LI Wai Yin Follow
Dec 23, 2017 · 14 min read

Python101－廣東話Python入門－Simple Linear Regression

. . .

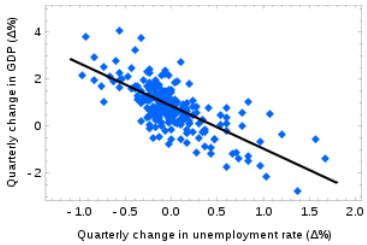
Python Logo

. . .

上一篇就講咗點用 *Pandas*，今次要講嘅係線性回歸（*Linear Regression*）。

簡單介紹下佢係乜嘢先，但因為個重點係 *Python* 度，所以都真係好簡單就算。類似小學生個陣，收集到植物生長高度之後，畫一張散點圖（*Scatter Plot*），當時唔識 *Regression* 嘅小學生就可以憑感覺搵一條最近所有點嘅線。咁呢個搵最近所有點嘅線嘅過程就係 *Regression* 做緊嘅嘢。

咁識得 *Regression* 就會做到類似下面張圖咁嘅嘢：



File: Okuns_law_quarterly_differences.svg, Source: Wiki

因為數學唔可以憑感覺，雖然數學家做證明係講靈感，咁所以就有好一堆方法去計一條「最近所有點嘅線」出嚟。

咁最基本嘅樣就係假設 *Error Terms* 係 *Follow Normal Distribution*。之後就會 *Form* 到條式出嚟：

> $y = \beta 0 + \beta 1 * X1 + \beta 2 * X2 + \ldots + \beta n * Xn + \varepsilon$

咁 *Error*（ε）*follow Normal* 嘅一般就叫 *Ordinary Least Squares*，而更 *general*（可以非 *Normal Error*）嘅 *form* 就叫 *General Linear Model*。如果想知多啲，就睇下 *Gauss–Markov theorem*。咁唔講咁複雜，係咁先。

. . .

前言

咁 *Regression* 或者呢類 *Model* 可以用嚟做咩呢？咁我上第一堂 *Econometrics* 個時，記得就講過有三種功能：一，描述、二，解釋因果、三，預測。例如話想描述下身高同體重嘅關係，解釋失業率點影響生產力，同埋預測出年某段時間嘅蘋果產量之類都係呢類 *Model* 嘅功能。描述同解釋亦都係 *Neural Network* 之類嘅複雜 *Model* 做唔到嘅嘢。

. . .

設置

要用嘅 *Library* 有：
1. scikit learn
2. StatsModels
3. SciPy
同埋 *numpy* 加 *pandas*。

今次嘅材料都係上次個 *Set*：
1. Okun's law data (1948-I — 2002-I)

想搵材料可以去：

咁好好彩有人整理咗一大堆 *Data* 嘅網站 — *Awesome Public Datasets*。

另外仲有 *Kaggle* 或者 *UC Irvine Machine Learning Repositor* 之類。

咁甚至 *sklearn*、*tensorflow* 佢地都有內置一啲好出名嘅 *dataset*，例如 *iris* 同 *MINST*。

. . .

1. Sci-kit Learn嘅 Linear Regression

```
model = sklearn.linear_model.LinearRegression(fit_intercept=True,
normalize=False, copy_X=True, n_jobs=1).fit(X, y)
```

sklearn.linear_model.LinearRegression - scikit-learn 0.19.1 documentation

This parameter is ignored when fit_intercept is set to False. If True, the regressors X will be normalized before...

scikit-learn.org

優點：

- *Sklearn* 連帶有好多其他功能，例如 *Dimension Deduction（Component Analysis 同 Isomap）*、*Discriminant Analysis*、*Latent Dirichlet Allocation* 之類嘅 *Topic Model*。
- 夠簡潔，連 *Google* 嘅 *Tensorflow* 都係抄 *Sklearn Syntax*。

缺點：

- 咁 *Sklearn* 係一個做 *predictive modelling / machine learning* 嘅 *library*，所以佢嘅 *validation* 方法係用 *cross validation*，即係就咁塞 *testing set* 去睇下佢係咪 *predictable*，咁所以做正統統計嘅時候要用到嘅嘢係會冇嘅。
- 例如係 *R* 或者其他 *Stat* 用嘅 *s/w* 做完 *Regression* 之後都會有 *Report*，好似 *ANOVA* 之類，不過 *Sklearn* 就冇。

. . .

