PharmaSUG 2025 - Paper AP-162

Mining Data from PDF Files Using SAS

Michael Stout, and Brian Knepple, J&J Orthopaedics

ABSTRACT

SAS is a powerful software tool and can read and process data from multiple file formats. SAS can efficiently read non-SAS datasets, such as Text, EXCEL, and CSV files. It is much more challenging to read and process files in PDF (portable document format) format. This paper describes an approach and pitfalls of mining data embedded in PDF files. Reading and processing data from PDF files can be challenging, but the gain is worth the pain. This process has reduced the amount of time and improved accuracy of manually collating data for summary reports.

INTRODUCTION

Clinical needed a way to quickly summarize standardized reports from external sources in PDF format. Mined data from these reports support post-market surveillance activities. The Clinical group was manually creating summary reports, and wanted to automate the process as the creation of these reports was time consuming and led to human errors. Human error may happen when numbers are mistyped or transposed. Extensive validation activities were required for every report to ensure that the reports were accurate.

The external PDF reports have numerous tables and figures. Fortunately, the format of the tables and data are consistent across the standardized reports.

The team started a project to automate the summarization of the standardized reports. The team used SAS® to summarize the results because SAS resources were available to create a prototype. The goal of this project was to automate the extraction and summary of data from external PDF reports. For now, the export process is manual. In the future, we will attempt to automate the exporting of data from PDF format to CSV.

METHODS

Parsing of data from PDF files are more complex than processing data in other file formats. Data step processing and string functions are used to summarize the data.

Below are actions taken to summarize data from external reports in PDF format.

1. Export PDF file(s) into CSV format
2. Remove special characters
3. Read CSV Data
4. Group data (general, tables and figures)
5. Clean data
6. Find and pull data
7. Save Summary data
8. Review output

Reading this paper will provide knowledge on how to mine data from PDF files.

DATA FLOW

Mining data involves pulling individual datapoints from PDF files. The first step is converting the PDF file into CSV format. A SAS program is then used to find the datapoints, extract the values and perform derivations, as needed. Finally, data is formatted and written to a summary dataset in EXCEL format. Output could be written to one or more output destinations. Figures 1-4 show how data, highlighted in yellow, flows from PDF format to EXCEL Summary File.

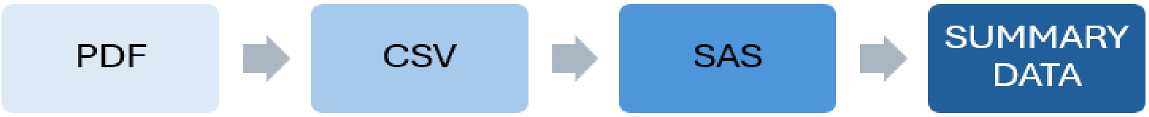


Figure 1 shows table in PDF format.

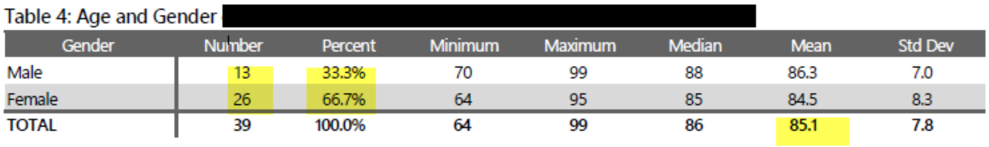
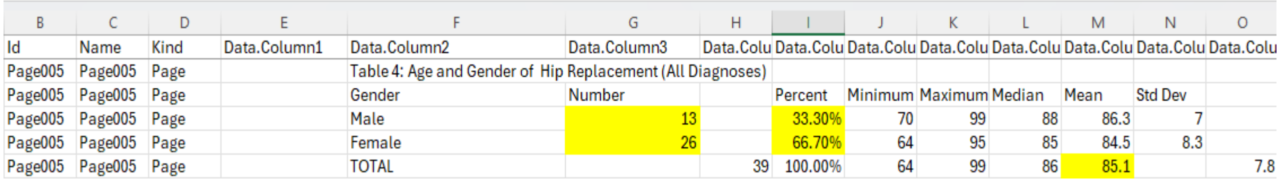


Table 4: Age and Gender [redacted]

