2019.6.25
山本

scoringパッケージ

1　scoringパッケージの概要

(このreadme.mdをtexでpdfにしたファイルがmanual/scoring_manual.pdfにある)
採点作業の流れの中で以下のような作業をscoringパッケージはサポートする。

1. マークシートの採点
    - マークシートの読取結果(csvファイル)を採点し、学籍番号と点数のcsvファイルを作成する
2. 統合
    - マークシート以外の成績情報を統合する(例えば、宿題や筆記問題の採点結果などとの統合)
3. 分析
    - 以下のような統計データを出力できるので、それを見て分析する
    - 各問題の正解割合, 評語(A+ABCD)の割合, 点数のヒストグラム, 基本統計, 学類別基本統計
4. 点数調整
    - 点数調整を行って配点を調整する
5. Twinsへの成績アップロード用ファイルの作成
    - 点数調整を行った最終結果をtwinsのアップロード用ファイルに統合する
6. 記録: twins名簿をベースに記録用のファイル作成

2　install

installは以下の2つのファイルを正しい場所に配置すればよい。

- src/scoring.py: pythonのプログラム(モジュール)なので、pythonのライブラリパスが通っているところにコピーする。
- bin/score: scoringパッケージを呼び出すbashシェルなので、PATHが通っているところにコピーする。scoringパッケージのすべての機能はこのscoreコマンドを通して提供される。

make installで配置したい場合はMakefileの中の以下の2つの変数を正しくセットしてから、make installする。

```
pylib = scoring.pyを入れるディレクトリ(普通は$PYTHONPATH)
bindir = scoreコマンドを入れるディレクトリ($PATHのどれか)
```

前提条件は以下。

- python3.6以上
- pythonのpandasモジュール

3　scoreコマンドの使い方

scoringパッケージのすべての機能はscoreコマンドを経由して提供される。すべての成績データ(入力・出力・途中結果)は学籍番号を必ず含むcsvファイルで表現される。scoreコマンドは少なくとも1つのcsvファイルを扱うので、1つのcsvファイルは必須引数である。基本的に出力はstdoutにcsvの形式で加工された成績が出力される。また、分析用の情報はstderrに出力される。

```
$ score -h
usage: score [-h] [-marksheet ref-file desired_pscore] [-crate]
             [-twins] [-join csvfile2] [-record csvfile2 [csvfile2 ...]]
             [-adjust x y xmax] [-interval min max]
             [-distribution] [-abcd] [-statistics] [-gakuruistat csv-meibo-utf8]
             [-nostdout] [-output filename]
             csvfile
```

オプション名は区別が可能な限り短縮してよい。‘-adjust’と‘-abcd’オプション以外はすべて1文字で区別できる。
必須引数であるcsvfileは次のように4種類に解釈される。

- -marksheetオプションが指定された場合
  - マークシートの読取結果(学生の解答ファイル)とみなされる
    - headerなし
    - 1列目が学籍番号(下7桁), 2列目以降はマークシートのマーク番号
- -twinsオプションが指定された場合
  - twinsからダウンロードした成績アップロード用csvファイルとみなされる
    - twinsからダウンロードしたままのファイルでOK
    - headerあり(ただし、読み込み時には無視する(sjis&ラベルに変なスペースが入っているため))
    - 列は左から’科目番号‘,’学籍番号‘, ‘学期区分’,’学期評価’, ‘総合評価’
- -recordオプションが指定された場合
  - twinsからダウンロードされた名簿ファイルとみなされる
    - twinsからダウンロードしたままのファイルでOK
    - ただし、‘csv’と‘utf8’を指定してダウンロードしたファイルに限る
- 上記以外
  - 採点途中結果のcsvファイル
    - headerなし
    - 1列目が学籍番号(9桁), 2列目が点数

結果的に、上の3つのオプションは排他的である。

scoreコマンドの出力は‘-twins’または‘-record’オプションが指定された場合を除いて、すべて「採点途中結果のcsvファイル(headerなしの1列目が学籍番号(9桁),2列目が点数)」である。

以下では、作業の種別毎にオプションを説明する。

3.1 マークシート採点

3.1.1 score csvfile -marksheet ref-file desired-pscore [-crate] ...

csvfileをマークシート読み取り結果ファイルとして読み込み、採点を行う。マークシートの正解を定義したファイル(ref-file)を引数として与える。また、マークシートの満点(desired-pscore)を与える。正解定義中の各問題のweightにしたがって各問題に配点する(配点は整数なので要求した満点と少し違うこともありえる)。-twinsと-record以外の全てのオプションを同時に使用可能。チェック用として各学生・各問題ごとの正誤(正=1, 誤=0)のcsvファイルを以下の名称で保存する。

- $(csvfile).marubatu

マークシート採点のための正解定義は5.2節参照。学籍番号はすべて頭に20が付けられるので(200000000を足す)、必ず下7桁になっている必要がある。

-crateオプションは、-marksheetオプションが指定されているときだけ有効なオプションで、各小問毎の正解率、配点、問題タイプをstderrに出力する。

3.2 統合とtwinsアップロードファイルの作成

3.2.1 score csvfile -join csvfile2 ...

