

# Order Statistics & Data Intake

Instructor - Soumya Mukherjee

Content Credit- Dr. Matthew Beckman and Olivia Beck

July 21, 2023

## Agenda

- Chapter 13 (Ranks) reading
- Lists
- Data Types
- Webscraping

### Chapter 13 (Ranks)

- `rank()` is pretty useful
- `row_number()` is too
- I think the DC Chapter is self-evident on this one, so I don't think we need to spend time on it in class. Read it through the weekend and I can clarify any doubts on Monday.

## Lists

Lists are the R objects which contain elements of different types like – numbers, strings, vectors and another list inside it. A list can also contain a matrix or a function as its elements. List is created using `list()` function.

[https://www.tutorialspoint.com/r/r\\_lists.htm](https://www.tutorialspoint.com/r/r_lists.htm) ([https://www.tutorialspoint.com/r/r\\_lists.htm](https://www.tutorialspoint.com/r/r_lists.htm))

- A list can have any number of elements
  - each element in the list can have any number of (inner) elements in it
  - use double square elements to access the elements
  - use the appropriate mechanisms to access the inner elements
- element do not have to be of the same type or the same length
- Compare and contrast lists and data frames
  - data frames are essentially columns of vectors of the same length

- each element can only have one thing (character or number) in it
  - we won't cover it in this class, but it is technically possible for a cell of data frame to contain another data frame (using `nest` ).

Hide

```
temp_list <- list(numbers = 1:10,  
                 letters = c("A", "B", "C"),  
                 words = c("These", "are", "words", "."),  
                 innerlist = list( inner.numbers = 100:200,  
                                   states = state.abb),  
                 innerframe = data.frame(inner.numbers = 1:26,  
                                          inner.letters = letters),  
                 innermatrix = matrix(1:20, nrow = 10, ncol = 2)  
                 )
```

Access the first element in the list

1. use double square brackets
2. if we know what it is called, we can use `$`

Hide

```
temp_list[[1]]
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Hide

```
temp_list$numbers
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Access the 3rd element of the 2nd element

Hide

```
temp_list[[2]][3]
```

```
[1] "C"
```

[Hide](#)

```
temp_list$letters[3]
```

```
[1] "C"
```

Access the list of states

[Hide](#)

```
temp_list[[4]][[2]]
```

```
[1] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "FL" "GA" "HI"  
[12] "ID" "IL" "IN" "IA" "KS" "KY" "LA" "ME" "MD" "MA" "MI"  
[23] "MN" "MS" "MO" "MT" "NE" "NV" "NH" "NJ" "NM" "NY" "NC"  
[34] "ND" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT"  
[45] "VT" "VA" "WA" "WV" "WI" "WY"
```

[Hide](#)

```
temp_list$innerlist$states
```

```
[1] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "FL" "GA" "HI"  
[12] "ID" "IL" "IN" "IA" "KS" "KY" "LA" "ME" "MD" "MA" "MI"  
[23] "MN" "MS" "MO" "MT" "NE" "NV" "NH" "NJ" "NM" "NY" "NC"  
[34] "ND" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT"  
[45] "VT" "VA" "WA" "WV" "WI" "WY"
```

Access the 24th state

[Hide](#)

```
temp_list[[4]][[2]][24]
```

```
[1] "MS"
```

Hide

```
temp_list$innerlist$states[24]
```

```
[1] "MS"
```

Access the inner letters

Hide

```
temp_list$innerframe$inner.letters
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"  
[14] "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

Hide

```
temp_list[[5]][ , 2]
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"  
[14] "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

Access the 8th inner letter

Hide

```
temp_list$innerframe$inner.letters[8]
```

```
[1] "h"
```

Hide

```
temp_list[[5]][ , 2][8]
```

```
[1] "h"
```

When you try to access things that aren't there you WILL NOT get an error. You will get an NULL or NA (depending on what level of the structure you are on )

Hide

```
temp_list$not_here      #Null
```

```
NULL
```

Hide

```
temp_list$numbers[56]   #NA
```

```
[1] NA
```

## A word about data structures...

- R accommodates many different sorts of data structures
- One natural way to differentiate many of them is to consider
  - **dimensionality** (e.g. 1d, 2d, ... N-d)
  - **heterogeneity** (e.g., can elements have different types within the object?)
- R doesn't have any 0d types... scalar numbers or strings are treated as vectors with length 1.
- `str()` function is great to learn about the structure of an object in R
- The 5 following data structures are among the most common (but there are others):

	Homogeneous	Heterogeneous
1-dimensional	<b>Atomic vector</b>	<i>List</i>
2-dimensional	Matrix	<b>Data Frame</b>
N-dimensional	Array	

# More on data types

- variables (vectors) can be classified with different types as well
  - factors
  - character vectors
  - numeric
  - character
  - POSIXct (use `lubridate` package)
- mixed variables are automatically coerced to the most flexible type:
  - logical (e.g. `TRUE` ; `FASLE` ) is **least** flexible
  - integer (e.g., `-20` , `0` , `406` )
  - double (e.g. `3.14159` , `-2.17` , `1` , `0` )
  - character (e.g. `as;lke` , `3.14159` , `TRUE` ) is the **most** flexible type
- a “factor” is an important type of vector that may contain only predefined values, and is used to store categorical data

## Chapter 16 (Data Scraping & Cleaning–Data Intake)

- There are a ton of ways to get data into R (often with dedicated packages)
  - CSV (comma-separated-values) is a really common format
    - Lots of software export to CSV
    - many functions to read CSV's into R (e.g., we've seen `read_csv( )` from `readr` package)
    - `file.choose()` is handy to get file paths
  - R can handle lots of proprietary formats too (e.g., `foreign` package)
  - R can query relational databases like MS Access, Oracle, SAP, MySQL, etc (e.g, `rodbc` package)
  - Scraping web data

## Scraping Pole Vault Records from Wikipedia

Let's say we want to scrape pole vault World Records from Wikipedia...

