



# SAS and Python For Advanced Analytics: A Comparative Case Study

Doug Thompson, Director of Advanced Analytics, Rush  
Health, Chicago IL

Michigan SAS User Group, March 16, 2023



# Objectives

- While SAS was arguably the tool of choice for advanced analytics some years ago, Python has become an increasingly popular alternative
- Opinions on the relative merits of SAS and Python abound, but there have been few published comparative case studies
- Objectives of this presentation
  1. Compare SAS and Python in an illustrative advanced analytics study
  2. Provide a sense of the syntax, output and user experience of the two packages
  3. In this context, discuss similarities and differences, and summarize some thoughts on relative strengths and weaknesses



# Caveats

- Syntax *that works* is presented, but may not be the best or most efficient syntax
- The presenter has been a SAS user for 24 years and a Python user for 4 years – much more extensive SAS experience, no claim to be a deep Python expert
- Analyses were done for purposes of illustrating and comparing the two software packages – if the analyses themselves had been the emphasis, probably would have used some different techniques



# Supplemental Material Available on Request

- A more detailed paper on the material in this presentation, including more details of SAS and Python code, is available on request
- Another Python analysis that may be useful to SAS advanced analytic users with code is also available on request
  - Thompson D. Risk adjustment including social determinants of health: Insights from the Medical Expenditure Panel Survey. Poster presented at: Academy Health Annual Research Meeting (virtual due to COVID-19), June 14-17, 2021.
- E-mail: [doug\\_Thompson@rush.edu](mailto:doug_Thompson@rush.edu)



# Background

- ▶ Example case study
  - ▶ Examine the association between having a **primary healthcare provider** and **healthcare expenditures**
    - ▶ Primary healthcare provider coordinates all of a patient's healthcare (often called a primary care provider, abbreviated PCP)
  - ▶ **Question 1:** What personal characteristics explain having a primary healthcare provider vs. not having one?
    - ▶ Role of “gatekeeper-type” health insurance plan (e.g., HMO, MA, managed care plans)
  - ▶ **Question 2:** Does having a primary healthcare provider contribute to reduced healthcare expenditures, all else being equal?
    - ▶ The US federal government as well as commercial health insurance companies are heavily betting on this being the case





# Data



- ▶ **Medical Expenditure Panel Survey (MEPS)**
- ▶ Conducted annually since 1996 by the Agency for Healthcare Research and Quality
- ▶ Describes healthcare expenditures, healthcare utilization and health insurance among the U.S. non-institutionalized, non-military population.
- ▶ MEPS samples households. Information regarding each sampled household is collected for a 2-year period (“panel”) in 5 “rounds” of interviews spaced across 2.5 years. This enables longitudinal analysis of healthcare for individuals in the sampled households during the 2-year period covered in the panel.
- ▶ The data are freely available for download.
- ▶ This presentation used MEPS Panel 20, covering 2015 (“Y1”) and 2016 (“Y2”). For some measures, Round 2 (“R2”) is used to represent 2015 and Round 4 (“R4”) is used to represent 2016.
- ▶ Data were limited to Panel 20 who had health insurance in both 2015 and 2016, and who had data in both 2015 and 2016 (the latter condition was true of the vast majority of panel 20 participants; n = 14,422)



# Analytic Methods and Approach

## ► Analytic methods illustrated

1. Import data
2. Data manipulation (e.g., define measures, handle missing data)
3. Descriptive analyses
4. Statistical modeling – logistic regression (Analysis 1), ordinary least squares (OLS) regression (Analysis 2)

## ► Approach

- Parallel analyses of MEPS data were conducted using Python and SAS. The Python analyses were conducted first, then mirrored using SAS. The SAS analyses generally followed the same steps as Python, with some exceptions (e.g., output formatting).



