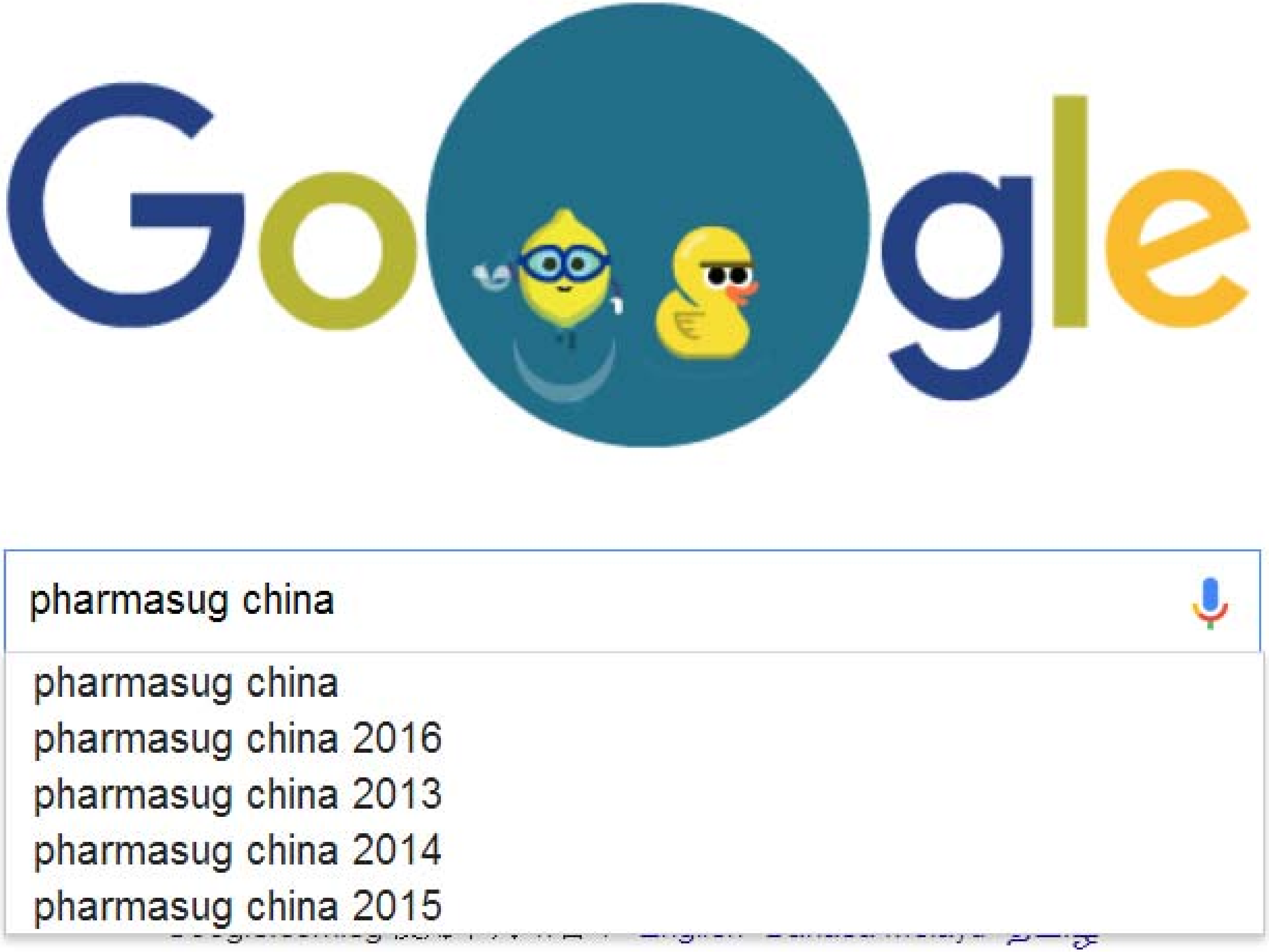
How to give SAS ambiguous instructions and still win big?

