HyperText Transfer Protocol

HTTP

HTTP

- Protocol that web browsers and web servers use to communicate with each other
- It is a *request-response* protocol between a client and a server

Example: Browsing the Web

- Client machine (your computer) is the one to initiate a request for a web page
- The request is sent to the Web server (hosts Web site(s) and provides HTML files)
- The server receives the request and sends back the desired information along with a status report

HTTP Request Pieces: *First Line*

- First line contains:
  - a method, e.g., GET or POST (ones we will cover)
  - a URL or path to the document
  - The protocol and its version

For example:

```
GET /RCurl/index.html HTTP/1.1
```

HTTP Request Pieces: *Header*

- Provides auxiliary information about the request via key:value pairs, e.g.

```
User-Agent: R version 2.15.0 (2012-03-30)
Host: www.omegahat.org
Accept: */*
Authorization: xxx
From: login@mail.com
X-Do-Not-Track: 1
Blank line
```

Blank Line Indicates end of header

HTTP Request Pieces: *Body*

- Optional body for the request
- Contains the data characterizing the request
- Used for a POST method of request, not for GET

HTTP Response Pieces

- First Line
  - Status of the request
- Remaining Header
  - key:value pairs about the content of the response, e.g., character encoding, type of content (image, HTML document, …).
- Body: typically what we want, e.g., the HTML document.

GET Method

- Request a document from a server
- Request has no side effect.

Example, type a URL in a browser

`http://www. omegahat.org/RCurl/index.html`

Produces the following HTTP request:

```
GET /RCurl/index.html HTTP/1.1
Host: www.omegahat.org
Blank line
```

Server's Response: Header

HTTP/1.1 200 OK

Status 200 indicates success

Date: Fri, 01 Jun 2012 22:56:46 GMT
Server: Apache/2.2.14 (Ubuntu)
Last-Modified: Wed, 01 Feb 2012 04:08:30 GMT
ETag: "3262089-10bf-4b7df3b75ab80"
Accept-Ranges: bytes
Content-Length: 4287
Vary: Accept-Encoding
Content-Type: text/html
Blank line

Status Codes

- 100 Informational – Communication continuing, more input expected from client or server
- 200 Success - e.g., 200 - general success;
- 300 Redirection or Conditional Action – requested URL is located somewhere else.
- 400 Client Error – e.g., 404 indicates the document was not found
- 500 Internal Server Error or Broken Request – error on the server side

Visit Wikipedia Page

```
pageContents = getURLContent(wikiURL,
                             verbose = TRUE)

*   Trying 2620:0:863:ed1a::1...
* Connected to en.wikipedia.org (2620:0:863:ed1a::1)
port 443 (#0)
* TLS 1.2 connection using
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
* Server certificate: *.wikipedia.org
* Server certificate: GlobalSign Organization
Validation CA - SHA256 - G2
* Server certificate: GlobalSign
> GET
/wiki/United_States_presidential_election_in_Virgini
a,_2004 HTTP/1.1
Host: en.wikipedia.org
Accept: */*
```

Response Header

```
< HTTP/1.1 200 OK
< Date: Mon, 28 Nov 2016 18:08:55 GMT
< Content-Type: text/html; charset=UTF-8
< Content-Length: 164960
< Connection: keep-alive
< Server: mw1272.eqiad.wmnet
< X-Powered-By: HHVM/3.12.7
< Vary: Accept-Encoding,Cookie,Authorization
< X-UA-Compatible: IE=Edge
< Content-language: en
…
```

Example: Other Status Codes

India stock exchange

```
uIn =
"http://www.nseindia.com/archives/nsccl/volt/CMVOLT_070
62012.CSV"
data = read.csv(url(uIn))
```

Error in open.connection(file, "rt") : cannot open the connection
In addition: Warning message:
In open.connection(file, "rt") :
  cannot open URL
'http://www.nseindia.com/archives/nsccl/volt/CMVOLT_07062012.CSV':
HTTP status was '403 Forbidden'

