A Brief Comparison of the SAS and Python Languages

Vince DelGobbo (vincent.delgobbo@gmail.com)

April 20, 2022

Based on SAS/STAT User's Guide Example 104.6 - Chemical Reaction Response

https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/statug/statug_reg_examples06.htm

In [1]:
```
#  Import the module used to create DataFrame objects

import pandas as pd

#  Import the module used for the regression analysis

import statsmodels.api as sm
```

Sample Data - Import

SAS Code:

```
*  Create a global macro variable and then use it in the import;

%let XLSX_FILE=C:\temp\basug2022\reaction.xlsx;
```

In [2]:
```
#  Create a global string variable and then use it in the import

XLSX_FILE = 'c:\\temp\\basug2022\\reaction.xlsx'

# Or XLSX_FILE = r'c:\temp\basug2022\reaction.xlsx'
```

SAS Code:

```
*  Create the sample data (requires a SAS/ACCESS Interface to PC Files license);

proc import out=work.reaction
  file="&XLSX_FILE"
  dbms=xlsx
  replace;
run; quit;
```

In [3]:
```
#  Create the sample data

reaction = pd.read_excel(XLSX_FILE)

print('The "reaction" object is a DataFrame:', type(reaction))
```

```
The "reaction" object is a DataFrame: <class 'pandas.core.frame.DataFrame'>
```

Sample Data - Contents

SAS Code:

```
*  Briefly examine the data (included with your Base license);

title 'Sample Data - Contents';
proc contents data=work.reaction varnum; run; quit;
```

In [4]:
```
#  Briefly examine the data

reaction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   FeedRate            12 non-null     float64
 1   Catalyst            12 non-null     float64
 2   AgitRate            12 non-null     int64  
 3   Temperature         12 non-null     int64  
 4   Concentration       12 non-null     float64
 5   ReactionPercentage  12 non-null     float64
dtypes: float64(4), int64(2)
memory usage: 704.0 bytes
```

Sample Data - First 5 Records

SAS Code:

```
title 'Sample Data - First 5 Records';
proc print data=work.reaction(obs=5); run; quit;
```

In [5]:
```
reaction.head() # or head(5)
```

Out[5]:

	FeedRate	Catalyst	AgitRate	Temperature	Concentration	ReactionPercentage
0	10.0	1.0	100	140	6.0	37.5
1	10.0	1.0	120	180	3.0	28.5
2	10.0	2.0	100	180	3.0	40.4
3	10.0	2.0	120	140	6.0	48.2
4	15.0	1.0	100	180	6.0	50.7

Sample Data - Basic Statistics

SAS Code:

```
title 'Sample Data - Basic Statistics';
proc means data=work.reaction n mean std min p25 p50 p75 max; run; quit;
```

In [6]:
```
reaction.describe()
```

Out[6]:

	FeedRate	Catalyst	AgitRate	Temperature	Concentration	ReactionPercentage
count	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
mean	12.500000	1.500000	110.000000	160.000000	4.500000	41.658333
std	2.132007	0.426401	8.528029	17.056057	1.279204	9.657636
min	10.000000	1.000000	100.000000	140.000000	3.000000	28.500000
25%	10.000000	1.000000	100.000000	140.000000	3.000000	38.400000
50%	12.500000	1.500000	110.000000	160.000000	4.500000	40.000000
75%	15.000000	2.000000	120.000000	180.000000	6.000000	44.675000
max	15.000000	2.000000	120.000000	180.000000	6.000000	64.500000

In [7]:
```
#  Make the output resemble PROC MEANS output

reaction.describe().transpose()
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
FeedRate	12.0	12.500000	2.132007	10.0	10.0	12.5	15.000	15.0
Catalyst	12.0	1.500000	0.426401	1.0	1.0	1.5	2.000	2.0
AgitRate	12.0	110.000000	8.528029	100.0	100.0	110.0	120.000	120.0
Temperature	12.0	160.000000	17.056057	140.0	140.0	160.0	180.000	180.0
Concentration	12.0	4.500000	1.279204	3.0	3.0	4.5	6.000	6.0
ReactionPercentage	12.0	41.658333	9.657636	28.5	38.4	40.0	44.675	64.5

Sample Data - Regression Analysis

SAS Code:

```
*  Perform the regression analysis (requires a SAS/STAT license);

title 'Sample Data - Regression Analysis';
proc reg data=work.reaction;
  model  ReactionPercentage=FeedRate Catalyst AgitRate
                            Temperature Concentration;
  output out=work.reg predicted=PredictedReactionPercentage;
run; quit;
```

In [8]:
```
#  Create DataFrames with the dependent and regressor variables

y = reaction[['ReactionPercentage']]

X = reaction[['FeedRate', 'Catalyst', 'AgitRate', 'Temperature', 'Concentration']]

#  Add constant to regressor variables to include the intercept in the model

X = sm.add_constant(X)

#  Display the new regressors and note the constant that was added

print(X)
```

```
    const  FeedRate  Catalyst  AgitRate  Temperature  Concentration
0     1.0      10.0       1.0       100          140            6.0
1     1.0      10.0       1.0       120          180            3.0
2     1.0      10.0       2.0       100          180            3.0
3     1.0      10.0       2.0       120          140            6.0
4     1.0      15.0       1.0       100          180            6.0
5     1.0      15.0       1.0       120          140            3.0
6     1.0      15.0       2.0       100          140            3.0
7     1.0      15.0       2.0       120          180            6.0
8     1.0      12.5       1.5       110          160            4.5
9     1.0      12.5       1.5       110          160            4.5
10    1.0      12.5       1.5       110          160            4.5
11    1.0      12.5       1.5       110          160            4.5
```

In [9]:
```
#  Describe the model

model = sm.OLS(y, X)

#  Fit the model

results = model.fit()

#  Display the results

results.summary()

# Or sm.OLS(y,X).fit().summary()
```

```
C:\Users\vince\anaconda3\lib\site-packages\scipy\stats\stats.py:1603: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=12
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

Out[9]:

OLS Regression Results

Dep. Variable:	ReactionPercentage	**R-squared:**	0.965
Model:	OLS	**Adj. R-squared:**	0.936
Method:	Least Squares	**F-statistic:**	33.29
Date:	Wed, 20 Apr 2022	**Prob (F-statistic):**	0.000266
Time:	08:44:13	**Log-Likelihood:**	-23.569
No. Observations:	12	**AIC:**	59.14
Df Residuals:	6	**BIC:**	62.05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-43.6917	13.041	-3.350	0.015	-75.602	-11.782
FeedRate	1.6500	0.345	4.783	0.003	0.806	2.494
Catalyst	12.7500	1.725	7.392	0.000	8.530	16.970
AgitRate	-0.0250	0.086	-0.290	0.782	-0.236	0.186
Temperature	0.1625	0.043	3.769	0.009	0.057	0.268
Concentration	4.9667	0.575	8.639	0.000	3.560	6.373

Omnibus:	2.026	**Durbin-Watson:**	0.830
Prob(Omnibus):	0.363	**Jarque-Bera (JB):**	1.178
Skew:	-0.477	**Prob(JB):**	0.555
Kurtosis:	1.798	**Cond. No.**	3.62e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.62e+03. This might indicate that there are
strong multicollinearity or other numerical problems.