# SDS 264 HW3

```r
# Libraries
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becon
```

```r
library(stringr)
library(rvest)
```

```
Attaching package: 'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

```r
library(polite)
library(sf)
```

```
Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
library(maps)
```

```
Attaching package: 'maps'

The following object is masked from 'package:purrr':

    map
```

```r
library(viridis)
```

```
Loading required package: viridisLite

Attaching package: 'viridis'

The following object is masked from 'package:maps':

    unemp
```

```r
library(leaflet)
library(htmltools)
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library(httr2)
library(httr)
library(lubridate)
library(tidycensus)
library(purrr)
```

**07 - On Your Own #2-3** 2. Write a function to give choices about year, county, and variables

```r
acs_function <- function(year, county, vars) {
    Sys.sleep(0.5)
tidycensus::get_acs(
    year = year,
    state = "MN",
    geography = "tract",
    variables = vars,
    output = "wide",
    geometry = TRUE,
    county = county,   # specify county in call
    show_call = TRUE       # see resulting query
)  |>
    mutate(year = year)
  }

acs_function(2021, "Hennepin", c("B01003_001", "B19013_001"))
```

```
Getting data from the 2017-2021 5-year ACS

Warning: * You have not set a Census API key. Users without a key are limited to 500
queries per day and may experience performance limitations.
i For best results, get a Census API key at
http://api.census.gov/data/key_signup.html and then supply the key to the
`census_api_key()` function to use it throughout your tidycensus session.
This warning is displayed once per session.

Downloading feature geometry from the Census website.  To cache shapefiles for use in future

Census API call: https://api.census.gov/data/2021/acs/acs5?get=B01003_001E%2CB01003_001M%2CB


  |
  |                                                                      |   0%
  |
  |=                                                                     |   2%
  |
  |==                                                                    |   3%
  |
  |===                                                                   |   5%
  |
  |====                                                                  |   6%
  |
  |======                                                                |   8%
  |
  |========                                                              |  11%
  |
  |=========                                                             |  13%
  |
  |============                                                          |  18%
  |
  |===============                                                       |  21%
  |
  |================                                                      |  22%
  |
  |====================                                                  |  28%
  |
  |=====================                                                 |  30%
  |
  |========================                                              |  35%
  |
  |============================                                          |  40%
  |
  |========================================                              |  57%
  |
  |==========================================                            |  60%
  |
  |===================================================                   |  73%
  |
  |=====================================================                 |  76%
  |
  |==================================================================    |  94%
  |
  |======================================================================| 100%

Simple feature collection with 329 features and 7 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -93.76838 ymin: 44.78538 xmax: -93.17722 ymax: 45.24662
Geodetic CRS:  NAD83
First 10 features:
         GEOID                                            NAME B01003_001E
1  27053024300    Census Tract 243, Hennepin County, Minnesota        4744
```

```
2  27053110500   Census Tract 1105, Hennepin County, Minnesota        4969
3  27053024006 Census Tract 240.06, Hennepin County, Minnesota        2205
4  27053022801 Census Tract 228.01, Hennepin County, Minnesota        2481
5  27053026908 Census Tract 269.08, Hennepin County, Minnesota        6139
6  27053025401 Census Tract 254.01, Hennepin County, Minnesota        4428
7  27053108600   Census Tract 1086, Hennepin County, Minnesota        2947
8  27053026824 Census Tract 268.24, Hennepin County, Minnesota        4551
9  27053106000   Census Tract 1060, Hennepin County, Minnesota        3375
10 27053000102   Census Tract 1.02, Hennepin County, Minnesota        4896
   B01003_001M B19013_001E B19013_001M                       geometry year
1          481       72240        5745 MULTIPOLYGON (((-93.31881 4... 2021
2          651       80157        5307 MULTIPOLYGON (((-93.22237 4... 2021
3          270      143125       22624 MULTIPOLYGON (((-93.35044 4... 2021
4          359      133958       34619 MULTIPOLYGON (((-93.34793 4... 2021
5          792      110246        3614 MULTIPOLYGON (((-93.39145 4... 2021
6          648       68711       11097 MULTIPOLYGON (((-93.28347 4... 2021
7          587       57470       15799 MULTIPOLYGON (((-93.24995 4... 2021
8          483      127819       26964 MULTIPOLYGON (((-93.36073 4... 2021
9          622       23492        5316 MULTIPOLYGON (((-93.25966 4... 2021
10         597       59750       11634 MULTIPOLYGON (((-93.29919 4... 2021
```

3. Use your function from (2) along with `map` and `list_rbind` to build a data set for Rice county for the years 2019-2021

```r
years <- c(2019:2021)

ricedata <- years |>
  purrr::map(acs_function,
             county = "Rice",
             vars = c("B01003_001", "B19013_001")) |>
  list_rbind()
```

