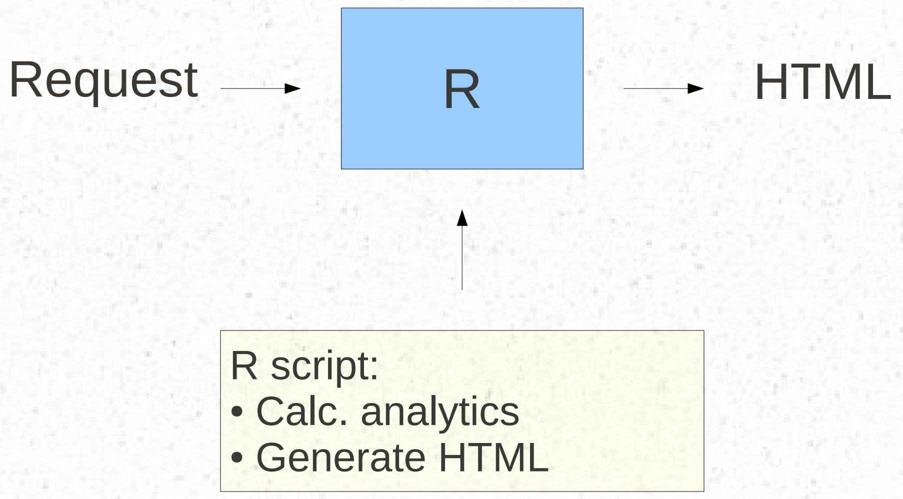
Formatting R Output with HTML

What's the problem?

- Want to present R output in readable fashion.
- HTML is a great presentation tool.
- Allows web page presentation.
- Great for e-mail generation, too.
- Satisfactory for printing.

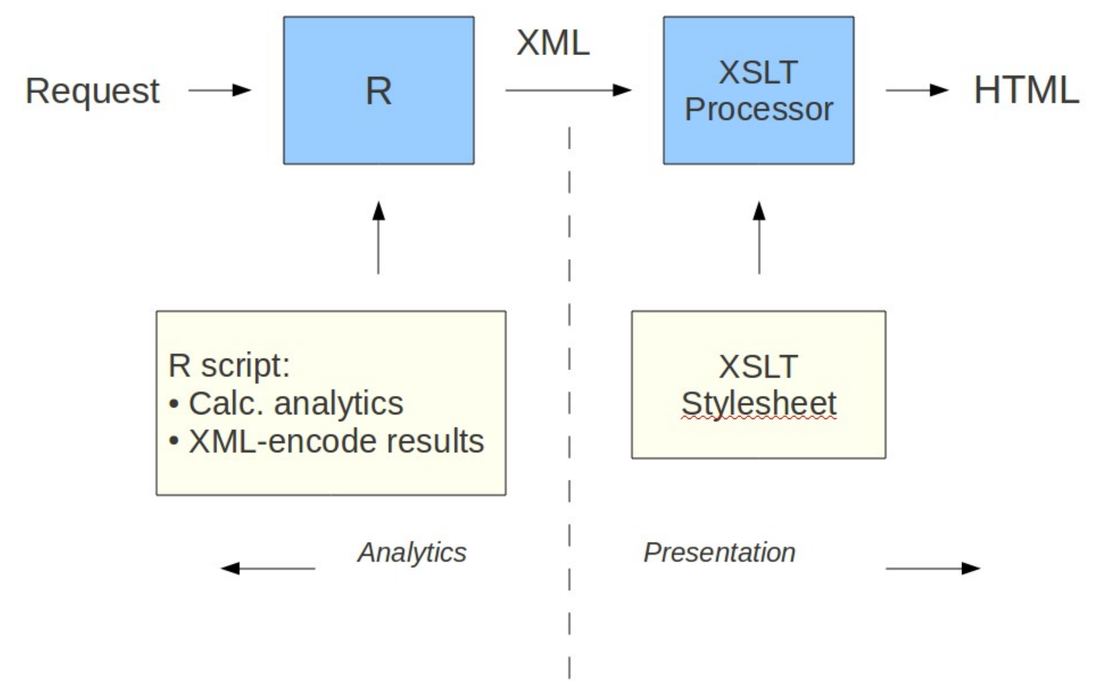
Solution #1: Generate HTML from within R



Solution #1 is quick and dirty, but it has problems.

