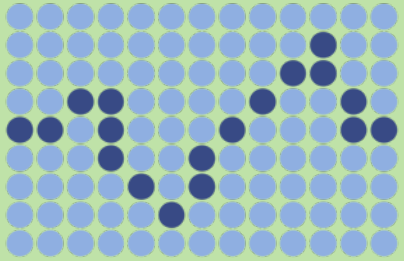


PharmaStat

Perfect Patient Profiles in SAS® using ODS Statistical Graphics

Josh Horstman, MS
PharmaStat
Indianapolis, IN

Richann Watson, MS
DataRich Consulting
Batavia, OH

PharmaSUG 2025
San Diego, CA
June 3, 2025

Overview

- Introduction
  - What are Patient Profiles
  - Design Objectives
- Programming Framework
  - Loading the Data
  - Setting up the Subject List
  - Building the Outputs
- Additional Features
- Conclusion

Introduction

What are Patient Profiles?

- Individual reports for each subject in a clinical study
- Contain all data collected, or a specific subset
- Typically raw data – no summary, derivations or analysis
- Used for data cleaning and study monitoring

Design Objectives

- Input: SDTM data sets
- Output: One PDF file per subject
- Can be rerun iteratively as data is collected
- No hard-coding subject numbers – must be dynamic!
- Include both tabular and graphical output
- Provide output data sets for independent QC
- Use color coding to highlight changes since prior run

Study: CDISCPILOT01
Subject: 01-701-1015 (1015)
Status: COMPLETED

Page 1 of 2
First Dose Date: 2014-01-02
Current Visit: WEEK 26

Demographics

Country	Age (y)	Sex	Race	Ethnicity	Planned Arm	Actual Arm
USA	63	F	WHITE	HISPANIC OR LATINO	Placebo	Placebo

Subject Visits

Visit	Visit Day	Visit Date
SCREENING 1	-7	2013-12-26
SCREENING 2	-1	2013-12-31
BASELINE	1	2014-01-02
AMBUL ECG PLACEMENT	13	2014-01-14
WEEK 2	14	2014-01-16
WEEK 4	28	2014-01-30
AMBUL ECG REMOVAL	30	2014-02-01
WEEK 6	42	2014-02-12
WEEK 8	56	2014-03-05
WEEK 12	84	2014-03-26
WEEK 14 (T)	98	2014-04-09
WEEK 16	112	2014-05-07
WEEK 20	140	2014-05-21
WEEK 22 (T)	154	2014-06-04
WEEK 24	168	2014-06-18
WEEK 26	182	2014-07-02

Adverse Events

Sponsor Defined Identifier and Verbatim Term	Start Date (Day)	End Date (Day)	Ser-ious?	Dur-ation (days)	Sever-ity	Relation-ship	Outcome	Action Taken	Treatment Emergent?
E06: DIARRHOEA	2014-01-09 (8)	2014-01-11 (10)	No	3	Mild	Remote	Recovered/Resolved		Yes
E07: APPLICATION SITE ERYTHEMA	2014-01-03 (2)		No	.	Mild	Probable	Not Recovered/Not Resolved		Yes
E08: APPLICATION SITE PRURITUS	2014-01-03 (2)		No	.	Mild	Probable	Not Recovered/Not Resolved		Yes

FULLY VALIDATED, For Internal Use Only

Basic subject information in titles

Key variables from Demographics domain

List of subject visits and dates

Detail report of adverse events

Orange row is new data since the prior run. Yellow cells have changed since the prior run.

Vital signs in tabular format:
One row per parameter,
Visits transposed to columns

