

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)

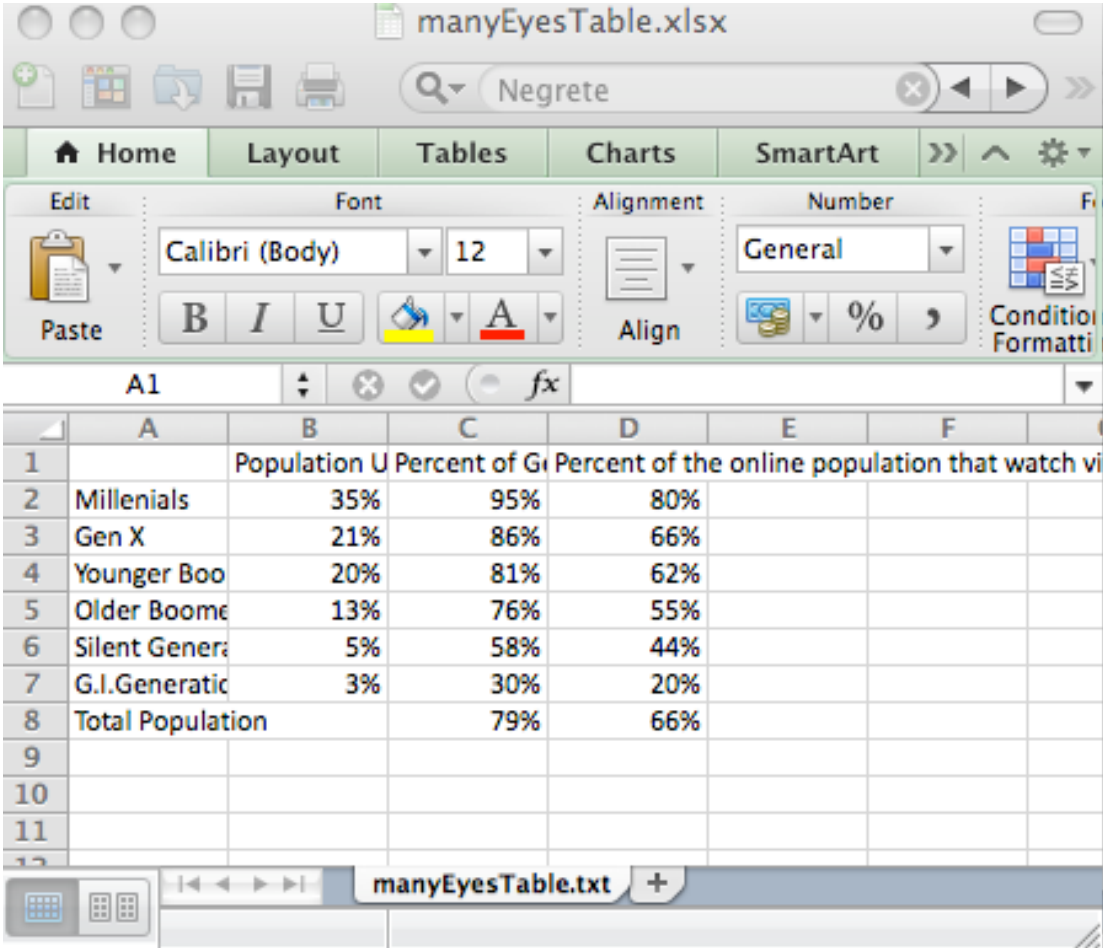
ManyEyes text

| Population Using Internet | | | | Percent of Generation |
|---------------------------|-----|-----|-----|-----------------------|
| 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% |

ASCII & Unicode

| Character | ASCII | Unicode |
|-----------|-----------|---------------------|
| A | 0100 0001 | 0000 0000 0100 0001 |
| a | 0110 0001 | 0000 0000 0110 0001 |
| <i>α</i> | | 0000 0011 1011 0001 |

ManyEyes xlsx



| | A | B | C | D | E | F |
|----|------------------|--------------|--------------|--|---|---|
| 1 | | Population U | Percent of G | Percent of the online population that watch vi | | |
| 2 | Millenials | 35% | 95% | 80% | | |
| 3 | Gen X | 21% | 86% | 66% | | |
| 4 | Younger Boo | 20% | 81% | 62% | | |
| 5 | Older Boome | 13% | 76% | 55% | | |
| 6 | Silent Genera | 5% | 58% | 44% | | |
| 7 | G.I. Generatio | 3% | 30% | 20% | | |
| 8 | Total Population | | 79% | 66% | | |
| 9 | | | | | | |
| 10 | | | | | | |
| 11 | | | | | | |
| 12 | | | | | | |

