Data analysis and quality indices for data collected in a rural, low-development area

Gustaf Rydevik

U.U.D.M. Project Report 2007:8

Examensarbete i matematisk statistik, 20 poäng

Handledare och examinator: Hans Garmo

Februari 2007

Department of Mathematics

Uppsala University

Data analysis and quality indices for data collected in a rural, low-development area

Degree thesis for a Msc in mathematical statistics, with work conducted at the Iganga/Mayuge Demographic Surveillance Site, Uganda. The analysis on missed pregnancies co-written with Dorean Nabukalu, Iganga/Mayuge DSS.

Gustaf Rydevik

February 2007

Abstract

The thesis contains two parts. The first part is an overview of the Iganga/Mayuge DSS site, and the work conducted there. The second part documents quality control work and data analysis conducted during autumn 2006. Two main results were found. First, that the quality of the data was somewhat low, with unreasonable estimates of demographic rates. Second, that there is a large amount of pregnancies being missed, with large effects on estimates of miscarriage and young deaths rates.

Sammanfattning

Detta arbete är uppdelat i två delar. Den första delen är en översikt över det arbete som pågår vid Iganga/Mayuge DSS, en plattform för statistiska undersökningar i södra Uganda. Del två innehåller det kvalitetsarbete och de dataanalyser som genomfördes på plats av författaren under hösten 2006. Två huvudresultat presenteras. Kvalitetsundersökningen visade att datakvaliten var undermålig i vissa avseenden. Framförallt gav en del demografiska statistika orimliga siffror, i flera fall lägre än halva de förväntade värdena. Det gjordes också en korstabulering mellan registrering av graviditeter och graviditetsutfall som visade att ett stort antal graviditeter missas, och att detta gör att missfall och spädbarnsdödlighet underskattas med upp till 30 procent.

Acknowledgements: First and foremost, I wish to thank all the people working at the Iganga/Mayuge project. Dorean Nabukalu for teaching me about all the big and small parts that make up the DSS, and for helping me out in bug finding and editing. Daniel Kadobera, for keeping the computers up and running, and for the great support. Eddie Galiwango and the rest of office for the great work you are doing. All the Team Leaders and Field Assistants; you guys are the heart of the DSS! Stefan Peterson, for helping me come to Iganga in the first place, and for keeping an interest in the work I've been doing. To all of you in Uganda: a big Weebale nnyo!
I would also like to thank my supervisor in Uppsala, Hans Garmo, who has helped make the thesis readable and understandable.
Finally, I would like to thank the department of linguistics in Uppsala, and SIDA, who awarded me a Minor Field Study grant, and made it economically possible to travel to Uganda.

Contents

1 The Iganga/Mayuge DSS site

1.1 Introduction

The need for research is often large in developing countries,in order to find methods for solving the wide variety of problems the countries face. Additionally, as in the west, administrators and policy makers have to distribute their limited resources in an efficient and productive way. One of the biggest obstacles hindering researchers, administrators, and policy makers is the lack of an information infrastructure. There is no population register from which to draw random samples, little or no statistical or demographical information, and no way to keep track of individuals, making sampling studies very difficult. The lack of detailed knowledge about indicators such as expected life length, causers of death, child mortality, incidence of various diseases, or how well off the population is from a socio-economic viewpoint, means that it is very hard to know what measures are needed to lower death rates, increase the health standards, or lift a larger proportion of the population out of poverty.

To implement an information infrastructure full scale, as developed countries have done, would be far too costly for most of the worlds poorer countries. In many cases it would even be impossible, due to the high demands statistical surveys and vital registration place on both economy and infrastructure.

Demographic Surveillance Sites (DSS's) is a method with which to bridge the gap between information needs and available funds. The goal of such sites is to capture detailed, longitudinal information about a small enough area that logistical and economical issues are surmountable. The concept has become more and more popular in recent years, with several newly established sites around the world. Most of these sites are organised organised in an international network known as INDEPTH [INDEPTH 2007].

Iganga/Mayuge DSS is a demographic survaillance site that was started in 2004 as a collaboration between the Makerere University in Kampala, and Karolinska Institutet in Stockholm. The purposes of the Iganga/Mayuge DSS are twofold, depending on the perspective. From a local perspective: the district planners, the village leaders and the people living in the area, the purpose of the DSS is to strengthen the district. By generating information about household living standards, deaths, births and migrations, it is hoped that the public resources can be distributed more efficiently, and that initiatives to increase life length and public health standards can be directed towards the areas where they generate the most benefits.

From the perspective of the Makerere University, and of researchers, the purpose is to have an area where research and field projects can be easily conducted, and to a high standard. An area where a sampling frame is already in place, where follow-ups of individuals included in a study can be done easily and at a low cost (once the baseline costs of running the site has been taken into account), and where the demographic, health, and socioeconomic composition of the population are well understood.

One of the biggest challenges facing the Iganga/Mayuge DSS is to reconcile these two views. To provide for the needs of the community, while at the same time maintaining high standards of data quality without unnecessary expenditures.

1.2 Characteristics of the Iganga/Mayuge Demographic Surveillance Area

Iganga is an administrative region in Uganda, wtih a socio-economic status slightly below average. It has about 550 000 inhabitants, and is located about 120 km east of Kampala, the capital. The district is predominantly rural, with matoke, cassava and maize as the main crops. The crops are mainly grown for sustenance farming, with only a small proportion being sold. There is one hospital in the region, located in Iganga town, and a total of 93 health centers. Some general characteristics of the district can be found in the 1999/2000 household survey published by the Uganda Bureau of Statistics[UBOS, 2000]. At the time of the survey,

- Literacy rate was 63%
- 92% were either self-employed, or "unpaid family worker", and received no monthly salaries.
- 82% of the working age population (i.e. all above seven years old) were engaged in crop farming.
- Average household size was 5.8 persons.
- Average household income was shs 116 400/month (500 sek).
- Average per capita household spending was shs 21 300/month (80 sek).
- 42% of the population had "fallen sick" in the month preceding the survey. Of those, 42% suffered from malaria.

While these numbers are several years old, they nevertheless give an indication of the standard of living one can expect in the district as a whole. The reason for the data being old is the same reason that the DSS was set up for: There is no newer information available. The demographic surveillance area (DSA) covers a fairly small portion of the Iganga District, and additionally covers a part of the Mayuge District. A part of Iganga Town is included in the DSA, but the majority of the inhabitants live in rural areas. As of the end of round one, 67522 individuals had been registered in the DSA. 377 of those had been recorded dead, and an additional number had migrated out of the area.

1.3 Task description

The Iganga DSS site was starting to be well established by the start of round two. Routines were in place, most of the persons working had one or more year

of experience, and the work flowed fairly smooth. However, there was still a lack of documentation of the routines established, as well as a lack of knowledge concerning the quality of the database at the end of rounds. Therefore, it was decided to:

- develop flowcharts of the work, in order to get an overview of what were and/or should be done when collecting the data.
- Set up routines for generating standard demographic variables, making it easy to get continuos feedback on what stories the collected data were telling.
- Try to get measures on the quality of the data, and figure out how to catch mistakes and errors.

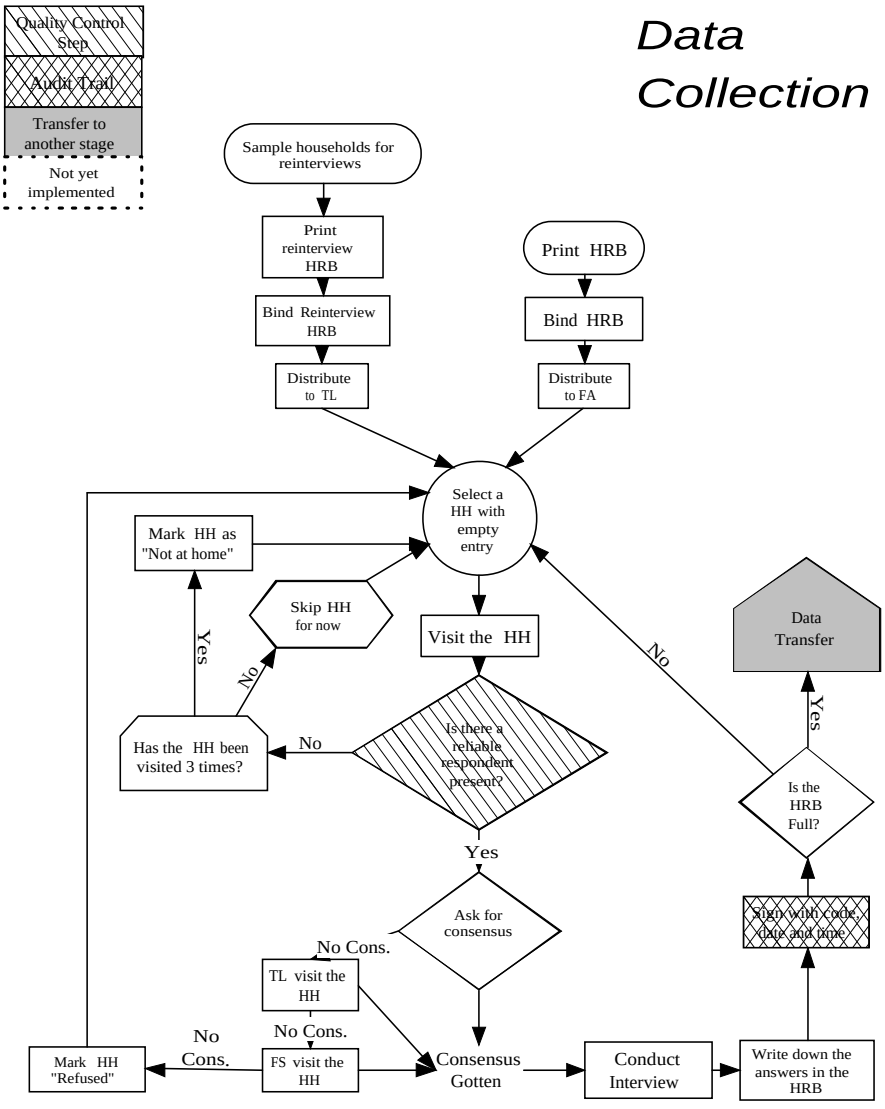
The rest of the thesis documents this work.

1.4 Flow of work at the DSS

For collecting the data, the DSS uses interview rounds, with an ideal spacing of three months. There are 36 Interviewers, known as Field Assistants (FA's) employed. each FA are responsible for a number of villages in the area where they live, and visit each and every household in the villages. For each household, they find one reliable respondent, who gives information about every member in his household: If someone has died, moved, given birth, gotten pregnant and so on. This is written down in a Household Registration Book (HRB), containing the id-numbers and personal information for the members of 25-36 households (the workload expected to be finished in a week). The entire interview process can be seen in fig. 1.

The HRB is handed over to Team Leaders (TL's) who check it for consistency and bad entries. If necessary a revisit to households with unclear responses is made. When finished, the TL passes on the HRB to the data entry department, where clerks copy the information from paper into the database, which is based on a program called Household Registration System (HRS)[Mcleod et al, 2000]. The database has a number of logical checks implemented, which means that if the data has illogical entries (Girls under ten giving birth, infants dying before they are born etc.), the HRS prompts. In such cases, the HRB's are returned to the field for additional clarification. See fig. 2 for an overview of this part of the process.

After all the data for one update round has been entered into the database, it is subjected to analysis. After a number of internal consistency checks, standard statistics (population pyramid, mortality rate, death rate) are calculated . Additionally, more specific analysis is conducted, depending on the interests of the local planners, special questions that have been asked during the round, etcetera. This part can be seen in fig. 3.



By Gustaf Rydevik, Nov. 2006

Figure 1: The first step is Field Assistants interviewing in the field.

