Stata Python Rosetta Stone

Side-by-side code examples v 1.0

Prepared by Adam Ross Nelson JD PhD @adamrossnelson

Recommended Stata Setup

```
cls                                 // Clear the screen
set more off                        // Disable 'More' prompt
clear all                           // Clear memory
capture log close                   // Close open logs
log using my_logfile.txt, text      // Begin new log file
. . .
log close                           // Close log flie
```

Recommended Python Setup

```
from sfi import Data                # Stata's Pyhon API
import pandas as pd                 # A Popular DataFrame module
import numpy as np                  # A Popular scientific module

# Display parameters that will approximate Stata output
pd.set_option('display.max_columns', None)
pd.set_option('display.expand_frame_repr', False)

# Store dataset variable names for later reference
vars = [Data.getVarName(x) for x in range(0,Data.getVarCount())]

# Store dataset in a Pandas dataframe
df = pd.DataFrame(Data.getAsDict(valuelabel=True,
                                 missingval=np.nan))
```

	Stata	Python	Notes & Additional Options

List Observations	`// List first five observations`
	`list in 1/5`
	
	`// List specific variables in first 5 obs`
	`list make weight in 1/5`
	
	`// List last 5 observations`
	`list in -5/-1`
	
	`// List specific variables in last 5 obs`
	`list make weight in -5/-1`
		`# Using Stata's Python API`
		`Data.list(obs=range(0,5))`
		
		`# Stata's API - specific vars in first 5 obs`
		`Data.list('make weight', obs=range(0,5))`
		
		`# Using Pandas dataframe`
		`df.head()`
		
		`# Using Pandas dataframe, specific vars`
		`df[['make','weight']].head()`
			`# Last 5, Using Pandas dataframe`
			`df.tail()`
			
			`# Last 5, Pandas, specific vars`
			`df[['make','weight']].tail()`

Describe / Inspect Data	`// Describe the dataset in memory`
	`describe`
	
	`// Describe can be abbreviated`
	`desc`
	
	`// Save output as a dataset`
	`describe, replace`
	
	`// Summary statistics`
	`summarize`
		`# Using Stata's Python API (With a loop)`
		`for var in vars:`
		`  print('{:18}{:12}{}'.format(var, Data.getVarType(var), Data.getVarLabel(var)))`
		
		`# Using Pandas dataframe`
		`df.info()`
		
		`# Combine Stata's Python API, list comprehension, and Pandas to output as a dataset`
		`pd.DataFrame({'Variable Name':vars,`
		`  'Data Type':[Data.getVarType(v) for v in vars],`
		`  'Variable Label':[Data.getVarLabel(l) for l in vars]})`
		
		`# Summary statistics`
		`df.describe().transpose()`	

Merge Datasets	`// Load example dataset (make, price, mpg)`
	`use http://www.stata-press.com/data/r15/autoexpense.dta, clear`
	
	`// Merge dataset (make, weight, length)`
	`merge 1:1 make using http://www.stata-press.com/data/r15/autosize.dta`
		`# Load each dataset`
		`expns_df = pd.read_stata('http://www.stata-press.com/data/r15/autoexpense.dta')`
		`sizes_df = pd.read_stata('http://www.stata-press.com/data/r15/autosize.dta')`
		
		`# Load each dataset`
		`df = pd.merge(expns_df, sizes_df, on='make', how='outer', indicator=True)`
		
		`# Replicate Stata's output with the value_counts() method and the _merge indicator`
		`df['_merge'].value_counts()`	

Data Cleaning	`// Generate new text variable`
	`gen newtxt = "Some text here"`
	
	`// Transform continuous to binary`
	`gen isExpensive = price > 3000`
	
	`// Transform linear to quadratic`
	`gen price2 = price * price`
		`# Generate new text variable using Stata's API`
		`Data.addVarStr('newtxt', 20)`
		`Data.store('newtxt', None, ['Some text here'] * Data.getObsTotal())`
		
		`# Generate new text, Combine Stata's API with Pandas`
		`df['newtxt'] = ['Some text here']`
		`Data.store('newtxt', None, df['newtxt'])`
		
		`# Transform continuous to binary Combine Stata's API with Pandas`
		`Data.addVarByte('isExpensive')`
		`df['isExpensive'] = [1 if p > 4000 else 0 for p in df['price']]`
		`Data.store('isExpensive', None, df['isExpensive'])`
		
		`# Transform linear to quadratic`
		`Data.addVarInt('price2')`
		`df['price2'] = df['price'].apply(lambda p: p * p)`
		`Data.store('price2', None, df['price2'])`	

Additional Notes & Resources

Environmental Setup

The examples in this guide repeatedly depend on executing the provided recommended setup. For example, for quick reference the recommended setup stores a list of Stata's variable names in a Python list called `vars`.

The recommended setup also stores the dataset in a Pandas dataframe called `df`. Using a Python variable called `df` is a widely accepted convention when working with Pandas.

Log Files

A convention common among Stata's user communities is to save output to a log file. In Stata, saving output to a log file is a simple matter using the log command and its options. Conveniently, Python provides no similar options. When using Python through Stata's UI, Python's output will save along with Stata's output.

Variable Name References & Assumptions

This guide's examples also rely on an assumption that the `auto.dta` dataset is in memory. Example code that references specific variable names (e.g. `make`, `weight`, `price`, etc.), will not work with other datasets which do not include those variable names.

Merging Datasets

By default Stata performs what Pandas would refer to as an `outer` merge. Meaning "use union of keys from both frames, similar to a SQL full outer join; sort keys lexicographically." [1]. the result will include all records from both datasets.

The default in Pandas performs an `inner` merge which means "use intersection of keys from both frames, similar to a SQL inner join; preserve the order of the left keys." [2]; the result will only include records that matched in both datasets.

For Stata users looking to replicate Stata's behavior on a merge operation it is necessary to specify the `how='outer'` argument in the `pd.merge()` statement.

Stata's API

Most tasks that can be accomplished in Stata using one line of code can also be accomplished in Python one line of code. These examples provide additional lines of code. that leverage Stata's API to move data from the Python environment back into Stata's environment.

Adam Ross Nelson