Try Verbose option of getURL

```
d = getURL(uIn, verbose = TRUE)
*   Trying 23.7.66.103...
* Connected to www.nseindia.com (127.0.0.1) port 80 (#0)
> GET /archives/nsccl/volt/CMVOLT_07062012.CSV HTTP/1.1
 Host: www.nseindia.com
 Accept: */*

< HTTP/1.1 403 Forbidden
< Server: AkamaiGHost
< Mime-Version: 1.0
< Content-Type: text/html
< Content-Length: 325
< Expires: Mon, 28 Nov 2016 16:31:32 GMT
< Date: Mon, 28 Nov 2016 16:31:32 GMT
< Connection: close
<
* Closing connection 0
Error: Forbidden
```

Need Some Expertise in HTTP status codes – We need to supply a User-Agent key-value pair

What is a 403 error?

- 403 status code indicates that the server understood the request but refuses to authorize it.
- Curious: We can paste the url in the browser and download the data. No special login or permission is needed to do this
- The server can make public the reason for forbidding the request

Contents of the Response

file:///Users/nolan/indiaNSE403.html

Access Denied

You don't have permission to access "http://www.nseindia.com/archives/nsccl/volt/CMVOLT_07062012.CSV" on this server.

Reference #18.15e8d93f.1480375498.16d0b481

What are the terms of use?

https://www.nseindia.com/global/content/termsofuse.htm

You may not conduct any systematic or automated data collection activities (including scraping, data mining, data extraction and data harvesting) on or in relation to our site without our express written consent.

Possible Work Around

- Set a browser-like user agent string so the site thinks your request is from a browser
- Specify

User-Agent: string

Try Again with User-Agent

```
d = getURL(uIn, useragent = "R", verbose = TRUE)
*   Trying 23.7.66.103...
* Connected to www.nseindia.com (127.0.0.1) port 80 (#0)
> GET /archives/nsccl/volt/CMVOLT_07062012.CSV HTTP/1.1
Host: www.nseindia.com
User-Agent: R
Accept: */*

< HTTP/1.1 302 Moved Temporarily
< Server: AkamaiGHost
< Content-Length: 0
< Location:
https://www.nseindia.com/archives/nsccl/volt/CMVOLT_07062012.CSV
< Date: Mon, 28 Nov 2016 16:33:23 GMT
< Connection: keep-alive
<
* Connection #0 to host www.nseindia.com left intact
```

New Status Code indicates the resource has been moved.

```
d = getURL(uIn, useragent = "R”,
           followlocation = TRUE, verbose = TRUE)
```

```
*   Trying 23.192.90.148...
* Connected to www.nseindia.com (127.0.0.1) port 80 (#0)
> GET /archives/nsccl/volt/CMVOLT_07062012.CSV HTTP/1.1
Host: www.nseindia.com
User-Agent: R
Accept: */*


< HTTP/1.1 302 Moved Temporarily
< Server: AkamaiGHost
< Content-Length: 0 …
* Found bundle for host www.nseindia.com: 0x10ac74540
*   Trying 23.192.90.148...
> GET /archives/nsccl/volt/CMVOLT_07062012.CSV HTTP/1.1
Host: www.nseindia.com
User-Agent: R
Accept: */*

< HTTP/1.1 200 OK
< Server: -
< Content-Length: 96219
```

It’s a good idea to always set the follow location option to TRUE

Now we have the text

```
head(read.csv(textConnection(d)))
         Date     Symbol Underlying.Close.Price..A. …
1 07-JUN-2012  20MICRONS                      87.30
2 07-JUN-2012 3IINFOTECH                      11.45
3 07-JUN-2012    3MINDIA                    3664.20
4 07-JUN-2012     A2ZMES                     103.30
5 07-JUN-2012  AANJANEYA                     489.45
6 07-JUN-2012 AARTIDRUGS                     102.60
```

A GET Example with a Form

One of the “simplest” forms

View the Source (pared down)

```
<form action="/search" id="f" method="get">
  <div class="init" id="fkbx">
    <div id="fkbx-text">
      Search Google or type URL
    </div>
    <input id="q" aria-hidden="true"
       autocomplete="off" name="q"
       tabindex="-1" type="url"
       jsaction="mousedown:ntp.fkbxclk">
  </div>
</form>
```

17 only truly random number

About 109,000,000 results (0.29 seconds)

Is 17 the "most random" number? – Cognitive Daily - ScienceBlogs
scienceblogs.com/cognitivedaily/2007/02/05/is-**17**-the-most-**random-number**/
Feb 5, 2007 - Perhaps in a **truly random** sample, we'd see a similar distribution. ... the **number** 19 was most common, but it was chosen **just** 8 percent of the ...

