,386        7.18      100.00
------------+-----------------------------------
      Total |     19,314      100.00
```

Identifying spells without gaps

Indicator for the start of the spell

```
. * Using a variant of Cox's [page 254]:  gen byte begin = (state == 2) & (state !=
state[_n-1]) [where "!" is the "not" operator in Stata, as usual]
.
. * with by: for the panel structure:

. by  uccode: gen byte beginDeliverySpell =  ( HadDelivery== 1) & ( HadDelivery !=
HadDelivery[_n-1])
```

Spell identifiers

```
. * Differently from Cox, I created an identifier for all spells in all Unions, and
another specific of the spells in each Union:
.

. * IDs for all spells in all Unions:
. * Note: This requires the "by  uccode:" not be used.

. gen DeliverySpellID_allUnions = cond( HadDelivery == 1, sum( beginDeliverySpell), 0)

. summ  DeliverySpellID_allUnions

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
DeliverySp~s |     19314    29.17521    122.1556          0        808

 * which is a quick way of finding out that there are 808 such spells all in all


. * IDs for spells in each Union:
.
.  by   uccode:  gen  DeliverySpellID_eachUnion  =  cond(  HadDelivery  ==  1,
sum( beginDeliverySpell), 0)


. summ  DeliverySpellID_eachUnion
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
DeliverySp~n |     19314    .7226364    3.370517          0         36
```

```
. * telling us that the Union with the most delivery spells had 36 of them.

. * [NB for STATA egen lovers: Cox shows that the former could be generated also with the help of egen   DeliverySpellID_allUnions = group( uucode   DeliverySpellID_eachUnion), label - see page 257 of his paper].
```

Spell length

```
. * Using "by … , sort:"

. by  DeliverySpellID_allUnions, sort: gen DeliverySpellLengthDays = _N


. * For an empirical result:

. summ  DeliverySpellLengthDays if  beginDeliverySpell, detail

               Length of delivery spell (days)
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              1
10%            1              1       Obs                 808
25%            1              1       Sum of Wgt.         808

50%            1                      Mean           1.715347
                        Largest       Std. Dev.      1.448209
75%            2              9
90%            3             10       Variance       2.097311
95%            5             14       Skewness       3.913255
99%            7             17       Kurtosis       28.14393

.  *  [The  if-condition  in  "summ   ..  if   beginDeliverySpell"  ensures  that  each  spells enters the calculation only once]
```

Spells with permissible gaps

Indicator for the start of gaps

```
.
. * Adapting Cox's syntax: by id (time), sort: gen byte gapbegin = !spell & spell[_n-1]
.
.  by    uucode   (   dayssincedisaster),   sort:   gen   byte   gapbegin   =   !HadDelivery   & HadDelivery[_n-1]

. * where the parentheses around dayssincedisaster ensure that the data get sorted first by uucode then by dayssincedisaster, but the operation be effected on all records within this particular uucode.
```

Gap identifier

```
.
. by  uucode: gen gapIDwithinUnion = cond(!HadDelivery, sum(gapbegin), 0)
```

Gap length

```
. * The "by .., sort" is modified again; there are no parentheses around gapIDwithinUnion because the operation is to be carried out separately for each gap:
```

```
. by  uccode gapIDwithinUnion, sort: gen gaplength = cond( gapIDwithinUnion, _N, 0)

. * [following Cox p.263 top, to ensure that outside gaps “0” be entered as the gap length]
```

Permissible gap length and indicator variable “inspell”

```
. * For our purposes, we admit spells with gaps no longer than 2 days (which includes those with 2-day weekends during which no deliveries were made):
.
. gen byte inspellGapsMax2days =  HadDelivery |  (gaplength < 3)
```

Start indicators for these more liberal spells

```
.
. by uccode ( dayssincedisaster), sort: gen byte begingSpellGapsMax2days = inspellGapsMax2days & !inspellGapsMax2days[_n-1]
```

Spell identifiers for all spells in all Union Councils

