



ETC5512: Wild Caught Data

Week 11

Sports data and web scraping

Lecturer: *Dianne Cook*

Department of Econometrics and Business Statistics

✉ ETC5512.Clayton-x@monash.edu

Image source: <https://www.welcometocountry.org/digital-genocide-ash-bartys-race-repeatedly-removed/>

What about Barty!



- ◎ current ranking: 1
- ◎ singles titles: 8
- ◎ Prize Money:
\$17,594,569
- ◎ Win/Loss Singles:
252/94

Home-grown champion

Sports data

There's a treasure trove of data on sports, buried in web sites.



html source

```
<source srcset="https://photoresources.wtatennis.com/photo-resources/2019/10/08/f14eec26-4f99-4563-b904-d94b7c70b7a1/vnoiRejq.jpg?width=56, https://photoresources.wtatennis.com/photo-resources/2019/10/08/f14eec26-4f99-4563-b904-d94b7c70b7a1/vnoiRejq.jpg?width=112 2x" media="(max-width: 840px)">
<source srcset="https://photoresources.wtatennis.com/photo-resources/2019/10/08/f14eec26-4f99-4563-b904-d94b7c70b7a1/vnoiRejq.jpg?width=56, https://photoresources.wtatennis.com/photo-resources/2019/10/08/f14eec26-4f99-4563-b904-d94b7c70b7a1/vnoiRejq.jpg?width=112, https://photoresources.wtatennis.com/photo-resources/2019/10/08/f14eec26-4f99-4563-b904-d94b7c70b7a1/vnoiRejq.jpg?width=168 2x" media="(min-width: 840px)">
![Ashleigh Barty – Default Crop](https://photoresources.wtatennis.com/photo-resources/2019/10/08/f14eec26-4f99-4563-b904-d94b7c70b7a1/vnoiRejq.jpg?width=56) = $0
  </picture>
</div>
▶ <div class="player-name">...</div>
</td>
▶ <td class="stats-list__cell stats-list__cell--rank stats-list__cell--fixed-width stats-list__cell--current">...</td>
▶ <td class="stats-list__cell stats-list__cell--matches stats-list__cell--fixed-width ">...</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="Aces">79</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="Double_Faults">26</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="first_serve_percent">61 %</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="first_serve_won_percent">73.4%</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="second_serve_won_percent">49.5%</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="service_points_won_percent">64.1%</td>
  <td class="stats-list__cell stats-list__cell--stat stats-list__column-serve " data-stat="breakpoint_saved_percent">65.9%</td>
```

html is text, very long-winded, but nicely organised by tags. Web scraping allows harvesting data provided in web pages, by extracting the data components of the text.

```

## # A tibble: 22 x 4
##   Player           Rank Matches Aces
##   <chr>          <int>  <int> <int>
## 1 ASHLEIGH A. BARTY     1      14    79
## 2 KAROLINA K. PLISKOVA  3      11    76
## 3 SOFIA S. KENIN      4      18    50
## 4 ELINA E. SVITOLINA   5      14    48
## 5 KIKI K. BERTENS     7      13    67
## 6 BELINDA B. BENCIC    8      13    46
## 7 SERENA S. WILLIAMS  9       8    50
## 8 NAOMI N. OSAKA       10     7    67
## 9 ARYNA A. SABALENKA  11     15    64
## 10 PETRA P. KVITOVA    12     15    77
## 11 MADISON M. KEYS     13     8    46
## 12 GARBIÑE G. MUGURUZA 16     20   122
## 13 ELENA E. RYBAKINA   17     25   146
## 14 MARIA M. SAKKARI    20     15    60
## 15 ELISE E. MERTENS    23     14    46
## # ... with 7 more rows

```

That took me about a half day to work out.

- ◎ The WTA (women's tennis) web site is difficult to scrape because the table content is dynamic. There are numerous javascripts which extract and load the data.
- ◎ The trick for a page like this is to save a local copy of the web page, and read it into R from this. Directly reading from the URL gets empty objects.
- ◎ It's not easy to tell that a page is dynamic, and its hard to determine if its just stupid me. Need more practice.
- ◎ ATP (men's tennis site) is much easier - its just tables, even though the reader can choose to display different tables in the page. This format is easier to automate.

```
library(rvest)
library(tidyverse)
url_atp <- "https://www.atptour.com/en/rankings/s
atp_html <- read_html(url_atp)
atp_rankings <- html_node(atp_html, "table") %>%
  html_table(fill=TRUE)
```

```

## # A tibble: 28 x 6
##   Player          Age Points `Tourn Played` `Points Droppin... `Next Best`
##   <chr>        <int>  <chr>      <int>  <chr>           <int>
## 1 Dominic Thiem    26 7,045       21 1,000            90
## 2 Daniil Medvedev  24 5,890       23 45              45
## 3 Stefanos Tsitsipas 21 4,745       26 10              90
## 4 Alexander Zverev  22 3,630       25 45              45
## 5 David Goffin     29 2,555       27 10              45
## 6 Roberto Bautista Ag. 31 2,360       23 10              45
## 7 Diego Schwartzman 27 2,265       23 45              45
## 8 Andrey Rublev     22 2,234       24 53              45
## 9 Denis Shapovalov  20 2,075       27 90              45
## 10 Felix Auger-Aliassi... 19 1,771       26 45              45
## 11 Benoit Paire     30 1,738       32 10              45
## 12 Dusan Lajovic    29 1,695       27 25              20
## 13 Taylor Fritz     22 1,510       31 10              45
## 14 Filip Krajinovic 28 1,343       21 106             20
## 15 Borna Coric      23 1,320       24 10              45
## 16 Jan-Lennard Struff 29 1,315       27 90              45
## 17 Adrian Mannarino 31 1,191       30 25              20
## 18 Albert Ramos-Vinolas 32 1,130       31 45              45
## 19 Ugo Humbert      21 1,111       31 26              20
## 20 Miomir Kecmanovic 20 1,028       26 188             20
## # ... with 8 more rows

