It's Simpler Than You Think:
An Example of a Patient Summary Report Using SAS® and ODS PDF

Julie Kilburn and Leanne Goldstein
City of Hope, Duarte, California

At City of Hope, we are always trying to find ways to make everyone's job a little easier. Our current challenge was to provide a medical oncologist with an automated one-page summary of the relevant diagnostic and treatment information for the patients that she will see this week. This report was designed to provide information to be available at her fingertips so she won't have to flip through bulky files as she prepares for each patient visit. A mock up of the desired form layout was introduced, and a debate ensued about whether SAS create such a highly customized report. We knew it could be done, but we had no idea how simple the code would be until it came together. This presentation will walk through the steps taken to make this esthetically pleasing and very functional report using little more than PROC TRANSPOSE, PROC REPORT, and ODS PDF.

INTRODUCTION: THE CHALLENGE

Exhibit 1 in the Appendix is a mock up of the requested Patient Summary report. Note the sections separated by headers, with two or three columns, labels for everything but just one observation for each piece of information, since we are only looking at one patient. The SAS data set containing all the relevant information is entirely horizontal, having only one row and many columns. The quandary here is how to get the horizontal data into a form that can be displayed in chunks and make it 'pretty' with headers and columns at the right places, including bold labels for each element in the tables.

THE SOLUTION

The solution turned out to be quite simple: One of the many variations of PROC TRANSPOSE, possibly the simplest one, creates vertical data from horizontal data, and this is a key feature to generating the report. Using PROC REPORT in conjunction with ODS PDF will give us control over the appearance of our desired one-page summary.

ODS PDF

We will create the document using ODS PDF, so first we set some global options. We set the paper size, orientation, and the specific margins, and we don't want to see any titles at this point. We'll create them later in the code.

```
OPTIONS       orientation = landscape
             papersize = letter
             nodate
             nonumber
             topmargin= 0.50in
             bottommargin=0.25in
             leftmargin = 0.75in
             rightmargin = 0.75in;
TITLE;

ODS ESCAPECHAR = "^";              *1
FOOTNOTE h=8pt f=Arial j=r "^{thispage}/^{lastpage}";       *2

ODS LISTING CLOSE;         *3
ODS HTML CLOSE;
```

```
ODS PDF FILE="c:\WUSS 2014\quick tips\Pt_Summary.pdf" STYLE=journal NOTOC
STARTPAGE=NO;       *4
```

***1**. This statement is a very powerful feature that defines a character that designates the beginning of a series of formatting commands. This will be used if we add page numbers to the footnote, and to make the labels bold.

***2** . This footnote is optional, and since the document we are creating in this instance is one page, it is not necessary, but shows one of the cool tricks we can do with the escape character we just defined.

***3.** We close the other possible ODS destinations that might be open before we open the PDF destination. We promise we won't forget to open them again when we're done.

***4**. The ODS PDF statement starts the creation of the PDF document. The journal style is simple and clean. The NOTOC option turns off the Table of Contents that is generated by default with a PDF document. STARTPAGE=NO keeps SAS from putting the output from each procedure we use on a separate page.

THE DATA

Let's make sure our data is prepared. We want a data set for the one patient from whom we are creating the summary report. We need to do a little formatting to get the visit date/time to display the way we want. We will also assign labels to each variable. These need to be specified exactly how we want them to appear in the report, even including the colon.

```
PROC FORMAT; *5
PICTURE newt
  low-high='%0I:%0M %p' (datatype=time);
RUN;

DATA one_pt_summary; SET all_mrns (WHERE=(mrn=11111));

  VisitDateTime=COMPBL(PUT(DATEPART(ApptStart),weekdate.)||"at
  "||PUT(TIMEPART(ApptStart),newt.)); *6

  LABEL
       mrn='^{style [font_weight=bold]Medical Record Number:}'  *7
       ptname='^{style [font_weight=bold]Patient Name:}'
       institution='^{style [font_weight=bold]Institution:}'
       provider='^{style [font_weight=bold]Provider:}'
       visitdatetime='^{style [font_weight=bold]Patient Visit Date/Time:}'
       reportdate='^{style [font_weight=bold]Date Report Printed:'};
       …
RUN;
```

*5. Because there isn't a standard time format available that shows the 12-hour time with minutes but not seconds, we use the PROC FORMAT above.