(reasonablely delegate everything to SAS)

Hui Liu : Eli Lilly China/ Stats group, Medical Dept

Pharma SUG China 2016, Beijing

- SAS is eager to be your ALPHAGO (the artificial intelligence who defeated the Korean Go game world Champion). The main reason you two are not getting it there is just because you are too nice to SAS. The most valuable, expensive factor in the field of modern programming is the human-time. However, we are wasting much time by doing things that SAS can do itself with less intervention.



pharmasug china

pharmasug china
pharmasug china 2016
pharmasug china 2013
pharmasug china 2014
pharmasug china 2015

organization=pharmasug location=china

- We do not silly enough doing the keyword=value search style in google or baidu.
- But we did do a lot of operations in SAS by giving it information it already knows or manually doing things that can be done by SAS automatically.

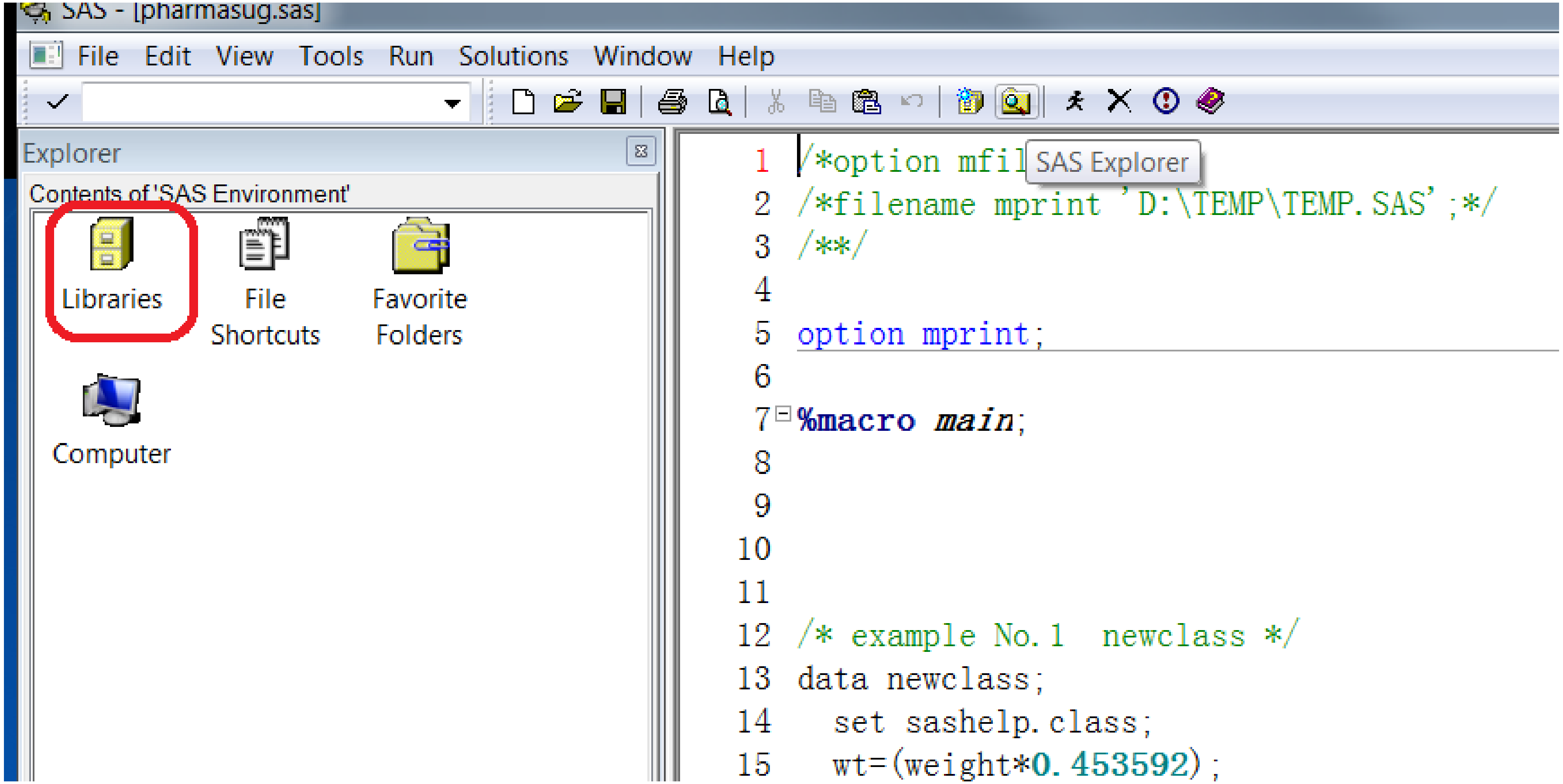
Review/debug SAS code example.

