A User Interface To Your SAS Programs

Hailemichael M. Worku, OCS Life Sciences, the Netherlands

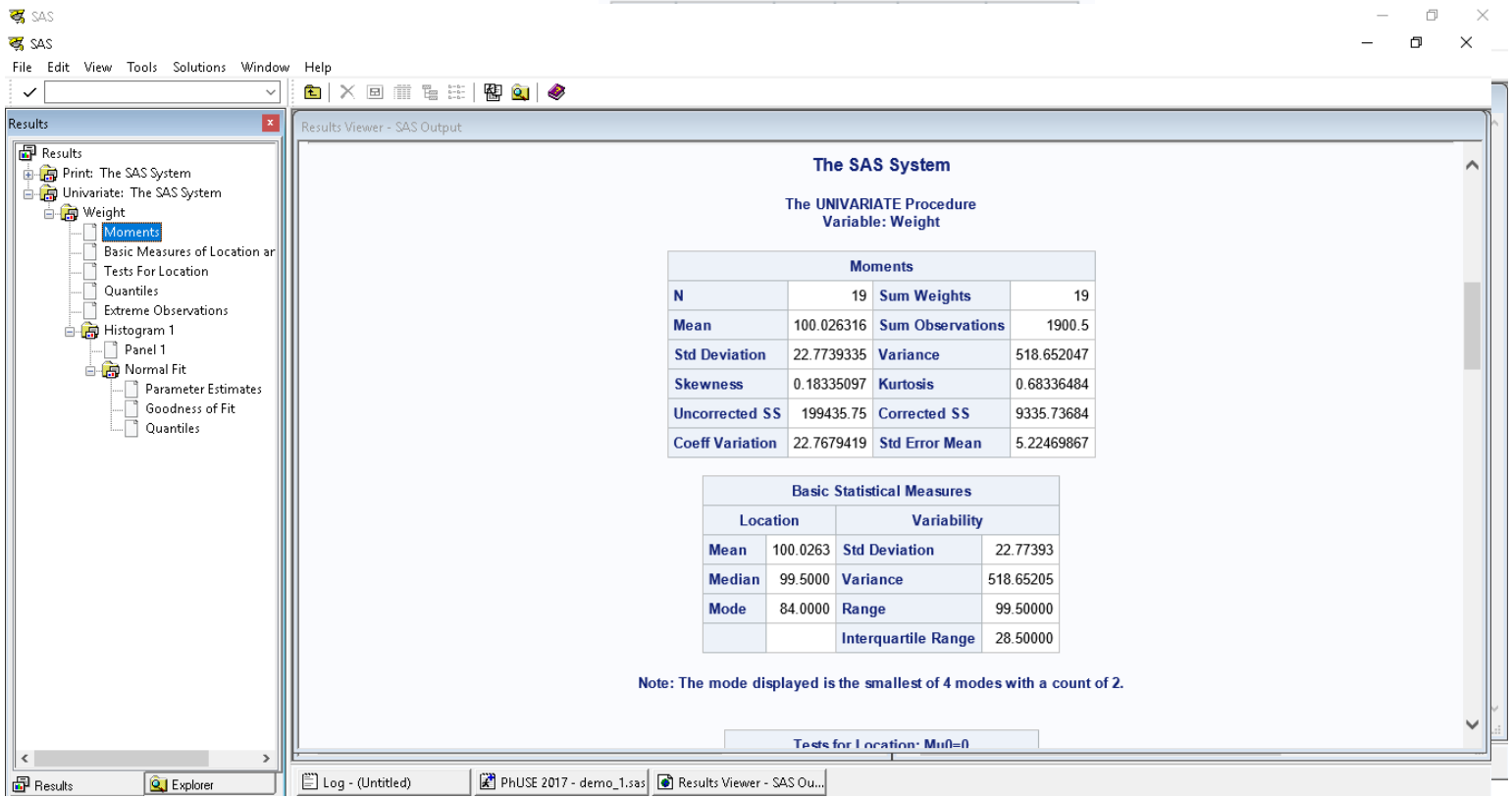
Agenda

- Introduction
- **Case Study**: SAS Base for a Reproducible Research
- **An Alternative Approach**: via SAS EG
- Application:
  - Front-end / Interface
  - Back-end / 'The Magic'
  - Process Flow Diagram / logic
- Conclusion

Introduction

- SAS programs are usually plain text SAS scripts.
- We ran them in a manual or automated fashion.
- Challenges for ‘normal’ Users:
  - Lack of interactivity with users (e.g., input parameters).
  - Difficult for users who have little or no affinity for programming.
  - Users are often interested on the final report for review or submission.