- Many packages help with HTML generation: R2HTML, hwriter, xtable, GGIwithR, HTMLUtils, prettyR, Rpad, and more.
- Great for simple output and basic HTML.
- But does not scale well and hard to modify.
- Violates a basic software design principle: *Separate the analytics from the presentation layer.*

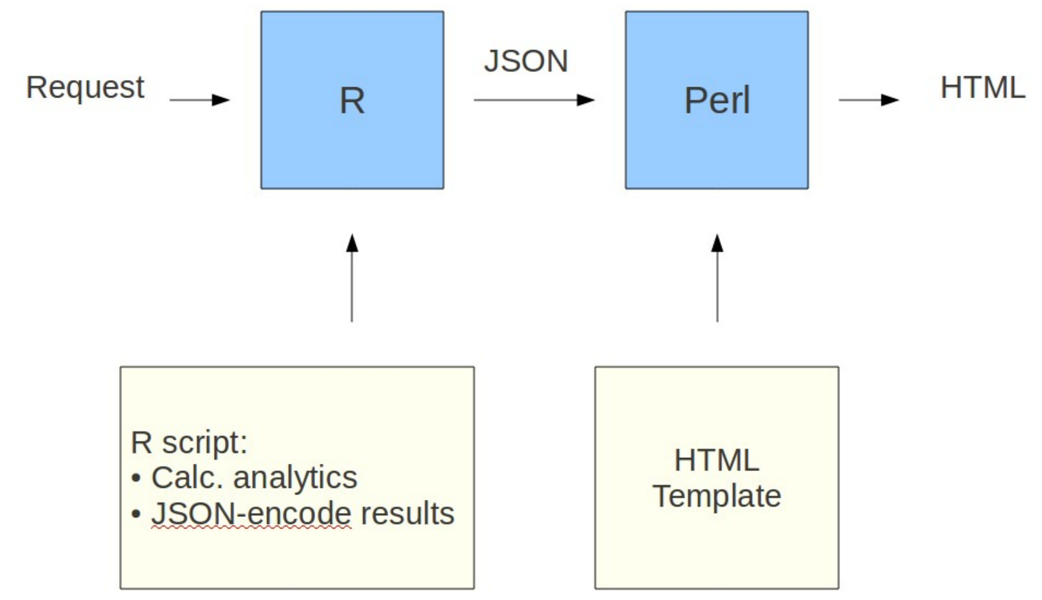
Solution #2: Generate XML, transform to HTML



Solution #2 is a better design, but XML and XSLT can be clumsy.

- XSLT is very powerful, but it's a crappy programming language.
- Hard to code, hard to read.
- Also, XML tools are very heavy-weight due to semantics of the language.
- Bottom line: Powerful solution, but annoying to implement.

Solution #3:
Generate JSON, transform to HTML



JSON is “JavaScript Object Notation”