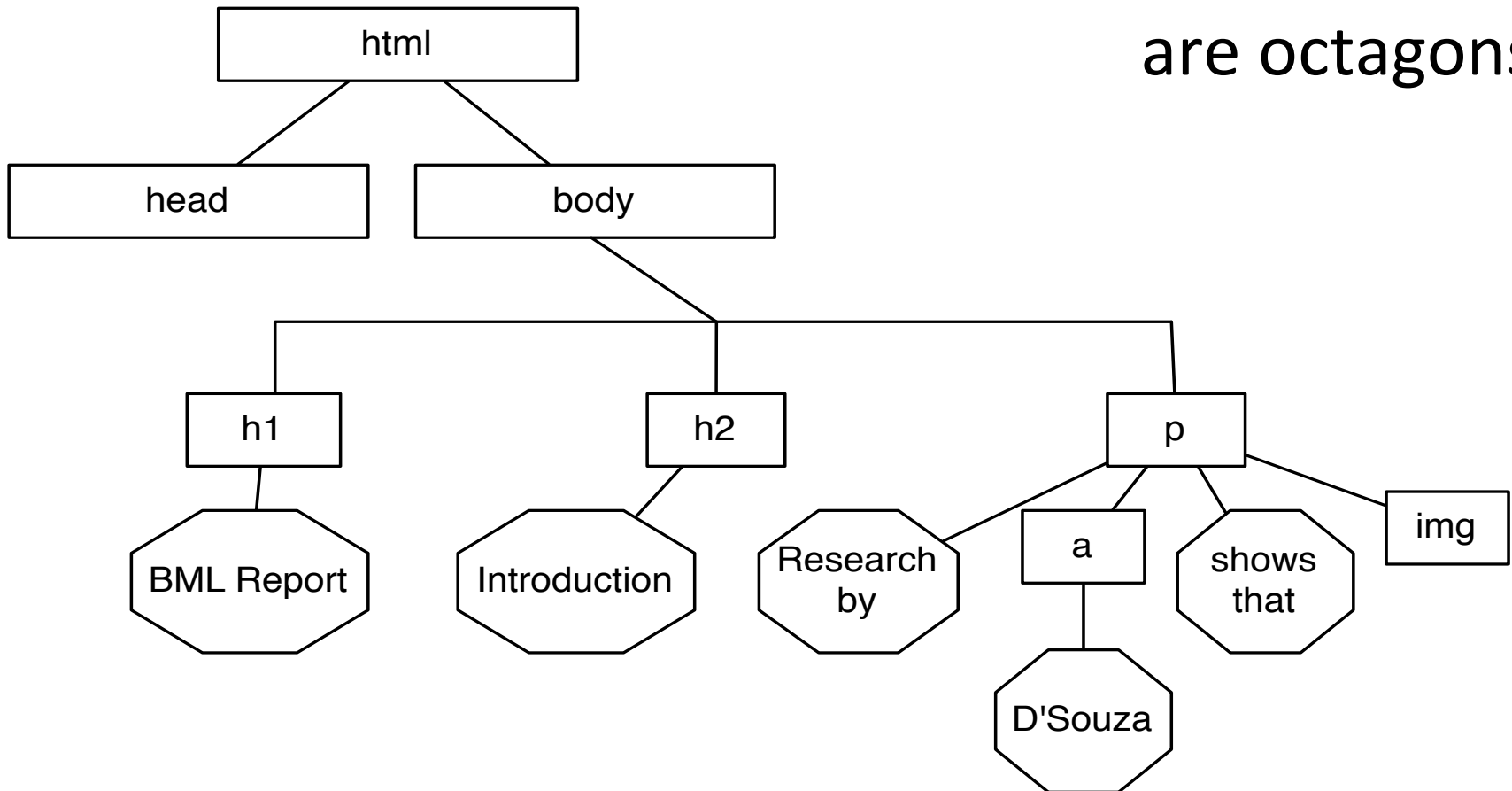
| | txt | html | xlsx |
|------------|-----------------------------|-------------------------------------|-----------------------------|
| browser | Render w/ no markup | Format according to markup | Open file in Excel |
| Excel | Display as Excel | Display as Excel | Display |
| TextEditor | Display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

Hypertext Markup Language

Tree

Tree Data Structure

Text nodes
are octagons



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

Examples of HTML

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?

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.

Modified Table

```
<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>
```

Appears as:

| A | B |
|---|---------|
| 1 | 25,000 |
| 7 | 100,000 |

Unordered Lists

- Unordered lists have items marked with bullets.

Appears as:

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Milk</li>
```

```
</ul>
```

- 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. Appears as:

``

`Coffee`

`Milk`

``

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...

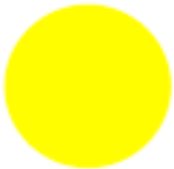
We studied the BML model behavior for...

Colors: (rgb)



(255, 0, 0)

#FF0000



(255, 255, 0)

#FFFF00



(100, 149, 237)

#6495ED



#E41A1C99

Images

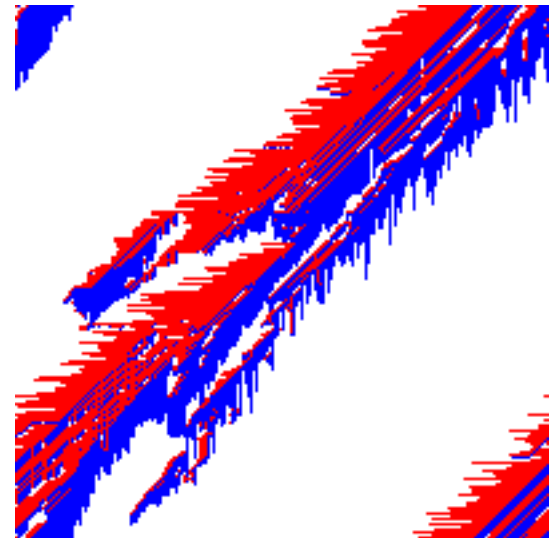
- The img tag is used to embed images in a Web page