Are people capable of generating a random number?
philosophy.stackexchange.com/.../are-people-capable-of-generating-a-**random-numb**...
Dec 23, 2011 - But the OP is **just** asking about a **random number**. ... Empirically, humans can't choose **truly random numbers**. .... mfloren Oct 26 at **17**:38 ...

I need a true random number generator web service - Software ...
https://softwareengineering.stackexchange.com/.../i...**true-random-number**.../100826
... a **true random number** for good simulation. A pseudo-**random number** is **just** "not good" enough for a "good" simulation you see. ... The bigger question is "why" do you need a "**truly**" **random number** generator? – Darknight May **17** '11 at 9:46 ...

Random number generation - Wikipedia
https://en.wikipedia.org/wiki/**Random_number**_generation

URL for the Google Search Results

The URL contains the inputs from the form

? Separates url from the inputs

https://www.google.com/search?q=17%20only%20truly%20random%20number&sourceid=chrome-instant&ion=1&ie=UTF-8&rct=j

q = our query
With %20 instead of blanks

& separates inputs

We can perform Google Searches programatically

```
queryURL =
"https://www.google.com/search?q=17+only+truly+random+number&sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8”
get17Query = getURL(queryURL)
```

OR

```
get17Info =
  getForm("https://www.google.com/search",
        q = "17+only+truly+random+number",
        sourceid = "chrome-instant",
        ion = "1", espv = "2",ie = "UTF-8")
```