- Let’s see usually what we do when review/debug SAS code.
- The below example code is a piece of code for a simple statistical summary table.

There are a few frequently taken activities when reviewing code

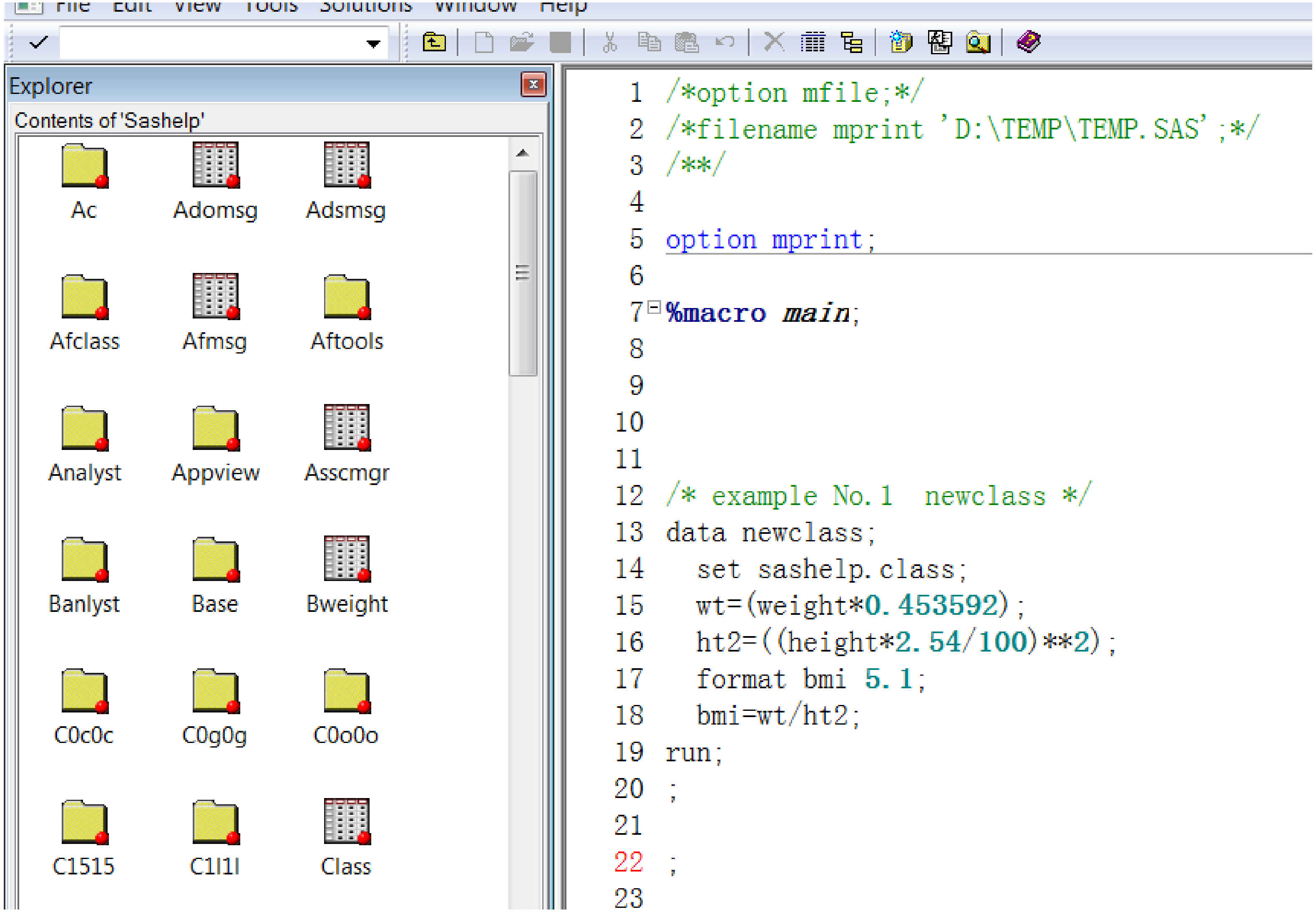
- 1. Open a dataset to review.
- 2. Open a directory to review.
- 3. Open a file to review.
- 4. Open a library location.
- 5. Open a dataset and subset.

