

Bits and Bytes

How does the computer store numbers and other information?

- A **bit** is a single binary digit – 0 or 1
- Short for **binary digit**
- Tukey – Legendary statistician and the father of “Exploratory Data Analysis” coined the term
- A byte is a collection of 8 bits – 0001 0011

Usually written with a space between the two sets of 4 digits for readability

Bits and Characters

- ASCII – American Standard Code for Information Interchange
- Character encoding scheme – each upper and lower case letter in the English alphabet and other characters such as # and \$ represented as a sequence of 7 bits
- First introduced in the 1960s
- Today Universal Character Set (aka Unicode) is more common UTF-8, UTF-16 and UTF-32

Glyph	ASCII	Unicode
#	0010 0011	0000 0000 0010 0011
\$	0010 0100	0000 0000 0010 0100
A	0100 0001	0000 0000 0100 0001
a	0110 0001	0000 0000 0110 0001
©		0000 0000 1010 1001
æ		0000 0000 1110 0110
Δ		0000 0011 1001 0100
α		0000 0011 1011 0001

ASCII and Unicode mappings are compatible for the $2^7 = 128$ ASCII characters. The bottom 4 characters do not have encodings in ASCII

Representing Numbers

Recall that when we write a 3-digit number, e.g.,

105

We are using the decimal system and we mean:
1 hundred, **0** tens, **5** ones,

That is: $(1 * 10^2) + (0 * 10^1) + (5 * 10^0)$
where the digits range from 0, 1, 2, ..., 9

What is the decimal value of the
following 8-digit binary number?

00110001

Value	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
Position	7	6	5	4	3	2	1	0	
Base 2	0	0	1	1	0	0	0	1	00110001
Decimal	0	0	32	16	0	0	0	1	32+16+1 = 49

Representing Numbers in Binary

- We can do the same to represent numbers in binary
- The binary number:

1101001

- Now we have powers of 2 and digits 0 and 1:
 $(1 * 2^6) + (1 * 2^5) + (0 * 2^4) + (1 * 2^3)$
 $+ (0 * 2^2) + (0 * 2^1) + (1 * 2^0)$
- In decimal this is $64+32+8+1 = 105!$

Different Types of Numbers

- Integer types are stored in the computer as described
- But what about numeric types, e.g.
0.25? Or -3.14? Or 1/3?
- Notice that the computer cannot store 1/3 because it only has so many digits to use
- The computer uses the notion of scientific notation to store numbers

Scientific Notation

- General form:

$$a * 10^b$$

a: mantissa b: exponent

10: base And sign +/-

0.023 -> $2.3 * 10^{-2}$

-2100 -> $-2.1 * 10^3$

How does this impact our work?

- There is a limit to how precisely we can represent numbers.
- Need to be aware of this when doing calculations.
- For example, in many cases it is better to do calculations on the log scale.
- Example: instead of multiplying two numbers, take sum of logs, then exponentiate back only when strictly necessary.

Double-Precision Floating Point

- 8 bytes (64 bits)
- Sign bit: 1 bit
- Exponent: 11 bits
- Mantissa/significand: 53 bits (stored as 52)

$$(-1)^{\text{sign}}(1.b_{51}b_{50}...b_0)_2 \times 2^{e-1023}$$





or

$$(-1)^{\text{sign}} \left(1 + \sum_{i=1}^{52} b_{52-i} 2^{-i} \right) \times 2^{e-1023}$$

EXAMPLES IN R

Representation of Colors, Data, and HTML basics

Colors: (rgb)

	(255, 0, 0)	#FF0000
	(255, 255, 0)	#FFFF00
	(100, 149, 237)	#6495ED
	(241, 123, 139)	#F17B8B



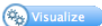

Representation of Data

HTML table, Excel Spreadsheet, plain
text

ManyEyes html

[View as text](#)

		Population Using Internet	Percent of Generation that goes online	Percent of the online population that watch video online
1	Millenials	35%	95%	80%
2	Gen X	21%	86%	66%
3	Younger Boomers	20%	81%	62%
4	Older Boomers	13%	76%	55%
5	Silent Generation	5%	58%	44%
6	G.I.Generation	3%	30%	20%
7	Total Population		79%	66%

 watch this  add to topic center  Visualize  rate this

Versions (1)

