MATH 118: Notes K

Data from the web

```
#LOAD PACKAGES
library(tidyverse)
```

Data doesn't just magically appear on your computer you need to get it from somewhere.

Often times, we download data (.csv files or other) and save it locally on our computer.

Other times, we download it from R packages (like we did with the gapminder dataset).

Obtaining Data From The Web

For example, maybe we are interested in renting an apartment or house in Vermont (or studying the rental market in Vermont). You might navigate to Craigslist to get some information: https://vermont.craigslist.org/search/apa

We could spend many hours writing down and creating a spreadsheet with the information about each available apartment... or...

When you enter a URL into your browser, your browser connects to the web server at that URL and asks for the *source code* for that website. We can view the source code in a web brower by clicking on *view source*.

Web scraping is a process by which we can use R (or other software) to systematically go through the source code to extract content and data.

STOP: Are you allowed to scrape that website?

Before scraping data from the web, you should always check whether or not your are *allowed* to scrape it. There are two places you can look: the `robots.txt` file and the Terms of Service Document.

In the Craigslist terms of service document, we find the following text *"You agree not to copy/collect CL content via robots, spiders, scripts, scrapers, crawlers, or any automated or manual equivalent (e.g. by hand).

Wikipedia on the other hand, doesn't explicit state that web scraping is disallowed.

Web Scraping

We need the package `rvest` to help us with this.

```
library(rvest)
```

Webscraping Tables from Wikipedia

```
youtube_videos <- read_html("https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos") %>%
  html_element(".wikitable") %>%
  html_table()

youtube_videos
```

```
## # A tibble: 31 x 7
##    No.   `Video name`                          Uploa~1 Views~2 Publi~3 Note  ``
##    <chr> <chr>                                 <chr>   <chr>   <chr>   <chr> <chr>
##  1 1.    "\"Baby Shark Dance\"[4]"             Pinkfo~ 12.55   June 1~ "[A]" <NA>
##  2 2.    "\"Despacito\"[7]"                    Luis F~ 8.11    Januar~ "[B]" <NA>
##  3 3.    "\"Johny Johny Yes Papa\"[14]"        LooLoo~ 6.66    Octobe~ "[C]" <NA>
##  4 4.    "\"Bath Song\"[15]"                   Cocome~ 6.08    May 2,~ "[D]" <NA>
##  5 5.    "\"Shape of You\"[16]"                Ed She~ 5.95    Januar~ "[E]" <NA>
##  6 6.    "\"See You Again\"[18]"               Wiz Kh~ 5.82    April ~ "[F]" <NA>
##  7 7.    "\"Phonics Song with Two Words\"[2~ ChuChu~ 5.22    March ~ "[G]" <NA>
##  8 8.    "\"Wheels on the Bus\"[24]"           Cocome~ 5.04    May 24~ ""    <NA>
##  9 9.    "\"Uptown Funk\"[25]"                 Mark R~ 4.87    Novemb~ "[H]" <NA>
## 10 10.   "\"Learning Colors - Colorful Eggs~ Mirosh~ 4.86    Februa~ "[I]" <NA>
## # ... with 21 more rows, and abbreviated variable names 1: Uploader,
## #   2: `Views (billions)`, 3: `Publication date`
```

Web scraping doesn't always format perfectly. Let's clean it up!

