Solution to Exercise 4: From a spreadsheet to an EpiData file

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

a. Master the import of spreadsheet content into an EpiData file
b. Recode the imported content to an analyzable file

Task:

- ***The B_EX04_TASK.XLSX is a real data set from a study on laboratory drug susceptibility test results from several countries / jurisdictions. Create a sheet satisfying EpiData format requirements. Copy the rectangular selection to the clipboard and read it into EpiData Analysis and save it to an EpiData *.REC file. Note that it might be tricky to deal properly with some variables that may prevent correct reading of the rectangular file!***

Solution:

The tricky part was perhaps to deal with the original sheet variable `IDCODE`. If the values were left unchanged, EpiData may misread it as a date variable with erroneous dates that did not exist and in consequence the import failed. We solved it by unambiguously making it to a text field. The value:

```
2014-01-006
```

was made to be:

```
x-2014-01-006
```

by the following code:

```
=IF(original!G2<>"",CONCATENATE("x-", original!G2),"xxx")
```

The preceding “`x-`” identified the value unambiguously to a string field for EpiData Analysis. The extraneous “`x-`” was then easily stripped in EpiData Analysis with:

```
gen s(11) idcode0=substr(idcode2,3,11)
```

Task:

- ***The data set has a patient identifier, IDCODE. This identifier is unique for the patient in a given jurisdiction, but it is not unique for the data set. Specimens might be taken at the start of treatment or at any month on treatment. One patient in the data set thus may have several results. Make a variable IDCODE2 by adding the month to IDCODE. This will make it unique for a given patient in a given month. Actually – not quite: there may be more than one specimen in a given month, but only one result should count.***

The IDCODE2 could also come from two different patients if they are from a different jurisdiction. Make a third identifier so that you get in total three: 1) taking jurisdiction, patient and month of treatment, 2) taking jurisdiction and patient and 3) taking patient into account.

Solution:

This is relatively simple to accomplish. We chose this approach:

```
cls
* freq country
define countryn #
if country="A" then countryn=1
if country="B" then countryn=2
if country="C" then countryn=3
if country="D" then countryn=4
if country="E" then countryn=5
if country="F" then countryn=6
if country="G" then countryn=7
if country="H" then countryn=8
drop country
rename countryn to country
label country "Name of country / jurisdiction"
labelvalue country /1="Jurisdiction A"
labelvalue country /2="Jurisdiction B"
labelvalue country /3="Jurisdiction C"
labelvalue country /4="Jurisdiction D"
labelvalue country /5="Jurisdiction E"
labelvalue country /6="Jurisdiction F"
labelvalue country /7="Jurisdiction G"
labelvalue country /8="Jurisdiction H"

cls
* Create derived identifiers
gen s(16) idcoden=country+substr(idcode,2,12)+"-"+mmtxt
drop idcode2 idcode
rename idcoden to idcode2
label idcode2 "Country-patient-month identifier"
cls
gen s(13) idcode=substr(idcode2,1,13)
label idcode "Country-patient identifier"
cls
gen s(11) idcode0=substr(idcode2,3,11)
label idcode0 "Patient identifier"
```

Task:

- ***o*** ***Sort the data set so that all specimens for a given patient-month are next to each other, then recode all variable pertaining to isoniazid in a hierarchical manner, so that it becomes easy to assign the value that will be retained to the first specimen for that patient-month. Hierarchy: high-level resistance>low-level resistance>susceptible>no result. Then select to retain only the first specimen per patient-month.***

Solution:

First, we determined the frequency of the number of specimens per patient-month, using a standard approach which you will be using very often as this question arises very frequently:

```
cls
* Identify multiple specimens per patient-month
sort idcode2 mm
gen i specseq=1
if idcode2=idcode2[_n-1] then specseq=specseq[_n-1]+1
* freq specseq
* =>
*       N
* 1     582
* 2     26
* 3     5
* 4     4
* 5     2
* 6     2
* Total 621
* => Up to 6 specimens for 1 patient in 1 month
```

