Web Scraping for Sports Data with R

Yaqiong Yao

10/1/2020

Outline

- Introduction
- Web Scraping Techniques Using R
  - Import files downloaded from websites
  - Static data
  - Dynamic data
- Summary

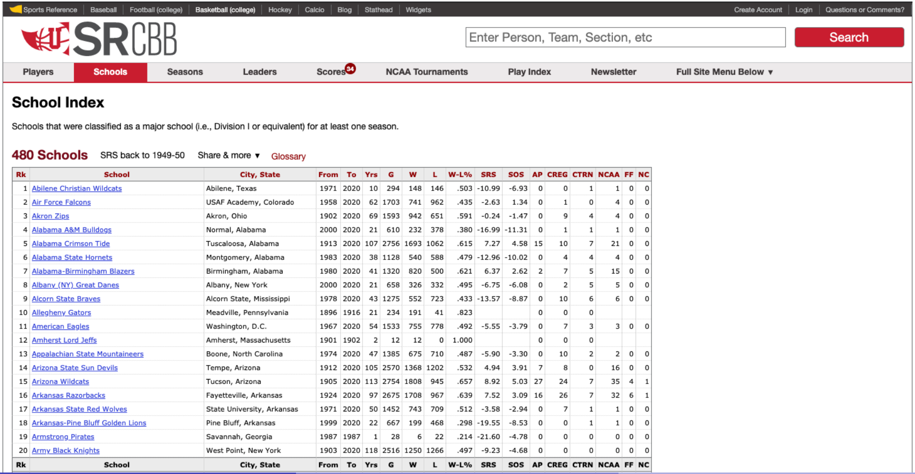
Introduction

- Web scraping technique is used for capturing data from websites.
- Motivation of Web Scraping
  - Need to extract data from websites
  - A reproducible way of capturing data online
- Prerequisite
  - Having experience with R
  - A laptop with R and R studio installed

Example

College basketball school index

- These data can be obtained by copying and pasting manully.
- Web scraping technique helps capture the data efficiently.



School Index

Schools that were classified as a major school (i.e., Division I or equivalent) for at least one season.

480 Schools SRS back to 1949-50 Share & more ▾ Glossary

Rk	School	City, State	From	To	Yrs	G	W	L	W-L%	SRS	SOS	AP	CREG	CTRN	NCAA	FF	NC
1	Abilene Christian Wildcats	Abilene, Texas	1971	2020	10	294	148	146	.503	-10.99	-6.93	0	0	1	1	0	0
2	Air Force Falcons	USAF Academy, Colorado	1958	2020	62	1703	741	962	.435	-2.63	1.34	0	1	0	4	0	0
3	Akron Zips	Akron, Ohio	1902	2020	69	1593	942	651	.591	-0.24	-1.47	0	9	4	4	0	0
4	Alabama A&M Bulldogs	Normal, Alabama	2000	2020	21	610	232	378	.380	-16.99	-11.31	0	1	1	1	0	0
5	Alabama Crimson Tide	Tuscaloosa, Alabama	1913	2020	107	2756	1693	1062	.615	7.27	4.58	15	10	7	21	0	0
6	Alabama State Hornets	Montgomery, Alabama	1983	2020	38	1128	540	588	.479	-12.96	-10.02	0	4	4	4	0	0
7	Alabama-Birmingham Blazers	Birmingham, Alabama	1980	2020	41	1320	820	500	.621	6.37	2.62	2	7	5	15	0	0
8	Albany (NY) Great Danes	Albany, New York	2000	2020	21	658	326	332	.495	-6.75	-6.08	0	2	5	5	0	0
9	Alcorn State Braves	Alcorn State, Mississippi	1978	2020	43	1275	552	723	.433	-13.57	-8.87	0	10	6	6	0	0
10	Allegheny Gators	Meadville, Pennsylvania	1896	1916	21	234	191	41	.823			0	0	0			
11	American Eagles	Washington, D.C.	1967	2020	54	1533	755	778	.492	-5.55	-3.79	0	7	3	3	0	0
12	Amherst Lord Jeffs	Amherst, Massachusetts	1901	1902	2	12	12	0	1.000			0	0	0			
13	Appalachian State Mountaineers	Boone, North Carolina	1974	2020	47	1385	675	710	.487	-5.90	-3.30	0	10	2	2	0	0
14	Arizona State Sun Devils	Tempe, Arizona	1912	2020	105	2570	1368	1202	.532	4.94	3.91	7	8	0	16	0	0
15	Arizona Wildcats	Tucson, Arizona	1905	2020	113	2754	1808	945	.657	8.92	5.03	27	24	7	35	4	1
16	Arkansas Razorbacks	Fayetteville, Arkansas	1924	2020	97	2675	1708	967	.639	7.52	3.09	16	26	7	32	6	1
17	Arkansas State Red Wolves	State University, Arkansas	1971	2020	50	1452	743	709	.512	-3.58	-2.94	0	7	1	1	0	0
18	Arkansas-Pine Bluff Golden Lions	Pine Bluff, Arkansas	1999	2020	22	667	199	468	.298	-19.55	-8.53	0	0	1	1	0	0
19	Armstrong Pirates	Savannah, Georgia	1987	1987	1	28	6	22	.214	-21.60	-4.78	0	0	0	0	0	0
20	Army Black Knights	West Point, New York	1903	2020	118	2516	1250	1266	.497	-9.23	-4.68	0	0	0	0	0	0
Rk	School	City, State	From	To	Yrs	G	W	L	W-L%	SRS	SOS	AP	CREG	CTRN	NCAA	FF	NC

Web Scraping Using R

- Different web scraping techniques are required when we are facing different kinds of data.
- Data have been organized into files.
  - Directly download it and read it in R
- Data are contained in HTML pages.
  - Static data
  - Dynamic data

Import Data Files from Websites

- These files that can be read by **read.csv** or related functions.
- They can be directly imported from a URL.
- Example: we extract the most recent Australian Open Tennis Championships match (AUS Open):

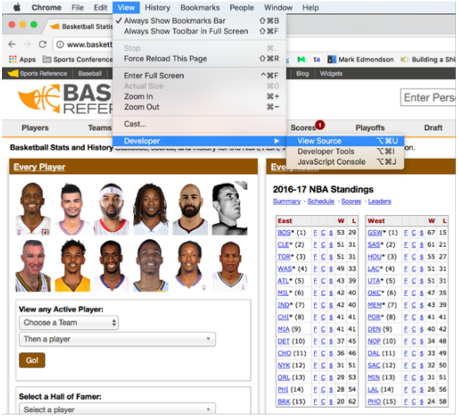
```r
url <- "http://www.tennis-data.co.uk/2020/ausopen.csv"
tennis_aus <- read.csv(url)
str(tennis_aus)
```

Static Data and Dynamic Data

- Most of data in the web are not organized into files, which can be directly imported into R.
- Before we capture these data, we need to determine whether the data are static or dynamic based on the source code.
- Static data is the data that can be seen in the source code.
- We cannot see the dynamic data in the source code.

Static Data and Dynamic Data

- The source code can be accessed by View → Developer → View Source in Chrome. Or right click the website and choose “View Page Source”.