[https://en.wikipedia.org/wiki/Men%27s\\_pole\\_vault\\_world\\_record\\_progression](https://en.wikipedia.org/wiki/Men%27s_pole_vault_world_record_progression)

([https://en.wikipedia.org/wiki/Men%27s\\_pole\\_vault\\_world\\_record\\_progression](https://en.wikipedia.org/wiki/Men%27s_pole_vault_world_record_progression))

# What's a pole vault?

It's an event in track and field competitions in which the athlete attempts the following (crudely speaking):

- Run as fast as possible while carrying a very long pole
- Jam the pole into a box in the ground
- Use the momentum to launch yourself as high as possible into the air
- Land safely on a huge cushion

Athletes repeat this as many times as they can while moving the crossbar up higher and higher.

It looks like this when it goes (extremely) well...

<https://www.youtube.com/watch?v=OAVNb2N7ntM> (<https://www.youtube.com/watch?v=OAVNb2N7ntM>)

...but sometimes turns out like this

<https://www.youtube.com/watch?v=iN-rWSM0ZzM> (<https://www.youtube.com/watch?v=iN-rWSM0ZzM>)

## Scraping Pole Vault Records from Wikipedia

Let's say we want to "scrape" pole vault world records from Wikipedia...

Here's the webpage: [https://en.wikipedia.org/wiki/Men%27s\\_pole\\_vault\\_world\\_record\\_progression](https://en.wikipedia.org/wiki/Men%27s_pole_vault_world_record_progression)  
([https://en.wikipedia.org/wiki/Men%27s\\_pole\\_vault\\_world\\_record\\_progression](https://en.wikipedia.org/wiki/Men%27s_pole_vault_world_record_progression))

## Steps to scrape HTML data

1. Locate webpage
2. Identify data table(s) to scrape
3. Edit the R code chunk shown to paste webpage URL with quotes around it as shown.
4. Execute the code chunk to scrape all HTML tables found on the page into a "list" object in the R environment called `table_list` here

```
library("rvest")

webpage <- "page_url"

table_list <- webpage %>%
  read_html(header = TRUE) %>%
  html_nodes(css = "table") %>%
  html_table(fill = TRUE)

str(table_list)
```