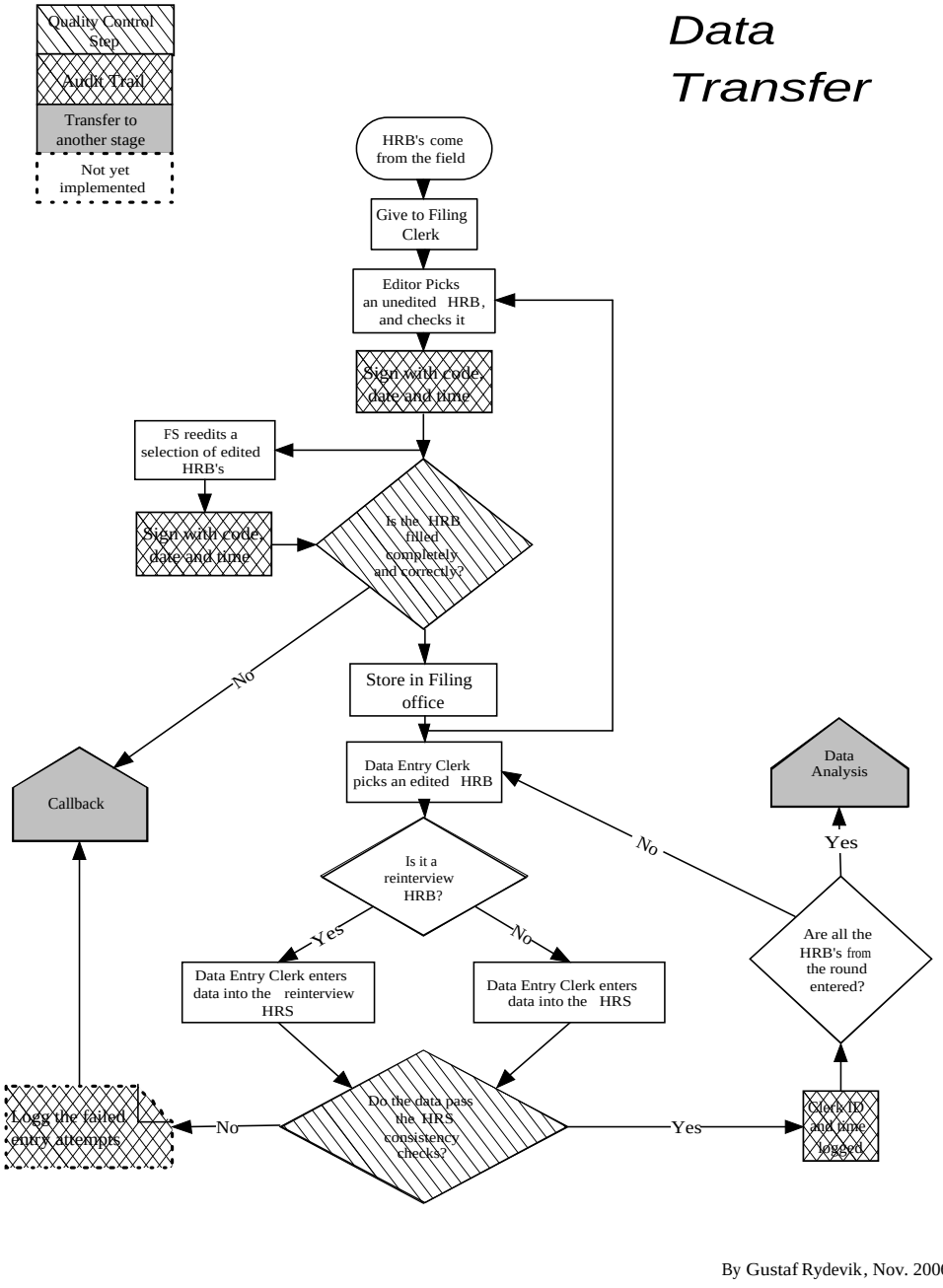


Figure 2: The data is then subjected to various checks before becoming a part of the data base.

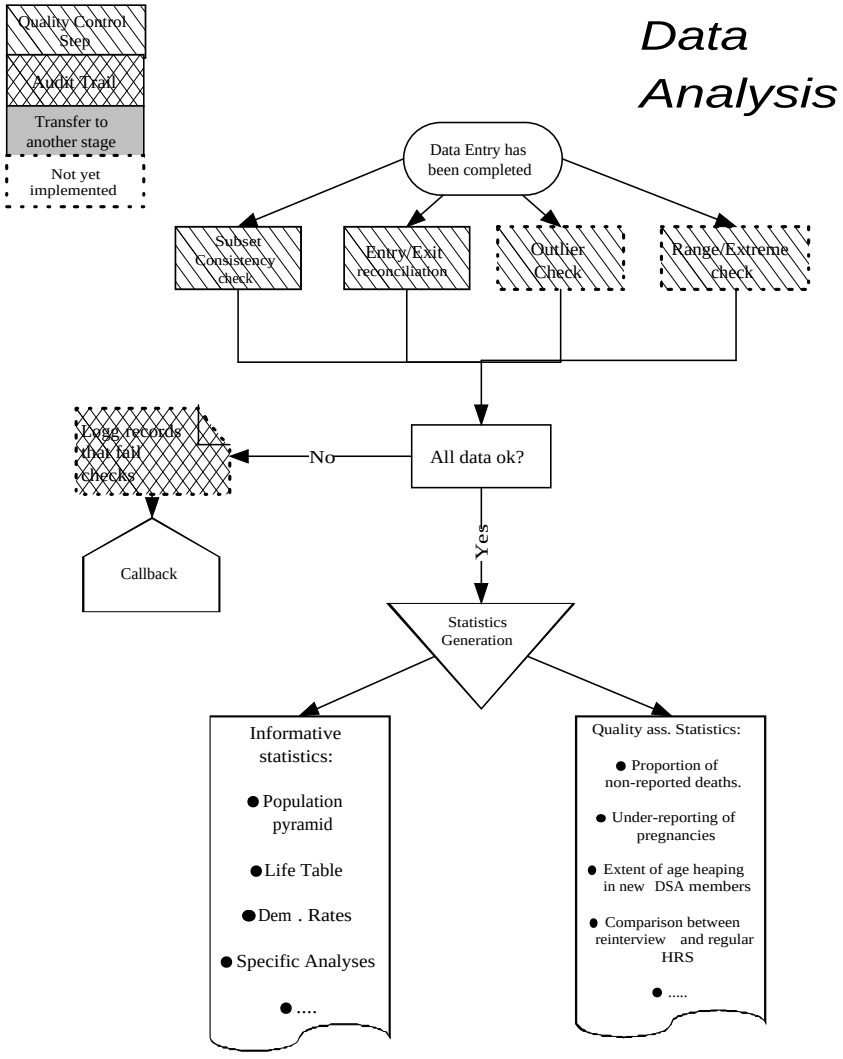


Figure 3: Additional quality analysis is conducted on the entire corpus of data before desired analysis is conducted.

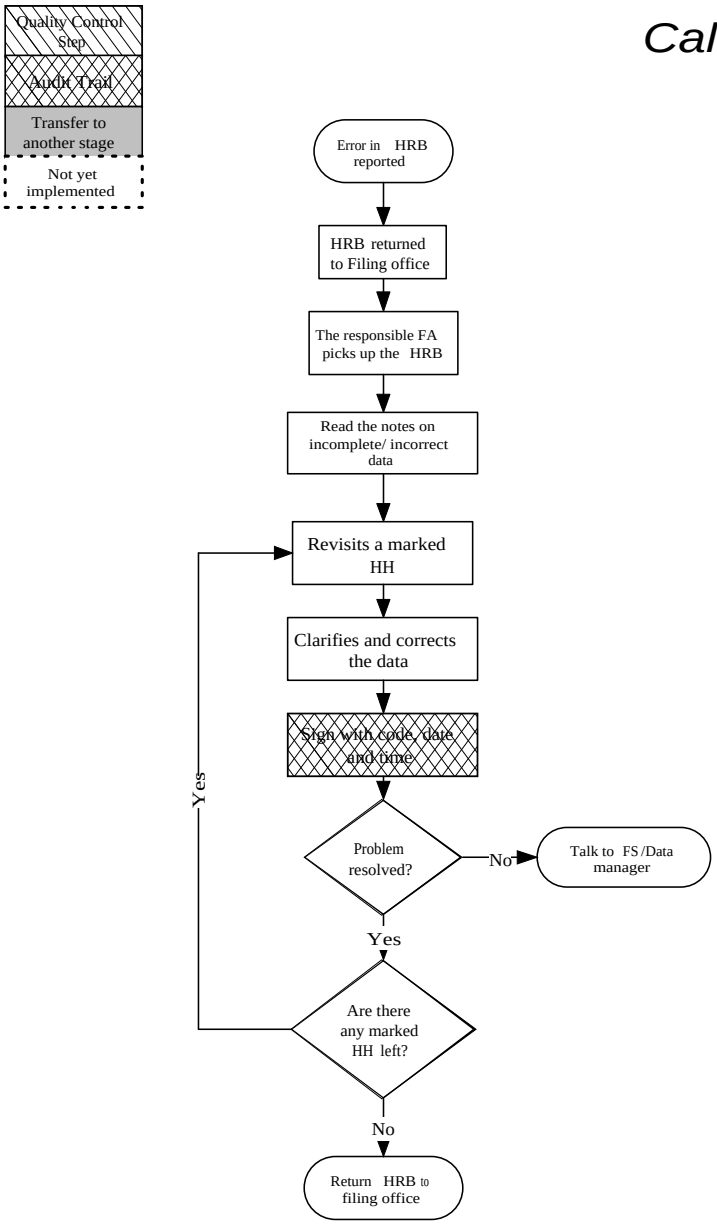


Figure 4: For any case where an error is suspected, additional field visits are conducted for to verify the data.

2 Work conducted during the Iganga DSS visit

2.1 Sampling of households selected for re-interviews

At the start of an update round, 4 % of the households assigned to each FA are sampled, using the FA_ sampling.do Stata procedure (See appendix and the upper part of fig. 1). The sampled households are collected into reinterview HRB's, with each TL getting an HRB of the sampled household his/her FA's are responsible for.

Then, during the collection period, each TL should conduct reinterviews of the households contained in the rHRB. Once the rHRB's are filled, the content is entered into a separate reinterview database, and used to get an estimate of the uncertainty of the variables. Additionally, each reinterview is used by the TL's for giving feedback to the FA on how their work is or should be conducted.

The reasons for stratifying according to FA before sampling were mainly practical: this allows each TL an equal amount of feedback for each FA, and it can be expected that there is one reinterview per FA per week, thus making it easier to set up the interviews as a weekly routine. In a first trial during round one, it was implemented in the middle of the round. Therefore, only the population left to be interviewed was subjected to sampling. Additionally, due to resource constraints, only about 1% were sampled. The end result was that only 45 household were selected for re-interviews, making it difficult to generate error rate estimates.

2.2 Demographic variable definitions

The INDEPTH network requires all members to produce certain basic demographical statistics. These include, among other things, a life table, fertility rates, and a population pyramid. The following definitions are taken from [Haupt, Kane, 2004].

Population pyramid

A population pyramid is a vertical histogram by age-group of observed person years, divided into males and females. The person years are calculated, for each individual, as the time he/she was observed during each specific age of life, starting from either the time of the baseline visit, or the time the individual came into the DSA (in-migration or birth). The observations stops either at the time of the round one visit or the time the individual stopped being a member of the DSA (out-migration or death). The person years are then aggregated by the ages they belong to, to serve as denominators for most of the calculated rates.

CDR-Crude Death Rate

Defined as the total number of deaths divided by the total number of person years observed.

Birth Rate

Defined as the number of births divided by observed person years.

GFR-General Fertility Rate

Defined as the number of births divided by the observed person years of women in fertile age (15-49 years).

AFR-Age specific Fertility Rate

Defined as the number of births to women in an agegroup, divided by the observed female person years within that age-group. Only women in fertile age (15-49 years old) are considered

TFR - Total Fertility rate.

This is calculated by taking the AFR of each age-group, multiplying by the numbers of years covered by that group, and adding together the numbers thus generated. It gives a measure of the number of children a woman could be expected to produce if subjected to the now prevalent AFR throughout her fertile life.

Neonatal, infant and under five mortality rates

These are calculated by dividing the number of deaths before 28 days, one year and five years of age, respectively, by the number of observed live births during the period.