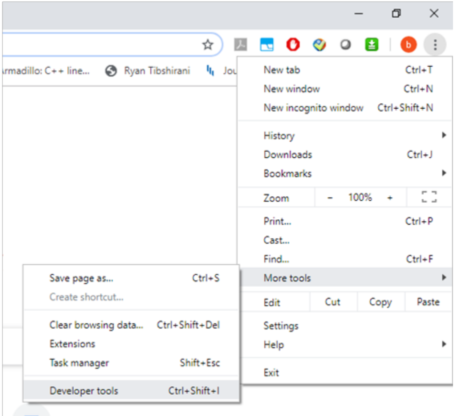


2016-17 NBA Standings

Summary · Schedule · Scores · Leaders

East				W	L
BOS* (1)	F	C	$	53	29
CLE* (2)	F	C	$	51	31
TOR* (3)	F	C	$	51	31
WAS* (4)	F	C	$	49	33
ATL* (5)	F	C	$	43	39
MIL* (6)	F	C	$	42	40
IND* (7)	F	C	$	42	40
CHI* (8)	F	C	$	41	41
MIA (9)	F	C	$	41	41
DET (10)	F	C	$	37	45
CHO (11)	F	C	$	36	46
NYK (12)	F	C	$	31	51
ORL (13)	F	C	$	29	53
PHI (14)	F	C	$	28	54
BRK (15)	F	C	$	20	62

West				W	L
GSW* (1)	F	C	$	67	15
SAS* (2)	F	C	$	61	21
HOU* (3)	F	C	$	55	27
LAC* (4)	F	C	$	51	31
UTA* (5)	F	C	$	51	31
OKC* (6)	F	C	$	47	35
MEM* (7)	F	C	$	43	39
POR* (8)	F	C	$	41	41
DEN (9)	F	C	$	40	42
NOP (10)	F	C	$	34	48
DAL (11)	F	C	$	33	49
SAC (12)	F	C	$	32	50
MIN (13)	F	C	$	31	51
LAL (14)	F	C	$	26	56
PHO (15)	F	C	$	24	58

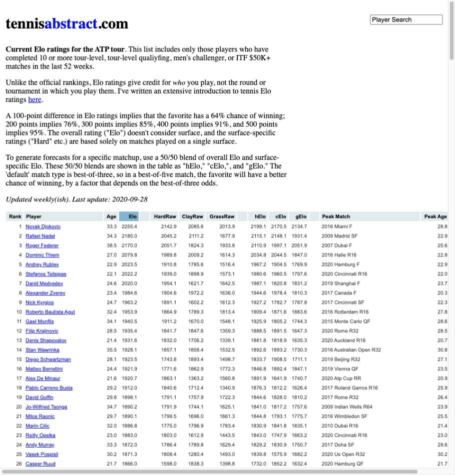


Static Data and Dynamic Data

Exerciese: Determine what kind of the data are in the following examples, static or dynamic.

- http://tennisabstract.com/reports/atp_elo_ratings.html
- https://www.flashscore.com/team/connecticut-huskies/8rqVf3Tj/results/

Static Data and Dynamic Data



tennisabstract.com

Player Search

Current Elo ratings for the ATP tour. This list includes only those players who have completed 10 or more tour-level, tour-level qualiyfing, men's challenger, or ITF $50K+ matches in the last 52 weeks.

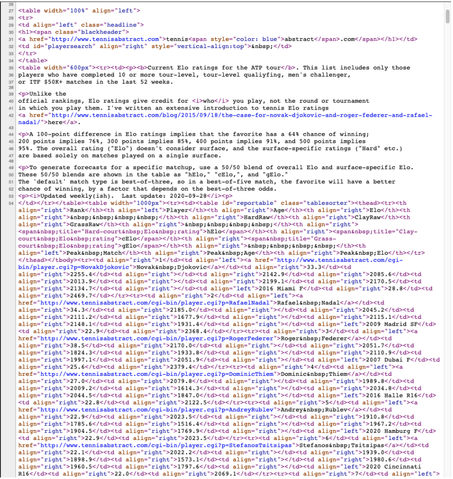
Unlike the official rankings, Elo ratings give credit for *who* you play, not the round or tournament in which you play them. I've written an extensive introduction to tennis Elo ratings here.

A 100-point difference in Elo ratings implies that the favorite has a 64% chance of winning; 200 points implies 76%, 300 points implies 85%, 400 points implies 91%, and 500 points implies 95%. The overall rating ("Elo") doesn't consider surface, and the surface-specific ratings ("Hard" etc.) are based solely on matches played on a single surface.

To generate forecasts for a specific matchup, use a 50/50 blend of overall Elo and surface-specific Elo. These 50/50 blends are shown in the table as "hElo," "cElo,", and "gElo." The 'default' match type is best-of-three, so in a best-of-five match, the favorite will have a better chance of winning, by a factor that depends on the best-of-three odds.

Updated weekly(ish). Last update: 2020-09-28

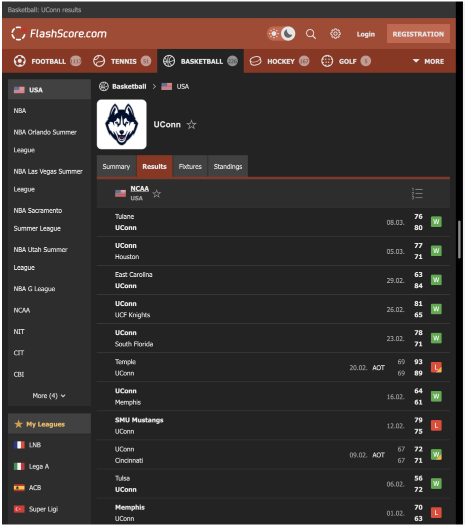
Rank	Player	Age	Elo		HardRaw	ClayRaw	GrassRaw		hElo	cElo	gElo		Peak Match	Peak Age
1	Novak Djokovic	33.3	2255.4		2142.9	2085.6	2013.9		2199.1	2170.5	2134.7		2016 Miami F	28.8
2	Rafael Nadal	34.3	2185.0		2045.2	2111.2	1677.9		2115.1	2148.1	1931.4		2009 Madrid SF	22.9
3	Roger Federer	38.5	2170.0		2051.7	1824.3	1933.8		2110.9	1997.1	2051.9		2007 Dubai F	25.6
4	Dominic Thiem	27.0	2079.8		1989.8	2009.2	1614.3		2034.8	2044.5	1847.0		2016 Halle R16	22.8
5	Andrey Rublev	22.9	2023.5		1910.8	1785.6	1516.4		1967.2	1904.5	1769.9		2020 Hamburg F	22.9
6	Stefanos Tsitsipas	22.1	2022.2		1939.0	1898.9	1573.1		1980.6	1960.5	1797.6		2020 Cincinnati R16	22.0
7	Daniil Medvedev	24.6	2020.0		1954.1	1621.7	1642.5		1987.1	1820.8	1831.2		2019 Shanghai F	23.7
8	Alexander Zverev	23.4	1984.6		1904.6	1972.2	1636.0		1944.6	1978.4	1810.3		2017 Canada F	20.3
9	Nick Kyrgios	24.7	1963.2		1891.1	1602.2	1612.3		1927.2	1782.7	1787.8		2017 Cincinnati SF	22.3
10	Roberto Bautista Agut	32.4	1953.9		1864.9	1789.3	1813.4		1909.4	1871.6	1883.6		2016 Rotterdam R16	27.8
11	Gael Monfils	34.1	1940.5		1911.2	1670.0	1548.1		1925.9	1805.2	1744.3		2015 Monte Carlo QF	28.6
12	Filip Krajinovic	28.5	1935.4		1841.7	1847.6	1359.3		1888.5	1891.5	1647.3		2020 Rome R32	28.5
13	Denis Shapovalov	21.4	1931.6		1832.0	1706.2	1339.1		1881.8	1818.9	1635.3		2020 Auckland R16	20.7
14	Stan Wawrinka	35.5	1928.1		1857.1	1858.4	1532.5		1892.6	1893.2	1730.3		2016 Australian Open R32	30.8
15	Diego Schwartzman	28.1	1923.5		1743.8	1893.4	1498.7		1833.7	1908.5	1711.1		2019 Beijing R32	27.1
16	Matteo Berrettini	24.4	1921.9		1771.6	1862.9	1772.3		1846.8	1892.4	1847.1		2019 Vienna QF	23.5
17	Alex De Minaur	21.6	1920.7		1863.1	1363.2	1560.8		1891.9	1641.9	1740.7		2020 Atp Cup RR	20.9
18	Pablo Carreno Busta	29.2	1912.0		1840.6	1712.4	1340.9		1876.3	1812.2	1626.4		2017 Roland Garros R16	25.9
19	David Goffin	29.8	1898.1		1791.1	1757.9	1722.3		1844.6	1828.0	1810.2		2017 Rome R32	26.4
20	Jo-Wilfried Tsonga	34.7	1890.2		1791.9	1744.1	1625.1		1841.0	1817.2	1757.6		2009 Indian Wells R64	23.9
21	Milos Raonic	29.7	1890.1		1799.5	1696.0	1661.3		1844.8	1793.1	1775.7		2016 Wimbledon SF	25.5
22	Marin Cilic	32.0	1886.8		1775.0	1796.9	1783.4		1830.9	1841.8	1835.1		2010 Dubai R16	21.4
23	Reilly Opelka	23.0	1883.0		1803.0	1612.9	1443.5		1843.0	1747.9	1663.2		2020 Cincinnati R16	23.0
24	Andy Murray	33.3	1872.0		1786.4	1789.8	1629.4		1829.2	1830.9	1750.7		2017 Doha SF	29.6
25	Vasek Pospisil	30.2	1871.3		1808.4	1280.4	1493.0		1839.8	1575.9	1682.2		2020 Us Open R32	30.2
26	Casper Ruud	21.7	1866.0		1598.0	1838.3	1398.8		1732.0	1852.2	1632.4		2020 Hamburg QF	21.7