```

```

- The src attribute give 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

`
D'Souzza` discovered

Appears as: **Introduction**

D'Souzza discovered

`<a>` is an anchor tag

The content is the text that is “clickable”

The link can be to another place within the document

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

<h1> This is a title </h1>

Start tag Text Content End tag

- 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 link and sentences.</p>

How many child elements does this <p> node contain? 3

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)

A BML Report

Mozilla Firefox

file:///Users/nolan/Courses/Stat1: css external s

file:...html file:...html file:...html fil...tml file:...html

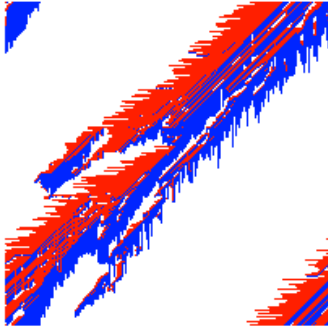
BML Model Simulation Study

Introduction

The BML model is a simple traffic model...

Earlier Findings

[D'Souzza](#) discovered



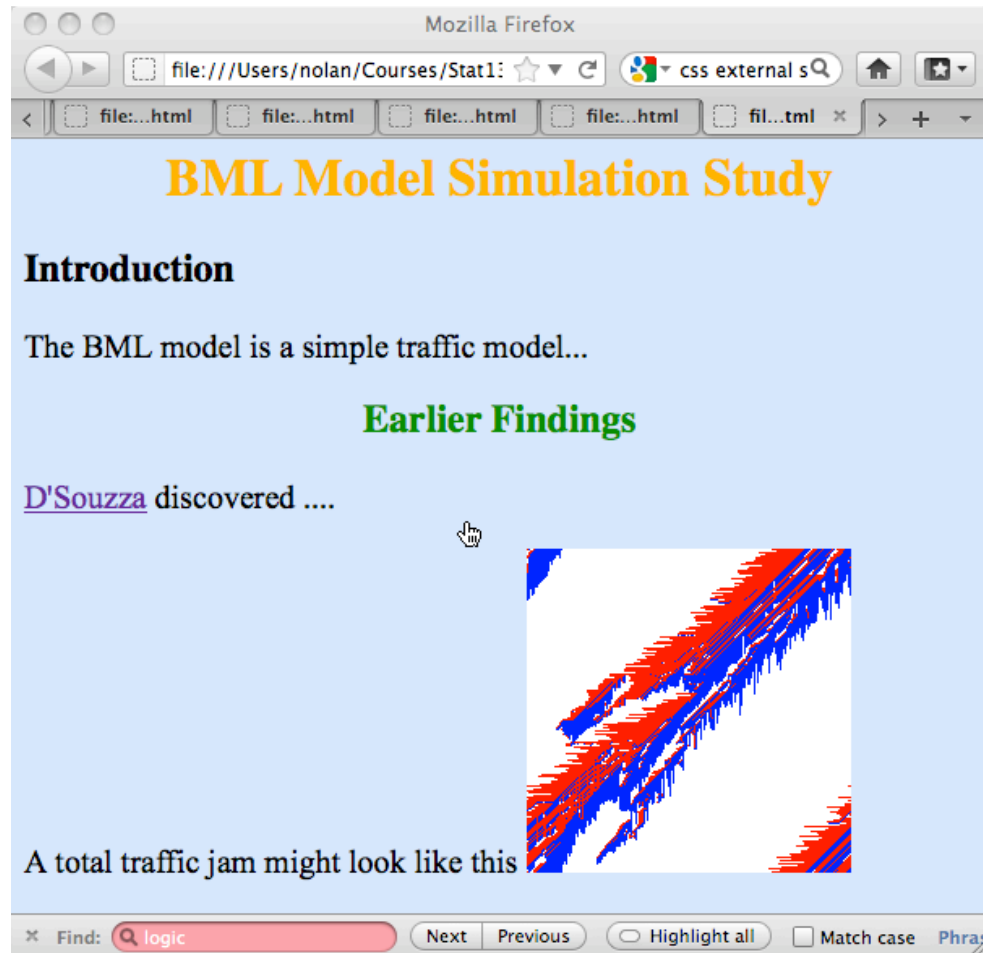
A total traffic jam might look like this

Find: logic Next Previous Highlight all Match case Phras

Raw HTML for the Stylized 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>Introduction</h2>
<p> The BML model is a simple traffic model... </p>
<h2 class="bml">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

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; }
```


Web Scraping

Data Available on the Web

- HTML
 - Table
 - plain text format
 - HTML form to request data
- Other Format:
 - JSON
 - XML

Scraping data from a Web page

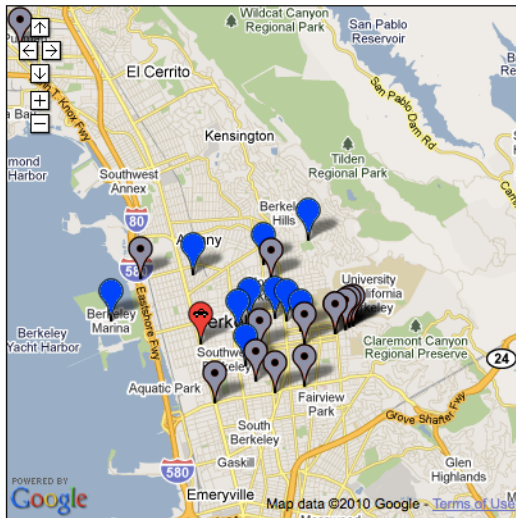
- Means to write code to automatically extract data from one or more web pages.
- HTML is like XML – We can use parsing capabilities in the XML package.
`htmlParse()` can create a tree structure from ill-formed HTML.
- The information is all in text and we may need to use regular expressions to extract the relevant pieces

HTML

- Much of the data available on the web is not provided as a separate downloadable file; it's embedded in the website itself.

All incidents within 7 days

Displaying 1 - 25 of 28 incidents. < Previous | Next >



| Top categories | | |
|--------------------|----|-----|
| Theft | 10 | 36% |
| Disturbances | 5 | 18% |
| VANDALISM | 5 | 18% |
| Aggravated Assault | 3 | 11% |
| Burglary | 3 | 11% |
| Stolen Auto | 1 | 4% |
| Robbery | 1 | 4% |

| Case num | Date | Category | Offense | Location |
|--------------------------|--------------------|--------------------|------------------------|-----------------------|
| 10066504 | 2010-11-01 03:40PM | Disturbances | Disturbance | 1600 Block Milvia St |
| 10066476 | 2010-11-01 11:45AM | Burglary | BURGLARY RESIDENTIAL | 2400 Block Warring St |
| 10066502 | 2010-11-01 10:30AM | Aggravated Assault | ASSAULT/BATTERY FELONY | 600 Block Gilman St |
| 10066447 | 2010-11-01 09:59AM | Theft | THEFT FELONY | 2100 Block McGee Av |
| 10066443 | 2010-11-01 09:35AM | Theft | THEFT FROM AUTO | 1600 Block Carleton |
| 10066431 | 2010-11-01 07:58AM | Disturbances | DOMESTIC VIOLENCE | 3000 Block Martin Lu |
| 10066419 | 2010-11-01 03:05AM | Robbery | Robbery | 2700 Block Haste St |