Life Table

A life table is a set of variables relating to mortality that is compiled into a table. (Note: The following definitions taken from [CD Mathers et. al. 2001])

Dx

The observed number of deaths during the observation period, divided into age groups.

px

The number of person years observed, divided into age groups.

qx

This is calculated as $\frac{Dx_n}{px_n + n \cdot Dx_n \cdot a_n}$, where n is the length of the age interval, and gives the probability of death during an age interval. The denominator is the population at risk at mid-interval, adjusted for expected deaths. a_n is the proportion of deaths expected to occur after half

an interval, and is 0.5 for all intervals except the first, when it usually comes to 0.7.

lx

This gives the population remaining at the start of each age interval of a synthetic cohort of 100 000 people that is subjected to the now prevailing death rates during their course of life. It is calculated by $lx_n = lx_{n-1} \cdot (1 - qx_{n-1})$.

dx

This is the number of deaths that occur in the synthetic cohort during an interval, given by $dx = lx_{n+1} - lx_n$.

Lx

This is the number of person years lived by the synthetic cohort during each interval, adjusted for the calculated deaths. It is given by$Lx_n = n * (lx_n - dx_n * a_n)$, where n is the length of the interval. For the 85+ age-group, it is given by $lx_n/(Dx_n/px_n)$.

Tx

Tx is calculated by $Tx_n = \sum_n^{85} Lx_i$, and is the total number of remaining person years to be lived at the start of each age interval.

ex

ex is the life expectancy at the start of each interval, and is given by $ex_n = lx_n/Tx_n$. It is the number of person years left for each cohort individual.

2.3 Results of the demographic calculations

2.3.1 Population Pyramid

2.3.2 Fertility Rates

Birth Rate	GFR	TFR	Inf. mort.	Neo. mort.	< 5 mort.	CDR
.0292	.1356	4.256	.0336	.0196	.0836	.0039

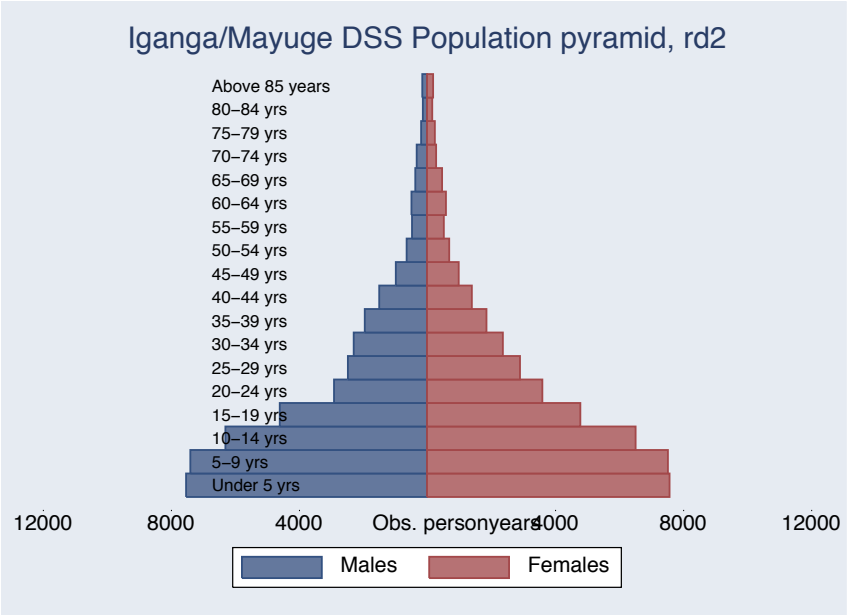


Figure 5: Person years observed, divided into males and females by agegroup.

Agegroup	**livebirths**	**px**	**AFR**
15-19 yrs	381	4789	.0796
20-24 yrs	692	3602	.1921
25-29 yrs	590	2904	.2032
30-34 yrs	452	2369	.1908
35-39 yrs	232	1857	.1249
40-44 yrs	75	1399	.0536
45-49 yrs	7	989	.0071

2.3.3 Life Tables

Male:

agegroup	Dx	px	qx	lx	dx	Lx	Tx	ex
Under 1 year	52	1442.0	.03517	100000	3517.3	97538	6253320	62.533
1-4 yrs	77	6084.1	.04913	96482	4740	340424	6155782	63.802
5-9 yrs	13	7393.3	.00875	91742.	803	456704	5815358	63.388
10-14 yrs	8	6298.3	.00633	90939	576	453258	5358654	58.926
15-19 yrs	9	4600.6	.00973	90363	880	449619	4905397	54.285
20-24 yrs	2	2904.7	.00344	89484	308	446652	4455777	49.794
25-29 yrs	8	2472.1	.01605	89176	1431	442305	4009126	44.957
30-34 yrs	14	2290.4	.03010	87745	2641	432123	3566821	40.650
35-39 yrs	16	1945.0	.04030	85103	3430	416945	3134698	36.834
40-44 yrs	15	1492.8	.04901	81674	4003	398364	2717753	33.276
45-49 yrs	16	975.64	.07877	77671	6118	373061	2319389	29.861
50-54 yrs	8	637.19	.06087	71553	4355	346879	1946328	27.201
55-59 yrs	5	468.93	.05192	67198	3490	327267	1599449	23.802
60-64 yrs	7	486.70	.06942	63709	4422	307487	1272182	19.969
65-69 yrs	11	366.72	.13951	59286	8271	275753	964695	16.272
70-74 yrs	15	323.30	.20787	51015	10604	228563	688942.3	13.504
75-79 yrs	11	184.32	.25965	40410	10492	175821	460378.8	11.393
80-84 yrs	11	130.72	.34762	29918	10400	123587	284558.4	9.5114
Above 85 years	18	148.45	1	19518	19518	160970	160969.8	8.2474

Female:

agegroup	Dx	px	qx	lx	dx	Lx	Tx	ex
Under 1 year	46	1434.4	.03136	100000	3136	97804	6778425	67.784
1-4 yrs	69	6141.3	.04376	96864	4239	346761	6680620	68.969
5-9 yrs	11	7523.91	.00728	92625	675	461437	6333859	68.382
10-14 yrs	3	6514.7	.00230	91950	211	459222	5872423	63.865
15-19 yrs	8	4788.6	.00832	91739	763	456785	5413201	59.007
20-24 yrs	11	3602.1	.01515	90975	1379	451431	4956416	54.48
25-29 yrs	9	2904.0	.015377	89597	1378	444540	4504985	50.281
30-34 yrs	13	2368.7	.02707	88219	2388	435126	4060445	46.027
35-39 yrs	9	1856.9	.02394	85831	2055	424018	3625319	42.238
40-44 yrs	10	1399.3	.03510	83776	2941	411527	3201301	38.213
45-49 yrs	6	989.47	.02987	80835	2414	398139	2789774	34.512
50-54 yrs	9	692.99	.06289	78421	4932	379773	2391635	30.497
55-59 yrs	3	524.34	.02820	73489	2073	362261	2011862	27.377
60-64 yrs	7	590.40	.05757	71416	4112	346800	1649601	23.099
65-69 yrs	10	470.20	.10097	67304	6796	319532	1302801	19.357
70-74 yrs	12	283.92	.19113	60509	11565	273630	983269.2	16.250
75-79 yrs	12	244.51	.21857	48944	10698	217973	709639.1	14.499
80-84 yrs	9	162.39	.24339	38245	9309	167957	491665.9	12.855
Above 85 years	17	190.17	1	28937	28937	323709	323708.9	11.187

2.4 Discussion of the results of the demographic calculations

The population pyramid for Iganga round 1 looks as expected, for an area in Uganda. It has a very broad base, signifying a young population experiencing very high growth rates, and a very narrow peak, meaning that death rates are quite high. There are two minor surprises however. The first one is the indent at ages 55-59. This can possibly be related to Idi Amin's rule, during the years 1971-1978. The other surprise is the population aged below five years of age. The jump between this step and the next is much less than expected. Two possible reasons are:

1: The fertility rates have decreased significantly within the past five years. Reasons could be increased knowledge about family planning methods, or increased standards of living. However, there is nothing except the pyramid signifying this to be the case.

2: The DSS are missing many young children when conducting data collection. The missed ones could be newborns that the respondent forgets about, or children that are considered unimportant for one reason or another. In any case, the issue merits further investigation.

The fertility figures were also a source of surprise. The Ugandan national TFR is 6.71 , and the birth rate is 0.04735 [CIA,2006]. In the light of this, the figures for the DSS are suspiciously low, though still high if compared to developed countries. Again, the reasons could either be unknown factors that affect the women within the DSA, or unreported births during the update visits. A third reason could be that the national figures are overestimating the TFR, although this seems unlikely.

As could be guessed from the above results, the life tables exhibit certain unexpected characteristics as well. The national life expectancy for Uganda is 52 years for men and 54 years for women [Ibid.]. A result of 62.5 years for men and 67.8 years for females is therefore much higher than expected. While these numbers could be reasonable, seeing as the Iganga DSS lies in the eastern part of the country which is not affected by the civil war instigated by the Lord's Resistance Army in the north, the very high life expectancies at higher ages (i.e. 8/11 years at age 85) are probably spurious. Since the numbers of deaths observed at ages above 55 years are low, the estimate of ex is clearly vulnerable to random fluctuations. Most likely, the numbers for ex at higher ages will drop as more deaths and person years are observed.

2.5 Indications of data quality problems

In the light of the above findings, three indices measuring differing aspects of data quality were calculated. Whipple's age heaping index measures the extent to which reported ages between 25 and 60 are concentrated at "nice" ages, ages that ends in a five or a zero [UN populations Division 2003(?)]. Let $x \mid y$ denote

"x divides y". The index is then calculated as

$$\frac{(\sum_{i=23}^{62} \#(agegroup_i) \cdot I(5|i) \cdot 500)}{\sum_{i=23}^{62} \#(agegroup_i)}).$$

The UN classifies data into five categories as follows:

Classification	Value of Whipple's
I. Highly accurate data:	Less than 105.
II. Fairly accurate	105 - 109.9
III. Approximate data:	110 - 124.9
IV. Rough data:	125 - 174.9
V. Very rough data:	175 and more.

The value calculated for the data at Iganga/Mayuge DSS was 124.62, meaning the age figures are somewhat rough.
A similar index is the Myer's blended index[Spiegelman, 1955]. Briefly, it calculates the distribution of ages ending with each of the digits zero to nine, adjusted for the skewness inherent in such data. It then calculates half the sum of absolute deviations from ten percent for the different digits, to give an index indicating the percentage of ages that are misreported. For our data, the value of the index was 5.36. Since the data seem to indicate missed deaths, the Brass Growth Balance method[CD Mathers et. al. 2001] was used to give a rough estimate of the amount of underreporting. This method assumes that population growth rates have been reasonably stable, that migration is negligible, and that an equal proportion of deaths go unreported for all age groups. The growth rate of an open-ended age segment in a population is equal to the rate of entry to the segment (people whose age increase above the limit age, and inmigration) minus the rate of exit (i.e. death or outmigration). Given the assumptions above, the growth rate should be equal for all open-ended age groups, and the relationship between exit and entry rates should be linear with a slope of one. If there is an underreporting of deaths, it will manifest itself as an increase in the slope. The inverse of the slope gives an estimate of the proportion of deaths missed.

When entry and exit rates are calculated for the DSS data, there seem to be a linear trend only for the part of the population aged 50 or higher. Below this age, the numbers vary quite a lot, and no trend is apparent. Because of this, a line was not fitted to the data. The absence of a linear trend probably means that the assumptions are violated. Either there is a significant migration at ages below fifty, or the fertility rate (growth rate) in Iganga has changed to some extent in the previous years.

As a final indication, if one looks at the number of very old individuals, it becomes obvious that these persons have a strong tendency to report too high ages. There are 35 persons above age 100 in the DSA, which gives 4.5 per 10 000 persons. Compare this number with that of Sweden, which has 1.5 per 10 000 persons [SCB 2006].