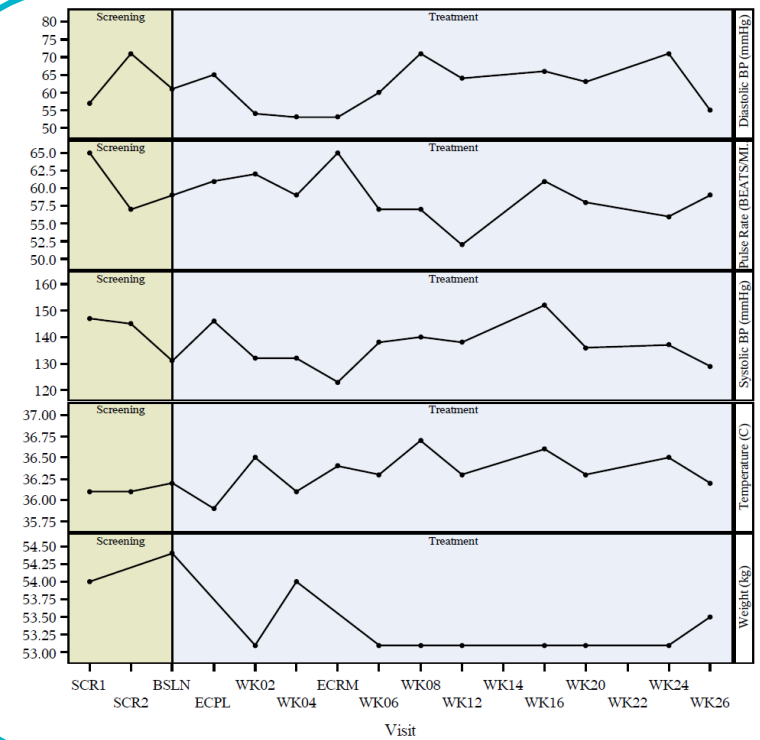
Vital signs in graphical format:
One panel per parameter,
Visits across X axis

Vital Signs

Vital Sign Parameter	SCR1	SCR2	BSLN	ECPL	WK02	WK04	ECRM	WK06	WK08
Diastolic BP (mmHg)	57	71	61	65	54	53	53	60	71
Pulse Rate (BEATS/MIN)	65	57	59	61	62	59	65	57	57
Systolic BP (mmHg)	147	145	131	146	132	132	123	138	140
Temperature (C)	36.1	36.1	36.2	35.9	36.5	36.1	36.4	36.3	36.7
Weight (kg)	54	.	54.4	.	53.1	54	.	53.1	53.1

Vital Signs

WK12	WK16	WK20	WK24	WK26
64	66	63	71	55
52	61	58	56	59
138	152	136	137	129
36.3	36.6	36.3	36.5	36.2
53.1	53.1	53.1	53.1	53.5

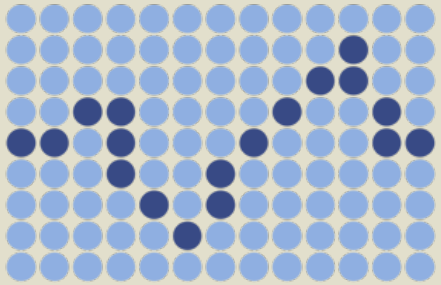


This programming framework allows you to:

- Customize titles and footnotes per subject
- Include desired data elements in any order
- Paginate as desired
- Mix both tabular and graphical output
- Customize colors, fonts, borders, etc. using ODS
- Present data both:
  - Vertically – list individual records (adverse events, medical history, etc.)
  - Horizontally – transposed by visit (labs, vital signs, ECGs, etc.)

Everything is done using the base SAS 9.4 product.

Loading the Data



PharmaStat

The Section List

- The list of SDTM domains to be included is stored in a macro variable:

```
%let sectionlist = DM SV AE VS MH CM QS LB EG;
```

- These domains will appear in the output in the order listed.
- For each domain, we must create two macros:
  - m_getdata_XX (to load the data and perform any manipulations)
  - m_printdata_XX (to generate tabular and/or graphical output)
- Demographics (DM) must be included - it is where we get the list of subjects (more on that later)

The Get Data Loop

```
%do secnum = 1 %to %sysfunc(countw(&sectionlist));
   %let sec = %scan(&sectionlist, &secnum);
   %put === Getting Data for &sec section ===;
   %m_getdata_&sec;
%end;
```

Macro %DO loop executes once for each section in the list.

Get the corresponding section name from the list.

A custom log message makes the execution easier to trace.

Call the macro to load the data.

A Simple Get Data Macro

```
%macro m_getdata_dm;

   data dmfinal;
      set sdtm.dm;
      %if &subj_subset ne %then where &subj_subset;;
   run;

%mend m_getdata_dm;
```

Two semicolons are needed here. One ends the %IF macro statement and the other ends the WHERE statement being generated.

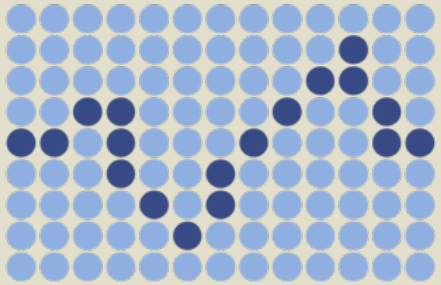
Optional subsetting criteria are placed into another macro variable defined at the top of the main program.

