報道実務家フォーラム

データは整形が７割 & 付き合わせてなんぼ

2017/05/19修正版

（１）はじめに

せっかく入手した貴重なデータを報道に生かす際、そのまま使えることはまれ。何かしら整形が必要になることが多い。時間の７割は地道な整形作業に費やされる、と思っておいたほうがいい。数値として統計的に処理したいのに、丁寧に単位を付けてくれているのが災いしたり、半角と全角が混在しているために、本当は同一のものなのに名寄せされなかったり。

これを手作業で直すのは大変。うんざりするし、ミスが紛れ込むかもしれない。パソコンの「中の人」に指示を出して機械的にやらせる方法があれば、積極的に活用すべし。それが、表計算ソフトのファンクションであり、正規表現と呼ばれるものだ。

ファンクションは関数と訳したから身構えるけれど、文字どおり、それこそ表計算ソフトの本来「機能」ってこと。正規表現も、名前がいかめしいだけで、何らかの共通パターンを見つけてＰＣに仕事をさせるためのメモだ。どちらも、「こういうときは、こうして」という命令をパソコンに理解させるための道具。１行だけの極小プログラムを書くんだぞ、という気持ちで取り組めば、なかなかに使える。食わず嫌いより、外国語をひとつ勉強すると思って、手を出したほうがお得なはず。

ところで、分析に使いたいデータが一度に入手できるとは限らない。複数の表を付き合わせて、計算して、初めて分かることだってある。その付き合わせに欠かせないのが、すべての表に共通する（そして「元帳」では一意の）KEY。ここでも、整形のテクニックが役に立つ。

（２）準備のお願い＝インストールとダウンロード

ＰＣを持ってくるのが苦にならない方は、なるべくお持ち下さい。無謀かもしれないが、各自、ＰＣを実際に操作しながら、という進行を予定している。ファイル名の末尾の「拡張子」（.txtとか.xlsxや.csvのたぐい）が表示される設定にしておいて下さい。ＰＣが得意なかたは、ぜひ、ご近所で困っているかたを巡回して、ヘルプにご協力お願いします。

当日使うファイルは、GitHub（https://github.com/nishioWU/WASEDA/）に。５月20日中には資料とデータ一式をアップロードしておくので、 >> Clone or download >> Download ZIP でダウンロードしておいて下さい。解凍にパスワードが必要なものについては、会場でお知らせします。

表計算ソフトが入っていないかたは、あらかじめLibre Officeのインストールを（使うのは、そのうちCalcのみ）。正規表現が使えるテキストエディタが入っていないかたは、ATOMのインストールまたは、regex101のURLのブックマークを。

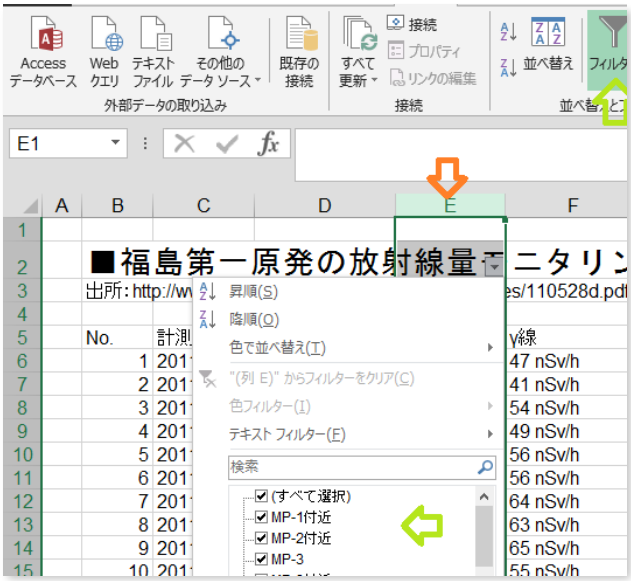
あと、TabulaとOpenRefineも入れておいて損はない。講義冒頭に簡単なデモを予定しています。着席したら、Tabulaを起動して下さい。ブラウザが起動して画面が変わるまで、１～２分

かかります。実例として使用予定のモニタリングデータの取り込みに挑戦してみて下さい。データのありかと方法は、「（５）実例の入り口」を見て下さい。

Libre Office＝表計算ソフトが入っていない人だけ

Excelなど普段使いのものがあれば、それで十分。講師もExcelを使います。ＰＣに入っていない、というかたのみ、下記の場所から「Libre Office」のインストールを。
https://ja.libreoffice.org/download/libreoffice-stable/

Calcと違い、素のままのExcelでは正規表現は使えない。しかし、フィルターや置換を使って、工夫次第でやれることはある。例えば「長島」→「長嶋」の一括変換とか、表記の揺れを検出するとか。



EXCELのフィルターを使って、放射線量データの計測地点の表記の揺れを調べているところ

ATOM

多機能なテキストエディタ（文書入力・編集のためのソフト）。「正規表現」を使って検索、置換ができる。コンピューターの基本文化として、あくまでヨコ方向（行）優位。お任せで文字化けしてしまったときは、「Shift+Ctrl+U」で文字コードを選び直すことが可能。欠点は動作が重たいこと。
https://atom.io/

regex101=URLをブラウザでブックマーク（インストールではありません）

正規表現の動作テストができるサイト。ATOMと同じ動きにするには、左上の設定から >> FLAVOR >> javascriptに。上の窓の末尾のoptionはg(global)とm(multiline)を選ぶ。
https://regex101.com/

Tabula

PDF形式のファイルを、表計算ソフトでも読めるcsv形式に変換する。必ず成功するわけではないが、かなり便利。TabulaもOpenRefineも、自分のＰＣ上に作ったサーバーに、ブラウザを使って接続する、という仕組みで動く。ブラウザが起動するが、外部にデータを送っているわけではないので、そこは安心を。終了は、両ソフトとも、ターミナル（Windowsなら黒い窓）で「Ctrl+C」。
http://tabula.technology/

下記のJREが入っていないと動かない。

OpenRefine

データの下処理に威力を発揮。正規表現が使えて、タテ方向（列）の一括変換も得意。とっつきにくいが、とても強力で役に立つ。Facet（自動フィルター）や名寄せ機能あり。
http://openrefine.org/download.html
これも下記のJREが必要。

JRE

TabulaもOpenRefineも、Java Runtime Environment(JRE)というものがないと動作しないので、入っていないなら、インストールを。１度入れればＯＫ（TabulaとOpenRefineで同じものを使う）。ただし、会社貸与のＰＣでは、JREのインストールに制限がかかっている場合もある。そのときは、TabulaとOpenRefineとJREは入れないままで結構。
https://java.com/ja/download/

（3）きょうのキーワード

csv形式

テキストファイルのうち、コンマでデータを区切った構成のもの。テキストエディタと表計算など、異なるソフト間でのデータの受け渡しに重宝する。Excelで扱う場合は、先にExcelを起動しておき、>> データ >> テキストファイル で対象のcsvファイルを選んで >> インポート。文字コード（UTF-8やシフトJIS）、先頭行は項目名かデータか、区切りに使われている文字（csvなので、タブではなくてコンマ）、の指定を忘れずに。