Gender	Number	Percent	Minimum	Maximum	Median	Mean	Std Dev
Male	13	33.3%	70	99	88	86.3	7.0
Female	26	66.7%	64	95	85	84.5	8.3
TOTAL	39	100.0%	64	99	86	85.1	7.8

Figure 1. Data in PDF format

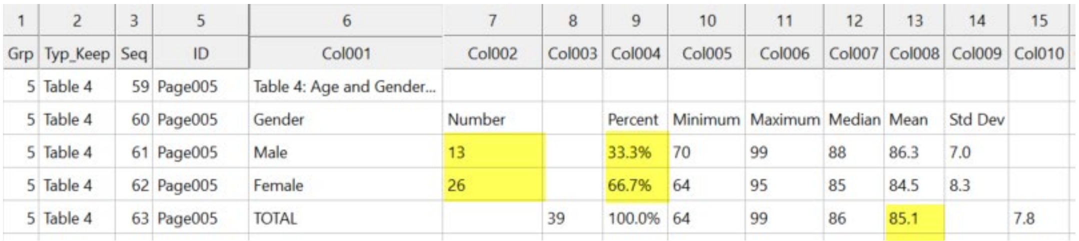
Figure 2 shows how data from Figure 1 is rendered in CSV format.



B	C	D	E	F	G	H	I	J	K	L	M	N	O
Id	Name	Kind	Data.Column1	Data.Column2	Data.Column3	Data.Colu	Data.Colu	Data.Colu	Data.Colu	Data.Colu	Data.Colu	Data.Colu	Data.Colu
Page005	Page005	Page		Table 4: Age and Gender of Hip Replacement (All Diagnoses)									
Page005	Page005	Page		Gender	Number		Percent	Minimum	Maximum	Median	Mean	Std Dev	
Page005	Page005	Page		Male	13		33.30%	70	99	88	86.3	7	
Page005	Page005	Page		Female	26		66.70%	64	95	85	84.5	8.3	
Page005	Page005	Page		TOTAL		39	100.00%	64	99	86	85.1		7.8

Figure 2. Data converted to CSV format

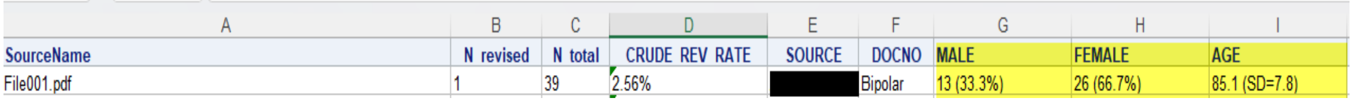
Figure 3 shows output generated by SAS after grouping and manipulating data.



1	2	3	5	6	7	8	9	10	11	12	13	14	15
Grp	Typ_Keep	Seq	ID	Col001	Col002	Col003	Col004	Col005	Col006	Col007	Col008	Col009	Col010
5	Table 4	59	Page005	Table 4: Age and Gender...									
5	Table 4	60	Page005	Gender	Number		Percent	Minimum	Maximum	Median	Mean	Std Dev	
5	Table 4	61	Page005	Male	13		33.3%	70	99	88	86.3	7.0	
5	Table 4	62	Page005	Female	26		66.7%	64	95	85	84.5	8.3	
5	Table 4	63	Page005	TOTAL		39	100.0%	64	99	86	85.1		7.8

Figure 3. Data converted to SAS format

Figure 4 shows how data is represented in the final output file.



A	B	C	D	E	F	G	H	I
SourceName	N revised	N total	CRUDE REV RATE	SOURCE	DOCNO	MALE	FEMALE	AGE
File001.pdf	1	39	2.56%	[redacted]	Bipolar	13 (33.3%)	26 (66.7%)	85.1 (SD=7.8)

Figure 4. Final Output File

EXPORT PDF FILES

SAS can read uncompressed PDF files created by ODS PDF and PROC REPORT. Unfortunately, we were not able to read our PDF files directly into SAS. This required a different approach! Bummer.

Software tools are available that convert PDF files to CSV format. We tried multiple methods to convert the PDF files, before selecting a tool. We experimented with Adobe Acrobat, R, and Excel Power Query Tool. Other tools are available but not evaluated. We selected the Excel Power Query Tool. The tool can combine and process multiple PDF files and put them in two formats: page, and table. Although data generated in table format renders better than page format, we selected data in page format. Page format retained data for text and tables. Table format only keeps information data associated with tables. Choose a tool that renders data in a format that makes it easiest to read in SAS.