```
.
. sort  uccode dayssincedisaster

. * [No “by ..:”]

. gen DeliverySpellID_allGapsMax2Days = cond( inspellGapsMax2days, sum( begingSpellGapsMax2days), 0)

. summ  DeliverySpellID_allGapsMax2Days

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
Delive~2Days |     19314    24.02713    85.49059          0        486

. * which constitutes a significant reduction of the number of spells (down from 808 spells without gaps).
```

Spell lengths

```
. by DeliverySpellID_allGapsMax2Days (dayssincedisaster), sort: gen DelSpellLength_allGapsMax2Days = _N

. summ  DelSpellLength_allGapsMax2Days if  begingSpellGapsMax2days, detail

               DelSpellLength_allGapsMax2Days
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              1
10%            1              1       Obs                 486
25%            1              1       Sum of Wgt.         486

50%            2                      Mean           3.794239
                        Largest       Std. Dev.      4.685011
75%            4             26
90%            9             29       Variance       21.94933
95%           14             31       Skewness       2.839585
99%           24             32       Kurtosis       12.77161

. * with the maximum spell length of 32, this being the spell in our detailed example.
```

```
. * Plus, create a tag variable for convenience, such as for creating tables and graphs:

. egen  DelSpellTag_allGapsMax2Days = tag(DeliverySpellID_allGapsMax2Days)

. * which we use right here to ensure each spell is counted only once:

. tab  DelSpellLength_allGapsMax2Days if  DelSpellTag_allGapsMax2Days

DelSpellLen |
gth_allGaps |
   Max2Days |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |        217       44.56       44.56

[output omitted]

         32 |          1        0.21       99.79
      17470 |          1        0.21      100.00
------------+-----------------------------------
      Total |        487      100.00

* * which calls for a cosmetic correction at the end:
.

.  replace   DeliverySpellID_allGapsMax2Days  =  .  if   DelSpellLength_allGapsMax2Days  ==
17470
(17470 real changes made, 17470 to missing)

. replace  DelSpellLength_allGapsMax2Days = . if  DelSpellLength_allGapsMax2Days == 0
(17470 real changes made, 17470 to missing)

DelSpellLen |
gth_allGaps |
   Max2Days |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |        217       44.65       44.65
          2 |         54       11.11       55.76
          3 |         57       11.73       67.49
          4 |         43        8.85       76.34
          5 |         22        4.53       80.86
          6 |         12        2.47       83.33
          7 |         15        3.09       86.42
          8 |         13        2.67       89.09
          9 |          8        1.65       90.74
         10 |          9        1.85       92.59
         11 |          3        0.62       93.21
         12 |          4        0.82       94.03
         13 |          3        0.62       94.65
         14 |          4        0.82       95.47
         15 |          4        0.82       96.30
         16 |          3        0.62       96.91
         17 |          3        0.62       97.53
         18 |          1        0.21       97.74
         19 |          1        0.21       97.94
         20 |          1        0.21       98.15
         21 |          1        0.21       98.35
         22 |          1        0.21       98.56
         23 |          2        0.41       98.97
         24 |          1        0.21       99.18
         26 |          1        0.21       99.38
         29 |          1        0.21       99.59
         31 |          1        0.21       99.79
         32 |          1        0.21      100.00
------------+-----------------------------------
      Total |        486      100.00

. * possibly to be made more graphic through regrouping > 10 days and a histogram.


. * and careful to reestablish the original sort order:
```

```
. sort uccode dayssincedisaster

. save … etc.
```

An illustration: Quantities moved in one-day deliveries vs. during spells with short gaps

In our detailed study, we estimated separate models for food, shelter and clothing, and construction material deliveries. Here we combine them, for didactic simplicity.

