Int J Public Health (2014) 59:415–422
DOI 10.1007/s00038-013-0538-2

SSPH+ SWISS SCHOOL OF PUBLIC HEALTH +

HINTS & KINKS

Loading, merging and analysing demographic and health surveys using R

Dieter Vanderelst · Niko Speybroeck

Received: 31 May 2013 / Revised: 12 December 2013 / Accepted: 18 December 2013 / Published online: 11 February 2014

Introduction

The MEASURE DHS project collects nationally representative data on health and population. Typically, multiple surveys have been conducted for each country. To date (October 2013), the DHS website provides data from over 300 surveys conducted in more than 90 countries. This makes the DHS a very interesting data source. Needless to say, MDHS data have been used in a wide variety of studies (Van Malderen et al. 2013; Van de Poel and Speybroeck 2009; Masanja et al. 2008). The MEASURE DHS project maintains a database of survey-based publications.[1]

Analysing DHS data can be complicated because of the format in which the data are made available. The data for each survey are provided as a number of dataset types with records for different units of analysis such as households, household members, women or children. The same variables are often already included in different dataset types to avoid having to merge them before analysis. For example, most variables for household characteristics are included in the women, men, and children dataset types. In spite of this, there are instances when researchers have to merge different dataset types to obtain a data set including all required variables ready for analysis.

This paper is aimed at lowering the barrier to using the DHS by researchers. As illustration we will use the inequality in child mortality and access to medical care among households with different wealth levels. We investigate whether or not, in the DR of the Congo, poorer households (1) live further away from a health facility and (2) have a higher child mortality rate. Using this example, we show how the open source software R can be used to perform the following tasks:

- Load DHS data and extract variables
- Merge data from different files
- Analyse the data using the `survey` package

While we use an example on health inequality and child mortality, it should be noted that exactly the same procedure can be used to analyse the DHS data on HIV, malaria or any other topic the DHS collects data on.

We assume a basic knowledge of R for which good tutorials are available online.[2] Furthermore, many introductory books, including on using R to analyse survey data, e.g. Lumley (2011), can be found.[3]

Electronic supplementary material The online version of this article (doi:10.1007/s00038-013-0538-2) contains supplementary material, which is available to authorized users.

D. Vanderelst (✉)
Department of Environment, Technology and Technology Management, Faculty of Applied Economics, University Antwerp, Prinsstraat 13, 2000 Antwerpen, Belgium
e-mail: dieter.vanderelst@ua.ac.be

N. Speybroeck
Faculté de santé publique, Université catholique de Louvain, Clos Chapelle-aux-champs 30, 1200 Woluwe-Saint-Lambert, Belgium
e-mail: niko.speybroeck@uclouvain.be

[1] Database of DHS based publications: http://www.measuredhs.com/publications/index.cfm.

[2] One very well-written hands-on tutorial can be found at http://personality-project.org/r/.

[3] List of books on R: http://www.r-project.org/doc/bib/R-books.html.

Getting the data

Downloading the data

An overview of available surveys can be found on the DHS website.[4] The DHS website also contains instructions on how to obtain access to the data.[5] The supplementary material of this paper contains a step-by-step tutorial on downloading and organising the data such that the example code can be run. The supporting material also shows how to install three R packages required for running the example code. The data for this paper were downloaded from the DHS website in February 2013.

Identifying variables for analysis

The variables needed to answer the questions outlined in "Introduction" can be looked up in the DHS recode manual which can be found online.[6] The list of available datasets indicates for each survey which version of the recode manual should be used. For the Congo Democratic Republic Demographic and Health Survey 2007, this is recode manual DHS-V. Some country-specific variables are not listed in the recode manuals and can only be found in the DHS report accompanying the data. More information about these variables can be found in the supporting material of this paper.

The children dataset type lists all births to interviewed women in the 5 years preceding the survey. The data contain a variable `B5` indicating whether the child was alive at the time of the survey. Both the child and household dataset types contain a variable `V190` listing the wealth quintile of the household. However, the time to get to the nearest health facility in minutes, variable `SH108A`, is only available in the household dataset type. Therefore, we need to merge the household dataset with the children dataset. Merging data is done by constructing a new variable in both datasets which allows to merge the data for the correct household to the record for each child.

