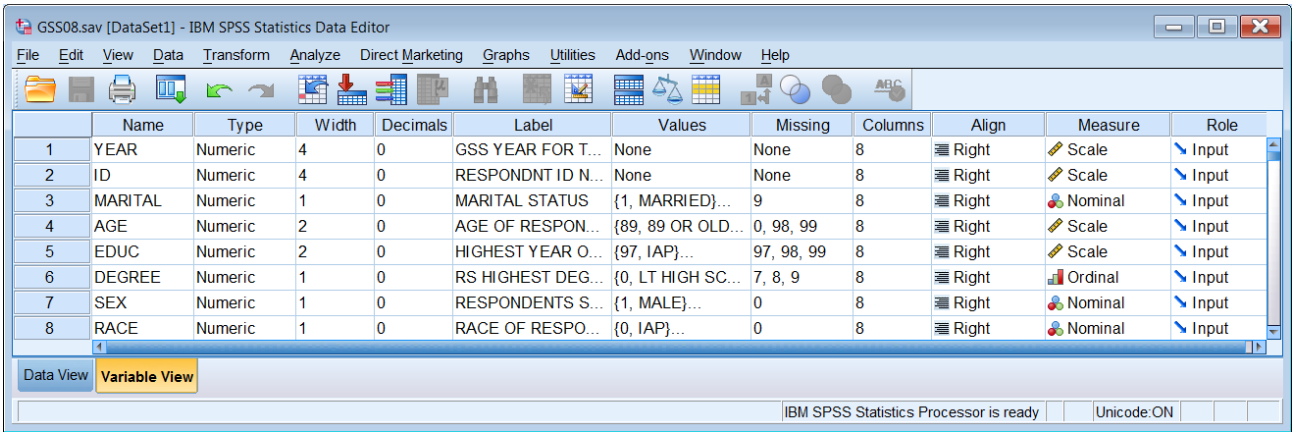
**First encounter with GSS08.SAV**[1] (subset derived from full NORC version gss2008.sav)

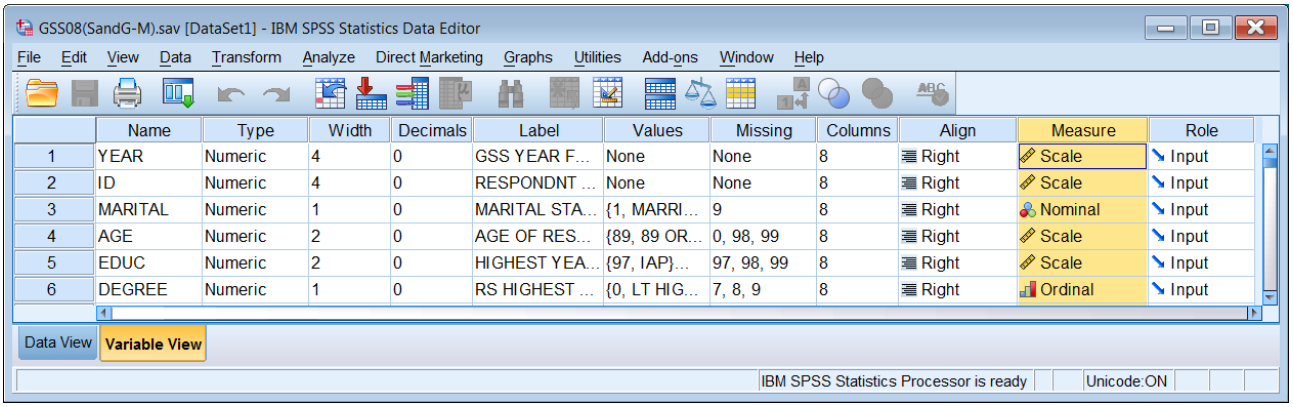
John F Hall [Draft only 15 December 2013]

Here's what I tend to do with new-to-me SPSS saved files the first time I open them.

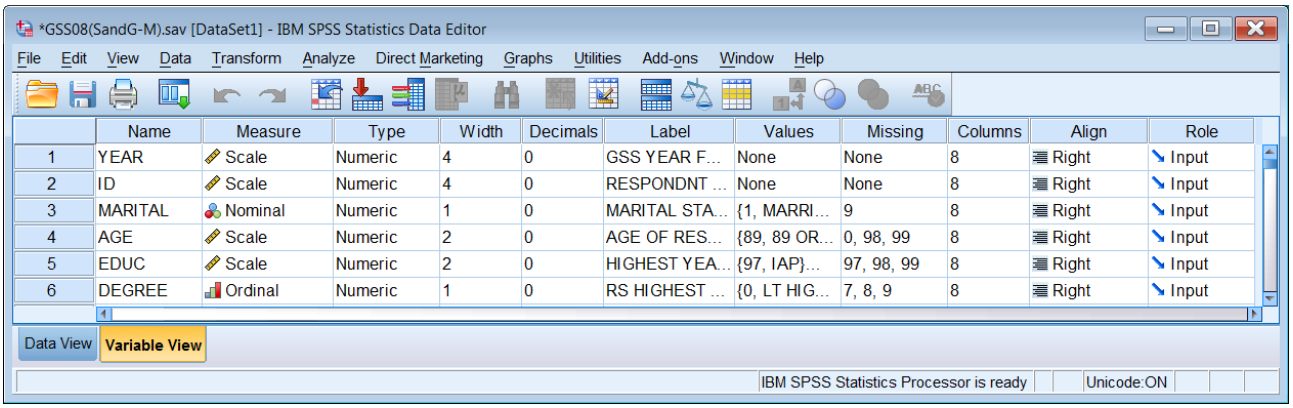



For historical reasons, GSS08.sav has 8-character maximum variable names, but SPSS can now have much longer names. Variable labels and value labels have a maximum of 40 characters and 20 characters respectively, but both can also now be much longer. All text is in UPPER CASE.

The first thing I tend to do is change the column positions around, moving the (to me) more important columns to the left. Left click once on **Measure** to highlight the whole column:

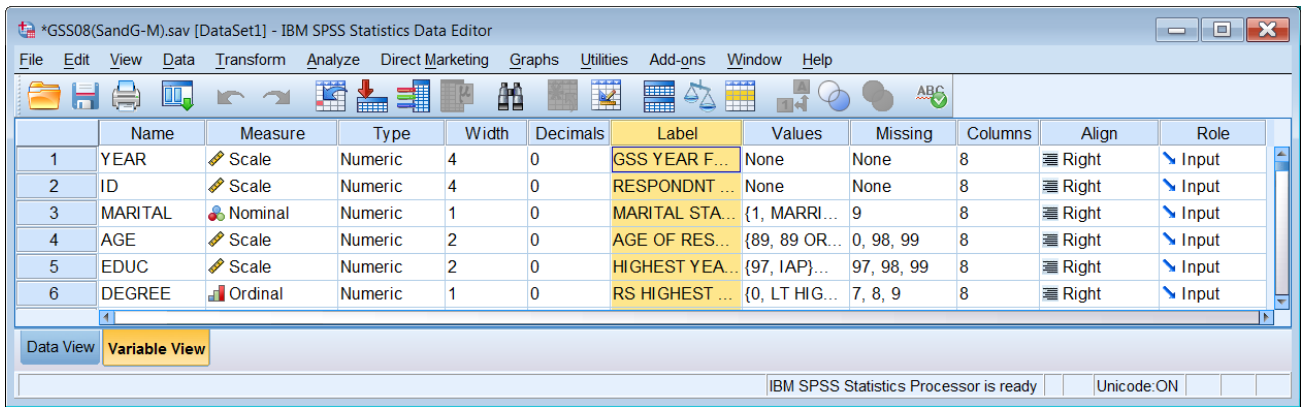


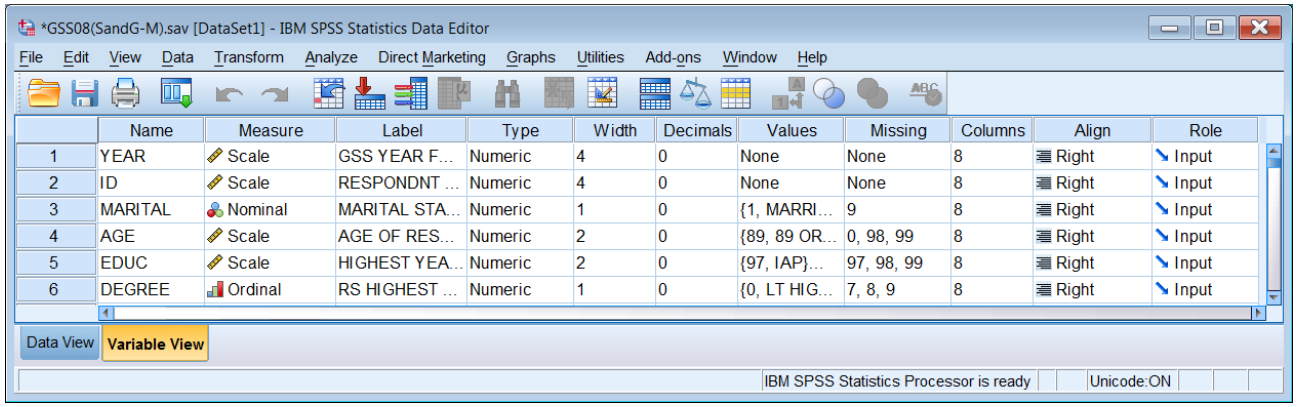

Click and hold down the left mouse button on **Measure** to drag it to the new position after **Name**: a thin vertical red line indicates the destination currently reached.

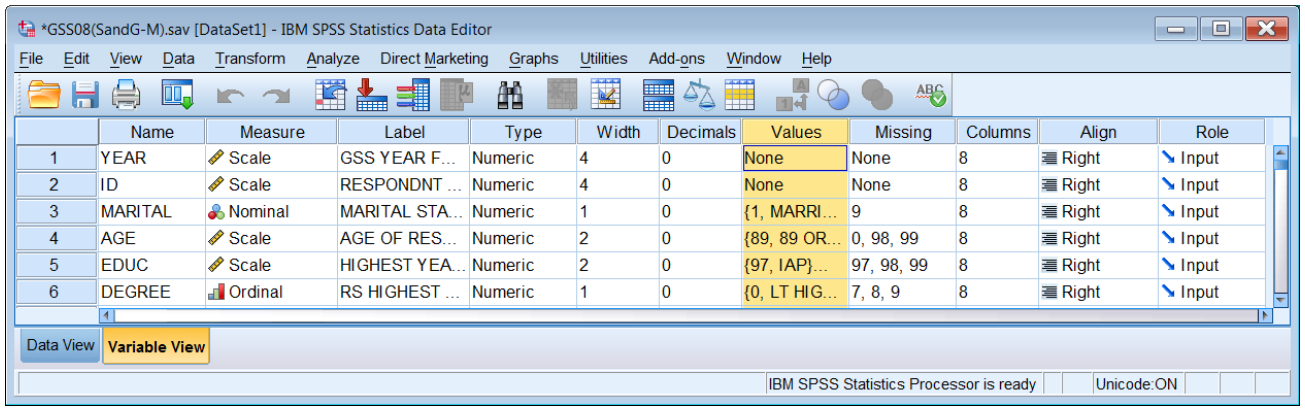


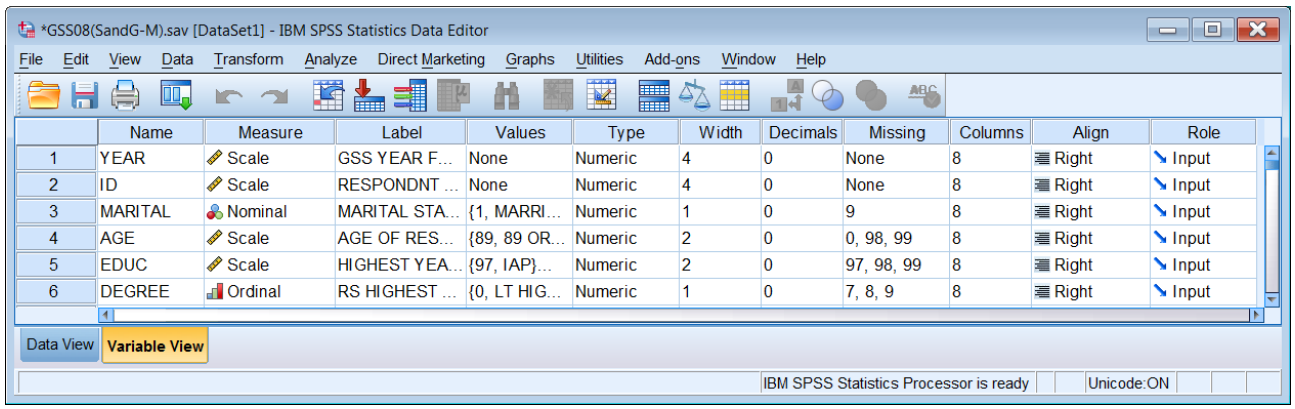

[1] gss08.sav is a reduced data set (125 variables) used by Sweet and Grace-Martin, selected from the full gss2008.sav (843 variables) distributed by the Roper Center. Both sets have 2023 cases.