```



GIFSec.com

via GIPHY

Data will require more processing

Notice the format of variables:

- ◎ Points is interpreted as a character
- ◎ Points dropping is also a character

the "," in the field isn't read as a separator in a number. These columns will need to be converted to numeric, after stripping out the "," with a text substitution.

case study |

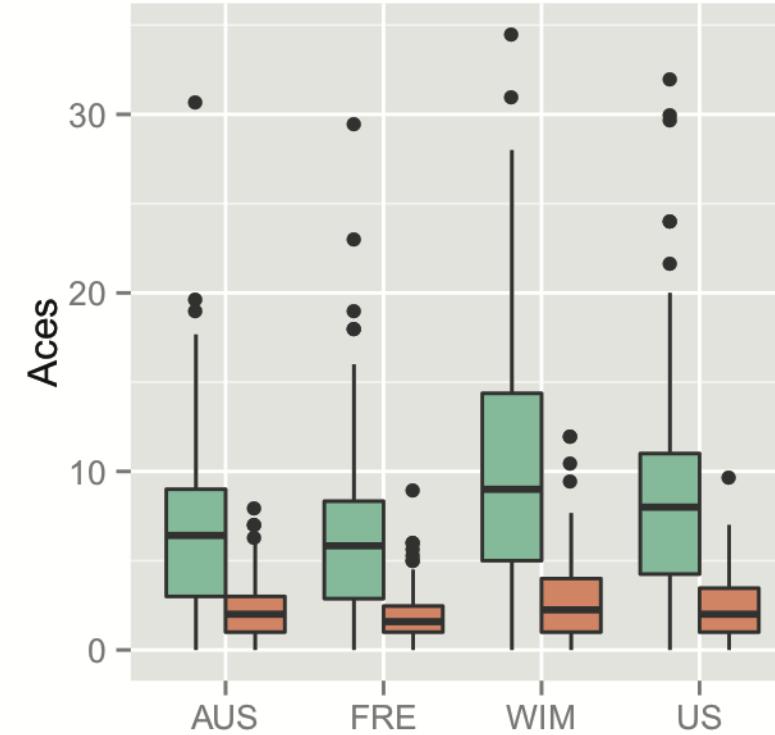
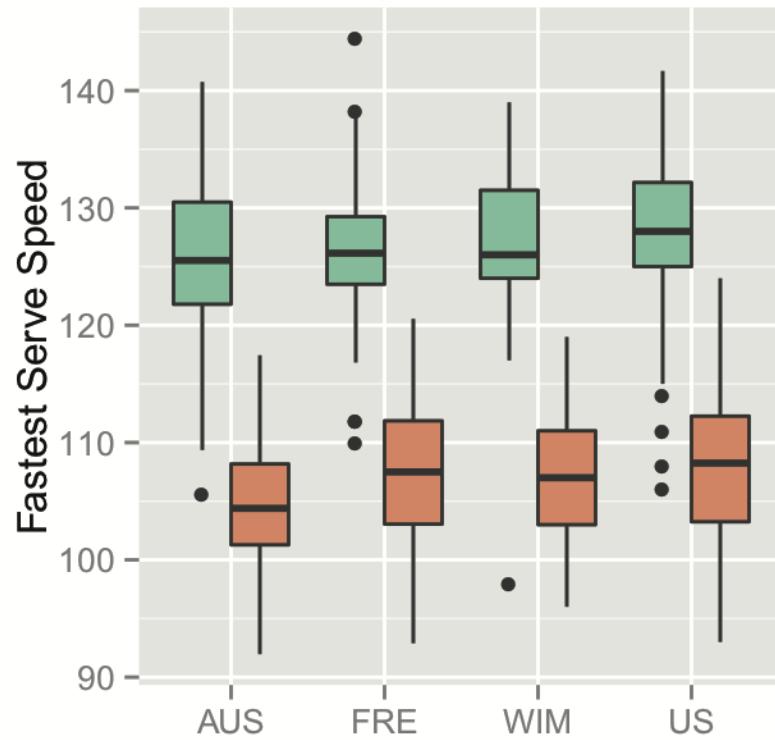
In tennis, do smashes
win matches?

