CSVファイル

データベースと情報検索　第8回

横浜市立大学　坂巻顕太郎
kentaro.sakamaki@gmail.com

エクセルで開いたCSVファイル

	A	B	C	D	E	F	G	H
1	Date	Hour	Age	Sex	Number	Area	Amount	Result
2	2015/1/5	9	48	2	4	1	25200	0
3	2015/1/5	9	43	1	5	1	40800	0
4	2015/1/5	9	47	1	1	1	23800	0
5	2015/1/5	9	45	1	1	1	29500	0
6	2015/1/5	9	34	1	2	1	26700	1
7	2015/1/5	9	35	2	3	1	38000	0
8	2015/1/5	9	68	2	3	1	30900	1
9	2015/1/5	9	30	1	2	1	52100	1
10	2015/1/5	9	36	1	1	1	40400	1
11	2015/1/5	9	38	2	1	1	22400	0

メモ帳で開いたCSVファイル

Kaden.csv - メモ帳

ファイル(F)　編集(E)　書式(O)　表示(V)　ヘルプ(H)

```
Date,Hour,Age,Sex,Number,Area,Amount,Result
2015/1/5,9,48,2,4,1,25200,0
2015/1/5,9,43,1,5,1,40800,0
2015/1/5,9,47,1,1,1,23800,0
2015/1/5,9,45,1,1,1,29500,0
2015/1/5,9,34,1,2,1,26700,1
2015/1/5,9,35,2,3,1,38000,0
2015/1/5,9,68,2,3,1,30900,1
2015/1/5,9,30,1,2,1,52100,1
2015/1/5,9,36,1,1,1,40400,1
2015/1/5,9,38,2,1,1,22400,0
2015/1/5,9,50,2,2,1,23800,1
```

CSV: Comma-Separated Values

- （リレーショナル）データをやりとりする際のデファクトスタンダード
  - いくつかのフィールド（項目）に分かれるテキストデータ
    CSVの実質はプレーンテキスト
  - カンマ（,）を区切り文字とするデータ形式
  - 拡張子は .csv

- 類似したフォーマット
  - tab-separated values: 区切り文字がタブ（　　）
  - space-separated values: 区切り文字が半角スペース（ ）
  - Semicolon Separated Values: 区切り文字がセミコロン（;）
    - 小数点にピリオドではなくカンマを用いる文化圏で使用される

RFC 4180：CSVの国際標準の使用

- １行は1レコードからなる
  - レコードは改行文字列（CRLF、U+000D U+000A）で区切る
  - ファイルの先頭にレコードと同一の書式の「ヘッダ行」があってもよい

- レコードは１つ以上の同じ個数のフィールドからなる
  - フィールドはコンマ（U+002C）で区切る
    - 空文字列（長さ0の文字列）からなるフィールドもある
    - 最後のフィールドの後にはコンマはつけない
  - フィールドはダブルクォート（U+0022）で囲んでも囲まなくてもよい
    - フィールドが、コンマ、ダブルクォート、改行を含む場合は、かならずダブルクォートで囲む
    - フィールドに含まれるダブルクォートはダブルクォートを2つ並べてエスケープする

csvファイルの例

- 以下のデータのcsvファイルをエクセルで作成せよ

```
青木,93,67,82,88
井上,67,80,76,73
上田,83,78,84,75
遠藤,86,100,88,76
大野,70,75,67,63
```

- 作成したcsvファイルをod コマンドで確認せよ

```
od -Ax -tx1z test.csv
```

「ヘッダ行」のあるデータ

名前, 国語, 算数, 理科, 社会
青木,93,67,82,88
井上,67,80,76,73
上田,83,78,84,75
遠藤,86,100,88,76
大野,70,75,67,63

フィールドの表現（ダブルクォート）

名前, 国語, 算数, 理科, 社会
"青木", "93", "67", "82", "88"
井上,67,80,76,73
上田,83,78,84,75
遠藤,86,100,88,76
大野,70,75,67,63

空白列（データがない）

```
名前, 国語, 算数, 理科, 社会
"青木", "93",, "82", "88"
井上,67,80,76,73
上田,83,78,84,75
遠藤,86,100,88,76
大野,70,75,67,63
```

フィールドの数が合わない場合

```
名前, 国語, 算数, 理科, 社会
"青木", "93", "82", "88"
井上,67,80,76,73
上田,83,78,84,75
遠藤,86,100,88,76
大野,70,75,67,63
```

- 以上のcsvファイルを作成し、csv ファイルをcat コマンドなどで確認してみよ

ダブルクォートや改行がある場合

```
"名前","国語","算数","理科","社会","成績"
"青木
太郎",93,67,82,88,"""優"""
"井上
二郎",67,80,76,73,"""良"""
"上田
三郎",83,78,84,75,"""優"""
"遠藤
四郎",86,100,88,76,"""優"""
"大野
五郎",70,75,67,63,"""可"""
```

sortコマンド

- テキストファイルを「行単位で並べ替える」コマンド
  - デフォルトでは、数値も文字と同じように並べ替えられる
    - 1, 11, 100は、「1」→「100」→「11」の順番になる
  - 数値の順にしたい場合は、-nオプションを用いる