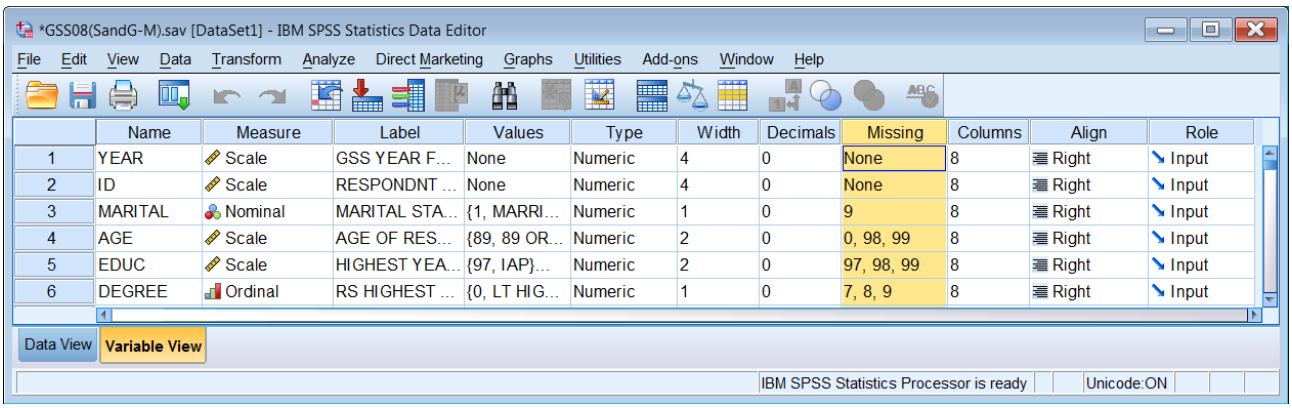
Repeat for **Label**, **Values** and **Missing**:

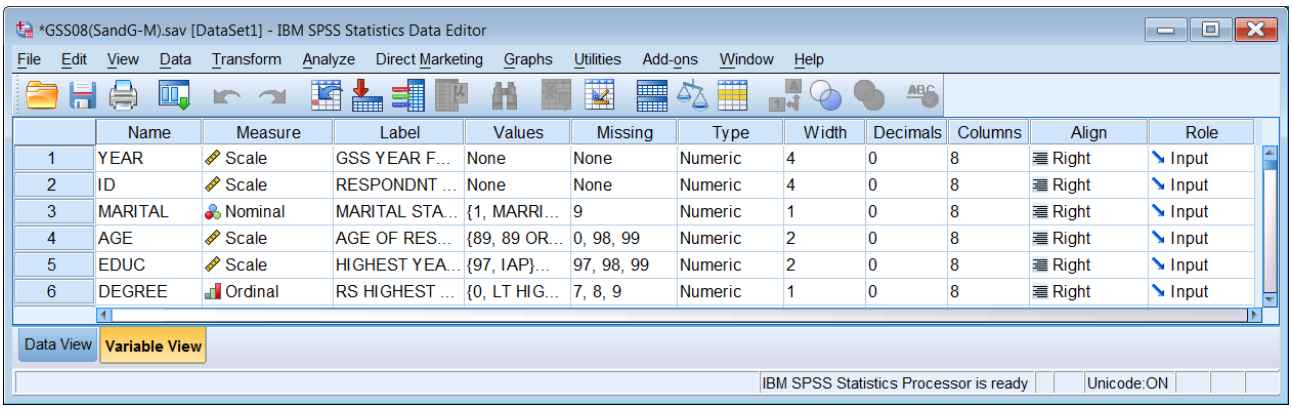


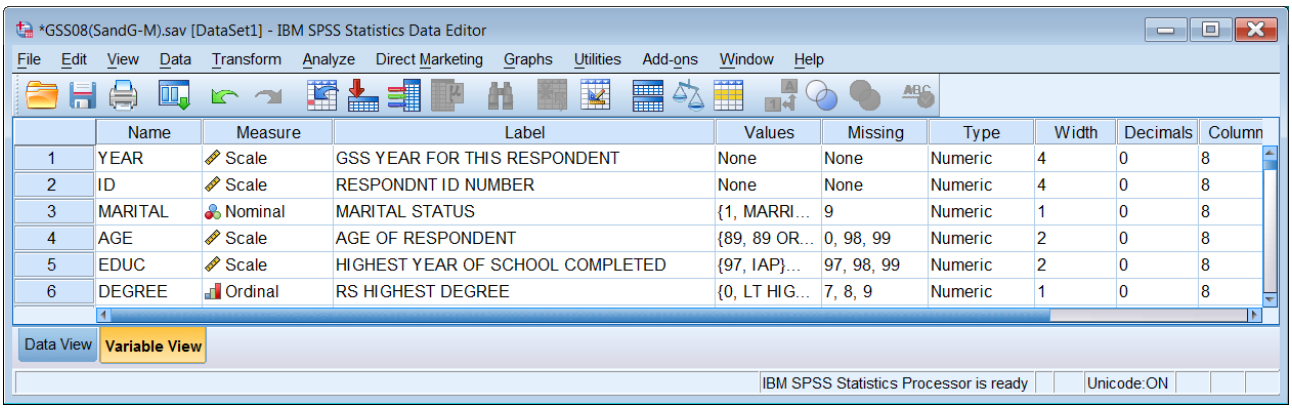



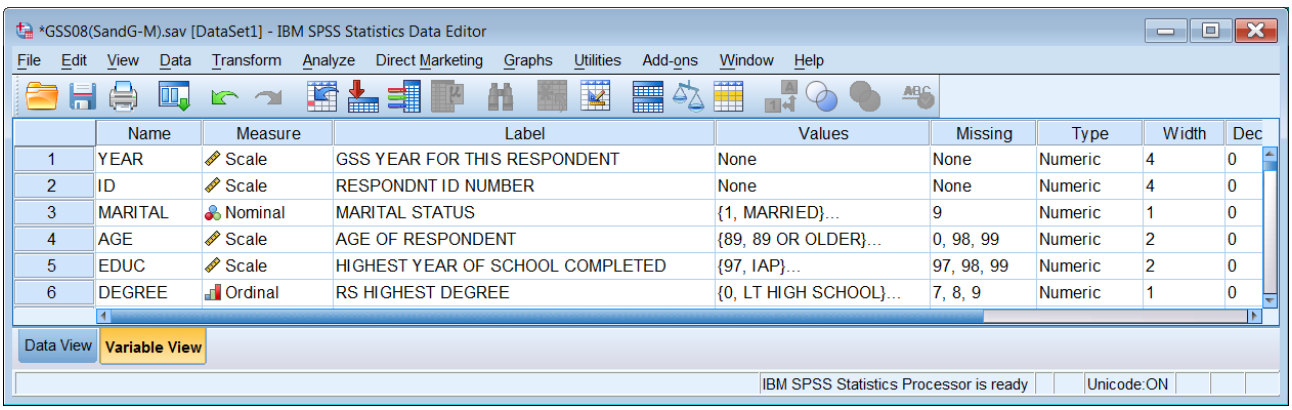


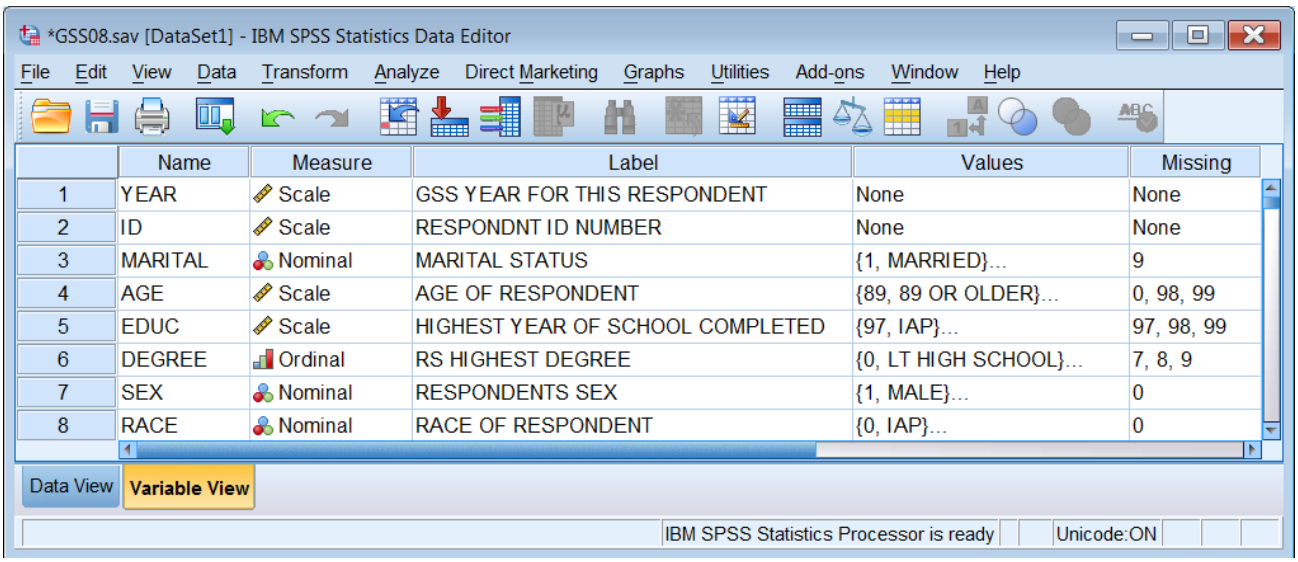

In the **Label** column header, drag right hand column separator sideways to see the full text:




Now do the same for **Values**:

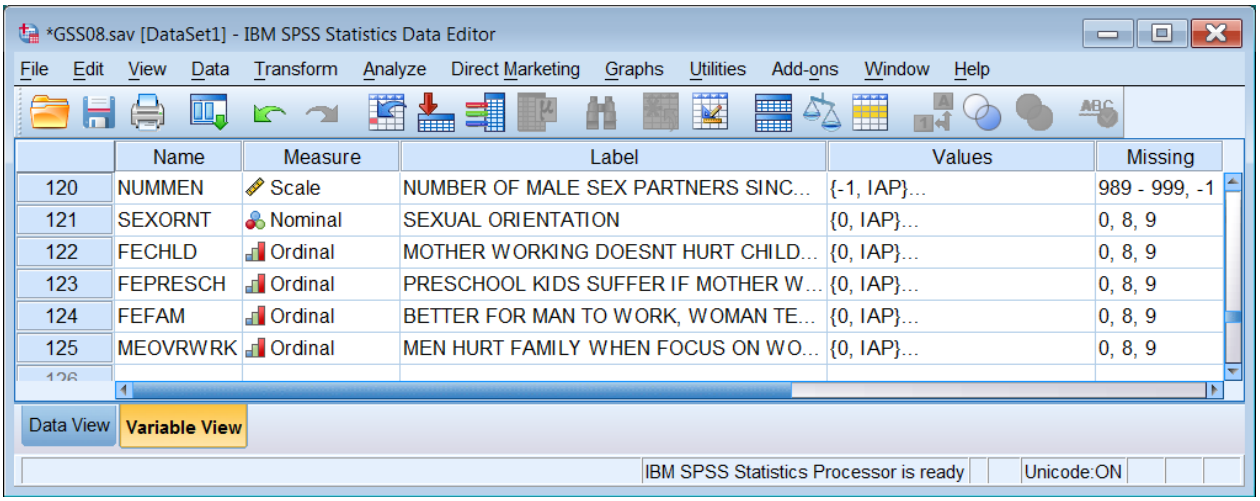



and reduce the window as you don't really need the other columns (yet).

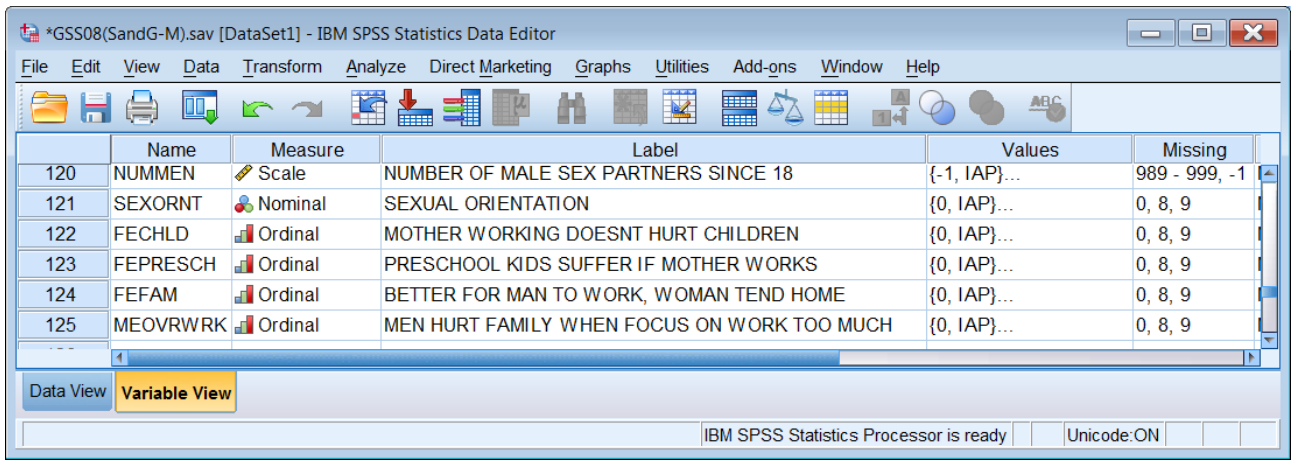



Variable levels have been set, but ID should perhaps be Nominal (or at least Ordinal)?

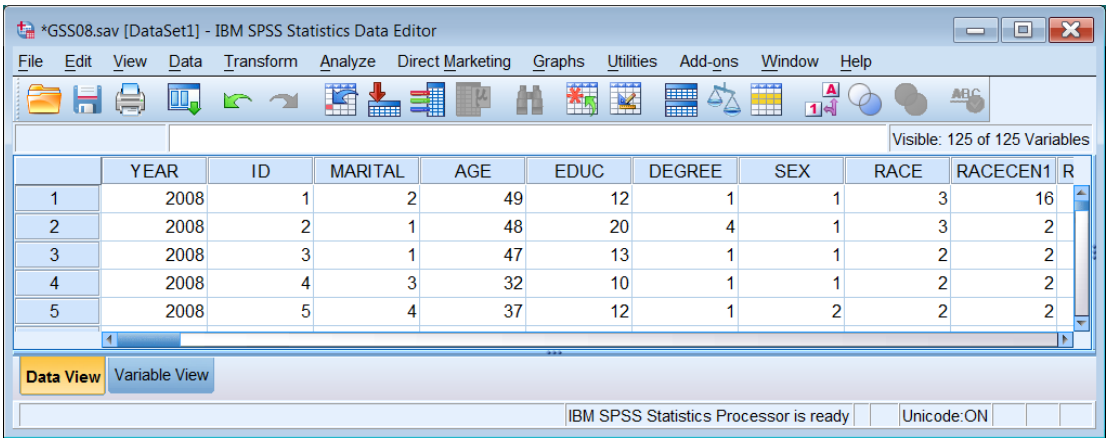
There are 125 variables (scroll to end of file in Variable View):



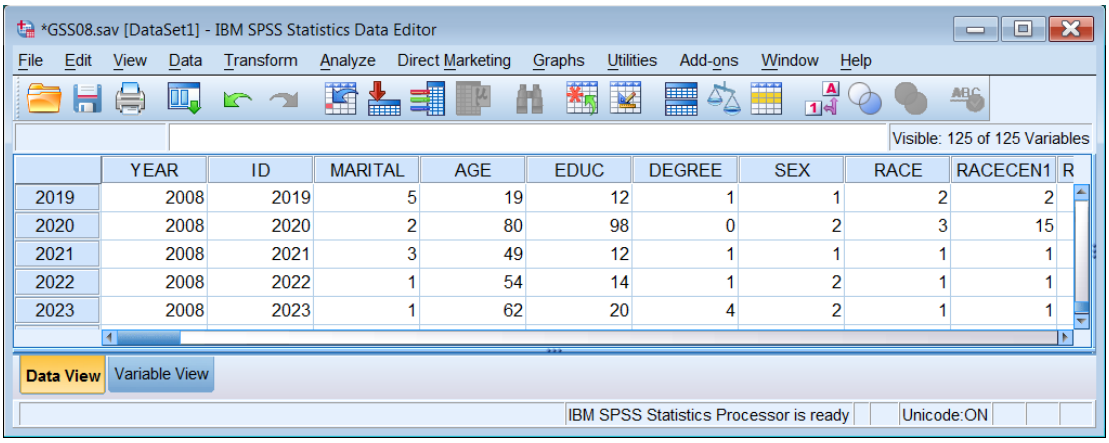

The **Label** column is not quite wide enough to see the full text, so drag separator out a bit more:



Switch to Data View:



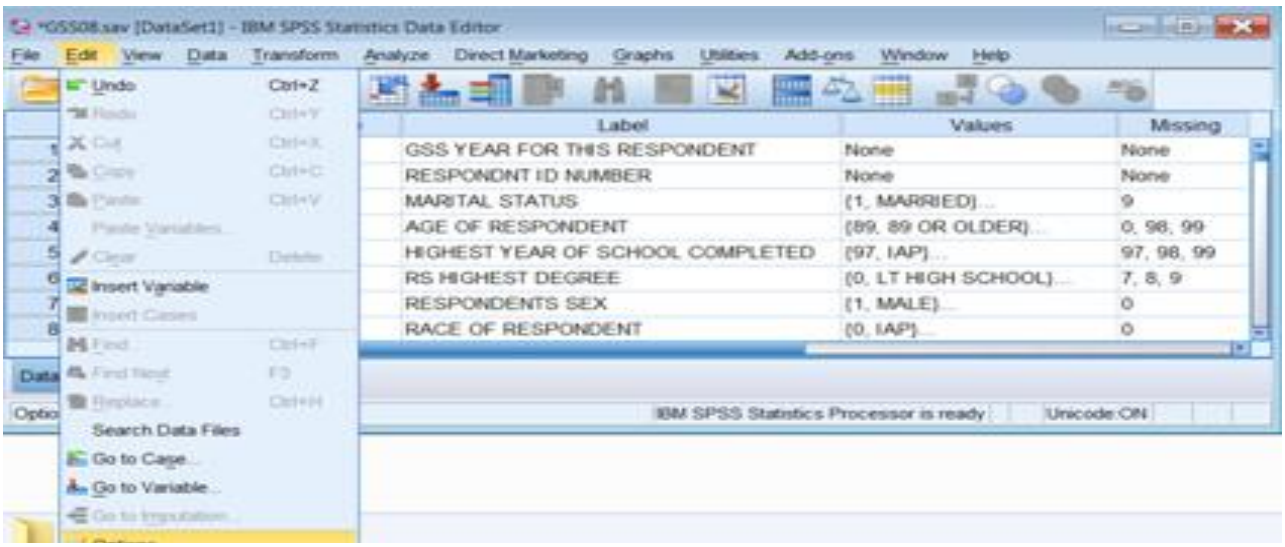

and scroll to end of file to see that there are 2023 cases:



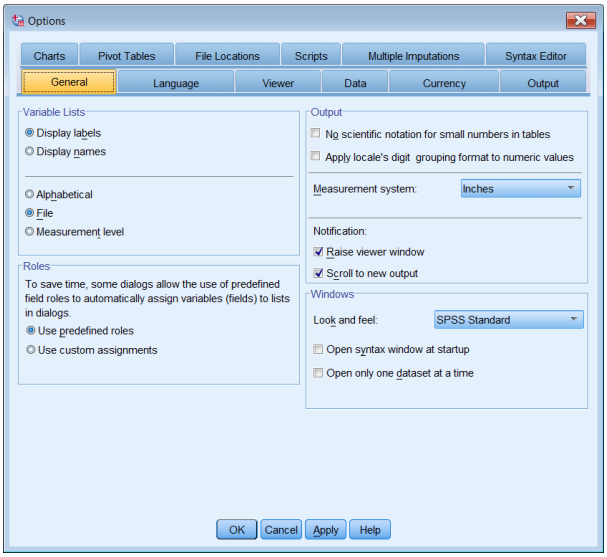

I prefer working in syntax to point-and-click from the GUI menus, so I always change the SPSS settings to open a new syntax file on startup.

In Variable View click on:

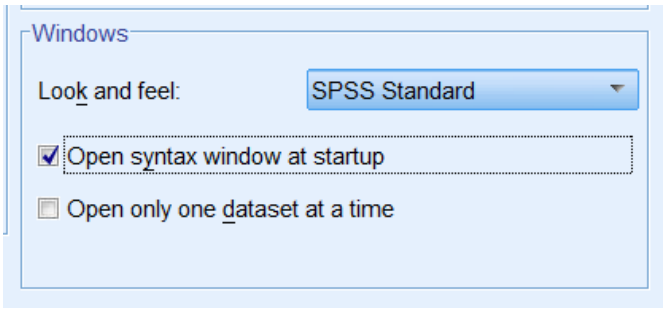
Edit > Options



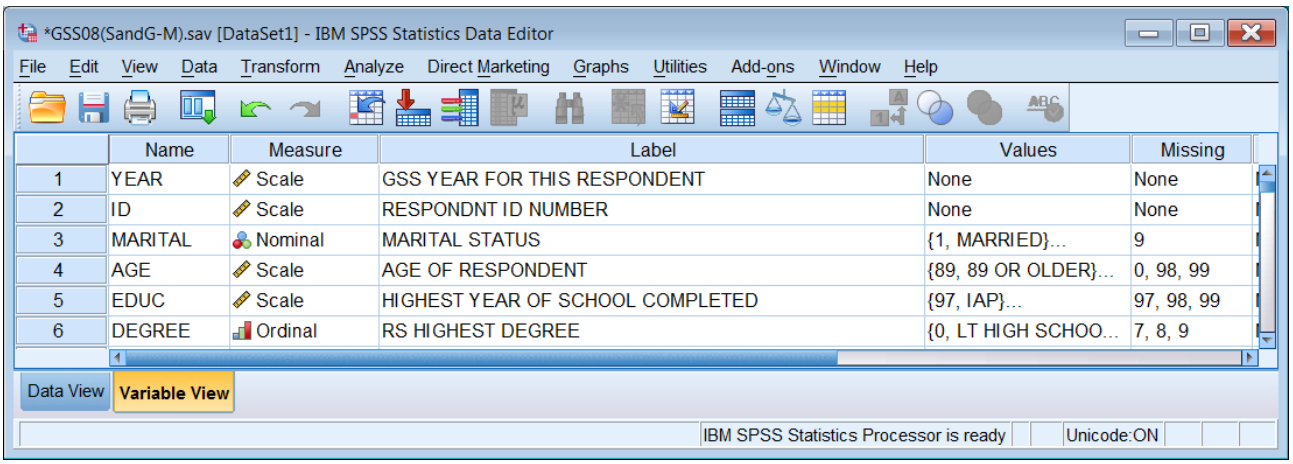
In the General tab




Check box Open syntax window at startup:

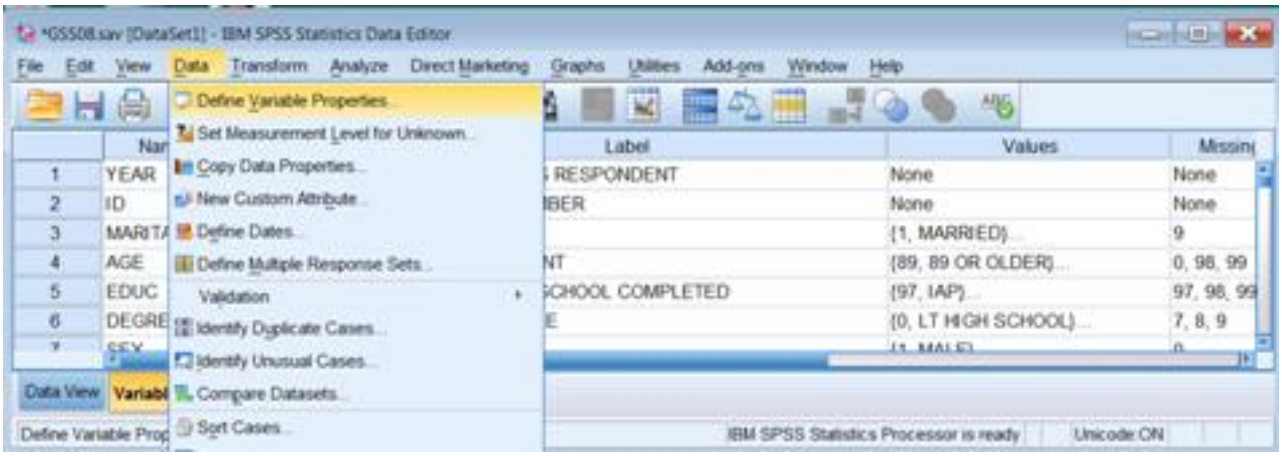



Click on Apply and then OK.

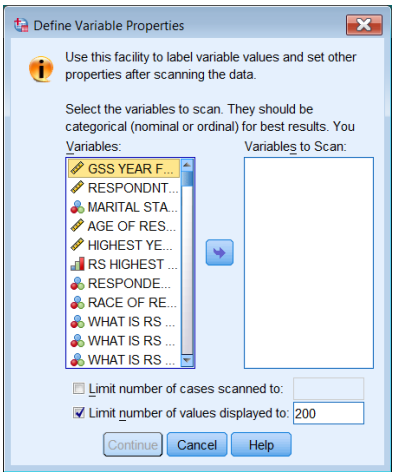


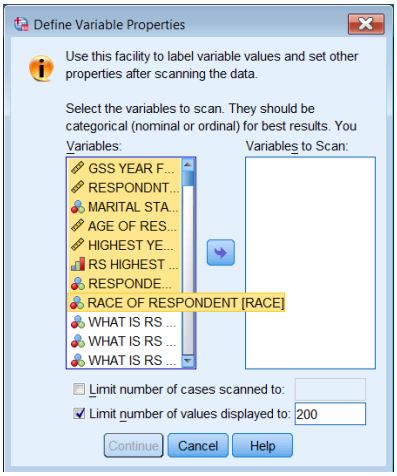

You can do quick checks on the file with:

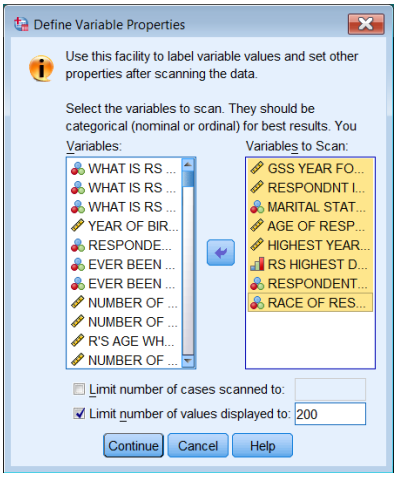
Data > Define Variable Properties



Opening window: Highlight variables in left pane Transfer to right pane

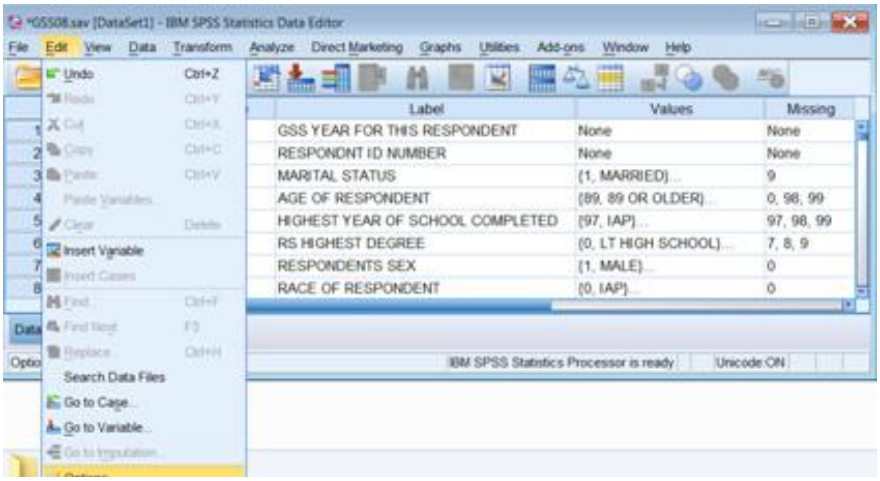


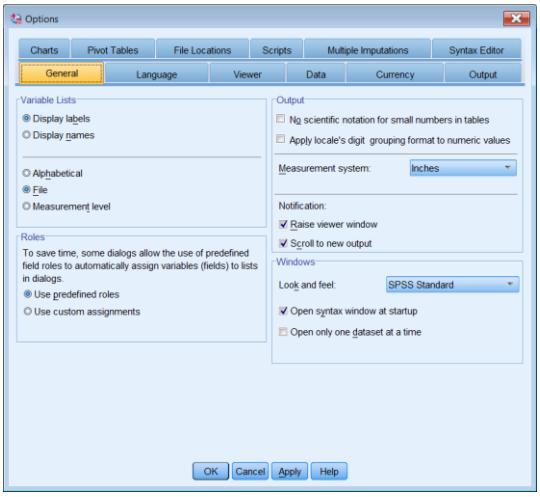



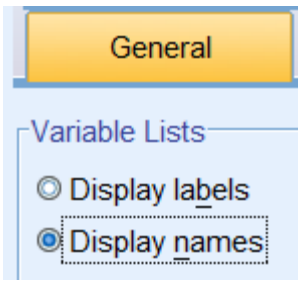

This is somewhat cumbersome as the display is of variable labels. There aren't any question numbers to help working from the questionnaire, so we need to use variable names instead. Again this can be done by changing the SPSS settings.

Edit > Options



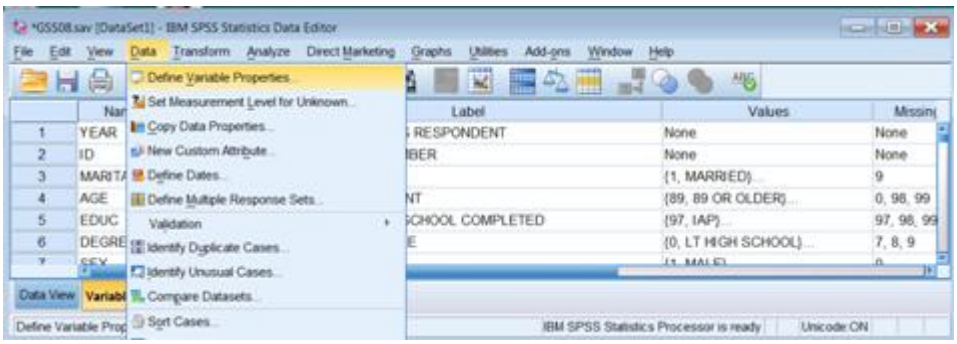

In the General tab click the Display names button



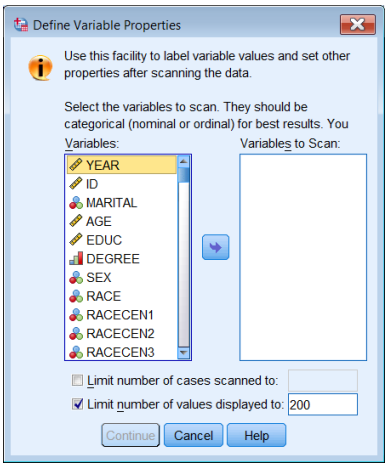

Click on Apply and then OK to go back to the Data Editor.