拡張子「.csv」がExcelに関連づけられていると、ファイルをクリックしたときにExcelが開きに行ってしまい、文字コードが違えば化ける。でも、上の方法なら大丈夫。左ではなく右クリックして、どのソフトで開くかを指定してやれば、テキストエディタなどで開くことも可能。

UTF-8

これからの標準の文字コード体系。Macはこちら。SHIFT-JISだと決めてかかってファイルを開くと文字化けする。

SHIFT-JIS

今まで広く使われてきた文字コード体系。Windowsは表面上はこの系統。

LF

改行コードの一種。正規表現だと「\n」。

CR

改行コードの一種。正規表現だと「\r」。Windowsは「CR」+「LF」（正規表現だと「\r\n」）で改行しているので、「LF」のみのファイルを読み込むと、区切り位置が変になっ

たり、Excelシートの１行目に無理やりすべてを入れにかかってエラーしたりすることもある。

絶対参照

表計算ソフトで数式をコピーすると、気を利かして、計算対象の行や列をずらしてくれる。普通はそれが便利だからだが、かえって困ることもある。そのときは、ずらされては困る行や列に「$」マークをつけておく。コピー先でも、元のままに固定される。

VLOOKUP

別々の表に分かれているデータの共通の項目を付き合わせ、新たな表を作るときに便利なExcelの「関数」。元帳の一番左の列に、一意の検索KEYがないとだめ。

IF

「このときは、こうして」「そうでなければ、こうして」と、条件によって処理を枝分かれさせるのに使う。

（４）取りかかる前に

ファイルは開ける？　文字化けしていたら…

csv（コンマ区切り）かtxt（テキスト）形式のデータが文字化けしているときは、エンコードの違いが原因。UTF-8かSHIFT-JISを試してみる。ATOMなら「Shift+Ctrl+U」で切り替えられる。自動検知機能もある。

Internet Exploreを使って、文字コードを変換して保存し直す手もある。割と役立つ。

① 拡張子が「.csv」の場合は「.txt」に変えて保存。ピリオドまで消さないように注意
② IEでファイルを開く。化けていれば、画面を右クリックしてエンコードを直す。たいてい、自動認識してくれる
③ 保存したい形式＝上記２通りのどちらか＝を選び、別名で保存。別名にしないと、元が消えてしまう
④ 必要なら拡張子を「.csv」に戻す。戻さなくても、表計算ソフトに読み込むことは可能

現行のChromeではこの方法は使えない。

文字コードは合っていてテキストエディタでは開けるのに、Excelでファイルを開くときに変になる、という場合は、改行コードだけの違いが原因のこともある。

原本保存と編集のバージョン管理

大事なファイルを書き換えてしまっては困る。原本はプロパティを「読み取り専用」に変更しておき、作業するたびにファイル名を付け替えて履歴を残していくことを鉄則にする。なお、「読み取り専用」にしておけば書き換えはできないが、削除はできてしまうので注意。

ちょっと待った

いきなりセルのデータを書き換えないで、隣に１列増やして、そこで試すようにする。表計算ソフトなら、列をコピーしてから触るとか。OpenRefineなら、>> Edit cells >> Transformではなく、>> Edit column >> Add column based on this columnを使うとよい。不要になった列は、>> Edit column >> Remove this columnで削除できる。それから、ソートする前に通し番号を振ること。元の順に戻すときに必要。数式が入っているセルは「値だけ」にしておく。

データの「乱れ」が起きるわけ

・雛型を設計した人や、入力した人の親切心
・表計算ソフトを割付用紙としてだけ使っている
・入力作業者が何人もいたり、長期間に渡ったりするための不統一
・誤変換やOCR時の文字化け
・改名やそもそもの表記の変更（スポーツ選手や会社の名）

などが考えられる。

助数詞や単位語がついていたり、略語・通称を使ったり、平仮名・カタカナ・記号が入れ替わったり（「へり」「ヘリ」／「ロート」「ロート」）、小数点とコンマの取り違え（または米欧の用法の違い）等々、自社の記事データベースを検索してみれば類例に事欠かないはず。

表計算ソフトで作られた元データであっても、作った人が「計算」に使っていないケースは危うい。印刷時のフォーマットに合わせることを優先して作ったファイルだったら、「セル結合」されていて読み取れなかったり、数字でないものが紛れ込んでいたりする可能性が高い。福島第一原発のモニタリングのデータでは過去に、数字のゼロでなく英字オーで入力したものがカナ変換されたと思われる「お」が入っていたことさえあった。詳しくは下記を参照のこと。

>> 奥村晴彦「ネ申 Excel」問題(https://oku.edu.mie-u.ac.jp/~okumura/SSS2013.pdf)

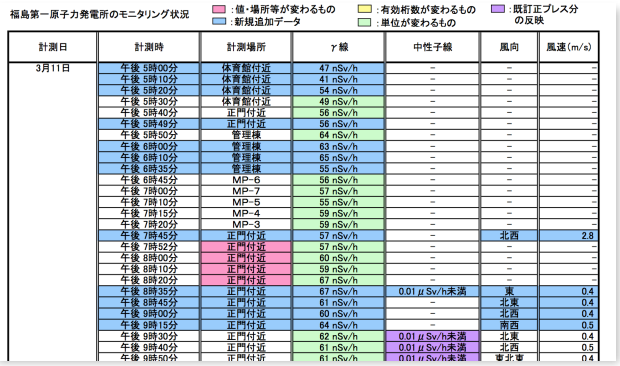
奥村先生のサイトや著作は、RやTeXなどついても、あらゆる点でとても参考になる。

（５）実例の入り口

日時の変換・単位を外す・表記の揺れ

データの実例として、放射線量モニタリングのデータを見てみる。

>> http://www.tepco.co.jp/cc/press/betu11_j/images/110528d.pdf


福島第一原子力発電所のモニタリング状況

：値・場所等が変わるもの　：有効桁数が変わるもの　：既訂正プレス分の反映
：新規追加データ　：単位が変わるもの

計測日	計測時	計測場所	γ線	中性子線	風向	風速(m/s)
3月11日	午後 5時00分	体育館付近	47 nSv/h	-	-	-
	午後 5時10分	体育館付近	41 nSv/h	-	-	-
	午後 5時20分	体育館付近	54 nSv/h	-	-	-
	午後 5時30分	体育館付近	49 nSv/h	-	-	-
	午後 5時40分	正門付近	56 nSv/h	-	-	-
	午後 5時49分	正門付近	56 nSv/h	-	-	-
	午後 5時50分	管理棟	64 nSv/h	-	-	-
	午後 6時00分	管理棟	63 nSv/h	-	-	-
	午後 6時10分	管理棟	65 nSv/h	-	-	-
	午後 6時35分	管理棟	55 nSv/h	-	-	-
	午後 6時45分	MP-6	56 nSv/h	-	-	-
	午後 7時00分	MP-7	57 nSv/h	-	-	-
	午後 7時10分	MP-5	55 nSv/h	-	-	-
	午後 7時15分	MP-4	59 nSv/h	-	-	-
	午後 7時20分	MP-3	59 nSv/h	-	-	-
	午後 7時45分	正門付近	57 nSv/h	-	北西	2.8
	午後 7時52分	正門付近	57 nSv/h	-	-	-
	午後 8時00分	正門付近	60 nSv/h	-	-	-
	午後 8時10分	正門付近	59 nSv/h	-	-	-
	午後 8時20分	正門付近	67 nSv/h	-	-	-
	午後 8時35分	正門付近	67 nSv/h	0.01μSv/h未満	東	0.4
	午後 8時45分	正門付近	61 nSv/h	-	北東	0.4
	午後 9時00分	正門付近	60 nSv/h	-	北西	0.4
	午後 9時15分	正門付近	64 nSv/h	-	南西	0.5
	午後 9時30分	正門付近	62 nSv/h	0.01μSv/h未満	北東	0.4
	午後 9時40分	正門付近	61 nSv/h	0.01μSv/h未満	北西	0.5
	午後 9時50分	正門付近	61 nSv/h	0.01μSv/h未満	東北東	0.4