## Scraping Pole Vault Records from Wikipedia

Using our handy template, we replace the `page_url`

Hide

```
webpage <- "https://en.wikipedia.org/wiki/Men%27s_pole_vault_world_record_progression"

table_list <-
  webpage %>%
  read_html() %>%
  html_nodes(css = "table") %>%
  html_table(fill = TRUE)

str(table_list) # looks like a bit of a mess if you are new to this
```



```
$ : tibble [4 × 2] (S3: tbl_df/tbl/data.frame)
..$ X1: logi [1:4] NA NA NA NA
..$ X2: chr [1:4] "Ratified" "Not ratified" "Ratified but later rescinded" "Pending ratification"
$ : tibble [78 × 6] (S3: tbl_df/tbl/data.frame)
..$ Mark : chr [1:78] "4.02 m (.mw-parser-output .frac{white-space:nowrap}.mw-parser-output .frac .num,.mw-parser-output .frac .den{fo|__truncated__ "4.09 m (13 ft 5 in)" "4.12 m (13 ft 6 in)" "4.21 m (13 ft 9+1/2 in)" ...
..$ Athlete: chr [1:78] "Marc Wright" "Frank Foss" "Charles Hoff" "Charles Hoff" ...
..$ Nation : chr [1:78] "United States" "United States" "Norway" "Norway" ...
..$ Venue : chr [1:78] "Cambridge, U.S." "Antwerp, Belgium" "Copenhagen, Denmark" "Copenhagen, Denmark" ...
..$ Date : chr [1:78] "June 8, 1912[1]" "August 20, 1920[1]" "September 22, 1922[1]" "July 22, 1923[1]" ...
..$ #[4] : int [1:78] 1 1 1 2 3 4 1 1 1 1 ...
$ : tibble [12 × 20] (S3: tbl_df/tbl/data.frame)
..$ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteAthletics record progressions: chr [1:12] "World" "Sprinting" "Middle distance" "Long distance" ...
..$ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteAthletics record progressions: chr [1:12] "Sprinting\50 metres\60 metres\80 metres\100 metres\110 metres\120 metres\130 metres\140 metres\150 metres\160 metres\170 metres\180 metres\190 metres\200 metres\210 metres\220 metres\230 metres\240 metres\250 metres\260 metres\270 metres\280 metres\290 metres\300 metres\310 metres\320 metres\330 metres\340 metres\350 metres\360 metres\370 metres\380 metres\390 metres\400 metres\410 metres\420 metres\430 metres\440 metres\450 metres\460 metres\470 metres\480 metres\490 metres\500 metres\510 metres\520 metres\530 metres\540 metres\550 metres\560 metres\570 metres\580 metres\590 metres\600 metres\610 metres\620 metres\630 metres\640 metres\650 metres\660 metres\670 metres\680 metres\690 metres\700 metres\710 metres\720 metres\730 metres\740 metres\750 metres\760 metres\770 metres\780 metres\790 metres\800 metres\810 metres\820 metres\830 metres\840 metres\850 metres\860 metres\870 metres\880 metres\890 metres\900 metres\910 metres\920 metres\930 metres\940 metres\950 metres\960 metres\970 metres\980 metres\990 metres\1000 metres\1010 metres\1020 metres\1030 metres\1040 metres\1050 metres\1060 metres\1070 metres\1080 metres\1090 metres\1100 metres\1110 metres\1120 metres\1130 metres\1140 metres\1150 metres\1160 metres\1170 metres\1180 metres\1190 metres\1200 metres\1210 metres\1220 metres\1230 metres\1240 metres\1250 metres\1260 metres\1270 metres\1280 metres\1290 metres\1300 metres\1310 metres\1320 metres\1330 metres\1340 metres\1350 metres\1360 metres\1370 metres\1380 metres\1390 metres\1400 metres\1410 metres\1420 metres\1430 metres\1440 metres\1450 metres\1460 metres\1470 metres\1480 metres\1490 metres\1500 metres\1510 metres\1520 metres\1530 metres\1540 metres\1550 metres\1560 metres\1570 metres\1580 metres\1590 metres\1600 metres\1610 metres\1620 metres\1630 metres\1640 metres\1650 metres\1660 metres\1670 metres\1680 metres\1690 metres\1700 metres\1710 metres\1720 metres\1730 metres\1740 metres\1750 metres\1760 metres\1770 metres\1780 metres\1790 metres\1800 metres\1810 metres\1820 metres\1830 metres\1840 metres\1850 metres\1860 metres\1870 metres\1880 metres\1890 metres\1900 metres\1910 metres\1920 metres\1930 metres\1940 metres\1950 metres\1960 metres\1970 metres\1980 metres\1990 metres\2000 metres\2010 metres\2020 metres\2030 metres\2040 metres\2050 metres\2060 metres\2070 metres\2080 metres\2090 metres\2100 metres\2110 metres\2120 metres\2130 metres\2140 metres\2150 metres\2160 metres\2170 metres\2180 metres\2190 metres\2200 metres\2210 metres\2220 metres\2230 metres\2240 metres\2250 metres\2260 metres\2270 metres\2280 metres\2290 metres\2300 metres\2310 metres\2320 metres\2330 metres\2340 metres\2350 metres\2360 metres\2370 metres\2380 metres\2390 metres\2400 metres\2410 metres\2420 metres\2430 metres\2440 metres\2450 metres\2460 metres\2470 metres\2480 metres\2490 metres\2500 metres\2510 metres\2520 metres\2530 metres\2540 metres\2550 metres\2560 metres\2570 metres\2580 metres\2590 metres\2600 metres\2610 metres\2620 metres\2630 metres\2640 metres\2650 metres\2660 metres\2670 metres\2680 metres\2690 metres\2700 metres\2710 metres\2720 metres\2730 metres\2740 metres\2750 metres\2760 metres\2770 metres\2780 metres\2790 metres\2800 metres\2810 metres\2820 metres\2830 metres\2840 metres\2850 metres\2860 metres\2870 metres\2880 metres\2890 metres\2900 metres\2910 metres\2920 metres\2930 metres\2940 metres\2950 metres\2960 metres\2970 metres\2980 metres\2990 metres\3000 metres\3010 metres\3020 metres\3030 metres\3040 metres\3050 metres\3060 metres\3070 metres\3080 metres\3090 metres\3100 metres\3110 metres\3120 metres\3130 metres\3140 metres\3150 metres\3160 metres\3170 metres\3180 metres\3190 metres\3200 metres\3210 metres\3220 metres\3230 metres\3240 metres\3250 metres\3260 metres\3270 metres\3280 metres\3290 metres\3300 metres\3310 metres\3320 metres\3330 metres\3340 metres\3350 metres\3360 metres\3370 metres\3380 metres\3390 metres\3400 metres\3410 metres\3420 metres\3430 metres\3440 metres\3450 metres\3460 metres\3470 metres\3480 metres\3490 metres\3500 metres\3510 metres\3520 metres\3530 metres\3540 metres\3550 metres\3560 metres\3570 metres\3580 metres\3590 metres\3600 metres\3610 metres\3620 metres\3630 metres\3640 metres\3650 metres\3660 metres\3670 metres\3680 metres\3690 metres\3700 metres\3710 metres\3720 metres\3730 metres\3740 metres\3750 metres\3760 metres\3770 metres\3780 metres\3790 metres\3800 metres\3810 metres\3820 metres\3830 metres\3840 metres\3850 metres\3860 metres\3870 metres\3880 metres\3890 metres\3900 metres\3910 metres\3920 metres\3930 metres\3940 metres\3950 metres\3960 metres\3970 metres\3980 metres\3990 metres\4000 metres\4010 metres\4020 metres\4030 metres\4040 metres\4050 metres\4060 metres\4070 metres\4080 metres\4090 metres\4100 metres\4110 metres\4120 metres\4130 metres\4140 metres\4150 metres\4160 metres\4170 metres\4180 metres\4190 metres\4200 metres\4210 metres\4220 metres\4230 metres\4240 metres\4250 metres\4260 metres\4270 metres\4280 metres\4290 metres\4300 metres\4310 metres\4320 metres\4330 metres\4340 metres\4350 metres\4360 metres\4370 metres\4380 metres\4390 metres\4400 metres\4410 metres\4420 metres\4430 metres\4440 metres\4450 metres\4460 metres\4470 metres\4480 metres\4490 metres\4500 metres\4510 metres\4520 metres\4530 metres\4540 metres\4550 metres\4560 metres\4570 metres\4580 metres\4590 metres\4600 metres\4610 metres\4620 metres\4630 metres\4640 metres\4650 metres\4660 metres\4670 metres\4680 metres\4690 metres\4700 metres\4710 metres\4720 metres\4730 metres\4740 metres\4750 metres\4760 metres\4770 metres\4780 metres\4790 metres\4800 metres\4810 metres\4820 metres\4830 metres\4840 metres\4850 metres\4860 metres\4870 metres\4880 metres\4890 metres\4900 metres\4910 metres\4920 metres\4930 metres\4940 metres\4950 metres\4960 metres\4970 metres\4980 metres\4990 metres\5000 metres\5010 metres\5020 metres\5030 metres\5040 metres\5050 metres\5060 metres\5070 metres\5080 metres\5090 metres\5100 metres\5110 metres\5120 metres\5130 metres\5140 metres\5150 metres\5160 metres\5170 metres\5180 metres\5190 metres\5200 metres\5210 metres\5220 metres\5230 metres\5240 metres\5250 metres\5260 metres\5270 metres\5280 metres\5290 metres\5300 metres\5310 metres\5320 metres\5330 metres\5340 metres\5350 metres\5360 metres\5370 metres\5380 metres\5390 metres\5400 metres\5410 metres\5420 metres\5430 metres\5440 metres\5450 metres\5460 metres\5470 metres\5480 metres\5490 metres\5
```

```

: chr [1:12] "Middle distance" NA NA NA ...
..$
: chr [1:12] "800 metres\n1000 metres\n1500 metres\nMile run\n2000 metres\n3000 metres\nmen\nwomen" NA NA NA ...
..$
: chr [1:12] "Long distance" NA NA NA ...
..$
: chr [1:12] "5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf marathon\nMarathon\n50K\n100K" NA NA NA ...
..$
: chr [1:12] "Hurdles" NA NA NA ...
..$
: chr [1:12] "50 metres hurdles\n60 metres hurdles\nWomen's 80 metres hurdles\n110/100 metres hurdles\nmen\nwomen\n400 metre
s"| __truncated__ NA NA NA ...
..$
: chr [1:12] "Relay" NA NA NA ...
..$
: chr [1:12] "4 x 100 metres\nmen\nwomen\n4 x 200 metres\nmen\nwomen\n4 x 400 metres\nmen\nwomen\nmixed\n4 x 800 metres\nmen
\n"| __truncated__ NA NA NA ...
..$
: chr [1:12] "Walking" NA NA NA ...
..$
: chr [1:12] "10 km\nmen\nwomen\n20,000 metres (track)\nmen\nwomen\n20 km (road)\nmen\nwomen\n35 km\nmen\nwomen\n50 km\nmen
\nwomen" NA NA NA ...
..$
: chr [1:12] "Jumping" NA NA NA ...
..$
: chr [1:12] "High jump\nmen outdoor\nmen indoor\nwomen\nLong jump\nmen\nwomen\nTriple jump\nPole vault\nmen\nmen indoor\nwo
m"| __truncated__ NA NA NA ...
..$
: chr [1:12] "Throwing" NA NA NA ...
..$
: chr [1:12] "Shot put\nmen\nwomen\nDiscus\nmen\nwomen\nHammer\nmen\nwomen\nJavelin\nmen\nwomen" NA NA NA ...
..$
: chr [1:12] "Combined events" NA NA NA ...
..$
: chr [1:12] "Decathlon\nHeptathlon\nmen\nwomen\nPentathlon" NA NA NA ...
$ : tibble [9 x 2] (S3: tbl_df/tbl/data.frame)
..$ X1: chr [1:9] "Sprinting" "Middle distance" "Long distance" "Hurdles" ...
..$ X2: chr [1:9] "50 metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nwome