Many of the above activities require usually a series steps to perform.



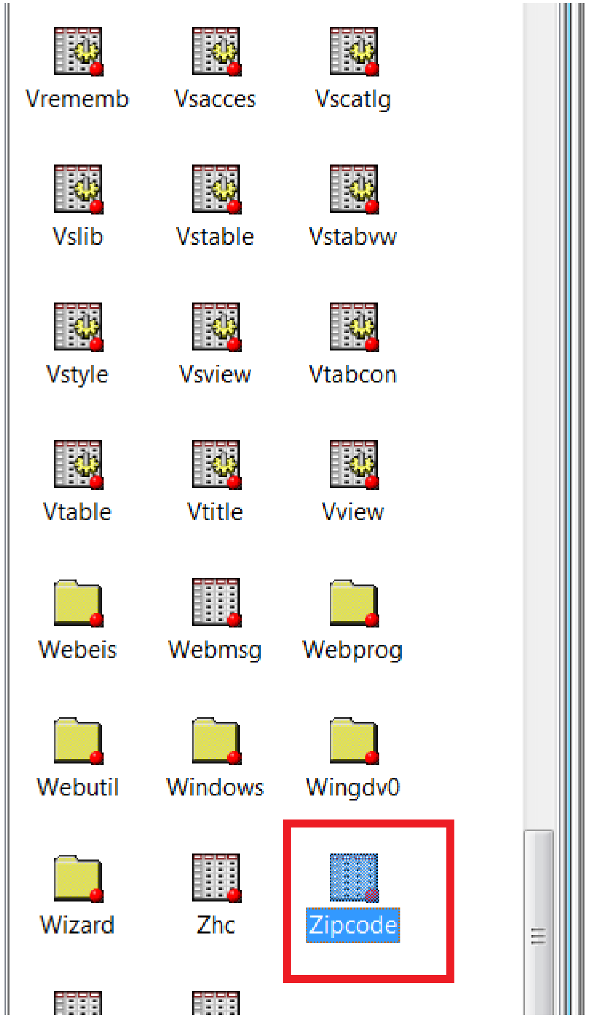
```
/*option mfil
/*filename mprint 'D:\TEMP\TEMP.SAS' ;*/
/**/

option mprint;

%macro main;




/* example No.1  newclass */
data newclass;
  set sashelp.class;
  wt=(weight*0.453592);
```