Loading and merging data

The first step in creating the appropriate data format is to load the data files and select the required variables. For the purpose of this article, a function `read.file()` is provided (see lines 5–20 in Listing 1) to read a data file and select the number of variables.

The supporting material contains a table showing how matching variables can be constructed from the variables in each dataset type. From this table, we read that the matching variables can be created by concatenating `V001` & `V002` in the children file and `HV001` & `HV002` in the household file. The variables to be analysed in the children data are `V190` (wealth index) and `B5` (whether the child was alive at the time of the interview). In the household data we select variable `SH108A` listing the time it takes to get to the nearest health facility from the place of residence. Finally, three additional variables, `V005`, `V021` and `V025`, are loaded from the children data. These will be needed when analysing the data using the `survey` package (see further). We can select all the data required by executing two lines of code (i.e. lines 23–24 in Listing 1) using the `read.file()` function. Table 1 lists the variables used in this example.

Table 1 Variables used in the example

Variable locations		
Variable	Content	Dataset type
Matching variables		
V001	Cluster number	Children data set
V002	Household number in cluster	Children data set
HV001	Cluster number	Household data set
HV002	Household number in cluster	Household data set
Design variables		
V021	Primary sampling unit	Children data set
V005	Sample weight	Children data set
V025	Type of place of residence	Children data set
Analysis variables		
V190	Wealth index (quintile)	Children data set
SH108A	Time to get to health facility	Household data set
B5	Child alive or dead	Children data set

Descriptive statistics			
Variable	Range	Mean (SD)	Value counts
V021	1-300	NA	NA
V025	1–2, 1: urban, 2: rural	NA	1: 3,575, 2: 5,417
V190	1–5, 1: poorest, 5: richest	02.87 (1.40)	1: 2,038, 5: 1,483
SH108A	0–900, in minutes	61.36 (83.29)	NA
B5	0–1, 0: no, 1: yes	00.88 (0.31)	0: 1,005, 1: 7,987

Variables are used for three different purposes in this paper: (1) to construct a new matching variable to merge the children and household data, (2) to specify the survey design to allow for proper analysis, (3) as variables of interest that are being analysed. The lower part of the table lists the descriptive statistics of variables if applicable

[4] The listing of all available DHS surveys: http://www.measuredhs.com/data/available-datasets.cfm.

[5] DHS instructions on acessing the data http://www.measuredhs.com/data/Using-Datasets-for-Analysis.cfm.

[6] DHS Recode Manuals: http://www.measuredhs.com/publications/publication-dhsg4-dhs-questionnaires-and-manuals.cfm.

Line 23 in Listing 1 reads the children dataset type from file `CDKR50FL.SAV` and retrieves variables `V001`, `V002`, `V005`, `V021`, `V025`, `V190` and `B5`. Variables `V005`, `V021` and `V025` will be used when analysing the data with the `survey` package. Line 24 loads the household dataset type from file `CDHR50FL.SAV` and retains three variables: `V001`, `V002`, `SH108A`. The data are stored in two data frames named `children` and `households`, respectively.

After loading the data, the second step is to construct a matching variable in each data frame by concatenating the appropriate variables. This can be done by the native R function `paste()`. Lines 26–27 of Listing 1 create a new variable `matching` in data frames `children` and `households`. Notice, that we concatenate the variables as listed in table provided in the supporting material. When concatenating variables to make a matching variable, a character separating the variables should be added. Here

