

EViews から Python を実行する

はじめに

- Python のパッケージとコードを EViews で実行することができます。
- EViews を使って EViews のデータを Python 環境に送り、Python の機能を実行し、得られたデータを簡単なコマンドで EViews に返します。
- EViews 11 は、Python 2 (version 2.7.15 以上) と Python 3 (version 3.6.5 以上)をサポートしています。

Python を実行する

- Python に関する **xopen** オプションの構文

`XOPEN(p)`

または

`XOPEN(type=p)`

で Python を起動します。

- EViews は、下記の Python のデータ形式をサポートしています。
 - ・ list
 - ・ tuple
 - ・ dictionary
 - ・ `numpy.ndarray` ※numpy パッケージのインストールが必要
 - ・ `pandas.series` ※pandas パッケージのインストールが必要
 - ・ `pandas.dataframe` ※pandas パッケージのインストールが必要

- Python に関する **xput** コマンドの構文

`xput(ptype=list|tuple|dictionary|ndarray|series|dataframe)`

使用例 マルチレベル混合効果モデル

- 例題では、EViews 内で Python を呼び出しマルチレベル混合効果モデルをフィットさせます。
- EViews のワークファイルにあるデータを Python に送信し、Python でモデルを推定します。
- 最終的に、推定した結果を Python から EViews に戻し、ワークファイルオブジェクトとして保存します。

マルチレベル混合効果モデル

- 複数の個体を繰り返し観測したデータによる、ランダムスロープを持つマルチレベル混合効果モデルは、下記の方程式で表すことができます。

$$y = X\beta + Zu + \epsilon$$

- y は $n \times 1$ の応答ベクトル、 X は固定効果に対する $n \times p$ の共分散行列、 Z はランダム効果に対する $n \times q$ の共分散行列です。
- ϵ は多変量正規分布に従い、平均 0 、分散行列 $\sigma_\epsilon^2 R$ です。
- u の共分散行列を G とし、 u と ϵ は直交していると仮定します。

$$\text{Var} \begin{bmatrix} u \\ \epsilon \end{bmatrix} = \begin{bmatrix} G & 0 \\ 0 & \sigma_\epsilon^2 R \end{bmatrix}$$

事前準備

- Python 3 をインストールします。本マニュアルの解析では、Python version 3.6.8 を使用しています。

▼Python

<https://www.python.org/downloads/release/python-368/>

- 64bit 版 Windows の場合、Python は通常下記のディレクトリにインストールされます。

C:\Users\¥USER_NAME¥AppData\Local¥Programs¥Python¥Python36_64

- 上記のフォルダに移動して、下記のコマンドを実行します。

```
python -m pip install --upgrade pip
python -m pip install PACKAGE_NAME
```

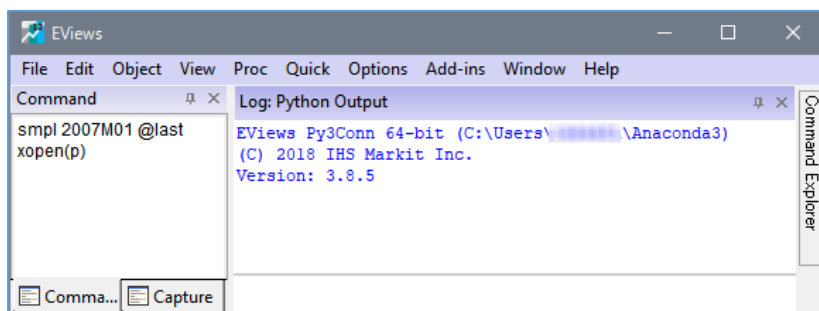
- EViews のオプションで Python へのパスが指定されていることを確認します。EViews のメニューで Options>General Options...を選択し、左のツリーから External program interface を開きます。そして、Home Path で Python がインストールされているディレクトリを指定します。通常は EViews がディレクトリを自動検出するため、設定を変更する必要はありません。
- 以下の Python パッケージをインストールします。
 - ・ NumPy
 - ・ Pandas
 - ・ matplotlib
 - ・ seaborn
 - ・ statsmodels