```

```

n" "800 metres\n1000 metres\n1500 metres\nMile run\n2000 metres\n3000 metres\nmen\nwomen" "5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf marathon\nMarathon\n50K\n100K" "50 metres hurdles\n60 metres hurdles\nWomen's 80 metres hurdles\n110/100 metres hurdles\nmen\nwomen\n400 metres"| __truncated__ ...
$ : tibble [19 × 12] (S3: tbl_df/tbl/data.frame)
  ..$ vteRecords in athletics: chr [1:19] "World records\nWorld U23\nWorld U20\nWorld U18\nWorld masters (centenarian)\nWorld IPC\nWorld deaf" "Area records" "Senior" "Under-23" ...
  ..$ vteRecords in athletics: chr [1:19] "World records\nWorld U23\nWorld U20\nWorld U18\nWorld masters (centenarian)\nWorld IPC\nWorld deaf" "Senior\nAfrica\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth AmericaUnder-23\nAfrican U2"| __truncated__ "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" "African U23\nAsian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceania U23\nSouth American U23" ...
  ..$ : chr [1:19] NA "Senior" NA NA ...
  ..$ : chr [1:19] NA "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" NA NA ...
  ..$ : chr [1:19] NA "Under-23" NA NA ...
  ..$ : chr [1:19] NA "African U23\nAsian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceania U23\nSouth American U23" NA NA ...
  ..$ : chr [1:19] NA "Junior (U-20)" NA NA ...
  ..$ : chr [1:19] NA "African U20\nAsian U20\nCAC U20\nEuropean U20\nNorth, Central American and Caribbean U20\nOceania U20\nSouth American U20" NA NA ...
  ..$ : chr [1:19] NA "Youth (U-18)" NA NA ...
  ..$ : chr [1:19] NA "African Youth\nAsian Youth\nCAC Youth\nEuropean Youth\nNorth, Central American and Caribbean Youth\nOceania Yo"| __truncated__ NA NA ...
  ..$ : chr [1:19] NA "Others" NA NA ...
  ..$ : chr [1:19] NA "Baltic\nCentral American and Caribbean\nCommonwealth\nNorth America\nOECS\nPan america" NA NA ...
$ : tibble [5 × 2] (S3: tbl_df/tbl/data.frame)
  ..$ X1: chr [1:5] "Senior" "Under-23" "Junior (U-20)" "Youth (U-18)" ...
  ..$ X2: chr [1:5] "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" "African U23\nAsian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceania U23\nSouth American U23" "African U20\nAsian U20\nCAC U20\nEuropean U20\nNorth, Central American and Caribbean U20\nOceania U20\nSouth American U20" "African Youth\nAsian Youth\nCAC Youth\nEuropean Youth\nNorth, Central American and Caribbean Youth\nOceania Yo"| __truncated__ ...
$ : tibble [4 × 2] (S3: tbl_df/tbl/data.frame)
  ..$ X1: chr [1:4] "North, Central America and Caribbean" "Central America and Caribbean" "Central America" "South America"
  ..$ X2: chr [1:4] "NACAC Championships\nNACAC U23 Championships\nNACAC U20 Championships\nNACAC U18 Championships" "CAC Championships\nCAC Games\nCAC Junior and Youth Championships\nCAC Age Group Championships" "Central American Championships\nCe