# Some Terminology

- ▶ Python **data frame** like SAS **dataset**
- ▶ Python **library** like SAS **module** or product (e.g., Base, ETS, QC)
- ▶ Python **Jupyter Notebook** interface is in some ways similar to SAS **Enterprise Guide** (programming windows)



# Jupyter Notebook Interface

Files/Jupyter/ x meps\_usual\_care\_MSUG23 - Jupy x +

localhost:8888/notebooks/Files/Jupyter/meps\_usual\_care\_MSUG23.ipynb

Aktuelles Program...

jupyter meps\_usual\_care\_MSUG23 Last Checkpoint: a minute ago (autosaved) Python 3 (ipykernel) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

Analysis plan: Load MEPS 2 year longitudinal data Covered by public or private insurance throughout Y1 and Y2 (INSURCY1,INSURCY2) Had usual care provider -- Round 2 and 4 (HAVEUS2,HAVEUS4), and usual care provider is a person (PROVTY2,PROVTY4) who is an MD with primary care specialty, NP or PA (TYPEPE2,TYPEPE4)

Analysis 1: Factors associated with having USC in Round 4 -- expenditures in Y1 (TOTEXPY1), self-reported health (RTHLTH2,MNHLTH2), age (AGEY1X), family income (FAMINCY1), covered by managed care or gatekeeper plan (MCRPHOY1,MCDHMOY1,MCDMCY1,PRVHMOY1)

Analysis 2: Do those with usual care provider in Y1 have lower healthcare expenditures in Y2, adjusting for other factors? Adjustments use same factors as in analysis 1

```
In [2]: # Load MEPS data
import pandas as pd
import numpy as np

df = pd.read_sas(r'C:\Users\dt2t\OneDrive\Files\MEPS\h193.sas7bdat')
df.set_index(['DUPERSID'],drop=False,inplace=True)
```

```
In [3]: # Limit to yearind=1 (respondent in both 2015 and 2016) and panel=20 and
# had insurance in both years
df2 = df.loc[(df['YEARIND']==1) & (df['PANEL']==20) & (df['INSURCY1'].isin([-1,3,7])!=False) &
             (df['INSURCY2'].isin([-1,3,7])!=False)]

print('df shape:', df.shape)
print('df2 shape:', df2.shape)
# Check that values are being properly excluded
```

# Import Data

## Python

```
import pandas as pd
import numpy as np

df =
pd.read_sas('C:\projects\MEPS\h193.sas7bdat')
df.set_index(['DUPERSID'],drop=False,
inplace=True)
```

## SAS

```
libname mepsdat 'C:\projects\MEPS';

data df;
set mepsdat.h193;
df2_ind = (YEARIND=1 and PANEL=20 and
(INSURCY1 not in(-1,3,7)) and (INSURCY2 not
in(-1,3,7)));
run;
```

Note: Python inplace=True specifies that a permanent change will be made to data frame

# Select Sample

Python	SAS
<pre># Limit to yearind=1 (respondent in both 2015 and 2016) and panel=20 and # had insurance in both years  df2 = df.loc[(df['YEARIND']==1) &amp; (df['PANEL']==20) &amp; (df['INSURCY1'].isin( [-1,3,7])==False) &amp;               (df['INSURCY2'].isin([-1,3,7])==False)]</pre>	<pre>** Already defined in import; df2_ind = (YEARIND=1 and PANEL=20 and (INSURCY1 not in(-1,3,7)) and (INSURCY2 not in(-1,3,7)));  data df2; set df(where=(df2_ind=1)); run;</pre>

Notes: Python loc is used to access a group of rows or columns or a Boolean array. = assigns values from the right of operand to left of operand, while == checks to see if values to the right and left are equal.