2. statsmodels

```
model = statsmodels.regression.linear_model.OLS(Y, x)
```

Linear Regression - statsmodels 0.8.0 documentation

Linear models with independently and identically distributed errors, and for errors with heteroscedasticity or...

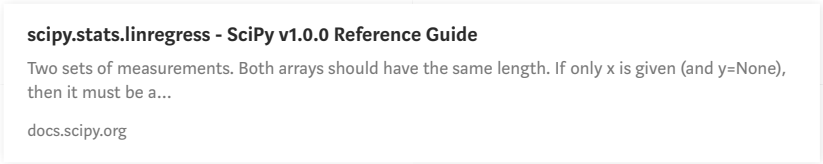
www.statsmodels.org

咁所以就要介紹另一個係 *Python* 入面做正統統計嘅 *Library – statsmodels*。

. . .

3. SciPy

```
slope, intercept, r_value, p_value, std_err =
scipy.stats.linregress(x, y)
```



咁仲有個叫 *SciPy* 嘅 *Library*，佢就係唔止 *For Stat / Math* 嘅，仲有 *Science* 同 *Engineering* 嘅功能，但我就少用，不過都介紹下。

. . .

咁開始啦！

首先用返上一次個 *set Okun's law* 嘅 *data* 先，之後就試下 *sklearn* 個 *linear regression*。

In [1]:
```
import matplotlib.pyplot as plt
```

In [2]:
```
import pandas as pd
```

In [3]:
```
okun = pd.read_excel('okun.xls')
```

In [4]:
```
okun.head()
```

Out[4]:

	gnp	un
0	1626.4	3.733333
1	1655.5	3.666667
2	1665.1	3.766667
3	1669.0	3.833333
4	1643.8	4.666667

In [5]:
```
okun['%change_gnp'] = okun['gnp'].pct_change() * 100
```

In [6]:
```
okun['%change_un'] = okun.un - okun.un.shift(1)
```

In [7]:
```
okun.head()
```

medium_sklearn_ols.ipynb hosted with ❤ by **GitHub** **view raw**

咁因為 *okun's law* 係講緊失業率嘅變動同埋生產力嘅變動之間嘅關係，所以就用 *Series.pct_change() * 100* 去計咗 *GNP* 嘅變動出嚟先。至於失業率就用 *Series.shift(1)* 去減出嚟。點解失業率唔洗除返自己？你望下失業率個單位係咩就明。

（*Rmk: .shift(1)* 係成條 *series* 向下一格， *.shift(-1)* 就係向上一格。）

咁之後就用上次提過嘅 *DataFrame.dropna()* 去將第一行 *drop* 走佢。之後用散點圖（*scatter plot*）去望下組 *data* 係咪真係有關係先，咁望落係似有嘅，當失業率下跌時，生產力就會上升。

咁但係我地想知道佢地數字上有咩關係，就係 *linear regression* 出手嘅時間！

```
linreg = sklearn.linear_model.LinearRegression(…)
```

呢句係講緊 *save* 一個 *LinearRegression* 嘅 *class* 落 *linreg* 度。簡單嚟講就係依家有一個叫 *linreg* 嘅 *Linear Regression Model* 啦。之後就 *fit* 我地嘅 *data* 入去。咁望返 *document* 會見到：

```
fit(X, y, sample_weight=None)
X : numpy array or sparse matrix of shape [n_samples,n_features]
y : numpy array of shape [n_samples, n_targets]
```

. . .