ManyEyes text

Population Using Internet			Percent of Generations
Millennials	35%	95%	80%
Gen X	21%	86%	66%
Younger Boomers	20%	81%	62%
Older Boomers	13%	76%	55%
Silent Generation	5%	58%	44%
G.I. Generation	3%	30%	20%
Total Population		79%	66%

ManyEyes xlsx

	A	B	C	D	E	F
1		Population U	Percent of Gr	Percent of the online population that watch vi		
2	Millennials	35%	95%	80%		
3	Gen X	21%	86%	66%		
4	Younger Boo	20%	81%	62%		
5	Older Boome	13%	76%	55%		
6	Silent Gener	5%	58%	44%		
7	G.I. Generati	3%	30%	20%		
8	Total Population		79%	66%		
9						
10						
11						

	txt	html	xlsx
browser	Render w/ no markup	Format according to markup	Open file in Excel
Excel	Display as Excel	Display as Excel	Display
TextEditor	Display ASCII characters	See markup as well as content	See nothing or gibberish

Hypertext Markup Language (HTML)

Useful for data scientists: Primarily to know what to do when we need to go through html files to get to data.

What is HTML?

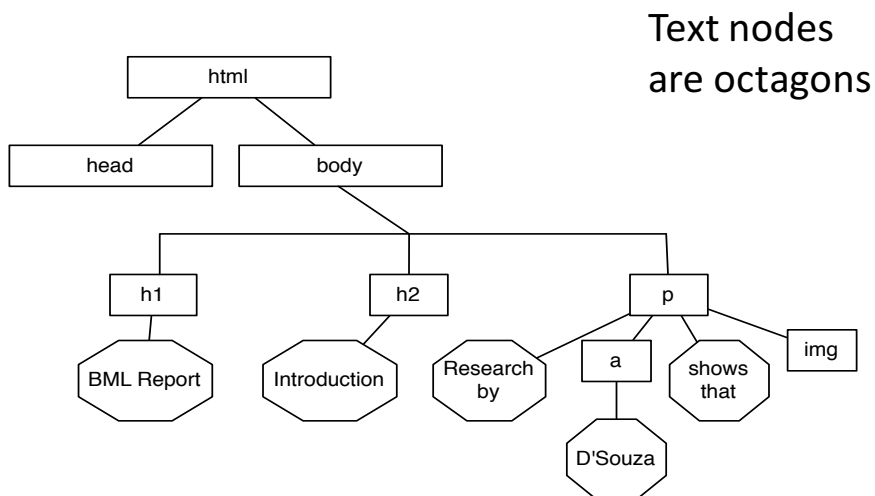
- Hypertext Markup Language
- Describes the structure of web pages by adding annotations to the content
- HTML *elements* are the building blocks
- Elements are labeled by *tags* – paragraph, table, image, etc.
- Web browsers do not display tags, but use them to decide how to display the content

An HTML Document

```
<html>
<head></head>
<body>
  <h1>BML Report</h1>
  <h2>Introduction</h2>
  <p>
    Research by
    <a href="http://google.com">D'Souza</a>
    that...
    
  </p>
</body>
</html>
```

shows

Tree Data Structure



Tree Hierarchy

- One root node
- Root node has child nodes and each of these can have child nodes and so on
- Any node must have one and only one parent

Element Syntax

- Each HTML element has an element name, e.g.
 - `body` : the main content of the page
 - `h1` : largest header
 - `p` : paragraph
 - `br` : line break

Element Syntax



- The end tag is a slash and the name surrounded by angle brackets: `</h1>`
- Some HTML elements have no content: `
` is for a line break

Element Content

- Simple content is plain text:

```
<h1>This is a title</h1>
```

- Complex content includes other elements.

```
<p>This paragraph includes  
<a href="http://...">a link</a>  
and sentences.</p>
```

How many child elements does this `<p>` node contain?

3: the text before the `<a>`, the `<a>` node and the text after the `<a>` node

Attribute Syntax

- *Attributes* provide additional information to an HTML element.
- Attributes always come in name/value pairs like this: `name="value"`
- Attributes are always specified in the start tag of an HTML element.

Well-formed XHTML

- Well-formed HTML is called XHTML.
- Tag names follow strict rules for matching case
- Attribute values must be in quotes
- Elements must be properly nested (i.e. you can draw a tree with it)

Examples of HTML

An HTML Table

- Tables are defined with the `<table>` tag.
- A table has rows marked up with the `<tr>` tag.
- Each row is divided into data cells with the `<td>` tag. (td stands for table data).
- A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.
- Headings in a table are defined with the `<th>` tag.