Smashes win matches analysis

Data for women and men's singles matches was scraped from the 2012 Grand Slam web sites. Statistics for each match were recorded:

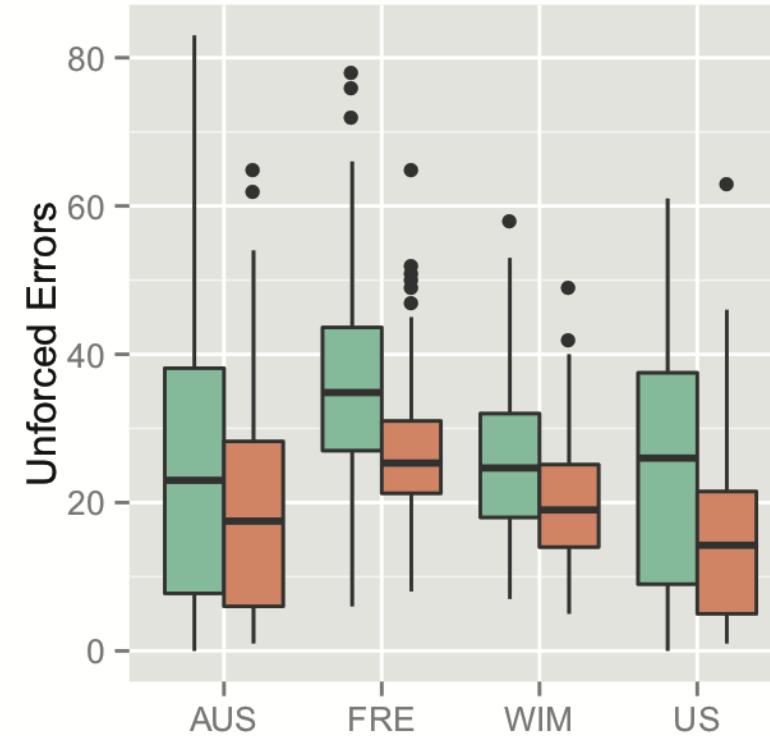
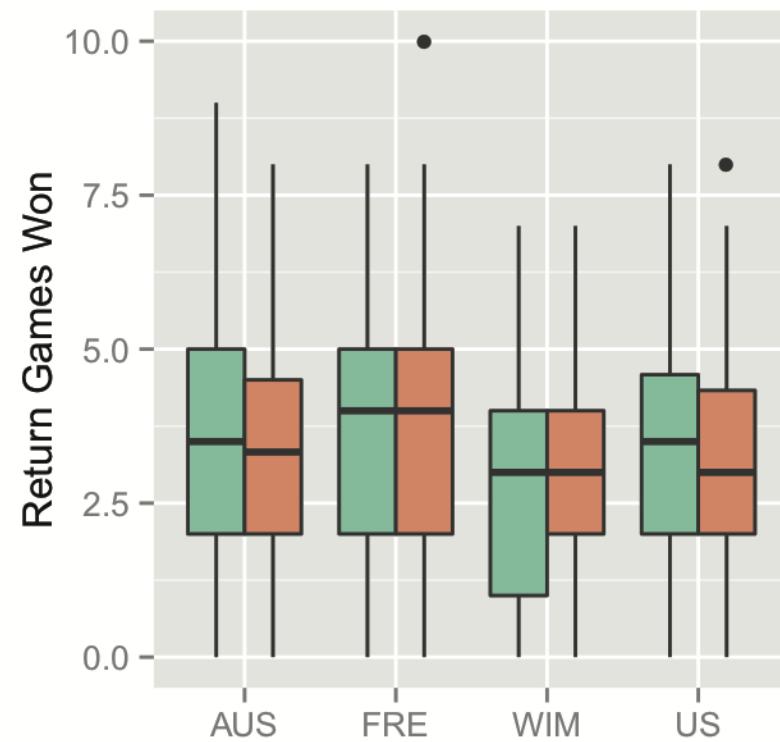
- ◉ Aces
- ◉ Fastest serve speed
- ◉ Winners
- ◉ Unforced errors
- ◉ Return games won
- ◉ First serve %
- ◉ Second serve %
- ◉ Receiving points win

