Homework 3

```
session <- bow("https://www.usclimatedata.com/climate/minneapolis/minnesota/united-states/usn

result <- scrape(session) |>
  html_nodes(css = "table") |>
  html_table(header = TRUE, fill = TRUE)
mpls_data1 <- result[[1]]
mpls_data2 <- result[[2]]
```

[Pause to Ponder:] What is each line of code doing below?

```
#combines into one wide table
bind_cols(mpls_data1, mpls_data2) |>
  #makes into a tibble
  as_tibble() |>
  #deletes the ...8 column
  select(-`...8`) |>
  #shortens names in first column to just the first three words
  mutate(`...1` = str_extract(`...1`, "[^ ]+ [^ ]+ [^ ]+")) |>
  pivot_longer(cols = c(`JanJa`:`DecDe`),
   #makes a longer table where each month has its own row for each ...1 category
               names_to = "month", values_to = "weather") |>
  #makes a wider table where each month has its own row
  pivot_wider(names_from = `...1`, values_from = weather) |>
  #fixes month names to just the first three letters
  mutate(month = str_sub(month, 1, 3))  |>
  #changes var names to be nicer
  rename(avg_high = "Average high in",
         avg_low = "Average low in",
         precip_days = "Days with precipitation",
         avg_precip = "Av. precipitation in",
         avg_snow = "Av. snowfall in")
```

```
New names:
* `` -> `...1`
* `` -> `...8`
```

```
# A tibble: 12 x 6
   month avg_high avg_low precip_days avg_precip avg_snow
   <chr>    <dbl>   <dbl>       <dbl>      <dbl>    <dbl>
 1 Jan         24       9           9       0.89       11
 2 Feb         29      13           7       0.87       10
 3 Mar         42      25           9       1.68        8
 4 Apr         57      38          11       2.91        4
 5 May         69      50          12       3.91        0
 6 Jun         79      60          11       4.58        0
 7 Jul         83      65          10       4.06        0
 8 Aug         81      63           9       4.34        0
 9 Sep         73      54           9       3.02        0
10 Oct         58      41           9       2.58        1
11 Nov         42      28           8       1.61        7
12 Dec         29      15           9       1.17       11
```

```
# Probably want to rename the rest of the variables too!
```

On Your Own

1. In 13_maps.qmd we will see how to create an interactive U.S. map showing population densities by state. Right now, let's see if we can use our new web scraping skills to scrape the correct population density data and create a tidy data frame!

A quick wikipedia search yields this webpage with population densities in a nice table format. Use our 4 steps to `rvest`ing data to acquire the data, and then create a tidy tibble with one row per state.

```
session <- bow("https://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_United_St

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

states <- result[[1]] |>
  select(-3,-6) |>
  filter(row_number()!=1) |>
```

```
  mutate(state_name = str_to_lower(as.character(Location)),
         state_name = str_replace(state_name, "\\[4\\]", ""),
         Density = parse_number(Density),
         Population = parse_number(Population),
         `Land area` = parse_number(`Land area`)) |>
  select(-Location)
states
```

```
# A tibble: 60 x 4
   Density Population `Land area` state_name
     <dbl>      <dbl>       <dbl> <chr>
 1   11131     678972          61 district of columbia
 2    1263    9290842        7354 new jersey
 3    1060    1095962        1034 rhode island
 4     936    3205691        3424 puerto rico
 5     898    7001399        7800 massachusetts
 6     824     172952         210 guam
 7     747    3617176        4842 connecticut
 8     737      98750         134 u.s. virgin islands
 9     637    6180253        9707 maryland
10     578      43915          76 american samoa
# i 50 more rows
```

3. 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 2001 season.

2) Organize your `rvest` code from (1) into functions from the `polite` package.

```r
session <- bow("https://www.hockey-reference.com/teams/MIN/2001.html", force = TRUE)

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

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

```r
hockey_table <- result[[4]] |>
  row_to_names(row_number = 1) |>
  clean_names() |>
  select(player, age, pos, gp, pts) |>
  mutate(age = parse_number(age),
         gp = parse_number(gp),
         pts = parse_number(pts),
         year = 2001)
```

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.