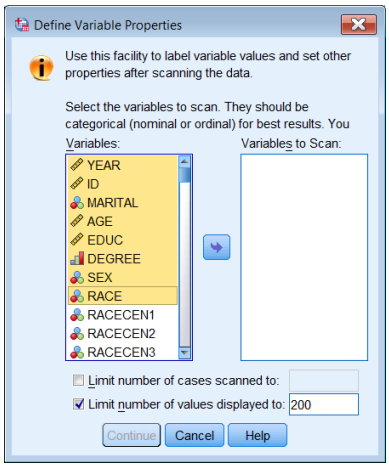
Click on Data > Define Variable Properties again:

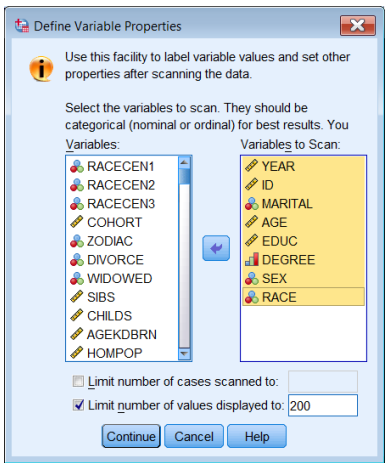



That's better: we can see what we're doing now.

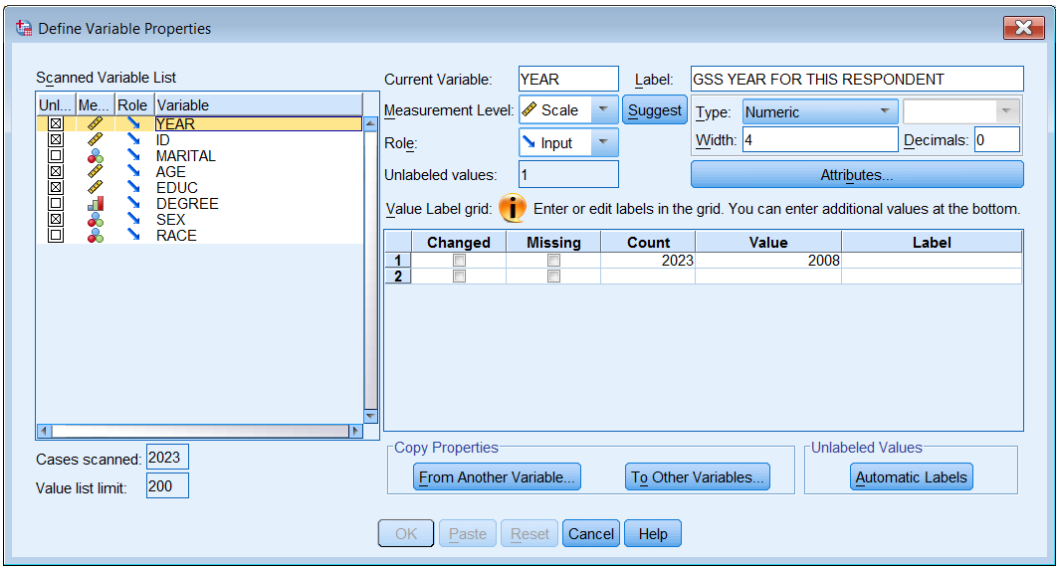
Opening window: Highlight variables in left pane Click  to transfer to right pane



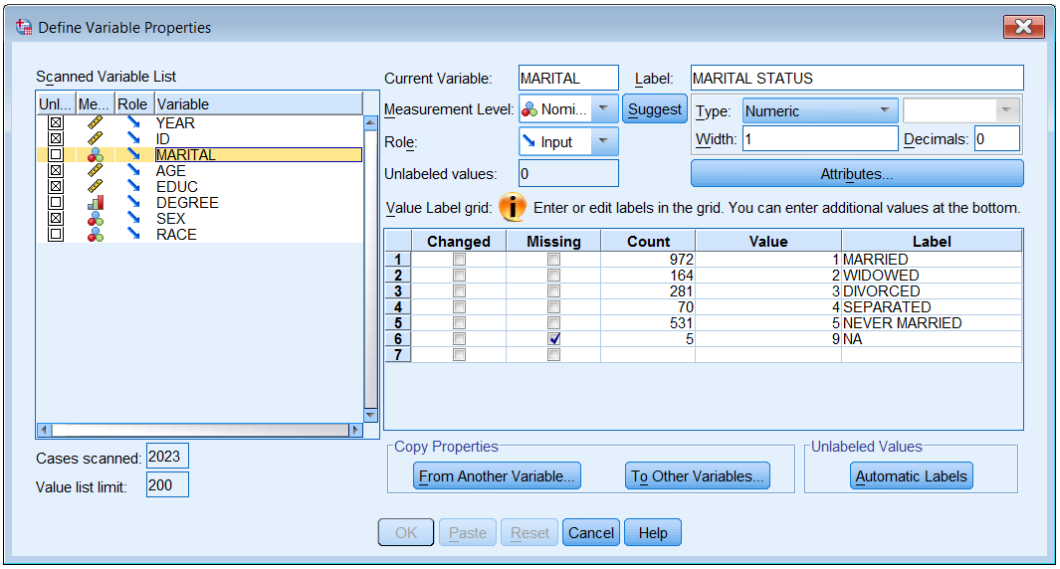



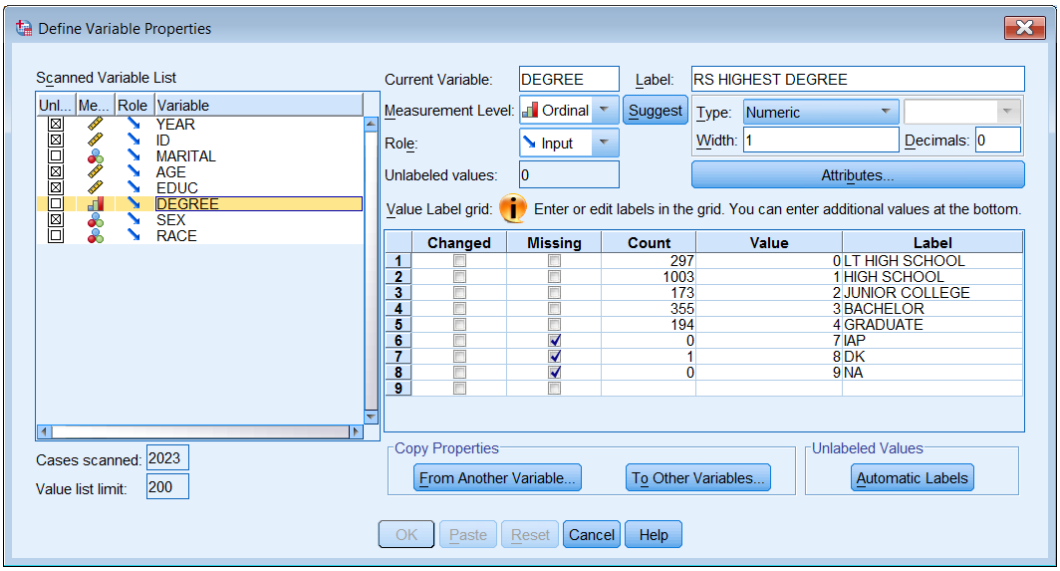




Click Continue:




Click on any variable in the Scanned Variable List pane and all its properties appear on the right, including all values found, checkboxes for any values specified as missing, and (most useful of all) counts for each value.



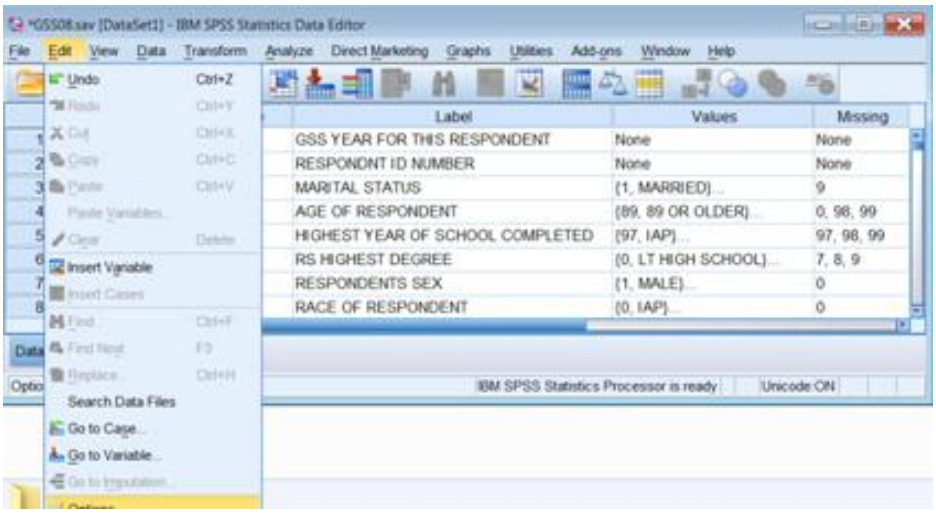




You can edit any or all of these properties for any or all variables. If you do this the Paste button will be un-grayed. If you click on it, SPSS will save the automatically generated syntax to the active syntax editor. The resulting syntax can look a bit unwieldy, especially if you have made amendments to several variables: commands are repeated for each variable, but at least it's accurate. If I ever do this I always edit the result back to something much shorter and succinct.

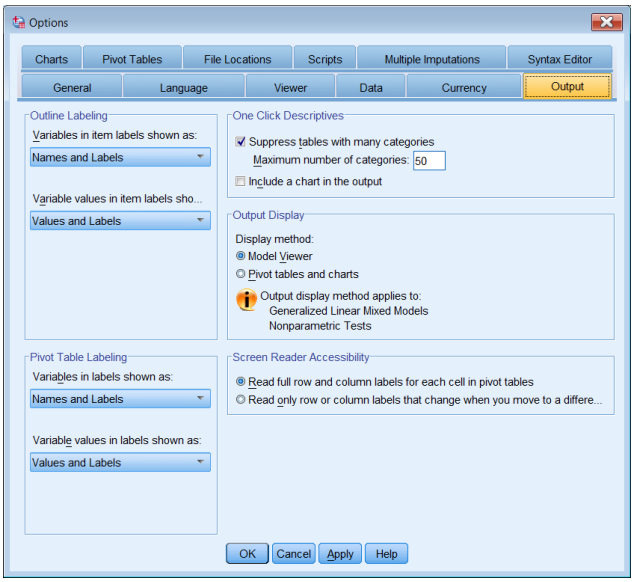
File **gss08.sav** has already been finalised, so no modifications are needed at this stage. However I prefer working with mixed case text, so for aesthetic reasons (for tables and charts in reports) it might be worth changing all the labels. There's a crafty way of doing this with variable labels using copy/pasting to and from Word (changing all letters to lower case) and a nifty Python program by Jon Peck (Senior Software Engineer, IBM/SPSS) to restore first letters to upper case. Changing the value labels is more complicated, but Jon has very kindly supplied yet more Python code for the reverse case setting to change all upper case labels to lower case. I can then run his earlier Python code to restore all first letters of al labels to upper case.

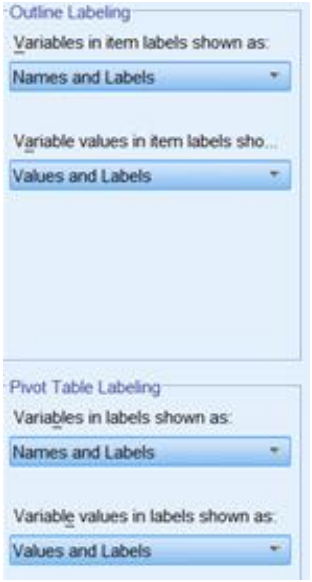
Another thing I tend to do is to make sure that, in output and pivot tables, variables are printed with names and labels and values printed with values and labels.

Edit > Options



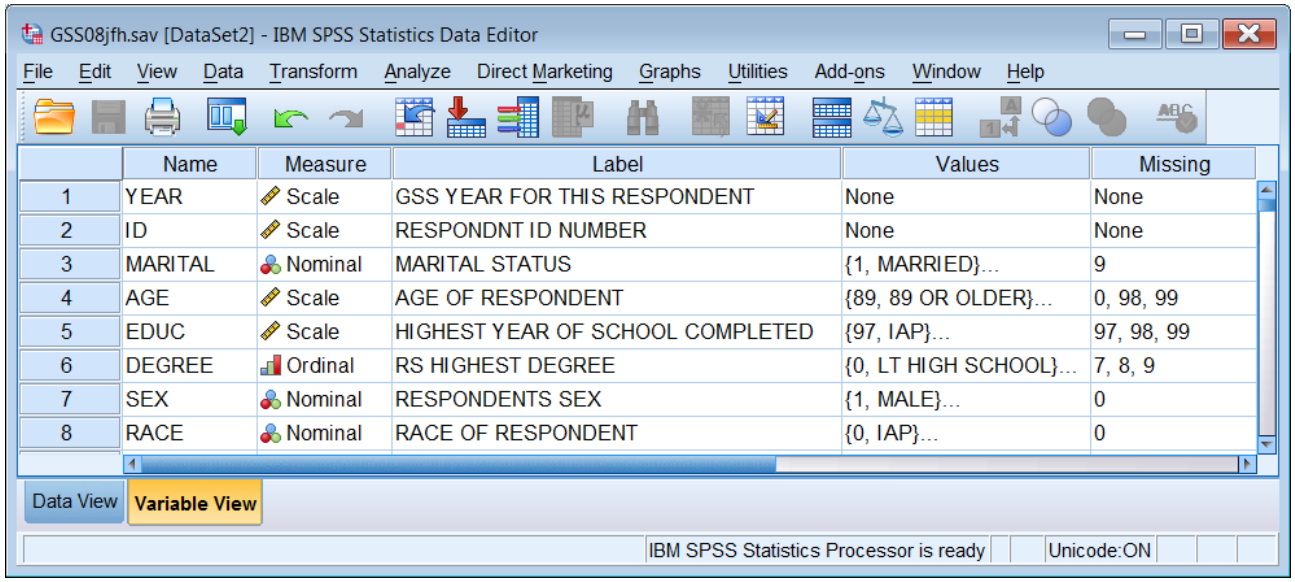

In the Output tab make sure you have Names and Labels and Values and Labels for both Output Labelling and Pivot Table Labelling.



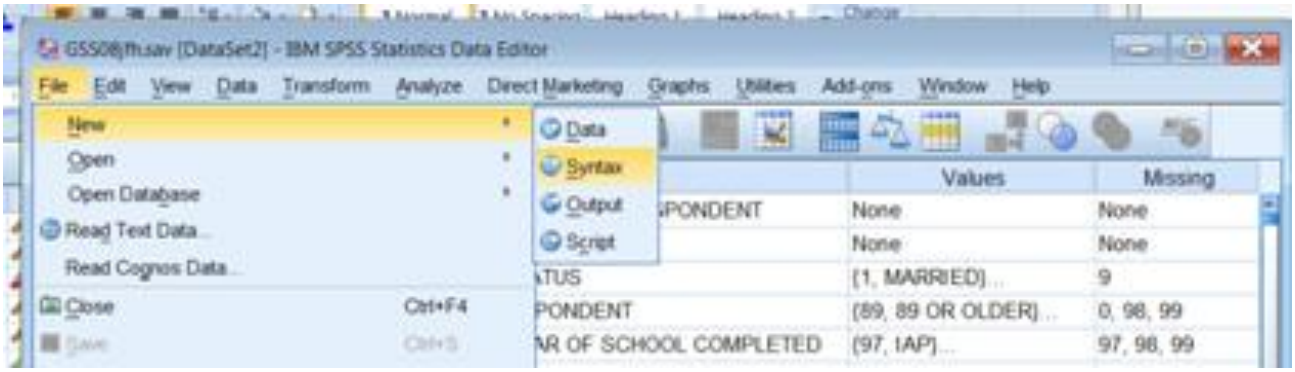
## Initial analysis

Never change an original SPSS file downloaded from an archive or other repository! Normally you can't anyway as they'll be **Read Only**. Always make a copy and do your initial analyses and modifications on that. File **gss08jfh.sav** below has been copied from the original file **gss08.sav** used in the previous examples. This way, if I make a mistake I can always go back to the original file and start over.