必須ファイルcsvfileで指定されたデータ[*1]にcsvfile2で指定されるデータを統合(点数の合算)する。統合は、keyとして学籍番号を用い、pandasのmerge関数を用いる。統合(join)方法は以下の2種類。

- -twinsオプションが指定されている場合:
  - twinsアップロードファイルへleft-joinを行う。
    - twinsのファイルにない学生は取り除かれる
    - 取り除く学生の学籍番号はstderrに出力される
- -twinsオプションがない場合:
  - outer-joinを行う
    - 片方のデータにしかない学生も残る。
    - データがない方の学生の点数は0点

-twinsオプションがある場合の出力はtwinsへのアップロード用のファイルとして出力される。

- headerがshift-jisでないといけないので-outputオプションでファイルを指定する。
  - 標準出力に出すとなぜかshift-jisにならない☹
- twinsデータの'総合評価'に数値が入っている場合はそれとcsvfile2の点数の合計が'総合評価'になる
- twinsデータの'総合評価'が空の場合はcsvfile2の点数が'総合評価'になる

-joinオプションの引数であるcsvfile2のファイルの仕様は以下。

- headerなし
- 1列目: 学籍番号
- 2列目以降(いくつでもOK): 評価点
  - 2列目以降の点数は合計される
  - 統合後にcsvfileとcsvfile2の点数は合計される

他のあらゆるオプションを同時使用可能。

3.3 点数調整

点数調整は、マークシート採点後または統合後、あるいはどちらも指定されてない場合はcsvfileで与えられた処理途中のデータに対して適用される。

3.3.1 score csvfile ... -adjust x y xmax

素点xをyに線型に持ち上げる。xmax以上は素点のまま。図1を参照。

3.3.2 score csvfile ... -interval min max

点数の最低点をmin、最高点をmaxにする。単に範囲からはみ出した点数をminとmaxに置き換えるだけ。線型変換などはしない。

3.4 分析

分析は点数調整後の成績に対して行われる(すなわち出力ファイルに対する分析である)。以下のような種類(オプション名)があり、同時に指定できる。出力はすべてstderrに出力される。

[*1] -marksheetオプションが指定されている場合は、採点結果に対して適用される。

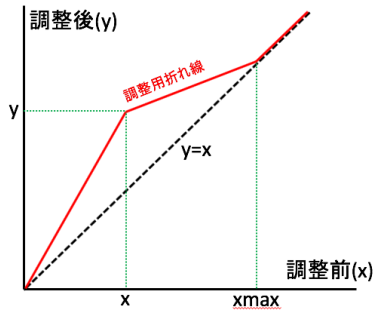

図1　-adjust x y xmaxの点数調整

- -distribution: 点数分布のヒストグラム出力
- -abcd: 標語(A+, A, B, C, D)の人数分布
- -statistics: 平均, 標準偏差, 最高・最低点, 4分位点
- -gakuruistat csv-meibo-utf8: 学類毎のstatistics
  - csv-meibo-utf8はtwinsからダウンロードした名簿ファイル
    * csvでかつutf8を指定してダウンロード
    * ('-record'オプションで与える名簿と同じファイルでOK)

-recordオプションを除くあらゆる他のオプションと同時併用できる。最終結果(すなわち出力データ)に対する分析を行う。先に行われるのは、マークシート採点、統合、点数調整である。いずれも指定されていない場合は、入力としてのcsvfileがそのまま分析対象となる。-twinsオプションが指定されている場合は、twinsへのアップロードファイルの分析が行われる。

点数調整と分析は同時に指定しながら調整するのが普通の使い方と思われる。

3.5　記録

採点途中の結果を学籍番号・氏名付きのファイルにまとめて記録用のファイルを作成するオプションが以下の-recordである。

- score csvfile -record csvfile2 [csvfile2 ...] [-output excel-filename]

このオプションが指定されると、必須引数csvfileはtwinsからダウンロードされた名簿ファイルとみなされる。これは-gakuruistatの引数と同じファイルで、twinsからダウンロードしたままの名簿ファイルでOK。ただし、'csv'と'utf8'を指定してダウンロードしたファイルに限る。以下のような処理を行う。