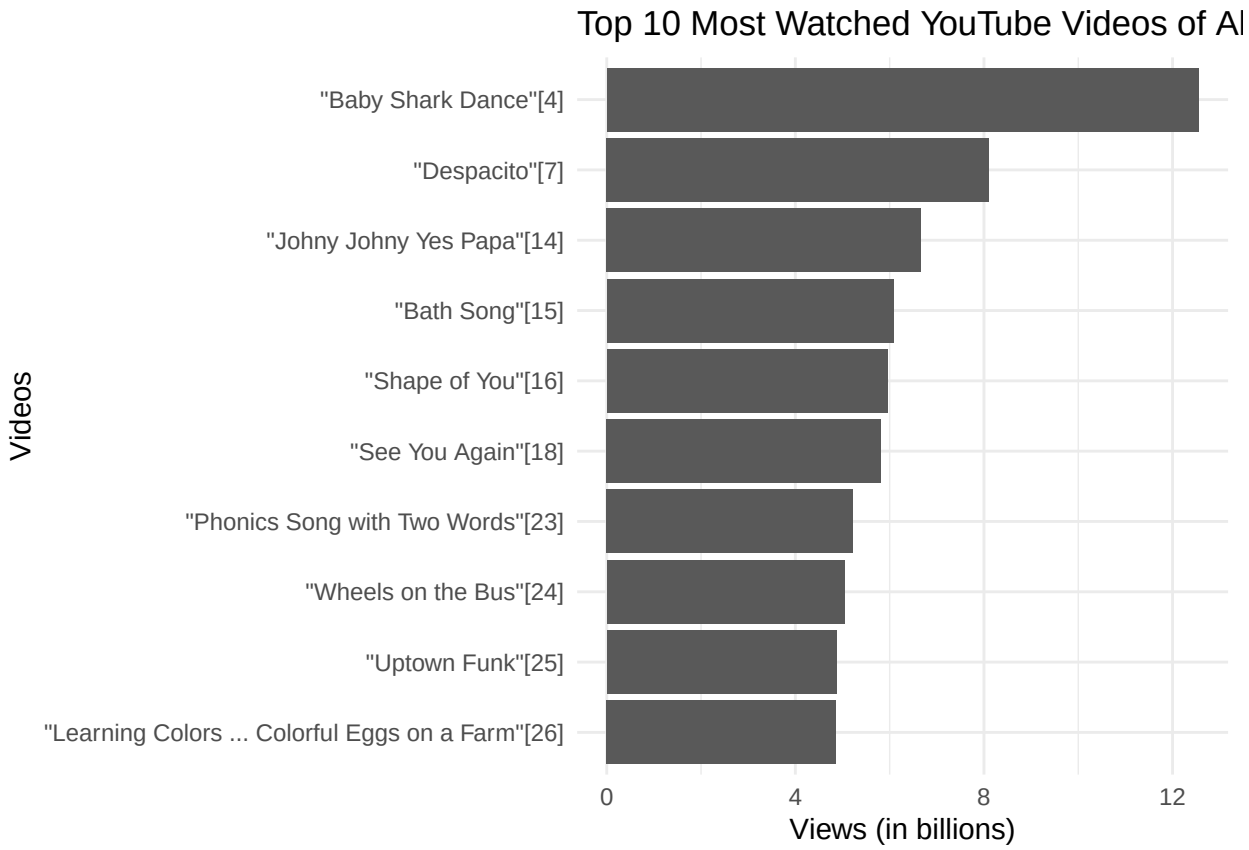
```
library(janitor)
```

```
youtube_videos <- clean_names(youtube_videos)

youtube_videos$views_billions <- as.numeric(youtube_videos$views_billions)

top10 <- youtube_videos %>%
  arrange(desc(views_billions)) %>%
  slice(1:10)

top10
```

```
## # A tibble: 10 x 7
##    no    video_name                            uploa~1 views~2 publi~3 note  x
##    <chr> <chr>                                 <chr>     <dbl> <chr>   <chr> <chr>
##  1 1.    "\"Baby Shark Dance\"[4]"             Pinkfo~   12.6  June 1~ "[A]" <NA>
##  2 2.    "\"Despacito\"[7]"                    Luis F~    8.11 Januar~ "[B]" <NA>
##  3 3.    "\"Johny Johny Yes Papa\"[14]"        LooLoo~    6.66 Octobe~ "[C]" <NA>
##  4 4.    "\"Bath Song\"[15]"                   Cocome~    6.08 May 2,~ "[D]" <NA>
##  5 5.    "\"Shape of You\"[16]"                Ed She~    5.95 Januar~ "[E]" <NA>
##  6 6.    "\"See You Again\"[18]"               Wiz Kh~    5.82 April ~ "[F]" <NA>
##  7 7.    "\"Phonics Song with Two Words\"[2~ ChuChu~    5.22 March ~ "[G]" <NA>
##  8 8.    "\"Wheels on the Bus\"[24]"           Cocome~    5.04 May 24~ ""    <NA>
##  9 9.    "\"Uptown Funk\"[25]"                 Mark R~    4.87 Novemb~ "[H]" <NA>
## 10 10.   "\"Learning Colors - Colorful Eggs~ Mirosh~    4.86 Februa~ "[I]" <NA>
## # ... with abbreviated variable names 1: uploader, 2: views_billions,
## #   3: publication_date
```

Once we have this data, we can make cool plots!

```
ggplot(top10, aes(x=views_billions, y=reorder(video_name, views_billions))) +
  geom_bar(stat="identity") +
  xlab("Views (in billions)") +
  ylab("Videos") +
  ggtitle("Top 10 Most Watched YouTube Videos of All Time") +
  theme_minimal()
```



