

# Class Activity 16

Your name here

April 30 2024

## Group Activity 1

a. Scrape the first table in `List_of_NASA_missions` wiki page. Additionally, use `janitor::clean_names()` to clean the column names and store the resulting table as `NASA_missions.csv` in your working folder.

```
wiki_NASA <- "https://en.wikipedia.org/wiki/List_of_NASA_missions"

# Scrape the data and write the first table to a CSV file
bow(wiki_NASA) %>%
  scrape() %>%

Error: <text>:7:0: unexpected end of input
5:   scrape() %>%
6:
^
```

b. Now, write a code snippet to scrape all the URLs from the anchor tags (`()`) on a given Wikipedia page, convert the relative URLs to absolute URLs, and store the results in a tibble and save it as `NASA_missions_urls.csv` in your working folder.

```
# Scrape the data and write the URLs to a CSV file
bow(wiki_NASA) %>%
  scrape() %>%

Error: <text>:5:0: unexpected end of input
3:   scrape() %>%
4:
^
```

## Group Activity 2

a. How do you scrape a table from a web page using rvest? After scraping the data, clean the column names with janitor, and prepare the data for analysis in R?

```
yf <- "https://finance.yahoo.com/quote/CL%3DF/history?p=CL%3DF"
bow(yf) %>% scrape()
NULL
```

b. Write the R code to create a time trend plot of opening prices from the scraped data using ggplot2.

```
ggplot(, aes(x = , y = )) +
  geom_line() +
  geom_point() +
  scale_x_date(date_labels = "%b %d, %Y", date_breaks = "1 week") +
  labs(title = "Time Trend of Opening Prices", x = "Date", y = "Opening Price") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Time Trend of Opening Prices

Opening Price

Date

c. How can you transform the data into a long format suitable for plotting multiple price types with ggplot2?

```
ticker_long <- ticker %>%
  pivot_longer(cols = , names_to = , values_to = )
Error in eval(expr, envir, enclos): object 'ticker' not found
```

d. Show how to create a ggplot2 visualization that includes lines and points, with different colors and shapes for each price type, and make the x-axis dates legible.

```
ggplot(ticker_long, aes(x = , y = , color = )) +
  geom_line() +
  geom_point(aes(shape = ), size = 2) + # Different shapes for each price type
  scale_color_manual(values = c("open" = "blue", "close" = "green", "adj_close" = "red")) +
  scale_x_date(date_labels = "%b %d, %Y", date_breaks = "1 week") +
  labs(title = "Time Trend of Stock Prices", x = "Date", y = "Price") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  ) +
  guides(shape = guide_legend(title = "Price Type"), color = guide_legend(title = "Price Type"))
Error in eval(expr, envir, enclos): object 'ticker_long' not found
```

## Group Activity 3

In this activity, you'll scrape web data using `rvest` and tidy up the results into a well-formatted table from this web page. Start by extracting job titles from a given URL, then gather the associated company names, and trim any leading or trailing whitespace from the location data. Next, retrieve the posting dates and the URLs for the full job descriptions. Finally, combine all these elements into a single dataframe, ensuring that each piece of information aligns correctly. Your task is to produce a clean and informative table that could be useful for job seekers. o facilitate the selection of the correct CSS selectors, you may find the `SelectorGadget` Chrome extension particularly useful.

```
url <- "https://realpython.github.io/fake-jobs/"

title <- bow(url) %>% scrape() %>% # part 1
company <- bow(url) %>% scrape() %>% # part 2
location <- bow(url) %>% scrape() %>% # part 3
time <- bow(url) %>% scrape() %>% # part 4
html <- bow(url) %>% scrape() %>% # part 5

# Create a dataframe
tibble(title = title, company = company, location = location, time = time, html = html) # part 6
Error in `tibble()`:
! All columns in a tibble must be vectors.
x Column `` is a `xml_document/xml_node` object.
```