Listing 1: R script to analyze the DHS data. This code was tested in R version 3.0.1 using survey package version 3.29-4 and foreign package version 0.8-54 on Windows 7 (64 bit)

```
1  # Clearing the R workspace
2  rm( list =ls( all =TRUE))
3  #Loading the required foreign package
4  require ( foreign )
5  #Definition of the read. file function .The readfile function reads SPPS format data.
       Therefore the foreign package must be loaded. Arguments: (1) filename: Filename of
       a data file in the SPSS format,(2) selected .vars: An optional vector of variable
       names which will be extracted from the data. Extracting the variable names is
       done by pattern matching. Any variable name beginning with one of the strings in
       the vector will be returned. This allows for easy extraction of many related
       variables . For example specifying 'V01' will results in V011, V012, V013, V014,
       ... being returned .
6  read. file <-function(filename, selected . vars=c())
7  {
8      dta<-read.spss(filename,use. value . labels =FALSE,to.data.frame=TRUE)
9      names(dta)<-sub("[[:punct:]]$","." ,names(dta))
10     checkna<-!is.na(dta)
11     checkna<-colSums(checkna)
12     dta<-dta[,checkna>0]
13     if ( length ( selected . vars )>0)
14     {
15         matches<-unique(grep(paste(selected.vars, collapse ="|" ),names(dta),value=TRUE
               ))
16         if (length(matches)<1){warning('No matching variables were found.')}
17         dta<-dta[matches]
18     }
19 dta
20 }
21
```

```
#Step1: Read in children and household data. Retain only the necessary variables
children <-read.file('CDKR50FL.SAV',c('V001','V002','V021','V190','B5','V005','V025'))
households<-read.file('CDHR50FL.SAV',c('HV001','HV002','SH108A'))
#Step2: Add a matching variable to both the children and the household dataframe
children$matching<-paste(children$V001,children$V002,sep="_")
households$matching<-paste(households$HV001,households$HV002,sep="_")
#Step3: Merge the household data to the children data
merged<-merge(children,households,by="matching")

#Analysis
#Loading the survey package
library(survey)
#Specifying the survey design
merged$sampleweights<-merged$V005/1000000
design<-svydesign(ids=~V021+V002,strata=~V025,weights=~sampleweights,data=merged)
#Descriptives
survival <-svyby(~B5,by=~V190,design=design,FUN=svymean,na.rm=TRUE,vartype=c('se','ci'))
distances<-svyby(~SH108A,by=~V190,design=design,FUN=svymean,na.rm=TRUE,vartype=c('se','ci'))
survival
distances
#Plot descriptives (figure 2)
##Loading the plotrix package which allows plotting confidence intervals
require(plotrix)
##prepare a figure with two panels
par(mfrow=c(1,2))
##plot left pane
plotCI(survival$V190,survival$B5,li=survival$ci_l,ui=survival$ci_u,main='Survival',xlab='Wealth index, Quintile, V190',ylab='Prop. of child. alive, B5')
##plot right pane
plotCI(distances$V190,distances$SH108A,li=distances$ci_l,ui=distances$ci_u,main='Distance',xlab='Wealth index, Quintile, V190',ylab='Time to nearest health fac., minutes, SH108A')
#Fit GLM models
model1<-svyglm(B5~V190,design=design,family=quasibinomial)
model2<-svyglm(SH108A~V190,design=design)
summary(model1)
summary(model2)
```

we use "_" as a separating character (see lines 26–27). This avoids the possibility of different values being concatenated into the same matching value. If no separating character is added `V001 = 123 & V002 = 45` and `V001 = 12 & V002 345` will, for example, result in the same matching variable value.

The final step is to merge both data frames using the native R function `merge()`. By calling `merged <- merge(children,households,by=''matching'')` the data frame `households` will be merged to the `children` using the newly constructed `matching` variable (line 29). This is to say, to each child the data for the corresponding household will be added and the resulting data frame `merged` has the same number of records as `children`. The data frame `merged` now contains all variables necessary to address the questions outlined above. The first ten lines of the data frame `merged` are shown in Listing 2.

Specifying the survey design

Specifying the design is done by setting the following parameters using the `svydesign()` function of the `survey` package. Three parameters should be set:

1. Sampling weights: Weights are used to make calculated averages representative of the population. Weights can be constructed by dividing the appropriate DHS variable by 1,000,000 as required by the DHS. A table in the supplementary material lists the DHS variables to be used for this purpose.
2. Sampling units: The primary sampling unit (PSU) is given by variable `V021` in women's and children's files, `MV021` in men's dataset types, and `HV021` in household dataset types. The secondary sampling unit (SSU) is the household and is given by variable `V002` in women's and children's files, `MV002` in men's dataset types, and `HV002` in household dataset types.