咁上面一開始提到 *Error follow normal distribution* 又係咩意思呢？（幫自己賣下廣告，關於 normal 同現實（極端）世界嘅廢話：繼《黑天鵝效應》｜與｜《隨機騙局》）

```
In [1]: import matplotlib.mlab as mlab
        import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
```

```
In [2]: okun = pd.read_excel('okun.xls')
        okun['%change_gnp'] = okun.gnp.pct_change() * 100
        okun['%change_un'] = okun.un - okun.un.shift(1)
        okun = okun.dropna()
        okun.head()
```

Out[2]:

	gnp	un	%change_gnp	%change_un
1	1655.5	3.666667	1.789228	-0.066667
2	1665.1	3.766667	0.579885	0.100000
3	1669.0	3.833333	0.234220	0.066667
4	1643.8	4.666667	-1.509886	0.833333
5	1638.6	5.866667	-0.316340	1.200000

```
In [3]: okun['predicted_gnp'] = 0.8502 + okun['%change_un'] * -1.809
        7
```

medium_error.ipynb hosted with ❤ by **GitHub** **view raw**

咁上面 *[3]* 就係講解用呢個 *linear regression* 預測嘅結果，同真實嘅數值相減 *[4]* 就係 *Error*。咁 *print* 出嚟可以見到其實唔係差好多，咁之後就用呢個 *Error* 去計返 *mean* 同 *standard deviation*，再轉返做 *normal curve* 就可以 *plot* 到圖入面個個鐘形曲線，你可以見到 *normal curve* 係集中係中間，兩端只有極少數，同埋左右對稱。而 *histogram* 個結果都係差唔多。

咁當然， 統計係有好多方法去計或者判定 *regression error* 係咪真係 *follow normal* 或者其他 *distribution*。咁例如比較易睇嘅就係 *QQ Plot*（同 *700* 個隻 *QQ* 冇關係，呢度個 *QQ stand for quantile-quantile*）。

咁 *QQ plot* 都話比我地知到呢個 *error* 其實係幾 *follow normal*。亦都有計數去 *check* 佢係咪真係 *normal* 嘅。

至於唔 *follow normal* 代表咩呢？就係你個 *model* 解釋唔到 *error* 入面嘅某啲嘢。咁解釋唔到又有咩問題呢？就好似上面篇文咁講，你會高估或者低估咗好多嘢，好似萬年一遇嘅美少女同天氣咁。

. . .

Residual

咁之後就會用 *statsmodel* 同 *scipy* 去 *fit* 一次呢組 *data*。

```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        okun = pd.read_excel('okun.xls')

In [2]: from scipy.stats import linregress

In [3]: import statsmodels.api as sm

        /usr/local/lib/python3.5/dist-packages/statsmodels/compat/pa
        ndas.py:56: FutureWarning: The pandas.core.datetools module
        is deprecated and will be removed in a future version. Pleas
        e use the pandas.tseries module instead.
          from pandas.core import datetools

In [4]: okun['%change_gnp'] = okun['gnp'].pct_change() * 100
        okun['%change_un'] = okun.un - okun.un.shift(1)
        okun = okun.dropna()

In [5]: slope, intercept, r_value, p_value, std_err = linregress(oku
        n['%change_un'].values, okun['%change_gnp'].values)

In [6]: print("r-squared:", r_value**2)

        r-squared: 0.495337954797
```

medium_ols.ipynb hosted with ❤ by **GitHub** **view raw**

點用個 *function* 就睇返上面啦。你會見到佢地計出嚟係一樣嘅，都係

> $y = 0.8502 - 1.8097 * X$

咁 R^2 *Score* 都一樣係 *0.4953*。除咗 *Syntax* 之外，你會見到最大分別就係 *statsmodels* 有一個叫 *model.summary* 嘅 *function*，亦即係會自動幫你做上面提過嘅 *Test* 或者其他 *statistics*，例如右下角嘅 *Durbin–Watson* 係計 *prediction errors* 有冇 *autocorrelation*、*Jarque-Bera* 就係計個 *model* 嘅 *goodness of fit*。要講有排講，有興趣嘅人可以自己 *google* 下。或者之後有機會再講。

. . .

