Data Appendix for "Child-Adoption Matching: Preferences for Gender and Race"

Mariagiovanna Baccara Allan Collard-Wexler
Leonardo Felli Alistair Wilson Leeat Yariv

September 2010

Abstract

We document the construction of the data used in "Child-Adoption Matching: Preferences for Gender and Race."

1 Data Construction

1.1 Data Sources

The data were collected from the adoption facilitator's website. On this website, there are two linked pages that we utilized (both publicly accessible):

- *"List of Currently Available Children,"* containing the list of children currently available on the website. We refer to this page as CA for short.
- *"Archive,"* containing the list of children who have been placed on the website in the past.

The data used in this project originate from four separate collection efforts:

1. **Perlscript Data** correspond to CA and archive data harvested via HTML on a daily basis. These data refer to the period from September 2008 to August 2009.
2. **PDF Data** correspond to data harvested from the same sources as above (the CA and archive pages), but transcribed from screengrabs in pdf using an external company. These data refer to the period from May 2008 to September 2008.
3. **RA data** contain CA data only. They were assembled by a research assistant who manually uploaded a spreadsheet with daily observations. These data were gathered from May 2007 to January 2008.
4. **Archive Data** contain CA data only and were put together using an Internet archive. We used this source to generate data between June 2004 and September 2007.

Data Source	Frequency	Percent
RA Data	9,819	1
Internet data	191,807	19
Interpolated from Internet data	531,590	53
PDF data	22,872	2
Interpolated PDF data	35,941	4
PerlScript data	205,175	21
Total	997,204	100

Table 1: Provenance of CA Data.

Table 1 specifies the distribution of our data across these sources. Figure 1 depicts the data collection efforts across time.

1.2 PAP Activity

The activity period of individual PAPs on the site is defined using two dates: the first time that an individual PAP appears in our records (i.e., the first application for a child, which is conditional on the PAP having become a client of the facilitator by paying an initial fee), and the last time that same PAP submits an application for a child. We assume that PAPs are actively checking the website and are aware of each child available on each day between these two end-points. Moreover, for some results in the paper, we define a PAP as 'active' up to either 10 or 90 days following their last application. In the case that a PAP was eventually matched to a BMO on the website, we consider the PAP inactive since the last application they submitted, assuming that the PAP became aware of the match as soon as the BMO made her choice (possibly a few days before the match appears on the website).

1.3 Interpolation

Some of our data points (in particular, the PDF and Archive Data) have resolution smaller than one day. In these cases, the data are filled via a one-sided interpolation: If an observation on a given day is missing, the data are assumed identical to the data point on the day before (that includes available children, outstanding applications for children, etc.). That is, if we observed data A on day 1, data B on day 5, and data C on day 7, our filled data set was constructed as A, A, A, A, B, B, C. Additionally, we coded the resolution of each element in our data as the time lag between actual observations, so the resolution for the example above would be: $0, 4, 4, 4, 0, 2, 0$.

1.4 BMOs' Attributes and Restrictions

BMO data were entered using the text produced by the HTML files in the CA data. Exploiting the consistency of the organization of the website, we searched for specific strings within specific columns of the data table.