A More Complex Get Data Macro

```
%macro m_getdata_ae;
  /* Omitted: code to merge SDTM.AE and SDTM.SUPPAE */
  data aefinal;
    set aemerge;
    %if &subj_subset ne  %then where &subj_subset;;
    length _aeterm $200 aestart aeend $20 _aeser $3 _aesev $8 _aerel $8
           _aeout $200  aeacn $30 _aetrtem $3;
    _aeterm    = catx(': ', AESPID, AETERM);
    aestart    = catx(' ', AESTDTC, ifc(AESTDY, cats('(', AESTDY, ')'), ''));
    aeend      = catx(' ', AEENDTC, ifc(AEENDY, cats('(', AEENDY, ')'), ''));
    _aeser     = put(AESER, $yn.);
    _aesev     = propcase(AESEV);
    _aerel     = propcase(AEREL);
    _aeout     = propcase(AEOUT);
    _aeacn     = propcase(AEACN);
    _aetrtem   = put(AETRTEM, $yn.);
    if not nmiss(AESTDY, AEENDY) then aedur = AEENDY - AESTDY + 1;
  run;
%mend m_getdata_ae;
```

Custom variables are created to display data as desired.

Setting Up the Subject List



PharmaStat

Creating the Profile List Data Set

```
proc sql;
  create table profilelist as
  select a.USUBJID label = 'Unique Subject ID',
    a.SUBJID label = 'Subject ID',
    /* Additional subject-level fields to appear in header */
  from (select * from dmfinal where ACTARMCD not in ('Scrnfail','')) a
    left join
      /* Join with additional SDTM data sets as needed */
    on a.USUBJID = b.USUBJID;
quit;
```

PROFILELIST data set contains one row per subject and includes subject-level fields that will appear in the profile header. These are typically things like treatment status, date of first dose, study status, date of last visit, etc.

Generating the Table of Contents

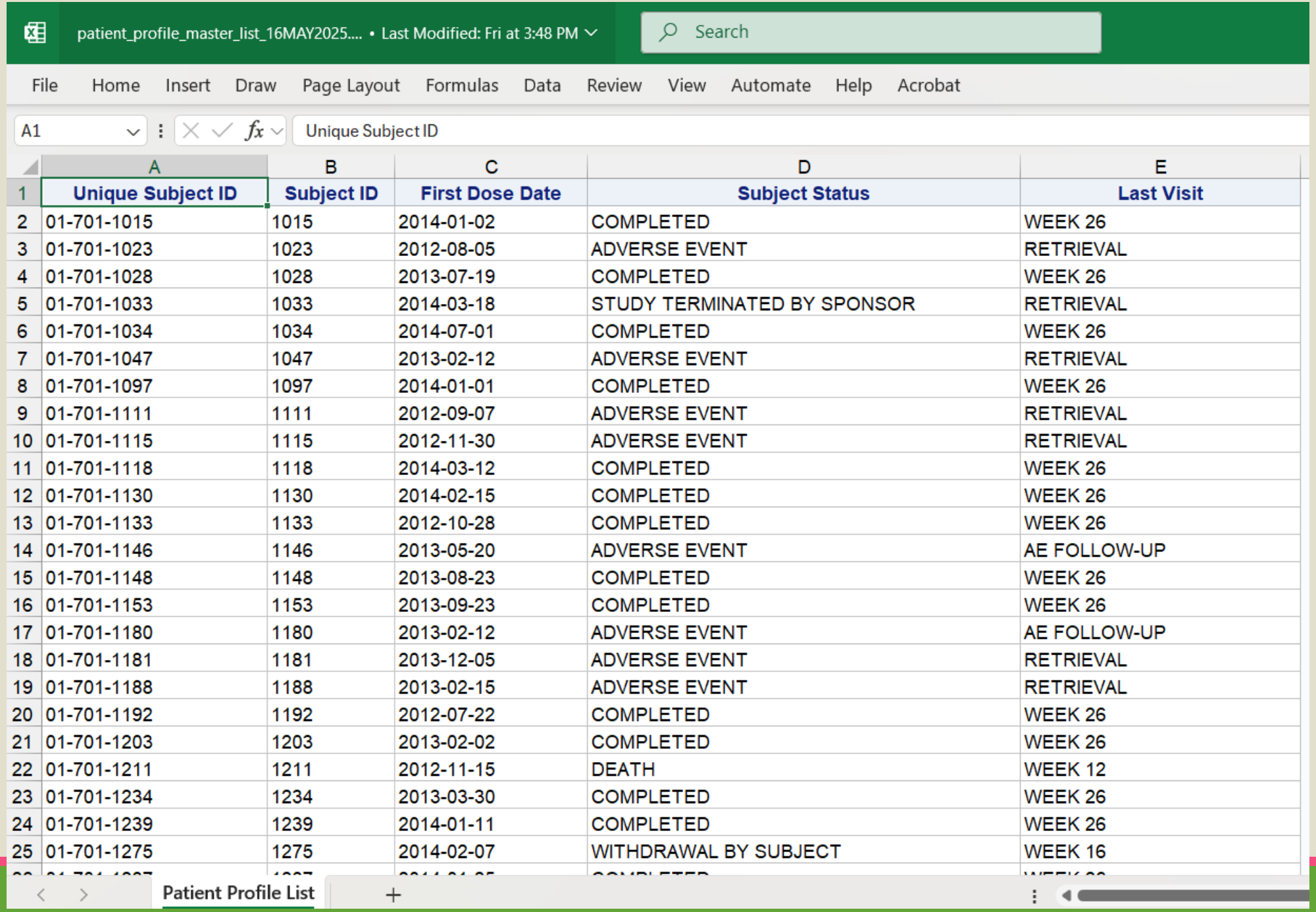
```
ods excel file="&outpath.\patient_profile_master_list_&rundate..xlsx"
          options (sheet_name = "Patient Profile List");

proc report data = profilelist;
   columns usubjid subjid /* Additional subject-level fields */;
run;

ods excel close;
```

