*Statistical Computing (36-350)*

# Regular Expressions I

Cosma Shalizi and Vincent Vu

*November 9, 2011*

# Agenda

- Pattern matching in strings – why?
- Basics of regular expressions
  - literals, classes, modifiers
- Search/Replace with grep() and gsub()
- Live examples
- Recommended reading: "?regexp" in R

# Why?

- Many text processing problems involving automatically matching **different**, but **regularly structured** parts of the text
- Want an expressive method for describing these **patterns of characters**

# Examples

# Extract the MPG related columns

```
> cars <- read.csv('http://www.stat.cmu.edu/~cshalizi/statcomp/labs/07/epafuel2010.csv')
> names(cars)
 [1] "year"                                "manufacturer"
 [3] "division"                            "country"
 [5] "carline"                             "manufacturer.code"
 [7] "model.type"                          "displ"
 [9] "cylinders"                           "mpg.city"
[11] "mpg.hwy"                             "mpg.combined"
[13] "mpg.city.unadjusted"                 "mpg.hwy.unadjusted"
[15] "mpg.combined.unadjusted"             "aspiration"
[17] "transmission"                        "gears"
[19] "drive"                               "Fuel.Usage.Desc...Conventional.Fuel"
[21] "X2Dr.Pass.Vol"                       "X2Dr.Lugg.Vol"
[23] "X4Dr.Pass.Vol"                       "X4Dr.Lugg.Vol"
[25] "Htchbk.Pass.Vol"                     "Htchbk.Lugg.Vol"
[27] "epa.annual.fuel.cost"                "vehicle.class"
[29] "vehicle.type"                        "batteries"
```

# Amateur approach: *scroll and count*

```
> names(cars)
 [1] "year"                                "manufacturer"
 [3] "division"                            "country"
 [5] "carline"                             "manufacturer.code"
 [7] "model.type"                          "displ"
 [9] "cylinders"                           "mpg.city"
[11] "mpg.hwy"                             "mpg.combined"
[13] "mpg.city.unadjusted"                 "mpg.hwy.unadjusted"
[15] "mpg.combined.unadjusted"             "aspiration"
[17] "transmission"                        "gears"
[19] "drive"                               "Fuel.Usage.Desc...Conventional.Fuel"
[21] "X2Dr.Pass.Vol"                       "X2Dr.Lugg.Vol"
[23] "X4Dr.Pass.Vol"                       "X4Dr.Lugg.Vol"
[25] "Htchbk.Pass.Vol"                     "Htchbk.Lugg.Vol"
[27] "epa.annual.fuel.cost"                "vehicle.class"
[29] "vehicle.type"                        "batteries"
```

# Pro approach: *grep* and *regular expressions*

```
> grep('^mpg', names(cars))
[1] 10 11 12 13 14 15
```

# Find the e-mail addresses

## B

Anthony Brockwell
Adjunct Associate Professor

---

## C

Jodi Casabianca
Postdoctoral Fellow
Office: Wean Hall 8119
Phone: (412) 268-6328
Email: jodicasa @ andrew.cmu.edu

---

## D

Bernie Devlin
Adjunct Associate Professor
Office: University of Pittsburgh
Phone: (412) 624-1432

---

George T. Duncan
Emeritus Professor
Office: Hamburg Hall 2102D
Phone: (412) 268-2172
Email: gd17 @ andrew.cmu.edu
Home Page