- 上記の名簿にこのオプションの引数csvfile2(複数指定できる)をouter-joinする。
- 点数の合算などはしない
- csvfile2は必ず1列目が学籍番号でなければならない(csvfile2にtwinsファイルは指定できない)。その他の列は文字列でもなんでもかまわない。
- '-output'オプション以外のすべてのオプションが無視される。
- '-output'オプションを指定するとExcelファイルが-outputオプションで指定したファイル名のファイルとして出力される。
  - この場合、filenameの拡張子はxlsxがよいと思われる。
  - Headerは名簿ファイルの項目についてしか入ってないので、出力後に各列の意味を手動で入力する必要がある。
- '-output'オプションがない場合は、utf8のcsvファイルとしてstdoutに出力される。

3.6 その他

その他のオプションは以下。

- -nostderr: 標準出力に結果のcsvファイルを出力しない
  - 調整時にこれを指定することが多い
- -output filename: 結果を標準出力ではなくファイルに出力する
  - twinsアップロード用ファイルはこのオプションで出力しないとheaderが正しくshift-jisで出力されない
  - '-record'オプションが指定されている場合は、Excelファイルが出力される。

4 使用例

4.1 使用例1

ある科目の使用例。この科目では期末試験でマークシート問題と記述問題の2種類の解答をしてもらった。また、宿題の点数が別に定義されている。

1. ファイルの準備
   - answer.csv: マークシートの読み取り結果
   - sup.csv: 記述問題の採点結果と宿題の点数(1列目が学籍番号, 2列目と3列目にそれぞれの点数)
   - upload.csv: twinsからダウンロードした成績アップロード用ファイル
     - ファイル形式の指定はできない(csvでshift-jis) (そのままでよい)
   - meibo.csv: twinsからダウンロードした名簿ファイル
     - csvかつutf8を指定してダウンロード
   - refrence.py: マークシートの正解と配点重みを定義したファイル
     - pythonのlist形式の記述なので拡張子をpyとしているが、気持ち悪ければtxtでもOK
2. マークシートの採点
   - 小問の正解率と満点を80としたときの配点等の情報を保存
     - $ score answer.csv -m reference.py 80 -c -n &> crate.txt
   - マークシートだけの得点状況を分析
     - $ score answer.csv -m reference.py 80 -s -d -abcd -n
     - 分析結果を残したい場合は最後に&> file.txtを付ける
     - この結果から正解の配点重みを変更してもよい
   - (マークシートだけの得点を少し調整する場合)
     - $ score answer.csv -m reference.py 80 -d -adjust x y xmax -n
   - 採点結果を書き出す
     - $ score answer.csv -m reference.py 80 [-adjust x y xmax] > mksheet.csv
3. マークシート以外の点数を統合
   - sup.csvを統合する
     - $ score mksheet.csv -j sup.csv > mksheet_sup.csv
4. 得点調整を行う
   - 例えば、-adjustオプションを使って調整する
     - $ score mksheet_sup.csv -abcd -d -adjust x y xmax -n
     - (最低・最高点が[0,100]をはみ出す場合は-i 0 100を加える)
   - 結果を書き出す
     - $ score mksheet_sup.csv -abcd -d -adjust x y xmax > mksheet_sup_adjust.csv
   - (上記の2つの手順は同時に行うこともできる)

    - $ score mksheet.csv -j sup.csv -abcd -d -adjust x y xmax > mksheet_sup_adjust.csv
5. twinsのアップロードファイルを作成
    - $ score upload.csv -t -j mksheet_sup_adjust.csv -d > twins_upload.csv
    - (最低・最高点が[0,100]をはみ出す場合は`-i 0 100`を加える)
6. 特別処理が必要な学生に対してtwins_upload.csvを直接修正
7. 最終結果に対する分析結果を保存
    - $ score twins_upload.csv -t -s -g meibo.csv -d -abcd -n &> result_analysis.txt
8. 氏名、学籍番号と採点途中結果を履歴として残す
    - $ score meibo.csv -record mksheet.csv sup.csv mksheet_sup.csv mksheet_sup_adjust.csv -o matome.xlsx
    - `matome.xlsx`に手作業でラベルなどを付ける

4.2 使用例2

`test`ディレクトリ以下に簡単なデータが準備してある。準備されているデータは例として参考にしてください。`test`ディクレトリで`auto.sh`を起動すると使用例1の簡単化された手続きが実行される。結果の例が`test/test-ref`ディレクトリ以下にあるので正しく動作しているかを確認のためにも使える。

5 付録

5.1 準備作業

必要な準備作業を以下にまとめる。

5.1.1 マークシート読み取り

学類所有のマークシート読み取り装置を使って読み取る。前提としているマークシートは以下。

- 教育ソフトウェア社の「総合カード053」(型番: C053)
    - 学生番号欄: 7桁(なので、学籍番号下7桁をマークさせる)
    - 解答欄: 「1–9,0」の0–9の数をマークできる列が50列
        * 上から1,2,3,…,9,0の順