An Example: GET and PUT

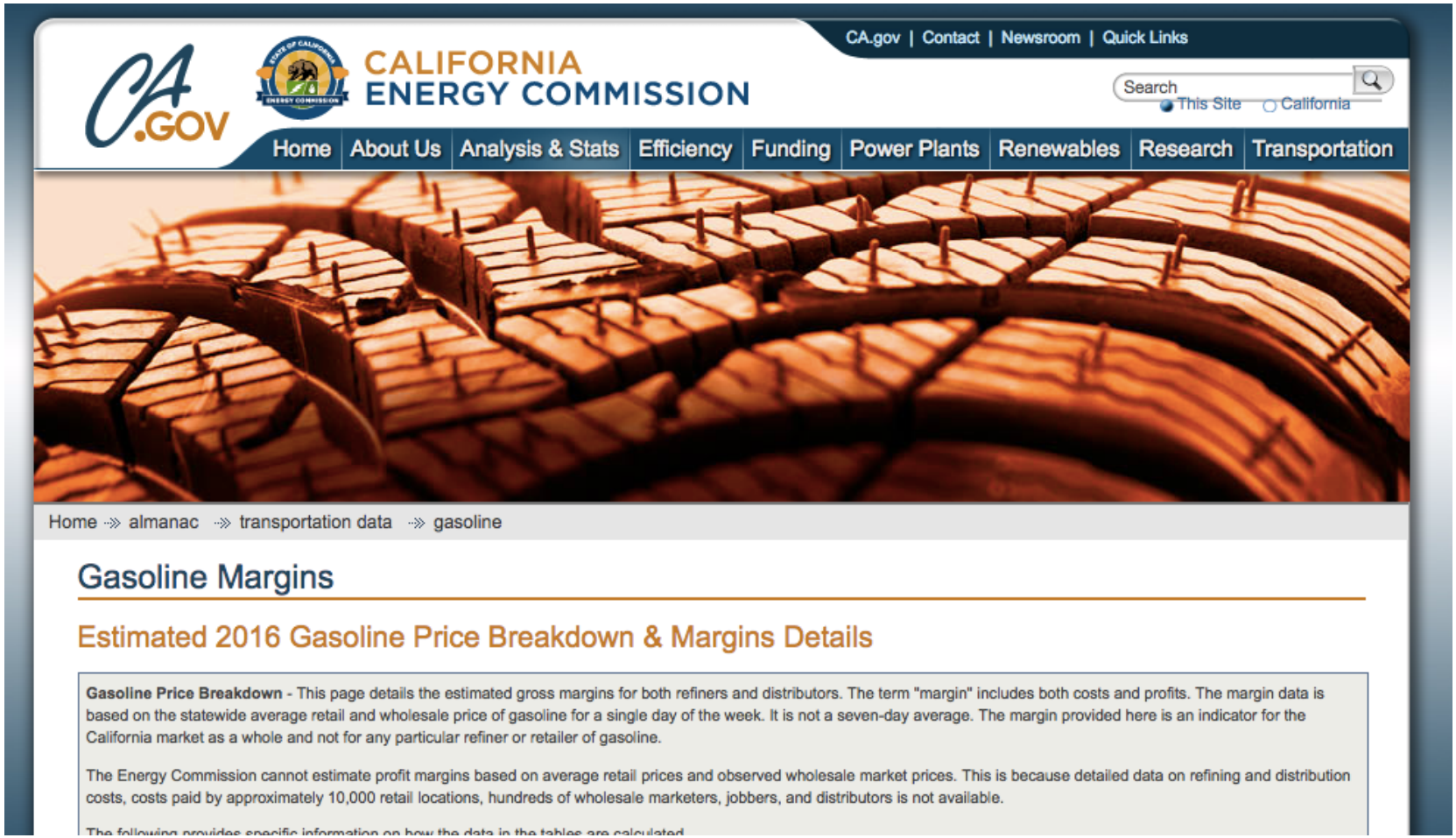
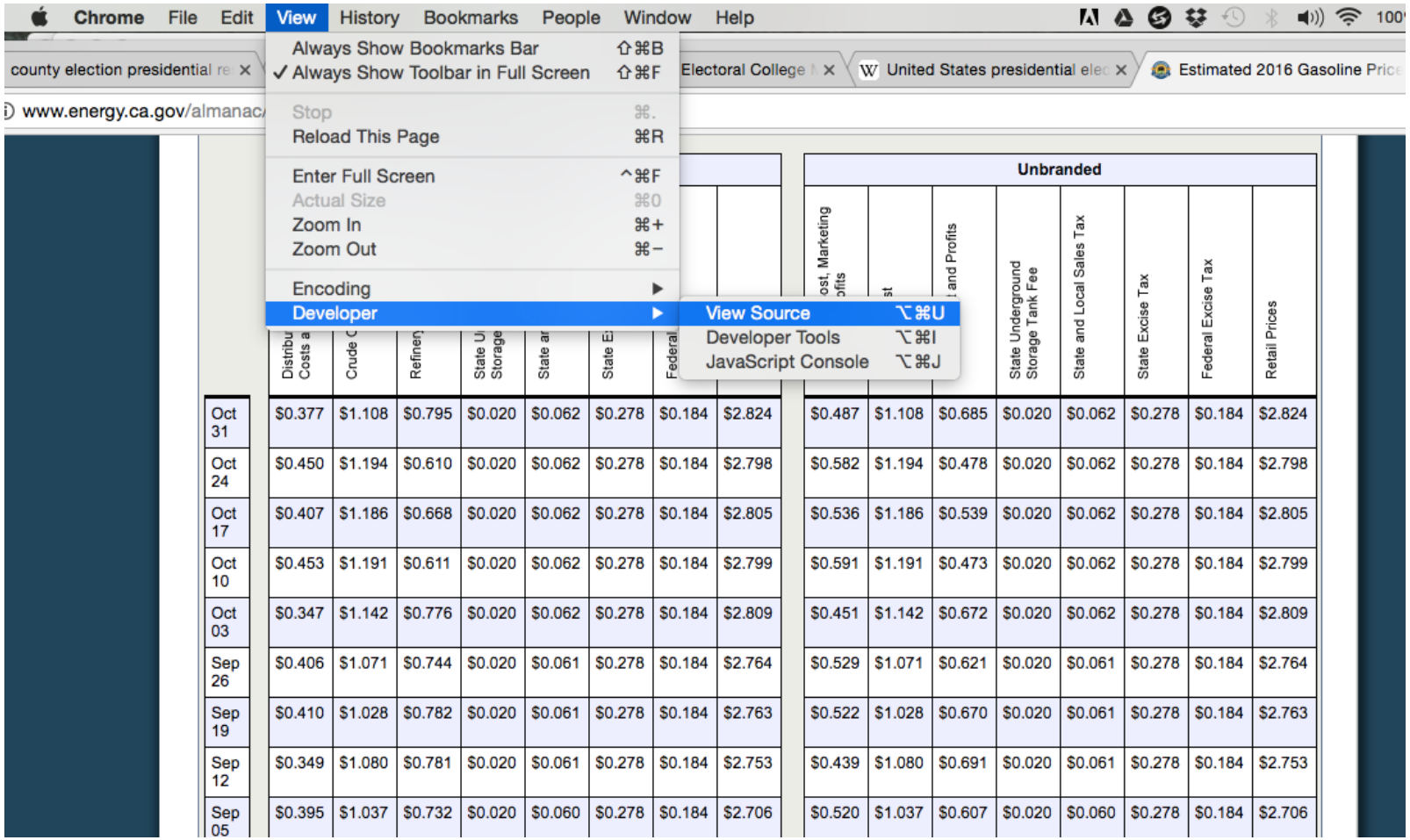