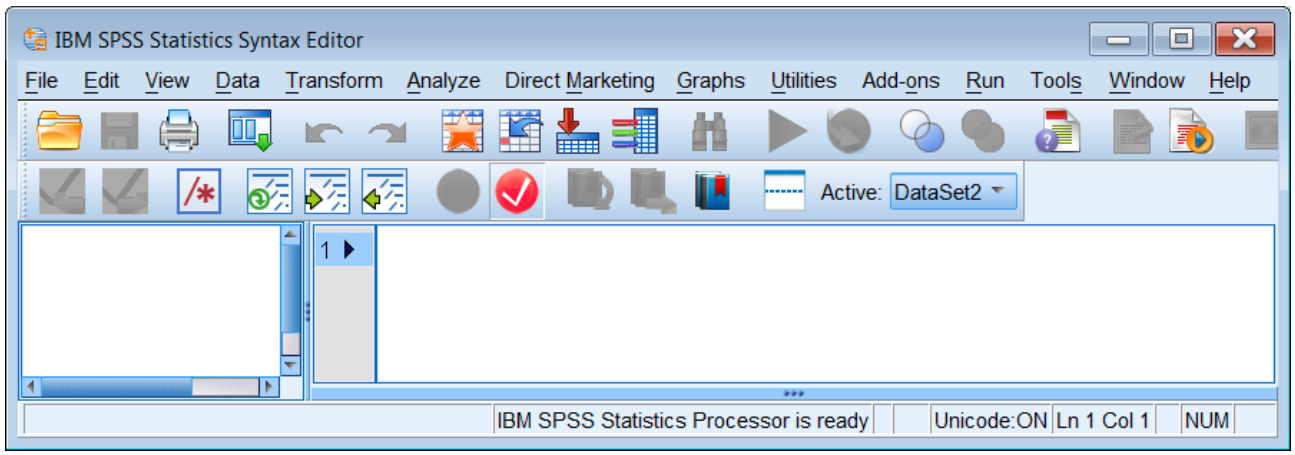



| | Name | Measure | Label | Values | Missing |
|---|---|---|---|---|---|
| 1 | YEAR | Scale | GSS YEAR FOR THIS RESPONDENT | None | None |
| 2 | ID | Scale | RESPONDNT ID NUMBER | None | None |
| 3 | MARITAL | Nominal | MARITAL STATUS | {1, MARRIED}... | 9 |
| 4 | AGE | Scale | AGE OF RESPONDENT | {89, 89 OR OLDER}... | 0, 98, 99 |
| 5 | EDUC | Scale | HIGHEST YEAR OF SCHOOL COMPLETED | {97, IAP}... | 97, 98, 99 |
| 6 | DEGREE | Ordinal | RS HIGHEST DEGREE | {0, LT HIGH SCHOOL}... | 7, 8, 9 |
| 7 | SEX | Nominal | RESPONDENTS SEX | {1, MALE}... | 0 |
| 8 | RACE | Nominal | RACE OF RESPONDENT | {0, IAP}... | 0 |

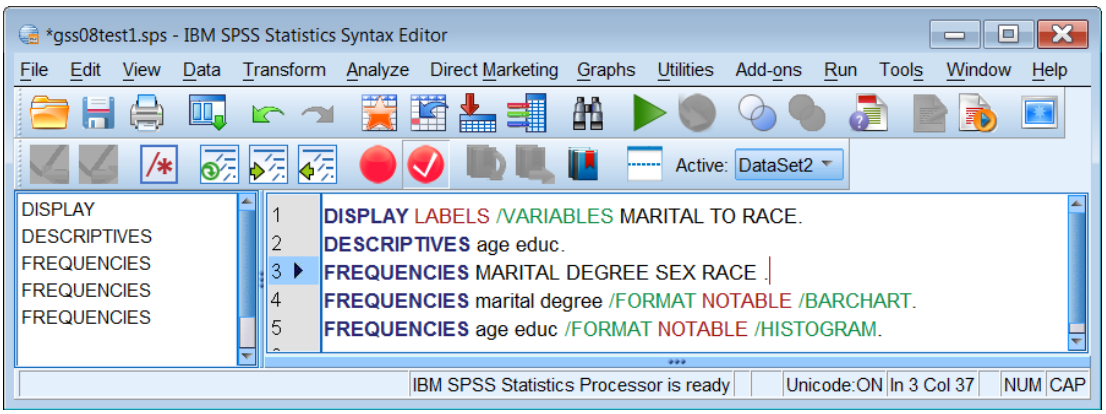
File > New > Syntax




. . opens a new **Syntax Editor**

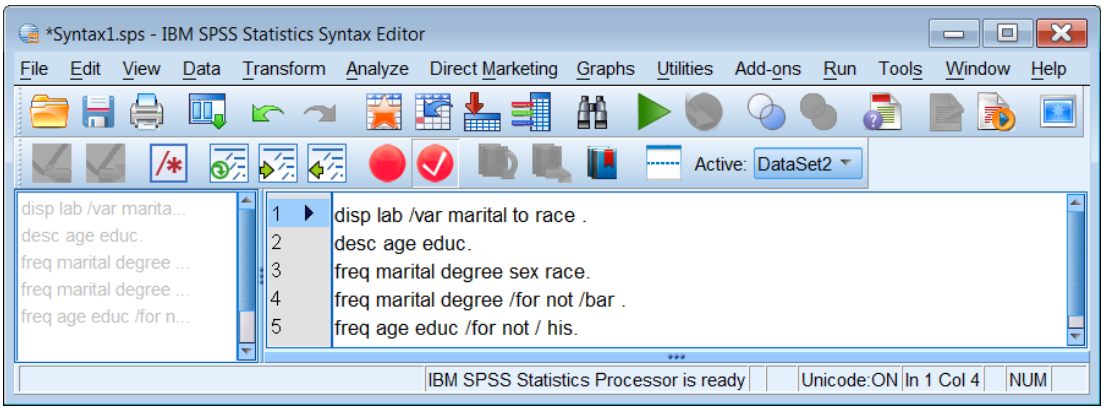


As you type in your commands, SPSS will prompt you with pop-up suggestions (and also for following sub-commands and keywords) If you press Enter for the prompts, SPSS will complete the syntax (in full and in UPPER CASE) using colour coded text, with current commands listed in the left pane. This is suitable for beginners as it's virtually impossible to make errors, but if you mis-spell a variable name SPSS will report an error.




You can then run the analysis in a variety of ways: one command at a time; current command to end; a (highlighted) selection of commands; or all commands at once.

To be totally error-free you need to use the GUI, but this can become quite cumbersome and time-consuming as well as confusing finding the variables you want. I prefer to use abbreviated syntax so no colour-coded text is produced, and the commands will be grayed out in the left pane.




You can produce a list of variable names and labels using the DISPLAY command (not available in the GUI). If you have many variables, it's best to restrict the request to fewer variables.

DISPLAY LABELS /VARIABLES MARITAL TO RACE. [is what SPSS will write as you type]

disp lab /var **marital** to **race** . [is what I type]

Both the above produce the following:

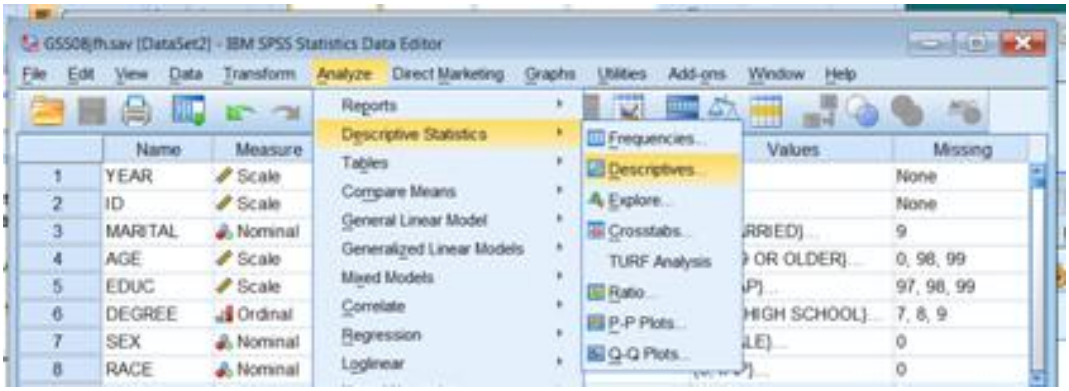
**Variable Labels**

| Variable | Position | Label |
|---|---|---|
| MARITAL | 3 | MARITAL STATUS |
| AGE | 4 | AGE OF RESPONDENT |
| EDUC | 5 | HIGHEST YEAR OF SCHOOL COMPLETED |
| DEGREE | 6 | RS HIGHEST DEGREE |
| SEX | 7 | RESPONDENTS SEX |
| RACE | 8 | RACE OF RESPONDENT |

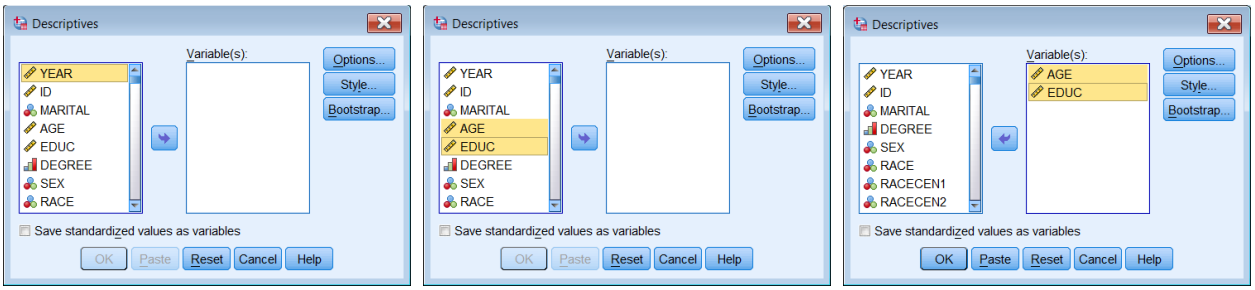
Variables in the working file

You can get a quick idea of the values for each variable from the Define Variable Properties facility outlined earlier, but for **scale** variables you can also get summary statistics, either via the drop-down menus in the GUI or by using direct syntax.

Analyze > Descriptive Statistics > Descriptives




Opening window: Highlight variables in left pane (use Ctrl**+click** for other vars) Click  to transfer to right pane

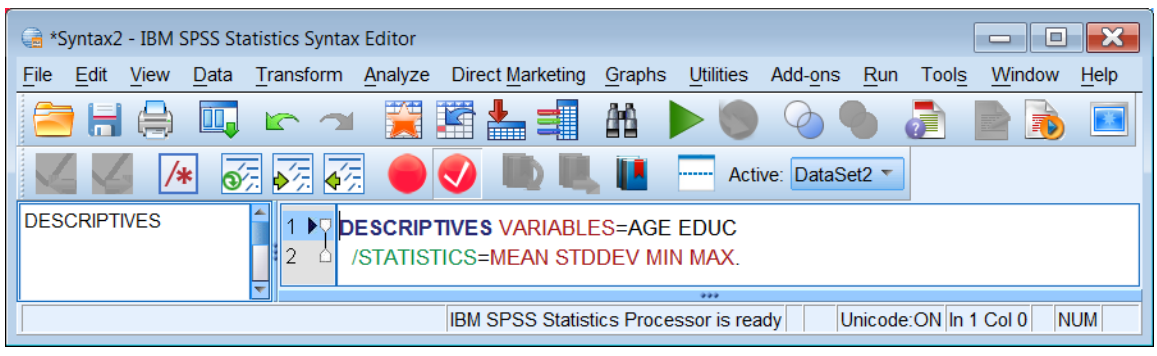



If you click on OK you will get the following table:

**Descriptive Statistics**

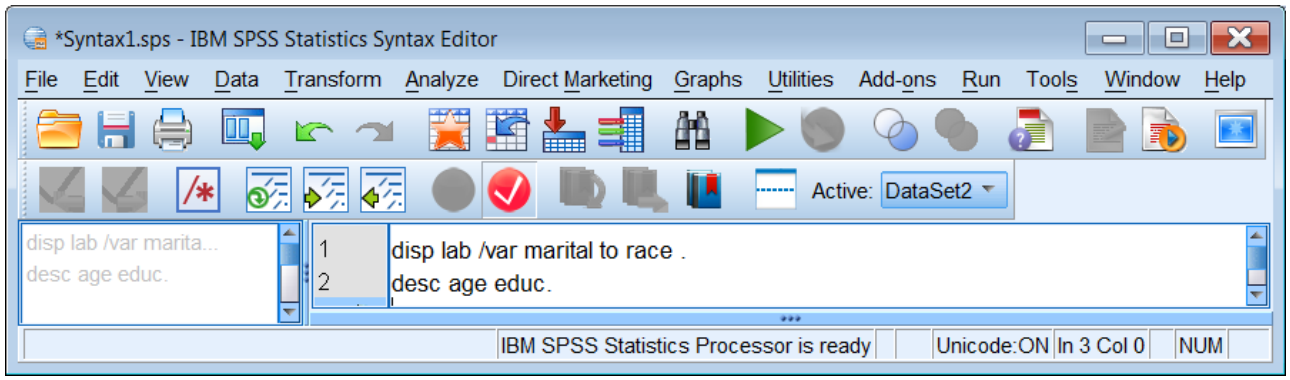
| | N | Minimum | Maximum | Mean | Std. Deviation |
|---|---|---|---|---|---|
| AGE OF RESPONDENT | 2013 | 18 | 89 | 47.71 | 17.351 |
| HIGHEST YEAR OF SCHOOL COMPLETED | 2018 | 0 | 20 | 13.43 | 3.079 |
| Valid N (listwise) | 2008 | | | | |

However, whilst you are busy clicking away on the GUI, SPSS is automatically saving the corresponding syntax. If you click on Paste, SPSS copies the syntax into the active syntax editor:



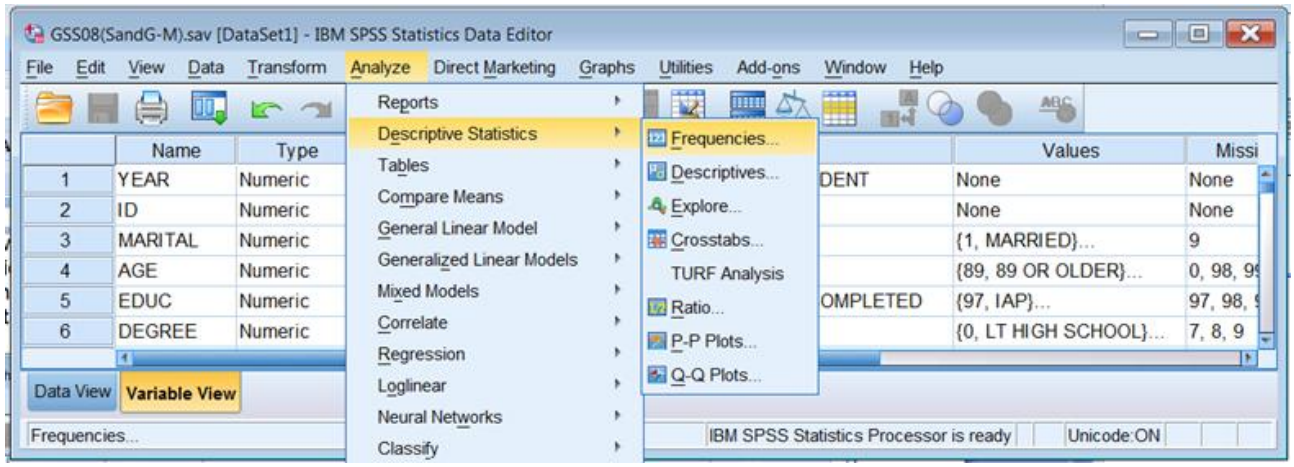

. . but you don't get any tables. This is nice and safe for beginners, but I find it takes too long and the syntax contains much that is superfluous. It's so much quicker to go to the syntax editor and write:

desc age educ.