```r
hockey_stats <- function(team, year){
  #make url based on input team and year
  url_name <- str_c("https://www.hockey-reference.com/teams/", team, "/", year, ".html")

  #scrape data
  session <- bow(url_name, force = TRUE)

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

  #extract and clean table
  hockey_table <- result[[4]] |>
    row_to_names(row_number = 1) |>
    clean_names() |>
    select(player, age, pos, gp, pts) |>
    mutate(age = parse_number(age),
           gp = parse_number(gp),
           pts = parse_number(pts),
           year = year) |>
    filter(player != "Team Totals")
```

```
  hockey_table
}

hockey_stats(team = "MIN", 2001)
```

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

```
# A tibble: 38 x 6
   player              age pos      gp   pts  year
   <chr>             <dbl> <chr> <dbl> <dbl> <dbl>
 1 Scott Pellerin       31 RW       58    39  2001
 2 Marián Gáborík       18 LW       71    36  2001
 3 Ľubomír Sekeráš      32 D        80    34  2001
 4 Wes Walz             30 C        82    30  2001
 5 Filip Kuba           24 D        75    30  2001
 6 Darby Hendrickson    28 LW       72    29  2001
 7 Jim Dowd             32 C        68    29  2001
 8 Antti Laaksonen      27 LW       82    28  2001
 9 Stacy Roest          26 C        76    27  2001
10 Aaron Gavey          26 C        75    24  2001
# i 28 more rows
```

4) Use `map2` and `list_rbind` to build one data set containing Minnesota Wild data from 2001-2004.

```
wild_01_04 <- map2("MIN", 2001:2004, hockey_stats) |>
  list_rbind()
```

```
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.
```

```
wild_01_04
```

```
# A tibble: 137 x 6
   player              age pos      gp   pts  year
   <chr>             <dbl> <chr> <dbl> <dbl> <int>
 1 Scott Pellerin       31 RW       58    39  2001
 2 Marián Gáborík       18 LW       71    36  2001
```

```
 3 Ľubomír Sekeráš      32 D         80    34  2001
 4 Wes Walz             30 C         82    30  2001
 5 Filip Kuba           24 D         75    30  2001
 6 Darby Hendrickson    28 LW        72    29  2001
 7 Jim Dowd             32 C         68    29  2001
 8 Antti Laaksonen      27 LW        82    28  2001
 9 Stacy Roest          26 C         76    27  2001
10 Aaron Gavey          26 C         75    24  2001
# i 127 more rows
```

On Your Own:

1. Finishing the NIH example

a) Repeat the process above to extract the description (which include the publication date and the abstract):

```
session <- bow("https://www.nih.gov/news-events/news-releases", force = TRUE)

nih_description <- scrape(session) |>
  html_nodes(".thumbnail-teaser__description") |>
  html_text()
nih_description
```

```
 [1] "September 30, 2025 -   Following an Executive Order by President Trump, HHS will increa
 [2] "September 29, 2025 -   Dr. Letai will serve as NCI's 18th director."
 [3] "September 25, 2025 -   The Standardized Organoid Modeling Center aims to produce standa
 [4] "September 22, 2025 -   Funded projects will drive discoveries across biology, environme
 [5] "September 17, 2025 -   NIH-funded study reveals brain changes long before chronic traum
 [6] "September 17, 2025 -   Approach involves looking at the whole person-
not just separate organs or body systems-and considering multiple factors that promote healt
 [7] "September 16, 2025 -   More than 60% of the nearly 24,000 stillbirth cases annually rem
 [8] "September 12, 2025 -   NIH-funded study supports acupuncture needling as a safe and eff
 [9] "September 10, 2025 -   NIH-funded study demonstrates life-saving potential of providing
[10] "September 3, 2025 -   NIH-funded clinical trial shows potential to simplify treatment f
```

b) Combine these extracted variables into a single tibble, with columns called `title`, `description`, and `pubdate`. Make sure the variables are formatted correctly - e.g. `pubdate` has `date` type, `description` does not contain the `pubdate`, etc.

```
nih_title <- scrape(session) |>
  html_nodes(".thumbnail-teaser__heading") |>
  html_text()
# use tibble() to put multiple columns together into a tibble
nih_top10 <- tibble(title = nih_title,
                    description = nih_description)
nih_top10
```

```
# A tibble: 10 x 2
   title                                                          description
   <chr>                                                          <chr>
 1 "HHS Doubles AI-Backed Childhood Cancer Research Funding"      September ~
 2 "Secretary Kennedy Swears in Dr. Anthony Letai as Director of th~ September ~
 3 "NIH establishes nation's first dedicated organoid development c~ September ~
 4 "NIH launches $50M Autism Data Science Initiative to unlock caus~ September ~
 5 "Repeated head impacts cause early neuron loss and inflammation ~ September ~
 6 "NIH launches landmark project on whole-person health and functi~ September ~
 7 "NIH launches consortium to reduce risk of stillbirth in the U.S~ September ~
 8 "Acupuncture treatment improves disabling effects of chronic low~ September ~
 9 "Treating opioid addiction in jails improves treatment engagemen~ September ~
10 "One dose of antibiotic treats early syphilis as well as three d~ September ~
```