Listing 2: First ten lines of the merged data frame

```
> merged[1:10,]
   matching V001 V002   V005 V021 V025 V190 B5 HV001 HV002 SH108A sampleweights
1     1_103    1  103 1098095    1    1    5  1     1   103     30     1.098095
2     1_103    1  103 1098095    1    1    5  1     1   103     30     1.098095
3     1_103    1  103 1098095    1    1    5  1     1   103     30     1.098095
4     1_103    1  103 1098095    1    1    5  0     1   103     30     1.098095
5     1_103    1  103 1098095    1    1    5  1     1   103     30     1.098095
6     1_103    1  103 1098095    1    1    5  1     1   103     30     1.098095
7     1_141    1  141 1098095    1    1    4  1     1   141     10     1.098095
8      1_19    1   19 1098095    1    1    5  1     1    19     30     1.098095
9      1_19    1   19 1098095    1    1    5  1     1    19     30     1.098095
10    1_279    1  279 1098095    1    1    5  1     1   279      7     1.098095
```

Data analysis

Having merged both dataset types, we can move on to analysing the data (Listing 1, lines 31–55). As the data comes from a survey study, the sample weights and the survey design must be taken into account to obtain correct point estimates and variances. This can be done using the `survey` package (Lumley 2004, 2013). This package provides a range of functions for calculating descriptives and fitting generalized linear models for survey data. Using the `survey` package entails two steps. First, the survey design must be specified and stored in a variable. Next, this variable can be used as an argument to the specialized functions provided by `survey` to analyse the data.

3. Stratification: In some surveys the stratification used to design the sample is captured in the variable `V023`. However, in other surveys, `V023` is blank or is set to "National". The description of the survey implementation in the survey's DHS report should be read to deduce the stratification variables.

Pseudocode for specifying the design for a survey is given in Listing 3. The parts between <> should be replaced by the appropriate variable names. The DHS report for the current dataset states that the stratification was done based on the type of residential area (urban/rural). This is given by variable `V025`. Therefore, our design specification should have the following arguments: `ids=~ V021+ V002` and `strata= ~V025`. The weights variable,

Listing 3: Pseudocode for specifying the survey design

```
1 design<-svydesign(ids=~<PSU>+<SSU>,strata=~<VAR1>+...+<VARN>,weights=~<
      sampleweights>, data=<data frame>)
```

Listing 4: Contents of pivot tables containing the descriptive statistics

```
1  > survival
2    V190        B5          se       ci_l       ci_u
3  1    1 0.8670902 0.016532146 0.8346878 0.8994926
4  2    2 0.8686005 0.013102888 0.8429193 0.8942816
5  3    3 0.9025407 0.011132484 0.8807214 0.9243600
6  4    4 0.8892728 0.010183122 0.8693143 0.9092314
7  5    5 0.9363800 0.008485803 0.9197481 0.9530119
8  > distances
9    V190   SH108A        se       ci_l       ci_u
10 1    1 85.26062 13.811118 58.19133 112.32992
11 2    2 78.43738 13.571159 51.83840 105.03636
12 3    3 67.46702 11.840813 44.25945  90.67458
13 4    4 40.02706  6.002517 28.26234  51.79177
14 5    5 20.81170  1.386224 18.09475  23.52864
```

defined on line 35 in Listing 1, is used to specify the weights: `weights=~ sampleweights` (see line 36 in Listing 1).

Descriptive statistics

Line 38 creates a table of the children survival rate `B5` as a function of the household wealth quintile `V190`. Notice that the variable `design`, created on line 36, is passed as an argument to the `svyby()` function. Line 39 creates a similar table for the time to get to the nearest health facility (`SH108A`). On both lines, we pass the optional argument `vartype=c('se','ci')`. This results in both the standard error and the 95 % confidence intervals being reported. The contents of pivot tables `survival` and `distances` is shown in Listing 4.