. . which produces exactly the same table.

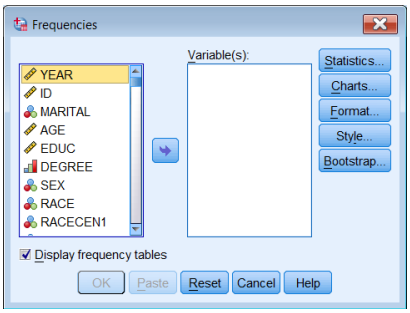
Then I might run off a few frequency counts for nominal and ordinal variables. Doing this for scale variables like **age** is really only useful for finding cutting points for groups: in the old days it was a waste of paper (and trees!)

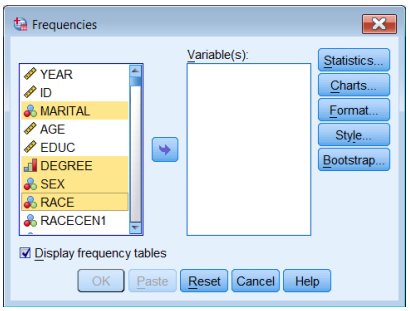


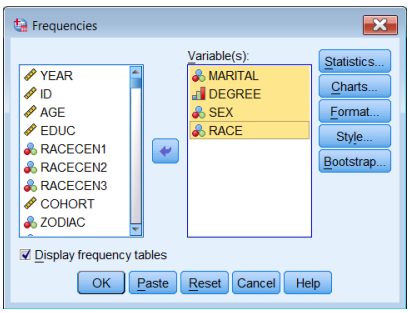

Opening window:

Highlight variables in left pane (use Ctrl**+click** for other vars)

Click  to transfer to right pane




If you click on OK you will get the following tables:

**Statistics**

| | | MARITAL STATUS | RS HIGHEST DEGREE | RESPONDENTS SEX | RACE OF RESPONDENT |
|---|---|---|---|---|---|
| N | Valid | 2018 | 2022 | 2023 | 2023 |
| | Missing | 5 | 1 | 0 | 0 |

**MARITAL STATUS**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | MARRIED | 972 | 48.0 | 48.2 | 48.2 |
| | WIDOWED | 164 | 8.1 | 8.1 | 56.3 |
| | DIVORCED | 281 | 13.9 | 13.9 | 70.2 |
| | SEPARATED | 70 | 3.5 | 3.5 | 73.7 |
| | NEVER MARRIED | 531 | 26.2 | 26.3 | 100.0 |
| | Total | 2018 | 99.8 | 100.0 | |
| Missing | NA | 5 | .2 | | |
| Total | | 2023 | 100.0 | | |

**RS HIGHEST DEGREE**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | LT HIGH SCHOOL | 297 | 14.7 | 14.7 | 14.7 |
| | HIGH SCHOOL | 1003 | 49.6 | 49.6 | 64.3 |
| | JUNIOR COLLEGE | 173 | 8.6 | 8.6 | 72.8 |
| | BACHELOR | 355 | 17.5 | 17.6 | 90.4 |
| | GRADUATE | 194 | 9.6 | 9.6 | 100.0 |
| | Total | 2022 | 100.0 | 100.0 | |
| Missing | DK | 1 | .0 | | |
| Total | | 2023 | 100.0 | | |

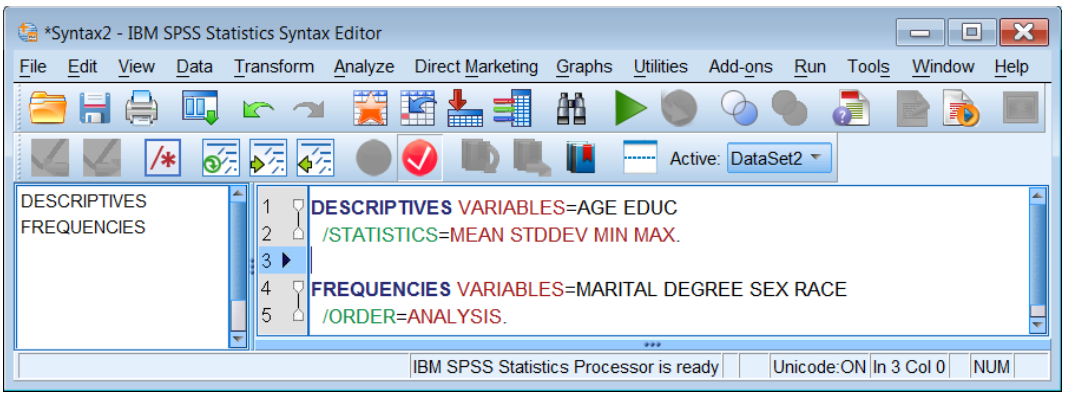
**RESPONDENTS SEX**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | MALE | 929 | 45.9 | 45.9 | 45.9 |
| | FEMALE | 1094 | 54.1 | 54.1 | 100.0 |
| | Total | 2023 | 100.0 | 100.0 | |

**RACE OF RESPONDENT**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | WHITE | 1559 | 77.1 | 77.1 | 77.1 |
| | BLACK | 281 | 13.9 | 13.9 | 91.0 |
| | OTHER | 183 | 9.0 | 9.0 | 100.0 |
| | Total | 2023 | 100.0 | 100.0 | |

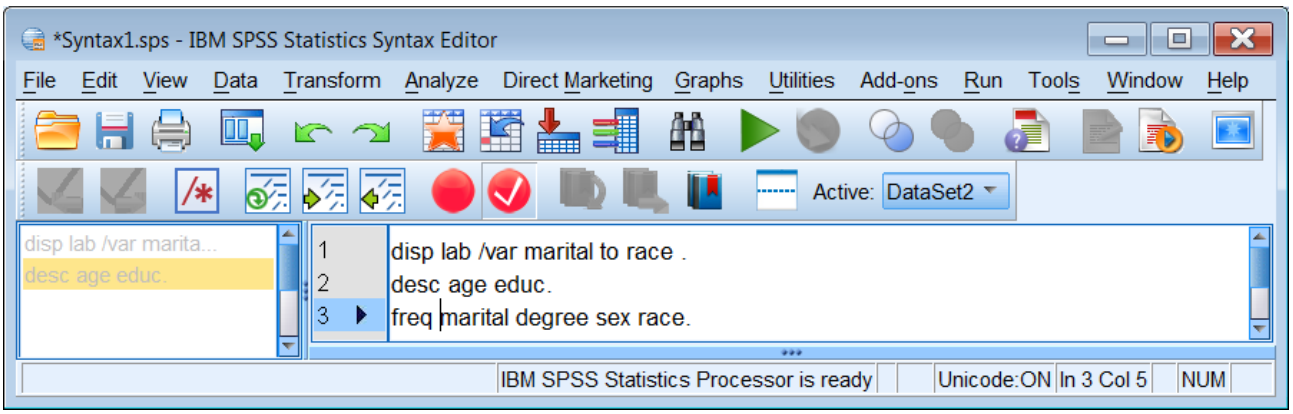
This is the syntax produced in the background by SPSS as you are clicking on the menus, and which Paste sends to the active **Syntax Editor**:




You then have to run this syntax to get your output. I find it so much quicker to work directly in the **Syntax Editor** and type:

freq marital degree sex race.

. . to get the same tables.




No more drop-down menus from now on.

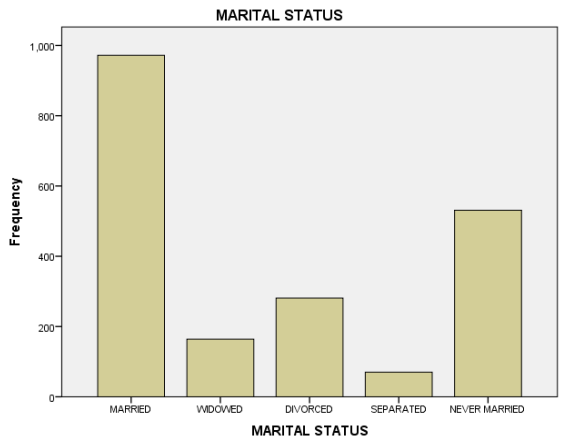
For some variables I might also produce barcharts with the tables suppressed.

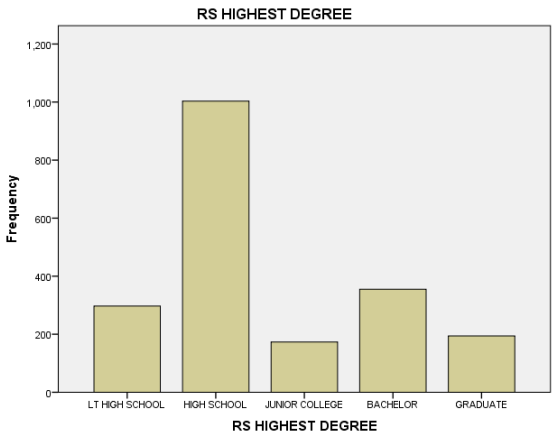
FREQUENCIES VARIABLES MARITAL DEGREE /FORMAT NOTABLE /BARCHART.

freq marital degree /for not /bar .

**Statistics**

| | | MARITAL STATUS | RS HIGHEST DEGREE |
|---|---|---|---|
| N | Valid | 2018 | 2022 |
| | Missing | 5 | 1 |



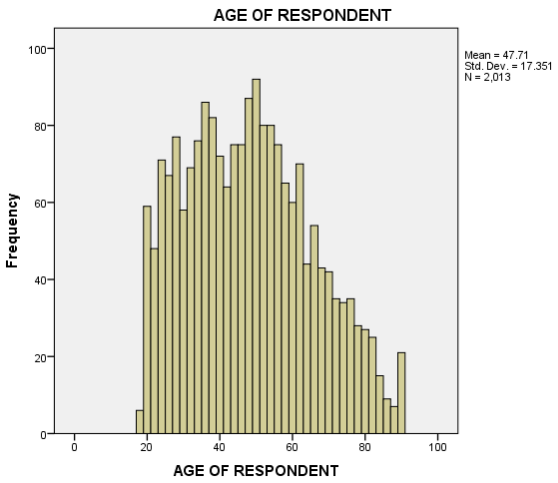

For scale variables with many values I run frequencies with the tables suppressed, but with a histogram request:

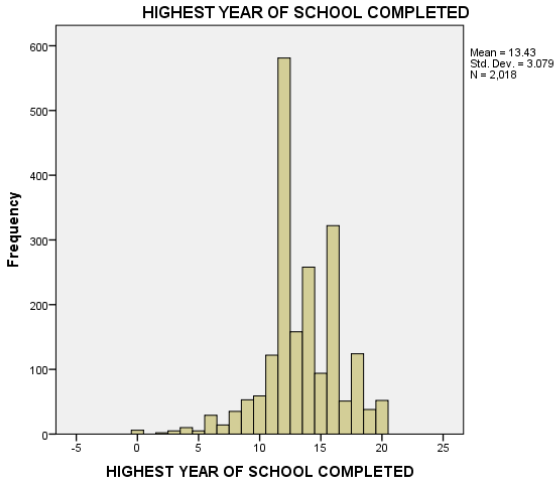
FREQUENCIES age educ /FORMAT NOTABLE /HISTOGRAM.

freq age educ /for not / his.

**Statistics**

| | | AGE OF RESPONDENT | HIGHEST YEAR OF SCHOOL COMPLETED |
|---|---|---|---|
| N | Valid | 2013 | 2018 |
| | Missing | 10 | 5 |




Then I might play around with items in attitude batteries such as the ones dealing with abortion:

| | | | |
|---|---|---|---|
| 47 | ABANY | Nominal | ABORTION IF WOMAN WANTS FOR ANY REASON |
| 48 | ABDEFCTW | Ordinal | WRONG FOR WOMAN TO GET ABORTION FOR BIRTH DE... |
| 49 | ABDEFECT | Nominal | STRONG CHANCE OF SERIOUS DEFECT |
| 50 | ABHLTH | Nominal | WOMANS HEALTH SERIOUSLY ENDANGERED |
| 51 | ABNOMORE | Nominal | MARRIED--WANTS NO MORE CHILDREN |
| 52 | ABPOOR | Ordinal | LOW INCOME--CANT AFFORD MORE CHILDREN |
| 53 | ABPOORW | Nominal | WRONG FOR WOMAN TO GET ABORTION IF LOW INCOME? |
| 54 | ABRAPE | Nominal | PREGNANT AS RESULT OF RAPE |

**Extract from gss08.sav** (subset used by Sweet & Grace-Martin)

[NB: There's a minor error in **gss08.sav**: the levels of **abpoor** and **abpoorw** need to be reversed. This affects analysis further down as we shall see] Six of these items are Yes – No, the other two are 5-point Agree-Disagree. I'd get frequency counts for all 8 items with:

```
freq abany to abrape .
```

I'd also get a summary table of the Yes – No answers using mult response in dichotomous mode:

```
mult resp groups proabort 'Pro-abortion when ...'
        (abany abdefect abhlth abnomore abpoorw abrape (1))
        /freq proabort.
```

**proabort Frequencies**

| | | Responses | | |
|---|---|---|---|---|
| | | N | Percent | Percent of Cases |
| Pro-abortion when ...[a] | ABORTION IF WOMAN WANTS FOR ANY REASON | 550 | 11.3% | 34.3% |
| | STRONG CHANCE OF SERIOUS DEFECT | 952 | 19.6% | 59.3% |
| | WOMANS HEALTH SERIOUSLY ENDANGERED | 1146 | 23.6% | 71.4% |
| | MARRIED--WANTS NO MORE CHILDREN | 579 | 11.9% | 36.1% |
| | WRONG FOR WOMAN TO GET ABORTION IF LOW INCOME? | 652 | 13.4% | 40.6% |
| | PREGNANT AS RESULT OF RAPE | 982 | 20.2% | 61.2% |
| Total | | 4861 | 100.0% | 302.9% |

a. Dichotomy group tabulated at value 1.