Webscraping Text... and sometimes making it into tables

Let's look at the top 100 feature films released in 2020 and 2021 available here: https://www.imdb.com/search/title/?count=100&release_date=2020,2020&title_type=feature

```
URL <- read_html("https://www.imdb.com/search/title/?count=100&release_date=2020,2021&title_type=feature
```

Scrape IMBD for the titles of the 100 most popular feature films in 2020.

```
title_data <- URL %>%
  html_elements(".lister-item-header a") %>%
  html_text()

title_data
```

```
##   [1] "Children of the Corn"
##   [2] "Dune"
##   [3] "Promising Young Woman"
##   [4] "Spider-Man: No Way Home"
##   [5] "The Black Phone"
##   [6] "Nobody"
##   [7] "The Suicide Squad"
##   [8] "Don't Look Up"
##   [9] "Zack Snyder's Justice League"
##  [10] "Girl in the Basement"
##  [11] "No Time to Die"
##  [12] "Tenet"
```

```
##  [13] "The Matrix Resurrections"
##  [14] "Sing 2"
##  [15] "Sky High"
##  [16] "Encanto"
##  [17] "Old"
##  [18] "Free Guy"
##  [19] "F9: The Fast Saga"
##  [20] "The Fallout"
##  [21] "Spiral"
##  [22] "CODA"
##  [23] "Wonder Woman 1984"
##  [24] "Last Night in Soho"
##  [25] "Eternals"
##  [26] "The King's Man"
##  [27] "Wrath of Man"
##  [28] "Ghostbusters: Afterlife"
##  [29] "Marcel the Shell with Shoes On"
##  [30] "Licorice Pizza"
##  [31] "Shang-Chi and the Legend of the Ten Rings"
##  [32] "After We Collided"
##  [33] "365 Days"
##  [34] "The Last Duel"
##  [35] "Pleasure"
##  [36] "West Side Story"
##  [37] "Godzilla vs. Kong"
##  [38] "Mortal Kombat"
##  [39] "Finch"
##  [40] "Cruella"
##  [41] "The Father"
##  [42] "Hamilton"
##  [43] "The Devil All the Time"
##  [44] "The Hunt"
##  [45] "The Voyeurs"
##  [46] "Black Widow"
##  [47] "Nightmare Alley"
##  [48] "The French Dispatch"
##  [49] "News of the World"
##  [50] "House of Gucci"
##  [51] "The Power of the Dog"
##  [52] "The Tomorrow War"
##  [53] "Red Notice"
##  [54] "Enola Holmes"
##  [55] "Army of the Dead"
##  [56] "The Worst Person in the World"
##  [57] "Another Round"
##  [58] "A Quiet Place Part II"
##  [59] "The Little Things"
##  [60] "Nomadland"
##  [61] "The Green Knight"
##  [62] "The Many Saints of Newark"
##  [63] "The Invisible Man"
##  [64] "The Harder They Fall"
##  [65] "The Guilty"
##  [66] "Venom: Let There Be Carnage"
```

```
##  [67] "Resident Evil: Welcome to Raccoon City"
##  [68] "Belfast"
##  [69] "The Call of the Wild"
##  [70] "Dolittle"
##  [71] "Run"
##  [72] "Soul"
##  [73] "Birds of Prey"
##  [74] "Greenland"
##  [75] "Army of Thieves"
##  [76] "Malignant"
##  [77] "Cuties"
##  [78] "Clifford the Big Red Dog"
##  [79] "Emma."
##  [80] "Willy's Wonderland"
##  [81] "The Old Guard"
##  [82] "Ron's Gone Wrong"
##  [83] "Mulan"
##  [84] "Hillbilly Elegy"
##  [85] "Extraction"
##  [86] "The Babysitter: Killer Queen"
##  [87] "Greyhound"
##  [88] "Pig"
##  [89] "Raya and the Last Dragon"
##  [90] "Luca"
##  [91] "The Lost Daughter"
##  [92] "Sonic the Hedgehog"
##  [93] "Operation Mincemeat"
##  [94] "I'm Thinking of Ending Things"
##  [95] "Infinite"
##  [96] "The Call"
##  [97] "Demon Slayer the Movie: Mugen Train"
##  [98] "Wrong Turn"
##  [99] "We Can Be Heroes"
## [100] "Underwater"
```

