USING PYTHON TO RESOLVE THE ECONOMIC PROBLEMS

Alexandru COLESNICOV
PhD in Computer Science
Institute of Mathematics and Computer Science
„Vladimir Andrunachievici"
acolesnicov@gmx.com
Ludmila MALAHOV
Scientific researcher
Institute of Mathematics and Computer Science
„Vladimir Andrunachievici"
lmalahov@gmail.com
Tatiana COLESNICOVA
PhD in Economics
National Institute of Economic Research
ctania@gmail.com
Serghey VERLAN
Dr.hab. in Computer Science
France - Université Paris Est Creteil
sverlan@gmail.com

Abstract

The work discusses the essential aspects of implementation of online economic applications using Python. The exemplified application performs econometric calculations over the data on an enterprise to analyze situation on labour market. Calculations include data grouping and statistics, extended Mincer earnings function, Duncan index of dissimilarity, the Oaxaca-Blinder decomposition.

Key words*: online economic applications, Python, econometric calculations, discrimination on labour market.*

JEL Classification: *C02, C80, C88, C89*

Introduction

The paper discusses use of Python at the development of applications that perform econometrical calculations on micro-economic level.

Python[1] is a high-level general-purpose programming language. Its advantages: code readability because of simple and clear language constructs; dynamic typing; automatic garbage collection; support of procedural, object-oriented, and functional programming; comprehensive standard library; rich collection of extensions (libraries) including complicated econometrical calculations, etc.

The data we process contain indicators of all employees of the company or its subdivision for a year. The indicators are: gender, age, position at the company, level of education, working experience (in general, and at the company), marital status, number of children, sum of all payments, number of days for business trips, trainings, and illness, type of working conditions, etc.

Calculations include different algorithms. Mostly it is grouping statistics, for example, levels of employees' education for all employees and for groups by different indicators. Results are shown in absolute quantities and in percent, and presented as tables and graphics (diagrams). More complicated algorithms include Duncan index of dissimilarity, extended Mincer equation, and the Oaxaca-Blinder decomposition.

We selected [1] the XWiki platform[2] to develop our applications. XWiki provides flexible and extensible tools to implement the dynamic content. The complexity of calculations necessary to generate pages defines the language we use to program them. We found that the available standard tools of XWiki do not cover all our needs, and developed a new approach integrating XWiki with Python 3.

XWiki and Python are free and open source.

1. Python for econometrical calculations

Python libraries mostly used for econometrical calculations are listed in Tab. 1 below.

Table 1. Python libraries for econometrical calculations

Library	Description
Pandas	Provides data structures that permits easy, intuitive, and flexible work with structured (tabular, multidimensional, heterogeneous, etc.) data.

[1] https://www.python.org/
[2] http://www.xwiki.org/xwiki/bin/view/Main/WebHome

	DataFrame object imitates an Excel sheet. **DataFrame** is inspired by **data.frame** from R but has much more features. **Series** is one-dimensional data structure. Other features include: handling of missing data; insertion and deletion for columns and rows; automatic explicit data alignment; aggregation; conversion between DataFrame and other Python data structures; label-based slicing, indexing, and filtering of large data sets; merging and joining data sets; reshaping and pivoting of data sets; hierarchical labeling of axes; IO tools for loading data from CSV and delimited files, Excel tables, databases, HDF5 format; time series-specific functionality, etc.
NumPy	Provides: a powerful *N*-dimensional array object; tools for integrating C/C++ and Fortran code; linear algebra, Fourier transform, and random number capabilities, etc.
Matplotlib	Matplotlib is a Python 2D plotting library, which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Covers all possibilities of Excel diagrams, and even more.
SciPy	A framework for scientific calculations organized in subpackages that should be loaded separately.

Subpackage	Description	Subpackage	Description
cluster	Clustering algorithms	**odr**	Orthogonal distance regression
constants	Physical and mathematical constants	**optimize**	Optimization and root-finding
fftpack	Fast Fourier Transform	**signal**	Signal processing
integrate	Integration and ordinary differential equation solvers	**sparse**	Sparse matrices and associated routines
interpolate	Interpolation and smoothing splines	**spatial**	Spatial data structures and algorithms
io	Input and Output	**special**	Special functions
linalg	Linear algebra	**stats**	Statistical distributions and functions
ndimage	*N*-dimensional image processing		

Source: created by authors

2. Regression analysis in Python

Python contains several packages that perform linear regression, for example, *statsmodels* and *scikit-learn*. Regression analysis is easy. We collect values of independent and dependent variables into the prescribed data structure, and then call the regression solver. The result will contain equation coefficients and error estimations. See [5, p. 279-291] for details.

