Exercise 3: Aggregating data and saving the summary data in a file

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

a. Understand “Aggregate” data and apply it to a specific task

We are quite familiar with:

```
cls
close
logclose

read "abcd.rec"

freq sex
```

getting as result:

Examinee's sex

	N
Female	109
Male	191
Total	300

If we replace “`freq`” with “`agg`” (short for `aggregate`):

```
agg sex
```

we get essentially the same information (without the marginal, i.e. here the total):

sex	N
Female	109
Male	191

Making a frequency of a variable is nothing other than aggregating the values of that variable to the smallest common denominator and the same thing is done with `aggregate`. We can do it analogously for a table:

```
tables sex labcode
```

This gives (discounting the marginal) 8 inner cells for the 2 by 4 possibilities:

	Examinee's sex		
Laboratory code	**Female**	**Male**	**Total**
A	33	42	75
B	31	44	75
C	24	51	75
D	21	54	75
Total	109	191	300

If we replace `tables` by `aggregate`:

```
agg sex labcode
```

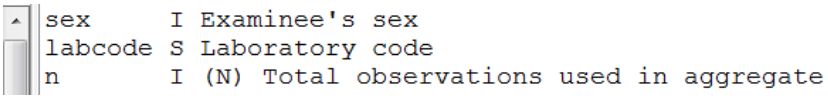
we get the 8 inner cells listed in a column sorted by `sex` then by `labcode`:

sex	labcode	N
Female	A	33
Female	B	31
Female	C	24
Female	D	21
Male	A	42
Male	B	44
Male	C	51
Male	D	54

As an option we can `close` the file that was open (the `abcd.rec`):

```
agg sex labcode /close
```

and we can see what remains in the Variables window:



We can `browse` it as it has been written to an EpiData REC file, aggregated into 8 records:

	sex	labcode	N
1	Female	A	33
2	Female	B	31
3	Female	C	24
4	Female	D	21
5	Male	A	42
6	Male	B	44
7	Male	C	51
8	Male	D	54

As an additional option we can save this aggregation to a REC file (note that the option is "`/save`", not "`/savedata`":

```
agg sex labcode /close /save="labsex_set.rec" /replace
```

then close everything and re-open either file.

The first property of the `aggregate` command is thus that it can replace the summary of a `tables` command and save the cells of the table (made from 1, 2, or more variables) in a REC file.

The second property of the `aggregate` command is related to the power it provides with options. For example:

```
cls
close
logclose
read "abcd.rec"
agg sex /mean=age
```

gives:

sex	N	Nage	MEAage
Female	109	109	39.11
Male	191	191	38.71

We have now the mean `age` by `sex`, but we could also get the mean `age` by `sex` and laboratory:

```
agg sex labcode /mean=age
```

sex	labcode	N	Nage	MEAage
Female	A	33	33	42.97
Female	B	31	31	34.87
Female	C	24	24	37.75
Female	D	21	21	40.86
Male	A	42	42	37.60
Male	B	44	44	35.11
Male	C	51	51	39.24
Male	D	54	54	42.00

and as above, all can be written into a `REC` file:

```
agg sex labcode /mean=age /close /save="labsex_set_2.rec" /replace
cls
close
read "labsex_set_2.rec"
```

	sex	labcode	N	Nage	MEAage
1	Female	A	33	33	42.9696969697
2	Female	B	31	31	34.8709677419
3	Female	C	24	24	37.7500000000
4	Female	D	21	21	40.8571428571
5	Male	A	42	42	37.5952380952
6	Male	B	44	44	35.1136363636
7	Male	C	51	51	39.2352941176
8	Male	D	54	54	42.0000000000

The automatically created variable `N` denotes the number of records in the original file, `Nage` the number of records in the original file with non-missing information on `age`, and `MEAage`, the mean `age` in the aggregated stratum.

There are more options of this kind, like:

```
/min
/max
/stat
/sum
```

You can also cumulate different options, and you can use the same option for different variables, e.g. "`/sum=var1 /sum=var2 /min=var3`".

It is the option `/sum` we are particularly interested here to apply. We can sum up all the values of a given variable across all records aggregated within a stratum. For instance, in the following task you will need to count the number of smears each examinee had, and then you can sum up all the smears within each stratum resulting from your aggregation.

Workload in the laboratory

One measure of the workload in laboratories is the number of smears they have to examine per day. Even the busiest laboratories do not work every day, they may close on weekends

and public holidays. One may get some approximate estimate of the number of working days, but a much cleaner way is to count the actual working days. The tuberculosis laboratory register gives a possible good approximation with the date of registration. While smears are also examined on other than the registration date (the first specimen defines that date, and a patient gives the first specimen on the spot but brings in an early morning specimen one day later), this is a reasonably good approximation to the number of days work was actually carried out, certainly better than some other approximation without a good data basis.

We provide a data set "`b_ex03_workload.rec`" from three laboratories in Zimbabwe (data courtesy Dr Biggie Mabaera), each complete but also limited to during one calendar year.

Task:

- ***The B_EX03_WORKLOAD.REC has been edited to contain only three laboratories (out of the original 30) and only the year 2002. Nonsensical results (e.g., first examination not recorded, followed by a valid result) have been excluded. Create a program B_EX03.PGM to provide the mean number of smears examined per registration day in each of the three laboratories.***