

# ACE Models with the NLSY

William Howard Beasley (Howard Live Oak LLC, Norman)  
Joseph Lee Rodgers (University of Oklahoma, Norman)  
Kelly Meredith (University of Oklahoma, Norman)  
David Bard (University of Oklahoma Health Sciences Center, OKC)

March 10, 2012

## Abstract

We describe how to use the `NlsyLinks` package to examine various biometric models, using the NLSY79.

## Contents

<b>1 Terminology</b>	<b>1</b>
<b>2 Example: DF analysis with a Simple Outcome for Gen2 Subjects, Using a Package Variable</b>	<b>2</b>
<b>3 Example: DF analysis with a univariate outcome from a Gen2 Extract</b>	<b>4</b>
<b>A Appendix: Creating and Saving R Scripts</b>	<b>6</b>
<b>B Appendix: Installing the NlsyLinks Package</b>	<b>6</b>

Not only am I hoping to capitalize on all researchers initially interested in NLSY BG studies, I'm also hoping for a small slice of attracting those that aren't initially interested in BG. If the vignette or articles are good, they should rank well among search engines for terms like "NLSY and R". Maybe some researchers using those terms (and not initially interested in BG) could stumble on the vignettes. If the examples are clear and are easy to run quickly, they may get interested in BG. Therefore, I'd like at least a few paragraphs explaining what BG is. This section is tentatively located in the appendix.

## 1 Terminology

This package considers both Gen1 and Gen2 subjects. **Gen1** refers to subjects in the original NLSY79 sample (<http://www.bls.gov/nls/nlsy79.htm>). **Gen2** subjects are the biological children of the Gen1 females -ie, those in the NLSY79 Children and Young Adults sample (<http://www.bls.gov/nls/nlsy79ch.htm>).

The **SubjectTag** variable uniquely identify NLSY79 subjects when a dataset contains both generations. For Gen2 subjects, the **SubjectTag** is identical to their CID (ie, C00001.00 -the ID assigned in the NLSY79-Children files). However for Gen1 subjects, the **SubjectTag** is their CaseID (ie, R00001.00), with "00" appended. This manipulation is necessary to identify subjects uniquely in inter-generational datasets. A Gen1 subject with an ID of 43 becomes 4300. The **SubjectTags** of her four children remain 4301, 4302, 4303, and 4304.

The **expected coefficient of relatedness** of a pair of subjects is typically represented by the variable **R**. Examples are: Monozygotic twins have **R**=1; dizygotic twins have **R**=0.5; full siblings (ie, those who share both biological parents) have **R**=0.5; half-siblings (ie, those who share exactly one biological parent) have **R**=0.25; adopted siblings have **R**=0.0. Other possibilities exist too. The font (and hopefully their context) should distinguish the variable **R** from the software R.

A subject's **ExtendedID** indicates their extended family. Two subjects will be in the same extended family if either: [1] they are Gen1 housemates, [2] they are Gen2 siblings, [3] they are Gen2 cousins (ie, they have mothers who are Gen1 sisters in the NLSY79, [4] they are mother and child (in Gen1 and Gen2, respectively), or [5] they are (aunt|uncle) and (niece|nephew) (in Gen1 and Gen2, respectively).

An **outcome variable** is directly relevant to the applied researcher; these might represent constructs like height, IQ, and income. A **plumbing variable** is necessary to manage BG datasets; examples are **R**, a subject's ID, and the date of a subject's last survey.

The **NLS Investigator** (<http://www.nlsinfo.org/investigator/>) is the best way to obtain the NLSY79 and NLSY97 datasets. See our vignette dedicated to the NLS Investigator by typing `vignette("NlsInvestigator")` or by visiting <http://cran.r-project.org/web/packages/NlsyLinks/>.

## 2 Example: DF analysis with a Simple Outcome for Gen2 Subjects, Using a Package Variable

The vignette's first example uses a simple statistical model and all available Gen2 subjects. The **CreatePairLinks** function will create a data frame where each represents one pair of siblings. This function examines the subjects' IDs and determines who is related to whom (and by how much). By default, each row it produces has at least six values/columns: (i) ID for the older subject **Subject1Tag**, (ii) ID for the younger subject **Subject2Tag**, (iii) ID for their extended family **ExtendedID**, (iv) their coefficient of expected relatedness **R**, (v *and beyond*) outcome values for the older subject; (vi *and beyond*) outcome values for the younger subject.