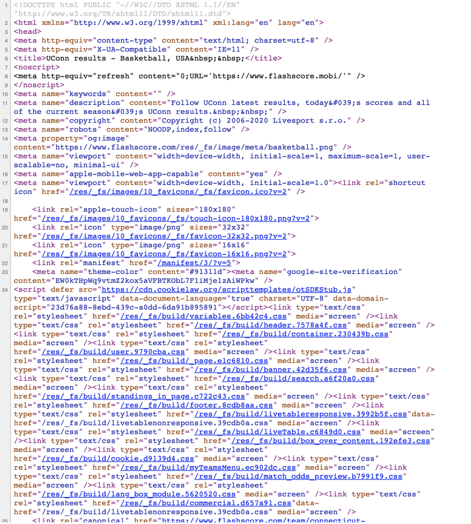
```
<table width="100%" align="left">
<tr>
<td align="left" class="headline">
<h1><span class="blackheader">
<a href="http://www.tennisabstract.com">tennis<span style="color: blue">abstract</span>.com</span></h1></td>
<td id="playersearch" align="right" style="vertical-align:top">&nbsp;</td>
</tr>
</table>
<table width="600px"><tr><td><p><b>Current Elo ratings for the ATP tour</b>. This list includes only those
players who have completed 10 or more tour-level, tour-level qualiyfing, men's challenger,
or ITF $50K+ matches in the last 52 weeks.

<p>Unlike the
official rankings, Elo ratings give credit for <i>who</i> you play, not the round or tournament
in which you play them. I've written an extensive introduction to tennis Elo ratings
<a href="http://www.tennisabstract.com/blog/2015/09/18/the-case-for-novak-djokovic-and-roger-federer-and-rafael-
nadal/">here</a>.

<p>A 100-point difference in Elo ratings implies that the favorite has a 64% chance of winning;
200 points implies 76%, 300 points implies 85%, 400 points implies 91%, and 500 points implies
95%. The overall rating ("Elo") doesn't consider surface, and the surface-specific ratings ("Hard" etc.)
are based solely on matches played on a single surface.

<p>To generate forecasts for a specific matchup, use a 50/50 blend of overall Elo and surface-specific Elo.
These 50/50 blends are shown in the table as "hElo," "cElo,", and "gElo."
The 'default' match type is best-of-three, so in a best-of-five match, the favorite will have a better
chance of winning, by a factor that depends on the best-of-three odds.
<p><i>Updated weekly(ish).  Last update: 2020-09-28</i><p>
</td></tr></table><table width="1000px"><tr><td><table id="reportable" class="tablesorter"><thead><tr><th
align="right">Rank</th><th align="left">Player</th><th align="right">Age</th><th align="right">Elo</th><th
align="right">&nbsp;&nbsp;&nbsp;&nbsp;</th><th align="right">HardRaw</th><th align="right">ClayRaw</th><th
align="right">GrassRaw</th><th align="right">&nbsp;&nbsp;&nbsp;&nbsp;</th><th align="right">
<span&nbsp;title="Hard-court&nbsp;Elo&nbsp;rating">hElo</span></th><th align="right"><span&nbsp;title="Clay-
court&nbsp;Elo&nbsp;rating">cElo</span></th><th align="right"><span&nbsp;title="Grass-
court&nbsp;Elo&nbsp;rating">gElo</span></th><th align="right">&nbsp;&nbsp;&nbsp;&nbsp;</th><th
align="left">Peak&nbsp;Match</th><th align="right">Peak&nbsp;Age</th><th align="right">Peak&nbsp;Elo</th></tr>
</thead></tbody><tr><td align="right">1</td><td align="left"><a href="http://www.tennisabstract.com/cgi-
bin/player.cgi?p=NovakDjokovic">Novak&nbsp;Djokovic</a></td><td align="right">33.3</td><td
align="right">2255.4</td><td align="right"></td><td align="right">2142.9</td><td align="right">2085.6</td><td
align="right">2013.9</td><td align="right"></td><td align="right">2199.1</td><td align="right">2170.5</td><td
align="right">2134.7</td><td align="right"></td><td align="left">2016 Miami F</td><td align="right">28.8</td><td
align="right">2469.7</td></tr><tr><td align="right">2</td><td align="left"><a
href="http://www.tennisabstract.com/cgi-bin/player.cgi?p=RafaelNadal">Rafael&nbsp;Nadal</a></td><td
align="right">34.3</td><td align="right">2185.0</td><td align="right"></td><td align="right">2045.2</td><td
align="right">2111.2</td><td align="right">1677.9</td><td align="right"></td><td align="right">2115.1</td><td
align="right">2148.1</td><td align="right">1931.4</td><td align="right"></td><td align="left">2009 Madrid SF</td>
<td align="right">22.9</td><td align="right">2368.4</td></tr><tr><td align="right">3</td><td align="left"><a
href="http://www.tennisabstract.com/cgi-bin/player.cgi?p=RogerFederer">Roger&nbsp;Federer</a></td><td
align="right">38.5</td><td align="right">2170.0</td><td align="right"></td><td align="right">2051.7</td><td
align="right">1824.3</td><td align="right">1933.8</td><td align="right"></td><td align="right">2110.9</td><td
align="right">1997.1</td><td align="right">2051.9</td><td align="right"></td><td align="left">2007 Dubai F</td><td
align="right">25.6</td><td align="right">2379.4</td></tr><tr><td align="right">4</td><td align="left"><a
href="http://www.tennisabstract.com/cgi-bin/player.cgi?p=DominicThiem">Dominic&nbsp;Thiem</a></td><td
align="right">27.0</td><td align="right">2079.8</td><td align="right"></td><td align="right">1989.8</td><td
align="right">2009.2</td><td align="right">1614.3</td><td align="right"></td><td align="right">2034.8</td><td
align="right">2044.5</td><td align="right">1847.0</td><td align="right"></td><td align="left">2016 Halle R16</td>
<td align="right">22.8</td><td align="right">2122.5</td></tr><tr><td align="right">5</td><td align="left"><a
href="http://www.tennisabstract.com/cgi-bin/player.cgi?p=AndreyRublev">Andrey&nbsp;Rublev</a></td><td
align="right">22.9</td><td align="right">2023.5</td><td align="right"></td><td align="right">1910.8</td><td
align="right">1785.6</td><td align="right">1516.4</td><td align="right"></td><td align="right">1967.2</td><td
align="right">1904.5</td><td align="right">1769.9</td><td align="right"></td><td align="left">2020 Hamburg F</td>
<td align="right">22.9</td><td align="right">2023.5</td></tr><tr><td align="right">6</td><td align="left"><a
href="http://www.tennisabstract.com/cgi-bin/player.cgi?p=StefanosTsitsipas">Stefanos&nbsp;Tsitsipas</a></td><td
align="right">22.1</td><td align="right">2022.2</td><td align="right"></td><td align="right">1939.0</td><td
align="right">1898.9</td><td align="right">1573.1</td><td align="right"></td><td align="right">1980.6</td><td
align="right">1960.5</td><td align="right">1797.6</td><td align="right"></td><td align="left">2020 Cincinnati
R16</td><td align="right">22.0</td><td align="right">2069.1</td></tr><tr><td align="right">7</td><td align="left">
```