---

## E

# Where are the e-mail addresses?

```
<ul class="peoplelisting">
<li><a name="B"><br />
<h3>B</h3>
<p></p></a>
<ul>
<li>Anthony Brockwell<br />
                  Adjunct Associate Professor</li>
</ul>
</li>
<li><a name="C"><br />
<h3>C</h3>
<p></p></a>
<ul>
<li>Jodi Casabianca<br />
                  Postdoctoral Fellow<br />
                  Office: Wean Hall 8119<br />
                  Phone: (412) 268-6328<br />
                  Email:   jodicasa @ andrew.cmu.edu</li>
</ul>
</li>
<li><a name="D"><br />
<h3>D</h3>
<p></p></a>
<ul>
<li>Bernie Devlin<br />
                 Adjunct Associate Professor<br />
                 Office: University of Pittsburgh<br />
                 Phone: (412) 624-1432</li>
<li>George T. Duncan<br />
                  Emeritus Professor<br />
                  Office: Hamburg Hall 2102D<br />
                  Phone: (412) 268-2172<br />
                  Email: gd17 @ andrew.cmu.edu<br />
                  <a href="http://duncan.heinz.cmu.edu/GeorgeWeb/">Home Page</a>
</li>
</ul>
</li>
```

# Where are the e-mail addresses?

```
<ul class="peoplelisting">
<li><a name="B"><br />
<h3>B</h3>
<p></p></a>
<ul>
<li>Anthony Brockwell<br />
                  Adjunct Associate Professor</li>
</ul>
</li>
<li><a name="C"><br />
<h3>C</h3>
<p></p></a>
<ul>
<li>Jodi Casabianca<br />
                  Postdoctoral Fellow<br />
                  Office: Wean Hall 8119<br />
                  Phone: (412) 268-3828<br />
                  Email: jodicasa @ andrew.cmu.edu</li>
</ul>
</li>
<li><a name="D"><br />
<h3>D</h3>
<p></p></a>
<ul>
<li>Bernie Devlin<br />
                 Adjunct Associate Professor<br />
                 Office: University of Pittsburgh<br />
                 Phone: (412) 624-1432</li>
<li>George T. Duncan<br />
                  Emeritus Professor<br />
                  Office: Hamburg Hall 2102D<br />
                  Phone: (412) 268-2172<br />
                  Email: gd17 @ andrew.cmu.edu<br />
                  <a href="http://dunc.heinz.cmu.edu/GeorgeWeb/">Home Page</a>
</li>
</ul>
</li>
```

# What's an e-mail address?

vqv@cmu.edu

cshalizi@cmu.edu

[alphanumeric]@[alphanumeric].[alpha]

# What's an e-mail address?

vqv@stat.cmu.edu

cshalizi@stat.cmu.edu

[alphanumeric]@[alphanumeric *or* .].[alpha]

# Find the links

# Find the links

```
<p class="summary">
Joe Paterno’s time as football coach will soon be over in the wake of a sex-abuse scandal, people briefed on the matter said.    </p>
<ul class="refer">
<li><a href="http://thequad.blogs.nytimes.com/2011/11/08/paterno-speaks-to-students-from-his-window/"><img src="http://graphics8.nytimes.com/images/
multimedia/icons/video_icon.gif" alt="Video" width="12" height="9" border="0"> Paterno Speaks to Students</a></li>
<li style="background-image: none; padding-left: 0pt;"><img src="http://i1.nyt.com/images/section/opinion/comment_icon.gif" alt="comment icon"
style="margin-right: 3px; padding-top: 3px;" border="0"><span class="commentCountLink" articleid="100000001159580" overflowurl="http://
community.nytimes.com/article/comments/2011/11/09/sports/ncaafootball/penn-state-said-to-be-planning-paternos-exit.html"></span></li>
</ul>
</div>
</div>
<div class="columnGroup last">
<div class="story">
<h3><a href="http://www.nytimes.com/2011/11/09/health/research/surgery-to-prevent-strokes-is-found-ineffective.html?hp">
Study Debunks Operation to Prevent Strokes</a></h3>
<h6 class="byline">
By DENISE GRADY        <span class="timestamp">9:22 PM ET</span>
</h6>
<p class="summary">
Doctors had hoped the operation would prevent strokes in people with poor circulation to the brain.    </p>
</div>
</div>
</div><!--close aColumn -->
<div class="bColumn opening">
<div id="photoSpotRegion">
<div class="columnGroup first">
<div class="story">
<div class="ledePhoto" id="ledePhoto">
<div class="image">
<a href="http://www.nytimes.com/2011/11/09/us/politics/ohio-turns-back-a-law-limiting-unions-rights.html?hp"><img src="http://i1.nyt.com/images/
2011/11/09/us/09ohio2_span/09ohio2_span-hpMedium.jpg" width="337" height="250" alt="Opponents of a bill that restricted public workers’ rights to
bargain collectively cheered at a rally in Cleveland after they heard election results on Tuesday." /></a>
</div>
<h6 class="credit">Amy Sancetta/Associated Press</h6>
</div>
<h3><a href="http://www.nytimes.com/2011/11/09/us/politics/ohio-turns-back-a-law-limiting-unions-rights.html?hp">
Ohio Repeals Law Limiting Union Rights</a></h3>
<h6 class="byline">
By SABRINA TAVERNISE        <span class="timestamp">17 minutes ago</span>
</h6>
<p class="summary">
Voters in Ohio delivered their verdict on a centerpiece of the conservative legislative agenda.        </p>
</div>
</div>
<div class="columnGroup last">
<div class="story">
<h5><a href="http://www.nytimes.com/2011/11/09/us/politics/votes-across-the-nation-could-serve-as-a-political-barometer.html?hp">
Hard-Fought Contests Watched for 2012 Clues</a></h5>
```