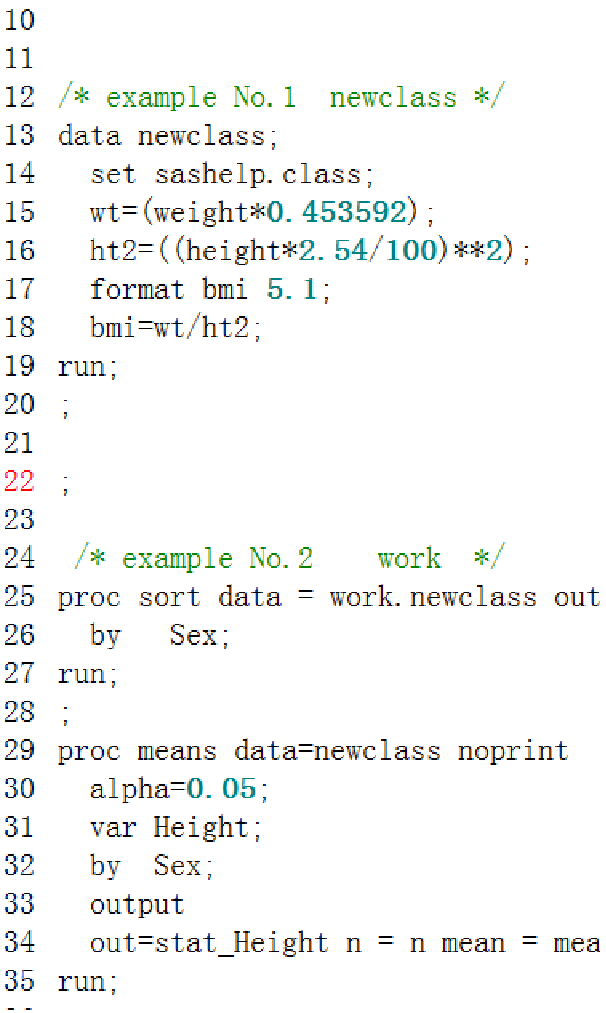
Review by dataset name



```
/*option mfile;*/
/*filename mprint ’D:\TEMP\TEMP.SAS’ ;*/
/**/

option mprint;

%macro main;




/* example No.1  newclass */
data newclass;
  set sashelp.class;
  wt=(weight*0.453592);
  ht2=((height*2.54/100)**2);
  format bmi 5.1;
  bmi=wt/ht2;
run;
;

;