# Define Study Groups

## Python

```
# Define having usual care providers in R2
and R4
# HAVEUS2=has usual care provider,
PROVTY2=usual care provider is person (as
opposed to facility), and
# TYPEPE2 is usual care provider is MD (family
med, internal med, peds) or NP or PA

df2['has_usc_R2'] = ((df2['HAVEUS2']==1) &
(df2['PROVTY2'].isin([2,3])) &
(df2['TYPEPE2'].isin([1,2,3,9,10]))) * 1
```

## SAS

```
data df2;
set df2;
has_usc_R2 = (HAVEUS2=1 and (PROVTY2
in(2,3)) and (TYPEPE2 in(1,2,3,9,10)));
has_usc_R4 = (HAVEUS4=1 and (PROVTY4
in(2,3)) and (TYPEPE4 in(1,2,3,9,10)));
run;
```

# Descriptives on Study Group

## Python

```
# Crosstab having usual care provider in R2 vs R4
```

```
print('What is the association of have_usc_R2 and have_usc_R4?', '\n',
```

```
pd.crosstab(df2['has_usc_R2'], df2['has_usc_R4'], margins=True), '\n')
```

```
# Of those with a usual care provider in R2, what percent also had a usual care provider in R4? i.e., compute row percentage
```

```
had_usc_r2 = df2.loc[(df2['has_usc_R2']==1)]
```

```
had_usc_r2 = had_usc_r2[['has_usc_R4']]
```

```
x=had_usc_r2.mean().astype(float).map("{:.2%}".format)
```

```
print('Percent with USC in R2 who also had USC in R4: ', x)
```

## SAS

```
ods rtf  
file='C:\projects\SAS_python_compare\crosstab_usucare.rtf';  
proc freq data=df2;  
tables has_usc_R2*has_usc_R4;  
run;  
ods rtf close;
```

# Output: Descriptives on Study Group

## Python

```
What is the association of have_usc_R2 and have_usc_R4?
  has_usc_R4    0    1    All
has_usc_R2
0              8423  1402  9825
1              1392  3205  4597
All            9815  4607  14422

Percent with USC in R2 who also had USC in R4:  has_usc_R4    69.72%
dtype: object
```

## SAS


Table of has_usc_R2 by has_usc_R4			
has_usc_R2	has_usc_R4		
Frequency			
Percent			
Row Pct			
Col Pct	0	1	Total
0	8423	1402	9825
58.40		9.72	68.13
85.73		14.27	
85.82		30.43	
1	1392	3205	4597
9.65		22.22	31.87
30.28		69.72	
14.18		69.57	
Total	9815	4607	14422
68.06		31.94	100.00





# Refer to Paper For Following


- Data preparation prior to regression modeling
  - Convert MEPS “don’t know,” “refused,” etc. (-1,-7,-8 in MEPS data) to missing
  - Impute median of non-missing for missing values (continuous variables)
  - Impute mode for missing values (categorical variables)
  - Variable names beginning with underscore indicate post-imputation variables (e.g., `_RTHLTH2` vs. `RTHLTH2`)
- Descriptive statistics for regression variables vs. study group variables



# Analysis 1: Which Personal Characteristics Are Associated With Having a PCP Next Year?

## ► Hypotheses

1. Individuals with gatekeeper-type insurance plans will be more likely to have a PCP
2. Older and sicker individuals will be more likely to have a PCP
3. Higher-income individuals will be more likely to have a PCP due to greater access to preventive services



# Analysis 1: Which Personal Characteristics Are Associated With Having a PCP Next Year?

- ▶ **Outcome variable:** Respondent has PCP in 2016, yes (=1) or no (=0) (has\_usc\_R4)
- ▶ **Main predictor of interest:** Whether or not respondent was in a gatekeeper-type health insurance plan in 2015 (mgd\_care\_ins\_R2)
- ▶ **Control variables:**
  - ▶ Self-reported health in 2015 (\_RTHLTH2)
  - ▶ Self-reported mental health in 2015 (\_MNHLTH2)
  - ▶ Respondent's total healthcare expenditures in 2015 (\$10Ks) (\_TOTEXPY1\_10k)
  - ▶ Respondent's age (\_AGEY1X)
  - ▶ Respondent's household income in 2015 (\$10Ks) (\_FAMINCY1\_10k)

