Code ▾

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)

- 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"
[11] "HI" "ID" "IL" "IN" "IA" "KS" "KY" "LA" "ME" "MD"
[21] "MA" "MI" "MN" "MS" "MO" "MT" "NE" "NV" "NH" "NJ"
[31] "NM" "NY" "NC" "ND" "OH" "OK" "OR" "PA" "RI" "SC"
[41] "SD" "TN" "TX" "UT" "VT" "VA" "WA" "WV" "WI" "WY"
```

Hide

```
temp_list$innerlist$states
```

```
 [1] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "FL" "GA"
[11] "HI" "ID" "IL" "IN" "IA" "KS" "KY" "LA" "ME" "MD"
[21] "MA" "MI" "MN" "MS" "MO" "MT" "NE" "NV" "NH" "NJ"
[31] "NM" "NY" "NC" "ND" "OH" "OK" "OR" "PA" "RI" "SC"
[41] "SD" "TN" "TX" "UT" "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"
[13] "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
[25] "y" "z"
```

Hide

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

```
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l"
[13] "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
[25] "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	**Atomic vector**	*List*
2-dimensional	Matrix	**Data Frame**

	Homogeneous	Heterogeneous
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;lkne` , `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)

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)

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
```

```
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:lef
t;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-s
pace:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-out
put .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-vari
ant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margi
n: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:lef
t;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-s
pace:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-out
put .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-vari
ant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margi
n:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteAthletics record progressions: chr [1:12] "Sprintin
g\n50 metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nw"| __truncated__ "50
metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nwomen" "800 metres\n1000 met
res\n1500 metres\nMile run\n2000 metres\n3000 metres\nmen\nwomen" "5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf m
arathon\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 N
A 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 × 100 metres\nmen\nwomen\n4 × 200 metres\nmen\nwomen\n4 × 400 metres\nmen\nwomen\nmixed\n4 × 800 metres\nmen
\"| __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 × 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\n1
0K\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)\nWorl
d 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)\nWorl
d IPC\nWorld deaf" "Senior\nAfrica\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth AmericaUnder-23\nAfr
ican U2"| __truncated__ "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" "African U23\n
Asian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceanian 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 Am
erica" 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 Car
ibbean U23\nOceanian 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 Car
ibbean U20\nOceanian 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\nOceanian 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\nAsia
n U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceanian U23\nSouth American U23" "African U20\nAsi
an U20\nCAC U20\nEuropean U20\nNorth, Central American and Caribbean U20\nOceanian U20\nSouth American U20" "African Youth\n
Asian Youth\nCAC Youth\nEuropean Youth\nNorth, Central American and Caribbean Youth\nOceanian 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 Americ