このPDFをTABULAでCSV形式に変換する

「内容のコピー」を「許可しない」設定になっているので、このままでは手出しができない。そこで、Tabulaでcsv形式のデータに変換する。人の目で見ることを前提に作ったPDF。まず、ざっと全体を眺めておく。色分けによる注記は人間には親切だが…。γ線は途中で単位が変わっていること、中性子線はほとんど数値に変動がないこと、単位だけでなく空白や「未満」が入っていること、などが分かる。

以下、説明を簡単にするため、日時、計測場所、γ線だけに絞る。計測場所は、かつては表記の揺れが多かったが、今はそうでもないようだ。

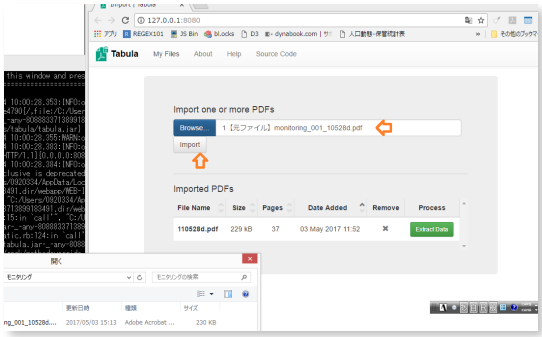
人間が読みやすいものと、コンピューターが扱いやすい形式は違う。人間には「読めてしまう」ものに、コンピューターは律義につまずく（つまずいてくれる）。太字、網掛け、カラーも、ＰＣにはそのままでは伝わらない。

ファイル原本は必ず保存しておく。以後、触るたびに名前を付け替え、履歴を保存していく。

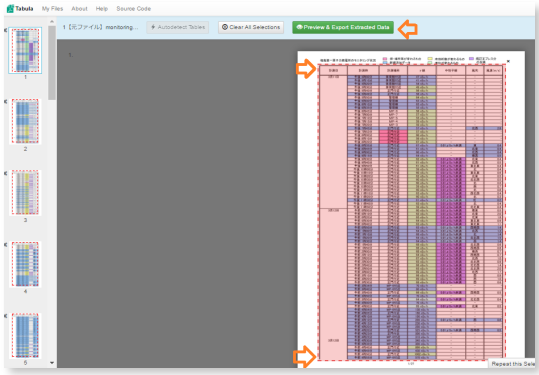
プレビューで問題なさそうなら、出力形式を「csv」にして（デフォルトで選ばれているはず）、>> Export する。できあがったcsvファイルを、テキストエディタで開いて確かめる。とりあえずは、ここまで。

あとで戻ってくる予定だが、時間切れになった場合は、「（１１）実例の続き」を見て下さい。OpenRefineの使い方も、末尾に付録でつけてあります。

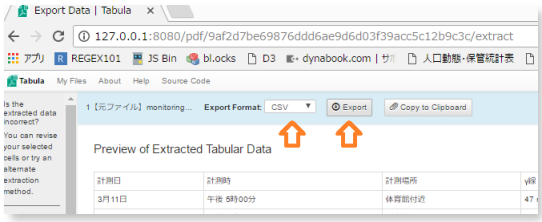
このような格子形式の表の場合、Tabulaのモードはまさに「lattice」がいい。試した限りでは、欄外の注釈やページ番号を含めないように、表の範囲を手動で指定してやったほうが、精度よく取り込まれる。自動検知に任せてみたら、タテの列（カラム）がずれたり、日付が抜けたりした。それをいちいち修正するよりも、範囲指定をやり直して再度スクレイピングしたほうが、断然速い。



TABULAを起動するとブラウザが立ち上がる。ファイル名を指定して >> IMPORT で読み込む



表の自動検知ではうまく取り込めないことが多い。欄外の注釈やページ番号は含めないように手動で範囲を設定する



出力形式はCSVを指定して >> EXPORT

（６）正規表現ことはじめ

ATOMかregex101で、パターンを捕まえる練習

コンピューターは大量のデータを扱うのが得意。野球選手1000人の打率の計算とか、今月起きた地震の震源を地図にプロットするとか。

でも、入手したデータが

```
選手名　　，試合数　，　打数　，　安打
鈴木　一朗　，40試合　，95打数　，24安打
長島　茂雄　,130試合　,502打数　,153安打
```

```
落合　博満　, 36試合　, 64打数　, 15安打
松井　秀喜　, 57試合　,184打数　, 41安打
```

こんなスタイルだったら、めげる。「安打数」割る「打数」で打率を計算しようにも、その前に「打数」や「安打」の文字を削って数字だけにしてやらなければならない。人間に読みやすいように配慮したのが邪魔をして、ＰＣ処理には不向きになってしまったケース。

ここで少しでも楽をしたい。それぞれの箇所を手作業で削るのではなく、ＰＣに「数字の後にコンマ以外の文字が続いていたら、それを削れ」と指示して、一括作業させれば、速いし間違いもない。このように、何らかの共通パターンを見つけて、コンピューターに指示を出すときに、「正規表現」を用いる。

ワープロや表計算ソフトの検索機能を使って、例えば「早稲田」を探したことなら、あるだろう。正規表現なら、単に対象の文字列を含むだけでなく、「早稲田」「早稲田大」の入った記事は探したいが、清宮くんフィーバーの「早稲田実」は除く、といった込み入った要望にも対応できる。「英字３文字と数字４桁がつながったもの」「0で始まる２〜４桁の数字が行頭にあるとき」といった、抽象的・一般化された条件をつけられるのが、正規表現のいいところ。文字の種類＋数＋場所、を組み合わせて指定する。詳しくは別紙のチートシートを。

ちなみに、上の例なら、

```
「早稲田[^実]」　　←末尾に*が付いていたら誤り。抜いてください
```

のように書くと、「早稲田」で終わるか、あとに「実」以外の文字が続くか、にマッチする。

ATOMの場合

メニューの >> Findまたは「Ctrl+F」で検索窓が開く。そのままだと通常の検索。右下の「.*」ボタンを押すと正規表現が使えるようになる。

regex101の設定

左上にある設定から >> FLAVOR >> javascript、上の小窓は「gm」(全体をスキャン)にしておく。こうすると、ATOMと同じ動きをする。下のSUBSTITUTIONの右の方にあるプラスマークを押すと、正規表現にマッチした部分を$1、$2、$3…に置き換えるモードになる。