```
Getting data from the 2015-2019 5-year ACS

Downloading feature geometry from the Census website.  To cache shapefiles for use in future

Census API call: https://api.census.gov/data/2019/acs/acs5?get=B01003_001E%2CB01003_001M%2CB



  |
  |                                                                      |   0%
  |
  |=============                                                         |  18%
  |
  |==================                                                    |  25%
  |
  |========================                                              |  34%
  |
  |============================                                          |  39%
  |
  |==============================                                        |  43%
  |
  |===================================                                   |  50%
  |
  |======================================                                |  54%
  |
  |=========================================                             |  59%
  |
  |==============================================                        |  66%
  |
  |==================================================                    |  72%
  |
  |=======================================================               |  79%
  |
  |===================================================================== |  99%
  |
  |======================================================================| 100%
```

```
Getting data from the 2016-2020 5-year ACS

Downloading feature geometry from the Census website.  To cache shapefiles for use in future

Census API call: https://api.census.gov/data/2020/acs/acs5?get=B01003_001E%2CB01003_001M%2CB
```

```

  |
  |                                                                      |   0%
  |
  |==                                                                    |   3%
  |
  |===                                                                   |   5%
  |
  |======                                                                |   9%
  |
  |=======                                                               |  11%
  |
  |========                                                              |  12%
  |
  |============                                                          |  17%
  |
  |=============                                                         |  18%
  |
  |=======================                                               |  33%
  |
  |===========================                                           |  39%
  |
  |============================                                          |  41%
  |
  |================================                                      |  45%
  |
  |==================================                                    |  48%
  |
  |=====================================                                 |  53%
  |
  |=======================================                               |  56%
  |
  |============================================                          |  63%
  |
  |==============================================                        |  66%
  |
  |================================================                      |  69%
  |
  |==================================================                    |  71%
  |
  |====================================================                  |  74%
  |
  |===============================================================       |  90%
  |
  |======================================================================| 100%

Getting data from the 2017-2021 5-year ACS

Downloading feature geometry from the Census website.  To cache shapefiles for use in future
```

```
Census API call: https://api.census.gov/data/2021/acs/acs5?get=B01003_001E%2CB01003_001M%2CB
```

```
ricedata
```

```
         GEOID                                        NAME B01003_001E
1  27131070504 Census Tract 705.04, Rice County, Minnesota        3933
2  27131070400    Census Tract 704, Rice County, Minnesota        4511
3  27131070300    Census Tract 703, Rice County, Minnesota        4551
4  27131070503 Census Tract 705.03, Rice County, Minnesota        3348
5  27131070601 Census Tract 706.01, Rice County, Minnesota        3526
6  27131070800    Census Tract 708, Rice County, Minnesota        8101
7  27131070901 Census Tract 709.01, Rice County, Minnesota        5509
8  27131070700    Census Tract 707, Rice County, Minnesota        7165
9  27131070100    Census Tract 701, Rice County, Minnesota        7333
10 27131070602 Census Tract 706.02, Rice County, Minnesota        5211
11 27131070200    Census Tract 702, Rice County, Minnesota        5463
12 27131070902 Census Tract 709.02, Rice County, Minnesota        3160
13 27131070501 Census Tract 705.01, Rice County, Minnesota        4374
14 27131070501 Census Tract 705.01, Rice County, Minnesota        4272
15 27131070504 Census Tract 705.04, Rice County, Minnesota        3941
16 27131070801 Census Tract 708.01, Rice County, Minnesota        4456
17 27131070200    Census Tract 702, Rice County, Minnesota        5508
18 27131070701 Census Tract 707.01, Rice County, Minnesota        3057
19 27131070400    Census Tract 704, Rice County, Minnesota        4686
20 27131070300    Census Tract 703, Rice County, Minnesota        4737
21 27131070601 Census Tract 706.01, Rice County, Minnesota        3669
22 27131070102 Census Tract 701.02, Rice County, Minnesota        3786
23 27131070802 Census Tract 708.02, Rice County, Minnesota        3873
24 27131070702 Census Tract 707.02, Rice County, Minnesota        3872
25 27131070901 Census Tract 709.01, Rice County, Minnesota        5681
26 27131070503 Census Tract 705.03, Rice County, Minnesota        3185
27 27131070902 Census Tract 709.02, Rice County, Minnesota        2992
28 27131070101 Census Tract 701.01, Rice County, Minnesota        3428
29 27131070602 Census Tract 706.02, Rice County, Minnesota        5406
30 27131070902 Census Tract 709.02, Rice County, Minnesota        3212
31 27131070601 Census Tract 706.01, Rice County, Minnesota        3775
32 27131070503 Census Tract 705.03, Rice County, Minnesota        3035
33 27131070702 Census Tract 707.02, Rice County, Minnesota        3738
34 27131070901 Census Tract 709.01, Rice County, Minnesota        5858
35 27131070801 Census Tract 708.01, Rice County, Minnesota        4618
36 27131070501 Census Tract 705.01, Rice County, Minnesota        4242
37 27131070300    Census Tract 703, Rice County, Minnesota        4657
```