This is static data.

Static Data and Dynamic Data



Tulane **UConn**	08.03.		76 80	W
UConn Houston	05.03.		77 71	W
East Carolina **UConn**	29.02.		63 84	W
UConn UCF Knights	26.02.		81 65	W
UConn South Florida	23.02.		78 71	W
Temple UConn	20.02. AOT	69 69	93 89	L
UConn Memphis	16.02.		64 61	W
SMU Mustangs UConn	12.02.		79 75	L
UConn Cincinnati	09.02. AOT	67 67	72 71	W
Tulsa **UConn**	06.02.		56 72	W
Memphis UConn	01.02.		70 63	L



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=11" />
<title>UConn results - Basketball, USA&nbsp;&nbsp;</title>
<noscript>
<meta http-equiv="refresh" content="0;URL='https://www.flashscore.mobi/'" />
</noscript>
<meta name="keywords" content="" />
<meta name="description" content="Follow UConn latest results, today&#039;s scores and all
of the current season&#039;s UConn results.&nbsp;&nbsp;" />
<meta name="copyright" content="Copyright (c) 2006-2020 Livesport s.r.o." />
<meta name="robots" content="NOODP,index,follow" />
<meta property="og:image"
content="https://www.flashscore.com/res/_fs/image/meta/basketball.png" />
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no, minimal-ui" />
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"><link rel="shortcut
icon" href="/res/_fs/images/10_favicons/_fs/favicon.ico?v=2" />

    <link rel="apple-touch-icon" sizes="180x180"
href="/res/_fs/images/10_favicons/_fs/touch-icon-180x180.png?v=2">
    <link rel="icon" type="image/png" sizes="32x32"
href="/res/_fs/images/10_favicons/_fs/favicon-32x32.png?v=2">
    <link rel="icon" type="image/png" sizes="16x16"
href="/res/_fs/images/10_favicons/_fs/favicon-16x16.png?v=2">
    <link rel="manifest" href="/manifest/3/?v=5">
    <meta name="theme-color" content="#91311d"><meta name="google-site-verification"
content="EW0kTHpWq9vtmZ2kox5aVFBTKObL7F1iMje1zAiWPkw" />
<script defer src="https://cdn.cookielaw.org/scripttemplates/otSDKStub.js"
type="text/javascript" data-document-language="true" charset="UTF-8" data-domain-
script="23d76a88-8ebd-439c-a0dd-6da91b895891"></script><link type="text/css"
rel="stylesheet" href="/res/_fs/build/variables.6bb42c4.css" media="screen" /><link
type="text/css" rel="stylesheet" href="/res/_fs/build/header.7578a4f.css" media="screen" />
<link type="text/css" rel="stylesheet" href="/res/_fs/build/container.230439b.css"
media="screen" /><link type="text/css" rel="stylesheet"
href="/res/_fs/build/user.9790cba.css" media="screen" /><link type="text/css"
rel="stylesheet" href="/res/_fs/build/_page.e1c6810.css" media="screen" /><link
type="text/css" rel="stylesheet" href="/res/_fs/build/banner.42d35f6.css" media="screen" />
<link type="text/css" rel="stylesheet" href="/res/_fs/build/search.a6f20a0.css"
media="screen" /><link type="text/css" rel="stylesheet"
href="/res/_fs/build/standings_in_page.c722c43.css" media="screen" /><link type="text/css"
rel="stylesheet" href="/res/_fs/build/footer.8cdb8aa.css" media="screen" /><link
type="text/css" rel="stylesheet" href="/res/_fs/build/livetableresponsive.3992b5f.css"data-
href="/res/_fs/build/livetablenonresponsive.39cdb0a.css" media="screen" /><link
type="text/css" rel="stylesheet" href="/res/_fs/build/liveTable.c6849d0.css" media="screen"
/><link type="text/css" rel="stylesheet" href="/res/_fs/build/box_over_content.192efe3.css"
media="screen" /><link type="text/css" rel="stylesheet"
href="/res/_fs/build/cookie.d9139d4.css" media="screen" /><link type="text/css"
rel="stylesheet" href="/res/_fs/build/myTeamsMenu.ec902dc.css" media="screen" /><link
type="text/css" rel="stylesheet" href="/res/_fs/build/match_odds_preview.b7991f9.css"
media="screen" /><link type="text/css" rel="stylesheet"
href="/res/_fs/build/lang_box_module.5620520.css" media="screen" /><link type="text/css"
rel="stylesheet" href="/res/_fs/build/commercial.d657a91.css"data-
href="/res/_fs/build/livetablenonresponsive.39cdb0a.css" media="screen" />
    <link rel="canonical" href="https://www.flashscore.com/team/connecticut-
```

This is dynamic data.

Web Scraping for Static Data in R

R provides several approaches for web scraping the static data. Two of them will be discussed in this workshop.

- **readLines** function: Read the source code of the HTML pages.
- **rvest** package: Capture useful data by identifying the elements contains the data in the source code.

Web Scraping for Static Data in R

Use **readLines** function for College basketball school index.

