Example and Tutorial for the StatRep Package

Martial Phélippé-Guinvarc'h
Le Mans Université
martial.phelippe-guinvarch@univ-lemans.fr

Based on the original work by Tim Arnold and Warren F. Kuhfeld (SAS Institute Inc. 2015)

April 28, 2026

Introduction

This article provides an example and a tutorial that show how to use the StatRep LaTeX package. For complete details see the *StatRep User's Guide* that accompanies the package (**statrepmanual.pdf**). The package is available for download at http://support.sas.com/StatRepPackage.

When you use the StatRep LaTeX package, you use the following four-step process to create an executable document that enables you to ensure that your research results are reproducible:

1. Create your LaTeX document so that it contains your text, data, and SAS code.
2. Compile your document with pdfLaTeX to generate the SAS program.
3. Execute the SAS program to capture your output. For each code block in your document, SAS creates a SAS Output Delivery System (ODS) document that contains the resulting output.

   For each output request in your document, SAS replays the specified output objects to external files. All your requested output is generated and captured when you execute the generated SAS program.
4. Recompile your LaTeX document. In this step, the requested outputs are embedded in the resulting final PDF document.

   You might need to repeat this step so that LaTeX can measure the listing outputs to ensure that they are framed appropriately.

Step 1: Create Your LaTeX Document

This article provides you with an example LaTeX document already created as in step 1.

The purpose of the **Datastep** environment is to read in data. It produces no output.

The SAS statements in the **Datastep** environment create a new data set called **Wine**. The **first=** and **last=** options specify that only a portion of the data set be displayed.

...descriptive text that introduces data...

```
data Wine;
   input WineType $ VisitLength @@;
   datalines;
white  80 white  98 white 115 white  89 white 103
white  91 white 119 white  31 white 109 white  95

... more data lines ...

red   101 red    92
;
```

The purpose of the **Sascode** environment is to generate output. The statements in the following **Sascode** environment perform an analysis that uses the ANOVA procedure (PROC ANOVA). You use the **store=** option so that later in your document you can refer to output that is created in the **Sascode** environment. In this example, all output that is generated by the analysis is stored in the ODS document **wineA**.

```
PROC SQL NOPRINT;
SELECT MEAN(VisitLength)
INTO :MeanOfVisitLength
FROM Wine;
QUIT;

PROC PRINT DATA=WINE(OBS=3);
RUN;

%STATREPMACROVAR(MeanOfVisitLength)
```

Figure 1: Proc print, with a mean \MeanOfVisitLength= 96,28

Obs.	WineType	VisitLength
1	white	80
2	white	98
3	white	115

...descriptive text that introduces the analysis...

```
ods graphics on;
proc anova data=Wine;
   class WineType;
   model VisitLength = WineType;
run;
ods graphics off;
```

The **\Listing** and **\Graphic** tags specify the output to be displayed. The purpose of the **\Listing** tag is to display tabular output and notes. The purpose of the **\Graphic** tag is to display graphical output.

...descriptive text that introduces output...

The **\Listing** tag selects three output tables from the **wineA** ODS document: **ClassLevels**, **NObs**, and **OverallANOVA**

