商管程式設計（106-1）

作業五

作業設計：盧信銘
國立台灣大學資管系

截止時間：2017 年 10 月 30 日凌晨 1 點
測資助教：張鑑霖

作業繳交請至 PDOGS (http://pdogs.ntu.im/judge/)。為第一題上傳一份，為第二題上傳兩份、為第三題上傳一份 Python 3.6 原始碼 (以複製貼上原始碼的方式上傳)。作業自己做。嚴禁抄襲。不接受紙本繳交，不接受遲交。請以英文或中文作答。

除了課本的內容外，Python 線上文件也很有用: https://docs.python.org/3/。好的程式設計師會把線上文件摸熟。

除下面所述之規定外，你可以使用任何 Python 內建的 Function 與 Library。但你的實作需依照題目的說明。如果你的結果正確，但沒有依照題目規定的方式實作，則不予計分。本次作業各題之實作規定如下：第一題需自行實作迴歸係數的計算。如使用現成的迴歸模型函數計算係數則不予計分。第二題如使用現成的常態分佈函數不予計分。第三題如使用現成的內部報酬率計算函數，或現成的二分逼近法函數，則不予計分。

第一題

(40 points) 迴歸分析是重要的資料分析工具。本題將實作單變數迴歸模型。給定因變數資料 $y_1, y_2, \dots, y_N$ 與自變數資料 $x_1, x_2, \dots, x_N$，我們要估計線性模型 $y_i = \alpha + \beta x_i + \epsilon_i$ 中 α 與 β 的數值。由最小平方法可得出

$$\hat{\beta} = \frac{\sum_{i=1}^{N}(y_i-\bar{y})(x_i-\bar{x})}{\sum_{i=1}^{N}(x_i-\bar{x})^2},$$

其中 $\bar{x}$ 與 $\bar{y}$ 分別是自變數與因變數的平均。而截距項則為 $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$。

樣本 i 的預測誤差為 $e_i = y_i - \hat{\alpha} - \hat{\beta}x_i$。噪音項 ($\epsilon_i$) 的變異數的估計值則為 $s = \sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N-2}}$。

請寫一個名叫 slm 的函數計算單變數迴歸的結果。你的程式應套用下面的範本：

```
#hw2_q1_skl_slm.py
def slm(xlist, ylist):
```

```
    #develop your function here...


#=======
xstr = input()
ystr = input()

xall = xstr.split(',')
yall = ystr.split(',')

out1 = slm(xall, yall)
if(out1 == None):
    print("DATA_ERROR")
else:
    print("%0.6f" % out1[0])
    print("%0.6f" % out1[1])
    print("%0.6f" % out1[2])
```

這個程式用 input()由使用者獲得一個逗點分隔的自變數資料，然後再由另一個 input()獲得逗點分隔的因變數資料。將自變數與因變數依照逗點切割後，傳入 slm()。slm()的第一個參數是自變數資料，第二個參數是因變數資料。自變數與因變數皆須為實數，數值大小不限。但為了測試方便，批改程式會將輸入資料限制在正負一億之間。

在以下狀況時，slm()直接輸出 None。這時主程式會輸出 DATA_ERROR ：

- 自變數與因變數的資料量不相同。
- 自變數或因變數的資料量小於五。

如果沒有上述問題，則 slm()依照題目給的公式計算$\hat{\alpha}, \hat{\beta}, s$，並組合成一個 list 回傳。主程式會印出結果。輸出的部分 (最後三行) 利用了字串格式化的功能，會將結果輸出至小數六位，四捨五入，不足補零。例如，如果 ans1=0.112，則這個指令：`print("%0.6f" % ans1)` 會印出 0.112000。

範例輸入

```
3.2,4.3,8.5,7.2,10,5
-1.3,-2,-5,3,4,3
```

範例輸出

```
-1.651069
0.303833
3.905542
```

範例輸入

```
1.4,-0.2,4.5,-3,0.9,-0.3
2,4,5
```

範例輸出

```
DATA_ERROR
```

範例輸入

```
1.4,-8.9,2.1,5,6,7,8,9,10
-1.4,8.9,-2,2.4,3,3,3,3,3
```

範例輸出

```
3.438685
-0.203236
3.085937
```

評分原則

依據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、輸入測試資料，並檢查輸出答案的正確性。一筆測試資料佔 2 分。

第二題

(30 points) 常態分布是一個相當重要的機率分布。標準常態分佈 (均數為零，變異數為一)的機率密度函數為$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ 。

(a) **請寫一個函數計算標準常態分佈的機率密度。**你的程式需由使用者獲得 x 的值，並輸出函數值。由於 PDOGS 是檢查結果是否完全一樣，我們統一規定輸出至小數六位，四捨五入，不足補零。例如，如果 ans1 是你的答案，則可以這樣做：`print("%0.6f" % ans1)`。輸入值必須是實數，數值大小不限。但為了測試方便，批改程式會將輸入資料限制在正負一億之間。

以下是一個簡單的範例：

```
ans1=0.112
print("%0.6f" % ans1)
```

執行結果是：

```
0.112000
```

(b) 標準常態的累積機率分布函數常出現在各種應用中。比如說在統計分析中求 p-value 就會用到標準常態的累積機率分布函數。由於這個函數沒有明解 (closed form solution)，實作時常採用函數逼近的方法。一個常用的逼近函數是：

$$\mathrm{N}(x) = \begin{cases} 1 - f(x)(a_1 k + a_2 k^2 + a_3 k^3 + a_4 k^4 + a_5 k^5), & when\ x \geq 0 \\ 1 - N(-x), & when\ x < 0 \end{cases}$$