For each quintile, the mean, standard error (`se`), lower (`ci_l`) and upper (`ci_u`) confidence interval are listed. From these tables it can be seen that richer households (higher values of `V190`) tend to live closer to a health facility and have more surviving children. Indeed, of all children born in the poorest households (i.e. `V190`=1) 86 % were still alive at the time of the survey. In the richest households (i.e. `V190`=5), 93 % of the children were alive. Likewise, members from the poorest households (i.e. `V190`=1) need on average 85 min to get to a health facility. Members of the richest households (i.e. `V190`=5) need only 21 min to get to the nearest facility.

Lines 48 and 50 in Listing 1 use the `plotCI()` function from the `plotrix` package to graph the descriptives (see Fig. 1). These plots visualize the trends that can be read in Listing 4.

Inferential statistics

Besides functions for calculating descriptives, the `survey` package also allows for inferential statistics. For example, the `svyglm()` function fits a generalized linear model to data from a complex survey design. On line 52 of Listing 1 this function is used to fit a GLM with `B5` as dependent and `V190` as predictor. Notice that, as before, the variable design is provided as an argument to specify the survey design. Because `B5` is a binary variable we use `family=quasibinomial`[7] to specify a logistic regression. Similarly, line 53 fits a GLM with `SH108A` as dependent and `V190` as predictor. Because the dependent is a continuous variable in this case, no family needs to be

[7] Normally, one would use `family=binomial` to specify a logistic regression in R. However, as we are using weighted data, we need to use `family=quasibinomial`. Please refer to the `survey` package help on the `svyglm()` function for more details.

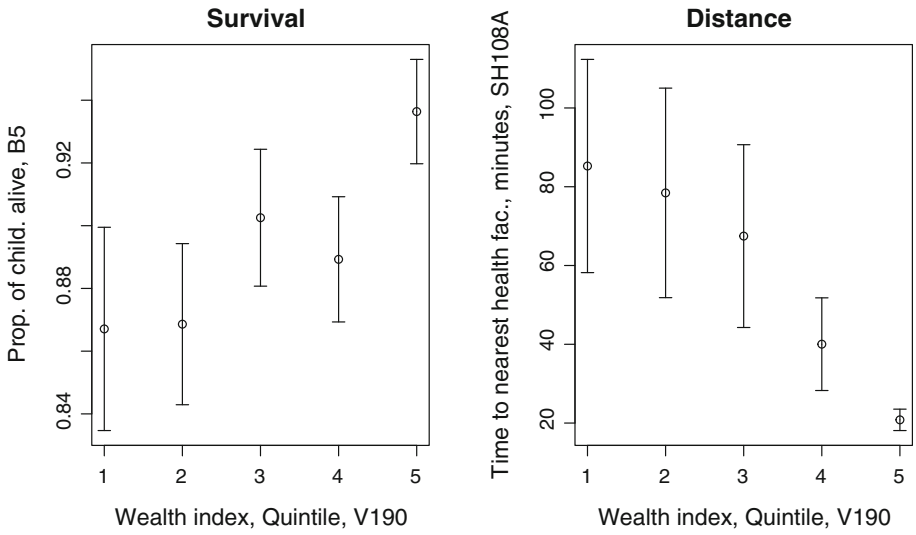

Fig. 1 Plots of the descriptive statistics. The descriptive statistics are stored in data frames `survival` (*left*) and `distances` (*right*). *Left* the child survival rate (`B5`) as a function of the households' wealth index (`V190`). *Right* the time to get to a health facility (`SH108A`) as a function of the households' wealth index (`V190`). The *vertical bars* denote 95 % confidence intervals

