Naruaki TOMA <tnal@ie.u-ryukyu.ac.jp>
2013年5月9日 20:53
宛先: "slab34-dm@ie.u-ryukyu.ac.jp" <slab34-dm@ie.u-ryukyu.ac.jp>
返信先: slab34-dm@ie.u-ryukyu.ac.jp
[slab34-dm:18] (Tips) listの各要素に何か処理したlistを作る, オプション課題

當間@情報工学科です。

Leve2 で補足し忘れてたのを思い出しました。

「リストの各要素に何かしら処理して別のリストにする」ケースでは、
リスト内包表記を使うと便利です。

例えば、今回の課題は「data[][] を縦に串刺ししてデータを抽出する」
ことがゴールですが、これは data[x][0] や data[x][1] といった
リストの各要素にアクセスしながらコピーを作成する処理になります。
これを内包表記で書くと、以下のように書けます。

```
def collect_attr(data, index):
  list = []
  list = [data[x][index] for x in range(len(data))]
  return list
```

関数にする必要あるか分からないぐらいですが、実質的な処理は1行で書けます。
別解としては、（関数ではないですが）lambda で以下のように書くこともできます。

```
collect_attr = lambda i:[data[x][0] for x in range(len(data))]
new_data = collect_attr(0) #0番目の特徴だけを抽出。
```

これがどういう時に便利かを感じる例をいくつか示してみます。

(例1) iris のサンプルデータのうち、1つ目の特徴は必要ないことが分かったので除外したい。

```
data = iris.data
data2 = [data[x][1:] for x in range(len(data))]
```

(例2) 標準偏差を求めたい。

```
# 以下は iris の1つ目の特徴について求める例。
# Σ(x_i - average) のように「要素毎に処理する」という単位を、
# そのまま1行で書きやすいです。
data = iris.data
data0 = [data[x][0] for x in range(len(data))]
ave = sum(data0) / len(data0)
diff = [data0[x]-ave for x in range(len(data))]
multi = [diff[x]*diff[x] for x in range(len(diff))]
var = sum(multi) / len(multi)
import math
sd = math.sqrt(var)
print sd
```

==========
余裕があって試してみたい人は、
以下のオプション課題についてもやってみてください。

[Option 課題: データセットの統計情報を確認してみる]
iris.DESCR には特徴毎に
「最小値、最大値、平均、標準偏差、(クラスとの)相関係数」
が示されている。それぞれを算出する関数を作成してみよう。
iris.DESCR に答えがある（四捨五入されてる点に注意）ので、
それを使って doctest しながら書いてみよう。

なお、相関係数は一般的なピアソン相関
　　http://ja.wikipedia.org/wiki/相関係数
で求められます。

Naruaki Toma
E-mail: tnal@ie.u-ryukyu.ac.jp, Tel: 098-895-8830
http://www.eva.ie.u-ryukyu.ac.jp/~tnal/