```

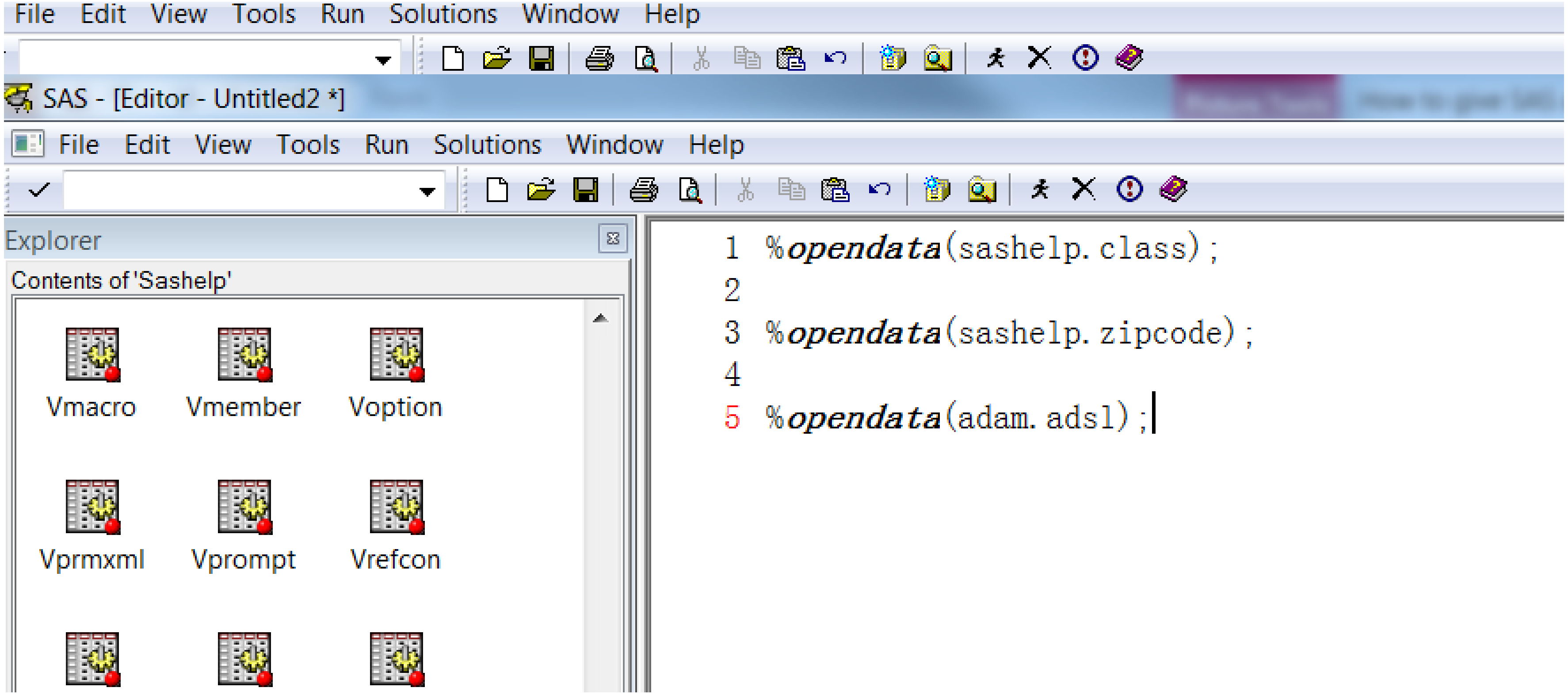
Usually stroll down a few screens.





```


/* example No.1  newclass */
data newclass;
  set sashelp.class;
  wt=(weight*0.453592);
  ht2=((height*2.54/100)**2);
  format bmi 5.1;
  bmi=wt/ht2;
run;
;

;

 /* example No.2    work  */
proc sort data = work.newclass out
  by   Sex;
run;
;
proc means data=newclass noprint
  alpha=0.05;
  var Height;
  by  Sex;
  output
  out=stat_Height n = n mean = mea
run;
```

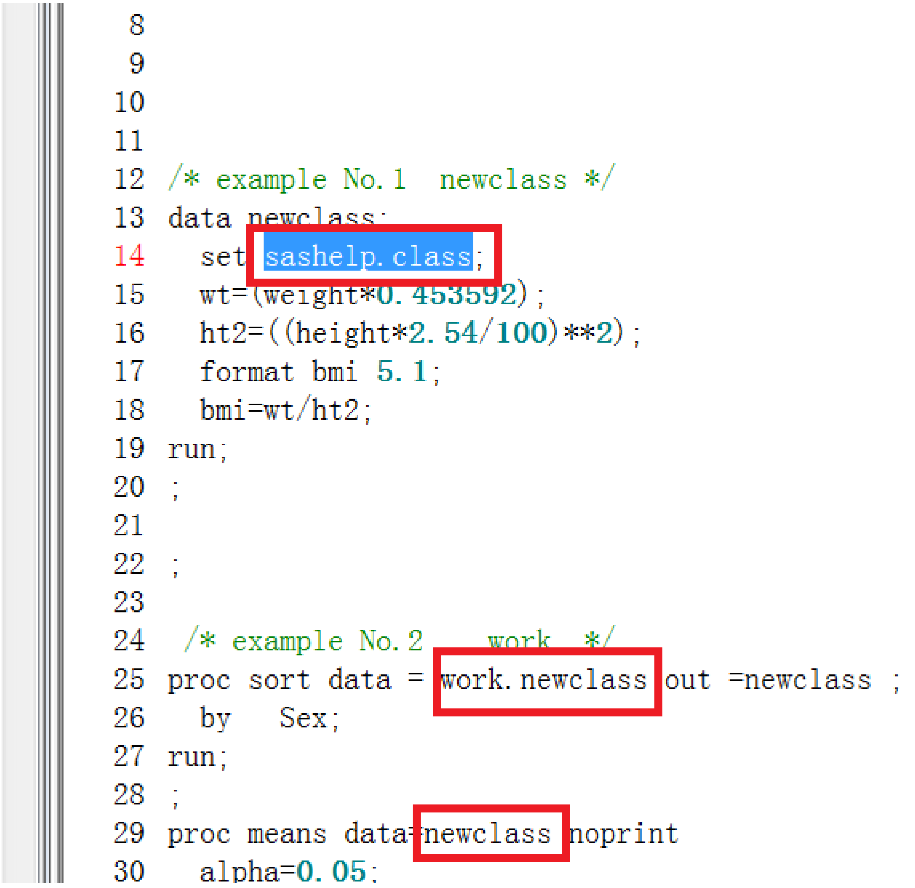
Maybe you can use a macro to open dataset. But still, you need to type the macro repeatedly



```
%opendata(sashelp.class);

%opendata(sashelp.zipcode);

%opendata(adam.adsl);
```