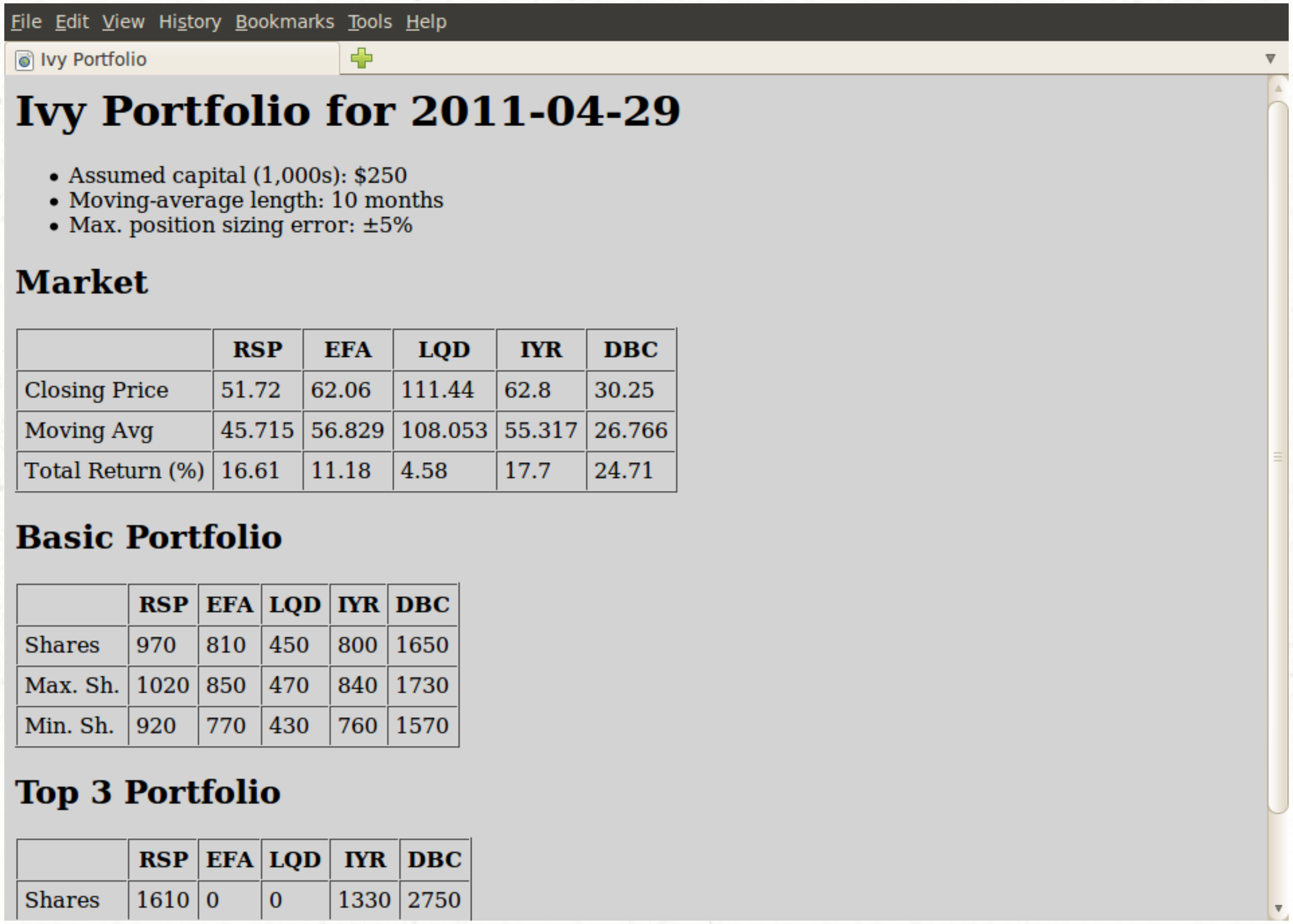
- ASCII-based data representation, good for exchanging structured data betw. programs
- In JSON, the R vector `c(1,3,5)` becomes:

```
[1,3,5]
```

- And `list(a=10,b=20,c=30)` becomes:

```
{"a":10,"b":20,"c":30}
```

Example: Meb Faber's Ivy Portfolio



Ivy Portfolio for 2011-04-29

- Assumed capital (1,000s): $250
- Moving-average length: 10 months
- Max. position sizing error: ±5%

Market

	RSP	EFA	LQD	IYR	DBC
Closing Price	51.72	62.06	111.44	62.8	30.25
Moving Avg	45.715	56.829	108.053	55.317	26.766
Total Return (%)	16.61	11.18	4.58	17.7	24.71

Basic Portfolio

	RSP	EFA	LQD	IYR	DBC
Shares	970	810	450	800	1650
Max. Sh.	1020	850	470	840	1730
Min. Sh.	920	770	430	760	1570

Top 3 Portfolio

	RSP	EFA	LQD	IYR	DBC
Shares	1610	0	0	1330	2750

The R script assembles its output into a list and calls **`toJSON`** *function*

```
library(rjson)
    (do calculations...)
output <- list(
     assumedCapital = CAPITAL / 1000,
     asof = format(end(closes)),  . . . )
cat(toJSON(output))
```

The Perl script decodes the JSON and expands the HTML template

```
use JSON;
use Template;
use constant TMPL_FILE => “/my/templates/file.tmpl”;
my %config = ( . . . );
my $raw = <>;
my $json = new JSON::XS;
my $rout = $json->allow_nonref->decode($raw);
my $tmpl = Template->new(\%config) or die;
my $html;
$tmpl->process(TMPL_FILE, $rout, \$html) or die;
print $html, “\n”;
```

The template is HTML with R-defined values inserted into it.

```
<html>
<head>
<title>Ivy Portfolio</title>
</head>
<body>
<h1>Ivy Portfolio for [% asof %]</h1>
<ul>
    <li>Assumed capital (1,000s): $[% assumedCapital %]</li>
    <li>Moving-average length: [% period %] months</li>
    <li>Max. pos. sizing error: &plusmn;[% maxSizeError %]%</li>
</ul>

(etc. . . .)
```

Works great!

- Pretty easy to implement.
- Cleanly separates analytics from presentation layer.
- Simplifies changes and maintenance.
- I currently use this design for
  - Web page generation
  - Batch jobs which generate HTML e-mail

Formatting R Output with HTML

Paul Teetor

paulteetor@yahoo.com
@pteetor
http://quanttrader.info/public