Optionally, we can write the contents of the PROFILELIST data set to an external file to serve as a Table of Contents for our patient profiles.

Profile list output as an Excel workbook


patient_profile_master_list_16MAY2025.... • Last Modified: Fri at 3:48 PM

Search

File Home Insert Draw Page Layout Formulas Data Review View Automate Help Acrobat

A1 Unique Subject ID

	A	B	C	D	E
1	Unique Subject ID	Subject ID	First Dose Date	Subject Status	Last Visit
2	01-701-1015	1015	2014-01-02	COMPLETED	WEEK 26
3	01-701-1023	1023	2012-08-05	ADVERSE EVENT	RETRIEVAL
4	01-701-1028	1028	2013-07-19	COMPLETED	WEEK 26
5	01-701-1033	1033	2014-03-18	STUDY TERMINATED BY SPONSOR	RETRIEVAL
6	01-701-1034	1034	2014-07-01	COMPLETED	WEEK 26
7	01-701-1047	1047	2013-02-12	ADVERSE EVENT	RETRIEVAL
8	01-701-1097	1097	2014-01-01	COMPLETED	WEEK 26
9	01-701-1111	1111	2012-09-07	ADVERSE EVENT	RETRIEVAL
10	01-701-1115	1115	2012-11-30	ADVERSE EVENT	RETRIEVAL
11	01-701-1118	1118	2014-03-12	COMPLETED	WEEK 26
12	01-701-1130	1130	2014-02-15	COMPLETED	WEEK 26
13	01-701-1133	1133	2012-10-28	COMPLETED	WEEK 26
14	01-701-1146	1146	2013-05-20	ADVERSE EVENT	AE FOLLOW-UP
15	01-701-1148	1148	2013-08-23	COMPLETED	WEEK 26
16	01-701-1153	1153	2013-09-23	COMPLETED	WEEK 26
17	01-701-1180	1180	2013-02-12	ADVERSE EVENT	AE FOLLOW-UP
18	01-701-1181	1181	2013-12-05	ADVERSE EVENT	RETRIEVAL
19	01-701-1188	1188	2013-02-15	ADVERSE EVENT	RETRIEVAL
20	01-701-1192	1192	2012-07-22	COMPLETED	WEEK 26
21	01-701-1203	1203	2013-02-02	COMPLETED	WEEK 26
22	01-701-1211	1211	2012-11-15	DEATH	WEEK 12
23	01-701-1234	1234	2013-03-30	COMPLETED	WEEK 26
24	01-701-1239	1239	2014-01-11	COMPLETED	WEEK 26
25	01-701-1275	1275	2014-02-07	WITHDRAWAL BY SUBJECT	WEEK 16