```
38 27131070200    Census Tract 702, Rice County, Minnesota        5419
39 27131070400    Census Tract 704, Rice County, Minnesota        4380
40 27131070701 Census Tract 707.01, Rice County, Minnesota        3028
41 27131070504 Census Tract 705.04, Rice County, Minnesota        3917
42 27131070101 Census Tract 701.01, Rice County, Minnesota        3417
43 27131070802 Census Tract 708.02, Rice County, Minnesota        3944
44 27131070102 Census Tract 701.02, Rice County, Minnesota        4201
45 27131070602 Census Tract 706.02, Rice County, Minnesota        5354
   B01003_001M B19013_001E B19013_001M                       geometry year
1          273       63989        9273 MULTIPOLYGON (((-93.19137 4... 2019
2          168       85952        2758 MULTIPOLYGON (((-93.40564 4... 2019
3          190       78343        4242 MULTIPOLYGON (((-93.52521 4... 2019
4          245       92321       14200 MULTIPOLYGON (((-93.16075 4... 2019
5          333       50368        9979 MULTIPOLYGON (((-93.17615 4... 2019
6          465       48403        7679 MULTIPOLYGON (((-93.29819 4... 2019
7          456       44417       10552 MULTIPOLYGON (((-93.30904 4... 2019
8          414       67868        9422 MULTIPOLYGON (((-93.27265 4... 2019
9          326       91667        8106 MULTIPOLYGON (((-93.52452 4... 2019
10         310       64479       12376 MULTIPOLYGON (((-93.22644 4... 2019
11         177      101359        4104 MULTIPOLYGON (((-93.5246 44... 2019
12         410       45230       12887 MULTIPOLYGON (((-93.30888 4... 2019
13         270       66188        9179 MULTIPOLYGON (((-93.16981 4... 2019
14         316       64792       13256 MULTIPOLYGON (((-93.16981 4... 2020
15         536       63500        7351 MULTIPOLYGON (((-93.1909 44... 2020
16         703       67625       23325 MULTIPOLYGON (((-93.29829 4... 2020
17         473      104011        5648 MULTIPOLYGON (((-93.5246 44... 2020
18         218       73750       13139 MULTIPOLYGON (((-93.26704 4... 2020
19         296       86094        3438 MULTIPOLYGON (((-93.40564 4... 2020
20         244       79068        4902 MULTIPOLYGON (((-93.52518 4... 2020
21         525       52936       10436 MULTIPOLYGON (((-93.17615 4... 2020
22         199       96023       13649 MULTIPOLYGON (((-93.44292 4... 2020
23         437       63924        8715 MULTIPOLYGON (((-93.28272 4... 2020
24         425       49811       16864 MULTIPOLYGON (((-93.27265 4... 2020
25         566       51595        9615 MULTIPOLYGON (((-93.30904 4... 2020
26         341      100516       11630 MULTIPOLYGON (((-93.16075 4... 2020
27         440       46750       15457 MULTIPOLYGON (((-93.30888 4... 2020
28         295      100563       15809 MULTIPOLYGON (((-93.52452 4... 2020
29         377       62078        5270 MULTIPOLYGON (((-93.22644 4... 2020
30         421       47059       15456 MULTIPOLYGON (((-93.30888 4... 2021
31         435       56319        4333 MULTIPOLYGON (((-93.17615 4... 2021
32         321      105952        8429 MULTIPOLYGON (((-93.16075 4... 2021
33         409       57126       13968 MULTIPOLYGON (((-93.27265 4... 2021
34         714       47344        9579 MULTIPOLYGON (((-93.30904 4... 2021
35          622       61193       23977 MULTIPOLYGON (((-93.29829 4... 2021
36          380       79063       15272 MULTIPOLYGON (((-93.16981 4... 2021
37          296       83911        7244 MULTIPOLYGON (((-93.52522 4... 2021
38          520      111711       10313 MULTIPOLYGON (((-93.5246 44... 2021
39          274       90179        4919 MULTIPOLYGON (((-93.40564 4... 2021
40          358       82500       20934 MULTIPOLYGON (((-93.26775 4... 2021
41          537       67219        9805 MULTIPOLYGON (((-93.1909 44... 2021
42          270      108490        1768 MULTIPOLYGON (((-93.52452 4... 2021
43          462       63679       12261 MULTIPOLYGON (((-93.28274 4... 2021
44          199       85789       20094 MULTIPOLYGON (((-93.44292 4... 2021
45          359       63835        4805 MULTIPOLYGON (((-93.22644 4... 2021
```

### 07 - OMDB example - obtain a key and assemble your own well-formatted 5x5 tibble with 5 movies of your choosing and 5 variables of your choosing (see what else is available)

Here's an example of getting data from a website that attempts to make imdb movie data available as an API.

Initial instructions:

- go to omdbapi.com under the API Key tab and request a free API key
- store your key as discussed earlier
- explore the examples at omdbapi.com