Popular Baby Names



Popular Names by Birth Year

November 7, 2010

Popularity in 2009

| Rank | Male name | Female name |
|------|-------------|-------------|
| 1 | Jacob | Isabella |
| 2 | Ethan | Emma |
| 3 | Michael | Olivia |
| 4 | Alexander | Sophia |
| 5 | William | Ava |
| 6 | Joshua | Emily |
| 7 | Daniel | Madison |
| 8 | Jayden | Abigail |
| 9 | Noah | Chloe |
| 10 | Anthony | Mia |
| 11 | Christopher | Elizabeth |
| 12 | Aiden | Addison |
| 13 | Matthew | Alexis |
| 14 | David | Ella |
| 15 | Andrew | Samantha |
| 16 | Joseph | Natalie |
| 17 | Logan | Grace |
| 18 | James | Lily |
| 19 | Ryan | Alyssa |
| 20 | Benjamin | Ashley |
| 21 | Elijah | Sarah |
| 22 | Gabriel | Taylor |

- Web pages are created when your browser software represents or “renders” a specially formatted (HTML) text file. Most browsers allow you to see this file under something like View > Page Source.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
<title>IMDb Charts</title>
<link rel="canonical" href="http://www.imdb.com/chart/" />
<meta name="title" content="IMDb Charts">
<meta name="description" content="IMDb: The biggest, best, most award-winning movie site on the planet.">

<meta name="keywords" content="movies,films,movie database,actors,actresses,directors,hollywood,stars,quotes">
<link rel="stylesheet" type="text/css" media="screen" href="http://i.media-imdb.com/images/SF05d2bd4730c135f1c4bceb
<script type="text/javascript" src="http://i.media-imdb.com/images/SF3ee6861263732f8e66aaecfd1850b466/a/js/ads.js">

<script type="text/javascript">
    generic.monitoring.set_twilight_info("chart", "US", "83e50b7a3b50b8f7118fc7864f59906f20c5aeb4", "2009-10-07T18%
</script>

<script type="text/javascript">
    generic.monitoring.start_timing("page_load");
</script>
<script type="text/javascript">
    var aan = {
        url:"http://aan.amazon.com/2009-05-01/imdb/default?slot=sitewide-iframe&ord=[CLIENT_SIDE_ORD]",
        oncall:custom.amazon.aan_iframe_oncall
    }
</script>
<iframe src="/images/SF0d0def01846066a8fbf875202fe91fcd/a/js/scriptloader.html#aan" style="width:0px;height:0px;dis
<link rel="icon" href="http://i.imdb.com/favicon.ico" />
<link rel="apple-touch-icon" href="http://i.media-imdb.com/apple-touch-icon.png" />
```

IMDb Charts: IMDb Top 250

IMDb Charts

[Main index](#)

IMDb Top 250

[IMDb Bottom 100](#)

US Box Office

[USA Top 10](#)

[USA Archive](#)

UK Box Office

[UK Top 10](#)

[UK Archive](#)

All-Time Box Office

[USA](#)

[Non-USA](#)

[World-wide](#)

DVD Rentals

[USA Weekly Top 20](#)

Top 250 movies as voted by our users

For this top 250, only votes from regular voters are considered.

Track which films you've seen from the Top 250 [right here!](#)

| Rank | Rating | Title | Votes |
|------|--------|---|---------|
| 1. | 9.2 | The Shawshank Redemption (1994) | 734,730 |
| 2. | 9.2 | The Godfather (1972) | 549,126 |
| 3. | 9.0 | The Godfather: Part II (1974) | 346,262 |
| 4. | 8.9 | Pulp Fiction (1994) | 577,751 |
| 5. | 8.9 | The Good, the Bad and the Ugly (1966) | 229,684 |
| 6. | 8.9 | 12 Angry Men (1957) | 180,224 |
| 7. | 8.9 | Schindler's List (1993) | 386,132 |
| 8. | 8.8 | The Dark Knight (2008) | 669,650 |
| 9. | 8.8 | The Lord of the Rings: The Return of the King (2003) | 516,592 |
| 10. | 8.8 | One Flew Over the Cuckoo's Nest (1975) | 309,957 |
| 11. | 8.8 | Star Wars: Episode V - The Empire Strikes Back (1980) | 374,476 |
| 12. | 8.8 | Fight Club (1999) | 556,117 |
| 13. | 8.8 | Inception (2010) | 524,009 |



1st pair free

Get your first pair of glasses FREE!

*See site for details

Coastal.com SHOP GLASSES »

[ad feedback](#)

Our goal:
Extract this
information and
put it in a
dataframe

- Look at website then switch to Rstudio.



Support



My Account



FAQ



My Order

search

Home

GeoIP

minFraud

Contact

Company

**GeoIP
Documentation**[ISO 3166-1 Codes](#)[ISO 3166-2 Codes](#)[FIPS 10-4 Codes](#)**Country Lat/Long**[US State Lat/Long](#)[US Metro Code](#)**MaxMind**[Support Center](#)

Average Latitude and Longitude for Countries

This page contains the average latitude and longitude for countries around the world.

Source: *CIA World Factbook*

"iso 3166 country", "latitude", "longitude"

```
AD,42.5000,1.5000
AE,24.0000,54.0000
AF,33.0000,65.0000
AG,17.0500,-61.8000
AI,18.2500,-63.1667
AL,41.0000,20.0000
AM,40.0000,45.0000
AN,12.2500,-68.7500
AO,-12.5000,18.5000
AP,35.0000,105.0000
AQ,-90.0000,0.0000
AR,-34.0000,-64.0000
AS,-14.3333,-170.0000
AT,47.3333,13.3333
AU,-27.0000,133.0000
```

We would like to
extract these data
programmatically,
but they are not in a
table


```

</td>
<td width="15" valign="top"></td>
<td valign="top">Average Latitude and Longitude for Countries </span><p>
<table cellpadding=0 cellspacing=0 border=0 width="100%">
  <tr>
    <td valign="top" bgcolor="#C39E7F"></td>
  </tr>
</table>
<p>

```

This page contains the average latitude and longitude for countries around the world.

 <i>Source: CIA World Factbook</i><p>

