



A User Interface To Your SAS Programs



Hailemichael M. Worku, OCS Life Sciences, the Netherlands

What's it about?



1. Working with SAS programs and its challenge for **Reproducible Reporting**.
2. Using **Add-In in SAS EG** for improving user productivity.
3. The process of building an add-in/application (**Problem => Design => Implementation**).
4. Recap & Conclusion.

Working with SAS script



The SAS System

The UNIVARIATE Procedure
Variable: Weight

Moments			
N	19	Sum Weights	19
Mean	100.026316	Sum Observations	1900.5
Std Deviation	22.7739335	Variance	518.652047
Skewness	0.18335097	Kurtosis	0.68336484
Uncorrected SS	199435.75	Corrected SS	9335.73684
Coeff Variation	22.7679419	Std Error Mean	5.22469867

Basic Statistical Measures			
Location		Variability	
Mean	100.0263	Std Deviation	22.77393
Median	99.5000	Variance	518.65205
Mode	84.0000	Range	99.50000
		Interquartile Range	28.50000

Note: The mode displayed is the smallest of 4 modes with a count of 2.

Tests for Location: Mu0=0

Challenges for Reproducible Reporting



- What If – a question by ‘normal’ SAS users?
 - A report about weight of subjects by gender (only one or both).
 - Exclude subjects whose weight value are below/above some threshold value.
 - The same analysis for age, height, or different class dataset.
- Users are often interested about their data and analysis results, but not primarily interested about the tool itself.