```r
myapikey <- Sys.getenv("omdbapi")
```

```r
# Must figure out pattern in URL for obtaining different movies
#  - try searching for others
movies <- c("Despicable+Me", "The+Wild+Robot", "Anyone+But+You",
            "Barbie", "Top+Gun:+Maverick")

# Set up empty tibble
omdb <- tibble(title = character(),
               released = character(),
               runtime_min = double(),
               imdbrating = character(),
               rottentomatorating = double())

# Use for loop to run through API request process 5 times,
#   each time filling the next row in the tibble
for(i in 1:5) {
  url <- str_c("http://www.omdbapi.com/?t=",movies[i],
               "&apikey=", myapikey)
  Sys.sleep(0.5)
  onemovie <- GET(url)
  details <- content(onemovie, "parse")
  omdb[i,1] <- details$Title
  omdb[i,2] <- details$Released
  omdb[i,3] <- parse_number(details$Runtime)
  omdb[i,4] <- details$Ratings[[1]]$Value
  omdb[i,5] <- parse_number(details$Ratings[[2]]$Value)
}

omdb |>
  mutate(released = dmy(released),
         imdbrating = as.numeric(str_remove(imdbrating, "/10")))
```

```
# A tibble: 5 x 5
  title             released   runtime_min imdbrating rottentomatorating
  <chr>             <date>           <dbl>      <dbl>              <dbl>
1 Despicable Me     2010-07-09          95        7.6                 80
2 The Wild Robot    2024-09-27         102        8.5                 98
3 Anyone But You    2023-12-22         103        6.1                 52
4 Barbie            2023-07-21         114        6.8                 88
5 Top Gun: Maverick 2022-05-27         130        8.2                 96
```

### 08 - On Your Own #2.2-2.4

2. We would like to create a tibble with 4 years of data (2001-2004) from the Minnesota Wild hockey team. Specifically, we are interested in the "Scoring Regular Season" table from this webpage and the similar webpages from 2002, 2003, and 2004. Your final tibble should have 6 columns: player, year, age, pos (position), gp (games played), and pts (points).

You should (a) write a function called `hockey_stats` with inputs for team and year to scrape data from the "scoring Regular Season" table, and (b) use iteration techniques to scrape and combine 4 years worth of data. Here are some functions you might consider:

- `row_to_names(row_number = 1)` from the `janitor` package
- `clean_names()` also from the `janitor` package
- `bow()` and `scrape()` from the `polite` package
- `str_c()` from the `stringr` package (for creating urls with user inputs)
- `map2()` and `list_rbind()` for iterating and combining years

Try following these steps:

1) Be sure you can find and clean the correct table from the 2021 season.
2) Organize your `rvest` code from (1) into functions from the `polite` package.
3) Place the code from (2) into a function where the user can input a team and year. You would then adjust the url accordingly and produce a clean table for the user.
4) Use `map2` and `list_rbind` to build one data set containing Minnesota Wild data from 2001-2004.

```
# To check if web scraping is allowed
robotstxt::paths_allowed("https://www.hockey-reference.com/teams/MIN/2001.html")
```

```
 www.hockey-reference.com
```

```
[1] TRUE
```

```
# Step 1: read_html()
url <- str_c("https://www.hockey-reference.com/teams/MIN/2001.html")
player_url <- read_html(url)

# 2: html_nodes()
playerstat <- html_nodes(player_url, css = "table")
playerstat  # have to guesstimate which table contains our desired info
```

```
{xml_nodeset (6)}
[1] <table class="sortable stats_table" id="team_stats" data-cols-to-freeze=" ...
[2] <table class="sortable stats_table" id="team_stats_adv" data-cols-to-free ...
[3] <table class="sortable stats_table" id="roster" data-cols-to-freeze=",2"> ...
[4] <table class="stats_table sortable per_toggler soc" id="player_stats" dat ...
[5] <table class="sortable stats_table" id="goalies" data-cols-to-freeze=",2" ...
[6] <table class="stats_table sortable per_toggler soc" id="stats_misc_plus"  ...
```

```
# 3: html_table()
html_table(playerstat, header = TRUE, fill = TRUE)    # find the right table
```

```
[[1]]
# A tibble: 2 x 29
  Team  AvAge    GP     W     L     T    OL   PTS `PTS%`    GF    GA   SRS   SOS
  <chr> <dbl> <int> <int> <int> <int> <int> <int>  <dbl> <int> <int> <dbl> <dbl>
```