# Analysis 1: Logistic Regression

## Python

```
import statsmodels.api as sm

# re-scale the $ variables because otherwise
# the coefficients are very small
df3['_TOTEXPY1_10k'] = df3['_TOTEXPY1']/10000
df3['_FAMINCY1_10k'] =
df3['_FAMINCY1']/10000

df3['intercept'] = 1

logit_model =
sm.Logit(df3['has_usc_R4'],df3[['intercept','_RT
HLTH2','_MNHLTH2','_TOTEXPY1_10k','_AGEY1X',
'_FAMINCY1_10k','mgd_care_ins_R2']])
result = logit_model.fit()

print(result.summary())
```

## SAS

```
data df2d;
set df2c;
_TOTEXPY1_10k=_TOTEXPY1/10000;
_FAMINCY1_10k=_FAMINCY1/10000;
run;

ods rtf
file='C:\projects\SAS_python_compare\logisti
c_compare1.rtf';
proc logistic data=df2d descending;
model has_usc_R4 = _RTHLTH2 _MNHLTH2
_TOTEXPY1_10k _AGEY1X _FAMINCY1_10k
mgd_care_ins_R2;
run;
ods rtf close;
```

# Logistic Regression Results: Python

Optimization terminated successfully.

Current function value: 0.609358

Iterations 5

## Logit Regression Results

```
=====
Dep. Variable:          has_usc_R4   No. Observations:          14422
Model:                  Logit        Df Residuals:              14415
Method:                  MLE         Df Model:                   6
Date:                   Sat, 11 Mar 2023   Pseudo R-squ.:            0.02728
Time:                   15:46:23         Log-Likelihood:           -8788.2
converged:              True          LL-Null:                   -9034.6
Covariance Type:       nonrobust       LLR p-value:              2.768e-103
=====
```


	coef	std err	z	P> z	[0.025	0.975]
intercept	-1.3942	0.058	-24.018	0.000	-1.508	-1.280
_RTHLTH2	-0.0094	0.023	-0.406	0.685	-0.055	0.036
_MNHLTH2	-0.0023	0.023	-0.097	0.923	-0.048	0.044
_TOTEXPY1_10k	0.0631	0.015	4.322	0.000	0.035	0.092
_AGEY1X	0.0150	0.001	17.962	0.000	0.013	0.017
_FAMINCY1_10k	0.0148	0.003	4.959	0.000	0.009	0.021
mgd_care_ins_R2	-0.0798	0.037	-2.149	0.032	-0.153	-0.007

```
=====
```

# Logistic Regression Results: SAS

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-1.3942	0.0580	576.8483	<.0001
_RTHLTH2	1	-0.00945	0.0233	0.1644	0.6851
_MNHLTH2	1	-0.00226	0.0234	0.0093	0.9230
_TOTEXPY1_10k	1	0.0631	0.0146	18.6833	<.0001
_AGEY1X	1	0.0150	0.000838	322.6157	<.0001
_FAMINCY1_10k	1	0.0148	0.00298	24.5925	<.0001
mgd_care_ins_R2	1	-0.0798	0.0371	4.6198	0.0316





# Analysis 1 Summary

- ▶ As hypothesized, older, sicker and higher income individuals in 2015 were more likely to have a PCP in 2016
- ▶ Contrary to hypothesis, individuals with gatekeeper-type insurance plans in 2015 were less likely to have a PCP in 2016 (even after adjusting for differences in age, income and health)
- ▶ Logistic regression in SAS and Python yielded identical results



## Analysis 2: Is Having a PCP Associated With Healthcare Expenditures Next Year?

### ► Hypotheses

1. Respondents with a PCP in 2015 will have lower healthcare expenditures in 2016
2. Younger, healthier individuals who spent less on healthcare in 2015 will spend less on healthcare in 2016
3. Household income will be important (although direction is ambiguous – higher income individuals may have greater ability to spend on healthcare, but their greater access to preventive services may be associated with reduced healthcare expenditures)



