Using Manipula and CATI SubPriorities to Adjust and Fine-Tune Sample Delivery

Dave Dybicki, Joe Matuzak & James Wagner, The University of Michigan

Introduction

A common concern of study managers is controlling sample delivery to maximize productivity. From a business perspective, the goal is to increase efficiency and thereby cut costs. Blaise 4.8 introduced new instructions that allow easy adjustment of key fields within the current daybatch. [Daybatch, Transfer, and Daybatch Modi] and combining these with the SubPriority field introduced in Blaise 4.11 provides the ability to shape delivery on a sample-line level.

Other references to how this concern is handled can be found elsewhere. 

Goals and Challenges

There were three main goals to our effort:

1. Maximizing delivery control for a propensity modeling experiment.
2. Several groups of interviewers had to be included in the experiment.
3. The “manual” management of sample had to be tracked in order to detect any confounding of the experiment that may occur.

The first requirement was that the Blaise SMS sort the list of cases in a priority order that included both the experimental and control cases. It was decided that this could be done by interleaving cases from each group. One difficulty was the sort needed to be performed on the daybatch after the cases had been delivered to interviewers. Two methods were evaluated to perform the sort. The first method involved creating extremely large daybatches in order to accommodate the sorting.

Program Flow

1. The experimental and control cases had to be sorted at the beginning of each call window in an interleaving fashion so that each condition (experiment and control) would receive approximately equal treatment.
2. Several groups of interviewers had to be included in the experiment.
3. The “manual” management of sample had to be tracked in order to detect any confounding of the experiment that may occur.

Setting Priorities

A key goal was creating a sorting method that gave greater control than the internal Blaise algorithms so study managers could control the delivery of their sample according to changing study needs. Conversations with various study managers produced a list of areas they wanted to prioritize, including time zone, call number, release or replicate number, whether or not contact was made on a case, whether or not a household testing was completed, whether there was resistance, and the sampletype of the case.

Managers can apply whatever numeric weight they desire to any of these criteria, and those weights can be changed at any time. This allows them to give huge priority to any particular criterion by giving it a large number, or to aggregate small-weighting cases so cases meeting multiple criteria rise to the top of the list.

A timed script simply calculates the numeric value for each sample line, and then sorts them in descending order, so those with the highest total come first in the list and will thus be assigned the highest subpriority.

Subpriority Logic Flow

1. Manager must edit script to set priorities
2. Use SMS_SUBPRIORITY to sort the daybatch by subpriority
3. Display the value of each subpriority field
4. SMS refers to our sample management Blaise database

SMS is a constant concern of study managers is controlling sample delivery to maximize productivity. From a business perspective, the goal is to increase efficiency and thereby cut costs. Blaise 4.8 introduced new instructions that allow easy adjustment of key fields within the current daybatch. [Daybatch, Transfer, and Daybatch Modi] and combining these with the SubPriority field introduced in Blaise 4.11 provides the ability to shape delivery on a sample-line level.

++

Automation

Using Windows Scheduled Tasks, we are able to have variations of these scripts update the daybatch at specific times, which can vary by day, so we can coordinate efficiency with variations in factor's hours, with experimental priorities, etc. The script can also be run on demand, of course, so the daybatch can immediately be updated after creation or after any change a manager needs to make either in the daybatch or in the sorting priorities.

In this instance, there are two scripts: one to sort priorities and another to rewrite the subpriorities. One time window covers daytime coding, the daybatch is updated after the evening meetings, and weekend takes its own schedule.

Outcomes

If this procedure is successful, it should result in a relatively even balance of calls on experimental and control cases governed by the logic of the script. This is a call window. Under the experimental condition, if a case had its highest probability of contact in the current window, then its daybatch would be sorted to the top of the case priority list for calling. This was done for a randomly selected half of the sample, and the remaining cases were sorted in the second half.

Future Directions

Our initial development effort was spurred by trying to get this experiment working successfully, but there are many other paths that can be added. First, a control panel would allow project managers to make changes to priorities without needing to understand Manipula, and that panel could potentially contain a flybox to turn on or off the influence of certain subpriorities. The daybatch could also allow immediate running of the script, as managers could make changes and have them take effect right away as opposed to waiting for a scheduled task.

For Further Information

Please email ddybicki@isr.umich.edu or matuzak@isr.umich.edu for more information.