```
1 Minn~  27.4    82    25    39    13     5    68  0.415   168   210 -0.42  0.09
2 Leag~  27.8    82    36    32    10     4    86  0.525   226   226 NA    NA
# i 16 more variables: `GF/G` <dbl>, `GA/G` <dbl>, PP <int>, PPO <int>,
#   `PP%` <dbl>, PPA <int>, PPOA <int>, `PK%` <dbl>, SH <int>, SHA <int>,
#   S <int>, `S%` <dbl>, SA <int>, `SV%` <dbl>, PDO <lgl>, SO <int>

[[2]]
# A tibble: 2 x 22
  Team  `S%`  `SV%` PDO   CF    CA    `CF%` xGF   xGA   aGF   aGA   axDiff SCF
  <chr> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl>  <lgl>
1 Minn~ NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA     NA
2 Leag~ NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA     NA
# i 9 more variables: SCA <lgl>, `SCF%` <lgl>, HDF <lgl>, HDA <lgl>,
#   `HDF%` <lgl>, HDGF <lgl>, `HDC%` <lgl>, HDGA <lgl>, `HDCO%` <lgl>

[[3]]
# A tibble: 38 x 11
   No.   Player   Birth Pos     Age Ht       Wt `S/C` Exp   `Birth Date` Summary
   <chr> <chr>    <chr> <chr> <int> <chr> <int> <chr> <chr> <chr>        <chr>
 1 40    Chris A~ ca CA D         25 6-0     205 L/-   R     June 26, 19~ 0 G, 0~
 2 45    Peter B~ cs CS RW        27 6-0     185 R/-   R     September 5~ 4 G, 2~
 3 3     Ladisla~ cs CS D         25 6-2     190 L/-   1     March 24, 1~ 2 G, 5~
 4 31    Zac Bie~ ca CA G         24 6-5     205 -/L   3     September 1~ 0-1-0,~
 5 36    Sylvain~ ca CA LW        26 6-2     215 L/-   3     May 21, 1974 3 G, 2~
 6 5     Brad Bo~ ca CA D         28 6-1     205 L/-   3     May 5, 1972  0 G, 1~
 7 32    Brian B~ us US LW        27 5-10    186 L/-   1     November 28~ 0 G, 0~
 8 15    J.J. Da~ ca CA D         35 5-10    192 L/-   15    October 12,~ 0 G, 0~
 9 34    Jim Dowd us US C         32 6-0     180 R/-   9     December 25~ 7 G, 2~
10 11    Pascal ~ ca CA LW        21 6-1     205 L/-   R     April 7, 19~ 1 G, 0~
# i 28 more rows

[[4]]
# A tibble: 40 x 22
   ``    ``    ``    ``    ``    Scoring Scoring Scoring ``    ``    Goals Goals
   <chr> <chr> <chr> <chr> <chr> <chr>   <chr>   <chr>   <chr> <chr> <chr> <chr>
 1 Rk    Play~ Age   Pos   GP    G       A       PTS     +/-   PIM   EVG   PPG
 2 1     Scot~ 31    RW    58    11      28      39      6     45    7     2
 3 2     Mari~ 18    LW    71    18      18      36      -6    32    12    6
 4 3     Ľubo~ 32    D     80    11      23      34      -8    52    7     4
 5 4     Wes ~ 30    C     82    18      12      30      -8    37    11    0
 6 5     Fili~ 24    D     75    9       21      30      -6    28    5     4
 7 6     Darb~ 28    LW    72    18      11      29      1     36    14    3
 8 7     Jim ~ 32    C     68    7       22      29      -6    80    7     0
 9 8     Antt~ 27    LW    82    12      16      28      -7    24    10    0
10 9     Stac~ 26    C     76    7       20      27      3     20    6     1
# i 30 more rows
# i 10 more variables: Goals <chr>, Goals <chr>, Assists <chr>, Assists <chr>,
#   Assists <chr>, Shots <chr>, Shots <chr>, `Ice Time` <chr>,
#   `Ice Time` <chr>, `` <chr>

[[5]]
# A tibble: 6 x 22
  ``    ``     ``    `Goalie Stats` `Goalie Stats` `Goalie Stats` `Goalie Stats`
  <chr> <chr>  <chr> <chr>          <chr>          <chr>          <chr>
1 "Rk"  Player "Age" "GP"           "GS"           W              L
2 "1"   Manny~ "26"  "42"           ""             19             17
3 "2"   Jamie~ "29"  "38"           ""             5              23
4 "3"   Derek~ "21"  "4"            ""             1              3
5 "4"   Zac B~ "24"  "1"            ""             0              1
6 ""    Team ~ ""    ""             ""             25             44
# i 15 more variables: `Goalie Stats` <chr>, `Goalie Stats` <chr>,
#   `Goalie Stats` <chr>, `Goalie Stats` <chr>, `Goalie Stats` <chr>,
#   `Goalie Stats` <chr>, `Goalie Stats` <chr>, `Goalie Stats` <chr>,
#   `Goalie Stats` <chr>, `Goalie Stats` <chr>, `Goalie Stats` <chr>,
#   `Goalie Stats` <chr>, `Goalie Stats` <chr>, `` <chr>, `` <chr>

