Universiteit Utrecht

Regular Expressions
a very quick introduction

Part of: R café 11 december 2017
Jonathan de Bruin Data scientist, ITS Utrecht
Tessa Pronk Data- and Informationspecialist, UB Utrecht

RDM Support

Resources for learning about regular expressions:

https://cran.r-project.org/web/packages/stringr/vignettes/regular-expressions.html
http://r4ds.had.co.nz/strings.html#matching-patterns-with-regular-expressions
https://regexcrossword.com/

This short introduction was modified from a Python regular expression workhop given by Harrison Dekker (University of California Berkely) at Iassist 2016.

What are regular expressions?

Regular expressions are *specific sequences of characters* that broadly or narrowly match patterns

Why regular expressions?

How would you extract the product codes for 'company X' products (a capital followed by three numbers) in these examples?

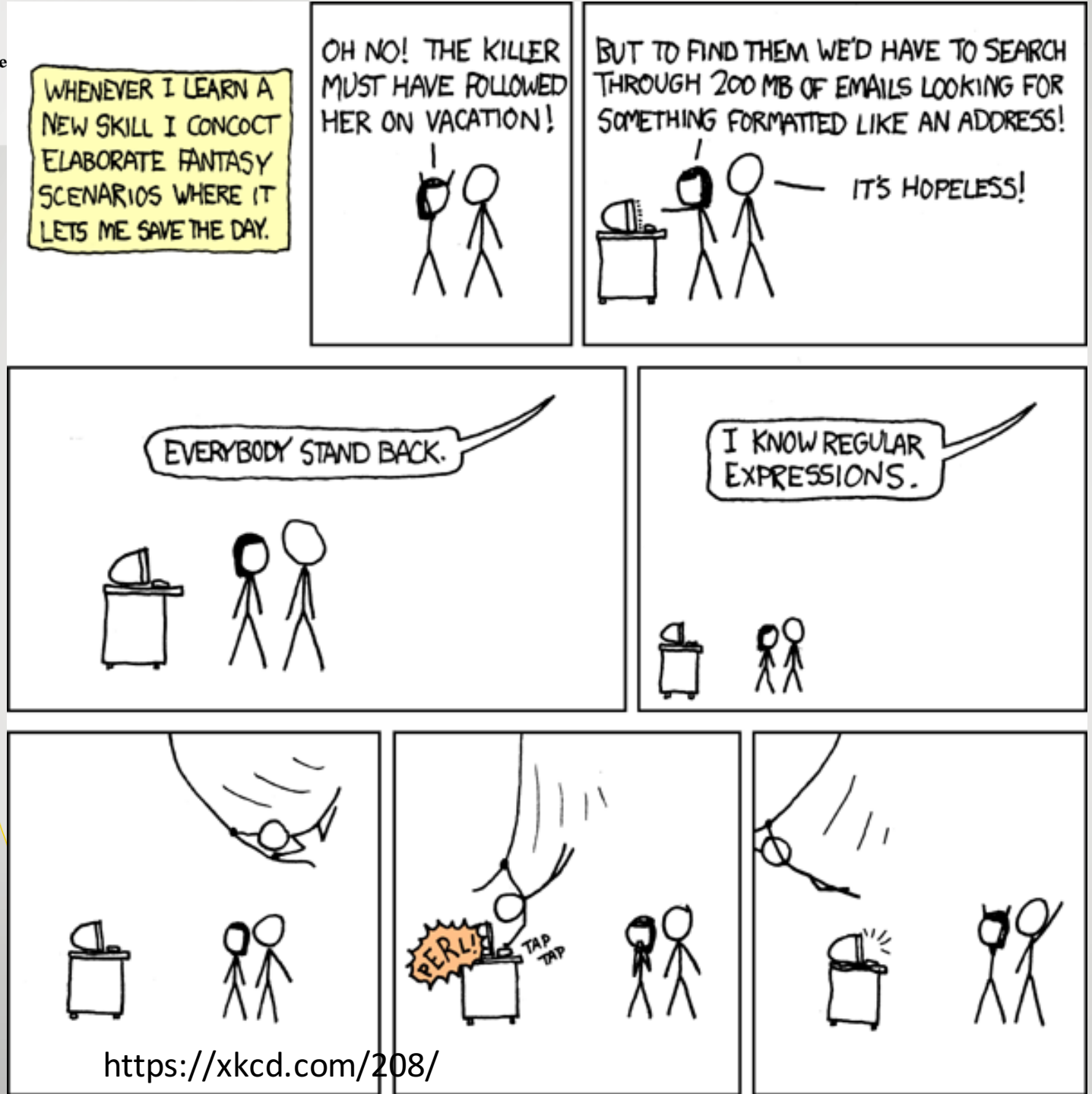
"[A-Z]\\d{3}"