Listing 5: summary of the results of fitting two GLM models.

```
> summary(model1)
Call:
svyglm(formula = B5 ~ V190, design = design, family = quasibinomial)
Survey design:
svydesign(ids = ~V021 + V002, strata = ~V025, weights = ~sampleweights,
    data = merged)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.65235    0.14422  11.457  < 2e-16 ***
V190         0.15899    0.04199   3.786 0.000185 ***
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
(Dispersion parameter for quasibinomial family taken to be 0.9989899)
Number of Fisher Scoring iterations: 4
> summary(model2)
Call:
svyglm(formula = SH108A ~ V190, design = design)
Survey design:
svydesign(ids = ~V021 + V002, strata = ~V025, weights = ~sampleweights,
    data = merged)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  108.241     16.924   6.396 6.19e-10 ***
V190         -16.504      3.616  -4.564 7.35e-06 ***
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
(Dispersion parameter for gaussian family taken to be 8138.299)
```

specified (by default `family=gaussian` is used). Calling `summary(model1)` and `summary(model2)` on lines 54 and 55 returns the results for both GLMs. The output of these calls is shown in Listing 5. This output reveals a statistically significant linear effect of wealth index on child mortality. The odds ratio of a child being alive is estimated to increase by 17 % for each unit of increase in wealth index.[8] Likewise, richer households turn out to live significantly closer to a health facility (`model2`). An increase in wealth quintile is associated with a decrease of about 16 min to a health facility.

Missing data

In the presented analysis we have ignored the presence of missing data. However, each DHS variable usually contains missing data. Moreover, when merging data set types, typically some records cannot be matched, for example, due to errors during data entry. The default behaviour of the `merge()` command on line 36 is to omit all non-matched records. Likewise, when calculating the point estimates in lines 38 and 39 of Listing 1, the optional argument `na.rm=TRUE` tells R to ignore missing values. While ignoring missing values was deemed acceptable in the current example, the reader should be aware of their presence and assess whether they might introduce a bias in the analysis.

Conclusion

An integrated solution to analysing DHS data would be a dedicated package that integrates downloading, organizing, merging and analysing the data—including the GPS data provided for some surveys. Much of the functionality needed for such a package is already in place across different existing packages (e.g. `foreign`, `prevR` and `survey`). In the absence of such a package, we hope that papers such as this (see also Speybroeck et al. 2010, 2012; Konings et al. 2010) for R-based DHS analysis tutorials) help lower the barrier to analysing DHS data and enable researchers to realize the full potential of the DHS data.

Acknowledgments We thank the anonymous reviewers for their insightful comments on an earlier version of the manuscript.

References

Konings P, Harper S, Lynch J, Hosseinpoor AR, Berkvens D, Lorant V, Geckova A, Speybroeck N (2010) Analysis of socioeconomic health inequalities using the concentration index. Int J Public Health 55(1):71–74

Lumley T (2004) Analysis of complex survey samples. J Stat Softw 9(1):1–19

Lumley T (2013) Survey: analysis of complex survey samples, R package version 3.29-4

Lumley T (2011) Complex surveys: a guide to analysis using R, vol 565. John Wiley, London

Masanja H, de Savigny D, Smithson P, Schellenberg J, John T, Mbuya C, Upunda G, Boerma T, Victora C, Smith T et al (2008) Child survival gains in Tanzania: analysis of data from demographic and health surveys. Lancet 371(9620):1276–1283

Speybroeck N, Harper S, de Savigny D, Victora C (2012) Inequalities of health indicators for policy makers: six hints. Int J Public Health 57:859–860

Speybroeck N, Konings P, Lynch J, Harper S, Berkvens D, Lorant V, Geckova A, Hosseinpoor AR (2010) Decomposing socioeconomic health inequalities. Int J Public Health 55(4):347–351

Van de Poel E, Speybroeck N (2009) Decomposing malnutrition inequalities between scheduled castes and tribes and the remaining Indian population. Ethn Health 14(3):271–287

Van Malderen C, Van Oyen H, Speybroeck N (2013) Contributing determinants of overall and wealth-related inequality in under-5 mortality in 13 African countries. J Epidemiol Commun Health 67:667–67

[8] The coefficient denotes the log of the increase in odds ratio associtated with increasing the predictor by one unit. Hence, $\Delta \frac{p}{1-p} = e^{0.158} = 1.172$. Hence, odds ratio is predicted to increase by 1.17 for each increase in wealth index.