```
web_page <- readLines("https://www.sports-reference.com/cbb/schools/")
head(web_page, n = 10L)
```

```
##  [1] ""
##  [2] "<!DOCTYPE html>"
##  [3] "<html data-version=\"klecko-\" data-root=\"/home/cbb/build\" itemscope itemtype=\"https://schema.org/W
##  [4] "<head>"
##  [5] "    <meta charset=\"utf-8\">"
##  [6] "    <meta http-equiv=\"x-ua-compatible\" content=\"ie=edge\">"
##  [7] "    <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, maximum-scale=2.0\" />"
##  [8] "    <link rel=\"dns-prefetch\" href=\"https://d2p3bygnnzw9w3.cloudfront.net/req/202009101\" />"
##  [9] ""
## [10] "    <title>School Index | College Basketball at Sports-Reference.com</title>"
```

- Gives the source code.
- Needs data cleaning and organization.

Web Scraping for Static Data in R

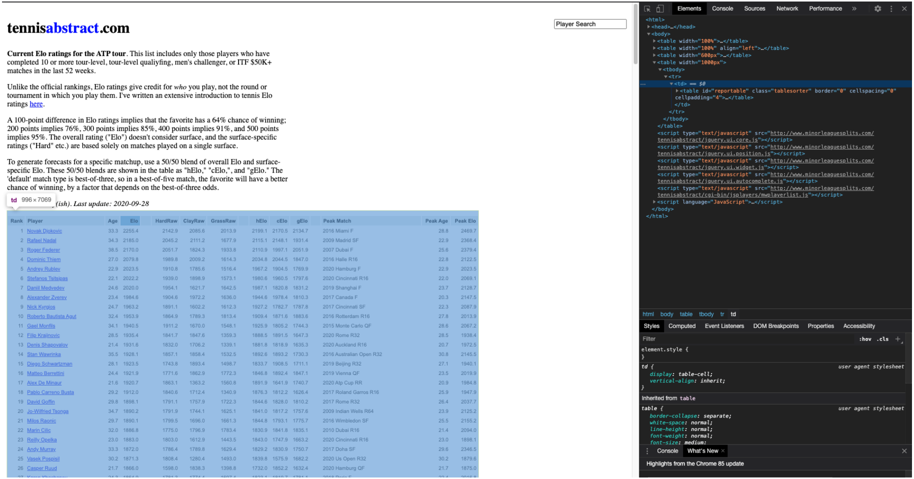
Before we talk about web scraping by **rvest** package, we need to know how to locate the elements containing the data in the source code.

- Right click the page and choose “Inspect”.
- Click “Select an element in the page to inspect it”.
- We can locate the elements by CSS selector or XPATH.

Web Scraping for Static Data in R

Use http://tennisabstract.com/reports/atp_elo_ratings.html as an example

- CSS selector: id = “reportable”, class = “tablesorter”



tennisabstract.com

Current Elo ratings for the ATP tour. This list includes only those players who have completed 10 or more tour-level, tour-level qualiyfing, men's challenger, or ITF $50K+ matches in the last 52 weeks.

Unlike the official rankings, Elo ratings give credit for *who* you play, not the round or tournament in which you play them. I've written an extensive introduction to tennis Elo ratings here.

A 100-point difference in Elo ratings implies that the favorite has a 64% chance of winning; 200 points implies 76%, 300 points implies 85%, 400 points implies 91%, and 500 points implies 95%. The overall rating ("Elo") doesn't consider surface, and the surface-specific ratings ("Hard" etc.) are based solely on matches played on a single surface.

To generate forecasts for a specific matchup, use a 50/50 blend of overall Elo and surface-specific Elo. These 50/50 blends are shown in the table as "hElo," "cElo,", and "gElo." The 'default' match type is best-of-three, so in a best-of-five match, the favorite will have a better chance of winning, by a factor that depends on the best-of-three odds.

(ish). Last update: 2020-09-28

Rank	Player	Age	Elo		HardRaw	ClayRaw	GrassRaw		hElo	cElo	gElo		Peak Match	Peak Age	Peak Elo
1	Novak Djokovic	33.3	2255.4		2142.9	2085.6	2013.9		2199.1	2170.5	2134.7		2016 Miami F	28.8	2469.7
2	Rafael Nadal	34.3	2185.0		2045.2	2111.2	1677.9		2115.1	2148.1	1931.4		2009 Madrid SF	22.9	2368.4
3	Roger Federer	38.5	2170.0		2051.7	1824.3	1933.8		2110.9	1997.1	2051.9		2007 Dubai F	25.6	2379.4
4	Dominic Thiem	27.0	2079.8		1989.8	2009.2	1614.3		2034.8	2044.5	1847.0		2016 Halle R16	22.8	2122.5
5	Andrey Rublev	22.9	2023.5		1910.8	1785.6	1516.4		1967.2	1904.5	1769.9		2020 Hamburg F	22.9	2023.5
6	Stefanos Tsitsipas	22.1	2022.2		1939.0	1898.9	1573.1		1980.6	1960.5	1797.6		2020 Cincinnati R16	22.0	2069.1
7	Daniil Medvedev	24.6	2020.0		1954.1	1621.7	1642.5		1987.1	1820.8	1831.2		2019 Shanghai F	23.7	2128.7
8	Alexander Zverev	23.4	1984.6		1904.6	1972.2	1636.0		1944.6	1978.4	1810.3		2017 Canada F	20.3	2147.5
9	Nick Kyrgios	24.7	1963.2		1891.1	1602.2	1612.3		1927.2	1782.7	1787.8		2017 Cincinnati SF	22.3	2087.9
10	Roberto Bautista Agut	32.4	1953.9		1864.9	1789.3	1813.4		1909.4	1871.6	1883.6		2016 Rotterdam R16	27.8	2013.9
11	Gael Monfils	34.1	1940.5		1911.2	1670.0	1548.1		1925.9	1805.2	1744.3		2015 Monte Carlo QF	28.6	2067.2
12	Filip Krajinovic	28.5	1935.4		1841.7	1847.6	1359.3		1888.5	1891.5	1647.3		2020 Rome R32	28.5	1938.4
13	Denis Shapovalov	21.4	1931.6		1832.0	1706.2	1339.1		1881.8	1818.9	1635.3		2020 Auckland R16	20.7	1972.5
14	Stan Wawrinka	35.5	1928.1		1857.1	1858.4	1532.5		1892.6	1893.2	1730.3		2016 Australian Open R32	30.8	2145.5
15	Diego Schwartzman	28.1	1923.5		1743.8	1893.4	1498.7		1833.7	1908.5	1711.1		2019 Beijing R32	27.1	1940.1
16	Matteo Berrettini	24.4	1921.9		1771.6	1862.9	1772.3		1846.8	1892.4	1847.1		2019 Vienna QF	23.5	2019.9
17	Alex De Minaur	21.6	1920.7		1863.1	1363.2	1560.8		1891.9	1641.9	1740.7		2020 Atp Cup RR	20.9	1984.8
18	Pablo Carreno Busta	29.2	1912.0		1840.6	1712.4	1340.9		1876.3	1812.2	1626.4		2017 Roland Garros R16	25.9	1947.9
19	David Goffin	29.8	1898.1		1791.1	1757.9	1722.3		1844.6	1828.0	1810.2		2017 Rome R32	26.4	2037.7
20	Jo-Wilfried Tsonga	34.7	1890.2		1791.9	1744.1	1625.1		1841.0	1817.2	1757.6		2009 Indian Wells R64	23.9	2125.2
21	Milos Raonic	29.7	1890.1		1799.5	1696.0	1661.3		1844.8	1793.1	1775.7		2016 Wimbledon SF	25.5	2155.2
22	Marin Cilic	32.0	1886.8		1775.0	1796.9	1783.4		1830.9	1841.8	1835.1		2010 Dubai R16	21.4	2094.0
23	Reilly Opelka	23.0	1883.0		1803.0	1612.9	1443.5		1843.0	1747.9	1663.2		2020 Cincinnati R16	23.0	1898.1
24	Andy Murray	33.3	1872.0		1786.4	1789.8	1629.4		1829.2	1830.9	1750.7		2017 Doha SF	29.6	2346.5
25	Vasek Pospisil	30.2	1871.3		1808.4	1280.4	1493.0		1839.8	1575.9	1682.2		2020 Us Open R32	30.2	1879.6
26	Casper Ruud	21.7	1866.0		1598.0	1838.3	1398.8		1732.0	1852.2	1632.4		2020 Hamburg QF	21.7	1875.0
27	Karen Khachanov	24.3	1864.0		1781.3	1774.4	1697.4		1823.1	1819.7	1781.1		2018 Paris F	22.4	2016.8

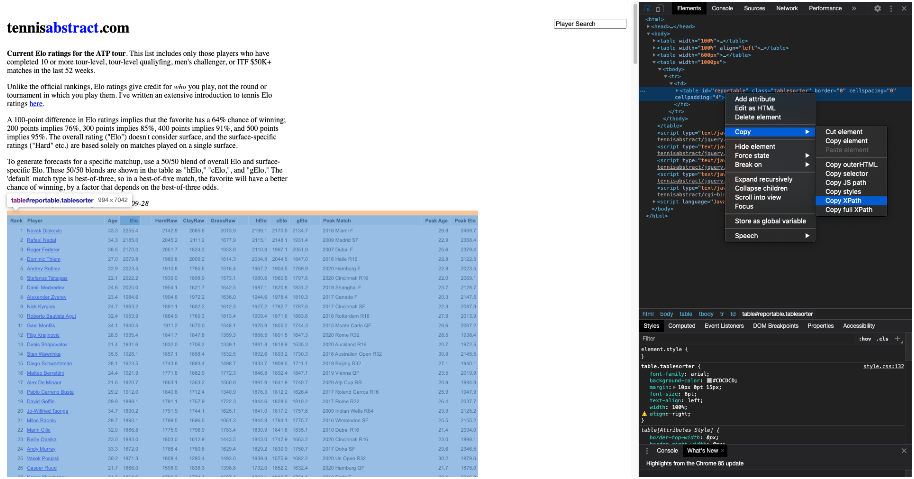
```
<html>
 ▶<head>…</head>
 ▼<body>
   ▶<table width="100%">…</table>
   ▶<table width="100%" align="left">…</table>
   ▶<table width="600px">…</table>
   ▼<table width="1000px">
     ▼<tbody>
       ▼<tr>
...      ▼<td> == $0
           ▶<table id="reportable" class="tablesorter" border="0" cellspacing="0"
           cellpadding="4">…</table>
          </td>
        </tr>
      </tbody>
    </table>
    <script type="text/javascript" src="http://www.minorleaguesplits.com/
    tennisabstract/jquery.ui.core.js"></script>
    <script type="text/javascript" src="http://www.minorleaguesplits.com/
    tennisabstract/jquery.ui.position.js"></script>
    <script type="text/javascript" src="http://www.minorleaguesplits.com/
    tennisabstract/jquery.ui.widget.js"></script>
    <script type="text/javascript" src="http://www.minorleaguesplits.com/
    tennisabstract/jquery.ui.autocomplete.js"></script>
    <script type="text/javascript" src="http://www.minorleaguesplits.com/
    tennisabstract/cgi-bin/jsplayers/mwplayerlist.js"></script>
   ▶<script language="JavaScript">…</script>
  </body>
</html>
```