```

<pre>
"iso 3166 country","latitude","longitude"
AD,42.5000,1.5000
AE,24.0000,54.0000
AF,33.0000,65.0000
AG,17.0500,-61.8000
AI,18.2500,-63.1667
AL,41.0000,20.0000
AM,40.0000,45.0000
AN,12.2500,-68.7500
AO,-12.5000,18.5000
AP,35.0000,105.0000
AQ,-90.0000,0.0000
AR,-34.0000,-64.0000
AS,-14.3333,-170.0000

```

The page source
shows us that the
data are within a
<pre> element

htmlParse

```
> library(XML)
> latlon = htmlParse(
  "http://www.maxmind.com/app/country_latlon")
> llRoot = xmlRoot(latlon)
> xmlName(llRoot)
[1] "html"
> pres = getNodeSet(llRoot, "//pre")
> length(pres)
[1] 1
> xmlValue(pres[[1]])
[1] "\n\"iso 3166 country\", \"latitude\",
\"longitude\" \nAD, 42.5000, 1.5000 \nAE,
24.0000, 54.0000 \nAF, 33.0000, 65.0000 \nAG,
```

Extract the data from pres

```
> content = xmlValue(pres[[1]])
```

```
> IIDF = read.table(text = content)
```

Error in scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings, : line 2 did not have 2 elements

```
> IIDF = read.table(text = content, skip = 2)
```

```
> head(IIDF)
```

V1

1 AD,42.5000,1.5000

2 AE,24.0000,54.0000

**Remember other
arguments:
header
sep
colClasses**

Warning: Regular Expressions
and HTML strings

The regular expression matching engine performs something called “greedy matching.” This means that it will always try to find the *longest* pattern that satisfies the match. So we need to be careful about specifying exactly what we want.

```
> htmls <- c('An image: <IMG SRC="rx.b.gif">',  
+           '<a href="somepage.html">click here</a>')  
> gregexpr('<.*>', htmls)  
[[1]]  
[1] 11  
attr(,"match.length")  
[1] 19
```

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

```
• [1] 38
```

An HTML tag is surrounded by
< and >; why is this the wrong
way to extract them?

- To specify what was inside the HTML tag, what we really wanted was anything *except* the character “>”:

```
> htmls <- c('An image: <IMG SRC="rx.gif">',  
+            '<a href="somepage.html">click here</a>')  
> gregexpr('<[^>]*>', htmls)  
[[1]]  
[1] 11  
attr(,"match.length")  
[1] 19  
  
[[2]]  
[1] 1 35  
attr(,"match.length")  
[1] 24 4
```

JSON

JavaScript Object Notation

- Text format
- Lightweight data-interchange
- Easy for humans to read and write.
- Easy for machines to parse and generate

JSON Structure

- JSON is built on two structures:
- An unordered collection of comma-separated name:value pairs

`{"lender_id":"matt", "loan_count":23}`

- An ordered array of values

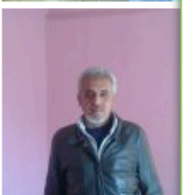
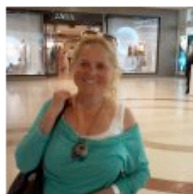
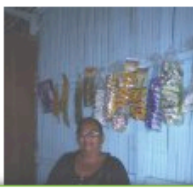
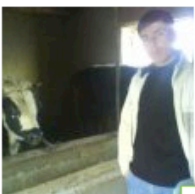
`[{"lender_id":"matt", "loan_count":23}, {"lender_id":"skylar", "loan_count":1}]`

Comparison to XML

- JSON is simpler
- Not as rich – no attributes, unordered, no schema for describing acceptable format
- Compressed JSON and XML not much different in size

[Lend](#)[About](#)[Community](#)[Updates](#)[My Portfolio](#)

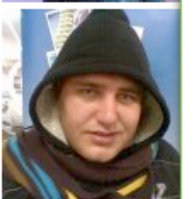
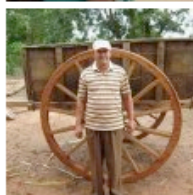
Empower people around the world with a \$25 loan



Suren

 Armenia | Agriculture | Farming

Suren is 58 years old and he lives in Tsovagyugh village, an area of Sevan. Suren is married and he has two children: a daughter and a son....[more](#)

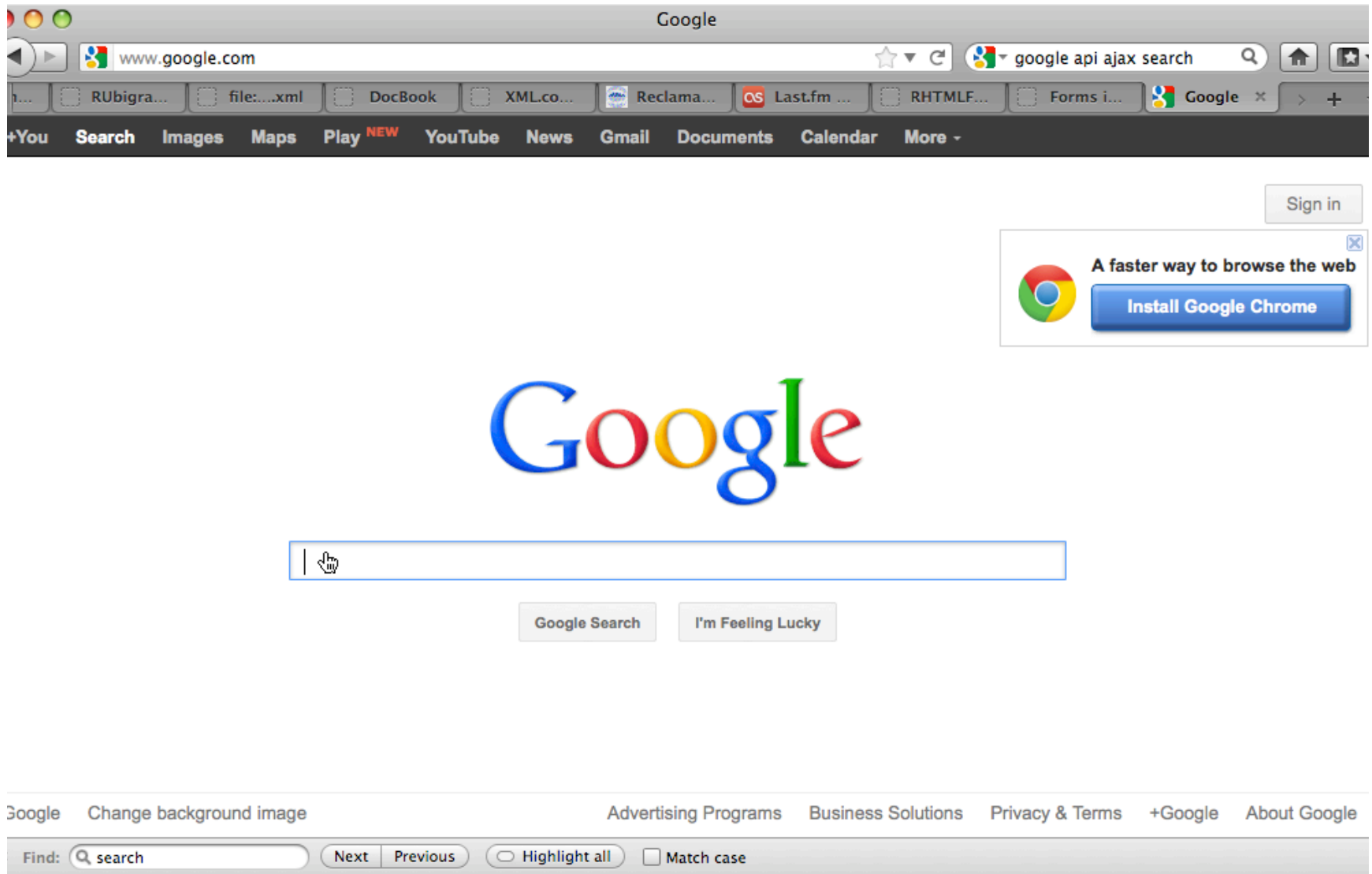
[Lend Now](#)[or Browse all loans](#)