学籍番号のマークをミスる学生がいるので手で修正(2019年度のある科目では、320人中2人の学生がおかしな番号を入れていた)。解答欄は修正しない。結果として、1列目が学籍番号(下7桁)、2列目以降がマークシートの各列の解答番号となっているcsvファイルができればよい。

■

マークシート読取手順　以下のようにしてマークシートの解答をcsvファイルにする。

- 学類のマークシート読取装置と専用ノートパソコンを接続・起動
- 「MarkView」というソフトを起動
- 「シート読取り」を選択
- レイアウト切り替え(「総合053(濃度差有り).LAY」というレイアウトを選択)
- ファイル→読取りデータ保存先の変更
    - これでファイルが初期化されるみたい
- 「読取」をクリック(押すたびに累積される)
    - 100枚くらいは1度に読み取れる(実績89枚は1度でいけた)
    - 明示的なsave命令はない。自動で保存される。
- ソフトを終了

5.1.2　成績アップロード用ファイル

成績アップロード用の履修者名簿をtwinsからダウンロード。処理の最後にこのファイルに成績を統合する。headerがshift-jisであるが、そのままでよい(scoringパッケージはheaderを読み飛ばす)。アップロードファイルを出力する場合はshift-jisのheaderを`score`コマンドが付ける。

5.1.3　名簿

twinsから名簿をダウンロード。形式はcsvとutf8を指定してダウンロードする。このファイルは`-gakuruistat`オプションに与え、学籍番号から所属学類を決定するために使われる。また、`-record`オプションで氏名付きの記録ファイルを作成するときにも使われる。

5.1.4　正解の定義と配点重み作成

マークシート読み取り結果に対する正解と配点重みを定義するファイルを作成する必要がある。以下の種類の解答に対応。

- `S`: 選択肢から1つ選択
- `SS`: 選択肢から1つ選択の連続
- `MS`: 選択肢から複数選択(順不同)
- `Num`: 数値。複数桁もOK

それぞれの種類について、マークシートの位置と正解を定義。配点重みは指定しなければdefaultの100(%)が各小問に設定される。配点重みを調整したい場合は100(%)を基準に上下する割合を指定する。詳しくは5.2節を参照。

5.2　マークシートの正解定義

マークシート問題の正解定義はpythonのlist形式のファイルを作成し、ファイル名を指定する。リストの要素が各問題の定義である。各問題は次のような4種類である。以下の「列番号」とはマークシートの列の位置である(1から50)。

- `S`: 選択問題
  - Format: [S, 列番号, 正解 (, 配点重み)]
- `SS`: 選択問題の連続
  - Format: [SS, [列番号start, 列番号end], [正解1, 正解2, …] (, 配点重み)]
    - 列番号startから列番号endまでの連続した列番号に対して、正解1,2,…を正解とする。
  - 各列番号は独立の小問と見なされる
    - 配点重みは各小問にそのまま適用される(小問数で割ったりしない)
- `MS`: 複数選択問題
  - Format: [MS, [列番号1, 列番号2, …], [正解1, 正解2, …] (, 配点重み)]
    - 正解は順不同
    - 解答に同じ答えがある場合は1つしか正解としない
  - **正解**の数だけ小問があるとみなす
    - 配点重みは各小問についてそのまま適用される(小問数で割ったりしない)
  - 列番号の数と正解の数は一致しなくてもよい(**正解**の数だけ小問が定義される)。
    - 選択する正解の数を指定しない場合(多めに列番号を指定しておく)
    - 正解の方が列数より多い場合もありえる
- `Num`: 数値問題
  - Format: [Num, [列番号1, 列番号2, …], [正解1, 正解2, …] (, 配点重み)]
    - 正解の順番も一致しないと誤りと判断
  - 数値全体がすべて一致してこの小問1問正解とみなす

「配点重み」は省略されている場合100となり、100でない場合は100を基準とした割合と解釈される。

例えば以下の通り。4つ目のMS問題だけ3つの各小問を配点重み50の割合いとしている。他の(小)問の配点重みは指定していないのでdefaultの100となる。実際の配点は-marksheetオプションで指定した満点に対する配点重みの割合で決定される(整数化するため指定した満点と少し誤差が出ることがある)。

正解定義ファイルの例

```
[
    [S, 1, 1],
    [S, 2, 3],
    [S, 3, 6],
    [MS, [4,5,6], [2,4,5], 50],
    [Num, [7,8,9,10], [1,0,4,0]],
    [Num, [11,12,13], [5,4,0]]
]
```

最初の3行は[SS, [1,3], [1,3,6]]と同じである。ただし、SSを使うと小問毎の配点重みが同じになるので、小問毎に配点重みを変えたい場合はSで記述する必要がある。

以上