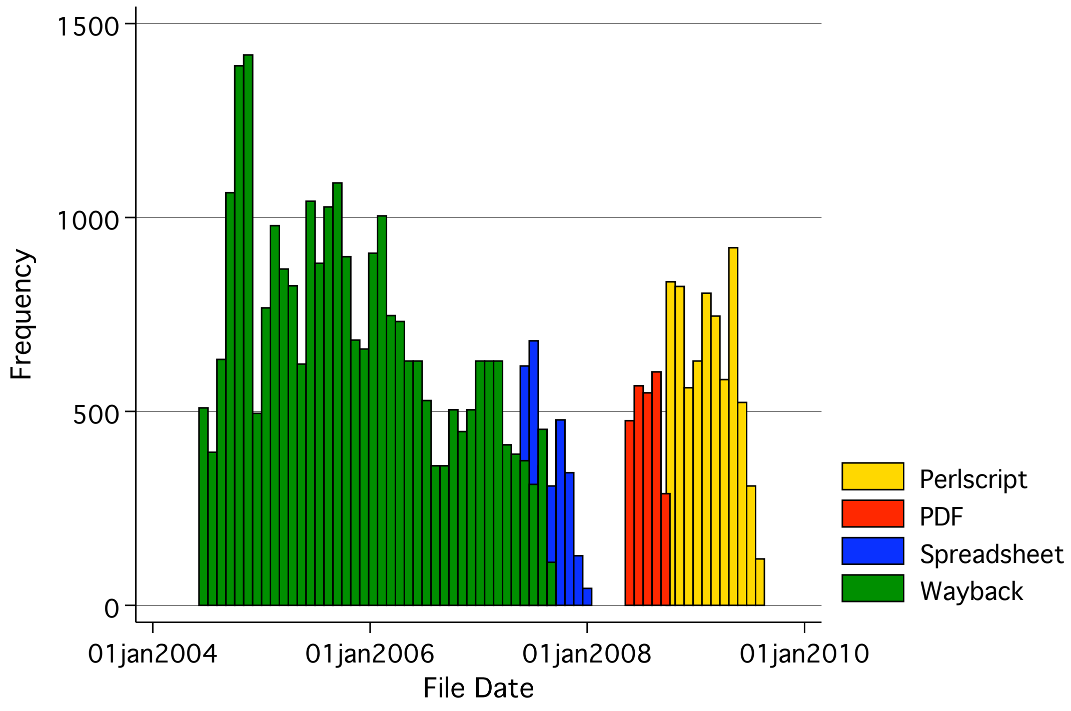

Figure 1: Data Collection over Time.

For instance the string ‘lesbian’ in the column of the CA data detailing the PAPs types acceptable to the BMO was used to code BMOs open to lesbian couples (note that all restrictions are worded in the direction of acceptance; e.g., ‘BMO wants a married couple or a single woman,’ ‘BMO will consider all families including lesbians, gay, single,’ etc.). Race percentages were coded using a similar method, using a database of words used in referring to ethnicities within the BMO characteristics column (e.g., ‘3/4 Caucasian, 1/4 African-American,’ etc.).

The BMO’s due date and the date on which the case was presented to the facilitator were captured searching for several alternative date formats and accuracies, as well as performing a local search in nearby lines for explanatory strings. For example, ‘Due Date: 08-Feb’ in a data point with date 25 December 2008 would translate into a coded due date of 02-08-2009. ‘Presented on 08-05-09’ would force the presentation date to be coded as 08-05-2009.

Finally, to code the adoption finalization costs, a research assistant went through the raw data determining the final monetary costs associated with every BMO.[1]

1.5 PAPs’ Attributes

1.5.1 Single and Same-Sex Scores

The website refers to PAPs reporting their first names or initials only. Thus, PAPs are coded in the data via strings such as ‘jack&jill,’ ‘mary,’ or ‘a&b.’[2] We used this information to determine the sexual orientation of a PAP, as well as whether the PAP is a couple or a single woman. When the names or initials did not indicate a couple, we assigned a value of 1 to the “Single PAP” score. We classified PAPs’ sexual preferences as follows:

1. We determined the gender of each name according to the classification ‘male,’ ‘female,’ and ‘unisex.’ In particular:
   (a) For well-known anglo and foreign names, coding was automatic.
   (b) For obscure names that were unknown to the coders, we checked with online child name databases to determine the classification of the name.
   (c) If a name’s gender specificity could not be determined, or the PAP couple was identified only through its initials, each name was assumed to be ‘unisex.’
2. If the couple was identified by one ‘male’ and one ‘female’ name, we assigned a value of 1 to the “straight couple” score. Similarly, if the couple were identified by two ‘male’ names or two ‘female’ names, we assigned a value of 1 to the “Gay PAP” or “Lesbian PAP” score, respectively.