In the following example, lines 1 and 2 of the Python program load packages *pandas* and *statsmodels*, line 3 creates Pandas DataFrame structure with data, line 4 performs calculations, and lines 5-6 print results.

Listing 1. Example of regression analysis in Python

```
(Program)
1 import pandas as pd
2 import statsmodels.formula.api as sm
3 df = pd.DataFrame({"x": [0,1,2,3,4,5,6,7,8,9,10,11], "y": [0,1,1,2,2,3,4,4,5,5,6,7]})
4 result = sm.ols(formula="y ~ x", data=df).fit()
5 print(result.summary())
6 print()

(Result)
...>python ex3.py
...UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=12

                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     546.5
Date:                Mon, 25 Nov 2019   Prob (F-statistic):           4.65e-10
Time:                        15:44:14   Log-Likelihood:                -1.7872
No. Observations:                  12   AIC:                             7.574
Df Residuals:                      10   BIC:                             8.544
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0256      0.167      0.153      0.881      -0.347       0.398
x              0.6014      0.026     23.377      0.000       0.544       0.659
==============================================================================
Omnibus:                        1.394   Durbin-Watson:                   2.704
Prob(Omnibus):                  0.498   Jarque-Bera (JB):                0.778
Skew:                          -0.124   Prob(JB):                        0.678
Kurtosis:                       1.778   Cond. No.                         12.4
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

3. Mincer earning function

Mincer earning function is an empirical presentation of dependence of a person's earning (wage) of person's education and experience. It states that logarithm of wage is equal to the sum of a constant, a linear function of years of schooling, and a quadratic function of experience [8, p. 13].

To calculate coefficients of Mincer function, we use linear regression. We calculate in our next example two sets of coefficients separately for women (2,363 observations) and for men (3,743 observations). Note the simplicity of the program: import of necessary libraries, input of the data from the Excel table, one-line calculation for women, result print, and the same for men. To separate data for women and men, a filter over the *df0* DataFrame object was used. To perform the same calculations in Excel, we were to sort data by gender forming continuous data ranges for women and men. The compared results were the same. Equally, we do not need to introduce new columns with ln(w) and ex^2 values using R-style formulas.

The printed below (Listing 2) fragment of the Excel table contains:

- *code* is depersonalized employee's code;
- *gender*;
- *ed* is code of education;
- *ex* is working experience in years;
- *w* is wage per hour.

Listing 2. Mincer earning function of real data

```
(Data, first 5 lines)

code gender ed  ex       w
   2      F  4  4.0833 39.7184
   4      M  1 29.6694 37.2090
   5      M  3 18.4974 35.8457
   6      F  3 16.3844 35.5105
   7      M  3 27.5645 44.6152
(... total 6106 lines of data)

(Program)
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
df0 = pd.read_excel("./data.xls", sheet_name="Data", header=[0])
result1 = smf.ols(formula="np.log(w) ~ ed + ex +  np.square(ex)", \
  data=df0.loc[df0["gender"] == "F"]).fit()
print(result1.summary())
print()
result2 = smf.ols(formula="np.log(w) ~ ed + ex + np.square(ex)", \
  data=df0.loc[df0["gender"] == "M"]).fit()
print(result2.summary())
print()
```