Then I'd compute a score by adding up all 6 items, but only for those who answered all six.

```
compute proabort = sum.6 (abany, abdefect, abhlth, abnomore, abpoorw ,abrape).
format proabort (f2.0).
freq proabort.
```

**proabort**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | 1 | 45 | 2.2 | 7.6 | 7.6 |
| | 2 | 46 | 2.3 | 7.7 | 15.3 |
| | 3 | 58 | 2.9 | 9.8 | 25.1 |
| | 4 | 93 | 4.6 | 15.7 | 40.7 |
| | 5 | 35 | 1.7 | 5.9 | 46.6 |
| | 6 | 22 | 1.1 | 3.7 | 50.3 |
| | 7 | 24 | 1.2 | 4.0 | 54.4 |
| | 8 | 47 | 2.3 | 7.9 | 62.3 |
| | 9 | 224 | 11.1 | 37.7 | 100.0 |
| | Total | 594 | 29.4 | 100.0 | |
| Missing | System | 1429 | 70.6 | | |
| Total | | 2023 | 100.0 | | |

Oops! A score of 9 is impossible! Go back and check right variables are used and also missing values. Actually it's the levels that are wrong for **abpoor** and **abpoorw**. This means the mult resp output is wrong as well. Change the levels with:

```
var lev abpoor (nom) abpoorw (ord).
```

and run the job again:

```
mult resp groups proabort 'Pro-abortion when ...'
        (abany abdefect abhlth abnomore abpoor abrape (1))
        /freq proabort.
```

**proabort Frequencies**

| | | Responses | | |
|---|---|---|---|---|
| | | N | Percent | Percent of Cases |
| Pro-abortion when ...[a] | ABORTION IF WOMAN WANTS FOR ANY REASON | 550 | 11.5% | 45.7% |
| | STRONG CHANCE OF SERIOUS DEFECT | 952 | 20.0% | 79.1% |
| | WOMANS HEALTH SERIOUSLY ENDANGERED | 1146 | 24.1% | 95.3% |
| | MARRIED--WANTS NO MORE CHILDREN | 579 | 12.2% | 48.1% |
| | LOW INCOME--CANT AFFORD MORE CHILDREN | 556 | 11.7% | 46.2% |
| | PREGNANT AS RESULT OF RAPE | 982 | 20.6% | 81.6% |
| Total | | 4765 | 100.0% | 396.1% |

a. Dichotomy group tabulated at value 1.

```
compute proabort2 = sum.6 (abany, abdefect, abhlth, abnomore, abpoor ,abrape).
format proabort2 (f2.0).
freq proabort2.
```

**Statistics**

proabort2

| | | |
|---|---|---|
| N | Valid | 1166 |
| | Missing | 857 |

**proabort2**

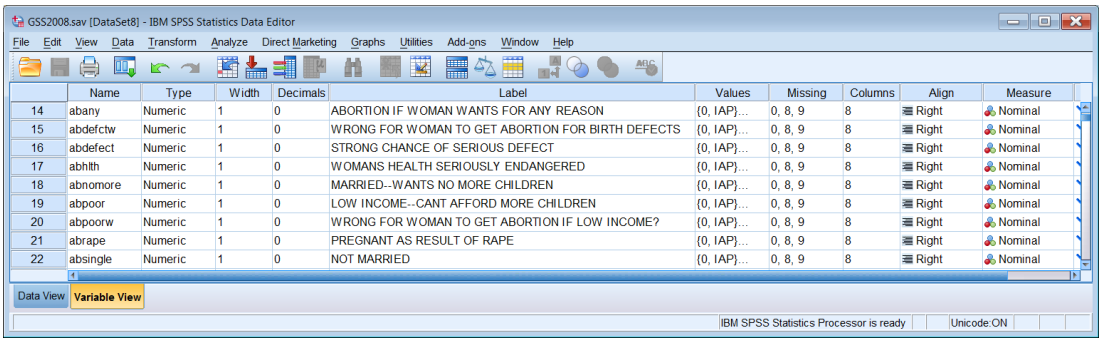
| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | 0 | 100 | 4.9 | 8.6 | 8.6 |
| | 1 | 104 | 5.1 | 8.9 | 17.5 |
| | 2 | 126 | 6.2 | 10.8 | 28.3 |
| | 3 | 224 | 11.1 | 19.2 | 47.5 |
| | 4 | 74 | 3.7 | 6.3 | 53.9 |
| | 5 | 99 | 4.9 | 8.5 | 62.3 |
| | **6** | **439** | **21.7** | **37.7** | 100.0 |
| | Total | 1166 | 57.6 | 100.0 | |
| Missing | System | 857 | 42.4 | | |
| Total | | 2023 | 100.0 | | |

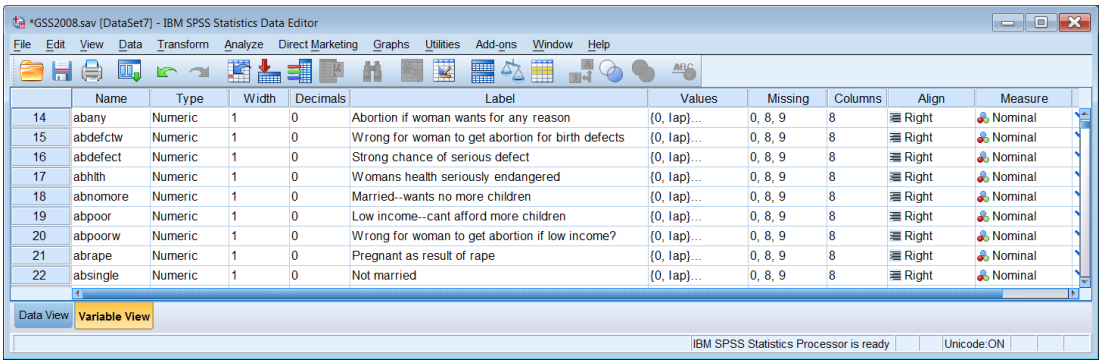
The category 6 is suspiciously large at 37.7%. Is this genuine? Seems to be after checking values and missing values for the constituent items, but I'm still not convinced.

For variables derived by summing across several variables (eg attitude scales) I'd overlay the histogram with a normal distribution. However the more items are summed, the more the distribution of the "score" approaches normal anyway.

## Modifying the file

Changing the case of labels from upper to mixed. Using Jon Peck's original Python code leaves first letter of every word in upper case and also produces strings such as 2Nd and 3Rd , so a second code was run to change all letters to lower case, then a third to change first letters back to upper case. The following uses the full original **GSS2008.sav** file (843 variables).

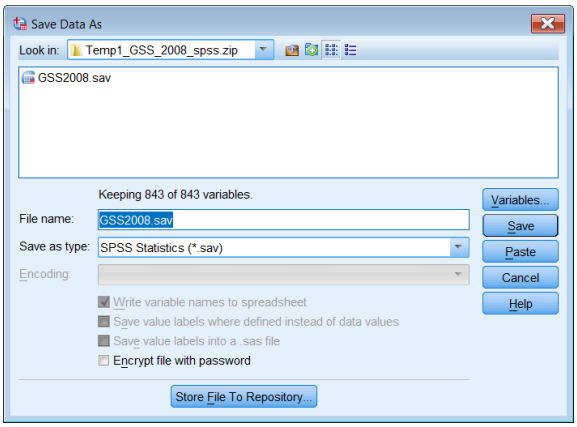


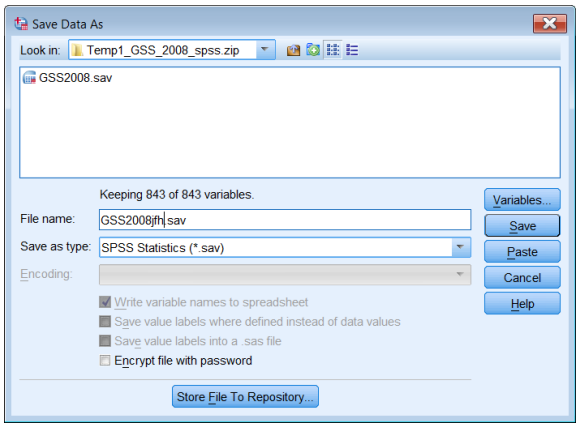




This file was then saved with a new name

File > Save As

Change gss2008.sav to gss2008jfh.sav

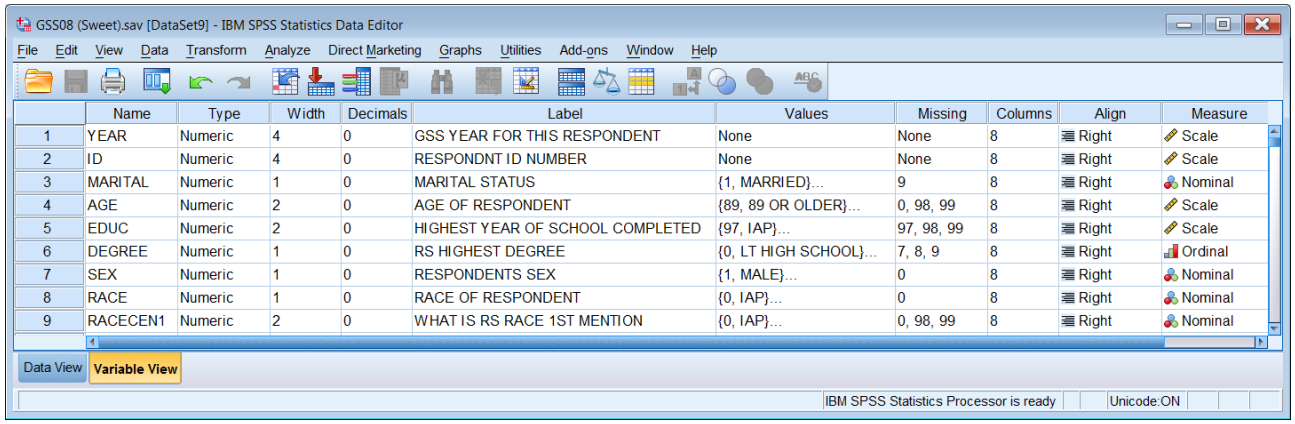


However the Python code changed **IAP** to **Iap** throughout the value labels.

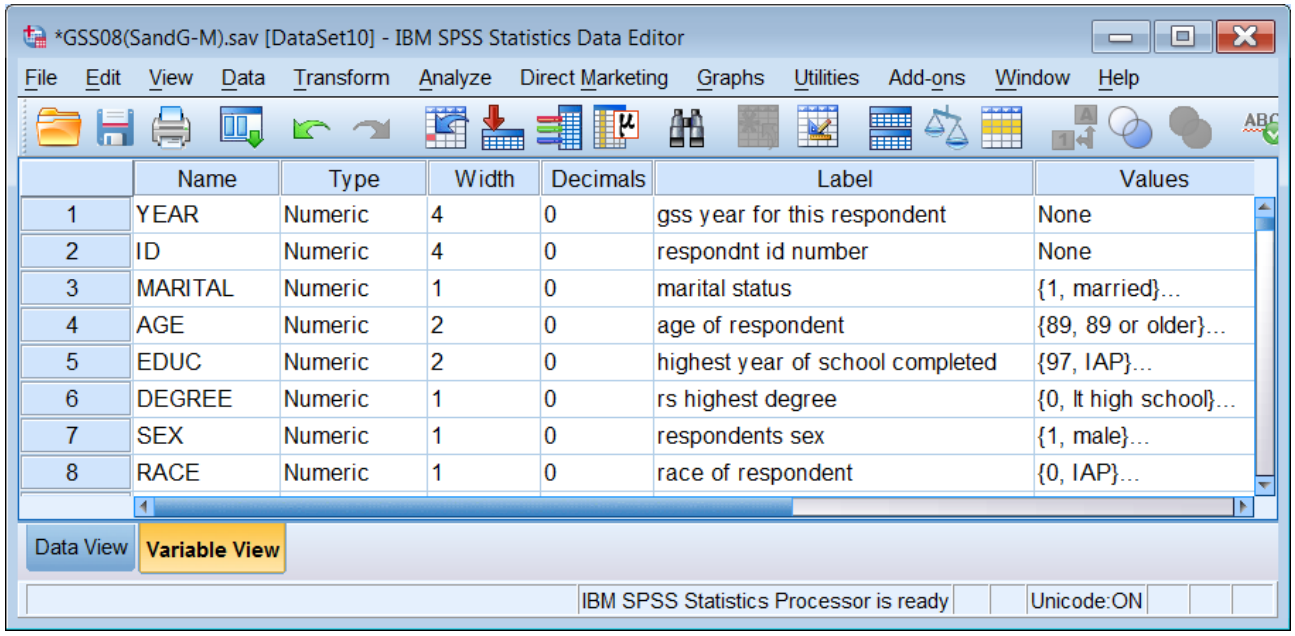
The main file with 843 variables is actually in alphabetical order, not questionnaire order, but the options clearly state file order. Need to check against questionnaire.

The Sweet & Grace-Martin version is in a different order.




The following Python code changes the text in all variable and value labels to lower case.

```
begin program.
import spss, spssaux
vardict = spssaux.VariableDict()
for var in vardict:
    var.VariableLabel = var.VariableLabel.lower()
    vallabels = var.ValueLabels
    for k,v in vallabels.items():
        if not v in ['IAP', 'DK', 'NA']:
            vallabels[k] = v.lower()
    var.ValueLabels = vallabels
end program.
```

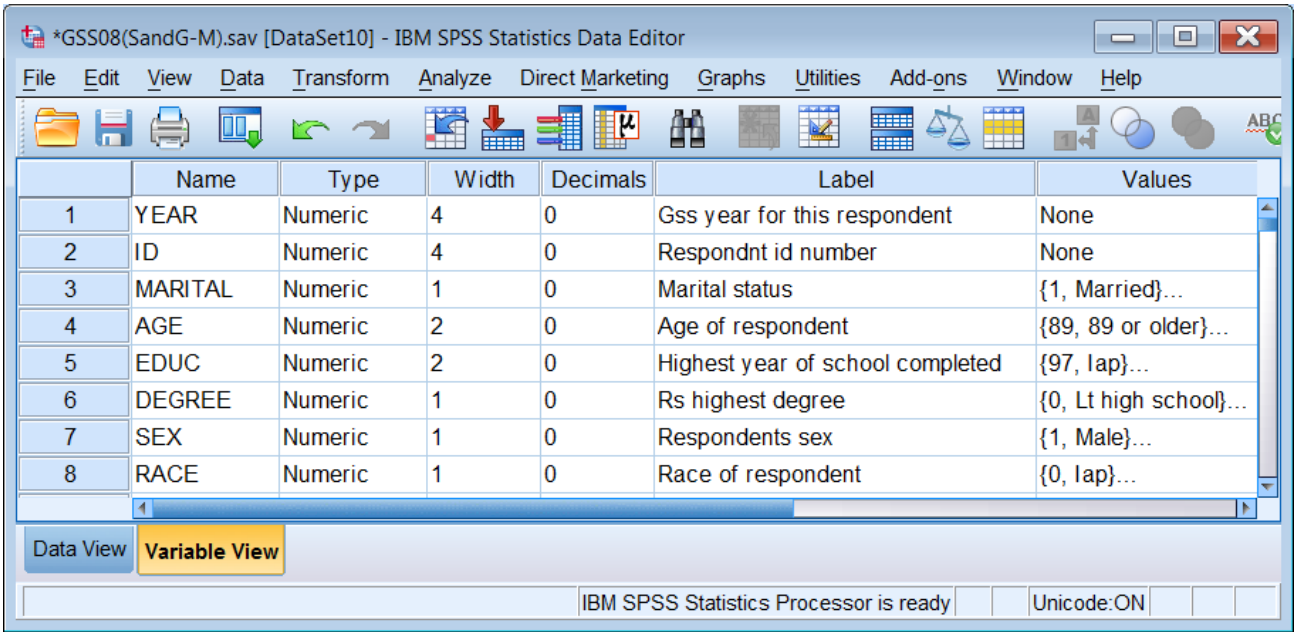



It's fascinating to watch it at work, but don't be put off by any warning messages. These are usually caused by long labels which SPSS simply truncates.