Patient Profile List

Building the Macro Variable Lists

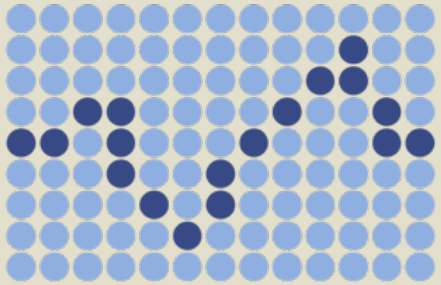
```
proc sql noprint;
   select distinct USUBJID, SUBJID
     into :usubj1 - , :subj1 -
   from profilelist
   order by USUBJID;
   %let subjcount = &sqlobs;
quit;
```

```
%put _user_;

GLOBAL SUBJ1 1015
GLOBAL SUBJ2 1023
GLOBAL SUBJ3 1028
GLOBAL SUBJ4 1033
...
GLOBAL SUBJCOUNT 254
...
GLOBAL USUBJ1 01-701-1015
GLOBAL USUBJ2 01-701-1023
GLOBAL USUBJ3 01-701-1028
GLOBAL USUBJ4 01-701-1033
```

Create two series of macro variables:
USUBJ1, USUBJ2, … contains USUBJID for each subject
SUBJ1, SUBJ2, … contains SUBJID for each subject

Building the Outputs



PharmaStat

The Print Data Loop (Slide 1 of 2)

```
%do i = 1 %to &subjcount;

  proc sql noprint;
    select coalescec(firstdose, 'No Data'),
      subjstatus, lastvis, visabbr
      into
      :firstdose, :subjstatus, :lastvis, :visabbr trimmed
    from profilelist
    where USUBJID = "&&usubj&i";
  quit;
```

The %DO macro loop executes once for each subject.

The header values for the current subject are placed into macro variables.

The Print Data Loop (Slide 2 of 2)

We're still inside the %DO loop!

```
  ods pdf file = "&outpath\profile_&&subj&i.._&visabbr._&rundate..pdf"
                 startpage=no nogtitle nogfootnote;

  title1 j = l "Study: CDISCPILOT01"
         j = r "Subject: &&usubj&i (&&subj&i)";
  /* Additional titles and footnotes as desired */

  %do secnum = 1 %to %sysfunc(countw(&sectionlist));
    %let sec = %scan(&sectionlist, &secnum);
    %put === Printing Data for &sec for Subject &&usubj&i (&&subj&i) ===;
    %m_printdata_&sec;
  %end;
  ods pdf close;
%end;
```

Open a PDF for the current subject.

Inner %DO loop prints each section of the report.

Vertical Tabular Output

```
%macro m_printdata_ae;
  proc sql noprint;
    select count(*) into :numaerecs from aefinal where USUBJID="&&usubj&i";
  quit;


  %if &numaerecs > 0 %then %do;
    proc report data=aefinal split='~' style(report)=[just=left];
      where USUBJID="&&usubj&i";
      columns AESPID ('Adverse Events' _aeterm aedtc aestart aeend
        _aeser aedur _aesev _aerel _aeout _aeacn _aetrtem);
      /* DEFINE statements */
    run;
  %end;
%mend m_printdata_ae;
```

Only print this section of the report if the current subject has data!

PROC REPORT subsets for the current subject and displays the columns we choose.

Horizontal Tabular Output (Slide 1 of 2)

```
%macro m_printdata_vs;

  /* Omitted: code for selecting and formatting data */

 %if &numvsrecs > 0 %then %do;

  /* Omitted: sort data for transposing */

  proc transpose data = vssubset out = vstrans(drop = _:);
    by VSTESTCD;
    var VSSTRESN;
    id vislbl;
  run;
```

Transpose visits to columns. VISLBL is a derived variable containing the desired column labels for each visit.

Horizontal Tabular Output (Slide 2 of 2)

```
  proc sql noprint;
    select distinct NAME, input(NAME, visord.) as sort1
      into :vitalsvislist separated by ' ', :dummy1
    from DICTIONARY.COLUMNS
    where LIBNAME = 'WORK' and MEMNAME = 'VSTRANS' and
      /* Criteria for selecting visit columns */
      order by sort1, sort2;
    %let numvitalsvisits = &sqlobs;
  quit;

  proc report data = vstrans split = '~' style(report) = [just = left];
    columns dummy ('Vital Signs' VSTESTCD &vitalsvislist);
    /* DEFINE statements */
  run;
%end;
%mend m_printdata_vs;
```

We don't want to hardcode the visits, so we place the list of visits into a macro variable.

The list of visits could vary by subject or even by report section!