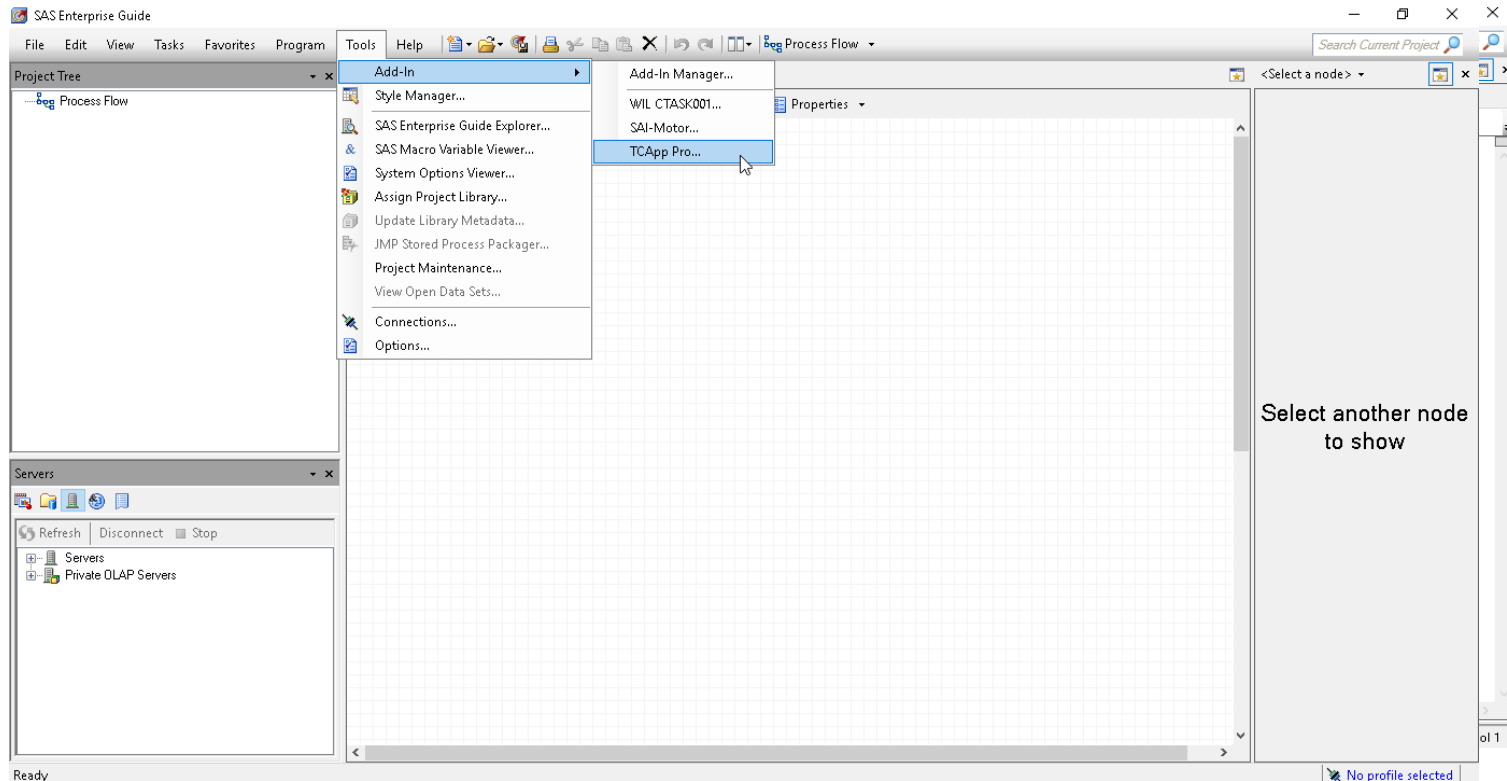


The Other Challenges

- Lack of interactivity with users:
 - **Input parameters** (e.g., loading data from different source, cleaning data, testing hypothesis, generate report).
- The tool (SAS) is sometimes difficult for users who have little or no affinity for programming.



The Power of Both Worlds: Interface + SAS





Case Study: Temperature Conversion App

The Interface:

Temp Converter App (Pro)

Enter Temperature:

37.75

From / To:

°C / °F

Convert

Result:

99.95

Reset

The Back-end (SAS programs):

```
%let txtFld_tempInput = 37.75;
```

```
%let comboBox_fromTo = 2;
```

```
%include "temp_converter_pro.sas";
```

Real-world Applications

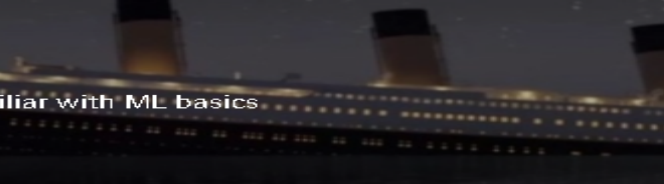


  [Competitions](#) [Datasets](#) [Kernels](#) [Discussion](#) [Jobs](#) [...](#) [Sign In](#)

 Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics



 Kaggle · 8,447 teams · 2 years to go

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#)

Overview

[Description](#)
[Evaluation](#)
[Frequently Asked Questions](#)
[Tutorials](#)

Start here if...

You're new to data science and machine learning, or looking for a simple intro to the Kaggle prediction competitions.

Competition Description

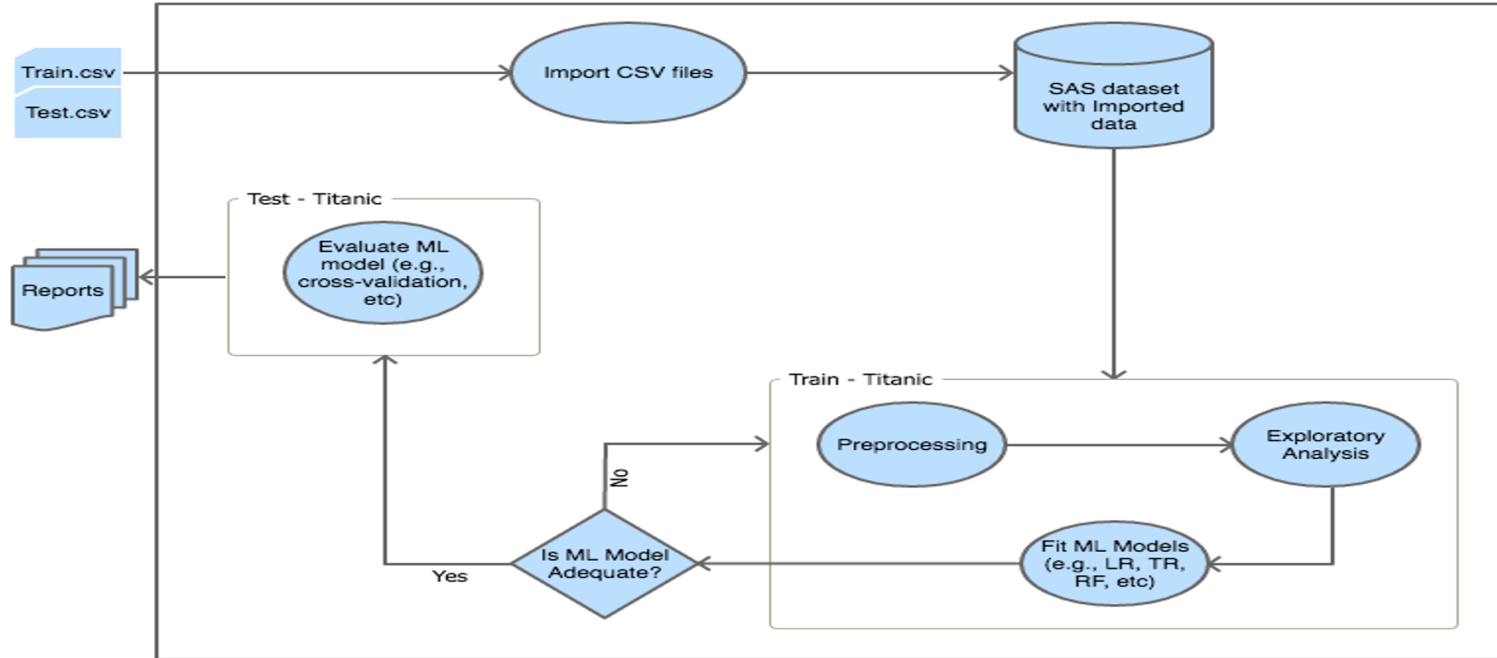
The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this challenge, we ask you to complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy.