【練習】正規表現１

練習用ファイルをATOMで開く。regex101の場合は、コピーしてペーストしてやる。

東京03地区の電話番号だけを取り出す

→行の先頭に「03」が来て、その後にハイフンが続いている（030-などをはじくため）、と考えれば

「^03-」

通常の検索に、行頭という位置指定が加わっただけ。キャレット（^）は半角で。

→念には念を入れて、ハイフンの後には４ケタ数字（市内局番）が来ているはずだ、と考えるなら

「^03-\d{4}-」

このように「数字なら何でも」とか「４文字ぴったり」といった指定ができる。

→市内局番の冒頭に0と1は来ない、市内局番の後もちょうど４ケタの数字が続いて計10ケタになる、ということまで考慮すると

「^03-[2-9]\d{3}-\d{4}(\D|$)」

→行の途中に出てきたものでもマッチさせたいなら、行頭または数字以外のものに続き、行末または数字以外のものが後に来る、と絞り込んで

「 (\D|^)(03-[2-9]\d{3}-\d{4})(\D|$)」

となる。切り出された電話番号は「$2」に保存されているので、置換する際に使える。

市外局番のハイフン区切りをやめ、（ ）で囲むスタイルに一括修整する

→行頭から始まっている数字で、先頭が0で２～５ケタ、後にハイフンが続くものが市外局番、と考えるなら

「^(0\d{1,4})-」を「($1)」に

置き換える。

→さらに、市内局番は１～４ケタで最初は0と1は来ない、最後のブロックはきっかり４ケタ、と厳しく絞り込めば

「 (\D|^)(0\d{1,4})-([2-9]\d{0,3}-\d{4})(\D|$)」を「$1($2)$3$4」に

置き換える。これは、行頭以外にあっても動作する。

ファイルの中身がすべて電話番号であると分かっていて、03地区分を抜き出すだけなら、最小限の条件をつければ十分。電話番号以外の数字も混在していたり文章中に埋もれたりしているなら、条件を増やして絞り込む必要がある。そこは臨機応変に。

【練習】正規表現２

名選手データから、桁揃えの半角空白や余分な漢字を取り除く
→ 正直に列挙すると

「(\d+)(試合|打数|安打)」を「$1」に

置換。分かりやすい。前に半角数字が最低 1 文字ある、と条件をつけているのがポイント。なので、項目名の行が削られることはない。

→発想を変えてみる。半角数字のあとに、それ以外のものが続いていたら、コンマか行末を残してその前を削る（結果的に、数字に付随する漢字部分を削ることになる）ようにすればいい。決まった漢字・文字数しか整形できない例よりも汎用性があり、使い回しが効く。

「\s*(\d+)[^\d,\r\n]*」を「$1」に

置換。桁揃えの半角空白も削っている。改行コードがCR+LFの場合と、LFだけの場合の両方に対応するよう、\ r も入れているが、LFだけならば不要。改行コードに何を使っているかは、ATOMの画面下に表示されている。

選手名の末尾の全角空白を取り除く
→全角空白は「\s」にはマッチしないので、「　」と入力する。姓名の間の空白は削りたくないので、こんな工夫を。コンマを添えて、条件を厳しくして

「　+,」を「,」に

置換。

（7）Excelの小技１＝$の活用とVLOOKUP

入力は原則小文字で

Excelのセルに数式を書く際には、引用符の中以外は、小文字を使うといい。関数名などの綴りの誤りがあれば小文字のまま残るので、気付きやすい。綴りが正しければ、大文字に変換される。それなのに望んだ通りに動作しないなら、文法の勘違いとか、目的に合わない関数を選んでいるとか、参照先のセルを間違えている、などが原因のはず。

＄マーク

絶対参照といわれているもの。計算式でセルを指定する際、「F4」を押すと、「$」のマークがついたり消えたりする。行だけ、列だけにつけることもできる。

セルを参照している数式（や、その数式が入っているセル）をコピーすると、デフォルトでは、参照先も一緒にずれていく。たいていは、それが便利だから。相対参照という。

でも、ここはずらしてくれなくて結構、余計なことしないで、という場合もある。そういうときは、行や列に「$」をつけると、コピーしたときも参照範囲が平行移動せずに、固定されるようになる。これが絶対参照。知らないと、ひどく苦労する。

行が増減しそうなときには、列全体を指定することもできる。「A:A」なら、Aカラム全体という意味になる。が、時として予期せぬ動作の原因にもなる。

VLOOKUP関数

「=VLOOKUP(イ, ロ, ハ, ニ)」という書式。KEYが共通するデータを元帳から探し、指定の列を取ってくる。イ、ロ、ハ、は必須、ニは省略ＯＫだが、そこが落とし穴。慣れるまでは省略せず、「FALSE」の決め打ちに！

一致するものが見つからなければエラー表示で注意喚起してくれる

・イ＝これと同じものを探せ

・ロ＝この範囲の、一番左の列で探せ。突き合わせ先の「元帳」の範囲

・ハ＝指定した分だけ、左端から１、２、３…と数えて、そのセルの中身を返せ

・ニ＝ここでは「FALSE」か「0」を指定（完全一致）。空欄は「TRUE」や「1」の意味になり、違う動作をするので、間違いのもと

「元帳」には「名前をつける」と楽。絶対参照で範囲指定される。この「元帳」は同じシートになくてもよい。検索キーが一番左の列でないときは、工夫が必要。並び替えるとか、MATCHとOFFSETやINDEXを組み合わせるとか。

IFと仲間たち

基本はIF。「=IF（イ, ロ, ハ）」の書式。ハは省略ＯＫ。

・イ＝この条件が成立しているか、真偽を調べて

・ロ＝成立している（つまり「TRUE」のとき）なら、セルの中身をこれに

・ハ＝成立していない（つまり「FALSE」のとき）なら、これに

COUNTIF、SUMIFなど、条件付きの処理をするIF仲間もある。便利。

条件つき書式

TOP10とか平均以上をハイライトするだけでなく、一意であるべきKEYに重複がないかのチェックにも有用。一意の値だけ、重複のある値だけ、エラーの出ているセルだけ、などの書式を変えて目立たせることができるからだ。

なお、Excelの「重複の削除」機能は使わないこと。後から出てきたほうを、確認を求めずに削除するので、ダメージが大きすぎる。バグがあるという情報もある。

INDIRECT関数

単に「=A1」だと、A1セル（の中身）の意味。これを、「=INDIRECT(A1)」とすると、A1セルに書いてある先の中身、になる。

ただし、INDIRECTでほかのシートを指定するのはちょっと難しい。ほかのシートを指定するときの通常の記法は「シート名」＋「!」＋「セル」なのだが、INDIRECTを使うときには、「&」やクォーテーションが必要になる。きょうは深入りしないが、誕生日分析などで使用。