```
{"header":{"total":"576803","page":1,"date":"2010-01-29T20:00:23Z","page_size":1000},
"lenders":[
{"lender_id":"matt",
  "name":"Matt",
  "image":{"id":12829,"template_id":1}, "whereabouts":"San Francisco CA",
  "country_code":"US",
  "uid":"matt",
  "member_since":"2006-01-01T09:01:01Z",
  "personal_url":"www.socialedge.org/blogs/kiva-chronicles",
  "occupation":"Entrepreneur",
  "loan_because":"I love the stories. ",
  "occupational_info":"I co-founded a startup nonprofit (this one!) and I work with an amazing group of people dreaming up ways to alleviate poverty through personal lending. ",
  "loan_count":89,
  "invitee_count":23},
{"lender_id":"jessica",
  "name":"Jessica",
  "image":{"id":197292,
  "template_id":1}, ...
```

Let's Try It

HTML Forms

Simple Form



View the Google page source and See

```
<form id=gbqf name=gbqf  
    method=get  
    action="/search"  
    onsubmit="gbar.logger.il(31);">
```

...

```
<input id=gbqfq class=gbqfif  
    name=q type=text  
    autocomplete=off value="">
```


The GET method

- This is the default method so if no method attribute appears in <form> then it is GET
- The query is constructed as a URL
 - with ? separating URL from parameters
 - & separating parameters
 - parameters supplied as name=value

For example:

`URL?x=2&y=ab+cdef`

There are two parameters here, x and y. The value of x is 2 and the value of y is “ab cdef”

Google API

- The Google Custom Search API lets you develop websites and programs to retrieve and display search results from Google Custom Search programmatically.
- With this API, you can use RESTful requests to get either **web search** or **image search** results in JSON or Atom format.

Older Google Web Search API

- Try entering this URL in a browser:

`"http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=Stat+133")`

- You will get back:

```
{"responseData": {"results":  
[{"GsearchResultClass": "GwebSearch", "unescapedUrl": "http://  
www.stat.berkeley.edu/users/spector...
```

- Do you recognize the format of the return request? JSON

In R we can do the search programmatically

```
> searchResults = getURL(  
  "http://ajax.googleapis.com/ajax/services/  
  search/web?v=1.0&q=Stat+133")
```