Python でのデータ分析

- まずは例題ワークファイル `pig.wf1` を EViews で開き、データを Python に送ります。このサンプルファイルは、48 匹の豚の 9 週間分の体重の変化を記録したデータです。EViews の Command 欄で下記を実行して、Python を呼び出します。

```
xopen(p)
```

- Log: Python Output ウィンドウが開き、Python と双方向のやり取りができるようになります。このウィンドウでは、Python インスタンスのコマンドプロンプトに表示されるようなコマンド実行結果を、直接確認することができます。EViews の Command ウィンドウからは、Python にコマンドを送信できます。



- EViews の Command ウィンドウで次のコマンドを入力、または Python ウィンドウに直接 Python のコードを入力し、必要な Python パッケージをインポートします。EViews の xrun コマンドは xrun 以降の文を既に開かれている COM アプリケーションに実行させるものです。EViews 12 では、R、Matlab および Python をサポートしています。

```
xrun import numpy as np
xrun import pandas as pd
xrun import statsmodels.formula.api as sm
xrun import matplotlib.pyplot as plt
```

エラーが出る場合のヒント

- パッケージがインストールされていることを確認
- パッケージのアップグレードを実行

- 次に、推定に用いる系列を含むグループオブジェクト `gr1` を作成し、Python に送信します。ここでは、`ptype=オプション`で Pandas DataFrame オブジェクトを指定し、グループオブジェクトをデータフレームとして python に手渡します。

```
group gr1 id week weight
xput(ptype=dataframe) gr1
```

```
>>> xput(ptype=dataframe) gr1
'gr1' (pandas.core.frame.DataFrame) created successfully.
```

- Python に渡したデータフレームの中身を確認します。head()と describe()でデータフレームの先頭と id、week、weight 各列の記述統計量を表示します。

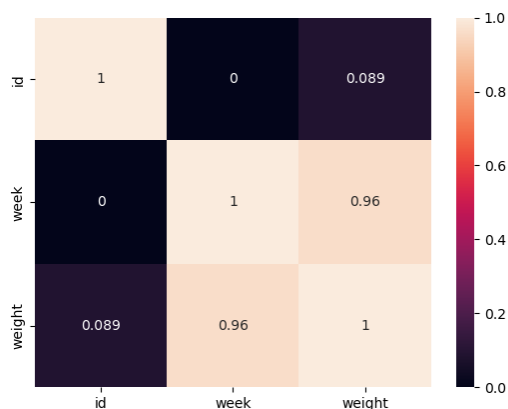
```
xrun gr1.head()
xrun gr1.describe()
```

```
>>> gr1.head()
   id  week  weight
0  1.0   1.0   24.0
1  1.0   2.0   32.0
2  1.0   3.0   39.0
3  1.0   4.0   42.5
4  1.0   5.0   48.0
```

```
>>> gr1.describe()
          id          week          weight
count  432.000000  432.000000  432.000000
mean    24.500000    5.000000   50.405093
std     13.869461    2.584983   16.641129
min      1.000000    1.000000   20.000000
25%     12.750000    3.000000   37.000000
50%     24.500000    5.000000   50.500000
75%     36.250000    7.000000   63.500000
max      48.000000    9.000000   88.000000
```

- さらに seaborn を利用して、ヒートマップを作成します。

```
xrun sns.heatmap(gr1.corr(), annot=True)
xrun plt.show()
```



- データフレームの確認が出来たら、Statsmodels.formula.api で混合効果モデルを推定します。

```
xrun model = sm.mixedlm("weight ~ week", gr1, groups = gr1["id"], re_formula =
"~ week")
xrun fit = model.fit(method = ["lbfgs"])
xrun print(fit.summary())
```

