



BIOMETRICS INFORMATION

(You're 95% likely to need this information)

PAMPHLET NO. # 24

DATE: February 9, 1990

SUBJECT: Reading WATFILE files into SAS

WATFILE is a useful package for data entry because it is easy to use. But it is annoying to define variable names and formats twice -- once for WATFILE and again for SAS. To avoid this duplication I wrote a SAS program that reads the variable names and formats defined in WATFILE and creates a SAS data step to read in the data. This program is called WATREAD.sas.

A copy of the program is attached to this pamphlet¹. To avoid typing, you may send me a diskette which will be returned bearing a copy of the program.

Use these steps for WATFILE data entry:

1. Sign onto WATFILE. (If you are unfamiliar with WATFILE you may find the HELP command, or the F1 key, useful).
2. Use the DEFINE statement to:
 - i) Name the variables.
 - ii) Indicate whether they are character or numeric.
 - iii) Define their length and, if numeric, the number of decimal places. Decimal values will be truncated if the number of places indicated is less than the number input.and - iv) Optionally, set limits on allowable values for each variable.

3. Type in data.

4. Save a file in WATFILE format by using `SAVE filename` (the file will be stored in DOS without an extension as *filename.*).

5. Start a SAS session by typing in SAS.

6. On the command line type in:

```
INC 'WATREAD.SAS'.
```

```
CHANGE 'inpdata' 'filename' ALL (This step is necessary with the original program.)
```

(you should see 11 occurrences changed).

```
SUBMIT or press F10.
```

7. When WATREAD.sas has finished, the program window will contain the data step for reading in the data. Type in SUBMIT or press F10 to start the process. Some changes you might want to make to the program before submitting are:

- i) Change the data step name to a two-level name so that a permanent dataset is created. Don't forget to add the appropriate LIBNAME statement before the data step. If you put this LIBNAME statement in the AUTOEXEC.SAS file then it is already established when

¹Ron Kozsan has since improved this program by using SAS MACRO's. See System Information pamphlet #25. His program is on pages 3 and 4 while the original is on pages 5 and 6.



you start typing in SAS commands.

- ii) Change the format of some of the variables. For instance:
 - a) If you used missing values such as '.', 'D' or 'M' for numeric data, you will have had to define those variables as character. WATFILE will not let you enter values such as 'D' or 'M' in numeric fields and changes '.' to zero (0)! In this case you will want to delete the '\$' occurring after the name of the variable so that SAS will read it as numeric.
 - b) If you used implicit decimal points in your data you will want to change the format from **w.** to **w.d** (this is SAS format notation, where **w** is the width of the field or total number of digits, and **d** indicates the number of decimal places). The **w.** format does not have to be changed for data with explicit decimal points. Hence, if you have typed in the decimal place you will not need to make this change.
- iii) You may also want to create new variables as functions of the ones you have read in. If so, put the corresponding statements in after the /* End of INPUT statement */ and before the RUN; statement.
- iv) Add titles, labels, your own formats (with PROC FORMAT) and/or assign new formats with a FORMAT statement.
- v) Your other SAS code to analyze the data.

When setting up the file in WATFILE, note the following.

1. Variable names should be 8 digits or less in length for SAS. Since they can be up to 20 characters long in WATFILE, the program will truncate them to the first 8. So, if you do use longer names keep the first 8 characters unique for each variable or workspace field.
2. WATFILE changes values of '.' to zero in numeric fields. Missing values can be left blank and SAS will read them correctly.
3. WATFILE does not allow any alphabetic characters in numeric fields.
4. WATFILE can be set up to check for incorrect values. This is particularly easy to do for numeric variables and one-digit character variables. See Chapter III.1.3 of the WATFILE manual.

Reference:

Wilkinson, Terry. 1987. *WATFILE/Plus data manipulation system: Tutorial and Reference*, 3rd ed., WATCOM Publications Ltd., Waterloo, Ontario.