a"
  ..$ X2: chr [1:4] "NACAC Championships\nNACAC U23 Championships\nNACAC U20 Championships\nNACAC U18 Championships" "CAC Ch
ampionships\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__
```

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:lef
t;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-s
pace:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-out
put .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-vari
ant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margi
n: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:lef
t;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-s
pace:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-out
put .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-vari
ant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margi
n:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteAthletics record progressions: chr [1:12] "Sprintin
g\n50 metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nw"| __truncated__ "50
metres\n60 metres\nmen\nwomen\n100 metres\nmen\nwomen\n200 metres\nmen\nwomen\n400 metres\nmen\nwomen" "800 metres\n1000 met
res\n1500 metres\nMile run\n2000 metres\n3000 metres\nmen\nwomen" "5000 metres\n5K\n10,000 metres\n10K\nOne hour run\nHalf m
arathon\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 N
A 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 × 100 metres\nmen\nwomen\n4 × 200 metres\nmen\nwomen\n4 × 400 metres\nmen\nwomen\nmixed\n4 × 800 metres\nmen
\"| __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 × 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\n1
0K\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)\nWorl
d 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)\nWorl
d IPC\nWorld deaf" "Senior\nAfrica\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth AmericaUnder-23\nAfr
ican U2"| __truncated__ "Africa\nAsia\nEurope\nNorth, Central American and Caribbean\nOceania\nSouth America" "African U23\n
Asian U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceanian 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 Am
erica" 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 Car
ibbean U23\nOceanian 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 Car
ibbean U20\nOceanian 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\nOceanian 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\nAsia
n U23\nCAC U23\nEuropean U23\nNorth, Central American and Caribbean U23\nOceanian U23\nSouth American U23" "African U20\nAsi
an U20\nCAC U20\nEuropean U20\nNorth, Central American and Caribbean U20\nOceanian U20\nSouth American U20" "African Youth\n
Asian Youth\nCAC Youth\nEuropean Youth\nNorth, Central American and Caribbean Youth\nOceanian 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 Americ
a"
  ..$ X2: chr [1:4] "NACAC Championships\nNACAC U23 Championships\nNACAC U20 Championships\nNACAC U18 Championships" "CAC Ch
ampionships\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+¼ in)
4.09 m (13 ft 5 in)
4.12 m (13 ft 6 in)
4.21 m (13 ft 9+½ in)
4.23 m (13 ft 10+½ in)
4.25 m (13 ft 11+¼ 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

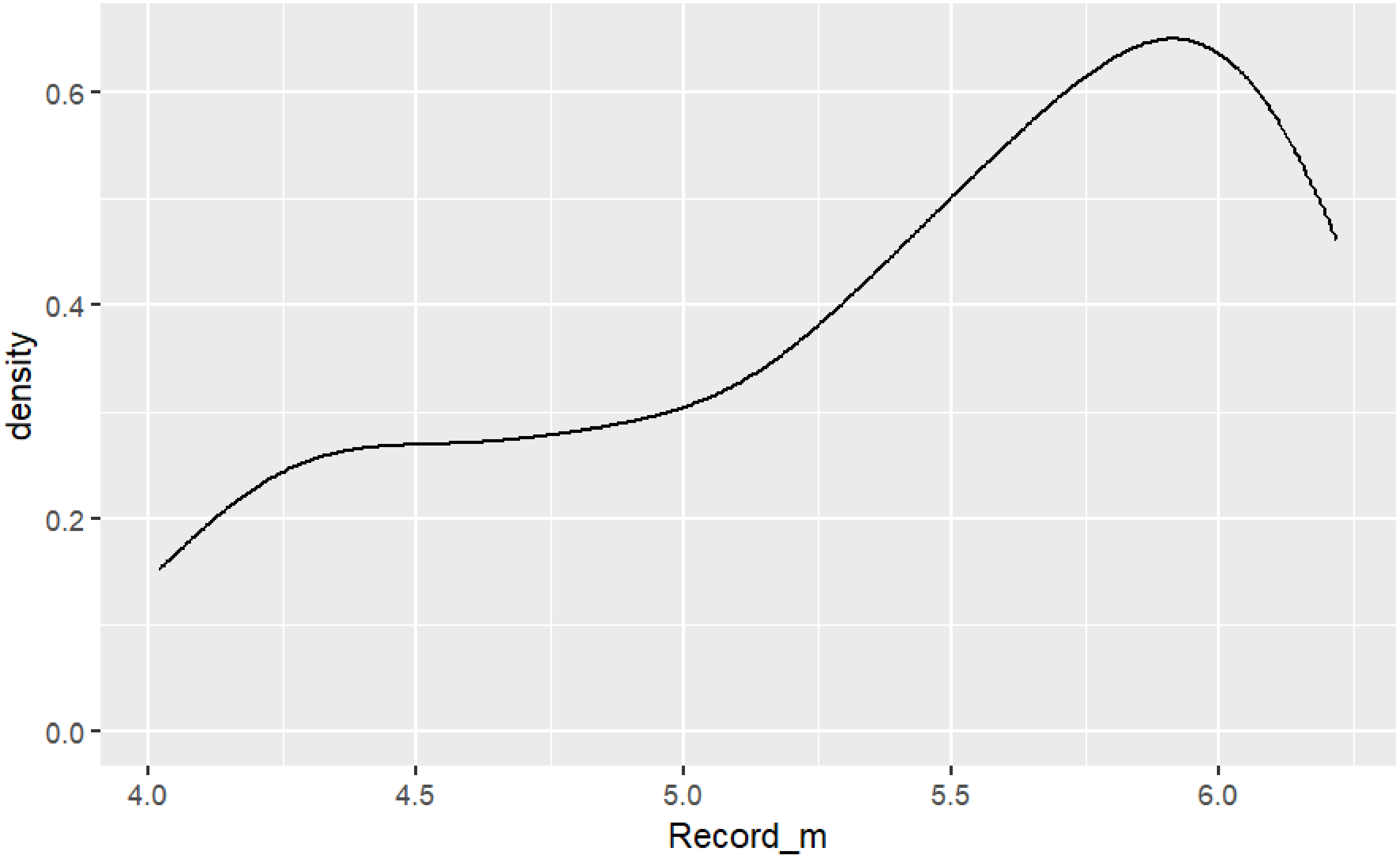
```
PVRecordsData <-
  PVrecords %>%
  mutate(Record_m = parse_number(Mark)) %>%
  select(Mark, Record_m)

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+¼ in)
4.09 m (13 ft 5 in)
4.12 m (13 ft 6 in)
4.21 m (13 ft 9+½ in)
4.23 m (13 ft 10+½ in)
4.25 m (13 ft 11+¼ 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 at 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()`)

```
library("rvest")

page_url <- "https://en.wikipedia.org/wiki/Mile_run_world_record_progression"
XPATH <- '//*[@id="mw-content-text"]/div/table'

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

Assignments

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