Smashes win matches analysis



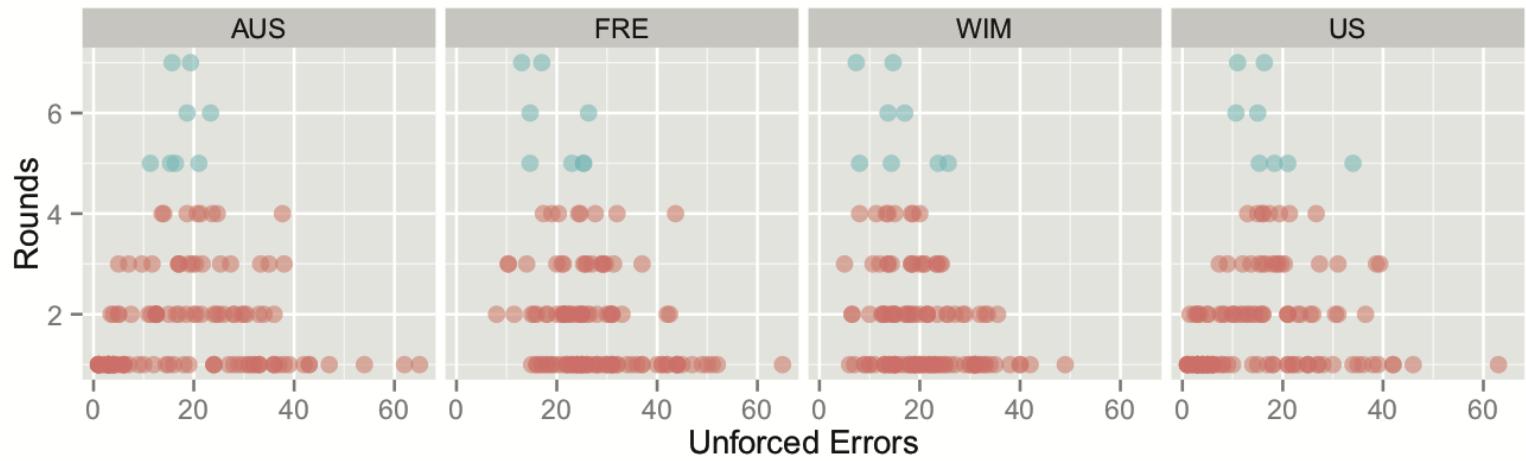
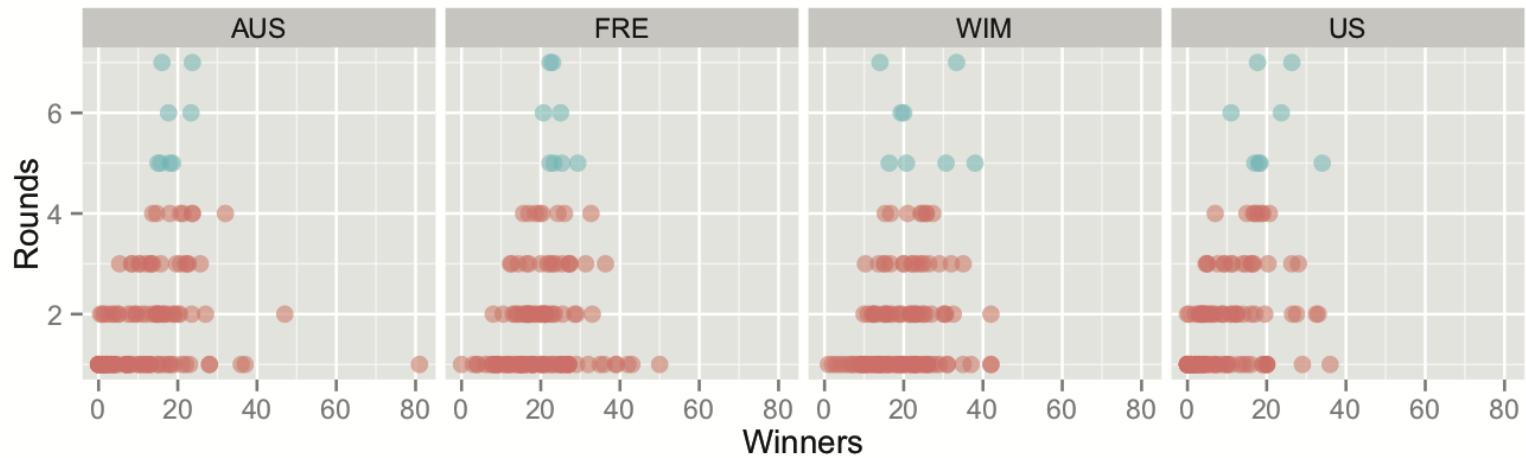
Fastest serve speed, and number of aces, in a match, by Grand Slam, and comparing men and women.