```

```
ntal American Games\nCentral American Junior and Youth Championships" "South American Championships\nSouth American Indoor Championships\nSouth American Games\nSouth American Under-2"| __truncated__
```

# Scraping Pole Vault Records from Wikipedia

Now we can use the data to answer lots of interesting questions

- RQ: Which nation has broken the record most frequently?
- RQ: Which athlete has broken the record most frequently?
- RQ: Which venue has seen the most record-breaking performances?

We'll learn additional tools (e.g., Regular Expressions) in coming weeks that will allow us to parse the text strings like `Record` or `Date` for further analysis

Hide

```
# Look at the structure (look for how many tables are in the list; verify they are "data.frame" format)
str(table_list)
```

List of 7

```
$ : tibble [4 × 2] (S3: tbl_df/tbl/data.frame)
```

```
..$ X1: logi [1:4] NA NA NA NA
```

```
..$ X2: chr [1:4] "Ratified" "Not ratified" "Ratified but later rescinded" "Pending ratification"
```

```
$ : tibble [78 × 6] (S3: tbl_df/tbl/data.frame)
```

```
..$ Mark : chr [1:78] "4.02 m (.mw-parser-output .frac{white-space:nowrap}.mw-parser-output .frac .num,.mw-parser-output .frac .den{fo}|__truncated__ "4.09 m (13 ft 5 in)" "4.12 m (13 ft 6 in)" "4.21 m (13 ft 9+1/2 in)" ...
```

```
..$ Athlete: chr [1:78] "Marc Wright" "Frank Foss" "Charles Hoff" "Charles Hoff" ...
```

```
..$ Nation : chr [1:78] "United States" "United States" "Norway" "Norway" ...
```

```
..$ Venue : chr [1:78] "Cambridge, U.S." "Antwerp, Belgium" "Copenhagen, Denmark" "Copenhagen, Denmark" ...
```

```
..$ Date : chr [1:78] "June 8, 1912[1]" "August 20, 1920[1]" "September 22, 1922[1]" "July 22, 1923[1]" ...
```

```
..$ #[4] : int [1:78] 1 1 1 2 3 4 1 1 1 1 ...
```

```
$ : tibble [12 × 20] (S3: tbl_df/tbl/data.frame)
```

```
..$ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteAthletics record progressions: chr [1:12] "World" "Sprinting" "Middle distance" "Long distance" ...
```

```
..$ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteAthletics record progressions: chr [1:12] "Sprinting\n50 metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nwomen\n500 metres\n600 metres\nmen\nwomen\n800 metres\n1000 metres\n1500 metres\n1 mile run\n2000 metres\n3000 metres\nmen\nwomen\n5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf marathon\nMarathon\n50K\n100K" ...
```

```
..$
```

```
: chr [1:12] "Sprinting" NA NA NA ...
```

```
..$
```

```
: chr [1:12] "50 metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nwomen" NA NA NA ...
```

```
..$
```

```

: chr [1:12] "Middle distance" NA NA NA ...
..$
: chr [1:12] "800 metres\n1000 metres\n1500 metres\nMile run\n2000 metres\n3000 metres\nmen\nwomen" NA NA NA ...
..$
: chr [1:12] "Long distance" NA NA NA ...
..$
: chr [1:12] "5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf marathon\nMarathon\n50K\n100K" NA NA NA ...
..$
: chr [1:12] "Hurdles" NA NA NA ...
..$
: chr [1:12] "50 metres hurdles\n60 metres hurdles\nWomen's 80 metres hurdles\n110/100 metres hurdles\nmen\nwomen\n400 metre
s"| __truncated__ NA NA NA ...
..$
: chr [1:12] "Relay" NA NA NA ...
..$
: chr [1:12] "4 x 100 metres\nmen\nwomen\n4 x 200 metres\nmen\nwomen\n4 x 400 metres\nmen\nwomen\nmixed\n4 x 800 metres\nmen
\n"| __truncated__ NA NA NA ...
..$
: chr [1:12] "Walking" NA NA NA ...
..$
: chr [1:12] "10 km\nmen\nwomen\n20,000 metres (track)\nmen\nwomen\n20 km (road)\nmen\nwomen\n35 km\nmen\nwomen\n50 km\nmen
\nwomen" NA NA NA ...
..$
: chr [1:12] "Jumping" NA NA NA ...
..$
: chr [1:12] "High jump\nmen outdoor\nmen indoor\nwomen\nLong jump\nmen\nwomen\nTriple jump\nPole vault\nmen\nmen indoor\nwo
m"| __truncated__ NA NA NA ...
..$
: chr [1:12] "Throwing" NA NA NA ...
..$
: chr [1:12] "Shot put\nmen\nwomen\nDiscus\nmen\nwomen\nHammer\nmen\nwomen\nJavelin\nmen\nwomen" NA NA NA ...
..$
: chr [1:12] "Combined events" NA NA NA ...
..$
: chr [1:12] "Decathlon\nHeptathlon\nmen\nwomen\nPentathlon" NA NA NA ...
$ : tibble [9 x 2] (S3: tbl_df/tbl/data.frame)
..$ X1: chr [1:9] "Sprinting" "Middle distance" "Long distance" "Hurdles" ...
..$ X2: chr [1:9] "50 metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nwome