A DeFries-Fulker (**DF**) Analysis uses linear regression to estimate the  $h^2$ ,  $c^2$ , and  $e^2$  of a univariate biometric system. The steps are:

1. Use the **NLS Investigator** to select and download a Gen2 dataset.
2. Open R and create a new script (see [Appendix A](#)) and load the **NlsyLinks** package. If you haven't done so, first install the **NlsyLinks** package (see [Appendix B](#)).
3. Within the R script, load the linking dataset. Then select only Gen2 subjects. The 'Pair' version of the linking dataset is essentially a sparse matrix.
4. Load and assign the **ExtraOutcomes79** dataset.
5. Specify the outcome variable name and filter out all subjects who have a negative value in this variable. The NLSY typically uses negative values to indicate different types of missingness (see 'Further Information' below).
6. Create a double-entered file by calling the **CreatePairLinks** function. At minimum, pass the (i) outcome dataset, th (2) raw dataset, and the (iii) name(s) of the outcome variable(s).
7. Use **DeFriesFulkerMethod3** function (i.e., general linear model) to estimate the coefficients of the DF model. The interpretations of the DF analysis can be found in Rodgers, Buster and Rowe (2001).

```
> ### R Code for Example DF analysis with a simple outcome and Gen2 subjects
> #Step 2: Load the package containing the linking routines.
> require(NlsyLinks)
> #
> #Step 3: Load the LINKING dataset and filter for the Gen2 subjects
> data(Links79Pair)
> dsLinking <- subset(Links79Pair, RelationshipPath=="Gen2Siblings")
> summary(dsLinking)
```

ExtendedID	Subject1Tag	Subject2Tag	R
Min. : 2	Min. : 201	Min. : 202	Min. : 0.2500
1st Qu.: 3159	1st Qu.: 315901	1st Qu.: 315902	1st Qu.: 0.2500
Median : 6116	Median : 611901	Median : 611902	Median : 0.5000
Mean : 5937	Mean : 593989	Mean : 593991	Mean : 0.4192

```

3rd Qu.: 8511    3rd Qu.: 851103    3rd Qu.: 851104    3rd Qu.: 0.5000
Max.      :12673    Max.      :1267301    Max.      :1267302    Max.      : 1.0000
                                     NA's      :563.0000

```

```

RelationshipPath
Gen2Siblings:11075

```

```

> #
> #Step 4: Load the OUTCOMES dataset and the linking dataset, and then examine the summary.
> data(ExtraOutcomes79)
> dsOutcomes <- ExtraOutcomes79 #'ds' stands for 'DataSet'
> summary(dsOutcomes)

```

```

      SubjectTag      SubjectID      Generation MathStandardized
Min.      :    201    Min.      :    201    Min.      :2    Min.      : 65.00
1st Qu.: 310302    1st Qu.: 310302    1st Qu.:2    1st Qu.: 91.00
Median : 604607    Median : 604607    Median :2    Median : 100.00
Mean      : 601313    Mean      : 601313    Mean      :2    Mean      : 99.97
3rd Qu.: 876203    3rd Qu.: 876203    3rd Qu.:2    3rd Qu.: 110.00
Max.      :1267501    Max.      :1267501    Max.      :2    Max.      : 135.00
                                     NA's      :2353.00

```

```

      Weight      WeightForAge19To25 WeightStandardized
Min.      : 75.0    Min.      : 75.0    Min.      : -2.768e+00
1st Qu.: 130.0    1st Qu.: 140.0    1st Qu.: -7.059e-01
Median : 155.0    Median : 165.0    Median : -2.207e-01
Mean      : 161.9    Mean      : 172.1    Mean      : -4.252e-11
3rd Qu.: 186.0    3rd Qu.: 195.0    3rd Qu.: 4.739e-01
Max.      : 485.0    Max.      : 485.0    Max.      : 7.178e+00
NA's      :3475.0    NA's      :6380.0    NA's      : 3.475e+03