Table of Gas Prices

	Branded								Unbranded							
	Distribution Cost, Marketing Costs and Profits	Crude Oil Cost	Refinery Cost and Profits	State Underground Storage Tank Fee	State and Local Sales Tax	State Excise Tax	Federal Excise Tax	Retail Prices	Distribution Cost, Marketing Costs and Profits	Crude Oil Cost	Refinery Cost and Profits	State Underground Storage Tank Fee	State and Local Sales Tax	State Excise Tax	Federal Excise Tax	Retail Prices
Oct 31	$0.377	$1.108	$0.795	$0.020	$0.062	$0.278	$0.184	$2.824	$0.487	$1.108	$0.685	$0.020	$0.062	$0.278	$0.184	$2.824
Oct 24	$0.450	$1.194	$0.610	$0.020	$0.062	$0.278	$0.184	$2.798	$0.582	$1.194	$0.478	$0.020	$0.062	$0.278	$0.184	$2.798
Oct 17	$0.407	$1.186	$0.668	$0.020	$0.062	$0.278	$0.184	$2.805	$0.536	$1.186	$0.539	$0.020	$0.062	$0.278	$0.184	$2.805
Oct 10	$0.453	$1.191	$0.611	$0.020	$0.062	$0.278	$0.184	$2.799	$0.591	$1.191	$0.473	$0.020	$0.062	$0.278	$0.184	$2.799
Oct 03	$0.347	$1.142	$0.776	$0.020	$0.062	$0.278	$0.184	$2.809	$0.451	$1.142	$0.672	$0.020	$0.062	$0.278	$0.184	$2.809
Sep 26	$0.406	$1.071	$0.744	$0.020	$0.061	$0.278	$0.184	$2.764	$0.529	$1.071	$0.621	$0.020	$0.061	$0.278	$0.184	$2.764
Sep 19	$0.410	$1.028	$0.782	$0.020	$0.061	$0.278	$0.184	$2.763	$0.522	$1.028	$0.670	$0.020	$0.061	$0.278	$0.184	$2.763
Sep 12	$0.349	$1.080	$0.781	$0.020	$0.061	$0.278	$0.184	$2.753	$0.439	$1.080	$0.691	$0.020	$0.061	$0.278	$0.184	$2.753
Sep 05	$0.395	$1.037	$0.732	$0.020	$0.060	$0.278	$0.184	$2.706	$0.520	$1.037	$0.607	$0.020	$0.060	$0.278	$0.184	$2.706
Aug 29	$0.391	$1.099	$0.677	$0.020	$0.060	$0.278	$0.184	$2.709	$0.517	$1.099	$0.551	$0.020	$0.060	$0.278	$0.184	$2.709

