Pick Your Variables Out
Chen Cao and Xi Chen, University at Albany, NY

ABSTRACT
How could one efficiently subset a large dataset with respect to the variables whose names are 10 digits long and have "V" in the last second position? Here we show a general method when the names of variables of interest share a particular pattern. It consists of two major steps: 1) PROC CONTENTS to store all the variable names as observations of one column in another dataset; 2) PROC SQL to select the names of variables of interest into a macro variable, which serves in a KEEP= option later. This method not only reduces typing or copy-pasting load, but also provides programmers great flexibility.

INTRODUCTION
Sometimes data files contain information that is superfluous to a particular analysis, in which case we might want to create a data file to contain only variables of interest. Programs will run more quickly and occupy less storage space. The main way is to manually type (or copy-paste) variables' names after KEEP= or DROP= options in a data procedure.

If the variable names share a certain pattern, the wildcard characters can make programmers' life easier. For example, we want to select all the variables whose names start with a letter ‘A’.

```sas
data new1;
  set sashelp.class (KEEP=A:);
run;
```

However, when the pattern on the names get complex and the number of variables is large, typing or copy-paste is tedious and thus it is better to make use of SAS not us.

A SIMPLE EXAMPLE
The following simple example demonstrates the general method we propose,

From sashelp.class, now one would like to select variables whose names are six-digit long and ends with the letter “t”. Our idea unfolds below:

First of all, store all the variable names as the observations of one column called “name” in another dataset.

```sas
proc contents data=sashelp.class noprint out=temp(keep=name);
run;
```

Next, create a variable denoting the length of names

```sas
data temp;
  set temp;
  len_varname=length(name);
run;
```

The structure of dataset temp is
Now create a macro variable to store those names with the aforementioned pattern, and separate them by blank for easy use in KEEP= option. Note that TRIM and LEFT functions kill any trailing and leading blanks of each name.

```sql
proc sql noprint;
  select trim(left(name)), count(*)
  into :list separated by ' ', :nlist /*monitor the number of names in this list*/
  from temp
  where name like '%t' and len_varname=6; /*ending with "t" and six-digit long*/
quit;
```

We could ask SAS to show the content of the list explicitly.

```sas
%PUT list contains >&list<;
%PUT number of names= %trim(&nlist);
```

SAS log confirms that the list is ready for use:

```
list contains >Height Weight<
number of names= 2
```

Finally, complete the variable selection.

```sas
data out;
  set sashelp.class (keep=&list);
run;
```

The final dataset containing only variables of interest looks like
Besides, it is worthwhile to mention that '%' can handle other situations. For example, "'e' is contained in the name" corresponds to "WHERE name like '%e%'"; "'H' is the leading characters" can be coded as "WHERE name like 'H%'". One would also consider functions such as UPCASE, SUBSTR and SCAN to meet more demanding needs.

A REAL EXAMPLE

PSID (Panel Study of Income Dynamics) is an annual survey conducted by University of Michigan since 1968 to track demography, economics and health information of around 4800 households. The variable names in family files are initiated by 'v' and followed by numbers. For example, we rename those variables in a subset of 1977 data in a mnemonic way to help further research. The dataset called "fam1977a" contains the following variables