- CSVの並べ替え
  - -kオプション：空白文字を区切りとして、並べ替えに使うフィールド（列）を指定
  - -tオプション：使用する区切り文字を指定
    - CSVの場合、「,」を指定するため、「-t ,」または「-t “,”」とオプションを書く

- 区切り文字を「,」として、3番目のフィールドの値を数値として並べ替える
  - sort -k 3n -t , sampledata.csv

joinコマンド

- joinコマンド
  - 2つのテキストファイルの内容を比較し、共通する項目がある行を連結する
  - 1つ目のテキストファイルに「YCU2021001 青木」、2つ目のテキストファイルに「YCU2021001 93 67」という行がある場合、「YCU2021001 青木 93 67 」と出力

- オプション
  - -1 番号 : 1つ目のファイルで比較に使用する項目番号
    - -2で2つ目のファイルの指定
  - -a 1 (or 2) : 1つ目のファイルに対応するフィールドがない場合も出力

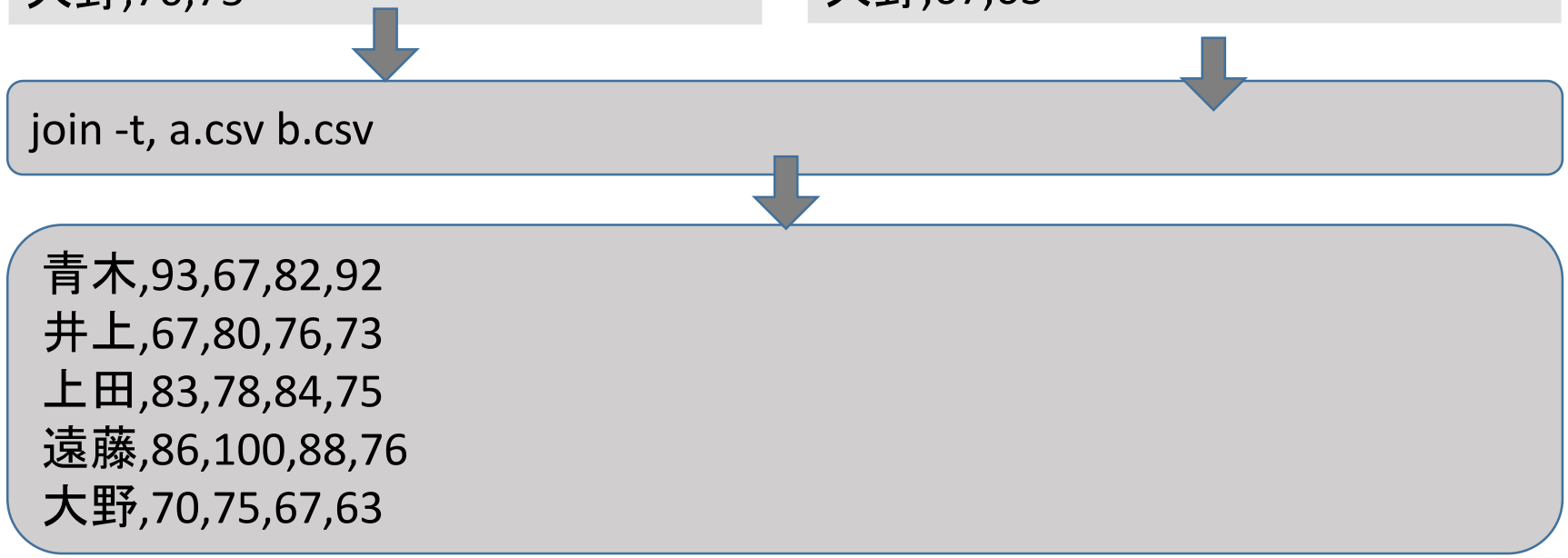
2つのファイルを統合する

a.csv

青木,93,67
井上,67,80
上田,83,78
遠藤,86,100
大野,70,75

b.csv

青木,82,92
井上,76,73
上田,84,75
遠藤,88,76
大野,67,63



練習

- 各行に名前が書かれたファイル、国語の点数が書かれたファイル、算数の点が書かれたファイルを用意し、1つのcsvファイルにまとめよ

- 出来たファイルを実際にエクセルで開いてみよ

特定のフィールドを取り出す

- cutコマンド
  - ファイルを読み込み、「それぞれの行」から指定した部分だけを切り出す
  - -f: 切り出すフィールド（列）の指定、-d: 区切り文字の指定

- 使用例
  - cut -d, -f 2 test.csv
    - 国語の点数だけを切り出す
  - iconv -f SJIS -t UTF8 test.csv | cut -d, -f 1
    - 名前のフィールドを切り出す
  - cut -d, -f 1-3 test.csv | iconv -f SJIS -t UTF8
    - 名前、国語、算数を切り出す
  - cut -d, -f 1,3 test.csv | iconv -f SJIS -t UTF8
    - 名前と算数を切り出す