Figure 2: Analysis of Variance for Visit Length

La procédure ANOVA

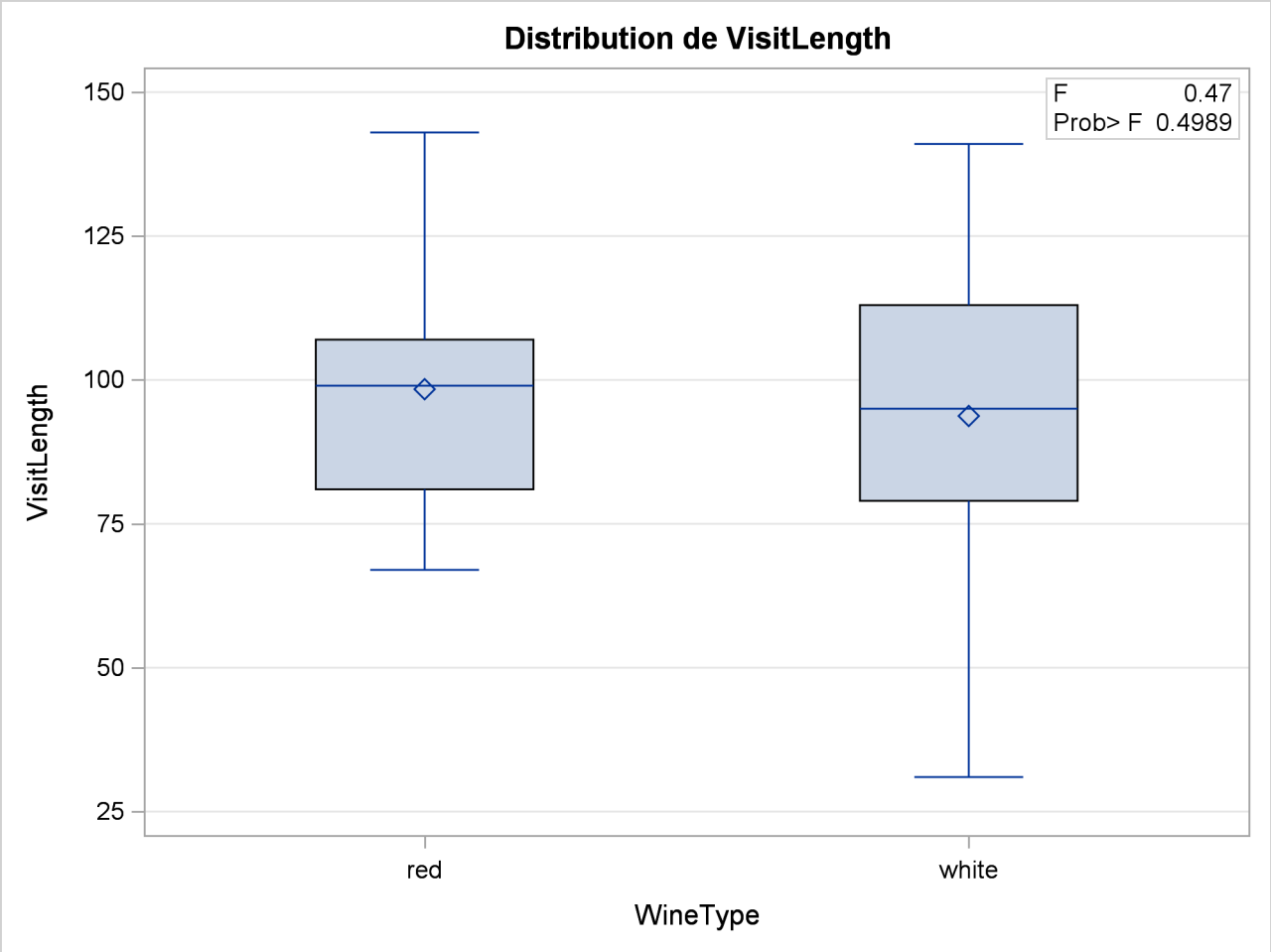
Variable dépendante : VisitLength

Source	DDL	Somme des carrés	Carré moyen	Valeur F	Pr > F
Modèle	1	225.40896	225.40896	0.47	0.4989
Erreur	40	19363.16247	484.07906		
Total sommes corrigées	41	19588.57143			

The **\Graphic** tag selects the **BoxPlot** graph from the **wineA** ODS document.

...descriptive text that introduces the graph...

Figure 3: Box Plots for Visit Length



Step 2: Compile Your Document

You can compile your document as follows from the command line:

```
pdflatex quickstart.tex
```

If you use a LaTeX-aware editor, such as TeXworks, you can use it to compile your document.

In either case, when you compile your document, the StatRep system produces a PDF file and generates a SAS program.

Note: The requested output is missing after the first compile step. (This is normal at

this step.)

Step 3: Execute the SAS Program

Open the generated SAS program **`quickstart_SR.sas`** in a SAS session. From the SAS main menu, select Run→Submit.

In this step, SAS generates the results that are requested in your document. By default, tabular output files are created in the 'lst' subdirectory and graphic output files are created in the 'png' subdirectory.

The file `quickstart_VarSAStoTex.tex` is automatically generated by the SAS program and contains `\def` commands of the form `\def\MacroName{value}`. These definitions are loaded at compile time and can be used directly in the LaTeX document.

Step 4: Recompile Your LaTeX Document

The last step is to recompile your document with pdfLaTeX. As in the first compilation, you can use a LaTeX-aware editor such as TeXworks, or you can use the **`pdflatex`** command in a terminal window.

In this recompilation step, the outputs that are captured by the SAS program are included in the final PDF document.

Conclusion

When you generate the SAS program by compiling your document with pdfLaTeX, the StatRep package does the following:

- The lines in the **`Datastep`** environment are passed unchanged to the program.
- The lines in the **`Sascode`** environment are parsed for line commands and passed to the program.
- Each **`\Listing`** tag selects the specified notes and tables.
- Each **`\Graphic`** tag selects the specified graphs.

When you execute the generated SAS program, the output that is specified in the **`\Listing`** and **`\Graphic`** tags is automatically captured.

When you recompile your LaTeX document, the **`\Listing`** and **`\Graphic`** tags insert the requested SAS results and page breaks are handled automatically.