CONTACT: Wendy Bergerud
387-5676

```

/* Watread.sas */ /* New program as of July, 1995 */

%let watfile=          ;
%let pgmfile=          ;
%let ssdfile=          ;

%window watread color=cyan
#1 @5 'Welcome to the interactive Watread Program.' color=red
#3 @12 ' * Utility to convert Watfile database *' color=blue
#4 @12 ' *          files to SAS data sets          *' color=blue

#8 @9 "Please enter the names of:"
#10 @16 "Watfile data file   :" @39 watfile 40 attr=underline
required=yes color=blue
#12 @16 "SAS program file    :" @39 pgmfile 12 attr=underline
required=yes color=blue
#13 @16 " (to be created)  "
#15 @16 "SAS data set       :" @43 ssdfile 8 attr=underline
required=yes color=blue
#16 @16 " (to be created)  ";

%display watread;

/*==== DATASTEP INITIALIZATION =====*/
data _null_;
  infile "&watfile" missover ;          *<=====INPUT DATA FILE=====;
  file   "&pgmfile" lrecl = 100;       *<=====OUTPUT PGM FILE=====;
  retain first add 0 col 1 inp1-inp70 number;
  length col 4 var $ 8 type $ 1 length 4 inp1-inp69 $ 80 inp70 $ 69;
  array inp {70} $ inp1-inp70;
/*==== READING THE WATFILE COLUMN DEFINITIONS =====*/
  input first $ 1-3 @;                  line + 1;
  if _n_ = 1 then first=line; *<=====START OF FILE=====;
/* if first = '% S' then do; add = 1; return; end; */
  if (substr(first,1,1) = '%') or
      first = 'tit' then do; *<=====WATFILE COMMENTS=====;
    add + 1;
    return;
  end;
  if first = 'def' then do;              *<=====COLUMN DEFINITIONS==;
    input var $ 8-25 digit 30-32 decimal 34 type $ 36;
    if type = 'L' or type = 'R' then type = '$'; **Character vars;
                                          else type=' ';   **Numeric vars;
    if decimal = . then length = digit;
      else length = digit + decimal + 1;
    var = substr(var,1,8); *** Shortening var names to 8 digits;
/*==== FORMING THE LINES OF THE INPUT STATEMENT =====*/
    inp{70} = col||'|' ||var||'|' ||type||'|' ||length||'|'.';
    if line = first + add then do;      **First line of INPUT statement;
      number=4;      inp{number} = '/**/input @'||inp{70}; end;
    else do;          **Other lines of INPUT statement;
      number + 1;    inp{number} = '/**/          @'||inp{70}; end;
    col + length; output; return; end;

```

```

/*==== AT END OF WATFILE COLUMN DEFINITIONS ====*/;
  if first = 'bye' then do;
    col + length;
/*==== FORMING THE FIRST THREE LINES OF PROGRAM ====*/
  inp1 = "Data &ssdfile ; ";          *<====SAS DATA SET NAME=====;
  inp2 = "  infile '&watfile'";      *<====WATFILE TO BE READ=====;
  inp3 = "/**/missover " ||
    "firstobs = " || line+1 || " lrecl = " || col || ";" ;
/*==== OUTPUTTING PROGRAM STATEMENTS ====*/

  put "TITLE1 'Reading in data from &watfile '";
do i = 1 to number;
  put inp{i}; end;
  put ";;; /* End of INPUT statement */   ;;";
  put "run;";
output; stop; end;
else return;
run;

/*==== MOVING PROGRAM TO WINDOW AND MAKING SOME MINOR CHANGES ====*/
dm "inc '&pgmfile'";
dm "change '@          ' '@' all";
dm "change '=          ' '=' all";
dm "change '          ' ' ' all";
dm "change '          ' ' ' all";
dm "change '/**/'      ' ' all";
dm "file '&pgmfile' replace ";
dm "top; zoom on;";
run;

/*****/