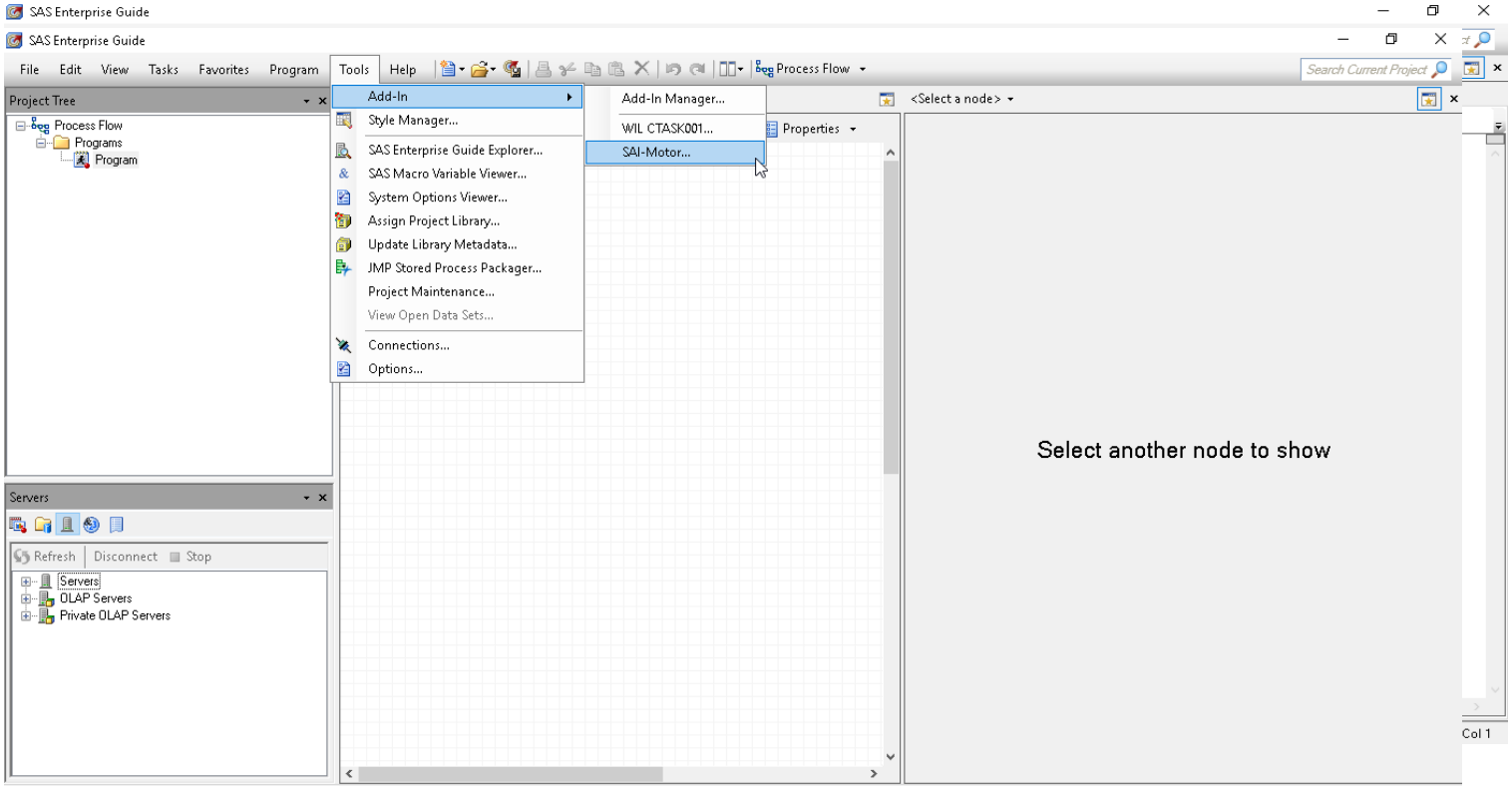
Case Study 1: SAS Base for Data Analysis



Case Study 1 (Cont'd): The Challenges?