*6. The COMPBL function condenses any extra spaces, leaving one space between words, as opposed to the COMPRESS function that removes all spaces.

*7. One of the formatting requirements was that the field labels be bolded on the report. We use inline formatting to control just the label and not the field value. Note that the ESCAPECHAR comes in handy again.

PROC TRANSPOSE

With our data set prepared, we can get to work. For each section of the report we separately transpose just those data that we want to include. The first section contains the institution name, the patient name, and the medical record number.

```
PROC TRANSPOSE DATA=one_pt_summary out=section1a prefix=value;
  VAR institution name mrn;
RUN;
```

Obs	_NAME_	_LABEL_	value1
1	institution	**Institution:**	City of Hope
2	ptname	**Patient Name:**	Jones, Barbara
3	mrn	**Medical Record Number:**	11111

Output 1: Transposed Data

The resulting data has columns for the variable name, the label (_label_), and the value (value1). We combine the label and the value into one variable by using the COMPBL function again, and we keep only that data element.

```
DATA section1a (KEEP=outlist1); SET section1;
  outlist1=COMPBL(_label_||value1);
RUN;
```

In the top section we need a second column, which will contain the report date, the patient's visit date and time, and the medical provider's name. We follow the same process for this section, using PROC TRANSPOSE and the DATA step, and then combine the two to complete the data for our top section.

```
PROC TRANSPOSE DATA= one_pt_summary OUT=section2;
  VAR reportdate visitdatetime provider;
RUN;

DATA section2a (KEEP=outlist2); SET section2;
  outlist2=COMPBL(_label_||value1);
RUN;

DATA topsection;
MERGE section1a section2a;
RUN;
```

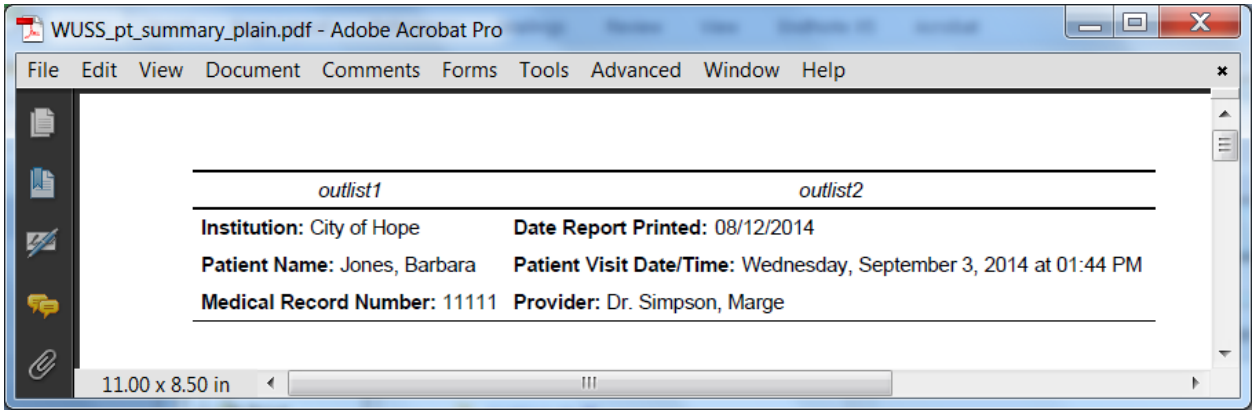
outlist1	outlist2
Institution: City of Hope	**Date Report Printed:** 08/12/2014
Patient Name: Jones, Barbara	**Patient Visit Date/Time:** Wednesday, September 3, 2014 at 01:44 PM
Medical Record Number: 11111	**Provider:** Dr. Simpson, Marge

Output 2: The top section

PROC REPORT

We are now ready to display the first section of our summary report. We have control over modifying the appearance using some simple features of PROC REPORT. The first figure below shows the output of a PROC REPORT without any modifications using no-frills code.

```
PROC REPORT DATA = topsection  NOWD;
  COLUMN outlist1 outlist2;
RUN;
```



Display 1: PDF output from basic PROC REPORT

Adding some style, we can make it exactly like the mock up. We need a title that is centered, larger font than the body, and highlighted in gray. We need to remove the black lines and the column headers. We want the columns to be left-justified, and the report to take up the width of the usable paper space. It will become evident after building the other sections why this is important. The code below is broken down into our needed elements.