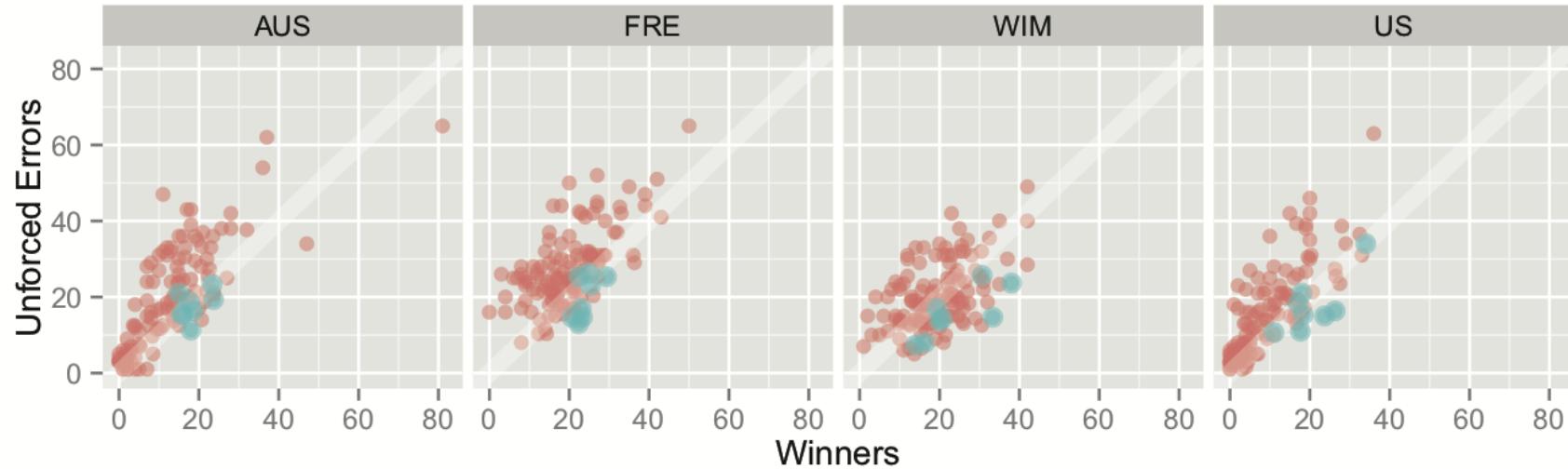
Smashes win matches analysis



Return games won, and number of unforced errors.

Higher Round number indicates player made it further in the tournament. Statistics for women's singles matches.

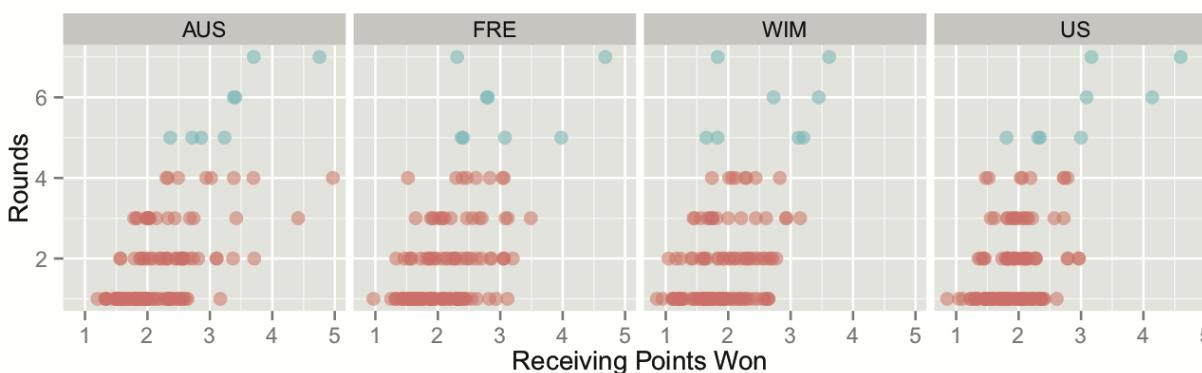
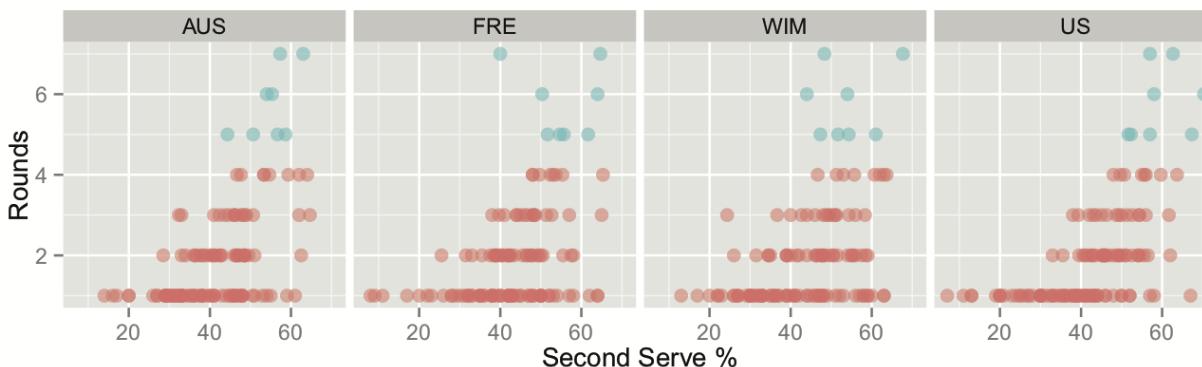
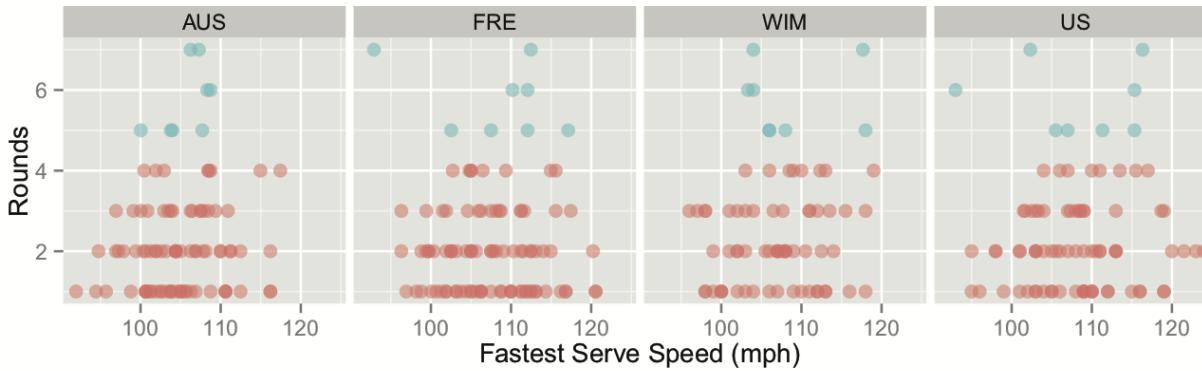