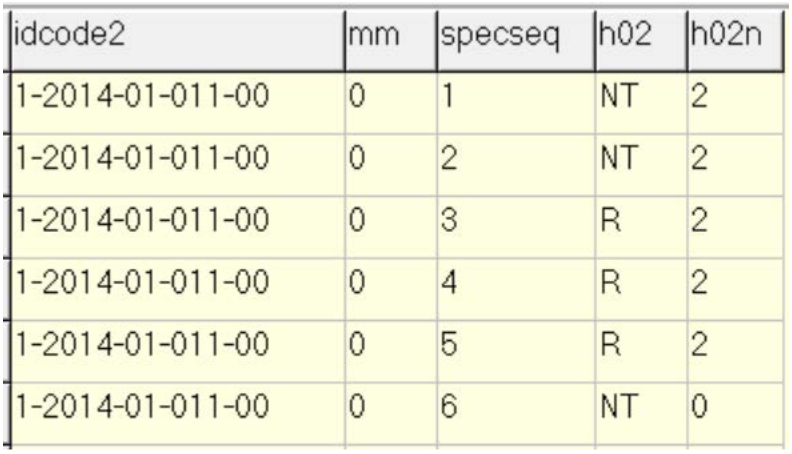
It is not necessary to know in this specific situation that at most 6 specimens are available per patient months, but it will be of key importance to know it for the next step. We show this here for isoniazid phenotypic result at 0.2 mg/L, i.e. the variable `H02`. Currently, these are the original (spreadsheet-coded) results:

```
NT    490
R     126
S     5
Total 621
```

As we recommended a hierarchical numerical coding, we create a new numeric variable `H02N` and then "reshuffle the values" hierarchically to the first record within IDCODE2:

```
cls
define h02n #
h02n=0
if h02="S" then h02n=1
if h02="R" then h02n=2
if idcode2[_n+1]=idcode2 and h02n[_n+1]>h02n then h02n=h02n[_n+1]
if idcode2[_n+2]=idcode2 and h02n[_n+2]>h02n then h02n=h02n[_n+2]
if idcode2[_n+3]=idcode2 and h02n[_n+3]>h02n then h02n=h02n[_n+3]
if idcode2[_n+4]=idcode2 and h02n[_n+4]>h02n then h02n=h02n[_n+4]
if idcode2[_n+5]=idcode2 and h02n[_n+5]>h02n then h02n=h02n[_n+5]
drop h02
rename h02n to h02
label h02 "INH result at 0.2 mg/L"
```

Before we drop h02 we browse to see all relevant variables, picking the case with 6 specimens in a given patient-month:



idcode2	mm	specseq	h02	h02n
1-2014-01-011-00	0	1	NT	2
1-2014-01-011-00	0	2	NT	2
1-2014-01-011-00	0	3	R	2
1-2014-01-011-00	0	4	R	2
1-2014-01-011-00	0	5	R	2
1-2014-01-011-00	0	6	NT	0

Note that the importance is that the first line of data (that is the record with `SPECSEQ=1`) must be correct in that it takes the hierarchically highest among all results from the total of specimens for the given patient-month. The highest value for `H02` is `R` which translates numerically to `2` and this is correctly appearing in the first record of the sequence. It is thereby irrelevant that the last value is `0` (it is logical that it retains the default because there is no other record afterwards with the same `IDCODE2`) because after completing this approach for each variable (on isoniazid, then other drugs), all that is retained are the records in which `SPECSEQ=1`:

```
select specseq=1
```

Task:

- o ***Recode the various variables for isoniazid drug susceptibility test results (phenotypic and genotypic results). Recode to a single variable for the isoniazid result. Discuss the hierarchy that you wish to assign to this end.***

Solution:

This might be a somewhat arbitrary classification and will surely required expert input. We chose here the following hierarchy (this is specific for isoniazid, but must be defined for each individual drug, and for other drugs this may differ):