Table in HTML

```
<table>
  <tr>
    <th>A</th>
    <th>B</th>
  </tr>
  <tr>
    <td>1</td>
    <td>25,000</td>
  </tr>
  <tr>
    <td>7</td>
    <td>100,000</td>
  </tr>
</table>
```

Appears as:

A	B
1	25,000
7	100,000

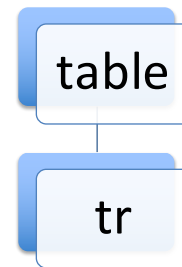
Can you draw the tree for this document?

<table>



<table>

<tr></tr>



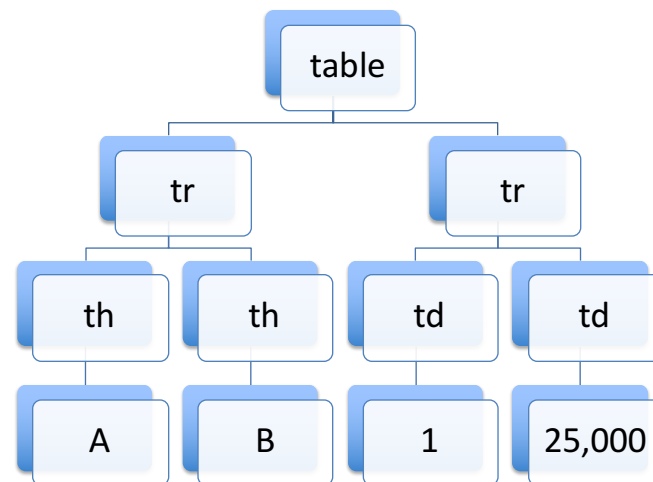
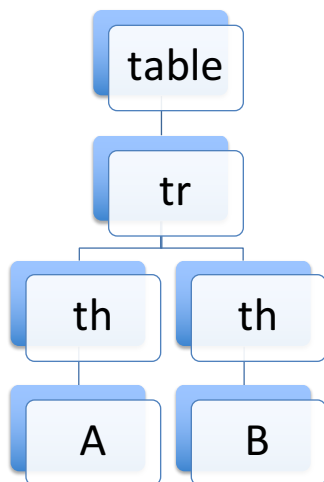
<table>

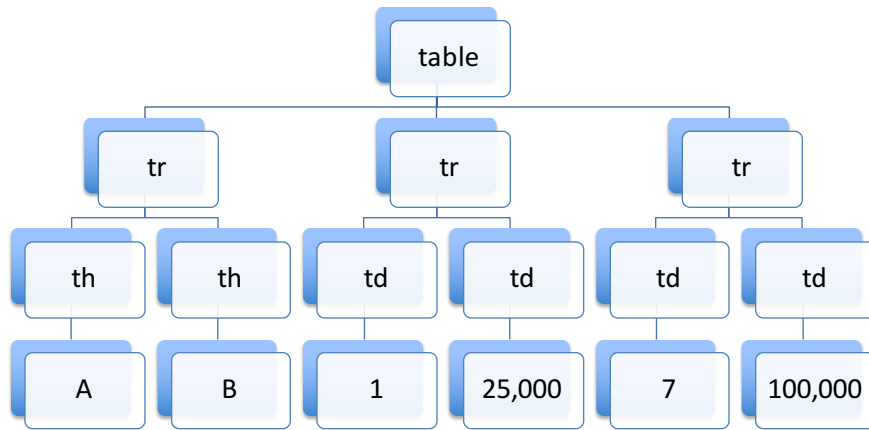
<tr>

<th>A</th>

<th>B</th>

</tr>





```

<table cellpadding="6" border="2">
<tr>
  <th>A</th>
  <th>B</th>
</tr>
<tr align="right">
  <td>1</td>
  <td>25,000</td>
</tr>
<tr align="right">
  <td>7</td>
  <td>100,000</td>
</tr>
</table>
  
```

Modified Table

Appears as:

A	B
1	25,000
7	100,000

Unordered Lists

- Unordered lists have items marked with bullets.

```

<ul>
  <li>Coffee</li>
  <li>Milk</li>
</ul>
  
```

Appears as:

- Coffee
- Milk

- Paragraphs, line breaks, images, links, other lists, etc. can be placed in a list

Ordered Lists

- Ordered lists have items marked with numbers.

```

<ol>
  <li>Coffee</li>
  <li>Milk</li>
</ol>
  
```

1. Coffee
2. Milk

Paragraphs and Sections

```
<h1>My BML Report</h1>
```

```
<h2>Introduction</h2>
```

```
<p>
```

The BML model is a simple traffic model...

```
</p>
```

```
<p>
```

We studied the BML model behavior for...

```
</p>
```

Appears as:

My BML Report

Introduction

The BML model is a simple traffic model...