1 行目のコマンドで混合効果モデルを作成しています。groups でランダム切片を、re_formula で week がランダムスロープを持つことを想定します。ここでは、次のようなモデルを作成しています。

$$weight_{ij} = \beta_0 + \beta_1 week_{ij} + u_{0j} + u_{1j} week_{ij} + \epsilon_{ij}$$

- ここではランダム効果の分散と共分散が全て個別の値を持つと仮定します。

$$\Sigma = \text{Var} \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} \sigma_{u0}^2 & \sigma_{01} \\ \sigma_{01} & \sigma_{u1}^2 \end{bmatrix}$$

- 推定値は下記ようになります。

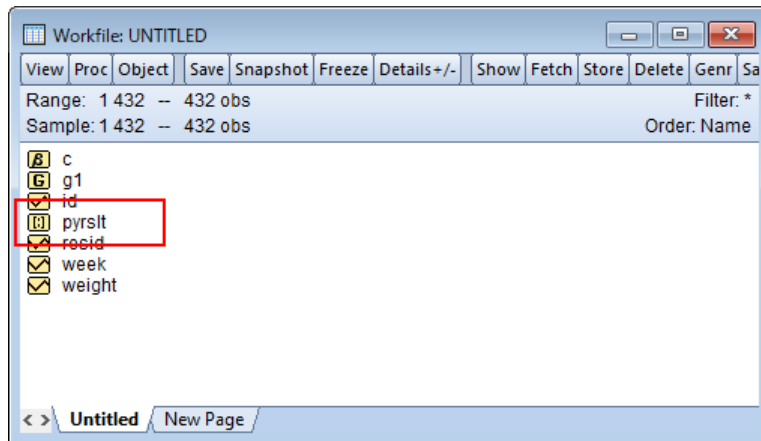
```
Mixed Linear Model Regression Results
=====
Model:                MixedLM    Dependent Variable:  weight
No. Observations:    432        Method:              REML
No. Groups:          48         Scale:              1.5969
Min. group size:     9          Log-Likelihood:    -870.4356
Max. group size:     9          Converged:         Yes
Mean group size:     9.0

-----
                Coef.  Std.Err.    z    P>|z|  [0.025  0.975]
-----
Intercept      19.356    0.404  47.929  0.000  18.564  20.147
week           6.210    0.092  67.471  0.000   6.030   6.390
Group Var      6.985    1.364
Group x week Cov -0.103    0.208
week Var       0.380    0.071
=====
```

- 推定が出来たら、係数を xget コマンドで EViews に取り込みます。

```
xget(name = pyresult, type = vector) fit.params
```

xget では EViews に取り込んだ際の EViews オブジェクト名とオブジェクトタイプを指定する必要があります。ここでは、推定した係数を pyresult という名前のベクトルオブジェクトとしてワークファイルに保存します。



- 推定されたパラメータの分布を表示することができましたが、Pyflux では時間変化をグラフにすることができません。そこで、推定結果を EViews に送りってグラフを作成します。

参考資料

IHS Global Inc. “Generalized Autoregressive Score (GAS) Models: EViews Plays with Python”. EViews Blog. 2019-4-23. <http://blog.eviews.com/2019/04/generalized-autoregressive-score-gas.html>, (2022-2-10 閲覧)

IHSEViews “Integrating Python with EViews 11” Online video. YouTube, 2019-04-04. Web.

Retrieved from <<https://www.youtube.com/watch?v=ESnVodbYkbg&t=1s>> (2022-2-10 閲覧).

Ross Taylor “GAS Models”. PyFlux. 2016. <https://pyflux.readthedocs.io/en/latest/gas.html>, (2022-3-60 閲覧)

- サンプルファイル (pygas.WF1、seasuroot.prg) は下記よりダウンロードできます。
<http://blog.eviews.com/2019/04/generalized-autoregressive-score-gas.html#sec7>