In summary, the following are indicators of problems with the data quality at the Iganga-Mayuge DSS:

- The fertility rates are very low compared with national numbers, indicat-

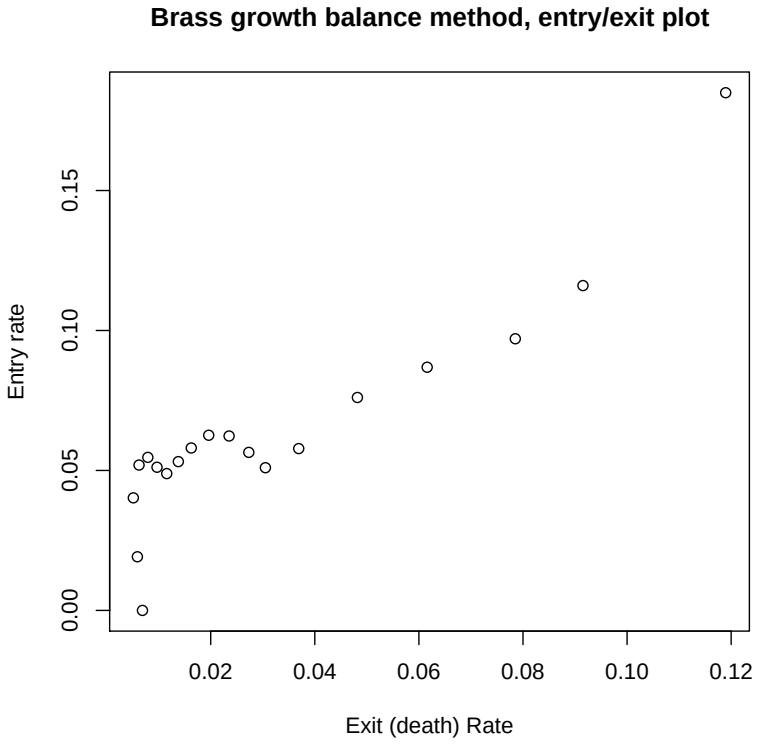


Figure 6: The graphs shows the fairly non-linear relationship between entry and exit rates, indicating that the growth rate is non-constant for differing agegroups in the data.

ing that births are being missed.

- Likewise, the mortality rates are low, with very high life expectancy as a result.

- Whipple's and Myer's indices indicate significant age misreporting. This might affect the mortality rates at young ages (by giving too few persons in the under-5 age group), and is generally indicative of difficulties in collecting correct quantitative data.

- There is a tendency to shift ages upwards at the old ages, which might affect the number of persons above 85 years old, and in turn could increase the estimated life expectancy.

2.6 Missed pregnancies

The African Population and Health Research Centre [Woubalem et. al., 2006],gave a presentation during the annual INDEPTH meeting. Inspired by this presentation, we calculated the ratio between births where the mother's pregnancy had been recorded, and the births where the pregnancies had been missed to be recorded. Those women whose previous visit where more than 256 days before the date of birth were removed. The remaining data were then cross-tabulated against various covariates. Finally, using the risk of a pregnancy being missed, estimates and confidence intervals for quotients between various groups were calculated (so called "Relative Risk Ratios").

Results: Frequency of underreporting by different covariates

```
. tab pregnancy if pregNoticable==1

    Was the |
  pregnancy |
   reported |
     during |
    visit?  |      Freq.     Percent        Cum.
------------+-----------------------------------
         No |        784       36.84       36.84
        yes |      1,344       63.16      100.00
------------+-----------------------------------
      Total |      2,128      100.00
```

```
Gender of the fieldworker

           |   Was the pregnancy
           |    reported during
Fieldworke |        visit?
  r gender |        No        yes |     Total
-----------+----------------------+----------
         F |       511        877 |     1,388
           |     36.82      63.18 |    100.00
-----------+----------------------+----------
         M |       273        467 |       740
           |     36.89      63.11 |    100.00
-----------+----------------------+----------
     Total |       784      1,344 |     2,128
           |     36.84      63.16 |    100.00

          Pearson chi2(1) =   0.0012   Pr = 0.972
```

Gender of the respondent

Gender of the respondent	Was the pregnancy reported during visit? No	yes	Total
F	469	966	1,435
	32.68	67.32	100.00
M	315	378	693
	45.45	54.55	100.00
Total	784	1,344	2,128
	36.84	63.16	100.00

Pearson chi2(1) = 32.7592 Pr = 0.000

Respondent vs FW gender

resp_fwsex	Was the pregnancy reported during visit? No	yes	Total
Both Male	101	150	251
	40.24	59.76	100.00
Both Female	297	649	946
	31.40	68.60	100.00
FW Female & Responden	214	228	442
	48.42	51.58	100.00
FW Male & Respondent	172	317	489
	35.17	64.83	100.00
Total	784	1,344	2,128
	36.84	63.16	100.00

Pearson chi2(3) = 39.3376 Pr = 0.000

Education level of respondent

High class

grouped in 5 different levels for all individuals in DSS	Was the pregnancy reported during visit? No	yes	Total
Never	86	183	269
	31.97	68.03	100.00
Lower Primary	146	285	431
	33.87	66.13	100.00
Upper Primary	302	581	883
	34.20	65.80	100.00
O'level	170	256	426
	39.91	60.09	100.00
Higher	15	18	33
	45.45	54.55	100.00
Total	719	1,323	2,042
	35.21	64.79	100.00

Pearson chi2(4) = 7.6045 Pr = 0.107

The household position of the respondent

What was the position of the respondent in the household?	Was the pregnancy reported during visit? No	yes	Total
The pregnant woman	344	846	1,190
	28.91	71.09	100.00
Not a member	45	16	61
	73.77	26.23	100.00
Other relationship	77	97	174
	44.25	55.75	100.00
Head of Household	318	385	703
	45.23	54.77	100.00
Total	784	1,344	2,128
	36.84	63.16	100.00

```
          Pearson chi2(3) =  93.3345   Pr = 0.000
```

Age of the mother

Was the mother over 18?	Was the pregnancy reported during visit? No	yes	Total
No	51	50	101
	50.50	49.50	100.00
yes	733	1,294	2,027
	36.16	63.84	100.00
Total	784	1,344	2,128
	36.84	63.16	100.00

```
          Pearson chi2(1) =   8.4941   Pr = 0.004
```

Distribution of outcomes of the pregnancy

Outcome of pregnancy	Was the pregnancy reported during visit? No	yes	Total
Stillbirth	3	19	22
	13.64	86.36	100.00
Live birth	772	1,297	2,069
	37.31	62.69	100.00
Misscarriage	9	28	37
	24.32	75.68	100.00
Total	784	1,344	2,128
	36.84	63.16	100.00

```
          Pearson chi2(2) =   7.7800   Pr = 0.020
```

Distribution of neonatal deaths

Did the child die	Was the pregnancy

```
 before 28 |    reported during
   days of |        visit?
      age? |        No        yes |     Total
-----------+----------------------+----------
        No |       782      1,326 |     2,108
           |     37.10      62.90 |    100.00
-----------+----------------------+----------
       yes |         2         18 |        20
           |     10.00      90.00 |    100.00
-----------+----------------------+----------
     Total |       784      1,344 |     2,128
           |     36.84      63.16 |    100.00

          Pearson chi2(1) =   6.2516   Pr = 0.012
```

Predicted Relative Risk Ratios of reporting pregnancies, with confidence intervals

Variables	RRR	95 % Confidence Interval	
Fieldworker gender(ref is female)			
male	1.00	0.83	1.20
Respondent gender (ref is female)			
male	0.58	0.48	0.70
FW vs resp gender(ref is both female)			
Both Male	.68	0.51	0.91
FW female, resp male	0.49	0.39	0.61
FW male, resp female	0.84	0.67	1.06
Resp. education level (ref is upper primary)			
No Education	1.11	0.83	1.48
Lower Primary	1.01	0.79	1.29
O' level	0.78	0.62	0.99
Higher Ed.	=.62	0.31	1.26
Household position (ref is the pregnant woman)			
Not a member	0.14	0.08	0.26
Other relation	0.51	0.37	0.71
Head of HH	0.49	0.40	0.60
Age of mother (ref is over 18)			
Under 18	0.56	0.37	0.82
Pregnancy outcome (ref is livebirth)			
Stillbirth	3.77	1.11	12.78
Miscarriage	1.85	0.87	3.94
Neonatal death (ref is no death)			
Neonatal death	5.30	1.22	22.94

Discussion of the pregnancy report analysis

There are a number of conclusions that can be drawn from above statistics. The first, and main result is that 36 % of the pregnancies went unreported, only being registered once an outcome had occurred. While this is a high number, it is nevertheless better than the situation reported from APHRC, where as many as half of the pregnancies were missed. The second conclusion is that pregnancies in under-aged women are significantly more likely to be missed. Reasons could be feelings of shame in the women, the unexpectedness of such pregnancies, or other factors. Whatever the reasons, this result is problematic because it is well known that the incidences of abortion and miscarriage are much higher in younger women, which would lead us to miss a disproportionate number of such occurrences.

Yet another, somewhat surprising conclusion is that the frequency of reported pregnancies is higher if the field worker interviews someone of the same gender as him/herself. In fact, the reporting rate for male/male interviews is almost as high as when looking at only those interviews were a female was the respondent. While it might be difficult to act according to this information in actual fieldwork, it does raise interesting questions. Possibly, the level of trust a respondent feel towards the interviewer is higher when they share the same gender. If that is the case, the same situation might apply at other times when sensitive information is collected (for example on socioeconomic status).

Finally, the most revealing statistic is that related to the outcome of the pregnancy. The reason for conducting this analysis was a suspicion that unreported pregnancies might lead to an underreporting of prenatal mortality. The numbers prove unequivocally that such is the case. All but one of the reported cases of prenatal mortality in the database had a pregnancy status registered previously. The only reasonable explanation for this is that prenatal deaths are not reported unless specifically asked for. A registered pregnancy in the previous round prompts the field worker to ask for the outcome of the pregnancy. If no such prompt exists, specific questions about miscarriages or stillbirths are not asked. Instead, a general question about new household members is asked. Obviously, looking at the numbers, this is not enough for the respondent to volunteer information about those cases where a new household member almost came into being.

To get a feeling for the magnitude of the problem, a rough estimate of the number of failed pregnancies that went unnoticed was calculated. Under the assumption that the incidence of failed pregnancies is equal among missed and non-missed pregnancies, we get that about 18 cases were missed. While this is not a very large number, it is still significant at about 40% of the number of reported cases. Such a difference might well affect the decisions on where to best focus efforts in order to save lives.

2.7 Permutation method for finding discrepant Field Assistants

Using Stata, We started looking at how the deaths were distributed between different field assistants. We started by counting the number of death events reported by each FA, and then all the individuals in the database that had been entered by each FA. The time period in both cases was during the first update round. At first, the distribution of reported deaths seemed to have a large spread, but to be reasonable, going between 0 and up to 30. In order to standardize the numbers, we calculated the quota, and several outliers appeared. Seven FAÕs had a ratio of 1 in 15 or above, while all the others were on the order of one in one hundred.

By calculating the time between the first and the last entry into the database for each FA, a pattern started to appear. All of the outlying FA's had very few individuals to their name, and all had worked less than half a year during the time of round one. Thus, it was a case of FA's letting trainee fieldworkers handle the interesting cases, where events had occurred. However, the finds highlighted a need for detecting field workers submitting unusual data.

One way of detecting such field workers is by a so-callled permutation test. The procedure goes as follows:

- A list of all observations (on a houehold level), whether each observation captured an event, and the FW that did the observation is generated.
- The FW labels are permutated, so that each observation is assigned a new FW.
- The minimum ratio of events to observations, by FW, is calculated
- The procedure is repeated between 500-3000 times
- The minimum ratio of the original data is compared to the set of minimum ratios of permutated data, and the percentage of generated ratios that are lower is calculated.

The percentage will give an indication if any fieldworker is reporting a suspiciously low event rate. If it is 5% or less, that FW with the lowest should probably be investigated, since that would indicate that something is affecting the rates in his group of observations. It could be that the data is faked, or that the region he is assigned have special characteristics. In either case, it is important to know about it. The procedure was tested using deaths in households as the event to be investigated. 3000 permutations were done using a purpose-written routine in the R statistics program. The minimum rate rate in the original data was 8.9 dead/1000 observations. Of the calculated rates using permuted data, 13.57% were lower than the observed (see Fig.7). Therefore, nothing was done. However, the fact that a fairly low percentage was generated is indicative of a functional procedure. It is therefore recommended that this included in standard quality control measures, to safeguard against field workers missing large numbers of events.