（8）Excelの小技２＝作業能率を上げるショートカット

以下は、よく使うショートカットキー。ざっと目を通して、知らないものがあったら、なるべく活用を。マウスを使うよりも早く済む。

Ctrl+ドラッグ
　　シート名のタブをつかみながらだと、そのシートのコピーを作成
Shift＋ドラッグ
　　行や列を選択し、その境目をつかみながらだと、並び替え
Ctrl+1　セルの書式設定。エルではなくて数字の一（テンキーの 1 はダメ）
Ctrl+Z　直前の変更を元に戻す
Ctrl+A　シート全体を選択
Ctrl+C　コピー
Ctrl+V　通常の貼り付け。セル幅以外すべて引き継ぐので、煩わしいこともある。
　　もう一度押すと、貼り付けの形式を選べる
Alt+Ctrl+V　形式を選択して貼り付け。Ctrl+Vでは困るときによく使う。
　　関数を使って整形をした後、貼り直して「値だけ」にするのに便利（Vを選ぶ）
Alt+;　絞り込み時に表示されているセルだけをコピー元にする。重宝する
Ctrl+S　ファイルを上書き保存
「F12」　ファイル名を付け替えて保存
「F2」　セルの編集
Shift+Ctrl+@　押すたびに、セルの表示を「処理の結果」か「数式そのもの」かを切り替える
Ctrl+F　検索
Ctrl+H　置換
Ctrl+矢印　空白セルは飛ばし、その次にデータの入っているセルにジャンプ
Ctrl+Home　A1セル（左上）にジャンプ
Ctrl+End　データの入っている最終セルにジャンプ
Ctrl+;　きょうの日付を入力。便利
Ctrl+:　現在の時刻を入力。便利
Ctrl+*　データが入っている範囲を選択。離れ小島は選択されない
Ctrl+T　テーブルにする
Ctrl+Enter　複数のセルに同じデータを入れる。一括して修正するときに便利
Alt+下矢印　そのカラムに入力済みのデータのリストから選ぶ

（9）Excelの小技３＝ファンクションいろいろ

複雑な処理をするもの

IFERROR　エラーが出ている場合の処理を枝分かれさせる
COUMTIF　条件に合うセルの数を数える。複数の条件をつける場合はCOUNTIFS
COUNTA　空白以外のセルの数を数える。COUNTなら、数値の入っているセルのみ数える
SUMIF　条件に合うものを合計。SUMIFSもある
RANK.EQ　順位を返す。同順位の処理の違いでバリエーションあり
MAX　最大値を返す
MIN　最小値を返す
LARGE　大きい方から数えて指定の順位のものを返す。１位はMAXと同じ
SMALL　LARGEの逆

文字列の整形に使えるもの

ASC　全角文字を半角に変換。反対はJIS
CONCATENATE
　複数の文字列をつなぐ。「&」でつないでいくのと働きは同じ
EXACT　２つの文字列が等しいかどうか判定
FIND　ある文字列が他の文字列の中にあるか検索。SEARCHと似ているが、大文字と小文字を区別。ワイルドカードは使えない
JIS　半角文字を全角に変換。反対はASC
LEFT　文字列の左から、指定された字数を取り出す
LEN　文字列の字数が分かる
LOWER　英字を小文字に変換
MID　文字列の途中の指定の位置から、指定の字数の文字を取り出す
PHONETIC　文字列からふりがなを取り出す
PROPER　英字の語の先頭だけ大文字に変換
REPLACE　指定された位置の文字を他の文字に置き換える
REPT　指定回数だけ繰り返して表示
RIGHT　文字列の右から、指定された字数を取り出す
SEARCH　ある文字列が他の文字列の中にあるか検索。FINDと違い、大文字と小文字は区別せず。ワイルドカードが使える
SUBSTITUTE
　指定された文字を他の文字に置き換える。””（空っぽ）と置換してやれば、文字を削る処理になる
TEXT　数値を書式設定した文字列に変換。日付のシリアル値を曜日に直すときなどに
TRIM　前後の余分な空白を取り除く。途中の空白は１つだけ残す。とてもよく使う
UPPER　英字を大文字に変換
VALUE　数字だけの文字列を数値に変換

（１０）複数の表の付き合わせ＝一意のKEYが必要

年俸データ２種を結びつけるには

本日のメーン。よく分からん、というかたも、作業例を見て、流れを追ってほしい。そういや、こんなことに使えたはずだ、ということだけでもインプットを。いずれ役に立つ。

【練習】絶対参照

掛け算の九九の表を、できるだけ手抜きして作りたい。Excel練習その１のブックの、２枚目のシートを使う。１つのセルにある式を入れると、下と右にコピーして増殖させるだけで済んで、書き換え不要なのだが、さてどうすればいい？

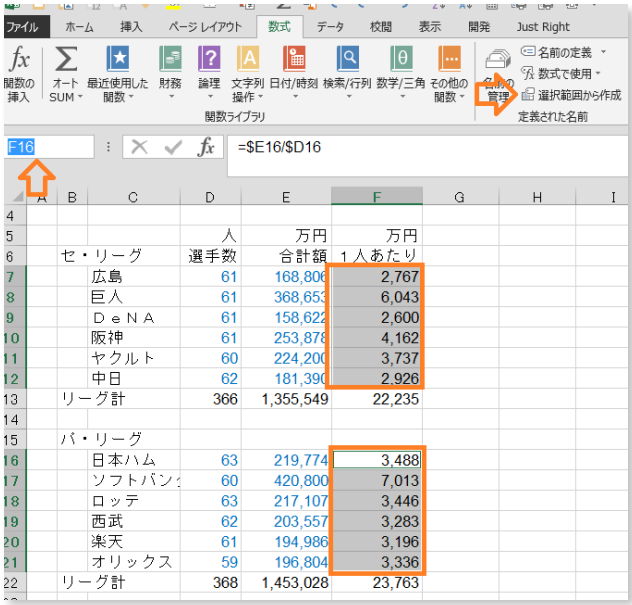
【練習】VLOOKUP

こちらは３枚目のシート。医療費の集計用紙を作るとしたら、こんな感じか。よければご活用を。入力と集計を簡単にするために、医療機関名を入れる番号付きのサブ表（これが元帳になる）を作り、その番号をKEYにして、メインの表に引っ張ってきている。

医療機関ごとの小計をどうやって出しているかもポイント。SUMIFを使っている。この式も絶対参照を使って１回だけ書き、あとはコピーしただけ。

【練習】年俸データ

プロ野球選手の年俸に関するデータの表が２つ。１億円プレーヤーの人数を球団ごとにカウントしたり、スター選手の年俸が球団総額に占める割合を計算したり、をやってみる。球団トータルはワークシートにしてあるが、選手リストはcsv形式のファイルなので、まず、これをExcel練習用２のブックに読み込む。>> データ >> テキストファイル と進み、文字コードや区切り記号の指定を。列ごとに、データの種別（お任せの標準でいい？ 文字列と指定？ 数値と指定？）も考えながら。



	A	B	C	D	E	F	G	H	I
4									
5				人	万円	万円			
6		セ・リーグ		選手数	合計額	１人あたり			
7			広島	61	168,806	2,767			
8			巨人	61	368,653	6,043			
9			ＤｅＮＡ	61	158,622	2,600			
10			阪神	61	253,878	4,162			
11			ヤクルト	60	224,200	3,737			
12			中日	62	181,390	2,926			
13		リーグ計		366	1,355,549	22,235			
14									
15		パ・リーグ							
16			日本ハム	63	219,774	3,488			
17			ソフトバン	60	420,800	7,013			
18			ロッテ	63	217,107	3,446			
19			西武	62	203,557	3,283			
20			楽天	61	194,986	3,196			
21			オリックス	59	196,804	3,336			
22		リーグ計		368	1,453,028	23,763			