```
. * The following two variables are used:
```

Variable name	**Variable label**
totalWeightKg	Commodities of three major types delivered (kg)
totalWeightKgInSpellwithGaps	Commodities delivered during spell (kg)

```
. * Single day quantities moved:
.
.  summ  totalWeightKg if  totalWeightKg > 0, detail

       Commodities of three major types delivered (kg)
-------------------------------------------------------------
      Percentiles      Smallest
 1%          400             20
 5%     1265.182             60
10%         2000            100       Obs                1386
25%         5500            100       Sum of Wgt.        1386

50%        10500                      Mean           20760.39
                        Largest       Std. Dev.      29239.03
75%        26000         193382
90%        49400         270000       Variance       8.55e+08
95%        75000         320000       Skewness       5.192197
99%       126360         458000       Kurtosis        53.8801

. * Total quantities delivered within spells with gaps:

.
.     by          DeliverySpellID_allGapsMax2Days     (dayssincedisaster),     sort:     egen
totalWeightKgInSpellwithGaps = sum(totalWeightKg)


. replace  totalWeightKgInSpellwithGaps = . if  totalWeightKgInSpellwithGaps == 0
(17470 real changes made, 17470 to missing)

. summ  totalWeightKgInSpellwithGaps if  DelSpellTag_allGapsMax2Days, detail

           Commodities delivered during spell (kg)
-------------------------------------------------------------
      Percentiles      Smallest
 1%          229             20
 5%         1700            100
10%         3500            200       Obs                 486
25%         8000            200       Sum of Wgt.         486

50%      22518.5                      Mean           59205.55
                        Largest       Std. Dev.      111087.5
75%        65650         728500
90%       136370       808832.5       Variance       1.23e+10
95%     220938.4         867820       Skewness       4.868499
99%       600000        1118273       Kurtosis       34.38115
```

Conclusion

Cox's technique makes the efficient identification of delivery runs with short gaps possible. In our example, the 1,386 instances of daily deliveries to 87 Council areas during the 222-day observation period can be collapsed into 486 delivery runs with gaps inside runs shorter than three days. The maximum cargo moved to a particular area on a single day was 458 metric tons of relief goods; the maximum moved in a delivery run was 1,118 metric tons. More than half of all runs, however, lasted only one or two days, and as such, by definition, did not have gaps.

This is essentially a *data management* job, not an analytic one. It removes an obstacle to the preparation of certain kinds of calculated variables that may be useful in subsequent statistical model choices and their estimation. Whether quantities based on identified delivery runs are valid measures of the underlying theoretical concepts – such as, for our motivation, "operational decision-making" - has to be determined on grounds that lie outside the data management technique itself.

Moreover, it cannot be taken for granted that the same statistical models that work for strictly daily delivery data would do so for run data. For example, we used Heckman selection models (Sigelman and Zeng 1999), which commonly need a so-called identifying variable; we found it in the question whether the Union Council area was the object of an open movement request the previous day or not. This would in all plausibility not work for run data models; for one would assume that such a request existed prior to the start of most, if not all, runs. Other classes of models may be appropriate, such as survival analysis models studying the time that lapsed till the onset of a delivery run, perhaps using the quantities shipped taken as additional needs indicators on the covariate side. All this, as far as I know, is unexplored territory in humanitarian logistics research.

Bibliography

Benini, A. A., C. Conley, et al. (2006). Survivor Needs or Logistical Convenience? – Factors shaping decisions to deliver relief to earthquake-affected communities, Pakistan 2005-06 [Version 30 October 2006]. Navigating Post-Conflict Environments. Washington DC, Vietnam Veterans of America Foundation / Information Management and Mine Action Programs (VVAF / iMMAP).

Cox, N. J. (2007). "Speaking Stata: Identifying spells." The Stata Journal 7(2): 249-265.

Sigelman, L. and L. Zeng (1999). "Analyzing Censored and Sample-Selected Data with Tobit and Heckit Models." Political Analysis 8(2): 167-182.

Stata Corporation (2003). STATA Base Reference Manual, Release 8. College Station, TX, STATA Press.

July 11, 2007 [Link updated Dec 30, 2008]