The Titanic Explorer App: Process Flow Diagram



The Titanic Explorer App: The Front-end / Interface



Titanic Explorer App

Import Data Load Data

Machine Learning (ML) ☐ Train - Titanic ☐ Test - Titanic

Train - Titanic

Preprocessing

Select covariate: Age

Drop missing values: ☐

Exploratory Analysis

Summary statistics: ☐

Graphs: Histogram Display

Fit ML Model

Model type: Logistic regression

Apply Transformation: Log Display

Reset View Log Generate Report

Output Window

Progress bar: 10 green squares

The Titanic Explorer App: The Back-end / SAS programs



```
73
74 /*****
75 Step 0. Load Titanic raw dataset.
76 *****/
77 %load_data(raw_dsin = &intrfcToSAS_LoadArea_btnLoad.);
78
79
80 /*****
81 Titanic Explorer App.
82 *****/
83 %macro TitExpApp(MLArea = );
84
85 /* Training phase. */
86 %if %sysfunc(strip(&MLArea.)) = 1 %then %do;
87 /*****
88 Step 1. Preprocess raw Titanic data.
89 *****/
90 %preprocess_data(dsin = work.train,
91                 varin = &intrfcToSAS_preprArea_comboBox.,
92                 drop_missing = &intrfcToSAS_preprArea_chkBox.,
93                 debugme = n);
94
95 /*****
96 Step 2. Explore processed dataset.
97 *****/
98 %explore_data(show_descStat = &intrfcToSAS_explArea_chkBox.,
99              graph_type = &intrfcToSAS_explArea_comboBox.,
100              debugme = n);
101
102 /*****
103 Step 3. Fit ML model on processed data.
104 *****/
105 %fit_MLModel(model_type = &intrfcToSAS_fitMLArea_comboBox.,
106              apply_transformation = &intrfcToSAS_fitMLArea_comboBox.,
107              debugme = n);
108 %end;
109 %else %if %sysfunc(strip(&MLArea.)) = 2 %then %do;
110 /* %test_MLModel(...); */
111 %end;
112
113 %mend TitExpApp;
114
115 /* Run the app. */
116 %TitExpApp(MLArea = &intrfcToSAS_MLArea_RadioBtn.);
117
118 /*****
119 Step 4. Generate report.
120 *****/
121 /*%generate_report(...); */
122
```

Recap & Conclusion



- **The Power of Both Worlds:**
 1. The Interface allows users to interact with the application.
 2. SAS (the 'brain' of the application) does house-cleaning for us and perform the real task.
- Add-Ins can provide us an opportunity to build in-house solutions based on users experience/needs and company standards.
- The SAS program that runs at the back-end is platform independent.



Thank You