The following Python code restores the first letter of all first words to upper case:

```
begin program.
import spss, spssaux
vd = spssaux.VariableDict()
for v in vd:
    varlabel = v.VariableLabel
    if varlabel:
        v.VariableLabel = varlabel.capitalize()
    vallbls = v.ValueLabels
    for k in vallbls:
        vallbls[k] = vallbls[k].capitalize()
    if vallbls:
        v.ValueLabels = vallbls
end program.
```

. . but this still leaves some standard abbreviations and proper names in lower case.



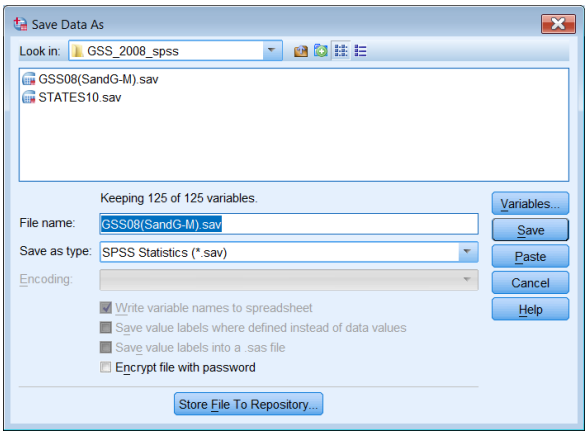

It is perhaps quicker to correct some of these manually, either directly in the **Data Editor** (which leaves a syntax trail) or by highlighting the **Labels** or **Values** column and using Ctrl+H to replace lower case with UPPER case. Ctrl+H is incredibly useful to make the same substitutions throughout either the **Labels** or the **Values** columns. It even makes substitutions in all value labels (only the first value label is displayed in the **Data Editor**). However you have to be careful to enter unique strings, otherwise you may get unwanted substitutions in unsuspected places. If this happens you can always use Ctrl +Z to restore the immediately preceding version. Neither of these methods leaves a syntax trail, so you need to keep a note what you've done.

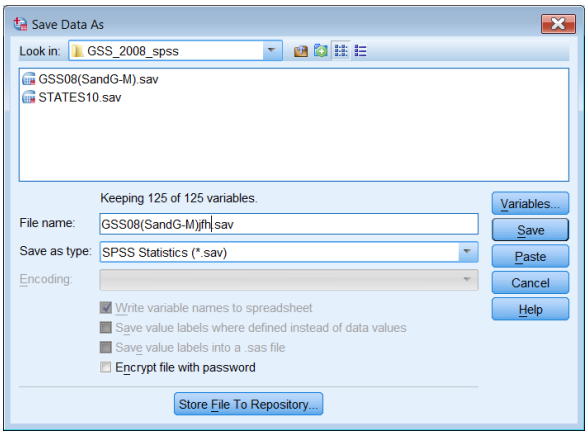
An alternative is to use the Data > Define Variable Properties (DVP) facility. Any alterations to any properties of any or all variables will generate syntax which can be recovered using Paste but you then have to go back to your syntax editor and run the syntax to effect the changes.

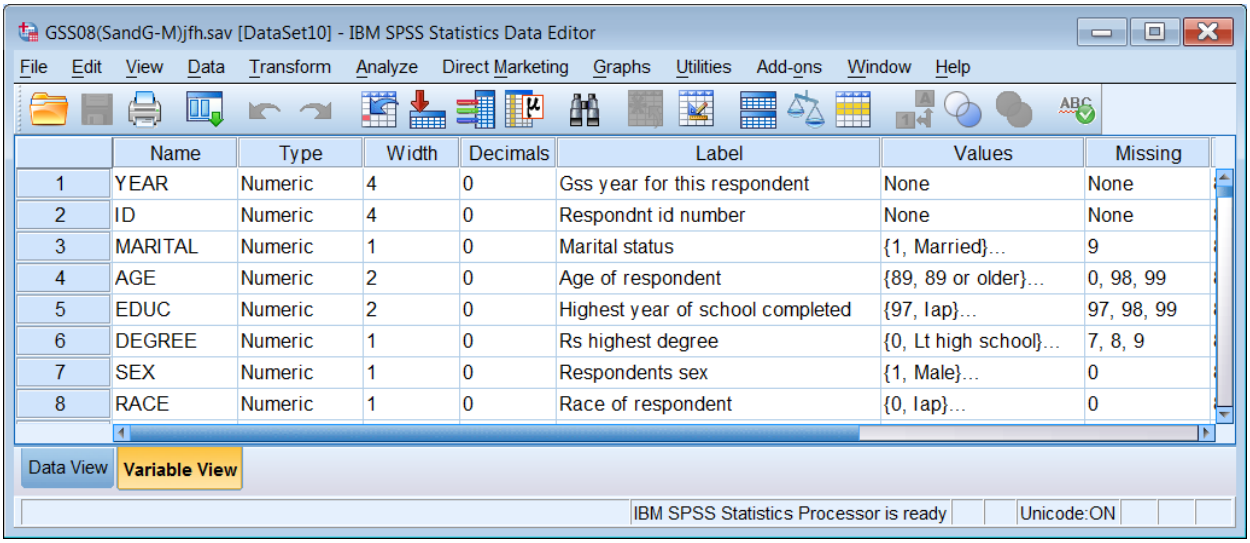
At this point it is good practice to save the modified file with a different name so as to leave the original intact. I frequently tack my initials to the end of the modified filename. [NB: I've switched back to the Sweet and Grace-Martin version for this]

File > Save As

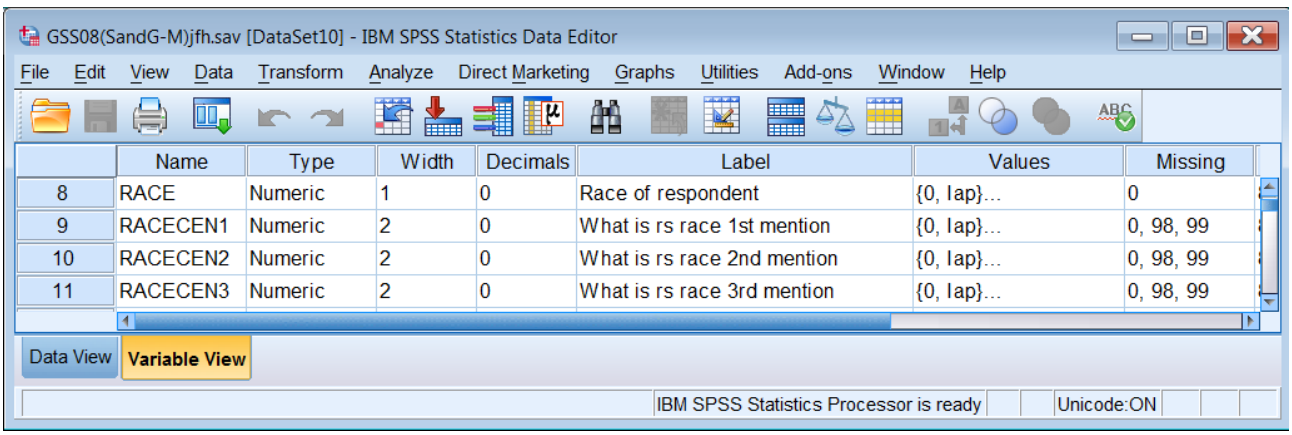
Change gss08(sandg-m) to gss08(sandg-m)jfh



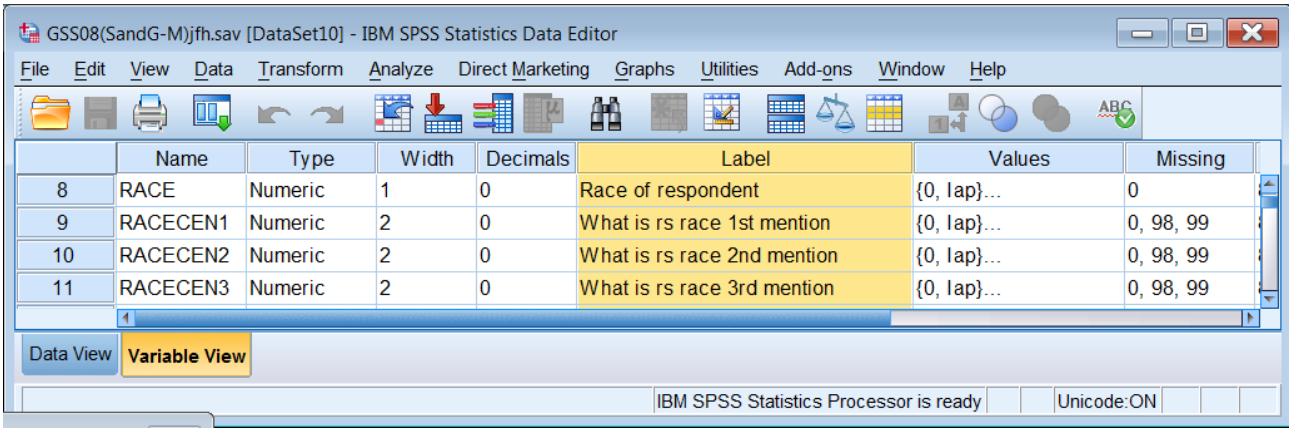



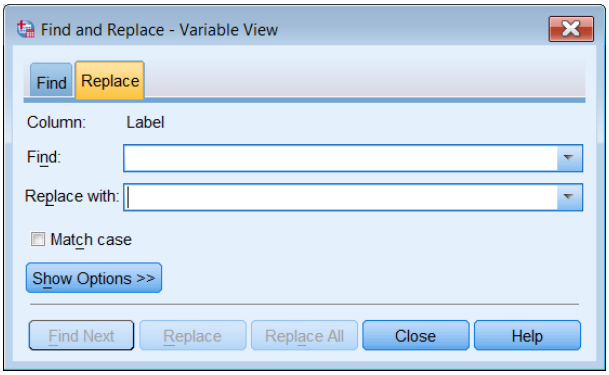

Here are some examples of how to use Ctrl+H to make substitutions. Many of the variable labels in the file contain the string "rs" to indicate "respondent's". Most of the value labels contain the strings Iap Dk and Na which needs changing back to IAP DK and NA:



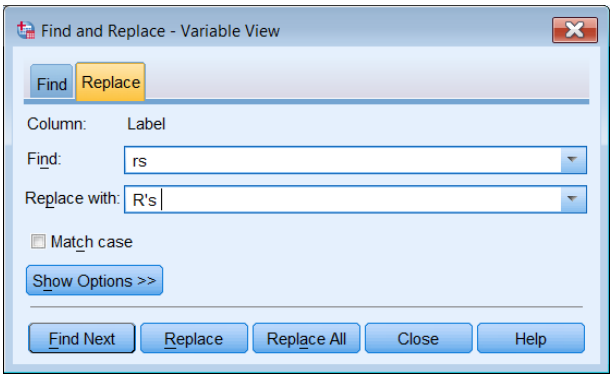
Standard abbreviation for respondent is usually "R". Using Ctrl+H to substitute R's for rs will do this for any string rs, so we need to include the preceding and following space characters □ in both strings. Highlight the **Label** column by clicking on Label



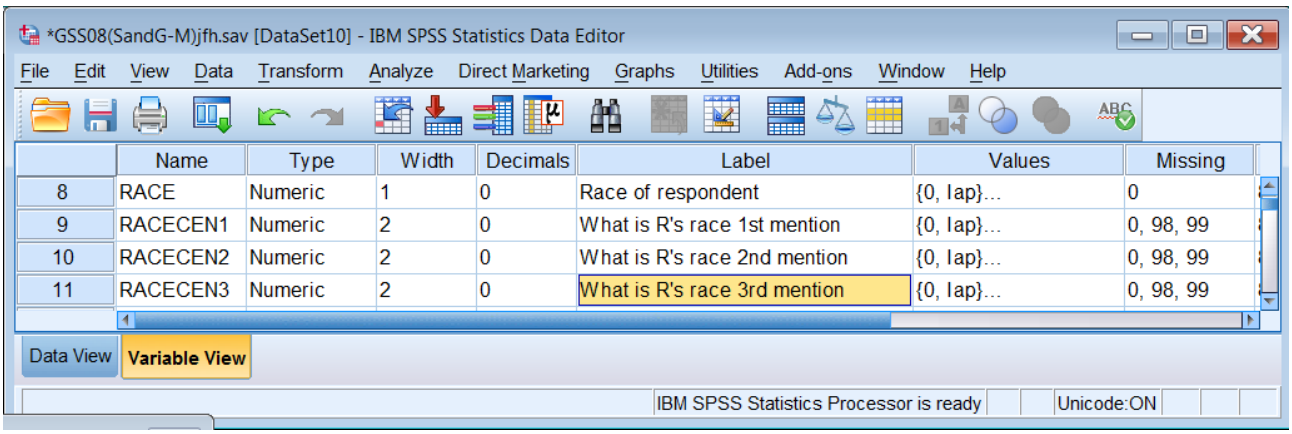

Ctrl+H to open Find and Replace




Type in □rs□ in Find then □R's□ in Replace with




Press Replace All




This still misses a few instances which you have to search for by scrolling around the **Data Editor**, but it certainly saves time and was a pleasant surprise when I played with Ctrl+H and found that it worked not only on variable labels with the Labels column highlighted, but also on (hidden) value labels with the Values column highlighted. The procedure above can be repeated for frequently occurring strings such as Gss, Usa, Sec etc.

However, all this would be much easier in Python code, so I've asked Jon Peck if he can merge the two sets of code.

[to be continued. . .]