Generally, you want to have more winners than unforced errors. Too few winners might indicate that you are not working hard enough, to be able to win the match, and too many might indicate having to work too hard, so too risky, to win the match. Similar for men's matches.

How important is serving?

Turns out, first serve not so much, as long as its reasonably good. The second serve % is a big indicator of progressing through a tournament, along with receiving points won.



Rules for tournament progression

Table 1. The most important statistics and values to match for men and women aiming to make the quarter-finals

<i>Statistic</i>	<i>Men's rule</i>	<i>Women's rule</i>
Return points won	> 39.0	> 45.6
First serve winning %	> 74.3	> 61.8
Second serve winning %	> 52.9	> 48.8
Unforced errors	between 22.3, 35.2	between 12.3, 26.5

Odds to win

Knowing the statistics of players in the first two rounds, gives pretty reliable odds of predicting the quarter finalists.

Table 2. Example odds for the women's and men's 2013 Australian Open

<i>Player</i>	<i>Odds</i>	<i>Player</i>	<i>Odds</i>
Serena Williams	1.1	Novak Djokovic	1.1
Maria Sharapova	1.2	Andy Murray	1.1
Sloane Stephens	1.2	Roger Federer	1.2
Caroline Wozniacki	1.2	Marin Cilic	1.3
Venus Williams	1.3	Juan Martin Del Potro	1.3
Victoria Azarenka	1.3	Jo-Wilfried Tsonga	1.4
Svetlana Kuznetsova	1.4	Tomas Berdych	1.7
Jamie Hampton	1.4	Stanislas Wawrinka	1.7

Women: S. Williams, Sharapova, Stephens, Azarenka, Kuznetsova, Radwańska, Na, Makarova

Men: Djokovic, Murray, Federer, Tsonga, Berdych, Chardy, Almagro, Ferrer

Smashes win matches analysis

Smashes are important, but only up to a point! The players who are successful are those who force the pace of the game with smashes, but who do not overdo it. Defensive play is probably more important: being able to win points on the opponent's serve, and winning points on one's own second serve, correlates best with getting through to the quarter-finals and the big money prizes.

Legality of scraping

- ◉ Is web scraping legal? Yes, unless you use it unethically.
 - ◉ Search engines started as web scrapers, and it boost the visibility of the page, increasing the positive sentiment towards scraping.
 - ◉ **Copyright infringement:** if the data is copyright protected, you can't upload it to your own site, or use it for commercial purposes
 - ◉ **Violation of the Computer Fraud and Abuse Act:** unauthorised access, eg jerk.com
 - ◉ **Trespass to Chattel:** Don't make so many requests that you slow the web site's performance

What do you think are reasonable uses of scraping the ATP data?

