Statistical Computing (36-350)

Importing Data from the Web II

Cosma Shalizi and Vincent Vu

November 21, 2011

Agenda

- Regular expressions
  - Construction
  - Debugging
- Example: Continuation from Friday's Lab

Agenda

- **Continuation** from Friday's Lab
  - Forbes.com: Celebrity 100 List
    - *The World's Most Powerful Celebrities*
    - *http://www.forbes.com/wealth/celebrities*
      - *http://www.forbes.com/wealth/celebrities/list*
- **Goal**: Scrape the list into a data frame in R

First steps

- Read the webpage into R or a text editor
- Identify two cases (say, "Tiger Woods" and "Lady Gaga")
  - How do we find these cases in the html?
  - How are these case coded in the html?

```
html <- readLines('http://www.forbes.com/wealth/celebrities/list')
```

```
i <- grep('Tiger Woods', html)
print(html[(i-4):(i+11)])
```

```
<tr>
<td class="rank">6</td>
<td class="company"><a rel="/profile/tiger-woods"></a><img src=" http://
images.forbes.com/media/lists/people/tiger-woods_50x50.jpg " alt="Tiger
Woods" /> <h3>Tiger Woods </h3></td
<td>$75 M</td>
<td class="smallrank">14</td>
<td class="smallrank">6</td>
<td class="smallrank">5</td>
<td class="smallrank">40</td>
<td class="smallrank">38</td>
</tr>
```

Next steps

- Construct a **regular expression** to match these strings
- Use **capture groups** to specify the parts that you want
- **Hint**: Do it in pieces – i.e. combine smaller, simpler regular expressions

```
<tr>
<td class="rank">6</td>
<td class="company"><a rel="/profile/tiger-woods"></a><img src="
http://images.forbes.com/media/lists/people/tiger-woods_50x50.jpg
" alt="Tiger Woods" /> <h3>Tiger Woods </h3></td
<td>$75 M</td>
<td class="smallrank">14</td>
<td class="smallrank">6</td>
<td class="smallrank">5</td>
<td class="smallrank">40</td>
<td class="smallrank">38</td>
</tr>
```

Problem?

- A single case is split over multiple lines, and so multiple strings
- **Solution**: Paste lines together, separated by ‘\n’ (newline)

Make one long string

```
html <- paste(html, collapse = '\n')
```

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([[:alpha:][:space:]]+)</h3></td>',
    '<td>\\$([[:digit:]]+)M</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '</tr>',
sep = '\\s*')

# Make one long string
html <- paste(html, collapse = '\n')

# First pass: extract disjoint cases
m <- gregexpr(pat, html, ignore.case = TRUE)
x <- regmatches(html, m)
x <- do.call(c, x)
```

Debugging a Regular Expression

- Test subsets of the regular expression to ensure that they work correctly
  - Easier if we use `paste()` to construct the regular expression.
  - Comment out some parts and then test the *regex*

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([[:alpha:][:space:]]+)</h3></td>',
    '<td>\\$([[:digit:]]+)M</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '</tr>',
sep = '\\s*')
```

Test small subsets of the regular expression

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    # '<img src="[^"]*" alt="[^"]*" />',
    # '<h3>([[:alpha:][:space:]]+)</h3></td>',
    # '<td>\\$([[:digit:]]+)M</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '</tr>',
sep = '\\s*')

print(gregexpr(pat, html, ignore.case = TRUE))
```

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([[:alpha:][:space:]]+)</h3></td>',
    '<td>\\$([[:digit:]]+)M</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '</tr>',
sep = '\\s*')

print(gregexpr(pat, html, ignore.case = TRUE))
```

```
<tr>
<td class="rank">6</td>
<td class="company"><a rel="/profile/tiger-woods"></a><img src="
http://images.forbes.com/media/lists/people/tiger-woods_50x50.jpg
" alt="Tiger Woods" /> <h3>Tiger Woods </h3></td
<td>$75 M</td>
<td class="smallrank">14</td>
<td class="smallrank">6</td>
<td class="smallrank">5</td>
<td class="smallrank">40</td>
<td class="smallrank">38</td>
</tr>
```

Space!

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([[:alpha:][:space:]]+)</h3></td>',
    '<td>\\$([[:digit:]]+)M</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '</tr>',
sep = '\\s*')
```

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([[:alpha:][:space:]]+)</h3></td>',
    '<td>\\$([[:digit:]]+)\\s*M</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '<td class="smallrank">(\\d+)</td>',
    # '</tr>',
sep = '\\s*')
```

Continue the process of testing increasing larger subsets of the regular expression, until the entire regular expression is verified

Final steps

- Extract the matches and capture groups
- Convert to a data frame
  - Use ldply() or do.call(rbind, ...)

```r
# Second pass: extract capture groups
m <- regexec(pat, x, ignore.case = TRUE)
celebs <- regmatches(x, m)


# Put it all together
library(plyr)
df <- ldply(celebs, function(x) data.frame(
          rank = as.numeric(x[2]),
          name = x[3],
          pay = as.numeric(x[4]),
          money.rank = as.numeric(x[5]),
          tvradio.rank = as.numeric(x[6]),
          press.rank = as.numeric(x[7]),
          web.rank = as.numeric(x[8]),
          social.rank = as.numeric(x[9]),
stringsAsFactors = FALSE))
```

Problems?

- Did we miss anybody?
- Who?
- Go back to the regular expression and fix it

Need to allow digits and punctuation marks too

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([[:alpha:][:space:]]+)</h3></td>',
    '<td>\\$([[:digit:]]+)\\s*M</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '</tr>',
sep = '\\s*')
```

Need to allow digits and punctuation marks too

```
pat <- paste(
    '<tr>',
    '<td class="rank">(\\d+)</td>',
    '<td class="company"><a rel="[^"]*"></a>',
    '<img src="[^"]*" alt="[^"]*" />',
    '<h3>([^>]+)</h3></td>',
    '<td>\\$([[:digit:]]+)\\s*M</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '<td class="smallrank">(\\d+)</td>',
    '</tr>',
sep = '\\s*')
```

Summary

- Construct regular expressions in parts
  - Use paste()
- Test subsets of the regular expression
  - Use comment marker #
- Next: Reshaping data and databases