2次元正規表現で
腐った表から情報を取り出す

田中 哲
Free Software Initiative of Japan (FSIJ)
Ruby Hiroba 2013
2013-06-02

残念ながら
世の中は Excel で
動いている

Excel は使いたくない

- 理由は自明
- しかたないので以下のようにする

CSV になれば楽勝
というわけではない

- 結合セル
- 罫線
- 複数シート

情報を取り出すのに苦労する

今日の例

- Excel による例は都合により用意できなかった
  (Excel が動くマシンは手元にない)
- かわりにでんき予報の CSV を使う
  クリエイティブな Excel ファイルに比べると
  問題は少ない

でんき予報の CSV

2013/6/2 13:25 UPDATE
ピーク時供給力(万kW),時間帯,供給力情報更新日,供給力情報更新時刻
3878,19:00～20:00,6/1,17:30

予想最大電力(万kW),時間帯,予想最大電力情報更新日,予想最大電力情報更新時刻
3090,19:00～20:00,6/1,17:30

DATE,TIME,当日実績(万kW),予測値（万kW)
2013/6/2,0:00,2396,0
…(中略)...
2013/6/2,23:00,0,0

翌日のピーク時供給力(万kW),時間帯,供給力情報更新日,供給力情報更新時刻
,,,

翌日の予想最大電力(万kW),時間帯,予想最大電力情報更新日,予想最大電力情報更新時刻
,,,

メッセージＮｏ,節電お願い文
,
,

DATE,TIME,当日実績（５分間隔値）(万kW)
2013/6/2,0:00,2470
...(後略)...

でんき予報の CSV の問題点

- ひとつのファイルにいくつも表が入っている
- CP932

興味がある部分を取り出したい

2013/6/2 13:25 UPDATE
ピーク時供給力(万kW),時間帯,供給力情報更新日,供給力情報更新時刻
3878,19:00～20:00,6/1,17:30

予想最大電力(万kW),時間帯,予想最大電力情報更新日,予想最大電力情報更新時刻
3090,19:00～20:00,6/1,17:30

DATE,TIME,当日実績(万kW),予測値（万kW)
2013/6/2,0:00,2396,0
…(中略)...
2013/6/2,23:00,0,0

ここを取り出したい

翌日のピーク時供給力(万kW),時間帯,供給力情報更新日,供給力情報更新時刻
,,,

翌日の予想最大電力(万kW),時間帯,予想最大電力情報更新日,予想最大電力情報更新時刻
,,,

メッセージＮｏ,節電お願い文
,
,

DATE,TIME,当日実績（５分間隔値）(万kW)
2013/6/2,0:00,2470
...(後略)...

例題

DATE, TIME, 当日実績(万kW) の 3つのフィールドを取り出したい
(予測値 (万kW) はいらない)

DATE,TIME,当日実績(万kW),予測値（万kW)
2013/6/2,0:00,2396,0
…(中略)...
2013/6/2,23:00,0,0

どうやって?

- ループと条件分岐でちゃんと書く?
  面倒くさい
- 正規表現で？
  行単位ならできそうだけどそうでない

2次元正規表現

- 正規表現エンジンは文字列の上でバックトラックしながら探索をおこなう
- 2次元空間をバックトラックしながら探索を行うものがあってもいいのでは?
- ふつうの正規表現は文字にマッチすると右に移動する
- かわりに上下左右に移動できるようにすればいいのでは？

作ってみた

- tb gem 内にライブラリとして入っている
- tb gem の主な機能ではない
- 主な機能は tb コマンド

サンプルプログラム

```
require 'tb'

aa = Tb.csv_read_aa(ARGF.read)

(x1, y1), (x2, y2) = Tb::Search.match(
  [:cat,
    "DATE", :e, "TIME", :e, "当日実績(万kW)",
    [:rep, :s, /\d+/]
  ], aa)

y1.upto(y2) {|y|
  x1.upto(x2) {|x|
    print aa[y][x].to_s
    print "\t"
  }
  puts
}
```

読み込み

マッチ

結果表示

実行結果

DATE　TIME　当日実績(万kW)
2013/6/2　0:00　2396
2013/6/2　1:00　2281
2013/6/2　2:00　2240
(後略)

マッチ部分

```
(x1, y1), (x2, y2) = Tb::Search.match(
  [:cat,
    "DATE", :e, "TIME", :e, "当日実績(万kW)",
    [:rep, :s, /\d+/]
  ], aa)
```

aa: 配列の配列

文字列: 同じ文字列にマッチ

正規表現: マッチする文字列にマッチ

:e 東へ動く　　:s 南へ動く

:cat 連結　　:rep 繰り返し

もっと複雑なことも可能

- 東西南北、北東、北西、南東、南西への移動
- 貪欲でない繰り返し
- 回数指定繰り返し
- キャプチャおよびキャプチャ文字列とのマッチ
- 現在位置の保存と復帰
- 表明
- 横型探索
- などなど

腐った表に悩まされている人は使ってみるといいかも

- きっかけはでんき予報ではく
  もっとずっと腐ったもののために作った
- 結合セルと罫線をとてもクリエイティブに活用した Excel ファイルを扱うために作った
- 結合セルと罫線を含めて CSV に変換するサンプルスクリプトも作った
  (複数シートも扱える)
- tb コマンドも便利です
  tb grep とか tb sort とか