```
(Result for women)
                            OLS Regression Results
==============================================================================
Dep. Variable:              np.log(w)   R-squared:                       0.324
Model:                            OLS   Adj. R-squared:                  0.323
Method:                 Least Squares   F-statistic:                     377.0
Date:                Fri, 29 Nov 2019   Prob (F-statistic):          5.42e-200
Time:                        14:55:35   Log-Likelihood:                -1014.2
No. Observations:                2363   AIC:                             2036.
Df Residuals:                    2359   BIC:                             2059.
Df Model:                           3
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept         2.8512      0.033     86.312      0.000       2.786       2.916
ed                0.2151      0.006     33.386      0.000       0.202       0.228
ex                0.0204      0.003      7.371      0.000       0.015       0.026
np.square(ex)    -0.0003   6.78e-05     -4.543      0.000      -0.000      -0.000
==============================================================================
Omnibus:                      528.067   Durbin-Watson:                   1.961
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4353.043
Skew:                           0.820   Prob(JB):                         0.00
Kurtosis:                       9.444   Cond. No.                     3.13e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
 specified.
[2] The condition number is large, 3.13e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
(Result for men)
                            OLS Regression Results
==============================================================================
Dep. Variable:              np.log(w)   R-squared:                       0.350
Model:                            OLS   Adj. R-squared:                  0.350
Method:                 Least Squares   F-statistic:                     671.2
Date:                Fri, 29 Nov 2019   Prob (F-statistic):               0.00
Time:                        14:55:35   Log-Likelihood:                -693.88
No. Observations:                3743   AIC:                             1396.
Df Residuals:                    3739   BIC:                             1421.
Df Model:                           3
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept         3.0118      0.018    171.540      0.000       2.977       3.046
ed                0.1628      0.004     41.927      0.000       0.155       0.170
ex                0.0254      0.002     16.605      0.000       0.022       0.028
np.square(ex)    -0.0004   3.61e-05    -11.327      0.000      -0.000      -0.000
==============================================================================
Omnibus:                       64.000   Durbin-Watson:                   1.763
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              116.897
Skew:                           0.097   Prob(JB):                     4.13e-26
Kurtosis:                       3.844   Cond. No.                     2.57e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
 specified.
[2] The condition number is large, 2.57e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Mincer earning function is:

$\ln(w_i) = \beta_0 + \beta_1 ed_i + \beta_2 ex_i + \beta_3 ex_i^2 + \varepsilon_i$

The following Tab. 2 presents results of calculation selected from Listing 2.

Table 2. Mincer function, by gender

Gender	β_0 (intercept)	β_1 (*ed*)	β_2 (*ex*)	β_3 (ex^2)
F	2.8512	0.2151	0.0204	-0.0003
M	3.0118	0.1628	0.0254	-0.0004

Source: created by authors

These results show gender misbalance. Base wage exp(β_0) for women is less, and their earnings grow in a lesser proportion with the accumulation of experience. Instead, wages of women are more dependent of education. More refined investigations may involve other independent variables and additional calculations in Python like Oaxaca-Blinder decomposition, to support or to reject existence of discrimination by gender.

4. Problems at the implementation of the application

The ability to use the necessary additional development tools is one of the power features of the XWiki environment.

The XWiki environment provides as an extension a macro to include a Python program directly in the code of a dynamic Web page. We tried this extension and found out that its functionality is not enough. Namely, this extension uses Jython that implement Python in the Java environment. With this approach, translation from the Python language is performed not into the Python interpreter codes, but into the codes of the Java virtual machine. Their execution is performed in the Java environment. Two restrictions follow from this: on the version of the language, and on plug-in library technologies.

Jython implements an older version of Python 2 (version 2.7), on which development of Python 2 was stopped. All innovations from Python 3 are thus unavailable.

In addition, only several Python libraries can be transferred under Jython, namely, those ones that use the binary code of the Python interpreter, which can be easily translated into Java virtual machine code. However, most modern Python libraries make extensive use of machine code for optimization. Such libraries cannot be connected with Jython.

Unfortunately, this concerns just the most needed libraries that implement complex calculations, for example, the least squares method. These include the NumPy, SciPy, Pandas and several others libraries. Finally, since July 2019 support of the Pandas library for Python 2 has been discontinued.

We can recommend to perform simplest calculations in Python 2 including the Python code directly in the dynamic page. For more complex calculations (regression analysis, etc.), we propose another approach.

We used the most modern standard implementation of Python 3. This implementation has access to a full set of libraries.

The JPserve library was installed. It is an implementation of Python calling tools from a Java program. This library consists of a server written in Python and a client implemented in Java. Being a Python module, the server part is installed in the usual way: with the **`pip`** command from the PyPI repository. The client library is added (copied) to the root directory of the Tomcat Web application server along with other Java archives that collectively implement the XWiki environment, and thus become part of XWiki.

We cannot use the root user to start the Python server for security reasons. A separate user was specially created, in whose partition the server is started and all calculations are performed, and all Python programs and intermediate data are stored. Using a dedicated user makes it impossible to damage areas belonging to other Linux users.

To run the program in Python and include the result of the calculations in a dynamic XWiki page, the Groovy macro is included in the code of this page. (Groovy is an extension over Java. In fact, in this way Java code is included in the page.) In this case, everything displayed by the **`print()`** or **`println()`** commands is considered the XWiki code (extended HTML) and displayed by the browser.