[1]We discarded the few cases in which the BMO’s ID name changed over the period in which the case was posted on the website. This occurred in about four cases.

[2]Names were sorted alphabetically to make sure their reversal on the website was not coded as identifying a separate PAP unit.

3. Of the PAPs that had names with unambiguous gender classification, 73.7% were straight, 6.1% were gay, 6.9% were lesbian, and 13.3% were single. We used these priors to construct scores for PAPs with names entailing some gender ambiguity. In particular:
   (a) If the couple was identified by one 'unisex' name and one 'male' name, we assigned a value of 0.92 to the "Straight PAP" score and a value of 0.08 to the "Gay PAP" score.
   (b) If the couple was identified by one 'unisex' name and one 'female' name, we assigned a value of 0.91 to the "Straight PAP" score and a value of 0.09 to the "Lesbian PAP" score.
   (c) If the couple was identified by two 'unisex' names, we assigned a value of 0.85 to the "Straight PAP" score, a value of 0.07 to the 'Gay PAP' score, and a value of 0.08 to the 'Lesbian PAP' score.

1.5.2 Foreign Score

To code the "Foreign PAP" score, assuming a symmetric prior, we used Bayesian updating over a multinomial process to update the probability of a PAP being foreign. Specifically, we assumed a 10% error-probability for foreign parents applying for children they are barred from adopting. Then, given a 50/50 division of foreign and domestic PAPS (our symmetric prior), and an observed count of applications for "Foreign PAPs Allowed" and for "Foreign PAPs Not Allowed" children, we computed the posterior probability of each PAP being foreign.

The probability of observing n_f applications for "Foreign PAP Allowed" children and n_d applications for "Foreign PAP Not Allowed" children conditional on a PAP being foreign is:

$$Q_f = \Pr\{(n_f, n_d)\,|\text{Foreign}\} = \begin{pmatrix} n_d + n_f \\ n_f \end{pmatrix} (0.1)^{n_d} (0.9)^{n_f}.$$

The computation of the probability of observing n_f applications for "Foreign PAP Allowed" children and n_d applications for "Foreign PAP Not Allowed" children conditional on a PAP being domestic assumes that domestic PAPs are equally likely to apply for a "Foreign PAP Allowed" child and a "Foreign PAP Not Allowed" child. We denote by p_d the proportion of children available (over the entire period of activity of a PAP) for which foreign PAPs are not allowed to apply. The probability of observing the distribution of n_f and n_d applications as above, conditional on the PAP being domestic, is then given by:

$$Q_d = \Pr\{(n_f, n_d)\,|\text{Domestic}\} = \begin{pmatrix} n_d + n_f \\ n_f \end{pmatrix} p_d^{n_d} (1 - p_d)^{n_f}.$$

Bayesian updating with our symmetric prior over PAPs being foreign or domestic yields the posterior probability of a PAP being foreign:

$$\Pr\{\text{Foreign}\,|(n_f, n_d)\} = \frac{Q_f}{Q_f + Q_d}.$$

2 Data and Program Glossary

2.1 Data Glossary

case_data_all.dta:	data from the archive webpage.
ChoicePanel.dta:	combination of PAP choices for each child on each day.
grid_data.dta:	data from the CA webpage.

2.2 Statistical Program Glossary

grid_hedonic_regression1.do:	runs the finalization cost regressions.
Matching_Regression_Match-Not.do:	runs the regression on finding a match or not.
Matching_Regression.do:	runs the regressions on the BMO's choice of a PAP.
ChoicePanel_Sum9.do:	creates all the tables and figures in the paper, except for those describing matching, finalization cost, and the determinants of a BMO's choice.