Scrape IMBD for the runtime of the 100 most popular feature films in 2020.

```
library(readr)
# need this package for parse_number()
```

```
runtime_data <- URL %>%
  html_nodes(".text-muted .runtime") %>%
  html_text() %>%
  parse_number() %>% #this picks out only the numbers (and drops characters, in this case, "mins")
  as.numeric()

runtime_data
```

```
##   [1]  93 155 113 148 103  92 132 138 242  88 163 150 148 110 121 102 108 115
##  [19] 143  96  93 111 151 116 156 131 119 124  90 133 132 105 114 152 109 156
##  [37] 113 110 115 134  97 160 138  90 116 134 150 107 118 158 126 138 118 123
##  [55] 148 128 117  97 128 107 130 120 124 139  90  97 107  98 100 101  90 100
##  [73] 109 119 127 111  96  96 124  88 125 107 115 116 116 101  91  92 107  95
##  [91] 121  99 128 134 106 112 117 109 100  95
```

Scrape IMBD for the ratings of the 100 most popular feature films in 2020

```
rating_data <- URL %>%
  html_elements(".ratings-imdb-rating strong") %>%
  html_text() %>%
  as.numeric()

rating_data
```

```
##   [1] 3.6 8.0 7.5 8.2 6.9 7.4 7.2 7.2 8.0 6.3 7.3 7.3 5.7 7.4 5.6 7.2 5.8 7.1
##  [19] 5.2 7.0 5.2 8.0 5.4 7.0 6.3 6.3 7.1 7.1 7.7 7.2 7.4 5.0 3.3 7.4 6.4 7.2
##  [37] 6.3 6.0 6.9 7.3 8.2 8.4 7.1 6.5 6.0 6.7 7.0 7.1 6.8 6.6 6.8 6.5 6.3 6.6
##  [55] 5.7 7.8 7.7 7.2 6.3 7.3 6.6 6.3 7.1 6.6 6.3 5.9 5.2 7.3 6.7 5.6 6.7 8.0
##  [73] 6.0 6.4 6.4 6.2 3.6 5.9 6.7 5.5 6.6 7.0 5.7 6.7 6.7 5.8 7.0 6.9 7.3 7.4
##  [91] 6.7 6.5 6.6 6.6 5.4 7.1 8.2 5.5 4.7 5.8
```

Scrape IMBD for the number of votes of the 100 most popular feature films in 2020

```
votes_data <- URL %>%
  html_elements(".sort-num_votes-visible span:nth-child(2)") %>%
  html_text() %>%
  parse_number() %>%
  as.numeric()

votes_data
```

```
##   [1]   1299 658998 183246 781610 159485 269162 364711 550963 407816   9224
##  [11] 413361 522580 257403  73291   4175 232138 130113 382991 139067  26075
##  [21]  56791 142423 275458 146570 353051 154437 186020 189402  16687 124451
##  [31] 395177  33767  92293 162719  18816  87036 216494 178727  86947 241291
##  [41] 163518  99262 140460 118176  24437 390955 150319 129420  90105 143014
##  [51] 180564 211282 285983 201153 176799  76809 169216 241347 109657 168032
##  [61] 105122  56755 234710  65920 132070 233119  56771  79334  51791  66650
##  [71]  84559 340316 246195 122013  80896  95801  31297  14519  57159  31910
##  [81] 170725  36104 151276  43798 209945  44636 101544  79456 157314 170213
##  [91]  68393 144392  29344  91215  54331  35983  62293  32855  15693  86640
```

Scrape IMBD for the gross earnings of the 100 most popular feature films in 2020

```
gross_data <- URL %>%
  html_elements(".ghost~ .text-muted+ span") %>%
  html_text() %>%
  parse_number() %>%
  as.numeric()

gross_data
```

```
##  [1] 108.33 804.75  90.12  27.27  55.82 160.87  58.46 162.79  96.09 121.63
## [11] 173.20  46.37 164.87  37.18  27.47 129.36   5.79 224.54   2.39  10.85
## [21] 100.92  42.20  86.10   5.81 183.65  53.81   1.00 160.07  70.41 213.55
## [31]  17.00  62.34  77.05  84.16  48.95  54.72 148.97  47.70  17.29
```

```
#Notes this one has less entries than the rest and we can't figure out which one goes with which movies
```