其中 $\mathrm{k} = \frac{1}{1+\gamma x}, \gamma = 0.2316419,$

$$\mathrm{a}_1 = 0.319381530$$

$$\mathrm{a}_2 = -0.356563782$$

$$\mathrm{a}_3 = 1.781477937$$

$$\mathrm{a}_4 = -1.821255987$$

$$\mathrm{a}_5 = 1.330274429$$

請寫一個函數計算標準常態累積機率分布函數。你的程式需由使用者獲得 x 的值，並輸出函數值。由於 PDOGS 是檢查結果是否完全一樣，我們統一規定輸出至小數六位，四捨五入，不足補零。輸入值必須是實數，數值大小不限。但為了測試方便，批改程式會將輸入資料限制在正負一億之間。

下面是本題的範例輸出入。

(a)

範例輸入

```
0
```

範例輸出

```
0.398942
```

範例輸入

```
0.1
```

範例輸出

```
0.396953
```

範例輸入

```
-0.2
```

範例輸出

```
0.391043
```

(b)

範例輸入

```
0.1
```

範例輸出

```
0.539828
```

範例輸入

```
-0.3
```

範例輸出

```
0.382089
```

範例輸入

```
1
```

範例輸出

```
0.841345
```

評分原則

(a)小題 10 分,(b)小題 20 分。本題分數都由程式運算的正確性給分，一筆測試資料佔 2 分。

第三題

(30 points) 內部報酬率 (Internal Rate or Return; IRR) 是一個評估投資報酬的方法。其基本的想法是找出一個讓投資專案現金流淨現值 (Net Present Value; NPV) 為零的折現率。舉例來說，一個投資專案期初 (第零期) 投資 123400 元，第一期至第三期分別回收 36200, 54800, 與 48100。則此投資案的內部報酬 r 須讓 NPV 為零：

$$\text{NPV}=-123400+\frac{36200}{1+\text{r}}+\frac{54800}{(1+\text{r})^2}+\frac{48100}{(1+r)^3}=0$$

本例子的 r=0.059601，即為本投資案的 IRR。

一般而言，對一個現金流為$\text{C}_0, C_1, .., C_n$的投資案而言，其 IRR 須滿足

$\text{NPV}=0=\sum_{\text{i}=0}^{\text{n}}\frac{C_i}{(1+r)^i}$。

因此要計算內部報酬率，需要能找出滿足上式的 r。本題將利用二分逼近法來求 IRR。我們的目標函數$f(r) = \sum_{i=0}^{n} \frac{C_i}{(1+r)^i}$。假設$0 \le r \le 1$，因此我們的起始兩端點為 a=0, b=1。你應該要確認 f(a)跟 f(b)是一正一負。如不滿足則不應繼續，並請輸出"stop"。接下來二分逼近法的步驟如下：

1. 計算 a 跟 b 的中點 c = 0.5 * (a + b).
2. 計算 c 的函數值 f(c).
3. 如果收斂條件達成 (亦即 a - c 的絕對值很小 (<TOL), 或 f(c) 的絕對值很小 (<TOL); TOL 是一個很小的正數), 停止計算。這個 c 為所求。
4. 檢查 f(c) 的正負號，將 (a, f(a)) 或 (b, f(b)) 取代 (c, f(c))，好讓新的區間的兩個端點依然一正一負。

請設 $TOL = 10^{-4}$ 且二分逼近法最多重複 1,000 次。你的程式需由使用者獲得一個逗點分隔的現金流。這個現金流視為由第零期開始，計算出 IRR 之後輸出。由於 PDOGS 是檢查結果是否完全一樣，我們統一規定輸出至小數六位，四捨五入，不足補零。例如，如果 ans1 是你的答案，則可以這樣做：`print("%0.6f" % ans1)`。

現金流有可能是正 (獲得) 或負 (付出)。一般的狀況第零期是負數，後面會有一些正數。一個典型的狀況是第零期為一個絕對值較大的負數，後面都是正數，而且第一期與之後的現金流加起來應該要大於第零期現金流的絕對值。這個設定通常會讓 $r > 0$。至於會不會滿足$r < 1$則要看實際的數值。輸入的現金流每筆都必須是實數，數值大小不限。但為了測試方便，批改程式會將輸入資料限制在正負一億之間。

以下是一個簡單的範例：

```
ans1=0.112
print("%0.6f"% ans1)
```

執行結果是：

```
0.112000
```

下面是本題的範例輸出入。

範例輸入：

```
-123400,36200,54800,48100
```

範例輸出：

```
0.059631
```

範例輸入：

```
-312225,92318,58633,79130,124224
```

範例輸出：

```
0.049255
```

範例輸入：

```
-127987,31770,94628
```

範例輸出：

```
stop
```

評分原則

本題總分中的 24 分會依據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、輸入測試資料，並檢查輸出答案的正確性。一筆測試資料佔 2 分。

剩下的 6 分會根據你所寫的程式品質來給分。助教會打開你的程式碼並檢閱你的程式的運算邏輯、可讀性，以及可擴充性。請寫一個「好」的程式吧！