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Food_in</td>
<td>Num</td>
<td>8</td>
<td>Food at home 1975</td>
</tr>
<tr>
<td>4</td>
<td>Food_out</td>
<td>Num</td>
<td>8</td>
<td>Food out 1976</td>
</tr>
<tr>
<td>5</td>
<td>Food_stp</td>
<td>Num</td>
<td>8</td>
<td>Food stamp 1976</td>
</tr>
<tr>
<td>1</td>
<td>ID77</td>
<td>Num</td>
<td>8</td>
<td>1977 ID</td>
</tr>
<tr>
<td>24</td>
<td>den_children</td>
<td>Num</td>
<td>8</td>
<td># of children ??</td>
</tr>
<tr>
<td>20</td>
<td>den_head_age</td>
<td>Num</td>
<td>8</td>
<td>age of head ??</td>
</tr>
<tr>
<td>22</td>
<td>den_head_edu</td>
<td>Num</td>
<td>8</td>
<td>education of head ??</td>
</tr>
<tr>
<td>25</td>
<td>den_head_neworid</td>
<td>Num</td>
<td>8</td>
<td>whether New head ??</td>
</tr>
<tr>
<td>23</td>
<td>den_head_race</td>
<td>Num</td>
<td>8</td>
<td>race of head ??</td>
</tr>
<tr>
<td>21</td>
<td>den_head_sex</td>
<td>Num</td>
<td>8</td>
<td>sex of head ??</td>
</tr>
<tr>
<td>6</td>
<td>den_region</td>
<td>Num</td>
<td>8</td>
<td>current region</td>
</tr>
<tr>
<td>8</td>
<td>den_renter</td>
<td>Num</td>
<td>8</td>
<td>renter</td>
</tr>
<tr>
<td>7</td>
<td>den_snsa</td>
<td>Num</td>
<td>8</td>
<td>SNSA</td>
</tr>
<tr>
<td>12</td>
<td>hse_famsize</td>
<td>Num</td>
<td>8</td>
<td>1977 FAMILY SIZE</td>
</tr>
<tr>
<td>9</td>
<td>hse_rentnum</td>
<td>Num</td>
<td>8</td>
<td>rent number</td>
</tr>
<tr>
<td>10</td>
<td>hse_type</td>
<td>Num</td>
<td>8</td>
<td>Housing type</td>
</tr>
<tr>
<td>11</td>
<td>hse_value</td>
<td>Num</td>
<td>8</td>
<td>House value ??</td>
</tr>
<tr>
<td>27</td>
<td>inc_hlabr</td>
<td>Num</td>
<td>8</td>
<td>Labor income-head 76</td>
</tr>
<tr>
<td>25</td>
<td>inc_nmoney</td>
<td>Num</td>
<td>8</td>
<td>money income 76</td>
</tr>
<tr>
<td>2</td>
<td>inc_mrate</td>
<td>Num</td>
<td>8</td>
<td>Marginal Tax Rate 1976</td>
</tr>
<tr>
<td>15</td>
<td>inc_tax</td>
<td>Num</td>
<td>8</td>
<td>Taxes1976</td>
</tr>
<tr>
<td>14</td>
<td>inc_trasf</td>
<td>Num</td>
<td>8</td>
<td>Transfer income1976</td>
</tr>
<tr>
<td>13</td>
<td>inc_table</td>
<td>Num</td>
<td>8</td>
<td>Taxable Income1976</td>
</tr>
<tr>
<td>28</td>
<td>inc_wlabr</td>
<td>Num</td>
<td>8</td>
<td>labor income-wife 75</td>
</tr>
<tr>
<td>29</td>
<td>work_emply</td>
<td>Num</td>
<td>8</td>
<td>employment status 76</td>
</tr>
<tr>
<td>31</td>
<td>work_quitreason</td>
<td>Num</td>
<td>8</td>
<td>reason of quitting last job 76</td>
</tr>
<tr>
<td>30</td>
<td>work_unemploywe</td>
<td>Num</td>
<td>8</td>
<td>num of unemployment weeks76</td>
</tr>
<tr>
<td>16</td>
<td>workhrs_head</td>
<td>Num</td>
<td>8</td>
<td>AnnualWorkHours_Head76</td>
</tr>
<tr>
<td>17</td>
<td>workhrs_wife</td>
<td>Num</td>
<td>8</td>
<td>AnnualWorkHours_Wife76</td>
</tr>
<tr>
<td>19</td>
<td>workhushrs_head</td>
<td>Num</td>
<td>8</td>
<td>AnnualHouseworkHours_Head77</td>
</tr>
<tr>
<td>18</td>
<td>workhushrs_wife</td>
<td>Num</td>
<td>8</td>
<td>AnnualHouseworkHours_Wife77</td>
</tr>
</tbody>
</table>

We are only interested in ten variables, which are (1) household head group: dem_head_age, dem_head_edu, den_head_race and den_head_sex; (2) household income group: inc_hlabr, inc_wlabr, inc_nmoney, inc_mrate and inc_trasf.

Our method suggests the SAS program below:

```sas
proc contents data=fam1977a noprint out=temp(keep=name);
run;

data temp;
set temp;
len_varname=length(name);
run;

proc sql noprint;
select trim(left(name)), count(*)
```

3
into :list separated by ' ', :nlist
from temp
where (name like 'inc%' and len_varname=9) or (substr(name,5,4)=‘head’ and len_varname=12);
quit;

%put list contains >&list<;
%put number of names= %trim(&nlist)

Now we can check the log and find:

%PUT list contains >&list<;
list contains >dem_head_age dem_head_edu dem_head_rac dem_head_sex inc_hlabr inc_money inc_mrate
inc_trasf inc_txble inc_wlabr<
%PUT number of names= %trim(&nlist);
number of names= 10

The list is desired and safe to use

data fam1977b;
  set fam1977a (keep=&list);
  run;

As we know, there definitely exist alternatives to the conditions in the WHERE statement. For example,
substr(name,5,4)=‘head’ can be substituted by name like ‘dem_head_%’. Moreover, correctly specifying the
conditions will be tricky under some circumstances because it depends on the names of those undesirable variables. Given the narrow scope of present paper, we will not go further on this point.

CONCLUSIONS
The method we proposed is practically valuable because it allows us to set conditions on the length of variable names and some special name patterns, while it might not be trivial to correctly specify the conditions in WHERE statement.

ACKNOWLEDGEMENTS
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at

Chen Cao
Business Administration Building, Room123C
1400 Washington Ave.
Albany, NY 12222
cc971113@albany.edu