View the HTML Source



Oct 31	$0.377	$1.108	$0.795	$0.020	$0.062	$0.278	$0.184	$2.824	$0.487	$1.108	$0.685	$0.020	$0.062	$0.278	$0.184	$2.824
Oct 24	$0.450	$1.194	$0.610	$0.020	$0.062	$0.278	$0.184	$2.798	$0.582	$1.194	$0.478	$0.020	$0.062	$0.278	$0.184	$2.798
Oct 17	$0.407	$1.186	$0.668	$0.020	$0.062	$0.278	$0.184	$2.805	$0.536	$1.186	$0.539	$0.020	$0.062	$0.278	$0.184	$2.805
Oct 10	$0.453	$1.191	$0.611	$0.020	$0.062	$0.278	$0.184	$2.799	$0.591	$1.191	$0.473	$0.020	$0.062	$0.278	$0.184	$2.799
Oct 03	$0.347	$1.142	$0.776	$0.020	$0.062	$0.278	$0.184	$2.809	$0.451	$1.142	$0.672	$0.020	$0.062	$0.278	$0.184	$2.809
Sep 26	$0.406	$1.071	$0.744	$0.020	$0.061	$0.278	$0.184	$2.764	$0.529	$1.071	$0.621	$0.020	$0.061	$0.278	$0.184	$2.764
Sep 19	$0.410	$1.028	$0.782	$0.020	$0.061	$0.278	$0.184	$2.763	$0.522	$1.028	$0.670	$0.020	$0.061	$0.278	$0.184	$2.763
Sep 12	$0.349	$1.080	$0.781	$0.020	$0.061	$0.278	$0.184	$2.753	$0.439	$1.080	$0.691	$0.020	$0.061	$0.278	$0.184	$2.753
Sep 05	$0.395	$1.037	$0.732	$0.020	$0.060	$0.278	$0.184	$2.706	$0.520	$1.037	$0.607	$0.020	$0.060	$0.278	$0.184	$2.706

```
613 <tr>
614 <td style='background: #eeeeff;' class='ltr2'>Oct 31</td>
615 <td style='background-color:#ecece7;'> </td>
616 <td style='background: #eeeeff;' class='tl2'>$0.377</td>
617 <td style='background: #eeeeff;' class='tl2'>$1.108</td>
618 <td style='background: #eeeeff;' class='tl2'>$0.795</td>
619 <td style='background: #eeeeff;' class='tl2'>$0.020</td>
620 <td style='background: #eeeeff;' class='tl2'>$0.062</td>
621 <td style='background: #eeeeff;' class='tl2'>$0.278</td>
622 <td style='background: #eeeeff;' class='tl2'>$0.184</td>
623 <td style='background: #eeeeff;' class='ltr2'>$2.824</td>
624 <td style='background-color:#ecece7;'> </td>
625 <td style='background: #eeeeff;' class='tl2'>$0.487</td>
626 <td style='background: #eeeeff;' class='tl2'>$1.108</td>
627 <td style='background: #eeeeff;' class='tl2'>$0.685</td>
628 <td style='background: #eeeeff;' class='tl2'>$0.020</td>
629 <td style='background: #eeeeff;' class='tl2'>$0.062</td>
630 <td style='background: #eeeeff;' class='tl2'>$0.278</td>
631 <td style='background: #eeeeff;' class='tl2'>$0.184</td>
632 <td style='background: #eeeeff;' class='ltr2'>$2.824</td>
633 </tr><tr>
```

```
> gasURL =
"http://www.energy.ca.gov/almanac/transp
ortation_data/gasoline/margins/"

> tbl =
  readHTMLTable(gasURL, which = 1,
                stringsAsFactors = FALSE)

> dim(tbl)
[1] 47 19
```

Want Data for Additional Years

Jan 04	$0.315	$0.856	$1.137	$0.020	$0.063	$0.3

Select Year Get different year

Definitions

Wholesale Gasoline Price: The average wholesale gasoline price is t
state. This average price is for a single day. The wholesale gasoline pr

Branded and Unbranded Gasoline: Branded gasoline refers to fuel th
proprietary fuel additives. Unbranded gasoline is not associated with a