If you are adventurous, try reading PDF files using open-source methods such as JavaScript. APIs are available to parse and extract text from PDF files programmatically.

REMOVE SPECIAL CHARACTERS

It is important to remove special characters from the CSV file before reading the data into SAS. Remove tab and carriage returns by reading the CSV file as a binary file and replacing them with an acceptable value. Refer to SAS Paper CC06, PharmaSUG 2008) for more information on removing special characters.

READ CSV DATA

Read the CSV file using a SAS input statement. This provides more control on how SAS reads the data and reduces the risk of truncating the data. The input statement allows SAS to assign the name and length to each field. Embedded quotes can cause issues, so this code drops double quotes found in text fields. The code assigns a general variable name for each column of data in the CSV files, which represent the text, tables, and figures from the PDF file. The code reads 69 columns of data from the CSV file. Update the code to increase the number of columns processed. Generic variable names make it easier to scan the datasets for keywords and phrases.

```
data Read_CSV;
infile "&fname..csv" linesize=10000 DLM=',' dsd firstobs=2 missover truncover;

input SourceName                      :$150.
      ID                              :$10.
      NamePrimaryProcedureID          :$10.
      Kind                            :$10.

%macro mymac;
      %do i = 1 %to 69;
        Col%sysfunc(putn(&i,z3.))        :$200.
      %end;
                ;

      /* drop double quotes */
      %do i = 1 %to 69;
        Col%sysfunc(putn(&i,z3.)) = compress(Col%sysfunc(putn(&i,z3.)),'"');
      %end;
      ;
%mend;
%mymac;

run;
```

Program 1. Caption for Sample Program

GROUP DATA

Group data is extremely helpful. The program classifies the data as general, table, or figure. Understanding the format of the data is important. The program scans the text and looks for keywords to group the data. The program assumes that all tables and figures have a title line that has the keyword "Title" or "Figure". The keywords identify the beginning of a new table/figure and the end of the previous table/figure. Text prior to the first "Table" or "Figure", is set to "General" text. “General” text has information about the report, product information, and run date. The program assigns a unique group number to each group.

CLEAN DATA

None of the tools we reviewed rendered the data cleanly. The programmer may need to manipulate the data before using it. Table values may need to be aligned, and empty columns removed. Notice how data in columns H and O, in Figure 2, were shifted to the right by the PDF conversion tool.

FIND AND PULL DATA

Write SAS code to find the location of data points needed for further analysis. Use unique words/text to pinpoint each datapoint and then save the text value. Continue pulling data for all required data points. Consolidate the data into a single data set. Most data come from a single row. Other data values are on multiple records. For example, revision reasons come from multiple rows in the table.

SAVE SUMMARY DATA

After finding and pulling desired data points, format the data and save it to SAS, Excel, RTF, or another format. Clinical wanted data output to an Excel file so they could review and then cut and paste information as needed. As requested by Clinical, the program wrote data to multiple tabs on the spreadsheet.

REVIEW OUTPUT

Always review the summary data, since the tools used to create the PDF files may render the text, tables, and figures in unexpected ways. The programmer can manually review the results or program checks to flag inconsistencies in the summary data.

PITFALLS

Although PDF files offer a fantastic way to view tabular reports, it does not have methods to easily process PDF files. Data can become corrupted if special characters are present in the PDF file and need removed. The tool used to convert the data to CSV format may not understand all characters. Programmers must add logic to account for unusual formatting of data in tables. The creator of the PDF file may have changed the format of the report. As a result, keywords may disappear over time. Keywords in the program may need to be updated. Clinical should monitor and carefully review the summary data. Clinical needs to investigate any irregularities and work with the SAS team to resolve the issue, if needed.

CONCLUSION

Pulling data from PDF files is more difficult than other types of files. With good planning and creative programming, you can efficiently pull data from PDF datasets to summarize results. This is a good approach when the general format and structure of the report does not change very often. Replace manual processes, reduce validation activities, and improve overall quality of data pulled from PDF files. Using a similar approach, you can pull data from internal and external reports that are in PDF format.

ACKNOWLEDGMENTS

Special thanks to Sean Croker for evaluating PDF conversion tools and converting PDF files into CSV format.

CONTACT INFORMATION

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

<Michael Stout>
<Mstout2@its.jnj.com>