「名前をつける」方法。範囲指定してから、画面左上の小窓に名前を入力してやる。または、 >> 名前の管理 >> 選択範囲から作成 で。F7（広島）でクリック→F12（中日）でSHIFT+クリック。続いてF16（日本ハム）でCTRL+クリック→F21（オリックス）でSHIFT+クリック、でうまく範囲指定できる

球団ごとに一意であるべき選手の背番号は、確かに数字は使っているけれど、足したり掛けたりといった計算ができる数値（間隔尺度・比例尺度）ではない。背番号10の選手が20の選手の半分しか活躍しなかったり、年俸が半分だったり、ということはないし、「00」と「0」の両方の選手がいるチームもある。量を表すのではなく、個々の選手の識別につかっているだけで、飛行機の便名などと同じく文字列

扱いすべきものだ。なので、背番号の列は文字列と指定して読み込むこと。

やっていることは、単位の除去や、球団名をKEY（同名の球団はない）にした付き合わせ、比率や合計の計算など。それぞれのステップは、数式を日常語に翻訳できれば、なーんだ、ということばかり。どうか、分からない、と決めこまずに辛抱強くお付き合いを。

厄介なのは、各球団の選手１人当たり年俸の順位。セ・パぞれぞれのリーグ内での順位と同様、RANK.EQを使うのだが、「飛び地」になっているせいで、そのままでは範囲指定ができない（Excelの仕様がそうなってしまっている）。それで「名前をつける」機能を使った。この名前の先頭に洋数字が使えないという制約がある（これまた仕様）ため、「１人あたり〜」ではなく漢数字で「一人あたり」にしている。こういうところで引っかかると、原因が分からずへこむし、時間も食うので、本筋ではないが参考までにトリビアを。

■プロ野球の球団別年俸合計

出所：日本プロ野球選手会(http://jpbpa.net/research/)

セ・リーグ	人 選手数	万円 合計額	万円 １人あたり	リーグ内 順位	12球団中 順位
広島	61	168,806	2,767	5	11
巨人	61	368,653	6,043	1	2
ＤｅＮＡ	61	158,622	2,600	6	12
阪神	61	253,878	4,162	2	3
ヤクルト	60	224,200	3,737	3	4
中日	62	181,390	2,926	4	10
リーグ計	366	1,355,549	22,235		

セル内の小さな棒グラフは「条件付き書式」の「データバー」

セル内の小さな棒グラフは、 >> 条件付き書式 >> データバー で出せる。

（１１）実例の続き＝時間切れだったら各自で

日付を補う

モニタリングの処理の続き。Tabulaで書き出したcsv形式のファイルを編集していく。「モニタリング」フォルダ内のファイルを番号順に見てほしい。

日付は、テキストエディタで「2011/3/xx」のスタイルに直す。ATOMで正規表現を使うなら、Ctrl+Fで検索モードに。続いて、画面右下の「.*」ボタンを押す（逆向きスラッシュ「\」と半角の「￥」は、とりあえず同じものと考えてよい）。上の窓に検索対象のパターン、下の窓に置き換え後のパターンをいれる。

ATOMで正規表現を使い、日付を修正する

「`^(\d{1,2})月(\d{1,2})日`」を「`2011/$1/$2`」に

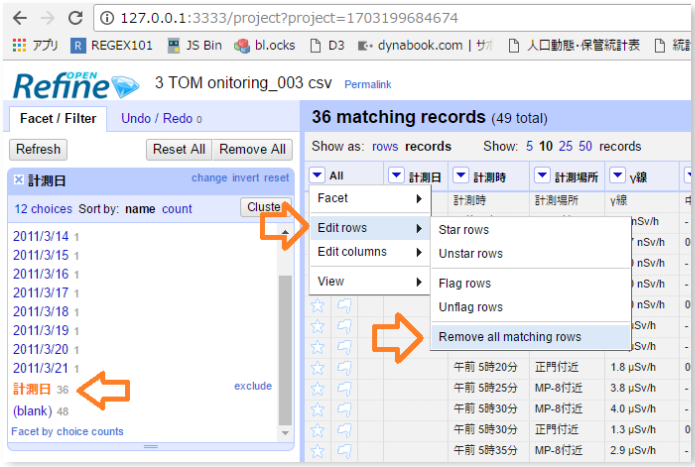
置換すればよい。行頭に「数字１字か２字」「月」「数字１字か２字」「日」というパターンに当てはまるものがあったら、「2011/最初に捕まえたほうの数字/次に捕まえたほうの数字」

に置き換えなさい、という指示になる。よほど自信がない限り、一括置換の前に一つ二つは試してみること。

不要な行や単位の削除、時刻の変換

次にOpenRefineで開いて、ページの変わり目の不要な行（項目名の再掲）を削除する。日付が「計測日」になっている行がそうなので、削除。日付はフィルダウン（入っていないセルにも補う）してやる。OpenRefineの操作については、すぐ後の「付録」にまとめた。

単位は、削るだけなら、ATOMでもOpenRefineでもExcelでも、どれでもOK。ただ、放射線量の単位がnSv/hからμSv/hに変わっている、そのぐらいとんでもないことが起きた、ということこそがニュースなので、単に削るのではなく、ついていた単位がどちらだったかで列を分けて処理し、あとでμSv/hのほうを1000倍して換算するのがいい（μは10の-6乗、nは-9乗）。となると、縦の列（カラム）の処理に向くOpenRefineかExcel。



各ページ冒頭の不要な行（項目名再掲）をOPENREFINEで一括削除する

日付のフィルダウンは、Excelは不便。OperRefineやCalcのほうがやりやすい。ページの変わり目の項目名再掲の行を一括して削るのは、フィルターを使えばExcelでも可能。

日付の「yyyy/mm/dd」スタイルや、時刻の24時間制の「hh:mm」への変換は、Excelで関数を使うのがスマート。ワークシートを見てほしい。本日のオススメはこちら。

でも、繰り返しを厭わずに一つずつ置換するなら、エディタは分かりやすい。ただし、指定の仕方や作業順はよく考えて。「午後11時」を「23時」に、というふうに「時」まで指定する。「時」をつけないまま、しかも昇順でやっていくと……。「午後1」を「13」に、から始めたとしたら、午後11時は「131時」になってしまう。

測定場所の表記の揺れの修正は、OpenRefineがやりやすいが、Excelのフィルター機能で揺れを探し、「Ctrl+改行」による一括入力で修正することもできる。

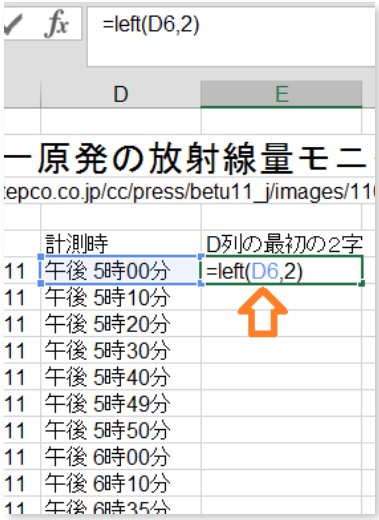
表計算ソフトに読み込む

csvファイルをいよいよExcelに読みこむときは、 >> データ >> テキストファイル で対象を選び、 >> インポート。文字コード（UTF-8やシフトJIS）、先頭行は項目名かデータか、区切りに使われている文字（csvなので、タブではなく、コンマ）、の指定を忘れずに。必要に応じて、日付と時間の列は「文字列」指定で読み込む。でないと、年を補っていないときは、勝手に2017年の日時と解釈されてしまう。