We studied the BML model behavior for...

Images

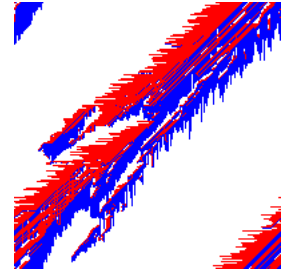
- The img tag is used to embed images in a Web page

```

```

- The src attribute gives the file name for the image
- The width attribute is optional
- This tag is empty – the start and end tag are collapsed.

Appears as:



Links

```
<a href="http://mae.ucdavis.edu/dsouza/">D'Souza</a>  
discovered ....
```

Appears as:

Introduction

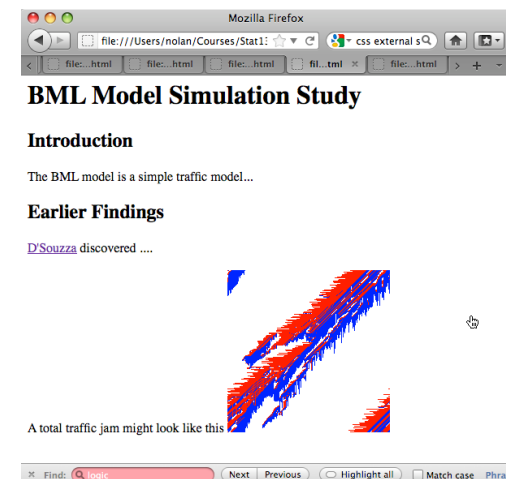
[D'Souza](#) discovered

`<a>` is an *anchor tag*

The content is the text that is “clickable”

The link can be to another place within the document

A BML Report

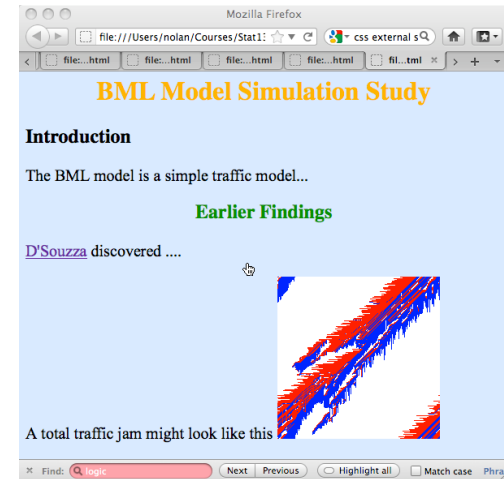


Raw HTML for the Report

```
<html>
<head></head>
<body>
<h1>BML Model Simulation Study</h1>
<h2>Introduction</h2>
<p>The BML model is a simple traffic model...</p>
<h2>Earlier Findings</h2>
<p>
<a href="http://mae.ucdavis.edu/dsouza/">D'Souzza</a>      discovered ....
</p>
<p>
A total traffic jam might look like this

</p>
</body>
</html>
```

A prettied up BML Report



Raw HTML for the Stylized Report

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="bmlStyle.css" />
</head>
<body>
<h1>BML Model Simulation Study</h1>
<h2 class="bml">Introduction</h2>
<p>The BML model is a simple traffic model...</p>
<h2>Earlier Findings</h2>
<p>
<a href="http://mae.ucdavis.edu/dsouza/">D'Souzza</a>      discovered ....
</p>
<p>
A total traffic jam might look like this

</p>
</body>
</html>
```

Cascading Style Sheet (CSS)

selector { property: value; }

Selector may be:

- HTML tag name `h1 { color: green; }`
- attribute value for id `#idXYZ { color: blue; }`
- class `.bml { font-size: 2em; }`

bmlStyle.css

```
body  
{ background-color: #d0e4fe; }
```

```
h1  
{ color: orange; text-align: center; }
```

```
h2.bml  
{ color: green; text-align: center; }
```

```
p  
{ font-family: "Times New Roman"; font-size: 20px; }
```