Example 1

PCOD	QTY	DEPT	COST
A169	100	Micro	0.58
PDA1	1	Xray	600.00
X280	5	ER	199.99

Example 2

```
'...The X701 vacuum cleaner really sucked!...'
'...The gloves(P180) felt sticky...'
```



https://xkcd.com/208/

Universiteit Utrecht

```
FA  - Koteshwar, Prakashini
FA  - Singh, Jyoti
IN  - Puppala,Radha. Department of Radiodiagnosis, Kasturba Medical College, Manipal, Karnatak
      Sripathi,Smiti. Department of Radiodiagnosis, Kasturba Medical College, Manipal, Karnata
      Kadavigere,Rajagopal. Department of Radiodiagnosis, Kasturba Medical College, Manipal, K
      Koteshwar,Prakashini. Department of Radiodiagnosis, Kasturba Medical College, Manipal, K
      Singh,Jyoti. Malathi Manipal Hospital, Bangalore, Karnataka, India.
TI  - Abdominal cocoon secondary to disseminated tuberculosis.
SO  - BMJ Case Reports. 2014, 2014.
AS  - BMJ Case Rep. 2014, 2014.
NJ  - BMJ case reports
PI  - Journal available in: Electronic
PI  - Citation processed from: Internet
JC  - 101526291
SB  - Index Medicus
CP  - England
MH  - *Abdomen/pa [Pathology]
MH  - Abdominal Pain/di [Diagnosis]
MH  - Humans
MH  - Intestinal Diseases/di [Diagnosis]
MH  - *Intestinal Diseases/et [Etiology]
MH  - *Intestine, Small/pa [Pathology]
MH  - Male
MH  - Middle Aged
MH  - *Peritonitis, Tuberculous/pa [Pathology]
MH  - Tomography, X-Ray Computed
MH  - *Tuberculosis, Miliary/pa [Pathology]
AB  - Abdominal cocoon, also known as sclerosing encapsulating peritonitis, represents a rare
ES  - 1757-790X
DI  - bcr2013202568
DO  - http://dx.doi.org/10.1136/bcr-2013-202568
PT  - Case Reports
PT  - Journal Article
```

My case study: extract Title and Abstract from many, many, Medline records.

What you need to know to write regular expressions:

How to define sets of characters

- Metacharacters: i.e. all digits, all characters, all tabs
- Character sets: i.e. only these digits, only these characters

How many times to repeat them

- A specific number of times
- An unlimited number of times

How to define their position

- End of line
- Beginning of line
- Word boundary

Sets of characters

Metacharacters

Metacharacters are pre-defined sets of characters.

. matches ANY character except the newline character \n
\d matches digits 0 through 9
\w matches alphanumeric characters and underscore
\s matches any whitespace
\t Tabs
\n Newline

ETC.

TYPE ALONG in R:

```
Txt <- "SA_1234"
regmatches ( Txt , regexpr ("\\w", Txt ))
regmatches ( Txt , gregexpr ("\\w", Txt ))
regmatches ( Txt , gregexpr ("\\d" , Txt ))
```

R treats backslashes as escape values for character constants. (... and so do regular expressions. Hence the need for two backslashes when supplying a character argument for a pattern. The first one isn't actually a character, but rather it makes the second one into a character.)

Sets of characters

Character sets

[AGCT] matches one character A, G, C or T.
[\s\d] matches one whitespace character or digit

Define a range of characters

[A-T] matches one character between A and T.
[1-7] matches one digit between 1 and 7.

Ranges as defined by ASCII or Unicode tables
You can combine ranges: [a-cA-C]

TYPE ALONG in R:

```
regmatches(d , gregexpr("[A-Z]", Txt))
```

How to define where they are

Position of the pattern

We can say a regex has to be at the start or the end of the string, or at word boundaries, with more special characters.

^ beginning of line
$ endofline
\b word boundary

Examples:
^Hallo$
\bHallo\b

TYPE ALONG in R:

```
regmatches(d , gregexpr("^[A-Z]", Txt))
```

Repetitions

How to define repetitions

Three digits: \d\d\d. How about matching thirty or a thousand digits?

Fortunately, regular expressions let us express this very succinctly.

* means 0 or more times
\+ means 1 or more times
? means 0 or 1 times
{n} means n times exactly
{m,} means m or more times
{m,n} means m–n times

Example: .* matches anything
Example: [a-c]{3} matches 'abc' etc

TYPE ALONG in R:

```
regmatches(d , gregexpr("[A-Z]{2}", Txt))
regmatches(d , gregexpr("[0-9]{3}", Txt))
```

Specifying alternatives

Sometimes you want to say match this OR that. You can do that with the | operator.

Alex|Bill|Conrad matches any of these three names.

Sometimes there can be confusion about what | refers to. In such cases, put brackets around the alternatives.

Jim and (Alex|Bill|Conrad) matches 'Jim and Alex', 'Jim and Bill', etc

Defining complements of a set

Sometimes it's easier to define a set of characters as "everything other than X".