[[6]]
# A tibble: 35 x 18
   ``    ``                 ``    ``    ``    Adjusted Adjusted Adjusted Adjusted
   <chr> <chr>             <chr> <chr> <chr> <chr>    <chr>    <chr>    <chr>
 1 Rk    Player             Age   Pos   GP    G        A        PTS      GC
 2 1     Scott Pellerin     31    RW    58    12       30       42       14.7
 3 2     Marián Gáborík     18    LW    71    20       19       39       16.1
 4 3     Ľubomír Sekeráš    32    D     80    12       25       37       13.4
 5 4     Wes Walz           30    C     82    20       13       33       14.5
 6 5     Filip Kuba         24    D     75    10       22       32       11.5
 7 6     Jim Dowd           32    C     68    8        23       31       10.6
 8 7     Darby Hendrickson 28    LW    72    20       12       32       14.2
 9 8     Antti Laaksonen    27    LW    82    13       17       30       11.7
10 9     Stacy Roest        26    C     76    8        21       29       10.1
# i 25 more rows
# i 9 more variables: `Plus/Minus` <chr>, `Plus/Minus` <chr>,
#   `Plus/Minus` <chr>, `Plus/Minus` <chr>, `Plus/Minus` <chr>,
#   `Point Shares` <chr>, `Point Shares` <chr>, `Point Shares` <chr>, `` <chr>
```

```
player_table <- html_table(playerstat, header = TRUE, fill = TRUE)[[4]]
player_table |>
  row_to_names(row_number = 1)
```

```
# A tibble: 39 x 22
   Rk    Player      Age   Pos   GP    G     A     PTS   `+/-` PIM   EVG   PPG
   <chr> <chr>       <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
 1 1     Scott Pell~ 31    RW    58    11    28    39    6     45    7     2
 2 2     Marián Gáb~ 18    LW    71    18    18    36    -6    32    12    6
 3 3     Ľubomír Se~ 32    D     80    11    23    34    -8    52    7     4
 4 4     Wes Walz    30    C     82    18    12    30    -8    37    11    0
 5 5     Filip Kuba  24    D     75    9     21    30    -6    28    5     4
 6 6     Darby Hend~ 28    LW    72    18    11    29    1     36    14    3
 7 7     Jim Dowd    32    C     68    7     22    29    -6    80    7     0
 8 8     Antti Laak~ 27    LW    82    12    16    28    -7    24    10    0
 9 9     Stacy Roest 26    C     76    7     20    27    3     20    6     1
10 10    Aaron Gavey 26    C     75    10    14    24    -8    52    9     1
# i 29 more rows
# i 10 more variables: SHG <chr>, GWG <chr>, EV <chr>, PP <chr>, SH <chr>,
#   SOG <chr>, SPCT <chr>, TOI <chr>, ATOI <chr>, Awards <chr>
```

```
player_tibble <- html_table(playerstat, header = TRUE, fill = TRUE)[[4]] |>
  row_to_names(row_number = 1) |>
  clean_names()
player_tibble
```

```
# A tibble: 39 x 22
   rk    player      age   pos   gp    g     a     pts   x     pim   evg   ppg
   <chr> <chr>       <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
 1 1     Scott Pell~ 31    RW    58    11    28    39    6     45    7     2
 2 2     Marián Gáb~ 18    LW    71    18    18    36    -6    32    12    6
 3 3     Ľubomír Se~ 32    D     80    11    23    34    -8    52    7     4
 4 4     Wes Walz    30    C     82    18    12    30    -8    37    11    0
 5 5     Filip Kuba  24    D     75    9     21    30    -6    28    5     4
 6 6     Darby Hend~ 28    LW    72    18    11    29    1     36    14    3
 7 7     Jim Dowd    32    C     68    7     22    29    -6    80    7     0
 8 8     Antti Laak~ 27    LW    82    12    16    28    -7    24    10    0
 9 9     Stacy Roest 26    C     76    7     20    27    3     20    6     1
10 10    Aaron Gavey 26    C     75    10    14    24    -8    52    9     1
# i 29 more rows
# i 10 more variables: shg <chr>, gwg <chr>, ev <chr>, pp <chr>, sh <chr>,
#   sog <chr>, spct <chr>, toi <chr>, atoi <chr>, awards <chr>
```