```

PROBLEMS FROM BI #23

1. You should get the same answer.
 2. For the second contrast (with coefficients: -7, -2, 3, 3, 3) the correlation coefficient is 0.919 and the SSC is $.919^2(4000) = 3380$. Thus the F-value is $3380/500 = 6.76$. If you reverse the order of the coefficients the correlation coefficient becomes -0.778 with an SSC of 2420. Thus both the value and the sign of the correlation changes implying that the patterns fit are different when the order is reversed. While the correlation for the linear contrast changes sign when the order is reversed, the value and corresponding SSC do not change, implying that the fundamental pattern fit is the same.
 3. The present coefficients: 1 3 5 5 4 3 have a mean of $21/6$.
So multiply by 2 (to avoid fractions):
to get coefficients: 2 6 10 10 8 6 with a mean of 7.
Then subtract the mean of 7 so that
the final coefficients: -5 -1 3 3 1 -1 have a mean of 0.
-

```

/* Watread.sas */ /* Original program */

/*****
/*          Creating a SAS data step from a Watfile          */
*****/

/* This program requires three changes:

1.  Change 'inpdata.'      to the name of your WATFILE file.
2.  Change 'inpdata.inp'   to a name for your output program
                                file.
3.  Change 'inpdata'      to a name for the SAS dataset you
                                want to create.

-- the SAS dataset name can be either temporary (one-level
name) or permanent (a two-level name).

Note: if the names are all the same (and you did not use an
extension when saving the file in WATFILE) then use:

        Change 'inpdata' 'example' all

on the command line to make the changes.          */

*****/

/*==== DATASTEP INITIALIZATION =====*/
data _null_;
  infile 'inpdata.' misover ;          *<=====INPUT DATA FILE=====;
  file  'inpdata.inp' ;                *<=====OUTPUT PGM FILE=====;
  retain first add 2 col 1 inp1-inp70;
  length col 4 var $ 8 type $ 1 length 4 inp1-inp69 $ 80 inp70 $ 69;
  array inp {70} $ inp1-inp70;
/*==== READING THE WATFILE COLUMN DEFINITIONS =====*/
  input first $ 1-3 @;                  line + 1;
  if first = '%WA' then do; frst=line; *<=====START OF FILE=====;
  line + 1; return; end;
  if (substr(first,1,1) = '%') then do; *<=====WATFILE COMMENTS=====;
  add + 1; return; end;
  if first = 'def' then do;              *<=====COLUMN DEFINITIONS==;
  input var $ 8-25 digit 30-32 decimal 34 type $ 36;
  if type = 'L' or type = 'R' then type = '$'; **Character vars;
  else type=' '; **Numeric vars;
  if decimal = . then length = digit;
  else length = digit + decimal + 1;
  var = substr(var,1,8); *** Shortening var names to 8 digits;
/*==== FORMING THE LINES OF THE INPUT STATEMENT =====*/
  inp{70} = col||' '||var||' '||type||' '||length||'.';
  if line = frst + add then do; **First line of INPUT statement;
  number=4; inp{number} = '/*input @'||inp{70}; end;
  else do; **Other lines of INPUT statement;
  number + 1; inp{number} = '/*/' @'||inp{70}; end;
  col + length; return; end;

```

```

/*==== AT END OF WATFILE COLUMN DEFINITIONS ====*/;
  if first = 'bye' then do;
    col + length;
/*==== FORMING THE FIRST THREE LINES OF PROGRAM ====*/
  inp1 = "Data inptdata ; ";          *<====SAS DATA SET NAME=====;
  inp2 = "  infile 'inptdata.'";      *<====WATFILE TO BE READ=====;
  inp3 = "/**/missover " ||
    "firstobs = " || line || " lrecl = " || col || ";" ;
/*==== OUTPUTTING PROGRAM STATEMENTS ====*/
  do i = 1 to number;
    put # i inp{i}; end;
  put ";;; /* End of INPUT statement */   ;";
  put "TITLE1 'Reading in data from inptdata. '";
  put "run;";
stop; end;
run;
/*==== MOVING PROGRAM TO WINDOW AND MAKING SOME MINOR CHANGES ====*/
dm "inc 'inptdata.inp'";
dm "change ' ' ' ' all";
dm "change '/**/' ' ' all ";
dm "file 'inptdata.inp'";
dm "top; zoom on;";
run;

/*==== SAS INPUT PROGRAM NOW READY TO RUN ====*/

/*****/

```