程式設計概論
Programming 101
–SciPy進行科學運算

授課老師：邱淑怡

Date:5/11/2021

Outline

- 模組的比較
- Pandas 如何處理相關係數
- SciPy(https://www.scipy.org/)
  - SciPy是基於Python的Numpy擴展構建的數學算法和便利函數的集合
  - 是建構在Numpy之上，用來進行科學計算的模組
- SciPy 統計分析(https://scipy-lectures.org/packages/statistics/index.html)
  - 統計子模組: scipy.stats

Python模組的比較

- pandas用來整理數據表格
- Numpy是以矩陣基礎做數據的數學運算
- Matplotlib是Python繪圖的它包含了大量的工具，你可以使用這些工具創建各種圖形，包括簡單的折線圖、散佈圖、直方圖，甚至是三維圖形
- SciPy就是以Numpy為基礎做科學、工程的運算處理的package，包含統計、優化、整合、線性代數、傅立葉轉換圖像等較高階的科學運算。

Pandas 如何處理相關係數

- DataFrame.corr(method='pearson', min_periods=1)
- 參數說明：
  - method：可選值為‘pearson’, ‘kendall’, ‘spearman’}
    - pearson：Pearson相關係數來衡量兩個數據資料是否在同一線條上，即針對線性資料的相關係數而計算
    - kendall
    - Spearman: 非線性的相關係數
  - min_periods：樣本最少的資料量
  - Return value(返回值)：各類資料間的相關關係數

Pandas 如何處理相關係數_範例說明

```
# 建立範例的dataframe
Data1=[{'a':1,'b':2,'c':3}, {'a':4,'b':5,'c':6,'d':7},
{'a':4,'b':2,'c':3} ,{'a':3,'b':5,'c':4,'d':7}]
Df1=pd.DataFrame(Data1)
print(Df1)
```

```
#相關係數
Df1_p=Df1.corr()
Df1_k=Df1.corr('kendall')
Df1_s=Df1.corr('spearman')
print(Df1_s)
print('欄位a與b的相關係數:',Df1_s.at['a','b'])
```

SciPy被組織成覆蓋不同科學計算領域的子模組

子模組名稱	說明
scipy.constants	物理和數學常數
Scipy.fftpack	傅里葉變換
scipy.integrate	整合例程
scipy.interpolate	插值
scipy.io	資料輸入和輸出
scipy.linalg	線性代數例程
scipy.optimize	優化
scipy.signal	信號處理
scipy.sparse	稀疏矩陣
scipy.spatial	空間資料結構和演算法
scipy.special	任何特殊的數學函式
scipy.stats	統計

SciPy 統計分析

```
import scipy.stats as stats
```

SciPy的統計分析_例子

- python 來做統計分析時一般使用 scipy 中的 stats
- 問題:是否有某些產品，在颱風天時會與平時的銷售狀況特別不一樣?
  - T 檢定的功能是比較**兩平均數是否有顯著差異**，並且告訴你有多顯著。例如：颱風天跟非颱風天產品銷售數量是否有顯著差異？

$$H_0 : \mu_1 = \mu_2, H_1 : \mu_1 \neq \mu_2$$

T test

- T test功能是比較**兩平均數是否有顯著差異**，並且告訴你有多顯著
  - 男性與女性的平均身高是否有顯著差異？
- T test的虛無假設
  - 兩群的分佈是相似的
  - 透過檢定結果，通常以 P-Value 小於 0.05 為基準，推翻虛無假設，代表兩群的分佈是不一樣的
  - 就是男性與女性的平均身高有顯著差異

T test使用 Scipy

- 在 Python 中主要是使用 Scipy 這個套件，其中提供了兩個方法：
  - scipy.stats.ttest_ind
  - scipy.stats.ttest_*ind*_from_stats

T_test 檢定的程式

```
#two sample student t test
import numpy as np
from scipy import stats

mean1 = 30.97
mean2 = 21.79
std1 = 26.7
std2 = 12.1
nobs1 = 10
nobs2 = 10
modified_std1 = np.sqrt(np.float32(nobs1)/np.float32(nobs1-1)) *std1
modified_std2 = np.sqrt(np.float32(nobs2)/np.float32(nobs2-1)) * std2
(statistic, pvalue) = stats.ttest_ind_from_stats(mean1=mean1, std1=modified_std1,
nobs1=10, mean2=mean2, std2=modified_std2, nobs2=10)
print("t statistic is: ", statistic)
print("pvalue is: ", pvalue)
```

T_test 函示程式

```
def t_test(group1, group2):
    mean1 = np.mean(group1)
    mean2 = np.mean(group2)
    std1 = np.std(group1)
    std2 = np.std(group2)
    nobs1 = len(group1)
    nobs2 = len(group2)

    modified_std1 = np.sqrt(np.float32(nobs1)/
                    np.float32(nobs1-1)) * std1
    modified_std2 = np.sqrt(np.float32(nobs2)/
                    np.float32(nobs2-1)) * std2
   (statistic, pvalue) = stats.ttest_ind_from_stats(
              mean1=mean1, std1=modified_std1, nobs1=nobs1,
              mean2=mean2, std2=modified_std2, nobs2=nobs2 )
    return statistic, pvalue
```

上述例子的結果

- P value(=0.3599)大於0.05
- 不能拒絕原假設，就是這兩種作物產量沒有顯著差異

統計分析: 線性回歸

- stats.linregress( )

scipy.stats.linregress

scipy.stats.**linregress**(*x*, *y=None*) [source]

Calculate a linear least-squares regression for two sets of measurements.

Parameters: x, y : *array_like*

Two sets of measurements. Both arrays should have the same length. If only *x* is given (and `y=None`), then it must be a two-dimensional array where one dimension has length 2. The two sets of measurements are then found by splitting the array along the length-2 dimension. In the case where `y=None` and *x* is a 2x2 array, `linregress(x)` is equivalent to `linregress(x[0], x[1])`.

Returns:

slope : *float*

Slope of the regression line.

intercept : *float*

Intercept of the regression line.

rvalue : *float*

Correlation coefficient.

pvalue : *float*

Two-sided p-value for a hypothesis test whose null hypothesis is that the slope is zero, using Wald Test with t-distribution of the test statistic.

stderr : *float*

Standard error of the estimated gradient.

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

統計分析: 線性回歸範例

```
# create data frame
import pandas as pd
Data1=[{'a':1,'b':2,'c':3}, {'a':4,'b':5,'c':6,'d':7}, {'a':4,'b':2,'c':3} ,{'a':3,'b':5,'c':4,'d':7}]
Df1=pd.DataFrame(Data1)
print(Df1)
```

```
# linear regre
from scipy import stats
list1=Df1['c'].tolist()
list2=Df1['b'].tolist()
slope, intercept, r_value, p_value, std_err = stats.linregress(list1, list2)
print("slope: %f    intercept: %f" % (slope, intercept))
print("R-squared: %f" % r_value**2)
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

專題的流程圖、
資料前處理、資料分析