```
element.style {
}
td {                                        user agent stylesheet
   display: table-cell;
   vertical-align: inherit;
}
Inherited from table
table {                                     user agent stylesheet
   border-collapse: separate;
   white-space: normal;
   line-height: normal;
   font-weight: normal;
   font-size: medium;
```

Highlights from the Chrome 85 update

Web Scraping for Static Data in R

- XPATH: '//*[@id="reportable"]'



tennisabstract.com

Current Elo ratings for the ATP tour. This list includes only those players who have completed 10 or more tour-level, tour-level qualiyfing, men's challenger, or ITF $50K+ matches in the last 52 weeks.

Unlike the official rankings, Elo ratings give credit for *who* you play, not the round or tournament in which you play them. I've written an extensive introduction to tennis Elo ratings here.

A 100-point difference in Elo ratings implies that the favorite has a 64% chance of winning; 200 points implies 76%, 300 points implies 85%, 400 points implies 91%, and 500 points implies 95%. The overall rating ("Elo") doesn't consider surface, and the surface-specific ratings ("Hard" etc.) are based solely on matches played on a single surface.

To generate forecasts for a specific matchup, use a 50/50 blend of overall Elo and surface-specific Elo. These 50/50 blends are shown in the table as "hElo," "cElo,", and "gElo." The 'default' match type is best-of-three, so in a best-of-five match, the favorite will have a better chance of winning, by a factor that depends on the best-of-three odds.

Rank	Player	Age	Elo		HardRaw	ClayRaw	GrassRaw		hElo	cElo	gElo		Peak Match	Peak Age	Peak Elo
1	Novak Djokovic	33.3	2255.4		2142.9	2085.6	2013.9		2199.1	2170.5	2134.7		2016 Miami F	28.8	2469.7
2	Rafael Nadal	34.3	2185.0		2045.2	2111.2	1677.9		2115.1	2148.1	1931.4		2009 Madrid SF	22.9	2368.4
3	Roger Federer	38.5	2170.0		2051.7	1824.3	1933.8		2110.9	1997.1	2051.9		2007 Dubai F	25.6	2379.4
4	Dominic Thiem	27.0	2079.8		1989.8	2009.2	1614.3		2034.8	2044.5	1847.0		2016 Halle R16	22.8	2122.5
5	Andrey Rublev	22.9	2023.5		1910.8	1785.6	1516.4		1967.2	1904.5	1769.9		2020 Hamburg F	22.9	2023.5
6	Stefanos Tsitsipas	22.1	2022.2		1939.0	1898.9	1573.1		1980.6	1960.5	1797.6		2020 Cincinnati R16	22.0	2069.1
7	Daniil Medvedev	24.6	2020.0		1954.1	1621.7	1642.5		1987.1	1820.8	1831.2		2019 Shanghai F	23.7	2128.7
8	Alexander Zverev	23.4	1984.6		1904.6	1972.2	1636.0		1944.6	1978.4	1810.3		2017 Canada F	20.3	2147.5
9	Nick Kyrgios	24.7	1963.2		1891.1	1602.2	1612.3		1927.2	1782.7	1787.8		2017 Cincinnati SF	22.3	2087.9
10	Roberto Bautista Agut	32.4	1953.9		1864.9	1789.3	1813.4		1909.4	1871.6	1883.6		2016 Rotterdam R16	27.8	2013.9
11	Gael Monfils	34.1	1940.5		1911.2	1670.0	1548.1		1925.9	1805.2	1744.3		2015 Monte Carlo QF	28.6	2067.2
12	Filip Krajinovic	28.5	1935.4		1841.7	1847.6	1359.3		1888.5	1891.5	1647.3		2020 Rome R32	28.5	1938.4
13	Denis Shapovalov	21.4	1931.6		1832.0	1706.2	1339.1		1881.8	1818.9	1635.3		2020 Auckland R16	20.7	1972.5
14	Stan Wawrinka	35.5	1928.1		1857.1	1858.4	1532.5		1892.6	1893.2	1730.3		2016 Australian Open R32	30.8	2145.5
15	Diego Schwartzman	28.1	1923.5		1743.8	1893.4	1498.7		1833.7	1908.5	1711.1		2019 Beijing R32	27.1	1940.1
16	Matteo Berrettini	24.4	1921.9		1771.6	1862.9	1772.3		1846.8	1892.4	1847.1		2019 Vienna QF	23.5	2019.9
17	Alex De Minaur	21.6	1920.7		1863.1	1363.2	1560.8		1891.9	1641.9	1740.7		2020 Atp Cup RR	20.9	1984.8
18	Pablo Carreno Busta	29.2	1912.0		1840.6	1712.4	1340.9		1876.3	1812.2	1626.4		2017 Roland Garros R16	25.9	1947.9
19	David Goffin	29.8	1898.1		1791.1	1757.9	1722.3		1844.6	1828.0	1810.2		2017 Rome R32	26.4	2037.7
20	Jo-Wilfried Tsonga	34.7	1890.2		1791.9	1744.1	1625.1		1841.0	1817.2	1757.6		2009 Indian Wells R64	23.9	2125.2
21	Milos Raonic	29.7	1890.1		1799.5	1696.0	1661.3		1844.8	1793.1	1775.7		2016 Wimbledon SF	25.5	2155.2
22	Marin Cilic	32.0	1886.8		1775.0	1796.9	1783.4		1830.9	1841.8	1835.1		2010 Dubai R16	21.4	2094.0
23	Reilly Opelka	23.0	1883.0		1803.0	1612.9	1443.5		1843.0	1747.9	1663.2		2020 Cincinnati R16	23.0	1898.1
24	Andy Murray	33.3	1872.0		1786.4	1789.8	1629.4		1829.2	1830.9	1750.7		2017 Doha SF	29.6	2346.5
25	Vasek Pospisil	30.2	1871.3		1808.4	1280.4	1493.0		1839.8	1575.9	1682.2		2020 Us Open R32	30.2	1879.6
26	Casper Ruud	21.7	1866.0		1598.0	1838.3	1398.8		1732.0	1852.2	1632.4		2020 Hamburg QF	21.7	1875.0