```

```

n" "800 metres\n1000 metres\n1500 metres\nMile run\n2000 metres\n3000 metres\nmen\nwomen" "5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf marathon\nMarathon\n50K\n100K" "50 metres hurdles\n60 metres hurdles\nWomen's 80 metres hurdles\n110/100 metres hurdles\nmen\nwomen\n400 metres"| __truncated__ ...
$ : tibble [19 × 12] (S3: tbl_df/tbl/data.frame)
  ..$ vteRecords in athletics: chr [1:19] "World records\nWorld U23\nWorld U20\nWorld U18\nWorld masters (centenarian)\nWorld IPC\nWorld deaf" "Area records" "Senior" "Under-23" ...
  ..$ vteRecords in athletics: chr [1:19] "World records\nWorld U23\nWorld U20\nWorld U18\nWorld masters (centenarian)\nWorld IPC\nWorld deaf" "Senior\nAfrica\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth AmericaUnder-23\nAfrican U2"| __truncated__ "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" "African U23\nAsian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceania U23\nSouth American U23" ...
  ..$ : chr [1:19] NA "Senior" NA NA ...
  ..$ : chr [1:19] NA "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" NA NA ...
  ..$ : chr [1:19] NA "Under-23" NA NA ...
  ..$ : chr [1:19] NA "African U23\nAsian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceania U23\nSouth American U23" NA NA ...
  ..$ : chr [1:19] NA "Junior (U-20)" NA NA ...
  ..$ : chr [1:19] NA "African U20\nAsian U20\nCAC U20\nEuropean U20\nNorth, Central American and Caribbean U20\nOceania U20\nSouth American U20" NA NA ...
  ..$ : chr [1:19] NA "Youth (U-18)" NA NA ...
  ..$ : chr [1:19] NA "African Youth\nAsian Youth\nCAC Youth\nEuropean Youth\nNorth, Central American and Caribbean Youth\nOceania Yo"| __truncated__ NA NA ...
  ..$ : chr [1:19] NA "Others" NA NA ...
  ..$ : chr [1:19] NA "Baltic\nCentral American and Caribbean\nCommonwealth\nNorth America\nOECS\nPan america" NA NA ...
$ : tibble [5 × 2] (S3: tbl_df/tbl/data.frame)
  ..$ X1: chr [1:5] "Senior" "Under-23" "Junior (U-20)" "Youth (U-18)" ...
  ..$ X2: chr [1:5] "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" "African U23\nAsian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceania U23\nSouth American U23" "African U20\nAsian U20\nCAC U20\nEuropean U20\nNorth, Central American and Caribbean U20\nOceania U20\nSouth American U20" "African Youth\nAsian Youth\nCAC Youth\nEuropean Youth\nNorth, Central American and Caribbean Youth\nOceania Yo"| __truncated__ ...
$ : tibble [4 × 2] (S3: tbl_df/tbl/data.frame)
  ..$ X1: chr [1:4] "North, Central America and Caribbean" "Central America and Caribbean" "Central America" "South America"
  ..$ X2: chr [1:4] "NACAC Championships\nNACAC U23 Championships\nNACAC U20 Championships\nNACAC U18 Championships" "CAC Championships\nCAC Games\nCAC Junior and Youth Championships\nCAC Age Group Championships" "Central American Championships\nCe

```

```
ntral American Games\nCentral American Junior and Youth Championships" "South American Championships\nSouth American Indoor Championships\nSouth American Games\nSouth American Under-2"| __truncated__
```

Hide

```
# Inspect the first table in the list (IAAF Men from the Wikipedia Page)
PVrecords <- table_list[[2]]
head(PVrecords)
```

### Mark

<chr>

4.02 m (.mw-parser-output .frac{white-space:nowrap}.mw-parser-output .frac .num,.mw-parser-output .frac .den{font-size:80%;line-height:0;vertical-align:super}.mw-parser-output .frac .den{vertical-align:sub}.mw-parser-output .sr-only{border:0;clip:rect(0,0,0,0);clip-path:polygon(0px 0px,0px 0px,0px 0px);height:1px;margin:-1px;overflow:hidden;padding:0;position:absolute;width:1px}13 ft 2+1<e2>\u0081\u00844 in)

4.09 m (13 ft 5 in)

4.12 m (13 ft 6 in)

4.21 m (13 ft 9+1<e2>\u0081\u00842 in)

4.23 m (13 ft 10+1<e2>\u0081\u00842 in)

4.25 m (13 ft 11+1<e2>\u0081\u00844 in)

6 rows | 1-1 of 6 columns

## Pole Vault Records from Wikipedia

Maybe we plot the density of records?

- what would low density mean?
- what would high density mean?