# Analysis 2: Is Having a PCP Associated With Healthcare Expenditures Next Year?

- ▶ **Outcome variable:** Respondent's total healthcare expenditures in 2016 (TOTEXPY2)
- ▶ **Main predictor of interest:** Whether or not the respondent had a PCP in 2015 (has\_usc\_R2)
- ▶ **Control variables:**
  - ▶ Self-reported health in 2015 (\_RTHLTH2)
  - ▶ Self-reported mental health in 2015 (\_MNHLTH2)
  - ▶ Respondent's total healthcare expenditures in 2015 (\$10Ks) (\_TOTEXPY1\_10k)
  - ▶ Respondent's age (\_AGEY1X)
  - ▶ Respondent's household income in 2015 (\$10Ks) (\_FAMINCY1\_10k)
  - ▶ Whether or not respondent was in a gatekeeper-type health insurance plan in 2015 (mgd\_care\_ins\_R2)



# Descriptive Statistics of Analysis Variables

- Bivariate associations between the outcome variable and each analysis variable were examined prior to regression modeling
- Python: Used plotting functions in Python library MATPLOTLIB
- SAS: PROC SGPLOT
- See paper for syntax of each

# Custom Functions in Python

- ▶ Similar to SAS macros

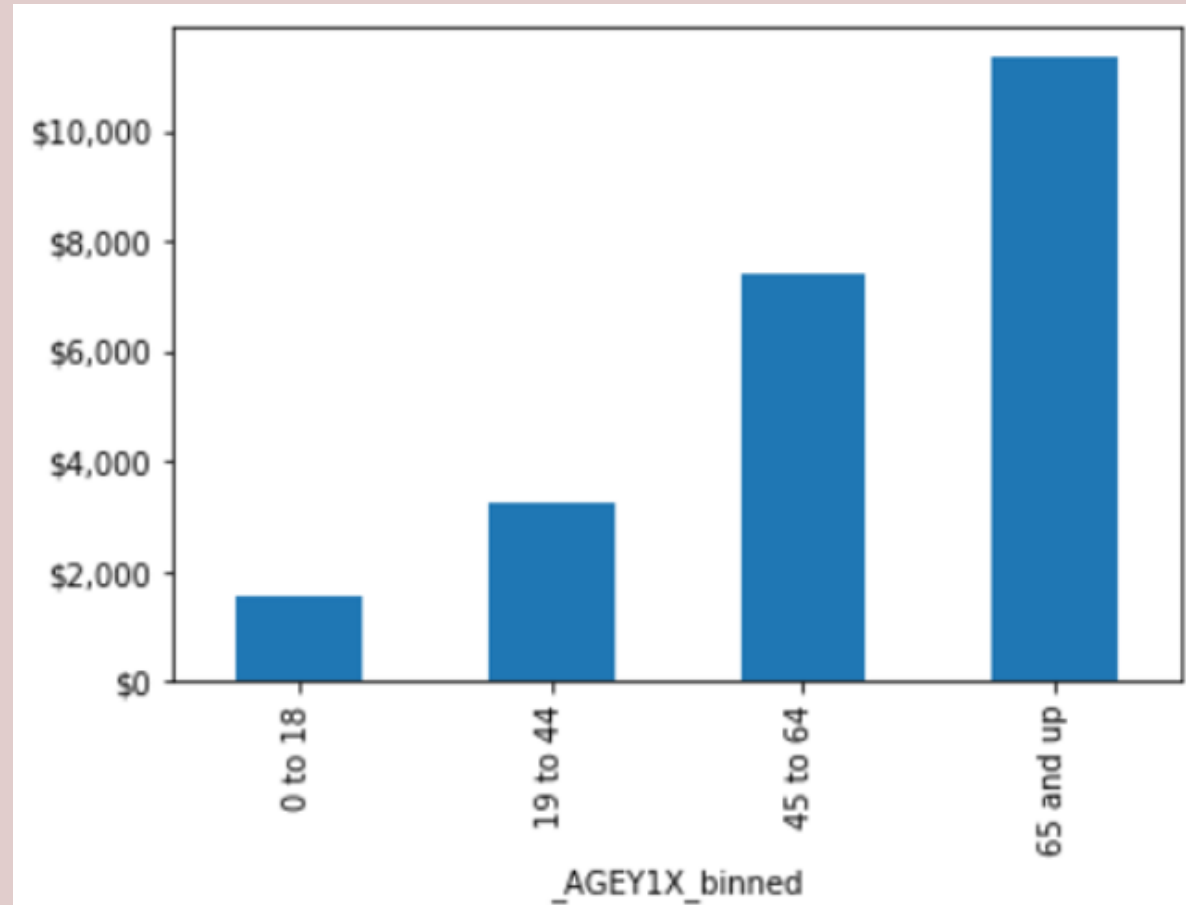
## Python example

```
def exp2_byvar(byvar):
    print('byvar is:',byvar,'\n',pd.value_counts(df4[byvar]))
    temp = df4['TOTEXPY2'].groupby(df4[byvar]).mean()
    print(temp)
    plt.figure();
    ax = temp.plot.bar()
    ax.yaxis.set_major_formatter(FuncFormatter(lambda y, _: '${0:,.0f}'.format(y)))
    print('\n')

byvars =
['_TOTEXPY1_binned','_FAMINCY1_binned','_AGEY1X_binned','_RTHLTH2','_MNHLTH2',
'mgd_care_ins_R2','has_usc_R2']
for var in byvars:
    exp2_byvar(var)
```