Graphical Output

Dynamically determine X axis range

```
proc sql noprint;
  select min(xvar), max(xvar) into :xmin, :xmax
  from svfinal
  where VISITNUM = int(VISITNUM) and USUBJID = "&&usubj&i";
quit;

/* Omitted: code for removal of unscheduled visits and block rendering */
proc sgpanel noautolegend data = vsfigure dattrmap = vsattrmap;
  format xvar xvarfmt. VSTESTCD $vsfmt.;
  panelby VSTESTCD / onepanel uniscale = column novarname
                     layout = rowlattice headerattrs = (size = 8pt);
  block x = xvar block = blocklbl / /* Additional formatting options */;
  series x = xvar y = VSSTRESN    / /* Additional formatting options */;
  rowaxis display = (nolabel) thresholdmax=1 thresholdmin=1 offsetmax=0.1;
  colaxis label = 'Visit' values = (&xmin to &xmax by 1) fitpolicy=staggerthin;
run;
```

SGPANEL creates a separate plot for each parameter.

Additional Features

Creating Validation Data Sets (1 of 2)

- During the Get Data loop, we initialize an empty QC data set for each section of the report:

```
data PDATA.PP_&sec;
   length USUBJID $23;
   call missing(USUBJID);
   stop;
run;
```

Creating Validation Data Sets (2 of 2)

- We create an output data set from PROC REPORT each time it is called and append to the QC data set for that report section.

```
proc report data=aefinal split='~' style(report)=[just=left]
                        out=work.qc_ae_&i;
    ...
run;

data PDATA.PP_AE;
  set PDATA.PP_AE
      work.qc_ae_&i(in=currsubj where=(_break_=''));
  if currsubj then USUBJID="&&usubj&i";
  drop dummy _break_;
run;
```

Highlighting New/Changed Data

- To facilitate data review, changes and additions since the previous run can be highlighted in different colors
- During the Get Data loop, an additional macro is called that adds two variables to each data set:
  - newflag – a 0 or 1 flag to indicate whether a record is new
  - modcols – a list of variables that have different values from the prior run
- During the Print Data loop, additional logic is added to each PROC REPORT to modify the attributes based on these variables

Highlighting New/Changed Data

```
%macro m_add_update_vars(dsetin_curr=, dsetin_prev=, dsetout=,
  keyvarlist=, othvarlist= );
  /* Omitted: code to sort input data sets */
  data &dsetout;
    merge &_prefix._curr ( in = incurr)
          &_prefix._prev ( in = inprev keep = &keyvarlist &othvarlist rename = (
             %do _i=1 %to %sysfunc(countw(&othvarlist,%str( )));
                %scan(&othvarlist,&_i) = _%scan(&othvarlist,&_i) %end;));
    by &keyvarlist; if incurr;
    length modcols $200; call missing(modcols);
    if not(inprev) then newflag=1;
    else do;
       %do _i=1 %to %sysfunc(countw(&othvarlist,%str( )));
          if %scan(&othvarlist,&_i) ne _%scan(&othvarlist,&_i) then modcols =
            catx(' ',modcols,"%upcase(%scan(&othvarlist,&_i))"); %end;
    end;
run;
%mend m_add_update_vars;
```

Merge together current and prior data sets, check for new records, and compare variables across existing records.