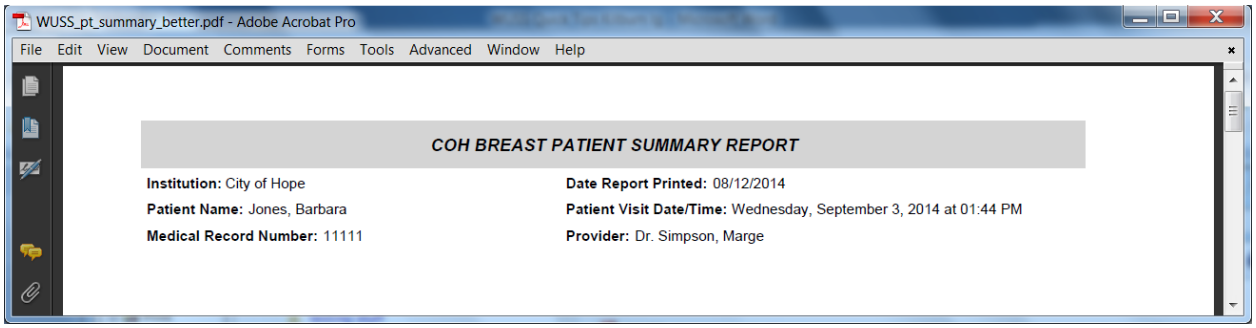
```
PROC REPORT DATA = topsection  NOWD
STYLE(header)=[font_size=12pt                    *8
       cellheight=70
       font_face=arial
       fontweight=bold
       bordercolor=white
       borderwidth=1
       vjust=c
       just=c
       background = very light grey]

STYLE(column)=[font_size=10pt                    *9
       font_face=arial
       borderwidth=1
       vjust=t
       just=l
       background = white]

STYLE(report)=[background=white                  *10
bordercolor=white];

COLUMN ("BREAST PATIENT SUMMARY REPORT" outlist1 outlist2);        *11
DEFINE outlist1     /display " " style(column)=[cellwidth=4.0in];
DEFINE outlist2     /display " " style(column)=[cellwidth=5.0in];
RUN;
```

*8. The header becomes a wider gray band with vertically and horizontally centered bold print. Using a tiny border width and a border color of white makes those black lines effectively disappear.

*9. The columns need to have smaller font, be left and top justified, and have a white background.

*10. We want to make sure that line at the bottom of the output goes away, so we color the background and the border of the report white.

*11. The COLUMN statement with the header that covers both data elements creates the title for this first section. In the DEFINE statement we create a blank column header, and control the width of our report section as well as the placement of the two fields. In the figure below, we see that our header stretches across a full nine inches of our page.



Display 2: PDF Output from modified PROC REPORT

REPEAT

Building the other sections is similar to what has just been described. We transpose the fields we want to use for each of the columns, combine the labels and values into one variable, and use them in PROC REPORT. The exception to this repeated exercise is the second section where there is only one row needed. In this case, we do not need to transpose anything. This creates a small problem since we won't have the label in a field already. The simplest solution will be to create the display variables for age, race, and ethnicity by adding on the label's text, including the inline formatting to keep the bold lettering consistent with the other labels in the report.

```
DATA secondsection; SET one_pt_summary;
  display_age='^{style [font_weight=bold]Current Age:  }' ||   put(age,3.) || '
   years';
  display_race='^{style [font_weight=bold]Race:  }' || strip(race);
   display_eth='^{style [font_weight=bold]Ethnicity:  }' ||    strip(ethnicity);
RUN;
```

For this section and subsequent sections, the header height should be a little smaller, and we can adjust the width of the cells of each section to line up as much as is reasonable with the other sections. The final steps in creating the report are to close the PDF destination so that the PDF document will be generated, and then to reopen the other output destinations that were open before.

```
ODS PDF CLOSE;
ODS LISTING;
ODS HTML;
```

The final report is shown as Exhibit 2 in the Appendix.