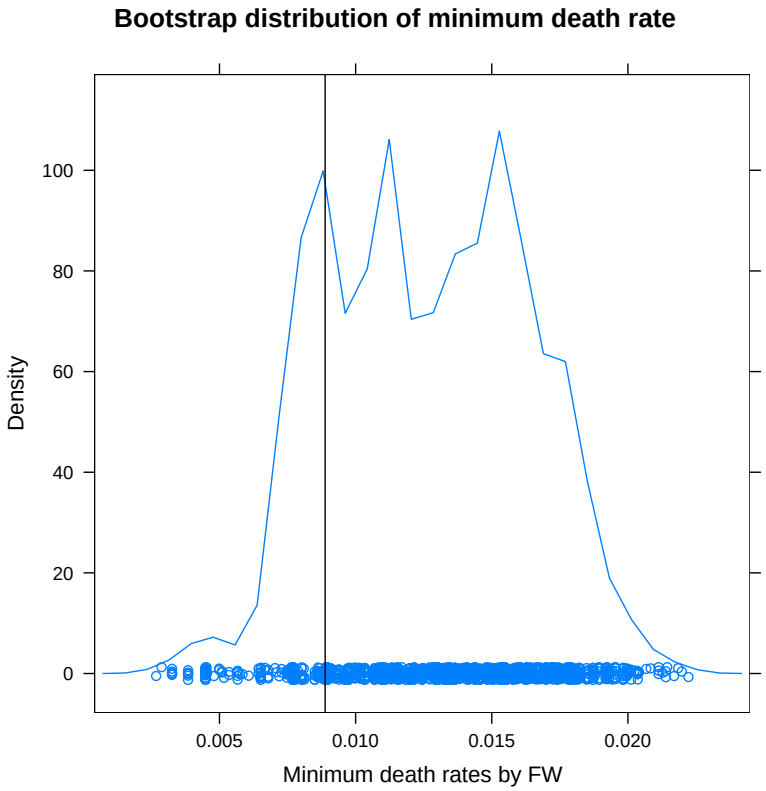


Figure 7: The generated distribution of the minimum death rate. The line marks the observed 8.9/1000 dead.

References

[INDEPTH 2007] *INDEPTH Network*, http://www.indepth-network.org

[UBOS, 2000] *Uganda National Household Survey 1999/2000-Report on the Socioeconomic*, regional supplement. Uganda Bureau of Statistics 2000.

[Mcleod et al, 2000] *The Household Registration System: Computer Software for the Rapid Dissemination of Demographic Surveillance Systems.* James F. Phillips, Bruce B. MacLeod, Brian Pence. Demographic res. 2,6 (2000)

[Haupt, Kane, 2004] *Population Reference Bureau's Population Handbook, 5th Edition.* Arthur Haupt and Thomas T. Kane

[CIA,2006] *US CIA World Factbook, 2006.* https://www.cia.gov/cia/publications/factbook/geos/ug.html

[UN populations Division 2003(?)] *UN Demographic Yearbook,* special topics volume 1, table 1c. Can be found at: http://unstats.un.org/unsd/demographic/products/dyb/dybcens.htm

[Spiegelman, 1955] *Introduction to Demography.* M. Spiegelman 1955.

[SCB 2006] *Tables on population in Sweden 2005.* SCB, 2006.

[Woubalem et. al., 2006] *Challenges in collecting data on Pregnancy in NUHDSS.* Z. Woubalem, E. Wekesa, A. Ezeh, E. Zulu, N. Madise. Presentation, INDEPTH annual meeting, 2006.

[Anderson,1999] *Method for constructing complete annual U.S. life tables.* RN Anderson, National Center for Health Statistics. Vital Health Stat 2(129). 1999.

[CD Mathers et. al. 2001] *National Burden of Disease Studies: A Practical Guide.*Mathers CD, Vos T, Lopez AD, Salomon J, Ezzati M (ed.) Edition 2.0. Global Program on Evidence for Health Policy. Geneva: World Health Organization. 2001