```
#Perform Steps 0-3 using the polite package
session <- bow("https://www.hockey-reference.com/teams/MIN/2001.html", force = TRUE)

result <- scrape(session) |>
  html_nodes(css = "table") |>
  html_table(header = TRUE, fill = TRUE)
```

```
No encoding supplied: defaulting to UTF-8.
```

```
player_table <- result[[4]] |>
  row_to_names(row_number = 1) |>
  clean_names()
player_table
```

```
# A tibble: 39 x 22
   rk    player      age   pos   gp    g     a     pts   x     pim   evg   ppg
   <chr> <chr>       <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
 1 1     Scott Pell~ 31    RW    58    11    28    39    6     45    7     2
 2 2     Marián Gáb~ 18    LW    71    18    18    36    -6    32    12    6
 3 3     Ľubomír Se~ 32    D     80    11    23    34    -8    52    7     4
 4 4     Wes Walz    30    C     82    18    12    30    -8    37    11    0
 5 5     Filip Kuba  24    D     75    9     21    30    -6    28    5     4
 6 6     Darby Hend~ 28    LW    72    18    11    29    1     36    14    3
 7 7     Jim Dowd    32    C     68    7     22    29    -6    80    7     0
 8 8     Antti Laak~ 27    LW    82    12    16    28    -7    24    10    0
 9 9     Stacy Roest 26    C     76    7     20    27    3     20    6     1
10 10    Aaron Gavey 26    C     75    10    14    24    -8    52    9     1
# i 29 more rows
# i 10 more variables: shg <chr>, gwg <chr>, ev <chr>, pp <chr>, sh <chr>,
#   sog <chr>, spct <chr>, toi <chr>, atoi <chr>, awards <chr>
```

```
input_team_year <- function(team, year){

url <- str_c("https://www.hockey-reference.com/teams/", team, "/", year, ".html")
session <- bow(url, force = TRUE)

result <- scrape(session) |>
  html_nodes(css = "table") |>
  html_table(header = TRUE, fill = TRUE)
```

```r
html_table <- result[[4]]

player_stats <- html_table |>
  row_to_names(row_number = 1) |>
  clean_names() |>
  mutate(year = year) |>
  select(player, year, age, pos, gp, pts)
player_stats
}

input_team_year("DAL", "2001")
```

```
No encoding supplied: defaulting to UTF-8.
```

```
# A tibble: 33 x 6
   player              year  age   pos   gp    pts  
   <chr>               <chr> <chr> <chr> <chr> <chr>
 1 Mike Modano         2001  30    C     81    84   
 2 Brett Hull          2001  36    RW    79    79   
 3 Joe Nieuwendyk      2001  34    C     69    52   
 4 Sergei Zubov        2001  30    D     79    51   
 5 Darryl Sydor        2001  28    D     81    47   
 6 Jere Lehtinen       2001  27    LW    74    45   
 7 Brenden Morrow      2001  22    LW    82    44   
 8 Grant Marshall      2001  27    RW    75    37   
 9 Jamie Langenbrunner 2001  25    RW    53    30   
10 Ted Donato          2001  31    LW    65    25   
# i 23 more rows
```

```r
teams <- rep("MIN", 4)
years <- 2001:2004

temp <- map2(teams, years, input_team_year)
```

```
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
```

```
hockey_data4yrs <- list_rbind(temp)
hockey_data4yrs |>
  filter(player != "Team Totals")
```

```
# A tibble: 137 x 6
   player             year age   pos   gp    pts  
   <chr>             <int> <chr> <chr> <chr> <chr>
 1 Scott Pellerin     2001 31    RW    58    39   
 2 Marián Gáborík     2001 18    LW    71    36   
 3 Ľubomír Sekeráš    2001 32    D     80    34   
 4 Wes Walz           2001 30    C     82    30   
 5 Filip Kuba         2001 24    D     75    30   
 6 Darby Hendrickson  2001 28    LW    72    29   
 7 Jim Dowd           2001 32    C     68    29   
 8 Antti Laaksonen    2001 27    LW    82    28   
 9 Stacy Roest        2001 26    C     76    27   
10 Aaron Gavey        2001 26    C     75    24   
# i 127 more rows
```

### 09 - Pause to Ponder - 3 items on NIH News Releases right before the On Your Own section

[**Pause to Ponder:**] Create a function to scrape a single NIH press release page by filling missing pieces labeled `???`:

```
# Helper function to reduce html_nodes() |> html_text() code duplication
get_text_from_page <- function(page, css_selector) {
  page |>
  html_nodes(css_selector) |>
  html_text()
}