In the Groovy code, a client program is called, one of the parameters of which is the name of the Python program to be executed. The Java client (that is, XWiki) sends a request to execute the program on the port allocated for this purpose to the server. The running in parallel Python server accepts the request and runs the specified program. The result must be assigned to the variable **`_result_`** and is returned to the client (in the Java program) in standard JSON encoding. Conversion of a JSON string into XWiki variables or into HTML code is performed by a standard program available in Java.

Conclusion

The developed new technology of integration Python 3 into XWiki environment permitted us to use powerful Python 3 libraries to implement necessary algorithms to solve economic problems that needs complex calculations. The difficulty level of this implementation is comparable with direct calculations in Excel or other econometrical application. The results of calculations were checked by comparison with the results of calculations in Excel and Stata.

We develop applications instead of use Excel or any other system because any general purpose tool like Excel or Stata is too cumbersome. With the restricted and fixed repertoire of calculations needed for our research, and with fixed data structure, it's better to develop and use a specialized program. Small changes, and even inclusion of new calculations can be made by relatively simple Python programming in the XWiki environment.

Acknowledgement. Examples of econometrical calculations for this paper are taken from the scientific project for the years 2018-2020, entered in the State Register of Science and Innovation Projects with the code 18.80013.0807.06.STCU / 6336 "Innovative approaches to applied computations and software development for gender equality regulation on labor market" („Abordări inovaționale privind dezvoltarea calculelor aplicative și software-ului pentru reglementarea egalității de gen pe piața forței de muncă"), within the framework of the STCU-ASM Joint Research-Development Initiatives Program, the call from 10.04.2017.

References

[1]. Colesnicov A., Malahov L., Lucasenco. E. On Implementation Platform for Gender-Oriented Econometrical Calculations, pp. 232-244. Proceeding of the International Symposium "Experience. Knowledge. Contemporary Challenges", 3rd Edition "Romania in the Year of the Centenary. The European and global socio-economic context". December 13th-14th, 2018, Editura "Artifex", Bucharest, Romania. - 696 p. - ISBN 978-606-8716-43-5

[2]. Colesnicova T. Gender equality regulation in the sphere of employment in the Republic of Moldova. Monografie. Ch.: Complex

Ed. al IEFS. 2012, CZU 316.346.2(478), C62, ISBN 978-9975-4326-5-8, - 175 p.

[3]. Colesnicova T., Malahov L., Colesnicov A. Some applications of IT in gender studies. În: Economic growth in conditions of internationalization=Creşterea economică în condiţiile internaţionalizării: Conferinţa X ştiinţifico-practică internaţională din 15-16 octombrie 2015: (în 2 vol.). Ch.: INCE, 2015, vol. II, - 252 p. ISBN 978-9975-4185-2-2, p.201-204.

[4]. Colesnicova T. Segregarea de gen pe piața forței de muncă a Republicii Moldova: măsurarea și analiza comparativă. În: Economic growth in conditions of globalization: sutainable development models=Creşterea economică în condiţiile globalizării: modele de dezvoltare durabilă - Conferinţa XII ştiinţifico-practică internaţională din 12-13 octombrie 2017. Ch.: INCE, 2017: (în 2 vol.), vol. II, - 295 p. ISBN 978-9975-3171-2-2, p.238-242

[5]. Sargent T.J., Stachurski, J. Lectures in Quantitative Economics with Python. July 27, 2019. 1560 p. https://delong.typepad.com/files/quantitative-economics-with-python.pdf

[6]. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd Edition. O'Reilly Media, Inc., 2017. ISBN: 9781491957653

[7]. Bell A. Python for Economists. October 2016. 35 p. https://scholar.harvard.edu/files/ambell/files/python_for_economists.pdf

[8]. Polachek, S.W. Earnings Over the Lifecycle: The Mincer Earnings Function and Its Applications. Institute for the Study of Labor, IZA DP No. 3181. November 2007. 111 p.

[9]. Сузи Р. А. Язык программирования Python: Учебное пособие. — М.: ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. — 328 с. — ISBN 5-9556-0058-2, ISBN 5-94774-442-2

[10]. Доусон М. Программируем на Python. — СПб.: Питер, 2012. — 432 с. — ISBN 978-5-459-00314-7.

[11]. Саммерфилд М. Python на практике. — Перевод с английского. — М.: ДМК Пресс, 2014. — 338 с. — ISBN 978-5-97060-095-5.

[12]. Лутц М. Программирование на Python / Пер. с англ. — 4-е изд. — СПб.: Символ-Плюс, 2011. — Т. II. — ISBN 978-5-93286-211-7.