```
<html>
 ▸<head>…</head>
 ▾<body>
   ▸<table width="100%">…</table>
   ▸<table width="100%" align="left">…</table>
   ▸<table width="600px">…</table>
   ▾<table width="1000px">
     ▾<tbody>
       ▾<tr>
         ▾<td>
... ▸<table id="reportable" class="tablesorter" border="0" cellspacing="0"
           cellpadding="4">
           </td>
         </tr>
       </tbody>
     </table>
     <script type="text/jav
     tennisabstract/jquery.
     <script type="text/jav
     tennisabstract/jquery.
     <script type="text/jav
     tennisabstract/jquery.
     <script type="text/jav
     tennisabstract/jquery.
     <script type="text/jav
     tennisabstract/cgi-bin
   ▸<script language="Java
   </body>
</html>
```

```
element.style {
}
table.tablesorter {                                   style.css:132
   font-family: arial;
   background-color: ■#CDCDCD;
   margin: ▸ 10px 0pt 15px;
   font-size: 8pt;
   text-align: left;
   width: 100%;
⚠ ~~align: right;~~
}
table[Attributes Style] {
   border-top-width: 0px;
```

Highlights from the Chrome 85 update

Web Scraping for Static Data in R

Next, we are going to talk about how to use **rvest** for web scraping by using an example.

- Install **rvest** package from cran.

```
install.packages("rvest", repos = "http://cran.us.r-project.org")
require("rvest")
```

Web Scraping for Static Data in R

- Web scraping data from http://tennisabstract.com/reports/atp_elo_ratings.html

```r
url_elo <- "http://tennisabstract.com/reports/atp_elo_ratings.html"
webpage <- read_html(url_elo)
elo_class <- webpage %>%
  html_nodes(".tablesorter") %>%
  html_table()
elo_id <- webpage %>%
  html_nodes("#reportable") %>%
  html_table()
identical(elo_class, elo_id)
```

```
## [1] TRUE
```

Web Scraping for Static Data in R

```r
elo_xpath <- webpage %>%
  html_nodes(xpath = '//*[@id="reportable"]') %>%
  html_table()
identical(elo_class, elo_xpath)
```

```
## [1] TRUE
```

```r
head(elo_class[[1]])
```

```
##   Rank             Player  Age    Elo    HardRaw ClayRaw GrassRaw      hElo
## 1    1     Novak Djokovic 33.3 2255.4 NA  2142.9  2085.6   2013.9 NA 2199.1
## 2    2       Rafael Nadal 34.3 2185.0 NA  2045.2  2111.2   1677.9 NA 2115.1
## 3    3      Roger Federer 38.5 2170.0 NA  2051.7  1824.3   1933.8 NA 2110.9
## 4    4      Dominic Thiem 27.0 2079.8 NA  1989.8  2009.2   1614.3 NA 2034.8
## 5    5      Andrey Rublev 22.9 2023.5 NA  1910.8  1785.6   1516.4 NA 1967.2
## 6    6 Stefanos Tsitsipas 22.1 2022.2 NA  1939.0  1898.9   1573.1 NA 1980.6
##      cElo   gElo             Peak Match Peak Age Peak Elo
## 1 2170.5 2134.7 NA        2016 Miami F     28.8   2469.7
## 2 2148.1 1931.4 NA      2009 Madrid SF     22.9   2368.4
## 3 1997.1 2051.9 NA        2007 Dubai F     25.6   2379.4
## 4 2044.5 1847.0 NA      2016 Halle R16     22.8   2122.5
## 5 1904.5 1769.9 NA      2020 Hamburg F     22.9   2023.5
## 6 1960.5 1797.6 NA 2020 Cincinnati R16     22.0   2069.1
```

Web Scraping for Static Data in R

- Except **html_nodes** and **html_table**, there are many other frequently used functions in **rvest**.
  - **html_node** : extract element
  - **html_text** : extract text
  - **html_attrs** : extract attributes
  - **html_form** : extract forms
- Please look up rvest cran for more information.
- SelectorGadget is a convenient tool to identify CSS selector.

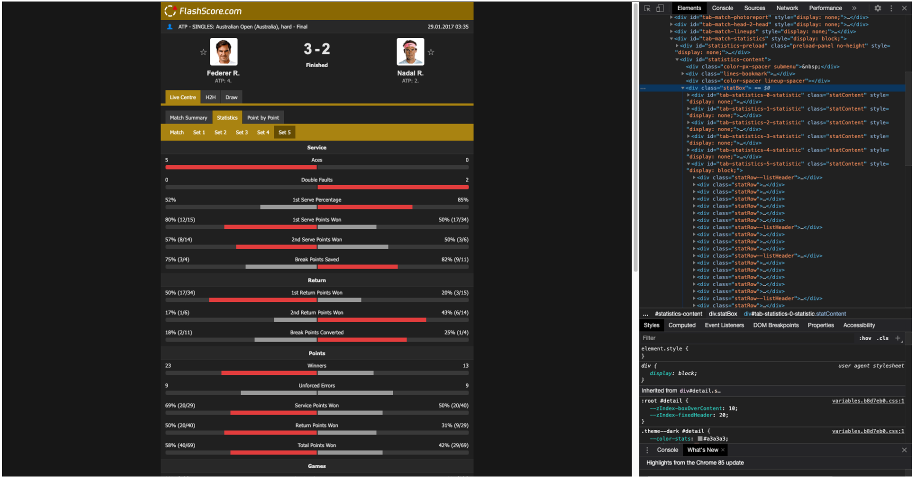
Web Scraping for Dynamic Data in R

- What dynamic data display in the website can be changed in response to the user interaction.
- We need to automate the web browsing process in R for the dynamic data.
- **RSelenium** package helps this automating process by providing connection to Selenium Server.
- Install **RSelenium** package.

