

The log window tells us that we have 4 observations with 9 variables. We have lost a person!!! SAS doesn't recognised the extra spaces as missing values – it just keeps reading along the data line until it hits a number and reads that.

So for person number 3 who has missing value for weight. SAS has read in their SES score of 4 for their weight. And subject 4's id for subject 3's SES score. SAS has then discarded the rest of subject 4's data and reads in the data for subject 5.

To get round this problem of the data not being read in correctly – we have to give SAS more instructions. We need to tell SAS which column each of the variables is in using @

ID variable is in column 1 – so before the variable name ID we put @1

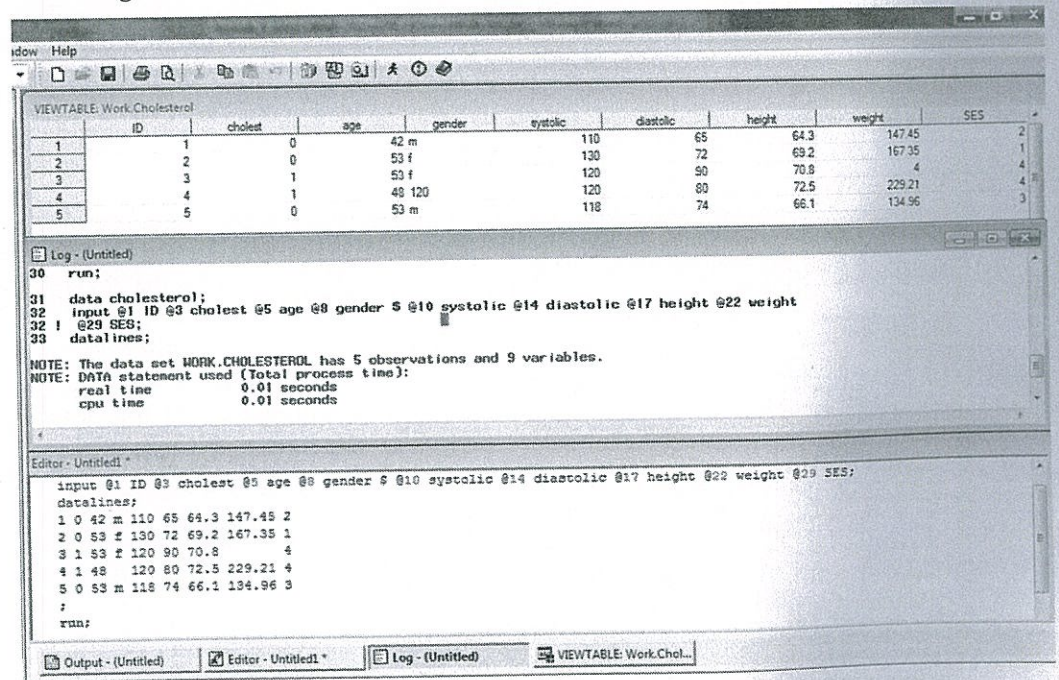
input @1 ID cholest age gender \$ systolic diastolic height weight SES; *This is the question*

Now count along a line of data – what column does the variable cholest start? Column 3. Adding all the columns where the variables start to the SAS program – you should have

```
data cholesterol;
input @1 ID @3 cholest @5 age @8 gender $ @10 systolic @14 diastolic @17 height @22 weight @29 SES;
datalines;
1 0 42 m 110 65 64.3 147.45 2
2 0 53 f 130 72 69.2 167.35 1
3 1 53 f 120 90 70.8 4
4 1 48 120 80 72.5 229.21 4
5 0 53 m 118 74 66.1 134.96 3
run;
```

how do you count?

You will get



The log window tells us that we have 5 observations with 9 variables, which is what we were expecting. But SAS has filled in missing values with values for the next variables. Subject with ID 4 has a gender of 120. So still not correct!!!

When we have missing data, we need to tell SAS which column the data is in and also the format of the data for each variable.

In SAS, formats in the data statements are called *informats*. Informats determine how raw data values are read and stored. For string variables we tell SAS how long the variable is and it has the syntax \$w. - \$ to show it is a character variable and length w.