R^2 Score 同 Adj R^2 Score

咁首先就講下 R^2 點計出嚟先：

> R^2 = SSR(Regression sum of squares)/SST(total sum of squares)

咁直接嚟講 R^2 就係講緊個 *model* 解釋到成組 *data* 嘅幾多變量：*0* 就係咩都解釋唔到，*1* 就係解釋到全部。

咁有幾點要留意：第一，R^2 係講緊解釋依家呢 *set data* 嘅能力，而唔係預測能力。咁所以會有人計一個 *predicted* R^2 去睇下個 *model* 係咪 *overfit*。

第二， R^2 並唔係愈高愈好，過高嘅 R^2 可能會係 *fit* 埋 *noise*。

第三，R^2 低唔代表佢地冇關係，可能你用緊 *Linear Model* 而佢地係 *square* 嘅關係。咁點睇 R^2 就真係 *case by case*。

上面提到嘅 R^2 過高，有一個好常見嘅理由，就係 *parameters* 太多，咁因為 R^2 條式對於好多特徵嘅 *inputs* 係冇懲罰，咁所以只要你是旦乜都扔曬落去，舉例的話，賽馬，你除咗隻馬幾歲，幾重幾高之外，連騎師老豆幾多歲、練馬師老婆係咪大肚、如果馬主當年冇買馬咁佢個仔今年幾多歲呢啲都扔曬比佢，係會令到 R^2 增大嘅。

咁所以統計佬就整咗個叫 *Adj* R^2 出嚟，就係會對多特徵作一個懲罰，咁就算 R^2 好高，裡面太多垃圾嘅時候， *Adj* R^2 就會好低。

. . .

咁上面就講咗嘅其實只係 *simple linear regression* 係點用 Python 做，咁跟住下一篇可能就會講 *multivariable linear regression* 同 *polynomial regression*。

. . .

後記

講講下好似講緊統計咁，但其實我係想講 *Python*。咁睇嚟下一篇用畫龜講 *Python Syntax* 先會好啲。咁應該要重新排過曬啲文嘅 *number* 佢？

講完一個接近 *Econometrics* 多過 *Statistics* 嘅嘢之後，諗返 *Statistics* 大部份時間其實都係計 *E(X)* 同 *Var(X)* ，之後再 *prove* 一堆 *prove* 完都唔知自己 *prove* 咗咩嘅嘢，反而好少會真係落手去掂 *data*。

其實比起呢啲 *data*，我更加想講下試下 *apply sabermetrics*（直譯係賽伯計量學，但我覺得叫棒球統計學會合意啲）落其他運動或者今朝見到人地 *apply* 落戰爭史度。咁樣睇落故事好多，講起上嚟都有趣好多，仲可以講多啲廢話。

講起呢樣嘢就想講棒球真係一樣幾另類嘅運動，佢應該係最早有一套完整嘅數據收集系統同最早用數字評價球員嘅運動，當然啱唔啱係後話。（係黑天鵝篇文到推介過，係呢度再推介多次 *Money Ball*。真係好睇。）

不過我唔太熟 *sabermetrics*，花多少少時間鋤下佢先再講。

. . .

延伸閱讀：

Linear Regression

"An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem."...

towardsdatascience.com

Mathematics of simple regression

This is the estimated standard deviation of the error in estimating the mean. Notice that it is inversely proportional...

people.duke.edu

Data science with Python: 8 ways to do linear regression and measure their speed

We discuss 8 ways to perform simple linear regression in Python ecosystem. We gloss over their pros and cons, and show...

medium.freecodecamp.org

. . .

附錄：

上面用過嘅嘢都打曬落 *Github: 1. simple regression with okun's law* 裡面。

. . .

多謝收看！

Python Linear Regression 香港 中文 教學