IFなどを使って24時間制に直す

12時間制の時刻データをExcelで整形し、最後に日付と連結して、やっと扱いやすい形の日時データになる。「時刻を修正」のシートでやっていることは、

D列　12時間制の時刻が文字列で収容されている
E列　D列のセルの先頭の２字、つまり「午前」か「午後」かをLEFT関数で取り出す
F列　「時」は何文字目にあるのかを、FIND関数で調べる。SEARCHを使ってもよい
G列　MID関数を使って、××時の数字部分を取り出し、TRIMで不要な空白を除去。
　　それをVALUEで数値に変換する
H列　E列が「午後」の場合のみ、G列の値に12を加える。IFを使っている
I列　F列の結果を再利用。「時」の後の２文字は、××分を表す数字なので、それを取り出す
J列　TIME関数は時・分・秒を必要とするので、「0秒」を添えて渡している。
　　最後に日付と連結



12時間制表記の時刻を、関数を使って順次整形していく

細かく段階分けしたのと、PDFに見えない空白が入っている可能性を考慮してTRIMを多用したのとで、くどくなっている。慣れてきたら、ここまで分けなくてもOK。

このシートをコピーして作った「時刻を値だけに」のシートでは、J列をコピペする際に形式「値」を選んでいる。関数を駆使して処理し終えたら、このように数式の参照元と関係を切って、値だけにしておくとよい。

γ線の単位を削る

放射線量の単位をExcelで削るなら、こんな式で。「線量を数値に」のシートを参照のこと。単位つきの元データが仮にE列にあるなら、EやF列に

`「=VALUE(TRIM(SUBSTITUTE(E××,"nSv/h","")))」`　→これをF列に
`「=VALUE(TRIM(SUBSTITUTE(E××,"μSv/h","")))」`　→これをG列に

と入れる。××は、この式を入れる先頭行の数字に差し替えて、その式を最終行までコピーすること。これで、単位が外れて、計算や並び変え可能な数値になる。削ろうとした単位と違っていれば、VALUEで数値化しようとした際にエラーするので、安全。IFを使って丁寧にやっても、もちろんいい。

μSv/hをnSv/hに換算する際に、H3セルの値「1000」を掛け算している。ここは数式を下にコピーしていくときにも、ずれてほしくない箇所なので、絶対参照にしている。

マイクロとナノの関係は不変だから、ここまでする必要はないが、消費税率など、あちこちで使う数字で、しかも将来、変わる可能性があるものは、セルの数式に直打ちするのではなく、

どこかのセルに入れておいて参照するのが、メンテナンスしやすい。参考までに。
　単位を削る方法はいろいろ考えられて、

「=IF(RIGHT(E××,5)=“nSV/h”, VALUE(TRIM(LEFT(E××,LEN(E××)-5))」
　→前半のIFで単位を判別していて、これはF列用。後半はG列も共用

とか

「=VALUE(TRIM(LEFT(E××,SEARCH(“nSV/h”,E××)-1)))」
　→削る字数を決め打ちしていないので、単位の後に「未満」などがあっても動く

なども。ただし、単位以外のものもまとめて削れるからといって、それが妥当か、そうすべきかどうかは、また別の話。
　「テーブルにした」のシートでは、セル内に棒グラフを表示するのに、条件付き書式のデータバーを使っている。

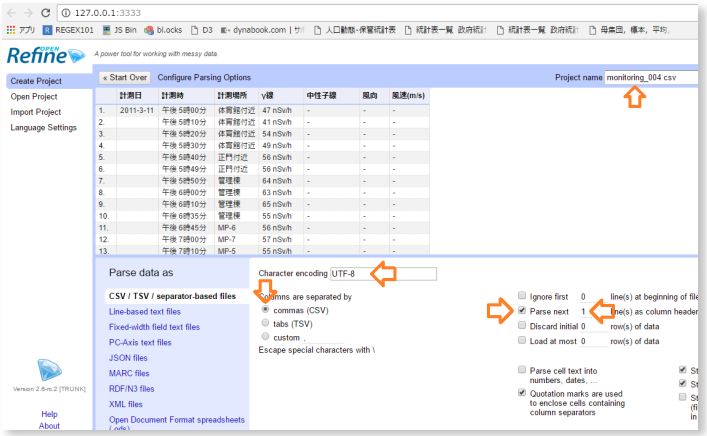
そのほか

　中性子線、数値がちっとも動いてないし、数字が入っているところは全部「未満」ってついているでしょ。と思わなかっただろうか。Excelに数えさせてみたらいい。実は、そうでない行が22ある。「中性子線」のシートのJ列で、ANDを使って「0.01未満」でも欠測でもないもの（この２つの条件を同時に満たすもの）を洗い出している。K列とL列は、それを数えたり、目立たせたりしている。フィルター機能を使ってもよいが、こんなふうに関数や条件付き書式を使って、ハイライトすることも可能だ。とにかく、数値の後についているものを乱暴に削ってしまう（L列）、というやり方は賢明ではない。
　ではどうする。それはソフト任せでなく、記者が考えるべきこと。「未満」ってどういうことなのか、を取材することから始めないといけないだろう。

【付録】OpenRefineの操作

起動してファイルを読み込む

　起動にしばらく時間がかかる。ブラウザが立ち上がると、「http://127.0.0.1:3333/」という、自分のＰＣ内のローカルサーバーに接続しているはず。ここで、>> Create Project >>This Computer >> ファイル選択と進む。ファイルを指定したら、>> Next。
　>> Character　encodigを、たとえば



OPENREFINEでファイルを読み込む

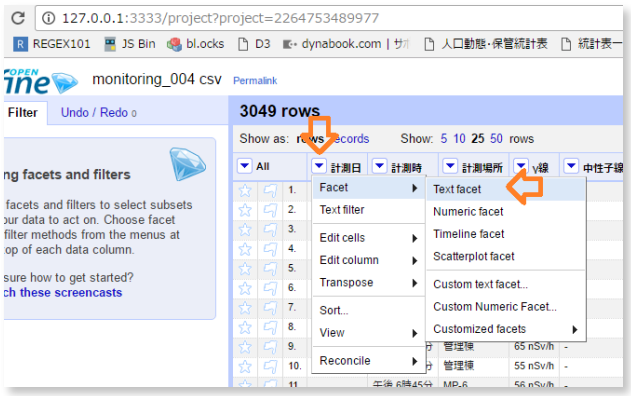
UTF-8やSHIFT_JISに指定。文字化けしていないかプレビューで確かめる。区切り記号は（csvなので）コンマ。１行目は項目名かデータの数値かを指定する。>> Project nameで名前を適宜つけ、>> Create Projectを押す。

行数表示とモード

検索条件をつけて絞り込むと、操作対象の行数表示が減る。全部の行を捕まえるつもりだったのに、ここの数字が減ったなら、何か漏れがあるということ。いつも注意して見ておくとよい。rowモードは行数をカウントするが、recordモードは先頭列で見て、大きな塊で区切ってカウントする。通常はrowで困らない。