```
> SRlist = fromJSON(searchResults)
```

```
> SRlist[[1]]
```

```
http://www.stat.berkeley.edu/users/spector/s133/  
index.html
```

```
https://schedulebuilder.berkeley.edu/explore/  
courses/FL/2012/688
```

RCurl – package in R

- R interface to libcurl - curl is a command line tool for transferring data with URL syntax
- Provides HTTP facilities such as GET and POST to:
 - Post forms to extract data
 - Download files from Web servers
- Functions `getForm()` and `postForm()`

HTTP

Hypertext Transfer Protocol

HTTP

- Protocol for exchanging information
- Foundation of data communication on the Web
- Client-server computing model:
 - Client: e.g. Browser
 - Server: e.g. Application hosting a Web Site

Transaction

- Client submits an HTTP *request* message to the server.
- Server returns a response message to the client.
- Response message contains status information about the request and, possibly, content requested by the client.

POST Forms



RECLAMATION

Great Plains Region

Managing Water in the West

[Reclamation Home](#)[Regional Offices](#)[Newsroom](#)[Library](#)[Projects & Facilities](#)[Search Reclamation](#)[Great Plains Home](#)[About Us](#)[Area Offices](#)[Multimedia](#)[Newsroom](#)[Programs & Activities](#)[Reservoirs, Dams & Hydropower](#)[AgriMet](#)[AgriMet Station List](#)[About Crop Water Use Charts](#)[AgriMet Weather Parameters](#)[Other Crop Water Use & ET Programs in the GP Region](#)

AgriMet Data Collection Station, near Bozeman, Montana, part of the Great Plains Agricultural Data Collection System. For more information about AgriMet, see the About AgriMet page, accessible from the menu on the left.

Reclamation, Gallatin County Soil Conservation District, and Department of Natural Resources and Conservation.

The site is approximately 4 miles west of Bozeman, Montana. latitude: 45 40 25 longitude: 111 09 00 elevation-4775'

AgriMet: Weather & Crop Water Use Charts

Bozeman, Montana



- [AgriMet Home](#)
- [AgriMet Station List](#)
- [Daily Weather Information](#)
- [Archival Weather Information](#)
- [Crop Water Use Charts](#)
- [Crop Evapotranspiration Summaries](#)

The AgriMet station BOZM was installed October 17, 1989. The cooperating agencies involved with this station are the Bureau of

www.usbr.gov/gp/agrimet/station_bozm_bozeman.cfm

Google

Ubigraph Omegah... RUBigra... file:...xml DocBook XML.co... Recla...

Irrigation Guide
Pacific Northwest
Region AgriMet Home
AgriMet Partners &
Water Resource Info
outside GP Region
Boat Ramps
HydroMet
Project & Facilities
Recreation
Safety of Dams
Water Management Info
Water Operations
GP Multimedia
Podcasts RSS
Facebook
YouTube
Flickr
Twitter
Multimedia
Reclamation RSS

Dayfiles Information:

[<top of page>](#) [<station list>](#) [<dayfiles>](#) [<archival water info>](#) [<crop water use charts>](#) [<crop ET summaries>](#)

Enter date, time period, and parameter codes then submit your request
(please note, all fields must be completed, with times in a [24-hour format](#))

Station Code:

Enter date (YYYYMMDD no spaces, e.g., [96JAN01](#)):

Enter start time using [24-hour clock](#) (HH:MM, e.g., 05:15):

Enter end time using [24-hour clock](#) (HH:MM, e.g., 14:45):

(Note: To obtain all current data for the day for selected parameters, enter a start time of 00:00 and an end time of 23:45)

| | |
|---|-------------------------------------|
| BV = BATTERY VOLTAGE | TU = RELATIVE HUMIDITY (%) |
| OB = TEMPERATURE (deg F) | UI = ACCUM WIND RUN (miles) |
| PC = ACCUM PRECIPITATION (inches) | WD = WIND DIRECTION (deg az) |
| SQ = ACCUM SOLAR RADIATION (langley) | WG = PEAK WIND GUST (mph) |
| TP = DEW POINT TEMPERATURE (deg F) | WS = AVG WIND SPEED (mph) |

Enter up to six parameters from above list, separated by commas, but without spaces (e.g., OB,PC,SQ):

Select Menu

Text boxes

Submit Button

```

<span class="red12px">(please note, all fields must be completed, with
times in a <a href="javascript:;" onclick="MM_openBrWindow('../info/24_hour_clock.cfm','','scro
<form action="http://www.usbr.gov/gp-bin/agrimet_dayfiles.pl" method="post" name="dayfiles" t
<p class="agrimetinfoheadersm"><span class="subheader3blue10ptbold">Station Code:</span>&nb
<select name="station_code" size="1" id="station_code">
<option selected="selected">BOZM = Bozeman, MT </option>
<option>BOMT = Broken-O Ranch, MT </option>
<option>BFTM = Big Flat, Turner, MT </option>
<option>BRGM = Buffalo Rapids, Glendive, MT </option>
<option>BRTM = Buffalo Rapids, Terry, MT </option>
<option>BFAM = Blackfeet, MT </option>
<option>DLNM = Dillon, MT </option>
<option>GLGM = Glasgow, MT </option>
<option>GFMT = Greenfields, MT </option>
<option>HRLM = Harlem, MT </option>
<option>HVMT = Helena Valley, MT </option>
<option>JVWM = Jefferson Valley, MT </option>
<option>LMMM = Lower Musselshell, Melstone, MT </option>
<option>MATM = Malta, MT </option>
<option>MWSM = Moccasin, MT </option>
<option>RBYM = Ruby Valley, MT </option>
<option>SVWM = Shields River Valley, MT </option>
<option>TRFM = Teton River, MT </option>
<option>TOSM = Toston, MT </option>
<option>UMHM = Upper Musselshell, MT </option>
<option>WSSM = White Sulphur Springs, MT </option>
</select>
dEditable --> </p>
<p align="center" class="agrimetinfoheadersm"><span class="subheader3blue10ptbold">Enter da
<input name="date" size="7" ,type="text" , />
</p>
<p align="center" class="agrimetinfoheadersm"><span class="subheader3blue10ptbold">Enter st
<input name="start_time" size="5" ,type="text" , />
</p>
<p align="center" class="agrimetinfoheadersm"><span class="subheader3blue10ptbold">Enter er
<input name="end_time" ,type="text" , size="5" />
</p>
<p align="center" class="red12px">(Note: To obtain all current data for the day for selecte

```

HTML Form

```
<form action="url" method="post">  
  <select name="station_code">  
    <option>Bozeman</option>  
    <option>Teton River</option>  
  </select>  
  <input name="date" size="7" type="text"/>  
  <input name="submit2" type="submit"  
  class="search" value=" Submit " />  
</form>
```

HTTP - facilities

```
<form action="http://www.url" method="post">
```

If the method is "post" and the action is an HTTP URI, the user agent conducts an HTTP "post" transaction using the URL and a message created according to the content type specified in the body of the request.

We can examine the form in R

```
> formURL = "http://www.usbr.gov/gp/agrimet/
station_bozm_bozeman.cfm"
> doc = htmlParse(formURL)
> forms = getNodeSet(doc, "//form")
> length(forms)
[1] 5
xmlAttrs(forms[[2]])
```

We can examine the form in R

```
> xmlAttrs(forms[[2]])
```

```
action
```

```
"http://www.usbr.gov/gp-bin/agrimet_dayfiles.pl"
```

| method | name | target | id |
|--------|------|--------|----|
|--------|------|--------|----|

| | | | |
|--------|------------|----------|------------|
| "post" | "dayfiles" | "_blank" | "dayfiles" |
|--------|------------|----------|------------|

Construct a call to postForm