```

```

WeightStandardizedForAge19To25
Min.      : -2.772e+00
1st Qu.: -6.879e-01
Median : -1.969e-01
Mean      : -4.086e-11
3rd Qu.: 4.905e-01
Max.      : 7.489e+00
NA's      : 6.380e+03

```

```

> #
> #Step 5: This step isn't necessary for this example, because Kelly Meredith already
> # groomed the values. But if not if the negative values
> # (which represent NLSY missing or skip patterns) still exist, then:
> dsOutcomes$MathStandardized[dsOutcomes$MathStandardized < 0] <- NA
> #
> #Step 6: Create the double entered dataset.
> dsDouble <- CreatePairLinksDoubleEntered(
+   outcomeDataset=dsOutcomes,
+   linksPairDataset=dsLinking,
+   outcomeNames=c('MathStandardized')
+ )
> summary(dsDouble)

```

```

      Subject1Tag      Subject2Tag      ExtendedID      R
Min.      :    201    Min.      :    201    Min.      : 2    Min.      : 0.2500
1st Qu.: 315752    1st Qu.: 315752    1st Qu.: 3158    1st Qu.: 0.2500
Median : 611901    Median : 611901    Median : 6116    Median : 0.5000
Mean      : 593990    Mean      : 593990    Mean      : 5937    Mean      : 0.4192
3rd Qu.: 851104    3rd Qu.: 851104    3rd Qu.: 8511    3rd Qu.: 0.5000
Max.      :1267302    Max.      :1267302    Max.      :12673    Max.      : 1.0000
                                     NA's      :1126.0000

```

```

RelationshipPath MathStandardized_1 MathStandardized_2

```

```

Gen2Siblings:22150   Min.    : 65.00   Min.    : 65.00
                    1st Qu.: 89.00   1st Qu.: 89.00
                    Median : 98.00   Median : 98.00
                    Mean    : 98.29   Mean    : 98.29
                    3rd Qu.: 108.00  3rd Qu.: 108.00
                    Max.    : 135.00  Max.    : 135.00
                    NA's    :3791.00  NA's    :3791.00

> #
> #Step 7: Estimate the ACE components with a DF Analysis
> ace <- DeFriesFulkerMethod3(outcomeForSubject1=dsDouble$MathStandardized_1, outcomeForSubject2=
> ace

$HSquared
[1] 0.8595078
$CSquared
[1] 0.03879863

$ESquared
[1] 0.1016935

$RowCount
[1] 16588

```

*Further Information:* If the different reasons of missingness are important, further work is necessary. For instance, some analyses that use item Y19940000 might need to distinguish a response of “Don’t Know” (which is coded as -2) from “Missing” (which is coded as -7). For this vignette example, we’ll assume it’s safe to clump the responses together.

### 3 Example: DF analysis with a univariate outcome from a Gen2 Extract

The vignette’s second example differs from the previous example in two ways. First, the outcome variables are read from a CSV ([comma separated values](#) file) that was downloaded from the [NLS Investigator](#). Second, the DF analysis is called through the function `AceUnivariate`; this function is a wrapper around some simple ACE methods, and will help us smoothly transition to more techniques later in the vignette.

The steps are:

1. Use the [NLS Investigator](#) to select and download a Gen2 dataset. Select the variables ‘length of gestation of child in weeks’ (C03280.00), ‘weight of child at birth in ounces’ (C03286.00), and ‘length of child at birth’ (C03288.00), and then download the \*.zip file to your local computer.
2. Open R and create a new script (see [Appendix A](#)) and load the `NlsyLinks` package.
3. Within the R script, load the linking dataset. Then select only Gen2 subjects.
4. Read the CSV into R as a `data.frame` using `ReadCsvNlsy79Gen2`.
5. Verify the column exists, and rename it something meaningful to your project. It is important that the `data.frame` is reassigned (ie, `ds <- RenameNlsyColumn(...)`).
6. Specify the outcome variable name and filter out all subjects who have a negative value in this variable. The NLSY typically uses negative values to indicate different types of missingness.
7. Create a double-entered file by calling the `CreatePairLinks` function. At minimum, pass the (i) outcome dataset, th (2) raw dataset, and the (iii) name(s) of the outcome variable(s).
8. Use `AceUnivariate` function to estimate the coefficients.

```

> ### R Code for Example of a DF analysis with a simple outcome and Gen2 subjects
> #Step 2: Load the package containing the linking routines.