We can combine all this data into one data frame:

```
movies<-data.frame(Title = title_data,
Runtime = runtime_data,
Rating = rating_data
```

```
)


movies
```

```
##                                       Title Runtime Rating
## 1                         Children of the Corn      93    3.6
## 2                                      Dune     155    8.0
## 3                       Promising Young Woman     113    7.5
## 4                     Spider-Man: No Way Home     148    8.2
## 5                             The Black Phone     103    6.9
## 6                                      Nobody      92    7.4
## 7                           The Suicide Squad     132    7.2
## 8                              Don't Look Up     138    7.2
## 9                  Zack Snyder's Justice League     242    8.0
## 10                        Girl in the Basement      88    6.3
## 11                             No Time to Die     163    7.3
## 12                                      Tenet     150    7.3
## 13                  The Matrix Resurrections     148    5.7
## 14                                     Sing 2     110    7.4
## 15                                   Sky High     121    5.6
## 16                                    Encanto     102    7.2
## 17                                        Old     108    5.8
## 18                                   Free Guy     115    7.1
## 19                           F9: The Fast Saga     143    5.2
## 20                                 The Fallout      96    7.0
## 21                                      Spiral      93    5.2
## 22                                       CODA     111    8.0
## 23                           Wonder Woman 1984     151    5.4
## 24                          Last Night in Soho     116    7.0
## 25                                    Eternals     156    6.3
## 26                               The King's Man     131    6.3
## 27                                 Wrath of Man     119    7.1
## 28                       Ghostbusters: Afterlife     124    7.1
## 29               Marcel the Shell with Shoes On      90    7.7
## 30                                Licorice Pizza     133    7.2
## 31  Shang-Chi and the Legend of the Ten Rings     132    7.4
## 32                             After We Collided     105    5.0
## 33                                     365 Days     114    3.3
## 34                                The Last Duel     152    7.4
## 35                                     Pleasure     109    6.4
## 36                              West Side Story     156    7.2
## 37                             Godzilla vs. Kong     113    6.3
## 38                                 Mortal Kombat     110    6.0
## 39                                         Finch     115    6.9
## 40                                       Cruella     134    7.3
## 41                                   The Father      97    8.2
## 42                                     Hamilton     160    8.4
## 43                       The Devil All the Time     138    7.1
## 44                                     The Hunt      90    6.5
## 45                                   The Voyeurs     116    6.0
## 46                                   Black Widow     134    6.7
## 47                               Nightmare Alley     150    7.0
## 48                           The French Dispatch     107    7.1
```

```
## 49                          News of the World     118    6.8
## 50                             House of Gucci     158    6.6
## 51                       The Power of the Dog     126    6.8
## 52                           The Tomorrow War     138    6.5
## 53                                 Red Notice     118    6.3
## 54                              Enola Holmes     123    6.6
## 55                            Army of the Dead     148    5.7
## 56               The Worst Person in the World     128    7.8
## 57                                Another Round     117    7.7
## 58                       A Quiet Place Part II      97    7.2
## 59                           The Little Things     128    6.3
## 60                                   Nomadland     107    7.3
## 61                            The Green Knight     130    6.6
## 62                  The Many Saints of Newark     120    6.3
## 63                            The Invisible Man     124    7.1
## 64                         The Harder They Fall     139    6.6
## 65                                    The Guilty      90    6.3
## 66                  Venom: Let There Be Carnage      97    5.9
## 67      Resident Evil: Welcome to Raccoon City     107    5.2
## 68                                      Belfast      98    7.3
## 69                         The Call of the Wild     100    6.7
## 70                                     Dolittle     101    5.6
## 71                                          Run      90    6.7
## 72                                         Soul     100    8.0
## 73                                Birds of Prey     109    6.0
## 74                                    Greenland     119    6.4
## 75                               Army of Thieves     127    6.4
## 76                                    Malignant     111    6.2
## 77                                       Cuties      96    3.6
## 78                    Clifford the Big Red Dog      96    5.9
## 79                                        Emma.     124    6.7
## 80                            Willy's Wonderland      88    5.5
## 81                                 The Old Guard     125    6.6
## 82                               Ron's Gone Wrong     107    7.0
## 83                                         Mulan     115    5.7
## 84                               Hillbilly Elegy     116    6.7
## 85                                    Extraction     116    6.7
## 86                The Babysitter: Killer Queen     101    5.8
## 87                                     Greyhound      91    7.0
## 88                                           Pig      92    6.9
## 89                    Raya and the Last Dragon     107    7.3
## 90                                          Luca      95    7.4
## 91                           The Lost Daughter     121    6.7
## 92                          Sonic the Hedgehog      99    6.5
## 93                         Operation Mincemeat     128    6.6
## 94               I'm Thinking of Ending Things     134    6.6
## 95                                    Infinite     106    5.4
## 96                                     The Call     112    7.1
## 97          Demon Slayer the Movie: Mugen Train     117    8.2
## 98                                    Wrong Turn     109    5.5
## 99                           We Can Be Heroes     100    4.7
## 100                                  Underwater      95    5.8
```

Make a list OR Make a plot!

```
ggplot(movies, aes(x=runtime_data, y=rating_data)) +
  geom_point()
```