\S – all non-whitespace characters

\W – all non-alphanumeric characters (also excludes underscore)

\D – all non-numeric characters

[^A-D] – all characters other than A, B, C, D

Doing stuff with regular expressions

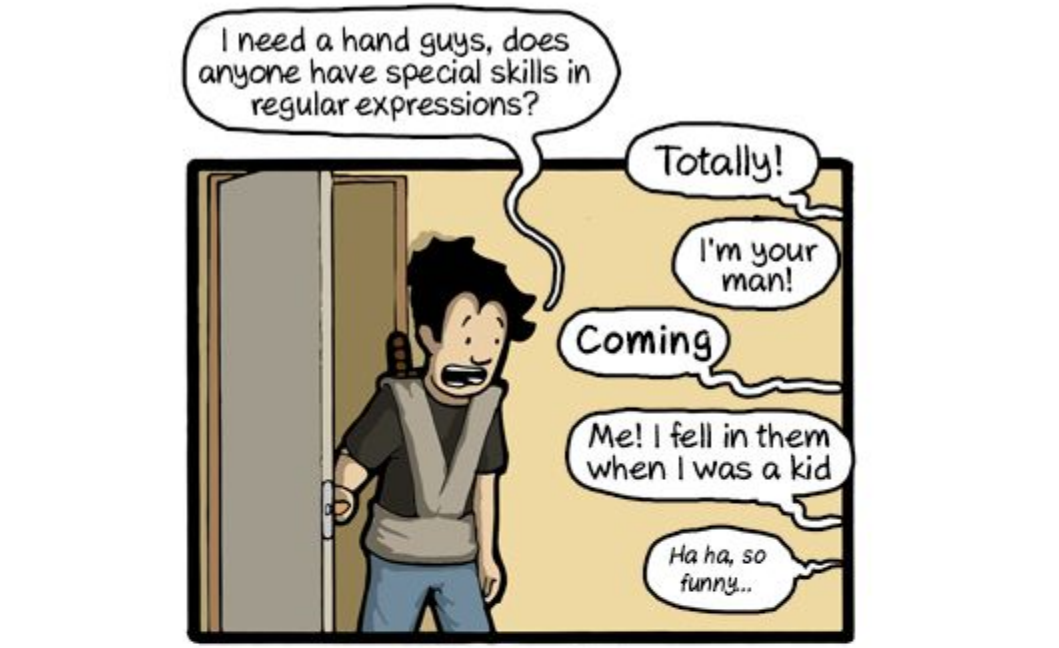
For instance:

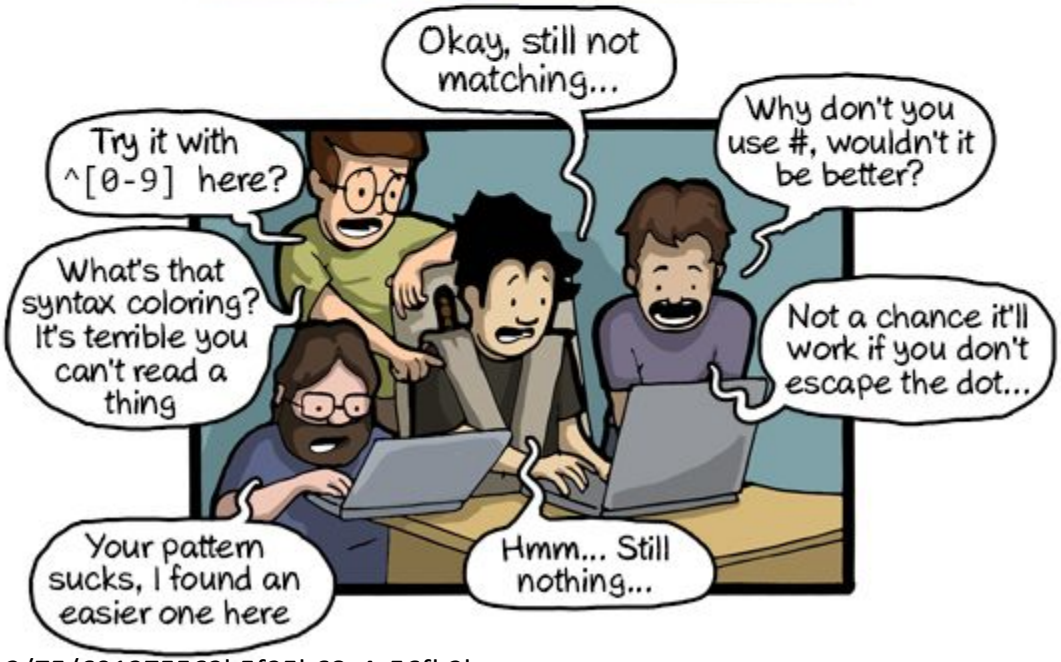
Capture the matches and do stuff with them

Replace the match with something else

Split a string whenever you match the regex

Universiteit Utrecht





https://i.pinimg.com/736x/69/12/75/691275562b5f25b68c4c56fb2bce92d9--regular-expression-programming.jpg