# What's a link?

```
<a href="http://www.nytimes.com/2011/11/09/
us/politics/ohio-turns-back-a-law-limiting-
unions-rights.html?hp">
```

```
<a href = "http://www.stat.cmu.edu/" >
```

The part in orange is **anchor text** – it surrounds what we want

# Search/Replace

- Search strings for a pattern
- Replace substrings that match a pattern
- Very flexible if when regular expressions are used for patterns

# grep()

```
y <- grep(pattern, x, ignore.case = FALSE,
          value = FALSE, fixed = FALSE)
```

- `pattern` pattern to match
- `x` string to search within
- `ignore.case` ignore case when matching?
- `value` return matching strings or indices?
- `fixed` literal pattern? (instead of regexp)

# Other search functions

- grepl() – returns logical vector of matching elements (does x contain pattern?)
- regexpr() – returns positions of **first** match
- gregexpr() –returns positions of **all** matches

# grepl()

```
y <- grepl(pattern, x, ignore.case = FALSE)
```

- `pattern` pattern to match
- `x` string to search within
- `ignore.case` ignore case when matching?
- Returns a logical vector indicating elements of x that matched

# grep()

```
> fruits <- c(
"apples and oranges and pears and bananas",
"pineapples and mangos and guavas"
)
> grep("mango", fruits)
[1] 2
> grepl("mango", fruits)
[1] FALSE  TRUE
```

# gregexpr()

```
> fruits <- c(
"apples and oranges and pears and bananas",
"pineapples and mangos and guavas"
)
> gregexpr("mango", fruits)
[[1]]
[1] -1
attr(,"match.length")
[1] -1

[[2]]
[1] 16
attr(,"match.length")
[1] 5
```

# gsub()

```
y <- gsub(pattern, replacement, x,
          ignore.case = FALSE)
```

- `pattern` pattern to match
- `replacement` replacement string
- `x` string to search within
- `ignore.case` ignore case when matching?
- Returns a string with occurrences of pattern replaced by `replacement`

# gsub()

```
> fruits <- c(
"apples and oranges and pears and bananas",
"pineapples and mangos and guavas"
)

> gsub("and", ",", fruits)
[1] "apples , oranges , pears , bananas"
[2] "pineapples , mangos , guavas"

> gsub("apples", "nuts", fruits)
[1] "nuts and oranges and pears and bananas"
[2] "pinenuts and mangos and guavas"
```

# gsub()

```
> gsub(pattern = 'aa', replacement = 'b',
       x = 'aaacc')
[1] "bacc"
```

# Regular Expressions

- Method of expressing patterns in strings
- Formally
  - Describe a set of strings
  - Matching is checking whether a given (sub)string belongs to the set
- Are strings themselves!

