XVI Convegno Italiano degli Utenti di Stata - Firenze, 26-27 September, 2019

Calling Python Scripts in Stata: a Power-Law application

Antonio Zinilli
National Research Council (CNR)

IRCrES

Aim

Stata allows to call external routines, written in other software, to perform specific tasks within Stata.

This talk offers some insights on how to develop a Stata procedure embedding an external software routine (Python, in this case).

We provide a user-written Stata module python, written to allow users to run scripts by calling back Python programming language.

IRCrES

Motivation

The -python- package facilitates integrating Python with Stata by allowing automatic interprocess communication between the two software. Here we present a statistical application implemented in Python and called in Stata for discerning and quantifying power-law behavior in empirical data.

In a nutshell

- Call Python from Stata
- Output Python results within Stata
- Invoke Python interactively in do-files and ado-files
- Execute a Python script file(.py) directly through Stata

IRCES

3

Motivation

Mix Stata code and Python code in a single do-file. We can take advantage of Stata's data management capabilities and packages implemented in Python (and not yet in Stata).

There are things we can do in Python that would be easier in Stata (e.g. graphs, descriptive analysis etc.).

Run multiple Python statements.

IRCES

4

Case study: a Power-Law application

Data: citation count of the 1120 top-cited physicists from 1998-2014 (Source: Scopus)

Power Law function implements both the discrete and continuous maximum likelihood estimators for fitting the power-law distribution to data, along with the goodness-of-fit based approach to estimating the lower cut-off for the scaling region.

The probability $P(x)$ that the citations decays with the law:

$$P(x) \propto x^{-\alpha}$$

Where P(x) is the probability to encounter value x and α as the scaling exponent ($\alpha \succ 0$)

IRSTEA

5

Functional form by distribution

Distribution	f(x)
Log-normal	$\frac{1}{\sqrt{2\pi}^{\sigma x}} e^{-(\ln x - \ln x_0)^2/(2\sigma^2)}$
Exponential	$e^{-\lambda x}$
Stretched exponential (Weibull)	$e^{-x^{\beta}}$
Power-law + cut off	$x^{-\alpha} e^{-\lambda x}$

IRSTEA

6

Intro to Python from Stata

Download the latest version of Python through Anaconda

It is the standard platform for Python data science

python search ← Search for Python installations on the current system

set python_exec "C:\Users\Zinilli\Anaconda3\python.exe", permanently

Set which version of Python to use

Python: ← Enter Python interactive environment

Issuing python (without a colon) will allow you to remain in the Python environment

IRCES

7

Procedure for Power Law function

```
clear all
cd "C:\Users\Zinilli\Dropbox\Sug2019"
import excel "C:\Users\Zinilli\Dropbox\Sug2019\Citations.xlsx", sheet("Foglio1") firstrow
sum Citations


python:
from __future__ import division
import powerlaw
import numpy
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
[pip install package]

data=np.array(Data.get("Citations"))

results = powerlaw.Fit(data,fit_method='Likelihood',discrete=True,estimate_discrete=True)
```

Download the libraries

Additional package that is not part of the standard library

Function Power Law

IRCES

8

Procedure for Power Law function

```
results.fixed_xmin
results.power_law.alpha
```

The search for the optimal xmin

α of the power law and its standard error, σ, are both calculated using the maximum likelihood method

```
fit.power_law
fit.power_law.alpha
fit.power_law.parameter1
fit.lognormal.mu
fit.lognormal.parameter1
fit.exponential.parameter1
R,p=fit.loglikelihood_ratio('power_law', 'lognormal',normalized_ratio=True)
print ( R, p)
R,p=fit.loglikelihood_ratio('power_law', 'exponential',normalized_ratio=True)
print ( R, p)
R,p=fit.loglikelihood_ratio('power_law', 'truncated_power_law',nested=True)
print ( R, p)
R,p=fit.loglikelihood_ratio('power_law', 'stretched_exponential', nested=True)
print ( R, p)
```

Within the Fit object there are individual distribution objects for different distributions.
Each Distribution has the best fit parameters for that distribution.

R is the loglikelihood ratio between the two candidate distributions. This number will be positive if the data is more likely in the first distribution, and negative if the data is more likely in the second distribution.
The significance value for that direction is p.

IRCES

9

Procedure for Power Law function

```
fig1 = fit.plot_ccdf(linewidth=3,label='Empirical Data')
fit.power_law.plot_ccdf(ax=fig1, color='black', linestyle='-.',label='Power Law', linewidth=0.8)
fit.lognormal.plot_ccdf(ax=fig1, color='g', linestyle='--',label='Lognormal')
fit.exponential.plot_ccdf(ax=fig1, color='y', linestyle='--',label='Exponential',linewidth=0.8)
fit.truncated_power_law.plot_ccdf(ax=fig1, color='m', linestyle=':', survival=True, label='Power Law + cut-off', linewidth=2)
fit.stretched_exponential.plot_ccdf(ax=fig1, color='b', linestyle='--',label='Stretched_exponential',linewidth=0.8)
fig1.set_ylabel(r"$p(X\geq x)$")
fig1.set_xlabel(r"Degree")
plt.xlabel("Strenght Distribution --> Black = Empirical Data")
handles, labels = fig1.get_legend_handles_labels()
fig1.legend(handles, labels, loc=3, fontsize=9)
plt.title("Comparison of Power Law with Other Fits")
plt.savefig("powerlaw.png")
```

```
end
```