3 Appendix

3.1 .do-files

```
RoutineCollectionReadme.txt

A collection  of Stata and R routines  for standard, end-of-round analysis of DSS data.
-----------------------------------------------------------------------

This directory is supposed to contain the following files:

PregnancyReport.do
Life_table_5yr1.do
Age_heaping_and_Brass.do
childmort_fertility_poppyr.do
px_calculator.do
AnalysisSettings.do
FA_sampling_rd2.do
sample1.1.do

AnalysisSettings.do should be edited to fit your preferences, and placed in
the working directory of your Stata program.
All the .do files have more or less commented code,
so it should be possible to figure out what they're doing.


With many thanks to all the people at Iganga/Mayuge DSS,
```

```
Gustaf Rydevik, December 2006.

Comments or other communications can be sent to gustaf.rydevik@gmail.com.

---------------------

// AnalysisSettings.do

/*This is the main preference file. Should be saved in Stata's working directory to work.
All directories should end with a slash to be interpretated correctly.
 */


glo startTime= mdy(1,1,2005)  //From what date should observations be included?
glo  endTime=mdy(12,8,2006) // Until what date should observations be included?
glo startRound=0 // Which round do we begin analysis at?

glo endRound=2
// And which round do we end it? Note that the start and end round should be
 //different for meaningful analysis.

glo dodir="E:\Iganga-data\Stata-data\RoutineCollectionIMDss\"
// Where is the RoutineCollection placed?

glo outdir="E:\Iganga-data\Stata-data\prelrd2results13thdec\"
//Where should results be stored?

glo logdir="E:\Iganga-data\Stata-data\RoutineCollectionIMDss\logs\"
// Where should logs be stored?

glo tempdir="E:\Iganga-data\Stata-data\RoutineCollectionIMDss\temp\"
//Where should temporary files be stored?

glo datadir="C:\preliminary_round2_data\"
//Finally, Where is the data to be analyzed stored?

-----------------------

// do_all_analysis.do

set more off, permanently

global all_analysis=1
do AnalysisSettings.do

do ${dodir}px_calculator.do

do ${dodir}PregnancyReport.do
```

```
do ${dodir}Life_table_5yr1.do

do ${dodir}Age_heaping_and_Brass.do

do ${dodir}childmort_fertility_poppyr.do

do ${dodir}FA_sampling_rd2.do

macro drop all_analysis
set more on, permanently

-----------------------

// age_heaping_and_brass.do

/* Generates a measure of age-heaping as well as a data set prepared for analysis of
mortality-under reporting by the Brass
Growth Balance Method, as described in National Burden Of Disease Studies, WHO 2001.
By Gustaf Rydevik, November 2006.
*/

do AnalysisSettings.do //Makes sure that the global parameters are set.

log using ${logdir}Brass.log, replace

/* The following sorts the death data set, and limits it to
 observations inside the analysis time span. */

clear
use ${datadir}death
drop if dthdate < $startTime | dthdate > $endTime
drop if round< $startRound | round> $endRound
sort individid
save ${tempdir}temp3.dta, replace


//The following generates person-years and death indicators for all individuals.

local test=$all_analysis+1
if ‘test’==2 {
use ${tempdir}pyoIndividual.dta
}
else {
do ${dodir}px_calculator.do
}
sort individid
merge individid using ${tempdir}temp3.dta,keep(dthdate) nokeep
bys individid (age): gen Dx=1 if dthdate~=. & _n==_N
save ${tempdir}temp3.dta,replace
```

```
// The following generates the age at first observation of all individuals

use ${datadir}observation //
drop if date < $startTime | date > $endTime
drop if round< $startRound | round> $endRound
keep date round locationid
rename date obsdate
gen str2 obsround=string(round,"%02.0f")
drop round
reshape wide obsdate, i(locationid) j(obsround) string
sort locationid
save ${tempdir}temp.dta,replace

clear
use ${datadir}individual
sort individid
save ${tempdir}temp2,replace
clear

use ${datadir}residency,clear
egen firstsdate=min(sdate), by(individid)
drop if sdate~=firstsdate
drop sdate
sort locationid
merge locationid using ${tempdir}temp.dta, nokeep
drop _merge
foreach V of var obsdate* {
replace `V'=. if `V'<firstsdate
}

egen firstobs=rmin(obsdate*)
drop obsdate*
bys individid (firstobs): drop if _n~=1

sort individid
merge individid using ${tempdir}temp2.dta, keep(birth_date)
gen age = int((firstobs-birth_date)/365.25)
egen Age=min(age), by(individid)
drop if age~=Age
drop if age<0|age==.
drop Age
gen reported=1

/* Now, we calculate the numbers of persons reported as a specific age, and then generate
s Whipple's index as
Described in Special Topics Volume 1, table 1c, UN publications. */

collapse (sum) reported, by(age)
egen temp=sum(reported) if age>22 & age<63
```

```
egen temp2=sum(reported)  if  5*int(age/5)==age &age>22 & age<63
gen Whipple=temp2*500/temp
drop temp temp2


/* And then, Myer's Blended index is calcualted, as described in Introduction to Demography,
 by Mortimer Spiegelman */

gen enddigit= age-int(age/10)*10
egen myers10= sum(reported) if age>=10&age<100, by(enddigit)
egen myers20= sum(reported) if age>=20&age<110, by(enddigit)
gen myersblend=(myers10*(enddigit+1)+myers20*(9-enddigit))/10
bys enddigit (age): replace myersblend=myersblend[3]
sort age
egen temp=sum(myersblend) if age<10.
replace temp=temp[1]
gen myersprop= myersblend/temp
drop temp
gen temp=abs(myersprop-.10)*100
egen myersindex= sum(temp) if age<10
drop myers10 myers20 temp enddigit
replace Whipple=Whipple[31]
replace myersindex=myersindex[1]
replace myersindex=myersindex/2
l
save ${outdir}ageindex.dta, replace

//Finally, we prepare a data set for analysis acc. to Brass

use ${tempdir}temp3.dta
collapse (sum)Dx px, by(agegroup)
gen N_exact= (px[_n-1]+px[_n])/10
egen totalpx=sum(px)
gen N_plus=totalpx-sum(px[_n-1])
egen totalDx=sum(Dx)
gen Dx_plus=totalDx-sum(Dx[_n-1])
gen Xi=Dx_plus/N_plus
gen Yi=N_exact/N_plus

drop totalDx totalpx
gen byte mysubset= agegroup >=1825 & agegroup <=10950
gen byte mysubset2= agegroup >10950 & agegroup <=21800

sum Yi if mysubset==1
local ymean0=r(mean)
sum Yi if mysubset2==1
local ymean1=r(mean)
sum Xi if mysubset==1
local xmean0=r(mean)
sum Xi if mysubset2==1
```

```
local xmean1=r(mean)
local slope=(‘ymean1’-‘ymean0’) / (‘xmean1’-‘xmean0’)
gen K= (‘ymean1’-‘ymean0’)/(‘xmean1’-‘xmean0’)
save ${outdir}Brass.dta, replace
log close

--------------------------
// birth_quality_coordinates.do

do AnalysisSettings.do
log using ${logdir}Birth_coordinates.log,replace
clear
use ${datadir}villages
sort villagecod
save ${tempdir}temp2, replace

clear
use ${datadir}location
gen gridn=floor(northings/1000)*1000
gen gride=floor(eastings/1000)*1000
gen gridID=gridn+gride/1000
keep locationid eastings northings gridn gride gridID
sort locationid
save ${tempdir}temp.dta,replace

clear
use ${datadir}residency
bys individid (edate): drop if _n~=_N
sort locationid
merge locationid using ${tempdir}temp.dta
keep individid locationid  eastings northings gridn gride gridID
gen villagecod=substr(locationid,1,6)
sort villagecod
merge villagecod using ${tempdir}temp2, keep(villagenam)
drop _merge
sort individid
save ${tempdir}temp.dta, replace

use pregoutcome
gen qualified=1 if birthatten=="3"
replace qualified=0 if qualified==.
keep individid type date birthatten prgplace qualified
sort individid
merge individid using ${tempdir}temp.dta, nokeep
save ${outdir}Birth_quality_coordinates.dta
log close

--------------------

// childmort_fertility_poppyr.do
```

```
/* Do-file for generating fertility rates of the DSS population.
Also generates a population pyramid.
Require a more-or-less complete sets of stata files, generated by the HRS
in the data-directory.
*/

log using ${logdir}Mortality_fertility_population.log, replace

do AnalysisSettings.do
clear
set memory 20m

local test=$all_analysis+1
if ‘test’==2 {
use ${tempdir}pyoIndividual.dta
}
else {
do ${dodir}px_calculator.do
}

//Fertility-related rates


// Prepares pregoutcome.dta for analysis
clear
use ${datadir}pregoutcome
drop if type~="LBR"
drop if date < $startTime | date > $endTime
drop if round< $startRound | round> $endRound
keep individid date livebirths
rename date child_bd
sort individid
save ${tempdir}temp.dta, replace

//Prepares death.dta for analysis.
use ${datadir}death
drop if dthdate < $startTime | dthdate > $endTime
drop if roundnum< $startRound | roundnum> $endRound
keep individid dthdate
sort individid
save ${tempdir}temp2.dta,replace


//Attaches indicators for death, under-5 death, infant death, and neonatal death to individuals.
use ${tempdir}pyoIndividual.dta
sort individid
merge individid using ${tempdir}temp2.dta
drop _merge
sort individid
```

```
merge individid using ${tempdir}temp.dta
replace livebirths=0 if livebirths==.
bys individid (age): gen Dx=1 if dthdate~=. &_n==_N
bys individid (age): gen Dx_infant=1 if age <=365 & dthdate ~=. &_n==_N
bys individid (age): gen Dx_neonatal=1 if age <=28 & dthdate ~=. &_n==_N
bys individid (age): gen Dx_under5=1 if age <=1825 & dthdate ~=. &_n==_N
bys individid (age):replace livebirths=0 if child_bd>=endInterval |child_bd<beginInterval

/* Now, we generate all the interesting mortalityrates,
 using livebirths as denominators for all but the Crude Mortality Rate. */

collapse (sum) Dx_infant Dx_neonatal Dx_under5 Dx livebirths px, by(agegroup)
egen temp=sum(Dx_infant)
egen temp2= sum(Dx_neonatal)
egen temp3=sum(livebirths)
egen temp4=sum(Dx_under5)
egen temp5=sum(Dx)
egen temp6=sum(px)
gen Infant_mort=temp/temp3
gen Neonatal_mort=temp2/temp3
gen Under5_mort=temp4/temp3
gen Crude_death_rate=temp5/temp6
keep agegroup Infant_mort Neonatal_mort Under5_mort Crude_death_rate
sort agegroup
save ${tempdir}temp2.dta, replace

//And then the fertility-related rates. Attach birth-indicators...

use ${tempdir}pyoIndividual.dta
sort individid
merge individid using ${tempdir}temp.dta
replace livebirths=0 if livebirths==.
bys individid (age):replace livebirths=0 if child_bd>=endInterval |child_bd<beginInterval
save ${tempdir}temp.dta, replace
egen temp=sum(livebirths)
egen temp2=sum(px)
gen Birth_rate=temp/temp2
drop temp temp2
drop if gender=="M" |agegroup<=3650|agegroup>=18250

// ...and then caclulate rates,using the person years of females in fertile age as denominator

collapse (sum) livebirths (sum) px (median) Birth_rate,by(agegroup)
egen temp=sum(livebirths)
egen temp2=sum(px)
gen GFR=temp/temp2
drop temp temp2
gen AFR=livebirths/px
egen TFR=sum(AFR*5)
sort agegroup
```

```
merge agegroup using ${tempdir}temp2.dta
drop if _merge~=3
drop _merge
l
save ${outdir}fertility_childmortRD${endRound}.dta, replace



// Finally, we generate a population pyramid for the entire population.
clear
use ${tempdir}pyoIndividual.dta
lab var agegroup "Age strata"
gen px_male= px if gender=="M"
gen px_female=px if gender=="F"
replace agegroup=0 if agegroup==365
// Note that all agegroups must be equal in length for a sensible graph.

lab def A_group 0 "Under 5 yrs",modify
collapse (sum) px_male px_female, by(agegroup)
drop if agegroup==.


//The following code is from http://www.ats.ucla.edu/stat/stata/Library/GraphExamples/

gen zero = 0
gen offset = -7000
gen temp=-px_male
#delimit ;
twoway
bar temp agegroup, horizontal xvarlab(Males) barw(1825)
||
  bar px_female agegroup, horizontal xvarlab(Females) barw(1825)
||
sc agegroup offset                , mlabel(agegroup) mlabcolor(black) msymbol(i)
||

,
xtitle("          Obs. personyears") ytitle("")
plotregion(style(none))
ysca(noline) ylabel(none)
xsca(noline titlegap(-3.5))
xlabel(-12000 "12000" -8000 "8000" -4000 "4000" 4000 8000 12000, tlength(0))
legend(label(1 Males) label(2 Females)) legend(order(1 2))
title("Iganga/Mayuge DSS Population pyramid, rd${endRound}")

;

#delimit cr

graph export ${outdir}PopulationPyramidRound${endRound}.eps, as(eps) replace
```

```
log close
-----------------

//FA_sampling_rd2.do

/* A Routine for Sampling the households to reinterview,
 stratified according to Field Assistants areas of responsibility.
 Written by Gustaf Rydevik 5th October 2006.
Adjusted for the sampling at mid-rd2, limiting the villages availible,
 as well as the percentage sampled to fit the availible resources.
To be run in a directory containing the following: sample1.1.do ${dodir}(see below),
villages.dta containing all the villages with codes, and socialgroup.dta,
containing the households of interest.
*/

// After the FA numbers, the Villages/LC names that FA is responible for is written
clear
set memory 20m
use ${datadir}villages
keep villagenam villagecod
sort villagecod
save ${tempdir}temp.dta, replace
use membership
drop if edate~=.
bys socialgpid: keep if _n==1
sort individid
save ${tempdir}temp2.dta,replace
clear
use ${datadir}residency
drop if edate~=.
sort individid
merge individid using ${tempdir}temp2.dta
drop if _merge~=3
keep socialgpid locationid
gen villagecod=upper(substr(locationid,1,6))
sort villagecod
merge villagecod using ${tempdir}temp.dta
drop if _merge~=3
save ${tempdir}temp2.dta, replace

//FA 1: Busowobi C + Nenga

clear
use ${tempdir}temp2.dta //Load the prepared data set
keep if villagecod=="I10503" | villagecod =="I31503" // Change this if the areas are altered
// bys villagecod (socialgpid): drop if _n<=floor(_N*220/320) & villagecod=="I10503" //
do ${dodir}sample1.1.do
// Sample the households using the sample1.1.do routine included at the end of this file
```

```
keep if sampled==1 // drop nonsampled households
keep socialgpid villagecod
save ${outdir}Reint_FA1, replace // Change number acc. to which FA we're dealing with.


//FA 2: Bunyama+Bukwaya

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10103" | villagecod =="I10104"
// bys villagecod (socialgpid): drop if _n<=floor(_N*210/305) & villagecod=="I10103"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA2 , replace

//FA 3: Namirali + Kiboyo

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10102" | villagecod =="I10101"
bys villagecod (socialgpid): drop if _n<=floor(_N*135/300) & villagecod=="I10101"
drop if villagecod=="I10102"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA3, replace

//FA 4: Butende Mulanga +Bukoteka

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10301" | villagecod =="I10302"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/245) &  villagecod=="I10301"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA4 , replace

//FA 5: Izimba +Wairama

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10201" | villagecod =="I10202"