# Main function to scrape and tidy desired attributes
scrape_page <- function(url) {
    Sys.sleep(2)
    page <- read_html(url)
    article_titles <- get_text_from_page(page, ".teaser-title")
    article_dates <- get_text_from_page(page, ".date-display-single")
    article_dates <- mdy(article_dates)
    article_description <- get_text_from_page(page, ".teaser-description")
    article_description <- str_trim(str_replace(article_description,
                                                ".*\\n",
```

```r
                                                "")
                                    )

    tibble(
      title = article_titles,
      pub_dates = article_dates,
      description = article_description
    )
}

scrape_page("https://www.nih.gov/news-events/news-releases")
```

```
# A tibble: 10 x 3
   title                                                  pub_dates  description
   <chr>                                                  <date>     <chr>
 1 Kidney transplantation between donors and recipients ~ 2024-10-16 NIH-funded~
 2 Mpox vaccine is safe and generates a robust antibody ~ 2024-10-16 NIH clinic~
 3 NIH and FDA leaders call for innovation in developmen~ 2024-10-15 Commentary~
 4 Alzheimer’s disease may damage the brain in two phases 2024-10-15 NIH-funded~
 5 First wave of COVID-19 increased risk of heart attack~ 2024-10-10 NIH-funded~
 6 NIH launches large study to tackle type 2 diabetes in~ 2024-10-09 Effort to ~
 7 Researchers fully map neural connections of the fruit~ 2024-10-02 NIH-suppor~
 8 Scientists discover gene responsible for rare, inheri~ 2024-09-26 NIH-suppor~
 9 Statement by NIH on Research Misconduct Findings       2024-09-26 NIH takes ~
10 Higher doses of buprenorphine may improve treatment o~ 2024-09-25 NIH-funded~
```

[**Pause to Ponder:**] Use a for loop over the first 5 pages:

```r
pages <- vector("list", length = 5)

for (i in 1:5) {
    Sys.sleep(2)
    base_url <- "https://www.nih.gov/news-events/news-releases"
    if (i==1) {
        url <- base_url
    } else {
        url <- str_c(base_url, "?page=", i-1)
    }
    pages[[i]] <- scrape_page(url)
}
```

```r
df_articles <- bind_rows(pages)
df_articles
```

```
# A tibble: 50 x 3
   title                                                  pub_dates  description
   <chr>                                                  <date>     <chr>
 1 Kidney transplantation between donors and recipients ~ 2024-10-16 NIH-funded~
 2 Mpox vaccine is safe and generates a robust antibody ~ 2024-10-16 NIH clinic~
 3 NIH and FDA leaders call for innovation in developmen~ 2024-10-15 Commentary~
 4 Alzheimer’s disease may damage the brain in two phases 2024-10-15 NIH-funded~
 5 First wave of COVID-19 increased risk of heart attack~ 2024-10-10 NIH-funded~
 6 NIH launches large study to tackle type 2 diabetes in~ 2024-10-09 Effort to ~
 7 Researchers fully map neural connections of the fruit~ 2024-10-02 NIH-suppor~
 8 Scientists discover gene responsible for rare, inheri~ 2024-09-26 NIH-suppor~
 9 Statement by NIH on Research Misconduct Findings       2024-09-26 NIH takes ~
10 Higher doses of buprenorphine may improve treatment o~ 2024-09-25 NIH-funded~
# i 40 more rows
```

**[Pause to Ponder:]** Use map functions in the purrr package:

```r
# Create a character vector of URLs for the first 5 pages
base_url <- "https://www.nih.gov/news-events/news-releases"
urls_all_pages <- c(base_url, str_c(base_url, "?page=", seq(1:4)))

pages2 <- purrr::map(urls_all_pages, scrape_page)
df_articles2 <- bind_rows(pages2)
df_articles2
```

```
# A tibble: 50 x 3
   title                                                  pub_dates  description
   <chr>                                                  <date>     <chr>
 1 Kidney transplantation between donors and recipients ~ 2024-10-16 NIH-funded~
 2 Mpox vaccine is safe and generates a robust antibody ~ 2024-10-16 NIH clinic~
 3 NIH and FDA leaders call for innovation in developmen~ 2024-10-15 Commentary~
 4 Alzheimer’s disease may damage the brain in two phases 2024-10-15 NIH-funded~
 5 First wave of COVID-19 increased risk of heart attack~ 2024-10-10 NIH-funded~
 6 NIH launches large study to tackle type 2 diabetes in~ 2024-10-09 Effort to ~
 7 Researchers fully map neural connections of the fruit~ 2024-10-02 NIH-suppor~
 8 Scientists discover gene responsible for rare, inheri~ 2024-09-26 NIH-suppor~
 9 Statement by NIH on Research Misconduct Findings       2024-09-26 NIH takes ~
10 Higher doses of buprenorphine may improve treatment o~ 2024-09-25 NIH-funded~
# i 40 more rows
```