Hide



```
PVRecordsData <-
  PVrecords %>%
  mutate(Record_m = parse_number(Mark)) %>%
  select(Mark, Record_m)

head(PVRecordsData)
```

## Mark

<chr>

4.02 m (.mw-parser-output .frac{white-space:nowrap}.mw-parser-output .frac .num,.mw-parser-output .frac .den{font-size:80%;line-height:0;vertical-align:super}.mw-parser-output .frac .den{vertical-align:sub}.mw-parser-output .sr-only{border:0;clip:rect(0,0,0,0);clip-path:polygon(0px 0px,0px 0px,0px 0px);height:1px;margin:-1px;overflow:hidden;padding:0;position:absolute;width:1px}13 ft 2+1<e2>\u0081\u00844 in)

4.09 m (13 ft 5 in)

4.12 m (13 ft 6 in)

4.21 m (13 ft 9+1<e2>\u0081\u00842 in)

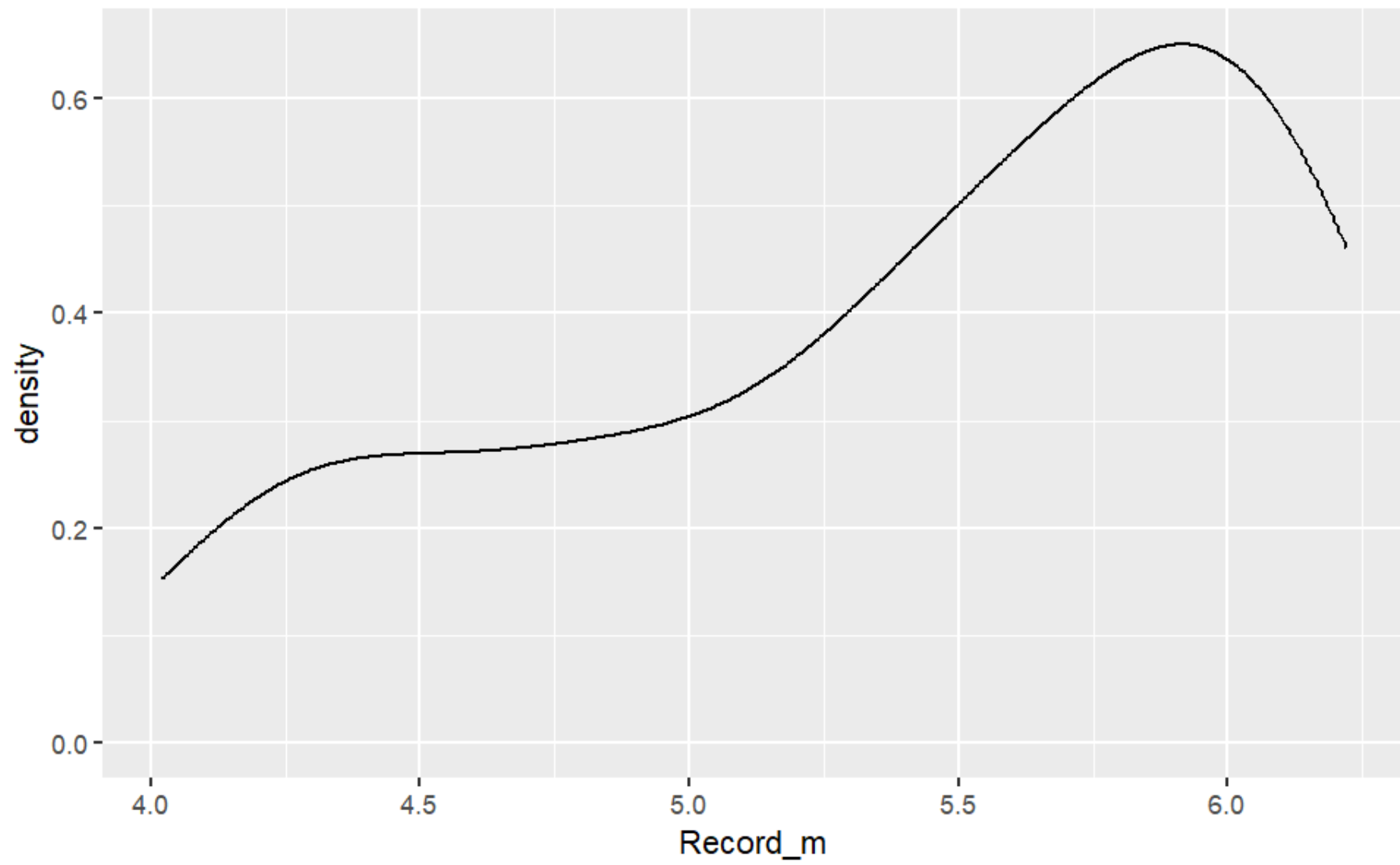
4.23 m (13 ft 10+1<e2>\u0081\u00842 in)

4.25 m (13 ft 11+1<e2>\u0081\u00844 in)

6 rows | 1-1 of 2 columns

Hide

```
PVRecordsData %>%
  ggplot(aes(x = Record_m)) +
  geom_density()
```



Hide

```
round(PVRecordsData$Record_m) %>%  
  table()
```

```
.  
4 5 6  
11 26 41
```

## Penn State Football Receiving Statistics

1. Google Penn State Football Statistics
2. Edit the R code chunk shown to paste `webpage` URL with quotes around it as shown.
3. Execute the code chunk to scrape all HTML tables found on the page into a “list” object in the R environment called `Tables` here
4. Identify a data table from the source (for example, “receiving statistics”) and find it in the list object in your R environment

```
library("rvest")  
page <- "http://www.espn.com/college-football/team/stats/_/id/213/penn-state-nittany-lions"  
  
Tables <- page %>%  
  read_html(header = TRUE) %>%  
  html_nodes(css = "table") %>%  
  html_table(fill = TRUE)
```

```
Tables[[1]]
```