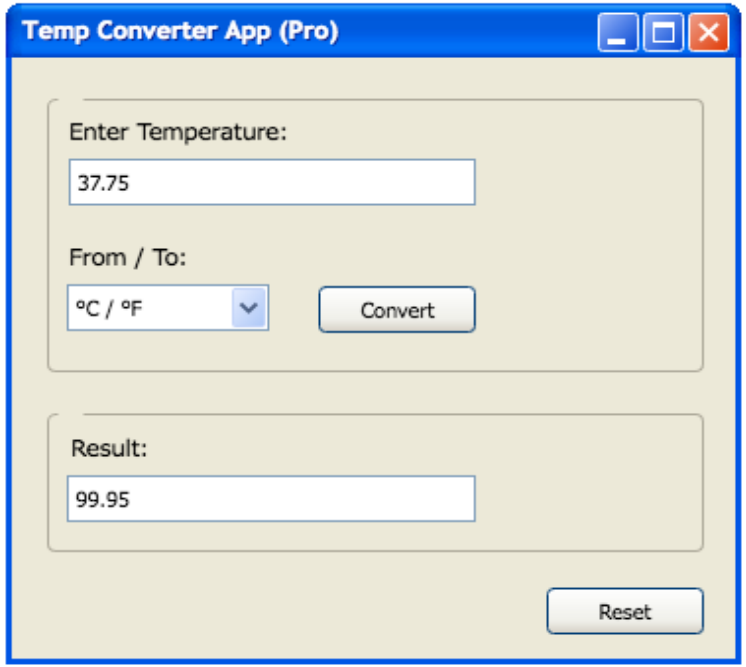
- What If – a question by 'normal' users?
  - A report for weight of subjects by gender (both or single).
  - Exclude a subject whose weight value is below/above certain threshold value.
  - Import any type of class dataset and perform similar tasks.
- Do users 'care' about what is happing in the back-end / The 'Magic'???

Alternative Approach: via SAS EG



Case Study 2: Temp Converter App

The Interface:

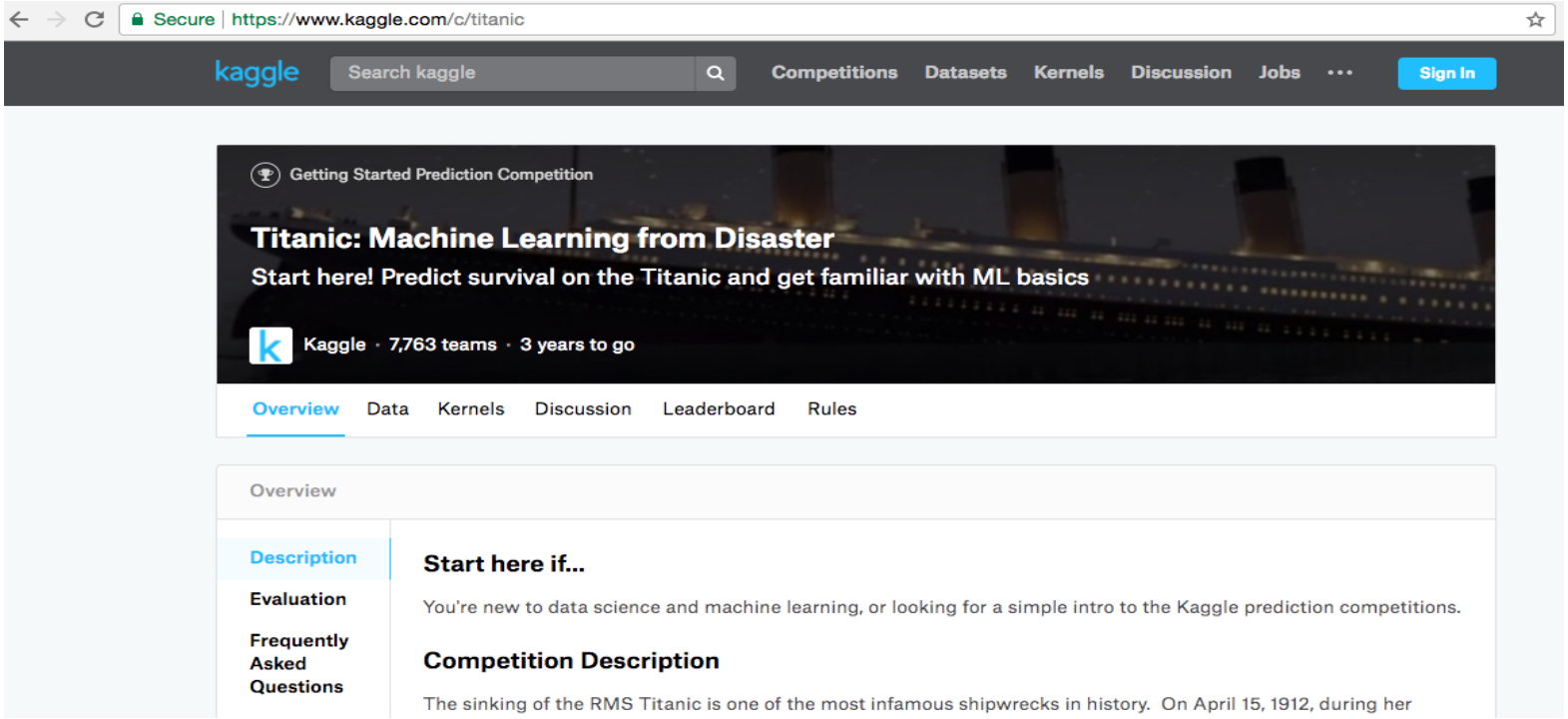


The Back-end / 'Magic':

```
%let txtFld_tempInput = 37.75;
%let comboBox_fromTo = 2;

%include "temp_converter_pro.sas";
```