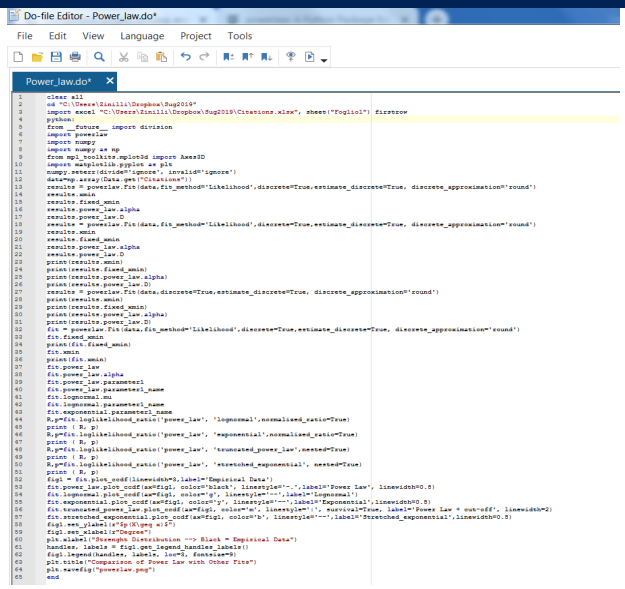
Closing Python environment

Via the library matplotlib. Function to create a single plot.

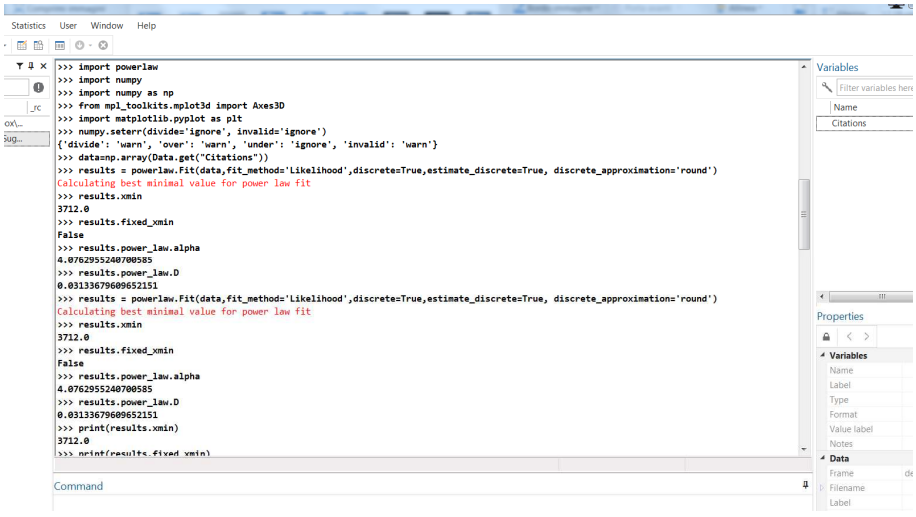
IRCES

10

Screenshot



Python output visible as Stata output

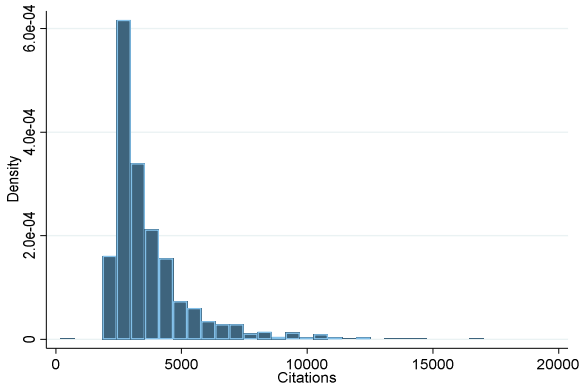


Descriptive analysis

```
. sum Citations
```

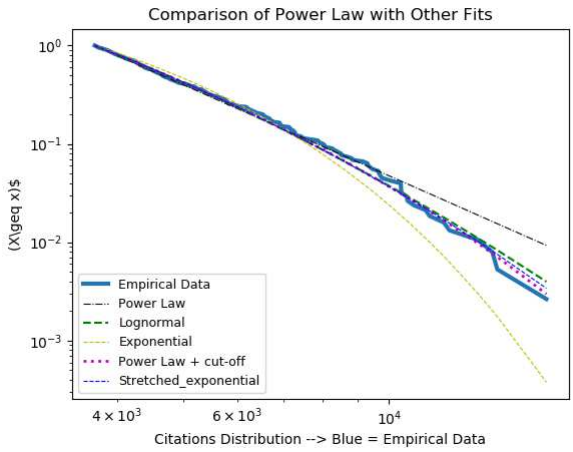
Variable	Obs	Mean	Std. Dev.	Min	Max
Citations	1,119	3705.532	1690.558	178	16994

```
. twoway (histogram Citations)
```



13

Complementary cumulative distribution functions of example data and fitted distributions



14

Power-law and alternative distributions

alpha= 4
xmin=3712

$$P(x) \propto x^{-\alpha}$$

	Power law	Log-normal		Exponential		Stretched exponential		Power-law + cut off	
Citations	p*	LR	p*	LR	p*	LR	p*	LR	p*
	0,03	-1.32	0,18	0,72	0,47	-2,23	**0,03**	-2,55	**0,02**

IRSTES

15

Conclusions

Summarizing, we cannot conclude on the exact nature of the far-tail of distributions of citations.

Power Law with cut off and Stretched exponential seems to be both good, we need further analysis to investigate the best.

IRSTES

16


thank you
danke
gracias
merci
obrigado
dziękuję
teşekkür ederim
ngiyabonga
спасибо
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
grazie
arigatô
dakujem
terima kasih
tapadh leat
bedankt
hvala
dank je
takk
taiku
sagolun
mauruuru
мерси
謝謝
감사합니다
ありがとう
شكراً جزيلاً

17