# Custom Functions in Python (Cont'd)

Python example -- output





# Some Data Manipulation Prior to Modeling

## Python

```
# Log transform Y2 expenditures  
df4['ln_TOTEXPY2'] = np.log(df4['TOTEXPY2']+1)  
  
df4['intercept'] = 1
```

## SAS

```
data df2e;  
set df2d;  
ln_TOTEXPY2 = log(TOTEXPY2+1);  
run;
```

# Regression Modeling

## Python

```
# Log transformed Y2 spend
ols_In_model =
sm.OLS(df4['ln_TOTEXPY2'],df4[['intercept',
'_RTHLTH2','_MNHLTH2','_TOTEXPY1_10k',
'_AGEY1X',
'_FAMINCY1_10k','mgd_care_ins_R2',
'has_usc_R2']])

result_In = ols_In_model.fit()
print('*** Model of log-transformed Y2 spend
***', '\n', result_In.summary(), '\n')
```

## SAS

```
ods rtf
file='C:\projects\SAS_python_compare\spen
d_regression.rtf';
proc reg data=df2e;
model TOTEXPY2 = _RTHLTH2 _MNHLTH2
_TOTEXPY1_10k _AGEY1X _FAMINCY1_10k
mgd_care_ins_R2 has_usc_R2 / vif;
run;
quit;
ods rtf close;
```


# OLS Regression Results: Python

```
*** Model of log-transformed Y2 spend ***
                        OLS Regression Results
=====
Dep. Variable:          In_TOTEXPY2    R-squared:                0.183
Model:                  OLS           Adj. R-squared:           0.182
Method:                 Least Squares  F-statistic:              459.7
Date:                   Sun, 19 Jan 2020  Prob (F-statistic):       0.00
Time:                   14:14:32      Log-Likelihood:          -35245.
No. Observations:      14422         AIC:                     7.051e+04
Df Residuals:          14414         BIC:                     7.057e+04
Df Model:               7
Covariance Type:       nonrobust
=====
                        coef    std err          t      P>|t|      [0.025    0.975]
-----
intercept              3.6548     0.073     49.981     0.000     3.511     3.798
_RTHLTH2               0.2702     0.030     8.959     0.000     0.211     0.329
_MNHLTH2               0.0697     0.030     2.305     0.021     0.010     0.129
_TOTEXPY1_10k         0.4598     0.018    25.049     0.000     0.424     0.496
_AGEY1X                0.0302     0.001    27.909     0.000     0.028     0.032
_FAMINCY1_10k         0.0335     0.004     8.541     0.000     0.026     0.041
mgd_care_ins_R2       -0.0615     0.047    -1.298     0.194    -0.154     0.031
has_usc_R2             0.6611     0.050    13.106     0.000     0.562     0.760
=====
```

