Part C. Operations Research

Part C: Operations research

Introductory note

In this Part C three operationally relevant research questions will be answered:

- Does the dataset tell us something about how diligent the work was performed in the tuberculosis microscopy laboratory?

- Is the third serial smear examination associated with an excessive amount of work for little gain?

- Is it necessary to confirm a positive smear result?

These and related questions were asked by graduates from The Union's operations research courses in fulfillment of the field component of the course. The data were collected in Moldova (Dr Dumitru Laticevschi, fifth course, Paris, 2003), Mongolia (Dr Nymadawaa Naranabat, seventh course, Paris, 2004), Uganda (Dr Achilles Katamba, fifth course, Paris, 2003), and Zimbabwe (Dr Biggie Mabaera, seventh course, Paris, 2004). Six publications have resulted from this study:

Mabaera B, Naranbat N, Dhliwayo P, Rieder H L. Efficiency of serial smear examinations in excluding sputum smear-positive tuberculosis. Int J Tuberc Lung Dis 2006;10:1030-5.

Katamba A, Laticevschi D, Rieder H L. Efficiency of a third serial sputum smear examination in the diagnosis of tuberculosis in Moldova and Uganda. Int J Tuberc Lung Dis 2007;11:659-64.

Mabaera B, Lauritsen J M, Katamba A, Laticevschi D, Naranbat N, Rieder H L. Sputum smear-positive tuberculosis: empiric evidence challenges the need for confirmatory smears. Int J Tuberc Lung Dis 2007;11:959-64.

Mabaera B, Lauritsen J M, Katamba A, Laticevschi D, Naranbat N, Rieder H L. Making pragmatic sense of data in the tuberculosis laboratory register. Int J Tuberc Lung Dis 2008;12:294-300.

Mabaera B, Naranbat N, Katamba A, Laticevschi D, Lauritsen J M, Rieder H L. Seasonal variation among tuberculosis suspects in four countries. International Health 2009;1:53-60.

Rieder H L, Lauritsen J M, Naranbat N, Katamba A, Laticevschi D, Mabaera B. Quantitative differences in sputum smear microscopy results for acid-fast bacilli by age and sex in four countries. Int J Tuberc Lung Dis 2009;13:1393-8.

With permission of the investigators, the datasets have been made publicly accessible for use in this course exactly as they have been collected.

At the end of this exercise you should be able to:

a. Combine different datasets into one combined dataset

b. Recode 'text variables' to 'numeric variables'

c. Remove 'undesirable' records from a dataset

d. Correct obvious gross errors from the datasets

e. Create a 'cleaned' final working dataset from available datasets

Moldova and Uganda worked together using the same data entry forms. You obtained `MOL_25.ZIP` and `UGA_30.ZIP`. These two files contain respectively the data files obtained from the 25 laboratories in Moldova and the data files obtained from the 30 laboratories in Uganda. In addition, each of the zip files contains the base pair of `QES` and `CHK` files (which are identical for both countries, except for the field name for the laboratory).

Mongolia and Zimbabwe worked together using the same data entry forms. You obtained `MON_31.ZIP` and `ZIM_23.ZIP`. These two files contain respectively the data files obtained from the 31 laboratories in Mongolia and the data files obtained from the 23 laboratories in Zimbabwe. In addition, each of the zip files contains the base pair of `QES` and `CHK` files (which are identical for both countries).

The two pairs of countries collected exactly the same information from the laboratory register, but their data collection forms (the `QES` files, and thus `REC` files) and `CHK` files had small differences. You can find these by inspecting the respective files. However, as you come in here as an outsider, we summarize these in the following table, and also give you the field names that the final data set combining all files should have.

Field label	Field name Moldova / Uganda	Field name Mongolia / Zimbabwe	Final Field name
Study country	--	--	country
Laboratory code	labcode / labno	laboratory	laboratory
Laboratory serial number	serno	serno	--
Registration date	labdate	regdate	regdate
Year of registration	--	--	regyear
Created unique identifier	unique	id	--
Sex of examinee	sex	sex	sex
Age (in years) of examinee	age	age	age
Reason for examination	reason	reason	reason
Result of first examination	res1	res1	result1
Result of second examination	res2	res2	result2
Result of third examination	res3	res3	result3

Omissions and commissions

In contrast to what you learned in Part A, the data entry form used only field names but had no field labels.

In both studies `SEX` and `REASON` were coded as text fields rather than numerically (but label blocks were used). The fields `RES1`, `RES2`, and `RES3` also differed slightly: a value of 4.0 did not exist in Moldova / Uganda, but denoted "Positive, not quantified" in Mongolia / Zimbabwe, while "Positive, not quantified" was coded as 8.0 in the latter but did not exist in the former. "Scanty, not quantified" was coded as 5.0 in Mongolia / Zimbabwe, but was forgotten as a possible value in Moldova / Uganda.