Highlighting New/Changed Data

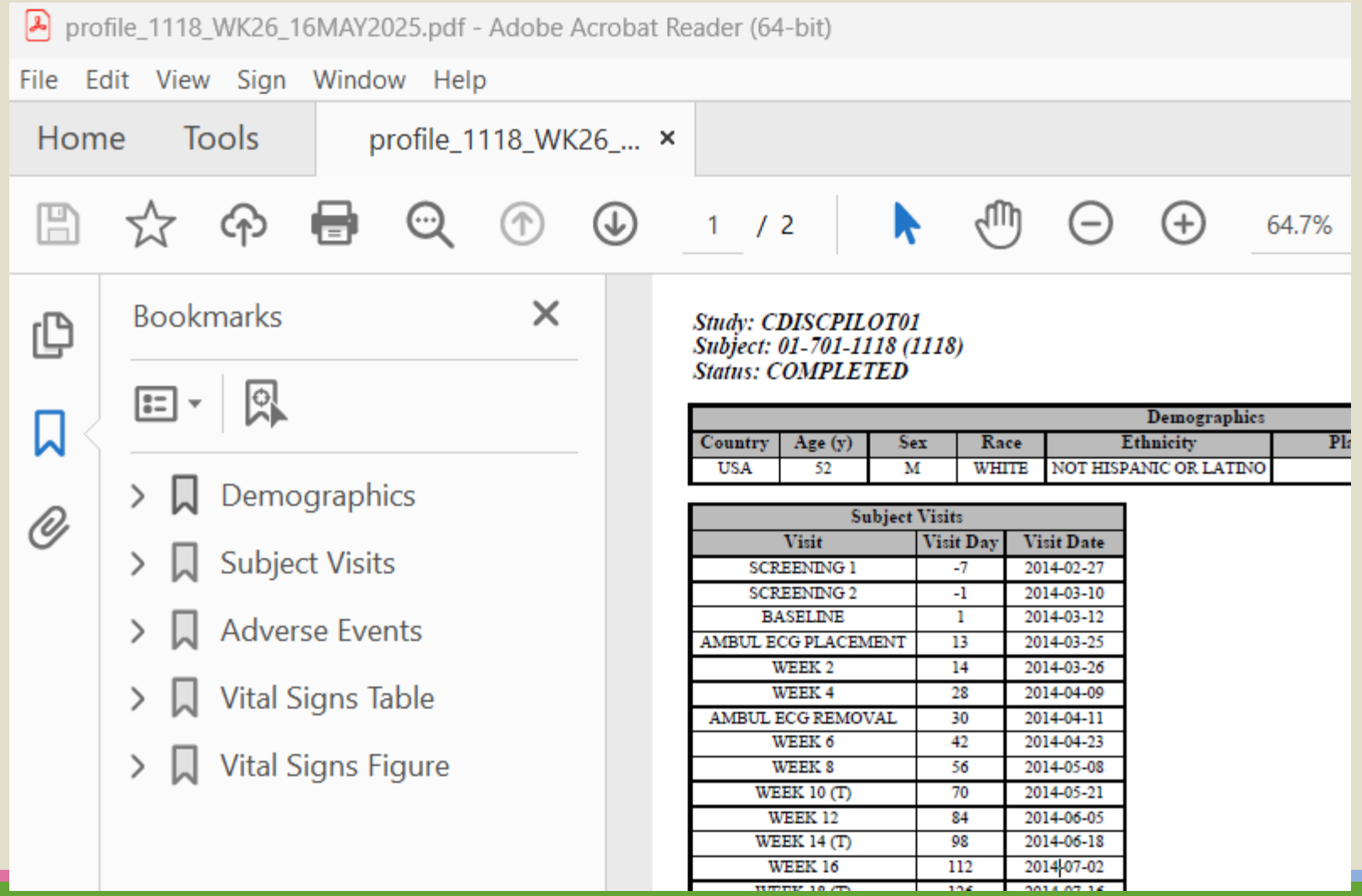
```
proc report data=aefinal split='~' style(report)=[just=left] out=work.qc_ae_&i;
  where USUBJID="&&usubj&i";
  columns AESPID /* additional variables */
    %if %nrbquote(&highlight_updates)=Y %then newflag modcols;;
  /* DEFINE statements */
  %if %nrbquote(&highlight_updates)=Y %then %do;
    define newflag    / display noprint;
    define modcols    / display noprint;
  %end;
  compute _aetrtem;
    %if %nrbquote(&highlight_updates)=Y %then %do;
    if newflag=1 then call define(_row_, "style","style=[background=lightorange]");
      do i=1 to countw(modcols);
        call define(scan(modcols, i), "style", "style=[background=lightyellow]");
      end;
    %end;
  endcomp;
run;
```

A global macro variable HIGHLIGHT_UPDATES is used to turn the highlighting on or off.

When highlighting is enabled, the COLUMNS statement is modified and additional DEFINE statements and a COMPUTE block are added.

Adding PDF Bookmarks

PDF bookmarks make it easier to navigate the patient profile output, particularly when the report contains many sections.



Adding PDF Bookmarks

- Getting the bookmarks just right requires a few adjustments:
  - On the ODS PDF statement, change the behavior with the PDFTOC=1 option.
  - Use ODS PROCLABEL to override the bookmark name for each PROC REPORT.
  - Use the CONTENTS=‘ ‘ option on PROC REPORT to suppress second level bookmarks
  - Use a BREAK statement with a dummy variable on each PROC REPORT to suppress third level bookmarks.