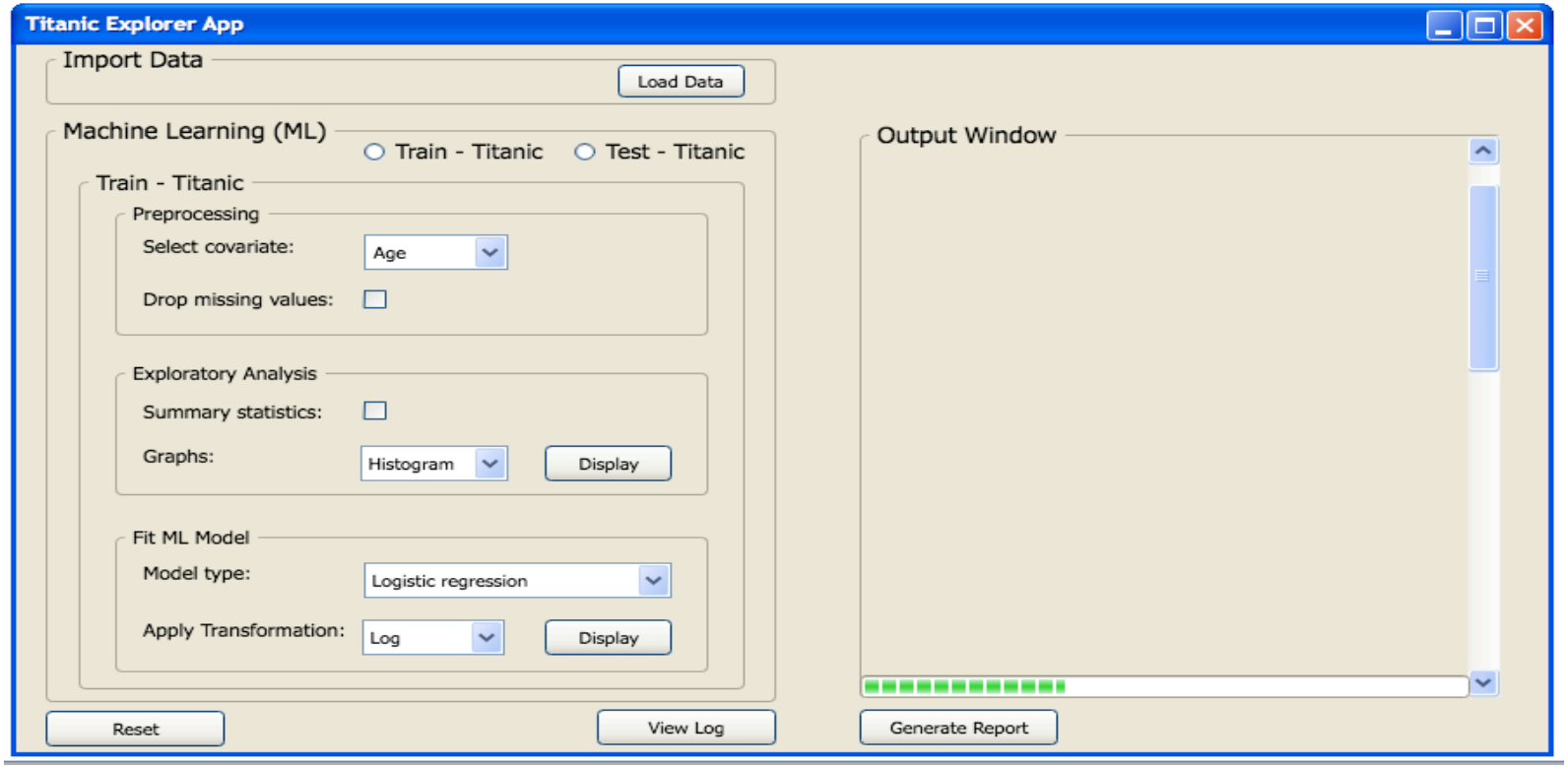
Case Study 3: Real-world Applications



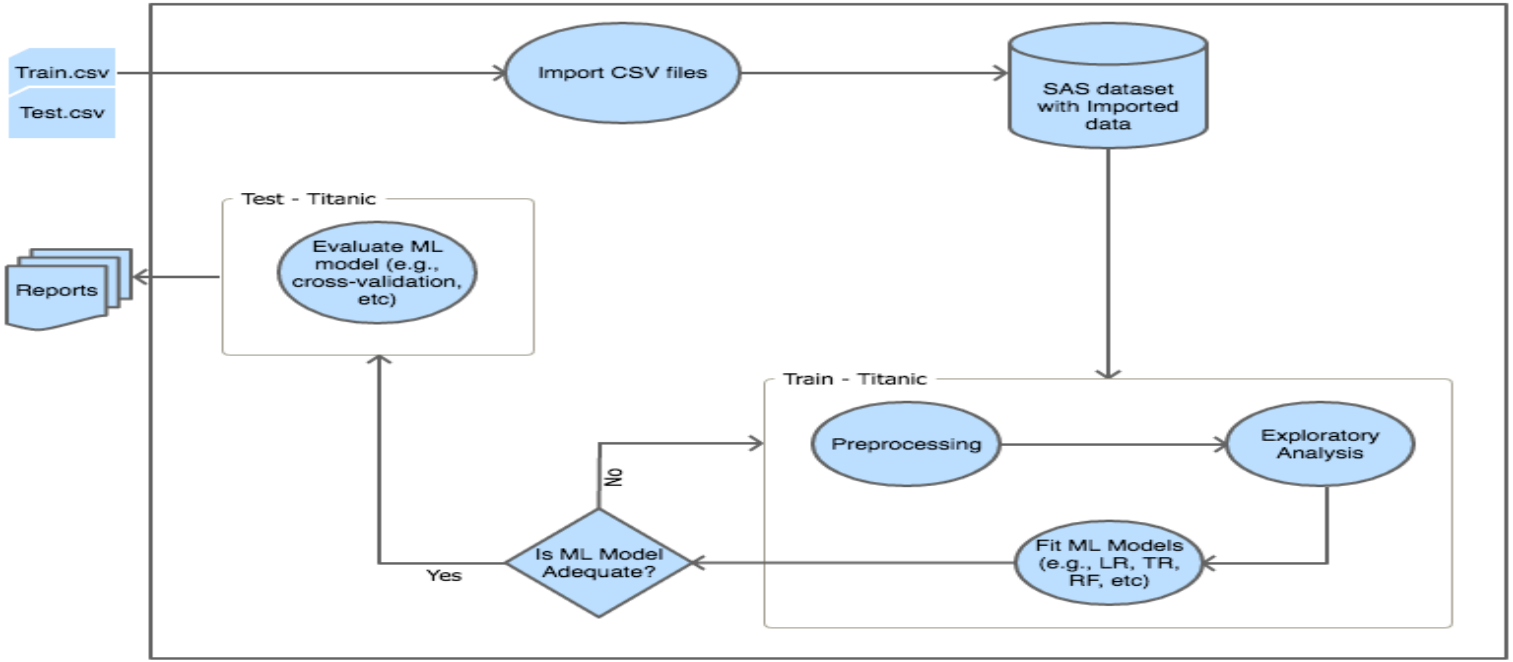
Titanic: Machine Learning from Disaster

- Titanic Dataset:
  - Kaggle provides two set of data sets: the Training set (train.csv) and the Test set (test.csv).
  - The dataset is about 2,224 travelers and crew who were on the ship (e.g., age, gender, #siblings, #parents, fare, etc)
  - The main difference between the two input files is the response variable (survived: 1=Yes; 0=No).
- Prizes for winners and job opportunity.

The Titanic Explorer App: **Front-end / Interface**



The Titanic Explorer App: **Process Flow Diagram**



The Titanic Explorer App: **Back-end / The 'Magic'**

```
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(...);*/
```

Conclusion

- Users are mainly interested on the following 4 tasks:
  - Loading study/research data.
  - Preprocess imported data.
  - Perform main tasks (e.g., explore, analyze, etc).
  - Generate a report for review/submission.
- Users are not often interested about the ‘Magic'.
- Building such application/add-in could improve User’s experience.
- SAS programs used for the back-end is platform independent.

Thank You!

Questions???