You could obtain the information from the `CHK` files, but the summary of the coding for the fields of relevance with the differences is as follows:

Field name	Field value Moldova / Uganda	Field value Mongolia / Zimbabwe	Value label
Sex	F	F	Female
	M	M	Male
	9	9	Unknown sex
Reason	D	D	Diagnosis
	F	F	Follow-up, month not stated
	9	9	Reason not stated
	--	1	Follow-up at 1 month
	--	2	Follow-up at 2 months
	--	3	Follow-up at 3 months
	--	4	Follow-up at 4 months
	--	5	Follow-up at 5 months
	--	6	Follow-up at 6 months
	--	7	Follow-up at 7 months
	--	8	Follow-up at 8 months or later
res1 (also res2, res3)	0.0	0.0	Negative
	0.1	0.1	Scanty, 1 AFB / 100 fields
	0.2	0.2	Scanty, 2 AFB / 100 fields
	0.3	0.3	Scanty, 3 AFB / 100 fields
	0.4	0.4	Scanty, 4 AFB / 100 fields
	0.5	0.5	Scanty, 5 AFB / 100 fields
	0.6	0.6	Scanty, 6 AFB / 100 fields
	0.7	0.7	Scanty, 7 AFB / 100 fields
	0.8	0.8	Scanty, 8 AFB / 100 fields
	0.9	0.9	Scanty, 9 AFB / 100 fields
	1.0	1.0	1+ positive
	2.0	2.0	2+ positive
	3.0	3.0	3+ positive
	--	4.0	Positive, not quantified
	--	5.0	Scanty, not quantified
	8.0	--	Positive, not quantified
	9.0	9.0	No result recorded

Tasks:

o *Create a combined dataset C_EX01_COMBINE.REC from all 107 files with a program C_EX01_COMBINE.PGM.*

Notes to the first task:

From the dataset from Moldova, drop the data for the laboratory "BND" (containing data from only 1 week) and remove one empty record.

From the dataset from Mongolia, remove the empty records

In Zimbabwe, one record has no laboratory value, but it has an ID (this is most likely attributable to some manipulation with the mouse after ID creation). You can retain this record by giving the laboratory the correct code that we know from the ID.

If you have removed all empty records (plus the one laboratory from Moldova) and you make a frequency of `COUNTRY` you should get:

country	N	%
MOL	17865	13.7
MON	22588	17.3
UGA	55114	42.3
ZIM	34744	26.7
Total	130311	100.0

- o *Create a "cleaned" final working dataset C_EX01.REC with a program C_EX01_2_RESTRUCTURE.PGM which excludes non-sensically coded result sequences, and with all fields codes numerically (including COUNTRY and LABORATORY).*

Notes to the second task:

For the numeric coding of the `COUNTRY` follow the alphabet: 1 for Moldova, 2 for Mongolia, …, 4 for Zimbabwe.

For the numeric coding of the laboratories, make a frequency for each country, and then code numerically following the country notation:

Moldova laboratories:

```
if laboratory="ANR" then lab0=101
if laboratory="BLM" then lab0=102
if laboratory="BRL" then lab0=103
if laboratory="BSR" then lab0=104
if laboratory="CCE" then lab0=105
…etc
```

Mongolia laboratories:

```
if laboratory="AR_B" then lab0=201
if laboratory="BG_B" then lab0=202
if laboratory="BN_B" then lab0=203
…etc
```

Uganda laboratories:

```
if trim(laboratory)="1"  then lab0=301
if trim(laboratory)="2"  then lab0=302
if trim(laboratory)="3"  then lab0=303
…etc
```

Zimbabwe laboratories:

```
if laboratory="BY_A" then lab0=401
if laboratory="MC_A" then lab0=402
if laboratory="MC_B" then lab0=403
if laboratory="MC_C" then lab0=404
…etc
```

We also propose to correct some obvious gross errors (which are obvious from the sequence in recording what they should have been) in the registration date. In order to get a common ground, we point these out here and provide the program file commands for these (note that we made a date variable just for this manipulation here):

```
define regyear0 ####
regyear0=year(regdate)
define regyear ####
regyear=regyear0
* correct errors in year of recording
if regyear0=1990 and laboratory=301 then regyear=1999
if regyear0=1990 and laboratory=306 then regyear=1999
if regyear0=1990 and laboratory=319 then regyear=1999
if regyear0=1990 and laboratory=320 then regyear=2000
if regyear0=1990 and laboratory=410 then regyear=2002

if regyear0=2000 and laboratory=408 then regyear=2002
if regyear0=2000 and laboratory=416 then regyear=2002
if regyear0=2000 and laboratory=419 then regyear=2002

if regyear0=2004 and laboratory=211 then regyear=2003
if regyear0=2004 and laboratory=223 then regyear=2003
if regyear0=2004 and laboratory=401 then regyear=2002
if regyear0=2004 and laboratory=408 then regyear=2002
if regyear0=2004 and laboratory=412 then regyear=2003
if regyear0=2004 and laboratory=413 then regyear=2002

if regyear0=2005 and laboratory=207 then regyear=2003
if regyear0=2033 and laboratory=207 then regyear=2003
```

If you have cleaned the dataset and you make a table of COUNTRY by REGYEAR you should get:

Study country					
Year of registration	Moldova	Mongolia	Uganda	Zimbabwe	Total
1999	0	0	17308	0	17308
2000	0	0	18655	0	18655
2001	0	0	18087	1213	19300
2002	0	149	0	29307	29456
2003	17725	22406	0	3958	44089
Total	17725	22555	54050	34478	128808

Note the following on the CHK and QES files:

If you start with a REC file that is accompanied by its CHK file and then create new variables with Field values and Value labels using the LABELVALUE, EpiData Analysis takes the original CHK file and appends it with the new Field values and their Value labels when you create a new REC file. You can also define a Field label (command LABEL newvar "X").