Phenotypic result > genotypic result

In the absence of a phenotypic result, the genotypic result counts

If there are genotypically mutations in both the *katG* and *inhA* promoter gene, then the resistance is high-level, whatever the result of phenotypic testing may be. Thus, we coded:

```
define inh #
inh=9
if inhlevel=0  then inh=inhmolsum
if inhlevel=1  then inh=1
if inhlevel=2  then inh=2
if inhlevel=3  then inh=2
if inhlevel=4  then inh=3
if inhmolsum=3 then inh=3
if inha=2 and katg=2 then inh=3 // already done, but make sure
label inh "Resistance to INH"
```

```
labelvalue inh /0="No result"
labelvalue inh /1="Susceptible"
labelvalue inh /2="Low-level resistance"
labelvalue inh /3="High-level resistance"
```

The entire `b_ex04_solution.pgm` reads as:

```
* Part B, Exercise 4
*  Import spreadsheet data and create identifiers
* EpiData course
* Author: Hans L Rieder
* First   version: 05 Feb 2018
* Current version: 02 Mar 2018


*************************************************
* 1) Read data from spreadsheet and save to EpiData file
cls
close
logclose

* read /cb
* savedata "b_ex04_task_in.rec" /replace


*************************************************
* 2) Read EpiData file, make basic checks and create new
*    identifiers:
*    idcode : idcode                [original, derived]
*    idcode0: country-idcode        [original extended]
*    idcode2: country-idcode-month [original extended]
cls
close
logclose

read "b_ex04_task_in.rec"

cls
* Manually inspect for typing errors and correct
if substr(idcode,1,4)="x- 2" then \
   idcode=substr(idcode,1,2)+substr(idcode,4,11)
if substr(idcode,7,1)<>"-"   then \
   idcode=substr(idcode,1,6)+"-"+substr(idcode,7,6)

cls
* Exclude records without valid identifier
*  <= they cannot be used
gen i hasid=1
if lower(substr(idcode, 1,3))="xxx" then hasid=0
if lower(substr(idcode,11,3))="xxx" then hasid=0
select hasid=1
drop hasid

cls
* freq mm /m
define mmn ##
mmn=99
mmn=integer(mm) if mm<>"Unknown"
drop mm
rename mmn to mm
label mm "Month of treatment"
```

```
cls
* Create text month with leading zero
*  to allow correct alphabetical sorting
gen s(2) mmtxt=mm
if mm<10 then mmtxt="0"+mm
select mm<>99

cls
* freq country
define countryn #
if country="A" then countryn=1
if country="B" then countryn=2
if country="C" then countryn=3
if country="D" then countryn=4
if country="E" then countryn=5
if country="F" then countryn=6
if country="G" then countryn=7
if country="H" then countryn=8
drop country
rename countryn to country
label country "Name of country / jurisdiction"
labelvalue country /1="Jurisdiction A"
labelvalue country /2="Jurisdiction B"
labelvalue country /3="Jurisdiction C"
labelvalue country /4="Jurisdiction D"
labelvalue country /5="Jurisdiction E"
labelvalue country /6="Jurisdiction F"
labelvalue country /7="Jurisdiction G"
labelvalue country /8="Jurisdiction H"

cls
* Create derived identifiers
gen s(16) idcoden=country+substr(idcode,2,12)+"-"+mmtxt
drop idcode2 idcode