File Edit View Tools Run Solutions Window Help
SAS - [Editor - Untitled2 *]
File Edit View Tools Run Solutions Window Help
Explorer
Contents of 'Sashelp'
Vmacro
Vmember
Voption
Vprmxml
Vprompt
Vrefcon

It is a waste of time not being able to use dataset names in existing code.



```




/* example No.1  newclass */
data newclass;
  set sashelp.class;
  wt=(weight*0.453592);
  ht2=((height*2.54/100)**2);
  format bmi 5.1;
  bmi=wt/ht2;
run;
;

;

 /* example No.2    work  */
proc sort data = work.newclass out =newclass ;
  by   Sex;
run;
;
proc means data=newclass noprint
  alpha=0.05;
```

- The below example is how to use existing “dataset name” “path” “libname” in the SAS code itself to open them.
- General speaking our goals are
  - What should be opened can be opened.
  - Any piece of code can be used and should be used.

Code pieces can be “OPENED”

- C:\temp\
- Sashelp.class
- Sasuser
- Sashelp.class ‘mary’
- Sashelp.class ‘m’
- Sashelp.class =‘m’
- ‘Ja’

Coded/string based actions

- Things can be OPENED very quickly and easily within SAS PGM window.
- Datasets can be not only opened but also subset with ambiguous wildcard.
- A “DATA Step” can be opened.
- A “PROC STEP” can be opened.

Any other strings can also be “OPENED/PROCESSED” if we want. Such as to resolve a sudoku puzzle.

```
7 . . . 3 9 . . 4
5 . 4 6 . . . . .
. . . . . . 8 1 .
. 2 3 . . . . 7 .
4 . . . 6 8 . . .
. . 1 3 2 . . . .
. . 8 . . . 2 5 .
. . . . . 2 7 4 .
. . 6 . 4 3 9 8 .
```

We can try one alive by creating a new matrix together.

$$\begin{matrix}
7 & . & . & . & . & . & . & . & . \\
. & 4 & . & . & . & . & . & . & . \\
. & . & 1 & . & . & . & . & . & . \\
. & . & . & 5 & . & . & . & . & . \\
. & . & . & . & 8 & . & . & . & . \\
. & . & . & . & . & 3 & . & . & . \\
. & . & . & . & . & . & 6 & . & . \\
. & . & . & . & . & . & . & 2 & . \\
. & . & . & . & . & . & . & . & 9
\end{matrix}$$

Things behind.

- There is a virtual engine of SAS code behind of all the actions.
- Anytime a piece of information is sent to the engine the engine will guess/interpret what actions we want it to take and then take the most possible action.
- The actions are mostly what you want and if necessary a second guess is possible.
- More actions can be added in to the engine library without conflict to existing actions.