```
# now clean the data
nih_top10 <- nih_top10 |>
  mutate(
    pubdate = str_extract(description, "^.*(   )"),
    pubdate = str_replace(pubdate, " -   ", ""),
    pubdate = mdy(pubdate),
    description = str_replace(description, "^.*(   )", "")
  )
nih_top10
```

```
# A tibble: 10 x 3
   title                                                 description pubdate
   <chr>                                                 <chr>       <date>
 1 "HHS Doubles AI-Backed Childhood Cancer Research Fund~ Following ~ 2025-09-30
 2 "Secretary Kennedy Swears in Dr. Anthony Letai as Dir~ Dr. Letai ~ 2025-09-29
 3 "NIH establishes nation's first dedicated organoid de~ The Standa~ 2025-09-25
 4 "NIH launches $50M Autism Data Science Initiative to ~ Funded pro~ 2025-09-22
 5 "Repeated head impacts cause early neuron loss and in~ NIH-funded~ 2025-09-17
 6 "NIH launches landmark project on whole-person health~ Approach i~ 2025-09-17
```

```
 7 "NIH launches consortium to reduce risk of stillbirth~ More than ~ 2025-09-16
 8 "Acupuncture treatment improves disabling effects of ~ NIH-funded~ 2025-09-12
 9 "Treating opioid addiction in jails improves treatmen~ NIH-funded~ 2025-09-10
10 "One dose of antibiotic treats early syphilis as well~ NIH-funded~ 2025-09-03
```

c) Continue this process to build a tibble with the most recent 50 NIH news releases, which will require that you iterate over 5 webpages! You should write at least one function, and you will need iteration–use both a `for` loop and appropriate `map_()` functions from `purrr`. Some additional hints:

- Mouse over the page buttons at the very bottom of the news home page to see what the URLs look like.
- Include `Sys.sleep(2)` in your function to respect the `Crawl-delay: 2` in the NIH `robots.txt` file.
- Recall that `bind_rows()` from `dplyr` takes a list of data frames and stacks them on top of each other.
- Note that the initial page can be considered "page=0" in the URL

Create a function to scrape a single NIH press release page by filling missing pieces labeled `** FILL IN **`:

```
# 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)
  session <- bow(url, force = TRUE)
  page <- scrape(session)
  nih_title <- get_text_from_page(page, ".thumbnail-teaser__heading")
  nih_description <- get_text_from_page(page, ".thumbnail-teaser__description")
  tibble(title = nih_title, description = nih_description) |>
    mutate(
      pubdate = str_extract(description, "^.*(   )"),
      pubdate = str_replace(pubdate, " -   ", ""),
      pubdate = mdy(pubdate),
      description = str_replace(description, "^.*(   )", "")
    )
```

```
}

# Test your new function
scrape_page("https://www.nih.gov/news-events/news-releases")
```

```
# A tibble: 10 x 3
   title                                                  description pubdate
   <chr>                                                  <chr>       <date>
 1 "HHS Doubles AI-Backed Childhood Cancer Research Fund~ Following ~ 2025-09-30
 2 "Secretary Kennedy Swears in Dr. Anthony Letai as Dir~ Dr. Letai ~ 2025-09-29
 3 "NIH establishes nation's first dedicated organoid de~ The Standa~ 2025-09-25
 4 "NIH launches $50M Autism Data Science Initiative to ~ Funded pro~ 2025-09-22
 5 "Repeated head impacts cause early neuron loss and in~ NIH-funded~ 2025-09-17
 6 "NIH launches landmark project on whole-person health~ Approach i~ 2025-09-17
 7 "NIH launches consortium to reduce risk of stillbirth~ More than ~ 2025-09-16
 8 "Acupuncture treatment improves disabling effects of ~ NIH-funded~ 2025-09-12
 9 "Treating opioid addiction in jails improves treatmen~ NIH-funded~ 2025-09-10
10 "One dose of antibiotic treats early syphilis as well~ NIH-funded~ 2025-09-03
```

d) Use a for loop over the first 5 pages by filling in code where asked:

```
pages <- vector("list", length = 5)
for (i in 0:4) {
  url <- str_c("https://www.nih.gov/news-events/news-releases?page=", i)
  pages[[i + 1]] <- scrape_page(url)
}

df_articles <- bind_rows(pages)
df_articles
```

```
# A tibble: 50 x 3
   title                                                  description pubdate
   <chr>                                                  <chr>       <date>
 1 "HHS Doubles AI-Backed Childhood Cancer Research Fund~ Following ~ 2025-09-30
 2 "Secretary Kennedy Swears in Dr. Anthony Letai as Dir~ Dr. Letai ~ 2025-09-29
 3 "NIH establishes nation's first dedicated organoid de~ The Standa~ 2025-09-25
 4 "NIH launches $50M Autism Data Science Initiative to ~ Funded pro~ 2025-09-22
 5 "Repeated head impacts cause early neuron loss and in~ NIH-funded~ 2025-09-17
 6 "NIH launches landmark project on whole-person health~ Approach i~ 2025-09-17
 7 "NIH launches consortium to reduce risk of stillbirth~ More than ~ 2025-09-16
 8 "Acupuncture treatment improves disabling effects of ~ NIH-funded~ 2025-09-12
```

```
 9 "Treating opioid addiction in jails improves treatmen~ NIH-funded~ 2025-09-10
10 "One dose of antibiotic treats early syphilis as well~ NIH-funded~ 2025-09-03
# i 40 more rows
```

e) Instead, form a final data set by using map functions in the purrr package:

```
# Create a character vector of URLs for the first 5 pages
base_url <- "https://www.nih.gov/news-events/news-releases?page="
urls_all_pages <- str_c(base_url, 0:4)

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

```
# A tibble: 50 x 3
   title                                                  description pubdate
   <chr>                                                  <chr>       <date>
 1 "HHS Doubles AI-Backed Childhood Cancer Research Fund~ Following ~ 2025-09-30
 2 "Secretary Kennedy Swears in Dr. Anthony Letai as Dir~ Dr. Letai ~ 2025-09-29
 3 "NIH establishes nation's first dedicated organoid de~ The Standa~ 2025-09-25
 4 "NIH launches $50M Autism Data Science Initiative to ~ Funded pro~ 2025-09-22
 5 "Repeated head impacts cause early neuron loss and in~ NIH-funded~ 2025-09-17
 6 "NIH launches landmark project on whole-person health~ Approach i~ 2025-09-17
 7 "NIH launches consortium to reduce risk of stillbirth~ More than ~ 2025-09-16
 8 "Acupuncture treatment improves disabling effects of ~ NIH-funded~ 2025-09-12
 9 "Treating opioid addiction in jails improves treatmen~ NIH-funded~ 2025-09-10
10 "One dose of antibiotic treats early syphilis as well~ NIH-funded~ 2025-09-03
# i 40 more rows
```

On Your Own - Best Places

2. Go to https://www.bestplaces.net and search for Minneapolis, Minnesota. This is a site some people use when comparing cities they might consider working in and/or moving to. Using SelectorGadget, extract the following pieces of information from the Minneapolis page:

- property crime (on a scale from 0 to 100)
- minimum income required for a single person to live comfortably
- average monthly rent for a 2-bedroom apartment
- the "about" paragraph (the very first paragraph above "Location Details")

```
session <- bow("https://www.bestplaces.net/city/minnesota/minneapolis", force = TRUE)

crime_mn_data <- scrape(session) |>
  html_nodes(".col-4") |>
  html_text()
crime_mn <- crime_mn_data[[3]] |>
  parse_number()
crime_mn
```

```
[1] 63.3
```

```
comfy_income_mn_data <- scrape(session) |>
  html_nodes(".col-8") |>
  html_text()
comfy_income_mn_int <- comfy_income_mn_data[[1]] |>
  str_extract_all("\\$\\d*,\\d*")
comfy_income_mn <- comfy_income_mn_int[[1]][[2]]
comfy_income_mn
```

```
[1] "$46,800"
```

```
avg_rent_mn_data <- scrape(session) |>
  html_nodes(".mb-2") |>
  html_text()
avg_rent_mn <- avg_rent_mn_data[[8]] |>
  parse_number()
avg_rent_mn
```

```
[1] 1440
```

```
about_mn_data <- scrape(session) |>
  html_nodes(".ms-3") |>
  html_text()
about_mn <- about_mn_data[[1]]
about_mn
```

```
[1] " Minneapolis, Minnesota is a vibrant and bustling city with a rich blend of cultures, a
```