drop if villagecod=="I10201"
bys villagecod (socialgpid): drop if _n<=floor(_N*86/226) & villagecod=="I10202"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA5 , replace
```

```
//FA 6: Matovu+Busaakala+Bubonde

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10402" | villagecod =="I10403" | villagecod=="I10405"
drop if villagecod=="I10402"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA6 , replace

//FA 7: Kabira+Nawanzu

clear
use ${tempdir}temp2.dta
keep if villagecod=="I21102" | villagecod =="I21103"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/322) & villagecod=="I21102"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA7 , replace

//FA 8: Bulubandi Nandekula

clear
use ${tempdir}temp2.dta
keep if villagecod=="I31402"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/401)
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA8 , replace

//FA 9: Buseyi Central+ Nakavule A

clear
use ${tempdir}temp2.dta
keep if villagecod=="I31202" | villagecod =="I41702"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/216) & villagecod=="I31202"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA9 , replace

//FA 10: Bulubandi Central

clear
use ${tempdir}temp2.dta
keep if villagecod=="I31401"
```

```
bys villagecod (socialgpid): drop if _n<=floor(_N*210/692)
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA10 , replace

//FA 11: Bulubandi Bugabwe+ Buluza

clear
use ${tempdir}temp2.dta
keep if villagecod=="I31403" | villagecod =="I21101"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/312) & villagecod=="I31403"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA11 , replace

//FA 12: Buligo North + Buligo South

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41604 " | villagecod =="I41605"

bys villagecod (socialgpid): drop if _n<=floor(_N*210/259) & villagecod=="I41604"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA12 , replace

//FA 13: Kasokoso South

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41602"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/295)
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA13 , replace

//FA 14: Kasokoso North

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41601"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/302)
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA14 , replace
```

```
//FA 15: Nabidongha C

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41607"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/355)
gen indicator=1
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA15 , replace

//FA 16: Nabidongha Prisons + Nabidondgha B

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41606" | villagecod =="I41704"
drop if villagecod=="I41606"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA16 , replace

//FA 17: Kayaga + Walugogo Estate

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41603" | villagecod =="I41801"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/222) & villagecod=="I41603"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA17 , replace

//FA 18: Busei South + Busei B

clear
use ${tempdir}temp2.dta
keep if villagecod=="I31203" | villagecod =="I31201"
drop if villagecod=="I31203"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA18 , replace

//FA 19: Nakalama + Bukoboli + Nabitovu A&B

clear
use ${tempdir}temp2.dta
```

```
keep if villagecod=="I31301" | villagecod =="I31302" | villagecod =="I31504" | villagecod=="I10504"
drop if villagecod=="I31301" | villagecod =="I31302" | villagecod =="I31504"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA19 , replace

//FA 20: Nakavule B + Lubale

clear
use ${tempdir}temp2.dta
keep if villagecod=="I41509" | villagecod =="I41701"
drop if villagecod=="I41509"
bys villagecod (socialgpid): drop if _n<=floor(_N*70/341) & villagecod=="I41701"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA20 , replace

//FA 21: Magada + Mbaale TC + Isikiro TC

clear
use ${tempdir}temp2.dta
keep if villagecod=="M20701" | villagecod =="M20901" | villagecod=="M20801"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/340) & villagecod=="M20701"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA21 , replace

//FA 22: Nawansinge + Bukoyo + Buwooya

clear
use ${tempdir}temp2.dta
keep if villagecod=="I21002" | villagecod =="I21001" | villagecod=="I21004"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/249) & villagecod=="I21002"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA22 , replace

//FA 23: Budhwege + Luyira + Namakakale

clear
use ${tempdir}temp2.dta
keep if villagecod=="I21003" | villagecod =="M20702" | villagecod=="M20706"
drop if villagecod=="I21003"
bys villagecod (socialgpid): drop if _n<=floor(_N*13/188) & villagecod=="M20702"
do ${dodir}sample1.1.do
keep if sampled==1
```

```
keep socialgpid villagenam
save ${outdir}Reint_FA23, replace

//FA 24: Bulyampindi + Namadudu + Wante

clear
use ${tempdir}temp2.dta
keep if villagecod=="M20705" | villagecod =="M20703" | villagecod=="M20704"
drop if villagecod=="M20705"
bys villagecod (socialgpid): drop if _n<=floor(_N*36/144) & villagecod=="M20703"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA24 , replace

//FA 25: Walanga + Nawampendo + Kabayingire + Nakate

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10401" | villagecod =="I10404" | villagecod=="M20606" |villagecod=="M20601
drop if villagecod=="I10401" | villagecod=="I10404"
bys villagecod (socialgpid): drop if _n<=floor(_N*35/153) & villagecod=="M20606"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA25, replace

//FA 26: Namadhi + Buwaiswa + Bubago + Buwaaya + Ntafungirwa

clear
use ${tempdir}temp2.dta
keep if villagecod=="M20603" | villagecod =="M20605" | villagecod=="M20602" |
villagecod=="M20608" |villagecod=="M20604"
drop if villagecod=="M20603" | villagecod=="M20605"
bys villagecod (socialgpid): drop if _n<=floor(_N*83/153) & villagecod=="M20602"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA26, replace


//FA 27: Kakombo + Nakisenyi

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10203" | villagecod =="I10204"
drop if villagecod=="I10203"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
```

```
save ${outdir}Reint_FA27 , replace

//FA 28: Nakigo 11B + Nakigo 11A

clear
use ${tempdir}temp2.dta
keep if villagecod=="I10501" | villagecod =="I31501"
bys villagecod (socialgpid): drop if _n<=floor(_N*210/241) & villagecod=="I10501"
do ${dodir}sample1.1.do
keep if sampled==1
keep socialgpid villagenam
save ${outdir}Reint_FA28 , replace
/*

/*  This is the sample1.1.do routine called above. Written by Gustaf Rydevik 2nd October 2006

gen rand=uniform()
sort rand  // gives the data a random order
gen sampled=1 if _n <= floor(_N*0.0095)
 //assigns the first percent (rounded down) a "sampled" indicator

replace sampled=0 if sampled~=1
//assigns all other observations a "zero" sampled flag.

*/
--------------------------

// px_calculator.do

// Due to Bruce Mcleod, slightly edited by Gustaf Rydevik //

do AnalysisSettings.do
/* Determine the last location that an individual was resident at and the
last time that location was visited. */

clear
use ${datadir}individual, clear
sort individid
save ${tempdir}individual_sort, replace

* Calculate the last date that a location was visited

use ${datadir}observation
egen lastdate=max(date), by(locationid)
format lastdate %d
drop if !(date==lastdate)
keep locationid date
sort locationid
save ${tempdir}lastvisit,replace
```

```
clear

* Determine the last location that an individual was resident

use ${datadir}residency
egen lastsdate=max(sdate), by(individid)
format lastsdate %d
drop if !(sdate==lastsdate)
keep individid locationid edate eeventtype
sort individid
save ${tempdir}lastlocation,replace

* Combine last location an individual was resident and the last visit date of the location

sort locationid
merge locationid using ${tempdir}lastvisit, nokeep keep(date)
rename date lastvisit
drop _merge
sort individid
save ${tempdir}lastobservation, replace
* Calculate Person days of Observation
use ${datadir}residency

drop episodeid type seventtype sobserveid eeventtype eobserveid de_date data_clerk status_dat

* Bring the gender and birth date into the residency record

sort individid
merge individid using ${tempdir}individual_sort, nokeep keep(gender birth_date)
drop _merge

* Adding the lastvisit for each individual

sort locationid
merge locationid using ${tempdir}lastvisit, nokeep keep (date)
rename date lastvisit
format lastvisit %d
drop _merge

* When you load a blank FoxPro date into the system, it initialize the value to 12/30/1899

replace edate = mdy(1,1,3000) if (edate == mdy(12,30,1899))
replace edate=lastvisit if edate == mdy(1,1,3000)
replace edate=lastvisit if edate==.

* If the residency start date is before the period of analysis, then we up the sdate
* similar process for edate

replace sdate=$startTime if $startTime>sdate
replace edate=$endTime if $endTime<edate
```

```
drop if edate<sdate

* Generate the age at the start and the end of the interval

generate StartAge = (sdate - birth_date)
generate EndAge = (edate - birth_date)


* years represents the # of age groups the individual passes thru during the residency

generate years = int(EndAge/365.25) - int(StartAge/365.25) + 1

* both conditions should not happen (but do)

drop if years==.
drop if years<=0

* for every age group the individual goes through during the residency, create new records

expand years

* create an age variable to allocate each record to one of the age groups

sort individid sdate
quietly by individid sdate : ge seqvar = _n
generate age = StartAge if seqvar==1
replace age= int(StartAge/365.25)*365.25+365.25*seqvar -365.25 if seqvar~=1

* this should not be necessary -- do it anyway

drop if age < 0
drop if age > 50000


* this calculates the begin and end of the interval in which the individual is the age
*   designated by the record

/*  Next three lines are a bit incorrect -- I count the birthday as belonging to the older age --
it is simply easier to program.
To fix, we need to find the last day of a month and define  for  case
when the person is born on the first of the month (must be some stata code to do this) */

generate beginInterval=birth_date + int(age/365.25)*365.25
generate endInterval=birth_date+ int(age/365.25)*365.25 + 365

* check this against the residency start date (sdate) and end date

replace beginInterval=sdate if beginInterval < sdate
replace endInterval=edate if endInterval > edate
```

```
* Calculate the number of days that a person is the age of the record

generate days = endInterval-beginInterval

format beginInterval %d
format endInterval %d

drop if days < 0

* Could have part of the age in one residency record and the other part in another residency

rename days px
replace px= px/365.25
* This saves Person years for each individual

egen agegroup= cut(age), at(0(1825)45625) // Group people into ages in 5yr stratas
replace agegroup=31025 if age >= 31025 // Group all those above 85 yrs together
replace agegroup=365 if age>=365 & age<1825
drop if agegroup==.
lab def A_group 0 "Under 1 year"
lab def A_group 365 "1-4 yrs", add
lab def A_group 1825 "5-9 yrs", add
lab def A_group 3650 "10-14 yrs", add
lab def A_group 5475 "15-19 yrs", add
lab def A_group 7300 "20-24 yrs", add
lab def A_group 9125 "25-29 yrs", add
lab def A_group 10950 "30-34 yrs", add
lab def A_group 12775 "35-39 yrs", add
lab def A_group 14600 "40-44 yrs", add
lab def A_group 16425 "45-49 yrs", add
lab def A_group 18250 "50-54 yrs", add
lab def A_group 20075 "55-59 yrs", add
lab def A_group 21900 "60-64 yrs", add
lab def A_group 23725 "65-69 yrs", add
lab def A_group 25550 "70-74 yrs", add
lab def A_group 27375 "75-79 yrs", add
lab def A_group 29200 "80-84 yrs", add
lab def A_group 31025 "Above 85 years", add
label values agegroup A_group

save ${tempdir}pyoIndividual, replace
-------------------------

// life_table_5yr1.do

/* A life table generator for stata, rev2. By Gustaf Rydevik 18th Oct 2006.
Based on "A Method for Constructing
Complete Annual U.S. Life Tables", by Robert N. Anderson, Ph.D.
, Division of Vital Statistics, Vital Health Stat2 2000;(129):1-28,
```

```
and "NATIONAL BURDEN OF DISEASE STUDIES:
A PRACTICAL GUIDE Edition 2.0", WHO Global Program on Evidence for Health Policy, 2001.
All faults and miscalculations are mine.

Person years calculated using a file written by Bruce Mcleod.

The format for data files is based on the file names generated by
 the HRS2 database system for DSS sites.
Required files in the folder are:

-observation.dta, containing the dates for visits to locations,
location id's and a round indicator.
-death.dta, containing the id's of people that have died, together with deathdates.
-individual.dta, containing id's of people in the Demographic Survaillance Area.
-residency.dta, containing the residence for individuals.
*/

do AnalysisSettings.do
log using ${logdir}Life_table.log,replace
clear
use ${datadir}death
drop if dthdate < $startTime | dthdate > $endTime
drop if round< $startRound | round> $endRound
sort individid
save ${tempdir}temp.dta, replace


local test=$all_analysis+1
if 'test'==2 {
use ${tempdir}pyoIndividual.dta
}
else {
do ${dodir}px_calculator.do
}
sort individid
merge individid using ${tempdir}temp.dta,keep(dthdate) nokeep
bys individid (age): gen Dx=1 if dthdate~=. & _n==_N
save ${tempdir}temp2.dta,replace
//Calculations are now repeated once each for males and females.

// Male life table

drop if gender =="F"
collapse (sum)Dx (sum)px,by(agegroup)  // And aggregate the variables
sort agegroup
gen qx=5*Dx/(px +5*Dx/2)
/* Estimate the prob of dying in agegroup by the proportion of deaths to personyears,
adjusted for occ. deaths */

replace qx =1 if agegroup==31025
```

```
//Prob of dying at age 85 or above, when in the group is one.

replace qx=Dx/(px+.7*Dx) if agegroup==0
replace qx=4*Dx/(px+4*.6*Dx) if agegroup==365
gen lx = 100000
local number=_N
 quietly forv i = 2/‘number’ {
replace lx=lx[_n-1]*(1-qx[_n-1]) if _n==‘i’
}
// The amount of persons left after x years,by current rates, starting with 100 000

gen dx= lx*qx
gen Lx=5*(lx[_n]-dx/2) if agegroup ~=31025 //The stationary population per age interval
replace Lx=(lx[_n]-.7*dx) if agegroup==0
replace Lx=4*(lx[_n]-4*.6*dx) if agegroup==365