- ◉ Pull the statistics of your next opponent to find their strengths and weaknesses
- ◉ Develop odds for a gambling enterprise
- ◉ Examine the statistics of a player prior to an injury to determine if it might be preventable
- ◉ Develop a player ranking to build the draw for a tournament

Keep in mind

- ⌚ Web scraping doesn't work forever
 - ⚽ Web sites change, and code needs to be rewritten
- ⌚ A web site can be made to be almost scrape-proof, but technically if its visible is scrapable
- ⌚ Its more than just coding. Its pretty time-intensive to build a scraper, and then the data extracted needs to be wrangled into shape

Be polite!

```
library(polite)
tennis_bow <- bow(
  url = "https://www.atptour.com/en", # base URL
  user_agent = "Wild-caught Data <https://wcd.numbat.space>", # identify
  force = TRUE
)
tennis_bow

## <polite session> https://www.atptour.com/en
##   User-agent: Wild-caught Data <https://wcd.numbat.space>
##   robots.txt: 20 rules are defined for 0 bots
##   Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

Hypertext Markup Language

HTML Introduction

```
<!DOCTYPE html>
<html>
<body>
    This is my first web page
</body>
</html>
```

Common tags

- ➊ *html*: opening tag declaring everything between is html format
- ➋ *body*: main content appearing in the page
- ➌ *title*: title in browser border
- ➍ *table, tr, td*: table start, row and column starts
- ➎ *img*: insert an image here
- ➏ *a*: insert a link, could be external, or internal
- ➐ *h1, h2, h3*: headings in the document

Attributes

```
<h1 style="color:Tomato;"> Bilby </h1>



<a title="my image" href="https://commons.wikimedia.org/wiki/image.jpg">
</a>
```

Elements

- ◉ Bilby
- ◉ The image is the element in the third example

Learning to scrape with rvest

```
library(rvest)
library(tidyverse)
url_atp <- "https://www.atptour.com/en/rankings/s
atp_html <- read_html(url_atp)
atp_rankings <- html_node(atp_html, "table") %>%
  html_table(fill=TRUE)
```

Different selector

```
lego_movie <- read_html("http://www.imdb.com/titl  
rating <- lego_movie %>%  
  html_nodes("strong span") %>%  
  html_text() %>%  
  as.numeric()  
rating  
## [1] 7.7
```

Intro to Cascading Style Sheets (css)

- ◎ a way to style and present HTML.
- ◎ to understand parts of the html, requires knowledge of the styling components too



via GIPHY

```
cast <- lego_movie %>%
  html_nodes("#titleCast .primary_photo img") %>%
  html_attr("alt")
cast

## [1] "Will Arnett"      "Elizabeth Banks" "Craig
## [5] "David Burrows"    "Anthony Daniels"  "Charl
## [9] "Keith Ferguson"   "Will Ferrell"     "Will
## [13] "Morgan Freeman"  "Todd Hansen"    "Jonah
```

When pages make it difficult

```
url <- "https://www.wtatennis.com/stats"  
wta_html <- read_html(url)  
wta_rankings <- html_node(wta_html, "table")  
wta_rankings  
  
## {xml_missing}  
## <NA>
```

Download a copy first

```
wta_html <- read_html("wta_rankings2.htm")
wta_rankings <- html_node(wta_html, "table") %>% html_table(fill=TRUE)
wta_rankings <- wta_rankings %>%
  janitor::remove_empty() %>%
  as_tibble()
wta_rankings

## # A tibble: 207 x 17
##       Pos Player   Rank Matches    Aces `DF` Double Fau... `1st` Srv %
##   <int> <chr>   <int>   <int>   <int>   <int> <chr>   <int> <chr>
## 1      1 ASHLE...     1      14      79          26  61 %
## 2      2 SIMON...     2      12      29          21 68.8 %
## 3      3 KAROL...     3      11      76          36 63.2 %
## 4      4 SOFIA...     4      18      50          48 69.8 %
## 5      5 ELINA...     5      14      48          32 62.1 %
## 6      6 KIKI ...    7      13      67          50 62.9 %
## 7      7 BELIN...     8      13      46          69 61.8 %
## 8      8 SEREN...     9       8      50          10 62.6 %
## 9      9 NAOMI...    10       7      67          14 63.8 %
```

Sports statistics scraping packages

- ◉ Tennis: deuce package (<https://github.com/skoval/deuce>)
- ◉ Cricket: cricketdata
(<https://github.com/ropenscilabs/cricketdata>)
- ◉ AFL: fitzRoy (<https://jimmyday12.github.io/fitzRoy/>)
- ◉ baseball: Lahman, pitchRx
- ◉ basketball: ballr
- ◉ soccer: <https://github.com/statsbomb/open-data>,
<https://github.com/JoGall/soccermatics>

deuce

```
# remotes::install_github("skoval/deuce")
library(deuce)
```

- ◉ Scraps data from <http://www.atpworldtour.com/>,
<https://www.flashscore.com/tennis>.
- ◉ Developed by a Tennis Australia data scientist Stephanie Kovalchik.

cricketdata

```
# remotes::install_github("ropenscilabs/cricketdata")
library(cricketdata)
```

- ◉ Scraps data from <https://docs.ropensci.org/cricketdata/>
- ◉ Developed by Rob Hyndman, Timothy Hyndman, Charles Gray, Sayani Gupta
- ◉ Interesting approach to getting the URLs for the data pages

cricketdata

[https://stats.espncricinfo.com/ci/engine/stats/index.html?
class=10;team=289;template=results?type=batting](https://stats.espncricinfo.com/ci/engine/stats/index.html?class=10;team=289;template=results?type=batting)