# Syntax

- Regular expression syntax varies somewhat between languages
  - R (>2.10.0) tries to be flexible
    - supports Perl syntax
    - supports POSIX 1003.2 standard (we will use this one)

# Components

- Literal characters – match a single character
- Character classes – match any character in class
- Modifiers – operate on the above

# Literals

| regular expression | example matches* |
|---|---|
| “a” | “apple”, “car”, “orchestra” |
| “ee” | “eel”, “between”, “feed” |
| “cat” | “cat”, “catch”, “scatter” |
| “19” | “19”, “1984”, “4519” |

** more precisely: strings that contain matching substrings*

# Special characters

These characters have special meaning:

```
. ^ $ + ? * ( ) [ ] { } \
```

Use the backslash \ to signal whether you want them to be taken literally:

```
\. \^ \$ \+ \? \* \( \) \[ \] \{ \} \\
```

*Actually, 2 backslashes \\ in R, since \ is special for R too*

# Literals

| regular expression | example matches* |
| --- | --- |
| “\.com” | “google.com”, “apple.com” |
| “\$100” | “$100”, “$100,999” |
| “\!” | “!!!”, “bang!”, “omg!” |

# Character classes

- Specified with square brackets [ ]
- Match any characters within square brackets
- Use dash “-” to specify a range of characters
- Use [^ ] to invert the class – all characters not inside [^ ]

# Character classes

| regular expression | example matches* |
|---|---|
| [ab] | “apple”, “bump”, “abacus” |
| [0-9] | “1978”, “2011”, “3.14” |
| [a-zA-z] | “car”, “Vince Q.Vu” |
| [^0-9] | “apple”, “car”, “pic.jpg” |

# Named classes

For convenience many commonly used classes can be referred to by name

| named class | members |
|---|---|
| [:alnum:] | alphanumeric characters |
| [:alpha:] | alphabetic characters |
| [:digit:] | digits 0 1 2 ... 9 |
| [:space:] | whitespace characters |
| [:punct:] | punctuation characters |
| [:graph:] | [:alnum:] and [:punct:] |

** see help for 'regex' in R for more*

# Dot

“.” matches any single character except newline

| regular expression | example matches* |
|---|---|
| “c.t” | “cat”, “mascot”, “c0t” |
| “.ing” | “king”, “ding”, “9ing” |
| “[[:alnum:]]” | “hw5”, “vqv1978” |

# Modifiers

Allow us to construct more complex patterns

| modifier | meaning |
|---|---|
| ^ | anchor expression to beginning |
| $ | anchor expression to end |
| * | match 0 or more occurrences of preceding |
| ? | match 0 or 1 occurrences of preceding |
| + | match 1 or more occurrences of preceding |
| {n} | match exactly *n* occurrences of preceding |
| {n,} | match at least *n* occurrences of preceding |
| {n,m} | match between *n* and *m* occurrences |
| ( ) | group patterns together |

# Modifiers

| regular expression | example matches* |
|---|---|
| “aa+” | “aardvark”, “aaa” |
| “[:alnum:]+\.jpg” | “picture01.jpg”, “filename.jpg” |

# Warning!

The quantifiers

`+ * {n,}`

are **greedy**!

# Modifiers

| regular expression | example matches* |
| --- | --- |
| “<.*>” | “<July 11> <June 10>” |

# Alternatives (ORing)

- Use the vertical bar '|' to separate alternative patterns
- This is like the logical OR operation, or taking the union of sets

# Alternatives

| regular expression | example matches* |
| --- | --- |
| “apple\|orange” | “pineapple”, “orange” |
| “cat\|dog” | “cat”, “dogs” |
| “\.edu\|\.org\|\.net” | “cmu.edu”, “xkcd.org”, “php.net” |

# Regular Expressions in R

- Regular expressions are simply character strings in R
- Need to double backslash \\ to signal specials to be taken literally – because \ is a special character in R too
- Can use R to programmatically construct regular expressions

# Summary

- Use regular expressions to describe patterns
- grep() to search strings for matches to a pattern
- gsub() to replace matching patterns in a string
- Next: Regular Expressions II