rename idcoden to idcode2
label idcode2 "Country-patient-month identifier"
cls
gen s(13) idcode=substr(idcode2,1,13)
label idcode "Country-patient identifier"
cls
gen s(11) idcode0=substr(idcode2,3,11)
label idcode0 "Patient identifier"

label spnr "Specimen number"

cls
* Reduce dataset size for faster processing
keep idcode0 idcode idcode2 country mm spnr \
     h02 h10 h50 inhlevel katg inha inhmol inhmolsum

cls
* Identify multiple specimens per patient-month
sort idcode2 mm
gen i specseq=1
if idcode2=idcode2[_n-1] then specseq=specseq[_n-1]+1
```

```
* freq specseq
* =>
*       N
* 1     582
* 2     26
* 3     5
* 4     4
* 5     2
* 6     2
* Total 621
* => Up to 6 specimens for 1 patient in 1 month

savedata "temp_01.rec" /replace


*************************************************
* 3) Recode to priority on first of multiple specimens
* => Isoniazid

* Hierarchy (code numerically from 0 to highest):
*  High level resistance > low level resistance
*  Resistant > Susceptible
*  Susceptible > No result

cls
close
logclose

read "temp_01.rec"

cls
define h02n #
h02n=0
if h02="S" then h02n=1
if h02="R" then h02n=2
if idcode2[_n+1]=idcode2 and h02n[_n+1]>h02n then h02n=h02n[_n+1]
if idcode2[_n+2]=idcode2 and h02n[_n+2]>h02n then h02n=h02n[_n+2]
if idcode2[_n+3]=idcode2 and h02n[_n+3]>h02n then h02n=h02n[_n+3]
if idcode2[_n+4]=idcode2 and h02n[_n+4]>h02n then h02n=h02n[_n+4]
if idcode2[_n+5]=idcode2 and h02n[_n+5]>h02n then h02n=h02n[_n+5]
drop h02
rename h02n to h02
label h02 "INH result at 0.2 mg/L"

cls
define h10n #
h10n=0
if h10="S" then h10n=1
if h10="R" then h10n=2
if idcode2[_n+1]=idcode2 and h10n[_n+1]>h10n then h10n=h10n[_n+1]
if idcode2[_n+2]=idcode2 and h10n[_n+2]>h10n then h10n=h10n[_n+2]
if idcode2[_n+3]=idcode2 and h10n[_n+3]>h10n then h10n=h10n[_n+3]
if idcode2[_n+4]=idcode2 and h10n[_n+4]>h10n then h10n=h10n[_n+4]
if idcode2[_n+5]=idcode2 and h10n[_n+5]>h10n then h10n=h10n[_n+5]
drop h10
rename h10n to h10
```

```
label h10 "INH result at 1.0 mg/L"

cls
define h50n #
h50n=0
if h50="S" then h50n=1
if h50="R" then h50n=2
if idcode2[_n+1]=idcode2 and h50n[_n+1]>h50n then h50n=h50n[_n+1]
if idcode2[_n+2]=idcode2 and h50n[_n+2]>h50n then h50n=h50n[_n+2]
if idcode2[_n+3]=idcode2 and h50n[_n+3]>h50n then h50n=h50n[_n+3]
if idcode2[_n+4]=idcode2 and h50n[_n+4]>h50n then h50n=h50n[_n+4]
if idcode2[_n+5]=idcode2 and h50n[_n+5]>h50n then h50n=h50n[_n+5]
drop h50
rename h50n to h50
label h50 "INH result at 5.0 mg/L"

labelvalue h02-h50 /0="No result"
labelvalue h02-h50 /1="Susceptible"
labelvalue h02-h50 /2="Resistant"

cls
define inhleveln #
inhleveln=0
if inhlevel="-"    then inhleveln=1 // Susceptible!
if inhlevel="S"    then inhleveln=1 // Susceptible!
if inhlevel="H0,2" then inhleveln=2
if inhlevel="H1"   then inhleveln=3
if inhlevel="H5"   then inhleveln=4
if   idcode2[_n+1]=idcode2   and   inhleveln[_n+1]>inhleveln   then
inhleveln=inhleveln[_n+1]
if   idcode2[_n+2]=idcode2   and   inhleveln[_n+2]>inhleveln   then