```
auswt20 <- fetch_cricinfo("T20", "Women", country="Aust")
auswt20

## # A tibble: 53 x 17
##   Player Start   End Matches Innings NotOuts   Runs HighScore HighScoreNo
##   <chr>  <int> <int>    <int>    <int>    <int>    <dbl>    <lgl>
## 1 MM La... 2010  2020     104      98      21    2788    133 TRUE
## 2 AJ He... 2010  2020     112      97      16    2060    148 TRUE
## 3 BL Mo... 2016  2020     52       49      11    1452    117 TRUE
## 4 EJ Vi... 2009  2018     62       58      10    1369     90 TRUE
## 5 AJ Bl... 2005  2017     95       81      19    1314     61 FALSE
## 6 EA Pe... 2008  2020     120      72      29    1218     60 TRUE
## 7 JE Du... 2009  2015     64       55      10    941      68 TRUE
## 8 LJ Po... 2006  2012     40       40      2     784      61 FALSE
## 9 S Nit... 2005  2011     36       35      2     776      56 FALSE
## 10 LC St... 2005  2013     54       50      14    769      52 FALSE
## # ... with 43 more rows, and 8 more variables: Average <dbl>, BallsFaced <
```

fitzRoy

```
# From CRAN  
# install.packages("fitzRoy")  
# or from GitHub  
# remotes::install_github("jimmyday12/fitzRoy")  
library(fitzRoy)  
aflw <- get_aflw_match_data(start_year = 2020)
```

- ◉ Gets statistics from <https://womens.afl> and <https://aftables.com>
- ◉ Developed by James Day, Robert Nguyen, Matthew Erbs, Oscar Lane, Jason Zivkovic
- ◉ Combination of scraping for men's data, and reading protected JSON data for women's by requesting a permission token



Beginners guide to html

[Search](#)[Menu](#)

[Home](#) → [Tutorials](#) → [HTML](#) →

HTML Beginner Tutorial

And so it begins! The HTML Beginner Tutorial assumes that you have absolutely no previous knowledge of HTML or CSS.

It should be easy to follow if you work through each page and then, to celebrate everything that's covered is brought together at the end

Home → Tutorials → CSS →

CSS Beginner Tutorial

I like the [HTML Beginner Tutorial](#) the **CSS Beginner Tutorial** assumes

**Fun interactive way to learn about css selectors
at <http://flukeout.github.io/>.**

[Tournaments](#)[Rankings](#)[Players](#)[News](#)[Videos](#)[More ▾](#)

Stats Hub SAP

2020 ▾ [Serve Stats](#) [Return Stats](#)

About WTA's Privacy and Cookie Policies

We use cookies to provide our services and for analytics and marketing. To find out more about our use of cookies and how you can disable them, please see our Privacy Policy. By continuing to browse our website, you agree to our use of cookies. [Click here to find out more info.](#)

[Accept Cookies](#)[X Only Essential Cookies](#)

ATP

Infosys 

Menu 

Watch Listen Shop Search  ATP TOUR PREMIER PARTNER

FedEx ATP RANKINGS Rankings Home Singles Doubles Race To London Doubles Race Race to Milan More ▾

2020.03.16 Top 100 All Countries Go 

Ranking ^	Move ^	Country ^	Player ^	Age ^	Points ^	Tourn Played ^	Points Dropping ^	Next Best ^
1	-		Novak Djokovic	32	10,220	18	45	0
2	-		Rafael Nadal	33	9,850	18	360	0
3	-		Dominic Thiem	26	7,045	21	1,000	90
4	-		Roger Federer	38	6,630	16	600	0
5	-		Daniil Medvedev	24	5,890	23	45	45

Scores Latest

ATP terms and conditions

Infosys



Terms & Conditions

Please read this Terms of Use Agreement ("the Agreement") carefully before using ATPTour.com, its affiliated sites and mobile products and services (the "Web site"). By visiting or using the Web site or any mobile product or service in any way, or linking to the Web site, you are entering into an agreement with ATP Tour, Inc. ("ATP"), owners of ATPTour.com its affiliated sites and mobile products and services. At all times, you are bound by the then-current version of this Agreement and all applicable laws. ATP highly recommends that you review this Agreement from time to time to ensure that you are familiar with the most recent version as ATP reserves the right to change these terms and conditions at any time without notice.

2. ENTIRE AGREEMENT

This Agreement constitutes the entire agreement between you and ATP concerning their subject matter. If any portion of this Agreement is deemed unlawful, invalid, or unenforceable, then such portion of this Agreement shall be deemed severed herefrom and shall not affect the validity or enforceability of the remainder of this Agreement.

ATP may, from time to time, offer you the opportunity to use interactive services on the Website, such as the ability to vote in connection with certain events, enter into contests or sweepstakes (the "Contest(s)"). and/or participa

Scores
Latest