View Source

POST method

```
<form action='index.php' method='post'>
<label for='year'><select name='year' id='year'>
<option value='2016'>Select Year</option>
<option value='2015'>2015</option>
<option value='2014'>2014</option>
<option value='2013'>2013</option>
<option value='2012'>2012</option>
<option value='2011'>2011</option>
<option value='2010'>2010</option>
<option value='2009'>2009</option>
<option value='2008'>2008</option>
<option value='2007'>2007</option>
<option value='2006'>2006</option>
<option value='2005'>2005</option>
<option value='2004'>2004</option>
<option value='2003'>2003</option>
<option value='2002'>2002</option>
<option value='2001'>2001</option>
<option value='2000'>2000</option>
<option value='1999'>1999</option>

</select></label>
<input name='newYear' type='submit' value='Get different year' />
</form>
```

We have a <select> widget
And an <input> widget which is a Submit button

POST Method

- Requests the server to accept the entity enclosed in the body of the request
- For example, the information in a web form to a data handling process

View Source

year is the name of this input

2013 is its value

newYear is the name of this input

'Get different year' is its value

```
1498
1499 <form action='index.php' method='post'>
1500 <label for='year'><select name='year' id='year'>
1501 <option value='2016'>Select Year</option>
1502 <option value='2015'>2015</option>
1503 <option value='2014'>2014</option>
1504 <option value='2013'>2013</option>
1505 <option value='2012'>2012</option>
1506 <option value='2011'>2011</option>
1507 <option value='2010'>2010</option>
1508 <option value='2009'>2009</option>
1509 <option value='2008'>2008</option>
1510 <option value='2007'>2007</option>
1511 <option value='2006'>2006</option>
1512 <option value='2005'>2005</option>
1513 <option value='2004'>2004</option>
1514 <option value='2003'>2003</option>
1515 <option value='2002'>2002</option>
1516 <option value='2001'>2001</option>
1517 <option value='2000'>2000</option>
1518 <option value='1999'>1999</option>
1519
1520 </select></label>
1521 <input name='newYear' type='submit' value='Get different year' />
1522 </form>
```

```
txt = postForm(gasURL,
               year = "2013",
               newYear = 'Get different year')

gas13 = readHTMLTable(txt, which = 1,
                      stringsAsFactors = FALSE)
```

Authentication

OAuth

- Authentication protocol that allows
  - you (User)
  - to approve one application (consumer)
  - to interact with another app (service provider) on your behalf
- User's password to service provider is not shared with the consumer application

Scenario

- User: Joe
- Consumer: Bitly
- Service provider: Twitter

Joe wants to allow bitly to post shortened links to his Twitter stream

See https://blog.varonis.com/introduction-to-oauth/

Series of Exchanges

- *User shows intent* (Joe asks bitly to do it)
- *Consumer gets permission from provider* (bitly contacts twitter and gets consumer key & secret)
- *User goes to Provider and approves consumer access* (shows consumer key and user password)
- *Consumer obtains access from provider* (twitter gives bitly access token and secret (bitly provides consumer secret to get it))
- Consumer can now access User's stuff from service provider (with access token)

Pulling tweets for analysis in R

Who is the User?

A. my twitter acct
B. twitteR function
C. me
D. Twitter

Twitter scraping from within R

- We have a User account with Twitter
- Tell Twitter we are setting up an app. We get its consumer key and secret.
- We verify the app with twitter and get an access token and secret
- R package twitteR – has a function searchTwitter() which we use as the consumer and as the user

Pull tweets

```
library(twitteR)

consumer_key = "7P5a…….uSR"
consumer_secret = "Uyc….Q0c"
access_token = "3684….d6x"
access_secret = "7LpQ….T3e”

setup_twitter_oauth(consumer_key,
                    consumer_secret,
                    access_token,
                    access_secret)
trumpTwts = userTimeline("realDonaldTrump”)
```

Before you scrape:

- Check to see if CSV, JSON, or XML version of an HTML page are available – better to use those
- Check to see if there is an R package that provides structured access (e.g., twitteR)
- Check that you have permission to scrape

If you do scrape:

- Be careful to not to overburden the site with your requests
- Test code on small requests
- Use try() so that you don’t lose your results when one request triggers an error
- Save the results of each request so you don’t have to repeat the request unnecessarily