日付の処理（まだだったら）

計測日のカラムで、>> Facet >> Text facetと進む。「3月11日」なら、>> editで窓を開き、「2011/3/11」に打ち替えて、>> Apply。以下同様。数が少ないので、これが現実的。



FACETを使って絞り込む

または、>> Edit column >> Add column based on this columnと進み、列名を「Date」などとして、>> Expression（式を書く）に「value.match(/(\d+/)月(\d+)日/)[0]」とすれば月の、最後を[1]とすれば、日の数字が取り出せるので、年やスラッシュを補ってつなぐ手もある。

```
「"2011/" + value.match(/(\d+)月(\d+)日/)[0] + "/"+value.match(/(\d+)月(\d+)日/)[1]」
```

となる。

```
「if(substring(value,0,1)=="3", "2011/3/" + substring(chomp(value,"日"),-2), value)」
```

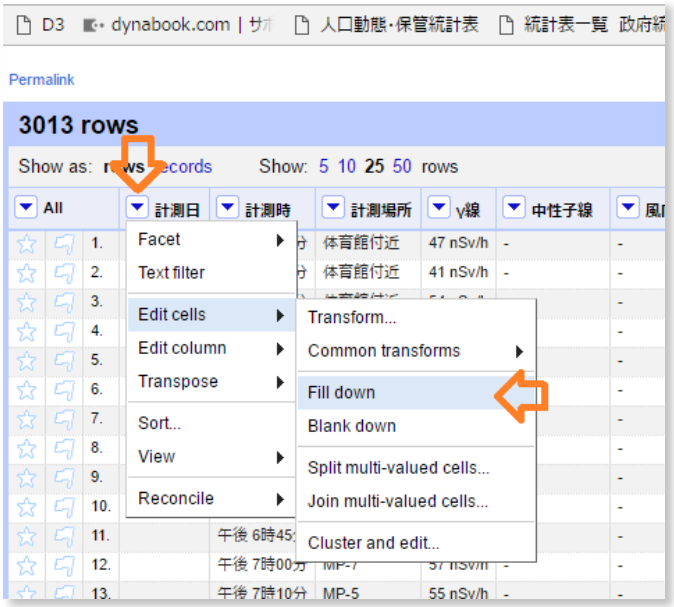
と入れて、置き換えることもできる。3で始まっていたら、末尾の「日」は削って日にちの数字だけにしたものを、「2011/3/」のあとにつなぐ、という指示だが、いかにも力技。別の月だったり、日付が１桁だったら動かない。その場しのぎにはなるが、スマートではない。

行の削除

計測日のカラムが「計測日」になっている行を選択。これが、ページの変わり目の不要な行。左端の >> All >> Edit rows >> Remove all matching rowsと選んで、一括削除。

フィルダウン

最初の行にだけデータがあり、あとは次に違うデータが出てくるまで省略している、というスタイルのときに使う。計測日のカラムで Edit cells >> Fill down。これを使って、全データに日付を付してやる。逆はEdit cells >> Blank down。



日付をフィルダウン。3000行が瞬時に処理できる

計測場所の表記の揺れ

「計測場所」のカラムで >> Facet >> Text facet と進めば、一目瞭然。今は、気になるところは１つ（ハイフンの抜け）しか残っていないようだ。

数値データの付随文字を削る

計算した形跡のない「数字」（単位つき、とはそういうこと）は、数字に見えても英文字のアイやエルやオーが紛れ込んでいる可能性が十分にある。toNumber()を使って数値に変換した上で、Numeric facetでエラー（数値に変換されずに残ったもの）を監視しつつ作業する。数値かそうでないかで、色が違うし、ヒストグラム表示もあるので、問題が生じたときに気付きやすい。

→γ線を例にとると、 >> Edit column >> Add column based on this columnと進み、列名を「γ線（nSv/h）」などとして

```
「value.trim().match(/([.\d]+)\s*(nSv\/h).*/)[0].toNumber()」
```

で数値に。同様に

>> Edit column >> Add column based on this columnから、列名を「γ線（元の単位はμSv/h）」などとして

```
「value.trim().match(/([.\d]+)\s*(μSv\/h).*/)[0].toNumber()」
```

のように、もともとnSv/hだったものと、μSv/hから換算したものとで、別々の列に書き出すと、混同しなくてよい。最終的には、μSv/hのほうは表計算ソフトに渡した後で1000倍してnSv/hに換算し、同じ列にまとめるにしても（グラフ化などするには同じ列が便利）。

なお、「Numeric facet」で経過を確かめながら進めるためにtoNumber()を使ってみたが、μSv/hのほうを「1000倍する」計算までをOpenRefineでやってみたら、誤差が出た。整数を1000倍したのに整数にならなかった。データの整形に徹し、最終的な計算は表計算ソフトに任せる、と使い分けるほうが安全だ。

単位の違いを気にせずに削れるケースなら

>> Edit column > Add column based on this columnで隣に列を増やす。New column nameの小窓に列名を適宜つけるまでは同じ。で、式を書くExpressionの欄をどうするか。
→後ろを削るchomp()は分かりやすいが、前にあるものは削れない。モニタリングでなく地震の震源データだったら、Mが数字の前に来て、その後に半角空白も入っていたりする。
→どうせなら、使い回しがきくように考えてみる。Expressionの窓に

```
「value.match(/[^-.\d]*([-.\d]+)[^-.\d]*/)[0].toNumber()」
```

または

```
「toNumber(value.match(/[^-.\d]*([-.\d]+)[^-.\d]*/)[0])」
```

と入力する。書き方の流儀が違うだけで、意味は同じ。

捕まえたいものは、数字と小数点とマイナス符号。Mや気温など、マイナスもありうるので。それ以外のもの（空白も）が前後にあったら拾わずに捨ててしまう。最後に数値化しておくと点検しやすい、という発想。なお、この式のmatchの中に３つ出てくる「.」は「小数点」のことで、ここでは「どんな文字でも」の意味ではない。この式には汎用性があり、数値だけ切り出すのに使える。そのかわり、付いていた元の単位が何だったか（あるいは、前後についているのは、そもそも単位だったのか）に気を配ることなく、切り離して捨てている。

書式の注意

OpenRefineでは、Expressionの窓のmatch()で正規表現を使う場合だけ、match(//)のように「//」で挟む必要がある。また、１行におさめる独特の記法になっている。ATOMやregex101と違い、「何かを捕まえて」「何かと置き換える」と２行で書く分かりやすいスタイルではない。２種類ある記法のうち１種は、Excelの関数の書き方に近い。

作業後のデータの保存

右上に並んだボタンのうち、>> Exportをクリック。メニューが開くので、上から３番目の >> Comma-separated value（コンマ区切り。この略称がcsv）を選ぶと、csv形式で保存される。その他の形式で保存することもできる。

終了

ターミナル画面で「Ctrl+C」を押す。
以上です。お疲れさまでした。

表計算ソフトの発明者が語る動画(https://goo.gl/FCcfyE)を見て息抜きを。こんな便利なものを一生懸命考えてくれたのなら、活用してあげなきゃ、という気がしてきます、きっと。