2.3 Data Construction Programs

HTMLdata.m:	reads in data on archive from html and pdf files and imports them; general purpose script file calling the main functions, and getting data into matlab through the *outdata* cell variable.
generate_PAP_file.do:	generates the data set file *ChoicePanel.dta* from the data. Uses *pap_data.dta* and *CA_data.dta*
replacePAPnames.do:	changes a long list of misspellings, errors, etc. to the 'correct' values, as coded by hand.
import_CA_data-AJW.do:	imports the csv file generated by *FlatFileOutputPAP.m*, changes the names and various details that need correction. Also assigns unique IDs as necessary and generates a couple of diagnostic values. The main function is generating the file *pap_data.dta*.
DateEnter.m:	finds and codes date information using differing formats and regular expressions. In particular, 'mm-dd-yy' and 'mm-dd-yyyy' formats. Pre-processes the strings to replace words and other formats to create richer information. Dates are attributed to events via the strings on the same or previous lines.
DateEnterCD.m:	customized version of *DateEnter.m* for use with the Cases data. Changes where the algorithm looks for explanatory strings and dates.
DateExtract.m:	similar date extraction routine to *DateEnter.m*, but used with Cases data.
GenerateData.m:	global Script. Runs the data entry part within MATLAB.
InterestedPersonsVector.m:	formats the interested PAPs data from a row of the CA file. Takes the interested PAPs string and converts to a cell array.
MatchedPersonsVector.m:	similar to *InterestedPersonsVector.m*, but customized for data from the Cases file.

FlatFileOutputPAPs.m:	converts data from cell variable in MATLAB through to csv file for entry into STATA file for Grid data, where a row is a day-mother-pap entry.
HTMLtimemachine.m:	enters data from the HTML data captured from Internet Archive.
EnterHongData.m:	enters data from a customized csv-version of the RA entered data. Each i entry in outdata $(i,\cdot)$ represents a BMO, where the second element represents time on the site.
StripArchive.m:	function that saves HTML for targeted website for specified date ranges.
RaceFractionCode.m:	For each column entry in the cell given by the coordinate system, codes the racial fraction and word given in the text. Used to extract well-specified race data from the HTML.
AgeCode.m:	codes ages of children from string data.
CodeLanguage.m:	script file that runs the data refinement routines—i.e., those that convert string data to numeric coded data.
CodeLanguageCD.m	retasked version of *CodeLanguage.m* for the Archive data instead of the CA data.
CreateMatchInformation.m:	tries to match string data near to date information with known phrases, thereby coding matches, cases closed, missing, etc.
DateReplaceWords.m:	pre-formatting for dates; tries to put dates into a systematic format for subsequent data capture.
MoneyCode.m:	extracts monetary amounts from string data, looking for date ranges and stated amounts. For date ranges, the code is the top limit of the range. These data are superseded in the final analysis by the hand-entered amounts for each BMO.
RearrangePAP.m:	orders PAP pairs so that the names are listed in alphabetical order.
FlatFileOutputCaseData.m:	converts data from cell variable in MATLAB through to csv file for entry into STATA file for Archive data where a row is a date-mother entry.
HTMLcaseData.m:	enters information from the Archive page html.
FlatFileOutput.m:	converts data from cell variable in MATLAB through to csv file for entry into STATA file for CA data where a row is a date-mother entry.

2.4 Helper Programs Glossary

The following codes are "helper" functions in that they perform specific tasks such as manipulating strings and so on.

coderow.m:	enters data from a HTML-table tow into MATLAB. Used as an extraction tool for rows after the *gettabledata.m* file populates from the string.
coderowCaseData.m:	customized version of *coderow.m* for use with data from the Archive data rather than the CA data.
gettabledata.m:	finds the first table within an HTML file.

PreviousLine.m:	string manipulation utility. Function returns the line above/or below the current position within the string using custom line delimiters.
LineContents.m:	string manipulation utility. Function returns the line above/or below the current position.
replacestring.m:	string manipulation utility. Replaces one string with another.
RenameFiles.m:	unused. File manipulation utility. Basic utility for renaming files in a particular directory.
striptags.m:	string manipulation utility. Removes HTML tag information—i.e., transforms \¡a href=“link.htm”\¿link address\¡\/a\¿ to “link address” using regular expressions.