inhleveln=inhleveln[_n+2]
if   idcode2[_n+3]=idcode2   and   inhleveln[_n+3]>inhleveln   then
inhleveln=inhleveln[_n+3]
if   idcode2[_n+4]=idcode2   and   inhleveln[_n+4]>inhleveln   then
inhleveln=inhleveln[_n+4]
if   idcode2[_n+5]=idcode2   and   inhleveln[_n+5]>inhleveln   then
inhleveln=inhleveln[_n+5]
drop inhlevel
rename inhleveln to inhlevel
label inhlevel "INH result summary level"
labelvalue inhlevel /0="No result"
labelvalue inhlevel /1="Susceptible"
labelvalue inhlevel /2="Resistant at 0.2 mg/L"
labelvalue inhlevel /3="Resistant at 1.0 mg/L"
labelvalue inhlevel /4="Resistant at 5.0 mg/L"

cls
define katgn #
katgn=0
if katg="-"                        then katgn=1
if lower(substr(katg,1,4))="wild" then katgn=1
if lower(substr(katg,1,3))="del"  then katgn=2
if (substr(katg,1,3))     ="315"  then katgn=2
if lower(substr(katg,1,3))="mix"  then katgn=2
if lower(substr(katg,1,3))="mut"  then katgn=2
if idcode2[_n+1]=idcode2 and katgn[_n+1]>katgn then katgn=katgn[_n+1]
if idcode2[_n+2]=idcode2 and katgn[_n+2]>katgn then katgn=katgn[_n+2]
if idcode2[_n+3]=idcode2 and katgn[_n+3]>katgn then katgn=katgn[_n+3]
if idcode2[_n+4]=idcode2 and katgn[_n+4]>katgn then katgn=katgn[_n+4]
if idcode2[_n+5]=idcode2 and katgn[_n+5]>katgn then katgn=katgn[_n+5]
drop katg
rename katgn to katg
label katg "INH katG result"
labelvalue katg /0="No result"
labelvalue katg /1="No mutation"
labelvalue katg /2="Mutation"

cls
define inhan #
inhan=0
if lower(substr(inha,11,4))="wild" then inhan=1
if lower(substr(inha,15,4))="wild" then inhan=1
if lower(substr(inha, 1,1))="c"    then inhan=2
if lower(substr(inha, 1,1))="g"    then inhan=2
if lower(substr(inha, 1,3))="del"  then inhan=2
if lower(substr(inha, 1,3))="mut"  then inhan=2
if lower(substr(inha,11,3))="mut"  then inhan=2
if idcode2[_n+1]=idcode2 and inhan[_n+1]>inhan then inhan=inhan[_n+1]
if idcode2[_n+2]=idcode2 and inhan[_n+2]>inhan then inhan=inhan[_n+2]
if idcode2[_n+3]=idcode2 and inhan[_n+3]>inhan then inhan=inhan[_n+3]
if idcode2[_n+4]=idcode2 and inhan[_n+4]>inhan then inhan=inhan[_n+4]
if idcode2[_n+5]=idcode2 and inhan[_n+5]>inhan then inhan=inhan[_n+5]
drop inha
rename inhan to inha
label inha "INH inha result"
labelvalue inha /0="No result"
labelvalue inha /1="No mutation"
labelvalue inha /2="Mutation"

cls
define inhmoln #
inhmoln=0
if lower(substr(inhmol,1,1))="s"  then inhmoln=1
if lower(substr(inhmol,1,1))="s?" then inhmoln=0
if lower(substr(inhmol,1,2))="rb" then inhmoln=2
if lower(substr(inhmol,1,2))="rh" then inhmoln=3
if    idcode2[_n+1]=idcode2    and    inhmoln[_n+1]>inhmoln    then
inhmoln=inhmoln[_n+1]
if    idcode2[_n+2]=idcode2    and    inhmoln[_n+2]>inhmoln    then
inhmoln=inhmoln[_n+2]
if    idcode2[_n+3]=idcode2    and    inhmoln[_n+3]>inhmoln    then
inhmoln=inhmoln[_n+3]
if    idcode2[_n+4]=idcode2    and    inhmoln[_n+4]>inhmoln    then
inhmoln=inhmoln[_n+4]
if    idcode2[_n+5]=idcode2    and    inhmoln[_n+5]>inhmoln    then
inhmoln=inhmoln[_n+5]