```
> args = list(  
    station_code = "BOZM",  
    date = "11NOV27",  
    start_time = "00:00", end_time = "23:45",  
    parameters = "OB,TP")  
  
> url = "http://www.usbr.gov/gp-bin/agrimet_dayfiles.pl"  
  
> ans = postForm(url, .params = args,  
    style = "POST")
```

Examine the results

```
> htmlTree = htmlParse(ans, asText = TRUE)
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<html>
<head><title>Agrimet Hourly Data for BOZM</title></head>
<body><pre>
BOZEMAN |11NOV27| AIRTEMP | DEWPT |
| 00:00 | 29.29 | 17.86 |
| 00:15 | 29.12 | 16.72 |
| 00:30 | 29.79 | 16.47 |
| 00:45 | 29.92 | 16.57 |
| 01:00 | 30.16 | 14.13 |
| 01:15 | 35.71 | 16.17 | .....
```

Extract the
data from
within the
<pre>

```
> weatherData = xpathSApply(htmlTree,  
                             "//pre", xmlValue)
```

The rest is regular expressions!

REST

Representational State Transfer



Homes Rentals Mortgage Rates Advice Professionals Local Info Blog more

Find

Homes For Sale
Homes For Sale By Owner
Foreclosures
Open Houses
New Homes
Recent Home Sales

Post a Home

For Sale by Owner
For Sale by Agent
Make Me Move

Neighborhood or City or ZIP



For Sale
Homes



Take Zillow With You
Download Zillow Mobile Apps

Zillow

- Zillow is a home and real estate marketplace with information about homes, real estate and mortgages.
- It is a “living database” of more than 100 million U.S. homes - including homes for sale and rent.
- Borrowers can connect with lenders to find loans and get the best mortgage rates

illow

Neighborhood or City or ZIP Code or Address

GO

ew

[ook Apps](#)[ts for Your Site](#)[s for Your Site](#)[or Your Site](#)[enefits](#)[artners](#)[ing Requirements](#)[Valuation API](#)[ighborhood Data](#)[age API](#)[erty Details API](#)[ighborhood Boundaries](#)[nology Providers](#)[s Of Use](#)[ress Plugins](#)

Real Estate and Mortgage Data for Your Site

Turn Your Site Into a Real Estate Portal With Zillow

The new Zillow API Network turns member sites into mini real estate portals by offering fresh and provocative real estate content to keep people coming back.

Home Valuation

Search results list, Zestimate®, Rent Zestimate®, home valuations, home valuation charts, comparable houses, and market trend charts.

API calls of interest:

1. [GetZestimate](#)
2. [GetSearchResults](#)

REAL ESTATE
+
TECHNOLOGY
=
GEEK ESTATE

Q gcc

Next

Previous

☐ Highlight all☐ Match case

Phrase not found

Zillow API

- Zillow offers a portal to connect to its data base programmatically
- Get Estimates: For a specified property you can get the most recent Zillow estimate of the property value, a valuation range, and 30 day change.
- Get Comparables: For a specified property you can get the Zillow estimates for comparable properties



Zillow API Network

GetSearchResults API

The GetSearchResults API finds a property for a specified address. The content returned contains the address for the property or properties as well as the Zillow Property ID (ZPID) and current Zestimate®. It also includes the date the Zestimate was computed, a valuation range and the Zestimate ranking for the property within its ZIP code.

The GetSearchResults API Web Service is located at: <http://www.zillow.com/web ... sults.htm>

The parameters of the API are:

| Parameter | Description | Required |
|---------------|---|----------|
| zws-id | The Zillow Web Service Identifier. Each subscriber to Zillow Web Services is uniquely identified by an ID sequence and every request to Web services requires this ID. Click here to get yours. | Yes |
| address | The address of the property to search. This string should be URL encoded. | Yes |
| citystatezip | The city+state combination and/or ZIP code for which to search. This string should be URL encoded. Note that giving both city and state is required. Using just one will not work. | Yes |
| rentzestimate | Return Rent Zestimate information if available (boolean true/false, default: false) | No |

GetSearchResults API

- URL

<http://www.zillow.com/webservice/GetSearchResults.htm>

- Parameters:

- zws-id: required identifier of the subscriber
- address: required property address,
e.g. 1280 Monterey Avenue
- citystatezip: either the city+state or the zip code
- rentzestimate: optional boolean to request rental
information

GET request

> zid = "X1-ZWz1c...."

> address = "1280 Monterey Avenue"

> zip = "94707"

> zillowURL = "http://www.zillow.com/webservice/
GetSearchResults.htm"

> reply = **getForm**(zillowURL,
 'zws-id'= zid, address = address,
 citystatezip = zip)

Return

```
<?xml version="1.0" encoding="utf-8"?>
<SearchResults:searchresults .... >
  <request>
    <address>1280 Monterey Avenue</address>
    <citystatezip>94707</citystatezip>
  </request>
<message>
  <text>Request successfully processed</text>
  <code>0</code>
</message>
```

Return

```
<response>
```

```
  <results>
```

```
    <result>
```

```
      <zpid>24842790</zpid>
```

```
....
```

```
  <zestimate>
```

```
    <amount currency="USD">663900</amount>
```

```
    <last-updated>04/04/2012</last-updated>
```

```
    <oneWeekChange deprecated="true"/>
```

```
    <valueChange duration="30" currency="USD">6000</valueChange>
```

```
    <valuationRange>
```

```
      <low currency="USD">199170</low>
```

```
      <high currency="USD">843153</high>
```

```
    </valuationRange>
```

```
    <percentile>0</percentile>
```

```
  </zestimate>
```

The Zillow Property
ID can be used to
get comparables

Extract the data from the return

```
> doc = xmlParse(reply, asText = TRUE)
> amount =
as.numeric(xmlValue(est[["amount"]]))
> amount
[1] 663900
```

Technologies for Accessing Data on the Web

- HTML page
- HTML Forms: GET and POST
- REST
- SOAP – a rich (XML-based technology)
- XML-RPC – an older (XML-based) technology