## Penn State Football Receiving Statistics

Hide

```
url <- "http://www.espn.com/college-football/team/stats/_/id/213/penn-state-nittany-lions"  
  
PlayerStats <- url %>%  
  read_html() %>%  
  html_nodes(css = "table") %>%  
  html_table(fill = TRUE)
```

Hide

```
# R stores the result as a "list" object, so the double square brackets select an
#   element of the list, and we store it as a data frame
```

```
ReceivingRaw <- PlayerStats[[6]]
```

```
# Inspect the Data Table
ReceivingRaw
```

REC YDS		AVG	LNG	TD
<int>	<chr>	<dbl>	<int>	<int>
46	611	13.3	58	2
51	577	11.3	34	5
24	389	16.2	88	4
32	362	11.3	67	5
20	328	16.4	48	4
19	273	14.4	48	1
20	188	9.4	45	1
10	123	12.3	38	3
8	89	11.1	20	0
11	85	7.7	22	1
1-10 of 21 rows			Previous	1 2 3 Next

Hide

```
# Add player names and remove totals
ReceivingStats <-
  bind_cols(PlayerStats[[5]], PlayerStats[[6]]) %>%
  filter(Name != "Total")

# Inspect FootballStatsClean
ReceivingStats
```

Name <chr>	REC <int>	YDS <chr>	AVG <dbl>	LNG <int>	TD <int>
Parker Washington WR	46	611	13.3	58	2
Mitchell Tinsley WR	51	577	11.3	34	5
KeAndre Lambert-Smith WR	24	389	16.2	88	4
Brenton Strange TE	32	362	11.3	67	5
Theo Johnson TE	20	328	16.4	48	4
Harrison Wallace III WR	19	273	14.4	48	1
Kaytron Allen RB	20	188	9.4	45	1
Tyler Warren TE	10	123	12.3	38	3
Liam Clifford WR	8	89	11.1	20	0
Nicholas Singleton RB	11	85	7.7	22	1
1-10 of 20 rows			Previous	1	2 Next

## Aside: XPath selector

- so far, we have been scraping every table in sight and then hunting through the results for the ones we want.
- you can scrape one specific table with an XPath selector
- some example code is below... basically you only need to change one line

- use `html_node()` (singular) rather than `html_nodes()` (plural)
- specify the XPath selector
- see the help documentation for `html_nodes()` to learn more about the syntax
- helpful instructions for getting the XPath to an element on a web page using Google Chrome browser: <http://www.r-bloggers.com/using-rvest-to-scrape-an-html-table/> (<http://www.r-bloggers.com/using-rvest-to-scrape-an-html-table/>)
- CSS selectors for single table also available through the `selectr` package, which is a port of the python `cssselect` library (see help documentation for `html_nodes()` )

Hide

```
library("rvest")

page_url <- "https://en.wikipedia.org/wiki/Mile_run_world_record_progression"
XPATH <- '//*[@id="mw-content-text"]/div/table'
cssPATH <- '#mw-content-text > div.mw-parser-output > table:nth-child(11)'

table_list_1 <-
  page_url %>%
  read_html() %>%
  html_node(xpath = XPATH) %>%
  html_table(fill = TRUE)

table_list_2 <-
  page_url %>%
  read_html() %>%
  html_node(css = cssPATH) %>%
  html_table(fill = TRUE)

table_list_1
```

Time <chr>	Athlete <chr>	Nationality <chr>	Date <chr>	Venue <chr>
4:28	Charles Westhall	United Kingdom	26 July 1855	London
4:28	Thomas Horspool	United Kingdom	28 September 1857	Manchester
4:23	Thomas Horspool	United Kingdom	12 July 1858	Manchester

Time <chr>	Athlete <chr>	Nationality <chr>	Date <chr>	Venue <chr>
4:221<e2>\u0081\u00844	Siah Albison	United Kingdom	27 October 1860	Manchester
4:213<e2>\u0081\u00844	William Lang	United Kingdom	11 July 1863	Manchester
4:201<e2>\u0081\u00842	Edward Mills	United Kingdom	23 April 1864	Manchester
4:20	Edward Mills	United Kingdom	25 June 1864	Manchester
4:171<e2>\u0081\u00844	William Lang	United Kingdom	19 August 1865	Manchester
4:171<e2>\u0081\u00844	William Richards	United Kingdom	19 August 1865	Manchester
4:161<e2>\u0081\u00845	William Cummings	United Kingdom	14 May 1881	Preston
1-10 of 11 rows			Previous	1 2 Next

Hide

table\_list\_2

Time <chr>	Athlete <chr>	Nationality <chr>	Date <chr>	Venue <chr>
4:28	Charles Westhall	United Kingdom	26 July 1855	London
4:28	Thomas Horspool	United Kingdom	28 September 1857	Manchester
4:23	Thomas Horspool	United Kingdom	12 July 1858	Manchester
4:221<e2>\u0081\u00844	Siah Albison	United Kingdom	27 October 1860	Manchester
4:213<e2>\u0081\u00844	William Lang	United Kingdom	11 July 1863	Manchester
4:201<e2>\u0081\u00842	Edward Mills	United Kingdom	23 April 1864	Manchester
4:20	Edward Mills	United Kingdom	25 June 1864	Manchester
4:171<e2>\u0081\u00844	William Lang	United Kingdom	19 August 1865	Manchester

Time <chr>	Athlete <chr>	Nationality <chr>	Date <chr>	Venue <chr>
4:171<e2>\u0081\u00844	William Richards	United Kingdom	19 August 1865	Manchester
4:161<e2>\u0081\u00845	William Cummings	United Kingdom	14 May 1881	Preston
1-10 of 11 rows				Previous 1 2 Next

# Assignments

- Reading Quiz Chapters 13 and 16 (due Tuesday July 25 9:59 am)