drop inhmol
rename inhmoln to inhmol
label inhmol "INH inhmol result"
labelvalue inhmol /0="No result"
labelvalue inhmol /1="Susceptible"
labelvalue inhmol /2="Low level resistance"
labelvalue inhmol /3="High level resistance"

cls
define inhmolsumn #
inhmolsumn=0
if lower(substr(inhmolsum,1,1))="-"  then inhmolsumn=1
if lower(substr(inhmolsum,1,2))="hb" then inhmolsumn=2
if lower(substr(inhmolsum,1,2))="hh" then inhmolsumn=3
if   idcode2[_n+1]=idcode2   and   inhmolsumn[_n+1]>inhmolsumn   then
inhmolsumn=inhmolsumn[_n+1]
if   idcode2[_n+2]=idcode2   and   inhmolsumn[_n+2]>inhmolsumn   then
inhmolsumn=inhmolsumn[_n+2]
if   idcode2[_n+3]=idcode2   and   inhmolsumn[_n+3]>inhmolsumn   then
inhmolsumn=inhmolsumn[_n+3]
if   idcode2[_n+4]=idcode2   and   inhmolsumn[_n+4]>inhmolsumn   then
inhmolsumn=inhmolsumn[_n+4]
if   idcode2[_n+5]=idcode2   and   inhmolsumn[_n+5]>inhmolsumn   then
inhmolsumn=inhmolsumn[_n+5]
drop inhmolsum
rename inhmolsumn to inhmolsum
label inhmolsum "INH inhmolsum result"
labelvalue inhmolsum /0="No result"
labelvalue inhmolsum /1="Susceptible"
labelvalue inhmolsum /2="Low level resistance"
labelvalue inhmolsum /3="High level resistance"

savedata "temp_02.rec" /replace


*************************************************
* 4) Recode to priority on first of multiple specimens
* => Next drug

cls
close
logclose
read "temp_02.rec"

* Do this for each drug
* At the end of the process, the hierarchically
*   highest value will be in the sequentially
*   first SPNR if there are multiple SPNRs
* => keeping only the first SPNR suffices:

select specseq=1

* Create a single result for isoniazid

define inh #
inh=9
if inhlevel=0  then inh=inhmolsum
if inhlevel=1  then inh=1
if inhlevel=2  then inh=2
if inhlevel=3  then inh=2
if inhlevel=4  then inh=3
if inhmolsum=3 then inh=3
if inha=2 and katg=2 then inh=3 // already done, but make sure
label inh "Resistance to INH"
labelvalue inh /0="No result"
labelvalue inh /1="Susceptible"
labelvalue inh /2="Low-level resistance"
labelvalue inh /3="High-level resistance"

savedata "temp_03.rec" /replace


*************************************************
* 5) Any other recoding

cls
close
logclose
read "temp_03.rec"

* xxx

drop spnr specseq idcode0 idcode2
drop h02 h10 h50 inhlevel
drop katg inha inhmol inhmolsum

sort country idcode mm

savedata "b_ex04_task_out.rec" /replace


**************************************
* Clean up and erase temporary session files

set echo=off
close
define yesno # global
yesno=?Delete all temporary files: 1=yes 0=no?
imif yesno=1 then
cls
type "Be patient ... you will be alerted upon completion" /h2
   erase "temp_01.chk"
   erase "temp_01.rec"
   erase "temp_02.chk"
   erase "temp_02.rec"
   erase "temp_03.chk"
   erase "temp_03.rec"
  cls
  type "All temporary files erased" /h2
 else
  type "All temporary files retained" /h2
endif
set echo=on


**************************************
cls
```

```
close
read "b_ex04_task_out.rec"
```