```
devtools::install_github("ropensci/RSelenium")
require("RSelenium")
```

Web Scraping for Dynamic Data in R

- Use **RSelenium** to extract data on 2017 Australian Open Final



FlashScore.com

ATP - SINGLES: Australian Open (Australia), hard - Final 29.01.2017 03:35

Federer R. ATP: 4. 3 - 2 Finished Nadal R. ATP: 2.

	Service	
5	Aces	0
0	Double Faults	2
52%	1st Serve Percentage	85%
80% (12/15)	1st Serve Points Won	50% (17/34)
57% (8/14)	2nd Serve Points Won	50% (3/6)
75% (3/4)	Break Points Saved	82% (9/11)
	Return	
50% (17/34)	1st Return Points Won	20% (3/15)
17% (1/6)	2nd Return Points Won	43% (6/14)
18% (2/11)	Break Points Converted	25% (1/4)
	Points	
23	Winners	13
9	Unforced Errors	9
69% (20/29)	Service Points Won	50% (20/40)
50% (20/40)	Return Points Won	31% (9/29)
58% (40/69)	Total Points Won	42% (29/69)
	Games	

```
▸<div id="tab-match-photoreport" style="display: none;">…</div>
▸<div id="tab-match-head-2-head" style="display: none;">…</div>
▸<div id="tab-match-lineups" style="display: none;">…</div>
▾<div id="tab-match-statistics" style="display: block;">
  ▸<div id="statistics-preload" class="preload-panel no-height" style=
  "display: none;">…</div>
  ▾<div id="statistics-content">
     <div class="color-px-spacer submenu">&nbsp;</div>
    ▸<div class="lines-bookmark">…</div>
     <div class="color-spacer lineup-spacer"></div>
...   ▾<div class="statBox"> == $0
      ▸<div id="tab-statistics-0-statistic" class="statContent" style=
      "display: none;">…</div>
      ▸<div id="tab-statistics-1-statistic" class="statContent" style=
      "display: none;">…</div>
      ▸<div id="tab-statistics-2-statistic" class="statContent" style=
      "display: none;">…</div>
      ▸<div id="tab-statistics-3-statistic" class="statContent" style=
      "display: none;">…</div>
      ▸<div id="tab-statistics-4-statistic" class="statContent" style=
      "display: none;">…</div>
      ▾<div id="tab-statistics-5-statistic" class="statContent" style=
      "display: block;">
        ▸<div class="statRow--listHeader">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow--listHeader">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow--listHeader">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow">…</div>
        ▸<div class="statRow--listHeader">…</div>
        ▸<div class="statRow">…</div>
```

... #statistics-content div.statBox div#tab-statistics-0-statistic.statContent

```
element.style {
}
div {                                    user agent stylesheet
   display: block;
}
Inherited from div#detail.s…
:root #detail {                       variables.b8d7eb0.css:1
   --zIndex-boxOverContent: 10;
   --zIndex-fixedHeader: 20;
}
.theme--dark #detail {                variables.b8d7eb0.css:1
   --color-stats: ▨#a3a3a3;
```

Highlights from the Chrome 85 update

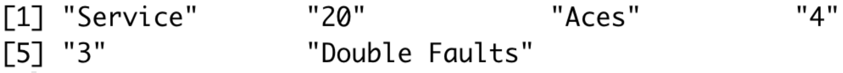
Web Scraping for Dynamic Data in R

- Connect to a selenium server and open brower.

```r
rD <- rsDriver(port = 5561L, chromever = "85.0.4183.87")
remDr <- rD$client
```

- Extract Information and organize data.

```r
url <- "http://www.flashscore.com/match/Cj6I5iL9/#match-statistics;0"
remDr$navigate(url)
webElem <- remDr$findElements(using = 'class', "statBox")
webElem <- unlist(lapply(webElem, function(x){x$getElementText()}))[[1]]
# head(unlist(strsplit(webElem, split = '\n')))
remDr$close()
```



```
 [1] "Service"           "20"                "Aces"              "4"
 [5] "3"                 "Double Faults"
```

Web Scraping for Dynamic Data in R

- Frequently used functions of **RSelenium**:
  - rsDriver() : start a selenium server
  - navigate() : navigate web pages
  - findElements() : find elements by CSS seclector or XPATH
  - getPageSource() : get current page source
  - clickElement() : click element
- Please go to RSelenium cran for more details.

Web Scraping for Dynamic Data in R

Exercise: Web Scraping for the history basketball recording of UConn

https://www.flashscore.com/team/connecticut-huskies/8rqVf3Tj/results/

- Start a selenium server and open web brower.

```
require("RSelenium")
rD <- rsDriver(port = 5533L, chromever = "85.0.4183.87")
remDr <- rD$client
url <- "https://www.flashscore.com/team/connecticut-huskies/8rqVf3Tj/result
remDr$navigate(url)
```

Web Scraping for Dynamic Data in R

- Automate to click all “show more results”.

```r
repeat{
  b <- tryCatch({
    suppressMessages({
      webElemMore <- remDr$findElement(using = 'xpath',
                        '//*[@id="live-table"]/div[1]/div/div/a')
      webElemMore$clickElement()
    })
  }, error = function(e) e)
  if(inherits(b, "error")) break
}
```

- Extract data, such as time, home/away, score and result.

```r
webElemTime <- remDr$findElements(using = 'xpath',
                              '//*[@class="event__time"]')
webElemTime <-
  unlist(lapply(webElemTime, function(x){x$getElementText()}))
webElemTime <- gsub("\\n", " ", webElemTime)
```

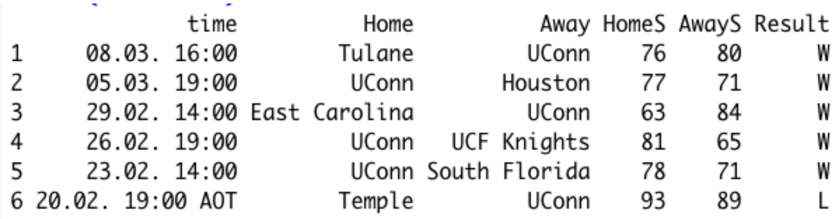
Web Scraping for Dynamic Data in R

```r
webElemHome <-
  remDr$findElements(using = 'class',
                     'event__participant')
webElemHome <-
  unlist(lapply(webElemHome, function(x){x$getElementText()}))

webElemScore <-
  remDr$findElements(using = 'class', 'event__score')
webElemScore <-
  unlist(lapply(webElemScore, function(x){x$getElementText()}))

webElemResult <-
  remDr$findElements(using = 'class', 'wld')
webElemResult <-
  unlist(lapply(webElemResult, function(x){x$getElementText()}))
```

Web Scraping for Dynamic Data in R

- Organize dataset.

```
n <- length(webElemHome)
basketball <-
  data.frame(time = webElemTime,
             Home = webElemHome[seq(n) %% 2 == 1],
             Away = webElemHome[seq(n) %% 2 == 0],
             HomeS = webElemScore[seq(n) %% 2 == 1],
             AwayS = webElemScore[seq(n) %% 2 == 0],
             Result = webElemResult)

head(basketball)
remDr$close()
```



```
            time          Home          Away HomeS AwayS Result
1      08.03. 16:00        Tulane         UConn    76    80      W
2      05.03. 19:00         UConn       Houston    77    71      W
3      29.02. 14:00 East Carolina         UConn    63    84      W
4      26.02. 19:00         UConn   UCF Knights    81    65      W
5      23.02. 14:00         UConn South Florida    78    71      W
6 20.02. 19:00 AOT        Temple         UConn    93    89      L
```

Summary

- For different kinds of data, we need to use different web scraping techiniques with R.
- One can simply use **read.csv** or related functions to directly import organized files from web pages.
- The static data can be extract with the help of **rvest**.
- We could use **RSelenium** to parse the dynamic data.

Resources

- CSS and HTML crash course
- rvest
- RSelenium
- R task view: web technology

Acknowledgement

This slides are modified from Dr. Kovalchik's material and Wanwan Xu's slides.