CONCLUSION

We have successfully demonstrated that SAS can be use to tackle challenges of creating reports with horizontally displayed data. It was actually quite satisfying to realize that what we first thought would be difficult proved to be rather simple using SAS. Our goal of creating a nice and functional report that would help the doctors have quick access to relevant patient data was met. Since its inception the project has taken on a new dimension: The

expanded report can be automatically generated for all of a given doctor's weekly appointments, and a link to the individual reports can be emailed so the doctor truly has her patients' information at her fingertips.

Contact Information

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

Julie Kilburn
City of Hope
883 E. Kanai Ave.
Porterville, CA 93257
(559) 791-9384
juliekilburn@sbcglobal.net

Exhibit 1: The Mockup

DRAFT

COH BREAST PATIENT SUMMARY REPORT

Institution: City of Hope

Date Report Printed: XX/XX/XXXX

Name: Last Name, First Name

Patient Visit Date/Time: XX/XX/XXXX at XX:XX am/pm

Medical Record Number: XXXXXXXX

Provider: Dr. XXXXXXXXX

PATIENT CHARACTERISTICS

Current Age: XXX years

Race: XXX

Ethnicity: XXX

DIAGNOSIS

Diagnosis Date: XX/XX/XXXX

Diagnosis: (text)

Biopsy Date: XX/XX/XXXX

Pathology Diagnosis: (text)

Definitive Surgery Date: XX/XX/XXXX

Definitive Surgery: (text)

STAGING and PATHOLOGY

Histology: XXX

Pathologic T Stage: XXX

ER Status: XXX

Laterality: XXX

Pathologic N Stage: XXX

PR Status: XXX

Tumor Site: XXX

Pathologic M Stage: XXX

Her2Neu Status (via FISH): XXX

Tumor Size: XXX

Lymph Node Sampling: XXX

Her2Neu Status (via IHC): XXX

Surgical Margins: XXX

Exhibit 2: The Final Report

COH BREAST PATIENT SUMMARY REPORT

Institution: City of Hope
Date Report Printed: 08/12/2014
Patient Name: Jones, Barbara
Patient Visit Date/Time: Wednesday, September 3, 2014 at 01:44 PM
Medical Record Number: 11111
Provider: Dr. Simpson, Marge

PATIENT CHARACTERISTICS

Current Age: 51 years
Race: White
Ethnicity: Non-Hispanic

DIAGNOSIS

Diagnosis Date: 03/09/09
Diagnosis: L breast ca; Lbreast, UOQ; infil Lobular CA

Biopsy Date: 03/09/09
Pathology Diagnosis: 3/9/09(No ROS)L breast mass core bx:c/w infil lobular ca R1. L breast lumpectomy:infil lobular ca, MD. BR score:6/9No insitu present. TS:1.5cm Marge(+) L ax sentinel LN Bx:0+/2 R2. L breast mastecotmy:resid foci of lobular ca. Margs:(-) Skin & Nipple:(-) R breast mastectomy:(-)

Definitive Surgery Date: 05/24/09
Definitive Surgery: L breast total mastecotmy w/Rbreast and tram flap

STAGING and PATHOLOGY

Histology: Infil Lobular Ca
Pathologic T Stage: 1C
ER Status: Positive
Laterality: LEFT
Pathologic N Stage: 0
PR Status: Negative
Tumor Site: L breast, UOQ
Pathologic M Stage: NA
Her2Neu Status (FISH): Negative/Normal
Tumor Size (in mm unless specified): 015
Lymph Node Sampling: SENTINEL LN BX
Her2Neu Status (IHC): Test not done
Surgical Margins: MICROSCOPIC

1/1