```

```

> require(NlsyLinks)
> #
> #Step 3: Load the linking dataset and filter for the Gen2 subjects
> data(Links79Pair)
> dsLinking <- subset(Links79Pair, RelationshipPath=="Gen2Siblings")
> #
> #Step 4: Load the outcomes dataset and the linking dataset, and then examine the summary.
> # Your path might be: filePathOutcomes <- 'C:/BGRResearch/NlsExtracts/Gen2Birth.csv'
> filePathOutcomes <- file.path(path.package("NlsyLinks"), "extdata", "Gen2Birth.csv")
> dsOutcomes <- ReadCsvNlsy79Gen2(filePathOutcomes)
> summary(dsOutcomes)

```

SubjectTag	SubjectID	ExtendedID	Generation
Min. : 201	Min. : 201	Min. : 2	Min. : 2
1st Qu.: 310302	1st Qu.: 310302	1st Qu.: 3103	1st Qu.: 2
Median : 604607	Median : 604607	Median : 6045	Median : 2
Mean : 601313	Mean : 601313	Mean : 6008	Mean : 2
3rd Qu.: 876203	3rd Qu.: 876203	3rd Qu.: 8757	3rd Qu.: 2
Max. : 1267501	Max. : 1267501	Max. : 12675	Max. : 2
SubjectTagOfMother	C0005300	C0005400	C0005700
Min. : 200	Min. : 1.000	Min. : -3.000	Min. : -3
1st Qu.: 310300	1st Qu.: 2.000	1st Qu.: 1.000	1st Qu.: 1981
Median : 604600	Median : 3.000	Median : 1.000	Median : 1985
Mean : 601311	Mean : 2.338	Mean : 1.489	Mean : 1986
3rd Qu.: 876200	3rd Qu.: 3.000	3rd Qu.: 2.000	3rd Qu.: 1990
Max. : 1267500	Max. : 3.000	Max. : 2.000	Max. : 2008
C0328000	C0328600	C0328800	
Min. : -7.00	Min. : -7.0	Min. : -7.00	
1st Qu.: 37.00	1st Qu.: 99.0	1st Qu.: 18.00	
Median : 39.00	Median : 115.0	Median : 20.00	
Mean : 33.51	Mean : 103.9	Mean : 16.51	
3rd Qu.: 39.00	3rd Qu.: 128.0	3rd Qu.: 21.00	
Max. : 51.00	Max. : 768.0	Max. : 48.00	

```

> #
> #Step XX: Verify and rename an existing column.
> VerifyColumnExists(dsOutcomes, "C0328800") #Should return '11' in this example.
[1] 11
> dsOutcomes <- RenameNlsyColumn(dsOutcomes, "C0328800", "BirthWeightInOunces")
> #
> #Step XX: For this item, a negative value indicates the parent refused, didn't know,
> # invalidly skipped, or was missing for some other reason.
> # For our present purposes, we'll treat these responses equivalently.
> dsOutcomes$BirthWeightInOunces[dsOutcomes$BirthWeightInOunces < 0] <- NA
> #
> #Step XX: Create the double entered dataset.
> dsDouble <- CreatePairLinksDoubleEntered(
  outcomeDataset=dsOutcomes,
  linksPairDataset=dsLinking,
  outcomeNames=c('BirthWeightInOunces')
)
> #
> #Step XX: Estimate the ACE components with a DF Analysis
> ace <- AceUnivariate(
  outcomeForSubject1=dsDouble$BirthWeightInOunces_1,
  outcomeForSubject2=dsDouble$BirthWeightInOunces_2,
  relatedness=dsDouble$R,
  method="DeFriesFulkerMethod3"
)

```

```
)  
> ace  
$HSquared  
[1] 0.4822847  
$CSquared  
[1] 0.04685057  
  
$ESquared  
[1] 0.4708647  
  
$RowCount  
[1] 14822
```

## **A   Appendix: Creating and Saving R Scripts**

bla bla

## **B   Appendix: Installing the NlsyLinks Package**

bla bla bla