Demographics						
Country	Age (y)	Sex	Race	Ethnicity	Planned Arm	Actual Arm
USA	63	F	WHITE	HISPANIC OR LATINO	Placebo	Placebo

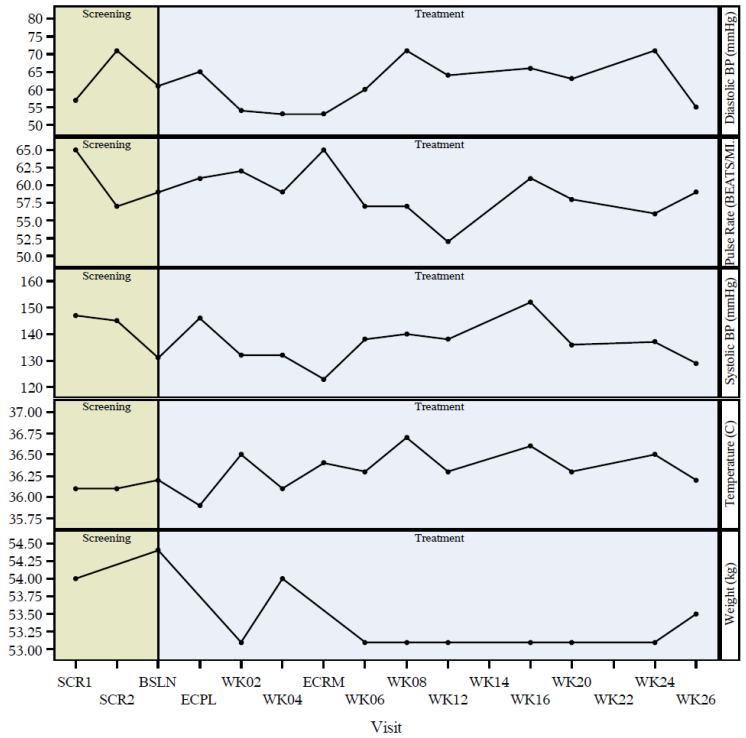
Subject Visits		
Visit	Visit Day	Visit Date
SCREENING 1	-7	2013-12-26
SCREENING 2	-1	2013-12-31
BASELINE	1	2014-01-02
AMBUL ECG PLACEMENT	13	2014-01-14
WEEK 2	14	2014-01-16
WEEK 4	28	2014-01-30
AMBUL ECG REMOVAL	30	2014-02-01
WEEK 6	42	2014-02-12
WEEK 8	56	2014-03-05
WEEK 12	84	2014-03-26
WEEK 14 (T)	98	2014-04-09
WEEK 16	112	2014-05-07
WEEK 20	140	2014-05-21
WEEK 22 (T)	154	2014-06-04
WEEK 24	168	2014-06-18
WEEK 26	182	2014-07-02

Adverse Events									
Sponsor Defined Identifier and Verbatim Term	Start Date (Day)	End Date (Day)	Ser-ious?	Dur-ation (days)	Sever-ity	Relation-ship	Outcome	Action Taken	Treatment Emergent?
E06: DIARRHOEA	2014-01-09 (8)	2014-01-11 (10)	No	3	Mild	Remote	Recovered/Resolved		Yes
E07: APPLICATION SITE ERYTHEMA	2014-01-03 (2)		No	.	Mild	Probable	Not Recovered/Not Resolved		Yes
E08: APPLICATION SITE PRURITUS	2014-01-03 (2)		No	.	Mild	Probable	Not Recovered/Not Resolved		Yes

Study: CDISCPILOT01
Subject: 01-701-1015 (1015)
Status: COMPLETED

Vital Signs									
Vital Sign Parameter	SCR1	SCR2	BSLN	ECPL	WK02	WK04	ECRM	WK06	WK08
Diastolic BP (mmHg)	57	71	61	65	54	53	53	60	71
Pulse Rate (BEATS/MIN)	65	57	59	61	62	59	65	57	57
Systolic BP (mmHg)	147	145	131	146	132	132	123	138	140
Temperature (C)	36.1	36.1	36.2	35.9	36.5	36.1	36.4	36.3	36.7
Weight (kg)	54	.	54.4	.	53.1	54	.	53.1	53.1

Vital Signs				
WK12	WK16	WK20	WK24	WK26
64	66	63	71	55
52	61	58	56	59
138	152	136	137	129
36.3	36.6	36.3	36.5	36.2
53.1	53.1	53.1	53.1	53.5



FULLY VALIDATED, For Internal Use Only

Conclusion

Summary

- Patient profiles are a useful tool for clinical data review and monitoring.
- The programming framework presented provides a flexible and customizable way to create highly customized patient profiles.
- The techniques used adapt to whatever subjects and visits appear in the data without the need for code modifications.

Possible Future Enhancements

- Use of a control file to specify:
  - Specify which subjects to include
  - Specify which data sections in include and their order
  - Allow for different sections to be specified for different subjects
  - Control grouping of parameters into pages

Contact Information

Richann Watson

DataRich Consulting

richann.watson@datarichconsulting.com

Josh Horstman

PharmaStat

jhorstman@pharmastat.com