# OLS Regression Results: SAS

Root MSE	2.78754	R-Square	0.1825
Dependent Mean	6.09194	Adj R-Sq	0.1821
Coeff Var	45.75794		

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr >  t	Variance Inflation
Intercept	1	3.65483	0.07312	49.98	<.0001	0
_RTHLTH2	1	0.27019	0.03016	8.96	<.0001	1.98025
_MNHLTH2	1	0.06967	0.03023	2.30	0.0212	1.76616
_TOTEXPY1_10k	1	0.45985	0.01836	25.05	<.0001	1.10259
_AGEY1X	1	0.03018	0.00108	27.91	<.0001	1.22746
_FAMINCY1_10k	1	0.03349	0.00392	8.54	<.0001	1.07288
mgd_care_ins_R2	1	-0.06148	0.04737	-1.30	0.1944	1.02742
has_usc_R2	1	0.66115	0.05045	13.11	<.0001	1.02566



# Analysis 2 Summary

- As hypothesized, older individuals who spent more on healthcare in 2015 tended to spend more on healthcare in 2016
- As expected, income in 2015 was associated with healthcare expenditures in 2016
- Contrary to hypothesis
  - Individuals who rated themselves as healthier in 2015 spent more on healthcare in 2016
  - Individuals with a PCP in 2015 spent more on healthcare in 2016
- OLS regression in SAS and Python yielded identical results



# SAS and Python: Similarities

1. Both are relatively easy to use – fairly advanced analytics can be conducted after writing a few dozen to a couple hundred lines of code.
2. Both are roughly close to the English language, thus fairly interpretable when reading the code.
3. Both are good at dealing with structured tabular data – importing, subsetting or “slicing,” defining and transforming variables, and joining tables can all be done fairly easily.
4. Both exhibited good capabilities for regression modeling and yielded the same statistical estimates.
5. Both are good at looping through lists of variables and executing operations on each variable within a list.
6. Both have simple and attractive plotting capabilities (in the past, this was not so much the case for SAS, but SAS’s capabilities and ease of use for plotting have increased since the introduction of PROC SGPLOT).
7. The Jupyter Notebook interface, which was used for the Python analyses described in this paper, has a similar look-and-feel to the programming window interface of SAS Enterprise Guide.
8. Both Python and SAS gave informative error messages, while both gave warning messages that can be ambiguous (this may just be the nature of warning messages).





# SAS and Python: Differences

1. In some cases the Python syntax seemed a little more “wordy” with more typing than SAS, while in other cases the SAS syntax seemed more wordy – this seems to be situation-specific.
2. Python was excellent at flexibly imputing missing data (e.g., median, mode); to do the same operations in SAS would likely be more complicated.
3. Python has excellent capabilities for producing annotated output, mixing output tables with descriptive language using print statements. SAS’s capabilities for this are a bit more clunky, for example, using title statements for PROCs and writing to the log, neither of which seems as nice as Python’s capabilities in this aspect.
4. Some of the code seemed to run a bit slower in Python than in SAS, although the difference was not drastic and may be situation-specific.
5. SAS’s regression modeling procedures have excellent default output; the same can be produced from Python/statsmodels, but seems to require more coding.



# Concluding Thoughts

## ➤ SAS

- Commercial software; although SAS is not cheap, most organizations can afford some PC SAS 9.4 licenses at a minimum
- It has been around for a long time and is considered to be very accurate and reliable, including by the U.S. Federal Government. CMS still publishes models and data in SAS. Anecdotally, based on a recent conversation with an analyst who does a lot of FDA work, although FDA does not require submissions to be in SAS, FDA staff will often check results using SAS, given that SAS has been used for a long time for FDA submissions and is considered very credible.

## ➤ Python

- As open source code, it is available free of charge
- Has a large analytics user community that is already much larger than the SAS analytics user community, and growing
- New analytic capabilities may be available earlier in Python