// The next line is a rough estimate of the stationary population at age 85 or above

replace Lx=lx/(Dx/px) if agegroup ==31025

egen temp=sum(Lx)
gen Tx=temp- sum(Lx[_n-1]) if _n~=1
//Total man years left to live by the starting 100000 people population at year _n.

replace Tx=temp if _n==1
drop temp
gen ex=Tx/lx
// Expected life length is years left divided by persons left in starter pop.

//Male Life Table
l
save ${outdir}male_life_table.dta, replace


//Female Calculations
clear
use ${tempdir}temp2.dta
drop if gender=="M"
collapse (sum)Dx (sum)px,by(agegroup)  // And aggregate the variables
sort agegroup
gen qx=5*Dx/(px +5*Dx/2)
/* Estimate the prob of dying in agegroup by the proportion of deaths to personyears,
adjusted for occ. deaths */

replace qx =1 if agegroup==31025
//Prob of dying at age 85 or above, when in the group is one.

replace qx=Dx/(px+.7*Dx) if agegroup==0
replace qx=4*Dx/(px+4*.6*Dx) if agegroup==365
```

```
gen lx = 100000
 quietly forv i = 2/‘number’ {
replace lx=lx[_n-1]*(1-qx[_n-1]) if _n==‘i’
 // The amount of persons left after x years,by current rates, starting with 100 000
}
gen dx= lx*qx
gen Lx=5*(lx[_n]-dx/2) if agegroup ~=31025 //The stationary population per age interval

// The next line is a rough estimate of the stationary population at age 85 or above

replace Lx=lx/(Dx/px) if agegroup ==31025
replace Lx=(lx[_n]-.7*dx) if agegroup==0
replace Lx=4*(lx[_n]-4*.6*dx) if agegroup==365

egen temp=sum(Lx)
gen Tx=temp- sum(Lx[_n-1]) if _n~=1
//Total man years left to live by the starting 100000 people population at year _n.

replace Tx=temp if _n==1
drop temp
gen ex=Tx/lx
// Expected life length is years left divided by persons left in starter pop.

//Female Life Table

l
save ${outdir}female_life_table.dta, replace

*/
log close
---------------------------

// pregnancyreport.do

/* This file generates an analysis of women whose pregnancies
 have been registered before giving births, vs women whose pregnancies were missed.
Co-written by Dorean Nabukalu and Gustaf Rydevik, November 2006 */

//Making sure that all global options are set.
do AnalysisSettings.do

log using ${logdir}Pregnancy.log, replace
clear
set memory 50m
set more off

// Prepare all the relevant datasets for merging: sorting and removing duplicates.

use ${datadir}observation
keep date round locationid  respid field_wrkr observeid
```

```
rename field_wrkr resp_fw
rename date obsdate
rename round obsround
sort locationid
save ${tempdir}temp,replace
clear

use ${datadir}residency
keep individid locationid sdate edate
rename sdate res_start
rename edate res_end
sort locationid
joinby locationid using ${tempdir}temp
sort individid
save ${tempdir}temp2,replace


use ${datadir}membership
keep individid rltn_head socialgpid sdate
sort individid
save ${tempdir}temp3,replace
clear

use ${datadir}indvstatus
keep individid pregdate roundnum field_wrkr
sort individid
save ${tempdir}temp4,replace
clear

use ${datadir}individual
sort individid
save ${tempdir}temp5, replace
clear

use ${datadir}education
bys individ (de_date): drop if _n~=_N
sort individid
save ${tempdir}temp6, replace
clear

use ${datadir}fieldassistants
rename facode field_wrkr
rename gender fwgender
sort field_wrkr
save ${tempdir}temp7, replace
clear

// The following generates a variable of neonatal  deaths,
// tied to the observation ID of the birth.
```

```
use ${datadir}death
sort individid
save ${tempdir}temp8,replace
use ${datadir}birth
rename observeid birthobs
rename field_wrkr birth_fw
gen childid=individid
rename birthdate child_bd
drop eventid episodeid locationid socialgpid type data_clerk
sort individid
merge individid using ${tempdir}temp8, keep(dthdate observeid )
rename observeid childDeathobsID
rename birthobs observeid
rename dthdate child_dthdate
drop individid rltn_head othrltn status_dat
gen dthage= child_dthdate-child_bd
drop _merge
drop if dthage>28
gen NeoNatalDeath=1
sort observeid
save ${tempdir}temp8,replace


// Prepares the main data set of interest,
// limiting the analysis to events inside the analysis time span.
use ${datadir}pregoutcome.dta
drop if date > $endTime | date <$startTime
drop if roundnum<$startRound | roundnum>$endRound
keep individid date type totalborn de_date observeid round
rename round birth_round
rename date date_birth
rename type birth_type
rename observeid OtcObsID
sort individid

// Adds the observation ID for where the women was in the round before birth,
//  when the pregnancy could have been registered.

joinby individid using ${tempdir}temp2,unmatched(master)
count
bys OtcObsID individid: keep if obsround==birth_round-1
// Only keep observations of the round previous to the birth registration.

bys OtcObsID individid: drop if res_start>date_birth
// Drop all residencies where the mother started residing after giving birth.

egen Birth_res_time=min(date_birth-res_start), by(OtcObsID individid)
// Find out how long the mother had lived in the residency she belonged to when giving birth.

gen Birth_res=1 if date_birth-res_start==Birth_res_time // Mark this household.
```

```
drop if date_birth-obsdate>Birth_res_time & Birth_res==1
// If the observation occured before the women entered, she could not have been registered.

drop Birth_res_time Birth_res
egen Birth_res_time=min(date_birth-res_start), by(OtcObsID individid)
// In the case, find out where she lived before coming to the last household.

gen Birth_res=1 if date_birth-res_start==Birth_res_time
keep if Birth_res==1 // Keep the observations we want.

bys OtcObsID individid: drop if _n~=1
//Should not be necessary unless strange things happen.

count
sort individid
joinby individid using ${tempdir}temp3
rename sdate membership_sdate
drop if membership_sdate>obsdate
egen birth_gp_time=min(obsdate-membership_sdate),by(OtcObsID individid)
drop if obsdate-membership_sdate~=birth_gp_time
drop birth_gp_time


// Adds information of the respondent at the possible pregnancy registration interview.
sort individid
count
count if individid==individid[_n+1]
rename socialgpid mother_group
rename individid mother_id
rename rltn_head mother_rltn
rename respid individid
sort individid
drop _merge
joinby individid using ${tempdir}temp3,unmatched(master)
rename sdate resp_gp_sdate
egen resp_gp_time=min(obsdate-resp_gp_sdate),by(OtcObsID individid)
drop if obsdate-resp_gp_sdate~=resp_gp_time
count
sort individid
drop _merge
merge individid using ${tempdir}temp5, keep(gender birth_date ethnicgrp field_wrkr) nokeep
drop _merge
rename individid respid
rename gender respgender
rename birth_date resp_bday
rename ethnicgrp respethnic


// Adds variables about the mother, including one that indicates
// wether the pregnacy was reported or not.
```

```
gen individid=mother_id
rename rltn_head resp_rltn
rename socialgpid resp_group
sort individid
merge individid using ${tempdir}temp5, keep(birth_date ethnicgrp) nokeep
drop _merge
count
rename birth_date mother_bd
rename ethnicgrp mother_ethnic
sort individid
joinby individid using ${tempdir}temp4,unmatched(master)
count

gen pregnancy_reported=1 if _merge==3 & roundnum==birth_round-1
replace pregnancy_reported=0 if pregnancy_reported==.

bys OtcObsID individid: drop if _n~=1 //Should not be necessary...

// Adds an indicator, marking if the pregnancy mature enough at the time of the previous visit,
//  that it had a chance of being captured.
gen pregNoticable=1 if obsdate>date_birth-256
replace pregNoticable=0 if pregNoticable==.


// Now, it's just defining all the covariates of interest...


//Age of respondent
gen respage=int((date-resp_bday)/365.25)
gen mother_age=int((date-mother_bd)/365.25)
gen resp_moth_rel=4 if resp_rltn=="HHH"
replace resp_moth_rel=1 if respid==individid
replace resp_moth_rel=2 if mother_group~=resp_group
replace resp_moth_rel=3 if resp_moth_rel==.


//Education of mother.
drop _merge
sort individid
merge individid using ${tempdir}temp6,keep(hclass respid field_wrkr)
count
drop if _merge==2
count
drop individid
gen class_gp=1 if hclass<=0 & hclass!=.
 replace class_gp=2 if hclass >=1 &  hclass<=4
 replace class_gp=3 if hclass >=5 &  hclass<=7
 replace class_gp=4 if hclass >=8 &  hclass<=11
 replace class_gp=5 if hclass >=12 &  hclass<=13
```

```
// The relation between fieldworker and respondent gender.
drop _merge
sort field_wrkr mother_id
count
merge field_wrkr using ${tempdir}temp7, keep(fwgender field_wrkr)
count
tab _merge
drop if _merge!=3

gen resp_fwsex=1 if respgender=="M" & fwgender=="M"
replace resp_fwsex=2 if respgender=="F" & fwgender=="F"
replace resp_fwsex=3 if respgender=="M" & fwgender=="F"
replace resp_fwsex=4 if respgender=="F" & fwgender=="M"
replace resp_fwsex=5 if respgender=="" & fwgender!=""

encode fwgender, gen(Fwgender)
encode respgender, gen(Respgender)
drop respgender fwgender
rename Fwgender fwgender
rename Respgender respgender

// Neonatal death of the child.
drop _merge
rename observeid pregrepObsID
gen observeid=OtcObsID
sort observeid
merge observeid using ${tempdir}temp8
drop if _merge==2
replace NeoNatalDeath=0 if child_bd~=date_birth


// Type of pregnancy outcome
gen Type=1 if birth_type=="STB"
replace Type=2 if birth_type=="LBR"
replace Type=3 if birth_type=="MIS"
drop birth_type
rename Type type


// Age of mother
gen over18 = 1 if mother_age>=18
replace over18=0 if mother_age<18

label var over18 "Was the mother over 18?"
label define yesno_label 1 "yes" 0 "No
label value over18 yesno_label

label var type "Pregnancy outcome"
label define pregotc_label 1 "Stillbirth" 2 "Live birth" 3 "Misscarriage"
```

```
label value type pregotc_label

label value NeoNatalDeath yesno_label
label var NeoNatalDeath "Did the child die before 28 days of age?"

label var class_gp " High class grouped in 5 different levels for all individuals in DSS "
label define high_class_groups 1 "Never" 2 "Lower Primary" 3 "Upper Primary" 4 "O'level" 5 "Higher"
label value class_gp high_class_groups
rename class_gp resp_educ_level


label define Respondent_FW_Sex 1 "Both Male" 2 "Both Female"
3 "FW Female & Respondent Male" 4 "FW Male & Respondent Female" 5 "Not reported"
label value resp_fwsex Respondent_FW_Sex

label var resp_moth_rel "What was the position of the respondent in the household?
label define Res_moth_label 4 "Head of Household" 1 "The pregnant woman"
2 "Not a member" 3 "Other relationship"
label value resp_moth_rel Res_moth_label

label var pregnancy_reported " Was the pregnancy reported during visit? "
label value pregnancy_reported yesno_label
label var respgender "Gender of the respondent"

label var fwgender "Fieldworker gender"

label var type "Outcome of pregnancy"


// Using tab and mlogit,significance fighures and relative risk ratios are calcualated.

tab pregnancy if pregNoticable==1

ta resp_fwsex pregnancy if pregNoticable==1, chi2 row
mlogit resp_fwsex pregnancy if pregNoticable==1, rrr

ta fwgender pregnancy if pregNoticable==1, chi2 row
mlogit fwgender pregnancy if pregNoticable==1, rrr

ta resp_educ_level pregnancy if pregNoticable==1, chi2 row
mlogit resp_educ_level pregnancy if pregNoticable==1, rrr

ta resp_moth_rel pregnancy if pregNoticable==1, chi2 row
mlogit resp_moth_rel pregnancy if pregNoticable==1 ,rrr

ta over18 pregnancy if pregNoticable==1,chi2 row
mlogit over18 pregnancy if pregNoticable==1, rrr

ta type pregnancy if pregNoticable==1, chi2 row
mlogit type pregnancy if pregNoticable==1, rrr
```

```
ta NeoNatalDeath pregnancy if pregNoticable==1,chi2 row
mlogit NeoNatalDeath pregnancy if pregNoticable==1, rrr

ta respgender pregnancy if pregNoticable==1,chi2 row
mlogit respgender pregnancy if pregNoticable==1, rrr


save ${outdir}pregnancydata.dta,replace
log close
-----------------------

R-code for functions related to the permutation test


# Calculates the rate of an event in a population, given individual level observation.
# Split up by FW-id's
fw.event.rates<-function(fwEvent,fwPop,Event,Pop ) {
x<-c(by(fwEvent,fwEvent,length))
y<-c(by(fwPop,fwPop,length))
Fw.data<-merge(data.frame(n.event=x,fw=names(x)),data.frame(n.pop=y,fw=names(y)),all=TRUE)
Fw.data$rates<-Fw.data$n.event/Fw.data$n.pop
return(Fw.data)
}


#Gives the lowest rate for an event among all   Field workers.

fw.minrate<-function(fwEvent, fwPop, Event, Pop){
x<-fw.event.rates(fwEvent, fwPop, Event, Pop)
minrate<-min(x$rate,na.rm=T)
return(minrate)
}

#Do a bootstrap sampling of a data frame.

min.sample<-
function(d,mle=NULL) {
x<- sample(1:nrow(d), nrow(d),replace=T)
return(d[x,])
}


# Permutes the FW id's for observations and events.

syntevent<-function(data,fw.col,fwevent.col){
y<-data
event.indices<-which(!is.na(data[,fwevent.col]))
y[,fw.col]<-sample(y[,fw.col])
y[event.indices,fwevent.col]<-y[event.indices,fw.col]
```

```
return(y)
}

#A general bootstrap function. Does R bootstrap samples for each of R2 permutations of the data,
#  and returns a vector of values for a statistic for each iteration.

min.bootstrap<-function(fwEvent,fwPop,Event,Pop, statistic,R, R2) {
if (R2!=0) {
                if (R!=0) {
                        temp2<-c()
                        for (i in 1:R2) {
                                syntData<-syntevent(data.frame(fwEvent,fwPop,Event,Pop),2,1)
                                temp<-c()
                                for (i in 1:R){
                                        x<-min.sample(syntData)
                                        temp<-c(temp,statistic(x[,1],x[,2],x[,3],x[,4]))
                                        }
                                temp2<-c(temp2,temp)
                        }
                        temp2<-temp[order(temp2)]
                        return(temp2)
                } else{
                        temp<-c()
                        for (i in 1:R2){
                                x<-syntevent(data.frame(fwEvent,fwPop,Event,Pop),2,1)
                                temp<-c(temp,statistic(x[,1],x[,2],x[,3],x[,4]))
                                }
                        temp<-temp[order(temp)]
                        return(temp)
                        }
        } else {
                temp<-c()
                for (i in 1:R){
                        x<-min.sample(data.frame(fwEvent,fwPop,Event,Pop))
                        temp<-c(temp,statistic(x[,1],x[,2],x[,3],x[,4]))
                        }
                temp<-temp[order(temp)]
                return(temp)
                }
}

-------
```